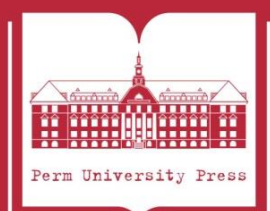


**ПЕРМСКИЙ  
ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ**

## **АКТУАЛЬНЫЕ ПРОБЛЕМЫ МАТЕМАТИКИ, МЕХАНИКИ И ИНФОРМАТИКИ 2023**

**Сборник статей по материалам  
студенческой конференции**



**Пермь 2023**

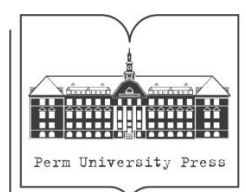
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

# **АКТУАЛЬНЫЕ ПРОБЛЕМЫ МАТЕМАТИКИ, МЕХАНИКИ И ИНФОРМАТИКИ 2023**

*Сборник статей по материалам студенческой конференции*

*г. Пермь, ПГНИУ, 24–28 апреля 2023 г. (1 сессия),  
13–20 июня 2023 г. (2 сессия)*



Пермь 2023

УДК 51+531+004.8](082)

ББК 22+32.81

A437

**Актуальные** проблемы математики, механики и информатики  
A437 2023 [Электронный ресурс] : сборник статей по материалам студенческой конференции (г. Пермь, ПГНИУ, 24–28 апреля 2023 г. (1 сессия), 13–20 июня 2023 г. (2 сессия) / под редакцией М. М. Бузмаковой ; Пермский государственный национальный исследовательский университет. – Электронные данные. – Пермь, 2023. – 15,1 Мб ; 385 с. – Режим доступа: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematiki-mekhaniki-informatiki-2023.pdf>. – Заглавие с экрана.

ISBN 978-5-7944-4043-0

В сборнике представлены материалы студенческой конференции «Актуальные проблемы математики, механики и информатики 2023», которая проводилась 24–28 апреля 2023 г. (1 сессия), 13–20 июня 2023 г. (2 сессия) в г. Перми.

В сборнике представлены работы студентов, выполненные под руководством ученых Пермского государственного национально исследовательского университета. Работы обладают актуальностью и научной новизной, часто они имеют междисциплинарный характер. Большой блок публикаций связан информационными технологиями, так как это направление в настоящее время активно развивается, и в ПГНИУ имеются научные школы, связанные с информационными технологиями.

Сборник предназначен для научных и педагогических работников, преподавателей, аспирантов, магистрантов, студентов и всех, кто интересуется проблемами математики, механики и информатики.

УДК 51+531+004.8](082)

ББК 22+32.81

*Издается по решению кафедры прикладной математики и информатики  
Пермского государственного национального исследовательского университета*

*Рецензенты:* доцент кафедры физико-математического образования Астраханского государственного университета им. В. Н. Татищева, канд. физ.-мат. наук **В. А. Черкасова**

зав. кафедрой «Вычислительная математика, механика и биомеханика» ПНИПУ, д-р техн. наук, профессор **В. Ю. Столбов**

ISBN 978-5-7944-4043-0

© ПГНИУ, 2023

## СОДЕРЖАНИЕ

<b>Агафонова А.А., Аверин С.И.</b> РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ ОНЛАЙН КОНКУРСОВ СТЕНДОВОГО МОДЕЛИЗМА.....	8
<b>Агеев В.А., Кузаев А.Ф.</b> ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПЕРМСКАЯ ЛЮБИТЕЛЬСКАЯ ВОЛЕЙБОЛЬНАЯ ЛИГА».....	14
<b>Баишев Б.А., Селетков И.П.</b> АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ ИНТЕРНЕТ НЕЗАВИСИМОГО ГОЛОСОВОГО ПОМОЩНИКА.....	22
<b>Беляков К.В.</b> РАЗРАБОТКА МЕХАНИЗМОВ ОТОБРАЖЕНИЯ УДАЛЁННОГО РАБОЧЕГО СТОЛА В ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ.....	30
<b>Бердников К.С., Ракина В.Д.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ЭКОЛОГИЧНОЙ УТИЛИЗАЦИИ ОТХОДОВ.....	41
<b>Валиев А.Д., Ракина В.Д.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ АБИТУРИЕНТА ДЛЯ ВЫБОРА ФАКУЛЬТЕТА И НАПРАВЛЕНИЯ В ПГНИУ ПРИ ПОСТУПЛЕНИИ.....	52
<b>Васильев И.Д., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ВРЕМЕНИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ В КОМПЬЮТЕРНОМ КЛАССЕ.....	60
<b>Васильев Г.Е., Кузаев А.Ф.</b> ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ЧАТ-БОТА ДЛЯ СОЗДАНИЯ НАПОМИНАНИЙ.....	72
<b>Вахрушев Г.С., Кушев В.О.</b> ТРЕБОВАНИЯ К РАСПИСАНИЮ В СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ ПРИ АВТОМАТИЗАЦИИ ЕГО СОСТАВЛЕНИЯ.....	78
<b>Вдовин И.Е., Бузмакова М.М.</b> КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СЛУЧАЙНОЙ ПЛОТНОЙ УПАКОВКИ НЕСЖИМАЕМЫХ ОКРУЖНОСТЕЙ НА ПЛОСКОСТИ.....	85



<b>Верещагина М.Н., Аверин С.И.</b> ЦИФРОВИЗАЦИЯ РИТЕЙЛА: ПРОЕКТ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОБЛАЧНОГО МАГАЗИНА САМООБСЛУЖИВАНИЯ.....	90
<b>Власов А.А.</b> МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ ИНФЕКЦИОННОГО ЗАБОЛЕВАНИЯ НА ОСНОВЕ КЛЕТОЧНЫХ АВТОМАТОВ.....	95
<b>Волков И.И., Ромашкина Т.В.</b> ВИЗУАЛИЗАТОР АЛГОРИТМОВ РАБОТЫ С ДИНАМИЧЕСКИМИ СТРУКТУРАМИ ДАННЫХ (ОДНОНАПРАВЛЕННЫЕ СПИСКИ).....	107
<b>Гавгянен А.А., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ АВТОМАТИЧЕСКОЙ ФИКСАЦИИ ПРИСУТСТВИЯ УЧАЩИХСЯ НА ЗАНЯТИЯХ В УНИВЕРСИТЕТЕ.....	113
<b>Деменьтев М.А., Данилова Е.Ю.</b> ОПТИМИЗАЦИЯ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ВСТРОЕННОГО ДИАГНОСТИРОВАНИЯ ПЛИС НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА STM32F401RE.....	118
<b>Демидов М.А., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВЫДАЧИ СИЗ УЧАЩИМСЯ И ПРЕПОДАВАТЕЛЯМ В ПЕРИОД ЭПИДЕМИЧЕСКОЙ УГРОЗЫ.....	123
<b>Демин Э.Ф., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭЛЕКТРОННЫЙ ПРОЕЗДНОЙ».....	129
<b>Дружинина Е.В., Раевский В.Н.</b> ERP-СИСТЕМЫ РОССИЙСКОГО ПРОИЗВОДСТВА.....	135
<b>Епишина Н.В., Селетков И.П.</b> РАЗРАБОТКА БИБЛИОТЕКИ ДЛЯ РЕАЛИЗАЦИИ МАТРИЧНОГО НЕЧЁТКОГО ЛОГИЧЕСКОГО ВЫВОДА НА ЯЗЫКЕ PYTHON.....	140
<b>Ермаков В.А., Селетков И.П.</b> БИБЛИОТЕКА PYTHON ДЛЯ РАЗРАБОТКИ НЕЙРО-НЕЧЁТКИХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ (НА ПРИМЕРЕ ЗАДАЧИ ДИАГНОСТИКИ САХАРНОГО ДИАБЕТА).....	146
<b>Захаров А.И., Селетков И.П.</b> WEB-ИНТЕРФЕЙС СИСТЕМЫ ГРАФИЧЕСКОГО МОДЕЛИРОВАНИЯ....	156

<b>Зольников В.В., Кузаев А.Ф.</b> ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ПРОЕКТОВ ОРГАНИЗАЦИИ.....	163
<b>Колодкина М.В., Аверин С.И.</b> ОПЫТ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ВЕТЕРИНАРНОЙ КЛИНИКИ.....	169
<b>Колпашиков М.Н., Кнутова Н.С.</b> ПРОЕКТИРОВАНИЕ ОНЛАЙН-СЕРВИСА С API ДЛЯ ЗАПОЛНЕНИЯ ДОКУМЕНТОВ ПО ШАБЛОНАМ.....	178
<b>Костинын В.В., Кнутова Н.С.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОЙ ТРАНСКРИБАЦИИ РЕЧИ.....	188
<b>Кострикин Е.А., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ УЧЕТА ЗАБОЛЕВАЕМОСТИ В ПЕРМИ.....	197
<b>Кузаев А.Ф., Шестаков К.Н.</b> ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ «КАЧЕСТВО ВОДЫ В РОДНИКАХ ПЕРМСКОГО КРАЯ».....	203
<b>Кузовов Д.Д., Аверин С.И.</b> ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ БИЗНЕС-АНАЛИЗА.....	211
<b>Кулаков А.М., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ БЕСКЛЮЧЕВОГО ДОСТУПА ПРЕПОДАВАТЕЛЕЙ В АУДИТОРИИ УНИВЕРСИТЕТА.....	215
<b>Ладыгин В.А., Ладыгин В.А., Иванов И.Б., Степанов В.А.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ ПОДГОТОВКИ АНАЛИТИКОВ.....	221
<b>Лаптев И.С., Кузаев А.Ф.</b> ПРОЕКТИРОВАНИЕ АЛГОРИТМА ПОИСКА ПУТИ В КОМПЬЮТЕРНЫХ ИГРАХ.....	229
<b>Лязгин Д.С., Ракина В.Д.</b> АВТОМАТИЗАЦИИ ДОКУМЕНТООБРАЗОВАНИЯ НА ПРЕДПРИЯТИИ ПО РЕМОНТУ ЭЛЕКТРОДВИГАТЕЛЕЙ.....	234

<b>Маткин М.А., Аверин С.И.</b> ПРЕПОДАВАТЕЛИ ПЕРМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА 1916–2023: ПРОЕКТ ПРОСОПОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	243
<b>Миргалеев А.А., Аверин С.И.</b> ОПЫТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ «АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ КАРТ КОНДИТЕРСКИХ ИЗДЕЛИЙ ДЛЯ КОНДИТЕРА».....	248
<b>Мосеева Ю.С., Огурецкая А.И.</b> ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭЛЕКТРОННАЯ КУЛИНАРНАЯ КНИГА».....	254
<b>Музафаров В.И., Кнутова Н.С.</b> ПРОЕКТИРОВАНИЕ СИСТЕМЫ АГРЕГАЦИИ RSS-КАНАЛОВ (RSS-РИДЕР).....	259
<b>Наумов В.Г., Кнутова Н.С., Зайцев А.А., Кулакова С.А.</b> ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА УЧЕБНОГО СИМУЛЯТОРА: «НЕФТЯНОЕ МЕСТОРОЖДЕНИЕ: ОХРАНА ОКРУЖАЮЩЕЙ СРЕДЫ»...	267
<b>Низаметдинов Р.Э., Аверин С.И.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭКОЛОГИЧЕСКИЙ ТУРИЗМ ПЕРМСКОГО КРАЯ».....	272
<b>Николаев А.П., Анисимова С.И.</b> ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕРКИ РАСПИСАНИЯ НА СООТВЕТСТВИЕ НОРМАМ.....	280
<b>Перебиковский К.А., Ракина В.Д.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ МОБИЛЬНОЙ СИСТЕМЫ ОПОВЕЩЕНИЯ ЭКСТРЕННЫХ СЛУЖБ О ДОРОЖНО- ТРАНСПОРТНЫХ ПРОИСШЕСТВИЯХ.....	284
<b>Писцов С.М.</b> РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ СИНТЕЗА И РЕДАКТИРОВАНИЯ ИЗОБРАЖЕНИЙ С ПРИМЕНЕНИЕМ МОДЕЛЕЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА.....	291
<b>Полынский А.А., Ракина В.Д.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ «РАДИО ПГУ».....	297
<b>Пономарев А.Ф., Кнутова Н.С.</b> ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ СБОРА WHOIS ДАННЫХ ДОМЕННЫХ ИМЕН И ИХ АНАЛИЗА.....	306

<b>Русаков В.В., Мокунева В.Е., Селетков И.П.</b> РАЗРАБОТКА NO-CODE СИСТЕМЫ ОБРАБОТКИ И ХРАНЕНИЯ ДАННЫХ.....	312
<b>Садилов М.В., Кнутова Н.С.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ РЕЧЕВОЙ ФИЛЬТРАЦИИ.....	318
<b>Семёнов С.П., Саломатов Д.А., Селетков И.П.</b> РАЗРАБОТКА СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ И РОБОТИЗАЦИИ РАБОЧИХ ПРОЦЕССОВ.....	324
<b>Соколов А.В., Соколова О.Л., Яшичев Д.Л.</b> МОДЕЛЬ ЦИФРОВОЙ ПЛАТФОРМЫ ДОСТАВКИ ПРОДУКТА: ПОЛИМОРФИЗМ ОСНОВНЫХ МОДУЛЕЙ, АВТОНОМНОСТЬ И СПОСОБНОСТЬ К ИНТЕГРАЦИИ В СТОРОННИЕ ЦИФРОВЫЕ ЭКОСИСТЕМЫ.....	329
<b>Софронов А.М., Русаков С.В.</b> ИССЛЕДОВАНИЕ ОСОБЕННОСТИ ВЕСЕННИХ ПАВОДКОВЫХ РЕЖИМОВ НА РЕКАХ КАМА И КОСА НА ОСНОВЕ ГИДРОЛОГИЧЕСКИХ ДАННЫХ.....	335
<b>Филиппов С.И., Аверин С.И.</b> ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПЕРМСКИЕ БЫЛИЧКИ. МИФОЛОГИЧЕСКИЕ РАССКАЗЫ 19-20 ВВ».....	347
<b>Филиппов А.С., Раевский В.Н.</b> ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ЭЛЕКТРОННОГО ГОЛОСОВАНИЯ С ВОЗМОЖНОСТЬЮ ПРОВЕРКИ РЕЗУЛЬТАТА ИЗБИРАТЕЛЯМИ.....	350
<b>Черемных А.Н.</b> РАЗРАБОТКА СИСТЕМЫ АНАЛИЗА И ПРОЕКТИРОВАНИЯ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВЫСШЕГО ОБРАЗОВАНИЯ НА БАЗЕ ОНТОЛОГИЧЕСКОГО ПРЕДСТАВЛЕНИЯ КОМПЕТЕНЦИЙ И ДИСЦИПЛИН.....	356
<b>Шубинцева Я.Е., Василюк Н.Н.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ АНАЛИЗА ИНВЕСТИЦИОННЫХ ПРОЕКТОВ.....	368
<b>Япаров А.Э., Ракина В.Д.</b> ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ТЕСТИРОВАНИЯ ЗНАНИЙ ПО ДИСЦИПЛИНЕ «ЯЗЫКИ ПРОГРАММИРОВАНИЯ».....	374

## **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ ОНЛАЙН-КОНКУРСОВ СТЕНДОВОГО МОДЕЛИЗМА**

*Агафонова Анастасия Александровна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, bparty2004@gmail.com

*Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, averin-s-i@yandex.ru

Рассматриваются основные средства для проектирования, документирования и разработки информационной системы для проведения онлайн-конкурсов по моделированию. Разработка такой системы актуальна, поскольку она позволит расширить географию участников, предоставив новые возможности для общения и совместного творчества для энтузиастов моделирования. Система позволит организаторам анализировать результаты предыдущих конкурсов и улучшать качество мероприятий, добавляя новые категории моделирования или создавая новый формат конкурса. В работе также описаны требования пользователей, функциональные требования, проектирование базы данных и тестирование системы.

Ключевые слова: стендовый моделизм, информационная система, пользовательский интерфейс.

Стендовые модели – это масштабные копии, которые точно отображают свой прототип, будь то автомобиль, корабль или самолет, выполненный согласно своему историческому или действующему оригиналу [1].

Разработка системы для проведения онлайн конкурса стендового моделизма является актуальной поскольку обычно такого рода конкурсы проводятся на музейных площадках или тематических клубах в формате очного мероприятия, однако не для всех участников есть возможность посетить данный проект. Безусловно проводятся и онлайн конкурсы, но в качестве платформы часто используются различные социальные сети, которые весьма неудобны для массового формата проведения [2, 3]. Сами мероприятия помогают развивать и поддерживать интерес к моделированию, а также способствуют обмену опытом между участниками. Создание системы для проведения онлайн конкурсов стендового моделизма позволит повысить качество и доступность данного направления, а главное сделать это мероприятие еще более массовым.

Объектом исследования являются онлайн конкурсы стендового моделизма. Предметом исследования – информационная система для проведения онлайн конкурсов стендового моделизма.

Цель работы: проектирование, документирование информационной системы для проведения онлайн конкурсов стендового моделизма.

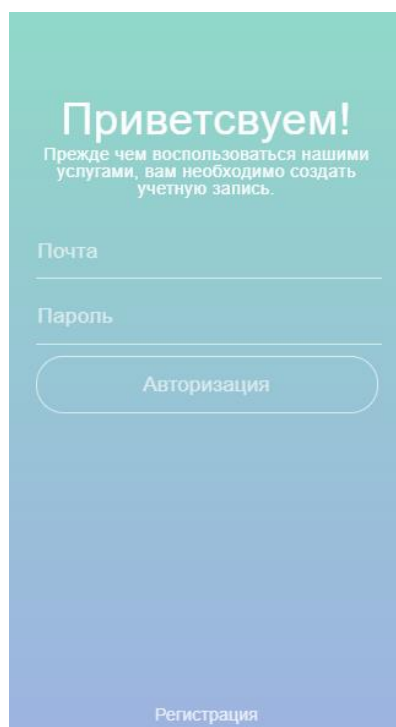
Разрабатываемая информационная система может включать в себя функционал для регистрации участников, загрузки фотографий и описаний работ, оценки работ экспертами и публичного голосования за лучшую работу. Также система может иметь встроенную систему комментариев, обсуждений и возможности для определения победителей и их награждение.

В последующей перспективе одним из основных преимуществ такой системы является автоматизация процесса оценки моделей. Это позволит ускорить процесс и повысить точность результатов. Кроме того, система может хранить информацию о предыдущих конкурсах, что даст возможность организаторам анализировать результаты и улучшать качество мероприятий, добавляя новые категории моделирования или создавая новый формат конкурса.

Были разработаны функциональные требования, которые будут описывать процесс работы системы:

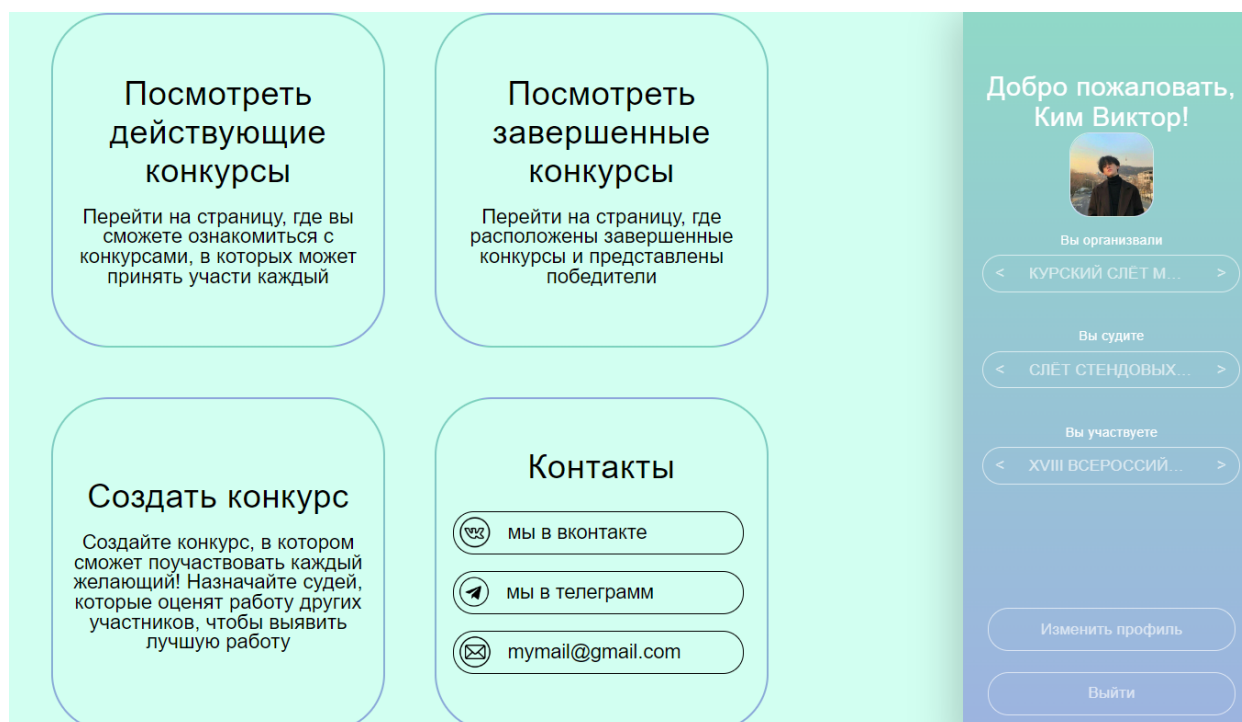
- Система должна иметь 4 типа уровня пользователя (участник, организатор, судья, администратор), которым будут предоставляться функционал в соответствии с уровнем доступа.
- Доступ поиска и просмотра информации о конкурсных проектах всем пользователям, а также просмотр конкурсных работ.
- Предоставление пользователю роли организатора после создания конкурса.
- Предоставление пользователю роли участника после публикации работы на конкурс.
- Предоставление роли судьи после того, как пользователь подаст заявку на судейство мероприятия, а организатор ее одобрит.
- Возможность участникам совершать следующие манипуляции с информацией о конкурсной работе (название, описание, вид модели, масштаб, изображения): добавление, изменение, удаление.
- Возможность организаторам совершать следующие манипуляции с информацией о конкурсном проекте (название, краткое описание, полное описание, категория моделирования, изображение, критерии оценивания, максимальная оценка по критерию, даты проведения): добавление, изменение, удаление.
- Возможность организаторам совершать следующие манипуляции с заявками на судейство конкурса: утверждение, отказ.
- Возможность судьям совершать следующие манипуляции с информацией об оценке за конкурсную работу: добавление, изменение, удаление.
- Возможность администраторам совершать следующие манипуляции с информацией о конкурсе, работе участника: добавление, изменение, удаление.
- Возможность авторизованным пользователям совершать следующие манипуляции с информацией учетной записи (логин, пароль, изображение профиля): изменение.
- Автоматический подсчет баллов конкурсной работы по критериям.
- Вывод результатов конкурса после завершения (исходя из дат проведения) и определение трех призовых мест, наибольшему количеству баллов.

При входе в систему появляется окно аторизации или регистрации для пользователя. Неавторизованный пользователь не может просматривать информацию, поэтому ему следует зарегистрироваться или пройти авторизацию на соответствующих страницах (см. рис. 1).



**Рис. 1. Страница авторизации**

На рисунке (см. рис. 2) представлена главная страница, где пользователь может просмотреть мероприятия в которых он участвовал. Затем выбрав одну из кнопок пользователь может перейти на страницы с действующими конкурсами, завершенными конкурсами и стать организатором собственного конкурса.



**Рис. 2. Главная страница**

Например, на вкладке с действующими конкурсами на рисунке (см. рис. 3) предоставляется краткая информация о конкурсе: название и даты проведения. Наведя на поле с конкретным конкурсом, можно увидеть краткое описание конкурса. Также на самой странице можно произвести фильтрацию конкурсов по категории моделирования.



Рис. 3. Действующие конкурсы

Выбрав интересующее мероприятие, пользователь может рассмотреть подробное описание, критерии оценивания, организатора и судью. Затем нажав кнопку «Положение» будет скачиваться файл с положением конкурса, где более подробно прописаны правила участия. Пользователь может подать заявку на участие или судейство конкурса.

На рисунке (см. рис. 4) промотав страницу ниже, можно увидеть выставленные работы участников с кратким описанием работы, именем участника, видом модели, масштабом, выставленной оценкой и датой загрузки.



Рис. 4. Страница конкурса 2

Чтобы рассмотреть модель детально пользователю нужно нажать на само изображение миниатюры.

Если пользователь решает принять участие в мероприятии и выложить свою модель для оценивания, то на странице конкурса, он нажимает кнопку «Отправить модель» и затем он переходит на страницу «Шаблон создания модели». Он вводит необходимую информацию об изделии: название, вид модели, масштаб, описание и загружает несколько изображений для дальнейшего визуального оценивания.



Для судьи на странице с моделью высвечиваются критерии оценивания, по которым судья выставляет баллы, также имеется возможность просмотра изображений миниатюры конкурсанта на рисунке (см. рис. 5) для визуальной оценки.

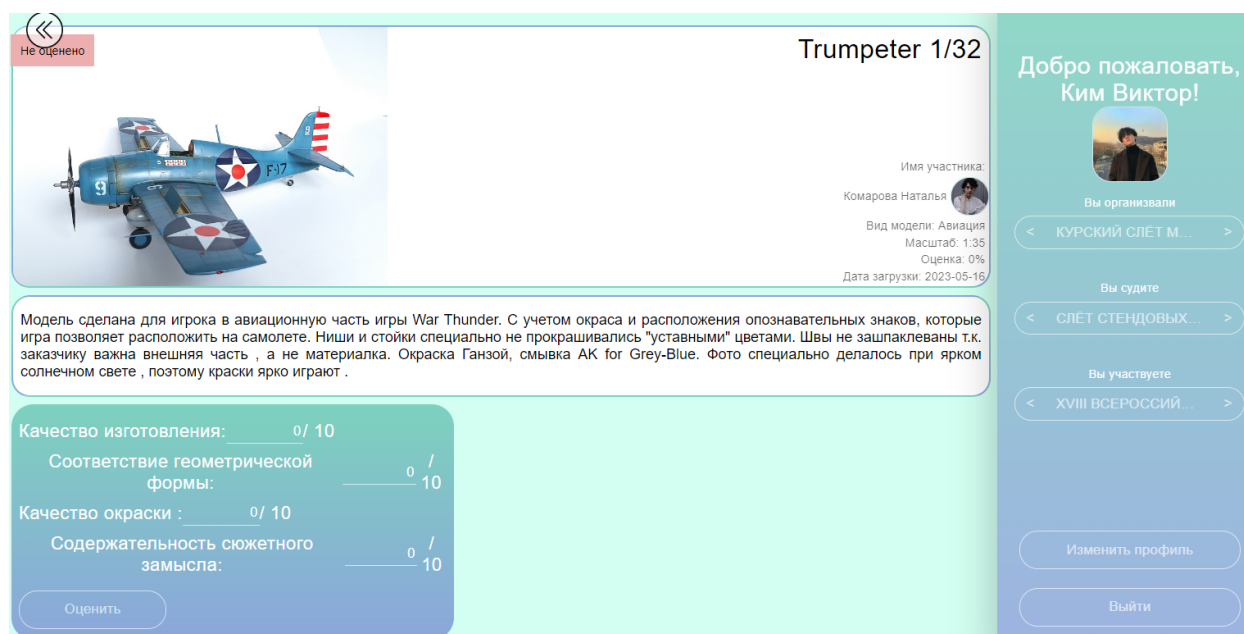


Рис. 5. Оценка конкурсной работы

Переход к шаблону формирования мероприятия осуществляется через главную страницу. Там заполняется информация о мероприятии, также возможно принимать или отвергать заявки на судейство.

Через главную страницу можно посмотреть завершенные конкурсы. Результаты конкурса показываются в виде трех призовых мест.

В результате проделанной работы была спроектирована и реализована информационная система для проведения онлайн конкурсов стендового моделизма. Данная информационная система может быть использована для всех пользователей, интересующихся моделизмом или планирующих проводить конкурсные работы.

### Библиографический список

1. История стендового моделизма [Электронный ресурс]. – URL: <https://collection.rin.ru/cgi-bin/article.pl?id=1234> (дата обращения: 3.03.2023).
2. Начался прием заявок на XVI Выставку-конкурс стендовых моделей и миниатюры «День Победы» 2021 г., посвященной 80-летию разгрома немецко-фашистских войск под Москвой [Электронный ресурс]. – URL: <https://stend-modelist.club/news/nachalsya-priem-zayavok-na-xvi-vyhstavku-konkurs-stendovykh-modelej-i-miniatury-den-pobedyh-2021-g> (дата обращения: 11.04.2023).
3. ОТКРЫТЫЙ МЕЖДУНАРОДНЫЙ ОНЛАЙН КОНКУРС МОДЕЛИСТОВ «КУЗНИЦА МАСТЕРОВ – 2021» [Электронный ресурс]. – URL: <https://stend-modelist.club/events/otkrytyh-mezhdunarodnyh-onlajn-konkurs-modelistov-k-uznica-masterov-2021> (дата обращения: 11.04.2023).

### References

1. The history of bench modeling [Electronic resource]. – URL: <https://collection.rin.ru/cgi-bin/article.pl?id=1234> (accessed: 3.03.2023).

2. The acceptance of applications for the XVI Exhibition-competition of poster models and miniatures "Victory Day" 2021, dedicated to the 80th anniversary of the defeat of the Nazi troops near Moscow, has begun [Electronic resource]. – URL: <https://stend-modelist.club/news/nachalsyapriem-zayavok-na-xvi-vyhstavku-konkurs-stendovyhkh-modelej-i-miniatyuryh-den-pobedyh-2021-g> (accessed: 11.04.2023).

3. OPEN INTERNATIONAL ONLINE MODELING CONTEST "FORGE OF MASTERS – 2021" [Electronic resource]. – URL: <https://stend-modelist.club/events/otkrytyhj-mezhdunarodnyhj-onlajn-konkurs-modelistov-k-uznica-masterov-2021> (accessed: 11.04.2023).

## **DEVELOPMENT OF AN INFORMATION SYSTEM FOR CONDUCTING ONLINE-CONTESTS OF BENCH MODELING**

*Agafonova Anastasia A.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
bparty2004@gmail.com

*Averin Sergey I.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
averin-s-i@yandex.ru

Design and documentation of an information system for conducting online competitions on diorama modeling. The development of such a system is relevant because it will expand the geography of participants, providing new opportunities for communication and joint creativity for enthusiasts of diorama modeling. The system will allow organizers to analyze the results of previous contests and improve the quality of events by adding new modeling categories or creating a new competition format. The paper also describes user requirements, functional requirements, database design and system testing.

**Keywords:** bench modeling, information system, user interface

## **ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПЕРМСКАЯ ЛЮБИТЕЛЬСКАЯ ВОЛЕЙБОЛЬНАЯ ЛИГА»**

*Агеев Владислав Александрович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, brafelinefun@gmail.com

*Кузаев Айдар Файзуллович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kuzaevaydar@mail.ru

Рассматривается создание информационной системы для пермской любительской волейбольной лиги, включая ее проектирование, разработку и документирование. Исследуются уже существующие информационные системы с аналогичным назначением. На основе анализа этих систем формулируются основные функциональные требования к разрабатываемому приложению. Описывается обоснованный выбор инструментов для проектирования и разработки, а также создание прототипа системы. При проектировании применяется язык моделирования UML и CASE-средство Diagrams.net. В роли СУБД выступает MySQL для хранения данных. Для разработки серверной части системы используется язык программирования PHP, а для придания пользовательскому интерфейсу динамики – язык JavaScript. Ключевые слова: информационная система, проектирование, волейбол, любительская лига, Пермский край.

### **Введение**

В современном постиндустриальном обществе, информационные технологии уже прочно вошли в жизнь большого количества людей. От самых простейших систем, где можно посмотреть информацию о расписании междугородних автобусов или посадочных страниц для малого бизнеса и до крупных систем с агрегацией нескольких онлайн площадок, возможностью заказать товары онлайн и сразу же отследить процесс доставки. Таким образом можно заключить, что информационные системы сейчас нужны для многих, если не для всех видов деятельности человека – для упрощения и автоматизации процессов с этой деятельностью связанных. Что же касается спорта, за последнее время все большее количество россиян увлекаются этим видом деятельности [1], будь то самостоятельные тренировки или наоборот, командные виды спорта, в том числе и волейбол. Значит тема работы актуальна и для объекта исследования ВКР. Ведь чем больше людей вовлечено, тем большее количество трудозатрат необходимо вложить в организацию процессов и тем большему количеству людей необходимо доставлять информацию, а значит вполне уместно использование информационной системы для пермской любительской волейбольной лиги.

Объект исследования – любительская волейбольная лига.

Предметная область работы – информационная система для пермской волейбольной лиги.

Цель работы – спроектировать и реализовать прототип информационной системы для пермской волейбольной лиги.

В ходе выполнения работы будут решены следующие задачи:

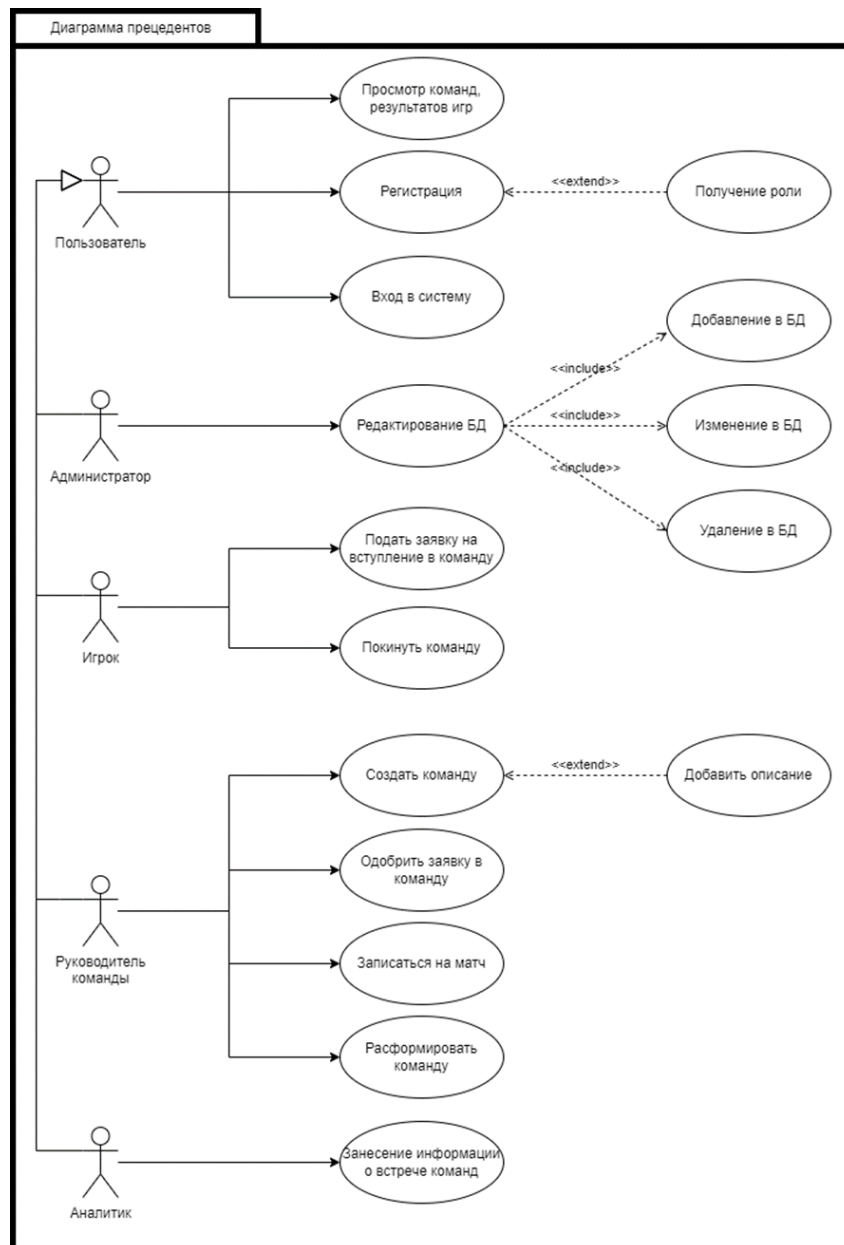
- изучение предметной области любительских лиг и волейбола, как спорта в целом;
- обзор существующих информационных систем аналогов в изучаемой области;
- выделение основных функций, которые необходимы в проектируемой ИС;
- выбор средств для проектирования ИС и методологии;
- создание модели ИС;
- создание прототипа интерфейса;
- реализация прототипа ИС;
- выполнение обзора уязвимостей системы и способов их устранения.

### **Моделирование системы**

Рассмотрим процесс проектирования и моделирования информационной системы с использованием языка графического описания UML (Unified Modeling Language) [2]. Было принято решение начать моделирование системы с определения основных функциональных возможности и сценариев ее использования. Эти элементы могут быть представлены в виде прецедентов, которые описывают конкретные действия и поведение системы.

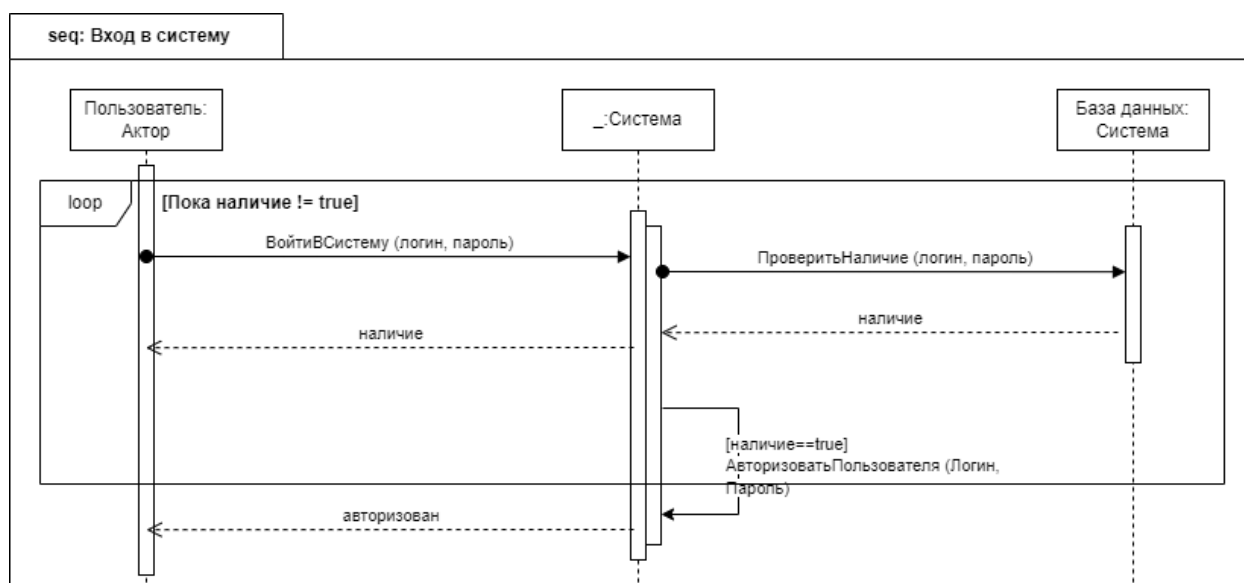
В моделируемой информационной системе присутствуют пять акторов. Первым актором является пользователь, который состоит в отношении обобщения со всеми остальными актерами и отражает базовые действия, которые может совершить любой посетитель ИС. Вторым актором является администратор базы данных, ответственный за задачи, связанные с управлением базы данных. Третий актер – Игрок, как неотъемлемая часть любого спорта и, как следствие, нашей системы. Имеет возможности взаимодействия с командами. Четвертым актором является Руководитель команды, как лицо, которое представляющее и ответственное за действия всей команды. И, наконец, пятый актер – это аналитик, который отражает все действия на встрече команд в системе.

Каждый из этих акторов имеет свои роли и обязанности в системе, и их взаимодействие играет важную роль в функционировании ИС. Акторов и прецеденты с ними связанные иллюстрирует рис. 1.



**Рис. 1. Диаграмма прецедентов**

На диаграмме последовательностей, представленной на рис. 2, показано, как актер "Пользователь" вызывает операцию "Войти в систему". Вместе с вызовом передаются и параметры – логин и пароль, введенные пользователем. Система запрашивает у базы данных, существует ли запись с таким логином и паролем и, если запись не найдена, система сообщает об этом пользователю и предлагает повторить попытку входа. Таким образом, устанавливается цикл, в котором пользователь и, в том числе другие акторы, не может взаимодействовать с системой, пока не будут введены правильные данные – логин и пароль. Когда данные верны, система выполняет операцию аутентификации пользователя и возвращает сообщение об успешной авторизации.



**Рис. 2. Диаграмма последовательностей – Вход в систему**

Следующая диаграмма, которая была создана в рамках проектирования системы – диаграмма разведывания. Она состоит из нескольких кубов – так визуализируются элементы аппаратного комплекса, которые представляют вычислительные устройства и среду выполнения. Внутри кубов есть более точная спецификация характеристик. Так, на диаграмме разветвления на рис. 3 имеется два куба с аннотацией <<device>>, которые указывают на устройство, внутри которого будет выполняться программа. Так как система реализует клиент серверную архитектуру, то существует два основных куба – устройство пользователя и устройство, которое обеспечивает работу сервера. Устройство пользователя не подразумевает каких-либо специфичных требований, однако, спецификация исключает использование браузера Internet Explorer в силу того, что этот браузер требует особых трудозатрат с точки зрения разработки.

Второе устройство – арендованный виртуальный выделенный сервер, с операционной системой Linux. Внутри находится куб среды выполнения – HTTP Server Apache, при помощи которого обрабатываются все входящие запросы в режиме реального времени, конфигурируется доступность файлов и предоставляются файлы HTML, CSS и JS для клиента. На сервере находятся выполняемые скрипты языка PHP и файлы .htaccess, которые являются конфигурационными и влияют на работу самого HTTP сервера. Также на сервере будет развернута СУБД MySQL, на что указывает <<database>>. Взаимодействие между клиентом и сервером осуществляется через HTTPS протокол, который является надстройкой над HTTP, расширенный шифрованием передаваемого через него трафика при помощи SSL/TLS протоколов, что важно как с точки зрения индексации поисковых систем, так и безопасности системы в целом.

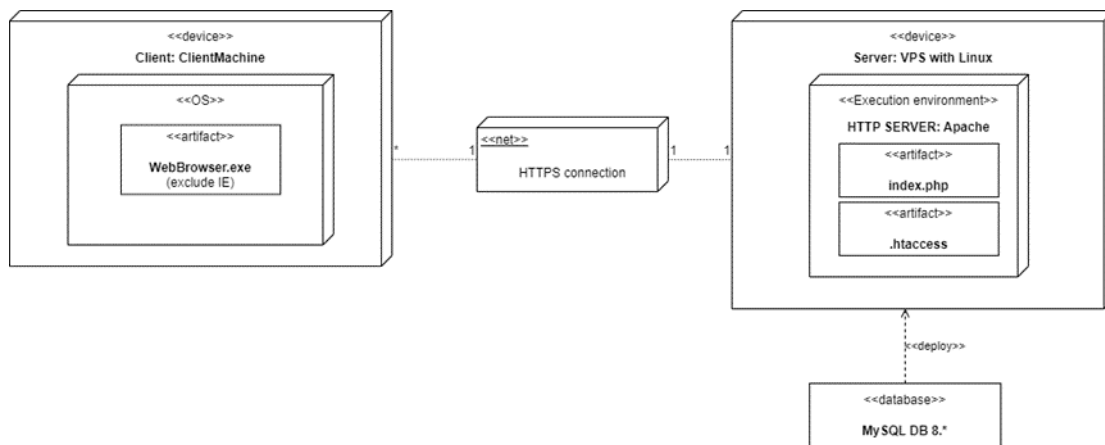


Рис. 3. Диаграмма развертывания

### Разработка информационной системы

Перед разработкой ИС была реализована схема данных для реляционной базы данных (представлена на рис. 4). В ней будут храниться данные о пользователях; игроках; командах; отношении между командой и игроком (бывший игрок команды; текущий игрок команды; игрок, который только подал заявку в команду); о встречах команд; о площадке, на которой проводится встреча и о ярких моментах встреч.

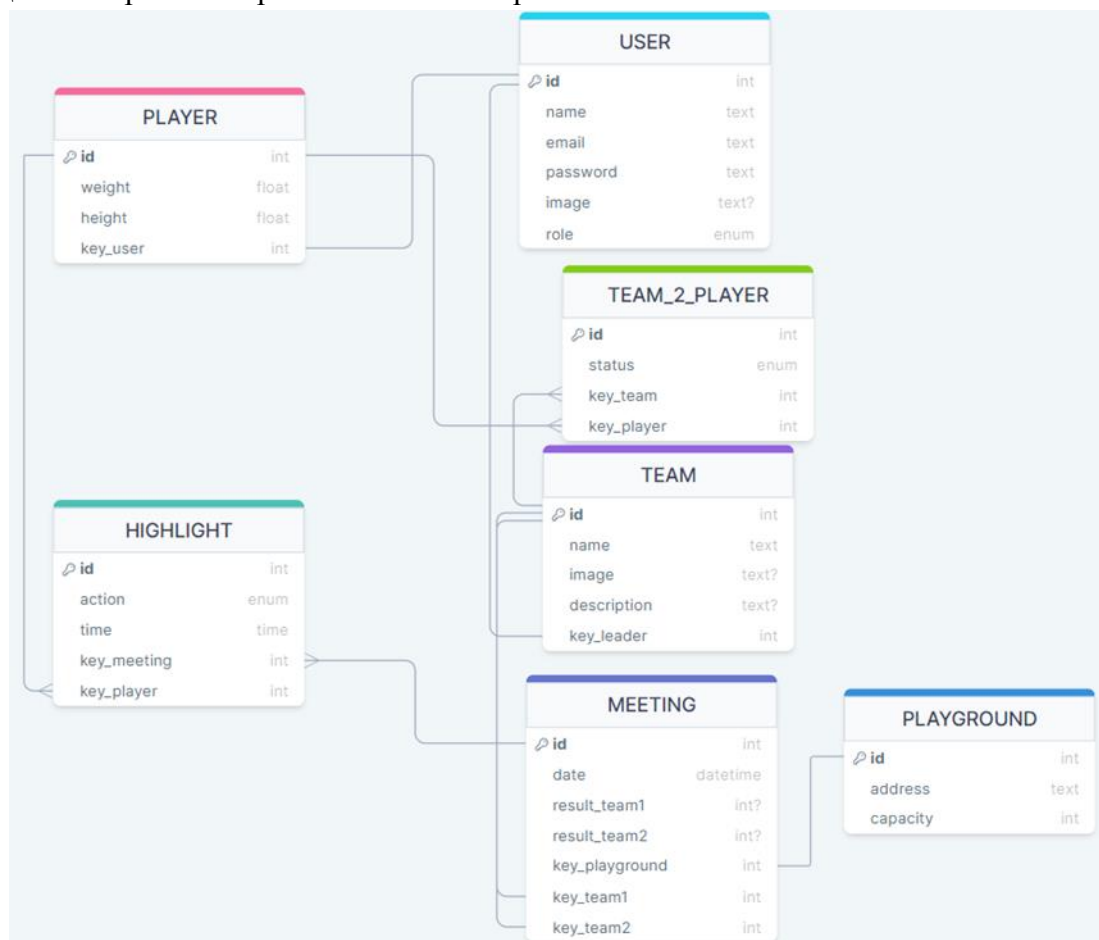


Рис. 4. Реляционная схема данных

Пользовательский интерфейс является непосредственным способом взаимодействия между системой и пользователями, которые имеют самые разные устройства для получения доступа к ИС, расположенной в сети интернет. Поэтому все страницы интерфейса полностью

адаптированы под набор разрешений от маленьких мобильных устройств в вертикальном положении, до горизонтальных мониторов и телевизоров с высоким разрешением. Пример адаптации интерфейса показан на рис. 5 и рис. 6. Интерфейс подстраивается и под разные роли, так в личном кабинете, к примеру, пользователь имеет возможность только изменять свои данные, а руководитель клуба также может и принять новых игроков. Администратор ИС имеет доступ к панели, в которой может редактировать базу данных в пределах сохранения целостности данных.

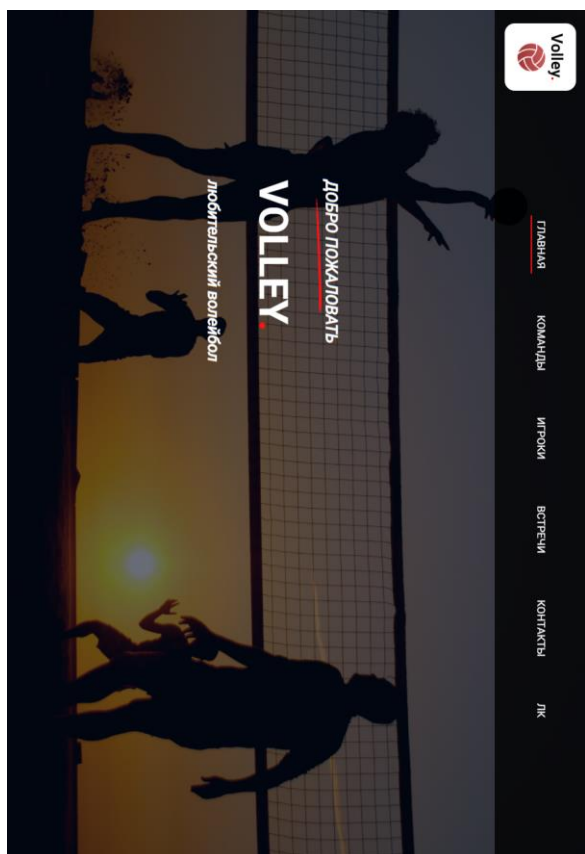


Рис. 5. Главная страница

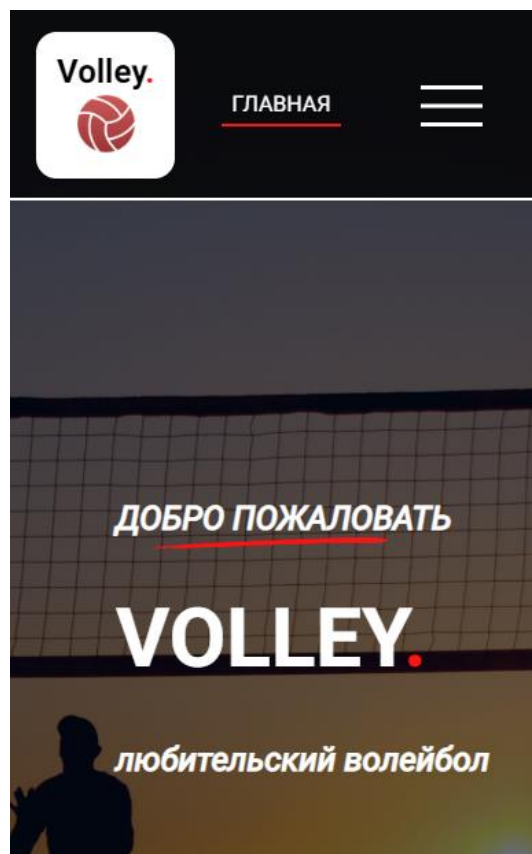


Рис. 6. Главная страница (моб)

При разработке ИС, на языке PHP была реализована почти вся логика приложения, будь то вход в систему или отправка запроса на добавление в команду, а также PHP в некоторой степени влияет на структуру DOM. Информация о пользователе хранится в сессиях, которые PHP реализует, а информация хешировалась при помощи API PHP[3], который для этого предназначен. Для обращения к базе данных использовалась библиотека mysqli, позволяющая взаимодействовать с СУБД MySQL, а логика по работе с базами данных была инкапсулирована в класс, чтобы совершать меньше ошибок при взаимодействии. Также наиболее важные данные, которые необходимы из любой части приложения, были вынесены в отдельный подключаемый файл конфигурации.

В ходе реализации приложения были проработаны и некоторые технические моменты, связанные с уменьшением количества уязвимостей, что является актуальной темой для ресурса, который доступен из сети Интернет и открыт для всех желающих. Были выделены следующие пункты:

- Конфигурация файла веб-сервера .htaccess [4].
- Защита от SQL-инъекций (mysqli – prepared statements) [5].



- Хеширование паролей (PHP API).
- Ограничение прав пользователей БД (выдача только CRUD прав для пользователя ИС).
- Использование тела запроса (POST) для отправки всех чувствительных данных.

### **Заключение**

В результате проделанной работы была спроектирована и разработана информационная система для пермской любительской волейбольной лиги. Система может работать на разных устройствах и имеет разделение функциональности по ролям пользователей. Было проведено исследование уязвимостей системы и рассмотрены меры по защите. При должном развитии, приложение может быть интересно организаторам и прочим участникам любительской лиги в пермском крае.

### **Библиографический список**

1. Все больше россиян занимаются спортом или физкультурой – Ведомости [Электронный ресурс]. – «Ведомости» – ведущее деловое издание России. Режим доступа: <https://www.vedomosti.ru/society/articles/2023/02/14/962911-bolshe-rossiyan-zanimayutsya-sportom>, свободный (Дата обращения 13.03.2023).
2. *Ларман К.* Применение UML 2.0 и шаблонов проектирования. – 3-е изд. – М.: Вильямс, 2006. – 736 с.
3. PHP: функции хеширования паролей – Manual [Электронный ресурс]. – PHP: Hypertext processor. Режим доступа: <https://www.php.net/manual/ru/ref.password.php>, свободный. (дата обращения 16.04.2023).
4. Apache HTTP Server Tutorial: .htaccess files [Электронный ресурс]. – The Apache HTTP Project. Режим доступа: <https://httpd.apache.org/docs/current/howto/htaccess.html>, свободный (дата обращения 19.04.2023).
5. PHP: подготавливаемые запросы – Manual [Электронный ресурс]. – PHP: Hypertext processor. Режим доступа: <https://www.php.net/manual/ru/mysqli.quickstart.prepared-statements.php>, свободный (дата обращения 01.05.2023).

### **References**

1. Vse bolshe rossiyan zanimayutsya sportom ili fizkulturoy – Vedomosti [Electronic resource]. – «Vedomosti» – vedushee delovoe izdanie Rossii. URL: <https://www.vedomosti.ru/society/articles/2023/02/14/962911-bolshe-rossiyan-zanimayutsya-sportom>, (Accessed 13.03.2023). (In Russ.)
2. *Larman K.* Primenenie UML 2.0 I shablonov proektirovaniya – 3-e izd. – M.: Vilyams, 2006. – 736 s. (In Russ.)
3. PHP: password hashing functions – Manual [Electronic resource]. – PHP: Hypertext processor. URL: <https://www.php.net/manual/ru/ref.password.php>,. (Accessed 16.04.2023).
4. Apache HTTP Server Tutorial: .htaccess files [Electronic resource]. – The Apache HTTP Project. URL: <https://httpd.apache.org/docs/current/howto/htaccess.html>, (Accessed 19.04.2023).
5. PHP: prepared statements – Manual [Electronic resource]. – PHP: Hypertext processor. URL: <https://www.php.net/manual/ru/mysqli.quickstart.prepared-statements.php>, (Accessed 01.05.2023).

# **DESIGN, DEVELOPMENT AND DOCUMENTATION OF THE INFORMATION SYSTEM FOR THE PERM AMATEUR VOLLEYBALL LEAGUE**

*Ageev Vladislav A.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, brafelinefun@gmail.com

*Kuzaev Aidar F.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, kuzaevaydar@mail.ru

The creation of an information system for the Perm amateur volleyball league is considered, including its design, development and documentation. Existing information systems with a similar purpose are reviewed. Based on the analysis of these systems, the main functional requirements for the developed application are formulated. Reasonable choice of tools for design and development are described and system prototype is created. The UML modeling language and the Diagrams.net CASE tool are used in process of system designing. MySQL acts as a DBMS for data storage. The PHP programming language is used to develop the server part of the system, and the JavaScript language is used to give the user interface dynamics.

Keywords: information system, design, volleyball, amateur league, Perm region.

## **АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ ИНТЕРНЕТ-НЕЗАВИСИМОГО ГОЛОСОВОГО ПОМОЩНИКА**

*Баишев Борис Андреевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, [baishev.boris@mail.ru](mailto:baishev.boris@mail.ru)

*Селетков Илья Павлович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, [iseletkov@gmail.com](mailto:iseletkov@gmail.com).

В работе формулируется задача построения интернет-независимого голосового помощника, позволяющего строить автономные системы управления умными домами. Приводятся требования, предъявляемые к устройствам такого класса. Далее указываются основные этапы процесса обработки голосовых команд. Производится обзор и сравнительный анализ существующих готовых решений и современных бесплатных инструментов и технологий, применимых для преобразования информации на каждом из этапов обработки. В конце предлагается микросервисная архитектура программного обеспечения для построения автономного контроллера умного дома с встроенным голосовым помощником.

Ключевые слова: голосовой помощник, Python, микросервисная архитектура, распознавание речи, генерация речи, нейронные сети.

### **Введение**

Низкая рыночная цена и массовость производства микрочипов позволили людям создавать устройства интеллектуального управления в новые сферы деятельности, например, в так называемых «умных домах». Умное устройство – это электронное устройство, обычно подключаемое с помощью беспроводных сетей, которое может работать в некоторой степени автономно и интерактивно [1].

На рынке умных устройств есть самые различные устройства: от чайников и термометров, до звуковых систем и подушек. «Умный» дом объединяет в себе множество «умных устройств», образующих систему, управляемую единым центром. Также, в планах у учёных есть идея о построении умного кампуса или даже университета [2].

Однако, интерфейс взаимодействия пользователя с «умным» домом зачастую остается традиционным, с помощью ручного ввода команд или с помощью графической консоли, что не во всех случаях является удобным. В некоторых случаях удобнее использовать голосовые команды для управления устройствами. С данной задачей могут помочь голосовые помощники, такие как: Cortana, Siri, Алиса, Google Assistant, Google Now, Alexa, Viv, Ozlo и д.р.. Из всех перечисленных только Siri и Алиса могут понимать русскую речь и взаимодействовать с устройствами умного дома. Однако, Siri доступна только для устройств на операционной системе iOS, а Алисе для каждого запроса необходима постоянная подключение к интернету, так как каждый запрос производится через поисковую систему, и весь функционал представляет из себя модифицированное взаимодействие с умной поисковой строкой в браузере [3]. А значит для любого, даже самого незначительного запроса отсутствие интернета стано-

вится непреодолимой преградой. Так же постоянная передача пользовательских и телеметрических данных умного дома в интернет влечёт за собой угрозу нарушения безопасности и конфиденциальности.

В итоге, принято решение создать голосового помощника для управления умным домом и умными вещами, который, на сколько это возможно, не требует доступа в интернет.

### **Постановка задачи**

Для создания качественной системы требуется предварительное создание её проекта (логической модели) и построение прототипа, на котором можно будет производить тестирование.

Основные требования к системе:

- поддержка русской речи;
- отсутствие необходимости подключения к сети интернет для управления устройствами «умного» дома;
- небольшие аппаратные требования (потенциально аппаратная реализация на операционной системе Linux на базе платформы Raspberry Pi);
- использование только бесплатных инструментов.

Для учёта существующих наработок в данной предметной области необходимо рассмотреть, оценить достоинства и недостатки существующих готовых решений. Также из их описания можно понять принципы работы, алгоритмы обработки данных, используемые инструменты.

После составления перечней доступных инструментов на каждый этап обработки команд необходимо их проанализировать и выбрать наиболее подходящие под перечисленные выше условия.

Далее требуется построить логическую модель системы с учётом требований и функциональных характеристик всех выбранных инструментов.

### **Обзор существующих решений**

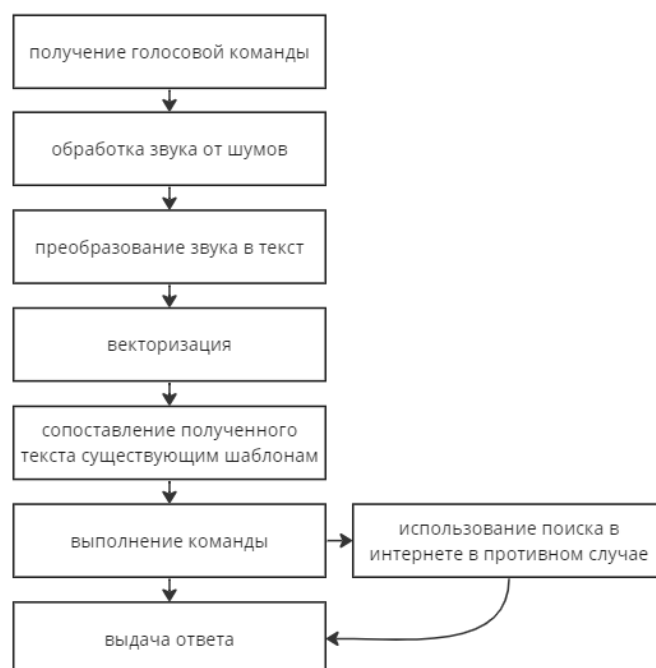
На данный момент, существует несколько схожих решений. Так, например, в работе [4] авторы спроектировали и построили голосового помощника для взаимодействия с операционной системой Windows на русском языке, однако библиотека для распознавания голоса использует интернет, и голосовой помощник может взаимодействовать только с Windows. В работе [5] голосовой помощник рассчитан на работу с Raspberry Pi и взаимодействием с умными устройствами, не требует подключения к интернету, но рассчитан на работу с английским языком.

Автор статьи [6] смог реализовать голосовой помощник для управления умным устройством, не требующий подключения к интернету, на языке программирования C#, однако архитектура приложения является монолитной и плохо расширяемой.

### **Выбор инструментария**

Для создания приложения решено использовать язык программирования Python, так как, благодаря своей гибкости он позволит быстро создать и настроить прототип будущего приложения. Кроме того, все основные инструменты обработки речи и текста, описанные ниже, разработаны в виде библиотек именно для использования в программах, написанных на этом языке.

На рис. 1 приведён целевой алгоритм обработки голосовых команд.



**Рис. 1. Алгоритм обработки команд в голосовом помощнике**

1. Получение голосовой команды может быть реализовано как постоянное прослушивание и ожидание специальной команды, например, «Эй, Гугл» или как нажатие клавиши и голосовой ввод команды. Каждый из этих методов имеет недостаток, постоянное прослушивание звука – это потенциальная угроза безопасности для пользователя, а использование ввода с помощью дополнительных действий снижает удобство использования.

2. Автоматическая обработка звука от посторонних шумов может иметь негативный эффект, поэтому целесообразно провести отдельное исследование эффективности существующих библиотек.

3. Распознавание речи и преобразование её в текст может быть выполнено как с помощью готовых библиотек, так и с помощью собственной нейросетевой модели, которую необходимо будет обучить на большом количестве данных.

4. Выполнение команды предполагает использование ОС Raspberry Pi, однако нужно предоставить пользователю возможность использовать команды операционной системы Windows и macOS.

5. Выдача ответа пользователю должна быть реализована как с помощью голоса, так и с помощью текста, причём голос должен быть реализован на разных языках и потенциально обладать разными голосами.

Для распознавания речи можно или самим придумать решение, или использовать готовые библиотеки. Так как существует огромное количество вариаций готовых решений, создавать новое является не совсем целесообразным. Различные облачные сервисы, например, Google API, обладают высоким качеством по сравнению с аналогами [7], но всё же требуют подключения к интернету, и поэтому данный вариант не подходит для решения поставленной задачи.

Библиотека Speech Recognition [8] предназначена для обработки голосовых сообщений и приведению их в текстовый формат. Модель Vosk [9], работу с которой поддерживает Speech Recognition, предоставляет более качественное распознавание голосов на многих популярных языках. Достоинства и недостатки указаны в табл. 1. Для реализации системы выбрана легковесная модель Vosk.

Таблица 1. Инструменты для преобразования речи в текст

Название	Достоинства	Недостатки
Speech recognition	Поддержка многих популярных языков; поддержка google API, CMU Sphinx, Pocket Sphinx, Julius; бесплатность; может использовать как аудиофайлы формата FLAC, так и запись звука с микрофона	Наличие интернета как необходимость; Ограниченный выбор языков; большие требования к звуковому файлу; качество распознавания без подключения к интернету низкое
Vosk маловесная	Малый размер; распознаёт русскую речь; бесплатность; возможность дообучения	Часто неправильно понимает слова; сильно зависит от шумов в звуке [10]
Vosk тяжеловесная	Распознаёт русскую речь; качество распознавания гораздо выше, включает в себя несколько диалектов одного языка; бесплатность; возможность дообучения	Большой размер; долго загружается

Далее этап очистки текста можно выполнить стандартными средствами языка Python. На этапе векторизации тоже доступно несколько хороших, но альтернативных решений, приведённых в табл. 2.

Таблица 2. Инструменты для векторизации текста

Название	Достоинства	Недостатки
Bag of Words	Простота в использовании; подходит для небольших корпусов текстов; не требует много вычислительных ресурсов; учитывает частотность слов; хорошо работает на задачах классификации текстов.	Не учитывает порядок слов; не учитывает семантику слов; не работает на задачах генерации текста; не учитывает контекст.
TF-IDF	Учитывает важность слов; учитывает частотность слов в документе и корпусе текстов; подходит для больших корпусов текстов; хорошо работает на задачах классификации текстов.	Не учитывает семантику слов; не работает на задачах генерации текста; не учитывает контекст.
Embedding	Учитывает семантику слов; учитывает контекст; подходит для больших корпусов текстов; хорошо работает на задачах классификации текстов; подходит для задач генерации текста.	Требует много вычислительных ресурсов; не подходит для небольших корпусов текстов; требует большого количества данных для обучения; может быть сложен в использовании.
fuzzywuzzy	Бесплатность; высокая точность; простота использования; поддержка многопоточности.	Большое время выполнения; ограниченный выбор методов; возможная утечка памяти.

Из перечисленных вариантов в табл. 2, для того чтобы уменьшить, которая может появиться при распознавании голосовых команд, можно использовать библиотеку fuzzywuzzy [11]. Данная библиотека использует расстояние Левенштейна для нечёткого сравнения строк.

Для преобразования текста в речевые ответы подходят библиотека pyttsx3 [12] и синтезатор речи RHVoice [13], особенности которых указаны в табл. 3 Для синтеза речи на русском языке выбран синтезатор речи RHVoice, голос с именем «Ирина».

Таблица 3. Инструменты для синтеза речи

Название	Достоинства	Недостатки
pyttsx3	Легкость использования; гибкость настроек; бесплатность; не требует интернета; может работать с драйверами разных операционных систем	Низкое качество звука; ограниченный выбор языков; необходимость установки дополнительных зависимостей
RHVoice	Открытый исходный код; бесплатность; более высокое качество синтезируемой речи на русском языке по сравнению с pyttsx3	

### Построение структуры решения

На рис. 2 в виде блоков представлена целевая архитектура программно-аппаратного комплекса для решения поставленной задачи. У каждого из блоков своя зона ответственности. Для программной части системы можно использовать микросервисную архитектуру, позволяющую осуществлять разработку и обновление программы по частям, внести модульность в систему [14].

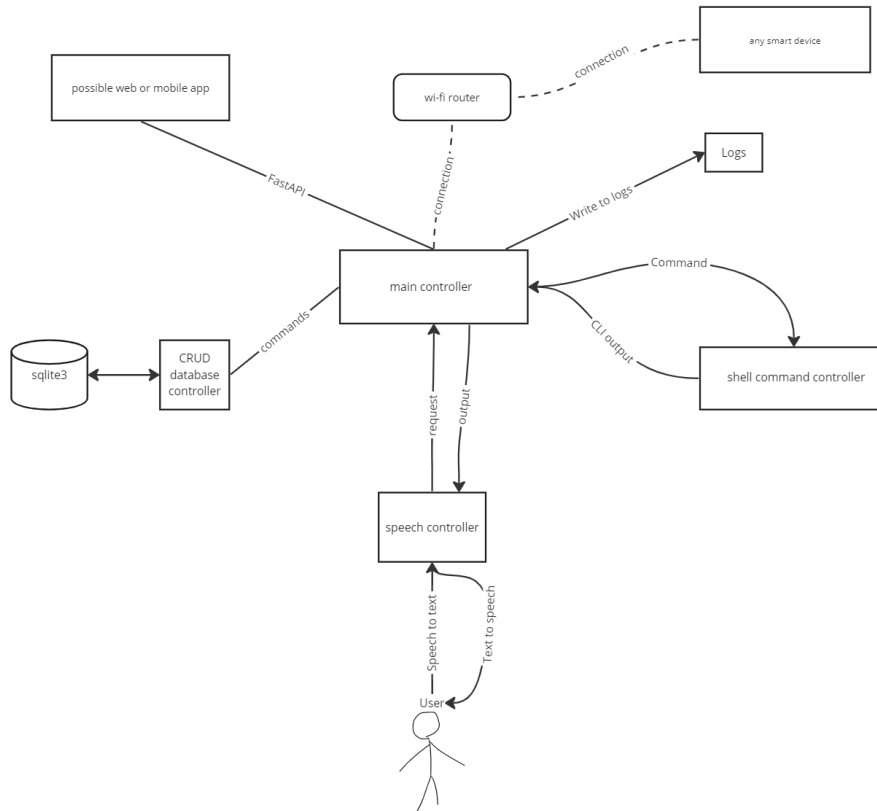


Рис. 2. Архитектура приложения

В первую очередь, в независимости от выбранной архитектуры, нам нужен сервис для взаимодействия с базой данных. В ней будут храниться шаблоны голосовых команд. Пользователь сможет самостоятельно создать шаблоны команд и привязать к ним соответствующие фрагменты исполняемого кода для взаимодействия с устройствами «умного» дома. Также база данных должна содержать таблицу пользователей для распознавания по голосу и таблицу устройств, в которой будут храниться параметры взаимодействия (протоколы, адреса, порты и т.д.).

Для настройки системы планируется разработать специальное приложение для мобильных устройств.

Блок, отвечающий за распознавание и синтез речи, не должен использовать подключение к интернету и при этом достаточно быстро обрабатывать запросы и отвечать на них.

Выполнение команд производится с помощью библиотеки OS, позволяющей запускать команды на различных операционных системах.

Код основного блока должен отвечать за то, чтобы связывать разные блоки для возможности сохранять модульную структуру приложения.

Блок логирования должен отвечать за запись чтение логов.

Блок команд отвечает за выполнение команд под разными операционными системами.

Так же в будущем возможно расширение проекта: добавление чат бота для взаимодействия преимущественно на мобильных устройствах, а подключение к умным устройствам должно быть реализовано через роутер.

### **Заключение**

В данной работе был проведён анализ существующей проблемы: отсутствия интернет независимого голосового помощника для управления умными вещами. Были рассмотрены схожие готовые решения, был проанализирован ряд инструментов для построения программы, была спроектирована архитектура приложения.

### **Библиографический список**

1. Smart device // Википедия [Электронный ресурс]. – URL: [https://en.wikipedia.org/wiki/Smart\\_device](https://en.wikipedia.org/wiki/Smart_device) (дата обращения: 17.04.2023).
2. «Умный» кампус: когда сами стены учат [Электронный ресурс]. – URL: <https://lala.lanbook.com/umnyj-kampus-kogda-sami-steny-uchat> (дата обращения: 17.04.2023).
3. *Ахмаева Л.Г.* Пользовательский опыт и возможности использования голосовых ассистентов в интерактивных маркетинговых коммуникациях: Amazon Alexa, Google home, Apple Siri, Яндекс Алиса // Вестник университета. 2020. № 5. С. 49–57.
4. *Скочко П.С., Барабанов В.Ф., Гребенникова Н.И., Кенин С.Л.* Голосовой помощник для управления операционной системой // Вестник Воронежского государственного технического университета. 2022. Т. 18. № 2. С. 13–16.
5. *Pazhamalai R. и др.* IoT Based Voice Assistant using Raspberry Pi and Natural Language Processing // Материалы международной конференции International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2022. С. 1–4. DOI:10.1109/ICPECTS56089.2022.10046890
6. Собственный голосовой помощник off-line // Хабр [Электронный ресурс]. – URL: <https://habr.com/ru/articles/657769/> (дата обращения: 16.04.2023).
7. *Filippidou F., Moussiades L.* A Benchmarking of IBM, Google and Wit Automatic Speech Recognition Systems // Artificial Intelligence Applications and Innovations IFIP Advances



in Information and Communication Technology. / под ред. I. Maglogiannis, L. Iliadis, E. Pimenidis. Cham: Springer International Publishing, 2020. С. 73–82.

8. SpeechRecognition: Library for performing speech recognition, with support for several engines and APIs, online and offline // PyPI [Электронный ресурс]. – URL: <https://pypi.org/project/SpeechRecognition/> (дата обращения: 17.04.2023).

9. vosk: Offline open source speech recognition API based on Kaldi and Vosk. [Электронный ресурс]. – URL: <https://alphacephei.com/vosk/> (дата обращения 16.04.2023).

10. NTA. Офлайн распознавание речи. Библиотека Vosk // vc.ru [Электронный ресурс]. – URL: <https://vc.ru/dev/247450-oflayn-raspoznavanie-rechi-biblioteka-vosk> (дата обращения: 16.04.2023).

11. fuzzywuzzy // PyPI [Электронный ресурс]. – URL: <https://pypi.org/project/fuzzywuzzy/> (дата обращения: 23.04.2023).

12. pyttsx3: Text to Speech (TTS) library for Python 2 and 3. [Электронный ресурс]. – URL: <https://pyttsx3.readthedocs.io/en/latest/> (дата обращения: 23.04.2023).

13. RHVoice/RHVoice: a free and open source speech synthesizer for Russian and other languages [Электронный ресурс]. – URL: <https://github.com/RHVoice/RHVoice> (дата обращения: 23.04.2023).

14. Микросервисы // Хабр [Электронный ресурс]. – URL: <https://habr.com/ru/articles/249183/> (дата обращения: 16.04.2023).

### References

1. Smart device // Wikipedia [Electronic resource]. – URL: [https://en.wikipedia.org/wiki/Smart\\_device](https://en.wikipedia.org/wiki/Smart_device) (accessed 04/23/2023).

2. "Smart" campus: when the walls themselves teach [Electronic resource]. – URL: <https://lala.lanbook.com/umnyj-kampus-kogda-sami-steny-uchat> (accessed 04/17/2023).

3. *Akhmaeva L.G.* User experience and opportunities in the use of voice assistants in interactive marketing communications: Amazon Alexa, Google Home, Apple Siri, Yandex Alice // *Vestnik Universiteta*. 2020. No. 5. pp. 49-57. DOI: 10.26425/1816-4277-2020-5-49-57 (In Russ.)

4. *Skochko P.S., Barabanov V.F., Grebennikova N.I., Kenin S.L.* Voice assistant for operation system management // *Bulletin of the Voronezh State Technical University*. 2022. V. 18. No. 2. pp. 13–16.

5. *Pazhamalai R. et al.* IoT Based Voice Assistant using Raspberry Pi and Natural Language Processing // *Proceedings of International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, 2022, pp. 1–4. DOI:10.1109/ICPECTS56089.2022.10046890.

6. Sobstvenniy golosovoi pomoshchnik off-line // Habr [Electronic resource]. – URL: <https://habr.com/ru/articles/657769/> (accessed 04/16/2023).

7. *Filippidou F., Moussiades L.* A Benchmarking of IBM, Google and Wit Automatic Speech Recognition Systems // *Artificial Intelligence Applications and Innovations IFIP Advances in Information and Communication Technology*. / ed. I. Maglogiannis, L. Iliadis, E. Pimenidis. Cham: Springer International Publishing, 2020. pp. 73–82.

8. SpeechRecognition: Library for performing speech recognition, with support for several engines and APIs, online and offline // PyPI [Electronic resource]. – URL: <https://pypi.org/project/SpeechRecognition/> (accessed 04/17/2023).

9. vosk: Offline open source speech recognition API based on Kaldi and Vosk. [Electronic resource]. – URL: <https://alphacephei.com/vosk/> (accessed 04/16/2023).

10. NTA. Offline speech recognition. Vosk Library // vc.ru [Electronic resource]. – URL: <https://vc.ru/dev/247450-oflayn-raspoznavanie-rechi-biblioteka-vosk> (accessed 04/16/2023).
11. fuzzywuzzy // PyPI [Electronic resource]. – URL: <https://pypi.org/project/fuzzywuzzy/> (accessed 04/23/2023).
12. pyttsx3: Text to Speech (TTS) library for Python 2 and 3. [Electronic resource]. – URL: <https://pyttsx3.readthedocs.io/en/latest/> (accessed 04/23/2023).
13. RHVoice/RHVoice: a free and open source speech synthesizer for Russian and other languages [Electronic resource]. – URL: <https://github.com/RHVoice/RHVoice> (accessed 04/23/2023).
14. Microservices // Habr [Electronic resource]. – URL: <https://habr.com/ru/articles/249183/> (accessed 04/16/2023).

## **ANALYSIS OF TOOLS FOR CREATING INTERNET INDEPENDENT VOICE ASSISTANT**

*Baishev Boris A.*

Perm State University, 15 Bukireva St., 614990, Perm, [baishev.boris@mail.ru](mailto:baishev.boris@mail.ru)

*Seletkov Ilya P.*

Perm State University, 15 Bukireva St., 614990, Perm, [iseletkov@gmail.com](mailto:iseletkov@gmail.com).

The paper formulates the task of building an Internet-independent voice assistant that allows you to build autonomous control systems for smart homes. Authors give the requirements for this device class. The following are the main stages of the processing of voice commands. The review and comparative analysis of existing ready-made solutions and modern free tools and technologies applicable to the transformation of information at each stage of processing is carried out. A micro-service software architecture is proposed for building an autonomous smart home controller with a built-in voice assistant at the end.

**Keywords:** voice assistant, Python, microservice architecture, speech recognition, speech generation, neural network.

## РАЗРАБОТКА МЕХАНИЗМОВ ОТОБРАЖЕНИЯ УДАЛЁННОГО РАБОЧЕГО СТОЛА В ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

*Беляков Кирилл Викторович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kiribel2000@gmail.com

В данной статье представлено описание разработки системы, реализующей клиентскую часть протокола VNC (Virtual Network Computing) в виде плагина графического движка Unreal Engine 4. Система предоставляет возможности по отображению удаленного рабочего стола в виртуальной реальности. Предусмотрена гибкая настройка качества передаваемого изображения: качество сжатия JPEG, сила сжатия zlib, выбор глубины цвета. Также в работе представлено описание использования плагина для отображения удаленного рабочего стола на объекте-плоскости среды Unreal Engine 4. Выполнено сравнение наиболее популярных VNC серверов по качеству передаваемого видеопотока и сформулированы рекомендации по организации серверной части. В отличие от других VNC клиентов, данная система интегрирована со средой виртуальной реальности, что позволяет внедрять удаленный рабочий стол в настраиваемое виртуальное окружение. Реализованная система упростит сбор данных, необходимых для исследований рекламных баннеров с целью поиска оптимального решения для создания рекламы. При использовании технологий отслеживания взгляда и виртуальной реальности возможно собирать данные о том, на какие зоны баннера обращено больше всего внимания и какое окружение больше всего подходит для его привлечения. Для нахождения оптимального дизайна баннера необходимо постоянное внесение правок. Система поможет отображать рекламные баннеры динамически, с возможностью их изменения в реальном времени на стороне сервера, что ускорит процесс проведения экспериментов и позволит выполнять правки дизайна в привычном для дизайнера окружении. В дальнейшем планируется реализация кодировки H.264, добавление функциональности управления удаленным рабочим столом и интеграция с инструментом Creative Maps Studio.

Ключевые слова: виртуальная реальность, удаленный рабочий стол, Unreal Engine 4, Virtual Network Computing, Remote Framebuffer Protocol.

### **Введение**

В настоящее время существует множество технологий для демонстрации виртуального окружения в реальном времени и интерактивного взаимодействия с ним. Одним из самых популярных приложений подобного рода является игровой движок Unreal Engine 4 [1]. Он позволяет моделировать игровые сцены из готовых 3D моделей и вносить в них интерактивность при помощи скриптов на встроенном визуальном языке Blueprint, либо же при помощи написания программного кода на C++. Однако программная модификация свойств объектов не всегда позволяет добиться желаемого результата за требуемое время. Рассмотрим, например, следующую ситуацию. Заказчик имеет 3D сцену городского района, на которой стоит рекламный билборд. Дизайнер подготавливает макеты рекламы для этого билборда. В этом случае Unreal Engine 4 позволяет продемонстрировать билборд в интерактивном виртуальном окружении заказчику. Но для размещения макетов необходимо перенастраивать текстуру объекта билборда и перекомпилировать всю сцену. Если заказчик захочет внести небольшую правку, необходимо выполнить несколько времязатратных действий: сообщить дизайнеру об изменениях, отправить макеты заказчику, настроить и перекомпилировать сцену.

Один из вариантов решения данной проблемы – использование системы отображения удаленного рабочего стола, где дизайнер транслирует свой рабочий стол в виртуальное окружение заказчика в реальном времени. В таком случае заказчик может рассматривать разнообразные макеты в нужном окружении, а также в режиме прямого диалога просить дизайнера вносить необходимые правки, которые сразу же будут становиться видны.

По результатам выполненного анализа доступных программных средств был сделан вывод, что наиболее подходящим протоколом передачи удаленного рабочего стола для эффективного решения данной задачи является протокол RFB (Remote Framebuffer Protocol) [2] в составе технологии VNC (Virtual Network Computing) [3], ввиду его кроссплатформенности, простоты реализации клиентской части и хорошей документированности. Однако нет готовых реализаций клиентских систем RFB, адаптированных под использование в виртуальном окружении.

### **Проектирование**

Протокол RFB подразумевает обновление экрана клиента посредством запроса на сервер: клиент запрашивает обновление экрана, после этого получает набор изображений в виде закодированных (сжатых) прямоугольников, которые затем отображает. Клиент может сконфигурировать текущую сессию, отправив предпочтительные кодировки, в которых будут кодироваться изображения, и цветовой формат. Помимо кодировок есть расширяющие протокол псевдокодировки. Прямоугольники не могут быть закодированы ими, но их можно указать для гибкой настройки других кодировок. Также протокол включает различные типы защиты, одним из которых является VNC Authentication (аутентификации клиента по заданному на сервере паролю).

В расширенной документации протокола RFB [4] представлено 11 различных кодировок для передаваемого изображения:

1. Raw – кодировка по умолчанию. Данные представлены в несжатом исходном виде: набор пикселей в порядке сверху-вниз, слева-направо.

2. CopyRect – предоставляет координаты прямоугольника в кадровом буфере клиента, который необходимо скопировать.

3. RRE – представляет исходный прямоугольник в виде совокупности подпрямоугольников, каждый из которых залит одним цветом.

4. CoRRE – вариация кодировки RRE, где характеристики подпрямоугольников (координаты, ширина, высота) ограничены 255 пикселями.

5. Nextile – вариация кодировки RRE, где все подпрямоугольники (тайлы) имеют размер 16 пикселей в ширину и в высоту (возможны меньшие размеры крайних в ряду, в колонке). Тайлы также могут включать в себя подпрямоугольники.

6. zlib – сжатие исходных данных при помощи библиотеки zlib.

7. Tight – включает в себя несколько видов сжатия: заполнение прямоугольника одним переданным цветом, сжатие в формате JPEG и сжатие при помощи zlib с предварительной обработкой изображения.

8. Zlibhex – вариация кодировок Nextile и zlib, где данные тайлов сжимаются при помощи zlib.

9. ZRLE – комбинирует подход кодировки RRE, разбиение на тайлы, использование палитры и сжатие при помощи zlib.

10. JPEG – отправляет данные сжатыми в формате JPEG.

11. Open H.264 – передает данные в формате H.264.

12. Tight PNG – аналогична кодировке Tight, за исключением замены JPEG на PNG.

Выделены функциональные требования к системе клиентской части VNC: генерация текстуры на основе получаемых от сервера данных; поддержка кодировок Raw, CopyRect, Tight; поддержка псевдо-кодировок, выполняющих функцию настройки кодировки Tight: JPEG Quality Level Pseudo-encoding (для указания качества сжатия JPEG) и Compression

Level Pseudo-encoding (для указания силы сжатия zlib); возможность настройки цветового формата и поддерживаемых кодировок; поддержка VNC Authentication.

Выделены нефункциональные требования: производительность – передача изображения в реальном времени; отказоустойчивость – корректная обработка разрыва соединения на различных этапах работы программы.

На основе функциональных требований были выделены акторы и прецеденты системы (диаграмма, построенная средствами языка UML, приведена на рис. 1), а также построена контекстная диаграмма IDEF0 (см. рис. 2).

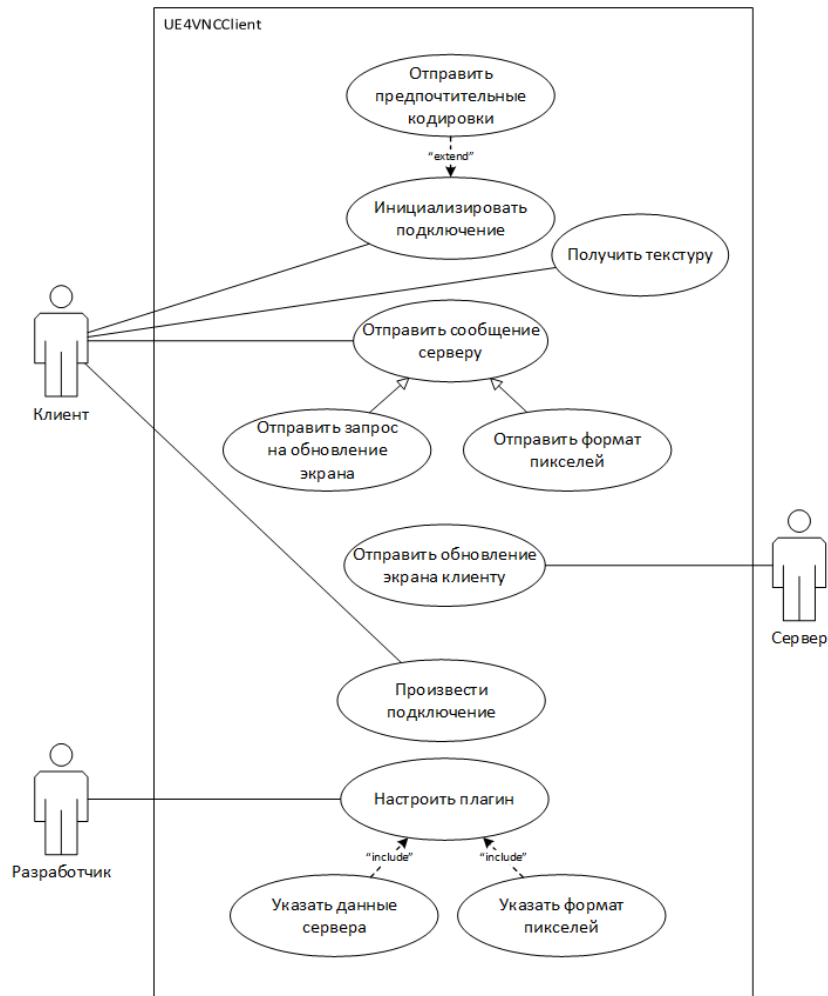


Рис. 1. Диаграмма прецедентов разрабатываемой системы

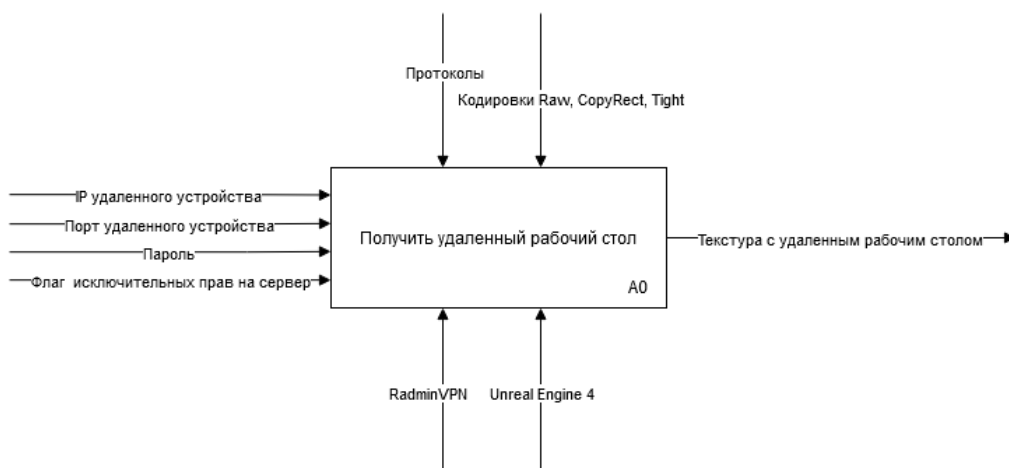


Рис. 2. Контекстная диаграмма разрабатываемой системы

Выделены классы системы VNC клиента: классы сообщений протокола RFB, классы кодировок (см. рис. 3).

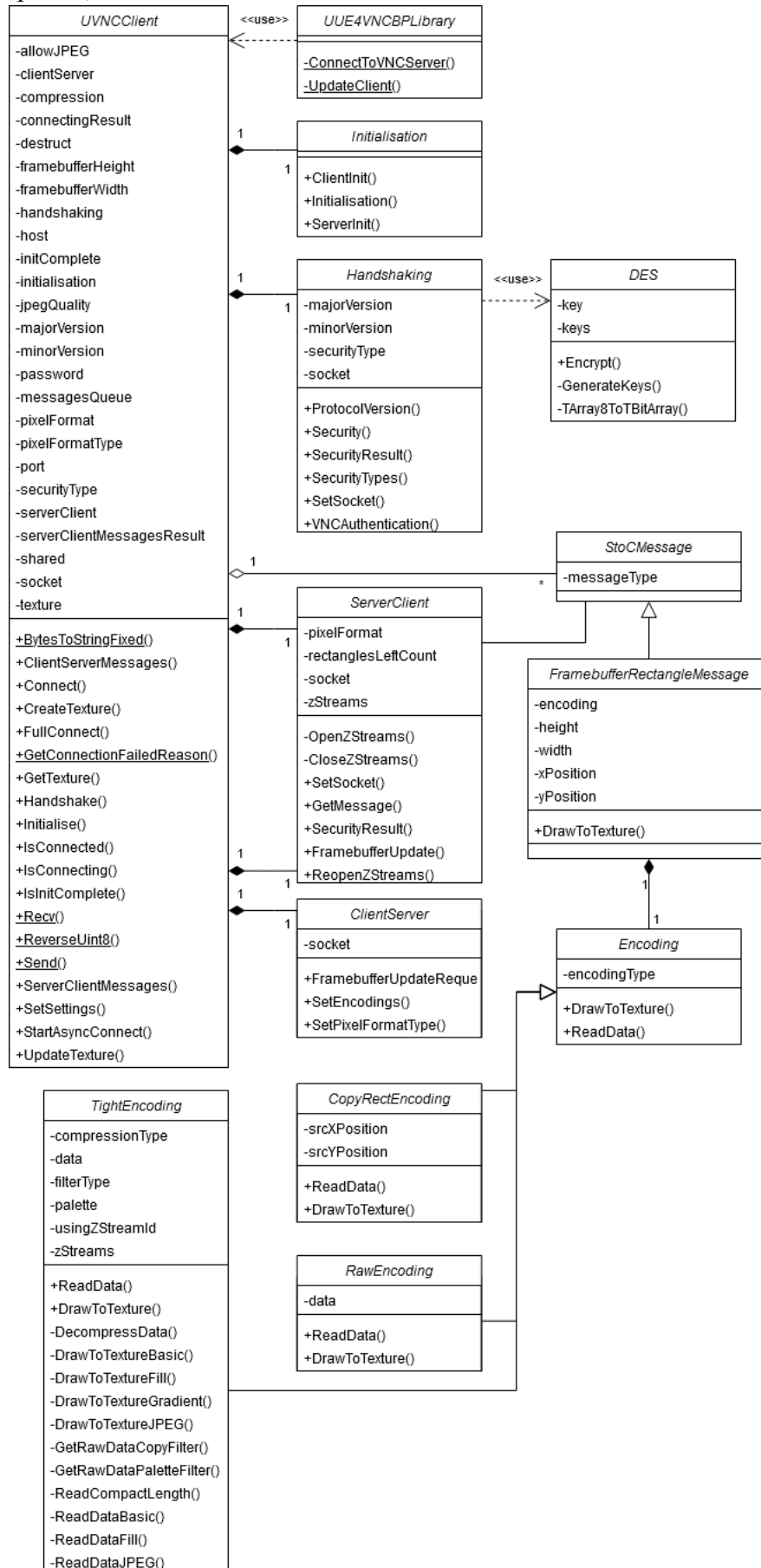


Рис. 3. Диаграмма классов разрабатываемой системы

## Реализация

На основе спроектированных классов реализованы компоненты системы VNC клиента (см. рис. 4).

Каждый вид сообщения протокола был представлен в виде отдельного компонента системы: Handshaking (обмен сообщениями для выполнения соглашений о типе защиты подключения), Initialization (обмен начальными данными о сервере и клиенте, такими как размер кадрового буфера и цветовой формат сервера), ServerClient (передача сообщений от сервера клиенту), ClientServer (передача сообщений от клиента серверу). Аналогично, каждая реализованная кодировка была представлена в виде отдельных компонентов. В целях поддержки VNC Authentication реализован компонент DES шифрования [5].

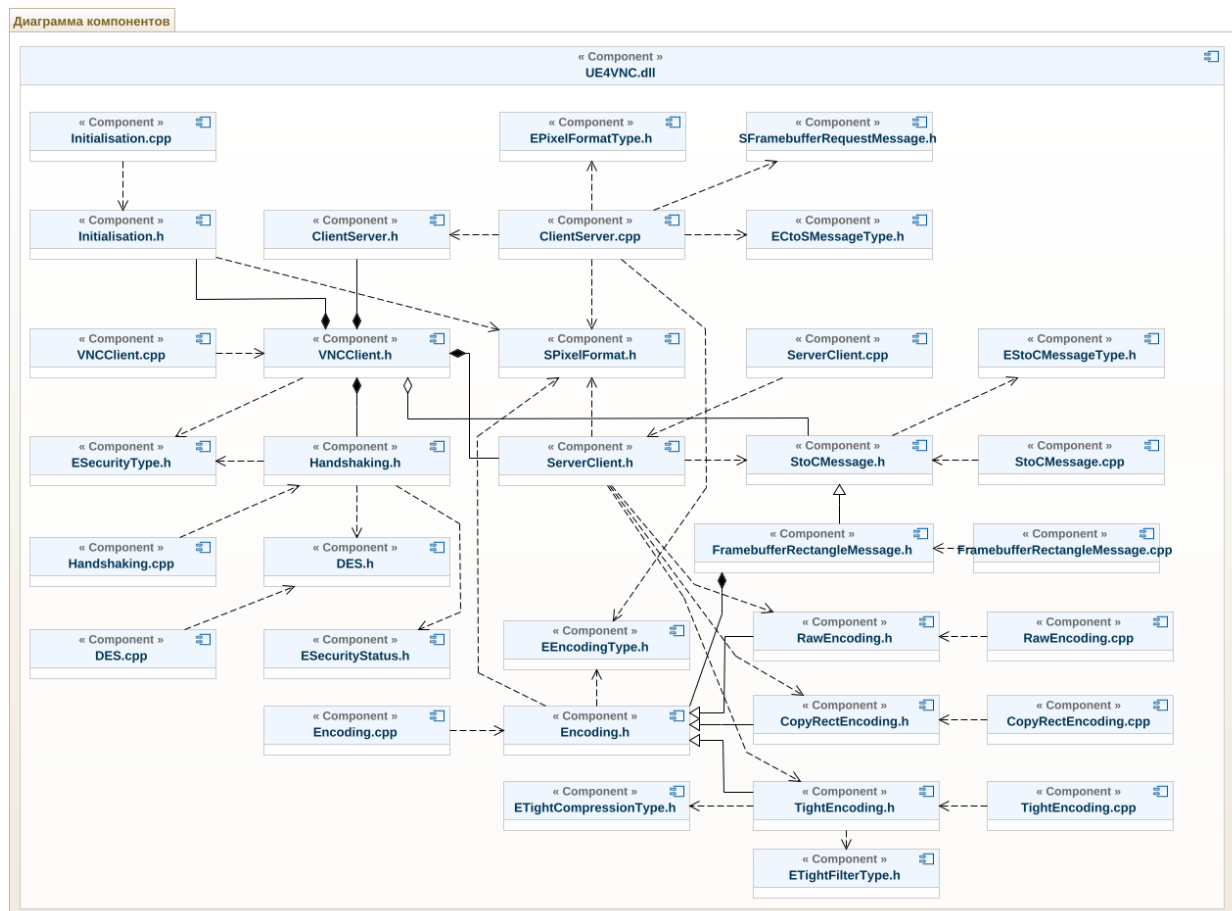


Рис. 4. Диаграмма компонентов разработанной системы

Система реализована с использованием языка программирования C++ [6] и встроенных в Unreal Engine 4 библиотек, используемых для генерации текстур, декодирования JPEG и zlib, а также организации асинхронности. Она реализована по лицензии MIT и расположена в открытом доступе на сервисе GitHub (<https://github.com/kyrillWhite/UE4VNCCClient>).

Итоговый плагин представляет собой библиотеку Blueprint-функций, состоящую из двух функций: VNC Connect – функция подключения клиента к серверу, VNC Update Client – функция запроса обновления кадрового буфера и получения графических данных от сервера. Результатом работы плагина является текстура, которую можно назначить произвольному объекту на 3D-сцене (например, билборду, экрану ситивизора в виртуальном городе и т.п.). При этом вызов функции VNC Update Client необходимо выполнять с определенной частотой для получения новых данных от сервера.

В системе визуального программирования Blueprint была реализована программа отображения виртуального рабочего стола на объекте сцены Unreal Engine 4 с использованием функций реализованного плагина (см. рис. 5–8).

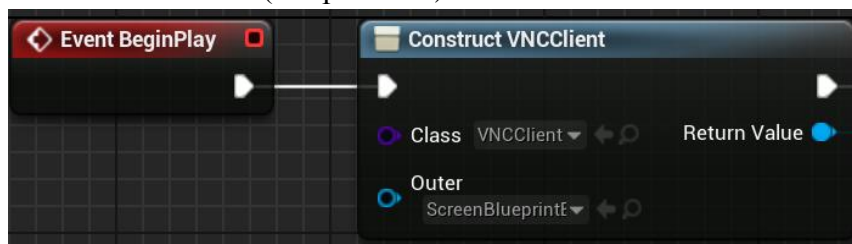


Рис. 5. Blueprint-код начала выполнения скрипта запуска клиента разработанной системы

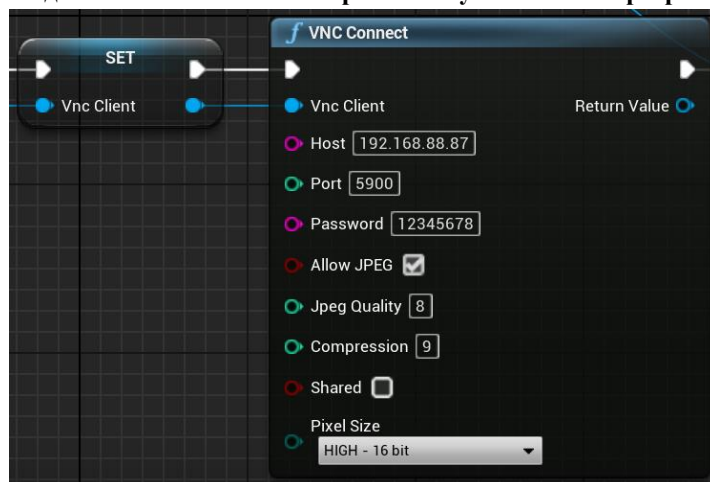


Рис. 6. Blueprint-код инициализации переменной и подключения к VNC серверу

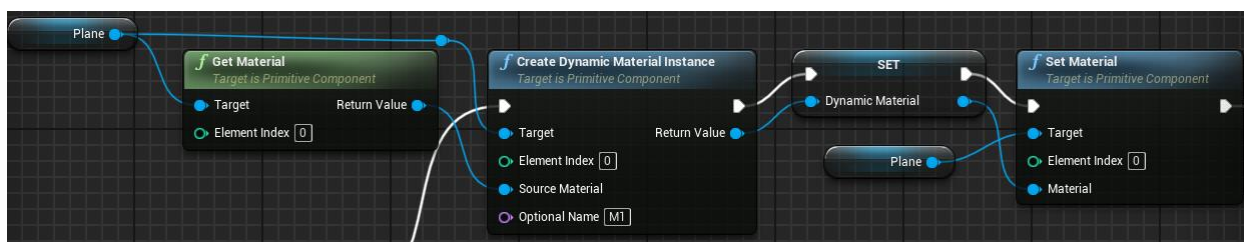


Рис. 7. Blueprint-код создания и установки динамического материала

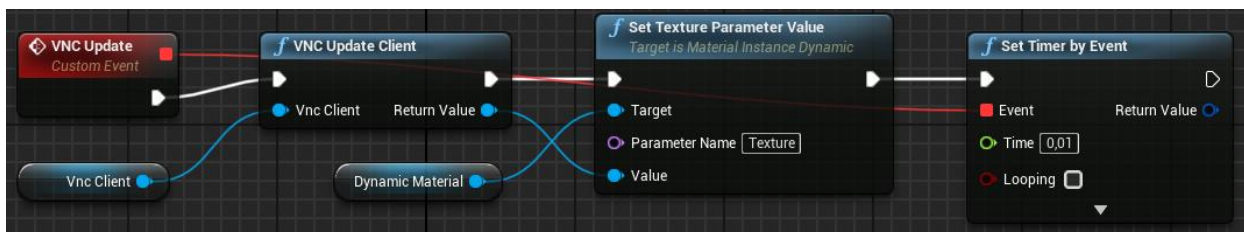


Рис. 8. Blueprint-код цикла обновлений текстуры

## Тестирование

Был определен набор из 37 тестов, на основе которого было выполнено тестирование работоспособности реализованной системы. Все данные, получаемые системой от сервера, представлены в корректном виде. Система демонстрирует отказоустойчивость при разрывах соединения на различных этапах работы.

Проведено сравнение VNC серверов, распространяющихся в открытом доступе, по качеству передаваемого потока данных при использовании с реализованным VNC клиентом. Рассматриваемые VNC серверы: TigerVNC [7], TightVNC [8], UltraVNC [9]. Для сравнения



было подготовлено две видеозаписи (см. рис. 9) с различной динамичностью: запись гонок «Формулы-1» (высокая динамичность) и запись рабочего стола с перемещением окна приложения (низкая динамичность).

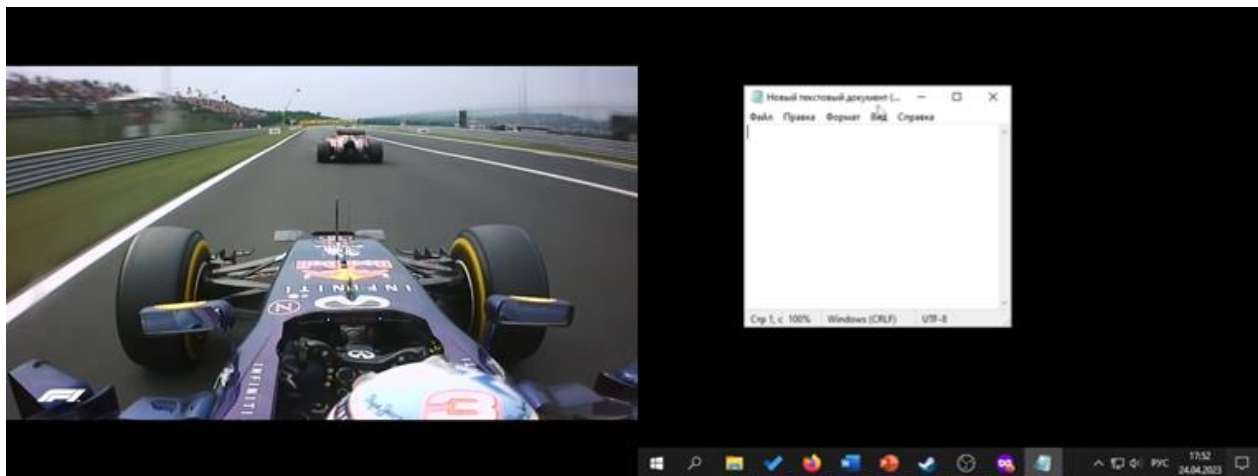


Рис. 9. Кадры тестовых видеозаписей

Для целей тестирования и отладки, система клиентской части VNC была расширена функцией сохранения получаемых данных в виде последовательности изображений.

Для каждого рассматриваемого VNC сервера через клиент были воспроизведены обе видеозаписи и сохранены в виде последовательностей PNG файлов. После сохранения переданных клиенту видеопотоков, была вручную проведена их синхронизация и выравнивание (сопоставление и удаление лишних кадров, возникших из-за ручного запуска и остановки передачи). В результате для каждого вида сервера было получено 268 кадров видеозаписи гонки и 224 кадра видеозаписи рабочего стола.

В качестве метрики сравнения двух изображений было выбрано среднеквадратичное отклонение компонента цвета пикселя.

С использованием языка программирования Python и библиотеки OpenCV [10] был реализован скрипт, выполняющий расчет метрики между двумя последовательностями изображений. Также при помощи скрипта были сгенерированы изображения (см. рис. 10–12), отображающие среднюю ошибку, полученную на протяжении всей последовательности изображений (слева – ошибка передачи видеозаписи гонки, справа – ошибка передачи видеозаписи рабочего стола).

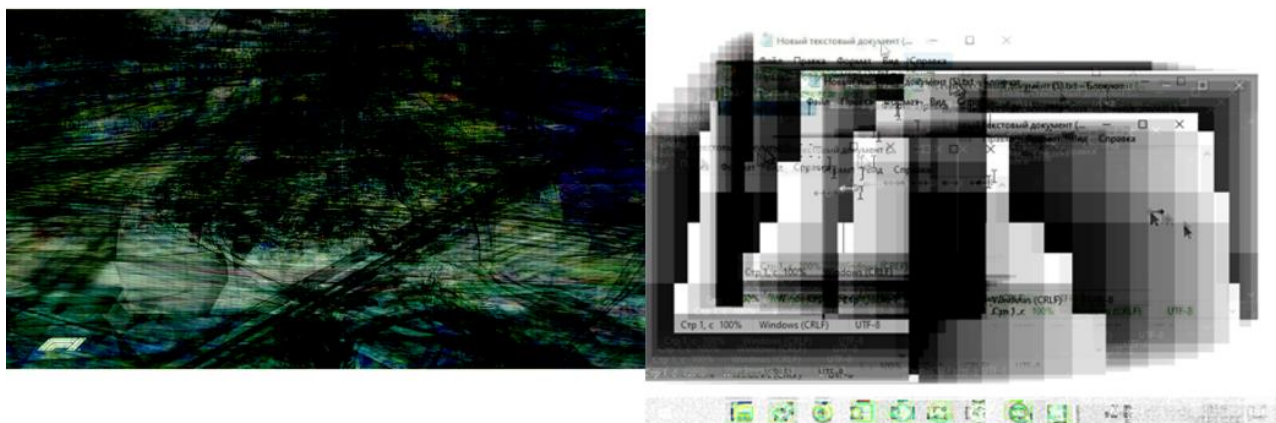


Рис. 10. Средние ошибки передачи изображения в TigerVNC

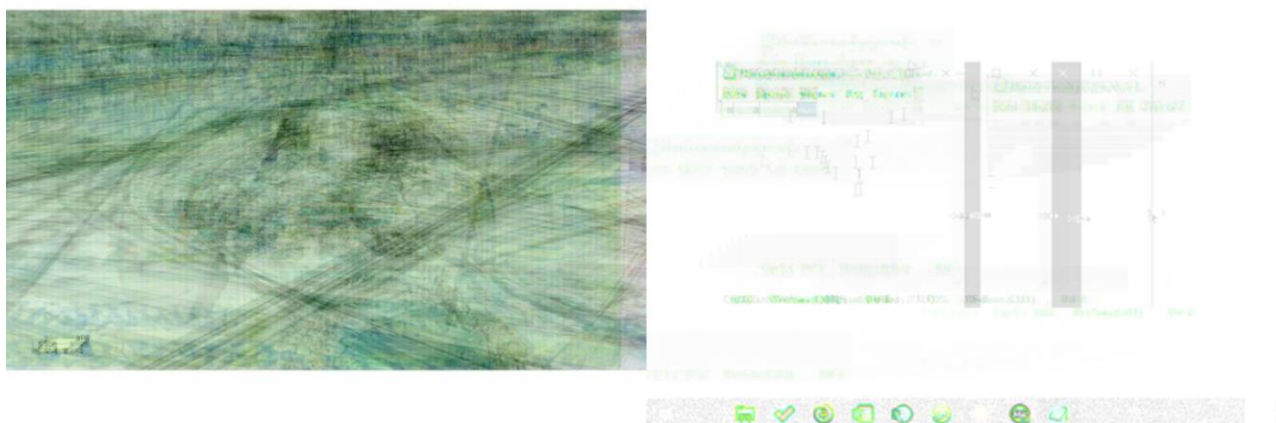


Рис. 11. Средние ошибки передачи изображения в TigthVNC

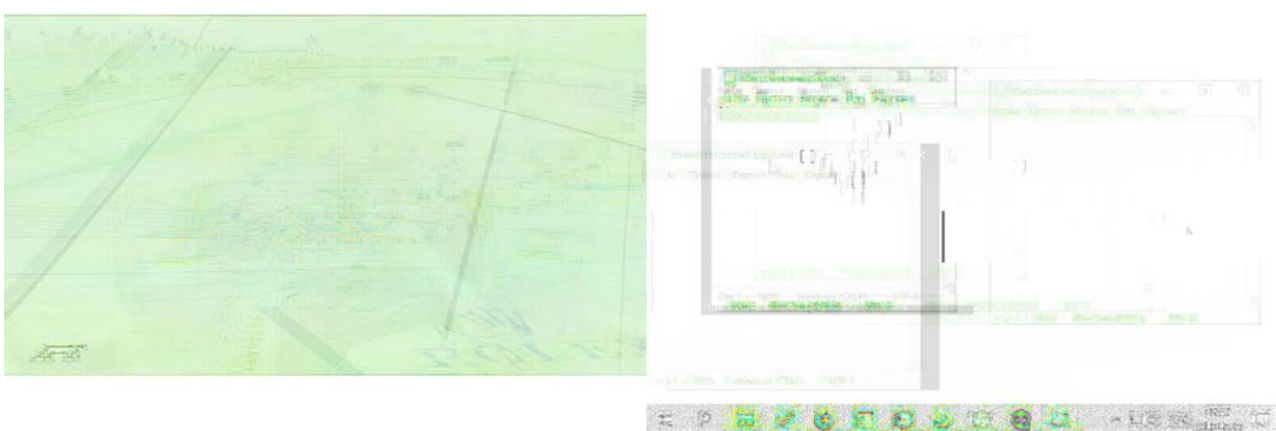


Рис. 12. Средние ошибки передачи изображения в UltraVNC

На основе полученных данных построена диаграмма сравнения среднеквадратичных отклонений последовательностей изображений, передаваемых рассматриваемыми серверами, в сравнении с исходными последовательностями (см. рис. 13).

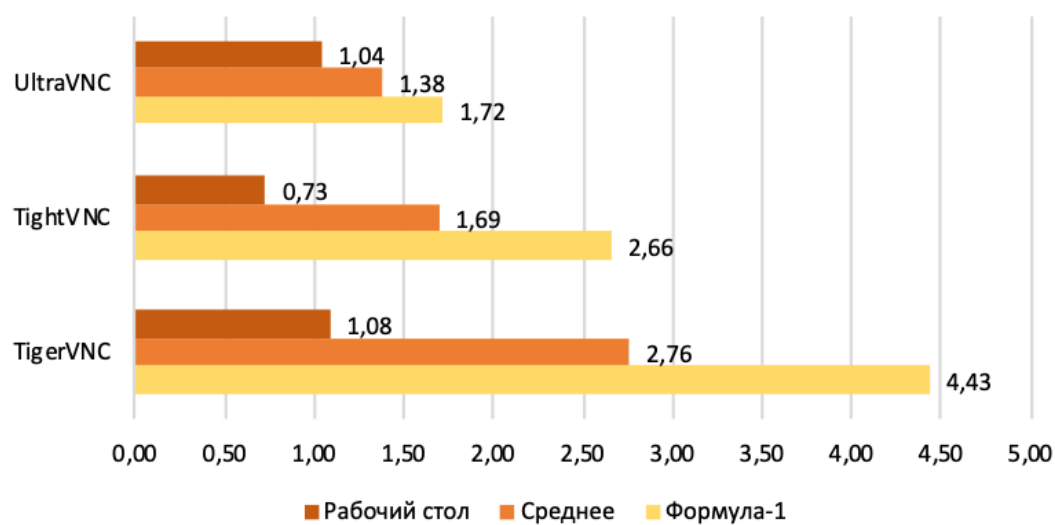


Рис. 13. Диаграмма сравнения среднеквадратичных отклонений полученных последовательностей изображений от исходных (меньше – лучше)

На диаграмме видно, что реализованный клиент с использованием сервера TigerVNC наиболее корректно (0,7282) отображает передаваемый видеопоток рабочего стола. Для случая динамичного изображения лучший результат (1,7179) показывает работа клиента с сервером UltraVNC. Наилучшим в среднем оказался UltraVNC (1,38). Работа с TigerVNC показала наихудший результат в обоих тестах.

### **Заключение**

По результатам работы была спроектирована, реализована и протестирована система клиентской части VNC в виде библиотеки Blueprint-функций для игрового движка Unreal Engine 4, позволяющая отображать удаленный рабочий стол в виртуальной реальности. Также были реализованы примеры использования разработанного VNC клиента на визуальном языке программирования Blueprint в среде движка Unreal Engine 4.

Кроме того, было выполнено сравнение наиболее популярных VNC серверов по качеству видеопотока, передаваемого в реализованный VNC клиент, с использованием двух видеопотоков разной динамичности. В среднем, лучшим сервером оказался UltraVNC. Лучшим для передачи рабочего стола оказался TigerVNC. Худшим в обоих случаях оказался TigerVNC.

Реализованная система будет использована в исследованиях рекламных баннеров с применением технологий виртуальной реальности (на базе Unreal Engine 4) и отслеживания взгляда, где основной целью является поиск оптимального решения для создания рекламы. Используемые технологии позволят собрать данные о том, на какие зоны баннера обращено больше всего внимания и какое окружение больше всего подходит для его привлечения. Для нахождения оптимального дизайна баннера необходимо постоянное внесение правок. Система поможет отображать рекламные баннеры динамически, с возможностью их изменения в реальном времени, что упростит взаимодействие с Unreal Engine 4 (пропадет необходимость перезапускать виртуальную сцену при обновлении баннера), позволит выполнять правки дизайна баннера в привычном окружении (в любом графическом редакторе), а также ускорит процесс проведения экспериментов.

В дальнейшем планируется:

- реализация кодировки H.264, которую поддерживает TigerVNC;
- добавление функциональности управления удаленным рабочим столом;
- интеграция с инструментом Creative Maps Studio [11];
- расширение плагина для передачи данных векторной графики.

### **Библиографический список**

1. *Chen X., Wang M., Wu Q.* Research and development of virtual reality game based on unreal engine 4 // 2017 4th International Conference on Systems and Informatics (ICSAI). Hangzhou, China, 2017. P. 1388-1393.
2. The Remote Framebuffer Protocol [Электронный ресурс]. – UPL: <https://www.rfc-editor.org/rfc/rfc6143/> (дата обращения 23.04.2023).
3. *Richardson T., Stafford-Fraser Q., Wood K. R., Hopper A.* Virtual network computing // IEEE Internet Computing. 1998. vol. 2, no. 1. P. 33-38.
4. The RFB Protocol [Электронный ресурс]. – UPL: <https://github.com/rfbproto/rfbproto/blob/master/rfbproto.rst> (дата обращения 23.04.2023).

5. Алфёров А.П., Zubov A.Yu., Кузьмин А.С., Черемушкин А.В. Основы криптографии – 3-е изд. – Москва : Гелиос АРВ, 2002. – 480 с.
6. Страуструп Б. Язык программирования C++ // Пер. с англ. – 3-е изд. – СПб.; М.: Невский диалект – Бином, 1999. – 991 с.
7. TigerVNC [Электронный ресурс]. – UPL: <https://tigervnc.org/> (дата обращения 23.04.2023).
8. TightVNC: VNC-Compatible Free Remote Desktop Software [Электронный ресурс]. – UPL: <https://www.tightvnc.com/> (дата обращения 23.04.2023).
9. UltraVNC: Remote Desktop Free Opensource [Электронный ресурс]. – UPL: <https://uvnc.com/> (дата обращения 23.04.2023).
10. Culjak I., Abram D., Pribanic T., Džapo H. and Cifrek M. A brief introduction to OpenCV in 2012 Proceedings of the 35th International Convention MIPRO. Opatija, Croatia, 2012. P. 1725-1730.
11. Chumakov R.V., Ryabinin K.V., Belousov K.I., Duan J. Creative Map Studio: A Platform for Visual Analytics of Mental Maps in Scientific Visualization. 2021. vol. 13, P. 79-93.

### References

1. X. Chen, M. Wang and Q. Wu, “Research and development of virtual reality game based on unreal engine 4,” in *2017 4th International Conference on Systems and Informatics (ICSAI)*, Hangzhou, China, 2017, pp. 1388-1393.
2. The Remote Framebuffer Protocol [Electronic resource]. – UPL: <https://www.rfc-editor.org/rfc/rfc6143/> (accessed 23 April 2023).
3. Richardson T., Stafford-Fraser Q., Wood K.R., Hopper A., “Virtual network computing,” in *IEEE Internet Computing*, 1998, vol. 2, no. 1. pp. 33-38.
4. The RFB Protocol [Electronic resource]. – UPL: <https://github.com/rfbproto/rfbproto/blob/master/rfbproto.rst> (accessed 23 April 2023).
5. Alferov A.P., Zubov A.Yu., Kuzmin A.S., Cheremushkin A.V., *Osnovy kriptografii [Basics of Cryptography]*, 3rd ed. Moscow: Gelios ARV Publ., 2002. 480 p. (in Russian).
6. Stroustrup B., *The C++ Programming Language*, 3rd ed. Boston: Addison-Wesley Professional, 2000.
7. TigerVNC [Electronic resource]. – UPL: <https://tigervnc.org/> (accessed 23 April 2023).
8. TightVNC: VNC-Compatible Free Remote Desktop Software [Electronic resource]. – UPL: <https://www.tightvnc.com/> (accessed 23 April 2023).
9. UltraVNC: Remote Desktop Free Opensource [Electronic resource]. – UPL: <https://uvnc.com/> (accessed 23 April 2023).
10. Culjak I., Abram D., Pribanic T., Džapo H. and Cifrek M. “A brief introduction to OpenCV,” in *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, Croatia, 2012, pp. 1725-1730.
11. Chumakov R.V., Ryabinin K.V., Belousov K.I., Duan J. “Creative Map Studio: A Platform for Visual Analytics of Mental Maps,” in *Scientific Visualization*, 2021, vol. 13, no. 2, pp. 79-93, DOI: 10.26583/sv.13.2.06.

## **DEVELOPMENT OF REMOTE DESKTOP DISPLAY MECHANISMS FOR VIRTUAL REALITY**

*Belyakov Kirill V.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, kiribel2000@gmail.com

This paper describes the development of a system that implements the client part of the VNC (Virtual Network Computing) protocol as a plug-in for the Unreal Engine 4 graphics engine. The system provides capabilities for displaying the remote desktop in virtual reality. There are flexible settings of transmitted image quality: JPEG compression quality, zlib compression level, color depth selection. Also, the paper describes the use of a plug-in to display a remote desktop on the object-plane of the Unreal Engine 4 environment. The most popular VNC servers were compared in terms of video stream quality, where UltraVNC showed the best result on average, and TigerVNC the worst. Unlike other VNC clients, this system is integrated with the virtual reality environment, allowing the remote desktop to be embedded in a customized virtual environment. The implemented system makes it easier to collect the data needed to research advertising banners in order to find the optimal solution for the advertisement creation. By using eye-tracking and virtual reality technologies it is possible to collect data on which areas of the banner attract the most attention and which environment is most suitable for its attraction. Constant adjustments are needed to find the best banner design. The system will help display advertising banners dynamically, with the ability to change them in real time on the server side, which speeds up the experimentation process and allows you to make design edits in the familiar environment for the designer. Future plans include implementing H.264 encoding, adding remote desktop management functionality and integration with the Creative Maps Studio tool.

**Keywords:** virtual reality, remote desktop, Unreal Engine 4, Virtual Network Computing, Remote Framebuffer Protocol.

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ЭКОЛОГИЧНОЙ УТИЛИЗАЦИИ ОТХОДОВ**

*Бердников Кирилл Сергеевич, Ракина Валерия Денисовна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kirill.berdnikov27052002@gmail.com

В статье рассмотрены основные понятия в области экологической утилизации отходов и аналогичные информационные системы. Выявлены основные задачи и отличительные черты проектируемой системы. Произведен выбор средств проектирования и разработки информационной системы, системы управления базой данных, средств проектирования прототипа пользовательского интерфейса. Спроектированы модели информационной системы, модели базы данных, прототипа пользовательского интерфейса.

Ключевые слова: проектирование и документирование информационной системы, экологическая утилизация отходов.

### **Введение:**

В настоящее время проблема обращения с бытовыми отходами является одной из наиболее актуальных для городов и населения в целом. С каждым годом количество производимых неорганических отходов только увеличивается, но лишь малая часть из них идет на переработку. Остальная часть отходов остается лежать на свалках, постепенно загрязняя окружающую среду. Все это происходит из-за отсутствия должной сортировки отходов.

Целью данной статьи является проектирование информационной системы для экологической утилизации отходов, которая будет информировать граждан о пунктах приема тех или иных отходов, что позволит не только увеличить процент перерабатываемых отходов, но и снизит ущерб, причиняемый окружающей среде.

Предметом исследования является автоматизация процесса информирования граждан о пунктах приема тех или иных видов отходов. В качестве объекта исследования выступает информация о пунктах приема и видах отходов и их утилизации.

### **Анализ предметной области:**

Пункты приема неорганических отходов являются небольшими складами, куда граждане могут сдать вторсырье для последующей переработки. Многие пункты выплачивают денежные средства за принесенное вторсырье, но существуют те, которые принимают утиль на безвозмездной основе. В последствие пункт приема отправляет все полученное вторсырье на переработку.

На данный момент в Перми находятся 252 пункта, принимающие неорганические отходы для последующей переработки. Таким образом, благодаря развитой инфраструктуре этой отрасли, у граждан появляется возможность сдавать вторсырье на переработку без особых трудностей. В дальнейшем сотрудничество горожан и пунктов приема может привести к более эффективному использованию ресурсов и значительному снижению вреда для окружающей среды.

Неорганические отходы – это отходы, которые не могут разлагаться или распадаться на естественные компоненты, поскольку они состоят из неорганических материалов [1]. Основными видами неорганических отходов являются:

- пластиковые отходы [2];
- стеклянные отходы [3];
- текстильные отходы [4];
- металлические отходы [5];
- химические отходы [6];
- электронные отходы [7];
- резиновые отходы [8].

Экологическая утилизация неорганических отходов включает сбор, сортировку и переработку, чтобы извлечь ценные ресурсы, снизить загрязнение окружающей среды и минимизировать отрицательное воздействие на экосистемы. Некоторые методы экологической утилизации неорганических отходов включают:

- рециклинг (переработка) [9];
- регенерация [9];
- рекуперация [9];
- захоронение на специальных свалках [9];
- энергетическое использование [9];
- пиролиз [10];
- повторное использование.

В целом, эффективная утилизация неорганических отходов требует сочетания сбора, сортировки, переработки, утилизации, а также сознательности со стороны общества. Повышение осведомленности о проблеме отходов и развитие современных методов утилизации помогут минимизировать воздействие неорганических отходов на окружающую среду.

Анализ аналогичных информационных систем:

Были рассмотрены преимущества и недостатки аналогичных информационных систем на примере RECYCLEMAP [11] и OPENRECYCLEMAP [12].

Веб-сайт RECYCLEMAP представляет собой карту России с отмеченными на ней пунктами приема разных типов отходов. На сайте доступна фильтрация отображаемых пунктов приема на основе местоположения, принимаемых отходов, графика работы и доступности для граждан. О каждом пункте приема доступна информация, такая как точный адрес, название, время работы, тип принимаемых отходов, контактный телефон, отзывы пользователей, а также дата последнего обновления информации. Пользователи могут зарегистрироваться, чтобы ставить оценки пунктам приема, писать отзывы и добавлять новые, если таковых нет на карте. Сайт имеет минималистичный и приятный интерфейс, в котором не составит труда разобраться каждому пользователю.

Из недостатков были выделены отсутствие пунктов приема вещей на благотворительность, так как многие ненужные вещи могут быть в хорошем состоянии, что очень поможет нуждающимся. Также нигде нет указания, что тот или иной пункт приема выплачивает денежные средства за принесенные отходы, что может быть важным критерием для некоторых людей.

Следующий веб-сайт OPENRECYCLEMAP так же, как и предыдущая информационная система представляет собой карту с отмеченными на ней пунктами приема разных типов отходов, но уже не только России, а всего мира. Фильтрация пунктов приема типу принимаемых отходов более обширная, в сравнении с предыдущим сайтом. Интерфейс приятный, минималистичный и интуитивно понятный, что позволяет без проблем разобраться с функционалом сайта.

К недостаткам этой информационной системы относятся все вышеперечисленные недостатки предыдущего сайта, отсутствие какой-либо информации о пунктах приема, отсутствие фильтрации по местоположению (необходимо искать нужный город и адрес вручную, если нет желания делиться своим местоположением) и графику работы. Также стоит отметить малое количество пунктов приема на карте.

Определение основных задач и особенностей:

У каждой рассмотренной аналогичной информационной системы есть свои преимущества и недостатки, которые были учтены при моделировании и проектировании информационной системы.

На основе преимуществ и недостатков проанализированных информационных систем, были выделены основные требования для разрабатываемой системы:

- интеграция с картографическими сервисами для отображения пунктов приема на карте с удобным навигацией;
- предоставление списка и местоположения пунктов приема отходов для каждого типа отходов;
- отображение информации о точном адресе, названии, графике работы, контактных данных, отзывах пользователей, наличии денежных вознаграждений, возможности вывоза и типах принимаемых видов отходов для каждого пункта приема;
- хранение данных о пунктах приема в базе данных;
- возможность регистрации и авторизации в системе;
- возможность фильтрации пунктов приема по местоположению, типу принимаемых отходов, графику работы, возможности вывоза и выплате денежных средств;
- предоставление раздела с информацией о различных типах отходов и правильных способах их утилизации;
- предоставление удобного и интуитивно понятного пользовательского интерфейса;
- возможность исправления информации о пунктах приема, добавления новых пунктов, оценивания и написания отзывов.

Вся добавленная или измененная информация перед публикацией будет проверяться и корректироваться модератором. Также отзывы пользователей будут модерироваться во избежание публикации спам сообщений, ненормативной лексики и других неправомерных сообщений.

Выбор средства моделирования:

Для моделирования информационной системы было выбрано CASE-средство StarUML [13]. Основными критериями выбора являлись наличие бесплатной версии и локализованной документации.

Выбор средств разработки:

Исходя из того, что реализуемая информационная система будет представлена в виде клиент-серверной модели web-приложения, были выбраны средства для разработки клиентской и серверной части системы.

Для разработки клиентской части были выбраны языки программирования HTML [14], CSS [15] и JavaScript [16].

Для создания серверной части информационной системы был выбран язык программирования Python [17]. Основными критериями выбора Python, являлись наличие большого количества различных API (Application Programming Interface) и библиотек, позволяющих интегрировать картографические сервисы в web-приложение, и высокая масштабируемость языка. Также немаловажным фактором стало наличие большого количества локализованной литературы, руководств и технической документации, что значительно поможет при разработке.



Выбор системы управления базой данных:

При выборе системы управления базой данных (СУБД), было учтено то, что система будет являться web-приложением. Исходя из этого выбор сделан в пользу PostgreSQL [18], так как эта СУБД обладает архитектурой «клиент-сервер», имеет реляционную модель данных, высокую масштабируемость и надежность. Немаловажными факторами выбора также являлись наличие бесплатной лицензии и большого количества локализованной документации.

Выбор средства для проектирования пользовательского интерфейса:

Для проектирования интерфейса было выбрано средство для разработки web-дизайна Figma [19], так как оно имеет множество собственных библиотек, возможность использования сторонних расширений и собственных материалов, поддерживает импорт и экспорт в HTML, CSS, JavaScript, а также обладает множеством локализованной документации и обучающих материалов.

Проектирование модели информационной системы:

В начале проектирования, были определены виды пользователей и доступные для них функциональные возможности. В системе будет два вида пользователей: обычные пользователи и администраторы. Функциональные возможности для каждого вида пользователей представлены с помощью диаграмм прецедентов на рис. 1 и рис. 2 соответственно.



Рис. 1. USE-CASE диаграмма пользователя



Рис. 2. USE-CASE диаграмма администратора

Далее был подробно рассмотрен процесс добавления нового пункта приема пользователем. Соответствующая диаграмма деятельности представлена на рис. 3.

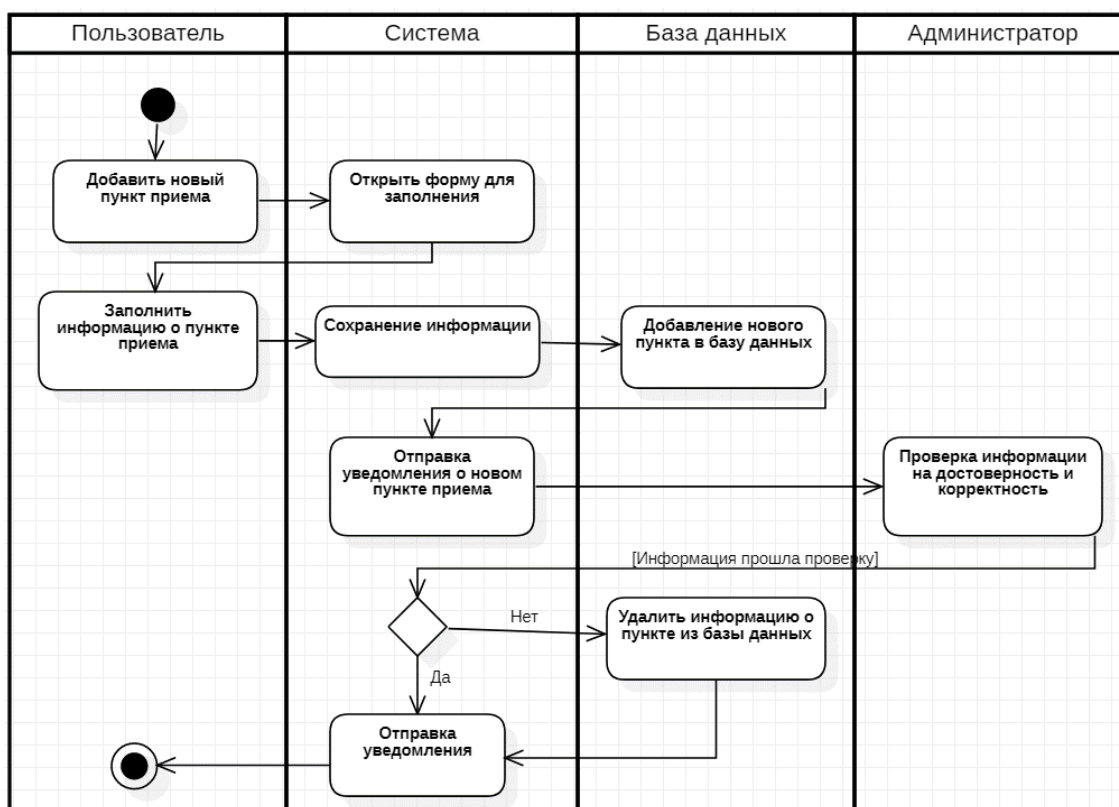


Рис. 3. Диаграмма деятельности для добавления нового пункта приема пользователем

На следующем шаге была описана разрабатываемая система. Для этого была использована диаграмма классов, благодаря которой была визуализирована структура системы и взаимосвязи между ее компонентами. Диаграмма классов представлена на рис. 4.

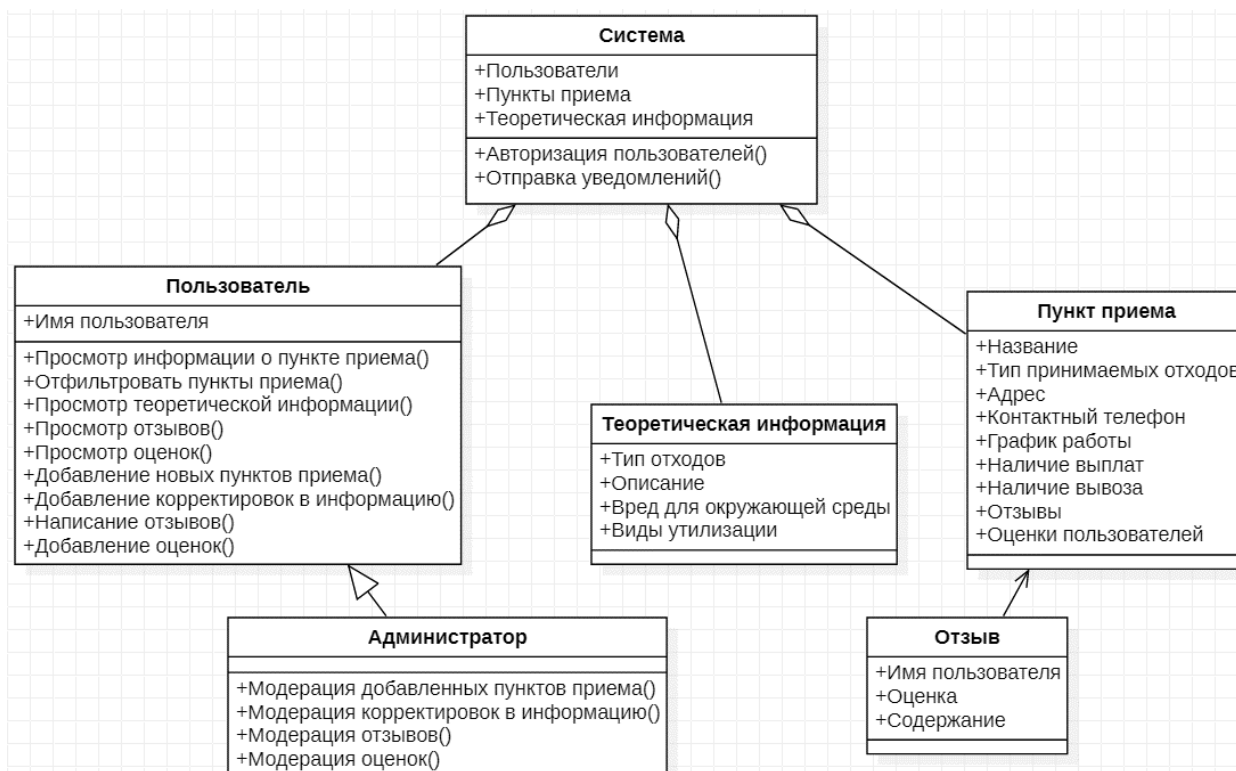


Рис. 4. Диаграмма классов

Проектирование модели базы данных:

Для проектирования базы данных, были определены данные, которые будут храниться, а также их атрибуты и взаимосвязи.

В базе данных будут храниться данные об учетных записях пользователей, информация о пунктах приема, информация о новых пунктах приема, информация о корректировках пользователей и их отзывах.

Также необходимо определить типы связей между сущностями. Каждый пользователь может написать множество отзывов для множества пунктов, поэтому отношение между «Пользователем» и «Отзывом» являются отношением один ко многим.

Отношение между «Пунктом приема» и «Отзывом» является отношением один ко многим, так как для одного пункта приема может быть написано несколько отзывов.

Отношение сущностей «Пользователь» и «Пункт приема» к сущности «Корректировка информации» является отношением один ко многим, так как один пользователь может написать несколько корректировок, а к информации об одном пункте приема также могут быть написаны несколько корректировок.

Отношение между «Пользователем» и «Новым пунктом приема» также является отношением один ко многим, так как пользователь может добавить несколько новых пунктов.

Далее создадим модель базы данных (см. рис. 5).

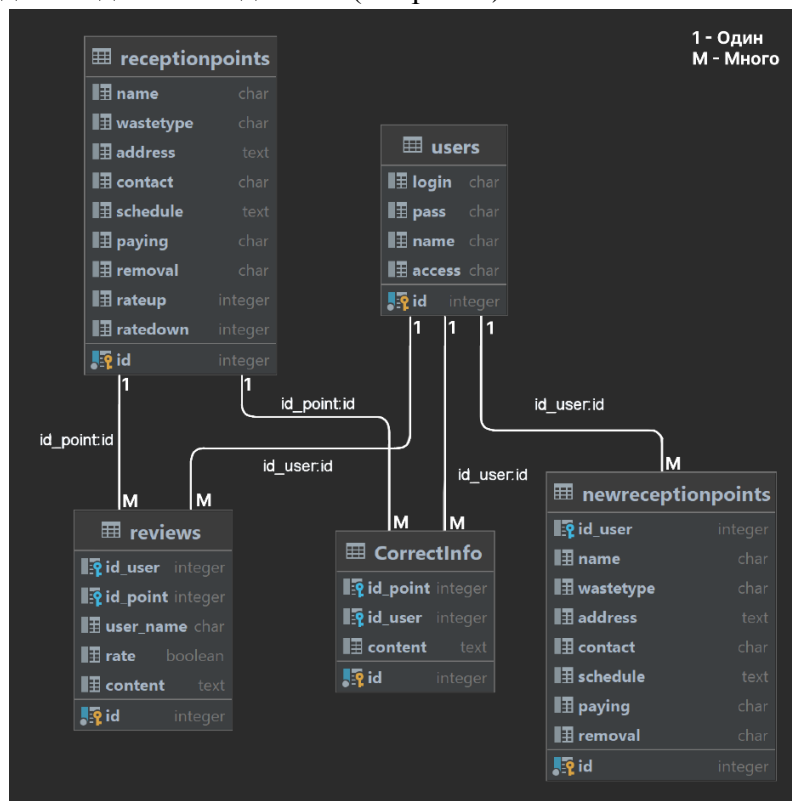


Рис. 5. Модель базы данных

Проектирование прототипа пользовательского интерфейса:

Основываясь на полученной структуре информационной системы и модели базы данных, был спроектирован прототип интерфейса для будущего web-приложения. Для начала был создан прототип главной страницы системы (рис. 6).

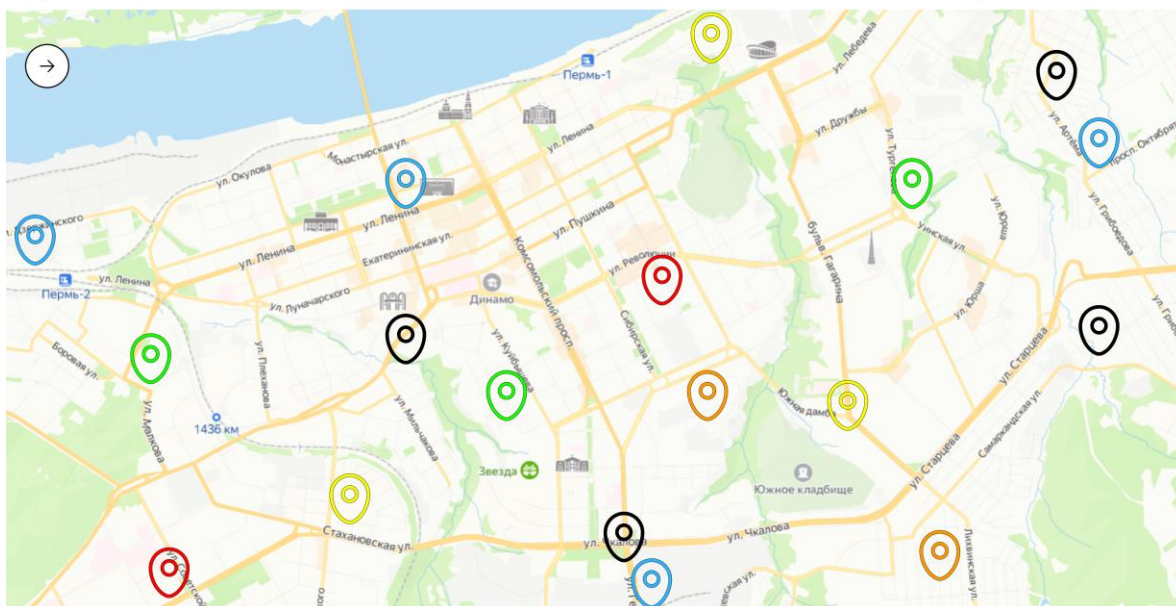


Рис. 6. Прототип интерфейса главной страницы

На главной странице размещена интерактивная карта с отмеченным на ней пунктами приема, кнопки авторизации, перехода к теоретической информации и вызова вкладки с фильтрацией и списком пунктов (рис. 7).

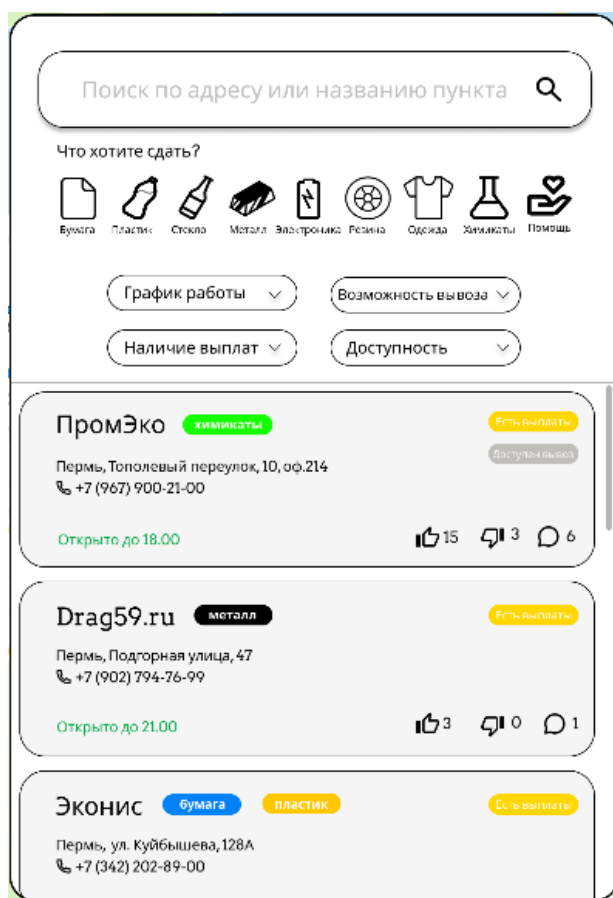
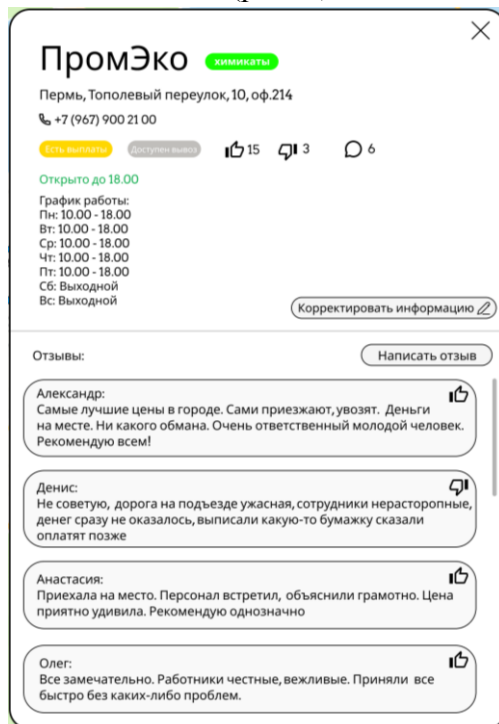


Рис. 7. Прототип интерфейса вкладки фильтрации

Из этой вкладки у пользователей есть возможность просмотреть полную информацию о пункте приема, а также отзывы людей о нем (рис. 8).



**ПромЭко** химикаты

Пермь, Тополевый переулок, 10, оф.214  
+7 (967) 900 21 00

Есть выплаты Доступен вывоз 15 3 6

Открыто до 18.00

График работы:  
Пн: 10.00 - 18.00  
Вт: 10.00 - 18.00  
Ср: 10.00 - 18.00  
Чт: 10.00 - 18.00  
Пт: 10.00 - 18.00  
Сб: Выходной  
Вс: Выходной

Корректировать информацию

Отзывы: Написать отзыв

Александр:  
Самые лучшие цены в городе. Сами приезжают, увозят. Деньги на месте. Ни какого обмана. Очень ответственный молодой человек. Рекомендую всем!

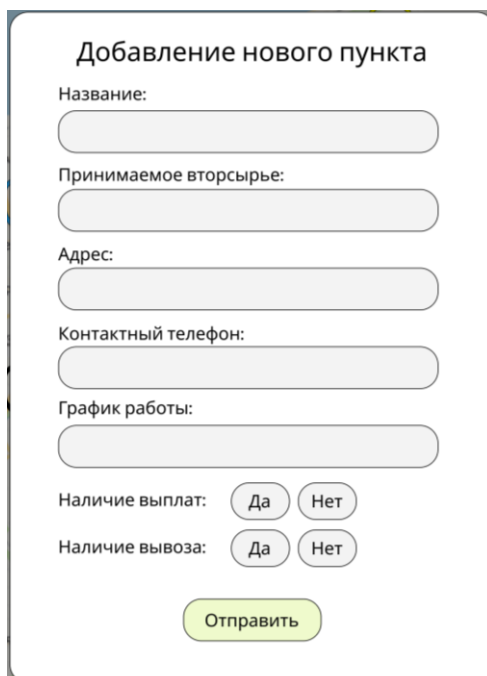
Денис:  
Не советую, дорога на подъезде ужасная, сотрудники нерасторопные, денег сразу не оказалось, выписали какую-то бумажку сказали оплатят позже

Анастасия:  
Приехала на место. Персонал встретил, объяснили грамотно. Цена приятно удивила. Рекомендую однозначно

Олег:  
Все замечательно. Работники честные, вежливые. Приняли все быстро без каких-либо проблем.

Рис. 8. Прототип интерфейса вкладки с информацией о пункте приема

Помимо этого, авторизованные пользователи могут добавлять новые пункты приема, если они еще не представлены на карте. Прототип интерфейса окна добавления нового пункта приема представлен на рис. 9.



**Добавление нового пункта**

Название:

Принимаемое вторсырье:

Адрес:

Контактный телефон:

График работы:

Наличие выплат: Да Нет

Наличие вывоза: Да Нет

Отправить

Рис. 9. Прототип интерфейса окна добавления нового пункта приема

Также, всем пользователям, вне зависимости от того, зарегистрированы они или нет, доступна возможность ознакомиться с теоретической информацией о каждом типе неорганических отходов, их опасности для окружающей среды и методах экологичной утилизации (см. рис. 10).



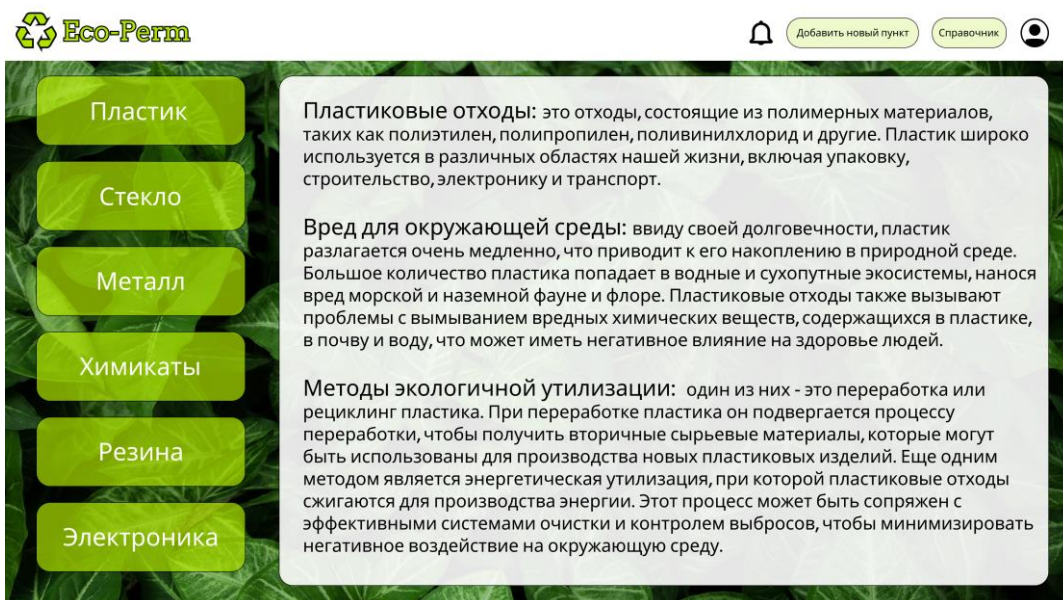


Рис. 10. Прототип интерфейса страницы с теоретической информацией о типах отходов

### Библиографический список

1. Неорганические отходы: характеристика, классификация, обработка // WARBLETONCOUNCIL [Электронный ресурс]. – URL: <https://ru1.warbletoncouncil.org/residuos-inorganicos-101> (дата обращения: 05.02.2023).
2. Утилизация пластмассовых отходов: основные экологические проблемы переработки пластика и понятие биоразлагаемого пластика // Greenologia [Электронный ресурс]. – URL: <https://greenologia.ru/othody/utilizatsiya-plastika.html#i-4> (дата обращения: 06.02.2023).
3. Весь процесс переработки стекла: утилизация как способ сохранить природу и заработать // RECYCLE.NET [Электронный ресурс]. – URL: <https://rcycle.net/steklo/pererabotka-utilizatsiya> (дата обращения: 06.02.2023).
4. Экологическая потребность и экономическая выгода переработки отходов швейного производства – куда сдать отходы, оборудование, процессы утилизации текстиля // Экспертный журнал о мусоре, отходах производства [Электронный ресурс]. – URL: <https://vseomusore.com/pererabotka-otkhodov/ekologicheskaya-potrebnost-i-ekonomicheskaya-vygoda-pererabotki-othodov-shvejnogo-proizvodstva-kuda-sdat-othody-oborudovanie-protsessy-utilizatsii-tekstilya/> (дата обращения: 06.02.2023).
5. Как неутилизированные металлические отходы вредят экологии // ПКВМ [Электронный ресурс]. – URL: <https://metallolom777.ru/blog/kak-neutilizirovannye-metallicheskie-otkhody-vredyat-ekologii/> (дата обращения: 06.02.2023).
6. Химические отходы и их утилизация: порядок, способы и стоимость // VYVOZ.ORG [Электронный ресурс]. – URL: <https://vyvoz.org/blog/himicheskie-othody/#Способы-утилизации-химических-отходов> (дата обращения: 06.02.2023).
7. Электронные отходы: проблемы для окружающей среды и способы утилизации // RECYCLE.NET [Электронный ресурс]. – URL: <https://rcycle.net/othody/vidy/elektronnye-problemy-dlya-okruzhayushhej-sredy-i-sposoby-utilizatsii> (дата обращения: 06.02.2023).
8. Резина // RECYCLE.NET [Электронный ресурс]. – URL: <https://rcycle.net/rezina> (дата обращения: 06.02.2023).
9. Утилизация отходов // EcoProverka.ru [Электронный ресурс]. – URL: <https://ecoproverka.ru/utilizatsiya-othodov/> (дата обращения: 06.02.2023).

10. Переработка отходов методом пиролиза: характеристика, преимущество, виды и сырье // VYVOZ.ORG [Электронный ресурс]. – URL: <https://vyvoz.org/blog/pererabotka-othodov-metodom-piroliza/#Что-такое-пиролиз-мусора> (дата обращения: 06.02.2023).
11. RECYCLEMAP [Электронный ресурс]. – URL: <https://recyclemap.ru> (дата обращения: 10.02.2023).
12. OPENRECYCLEMAP [Электронный ресурс]. – URL: <https://openrecyclemap.org> (дата обращения: 10.02.2023).
13. StarUML [Электронный ресурс]. – URL: <https://staruml.io> (дата обращения: 10.03.2023).
14. Обзор языка разметки HTML для новичков // Комьюнити [Электронный ресурс]. – URL: <https://timeweb.com/ru/community/articles/chto-takoe-html> (дата обращения: 15.04.2023).
15. Что такое CSS и зачем нужны каскадные таблицы стилей // Комьюнити [Электронный ресурс]. – URL: <https://timeweb.com/ru/community/articles/chto-takoe-css> (дата обращения: 15.04.2023).
16. Что такое JavaScript: назначение, особенности и сферы применения языка // Комьюнити [Электронный ресурс]. – URL: <https://timeweb.com/ru/community/articles/chto-takoe-javascript> (дата обращения: 15.04.2023).
17. Язык программирования Python: применение, особенности и перспективы // Комьюнити [Электронный ресурс]. – URL: <https://timeweb.com/ru/community/articles/chto-takoe-python> (дата обращения: 15.04.2023).
18. PostgreSQL [Электронный ресурс]. – URL: <https://www.postgresql.org> (дата обращения: 20.04.2023).
19. Figma [Электронный ресурс]. – URL: <https://www.figma.com> (дата обращения: 23.04.2023).

## References

1. Neorganicheskie otkhody: kharakteristika, klassifikatsiia, obrabotka // WARBLETONCOUNCIL [Electronic resource]. – URL: <https://ru1.warbletoncouncil.org/residuos-inorganicos-101>.
2. Utilizatsiia plastmassovykh otkhodov: osnovnye ekologicheskie problemy pererabotki plastika i poniatie biorazлагаемого plastika // Greenologia [Electronic resource]. – URL: <https://greenologia.ru/othody/utilizaciya-plastika.html#i-4>.
3. Ves' protsess pererabotki stekla: utilizatsiia kak sposob sokhranit' prirodu i zarabotat' // RECYCLE.NET [Electronic resource]. – URL: <https://rcycle.net/steklo/pererabotka-utilizatsiya>.
4. Ekologicheskaiia potrebnost' i ekonomicheskaiia vygoda pererabotki otkhodov shveinogo proizvodstva – kuda sdat' otkhody, oborudovanie, protsessy utilizatsii tekstilia // Ekspertnyi zhurnal o musore, otkhodakh proizvodstva [Electronic resource]. – URL: <https://vseomusore.com/pererabotka-otkhodov/ekologicheskaya-potrebnost-i-ekonomicheskaya-vygoda-pererabotki-otkhodov-shvejnogo-proizvodstva-kuda-sdat-othody-oborudovanie-protsessy-utilizatsii-tekstilya/>.
5. Kak neutilizirovannye metallicheskie otkhody vrediat ekologii // PKVM [Electronic resource]. – URL: <https://metallolom777.ru/blog/kak-neutilizirovannye-metallicheskie-otkhody-vredyat-ekologii/>.
6. KHimicheskie otkhody i ikh utilizatsiia: poriadok, sposoby i stoimost' // VYVOZ.ORG [Electronic resource]. – URL: <https://vyvoz.org/blog/himicheskie-othody/#Способы-утилизации-химических-отходов>.

7. Elektronnye otkhody: problemy dlia okruzhaiushchei sredy i sposoby utilizatsii // RECYCLE.NET [Electronic resource]. – URL: <https://rcycle.net/othody/vidy/elektronnye-problemy-dlya-okruzhayushhej-sredy-i-sposoby-utilizatsii>.
8. Rezina // RECYCLE.NET [Electronic resource]. – URL: <https://rcycle.net/rezina> (дата обращения: 06.02.2023).
9. Utilizatsiia otkhodov // EcoProverka.ru [Electronic resource]. – URL: <https://ecoproverka.ru/utilizatsiya-othodov/>.
10. Pererabotka otkhodov metodom piroliza: kharakteristika, preimushchestvo, vidy i syr'e // VYVOZ.ORG [Electronic resource]. – URL: <https://vyvoz.org/blog/pererabotka-othodov-metodom-piroliza/#Что-такое-пиролиз-мусора>.
11. RECYCLEMAP [Electronic resource]. – URL: <https://recyclemap.ru>.
12. OPENRECYCLEMAP [Electronic resource]. – URL: <https://openrecyclemap.org>.
13. StarUML [Electronic resource]. – URL: <https://staruml.io>.
14. Obzor iazyka razmetki HTML dlia novichkov // Kom'iuniti [Electronic resource]. – URL: <https://timeweb.com/ru/community/articles/что-такое-html>.
15. CHto takoe CSS i zacheм nuzhny kaskadnye tablitsy stilei // Kom'iuniti [Electronic resource]. – URL: <https://timeweb.com/ru/community/articles/что-такое-css>.
16. CHto takoe JavaScript: naznachenie, osobennosti i sfery primeneniia iazyka // Kom'iuniti [Electronic resource]. – URL: <https://timeweb.com/ru/community/articles/что-такое-javascript>.
17. Iazyk programmirovaniia Python: primeneniе, osobennosti i perspektivy // Kom'iuniti [Electronic resource]. – URL: <https://timeweb.com/ru/community/articles/что-такое-python>.
18. PostgreSQL [Electronic resource]. – URL: <https://www.postgresql.org>.
19. Figma [Electronic resource]. – URL: <https://www.figma.com>.

## **DESIGN AND DOCUMENTATION OF AN INFORMATION SYSTEM FOR ENVIRONMENTALLY FRIENDLY WASTE MANAGEMENT**

*Berdnikov Kirill S., Rakina Valeriia D.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia,  
[kirill.berdnikov27052002@gmail.com](mailto:kirill.berdnikov27052002@gmail.com)

The article considers the basic concepts in the field of environmental waste management and similar information systems. The main tasks and distinctive features of the designed system are revealed. The choice of means of design and development of the information system, database management system, the means of designing a prototype user interface. Models of information system, database model, user interface prototype were designed.

**Keywords:** design and documentation of the information system, environmentally friendly waste management.



## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ АБИТУРИЕНТА ДЛЯ ВЫБОРА ФАКУЛЬТЕТА И НАПРАВЛЕНИЯ В ПГНИУ ПРИ ПОСТУПЛЕНИИ**

*Валиев Антон Дмитриевич, Ракина Валерия Денисовна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, antonvaliev86@gmail.com

В статье проанализировано социологическое исследование о востребованности информационной системы для абитуриентов, путем анкетирования, в котором принимали участие учащиеся выпускных классов и первокурсники направления информационные системы и технологии, пермского государственного национально исследовательского университета, и рассмотрены существующие информационные системы для помощи абитуриентам с поиском информации о поступлении и сравнены между собой по ряду критериев. На основе анализа сформулированы функциональные требования разрабатываемой системы. Было выбрано программное обеспечение для проектирования и разработки информационной системы, а также инструменты для работы с базами данных на основе сравнительного анализа. Информационная система была спроектирована и задокументирована. Разработан макет интерфейса информационной системы. Спроектированная система предназначена для администрации ПГНИУ и способствует увеличению количества абитуриентов.

Ключевые слова: информационная система, абитуриенты, выбор направления обучения, uml-диаграммы.

### **Введение**

Для абитуриентов поступление в университет может стать одним из самых важных событий в их жизни. Однако, процесс поступления может быть очень сложным и запутанным, особенно когда речь идет о поиске информации о баллах и направлениях.

Из-за подобной ситуации абитуриенты могут тратить много времени на поиск и сбор информации. Таким образом, проектирование информационной системы (ИС) для абитуриентов, является актуальной проблемой, влияющей как на абитуриентов, так и на университет.

Целью данной работы является проектирование и документирование ИС абитуриента для выбора факультета и направления в ПГНИУ.

Предметом изучения в данной работе является автоматизация процесса поиска информации о направлениях в ПГНИУ. В свою очередь в качестве объекта изучения, выступает информация о направлениях обучения ПГНИУ.

### **Актуальность информационной системы для абитуриентов в ПГНИУ**

Для определения актуальности подобной ИС, необходимо провести социологическое исследование.

Социологическое исследование проводилось посредством анкетирования. Анкетирование состояло из ряда вопросов:

«Возникали ли у вас сложности с поиском информации о списках, поступивших за прошлый год?» (см. рис. 1);

«Возникали ли у вас сложности с поиском информации о направлениях?» (см. рис. 2);

«Хотели бы вы видеть систему для абитуриентов, в которой воедино собрана вся необходимая информация с возможностью фильтрации?» (см. рис. 3);



**Рис. 1. Определение сложности поиска информации о поступивших за прошлый год**



**Рис. 2. Определение сложности поиска информации о направлениях**



**Рис. 3. Необходимость информационной системы**

Анализируя полученные сведения, можно сделать следующие выводы: у каждого второго, возникали те или иные сложности с поиском информации о направлениях, в среднем каждый десятый сталкивался с критическими сложностями, в конечном счете 96% опрошенных, хотели бы видеть информационную систему абитуриентов.

### Анализ существующих систем

Прежде чем начать разрабатывать ИС стоит сравнить уже существующие системы.

К наиболее распространенным среди систем для абитуриентов можно выделить следующие системы: «Учёба.ру» [1], «Поступай правильно» [2], «Vuzopedia.ru» [3], «Vuzoteka.ru» [4]. Рассмотрение будем проводить по следующим критериям (см. табл. 1):

1. наличие фильтрации направлений по предметам;
2. наличие фильтрации направлений по баллам;

3. наличие фильтрации направлений по факультетам;
4. отображение минимального проходного балла за прошлый год;
5. актуальность информации;
6. удобство интерфейса;

Таблица 1. Сравнение аналогов

Наименование системы	Фильтрация по предметам	Фильтрация по баллам	Фильтрация по факультетам	Отображение минимального балла	Актуальность	Удобство интерфейса
Учёба.ру	+	+	+	+	+	-
Поступай правильно	-	-	-	+	+	-
Vuzopedia.ru	+	-	-	+	-	+
Vuzoteka.ru	+	-	-	+	-	+

Анализируя полученные данные, можно сделать вывод, что ни одна система не удовлетворяет поставленным требованиям полностью.

### **Функциональные требования ИС**

Анализируя существующие системы и результаты социологического опроса, можно выделить следующие функциональные требования к ИС для абитуриентов:

1. функция фильтрации по баллам;
2. функция фильтрации по предметам;
3. функция фильтрации по факультетам;
4. функция поиска направления;
5. система должна отображать наименование направления, минимальный проходной балл за прошлый год, предметы необходимые при поступлении;
6. функция проверки корректности введённых пользовательских данных;
7. функция проверки соединения с БД;
8. функция отображение ошибок в случае их наличия;

### **Программное обеспечение для проектирования ИС**

Правильно выбранные инструменты проектирования ИС помогают повысить эффективность и качество разработки, сократить время и затраты на проект.

К наиболее распространенным инструментам проектирования можно отнести Altova UModel [7], Star UML [8], Gliffy [9], Microsoft Visio [10], diagrams.net [11].

В качестве инструмента проектирования, был выбран StarUML, поскольку:

1. существует версия для стационарных компьютеров;
2. имеет свободную форму распространения и не требует оплаты при использовании;
3. имеет поддержку сохранения проекта;
4. поддерживает режим администратора предоставляющий возможность осуществлять настройку и управление функциональностью системы, а также управление учётными записями и правами доступа к системе
5. имеет достаточно наглядный и интуитивный интерфейс;

## **Выбор языка программирования**

Выбор языка программирования является важным этапом в разработке любой ИС или приложения. От правильного выбора языка зависит качество и эффективность создаваемой системы, ее производительность, масштабируемость и возможность интеграции с другими системами и многое другое.

Среди наиболее распространенных языков программирования выделяются C++, C#, Java, Python, PHP [5].

В качестве языка программирования, было принято решение, использовать язык C# поскольку:

1. поддерживает парадигму программирования ООП;
2. имеет встроенный инструмент сбора мусора;
3. имеет большую сферу применения;
4. синтаксис способствует увеличению читабельности кода;

## **Выбор среды разработки**

Среда разработки (IDE) обеспечивает удобное и эффективное, в плане временных затрат, окружение для создания, отладки и управления кодом программы.

На данный момент наиболее распространены такие среды разработки как Visual Studio Community, Eclipse, Spyder, IntelliJ IDEA.

В качестве среды разработки, было принято решение, использовать среду разработки Visual Studio Community разработки поскольку:

1. бесплатна к распространению для проектов, чей годовой доход менее одного миллиона долларов США;
2. имеет общего разработчика с языком программирования в лице компании Microsoft;
3. имеет встроенный инструмент работы с пакетами;
4. имеет интуитивный интерфейс;

## **Выбор инструмента для хранения и оперирования данными**

На рынке представлен достаточно большой спектр различных систем управления базами данных таких как: FireBird, PostgreSQL, MySQL, MS SQL [6].

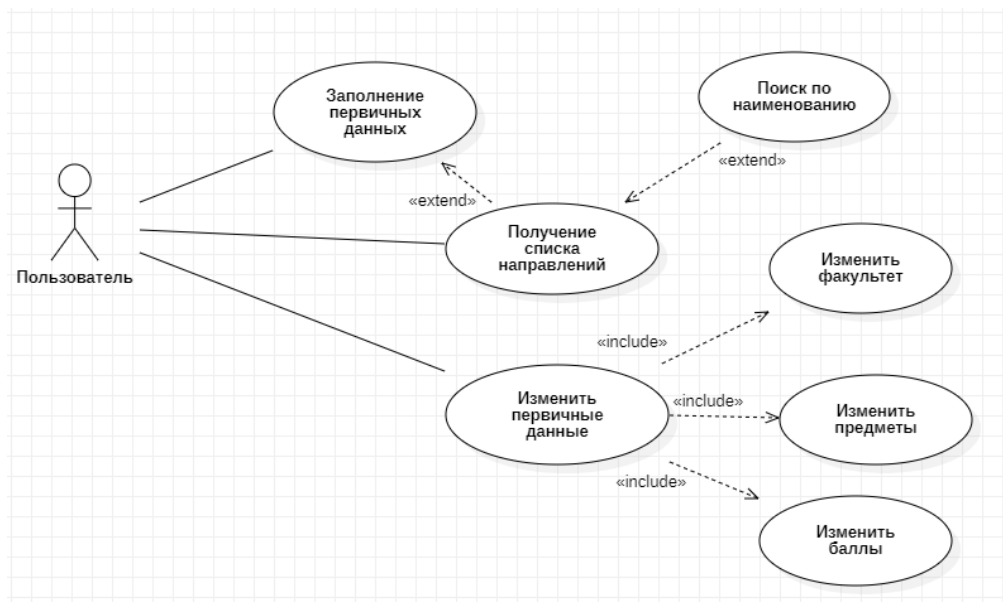
В качестве СУБД, было принято решение, использовать систему управления базами данных PostgreSQL поскольку:

1. ограничение на размер таблицы превышает тридцать два терабайта;
2. не имеет ограничения на размер базы данных;
3. имеется провайдер к EntityFramework;
4. имеет интуитивно понятный интерфейс панели администратора базы данных;

## **Проектирование информационной системы**

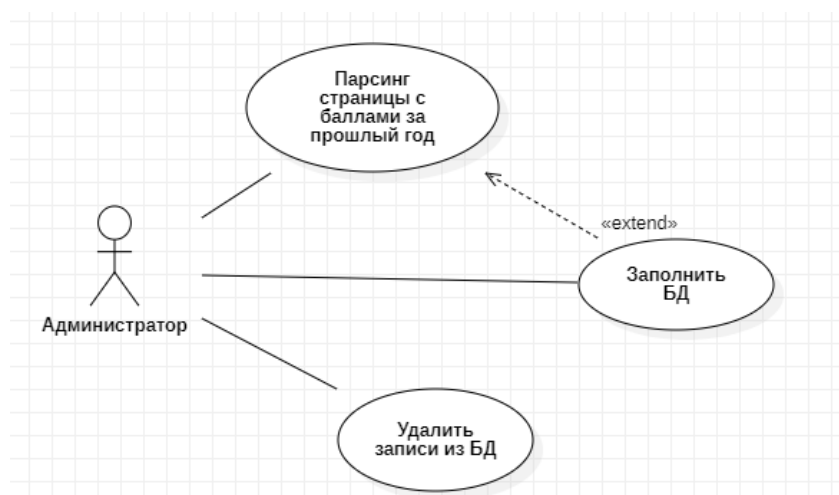
Разрабатываемая система предполагает два вида ролей: пользователь (абитуриент) и администратор базы данных.

Для описания возможностей каждой роли воспользуемся диаграммой прецедентов [1], описывающей, какой функционал информационной системы доступен каждой роли (см. рис. 4 и рис. 5).



**Рис. 4. Диаграмма прецедентов для пользователя**

Пользователь – это абитуриент, ищущий направление к поступлению.



**Рис. 5. Диаграмма прецедентов для администратора**

Администратора – это человек, который поддерживает актуальность данных, корректирующий сведения о направлениях и ежегодно обновляющий базу данных направлений. Также стоит учесть, что в функционале администратора должен быть инструмент, позволяющий автоматизировано получать информацию с сайта ПГНИУ.

### Прототип интерфейса

Макет был разработан при помощи конструктора шаблонов интерфейсов Just in mind [13].

Раздел отображения направлений, отображает список направлений университета и будет иметь подобный внешний вид (см. рис. 6).

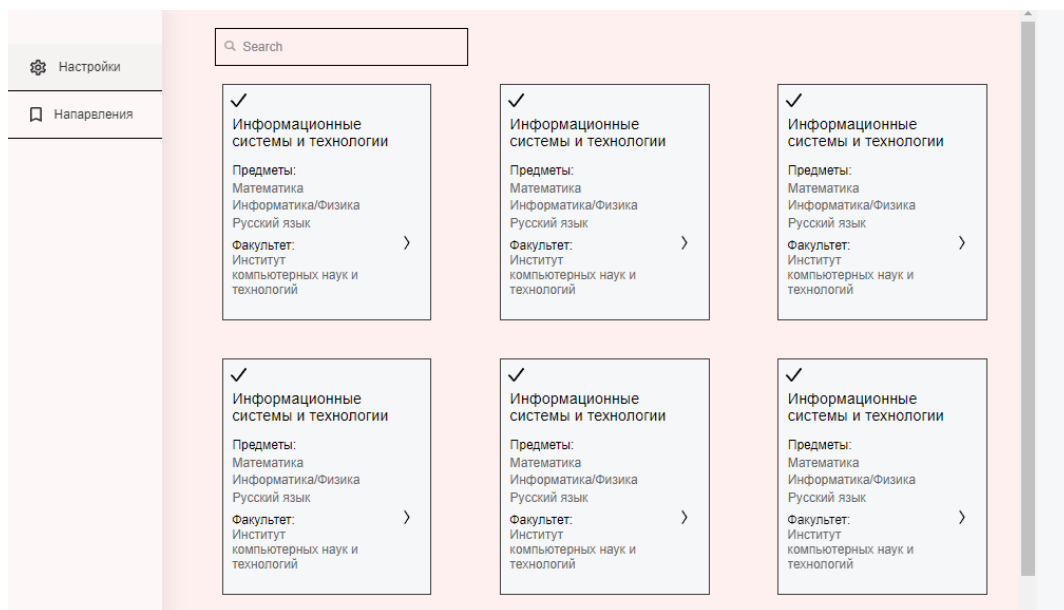


Рис. 6. Список направлений

Раздел редактирования настроек фильтрации представляет из себя список параметров, по которым фильтруются направления и будет иметь подобный внешний вид (см. рис. 7).

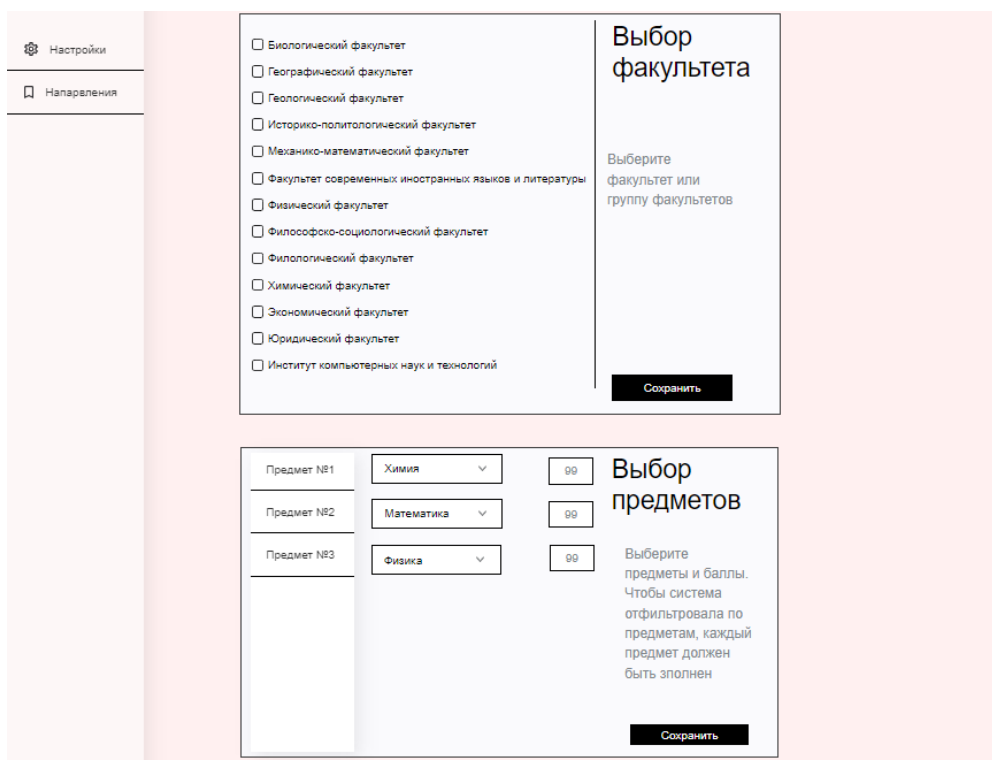


Рис. 7. Список направлений

### Библиографический список

1. Новиков Ф.А. Анализ и проектирование на UML. Учебное пособие. – СПб: Университет ИТМО, 2007. – 283 с.
1. Учеба.Ру. Специальности [Электронный ресурс]. – 2023. – URL: <https://perm.ucheba.ru/for-abiturients/speciality> (дата обращения 16.05.2023)

2. Поступай правильно. Государственные вузы [Электронный ресурс]. – 2023. – URL: [https://поступай-правильно.рф/search?tab=universities&type\\_code=PUBLIC](https://поступай-правильно.рф/search?tab=universities&type_code=PUBLIC) (дата обращения 16.05.2023)
3. Vuzopedia.ru. Программы [Электронный ресурс]. – 2023. – URL: <https://vuzopedia.ru/program/bakispec> (дата обращения 16.05.2023)
4. Vuzoteka. Специальности [Электронный ресурс]. – 2023. – URL: <https://vuzoteka.ru/вузы/специальности> (дата обращения 16.05.2023)
5. Cyclowiki. Сравнение языков программирования [Электронный ресурс]. – 2023. – URL: [https://cyclowiki.org/wiki/Сравнение\\_языков\\_программирования#Сопоставление\\_с\\_образцом](https://cyclowiki.org/wiki/Сравнение_языков_программирования#Сопоставление_с_образцом) (дата обращения 16.05.2023)
6. Wikipedia. Comparison of relational database management systems [Электронный ресурс]. – 2023. – URL: [https://en.wikipedia.org/wiki/Comparison\\_of\\_relational\\_database\\_management\\_systems#Fundamental\\_features](https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems#Fundamental_features) (дата обращения 16.05.2023)
7. Altova UModel. [Электронный ресурс]. – 2023. – URL: <https://www.altova.com/umodel> (дата обращения 18.05.2023)
8. StarUML. [Электронный ресурс]. – 2023. – URL: <https://staruml.io/> (дата обращения 18.05.2023)
9. Gliffy. [Электронный ресурс]. – 2023. – URL: <https://www.gliffy.com/> (дата обращения 18.05.2023)
10. Microsoft Visio. Функции и возможности [Электронный ресурс]. – 2023. – URL: <https://www.microsoft.com/ru-ru/microsoft-365/visio/flowchart-software> (дата обращения 18.05.2023)
11. diagrams.net. [Электронный ресурс]. – 2023. – URL: <https://app.diagrams.net/index.html> (дата обращения 18.05.2023)
12. Just in Mind. [Электронный ресурс]. -2023. – URL: <https://www.justinmind.com/> (дата обращения 20.05.2023)

## References

1. *Novikov F.A. Analiz i proektirovanie na UML. Uchebnoe posobie.* SPb: Universitet ITMO; 2007. 283 p. (In Russ.).
2. Ucheba.ru. Professions. 2023. [Electronic resource]. – URL: <https://perm.uceba.ru/for-abiturients/speciality>. (In Russ).
3. Postupaj pravil'no. State universities. 2023. [Electronic resource]. – URL: [https://поступай-правильно.рф/search?tab=universities&type\\_code=PUBLIC](https://поступай-правильно.рф/search?tab=universities&type_code=PUBLIC). (In Russ).
4. Vuzopedia.ru. Programs. 2023. [Electronic resource]. – URL: <https://vuzopedia.ru/program/bakispec>. (In Russ).
5. Vuzoteka. Professions. 2023. [Electronic resource]. – URL: <https://vuzoteka.ru/вузы/специальности>. (In Russ).
6. Cyclowiki. Comparison of programming languages. 2023. [Electronic resource]. – URL: [https://cyclowiki.org/wiki/Сравнение\\_языков\\_программирования#Сопоставление\\_с\\_образцом](https://cyclowiki.org/wiki/Сравнение_языков_программирования#Сопоставление_с_образцом). (In Russ).
7. Wikipedia. Comparison of relational database management systems. 2023. [Electronic resource]. – URL: [https://en.wikipedia.org/wiki/Comparison\\_of\\_relational\\_database\\_management\\_systems#Fundamental\\_features](https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems#Fundamental_features).

8. Altova UModel. 2023. [Electronic resource]. – URL: <https://www.altova.com/umodel>.
9. StarUML. 2023. [Electronic resource]. – URL: <https://staruml.io/>.
10. Gliffy. 2023. [Electronic resource]. – URL: <https://www.gliffy.com/>.
11. Microsoft Visio. – 2023. [Electronic resource]. – URL: <https://www.microsoft.com/ru-ru/microsoft-365/visio/flowchart-software>.
12. diagrams.net. 2023. [Electronic resource]. – URL: <https://app.diagrams.net/index.html>.
13. Just in Mind. 2023. [Electronic resource]. – URL: <https://www.justinmind.com/>.

## **DESIGNING AND DOCUMENTING AN APPLICANT'S INFORMATION SYSTEM FOR CHOOSING A FACULTY AND A REFERRAL TO PSU UPON ADMISSION**

*Valiev Anton D., Rakina Valeria D.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, antonvaliev86@gmail.com

In the course of the work, the applicant's information system was designed and documented for choosing the faculty and referral to PSU upon admission. In the course of the work, a sociological study was conducted. Similar information systems were analyzed. Functional requirements of the developed system were formulated. Software for the design and development of IP was selected, as well as tools for working with databases based on comparative analysis. An information system was designed. The layout of the information system interface has been developed. The developed system is intended for the administration of Perm State University.

Keywords: information system, applicants, choosing the direction of study, uml-diagrams.



## **ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ВРЕМЕНИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ В КОМПЬЮТЕРНОМ КЛАССЕ**

*Васильев Игорь Дмитриевич, Василюк Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, vasilevid1111@mail.ru

Рассматривается автоматизация учета времени самостоятельной работы студентов в компьютерном классе. Работа содержит описание логики работы информационной системы в виде диаграмм и модели данных. Дана теоретическая основа для разработки информационной системы учета времени студентов. Проведен анализ аналогов, выявлены требования к разрабатываемой системе. Описаны процесс выбора инструментальных средств проектирования и создания модели информационной системы. Представлено описание используемых инструментальных средств разработки. Данная информационная система может быть использована работником компьютерного класса и студентами, желающими записаться в компьютерный класс на самостоятельную работу.

Ключевые слова: учет времени, самостоятельная работа студента, компьютерный класс, информационная система, uml-диаграммы.

### **Введение**

В современном мире решающим фактором развития является информация. Используя её, человек может строить заключения и формировать свое личное и коллективное будущее. Для того чтобы не запутаться и не потеряться в многочисленных информационных потоках, человек создает и совершенствует инструменты управления информацией. Основным инструментом управления сведениями об окружающем мире и протекающих в нем процессах является информационная система.

Информационная система (ИС) – система, которая организует хранение и манипулирование информацией о предметной области. Такое определение в широком смысле описывает ГОСТ 34.321-96 [1]. В узком смысле, информационная система – это система, предназначенная для автоматизации бизнес-процессов компании или отдельных пользователей, позволяющая обработку данных, включая удаление, изменение и добавление. Главной задачей ИС является полное удовлетворение информационных потребностей конечного пользователя.

Разработка информационной системы для учета времени студентов в компьютерном классе становится актуальной задачей, поскольку в высших учебных заведениях происходит компьютеризация и переход к электронному документированию, а также возникает потребность в быстром и удобном доступе к данным и их обработке. Кроме того, важным аспектом является возможность доступа к ресурсу из любого места, где имеется Интернет-соединение.

Таким образом, объектом исследования является учет времени самостоятельной работы студентов. Предметом исследования является автоматизация учета времени самостоятельной работы студентов в компьютерном классе.

## Обзор литературы

Целью данного обзора является выяснение основных проблем в управлении процесса учета самостоятельной работы студентов (СРС) в компьютерном классе. Самостоятельная работа имеет огромное значение в жизни студентов, так как она способствует развитию творческого мышления и формированию уникального взгляда на дисциплины, которые нельзя достичь без личного развития. Данное утверждение подтверждается учебным планом для направления "Информационные системы и технологии" [2], в котором отведено 5042 часа для самостоятельной работы из общей планируемой трудоемкости в 7996 часов. Это составляет около 63% всей учебной деятельности студента в университете. Так как во множестве дисциплин требуется дорогостоящее компьютерное оснащение и программное обеспечение, учащимся приходится пользоваться вычислительными возможностями университета. По причине этого, студент должен правильно распределять свои временные ресурсы. Поэтому особенно важно знать наиболее надежные и эффективные методы создания благоприятных условий для организации и учета СРС в компьютерном классе.

Данную тему частично рассматривает методист Чебаркульского профессионального техникума Т.А. Кудрявцева в своей статье «Проблемы организации самостоятельной работы студентов в условиях реализации федеральных государственных образовательных стандартов среднего профессионального образования третьего поколения» [3]. Основными трудностями при организации СРС в компьютерном классе стали:

- 1) недостаток навыков планирования самостоятельной работы у студентов;
- 2) отсутствие достаточных навыков у студентов по оформлению СРС;
- 3) ограниченные знания технологий у сотрудников и студентов;
- 4) неумение рассчитать трудоемкость заданий СРС.

Таким образом, одним из ключевых решений является обеспечение необходимыми материально-техническими условиями, включающих наличие необходимого уровня оснащённости кабинетов, предоставление методических материалов для ознакомления работников с бизнес-процессом и наглядных инструкций для оформления СРС, обеспечение необходимой информацией для обучения студента самостоятельному планированию своего времени в компьютерном классе и организацию учета для СРС.

В статье "Организация самостоятельной работы студентов в условиях реализации ФГОС" [4], заместитель директора по учебно-методической работе Н. В. Ананьина рассмотрела вопрос о реализации учета СРС с точки зрения локальных актов. В статье было указано, что основными документами, регламентирующими этот процесс, являются:

- 1) положение о самостоятельной работе студентов, определяющее сущность самостоятельной работы студентов, ее назначение, планирование, формы организации и виды контроля;
- 2) положение о журнале учета самостоятельной работы студентов, определяющее порядок оформления и ведения журналов учета СРС.

В связи с этим, можно предположить, что заполнение журналов учета СРС, подготовка каждого студента к самостоятельной работе без использования современных информационных технологий в сфере обработки данных замедляет процесс осуществления СРС. В свою очередь, возникает проблема хранения, сортировки, доступа к данным и их своевременной передачи. Хранение бумажных документов требует значительных затрат и усилий, поскольку

ку необходимо обеспечить их безопасное хранение, оценить риски повреждения информации и предусмотреть дублирование на бумажных носителях. Кроме того, заполнение документов в письменной форме может привести к многочисленным ошибкам и опечаткам, которые могут иметь неблагоприятные последствия.

Важно отметить, что проектирование, разработка и документирование информационной системы учета времени студентов для самостоятельной работы в компьютерном классе является необходимостью, потому что использование информационных технологий и вычислительной техники существенно снизит имеющиеся недостатки в процессе заполнения документов, записи СРС. Это позволит сократить трудозатраты и повысить достоверность данных.

### **Теоретическое обоснование разработки ИС**

В настоящее время все большее число учебных учреждений внедряют и используют различные ИС для автоматизации работы преподавателей и подразделений университета. Компьютеризация бизнес-процессов позволяет добиваться высокой скорости ведения документов и предотвращает большинство допускаемых ошибок при фиксации информации.

Как правило, для того чтобы студенту записаться на самостоятельную работу в компьютерный класс, ему каждый раз необходимо лично подходить к лаборанту или ответственному за компьютерный класс, который, в свою очередь, по разным причинам может отсутствовать на рабочем месте. Использование информационной системы учета СРС может решить эту проблему, предоставляя возможность студентам осуществлять запись через сеть Интернет.

Более того, сотрудник компьютерного класса должен понимать, сколько студентов записалось и в какое время, за какими компьютерами они будут работать, сколько времени потребуется для решения их задач, и с какой целью им потребовался компьютер. Сохранение такой информации на бумажных носителях не является эффективным, поскольку анализировать такие данные гораздо сложнее по сравнению с использованием информационных систем.

Таким образом, проектирование такой ИС является актуальной проблемой, которая позволит упростить доступ студентов к записи и облегчит анализ отчетов.

### **Обзор и сравнение существующих аналогов ИС**

На сегодняшний день в открытом доступе существует некоторое количество систем для учета времени СРС студента. Многие такие системы разрабатываются внутри учебных учреждений и являются частью системы университета. Однако большую часть рынка занимают коммерческие продукты для учета клиентов в разных предметных областях. Были рассмотрены следующие системы, предназначенные для СРС студентов, а также изучим наиболее подходящие и универсальные продукты, которые можно приобрести и использовать в компьютерном классе:

- 1) «1С: Университет» [5]
- 2) «Электронный журнал» [6]
- 3) «Yclients» [7]
- 4) «Универсальная система учета» [8]

Рассмотренные выше системы являются многофункциональными и имеют как преимущества, так и ряд недостатков. Например, одна из рассмотренных систем не предполагает использование в учебных учреждениях, не являющихся колледжами Подмоскovie, к тому же, эта система рассчитана на форму и структуру организации только для этих образовательных организаций.

Вместе с тем, решение от «1С», хоть и является универсальным, но, тем не менее, внедрение такой огромной системы является ресурсозатратным процессом, и не является оптимальным решением в данной предметной области.

Кроме того, многие из представленных оболочек не ориентированы на учет времени самостоятельной работы студента, что является критическим в выборе программного продукта. Особенно важной чертой продукта должна быть запись студента через Интернет, что присутствует только в 2-х рассмотренных системах.

Анализ существующих аналогов демонстрирует, что разработка своей ИС по учету времени СРС в компьютерном классе предпочтительнее, чем покупка аналогичного продукта у стороннего разработчика. При этом проектирование, документирование и разработка программного продукта позволит организовать именно те требования, которые необходимы работникам компьютерного класса и студентам, а также избежать излишней функциональности, присутствующей в других программных продуктах.

### **Требования к проектируемой ИС**

На основе анализа существующих систем сформулированы следующие функциональные требования, которые должны осуществляться разрабатываемой ИС:

- учет времени работы студента в компьютерном классе;
- учет времени бронирования компьютера студентом;
- контроль свободных компьютеров в классе;
- хранение данных о компьютерах, студентах и времени их работы;
- предоставление отчетов по запросу руководителя подразделения;
- удобное представление информации работнику образовательного учреждения;
- дистанционный доступ к записи для студентов;
- отметка о посещении.

### **Моделирование ИС**

Доступ к ИС для лаборанта и для студентов будет осуществляться через Интернет-сайт. Для достижения целей визуального проектирования с использованием нотации UML вначале строится диаграмма вариантов использования. Диаграмма вариантов использования – это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки [9].

Таким образом, логика работы информационной системы представлена на следующих рисунках (рис. 1, рис. 2).

Рассмотрим диаграмму прецедентов № 3 (рис. 3). На ней изображены два актера: студент и лаборант. Лаборант, помимо записи в компьютерный класс, может создать компьютер для последующей записи студентов.

Для студентов, которые хотят записаться в компьютерный класс, не требуется указывать свой студенческий номер, поскольку система авторизации уже точно определяет, какой

пользователь работает с системой. При создании записи автоматически подставляется информация о текущем студенте. Таким образом, процесс записи для студента аналогичен процессу для лаборанта, за исключением ввода студенческого номера.

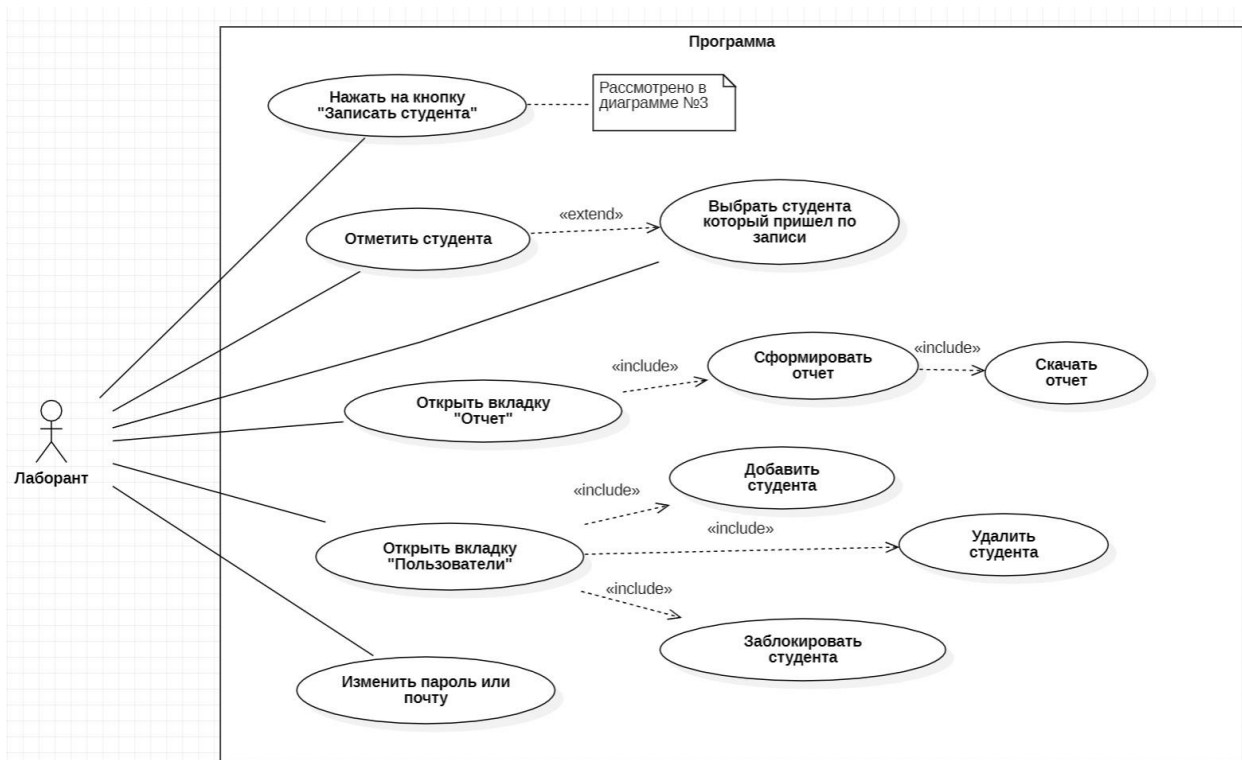


Рис. 1. Диаграмма прецедентов «Лаборант» №1

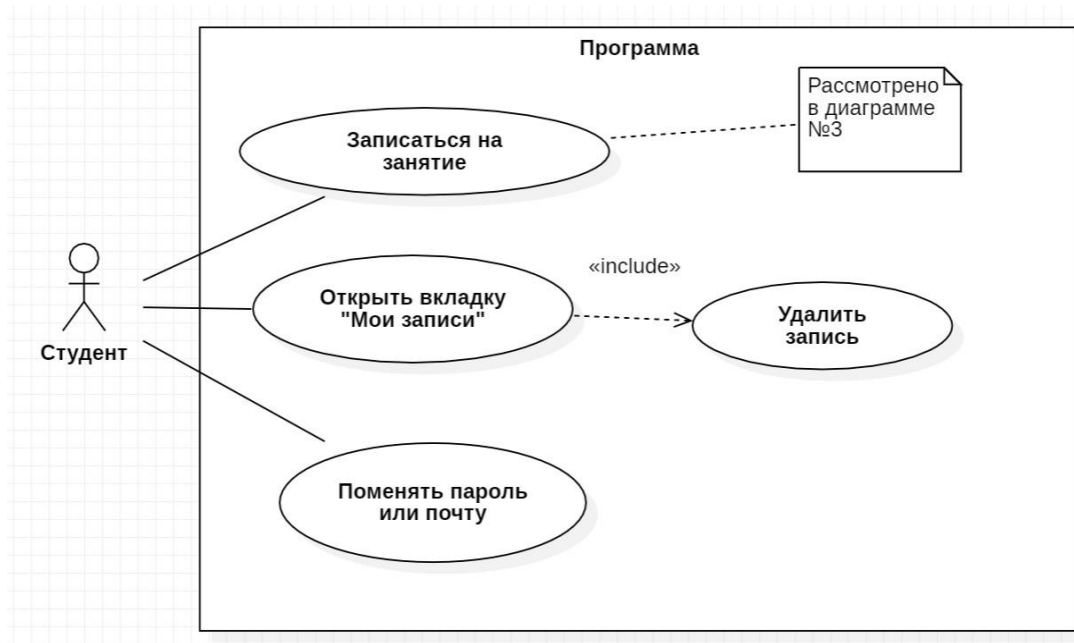


Рис. 2. Диаграмма прецедентов «Студент» №2

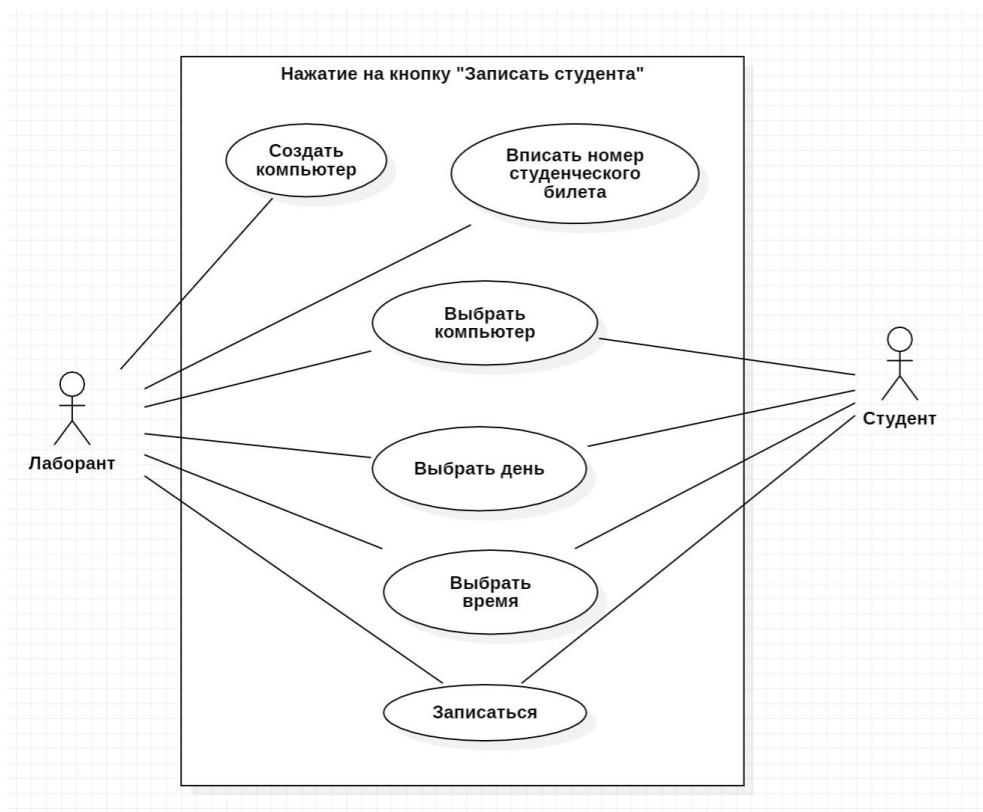


Рис. 3. Диаграмма прецедентов «Записать студента» №3

### Серверная и клиентская части

В информационной системе для учета работы студентов в компьютерном классе наиболее приоритетными составляющими являются безопасность, производительность, надежность и количество разработанных библиотек. Для сочетания всех свойств системы были выбраны следующие технологии: язык программирования Java [10], фреймворк Spring Boot [11], библиотека Spring JPA [12], библиотека Spring Security [13], обработчик шаблонов FreeMarker [14] и другие библиотеки, система управления базами данных PostgreSQL [15]. Важно отметить, что все эти инструменты являются бесплатными и могут быть использованы в коммерческих проектах без необходимости платить за лицензии.

Также в работе был использован git [16] для удобства отслеживания версионности приложения и библиотека Lombok [17] для упрощения написания кода и ускорения разработки.

В итоге, благодаря объединению этих технологий, конечное приложение будет удовлетворять потребностям пользователя, соответствовать нормам безопасности и надежности, а также позволит повысить читаемость и поддерживаемость кода, что послужит дальнейшему развитию проекта.

Для реализации клиентской части веб-приложения было решено использовать технологии HTML, CSS и JavaScript. Для ускорения процесса разработки клиентской части веб-приложения был использован Bootstrap [18].

### Создание базы данных

Созданная база данных в PostgreSQL представлена на рис. 4. Связи между таблицами выглядят следующим образом:

- таблица «users» связана с таблицей «visit» по полю user\_id;
- таблица «computer» связана с таблицей «visit» по полю comp\_id.

При попытке удаления информации из таблиц «users» и «computer», если есть записи с этим идентификатором в таблице, «visits», запись не удалится и выдаст ошибку, т.е. удаление будет неуспешным. Это поможет поддерживать целостность данных и избегать неконсистентности информации в базе данных.

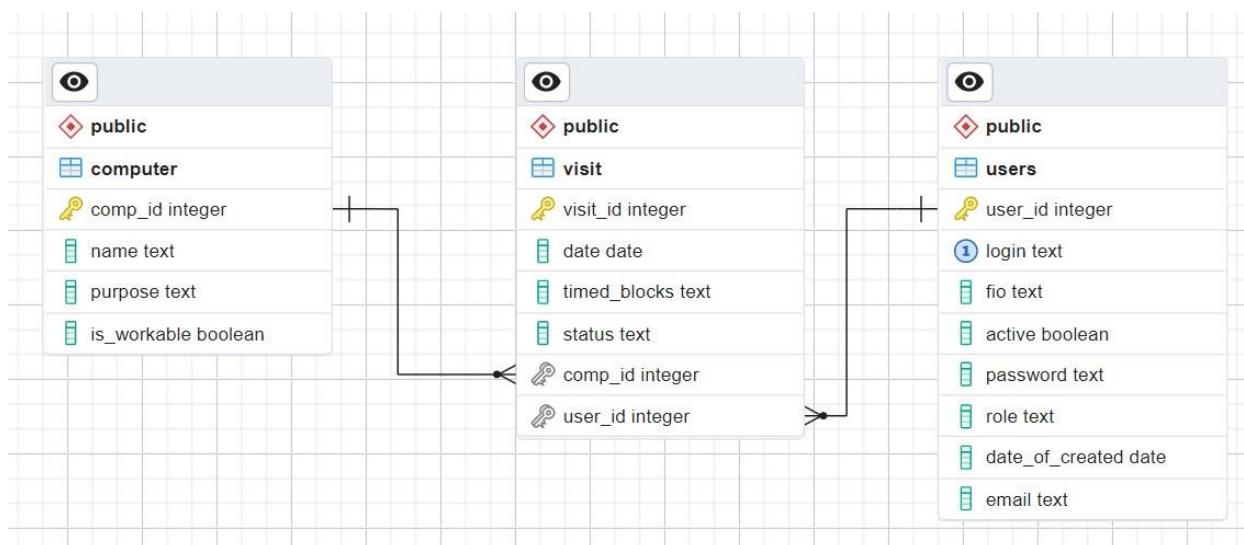


Рис. 4. База данных студентов

### Реализация прототипа ИС

Разработанный прототип имеет множество классов. Ниже представлены основные классы и их функциональность:

- Controller отвечает за управление потоком приложения. Он принимает запросы пользователей, обрабатывает их, и затем отвечает соответствующим образом клиенту. Для получения или изменения данных он взаимодействует с классом Service.
- Service отвечает за бизнес-логику приложения, которая обрабатывается сервисным уровнем. Данные, полученные из класса DAO, обрабатываются и затем передаются обратно контроллеру. Для того чтобы получить или изменить данные, он также взаимодействует с классом DAO.
- DAO отвечает за взаимодействие с базой данных. Он возвращает данные в класс Service после получения их из БД. При необходимости он также сохраняет данные в ней.
- DTO отвечает за объекты из базы данных. С помощью него производится обмен информации между классами, а также осуществляется чтение и запись в БД.

При разработке во фреймворке Spring Boot используется паттерн проектирования Model-View-Controller (MVC) [19], и поставляемые классы соответствуют этому паттерну. Каждый слой взаимодействует только с выше- или нижележащим слоем, что делает приложение более структурированным и простым в обслуживании.

При рассмотрении логики информационной системы, отдельное внимание уделим организации защиты доступа к веб-сайту. Благодаря настройке класса SecurityConfig, каждого пользователя, заходящего неавторизованным на любой каталог сайта, автоматически перенаправляет на страницу авторизации, где его просят ввести логин и пароль. После того, как пользователь аутентифицировался, сервер генерирует csrf-токен [20], и теперь при отправке post-запросов от клиента к серверу вместе с данными передается csrf-токен, который используется для проверки того, что аутентифицированный пользователь действительно является лицом, выполняющим запросы к приложению. Поскольку токен хранится в сессии пользова-

теля и изменяется каждый раз при повторном создании сессии, вредоносное приложение не может получить к нему доступ.

После прохождения формы авторизации, пользователю становятся доступны только те веб-страницы, которые относятся к его роли. Например, пользователю с ролью «ROLE\_STUDENT» доступны следующие веб-документы, не считая формы login.html:

- computers.html (выбор компьютера);
- shedule.html (выбор даты и времени);
- visits.html (просмотр своих записей);
- profile.html (изменение почты/пароля).

В свою очередь, в базе данных хранятся логин и пароль, захешированный функцией Bcrypt [21]. Bcrypt – это популярная криптографическая функция хэширования, которая часто используется для безопасного хранения паролей в базах данных. Это распространенный вариант хэширования паролей во многих приложениях, поскольку он не требует больших вычислительных затрат и устойчив к атакам методом перебора.

Теперь рассмотрим, процесс записи студента в компьютерный класс с точки зрения работы сервера. При входе на сайт после авторизации его встречает таблица с выбором компьютеров – эта таблица сформирована путем чтения всех компьютеров, у которых поле «is\_workable» принимает значение «true», так как студентам не нужно видеть неработающие компьютеры. При выборе компьютера студент нажимает кнопку около нужного устройства и переходит на следующий документ, в котором изображен календарь и время для записи. В это же время на сервере считываются все записи, начиная с текущей даты, и с помощью метода getShedulesByCompIdOnDate создается HashTable, в котором хранятся значения даты и занятые дни. Если этот день полностью занят (нет свободного времени), то тогда напротив даты проставляется значение «out». После этого, HashTable передается в шаблон FreeMarker и обрабатывается двумя скриптами showCalendar и showTimeTable для отображения информации, полученной из базы данных, пользователю. Следующим шагом пользователю следует выбрать подходящую дату и время, но есть ограничения, которые не позволят выбрать неподходящую дату, а именно будут недоступны следующие варианты:

- 1) даты в календаре, которые находятся раньше сегодняшней;
- 2) суббота и воскресенье;
- 3) даты, в которых все время уже выбрано другими студентами (нет свободного времени);
- 4) время, которое превосходит 4 часа за одну запись;
- 5) время, в интервале которого, уже есть другая запись студента.

При выборе времени студенту также дублируется время записи на странице. После выбора даты и времени из доступных пользователь нажимает на кнопку «Записаться», и информация о выбранных компьютере, времени и дате отправляется post-запросом на сервер, где и сохраняются в базу данных.

Для работника компьютерного класса, если он хочет записать студента в компьютерный класс, на странице с выбором даты и времени добавляется сверху форма с вводом логина пользователя, а в остальном, порядок действий тот же.

На странице о посещениях студента есть таблица его записей, где он может отменить любую из них, нажав кнопку «Отменить запись», тем самым отправляя post-запрос с идентификатором отмененной записи серверу.



Обратим внимание на действия работника компьютерного класса и его взаимодействие с системой с точки зрения работы сервера. После входа в систему перед пользователем открываются две таблицы с отмеченными и неотмеченными посещениями студентов. В первой таблице перед каждой записью располагаются кнопки «Пришел» и «Пропустил», во второй, соответственно, кнопка «Вернуть». Путем нажатия пользователь генерирует запрос серверу в следующем виде: «/check/{status}/{visit\_id}», где {status} – это статус, на который нужно поменять запись, а {visit\_id} – идентификатор записи. Далее, этот запрос отправляется в функцию checkUser, где изменяется статус записи и сохраняется в базе данных.

Следующим шагом рассмотрим формирование отчета для работника. Путем выбора логина студента, имени компьютера, даты «с» и даты «по», пользователь может сформировать отчет обо всех записях по заданным критериям. Если пользователь предпочел не вводить данных, тогда это поле не учитывается при выборе.

В дополнение к этому, пользователь может скачать отчет в формате электронных таблиц (.xlsx) путем отправки post-запроса. На сервере создается объект Workbook из библиотеки «org.apache.poi» в функции createXslxFile, далее из объекта Shedule преобразуется к странице электронной таблицы и сохраняется в Workbook. Сформированная таблица в точности выглядит, как таблица из веб-документа.

Работник компьютерного класса может управлять пользователями. На странице users.html может редактировать, удалять и добавлять новых пользователей. При регистрации пользователя, пароль отправляется на почту, указанную при регистрации, и сохраняется в базе данных в зашифрованном виде с помощью BCrypt, чем обеспечивает безопасное хранение паролей.

### **Заключение**

В результате проделанной работы была спроектирована, разработана и задокументирована информационная система учета времени самостоятельной работы студентов в компьютерном классе. Данная информационная система может быть использована как для студентов ВУЗа, так и для учеников школ, для записи на самостоятельную работу в компьютерном классе.

### **Библиографический список**

1. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200017662> (дата обращения: 22.05.2022)
2. Учебный план. Направление 09.03.02. Информационные системы и технологии направленность (профиль) Безопасность информационных систем степень бакалавр [Электронный ресурс]. – 2020. – URL: [http://www.psu.ru/files/docs/obrazovanie/bachelors/2020/up/09\\_03\\_02\\_bis\\_up.pdf](http://www.psu.ru/files/docs/obrazovanie/bachelors/2020/up/09_03_02_bis_up.pdf) (дата обращения 03.03.2022)
3. Кудрявцева Т.А., Пуртова Т.И., Соколова М.Г. Проблемы организации самостоятельной работы студентов в условиях реализации федеральных государственных образовательных стандартов среднего профессионального образования третьего поколения // Инновационное развитие профессионального образования. – 2015. – N 2. – С. 39-42.
4. Ананьина Н. В. Организация самостоятельной работы студентов в условиях реализации ФГОС // Образование. Карьера. Общество. – 2013. – N 4. – С. 51-55.
5. 1С:Университет. Возможности продукта [Электронный ресурс]. – 2022. – URL: <https://solutions.1c.ru/catalog/university/features> (дата обращения 03.03.2022)

6. Платформа «Цифровой колледж Подмосковья». Руководство пользователя [Электронный ресурс]. – 2022. – URL: [https://e-learning.tsppk-mo.ru/seo/help/module\\_2\\_2\\_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAAABAI%3D](https://e-learning.tsppk-mo.ru/seo/help/module_2_2_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAAABAI%3D) (дата обращения 05.03.2022)
7. Yclients. Описание продукта [Электронный ресурс]. – 2022. – URL: <https://www.yclients.com/ru> (дата обращения 05.03.2022)
8. Универсальная система учета [Электронный ресурс]. – 2022. – URL: [usu.kz](https://usu.kz) (дата обращения 05.03.2022)
9. Леоненков А. Нотация и семантика языка UML [Электронный ресурс]. – 2022. – URL: <https://intuit.ru/studies/courses/32/32/info> (дата обращения 10.03.2022)
10. Java Documentation // Oracle [Электронный ресурс]. – URL: <https://docs.oracle.com/en/java/> (дата обращения: 22.05.2023).
11. Spring Boot Reference Documentation // Spring [Электронный ресурс]. – URL: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (дата обращения: 22.05.2023).
12. Spring Data JPA – Reference Documentation // Spring [Электронный ресурс]. – URL: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> (дата обращения: 22.05.2023).
13. Spring Security // Spring [Электронный ресурс]. – URL: <https://docs.spring.io/spring-security/reference/> (дата обращения: 22.05.2023).
14. Apache FreeMarker Manual // FreeMarker [Электронный ресурс]. – URL: <https://freemarker.apache.org/docs/> (дата обращения: 22.05.2023).
15. Documentation // PostgreSQL URL: <https://www.postgresql.org/docs/> (дата обращения: 22.05.2023).
16. git // git [Электронный ресурс]. – URL: <https://git-scm.com/> (дата обращения: 22.05.2023).
17. Project Lombok // Project Lombok [Электронный ресурс]. – URL: <https://projectlombok.org/> (дата обращения: 22.05.2023).
18. Bootstrap // Bootstrap [Электронный ресурс]. – URL: <https://getbootstrap.com/> (дата обращения: 22.05.2023).
19. Model View Controller (MVC) Design Pattern in Java // Java Guides [Электронный ресурс]. – URL: <https://www.javaguides.net/2019/08/model-view-controller-mvc-design-in-java.html> (дата обращения: 22.05.2023).
20. A Guide to CSRF Protection in Spring Security // Baeldung [Электронный ресурс]. – URL: <https://www.baeldung.com/spring-security-csrf> (дата обращения: 27.05.2023).
21. Bcrypt Step by Step // medium [Электронный ресурс]. – URL: <https://medium.com/bootdotdev/bcrypt-step-by-step-99dd76a3b16a> (дата обращения: 22.05.2023).

## References

1. GOST 34.321-96 Information Technology. Database System Standards. Reference Data Management Model. [Electronic resource]. – URL: <https://docs.cntd.ru/document/1200017662> (accessed on 22.05.2022) (in Russian)
2. Curriculum. Major 09.03.02. Information Systems and Technologies, specialization Information Systems Security, Bachelor's degree. [Electronic resource]. – URL: [http://www.psu.ru/files/docs/obrazovanie/bachelors/2020/up/09\\_03\\_02\\_bis\\_up.pdf](http://www.psu.ru/files/docs/obrazovanie/bachelors/2020/up/09_03_02_bis_up.pdf) (accessed on 03.03.2022) (in Russian)

3. Kudryavtseva, T. A., Purtove, T. I., Sokolova, M. G. Problems of Organizing Students' Independent Work in the Conditions of Implementing Federal State Educational Standards of the Third Generation of Vocational Education. *Innovacionnoe razvitie professional'nogo obrazovaniya* [Innovative Development of Vocational Education], 2015, no. 2, pp. 39-42. (in Russian)
4. Ananina, N. V. Organizing Students' Independent Work in the Conditions of Implementing Federal State Educational Standards. *Obrazovanie. Kar'era. Obshchestvo* [Education. Career. Society.], 2013, no. 4, pp. 51-55. (in Russian)
5. 1C:University. Product Features. [Electronic resource]. – URL: <https://solutions.1c.ru/catalog/university/features> (accessed on 03.03.2022) (in Russian)
6. "Digital College of Moscow Region" Platform. User Manual. [Electronic resource]. – URL: [https://e-learning.tspk-mo.ru/seo/help/module\\_2\\_2\\_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAAABAI%3D](https://e-learning.tspk-mo.ru/seo/help/module_2_2_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAAABAI%3D) (accessed on 05.03.2022) (in Russian)
7. Yclients. Product Description. [Electronic resource]. – URL: <https://www.yclients.com/ru> (accessed on 05.03.2022) (in Russian)
8. Universal Accounting System. [Electronic resource]. – URL: [usu.kz](http://usu.kz) (accessed on 05.03.2022).
9. Leonenkov, A. UML Language Notation and Semantics. [Electronic resource]. – URL: <https://intuit.ru/studies/courses/32/32/info> (accessed on 10.03.2022).
10. Java Documentation. Oracle. [Electronic resource]. – URL: <https://docs.oracle.com/en/java/> (accessed on 22.05.2023).
11. Spring Boot Reference Documentation. Spring. [Electronic resource]. – URL: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (accessed on 22.05.2023).
12. Spring Data JPA – Reference Documentation. Spring. [Electronic resource]. – URL: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> (accessed on 22.05.2023).
13. Spring Security. Spring. [Electronic resource]. – URL: <https://docs.spring.io/spring-security/reference/> (accessed on 22.05.2023).
14. Apache FreeMarker Manual. FreeMarker. [Electronic resource]. – URL: <https://freemarker.apache.org/docs/> (accessed on 22.05.2023).
15. Documentation. PostgreSQL. [Electronic resource]. – URL: <https://www.postgresql.org/docs/> (accessed on 22.05.2023).
16. Git. [Electronic resource]. – URL: <https://git-scm.com/> (accessed on 22.05.2023).
17. Project Lombok. [Electronic resource]. – URL: <https://projectlombok.org/> (accessed on 22.05.2023).
18. Bootstrap. [Electronic resource]. – URL: <https://getbootstrap.com/> (accessed on 22.05.2023).
19. Model View Controller (MVC) Design Pattern in Java. Java Guides. [Electronic resource]. – URL: <https://www.javaguides.net/2019/08/model-view-controller-mvc-design-in-java.html> (accessed on 22.05.2023).
20. A Guide to CSRF Protection in Spring Security. Baeldung. [Electronic resource]. – URL: <https://www.baeldung.com/spring-security-csrf> (accessed on 27.05.2023).
21. Bcrypt Step by Step. [Electronic resource]. – URL: <https://medium.com/bootdotdev/bcrypt-step-by-step-99dd76a3b16a> (accessed on 22.05.2023).

## **DESIGNING, DEVELOPING, AND DOCUMENTING AN INFORMATION SYSTEM FOR RECORDING STUDENTS' INDEPENDENT WORK TIME IN THE COMPUTER LAB**

*Vasilyev Igor D., Vasilyuk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, vasilevid1111@mail.ru

This article is devoted to the automation of students' work time accounting in a computer class. The paper describes the logic of the information system in the form of diagrams and data model. The theoretical basis for the development of an information system for the accounting of students' time is given. The analysis of analogues is carried out, the requirements for the developed system are identified. The process of choosing design tools and creating an information system model is described. The description of the used instrumental means of development is presented. This information system can be used by the employee of the computer class and students wishing to enroll in the computer class for independent work.

**Keywords:** timekeeping, student's independent work, computer class, information system, uml diagrams.

## ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ЧАТ-БОТА ДЛЯ СОЗДАНИЯ НАПОМИНАНИЙ

*Васильев Георгий Евгеньевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, egorlool@mail.ru

*Кузаев Айдар Файзуллович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kuzaev@pspu.ru

В данной статье описывается процесс проектирование и разработки чат-бота для создания напоминаний со сбором информации и последующим хранением её в базе данных. Определён предмет и объект исследования, сформулирована актуальность разработки чат-бота, поставлен ряд задач, которые необходимо решить в процессе проектирования и разработки. Цель работы заключается в создании автоматизированного программного обеспечения, которое способно взаимодействовать с пользователем посредством диалога и собирать необходимую для создания напоминания информацию для последующего использования. В рамках работы был спроектирован чат-бот, с помощью языка проектирования UML, и разработан с помощью языка программирования C# и библиотек Bot Framework, Entity Framework. Чат-бот оснащён таким функционалом как сбор пользовательской информации, создание записи в базе данных, вывод пользователю списка созданных напоминаний, а также изменение созданных напоминаний. В качестве базы данных для хранения информации была выбрана база данных и СУБД Microsoft SQL Server.

Ключевые слова: чат-бот, напоминания, приложение, база данных.

### **Введение**

В современном мире пунктуальность играет немаловажную роль в жизни и работе человека, но не всегда удаётся делать всё вовремя по ряду разных причин. Чтобы справиться с этой проблемой я решил разработать приложение чат-бот для напоминания.

Создание напоминаний включает в себя процесс взаимодействия пользователя и приложения, при котором пользователь создаёт напоминания, указывая следующую информацию:

- Название напоминания;
- Текст напоминания;
- Дату;
- Время;
- Повторяемость напоминания.

Некоторые из этих параметров могут быть опциональными.

Исходя из вышеперечисленных процессов мы можем определить функциональные требования. Приложение чат-бота должно:

- Собирать информацию от пользователя о напоминаниях;
- Сохранять информацию в базу данных;

- Считывать информацию из базы данных;
- Иметь функционал к изменению созданных напоминаний;
- Отправлять напоминания.

Цель работы – Проектирование и разработка чат-бота для создания напоминаний.

Задачи, которые необходимо решить в ходе проектирования и разработки чат-бота:

- 1) Получение теоретических результатов о актуальности темы и о существующих решениях;
- 2) Выбор языка программирования и среды программирования для проектирования и разработки;
- 3) Проектирование логики приложения;
- 4) Разработка приложения;
- 5) Создание и подключение к базе данных.

Объектом исследования является область информационных технологий.

Предметом исследования является приложения для персональных компьютеров для создания напоминаний.

Разобраны приложения-аналоги для создания напоминаний, такие как

- Todoist – мультиплатформенное приложение для создания напоминаний с широким спектром возможностей включая: повторяющиеся сроки напоминаний, разделы и подзадачи, возможность использовать шаблоны, уровни приоритета, избранное. Поставляется на бесплатной основе, но можно купить платную версию с расширенным функционалом;
- Microsoft To Do – приложения для ведения списка дел. Из особенностей можно выделить простоту, списки дел реализованы в виде чек-листов. Также оно интегрировано с другими продуктами Microsoft, например Outlook;
- Any.do – ещё одно популярное приложение для создания напоминаний. В целом похожее на предыдущие варианты, но не имеющее приоритетности задач, не мультиплатформенное.

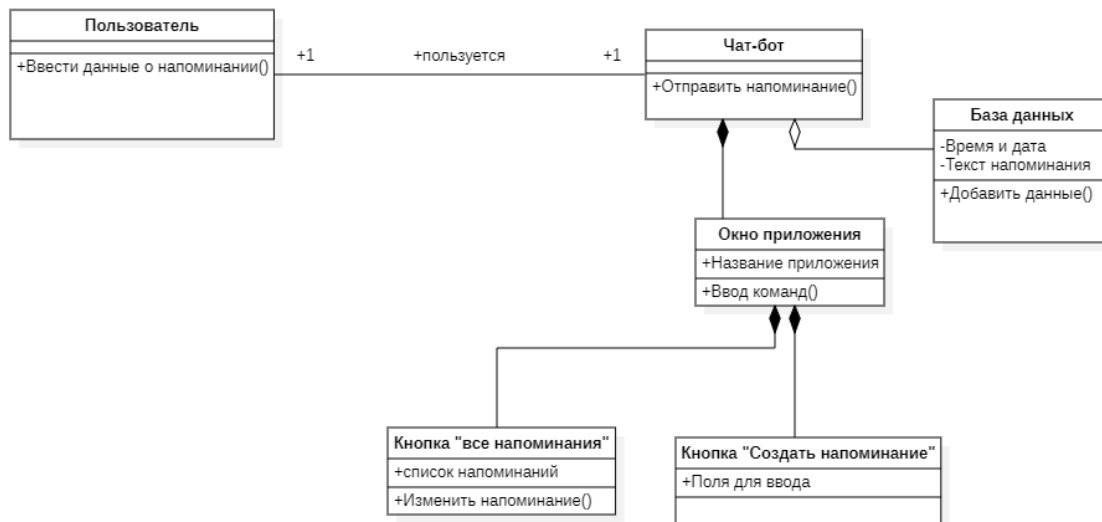
Рассмотрев данные аналоги, был сделан вывод, что разрабатываемое приложение обладает интерактивностью взаимодействия с пользователем по средствам диалога.

### **Проектирование приложения чат-бота**

Все диаграммы были созданы с помощью приложения StarUML [1].

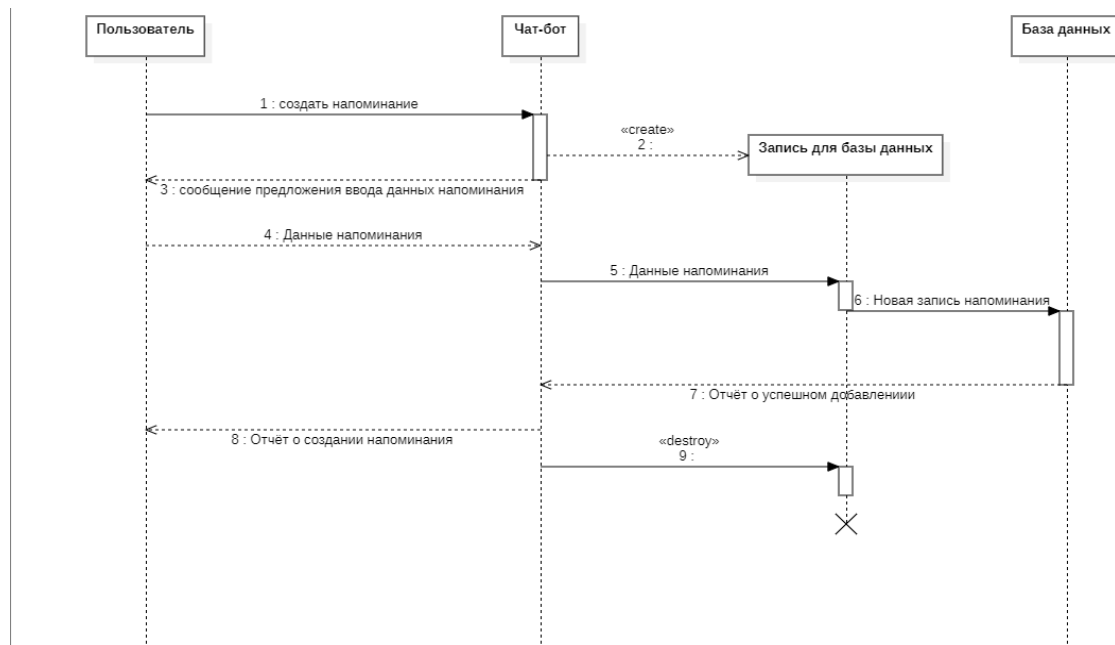
Для описания функционала чат-бота была выбрана диаграмма классов (см. рис. 1) и последовательностей (см. рис. 2, 3) для более полного описания работы чат-бота.

На диаграмме классов определены такие классы как чат-бот, пользователь, база данных, окно приложения и кнопки, «Все напоминания» и «Создать напоминание». Две кнопки связаны с окном приложения отношением композиции, так как при отсутствии окна не будет и кнопок, аналогично с окном приложения и чат-ботом.



**Рис. 1. Диаграмма классов**

Данная диаграмма последовательностей описывает динамику взаимодействия объектов. На данной диаграмме представлен процесс создание напоминания. Пользователь вводит команду создания напоминания, чат-бот создаёт запись для базы данных, пользователь вводит данные напоминания, эти данные добавляются в запись для БД, и происходит добавление её в БД, пользователь уведомляется о создании.



**Рис. 2. Диаграмма последовательностей для создания напоминания**

На следующей диаграмме последовательностей представлен процесс просмотра и изменения напоминаний. Пользователь вводит команду для просмотра всех напоминаний. Чат-бот обращается к БД с запросом списка всех напоминаний и выводит его пользователю. Пользователь может изменить напоминание.

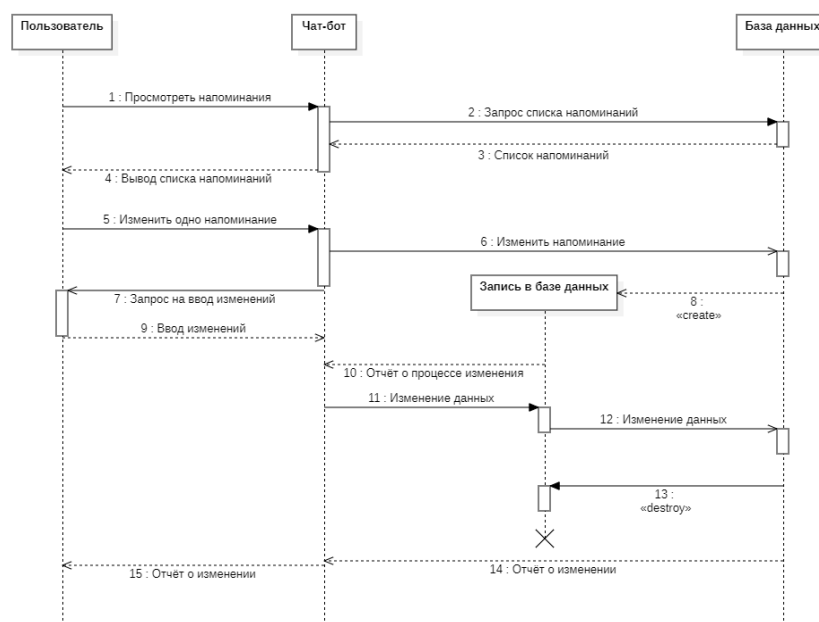


Рис. 3. Диаграмма последовательностей для просмотра и изменения напоминаний

### Разработка приложения чат-бота

Разработка приложения чат-бота состоит из следующих этапов:

- Разработка логики работы чат-бота;
- Разработка пользовательских действий;
- Подключить базу данных для сохранения и вывода информации о напоминаниях.

Для разработки приложения чат-бота была использована библиотека Bot Framework на языке C# [2], а также для разработки логики чат-бота было использовано приложение Bot Framework Composer [3]. Оно позволяет выстроить последовательность команд для чат-бота в виде блок-схемы с такими блоками как:

- Вывод сообщения пользователю;
- Запрос на ввод числа, текста, даты, времени;
- Логические ветвления if/else и switch;
- Циклы;
- Адаптивные карточки;
- Занесение в память и использование значений переменных;
- Обращение к внешним ресурсам, таким как http запросы.

Отдельное внимание уделено адаптивным карточкам. Бывают случаи, когда вам нужны сообщения, состоящие только из простого текста, а бывают случаи, когда вам нужно более богатое содержимое сообщений, например изображения, анимированные GIF-файлы, видеоклипы, аудиоклипы и кнопки. Адаптивные карты создаются самим разработчиком, и вследствие этого можно более гибко собирать данные от пользователя.

Пользовательские действия создаются с помощью библиотеки Bot Framework Custom Action [4]. Они позволяют расширить функционал чат-бота, созданного с помощью Bot Framework Composer. Например, обращение к базе данных Microsoft SQL Server [5], добавление записей в неё, считывание данных.

Основная логика приложения строится на использовании диалогов – отдельных этапов выполнения программы чат-бота. Были созданы диалоги для обработки информации о том, что пользователь хочет сделать, создать напоминание или просмотреть их список, для создания напоминания, для вывода информации о созданных напоминаниях, для изменения напо-



минания. Также стоит упомянуть, что весь диалог с чат-ботом цикличен, то есть после завершения создания, или просмотра и изменения напоминаний чат-бот продолжает работу с момента выбора пользователем действия по созданию или просмотру напоминаний.

### **Заключение**

В результате выполнения работы выполнены следующие задачи:

- Получены теоретические результаты о актуальности темы и о существующих решениях;
- Выбран язык программирования и среда программирования для проектирования и разработки;
- Спроектирована логика приложения;
- Разработано приложение;
- Создана база данных и выполнено подключение ней.

Разработанное приложение позволяет создавать напоминания, информация о которых заносится в базу данных, считывать и выводить пользователю информацию о созданных напоминаниях.

### **Библиографический список**

1. Среда проектирования StarUML // StarUML [Электронный ресурс]. – URL: <https://staruml.io> (дата обращения 29.03.2023)
2. Bot Framework Documentation // Microsoft [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-4.0> (дата обращения 10.04.2023)
3. Bot Framework Composer Documentation // Microsoft [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/composer/> (дата обращения 14.04.2023)
4. Bot Framework Custom Action // Custom Action [Электронный ресурс]. – URL: <https://learn.microsoft.com/ru-ru/composer/how-to-create-custom-actions> (дата обращения 10.04.2023)
5. Microsoft SQL Server Documentation // Microsoft [Электронный ресурс]. – URL: <https://www.microsoft.com/en-in/sql-server> (дата обращения 10.05.2023)

### **References**

1. StarUML Design Environment // StarUML [Electronic resource]. – URL: <https://staruml.io> (accessed 29.03.2023)
2. Bot Framework Documentation // Microsoft [Electronic resource]. – URL: <https://learn.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-4.0> (accessed 10.04.2023)
3. Bot Framework Composer Documentation // Microsoft [Electronic resource]. – URL: <https://learn.microsoft.com/en-us/composer/> (accessed 14.04.2023)
4. Bot Framework Custom Action // Custom Action [Electronic resource]. – URL: <https://learn.microsoft.com/ru-ru/composer/how-to-create-custom-actions> (accessed 10.04.2023)
5. Microsoft SQL Server Documentation // Microsoft [Electronic resource]. – URL: <https://www.microsoft.com/en-in/sql-server> (accessed 10.05.2023)

## DESIGN AND DEVELOPMENT OF A CHATBOT CREATING REMINDERS

*Vasiliev Geogrii E.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, egorlool@mail.ru

*Kuzaev Aidar F.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, kuzaev@pspu.ru

This article describes the process of designing and developing a chatbot for creating reminders with collecting information and then storing it in a database. The subject and object of research are defined, the relevance of the chatbot development is formulated, a number of tasks that need to be solved in the design and development process are set. The purpose of the work is to create automated software that is able to interact with the user through a dialog and collect the information necessary to create a reminder for subsequent use. As part of the work, a chatbot was designed using the UML design language, and developed using the C# programming language and the Bot Framework libraries, Entity Framework libraries. The chatbot is equipped with such functionality as collecting user information, creating an entry in the database, displaying a list of created reminders to the user, as well as changing created reminders. Microsoft SQL Server database was chosen as the database for storing information.

Keywords: chatbot, reminders, application, database.

## ТРЕБОВАНИЯ К РАСПИСАНИЮ В СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ ПРИ АВТОМАТИЗАЦИИ ЕГО СОСТАВЛЕНИЯ

*Вахрушев Геннадий Сергеевич, Кушев Вадим Олегович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, gena.vaxrushev@gmail.com

Производится анализ литературных источников, рассматривающих какие-либо подходы к составлению расписания учебных занятий, учитывая применимость предложенного метода к школьному расписанию, классификацию требований и способ учёта требований. На основании анализа сформирован список требований к расписанию в средней общеобразовательной школе (СОШ). Для выявления трудностей при составлении расписания проведён опрос составителей расписания в СОШ о влиянии выявленных ранее требований, проведён анализ полученных результатов тестирования. На основании анализа литературы и опроса специалистов по составлению расписания сформирована, приведена и описана классификация требований к расписанию, приведены примеры к каждому типу требований. Выявленные ранее требования распределены по типам, определённым в полученной классификации. Произведена интерпретация классификации по отношению к программе составления расписания.

Ключевые слова: расписание, автоматизация, анализ, классификация, требования.

Важно уметь качественно и быстро составлять расписание, потому что от него зависит качество образовательного процесса. Хорошо составленное расписание позволяет рационально использовать время участников образовательного процесса, улучшает усваиваемость материала и т.д.

Составление расписания является трудоёмкой задачей, на которую уходит много времени и сил у его составителя, ведь надо учитывать множество постоянных факторов и факторов, возникающих случайно, другими словами инвариантных и вариативных факторов.

Следствием сложности составления расписания являются большие временные затраты. Специалист занят составлением большой промежуток времени, из-за чего он часто не может выполнять другие свои обязанности в полном объёме, что ведёт к потенциальным затратам на их позднее решение, либо решение другим специалистом, который может выполнять данные ему поручения хуже, чем основной специалист. Помимо этого, время, потраченное на составление, необходимо оплачивать, что ведёт к явным финансовым затратам.

Не менее важной является возможность изменения составленного расписания. При изменении вариативных факторов необходимо иметь возможность подстроить готовое расписание в соответствии с произошедшими изменениями, причём желательно иметь возможность как ручного, так и автоматизированного изменения.

Важно заметить, что при составлении и изменении расписания оно должно оставаться корректным, то есть непротиворечивым, что значит соответствовать следующим условиям:

1. Занятия с одной и той же группой учеников не должны проводиться в одно и то же время.
2. Занятия с одним и тем же учителем не должны проводиться в одно и то же время.

3. Занятия в одной и той же аудитории в одно и то же время не должны проводиться несколькими группами или преподавателями.

Автоматизированное составление расписания с учетом предъявляемых к этому процессу требований позволило бы сократить время его составления, а также дало бы возможность быстрее подстраиваться под возможные изменения в учебном заведении в процессе обучения. Для этого потребуются определить и классифицировать как сами требования, так и наиболее приемлемые для автоматизации алгоритмы составления расписания.

Для того, чтобы определиться с тем какие на сегодняшний день существуют алгоритмы составления расписания и каким образом они учитывают те реалии, которые происходят в СОШ, необходимо провести анализ подходов, методов и методологий по составлению расписаний. Был проведён анализ источников, который приведён ниже.

Основная информация по теории расписаний описана в учебном пособии Лазарева А.А. и Гафарова Е.Р. «Теория расписаний. Задачи и алгоритмы» [1]. В ней подробно расписаны типы задач теории расписаний, методы их решения с примерами, а также определения, формулы и т.п. В данном источнике не описано применение какого-либо конкретного алгоритма для составления школьного и расписания, поэтому служит источником для общего понимания темы.

В статье Сорокиной О.П. и О.А. Медведевой «Автоматизация процесса составления расписания» [2] кратко описаны 6 основных подходов для составления расписания и способы их применения. На основе этой статьи необходимо изучить данные алгоритмы и определить наиболее подходящие для составления школьного расписания.

В статье «Композиционный генетический алгоритм составления расписания учебных занятий» [3], авторами которой являются: Кабальнов Ю.С., Шехтман Л.И., Низимова Г.Ф., Земченокова Н.А., описан генетический алгоритм составления расписания. Алгоритм описан с использованием математических обозначений и описывает как сам генетический алгоритм, так и способ задания объектов алгоритма, а также их структуру. Требования задаются в виде формул, обозначающих штрафы за их несоблюдение. Целью алгоритма является минимизация суммы штрафов.

Требования к расписанию разбиваются на две группы: обязательные и желательные. К первой группе относятся требования, невыполнение которых делает невозможным осуществление образовательного процесса, например, отсутствие накладок на учебные группы, а вторая группа включает в себя желательные, но необязательные требования, соблюдение которых является дополнительным плюсом, например, учёт пожеланий преподавателя относительно своего расписания занятий.

В данной статье алгоритм используется для составления расписания в ВУЗе, поэтому он не подходит для СОШ в полной мере. К примеру, в ВУЗе происходит разбиение на практические и лекционные занятия, которые могут проводиться одновременно с группами разных направлений, в то время как в школе один учитель проводит урок с одним классом. Поэтому предложенные задания требований требуют корректировок.

В статье Т.В. Сиркина, А.П. Чернышовой, П.А. Мартынова, А.Д. Морозова «Разработка автоматизированной системы составления и оптимизации расписания занятий» [4] также описан генетический алгоритм, но в более простой форме и меньшим количеством математических обозначений по сравнению с предыдущей статьёй. В данной статье алгоритм применяется уже к школьному расписанию. Отсутствует разбиение требований на какие-либо группы. Стоит заметить более понятные и наглядные схемы и изображения, представленные на рисунках 1 и 2.

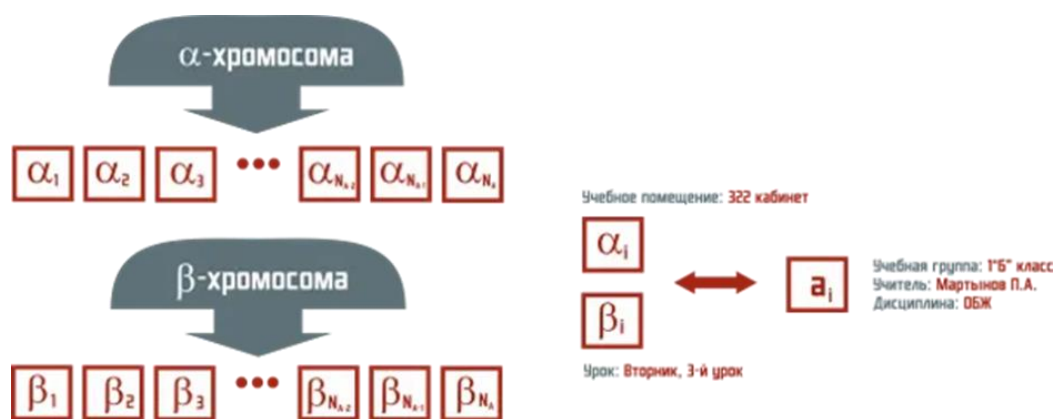


Рис. 1. Представление особи в генетическом алгоритме

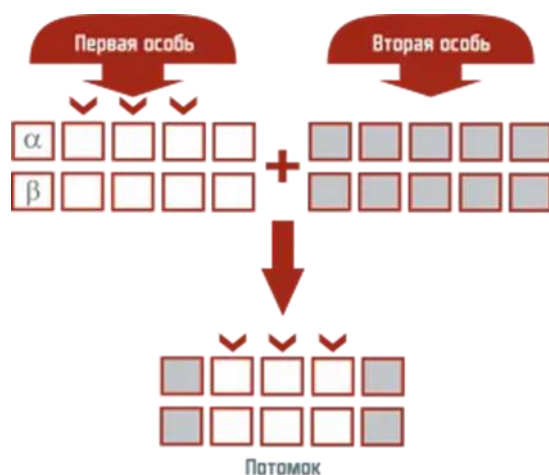


Рис. 2. Скрещивание особей

В работе Балтака С.В., Сотскова, Ю.Н. «Построение расписаний учебных занятий на основе раскраски вершин графа» [5] описан алгоритм составления расписания при помощи составления графа и раскраски его вершин. Описан как сам алгоритм, так значения вершин, рёбер, цветов и т.п. В работе также используется генетический алгоритм, но не как основной, а лишь для распределения занятий по аудиториям. Отсутствует разбиение требований на какие-либо группы, однако указано, какие условия необходимо выполнить, а какие следует предусмотреть, что аналогично разбиению из статьи [3]. Также следует заметить, что расписание составляется для школы.

В статье Бабкиной Т.С. «Задача составления расписаний: решение на основе многоагентного подхода» [6] рассматривается составление расписания на основе многоагентного подхода. Описываются введённые обозначения при помощи математических формул, описаны плюсы подхода, типы алгоритмов для решения задачи, классификации требований нет.

В электронном ресурсе «Решение задач целочисленного программирования: методы и примеры» [7] рассмотрены методы решения задач целочисленного программирования, к которой может быть сведена задача составления расписания.

В электронных ресурсах «Муравьиные алгоритмы» [8] и «Метод муравьиной колонии (Ant colony optimization)» [9] рассмотрен метод муравьиной колонии, его смысл, алгоритм, принцип и некоторые задачи, при которых он применяется.

В статье Лопатина А.С. «Метод отжига» [10] и электронном ресурсе «Алгоритм имитации отжига» [11] рассмотрен метод имитации отжига, его алгоритмы, принцип работы,

В результате анализа литературы выявлены подходы к составлению расписания, классификации требований, способы их задания и соблюдения.

На основе анализа литературы составлен список требований к расписанию, который представлен ниже:

- 1) запрет назначения нескольких занятий в одно время для одной учебной группы;
- 2) запрет назначения нескольких занятий в одно время в одном кабинете;
- 3) запрет назначения нескольких занятий в одно время для одного учителя;
- 4) отсутствие «окон» для учеников;
- 5) ограничение на количество занятий в день для учебной группы;
- 6) ограничение на количество занятий в день для учителя;
- 7) соответствие типа класса виду занятия;
- 8) соответствие количества занятий учебному плану;
- 9) запрет назначения занятий для учебной группы на тот или иной интервал времени;
- 10) запрет назначения занятий для учителя на тот или иной интервал времени;
- 11) возможность проведения спаренных уроков;
- 12) ограничение на количество занятий по одному предмету в день.

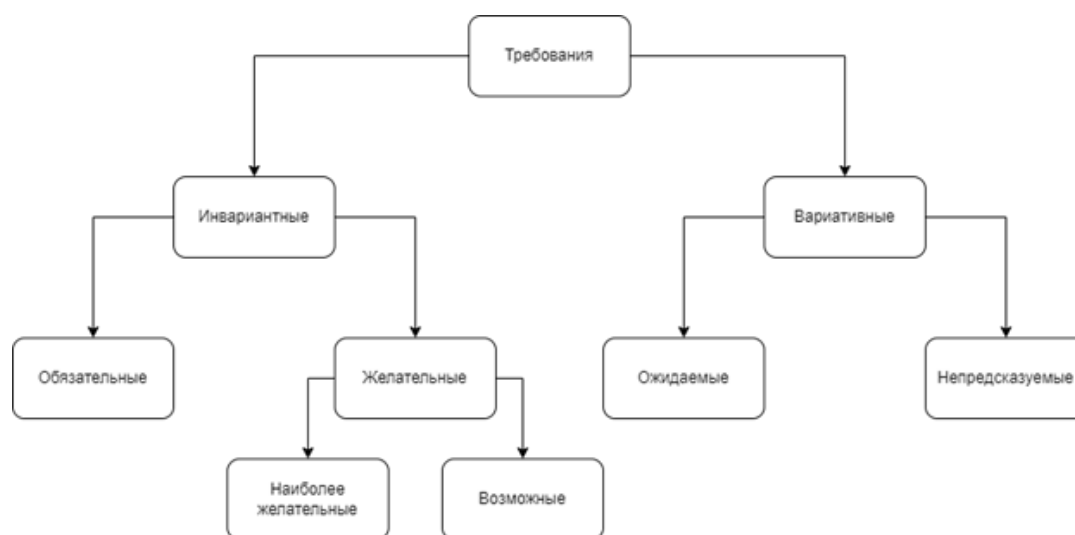
Для выявления трудностей какой-либо предметной области эффективным подходом является опрос специалистов этой предметной области. Для выявления трудностей при составлении расписания и проверки правильности определения требований был проведён опрос составителей расписания в различных СОШ, включающий следующие пункты:

1. Оценка важности описанных выше требований по шкале от 1 до 5.
2. Какие из описанных выше требований должны соблюдаться обязательно.
3. Как часто встречаются проблемы с описанными выше требованиями по шкале от 1 до 5.
4. Классификация требований (при наличии).
5. Трудности при составлении расписания.
6. Пожелания для программы составления расписания.

Опрос показал, что:

1. Все описанные выше требования важны.
2. Обязательными должны быть требования: 1, 2, 3, 4, 8, 12.
3. Чаще всего возникают проблемы с требованиями: 1, 2, 3.
4. Классификация требований не производится.
5. При составлении возникают следующие трудности:
  - необходимо пересчитывать недельную нагрузку по классам;
  - необходимо пересчитывать расписание по трудности предметов, после чего вносить изменения;
  - необходимо следить за тем, чтобы уроки одной параллели у учителей-предметников были в один день;
  - необходимо следить за нагрузкой учителя, который ведет несколько предметов.
6. В программе помимо выделенных функций специалистам хотелось бы иметь возможность учитывать трудность предметов и распределение недельной нагрузки по СанПиН.

Исходя из опроса и анализа литературы была сформирована классификация требований, представленная на рис. 3.



**Рис. 3. Классификация требований**

В приведённой классификации требования делятся на инвариантные, то есть те, которые не меняются со временем и присутствуют всегда, а также вариативные, которые проявляются время от времени, которые делятся на ожидаемые, то есть те, которые очевидно случаются время от времени, а также на непредсказуемые, появление которых крайне маловероятно. Инвариантные в свою очередь делятся на обязательные требования, несоблюдение которых делает невозможным проведение занятий, а также на желательные требования, соблюдение которых улучшит качество расписания, которые в свою очередь делятся на наиболее желательные, то есть те, которые значительно улучшат расписания, и возможные, которые улучшат расписание в меньшей степени.

Примеры требований на каждый тип могут быть следующими:

- обязательное – запрет назначения нескольких занятий в одно время для одной учебной группы;
- наиболее желательное – отсутствие окон у учеников;
- возможное – ограничение на количество занятий в день для учителя;
- ожидаемое – возможность заменить учителя при его заболевании;
- непредсказуемое – землетрясение.

Используя полученную классификацию можно распределить выявленные требования:

**Обязательные:**

- запрет назначения нескольких занятий в одно время для одной учебной группы;
- запрет назначения нескольких занятий в одно время в одном кабинете;
- запрет назначения нескольких занятий в одно время для одного учителя;
- соответствие количества занятий учебному плану.

В системе наиболее желательно реализовать учёт следующих требований:

- отсутствие окно для учеников;
- ограничение на количество занятий по одному предмету в день;
- соответствие расписания требованиям СанПиН;
- проведение уроков одной параллели у учителей-предметников в один день.

В системе желательно реализовать учёт следующих требований:

- ограничение на количество занятий в день для учебной группы;
- ограничение на количество занятий в день для учителя;

- соответствие типа класса виду занятия;
- запрет назначения занятий для учебной группы на тот или иной интервал времени;
- запрет назначения занятий для учителя на тот или иной интервал времени;
- возможность проведения несколько занятий подряд.

Вариативные требования отсутствуют в приведённых выше требованиях.

Разработанная классификация требований позволит обозначить место и важность каждой функции в программе составления расписания: учёт обязательных требований является необходимым, учёт наиболее желательных требований значительно увеличит качество программы, соблюдение вариативных и возможных требований будет дополнительным, но не наиболее сильным преимуществом.

Таким образом, были изучены методы составления расписания, выявлены требования к расписанию, проведён опрос специалистов о влиянии этих требований, на основании анализа литературы и опроса разработана классификация выявленных требований.

### Библиографический список

1. Лазарев А.А., Гафаров Е.Р. «Теория расписаний. Задачи и алгоритмы». URL: <https://www.ipu.ru/sites/default/files/publications/12896/406-12896.pdf>
2. О.П. Сорокина, О.А. Медведева «Автоматизация процесса составления расписания» URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-protssessa-sostavleniya-raspisaniya-zanyatiy-1/viewer>
3. Кабальнов Ю.С., Шехтман Л.И., Низимова Г.Ф., Земченокова Н.А. «Композиционный генетический алгоритм составления расписания учебных занятий». URL: <https://cyberleninka.ru/article/n/kompozitsionnyy-geneticheskiy-algoritm-sostavleniya-raspisaniya-uchebnyh-zanyatiy/viewer>
4. Т.В. Сиркин, А.П. Чернышова, П.А. Мартынов, А.Д. Морозов «Разработка автоматизированной системы составления и оптимизации расписания занятий». URL: <https://moluch.ru/archive/317/72336/>
5. Балтак С.В., Сотсков, Ю.Н. «Построение расписаний учебных занятий на основе раскраски вершин графа» // Информатика. – 2006. – С. 58–69.
6. Т.С. Бабкина «Задача составления расписаний: решение на основе многоагентного подхода». URL: <https://cyberleninka.ru/article/n/zadacha-sostavleniya-raspisaniy-reshenie-na-osnove-mnogoagentnogo-podhoda/viewer>
7. Решение задач целочисленного программирования: методы и примеры [Электронный ресурс]. – URL: [https://function-x.ru/zadacha\\_celochislennogo\\_programmirovaniya.html](https://function-x.ru/zadacha_celochislennogo_programmirovaniya.html) (дата обращения: 13.12.2023).
8. Муравьиные алгоритмы [Электронный ресурс]. – URL: [http://www.machinelearning.ru/wiki/index.php?title=Муравьиные\\_алгоритмы](http://www.machinelearning.ru/wiki/index.php?title=Муравьиные_алгоритмы) (дата обращения: 13.12.2023).
9. Метод муравьиной колонии (Ant colony optimization) [Электронный ресурс]. – URL: <https://wiki.loginom.ru/articles/ant-colony-optimization.html> (дата обращения: 13.12.2023).
10. Лопатин А.С. «Метод отжига». URL: <https://www.math.spbu.ru/user/gran/sb1/lopatin.pdf>
11. Алгоритм имитации отжига [Электронный ресурс]. – URL: [http://old.math.nsc.ru/AP/benchmarks/UFLP/uflp\\_sa.html](http://old.math.nsc.ru/AP/benchmarks/UFLP/uflp_sa.html) (дата обращения: 14.12.2023).



## References

1. Lazarev A.A., Gafarov E.R. «Teorija raspisanij. Zadachi i algoritmy». URL: <https://www.ipu.ru/sites/default/files/publications/12896/406-12896.pdf> (In Russ.).
2. O.P. Sorokina, O.A. Medvedeva «Avtomatizacija processa sostavlenija raspisanija» URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-protssesa-sostavleniya-raspisaniya-zanyatiy-1/viewer>(In Russ.).
3. Kabal'nov Ju.S., Shehtman L.I., Nizimova G.F., Zemchenokova N.A. «Kompozicionnyj geneticheskij algoritm sostavlenija raspisanija uchebnyh zanjatij». URL: <https://cyberleninka.ru/article/n/kompozitsionnyy-geneticheskij-algoritm-sostavleniya-raspisaniya-uchebnyh-zanyatiy/viewer>(In Russ.).
4. T.V. Sirkin, A.P. Chernyshova, P.A. Martynov, A.D. Morozov «Razrabotka avtomatizirovannoj sistemy sostavlenija i optimizacii raspisanija zanjatij». URL: <https://moluch.ru/archive/317/72336/>(In Russ.).
5. Baltak S.V., Sotskov, Ju.N. «Postroenie raspisanij uchebnyh zanjatij na osnove raskraski vershin grafa» // Informatika. – 2006. – S. 58–69. (In Russ.).
6. T.S. Babkina «Zadacha sostavlenija raspisanij: reshenie na osnove mnogoagentnogo podhoda». URL: <https://cyberleninka.ru/article/n/zadacha-sostavleniya-raspisaniy-reshenie-na-osnove-mnogoagentnogo-podhoda/viewer>(In Russ.).
7. Reshenie zadach celochislennogo programmirovaniya: metody i primery [jelektronnyj resurs] – URL: [https://function-x.ru/zadacha\\_celochislennogo\\_programmirovaniya.html](https://function-x.ru/zadacha_celochislennogo_programmirovaniya.html) (In Russ.).
8. Murav'inye algoritmy [Electronic resource]. – URL: [http://www.machinelearning.ru/wiki/index.php?title=Муравьиные\\_алгоритмы](http://www.machinelearning.ru/wiki/index.php?title=Муравьиные_алгоритмы) (In Russ.).
9. Metod murav'inoj kolonii (Ant colony optimization) [Electronic resource]. – URL: <https://wiki.loginom.ru/articles/ant-colony-optimization.html> (In Russ.).
10. Lopatin A.S. «Metod otzhiga». URL: <https://www.math.spbu.ru/user/gran/sb1/lopatin.pdf>
11. Algoritm imitacii otzhiga [Electronic resource]. – URL: [http://old.math.nsc.ru/AP/benchmarks/UFLP/uflp\\_sa.html](http://old.math.nsc.ru/AP/benchmarks/UFLP/uflp_sa.html) (In Russ.).

## REQUIREMENTS FOR THE SCHEDULE IN THE SECONDARY SCHOOL WHEN AUTOMATING ITS COMPILATION

*Vakhrushev Gennadiy S., Kushev Vadim O.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, gena.vaxrushev@gmail.com

The analysis of literature considering any approaches to the preparation of the timetable of classes is carried out, taking into account the applicability of the proposed method to the secondary school timetable, the classification of requirements and the method of accounting for requirements. Based on the analysis, a list of requirements for the timetable in the secondary school was formed. In order to identify difficulties in scheduling, a survey of schedulers in the secondary schools was conducted on the impact of previously identified requirements, an analysis of the test results was carried out. Based on the analysis of the literature and a survey of specialists in scheduling, a classification of timetable requirements is formed, presented and described, examples for each type of requirements are given. The requirements identified earlier are distributed according to the types defined in the resulting classification. The interpretation of the classification in relation to the scheduling software program is made.

Keywords: timetable, automatization, analysis, classification, requirements.

## КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СЛУЧАЙНОЙ ПЛОТНОЙ УПАКОВКИ НЕСЖИМАЕМЫХ ОКРУЖНОСТЕЙ НА ПЛОСКОСТИ

*Вдовин Иван Евгеньевич, Бузмакова Мария Михайловна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, guzez212@gmail.com

В данной работе рассматривается задача максимально плотной укладки окружностей на ограниченной плоскости. Данная задача остаётся актуальной на сегодняшний день, так как её решение может быть применено во множестве практических сфер, включая дизайн интерьеров, оптимизацию расположения оборудования, размещение сенсорных элементов и т.д. В процессе исследования были рассмотрены различные методы решения задачи максимальной плотной укладки окружностей. Были проведены сравнительные анализы эффективности различных алгоритмов и предложены новые подходы к решению этой задачи. Одним из ключевых результатов работы является определение оптимальной зависимости порогового значения от размеров плоскости. Это уравнение позволит находить оптимальное количество окружностей, которые могут быть уложены на данной плоскости.

Ключевые слова: джемминг, окружности, компьютерное и математическое моделирование, теория перколяции

Ситуацию случайной максимально плотной упаковки объектами некоторой рассматриваемой системы называют джеммингом (от англ. jamming, [1-4]). Примером может служить парковка автомобилей: место вроде бы есть, но его недостаточно для размещения очередной машины. Значение максимально достижимой упаковки называют порогом джемминга. Модели максимально плотной случайной упаковки окружностей являются востребованным инструментом анализа и исследования во многих отраслях науки.

Ещё в 18 веке учёные занимались задачами плотной упаковки. Так в 1773 году Джозеф Луи Лаграндж изучал максимально плотную упаковку на решётках. В своих исследованиях Лаграндж использовал различные формы решёток. Он доказал [5], что наибольшая концентрация укладки окружностей достигается на гексагональных решётках, а значение концентрации равно  $\phi_{\max} = \pi / (2 \sqrt{3}) \approx 0,9069$ . Укладка на гексагональной решётке означает, что центры фигур, а именно окружностей, укладываются на узлы, образованные правильными шестиугольниками.

Форма окружности обладает низкой концентрацией укладки. Например, такие фигуры, как квадрат, ромб, гексагон можно уложить таким образом, что концентрация упаковки будет равна  $\phi = 1$ . Стоит отметить, что значение, полученное в [6] теоретически является максимальным, как для решёток, так и для континуальных задач. Однако, стоит отметить, что нет единого алгоритма или метода случайной укладки окружностей в континуальной плоскости, значение концентрации которого достигало максимально возможного значения. Зарубежные учёные [7] в своей работе получили число концентрации  $\phi = 0,826$ , используя упорядоченную модель. Упорядоченные модели позволяют достичь максимальной концентрации, но окружности укладываются не случайным, а на основе некоторых правил. Поэтому слу-

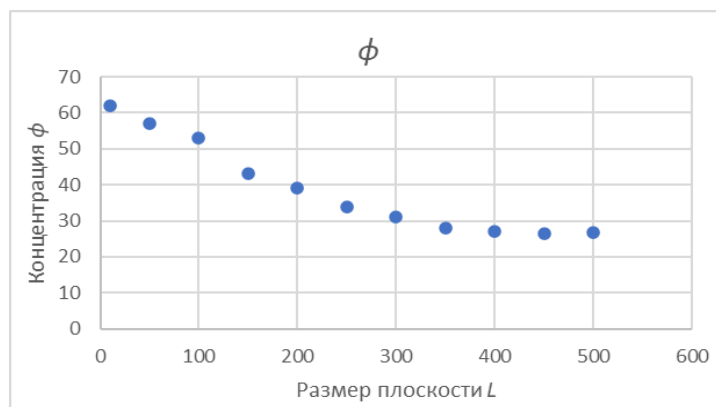
чайные модели имеют ряд существенных преимуществ, в сравнении с упорядоченными моделями. Случайные модели можно использовать при изучении природных явлений, так как упорядоченные модели не дают результат, приближенный к реальным явлениям, но при этом концентрация в случайных моделях намного ниже. Канадские учёные в исследовании случайных упаковок [8] получили значение концентрации  $\phi = 0,60 \pm 0,02$ . Однако, другие учёные утверждают, что значение концентрации иное. Берримен [9] выяснил, что плотность упаковки составляет  $\phi = 0,64 \pm 0,02$ . Однако и это не наилучший результат. Проанализировав результаты некоторых авторов [6, 7, 9-10] можно сделать вывод, что концентрация случайно упакованных одинаковых окружностей на плоскости лежит в интервале  $[0.58, 0.67]$ .

В данной работе авторами предложена математическая модель неупорядоченной случайной укладки несжимаемых окружностей на плоскости с применением подходов теории перколяции, учитывающая возможность рассмотрения объектов разного размера. Научная новизна настоящего исследования обусловлена тем, что предложена новая модель максимально плотной упаковки окружностей на плоскости. Модель отличается от существующих тем, что используется иной критерий останковки алгоритма, зависящий от размеров области. Для предложенной модели получены новые и оригинальные результаты.

Предложенная модель джемминга окружностей на плоскости может быть описана следующим образом. Окружности с радиусом  $r$  случайным образом размещаются на ограниченном участке плоскости – квадрате с линейным размером  $L$ . Окружности не могут пересекаться. При моделировании используются периодические граничные условия. В рамках модели рассмотрены два варианта для значения радиуса: одинаковый для всех окружностей и меняющийся в пределах заданного промежутка. Также вводится в рассмотрение критерий останковки упаковки. Критерий останковки – пороговое количество неуспешных итераций алгоритма подряд. Значение критерия останковки может быть абсолютно любым и задаётся исходя из цели исследования. В некоторых ситуациях [3, 4] удобно постоянное пороговое значение  $\mu = a$ , где  $a = \text{const}$ . При этом плотность упаковки для одинаковых  $\mu$  зависит от размеров плоскости. Если взять слишком малое пороговое значение, то концентрация будет слишком маленькой.

Порогом джемминга в рамках модели является максимально достигнутая концентрация окружностей в квадрате. Концентрация определяется, как отношение площади, занимаемой окружностями, к площади квадрата.

Для реализации модели был разработан алгоритм упаковки окружностей на плоскости, основанный на модифицированном методе сдвигов [5]. Был проведен ряд численных экспериментов с использованием подходов Монте-Карло и применением теории перколяции, математической статистики и теории вероятностей. Входные параметры первого этапа моделирования:  $L = \{10, 25, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ ,  $\mu = 100$ ,  $r = 0,05L$ , количество испытаний  $= L * L$ . Результаты представлены на рис. 1.



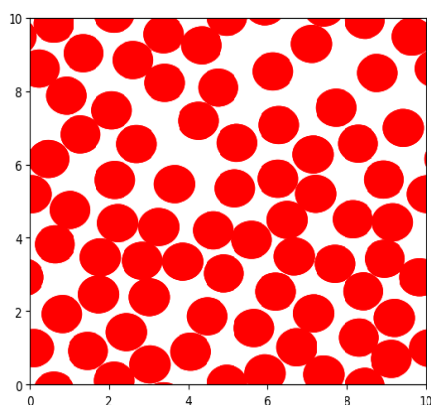
**Рис. 1. График зависимости концентрации от размера плоскости при постоянном пороговом значении критерия остановки**

Далее было проведено моделирование неупорядоченной случайной упаковки окружностей для параметров:  $L = 10, 25, 50, 100$ ;  $\mu = L^2$ ;  $r = 0,05L$ , количество испытаний = 100. На основе полученных результатов найдено среднее значение концентрации для каждой серии испытаний с учетом стандартного отклонения среднего. Данное значение принимается за оценку порога джемминга для конечной системы. Результаты представлены в табл. 1.

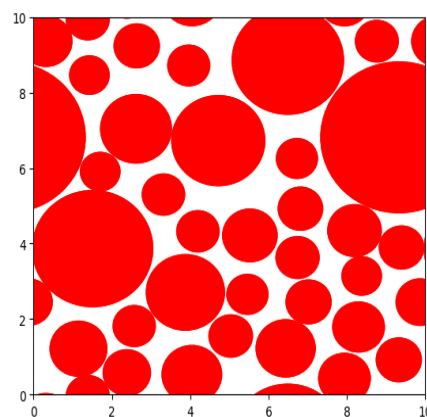
**Таблица 1. Данные порога джемминга для  $L$  конечного размера**

	$L = 10$	$L = 25$	$L = 50$	$L = 100$
$\varphi_j$	$0,573 \pm 0,0284$	$0,584 \pm 0,027$	$0,607 \pm 0,025$	$0,629 \pm 0,026$

На рис. 2, 3 показан результат работы программы для окружностей одного радиуса и разных радиусов соответственно.



**Рис. 2. Пример максимально плотной упаковки окружностей равного радиуса**



**Рис. 3. Пример максимально плотной упаковки окружностей разного радиуса**

Видно, что полученные значения порога джемминга совпадают с известными [9, 10], полученными другими исследователями, что подтверждает адекватность предложенной модели.

Авторами планируется дальнейшее исследование предложенной модели с другими параметрами и выявление зависимости порога джемминга от радиуса окружности, от диапазона радиусов окружностей и других параметров.

### Библиографический список

1. *Elizabeth Lawson-Keister*, Jamming and arrest of cell motion in biological tissues / Elizabeth Lawson-Keister, M. Lisa Manning // «Current Opinion in Cell Biology» vol. 72. P. 146-155.
2. *Bruce DeBruhl*, Digital Filter Design for Jamming Mitigation in 802.15.4 Communication / Bruce DeBruhl, Patrick Tague // Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN). 2011.
3. *Aleksandar Donev*, Jamming in hard sphere and disk packings / Aleksandar Donev // Journal of Applied Physics. 2004. Vol. 5. P. 989.
4. Mehta, Granular Matter / Mehta // Springer New York, 1994.
5. *Misaki Ozawa*, Jamming Transition and Inherent Structures of Hard Spheres and Disks / Misaki Ozawa, Takeshi Kuroiwa, Atsushi Ikeda, Kunimasa Miyazaki // Physical review Letters. 2011. P. 109.
6. *Shoujiang Yu*, Traffic Jams in Urban Areas – A Case Study in Gourd Island City of China / Shoujiang Yu, Guiping Xiao, Lei Nie, // Applications of Advanced Technologies in Transportation; Boston Marriot, Cambridge, Massachusetts, United States, 2002.
7. *Chang*, A Simple Proof of Thue's Theorem on Circle Packing / Chang, Hai-Chau; Wang, Lih-Chung // «Mathematics», Cornell University, 2010.
8. *S. Atkinson*, Existence of isostatic, maximally random jammed monodisperse hard-disk packings / Steven Atkinson, Frank H. Stillingerb and Salvatore Torquato // PNAS. Vol. 111, No 52.
9. *J. G. Berryman*, Random close packing of hard spheres and disks / James G. Berryman// Physical review A. 1983. Vol. 27. P.1053
10. *S. Torquato*, Is Random Close Packing of Spheres Well Defined? / S. Torquato; T. M. Truskett; P. G. Debenedetti // Physical Review Letters. 2000. Vol.84, No 10

### References

1. *Elizabeth Lawson-Keister*, Jamming and arrest of cell motion in biological tissues / Elizabeth Lawson-Keister, M. Lisa Manning // «Current Opinion in Cell Biology» vol. 72. P. 146-155.
2. *Bruce DeBruhl*, Digital Filter Design for Jamming Mitigation in 802.15.4 Communication / Bruce DeBruhl, Patrick Tague // Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN). 2011.
3. *Aleksandar Donev*, Jamming in hard sphere and disk packings / Aleksandar Donev // Journal of Applied Physics. 2004. Vol. 5. P. 989.
4. Mehta, Granular Matter / Mehta // Springer New York, 1994.
5. *Misaki Ozawa*, Jamming Transition and Inherent Structures of Hard Spheres and Disks / Misaki Ozawa, Takeshi Kuroiwa, Atsushi Ikeda, Kunimasa Miyazaki // Physical review Letters. 2011. P. 109.
6. *Shoujiang Yu*, Traffic Jams in Urban Areas – A Case Study in Gourd Island City of China / Shoujiang Yu, Guiping Xiao, Lei Nie, // Applications of Advanced Technologies in Transportation; Boston Marriot, Cambridge, Massachusetts, United States, 2002.
7. *Chang*, A Simple Proof of Thue's Theorem on Circle Packing / Chang, Hai-Chau; Wang, Lih-Chung // «Mathematics», Cornell University, 2010.
8. *S. Atkinson*, Existence of isostatic, maximally random jammed monodisperse hard-disk packings / Steven Atkinson, Frank H. Stillingerb and Salvatore Torquato // PNAS. Vol. 111, No 52.
9. *J. G. Berryman*, Random close packing of hard spheres and disks / James G. Berryman// Physical review A. 1983. Vol. 27. P.1053
10. *S. Torquato*, Is Random Close Packing of Spheres Well Defined? / S. Torquato; T. M. Truskett; P. G. Debenedetti // Physical Review Letters. 2000. Vol.84, No 10

## COMPUTER MODELING OF HARD DISK RANDOM PACKING

*Vdovin Ivan E., Buzmakova Maria M.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, guzez212@gmail.com

The maximally dense stacking of circles on a bounded plane problem is considered. It remains relevant today, as its solution can be applied in a variety of practical areas, including interior design, optimization of equipment location, placement of sensor elements, etc. Various methods for solving the maximum dense stacking of circles problem were considered. Effectiveness of various algorithms comparative analyses were carried out. New approaches to solving this problem were proposed. One of the main results is to determine the optimal threshold value dependence on the plane linear size. This equation will allow you to find the circles optimal number that can be packed on a the plane.

Keywords: jamming, circles, computer and mathematical modeling, percolation theory

## **ЦИФРОВИЗАЦИЯ РИТЕЙЛА: ПРОЕКТ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОБЛАЧНОГО МАГАЗИНА САМООБСЛУЖИВАНИЯ**

*Верещагина Мария Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, masha.vereshchagina@gmail.com

*Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, averin-s-i@yandex.ru

Беспилотные магазины становятся все более популярными благодаря развитию технологий, которые уменьшают необходимость в персонале. Несколько крупных компаний, таких как Amazon и Alibaba, открыли первые магазины без кассиров с самообслуживанием. В статье анализируется существующее программное обеспечение для беспилотных магазинов. Предлагается мобильное приложение для клиентов, которое позволяет им аутентифицироваться, сканировать товары и оплачивать их. Для менеджеров магазина разрабатывается компьютерное приложение для добавления информации о продуктах в базу данных. В заключение в статье определяются требования для создания программного обеспечения для беспилотного магазина.

Ключевые слова: магазин самообслуживания, кассы самообслуживания, информационная система магазина.

В настоящее время многие торговые сети внедряют систему самообслуживания для увеличения товарооборота за счет ускорения процесса выбора и продажи товара. Развитие технологий позволило сократить количество обслуживающего персонала с помощью современных кассовых аппаратов и терминалов, которые позволяют снизить потребность в участии продавца-консультанта в процессе выбора, взвешивания, продажи и оплаты товара. Такие магазины называются «Магазин без персонала» или «Беспилотный магазин» (англ. unmanned store ).[1]

Amazon и Alibaba еще в 2017 году открыли первые магазины без касс самообслуживания и продавцов. Самый известный пример – это Amazon Go. Пользователь скачивает специальное приложение, которое генерирует QR-код, позволяющий войти в магазин, и привязывает карту для оплаты. Передвижения покупателя в магазине отслеживает компьютерное зрение и множество датчиков. Камеры фиксируют товар, который берёт покупатель, и вносят его в виртуальную корзину приложения. Если покупатель вернул товар обратно на витрину, товар автоматически удаляется из корзины. На выходе с привязанной карты списывается стоимость взятых товаров.

В России в 2020 году сразу несколько крупных продуктовых магазинов, такие как ВкусВилл, Азбука вкуса и Пятёрочка, открыли свои магазины без персонала. Они также использовали компьютерное зрение и датчики, чтобы определять, какой товар взял покупатель.

Пятёрочка представила более технически упрощенную систему, где покупатель сканирует QR-код товара с помощью специального приложения «На лету» на своем смартфоне, а стоимость покупки списывается с привязанной к приложению банковской карты.

По статистике более половины покупателей предпочитают использовать кассы самообслуживания, чтобы не контактировать с персоналом и не стоять в очередях. Учитывая возрастающий спрос на КСО, система «Магазин без персонала» может стать востребованной, а для упрощения можно создать систему, которую будет легко встроить в другие магазины, что обуславливает актуальность данной темы.

Целью работы ставится проектирование информационной системы облачного магазина самообслуживания. Для ее выполнения необходимо решить ряд задач: провести анализ существующих систем, определить задачи системы и пользовательские требования, спроектировать систему.

Несмотря на актуальность темы, на данный момент в России мало компаний, которые бы занимались разработкой ПО для крупных магазинов без персонала. В основном компании занимаются разработкой ПО для микромаркетов, которые будут отличаться от магазинов размерами и объемами продукции.

Микромаркет – это самый маленький ритейл, в котором товары продаются без персонала. Он представляет собой небольшой холодильный шкаф с электронным замком и датчиками температуры. Чаще всего такими маркетами пользуются офисные центры, фитнес-клубы и кафе. Они продают готовую еду, снеки, напитки. Данные системы близки по теме, поэтому проанализируем некоторые из них. Например, системы компаний «Izipoint» и «Briskly».

Также рассмотрим уже запущенные в России и зарубежом рабочие проекты магазинов, работающие без персонала. Например «Пятёрочка», «ВкусВилл», «Amazon GO». Данные компании разработали собственные автоматизированные системы для своих магазинов.

Проведя сравнение ПО, делаем вывод, что все системы функциональны и используют современные технологии, позволяющие пользователю совершать быструю покупку товаров в обход классических касс и общения с персоналом. Также можно увидеть, что внедрение интернет-эквайринга уже давно используется в таких системах, поэтому можно быть уверенным в функциональности данной системы оплаты.

В дополнение отметим наличие удобного приложения, в котором можно не только просматривать покупаемые товары, но и отслеживать акции и предложения магазина. Также через приложение можно отслеживать спрос на товары, которые чаще всего покупают пользователи, что повлияет на рекламу, прибыль и выкладку товара в магазине.

В приведенной ниже таблице выполним сравнение выбранных ПО.

Таблица 1. Сравнение характеристик существующего ПО

	Авторизация в магазине через QR-код	Наличие камер с компьютерным зрением	Сканирование товара с помощью камеры телефона	Поддержка интернет-эквайринга	Функциональное приложение
Izipoint	+	-	+	+	+
B-pay	+	-	+	+	+
На лету	+	+	+	+	+
Amazon GO	+	+	-	+	+



Данные системы считаются популярными: по данным опроса аналитического агентства TAdviser, более 60% опрошенных ретейлеров считают важным наличие у себя кассы самообслуживания и считают систему selfscan функциональной [2]. Из чего можно сделать вывод, что спрос на автоматизацию и роботизацию как данного процесса, так и других сторон жизни только продолжит расти. В крупных городах, где уже внедрены системы КСО и selfscan систему магазина без персонала будет легче внедрить, так как покупатели уже привыкли к системе самообслуживания. Также развитие и благосостояние оказывает влияние на уровень доверия в обществе, что в свою очередь влияет на уменьшение количества краж на кассах самообслуживания. Отметим также экономическую выгоду магазина самообслуживания: сокращаются затраты на кассовое оборудование и его обслуживание, уменьшается количество персонала в магазине. Проанализировав и рассмотрев различные системы, можно сделать вывод, что систем для магазина без персонала не очень много, но на их примере можно выявить недоработки и спроектировать более функциональную систему.

Проведенный анализ существующих систем показал, что магазинам нужна функциональная и удобная система, которая увеличит покупательский трафик и скорость покупки.

Выделим задачи, необходимые в проектируемой системе:

- хранение данных о товаре в БД;
- добавление и удаление товара;
- редактирование информации о товарах;
- поиск товара по уникальному коду;
- кроссплатформенная система администраторов и пользователей, позволяющая использовать систему на разных устройствах разными людьми;
- поддержка интернет-эквайринга;
- анализ данных о товаре и продажах.

Существующие аналоги позволили нам выделить недостатки существующих систем и разработать основные пользовательские и функциональные требования для проектируемой системы.

Пользовательские требования:

- удобный и понятный интерфейс;
- возможность поиска товара по QR-коду и просмотра информации о нём;
- автоматизация процесса покупки товара;
- возможность создания, поиска, хранения, редактирования и удаления информации в соответствии с уровнем доступа
- информация о продажах;
- информация о продукции в зависимости от критерий поиска.

Функциональные требования к системе:

- доступ к системе только для авторизованных пользователей;
- возможность поиска продуктов по QR-коду (наименование, вес в граммах, цена, оставшееся в магазине количество);
- при регистрации пользователь должен ввести свои данные: имя, дату рождения, почту, данные банковской карты;
- регистрацию нужно подтвердить, введя код из сообщения на почте;
- редактирование виртуальной корзины;
- при регистрации сотрудников вводятся следующие данные: имя, фамилия, отчество, дата рождения, должность, почта. Новый сотрудник получит доступ к системе после того, как администратор проверит данные и выдаст ему логин и пароль;

- возможность администратора редактировать информацию реестра;
- создание позиции товара;
- изменение позиции товара;
- удаление позиции товара;
- анализ данных о товаре;
- часто покупаемые товары;
- количество оставшегося товара в магазине;
- анализ продаж.

Рассмотрим особенности взаимодействия пользователей с системой.

Неавторизованный пользователь воспользуется системой только после завершения регистрации. Для регистрации пользователь вводит свои данные: логин, пароль и электронную почту. Для подтверждения авторизации на электронную почту пользователя высылается письмо с одноразовым кодом, который он вводит в поле на экране регистрации.

Далее зарегистрированный пользователь попадает на главный экран, где есть переходы к разделам «Личный кабинет», «Способ оплаты», «Корзина». Также он может отсканировать QR-код магазина, чтобы получить доступ к торговому залу.

На экране «Личный кабинет» пользователь может добавить информацию о себе. Дата рождения поможет нам понять целевую аудиторию нашего продукта и в дальнейшем проводить анализ целевой аудитории.

Раздел «Способ оплаты» представляет собой список банковских карт, привязанных к приложению, если пользователь ещё не привязывал карту, он может добавить информацию о ней, нажав на кнопку и перейдя в окно «Добавить способ оплаты». После ввода данных о карте пользователь должен нажать «Сохранить», чтобы занести в систему новый способ оплаты.

Раздел «Корзина» является открытым сканером, в верхней части которого расположен сканер, в нижней поле – корзина. Если пользователь еще не отсканировал товар, корзина останется пустой. При сканировании QR-кода товар появляется в корзине. Если пользователь передумал, он может удалить товар из корзины. Если был отсканирован хотя бы 1 товар, появляется кнопка «Оплатить» и поле с указанием общей стоимости покупки. После оплаты появляется кнопка, при нажатии которой можно выйти из магазина.

Для сотрудников магазина (менеджеров) будет работать компьютерное приложение, включающее в себя авторизацию пользователя, добавление позиции товара и запрос на остатки. Для авторизации менеджер получает от администратора письмо с логином и паролем учётной записи. Также на почту, указанную при найме на работу, приходит одноразовый пароль для подтверждения входа. После входа в приложении менеджеру доступна информация об имеющемся товаре в магазине, есть возможность создать позицию и занести информацию о товаре с помощью сканирования сканером. После завершения работы менеджер выходит из учётной записи.

Администратор магазина обладает правами доступа ко всей базе через менеджер, может совершать все операции с БД и проводить SQL запросы.

Важно, чтобы интерфейс приложения был удобен и понятен для пользователя, так как это поможет снизить порог вхождения новых покупателей и повысить интуитивность интерфейса.

Проведенный анализ позволяет спроектировать информационную систему для магазина самообслуживания, которая повысит покупательский трафик и уменьшит затраты на обслуживание магазина.

### **Библиографический список**

1. Харитонов А.В. Магазин без кассы – пока фантастика, а без кассира – уже реальность: [Электронный ресурс] // Журнал для роста бизнеса 2023 URL: [https://kontur.ru/market/spravka/38156-magazin\\_bez\\_kassy\\_bez\\_kassira](https://kontur.ru/market/spravka/38156-magazin_bez_kassy_bez_kassira) (Дата обращения: 24.03.2023)
2. Сравнение кассовых решений:[Электронный ресурс] // TAdviser Москва 2 квартал 2021 URL: <https://www.tadviser.ru/a/620671>

### **References**

1. Kharitonov A.V. A store without a cashier is still a fantasy, but without a cashier is already a reality: [Electronic resource] // Journal for Business Growth 2023
2. Comparison of cash solutions: [Electronic resource] // TAdviser Moscow Q2 2021 URL: <https://www.tadviser.ru/a/620671>

## **DESIGNING AN INFORMATION SYSTEM FOR A SELF-SERVICE CLOUD STORE**

*Vereshchagina Maria N.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
[masha.vereshchagina@gmail.com](mailto:masha.vereshchagina@gmail.com)

*Averin Sergey I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [averin-s-i@yandex.ru](mailto:averin-s-i@yandex.ru)

Unmanned stores are becoming increasingly popular due to the development of technologies that reduce the need for personnel. Several large companies, such as Amazon and Alibaba, have opened the first stores without self-service cashiers. The article analyzes the existing software for unmanned stores. We offer a mobile application for customers that allows them to authenticate, scan products and pay for them. A computer application is being developed for store managers to add information about products to the database. In conclusion, the article defines the requirements for creating software for an unmanned store.

**Keywords:** Self-service store, self-service cash desks, store information system.

## МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ ИНФЕКЦИОННОГО ЗАБОЛЕВАНИЯ НА ОСНОВЕ КЛЕТОЧНЫХ АВТОМАТОВ

*Власов Андрей Алексеевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, vlasovavasoftware@gmail.com

Рассматривается моделирование распространения инфекционных заболеваний на основе модифицированной SEIR-модели, с помощью клеточных автоматов. Для этого приводится краткий обзор существующих методов прогнозирования, рассматриваются их достоинства и недостатки. Учитывая приведённые аргументы, построена система для построения прогнозов. Для создания данной системы была построена математическая модель её работы, на основе которой на языке программирования «C#» была разработана программа, позволяющая строить соответствующие прогнозы распространения инфекции. Тестирование программы проходило на примерах распространения ОРВИ в России, а также на примере штамма «Омикрон» COVID-19 в городе Нью-Йорк, США. Основываясь на результатах проведённых исследований, разработанная система может предсказывать ход распространения инфекции. С её помощью можно моделировать эффективность мер по борьбе с болезнью, не тратя огромные средства. Прогнозирование распространения инфекции на основе клеточных автоматов является простым и точным методом, который при этом универсален в применении. Дальнейшее развитие данного метода является перспективным и эффективным.

Ключевые слова: клеточные автоматы, SEIR-модель, инфекционные заболевания.

**Введение.** В настоящее время из-за недавней пандемии COVID-19, начавшейся в 2020 году, все люди, от простых граждан до глав государств обеспокоены распространением инфекционных заболеваний, которые способны ввергнуть нашу цивилизацию в настоящий кризис, в том числе экономический. Хотя на сегодняшний день пандемия COVID-19 официально завершена, в будущем человечество может столкнуться с новыми масштабными вспышками инфекционных заболеваний.

Исходя из вышесказанного, появляется острая необходимость в методах, позволяющих прогнозировать ход распространения заболевания, а также заранее предусмотреть необходимые меры для борьбы с ним. Настоящие математические модели и решения, на создание которых выделяются огромные ресурсы, часто бывают слишком «громоздкими», поскольку их сложно адаптировать под быстро меняющиеся условия пандемий. Необходимы надёжные решения для частных случаев пандемии, которые при этом должны обеспечивать универсальность и простоту в применении.

**Существующие системы прогнозирования распространения инфекционных заболеваний.** Число работ, посвящённых прогнозированию инфекционной заболеваемости, стремительно растёт по мере увеличения вычислительных мощностей современной техники, а главное, по мере появления статистики, позволяющей провести соответствующий анализ [1,2]. Частота сбора данных обуславливается видом инфекции, текущей эпидемиологической обстановкой и организационными возможностями. В частности, эпидемиологические данные по гриппу и острым респираторным вирусным инфекциям (ОРВИ) при отсутствии эпидемий собирают еженедельно, однако в связи с текущей глобальной пандемией здравоохранитель-

ные организации всех стран стараются обновлять свои данные ежедневно, некоторые – ежедневно.

Грипп и ОРВИ занимают ведущее место среди инфекционных болезней человечества и, несмотря на все проводимые противозидемические мероприятия, заболеваемость ими не имеет тенденции к снижению как в России, так и за рубежом на протяжении многих столетий [3-6]. Поэтому в настоящей работе тестирование моделей проводится на основе распространения гриппа. Техники прогнозирования гриппа подходят непосредственно или могут быть легко адаптированы для других инфекций, передающихся контактно-бытовым и воздушно-капельным путем, в частности COVID-19. Впрочем, значительная часть методов может быть использована для прогнозирования даже неинфекционной заболеваемости.

**Статистические методы прогнозирования.** Для подавляющего множества инфекционных заболеваний, как и для коронавируса, гриппа и ОРВИ, характерны вспышки заболевания, которые со временем приобретают сезонный характер, повторяющиеся из года в год. В таких условиях на основе известного периода сезонности прогноз для дальнейших периодов может быть вычислен посредством уже имеющихся статистических данных соответственно. Самый популярный подход получения оценок ординарной ожидаемой заболеваемости – усреднение. Он широко применяется в России, действующая методика по расчёту заболеваемости по ОРВИ и гриппу представляет собой усреднённые данные конкретных недель каждого года с одинаковым номером за последние 5-10 лет [4]. Такая методика предполагает, что исследуемый случайный процесс носит неизменный характер и наблюдения в течении следующего года – его следующая реализация.

Однако в большинстве практических случаев это допущение не соблюдается. Одно из расширений данного метода, при присутствии в динамике явных трендов, это – поточечные линейные оценки [7]. В данном случае наблюдения, которые соответствуют одному и тому же промежутку времени, аппроксимируются прямой линией тренда. При этом коэффициент наклона линий для сечений в рамках одного года общий, а свободный член в уравнениях прямых подбирается индивидуально.

Такие методы используются столь широко исходя ввиду своей простоты в применении, при этом они не учитывают корреляцию при последовательных наблюдениях, и, как следствие, они могут применяться только для получения грубых, долгосрочных проекций заболеваемости.

Ещё один популярный способ прогнозирования заболеваемости – регрессионный анализ. В теме, рассматриваемой в настоящей работе, основные задачи регрессии – это нахождение оценок неизвестных параметров и формировании функциональной зависимости между заболеваемостью и определяющими её факторами. Бурком Х.С. выделял две группы среди регрессионных моделей: неадаптивные и адаптивные модели [8-9].

В неадаптивной модели для оценки параметров используется вся имеющаяся статистика, но при этом модель должна учитывать сезонные тренды. В частности, Серфлинг Р.Е. предложил циклическую функцию [7].

Неадаптивные регрессионные модели призваны учитывать всю предысторию заболеваемости на анализируемой территории. Для их построения используются все имеющиеся данные или, по крайней мере, наблюдения последних лет, обладающие схожими характеристиками. Но, если свойства процесса заболеваемости изменились, вероятно, устаревшие данные уже не помогут уточнить прогноз.

Адаптивные регрессионные модели представляют ограниченный временной отрезок и должны быть чувствительны к меняющимся тенденциям заболеваемости. Важную роль при

этом играет число наблюдений, на основе которых верифицируются параметры модели. Чаще всего адаптивные модели применяются при построении краткосрочных прогнозов, и диапазон наблюдений составляет, как правило, один-два месяца. Однако, чем больше число исходных параметров, тем больше требуется наблюдений для получения их оценок, но при этом, чем больше период наблюдений, тем сложнее требуется функция для представления эпидемической кривой в выбранный временной участок. Например, при прогнозировании заболеваемости гриппом на 1-2 недели вперед, с помощью простой линейной регрессии скользящее окно должно вмещать данные 4–6 недель [6].

Подводя итоги, можно утверждать, что неадаптивные модели позволяют составить прогноз на любой заданный срок, однако они не подходят для краткосрочного прогнозирования, а также не могут учитывать локальные колебания показателей заболеваемости. Адаптивные модели, напротив, являются хорошим методом для краткосрочного прогнозирования. Однако при использовании их для расчётов среднесрочных и долгосрочных прогнозов может понадобиться большое количество наблюдений, что сильно увеличивает сложность функциональной зависимости параметров.

**Методы на основе машинного обучения.** Основным инструментом, используемым при прогнозировании с помощью машинного обучения, являются искусственные нейронные сети (ИНС) [10]. При прогнозировании распространения заболевания, входными данными являются статистика заболеваемости и при наличии, значения факторов, ее предопределяющих. Существует множество вариантов использования ИНС для построения прогнозов заболеваемости: ИНС могут отличаться своей внутренней архитектурой, количеством задействованных нейронов и способом обучения. Цель обучения ИНС – определение явных и неявных зависимостей между входными и выходными данными. Для построения прогнозов высокой точности необходимо большое количество наблюдений, которые будут являться входными данными для обучения нейронной сети. К сожалению, большинство специалистов приходят к единому мнению о том, что существующего объёма статистики недостаточно для создания универсальной нейронной сети, которая будет способна делать точные долгосрочные прогнозы распространения инфекционных заболеваний.

**Математическое моделирование распространения инфекции.** Данная группа методов создана в рамках “биологического” подхода [11]. В отличие от предыдущих методов, которые опирались только на статистические показатели заболеваемости, «биологический» подход основывается на самом механизме распространения инфекции и моделирует его. Основные модели распространения заболевания были опубликованы ещё в начале XX века, в их основе системы дифференциальных уравнений. Классической является модель SIR, представляющая из себя систему обыкновенных дифференциальных уравнений, разработанная в 1920-х годах Кермаком и МакКендриком. В данной модели изучаемое множество разбивают на группы, на здоровых ( $S$  – susceptible), больных ( $I$  – infectious) и выздоровевших людей ( $R$  – recovered). Модель типа SIR является аналогией взаимодействий людей исследуемого множества друг с другом. В зависимости от решаемой задачи, требуемой точности и особенности исследуемого заболевания могут выделяться различные категории людей, например, вакцинированные, с повышенным риском заражения и другие [5]. В современных исследованиях с использованием данной модели количество уравнений может достигать до 1000 и более.

Однако модели SIR трудно применимы к большой численности анализируемой популяции и не учитывают то, что отдельные представители непрерывно и равномерно перемещаются в популяции.

**Сравнение изученных методов.** Результаты проведённого анализа представлены в табл. 1. В ней перечислены изученные подходы к прогнозированию распространения инфекции, условными обозначениями кратко собрана общая характеристика каждого из методов по основным параметрам важности. Из таблицы можно сделать вывод о том, что методов, которые смогут составлять точные долгосрочные прогнозы для большой исследуемой популяции, практически отсутствуют. С другой стороны, почти все методы пригодны для создания среднесрочных прогнозов, которые являются наиболее важными и ценными для современного здравоохранения.

Таблица 1. Сравнение методов прогнозирования инфекционной заболеваемости

			Сложность	Дальность прогноза	Объём исследуемого множества	Количество предварительных статистических данных
Статистический анализ	Поточечные линейные оценки		Простые	Долгосрочные	Большое	Большой
	Регрессионный анализ	Неадаптивные модели	Средний	Среднесрочные	Большой	Большой
		Адаптивные модели	Средний	Краткосрочные	Большой	Маленький
Машинное обучение			Средние	Краткосрочные – среднесрочные	Большой	Очень большой
Математические модели			Сложные	Среднесрочные	Маленький-средний	Маленький

Однако среди представленных методов нет точных, простых в разработке и в применении, которые можно было бы использовать для конкретных ситуаций в условиях появления новых, потенциально опасных заболеваний. К сожалению, на текущий момент не существует универсальных методов прогнозирования, и исследователи вынуждены выбирать для каждой ситуации наиболее подходящий в текущих условиях метод.

**Использование клеточных автоматов для построения прогнозов распространения инфекционных заболеваний.** Клеточный автомат – динамическая, дискретная система, представляющая собой совокупность одинаковых клеток, которые соединены между собой. Всё множество клеток образует решётку клеточного автомата. Решётки могут отличаться как по размерности, так и по форме самих клеток. Зачастую решётки клеточных автоматов закольцовываются при достижении своей границы. Каждая отдельная клетка является конечным автоматом, то есть может находиться в одном из конечного множества состояний. Кроме того, для каждой такой клетки определено множество других клеток, это множество называется окрестностью. В классических реализациях в окрестность клетки попадают только близлежащие клетки, которые непосредственно с ней граничат, однако существуют модификации, где окрестность может представлять собой множество других клеток, отобранных по другому произвольному правилу.

Первоначально каждой клетке автомата задаётся исходное состояние, один шаг автомата подразумевает обход всех клеток, и на основе данных о текущем состоянии клетки определяются новые состояния клетки, которое она будет иметь при следующем шаге. Состояние клетки в следующий момент времени однозначно определяется с помощью функции от её собственного состояния и состояния её окрестности в текущий момент времени.

**Построение математической модели.** Клеточные автоматы использовались для создания моделей ещё в начале XX века. Математические модели, построенные на их основе, от-

личались главным образом тем, что имитировали саму эпидемию заболевания, тогда как другие методы пытались лишь спрогнозировать статистические данные на основе других статистических данных. У каждой болезни есть свои стадии, достаточно точно можно узнать среднее время болезни, а также её последующий ход. Все эти условия идеально подходят под определение клеточного автомата, стадии болезни составляют множество допустимых состояний клеток, течение болезни однозначно определяет правила перехода между состояниями. Вероятности перехода между состояниями также весьма легко просчитываются в ходе относительно небольшого числа наблюдений.

Одним из наиболее распространённых подходов является моделирование на основе SIR-модели, которая описывает распространение инфекции через три категории людей: подверженных (S), инфицированных (I) и выздоровевших (R). Этот подход был описан в работе Кермака и Маккендрика при построении модели распространения бубонной чумы в Бомбее (1927).

Дополненная модель SEIR является модификацией SIR-модели, в которой добавлено состояние «выздоровевшие, но всё ещё подверженные риску заражения» (E – exposed). Данный подход широко используется для моделирования распространения COVID-19. Он был описан в работе Ли и соавторов [6].

Исходя из всего вышесказанного, была составлена следующая математическая модель для прогнозирования распространения инфекционного заболевания.

Клетка будет представлять собой конкретного человека, а вся решётка клеточного автомата моделирует исследуемое множество людей. Выделяются следующие возможные состояния:

$H_1$  – здоровый, без иммунитета (может заразиться);

$H_2$  – здоровый, с врождённым иммунитетом (не может заразиться никогда);

$H_3$  – здоровый, с приобретённым иммунитетом (в результате вакцинации или перенесённой болезни);

$P_1$  – больной, без симптомов (инкубационный период или лёгкое течение болезни, тем не менее является активным распространителем болезни);

$P_2$  – больной, активная фаза болезни (человек знает, что болеет, контакты снижаются, но тем не менее, является распространителем);

$D$  – летальный исход.

Возможные переходы из состояний  $P_1, P_2, H_3$  происходят только после истечения определённого количества дней, которое задаётся в начальных условиях при моделировании конкретных заболеваний. Кроме того, в начальных условиях задаётся количество клеток в состояниях  $H_2, P_1$ , вероятности переходов между состояниями, все остальные клетки при этом по умолчанию находятся в состоянии  $H_1$ . В случае если сумма вероятностей переходов в другие состояния будет меньше единицы, клетка с некоторой вероятностью может остаться в текущем состоянии. В настоящем конечном автомате будет использоваться несколько типов решётки, а именно «Квадрат», «Треугольник», «Линия» и «Куб». В зависимости от типа клеточной решётки, будут рассмотрены разные варианты окрестностей клетки, начиная от 2



«соседей», заканчивая 26 (рис. 1). В конечном итоге будет построено 29 различных моделей на основе клеточного автомата.

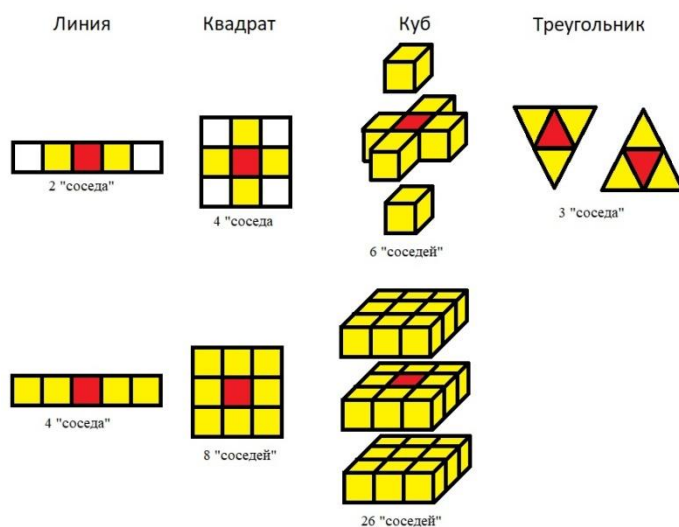


Рис. 1. Демонстрация выбранных окрестностей клетки, в зависимости от «топологии» решётки клеточного автомата

Полученная математическая модель:

Текущее состояние  $\sigma_{i,j}$  клетки с целочисленными координатами  $i, j$ :

$$\sigma_{i,j} \in \sum \{1, 2, 3, 4, 5, 6, 7\}$$

Изменение состояний происходит согласно правилам перехода:

$$\sigma_{i,j}(t+1) = f(\sigma_{k,l}(t) \mid (k,l) \in N(i,j)),$$

где  $N(i,j)$  – окрестность точки  $(i,j)$ :

$$N(i,j) = \{(k,l) \mid |i-k| + |j-l| \leq 1\}$$

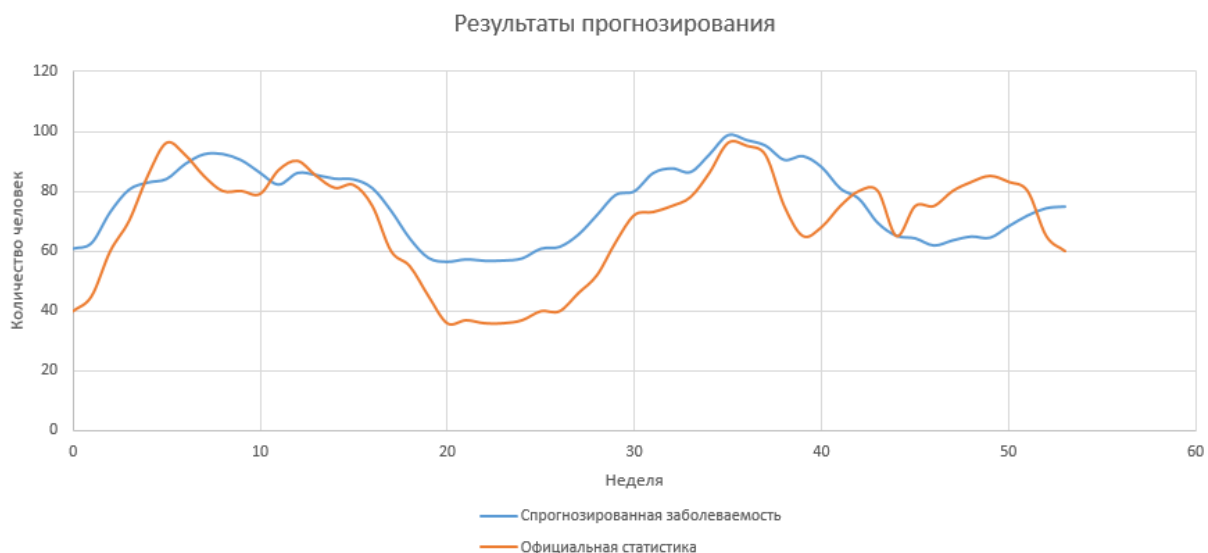
**Проектирование и создание системы для построения прогнозов распространения инфекционных заболеваний.** В настоящей работе система прогнозирования реализована с помощью платформы «WPF» на языке программирования «C#», являющимся частью программной экосистемы «MICROSOFT .NET». Для работы с множеством состояний, которые могут иметь клетки из решётки автомата, бизнес логика системы будет основана на шаблоне программирования «машина состояний». Каждое возможное состояние отдельной клетки будет выделено в собственный класс, методы которого будут реализовывать правила перехода между состояниями. Это позволит в будущем добавлять новые возможные состояния или изменять особенности уже существующих. Выделяются также такие понятия, как «клетка» и «клеточный автомат», для которых соответственно созданы отдельные классы.

Для реализации клеточных автоматов различных типов был применён шаблон программирования «Декоратор», для этого важно следовать принципу «Зависимости на абстракциях». Алгоритм по обновлению всех состояний клеток будет иметь сложность  $O(n)$ .

Интерфейс для пользователя – это главная и единственная часть системы, доступная ему. Для его реализации был использован шаблон «MVVM» (англ. Model-View-ViewModel), который позволяет разграничивать программные компоненты с их визуальным представлением.

**Тестирование работы системы по прогнозированию распространения инфекционного заболевания. Анализ полученных результатов.** В качестве примера для настройки и тестирования программы будем использовать данные по распространению гриппа и ОРВИ в России из расчёта на 10 тысяч человек. По данным существующей статистики были сформированы значения параметров, необходимых для построения прогноза.

Для прогнозирования был выбран автомат с решёткой типа «Квадрат» с окрестностью в 4 клетки. Вероятности заражения были рассчитаны из соотношения, что один больной человек может заразить одного-двух здоровых.



**Рис. 2. Заболеваемость гриппом.**

Исходя из результатов (рис. 2), можно сделать вывод о том, что данная система в целом пригодна для создания прогнозов по распространению инфекционных заболеваний. Полученные результаты, отражают сезонность, разность между пиками заболеваемости примерно 25 недель. Однако, тем не менее существует погрешность в нижней границе количества заболевших. Она обусловлена особенностями работы клеточного автомата, а именно невозможностью рассчитывать вероятности перехода между состояниями динамически (т. е. в реальном времени), а также тем, что решётка автомата строго структурирована.

Самая актуальная на сегодняшний день инфекция для прогнозирования – это COVID-19, особенность которой заключается в том, что её случайные мутации порождают большое количество новых штаммов, которые существенно отличаются друг от друга. Исходя из данных на настоящий момент, каждый последующий штамм является более заразным и легче переносится людьми, поэтому каждый отдельный штамм может рассматриваться как отдельная болезнь. Исходя из особенностей данной болезни, прогнозирование было выполнено с помощью автомата с решёткой типа «Куб» и окрестностью в 6 клеток. Прогноз был построен для города Нью-Йорк, США. Для него существует точная и обширная статистика, собранная за последнее время [12].

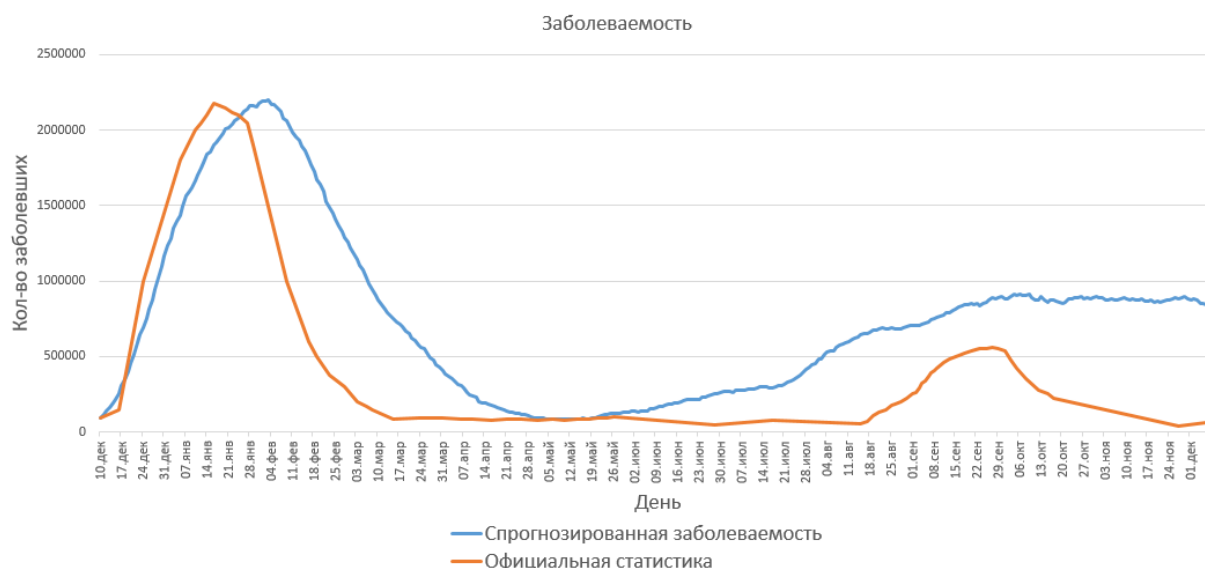


Рис. 3. Заболеваемость COVID-19.

Учитывая, что входные данные также были сформированы на основе реальной статистики, можно утверждать, что дальнейшее развитие распространения «Омикрон» штамма COVID-19 представлено на соответствующем графике (рис. 3).

Важной характеристикой моделей клеточных автоматов является их «топология». На рис. 4 представлена зависимость характера распространения заболеваний от «топологии» решётки клеточного автомата при одинаковых вероятностях перехода между состояниями.

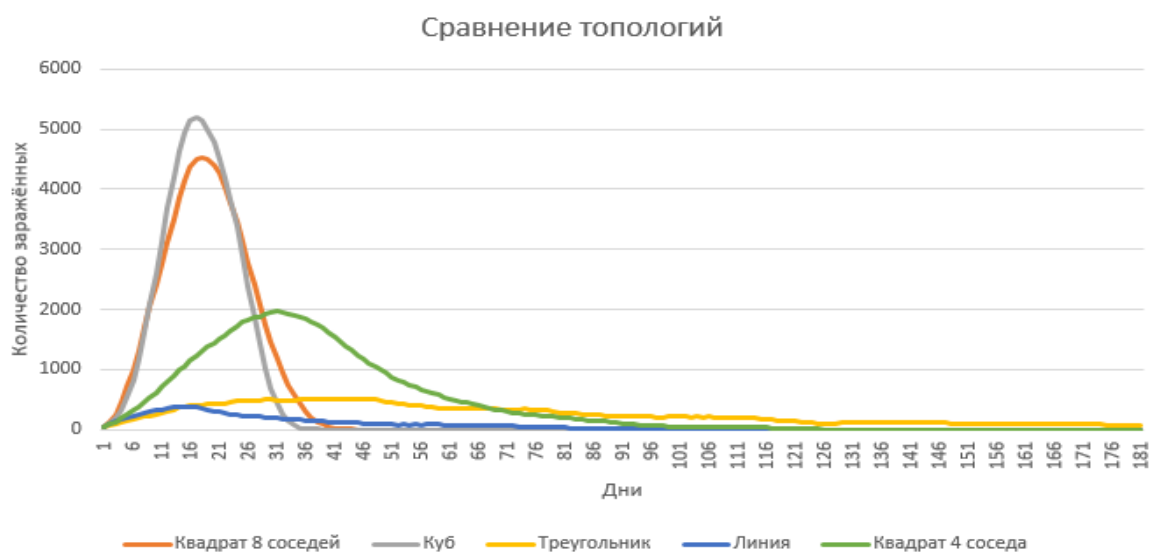


Рис. 4. Зависимость количества заражённых от выбранной «топологии» решётки клеточного автомата

Основываясь на данных графика, можно утверждать, что топологии с большим количеством соседей подходят для инфекционных заболеваний, которые передаются воздушно-капельным путём и имеют «взрывные» темпы заболеваемости. В тоже время топологии с меньшим количеством соседей помогают строить долгосрочные прогнозы для не густонаселённых регионов.

**Построение прогнозов распространения заболеваний.** В предыдущем разделе тестирование работоспособности программы проходило на основе данных по распространению гриппа, ОРВИ и COVID. Заболеваемость ОРВИ имеет явный сезонный характер, который не

меняется десятилетиями, кроме того, данные болезни не имеют существенных различий между их штаммами, которые могут появляться в результате случайных мутаций. Следовательно, прогнозирование распространения этих болезней в будущем не представляет интереса.

Исходя из полученных данных, спрогнозируем предполагаемое количество жертв COVID-19 на основе предыдущей модели прогноза для города Нью-Йорк. если немного увеличить темпы вакцинации (до 0.002) и ужесточить правила социальной дистанции (0.06), темпы распространения болезни должны существенно сократиться (рис. 5.).

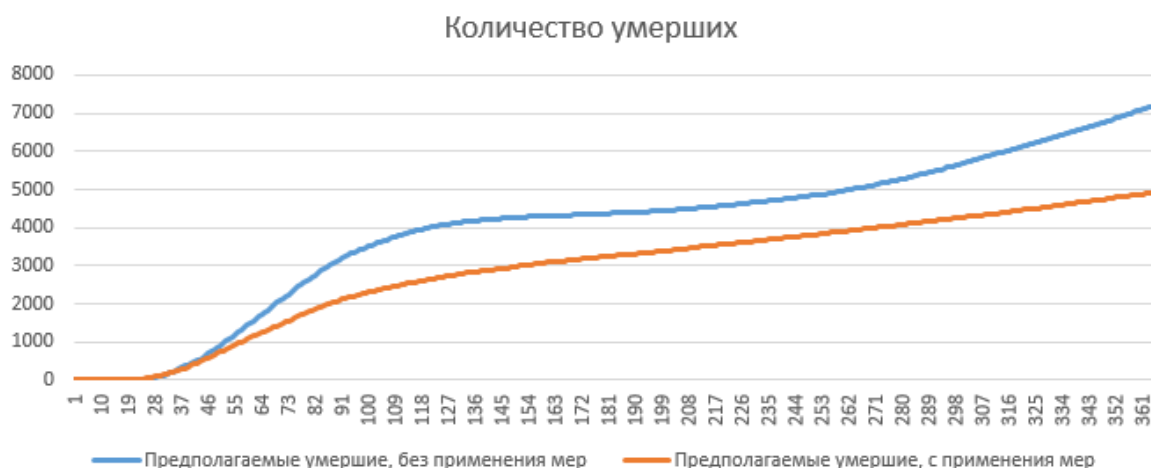


Рис. 5. График количества летальных исходов

В последнее время специалисты всё чаще утверждают, что вирус иммунодефицита человека в России приобретает характер эпидемии. Наша страна занимает пятое место по количеству новых случаев заражения ВИЧ. Построим долгосрочный прогноз распространения данного инфекционного заболевания. Для этого подойдёт клеточный автомат с решёткой типа «Линия» и окрестностью в две клетки. Судя по результатам (рис. 6), без принятия серьёзных мер, темпы заболеваемости в ближайшие 5 лет не будут снижаться и данное инфекционное заболевание может стать серьёзной проблемой. Ожидается увеличение количества заражённых до 2 млн. к 2025 году.

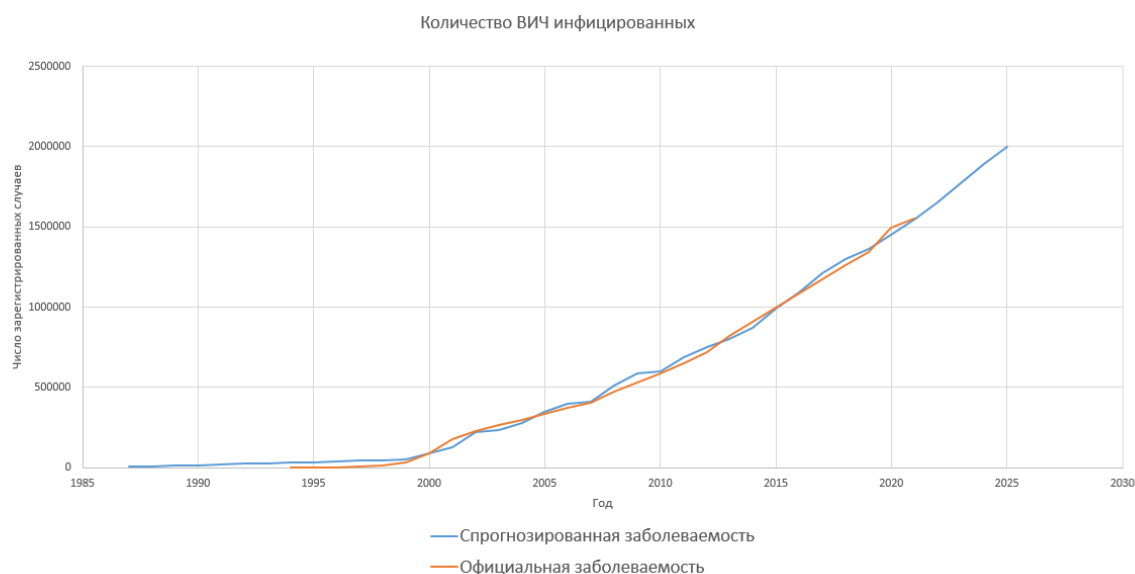


Рис. 6. Прогнозирование распространения ВИЧ в России.

В начале декабря 2022 года в России участились случаи свиного гриппа. На данный момент подтверждённые случаи были во всех регионах РФ, а в Пермском крае, по официальным данным, на карантине находилось около 50 тыс. человек. Исследователи полагают, что распространяемый штамм очень схож со штаммом H1N1 2009 года.



**Рис. 7. Прогнозирование распространения свиного гриппа H1N1 в Пермском крае. Официальная статистика за 2007-2011 года и спрогнозированная.**

Таким образом, используя статистику прошлой эпидемии свиного гриппа 2009 года, строился прогноз распространения данного инфекционного заболевания для Пермского края. Для этого использовался клеточный автомат с решёткой типа «Треугольник» и окрестностью в три клетки. Судя по результатам (рис. 7), данное инфекционное заболевание повторяет динамику эпидемии 2009 года, что влечёт за собой огромную нагрузку на систему здравоохранения, которая только восстановилась после пандемии COVID-19.

**Вывод.** В современных условиях глобализации появление новой болезни очень опасно. В то время как люди заняты обсуждением того, какие меры по борьбе болезнями стоит предпринять, инфекции распространяются всё быстрее. Промедление в решении этих сложных задач может стоить огромных издержек. В настоящей работе реализован программный продукт, использующий подход «клеточных автоматов».

Тестирование программы проходило на примерах распространения ОРВИ в России, а также на примере штамма «Омикрон» COVID-19 в городе Нью-Йорк, США. Основываясь на результатах проведённых исследований, полагаем, что разработанная система может предсказывать ход распространения инфекции. С её помощью можно моделировать эффективность мер по борьбе с болезнью, не тратя огромные средства.

Подводя итог всему вышесказанному, можно утверждать, что прогнозирование распространения инфекции на основе клеточных автоматов является простым и точным методом, который при этом универсален в применении. Дальнейшее развитие данного метода является перспективным и эффективным.

### Библиографический список

1. *Бейли Н.* Математика в биологии и медицине. М.: Мир, 1970. 327 с.
2. *Кондратьев М.А.* Методы прогнозирования и модели распространения заболевания // Компьютерные исследования и моделирование. 2013. Том 5, №5. С.863-882.
3. *Боев Б. В., Макаров В. В.* Компьютерное моделирование и прогнозирование эпидемий птичьего гриппа // Ветеринарная патология. 2005. № 3. С. 49–58.
4. *Киселев О. И., Маринич И. Г., Карпова Л. С., Ежлова Е. Б., Лазикова Г. Ф., Ватоллина А. А.* Методика расчета эпидемических порогов по гриппу и острым респираторным вирусным инфекциям по субъектам Российской Федерации. М.: НИИ гриппа Северо-Западного отделения РАМН, 2010. 88 с.
5. *Coburn B. J., Wagner B. G., Blower S.* Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1) // BMC Medicine. 2009. Vol. 7, № 30.
6. *Cowling B. J., Wong I. O. L., Ho L. M., Riley S., Leung G. M.* Methods for monitoring influenza surveillance data // International Journal of Epidemiology. 2006. Vol. 35, № 5. P. 1314–1321.
7. *Serfling R. E.* Methods for Current Statistical Analysis of Excess Pneumonia-influenza Deaths // Public Health Reports. 1963. Vol. 78, № 6. P. 494–506.
8. *Burkom H. S.* Development, Adaptation, and Assessment of Alerting Algorithms for Bio-surveillance // Johns Hopkins APL Technical Digest. 2003. Vol. 24, № 4. P. 335–342.
9. *Burkom H. S., Murphy S. P., Shmueli G.* Automated Time Series Forecasting for Biosurveillance // Statistics in Medicine. 2007. Vol. 26, № 22. P. 4202–4218.
10. *Ясницкий Л. Н.* Интеллектуальные системы. М.: Лаборатория знаний, 2016. 221 с.
11. *Myers M. F., Rogers D. J., Cox J., Flahault A., Hay S. I.* Forecasting Disease Risk for Increased Epidemic Preparedness in Public Health // Advances in Parasitology. 2000. Vol. 47. P. 309–330.
12. *COVID Live – Coronavirus Statistics – Worldometer.* [Электронный ресурс]. – URL: <https://www.worldometers.info/coronavirus/> (дата обращения: май 2023)

### References

1. *Bailey N.* Mathematics in biology and medicine. Moscow: Mir, 1970. 327 p. (In Russ)
2. *Kondratiev M.A.* Methods of forecasting and models of the spread of the disease // Computer research and modeling. 2013. Volume 5, No. 5. pp.863-882. (In Russ)
3. *Boev B. V., Makarov V. V.* Computer modeling and forecasting of avian influenza epidemics // Veterinary pathology. 2005. No. 3. P. 49-58. (In Russ)
4. *Kiselev O. I., Marinich I. G., Karpova L. S., Ezhlova E. B., Lazikova G. F., Vatolina A. A.* Methodology for calculating epidemic thresholds for influenza and acute respiratory viral infections in the subjects of the Russian Federation. Moscow: Research Institute of Influenza of the North-Western Branch of the Russian Academy of Medical Sciences, 2010. 88 p. (In Russ)
5. *Coburn B. J., Wagner B. G., Blower S.* Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1) // BMC Medicine. 2009. Vol. 7, № 30.
6. *Cowling B. J., Wong I. O. L., Ho L. M., Riley S., Leung G. M.* Methods for monitoring influenza surveillance data // International Journal of Epidemiology. 2006. Vol. 35, № 5. P. 1314–1321.
7. *Serfling R. E.* Methods for Current Statistical Analysis of Excess Pneumonia-influenza Deaths // Public Health Reports. 1963. Vol. 78, № 6. P. 494–506.

8. *Burkom H. S.* Development, Adaptation, and Assessment of Alerting Algorithms for Bio-surveillance // Johns Hopkins APL Technical Digest. 2003. Vol. 24, № 4. P. 335–342.
9. *Burkom H. S., Murphy S. P., Shmueli G.* Automated Time Series Forecasting for Biosurveillance // Statistics in Medicine. 2007. Vol. 26, № 22. P. 4202–4218.
10. *Yasnitskiy L. N.* Intelligent systems. M.: Laboratory of Knowledge, 2016. 221 p. (In Russ)
11. *Myers M. F., Rogers D. J., Cox J., Flahault A., Hay S. I.* Forecasting Disease Risk for Increased Epidemic Preparedness in Public Health // Advances in Parasitology. 2000. Vol. 47. P. 309–330.
12. *COVID Live – Coronavirus Statistics – Worldometer.* [Electronic resource]. – URL: <https://www.worldometers.info/coronavirus/> (дата обращения: май 2023)

## **SIMULATION OF THE SPREAD OF AN INFECTIOUS DISEASE BASED ON CELLULAR AUTOMATA**

*Vlasov Andrey A.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [vlasovavasoft@gmail.com](mailto:vlasovavasoft@gmail.com)

Modeling of the spread of infectious diseases based on a modified SEIR model using cellular automata is considered. For this purpose, a brief overview of existing forecasting methods is provided, their advantages and disadvantages are considered. Taking into account the arguments given, a system for making forecasts has been built. To create this system, a mathematical model of its operation was built, on the basis of which a program was developed in the programming language "C#", allowing to build appropriate forecasts of the spread of infection. The program was tested on the examples of the spread of SARS in Russia, as well as on the example of the "Omicron" COVID-19 strain in New York City, USA. Based on the results of the conducted studies, the developed system can predict the course of the spread of infection. With its help, you can simulate the effectiveness of measures to combat the disease without spending huge amounts of money. Predicting the spread of infection based on cellular automata is a simple and accurate method, which at the same time is universal in application. Further development of this method is promising and effective.

**Keywords:** cellular automata, SEIR model, infectious diseases.

## **ВИЗУАЛИЗАТОР АЛГОРИТМОВ РАБОТЫ С ДИНАМИЧЕСКИМИ СТРУКТУРАМИ ДАННЫХ (ОДНОНАПРАВЛЕННЫЕ СПИСКИ)**

*Волков Илья Игоревич, Ромашкина Татьяна Витальевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, t\_obuch\_11@mail.ru

В настоящей работе рассматриваются алгоритмы обработки динамических структур данных на примере однонаправленного списка. Описываются способы графической интерпретации действий над однонаправленными списками. Предлагается способ визуализации алгоритмов работы с однонаправленными списками, так как абстрактность представления подобных структур данных затрудняет понимание и восприятие работы с ними. Визуализация алгоритмов обработки однонаправленных списков происходит в соответствии со строками программного кода пользователя и с отображением реальных адресов размещения элементов списка в динамической памяти. Рассматривается проектирование и разработка приложения для визуализации алгоритмов работы с однонаправленными списками, на базе клиент-серверной структуры. Приложение создано с использованием языков программирования Golang, TypeScript, Dot и C++, библиотек ReactJS и LibTooling.

Ключевые слова: динамические структуры данных, визуализатор алгоритмов работы с однонаправленными списками, клиент-серверное приложение.

Под динамическими структурами данных понимают структуры данных, память под которые выделяется и освобождается во время работы программы. В процессе работы с подобными структурами может изменяться как количество элементов, так и характер связи между элементами [1]. С одной стороны, динамические структуры данных в отличие от статических обладают большей гибкостью и позволяют обрабатывать данные, размерность которых изменяется во время исполнения программы. С другой стороны, разработка программ с использованием динамических структур данных намного сложнее: логические ошибки в работе с динамическими структурами данных нельзя обнаружить до момента исполнения.

Был проведен анализ различных источников с графическим отображением списков [1,2,3,4,5], с возможностью визуализации алгоритмов работы со списками [6,7].

Визуализатор состоит из двух частей: клиентского и серверного приложения. Такое разделение обусловлено повышенной портируемостью и доступностью программы: для её запуска, требуется только браузер. Модули, связанные с разбором и обработкой исходного кода алгоритма, были вынесены на сервер.

Серверное приложение реализовано на языке Golang. Его выбор связан с большой стандартной библиотекой с фокусом на сетевые приложения, а также набором инструментов и абстракций для работы с асинхронными операциями.

Клиентское приложение реализовано на языке TypeScript. Этот язык является надмножеством языка JavaScript и позволяет добавить типизацию в данные, полученные от сервера. Для реализации пользовательского интерфейса была использована библиотека ReactJS. Она позволила декларативно описать структуру интерфейса, а также его визуальное представление на предметно-ориентированном языке JSX. Задача по «отрисовке» динамических струк-



тур данных была сформулирована через теорию графов и представляет собой расположение графа на плоскости.

Представление динамических структур в памяти было сформировано в виде ориентированного графа, где вершина – содержимое очередной ячейки памяти, а ребро – указатель на неё. Такое представление позволяет использовать существующие решения для визуализации графов, а также существующие алгоритмы по работе с ними. Одно из таких решений – система Graphviz [8]. Для описания структуры графа использовался предметно-ориентированный язык Dot [9]. Этот язык позволяет установить связь между любыми именованными элементами. Для визуального восприятия графа необходимо сохранять порядок следования вершин при его отображении. По умолчанию язык Dot не сохраняет порядок следования элементов без явного указания связей между ними. Для решения этой проблемы в описания на языке Dot были добавлены невидимые связи с атрибутом `invis`. Кроме того, необходимо сгруппировать узлы, входящие в однонаправленный список, и расположить их на одном уровне. Язык Dot поддерживает разделение описаний и связей, что позволяет разбить генерацию описаний на несколько стадий и упростить код. Для описания составных элементов была использована табличная верстка. Она позволяет описать элементы в виде электронной HTML таблицы. Чтобы создать связь с частью составного элемента, использовались порты. В рамках формирования описания на языке Dot создается ориентированный граф, где вершина – это адрес компоненты в памяти, а ребро – указатель на другую компоненту.

Исходный код алгоритма, предоставляемый пользователем, не содержит в себе логики, по выводу содержимого памяти – реальных адресов, формируемых во время выполнения алгоритма. Поэтому исходный алгоритм дополнялся специальными процедурами. Кроме того, все операторы по обработке динамических структур заменялись процедурами, которые выводят результаты этих операторов в виде записей на языке JSON. Такой формат данных является универсальным и позволяет обрабатывать результаты вне зависимости от языка программирования.

Для реализации была использована библиотека LibTooling [10] и входящий в неё предметно-ориентированный язык Matchers [11]. Она предоставляет доступ к абстрактному синтаксическому дереву компилятора Clang, которое содержит определение нужных операторов, и позволяет произвести их замену. Язык Matcher позволяет произвести необходимые модификации кода на основе декларативных описаний. Модифицированный таким образом исходный код можно выполнить и получить представление динамической структуры в памяти на каждом шаге исходного алгоритма. Для этого модифицированный код собирается и линкуется с библиотекой, включающей в себя специальные процедуры. Для описания процесса сборки используется конфигурационный файл системы генерации сборок CMake.

После сборки полученный исполняемый файл отправляется в отладчик GDB [12]. Поскольку в стандартном виде отладчик не разделяет вывод внутренних команд от вывода программы, был использован входящий в него машинно-ориентированный интерфейс GDB-MI [13]. В нем результаты вывода команд структурированы, и каждая запись начинается со специального разделителя.

Работа модифицированного кода требует поддержки перегрузки функций, а также создания сложных типов. Для решения данной проблемы вызовы процедур в отладчике дополнялись преобразованием их типов в соответствии с их сигнатурами. Эти преобразования основаны на типах, полученных из таблицы символов в компиляторе Clang.

После выполнения очередного шага алгоритма в отладчике результаты работы алгоритма отражаются в памяти в виде описания на языке Dot. Все элементы описания формиро-

ваны с использованием классов, разработанных с использованием паттерна проектирования «строитель». Это позволяет поэтапно сконструировать описание, а также абстрагироваться от деталей реализации и синтаксических особенностей языка Dot.

Каждое изменение памяти приводит к созданию нового описания. В результате после трансформации описаний средствами Graphviz получится набор SVG документов. Постепенное преобразование одного такого файла в другой позволяет реализовать анимацию изменения динамической структуры данных. Преимуществом данного подхода, к реализации анимации, является минимальный объем кода, необходимый для корректной работы. Кроме того, SVG имеет широкую поддержку в современных браузерах, что позволило реализовать масштабирование в визуализаторе без дополнительного программного кода.

В результате проектирования был разработан пользовательский интерфейс, представленный на рис. 1.

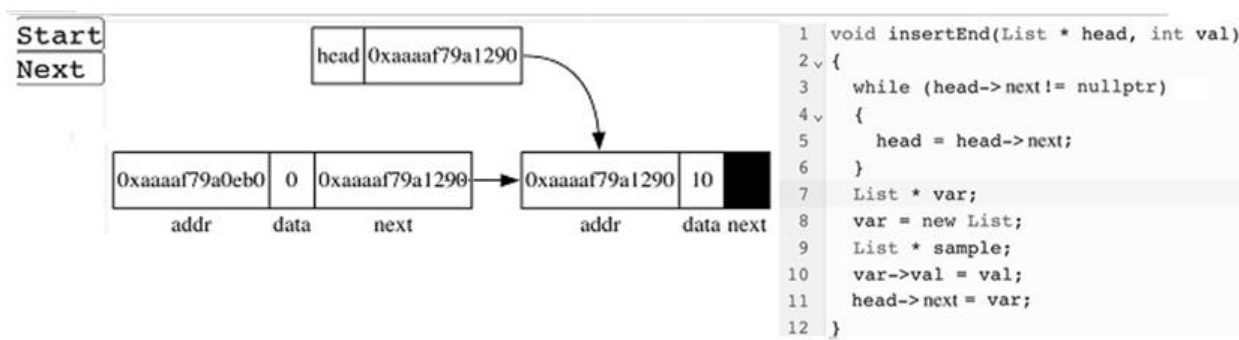


Рис. 1. Пользовательский интерфейс.

Интерфейс состоит из трех основных компонентов: меню управления ходом визуализации, окно с результатами визуализации алгоритма на очередном шаге и редактор кода для изменения алгоритма в ходе работы с программой. Текущая строка в редакторе кода соответствует текущей строке алгоритма, находящейся в исполнении.

На рисунке представлено создание списка из двух элементов со значениями ноль и десять, и указателем head на конец списка.

Визуализация алгоритмов продемонстрирована пошагово на рисунках 23. В первой строке программного кода создается указатель head (см. рис. 2). Визуализатор отображает реальный адрес данного указателя (0xaaaaafa6abeb0). Значение первого элемента списка равно нулю (поле data). Адрес следующего элемента (поле next) равен null. На рисунке адрес следующего элемента обозначен прямоугольником черного цвета. (см. рис. 2)

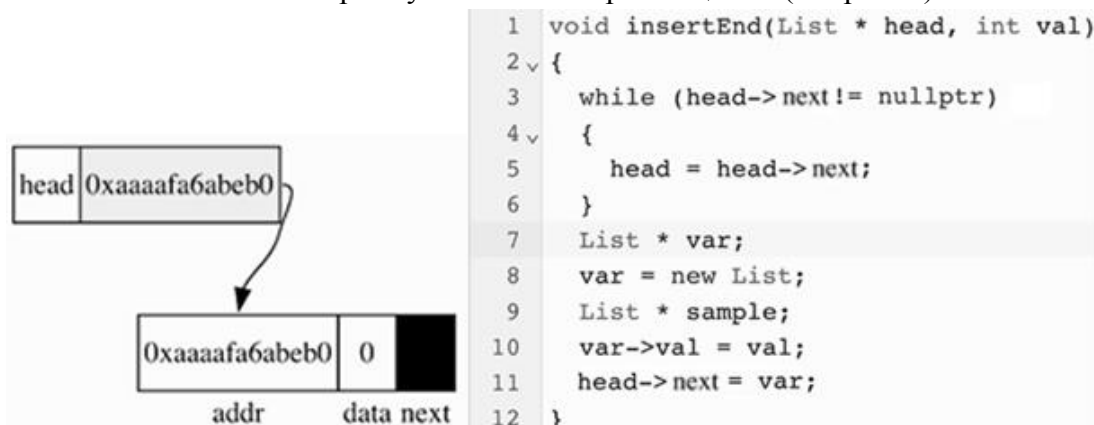


Рис. 2. Результат выполнения программного кода строки 1-6.

Для последующего добавления нового элемента в конец списка, создается указатель var для структуры List. Визуализация данного шага (строка 7 программного кода) представлена на рис. 3.

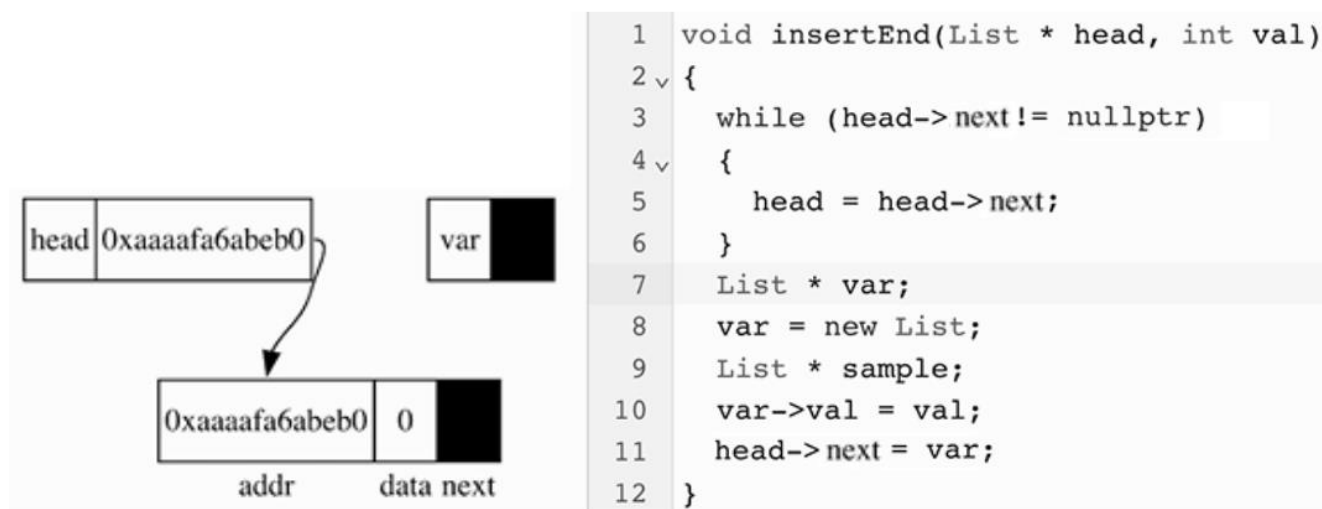


Рис. 3. Результат выполнения действия в строке 7 программного кода.

Разработанное пошаговое визуальное представление алгоритмов обработки однонаправленных списков (описание узла односвязного списка с отображением реальных адресов размещения элементов в динамической памяти; вставка узла в середину списка, в конец списка), может быть использовано в учебном процессе при освоении студентами динамических структур данных – однонаправленных списков.

### Библиографический список

1. НОУ «Интуит». Структуры и алгоритмы компьютерной обработки данных. Лекция 30: Динамические структуры данных: однонаправленные и двунаправленные списки. [Электронный ресурс]. – URL: <https://intuit.ru/studies/courses/648/504/lecture/11456> (дата обращения: 15.06.23).
2. Окулов. С. М. Основы программирования. М.: ЮНИМЕДИАСТАЙЛ, 2002. 424 с.
3. Симонова Е.В. Линейные динамические структуры данных в C#. Списки. М.: Изд-во Самарского университета, 2017. 48 с.
4. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона + CD / Пер. с англ. Ткачев Ф.В. М.: ДМК Пресс, 2010. 272 с.
5. Лафоре Р. Структуры данных и алгоритмы в Java. Классика Computers Science. 2-е изд. СПб.: Питер, 2013. 704 с.
6. Lecture Notes for Data Structures and Algorithms. [Электронный ресурс]. – URL: <https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf> (дата обращения 15.06.23).
7. Visualising data structures and algorithms through animation – VisuAlgo. [Электронный ресурс]. – URL: <https://visualgo.net/en> (дата обращения 15.06.23).
8. An open graph visualization system and its applications to software engineering. [Электронный ресурс]. – URL: <https://graphviz.org/documentation/GN99.pdf> (дата обращения 15.06.23).

9. Dot language. [Электронный ресурс]. – URL: <https://graphviz.org/doc/info/lang.html> (дата обращения 15.06.23).
10. Clang: LibTooling. [Электронный ресурс]. – URL: <https://clang.llvm.org/docs/LibTooling.html> (дата обращения 15.06.23).
11. AST Matcher Reference. [Электронный ресурс]. – URL: <https://clang.llvm.org/docs/LibASTMatchersReference.html> (дата обращения 15.06.23).
12. GDB: The GNU Project Debugger. [Электронный ресурс]. – URL: <https://www.sourceware.org/gdb/> (дата обращения 15.06.23).
13. GDB/MI (Debugging with GDB). [Электронный ресурс]. – URL: [https://sourceware.org/gdb/onlinedocs/gdb/GDB\\_002fMI.html](https://sourceware.org/gdb/onlinedocs/gdb/GDB_002fMI.html) (дата обращения 03.05.23).

### References

1. NOW "Intuit". Structures and algorithms of computer data processing. Lecture 30: Dynamic data structures: unidirectional and bidirectional lists. [Electronic resource]. – URL: <https://intuit.ru/studies/courses/648/504/lecture/11456> (accessed on June 15, 2023). (In Russ.).
2. *Okulov C.M.* Fundamentals of programming. M.: UNIMEDIASTAIL, 2002. 424 p. (In Russ.).
3. *Simonova E.V.* Linear dynamic data structures in C#. Lists. M.: Publishing House of Samara University, 2017. 48 p. (In Russ.).
4. *Virt, N.* Algorithms and Data Structures. New Version for Oberon + CD / Translated by Tkachev, F.V. Moscow: DMK Press, 2010. 272 p. (In Russ.).
5. *Lafore, R.* Data Structures and Algorithms in Java. Classic Computers Science. 2nd edition. St. Petersburg: Piter, 2013. 704 p. (In Russ.).
6. Lecture Notes for Data Structures and Algorithms. [Electronic resource]. – URL: <https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf> (accessed on June 15, 2023).
7. Visualising data structures and algorithms through animation – VisuAlgo. [Electronic resource]. – URL: <https://visualgo.net/en> (accessed on June 15, 2023).
8. An open graph visualization system and its applications to software engineering. [Electronic resource]. – URL: <https://graphviz.org/documentation/GN99.pdf> (accessed on June 15, 2023).
9. Dot language. [Electronic resource]. – URL: <https://graphviz.org/doc/info/lang.html> (accessed on June 15, 2023).
10. Clang: LibTooling. [Electronic resource]. – URL: <https://clang.llvm.org/docs/LibTooling.html> (accessed on June 15, 2023).
11. AST Matcher Reference. [Electronic resource]. – URL: <https://clang.llvm.org/docs/LibASTMatchersReference.html> (accessed on June 15, 2023).
12. GDB: The GNU Project Debugger. [Electronic resource]. – URL: <https://www.sourceware.org/gdb/> (accessed on June 15, 2023).
13. GDB/MI (Debugging with GDB). [Electronic resource]. – URL: [https://sourceware.org/gdb/onlinedocs/gdb/GDB\\_002fMI.html](https://sourceware.org/gdb/onlinedocs/gdb/GDB_002fMI.html) (accessed on May 3, 2023).

## **VISUALIZER OF ALGORITHMS FOR WORKING WITH DYNAMIC DATA STRUCTURES (UNIDIRECTIONAL LISTS)**

*Volkov Ilya I., Romashkina Tatiana V.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, t\_obuch\_11@mail.ru

The present work focuses on algorithms for processing dynamic data structures using the example of a singly linked list. It describes methods for graphical interpretation of operations on singly linked lists. A visualization approach for algorithms working with singly linked lists is proposed, as the abstract representation of such data structures makes understanding and perception of their operations challenging. Algorithms for processing singly directed lists are visualized in accordance with the lines of the user's program code and with the display of the real addresses of the list elements in dynamic memory. The design and development of an application for visualizing algorithms working with singly linked lists are explored, based on a client-server architecture. The application is created using Golang, TypeScript, Dot and C++ programming languages, ReactJS and LibTooling libraries.

**Keywords:** dynamic data structures, visualizer of algorithms for working with unidirectional lists, client-server application.

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ АВТОМАТИЧЕСКОЙ ФИКСАЦИИ ПРИСУТСТВИЯ УЧАЩИХСЯ НА ЗАНЯТИЯХ В УНИВЕРСИТЕТЕ**

*Гвягянен Артем Алексеевич, Василюк Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, a.hauhanen@gmail.com

В данной статье рассмотрены аналоги информационных систем поддержки автоматической фиксации учащихся на занятиях в университете, выделены преимущества и недостатки этих систем, и на их основе выявлены и определены функциональные требования к проектируемой информационной системе (ИС). ИС планируется использовать для фиксации студентов на занятиях в университете, а также оперативной генерации и представления отчетов о посещаемости для преподавателей. Для моделируемой ИС были определены средства для проектирования, набор инструментальных средств для разработки, а также средство хранения данных. Помимо этого, было определено средство для визуализации отчетов о посещаемости. Работа содержит описание логики работы информационной системы, а также примерный вариант работы системы генерации отчетов. Спроектированная система будет интересна высшим учебным заведениям, которые заинтересованы в организации системы поддержки автоматической фиксации учащихся на занятиях.

Ключевые слова: информационная система (ИС), фиксация учащихся, отчеты о посещаемости, высшие учебные заведения, uml-диаграммы.

### **Введение**

В современном образовательном процессе важным элементом является контроль посещаемости учащихся на занятиях. Оценка активности студентов, повышение внимания к проблемным группам, а также разработка стратегии по улучшению качества образования зависят от точности и своевременности получения информации о посещении занятий для преподавателей и администрации.

Ручной процесс фиксации, к сожалению, может быть неэффективным, трудоемким и подверженным ошибкам. В связи с этим, создание информационной системы поддержки автоматической фиксации присутствия учащихся на занятиях в университете является актуальной задачей. Эта система позволит автоматически отслеживать посещаемость студентов на занятиях, а также генерировать отчеты для преподавателей и администрации университета.

Цель настоящей работы состоит в проектировании и документировании информационной системы поддержки автоматической фиксации присутствия учащихся на занятиях в университете.

### **Анализ и сравнение существующих аналогов ИС**

Рассмотрим примеры некоторых информационных систем автоматической фиксации.

#### **1. Анализ видеоданных [1]**

Систему спроектировал И. А. Симагин, студент института математики и информационных технологий. ИС работает на основе анализа данных с онлайн-камер, установленных в учебных аудиториях. Среди преимуществ можно выделить следующие: система работает ав-

томатически и не требует дополнительного времени преподавателя на отметку присутствия студентов, система относительно дешевле, чем использование СКУД для фиксации посещаемости. Из недостатков можно выделить ограничение на точность, так как система может некорректно работать в условиях низкой освещенности и подобных, а также сложность внедрения такой системы.

## 2. QR-код [2]

Система спроектирована и представлена командой специалистов Московского автомобильно-дорожного государственного технического университета. ИС работает посредством сканирования QR-кодов, которые генерирует преподаватель. Преимущества данной системы в удобстве использования, низкие затраты на установку и поддержку, а также в легкости интегрирования с другими платформами электронного обучения, которые используются в учебных заведениях.

## 3. Bluetooth-модуль [3]

Систему спроектировала К.А. Голубева, студентка Санкт-Петербургского политехнического университета Петра Великого. Данная система является приложением для отслеживания посещаемости пользователей с использованием Bluetooth. Среди преимуществ можно выделить следующие: удобство использования системы, установка и поддержка не требует высоких затрат от учебного заведения, система не имеет сложностей при интегрировании с другими платформами электронного обучения, которые используются в учебных заведениях.

На основе анализа существующих систем сформулированы следующие задачи, которые необходимы в проектируемой ИС:

- легкость внедрения ИС;
- низкая стоимость внедрения и содержания системы;
- хранение в БД информации о расписании, преподавателях и учащихся;
- система администраторов и преподавателей;
- система генерации отчетов о посещаемости.

## **Выбор case-средства, средств разработки и хранения данных**

Для разработки информационных систем необходимо использовать программные средства, которые поддерживают объектно-ориентированный подход и включают в себя CASE-средства. Одним из основных инструментов является язык моделирования UML, с помощью которого будут построены диаграммы, необходимые для успешного выполнения проектирования системы.

В качестве инструмента проектирования был выбран StarUML [4]. Данный инструмент используется для создания диаграмм UML, включая диаграммы классов, диаграммы последовательностей, диаграммы состояний и другие. Среди основных преимуществ StarUML можно выделить следующие: стоимость, так как это бесплатный инструмент, а также удобный пользовательский интерфейс, что позволяет значительно упростить процесс создания и редактирования моделей систем.

Для реализации ИС автоматической фиксации планируется использовать платформы Arduino. Разработка программного обеспечения в Arduino осуществляется посредством использования языка программирования C++. Arduino IDE был выбран в качестве среды разработки, которая представляет собой наиболее простой и удобный интерфейс для написания, загрузки и отладки кода для микроконтроллеров Arduino.

Для разработки системы генерации отчетов для преподавателей и администраторов высших учебных заведений было принято использовать среду разработки Visual Studio, язык программирования C#. Для визуализации данных было принято использовать Telegram-бота.

В качестве средства хранения данных была выбрана PostgreSQL – удобная объектно-реляционная система управления базами данных.

### Проектирование информационной системы

Участвующими лицами в процессе фиксации являются преподаватель и студенты. Преподаватель может как подготовить устройство, так и оставить запрос на получение отчета о посещаемости. При выборе первого варианта, преподаватель также должен будет сканировать две необходимые ему карты или два брелока, в которые встроен соответствующий чип. Преподаватель имеет чип, в котором хранится идентификатор, соответствующий учебной дисциплине, а также чип с идентификатором самого преподавателя. Эти чипы используются для определения необходимого занятия в БД. После сканирования чипов данные отправляются в БД.

Если же преподаватель решил оставить запрос на получения отчета о посещаемости, то происходят в обязательном порядке два сопутствующих этому события. Первым событием является генерация самого отчета при помощи средств разработки и взаимодействия программы на языке программирования C# и БД PostgreSQL. Вторым событием является непосредственная отправка отчета преподавателю, которая осуществляется при помощи того же программного компонента, что и в первом событии, а также Telegram-бота, в котором изначально преподаватель и составляет запрос.

Студентам университета необходимо совершать лишь одно простое действие. Так как система разрабатывается на примере ПГНИУ, то принято считать, что у студентов уже есть карточки с чипом, хранящие уникальные идентификаторы учащихся. Студентам необходимо сканировать чип, находящийся в их студенческих билетах к специальному устройству у преподавателя, после чего будут внесены изменения в БД и принято считать, что студент оставил метку о присутствии на занятии (см. рис. 1).

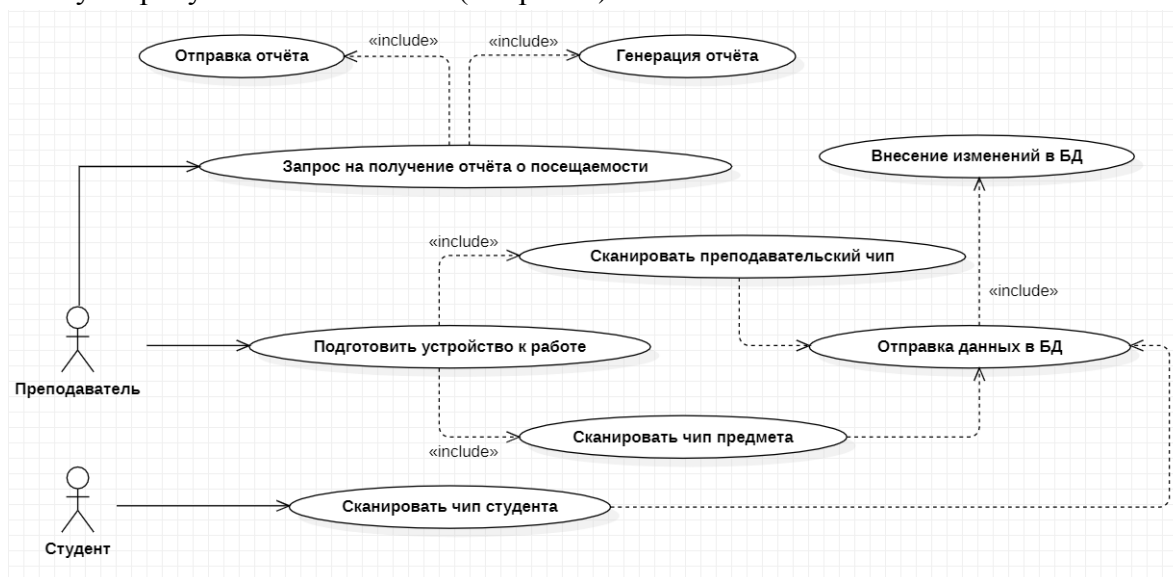


Рис. 1. Диаграмма прецедентов для преподавателя и студента

Система включает в себя разграничение прав и ролей пользователей для корректной работы ИС.



Права преподавателей распространяются только на редактирование отметок о присутствии студентов, связанные с теми учебными группами и дисциплинами, с которыми эти преподаватели работают непосредственно.

Администратор должен иметь права, позволяющие ему изменять расписание занятий, редактировать информацию о студентах и преподавателей, а также удалять устаревшие записи. Исключением можно считать редактирование меток о присутствии студентов на занятиях, такое право у администратора должно отсутствовать ввиду возможных несовпадений в списках о присутствии у преподавателей и реальными списками в БД.

Ниже представлен примерный вариант использования Telegram-бота для генерации отчета о посещаемости занятий для преподавателей (см. рис. 2).

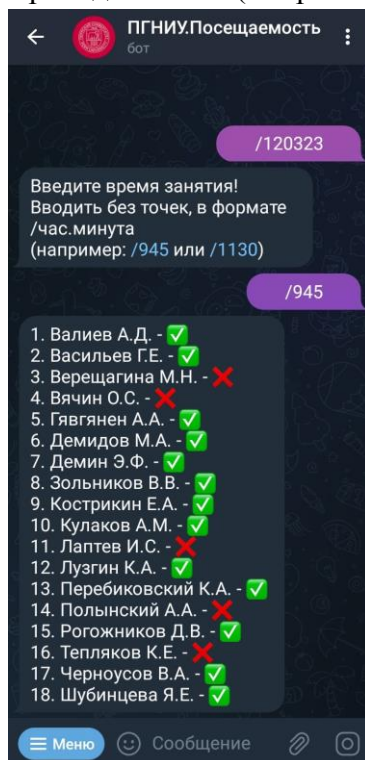


Рис. 2. Список посещаемости занятия

## Заключение

Спроектированная система будет удобна для использования в университетах преподавателями, так как она позволит быстро и удобно фиксировать посещаемость занятий студентами, а также оперативно получать отчеты для более эффективной организации учебного процесса.

## Библиографический список

1. Разработка сервиса автоматической фиксации посещаемости занятий через анализ видеоданных // Elibrary [Электронный ресурс]. – URL: <https://www.elibrary.ru/item.asp?id=46126938> (дата обращения 28.05.2023)
2. Система посещаемости студентов с использованием QR-кода: // Elibrary [Электронный ресурс]. – URL: <https://www.elibrary.ru/item.asp?id=44627167> (дата обращения: 20.05.2023)
3. Разработка приложения для контроля посещаемости с использованием модуля Bluetooth // Elibrary [Электронный ресурс]. – URL: <https://www.elibrary.ru/item.asp?id=45781350> (дата обращения 28.05.2023)

4. Проектирование информационной системы с применением CASE-инструмента StarUML // Elibrary [Электронный ресурс]. – URL: <https://www.elibrary.ru/item.asp?id=46126938> (дата обращения 28.05.2023)

### References

1. Development of a service for automatic tracking of attendance during classes through video data analysis // Elibrary [Electronic resource]. – URL: <https://www.elibrary.ru/item.asp?id=46126938> (in Russ)
2. Student attendance system using QR-code // Elibrary [Electronic resource]. – URL: <https://www.elibrary.ru/item.asp?id=44627167> (in Russ)
3. Development of attendance control application using Bluetooth module // Elibrary [Electronic resource]. – URL: <https://www.elibrary.ru/item.asp?id=45781350> (in Russ)
4. Designing an information system using StarUML CASE tool // Elibrary [Electronic resource]. – URL: <https://www.elibrary.ru/item.asp?id=46126938> (in Russ)

## DESIGNING AND DOCUMENTING AN INFORMATION SYSTEM TO SUPPORT THE AUTOMATIC RECORDING OF THE PRESENCE OF STUDENTS IN THE CLASSROOM AT THE UNIVERSITY

*Gyavgyanen Artem A., Vasiluk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [a.hauhanen@gmail.com](mailto:a.hauhanen@gmail.com)

This article examines analogs of information systems to support automatic fixation of students in classes at the university, highlights the advantages and disadvantages of these systems, and based on them, the functional requirements for the designed information system are identified and defined. The system is planned to be used to record students in classes at the university, as well as to promptly generate and submit attendance reports for teachers. For the simulated system, design tools, a set of development tools, as well as a data storage facility were identified. In addition, a tool for visualizing attendance reports was defined. The work contains a description of the logic of the information system, as well as an approximate version of the report generation system. The designed system will be of interest to higher educational institutions that are interested in organizing a system to support the automatic fixation of students in the classroom.

Keywords: information system, fixation students in the classroom, attendance reports, higher education institutions, uml diagrams.

## ОПТИМИЗАЦИЯ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ВСТРОЕННОГО ДИАГНОСТИРОВАНИЯ ПЛИС НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА STM32F401RE

*Дементьев Михаил Александрович, Данилова Екатерина Юрьевна*

Пермский государственный национальный исследовательский университет, 614990,  
Россия, г. Пермь, ул. Букирева, 15, mixail.demen.2002@gmail.com

В статье проводится оптимизация по памяти кода, предназначенного для построения диагностической последовательности, которая в свою очередь используется для обнаружения единичных константных отказов ПЛИС. ПЛИС – программируемая логическая интегральная схема, содержащая большое количество логических элементов. При этом разработчику необходимо самому определять взаимосвязи, назначение выводов и пути прохождения сигналов. Проведён анализ исходного кода. В качестве основного вектора оптимизации было выбрано сокращение используемой популяцией памяти за счёт упаковки нескольких значений в одно или же использование битовой упаковки. Проведено сравнение занимаемой памяти при выборе определенного типа элемента. Также приведена сравнительная таблица, отражающая занимаемую популяцией память до оптимизации и после. Сделаны вывод об эффективности использования алгоритмом памяти в применении к данной задаче.

Ключевые слова: ПЛИС, диагностирование, генетический алгоритм, оптимизация, оптимизация по памяти, сжатие данных, битовая упаковка, используемая память.

### **Введение**

ПЛИС является электронным компонентом, обладающим возможностью быть перепрограммированным для выполнения другой задачи. ПЛИС представляет собой массив логических блоков, соединённых между собой программируемыми маршрутами, позволяющими настраивать взаимосвязи между блоками [1].

ПЛИС – это микросхемы, в которых производителем не определены связи между логическими элементами. Разработчик самостоятельно определяет взаимосвязи, назначает выводы и пути нахождения сигналов. В этом заключается большой плюс ПЛИС, так как следствием такого подхода становятся исключительно гибкая архитектура и быстродействие полученной электронной схемы. Но за это приходится платить увеличением времени разработки устройства [1].

Именно из-за гибкости и быстродействия ПЛИС применяется в системах реального времени и отказоустойчивых системах.

Таким образом, для диагностирования константного отказа необходимо построить диагностическую последовательность. Данная диагностическая последовательность может быть найдена при помощи генетического алгоритма, который является эвристическим алгоритмом. ГА подробно описан в [2].

В статье рассмотрена оптимизация по памяти уже имеющейся программы генетического алгоритма, строящего диагностическую последовательность, которая в свою очередь предназначена для обнаружения константных отказов [3]. Данная программа способна выполняться на микроконтроллере STM32F401RE. Для запуска программы используется среда Proteus, которая эмулирует работу микроконтроллера.

В дальнейшем STM32F401RE должна внедряться в ПЛИС для внутреннего диагностирования, что позволит выявлять отказы и по возможности проводить реконфигурацию ПЛИС для её дальнейшего использования.

Поскольку микроконтроллер, предназначенный для встроенного диагностирования, имеет ограничение по памяти, из этого следует то, что появляется необходимость сократить используемую генетическим алгоритмом память. В дальнейшем это позволит одному микроконтроллеру диагностировать работу большего количества целевых функций ПЛИС, вследствие чего сократится количество используемых микроконтроллеров для выполнения поставленной задачи.

### Оптимизация

В работе [4] был приведен перечень вариантов реализации оптимизации генетического алгоритма по памяти. Анализ эффективности работы алгоритма приведен в [4]. Таким образом, одним из вариантов является сокращение использования памяти популяцией.

Популяция – это совокупность особей определенного вида. Популяция представляет собой двумерный массив указателей, который хранит лучшие особи, сформировавшиеся в процессе выполнения генетического алгоритма. Каждое новое поколение создается на основе предыдущего поколения путем скрещивания и мутации. Созданные потомки и мутированные индивиды помещаются в единый массив особей `new_gen` – динамический массив указателей, предназначенный для хранения особей. Размерность данного массива зависит от количества индивидов в текущей популяции; количества потомков, которые будут сформированы в результате скрещивания; количество индивидов, которые будут сформированы в результате мутации. Следовательно, необходимо выделять память под каждую особь, которая будет храниться в двумерном массиве указателей.

Одним из вариантов сокращения используемой памяти, занимаемой популяцией, является использование производного класса, реализующего битовую упаковку.

В [4] элементы каждой особи хранятся в динамическом массиве указателей, размерность которого зависит от длины диагностической последовательности. Таким образом, необходимо упаковывать каждую особь в битовые последовательности.

### Битовая упаковка

Сжатие данных предназначено для уменьшения количества битов, необходимых для хранения или передачи данных. Подробнее о сжатии данных описано в [5]. Одним из вариантов сжатия является битовая упаковка.

Битовая упаковка – это метод, предназначенный для оптимизации хранения данных, позволяющий уменьшать объем памяти, занимаемой данными, путём упаковки нескольких значений в одно целое.

`BitPack` – производный класс, который используется для упаковки, распаковки и хранения коротких целочисленных значений в памяти с помощью битовой маски. Каждый элемент данных занимает определенное количество битов. Число этих битов напрямую зависит от  $n$ , где  $n$  – это количество аргументов исходной функции. Таким образом, каждый элемент хранится в виде битовой маски, которые, в свою очередь, образуют общую битовую последовательность. Размер данной битовой последовательности зависит от количества выделенных блоков. Занимаемая память данными блоками зависит от выбранного типа элемента. Пусть этим типом является *char/unsigned char*, каждый элемент которого занимает 1 байт.

Ранее упоминалось, что количество битов, необходимых для упаковки значений зависит от количества входных аргументов. То есть, при  $n = 3$  особь принимает значения от 0 до  $2^n - 1$ , то есть от 0 до 7, следовательно каждый элемент особи может быть упакован в 3 бита.

При  $n = 4$  значения упаковываются в 4 бита. Но уже при  $n = 5$  нецелесообразно использовать упаковку значений в 1 байт. Причиной этому служит то, что 1 блок будет хранить 1 элемент.

Рассмотрим упаковку значений в 3 бита, то есть количество аргументов исходной функции равняется 3. До разработки производного класса *BitPack* каждый элемент хранился в динамическом массиве указателей типа *short*, таким образом, каждый элемент особи занимал 2 байта памяти. Следовательно, при  $l = 3$ , где  $l$  – это длина диагностической последовательности, занимаемая особью память будет составлять  $2 \cdot 3 = 6$  байт. Разработанный класс позволяет сократить данное число до 2 байт. Полученный выигрыш достигается за счет того, что каждый элемент занимает 3 бита, следовательно, в 1 байт можно упаковать 2 целых значения. Но стоит отметить, что для хранения 3 элементов потребуется 9 бит. Однако, обычно в компьютерных системах минимальная адресуемая единица памяти является байт, который приравнивается к 8 битам. Поэтому хранение 3 элементов потребует, как минимум двух байтов. В свою очередь, это всё равно выгоднее, чем использования типа элемента *short*.

При  $l = 4$  количество элементов особи будет равно 4. Следовательно, при применении упаковки в 3 бита под хранения данных выделяется 2 байта памяти. При  $l = 5$  также выделяется 2 байта памяти, то есть 5 элементов занимает 15 бит памяти. Но при  $l = 6$  уже необходимо выделить 3 байта памяти. В табл. 1 представлено подробное описание необходимой памяти для хранения данных, количество бит на одно значение равняется 3 битам.

Таблица 1. Расчет выделенной памяти при  $n = 3$

Длина диагностической последовательности	Память, занимаемая без упаковки	Занимаемая особью память	Выделенная память под хранение особи
3	6 байт	$9/8 = 1,125$ байт	2 байта
4	8 байт	$12/8 = 1,5$ байт	2 байта
5	10 байт	$15/8 = 1,875$ байт	2 байта
6	12 байт	$18/8 = 2,25$ байт	3 байта
7	14 байт	$21/8 = 2,625$ байт	3 байта
8	16 байт	$24/8 = 3$ байта	3 байта

В случае, когда количество аргументов исходной функции равняется 4, тогда каждый элемент принимает одно из шестнадцати значений, то есть каждый элемент может быть представлен в виде числа от 0 до 15. Следовательно, каждый элемент может быть упакован в 4 бита. В табл. 2 представлены данные использования битовой упаковки при  $n = 4$ .

Таблица 2. Расчет выделенной памяти при  $n = 4$

Длина диагностической последовательности	Память, занимаемая без упаковки	Занимаемая особью память	Выделенная память под хранение особи
3	6 байт	$12/8 = 1,5$ байт	2 байта
4	8 байт	$16/8 = 2$ байт	2 байта
5	10 байт	$20/8 = 2,5$ байт	3 байта
6	12 байт	$24/8 = 3$ байта	3 байта
...	...	...	...
14	28 байт	$56/8 = 7$ байт	7 байт
15	30 байт	$60/8 = 7,5$ байт	8 байт
16	32 байт	$64/8 = 8$ байт	8 байт

Ранее упоминалось, что нецелесообразно использовать битовую упаковку в 1 байт в случае  $n = 5$ . Причина, по которой это не имеет значения, заключается в том, что одно значение будет храниться в 1 байте. Поэтому нужно хранить значения в битовых блоках типа *short* или *integer*.

Но и при  $n = 3$  также нецелесообразно. Ведь, при упаковке в *short* выделяется 2 байта, а минимальная длина диагностической последовательности равняется 3. То есть, при  $l = 3$  будет также выделено 2 байта памяти. Но при  $l = 5$  при упаковке в *short* выделяется 2 байта, а при упаковке в *char* выделяется 3 байта.

В табл. 3 представлено соотношение количества входных аргументов к количеству возможных упакованных элементов при упаковке в *short* и *integer*.

Таблица 3. Сравнение вместимости типа *short* и *integer*

Количество аргументов исходной функции	Количество элементов маски	Количество элементов упакованных в <i>short</i>	Количество элементов упакованных в <i>integer</i>
5	5	$16/5 \approx 3$	$32/5 \approx 6$
6	6	$16/6 \approx 2$	$32/6 \approx 5$
7	7	$16/7 \approx 2$	$32/7 \approx 4$
8	8	$16/8 = 2$	$32/8 = 4$
9	9	$16/9 \approx 1$	$32/9 \approx 3$

### Занимаемая популяцией память

Предположим, что существует некоторая популяция, состоящая из 50 особей. Длина особи зависит от длины диагностической последовательности. В табл. 4 представлена зависимость занимаемой популяцией памяти при упаковке от количества входных аргументов и длины диагностической последовательности.

Таблица 4. Выделенная популяции памяти

Количество входных аргументов	Длина диагностической последовательности	Количество битовых блоков в <i>BitPack</i>	Необходимое количество памяти при упаковке в <i>short</i>	Необходимое количество памяти при упаковке в <i>BitPack</i>
3	3	1 блок	$2 \cdot 3 \cdot 50 = 300$	$2 \cdot 1 \cdot 50 = 100$
3	4	1 блок	$2 \cdot 4 \cdot 50 = 400$	$2 \cdot 1 \cdot 50 = 100$
3	5	1 блок	$2 \cdot 5 \cdot 50 = 500$	$2 \cdot 1 \cdot 50 = 100$
4	3	1 блок	$2 \cdot 3 \cdot 50 = 300$	$2 \cdot 1 \cdot 50 = 100$
4	4	1 блок	$2 \cdot 4 \cdot 50 = 400$	$2 \cdot 1 \cdot 50 = 100$
4	5	2 блока	$2 \cdot 5 \cdot 50 = 500$	$2 \cdot 2 \cdot 50 = 200$
5	3	1 блок	$2 \cdot 3 \cdot 50 = 300$	$2 \cdot 1 \cdot 50 = 100$
5	4	2 блока	$2 \cdot 4 \cdot 50 = 400$	$2 \cdot 2 \cdot 50 = 200$
5	5	2 блока	$2 \cdot 5 \cdot 50 = 500$	$2 \cdot 2 \cdot 50 = 200$
...	...	...	...	...
6	6	3 блока	$2 \cdot 6 \cdot 50 = 600$	$2 \cdot 3 \cdot 50 = 300$
6	7	4 блока	$2 \cdot 7 \cdot 50 = 700$	$2 \cdot 4 \cdot 50 = 400$

### Заключение

Предложенный метод позволяет существенно сократить объем памяти, занимаемый популяцией. Точное значение сокращения зависит от количества входных аргументов целевой функции, а также параметров генетического алгоритма. Например, при  $n = 4$  и  $l = 4$ , опираясь на табл. 4, можно сказать, что используемая популяцией память будет сокращена в 4 раза.

### Библиографический список

1. Что такое ПЛИС и с чего начать изучение FPGA Altera [Электронный ресурс]. – URL: <https://el-ra.ru/articles/izucheniye-fpga-altera/> (дата обращения: 17.07.2023).
2. Danilova E.Y., Kovylyaev D., Gorodilov A.Y. Advanced Genetic Algorithm for the Embedded FPGA Logic Diagnostic // 2021 International Conference on Information and Digital Technologies (IDT). Zilina, Slovakia. 2021.
3. Danilova E.Y. “Comparison of the genetic algorithms to build a diagnostic tree for diagnosing single stuck-at failures,” // Proceeding of The International Conference on “Information and Digital Technologies 2019”. Zilina, 2019. – pp. 93-97.
4. Danilova E.Y., Kovylyaev D. Embedded FPGA Diagnostics of Multiple Logic Elements using the Stm32 Microcontroller // 2023 Conference of Russian Young Researchers in Electrical and Electronic Engineering (2023 ElConRus). St. Petersburg, Russia. 2023.
5. Matt Mahoney Data Compression Explained [Электронный ресурс]. – URL: <https://www.mattmahoney.net/dc/dce.html> (дата обращения: 17.07.2023).

### References

1. Chto takoe PLIS i s chego nachat' izucheniye FPGA Altera [What FPGAs are and where to start learning about Altera FPGAs]. [Electronic resource]. – URL: <https://el-ra.ru/articles/izucheniye-fpga-altera/> (accessed 17.07.2023)
2. Danilova E.Y., Kovylyaev D., Gorodilov A.Y. Advanced Genetic Algorithm for the Embedded FPGA Logic Diagnostic, In: 2021 International Conference on Information and Digital Technologies (IDT), Zilina, Slovakia. 2021. Doi: 10.1109/IDT52577.2021.9497584
3. Danilova E.Y. “Comparison of the genetic algorithms to build a diagnostic tree for diagnosing single stuck-at failures,” In: Proceeding of The International Conference on “Information and Digital Technologies 2019”, Zilina, 2019. – pp. 93-97. Doi: 10.1109/DT.2019.8813684
4. Danilova E.Y., Kovylyaev D. Embedded FPGA Diagnostics of Multiple Logic Elements using the Stm32 Microcontroller, In: 2023 Conference of Russian Young Researchers in Electrical and Electronic Engineering [2023 ElConRus], St. Petersburg, Russia, 2023.
5. Matt Mahoney Data Compression Explained [Electronic resource]. – URL: <https://www.mattmahoney.net/dc/dce.html> (accessed 17.07.2023).

### OPTIMIZATION OF GENETIC ALGORITHM FOR INTEGRATED DIAGNOSTIC ON FPGA BASED ON STM32F401RE MICROCONTROLLER

*Dementyev Mikhail A., Danilova Ekaterina Y.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, mixail.demen.2002@gmail.com

The article performs memory optimization of a code intended for building a diagnostic sequence, which in turn is used to detect single constant FPGA failures. FPGA is a programmable logic integrated circuit containing a large number of logic elements. Also, the developer needs to determine the interconnections, pin assignments and signal paths himself. The source code was analyzed. As the main vector of optimization it was chosen to reduce the memory used by the population by packing several values into one, using bit packing. A comparison of the memory occupied when a certain type of element is selected has been performed. A comparative table showing the memory occupied by the population before and after optimization is also given. Conclusions are drawn about the efficiency of memory usage by the algorithm as applied to this problem.

Keywords: FPGA, diagnostic, genetic algorithm, optimization, memory optimization, data compression, bit packing, memory used.

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВЫДАЧИ СИЗ УЧАЩИМСЯ И ПРЕПОДАВАТЕЛЯМ В ПЕРИОД ЭПИДЕМИЧЕСКОЙ УГРОЗЫ**

*Демидов Матвей Александрович, Василиук Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, demidovmatvey@rambler.ru

В статье рассматривается проектирование и документирование информационной системы выдачи СИЗ (средств индивидуальной защиты) учащимся и преподавателям в период эпидемической угрозы. Автор представляет результаты исследования, в рамках которого были определены требования к системе, ее функциональные возможности, характеристики, архитектура и принцип работы, также описывается процесс разработки графического интерфейса. Разработанная система обеспечивает безопасность для здоровья пользователей, минимизируя контакты и уменьшая риск заражения вирусом, автоматизирует процесс выдачи СИЗ и создает удобство для обучающихся и преподавателей. Она может быть использована в школах, университетах и других учебных заведениях в условиях эпидемической ситуации.

**Ключевые слова:** обеспечение безопасности здоровья в учебных заведениях, эпидемическая угроза, автоматизация выдачи СИЗ.

Системы хранения делятся на хранение данных и хранение вещей. В системы хранения вещей на предприятиях входят системы учета запасов и складские помещения. В системах хранения данных выделяют три уровня, которые включают различные виды накопителей, включая SSD, HDD и облачное хранилище. Обе системы хранения важны для проекта, в котором требуется хранение данных о студентах и преподавателях, а также достаточное пространство для хранения средств индивидуальной защиты.

В качестве примера рассмотрим систему хранения и выдачи на пермской научно-производственной приборостроительной компании, которая занимается разработкой и производством датчиков и систем для навигации, стабилизации и ориентации подвижных объектов [1]. Она имеет следующие преимущества:

- 1) удобна в использовании, так как рабочие быстро получают нужные вещи с помощью электронных тегов и сканеров;
- 2) обеспечивает точное отслеживание местоположения и доступности инструментов и материалов, что снижает количество ошибок при выдаче и улучшает организацию склада;
- 3) обеспечивает улучшенную безопасность, связанную с системой безопасности, что защищает от краж и несанкционированного доступа;
- 4) обеспечивает детальную отчетность о взятии инструментов или материалов для контроля процессов на заводе.

В данный момент на рынке представлено множество продуктов по автоматизации процесса хранения и выдачи. Одними из крупных являются: АРМ «Склад временного хранения», 1С «Склад», продукт «МойСклад». АРМ "Склад временного хранения" – лучше всего для складов временного хранения, но имеет недостаточно гибкую конфигурацию и ограничен-



ные возможности по интеграции с другим ПО. 1С "Склад" – универсальное ПО для управления складской деятельностью, но является дорогим и сложным для небольших предприятий. Продукт "МойСклад" подходит для небольших и средних предприятий, имеет простой интерфейс, доступен из удаленных мест, но не специализирован на складах временного хранения и имеет ограниченные возможности интеграции с другим ПО. Резюмируя, каждая рассмотренная информационная система имеет определенную специализацию, поэтому продукт, предназначенный исключительно для выдачи СИЗ в период эпидемической угрозы, будет более удобным для конечного потребителя.

В РФ отсутствуют определенные требования к количеству одноразовых масок, который работодатель должен выдавать сотруднику, но в рамках профилактических мер по недопущению распространения новой коронавирусной инфекции работодателям рекомендуется обеспечить работников, контактирующих с посетителями, запасом одноразовых масок из расчета: смена маски не реже 1 раза в 3 часа, а также дезинфицирующих салфеток, кожных антисептиков и иных средств. Также не допускается повторное использование одноразовых масок или использование увлажненных масок [2]. Исходя из максимальной недельной нагрузки в размере 40 часов, следует выдавать не менее 14 одноразовых масок в неделю. Аналогично должна производиться смена тканевых масок: не реже 1 раза в 3-4 часа.

Наиболее удобным способом бесконтактной передачи являются постаматы (сеть автоматических почтовых ящиков, которые позволяют получать и отправлять посылки в удобное для вас время): сотрудник ВУЗа, отвечающий за обеспечение студентов и преподавателей СИЗ, сможет их загрузить в отдельную ячейку автоматизированной почтовой станции, после чего обучающийся или преподаватель в удобное для себя время сможет их забрать.

Информационная система хранения и выдачи СИЗ должна: быть надежна и безопасна, быть защищена от несанкционированного доступа, обеспечивать сохранность и целостность данных, быть эффективной, обеспечивать быстрый доступ к информации, иметь высокую производительность, быть способна адаптироваться к изменениям, масштабироваться при необходимости, быть удобной в использовании, простой и интуитивно понятной, быть способной интегрироваться с другими информационными системами и приложениями, быть доступна для пользователей вне зависимости от их текущего местоположения, поддерживать многопользовательскую работу. По мимо этого, требуется организовать обучение пользователей системе и добавить возможность повторного заказа СИЗ в случае, если у студента или преподавателей не получилось забрать их с первого раза, при этом проработать вопрос злоупотребления возможностями системы.

Проектирование системы выполняется посредством CASE-средства по созданию UML-диаграмм под названием draw.io, так как по сравнению с другими приложениями его не нужно загружать и устанавливать, он доступен для использования с любого устройства, имеет более простой интерфейс, универсальный и поддерживает экспорт/импорт диаграмм.

Наиболее подходящим СУБД для нашего проекта является MySQL, так как по сравнению с другими системами она имеет хорошую поддержку от сообщества разработчиков, может обеспечить быстрое выполнение запросов, имеет низкие требования к системным ресурсам, в то время как, например, PostgreSQL имеет излишне высокие требования к системным ресурсам для небольшого проекта.

Оптимальным средством для проектирования БД будет является MySQL Workbench, так как оно разработано специально для работы с СУБД MySQL. Главными ее преимуществами является наличие возможности автоматической генерации скриптов DDL, открытый исходный код и отсутствие платных версий – ПО полностью бесплатно.

Для реализации программы потребуется создать веб-приложение. Для создания сайта наиболее подходящим является использование языка PHP в среде разработки PhpStorm.

Главными преимуществами языка PHP является большая поддержка MySQL, простота синтаксиса, крупное сообщество разработчиков, а использование PhpStorm обуславливается специализацией данной среды для языка PHP.

Диаграмма классов представляет собой графическое представление классов и их отношений в системе. На рис. 1 представлен данный вид диаграммы, описывающий модель работы информационной системы. Она является основой для продолжения работы над проектом.

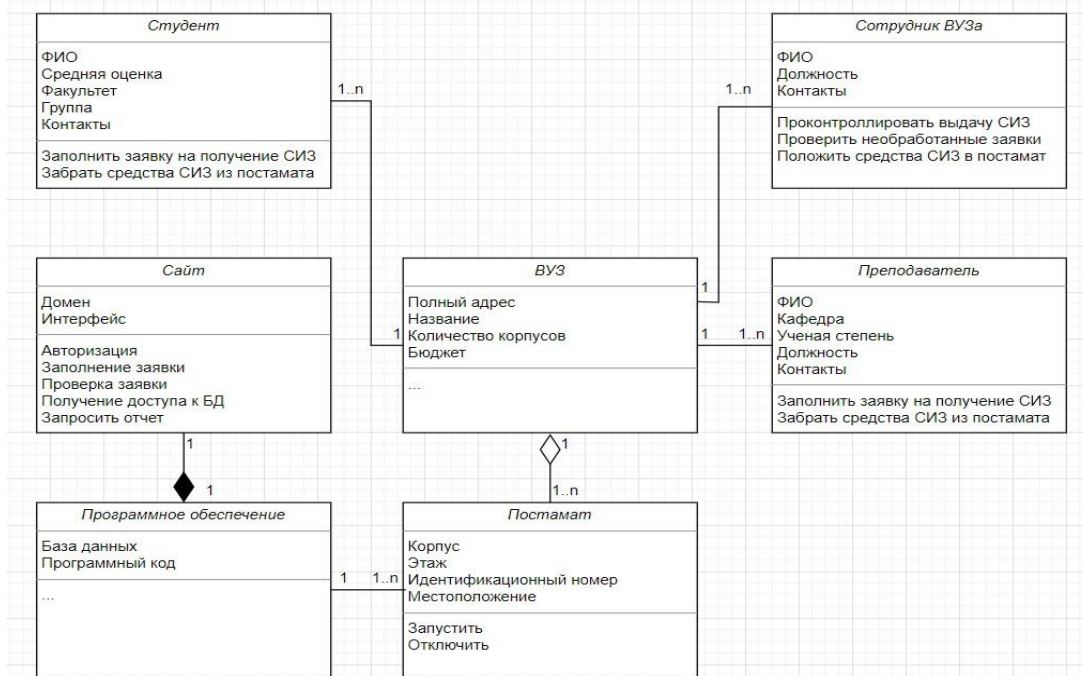


Рис. 1. Диаграмма классов

На рис. 2 представлена спроектированная база данных с помощью MySQL WorkBench. Нам требуется использовать пять соединенных таблиц: преподаватели, студенты, сотрудники, данные об университете и данные о постамате.

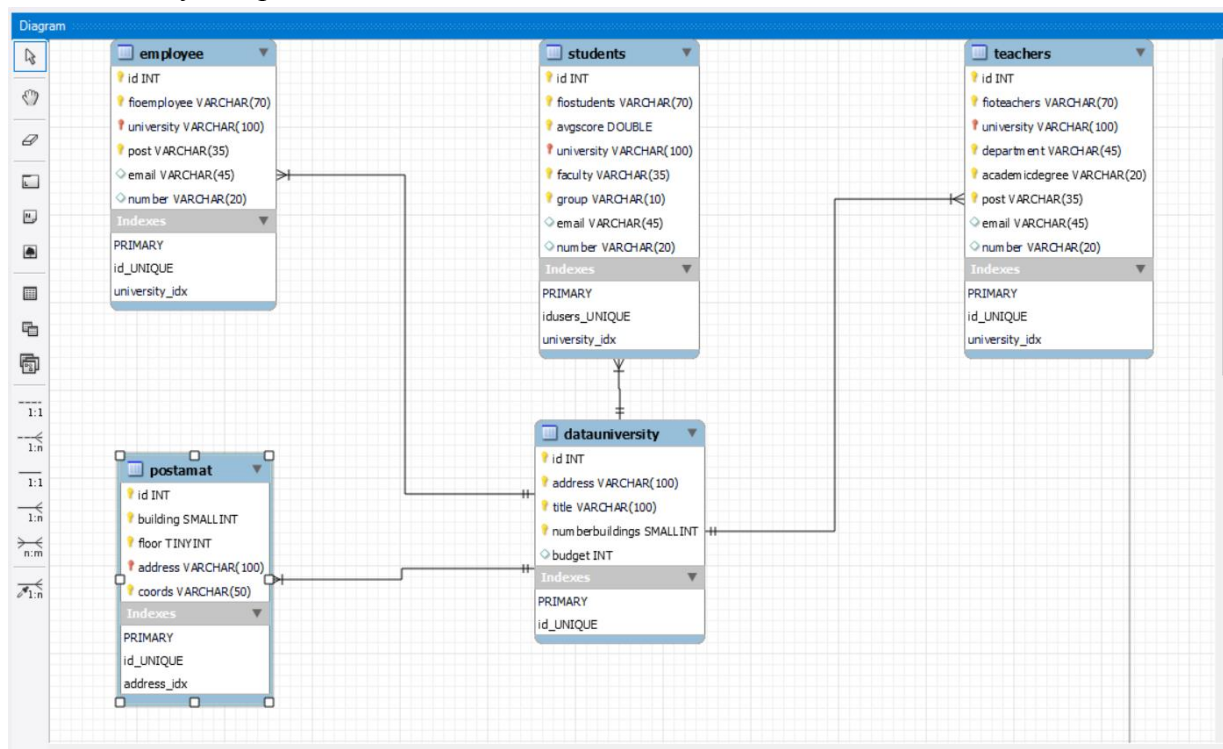
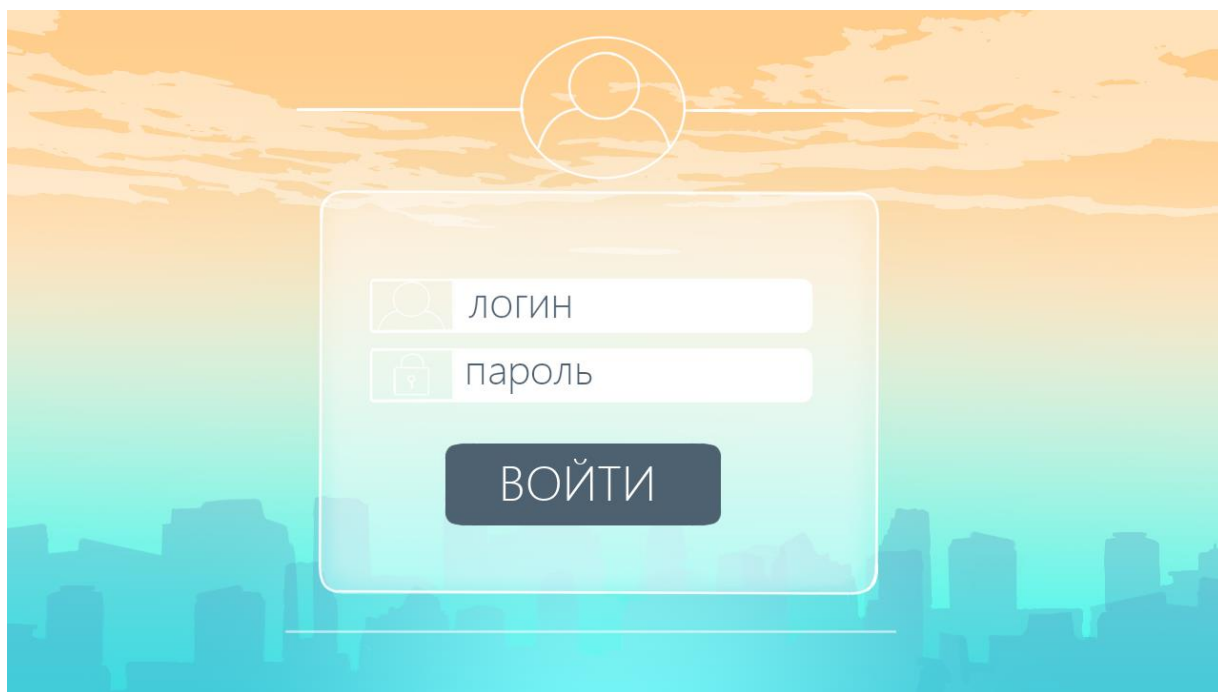


Рис. 2. Схема БД

Безопасность данных – это защитные меры, применяемые для защиты данных от несанкционированного доступа и сохранения конфиденциальности, целостности и доступности данных [3]. Для обеспечения безопасности данных в информационной системе будут приняты следующие меры: аутентификация пользователей, разделение прав доступа пользователей к информационному ресурсу, хранение данных в БД в зашифрованном виде, использование безопасного протокола передачи данных HTTPS, создание копий данных и хранение их в безопасном месте.

Обратимся к интерфейсу будущей системы. Основная цель проектирования интерфейса – создание удобного и интуитивно понятного интерфейса для пользователей, который позволит им максимально эффективно взаимодействовать с системой. Правильно разработанный интерфейс обеспечивает простоту и удобство в использовании, способствует повышению производительности, снижению ошибок и уменьшению времени на обучение пользователей.

На рис. 3 показан интерфейс авторизации: стандартные плашки «логин» и «пароль». В виду интеграции с другими сервисами университета, пользователь сам не придумывает пароль – ему его выдает ответственное лицо за выдачу СИЗ в электронном виде, так как контакты обучающегося университету известны.



**Рис. 3. Авторизация на сайт**

После авторизации система автоматически определяет, кто зашел: студент, преподаватель или сотрудник ВУЗа, ответственный за выдачу СИЗ. Разберем функционал, который будет доступен учащимся. Если авторизован студент, после нажатия на основную кнопку «Запросить выдачу СИЗ» ему будет необходимо заполнить специальную анкету, показанную на рис. 4. Обратим внимание, что поля «ВУЗ», «ФИО», «Факультет», «Группа» заполняются автоматически на основе имеющихся в БД данных. У пользователя остается возможность изменить данные в случае какой-либо ошибки, а остальные поля заполнить самостоятельно.

**Рис. 4. Интерфейс студента при заполнении заявки**

После заполнения формы ее можно отправить через специальную кнопку «Подтвердить и отправить». Предусмотрено, что пользователь может заполнить данные некорректно, в таком случае будет выведена ошибка с указанием необходимости проверить указанные данные.

В случае, когда запрашивает СИЗ преподаватель, для него будет выведена аналогичная форма, с минимальными изменениями.

После заполнения заявки студентам и преподавателям будет необходимо ожидать проверку. Во время проведения проверки специальным сотрудником ВУЗа, повторное заполнение заявки будет недоступно.

После проверки заявки и одобрении выдачи СИЗ, пользователю становится доступна информация о местоположении постамата, в котором он может их получить (в зависимости от выбора на стадии заполнения), а также кнопка с генерацией QR-кода для получения, при чем код будет действовать не более 15 минут, чтобы максимально обезопасить пользователя. Внизу экрана находится информации о дате, до которой необходимо забрать СИЗ. Все ранее перечисленное показано на рис. 5.

**Рис. 5. Интерфейс после одобрения заявки**

Система, позволяющая передавать средства индивидуальной защиты без прямого взаимодействия людей, однозначно повышает безопасность здоровья, минимизирует риски передачи инфекции и способствует более эффективному контролю за распространением инфекций.

### **Библиографический список**

1. О нас // ПНППК [Электронный ресурс]. – URL: <https://pnppk.ru/about-us> (Дата обращения: 27.02.2023).
2. Методические рекомендации № 02/6338-2020-15 «О рекомендациях по профилактике коронавирусной инфекции (COVID-19) среди работников». М.: Федеральный центр гигиены и эпидемиологии Роспотребнадзора, 2020, 5 с.
3. Что такое безопасность данных? // Oracle [Электронный ресурс]. – URL: <https://www.oracle.com/cis/security/database-security/what-is-data-security/> (Дата обращения: 29.05.2023)

### **References**

1. O nas // PNPPK [Electronic resource]. – URL: <https://pnppk.ru/about-us> (Date of application: 27.02.2023)
2. Methodological Recommendations № 02/6338-2020-15 «O rekomendaciyah po profilaktike koronavirusnoj infekcii (COVID-19) sredi rabotnikov». M.: Federal Center of Hygiene and Epidemiology of Rospotrebnadzor, 2020, 5 p.
3. Chto takoe bezopasnost' dannyh? // Oracle [Electronic resource]. – URL: <https://www.oracle.com/cis/security/database-security/what-is-data-security/> (Date of application: 29.05.2023)

## **DESIGNING AND DOCUMENTING THE INFORMATION SYSTEM FOR ISSUING PPE TO STUDENTS AND TEACHERS DURING THE EPIDEMIC THREAT**

*Demidov Matvey A., Vasiluk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, demidovmatvey@rambler.ru

The article discusses the design and documentation of an information system for issuing PPE (personal protective equipment) to students and teachers during the epidemic threat. The author presents the results of a study in which the requirements for the system, its functionality, characteristics, architecture and operating principle were determined, and the process of developing a graphical interface is also described. The developed system ensures safety for the health of users, minimizing contacts and reducing the risk of virus infection, automates the process of issuing PPE and creates convenience for students and teachers. It can be used in schools, universities and other educational institutions in an epidemic situation.

**Keywords:** ensuring health safety in educational institutions, epidemic threat, automation of the issuance of PPE.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭЛЕКТРОННЫЙ ПРОЕЗДНОЙ»

*Демин Эдуард Фаридович, Василюк Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, edik.demin@bk.ru

В работе представлены основные средства для проектирования и документирования информационной системы для безналичной оплаты проезда в общественном транспорте. Определены объект, предмет и цель исследования, представлены методы исследования и задачи для достижения цели исследования. Проанализированы схожие информационные системы в соответствии с предметной областью. Определены функциональные требования информационной системы. В результате анализа инструментального программного обеспечения было выбрано следующее: CASE-средство Software Ideals Modeler [1]; для хранения данных СУБД MySQL [2]; Java Script и React-Native для клиентской части; Node.js для реализации серверной части информационной системы; PaintTool SAI [3] для создания пользовательского интерфейса приложения. Спроектированные диаграммы демонстрируют варианты взаимодействия пользователей с системой, структуру хранилища данных, а также способы установки системы на устройствах. В соответствии с диаграммами был нарисован прототип пользовательского интерфейса.

Ключевые слова: проектирование, документирование, информационная система, электронный проездной, проездной.

Согласно исследованиям, проведенным организацией «ВЦИОМ» [4], каждый день общественным транспортом пользуются двадцать процентов россиян, чаще всего это жители мегаполисов. Организации, занимающиеся общественными перевозками, создают информационные системы, дабы облегчить себе работу, увеличить прибыль и сделать оплату проезда проще и удобнее.

Актуальность работы обусловлена ростом доли людей, совершающих поездки на общественном транспорте, а также их низким интересом к информационным системам электронных проездных.

Целью настоящего исследования было спроектировать и задокументировать информационную систему «Электронный проездной». Объектом исследования являлась безналичная оплата проезда в общественном транспорте. Предметом исследования – автоматизация безналичной оплаты проезда в общественном транспорте.

Опишем действия, которые пользователь может выполнять в нашей информационной системе. При использовании приложения пользователем, сначала ему необходимо будет пройти регистрацию, указать фамилию, имя отчество, пароль и контактный телефон, который в дальнейшем будет использоваться как логин, после чего он сможет авторизоваться. Попадая на главную страницу приложения, пользователю будут доступны такие опции как: просмотр привязанных карт, чат поддержки, история поездок, карта города, меню.



Для демонстрации вариантов взаимодействия пользователя с системой построена логика работы приложения с помощью диаграммы прецедентов представленной ниже (см. рис. 1.).

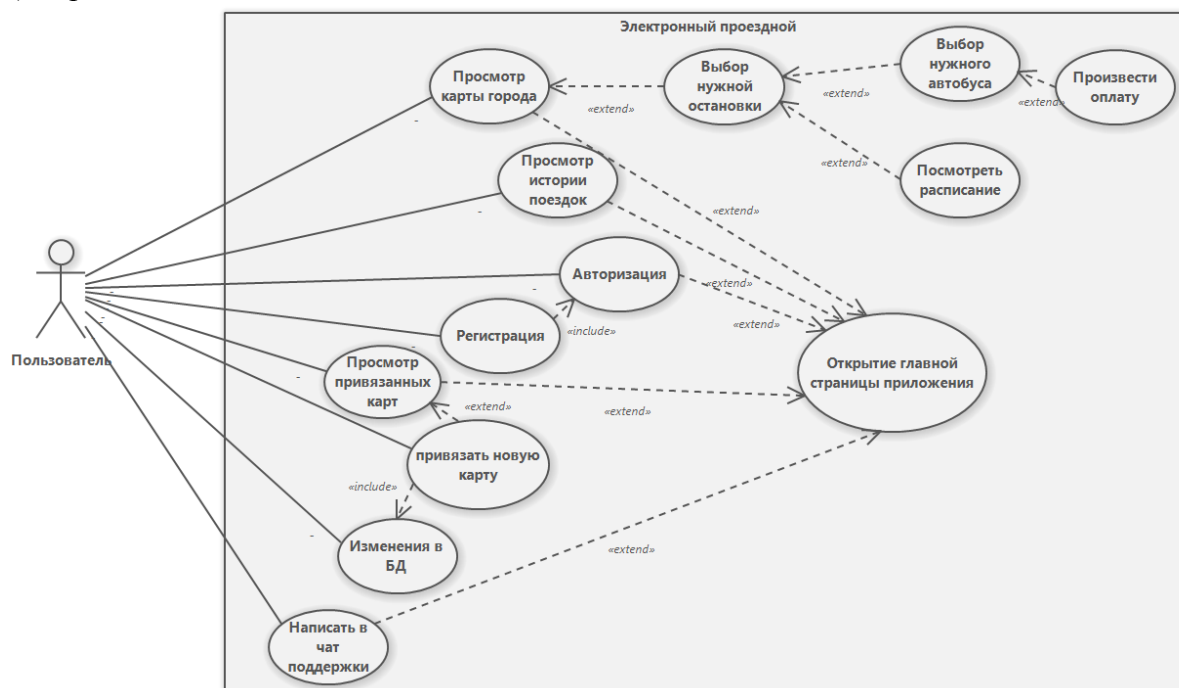


Рис. 1. Диаграмма прецедентов пользователя

ER-диаграмма представленная ниже (см. рис. 2.) демонстрирует структуру базы данных со всеми необходимыми и достаточными сущностями.

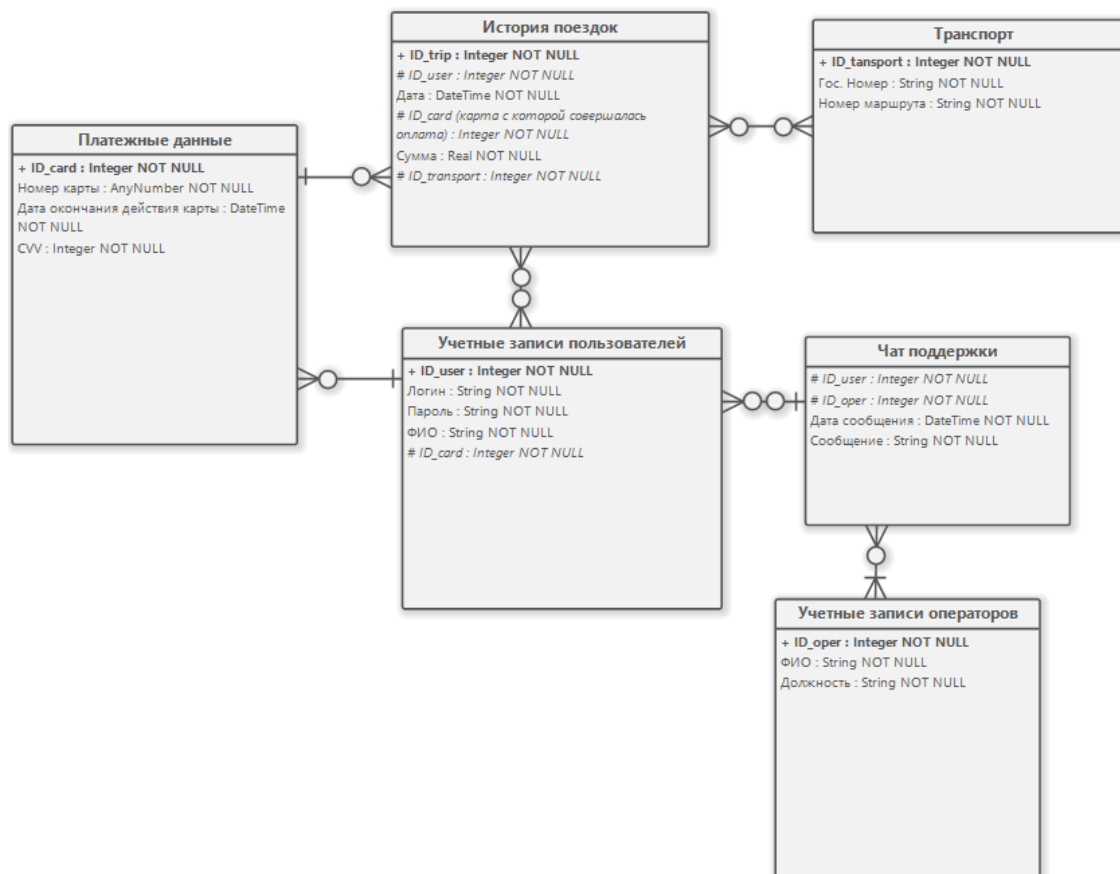


Рис. 2. ER-диаграмма структуры базы данных

Далее представлен прототип пользовательского интерфейса и его описание.

При запуске клиентского приложения открывается экран авторизации с полями для ввода данных и тремя кнопками «Войти», «Зарегистрироваться» и «Восстановить пароль». Пользователь может создать новый аккаунт или войти в уже имеющийся, а если забыл пароль, то есть возможность его восстановить. Для этого необходимо будет ввести код из смс.

Для входа в свой аккаунт пользователю необходимо ввести в поля данные, которые он указывал при регистрации. Проверка данных осуществляется через базу данных. Если пользователь введет неверные данные, на экране появится сообщение об ошибке входа и пользователю будет предложено ввести данные снова.

После входа в учетную запись открывается страница «Меню», продемонстрированная ниже (см. рис. 3.). Страница содержит информацию о профиле, кнопку «Редактировать профиль», а также пять дополнительных кнопок: «Тарифное меню», «Чат службы поддержки», «Обновления», «О приложении», «Выйти».

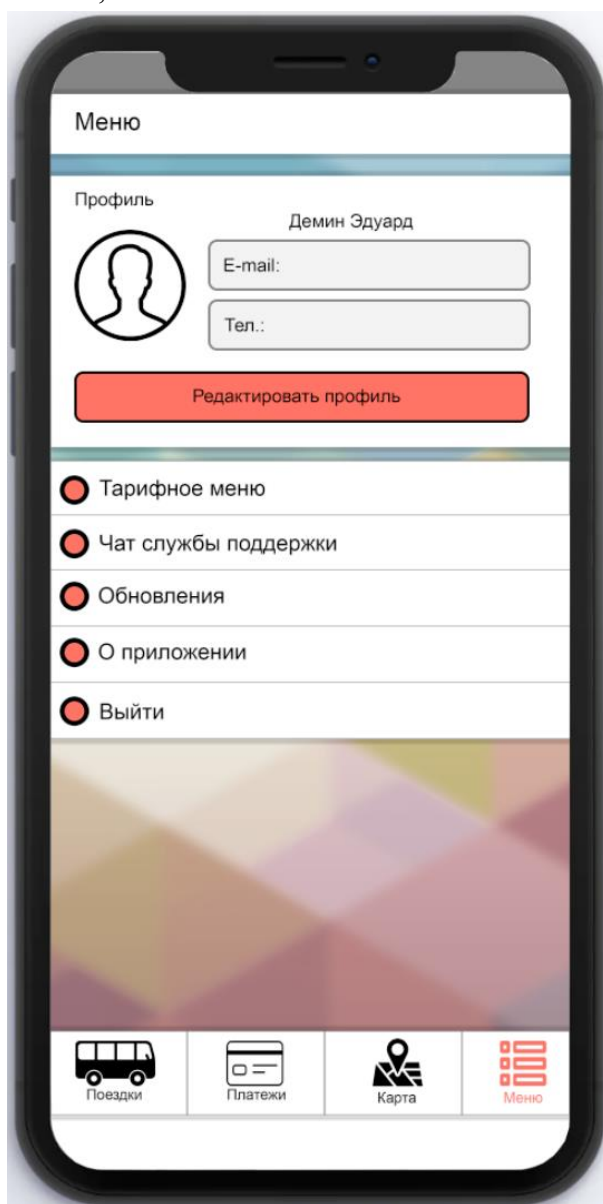


Рис. 3. Страница «Меню»

Снизу для перехода по главным страницам имеются четыре кнопки с одноименными названиями страниц.



Страница «Поездки» (см. рис. 4.) содержит список платежных карт, по каждой из которых отдельно показывается история поездок. В истории пишется дата поездки, время, маршрут, цена поездки и тип транспорта в виде иконки самого транспорта, каждый тип транспорта обозначен своим цветом и иконкой.

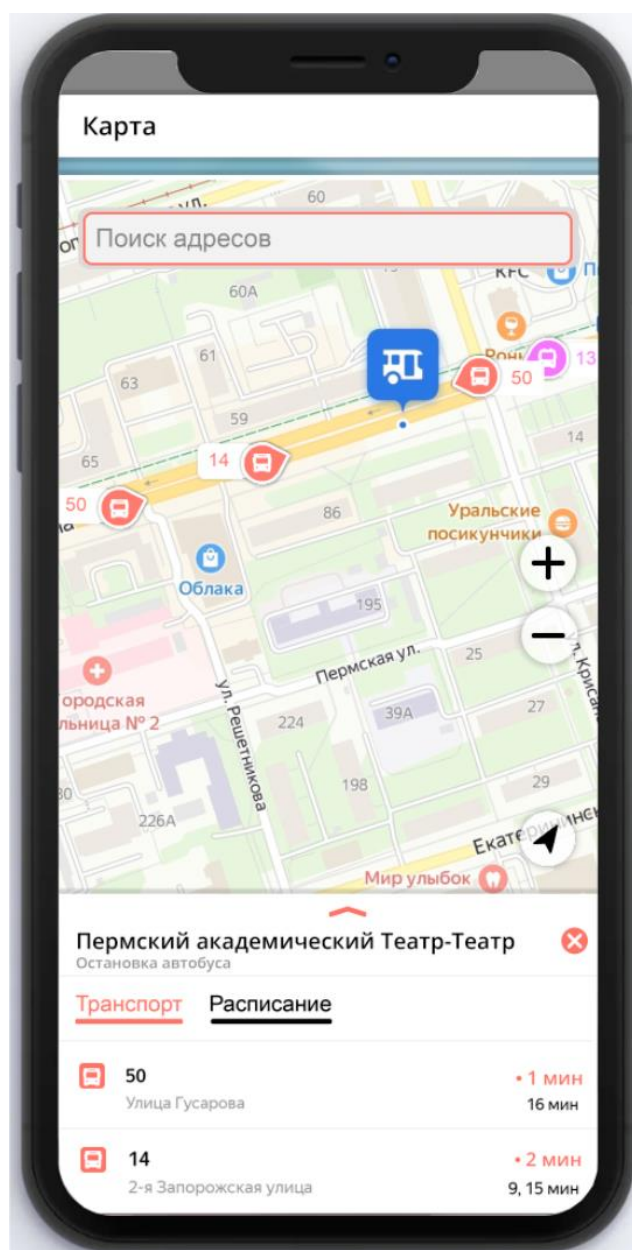
Также список карт можно дополнить при помощи специальной кнопки, при нажатии на которую пользователя перенесет на страницу «Платежи» где можно добавить или удалить карту, выбрать основную.



Рис. 4. Страница «Поездки»

Рассмотрим самую интересную страницу «Карта» (см. рис. 5.). На ней имеется карта города, на карте GPS-метка пользователя, его текущее местоположение, имеются также полезные кнопки для увеличения или уменьшения масштаба карты, центрирования. Отображается местоположение остановок общественного транспорта. При нажатии на остановку появляются GPS-метки общественного транспорта, также появляется всплывающее окно с информацией об остановке, двумя кнопками «Транспорт» и «Расписание».

При нажатии на транспорт также открывается всплывающее окно, в котором можно оплатить проезд.



**Рис. 5. Страница «Карта»**

В заключение отметим, что разрабатываемая ИС представляет научный и практический интерес для дальнейших исследований, имеет потенциал для внедрения в транспортную систему общественного транспорта.

### **Библиографический список**

1. CASE tool for diagrams, software design & analysis // Software ideas modeler [Электронный ресурс]. – URL: <https://www.softwareideas.net/en/about> (дата обращения 16.04.2023)
2. MySQL Documentation // MySQL [Электронный ресурс]. – URL: <https://dev.mysql.com/doc/> (дата обращения: 16.04.2023)
3. PaintTool SAI // PaintTool SAI: high quality and lightweight painting software. [Электронный ресурс]. – URL: <http://www.systemax.jp/en/sai/> (дата обращения: 20.04.2023)

### **References**

1. CASE tool for diagrams, software design & analysis // Software ideas modeler [Electronic resource]. – URL: <https://www.softwareideas.net/en/about> (date of application: 16.04.2023)

2. MySQL Documentation // MySQL [Electronic resource]. – URL: <https://dev.mysql.com/doc/> (date of application: 16.04.2023)

3. PaintTool SAI // PaintTool SAI: high quality and lightweight painting software [Electronic resource]. – URL: <http://www.systemax.jp/en/sai/> (date of application: 20.04.2023)

## **DESIGN AND DOCUMENTATION OF THE INFORMATION SYSTEM "ELECTRONIC TRAVEL TICKET"**

*Demin Eduard F., Vasiliuk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [edik.demin@bk.ru](mailto:edik.demin@bk.ru)

The paper presents the main tools for designing and documenting an information system for non-cash payment of public transport. The object, subject and purpose of the study are defined, research methods and tasks for achieving the research goal are presented. Similar information systems are analyzed in accordance with the subject area. The functional requirements of the information system are defined. As a result of the analysis of the instrumental software, the following was selected: CASE-tool Software Ideals Modeler [1]; for storing MySQL database data [2]; Java Script and React-Native for the client side; Node.js for the implementation of the server part of the information system; PaintTool SAI [3] for creating the user interface of the application. The designed diagrams demonstrate options for user interaction with the system, the structure of the data warehouse, as well as ways to install the system on devices. In accordance with the diagrams, a prototype of the user interface was drawn.

**Keywords:** design, documentation, information system, electronic travel card, travel card.

## ERP-СИСТЕМЫ РОССИЙСКОГО ПРОИЗВОДСТВА

*Дружинина Елена Владимировна, Раевский Виктор Николаевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, telena2007@ya.ru

В данной статье рассматриваются ERP-системы российского производства. Актуальность данного материала обусловлена ситуацией, сложившейся в данный момент на российском рынке систем управления производством. Проводится сравнительный анализ наиболее известных российских ERP-систем. Определяется набор критериев и механизм оценки в рамках выбранных критериев. На основе приведенного анализа выбрана лучшая система по техническим и экономическим показателям. Материал данной статьи может представлять интерес для специалистов предприятий, перед которыми встаёт вопрос выбора системы с целью повышения эффективности организации производственных процессов на предприятии, а также для студентов, проходящих обучение в рамках обобщённого направления «Информационные технологии».

Ключевые слова: ERP системы, российский рынок, российские ERP-системы, критерии.

Современные реалии таковы, что предприятия вне зависимости от вида деятельности все чаще сталкиваются с необходимостью автоматизации. Главная задача любого производства – повышение производительности и рост качества продукции без увеличения затрат. Ручное управление производственными процессами снижает качество и требует больше времени, поэтому процесс управления необходимо максимально автоматизировать.

Одним из инструментов автоматизации бизнес процессов являются системы ERP.

ERP-система – это корпоративная информационная система, предназначенная для автоматизации основных бизнес-процессов компании, учета и управления (планирование, контроль и анализ) ресурсами, управление персоналом, производством, закупками, услугами, поставками и т.д. Сам термин ERP – это аббревиатура, образованная из английских слов Enterprise Resource Planning, дословно «планирование ресурсов предприятия». Такие системы были наследием систем MRP и MRP II класса (англ. Material Requirements Planning – планирование потребности в материалах). Но за счет включения в себя множества других модулей или подсистем аббревиатура ERP стала собирательным наименованием систем планирования, учета, управления и принятия решений [1].

Основными преимуществами внедрения ERP-систем являются:

- прозрачные сквозные бизнес-процессы;
- повышение уровня информационной безопасности;
- снижение объемов материальных запасов, оборачиваемость складских запасов;
- снижение производственных издержек;
- снижение себестоимости выпускаемой продукции;
- сокращение длительности простоев оборудования;
- увеличение объемов выпускаемой продукции;
- сокращение сроков исполнения заказов;
- повышение качества принимаемых управленческих решений.

Основные модули (системы) которые могут входить в состав комплексной ERP-системы как платформы или быть отдельными системами показаны в табл. 1.

Таблица 1. Модули ERP

MES (Manufacturing Execution System) – производственные управляющие системы	CRM (Customer Relationship Management) – системы управления взаимоотношениями с клиентами	TMS (Transportation Management System) – системы управления транспортировками	KPI (Key Performance Indicators) – ключевые показатели эффективности
WMS (Warehouse Management System) – системы управления складами	SCM (Supply Chain Management) – системы управления цепочками поставок	PDM (Product Data Management) – системы управления данными о продуктах	EAM (Enterprise Asset Management) – управление основными фондами предприятия
MDM (Master Data Management) – системы управления мастер-данными	BPMS (Business Process Management System) – системы управления бизнес-процессами	HRM (Human Resources Management) – управление персоналом	CPM (Corporate Performance Management) – система управления эффективностью предприятия

Санкции оказывают очень сильное влияние на этот сегмент информационных технологий (ИТ). Российские ERP-системы продолжают развивать функциональность в направлении облачных решений, но основной тенденцией по-прежнему является развитие функциональности систем, необходимых для нормальной работы крупных компаний.

В 2022 году мощнейшее влияние на развитие ERP-рынка оказала специальная военная операция (СВО), начавшаяся 24 февраля, а также последующие за ней череда антироссийских санкций и практически синхронный уход из России зарубежных вендоров. В результате остановилось большинство проектов на SAP и других импортных платформах, а те внедрения, которые только планировались, компаниям пришлось реконфигурировать. Клиенты, использующие западные решения, оказались в сложной ситуации – возросли риски утратить стабильность ИТ-ландшафта.

Уход крупных вендоров с рынка и прекращение сервисной поддержки стали драйверами для запуска проектов перехода на отечественные ERP-системы. Теперь заказчикам надо выбирать из таких систем, как 1С (1С:ERP), «Парус», «Галактика», «Турбо ERP», «Компас ERP» и некоторых других [2].

Рассмотрим основные ERP системы российского рынка:

1. 1С:ERP является наиболее массово применяемой в настоящее время в России и странах СНГ системы ERP-класса «1С:Управление производственным предприятием» редакции 1.3 (1С:УПП). В новом программном продукте учтен опыт, накопленный за 13 лет использования этой системы в масштабных проектах, насчитывающих сотни и тысячи рабочих мест [3].

2. ПАРУС использует только импортонезависимые компоненты, включая СУБД PostgreSQL и отечественные операционные системы семейства Linux. Серверная часть открыта в части алгоритмов исполнения бизнес-процессов и структур хранения данных [4].

3. Галактика ERP. Модульная архитектура системы позволяет приобретать только нужную конфигурацию, необходимую для решения конкретных задач. Высокая производительность платформы проверена в действующих проектах на крупных предприятиях. Интеграция с решениями класса САПР, CAD / CAM / CAE и другим специализированным программным обеспечением [5].

4. ТУРБО ERP. Система, которая сохраняет продуктивность при использовании 100 и более сквозных аналитик финансового учёта, готова поддерживать одновременную работу нескольких тысяч пользователей, показывает отличные результаты на тестах производительности и входит в реестр российских программ вместе с платформой разработки [6].

5. Компас ERP. Гибкий инструментарий Мастера Компаса позволяет: самостоятельно вносить изменения в структуру и алгоритмы ПО; настраивать программу без знаний программирования, формировать отчёты любых форматов; адаптировать процессы под новые задачи; создавать собственные подсистемы на мастерах Компас с программированием и без [7].

Для сравнения российских ERP систем выделено несколько наиболее важных критериев, вес критериев расположен в диапазоне от 0 до 1.

При сборе информации были использованы открытые источники. Результаты сравнения приведены в табл. 2 на основании данных, полученных по формуле:  $I = \sum_1^i (P_i * K_i)$ ,

где I – индекс системы (чем выше индекс, тем система лучше), K – критерий системы, P – вес критерия.

Таблица 2. Сравнение ERP-систем по критериям

Критерии	Вес	1С:ERP	ПАРУС	Галактика ERP	ТУРБО ERP	Компас ERP
ведение бухгалтерского учета	0,7	9	9	9	8	8
подсистема планирования ресурсов предприятия	1,0	8	8	8	6	8
подсистема управления производством	1,0	6	7	7	0	7
подсистема управления закупками	1,0	8	7	7	6	6
подсистема управления продажами	1,0	8	7	8	7	7
подсистема управления отгрузкой	1,0	8	6	6	6	7
подсистема управления персоналом	0,8	8	6	8	5	8
подсистема управления техническим обслуживанием и ремонтом	0,9	8	6	5	8	8
подсистема управления качеством	0,9	6	0	5	0	0
электронный документооборот	0,5	6	6	9	6	8
экспорт отчетов в форматы Excel, Word, PDF, Open Office	0,5	7	7	7	7	7
включение различных функциональных частей прикладного решения без программирования	0,9	8	8	8	8	8
расположение базы на web-сервере	0,7	8	6	8	6	7
затраты на обучение	0,5	6	6	6	7	7
стоимость лицензии на использование программного продукта	1,0	6	7	8	9	9

Результаты распределились следующим образом: 1С = 77; ПАРУС=67; Галактика=76; ТУРБО=62; Компас=73.

Анализ полученных результатов привёл к выводу, что по техническим и экономическим критериям лучшей российской системой ERP можно считать 1С:ERP.

Следует отметить, что при сборе информации пришлось столкнуться с недостатком открытых данных для проведения более детального анализа. При выборе системы необходимо на первоначальном этапе сделать акцент на критериях, подходящих именно вашему предприятию, что в дальнейшем даст желаемый результат с меньшими затратами. Так же стоит отметить что, на российском рынке за последние два года количество разработчиков ERP систем значительно увеличилось, часть разработчиков делает упор на определенные отрасли промышленности.

### Библиографический список

1. Бобровников А.Э. Введение в управление проектами внедрения ERP-систем Электронная книга в формате ePub; ISBN 978-5-9677-3019-1. URL: <https://iknigi.net/avtor-a-bobrovnikov/204634-vvedenie-v-upravlenie-proektami-vnedreniya-erp-sistem-a-bobrovnikov/read/page-1.html>
2. Системы управления предприятием (ERP) рынок России. [Электронный ресурс]. – URL: [https://www.tadviser.ru/index.php/Статья:Системы\\_управления\\_предприятием\\_\(ERP-рынок\\_России\)](https://www.tadviser.ru/index.php/Статья:Системы_управления_предприятием_(ERP-рынок_России))
3. Конкурентные преимущества «1С: Управление предприятием» [Электронный ресурс]. – URL: <https://v8.1c.ru/erp/preimushchestva/>
4. ERP решения ПАРУС для крупного бизнеса [Электронный ресурс]. – URL: <https://erp.parus.com/>
5. TAdviser: На что способна отечественная ТУРБО ERP. Обзор продукта [Электронный ресурс]. – URL: <https://turbosolution.ru/news/tadviser-turbo-erp-product-review>
6. ТУРБО ERP. [Электронный ресурс]. – URL: <https://turbosolution.ru/products/erp>
7. Компас. ERP-системы для вашего бизнеса [Электронный ресурс]. – URL: <https://www.compas.ru/erp#about>

### References

1. Bobrovnikov A.E. Introduction to ERP system implementation project management ebook in ePub format; ISBN 978-5-9677-3019-1. URL: <https://iknigi.net/avtor-a-bobrovnikov/204634-vvedenie-v-upravlenie-proektami-vnedreniya-erp-sistem-a-bobrovnikov/read/page-1.html>
2. Enterprise management systems (ERP) market in Russia. [Electronic resource]. – URL: [https://www.tadviser.ru/index.php/Статья:Системы\\_управления\\_предприятием\\_\(ERP-market\\_Russia\)](https://www.tadviser.ru/index.php/Статья:Системы_управления_предприятием_(ERP-market_Russia))
3. Competitive advantages of "1C: Enterprise Management" [Electronic resource]. – URL: <https://v8.1c.ru/erp/preimushchestva/>
4. PARUS ERP solutions for large business [Electronic resource]. – URL: <https://erp.parus.com/>
5. TAdviser: What domestic TURBO ERP can do. Product review [Electronic resource]. – URL: <https://turbosolution.ru/news/tadviser-turbo-erp-product-review>
6. TURBO ERP. [Electronic resource]. – URL: <https://turbosolution.ru/products/erp>
7. Compass. ERP systems for your business [Electronic resource]. – URL: <https://www.compas.ru/erp#about>

## **RUSSIAN-MADE ERP SYSTEMS**

*Druzhinina Elena V., Rayevskiy Viktor N.*

Perm State National Research University, 15, Bukireva st., Perm, 614990, Russia,  
telena2007@ya.ru

This article discusses Russian-made ERP systems. The relevance of this material is due to the current situation in the Russian market of production management systems. A comparative analysis of the most famous Russian ERP systems is being carried out. A set of criteria and an evaluation mechanism within the selected criteria are defined. Based on the above analysis, the best system for technical and economic indicators was selected. The material of this article may be of interest to specialists of enterprises, who are faced with the question of choosing a system in order to increase the efficiency of organizing production processes at the enterprise, as well as for students studying in the framework of the generalized direction "Information Technologies".

**Keywords:** ERP systems, Russian market, Russian ERP systems, criteria.



## РАЗРАБОТКА БИБЛИОТЕКИ ДЛЯ РЕАЛИЗАЦИИ МАТРИЧНОГО НЕЧЁТКОГО ЛОГИЧЕСКОГО ВЫВОДА НА ЯЗЫКЕ PYTHON

*Епишина Наталья Валерьевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, epishina.nata@bk.ru

*Селетков Илья Павлович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, iseletkov@gmail.com.

В работе кратко описаны принципы нечёткой логики, особенности матричного представления операций над нечёткими переменными. Для упрощения построения систем поддержки принятия решений с использованием матричного представления принято решение разработки библиотеки python. Для этого проанализированы существующие библиотеки, выявлены преимущества, реализованные популярные нечёткие алгоритмы. Далее описывается процесс разработки и публикации в официальной репозитории новой библиотеки, позволяющей производить расчёты с помощью матричного логического вывода. Корректность работы библиотеки доказана путём решения демонстрационной задачи нечёткого логического вывода для управления работой химического реактора.

Ключевые слова: нечёткая логика, матричный нечёткий логический вывод, библиотека для python, системы поддержки принятия решений.

### Матричное представление нечёткой логики

«Классическая» математическая логика использует только строго формализованные данные, истинность высказываний является дискретной величиной, способной принимать два значения [1]:

"1" – если выражение истинно;

"0" – если ложно.

В своей теории нечетких множеств Лотфи Заде ввёл понятие "нечеткого множества" [2], "степень принадлежности" к которому может принимать значения не только из набора  $\{0,1\}$ , но и на целом отрезке  $[0, 1]$ .

В матричном представлении [3] в качестве нечётких переменных -значений функций принадлежности предложено использовать вектора вида

$$X = x_0 e^{(0)} + x_1 e^{(1)}, \quad (1)$$

где базисные вектора  $e^{(0)}$ ,  $e^{(1)}$  соответствуют полной ложности или истинности высказывания, а координаты  $x_0$  и  $x_1$  вектора обозначают степень ложности и истинности высказывания, описываемого вектором, и также принимает значения на отрезке  $[0,1]$ . При этом обязательно соблюдение условия полной истинности.

$$x_0 + x_1 = 1. \quad (2)$$

Логические операции конъюнкции, дизъюнкции и импликации над нечеткими векторами можно сформулировать с помощью матриц (3), детальное обоснование выбора которых представлено в работе [3].

$$C(x) = \begin{pmatrix} 1 & x_0 \\ 0 & x_1 \end{pmatrix}; D(x) = \begin{pmatrix} x_0 & 0 \\ x_1 & 1 \end{pmatrix}; I(x) = \begin{pmatrix} x_1 & 0 \\ x_0 & 1 \end{pmatrix}. \quad (3)$$

Само вычисление результата операции осуществляется умножением соответствующей матрицы на второй вектор-операнд, результатом тоже является вектор.

$$x \wedge y = C(x)y; x \vee y = D(x)y; x \rightarrow y = I(x)y. \quad (4)$$

Одним из основных методов представления знаний в экспертных системах являются продукционные правила, позволяющие приблизиться к стилю мышления человека. Любое правило продукций состоит из посылок и заключения. Возможно наличие нескольких посылок в правиле, в этом случае они объединяются посредством логических связок И, ИЛИ [1]. Обычно продукционное правило записывается в виде:

«ЕСЛИ (посылка\_1) (связка) (посылка\_2) (св) ... (св) (посылка\_n), ТО (заключение)».

В матричном представлении логический вывод по таким правилам сводится к решению системы линейных алгебраических уравнений (5) относительно неизвестного вектора  $y$ :

$$x \rightarrow y = z; I(x) * y = z, \quad (5)$$

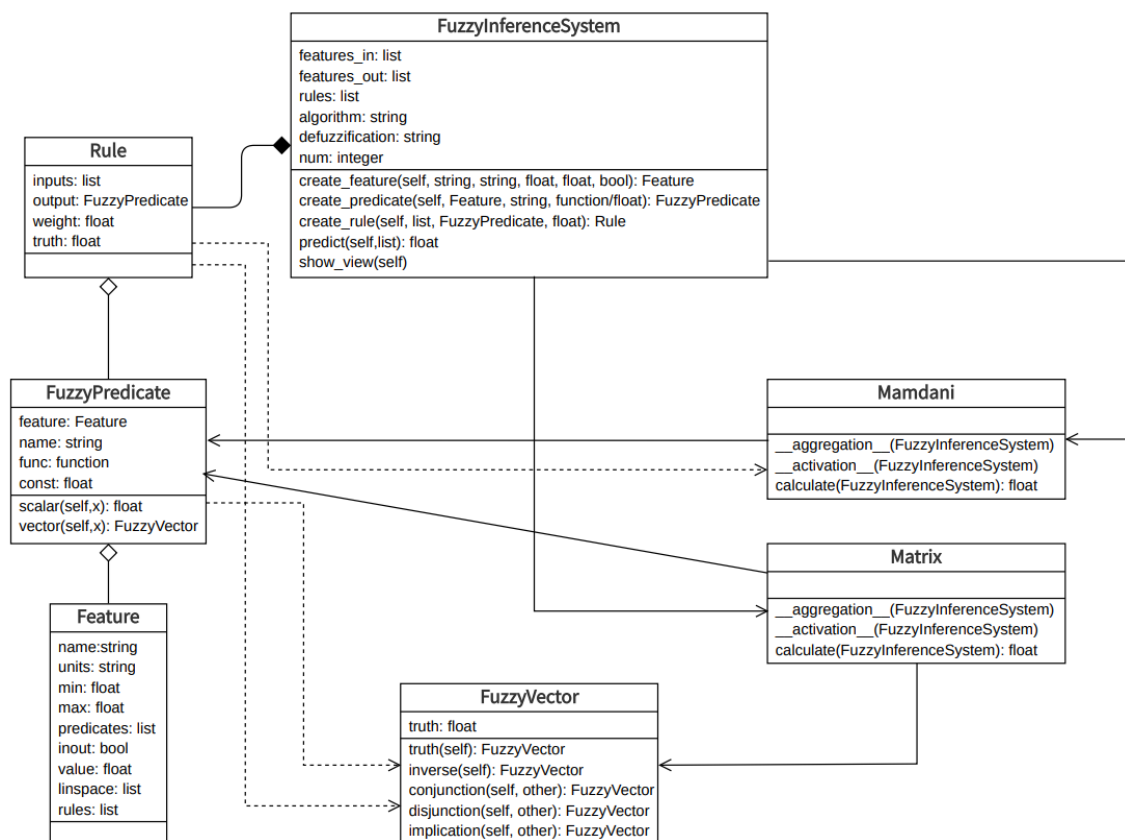
где вектор  $z$  описывает степень истинности самого правила, а вектор  $x$  – степень истинности всей предпосылки правила. Условия существования решения системы описаны в работе [3] и могут быть использованы для проверки полноты базы знаний в разрабатываемой библиотеке.

### **Библиотека для реализации матричного представления нечёткой логики на языке python**

Для написания библиотеки использован объектно-ориентированный подход к программированию. Диаграмма классов UML приведена на рис. 1.

Для функций принадлежности в библиотеке имеются 3 реализации: треугольная, трапециевидная и гауссовская, также возможно задание произвольной кусочно-ломанной функции.

Основной класс, отвечающий за взаимодействие внешней программы с библиотекой, является класс FuzzyInfernceSystem. Он содержит поля features\_in – список лингвистических переменных, которые описывают входные параметры, features\_out – список лингвистических переменных, которые описывают выходные параметры, rules – список правил, algorithm – выбор алгоритма (Mamdani или Matrix), defuzzification – выбор метода получения чёткого результата (Centroid или Simple, центр масс или упрощённый).



**Рис. 1. Диаграмма классов библиотеки для реализации матричного алгоритма нечёткого логического вывода**

Методы класса:

`create_feature (self, name, units, min, max, inout)` – создания признака или лингвистической переменной, возвращает объект класса `Feature`;

`create_predicate (self, feature: Feature, name, func=None, const=None)` – создание предиката или термина лингвистической переменной, возвращает объект класса `FuzzyPredicate`;

`create_rule (self, input_predicates, output_predicate, weight)` – создание правил, возвращает объект класса `Rule`;

`predict (self,*x)` – вычисление значений выходных параметров по заданным входным данным, возвращает `float`;

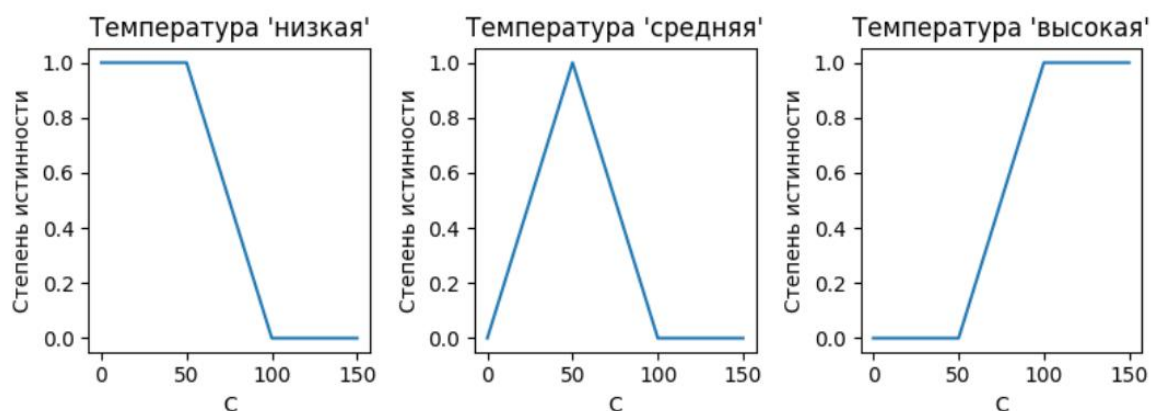
`show_view (self)` – вывод всех графиков принадлежности лингвистических переменных и соответствующих им термов.

### Цифровой двойник химического реактора

Для демонстрации работы библиотеки опишем реализацию задачи поддержки принятия решений.

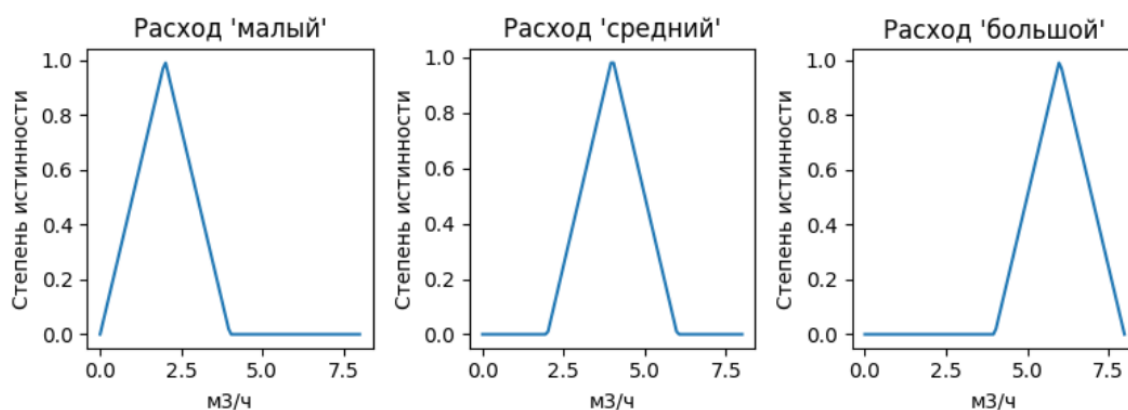
Пусть есть некоторая система, например реактор, описываемая тремя параметрами: температура, давление и расход рабочего вещества. Требуется производить расчёт значения давления, по заданным значениям температуры и расхода вещества.

Температура, принимает значения из интервала  $[0,150]$  °C, и описывается терминами { Низкая, Средняя, Высокая }. Графики функций принадлежности нечётким множествам приведены на рис. 2.



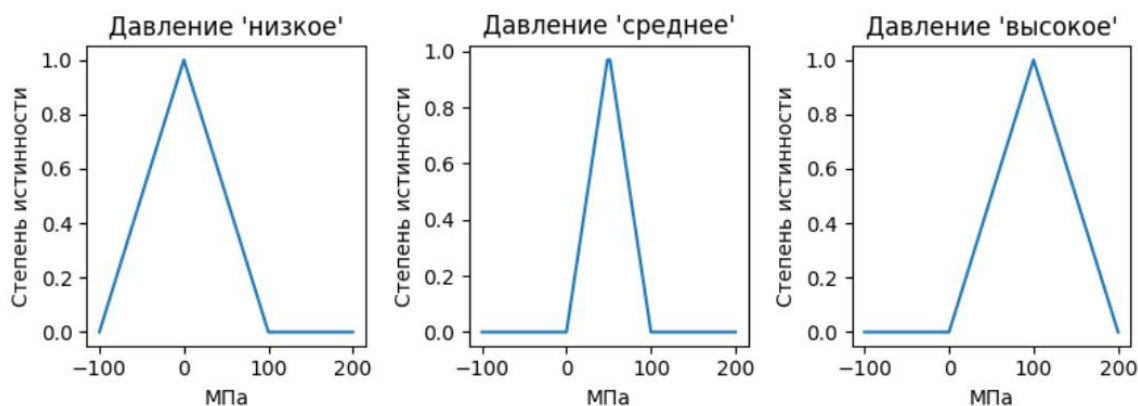
**Рис. 2. Графики функций принадлежности нечётким множествам, описывающих значение температуры**

Расход, принимает значения из интервала  $[0,8]$  м<sup>3</sup>/ч, описывается термами {Большой, Средний, Малый}. Графики функций принадлежности нечётким множествам приведены на рис. 3.



**Рис. 3. Графики функций принадлежности нечётким множествам, описывающих значение расхода рабочего вещества**

Давление, принимает значения из интервала  $[0, 100]$  МПа, но, в соответствии с особенностями алгоритма получения чёткого результата с помощью расчёта момента по оси абсцисс, располагаем крайние графики так, чтобы их моменты совпадали с границами области допустимых значений давления. Давление описывается терминами {Высокое, Среднее, Низкое}. Графики функций принадлежности нечётким множествам имеют следующий вид (рис. 4):



**Рис. 4. Графики функций принадлежности нечётким множествам, описывающих значение давления**

База знаний состоит из набора правил, приведённого в табл. 1

Таблица 1. База знаний

Температура	Расход	Давление	Степень истинности правила
«низкая»	«малый»	«низкое»	1
«средняя»		«среднее»	1
«высокая»		«высокое»	1
	«большой»	«высокое»	1

Пример исходного кода внешней программы, использующий разработанную библиотеку, реализующий решение данной задачи приведён на лист.1.

Листинг 1 – пример кода
<pre> from fuzzy_matrix_mamdani import FuzzyInferenceSystem, FuzzyVector, Feature, FuzzyPredicate, Rule, Mamdani, Matrix, Trapeze, Triangle, Gauss fis = FuzzyInferenceSystem() fis.algorithm = 'Matrix' fis.defuzzification = "Simple" f_temp = fis.create_feature("Температура", "C", 0, 150, True) f_flow = fis.create_feature("Расход", "м3/ч", 0, 8, True) f_pressure = fis.create_feature("Давление", "МПа", -100, 200, False) p_temp_low = fis.create_predicate(f_temp, 'низкая', Trapeze(0, 0, 50, 100)) p_temp_normal = fis.create_predicate(f_temp, 'средняя', Trapeze(0, 50, 100, 150)) p_temp_high = fis.create_predicate(f_temp, 'высокая', Trapeze(50, 100, 150, 150)) p_flow_low = fis.create_predicate(f_flow, 'малый', Triangle(0, 2, 4)) p_flow_normal = fis.create_predicate(f_flow, 'средний', Triangle(2, 4, 6)) p_flow_high = fis.create_predicate(f_flow, 'большой', Triangle(4, 6, 8)) p_pressure_low = fis.create_predicate(f_pressure, 'низкое', const=0) p_pressure_normal = fis.create_predicate(f_pressure, 'среднее', const=50) p_pressure_high = fis.create_predicate(f_pressure, 'высокое', const=100) r_1 = fis.create_rule([p_temp_low, p_flow_low], p_pressure_low, 1) r_2 = fis.create_rule([p_temp_normal], p_pressure_normal, 1) r_3 = fis.create_rule([p_temp_high], p_pressure_high, 1) r_4 = fis.create_rule([p_flow_high], p_pressure_high, 1) [presure] = fis.predict(85, 3.5) print(f"Значение давления: {presure}") </pre>

Результаты работы программы с использованием матричного алгоритма и алгоритма Мамдани, а также существующей популярной библиотеки skfuzzy приведены в табл. 2.

Таблица 2. Результаты расчётов давления несколькими способами

Температура, °C	Расход, м3/ч	Давление, МПа		
		Библиотека skfuzzy	Вывод Мамдани данной библиотеки	Матричный вывод данной библиотеки
95	7,5	99.75568	99.77199	95.99999
55	4,2	65.83261	65.84039	59.09091
75	0,8	54.37548	54.37718	62.5
45	7,3	88.53293	84.23278	64.0

Из таблицы видно, что реализация алгоритма Мамдани в новой библиотеке соответствует реализации в существующей библиотеке, матричный алгоритм даёт аналогичные результаты в крайних точках. Разница значений между матричным алгоритмом и алгоритмом Мамдани обсуждается в работе [3].

Библиотека опубликована в официальном репозитории PyPI, установить можно с помощью команды `pip install fuzzy_matrix_mamdani`.

### Библиографический список

1. *Коньшова А.К., Серова Т.А.* Элементы теории нечетких множеств: Учебное пособие // Екатеринбург, Изд-во ГОУ ВПО «Российский государственный профессионально-педагогический университет», 2007 – 129с.
2. *Заде Л.А.* Понятие лингвистической переменной и его применение к принятию приближенных решений // М.: Мир, 1976, С. 70-75.
3. *Марценюк М.А.* Матричная реализация алгоритмов нечёткого вывода / М.А. Марценюк, В.Б. Поляков, И.П. Селетков // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – 2012. – Т. 6 (162). – С. 133–141.

### References

1. *Konyshova A.K., Serova T.A.* Elements of the theory of fuzzy sets: Textbook. Yekaterinburg, Publishing House of the State Educational Institution of Higher Education "Russian State Vocational Pedagogical University", 2007 – 129c. (In Russ.).
2. *Zadeh L.A.* The concept of a linguistic variable and its application to making approximate decisions. Moscow: Mir, 1976, pp. 70-75. (In Russ.).
3. *Martsenyuk M.A.* Matrix implementation of fuzzy inference algorithms / M.A. Martsenyuk, V.B. Polyakov, I.P. Seletkov // Scientific and Technical Bulletin of St. Petersburg State Polytechnic University. Computer science. Telecommunications. Management. – 2012. – Vol. 6 (162). – p. 133–141. (In Russ.).

## DEVELOPMENT OF A LIBRARY FOR IMPLEMENTING MATRIX FUZZY INFERENCE IN PYTHON

*Epishina Natalya V.*

Perm State University, 15, Bukireva st., Perm, Russia, 614990, epishina.nata@bk.ru

*Seletcov Ilya P.*

Perm State University, 15, Bukireva st., Perm, Russia, 614990, iseletkov@gmail.com

The paper briefly describes the principles of fuzzy logic, the features of matrix representation of operations on fuzzy variables. To simplify the construction of decision support systems using matrix representation, Authors made decision to develop a python library. To do this, authors analyzed existing libraries, revealed the advantages and popular fuzzy algorithms implemented. The following describes the process of development and publication in the official repository of a new library that allows calculations using matrix inference. Later on, paper provides the prove of correctness of the library by solving the demonstration problem of fuzzy inference for controlling the operation of a chemical reactor.

Keywords: fuzzy logic, matrix fuzzy inference, library for python, decision support system.

## **БИБЛИОТЕКА PYTHON ДЛЯ РАЗРАБОТКИ НЕЙРО-НЕЧЁТКИХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ (НА ПРИМЕРЕ ЗАДАЧИ ДИАГНОСТИКИ САХАРНОГО ДИАБЕТА)**

*Ермаков Виталий Алексеевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, blackcheshireshusband@yandex.ru

*Селетков Илья Павлович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, iseletkov@gmail.com.

В работе формулируются требования и описывается структура библиотеки для построения и обучения нейро-нечётких систем принятия решений на языке python. Производится обзор существующих нейро-нечётких моделей вывода, а также сравнительный анализ библиотек для построения данных систем на языке python. Описывается общая структура нейро-нечёткой системы вывода на основе модели ANFIS и приводится пример данной системы. Далее указывается используемый инструментарий и способы использования разработанной библиотеки для построения нейро-нечётких систем вывода. В конце производится тестирование разработанной библиотеки на примере задачи диагностики сахарного диабета. Оценивается влияние неполноты исходных данных на точность прогнозирования. Сравняется точность модели, основанной на экспертных знаниях, с моделью, сформированной автоматически методом кластеризации и трёхслойным персептроном. Делается вывод о необходимости наличия экспертных знаний при формировании базы правил.

Ключевые слова: Нейро-нечёткие сети, Python, Нечёткий логический вывод, Диагностика, Сахарный диабет.

### **Введение**

В настоящее время в самые различные сферы деятельности внедряются компьютерные системы поддержки принятия решений, например для выявления заболеваний и мониторинга пациентов в медицинской диагностике [1,2,3,4]. Однако системы, работающие по принципу «черного ящика», сложно интерпретировать, и, как следствие, эксперты могут воспринимать их с недоверием [5], в отличие от систем, основанных на нечёткой логике [6-12] и представленных в виде набора легко интерпретируемых правил, благодаря которым теория нечётких множеств может использоваться для поддержки принятия решений [13–17] наравне с нейронными сетями [18,19]. Недостатком систем нечёткого вывода является высокая степень зависимости от субъективности знаний экспертов. Одним из способов повысить объективность логических выводов является использование данных с технических систем телеметрии/объективного контроля. Для этого необходима разработка моделей, позволяющих учитывать и знания экспертов, и объективные обучающие данные. В связи с этим возникает необходимость разработки библиотеки для нейро-нечёткой поддержки принятия решений, объединяющей достоинства нечёткой логики и искусственного интеллекта.

## Постановка задачи

Необходимо разработать библиотеку для языка python, которая позволит строить и обучать нейро-нечёткие модели логического вывода.

Требования к библиотеке:

- возможность формирования моделей нейро-нечёткого вывода с различной структурой (числом слоёв, числом нейронов в слое, функциями активации, отсутствием или необучаемостью некоторых связей);
- возможность формирования/дообучения моделей логического вывода на основании наборов объективных данных;
- построение графиков, иллюстрирующих процесс логического вывода и обосновывающих формируемые заключения;
- интерфейс библиотеки должен быть аналогичен существующим библиотекам машинного обучения.

Для этого необходимо:

- изучить существующие нейро-нечёткие модели вывода;
- изучить существующие библиотеки для построения нейро-нечётких моделей вывода;
- разработать соответствующий требованиям функционал библиотеки;
- разработать модель нечёткого вывода и произвести тестирование библиотеки; продемонстрировав её эффективность на примере решения задачи оценки вероятности наличия диабета по данным исследований пациентов;
- опубликовать библиотеку в общем репозитории.

## Обзор существующих решений

В настоящее время распространены следующие адаптивные системы нечеткого логического вывода: ANFIS, NEFCON, GARIC, FALCON. Параметры функций принадлежности в них настраиваются автоматически, без участия экспертов, в процессе обучения на экспериментальных данных [20]. Однако существующие адаптивные системы являются узкоспециализированными и предназначены для получения нечетких правил в рамках традиционных моделей представления знаний (Мамдани, Сугено, Цукamoto и др.), что не позволяет использовать реализованные в них методы и алгоритмы для формирования эффективных нечетких моделей оценки состояния объектов в условиях неполноты исходных данных.

В работах [21,22] описана наиболее общая структура нейро-нечёткой сети на примере ANFIS, соответствующая классическим алгоритмам вывода и способная учитывать неполноту исходных данных, однако в ней отсутствует возможность оценки и настройки важности отдельных термов для совокупной истинности предпосылки, а также отдельных заключений для совокупной истинности термов. При этом стоит отметить, что нейро-нечёткие сети структуры ANFIS используются для разработки нейро-нечётких контроллеров [23].

На языке python для построения адаптивных систем нечёткого вывода существуют такие библиотеки, как `anfis` и `sanfis`, а также в среде MATLAB для этого используется `fuzzy logic toolbox`. Преимуществами данных библиотек являются:

- Возможность дообучения на новых данных;
- Возможность использования генетических алгоритмов обучения;
- Графическая иллюстрация вывода (в `fuzzy logic toolbox`).

Недостатки:

- Нет возможности настраивать функции активации;



- Ограничен выбор форм функций принадлежности;
- Отсутствует возможность создания классификаторов.

### Используемый инструментарий и разработанный функционал

Для разработки библиотеки решено использовать язык программирования Python, так как он наиболее широко применяется в сфере анализа данных и разработки систем поддержки принятия решений.

На рис. 1 приведена общая структура нейро-нечёткой сети, на примере диагностики диабета по двум признакам.

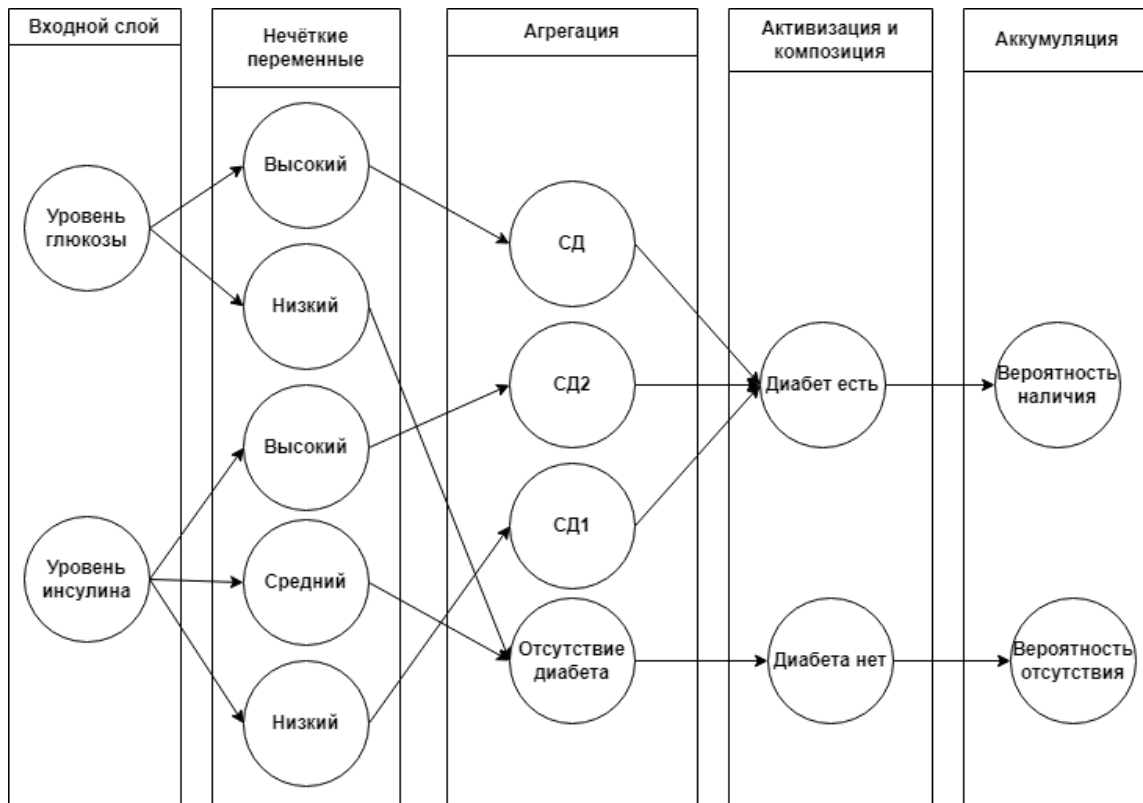


Рис. 1. Структура нейро-нечёткой сети для диагностики диабета

1. Слой является входным. Количество нейронов в нём соответствует числу входных лингвистических переменных соответствующей системы нечёткого вывода.

2. Слой осуществляет фазификацию значений входных параметров. Количество термов данного слоя соответствует совокупному количеству термов входных лингвистических переменных. В качестве его функции активации могут быть использованы гауссианы, треугольные и кусочно-линейные функции принадлежности. Матрица связей данного слоя отражает связи между соответствующими лингвистическими переменными и их термами. Матрица весов хранит параметры соответствующих функций принадлежности.

3. Слой осуществляет агрегирование, т.е. вычисляет значение истинности предпосылок всех правил. Количество нейронов данного слоя соответствует количеству различных антецедентов. В качестве его функции активации может использоваться нечёткая конъюнкция и дизъюнкция. Матрица связей данного слоя отражает наличие соответствующих термов в различных предпосылках. Матрица весов данного слоя отражает значимость термов в предпосылках.

4. Слой осуществляет активизацию и композицию, т.е. вычисляет значение истинности заключений всех правил и переводит её в истинность термов выходных лингвистических переменных. Количество нейронов данного слоя соответствует совокупному количеству термов выходных лингвистических переменных. В качестве его функции активации может использоваться нечёткая дизъюнкция и конъюнкция. Матрица связей данного слоя отражает связи между предпосылками и заключениями всех правил. Матрица весов отражает значимость предпосылок для соответствующих термов.

5. Слой осуществляет аккумуляцию, а также дефаззификацию если требуется. Количество нейронов соответствует числу выходных лингвистических переменных в случае дефаззификации и числу возможных классов в случае классификации. В качестве функций активации данного слоя могут использоваться линейная сумма, сумма полиномов Сугено или же softmax для нормировки при классификации. Матрица связей данного слоя отражает связи термов выходных лингвистических переменных с соответствующими лингвистическими переменными или классами в случае классификации.

Для тестирования разрабатываемой библиотеки была построена следующая модель нечёткого вывода:

Таблица 1. Реализация системы нечёткого вывода

Этап	Используемая функция
Фаззификация	Гауссиан
Агрегирование	Нечёткая матричная конъюнкция
Активизация и композиция	Нечёткая матричная дизъюнкция
Аккумуляция	Softmax

Далее среди имеющихся библиотек для построения подобных систем были выявлены следующие достоинства и недостатки:

Таблица 2. Достоинства и недостатки существующих решений

Название библиотеки	Достоинства	Недостатки
Anfis	Возможность дообучения/объективизации с использованием обучающей выборки. Возможность использования генетических алгоритмов для подбора оптимальных параметров. +Графическая иллюстрация логического вывода.	Нет возможности настраивать функции активации, выбирать количество промежуточных слоёв. Ограничен выбор форм функций принадлежности. Отсутствует возможность создания классификаторов.
Sanfis		
MATLAB Fuzzy-logic-toolbox (ANFIS)		

Для устранения существующих недостатков было решено разработать библиотеку, позволяющую варьировать количество слоёв, выбирать произвольные функции активации, строить как классические сети с дефаззификацией, так и классификаторы.

#### Взаимодействие с библиотекой

На рис. 2 представлена диаграмма классов, разработанной библиотеки Neuro\_Fuzzy\_EVAMortus.

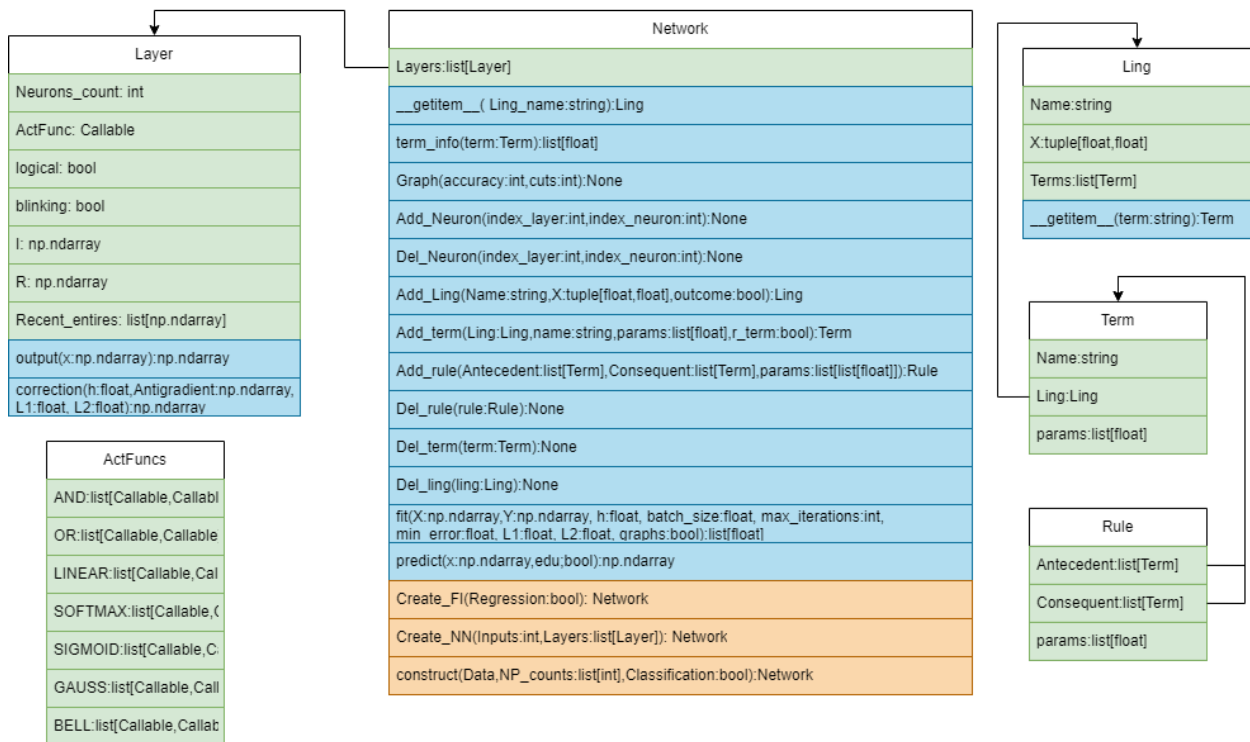


Рис. 2. Диаграмма классов библиотеки

Классы Ling, Term, Rule отвечают за хранение логической структуры нечёткого вывода, а классы Layer, Network, ActFuncs за хранение информации о генерируемой нейросети.

На данный момент разработано 2 способа построения нейросети с помощью данной библиотеки.

Первый способ заключается в автоматическом формировании нейро-нечёткой системы вывода, соответствующей модели, описанной в табл. 1, на основании статистических данных. Для этого с помощью кластеризации методом смеси гауссовых распределений определяется число термов каждой входной переменной, после чего формируется база правил методом декартова произведения всех терм множеств лингвистических переменных системы вывода. К недостаткам данного способа относится стохастический характер определения числа кластеров, зависящий от выбора начальных точек при кластеризации, а также неэффективный способ построения базы правил, приводящий к быстрорастущему числу вычислений. Для использования данного способа достаточно воспользоваться методом construct класса Network

Второй способ заключается задании системы нечёткого вывода вручную. Для этого существуют такие методы взаимодействия с нечёткой системой вывода класса Network, как Add\_Ling, Add\_term, Add\_rule и т.д. Данные методы позволяют учесть мнение экспертов при составлении системы вывода и имеют такие недостатки, как необходимость наличия экспертных знаний, трудоёмкость по сравнению с первым способом, выраженная в количестве строк кода и потраченного времени на анализ предметной области для формирования базы правил.

### Тестирование библиотеки на примере диагностики сахарного диабета

Для построения системы поддержки принятия решений в задаче оценки вероятности сахарного диабета использован набор данных [24]. В нём собраны результаты лабораторных исследований и соответствующие диагнозы для пациентов женского пола.

В качестве входных признаков для построения модели выбраны следующие показатели:

- уровень глюкозы в крови через 2 часа после нагрузки, ммоль/л;
- уровень инсулина в крови через 2 часа после нагрузки, мкЕд/мл;
- индекс массы тела, кг/м<sup>2</sup>;
- степень генетической предрасположенности, баллы;
- возраст, лет;
- количество перенесённых беременностей;
- систолическое артериальное давление, мм рт.ст.

В качестве выходных классов были выбраны соответственно отсутствие и наличие диабета. В соответствии с предметной областью была разработана экспертная база правил.

В табл. 3 представлены результаты обучения различных моделей на данной выборке методом обратного распространения ошибки.

Таблица 3. Результаты обучения моделей

Модель	Число итераций	LogLoss	Точность
Автоматически сгенерированный Нейро-нечёткий классификатор	450	0,48	0,73
Нейро-нечёткий-классификатор	160	0,44	0,76
Трёхслойный персептрон (7,14,2)	200	0,59	0,67

Также, в ходе тестирования была дана оценка степени влияния различных входных признаков на точность предсказания модели, основанной на экспертных знаниях.

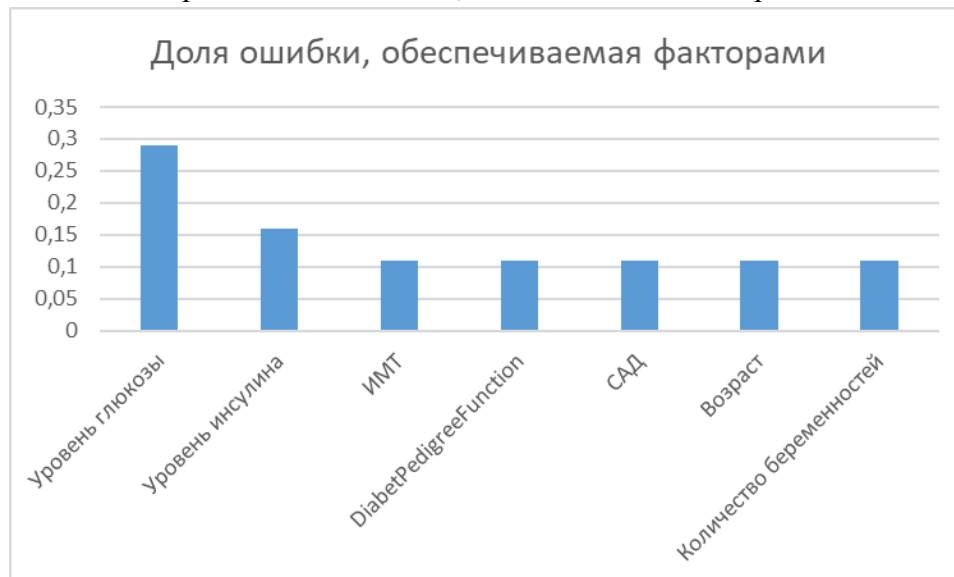


Рис. 3. Степень влияния факторов на точность прогноза

Исходя из результатов тестирования можно сказать, что наиболее быстро и эффективно обучается модель нейро-нечёткого классификатора, основанная на экспертных знаниях, однако даже классификатор, построенный автоматически на основании статистических данных оказался точнее стандартного алгоритма постановки диагноза. [25,26] Влияние неполноты исходных данных на точность прогнозов оказалось экспоненциальным, а наиболее значимыми факторами оказались уровень глюкозы, инсулина и ИМТ.

## Заключение

В данной работе был проведён анализ существующих нейро-нечётких моделей вывода, а также библиотек для их построения. Описана общая структура нейро-нечёткой модели вывода и приведён пример модели. Описана структура и способы использования библиотеки Neuro\_Fuzzy\_EVAMortus. Описаны результаты обучения построенной модели на примере задачи диагностики диабета.

## Библиографический список

1. *Fernandes M.* Clinical Decision Support Systems for Triage in the Emergency Department using Intelligent Systems: a Review / M. Fernandes [et al.] // *Artificial Intelligence in Medicine*. – 2020. – Vol. 102. – Clinical Decision Support Systems for Triage in the Emergency Department using Intelligent Systems. – P. 101762.
2. *Mustaqeem A.* A modular cluster based collaborative recommender system for cardiac patients / A. Mustaqeem, S.M. Anwar, M. Majid // *Artificial Intelligence in Medicine*. – 2020. – Vol. 102. – P. 101761.
3. *Souza-Pereira, L.* Clinical decision support systems for chronic diseases: A systematic literature review / L. Souza-Pereira, N. Pombo, S. Ouhbi, V. Felizardo, N. Garcia // *Comput Methods Program Biomed*. – 2020. – Vol. 195. – P. 105565.
4. *Olakotan, O.O.* Evaluating the alert appropriateness of clinical decision support systems in supporting clinical workflow / O.O. Olakotan, M.M. Yusof // *Journal Biomedical Informatics*. – 2020. – Vol. 106. – P. 103453.
5. *MsRae M.P.* Cardiac ScoreCard: A diagnostic multivariate index assay system for predicting a spectrum of cardiovascular disease / M.P. MsRae, B. Bozkurt, C.M. Ballantyne, X.Sanchez, N. Christodoulide, G.Simmons et al. // *Expert Systems with Applications: An International Journal*. – 2016. – Vol. 54. – P. 136-147.
6. *Thukral S.* Versatility of fuzzy logic in chronic diseases: A review / S. Thukral, V. Rana // *Medical Hypotheses*. – 2019. – Vol. 122. – P. 150-156.
7. *Gadaras I.* An interpretable fuzzy rule-based classification methodology for medical diagnosis / I. Gadaras, L. Mikhailov // *Artificial Intelligence in Medicine*. – 2009. – Vol. 47(1). – P. 25-41.
8. *Mokeddem S.A.* A fuzzy classification model for myocardial infarction risk assessment / S.A. Mokeddem // *Applied Intelligens*. – 2018. – Vol. 48. – P. 1233-1250.
9. *Nauck D.* Obtaining interpretable fuzzy classification rules from medical data / D. Nauck, R. Kruse // *Artificial Intelligence in Medicine*. – 1999. – Vol. 16(2). – P. 149-169.
10. *Kalantari A.* Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions / A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, A.T. Chronopoulos // *Neurocomputing*. – 2018. – Vol. 276. – P. 2-22.
11. *Jemal H.* Enhanced decision support systems in intensive care unit based on intuitionistic fuzzy sets / H. Jemal, Z. Kechaou, M.B. Ayed // *Advances in Fuzzy Systems*. – 2017. – Vol. 2017. – P. 1-8.
12. *Pota M.* Designing rule-based fuzzy systems for classification in medicine / M. Pota, M. Esposito, G. Pietro // *Knowledge-Based Systems*. – 2017. – Vol. 124. – P. 105-132.
13. *Minutolo A.* A fuzzy framework for encoding uncertainty in clinical decision-making / A. Minutolo, M. Esposito, G. Pietro // *Knowledge-Based Systems*. – 2016. – Vol. 98. – P. 95-116.
14. *Ahmadi H.* Diseases diagnosis using fuzzy logic methods: A systematic and meta-analysis review / H. Ahmadi, M. Gholamzadeh, L. Shahmoradi, M. Nilashi, P. Rashvand // *Comput Methods Program Biomed*. – 2018. – Vol. 161. – P. 145-172.

15. *Kour H.* Usage and implementation of neuro-fuzzy systems for classification and prediction in the diagnosis of different types of medical disorders: a decade review / H. Kour, J. Manhas, V. Sharma // *Artificial Intelligence in Medicine.* – 2020 – Vol. 53(7). – P. 4651-4706.
16. *Sajadi N.A.* Diagnosis of hypothyroidism using a fuzzy rule-based expert system / N.A. Sajadi, S. Borzouei, H. Mahjub, M. Farhadian // *Clinical Epidemiology and Global Health.* – 2019. – Vol. 7. – P. 519-524.
17. *Arji G.* Fuzzy logic approach for infectious disease diagnosis: A methodical evaluation, literature and classification / G. Arji, H. Ahmadi, M. Nilashi, T.A. Rashid, O.H. Ahmed, N. Aljojo, A. Zainol // *Biocybernetics and Biomedical Engineering.* – 2019. – Vol. 39(4). – P. 937-955.
18. *Amato, F.* Artificial neural networks in medical diagnosis / F. Amato, A. Lopez, E.M. Pena-Mendez, P. Vanhara, A. Hampl, J. Havel // *Journal of applied biomedicine.* – 2013. – Vol. 11. – P.47-58.
19. *Jiang J.* Medical knowledge embedding based on recursive neural network for multi-disease diagnosis / J. Jiang, H. Wang, J. Xie, X. Guo, Y. Guan, Q.Yu // *Artificial Intelligence in Medicine.* – 2020. – Vol. 103. – P. 101772.
20. *Yarushkina N.* Soft computing and complex system analysis / N. Yarushkina // *International Journal of General Systems.* – 2001. – Vol. 30 (1). – P. 71–88.
21. *Катасёв А.С.* Нейронечеткая модель формирования нечетких правил для оценки состояния объектов в условиях неопределенности / А.С. Катасёв // *Компьютерные исследования и моделирование* – 2019. – Т. 11. – № 3. – С. 477–492
22. *Хижняков Ю.Н.* Алгоритмы нечеткого, нейронного и нейро-нечеткого правления в системах реального времени: учеб. Пособие / Ю.Н. Хижняков – Пермь: ПНИПУ, 2013. – С. 86-150.
23. *Муравьева Е.А.* Разработка метода адаптации нечеткого регулятора, предназначенного для регулирования технологических параметров / Е. А. Муравьева, Д. С. Радакина, М. И. Шарипов // *Вестник Чувашского университета.* –2018. – № 1. – С. 137-145.
24. Набор данных для диагностирования диабета // Kaggle: Your Machine Learning and Data Science Community [Электронный ресурс]. – URL: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database> (дата обращения: 03.05.2023)
25. Рекомендации экспертов Всероссийского научного общества кардиологов по диагностике и лечению метаболического синдрома второй пересмотр // *Практическая медицина.* – 2010. – №44. [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/rekomendatsii-ekspertov-vserossiyskogo-nauchnogo-obschestva-kardiologov-po-diagnostike-i-lecheniyu-metabolicheskogo-sindroma-vtoroy-1> (дата обращения: 25.05.2023).
26. *Аметов А.С.* Сахарный диабет 2 типа. Проблемы и решения: учебное пособие / А.С. Аметов – 3-е издание, переработано и дополнено – Москва: ГЭОТАР-Медиа. – 2015. – Т. 1. – С.352.

## References

1. *Fernandes M.* Clinical Decision Support Systems for Triage in the Emergency Department using Intelligent Systems: a Review / M. Fernandes [et al.] // *Artificial Intelligence in Medicine.* – 2020. – Vol. 102. – Clinical Decision Support Systems for Triage in the Emergency Department using Intelligent Systems. – P. 101762.
2. *Mustaqeem, A.* A modular cluster based collaborative recommender system for cardiac patients / A. Mustaqeem, S.M. Anwar, M. Majid // *Artificial Intelligence in Medicine.* – 2020. – Vol. 102. – P. 101761.

3. *Souza-Pereira, L.* Clinical decision support systems for chronic diseases: A systematic literature review / L. Souza-Pereira, N. Pombo, S. Ouhbi, V. Felizardo, N. Garcia // *Comput Methods Program Biomed.* – 2020. – Vol. 195. – P. 105565.
4. *Olakotan, O.O.* Evaluating the alert appropriateness of clinical decision support systems in supporting clinical workflow / O.O. Olakotan, M.M. Yusof // *Journal Biomedical Informatics.* – 2020. – Vol. 106. – P. 103453.
5. *MsRae, M.P.* Cardiac ScoreCard: A diagnostic multivariate index assay system for predicting a spectrum of cardiovascular disease / M.P. MsRae, B. Bozkurt, C.M. Ballantyne, X.Sanchez, N. Christodoulide, G.Simmons et al. // *Expert Systems with Applications: An International Journal.* – 2016. – Vol. 54. – P. 136-147.
6. *Thukral, S.* Versatility of fuzzy logic in chronic diseases: A review / S. Thukral, V. Rana // *Medical Hypotheses.* – 2019. – Vol. 122. – P. 150-156.
7. *Gadaras, I.* An interpretable fuzzy rule-based classification methodology for medical diagnosis / I. Gadaras, L. Mikhailov // *Artificial Intelligence in Medicine.* – 2009. – Vol. 47(1). – P. 25-41.
8. *Mokeddem S.A.* A fuzzy classification model for myocardial infarction risk assessment / S.A. Mokeddem // *Applied Intelligens.* – 2018. – Vol. 48. – P. 1233-1250.
9. *Nauck, D.* Obtaining interpretable fuzzy classification rules from medical data / D. Nauck, R. Kruse // *Artificial Intelligence in Medicine.* – 1999. – Vol. 16(2). – P. 149-169.
10. *Kalantari, A.* Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions / A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, A.T. Chronopoulos // *Neurocomputing.* – 2018. – Vol. 276. – P. 2-22.
11. *Jemal, H.* Enhanced decision support systems in intensive care unit based on intuitionistic fuzzy sets / H. Jemal, Z. Kechaou, M.B. Ayed // *Advances in Fuzzy Systems.* – 2017. – Vol. 2017. – P. 1-8.
12. *Pota, M.* Designing rule-based fuzzy systems for classification in medicine / M. Pota, M. Esposito, G. Pietro // *Knowledge-Based Systems.* – 2017. – Vol. 124. – P. 105-132.
13. *Minutolo, A.* A fuzzy framework for encoding uncertainty in clinical decision-making / A. Minutolo, M. Esposito, G. Pietro // *Knowledge-Based Systems.* – 2016. – Vol. 98. – P. 95-116.
14. *Ahmadi, H.* Diseases diagnosis using fuzzy logic methods: A systematic and meta-analysis review / H. Ahmadi, M. Gholamzadeh, L. Shahmoradi, M. Nilashi, P. Rashvand // *Comput Methods Program Biomed.* – 2018. – Vol. 161. – P. 145-172.
15. *Kour, H.* Usage and implementation of neuro-fuzzy systems for classification and prediction in the diagnosis of different types of medical disorders: a decade review / H. Kour, J. Manhas, V. Sharma // *Artificial Intelligence in Medicine.* – 2020 – Vol. 53(7). – P. 4651-4706.
16. *Sajadi, N.A.* Diagnosis of hypothyroidism using a fuzzy rule-based expert system / N.A. Sajadi, S. Borzouei, H. Mahjub, M. Farhadian // *Clinical Epidemiology and Global Health.* – 2019. – Vol. 7. – P. 519-524.
17. *Arji, G.* Fuzzy logic approach for infectious disease diagnosis: A methodical evaluation, literature and classification / G. Arji, H. Ahmadi, M. Nilashi, T.A. Rashid, O.H. Ahmed, N. Aljojo, A. Zainol // *Biocybernetics and Biomedical Engineering.* – 2019. – Vol. 39(4). – P. 937-955.
18. *Amato, F.* Artificial neural networks in medical diagnosis / F. Amato, A. Lopez, E.M. Pena-Mendez, P. Vanhara, A. Hampl, J. Havel // *Journal of applied biomedicine.* – 2013. – Vol. 11. – P.47-58.
19. *Jiang, J.* Medical knowledge embedding based on recursive neural network for multi-disease diagnosis / J. Jiang, H. Wang, J. Xie, X. Guo, Y. Guan, Q.Yu // *Artificial Intelligence in Medicine.* – 2020. – Vol. 103. – P. 101772.

20. *Yarushkina, N.* Soft computing and complex system analysis / N. Yarushkina // International Journal of General Systems. – 2001. – Vol. 30 (1). – P. 71–88.
21. *Katasev A.S.* A neural fuzzy model of the formation of fuzzy rules for assessing the state of objects under uncertainty / A.S. Katasev // Komp'yuternye issledovaniya i modelirovanie– 2019. – Vol. 11. – No. 3. – P. 477-492. (In Russ.).
22. *Khizhnyakov, Yu.N.* Algoritmy nechetkogo, nejronnogo i nejro-nechetkogo pravleniya v sistemah real'nogo vremeni: ucheb. Posobie / D.N.Nizhnyakov – Perm: PNPu, 2013. – P. 86-150. (In Russ.).
23. *Muravyeva, E.A.* Development of a method of adaptation of a fuzzy controller designed to regulate technological parameters / E. A. Muravyeva, D. S. Radakina, M. I. Sharipov // Vestnik CHuvashskogo universiteta. –2018. – № 1. – P. 137-145. (In Russ.).
24. A data set for the diagnosis of diabetes // Kaggle: Your Machine Learning and Data Science Community [Electronic resource]. – URL: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>.
25. Recommendations of experts of the All-Russian Scientific Society of Cardiologists on the diagnosis and treatment of metabolic syndrome second revision // Prakticheskaya medicina.– 2010. – №44. [Electronic resource]. – URL: <https://cyberleninka.ru/article/n/rekomendatsii-ekspertov-vserossiyskogo-nauchnogo-obschestva-kardiologov-po-diagnostike-i-lecheniyu-metabolicheskogo-sindroma-vtoroy-1>. (In Russ.).
26. *Ametov, A.S.* Saharnyj diabet 2 tipa. Problemy i resheniya: uchebnoe posobie / A.S. Ametov – 3rd edition, revised and supplemented – Moscow: GEOTAR-Media. – 2015. – T. 1. – P.352. (In Russ.).

## **PYTHON LIBRARY FOR THE DEVELOPMENT OF NEURO-FUZZY DECISION SUPPORT SYSTEMS (BY THE EXAMPLE OF THE DIAGNOSIS OF DIABETES MELLITUS)**

*Ermakov Vitalii A.*

Perm State University, 15 Bukireva St., 614990, Perm, blackcheshirehusband@yandex.ru

*Seletkov Ilya P.*

Perm State University, 15 Bukireva St., 614990, Perm, iseletkov@gmail.com.

The paper formulates the requirements and describes the structure of the library for building and training neuro-fuzzy decision-making systems in the python language. A review of existing neuro-fuzzy inference models is made, as well as a comparative analysis of libraries for building these systems in the python language. The general structure of a neuro-fuzzy inference system based on the ANFIS model is described and an example of this system is given. Next, the tools used and ways to use the developed library for building neuro-fuzzy inference systems are indicated. At the end, the developed library is tested on the example of the task of diagnosing diabetes mellitus. The influence of incompleteness of initial data on the accuracy of forecasting is estimated. The accuracy of the model based on expert knowledge is compared with the model generated automatically by the clustering method and the three-layer perceptron. It is concluded that it is necessary to have expert knowledge in the formation of the rule base.

Keywords: Neuro-fuzzy networks, Python, Fuzzy inference, Diagnosis, Diabetes mellitus.



## WEB-ИНТЕРФЕЙС СИСТЕМЫ ГРАФИЧЕСКОГО МОДЕЛИРОВАНИЯ

*Захаров Алексей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, az9311159@gmail.com

*Селетков Илья Павлович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, iseletkov@gmail.com

В работе формулируется задача построения web-интерфейса системы графического моделирования, позволяющей моделировать бизнес-процессы, организационную структуру и архитектуру данных предприятий. Производится обзор и сравнительный анализ существующих готовых решений, а также современных инструментов и технологий для построения таких систем. Далее выделяются основные этапы разработки новой клиент-серверной системы такого класса. В конце предлагается архитектура web-интерфейса системы, примеры реализации некоторых окон.

Ключевые слова: web-интерфейс, vuejs, vuetify, vueflow, keycloak, среда моделирования.

### **Введение**

В настоящее время поддержание эффективности работы и конкурентоспособности предприятий связано с широким использованием информационных технологий, в т.ч. в процессах описания и оптимизации бизнес-архитектуры (клиентская база более 6000 компаний РФ) [1]. В этой связи, немаловажное значение имеет процесс описания и оптимизации бизнес-архитектуры, который позволяет определить ключевые бизнес-процессы и их взаимосвязи, выявить слабые места и узкие места в работе предприятия, а также разработать стратегии по их устранению и оптимизации.

Для описания бизнес-процессов на рынке существует большое количество ПО, в основном, иностранного производства, отличающегося проработанностью, глубиной, детальностью методологических подходов и решений, а также высокой стоимостью.

К сожалению, использование программного обеспечения иностранного производства в современных условиях может быть связано с риском попадания под действие санкций. Кроме того, использование зарубежных продуктов может представлять риск для безопасности информации и конфиденциальности данных.

В связи с этим, многие компании все чаще обращают внимание на ПО отечественного производства, которое не подвержено риску санкций и обеспечивает надежную защиту конфиденциальной информации. Кроме того, использование отечественного ПО может стать важным фактором в поддержании экономической независимости страны и развитии отечественной ИТ-инфраструктуры.

В свою очередь, существующие аналоги отечественного производства не всегда соответствуют всем актуальным потребностям бизнеса.

Целью данного исследования является уменьшение трудозатрат на моделирование бизнес-процессов и архитектуры предприятий, а также уменьшение негативных последствий

санкций, которые могут быть наложены на использование иностранного программного обеспечения.

### **Постановка задачи**

Требуется разработать веб-интерфейс системы моделирования логической архитектуры предприятия в части построения графических схем, который должен позволять:

- выбирать, создавать, удалять, настраивать репозитории;
- отображать структуру, создавать, удалять, настраивать логические «папки» в репозитории;
- создавать, редактировать, удалять графические схемы;
- создавать, удалять, редактировать атрибуты элементов графических схем.

### **Обзор существующих решений**

Таблица 1. Достоинства и недостатки существующих систем графического моделирования

Название	Достоинства	Недостатки
Aris	Позволяет моделировать, анализировать, оптимизировать бизнес-процессы;  обширный функционал;  встроенные механизмы автоматизации и интеграции.	Высокая стоимость;  требует определенных навыков и знаний для работы с ним;  является зарубежным продуктом;  может быть нестабильным при работе с большим количеством данных.
Mega Hopex	Позволяет моделировать различные аспекты бизнеса;  высокая степень гибкости и конфигурируемости;  встроенные механизмы автоматизации и интеграции.	Высокая стоимость;  сложный интерфейс;  некоторые функции недоступны без настройки и интеграции; является зарубежным продуктом
Sila Union	Обладает широким спектром возможностей для моделирования бизнес-процессов;  интуитивно понятный интерфейс и простота использования; поддерживает интеграцию с другими системами;  отечественный продукт.	Некоторые функции ограничены в бесплатной версии;  может быть нестабильным при работе с большим количеством данных.
Business Studio	Интуитивно понятный интерфейс и простота использования; поддерживает различные виды диаграмм и моделей; отечественный продукт.	Диаграммы не связаны между собой в единую модель;  не подходит для сложных и крупных проектов; ограниченный функционал для интеграции с другими системами;  требует для работы установки платного зарубежного ПО MS Visio.

Выявленные выше недостатки являются проблемами, которые могут негативно сказаться на работе организации. Для решения этих проблем требуется разработка нового Программного обеспечения с целью устранения существующих недостатков и внедрения новых функций и возможностей.

## Выбор инструментария

Для разработки клиентской части предлагается использовать Vue.js – это прогрессивный фреймворк для разработки пользовательских интерфейсов. Он позволяет создавать компоненты, которые могут быть использованы повторно в приложении, что упрощает разработку и сопровождение [2]. Также предлагается использовать готовые элементы графического интерфейса из библиотеки vuetify [3].

Исходный код предлагается писать на языке TypeScript – это расширение языка JavaScript, которое добавляет статическую типизацию. Статическая типизация помогает разработчикам избежать ошибок типов, что в свою очередь упрощает сопровождение кода [4].

Для реализации авторизации в системе предлагается использовать сервис Keycloak [5].

Для реализации процесса моделирования в системе предлагается использовать Vueflow – это библиотека компонентов для Vue.js, предназначенная для создания пользовательских интерфейсов и визуальных редакторов с помощью графовых элементов [6].

## Архитектура системы

Для решения поставленной задачи авторами предлагается архитектура, представленная на рис. 1. В ней подразумевается распределенная клиент-серверная система, состоящая из нескольких компонентов.

- Пользователь создаёт, редактирует и просматривает данные через web-интерфейс.
- WEB-интерфейс использует протокол REST HTTPS для обмена данными с серверной частью.
- Единая точка входа/балансировщик нагрузки обеспечивает распределение запросов, управления маршрутизацией и балансировкой нагрузки между различными экземплярами рабочих сервисов системы.
- Сервис авторизации: обеспечивает аутентификацию и авторизацию пользователей. После успешной аутентификации пользователей, сервис авторизации выдает токены доступа, которые используются для доступа к другим компонентам системы.
- Сервис обнаружения обеспечивает обнаружение друг другом сервисов системы.
- Сервис обмена сообщениями обеспечивает асинхронную связь между различными компонентами системы.
- Сервис данных моделирования обеспечивает анализ, моделирование и визуализацию бизнес-данных.
- Планировщик задач обеспечивает планирование и выполнение задач для определения порядка выполнения задач и оптимального распределения ресурсов.
- Сервис исполнения скриптов обеспечивает выполнение скриптов и других средств автоматизации, обработку результатов выполнения.
- СУБД обеспечивает хранение и управление данными.

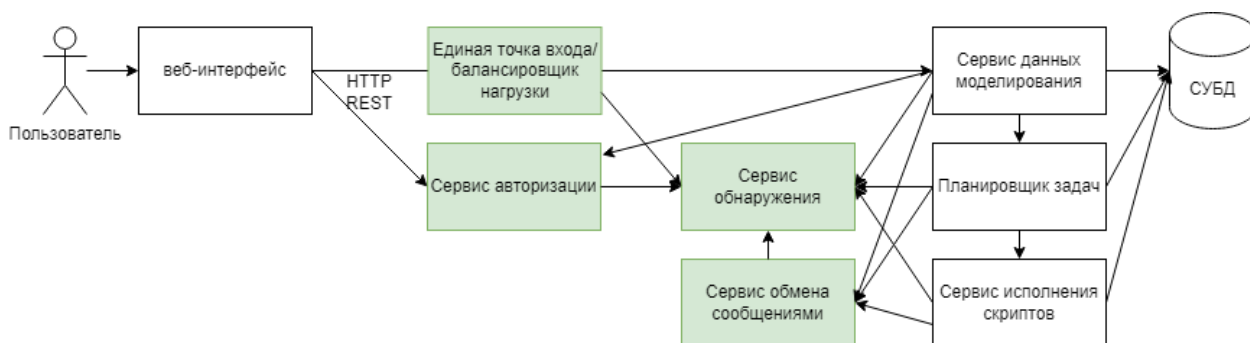


Рис. 1. Архитектура системы

На рисунке зелёным цветом выделены компоненты, разработка которых не требуется – необходимо только настроить готовое свободно распространяемое ПО.

На рис. 2 приведена архитектура клиентской части системы. Для хранения данных на клиентской стороне используется библиотека vue store, код процесса авторизации вынесен в отдельный вспомогательный файл CServiceKeycloak, код взаимодействия с API серверной части вынесен в отдельный файл CServiceAPIModels, код работы страницы построения диаграмм находится в скрипте CComponentDesigner.

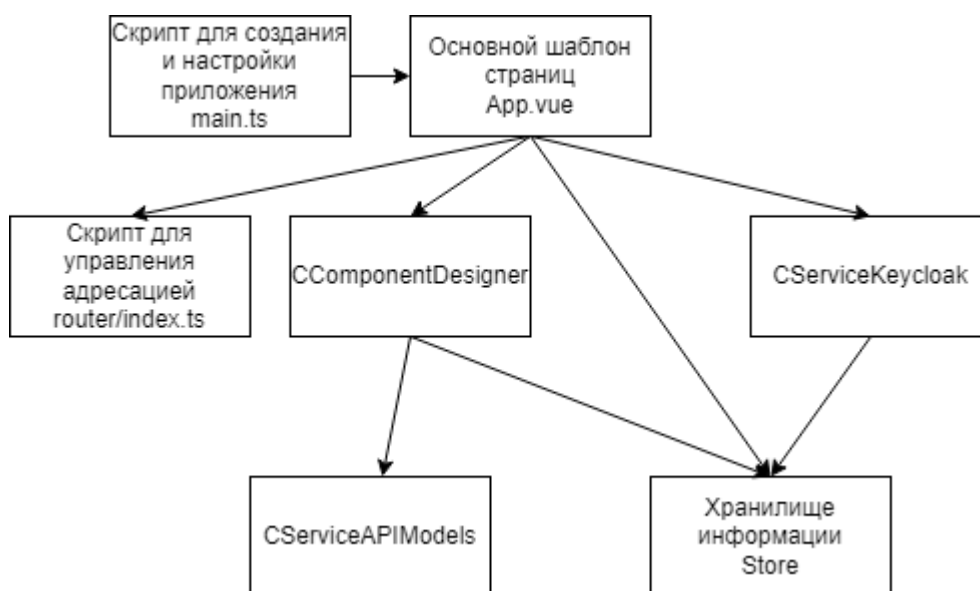


Рис. 2. Архитектура клиентской части

### Авторизация пользователей с использованием сервиса Keycloak

Для снижения риска перехвата учётных данных пользователей при передаче и хранении в системе, для исключения необходимости разработки страниц ввода учётных данных, восстановления забытого пароля, управления полномочиями пользователей, авторами предложено использовать сервис с открытым исходным кодом Keycloak [6]. Этот сервис может быть установлен на собственной инфраструктуре, расположенной на территории Российской Федерации, тем самым обеспечивается выполнение требований законодательства.

Для простоты взаимодействия с keycloak из клиентской части системы на языке JavaScript можно использовать библиотеку «keycloak-js». Тогда процесс авторизации можно выполнить с помощью нескольких действий. Сначала создаётся объект класса Keycloak.

```
this.keycloak = new Keycloak()
```

Далее собственно запрос авторизации реализуется следующим образом.

```
this.keycloak
  .init({
    onLoad: "login-required",
    checkLoginIframe: false
  })
  .then(function (authenticated) {
    authenticated ? onAuthenticatedCallback() : alert("non authenticated");
  })
```

Для указания места расположения сервиса keycloak требуется создать файл /public/keycloak.js следующего содержания.

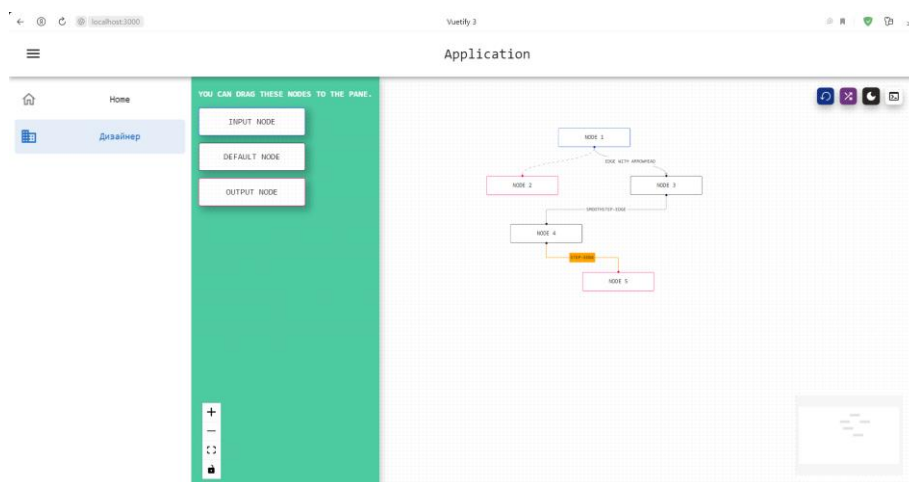
```
{
  "realm": "public",
  "auth-server-url": "здесь адрес",
  "ssl-required": "external",
  "resource": "здесь наименование приложения",
  "public-client": true,
  "confidential-port": 0,
  "client-id": "здесь наименование приложения"
}
```

В результате в системе появляется возможность производить авторизацию пользователей без необходимости вручную разрабатывать все необходимые для этого окна.

### Графическое моделирование

Для реализации процесса построения графических схем, диаграмм, графов, которые можно использовать для моделирования бизнес процессов, организационной структуры, архитектур данных и информационных систем, авторами предлагается использовать компонент vue flow, который интегрируется с остальными компонентами экосистемы vue, и для которого на официальном сайте опубликованы готовые примеры реализации страниц с палитрами инструментов и рабочей областью.

Vueflow – это гибкий и мощный инструмент для создания интерактивных диаграмм и потоковых графиков в веб-приложениях на Vue.js [3]. Пример использования библиотеки vueflow приведён на рис. 3.



**Рис. 3. Архитектура клиентской части**

### **Заключение**

В заключении, описанный в статье проект был разработан на Vue.js с использованием TypeScript в клиентской части. В проекте также была выполнена интеграция с Keycloak для обеспечения безопасности и управления доступом, а также с Vueflow для создания интерактивных диаграмм и потоковых графиков.

### **Библиографический список**

1. Официальный сайт компании ООО «БИТЕК» (Бизнес-инжиниринговые технологии) [Электронный ресурс]. – URL: <http://www.betec.ru/index.php?id=6&sid=118> (дата обращения: 26.04.2023).
2. Официальный сайт Vue.js (Введение – Vue.js) [Электронный ресурс]. – URL: <https://ru.vuejs.org/v2/guide/index.html> (дата обращения: 26.04.2023).
3. Официальный сайт Vuetify (Введение во Vuetify) [Электронный ресурс]. – URL: <https://vuetifyjs.com/en/introduction/why-vuetify/> (дата обращения: 26.04.2023).
4. Официальный сайт TypeScript Documentation (Документация) [Электронный ресурс]. – URL: <https://www.typescriptlang.org/docs/> (дата обращения: 26.04.2023).
5. Официальный сайт Keycloak (Документация) [Электронный ресурс]. – URL: <https://www.keycloak.org/> (дата обращения: 26.04.2023).
6. Официальный сайт Vue Flow (Введение | Vue Flow) [Электронный ресурс]. – URL: <https://vueflow.dev/guide/> (дата обращения: 26.04.2023).

### **References**

1. Official site of BITEK Ltd. [Electronic resource]. – URL: <http://www.betec.ru/index.php?id=6&sid=118> (accessed: 26.04.2023) (In Russ.).
2. Vue.js [Electronic resource]. – URL: <https://ru.vuejs.org/v2/guide/index.html> (accessed: 26.04.2023).
3. Vuetify [Electronic resource]. – URL: <https://vuetifyjs.com/en/introduction/why-vuetify/> (accessed: 26.04.2023).
4. TypeScript Documentation [Electronic resource]. – URL: <https://www.typescriptlang.org/docs/> (accessed: 26.04.2023).
5. Keycloak Documentation [Electronic resource]. – URL: <https://www.keycloak.org> (accessed: 26.04.2023).
6. Vue Flow [Electronic resource]. – URL: <https://vueflow.dev/guide/> (accessed: 26.04.2023).

## WEB INTERFACE OF SYSTEM FOR GRAPHIC MODELLING

*Zakharov Alexey I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, az9311159@gmail.com

*Seletkov Ilya P.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, iseletkov@gmail.com

This work formulates objective to construct web-interface for graphic modelling system which is useful for modelling of business processes, organizational structure, architecture of data. The article contains overview of existing ready-to-use solutions and also actual instruments and technologies for constructing such systems. There is list of main stages of development of new client-server system for modelling purpose. Authors propose architecture of web-interface as well as some screenshots of windows at the end of paper.

Keywords: web interface, vuejs, vuetify, vueflow, keycloak, modelling environment.

## **ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ПРОЕКТОВ ОРГАНИЗАЦИИ**

*Зольников Виталий Валерьевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, vvvzolnikov@gmail.com

*Кузаев Айдар Файзуллович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kuzaev@pspu.ru

Рассматриваются проектирование и разработка информационной системы для учета проектов в организации. На основе анализа предметной области выделяются основные функциональные требования. Производится обоснованный выбор средств проектирования и разработки приложения. Проектирование осуществляется в нотации UML, выбирается CASE-средство объектно-ориентированного анализа и моделирования StarUML. В результате проектирования информационной системы описываются поведение информационной системы, взаимодействие системы с пользователем, документируются основные процессы. На основе описанных процессов и функциональных требований разрабатывается прототип пользовательского интерфейса.

Ключевые слова: проект, информационная система, проектирование, учет проектов, разработка.

Актуальность данной темы обусловлена тем, что в настоящее время происходит автоматизация всевозможных процессов на промышленных предприятиях, поэтому объем работ для организаций, занимающихся проектированием, увеличивается. В связи с этим, закономерно возникает необходимость в продукте, который обеспечил бы удобную работу с содержимым проекта и процессом его наполнения. Таким образом, в данной сфере требуется информационная система, которая могла бы централизованно хранить данные, а также позволяла ускорить процесс проектирования, в том числе на этапе предварительного планирования путем автоматизации рутинных процессов наполнения проекта. Такая система поможет повысить скорость разработки и качество проектной документации.

Объектом исследования в данной работе являются проекты в организации.

Предмет исследования – автоматизация и систематизация управления проектами на предприятии.

Целью является проектирование информационной системы учета проектов организации с возможностью обработки содержимого проекта и последующей выгрузкой данных в виде отчетов.

Задачи, которые необходимо выполнить для достижения поставленной цели:

- анализ существующего программного обеспечения и систем;
- выбор средств проектирования и разработки;
- проектирование модели информационной системы;
- разработка прототипа пользовательского интерфейса.



Логика работы разрабатываемой информационной системы отражена на диаграмме ниже (см. рис. 1). Данная диаграмма была построена с помощью CASE-средства StarUML [1]. StarUML – это программный инструмент визуального моделирования с открытым исходным кодом, который поддерживает стандартизованный язык графического описания UML (Unified Modeling Language) [2] для моделирования систем и программного обеспечения.

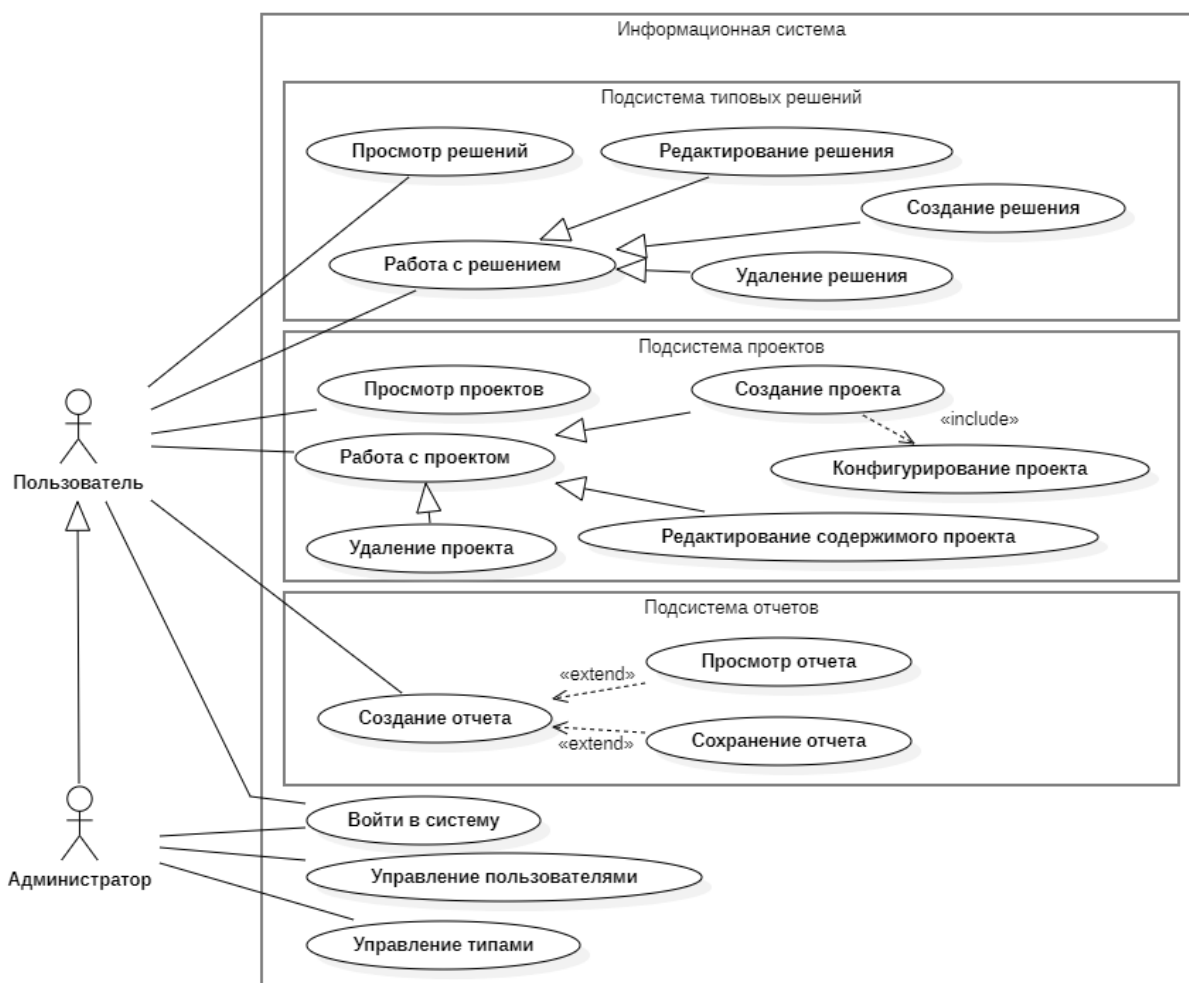


Рис. 1. Диаграмма прецедентов информационной системы

Уточним, кем являются пользователи и администраторы, отмеченные на диаграмме. Так как информационная система проектируется для нужд проектной организации, то пользователями будут инженеры-проектировщики, занимающиеся разработкой проектной документации.

Администраторами в данном случае являются непосредственно администраторы информационной системы, которые имеют полный доступ к базам данных, а также могут изменять структуру данных. Задача администратора – добавление и удаление пользователей в систему. Так как данные в информационной системе не являются публичными, то пользователи не должны иметь возможность зарегистрироваться, добавить нового пользователя и выполнять другие действия. Данные манипуляции должны осуществляться только администратором.

Пользователь имеет возможность войти в систему, создать, изменить или удалить элемент проекта. Также он может создать, сконфигурировать проект, а также редактировать наполнение проекта. Помимо этого, пользователь может работать с решениями: аналогично работе с проектом создавать, изменять и удалять решения. Пользователь может создать отчет и, по желанию, сохранить его или просмотреть.

Администратор системы отличается от пользователя тем, что имеет следующие доступные действия: управление пользователями, которые имеют доступ к системе, управление типами объектов проектов, а также полный доступ к базе данных.

Управление типами подразумевает под собой изменение и добавление сущностей в базах данных.

Особое внимание уделяется проектированию схемы базы данных. Для увеличения гибкости структуры данных был применен шаблон «Супер-тип и подтипы». В нашем случае этот шаблон можно удобно использовать, так как информационная система имеет нефиксированный набор сущностей и изменяемую структуру проекта. В данный момент в составе проекта представлены такие сущности, как комплект рабочей документации, система управления, оборудование, контур, устройство, сигнал. Каждая отдельная сущность будет являться подтипом, в то время как элемент проекта – супер-типом. Данный шаблон позволяет нам заложить возможность расширения состава проекта без кардинального изменения базы данных. На рис. 2 отображен данный шаблон на примере части базы данных, связанной с типовыми решениями.

Также был использован шаблон «Расширяемые атрибуты». Для каждой сущности были созданы дополнительные поля, хранящие название атрибута и его значение. Данный подход также позволит обеспечить большую гибкость данных.

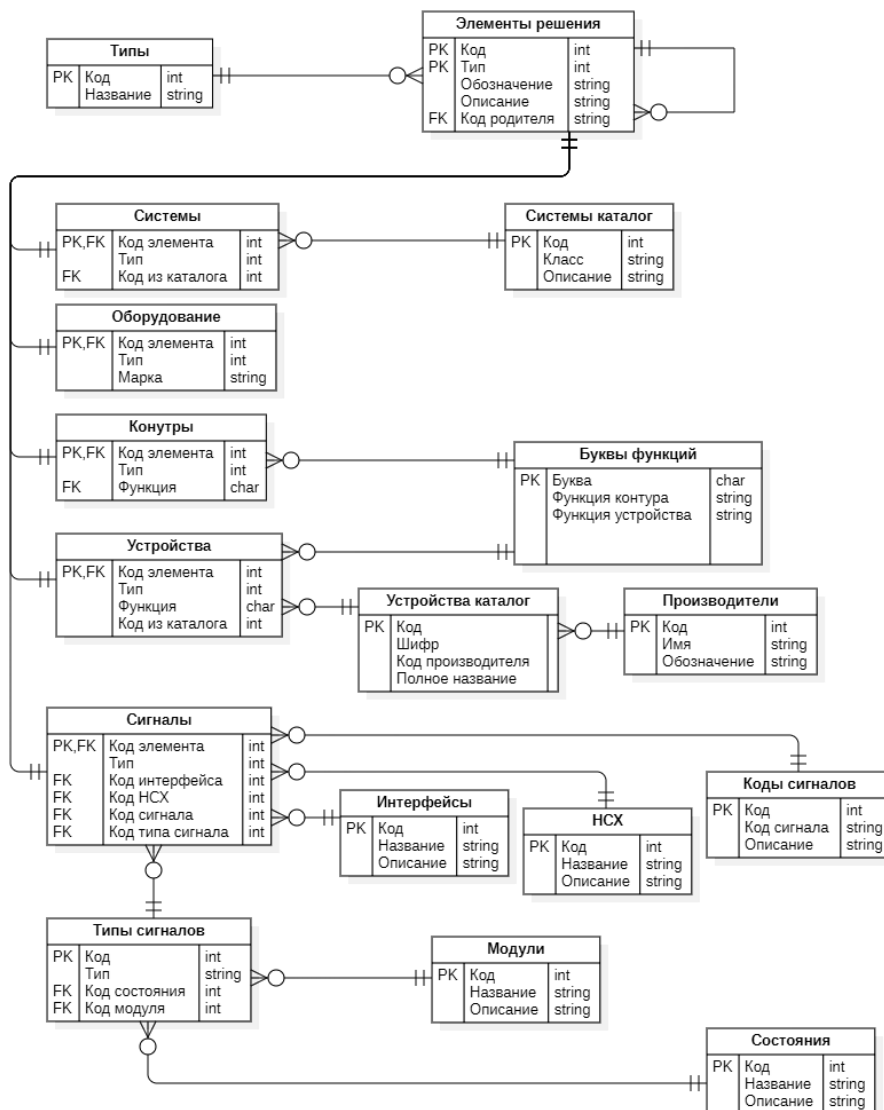


Рис. 2. Демонстрация шаблона «Супер-тип и подтипы»

Физическое расположение информационной системы в реальных условиях представляет собой классическую клиент-серверную архитектуру (см. рис. 3). Для удобства отображения на диаграмме, информационная система обозначена аббревиатурой «ISPM».

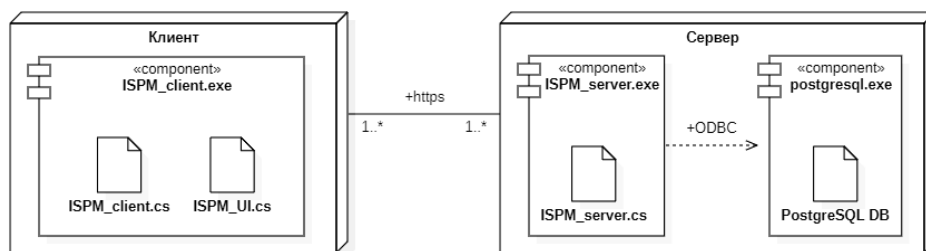


Рис. 3. Диаграмма развертывания

Для разработки прототипа пользовательского интерфейса используется фреймворк .Net Windows Forms [3]. Интерфейс пользователя должен быть понятным и доступным для уменьшения затрат на обучение работе в информационной системе.

После авторизации, пользователь попадает на главное окно – окно работы с проектом (см. рис. 4). В данном окне сверху располагается панель инструментов. На данной панели находятся кнопки для смены пользователя, открытия меню проектов, формирования отчета по текущему проекту, кнопка перехода в окно типовых решений и меню настроек. В левой части окна расположен навигатор, в котором в виде дерева отображается все содержимое проекта. В правой части находится информация о текущем открытом элементе проекта. Сначала идет базовая информация об элементе (полное имя, ссылка на родительский элемент), затем идут поля, характерные для типа текущего элемента. Кроме этого, у всех элементов присутствует ссылка на типовое решение.

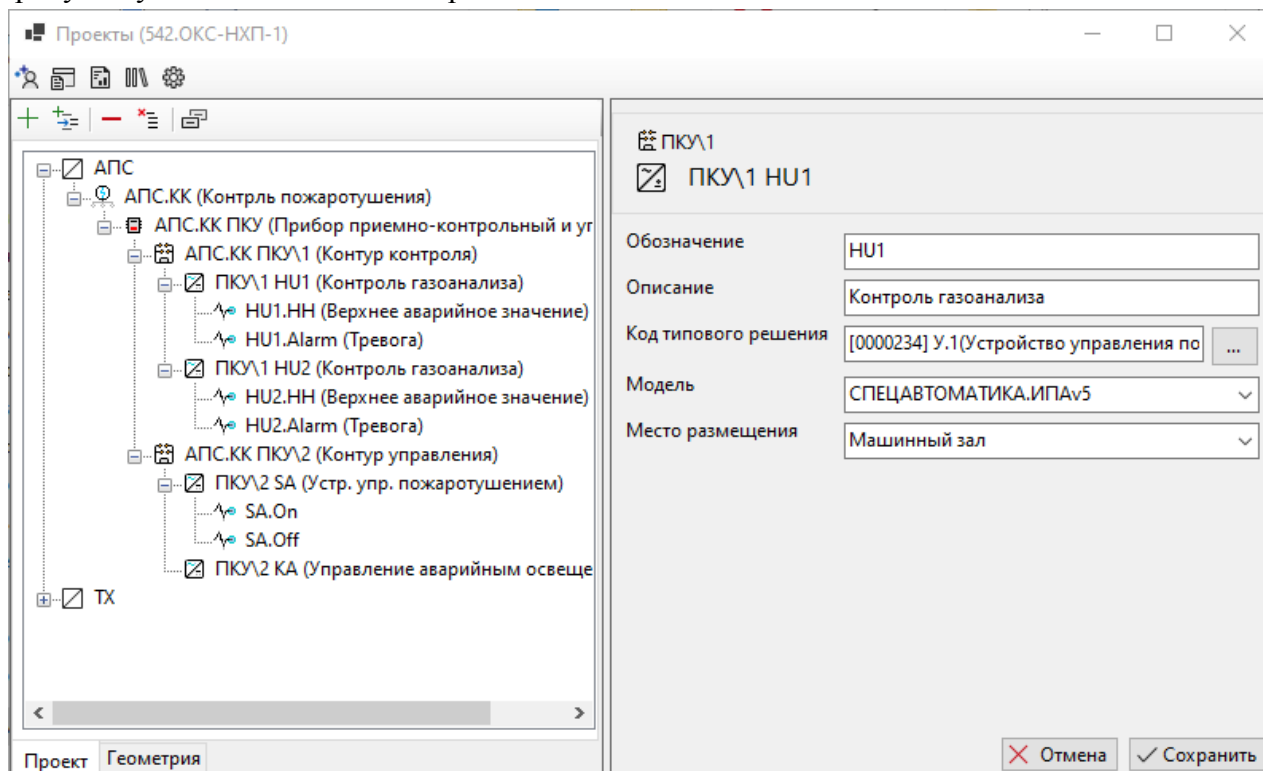


Рис. 4. Окно работы с проектом

При нажатии на кнопку «Типовые решения», осуществляется переход к окну типовых решений (см. рис. 5). Структура данного окна практически такая же, как и у окна проекта. В данном окне пользователь осуществляет работу с типовыми решениями – элементами, кото-

рые могут быть многократно переиспользованы в рамках как одного проекта, так и многих других. В нижней части панели с информацией о текущем объекте расположены дополнительные поля, которые задаются пользователем.

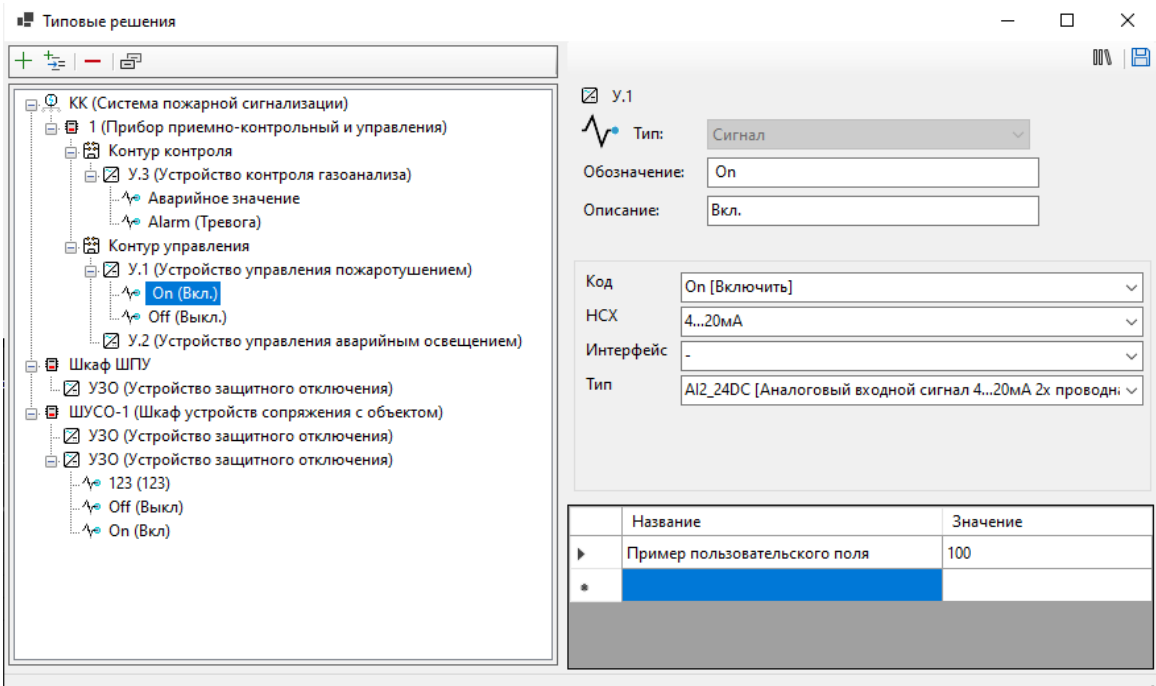


Рис. 5. Окно типовых решений

В результате работы в системе получается отчет по проекту. После нажатия на кнопку формируется отчет, который содержит информацию о каждом элементе. Ниже представлен пример формируемого отчета (см. рис. 6).

	A	B	C	D	E	F	G
1	Комплект	АПС	АПС	АПС	АПС	АПС	АПС
2	Система. Обозначение	АПС.КК	АПС.КК	АПС.КК	АПС.КК	АПС.КК	АПС.КК
3	Система. Класс	WP	WP	WP	WP	WP	WP
4	Система. Описание	Контроль пожаротушения	Контроль пожаротушения	Контроль пожаротушения	Контроль пожаротушения	Контроль пожаротушения	Контроль пожаротушения
5	Система. Место	Z1. Корпус 1	Z1. Корпус 1	Z1. Корпус 1	Z1. Корпус 1	Z1. Корпус 1	Z1. Корпус 1
6	Оборудование. Обозначение	АПС.КК ПКУ	АПС.КК ПКУ	АПС.КК ПКУ	АПС.КК ПКУ	АПС.КК ПКУ	АПС.КК ПКУ
7	Оборудование. Описание	Прибор приемно-контрольный и управления	Прибор приемно-контрольный и управления	Прибор приемно-контрольный и управления	Прибор приемно-контрольный и управления	Прибор приемно-контрольный и управления	Прибор приемно-контрольный и управления
8	Контур. Обозначение	АПС.КК ПКУ\1	АПС.КК ПКУ\1	АПС.КК ПКУ\1	АПС.КК ПКУ\1	АПС.КК ПКУ\2	АПС.КК ПКУ\2
9	Контур. Описание	Контур контроля	Контур контроля	Контур контроля	Контур контроля	Контур управления	Контур управления
10	Контур. Функция	Н	Н	Н	Н	А	А
11	Устройство. Обозначение	ПКУ\1 НУ1	ПКУ\1 НУ1	ПКУ\1 НУ2	ПКУ\1 НУ2	ПКУ\2 SA	ПКУ\2 SA
12	Устройство. Описание	Контроль газоанализа	Контроль газоанализа	Контроль газоанализа	Контроль газоанализа	Устр. Упр. Пожаротушением	Устр. Упр. Пожаротушением
13	Устройство. Функция	НУ	НУ	НУ	НУ	SA	SA
14	Устройство. Продукт	СПЕЦАВТ.ИПАв5	СПЕЦАВТ.ИПАв5	СПЕЦАВТ.ИПАв5	СПЕЦАВТ.ИПАв5		
15	Устройство. Место	Машинный зал	Машинный зал	Машинный зал	Машинный зал	Машинный зал	Машинный зал
16	Сигнал. Обозначение	НУ1.НН	НУ1.Alarm	НУ1.НН	НУ1.Alarm	SA.On	SA.Off
17	Сигнал. Описание	Верхнее аварийное	Сигнал тревоги	Верхнее аварийное	Сигнал тревоги	Запуск системы пожаротушения	Отключение системы
18	Сигнал. Код	НН	ALARM	НН	ALARM	ON	OFF
19	Сигнал. Тип	AI2_24DC	AI2_24DC	AI2_24DC	AI2_24DC	4...20mA	4...20mA
20	Сигнал. Интерфейс					Modbus	Modbus

Рис. 6. Пример формируемого отчета

Отчет представляет таблицу, в которой находятся все сводные данные по текущему проекту. Функция создания отчета в результате выполнения предоставляет файл, например, в формате Excel, который используется для дальнейшей работы в САПР.

Таким образом, разработанная и описываемая в работе информационная система для учета проектов организации позволяет повысить скорость проектирования, а также минимизировать количество рутинной работы и, как следствие, количество ошибок, связанных с человеческим фактором.

### **Библиографический список**

1. StarUML Documentation // StarUML [Электронный ресурс]. – URL: <https://docs.staruml.io> (дата обращения: 23.04.2023).
2. *Крэг Ларман*. Применение UML 2.0 и шаблонов проектирования. – 3-е изд. – М.: Вильямс, 2006. – 736 с.
3. Windows Forms documentation // Microsoft Learn [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/> (дата обращения: 30.04.2023).

### **References**

1. StarUML Documentation. StarUML. [Electronic resource]. – URL: <https://docs.staruml.io>.
2. Craig Larman. Applying UML 2.0 and patterns. 3rd edition. Williams Publishing. 2006. 736 p.
3. Windows Forms documentation. Microsoft Learn. [Electronic resource]. – URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/>.

## **DESIGN AND DEVELOPMENT OF INFORMATION SYSTEM FOR ORGANIZATION PROJECTS MANAGEMENT**

*Zolnikov Vitaliy V.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [vvvzolnikov@gmail.com](mailto:vvvzolnikov@gmail.com)

*Kuzaev Aidar F.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [kuzaev@pspu.ru](mailto:kuzaev@pspu.ru)

The design and development of an information system for project management within an organization are considered. The main functional requirements are identified based on the analysis of the subject area. A well-founded selection of design and development tools for the application is made. The design is conducted using the UML notation, and the StarUML CASE tool for object-oriented analysis and modeling is chosen. As a result of the system design, the behavior of the information system, its interaction with users, and the documentation of key processes are described. A user interface prototype is developed based on the defined processes and functional requirements.

Keywords: project, information system, design, project management, development.

## **ОПЫТ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ВЕТЕРИНАРНОЙ КЛИНИКИ**

*Колодкина Марья Владимировна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, normalsuperman@yandex.ru

*Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, averin-s-i@ya.ru

Рассматриваются основные средства для проектирования и документирования информационной системы для ветеринарной клиники. Определён предмет и объект исследования, сформулирована актуальность разработки данной информационной системы, а также поставлены задачи, которые необходимо решить в ходе проектирования. В ходе исследования проведен анализ существующих информационных систем, которые выполняют схожие функции. Определены функциональные возможности информационной системы. Произведен выбор средств проектирования и разработки. В качестве средств проектирования рассматриваются инструменты для UML-моделирования [1-3]. В результате анализа средств проектирования было выбрано инструментальное программное обеспечение проектирования информационной системы CASE-средство StarUML для проектирования системы, и было выбрано приложение Figma для разработки прототипа интерфейса приложения. В результате проектирования описываемой информационной системы было описано поведение системы и взаимодействие с ней пользователей. На основе построенных диаграмм был спроектирован интерфейс информационной системы и разработана информационная система.

Ключевые слова: ветеринарная клиника, ветеринария, домашние животные, проектирование, документирование.

В эпоху информационных технологий человек все больше и больше зависит от различных электронных ресурсов, эти технологии прочно вошли во многие сферы деятельности человека. Одной из таких областей деятельности стала ветеринария. Сейчас домашние животные заняли обширную нишу, как в жизни простого человека, так и в экономических сферах многих государств. Так, например, в России более половины населения являются владельцами домашних питомцев. На текущий момент существует множество учреждений, удовлетворяющих запросы потребителя, и почти каждое из них имеет свой сайт с перечнем предоставляемых услуг и товаров.

В ситуации серьезной конкуренции, существующей на рынке, наличие электронного ресурса, привлекающего клиентов и упрощающего работу клиники, является серьезным стратегическим преимуществом, что обосновывает актуальность данной темы.

Объектом исследования в данной работе являются информационные системы, предназначенные для реализации ветеринарных товаров и услуг, а также предоставляющие онлайн консультации со специалистами.

Целью работы было проектирование, документирование и создание информационной системы для ветеринарной клиники.

Для выполнения цели выделены следующие задачи:

- Анализ и обзор существующих систем для ветеринарных клиник;
- Определение основных характеристик проектируемой системы;
- Выбор методологии моделирования;
- Выбор технологии разработки для проектирования информационной системы;
- Разработка интерфейса информационной системы;
- Разработка информационной системы;
- Документирование информационной системы.

Логика работы системы построена в StarUML [2] с помощью диаграмм прецедентов и представлена ниже (см. рис. 1–3.).



Рис. 1. Диаграмма прецедентов «Ветеринарная клиника – Клиент»

Прецеденты «Авторизация» и «Регистрация» отвечают за авторизацию и регистрацию клиента в системе. «Заполнение анкеты животного» обозначает возможность пользователя заполнить данные о своем питомце. Прецедент «Задать вопрос в поддержку» позволяет клиенту решить проблемы и задать вопросы, возникшие в ходе использования информационной системы. «Покупка товара» – прецедент, отвечающий за покупку клиентом товаров, реализуемых системой. Прецедент «Запись к специалисту» позволяет клиенту записаться на прием к специалисту. При детальном рассмотрении данной диаграммы, можно заметить, что прецедент «Запись к специалисту» включает в себя прецедент «Выбор формата встречи», который в свою очередь содержит прецедент «Выбор даты и времени», данные прецеденты дают возможность выбора даты, времени и формата приема. «Просмотр истории чатов» – прецедент позволяющий клиенту просматривать истории диалогов со специалистами. Прецедент «Общение со специалистом online» отвечает за прием клиента специалистом, а прецедент «Просмотр рекомендаций врача» дает пользователю возможность просматривать назначения и рекомендации по лечению питомца, выданные специалистами (см. рис. 1.).



Рис. 2. Диаграмма прецедентов «Ветеринарная клиника – Специалист»

Прецедент «Авторизация» отвечает за авторизацию пользователя в системе. «Заполнение анкеты» дает пользователю ввести информацию о себе, как о специалисте. «Дать рекомендации и назначения» – прецедент, который позволяет специалисту дать клиенту рекомендацию и назначения по лечению животного. Прецедент «Задать вопрос в поддержку» позволяет специалисту решить проблемы и задать вопросы, возникшие в ходе использования информационной системы. «Прием клиентов» отвечает за прием клиентов специалистом. (см. рис. 2.)



Рис. 3. Диаграмма прецедентов «Ветеринарная клиника – Администратор»



Прецедент «Просмотр списка товаров» дает пользователю возможность просматривать список товаров, реализуемых в системе. Прецедент «Авторизация» отвечает за авторизацию пользователя в системе. «Добавление нового товара» отвечает за добавление нового товара в систему для его дальнейшей реализации в системе. Прецедент «Редактирование товара» отредактировать информацию о существующем товаре в системе. «Ответ на вопросы клиентов и специалистов» – прецедент, отвечающий за решение проблем и ответы на вопросы клиентов и специалистов. Прецедент «Просмотр списка специалистов» дает пользователю возможность просматривать список специалистов, которые используют систему для оказания ветеринарной помощи. «Добавление нового специалиста» отвечает за добавление нового специалиста в систему, прецедент «Редактирование специалиста» позволяет администратору отредактировать информацию о существующем в системе специалисте (см. рис. 3.).

Проектирование базы данных – один из наиболее важных процессов при создании информационной системы, он включает в себя дизайн таблиц, связей между ними и определение атрибутов данных, которые хранятся в таблицах. Спроектированные макеты базы данных в совокупности представляют единый макет и представлены ниже (см. рис. 4–6).

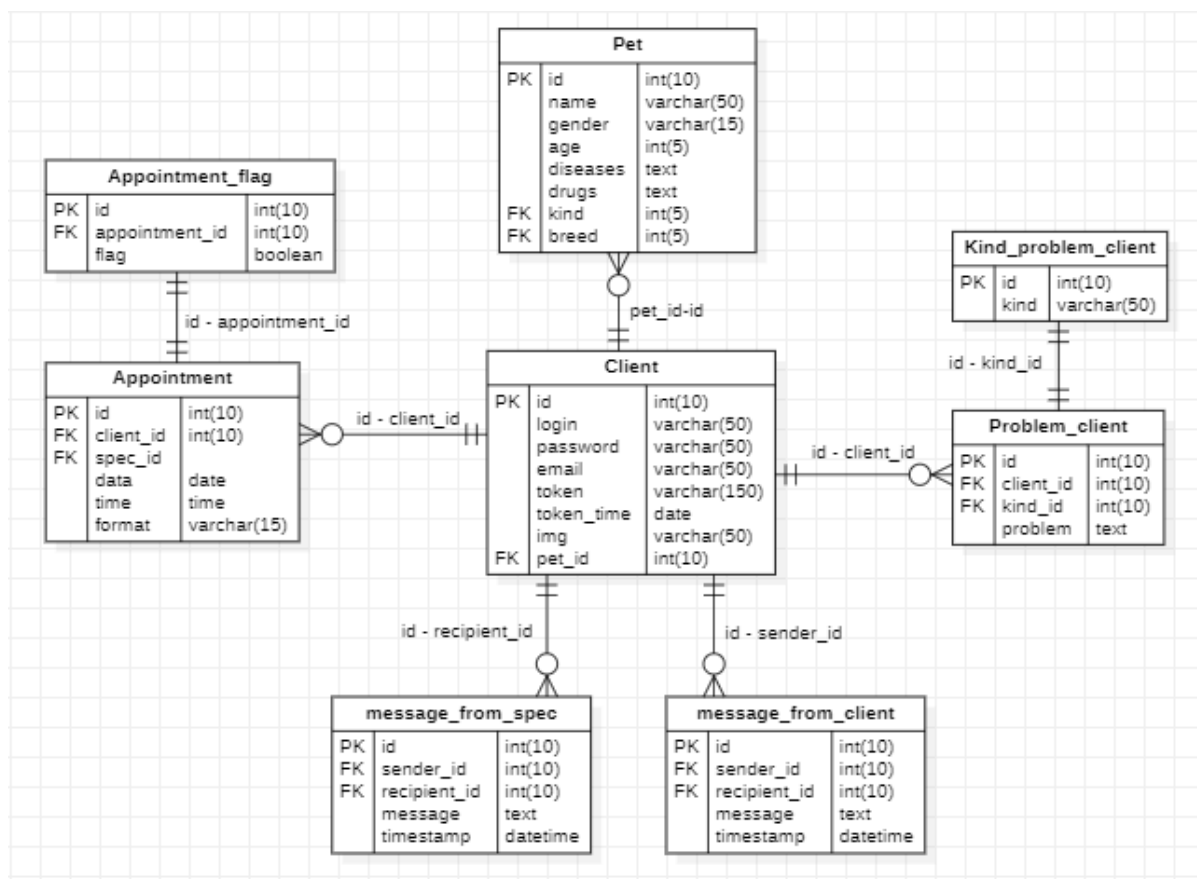


Рис. 4. Макет для хранения данных клиента

«Макет для хранения данных клиента» представляет таблицы, предназначенные для хранения данных клиента. Таблица «Client» содержит данные, связанные непосредственно с клиентом, она связана со следующими таблицами: «Appointment», «Pet», «Problem\_client», «message\_from\_spec» и «message\_from\_client». Таблица «Appointment» хранит данные, связанные с записью клиента на прием к специалисту. Таблица «Pet» содержит информацию о домашнем питомце клиента. Таблица «Problem\_client» хранит данные о проблемах клиента, которые связаны с информационной системой, специалистом или записью на прием. Табли-

цы «message\_from\_client» и «message\_from\_spec» хранят информацию о сообщениях от клиентов и специалистов соответственно. Также стоит заметить, что таблица «Appointment» связана с таблицей «Appointment\_flag», которая служит для определения доступности даты для записи на прием. Таблица «Problem\_client» связана с таблицей «Kind\_problem\_client», которая служит справочником для определения типа проблемы клиента (см. рис. 4.).

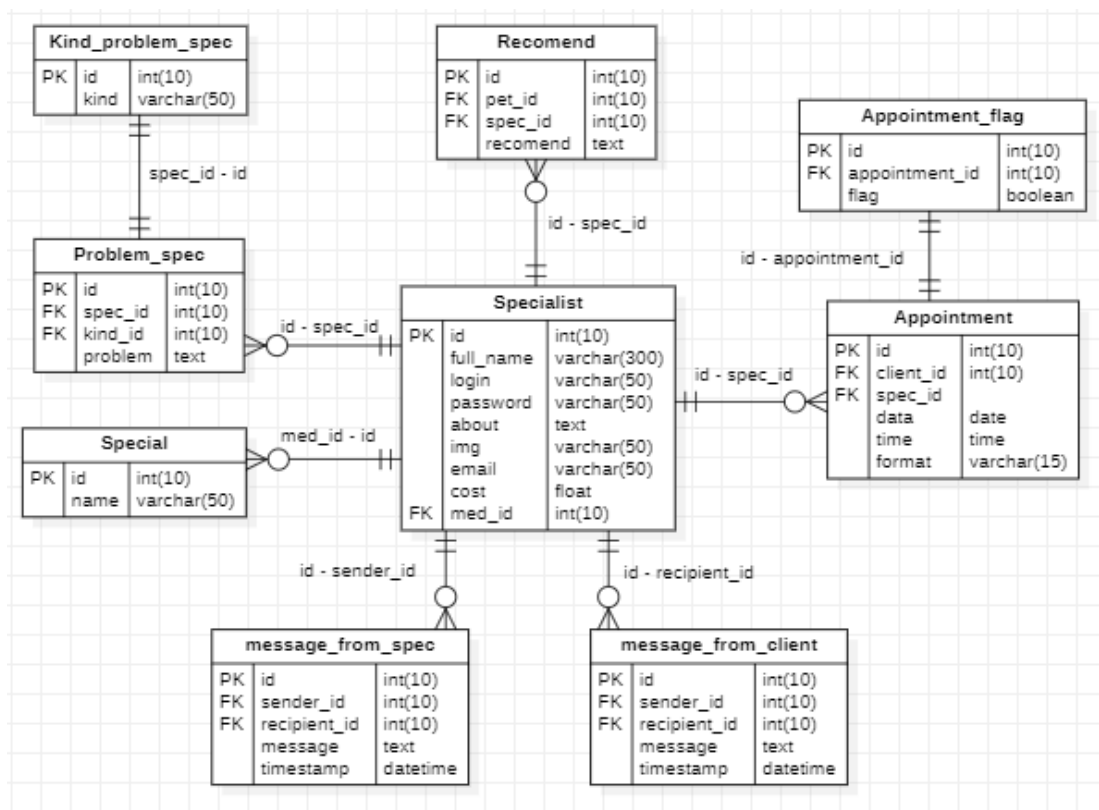


Рис. 5. Макет для хранения данных специалиста

«Макет для хранения данных специалиста» иллюстрирует набор таблиц, связанных со специалистом. «Specialist» – является таблицей для хранения данных, связанных со специалистом, использующим систему для оказания ветеринарной помощи, она связана со следующими таблицами: «Special», «Problem\_spec», «Recomend», «Appointment», «message\_from\_spec» и «message\_from\_client». Таблица «Special» является справочником и служит для хранения области работы специалиста. Таблица «Problem\_spec» применяется для хранения информации о проблемах специалиста. Таблица «Recomend» используется для хранения рекомендаций по лечению животного, выданных специалистом. Таблица «Appointment» хранит данные, связанные с записью на прием к специалисту. Таблицы «message\_from\_spec» и «message\_from\_client» хранят сообщения от специалистов и клиентов соответственно. Стоит отметить, что таблица «Appointment» связана с таблицей «Appointment\_flag», которая служит для определения доступности даты для записи на прием. Таблица «Problem\_spec» связана с таблицей «Kind\_problem\_spec», которая служит справочником для определения типа проблемы специалиста (см. рис. 5.).

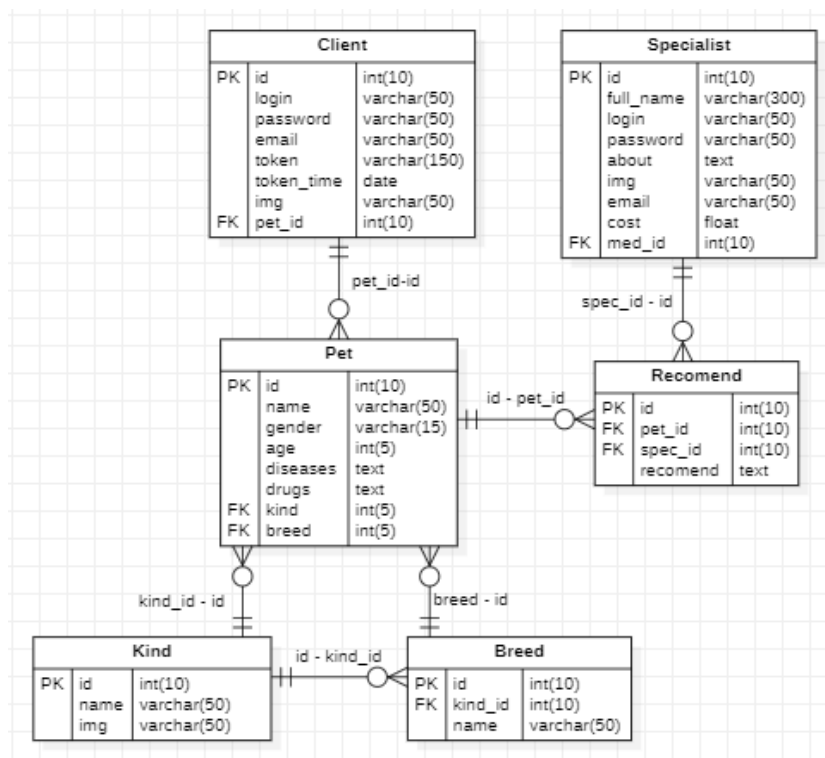


Рис. 6. Макет для хранения о питомце

«Макет для хранения данных о питомце» отображает таблицы, связанные с животным клиента. Таблица «Pet» хранит информацию о домашнем питомце клиента, она связана со следующими таблицами: «Client», «Breed», «Kind» и «Recomend». Таблица «Client» используется для хранения данных о владельце питомца. Таблицы «Kind» и «Breed» связаны друг с другом и хранят информацию о виде и породе животного соответственно. Следует обратить внимание на то, что таблица «Recomend» связана с таблицей «Specialist» и применяется для хранения рекомендаций по лечению животного, выданных специалистом (см. рис. 6.).

Информационные системы, вне зависимости от среды, где они используются должны иметь интуитивно понятный пользовательский интерфейс (см. рис. 7–10.).


 <p>test1</p> <p>Выйти из аккаунта</p>	Добавить информацию о животном				
	Информация о животном				
	Имя	Вид	Порода	Пол	Возраст (лет)
	Мелкая	Кот	Американский керл	Женский	4
<p>Рекомендации врача</p>		<p>Сменить почту</p>			
<p>Николай Алексеевич Герасимов был на приеме с Вами 20 мая 2023 в 08:00 Online</p> <p>Посмотреть историю</p>					
<p>Николай Алексеевич Герасимов ждет Вас на приеме 27 мая 2023 в 09:00 Online</p> <p>Начать диалог</p>					

Рис. 7. Личный кабинет – Клиент

«Личный кабинет – Клиент» отображает прототип интерфейса, отвечающего за личный кабинет клиента ветеринарной клиники. По нажатию на изображение пользователь может добавить фотографию к своему личному кабинету. Под фото находится логин пользователя и кнопка «Выйти из аккаунта», которая дает пользователю выйти из системы. Надпись «Доба-

вить информацию о животном» является ссылкой, отправляющей пользователя на страницу заполнения анкеты о питомце, ниже отображается информация пользователя из данной анкеты. Далее находятся две кнопки: «Рекомендации врача» и «Сменить почту», которые соответственно позволяют пользователю посмотреть рекомендации специалиста по лечению питомца и сменить почту. Ниже находится информация о записях клиента на прием к специалисту. С помощью кнопки «Посмотреть историю» пользователь может посмотреть историю диалога со специалистом. Кнопка «Начать диалог» позволяет пользователю открыть диалог со специалистом, данная кнопка становится активной по наступлению даты и времени приема (см. рис. 7.).

Запись к специалисту

Май 2023

Николай Алексеевич Герасимов



Терапевт

Высококвалифицированный специалист, врач общей практики, который осуществляет первичную диагностику и лечение заболеваний внутренних органов. Стаж работы - 7 лет

1800 руб.

Следующий месяц

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Рис. 8. Запись к специалисту

«Запись к специалисту» является прототипом интерфейса пользователя для записи на прием к специалисту и содержит информацию о выбранном специалисте и календарь его работ. Серым цветом в календаре отображаются прошедшие даты, на которые клиент не может записаться, зеленым цветом окрашены даты свободные для записи, красным – даты, на которые нельзя записаться по причине отсутствия свободного времени для записи. Кнопка «Следующий месяц» позволяет отобразить следующий месяц, при отображении месяца отличного от настоящего слева от кнопки «Следующий месяц» отображается кнопка «Предыдущий месяц» (см. рис. 8.).

23 мая 2023

Выберете время приема

☐ 08:00-08:30

☐ 08:30-09:00

☐ 09:00-09:30

☐ 09:30-10:00

☐ 10:00-10:30

☐ 10:30-11:00

☐ 11:00-11:30

☐ 11:30-12:00

☐ 12:00-12:30

☐ 12:30-13:00

☐ 14:00-14:30

☐ 14:30-15:00

☐ 15:00-15:30

☐ 15:30-16:00

☐ 16:00-16:30

☐ 16:30-17:00

☐ 17:00-17:30

☐ 17:30-18:00

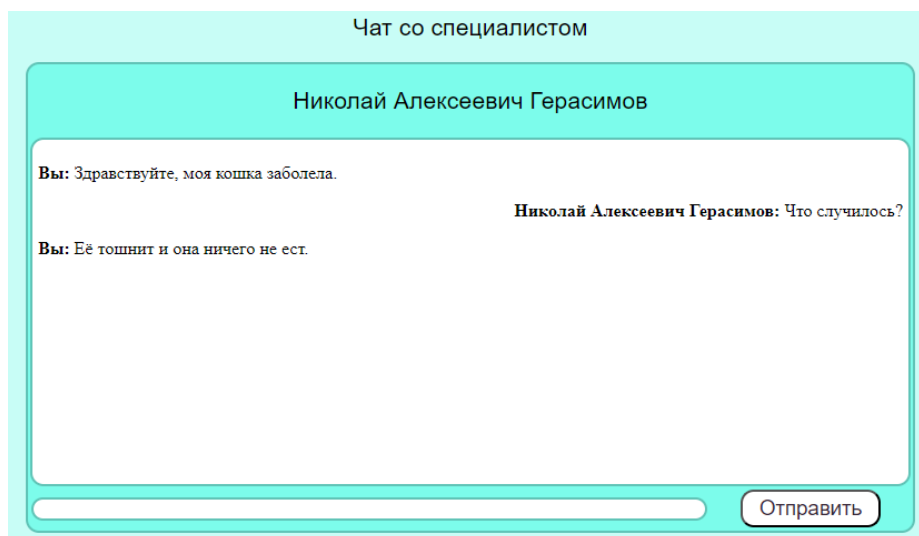
Выберете время приема

В клинике ☐ Online ☐

Записаться

Рис. 9. Выбор времени и формата записи к специалисту

«Выбор времени и формата записи к специалисту» иллюстрирует интерфейс клиента, отвечающий за выбор времени и формата приема. Сверху отображается выбранная клиентом дата, ниже время приема. Время приема, окрашенное в красный цвет, является недоступным для записи, так как занято другим клиентом, время, окрашенное зеленым цветом, является доступным для записи на прием. Ниже пользователю предоставляется выбор формата приема «В клинике» или «Online». Далее находится кнопка «Записаться», которая подтверждает запись на прием и перевод пользователя на страницу оплаты (см. рис. 9).



**Рис. 10. Диалог со специалистом**

«Диалог со специалистом» отображает прототип интерфейса клиента, отвечающего за обмен сообщений между клиентом и специалистом. Данный прототип содержит фамилию, имя и отчество специалиста, ниже находится поле, где отображаются сообщения клиента и специалиста, далее находится поле для ввода сообщения и кнопка «Отправить», которая отвечает за отправку сообщения специалисту (см. рис. 10.).

В результате проделанной работы была разработана информационная система ветеринарной клиники. Она позволяет покупать товары для животных, проходить онлайн консультации со специалистами и получать от них рекомендации по лечению питомца, а также учитывает опыт существующих систем, объединяя в себе их наиболее востребованные и значимые черты.

### **Библиографический список**

1. Введение в UML: Информация // ИНТУИТ [Электронный ресурс]. – URL: <https://intuit.ru/studies/courses/1007/229/info> (дата обращения: 15.05.2023).
2. Руководство пользователя UserGuide // StarUML: [Электронный ресурс]. – URL: <https://usermanual.wiki/Document/userguide.1128996068/html> (дата обращения: 16.05.2023).
3. *Шаврин С.М., Лядова Л.Н., Чуприна С.И.* Моделирование и проектирование информационных систем: учеб.-метод. пособие // Перм. гос. ун-т. – 2007. – с.152.

### **References**

1. Introduction to UML [Electronic resource]. – URL: <https://intuit.ru/studies/courses/1007/229/info>.
2. User Guide for use StarUML [Electronic resource]. – URL: <https://usermanual.wiki/Document/userguide.1128996068/html>.
3. *Shavrin S., Lyandova L., Chuprina S.* Modeling and design of information systems. Perm. 2007. 152

## **DESIGN, DOCUMENTING, AND DEVELOPMENT OF AN INFORMATION SYSTEM FOR A VETERINARY CLINIC**

*Kolodkina Marya V.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
normalsuperman@yandex.ru

*Averin Sergey I.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia, averin-s-i@ya.ru

The basic tools for designing and documenting an information system for a veterinary clinic are considered. The subject and object of research are defined, the relevance of the development of this information system is formulated, and tasks that need to be solved during the design are set. In the course of the study, an analysis of existing information systems that perform similar functions was carried out. The functionality of the information system is defined. The choice of design and development tools has been made. As design tools are considered tools for UML modeling [1]. As a result of the analysis of the design tools, the CASE information system design software was selected - the StarUML tool for system design, and the Figma application was selected to develop a prototype of the application interface [2, 3]. As a result of the design of the described information system, the behavior of the system and the interaction of users with it were described. Based on the constructed diagrams, the interface of the information system was designed and an information system was developed.

**Keywords:** veterinary clinic, veterinary medicine, pets, design, documenting.

## ПРОЕКТИРОВАНИЕ ОНЛАЙН-СЕРВИСА С API ДЛЯ ЗАПОЛНЕНИЯ ДОКУМЕНТОВ ПО ШАБЛОНАМ

*Колпациков Максим Николаевич, Кнутова Наталия Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, mxmbclck@gmail.com

Рассматриваются информационные системы, предназначенные для автоматизированного заполнения документов по шаблонам. Описываются общие принципы их функционирования, вводятся понятия формы документа, содержательных данных, процедуры слияния. Выявляется связь систем заполнения шаблонов с системами управления документооборотом. Приводится анализ существующих решений в области систем заполнения шаблонов. На основании проведённого анализа построена модель информационной системы, учитывающая достоинства и недостатки существующих решений и предлагающая новое решение универсального характера. Построенная модель включает в себя описание вариантов использования с точек зрения конечного пользователя и сторонней информационной системы, описание деятельности всех компонентов системы, схему реляционной базы данных и макет пользовательского веб-интерфейса.

Ключевые слова: заполнение документов по шаблону, документооборот, онлайн-сервис, API.

### **Введение**

Довольно часто возникает необходимость создания официальных документов. Видов документов существует очень много, например, заявления, договоры, акты, отчёты и прочее. Большинство этих документов имеют чётко определённую, или даже стандартизованную, структуру, поэтому зачастую нет нужды формировать их с нуля. Достаточно один раз создать шаблон документа, а затем лишь наполнять его необходимыми данными. Хотя это можно сделать средствами текстового процессора, для этих целей существует специализированное программное обеспечение (ПО).

Существует две точки зрения, обуславливающие актуальность такого ПО. Для индивидуальных пользователей ключевым фактором является уровень их квалификации в конкретной предметной области, которого может быть недостаточно для того, чтобы самостоятельно правильно оформить соответствующий документ. Иначе говоря, необходим надёжный способ получения готовых документов, при котором от пользователя требуются только содержательные данные.

Если же речь идёт о предприятиях, то на первый план выходит повышение производительности при решении больших задач. При этом часто возникает необходимость составления большого количества вспомогательных документов: согласий на обработку персональных данных, разнородных анкет, протоколов и др. Кроме того, задача сама по себе может быть связана с подготовкой большого объёма документации. В таких случаях производительность труда может быть резко увеличена за счёт использования программных средств, способных автоматически формировать комплекты документов на основе данных в табличной форме. Как показывает практика, данный подход имеет значительную эффективность [1].

Как показывает анализ существующих решений в этой области, качественных продуктов, в полной мере удовлетворяющих потребности как индивидов, так и бизнеса, крайне мало, что говорит о недостаточном внимании к данному виду программного обеспечения.

На основании вышеизложенного, создание гибких систем автоматизированного формирования документов на основе шаблонов является актуальной задачей на момент написания данной работы.

Объектом настоящего исследования является программное обеспечение для ЭВМ. Предмет исследования представляет собой прикладное ПО, предназначенное для автоматизированного создания документов по шаблонам.

Целью данной работы является создание модели информационной системы, предназначенной для формирования документов по шаблонам (или иначе – системы заполнения шаблонов).

Для достижения этой цели необходимо выполнение следующих задач:

- анализ общих свойств систем заполнения шаблонов;
- обзор и анализ существующих решений в этой области;
- выбор методологии проектирования;
- собственно проектирование указанной системы в рамках выбранной методологии.

Следует заметить, данная работа направлена на создание концептуального проекта, учитывающего преимущества и недостатки существующих решений, как основы для эталонной реализации рассматриваемой системы. Данный проект не будет привязан к какой-либо конкретной реализации и не будет представлять собой доработку какого-либо конкретного решения.

### **Общий анализ программ заполнения шаблонов**

Концептуально работа программы заполнения шаблонов сводится к тому, чтобы на основании формы и содержательных данных сформировать результирующий документ. Форма представляет собой артефакт, содержащий однородную информацию о результирующем документе. Строго говоря, любую форму в конечном итоге можно представить как текст – последовательность символов. Природа текста может быть разной: текст на естественном языке, текст на языке разметки (XML, HTML, Markdown и др.), текст на языке программирования, и т. д.

Форма включает в себя постоянную и переменную части. Постоянная часть отражает общее содержание результирующего документа и определяется его спецификой и назначением. Переменная часть служит для конкретизации содержания документа применительно к данным ситуации или объекту действительности.

Если детализировать эту модель, то содержательные данные можно представлять как гетерогенную информацию, включающую в себя конкретные значения различных атрибутов описываемого объекта. Переменную часть формы тогда естественно представлять как набор ячеек (или полей), предназначенных для хранения этих значений.

Процедура формирования документа, следовательно, сводится к последовательному сопоставлению и связыванию набора значений с набором полей формы (слиянию). Поскольку данные имеют гетерогенную природу, этапу связывания может предшествовать этап преобразования, в ходе которого осуществляется приведение данных к однородному виду.

Анализ существующих на данный момент решений позволяет разделить их на два типа: самостоятельные информационные системы и модули других информационных систем.



В первом случае программа заполнения шаблонов является автономным приложением. Чаще всего это веб-приложения, ориентированные на индивидуальных пользователей. Источником содержательных данных в таких приложениях всегда является конечный пользователь, а источником шаблонов (форм) может быть как пользователь, так и встроенная база данных.

Во втором случае программа заполнения шаблонов рассматривается как функциональный модуль системы управления документооборотом, которая в свою очередь может быть частью системы управления предприятием [2]. При таком подходе организуется программный интерфейс (API), через который осуществляется обмен данными между программой заполнения и внешней средой. Источником как содержательных данных, так и форм здесь является встроенная база данных объемлющей системы. Она же осуществляет дальнейшее управление результирующим документом.

### **Результаты анализа существующих решений**

Было выявлено, что наиболее удачными решениями для индивидуальных пользователей являются текстовые процессоры (Microsoft Word и LibreOffice Writer), а также сервисы, предлагающие встроенную коллекцию шаблонов (например, R6R [3] и Eforms [4]). Достоинствами данных решений является простота использования и умеренная гибкость настройки. Среди главных недостатков следует выделить отсутствие поддержки пользовательских шаблонов у систем уровня R6R и Eforms.

Среди достоинств систем корпоративного уровня (например, DocWebService [5] и 1С: Документооборот) высокий уровень гибкости, поддержка пакетного режима работы, широкие возможности по автоматизации и администрированию. Недостатком этих решений можно считать необходимость дополнительных трудозатрат на подготовку шаблона, поскольку готовых решений не предоставляется.

Программируемые системы (например, Aspose Words Assembly [6] и DocAssembly [7]) могут представлять интерес для разработчиков, а также для бизнеса, обладающего достаточными ресурсами для создания персонализированных программных решений. Главным достоинством таких систем является высочайший уровень гибкости, а также поддержка ряда отличительных особенностей:

- получение и внедрение в документы рукописных подписей;
- оптическое распознавание текста в загружаемых файлах;
- поддержка облачных хранилищ для извлечения шаблона документа;
- заполнение шаблона в формате интервью с использованием технологий машинного обучения.

### **Объектное моделирование системы**

Для описания модели системы был выбран язык UML и построен ряд диаграмм.

Диаграмма прецедентов призвана отразить функциональность системы в наиболее обобщенном виде. Диаграмма прецедентов изображена на рис. 1.

В качестве действующих лиц рассматриваются конечный пользователь и сторонняя система. В первом случае взаимодействие организуется через веб-интерфейс, а во втором – через API.

В первую очередь, действующее лицо совершает выбор шаблона для заполнения. Это может быть типовый шаблон из встроенной в систему коллекции, либо не типовый, загружаемый самим лицом.

После этого осуществляется заполнение шаблона содержательными данными. Конечному пользователю предлагается два способа: заполнить все поля шаблона вручную, используя визуальную форму, либо автоматически, путём загрузки данных в табличном формате. В API предполагается реализация только второго способа, чтобы с одной стороны сократить количество операций обмена данными с сервером (см. рис. 3), а с другой – чтобы оставить структуру шаблона непрозрачной.

Наконец, система осуществляет заполнение шаблона, вследствие чего формируется комплект выходных документов, который доставляется действующему лицу.



Рис. 1. Диаграмма вариантов использования

Диаграмма деятельности позволяет отразить логику работы системы на любом уровне детализации. На рис. 2 изображена диаграмма деятельности для ситуации, когда с системой работает конечный пользователь.

Вводится разделение системы на два модуля: фронтенд и бэкенд, реализующие соответственно пользовательский интерфейс и бизнес-логику системы. Всё взаимодействие с пользователем ведётся в рамках сеанса, который открывается и закрывается модулем бизнес-логики. В задачи последнего также входит работа с базой данных, проверка и преобразование содержательных данных, а также формирование комплекта результирующих документов.

Пользовательский интерфейс обеспечивает взаимодействие пользователя с модулем бизнес-логики. В его задачи входит отображение списка шаблонов, которые в данный момент имеются во встроенной коллекции, визуальной формы и сообщений об ошибках.

В случае, когда взаимодействие с системой ведётся через API, функции интерфейса выполняет сама сторонняя система. Функции модуля бизнес-логики, таким образом, определяют предоставляемый API.

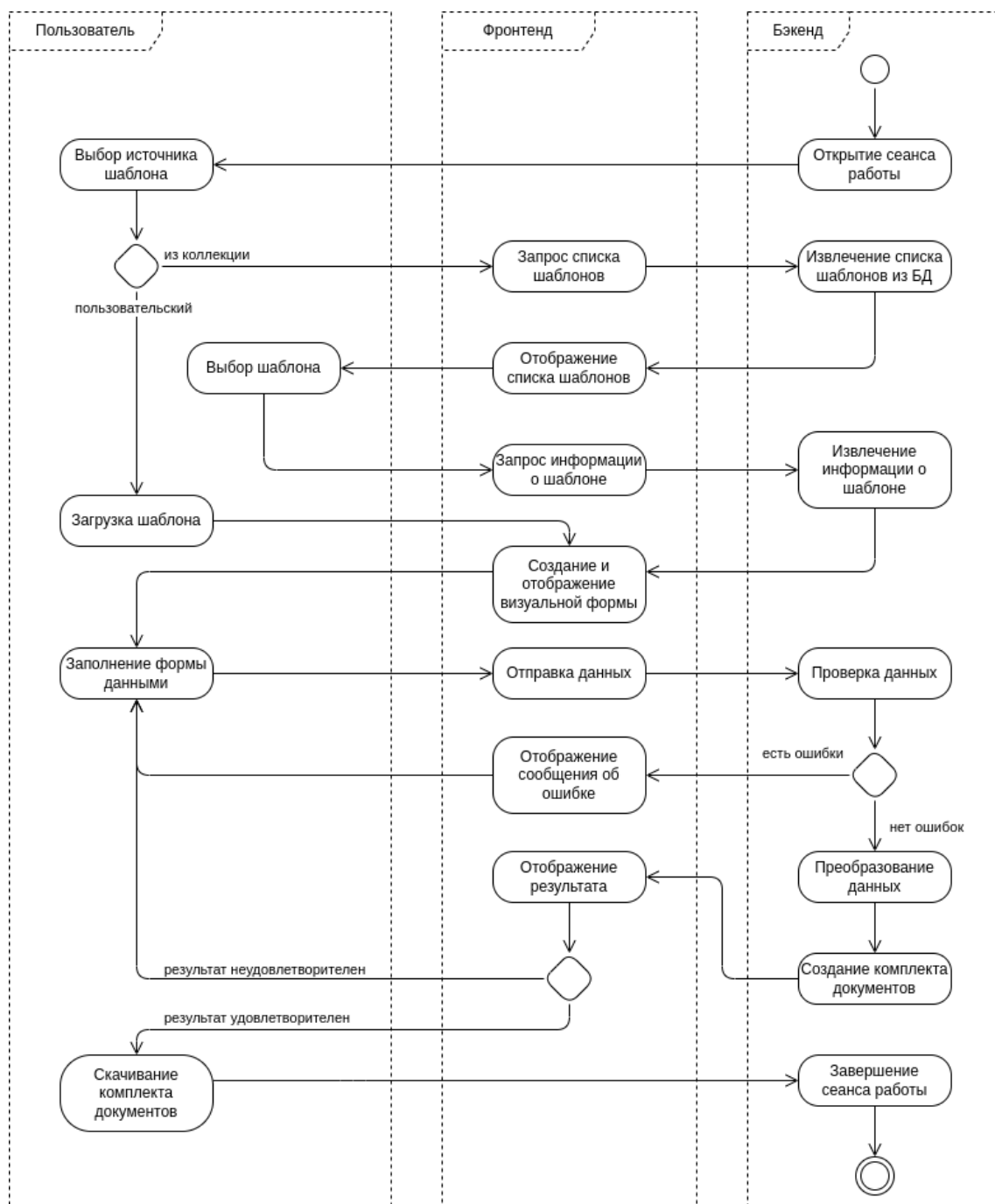


Рис. 2. Диаграмма деятельности

Диаграмма развёртывания позволяет взглянуть на физическую организацию системы. Автор также задействует диаграмму компонентов, поскольку она служит той же цели, но на другом уровне.

Принципиально в системе следует выделить три компонента: бэкенд (модуль бизнес-логики), фронтенд (модуль пользовательского интерфейса) и базу данных, как вспомогательный компонент, используемый при работе модуля бизнес-логики.

Физически система имеет клиент-серверную архитектуру. Канал связи между сервером и клиентом организуется с применением протокола SSL/TLS на уровне представления.

Для организации информационного обмена между компонентами системы предусмотрено три интерфейса:

– IEnumerate предназначен для извлечения списка шаблонов, имеющихся в коллекции. Обработка запроса, поступающего на сервер через данный интерфейс, сводится к выполнению SQL-запроса к базе данных;

– IExchange служит для обмена артефактами: файлом шаблона, содержательными данными, результирующими документами. Предполагается, что на данный интерфейс будет приходиться большая часть сетевого трафика;

– IControl – это управляющий интерфейс, который служит, в частности, для запуска процесса заполнения шаблона на сервере. Кроме того, данный интерфейс может использоваться для установки параметров сеанса работы.

Указанные интерфейсы в совокупности формируют API системы в целом.

Полностью диаграмма развёртывания представлена на рис. 3.

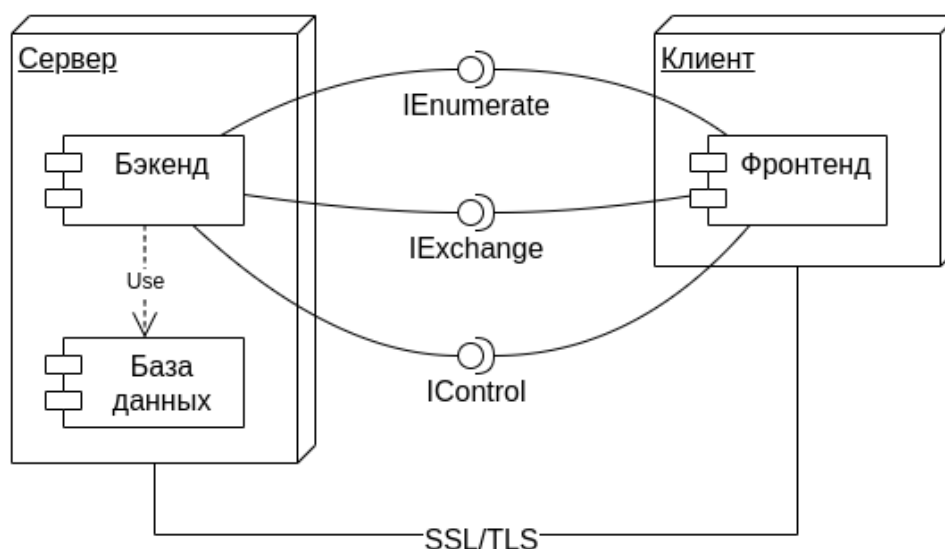


Рис. 3. Совмещённая диаграмма компонентов и развёртывания

Как уже было сказано выше, одним из компонентов проектируемой системы является реляционная база данных. Её схема представлена на рис. 4. Роль базы данных в системе заключается в хранении метаданных о шаблонах. Сами шаблоны представляют собой обычные файлы, хранящиеся на сервере.

Реляционная модель состоит из трёх главных таблиц, содержащих атрибуты самих шаблонов, имеющихся в них полей и меток. Вспомогательные таблицы служат для организации связи типа «многие ко многим». Выбор кардинальности связи Шаблон-Метка очевиден. Идея связи Шаблон-Поле заключается в том, что поля не привязаны к конкретному шаблону, а являются типовыми сущностями, содержание которых основано на практическом опыте. Например, обращения в формальных документах имеют повторяющуюся структуру, что позволяет использовать соответствующее поле в нескольких шаблонах.

Для шаблона необязательно указывается соответствие стандартам. Данный атрибут актуален для документов, форма которых закреплена в каком-либо государственном стандарте, приложении к приказу какого-либо ведомства и т. д. Данный атрибут может использоваться для организации поиска, а также может играть роль «знака качества» для пользователей.

Главным способом группировки и поиска шаблонов в базе являются метки – ключевые слова, описывающие тематику и область применения данного шаблона. Метки выделены в отдельную таблицу для ускорения поиска, а также для устранения аномалий данных.

Для шаблонов записывается дата последнего обновления связанного файла. Данный атрибут может использоваться для упорядочивания шаблонов в поисковой выдаче, а также как индикатор актуальности шаблона, если соответствующий документ часто обновляется.

Наконец, ссылка на файл в формате URL обеспечивает связь записи в базе с файлом шаблона на сервере.

Для полей задан тип данных, который используется механизмом слияния для определения способа его заполнения. Кроме того, для полей заданы наборы преобразований и ограничений. Оба атрибута представляют собой список функций программной реализации системы. В задачи функций преобразования входит трансформация входных данных в соответствии с требованиями конкретного шаблона. Функции ограничений представляют собой предикаты, обеспечивающие проверку входных данных.

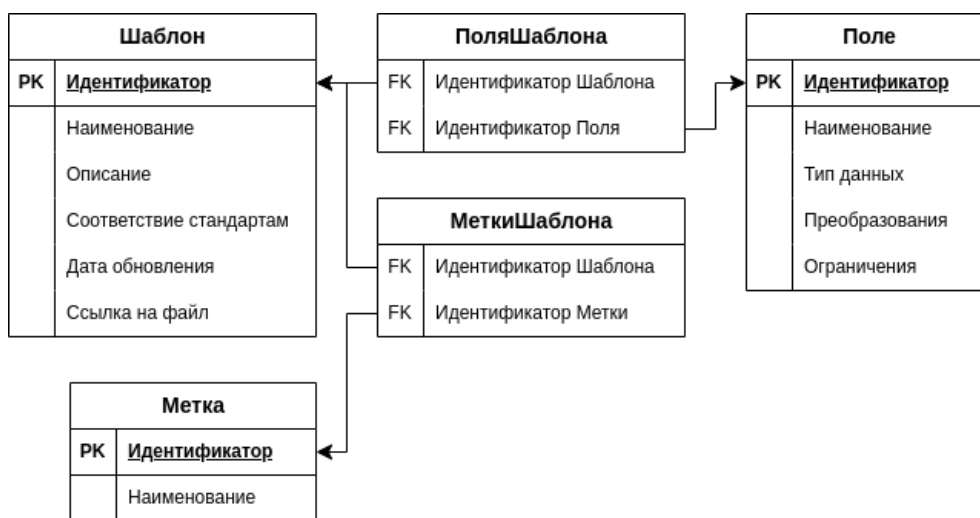


Рис. 4. База данных системы заполнения шаблонов

Пользовательский интерфейс системы состоит из двух основных веб-страниц: меню имеющихся в коллекции шаблонов и страницы заполнения шаблона данными. Основная идея состоит в том, чтобы использовать как можно меньше элементов для компактного и наглядного отображения всей необходимой информации.

Для быстрой навигации в системе предусмотрена навигационная строка. Она содержит гиперссылки на главную страницу, раздел справки, раздел для разработчиков, а также поисковую строку. В верхней части страницы имеется информационный виджет (элемент интерфейса), содержащий приглашение к выбору шаблона из списка и гиперссылку для загрузки пользовательского шаблона. Ниже располагается собственно меню, каждый пункт которого является раскрывающимся виджетом. В свёрнутом состоянии отображается только название шаблона, что способствует быстрой навигации. В раскрытом состоянии дополнительно отображаются описание шаблона, данные о соответствии стандартам, список меток и кнопка, позволяющая перейти к режиму заполнения, используя выбранный шаблон. Иллюстрация главной страницы системы представлена на рис. 5.

Заполнение шаблонов
Главная страница
Помощь
Для разработчиков

Ищете шаблон?

Поиск

Данный сервис предназначен для автоматизированного заполнения документов по шаблонам. Для начала работы выберите шаблон из списка ниже или [загрузите свой](#).

Налоговая декларация по НДФЛ (форма 3-НДФЛ)

^

Налог на доходы физических лиц, как правило, уплачивается автоматически – он удерживается с заработной платы. Но в ряде случаев физические лица должны самостоятельно рассчитать сумму налога и подать в налоговый орган декларацию по налогу на доходы физических лиц (форма 3-НДФЛ).

Утверждена Приказом ФНС России от 15.10.2021 N ЕД-7-11/903@

#налоги

Выбрать этот шаблон

Согласие на обработку персональных данных

^

Акт приема-передачи выполненных работ

^

Рис. 5. Главная страница веб-интерфейса системы

На странице заполнения шаблонов помимо упомянутой выше навигационной строки имеется строка управляющих кнопок, которые позволяют вернуться в меню выбора шаблона, добавить новый набор данных, загрузить таблицу данных и завершить заполнение. Ниже представлена собственно таблица данных. Её столбцы соответствуют полям в шаблоне, а строки – наборам данных. Для каждого набора системой будет создан отдельный документ. Иллюстрация страницы заполнения шаблона представлена на рис. 6.

Раздел справки предназначен для конечных пользователей и системных администраторов и содержит справочную информацию о порядке работы с системой и её веб-интерфейсом. Также рассматривается способ подготовки пользовательских шаблонов с помощью сторонних программных средств.

Раздел для разработчиков содержит техническую документацию о порядке работы с API системы.

Заполнение шаблонов
Главная страница
Помощь
Для разработчиков

Вернуться

Добавить набор данных

Загрузить таблицу

Готово

#	Фамилия	Имя	Отчество
1	Иванов	Иван	Иванович
2	Дубровский	Владислав	Игоревич
3	Ларина	Анна	Александровна

Рис. 6. Страница заполнения шаблона данными

## Заключение

На основании анализа существующих решений в области программ заполнения документов по шаблонам был сделан вывод об отсутствии таких решений, которые бы одинаково удовлетворяли потребности как индивидуальных пользователей, так и бизнеса. Главными выявленными недостатками являются:

- отсутствие встроенной коллекции шаблонов, использование которой позволило бы значительно увеличить производительность работы;

– отсутствие поддержки пакетного режима работы.

В настоящей работе описана модель системы заполнения шаблонов, стремящаяся устранить выявленные недостатки и сохранить выявленные преимущества. Уровень проработки модели оценивается автором как достаточный для построения программного решения на её основе.

Среди предусмотренных, но не описанных в настоящей работе доработок и улучшений следует выделить:

- возможность сохранения данных и результирующих документов на сервере,
- включение механизма авторизации для обеспечения безопасности при реализации предыдущего пункта,
- поддержку полиязычности пользовательского интерфейса,
- поддержку локального развёртывания с целью использования системы без доступа к Интернету или в условиях дополнительных ограничений.

### Библиографический список

1. *Рябунина О.С.* Разработка автоматизированной системы заполнения документов для приёма кандидатов на вакантное место в ООО «Газпром ПХГ» // *Инновации. Наука. Образование.* – 2022. – №50. – С. 1827-1832
2. Информационные технологии и управление предприятием / *В. В. Баронов, Г. Н. Калянов, Ю. Н. Попов, И. Н. Титовский.* – 2-е изд. – Саратов: Профобразование, 2019. – 327 с. – ISBN 978-5-4488-0086-3
3. Заполнение онлайн и создание (генерация) документов. // *r6r.ru* [Электронный ресурс]. – URL: <https://r6r.ru> (дата обращения: 27.05.2023)
4. eForms | The #1 website for free legal forms and documents. // *eforms.com* [Электронный ресурс]. – URL: <https://eforms.com> (дата обращения: 27.05.2023)
5. Заполнение документов, создание в Word, Excel. Автоматическое заполнение документов. Бесплатный сервис – DocWebService. // *docweb-service.ru* [Электронный ресурс]. – URL: <https://docweb-service.ru> (дата обращения: 27.05.2023)
6. Онлайн Генератор Документов. // *products.aspose.app* [Электронный ресурс]. – URL: <https://products.aspose.app/words/ru/assembly> (дата обращения: 27.05.2023)
7. Docassemble. // *docassemble.org* [Электронный ресурс]. – URL: <https://docassemble.org> (дата обращения: 27.05.2023)

### References

1. *Ryabunina O.S.* Development of an automated system for filling out documents for accepting candidates for a vacant position in OOO “Gazprom UGS”: *Innovations. Science. Education.* 2022. №50. pp. 1827-1832. (In Russ.)
2. *Baronov V.V., Kalyanov G.N., Popov Y.N., Titovsky I.N.* Information technologies and enterprise management. 2-nd ed. Saratov: Vocational education. 2019. 327 p. (In Russ.)
3. Online filling and creation (generation) of documents. // *r6r.ru* [Electronic resource]. – URL: <https://r6r.ru> (date of access: 05/27/2023)
4. eForms | The #1 website for free legal forms and documents. // *eforms.com* [Electronic resource]. – URL: <https://eforms.com> (date of access: 05/27/2023)
5. Filling out documents, creating in Word, Excel. Automatic completion of documents. Free service – DocWebService. // *docweb-service.ru* [Electronic resource]. – URL: <https://docweb-service.ru> (date of access: 05/27/2023)

6. Online Document Generator. // products.aspose.app [Electronic resource]. – URL: <https://products.aspose.app/words/en/assembly> (date of access: 05/27/2023)
7. Docassemble. // docassemble.org [Electronic resource]. – URL: <https://docassemble.org> (date of access: 05/27/2023)

## **DESIGNING ONLINE SERVICE WITH API FOR FILLING OUT DOCUMENTS FROM TEMPLATES**

*Kolpashchikov Maxim N., Knutova Nataliya S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, mxmblck@gmail.com

This article overviews the information systems designed for automated filling of documents according to templates. The general principles of their functioning are described, the concepts of the form of a document, meaningful data, and the merging procedure are introduced. The connection between template filling systems and document management systems is revealed. An analysis of existing solutions in the field of template filling systems is given. Based on the analysis carried out, an information system model was built, taking into account the advantages and disadvantages of existing solutions and offering a new solution of a universal nature. The constructed model includes a description of use cases from the point of view of the end user and a third-party information system, a description of the activities of all system components, a relational database schema, and a user interface layout.

**Keywords:** filling out documents according to a template, document management, online service, API.



## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОЙ ТРАНСКРИБАЦИИ РЕЧИ

*Костицын Владислав Витальевич, Кнутова Наталья Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kosticyn.vladislav@gmail.com

Технология транскрибации или распознавания речи является одним из важных направлений в IT-секторе. Это позволяет получать ценную информацию для различных сфер деятельности, таких как повседневная жизнь и рабочий процесс. Например, голосовой поиск и голосовые ассистенты используются для управления бытовыми устройствами, а автоматическое распознавание речи может помочь сохранить преимущество голосового обсуждения задач и удобство при обработке информации. Также транскрибация помогает людям легче понимать иноязычные материалы и может быть полезна для людей с нарушением слуха. В статье рассмотрены методы и технологии автоматического распознавания речи, спроектирована собственная система на базе современных технологий.

Ключевые слова: распознавание речи, программа транскрибации речи, транскрибация, сравнение программ для транскрибации разработка, информационная система приложение, разработка приложения.

Транскрибация [1] это расшифровка информации из аудио или видео в текстовую форму. Технология используется для формирования документации бизнес-встреч и собраний, составление скриптов продаж, добавления субтитров к видео, анализа или исследования речи лингвистами и много другого.

Целью транскрибации [2] является перевод речи в текстовый формат, который человек будет хорошо понимать при прочтении, при этом полностью передав смысл исходной речи.

Задачи расшифровки аудио:

- разобрать все, что сказал говорящий;
- разбить речь на предложения и абзацы;
- составление исходного текста речи;
- редактирования текста (убрать слова-паразиты, паузы).
- На данный момент не существует каких-либо обобщенных категорий транскрибации материала. Однако, исходя критериев для перевода аудио в текст [2], расшифровку можно разделить на несколько типов:

- по виду представленного аудио (Например: интервью, субтитры к видео, лекция и т.д.);
- по точности передачи текста (Например: дословная, художественная и т.д.).

Программное средство используемое для транскрибации называется транскрибатор. Транскрибатор – это программное обеспечение, которое позволяет автоматически переводить аудио- или видеозаписи в текстовый формат.

Среди главных преимуществ транскрибатора можно выделить высокую скорость работы и возможность значительно экономить временные и финансовые ресурсы, которые обычно тратят на найм переводчика.

Однако, недостатки транскрибатора тесно связаны с качеством переведенного текста. Ввиду того, что алгоритмы перевода не могут полностью запомнить всю произнесенную речь, в тексте могут появляться ошибки, неточности и смысловые искажения. Кроме того, перевод также включает в себя значительный объем времени, затрачиваемый на понимание множества специфических настроек и параметров. За счёт того, что информация сразу же загружается в RSS-файл, уведомление о выходе новой публикации сразу же передается пользователю.

Функции, которые должны быть реализованы в транскрибаторе речи, зависят от целей использования транскрибатора. Поэтому в первую очередь, будут рассмотрены функции обязательные для работы любого транскрибатора.

В основе любого транскрибатора лежит:

- распознавание и редактирование текста из аудиозаписи с помощью алгоритмов ML и нейронных сетей;
- Редактирование текста в программе транскрибатора позволяет исправлять ошибки распознавания и уточнять содержание текста, а также добавлять комментарии и пометать ключевые фразы и идеи для нахождения более точной информации;
- экспорт текста в различных форматах для дальнейшего использования;
- маркировка сегментов для удобства транскрибации и корректировки текста;
- воспроизведение аудио для проверки правильности распознавания и уточнения содержания текста;
- маркировка акцентов и интонаций для более точной транскрибации и понимания контекста сообщения.

Специфические требования включают в себя:

- поддержка работы с русским языком;
- поддержка работы с аудиофайлами и аудио записывающими устройствами;
- высокая точность распознавания текста.

При проектировании информационных систем необходимо использовать методы объектно-ориентированного подхода, а также иметь возможность описать функционал и структуру системы при помощи моделирования процессов. Для помощи при проектировании существуют специальные языки моделирования, с помощью которого появляется возможность описать объектную модель разрабатываемой информационной системы. Одним из самых популярных и востребованных языков является UML[3].

В общем, база данных в транскрибаторе используется для хранения любой информации, связанной с процессом транскрибации. СУБД в данной системе служит для хранения аудио файлов, метаданных аудио дорожек, словарей и результатов перевода. Все это будет использоваться для поиска и обработки полученных результатов перевода, а также для улучшения алгоритмов перевода в будущем. Поэтому для реализации подходит огромное количество СУБД например: MySQL, PostgreSQL, MongoDB, Oracle и так далее. MySQL имеет преимущества над остальными, так как имеет совместимость с выбранным ранее языком программирования Python. Поэтому выбор был остановлен на MySQL. Так же необходимо рассмотреть средство, которое может быть использовано для визуализации схемы базы данных для работы транскрибатора. Одним из самых популярных средств для этого является MySQL Workbench. MySQL Workbench – это CASE-средство для проектирования баз данных MySQL. MySQL Workbench позволяет создавать, модифицировать и управлять базами дан-

ных MySQL, а также проводить анализ данных и генерировать код SQL. В целом, выбор MySQL Workbench для проектирования схемы базы данных может быть хорошим выбором для разработки проектов, основанных на СУБД MySQL.

Для разработки системы перевода аудио дорожки в текст нужно рассмотреть языки программирования, на которых оно может быть написано. Для реализации данной системы существует множество языков программирования например: Python, Java, C++, Ruby и другие. Все языки имеют свои преимущества и свои недостатки, связанные с тем, какие имеют фреймворки, парадигмы и правила написания на языке. Однако Python предоставляет широкие возможности для создания систем автоматической транскрибации речи, и при достаточной экспертизе в области обработки звука и ML можно создать высокоэффективные и точные системы [4]. Для работы с Python подходит среда разработки PyCharm, так как он специально предназначен для языка программирования на Python, имеет бесплатную версию, которая имеет весь необходимый функционал и в нем можно редактировать, запускать и отлаживать Python-код сразу же после установки.

### Диаграмма прецедентов

Для удовлетворения всех требований функциональности создаваемого приложения была смоделирована диаграмма прецедентов, которая демонстрирует сценарии использования и взаимодействия пользователя с системой (см. рис. 1).

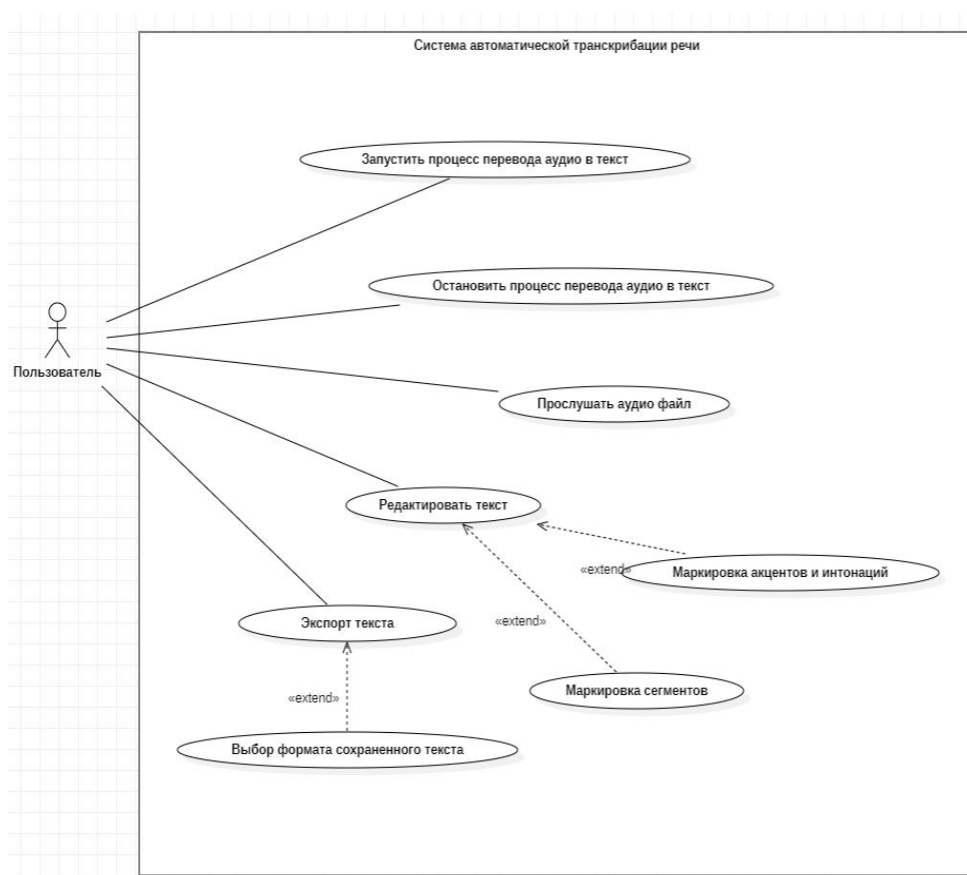


Рис. 1. Диаграмма взаимодействия пользователя и системы

Пользователь начинает взаимодействие с системой, загружая аудио файл. Затем система обрабатывает файл, распознает речь и переводит ее в текст. Результат перевода отобража-

ется пользователю. Также пользователь может отменить операцию, если заменит ошибки при переводе аудио в текст.

Пользователь имеет возможность перейти в окно для просмотра всех аудиофайлов, где у него будет возможность просмотреть и прослушать аудиофайлы, которые он загрузил в систему ранее.

Если пользователь перейдет в окно редактирования текста, то ему будет доступна возможность изменить полученный текст, добавить необходимые маркировки для сегментов и особенностей речи.

Пользователь может сохранить результат перевода в базу данных или на локальный диск. Пользователь также имеет возможность выбрать в каком формате система экспортирует результат. Система сохраняет результат и отображает сообщение об успешном сохранении.

### Проектирование модуля автоматической транскрипции речи

Для того чтобы разобраться в каких состояниях прибывает система в процессе своей работы воспользуемся диаграммой состояний (см. рис. 2).

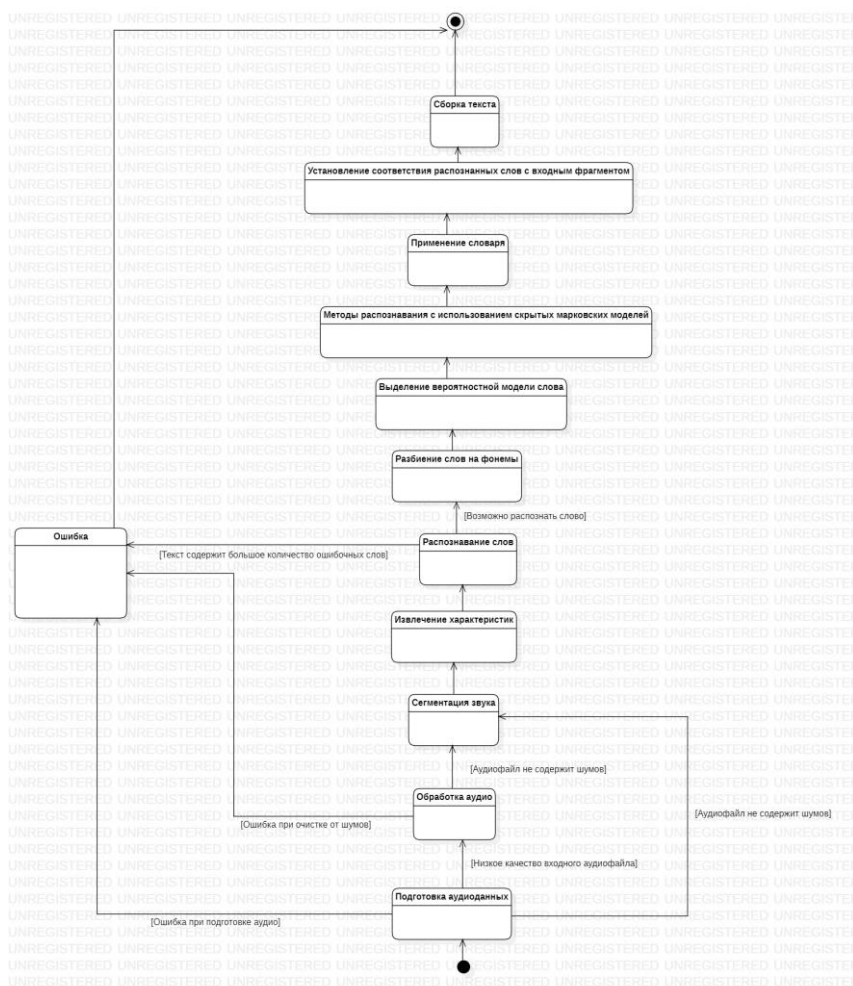


Рис. 2. Диаграмма работы модуля транскрипции

Диаграмма позволяет визуализировать процесс работы модуля автоматической транскрипции речи, а также понимать, какие состояния он может принимать и какие переходы между ними осуществляются.

Как результат работы данного модуля будет представлено текстовое представление речи, которая была записана на аудиофайле. Этот текст дальше передается в модуль постобработки для его анализа, добавление пунктуации, обработки и хранения.

### Диаграмма последовательности

С помощью диаграммы последовательностей можно увидеть, как за счет сообщений общаются части системы при выполнении определённых действий.

На данной диаграмме можно пронаблюдать жизненный цикл и сообщения между Пользователем, Системой и Базой данных (см. рис. 3).

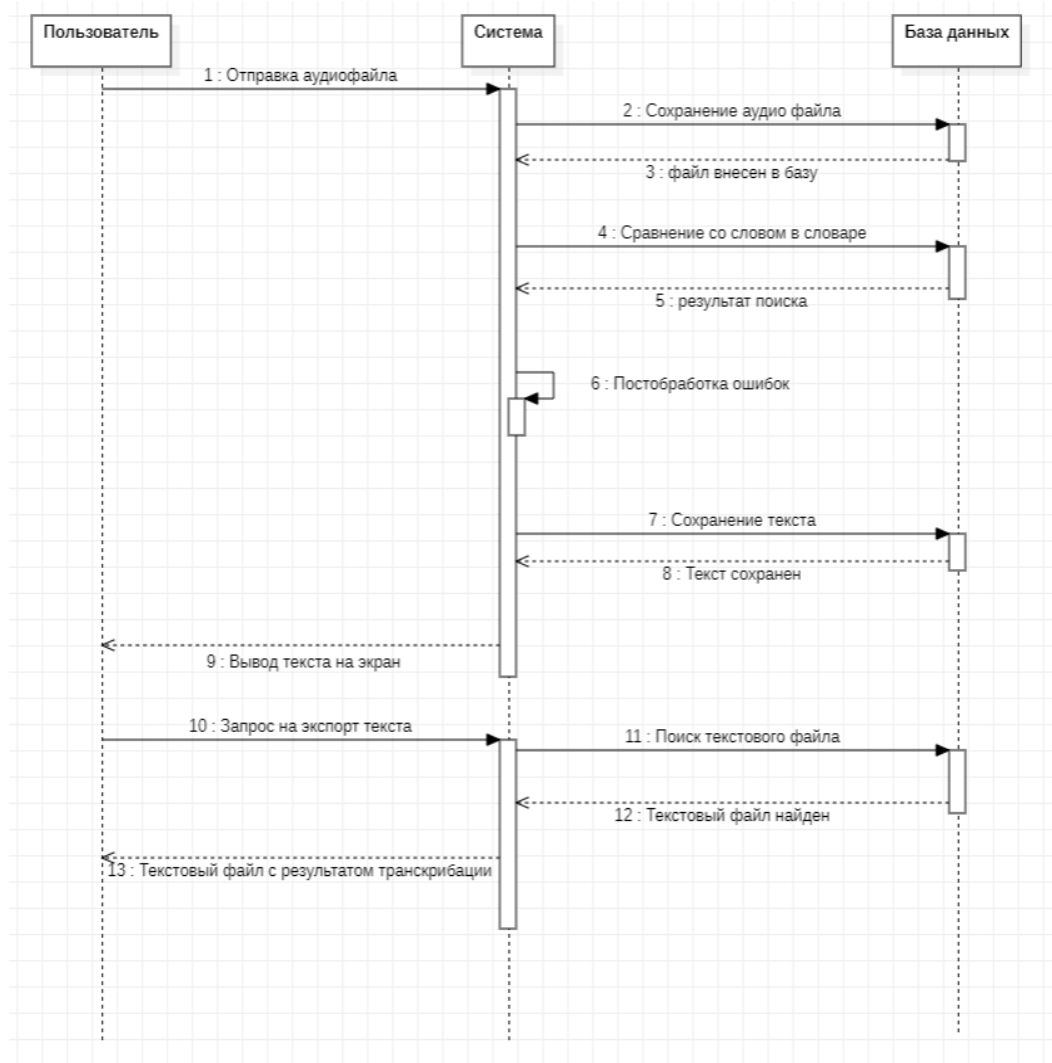


Рис. 3. Схема сообщений Пользователя, Системы и Базы данных

Диаграмма последовательности показывает последовательность шагов, которые выполняются для преобразования речи в текст и получения пользователем файла с результатом.

Диаграмма начинается с того, что пользователь отправляет системе аудиофайл. Система сохраняет полученный аудиофайл. Затем система использует алгоритм распознавания речи для преобразования звука в текст. В процессе распознавания система проверяет правильность преобразования. Результат сохраняется. Полученный текст выводится на экран в реальном времени. В конце процесса произнесенная речь сохраняется в формате текстового файла или базы данных для дальнейшей обработки и анализа.

## Проектирование базы данных

База данных для системы автоматического перевода аудио в текст будет состоять из четырех таблиц (см. рис. 4).

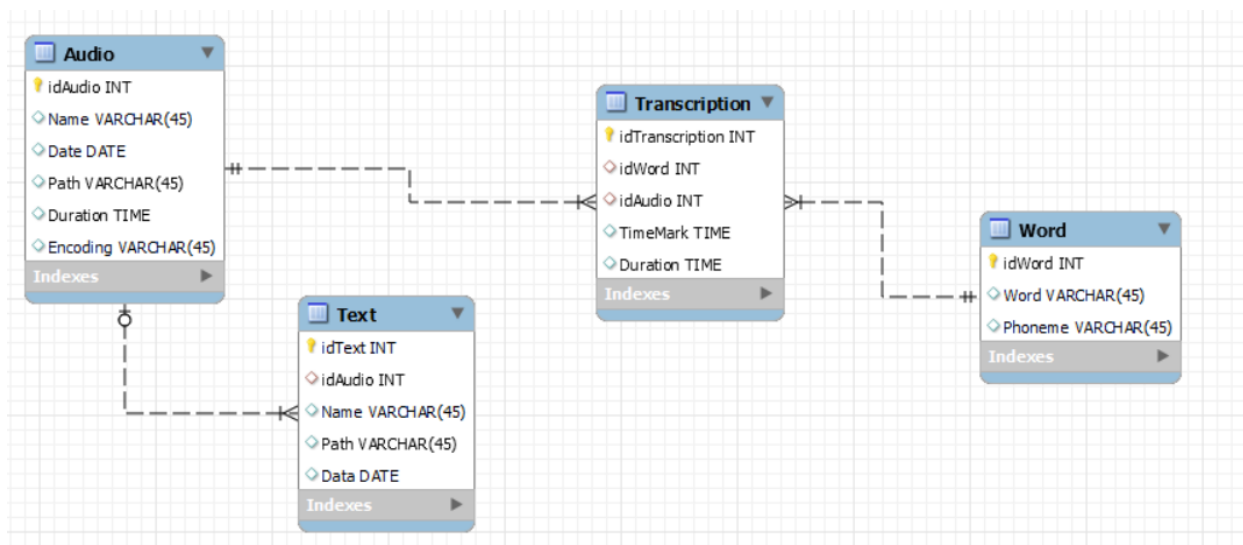


Рис. 4. База данных

Система автоматического транскрибирования аудио в текст будет использовать эти таблицы для хранения данных и обеспечения связи между ними. Каждый раз, когда пользователь загружает аудиофайл в систему, он будет сохраняться в таблице "Audio" и проходить процесс транскрибирования. Фонема и транскрипция каждой фразы будет храниться в таблице "Word", а каждое встречающееся слово будет проверяться в таблице "Transcription". Путь и имя для полученного в результате транскрибирования текста будет храниться в таблице Text.

Когда пользователь запрашивает транскрибированный текст для конкретного аудиофайла, система будет связывать данные этих таблиц и выводить результат на экран пользователя.

## Проектирование пользовательского интерфейса системы

Общими требованиями к интерфейсу являются его простота и легкость использования, доступность для пользователей с любым уровнем опыта, а также возможность расширения и добавления дополнительных функций в будущем. На интерфейсе размещены (см. рис. 5): поле для загрузки аудиофайла из локального хранилища, кнопка "Перевод", которая запускает процесс транскрипции аудио в текст, поле вывода текста, которое содержит результат транскрипции аудиофайла и позволяет редактирование, индикатор выполнения для уведомления пользователя о ходе транскрипции, кнопка "Остановить", чтобы прервать выполнение транскрипции, кнопка "Прослушать" для прослушивания аудиофайла и кнопка "Экспорт", которая позволяет выбрать формат сохранения и экспортировать результат транскрипции.

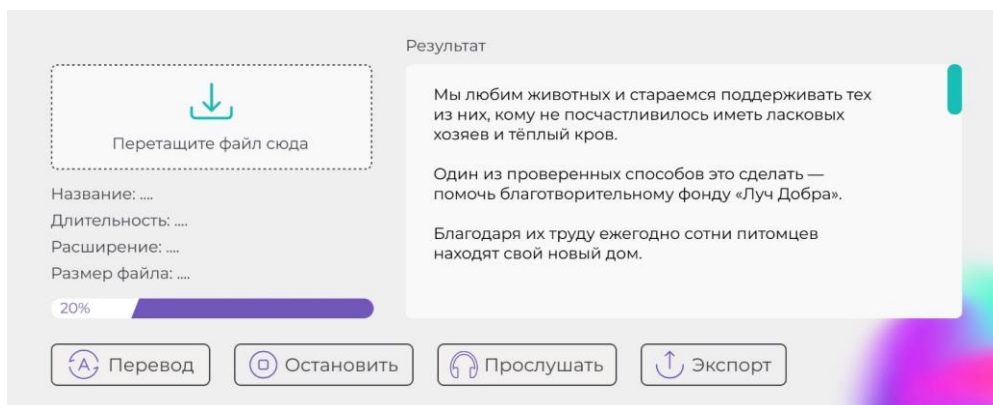


Рис. 5. Интерфейс системы

Окно (см. рис. 6), открытое при нажатии кнопки "Прослушать", содержит кнопки управления воспроизведением, график волны, регулятор громкости, поле для выбора участка и информацию о файле; есть также кнопка с крестиком для возврата к основному окну системы.

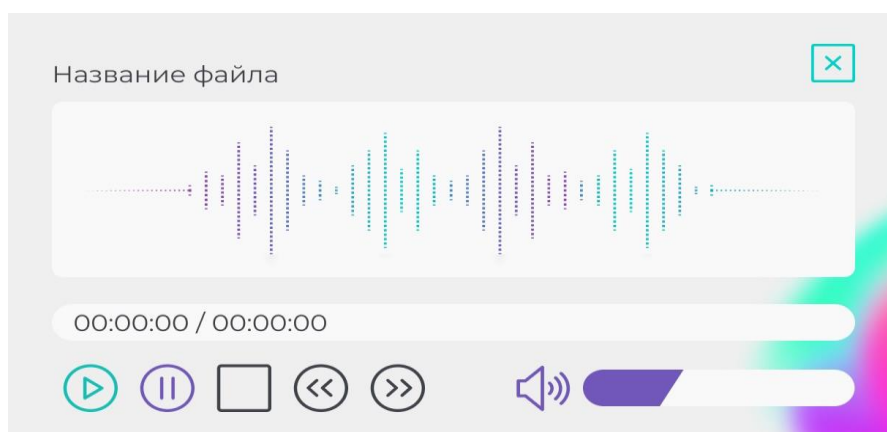


Рис. 6. Окно прослушивания аудио

### Библиографический список

1. What Does Transcribe Mean & Why Do You Need It? [Электронный ресурс]. – URL: <https://www.rev.com/blog/resources/what-does-transcribe-mean> (дата обращения: 22.05.2023)
2. Что такое транскрибирование [Электронный ресурс]. – URL: <https://mk-translations.ua/ru/blog/cto-takoe-transkribovanie/> (дата обращения: 22.05.2023)
3. StarUML [Электронный ресурс] URL: <https://staruml.io/> (дата обращения: 27.05.2022)
4. Python (in Machine Learning) [Электронный ресурс]. – URL: <https://corporatefinanceinstitute.com/resources/data-science/python-in-machine-learning/> (дата обращения: 22.05.2023)
5. SpeechPad [Электронный ресурс]. – URL: <https://speechpad.ru/> (дата обращения: 22.05.2023)
6. Python 3.11.3 documentation [Электронный ресурс]. – URL: <https://mybibliography.ru/sayt> (дата обращения: 22.05.2023)
7. Java Documentation [Электронный ресурс]. – URL: <https://docs.oracle.com/en/java/> (дата обращения: 22.05.2023)
8. C++ Programming Language [Электронный ресурс]. – URL: <https://devdocs.io/cpp/> (дата обращения: 22.05.2023)

9. Documentation – Ruby [Электронный ресурс]. – URL: <https://www.ruby-lang.org/en/documentation/> (дата обращения: 22.05.2023)
10. PyCharm Guide Well-organized collection of learning resources for PyCharm [Электронный ресурс]. – URL: <https://www.jetbrains.com/pycharm/guide/> (дата обращения: 22.05.2023)
11. Welcome to Spyder's Documentation [Электронный ресурс]. – URL: <https://docs.spyder-ide.org/current/index.html> (дата обращения: 22.05.2023)
12. Eclipse Documentation [Электронный ресурс]. – URL: <https://www.eclipse.org/documentation/> (дата обращения: 22.05.2023)
13. Google Cloud [Электронный ресурс]. – URL: <https://cloud.google.com/speech-to-text> (дата обращения: 22.05.2023)
14. Биктимиров, А. Р. Boosting speech-to-text software potential / А. Р. Биктимиров, Д. Ю. Груздев // Research result. theoretical and applied linguistics. – 2022. – Т. 8, № 4. – С. 72-89. – ISSN 2313-8912
15. Dictation.io [Электронный ресурс]. – URL: <https://dictation.io/> (дата обращения: 22.05.2023)
16. Yandex Cloud [Электронный ресурс]. – URL: <https://cloud.yandex.ru/services/speechkit> (дата обращения: 22.05.2023)
17. RealSpeaker [Электронный ресурс]. – URL: <https://www.realspeaker.net/> (дата обращения: 22.05.2023)
18. Благой, А. В. Информационная система для транскрибации аудиозаписей / А. В. Благой // Студент: наука, профессия, жизнь: Материалы IX всероссийской студенческой научной конференции с международным участием. В 4-х частях, Омск, 25–29 апреля 2022 года. Том Часть 1. – Омск: Омский государственный университет путей сообщения, 2022. – С. 72-76. – EDN QLURYF.
19. Иванов, Д. И. Транскрибация – современная стенография / Д. И. Иванов // НАУКА. Технологии. Инновации – 2023: Сборник статей Международной научно-практической конференции, Петрозаводск, 06 февраля 2023 года. – Петрозаводск: Международный центр научного партнерства «Новая Наука» (ИП Ивановская И.И.), 2023. – С. 107-112. – EDN KUUMSW.

### References

1. What Does Transcribe Mean & Why Do You Need It? [Electronic resource]. – URL: <https://www.rev.com/blog/resources/what-does-transcribe-mean> (дата обращения: 22.05.2023)
2. CHto takoe transkribovanie [Electronic resource]. – URL: <https://mk-translations.ua/ru/blog/chto-takoe-transkribovanie/> (дата обращения: 22.05.2023) (In Russ)
3. StarUML [Электронный ресурс] URL: <https://staruml.io/> (data obrashcheniya: 27.05.2022)
4. Python (in Machine Learning) [Electronic resource]. – URL: <https://corporatefinanceinstitute.com/resources/data-science/python-in-machine-learning/> (дата обращения: 22.05.2023)
5. SpeechPad [Electronic resource]. – URL: <https://speechpad.ru/> (data obrashcheniya: 22.05.2023)
6. Python 3.11.3 documentation [Electronic resource]. – URL: <https://mybibliography.ru/sayt> (дата обращения: 22.05.2023)
7. Java Documentation [Electronic resource]. – URL: <https://docs.oracle.com/en/java/> (дата обращения: 22.05.2023)



8. C++ Programming Language [Electronic resource]. – URL: <https://devdocs.io/cpp/> (дата обращения: 22.05.2023)
9. Documentation – Ruby [Electronic resource]. – URL: <https://www.ruby-lang.org/en/documentation/> (дата обращения: 22.05.2023)
10. PyCharm Guide Well-organized collection of learning resources for PyCharm [Electronic resource]. – URL: <https://www.jetbrains.com/pycharm/guide/> (дата обращения: 22.05.2023)
11. Welcome to Spyder's Documentation [Electronic resource]. – URL: <https://docs.spyder-ide.org/current/index.html> (дата обращения: 22.05.2023)
12. Eclipse Documentation [Electronic resource]. – URL: <https://www.eclipse.org/documentation/> (дата обращения: 22.05.2023)
13. Google Cloud [Electronic resource]. – URL: <https://cloud.google.com/speech-to-text> (дата обращения: 22.05.2023)
14. *Biktimirov, A. R.* Boosting speech-to-text software potential / A. R. Biktimirov, D. YU. Gruzdev // Research result. theoretical and applied linguistics. – 2022. – Т. 8, № 4. – С. 72-89. – ISSN 2313-8912
15. Dictation.io [Electronic resource]. – URL: <https://dictation.io/> (дата обращения: 22.05.2023)
16. Yandex Cloud [Electronic resource]. – URL: <https://cloud.yandex.ru/services/speechkit> (дата обращения: 22.05.2023) (In Russ)
17. RealSpeaker [Electronic resource]. – URL: <https://www.realspeaker.net/> (дата обращения: 22.05.2023)
18. *Blagoj, A. V.* Informacionnaya sistema dlya transkribacii audiozapisej / A. V. Blagoj // Student: nauka, professiya, zhizn': Materialy IX vs Rossijskoj studencheskoj nauchnoj konferencii s mezhdunarodnym uchastiem. V 4-h chastyah, Omsk, 25–29 aprelya 2022 goda. Tom CHast' 1. – Omsk: Omskij gosudarstvennyj universitet putej soobshcheniya, 2022. – С. 72-76. – EDN QLURYF. (In Russ)
19. *Ivanov, D. I.* Transkribaciya – sovremennaya stenografiya / D. I. Ivanov // NAUKA. Tekhnologii. Innovacii – 2023: Sbornik statej Mezhdunarodnoj nauchno-prakticheskoy konferencii, Petrozavodsk, 06 fevralya 2023 goda. – Petrozavodsk: Mezhdunarodnyj centr nauchnogo partnerstva «Novaya Nauka» (IP Ivanovskaya I.I.), 2023. – С. 107-112. – EDN KUUMSW. (In Russ)

## **DESIGN AND DOCUMENTATION OF AUTOMATIC SPEECH TRANSCRIPTION SYSTEM**

*Kostitsyn Vlavislav V., Knutova Nataliya S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [kosticyn.vladislav@gmail.com](mailto:kosticyn.vladislav@gmail.com)

The technology of transcription or speech recognition is one of the important areas in the IT sector. This allows you to receive valuable information for various areas of activity, such as daily life and workflow. For example, voice search and voice assistants are used to control home devices, and automatic speech recognition can help maintain the advantage of voice discussion of tasks and ease of processing information. Transcription also helps people understand foreign language materials more easily and can be useful for people with hearing impairments. The article will discuss the methods and technologies of automatic speech recognition, as well as analyze existing systems and design your own system based on modern technologies.

**Keywords:** speech recognition, speech transcription program, transcription, comparison of transcription programs development, information system application, application development.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ УЧЕТА ЗАБОЛЕВАЕМОСТИ В ПЕРМИ

*Кострикин Егор Алексеевич, Василюк Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kegor5122@gmail.com

В работе были рассмотрены основные инструменты, применяемые для проектирования и документирования информационной системы учёта заболеваемости в городе Пермь. Был определён предмет и объект исследования, а также обоснована актуальность разработки данной информационной системы. Задачи, которые необходимо решить в процессе проектирования, были поставлены. Проведён анализ существующих информационных систем, выполняющих аналогичные функции, что позволило выявить требования к разрабатываемой системе. В результате проведенного анализа была обоснована целесообразность создания проектируемой системы и выбора метода анализа инвестиционных проектов. Для проектирования информационной системы был выбран программный инструмент StarUML [1], который обладает необходимым функционалом для создания моделей системы. Для разработки вайрфрейма приложения было выбрано приложение Balsamiq [2]. В процессе проектирования информационной системы было описано взаимодействие с ней со стороны пользователей, а также спроектирован интерфейс системы на основе созданных диаграмм.

Ключевые слова: информационная система, учёт заболеваемости, UML-моделирование.

В современном обществе информационные технологии широко используются в различных сферах жизнедеятельности, в том числе и в здравоохранении. Здравоохранение является одной из ключевых сфер жизни общества, оказывающей непосредственное влияние на качество жизни каждого человека. В условиях современных вызовов, связанных с распространением инфекционных и хронических заболеваний, обеспечение эффективного учета и контроля заболеваемости становится особенно важным.

Система анализа может позволить проводить оперативный мониторинг ситуации, касаемой заболеваемости, а также проводить анализ больших объёмов данных и осуществлять прогнозы на основе проведенного анализа.

Объектом исследования является учет заболеваемости в Перми, а предметом исследования – автоматизация этого процесса.

Целью работы является проектирование и документирование системы учёта заболеваемости в городе Пермь.

Авторами исследования были поставлены следующие задачи:

- Анализ и сравнение существующих аналогов информационной системы;
- Выбор инструментального ПО;
- Определение требований к информационной системе учета заболеваемости на основе стандартов и нормативных документов, а также потребностей пользователей;
- Проектирование информационной системы;
- Документирование ИС, а именно создание ТЗ.

Проектирование новой системы учёта заболеваемости является целесообразным, основываясь на недостатках существующих аналогов в России. Например, многие информационные системы учёта заболеваемости имеют устаревший интерфейс, что затрудняет работу пользователей и уменьшает эффективность системы. Также часто отсутствуют стандарты для внедрения систем, что приводит к сложностям при интеграции с другими информационными системами и увеличивает затраты на поддержку и развитие системы. Кроме того, многие системы являются громоздкими и сложными, что затрудняет их использование и требует дополнительных затрат на обучение сотрудников и поддержку системы. Разработка новой системы учёта заболеваемости с современным интерфейсом и более простой, и удобной в использовании, может значительно улучшить эффективность и точность учёта заболеваемости, а также снизить затраты на поддержку и развитие системы.

Для моделирования поведения и структуры информационной системы был выбран язык моделирования общего назначения UML. Этот язык выбран из-за его многофункциональности и возможностей, которые достаточны для моделирования данной информационной системы. В процессе моделирования были созданы две модели: модель поведения информационной системы и модель внутренней структуры.

Логика приложения представлена при помощи диаграммы прецедентов (рис. 1).

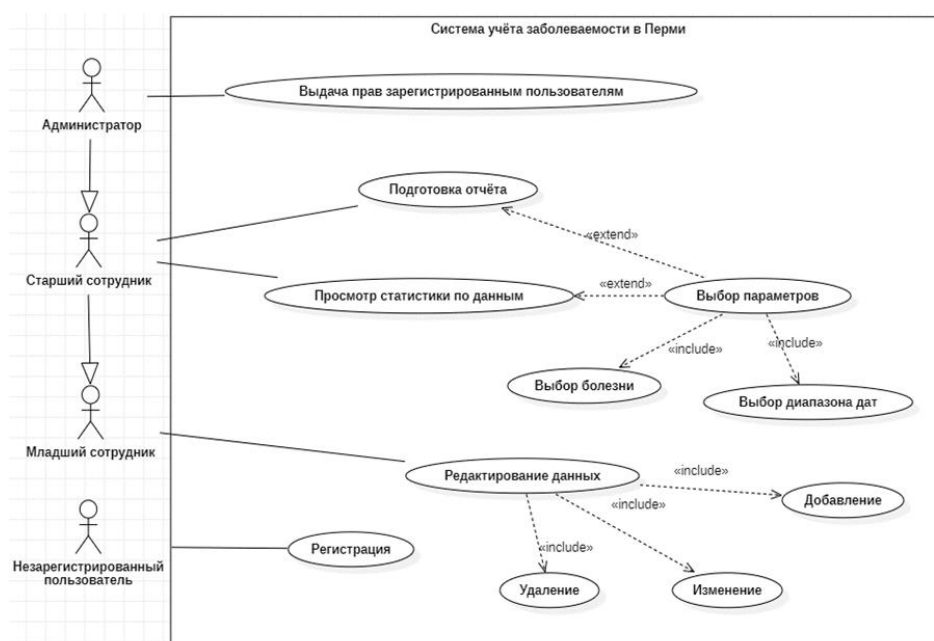


Рис. 1. Диаграмма прецедентов

Проектирование логики системы учёта заболеваемости должно учитывать различия между пользователями.

В проектируемой системе младший сотрудник может редактировать данные о пациентах, старший сотрудник может редактировать данные о пациентах, просматривать статистические данные по болезням и создавать отчёты на основе этих данных. Администратор может проводить все вышеперечисленные операции, а также изменять права пользователей в системе.

Кроме того, в системе могут существовать незарегистрированные пользователи, которые до выдачи прав администратором не могут осуществлять действий в проектируемой системе.

Для отображения взаимодействия в системе, была создана диаграмма классов (рис. 2).

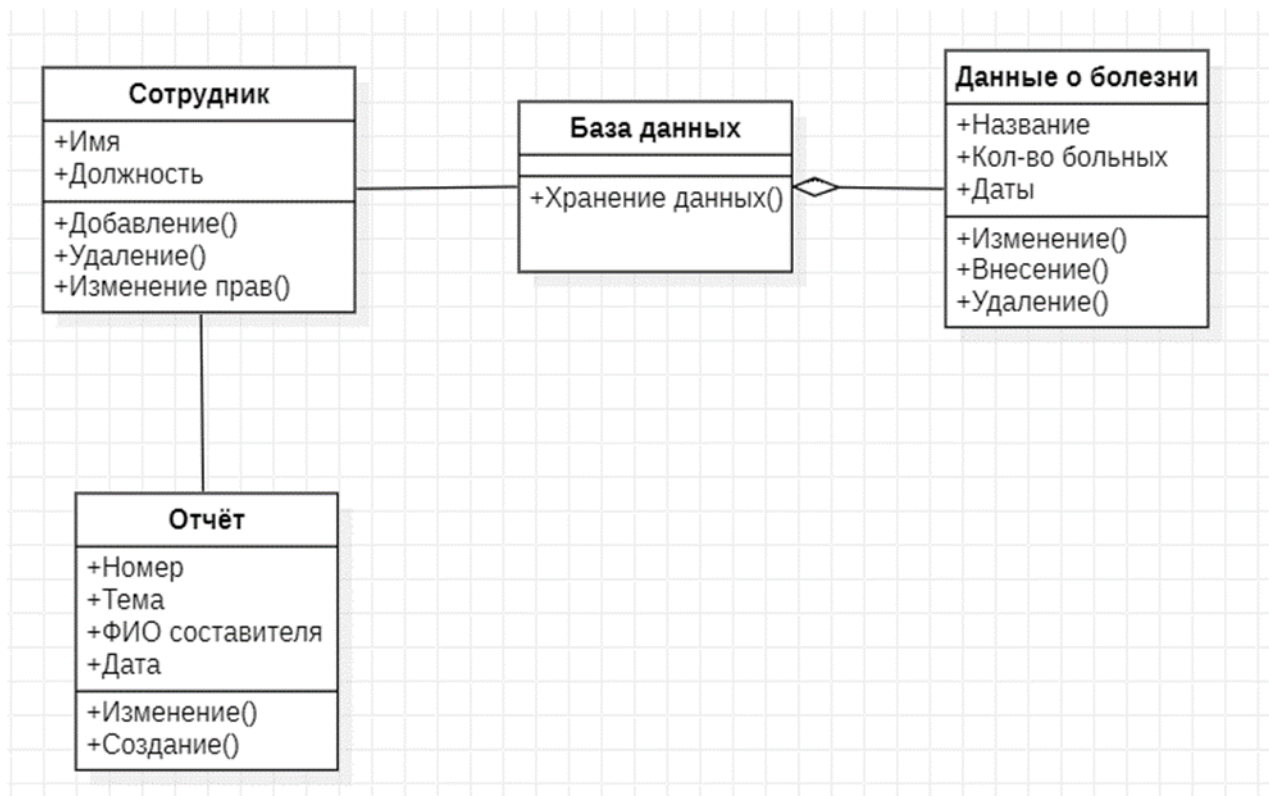


Рис. 2. Диаграмма классов

С помощью диаграммы классов можно производить анализ и проектирование информационной системы, а также определять связи между объектами в системе. В данном случае таких сущностей четыре: сотрудник, БД, данные о болезни и отчёт. В создании классов для каждого отдельного пользователя нет смысла, т.к. незарегистрированного пользователя нет в системе, а сотрудники внутри системы различаются только правами.

Вайрфрейм является важным инструментом при проектировании информационной системы, включая систему учета заболеваемости. Он представляет собой низкоуровневый концептуальный чертеж, который визуализирует основные элементы и структуру пользовательского интерфейса. Он позволяет определить последовательность действий пользователя в системе и взаимодействие с различными элементами интерфейса. Это помогает обнаружить возможные проблемы с навигацией и улучшить пользовательский опыт. Оптимизация потока взаимодействия может улучшить эффективность использования системы и повысить удовлетворенность пользователей. Кроме того, вайрфрейм может использоваться для проведения тестирования прототипа системы с реальными пользователями. Он позволяет оценить удобство использования, выявить проблемные места и собрать отзывы пользователей. Полученные результаты могут быть использованы для внесения изменений и улучшения дизайна системы до финальной реализации.

При запуске приложения открывается окно авторизации пользователей (рис. 3).

Рис. 3. Окно авторизации

На форме авторизации пользователю предлагается ввести свои логин и пароль, после чего он должен будет нажать на кнопку войти. В случае если у пользователя нет аккаунта, он должен нажать на кнопку «Регистрация», после чего откроется форма регистрации.

Для внесения данных о пациенте используется следующая форма (рис. 4).

Рис. 4. Внесение данных о пациенте

На рисунке представлена форма внесения данных о пациенте. В ней сотрудник вносит ФИО пациента, его данные, болезнь, дата заболевания и дата выздоровления при наличии, так как данная форма совпадает с формой изменения данных о пациенте. Перед выходом из формы необходимо нажать на кнопку «Сохранить изменения», иначе данные не будут сохранены. Также в правом нижнем углу расположена кнопка справки, которая пояснит поля на форме. При завершении внесения, данные сразу заносятся в БД и на их основе можно по-

лучить статистику. При завершении внесения, данные сразу заносятся в БД и на их основе можно получить статистику с помощью следующей формы (рис. 5).

**Получение статистики**

Выбор болезни ▼

Добавить болезнь

Первая дата

//



Вторая дата

//



Получить статистику



Рис. 5. Получение статистики

Сотрудник должен выбрать заболевание и даты, по которым составляется отчёт. При необходимости сравнения данных по нескольким заболеваниям или для экономии времени есть возможность добавить ещё одну болезнь с помощью кнопки «Добавить болезнь». В итоге после внесения всех параметров и нажатия кнопки «Получить статистику» сотрудник получит график со всеми данными и заболеваниями или о заболеваниях в соответствии с выбранными параметрами.

Для того, чтобы у сотрудника была возможность получения статистики, сотрудник должен обладать соответствующими правами, которые выдаёт администратор в следующей форме (рис. 6).

**Выдача прав сотрудникам**

Выбор пользователя ▼

Выбор прав ▼

Сохранение изменений




Рис. 6. Выдача прав пользователям

Используя данную форму, администратор наделяет правами сотрудников в соответствии с их должностями. В ней администратор системы выбирает сотрудника и права для него в соответствующих окнах. После этого он должен нажать кнопку «Сохранить изменения», чтобы сохранить их права.

В результате работы была спроектирована информационная система учёта заболеваемости в Перми. Затем данную информационную систему планируется реализовать в виде сайта и будет доступна через сеть Интернет с помощью любого браузера.

#### **Библиографический список**

1. Introduction – StarUML documentation // StarUML [Электронный ресурс]. – URL: <https://docs.staruml.io> (дата обращения: 01.05.2023).
2. Balsamiq Documentation // Balsamiq [Электронный ресурс]. – URL: <https://balsamiq.com/docs> (дата обращения: 05.05.2023).

#### **References**

1. Introduction – StarUML documentation // StarUML [Electronic resource]. – URL: <https://docs.staruml.io> (accessed: 01.05.2023).
2. Balsamiq Documentation // Balsamiq [Electronic resource]. – URL: <https://balsamiq.com/docs> (accessed: 05.05.2023).

### **DESIGN AND DOCUMENTATION OF THE MORBIDITY ACCOUNTING SYSTEM IN PERM**

*Kostrikin Egor A., Vasiluk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, shubintseva2014@yandex.ru

The paper considered the main tools used for designing and documenting the information system of morbidity accounting in the city of Perm. The subject and object of the study were determined, and the relevance of the development of this information system was justified. The tasks that need to be solved in the design process have been set. The analysis of existing information systems performing similar functions was carried out, which made it possible to identify the requirements for the system being developed. As a result of the analysis, the expediency of creating a projected system and choosing a method for analyzing investment projects was justified. For the design of the information system, the software tool StarUML was chosen, which has the necessary functionality to create models of the system. Balsamiq application was chosen to develop the wireframe of the application. In the process of designing the information system, the interaction with it from the users was described, and the system interface was designed based on the created diagrams.

**Keywords:** information system, morbidity accounting, UML modeling.

## **ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ «КАЧЕСТВО ВОДЫ В РОДНИКАХ ПЕРМСКОГО КРАЯ»**

*Кузаев Айдар Файзуллович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, kuzae vaydar@mail.ru

*Шестаков Кирилл Николаевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, zeldaone@mail.ru

Рассматриваются проектирование, разработка и документирование геоинформационной системы для хранения информации о родниках Пермского края и качестве воды в них. Изучаются существующие аналогичные информационные системы. На основе их анализа формулируются основные функциональные требования для проектируемого приложения. Описывается обоснованный выбор инструментов проектирования и разработки, а также создание прототипа системы. При проектировании используется язык моделирования UML и CASE-средство Diagrams.net. Для хранения данных выбирается СУБД MySQL. При разработке серверной части системы применяется язык программирования PHP, для реализации работы с картами и для визуализации – язык JavaScript. Прототип интерфейса информационный системы реализуется в графическом редакторе Figma.

Ключевые слова: информационная система, проектирование, родник, качество воды, Пермский край.

### **Введение**

Актуальность данной темы обусловлена тем, что вопрос о пригодности воды из открытых водоемов для употребления всегда стоял остро. Даже в современном мире, когда многие используют воду из-под кранов или из артезианских скважин, родники в Пермском крае являются одним из источников питьевой воды для населения. Поэтому информирование населения о таких источниках является нужным и важным. Также можно указать ещё на ряд причин в пользу этого.

Во-первых, качество питьевой воды – это один из основных факторов, влияющих на здоровье жителей региона. Поэтому важно обеспечить информирование о качестве воды в родниках, особенно для находящихся вблизи населенных пунктов.

Во-вторых, в последние годы наблюдается увеличение количества загрязнений и негативного воздействия на окружающую среду. Это может привести к ухудшению качества воды в родниках и повышению риска для здоровья людей, использующих эту воду для питья.

В-третьих, создание информационной системы обеспечит возможность оперативного информирования о качестве воды в родниках, что может поднять эту проблему в информационном поле региона.

В-четвертых, правительство Пермского края в апреле 2022 г. утвердило документ «Стратегия развития туризма в Пермском крае на период до 2035 года» [1], которая направ-



лена на долгосрочное устойчивое комплексное развитие туристско-рекреационного комплекса нашего региона, формирование и рост внутренних и въездных туристских потоков в Прикамье.

В связи с вышесказанным, разрабатываемая информационная система может быть полезна не только специалистам в области экологии и водоснабжения, но и туристам, путешественникам, а также местным жителям, которые используют воду из родников для питья и приготовления пищи.

Объектом исследования в данной работе является качество родниковой воды в Пермском крае. Геоинформационная система, реализующая функционал удобного информирования о родниках и качестве воды в них, выступает предметом исследования.

Целью является проектирование, разработка и документирование геоинформационной системы «Качество воды в родниках Пермского края».

Задачи, которые необходимо выполнить для достижения поставленной цели:

- изучение предметной области темы работы;
- поиск и анализ аналогов разрабатываемого приложения;
- выбор средств проектирования и разработки;
- проектирование модели информационной системы;
- разработка прототипа пользовательского интерфейса;
- разработка рабочего прототипа и документирование информационной системы.

Оценка качества питьевой воды включает в себя оценку различных параметров, таких как физические, химические, биологические, бактериологические и радиологические показатели. В СанПиН 2.1.3684-21 [2] указано, что «выбор показателей химического состава питьевой воды, подлежащих постоянному производственному контролю, проводится для каждой системы водоснабжения на основании анализа результатов расширенных исследований химического состава воды источников питьевого водоснабжения, а также технологии водоподготовки в системе водоснабжения».

К основным факторам, которые рассматриваются при анализе воды, относятся органолептические показатели, жесткость, содержание железа, водородный показатель вещества (рН), общая минерализация воды, содержание нитратов, наличие живых организмов и паразитов.

### **Моделирование системы**

Рассмотрим процесс проектирования и моделирования информационной системы с использованием языка графического описания UML (Unified Modeling Language) [3]. Вначале составляется поведенческая модель – создаются UML-диаграммы прецедентов и деятельности.

Диаграммы прецедентов несут в себе информацию об основных взаимодействиях с системой в упрощённом виде. Выделены три актора для разрабатываемой ИС:

- «Гость» (незарегистрированный пользователь, взаимодействующий с системой);
- «Авторизованный пользователь» (пользователь, прошедший процесс авторизации);
- «Суперпользователь» (администратор, имеющий особые права по модерации содержательной части системы).

Диаграмма прецедентов для актора «Суперпользователь» представлена ниже (см. рис. 1).

Перечислим основные прецеденты для актора «Суперпользователь»:

- модерирование материалов системы (осуществление контроля за публичными материалами, размещающимися в ИС);
- модерирование комментариев (удаление недобросовестных комментариев);
- внесение изменений в существующие карточки родников (обновление информации о родниках);
- наделение правами суперпользователя зарегистрированных пользователей (присвоение роли «Суперпользователь» обычным зарегистрированным пользователем);
- верификация информации о предложенных карточках родников (рецензия присланных карточек родников от пользователей, с последующим выводом или удалением);
- подтверждение фотографий, прикрепленных к родникам (подтверждать публикацию фотографий родников);
- подтверждение документов об экспертизах родников (подтверждать публикацию документов, доказывающих проведение экспертиз воды в выбранном роднике);
- подтверждение расположения родника (подтверждение координат предложенных родников).

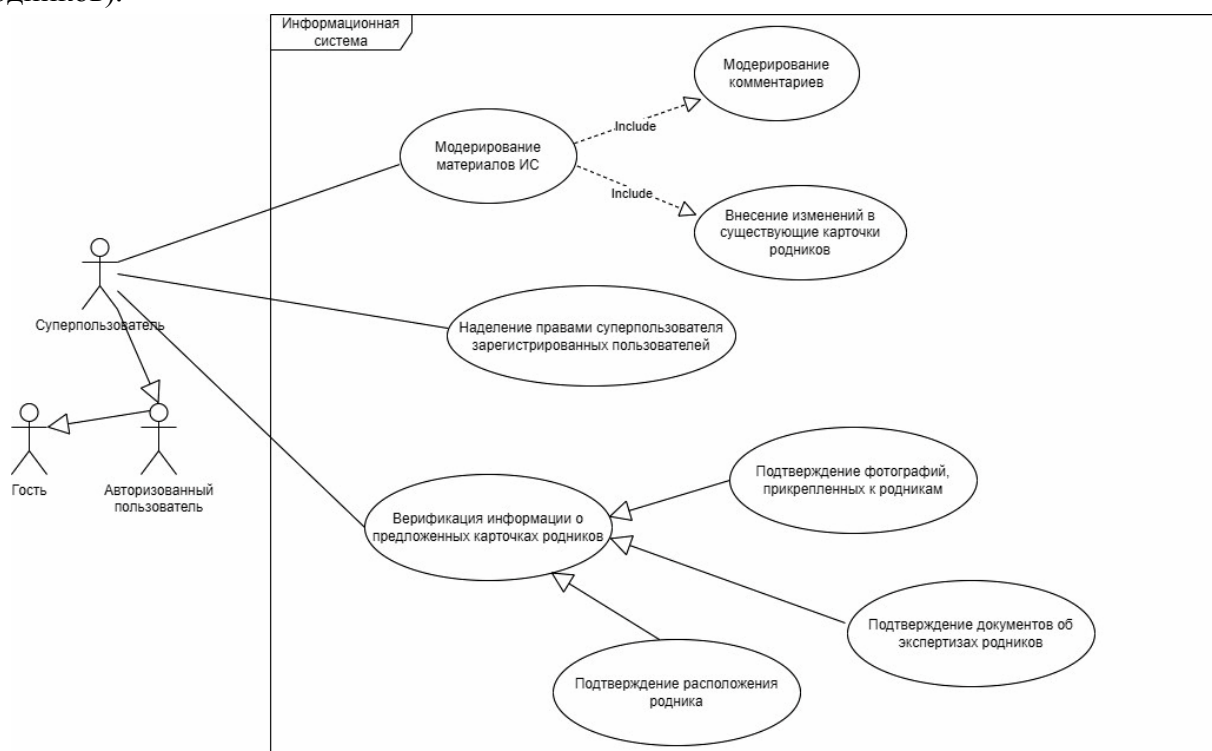


Рис. 1. Диаграмма прецедентов для актора «Суперпользователь»

Диаграмма деятельности (см. рис. 2) позволяет более наглядно смоделировать конкретный случай использования. В диаграмме деятельности отображены аспекты поведения системы в динамике. Проще говоря, эта диаграмма является блок-схемой, которая показывает переходы потока управления от одной деятельности к другой. На диаграмме располагаются активности на «плавательных дорожках», каждая из которых визуализирует поведение одного из объектов.

Взаимодействие пользователя с информационной системой начинается с открытия приложения. Следующим действием он может войти в систему (если он уже зарегистрирован), зарегистрироваться или же ознакомиться с ИС в режиме гостя. Диаграмма деятельности на рис. 2 наглядно показывает, что у авторизованных пользователей больше вариантов взаимодействия с системой.

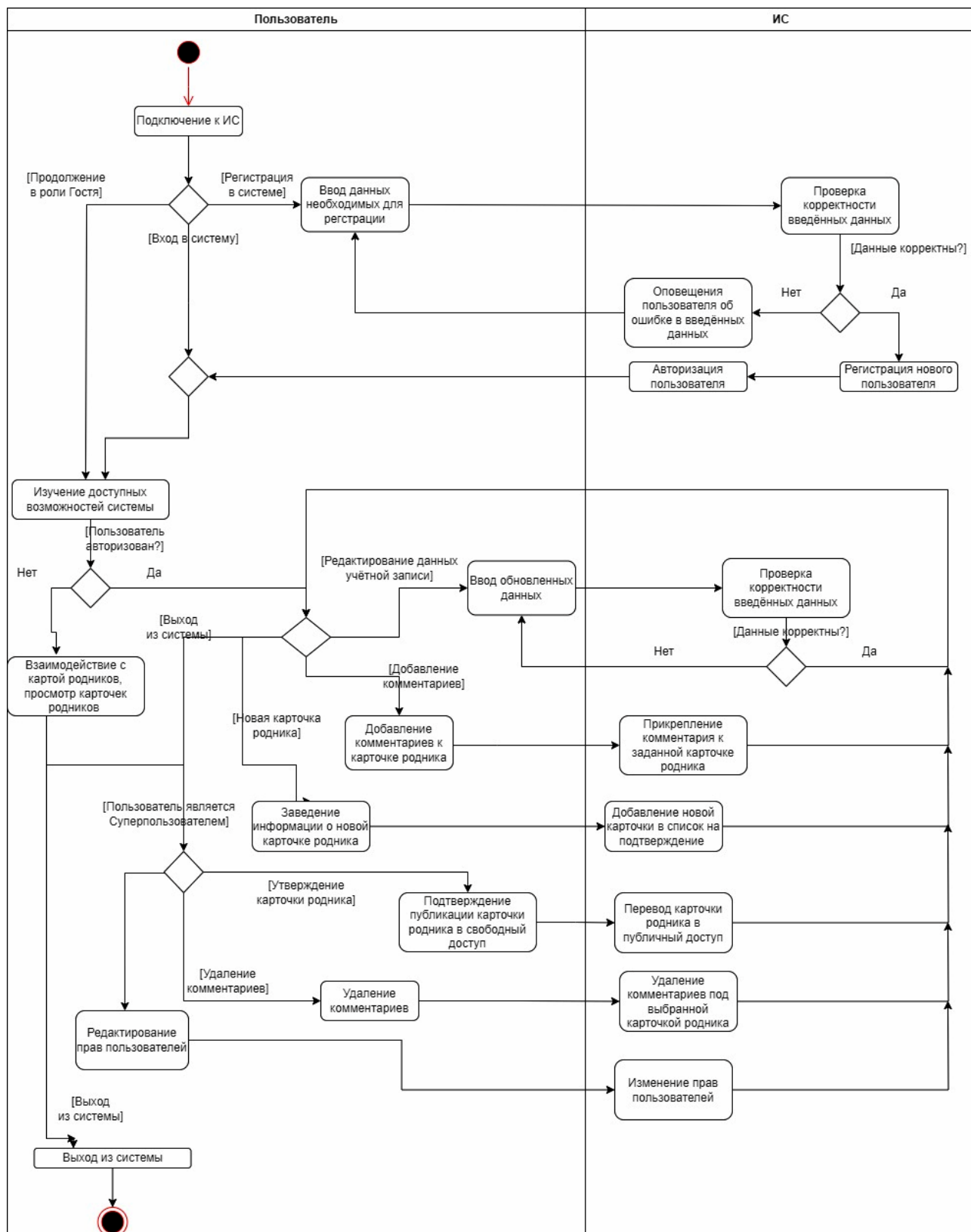


Рис. 2. Диаграмма деятельности

## Разработка информационной системы

При проектировании пользовательского интерфейса используется сервис Figma.

Предполагается, что приложение будет использоваться не только на устройствах с большими экранами, но и на мобильных устройствах. Определимся с основными экранами интерфейса и перечислим все его части:

- главная страница (титульная страница с информацией о системе);
- родники (экран с интерактивной картой родников);
- карточка отдельного родника (экран с информацией об отдельном роднике);
- авторизация (экран авторизации);
- регистрация (экран регистрации);
- личный кабинет пользователя (экран личного кабинета пользователя);
- экран добавления информации о роднике (интерфейс для добавления нового родника в систему);
- подсистема суперпользователей (интерфейсы для администраторов, где будет возможность осуществлять контроль над добавлением информации о новых родниках, модерировать комментарии, наделять правами суперпользователя).

Для примера на рис. 3 и 4 представлены интерфейс главной страницы (см. рис. 3) и интерфейс с интерактивной картой родников (см. рис. 4) для мобильного приложения.



Рис. 3. Интерфейс главной страницы

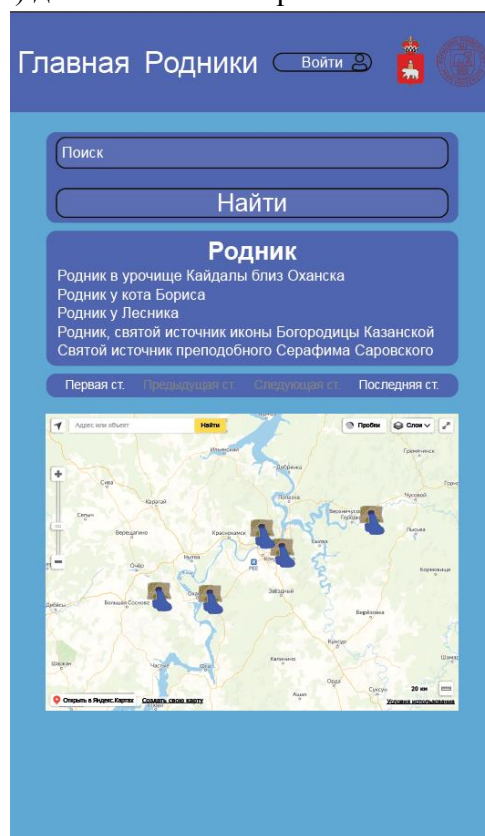


Рис. 4. Интерфейс с интерактивной картой родников

Перед началом разработки прототипа информационной системы необходимо проработать структуру базы данных, в которой будет храниться информация, необходимая для функционирования системы. Проектирование БД производилось сразу же с приведением её к третьей нормальной форме, что необходимо для исключения дублирования данных и кривых связей сущностей. Схема БД со всеми связями, а также с указанием типов данных атрибутов представлена на рис. 5.

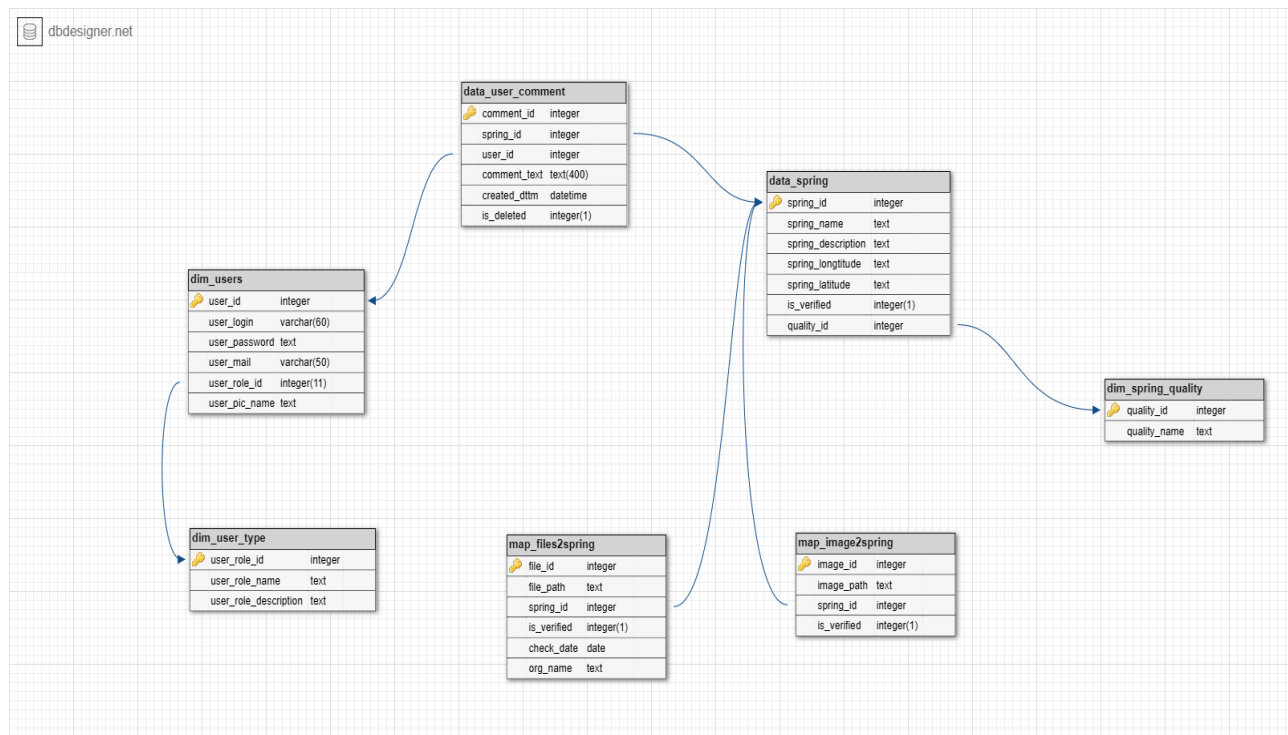


Рис. 5. Схема базы данных

Описываемая в работе информационная система разрабатывается как web-приложение для охвата максимального количества видов пользователей.

Вначале разрабатывается frontend-часть системы. Используя разработанные прототипы пользовательского интерфейса, при помощи языка разметки гипертекста HTML была реализована основа web-страниц всех элементов. Для придания соответствующего дизайна документам применяются каскадные таблицы стилей CSS. Для работы с картами использовался API Яндекс Карт [4], который в основном написан на языке JavaScript.

Параллельно разработке frontend-части системы разрабатывается её backend-часть. В качестве языка программирования серверной части используется язык PHP [5]. С помощью backend-части приложения происходит всё взаимодействие внутри сервера, а также обрабатываются обращения сервера к базе данных. Для работы с БД использовалась СУБД MySQL. В качестве инструмента для создания таблиц, написания запросов и добавления тестовых данных применялась IDE с открытым кодом DBeaver [6].

Для тестирования работоспособности прототипа информационной системы использовалась веб-сборка сервера XAMPP. Данное программное обеспечение позволяет локально запустить веб-сервер. XAMPP поддерживает весь стек технологий, выбранных для реализации данного проекта.

## Заключение

В результате проделанной работы была спроектирована и разработана геоинформационная система для хранения данных о родниках Пермского края и качестве воды в них. Про-

ведено тестирование работоспособности web-приложения на различных устройствах, включая ноутбук, настольный компьютер, планшет и смартфон. Приложение может быть использовано специалистами в области охраны природы, экологии, водоснабжения, туристами и путешественниками, а также жителями Пермского края.

### Библиографический список

1. Стратегия развития туризма в Пермском крае на период до 2035 года [Электронный ресурс]. – URL: <https://permraion.ru/ofitsialno/> (дата обращения 09.05.2023)
2. Постановление Главного государственного санитарного врача РФ от 28 января 2021 г. N 3 "Об утверждении санитарных правил и норм СанПиН 2.1.3684-21 «Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпидемических (профилактических) мероприятий» (с изменениями и дополнениями) [Электронный ресурс]. URL: <https://base.garant.ru/400289764/> (дата обращения 09.05.2023)
3. Ларман. Применение UML 2.0 и шаблонов проектирования. – 3-е изд. – М.: Вильямс, 2006. – 736 с.
4. Яндекс. Руководство разработчика // © 2023 ООО «Яндекс» [Электронный ресурс]. – URL: <https://yandex.ru/dev/maps/jsapi/doc/2.1/dg/concepts/general.html> (дата обращения 17.05.2023)
5. PHP Manual // © 2001-2023 The PHP Group. [Электронный ресурс]. – URL: <https://www.php.net/manual/en/> (Дата обращения 17.05.2023)
6. DB Beaver Community: Free Universal Database Tool [Электронный ресурс]. – URL: <https://dbeaver.io/> (Дата обращения 17.05.2023)

### References

1. Tourism development strategy in the Perm region for the period up to 2035 [Electronic resource]. – URL: <https://permraion.ru/ofitsialno/> (date of last access 09.05.2023)
2. Decree of the Chief State Sanitary Doctor of the Russian Federation of January 28, 2021 N 3 "On approval of sanitary rules and norms SanPiN 2.1.3684-21" Sanitary and epidemiological requirements for the maintenance of the territories of urban and rural settlements, for water bodies, drinking water and drinking water supply, atmospheric air, soils, residential premises, operation of industrial, public premises, organization and implementation of sanitary and anti-epidemic (preventive) measures" (with amendments and additions) [Electronic resource]. – URL: <https://base.garant.ru/400289764/> (date of last access 09.05.2023)
3. Larman. Application of UML 2.0 and design patterns. – 3rd ed. – M.: Williams, 2006. – 736 p.
4. Yandex. Developer Guide // © 2023 Yandex LLC [Electronic resource]. – URL: <https://yandex.ru/dev/maps/jsapi/doc/2.1/dg/concepts/general.html> (date of last access 17.05.2023)
5. PHP Manual // © 2001-2023 The PHP Group. [Electronic resource]. – URL: <https://www.php.net/manual/en/> (date of last access 17.05.2023)
6. DB Beaver Community: Free Universal Database Tool [Electronic resource]. – URL: <https://dbeaver.io/> (date of last access 17.05.2023)

## **DESIGN, DEVELOPMENT AND DOCUMENTATION OF THE GEOINFORMATION SYSTEM "WATER QUALITY IN THE SPRINGS IN THE PERM REGION"**

*Kuzaev Aidar F.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, kuzaevaydar@mail.ru

*Shestakov Kirill N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, zeldaone@mail.ru

The design, development and documentation of a geoinformation system for storing information about the springs in the Perm region and the quality of water in them are considered. Existing similar information systems are being studied. Based on their analysis, the main functional requirements for the designed application are formulated. It describes the reasonable choice of design and development tools, as well as the creation of a system prototype. When designing, the UML modeling language and the Diagrams.net CASE tool are used. The MySQL database management system is chosen for data storage. When developing the server part of the system, the PHP programming language is used, for the implementation of work with maps and for visualization – the JavaScript language. The prototype of the information system interface is implemented in the Figma graphic editor.

**Keywords:** information system, design, spring, water quality, Perm region.

## ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ БИЗНЕС-АНАЛИЗА

*Кузовов Дмитрий Денисович, Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, denvey2001@mail.ru

Информационная революция привела к появлению огромных объемов данных, которые создаются и обрабатываются с каждой секундой. Умение эффективно управлять данными стало ключевым преимуществом в современном мире. Системы бизнес-анализа играют важную роль в этом процессе. Они представляют собой мощный инструмент для обработки и анализа данных, позволяющий организациям принимать обоснованные решения. Однако вопросы их разработки и внедрения остаются предметом активных исследований. Доступные методики и подходы часто либо слишком общие, либо, наоборот, ориентированы на конкретные ситуации или типы организаций. Существует потребность в разработке универсальных методик и подходов, которые могли бы быть успешно применены в различных контекстах. Данная работа нацелена на проектирование и реализацию системы бизнес-анализа, способной собирать, анализировать и визуализировать большие объемы данных из различных источников. Работа включает в себя анализ, обзор существующих систем, выбор средств реализации, построение модели системы и разработку системы.

Ключевые слова: бизнес, информационная система, анализ, бизнес-анализ.

Принципы функционирования систем бизнес-анализа достаточно сложны и включают в себя множество этапов [1]. На первом этапе осуществляется сбор данных. Важно отметить, что BI системы могут использовать разнообразные источники данных. Вторым этапом является подготовка данных. Этот процесс включает в себя несколько ключевых этапов: очистка данных, трансформация данных, интеграция данных. Следующий этап, это хранение данных. Одним из основных инструментов для хранения данных является Data Warehouse. Важными особенностями хранилищ данных являются: интеграция данных, центральная база данных, инструменты доступа к данным, оптимизация для чтения [2]. Следующим и главным шагом является обработка и анализ данных. Этот процесс включает применение различных методов и техник для извлечения ценной информации из сырых данных. Наиболее распространенные подходы: статистический анализ, data mining, машинное обучение, глубокое обучение, текстовый анализ, предиктивный анализ [3]. После анализа данных следует этап визуализации. С помощью различных инструментов (таких как диаграммы, графики и т.д.) данные представляются в наглядном виде. Это позволяет специалистам быстро увидеть ключевые моменты и сделать выводы. На последнем этапе, на основе полученных данных и их анализа, принимаются бизнес-решения. Важно отметить, что все этапы тесно связаны друг с другом и взаимодействуют в рамках единого процесса. Эффективность работы системы бизнес-анализа зависит от качества выполнения каждого из этапов.

На сегодняшний день существует множество систем бизнес-анализа, каждая из которых имеет свои недостатки и преимущества. Среди наиболее известных можно выделить Microsoft Power BI, Tableau, QlikView, Yandex DataLens. Проанализировав их можно выделить общие достоинства и недостатки. К достоинствам можно отнести визуализацию данных, интеграцию с различными источниками данных, интуитивно понятный интерфейс. К недо-



статкам относится сложность в изучении и использовании, ограниченность аналитики и машинного обучения, необходимость большого количества ресурсов.

Основываясь на выводах, при проектировании собственной системы, необходимо учитывать, что система должна быть гибкой в отношении источников данных, предлагать эффективные инструменты визуализации данных, быть интуитивно понятной пользователю, а также, по возможности, включать возможности предиктивной аналитики и машинного обучения. Вместе с тем, стоит учесть и потенциальные сложности, связанные с требуемыми ресурсами.

Для реализации системы необходимо выбрать инструменты, которые обеспечат наибольшую эффективность, ее масштабируемость, а также простоту использования и поддержки.

В ходе анализа и сравнения различных методологий, CASE-средств, языков программирования и систем управления базами данных, были выбраны технологии, наиболее подходящие для системы бизнес-анализа.

В качестве методологии проектирования был выбран UML в силу своей универсальности и популярности. Его наглядность и структурированность позволяют легко представлять сложные системы и визуализировать процессы.

Для UML, было выбрано CASE-средство StarUML. Оно предлагает богатый набор функций для разработки и визуализации UML-диаграмм, включая поддержку. Его активная поддержка и обновления делают StarUML удобным и мощным инструментом для проектирования систем.

Для реализации системы на этапе разработки выбран Python. С его гибкостью, мощными библиотеками для работы с данными и высоким уровнем читаемости кода, Python отлично подходит для систем-бизнес анализа. Кроме того, Python легко интегрируется с различными системами и платформами, что упрощает работу с множеством источников данных.

В качестве системы управления базами данных был выбран PostgreSQL. Это мощная, надежная и масштабируемая система, которая поддерживает большое количество типов данных и имеет встроенные инструменты для анализа данных. Благодаря поддержке Python, PostgreSQL позволяет объединить обработку и анализ данных в одном месте, что увеличивает эффективность системы.

Логика работы системы бизнес-анализа построена с помощью диаграммы деятельности и представлена ниже (см. рис. 1).

Разобьем диаграмму на блоки для более удобного описания.

Блок 1. Вход в систему. Вход в систему инициируется пользователем. Система отображает форму авторизации. В случае, если пользователь не зарегистрирован в системе, он вводит данные для регистрации, после чего система регистрирует нового пользователя. Если пользователь уже зарегистрирован, он проходит процедуру авторизации: вводит учетные данные, которые система проверяет. При неверных данных процесс возвращается к этапу отображения формы авторизации. В случае верного ввода данных происходит вход в главное меню.

Блок 2. Роли и подключение к проекту. Система проверяет, будет ли подключение к другому проекту. Если так, происходит подключение или автоматическая авторизация на сервере. Далее на основании роли пользователя в подключенной БД система делает выбор: если роль "Администратор", пользователь проходит в следующий блок действий как администратор; если роль "Пользователь", система выдаёт соответствующую роль, и пользователь переходит к выбору дальнейших действий как пользователь. Если подключение к другому проекту не происходит, пользователь автоматически получает роль "Администратор".



помощи данного графического редактора, настройка визуального оформления всех объектов схожа с языком разметки HTML. Перенесение созданных файлов в код, было выполнено при помощи библиотеки PySide6. Также весь функционал нажатий был прописан при помощи данной библиотеки. Для реализации графиков были использованы такие библиотеки, как Matplotlib и Pandas, дающие множество типов визуализации данных. Для работы с СУБД PostgreSQL, была использована библиотека Psycopg2

В результате проведенного исследования была создана рабочая система бизнес-анализа, обеспечивающий возможность для пользователей просматривать, обрабатывать и анализировать различные данные. Целью этой системы является демонстрация потенциала и функциональных возможностей, служащих основой для дальнейшей разработки, усовершенствования и адаптации под конкретные бизнес-требования.

### **Библиографический список**

1. *Бариленко В.И.*, Основы бизнес-анализа [Текст]: учеб. пособие/ В.И.Бариленко, В.В. Бердников, Р.П. Булыга; под ред. В.И. Бариленко. – М.: Кнорус. – 2014. – 272 с.;
2. *Репин В.В.*, Бизнес-процессы. Моделирование, внедрение, управление [Текст] / В.В. Репин. – М.: Манн, Иванов и Фербер. – 2013. – 512 с.;
3. Технологии big data. – Текст: электронный // Мсс: [Электронный ресурс]. – URL: <https://mcs.mail.ru/blog/tekhnologii-big-data-kak-analiziruyut-bolshie-dannye> (дата обращения: 10.01.2023).

### **References**

1. *Barilenko V.I.*, Fundamentals of business analysis [Text]: textbook. allowance / Barylenko V.I., Berdnikov V.V., Bulyga R.P.; ed. V.I. Barylenko – M.: Knorus. – 2014. – 272 p. (In Russ.).
2. *Repin V. V.*, Business processes. Modeling, implementation, management / V.V. Repin. – M.: Mann, Ivanov and Ferber. – 2013. – 512 p. (In Russ.).
3. Big data technologies. [Electronic resource]. – URL: <https://mcs.mail.ru/blog/tekhnologii-big-data-kak-analiziruyut-bolshie-dannye> (Access 1 October 2023).

## **DESIGN AND DEVELOPMENT OF INFORMATION SYSTEM FOR BUSINESS ANALYSIS**

*Kuzovov Dmitry D., Averin Sergey I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [denvey2001@mail.ru](mailto:denvey2001@mail.ru)

The information revolution has led to the emergence of huge amounts of data that are being created and processed every second. The ability to effectively manage data has become a key advantage in today's world. Business intelligence systems play an important role in this process. They are a powerful data science and analysis tool that enables organizations to make informed decisions. However, the issues of their development and implementation remain the subject of active research. Available methodologies and approaches are often either too general or, on the contrary, focused on specific situations or types of organizations. There is a need to develop universal methodologies and approaches that can be successfully applied in various contexts. This work is aimed at designing and implementing a business intelligence system capable of collecting, analyzing and visualizing large amounts of data from various sources. The work includes analysis, review of existing systems, choice of means of implementation, building a system model and system development.

Keywords: business, information system, analysis, business analysis.

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ БЕСКЛЮЧЕВОГО ДОСТУПА ПРЕПОДАВАТЕЛЕЙ В АУДИТОРИИ УНИВЕРСИТЕТА**

*Кулаков Андрей Михайлович, Василюк Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, andrei.kul1610@gmail.com

Данная работа посвящена проектированию и документированию информационной системы бесключевого доступа преподавателей в аудитории университета. Целью проекта является создание системы, позволяющей преподавателям получить доступ к аудитории без необходимости использования физических ключей или карт доступа. В ходе работы был проведён обзор и анализ аналогов информационных систем, с выделением главных преимуществ и недостатков. По результатам анализа были выявлены требования к разрабатываемой системе, а также проведён выбор инструментальных средств проектирования. Далее было проведено проектирование модели информационной системы, описание используемых технологий, а также разработаны примеры возможного интерфейса пользователя. В итоге была спроектирована информационная система позволяющей преподавателям получить доступ в аудиторию с использованием бота в мобильном приложении Telegram [1].

Ключевые слова: бесключевой доступ, информационная система, telegram, бот, университет, аудитории.

### **Введение**

Современные технологии упрощают управление информацией и являются неотъемлемой частью современного образования. В университетах и других учебных заведениях преподаватели используют информационные системы для занятий и оценки студентов. Одна из таких технологий – бесключевой доступ в помещения, который широко применяется во многих областях. Бесключевая система позволит сократить время, затрачиваемое на поиск и использование ключей, а также повысит контроль и мониторинг доступа. Результаты работы могут быть использованы при создании подобных систем в других образовательных учреждениях и организациях.

### **Анализ и сравнение существующих систем**

Для сравнения и анализа был выбран ряд систем каждая из которой именуют системой бесключевого доступа. Рассмотрим их суть, различия, преимущества и недостатки.

Управление замком с помощью портативного устройства: используется электронный замок и пульт управления. Владелец пульта может открывать и закрывать дверь с определенного расстояния. Преимущества: использование смартфона в качестве пульта, возможность блокировки двери на большом расстоянии. Недостатки: необходимость закрывать дверь вручную, автоматическое закрытие не безопасно, утеря пульта приводит к потере доступа. Примером системы является мобильное приложение SmartAirkey [2].

Система входа и выхода без ключа: Используются радиоволны для передачи сигналов от передатчика к приемнику в замке. При попытке открыть замок, система автоматически определяет нахождения рядом передатчика и открывает дверь. Преимущества: использова-

ние смартфона в качестве передатчика, нет необходимости доставать ключ. Недостатки: при необходимости пустить людей в помещение нужно физически оставить замок открытым, потеря передатчика приводит к потере доступа. Примером также является мобильное приложение SmartAirkey [2].

Проверка доступа по биометрии в момент открытия двери: Система проверяет права доступа пользователя в режиме реального времени с использованием биометрических данных. Преимущества: высокая безопасность, отсутствие риска утери ключа. Недостатки: дороговизна и сложность массового применения, необходимость оставлять дверь открытой при запуске людей в помещение. Примером данной системы является платформа BioConnect [3].

Бесключевой доступ с помощью NFC: Используется беспроводная технология NFC для обмена данными между устройством с NFC и считывающим устройством в замке. При поднесении устройства к считывающему устройству и проверке идентификационных данных замок автоматически открывается. Преимущества: высокая безопасность, возможность использования смартфона с NFC. Недостатки: не все смартфоны поддерживают NFC, потеря устройства с NFC приводит к потере доступа. Примером доступа по NFC является мобильное приложение Nedap MACE [4].

Анализ существующих информационных систем показывает, что они не подходят для разрабатываемой информационной системы из-за неудобства использования, высокой стоимости и ограниченных функций. Однако, метод управления замком с помощью портативного устройства наиболее подходит по указанным параметрам. Требуется устранить недостатки и добавить новые функции. Для этого предлагается добавить промежуточный сервер между замком и портативным устройством, позволяющий управлять замками и иметь резервный доступ в случае происшествия или утери устройства. В качестве портативного устройства предлагается использовать смартфон для удобства. Реализацию данных условий можно осуществить с помощью бота в приложении Telegram, которое обладает рядом преимуществ, таких как популярность, доступность на различных платформах и интеграционные возможности с другими системами. В проектируемой системе необходимо реализовать задачи, такие как хранение журнала использования в базе данных, система авторизации пользователей, управление замком с сервера и управление замком со смартфона.

### **Выбор средств проектирование и разработки.**

Проектирование информационных систем осуществляется с помощью CASE-средств с объектно-ориентированным подходом, такими как MagicDraw, Astah и StarUML[5]. MagicDraw предоставляет богатый набор инструментов для создания и анализа диаграмм UML, поддерживает расширения плагинами и возможность совместной работы. Astah имеет интуитивно понятный интерфейс, позволяет генерировать код на различных языках программирования и поддерживает импорт и экспорт моделей. StarUML предоставляет бесплатную версию со многими основными функциями, поддерживает расширения плагинами и позволяет создавать различные типы диаграмм UML.

Для создания замка с бесключевым доступом одним из вариантов является использование плат Arduino[6]. Распространенными языками программирования для Arduino являются C++, Python и JavaScript. В данном случае был выбран язык Python, так как он прост в использовании, гибок, имеет обширную коллекцию готовых библиотек и инструментов, включая поддержку telegram-ботов, а также совместим с различными системами. Язык C++ слишком сложен для разработки системы бесключевого доступа и имеет меньше готовых библио-

тек, чем Python. JavaScript не подходит для разработки telegram-ботов. В качестве среды разработки будет использоваться бесплатный IDLE[7], поставляемый вместе с Python.

Для работы с данными в информационной системе подходят различные программы для работы с базами данных. Наиболее популярными являются PostgreSQL[8], MySQL и Oracle Database, которые являются реляционными системами управления базами данных. PostgreSQL обладает преимуществами в виде бесплатности, широкого функционала, высокой надежности и безопасности. MySQL отличается простотой использования и хорошей производительностью для простых запросов. Oracle Database предлагает мощные возможности и высокий уровень надежности и безопасности, но имеет сложность использования и требует больших ресурсов. После анализа выбран PostgreSQL как наиболее подходящая СУБД, учитывая недостатки других вариантов.

Таким образом, выбранный набор инструментов для проектирования информационной системы включает StarUML, Python, IDLE и PostgreSQL.

### Проектирование информационной системы

Доступ к ИС преподаватели будут осуществлять через бот в приложении telegram, администратор же будет работать через стационарное устройство. в процессе моделирования была создана диаграмма прецедентов (рис. 1). Диаграмма прецедентов – это графическое представление взаимодействия между пользователями и системой.

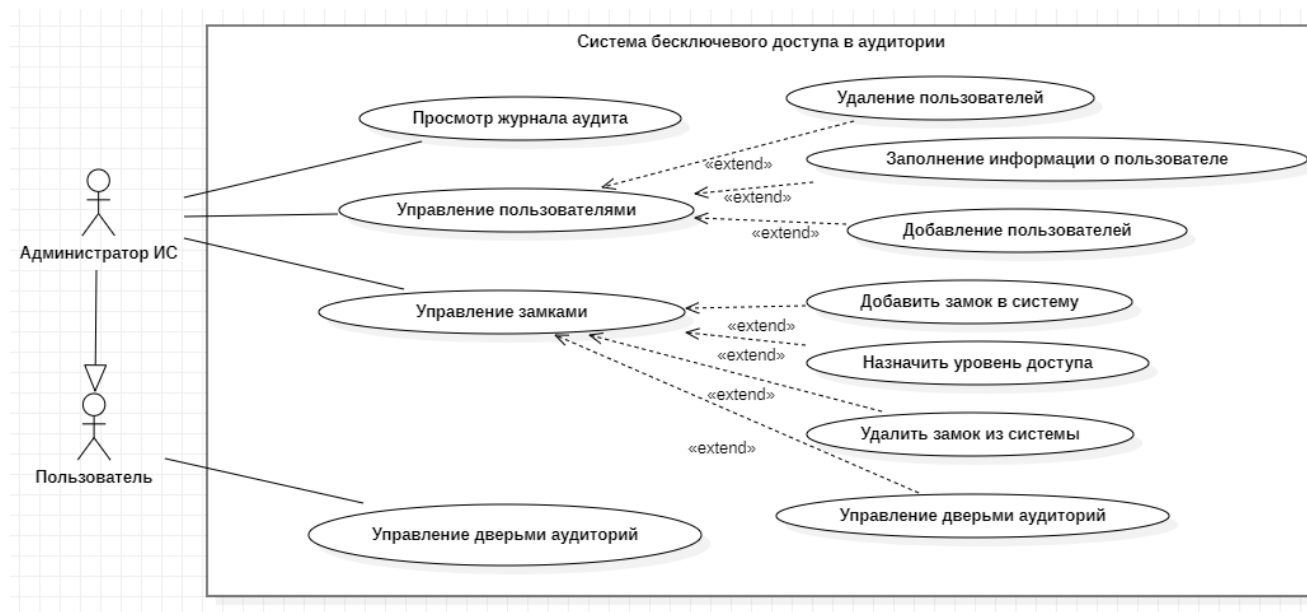


Рис. 1. Диаграмма прецедентов

Актер "Пользователь" – это уже авторизованный пользователь, он имеет только один прецедент "Управление дверьми аудиторий" который подразумевает возможность средствами системы открывать и закрывать аудитории, другой функционал этому актеру не доступен. Актер "Администратор ИС" имеет намного больше доступных действий. Прецедент "Просмотр журнала аудита" даёт возможность ему просмотреть записи журнала аудита которые содержат информацию об использовании пользователями системы бесключевого доступа. Прецедент "Управление пользователями" даёт ему право добавлять новых пользователей, записывая их логин, пароль, ФИО и уровень доступа в базу данных, удалить их и изменять записи уже добавленных пользователей. Прецедент "Управление замками" позволяет администратору добавлять замок в систему, назначить ему уровень доступа к нему, удалить его, а

также позволяет как актеру “Пользователь” открывать и закрывать их. Даная диаграмма полностью описывает функционал системы, исключая необходимость в дополнительных диаграммах прецедентов.

Диаграмма классов отображает классы, их атрибуты, методы и связи между классами. Эта диаграмма помогает понять структуру системы, идентифицировать классы и взаимодействия между ними. В диаграмме были созданы и описаны четыре класса: администратор, пользователь, замок и запись в журнале аудита. получившаяся диаграмма представлена на рисунке (рис. 2).



Рис. 2. Диаграмма классов

Класс “Замок” хранит данные о замке на двери, содержит поля “Номер двери”, “Уровень доступа” (необходимый уровень доступа, который должен быть у пользователя, что бы он мог открыть дверь).

Класс “Запись аудита” хранит данные которые заносятся в журнал аудита. Содержит поля “Номер записи”, “Время”, “Номер двери”, “Логин управляющего” (логин пользователя, который управлял замком).

Класс “Пользователь” хранит данные самого пользователя (преподавателя университета) и доступные ему функции. Содержит поля “Логин”, “Пароль”, “ФИО”, “Факультет”, “Уровень доступа” (числовое значение, отвечающее за возможность пользователя открывать аудитории разного уровня доступа). Возможность пользователя открывать и закрывать описана в методах “Открыть аудиторию.” и “Закрыть аудиторию”.

Класс “Администратор” является производным классом от родительского класса “Пользователь” и содержит как поля и методы родителя, так и свои собственные методы. Методы данного класса описывают возможности администратора, такие как: добавление и

удаление пользователей, изменение информации о нём, добавление и удаление замков из системы, изменения уровня доступа к ним, а также просмотр журнала аудита.

### **Заключение**

Была разработана информационная система бесключевого доступа преподавателей в аудитории университета. Система упрощает работу преподавателей и сотрудников университета, исключая необходимость поиска и ношения ключей. Она может применяться в различных учреждениях и сооружениях для контроля доступа. Например, офисные помещения, гостиничные номера, медицинские учреждения и спортивные комплексы.

### **Библиографический список**

1. Telegram // telegram.org [Электронный ресурс]. – URL: <https://telegram.org/faq> (дата обращения: 10.05.2023)
2. Бесплатное мобильное приложение SmartAirkey для работы с системой бесключевого доступа // securityrussia.com [Электронный ресурс]. – URL: <https://securityrussia.com/skud/soft-skud/smartairkey-mobile> (дата обращения: 10.05.2023)
3. Интеграционная платформа Suprema BioConnect // securityrussia.com [Электронный ресурс]. – URL: <https://securityrussia.com/skud/soft-skud/bioconnect#desc> (дата обращения: 10.05.2023)
4. Бесплатное приложение для смартфона Nedap MACE App Android // skud-system.ru [Электронный ресурс]. – URL: <https://skud-system.ru/mace-app-android> (дата обращения: 10.05.2023)
5. StarUML // staruml.io [Электронный ресурс]. – URL: <https://staruml.io/> (дата обращения: 7.05.2023)
6. Arduino // www.arduino.cc [Электронный ресурс]. – URL: <https://www.arduino.cc/> (дата обращения: 7.05.2023)
7. IDLE // docs.python.org [Электронный ресурс]. – URL: <https://docs.python.org/3/library/idle.html> (дата обращения: 7.05.2023)
8. PostgreSQL // www.postgresql.org [Электронный ресурс]. – URL: <https://www.postgresql.org/> (дата обращения: 7.05.2023)

### **References**

1. Telegram. [Electronic resource]. – URL: <https://telegram.org/faq>
2. Besplatnoe mobil'noe prilozhenie SmartAirkey dlya raboty s sistemoj besklyuchevogo dostupa. [Electronic resource]. – URL: <https://securityrussia.com/skud/soft-skud/smartairkey-mobile> (In Russ)
3. Integracionnaya platforma Suprema BioConnect. [Electronic resource]. – URL: <https://securityrussia.com/skud/soft-skud/bioconnect#desc> (In Russ)
4. Besplatnoe prilozhenie dlya smartfona Nedap MACE App Android. [Electronic resource]. – URL: <https://skud-system.ru/mace-app-android> (In Russ)
5. StarUML. [Electronic resource]. – URL: <https://staruml.io/>
6. Arduino. [Electronic resource]. – URL: <https://www.arduino.cc/>
7. IDLE. [Electronic resource]. – URL: <https://docs.python.org/3/library/idle.html>
8. PostgreSQL. [Electronic resource]. – URL: <https://www.postgresql.org/>



## **AN INFORMATION SYSTEM DESIGN AND DOCUMENTATION FOR TEACHERS KEYLESS ACCESS TO UNIVERSITY CLASSROOMS**

*Kulakov Andrei M., Vasiluk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, andrei.kul1610@gmail.com

This work is devoted to the design and documentation of the information system of keyless access of teachers in the university auditorium. The aim of the project is to create a system that allows teachers to access the classroom without the need to use physical keys or access cards. In the course of the work, a review and analysis of analogs of information systems was carried out, highlighting the main advantages and disadvantages. Based on the results of the analysis, the requirements for the system being developed were identified, as well as the selection of design tools was carried out. Next, the information system model was designed, the technologies used were described, and examples of a possible user interface were developed. As a result, an information system was designed that allows teachers to gain access to the classroom using a bot in the Telegram mobile application [1].

Keywords: keyless access, information system, telegram, bot, university, auditor

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ ПОДГОТОВКИ АНАЛИТИКОВ

*Ладыгин Владислав Андреевич<sup>1</sup>, Ладыгин Владимир Андреевич<sup>2</sup>, Иванов Илья Борисович<sup>3</sup>,  
Степанов Владимир Анатольевич<sup>4</sup>*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, itx.19.1.mex.mat@gmail.com<sup>1</sup>, ladawowa@mail.ru<sup>2</sup>,  
PyaIvanov2019@yandex.ru<sup>3</sup>, svaperm@gmail.com<sup>4</sup>

В статье рассмотрены технологии, используемые в процессе проектирования “система подготовки аналитика”, анализ аналитиков, анализ навыков аналитиков, диаграммы проектирования системы, разработан прототип интерфейса системы.

Ключевые слова: информационная система, анализ, аналитик, диаграмма, проектирование, навыки, знания, умения, среда разработки, методологии моделирования, среда базы данных, прототип интерфейса системы.

В современном обществе информация играет ключевую роль в развитии мира. Она становится основой для принятия решений и формирования индивидуального и коллективного будущего. Для того чтобы избежать замешательства и не потеряться среди множества информационных потоков, люди активно разрабатывают и совершенствуют инструменты управления информацией. В связи с растущими объемами данных возникла важная профессия – аналитик, чья задача заключается в эффективной обработке и анализе больших объемов информации.

Текущая значимость разработки информационной системы для проверки аналитических навыков человека обусловлена несколькими факторами. В высших учебных заведениях наблюдается растущая тенденция к компьютеризации и переходу к электронному документированию. Быстрый и удобный доступ к данным, а также их эффективная обработка становятся неотъемлемыми требованиями. Кроме того, важным фактором является возможность доступа к ресурсу из любой точки, где есть интернет.

Цель данной курсовой работы проектирование и документирование приложения для подготовки аналитиков.

Исходя из цели работы, выделим следующие задачи, которые необходимо решить в ходе проектирования:

- 1) Провести сравнительный анализ между аналитиками;
- 2) Провести анализ навыков, умений, знаний и качеств аналитиков;
- 3) Провести анализ существующих инструментов для работы с аналитикой;
- 4) Выявить и определить требования к разрабатываемой информационной системе проверки знаний аналитиков;
- 5) Определить набор инструментальных средств в соответствии с требованиями;
- 6) Спроектировать логическую и функциональную модель ИС;

## **Анализ аналитиков**

Современные процессы работы требуют включения аналитических навыков у множества специалистов, даже если аналитика не является их основной занятостью. Данная закономерность происходит потому, что аналитика стала неотъемлемой составляющей для многих рабочих процессов.

Рассмотрены некоторые специальности, которые работают с аналитикой: аналитик, инженер в области обработки данных и анализа, бизнес-аналитик, специалист по работе с большими данными (DATA SCIENTISTS), специалист по статистике, квант, специалист по экономическому анализу или финансовой аналитике, специалист по визуализации данных.

Разобраны подробнее виды аналитиков их особенности: бизнес-аналитик, системный аналитик, аналитик данных (или Data Analyst), маркетолог-аналитик, финансовый аналитик, продуктовый аналитик.

Рассмотрено, что является особенностью деятельности каждого из этих аналитиков:

- 1) Системный аналитик и аналитик данных лучше разбираются в программировании, инструментах BI, организации системы хранения данных.
- 2) Маркетолог аналитик занимается оптимизацией маркетинговой компанией, экономикой продаж, использованием инструментов BI.
- 3) Финансовый аналитик использует финансовые инструменты, разбирается в кредитах и займах, инвестициях, условиях финансирования.
- 4) Продуктовый аналитик работают с конкретными продуктами, используют метрики и инструменты для анализа работы эффективности продуктов.
- 5) Бизнес аналитик погружаются в создание и изменение процессов, экономику, исследования, финансы. Используют BI, работают с отчетностью и визуализацией данных.

Рассмотрены основные инструменты аналитиков:

- 1) Работа с электронными таблицами. (Excel или Google.Таблица, LibreOfficeCalc).
- 2) Работа с базами данных (SQL или MySQL, PostgreSQL).
- 3) Программирование (Python, R, SAS, SPSS).
- 4) Визуализация данных (Power BI, Google Charts, Datawrapper, Yandex DataLens и другие BI ситемы).

Рассмотрены области в которых работают аналитики [15]: управление требованиями, исследования и анализ данных, моделирование процессов, проектирование ИС, внедрение, консалтинг.

## **Навыки, и качества специалиста аналитика**

В ходе работы выделено пять групп ключевых навыков аналитиков: бизнес, математика / анализ операций, машинное обучение / большие данные, программирование, статистика.

А также выделено четыре кластера ролей на основе этих навыков: предприниматели, исследователи, разработчики, творческие специалисты.

Проведено сопоставление группы ключевых навыков и аналитические роли для демонстрации профиля специалиста и соответствия его навыкам. Представлены эти профили на рис. 1:

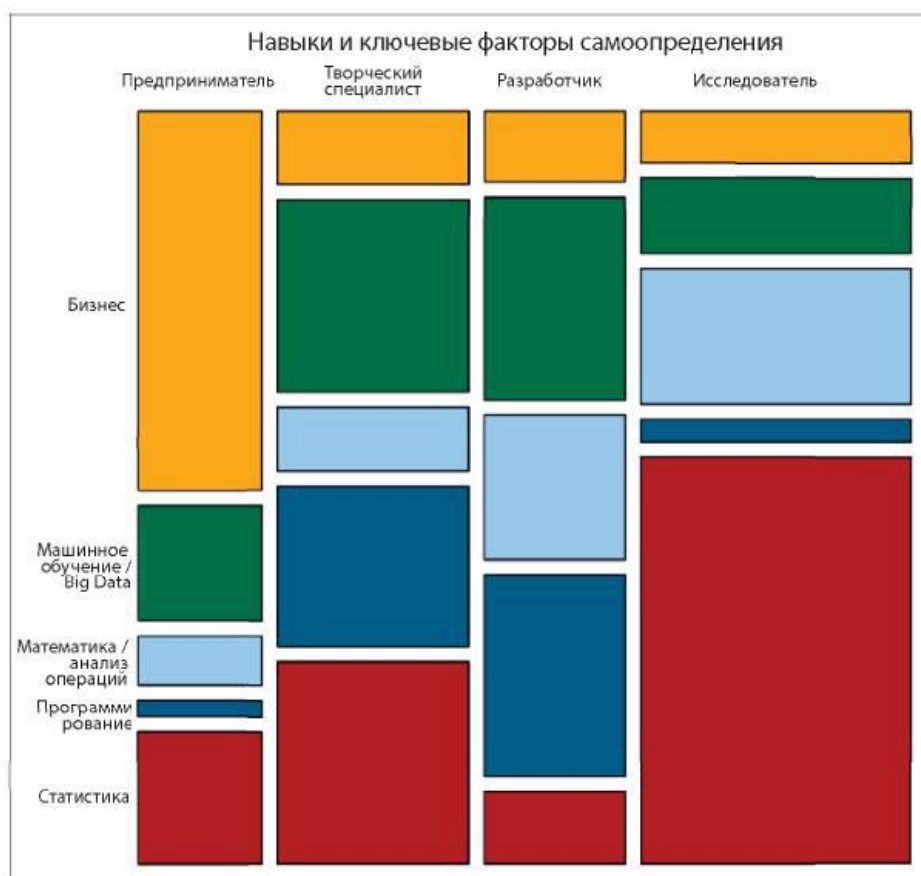


Рис. 1. Сопоставление группы ключевых навыков и аналитические роли

Разобраны психологические и личностные качества, которые присущи хорошему специалисту аналитику: аналитический склад ума, внимание к деталям, методичность, рациональный скептицизм, уверенность в себе, любопытство, терпение, любовь к данным, стремление учиться, деловой подход.

Разобраны, какие социальные навыки помогают в работе аналитиков: умение слушать, умение опрашивать и задавать вопросы, способность соображать на ходу, навыки анализа, навыки системного мышления, навыки обучения, навыки создания комфортных условий общения, лидерские качества, умение наблюдать, навыки общения, организационные навыки, навыки моделирования, навыки межличностного общения, творческий подход.

Выделены Hard skills, они подразделяются на: базовые компетенции, теория и техники анализа, основы бизнеса, знания по разработке информационных систем, документирование, дополнительные компетенции, иностранные языки. Выделены Soft skills, они подразделяются на: базовые компетенции, аналитические навыки, личностные навыки, межличностные навыки, деловые навыки.

Сопоставлены Hard skills, Soft skills и области работы аналитиков и рассмотрены на рис. 2 и 3, как они взаимодействуют между собой:

	Управление требованиями	Анализ данных	Моделирование БП	Проектирование ИС	Внедрение ПО	Консалтинг
1. Базовые компетенции	✓	✓	✓	✓	✓	✓
2. Теории и техники анализа	✓	✓	✓	✓	✓	
3. Основы бизнеса		✓	✓			✓
4. Разработка ИС			✓	✓		✓
5. Документирование	✓	✓	✓	✓	✓	✓
6. Доп. компетенции		✓			✓	✓
7. Иностранные языки	?	?	?	?	?	?

Рис. 2. Навыки Hard skills и области работы аналитиков

	Управление требованиями	Анализ данных	Моделирование БП	Проектирование ИС	Внедрение ПО	Консалтинг
1. Базовые компетенции	✓	✓	✓	✓	✓	✓
2. Аналитические навыки	✓	✓	✓	✓		✓
3. Личностные навыки	✓	✓	✓	✓		✓
4. Межличностные навыки	✓				✓	✓
5. Деловые навыки		✓	✓		✓	✓

Рис. 3. Навыки Soft skills и области работы аналитиков

### Требования к проектируемой системе

На основе анализа существующих системы выбраны следующие функциональные требования, которые должны осуществляться разрабатываемой ИС:

- 1) Система должна предлагать пользователю зарегистрироваться в системе.;
- 2) Система должна предлагать пользователю авторизоваться в системе по логину и паролю;
- 3) Система должна предлагать пользователю пройти систему для определения его навыков и знаний;
- 4) Система должна определять по навыкам и знаниям специалиста его цифровой образ как аналитика;
- 5) Система должна предлагать образовательные программы и курсы для повышения квалификации специалиста;
- 6) Система должна сохранять цифровой образ, результаты и рекомендации, полученные пользователем при прохождении системы;
- 7) Система должна обезопасить данные пользователей;
- 8) Система должна сохранять действия пользователя в системе;

9) Система должна разграничивать права и обязанности пользователя и администратора системы.

### Анализ инструментов проектирования системы

Сделана сравнительная характеристика инструментов моделирования в табл. 1.

Таблица 1. Сравнительная характеристика инструментов моделирования

Приложение / критерий	Бесплатная программа	Поддержка Windows	Интуитивно понятный интерфейс
StarUML	+	+	+
Violet UML Editor	+	+	+
Diagram Designer	+	+	+

Выбран инструмент моделирования StarUML.

Сделана сравнительная характеристика сред разработки в табл. 2.

Таблица 2. Сравнительная характеристика сред разработки

Критерий / среда разработки	Visual Studio	Qt Creator	Code::Blocks
Поддержка Windows	+	+	+
Поддержание объектно-ориентированных языков программирования	+	+	+
Поддержание Визуального редактора	+	+	+
Имеет удобный интерфейс	+	+	+
Имеет подробную документацию	+	+	+
Имеет бесплатную версию	+	+	+

Выбрана среда разработки Visual Studio.

Сделана сравнительная характеристика баз данных в табл. 3.

Таблица 3. Сравнительная характеристика баз данных

Критерий / среда разработки	MySQL	PostgreSQL	Firebird	Oracle
Бесплатная версия	+	+	+	+
Удобный интерфейс	+	+	+	+
Универсальность платформы	+	+	-	-
Высокий уровень безопасности	+	+	+	+
Подробная документация	+	+	+	+
Соответствие ACID	+	+	+	+

Выбрана база данных MySQL.

### Проектирование функциональной и логической модели

Общая диаграмма классов представлена на рис. 4.

Общая диаграмма развертывания представлена на рис. 5.

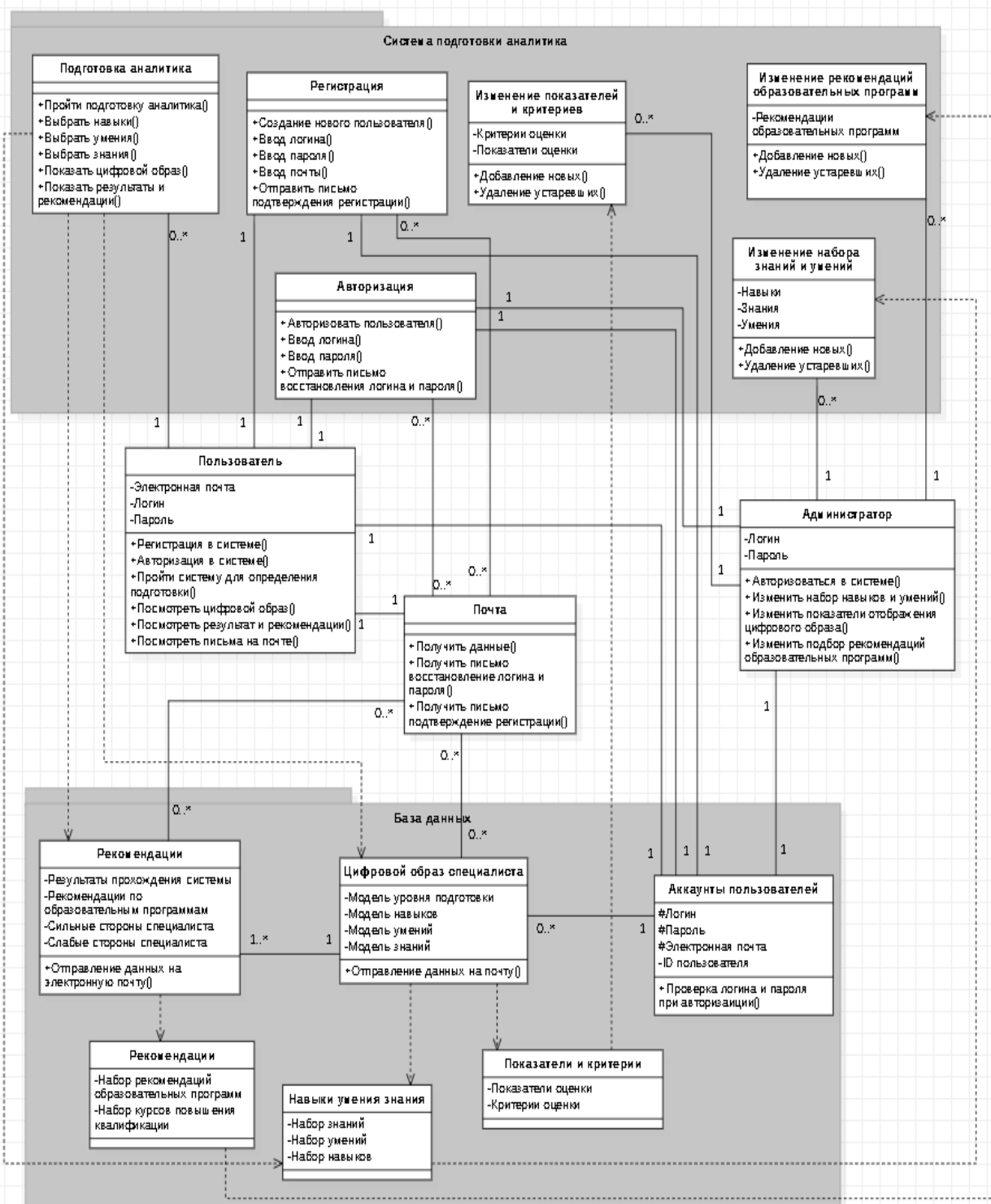


Рис. 4. Общая диаграмма классов

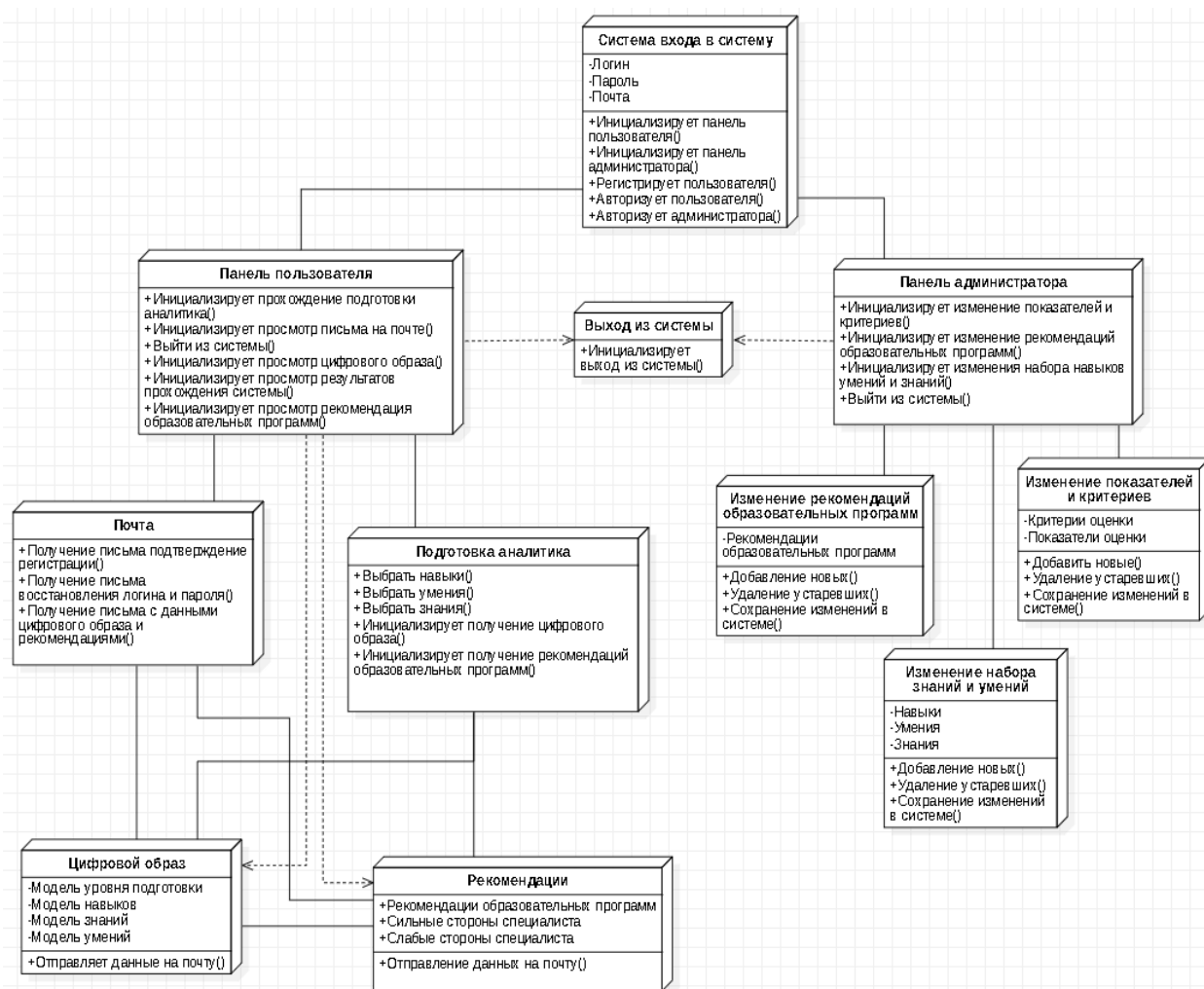


Рис. 5. Общая диаграмма развертывания

### Библиографический список

1. *Андерсон, Карл* Аналитическая культура. От сбора данных до бизнес-результатов / Карл Андерсон; пер. с англ. Юлии Константиновой; [науч. ред. Руслан Салахиев]. – М.: Манн, Иванов и Фербер, 2017. – 336 с. ISBN 978-5-00100-781-4.
2. Как освоить профессию аналитика данных [Электронный ресурс]. – URL: <https://storage2.bothelp.io/arnionschool/7f/7feb/7febb316439e8b620c005ffaa7ad5c42/Analitik-dannyh-Samoe-vagnoe.pdf> (Дата обращения 5.05.2023).
3. Полезные навыки аналитиков. Как стать профессионалом [Электронный ресурс]. – URL: <https://habr.com/ru/companies/sqalab/articles/219667/> (Дата обращения 5.05.2023).

### References

1. *Anderson, Carl* Analytic culture. From data collection to business results / Carl Anderson; per. from English. Yulia Konstantinova; [scient. ed. Ruslan Salakhiev]. – M.: Mann, Ivanov and Ferber, 2017. – 336 p. ISBN 978-5-00100-781-4
2. How to master the profession of a data analyst [Electronic resource]. – URL: <https://storage2.bothelp.io/arnionschool/7f/7feb/7febb316439e8b620c005ffaa7ad5c42/Analitik-dannyh-Samoe-vagnoe.pdf> (Accessed 05/05/2023);
3. Useful skills of analysts. How to become a professional [Electronic resource]. – URL: <https://habr.com/ru/companies/sqalab/articles/219667/> (Accessed 05/05/2023).



## **DESIGN AND DOCUMENTATION OF AN AUTOMATED SYSTEM FOR ANALYST TRAINING**

*Ladygin Vladislav A.<sup>1</sup>, Ladygin Vladimir A.<sup>2</sup>, Ivanov Ilya B.<sup>3</sup>, Stepanov Vladimir A.<sup>4</sup>*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, itx.19.1.mex.mat@gmail.com<sup>1</sup>,  
ladawowa@mail.ru<sup>2</sup>, [IlyaIvanov2019@yandex.ru](mailto:IlyaIvanov2019@yandex.ru)<sup>3</sup>, svaperm@gmail.com<sup>4</sup>

The article discusses the technologies used in the design process “analyst training system”, analyst analysis, analyst skills analysis, system design diagrams, developed a prototype of the system interface.

Keywords: information system, analysis, analyst, diagram, design, skills, knowledge, skills, development environment, modeling methodologies, database environment, system interface prototype.

## ПРОЕКТИРОВАНИЕ АЛГОРИТМА ПОИСКА ПУТИ В КОМПЬЮТЕРНЫХ ИГРАХ

*Лаптев Илья Сергеевич, Кузаев Айдар Файзуллович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, ilya.laptev.9090@mail.ru

Рассматриваются популярные алгоритмы поиска пути в компьютерных играх. Проанализированы алгоритмы поиска пути для компьютерных игр Дейкстры, A\*, Беллмана-Форда, D\* Lite, Theta\*, а также возможные алгоритмы для слияния. Описываются ключевые особенности, достоинства и недостатки этих алгоритмов. В результате проведенного анализа выбирается и реализуется алгоритм поиска пути A\*. На основе этого выбора проектируется тестовая информационная система – демо-сцена на движке Unity, которая включает в себя импровизированную карту и пользовательский интерфейс. Для нее формулируются описания логики в виде UML-диаграмм. Используемый в работе алгоритм может быть использован в дальнейшем в качестве инструмента для искусственного интеллекта в разработке компьютерной игры.

Ключевые слова: алгоритм, A\*, Unity, разработка игр, искусственный интеллект, поиск пути, компьютерный игры.

### Введение

В настоящее время компьютерные игры стали неотъемлемой частью повседневной жизни многих людей. Они предлагают возможность отвлечься от повседневной рутины и развивать различные навыки и умения, такие как командная работа, логическое мышление, а также улучшение реакции и принятия решений в динамичных ситуациях.

С развитием технологий и возросшей сложностью игровых миров возникает потребность в создании реалистичных и интеллектуальных персонажей, способных самостоятельно передвигаться по игровой среде. Одной из ключевых задач в разработке компьютерных игр является проектирование эффективных алгоритмов поиска пути [1] для игровых персонажей. Эти алгоритмы позволяют персонажам в игре перемещаться по виртуальному миру, избегать препятствий и достигать поставленных целей.

Однако, с ростом сложности игровых сценариев и увеличением количества объектов в игре, поиск оптимального пути становится более сложной задачей. Подобные проблемы возникают в играх различных жанров, от аркадных игр до ролевых и стратегий. Недостаточно просто перемещать персонажа от точки А к точке Б, необходимо учесть множество факторов, таких как препятствия, местность, противники и другие игровые объекты.

Для обеспечения удовлетворительного игрового опыта, важно, чтобы алгоритмы поиска пути были быстрыми и точными [2], не замедляя игровой процесс и не вызывая раздражения у игроков. Игровые разработчики стремятся создать персонажей с реалистичным поведением и интеллектом, способных принимать обоснованные решения и адаптироваться к изменяющимся условиям игры.

Целью данной работы является разработка эффективного алгоритма поиска путей для компьютерных игр в целом и дальнейшей интеграции в конкретную игру, в частности.

## Обзор существующих алгоритмов поиска пути

Были выделены несколько используемых алгоритмов для поиска пути в играх на данный момент:

- алгоритм Дейкстры: популярный и известный алгоритм поиска кратчайшего пути в взвешенном графе. Работает с положительными весами ребер;
- алгоритм A\*: модификация алгоритма Дейкстры для поиска кратчайшего пути в графе. Работает с взвешенным графом и может обрабатывать отрицательные веса ребер. Использует эвристику для прогнозирования расстояния до целевой вершины[3];
- алгоритм Беллмана-Форда: используется для поиска кратчайшего пути в графе с отрицательными весами ребер и отрицательными циклами. Работает на основе динамического программирования;
- алгоритм D\* Lite: модификация алгоритма D\*, предназначенная для поиска кратчайшего пути в динамических графах. Эффективно обрабатывает изменения в графе;
- алгоритм Theta\*: модификация алгоритма A\* для поиска кратчайшего пути в графах с препятствиями. Использует дополнительные проверки для оптимизации поиска.

Для достижения поставленной цели автором исследования был выбран алгоритм A\* в качестве основы исследования. A\* является одним из самых популярных алгоритмов поиска пути в компьютерных играх благодаря своей эффективности и точности. В данном исследовании было предусмотрено исследование различных аспектов этого алгоритма, включая его принципы работы, структуру данных и стратегии поиска.

## Проектирование информационной системы

Одним из важных аспектов данного исследования является разработка информационной системы, которая будет поддерживать работу алгоритма A\*. Для моделирования поведения и внутренней структуры этой системы был использован язык моделирования общего назначения UML [4]. Представленные в данном исследовании диаграммы UML позволяют лучше понять взаимодействие информационной системы с пользователями и провести анализ ее функциональности и эффективности в контексте алгоритма A\*.

В первую очередь было необходимо сформулировать описание будущей реализации алгоритма в виде диаграммы с использованием средства проектирования StarUML [5]. Для описания работы алгоритма была выбрана диаграмма классов, которая позволила описать структуру системы и взаимодействие между ее компонентами. Диаграмма классов представила конкретные методы реализации алгоритма A\* и связи между ними.

Исходя из проведенного исследования, автор смог разработать информационную систему, основанную на алгоритме A\*, и представил ее в виде диаграммы классов (см. рис. 1).

Для реализации алгоритма поиска пути A\* необходимо провести детальное проектирование системы с использованием диаграммы классов. Диаграмма классов поможет описать структуру системы, методы реализации алгоритма A\* и взаимодействие между компонентами.

Для реализации алгоритма A\* используются классы «User», «Map», «Grid» и «AStar». Класс «User» хранит начальную и конечную точки пути. Класс «Map» представляет карту сеткой клеток, а класс «Grid» описывает ячейки этой сетки. Класс «AStar» реализует сам алгоритм A\* и использует классы «Map» и «Grid».

По завершении работы алгоритма, можно использовать метод "getPath()" класса "AStar" для получения найденного пути.

Проектируемый алгоритм будет реализован на языке C# [6] и движке Unity [7] и показан через демо-сцену. Автор подготовил демо-сцену с картой шестиугольных ячеек, где указаны координаты и есть меню управления. В меню есть элементы "Checkbox" для включения ячейки, установки начальной и конечной точек, установки препятствий и проходимых ячеек. Доступны кнопки "AUTO" для автоматической генерации карты, "START" для запуска алгоритма, и "REFRESH" для сброса карты. Управление камерой осуществляется клавишами "WASD", а масштабирование карты – мышью.

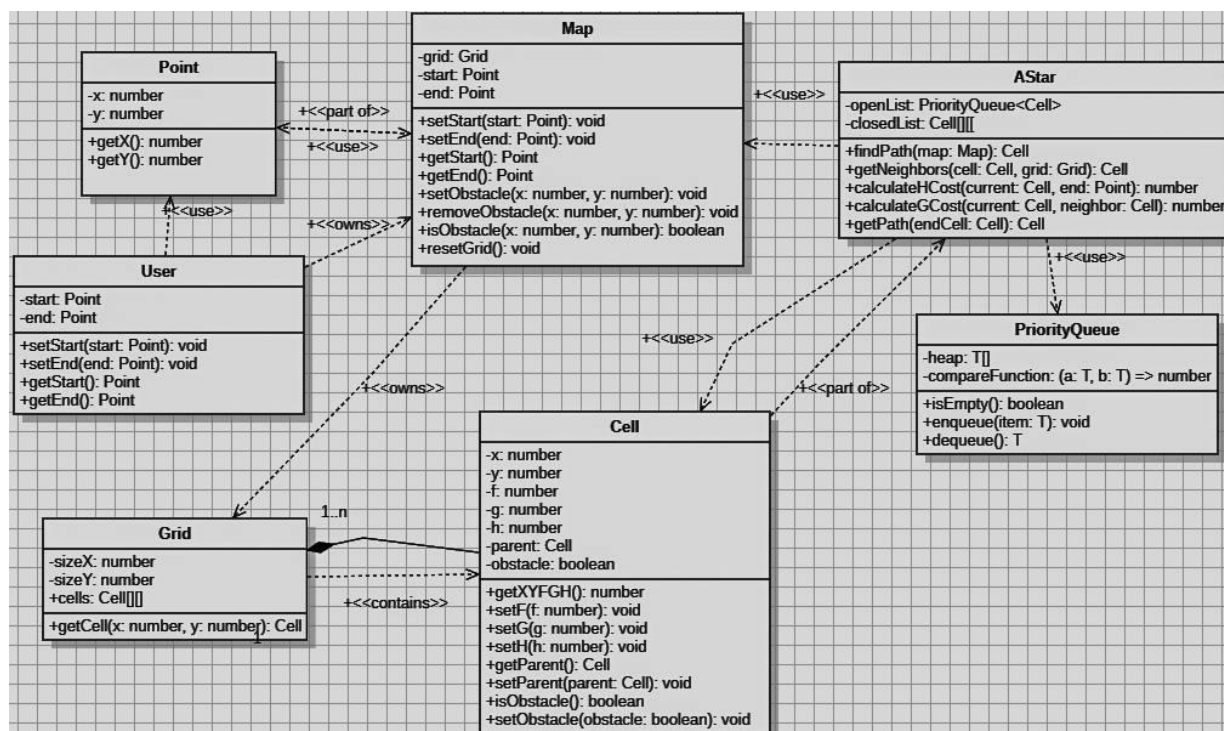


Рис. 1. Диаграмма классов алгоритма A\*

Алгоритм сканирует окружающие клетки, чтобы найти правильный путь[8]. Клетки красного цвета отображают возможные пути, но не выбраны как оптимальный маршрут из-за их высокого "веса" и неправильного направления. Клетки черного цвета отображают правильно выбранный и эффективный путь от точки старта до точки конца. Стоит отметить, что алгоритм поиска пути умеет учитывать вес ячеек и выбирать пути не только с минимальным весом. При дальнейшем развитии этого алгоритма в игре можно предусмотреть различные типы ячеек с разным весом.

Присутствие препятствий на карте помогает показать точность и эффективность работы алгоритма A\*. Кратчайший путь успешно обходит запрещенные клетки. Также необходимо добавить, что алгоритм позволяет создавать разнообразные сценарии прохождения, где игрок или искусственный интеллект будут учитывать не только препятствия, но и выбирать оптимальные маршруты с учетом условий и особенностей окружающей среды. Таким образом, игровой процесс становится более интересным и стратегическим. В данном случае существуют разные варианты продолжения пути до конечной точки, что позволяет выбрать наиболее кратчайший и оптимальный маршрут. Мы также можем наблюдать высокую эффективность алгоритма (см. рис. 2). Вначале он двигается по более длинной траектории, но как только появляется возможная клетка, через которую можно пройти, снижая тем самым преодолеваемое расстояние, алгоритм мгновенно перестраивает маршрут, чтобы он стал

кратчайшим до цели. Это демонстрирует гибкость и адаптивность алгоритма, который способен быстро изменить путь, чтобы достичь цели максимально эффективным способом.

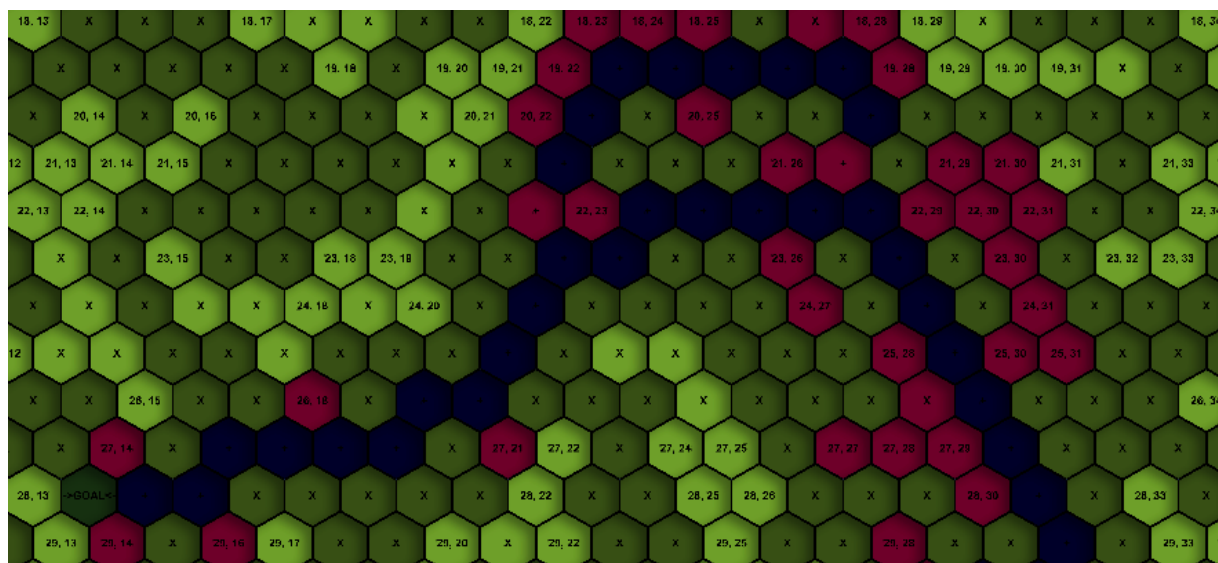


Рис. 2. Пример работы алгоритма

Использованный алгоритм поиска пути в компьютерных играх  $A^*$  был успешно реализован и протестирован. Он был выбран как оптимальный по сложности реализации, использования и производительности. Рекомендуется использовать данный алгоритм или его интерпретации для решения задач по поиску пути в компьютерных играх.

### Заключение

В результате проделанной работы был спроектирован и реализован эффективный алгоритм поиска путей для компьютерных игр[9], который позволяет искусственному интеллекту перемещаться по игровому миру и достигать поставленных целей, учитывая все возможные ситуации в игре. Для достижения данной цели рассмотрены ключевые понятия и определения, связанные с поиском путей в компьютерных играх.

### Библиографический список

1. Шатилов А., Минитаева А. Сравнительный анализ алгоритмов поиска пути. // Технологии инженерных и информационных систем. -2021. – № 2. – С.3-23.
2. Саблин А. Постановка и анализ задачи поиска пути между двумя точками при разработке игр. // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем. -2019. – С.295-298.
3. Ежов Г.В. Алгоритмы нахождения пути, их сравнение и визуализация на базе движка Unity / Г. В. Ежов, М. В. Димитренко. – Текст: непосредственный // Молодой ученый. – 2020. – № 50 (340). – С. 15-24.
4. CASE tool for diagrams, software design & analysis // Software ideas modeler [Электронный ресурс]. – URL: <https://www.softwareideas.net/en/about> (дата обращения 03.05.2022)
5. StarUML Documentation // StarUML, 2014-2019 [Электронный ресурс]. – URL: <https://docs.staruml.io/> (дата обращения: 03.05.2022)
6. C# Documentation // Learn Microsoft [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата обращения 03.05.2023)

7. Unity Engine // Unity [Электронный ресурс]. – URL: <https://unity.com/ru> (дата обращения 03.05.2023)
8. Наглядное описание алгоритма A\* // Red Blob Games [Электронный ресурс]. – URL: <https://www.redblobgames.com/pathfinding/a-star/introduction.html> (дата обращения 20.04.2023)
9. ResearchGate // ResearchGateNet [Электронный ресурс]. – URL: [https://www.researchgate.net/publication/2495826\\_Steering\\_Behaviors\\_For\\_Autonomous\\_Characters](https://www.researchgate.net/publication/2495826_Steering_Behaviors_For_Autonomous_Characters) (дата обращения 03.05.2023)

## References

1. *Shatilov A., Minitaeva A.* Comparative analysis of pathfinding algorithms. *Tekhnologii inzhenernyh i informacionnyh sistem = Technologies of engineering and information systems*. 2021; (2):3-23.
2. *Sablin A.* Statement and analysis of the problem of finding a path between two points in game development. *Informacionno-telekommunikacionnye tekhnologii i matematicheskoe modelirovanie vysokotekhnologichnyh sistem = Information and telecommunication technologies and mathematical modeling of high-tech systems*. 2019.:295-298.
3. *Ezhov, G.V.* Path finding algorithms, their comparison and engine-based visualization. *Molodoy ucheniy = Young scientist*. 2020;50(340):15-24.
4. CASE tool for diagrams, software design & analysis. [Electronic resource]. – URL: <https://www.softwareideas.net/en/about>
5. StarUML. [Electronic resource]. – URL: <https://docs.staruml.io/>
6. C# Documentation. [Electronic resource]. – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>
7. Unity Engine. [Electronic resource]. – URL: <https://unity.com/ru>
8. Visual description of the algorithm A\*. [Electronic resource]. – URL: <https://www.redblobgames.com/pathfinding/a-star/introduction.html>
9. ResearchGate. [Electronic resource]. – URL: [https://www.researchgate.net/publication/2495826\\_Steering\\_Behaviors\\_For\\_Autonomous\\_Characters](https://www.researchgate.net/publication/2495826_Steering_Behaviors_For_Autonomous_Characters)

## DESIGNING A PATH SEARCH ALGORITHM IN COMPUTER GAMES

*Laptev Ilya S., Kuzaev Aydar F.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [ilya.laptev.9090@mail.ru](mailto:ilya.laptev.9090@mail.ru)

In this article, popular pathfinding algorithms in computer games were considered. Pathfinding algorithms for the computer games of Dijkstra, A\*, Bellman-Ford, D\* Lite, Theta\*, as well as possible algorithms for merging, were chosen. The main features, advantages and disadvantages of the selected algorithms are considered. The pathfinding algorithm A\* was selected, designed, reviewed and implemented. Based on this choice, a test information system was designed – a demo scene on the Unity engine, which includes an impromptu map and user interface. For it, descriptions of logic were formulated in the form of UML diagrams. This algorithm can be further used as a tool for artificial intelligence in the development of a computer game.

Keywords: algorithm, A\*, Unity, game development, artificial intelligence, pathfinding, computer games.

## **АВТОМАТИЗАЦИИ ДОКУМЕНТООБРАЗОВАНИЯ НА ПРЕДПРИЯТИИ ПО РЕМОНТУ ЭЛЕКТРОДВИГАТЕЛЕЙ**

*Лязгин Даниил Сергеевич, Ракина Валерия Денисовна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, lyazg10@mail.ru

В данной статье рассматривается проектирование и документирование информационной системы автоматизации документообразования на предприятии по ремонту электродвигателей. Проект системы представляет из себя десктопное приложение способное значительно упростить ведение документооборота путем автоматизации заполнения шаблона документа и систематизацией требуемых для этого данных. В работе представлены результаты анализа предметной области, требований компании, выбора инструментов проектирования и разработки. Также приводятся результаты проектирования системы и прототипы базы данных и графического пользовательского интерфейса.

Ключевые слова: автоматизация, документообразование, средства заполнения шаблонов.

### **Введение**

На предприятии по ремонту электродвигателей ООО «ЭНЕРГОСФЕРА» для оценки пригодности к эксплуатации электродвигателя проводятся измерения и испытания, в которых он проверяется на соответствие требованиям СТО 34.01-23.1-001-2017 «Объем и нормы испытаний электрооборудования». И по результатам заключения о соответствии заданным требованиям выявляется пригодность электродвигателя к дальнейшей эксплуатации.

Написание, систематизация и обработка протоколов проведения испытаний занимают значительное количество времени и увеличивают затраты организации на управление. Актуальным решением этой проблемы будет информационная система, способная как автоматизировать процесс документообразования, так и предоставить в любой удобный момент быстрый доступ к важной информации для эффективного управления бизнес-процессами на предприятии.

В качестве объекта данной работы можно выделить процесс документообразования на предприятии по ремонту электродвигателей, а в качестве предмета автоматизацию данного процесса.

Цель работы – спроектировать и задокументировать информационную систему для автоматизации документообразования на предприятии по ремонту электродвигателей.

### **Анализ требований предприятия и структуры отчета**

ООО «ЭНЕРГОСФЕРА» требуется оптимальное по соотношению стоимости к качеству решение для автоматизации создания протоколов проведения испытаний. Также информационная система должна быть простой в освоении и обслуживании.

По своей структуре документ отчета представляет из себя протокол измерений и испытаний электродвигателя и состоит из шести пунктов:

- 1) наименование электродвигателя и возбудителя;
- 2) таблица основных характеристик электродвигателя и таблица основных характеристик возбудителя;

- 3) дата проведения работ;
- 4) таблица с перечнем средств испытаний и измерений;
- 5) шесть однотипных таблиц с результатами испытаний и измерений (на рис. 1 представлена одна из них. Зеленым цветом выделены вводимые данные, желтым – данные, хранящиеся в базе данных, а красным – выводимые);

#### 5.1. Измерение сопротивления изоляции:

№	Наименование испытания (измерения)	Сопротивление изоляции, МОм						Допустимое значение, не менее	Заключение
		между фазами			относительно корпуса				
		U-VW	V-WU	W-UV	U	V	W		
1	Обмоток статора	3800	3800	3500	3500	3100	3050	60 МОм	Соответствует
	Коэффициент абсорбции	2,2	2,1	2,0	2,3	1,9	2,0	1,3	
2	Обмотка ротора	200						0,2 МОм	Соответствует
3	Обмотка якоря	1000						1,0	Соответствует
4	Обмотка индуктора	1000						1,0	Соответствует

**Рис. 1. Измерение электрической прочности изоляции**

6) заключение о проведенных испытаниях и таблица с подписью ответственного за испытания инженера.

Чтобы получить заключение о соответствии электродвигателя требованиям СТО при составлении протокола необходимо знать допустимые значения для каждой испытуемой машины, чтобы сверять с ними полученные при испытаниях данные. Отчет в готовом виде представляет из себя электронный документ в формате .doc.

### Анализ аналогичных решений

Современный рынок систем автоматизации документооборота и документообразования можно разделить на системы электронного документооборота (СЭД) и электронные системы мониторинга (ЕСМ). Здесь будут рассматриваться СЭД решения, так как функционал ЕСМ продуктов избыточен для достижения поставленной цели.

Рассмотрим пять наиболее высоко оцененных продуктов, по версии рейтинга [1] «Market.CNews» [2].

ELMA365 ЕСМ [3] – это продукт, позволяющий организовать электронный документооборот на предприятии, не привязываясь к конкретной документации. По сути своей ближе у ЕСМ системам, но при этом высоко востребована в области СЭД решений.

СБИС [4] – облачная система документооборота с контрагентами и госорганами.

Directum [5] – представляет из себя систему автоматизации документооборота, включающую в себя модули для создания, редактирования, утверждения и учета документов. Продукт позволяет организовать работу с электронными документами и мониторить процесс из обработки.

СЭД Тезис [6] – программа для автоматизации документации и оптимизации процессов. Применяется не только в частных предприятиях, но и в государственных и муниципальных учреждениях.

LanDocs [7] – это платформа для управления ресурсом предприятия. В виду наличия электронного архива вся документация хранится в одном месте. Имеется возможность корпоративного поиска по реквизитам.

Все вышеперечисленные системы хоть и решают поставленные задачам, но при этом стоят больших средств, имеют избыточный функционал и, как следствие, требуют неоправ-



данно много времени и усилий для интеграции в работу предприятия. Проектируемая система будет отличаться менее обширным функционалом, но при этом способностью эффективнее решать поставленную задачу и большей простотой в эксплуатации и установке.

### **Основные черты проектируемой системы**

На основании требований к отчету, его структуры, а также рассмотренных аналогов был выделен ряд функциональных и нефункциональных требований к системе.

Основными функциональными требованиями к информационной системе автоматизации документообразования на данном предприятии являются:

- возможность записи в информационную систему модели электродвигателя и соответствующие ему табличные значения испытываемых параметров;
- хранение информации о табличных значениях испытываемых параметров для проверки соответствия с ними результатов полученных при проведении испытаний;
- автоматическая проверка и вывод соответствия или несоответствия результатов полученных при проведении испытаний с табличными значениями, хранимыми в информационной системе;
- возможность записи средств проведения испытаний, их тип и его заводской номер;
- хранение информации о наименовании средств испытаний, их типе и номере зав;
- автоматизированное формирование заключения о проведении испытаний
- автоматизированное формирование электронного документа в формате .doc;
- возможность хранения предыдущих отчетов и дополнительной информации о них.

Нефункциональные требования к проектируемой системе включают в себя обеспечение безопасности данных, высокую производительность и удобный пользовательский интерфейс. Также при проектировании информационной системы была учтена возможность ее дальнейшей масштабируемости и интеграции с другими информационными системами, используемыми на предприятии.

В протоколе можно выделить три вида переменных:

- вводимые каждый раз при создании документа, примером являются данные, полученные при испытаниях;
- вводимые единожды, но с возможностью изменения, например, основные характеристики электродвигателя или допустимых значений;
- выводимые при создании документа, например, заключения о результатах проверок, полученных при испытании данных.

Также стоит отметить и то, что программа планируется использоваться одним пользователем при поддержке системного администратора ответственного за все информационные системы на предприятии.

### **Средства для проектирования и разработки**

Для проектирования информационной системы выбрано CASE-средство StarUML [8], так как оно обладает продвинутым функционалом и строго придерживается UML спецификации. Для проектирования базы данных выбрана СУБД PostgreSQL [9], потому что она обладает широким функционалом. C# [10] выбран в качестве основного языка программирования, потому как поддерживает библиотеки для работы с Word документами и PostgreSQL. В качестве среды разработки выбрана Microsoft Visual Studio [11], поскольку предоставляет наиболее полную интеграцию с C#. Для проектирования прототипа графического пользовательского интерфейса были выбраны Microsoft Visual Studio в связке с Krypton Toolkit [12],

так как предоставляет обширный функционал для создания эстетически выверенных оконных форм.

### Проект системы

С помощью диаграммы прецедентов были смоделированы возможные варианты взаимодействия пользователя с информационной системой. Диаграмма прецедентов представлена на рис. 2 и 3.

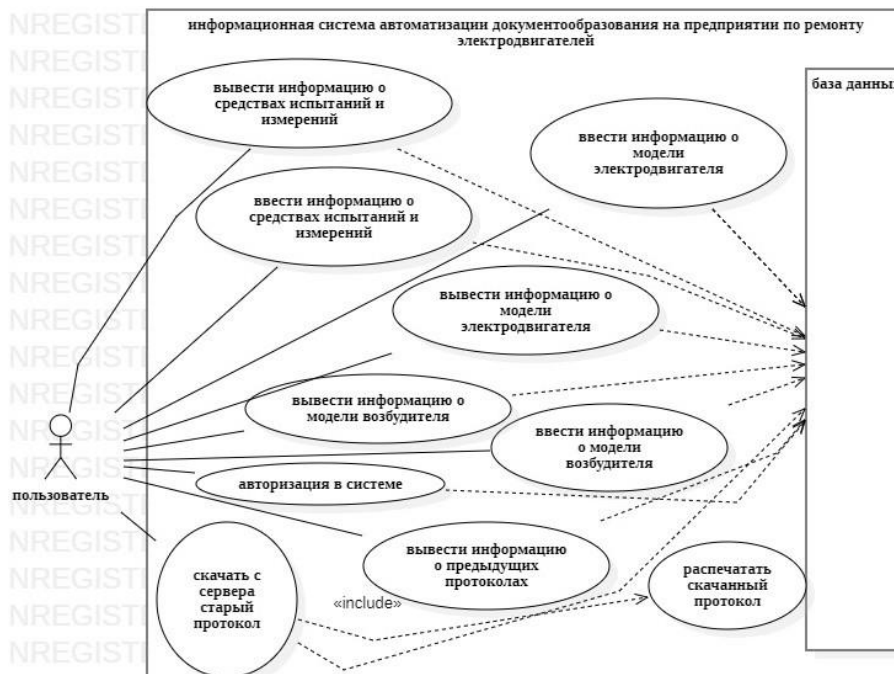


Рис. 2. Диаграмма прецедентов, часть первая

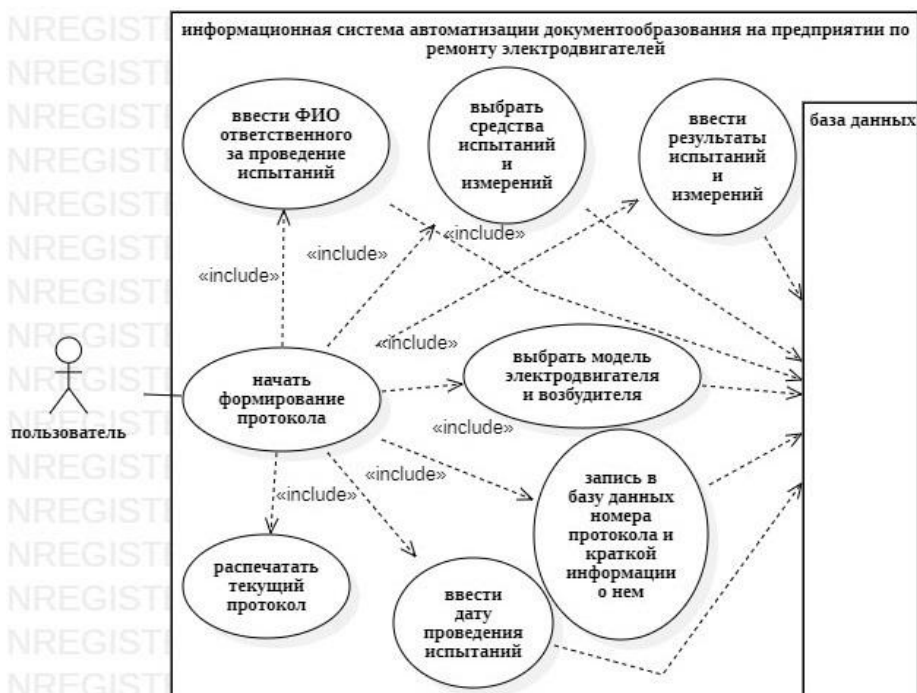


Рис. 3. Диаграмма прецедентов, часть вторая

Пользователь может авторизоваться в системе, ввести и вывести информацию об электродвигателе и возбудителе, а также о средствах испытаний. Кроме того, пользователь может инициализировать запуск создания протокола и во время создания отчета выбрать мо-

дель электродвигателя, возбудителя и средства испытаний и затем ввести дату проведения испытаний, данные, полученные при испытаниях и своё ФИО. Также у пользователя есть возможность распечатать только что созданный протокол, загрузить и распечатать резервную копию старого протокола.

После авторизации пользователю становится доступна возможность генерации протокола, этот процесс подробно рассмотрен на диаграмме деятельности на рис. 4. Кроме того, пользователь может вывести или ввести интересующую его информацию, как это описано ранее. При всех взаимодействиях с системой происходит и взаимодействие с базой данных.

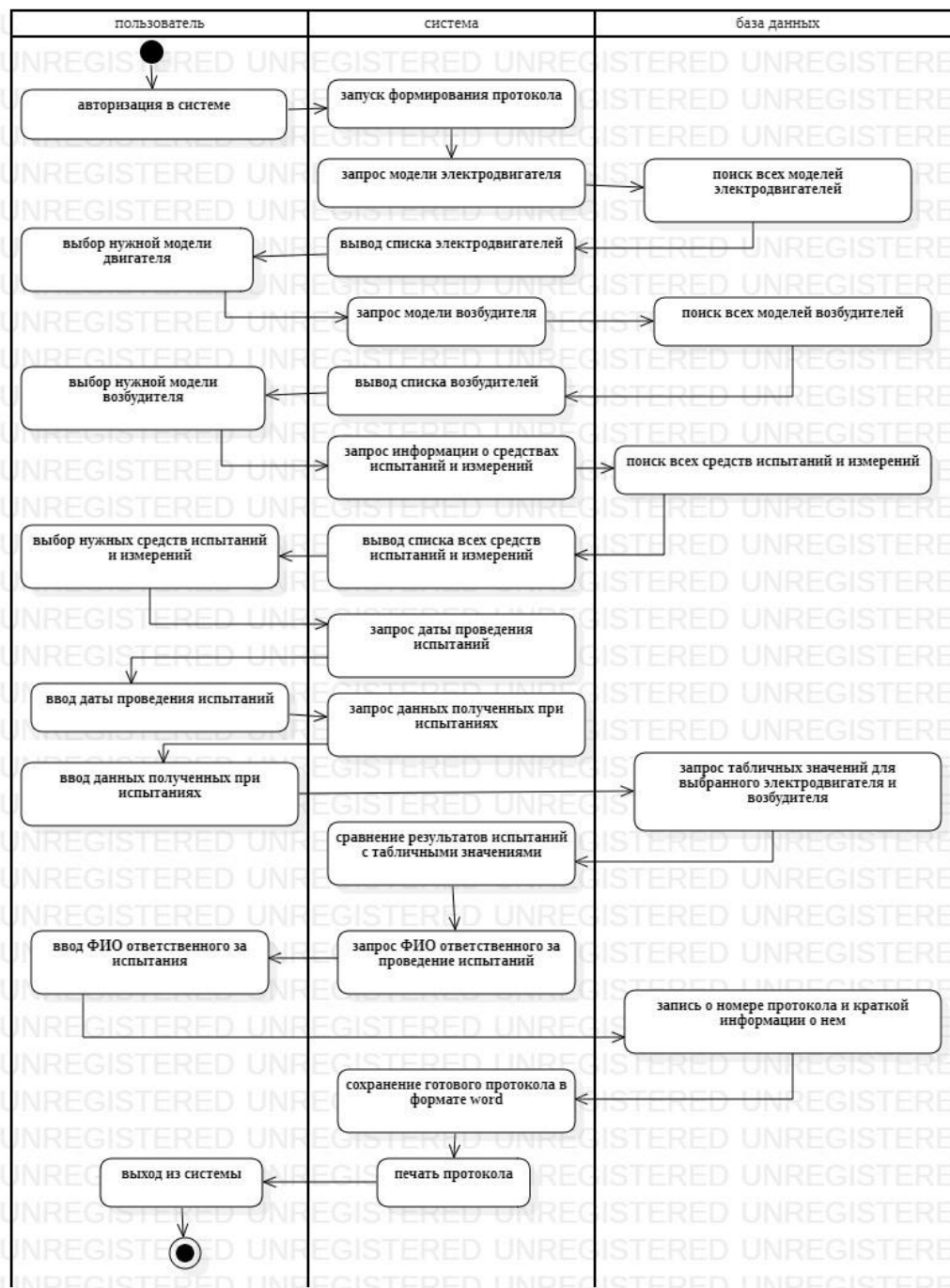


Рис. 4. Диаграмма деятельности заполнения документа

База данных будет хранить в себе информацию о каждом электродвигателе, возбудителе и средстве испытаний, а также контрольное значение для каждого теста и общую инфор-

мацию о каждом протоколе. Каждую из этих сущностей следует выделить в отдельную таблицу и определить их параметры. Модель базы данных представлена на рис. 5.

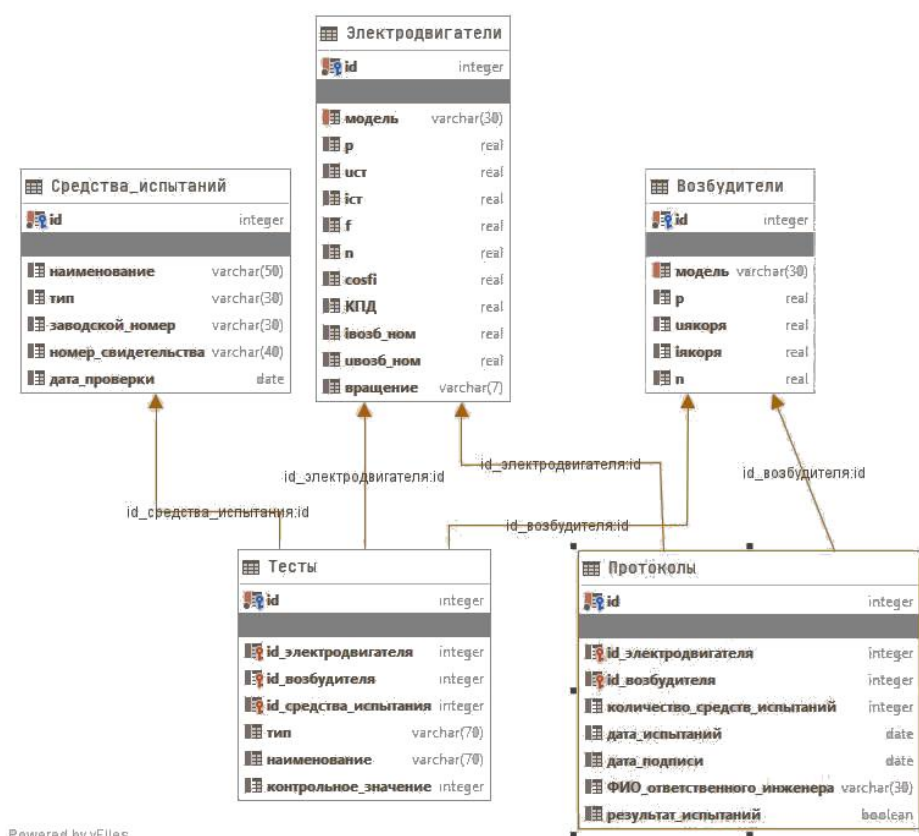


Рис. 5. Модель базы данных

Контрольное значение определяется средством испытаний, моделью возбудителя и электродвигателя. Связь таблиц средств испытаний, возбудителей и электродвигателей к таблице контрольное значение является один ко многим. Так, для одного средства испытаний может существовать множество тестов с контрольными суммами, также все это верно для возбудителей и электродвигателей.

Общая информация определяется моделью возбудителя и электродвигателя. Связь таблиц возбудителей и электродвигателей к таблице общая информация о протоколе является один ко многим. Так, для одной модели электродвигателя может существовать множество протоколов, также все это верно для возбудителей и электродвигателей.

Дизайн графического интерфейса был спроектирован с учётом современных тенденций, а именно в плоском стиле. Это подразумевает минималистичность, простоту используемых форм и монотонные цвета. Основными цветами интерфейса были выбраны синий и белый для чувства целостности при использовании системы и Microsoft word.

После авторизации пользователя встретит главное меню приложения. Главное меню состоит из семи опций для пользователя, описанных ранее. Главное меню представлено на рис. 6.

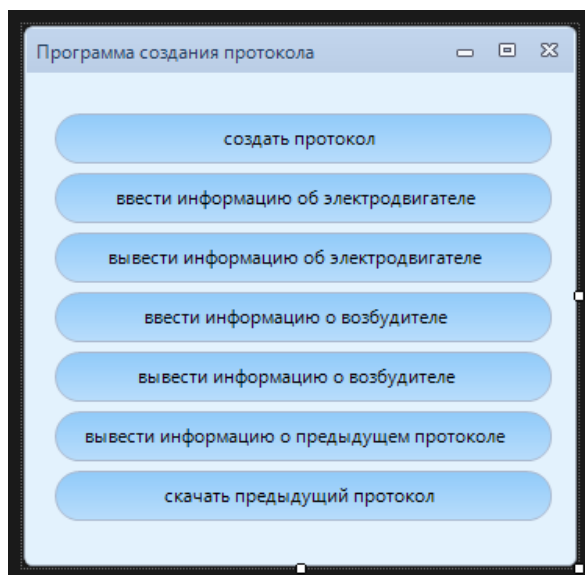


Рис. 6. Главное меню приложения

Создание протокола можно разделить на две стадии. Первая представляет из себя ввод основных данных, а вторая ввод данных, полученных при испытаниях. Для каждой из этих стадий есть отдельное окно в приложении. В первом окне выбирается модель электродвигателя и возбудителя, а также средства испытаний. Кроме того, в первом окне требуется ввести дату проведения испытаний и ФИО пользователя. Пример окна первой стадии создания протокола представлен на рис. 7. На второй стадии система подбирает подходящие тесты и требует ввести данные, полученные при испытаниях. Пример окна второй стадии создания протокола представлен на рис. 8.

Рис. 7. Первый этап создания протокола



Рис. 8. Второй этап создания протокола

### Библиографический список

1. *Перова М.В., Мартынова Д.В., Колодина А.А.* Анализ результатов рейтинга российского рынка СЭД от Market.cnews // Вызовы современности и стратегии развития общества в условиях новой реальности Москва, 21 февраля 2022 года. – Ростов-на-Дону: ИП Овчинников Михаил Артурович (Типография Алеф) (Махачкала), 2022. – С. 139-145.
2. Рейтинг Систем Электронного Документооборота 2021 [Электронный ресурс]. – URL: [https://market.cnews.ru/research/sed\\_2021/table](https://market.cnews.ru/research/sed_2021/table) (дата обращения: 11.02.23).
3. Единое цифровое пространство для корпоративного документооборота ELMA365 ECM [Электронный ресурс]. – URL: <https://elma365.com/ru/products/ecm/> (дата обращения: 11.02.23).
4. Электронный документооборот сбис [Электронный ресурс]. – URL: <https://sbis.ru/edo> (дата обращения: 11.02.23).
5. Цифровые решения для управления процессами и документами Directum [Электронный ресурс]. – URL: <https://www.directum.ru/> (дата обращения: 11.02.23).
6. СЭД ТЕЗИС – лидер на российском рынке тезис [Электронный ресурс]. – URL: <https://www.tezis-doc.ru> (дата обращения: 11.02.23).
7. LanDocs 3.8.1: больше, чем СЭД. ЛАНИТ представил новую версию платформы LanDocs [Электронный ресурс]. – URL: [https://landocs.ru/sed\\_landocs\\_381](https://landocs.ru/sed_landocs_381) (дата обращения: 11.02.23).
8. A sophisticated software modeler for agile and concise modeling StarUML [Электронный ресурс]. – URL: <https://staruml.io/> (дата обращения: 13.02.23).
9. PostgreSQL: The World's Most Advanced Open Source Relational Database PostgreSQL [Электронный ресурс]. – URL: <https://www.postgresql.org/> (дата обращения: 13.02.23).
10. C# documentation Microsoft [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата обращения: 13.02.23).
11. GitHub Copilot and Visual Studio 2022 Microsoft [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/> (дата обращения: 13.02.23).
12. Krypton-Suite [Электронный ресурс]. – URL: <https://github.com/Krypton-Suite/Standard-Toolkit> (дата обращения: 13.02.23).

## References

1. *Perova M.V., Martynova D.V., Kolodina A.A.* Analiz rezultatov reitinga rossiskogo rynka SED ot Market.cnews // Vyzovy sovremennosti i strategii razvitiya obshchestva v usloviyah novoy realnosi Moscow, 21 February 2022 year. – Rostov-on-Don: IP Ovchinnikov Mihail Arturovich (Typographiya Alef) (Mahachkala), 2022. – P. 139-145. (In Russ.).
2. Reiting System Electronnogo Dokumentooborota 2021 [Electronic resource]. – URL: [https://market.cnews.ru/research/sed\\_2021/table](https://market.cnews.ru/research/sed_2021/table) (date of application: 11.02.23). (In Russ.).
3. edinoye cifrovoye prostranstvo dlya korporativnogo dokumentooborota ELMA365 ECM [Electronic resource]. – URL: <https://elma365.com/ru/products/ecm/> (date of application: 11.02.23). (In Russ.).
4. Electronnuy dokumentooborot sbis [Electronic resource]. – URL: <https://sbis.ru/edo> (date of application: 11.02.23). (In Russ.).
5. Cifrovuye resheniya dlya upravleniya processami i documentami Directum [Electronic resource]. – URL: <https://www.directum.ru/> (date of application: 11.02.23). (In Russ.).
6. SED TEZIS – lider na rossiiskom rynke tezis [Electronic resource]. – URL: <https://www.tezis-doc.ru> (date of application: 11.02.23). (In Russ.).
7. LanDocs 3.8.1: bolshe, chem SED. LANIT predstavil novuyu versiu platformy LanDocs [Electronic resource]. – URL: [https://landocs.ru/sed\\_landocs\\_381](https://landocs.ru/sed_landocs_381) (date of application: 11.02.23). (In Russ.).
8. A sophisticated software modeler for agile and concise modeling StarUML [Electronic resource]. – URL: <https://staruml.io/> (date of application: 13.02.23).
9. PostgreSQL: The World's Most Advanced Open Source Relational Database PostgreSQL [Electronic resource]. – URL: <https://www.postgresql.org/> (date of application: 13.02.23).
10. C# documentation Microsoft [Electronic resource]. – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (date of application: 13.02.23).
11. GitHub Copilot and Visual Studio 2022 Microsoft [Electronic resource]. – URL: <https://visualstudio.microsoft.com/> (date of application: 13.02.23).
12. Krypton-Suite [Electronic resource]. – URL: <https://github.com/Krypton-Suite/Standard-Toolkit> (date of application: 13.02.23).

## AUTOMATION OF DOCUMENT GENERATION AT AN ELECTRIC MOTORS REPAIRING ENTERPRISE

*Lyazgin Daniil S., Rakina Valeria D.*

Perm State University, 15, Bukireva St., Perm, 614990, Russia, lyazg10@mail.ru

This paper considers design and documentation of the information system for the automation of document creation at an electric motors repairing enterprise. The system project is a desktop application that can significantly simplify document management by automating the filling of a document template and systematizing the data required for this. The paper presents the results of the analysis of the subject area, the requirements of the company, the choice of tools for design and development. The results of the system design and prototypes of the database and graphical user interface are also presented.

Keywords: automation, document generation, tools for document templates filling

## **ПРЕПОДАВАТЕЛИ ПЕРМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА 1916–2023: ПРОЕКТ ПРОСОПОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

*Маткин Михаил Алексеевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, matkinpmi@gmail.ru

*Аверин Сергей Игоревич*

Пермский национальный исследовательский политехнический университет, 614000, Россия,  
г. Пермь, Комсомольский проспект, д. 29, averin-s-i@ya.ru

Работа содержит анализ просопографического подхода к исследованиям историко-ориентированных информационных систем, описание подходов к проектированию информационных систем. В работе дается процесс проектирования и документирования информационной системы «Преподаватели пермского университета 1916-2023». Определён предмет и объект исследования, сформулирована актуальность разработки информационной системы. Поставлены задачи, которые необходимо решить в ходе проектирования. Определены функциональные возможности, которыми должна обладать разрабатываемая информационная система. В результате проектирования информационной системы описано поведение системы и взаимодействие с ней пользователя.

Ключевые слова: просопография, историко-ориентированные информационные системы.

Пермский государственный национальный исследовательский университет – это не только учебное заведение, но и история, люди и их научные достижения. Преподаватели, когда-либо работавшие в ПГНИУ, играют значительную роль в образовательном процессе, формируют интеллектуальный потенциал молодого поколения и вдохновляют студентов на совершение научных открытий. Они являются неотъемлемой частью столетней истории университета, её достижений. Создание коллективного портрета данного профессионального сообщества позволит изучить не только пермских ученых, но и то, как их личности и образование влияют на науку, что обуславливает актуальность данной темы.

В рамках работы рассмотрено проектирование и документирование информационной системы, которая будет совмещать в себе информацию о преподавателях ПГНИУ, об их биографии и научных достижениях.

Объект исследования: создание коллективных портретов профессиональных сообществ за счет разработки информационной системы. Предмет исследования – просопографические информационные системы, посвященные работникам высшего образования.

Целью работы является проектирование и документирование информационной системы «Преподаватели пермского университета 1916-2023».

Для достижения цели необходимо выполнить ряд задач:

1. Изучить понятие “просопография”;
2. Изучить понятие “информационная система”. Описать классификацию информационных систем;
3. Рассмотреть аналоги просопографических информационных систем;



4. Изучить подходы к моделированию ИС. Выбрать подходящий;
5. Изучить нотации и CASE-средства моделирования ИС. Выбрать подходящие;
6. Изучить средства прототипирования пользовательского интерфейса ИС. Выбрать подходящий;
7. Смоделировать систему через различные диаграммы;
8. Смоделировать прототип пользовательского интерфейса.

Проектируемая система является исторической (историко-ориентированной) по сфере применения. Историко-ориентированные информационные системы – “особый класс систем, предназначенных для хранения, организации исторической информации, обеспечения доступа к ней и её аналитической обработки в соответствии с потребностями исторических исследований и (или) образования” [1].

Подходом к созданию нашей информационной систем будет являться источник-ориентированный и проблемно-ориентированный одновременно, так как данные и модель представления данных будут браться, в основном, из одного источника, а также от системы будет требоваться не только накопление и систематизация сведений о работниках университета, но и решение задач просопографического исследования, которые будут затрагивать информационную картину о научной деятельности преподавателей в определённый период времени или по другим критериям, например, принадлежности конкретной кафедре.

Выделим главные роли пользователей системы: незарегистрированный пользователь, зарегистрированный пользователь и администратор. Все они являются пользователями системы, которым доступен общий функционал (просмотр карточек преподавателей, поиск карточки по параметрам, проведение исследования, предложение записи).

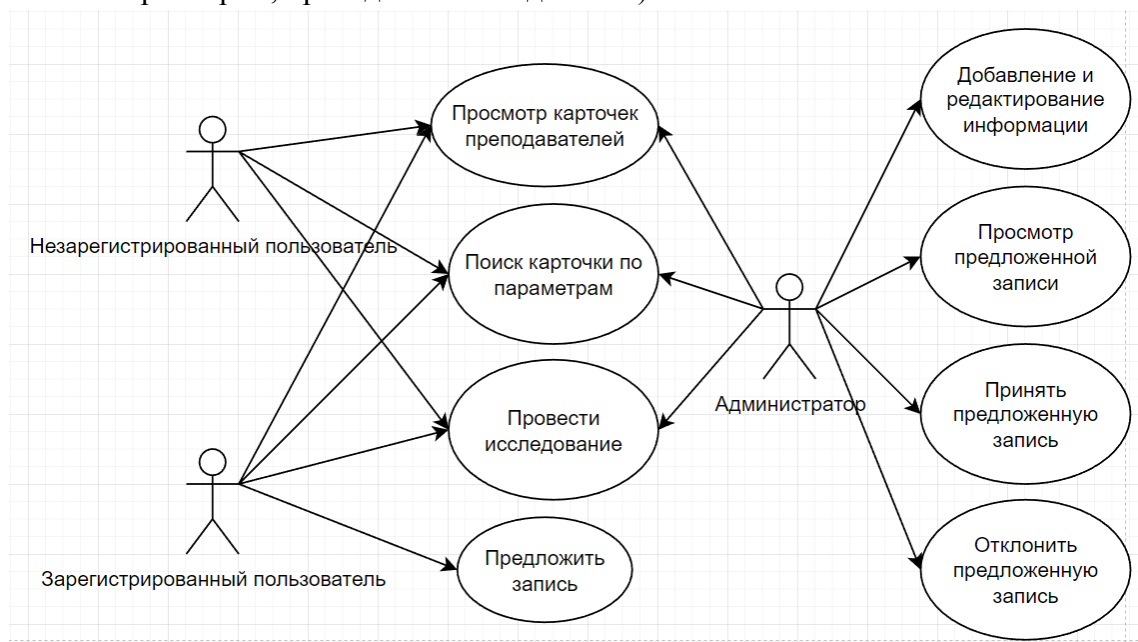


Рис. 1. Диаграмма прецедентов

Наиболее важная роль в системе – это роль администратора. Главная задача администратора в системе – это управление базой данных, а также взаимодействие с другими зарегистрированными пользователями. Процесс взаимодействия администратора и зарегистрированного пользователя, когда второй предлагает новую заявку, изображено на диаграмме деятельности (рис. 2).

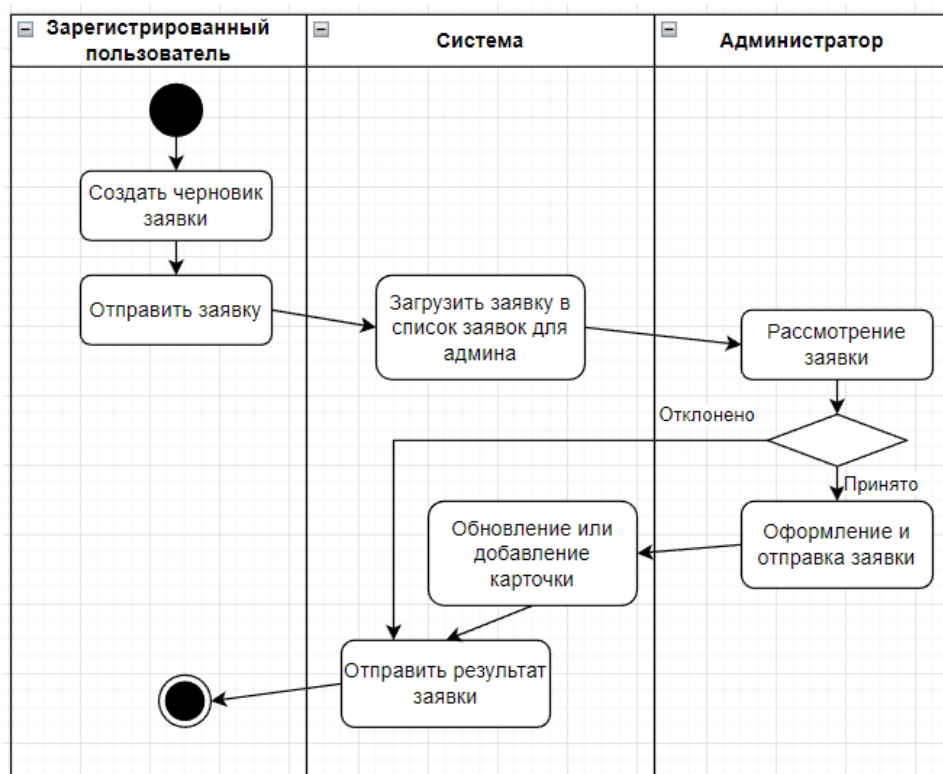


Рис. 2. Диаграмма деятельности

Главным источником информации о преподавателях будет служить книга “Профессора Пермского университета 1916-2016” [2], которая содержит информацию в виде “карточек”.

Диаграмма классов, изображённая на рис. 3, описывает структуру базы данных карточек преподавателей.

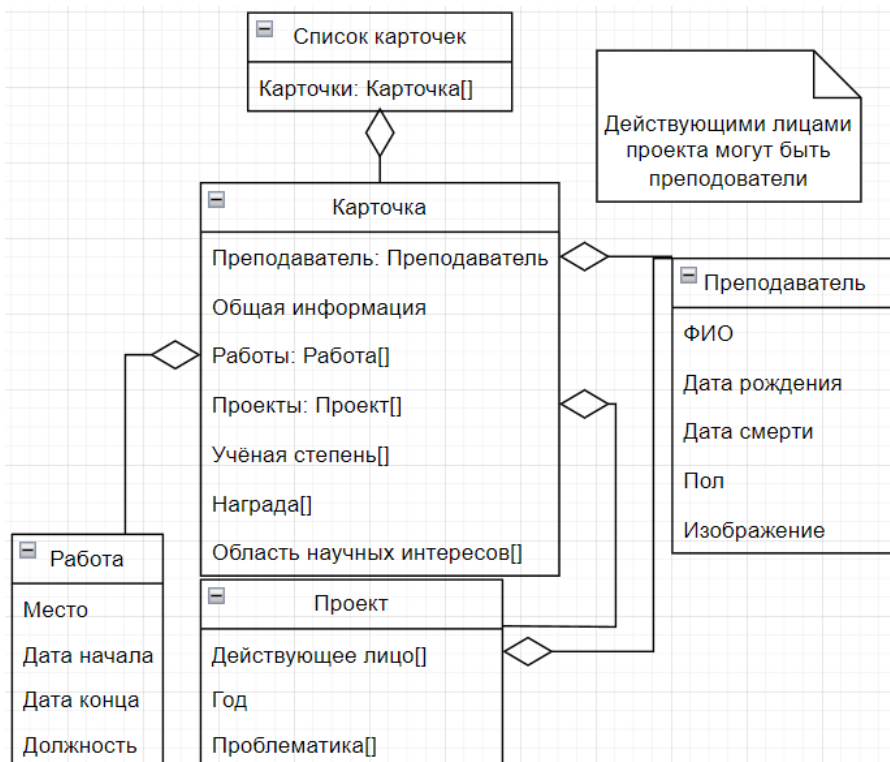


Рис. 3. Диаграмма классов

Далее рассмотрим страницу поиска информации о преподавателях. На странице можно задать поиск по алфавиту, выбрав букву из списка. Можно произвести простой поиск, введя информацию в большую строку, а также осуществить гибкий поиск, введя информацию в поля по категориям, например, по фамилии, дате рождения, наградам и многим другим. Можно провести исследование по введённым данным в гибкий поиск, нажав на кнопку “ПРОВЕСТИ ИССЛЕДОВАНИЕ”. В таком случае, если были введены в поля гибкого поиска какие-либо данные, система выдаст пользователю значения других полей, которые в каких-то карточках преподавателей были в сочетании с введёнными данными, а также сами карточки. Эта функция – пример автоматизации просопографического исследования. Также на странице есть кнопка “ПРЕДЛОЖИТЬ ЗАПИСЬ”, которая пересылает на страницу подачи заявки со стороны пользователя. Прототип заполненной страницы поиска представлен на рис. 4, а результат поиска – на рис. 5.

Рис. 4. Страница поиска

Рис. 5. Страница результата поиска

Выводимые карточки преподавателей придерживаются структуры, которая была описана на диаграмме классов. На карточке сначала пишется фамилия, имя, отчество, дата рождения и дата смерти, далее идёт общая информация, которая представлена в виде статьи. После этого идёт информация по категориям, которые используются в гибком поиске.

У каждой карточки справа-снизу есть звёздочка, при нажатии на которую пользователя перенаправляет на страницу, где можно предложить дополнить или внести изменения в эту уже существующую карточку. Эта страница является той самой страницей, на которой производится подача заявки со стороны пользователя, но где уже есть заполненные данные.

В результате была проведена работа по проектированию и документированию информационной системы “Преподаватели пермского университета 1916-2023”. Работа включала в себя проектирование системы при помощи диаграмм UML-нотации, а также описание пользования системой на примере прототипа пользовательского интерфейса. Разработанная система позволит не только собирать данные о преподавателях ПГНИУ за все годы его существования, но и создать их коллективный портрет.

### **Библиографический список**

1. *Кирьянов И.К., Корниенко С.И., Гагарина Д.А., Рябухин И.В.* Информационный ресурс по парламентской истории России начала XX в. // Власть. 2010. № 12. С. 80-85.
2. Профессора Пермского университета 1916-2016. Гл. ред. *В.И. Костицын* – Перм. гос. нац. исслед. ун-т. – Пермь. 2016. 1-336 с.

### **References**

1. *Kir'yanov I.K., Kornienko S.I., Gagarina D.A., Ryabuhin I.V.* Informacionnyj resurs po parlamentskoj istorii Rossii nachala XX v. // Vlast'. 2010. № 12. 80-85. (In Russ.).
2. *Kosticyn V.I.* Professora Permskogo universiteta 1916-2016. Perm State University – Perm. 2016. 1-336. (In Russ.).

## **TEACHERS OF PERM STATE UNIVERSITY 1916-2023: PROJECT OF PROSOPOGRAPHIC INFORMATION SYSTEM**

*Matkin Mikhail Alekseevich*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, matkinpmi@gmail.ru

*Averin Sergey Igorevich.*

State National Research Politechnical University of Perm, Komsomolsky Av. 29, Perm, Russia, 614000, averin-s-i@ya.ru

The work contains an analysis of the prosopographic approach to the research of historically-oriented information systems, a description of approaches to the design of information systems. The paper describes the process of designing and documenting the information system “Teachers of Perm University 1916-2023”. The subject and object of research are defined, the relevance of the development of an information system is formulated, tasks that need to be solved during the design are set. The functional capabilities that the information system under development should have are defined. As a result of the information system design, the behavior of the system and the user's interaction with it are described.

Keywords: prosopography, historical-oriented information systems.

## ОПЫТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ «АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ КАРТ КОНДИТЕРСКИХ ИЗДЕЛИЙ ДЛЯ КОНДИТЕРА»

*Миргалеев Артем Александрович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, artem.mirgaleev@mail.ru

*Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, averin-s-i@yandex.ru

В данной работе рассматривается опыт разработки информационной системы для автоматизации работы кондитера с технологическими картами кондитерских изделий. Кондитерская промышленность является одной из старейших и самых разнообразных отраслей пищевой промышленности. Кондитерская промышленность – это высококонкурентный и постоянно развивающийся рынок, на котором регулярно появляются новые продукты и инновации. Процесс производства кондитерских изделий требует строгого соблюдения определенных рецептов и процессов для обеспечения высокого качества конечного продукта. Одной из основных задач, стоящих перед кондитерами, является создание точных и последовательных технологических карт для своей продукции. Технологическая карта является ценным инструментом для кондитеров, поскольку дает четкое представление о процессе производства, начиная с сырых ингредиентов и заканчивая готовым продуктом. [1]

Ключевые слова: автоматизация, технологические карты, кондитер, кондитерские изделия.

Разработка веб-приложения обычно делится на два этапа: frontend и backend.

На frontend-е создаются пользовательский интерфейс и все, что видит пользователь, включая HTML, CSS и JavaScript. Frontend-разработчик отвечает за верстку страницы, стилизацию элементов и функциональность пользовательского интерфейса.

На backend-е создаются серверные части приложения, которые работают в фоновом режиме и обрабатывают данные. Это может быть написание кода на языках программирования, таких как PHP, Python, Ruby, Java и т.д., и управление базой данных.

Первым этапом был разработан frontend веб-приложения. Для разработки интерфейса использовались технологии HTML [2], CSS [3] и JavaScript [4]. Frontend веб-приложения был написан на основе прототипа интерфейса, разработанного в Figma.

Для backend разработки использован язык PHP [5], он был выбран ранее во второй главе.

Далее рассмотрим поэтапно процесс разработки информационной системы.

На странице авторизации создана форма для ввода логина и пароля, пароль будет скрыт, т. к. используется тип password. Код формы представлен на рис. 1. В теге input указан параметр required, этот параметр делает обязательным ввод и логина, и пароля, т. е. без ввода невозможно отправить форму. Форма отправляется в файл signin.php.

```
<form action="signin.php" method="post">
  <div><input class="inp_log" type="text" name="login" placeholder="Имя пользователя" required></div>
  <div><input class="inp_pass" type="password" name="password" placeholder="Пароль" required></div>
  <input type="submit" class="button-auth" value="Войти">
</form>
```

Рис. 1. Код формы авторизации

На рис. 2 показан код PHP для вывода сообщений о неправильности ввода логина или пароля.

```
<?php

    if(($SESSION['message'] ?? '') != '')
    {
        echo '<p class="msg">'. $SESSION['message'] .'
```

Рис. 2. Код для вывода сообщений

Теперь перед формой авторизации добавляем проверку на уже авторизованного пользователя. Происходит проверка, установлены ли значения для ключа 'login' в массиве \$SESSION. Если значение для ключа 'login' установлено, то это значит, что пользователь уже авторизован на сайте и происходит перенаправление на главную страницу сайта. Код проверки представлен на рис. 3.

```
session_start();

if (($SESSION['login'] ?? '') != '') {
    header('Location: index.php');
}
```

Рис. 3. Проверка на авторизацию 1

На всех остальных страницах похожая проверка, но на неавторизованного пользователя. Если пользователь не авторизован, то происходит переход на страницу авторизации. Проверка представлена на рис. 4.

```
session_start();
$db = @new mysqli('localhost', 'root', '', 'tk');

if (($SESSION['login'] ?? '') == '') { //если не вошли, то перенаправляем на страницу входа
    header('Location: login.php');
}
```

Рис. 4. Проверка на авторизацию 2

При попытке удалить тип изделия, открывается модальное окно с вопросом для возможности отмены операции. Код окна представлен на рис. 5. Изначально окно прозрачно, но при нажатии на кнопку удаления оно отображается за счет кода CSS: прозрачность меняется с 0 на 1. Код CSS показан на рис. 6.

```
<div id="openQuest" class="modal2">
    <div class="modal-dialog2">
        <div class="modal-content">
            <div class="modal-header">
                <h3 class="modal-title">Вы действительно хотите удалить тип изделия?</h3>
            </div>
            <div class="modal-body">
                <article class="popup-card">
                    <a class="button-order" >Да</a>
                    <a href="#" class="button-order" >Отменить</a>
                </article>
            </div>
        </div>
    </div>
</div>
```

Рис. 5. Код для отображения модального окна HTML

```
.modal2:target {
    opacity: 1; /* делаем окно видимым */
    pointer-events: auto; /* элемент видим для событий мыши */
    overflow-y: auto; /* добавляем прокрутку по y, когда элемент не помещается на страницу */
}
```

Рис. 6. Код для отображения модального окна CSS

После отправки формы происходит обработка загрузки изображения. Код проверяет наличие ошибок при загрузке и выводит соответствующее сообщение, если они есть. Если ошибок нет, то производится проверка на размер изображения, генерируется имя изображения и добавляется в папку '/techcards/imgs/'. Также выполняется запрос к базе данных для добавления информации о загруженной картинке и названии типа. Код проверки изображения и добавления в таблицу приведен на рис. 7.

```
// Проверим на ошибки
if ($errorCode != UPLOAD_ERR_OK || !is_uploaded_file($fileTmpName)) {
    // Массив с названиями ошибок
    $errorMessages = [
        UPLOAD_ERR_INI_SIZE => 'Размер файла превысил значение upload_max_filesize в конфигурации PHP.',
        UPLOAD_ERR_FORM_SIZE => 'Размер загружаемого файла превысил значение MAX_FILE_SIZE в HTML-форме.',
        UPLOAD_ERR_PARTIAL => 'Загружаемый файл был получен только частично.',
        UPLOAD_ERR_NO_FILE => 'Файл не был загружен.',
        UPLOAD_ERR_NO_TMP_DIR => 'Отсутствует временная папка.',
        UPLOAD_ERR_CANT_WRITE => 'Не удалось записать файл на диск.',
        UPLOAD_ERR_EXTENSION => 'PHP-расширение остановило загрузку файла.',
    ];
    // Зададим неизвестную ошибку
    $unknownMessage = 'При загрузке файла произошла неизвестная ошибка.';
    // Если в массиве нет кода ошибки, скажем, что ошибка неизвестна
    $outputMessage = isset($errorMessages[$errorCode]) ? $errorMessages[$errorCode] : $unknownMessage;
    // Выведем название ошибки
    die($outputMessage);
} else {
    // Создадим ресурс FileInfo
    $fi = finfo_open(FILEINFO_MIME_TYPE);
    // Получим MIME-тип
    $mime = (string) finfo_file($fi, $fileTmpName);
    // Проверим ключевое слово image (image/jpeg, image/png и т. д.)
    if (strpos($mime, 'image') === false) die('Можно загружать только изображения.');
```

// Результат функции запишем в переменную

```
$image = getimagesize($fileTmpName);
// Зададим ограничения для картинок
$limitBytes = 1024 * 1024 * 5;
$limitWidth = 1280;
$limitHeight = 768;
// Проверим нужные параметры
if (filesize($fileTmpName) > $limitBytes) die('Размер изображения не должен превышать 5 Мбайт.');
```

if (\$image[1] > \$limitHeight) die('Высота изображения не должна превышать 768 точек.');

if (\$image[0] > \$limitWidth) die('Ширина изображения не должна превышать 1280 точек.');

```
// Сгенерируем новое имя файла через функцию getRandomFileName()
$name = getRandomFileName($fileTmpName);
// Сгенерируем расширение файла на основе типа картинки
$extension = image_type_to_extension($image[2]);
// Сократим .jpeg до .jpg
$format = str_replace('jpeg', 'jpg', $extension);
// Переместим картинку с новым именем и расширением в папку /upload
if (!move_uploaded_file($fileTmpName, __DIR__ . '/techcards/imgs/' . $name . $format)) {
    die('При записи изображения на диск произошла ошибка.');
```

Рис. 7. Код добавления нового типа изделия

Функция для генерации названия изображения приведена на рисунке 8.



```
function getRandomFileName($path)
{
    $path = $path ? $path . '/' : '';
    do {
        $name = md5(microtime() . rand(0, 9999));
        $file = $path . $name;
    } while (file_exists($file));

    return $name;
}
```

Рис. 8. Функция генерации названия

Для формирования pdf-файла используется библиотека PHP tcpdf. Код представлен на рис. 9. Код использует cURL для получения содержимого указанного URL, который представляет собой PHP-файл, генерирующий содержимое для печати в PDF-документе. Затем полученное содержимое записывается в PDF-документ с помощью метода writeHTML. Наконец, вызывается метод Output для отправки PDF-документа в браузер для просмотра, причем параметр 'I' означает, что PDF будет отображаться в браузере, а не загружаться. Также отправляется заголовок refresh, чтобы предотвратить перезагрузку страницы после создания PDF-документа.

```
require_once('../tcpdf/tcpdf.php');

$pdf = new TCPDF();

$url = "http://test1.ru/techcards/tk_pdf.php?id=$tk_id";
if(isset($_GET['print']) && !empty($_GET['print'])) {
    if ($_GET["print"] == "true") { // Если GET-параметр равен
        ob_end_clean();
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        // Получение содержимого страницы
        $html = curl_exec($ch);
        // Закрытие сессии CURL
        curl_close($ch);
        $pdf->writeHTML($html, true, false, true, false, '');
        $pdf->Output('my_document.pdf', 'I');
        header("refresh: 0; url=#");
    }
}
```

Рис. 9. Код формирования pdf-файла

На странице типа изделия выводятся все фотографии изделий этого типа, а также считается и выводится цена этого изделия. Код для подсчета цены приведен на рис. 10. Для каждой записи формирует запрос для выбора цены продукта по его имени из таблицы products. Выполняет запрос для выбора цены продукта и сохраняет результат в переменной \$price. Далее вычисляет стоимость ингредиента как произведение массы ингредиента в граммах на цену продукта за килограмм.

```
$query = "SELECT * FROM ".$table_ingr;
$result = mysqli_query($connect, $query);
// Вычисляем стоимость всех ингредиентов
$total_cost = 0;
while ($row = mysqli_fetch_assoc($result)) {
    // считаем цену
    $query_price = "SELECT price FROM products WHERE name='".$row['name']."'";
    $price = mysqli_query($connect, $query_price);
    $product = mysqli_fetch_assoc($price);
    $total_cost += ($row['netto'] / 1000) * $product['price'];
}
```

Рис. 10. Код для подсчета цены



Также существует возможность списать продукты необходимые для приготовления изделия. Для этого пользователь нажимает «Приготовить». Далее происходит проверка, достаточно ли имеющихся запасов на складе для приготовления необходимого количества определенного продукта. Если запасов недостаточно, отображается модальное окно с сообщением об ошибке. Если запасов достаточно, другой цикл обновляет запасы, вычитая количество, необходимое для каждого продукта, и вставляет новую запись в таблицу "затраты" в базе данных. Код проверки представлен на рис. 11.

```
while ($row = mysqli_fetch_assoc($result)) {
    $query = "SELECT * FROM products WHERE name = '". $row['name']."'";
    $result2 = mysqli_query($connect, $query);

    $stock_quantity = mysqli_fetch_assoc($result2);
    $amount_to_cook = $row['netto']/1000;

    if ($amount_to_cook > $stock_quantity["quantity"]) {
        $i++;
    }
}
if ($i > 0) {
    >>
    //ОШИБКА Продуктов на складе не хватает!
    <div id="openModal" class="modal">
    <div class="modal-dialog2">
    </div>
    </div>
    <?php
}
else{
    while ($row = mysqli_fetch_assoc($result)) {
        //вычитаем продукты из склада
        $sql = "UPDATE products SET quantity = quantity - {$row['netto']}/1000 WHERE name = '". $row['name']."'";
        mysqli_query($connect, $sql);
    }
    $sql = "INSERT costs(name, price) values ('". $name."', '". $total_cost."'";
    mysqli_query($connect, $sql);
}
```

Рис. 11. Код для вычета продуктов со склада

Также имеется возможность сортировки по бисквиту и начинке, список формируется аналогичным способом со списком возможных типов при добавлении тех. карты. Сортировка производится аналогично поиску по тех. картам, поэтому так же не рассматривается.

Информационные системы являются неотъемлемой частью современного бизнеса и позволяют организациям значительно повысить свою эффективность и конкурентоспособность. Однако, как и любая другая технология, информационные системы нуждаются в постоянном улучшении и совершенствовании, чтобы оставаться актуальными и соответствовать требованиям рынка.

Рассмотрим рекомендации по совершенствованию ИС:

1. Добавление импорта технологических карт из разных источников рецептов.
2. Добавление уведомлений о том, что продукты на складе заканчиваются.
3. Внедрение возможности поставить свой процент наценки на кондитерское изделие, чтобы не менять цену в базе данных и сразу работать с процентами.

В результате работы была разработана система для автоматизации формирования технологических карт кондитерских изделий.

### **Библиографический список**

1. ГОСТ 31987-2012 Услуги общественного питания. Технологические документы на продукцию общественного питания. Общие требования к оформлению, построению и содержанию URL: <https://docs.cntd.ru/document/1200103473> (дата обращения: 15.06.2023)
2. HTML // HTML Standart [Электронный ресурс]. – URL: <https://html.spec.whatwg.org/> (дата обращения: 15.06.2023)
3. Каскадные таблицы стилей// W3C [Электронный ресурс]. – URL: <https://www.w3.org/Style/CSS/> (дата обращения: 15.06.2023)
4. Современный учебник JavaScript // Современный учебник JavaScript [Электронный ресурс]. – URL: <https://learn.javascript.ru/> (дата обращения: 15.06.2023)
5. PHP // PHP: Hypertext Preprocessor [Электронный ресурс]. – URL: <https://www.php.net/> (дата обращения: 15.06.2023)

### **References**

1. GOST 31987-2012 Public catering services. Technological documents on public catering products. General requirements for design, construction and content URL: <https://docs.cntd.ru/document/1200103473> (In Russ.).
2. HTML // HTML Standart [Electronic resource]. – URL: <https://html.spec.whatwg.org/>
3. Cascading Style Sheets// W3C [Electronic resource]. – URL: <https://www.w3.org/Style/CSS/>
4. Modern JavaScript Tutorial// Modern JavaScript Tutorial [Electronic resource]. – URL: <https://learn.javascript.ru/> (In Russ.).
5. PHP // PHP: Hypertext Preprocessor [Electronic resource]. – URL: <https://www.php.net/>

## **EXPERIENCE IN THE DEVELOPMENT OF THE INFORMATION SYSTEM "AUTOMATION OF TECHNOLOGICAL CHARTS OF CONFECTIONERY PRODUCTS FOR CONFECTIONER"**

*Mirgaleev Artem A.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [artem.mirgaleev@mail.ru](mailto:artem.mirgaleev@mail.ru)

*Averin Sergey I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [averin-s-i@yandex.ru](mailto:averin-s-i@yandex.ru)

This work considers the experience of developing an information system to automate the work of the confectioner with technological maps of confectionery products. Confectionery industry is one of the oldest and most diverse sectors of the food industry. The confectionery industry is a highly competitive and constantly developing market, where new products and innovations appear regularly. The confectionery manufacturing process requires strict adherence to certain recipes and processes to ensure a high quality end product. One of the main challenges confectioners face is creating accurate and consistent flow charts for their products. A flow chart is a valuable tool for confectioners as it provides a clear picture of the production process from raw ingredients to the finished product.

Keywords: automation, technological charts, confectioner, confectionery products.

## ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭЛЕКТРОННАЯ КУЛИНАРНАЯ КНИГА»

*Мосеева Юлия Сергеевна, Огурецкая Анна Игоревна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, super.xblegje2014@yandex.ru

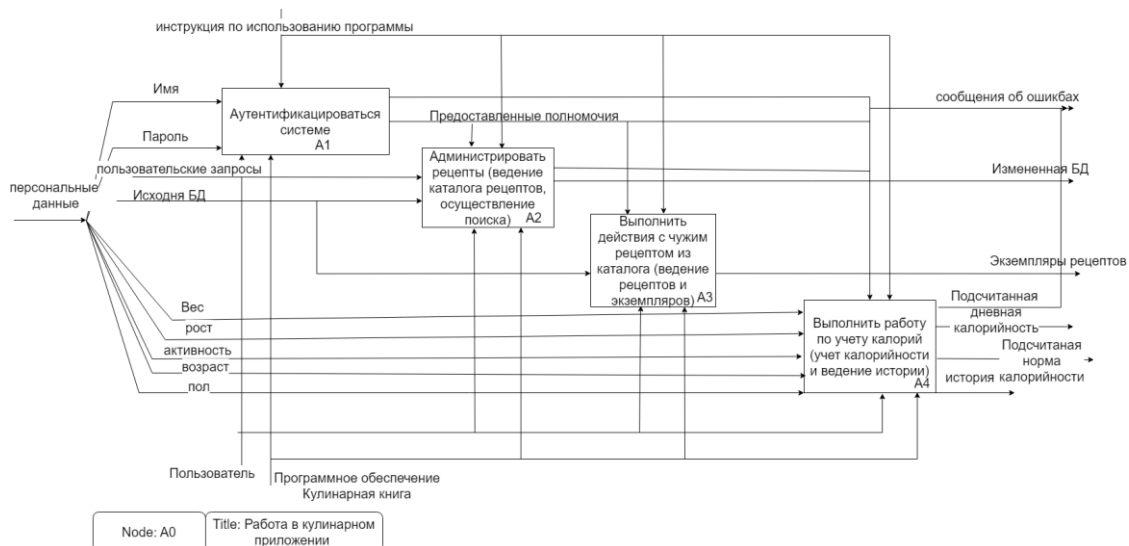
Рассматриваются вопросы проектирования и разработки прототипа информационной системы «Электронная кулинарная книга». Система содержит такие функции, как расчет ингредиентов на основании калорийности и количества порций по определённым формулам, расчет суточной калорийности с использованием личных данных пользователя, подсчет суммы суточной калорийности, учитывая приготовленные рецепты из книги и другую употребленную пользователем пищу. Работа содержит рассмотрение аналогичных систем и теории кулинарных книг, выбор инструментальных средств для проектирования и разработки, выбор архитектуры и типа приложения. Было произведено проектирование информационной системы с учетом выводов, сделанных после обзора аналогов. В ходе работы были выделены функции системы, разработана структура базы данных, построены диаграммы и разработан прототип информационной системы.

**Ключевые слова:** информационная система «Кулинарная книга», расчет ингредиентов, подсчет нормы калорийности, учет суточной калорийности.

Кулинарная книга – это справочник, содержащий различные рецепты приготовления пищи.

На первом этапе мы воспользуемся структурным подходом для выделения основных функций системы, потому что структурный подход удобен на начальных этапах проектирования для более точного и строго разбиения на подзадачи и функции.

На диаграмме IDEF0 A0 представлены основные функциональные блоки приложения (см. рис. 1). В скобках указаны пояснения соответственно функциям.



**Рис. 1. Диаграмма IDEF0 A0 Работа в кулинарном приложении**

Публикация подготовлена при финансовой поддержке РФФИ (грант № 02-01-00701 и 03-01-06350-МАС) и Американского Фонда Гражданских Исследований и Развития (грант PE-009-0).

© Мосеева Ю.С., Огурецкая А.И., 2023

## Каталог рецептов

При открытии бокового меню пользователь может перейти на различные вкладки:

- «Добавить рецепт» – содержит форму для добавления рецепта;
- «Каталог» – содержит список рецептов, распределенный по категориям;
- «Экземпляры рецептов» – содержит экземпляры рецептов, для которых рассчитано количество ингредиентов;

количество ингредиентов;

- «История» – содержит все употребленные блюда (ежедневно очищается);

• «Учет калорий» – содержит форму для заполнения личных параметров, форму для внесения дополнительных приемов пищи;

- «Созданные рецепты» – содержит рецепты, добавленные пользователем.

- «Выйти» – позволяет осуществить выход из аккаунта.

При необходимости найти рецепт пользователь может воспользоваться фильтрацией по названию. Также присутствует переход по категориям. Сверху расположена строка с суточной калорийностью (рассчитывается на основании личных параметров) и суммой калорийности (суммируется калорийность приготовленных блюд и дополнительных приемов пищи) (см. рис. 2).

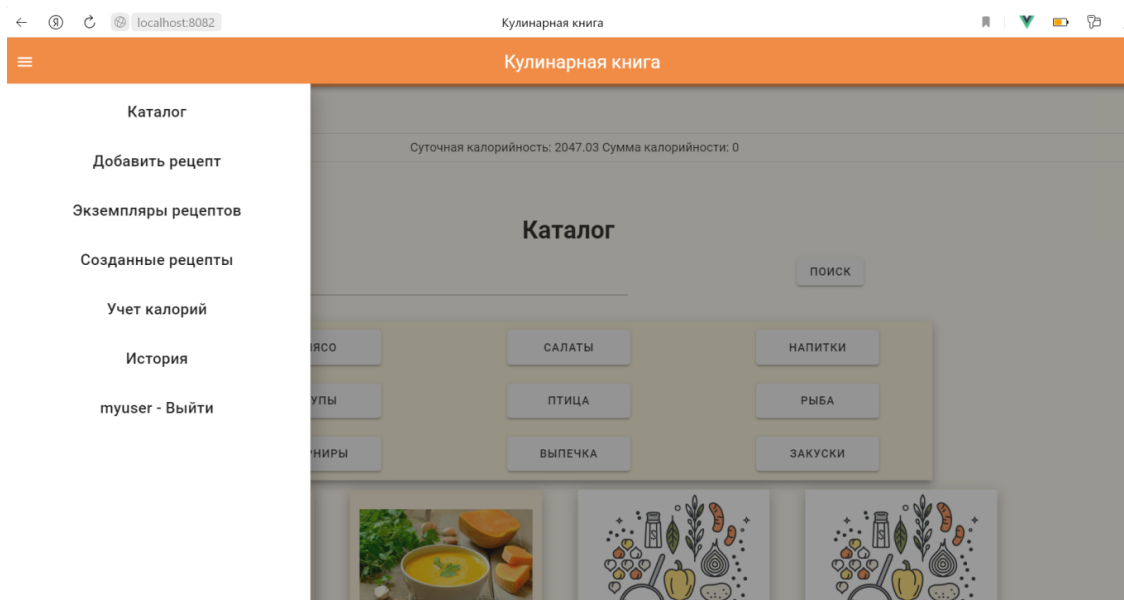


Рис. 2. Каталог рецептов

## Учет калорий

Если пользователь хочет получить свою норму калорийности и рассчитывать дневную калорийность, то при переходе в модуль «Учет калорий» необходимо заполнить форму с личными параметрами. После этого будет рассчитана суточная калорийность, которая отобразится в верхней строке. Также при нажатии кнопки «Приготовлено» на страницах экземпляров рецептов их калорийность будет прибавлена к сумме калорийности, указанной в верхней строке. Кроме этого, на этой же странице в нижней форме пользователь может внести другую употребленную пищу, указав ее название, калорийность и вес (см. рис. 3).

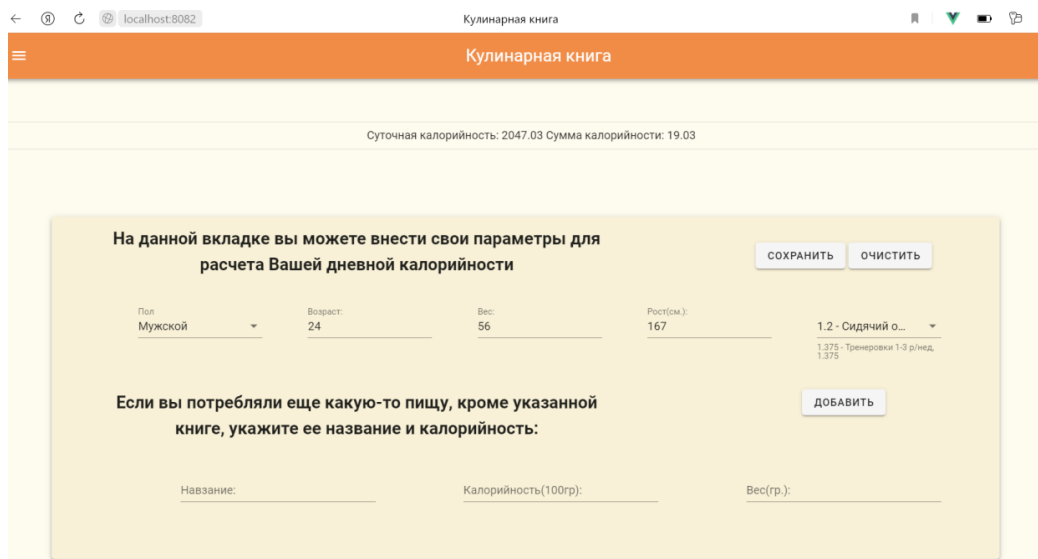


Рис. 3. Учет калорий

Для расчета нормы дневной калорийности используется формула Миффлин-Сан Жеора, которая содержит следующие данные: пол, вес, рост, возраст, коэффициент активности.

$K = (10 * \text{вес(кг)} + 6.25 * \text{рост(см)} - 5) * \text{возраст(г)} * A$  – для мужчин (1),

$K = (10 * \text{вес(кг)} + 6.25 * \text{рост(см)} - 5 * \text{возраст(г)} - 161) * A$  – для женщин (2),

Коэффициент А показывает степень активности человека.

### Расчет ингредиентов

В ходе работы с приложением пользователь может пересчитать количество ингредиентов, исходя из количества порций или калорийности блюда.

Формулы, с помощью которых производится расчет, представлены ниже.

Количество ингредиентов рецепта зависит от количества порций (задаются пользователем).

Расчетные формулы, учитывая количество порций:

$$k_{\text{пересч}} = \frac{V_{\text{общ}}}{V_{\text{исх}}} \quad (3),$$

где  $k_{\text{пересч}}$  – коэффициент пересчета ингредиентов,  $V_{\text{исх}}$  – исходный вес по рецепту.

$$P_{\text{инг}} = P_{\text{ингИсх}} * k_{\text{пересч}} \quad (4),$$

где  $P_{\text{инг}}$  – вес отдельного ингредиента,  $P_{\text{ингИсх}}$  – исходный вес ингредиента по рецепту

$$K_{\text{рец}} = K_{\text{рецИсх}} * k_{\text{пересч}} \quad (5),$$

где  $K_{\text{рец}}$  – калорийность полученного рецепта,  $K_{\text{рецИсх}}$  – калорийность исходного рецепта

Расчетные формулы, учитывая калорийность:

$$k_{\text{пересч}} = \frac{K_{\text{треб}}}{K_{\text{рецИсх}}} \quad (6),$$

где  $K_{\text{треб}}$  – требуемая калорийность.

## Архитектура прототипа приложения

ИС «Кулинарная книга» представляет собой одностраничное Web-приложение. Это увеличивает производительность за счет не полной перезагрузки страницы, а подгрузки отдельных элементов. Также это дает возможность использовать более гибкий и насыщенный интерфейс. ИС состоит из клиентской и серверной частей, разработанных в IDE IntelliJ IDEA.

Клиентская часть представляет собой HTML код страниц с использованием UI-компонентов фреймворка Vuetify. Функционал элементов задается посредством JavaScript, что предоставляет фреймворк Vue.js[1].

Серверная часть создана с использованием фреймворка Spring boot на языке Kotlin[2]. Серверная часть содержит модели данных, соответствующие структурам таблиц баз данных, а также контроллеры и сервисы. Контроллеры отвечают за взаимодействие с пользователем, принимая запросы от клиентской части и обрабатывая их через сервисы.

В качестве СУБД используется PostgreSQL[3].

Клиент-серверное взаимодействие организуется с помощью клиентской библиотеки Axios, основанной на Promise HTTP-клиента (со стороны клиента) и Spring Web (со стороны сервера).

В рамках работы была создана диаграмма развертывания (см. рис. 4).

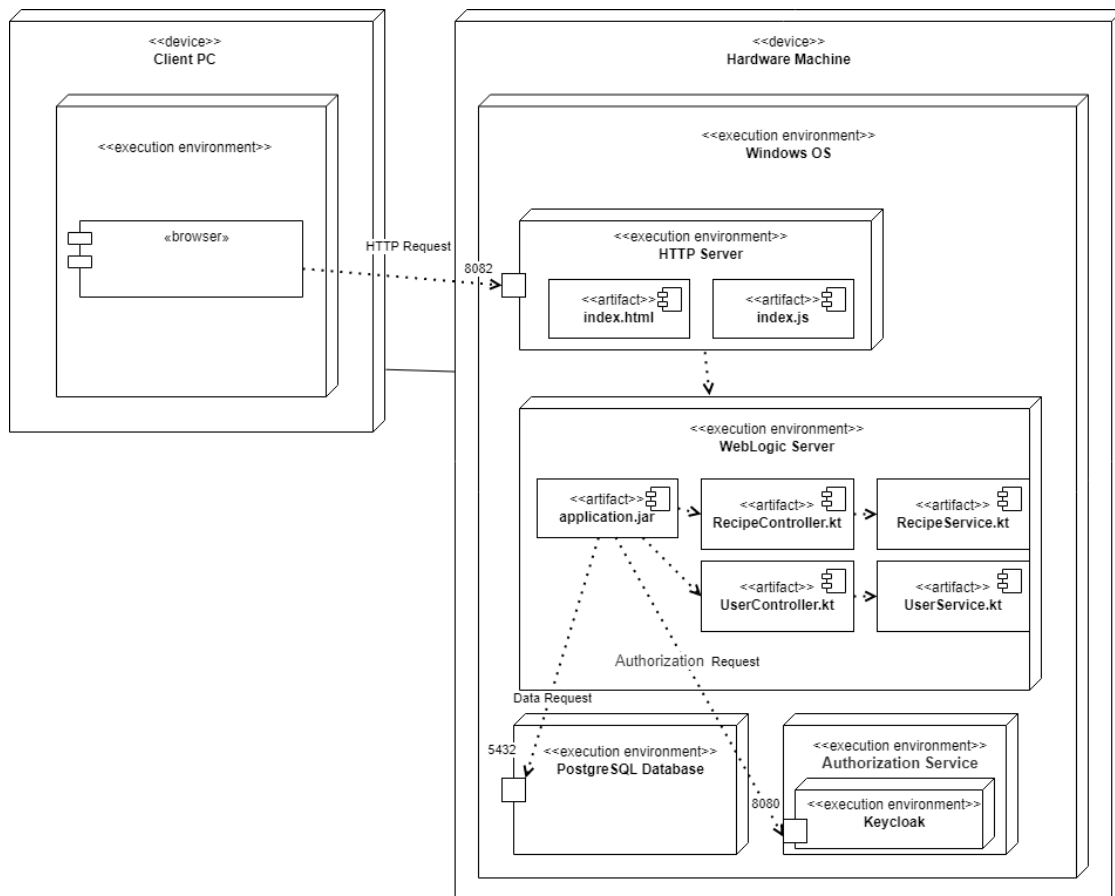


Рис. 4. Диаграмма развертывания

### Библиографический список

1. Документация Vue.js [Электронный ресурс]. – URL: <https://ru.vuejs.org/v2/guide/> (дата обращения: 15.05.2023).
2. Язык программирования Kotlin: что это, где применяется и за что его любят разработчики [Электронный ресурс]. – URL: <https://skillbox.ru/media/code/yazyk-programmirovaniya-kotlin/> (дата обращения: 15.05.2023).
3. Документация к PostgreSQL 15.3 [Электронный ресурс]. – URL: <https://postgrespro.ru/docs/postgresql/15/index> (дата обращения: 15.05.2023).

### References

1. Dokumentatsiya Vue.js [Electronic resource]. – URL: <https://ru.vuejs.org/v2/guide/> (date of the application: 15.05.2023). (In Russ.).
2. YAzyk programmirovaniya Kotlin: chto eto, gde primenyaetsya i za chto ego lyubyat razrabotchiki [Electronic resource]. – URL: <https://skillbox.ru/media/code/yazyk-programmirovaniya-kotlin/> (date of the application: 15.05.2023). (In Russ.).
3. Documentation for PostgreSQL 15. [Electronic resource]. – URL: <https://postgrespro.ru/docs/postgresql/15/index> (date of the application: 15.05.2023). (In Russ.).

### ELECTRONIC COOKBOOK

*Oguretskaya Anna I. Moseeva Yulia S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [super.xblegje2014@yandex.ru](mailto:super.xblegje2014@yandex.ru)

The issues of designing and developing a prototype of the information system "Electronic cookbook" are considered. The system contains functions such as calculating ingredients based on caloric content and the number of servings according to certain formulas, calculating daily calorie content using the user's personal data, calculating the amount of daily calorie content, taking into account cooked recipes from the book and other food consumed by the user. The work contains a review of similar systems and the theory of cookbooks, the choice of tools for design and development, the choice of architecture and type of application. An information system was designed taking into account the conclusions made after a review of analogues. In the course of the work, the functions of the system were identified, the structure of the database was developed, diagrams were built, and a prototype of the information system was developed.

Keywords: information system "Cookbook", calculation of ingredients, calculation of calorie content, daily calorie counting.

**ПРОЕКТИРОВАНИЕ СИСТЕМЫ АГРЕГАЦИИ RSS-КАНАЛОВ (RSS-РИДЕР)**

*Музафаров Владислав Игоревич, Кнутова Наталия Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, vlad.psu@mail.ru

RSS технология, предназначена для автоматического сбора и обновления информации с веб-сайтов. Она позволяет пользователям получать доступ к новостям, статьям, обзорам и другой информации не только на самом сайте, но и на других ресурсах. Один из самых удобных способов получить доступ к этой информации – использование RSS-reader. Это приложение, которое обеспечивает возможность подписываться на RSS-каналы, отслеживать обновления и получать своевременные уведомления об изменениях на сайтах. В последнее время набирает популярность мобильные приложения, которые позволяют управлять своим списком RSS каналов, но таких приложений не очень много, а в существующих есть свои недостатки. В статье рассматривается процесс проектирования RSS-reader, со стадии формирования функциональных требований и до разработки пользовательского интерфейса. Для проектирования RSS-reader будут рассмотрены несколько важных аспектов, которые включают изучение предметной области и существующих аналогичных систем, определение функциональных возможностей собственной системы, выбор средств проектирования и разработки, а также платформы для создания приложения.

Ключевые слова: RSS-reader, разработка, приложение, информация.

RSS – это автоматически генерируемый документ в формате XML, содержащий данные об опубликованном контенте на веб-сайтах.

Существуют специальные приложения, которые оповещают пользователя о новых материалах, появившихся на сайте. После чего пользователь может просмотреть новые материалы в этом приложении. [1,2].

Документ представляет собой блочную структуру, содержащую:

1. Метаданные, состоящие из заголовка, подзаголовков, изображений, аннотации и основного текста контента, опубликованного на сайте.
2. Гиперссылки на целевой ресурс для ознакомления с полным вариантом публикации.
3. Метаданные других публикаций, схожих по тематике.

Данная технология предоставляет ряд преимуществ:

1. Специальные приложения собирают RSS-файлы с источников, на которые подписался пользователь и формируют из них единую ленту, которую удобно быстро просматривать без необходимости посещения каждого источника по отдельности.
2. Технология RSS предоставляется полностью бесплатно и не требует никаких вложений.
3. За счёт того, что информация сразу же загружается в RSS-файл, уведомление о выходе новой публикации сразу же передается пользователю.

В наше время у каждого человека есть мобильный телефон, но при этом далеко не все имеют без лимитного доступа к мобильному интернету. В этой ситуации приложения, которые используют технологию RSS становятся особенно полезными, поскольку позволяют экономить интернет-трафик. Вместо открытия десятка страниц в браузере, оптимальнее под-



писаться на различные RSS каналы, чтобы всегда быть в курсе интересных новостей и событий, не перегружая свою систему.

### **RSS и концентрация внимания**

Каждый день человек обрабатывает большие объемы информации. Как показало исследование [3], проведенное Роджером Боном и Джеймсом Шортом в 2008 году. Человек в среднем потребляет в день в общей сложности 34 гигабайта информации. Из-за такого объема данных у человека происходит информационная перегрузка и его концентрация внимания снижается.

Информационная перегрузка – это проблема, с которой сталкиваются люди при получении большого количества информации, которое они не способны переработать [4].

Чтобы этого не происходило существуют программы, которые позволяют снизить шум или перераспределить нагрузку. Многие из них используют технологию RSS. Это онлайн-сервисы для чтения новостей, блогов, журналов и других материалов в одном месте. Они позволяют не тратить время на поиск материалов на разных сайтах, а один раз добавить все ваши любимые источники в приложение и получить все новости в одном месте. Рассмотрим, как это влияет на концентрацию:

Во-первых, добавив все источники в одно место, происходит экономия времени на поиски и сокращает количество входящей информации.

Во-вторых, RSS позволяет настроить фильтры, чтобы отображать только интересующие материалы, что способствует снижению негативного воздействия информационного потока и информационного шума на мозг.

Информационный шум [5] – это избыточное количество информации, которое усложняет доступ к важной информации. Он возникает, когда информационный объем превышает возможность восприятия пользователями. Например, когда в социальных сетях появляется множество новостей, сообщений и рекламных объявлений, которые затрудняют поиск и восприятие важной информации. Информационный шум может вызывать утомление, стресс и снижение производительности [6].

### **Анализ существующих систем и формирование требований к системе**

Был проведен анализ существующих систем и рассмотрены их сильные и слабые стороны после чего была составлена таблица, в которой представлены ключевые особенности аналогичных продуктов.

Таблица 1. Сравнительный анализ аналогов

Приложение	Feedly	Inoreader	NewsBlur
Поддержка Русского языка	-	+	-
Интуитивно-понятный интерфейс	+	-	+/-
Платные функции	+	+	+
Функционал базовой версии (количество источников/ количество каналов)	100/3	150/∞	∞/∞
фоновая подгрузка	-	-	-
поддержка	+	+	-
дополнительные функции	+	+	+

Основываясь на сравнительном анализе (табл. 1), были сформированы требования к разрабатываемому приложению.

Функциональные:

Источники информации: пользователь может получать информацию из различных источников такие как: google news, telegram, email newsletter, twitter feed, youtube, twitch, rutube, facebook (проект Meta Platforms Inc., деятельность которой в России запрещена), reddit, Яндекс дзен.

Функции: для того, чтобы приложение было удобно в пользование в нём будут представлены ряд полезных функций: перевод статей, просмотр истории, закладки, пометки в тексте, сохранения контента.

Персонализация: в приложение должны быть возможности настроить его под себя: изменять размер текста, поменять фон приложения, сменить аккаунт, настроить аккаунт и многое другое.

Рекомендации: Рекомендованный контент должен подстраиваться под предпочтения пользователя.

- Платная функция: возможность добавлять свой сайт в ленту рекомендаций.

Нефункциональные требования:

- Версия Android: приложение должно быть разработано для ОС Android (версия Lollipop и более поздние).

- Автономный режим: приложение должно будет предоставлять доступ, к контенту, который был последний раз загружен (без подключения к сети Интернет).

- Локализация: интерфейс приложения должен быть на русском языке.

Для описания поставленных функциональных требований и структуры приложения был выбран язык моделирования UNL. Схема база данных была спроектирована с помощью MySQL Workbench. Основным языком программирования, на котором будет разработано приложение – java, а для управления базой, данной в которой будет храниться информация о RSS-каналах будет использоваться СУБД MySQL.

### **Концепция мобильного приложения**

Мобильное приложение будет предоставлять пользователю возможность авторизации и регистрации. Для этого в приложение будет использоваться технологию под названием «Единая точка входа» [7]. Это система, которая позволяет пользователю авторизоваться с одним идентификатором и паролем, независимо от количества приложений или сервисов, в которых он нуждается. После успешной авторизации пользователь получает доступ ко всем разрешенным им ресурсам, не прибегая к повторному вводу учетных данных. SSO может работать как внутри компании, так и с внешними поставщиками услуг (например, социальными сетями). Он повышает безопасность, удобство использования и управляемость системы.

Существует много различных реализаций технологии SSO, которые можно легко интегрировать в систему. Для работы была выбрана реализация под названием «Avanpost FAM» [8] в которой есть режим работы IDP который подходит для мобильных приложений и использует протокол аутентификации «OAuth2.0» [9].

OAuth2.0 – это стандарт безопасности, позволяющий одному приложению получить разрешение на доступ к информации в другом приложении. Протокол позволяет предоставлять одной службе права доступа к пользовательским ресурсам другой службы. Он

устраняет необходимость доверять приложению логин и пароль, а также позволяет предоставлять ограниченный набор прав, а не все сразу.

В OAuth 2.0 существуют различные потоки авторизации, в реализации «Avanpost FAM» используется: «Authorization Code Flow» [10] с дополнением: «Authorization Code Flow with Proof Key for Code Exchange». Этот поток используется для мобильных приложений, а дополнение помогает реализовать безопасность пользовательского входа.

После авторизации в мобильном приложении перед пользователем появится главное меню, в котором будут находиться различные функции приложения, например, такие как поиск, рекомендации, «Закладки», «читать позже», лента новостей и настройки.

Главными особенностями мобильного приложения, отличающими его от других похожих приложений, является то, что оно будет поддерживать совместимость со стриминговыми сервисами и видеохостингами. У таких сервисов существует свой RSS-файл. Но при этом ни одно из приложений RSS-reader не реализует поиск и удобное взаимодействие с ними. Поэтому в приложение будет предоставляться возможность подписки на соответствующие каналы данных сервисов. Каналы будут собраны в одном месте и опубликованный контент можно будет просмотреть, не открывая сервисы в браузере. Также у пользователей будет бесплатный доступ ко всем сервисам новостей, в то время как в аналогичных приложениях данный функционал входит в платную подписку. Также в приложение будет присутствовать встроенный браузер и возможность добавлять свой собственный сайт.

### Проектирование интерфейса

Интерфейс системы будет состоять из 4 отдельных компонентов, которые в совокупности будут образовывать целое приложение. Первый компонент, с которым будет сталкиваться пользователь это окно авторизации. За его основу я взял мягкий синий цвет, который не раздражает взгляд и приятен на вид.

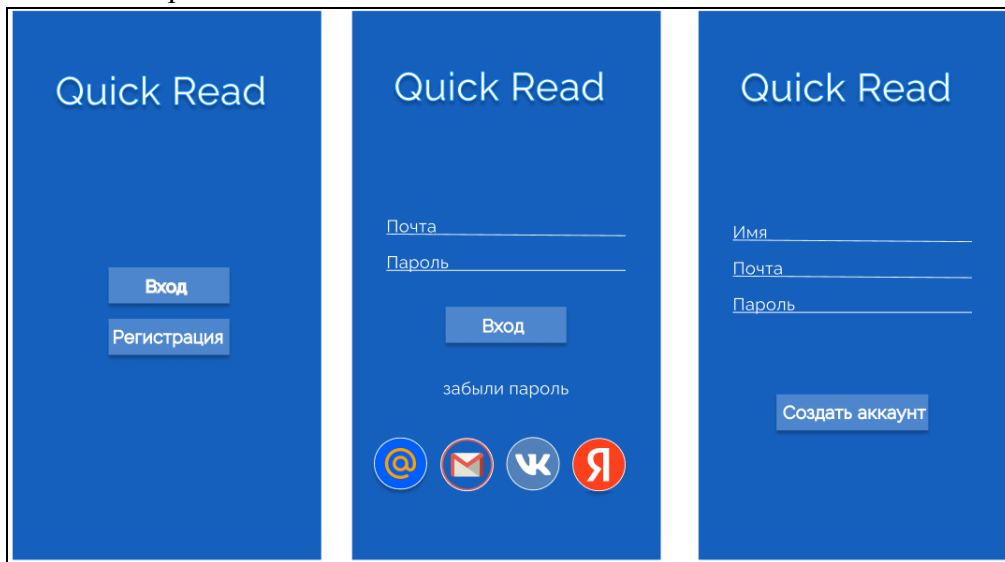
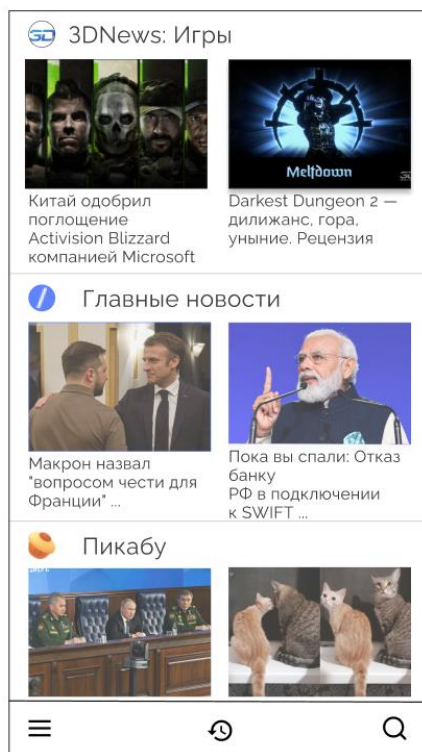


Рис. 1. Экраны авторизации

На спроектированных экранах (см. рис. 1) пользователю будет предоставляться возможность войти в систему используя личные данные или воспользовавшись одним из вариантов быстрого входа. А также есть возможность зарегистрироваться в приложение воспользовавшись окном регистрации.

В качестве следующего компонента будет реализован экран рекомендаций, который представляет из себя интерфейс, содержащий названия источников и блоки с уже

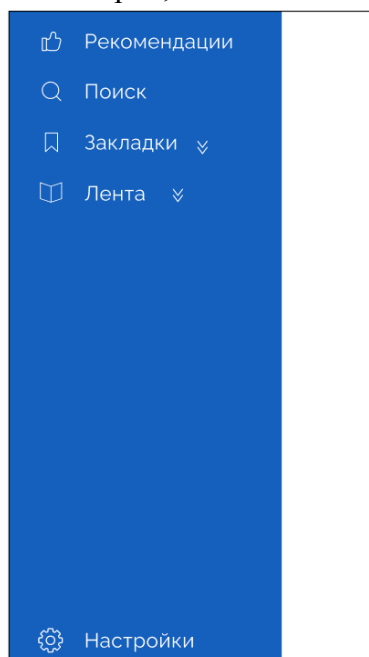
опубликованным на них контентом (см. рис. 2). Чтобы посмотреть больше контента будет присутствовать возможность прокручивать данные блоки вправо и влево. А чтобы посмотреть больше источников нужно будет прокручивать экран вниз. В нижней части экрана будут находиться три кнопки: меню, истории поиска.



**Рис. 2. Экран рекомендации**

Центральным компонентом откуда будут доступны все основные функции приложения является экран меню, он будет предоставлять доступ пользователю к таким функциям как: рекомендации, поиск, читать позже, просмотр ленты и настройкам приложения (см. рис. 3).

Ленту и закладки можно будет раскрыть и получить доступ к ленте подписанных новостей или к таким функциям как история, читать позже и сохраненные.



**Рис. 3. Экран главного меню**

Последним блоком, обладающим индивидуальным интерфейсом, включает в себя экран поиска.

Экран поиска состоит из: строки, в которую будет вводиться запрос поиска, ленты специальных каналов в которых можно осуществлять поиск и места с результатами поиска (см. рис. 4).

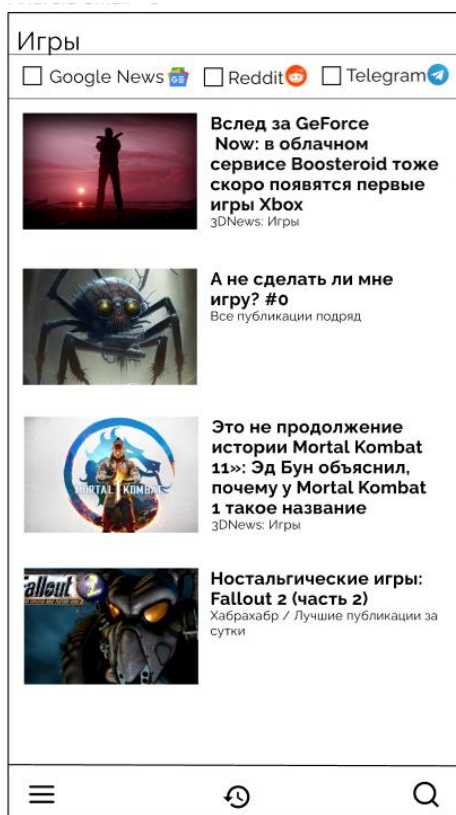


Рис. 4. Экран поиска.

### Библиографический список

1. *Земсков А.И.* Что такое RSS? / А.И. Земсков // Научные и технические библиотеки: Ежемес.науч.-практический журнал для специалистов библиотечно-информационной и смежных отраслей. – 2007. – №6. – С. 25-34. – ISSN 0130-9765
2. *Казенников А.О.* Извлечение информации из RSS-каналов / А.О. Казенников, А.Г. Тюрин // Труды всероссийской научной конференции "Инновационные стратегии развития науки, техники и общества. Социальная инноватика". – Москва: ФГУП ГНЦ РФ «ВНИИгеосистем», 2014. – С. 94-97.
3. *Bohn R.* Measuring Consumer Information / R. Bohn, J. Short // The International Journal of Communication. – 2012. – № 6. – С. 21-24.
4. *Стрекалова Н.Б.* Влияние информационной перегрузки на жизнедеятельность личности // Проблемы и перспективы развития образования в России. 2011. №12. URL: <https://cyberleninka.ru/article/n/vliyanie-informatsionnoy-peregruzki-na-zhiznedeyatelnost-lichnosti> (дата обращения: 22.05.2023).
5. *Игнашин А.А., Котлярова В.В.* Феномен Информационного Шума // Форум молодых ученых. 2019. №6 (34). URL: <https://cyberleninka.ru/article/n/fenomen-informatsionnogo-shuma> (дата обращения: 22.05.2023).
6. *Шараев М.Р.* Информационный шум в условиях развития цифрового общества / М.Р. Шараев, И.В. Жажнева, А.М. Ахмедова // Вызовы глобализации и развитие цифрового

общества в условиях новой реальности. – Москва: Общество с ограниченной ответственностью "Издательство АЛЕФ", 2023. – С. 181-183.

7. Демидова А.Ю., Жуков А. В. Технология Single Sign On: инструменты централизованной аутентификации для функциональной системы сервисов // ИВД. 2020. №3 (63). URL: <https://cyberleninka.ru/article/n/tehnologiya-single-sign-on-instrumenty-tsentralizovannoy-autentifikatsii-dlya-funktsionalnoy-sistemy-servisov> (дата обращения: 22.05.2023).

8. Avanpost FAM [Электронный ресурс]. – URL: <https://www.avanpost.ru/products/avanpost-fam/> (дата обращения: 22.05.2023)

9. OAuth 2.0 [Электронный ресурс]. – URL: <https://oauth.net/2/> (дата обращения: 22.05.2023)

10. Authorization Code Flow [Электронный ресурс]. – URL: <https://auth0.com/docs/get-started/authentication-and-authorization-flow/authorization-code-flow> (дата обращения: 22.05.2023)

### References

1. Zemskov A.I. What is RSS? / A. I. Zemskov // Scientific and technical libraries: Monthly scientific and practical journal for specialists in the library, information and related industries. – 2007. – №6. – S. 25-34. – ISSN 0130-9765

2. Kazennikov A.O. Extracting information from RSS channels / A. O. Kazennikov, A. G. Tyurin // Trudy vs Rossijskoj nauchnoj konferencii "Innovacionnye strategii razvitiya nauki, tekhniki i obshchestva. Social'naya innovatika . – Moscow: Federal State Scientific Center RF "VNIIGeo-sistem", 2014. – S. 94-97. (In Rus)

3. Bohn R. Measuring Consumer Information / R. Bohn, J. Short // The International Journal of Communication. – 2012. – № 6. – С. 21-24.

4. Strekalova N.B. Influence of information overload on the life of a person // Problemy i perspektivy razvitiya obrazovaniya v Rossii. 2011. No. 12. URL: <https://cyberleninka.ru/article/n/vliyanie-informatsionnoy-peregruzki-na-zhiznedeyatelnost-lichnosti> (date of access: 05/22/2023).

5. Ignashin A.A., Kotlyarova V.V. Phenomenon of Information Noise // Forum of Young Scientists. 2019. No. 6 (34). URL: <https://cyberleninka.ru/article/n/phenomen-informatsionnogo-shuma> (date of access: 05/22/2023).

6. Sharaev M.R. Information noise in the context of the development of a digital society / M.R. Sharaev, I.V. Zhazhneva, A.M. Akhmedova // Vyzovy globalizacii i razvitie cifrovogo obshchestva v usloviyah novoj real'nosti – Moscow: Limited Liability Company "ALEF Publishing House", 2023. – P. 181-183.

7. Demidova A.Yu., Zhukov A.V. Single Sign On Technology: Centralized Authentication Tools for a Functional Service System // IVD. 2020. No. 3 (63). URL:

8. Avanpost FAM [Electronic resource] URL: <https://www.avanpost.ru/products/avanpost-fam/> (date of access: 05/22/2023)

9. OAuth 2.0 [Electronic resource] URL: <https://oauth.net/2/> (date of access: 05/22/2023)

10. Authorization Code Flow [Electronic resource] URL: <https://auth0.com/docs/get-started/authentication-and-authorization-flow/authorization-code-flow> (accessed 05/22/2023)

## DESIGN OF RSS-CHANNELS AGGREGATION SYSTEM (RSS-READER)

*Muzafarov Vladislav I., Knutova Natalia S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, ivanov@email.ru vlad.psu@mail.ru

RSS is a technology that is designed to automatically collect and update information from websites. It allows users to access news, articles, reviews and other information not only on the site itself, but also on other resources. One of the most convenient ways to access this information is to use an RSS reader. This is an application that provides the ability to subscribe to RSS feeds, track updates and receive timely notifications of changes on sites. Recently, mobile applications have been gaining popularity that allow you to manage your list of RSS feeds, but there are not very many such applications, and the existing ones have their drawbacks. Within the framework of the article, the process of designing an RSS reader is considered, from the stage of formation of functional requirements to the development of a user interface. To design an RSS reader, several important aspects will be considered, which include studying the subject area and existing similar systems, determining the functionality of your own system, choosing design and development tools, and platforms for creating an application.

Keywords: RSS-reader, development, application, information.

## **ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА УЧЕБНОГО СИМУЛЯТОРА: «НЕФТЯНОЕ МЕСТОРОЖДЕНИЕ: ОХРАНА ОКРУЖАЮЩЕЙ СРЕДЫ»**

*Наумов Владислав Григорьевич, Кнутова Наталия Сергеевна, Зайцев Андрей Аркадьевич,  
Кулакова Светлана Александровна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, vlad.naumov.2000@gmail.com

Данная статья посвящена изучению и разработке программного обеспечения, а именно обучающего практическим навыкам симулятора в геологической сфере. Обучающие симуляторы набирают популярность во всем мире, потому что идет переходный период, когда традиционная система образования начинает использовать современные информационные технологии. Данный вид симулятора хорошо себя зарекомендовал в мировой практике. В статье описаны функциональные и технические требования, которые были использованы при создании данной программы. Представлено описание технической составляющей программно-го решения, используемого в симуляторе, а также интерфейсы информационной системы.

**Ключевые слова:** симулятор, информационная система, практические навыки, веб-приложение, нефтяное месторождение

Проблема отсутствия практических навыков у студентов остается актуальной уже на протяжении длинного периода времени. Современные технологии позволяют решить эту проблему. Однако, на данный момент, современные технологии используются всего лишь, в нескольких отраслях. Информационные системы для обучения людей в практической области называются симуляторы. Симуляторы зарекомендовали себя как хорошие средства для обучения практическим навыкам [1]. Актуальность выбранной темы находится на высоком уровне, в связи с началом использования симуляторов в обучении, а точнее в сфере геологических работ, поэтому географический факультет ПГНИУ принял решение создать и предоставить задания, которые легли в основу симулятора. В мире уже появляются научные работы по разработке геологических обучающих симуляторов, однако, тема остается плохо изученной.

Симуляторы являются достаточно популярными в некоторых отраслях. Раньше симуляторы использовались только в отраслях, где совершенная ошибка будет стоить жизни большому количеству людей и понесет большие финансовые потери. Самым первым симулятором принято считать летный симулятор, который появился в 1909 году [1]. Потому что во время полета пилоты должны быть максимально сосредоточены, а также знать, как себя вести в нештатных ситуациях. На реальных самолетах невозможно безопасно и без риска воспроизвести все экстренные ситуации, которые могут возникать при полете из-за технических проблем или погодных условий. Со временем, симуляторы показали хорошую обучаемость, настолько что их начали использовать и в других областях.

Существуют современные научные работы, в которых описываются компьютерные средства, с помощью которых возможно обучать врачей для проведения операций [2]. Однако такие технологии требуют использования дополнительных VR технологий, для точности воспроизведений обстановки. Для авиасимуляторов, тоже необходимо целое здание с ис-



пользованием специальных машин и технологий, которые позволяют создать точные копии внутренностей определенной модели самолета, а также воссоздать физические ощущения схожие с настоящим полетом.

В последнее десятилетие приобретают популярность экономические симуляторы, которые позволяют тренировать управление бизнесом и крупными бизнес-процессами. Такие симуляторы менее затратные в разработке, а также не требуют дополнительных технологий и позволяют приобрести нужные компетенции руководителями и персоналом.

Сфера нефтедобычи тоже является высоко рискованной и финансово затратной при ошибках на месторождениях, но симуляторы не требуют дополнительных технологий для обучения людей. Геологи на месторождении используют определенные алгоритмы, в которых прописаны действия и нормативные документы, регламентирующие действия и последствия. Это позволяет теоретически избежать ошибок на поле деятельности.

Большинству выпускников тяжело применить свои знания на практике из-за отсутствия необходимого практического опыта, что приводит к нескольким проблемам. Компании неохотно берут студентов на работу, соответственно студентам трудно устроиться на работу во время или после обучения. Компаниям необходимо тратить время и средства на обучение студентов. Поэтому необходимо создать симулятор, который позволит натренировать актуальные практические навыки у студентов, позволяющие в будущем не тратить время на переобучение для выполнения профессиональных работ.

Симулятор должен отвечать определенным требованиям, разработанным в ходе обсуждения концепции симулятора с географическим факультетом ПГНИУ:

- три вида пользователей – студент, преподаватель, администратор;
- несколько видов авторизации, например через популярные социальные сети;
- свой интерфейс для каждого задания;
- независимость от операционной системы используемого пользователем;
- хранение информации в одном месте;
- студент должен иметь доступ к заданиям и дополнительной информации о заданиях;
- автоматическая проверка на правильность решения;
- преподаватель должен иметь возможность увидеть оценки студентов;
- студент должен иметь возможность увидеть свою оценку;
- защита личной информации всех пользователей;
- недоступность правильных ответов для пользователей.

Исходя из всего вышеперечисленного было принято решение организовать симулятор в виде веб-приложения, которое будет состоять из сервера и веб-страниц. Это позволит избежать проблем с доступом к симулятору, благодаря независимости от операционных систем и их версий. Для доступа к информационной системе потребуется выход в интернет. У такой концепции есть неоспоримые преимущества:

- интернет-браузеры долго остаются актуальными и даже устаревшие версии браузеров позволяют почти беспрепятственно пользоваться новыми сайтами;
- в связи с распространением дистанционного формата обучения, необходимо сделать приложение доступным не только в университете;
- использование веб-технологий позволяет поддерживать и мобильные устройства, такие как планшеты и смартфоны, вне зависимости от операционных систем (Android или iOS) на этих устройствах.

Существуют и недостатки:

- требуется стабильное интернет-соединение;
- необходим сервер для размещения симулятора;
- серверное оборудование стоит дороже, чем клиентские машины пользователей.

Симулятор является клиент-серверным приложением. Для разработки серверной части использована платформа Node.js и соответственно язык программирования JavaScript. Это позволило создать серверную часть, которую легко будет поддерживать благодаря тому, что Node.js создан на двигателе Chrome V8 [3]. Также этой платформе характерна повышенная скорость выполнения исходного кода относительно других серверных платформ на C или Python. Применение Node.js позволило использовать большую базу готовых библиотек для написания кода, например «Express», которая идеально подходит для написания серверов [4].

Для написания клиентской части была использована библиотека «React». Это позволит обеспечить кроссплатформенность и независимость от версий браузеров на используемых компьютерах. Также использование этой библиотеки позволит легче разрабатывать интерактивные интерфейсы с динамической отрисовкой необходимых компонент [5]. «React» может работать на основе сервера на Node.js, что облегчает разработку всего симулятора из-за работы с одной платформой, как для серверной, так и для клиентской частях.

Для работы с базой данных используется ОПСУБД PostgreSQL. Это бесплатная ОПСУБД, которая имеет в своем арсенале широкий набор инструментов. Ей доверяет множество известных компаний таких как: Huawei, Hitachi, Amazon Redshift и т.д. Быстрота, безопасность и надежность данной ОПСУБД обосновывает ее использование в симуляторе.

В соответствии с предоставленными заданиями от географического факультета, необходимо использовать несколько видов форм с заданиями. Всего существует восемь видов форм: «Тест», «Выпадающий список», «Выбор», «Задача с фото», «Таблица», «Задания с последовательностью», «Текстовая задача», «Фото с выпадающими списками».

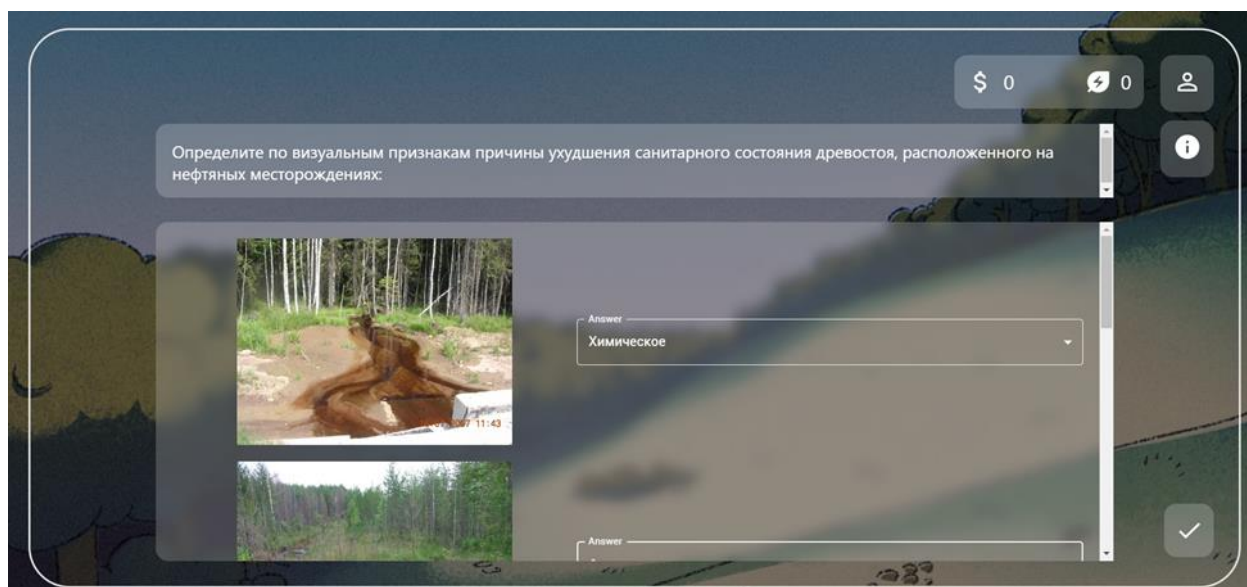
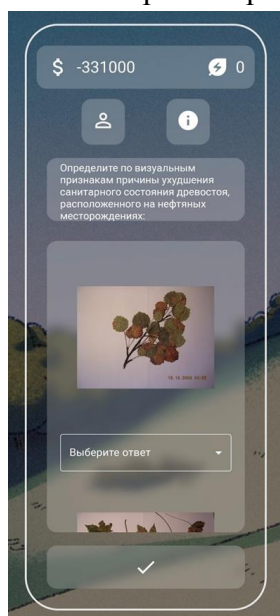


Рис. 1. Скриншот из симулятора задание «Фото с выпадающими списками»

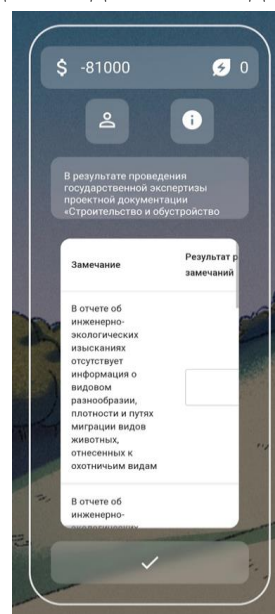
В соответствии с предоставленными заданиями от географического факультета, необходимо использовать несколько видов форм с заданиями. Всего существует восемь видов форм: «Тест», «Выпадающий список», «Выбор», «Задача с фото», «Таблица», «Задания с последовательностью», «Текстовая задача», «Фото с выпадающими списками».

Все формы объединены общим стилем оформления. В правом верхнем углу находится иконка личного кабинета, на случай если студент захочет выйти в личный кабинет. Слева от иконки личного кабинета находится поле с внутренними очками. Иконка валюта и цифровое значение рядом, обозначает доступное количество внутренней игровой валюты, доступной игроку. Иконка природы и цифровое значение рядом показывает на то сколько очков студент имеет за сохранение природы. Симулятор предполагает поиск определенного решения, которые позволит набрать максимальное количество как игровой валюты, так и очков за сохранение природы. Снизу под иконкой личного кабинета находится кнопка «Информация». При нажатии на эту кнопку выводится информация о задании, а также какими нормативными документа должен воспользоваться студент для выполнения этого задания. Слева от кнопки «Информация» находится поле, в котором находится формулировка задания. Все четыре вышеописанных поля имеет одинаковое цветовое оформление в виде белого полупрозрачного четырехугольника с закругленными краями. Под полем с формулировкой задания находится поле с разными вариантами заданий.

В ходе разработки информационной системы была спроектирована нормализованная база данных в ОПСУБД PostgreSQL, в которой хранится вся информация, начиная от заданий заканчивая личной информацией всех пользователей. Для реализации системы была выбрана клиент-серверная архитектура с серверной частью на Node.JS с фреймворком «express». Всего существует три вида пользователей это: «Студент», «Преподаватель», «Администратор». В системе реализовано 8 различных форм для существующих заданий на момент написания статьи. Автоматическая генерация страниц в зависимости от заданий в клиентской части реализована через библиотеку «React». Симулятор имеет адаптацию под мобильную версию, скриншоты которой представлены на рис. 2 и 3. Однако, некоторые виды заданий удобнее выполнять с компьютерной версии в связи с большими входными данными заданий.



**Рис. 2. Мобильная версия задания «Фото с выпадающим списком»**



**Рис. 3. Мобильная версия задания «Таблица»**

### Библиографический список

1. Росомед – Симуляционное обучение в медицине [Электронный ресурс]. – URL: <https://rosomed.ru> (дата обращения 09.03.2023)
2. Яндекс Патент – Способ лечения пациентов с хронической сенсоневральной тугоухостью путем слуховых тренировок в условиях виртуальной реальности [Электронный ресурс]. – URL: <https://yandex.ru/patents> (дата обращения 09.03.2023)
3. Node.js документация [Электронный ресурс]. – URL: <https://nodejs.org> (дата обращения 15.04.2023)
4. Habr – Руководство по Node.js [Электронный ресурс]. – URL: <https://habr.com> (дата обращения 15.04.2023)
5. React [Электронный ресурс]. – URL: <https://ru.react.js.org/docs/getting-started.html> (дата обращения 24.05.2023)

### References

1. Rosomed – Simulation training in medicine [Electronic resource]. – URL: <https://rosomed.ru> (accessed 09.03.2023) (In Russ.)
2. Yandex Patent – A method for treatment of patients with chronic sensorineural hearing loss by hearing training under virtual reality conditions [Electronic resource]. – URL: <https://yandex.ru/patents> (accessed 09.03.2023) (In Russ.)
3. Node.js documentation [Electronic resource]. – URL: <https://nodejs.org> (Accessed 15.04.2023)
4. Habr – Node.js Tutorial [Electronic resource]. – URL: <https://habr.com> (Accessed 15.04.2023) (In Russ.)
5. React [Electronic resource]. – URL: <https://ru.react.js.org/docs/getting-started.html> (Accessed 24.05.2023)

### DESIGN, DEVELOPMENT OF A TRAINING SIMULATOR: "OIL FIELD: ENVIRONMENTAL PROTECTION"

*Naumov Vladislav G., Knutova Natalia S., Zaitsev Andrey A., Kulakova Svetlana A.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [vlad.naumov.2000@gmail.com](mailto:vlad.naumov.2000@gmail.com)

This article is devoted to the study and development of software, namely, a simulator that teaches practical skills in the geological field. Educational simulations are gaining popularity around the world because there is a transitional period when the traditional education system begins to use modern information technologies. This type of simulator has proven itself well in world practice. The article describes the functional and technical requirements that were used to create this program. The description of the technical component of the software solution used in the simulator, as well as the interfaces of the information system, is presented.

Keywords: simulator, information system, practical skills, web application, oil field

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭКОЛОГИЧЕСКИЙ ТУРИЗМ ПЕРМСКОГО КРАЯ»**

*Низаметдинов Ростислав Эдуардович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, rostislav.nizametdinov@mail.ru

*Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, averin-s-i@yandex.ru

Рассматриваются основные средства для проектирования информационной системы для записи туристических маршрутов на карту Пермского Края. Определены объект и предмет исследования, сформулированы актуальность и задачи, которые нужно решить в ходе проектирования информационной системы. В ходе исследования проведен анализ информационных систем со схожей тематикой: выделены функциональные требования к будущей информационной системе, проведен анализ средств, позволяющих создавать диаграммы на объектном языке UML. В результате анализа было выбрано приложение StarUML[1]. Среди растровых графических редакторов для проектирования пользовательского интерфейса системы были выбраны приложения Paint.NET и GIMP[2,3]. В результате проектирования были созданы диаграммы, описывающие: базу данных системы, возможности разных групп пользователей, работа функции добавления маршрута на карту. Затем были спроектированы интерфейсы.

Ключевые слова: экологический туризм, туризм, Пермский Край, проектирование, документирование.

На данный момент туризм является активно развивающейся сферой общественной жизни. На его долю приходится 10,3% мирового ВВП. Туризм используется как инструмент для регионов, чтобы рекламировать свои достопримечательности и природные условия, что способствует дальнейшему вложению в эти места материальных средств от туристов, отдыхающих на данной территории.

Актуальность темы заключается в том, что Пермский Край имеет большой природный потенциал, выделенный в «Стратегии развития туризма в Пермском крае до 2035 года». Из природных условий региона в стратегии выделяются:

- Большое количество гор Северного и Среднего Урала;
- Проявления Пермского геологического периода;
- Два заповедника и 361 особенно охраняемых природных территорий, большинство которых открыты для туристов;
- Разнообразие рек и пещер. [4]

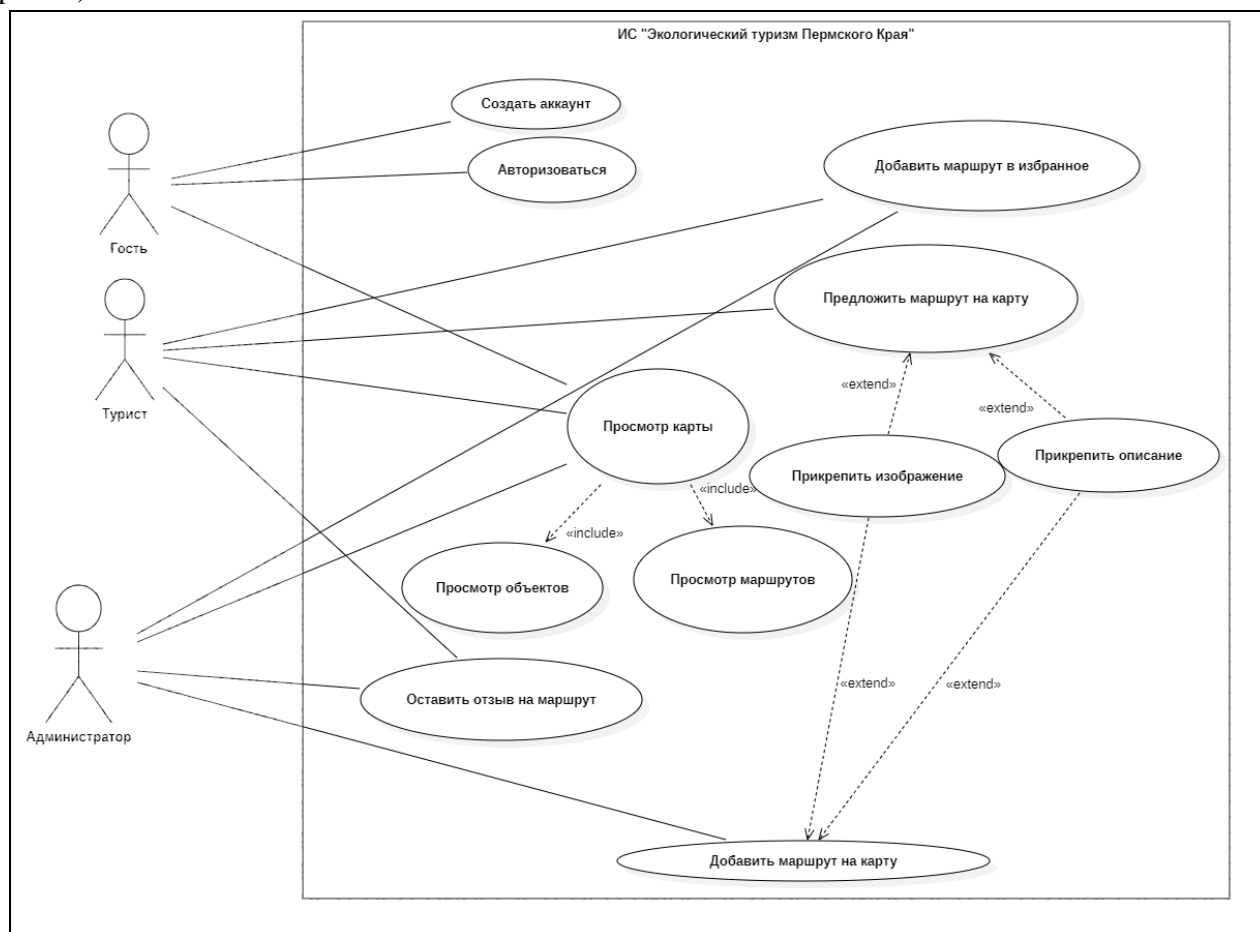
Цель исследования – совершенствование способов коммуникации между туристами и облегчение их деятельности посредством проектирования информационной системы.

Объект исследования – цифровизация туризма. Предмет исследования – информационная система для проведения экологического туризма по территории Пермского Края.

Задачами исследования являются:

- 1) Изучение явления «экологический туризм» в Пермском Крае;
- 2) Анализ существующих информационных систем;
- 3) Формирование требований к системе;
- 4) Выбор средств проектирования и документирования информационной системы;
- 5) Проектирование модели хранения данных и пользовательского интерфейса.

Проектирование системы производилось при использовании StarUML. Для обозначения возможностей пользователей используется диаграмма вариантов использования (см. рис. 1).



**Рис. 1. Диаграмма вариантов использования**

Гость – это неавторизованный пользователь, который обладает сильно ограниченными правами для использования приложения. Данный пользователь может только просматривать карту и отдельные объекты на ней или авторизоваться.

Турист – это авторизованный пользователь с возможностью предлагать новые маршруты на карту. При желании он может добавить фотографии или описания, связанные с данным маршрутом.

Администратор – это авторизованный пользователь, который добавляет предложенные маршруты или редактирует уже существующие, добавляя описания и фотографии в те маршруты, где данные объекты отсутствуют. Администратор может удалять отзывы о маршрутах других участников сервиса, если эти отзывы не несут в себе информацию, не связанную с маршрутом, или являются недостоверными.

Все группы пользователей при этом могут просматривать карту. У администратора и туриста есть возможность оставлять отзывы и мнение на маршруты других людей. Данные

отзывы составляют среднюю оценку маршрута, которая, в свою очередь, указывается при просмотре информации о данном пути.

Группы пользователей Администратор и Турист имеют права доступа для сохранения маршрутов в категорию «Избранное». Это маршруты, которые пользователи хотят посетить.

Основной функцией в системе является добавление туристического маршрута на интерактивную карту Пермского Края. Работа этой функции представлена с помощью диаграммы активности. (см. рис. 2).

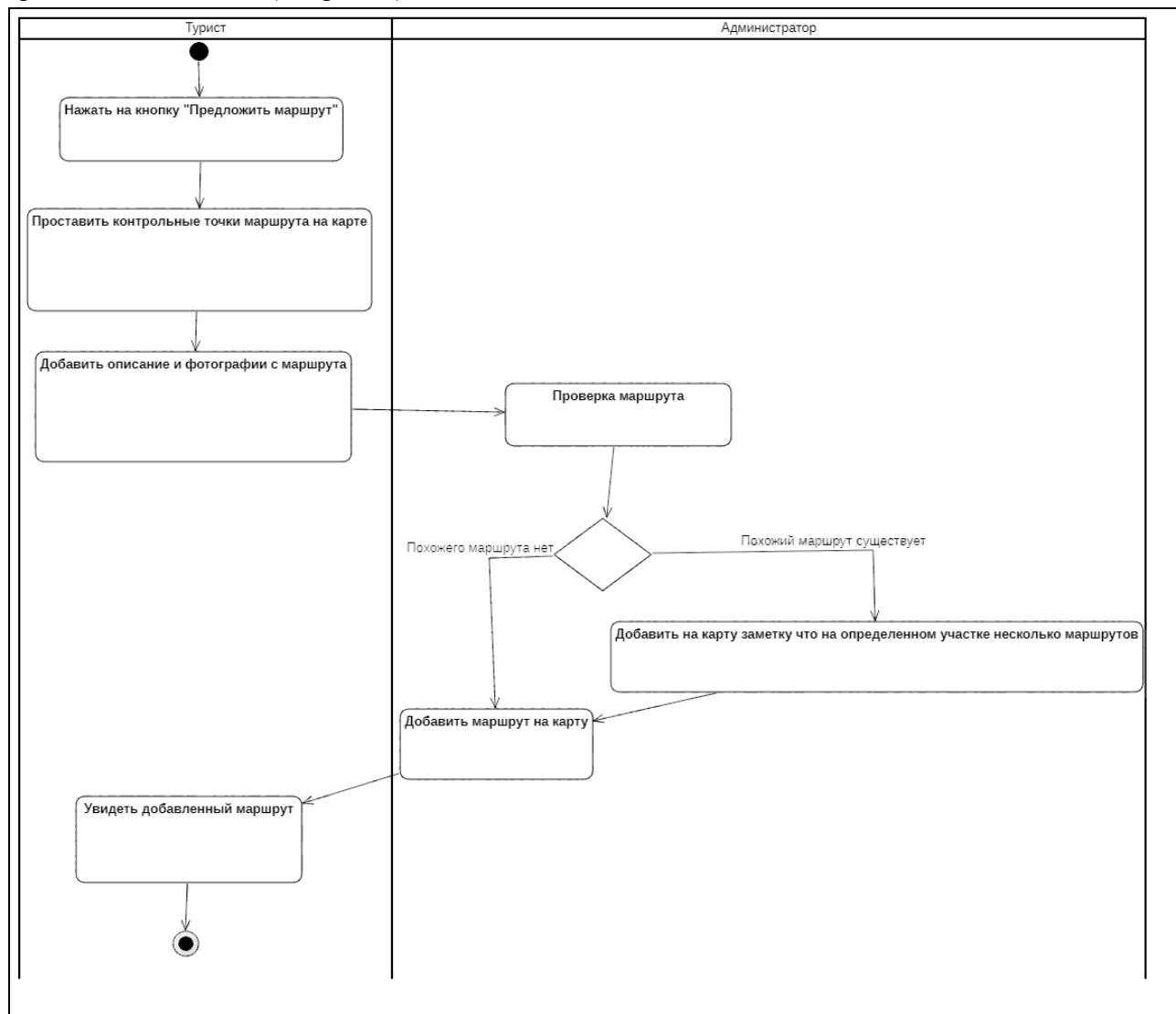


Рис. 2. Диаграмма активности

Авторизованный пользователь в своем личном кабинете нажимает на кнопку «Предложить маршрут». Перед ним появляется интерактивная карта Пермского Края. Пользователь проставляет контрольные точки на маршруте, между которыми строится линия. При желании пользователь может добавить к маршруту справочную информацию и изображения. Когда маршрут готов, пользователь отправляет запрос на его добавление – заявка отправляется администратору. Администратор проверяет маршрут. Если подобный маршрут существует, он добавляет заметку к местности, на которой проходит маршрут, или к объектам, что через них проходит несколько туристических путей. Если похожих маршрутов на территории нет, то предложенный маршрут добавляется на карту, где его могут видеть все пользователи.

Для создания модели базы данных, которая будет храниться в системе, используется диаграмма классов (см. рис. 3).

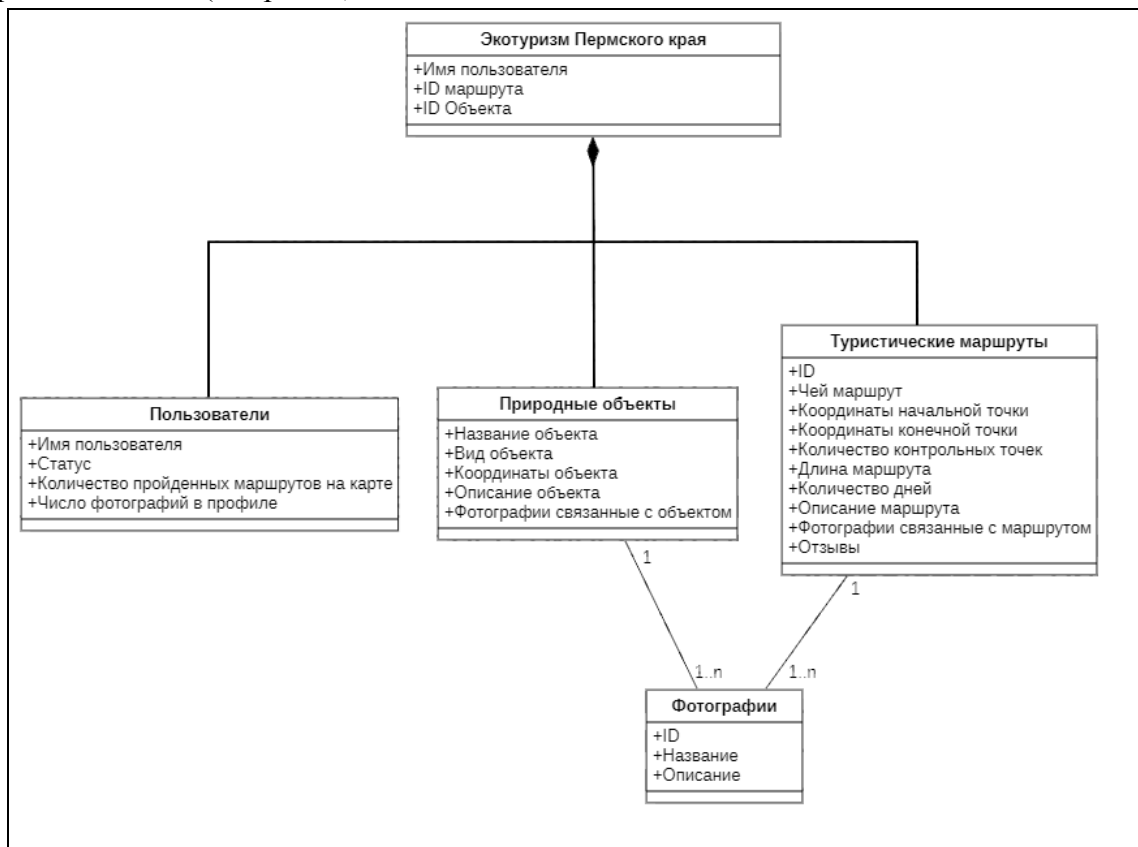


Рис. 3. Модель Базы Данных

Для обеспечения связи между разными данными создается реляционная база данных, состоящая из 5 таблиц.

Таблица «Пользователи» содержит информацию о зарегистрированных пользователях и обладает следующими атрибутами:

- Имя пользователя – имя, которое дает себе пользователь при регистрации на сайт;
- Статус – атрибут, который принимает одно из двух значений: турист, администратор;
- Количество пройденных маршрутов – общее число маршрутов, которые пользователь отметил пройденными;
- Число фотографий в профиле – общее число фотографий, выложенных у пользователя в профиле.

Таблица «Природные объекты» содержит в себе информацию о расположенных на карте природных объектах, доступных для экологического туризма. Таблица обладает 5 атрибутами:

- Название объекта – наименование, используемое на картах и в туристических справочниках;
- Вид объекта – то, чем является природный объект;
- Координаты объекта – координаты объекта на карте;
- Описание объекта – справочная информация об объекте;
- Фотографии, связанные с объектом – фотографии, на которых изображен данный объект.



Таблица «Туристические маршруты» содержит информацию о маршрутах, расположенных на карте. Обладает атрибутами:

- ID – уникальный ключ для каждого маршрута;
- Чей маршрут – имя человека, который предложил данный маршрут;
- Координаты начальной точки;
- Координаты конечной точки;
- Количество контрольных точек – количество мест, которые пользователь решил отдельно выделить;
- Длина маршрута – длина маршрута в километрах;
- Количество дней – сколько дней необходимо для преодоления маршрута;
- Описание маршрута – справочная информация о проходимом маршруте;
- Фотографии, связанные с маршрутом – фотографии с данного маршрута;
- Отзывы – средняя оценка маршрута.

Таблица «Экотуризм Пермского края» связывает таблицы «Пользователи», «Природные Объекты» и «Туристические маршруты» между собой и содержит три атрибута:

- Имя пользователя;
- ID объекта;
- ID маршрута.

Таблица «Фотографии» связана с таблицами «Природные объекты» и «Туристические маршруты» связями один ко многим и состоит из трёх атрибутов:

- ID;
- Название;
- Описание.

Для проектирования интерфейса были выбраны графические редакторы Paint.NET и GIMP за их простоту использования и работы со слоями.

Прототип интерфейса главной страницы представлен ниже (см. рис. 4).



Рис. 4. Интерфейс Главной страницы

При нажатии на «Личный кабинет» или «Войти» неавторизованный пользователь перемещается на страницу заполнения данных для входа в личный кабинет или для регистрации нового пользователя в системе.

Перед пользователями отображается два поля: «Имя пользователя» и «Пароль», рядом с которыми расположены места для введения данных. Под данными полями находятся две кнопки:

- «Войти» – в случае корректности данных позволяет пользователю авторизоваться в системе и получить права, соответствующие его роли;
- «Создать аккаунт» – в случае, если аккаунта с таким именем не существует в системе, то создается новый, с правами роли «Турист».

Если при авторизации или создании аккаунта вводятся некорректные данные, то пользователь информируется об этом, а выполняемый процесс завершает свою работу.

На верхней линии расположены кнопки: «Карта», «Личный кабинет», «Новости», «Тропы», «Объекты».

Список объектов и маршрутов также можно посмотреть в виде таблицы при перемещении на определённую страницу, прототип которой представлен ниже. (см. рис. 5).

 <div style="float: right; background-color: blue; color: red; padding: 5px; text-align: center;">ИЗБРАННЫЕ МАРШРУТЫ</div>							
[Кнопки: Карта, личный кабинет, новости, тропы, объекты]							
<b>ФИЛЬТРЫ ПОИСКА</b>  <b>ОБЪЕКТЫ</b>  <input type="text"/> <input type="button" value="▼"/>  В ГОРОДСКОЙ МЕСТНОСТИ <input type="checkbox"/>	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">МАРШРУТЫ ДОСТУПНЫЕ НА КАРТЕ</th> </tr> <tr> <td style="text-align: center;">   <b>ФОТО</b>   <b>[НАЗВАНИЕ]</b>            ОБЪЕКТЫ НАХОДЯЩИЕСЯ            НА МАРШРУТЕ: [СПИСОК ОБЪЕКТОВ]         </td> <td style="text-align: center;">   <b>ФОТО</b>   <b>[НАЗВАНИЕ]</b>            ОБЪЕКТЫ НАХОДЯЩИЕСЯ            НА МАРШРУТЕ: [СПИСОК ОБЪЕКТОВ]         </td> </tr> <tr> <td style="text-align: center;">   <b>ФОТО</b> </td> <td style="text-align: center;">   <b>ФОТО</b> </td> </tr> </table>	МАРШРУТЫ ДОСТУПНЫЕ НА КАРТЕ		 <b>ФОТО</b>  <b>[НАЗВАНИЕ]</b> ОБЪЕКТЫ НАХОДЯЩИЕСЯ НА МАРШРУТЕ: [СПИСОК ОБЪЕКТОВ]	 <b>ФОТО</b>  <b>[НАЗВАНИЕ]</b> ОБЪЕКТЫ НАХОДЯЩИЕСЯ НА МАРШРУТЕ: [СПИСОК ОБЪЕКТОВ]	 <b>ФОТО</b>	 <b>ФОТО</b>
МАРШРУТЫ ДОСТУПНЫЕ НА КАРТЕ							
 <b>ФОТО</b>  <b>[НАЗВАНИЕ]</b> ОБЪЕКТЫ НАХОДЯЩИЕСЯ НА МАРШРУТЕ: [СПИСОК ОБЪЕКТОВ]	 <b>ФОТО</b>  <b>[НАЗВАНИЕ]</b> ОБЪЕКТЫ НАХОДЯЩИЕСЯ НА МАРШРУТЕ: [СПИСОК ОБЪЕКТОВ]						
 <b>ФОТО</b>	 <b>ФОТО</b>						

Рис. 5. Прототип интерфейса списка троп

У каждого элемента списка присутствует фотография, указанная при добавлении маршрута на карту. Если фотография не была прикреплена, то вместо неё расположено изображение, обозначающее, что пользователь не прикрепил фотографию.

Под фотографией расположено название, которое пользователь дал тропе в случае, если у маршрута нет официального названия, которое используется на картах и в справочных материалах, связанных с Пермским Краем.

Ниже расположено поле «Объекты на маршруте», после которого списком перечисляются объекты, встречающиеся на пути следования.

В левой части вкладки расположены элементы, позволяющие выводить список, отфильтрованный по нуждам пользователя. Этими элементами являются:

- «Объекты» – пользователь обозначает те объекты, которые присутствуют в поле «Объекты на маршруте» в нужных элементах списка;

У каждого зарегистрированного пользователя должен быть свой личный кабинет, прототип которого расположен ниже (см. рис. 5).

**Рис. 6. Прототип личного кабинета**

В верхней части окна справа расположена кнопка «Избранные маршруты». При нажатии на данную кнопку пользователь переносится на окно, в котором представлены туристические пути и маршруты с интерактивной карты, которые пользователь пометил «Избранными».

### **Библиографический список**

1. StarUML 2. [Электронный ресурс]. – URL: <https://staruml.io> (Дата обращения: 8.04.2023)
2. Paint.NET – Free Software for Digital Photo Editing. [Электронный ресурс]. – URL: <https://getpaint.net/index.html> (Дата обращения: 13.05.2023)
3. GIMP – GNU Image Manipulation Program. [Электронный ресурс]. – URL: <https://www.gimp.org> (Дата обращения: 13.05.2023)
4. Правительство Пермского края. Распоряжение от 1 апреля 2022 года N 82-рп. Об утверждении Стратегии развития туризма в Пермском крае на период до 2035 года. С. 16-17.

### **References**

1. StarUML 2. [Electronic resource]. – URL: <https://staruml.io>
2. Paint.NET – Free Software for Digital Photo Editing. [Electronic resource]. – URL: <https://getpaint.net/index.html>
3. GIMP – GNU Image Manipulation Program. [Electronic resource]. – URL: <https://www.gimp.org> (Дата обращения: 13.05.2023)
4. Pravitel'stvo Permskogo kraja. Rasporyazhenie ot 1 aprelya 2022 goda N 82-rp. Ob utverzhdenii Strategii razvitiya turizma v Permskom krae na period do 2035 goda. 16-17 p.(In Russ.)

## **DESIGNING AND DOCUMENTING AN INFORMATION SYSTEM « ECOLOGICAL TOURISM OF PERM REGION»**

*Nizametdinov Rostislav E.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [rostislav.nizametdinov@mail.ru](mailto:rostislav.nizametdinov@mail.ru)

*Averin Sergey I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [averin-s-i@yandex.ru](mailto:averin-s-i@yandex.ru)

The main means for designing an information system for recording tourist routes on the map of the Perm Region are considered. The object and subject of the study are determined, the relevance and tasks that need to be solved during the design of the information system are formulated. In the course of the study, an analysis of information systems with similar topics was carried out. As a result of the analysis, functional requirements for the future information system are highlighted. The analysis of tools that allow to create diagrams in the UML object language is carried out. As a result of the analysis, the StarUML application was selected. As a result of the analysis of raster graphics editors, applications were selected for the design of the user interface of the system Paint.NET and GIMP. As a result of the design, diagrams were created describing: the database of the system, the capabilities of different user groups, the operation of the function of adding a route to the map. Then the interfaces were designed.

Keywords: ecological tourism, tourism, Perm Region, design, documentation

## **ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕРКИ РАСПИСАНИЯ НА СООТВЕТСТВИЕ НОРМАМ**

*Николаев Алексей Павлович, Анисимова Светлана Игоревна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, alexey.p.nikolaev@gmail.com

Рассматривается проектирование и разработка информационной системы для проверки расписания. Так как оно должно соответствовать большому количеству норм процесс проверки является сложным. Создание данной системы должно сократить нагрузку на диспетчеров по расписанию и повысить качество расписания занятий. Для выявления требований сначала рассмотрены документы, которым должно соответствовать расписание. Затем на их основе сформулированы критерии для проверки расписания. После этого был сделан выбор инструментов для работы и описан процесс моделирования и разработки приложения. В ходе разработки было описано преимущество параллельной реализации алгоритма проверки расписания на соответствие нормам. По итогам была спроектирована и разработана информационная система, которая проверяет расписание и даёт возможность вносить изменения в него, что позволило упростить работу диспетчера по расписанию.

**Ключевые слова:** Расписание занятий, проверка, проектирование, разработка, информационная система.

Расписание занятий – это последовательность учебных занятий в образовательном учреждении. От него зависит не только порядок занятий, а ещё и распределение нагрузки на обучающихся. Нарушение распределения нагрузки может привести к серьёзным проблемам со здоровьем. Для того чтобы не допустить этого разработаны нормы, которым должно соответствовать школьное расписание. Человеку, который составляет расписание, приходится держать их всех в голове и не редки ситуации, когда он допускает ошибки.

Актуальность работы состоит в том, чтобы спроектировать и реализовать информационную систему, которая позволит автоматизировать процесс проверки расписания занятий. Это поможет снизить нагрузку на диспетчера по расписанию, а также повысить качество расписания.

На данный момент школьное расписание регламентируют четыре документа:

- СанПиН 2.4.3648-20;
- СанПиН 1.2.3685-21;
- ФГОС;
- Федеральный закон от 29 декабря 2012 г. № 273 – ФЗ «Об образовании в Российской Федерации».

Изходя из вышеперечисленных документов, а также из анализа предметной области были сформулированы следующие требования, по которым необходимо осуществлять проверку расписания:

- в одном кабинете может быть только один урок одновременно;
- у одного учителя одновременно может быть только одно занятие;
- расписание учителя должно соответствовать графику его работы;

- с первого по четвёртый класс пик нагрузки должен приходиться на второй или третий урок, а пик недельной нагрузки должен быть во вторник или среду;
- с пятого по одиннадцатый класс пик нагрузки должен приходиться на второй, третий или четвертый урок, а пик недельной нагрузки должен быть во вторник или среду;
- с первого по четвёртый класс не должно быть спаренных уроков за исключением физической культуры;
- расписание должно соответствовать учебному плану.

Затем были сформулированы функциональные требования для информационной системы:

- пользователь должен иметь возможность ввода информации необходимой для проверки информации;
- пользователь должен иметь возможность редактирования ранее введенной информации;
- пользователь должен иметь возможность загрузки расписания в систему;
- пользователь должен иметь возможность просмотра расписания и найденных ошибок;
- пользователь должен иметь возможность редактирования расписания;
- пользователь должен иметь возможность выгрузки расписания из системы.

Также было принято решение о том, что информационная система должна быть реализована в виде настольного приложения с возможностью работы на персональных компьютерах под управлением операционных систем Windows и Linux.

Далее был произведен выбор средств для реализации.

В качестве средства моделирования был выбран инструмент diagrams.net, так как он позволяет разрабатывать UML диаграммы, распространяется полностью бесплатно, а также поддерживает большое количество платформ [1].

Для прототипирования была выбрана Figma, так как она является полностью бесплатным решением и имеет возможность запуска на всех популярных операционных системах [2].

Разработка системы осуществлялась при помощи технологии Avalonia UI и языка программирования C#, так как данное сочетание позволяет разрабатывать программы под большое количество платформ и распространяется бесплатно [3].

Работа с базой данных будет осуществляться при помощи СУБД SQLite, так как она является встраиваемой, поддерживает работу на большом количестве платформ и распространяется бесплатно [4].

Далее было произведено моделирование информационных процессов. В результате было создано четыре диаграммы активности и две диаграммы последовательностей, диаграмма прецедентов, диаграмма классов.

На основе составленных моделей была произведена разработка информационной системы с использованием ранее выбранных технологий.

Ввод информации включает в себя внесение пяти разных видов информации. Как только ввод закончен, происходит проверка информации и если ошибки не были обнаружены, то происходит запись в базу данных. В случае обнаружения ошибок пользователю выводится сообщение.

Затем необходимо загрузить расписания. Изначально оно хранится в виде электронной таблицы excel. Для работы с данным файлом используется библиотека EPPlus. Данная библи-

лиотека позволяет работать с таблицами Office Open XML. Она же используется для сохранения полученного расписания [5].

После загрузки расписания или внесения изменений происходит проверка расписания по требованиям, описанным ранее.

Для ускорения работы в каждом методе используются технологии параллельного программирования. Применение методов параллельного программирования позволило уменьшить время проверки. Ниже в таблице представлено сравнение последовательного и параллельного алгоритма.

Таблица 1. Сравнение последовательного и параллельного алгоритма.

Количество классов	Время последовательного алгоритма мс.	Время параллельного алгоритма мс.
1	96,66	98,218
10	211,87	193,243
21	356,048	292,89

Как видно из представленных данных параллельная реализация алгоритма работает быстрее на больших объёмах данных.

При необходимости пользователь может отредактировать данные. Для этого пользователю необходимо выбрать вид информации, который хочет исправить. Затем внести изменения и если ошибок не обнаружено, то информация сохраняется.

Редактирование расписания начинается с ввода пользователем класса, в котором нужно внести изменения. Далее пользователь может вносить изменения, добавлять новые занятия в нужный день и удалять не нужные. По нажатию кнопки «сохранить» происходит проверка на корректность введенных данных и запись в базу. После внесения всех изменений происходит проверка всего расписания.

По окончании разработки полученная система была протестирована на реальных данных из школы. В систему успешно была внесена вся необходимая информация и загружено расписание. После проверки было обнаружено сто шестьдесят одна ошибка. По завершению редактирования количество ошибок было сокращено до девяносто одной.

По итогу работы была спроектирована и реализована система для проверки расписания, которая позволила сократить нагрузку на диспетчеров по расписанию и повысить качество расписания.

### Библиографический список

1. diagrams.net [Электронный ресурс]. – URL: <https://www.diagrams.net/> (Дата обращения 08.04.2023)
2. Figma [Электронный ресурс]. – URL: <https://www.figma.com/> (Дата обращения 13.04.2023)
3. Avalonia UI [Электронный ресурс]. – URL: <https://avaloniaui.net/> (Дата обращения 16.04.2023)
4. SQLite [Электронный ресурс]. – URL: <https://www.sqlite.org/index.html> (Дата обращения 17.04.2023)
5. EPPlus Software [Электронный ресурс]. – URL: <https://www.epplussoftware.com/> (Дата обращения 13.05.2023)

## References

1. diagrams.net [Electronic resource]. – URL: <https://www.diagrams.net/>
2. Figma[Electronic resource]. – URL: <https://www.figma.com/>
3. Avalonia UI [Electronic resource]. – URL: <https://avaloniaui.net/>
4. SQLite [Electronic resource]. – URL: <https://www.sqlite.org/index.html>
5. EPPlus Software [Electronic resource]. – URL: <https://www.epplussoftware.com/>

## DESIGN AND DEVELOPMENT OF THE INFORMATION SYSTEM FOR CHECKING THE SCHEDULE FOR COMPLIANCE WITH REGULATIONS

*Nikolaev Aleksey P., Anisimova Svetlana I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, alexey.p.nikolaev@gmail.com

The design and development of an information system for checking the schedule is considered. Since it must comply with a large number of regulations, the verification process is complex. The creation of this system should reduce the load on schedulers and improve the quality of the class schedule. To identify the requirements, we first reviewed the documents that the schedule must comply with. Then, on their basis, criteria for checking the schedule are formulated. After that, a choice of tools for work was made and the process of modeling and developing an application was described. During development, the advantage of parallel implementation of the schedule compliance check algorithm was described. As a result, an information system was designed and developed that checks the schedule and makes it possible to make changes to it, which made it possible to simplify the work of the dispatcher according to the schedule.

Keywords: Class schedule, verification, design, development, information system.



## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ МОБИЛЬНОЙ СИСТЕМЫ ОПОВЕЩЕНИЯ ЭКСТРЕННЫХ СЛУЖБ О ДОРОЖНО-ТРАНСПОРТНЫХ ПРОИСШЕСТВИЯХ**

*Перебиковский Кирилл Алексеевич, Ракина Валерия Денисовна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, meduzya2018@gmail.com

Статья посвящена проектированию и документированию мобильной системы оповещения экстренных служб о дорожно-транспортных происшествиях. Целью данной работы было спроектировать и задокументировать мобильную систему оповещения экстренных служб о дорожно-транспортных происшествиях, способную автоматизировать процесс оповещения и оперативного реагирования на дорожно-транспортные происшествия. Предметом исследования являются дорожно-транспортные происшествия и способы их оперативного реагирования на них. Объектом исследования является автоматизация процесса оповещения экстренных служб о дорожно-транспортных происшествиях. Решенные задачи – анализ литературы по теме, выбор подходящих технологий для разработки системы, проектирование архитектуры, интерфейса и функциональности системы. В статье рассматриваются основные этапы проектирования системы, такие как анализ требований, разработка архитектуры и проектирование интерфейса. В статье также приведены примеры UML-диаграмм, использованных в документации. Исследование предназначено для студентов и преподавателей, интересующихся информационными системами и разработкой мобильных приложений.

Ключевые слова: Проектирование, дорожно-транспортные происшествия, мобильная система, архитектура.

### **Анализ предметов области мобильной системы оповещения экстренных служб о ДТП**

В данном исследовании был подробно рассмотрен один из докладов всемирной организации здравоохранения за 2018 год о состоянии безопасности дорожного движения [1], а также был изучен ряд тем, связанных с организацией Государственной инспекции безопасности дорожного движения (ГИБДД), включая структуру организации [2-4]. Был основательно рассмотрен процесс, включающий реагирование сотрудников ГИБДД на случаи дорожно-транспортных происшествий (ДТП), процедуру получения оповещений о ДТП, составление соответствующих протоколов и документов, таких как протокол ДТП и Европротокол, а также оформление ДТП со стороны сотрудников ГИБДД [5].

В рамках исследования был осуществлен анализ имеющегося программного обеспечения, которое может быть использовано в качестве инструмента для оперативного реагирования на случаи ДТП [6-8]. Особое внимание уделено изучению системы под названием "Аварийный вызов" от компании "ГЛОНАСС". Был представлен подробный обзор принципов работы мобильной системы, а также сформулированы необходимые требования к мобильной системе, которая могла бы быть использована в контексте оперативного реагирования на ДТП.

## Выбор технологий

Для проектирования архитектуры системы мобильного оповещения экстренных служб о ДТП был выбран фреймворк Flutter, написанный на языке программирования Dart. Он обеспечивает разработку кроссплатформенных приложений с высокой производительностью, имеет богатый набор компонентов для создания функционального интерфейса, а также имеет собственный движок для рендеринга [9]. Для интеграции с картографическим сервисом был выбран API Яндекс.Карты, поскольку у него отсутствует ограничение по запросам. Была выбрана база данных (БД) Realm, потому что она очень быстрая при чтении и записи данных т.к. использует сжатый бинарный формат и хранит данные в памяти [10-11].

## Проектирование системы

Для отображения основных взаимодействий объектов в системе были построены UML-диаграммы. Диаграмма прецедентов (см. рис. 1) позволяет видеть систему с точки зрения пользователя и описывает ее функциональность. В данном случае, на диаграмме присутствуют два актера: Пользователь и Оператор. Пользователь имеет возможность вызвать экстренные службы, авторизоваться и зарегистрироваться в системе, просматривать историю, получать рекомендации по первой помощи, заполнять отчеты о ДТП, изменять личные данные и добавлять/изменять медицинские данные. Оператор может обрабатывать полученные отчеты, осуществлять звонки пользователям и направлять экстренные службы в соответствии с предоставленной информацией.

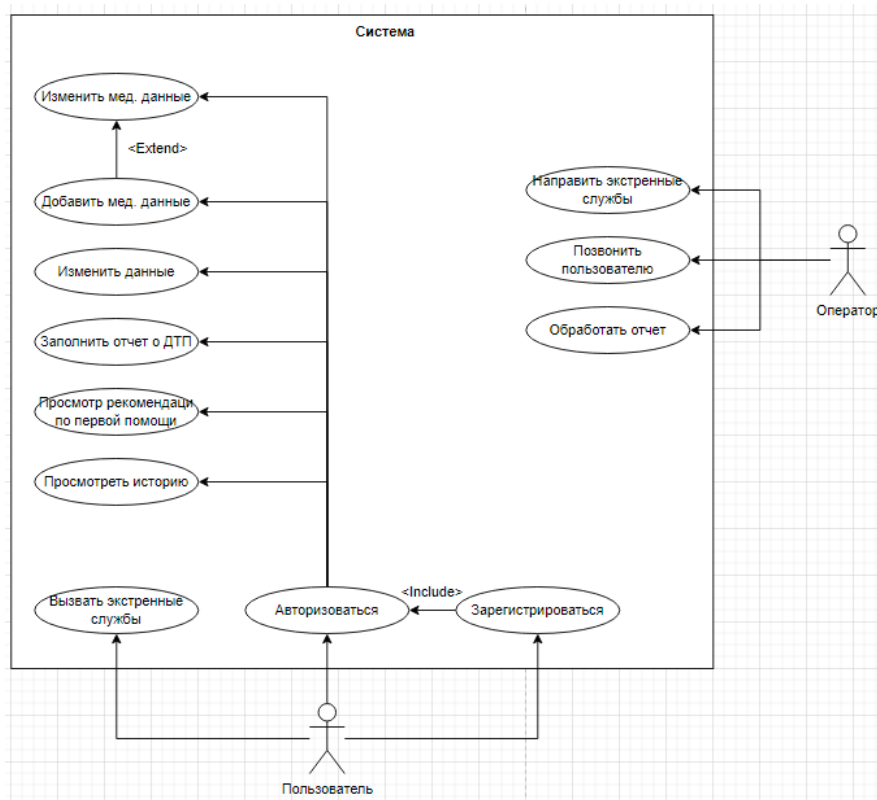


Рис. 1. Диаграмма прецедентов

Для визуализации конкретного случая использования системы была построена диаграмма деятельности (см. рис. 2) [12]. Пользователь авторизуется, после чего система проверяет данные, и, если они верные, пользователь заполняет отчет о ДТП и отправляет его, в обратном случае, у пользователя есть возможность экстренно вызвать экстренные службы. По-

сле заполнения и отправки отчета о ДТП, система обрабатывает отчет в базе данных, а оператор, получив уведомление, анализирует отчет и звонит пользователю для уточнения информации. Затем оператор направляет необходимые службы на основе решения.

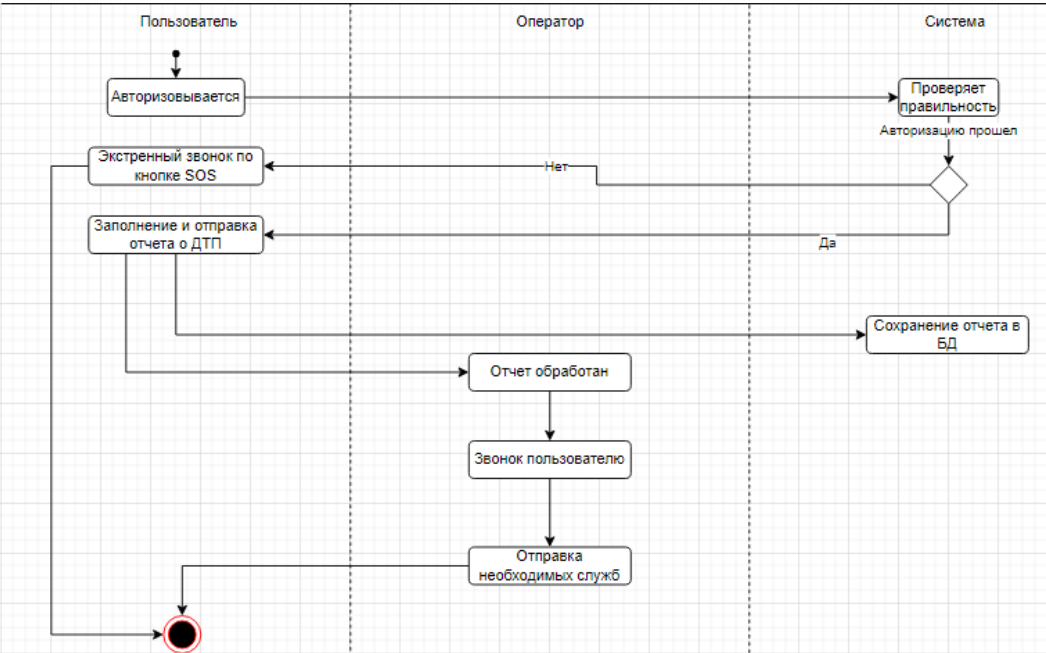


Рис. 2. Диаграмма деятельности

Визуализация структуры базы данных (БД) с типами данных и отображение связей между сущностями отображены на рис. 3.

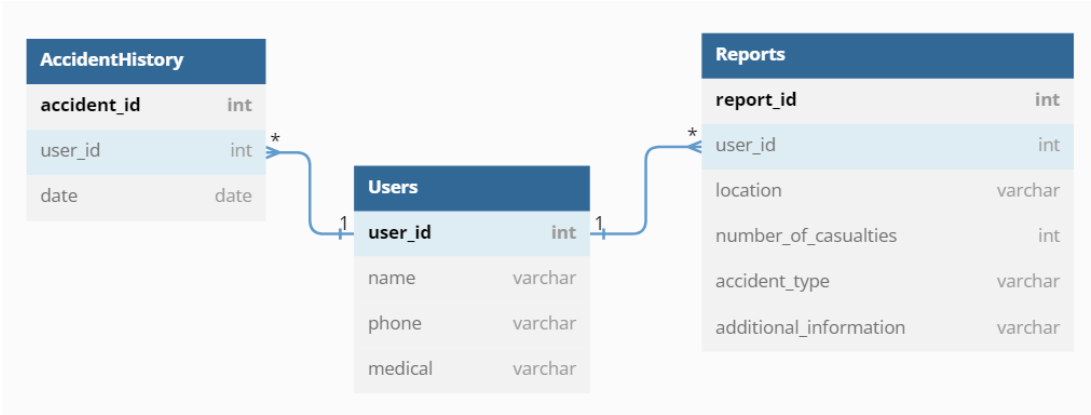
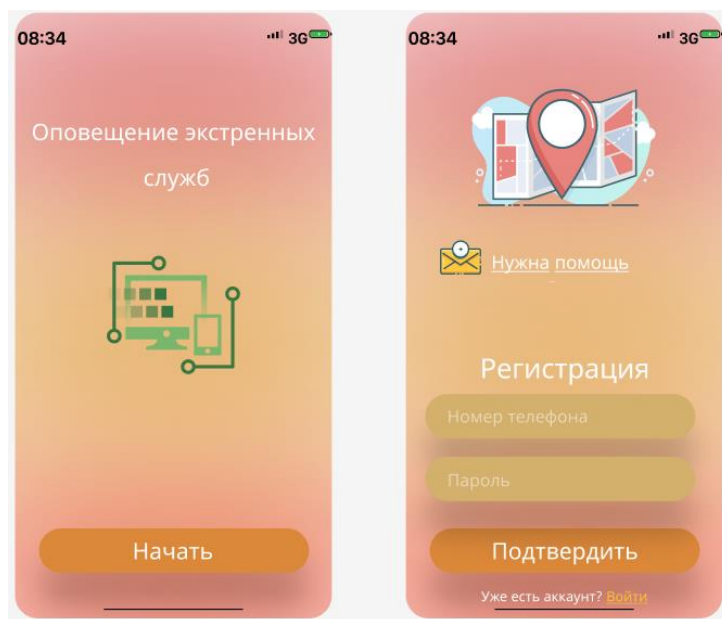


Рис. 3. Визуализация базы данных

В БД 3 сущности: пользователь, отчеты, история ДТП. Такая структура позволяет связать отчеты и историю ДТП с конкретным пользователем посредством внешнего ключа “user\_id”. Также у каждой сущности есть свои собственные атрибуты, что позволяет хранить и управлять специфичными данными для каждой сущности отдельно.

**Макет интерфейса**

Данный фрагмент макета интерфейса (см. рис. 4) отражает первоначальный вход пользователя в приложение.



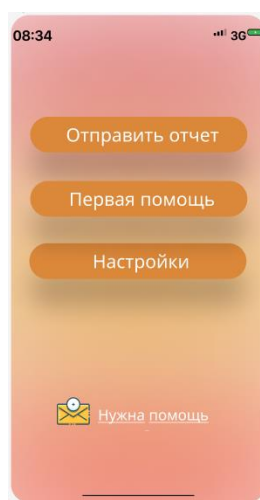
**Рис. 4. Вход в приложение**

Пользователь при входе в приложение видит кнопку “Начать”, которая отправит пользователя в меню регистрации / авторизации. Если пользователю нужна помощь, но он не зарегистрирован/авторизован – он может нажать на кнопку экстренного вызова “Нужна помощь”.

После успешной авторизации пользователь видит следующее меню (см. рис. 5). На данном макете можно наблюдать 4 кнопки. Первая кнопка “Отправить отчет” – перенесет пользователя в меню заполнения отчета. Вторая кнопка “Первая помощь” – отобразит рекомендации об оказании первой помощи пострадавшему по различным происшествиям. Третья кнопка “Настройки” – в настройках можно редактировать свои медицинские данные, а также просматривать историю ДТП.

Аналогично стартовому экрану, здесь можно экстренно начать звонок с помощью кнопки “нужна помощь” для незамедлительного соединения с экстренными службами.

Первая кнопка “Отправить отчет”, перенесет пользователя в форму заполнения отчета о ДТП (см. рис. 6).



**Рис. 5. Главный экран**

В отчете можно заполнить адрес происшествия, или же нажать на звезду справа от соответствующего поля. Звезда перенесет нас на интерактивную карту, где можно указать свое

местоположение. Чуть ниже заполняется такая информация, как: кол-во пострадавших, уточняется является ли сам пользователь пострадавшим, тип происшествия. Ниже располагается текстовое окно, куда вписываются оставшиеся переменные, так или иначе связанные с ДТП.



Рис. 6. Заполнение отчета

### Библиографический список

1. Global status report on road safety 2018 // Всемирная организация здравоохранения [Электронный ресурс]. – URL: <https://www.who.int/publications/i/item/9789241565684> (дата обращения: 19.04.2023).
2. Закон Российской Федерации "Об обязательном страховании гражданской ответственности владельцев транспортных средств" от 25.04.2002 № 40-ФЗ // Официальный интернет-портал правовой информации. – с изм. и допол. в ред. от 28.12.2022.
3. Акт правительства Российской Федерации "Об утверждении Правил оснащения транспортных средств категорий М2, М3 и транспортных средств категории N, используемых для перевозки опасных грузов, аппаратурой спутниковой навигации" от 22.12.2020 № 2216 // Официальный интернет-портал правовой информации. – 2021
4. Закон Российской Федерации "О внесении изменений в Федеральный закон "Об обязательном страховании гражданской ответственности владельцев транспортных средств" и отдельные законодательные акты Российской Федерации" от 04.07.2014 № 223-ФЗ // Собрание законодательства Российской Федерации.
5. Типовые схемы оформления ДТП // ИНГОССТРАХ [Электронный ресурс]. – URL: <https://www.ingos.ru/auto/osago/skhemy-oformlenija-dtp> (дата обращения: 01.05.2023).
6. Как выбрать мобильную кросс-платформу в 2021 году // Habr [Электронный ресурс]. – URL: <https://habr.com/ru/companies/tinkoff/articles/544228/> (дата обращения: 05.05.2023).
7. KMP vs Flutter vs React Native // Habr [Электронный ресурс]. – URL: <https://habr.com/ru/articles/645611/> (дата обращения: 05.05.2023).

8. Flutter Vs React Native Vs Kotlin: Which One to Choose? // Binary Informatics [Электронный ресурс]. – URL: <https://binaryinformatics.com/flutter-vs-react-native-vs-kotlin/> (дата обращения: 07.05.2023).
9. Flutter по сравнению с Kotlin в 2021 году: что лучше для разработки кроссплатформенных приложений? // Habr [Электронный ресурс]. – URL: <https://habr.com/ru/companies/otus/articles/595451/> (дата обращения: 07.05.2023).
10. System Properties Comparison Firebase Realtime Database vs. Realm vs. SQLite // DB-Engines [Электронный ресурс]. – URL: <https://db-engines.com/en/system/Firebase+Realtime+Database%3bRealm%3bSQLite> (дата обращения: 11.05.2023).
11. Comparing SQLite (Room & Core Data) and Realm // MongoDB [Электронный ресурс]. – URL: <https://www.mongodb.com/compare/sqlite-vs-realm> (дата обращения: 11.05.2023).
12. Простое руководство по диаграммам активности UML // Creatly [Электронный ресурс]. – URL: <https://creately.com/blog/ru/uncategorized-ru/%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA-%D0%BF%D0%BE-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B5-%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D0%B8/> (дата обращения: 12.05.2023).

### References

1. Global status report on road safety 2018 // World Health Organization [Electronic resource]. – URL: <https://www.who.int/publications/i/item/9789241565684> (Accessed: April 19, 2023).
2. Law of the Russian Federation "On compulsory motor third-party liability insurance" dated April 25, 2002, No. 40-FZ // Official Internet Portal of Legal Information. – as amended on December 28, 2022.
3. Government Decree of the Russian Federation "On Approval of the Rules for Equipping Vehicles of Categories M2, M3, and Vehicles of Category N Used for the Carriage of Dangerous Goods with Satellite Navigation Equipment" dated December 22, 2020, No. 2216 // Official Internet Portal of Legal Information. – 2021.
4. Law of the Russian Federation "On Amendments to the Federal Law 'On compulsory motor third-party liability insurance' and certain legislative acts of the Russian Federation" dated July 4, 2014, No. 223-FZ // Collection of Laws of the Russian Federation.
5. Typical accident documentation schemes // Ingosstrakh [Electronic resource]. – URL: <https://www.ingos.ru/auto/osago/skhemy-oformleniya-dtp> (Accessed: May 1, 2023).
6. How to choose a mobile cross-platform framework in 2021 // Habr [Electronic resource]. – URL: <https://habr.com/ru/companies/tinkoff/articles/544228/> (Accessed: May 5, 2023).
7. KMP vs Flutter vs React Native // Habr [Electronic resource]. – URL: <https://habr.com/ru/articles/645611/> (Accessed: May 5, 2023).
8. Flutter Vs React Native Vs Kotlin: Which One to Choose? // Binary Informatics [Electronic resource]. – URL: <https://binaryinformatics.com/flutter-vs-react-native-vs-kotlin/> (Accessed: May 7, 2023).
9. Flutter compared to Kotlin in 2021: what's better for cross-platform app development? // Habr [Electronic resource]. – URL: <https://habr.com/ru/companies/otus/articles/595451/> (Accessed: May 7, 2023).

10. System Properties Comparison Firebase Realtime Database vs. Realm vs. SQLite // DB-Engines [Electronic resource]. – URL: <https://db-engines.com/en/system/Firebase+Realtime+Database%3bRealm%3bSQLite> (Accessed: May 11, 2023).
11. Comparing SQLite (Room & Core Data) and Realm // MongoDB [Electronic resource]. – URL: <https://www.mongodb.com/compare/sqlite-vs-realm> (Accessed: May 11, 2023).
12. Simple Guide to UML Activity Diagrams // Creatly [Electronic resource]. – URL: <https://creately.com/blog/ru/uncategorized-ru/%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA-%D0%BF%D0%BE-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B5-%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D0%B8/> (Accessed: May 12, 2023).

## **DESIGN AND DOCUMENTATION OF A MOBILE SYSTEM FOR NOTIFICATION OF EMERGENCY SERVICES OF ROAD ACCIDENTS**

*Perebikovskii Kirill A., Rakina Valeria D.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, meduzya2018@gmail.com

The article is devoted to the design and documentation of a mobile system of notification of emergency services of road accidents. The purpose of this work was to design and document a mobile system of notification of emergency services of road accidents, capable of automating the process of notification and rapid response to road accidents. The subject of the study is road accidents and how to respond to them promptly. The object of the study is to automate the process of notification of emergency services of road accidents. Problems solved – the analysis of the literature on the subject, the choice of suitable technologies for the development of the system, designing the architecture, interface and functionality of the system. The article discusses the main stages of system design such as requirements analysis, architecture design and interface design. The paper also provides examples of UML diagrams used in the documentation. The study is intended for students and teachers interested in information systems and mobile application development.

**Keywords:** Design, traffic accidents, mobile system, architecture.

## **РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ СИНТЕЗА И РЕДАКТИРОВАНИЯ ИЗОБРАЖЕНИЙ С ПРИМЕНЕНИЕМ МОДЕЛЕЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

*Писцов Сергей Михайлович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, spistsov@gmail.com

Работа посвящена применению нейронных сетей на примере синтеза и редактирования изображений предметов одежды. Целью является создание инструмента, который позволит пользователям легко редактировать изображения с предметами одежды без необходимости понимания принципов работы нейронных сетей. Рассмотрены различные архитектуры и модели генерации изображений, выбраны наиболее оптимальные методы для редактирования изображения с учетом простоты взаимодействия пользователя с системой, а также реалистичности и соответствия ожиданиям пользователей итоговых изображений. В результате была выбрана модель генерации изображений Stable Diffusion и модель CLIP Interrogator для упрощения составления текстового описания итогового изображения пользователем. Проведен анализ и проектирование информационной системы (ИС) синтеза и редактирования изображения, также выполнено макетирование веб-приложения и выбор инструментов реализации. С учетом выдвинутых требований была реализована система на клиент-серверной архитектуре с возможностью гибкой взаимозаменяемости вычислительных ресурсов. Клиентская часть реализована на основе JavaScript-библиотеки React, серверная часть реализована с применением веб-фреймворка FAST API на языке Python. Тестирование системы проводилось на различных реализациях модели Stable Diffusion и различных параметрах. Для оценки результатов тестирования был проведен опрос с целью определения оптимальных параметров для получения наиболее реалистичных и соответствующих описанию изображений с точки зрения различных пользователей.

Ключевые слова: нейронные сети, веб-приложение, предмет одежды, редактирование изображений, текстовое описание, информационная система, проектирование системы, реализация приложения, реалистичные изображения, заменимость вычислительных ресурсов, оптимальные параметры.

### **Введение**

На данный момент область искусственного интеллекта продолжает стремительно развиваться, в особенности предоставляется большой спектр возможностей по части синтеза изображений. Большим потенциалом обладают нейронные сети для обработки естественного языка. Однако до сих пор существует множество отраслей, в которых не исследовано потенциальное применение нейронных сетей.

Специалисты по проектированию дизайна одежды используют графические редакторы для создания рисунка будущего продукта, для чего требуются определенные навыки, опыт, время, способность передать суть образа в изображении. Для непрофессионального пользователя воплощение своей идеи становится проблемой, для специалиста остается проблема временного ресурса.

Решением проблемы простого переноса идеи пользователя на изображение может стать использование нейронных сетей, способных редактировать и синтезировать изображения на



основе текстового описания. Пользователю важен удобный и понятный инструмент для работы с отдельными предметами одежды и передачи сути образа.

Целью работы является анализ, проектирование, разработка и тестирование веб-приложения для генерации предметов одежды людей на основе текстовых подсказок при помощи искусственных нейронных сетей.

### **Обзор методов генерации изображений**

Модель нейронной сети – архитектура, конфигурации и алгоритм обучения нейронной сети [12].

Обученная модель нейронной сети – набор весов, полученных в результате обучения модели нейронной сети на некоторой выборке.

Рассматриваются две модели нейронных сетей в контексте редактирования изображений: генеративно-состязательная (Generative Adversarial Network) [1] и диффузная модель (Denoising Diffusion Probabilistic Models, дословно – вероятностные модели рассеяния шума) [2]. Описаны принципиальные схемы работы данных моделей.

Текстовая подсказка – репрезентация изображения на естественном языке.

Текстовая подсказка рассматривается как достаточно простой способ управления процессом генерации изображений и передачи сути запроса пользователя. Однако составление правильной подсказки зачастую приводит к длительному подбору подходящей конструкции текста, ключевых слов и фраз. В качестве решения проблемы решено использовать модель CLIP Interrogator, позволяющую получить описание исходного изображения на естественном языке. Такой подход предоставляет наиболее подходящую конструкцию подсказки, пользователю остается лишь модифицировать ее с учетом своего запроса.

Рассматриваются различные методы редактирования изображения. Метод «img2img» предполагает редактирование исходного изображения при помощи текстовой подсказки. Метод «inpaint» позволяет редактирование отдельную область исходного изображения, выделенную маской. Метод «инструкции» также позволяет редактированию конкретную часть изображения, однако выделение и желаемый образ достаточно описать в подсказке.

Рассмотрены существующие генеративно-состязательные и диффузные модели редактирования изображения, выделены их преимущества и недостатки в контексте работы с предметами одежды. Выбрана предпочтительная модель Stable Diffusion [3], которая выделяется реалистичностью и детализацией итоговых изображений, а также широким спектром реализаций под различные задачи и поддержкой всех перечисленных методов генераций. Пример редактирования изображения приведен на рис. 1.



**Рис. 1. Результат редактирования входного изображения при помощи текстовой подсказки «yellow hoodie», модель «hardblend inpaint» (реализация Stable Diffusion)**

## **Анализ и проектирование информационной системы**

С учетом описанных моделей и методов была проанализирована соответствующая предметная область и спроектирована информационная система (ИС) на основе UML-диаграмм. Определены базовые понятия, на их основе создана онтология предметной области. В процессе этапа концептуализации были определены нефункциональные требования к системе:

1. ИС должна иметь возможность гибкой заменимости вычислительных ресурсов,
2. ИС должна поддерживать возможность одновременной работы множества пользователей,
3. приложение должно иметь клиент-серверную архитектуру,
4. пользовательский интерфейс должен быть удобным и понятным,

Также выделены и классифицированы прецеденты, определены функциональные требования. Алгоритм действий пользователя для получения итоговой изображения включает следующие этапы:

1. загрузить начальное изображение,
2. выделить контур предмета одежды на начальном изображении,
3. запросить список доступных инструментов генерации;
4. запросить текстовое описание выделенного изображения;
5. на основе описания выделенного и начальной идеи изображения создать текстовую подсказку для генератора изображений;
6. запросить список доступных моделей генерации изображения;
7. на основе выделенного, начального изображений и созданной подсказки создать запрос на генерацию изображения.

Идентифицированы атрибуты понятий; выделены ассоциации, отношения понятий, системные операции, классы предметной области. Результат этапов анализа и проектирования: диаграммы прецедентов, активностей, понятий, последовательностей, классов.

Для макетирования интерфейса приложения использовался сервис Figma. Взаимодействие пользователя с интерфейсом отражает диаграмма последовательностей.

## **Выбор архитектуры информационной системы и инструментов реализации**

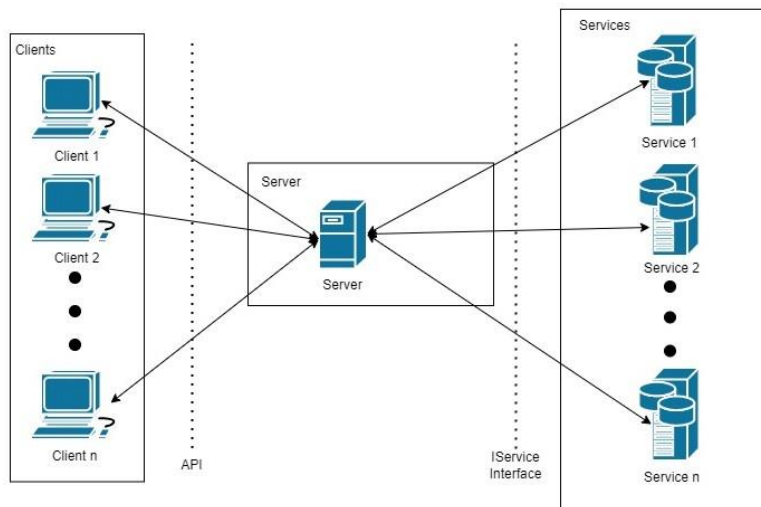
Спроектированная система имеет клиент-серверную архитектуру. Для реализации клиентской части приложения выбрана JavaScript-библиотека React, для серверной части приложения выбран фреймворк FAST API на языке Python.

Выбор архитектуры учитывает требование гибкой заменимости вычислительных ресурсов. Генерация изображений – ресурсоемкий процесс. Современные модели требуют производительные графические карты с достаточно большим объемом видеопамяти. В процессе эксплуатации системы появляется проблема переноса вычислений на другую машину и использования новых моделей нейронных сетей. Взаимозаменяемость вычислительных ресурсов позволяет легко обновлять систему при увеличении нагрузки и появлении более требовательных к ресурсам графических карт моделей.

Инструмент генерации – экземпляр программы, сервис или другие вычислительные ресурсы, которые позволяют генерировать описание изображения и сами изображения с помощью нейронных сетей.

Инструменты могут представлять собой аутсорсинговые и краудсорсинговые сервисы, а также предоставляющие API интерфейс приложения. В рамках работы рассматривается сервис AI Horde и интерфейс stable-diffusion-webui от пользователя Automatic1111 (далее под

термином Automatic1111 будет пониматься конкретно данный интерфейс). Таким образом серверная часть приложения включает отправку запросов клиентской части на генерацию изображения или описания изображения на выбранный инструмент генерации. Высокоуровневая схема взаимодействия компонентов архитектуры системы приведена на рис. 2.



**Рис. 2. Высокоуровневая схема взаимодействия компонентов архитектуры системы.**

Для добавления поддержки нового инструмента администратору достаточно добавить класс соответствующего инструмента, который реализует интерфейс `IService`. Данный интерфейс обеспечивает единообразный способ взаимодействия с различными вычислительными ресурсами. На текущий момент реализована поддержка инструментов Automatic1111 и AI Horde.

### Реализация и тестирование

Пользовательский интерфейс приложения реализован в соответствии с макетом, пример генерации изображения и описания начального изображения приведен на рис. 3.



**Рис. 3. Экран пользовательского интерфейса после генерации описания и итогового изображения**

Интерфейс включает описание последовательности действия для получения итогового изображения. Имеется возможность редактирования предметов одежды на изображении путем выделения области изображения инструментом «Лассо» и задании подсказки на основе сгенерированного описания (реализация метода «inpaint»), а также путем выделения области изображения в текстовой подсказке (реализация метода «инструкция»). Пользователь имеет возможность редактировать следующие параметры генерации:

1. высота,

2. ширина,
3. шаги генерации,
4. схожесть,
5. влияние подсказки.

Для упрощения взаимодействия с интерфейсом, оптимальные параметры, а также инструмент генерации и модель выбираются автоматически. Для реализации многопользовательского взаимодействия с системой была интегрирована распределенная очередь асинхронных задач Celery.

Тестирование системы включает подбор оптимальных параметров генерации изображения на примере модели «hardblend inpaint», а также проверку гипотезу о необходимости подбора оптимальных параметров для других моделей, которые имеют базовую архитектуру Stable Diffusion. Для подбора оптимальных параметров модели был сгенерирован ряд сеток изображений при различных значениях описанных параметров. Оценивание результатов тестирования проводилось на основе опроса среди студентов ПГНИУ. Основные критерии оценки: реалистичность, соответствие текстовой подсказке. В результате опроса были определены оптимальные параметры генерации изображений для модели «hardblend inpaint»:

1. схожесть – 10,
2. шаги генерации – 30,
3. влияние подсказки – не имеет значения.

Также для проверки гипотезы о необходимости подбора оптимальных параметров для других моделей в опрос были включены сетки результатов генерации изображений моделью «Instruct Pix2Pix». Опрос показал, что оптимальные параметры на примере данной модели отличаются от оптимальных параметров модели «hardblend inpaint». Таким образом в задачи администратора помимо реализации поддержки новых инструментов генерации входит подбор оптимальных параметров для различных моделей генерации изображений.

### **Заключение**

В ходе работы были выполнены все поставленные задачи. Определена оптимальная модель редактирования предметов одежды Stable Diffusion, решена проблема создания текстовой подсказки путем генерации описания исходного изображения для получения шаблона конструкции подсказки. Проведен анализ предметной области, спроектирована информационная система, создан макет, выбраны инструменты реализации веб-приложения. Выбрана архитектура системы с возможностью гибкого обновления системы путем легкой замены вычислительных ресурсов. Создана информационная система для легкого редактирования предметов одежды на изображения. Проведено тестирование системы с целью определения оптимальных параметров генерации изображений.

### **Библиографический список**

1. *Goodfellow I.* Generative Adversarial Networks / I. Goodfellow, J. Pouget-Abadie, M. Mirza // arXiv preprint arXiv: 1406.2661 – 2019. P. 1–3.
2. *Sohl-Dickstein J.* Deep Unsupervised Learning using Nonequilibrium Thermodynamics / J. Sohl-Dickstein Eric, A. Weiss, N. Maheswaranathan // arXiv preprint arXiv: 1503.03585 – 2015. P. 1–7.
3. *Rombach R.* High-Resolution Image Synthesis with Latent Diffusion Models / R. Rombach, A. Blattmann, Dominik // arXiv preprint arXiv: 2112.10752 – 2021. P. 1–4.

## References

1. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative Adversarial Networks. 2014. Available online: <https://arxiv.org/abs/1406.2661> (accessed on 25 July 2023).
2. Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; Ganguli, S. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics; 2015. Available online: <https://arxiv.org/abs/1503.03585> (accessed on 25 July 2023).
3. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High-Resolution Image Synthesis with Latent Diffusion Models; 2022. Available online: <https://arxiv.org/abs/2112.10752> (accessed on 25 July 2023).

## DEVELOPMENT OF A SOFTWARE SYSTEM FOR IMAGE SYNTHESIS AND EDITING USING ARTIFICIAL INTELLIGENCE MODELS

*Pistsov Sergey M.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [spistsov@email.ru](mailto:spistsov@email.ru)

The work is dedicated to the application of neural networks in the synthesis and editing of clothing images. The goal is to create a tool that allows users to easily edit images with clothing items without the need to understand the principles of neural networks. Various architectures and image generation models have been examined, the most optimal methods for image editing have been selected, taking into account user interaction simplicity, realism, and alignment with users' expectations of the final images. As a result, the Stable Diffusion image generation model and the CLIP Interrogator model were chosen to facilitate user-generated textual descriptions of the resulting images. The analysis and design of an information system (IS) for image synthesis and editing were conducted, along with the layout of the web application and the selection of implementation tools. Considering the requirements, a system was implemented using a client-server architecture with flexible interchangeability of computational resources. The client-side was developed using the JavaScript library React, while the server-side was implemented using the FAST API web framework in Python. System testing was conducted on different implementations of the Stable Diffusion model and various parameters. To evaluate the testing results, a survey was conducted to determine the optimal parameters for generating the most realistic and description-aligned images from the perspective of different users.

**Keywords:** neural networks, web application, clothing item, image editing, textual description, information system, system design, application implementation, realistic images, interchangeability of computational resources, optimal parameters.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ "РАДИО ПГУ"

*Полынский Арсений Андреевич, Ракина Валерия Денисовна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, apolynskiy@mail.ru

В данной статье рассмотрены аналоги информационных систем радиостанций, выделены преимущества и недостатки этих систем, и на их основе выявлены и определены функциональные требования к проектируемой информационной системе (ИС). ИС планируется использовать для трансляции вещания "Радио ПГУ" в реальном времени, предоставления возможности общения между ведущими и слушателями, просмотра и публикации материалов "Радио ПГУ". Для моделируемой ИС были определены средства для проектирования, набор инструментальных средств для разработки, а также средство хранения данных. Помимо этого, был разработан прототип интерфейса ИС. Работа содержит описание логики работы информационной системы, а также примерный прототип интерфейса ИС. Спроектированная система будет интересна радиостанциям различных организаций (на примере ПГНИУ).

Ключевые слова: информационная система (ИС), онлайн-вещание, "Радио ПГУ", мобильное приложение, uml-диаграммы.

### **Введение**

Информационные системы играют важную роль для радиостанций, так как они позволяют оперативно получать и обрабатывать информацию, необходимую для работы радиостанции. Не менее важным является возможность радиостанции вещать беспрепятственно, эффективно и на большую аудиторию. Одним из возможных способов обеспечить такое вещание является Интернет-радио [1]. Также для радиостанции важно иметь возможность общения со своими слушателями. Радиостанцией, наиболее тесно связанной с ПГНИУ, несомненно, является "Радио ПГУ". На данный момент, "Радио ПГУ" вещает только на территории кампуса и не имеет никакой системы для общения со слушателями в реальном времени. В связи с этим, создание информационной системы для "Радио ПГУ" является актуальной задачей. "Радио ПГУ" также имеет собственный веб-сайт, который уже представляет собой информационную систему. В академических целях было решено разработать отдельную систему, а не модифицировать существующую. Таким образом, было решено проектировать мобильное приложение "Радио ПГУ".

Цель данной работы состоит в проектировании и документировании информационной системы для "Радио ПГУ".

### **Анализ предметной области**

"Радио ПГУ" – студенческая радиостанция, ведущая своё вещание на базе ПГНИУ. "Радио ПГУ" предлагает своим слушателям разнообразную музыку, новости из жизни университета, разнообразные программы и интервью. Одним из главных преимуществ "Радио ПГУ" является то, что оно ведется студентами, что позволяет ему быть ближе к аудитории и понимать ее потребности. Кроме того, радиостанция является отличной площадкой для сту-

дентов, желающих получить опыт ведения радиопередач, работы в редакции и организации мероприятий.

Для разработки ИС для "Радио ПГУ" следует пообщаться непосредственно с работниками радиостанции и определиться с форматом проектируемой системы. Для ускорения процесса, было решено сначала продумать основные функции системы, после чего пообщаться с представителем "Радио ПГУ" для получения комментариев и пожеланий. В ходе беседы было решено проектировать мобильное приложение и был утверждён список задач и требований к проектируемой ИС, представленный в следующей главе.

### **Анализ и сравнение существующих ИС**

Рассмотрим и сравним функционал существующих мобильных приложений радиостанций.

#### **1. Maximum**

Мобильное приложение радиостанции «Maximum» [2]. Приложение обладает минималистичным дизайном, в функционал входит возможность прослушивания эфира и различных плейлистов, составленных ди-джеями радиостанции. Также присутствует возможность просмотра плейлиста за день.

#### **2. НАШЕ Радио**

Мобильное приложение радиостанции «Наше радио» [3]. Следует заметить, что фактически оно является также приложением таких радиостанций как «Rock FM», «Радио Jazz» и «Радио Ultra», входящих в состав одного радиовещательного холдинга. В функционал приложения входит возможность прослушивания эфира, просмотра новостей, прослушивания подкастов. Также имеется возможность просмотра частот вещания радиостанции в разных городах и списка имеющихся на данный момент рубрик. Имеются мобильный телефон и ссылки на контакты радиостанции в социальных сетях.

#### **3. Silver Rain**

Мобильное приложение радиостанции «Серебряный дождь» [4]. В функционал приложения входит возможность прослушивания эфира в разных городах, просмотра новостей и информации о событиях, связанных непосредственно с радиостанцией, прослушивания плейлистов, составленных ди-джеями. Также имеются ссылки непосредственно на веб-сайт радиостанции, на страницы с расписанием эфира и описанием плейлистов, на контакты радиостанции в социальных сетях и мессенджерах. Помимо этого, в приложении имеется возможность включить режим «Экономии трафика», поделиться ссылкой на само приложение и связаться с технической поддержкой.

Исходя из вышенаписанного и обсудив необходимый функционал с представителями "Радио ПГУ", были установлены основные задачи и требования к разрабатываемой ИС:

- возможность пользователя прослушивать эфир в реальном времени;
- возможность пользователя просматривать новости, касающиеся ПГНИУ и непосредственно «Радио ПГУ»;
- возможность пользователя прослушивать записи подкастов;
- возможность пользователя просматривать список прозвучавших композиций (плейлист);
- система чатов между слушателями и пользователями;
- хранение в базе данных информации о пользователях ИС, чатах и материалах "Радио ПГУ".

### **Выбор case-средства, средств разработки и хранения данных**

Для проектирования информационных систем целесообразно использовать программные средства, которые поддерживают объектно-ориентированный подход и включают CASE-средства. Один из основных инструментов – язык моделирования UML, который используется для создания диаграмм, необходимых для выполнения данной курсовой работы.

StarUML [5] – бесплатный инструмент для моделирования, который позволяет создавать диаграммы классов, диаграммы последовательностей, диаграммы состояний и многие другие типы диаграмм. Интерфейс StarUML понятен и удобен. Взяв во внимание перечисленные преимущества, в качестве средства проектирования был выбран инструмент StarUML.

Рассмотрим средства разработки самого приложения. Xamarin [6] – фреймворк для разработки кроссплатформенных мобильных приложений, который позволяет использовать язык программирования C# и платформу .NET. Он позволяет разработчикам создавать приложения для платформ iOS, Android и Windows, используя общий код. Помимо этого, язык C#, на котором работает фреймворк Xamarin, обладает большим сообществом разработчиков и интеграцией с платформой .NET.

В качестве средства хранения данных была выбрана PostgreSQL [7] – удобная объектно-реляционная система управления базами данных.

### **Проектирование информационной системы**

В соответствии с поставленными задачами, следует определиться с ролями участников ИС и их правами.

Любой пользователь, независимо от того, зарегистрирован и авторизован он в ИС или нет, обладает возможностью прослушивать эфир в реальном времени (если эфир идёт в данный момент), просматривать новости, прослушивать записи подкастов и просматривать актуальные плейлисты.

В свою очередь, любой авторизованный пользователь получает права слушателя. Слушатель имеет возможность отправки сообщений непосредственно ведущему (одному или нескольким), который находится в эфире в данный момент, если такая возможность была разрешена самим ведущим. Помимо этого, слушатель может редактировать собственный профиль на минимальном необходимом уровне: изменить имя пользователя, фотографию профиля и привязанный к профилю адрес электронной почты.

Ещё одна категория пользователей – ведущие. Ведущий обладает всеми правами слушателя. Основные особые функции ведущий получает при включении режима «В эфире». Имена ведущих, включивших режим «В эфире» отображаются на главной странице. Ведущий, находясь в эфире, имеет возможность разрешить или запретить отправки сообщений себе и своим соведущим. Из этого вытекает, что ведущий также может эти сообщения просматривать. В целях обеспечения возможности некоторой модерации сообщений, ведущий может временно или навсегда блокировать получение сообщений от конкретных слушателей. Ведущим также доступна функция создания и редактирования материалов, таких как новости, подкасты и плейлисты.

Роль, обладающая наибольшим набором прав – администратор. Администратор определяет, кто из пользователей может обладать правами ведущего и сам обладает ими. Ролью администратора одновременно может обладать только пользователь, и при необходимости роль может быть передана другому пользователю, при условии обладания того правами ведущего.



Диаграмма прецедентов, описывающая данную систему ролей, представлена ниже (см. рис. 1).

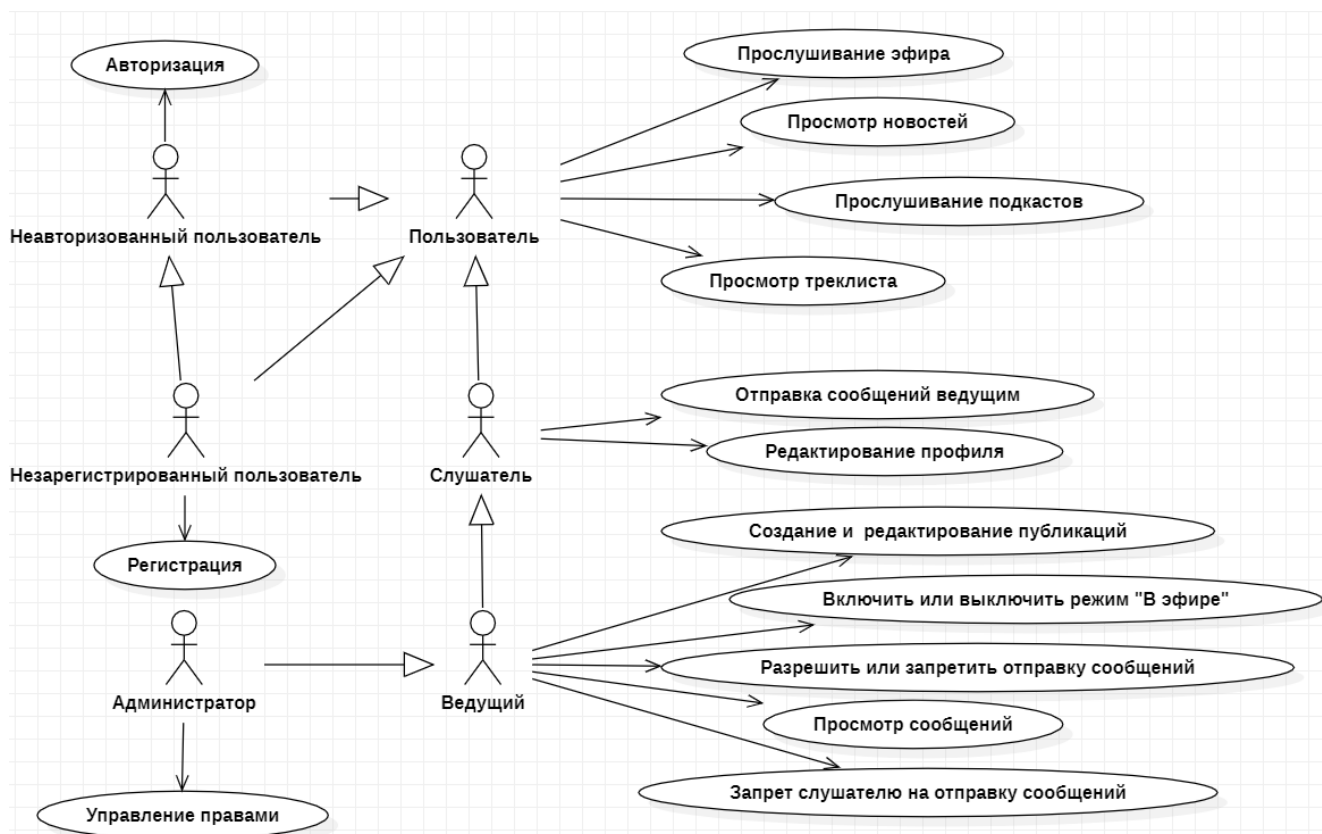


Рис. 1. Диаграмма прецедентов, описывающая ролевую систему

С точки зрения теории [8], для функционирования интернет-радио нужны три компонента: станция, сервер и клиент. Станция генерирует аудиопоток и направляет его серверу. Сервер принимает аудиопоток от станции и перенаправляет его копии всем подключённым клиентам, реплицирует данные. Клиент принимает аудиопоток от сервера и преобразует его в аналоговый аудиосигнал. Диаграмма развёртывания, иллюстрирующая инфраструктуру компонентов ИС, представлена ниже (см. рис. 2). В данном случае в качестве станции используется ПК с установленным соответствующим ПО (таким, как SAM Broadcaster, RadioBoss или другое в зависимости от потребностей "Радио ПГУ"), в качестве сервера – ПК с установленным дистрибутивом IceCast (также популярным программным решением является ShoutCast, предоставляющийся преимущественно на платной основе) и дистрибутивом СУБД PostgreSQL, под клиентом же подразумевается смартфон пользователя с подключением к сети Интернет и установленным приложением "Радио ПГУ". Аудиопоток вместе с метаданными об эфире генерируется на станции и передаётся серверу по протоколу HTTPS. Сервер ретранслирует поток для всех клиентов и обрабатывает запросы к базе данных. Далее копии аудиопотока также передаются по протоколу HTTPS всем подключенным клиентам, где декодируются и преобразуются непосредственно в аналоговый аудиосигнал (см. рис. 2).

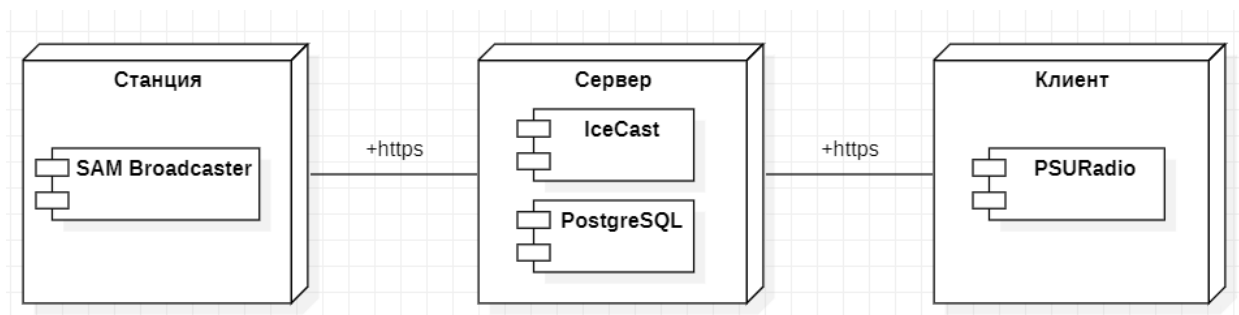


Рис. 2. Диаграмма развёртывания

Логика базы данных описана с помощью ER-диаграммы, представленной ниже (см. рис. 3). Были созданы 5 таблиц, описывающие основные сущности, которые будут храниться в базе данных.

Таблица users. Представляет собой хранилище всех пользователей, независимо от их прав. Содержит поля id – суррогатный первичный ключ, username – имя пользователя, login – логин, password – хешированный и «посоленный» пароль [9], pass\_salt – «соль» пароля, email – электронная почта, role – роль пользователя в системе (слушатель, ведущий или администратор).

Таблица chats. Представляет собой хранилище чатов. Содержит поля id – суррогатный первичный ключ, sender\_id – ID слушателя-отправителя, внешний ключ, отсылающийся к таблице users, receiver\_id – ID ведущего-получателя, внешний ключ, отсылающийся к таблице users, messages – массив строк, содержащий сообщения слушателя.

Таблица news. Представляет собой хранилище новостей. Содержит поля id – суррогатный первичный ключ, title – заголовок, text – сам текст новости, img\_path – путь к файлу с иллюстрирующим изображением, date – дата публикации, poster\_id – ID ведущего, который опубликовал новость, внешний ключ, отсылающийся к таблице users.

Таблица podcasts. Представляет собой хранилище подкастов. Содержит поля id – суррогатный первичный ключ, title – название, file\_path – путь к самому аудиофайлу с подкастом, img\_path – путь к файлу с иллюстрирующим изображением, date – дата публикации, poster\_id – ID ведущего, который опубликовал подкаст, внешний ключ, отсылающийся к таблице users.

Таблица playlists. Представляет собой хранилище плейлистов. Содержит поля id – суррогатный первичный ключ, date – дата, в которую плейлист проигрывался, songs – список песен, poster\_id – ID ведущего, который опубликовал плейлист, внешний ключ, отсылающийся к таблице users

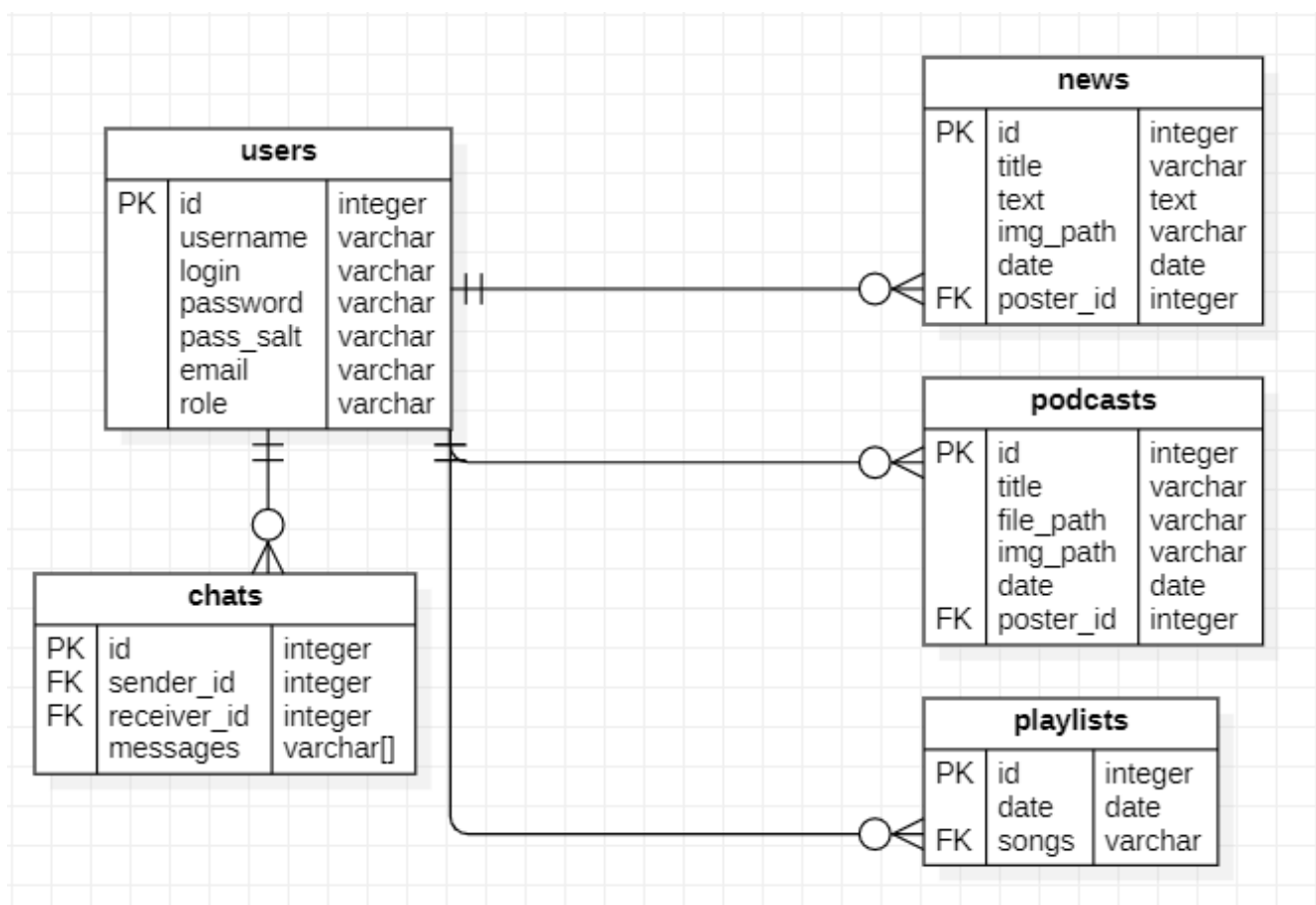


Рис. 3. ER-диаграмма БД

Ниже представлен макет главной страницы ИС (см. рис. 4).

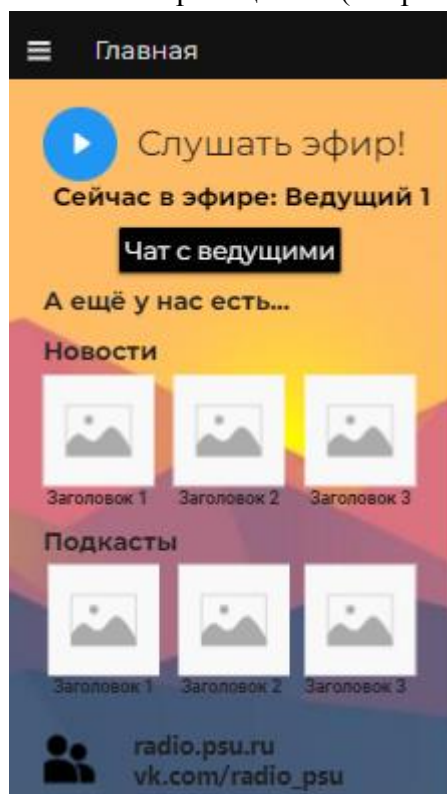


Рис. 4. Главная страница

На этой странице пользователь может начать прослушивание прямого эфира (при условии его активности) и увидеть имена ведущих, которые в данный момент находятся в эфире. Если пользователь нажмет кнопку "Чат с ведущими", система отобразит список всех ведущих, присутствующих в эфире в данный момент (если их несколько), и после выбора ведущего перенаправит пользователя на страницу диалога с выбранным ведущим.

Кроме того, на главной странице содержится информация о последних новостях и подкастах. Также имеются ссылки на веб-сайт "Радио ПГУ" и группу в социальной сети, которые пользователь может использовать для получения дополнительной информации и взаимодействия с сообществом "Радио ПГУ".

Ниже представлен макет страницы с меню приложения для авторизованного слушателя (см. рис. 5)

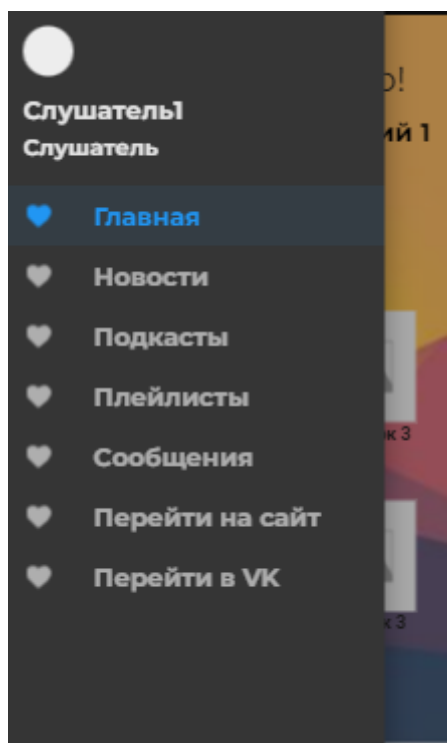
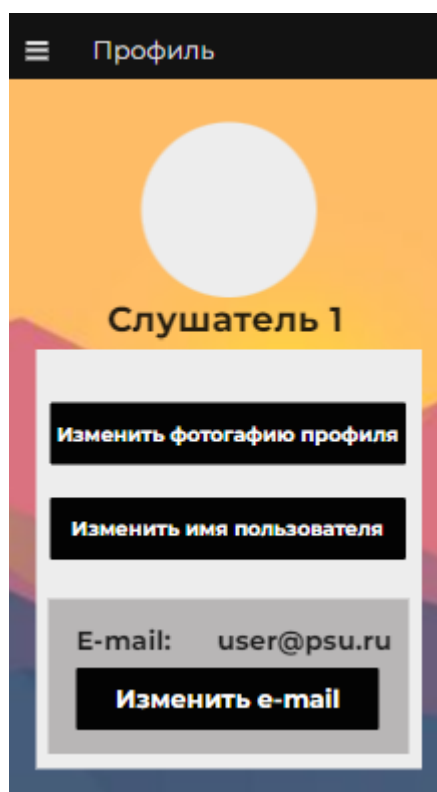


Рис. 5. Меню слушателя

При открытии меню авторизованным слушателем, появляется панель, которая содержит информацию о его профиле. В этой панели отображается имя пользователя, указание на его роль, фотография профиля, а также список доступных страниц, включая: "Главная", "Новости", "Подкасты", "Плейлисты", "Сообщения", "Перейти на сайт" и "Перейти в VK".

Ниже представлена страница профиля, отображаемая при нажатии авторизованным пользователем, находящимся в меню, на собственную фотографию профиля или имя пользователя (см. рис. 6).



**Рис. 6. Страница профиля**

На странице своего профиля пользователь видит увеличенную версию своей фотографии профиля, имя пользователя и привязанный к его профилю e-mail. Пользователь может изменить свою фотографию профиля, имя (но не логин) и привязанную к аккаунту электронную почту. При изменении фотографии профиля, открывается проводник, позволяющий найти фотографию на устройстве. При изменении имени пользователя, появляется поле для ввода текста. При изменении e-mail, привязанного к аккаунту, появляется поле для ввода текста и на сам e-mail отправляется письмо для подтверждения.

### **Заключение**

Спроектированная система будет полезна для представителей коллектива "Радио ПГУ", так как она позволяет транслировать интернет-вещание и обеспечивает общение со слушателями в реальном времени. Также следует отметить, что спроектированная система повторяет функционал веб-сайта в необходимом объёме, позволяя слушателям просматривать материалы и публикации, создаваемые корреспондентами и редакторами "Радио ПГУ".

### **Библиографический список**

1. Интернет-радио как новый тип медиа и основа новых дискурсивных практик // Elibrary [Электронный ресурс]. – URL: <https://elibrary.ru/item.asp?id=20744086> (дата обращения 21.05.2023)
2. Радио MAXIMUM [Электронный ресурс]. – URL: <https://www.maximum.ru/> (дата обращения: 5.05.2023)
3. Наше Радио [Электронный ресурс]. – URL: <https://www.nashe.ru/> (дата обращения 5.05.2023)
4. Приложение Серебряного Дождя // Серебряный дождь [Электронный ресурс]. – URL: <https://www.silver.ru/about/apps/> (дата обращения 5.05.2023)

5. StarUML // StarUML [Электронный ресурс]. – URL: <https://staruml.io> (дата обращения 12.05.2023)
6. Что такое Xamarin? // Microsoft [Электронный ресурс]. – URL: <https://learn.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin> (дата обращения 12.05.2023)
7. PostgreSQL // PostgreSQL [Электронный ресурс]. – URL: <https://www.postgresql.org> (дата обращения 12.05.2023)
8. Интернет-радио глазами современника // Elibrary [Электронный ресурс]. – URL: <https://elibrary.ru/item.asp?id=21623268> (дата обращения 18.05.2023)
9. Про хранение паролей в БД // Хабр [Электронный ресурс]. – URL: <https://habr.com/ru/companies/acribia/articles/413157/> (дата обращения 18.05.2023)

### References

1. Elibrary. Online radio as a new type of media and the basis of new discursive practices. [Electronic resource]. – URL: <https://elibrary.ru/item.asp?id=20744086> (In Russ).
2. Radio MAXIMUM [Electronic resource]. – URL: <https://www.maximum.ru/> (In Russ).
3. Nashe Radio [Electronic resource]. – URL: <https://www.nashe.ru/> (In Russ).
4. Silver.ru. Prilozhenie Serebryanogo Dozhdy [Electronic resource]. – URL: <https://www.silver.ru/about/apps/> (In Russ).
5. StarUML. [Electronic resource]. – URL: <https://staruml.io>.
6. Microsoft. Chto takoe Xamarin? [Electronic resource]. – URL: <https://learn.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin> (In Russ).
7. PostgreSQL. [Electronic resource]. – URL: <https://www.postgresql.org>.
8. Elibrary. Internet-radio glazami sovremennika. [Electronic resource]. – URL: <https://elibrary.ru/item.asp?id=21623268> (In Russ).
9. Habr. Pro hranenie parolei v BD. [Electronic resource]. – URL: <https://habr.com/ru/companies/acribia/articles/413157/> (In Russ).

## DESIGN AND DOCUMENTATION OF INFORMATION SYSTEM FOR "RADIO PSU"

*Polynskii Arsenii A., Rakina Valeria D.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [apolynskiy@mail.ru](mailto:apolynskiy@mail.ru)

This article examines analogs of radio station information systems, identifies the advantages and disadvantages of these systems, and based on them, determines the functional requirements for the designed information system (IS). The IS is planned to be used for real-time broadcasting of "Radio PSU," providing the ability for interaction between hosts and listeners, as well as viewing and publishing materials from "Radio PSU." For the simulated IS, design tools, a set of development tools, and a data storage facility have been identified. In addition, a prototype interface for the IS has been developed. The paper includes a description of the logic of the information system's operation, as well as an approximate prototype of the IS interface. The designed system will be of interest to radio stations of various organizations (using PSU as an example).

**Keywords:** information system (IS), online broadcasting, "Radio PSU," mobile application, UML diagrams.

## ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ СБОРА WHOIS ДАННЫХ ДОМЕННЫХ ИМЕН И ИХ АНАЛИЗА

*Пономарев Александр Федорович, Кнутова Наталия Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, airsunday2001@gmail.com

Доменные имена стали неотъемлемой частью бизнеса и IT-сферы, позволяя компаниям быть более доступными для клиентов и партнеров. Выбор и регистрация доменного имени играют важную роль в привлечении аудитории и продвижении бренда. Сложно найти свободное и привлекательное доменное имя, поэтому анализ рынка, регистраторов и серверов важен для успешной регистрации. Отслеживание доменных имен имеет несколько ролей, включая защиту бренда, поиск информации о конкурентах и поиск новых возможностей для бизнеса. Однако, существующие системы для отслеживания доменов имеют ограничения и устаревшую информацию, поэтому требуется создание удобной системы для анализа доменных имен. Эта работа целится на проектирование и реализацию системы сбора WHOIS данных и их анализ, включая обзор существующих систем, определение характеристик, разработку модели и интерфейса, программирование, внедрение и эксплуатацию.

Ключевые слова: домен, whois, анализ, информационная система, сбор данных.

Интернет за последнее время стал настолько разнообразным, очень сложно выбрать интересное и привлекательное имя для своего домена, которое еще не будет занято. Доменное имя важная составляющая сайта для привлечения и сохранения своих пользователей. Чтобы подобрать удачный домен, а также правильно его зарегистрировать, важно анализировать рынок, регистраторов и серверов, где чаще размещают домены.

Регистрация доменного имени не всегда является легкой задачей. Недобросовестные пользователи могут зарегистрировать доменное имя, которое совпадает с уже зарегистрированным в другой доменной зоне и принадлежит другому владельцу или организации, даже без их согласия. Также возможно зарегистрировать освобождающийся домен сразу же после того, как он будет освобожден. В ряде случаев, возможна регистрация домена на основании ранее зарегистрированного бренда. В таком случае, владелец прав на доменное имя может столкнуться с серьезными проблемами, такими как потеря брендовой репутации, потеря клиентов и партнеров, а также финансовые потери.

Для предотвращения таких нежелательных ситуаций была разработана система whois, которая позволяет собирать информацию о владельцах доменных имен, их контактных данных и других сведений, связанных с регистрацией доменов. Главной целью системы whois является обеспечение открытости и прозрачности владения доменными именами в Интернете. Она позволяет любому пользователю получать информацию о владельцах доменных имен, контактных данных, а также других сведений, связанных с регистрацией доменов. Это может быть полезно для того, чтобы связаться с владельцем домена или его администратором, решить спорные вопросы в случае нарушения прав, или проверить подлинность домена. Кроме того, система whois используется для контроля за регистрацией доменов и защиты от киберпреступлений.

Whois представляет из себя сетевой протокол прикладного уровня, базирующийся на 43-ем порту протокола TCP. Эти сервисы активно используются потребителями, государственными учреждениями и правоохранительными органами для связи с правообладателем доменного имени и определения источника продуктов и услуг, продвигаемых в интернете, а также для определения статуса домена.

Необязательно whois может предоставить точные данные настоящего владельца. Для защиты личной информации владельцев доменных имен существуют услуги скрытия (приватизации) whois-данных. Это позволяет скрыть контактную информацию владельца домена, такую как имя, адрес, электронная почта и номер телефона, от поисковых систем и других пользователей интернета. При использовании услуги скрытия whois-данных, контактная информация заменяется на контактные данные компании-регистратора, что обеспечивает конфиденциальность владельца домена. Однако в случае, если правоохранительные органы, регистратор или судебные органы потребуют предоставления контактной информации владельца домена, компания-регистратор может предоставить ее на основании соответствующих законов и правил ICANN [1].

Рассмотрения нарушений прав третьего лица администратором доменного имени, а также других доменных споров, в рамках традиционного судебного разбирательства, являются достаточно длительной и дорогостоящей процедурой. Для преодоления этих трудностей, как у нас в стране, так и за рубежом были предприняты попытки в разработке упрощенного механизма, компетентно разрешающего эти споры. Так, наиболее удачным решением, принявшее широкое распространение, стала система, разработанная ВОИС и принятая ICANN в 1999 году [2], основанная на двух нормативных документах: Единая Политика Рассмотрения Споров о Доменных Именах (Uniform Domain Name Dispute Resolution Policy или UDRP) [3] и Правила для Единой Политики Рассмотрения Споров о Доменных Именах [4]. В российском национальном сегменте сети интернета решение таких споров регулируется, как правило, решением арбитражного суда, отказавшись дать свое согласие на урегулирование доменных конфликтов посредством ICANN.

Когда доменное имя освобождается, можно его зарегистрировать и оформить на свое имя. Обычно, когда домен истекает и не продлевается его владельцем, он переходит в состояние "свободен". Это означает, что домен может быть зарегистрирован кем-то другим. Чтобы зарегистрировать освободившийся домен, необходимо следить за его статусом и ожидать, пока он станет доступен для регистрации. Как правило, после истечения срока регистрации домена, он проходит несколько стадий, прежде чем стать доступным для новой регистрации. Эти стадии могут изменяться в зависимости от доменной зоны и правил регистратора.

Обычно после окончания срока регистрации домена, его владелец имеет некоторое время для продления регистрации до окончательного освобождения домена. Если владелец не продлевает регистрацию домена, то домен переходит в состояние "просрочен" и может быть освобожден для регистрации другими пользователями. Когда домен становится доступен для регистрации, его можно зарегистрировать через регистратора доменных имен, который предоставляет доступ к доменным зонам.

Для русских национальных зон RU и РФ, когда на одно и то же доменное имя регистратор имеет несколько заявок, то принимается та, для которой раньше всех были соблюдены необходимые условия, и, если такие заявки существуют, то регистратор также не вправе производить регистрацию на собственное имя. Несмотря на успешность регистрации доменного имени, которая наступает с момента внесения соответствующих сведений в реестр, личная информация пользователя не разглашается.



Важно отметить, что удалось найти немного рабочих систем проводящих анализ изменений корневой российской зоны ru. Сервис, анализирующий корневые доменные зоны, например, может отслеживать изменения в зоне и определять доменные имена, которые были добавлены или удалены. Эта информация может быть полезна для поиска новых имен и анализа тенденций в использовании доменных имен. Отсутствие рабочих и бесплатных аналогов подтверждает необходимость создания информационной системы отслеживания доменных имен и их анализа.

Целевая аудитория – это, вероятно, профессиональные пользователи, которые работают в сфере IT. Некоторые потенциальные пользователи могут включать:

- Веб-разработчики, которые хотят найти свободные доменные имена для своих клиентов или проектов;
- Маркетологи, которые ищут доменные имена для своих кампаний;
- Эксперты по SEO, которые ищут доменные имена, которые могут помочь улучшить поисковую оптимизацию;
- Доменные трейдеры, которые ищут свободные доменные имена, чтобы купить и продать их.

Потребности и предпочтения целевой аудитории могут включать:

- Быструю и точную информацию об освобождающихся доменных именах и их статусах;
- Удобный интерфейс, который позволяет быстро и легко искать доменные имена;
- Инструменты для анализа корневых зон и доменных имен;
- Уведомления об освобождении доменного имени.

Ранее был произведен анализ аналогов разрабатываемой системы [5]. Сайты были нагружены различным контентом, усложняющим навигацию. Представленные аналитические данные, а также выдаваемая по домену информация, были тяжелыми для восприятия. Основываясь на анализе конкурентов, можно сделать вывод, что необходимо продумать удобное предоставление пользователю данных.

Чтобы не переполнять веб-страницу контентом, из-за чего пользователь может потеряться в представленном функционале, было решено ввести несколько разделов, где наполнение страниц будет структурировано. На сайте будут реализовано 4 страницы:

- Главная страница, где будет представлена основная информация о проекте, а также контактных данных поддержки;
- Страница для поиска whois данных доменных имен;
- Страница с анализом по доменной зоне ru;
- Страница новостей проекта.

Для реализации всего потенциала системы, лучший выбор – создание веб-приложения с использованием клиент-серверной архитектуры, чтобы реализовать работу с базами данных и передачу файлов посредством HTTP/HTTPS. Для управления информацией воспользуемся архитектурой REST, предназначенной для распределенных систем и являющейся очень простым решением для этих задач.

Для серверной части был выбран Node.js и его фреймворке для веб-приложений Express. Node.js – это JavaScript-окружение, построенное на движке Chrome V8. Разработка приложения с использованием этого языка будет простой и гибкой. Язык хорошо работает с REST API архитектурой, а также, благодаря множеству библиотек, отлично интегрируется с БД.

Для разработки клиентской части был использован: язык гипертекстовой разметки HTML5 (HyperText Markup Language), язык стилей CSS (Cascading Style Sheets) и язык программирования JavaScript. Выбор фреймворка для клиентской части остановился на Vue. Простота работы и подробная документация сделают разработку веб-приложения более продуктивной.

Для управления передачами запросов между клиентом и сервером, а также для подключения ssl сертификата, был использован nginx. Выбор данного HTTP-сервера зависит от многих факторов, включая производительность, безопасность, простоту настройки, наличие дополнительных функций и т.д.

Чтобы разместить веб приложение на сервере, а также для локального тестирования работы сервера, было использовано программное обеспечение для развертывания и управления проектом – Docker. Также для работы с базами данных используется СУБД PostgreSQL.

Разработка и внедрение информационной системы проходило в три этапа.

Первый этап – определение и нормализация базы данных. Это важный этап, на котором определяются необходимые таблицы, их поля и связи. Здесь важно разработать правильную структуру базы данных, которая соответствует требованиям приложения, так как это сильно влияет на производительность и работоспособность приложения. Важным аспектом этого этапа является нормализация базы данных, которая помогает избежать дублирования данных и обеспечить целостность и надежность базы данных.

Второй этап – создание серверной и клиентской частей. На этом этапе создаются серверные и клиентские приложения, которые обеспечивают взаимодействие между пользователем и сервером.

Третий этап – развертывание системы на удаленном сервере. На этом этапе необходимо развернуть приложение на удаленном сервере, чтобы пользователи могли им пользоваться.

Анализируя статистику доменной зоны ru, система получила результаты по следующим позициям: анализ распределения возрастов доменов, анализ NS-серверов и регистраторов. Анализ распределения возрастов доменов, получилось данная статистика:

- от 0 до 1 года: 39%;
- от 2 до 3 лет: 15%;
- от 4 до 6 лет: 18%;
- от 7 до 9 лет: 6%;
- от 10 до 12 лет: 8%;
- от 13 лет: 14%.

Эта статистика может быть использована для анализа того, как долго домены остаются активными и как быстро новые домены заменяют старые. Например, большинство доменов имеют возраст менее 1 года, что может указывать на то, что новые домены появляются достаточно быстро, а старые домены быстро удаляются или заменяются. С другой стороны, домены с возрастом от 4 лет составляют значительную долю (около 46%), что может указывать на то, что многие домены используются в течение длительного времени. Эта информация также может быть использована для анализа изменений в тенденциях регистрации доменных имен.

Далее обратимся ко второму анализу, где представлены NS-сервера с самым большим числом размещенных доменов:

- reg: 19%;
- beget: 13%;

- nic: 8%;
- timeweb: 8%;
- cloudflare: 5%;
- Другие: 47%.

Фаворитом являются сервера с доменом reg, принадлежащие сервису reg.ru. Третий анализ отображает самых популярных регистраторов в зоне ru:

- REGRU-RU: 43%;
- RU-CENTER-RU: 22%;
- R01-RU: 6%;
- BEGET-RU: 6%;
- SALENAMES-RU: 5%;
- Другие: 18%.

Явным фаворитом в выборе регистратора домена в зоне ru является REGRU-RU.

Система ежедневно формировала статистику изменения количества доменных имен в зоне. Последняя зафиксированная статистика:

- Было добавлено 14903 домена;
- Было удалено 6626 домена;
- Было изменено 51339 домена;
- Общее количество доменов в зоне ru 4942492.

Хочу отметить, что это достаточно большие изменения за один день. В основном количество добавленных доменов в день не превышает 5 тысяч, удаленные обычно также не превышают 5 тысяч, измененные же занимают промежуток 15 – 30 тысяч.

Система сильно нагружает сервер, для оптимизации необходим либо более мощный сервер, либо несколько серверов с распределенной нагрузкой. Для улучшения системы хорошим вариантом было бы второе решение проблемы. Также возникает проблема с получением whois данных из-за ограничения на количества запросов. Вариант решения – распределить нагрузку на ip адрес увеличением количества серверов, стал бы отличным способом улучшить систему. Все эти усовершенствования требуют дополнительных вложений, однако дают более мощную и быструю систему.

Анализ данных, производимый системой, ограничивается корневой зоной ru. Для усовершенствования системы логичным стало бы увеличение охвата анализируемых доменов. Например, в русском сегменте можно взять для анализа домены рф, su, москва, их количество не превышает миллиона и не станет сильно нагружать систему. Однако добавление других корневых зон может доставить осложнений и сильных затрат. Например, в зоне com находится более 156 миллионов доменов, что в более 30 раз больше, чем в ru.

### **Библиографический список**

1. Icann. Принципы единых правил рассмотрения споров о доменных именах (UDRP) (Принципы) Icann [Электронный ресурс]. – URL: [https:// www.icann.org/resources/pages/udrp-rules-2015-03-12-ru](https://www.icann.org/resources/pages/udrp-rules-2015-03-12-ru) (дата обращения: 3.03.2023).
2. Icann. Archived Rules for Uniform Domain Name Dispute Resolution Policy Icann [Электронный ресурс]. – URL: <http://archive.icann.org/en/dndr/udrp/uniform-rules-24oct99-en.htm> (дата обращения: 20.03.2023).

3. Icann. Принципы рассмотрения споров о единообразных доменных имен [Электронный ресурс]. – URL: <https://www.icann.org/resources/pages/policy-2012-02-25-en> (дата обращения: 20.03.2023).

4. Icann. Правила рассмотрения споров о доменных именах Icann [Электронный ресурс]. – URL: [icann.org/resources/pages/dndr-2012-02-25-ru](https://www.icann.org/resources/pages/dndr-2012-02-25-ru) (дата обращения: 20.03.2023).

5. Пономарев А. Ф., Кнутова Н. С. Проектирование системы сбора whois данных доменных имен и их анализа актуальные проблемы // Актуальные проблемы математики, механики и информатики 2022: Пермь: ПГНИУ, 2022. С. 96-99. [Электронный ресурс]. – URL: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematiki-mekhaniki-informatiki-2022.pdf> (дата обращения: 17.05.2023).

### References

1. Icann. Rules for Uniform Domain Name Dispute Resolution Policy (UDRP) [Electronic resource]. – URL: <https://www.icann.org/resources/pages/udrp-rules-2015-03-12-ru> (Accessed 3 March 2023).

2. Icann. Archived Rules for Uniform Domain Name Dispute Resolution Policy Icann [Electronic resource]. – URL: <http://archive.icann.org/en/dndr/udrp/uniform-rules-24oct99-en.htm> (Accessed 20 March 2023).

3. Icann. Uniform Domain Name Dispute Resolution Policy [Electronic resource]. – URL: <https://www.icann.org/resources/pages/policy-2012-02-25-en> (Accessed 20 March 2023).

4. Icann. Domain Name Dispute Resolution Policies [Electronic resource]. – URL: [icann.org/resources/pages/dndr-2012-02-25-ru](https://www.icann.org/resources/pages/dndr-2012-02-25-ru) (Accessed 20 March 2023).

## DESIGNING AND DEVELOPMENT OF A SYSTEM FOR COLLECTING WHOIS DOMAIN NAME DATA AND ANALYZING THEM

*Ponomarev Alexander F., Knutova Nataliya S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [airsunday2001@gmail.com](mailto:airsunday2001@gmail.com)

Domain names have become an integral part of the business and IT industry, enabling companies to be more accessible to their customers and partners. The choice and registration of a domain name play a crucial role in attracting an audience and promoting a brand. Finding an available and appealing domain name can be challenging, which is why market analysis, registrars, and servers are important for successful registration. Tracking domain names serves several purposes, including brand protection, competitor research, and discovering new business opportunities. However, existing domain tracking systems have limitations and outdated information, necessitating the development of a user-friendly system for domain name analysis. This project aims to design and implement a system for collecting WHOIS data and analyzing it, including reviewing existing systems, defining characteristics, developing a model and interface, programming, implementation, and operation.

Keywords: domain, whois, analysis, information system, data collection.

## РАЗРАБОТКА NO-CODE СИСТЕМЫ ОБРАБОТКИ И ХРАНЕНИЯ ДАННЫХ

*Русаков Вадим Вячеславович<sup>1</sup>, Мокунева Валерия Евгеньевна<sup>2</sup>, Селетков Илья Павлович<sup>3</sup>*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, Vadim.rusakov.02@email.ru<sup>1</sup>, mokuneva.valeria@gmail.com<sup>2</sup>,  
iseletkov@gmail.com<sup>3</sup>

Рассматривается процесс создания системы обработки и анализа данных. Был проведён анализ систем-конкурентов и выявлены их сильные и слабые стороны. Разработана структурная диаграмма системы в целом, которая включает в себя описание взаимодействия фронтенда и бэкенда по протоколу HTTP. Для взаимодействия пользователя с системой разработан web-интерфейс. Подключены и настроены внешние сервисы для хранения пользовательских файлов и метаданных о них.

Ключевые слова: обработка данных, фронтенд, бэкенд, микросервис, pandas

### **Введение**

На сегодняшний день в мире генерируется огромное количество данных, но далеко не все из них одинаково полезны и важны. Анализ данных позволяет выделить полезную информацию из больших объемов данных, выявить скрытые зависимости и тенденции, предсказать будущие события и принимать обоснованные решения. Однако анализ данных может быть трудоемкой задачей и требовать значительных знаний в области программирования и статистики, что является проблемой для экспертов-аналитиков, не обладающих такими навыками.

Создание универсальной системы хранения, обработки и анализа данных для специалистов без знаний программирования является эффективным способом решения этой проблемы.

### **Взаимодействие пользовательской и серверной части**

Взаимодействие пользовательской и серверной частей по протоколу HTTP основывается на модели клиент-сервер, где обе части обмениваются запросами и ответами на них. Процесс обмена сообщениями позволяет web-интерфейсу получать данные из внешнего хранилища или отправлять их на обработку на сервере.

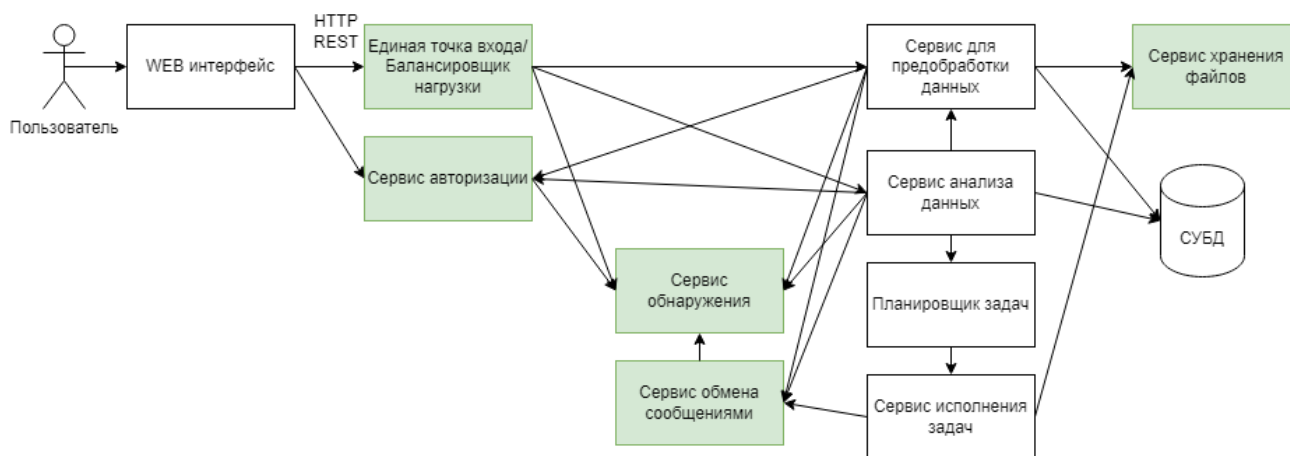


Рис. 1. Структура системы

### Архитектура серверной части

В качестве архитектуры для серверной части системы анализа данных выбрана микросервисная архитектура [1].

Микросервисы – это архитектурный подход, при котором распределенная система состоит из нескольких небольших сервисов, которые являются отдельными приложениями, работающими в собственных процессах [1].

Микросервис состоит из 3 слоёв [2]:

1 слой – Контроллер. Это класс с методами, вызываемыми из web-интерфейса по протоколу HTTP. Контроллер выполняет ряд операций, таких как аутентификация пользователя, авторизация доступа к данным, валидация запроса и т.д. После выполнения этих операций контроллер определяет, какой сервис должен обработать запрос.

Кроме того, контроллер может выполнять функцию балансировки нагрузки и масштабирования сервисов в зависимости от изменяющейся нагрузки на систему. Это может помочь обеспечить высокую производительность и доступность сервисов.

2 слой – Сервисы работы с данными и обработки наборов данных.

Сервис работы с данными обеспечивает ряд методов для выполнения различных операций с данными, которые могут быть использованы другими сервисами в системе. Он играет важную роль в управлении данными и другими сервисами в системе.

Другой важный сервис – это сервис обработки наборов данных. Он обеспечивает преобразование наборов данных, а также их обработку.

Обработка наборов данных происходит с помощью библиотеки Pandas языка программирования Python. Где наборы данных представляются во внутреннем формате -dataframe.

Ниже будут перечислены некоторые основные методы данного сервиса:

- удаление столбцов;
- удаление строк по критериям;
- замена категориальных признаков численными значениями;
- создание новых столбцов с помощью пользовательской математической формулы
- замена пропущенных значений средними по столбцам или константами.

3 слой – Классы репозитории для взаимодействия с сервисом хранения файлов и СУБД. Класс репозиторий для взаимодействия с сервисом хранения файлов в микросервисной архитектуре обеспечивает связь между микросервисом работы с данными и сервисом хранения файлов MinIo. Для удобства разбора и обработки, табличные файлы пользователя хранятся в едином системном формате “.xlsx”.

Класс репозиторий для взаимодействия с СУБД PostgreSQL обеспечивает сохранение и извлечение метainформации о наборах данных. Метаинформация включает в себя идентификатор пользователя, тип папки пользователя, размер набора данных и дата его создания.

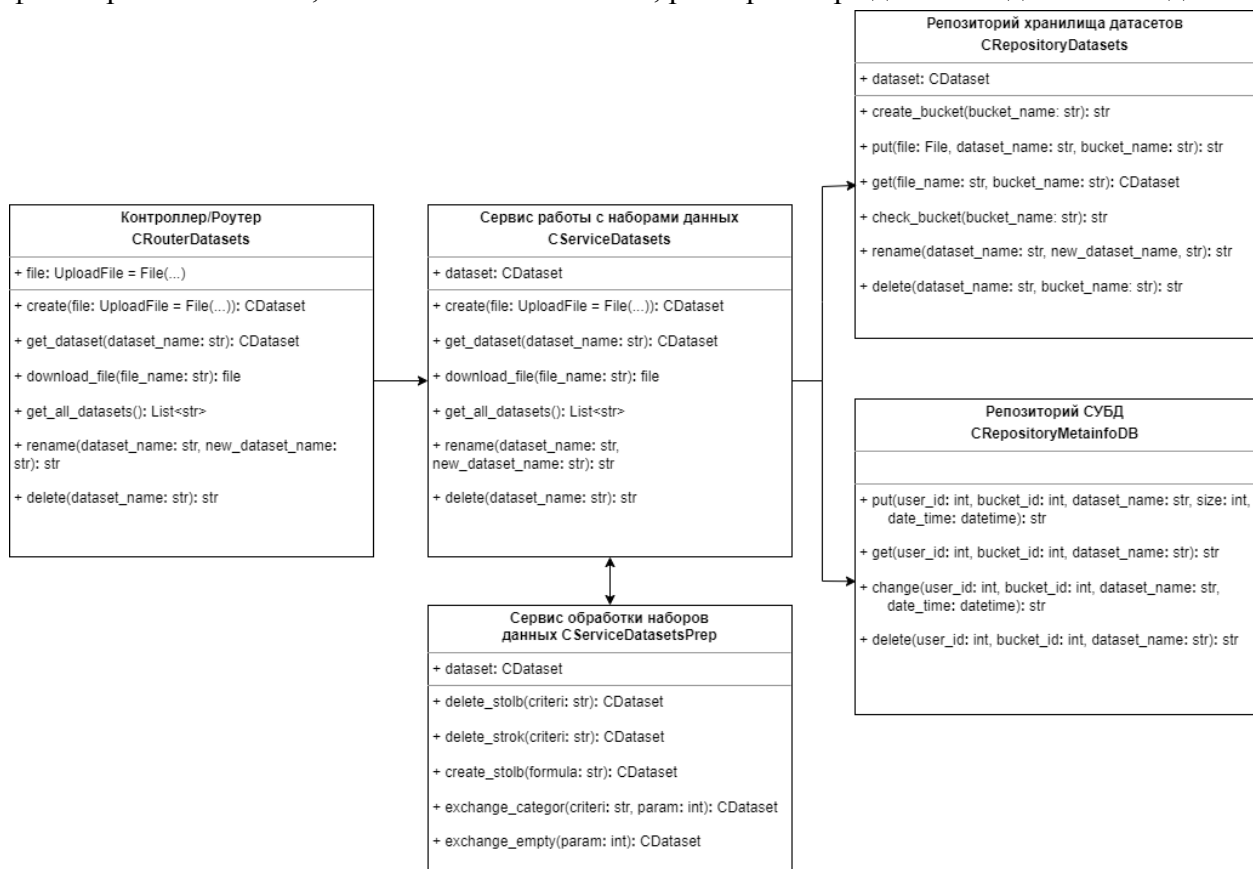


Рис. 2. Диаграмма классов сервиса управления наборами данных

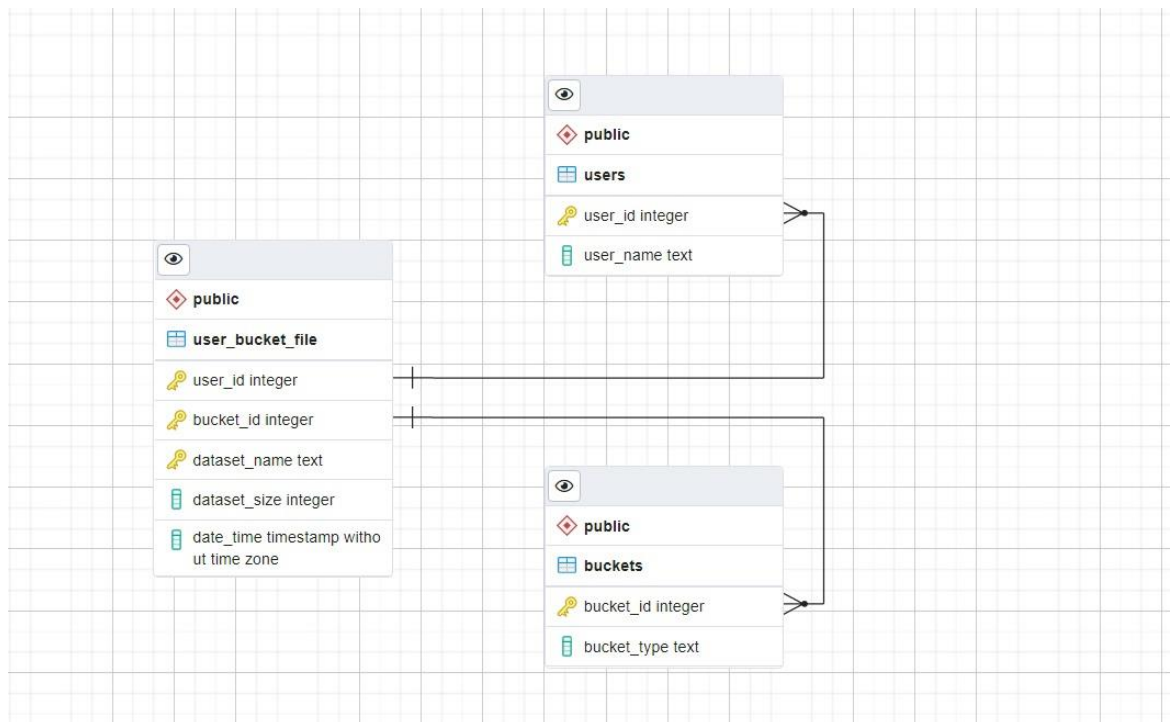


Рис. 3. Структура таблиц в СУБД

## Разработка графического интерфейса пользователя

Самый простой способ внедрения TypeScript в HTML-документ – использование тега `<script>`. Теги `<script>` часто помещают в элемент `<head>`, и ранее этот способ считался чуть ли не обязательным. Однако в наши дни теги `<script>` используются как в элементе `<head>`, так и в теле web-страниц (тег `<body>`) [3].

Таким образом, на одной web-странице могут располагаться сразу несколько сценариев. Как правило, выполнение сценариев браузерами происходит по мере их загрузки. Браузер читает HTML-документ сверху вниз и, когда он встречает тег `<script>`, рассматривает текст программы как сценарий и выполняет его. Остальной контент страницы не загружается и не отображается, пока не будет выполнен весь код в элементе `<script>`.

Далее о верстке. CSS (Cascading Style Sheets) – язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML) [4].

Обычно CSS-стили используются для создания и изменения стиля элементов web-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Отделяя стиль представления документов от содержимого документов, CSS упрощает создание web-страниц и обслуживание сайтов [3].

Также, стоит отметить, что одной из важных частей Frontend-разработки является грамотно продуманный дизайн будущего продукта, а также логика работы с пользователем. Именно от того, насколько комфортно будет сделан дизайн, будет зависеть популярность сервиса у пользователей.

На рис. 4 представлен экран загрузки файлов с наборами данных, а также слева панель управления и аккаунт пользователя. Помимо отображаемого названия и расширения указываются и характеристики – в данном случае это объем данных.



Рис. 4. Макет страницы со списком наборов данных

Когда пользователь открывает любой из наборов данных, которые были либо созданы, либо ранее загружены, он переходит на следующий экран для работы с данными, рис. 5.



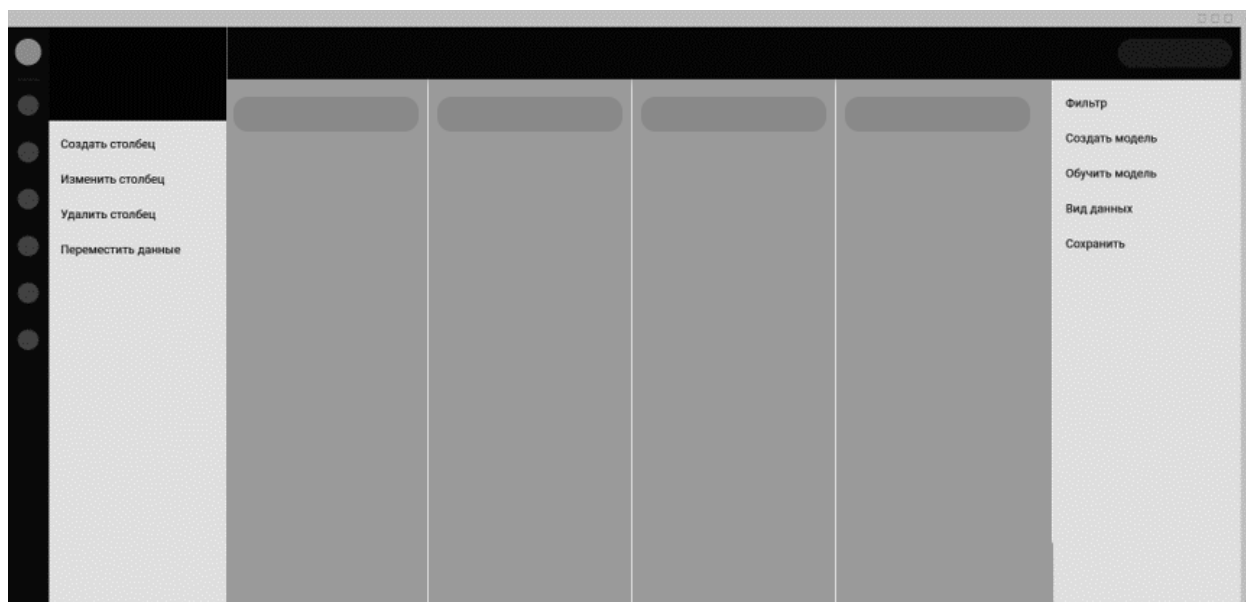


Рис. 5. Макет экрана для работы с данными

В левой панели располагается набор инструментов для работы со столбцами загруженных данных, в правой панели – инструментарий уже для работы со всем набором данных.

Также, при выборе любого инструмента из правой панели выдвигается окно выбора работы с данными, рис. 6.

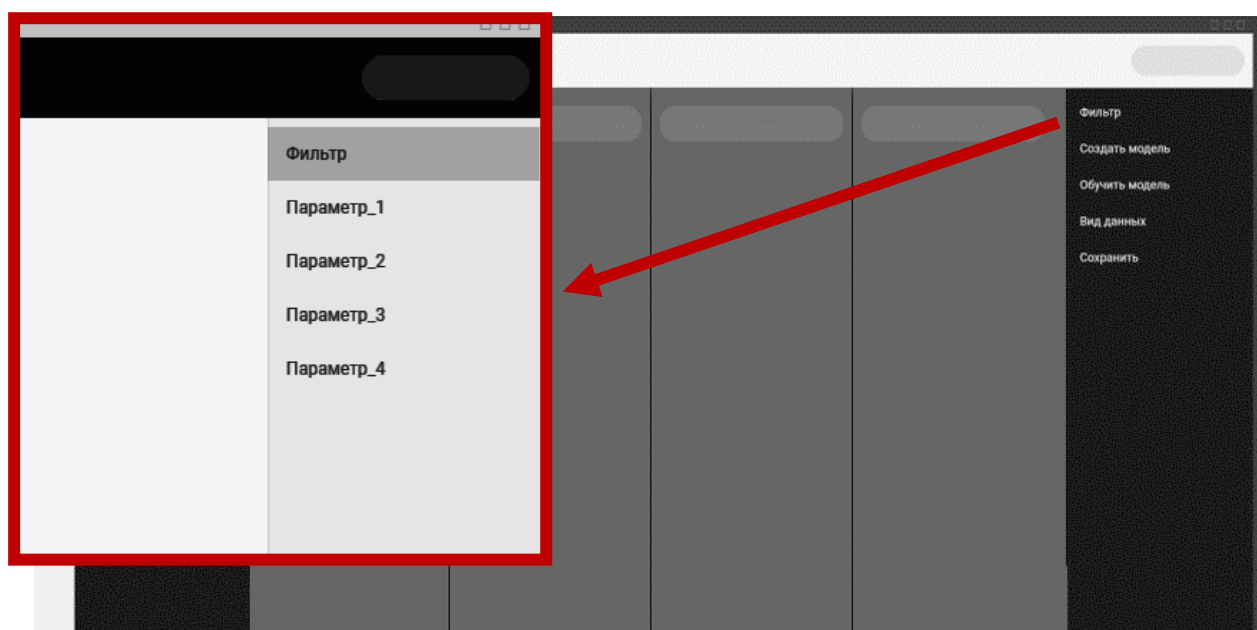


Рис. 6. Макет дополнительного окна с настройками инструмента

Панели инструментов располагаются по бокам экрана, чтобы пользователю было удобно видеть все загруженные данные, с которыми предстоит работать и строить модель в дальнейшем.

### Библиографический список

1. Balalaie A. Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture // IEEE Software. – 2016 – 33 (3) – P.42-52.

2. Creating a simple data-driven CRUD microservice // learn.microsoft.com [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/data-driven-crud-microservice> (дата обращения: 23.02.2023).

3. htmlbook.ru // htmlbook.ru [Электронный ресурс]. – URL: <http://htmlbook.ru/> (дата обращения: 04.04.2023).

4. javascript.ru // learn.javascript.ru [Электронный ресурс]. – URL: <https://learn.javascript.ru/> (дата обращения: 26.03.2023).

### References

1. Balalaie A. Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture // IEEE Software. – 2016 – 33 (3) – P.42-52.

2. Creating a simple data-driven CRUD microservice // learn.microsoft.com [Electronic resource]. – URL: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/data-driven-crud-microservice> (accessed: 23.02.2023).

3. htmlbook.ru // htmlbook.ru [Electronic resource]. – URL: <http://htmlbook.ru/> (accessed: 04.04.2023).

4. javascript.ru // learn.javascript.ru [Electronic resource]. – URL: <https://learn.javascript.ru/> (accessed: 26.03.2023).

## DEVELOPING OF NO-CODE DATA PROCESSING AND STORAGE SYSTEM

*Rusakov Vadim V.<sup>1</sup>, Mokuneva Valeria E.<sup>2</sup>, Seletkov Ilya P.<sup>3</sup>*

Perm State University, 15 Bukireva St., 614990, Perm,  
Vadim.rusakov.02@email.ru<sup>1</sup>, mokuneva.valeria@gmail.com<sup>2</sup>, iseletkov@gmail.com<sup>3</sup>

The process of creating a system for data processing and analysis is considered. The analysis of competitor systems was carried out and their strengths and weaknesses were identified. A structural diagram of the system as a whole was developed, which includes a description of frontend and backend interaction via HTTP protocol. For the user interaction with the system a web-interface was developed. External services for storing user files and meta-information about them were connected and configured.

Keywords: data processing, no-code, frontend, backend, microservices, pandas.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ РЕЧЕВОЙ ФИЛЬТРАЦИИ

*Садиллов Матвей Викторович, Кнутова Наталия Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, matvey.sadilov1@gmail.com

Сегодня все большее значение приобретает речевая фильтрация в приложениях и системах, где голосовое взаимодействие с техническими устройствами является наиболее удобным способом коммуникации. Для голосового взаимодействия важно не только устранить шумы с аудиозаписи, но и определить говорящего. Например, голосовые помощники могут выполнять определенный набор функций, если голос обратившегося принадлежит владельцу технического устройства. Существенным фактором является возможность более эффективного распознавания речи (транскрибации) при отсутствии ненужных шумов в аудио-поток. Речевая фильтрация выполняет шумоподавление и разделение входящего речевого сигнала по принадлежности к различным дикторам, что упрощает работу модуля транскрибации речи. В статье рассмотрены существующие методы шумоподавления и диаризации, а также спроектирована собственная система.

Ключевые слова: шумоподавление, диаризация, речевая фильтрация, процесс удаления шума из сигнала, сравнение методов шумоподавления, сравнение методов диаризации, разработка приложения.

Шумоподавление [1] – это процесс устранения шумов из сигнала с целью повышения его качества.

Диаризация [2] – процесс разделения входящего аудиопотока на однородные сегменты в соответствии с принадлежностью аудиопотока тому или иному говорящему.

Достижение диаризации возможно из-за такой характеристики речевого сигнала, как частота основного тона – вибрация голосовых связок. Среднее значение этой частоты у разных людей отличается, у каждого говорящего имеется отклонение в пределах октавы выше или ниже центральной частоты. Благодаря этому и появляется возможность идентификации человека по голосу.

При исследовании методов шумоподавления было проведено сравнение Spectral Substraction, RNNoise, DCCRN, PoCoNet и DTLN. Каждый метод имеет свои особенности: Spectral Substraction [3] основан на математических методах, что обеспечивает низкую эффективность работы, но быстрое выполнение; RNNoise [4] – гибридное решение, использующее меньшее число параметров для нейронной сети, чем полноценная нейронная сеть, что обеспечивает более высокую эффективность, чем у методов, полностью основанных на нейронной сети, но работает медленнее, чем Spectral Substraction; DCCRN [5] отличается использованием LSTM, что обеспечивает более долгосрочные зависимости, но не может работать в режиме реального времени на слабых устройствах; PoCoNet [6] – метод с лучшей эффективностью из представленных, использующий Dense Attention Block, но требующий значительных вычислительных мощностей и времени на обучение; DTLN [7] – метод с результатами, близкими к более сложным архитектурам, который дает шипящий звук после обработки сигнала и не работает с реверберирующим сигналом, однако может работать в режи-

ме реального времени. Выбор был сделан в пользу DCCRN-подобной архитектуры из-за лучшего соотношения требуемых ресурсов и качества выходной дорожки.

Проведено сопоставление двух методологий диаризации, а именно GMM-UBM [8] и X-vector [9]. Опытные данные говорят о том, что первый метод является менее требовательным к предварительной обработке данных для обучения, однако проявляет меньшую эффективность в сравнении с X-vector. Последний же метод, в свою очередь, требует значительно больших усилий и материальных затрат для подготовки обучающих данных нейронной сети, однако позволяет достичь более высокого качества выходных результатов. Исходя из проведенного сравнения, было сделано решение в пользу метода X-vector, как более точного и эффективного, благодаря способности данного метода к снижению количества возможных ошибок в идентификации дикторов и достижения более высокого качества результатов.

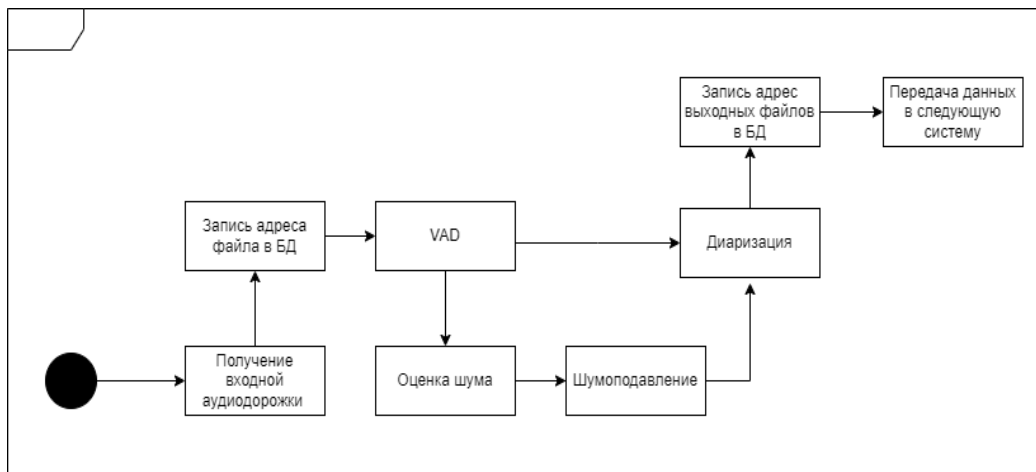
Для конструирования и разработки собственной системы принято решение использовать App.diagrams.net [10] в качестве средства проектирования, благодаря его бесплатности, удобству использования и библиотеке стандартных символов. Среди фреймворков глубокого обучения был выбран TensorFlow [11], ввиду его гибкости в выборе языка программирования и наличия стека технологий, а также наличия высокоуровневого API Keras, достаточного для регулирования проекта речевой фильтрации. Как язык программирования был выбран Python [12], поскольку TensorFlow специализирован на работе с платформой Python, и только на этом языке доступен API Keras. В роли СУБД была выбрана PostgreSQL [13], основными причинами выбора которой стали наличие открытого исходного кода, а также наличие Postgres Professional – российского вендора, что обеспечивает различные коммерческие возможности для его использования в различных организациях.

На этапе анализа предметной области были определены основные требования к системе речевой фильтрации.

1. Эффективность. Система должна быть способна эффективно удалять шумы, такие как фоновый шум, шум ветра, шум двигателей и так далее;
2. Точность. Система должна быть точной в определении голоса каждого участника в аудиозаписи;
3. Стабильность. Система должна быть стабильной и не вызывать ошибок при обработке аудиозаписей;
4. Совместимость. Система должна быть совместима с разными форматами аудио, программным обеспечением и оборудованием;
5. Легкость использования. Система должна быть удобной в использовании, чтобы пользователи могли легко настраивать систему и получать нужную информацию;
6. Гибкость. Система должна быть настраиваемой для работы с разнообразными источниками звука и типами шумов.

### **Диаграмма состояний**

Для рассмотрения состояний системы во время работы используем диаграмму состояний, представленную на рисунке 1.



**Рис. 1. Диаграмма состояний**

После получения входных данных для анализа и записи этого файла в БД происходит процедура Voice Activity Detection, которая нужна и для диаризации, и для оценки шума. Перед диаризацией также необходимо выполнить шумоподавление, так как очищенный от лишних шумов голос проще идентифицировать, тем самым повышая точность самой диаризации. Процесс диаризации в качестве выходных данных выдаёт несколько звуковых файлов, в принадлежности к различным дикторам, которые записываются в БД и связываются с исходным файлом, полученным на входе системы речевой фильтрации.

### Диаграмма последовательности

Диаграмма последовательности позволяет увидеть, как взаимодействуют различные части системы между собой во время работы системы. Также благодаря этой диаграмме можно увидеть последовательность шагов, которые проделывает система для достижения итогового результата. Такая диаграмма представлена на рис. 2.



**Рис. 2. Диаграмма последовательности**

Начало работы системы речевой фильтрации всегда одинаковое – внешняя система, будь то пользователь или другая программа, передаёт в нашу систему аудиофайл. Этот файл сохраняется в базе данных. Затем система проводит необходимые работы по шумоподавлению и диаризации, получая выходные аудиофайлы, которые также записываются в БД. В самом конце во внешнюю систему отправляется сообщение о завершении процесса.

### База данных

База данных в моей системы нужна для связи между входным аудиофайлом и выходными аудиофайлами. Сами файлы хранить в БД не имеет смысла, куда проще хранить ссылки на файлы на сервере. Таблицы и связи между ними представлены на рисунке 3.

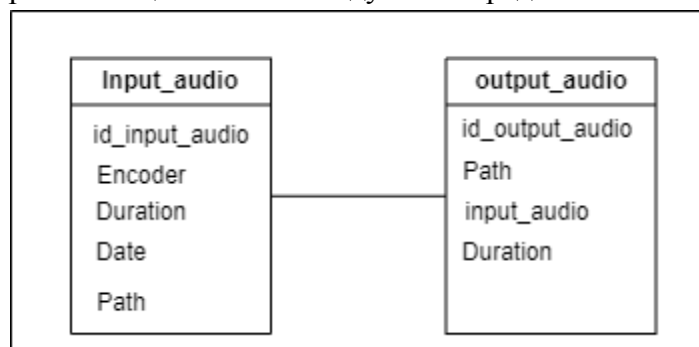


Рис. 3. База данных

Как видно из рисунка, смоделированная мной БД имеет всего 2 таблицы – таблицу для входных аудиофайлов и выходных аудиофайлов.

Таблица входных файлов “Input\_audio” хранит в себе информацию о всех файлах, полученных из внешних систем. Она имеет следующие поля:

- id\_input\_audio – уникальный идентификатор, ключ таблицы;
- Encoder – кодировка аудиофайла;
- Duration – длительность аудио фрагмента;
- Date – дата внесения аудиофайла в БД;
- Path – путь до файла.

Таблица выходных файлов “Output\_audio” создана для хранения аудиодорожек, полученных после работы системы речевой фильтрации. Она имеет следующие поля:

- id\_output\_audio – уникальный идентификатор, ключ таблицы;
- Path – путь до файла;
- input\_audio – информация о входном файле, из которого был получен данный файл;
- Duration – длительность аудио фрагмента.

Данные две таблицы связывает поле input\_audio, которое ставит в соответствие каждому выходному из системы файлу определённый входной файл. Поля Duration в обеих таблицах может казаться избыточным, однако благодаря им можно узнать, сколько времени во входном аудиофайле занимают голоса диктора, а сколько – только лишь шум.

### Библиографический список

1. Шкритек П. Справочное руководство по звуковой схемотехнике / пер. с нем. М.: Мир, 1991. 446 с.
2. Noulas A., Englebienne G., Kröse B.J.A. Multimodal Speaker Diarization // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012. № 34(1). P. 79–93.

3. Чуцупал В.Я., Чичагов А.С., Маковкин К.А. Цифровая фильтрация зашумленных речевых сигналов
4. A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement // Arxiv.org [Электронный ресурс]. – URL: <https://arxiv.org/pdf/1709.08243.pdf> (дата обращения: 10.06.2023)
5. DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement // Arxiv.org [Электронный ресурс]. – URL: <https://arxiv.org/abs/2008.00264> (дата обращения: 10.06.2023)
6. PoCoNet: Better Speech Enhancement with Frequency-Positional Embeddings, Semi-Supervised Conversational Data, and Biased Loss // Arxiv.org [Электронный ресурс]. – URL: <https://arxiv.org/pdf/2008.04470.pdf> (дата обращения: 10.06.2023)
7. Dual-Signal Transformation LSTM Network for Real-Time Noise Suppression // Arxiv.org [Электронный ресурс]. – URL: <https://arxiv.org/pdf/2005.07551.pdf> (дата обращения: 10.06.2023)
8. Scikit learn // Gaussian mixture models [Электронный ресурс]. – URL: <https://scikit-learn.org/stable/modules/mixture.html> (дата обращения: 10.06.2023)
9. X-Vectors: Robust DNN Embeddings for Speaker Recognition // Maelfabien [Электронный ресурс]. – URL: [https://maelfabien.github.io/assets/litterature/representation/x\\_vector.pdf](https://maelfabien.github.io/assets/litterature/representation/x_vector.pdf) (дата обращения: 10.06.2023)
10. App.diagrams.net [Электронный ресурс]. – URL: <https://app.diagrams.net/> (дата обращения: 10.06.2023)
11. TensorFlow [Электронный ресурс]. – URL: <https://www.tensorflow.org/?hl=ru> (дата обращения: 10.06.2023)
12. Python 3.11.3 documentation [Электронный ресурс]. – URL: <https://mybibliography.ru/sayt> (дата обращения: 10.06.2023)
13. PostgreSQL.com [Электронный ресурс]. – URL: <https://www.postgresql.org/> (дата обращения: 10.06.2023)

## References

1. *Shkritek P.* Spravochnoe rukovodstvo po zvukovoi skhemotekhnike [Reference book on audio circuitry]. Moscow, Mir Publ., 1991. 446 p (In Russ.).
2. *Noulas A., Englebienn G., Kröse B.J.A.* Multimodal Speaker Diarization // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012. №34(1). P. 79–93.
3. *CHuchupal V.YA., CHichagov A.S., Makovkin K.A.* Cifrovaya fil'traciya zashumlennyh rechevyh signalov (In Russ.).
4. A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement // Arxiv.org [Electronic resource]. – URL: <https://arxiv.org/pdf/1709.08243.pdf> (accessed: 10.06.2023).
5. DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement // Arxiv.org [Electronic resource]. – URL: <https://arxiv.org/abs/2008.00264> (accessed: 10.06.2023).
6. PoCoNet: Better Speech Enhancement with Frequency-Positional Embeddings, Semi-Supervised Conversational Data, and Biased Loss // Arxiv.org [Electronic resource]. – URL: <https://arxiv.org/pdf/2008.04470.pdf> (accessed: 10.06.2023).
7. Dual-Signal Transformation LSTM Network for Real-Time Noise Suppression // Arxiv.org [Electronic resource]. – URL: <https://arxiv.org/pdf/2005.07551.pdf> (accessed: 10.06.2023).

8. Scikit learn // Gaussian mixture models [Electronic resource]. – URL: <https://scikit-learn.org/stable/modules/mixture.html> (accessed: 10.06.2023).
9. X-Vectors: Robust DNN Embeddings for Speaker Recognition // Maelfabien [Electronic resource]. – URL: [https://maelfabien.github.io/assets/litterature/representation/x\\_vector.pdf](https://maelfabien.github.io/assets/litterature/representation/x_vector.pdf) (accessed: 10.06.2023).
10. App.diagrams.net [Electronic resource]. – URL: <https://app.diagrams.net/> (accessed: 10.06.2023)
11. TensorFlow [Electronic resource]. – URL: <https://www.tensorflow.org/?hl=ru> (accessed: 10.06.2023).
12. Python 3.11.3 documentation [Electronic resource]. – URL: <https://mybibliography.ru/sayt> (accessed: 10.06.2023).
13. PostgreSQL.com [Electronic resource]. – URL: <https://www.postgresql.org/> (accessed: 10.06.2023).

## **DESIGN AND DOCUMENTATION OF SPEECH FILTERING SYSTEM**

*Sadilov M.V., Knutova N.S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [matvey.sadilov1@gmail.com](mailto:matvey.sadilov1@gmail.com)

Today, speech filtering is becoming increasingly important in applications and systems where voice interaction with technical devices is the most convenient way of communication. For voice interaction, it is important not only to eliminate noise from the audio recording, but also to identify the speaker. For example, voice assistants can perform a certain set of functions if the voice of the applicant belongs to the owner of the technical device. An essential factor is the possibility of more effective speech recognition (transcription) in the absence of unnecessary noise in the audio stream. Speech filtering performs noise reduction and separation of the incoming speech signal by belonging to different speakers, which simplifies the operation of the speech transcription module. The article discusses the existing methods of noise reduction and diarization, and also designed its own system.

**Keywords:** noise reduction, diarization, speech filtering, the process of removing noise from the signal, comparison of noise reduction methods, comparison of diarization methods, application development.



## РАЗРАБОТКА СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ И РОБОТИЗАЦИИ РАБОЧИХ ПРОЦЕССОВ

*Семёнов Сергей Павлович<sup>1</sup>, Саломатов Данил Алексеевич<sup>2</sup>, Селетков Илья Павлович<sup>3</sup>*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, [gradvans@yanex.ru](mailto:gradvans@yanex.ru)<sup>1</sup>, [mr.soba@yandex.ru](mailto:mr.soba@yandex.ru)<sup>2</sup>, [iseletkov@gmail.com](mailto:iseletkov@gmail.com)<sup>3</sup>

В работе выявлена потребность в разработке программной платформы для автоматизации рабочих процессов с минимальным требованием к необходимости написания исходного кода, сформулированы требования к такой системе, описана её целевая архитектура, модель данных предметной области, описаны технологии построения клиентской части и организации клиент-серверного взаимодействия. Также рассмотрены особенности чтения информации о рабочих процессах из файлов формата bpmn. В конце приведены снимки макета графического интерфейса пользователя.

Ключевые слова: рабочие процессы, spring boot, vuejs, bpmn, микросервисы, клиент-серверная архитектура.

### Введение

В настоящее время развитие промышленности требует всё большей автоматизации процессов, в том числе процессов управления и обработки информации. Согласно исследованию McKinsey [1], в 2020 году автоматизацию проводили 66% опрошенных компаний, что говорит о росте по сравнению с 57% двумя годами до этого.

С другой стороны, в текущих условиях возросли риски, связанные с использованием ПрЭВМ западного производства. Представленные на рынке отечественные системы решают, в первую очередь, задачу документооборота, нежели обработки информации.

В связи с этим существует потребность в разработке инструментов, позволяющих упростить автоматизацию рабочих процессов управления, принятия решений и обработки информации.

Системы автоматизации и роботизации исполнения процессов – это платформы для автоматизации бизнес-процессов. Они используются для оптимизации и ускорения рабочих процессов, улучшения качества и эффективности работы, повышения безопасности и контроля над документами, а также уменьшения временных и финансовых затрат на бизнес-процессы.

### Постановка задачи

Требуется разработать сервис для управления рабочими процессами, найти, проанализировать и выбрать готовое ПО, которое может быть сервисом-исполнителем программного кода на языке python,

### Описание и архитектура системы

Архитектура системы включает три основных компоненты:

- интерфейсный компонент является ключевым компонентом, который взаимодействует с пользователем, получает входные данные и управляет логикой представления, контролируя взаимодействие пользователя с приложением

- web-сервер, также известный как внутренний компонент или компонент на стороне сервера, обрабатывает бизнес-логику и обрабатывает запросы пользователей, направляя запросы нужному компоненту и управляя всеми операциями приложения.
- Сервер базы данных предоставляет необходимые данные для приложения. Он обрабатывает задачи, связанные с данными.

Схематичное представление архитектуры системы см. на рис. 1.

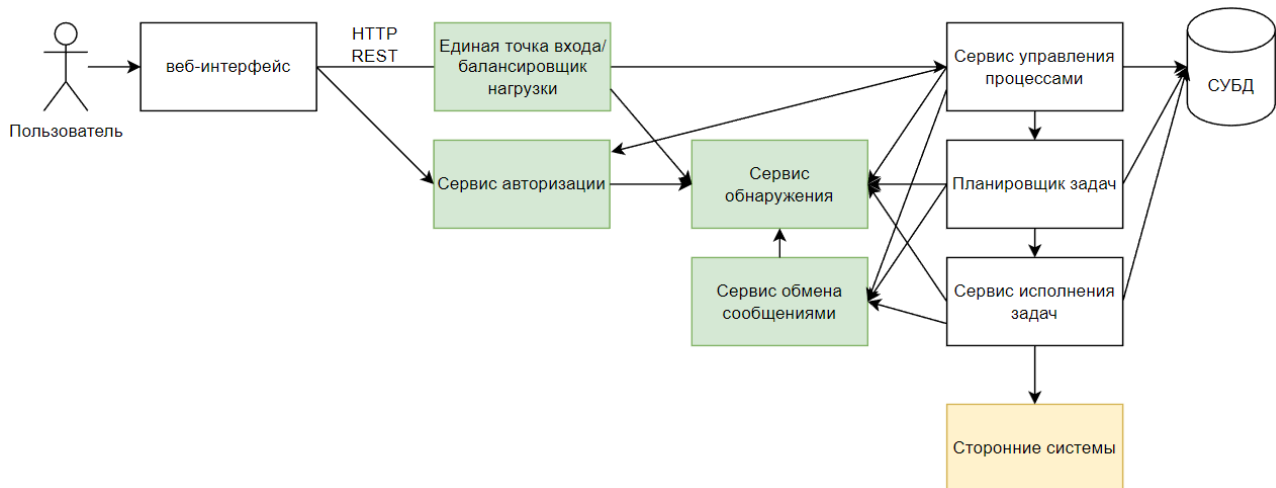


Рис. 1. Архитектура системы

### Технологии создания серверной части

Для разработки системы была выбрана микросервисная архитектура. Такой подход позволит масштабировать, усовершенствовать функционал без критических изменений кода. Spring Boot предоставляет удобные инструменты и функциональность для такого подхода создания приложений [2]. С ним удобно работать при помощи языка программирования Kotlin, так как это современный инструмент, облегчающий работу программиста, благодаря своему широкому функционалу. Для хранения данных была выбрана база данных PostgreSQL.

### Модель данных предметной области

После анализа предметной области были выделены основные компоненты, которые потребуются для работы с процессами. Для них были созданы классы, схему UML которых можно увидеть на рис. 2.

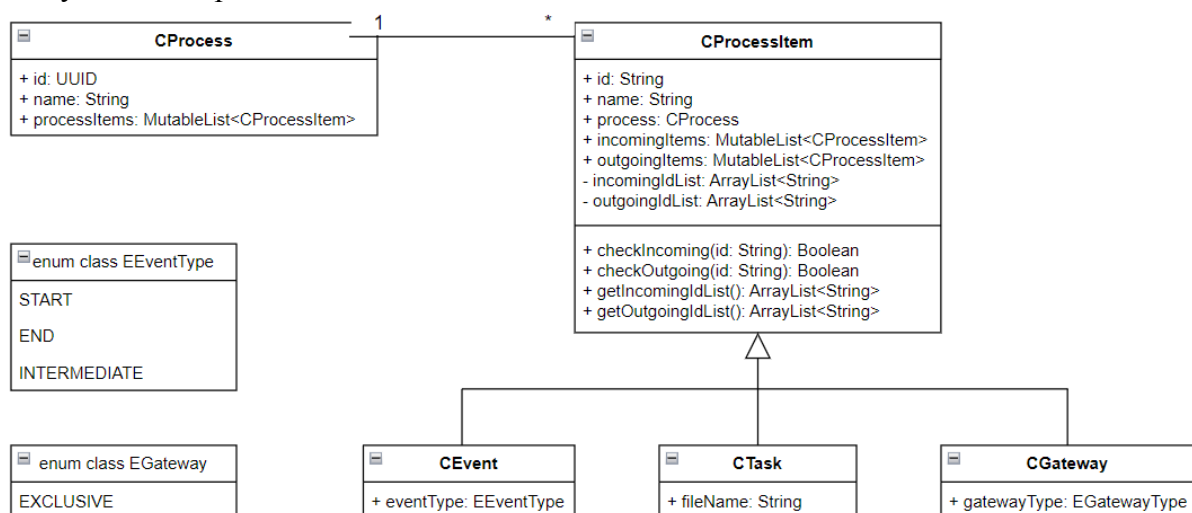


Рис. 2. Диаграмма классов UML предметной области

## Чтение файлов \*.BPMN

На вход в систему будут поступать файлы формата \*.BPMN, которые и будут представлять процесс. Сам файл содержит текст в виде XML объектов. Ключевыми элементами такого процесса являются сам процесс, его элементы (события, задачи, шлюзы) и их связи между собой. По завершении обработки, процесс собирается в программные классы и сохраняется в базу данных уже как сущности.

### *Листинг 1 – фрагмент примера файла BPMN*

```
<semantic:task name="Создание запроса на проверку" id="ID-09f595f6-d56b">
  <semantic:incoming>ID-09f595ff-d56b</semantic:incoming>
  <semantic:outgoing>ID-9b5688fa-d56b</semantic:outgoing>
</semantic:task>
<semantic:task name="Проверка корректности оформления" id="ID-09f59600-d56b">
  <semantic:incoming>ID-9b5688fa-d56b</semantic:incoming>
  <semantic:outgoing>ID-9b5688fc-d56b</semantic:outgoing>
</semantic:task>
```

В этом примере можно увидеть в каком виде приходят данные в систему. Представленные задачи, принадлежащие процессу по созданию отчета, расположены внутри блока `<semantic:process>`. Каждый объект имеет такие параметры как название и уникальный идентификатор. В нашей программе процесс можно создать как с собственным идентификатором, так и со сгенерированным автоматически при помощи класса `UUID`. Связи между элементами процессов устанавливаются при помощи объектов `<semantic:incoming>` и `<semantic:outgoing>`. Если у одного объекта “исходящий” идентификатор совпадает с “входящим” идентификатором второго объекта – это говорит о том, что из состояния первого объекта можно перейти в состояние второго.

## Технологии создания пользовательской части

`Vue.js` и `Vuetify` – это мощные инструменты, которые позволяют разработчикам создавать интерактивные и привлекательные пользовательские интерфейсы веб-приложений. `Vue.js` является прогрессивным JavaScript-фреймворком, который позволяет создавать одностраничные приложения (SPA) и компоненты пользовательского интерфейса [3]. Одной из ключевых особенностей `Vue.js` является его реактивность. `Vuetify`, с другой стороны, представляет собой набор готовых компонентов пользовательского интерфейса. Он интегрируется непосредственно с `Vue.js`. `Vuetify` значительно упрощает разработку пользовательского интерфейса, предлагая готовые компоненты с простыми API для настройки внешнего вида и поведения.

В дополнение к `Vue.js` и `Vuetify`, мы также используем язык программирования JavaScript, который является основным языком разработки веб-приложений. С помощью JavaScript мы можем легко расширять функциональность `Vue.js` и `Vuetify`, добавлять пользовательскую логику, обрабатывать события и осуществлять асинхронные запросы к серверу.

## Взаимодействие с серверной частью

Существует необходимость получения актуальных данных из серверной части и отображения их в компонентах. Для этого нам нужно использовать клиентскую библиотеку HTTP, а поскольку `Vue.js` не имеет своей собственной, мы собираемся использовать `Axios`. `Axios` – это JavaScript-библиотека, которая предоставляет простой и элегантный способ от-

правки HTTP-запросов из браузера. Axios обеспечивает функции для выполнения запросов к серверу, таких как GET, POST, PUT, DELETE и другие. Она предоставляет удобный API для работы с данными и обработки ответов от сервера [4].

В совокупности с Vue.js и Vuetify, Axios обеспечивает надежное и эффективное взаимодействие с серверной частью. Мы можем использовать Axios для получения и отправки данных на сервер, обновления информации на странице без перезагрузки и обработки ошибок, возникающих во время запросов.

### Интерфейс пользователя

Интерфейс веб-приложения играет важную роль в обеспечении позитивного пользовательского опыта и эффективного взаимодействия с приложением. Он является лицом и голосом вашего приложения, определяющим первое впечатление, которое оно производит на пользователей. Примеры стартовой страницы – рис. 3, и главной страницы – рис. 4, представлены ниже.



Рис. 3. Стартовая страница

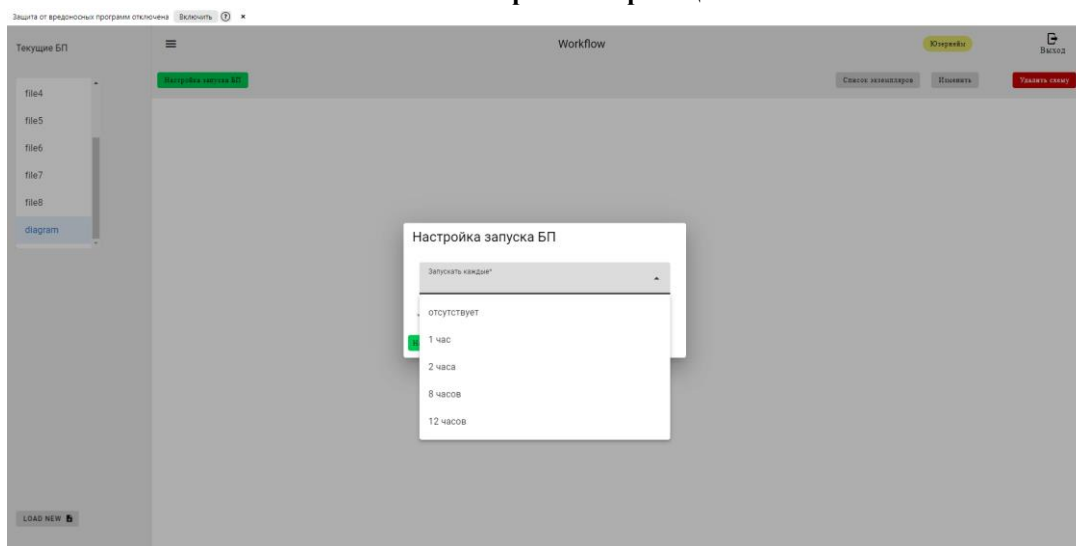


Рис. 4. Окно выбора периодичности автозапуска БП

### Заключение

В результате выполнения работы рассмотрена потребность в создании сервиса автоматизации исполнения рабочих и бизнес-процессов. В соответствии с постановкой задачей разработана часть web-интерфейса и основные микросервисы для работы с процессами. Даль-

нейшее развитие данной работы может включать расширение функциональности сервиса, оптимизацию его производительности и внедрение в реальные рабочие среды.

### Библиографический список

1. Современная workflow-система. Возможности, инструменты // it-guild [Электронный ресурс]. – URL: <https://it-guild.com/info/blog/sovremennaya-workflow> (дата обращения 15.02.2023).
2. Введение в Spring Boot // habr [Электронный ресурс]. – URL: <https://habr.com/ru/articles/435144/> (дата обращения 17.02.2023).
3. Документация по Vuetify 3 [Электронный ресурс]. – URL: <https://vuetifyjs.com/en/> (дата обращения: 23.01.2023).
4. Vue.js и слоистая архитектура: вынесение бизнес-логики в сервисы // habr [Электронный ресурс]. – URL: <https://habr.com/ru/companies/simbirsoft/articles/556270/> (дата обращения: 20.03.2023).

### References

1. Sovremennaya workflow-systema. Vozmojnosti, instrumenti // it-guild [Electronic resource]. – URL: <https://it-guild.com/info/blog/sovremennaya-workflow> (accessed 15.02.2023) (In Russ.).
2. Vvedenie v Spring Boot // habr [Electronic resource]. – URL: <https://habr.com/ru/articles/435144/> (accessed 17.02.2023) (In Russ.).
3. Vuetify 3 [Electronic resource]. – URL: <https://vuetifyjs.com/en/> (accessed: 23.01.2023).
4. Vue.js i sloistaya arhitektura: vinisenie bisnes logiki v servisy // habr [Electronic resource]. – URL: <https://habr.com/ru/companies/simbirsoft/articles/556270/> (accessed: 20.03.2023) (In Russ.).

## DEVELOPING OF SYSTEM FOR AUTOMATION AND ROBOTISATION OF WORKFLOWS

*Semenov Sergey P.<sup>1</sup>, Salomatov Danil A.<sup>2</sup>, Seletkov Ilya P.<sup>3</sup>*

Perm State University, 15 Bukireva St., 614990, Perm,  
rpadvans@yanex.ru<sup>1</sup>, mr.soba@yandex.ru<sup>2</sup>, iseletkov@gmail.com<sup>3</sup>

This work eliminates of neediness of developing software platform for automation of workflows with minimal requirements for programming skills. There are requirements for such systems, it's target architecture, model of processing data. The next is technologies for developing client part of the system and the client-system data exchange. Also, authors observe the problems of parsing of files in notation bpmn 2.0. Screenshots of prototype of pages of the web interface are in the end of paper.

Keywords: workflows, spring boot, vuejs, bpmn, microservices, client-server architecture.

## **МОДЕЛЬ ЦИФРОВОЙ ПЛАТФОРМЫ ДОСТАВКИ ПРОДУКТА: ПОЛИМОРФИЗМ ОСНОВНЫХ МОДУЛЕЙ, АВТОНОМНОСТЬ И СПОСОБНОСТЬ К ИНТЕГРАЦИИ В СТОРОННИЕ ЦИФРОВЫЕ ЭКОСИСТЕМЫ**

*Соколов Андрей Валерьевич, Соколова Ольга Леонидовна, Яшичев Дмитрий Львович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, as2710433@yandex.ru

**Аннотация.** Основным драйвером развития современного бизнеса становятся экосистемы цифровых платформ. В данной статье рассмотрены и систематизированы существующие модели цифровых платформ, их применение в бизнесе по доставке продуктов, проведен анализ моделей цифровых платформ в целом и моделей экосистем цифровых платформ доставки продуктов, в частности. На основе проанализированных данных, авторами разработана инновационная концепция архитектуры цифровой платформы с автономными ключевыми модулями, способными как к внутрисистемному взаимодействию, так и к успешной интеграции в сторонние экосистемы. Авторами представлена возможность применения указанной модели в сервисе по доставке товаров, рассмотрены основные функциональные возможности и преимущества такой архитектуры. Результатом данной разработки явилась готовая для применения модель цифровой платформы доставки продуктов, которая позволяет более быстро и гибко реагировать на изменения рынка, свободно интегрируется в сторонние экосистемы, позволяет модифицировать собственные модули без изменения системы в целом. Автономность модулей обеспечивает высокую надежность, гибкость и отказоустойчивость системы.

**Ключевые слова:** цифровые платформы, доставка продуктов, архитектура цифровых платформ, интернет-технологии, программное приложение, экосистема цифровых платформ

### **Введение**

Одной из актуальных тенденций современной мировой, и в том числе российской, экономики является рост объемов розничной торговли товарами через Интернет в форме B2C (Business-to-Consumer – Бизнес для Потребителя). Пандемия 2020 года привела к росту онлайн-торговли во всем мире. Самый большой скачок произошёл в России – рынок увеличился на 58%, темпы роста электронной торговли в 6–8 раз превышали темпы роста розницы, использующей традиционные каналы сбыта. По результатам 2022г. оборот интернет-торговли в России достиг 4,98 трлн руб. (58,82 млрд. долларов). По сообщению Ассоциации компаний интернет-торговли (АКИТ) этот показатель на 30% выше уровня 2021г. Ассоциация прогнозирует аналогичный прирост в 2023 г. [1].

Несмотря на значительный рост, российский рынок онлайн-торговли всё ещё уступает крупнейшим мировым рынкам. На тройку лидеров – Китай, Соединенные штаты Америки и Великобританию приходится 75,9% всего мирового объёма рынка электронной торговли [2].

Именно поэтому зарубежный ИТ-рынок сегодня задает тренд не только в электронной торговле в целом, но и ее инструментах в частности, в том числе на создание современных распределенных приложений на базе микросервисной архитектуры. Немаловажно, что многие эксперты в области электронной торговли отмечают, что успех во многом зависит от того, как организована система доставки.

Настоящая статья стала результатом начала разработки стартапа для создания экосистемы цифровой платформы доставки продуктов. Наблюдение за российским рынком показало, что в современной России были представлены лишь две крупные цифровые платформы доставки товаров – Delivery и Яндекс.Go. Однако, учитывая недавнее поглощение IT-гигантом Яндекс сервиса Delivery, можно подчеркнуть, что на рынке онлайн-заказа и доставки продуктов остался лишь один крупный игрок. Для сравнения, на американском рынке представлено не менее семи крупных игроков в этой области и множество небольших развивающихся стартапов. Очевидно, что монопольное положение сервиса Яндекс на российском рынке имеет негативные последствия как для рынка в целом, так и для конечного потребителя в частности.

Кроме того, было выявлено, что архитектурные модели цифровых платформ недостаточно описаны как в отечественной литературе, так и за рубежом. В основном, авторы статей затрагивают структуру цифровых платформ доставки продуктов в целом, не углубляясь в архитектуру платформ. В свете этого, авторами данной статьи рассмотрены подходы к формированию модели цифровой платформы, определен подход к классификации цифровых платформ с точки зрения ключевых модулей  $S_n$  и разработана уникальная модель цифровой платформы доставки продуктов.

#### **Анализ существующих подходов к архитектурным моделям цифровых платформ**

Существуют различные взгляды на архитектуру цифровых платформ, выбор конкретной при этом зависит от целей платформы, требований к производительности, стоимости и экономической целесообразности. Архитектура цифровой платформы может быть классифицирована по множеству критериев, рассмотрим основные из них:

1. Классификация по слоям, когда многие цифровые платформы строятся на основе трехслойной архитектуры, состоящей из слоя клиента, слоя приложений и слоя данных.
2. Среди типов архитектуры выделяют монолитную, микросервисную, событийно-ориентированную, серверлесс-архитектуру и прочие.
3. В классификации с точки зрения собственника платформы основной упор делается на организационно-правовые формы [3].
4. С точки зрения типа управления платформы делят на централизованные и децентрализованные. В данном подходе рассматриваются подход принятия решения единоличный или коллегиальный [4].
5. Классификация по типу задач платформы предполагает ориентацию на управление активами, сбор и анализ больших данных, расчеты и оптимизацию бизнес-процессов, дистрибуцию цифровых контента и т.д.
6. Классификация по типу участников процесса. Зарубежные авторы выделяют 3 типа экосистемы цифровой платформы: B2B – Business to business, B2C – Business to consumer и C2C – Consumer to consumer [5]. В любой из представленных моделей с одной стороны всегда продавец, а с другой – покупатель, а в варианте C2C потребитель может быть как продавцом, так и покупателем.

Применительно к авторской цифровой платформе использован подход с точки зрения участников бизнес-процесса и внедрена инновационная модель  $n$ -Client или  $nC$ , где  $n$  – количество клиентов той или иной экосистемы цифровой платформы. Модель функционирует на базе автономных микросервисов, лаконично вписывается в бизнес-логику системы, позволяет более быстро и гибко реагировать на изменения рынка, свободно интегрируется в сторонние экосистемы, позволяет модифицировать собственные модули без изменения системы в

целом. Автономность модулей обеспечивает высокую надежность, гибкость и отказоустойчивость системы (рис. 1).

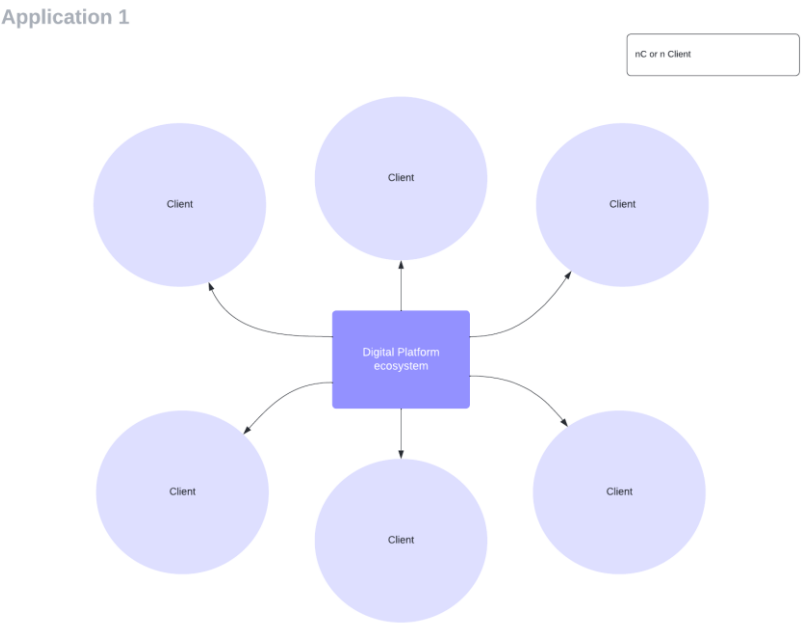


Рис. 1. Модель nC (n-Client)

**Цифровая платформа доставки продуктов**

На базе модели nC была разработана экосистема цифровой платформы доставки продукта, объединяющая три ключевые стороны сделки: заказчик товара (клиент), исполнитель (курьер) и точки заказа (рис. 2).



Рис. 2. Схема разработанной модели экосистемы цифровой платформы доставки

Несомненно, система также предполагает интеграцию стороннего внешнего модуля в части оплаты покупок – модуль платежной системы. Автономные модули в рамках разраба-



тываемой экосистемы также могут быть свободно интегрированы в сторонние внешние экосистемы цифровых платформ. Таким образом, данные модули могут являться компонентами как своей экосистемы, так и самостоятельными элементами для передачи в другие бизнес-логики.

Для более полной картины опишем алгоритм работы экосистемы цифровой платформы доставки продуктов. Клиент формирует заявку на товар в точке заказа, получает подтверждения заказа, осуществляет оплату, после чего идет поиск ближайшего свободного курьера, подключенного к модулю в данный момент, затем продукт доставляется клиенту в максимально короткий срок ближайшим курьером. Система оплаты в данном случае встроена как внешний модуль, интегрированный в экосистему цифровой платформы доставки продуктов, при этом модуль курьерской доставки и модуль точек заказа поддерживают собственную интеграцию в любую стороннюю систему, где это необходимо.

### Модуль система курьерской доставки

Перейдем к уникальной модели – автономный модуль курьерской доставки, который на сегодняшний день не имеет аналогов. По результатам анализа российских и зарубежных систем курьерской доставки мы не нашли моделей, подобных нашей бизнес-логике, ни в научной литературе, ни в реальной инфраструктуре. Система функционирует по принципу схожему с Яндекс.Такси, но вместо водителей в нашем случае выступают курьеры (рис. 3).

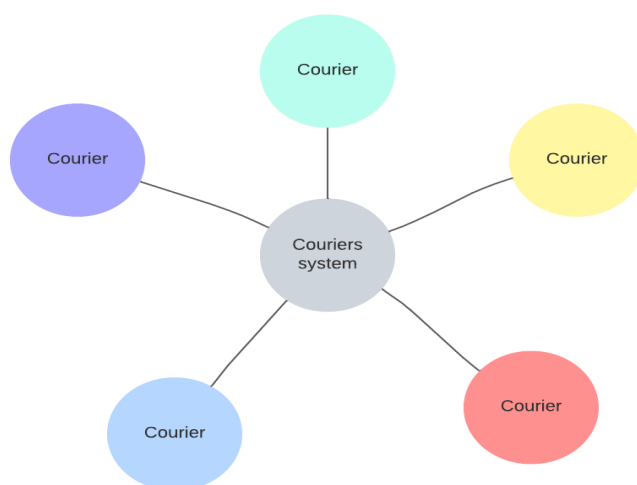


Рис. 3. Принцип функционирования курьерской системы

Желающие подработать проходят регистрацию в приложении и в последующем могут подключаться к модулю курьерской доставки в любое удобное для них время. Курьеры не имеют ограничений по количеству рабочих часов, свободно распоряжаются и планируют свое время, нет необходимости являться в офис, иметь дело с бухгалтерией, отчитываться перед вышестоящим руководством и уплачивать налоги – всё это модуль делает автоматически без вмешательства человека. Единая автономная система объединяет курьеров, которые выполняют задачи, поставленные клиентом напрямую без участия любых посредников.

Модуль курьерской доставки автономен и может быть успешно интегрирован в любую стороннюю внешнюю систему, такую как Авито, Яндекс Маркет, дискаунтер Ситилинк, Озон, СДЭК, ресторан, магазин продуктов или любой другой сервис электронной торговли, где может понадобиться доставка продукта. Таким образом, сторонам сделки доступна воз-

возможность прямого сотрудничества, внедрение позволит решить множество актуальных для них проблем, зачастую возникающих в отсутствие такой возможности.

### **Заключение**

В результате авторами получена модель экосистемы цифровой платформы доставки, включающая три автономных микросервиса, готовых к интеграции в сторонние экосистемы как по отдельности, так и совместно. Возможности интеграции самостоятельных модулей в сторонние экосистемы позволяют расширить функциональность существующих платформ, обеспечить их независимость от посреднических услуг, увеличить степень удобства использования сервиса, привлечь новых пользователей и успешно масштабировать бизнес. Такие преимущества делают предложенную модель привлекательной для бизнеса и конечных пользователей, обеспечивая высокую конкурентоспособность на рынке.

Авторами рассмотрены основные компоненты и архитектура цифровых платформ, а также представлена новая экосистема цифровой платформы для доставки, которая объединила в себе автономные модули с целью обеспечить бесперебойную и эффективную доставку продукта. Интеграция этих компонентов дает многочисленные преимущества для клиентов, партнеров по доставке и продавцов.

В заключение, дополнительно отметим актуальность развития такого направления как O2O (online-to-offline), так как сегодняшние реалии не предполагают существования без цифровых технологий. Все что может перейти в цифру активно внедряется как бизнесом, так и государством. Трудно представить, что человек, заказывающий такси, все еще набирает номер телефона, а не вызывает машину одним кликом в приложении. Авторы предполагают, что сервис независимых курьеров скоро будет нормой жизни как будто существовал всегда.

### **Библиографический список**

1. Открытый журнал [Электронный ресурс]. – URL: <https://journal.open-broker.ru/novosti/rost-oborota-internet-torgovli/> (дата обращения: 26.06.2023).
2. Мировой рынок e-commerce: развитие не остановить [Электронный ресурс]. – URL: <https://journal.open-broker.ru/investments/mirovoj-rynok-e-commerce/> (дата обращения: 26.06.2023).
3. Hein A., Schrieck M., Riasanow T. Digital platform ecosystems // *Electronic Markets* 30. 2020. С. 87–98.
4. Астахова Т., Колбанев М. Архитектура цифровой платформы // Перспективные направления развития отечественных информационных технологий: материалы IV межрегиональной научно-практической конференции, Севастополь, 2018. – С. 69-71.
5. Aneesh Zutshi A., Grilo A. The Emergence of Digital Platforms: A Conceptual Platform Architecture and impact on Industrial Engineering. // *Computers & Industrial Engineering* 136. 2019. С. 546-555.

### **References**

1. Open Journal [Electronic resource]. – URL: <https://journal.open-broker.ru/novosti/rost-oborota-internet-torgovli/> (accessed: 26.06.2023). (In Russ.).
2. Global e-commerce market: development can't be stopped [Electronic resource]. – URL: <https://journal.open-broker.ru/investments/mirovoj-rynok-e-commerce/> (accessed: 26.06.2023). (In Russ.).

3. *Hein A., Schrieck M., Riasanow T.* Digital platform ecosystems. *Electronic Markets* 30. 2020. 87–98.
4. *Astakhova T., Kolbanev M.* Architecture of the digital platform/ Т. Н. Астахова, М. О. Колбанев, А. А. Шамин // Promising directions of development of domestic information technologies: materials of the IV Interregional scientific and practical Conference, Sevastopol, 2018. 69-71. (In Russ.).
5. *Aneesh Zutshi A., Grilo A.* The Emergence of Digital Platforms: A Conceptual Platform Architecture and impact on Industrial Engineering. *Computers & Industrial Engineering* 136. 2019. 546-555.

**THE MODEL OF A DIGITAL PRODUCT DELIVERY PLATFORM:  
POLYMORPHISM OF CORE MODULES, AUTONOMY, AND ABILITY  
TO INTEGRATE INTO THIRD-PARTY DIGITAL ECOSYSTEMS**

*Sokolov Andrei V., Sokolova Olga L., Yashichev Dmitriy L.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, as2710433@yandex.ru

Digital platform ecosystems have become the main drivers of modern business development. This article examines and systematizes existing models of digital platforms, their application in the business of delivering products, and analyzes both overall models of digital platforms and models of digital platform ecosystems for product delivery in particular. Based on the analyzed data, the authors have developed an innovative model of digital platform architecture with autonomous key modules capable of both intra-system interaction and successful integration into third-party ecosystems. The authors have presented the possibility of applying such a model in a product delivery service, discussing its main functional capabilities and advantages. The result of this development is a ready-to-use model of digital platform architecture for delivering products. The model allows for a faster and more flexible response to market changes, is easily integrated into third-party ecosystems, and enables modification of its own modules without changing the system as a whole. Module autonomy ensures high reliability, flexibility, and system resilience. The ability to integrate independent modules into third-party ecosystems will allow for expanding the functionality of existing platforms, ensuring their independence from intermediary services, increasing the convenience of using the service, attracting new users, and successfully scaling the business. Such advantages make the proposed architecture attractive for both businesses and end-users, ensuring high competitiveness in the market.

**Keywords:** digital platforms, grocery delivery, digital platform architecture, internet technologies, software application, food delivery, digital platform ecosystem.

## ИССЛЕДОВАНИЕ ОСОБЕННОСТИ ВЕСЕННИХ ПАВОДКОВЫХ РЕЖИМОВ НА РЕКАХ КАМА И КОСА НА ОСНОВЕ ГИДРОЛОГИЧЕСКИХ ДАННЫХ

*Софронов Анатолий Михайлович, Русаков Сергей Владимирович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, anatoliy-sofronov@mail.ru

В данной работе решается задача исследования и изучения феномена под названием «Петля скорости» на основе гидрологических данных рек Кама и Коса. После введения необходимой для исследования терминологии проводится первичный анализ исходных данных, устанавливаются причины появления данного феномена, проводится классификация возможных вариантов, а также изучается характер его проявления. В результате классификации было выявлено 4 типа петель скорости. Было предложено обоснование причины появления данного феномена, основанное на решении Пуазейля. Для проведения исследования использовался программный пакет Microsoft Office, а также Visual Studio. Для построения моделей использовался набор гидрологических показаний из 8766 наблюдений.

Ключевые слова: моделирование, половодье, петля скорости, классификация, решение Пуазейля, точка бифуркации.

### **Введение**

Весеннее половодье характерно для большинства равнинных рек России. В этот относительно небольшой промежуток времени сток – объем воды, проходящей через определенный створ, – составляет около 60% объема годового стока, что приводит к выходу воды из меженного русла и затоплению поймы. Последствиями данного явления становится затопление обширных территорий, повреждение или разрушение зданий, дорог, сооружений, уничтожение растительного и животного мира в районе разлива рек. Кроме того, в зонах, подверженных потенциальному влиянию паводков, расположено множество населенных пунктов, промышленных объектов, объектов инфраструктуры, сельскохозяйственных территорий. Все это делает половодья серьезной сезонной проблемой, которая приводит к тяжелым эколого-экономическим последствиям. По оценкам Росгидромета наводнения на реках занимают первое место по суммарному среднегодовому ущербу среди всех опасных природных явлений (прямые экономические потери от наводнений составляют более 50% общего ущерба) [1, с. 9].

Предотвратить наступление половодья мы не можем, но в наших силах попытаться решить задачи, которые в свою очередь позволят своевременно проводить соответствующие мероприятия по подготовке и решению данной проблемы. Одной из таких задач является исследование феномена под названием «Петля скорости». Исследование было проведено в России, Пермский край на участке реки Кама от гидрологического поста (далее гидропост) Гайны до гидропоста Бондюг, включая промежуточный гидропост на реке Коса.

## Обзор литературы

«Петля скорости» – это форма образования функции зависимости средней скорости течения реки  $U(\text{м/с})$  от расхода  $Q(\text{м}^3/\text{с})$  в период половодья, построенной на одном и том же живом сечении [1].

Живое сечение – это сечение реки плоскостью перпендикулярно течению реки.

Расход  $Q(\text{м}^3/\text{с})$  – объем воды, протекающий через живое сечение за единицу времени.

Средняя скорость течения реки  $U(\text{м/с})$  – усредненное значение скорости всех потоков, проходящих через одно и то же живое сечение.

Как известно, скорость реки – величина непостоянная. Согласно тексту научной статьи Л.К. Давыдова, «Скорости течения в реках неодинаковы в различных точках потока: они изменяются и по глубине, и по ширине живого сечения. На каждой отдельно взятой вертикали наименьшие скорости наблюдаются у дна, что связано с влиянием шероховатости русла. От дна к поверхности нарастание скорости сначала происходит быстро, а затем замедляется, и максимум в открытых потоках достигается у поверхности или на расстоянии  $0,2H$  от поверхности» [2]. Именно из-за её непостоянства используют среднее значение скорости течения реки.

В качестве исходных данных для проведения исследования были предоставлены следующие данные по каждому из 3-х русел – Гайны, Бондюг и Коса – за 2008-2015 гг. посуточно:

- 1) Рельеф русла, представленный в виде набора точек с координатами,  $x, y$  (м)
- 2) Высота уровня воды на данном гидропосте,  $H$  (м)
- 3) Расход воды,  $Q(\text{м}^3/\text{с})$

Для математического моделирования рельефа дна реки используют способ гидрометрических измерений профиля дна реки промерами глубин. Алгоритм моделирования:

- 1) Выбирают поперечное сечение (гидрометрический створ), на котором будут моделировать рельеф
- 2) Проходя по створу, устанавливают промерные точки, в которых будут замерять глубину реки
- 3) Важным условием является замерять глубину при движении по линии створа от левого берега к правому
- 4) Имея на берегу точку, у которой известна высота относительно уровня моря, производят нивелирование поймы до этой отметки
- 5) Заносят полученные вычисления на график, получая математическую модель рельефа дна

Данный способ хорошо иллюстрирует рис. 1.[3]

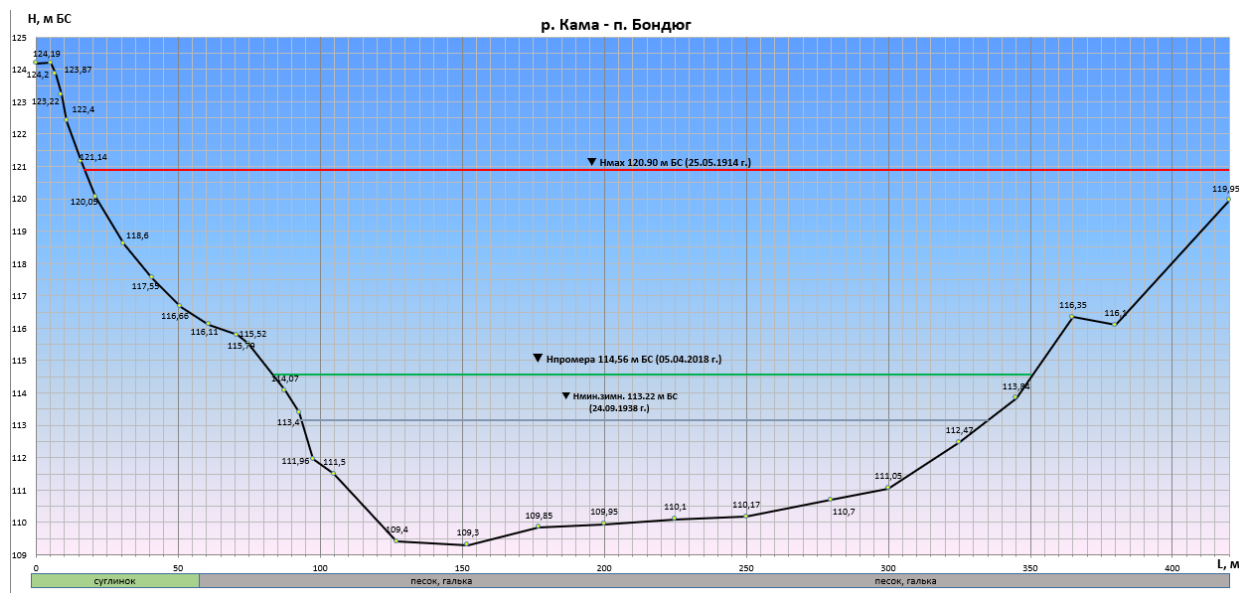


Рис. 1. Рельеф русла реки вблизи гидропоста Бондюг

### Постановка задачи

В качестве объекта для моделирования рассматривается участок реки Кама от п. Гайны до п. Бондюг с учётом правого притока р. коса (рис. 2).



Рис. 2. Расположение рассматриваемых гидропостов

Исходными данными для задачи являются:

- Значения суточного расхода воды  $Q$  и уровня воды в реке  $H$  (относительно уровня моря), измеренные на рассматриваемых гидропостах. Данные были взяты за 2008-2015 годы
- Информация о рельефе русла реки вблизи данных гидропостов (рис. 3 – рис. 5)

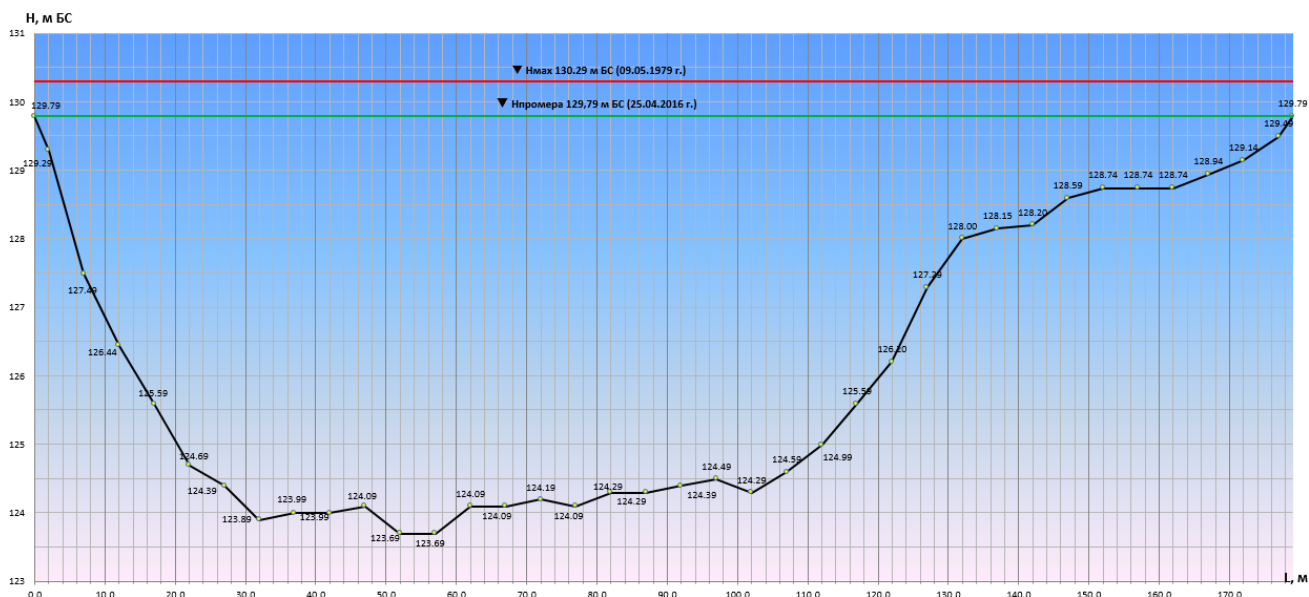


Рис. 3. Рельеф русла реки вблизи гидропоста на реке Коса (п. Усть-Коса)

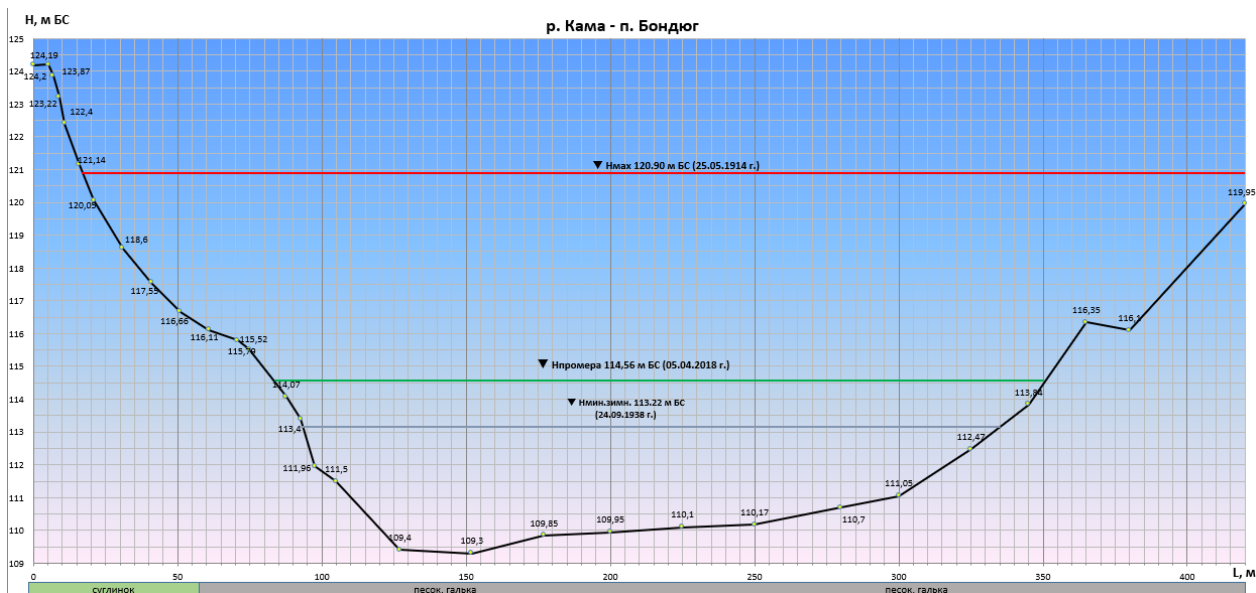


Рис. 4. Рельеф русла реки вблизи гидропоста в п. Бондюг

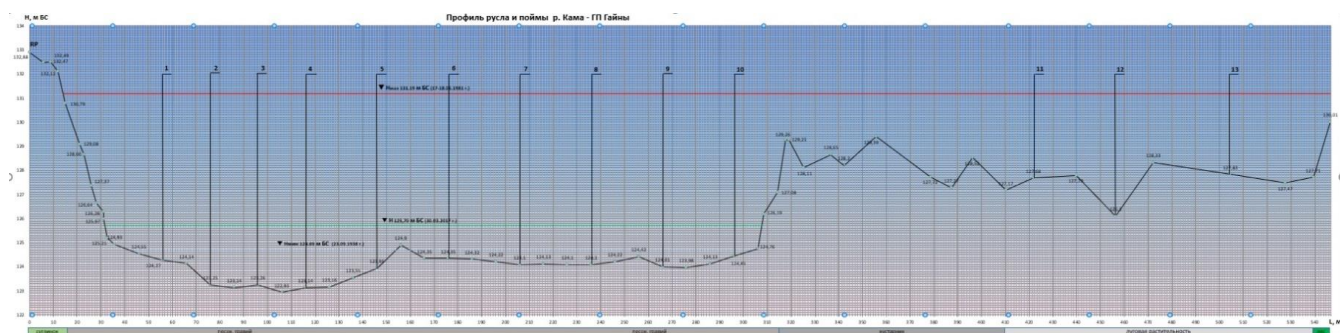


Рис. 5. Рельеф русла реки вблизи гидропоста в п. Гайны

Указанные исходные данные позволяют вычислить, помимо прочего, средние скорости реки, которые можно учитывать при построении зависимостей. Зная рельеф русла и уровень воды  $H$ , несложно найти площадь живого сечения реки  $S$ , например, численно интегрируя

методом трапеций. Средняя скорость реки находится, как отношение расхода воды, измеренного гидропостом, и найденной нами площади живого сечения по формуле

$$U = \frac{Q}{S}$$

Построим соответствующие таблицы для каждого русла, где зафиксируем все исходные данные для всех лет, а также вычислим все величины, необходимые для дальнейшего исследования. Этими величинами будут площадь живого сечения русла  $S(\text{м}^2)$  в соответствующий день и средняя скорость течения в этот же день.

Посмотрим, как меняются расход  $Q$ , уровень воды  $H$  и средняя скорость течения реки  $U$  в течение одного года (рис. 6. – рис. 8). Для примера возьмем Бондюг, 2008 год.

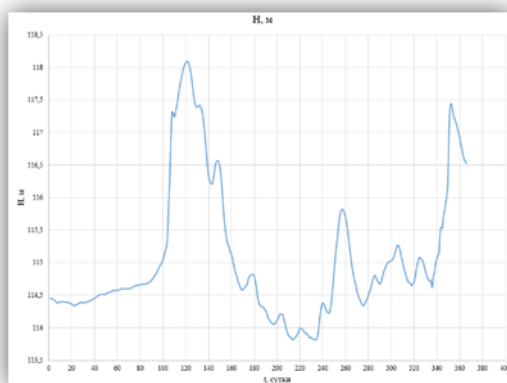


Рис. 6. Изменение уровня воды  $H$  в течение одного года, Бондюг, 2008 г.

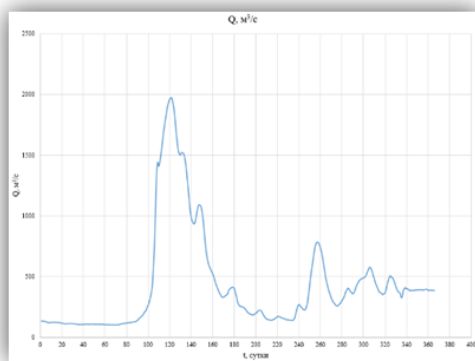


Рис. 7. Изменение расхода  $Q$  в течение одного года, Бондюг, 2008 г.

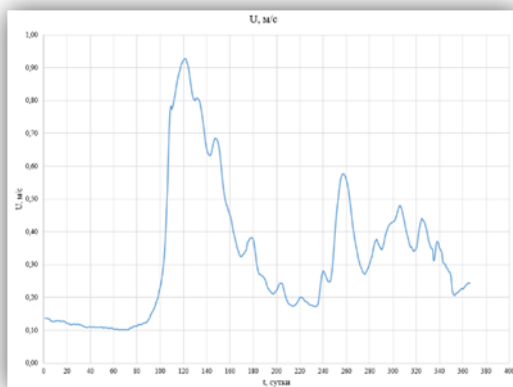


Рис. 8. Изменение средней скорости  $U$  течения реки в течение одного года, Бондюг, 2008 г.



Как можно заметить, все графики показывают одинаковый характер поведения функции, что говорит об однозначной связи этих величин между собой. Также выделим характерный пик роста функции. Он начинается примерно на 90 день, а заканчивается на 220 от начала календарного года и соответствует весеннему половодью. В дальнейшем эта закономерность пригодится для построения «петель скорости».

### Изучение феномена «Петля скорости»

Далее построим графики зависимости средней скорости течения  $U$ (м/с) от расхода  $Q$ (м<sup>3</sup>/с) в период половодья. Определим начало и конец периода половодья по характерному началу и концу роста расхода  $Q$ (м<sup>3</sup>/с). Будем искать начало роста расхода в марте-апреле, а конец в июне-июле. Получим 24 «Петли скорости».

Классифицируем получившиеся петли. Всего удалось выделить 4 вида зависимостей:

- 1) петлеобразная, с нижней ветвью, соответствующей подъему половодья (рис. 9)
- 2) петлеобразная, с нижней ветвью, соответствующей подъему половодья, с «носком» (рис. 10)
- 3) однозначная (рис. 11)
- 4) сложная (рис. 12)

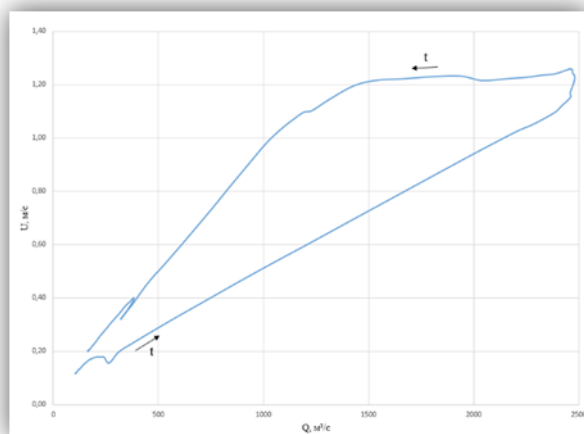


Рис. 9. «Петля скорости», Бондюг, 2013 г.

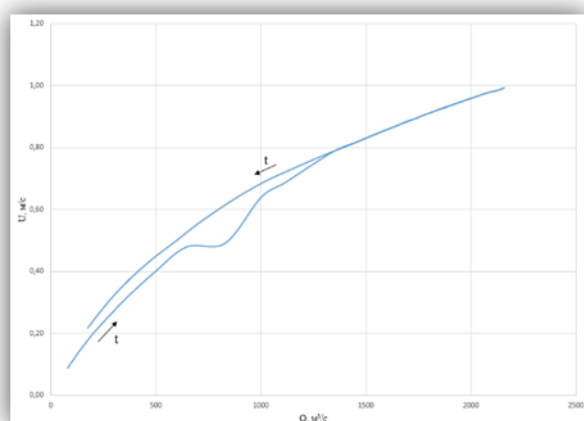


Рис. 10. «Петля скорости», Бондюг, 2012 г.

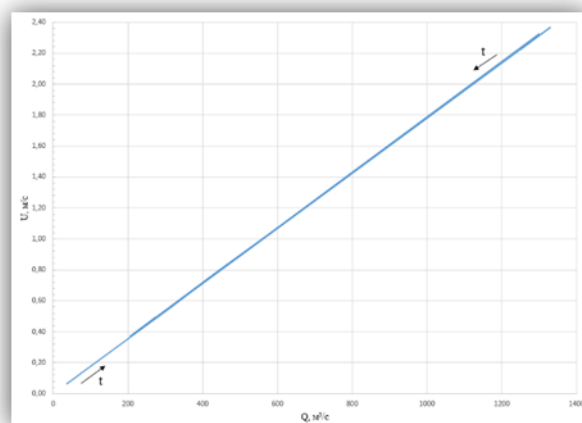


Рис. 11. «Петля скорости», Гайны, 2010 г.

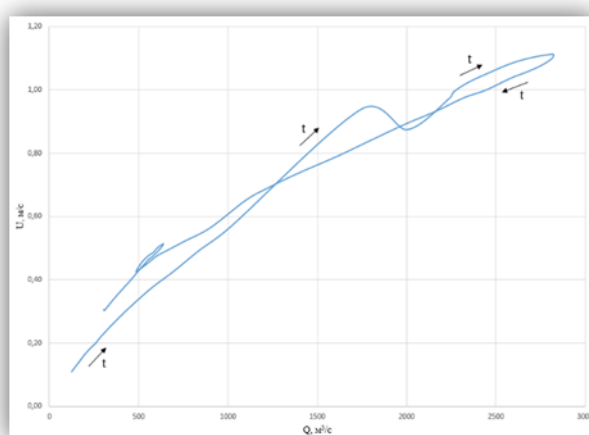


Рис. 12. «Петля скорости», Бондюг, 2009 г.

Исходя из предложенной нами классификации, наблюдаем за период 2008-2015 гг. следующее распределение петель по классам:

Таблица 1. Распределение петель по классам

	2008	2009	2010	2011	2012	2013	2014	2015
<b>Бондюг</b>	2	4	1	2	2	1	2	2
<b>Гайны</b>	3	3	3	3	3	3	3	3
<b>Коса</b>	2	2	2	2	2	2	2	2

Исходя из распределения петель скорости по классам, можно заметить, что преобладают петли 2 класса, а именно петли с нижней ветвью, соответствующей подъему половодья, с «носиком». Данные петли характерны тем, что точка бифуркации не совпадает с точкой, обладающей максимальным значением  $U$  или  $Q$ . Назовем точку бифуркации узловой точкой. Тогда будет иметь место сравнение значений узловых точек со значениями максимальных точек за все годы в рамках одного гидропоста. Очевидно, что будем подвергать сравнению только те русла, где имеются петли 2 класса, т.е. Бондюг и Коса.

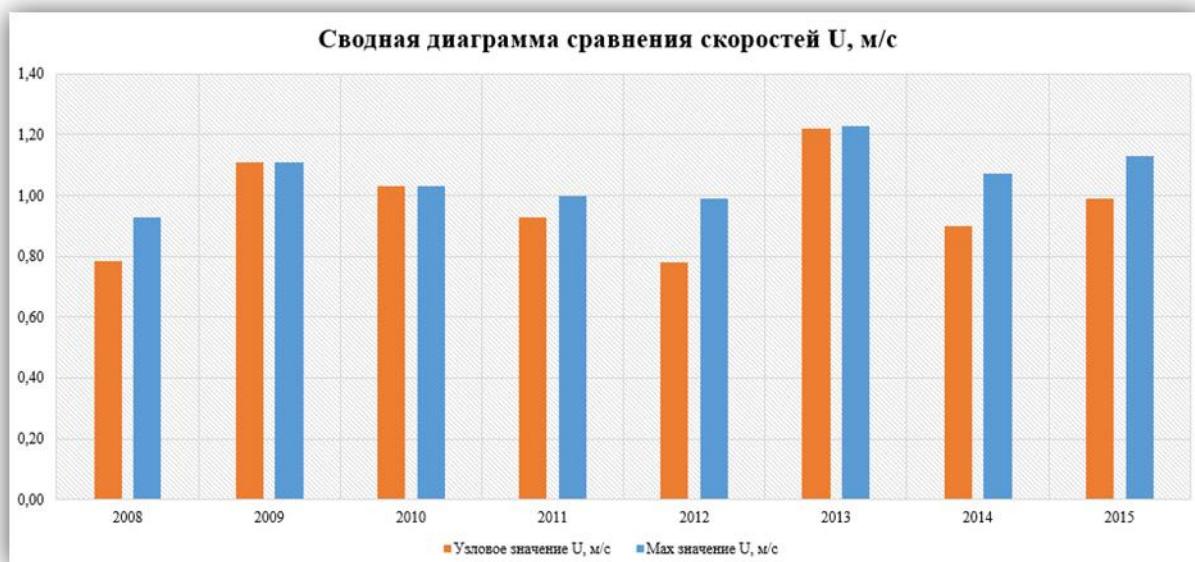


Рис. 13. Сводная диаграмма сравнения скоростей, Бондюг, 2008-2015 гг.



Рис. 14. Сводная диаграмма сравнения скоростей, Коса, 2008-2015 гг.

Проанализировав данные диаграммы (рис. 13, рис. 14), приходим к выводу, что в те года, когда значение скорости в узловой точке было больше или равно 1 м/с, эта точка совпала с максимальной.

Для удобства анализа и изучения характера поведения течения в исследуемых реках следует перейти от абсолютной величины высоты ( $H$ ) к текущей ( $h$ ). Введем понятие самой глубокой точки на данном гидрометрическом створе ( $H_{min}$ ). Тогда

$$h = H - H_{min}$$

Далее все графики будем анализировать, предварительно переходя к текущей величине высоты.

### Поиск обоснования возникновения бифуркации

Одной из важнейших характеристик петли, в которой может храниться ответ на причину возникновения бифуркации, является узловая точка. Чем же она характерна? В первую очередь, ее особенностью является то, что при одном и том же расходе значения скорости потока разнятся. Для того, чтобы понять, почему так происходит, обратимся к теории вязкости.

Вязкость (внутреннее трение) – одно из явлений переноса, свойство текучих тел (жидкостей и газов) оказывать сопротивление перемещению одной их части относительно другой. В результате макроскопическая работа, затрачиваемая на это перемещение, рассеивается в виде тепла.

Различают динамическую вязкость  $\eta$  (единица измерения – паскаль-секунда, Па·с) и кинематическую вязкость  $\nu$  (единица измерения – м<sup>2</sup>/с). Кинематическая вязкость может быть получена как отношение динамической вязкости к плотности вещества и своим происхождением обязана классическим методам измерения вязкости, таким как измерение времени вытекания заданного объема через калиброванное отверстие под действием силы тяжести.

$$\nu = \frac{\eta}{\rho}$$

Изотермическая жидкость стекает по наклонной поверхности под действием силы тяжести. Найти решение в предположении «слоистости» течения. Определить расход через поперечное сечение.

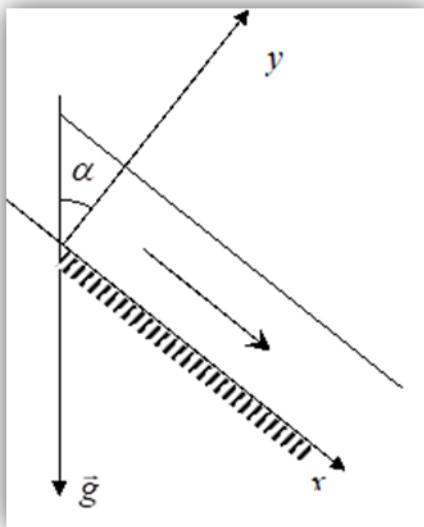


Рис. 15. Задача гидромеханики

Данная задача есть в точности та, исследование которой освещается в данной работе. Ведь любая река течет под действием силы тяжести, а значит под каким-то углом  $\alpha$ . Для её решения используют решение Пуазейля.

Согласно этому решению, расход  $Q$  вычисляется следующим образом:

$$Q_1 = C \frac{H_1^3}{\nu_1}, Q_2 = C \frac{H_2^3}{\nu_2}$$

Выразим из данного уравнения вязкость:

$$v_1 = C \frac{H_1^3}{Q_1}, v_2 = C \frac{H_2^3}{Q_2}$$

Рассмотрим петлю скорости в Бондюге, 2013 г. (рис. 9), т.к. её размах является наибольшим в Бондюге и составляет 0,49 м/с. Тогда  $Q_1$  и  $Q_2$  будут равны, а их значение будет 1220.

$$Q_1 = Q_2 = 1220$$

Уровень воды  $H$  возьмем относительно средней глубины в этот день:

$$H_1 = 5,59075; H_2 = 4,0552$$

Подставим известные значения в формулу, предварительно преобразовав ее

$$C \frac{H_1^3}{v_1} = C \frac{H_2^3}{v_2} \rightarrow \frac{CH_1^3}{CH_2^3} = \frac{v_1}{v_2} \rightarrow \frac{H_1^3}{H_2^3} = \frac{v_1}{v_2}$$

Из получившегося уравнения видим, что изменение уровня воды в реке  $H$  напрямую зависит от изменения вязкости  $v$ . А изменение вязкости, в свою очередь, зависит от изменения температуры, согласно теории вязкости жидкости и газов. Ниже приведена таблица соотношения значений температуры жидкой воды и её вязкости.

температура, °C	динамическая * 10 <sup>-3</sup> Па*с	кинематическая * 10 <sup>-6</sup> (м <sup>2</sup> /с)
0	1,787	1,787
5	1,519	1,519
10	1,307	1,307
20	1,002	1,004
30	0,798	0,801
40	0,653	0,658
50	0,547	0,658
60	0,467	0,475
70	0,404	0,413
80	0,355	0,365
90	0,315	0,326
100	0,282	0,294

Рис. 16. Значения динамической и кинематической вязкости жидкой воды при разных температурах

Подсчитаем значение отношения кубов уровней воды:

$$\frac{H_1^3}{H_2^3} = \frac{5,59075^3}{4,0552^3} \rightarrow \frac{H_1^3}{H_2^3} = \frac{174,7472}{66,6863} \rightarrow \frac{H_1^3}{H_2^3} = 2,6204$$

Согласно данным о температуре в этом районе, на момент  $v_1$  температура воздуха была равна 5,1°C, а на момент  $v_2$  – 15,7°C. Воспользовавшись онлайн ресурсом, был произведен расчёт величин вязкости, который показал, что  $v_1 = 1,515$ ;  $v_2 = 1,134$ .

Теперь подсчитаем значение отношения вязкостей:

$$\frac{v_1}{v_2} = \frac{1,515}{1,134} \rightarrow \frac{v_1}{v_2} = 1,33598$$

К сожалению, из-за недостатка информации и наличия большой погрешности в данных, а также многих принятых допущений, можно заметить, что значения отношений довольно сильно разнятся, в то время как должны быть равны. Данная разница заметно велика, в связи с чем увеличивается и разница в скорости течения реки в разные моменты времени при одном и том же  $Q$ .

### Заключение

Целью данной работы являлось исследование феномена под названием «Петля скорости» на участке реки Кама от гидрологического поста (далее гидропост) Гайны до гидропоста Бондюг, включая промежуточный гидропост на реке Коса на основе гидрологических показаний, измеренных соответствующими гидропостами.

В ходе работы были изучены методы, с помощью которых измеряют важные гидродинамические характеристики на гидропостах. На основе имеющихся показаний гидропостов были построены «петли скорости», введены важные термины и определения, проанализированы и классифицированы петли, а также была найдена причина возникновения бифуркации. Для формирования наборов данных реализована и описана программа на языке C#, а также построены графики, диаграммы и таблицы с помощью Microsoft Excel.

Цель, поставленная в начале работы, была достигнута, а проведенные исследования и полученные результаты могут быть полезны при разработке моделей для прогнозирования гидрологических показаний.

### Библиографический список

1. Чалов Р.С. Русловые процессы (Русловедение) / Инфра-М, 2015. – 110-112 с.
2. Механизм течения рек [Электронный ресурс]. – URL: <http://abratsev.ru/hydrosphere/mechanizm.html> (Дата обращения: 26.09.2021)
3. Мазуркин П.М., Колесников А.П. Способ гидрометрических измерений профиля дна и поймы реки / Журнал «Современные наукоемкие технологии.» – 2013. – № 9 – С. 93-98
4. Измерения уровня воды [Электронный ресурс]. – URL: <http://geoman.ru/books/item/f00/s00/z0000073/st044.shtml> (Дата обращения: 26.09.2021)
5. Методы измерения скорости течения в реке или канале [Электронный ресурс]. – URL: [http://www.xn--80aicmxhn.xn--p1ai/info/articles/izmerenie\\_skorostei\\_v\\_rekax.htm](http://www.xn--80aicmxhn.xn--p1ai/info/articles/izmerenie_skorostei_v_rekax.htm) (Дата обращения: 26.09.2021)
6. Масумов Р.Р. Методы измерения расхода воды на реках и каналах, в напорных трубопроводах насосных станций и оросительных систем / Библиотека водника, выпуск №11 – Ташкент 2015.– С. 5-15

### References

1. CHalov R.S. Ruslovye processy (Ruslovedenie) / Infra-M, 2015. – 110-112 s.
2. Mekhanizm techeniya rek [Electronic resource]. – URL: <http://abratsev.ru/hydrosphere/mechanizm.html> (Data obrashcheniya: 26.09.2021)
3. Mazurkin P.M., Kolesnikov A.P. Sposob gidrometricheskikh izmerenij profilya dna i pojmy reki / ZHurnal «Sovremennye naukoemkie tekhnologii.» – 2013. – № 9 – S. 93-98

4. Izmereniya urovnya vody [Electronic resource]. – URL: <http://geoman.ru/books/item/f00/s00/z00000073/st044.shtml> (Data obrashcheniya: 26.09.2021)
5. Metody izmereniya skorosti techeniya v reke ili kanale [Electronic resource]. – URL: [http://www.xn--80aicmxhn.xn--p1ai/info/articles/izmerenie\\_skorostei\\_v\\_rekax.htm](http://www.xn--80aicmxhn.xn--p1ai/info/articles/izmerenie_skorostei_v_rekax.htm) (Data obrashcheniya: 26.09.2021)
6. *Masumov R.R.* Metody izmereniya raskhoda vody na rekah i kanalah, v napornyh truboprovodah nasosnyh stancij i orositel'nyh sistem / Biblioteka vodnika, vypusk №11 – Tashkent 2015.– S. 5-15

## **RESEARCH OF SPRING FLOOD REGIMES ON KAMA AND KOSA RIVERS BASED ON HYDROLOGICAL DATA**

*Sofronov Anatoliy M., Rusakov Sergey V.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [anatoliy-sofronov@mail.ru](mailto:anatoliy-sofronov@mail.ru)

This paper solves the problem of research and study of the phenomenon called «Speed Loop» based on hydrological data of the rivers Kama and Kosa. After the introduction of the terminology necessary for the study, initial analysis of the initial data is carried out, the causes of the phenomenon are determined, possible variants are classified, and the nature of its manifestation is studied. As a result of the classification, 4 types of velocity loops were identified. A rationale for the emergence of this phenomenon was proposed, based on Poiseuille's equation. Microsoft Office and Visual Studio were used for the study. The models were based on a set of 8,766 hydrological observations.

**Keywords:** modeling, flood, speed loop, classification of speed loops, Poiseuille equation, bifurcation point.



## **ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПЕРМСКИЕ БЫЛИЧКИ. МИФОЛОГИЧЕСКИЕ РАССКАЗЫ 19-20 ВВ»**

*Филиппов Степан Игоревич, Аверин Сергей Игоревич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, pirogov1923@mail.ru

Рассматривается проектирование, документирование и разработка информационной системы «Пермские былички». Былички являются важной частью культурного наследия человечества. Чтобы иметь представление о культуре людей, живших раньше, былички необходимо собирать в специальных экспедициях, записывать и анализировать. Для выполнения этих целей была создана данная система. Обычным людям она позволяет ознакомиться с быличками, а тем, кто хочет подробнее их изучить, провести лексический анализ. В начале были сформулированы требования к системе. Затем были выбраны инструменты для работы. Потом был описан процесс моделирования и разработки. По итогам была спроектирована и разработана информационная система, которая хранит былички, позволяет добавлять в систему новые былички, изменять имеющиеся, а также проводить их анализ.

Ключевые слова: Пермские былички, анализ, проектирование, разработка, информационная система.

Мифы – это важная часть культурного наследия человечества, изучение которых может помочь понять наши корни и культурные традиции. В них объясняется происхождение мира, природы, мифических существ. В мифах заключаются ценные знания о традициях, обычаях, морали и этике, которые передаются из поколения в поколение. Кроме того, мифы – это важный источник вдохновения для литературы, искусства и кинематографа. Мифы становятся основой для создания новых произведений, которые в свою очередь влияют на мировоззрение людей. Некоторые профессии занимаются их изучением, для этого они используют средства лингвистического анализа. Этот процесс можно ускорить с помощью компьютера.

На основе этой информации можно сделать вывод, что создание информационной системы для сохранения, анализа и изучения мифов является актуальной. Такая информационная система будет полезна не только для ученых и исследователей, но и для широкой публики, которая сможет легко получить доступ к информации о мифах знакомой для них или нет культуры.

Основным источником данных для информационной системы выступила книга А. В. Черных Куединские былички. В ней описано множество быличек, которые были собраны в разных экспедициях по территории Пермского края [1].

Исходя из анализа имеющихся источников, были сформулированы основные требования к системе:

1. Пользователи должны иметь возможность просматривать рассказы;
2. Рассказы должны быть структурированы по категориям;
3. Рассказы должны иметь, соответствующие им тэги;
4. Пользователь должен иметь возможность отправки своего рассказа;
5. Пользователь должен иметь возможность просмотра статуса своего рассказа;



6. Пользователь с правами должен иметь доступ к автоматизированному анализу рассказов;
7. Администратор должен иметь возможность работы с рассказами;
8. Администратор должен иметь возможность работы с правами доступа.

Помимо этого, на основе требований было принято решение, что система будет иметь клиент-серверную архитектуру и будет являться сайтом.

Далее был произведен выбор средств для реализации.

В качестве средства моделирования был выбран инструмент diagrams.net, так как он позволяет разрабатывать UML диаграммы, распространяется полностью бесплатно, а также поддерживает большое количество платформ.

Для прототипирования была выбрана Figma, так как она является полностью бесплатным решением, имеет большой функционал и может работать как в настольной версии, так и в облачной.

Разработка клиентской части была выполнена с использованием HTML 5, CSS 3 и фреймворка Vue 3, так как он имеет низкий порог вхождения, большой функционал и легок в развертывании [2].

Разработка серверной части была выполнена на языке PHP и его фреймворк Laravel 9, так как он имеет хорошую документацию и имеет простой синтаксис [3].

Для анализа рассказов использовался язык Python, так как он имеет множество библиотек для лингвистического анализа.

Работа с базой данных будет осуществляться при помощи СУБД MySQL, так как она является бесплатной, имеет большой функционал, высокую производительность и проста в использовании.

Далее было произведено моделирование информационных процессов. В результате было создано три диаграммы прецедентов, две диаграммы активности, диаграмма последовательности и диаграмма классов.

На основе составленных моделей была произведена разработка информационной системы с использованием ранее выбранных технологий. В основе системы лежит база данных (см. рис. 1).

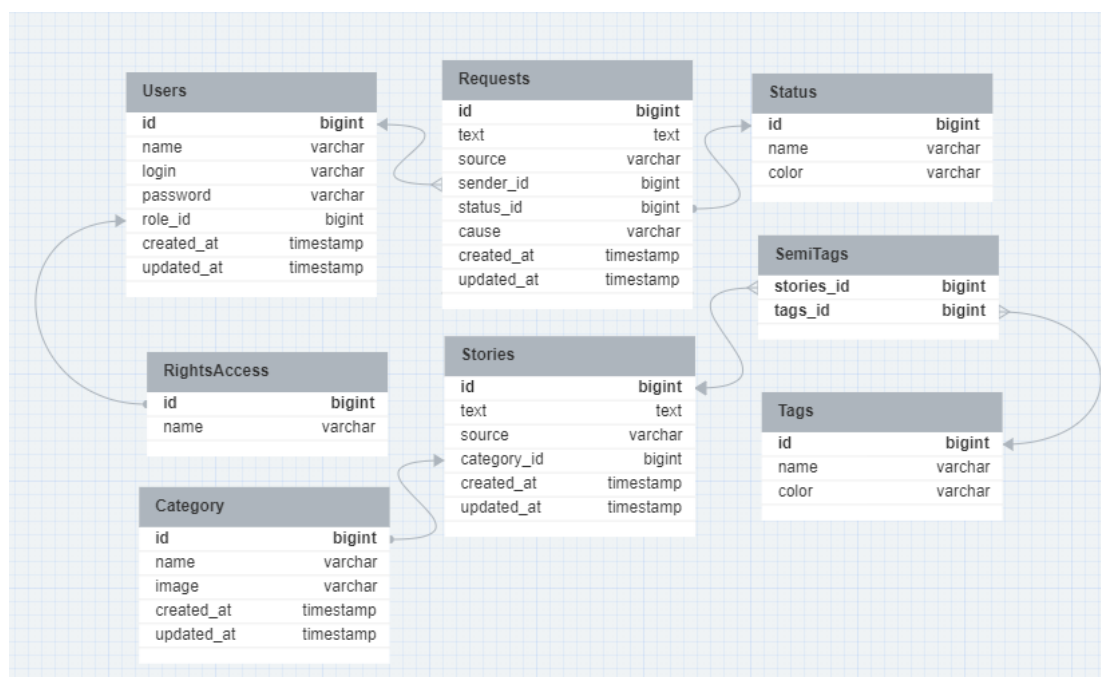


Рис. 1. Структура базы данных

Работа системы проходит следующим образом.

Пользователи могут добавлять свои рассказы в систему. Для этого им надо отправить рассказ на соответствующей странице. Далее администратор должен подтвердить её добавление или отклонить с указанием причины.

Пользователь может отсортировать рассказы по специальным тэгам, которые имеются у каждого рассказа, чтобы видеть только желаемые.

Администратор может добавлять, изменять и удалять рассказы, категории рассказов, тэги.

Пользователь может провести анализ рассказов, которые он выберет. Он получит результат частотного анализа в виде облака слов, где размер шрифта определяет частоту встречи слова. Также результат анализа тональности, который покажет положительный, нейтральный или отрицательный фон имеет рассказ.

По итогу работы была спроектирована и реализована система «Пермские былички», которая позволит ознакомиться с быличками и рассказами, а также провести их анализ.

### **Библиографический список**

1. *Черных А. В.* Куединские былички. Пермь, ПОНИЦАА, 2004
2. Vue 3 [Электронный ресурс] URL: <https://vuejs.org/> (Дата обращения 04.04.2023)
3. Laravel [Электронный ресурс] URL: <https://laravel.com/> (Дата обращения 05.04.2023)

### **References**

1. *Chernykh A. V.* Kuedinskie bylichki. Perm, PONITSAA, 2004.
2. Vue 3 [Electronic resource]. – URL: <https://vuejs.org/>
3. Laravel [Electronic resource]. – URL: <https://laravel.com/>

## **DESIGNING, DOCUMENTING, AND DEVELOPING THE INFORMATION SYSTEM «PERM BYLICHKAS. MYTHOLOGICAL STORIES OF THE 19TH-20TH CENTURIES»**

*Filippov Stepan I., Averin Sergey I.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, pirogov1923@mail.ru

The design, documentation and development of the information system «Perm bylichki» are considered. Bylichkas are an important part of the cultural heritage of mankind. In order to have an idea about the culture of people who lived in the past, folk tales should be collected in special expeditions, recorded and analyzed. This system was created to fulfill these purposes. It allows ordinary people to get acquainted with the bylaws, and for those who want to study them in more detail, to carry out a lexical analysis. In the beginning, the requirements for the system were formed. Then the tools to work with it were chosen. Then the modeling and development process was described. As a result, an information system was designed and developed that stores the skits, allows adding new skits to the system, modifying existing ones, as well as analyzing them.

Keywords: Permian bylichki, analysis, design, development, information system.

## **ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ЭЛЕКТРОННОГО ГОЛОСОВАНИЯ С ВОЗМОЖНОСТЬЮ ПРОВЕРКИ РЕЗУЛЬТАТА ИЗБИРАТЕЛЯМИ**

*Филиппов Александр Сергеевич, Раевский Виктор Николаевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, afilipov41@yandex.ru

В данной статье будет рассмотрен функционал приложения для электронного голосования с возможностью проверки результата избирателями. Приложение, разработанное в рамках выпускной квалификационной работы, предоставляет возможность избирателям анонимно голосовать, однако это не единственное преимущество данного приложения. Защита от подмены голоса в приложении будет основана на технологии Блокчейна: «Все будут знать, как проголосовали, но никто не будет знать, кто как проголосовал». Такая структура исключит возможность подмены голоса, потому что каждый последующий голос будет ссылаться на предыдущий, что и обусловит надёжность системы голосования. Анонимность голосования обеспечивается двухэтапным процессом регистрации результата с разделением факта голосования конкретного пользователя и результата голосования уже без привязки к конкретному пользователю.

Ключевые слова: электронное голосование, мобильное приложение, клиент-серверное взаимодействие, информационная система, UML-диаграммы.

Любой человек вне зависимости от типа выборов, всегда желает, чтобы его голос не просто учли, но ещё и учли правильно. В то время, как в динамично развивающемся мире технологии покоряют все сферы жизни человека, в области голосования также требуются технологии, которые позволят любому человеку отдать свой голос при том или ином голосовании.

Здесь появляются некоторые проблемы. Во-первых, избиратель может проголосовать только один раз. Если дать избирателю возможность проголосовать несколько раз, он может проставить голоса таким образом, чтобы его вариант оказался выигрышным. Во-вторых, важно, чтобы голос избирателя был учтён таким, какой он сам пожелал. В таком случае должен быть полностью исключён вариант подмены голоса. В-третьих, необходимо не ограничивать мобильность избирателя – он должен иметь возможность проголосовать где угодно. В-четвёртых, голосование должно быть анонимным – у пользователя должна быть возможность сохранения тайны своего голоса.

В связи с вышеперечисленными пунктами, идеальной возможностью сочетать в себе все эти требования является разработка мобильного приложения электронного голосования. Данное приложение может помочь решить все эти проблемы, но прежде, чем его разработать, необходимо детально исследовать все важные нюансы, смоделировать функционал приложения, а также привести необходимую документацию.

Целью данной работы является создание приложения электронного голосования, которое будет максимально эффективно в себе сочетать следующие свойства:

- одноразовость голосования;

- гарантированный учёт голоса;
- мобильность избирателя;
- анонимность голосования.

### Выбор методологии моделирования

Для того, чтобы успешно смоделировать предметную область, необходимо подобрать соответствующую методологию моделирования, которая в полной мере сможет описать возможности системы.

Среди разнообразных методологий моделирования диаграмм, выбраны три самых популярных – UML, BPMN и IDEF0. Рассмотрение предоставляемых ими возможностей позволяет сделать вывод, что благодаря разнообразию используемых диаграмм, в большей степени подходит UML. Данная методология позволит максимально ёмко описать проектируемую систему. Также, помочь с описанием системы может и BPMN – специальная нотация для моделирования бизнес-процессов. С её помощью можно построить диаграмму активностей. В UML такая диаграмма также есть, однако у неё есть один несущественный минус – в ней все действия обобщены и не поясняются конкретной областью действия. Поэтому, диаграмма активностей строится с помощью BPMN. Что касается методологии IDEF0, то данная работа не нуждается в этом варианте диаграмм, т.к. UML и BPMN смогут описать её в полной мере.

### Проектирование информационной системы

Диаграмма классов (англ. Class diagram) – структурная диаграмма UML, которая демонстрирует общую структуру иерархии классов проектируемой системы, атрибутов, методов, интерфейсов и взаимосвязей между ними [2]. Диаграмма классов – основная единица объектно-ориентированного моделирования.

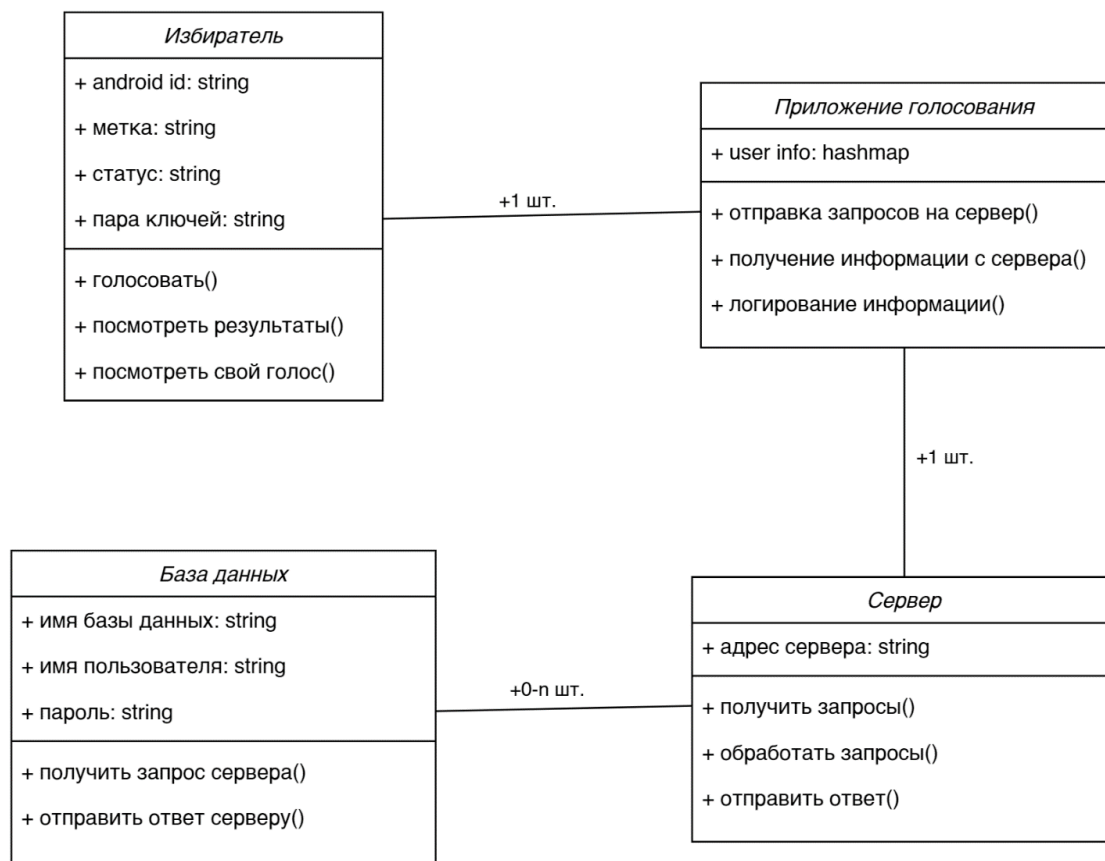


Рис. 1. Диаграмма классов

Рассмотрим диаграмму классов (рис. 1). На ней изображены задействованные классы в предметной области: Избиратель, Приложение голосования, Сервер и База данных. Соответственно, у *Избирателя* есть следующая необходимая информация: Android ID – который нужен для того, чтобы пользователь не смог голосовать дважды, метка – которая позволит сопоставить закрытый ключ сервера к текущему пользователю, статус – некоторая метка действия пользователя, чтобы сервер смог понять, какой этап проходит избиратель, и его пара ключей для шифрования информации. Функции, доступные избирателю, будут описаны в диаграмме прецедентов. У *Приложения* голосования в атрибутах есть информация пользователя – JSON файл с необходимой логированной информацией. Приложение может отправлять, получать и анализировать данные с сервера. В свою очередь *Сервер* имеет адрес – для того, чтобы приложение смогло подключиться, а также он может получать и обрабатывать запросы у Избирателя и Базы данных, и отправлять ответ Избирателю. При своей работе, Сервер подключается к *Базе данных*, у которой есть имя базы данных, имя пользователя и пароль. База данных получает запрос сервера и отправляет ему ответ.

Диаграмма прецедентов (англ. Use-case diagram) – это диаграмма, которая отражает отношения между акторами и прецедентами и позволяет описать систему на уровне концепции [3]. В свою очередь прецедент – это часть функциональности системы, позволяющей пользователю получить конкретный нужный ему результат. Он же определяет один из вариантов её использования и описывает типичный способ вариантов взаимодействия пользователя с системой.

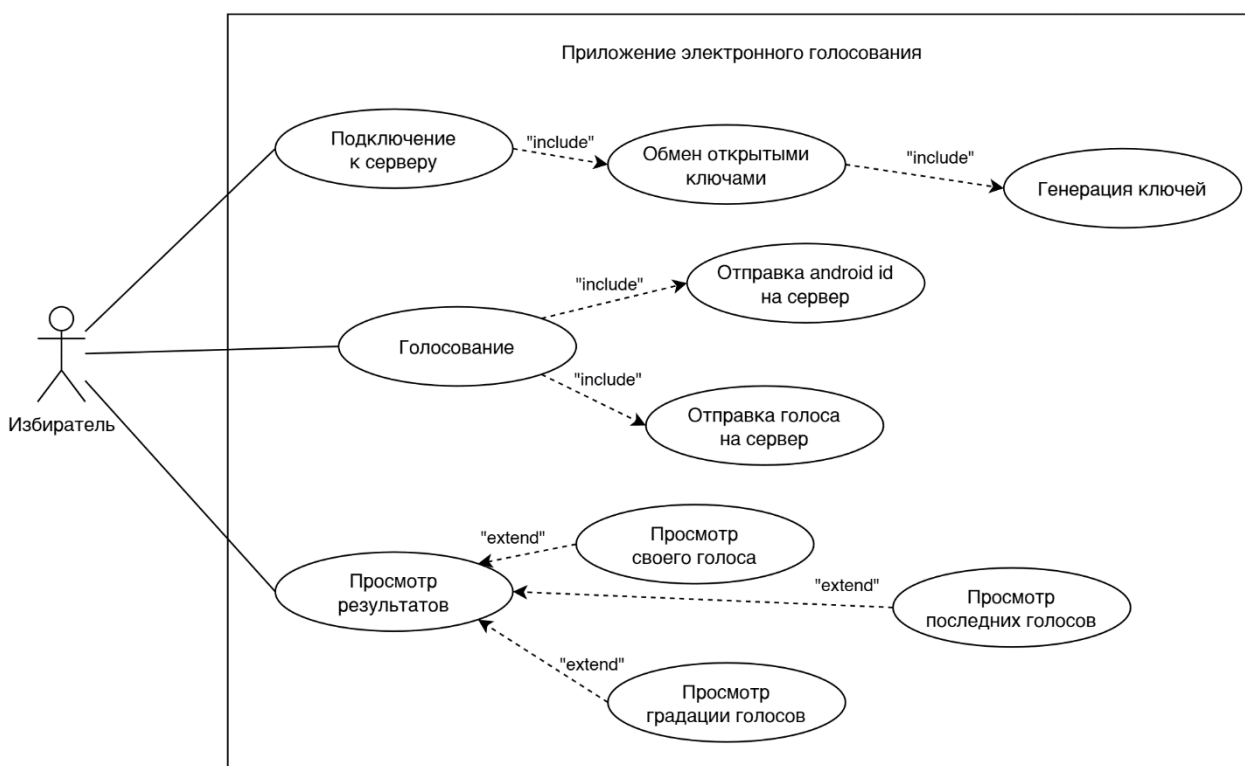


Рис. 2. Диаграмма прецедентов

Рассмотрим диаграмму прецедентов (рис. 2). В ней описаны действия *Избирателя* и то, что может и выполняется одновременно с ними. Рассмотрим по порядку:

6. Подключение к серверу. Действие приписано Избирателю, потому что он вручную нажимает кнопку «Подключиться». Во время подключения выполняется обмен открытыми ключами, которые, в свою очередь, сначала генерируются.

7. Голосование. Во время голосования пользователя, серверу отдельно отправляется Android ID и голос. Разделение призвано наглядно показать, что эти две функции выполняются отдельно друг от друга и не могут выполняться одновременно из соблюдения условия «Анонимность».

8. Просмотр результатов. Во время просмотра результатов есть расширяющие этот функционал действия. Пользователь может просмотреть свой голос, последние голоса из базы данных, а также зайти на вкладку всех голосов и просмотреть их градацию от самого популярного варианта к менее популярному.

### **Обоснование выбора инструментов разработки системы**

Решение разработать мобильное приложение для Android обусловлено распространённостью этой операционной системы из соображений обеспечения условия мобильности максимально широкому кругу избирателей. В то время, как не у всех дома есть компьютеры, у большинства есть смартфоны или планшеты и большинству будет удобно голосовать именно с них – даже если они находятся в движении.

В качестве языка программирования, на котором будет разрабатываться приложение выбран язык Java. Выбор обусловлен его возможностями: он является одним из самых популярных языков программирования и для него написано много упрощающих работу библиотек.

Языком программирования для сервера выбран PHP. PHP – самый популярный язык взаимодействия с базой данных, для него также есть много полезных возможностей, расширяющих функционал.

### **Описание функционала приложения**

Рассмотрим возможности приложения с точки зрения пользователя (избирателя).

После открытия приложения появляется приветственное окно. После того, как пользователь попытается войти в систему по нажатию кнопки «Подключиться», произойдут два последовательных подключения к серверу.

При первом подключении считывается Android ID – это уникальный идентификатор для каждого устройства. Он используется при идентификации устройства для загрузки с рынка определенных игровых приложений, которым необходима идентификация устройства (чтобы они знали, что это устройство использовалось для оплаты приложения) и тому подобное [1]. Помимо Android ID, генерируется пара ключей для клиента – открытый и закрытый. На сервере, после подключения, обрабатываются отправленные данные. В первую очередь проверяется по базе данных, голосовал ли данный Android ID раньше или нет. Если он голосовал, то сервер отправляет ответ клиенту, который перенаправляет на страницу с результатами. Если же избиратель ещё не голосовал, то данные продолжают обрабатываться: на стороне сервера генерируется пара ключей уже для сервера, все полученные и сгенерированные данные сохраняются в базу данных, а открытый ключ сервера отправляется клиенту (избирателю). После получения ответа от сервера, все сгенерированные и полученные данные сохраняются на устройство клиента.

При втором подключении у клиента генерируется уникальная метка – позже она пригодится для того, чтобы проверить возможность голосования, а также привязки закрытого ключа сервера к данной метке для того, чтобы расшифровать голос уже без Android ID. Метка зашифровывается открытым ключом сервера, и отправляется на сервер в зашифрованном виде. После получения метки, сервер подписывает её, причём саму подпись зашифровывает

своим же открытым ключом, чтобы пользователь не смог её расшифровать и изменить данные. После получения подписи метки, приложение перенаправляет избирателя на экран голосования.

Избиратель выбирает, за что хочет проголосовать. Затем, подтвердив свой голос, логируются данные и инициализируется третье подключение. В нём клиент отправляет серверу Android ID и сообщение о том, что клиент уже готов отправить голос. Сервер заносит в базу данных информацию о том, что данный Android ID уже проголосовал. Клиенту отправляется сообщение об успехе. То, что голос не отправляется вместе с Android ID – обеспечивает анонимность голосования, а занесение его в базу данных – исключит возможность голосования в дальнейшем.

При четвёртом подключении происходит уже учёт голоса пользователя. Серверу отправляется метка. С её помощью сервер узнает, во-первых, о подписи, а во-вторых, о том, каким закрытым ключом сервера нужно расшифровать метку и голос. Также серверу отправляется сам голос, зашифрованный открытым ключом сервера. Сервер расшифровывает полученные данные, анализирует подпись метки, и после успеха – заносит голос в базу данных с ссылкой на предыдущий голос. Это позволит исключить возможность подмены голоса. Клиенту отправляются данные для демонстрации предварительных результатов.

После всех подключений пользователю открывается возможность просмотреть свой голос, последние голоса, а также градацию выбора голосов от наиболее часто выбираемого варианта до наиболее редкого варианта без возможности просмотра конкретных цифр.

### **Заключение**

Разработанное приложение электронного голосования позволит упростить процесс голосования, а также обеспечит его самыми необходимыми свойствами в этой сфере: мобильность, анонимность, гарантированность и одноразовость учёта голоса. Все эти свойства позволят любому избирателю голосовать отсюда, откуда им будет удобно, обеспечит тайну их голоса – потому что голос, который они отправляют, будет отправляться отдельно от идентификатора Android, гарантированность голоса – система учтёт голос избирателя, а также один пользователь сможет проголосовать только один раз.

### **Библиографический список**

1. Что такое идентификатор Android? – BlogRound [Электронный ресурс]. – 2023. – URL: <https://bloground.ro/ru/что-такое-идентификатор-android> (дата обращения 14.04.2023)
2. Проектирование диаграммы классов UML (Class Diagram) – Национальная сборная Worldskills Россия [Электронный ресурс]. – 2023. – URL: <https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-klassov-uml-class-diagram/> (дата обращения 14.04.2023).
3. Диаграмма прецедентов [Электронный ресурс]. – 2023. – URL: <https://oktmo.ru/stati/24090-diagramma-precedentov.html> (дата обращения 14.04.2023)

### **References**

1. What is an Android ID? – BlogRound [Electronic resource]. – 2023. – URL: <https://bloground.ro/ru/что-такое-идентификатор-android> (date of appeal 14.04.2023)
2. UML Class Diagram Design – National team Worldskills Russia [Electronic resource]. – 2023. – URL: <https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-klassov-uml-class-diagram/> (date of appeal 14.04.2023).

3. Precedent Chart [Electronic resource]. – 2023. – URL: <https://oktmo.ru/stati/24090-diagramma-precedentov.html> (date of appeal 14.04.2023)

## **DESIGN, DOCUMENTATION AND DEVELOPMENT OF AN APPLICATION FOR ELECTRONIC VOTING WITH THE POSSIBILITY OF VERIFICATION OF THE RESULT BY VOTER**

*Filippov Aleksandr S., Rayevsiy Viktor N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [afilippov41@yandex.ru](mailto:afilippov41@yandex.ru)

This article will consider the functionality of the electronic voting application with the ability to check the result by voters. The application developed as part of the graduation qualification work provides an opportunity for voters to vote anonymously, but this is not the only advantage of this application. Protection against voice swapping in the app will be based on Blockchain technology: "Everyone will know how they voted, but no one will know who voted how." Such a structure will exclude the possibility of voting substitution, because each subsequent vote will refer to the previous one, which will determine the reliability of the voting system. The anonymity of voting is ensured by a two-stage process of registering the result with the division of the fact of voting of a specific user and the result of voting without binding to a specific user.

**Keywords:** electronic voting, mobile application, client-server interaction, information system, UML diagrams.



## **РАЗРАБОТКА СИСТЕМЫ АНАЛИЗА И ПРОЕКТИРОВАНИЯ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВЫСШЕГО ОБРАЗОВАНИЯ НА БАЗЕ ОНТОЛОГИЧЕСКОГО ПРЕДСТАВЛЕНИЯ КОМПЕТЕНЦИЙ И ДИСЦИПЛИН**

*Черемных Александра Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, rain.raicker@yandex.ru

Рассматриваются автоматизация составления индивидуальной образовательной траектории (ИОТ) на основе предпочтений и интересов студента, а также автоматизация составления проекта учебного плана (УП). Работа содержит описание основных понятий по теме исследования, обоснование актуальности ИОТ, подробный обзор проблем, которые она решает полностью или частично: устаревшая учебная программа, низкая вовлеченность студентов, высокий уровень отсева студентов. Рассматривается краткая теория по онтологиям, выявления требований к онтологии, обоснование ручному заполнению онтологии. Описаны алгоритмы построения ИОТ и проекта УП.

Ключевые слова: индивидуальная образовательная траектория, учебный план, онтология, информационная система.

### **Введение**

В связи со стремительным развитием технологий в непрерывно меняющемся окружающем мире, люди постоянно нуждаются в новой информации для актуализации своих знаний. Особенно это касается специалистов в сфере IT-технологий, где только постоянное совершенствование своих знаний, поможет работнику оставаться конкурентоспособным. Подобные обстоятельства вынуждают высшие учебные заведения преобразовывать свои образовательные программы и учебные планы и подстраиваться под современный мир, где особой ценностью для студента является возможность построения индивидуальной траектории обучения, которая будет соответствовать его интересам и будет практически направленной.

Составление учебного плана (УП) – это крайне важный и трудоёмкий процесс, который пока что не полностью автоматизирован и требует значительного количества ресурсов, включая время, опыт и финансирование. Чтобы студент во время обучения получил максимум пользы от изучения дисциплин, необходимо поставить их в УП в логически правильном порядке. Руководителю образовательных программ (ОП) не всегда удается это сделать в связи с ограниченными ресурсами.

По той же причине, в соответствии с федеральным государственным образовательным стандартом (ФГОС), где предусмотрены дисциплины по выбору (элективы), студенту бывает сложно сделать выбор дисциплин, которые бы он был способен и хотел изучить. Также еще одна проблема заключается в том, что профессиональные элективы выбираются только в пределах одного направления, в результате чего полные группы могут не набираться, а электив не проводится, что не позволяет некоторым студентам в полной мере изучать то, что требует рынок труда. Автоматизация построения индивидуальных образовательных траекторий позволила бы унифицировать набор элективов и решить указанную проблему.

Таким образом, разработка информационной системы (ИС) для составления проектов учебных планов и индивидуальных траекторий становится очень актуальной. Разрабатываемая ИС будет интеллектуальной и основанной на онтологии о знаниях (дисциплинах и компетенциях).

### **Основные понятия по теме исследования**

Образовательные программы (ОП) высшего образования – программы бакалавриата, программы специалитета, программы магистратуры, программы подготовки научных и научно-педагогических кадров в аспирантуре (адъюнктуре), программы ординатуры, программы ассистентуры-стажировки [1].

Учебный план (УП) – документ, который определяет перечень, трудоемкость, последовательность и распределение по периодам обучения учебных предметов, курсов, дисциплин (модулей), практики, иных видов учебной деятельности и, если иное не установлено Федеральным законом об образовании, формы промежуточной аттестации обучающихся [2]. Учебный план является частью образовательной программы.

Федеральные государственные образовательные стандарты (ФГОС) – это совокупность требований, обязательных при реализации основных образовательных программ начального общего, основного общего, среднего (полного) общего, начального профессионального, среднего профессионального и высшего профессионального образования образовательными учреждениями, имеющими государственную аккредитацию [3].

Для выявления требований к образовательным программам был изучен ФГОС высшего образования (ВО) 3++ по направлению подготовки 01.03.02 «Прикладная математика и информатика» [4]. Будут учтены требования только для очной формы обучения. По результатам анализа ФГОС ВО 3++ были выделены требования для формирования учебного плана, которые показали, что при построении УП нужно учитывать множество факторов, поэтому этот процесс является очень трудоемким. В данной работе будет сделан акцент на логически правильном порядке изучения дисциплин, и будет построен проект учебного плана. Под проектом УП будем подразумевать набор дисциплин, расположенных в логически правильном порядке, разбитый на периоды обучения.

### **Понятие индивидуальной образовательной траектории**

Термин "индивидуальная образовательная траектория" (ИОТ) не имеет общепринятого определения и может трактоваться по-разному в зависимости от контекста и заинтересованных сторон. Например, некоторые учебные заведения могут определять ИОТ как набор курсов, которые студенты могут выбрать, в то время как другие могут предлагать более гибкое проведение обучения в виде самостоятельного составления своего учебного плана, выбора различных онлайн-курсов, стажировок или исследовательских проектов. Далее рассмотрим предлагаемые определения ИОТ разными исследователями.

Н.Н. Суртаева [5] описывает ИОТ как последовательность учебной деятельности учащегося для достижения им образовательных целей в соответствии с его способностями, возможностями, мотивацией, интересами, которая осуществляется при консультирующей деятельности педагога, во взаимодействии с родителями.

И.Н. Бухтиярова [6] рассматривает ИОТ как проявление стиля обучения студента, соответствующего его мотивации и осуществляемого в сотрудничестве с преподавателем.

Т.А. Тимошина [7] рассматривает ИОТ как индивидуальный путь в образовании, определяемый учеником совместно с учителем и организованный с учетом различных факторов,

таких как психические, психологические и физиологические особенности обучающегося, а также социально-экономические и временные возможности субъекта образовательного процесса.

А.В. Хуторской [8] рассматривает ИОТ как личностный способ реализации потенциала каждого ученика в образовании и определяет этапы деятельности ученика как цели-план-деятельность-рефлексия-сравнение полученных продуктов с целями-самооценка.

Таким образом, из разных определений ИОТ можно сделать вывод о том, что внедрение ИОТ в вузы – это достаточно сложный процесс. Несмотря на общий образовательный стандарт, в реальной жизни каждый университет является уникальным и требует индивидуального подхода к изменениям.

В исследовании [9], которое было проведено в 2021 году, на вопрос о том, какой способ индивидуализации обучения будет наиболее эффективен, большинство студентов выбрали дисциплины по выбору. Также данный способ внедрения ИОТ является одним из самых доступных для вузов.

Приведенные выше определения ИОТ показывают, что в основном она рассматривается как форма организации учебной деятельности. В данной работе под понятием ИОТ будем подразумевать набор элективов, взаимосвязанных друг с другом, который будет формироваться на основе предпочтений и интересов студента.

Актуальность ИОТ обоснована тем, что она может полностью или частично решить 3 проблемы, приведенные ниже, которые есть во многих вузах России:

1) Устаревшая учебная программа:

Качество услуг, предоставляемых вузами, можно оценить по конкурентоспособности их будущих выпускников [10]. В наше время появляется все больше новых профессий, требующих обучения. Причем эти профессии очень быстро рождаются и так же стремительно умирают. Традиционное высшее образование не всегда успевает за быстрыми темпами смены профессий, в результате чего выпускники могут оказаться невостребованными на рынке труда и в итоге работать не по специальности.

Также в марте 2019 было проведено исследование [11], где выпускники разных ВУЗов выбрали факторы неудовлетворенности высшим образованием. И наиболее важным фактором для них стали устаревшие учебные программы.

Внедрение ИОТ может решить эту проблему, сократив количество выпускников, которые в итоге работают не по специальности, и помочь студентам избежать получения квалификации, которая может стать неактуальной.

2) Низкая вовлеченность студентов:

Студенческая вовлеченность – это степень готовности студента участвовать в повседневной университетской деятельности, такой как посещение занятий, активное слушание преподавателя во время пар, подготовка и сдача работ [12]. Растет количество исследований и доказательств, подтверждающих утверждение о том, что качество и результат обучения студентов зависят от студенческой вовлеченности [13].

Также студенческая вовлеченность напрямую влияет на уровень отчислений студентов в университете. Недостаточная степень вовлеченности запускает негативные процессы, которые могут привести к отчислению студента. Чтобы избежать этого, необходимы изменения, которые поспособствуют вовлечению студентов.

Таким изменением может стать внедрение ИОТ, которая поможет студентам не терять мотивацию и заинтересованность в учебе, тем самым повышая качество обучения и понижая вероятность отчисления обучающихся.

### 3) Высокий уровень отсева студентов:

На высокий уровень отсева в российских вузах влияет не только студенческая вовлеченность. Это достаточно сложная проблема, которая вызывает беспокойство уже много лет. Согласно официальной статистике [14], уровень отсева в российских вузах составляет около 22%, а некоторые эксперты считают, что реальная цифра может быть еще выше, так как на эту тему проводилось достаточно мало исследований.

По результатам пилотного исследования [15] одной из характеристик отсева является несовпадение изначальных целей студентов и результатов их обучения в вузе. Жесткая и стандартизированная учебная программа в университетах часто не отвечает индивидуальным потребностям и планам студентов, а также может быть мало практически направленной, что приводит к потере желания продолжать обучение у студента.

Внедрение ИОТ поможет студентам в какой-то степени подстраивать под себя учебный план, тем самым достижение изначальных целей будет более реальным.

### **Обзор существующих аналогов по составлению УП и ИОТ**

Составление УП – важная задача для учебных заведений, поскольку она закладывает основу для результатов обучения студентов. С развитием технологий были разработаны различные ИС, помогающие в этом процессе, три наиболее популярные ИС: Информационная система «Планы» [16], Информационная система «Апекс-ВУЗ» [17], Информационная система «Галактика ERP 9.1. Управление учебным процессом» [18].

Описанные выше системы имеют как ряд преимуществ, так и ряд недостатков. Из недостатков можно выделить то, что данные системы предназначены лишь для упрощения формирования УП, так как в любом случае разработчик ОП самостоятельно распределяет дисциплины по периодам обучения, а программа лишь проверяет соответствие учебных часов и зачетных единиц на соответствие нормативным документам. Взаимосвязь дисциплин при этом не учитывается. Помимо этого, ни одна из приведенных систем не строит ИОТ.

### **Основание связи между дисциплинами**

Для автоматического построения УП или его части основная сложность заключается в определении корректного порядка изучения дисциплин. Порядок изучения дисциплин определяется наличием связи между ними: одни дисциплины опираются на знания, формируемые другими дисциплинами. Например, дисциплина «Алгоритмы и анализ сложности» требует знания «Способы оценки сложности алгоритмов» на базовом уровне. Это знание, в свою очередь, формируется дисциплиной «Теоретические основы информатики». Следовательно, дисциплина «Алгоритмы и анализ сложности» должна в УП стоять позже, чем «Теоретические основы информатики».

Для успешного построения учебного плана необходимо знать, какие компетенции формирует каждая дисциплина и какими компетенциями необходимо обладать для ее успешного изучения. Построение проекта УП и ИОТ будет опираться на знания о связях между компетенциями и дисциплинами.

Важным аспектом является формализация знаний. Это означает, что связи между дисциплинами, компетенциями и другими элементами УП должны быть четко определены и систематизированы. Формализация знаний позволяет создать структуру УП и ИОТ, которая будет логически последовательной и понятной для студентов и экспертов. Она также облегчает анализ и внесение изменений в УП при необходимости.

## **Выявление требований к онтологии**

В данной работе было решено использовать онтологическую модель. Онтология сможет обеспечить структурированное, многократно используемое и масштабируемое представление знаний. С помощью онтологии ИС сможет предоставлять пользователям точную и содержательную информацию, соответствующую их запросам, а также адаптироваться к изменяющимся потребностям в будущем.

Предметная область создаваемой онтологии – это описание содержания дисциплин и элективов, изучаемых студентами вуза. Онтология будет охватывать часть базовых и профессиональных дисциплин по направлению «Прикладная математика и информатика». В нее будут включены понятия, описывающие дисциплины, элективы, направления и знания, которые нужны для изучения дисциплин и которые в них изучаются.

Каждая вершина в онтологии будет иметь атрибут «тип», домен значений которого представлен ниже: знание, дисциплина, электив, направление.

Некоторые знания будут разбиты на уровни: базовый, средний и продвинутый. Уровни будут дописываться в скобках в конце названия понятия. Это сделано для того, чтобы понимать на каком уровне изучаются знания в той или иной дисциплине.

Элективы будут разбиты по направлениям, для того чтобы пользователям было проще в них ориентироваться. Каждый электив должен принадлежать хотя бы одному направлению.

Множество отношений состоит из элементов: требует\_знания, изучает, a\_part\_of, is\_a.

Дисциплина и знания, которые нужны для изучения данной дисциплины, будут соединены отношением «требует\_знания». Дисциплина и знания, которые изучает данная дисциплина, будут соединены отношением «изучает».

Базовый уровень какого-либо знания соединяется отношением «a\_part\_of» со средним уровнем этого же знания. Аналогично со средним и продвинутым уровнем. Если среднего уровня у знания не было, то базовый уровень сразу соединяется с продвинутым уровнем таким же отношением. Если понятие-целое разделяется на понятия-части, то они будут соединяться отношением «a\_part\_of». Элективы будут соединяться с направлениями, к которым они относятся, отношением «is\_a».

Разрабатываемая ИС будет позволять эксперту загружать свои онтологии для построения проекта УП и ИОТ. Загружаемая экспертом онтология должна соответствовать требованиям, перечисленным ниже:

- онтология может содержать только отношения «изучает», «требует\_знания», «is\_a», «a\_part\_of»;
- онтология не должна иметь циклов;
- онтология должна иметь хотя бы одно отношение "изучает";
- онтология должна иметь хотя бы две вершины;
- у вершины может не быть отношений, но при этом она не будет учитываться при составлении проекта УП и ИОТ;
- каждая вершина должна иметь уникальное имя;
- между двумя вершинами может быть только одно отношение.

## **Обоснование ручному созданию онтологии**

Далее будет приведено обоснование того, почему данная работа не посвящена автоматической обработке данных из текстов УМК. Было изучено шестьдесят УМК разных вузов по направлению «Прикладная математика и информатика», в которых были взяты знания, умения и владения по результатам обучения дисциплин. Рассматриваемые дисциплины, по

которым проводилось сравнение: математическая логика, дискретная математика, теоретические основы информатики, прикладная теория графов, алгоритмы и анализ сложности.

Исследование проводилось следующим образом: в каждом университете был проведен поиск каждой из рассматриваемых дисциплин. Поиск проводился на основе названия дисциплины, и если похожих названий не было найдено, то просматривались все компетенции дисциплин. Дисциплина с самыми похожими компетенциями считалась одной из рассматриваемых. Затем у найденных дисциплин сравнивались компетенции и выделялись общие. На основе данного исследования были сделаны следующие четыре вывода:

- 1) дисциплины в разных университетах не соотносятся по названию и содержанию;
- 2) одна и та же компетенция имеет разную формулировку, а также у разных вузов может классифицироваться как знание, как умение и как владение;
- 3) многие УМК недостаточно проработаны и содержат неполную информацию о компетенциях;
- 4) разный уровень подробности описания компетенций.

Таким образом, подводя итог вышесказанному, можно сделать четыре вывода, перечисленных ниже:

- 1) невозможно сделать универсальную систему по обработке УМК для всех вузов;
- 2) УМК требуют ручной обработки, так как автоматическая обработка неунифицированных УМК невозможна;
- 3) от классификации понятий по знаниям, умениям и владениям стоит избавиться, и воспринимать все понятия как знания;
- 4) требуется включить в систему эксперта, который добавит знания в систему и построит связи между ними.

Так же были выявлены требования к системе:

- 1) автоматизировать процесс составления ИОТ, на основе выбранных пользователем элективов,
- 2) поддерживать 2 типа пользователей: эксперт и студент,
- 3) позволять эксперту составлять и редактировать проект УП,
- 4) отображать перечень дисциплин в логически правильном порядке их изучения в проекте УП.

### **Алгоритм построения ИОТ**

В этом разделе будет рассмотрен алгоритм построения ИОТ. Будем рассматривать данный алгоритм на примере из рис. 1. Данный рисунок стоит интерпретировать следующим образом: элективы 2 и 3 требуют изучения электива 1, электив 4 требует изучения элективов 1 и 2. Электив 7 требует изучения элективов 1, 3, 5 и т.п.

Прежде чем строить ИОТ, пользователь должен выбрать элективы, которые он хочет увидеть в своей ИОТ, и количество элективов, которые войдут в ИОТ. Допустим, пользователь выбрал элективы 2,3,4,7.

Первым этапом является построение графа  $\text{Map}\langle \text{Integer}, \text{List}\langle \text{Integer}\rangle\rangle \text{ graph}$ , где в ключах будут храниться id выбранных пользователем элективов, а в их значениях будут храниться списки id их родителей. Причем родителями могут быть только вершины-элективы, и в списке они стоят в правильном порядке изучения. Затем мы преобразовываем это в  $\text{List}\langle \text{List}\langle \text{Integer}\rangle\rangle \text{ info\_for\_iots}$ , таким образом, что во внутреннем списке идут сначала все родители выбранного электива в правильном порядке, и в конец списка добавляется сам вы-

бранный электив. Т.е. для электива 2 этот список будет выглядеть так: [1,2], а для электива 7 вот так: [1,3,5,7].

Затем проверяется есть ли в `info_for_iots` списки, количество элективов в которых равно или превышает количество элективов в ИОТ (`countOfElectives`). Если такие есть, то добавляем их в список финальных ИОТ `List<List<Integer>> final_iots`. Все списки, которые были добавлены в `final_iots`, удаляются из `info_for_iots`. Допустим, что `countOfElectives` в текущем примере равно 4, тогда в `final_iots` будет добавлен список [1,3,5,7].

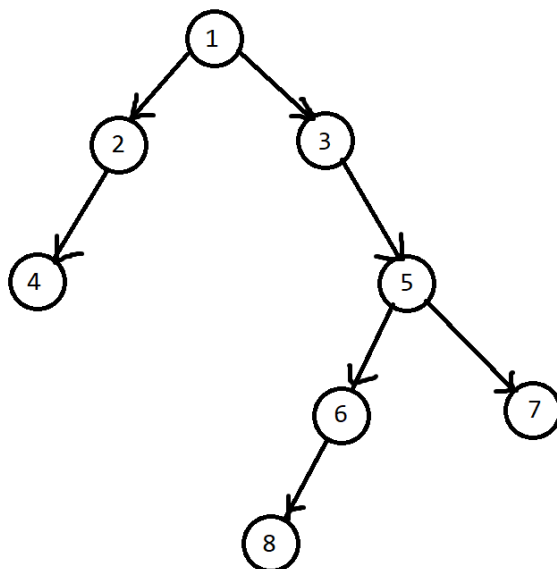


Рис. 1. Связи элективов

Далее запускается рекурсивная функция, которая формирует ИОТ из оставшийся списков. Проходимся по списку `info_for_iots` и получаем количество элективов, которых не хватает до формирования полноценной ИОТ. И далее сравниваем текущий список со всеми остальными, учитывая различные элементы, и если длина списка будет подходящей, т.е. при помощи комбинации двух списков будет формироваться полноценная ИОТ, то добавляем ее в финальные. Если же длина будет меньше, чем нужно, то в список `next_info_IOTs` добавляем соединенные два списка. Этот список будет хранить в себе новые комбинации ИОТ и передаваться при вызове рекурсии. Выходом из рекурсии будет следующее условие: если за проход по списку в `next_info_IOTs` ничего не было добавлено, то выходим из функции. В текущем примере будет образовано две ИОТ: [1,3,5,7], [1,2,3,4]. Таким образом, будут сформированы все ИОТ с теми элективами, которые выбрал пользователь.

### Алгоритм построения проекта УП

В этом разделе будет рассмотрен алгоритм построения проекта УП. Изначально формируется словарь `Map<Integer, List<Integer>> graph`, в котором ключами являются `id` всех дисциплин из онтологии и тех элективов, которые включает в себя выбранная экспертом ИОТ. В значениях этого словаря находятся `id` их родителей. После этого словарь используется для алгоритма Тарьяна [20], на выходе которого получается список отсортированных по порядку изучения дисциплин и элективов (`sortedNodes`).

Предположим, что у нас есть следующая онтология, изображенная на рис. 2, в которой вершины 7,8,9,10 – это выбранная пользователем ИОТ, которая должна быть включена в

проект УП. Алгоритм Тарьяна может начать со случайной вершины, например, с вершины 3. Тогда после обработки этой вершины, результирующий список будет выглядеть следующим образом: [1,2,3]. Все родители вершины 3 найдены, но еще не все вершины из примера посещены, поэтому снова начинаем алгоритм со случайно вершины. Допустим, это вершина 7, тогда результирующий список: [1,2,3,7]. После обработки всех вершин получим sortedNodes = [1,2,3,7,4,5,6,8,9,10].

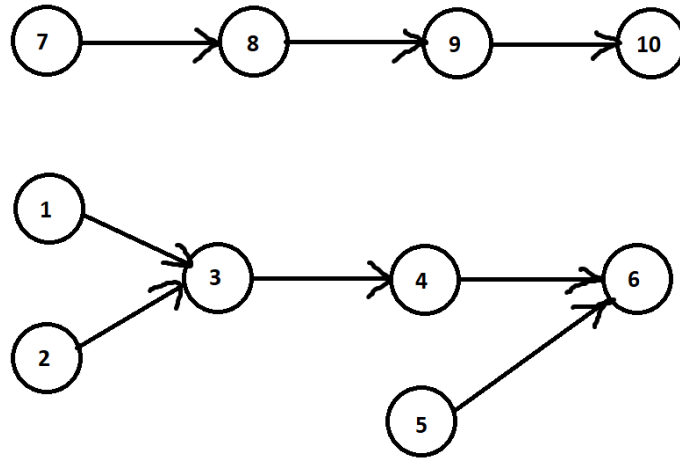


Рис. 2. Пример связи дисциплин и элективов

Эксперт выбирает максимальное количество предметов, которое может быть в триместре (countDisciplines). Триместров всего может быть 12. Если номер триместра меньше или равен 12, то будем его называть существующим триместром.

Изначально текущий триместр (curTrimestr) равен 1. Далее пока список sortedNodes не пустой и пока curTrimestr меньше или равен 12, то для каждой вершины из sortedNodes выполняем следующий алгоритм:

- 1) Проверяем есть ли родители вершины в текущем триместре. Если есть, то переходим к пункту 2, иначе к пункту 3.

- 2) Ищем максимальный триместр (maxTrim), который содержит хотя бы одного из родителей вершины. И если в следующем по счету триместре есть место (т.е. количество предметов в триместре меньше countDisciplines), то добавляем предмет туда, иначе добавляем предмет в ближайший после maxTrim существующий триместр, в котором есть свободное место. Если существующий триместр со свободным местом не найден, то удаляем предмет из sortedNodes.

- 3) Если в текущем триместре есть место, то добавляем в него предмет, иначе находим первый существующий триместр, в котором есть свободное место и заносим этот результат в curTrimestr. Добавляем предмет в найденный триместр. Если существующий триместр со свободным местом не найден, то удаляем предмет из sortedNodes.

- 4) Если вершина была добавлена в проект УП, удаляем ее из списка sortedNodes.

- 5) Если в текущем триместре закончилось место, то ищем ближайший триместр со свободным местом.

Допустим, эксперт выбрал, что в одном триместре может быть максимум 3 предмета. Вершины 1 и 2 добавятся в 1-ый триместр, т.к. они друг с другом не связаны. Далее по списку идет вершина 3, она связана с вершинами в текущем триместре, поэтому, ищем максимальный триместр, в котором есть ее родители (maxTrim = 1), и добавляем вершину в триместр, следующий по счету, т.к. там есть свободные места.



Вершину 7 добавляем в 1-ый триместр, т.к. в нем нет ее родителей и есть свободное место. Первый триместр полностью заполнен, поэтому  $curTrimestr$  становится равен 2. Во втором триместре у вершины 4 есть родитель, находим  $maxTrim = 2$ , следовательно, вершина добавится в триместр номер 3. Вершину 5 беспрепятственно добавляем во второй триместр. Вершину 6 добавить в текущий триместр ( $curTrimestr = 2$ ) нельзя, т.к. в нем есть ее родители, находим  $maxTrim = 3$ , следовательно, вершина добавится в триместр номер 4. Далее аналогичным образом распределяются все оставшиеся вершины. Конечный результат представлен на рис. 3.

Таким образом будет сформирован проект УП, в котором предметы стоят в правильном порядке изучения и разбиты по триместрам.

Стоит обратить внимание на то, что в данном примере элективы, принадлежащие ИОТ (вершины 7,8,9,10), не связаны с остальными дисциплинами (вершины 1-6), поэтому в проекте УП элективы могут стоять с первого триместра. Но если бы между элективами и дисциплинами была связь, то при построении проекта УП она была бы учтена. Например, если бы вершина 1 была родителем для вершины 7, то вершина 7 переместилась бы на второй триместр, 8-ая вершина на третий, 9-ая – на четвертый и десятая – на пятый.

Триместр 1	Триместр 2	Триместр 3	Триместр 4
1	3	4	6
2	5	9	10
7	8		

**Рис. 3. Результат алгоритма построения проекта УП**

### **Заключение**

В результате данной работы были изучены важные понятия по теме исследования, такие как: образовательная программа, учебный план, ФГОС, индивидуальная образовательная траектория (ИОТ). Была обоснована актуальность ИОТ и подробно рассмотрены 3 проблемы, которые она решает полностью или частично: устаревшая учебная программа, низкая вовлеченность студентов, высокий уровень отсева студентов.

Для представления знаний о взаимосвязях между дисциплинами была выбрана онтологическая модель. Было изучено понятие онтологии, ее преимущества и выявлены требования к онтологии. Было проанализировано 60 УМК из 12 различных вузов.

Были описаны алгоритмы построения ИОТ и проекта УП. Была разработана ИС и протестирована на онтологии, которая содержит в себе 303 понятия, а именно 14 дисциплин, 9 элективов и 280 знаний. Онтология была составлена совместно с семью экспертами. Были построены различные варианты ИОТ и проектов УП. Полученные результаты отображают последовательность изучения дисциплин в логически правильном порядке и являются адекватными.

### **Библиографический список**

1. "Федеральный закон "Об образовании в Российской Федерации"" от 29.12.2012 № 273-ФЗ Ст. 12 с изм. и допол. в ред. от 17.02.2023.

2. "Федеральный закон "Об образовании в Российской Федерации"" от 29.12.2012 № 273-ФЗ Ст. 2 с изм. и допол. в ред. от 17.02.2023.
3. ФГОС URL: <https://fgos.ru/> (дата обращения: 27.02.23).
4. Приказ Министерства образования и науки РФ "Федеральный государственный образовательный стандарт высшего образования – бакалавриат по направлению подготовки 01.03.02 Прикладная математика и информатика" от 10.01.18 № 9 с изм. и допол. в ред. от 16.06.2021.
5. *Суртаева Н.Н.* Педагогические технологии естественного обучения // Химия в школе. – 1998. – №7. – С. 13–17.
6. *Бухтиярова И.Н.* Метод проектов и индивидуальные программы в продуктивном обучении // Школьные технологии. – 2001. – № 2. – С. 108–114.
7. *Тимошина Т.А.* Концепция выстраивания индивидуальной образовательной траектории студента // Педагогика и психология как ресурс развития современного общества: сб. ст. 2-й Междунар. науч.-практ. конф. (Рязань, 7–9 окт. 2010 г.). – Рязань, 2010. – С. 315–320.
8. *Хуторской А.В.* Дидактическая эвристика: теория и технология креатив. обучения. – М.: Изд-во Моск. гос. ун-та, 2003. – 415 с.: ил., портр. – Библиогр.: с. 400–415.
9. *Арифулина Р.У., Катушенко О.А.* АНАЛИЗ ОТЕЧЕСТВЕННЫХ И ЗАРУБЕЖНЫХ ТРЕНДОВ ИНДИВИДУАЛИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА В ВУЗЕ: АНАЛИТИЧЕСКИЙ ДОКЛАД // Вестник Мининского университета. – 2021. – Том 9, №4
10. *Воронова Н.А.* Конкурентоспособность – показатель качества образования / Н.А. Воронова // Гуманитаризация инженерного образования: методологические основы и практика: материалы международной научно-практической конференции. – Тюмень, 2020. – С. 319-321.
11. Changellenge >>: Студентов топовых вузов не устраивают устаревшие учебные программы // HSE Career Центр развития карьеры ВШЭ [Электронный ресурс]. – URL: <https://career.hse.ru/news/284815461.html> (дата обращения: 27.02.23).
12. *Chapman E.* Assessing Student Engagement Rates // ERIC Clearinghouse on Assessment and Evaluation. 2003 [Электронный ресурс]. – URL: <https://files.eric.ed.gov/fulltext/ED482269.pdf> (дата обращения: 24.02.23)
13. *Coates H.* The Value of Student Engagement for Higher Education Quality Assurance // Quality in Higher Education. 2005. No. 11.
14. *Горбунова Е.В.* Выбытия студентов из вузов: исследования в России и США // Вопросы образования. – 2018. – №1
15. *Груздев И.А., Горбунова Е.В., Фрумин И.Д.* Студенческий отсев в российских вузах: к постановке проблемы // Вопросы образования. – 2013. – №2
16. ПРОГРАММНЫЙ КОМПЛЕКС "ПЛАНЫ" // ММИС Лаборатория [Электронный ресурс]. – URL: <https://www.mmis.ru/programs/plany> (дата обращения: 27.02.23).
17. Система автоматизации образовательного процесса «Апекс-ВУЗ» // Апекс-ВУЗ [Электронный ресурс]. – URL: <https://apeks-vuz.ru/> (дата обращения: 27.02.23).
18. Галактика ERP 9.1. Управление учебным процессом // Корпорация Галактика [Электронный ресурс]. – URL: <https://galaktika-it.ru/spb/erp-uup> (дата обращения: 27.02.23).
19. *Gruber T.R.* A translation approach to portable ontologies. Knowledge Acquisition. – 1993. – 5(2):199-220.
20. *Роберт Седжвик.* Алгоритмы на графах = Graph algorithms. – 3-е изд. – Россия, Санкт-Петербург: «ДиаСофтЮП», 2002. – С. 496.

## References

1. The federal law of the Russian Federation of December 29, 2012 №273-FZ «About education in the Russian Federation». Art. 12, as amended on 17.02.2023 (in Russian)
2. The federal law of the Russian Federation of December 29, 2012 №273-FZ «About education in the Russian Federation». Art. 2, as amended on 17.02.2023 (in Russian)
3. Federal State Educational Standard. Available at: <https://fgos.ru/> (accessed on 27.02.23) (in Russian)
4. Order of the Ministry of Education and Science of the Russian Federation "Federal State Educational Standard for Higher Education – Bachelor's Degree in the field of preparation 01.03.02 Applied Mathematics and Informatics" dated 10.01.18 No. 9, as amended on 16.06.2021 (in Russian)
5. *Surtayeva, N. N.* Pedagogical Technologies of Natural Learning *Himiya v shkole* [Chemistry in School], 1998, no. 7, pp. 13-17 (in Russian)
6. *Bukhtiyarova, I. N.* The Project Method and Individual Programs in Productive Learning *SHkol'nye tekhnologii* [School Technologies], 2001, no. 2, pp. 108–114 (in Russian)
7. *Timoshina, T. A.* The Concept of Building an Individual Educational Trajectory for a Student. *Pedagogika i psihologiya kak resurs razvitiya sovremennogo obshchestva*. [Pedagogy and Psychology as a Resource for the Development of Modern Society Collected Articles of the 2nd International Scientific and Practical Conference]. Ryazan, 2010, pp. 315–320 (in Russian)
8. *Khutorskoy, A. V.* Didakticheskaya evristika: teoriya i tekhnologiya kreativ.obucheniya [Didactic Heuristics: Theory and Technology of Creative Teaching]. Moscow, Moscow State University Publishing House, 2003, 415 p. (in Russian)
9. *Arifulina, R. U., Katushenko, O. A.* Analysis of Domestic and Foreign Trends in Individualization of the Educational Process in Higher Education: Analytical Report *Vestnik Mininskogo universiteta* [Bulletin of Mininn University], 2021, vol. 9, no. 4 (in Russian)
10. *Voronova, N. A.* Competitiveness as an Indicator of the Quality of Education. *Gumanitarizatsiya inzhenerenogo obrazovaniya: metodologicheskie osnovy i praktika* [Humanitarization of Engineering Education: Methodological Foundations and Practice: Proceedings of the International Scientific and Practical Conference], Tyumen, 2020, pp. 319-321 (in Russian)
11. Challenge >>: Students of Top Universities Are Dissatisfied with Outdated Curricula *HSE Career Centr razvitiya kar'ery VSHE* [HSE Career Career Development Center]. [Electronic resource]. – Available at: <https://career.hse.ru/news/284815461.html> (accessed on 27.02.23) (in Russian)
12. *Chapman E.* Assessing Student Engagement Rates. ERIC Clearinghouse on Assessment and Evaluation. 2003. [Electronic resource]. – Available at: <https://files.eric.ed.gov/fulltext/ED482269.pdf> (accessed on 24.02.23)
13. Coates H. The Value of Student Engagement for Higher Education Quality Assurance. *Quality in Higher Education*, 2005, no. 11.
14. *Gorbunova, E. V.* Student Attrition in Universities: Research in Russia and the USA. *Voprosy Obrazovaniya* [Educational issues], 2018, no. 1 (in Russian)
15. *Gruzdev, I. A., Gorbunova, E. V., Frumin, I. D.* Student Attrition in Russian Universities: Problem Statement *Voprosy Obrazovaniya* [Educational issues], 2013, no. 2 (in Russian)
16. Program Complex 'Plans' MMIS Laboratoriya [MMIS Laboratory]. [Electronic resource]. – Available at: <https://www.mmis.ru/programs/plany> (accessed on 27.02.23) (in Russian)
17. Automated System for the Educational Process 'Apex-VUZ' *Apeks-VUZ* [Apex-VUZ] [Electronic resource]. – Available at: <https://apeks-vuz.ru/> (accessed on 27.02.23) (in Russian)

18. Galaxy ERP 9.1. Educational Process Management Korporaciya Galaktika [Galaxy Corporation] [Electronic resource]. – Available at: <https://galaktika-it.ru/spb/erp-uup> (accessed on 27.02.23) (in Russian)
19. *Gruber, T. R.* A Translation Approach to Portable Ontologies. Knowledge Acquisition, 1993, pp. 199-220.
20. *Sedgewick, R.* Algoritmy na grafah [Algorithms on Graphs], 3rd ed, Russia, St. Petersburg, DiaSoftUP, 2002, 496 p.

## **DEVELOPMENT OF THE SYSTEM OF ANALYSIS AND DESIGN OF EDUCATIONAL PROGRAMS OF HIGHER EDUCATION BASED ON THE ONTOLOGICAL REPRESENTATION OF COMPETENCES AND DISCIPLINES**

*Cheremnykh Alexandra N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [rain.raicker@yandex.ru](mailto:rain.raicker@yandex.ru)

This article deals with automation of individual educational trajectory (IET) based on student's preferences and interests, as well as with automation of curriculum design. The paper contains a description of basic concepts on the topic of research, justification of relevance of IoT, a detailed review of the problems it solves fully or partially: outdated curriculum, low student engagement, high dropout rate of students. A brief theory on ontologies, identifying ontology requirements, justification for manual completion of the ontology is reviewed. The algorithms of constructing IoT and draft UI are described.

Keywords: individual educational trajectory, curriculum, ontology, information system.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ АНАЛИЗА ИНВЕСТИЦИОННЫХ ПРОЕКТОВ

*Шубинцева Яна Евгеньевна, Василюк Надежда Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, shubintseva2014@yandex.ru

Рассмотрены основные инструменты, используемые для проектирования и документирования информационной системы анализа инвестиционных проектов. Определены предмет и объект исследования, а также обоснована актуальность разработки данной информационной системы. Поставлены задачи, которые необходимо решить в процессе проектирования. Был проведен анализ уже существующих информационных систем, выполняющих аналогичные функции, что позволило выявить требования к разрабатываемой системе. В результате анализа была обоснована целесообразность проектируемой системы и выбора метода анализа инвестиционных проектов. В ходе исследования был выбран программный инструмент проектирования информационной системы StarUML [1]. Приложение Balsamiq выбрано для разработки прототипа интерфейса приложения [2]. В результате проектирования информационной системы было описано взаимодействие с ней со стороны инвестора и администратора. Также был спроектирован интерфейс информационной системы на основе созданных диаграмм и технического задания.

Ключевые слова: информационная система, анализ инвестиционных проектов, UML-моделирование.

Инвестиции имеют важную роль в развитии бизнеса, а значит многие компании рассматривают на вложение своих активов в перспективные проекты. Компании всегда стремятся увеличить доходность и оптимизировать процессы управления. Разнообразие и вариативность инвестиционных проектов, а, следовательно, потребность принятия решений в непредсказуемых условиях, требуют использования эффективных инструментов анализа и оценки.

Система анализа может позволить провести автоматизацию процессов, ускорить принятие решений, увеличить точность оценки перспективности и спрогнозировать дальнейшие действия. Это особенно необходимо для поддержания конкурентоспособности компании и быстрого изменения направления ее решений в экономической среде.

Объектом исследования будем считать процесс реализации системы анализа инвестиционных проектов.

Предметом исследования является система анализа инвестиционных проектов.

Целью работы является проектирование и документирование системы анализа инвестиционных проектов.

Выделены следующие задачи исследования:

- Анализ и описание существующих аналогов приложений на тему анализа инвестиций;
- Изучение основных методов анализа инвестиционных проектов и определение требований к системе анализа;
- Выбор программных и аппаратных средств для проектирования UML моделей и интерфейса;

- Проектирование моделей логики приложения и базы данных;
- Создание прототипа интерфейса системы;
- Создание ТЗ, описывающее основные требования к системе анализа инвестиционных проектов.

Хотя на рынке существует множество аналоговых систем для анализа инвестиционных проектов, разработка собственной системы может быть целесообразной по нескольким причинам:

1. Оригинальность идеи. Разработка собственной системы позволяет развивать уникальные идеи и подходы к анализу инвестиционных проектов, что может повысить конкурентоспособность на рынке.

2. Настраиваемость. Разработка системы собственными силами позволяет адаптировать ее с учетом различных требований и особенностей.

3. Управление данными. При использовании сторонних систем компании не всегда имеют полный контроль над данными и анализом, что может быть проблематично при работе с секретной информацией. В таких случаях могут быть разработаны собственные системы, обеспечивающие полный контроль доступа к данным и соблюдение конфиденциальности.

4. Необходимость интеграции с другими системами. Компании могут иметь существующие системы, которые используются для выполнения анализа проектов. В этом случае разработка собственной системы может позволить интегрировать новую систему с существующей, тем самым повысив ее эффективность.

Логика приложения представлена при помощи диаграммы прецедентов (0).

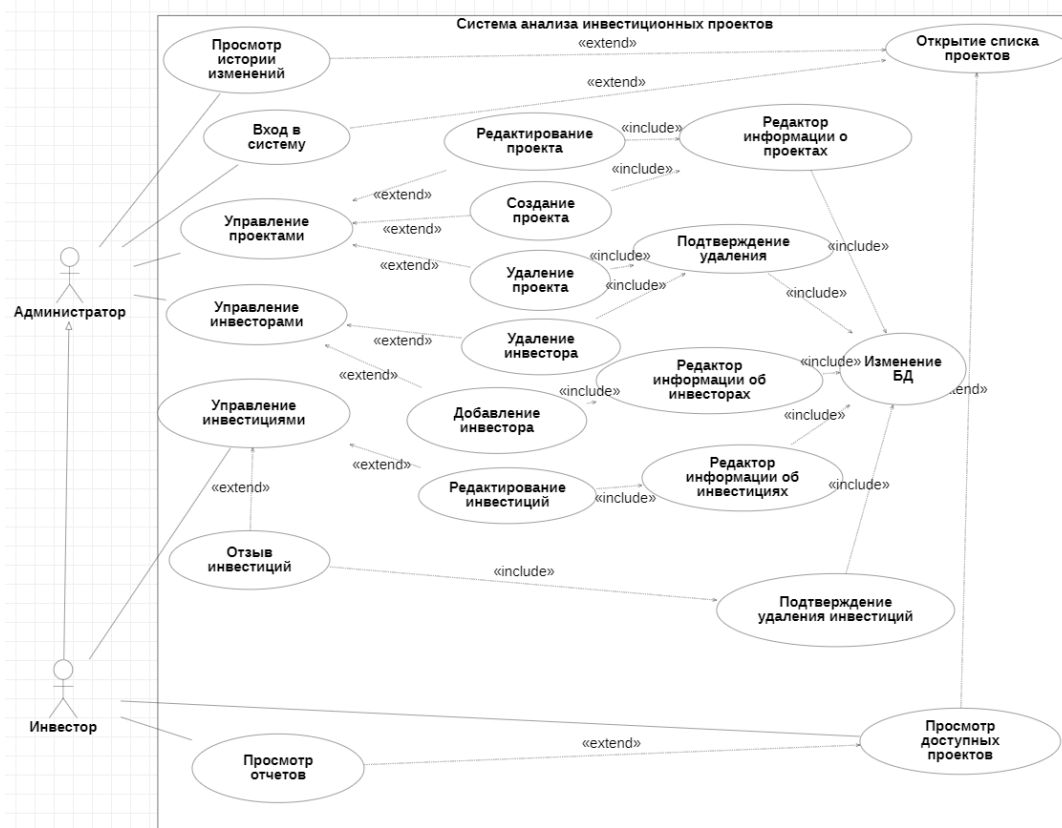


Рис. 1. Диаграмма прецедентов

Проектирование логики системы анализа инвестиционных проектов должно учитывать различия между инвестором и администратором.

В проектируемой системе инвестор является пользователем, который может:

- Просматривать доступные проекты
- Управлять инвестициями
- Просматривать отчеты

Администратор же имеет ряд других прав в проектируемой информационной системе:

- Просмотр истории изменений;
- Управление проектами;
- Управление инвесторами.

Следовательно, администратор и инвестор имеют разные права на использование системы, но могут выполнять и некоторые общие действия. Например, изменение БД при редактировании каких-либо данных.

Для отображения взаимодействия в системе, необходимо создать диаграмму классов (0).

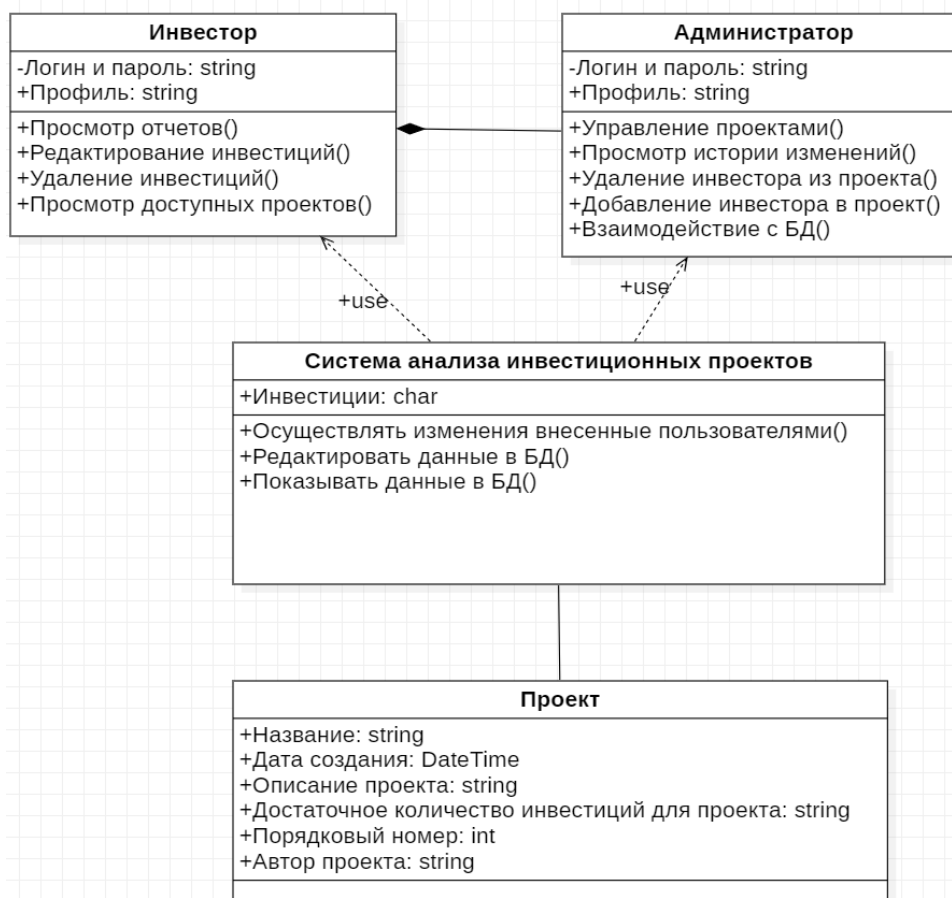


Рис. 2. Диаграмма классов

Она помогает определить классы и их связи, которые необходимы для реализации функциональности системы. Также позволяет представить структуру иерархии классов, что упрощает проектирование системы.

Система спроектирована с помощью диаграмм UML – это процесс создания графического описания системы, которое может быть использовано для понимания ее структуры, функций и поведения [3]. UML является стандартом, позволяющий разработчикам использовать единый язык для описания структуры и поведения системы.

Интерфейс имеет важную роль в проектировании системы, так как является основным средством взаимодействия пользователей с системой. Он обеспечивает удобство и эффективность использования системы, а также влияет на пользовательский опыт.

Понятный и интуитивно понятный интерфейс позволяет пользователям легко освоить систему и быстро выполнять необходимые операции. Это особенно важно в контексте анализа инвестиционных проектов, где пользователи могут быть неспециалистами в области информационных технологий.

Кроме того, хорошо визуально понятный интерфейс позволяет пользователю легко взаимодействовать с различными функциями и возможностями системы, предоставляя информацию в удобной форме. Это помогает улучшить качество принимаемых решений и повысить эффективность анализа инвестиционных проектов.

При запуске приложения открывается окно входа в систему с полями логина и пароля (0).

Рис. 3. Вход в систему

Окно входа в систему с полями логина, пароля и выбором пользователя является первым шагом в использовании системы. К тому же, это обеспечивает безопасный и индивидуальный доступ к соответствующим функциональным возможностям в зависимости от роли пользователя.

Рассмотрим вход в систему под пользователем Инвестор. При входе в систему открывается окно с основными действиями в системе для инвестора (0).

Рис. 4. Инвестор



Инвестор имеет право выбрать нужный проект, просмотреть отчеты и управлять своими инвестициями.

Все эти функциональности обеспечивают инвестору полный контроль и гибкость в управлении его инвестиционным портфелем. Инвестор может принимать основанные на фактах решения, анализируя информацию о проектах и эффективно управляя своими инвестициями через систему анализа инвестиционных проектов.

При нажатии на кнопку удаления проекта, пользователю открывается окно предупреждения (0).

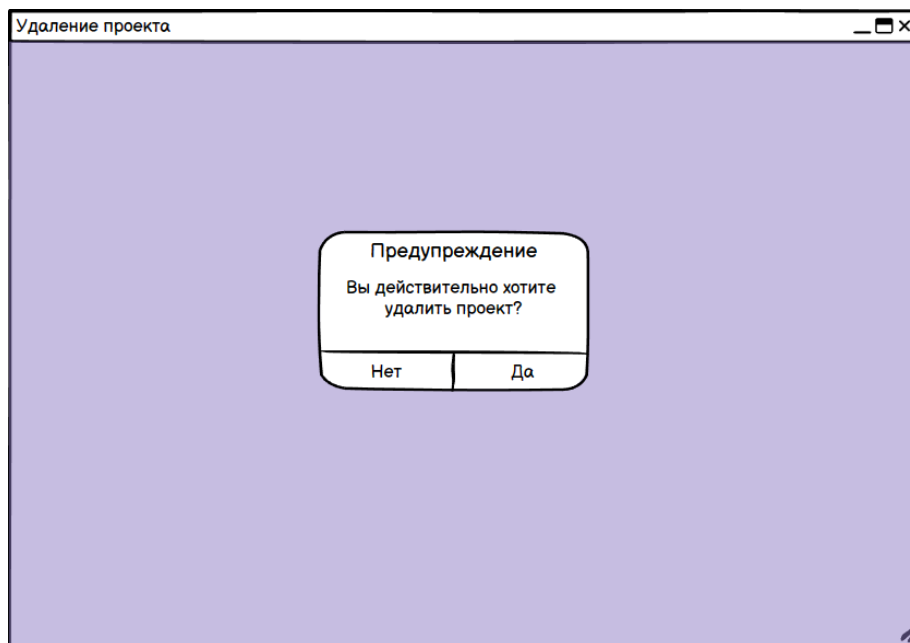


Рис. 5. Удаление проекта

Такой подход позволяет избежать случайного удаления проекта и защищает пользователя от возможной потери ценной информации. Он дает пользователю возможность дважды подумать перед принятием окончательного решения об удалении проекта и обеспечивает дополнительный уровень безопасности в процессе работы с системой.

### Библиографический список

1. Introduction – StarUML documentation// StarUML [Электронный ресурс]. – URL: <https://docs.staruml.io> (дата обращения: 27.04.2023).
2. Balsamiq Documentation // Balsamiq [Электронный ресурс]. – URL: <https://balsamiq.com/docs> (дата обращения: 10.05.2023).
3. Новиков Ф.А. Анализ и проектирование на UML: учебное пособие / Новиков Ф.А. – СПб: Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2007. – 286 с.

### References

1. Introduction – StarUML documentation // StarUML [Electronic resource]. – URL: <https://docs.staruml.io> (accessed: 27.04.2023).
2. Balsamiq Documentation // Balsamiq [Electronic resource]. – URL: <https://balsamiq.com/docs> (accessed: 10.05.2023)
3. Novikov F.A. Analiz i proektirovanie na UML: uchebnoe posobie / Novikov F.A. – SPb: Sankt-Peterburgskii gosudarstvennyi universitet informatsionnykh tekhnologii, mekhaniki i optiki, 2007, 286 pp.

## **DESIGN AND DOCUMENTATION OF THE INVESTMENT PROJECT ANALYSIS SYSTEM**

*Shubintseva Yana E., Vasiluk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, shubintseva2014@yandex.ru

The main tools used for designing and documenting an information system for analyzing investment projects are considered. The subject and object of research are determined, and the relevance of the development of this information system is justified. The tasks that need to be solved in the design process are set. The analysis of already existing information systems performing similar functions was carried out, which made it possible to identify the requirements for the system being developed. As a result of the analysis, the expediency of the designed system and the choice of the method of analysis of investment projects was justified. During the research, a software tool for designing the StarUML information system was selected. The Balsamiq application was chosen to develop a prototype of the application interface. As a result of the design of the information system, the interaction with it on the part of the investor and the administrator was described. The interface of the information system was also designed based on the created diagrams and technical specifications.

**Keywords:** information system, analysis of investment projects, UML modeling.

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ТЕСТИРОВАНИЯ ЗНАНИЙ ПО ДИСЦИПЛИНЕ «ЯЗЫКИ ПРОГРАММИРОВАНИЯ»**

*Япаров Артем Эдуардович, Ракина Валерия Денисовна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, Yaparov\_2002@mail.ru

Проектирование и документирование информационной системы для тестирования знаний по дисциплине «Языки программирования». В статье проанализирована предметная область и выявлены особенности информационной системы, выбраны средства для разработки и моделирования, разработаны модели и прототип информационной системы.

Ключевые слова: проектирование и документирование информационной системы, языки программирования, тестирование знаний

### **Введение**

Основные проблемы при изучении нового языка программирования – изучение и усвоение нового теоретического материала, конечно же, все необходимые знания студенты получают на лекциях, и преподаватели даже зачастую выкладывают презентации с лекций в общий доступ для студентов, но как правило, презентации не так насыщены информацией, как сами лекции, ведь преподаватель ведет рассказ по презентации, раскрывая и давая больше информации, чем есть на самом деле в презентации. Сложно на протяжении всей лекции записывать и запоминать все то, о чем говорит преподаватель, все равно что-нибудь забудется или будет пропущено, поэтому хотелось бы создать возможность для студентов полный текст лекции, чтобы освежить память или даже изучить что-нибудь новое. Также при изучении нового языка программирования проблемой может стать отсутствие дополнительной практики, с помощью которой можно закрепить знания и отработать приобретенные навыки. Поэтому информационная система, которая объединит два этих аспекта будет актуальным решением данной проблемы, что в свою очередь сильно упростит процесс обучения.

Цель работы заключается в проектировании и документировании информационной системы для тестирования знаний по дисциплине «Языки программирования».

### **Анализ предметной области**

В качестве аналогов разрабатываемой системы были выбраны следующие веб-сайты:

1. METANIT.COM – Сайт о программировании [5];
2. Prolog Site [6];
3. Code Abbey [7];
4. Programming Skills [8].

Для наглядности, преимущества и недостатки рассмотренных систем представлены в табл. 1.

Таблица 1. Преимущества и недостатки рассмотренных систем

Название информационной системы	METANIT.COM	Prolog Site	Code Abbey	Programming Skills
Теоретический материал по F#, Prolog и C#	Есть по языкам программирования F# и C#	Есть по языку программирования Prolog	Отсутствует	Отсутствует
Практические задания по F#, Prolog и C#	Отсутствуют	Есть по языку программирования Prolog	Есть задания общего характера, можно прикреплять решение на различных язык программирования	Отсутствуют
Тестирование знаний по F#, Prolog и C#	Отсутствует	Отсутствует	Отсутствует	Есть по языку программирования C#

Несмотря на то, что перечисленные системы хороши и удобны в использовании, нужно разработать более узконаправленную информационную систему для тестирования знаний по дисциплине «Языки программирования». Основываясь на плюсах и минусах каждой из проанализированных систем, можно выделить следующие разделы, которые целесообразно добавить в разрабатываемую систему:

- раздел с теоретическим материалом, где будут подразделы по каждому из языков программирования C#, F#, Prolog соответственно;
- раздел с практическими заданиями, которые будут представлены в соответствии с теоретическим материалом;
- раздел с онлайн-тестами для проверки самого себя по теоретическому материалу;

Процесс добавления и регистрации студентов можно будет проводить тремя способами: с помощью отправки студенту ссылки на информационную систему, где он самостоятельно регистрируется, но у него отсутствуют какие-либо возможности в системе, после чего преподаватель может дать ему разрешение или удалить; добавление студента вручную, информацию о студенте в этом случае заполняет преподаватель; добавление студентов через файл, где информация о них записана по определенному шаблону.

### Выбор средств для проектирования и разработки

Diagrams.net [9] – это бесплатное онлайн кроссплатформенное программное обеспечение для рисования графиков. Позволяет строить UML диаграммы, поддерживает построение различных типов диаграмм, принятых нотацией UML 2.0, обладает широкими функциональными возможностями. С наиболее простым, интуитивно понятным интерфейсом и с возможностью сохранять результаты работы в облаке или браузере, поэтому было использовано, именно, это программное обеспечение.

Также для работы любой информационной системы необходима база данных (БД). В соответствии с этим необходимо выбрать инструмент для проектирования БД. Выбор был сделан в пользу MySQL Workbench [10], т.к. данная программа является бесплатной и есть возможность скачивания без регистрации (из Российской Федерации).

Реализуемая информационная система будет представлять из себя клиент-серверную модель web-приложения. Поэтому были выбраны средства для реализации серверной и клиентской части приложения, а также СУБД (систему управления базами данных).

Для реализации серверной части выбран язык программирования C# – высокоуровневый язык программирования. .Net – фреймворк с открытым исходным кодом ASP.NET Core от Microsoft используется для создания веб-сайтов с помощью C# и не только. ASP.NET Core обладает широким набором различных функций и довольно-таки прост для разработки, также имеет очень высокую производительность. В качестве среды разработки выбрана Visual Studio [11], так как является одной из лучших сред разработки, а также является бесплатной и наиболее функциональной.

Для клиентской части приложения выбраны наиболее простые и популярные средства разработки такие, как: HTML5 (Hyper Text Markup Language) [12], CSS (Cascading Style Sheets) [13].

Для управления информационной системой тестирования знаний больше подойдет простая и надежная СУБД MySQL [14], которая сможет обеспечить хорошую скорость работы и безопасность. Поэтому выбор СУБД был сделан в пользу MySQL. Также создав модель базы данных в MySQL Workbench ее можно будет перенести в MySQL и тогда у нас будет уже готовая база данных, которую нужно будет только заполнить.

Для создания прототипа будущего интерфейса воспользуемся специальным инструментом для web-дизайна – Figma [15]. В Figma есть возможность бесплатно работать сразу с 3-мя проектами, огромная библиотека шаблонов и широкий спектр функциональных возможностей.

### **Проектирование информационной системы**

Для каждого вида пользователей определим возможности, которые будут доступны в информационной системе, и отобразим их на диаграмме прецедентов.

Всего в системе будет три различных вида пользователей:

- Преподаватель;
- Студент;
- Администратор.

Преподаватель будет иметь возможность редактировать и добавлять теоретический материал и практические задания, добавлять тесты и проводить контрольные мероприятия, выставлять баллы за контрольные работы, добавлять и удалять студентов из курса, создавать и удалять группы, а также общаться с другими преподавателями и студентами. Диаграмма прецедентов для преподавателя на рис. 1.

В возможности студента будет входить просмотр теоретического материала и практических заданий, прохождение тестов по темам и просмотр к ним ответов после прохождения, участие в контрольных мероприятиях, общение с преподавателем. Диаграмма прецедентов для студента на рис. 2.

Администратор будет иметь право добавлять и удалять из системы преподавателей и студентов, создавать и удалять группы, редактировать и добавлять теоретический материал, практические задания и тесты, смотреть историю собственных действий и историю различных изменений в системе (например, добавление новой темы в теоретический материал). Диаграмма прецедентов для администратора на рис. 3.

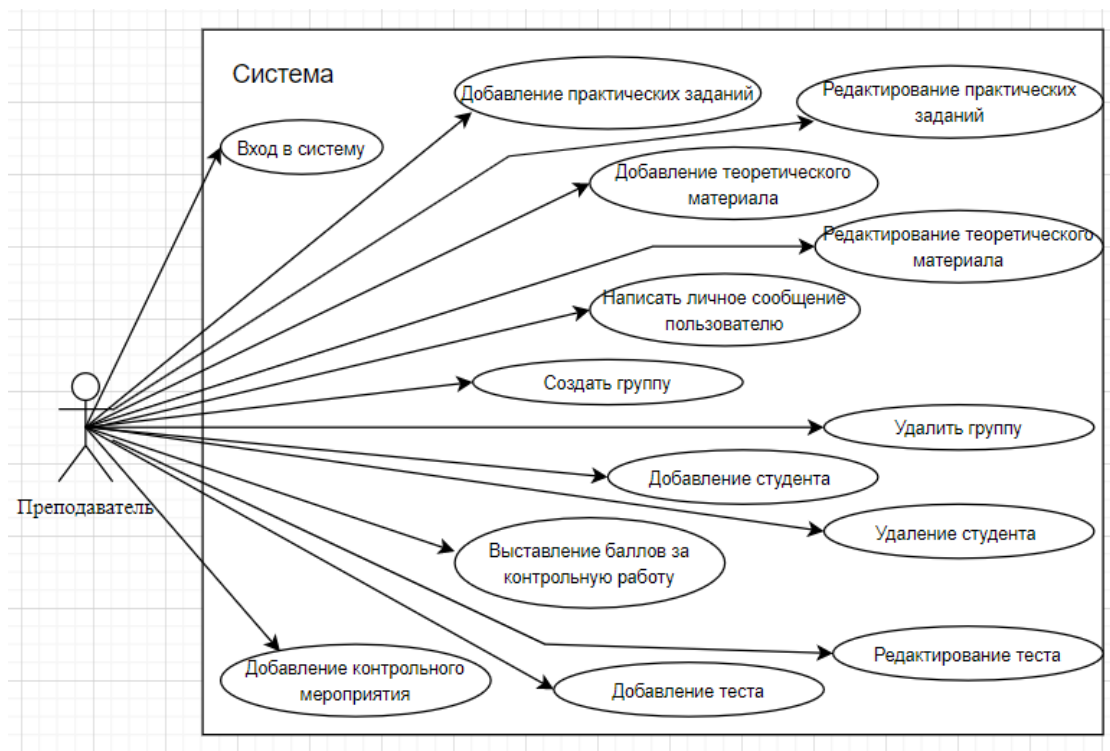


Рис. 1. Диаграмма прецедентов для преподавателя

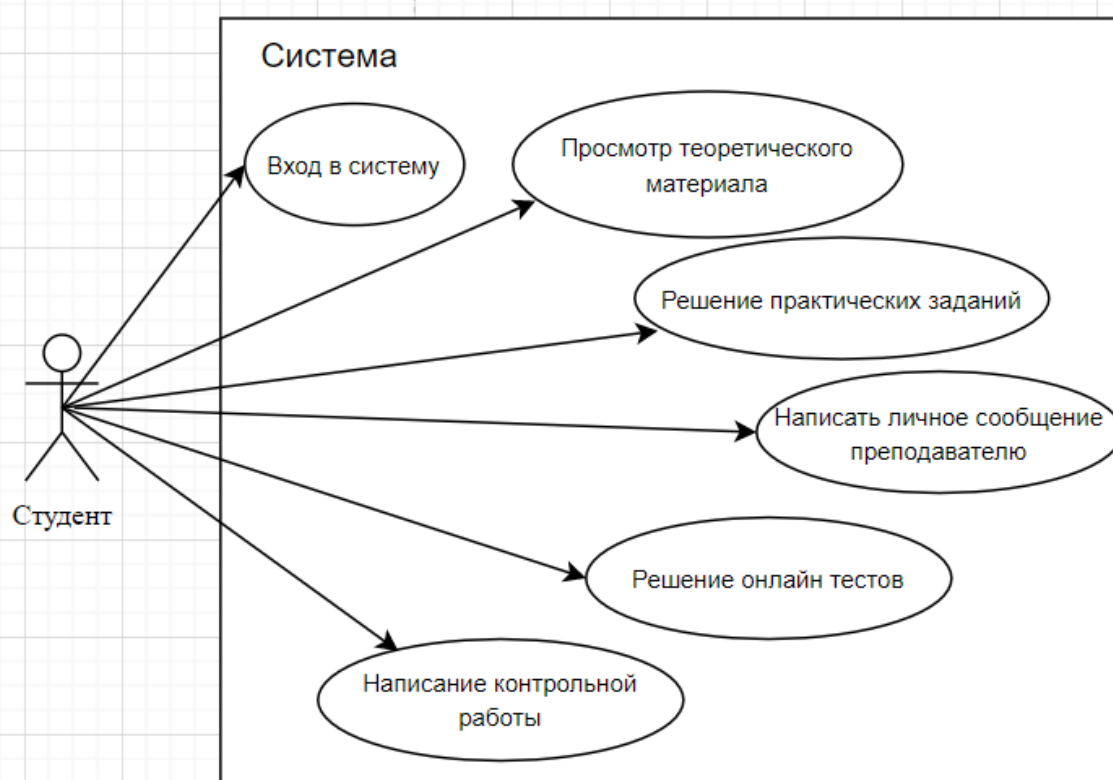


Рис. 2. Диаграмма прецедентов для студента

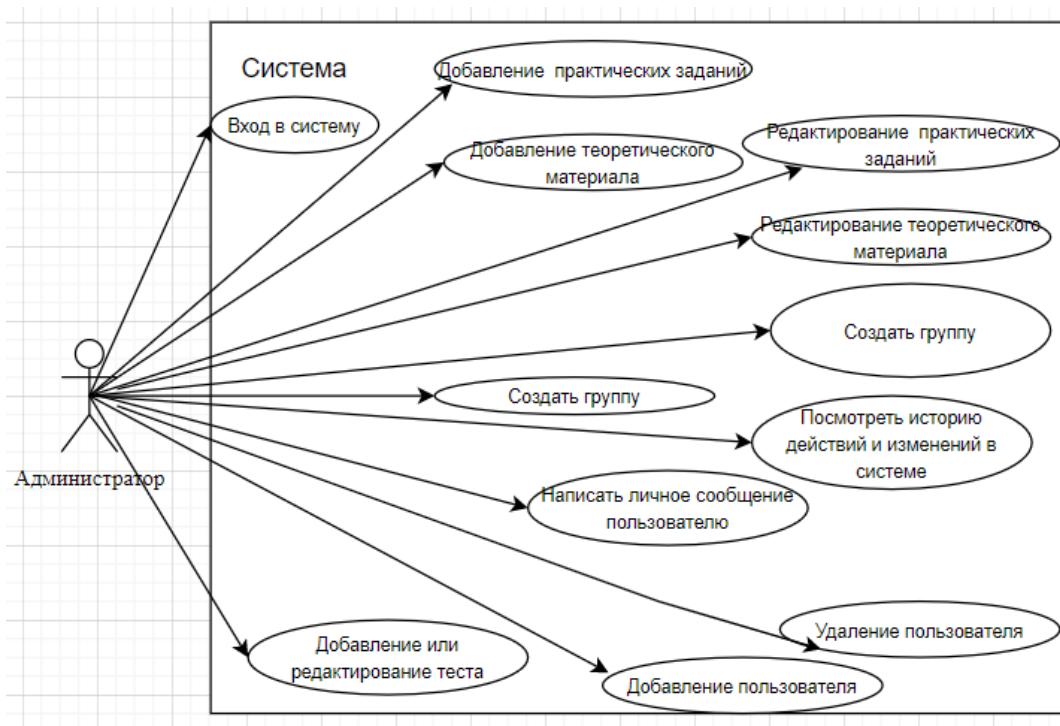


Рис. 3. Диаграмма прецедентов для администратора

С помощью диаграмм деятельности подробнее разобран процесс добавления и проведения контрольного мероприятия (см. рис. 4).

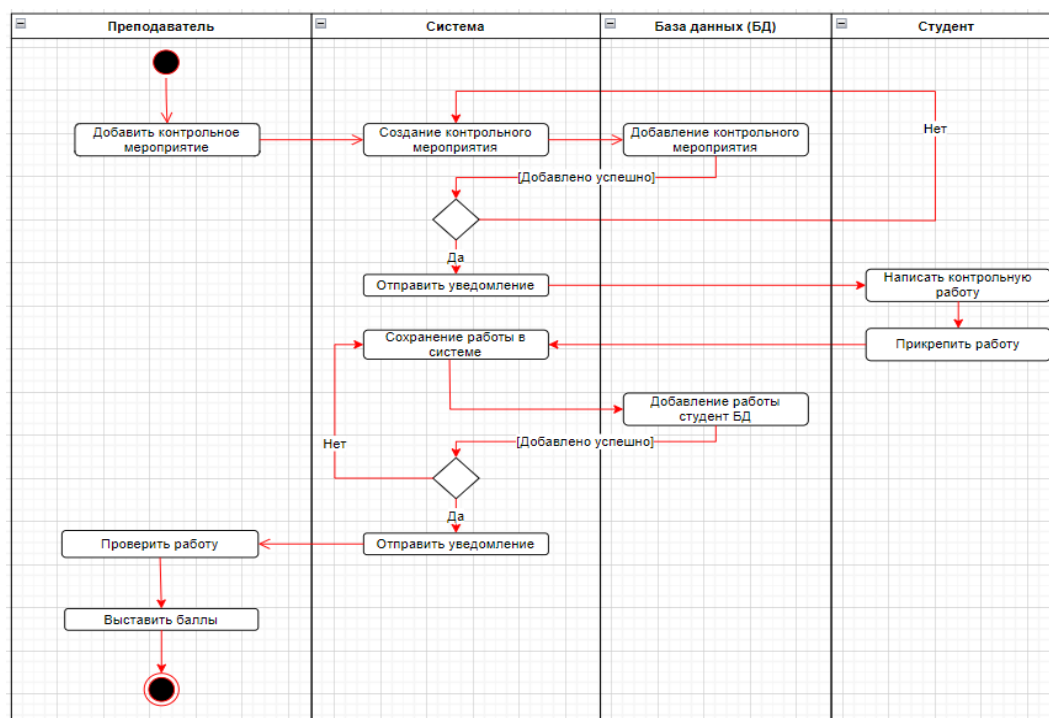


Рис. 4. Диаграмма деятельности для добавления и проведения контрольного мероприятия

Также с помощью диаграммы последовательностей, изображенной на рис. 5, рассмотрен процесс добавления студента в систему по ссылке от преподавателя.

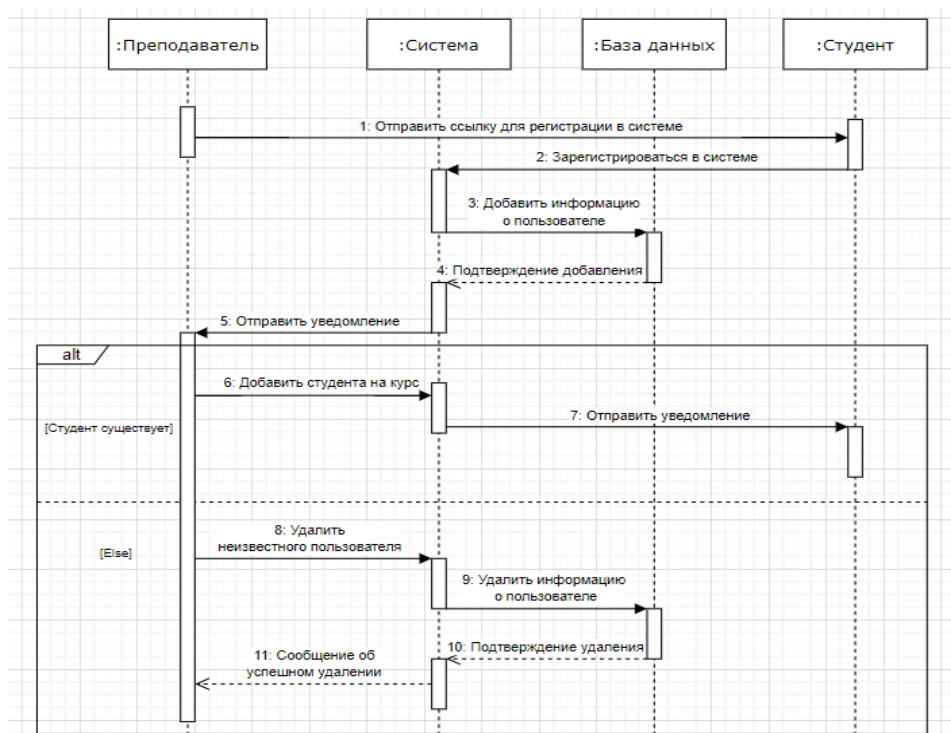


Рис. 5. Диаграмма последовательностей для процесса добавления студента в систему с помощью ссылки

С помощью диаграммы классов описана разрабатываемая система. На рис. 6 представлена диаграмма классов. Задания будут представлять из себя формулировку задания и возможное решение задания, на случай если у студента возникнут трудности при отработке своих навыков. В конце каждой главы теоретического материала студенту будет предложено пройти тест для проверки и оттачивания своих знаний.

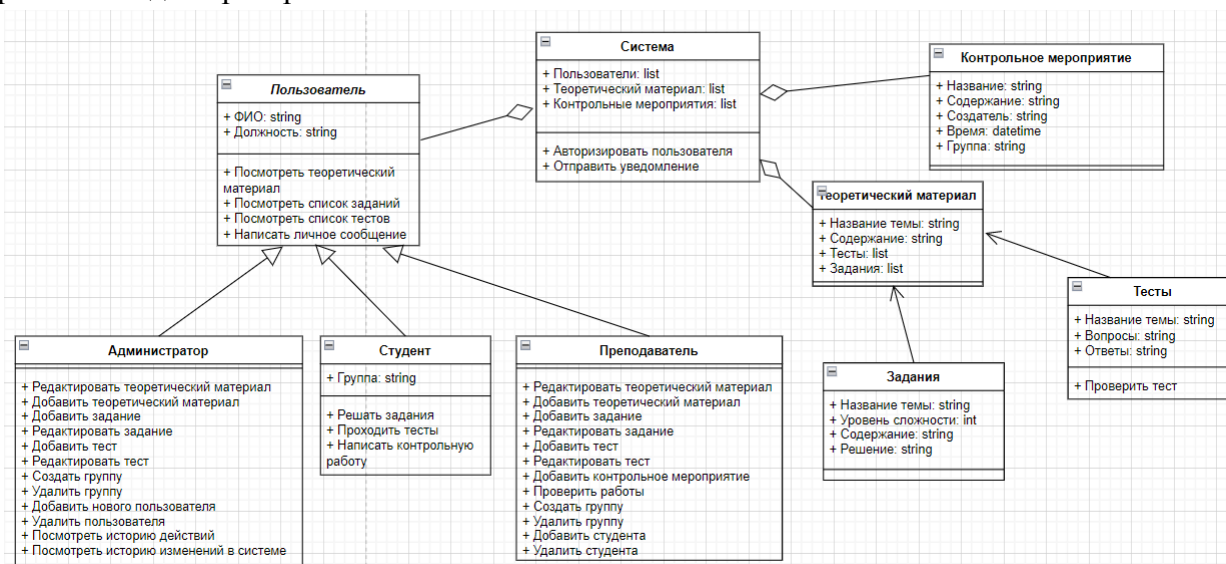


Рис. 6. Диаграмма классов

Перед этапом разработки базы данных были определены сущности, которые хранятся в БД, их взаимосвязь и атрибуты.



В БД хранится информация по каждому зарегистрированному пользователю; группы учащиеся в системе; теоретический материал, практические задания и тесты по каждой теме; личные сообщения; контрольные мероприятия.

Определены взаимосвязи между сущностями.

Отношение между «преподавателем» и «группой» является отношением многие ко многим, так как у группы может быть несколько преподавателей (например, если лекции и практические занятия ведут разные преподаватели), у каждого преподавателя в свою очередь может быть несколько групп.

Отношение между «преподавателем» и «контрольными мероприятиями» является отношением один ко многим, так как преподаватель может проводить несколько различных контрольных работ.

Отношение между «преподавателем» и «личными сообщениями» является отношением один ко многим, так как преподаватель может отправить более одного сообщения.

Отношение между «группой» и «студентом» является отношением один ко многим, так как в группе может учиться от одного и более студентов.

Отношение между «студентом» и «теоретическим материалом» является отношением один ко многим, так как студент может проходить одновременно несколько тем.

Отношение между «теоретическим материалом» и «практическими заданиями» является отношением один ко многим, так как по одной теме может быть предложено несколько заданий для практики.

Отношение между «теоретическим материалом» и «тестами» является отношением один к одному, так как на каждую тему рассчитан один, закрепляющий знания, тест.

Отношение между «студентом» и «контрольными мероприятиями» является отношением один ко многим, так как студент за время учебы может написать несколько контрольных работ.

Отношение между «студентом» и «личными сообщениями» является отношением один ко многим, так как студент может отправить более одного сообщения.

Основываясь на созданных UML диаграммах и на модели базы данных был создан возможный пользовательский интерфейс информационной системы. Он сделан максимально простым и интуитивно понятным.

После того, как пользователь войдет в систему, он попадет на главную страницу, где будет представлена информация о том, что это за информационная система, а также информация о новых добавлениях и изменениях в учебном материале. Также в системе есть меню с навигацией по ней, что существенно облегчает перемещение по ней. На рис. 7 представлен пример страницы, где в меню с навигацией выбрана вкладка «С#».

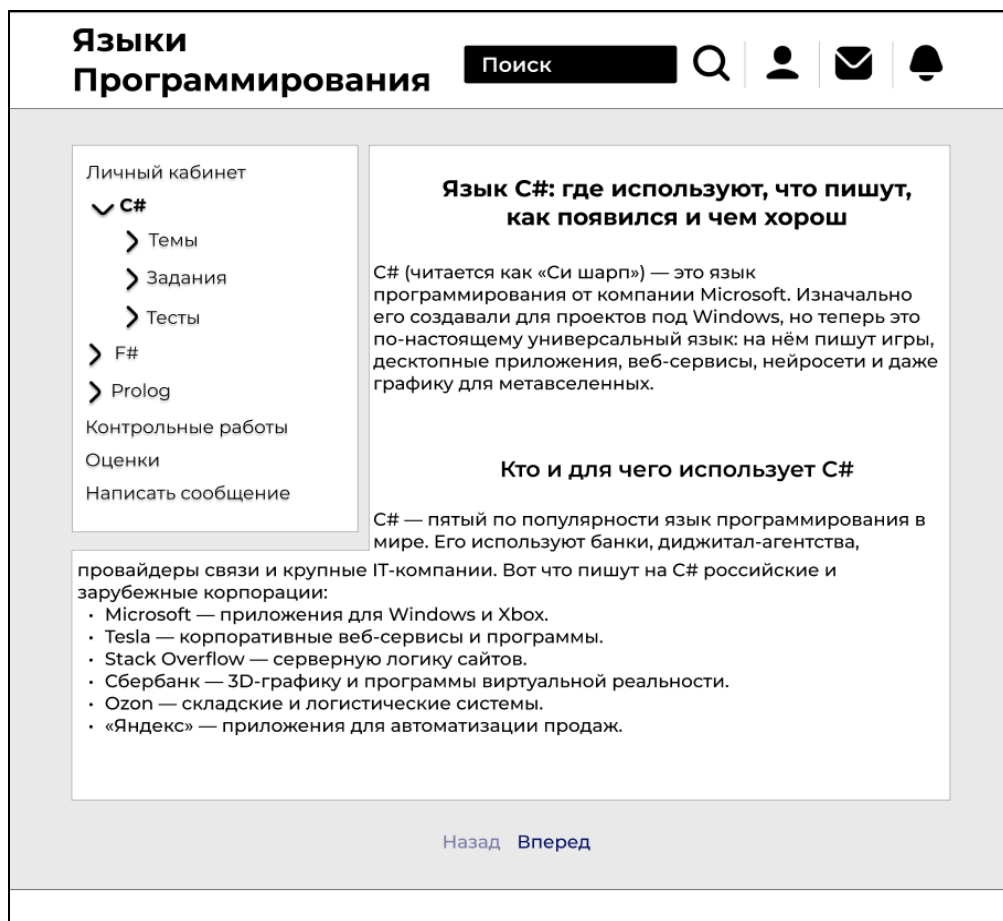


Рис. 7. Прототип интерфейса страницы про C#

Теперь рассмотрим внешний вид личного кабинета каждого из пользователей. Они различаются в плане функциональных возможностей для каждого вида пользователей. На рис. 8, 9, представлены варианты возможных интерфейсов личных кабинетов для студента и преподавателя, соответственно.

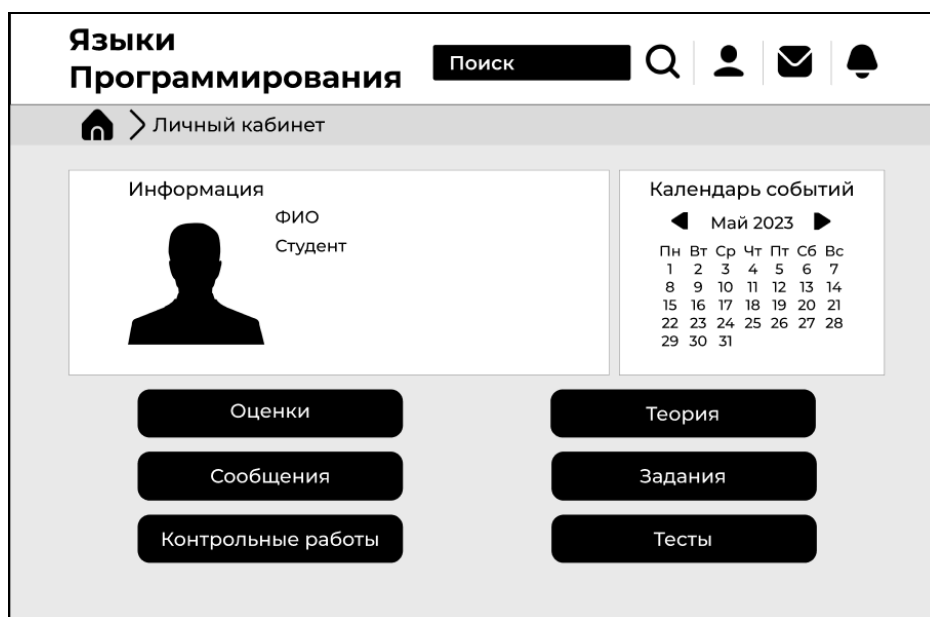


Рис. 8. Прототип интерфейса личного кабинета студента

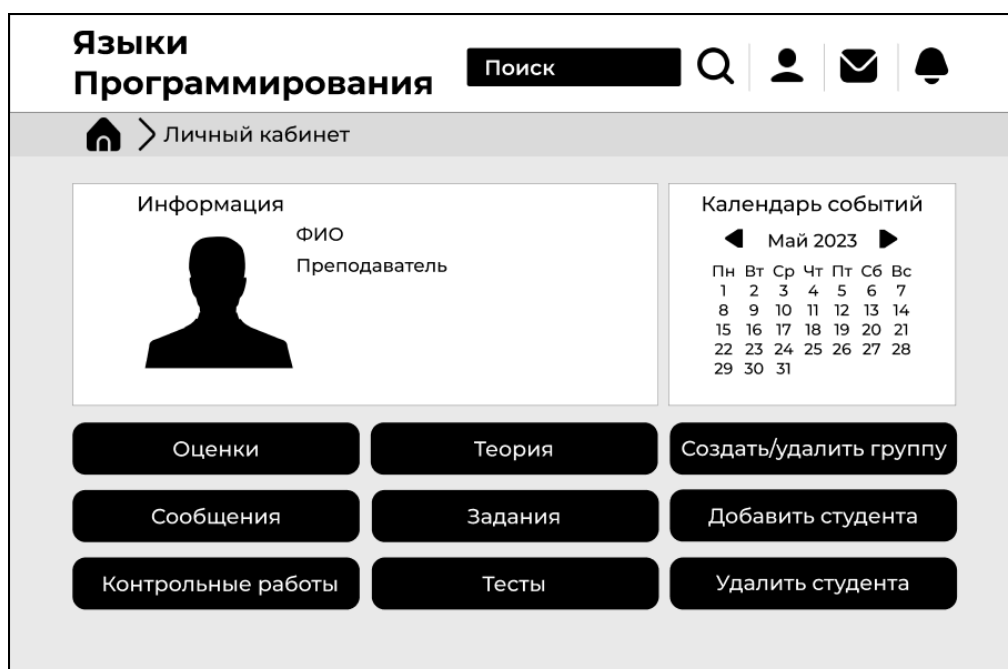


Рис. 9. Прототип интерфейса личного кабинета преподавателя

### Библиографический список

1. ЕТИС ПГНИУ [Электронный ресурс]. – URL: <https://student.psu.ru/> (дата обращения 26.04.23).
2. *Городняя Л.В.* Основы функционального программирования : учебное пособие / Л. В. Городняя. – 3-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 246 с. – ISBN 978-5-4497-0932-5. – Текст: электронный // Электронно-библиотечная система IPR BOOKS [Электронный ресурс].
3. *Братко И.* Алгоритмы искусственного интеллекта на языке Prolog: перевод с английского/И. Братко; пер. К. А. Птицына. – Москва: Вильямс, 2004. – 611 с. – ISBN 5-8459-0664-4.-640. – Текст: электронный // Электронно-библиотечная система RoyalLib.com [Электронный ресурс].
4. *Павловская Т.А.* Программирование на языке высокого уровня C#: учебное пособие / Т. А. Павловская. – 3-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 245 с. – ISBN 978-5-4497-0862-5. – Текст: электронный // Электронно-библиотечная система IPR BOOKS [Электронный ресурс].
5. METANIT.COM – Сайт о программировании [Электронный ресурс]. – URL: <https://metanit.com/> (дата обращения: 28.04.2023).
6. Prolog Site [Электронный ресурс]. – URL: <https://sites.google.com/site/prologsite/> (дата обращения: 28.04.2023).
7. Code Abbey [Электронный ресурс]. – URL: <http://www.codeabbey.com/> (дата обращения: 28.04.2023).
8. Programming Skills [Электронный ресурс]. – URL: <http://www.pskills.org/csharp.jsp> (дата обращения: 28.04.2023).
9. Draw.io [Электронный ресурс]. – URL: <https://www.drawio.com/> (дата обращения: 02.05.2023).
10. MySQL Workbench. – Текст: электронный // MySQL [Электронный ресурс]. – URL: <https://www.mysql.com/products/workbench/> (дата обращения: 02.05.2023).

11. Visual Studio: IDE. – Текст: электронный // Microsoft [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/> (дата обращения: 02.05.2023).
12. HTML.COM [Электронный ресурс]. – URL: <https://html.com/> (дата обращения: 02.05.2023).
13. Cascading Style Sheets. – Текст: электронный // W3 [Электронный ресурс]. – URL: <https://www.w3.org/Style/CSS/Overview.en.html> (дата обращения: 02.05.2023).
14. MySQL [Электронный ресурс]. – URL: <https://www.mysql.com/> (дата обращения: 03.05.2023).
15. Figma: the collaborative interface design tool [Электронный ресурс]. – URL: <https://www.figma.com/> (дата обращения: 03.05.2023).

### References

1. ETIS PGNIU [Electronic resource]. – URL: <https://student.psu.ru/> (date of application 26.04.23). (In Russ.).
2. *Gorodnyaya L.V. Osnovy funktsional'nogo programmirovaniya : uchebnoe posobie / L. V. Gorodnyaya.* – 3-e izd. – Moskva: Internet-Universitet Informacionnyh Tekhnologij (INTUIT), Aj Pi Ar Media, 2021. – 246 c. – ISBN 978-5-4497-0932-5. – Electronic text // Electronic library system IPR BOOKS [Electronic resource]. (In Russ.).
3. *Bratko I. Algoritmy iskusstvennogo intellekta na yazyke Prolog: perevod s anglijskogo/I. Bratko; per. K. A. Pticyna.* – Moskva: Vil'yams, 2004. – 611 c. – ISBN 5-8459-0664-4. – Electronic text // Electronic library system RoyalLib.com [Electronic resource]. (In Russ.).
4. *Pavlovskaya T.A. Programmirovaniye na yazyke vysokogo urovnya C#: uchebnoe posobie / T. A. Pavlovskaya.* – 3-e izd. – Moskva: Internet-Universitet Informacionnyh Tekhnologij (INTUIT), Aj Pi Ar Media, 2021. – 245 c. – ISBN 978-5-4497-0862-5. – Electronic text // Electronic library system IPR BOOKS [Electronic resource]. (In Russ.).
5. METANIT.COM – Sajt o programmirovanii [Electronic resource]. – URL: <https://metanit.com/> (date of application 28.04.2023). (In Russ.).
6. Prolog Site [Electronic resource]. – URL: <https://sites.google.com/site/prologsite/> (date of application 28.04.2023).
7. Code Abbey [Electronic resource]. – URL: <http://www.codeabbey.com/> (date of application 28.04.2023).
8. Programming Skills [Electronic resource]. – URL: <http://www.pskills.org/csharp.jsp> (дата обращения: 28.04.2023).
9. Draw.io [Electronic resource]. – URL: <https://www.drawio.com/> (дата обращения: 02.05.2023).
10. MySQL Workbench. – Electronic text // MySQL [Electronic resource]. – URL: <https://www.mysql.com/products/workbench/> (date of application 02.05.2023).
11. Visual Studio: IDE. – Electronic text // Microsoft [Electronic resource]. – URL: <https://visualstudio.microsoft.com/> (date of application 02.05.2023).
12. HTML.COM: [сайт]. – URL: <https://html.com/> (дата обращения: 02.05.2023).
13. Cascading Style Sheets. – Electronic text // W3 [Electronic resource]. – URL: <https://www.w3.org/Style/CSS/Overview.en.html> (date of application 02.05.2023).
14. MySQL [Electronic resource]. – URL: <https://www.mysql.com/> (дата обращения: 03.05.2023).
15. Figma: the collaborative interface design tool [Electronic resource]. – URL: <https://www.figma.com/> (date of application 03.05.2023).

## **DESIGNING AND DOCUMENTING AN INFORMATION SYSTEM FOR TESTING KNOWLEDGE IN THE DISCIPLINE «PROGRAMMING LANGUAGES»**

*Yaparov Artem E., Rakina Valeria D.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
Yaparov\_2002@mail.ru

This paper is devoted to the design and documentation of the informational system for examination of students knowledge of the university subject "programming languages". This paper consists of problem area analysis and selection of design tools, also core features have been identified, and UML diagrams have been designed with data base and graphical user interface.

Keywords: information system design and documentation, programming languages, knowledge testing.

*Научное издание*

**Актуальные проблемы математики,  
механики и информатики 2023**

Сборник статей по материалам студенческой конференции  
г. Пермь, ПГНИУ, 24–28 апреля 2023 г. (1 сессия),  
13–20 июня 2023 г. (2 сессия)

Статьи публикуются в авторской редакции, авторы несут ответственность  
за содержание статей, за достоверность приведенных в статье фактов, цитат,  
статистических и иных данных, имен, названий и прочих сведений.

Каждая статья была оценена двумя рецензентами

Ответственный редактор *М. М. Бузмакова*

Издается в авторской редакции  
Компьютерная верстка: *Д. Р. Бакирова*

---

Объем данных 15,1 Мб  
Подписано к использованию 01.11.2023

---

Размещено в открытом доступе  
на сайте [www.psu.ru](http://www.psu.ru)  
в разделе НАУКА / Электронные публикации  
и в электронной мультимедийной библиотеке ELiS

Управление издательской деятельности  
Пермского государственного  
национального исследовательского университета  
614068, г. Пермь, ул. Букирева, 15