

АКТУАЛЬНЫЕ ПРОБЛЕМЫ МАТЕМАТИКИ, МЕХАНИКИ И ИНФОРМАТИКИ 2024

Сборник статей по материалам
студенческой конференции

г. Пермь, ПГНИУ,
22–26 апреля 2024 г. (1 сессия),
20–24 мая 2024 г. (2 сессия),
17–28 июня 2024 г. (3 сессия)



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 2024

Сборник статей по материалам студенческой конференции

*г. Пермь, ПГНИУ,
22–26 апреля 2024 г. (1 сессия),
20–24 мая 2024 г. (2 сессия),
17–28 июня 2024 г. (3 сессия)*



Пермь 2024

УДК 51+531+004.8](082)

ББК 22+32.81

A437

A437 **Актуальные** проблемы математики, механики и информатики 2024 [Электронный ресурс]: сборник статей по материалам студенческой конференции «Актуальные проблемы математики, механики и информатики 2024» г. Пермь, ПГНИУ, 22–26 апреля 2024 г. (1 сессия), 20–24 мая 2024 г. (2 сессия), 17–28 июня 2024 г. (3 сессия) / под редакцией А. В. Черникова, М. М. Бузмаковой ; Пермский государственный национальный исследовательский университет. – Электронные данные. – Пермь, 2024. – 12,6 Мб ; 237 с. – Режим доступа: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematikimekhaniki-informatiki-2024.pdf>. – Заглавие с экрана.

ISBN 978-5-7944-4187-1

В сборнике представлены материалы студенческой конференции «Актуальные проблемы математики, механики и информатики 2024», которая проводилась 22–26 апреля 2024 г. (1 сессия), 20–24 мая 2024 г. (2 сессия), 17–28 июня 2024 г. (3 сессия) в г. Перми.

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

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 ученого совета физико-математического института
Пермского государственного национального исследовательского университета*

Рецензенты: канд. физ.-мат. наук, доцент кафедры информационной безопасности Астраханского государственного университета **В. А. Черкасова**;

д-р. техн. наук, доцент, заведующий кафедрой оборудования и автоматизации химических производств ПНИПУ **Е. Р. Мошев**

ISBN 978-5-7944-4187-1

© ПГНИУ, 2024

СОДЕРЖАНИЕ

Ахматова У. А., Кнутова Н. С. ПРОЕКТИРОВАНИЕ МОДУЛЯ «ИЗМЕРЕНИЕ СТРЕССОУСТОЙЧИВОСТИ» У МИКРОЗЕЛЕНИ ДЛЯ ИС «ПРАКТИКА НА БИОЛОГИЧЕСКОМ ФАКУЛЬТЕТЕ».....	5
Базеев А. К., Аверин С. И. ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ СОПРОВОЖДЕНИЯ РЕТЕЙЛА	16
Безматерных А. Е., Бузмакова М. М. ИДЕНТИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ ОБЪЕКТОВ НА МИКРОИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ПОДХОДОВ ТЕОРИИ ПЕРКОЛЯЦИИ.....	23
Вершинин Н. А. ПРОЕКТИРОВАНИЕ СИСТЕМЫ КЛАССИФИКАЦИИ ОДЕЖДЫ НА ФОТОГРАФИЯХ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ.....	35
Водяник Р. А., Ракина В. Д. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ ПО СНОУБОРДУ	43
Воробьева Д. Ю., Кнутова Н. С. ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «ПЛАНИРОВЩИК РОМАНОВ»..	51
Голдобин Д. А., Гасумова С. Е. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ СОЦИАЛЬНОГО ТАКСИ	62
Дьякова И. А., Степанов В. А. ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВЫБОРА И БРОНИРОВАНИЯ МЕСТ В РЕСТОРАНАХ	69
Золотарев Д. С. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ ВЫЯВЛЕНИЯ ФЕЙКОВЫХ ИЗОБРАЖЕНИЙ МЕТОДАМИ АНАЛИЗА УРОВНЕЙ СЖАТИЯ СВЕРТОЧНОЙ НЕЙРОСЕТЬЮ И АНАЛИЗА МЕТАДАННЫХ.....	77
Караваева К. А. ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МЕТОДОВ ТЕОРИИ ПЕРКОЛЯЦИИ ДЛЯ ИССЛЕДОВАНИЯ КРУПНОМАСШТАБНОЙ СТРУКТУРЫ ВСЕЛЕННОЙ	81
Колпашиков М. Н., Кнутова Н. С. ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ОНЛАЙН-СЕРВИСА С API ДЛЯ ЗАПОЛНЕНИЯ ДОКУМЕНТОВ ПО ШАБЛОНАМ	86
Костицын В. В., Кнутова Н. С. ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ANDROID- ПРИЛОЖЕНИЯ ДЛЯ КОНТРОЛЯ ЛИЧНЫХ ФИНАНСОВ И КУРСА ВАЛЮТ	98
Кузнецов А. М., Гасумова С. Е. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ФУДШЕРИНГА.....	105

Лешехва Е. А., Кушев В. О. ПРОЕКТИРОВАНИЕ И ПРОТОТИПИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ПОДДЕРЖКИ ПРОЕКТОВ ДЕТСКОГО ТЕХНОПАРКА «КВАНТОРИУМ ФОТОНИКА»	113
Мальцев В. Л., Кнутова Н. С. ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНОЙ ИГРЫ В ЖАНРЕ ПЛАТФОРМЕР	125
Маткин М. А., Аверин С. И. ИСТОРИКИ ПЕРМСКОГО КРАЯ 19–20 ВВ.: ПРОЕКТ ПРОСОПОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ	133
Моздакова А. А., Ракина В. Д. ПРОЕКТИРОВАНИЕ МОДУЛЯ «МУНИЦИПАЛЬНАЯ СОБСТВЕННОСТЬ» ИНФОРМАЦИОННОЙ СИСТЕМЫ ОТДЕЛА МУНИЦИПАЛЬНОГО ОБРАЗОВАНИЯ ДЕПАРТАМЕНТА ИМУЩЕСТВЕННЫХ ОТНОШЕНИЙ ГОРОДА ПЕРМИ	138
Низаметдинов Р. Э., Аверин С. И. ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭКОЛОГИЧЕСКИЙ ТУРИЗМ ПЕРМСКОГО КРАЯ»	146
Осипов А. Д., Барулина М. А. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ВОССТАНОВЛЕНИЕ ТРЕХМЕРНОЙ МОДЕЛИ СОСУДОВ СЕРДЦА ПО ДВУХМЕРНЫМ ИЗОБРАЖЕНИЯМ КОРОНАРОГРАФИИ»	151
Парамонов А. С., Кушев В. О. ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ УЧЕТА ТОВАРООБОРОТА МАГАЗИНА.....	165
Русаков В. В., Мокунева В. Е., Селетков И. П. УПРАВЛЕНИЕ НАБОРАМИ ДАННЫХ В NO-CODE СИСТЕМЕ АНАЛИЗА ДАННЫХ ..	175
Сычев И. А., Соколова О. Л., Яшичев Д. Л., Соколов А. В. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ УПРАВЛЕНИЯ БИЗНЕСОМ В СФЕРЕ ОБЩЕСТВЕННОГО ПИТАНИЯ	183
Кнутова Н. С., Уткина А. В. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПРАКТИКА НА БИОЛОГИЧЕСКОМ ФАКУЛЬТЕТЕ ПГНИУ»	198
Шаров Р. В., Ракина В. Д. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРЕПОДАВАНИЯ БЛОКА АЛГОРИТМИЗАЦИИ В ДИСЦИПЛИНЕ «ИНФОРМАТИКА»	210
Язык М. Д., Аверин С. И. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОНТРОЛЯ И ПРОГНОЗИРОВАНИЯ ФИНАНСОВ	220
Ясырев М. Е., Ракина В. Д. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОТСЛЕЖИВАНИЯ БРАЧНОГО ЛЁТА МУРАВЬЕВ.....	225

ПРОЕКТИРОВАНИЕ МОДУЛЯ «ИЗМЕРЕНИЕ СТРЕССОУСТОЙЧИВОСТИ У МИКРОЗЕЛЕНИ» ДЛЯ ИС «ПРАКТИКА НА БИОЛОГИЧЕСКОМ ФАКУЛЬТЕТЕ»

Ахматова Ульяна Александровна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, uahmatova@mail.ru

Кнутова Наталья Сергеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, talieknutova@gmail.com

Рассматривается проектирование модуля для информационной системы «Практика на биологическом факультете». Предполагается, что модуль будет решать задачу замера длин стеблей микрозелени, а также расчёта фитотоксичности – на основании этих данных биологи смогут сделать выводы для дальнейших исследований. В работе проанализированы существующие алгоритмы и системы автоматизации замеров и расчётов, определены требования к проектируемой системе и выбраны программные средства для разработки системы. Статья содержит описание логики работы модуля ИС, модель данных, а также пример возможного интерфейса пользователя.

Ключевые слова: стрессоустойчивость микрозелени, фитотоксичность почвы, компьютерное зрение.

В лабораториях проводится множество опытов, например, в области физики, биологии и других наук. Все они требуют точности, скорости и надёжности, однако обработка большого количества данных может быть сложной и трудоёмкой задачей особенно, если данные процессы осуществляются вручную.

Использование информационных систем позволяет существенно сократить время на выполнение задач, а также исключить возможные ошибки, связанные с человеческим фактором. Они обеспечивают автоматизацию процессов, что, в свою очередь, позволяет значительно повысить эффективность работы.

Системы позволяют получать более точные данные, что является ключевым фактором для принятия обоснованных решений. Кроме того, такие ИС позволяют анализировать большие объёмы данных, что помогает выявлять закономерности и тенденции.

Таким образом, актуальность темы данной работы обусловлена необходимостью повышения эффективности и точности процессов замера длины стеблей у микрозелени для дальнейшего анализа ситуации. Информационная система позволит в целом автоматизировать практику у биологов, а модуль – процесс работы с микрозеленью, помогая снизить ве-

роятность возможных ошибок и неточностей, вызванных человеческим фактором. Это, в свою очередь, сократит время на обработку данных и позволит получить точные результаты исследований для последующего изучения, мониторинга и корректировки условий.

Для того, чтобы понимать, какую систему спроектировать и как в дальнейшем реализовать, необходимо быть в контексте предметной области, поэтому рассмотрим её более подробно.

Микрозелень – молодые побеги растений. Она бывает различных видов, в зависимости от того, какие семена используются для её выращивания. Каждый вид микрозелени имеет свои уникальные свойства и применяется для разнообразных целей, например, в кулинарии, медицине или косметологии. На биологическом факультете ПГНИУ же используют кресс-салат.

Кресс-салат – однолетнее овощное растение, обладающее повышенной чувствительностью к разного рода стрессорам (загрязнение, нехватка влаги, света и т.д.). Этот биоиндикатор отличается быстрым прорастанием семян и почти стопроцентной всхожестью, которая заметно уменьшается в присутствии загрязнителей. Кроме того, побеги и корни этого растения под действием стрессоров подвергаются заметным морфологическим изменениям таким как задержка роста и искривление побегов, уменьшение длины и массы корней, а также числа и массы семян [1].

Кресс-салат как биоиндикатор удобен ещё и тем, что действие стрессоров можно изучать одновременно на большом числе растений при небольшой площади рабочего места. Привлекательны также и весьма короткие сроки эксперимента. Семена прорастают уже на 3-4 день, и на большинство вопросов эксперимента можно получить ответ в течение 2-х недель [1].

Через 10-15 дней роста можно провести замеры высоты стеблей 20-30 растений, срезая на уровне поверхности почвы. Результаты данного опыта подвергают математической обработке. Отличие средней высоты одного растения, выращенного на почвенных пробах, от тест-контроля обосновывают параметрическими и/или непараметрическими тестами [2].

Критерии оценки биологической активности и токсичности исследуемого почвенного материала основаны на том, что по сравнению с вермикулитом (субстратом неорганического и фактически естественного происхождения) растения на чернозёме (самой плодородной почве в регионе) имели высоту ниже на 30%, чем тест-контроль. Исходя из этих показателей, следует отметить, что при снижении показателей развития кресс-салата на 10-30% состояние почвы или грунта считать удовлетворительным; при снижении на 30-50% – неудовлетворительным; а при снижении показателей более, чем на 50% считать экологически опасным [2].

Рассмотрим на примере лабораторного опыта, в котором смотрели проявление фитотоксичности почвы. Фитотоксичность – подавление прорастания семян, роста и развития высших растений, т.е. чем выше уровень фитотоксичности почвы, тем ниже всхожесть.

Кресс-салат выращивался на почве, ранее загрязненной тяжелыми металлами. Для этого использовались различные образцы почвы, токсичность которых оценивалась по реакции микрозелени. В качестве тест-контроля использовали кресс-салат, выращенный на «чистой» почве, без примеси тяжёлых металлов. Измерения высоты были проведены на пятнадцати 10-дневных побегах и указаны в сантиметрах.

Результаты замеров стеблей на различных почвенных пробах представлены в таблице 1.

Таблица 1

Замеры стеблей

Контроль (суглинок)	Рь 1 ОДК	Рь 0,75 ОДК	Рь 0,5 ОДК
4,6	5,4	5	6,1
5	5,6	5,9	5,7
5,1	5,1	6,5	6,2
5	5,1	4,7	5,4
5	5,4	5,9	4,8

После того, как замеры были проведены, необходимо их математически обработать, а именно – определить проявление фитотоксичности.

Фитотоксичность рассчитывается по следующей формуле: $\Phi = (\text{длина ростка на контроле} - \text{длина на экспериментальном участке}) * 100 / \text{длина ростка на контроле}$.

Таблица 2

Расчёты фитотоксичности

Контроль (суглинок)	Рь 1 ОДК		Рь 0,75 ОДК	
Длина, см	Длина, см	Фито-ть, %	Длина, см	Фито-ть, %
4,60	5,40	-17,39	5,00	-8,70
5,00	5,60	-12,00	5,90	-18,00
5,10	5,10	0,00	6,50	-27,45
5,00	5,10	-2,00	4,70	6,00
5,00	5,40	-8,00	5,90	-18,00
Среднее		-7,88		-13,23

Основываясь на этих данных, биологи смогут сформулировать дальнейшие выводы для своих исследований: насколько фитотоксична почва или насколько стрессоустойчива зелень к внешним факторам.

В открытом доступе было найдено два приложения, которые решают похожую задачу – измерение площади листьев. Они представлены на рис. 1 и 2.

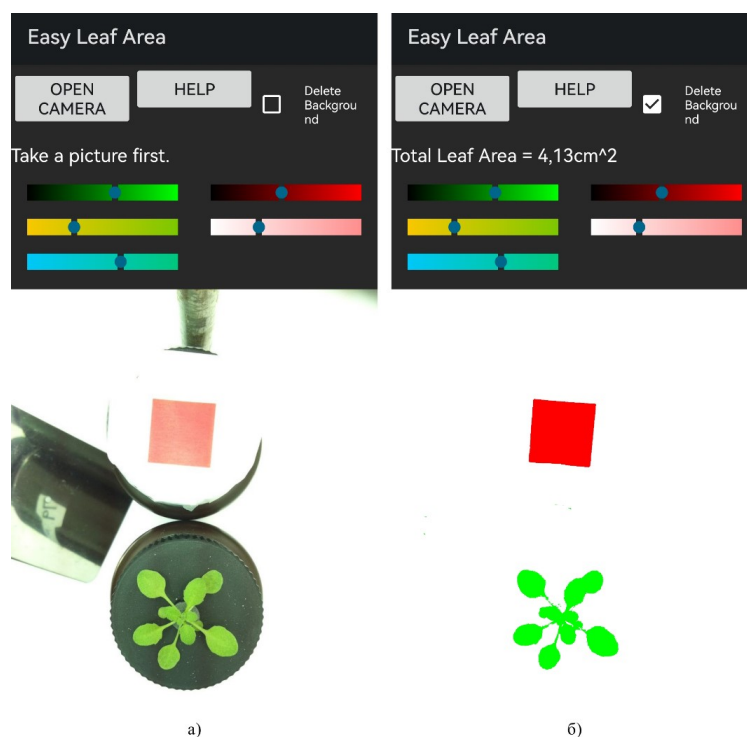


Рис. 1. Работа приложения Easy Leaf Area Free [3]

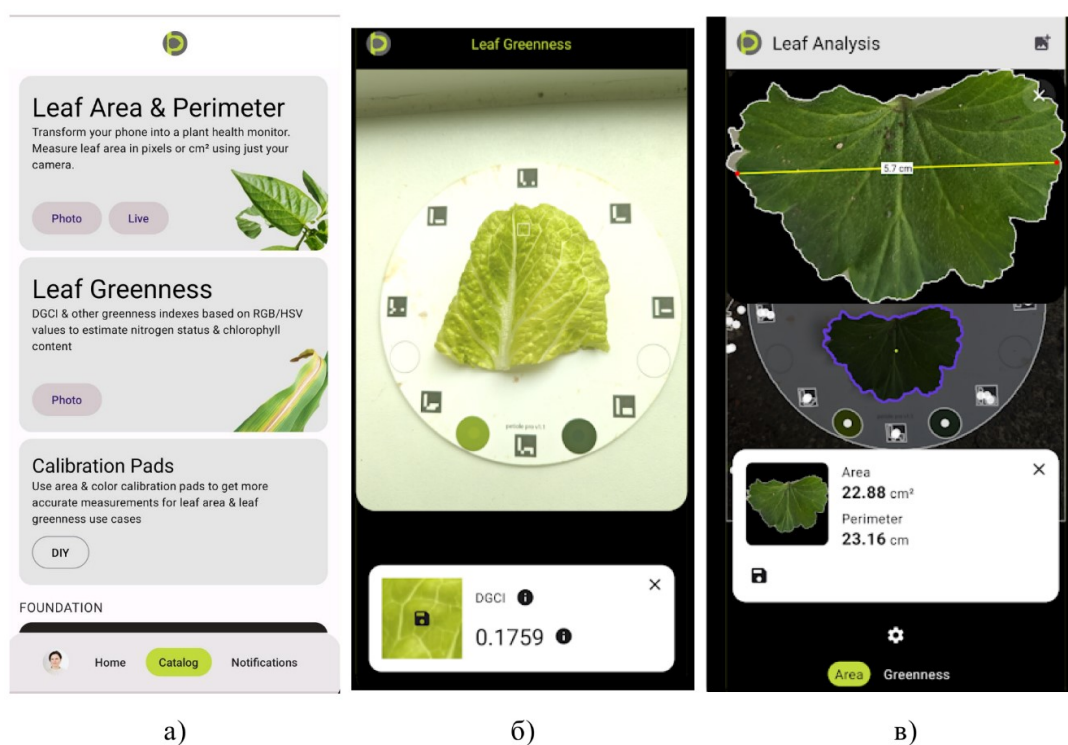


Рис. 2. Работа приложения Petiole [4]

Оба приложения используют технологию машинного обучения для точного определения площади листьев на изображениях, их основные функции: автоматическое обнаружение контуров листа; распознавание нескольких листьев; измерение площади листьев; калибровка; представление результатов замера.

Приложения имеют отличия только в дополнительных функциях, таких как: возможность делиться результатами; оценка содержания хлорофилла в листьях.

На основе проведённого анализа предметной области можно понять, что систем, посвященных именно измерению длины стеблей и расчету стрессоустойчивости у микрорастений, нет.

Для реализации такого модуля понадобится компьютерное зрение – технология на основе искусственного интеллекта, которая помогает обнаруживать объекты на изображении.

Рассмотрим основные процессы и методы компьютерного зрения [5].

1. Сбор данных и предварительная обработка. На данном этапе машина получает некое изображение из окружающей среды либо с помощью датчиков, либо человек загружает файл вручную. После того, как изображение было загружено, могут измениться его размеры, поменяться яркость/контраст и т.д. – это называется предварительной обработкой и используется для улучшения качества изображения для последующих действий.

2. Обработка и анализ изображений. Здесь определяются и выделяются какие-либо характеристики/признаки изображения, с помощью которых можно распознать объекты. Машина распознаёт части изображения, идентифицирует объекты и сопоставляет с существующими шаблонами.

3. Машинное обучение: с учителем и без. Данный этап позволяет системе классифицировать и интерпретировать полученные данные и работать с ними в дальнейшем.

Как будут выполняться замеры длин стеблей по фотографиям:

1. Пользователю необходимо срезать стебель кресс-салата.
2. Требуется расположить стебель на листке клетчатой бумаги и сфотографировать.
3. Для того, чтобы расчёты были корректными, обязательно надо задать длину стороны клетки в сантиметрах. Например, для стороны клетки тетрадного листа необходимо указать 0.5.

4. После загрузки изображения и указания масштаба приложение с помощью компьютерного зрения определит контуры стебля и произведёт расчёт длины.

5. Далее результат будет выведен на экран, а также передан для сохранения.

6. На основании замеров всех стеблей в выборке пользователь сможет рассчитать фитотоксичность.

В качестве средства для проектирования был выбран инструмент Diagrams.net [6] в связи с тем, что это доступный онлайн-ресурс, который подходит для всех типов ОС и в котором можно создавать диаграммы различных видов.

Для прототипирования интерфейсов была использована Figma [7]. Она предоставляет доступный широкий набор инструментов, который позволяет быстро и эффективно создавать прототипы.

Так как модуль подразумевает под собой измерение стебля по фотографии, то для этого понадобится машинное обучение и компьютерное зрение – в частности, библиотека OpenCV [8].

На данный момент библиотека поддерживается в таких языках программирования, как Python, C++, C#. Все эти языки являются высокоуровневыми и с помощью всех них можно разработать кроссплатформенные мобильные приложения.

Для написания кода модуля ИС был выбран язык программирования Python. Этот выбор обоснован тем, что он отличается наличием множества бесплатных материалов и ресурсов для изучения. Плюсом также является широкая поддержка данного ЯП в научных и исследовательских сферах.

Так как проектируемый модуль ИС будет встраиваться в мобильное приложение, то было принято решение использовать PyCharm в качестве основной среды разработки с подключением библиотеки Kivy.

В качестве дополнительной среды разработки остановимся на Google Colab. Он был выбран из-за бесплатного доступа, удобства использования в облачной среде и наличия мощных вычислителей для выполнения сложных задач. Он будет использоваться для написания и тестирования отдельных функций, которые можно в дальнейшем интегрировать в сам модуль.

Основной целью проектирования логической части является создание архитектуры, которая будет эффективно работать и обеспечивать требуемый функционал. На этом этапе необходимо определить базовые объекты, их взаимосвязи, принципы работы системы, а также описание процессов и логику выполнения операций.

Диаграммы прецедентов, которые описывают взаимодействие между системой и её окружением, позволяют увидеть проектируемый продукт с точки зрения пользователей, идентифицировать основные функциональные возможности и определить возможные сценарии поведения.

Рассмотрим отдельно диаграммы для администратора модуля ИС и для пользователя.

Так как модуль является частью информационной системы, то единственная функция, которая остаётся у администратора именно в модуле – его периодическое обновление. Взаимодействие представлено на рис. 3. Все остальные действия администратор будет совершать через саму ИС.

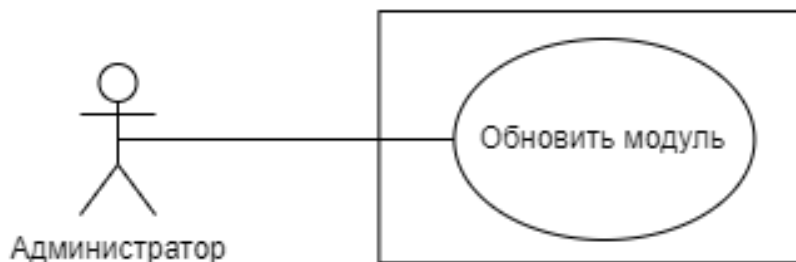


Рис. 3. Диаграмма прецедентов для администратора

У пользователя же набор доступного функционала гораздо больше. Пользователь запускает приложение, далее загружает либо файл из галереи, либо делает фотографию. После этого изображение автоматически передаётся в модуль и пользователь может продолжать работу.

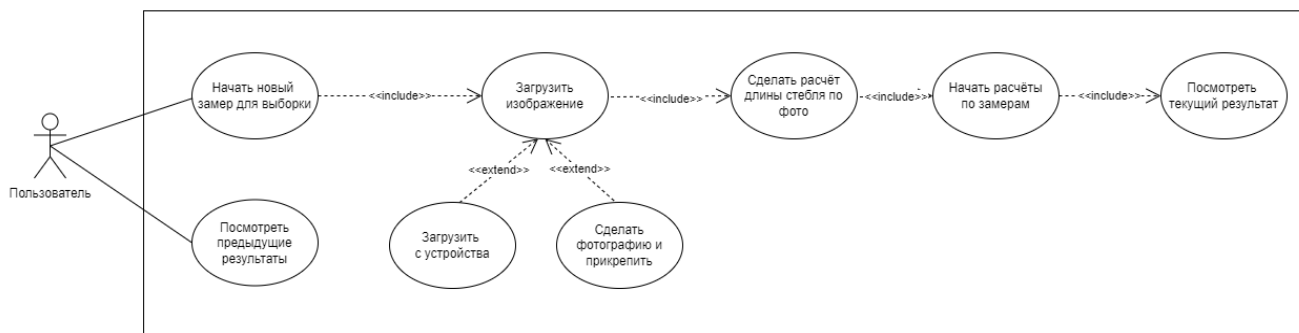


Рис. 4. Диаграмма прецедентов для пользователя

Рассмотрим прецеденты, которые представлены на рис. 4:

1. «Посмотреть предыдущие результаты»: пользователю будет предоставлен список ранее сделанных замеров, чтобы он мог выбрать один из них и ознакомиться таблицей результатов;

2. «Загрузить изображение»: пользователь может либо загрузить изображение из галереи, либо сделать фотографию прямо из приложения. После того, как пользователь подтвердил прикрепление изображения, то ИС передаёт его в модуль для дальнейшей обработки;

3. «Начать новый замер для выборки»: пользователю необходимо будет проименовать замер текущей выборки для того, чтобы в дальнейшем данные сохранились в таблице. После этого у него будет возможность ассчитывать длины стеблей, которая отражена в прецеденте «Сделать расчёт длины стебля по фото»;

4. «Начать расчёты по замерам»: модуль высчитывает показатели фитотоксичности на основе данного замера;

5. «Посмотреть текущий результат»: после того, как модуль определит длину стебля на фото, пользователь сможет увидеть информацию об этом на экране. Также, после замера всех стеблей в выборке, пользователю предоставится возможность ознакомиться с таблицей результатов. автоматически сохраняется и в дальнейшем передаются в оболочку ИС для хранения.

Диаграмма последовательностей, представленная на рис. 5, полностью отразила взаимодействие всех объектов в проектируемом модуле, а именно – как будет взаимодействовать модуль с оболочкой ИС и пользователь с модулем.

Проектирование, выполненное с помощью диаграмм, помогло полностью понять структуру работы модуля ИС.

Модуль «Измерение стрессоустойчивости у микрозелени» в дальнейшем, после его разработки, сможет автоматизировать и облегчить деятельность студентов, преподавателей и научных сотрудников биологического факультета.

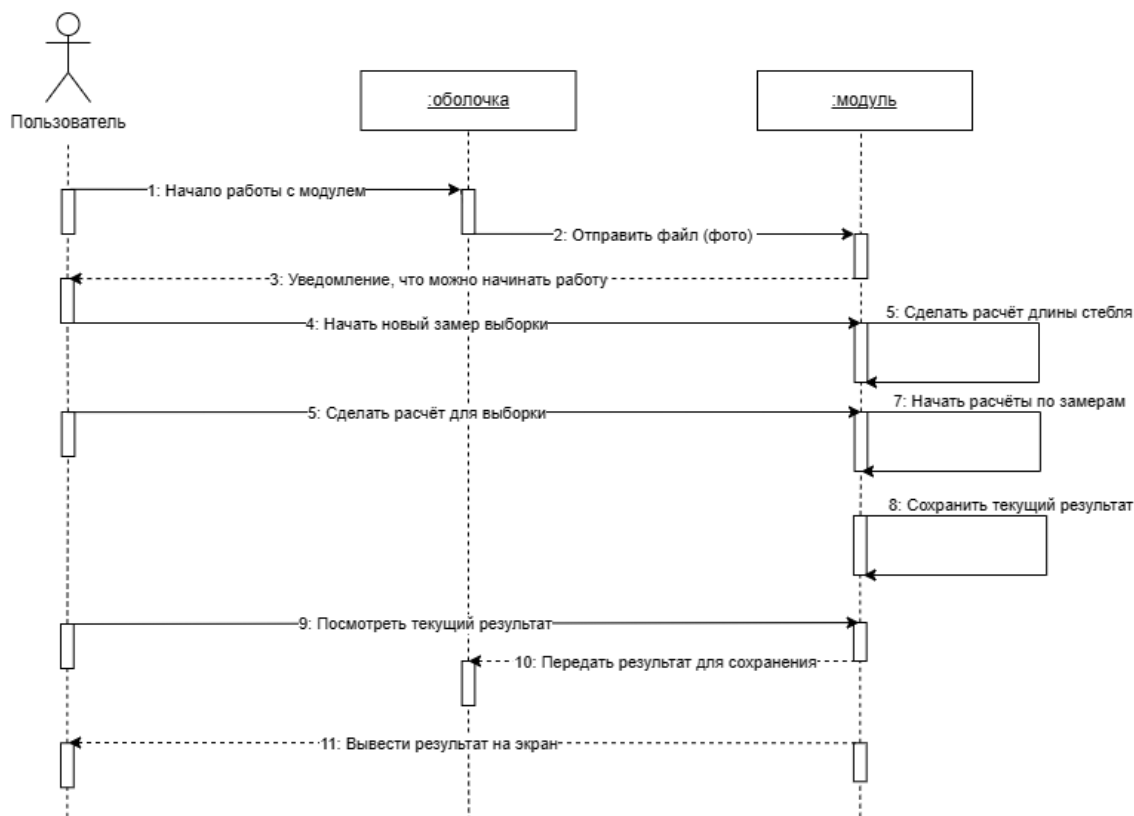


Рис. 5. Диаграмма последовательностей

На рисунках 6–8 показан примерный интерфейс проектируемого модуля.



Рис. 6. Интерфейс пользователя: начало работ



Рис. 7. Просмотр предыдущих результатов



Рис. 8. Новый замер

Библиографический список

1. Экологический мониторинг: Учеб.-метод. пособие. Изд. 3-е, испр. и доп. / Под ред. Т. Я. Ашихминой. М.: Академический Проект, 2006. 416 с. («Gaudeamus».)

2. Патент №2620555 Российская Федерация, МПК G01N 33/24 (2006.01). Способ оценки биологической активности и токсичности почв и техногенных почвогрунтов: №2016113050: заявл. 05.04.2016: опубл. 26.05.2017 / Еремченко О. З., Митракова Н. В. 15 с.
3. Мобильное приложение для определения площади листьев // Ботанический сад СВФУ. URL: <https://garden-nefu.ru/blog/front/view?id=7> (дата обращения: 01.05.2024).
4. Petiole Площадь листа // Apkpure. URL: <https://apkpure.net/ru/petiole-plant-leaf-area-meter/com.petioleapp.petiole> (дата обращения: 01.05.2024).
5. Компьютерное зрение в 2024 году: главные задачи и направления // Хабр. URL: <https://habr.com/ru/companies/otus/articles/810207/> (дата обращения: 20.05.2023).
6. Diagrams.net [Электронный ресурс] // Draw.io. URL: <https://www.drawio.com/> (дата обращения 23.04.2024).
7. Figma.com [Электронный ресурс] // Figma.com. URL: <https://www.figma.com/> (дата обращения 23.04.2024).
8. OpenCV // OpenCV – Open Computer Vision Library. URL: <https://opencv.org/> (дата обращения: 15.05.2024).

References

1. Environmental monitoring: An educational and methodological manual. Ed. 3rd, ispr. and add. / Edited by T. Ya. Ashikhmina. M.: Academic Project, 2006. 416 p. ("Gaudeamus"). (In Russ.).
2. Patent No.2620555 Russian Federation, IPC G01N 33/24 (2006.01). Method for assessing the biological activity and toxicity of soils and man-made soils: No.2016113050: application 05.04.2016: publ. 26.05.2017 / Eremchenko O. Z., Mitrakova N. V. 15 p. (In Russ.).
3. Mobile application for determining the area of leaves // NEFU Botanical Garden. URL: <https://garden-nefu.ru/blog/front/view?id=7> (date of access: 05/01/2024). (In Russ.).
4. Petiole Sheet area // Apkpure. URL: <https://apkpure.net/ru/petiole-plant-leaf-area-meter/com.petioleapp.petiole> (date of access: 05/01/2024). (In Russ.).
5. Computer vision in 2024: main tasks and directions // Habr. URL: <https://habr.com/ru/companies/otus/articles/810207> (date of access: 05/20/2023). (In Russ.).
6. Diagrams.net [Electronic resource] // Draw.io. URL: <https://www.drawio.com> (date of access 04/23/2024).
7. Figma.com [Electronic resource] // Figma.com. URL: <https://www.figma.com> (date of access 04/23/2024).
8. OpenCV // OpenCV – Open Computer Vision Library. URL: <https://opencv.org/> (date of access: 05/15/2024).

**DESIGNING THE MODULE «MEASURING STRESS RESISTANCE
IN MICRO-GREENERY» FOR THE IS «PRACTICE AT THE FACULTY OF BIOLOGY»**

Akhmatova Ulyana A.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, uahmatova@mail.ru

Knutova Nataliya S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, talieknutova@gmail.com

The design of the module for the information system "Practice at the Faculty of Biology" is being considered. It is assumed that the module will solve the problem of measuring the lengths of micro-greens stems, as well as calculating phytotoxicity – based on these data, biologists will be able to draw conclusions for further research. The paper analyzes the existing algorithms and automation systems for measurements and calculations, defines the requirements for the designed system and selects software tools for system development. The article contains a description of the logic of the IC module, a data model, as well as an example of a possible user interface.

Keywords: micro-greenery stress resistance, soil phytotoxicity, computer vision.

ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ СОПРОВОЖДЕНИЯ РЕТЕЙЛА

Базеев Александр Константинович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, bazeev03@mail.ru

Аверин Сергей Игоревич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, averin-s-i@yandex.ru

Все мы так или иначе сталкиваемся с таким понятием как ретейл. Ретейл – это продажа товаров или услуг небольшим количеством. Но современные магазины часто представляют из себя сложно организованные помещения с запутанным планом размещения товаров. По таким помещениям приходится долго ходить, чтобы понять и запомнить где лежит конкретный продукт. Следовательно, возникает вопрос об ускорение процесса покупок путем его автоматизации. Для решения данной проблемы была спроектирована информационная система сопровождения ретейла. В ходе работы была изучена предметная область и аналогичные системы, определены требования к информационной системе, выбрано программное обеспечение и язык программирования для дальнейшей реализации системы. Для проектирования системы были построены UML диаграммы и ER модель. А также был создан макет пользовательского интерфейса.

Ключевые слова: автоматизация, сопровождение, информационная система, ретейл, проектирование.

Все мы так или иначе сталкиваемся с таким понятием как ретейл. Ретейл – это продажа товаров или услуг небольшим количеством. Иными словами, вы можете столкнуться с этим термином при походе в продуктовый магазин. Зайдя в магазин, маркетологи начинают предлагать вам различные выгодные предложения и акции, которые сбивают с толку и навязывают приобретение товара. Также в магазинах товары первой необходимости выстраивают подальше от входа, чтобы вы обратили внимание и на другие товары. Данные уловки отнимают у посетителей силы и время, заставляя буквально ходить кругами по магазину. Из всего этого возникает вопрос об ускорение этого процесса путем его автоматизации. Актуальность работы состоит в том, чтобы спроектировать информационную систему, которая позволит автоматизировать процесс построения маршрута по конкретному списку товаров.

При создании системы нужно учитывать расположение товаров, отношение товаров к конкретным группам, цены товаров в ближайших магазинах, обновления характеристик товаров.

Перед разработкой системы были изучены аналоги. Рассмотрев возможности M.GO, ShopsNavigator [1], Indoor, был сделан вывод, что они не подходят к заявленной теме курсовой, в таких аспектах как: удобство, доступность функционала, стоимость. Наиболее близкой является система indoor от компании «Навигационные решения» [2]. Но в ней не хватает анализа цен таких же товаров в других магазинах и построения маршрута по списку товаров.

Для проектирования системы было необходимо выбрать подходящие для этого инструменты. Для создания UML-диаграмм, было выбрано приложение Diagrams.net [3]. Для создания макета пользовательского интерфейса была выбрана Figma [5]. В качестве языков программирования были выбраны Python и JavaScript. В качестве среды разработки была выбрана IDLE [7]. Для управления базой данных был выбран PostgreSQL. [4].

На диаграмме прецедентов (см. рис. 1) представлены возможности пользователя и администратора. Администратором в данной системе является человек, который работает в магазине.

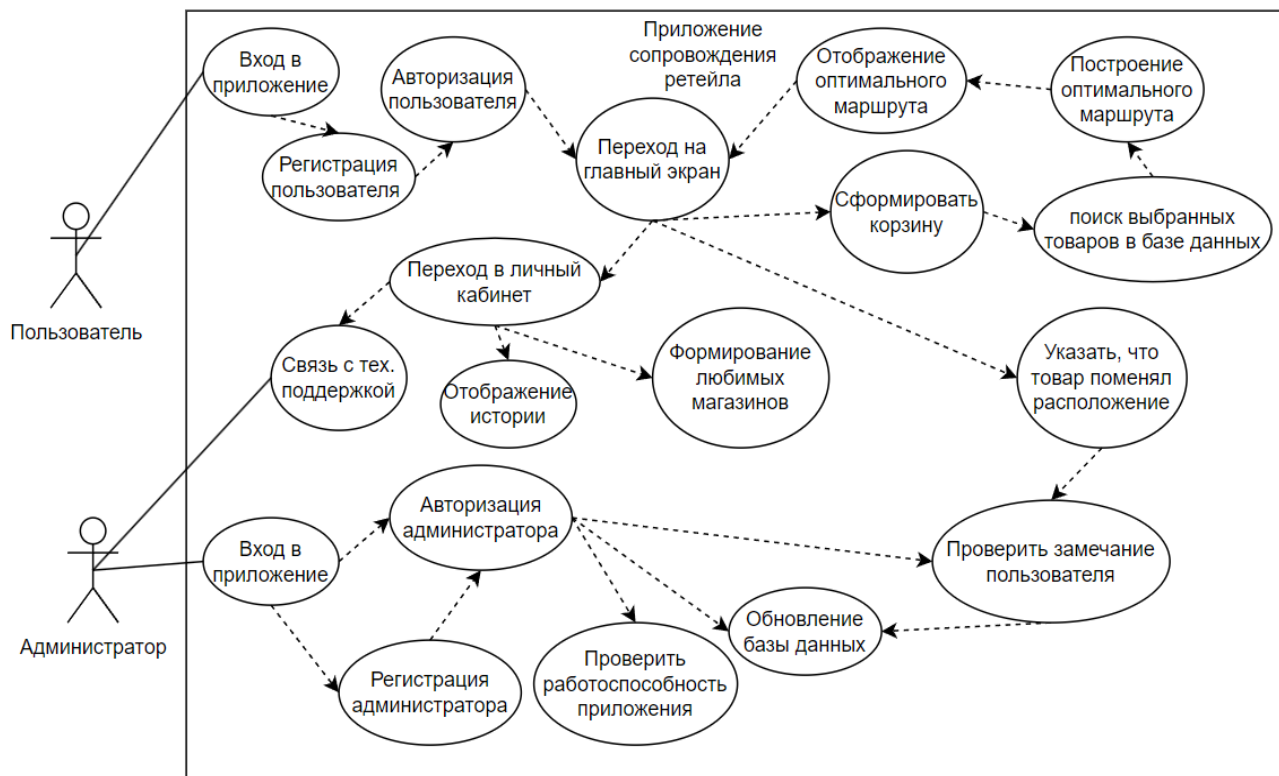


Рис. 1. Диаграмма прецедентов для пользователя и администратора

Также была разработана отдельная диаграмма прецедентов (см. рис. 2) для администратора информационной системы.

Чтобы более детально рассмотреть процесс создания маршрута для покупателя, была спроектирована диаграмма состояний. Она проиллюстрирована на рисунке ниже (см. рис. 3).

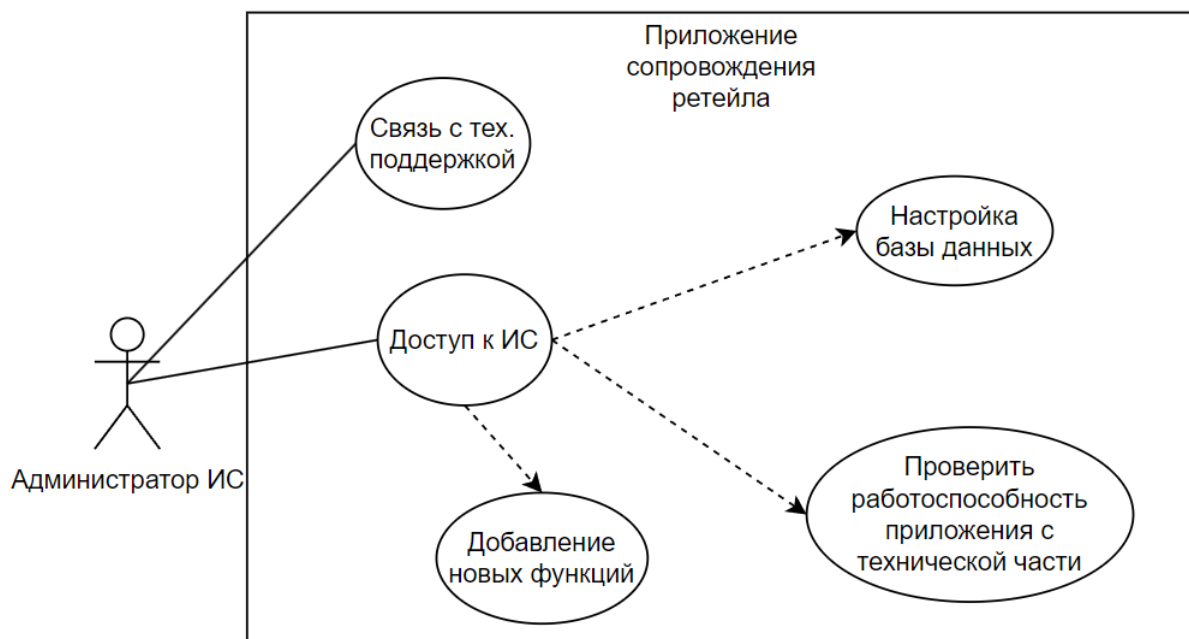


Рис. 2. Диаграмма прецедентов для администратора системы

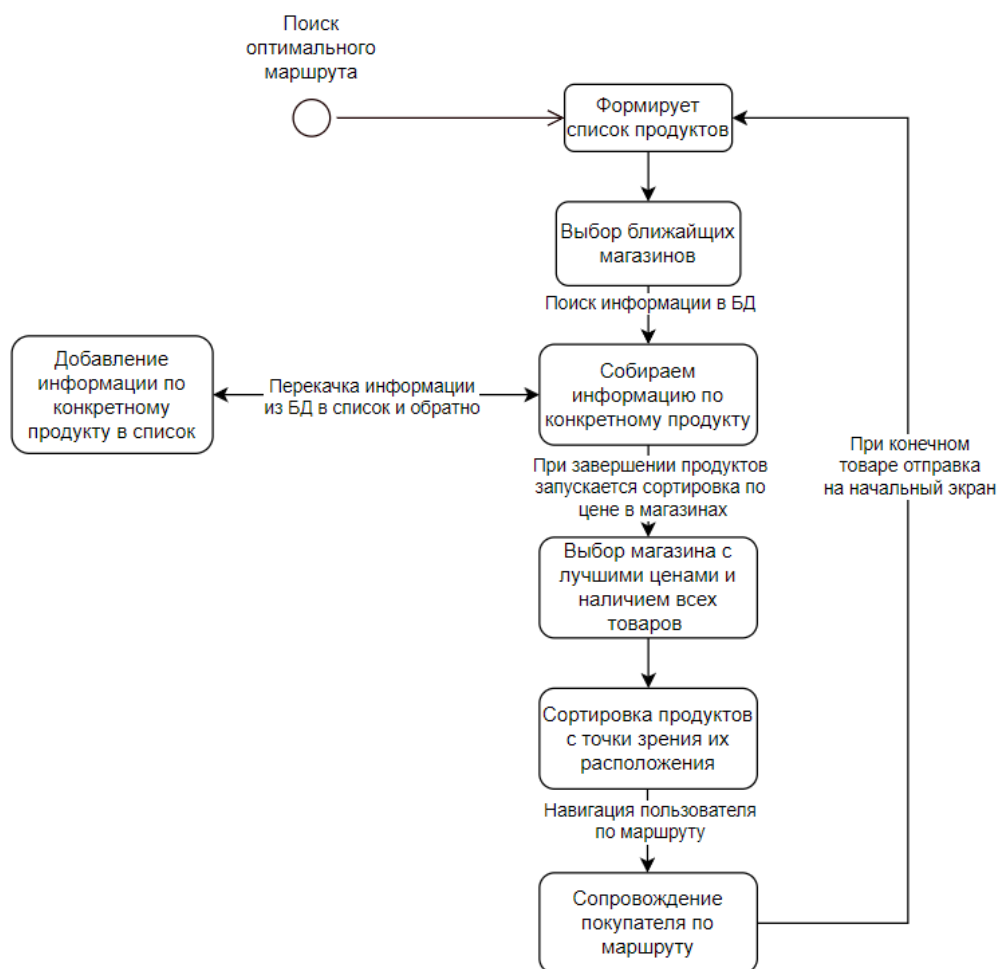


Рис. 3. Диаграмма состояний

На рисунке 4 изображена схема базы данных системы.



Рис. 4. Схема базы данных

В ходе работы был спроектирован макет пользовательского интерфейса. При первом запуске приложения пользователю открывается окно регистрации. Если же пользователь уже зарегистрирован он может перейти в окно авторизации, где он может ввести свои данные. В окне, представленном на рис. 5, пользователю открывается главный экран приложения.

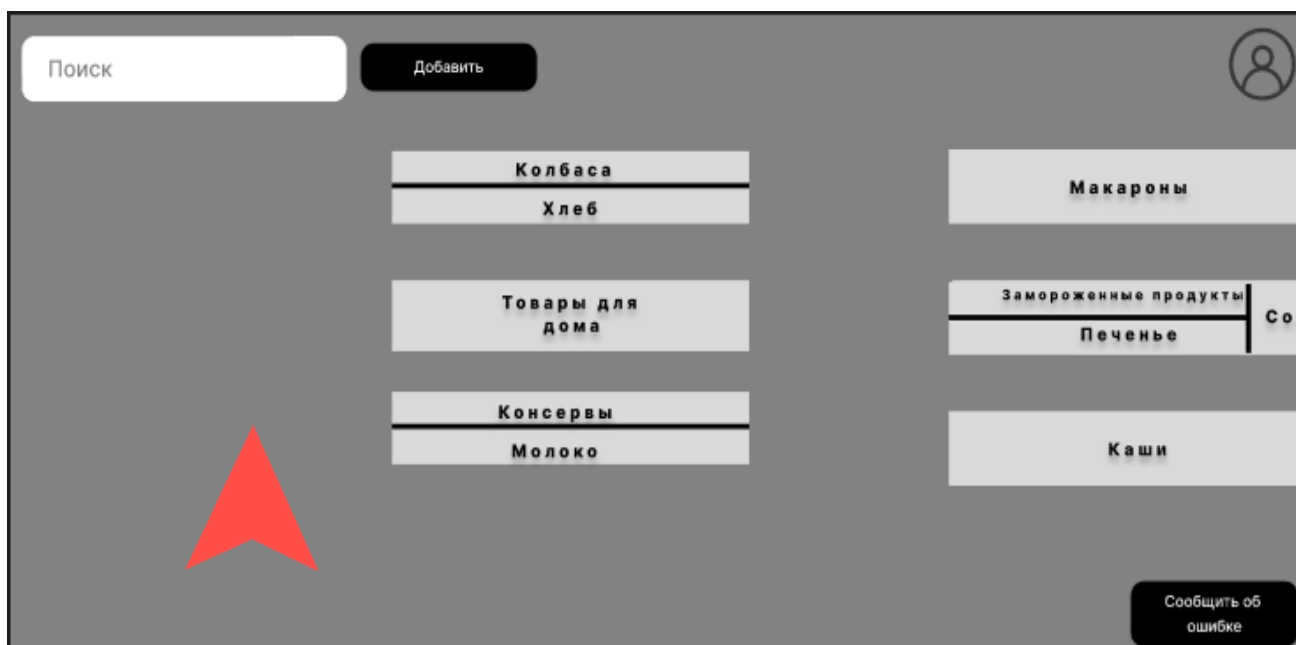


Рис. 5. Главный экран

На главном экране пользователь может добавить продукты в список и по этим продуктам построится маршрут по магазину (см. рис. 6).

Справа внизу мы можем наблюдать кнопку «Сообщить об ошибке», нажав на которую выведется экран с заполнением запроса пользователя (см. рис. 7).

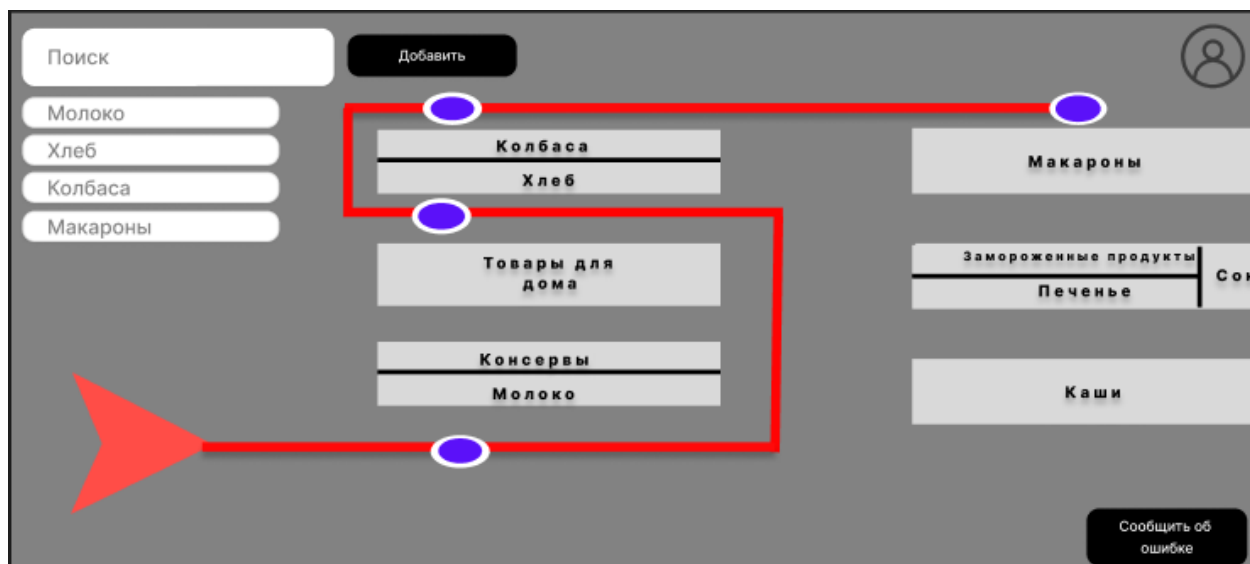


Рис. 6. Пример маршрута по магазину

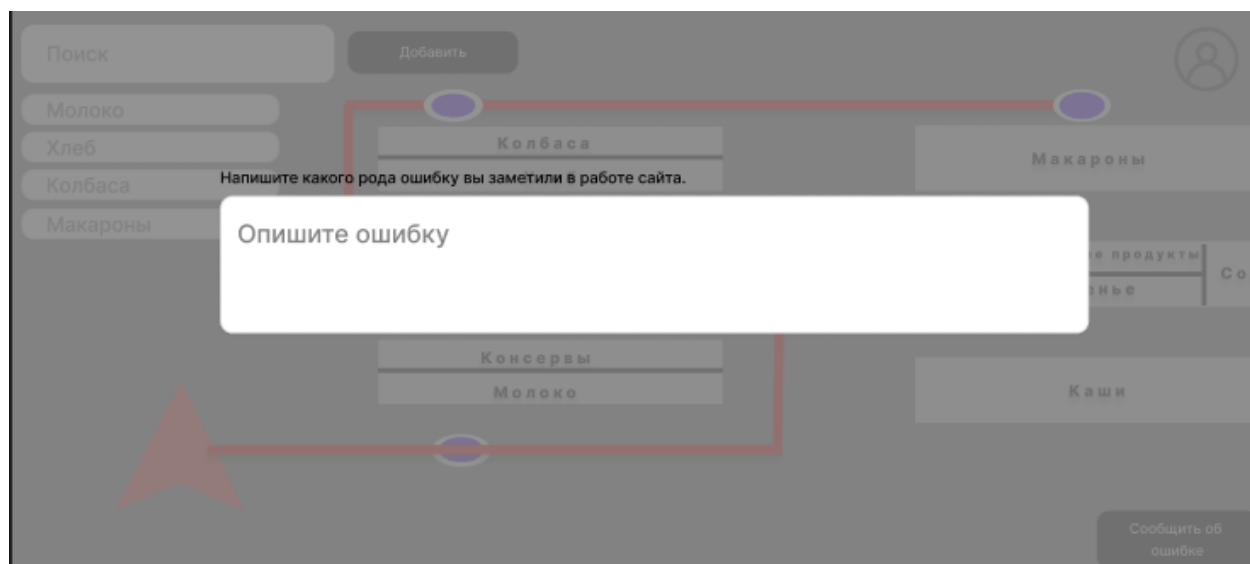


Рис. 7. Запрос пользователя

В правом верхнем углу расположилась иконка пользователя, при ее нажатии мы попадаем на экран с личным кабинетом пользователя

В результате работы были рассмотрены ключевые аспекты предметной области, было выбрано программное обеспечение и язык программирования для проектирования и дальнейшей разработки системы, а также система была спроектирована: построены UML диаграммы прецедентов и развёртывания, ER диаграмма и создан макет пользовательского интерфейса.

Библиографический список

1. ShopsNavigator [Электронный ресурс]. URL: <http://shopsnavigator.ru/#main-header> (дата обращения 05.05.2024).
2. Навигационные решения [Электронный ресурс]. URL: <https://nvgn.ru/> (дата обращения 05.05.2024).
3. Diagrams.net [Электронный ресурс]. URL: <https://app.diagrams.net/> (дата обращения 10.05.2024).
4. PostgreSQL [Электронный ресурс]. URL: <https://www.postgresql.org/> (дата обращения: 15.05.2023).
5. Figma [Электронный ресурс]. URL: <https://www.figma.com> (дата обращения: 15.05.2023).
6. Новиков Ф. А. Анализ и проектирование на UML: Учеб. пособие. СПб.: Университет ИТМО, 2007. 283 с.
7. IDLE [Электронный ресурс]. URL: <https://docs.python.org/3/library/idle.html> (дата обращения: 18.05.2024).

References

1. ShopsNavigator [Electronic resource]. URL: <http://shopsnavigator.ru/#main-header> (date of access: 05/05/2024).
2. Navigation solutions [Electronic resource]. URL: <https://nvgn.ru/> (date of access: 05/05/2024).
3. Diagrams.net [Electronic resource]. URL: <https://app.diagrams.net/> (date of access: 05/10/2024).
4. PostgreSQL [Electronic resource]. URL: <https://www.postgresql.org/> (date of access: 05/15/2023).
5. Figma [Electronic resource]. URL: <https://www.figma.com> (date of access: 05/15/2023).
6. Novikov F. A. Analysis and design in UML. A study guide. St. Petersburg: ITMO University, 2007. 283 p.
7. IDLE [Electronic resource]. URL: <https://docs.python.org/3/library/idle.html> (date of access: 05/18/2024).

DESIGNING AND DOCUMENTING AN INFORMATION SYSTEM FOR RETAIL SUPPORT

Bazeev Alexander K.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, bazeev03@mail.ru

Averin Sergey I.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, averin-s-i@yandex.ru

We all come across such a concept as retail in one way or another. Retail is the sale of goods or services in small quantities. But modern stores are often complexly organized rooms with a confusing layout of goods. It takes a long time to walk through such rooms in order to understand and remember where a particular product lies. Therefore, the question arises about speeding up the shopping process by automating it. To solve this problem, an information system for retail support was designed. In the course of the work, the subject area and similar systems were studied, the requirements for the information system were determined, software and programming language were selected for further implementation of the system. UML diagrams and an ER model were built to design the system. A user interface layout was also created.

Keywords: automation, maintenance, information system, retail, design.

ИДЕНТИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ ОБЪЕКТОВ НА МИКРОИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ПОДХОДОВ ТЕОРИИ ПЕРКОЛЯЦИИ

Безматерных Александра Евгеньевна, Бузмакова Мария Михайловна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, mbuzmakova@psu.ru

В настоящей работе описан подход идентификации объектов на микроизображениях, основанный на методах теории перколяции. Рассмотрена модифицированная перколяционная задача узлов на простой квадратной решетке. Для реализации задачи разработаны алгоритмы предобработки изображения, выделения границ объектов на изображении, идентификации объектов, кластеризации объектов. Была написана программа для проведения эксперимента по идентификации объектов, которую можно использовать для разных типов микроизображений.

Ключевые слова: теория перколяции, идентификация и кластеризация объектов, распознавание изображений.

Идентификация и кластеризация объектов на изображениях – это процесс автоматического обнаружения, выделения и группировки объектов с помощью различных компьютерных технологий. Начиная от решения простых задач, таких как обнаружение объектов, до более сложных, таких как выявление структурных характеристик изображений, эта область стремится к постоянному развитию с целью автоматизации процессов, которые ранее требовали вмешательства человека. До сих пор не существует метода или алгоритма, который мог бы эффективно справляться с любой задачей идентификации объектов. В зависимости от особенностей изображений и объектов, представленных на них, применяются различные методы, которые могут обеспечить быстрые и качественные результаты для конкретных задач. Идентификация и кластеризация объектов на микроизображениях имеют свои уникальные особенности и сложности. Микроизображения представляют собой изображения, полученные с использованием микроскопической техники, и используются для анализа объектов, невидимых невооруженным глазом. Это могут быть клетки, бактерии, вирусы, частицы, кристаллы или другие микрообъекты. Такие изображения требуют высокой точности и разрешения, чтобы эффективно выделять и анализировать мельчайшие детали.

В данной работе предлагается модель процесса идентификации объектов на микроизображении с использованием подходов теории перколяции. Рассмотрена модифицированная перколяционная задача узлов на квадратной решетке. Занятым узлом считается пиксель,

принадлежащий некоторому объекту, свободный узел – это пиксель фона. Пиксели одного объекта объединяются в группы – кластеры. Модификация модели состоит в том, что в кластеры объединяются соседние занятые узлы не только по горизонтали и вертикали, но еще и по диагонали. Для модели разработан алгоритм маркировки занятых узлов и алгоритм распределения узлов по кластерам. Для полученных результатов реализована возможность проводить анализ параметров объектов, таких как их размеры, распределение, количество, минимальные и максимальные значения и др.

Перед идентификацией объектов не обойтись без предобработки изображений. Обработка изображения включает в себя приведение изображения к тому состоянию, в котором можно будет выделить необходимые признаки объектов. Одним из ключевых шагов этапа обработки является процесс бинаризации изображения. Бинаризация [1, 2] – это процесс превращения исходного цветного или состоящего из оттенков серого изображения, в изображение, состоящее только из двух цветов. Суть бинаризации состоит в том, чтобы отделить пиксели объектов от пикселей фона. В процессе бинаризации происходит пороговая обработка, в которой порог – это фиксированное значение, подобранное вручную или автоматически, с применением различных методов. Интенсивность каждого пикселя изображения сравнивается со значением порога и, в зависимости от того, больше она или меньше этого значения, пиксель маркируется единицей или нулем. Пиксели объекта маркируются 1 и окрашиваются в черный цвет. Остальные пиксели приравниваются к точкам фона, принимают маркировку 0 и окрашиваются в белый цвет. Набор исходных изображений представлял собой изображения различных объектов под микроскопом. На них есть четкое разделение на фон и объекты, поэтому для бинаризации был реализован метод Оцу [3]. Результатом применения данного алгоритма к исходному изображению (рис. 1) является бинарное изображение (рис. 2) и гистограмма яркости (рис. 3). Оптимальный посчитанный порог бинаризации для исходного изображения в примере равен 90.

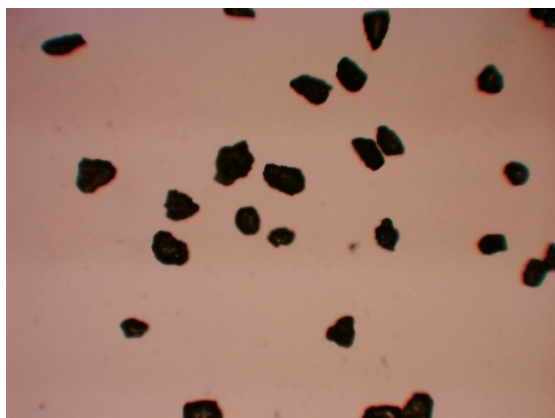


Рис. 1. Исходное изображение кварцевого песка



Рис. 2. Бинарное изображение кварцевого песка

Трудности на этом этапе могут возникать по нескольким причинам. Качество исходного изображения: большое количество шумов, неровное освещение, блики, слишком низкая контрастность. Характер гистограммы: неявные пики, несколько явных пиков.

Некоторые изображения могут показывать неочевидные результаты. На другом тестовом примере (изображение инвертировано и шумы удалены) представлены частицы металлического порошка под микроскопом (рис. 3.). Применение метода Оцу (рис. 4.) выдает оптимальный порог равный 46, что является неверным – это значение слишком низкое.

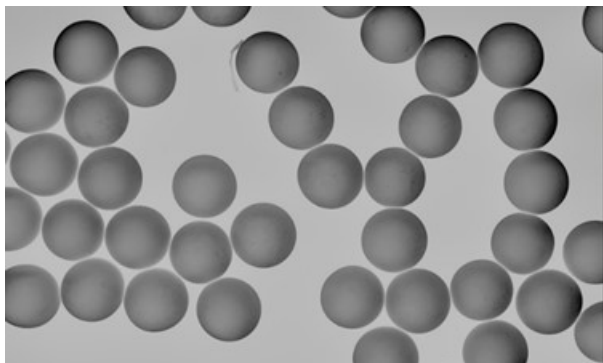


Рис. 3. Исходное изображение
металлического порошка



Рис. 4. Бинарное изображение
металлического порошка

Такой результат может возникать из-за неравномерного освещения на изображении. Некоторые участки объектов слишком темные, некоторые слишком светлые.

Исправить эту проблему можно применив метод Оцу локально. На первом тестовом примере (рис. 1) метод Оцу применялся глобально, т.е. один порог для всего изображения. Идея локального применения в том, чтобы разбить изображение на квадратные блоки и посчитать оптимальный порог для каждого блока отдельно.

После применения метода Оцу локально для блоков результат значительно улучшился (рис. 5).

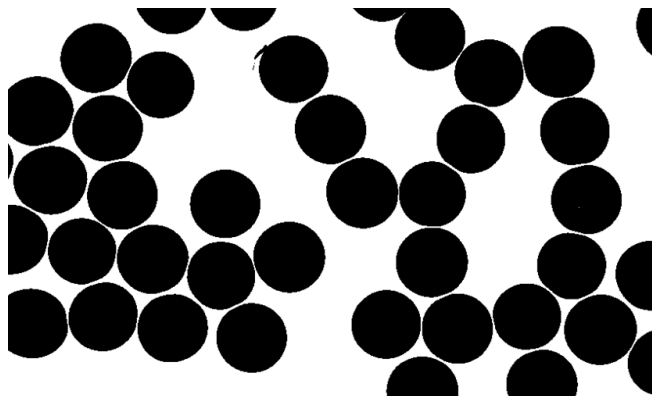


Рис. 5. Бинарное изображение металлического порошка,
полученное с использованием локальных порогов

Следующий этап –разделение «склеенных» объектов на изображении. Трудности на этом этапе могут возникнуть из-за специфики объектов: разные размеры, неравномерная форма и текстура объектов, низкое качество изображения. Для устранения проблемы «склеенных» объектов можно использовать два алгоритма, работающих с границами объектов. Первый из них основан на выделении границ изображений и состоит из следующих этапов.

1. Построение изображения, содержащего контуры объектов, и его бинаризация.
2. Увеличение толщины контуров.
3. Соединение обычного бинарного изображения и бинарного изображения контуров с утолщенными границами.

Такой алгоритм не сможет разъединить объекты, слишком близкие друг к другу, но сможет разделить те, которые слегка касаются.

Большинство методов выделения краев основывается на вычислении градиента [2] (анализ изменений яркости в определенном направлении). В таких алгоритмах происходит свертка изображения и градиентного оператора (его еще называют ядром или маской). Для каждого пикселя выбирается квадратная область вокруг него, размер которой зависит от применяемого метода. Операция свертки представляет собой поэлементное перемножение элементов маски с соответствующими элементами матрицы (текущей выбранной области изображения), а затем полученные произведения суммируются. Это значение становится новым значением для центрального пикселя.

В данной работе для выделения контуров объектов, к изображению был применен оператор Собеля [4] (рис. 6), а затем полученное изображение краев проходило пороговую обработку (рис. 7).

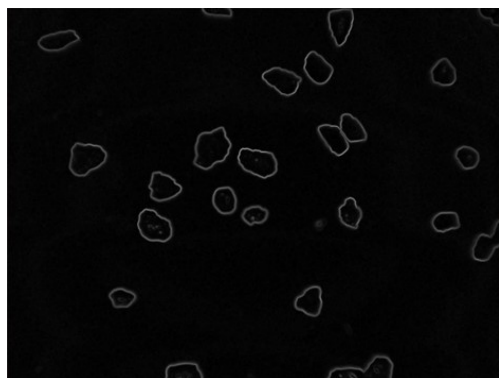


Рис. 6. Изображение выделенных краев

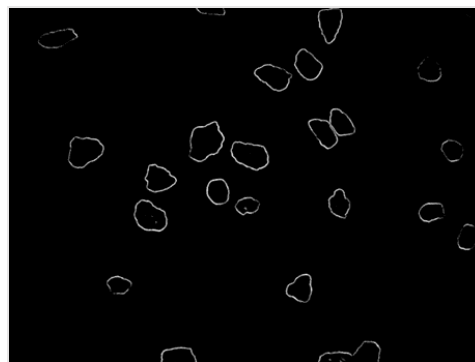


Рис. 7. Бинарное изображение выделенных краев

После выделения краев, к получившемуся изображению применяется функция утолщения границ. Происходит циклический проход по изображению, и к пикселям границ добавляются белые пиксели вокруг, что делает границу визуально толще.

Полученное изображение с толстыми границами сливается с исходным бинарным изображением. На получившемся изображении видно (рис. 8), что есть два объекта, которые отделились друг от друга, так как изначально у них не было слишком сильного касания. Но также есть элементы, которые остались склеенными, так как расположены слишком близко друг к другу. Задача разделения таких объектов может оказаться очень трудной и потребовать использования сложных методов локально.

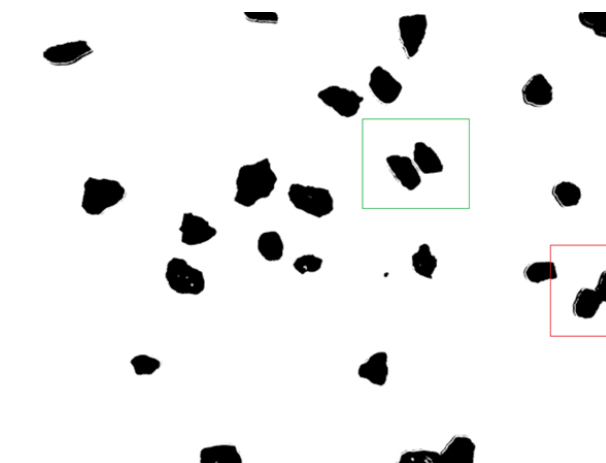


Рис. 8. Результат слияния двух изображений

Другой алгоритм реализован похожим образом. Для него используются две операции – эрозия и дилатация. Эрозия – это процесс уменьшения объектов на изображении, а дилатация – увеличения объектов. Применяя эти операции достаточное количество раз, можно добиться разделения объектов.

На рис. 9 показан результат после нескольких применений алгоритма эрозии и дилатации. Два склеенных объекта разъединились.

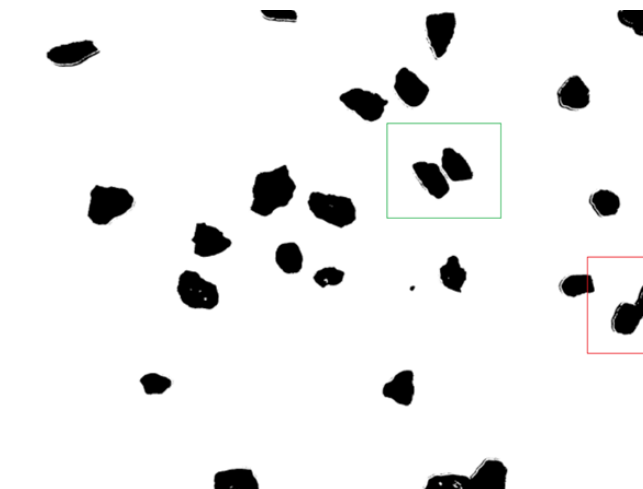


Рис. 9. Результат применения эрозии и дилатации

На основе предобработанного изображения строится модифицированная перколяционная задача узлов на простой квадратной решетке. Бинарное изображение переводится в двумерный массив единиц и нулей. Единица – пиксель закрашен, ноль – пиксель свободен (рис. 10 и 11). Для идентификации объектов реализован алгоритм распределения узлов по кластерам (Хошена-Копельмана, [5]). В рамках модели под кластером понимается объект. Результатом алгоритма Хошена-Копельмана получается массив, содержащий узлы, разбитые на кластеры. Результат для алгоритма Хошена-Копельмана – разбиение узлов на отдельные кластеры – представлен на рис. 12. В данном примере узлы разбиваются на 14 кластеров. При работе с со сложными изображениями большего размера будут получаться массивы больших размеров.

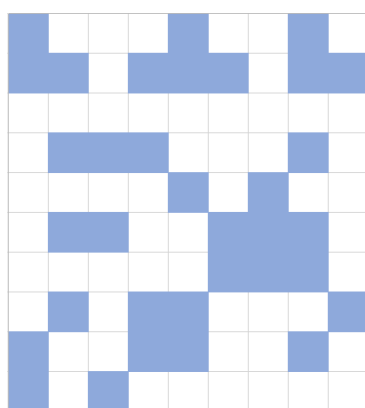


Рис. 10. Макет перколяционной решетки

1	0	0	0	1	0	0	1	0
1	1	0	1	1	1	0	1	1
0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0
0	0	0	0	1	0	1	0	0
0	1	1	0	0	1	1	1	0
0	0	0	0	0	1	1	1	0
0	1	0	1	1	0	0	0	1
1	0	0	1	1	0	0	1	0
1	0	1	0	0	0	0	0	0

Рис. 11. Двумерный массив нулей и единиц

1	0	0	0	2	0	0	3	0
1	1	0	2	2	2	0	3	3
0	0	0	0	0	0	0	0	0
0	4	4	4	0	0	0	5	0
0	0	0	0	6	0	7	0	0
0	8	8	0	0	7	7	7	0
0	0	0	0	0	7	7	7	0
0	9	0	10	10	0	0	0	11
12	0	0	10	10	0	0	13	0
12	0	14	0	0	0	0	0	0

Рис. 12. Результат разбиения узлов на отдельные кластеры

Для проведения эксперимента был разработан программный комплекс. Был проведен эксперимент на примере различных тестовых микроизображений. В программу загружалось исходное изображение, проводилась обработка. В результате были получены промаркированное изображение и вычисленные значения характеристик объектов, построена диаграмма распределения объектов по размерам.

Тестовый пример №1 представлен на рис. 13. Это изображение кварцевого песка под микроскопом. Результаты маркировки объектов на примере тестового изображения приведены на рис. 14. Характеристики объектов приведены в таблице 1. Исходное изображение удалось разбить на 24 кластера (объекта). Высокое значение размаха указывает на значительную вариативность размеров объектов. Медианный и средний размер близки по значению, что говорит о том, что размеры распределены симметрично относительно среднего значения. Коэффициент асимметрии указывает на наличие нескольких крупных объектов.

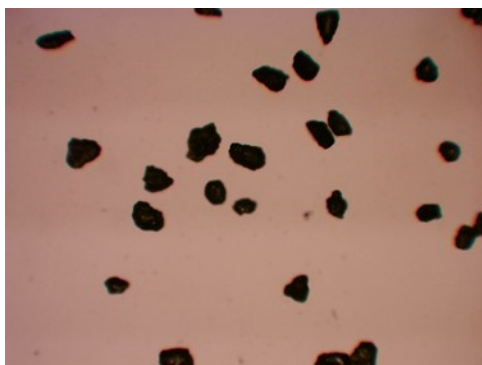


Рис. 13. Тестовое изображение №1



Рис. 14. Результат маркировки объектов

Таблица 1

Характеристики объектов тестового изображения №1

Характеристика	Значение	Номер кластера
Максимальный размер	26819	23
Минимальный размер	5281	1
Характеристика	Значение	
Количество объектов	24	
Общая площадь	349623	
Средний размер	14567	
Медианный размер	13938	
Стандартное отклонение	5579	
Коэффициент вариации	0,383	
Размах	21538	
Коэффициент асимметрии	0,624	

Тестовый пример №2 представлен на рис. 15. Это изображение аденоассоциированного вируса под микроскопом. Результаты маркировки объектов на примере тестового изображения приведены на рис. 16. Характеристики объектов приведены в таблице 2. Исходное изображение удалось разбить на 195 кластеров (объектов). Медианный и средний размер близки по значению. Стандартное отклонение и коэффициент вариации указывают на низкую изменчивость размеров объектов. Были отфильтрованы крупные объекты, являющиеся шумами.

Тестовый пример №3 представлен на рис. 17. Это изображение клеток крови под микроскопом. Результаты маркировки объектов на примере тестового изображения приведены на рис. 18.

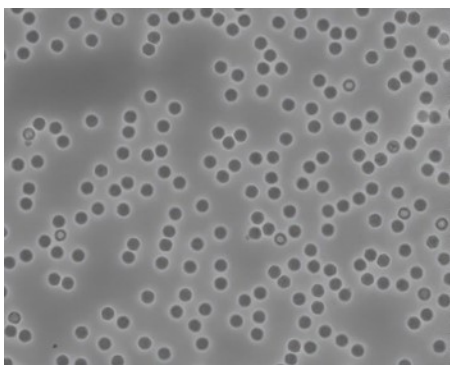


Рис. 15. Тестовое изображение №2

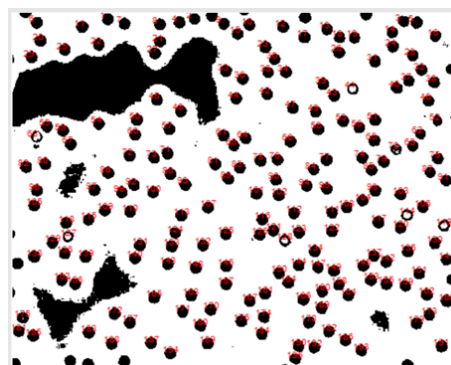


Рис. 16. Результат маркировки объектов

Таблица 2

Характеристики объектов тестового изображения №2

Характеристика	Значение	Номер кластера
Максимальный размер	143	117
Минимальный размер	52	63
Характеристика	Значение	
Количество объектов	195	
Общая площадь	24664	
Средний размер	126	
Медианный размер	128	
Стандартное отклонение	10,698	
Коэффициент вариации	0,085	
Размах	91	
Коэффициент асимметрии	-3,622	

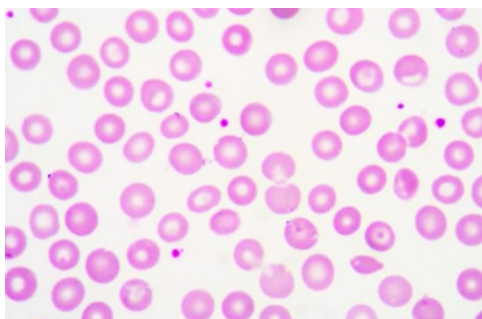


Рис. 17. Тестовое изображение №3

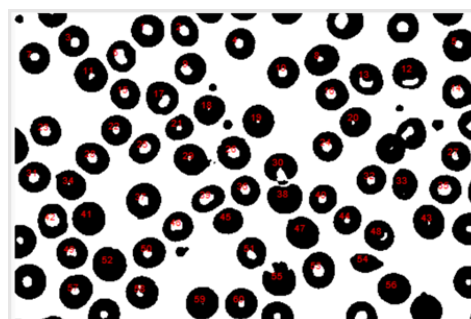


Рис. 18. Результат маркировки объектов

Характеристики объектов приведены в таблице 3. Исходное изображение удалось разбить на 60 кластеров (объектов).

Характеристики объектов тестового изображения №3

Характеристика	Значение	Номер кластера
Максимальный размер	1029	52
Минимальный размер	511	54
Характеристика		Значение
Количество объектов		60
Общая площадь		47881
Средний размер		798
Медианный размер		791
Стандартное отклонение		112,094
Коэффициент вариации		0,140
Размах		518
Коэффициент асимметрии		-0,545

В данном тестовом примере стоит принимать во внимание специфику объектов на изображении. Эритроциты обладают специфической формой – двояковогнутый диск, поэтому их центральная часть более тонкая и светлая. Из-за этого, некоторые объекты на изображении могут иметь пробелы внутри. Соответственно, на самом деле, характеристики могут немного отличаться.

Тестовый пример №5 представлен на рис. 19. Это изображение клеток растения под микроскопом. Результаты маркировки объектов на примере тестового изображения приведены на рис. 20.

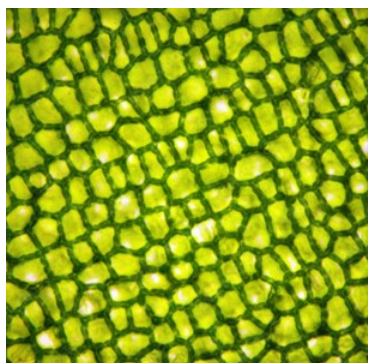


Рис. 19. Тестовое изображение №4

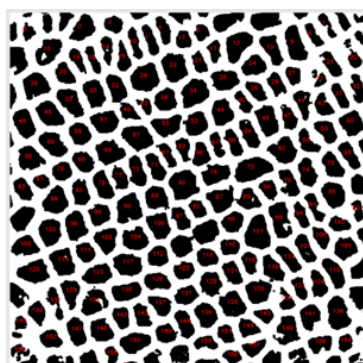


Рис. 20. Результат маркировки объектов

Характеристики объектов приведены в таблице 4. Исходное изображение удалось разбить на 160 кластеров (объектов).

Характеристики объектов тестового изображения №4

Характеристика	Значение	Номер кластера
Максимальный размер	1737	73
Минимальный размер	269	31
Характеристика	Значение	
Количество объектов	160	
Общая площадь	145517	
Средний размер	909	
Медианный размер	791	
Стандартное отклонение	312,276	
Коэффициент вариации	0,343	
Размах	1468	
Коэффициент асимметрии	0,378	

Стандартное отклонение равное и коэффициент вариации равный указывают на то, что размеры объектов имеют большую изменчивость, т.е. значительно разные размеры. Для подсчета характеристик в данном и некоторых других тестовых примерах не учитывались объекты, пересекающие границы изображения.

Тестовый пример №5 представлен на рис. 21. Это изображение металлического порошка под микроскопом. Результаты маркировки объектов на примере тестового изображения приведены на рис. 22.

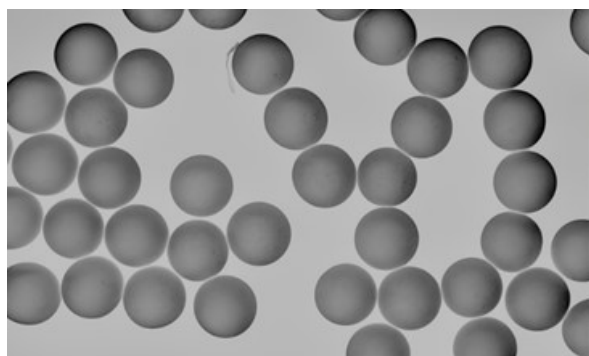


Рис. 21. Тестовое изображение №5

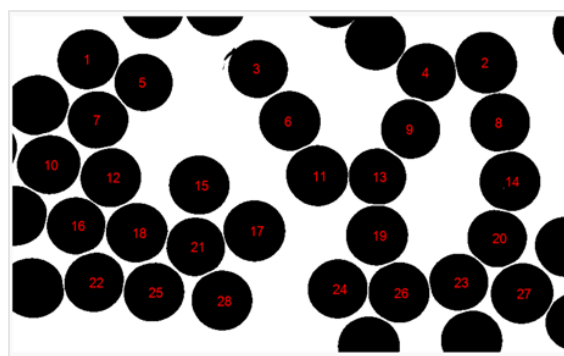


Рис. 22. Результат маркировки объектов

Характеристики объектов приведены в таблице 5. Исходное изображение удалось разбить на 28 кластеров (объектов).

Характеристики объектов тестового изображения №5

Характеристика	Значение	Номер кластера
Максимальный размер	6916	2
Минимальный размер	5861	13
Характеристика	Значение	
Количество объектов	28	
Общая площадь	180155	
Средний размер	6434	
Медианный размер	6442	
Стандартное отклонение	275,082	
Коэффициент вариации	0,042	
Размах	1055	
Коэффициент асимметрии	-0,189	

Значения стандартного отклонения, коэффициента вариации и коэффициента асимметрии указывают на то, что распределение объектов практически симметрично, а размеры объектов очень близки друг к другу по значению.

Результаты работы можно считать успешными. Полученные результаты показали, что методы теории перколяции можно успешно применять в практике для идентификации и изучения свойств объектов на микроизображениях.

Библиографический список

1. Бобков А. Системы распознавания образов: учеб. пособие. М.: Изд-во МГТУ им. Н. Э. Баумана, 2018. 187 с.
2. Гонсалес Р. С., Вудс Р. Е. Цифровая обработка изображений. 3-е изд. М.: Техносфера, 2012. 1104 с.
3. N. Otsu. 9. 1979. Vol. 9. P. 62–66.
4. Фисенко В. Т., Фисенко Т. Ю. Компьютерная обработка и распознавание изображений. СПб.:СПбГУ ИТМО, 2008. 192 с.
5. Hoshen J. and Kopelman R. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. Phys. Rev. B. 1976. 1 (14):3438-3445.

References

1. Bobkov A. Sistemy raspoznavaniya obrazov: uchebnoe posobie. M.: izdatel'stvo MGTU im. N. E. Baumana, 2018. 187 p.
2. Rafael S. Gonsales, Richard E. Vuds Cifrovaya obrabotka izobrazhenij. 3-e izd. M.: Tekhnosfera, 2012. 1104 p.
3. N. Otsu. 9. 1979. Vol. 9. Pp. 62–66.
4. Fisenko V. T., Fisenko T. YU. Komp'yuternaya obrabotka i raspoznavanie izobrazhenij. SPb.: SPbGU ITMO, 2008. 192 p.
5. Hoshen J. and Kopelman R. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. Phys. Rev. B. 1976.1(14):3438-3445.

OBJECT IDENTIFICATION AND CLUSTERING ON MICRO-IMAGES USING PERCOLATION THEORY

Bezmaternyh Alexandra E., Buzmakova Marya M.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, mbuzmakova@psu.ru

This paper describes an approach to object identification in micro-images based on percolation theory methods. The modified percolation model of nodes on a simple square lattice is considered. To implement the problem, algorithms for image preprocessing, object boundary extraction in the image, object identification, and object clustering have been developed. A program was written for the experiment on object identification, which can be used for different types of micro-images.

Keywords: percolation theory, object identification and clustering, image recognition.

ПРОЕКТИРОВАНИЕ СИСТЕМЫ КЛАССИФИКАЦИИ ОДЕЖДЫ НА ФОТОГРАФИЯХ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Вершинин Никита Александрович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, nikita.vershinin2003@mail.ru

Статья посвящена проектированию системы классификации одежды на фотографиях с использованием сверточных нейронных сетей. Целью данной работы является проектирование системы классификации одежды на фотографии, а также разработка пользовательского веб-интерфейса для взаимодействия с моделью. Объектом исследования выступает процесс автоматической классификации изображений. Предметом исследования является анализ и сравнение методов машинного обучения, применяемых для классификации изображений одежды, с акцентом на использование сверточных нейронных сетей. Были разработаны требования от системы и вызовы, с которыми можно будет столкнуться. В статье рассмотрены модели, используемые в качестве основы для нейронной сети, параметры для настройки данной сети, подходы для ее обучения, а также проектирование веб-сайта. Выбраны инструменты для проектирования пользовательского интерфейса, а также ресурсы для построения UML-диаграмм. Разработан макет пользовательского интерфейса, а также дополнена и настроена модель.

Ключевые слова: нейронная сеть, изображения, классификация, проектирование системы.

Введение

В современном мире, где информационные технологии проникают в самые разные аспекты нашей жизни, возникает всё больше возможностей для применения машинного обучения в различных отраслях. Одним из таких применений является автоматизация процессов в индустрии моды, где особое место занимает задача классификации одежды на фотографиях. Эта задача становится особенно актуальной в контексте развития онлайн-торговли и необходимости улучшения пользовательского опыта, оптимизации инвентаризации и предложить персонализированные рекомендации.

Постановка задачи

Основной целью данной работы является разработка системы, способной автоматически классифицировать одежду на фотографиях с высокой точностью и эффективностью. Для достижения этой цели были сформулированы следующие задачи:

- анализ существующих методов и технологий;
- определение способов обработки данных;

- проектирование архитектуры модели сверточной нейронной сети;
- проектирование веб-сайта и способа взаимодействия с системой.

Реализация этих задач позволит спроектировать систему, которая не только будет высокоэффективной в задаче классификации одежды на фотографиях, но и обеспечит удобный пользовательский опыт через веб-интерфейс.

Методы

В качестве анализа изображений будет использоваться свёрточная нейронная сеть [1, 2]. Сверточная нейронная сеть (CNN) – это одна из моделей нейронной сети для глубокого обучения, используемая в задачах с обработкой данных, представленной в виде сетчатой структуры. Они очень эффективно используются в задачах компьютерного зрения.

Трансферное обучение представляет собой мощный подход к обучению моделей глубокого обучения, который становится особенно актуальным в задачах классификации изображений, включая классификацию одежды. Этот метод позволяет использовать знания, полученные из одной задачи, для улучшения производительности или ускорения обучения в другой, схожей задаче. В контексте классификации одежды трансферное обучение может существенно сократить время и ресурсы, необходимые для разработки эффективных моделей.

При разработке системы необходимо учитывать различные требования от системы:

- Высокая точность классификации. Одним из главных требований является высокая точность классификации. Система должна быть способна точно определять тип одежды на изображениях с высоким уровнем достоверности. Это требование особенно важно для приложений, где пользователи полагаются на автоматизированный анализ изображений для принятия решений, например, при онлайн-шопинге.
- Масштабируемость. Модель должна работать с большими объемами данных и разнообразными классами одежды.
- Эффективность. Разработанная система должна обеспечивать быструю обработку изображений и оперативную их классификацию.
- Робастность к различным условиям. Модель должна быть устойчива к различным условиям по типу освещения, ракурсам съемки, а также к наличию различных фонов на фотографиях.
- Удобный интерфейс для пользователя. Система должна предоставлять удобный и интуитивно понятный интерфейс для ее использования. Он должен содержать визуальную обратную связь о процессе загрузки и классификации изображения, а также результаты данного процесса.
- Поддержка форматов JPEG и PNG, отклоняя другие форматы.
- Возможность дообучения модели на новых данных без переобучения на старых.

Рассмотрены различные модели в качестве основных для будущей нейронной сети. В контексте задачи классификации одежды EfficientNetV2B3 может стать лучшим решением для выбора ее как фундамента для взятия наиболее важных признаков с изображения благодаря её способности масштабирования и эффективности. Это особенно важно в индустрии моды, где изображения могут варьироваться по стилю и детализации. EfficientNetV2B3 позволит настроить глубину и ширину сети в соответствии с конкретными требованиями и ресурсами, доступными для обработки данных.

Для переобучения выбранной модели будет использоваться датасет DeepFashion1 [3] который является одним из наиболее масштабных и разнообразных наборов данных в области моды. Он включает в себя 46 классов одежды в общей сумме на 300,000 изображений в разных позах человека и с разным фоном позади него.

Обработка данных будет включать в себя следующие аспекты:

- Предварительная обработка: Изображения будут предварительно обработаны для обеспечения единообразия входных данных. В обработку будет включаться преобразование цветового пространства, нормализация пикселей, а также масштабирование изображений до одного размера, в данном случае будет использоваться размер 224×224 пикселя, так как такой формат стал основой в работе с изображениями в нейронных сетях из-за оптимального соотношения производительности и количества извлекаемых признаков с изображения.

- Разделение данных: Датасет будет разделен на такие наборы, как обучающий, валидационный и тестовый. Разделение в виде обучающего и валидационного наборов поможет следить за процессом обучения модели в конце каждой эпохи, в виде выведения оценки модели после текущей эпохи на валидационных данных. Тестовый же набор позволит оценить переобученную модель после процесса обучения.

Интеграция с моделью:

- Аугментация данных: перед подачей в модель, изображение будет аугментировано, а именно будут использоваться случайные повороты, небольшое изменение масштаба, горизонтальное отражение изображения.

- Подготовка пакетов данных: передача данных в модель в виде пакетов, а именно несколько изображений за раз, поможет эффективно использовать ресурсы и ускорить время обучения.

После полного обучения модели стоит проверить ее качество на тестовом наборе. Для такой проверки будет использоваться матрица ошибок (confusion_matrix). Она позволяет вывести производительность модели по каждому классу. Матрица показывает сколько было правильно определено для класса и сколько неправильных определено из каждого класса к другому классу.

Также будет использована ROC-кривая, обеспечивающая графическое представление эффективности классификации при различных порогах решения. ROC-кривая показывает соотношение между долей истинно положительных результатов (True Positive Rate, TPR) и долей ложно положительных результатов (False Positive Rate, FPR) по всем возможным порогам классификации. Это позволяет оценить, насколько хорошо модель различает классы. Площадь под ROC-кривой используется как мера обобщающей способности модели, где значение 1.0 представляет собой идеальное предсказание, а значение 0.5 – не лучше случайного угадывания.

Результаты

Взаимодействие с моделью будет происходить через веб-сайт. Была спроектирована диаграмма прецедентов. На Рис изображен актер «Пользователь». Он может загрузить изображения и получить его классификацию.

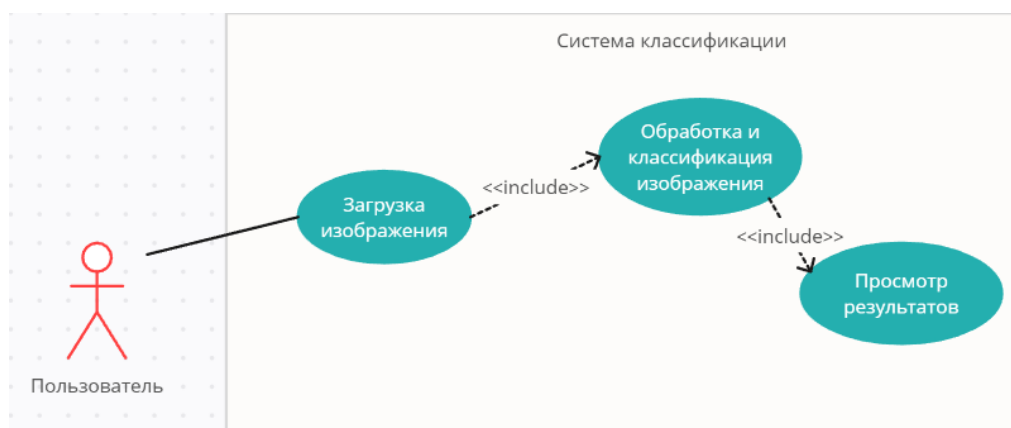


Рис. 1. Диаграмма прецедентов для пользователя

На рисунке 1 изображен актер «Администратор», который может следить за состоянием сайта, дополнять существующую модель, а также дообучать ее на новых данных.

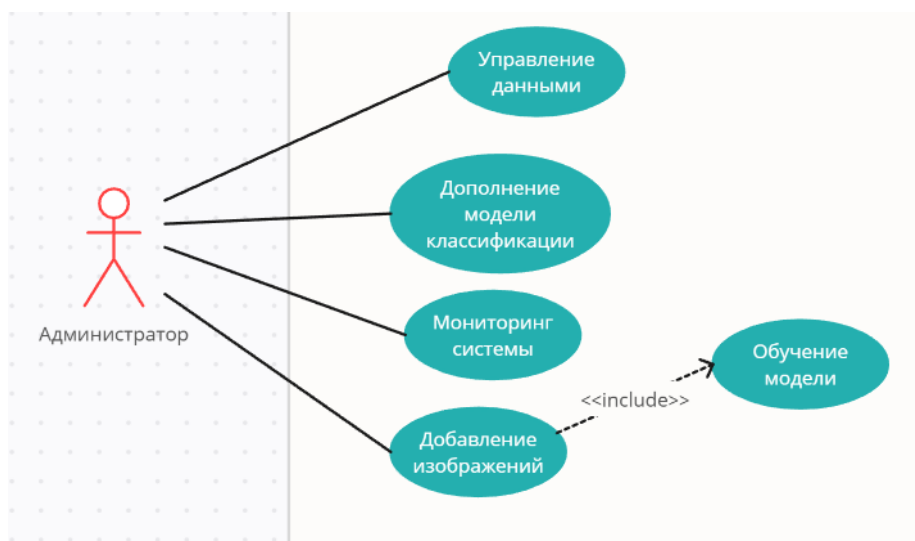


Рис. 1. Диаграмма прецедентов для администратора

При подключении модели будет установлен параметр «include_top» на значение «false», что позволит отключить верхние слои, которые были разработаны под конкретную задачу. Это позволит использовать предварительно обученные сверточные блоки добавляя при этом собственные слои для собственной задачи.

Дополнительные слои:

- GlobalAveragePooling2D layer;
- Dense layer (1024 нейронов, функция активации - ReLu, L2 регуляризация = 0.001);
- BatchNormalization layer;
- Dropout layer (коэффициент = 0.3);
- Dense layer (512 нейронов, функция активации - ReLu, L2 регуляризация = 0.001);
- BatchNormalization layer;
- Dropout layer (коэффициент = 0.5);
- Dense layer (256 нейронов, функция активации - ReLu, L2 регуляризация = 0.001);
- BatchNormalization layer;
- Dense layer (46 нейронов, функция активации – Softmax.

Также у модели будут заморожены нижние слои, чтобы они не обучались заново и будет присутствовать переобучение 20% слоев.

На рисунке 2 изображен главный экран, на котором можно перейти к загрузке изображения.

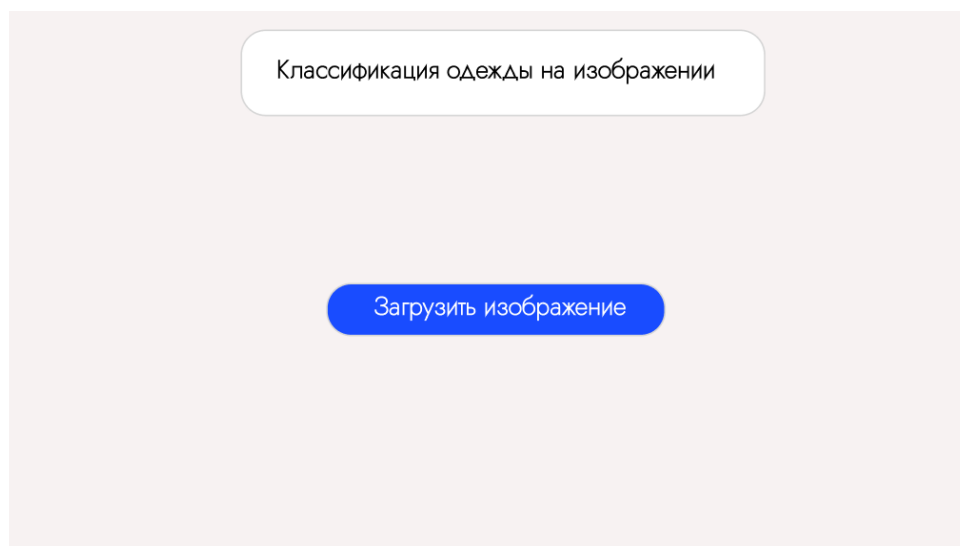


Рис. 2. Главный страница

После нажатия будет появляться диалоговое окно для загрузки изображения с устройства, оно представляет простое системное окно для выбора изображения.

После загрузки изображения будет переход на новое окно с отображением изображения и выбором действий (рис. 3).

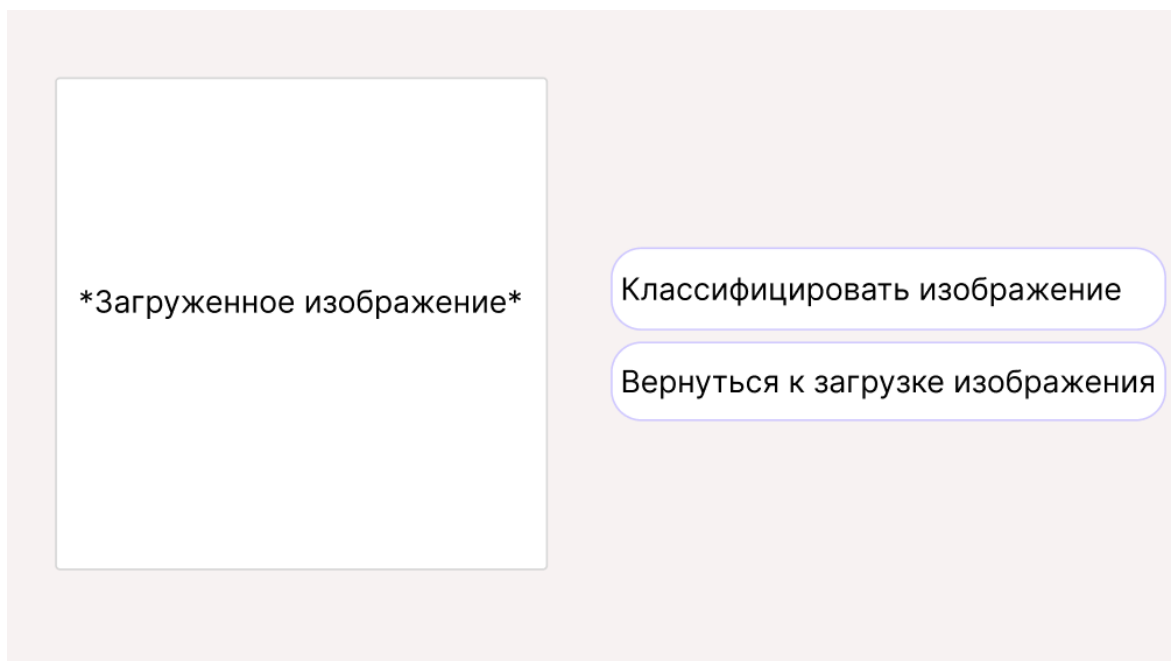


Рис. 3. Окно с загруженным изображением

После выбора действия «Классифицировать изображение» будет происходить сам процесс обработки и классификации, сопровождающий окном ожидания (рис. 4).

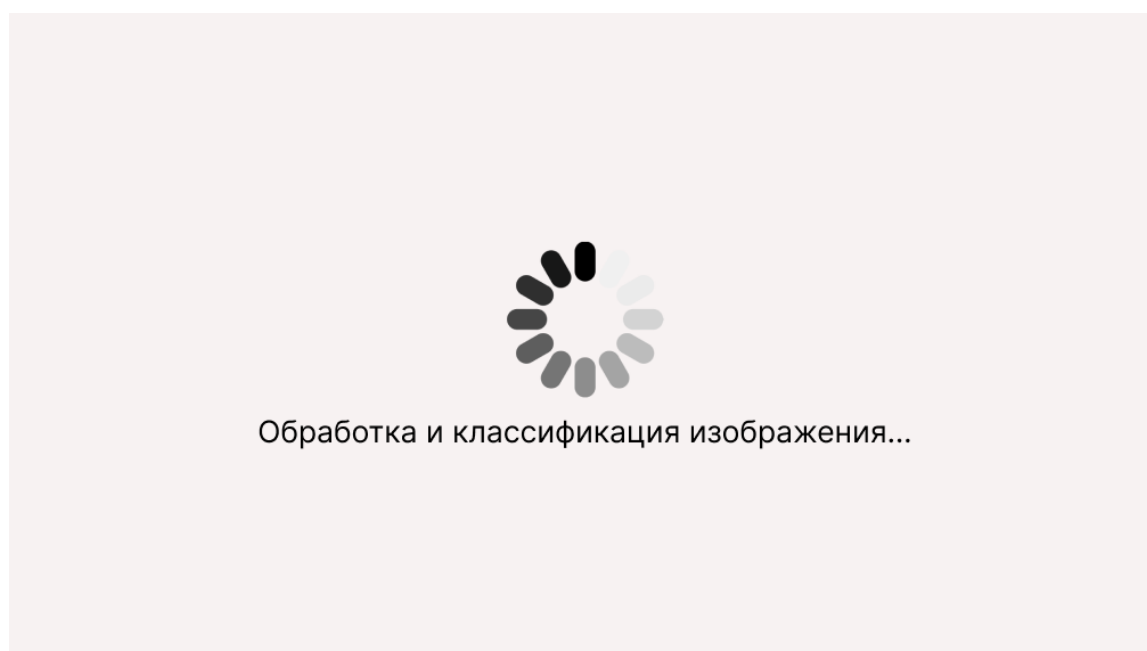


Рис. 4. Окно ожидания

Далее, после выполнения всех операций с изображением, пользователю будет выведен результат в новом окне (рис. 5), в котором отображается использованное пользователем изображение, результат классификации и переход на главную страницу.

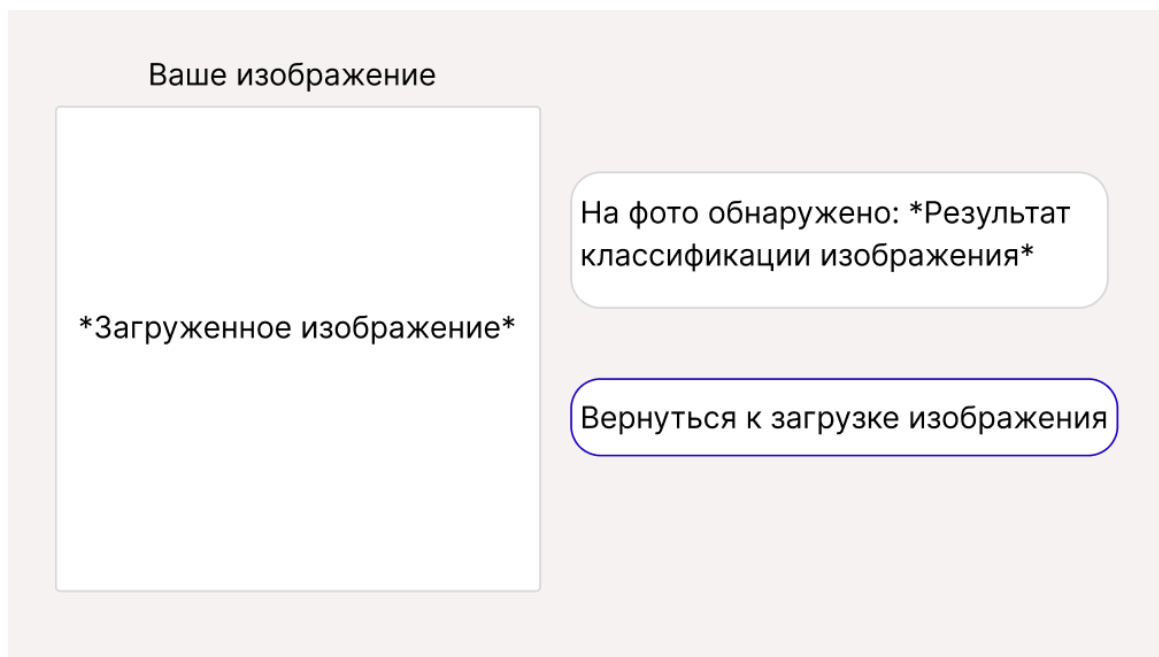


Рис. 5. Окно результата классификации

Заключение

В результате работы было спроектирована система позволяющая классифицировать одежду на изображении. Анализ предметной области показал, что таких систем не так много и главное качественных.

Библиографически список

1. Структура сверточной нейронной сети [Электронный ресурс] / Пояснение значимости каждого слоя сети. URL: <https://www.guru99.com/ru/convnet-tensorflow-image-classification.html> (дата обращения: 24.01.2024).
2. Обзор нейронных сетей для классификации изображений. [Электронный ресурс]. URL: <https://habr.com/ru/companies/intel/articles/415811/> (дата обращения: 30.01.2024).
3. Датасет DeepFashion1 [Электронный ресурс] / Расположен на GoogleDrive. URL: <https://drive.google.com/drive/folders/0B7EVK8r0v71pWGplNFhjc01NbZQ?resourcekey=0-BU3lAk-Nc7HscJu-CyC1yA> (дата обращения: 08.02.2024).

References

1. Structure of a rotational neural network [Electronic resource] / Explanation of the innovation of each layer of the network. URL: <https://www.guru99.com/ru/convnet-tensorflow-image-classification.html> (date of access: 01/24/2024).
2. Review of neural networks for image classification. [Electronic resource]. URL: <https://habr.com/ru/companies/intel/articles/415811/> (date of access: 01/30/2024).

3. Dataset DeepFashion1 [Electronic resource] / Located on Google Drive. URL: <https://drive.google.com/drive/folders/0B7EVK8r0v71pWGplNFhjc01NbzQ?resourcekey=0-BU3lAk-Nc7HscJu-CyC1yA> (date of access: 02/08/2024).

DESIGNING A SYSTEM FOR CLASSIFICATION OF CLOTHING IN PHOTOS USING CONVOLUTIONAL NEURAL NETWORKS

Vershinin Nikita A.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, nikita.vershinin2003@mail.ru

The article is devoted to the design of a system for classifying clothes in photographs using convolutional neural networks. The purpose of this work is to design a system for classifying clothing in photographs, as well as developing a user web interface for interacting with the model. The object of the study is the process of automatic image classification. The subject of the study is the analysis and comparison of machine learning methods used to classify clothing images, with an emphasis on the use of convolutional neural networks. The requirements from the system and the challenges that could be encountered were developed. The article discusses the models used as the basis for a neural network, parameters for setting up this network, approaches for training it, as well as website design. Selected tools for user interface design, as well as resources for constructing UML diagrams. A user interface layout was developed, and the model was expanded and configured.

Keywords: neural network, images, classification, system design.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ ПО СНОУБОРДУ

Водяник Ростислав Александрович, Ракина Валерия Денисовна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, fillmorancheat@gmail.com

Рассматривается проектирование информационной системы для проведения соревнований по сноуборду. Исследуются уже существующие информационные системы с аналогичным назначением. На основе анализа этих систем формулируются основные функциональные требования к разрабатываемому веб-приложению. Описывается обоснованный выбор инструментов для проектирования и разработки, а также создание прототипа пользовательского интерфейса. Для проектирования системы используется язык моделирования UML и инструмент PlantUML. В роли СУБД выступает PostgreSQL для хранения данных. Для разработки серверной части системы будут применяться язык программирования Python и фреймворк Django.

Ключевые слова: информационная система, проектирование, сноуборд, соревнования, проведение соревнований.

Введение

Цель работы – проектирование информационной системы, предназначенной для оптимизации процесса проведения соревнований по сноуборду с применением современных технологий.

Современные спортивные соревнования требуют внедрения цифровых решений для повышения эффективности организации и взаимодействия участников. Сноуборд, как динамичный и растущий вид спорта, предъявляет высокие требования к процессам судейства и оценки результатов, что делает автоматизацию неотъемлемой частью проведения таких мероприятий. Этот процесс включает в себя оценку технической сложности трюков, их исполнения, а также стиля и точности каждого спортсмена. Эти и многие другие аспекты судейских решений отражены на сайте федерации сноуборда России (ФСР) [1].

Были рассмотрены преимущества и недостатки аналогичных информационных систем на примере Longines Live Timing and Scoring [2], Omega Timing [3], FIS Live Timing [4], IFSC Climbing System [5].

Анализ функционала существующих информационных систем выявил ряд преимуществ, в том числе использование технологий реального времени и автоматизацию судейства. Однако, большинство этих систем ориентированы на широкий спектр спортивных дис-

циплин и не адаптированы к специфике и требованиям проведения соревнований по сноуборду, что может привести к определённым ограничениям в их использовании.

Одним из основных недостатков рассмотренных аналогов является их неспособность учитывать уникальные аспекты сноубордических соревнований, такие как оценка технического исполнения и стиля.

Постановка задачи и методы

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

- разработка функционала, позволяющего пользователям в реальном времени просматривать результаты, следить за ходом соревнований через веб-платформу;
- обеспечение доступа к подробной информации о спортсменах, включая их рейтинг, достижения и результаты прошедших соревнований;
- возможность спортсменов просматривать свои результаты, статистику выступлений, оценку судейской коллегии после соревнований;
- разработка модуля для судей, который позволит вносить оценки в режиме реального времени и автоматически обновлять суммарные результаты.

В качестве инструмента для создания UML-диаграмм, было принято решение использовать PlantUML [6], который выделяется своей возможностью быстрого создания и обновления диаграмм через текст, что делает его идеальным для интеграции и автоматизации в рамках разработки, комплексного моделирования и поддержки различных стандартов UML.

Для разработки веб-приложения был выбран язык Python [7]. Этот язык значительно ускоряет разработку, благодаря большому количеству встроенных библиотек, а также большому сообществу разработчиков.

Рассмотрев различные фреймворки для веб-разработки, было принято решение использовать Django [8]. Этот фреймворк предлагает высокую безопасность, встроенную админ-панель и множество готовых компонентов для быстрого создания приложений. Эти преимущества делают Django лучшим выбором для разработки веб-приложений, соответствующих требованиям проектирования ИС.

При выборе СУБД, было учтено то, что система будет являться веб-приложением, поэтому выбор был сделан в пользу PostgreSQL [9], так как эта СУБД обладает архитектурой «клиент-сервер», имеет реляционную модель данных, высокую масштабируемость и надежность.

Результаты проделанной работы

Были определены виды пользователей информационной системы и доступные для них функциональные возможности. В системе будет несколько видов пользователей: спортсмены и тренеры, организаторы, зрители, судейская коллегия. Функциональные возможности для каждого вида пользователей, а также взаимодействия между всеми участниками представлены с помощью диаграмм прецедентов (см. рис. 1) и деятельности (см. рис. 2) соответственно.

Для понимания работы системы необходимо изучить ее структуру и внутренние взаимосвязи подробнее. Диаграмма классов (см. рис. 3) предоставляет возможность визуализировать классы объектов, их атрибуты и методы, а также взаимосвязи между ними.

Логика базы данных о соревнованиях по сноуборду организована в виде ER-диаграммы (см. рис. 4). Она состоит из нескольких основных сущностей и их атрибутов, а также связей между ними. Использование такой базы данных позволит улучшить управление данными и обеспечит доступность информации для всех участников процесса.



Рис. 1. Диаграмма прецедентов, описывающая систему ролей ИС

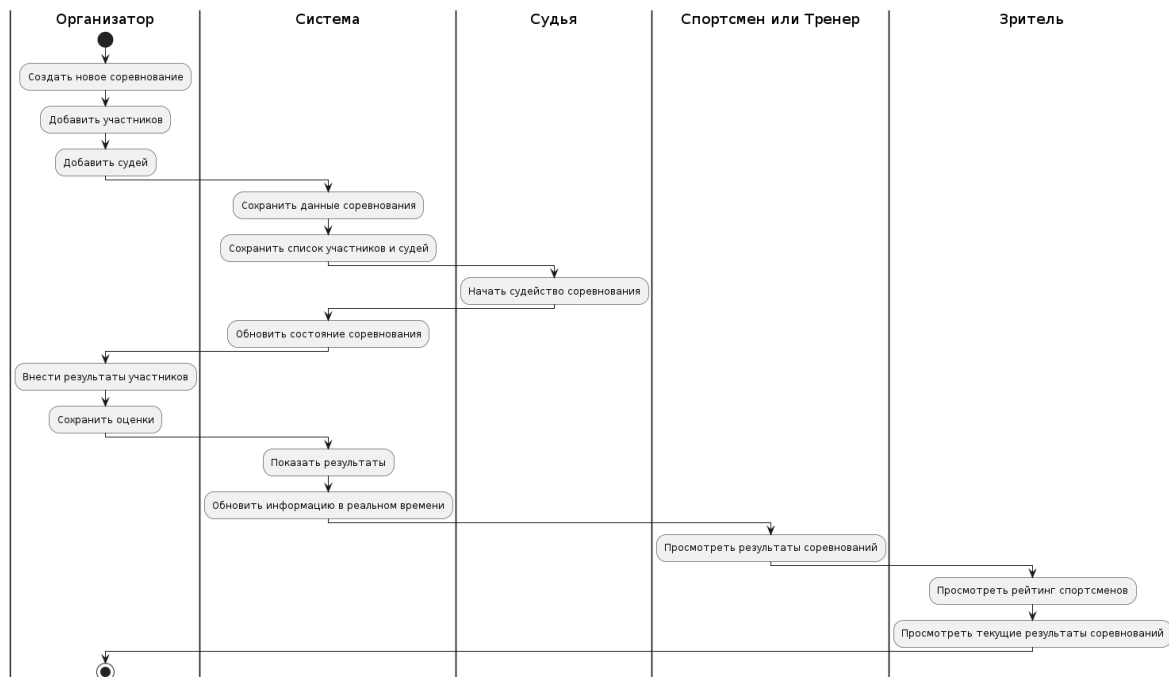


Рис. 2. Диаграмма деятельности, описывающая взаимодействие участников ИС

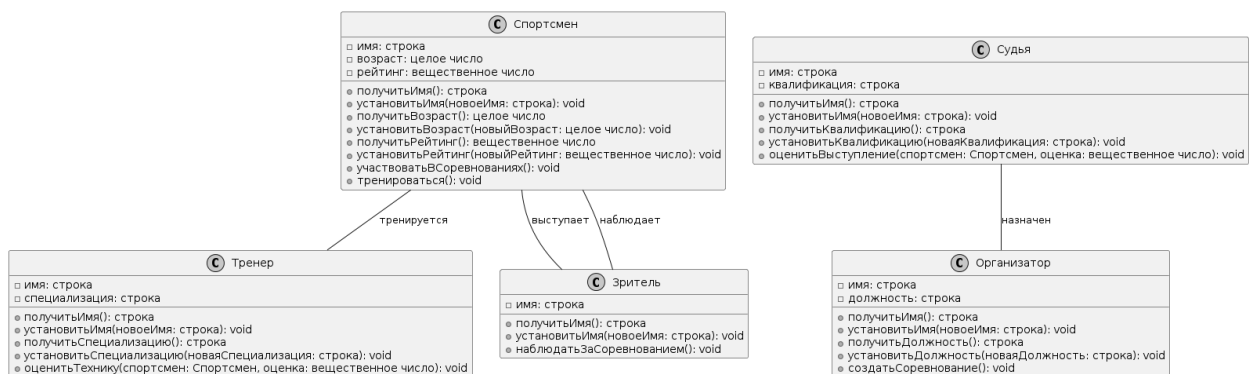


Рис. 3. Диаграмма классов, показывающая архитектуру ИС

В рамках ИС, прототип интерфейса представляет собой интуитивно понятное веб-приложение с разнообразным набором функций для реализации описанных ранее задач. Расположение элементов, представленных в качестве прототипа, может быть изменено в процессе разработки. Пользователь, открывший веб-приложение, автоматически попадает на главную страницу (см. рис. 5). На этой странице, пользователю доступен выбор некоторых действий, а именно: раздел с календарем и результатами соревнований, раздел с информацией о спортсменах, кнопка переключения двух доступных языков (русский, английский), а также выбор роли к одной из которых пользователь относится: сноубордист или тренер, судья, зритель, организатор соревнований.

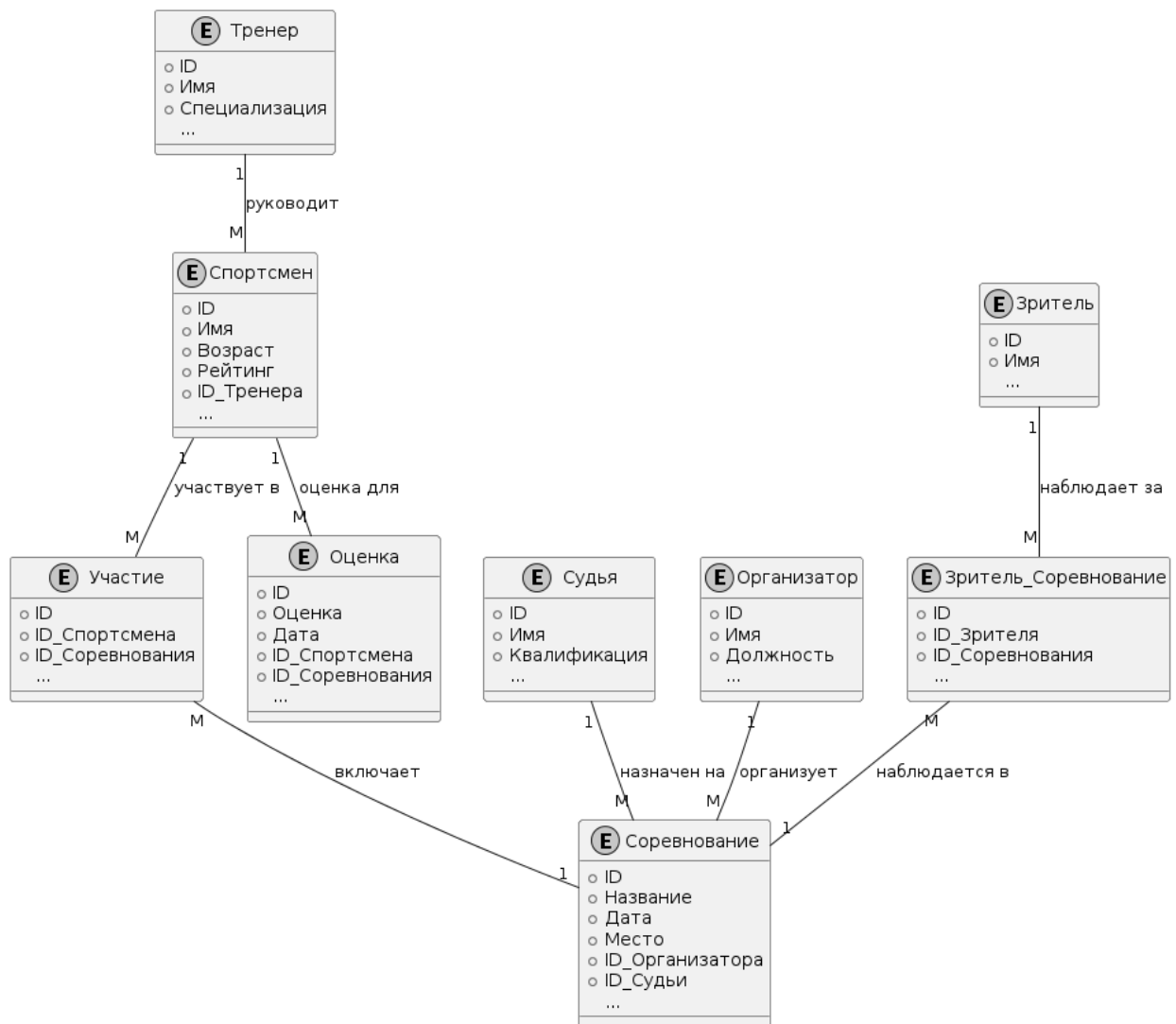


Рис. 4. ER-диаграмма



Необходимо выбрать роль для входа:



Рис. 5. Раздел «Домашняя страница»

Нажав на раздел «Календарь и результаты» (см. рис. 6), пользователю откроется окно, на котором будет доступен расширенный поиск соревнований.

Статус проведения	Дата	Место проведения	Дисциплина	Пол
Проведено	20.02.02024	Губаха	Слоупстайл	М
Не проведено
Не проведено

Рис. 6. Раздел «Календарь и результаты»

Также на главной странице пользователь может выбрать раздел «Спортсмены», который выдаст аналогичный предыдущему разделу расширенный поиск, который позволит найти сноубордиста по различным параметрам (см. рис. 7).

Рейтинг	Имя	Страна	Пол	Дисциплина	FIS код
1	Маркус Клингенд	Норвегия	М	Слоупстайл	9420145
...

Рис. 7. Раздел «Спортсмены»

Для того, чтобы открылось окно с выбранной ролью, пользователю необходимо нажать на одну из кнопок по центру экрана на главной странице. Для всех ролей доступен разный уровень доступа. Сноубордисты и тренеры, а также судьи получают логин и пароль у организаторов, в то время как зрители, могут попасть в свой раздел без регистрации. Также стоит упомянуть, что разделы, представленные выше, являются общедоступными для всех категорий пользователей.

Заключение

В ходе проектирования информационной системы для проведения соревнований по сноуборду был проведен детальный анализ существующих аналогов, что позволило определить основные функциональные требования к системе. Процесс разработки включал в себя выбор инструментов и технологий, таких как UML для моделирования, Python и Django для разработки серверной части, а также PostgreSQL для управления данными.

Были спроектированы ключевые элементы системы, включая диаграммы прецедентов, классов и сущностей базы данных, что позволило визуализировать взаимодействие пользователей с системой и структурировать данные. Кроме того, был спроектирован прототип пользовательского интерфейса, который обеспечивает интуитивно понятное взаимодействие участников соревнований с системой.

Система была спроектирована с учетом потребностей разных категорий пользователей: спортсменов, тренеров, судей и зрителей, что гарантирует удобство ее использования на всех уровнях. Все поставленные цели проектирования достигнуты, что свидетельствует о готовности системы к дальнейшему этапу разработки, включающему реализацию функционала и тестирование на реальных данных.

Библиографический список

1. Федерация сноубординга России [Электронный ресурс]. URL: <https://russnowboard.com/> (дата обращения 13.05.2024).
2. Longines Live Timing and Scoring [Электронный ресурс] // Longines Live Timing and Scoring. URL: <https://www.longineslive.com/> (дата обращения 13.05.2024).
3. Omega Timing [Электронный ресурс]. URL: <https://www.omegatiming.com/> (дата обращения 13.05.2024).
4. FIS Live Timing [Электронный ресурс]. URL: <https://www.fis-ski.com/> (дата обращения 13.05.2024).
5. IFSC Climbing System [Электронный ресурс]. URL: <https://www.ifsc-climbing.org/> (дата обращения 13.05.2024).
6. PlantUML [Электронный ресурс]. URL: <https://plantuml.com/ru/> (дата обращения 13.05.2024).
7. Чернышев С. А. Основы программирования на Python: учеб. пособие для вузов. 2-е изд., перераб. и доп. М.: Издательство Юрайт, 2024. 349 с.
8. Django [Электронный ресурс]. URL: <https://www.djangoproject.com/> (дата обращения 13.05.2024).
9. PostgreSQL [Электронный ресурс]. URL: <https://www.postgresql.org/> (дата обращения 13.05.2024).

References

1. Federatsiya snoubordinga Rossii [Electronic resource]. URL: <https://russnowboard.com/> (date of access: 05/13/2024). (In Russ.)
2. Longines Live Timing and Scoring [Electronic resource]. URL: <https://www.longines timing.com/> (date of access: 05/13/2024).
3. Omega Timing [Electronic resource]. URL: <https://www.omegatiming.com/> (date of access: 05/13/2024).
4. FIS Live Timing [Electronic resource]. URL: <https://www.fis-ski.com/> (date of access: 05/13/2024).
5. IFSC Climbing System [Electronic resource]. URL: <https://www.ifsc-climbing.org/> (date of access: 05/13/2024).
6. PlantUML [Electronic resource]. URL: <https://plantuml.com/ru/> (date of access: 05/13/2024). (In Russ.)
7. Chernyshov S. A. Osnovy programmirovaniya na Python: uchebnoye posobiye dlya vuzov. 2-ye izd., pererab. i dop. M.: Izdatel'stvo Yurayt, 2024. 349 p. (In Russ.)
8. Django [Electronic resource]. URL: <https://www.djangoproject.com/> (date of access: 05/13/2024).
9. PostgreSQL [Electronic resource]. URL: <https://www.postgresql.org/> (date of access: 05/13/2024).

DESIGNING AN INFORMATION SYSTEM FOR SNOWBOARDING COMPETITIONS

Vodyanik Rostislav A., Rakina Valeria D.

Perm State University, 614990, Russia, Perm, Bukireva St., 15, fillmorancheat@gmail.com

The design of an information system for organizing snowboarding competitions is considered. Existing information systems with similar purposes are studied. The main functional requirements for the developed web application are formulated based on the analysis of these systems. The choice of UML as a modeling language is presented. PlantUML is selected as the tool. Also a user interface prototype is created. PostgreSQL is used as the DBMS for data storage. Python programming language and Django framework are used for developing the server side of the system.

Keywords: information system, design, snowboarding, competitions, holding competitions.

ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «ПЛАНИРОВЩИК РОМАНОВ»

Воробьева Дарья Юрьевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, darvor2003@yandex.ru

Кнутова Наталия Сергеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, talieknutova@gmail.com

Рассматривается проектирование мобильного приложения «Планировщик романов». В ходе работы проведен анализ аналогов и результатов опроса потенциальных пользователей – писателей. На основе полученных данных определены требования к проектируемому приложению, а также выбраны программные средства для разработки. Статья включает в себя логику работы мобильного приложения, схему базы данных и будущий интерфейс.

Ключевые слова: мобильное приложение, приложение для писателей.

В современном мире мобильные приложения становятся все более популярными и полезными инструментами в различных сферах жизни. Одной из таких сфер является литература и, в частности, писательство. Авторы, особенно те, кто занимается написанием художественных произведений, часто сталкиваются с необходимостью создания и развития персонажей. Именно для этих целей будет спроектировано мобильное приложение «Планировщик романов».

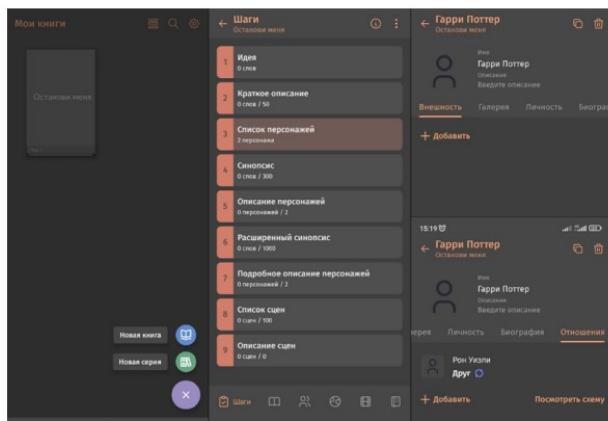
Актуальность данной темы обусловлена растущей потребностью писателей в инструментах, которые помогут им эффективно разрабатывать персонажей в своих художественных произведениях, а также хранить информацию о них.

Одним из ключевых аспектов успешного литературного произведения является наличие убедительных – «живых» – и запоминающихся персонажей. Читатели желают встретиться с реалистичными героями, которые вызывают эмоции и сопереживание. Однако, писатели сталкиваются с трудностями в разработке характеристик своих персонажей и их биографии, а также в последующем хранении этой информации. Использование традиционных методов, таких как записные книжки или отдельные текстовые документы, может быть неудобным и неэффективным.

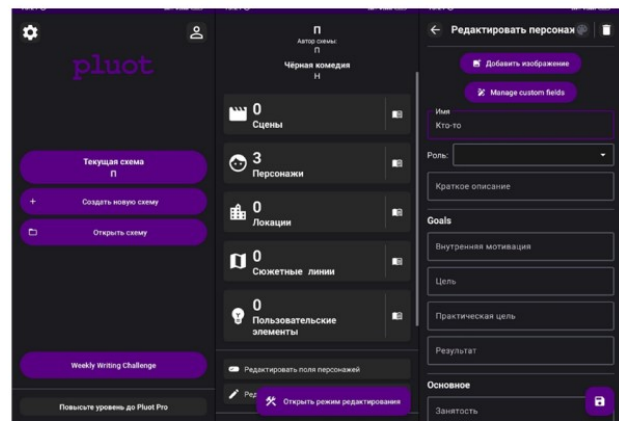
Разработка мобильного приложения «Планировщик романов» представляет собой перспективное решение в области писательства и литературы. Оно позволит писателям эффек-

тивно разрабатывать персонажей, организовывать информацию и создавать качественные художественные произведения. Ожидается, что в результате работы, с учетом выделенных далее вопросов и задач, будет создан удобный и надежный в использовании инструмент. Такое приложение может стать незаменимым помощником для многих писателей, обеспечивая им удобство и результативность в процессе творчества.

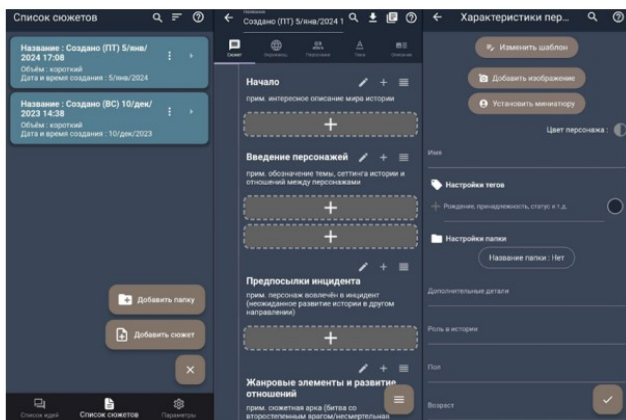
На данный момент уже существуют подобные приложения, такие как: Fabula [1], Story Plotter [2], Fortelling [3], Pluot [4]. Работа этих приложений приведена на рис. 1.



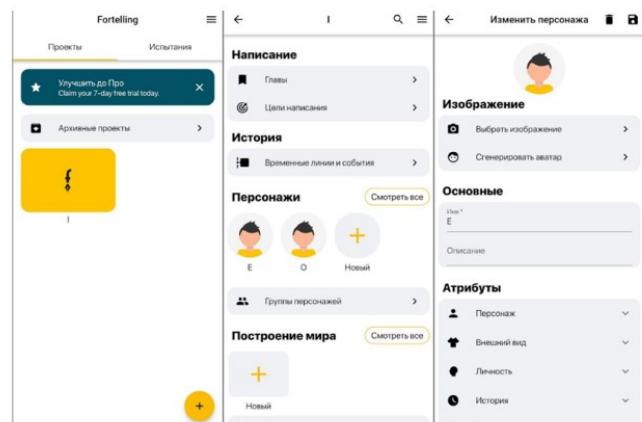
Fabula



Pluot



Story Plotter



Fortelling

Рис. 1. Аналоги приложений

Все рассмотренные приложения неплохи, с моей точки зрения как писателя, но в каждом из них есть что улучшить. Из недостатков можно выделить: отсутствие соавторства, платные функции, ограничения в редакторах сюжета и персонажей, а также отсутствие схемы связей между персонажами.

В рамках исследования потребностей целевой аудитории для разработки приложения по созданию и развитию персонажей в художественных произведениях был проведен опрос среди потенциальных пользователей [5]. Респондентам были заданы следующие вопросы:

1. Используете ли вы какие-либо приложения для хранения «досье» своих персонажей? Если да, то какие конкретно? Если нет, то как вы храните информацию?

2. Почему вы выбрали именно такой способ хранения?

3. Если бы у вас было удобное и понятное приложение (все интуитивно понятно, а также имеется принцип соавторства, создание связей между персонажами и таймлайн), вы бы пользоваться им? Имеете ли вы в нем потребность?

4. Что бы вы хотели, чтобы было в таком приложении помимо перечисленного в 3 вопросе? Если вы пользуетесь похожими приложениями, что мне бы стоило реализовать, а чего стоит избегать?

В опросе приняли участие 23 респондента, заинтересованных в создании и развитии персонажей для художественных произведений.

Исходя из результатов опроса, можно сделать следующие выводы:

- 65% респондентов заинтересованы в специализированном приложении и стали бы им пользоваться;
- 20% пользуются подобным, но могли бы перейти на новое, если оно окажется более удобным и функциональным;
- 30% пользуются офисными программами и облачными сервисами;
- причины выбора приложений: структурирование и систематизация информации, полнота описания аспектов персонажа, визуализация и дополнительные функции.

Итог опроса: многие отмечают потребность в специализированном инструменте и выражают готовность использовать его в своей писательской деятельности. Также в ответах прослеживается несколько пожеланий: простота, интуитивность и универсальность.

После выделения достоинств и недостатков к уже существующим приложениям и анализа мнений респондентов насчет данной темы, можно определить требования к приложению. Требования получились следующими:

1. Пользовательский интерфейс: интуитивно понятный и простой в освоении.
2. Редактор персонажа:
 - модульный, настраиваемый;
 - визуальная галерея, миниатюры.
3. Биография: возможность подробного описания событий.
4. Отношения: настройка различных типов отношений и связей.
5. Временная линия: схема событий в хронологическом порядке.
6. Совместная работа: возможность сотрудничества и обмена идеями.

Правильный выбор средства проектирования и разработки, а также системы управления базами данных, является важным шагом для успешной реализации мобильного приложения для создания и развития персонажей.

Для разработки приложения будет использоваться фреймворк для кроссплатформенной разработки [6]. Он предоставляет несколько преимуществ. Во-первых, совместимость с несколькими операционными системами (Android и iOS) позволяет охватить более широкую аудиторию. Во-вторых, единая кодовая база для разных платформ экономит время и ресурсы. И в-третьих, согласованный пользовательский интерфейс на всех платформах повышает узнаваемость и удобство использования приложения. Таким образом, кроссплатформенная разработка становится оптимальным выбором для эффективной разработки мобильных приложений.

После проведения анализа различных фреймворков для разработки мобильных приложений, было принято решение использовать Xamarin [7].

Проектирование базы данных также является одним из важных этапов разработки приложения, предназначенного для того, чтобы в нем хранили информацию.

В результате анализа СУБД выбор был сделан в пользу Microsoft SQL Server в сочетании с Entity Framework (объектно-реляционным сопоставлением для .NET). Это решение оптимально благодаря интеграции Xamarin с экосистемой Microsoft, а также преимуществам MSSQL в производительности, надежности и документации. Entity Framework позволяет абстрагироваться от деталей работы с базой данных, что упрощает и ускоряет разработку, делая комбинацию MSSQL и Entity Framework идеальной для проекта на Xamarin.

Рассмотрим взаимодействие пользователей с приложением на основе UML-диаграмм. На рисунке 2 представлена диаграмма прецедентов для неавторизованного пользователя.

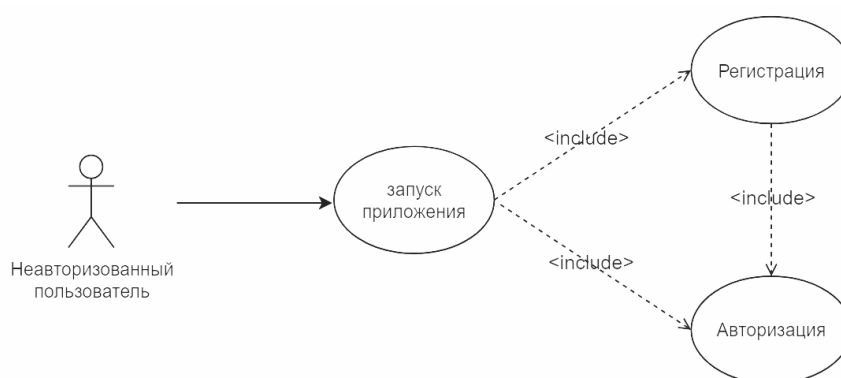


Рис. 2. Неавторизованный пользователь

После регистрации и/или авторизации функции пользователя расширяются. Функционал авторизованного пользователя представлен на рисунке 3.

Рассмотрим чуть подробнее этот этап. Поскольку одним из требований нашей системы является реализация принципа соавторства следует разграничить автора и соавтора. Соответствующая диаграмма представлена на рисунке 4.

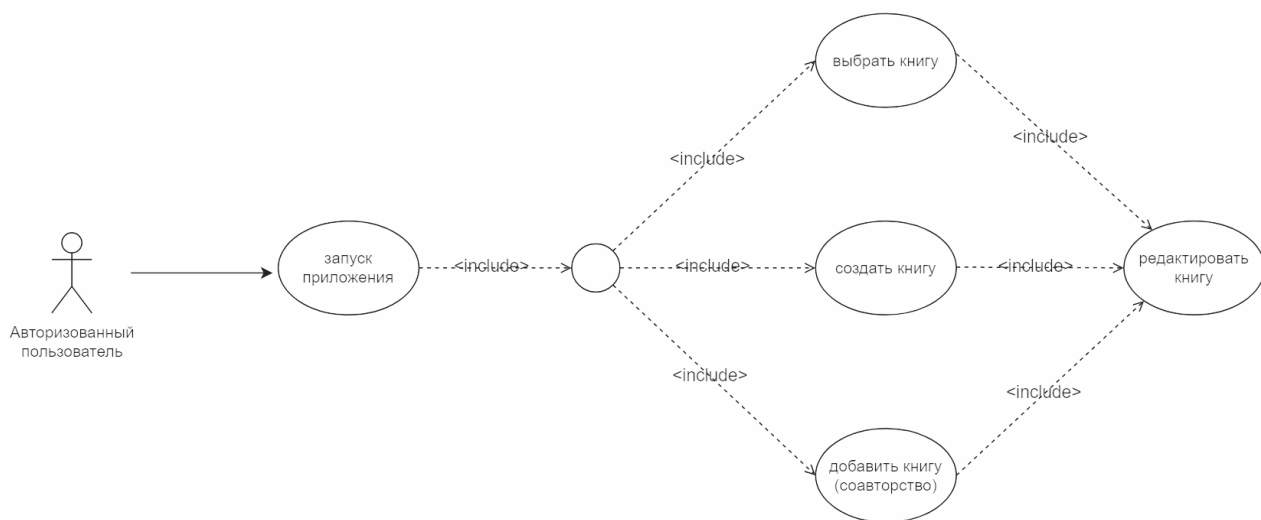


Рис. 3. Авторизованный пользователь

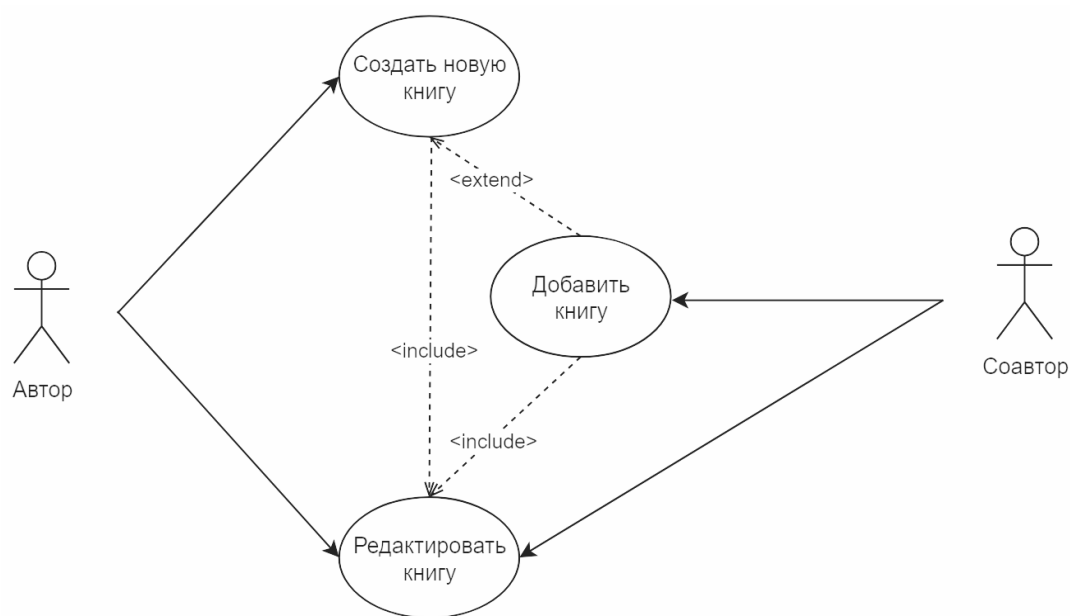


Рис. 4. Автор и соавтор

После того, как пользователь (автор или соавтор – не имеет значения) выбрал редактирование книги, у него есть три доступных функции:

1. Редактирование персонажей;
2. Редактирование связей;
3. Редактирование таймлайнов.

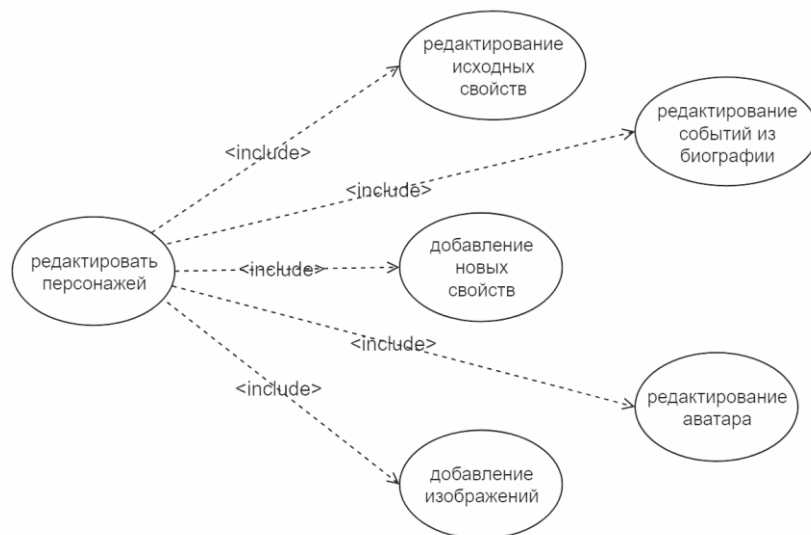


Рис. 5. Редактирование персонажей

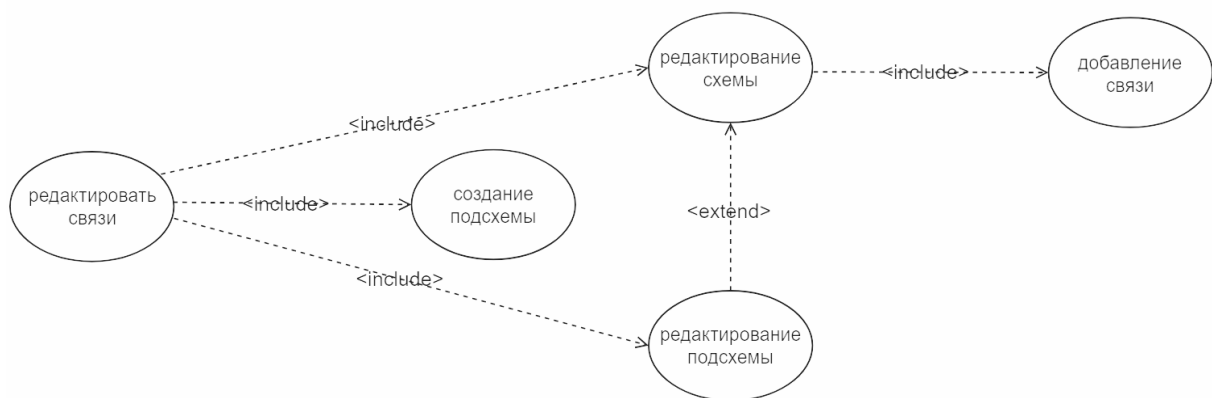


Рис. 6. Редактирование связей

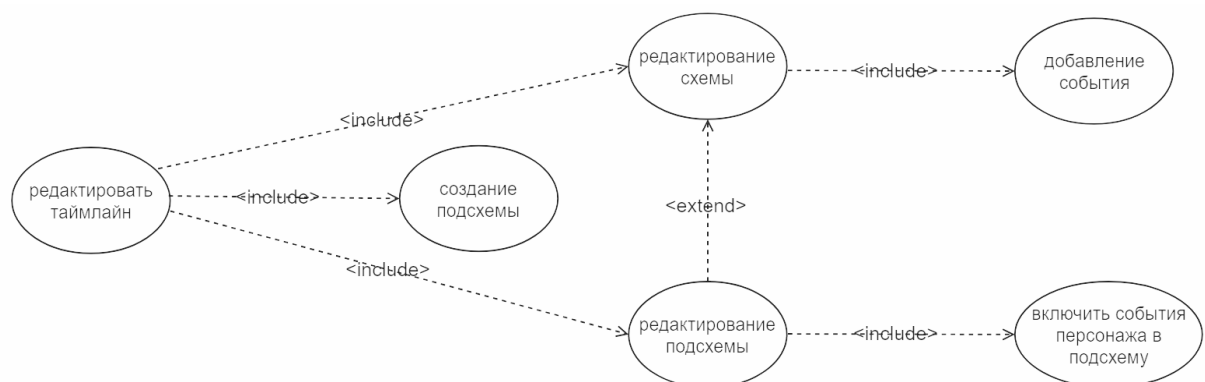


Рис. 7. Редактирование таймлайнов

Далее будут изображены схемы структуры базы данных, отображающие отношения между сущностями (таблицами) и их атрибутами, в виде ER-диаграмм в нотации Мартина.

На рисунке 8 представлены таблицы, отвечающие за хранение информации о пользователях, книгах, персонажах и свойствах персонажей. Изначальные свойства персонажа основаны на часто выбираемых [8][9].

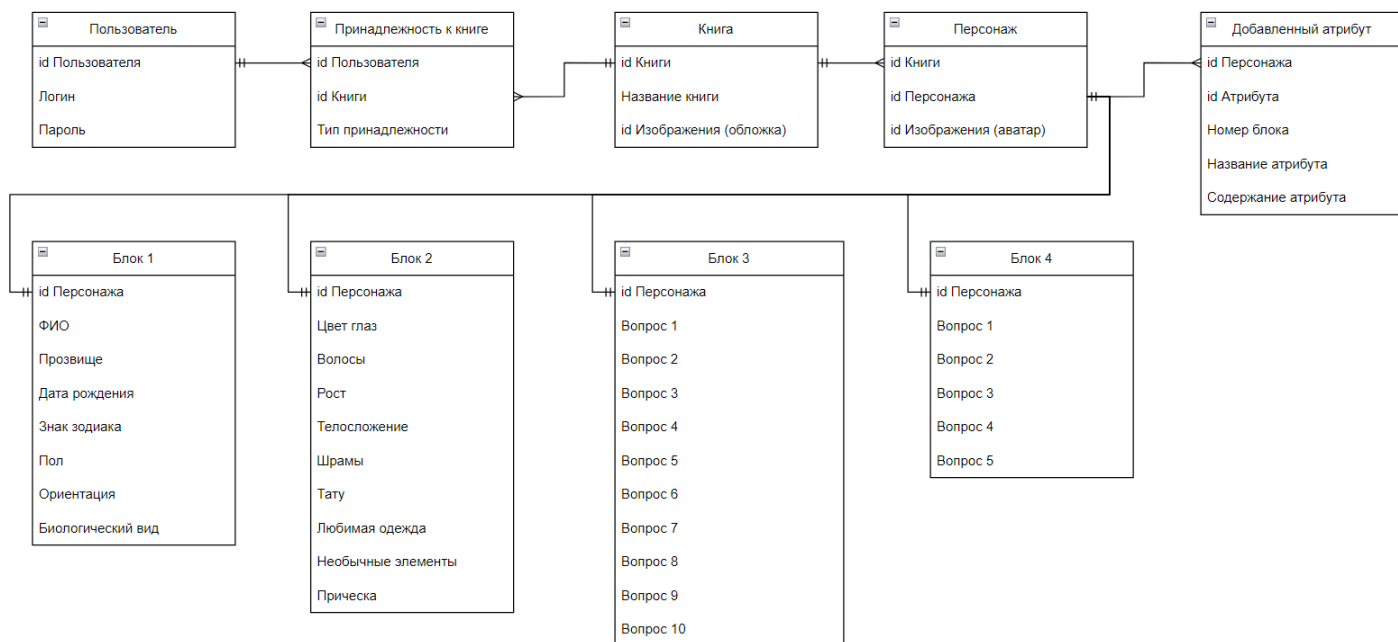


Рис. 8. Таблицы персонажей

На рисунке 9 повторяются ранее представленные таблицы «Книга» и «Персонаж», но с новыми связями к другим таблицам, содержащим в себе информацию об изображениях. Особые изображения, такие как обложка и аватар, хранятся в общей таблице «Изображений», а «Галерея» у каждого персонажа собственная.

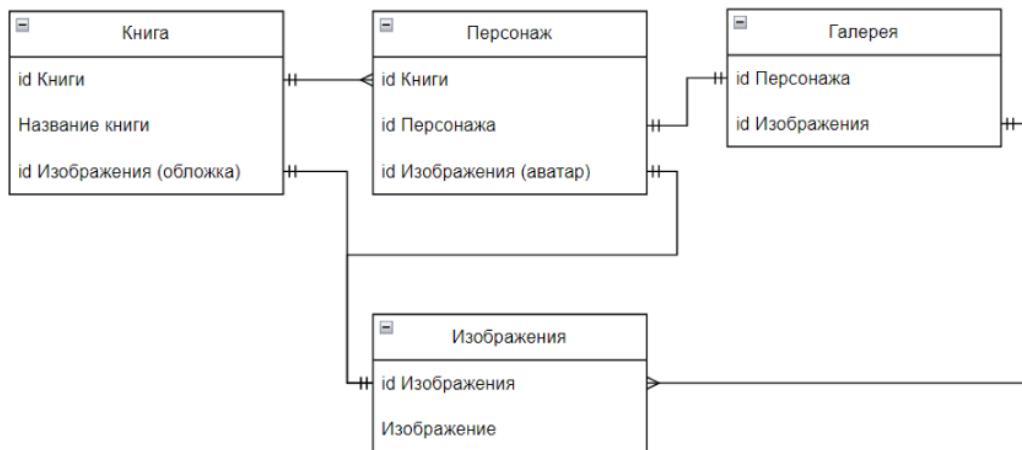


Рис. 9. Таблицы изображений

Структура таблицы «Схемы» и её отношение с таблицей «Связи» приведена на рисунке 10. Между данными таблицами была создана дополнительная, поскольку у каждой схемы может быть несколько связей, а каждая связь может принадлежать к нескольким схемам.

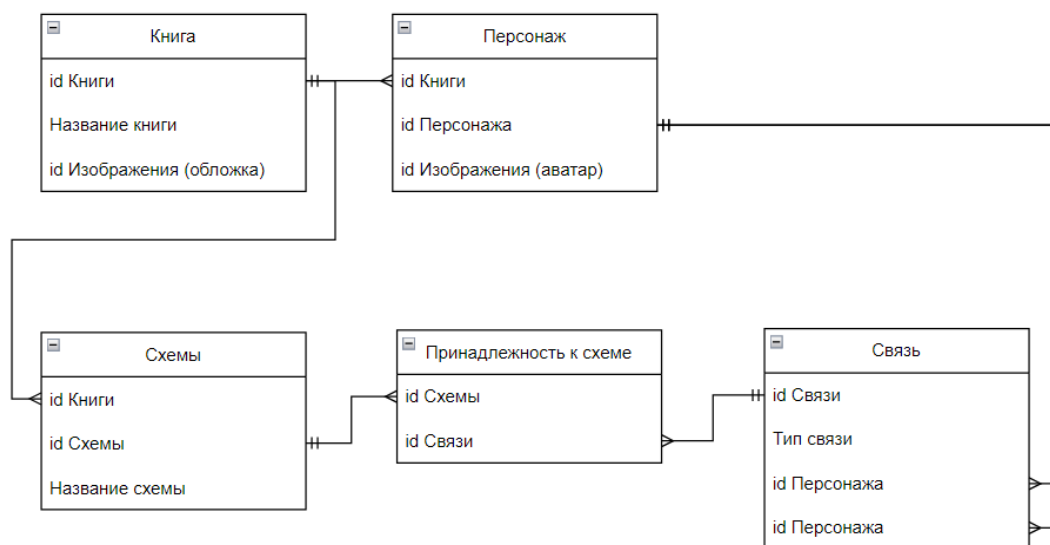


Рис. 10. Таблицы схемы

Аналогичная ситуация и с таблицами таймлайнов, которые представлены на рисунке 11. Отношение M:M между таймлайнами и событиями реализуется через таблицу «Принадлежность к таймлайну». А «Принадлежность к событию» показывает такой же вид отношения между событиями и персонажами.

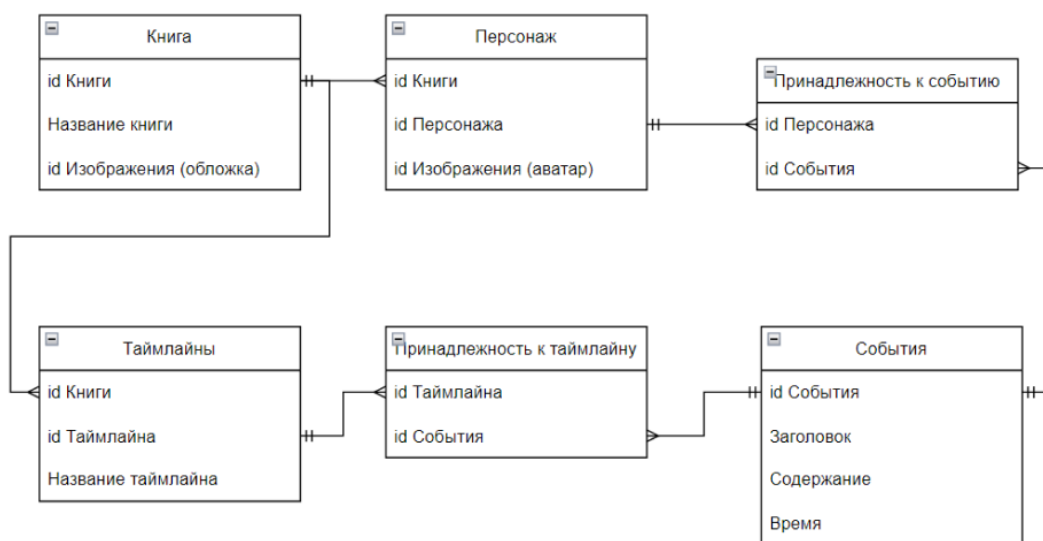


Рис. 11. Таблицы схемы

Исходя из ранее выдвинутых требований, приложение должно иметь простой и понятный интерфейс с набором функций, позволяющих легко создавать и развивать персонажей. Это обеспечит удобство использования и поможет пользователям достичь своих целей в развитии персонажей.

Для создания интерфейса был выбран инструмент Figma [10].

В соответствии с диаграммой прецедентов были разработаны страницы для входа неавторизованных пользователей; главного меню, где можно создавать, добавлять или открывать книги; а также меню книги, предоставляющее доступ к трем основным функциям.

Ниже, на рисунках 12–13, показан интерфейс будущего приложения.

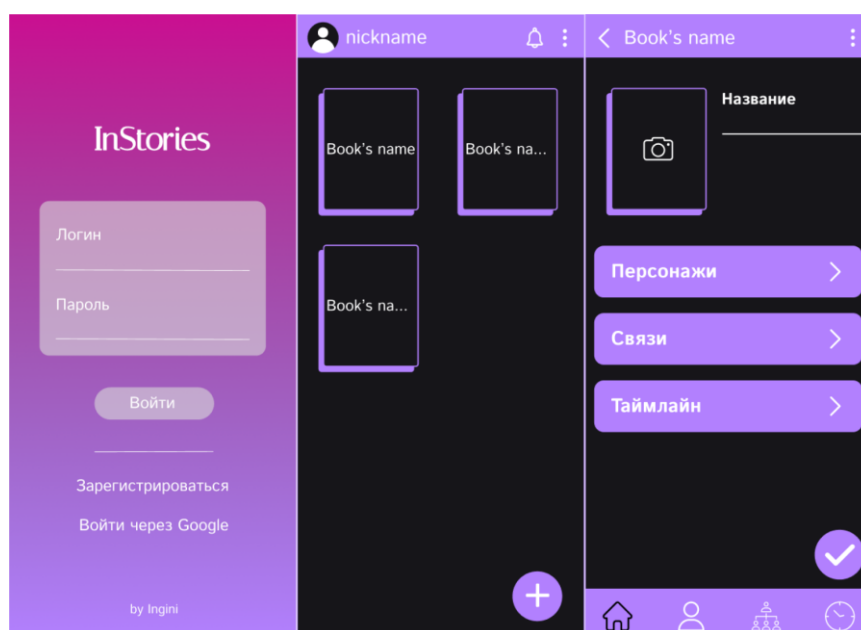


Рис. 12. Интерфейс начала работы

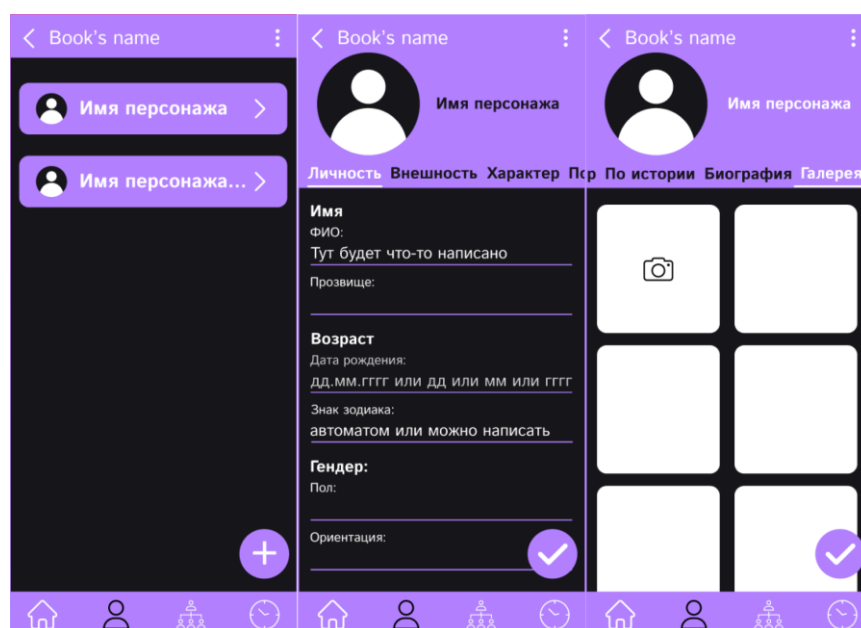


Рис. 13. Интерфейс для редактирования персонажа

Библиографический список

1. Fabula [Электронный ресурс]: мобильное приложение. URL: <https://play.google.com/store/apps/details?id=com.fabula.app> (дата обращения: 15.01.2024).
2. Story Plotter [Электронный ресурс]: мобильное приложение. URL: <https://play.google.com/store/apps/details?id=com.hhatamanga.storyplotter> (дата обращения: 15.01.2024).
3. Fortelling [Электронный ресурс]: мобильное приложение. URL: <https://play.google.com/store/apps/details?id=app.fortelling> (дата обращения: 15.01.2024).
4. Pluot [Электронный ресурс]: мобильное приложение. URL: <https://play.google.com/store/apps/details?id=app.cordes.pluot> (дата обращения: 15.01.2024).
5. Опрос писателей для курсовой работы [Электронный ресурс] / Яндекс.Формы: сервис. URL: <https://forms.yandex.ru/u/6644b91fc417f312aa67196b/> (дата обращения: 15.04.2024).
6. Кросс-платформенные фреймворки для мобильной разработки [Электронный ресурс] / Tproger: блог. URL: <https://tproger.ru/translations/cross-platform-frameworks-for-mobile-development> (дата обращения: 05.02.2024).
7. Xamarin [Электронный ресурс]: официальный сайт. URL: <https://xamarin.ru/> (дата обращения: 03.02.2024).
8. Анкета персонажа [Электронный ресурс] / BAND: блог. URL: <https://bandband.ru/blog/anketa-personazha?ysclid=lrkmsi5sjh275347382> (дата обращения: 13.04.2024).
9. Досье на героя [Электронный ресурс] / Справочник писателя: сайт. URL: <https://avtoram.com/dose-na-geroya/?ysclid=lrkmur02dw826083965> (дата обращения: 13.04.2024).
10. Figma [Электронный ресурс]: официальный сайт. URL: <https://www.figma.com/> (дата обращения: 22.11.2023).

References

1. Fabula [Electronic resource]: mobile application. URL: <https://play.google.com/store/apps/details?id=com.fabula.app> (date of access: 01/15/2024).
2. Story Plotter [Electronic resource]: mobile application. URL: <https://play.google.com/store/apps/details?id=com.hhatamanga.storyplotter> (date of access: 01/15/2024).
3. Fortelling [Electronic resource]: mobile application. URL: <https://play.google.com/store/apps/details?id=app.fortelling> (date of access: 01/15/2024).

4. Pluot [Electronic resource]: mobile application. URL: <https://play.google.com/store/apps/details?id=app.cordes.pluot> (date of access: 01/15/2024).
5. Survey of writers for course work [Electronic resource] / Yandex.Forms: service. URL: <https://forms.yandex.ru/u/6644b91fc417f312aa67196b/> (date of access: 04/15/2024).
6. Cross-platform frameworks for mobile development [Electronic resource] / Tproger: blog. URL: <https://tproger.ru/translations/cross-platform-frameworks-for-mobile-development> (date of access: 02/05/2024).
7. Xamarin [Electronic resource]: official website. URL: <https://xamarin.ru/> (date of access: 02/03/2024).
8. Overview of cross-platform frameworks for mobile development [Electronic resource] / Sibirix: blog. URL: <https://blog.sibirix.ru/crossplatform-frameworks/?ysclid=lvgljdn9r2298924484> (date of access: 05.02.2024).
9. Which framework should I choose for developing a mobile application? [Electronic resource] / Zen: articles. URL: <https://dzen.ru/a/ZAjCKTMUvhqw6SHh> (date of access: 02/15/2024).
10. Figma [Electronic resource]: official website. URL: <https://www.figma.com/> (date of access: 11/22/2023).

DESIGNING THE MOBILE APPLICATION “PLANNER OF NOVELLS”

Vorobyova Daria U.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, darvor2003@yandex.ru

Knutova Nataliya S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, talieknutova@gmail.com

The design of the mobile application “Planner of novels” is considered. In the course of the work, an analysis of analogues and the results of a survey of potential users – writers was carried out. Based on the data obtained, the requirements for the designed application are determined, and software tools for development are selected. The article includes the logic of the mobile application, the database schema and the future interface.

Keywords: mobile application, application for writers.

ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ СОЦИАЛЬНОГО ТАКСИ

Голдобин Денис Андреевич, Гасумова Светлана Евгеньевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, ptaha.re@yandex.ru

В данной работе рассматривается проектирование приложения для социального такси. Показана актуальность проблемы низкой мобильности людей с инвалидностью в России. Описано социальное такси и важность развития данной технологии для общества. Проведены сравнение и анализ существующих российских приложений для социального такси. На основе анализа определены задачи, которые должна решать проектируемая система. Выбраны инструменты для моделирования поведения приложения, создания прототипов пользовательского интерфейса и средства для его дальнейшей разработки. Работа содержит диаграмму прецедентов, отражающая взаимодействие акторов с системой, а также прототипы интерфейса.

Ключевые слова: социальное такси, мобильное приложение, информационные технологии, инвалиды.

Введение

На протяжении многих лет современное общество выступает за равные права и возможности людей с инвалидностью. Каждый год в рамках государственной политики принимаются новые меры поддержки лиц с ограниченными возможностями. Но на данный момент многие из них не имеют таких же шансов в жизни, как здоровые люди. Школа, работа, услуги, городская инфраструктура не всегда доступны для них.

В России одной из государственных мер поддержки людей с инвалидностью является услуга социального такси. Социальное такси – это специальный транспортный сервис, который обеспечивает доступный транспорт для людей с ограниченными возможностями. Сегодня во всех сферах нашей жизни происходит цифровизация, почти каждый бизнес или сервис услуг имеет мобильное приложение, сайт или любую другую информационную систему, которая упрощает получение услуги. Однако социальное такси в Пермском крае не имеет своего приложения или сайта для заказа, а вызывать такси по телефону не всегда удобно и такой способ имеет ряд недостатков. Поэтому для повышения качества оказания услуги социального такси необходимо создать мобильное приложение, которое упростило бы его заказ.

Таким образом, объектом исследования является автоматизация услуги социального такси в России, а предметом исследования выступает приложение для решения задач социального такси.

Актуальность проблемы мобильности людей с инвалидностью

В современном мире численность людей с инвалидностью достаточно высока. Это связано с высокими научными достижениями и открытиями в медицине, которые позволяют сохранять и продлевать жизни людей, имеющих даже стойкие нарушения здоровья. Согласно оценкам Всемирной организации здравоохранения, в мире 1,3 млрд людей страдают существенными ограничениями здоровья – это каждый шестой человек (или 16% мирового населения) [1]. В частности, в Европейском Союзе на данный момент число людей с инвалидностью достигло 87 млн человек [2]. В России, как и в других развитых странах, численность людей с инвалидностью довольно высока. По данным Федеральной службы государственной статистики в России количество людей с инвалидностью на конец 2023 г. достигло 11 млн человек (или каждый 8-9) [3].

Люди с инвалидностью имеют уникальные потребности, которые не всегда учитываются в современном обществе. Часть из таких людей маломобильны и испытывают большие трудности с передвижением. Ограниченная мобильность человека становится серьёзным препятствием для ведения активного образа жизни, что негативно сказывается на качестве жизни. Самое тяжелое положение в обществе занимают инвалиды-колясочники. Недоступность общественных пространств и услуг создаёт серьезные препятствия для нормальной жизни. Это включает в себя отсутствие подъездов к общественным зданиям, неадаптированные транспортные средства и отсутствие специальных инфраструктурных удобств. В Европейском союзе 52% людей с ограниченными возможностями чувствуют себя дискриминированными. А в России по опросу Всероссийского центра изучения общественного мнения в 2021 г., главными проблемами, с которыми сталкиваются инвалиды, называли трудности перемещения по городу и сложности со входом и выходом из дома [4].

Проблему повышения качества жизни маломобильных людей с инвалидностью в России в рамках государственной социальной политики решают с помощью сервиса социального такси.

Понятие социальное такси

Социальное такси – это специальный транспортный сервис, предоставляемый государством или общественными организациями для обеспечения доступности транспорта, для маломобильных групп населения. Этот сервис обеспечивает удобные условия для перевозки пассажиров с инвалидностью, такие как адаптированный салон, помощь в посадке и высадке пассажира. В России услуга социального такси гарантирована для людей с инвалидностью и регулируется постановлением Правительства РФ от 25 марта 2010 г. № 192 «Об утверждении Правил оказания транспортных услуг пассажирам и багажу автомобильным транспортом

общего пользования и о порядке осуществления таких услуг» [5]. В рамках этого нормативно-правового акта определены основные требования к организации и предоставлению социального такси, включая техническое состояние транспортного средства, стандарты обслуживания и квалификацию водителей. Социальное такси играет важную роль в обеспечении маломобильных граждан, оно позволяет им без особых усилий и дискомфорта добираться до социально значимых организаций в городе.

Опыт автоматизации предоставления услуги социального такси

В некоторых регионах России уже были предприняты попытки автоматизации сервиса социального такси с помощью мобильного приложения. Однако эти приложения не всегда удовлетворяют потребностям пользователей и имеют ряд недостатков, таких как сложности с доступностью и навигацией. Например, приложение «Социальное такси», созданное ГУП «Мосгортранс», в магазине приложений «Google Play» имеет оценку в 1,6 баллов из 5. Люди в отзывах массово жалуются на невозможность заказа такси, в приложении отсутствует доступное время для заказа, а при нахождении доступного времени отправить заказ не получается. Приложение не позволяет полноценно следить за статусом вашего заказа. При отмене заказа диспетчером в приложении для вас статус заказа не изменится. В статусе заказа также не отображается ни номер, ни марка автомобиля – её вам сообщает диспетчер по телефону, а в таком случае полностью отпадает надобность в приложении, так как информирование о состоянии заказа происходит через диспетчера.

Исходя из отзывов пользователей на существующее приложение, были составлены задачи, которые должна решать проектируемая система:

- 1) обеспечить регистрацию в приложении как для пользователей, так и для компании-оператора такси;
- 2) предоставить удобный функционал для создания заказа, его удаления и изменения, а также отслеживание актуальной информации о его статусе.

Для моделирования процессов информационной системы были выбраны язык UML и графический редактор diagrams.net [6]. В качестве языка программирования для реализации системы будет использоваться Dart, а в виде СУБД для управления базой данных – PostgreSQL.

Моделирование поведения системы

Диаграмма прецедентов позволит смоделировать всех действующих лиц и их возможные действия. Акторами системы будут являться пользователь, исполнитель (водитель), диспетчер и администратор – лицо, контролирующее работу системы. На представленной диаграмме можно увидеть всех акторов и их возможные действия (см. рис. 1).

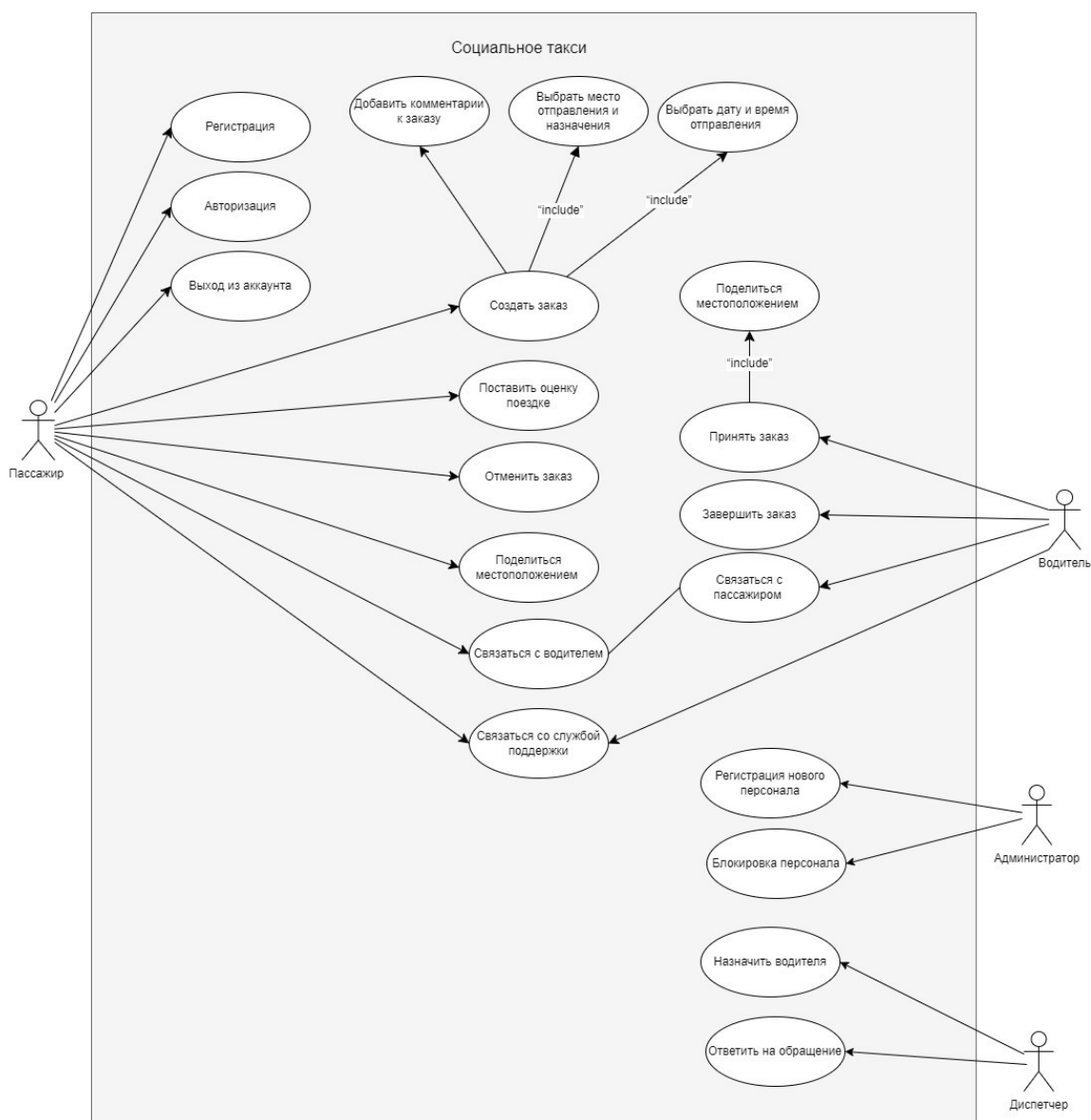


Рис. 1. Диаграмма прецедентов

Проектирование прототипа интерфейса

В качестве инструмента для создания прототипов был выбран Figma [7], так как оно имеет бесплатную версию и имеет подходящий набор инструментов для прототипирования интерфейса мобильного приложения, без каких-либо ограничений.

При создании макетов учитывались правила создания дизайна приложения для людей с ограниченными возможностями, например, максимально читабельный текст за счёт правильного выбора размера и цвета шрифта. Для улучшения восприятия к тексту добавлены значки и иллюстрации. Кнопки в приложении имеют большой размер и цветовой акцент, что снижает вероятность случайного клика по ненужному элементу интерфейса.

На основном экране приложения отображается карта, кнопки выбора точки отправления и точки назначения, а также выбор вида машины для заказа. Пользователь может выбрать обычный автомобиль или специализированный транспорт (см.

Рис. 2).

При выборе точки отправления и назначения можно пройти на следующий этап оформления заказа (см. Рис. 3).

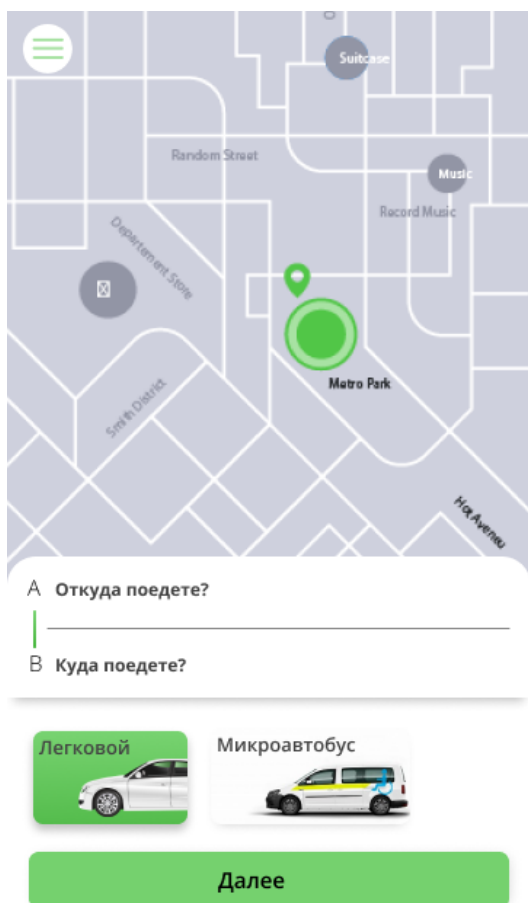


Рис. 2. Основной экран

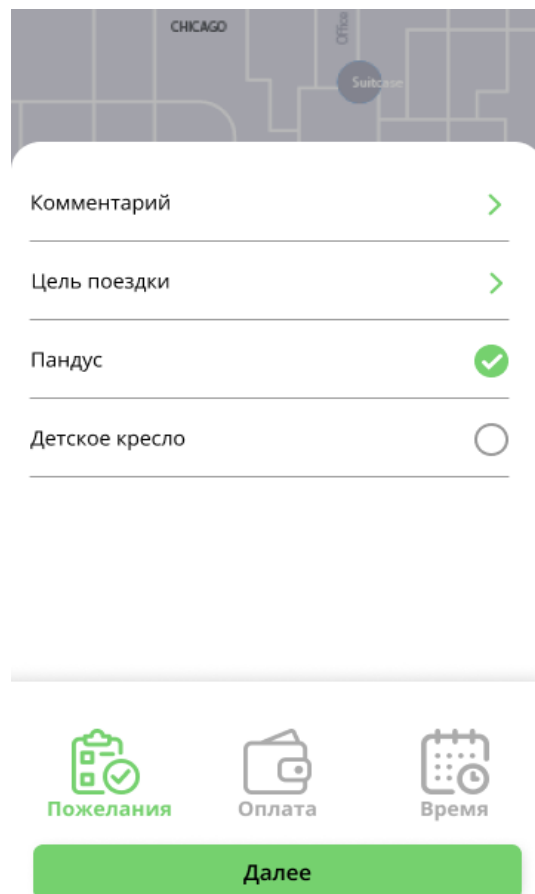


Рис. 3. Меню оформления заказа

Таким образом, опираясь на существующие проблемы в сфере мобильности людей с инвалидностью в России, спроектированное приложение для социального такси отвечает требованиям. Это удобно, доступен и понятный интерфейс. Также приложение позволяет обеспечить регистрацию как для пользователей, так и для компании-оператора такси, предоставляет удобный функционал для создания заказа, управления им и отслеживание актуальной информации о его статусе. Внедрение данного приложения позволит повысить мобильность людей с инвалидностью, обеспечить повышение качества их жизни, создав возможности для их социальной интеграции с помощью цифровых технологий.

Библиографический список

1. Инвалидность // Всемирная организация здравоохранения [Электронный ресурс]. URL: <https://www.who.int/ru/news-room/fact-sheets/detail/disability-and-health> (дата обращения 18.03.2024).
2. Занятость, социальные вопросы и инклюзивность // Европейский союз [Электронный ресурс]. URL: <https://ec.europa.eu/social/main.jsp?catId=1137> (дата обращения 18.03.2024).
3. Данные об общей численности инвалидов по группам инвалидности // Федеральная служба государственной статистики [Электронный ресурс]. URL: https://rosstat.gov.ru/storage/mediabank/pi_1.1.xlsx (дата обращения 25.03.2024).
4. Инвалиды и общество: положение, отношение, проблемы // ВЦИОМ [Электронный ресурс]. URL: <https://wciom.ru/analytical-reviews/analiticheskii-obzor/invalidy-i-obshchestvo-polozhenie-otnoshenie-problemy> (дата обращения 18.03.2024).
5. Постановление Правительства РФ от 01.10.2020 N 1586 (ред. от 23.03.2024) «Об утверждении Правил перевозок пассажиров и багажа автомобильным транспортом и городским наземным электрическим транспортом». URL: https://www.consultant.ru/document/cons_doc_LAW_364025/ (дата обращения 26.03.2024).
6. diagrams.net [Электронный ресурс]. URL: <https://www.diagrams.net/> (дата обращения: 16.04.2024).
7. Figma [Электронный ресурс]. URL: <https://www.figma.com/> (дата обращения: 17.04.2024).

References

1. Disability // World Health Organization [Electronic resource]. URL: <https://www.who.int/ru/news-room/fact-sheets/detail/disability-and-health> (date of access: 03/18/2024).
2. Employment, Social Affairs & Inclusion // European Commission [Electronic resource]. URL: <https://ec.europa.eu/social/main.jsp?catId=1137> (date of access: 03/18/2024).
3. Dannye ob obshchej chislennosti invalidov po gruppam invalidnosti // Federal'naya sluzhba gosudarstvennoj statistiki [Electronic resource]. URL: https://rosstat.gov.ru/storage/mediabank/pi_1.1.xlsx (date of access: 03/25/2024). (In Russ.).
4. Invalidy i obshchestvo: polozhenie, otnoshenie, problemy // VCIOM [Electronic resource]. URL: <https://wciom.ru/analytical-reviews/analiticheskii-obzor/invalidy-i-obshchestvo-polozhenie-otnoshenie-problemy> (date of access: 03/18/2024). (In Russ.).
5. Postanovlenie Pravitel'stva RF ot 01.10.2020 N 1586 (red. ot 23.03.2024) «Ob utverzhdenii Pravil perevozk passazhirov i bagazha avtomobil'nyim transportom i gorodskim nazemnym

elektricheskim transportom» URL: https://www.consultant.ru/document/cons_doc_LAW_364025/ (date of access: 03/26/2024). (In Russ.)

6. Diagrams.net [Electronic resource]. URL: <https://www.diagrams.net/> (date of access: 04/16/2024).

7. Figma [Electronic resource]. URL: <https://www.figma.com/> (date of access: 04/17/2024).

DESIGNING A SOCIAL TAXI APPLICATION

Goldobin Denis Andreevich, Gasumova Svetlana Evgenievna

Perm State National Research University, 15 Bukireva str., Perm, 614068, Russia,
ptaha.re@yandex.ru

In this paper, we consider the design of an application for a social taxi. The relevance of the problem of low mobility of people with disabilities in Russia is shown. Social taxi and the importance of the development of this technology for society are described. The comparison and analysis of existing Russian applications for social taxi are carried out. Based on the analysis, the tasks that the designed system should solve are determined. The tools for modeling application behavior, prototyping the user interface, and tools for its further development have been selected. The work contains a precedent diagram reflecting the interaction of actors with the system, as well as interface prototypes.

Keywords: social taxi, mobile application, information technology, people with disabilities.

ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВЫБОРА И БРОНИРОВАНИЯ МЕСТ В РЕСТОРАНАХ

Дьякова Ирина Андреевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, irinadyakova.mat@yandex.ru

Степанов Владимир Анатольевич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, svaperm@gmail.com

В данной статье рассматривается процесс проектирования и документирования информационной системы выбора и бронирования мест в ресторанах. Был проведен обзор аналогичных систем бронирования, выделены и проанализированы как их положительные, так и отрицательные аспекты. Полученные выводы и наблюдения послужили основой для определения функциональных требований, которыми должна обладать разрабатываемая система. В рамках исследования также была представлена архитектура мобильного приложения и подобрано необходимое программного обеспечения. Особое внимание в статье уделено разработке базы данных, которая будет использоваться во взаимодействии с мобильным приложением. Неотъемлемой частью исследования стали меры по обеспечению безопасности сервера с базой данных, что является критическим аспектом в современных информационных технологиях. Общая значимость данной статьи заключается в ее практической применимости и потенциальной пользе для дальнейшего развития мобильного приложения, основанного на системе выбора и бронирования мест в ресторанах.

Ключевые слова: мобильное приложение, бронирование.

С развитием интернета и технологий, мобильные приложения становятся все более популярными в сфере услуг. В настоящее время рестораны предпочитают создавать сайты или группы в социальных сетях для своего продвижения. На данных сайтах можно посмотреть меню заведения или забронировать место. Мобильные приложения для обзора ресторанов позволяют пользователям быстро и удобно найти подходящее заведение, основываясь на блюде и наличии свободных мест, сделать заказ и бронирование. Не придется лично связываться с заведением. Таким образом приложение является актуальным, так как облегчит работу не только ресторану, но и его клиентам.

В ходе данного исследования были выделены положительные стороны и недостатки систем.

LeClick [1] – это сервис онлайн бронирования столиков в ресторанах. Главным плюсом приложения является большое количество как разных сетей ресторанов, так и отдельных.

У пользователя есть личный кабинет, в котором бронь просматривается в online режиме. Также у данного приложения есть существенный недостаток: нет возможности посмотреть меню и цены. Чтобы сделать бронирование, необходимо или изначально знать меню ресторана, или сначала посмотреть его на самом сайте ресторана, а потом уже переходить на другой, чтобы сделать бронирование.

ToМесто [2] – это сервис онлайн бронирования столиков в ресторанах Перми. Главным отличием от предыдущего сервиса является то, что можно увидеть примерные цены в ресторане, ресторан можно выбрать по его кухне, категориям, районам и особенностям. Но здесь также нельзя посмотреть само меню.

Bourgeois Bohemians (BOBO) [3] – ресторан, у которого есть свой сайт. На нем можно забронировать место и посмотреть меню с ценами, но без указания состава блюд. Также нельзя при бронировании сразу сделать заказ.

Rostic's – Сеть быстрого питания со своим мобильным приложением. Там нет бронирования мест, но можно посмотреть меню ресторана с его ценами и составом этих блюд. Сделать заказ через приложение и оплатить его онлайн можно только в данный момент.

После анализа существующих систем бронирования мест в ресторанах были выделены следующие функциональные требования, которые должны быть реализованы в системе:

- управление профилем пользователя: регистрация и аутентификации пользователя, а так же редактирование профиля,
- просмотр ресторанов – у пользователя должна быть возможность посмотреть информацию по каждому ресторану, а именно адрес, контактный номер, график работы,
- выбор свободного места в ресторане – у пользователя должна быть возможность видеть свободные места в реальном времени или на небольшой срок вперед,
- просмотр меню ресторанов – пользователь должен видеть информацию о блюдах: цена и состав,
- добавление блюд и заказов в список понравившихся – у пользователя будет возможность сохранять любимы блюда или наборы с дальнейшим их заказам,
- чтение и добавление отзывов о блюдах – пользователь вправе читать оценки и отзывы других пользователей оставлять свои,
- редактирование блюд – из блюд пользователь может убирать ингредиенты, которые ему не нравятся,
- оплата бронирования – пользователь должен иметь возможность оплачивать бронь и заказ,
- смена владельца бронирования – пользователь вправе подарить свой заказ другому.

Архитектура приложения и выбор программного обеспечения для разработки является ключевым фактором, влияющим на успешность разработки системы и дальнейшего сопровождения. Архитектура определяет структуру, способы взаимодействия с внешними системами и базами данных, а также имеет значительное влияние на производительность, надежность и безопасность. Выбор программного обеспечения для разработки является одним из важных решений, которое может повлиять на различные аспекты разработки, такие как производительность и затраты, масштабируемость и гибкость.

Архитектура системы [4] – это описание основных компонентов и модулей, включая принципы их работы и взаимосвязи друг с другом. Кроме этого она отображает эволюцию и вариации развития. Многоуровневая архитектура позволяет четко разделять ответственность между уровнями, что облегчает разработку. Изменения на одном уровне не влияют на другие, что говорит о гибкости этой архитектуры. Также в нее можно добавлять новые уровни или расширять существующие.

Одним из видов многоуровневой архитектуры является чистая архитектура [5] – это концепция, разработанная Робертом Мартином, которая предполагает четкое разграничение обязанностей между различными компонентами мобильного приложения. Оно разбивается на функциональные независимые модули, которые взаимодействуют друг с другом. Между модулями передаются только те данные, которые необходимы для выполнения задачи. Этот подход минимизирует сложность каждого модуля и возникновение ошибок, а так же ускоряет их устранение.

В качестве языка программирования [6], на котором будет написано мобильное приложение, был выбран Kotlin. Kotlin – язык программирования, основанный на базе Java. Kotlin молодой и перспективный язык. Он активно развивается, и многие компании переходят на него. Согласно статистике [7], его рейтинг за последний год поднялся с 33 до 19 места. Код на языке Kotlin является лаконичным по сравнению с Java, также с этим языком у Kotlin полная совместимость. Это позволит использовать библиотеки и фреймворки, написанные на Java. Новые библиотек также в первую очередь ориентированы не на Java, а на Kotlin. В него встроена null-безопасность, поэтому если присвоить значение null – во время компиляции просто произойдет сбой, что позволит уберечь от ошибки. Он тоже является объектно-ориентированным. Недостатком Kotlin является то, что у него ниже скорость компиляции, чем Java.

В качестве среды разработки на языке Kotlin выбор пал на Android Studio [8]. Данное решение было обусловлено несколькими факторами. Во-первых, компания, которая занимается поддержкой данной среды разработки, активно продвигает этот язык и регулярно выпускает обновления для его использования. Во-вторых, у Android Studio интуитивно понят-

ный редактор кода, который дает возможности сосредоточиться на решении поставленной задачи, а не на синтаксисе. В-третьих, среда разработки предоставляет богатый набор инструментов, которые позволяют решить большое количество задач: визуальный редактор макетов, эмулятор устройств.

В качестве системы управления базами данных был выбран MySQL [9]. MySQL – самая популярная система управления базами данных. Поддерживает большую часть функций, включая безопасность: установка привилегий пользователям, аутентификация и аудит, шифрование данных. Но в то же время MySQL может работать с большим объемом данных и хорошо масштабируется. Еще у нее высокая производительность, за счет пренебрежения некоторыми стандартами. Также у MySQL есть недостатки: присутствуют ограничения и некоторые операции реализованы не очень надежно.

Для реализации функционала мобильного приложения необходимо создать серверное приложение. Оно будет хранить базу данных, содержащую всю информацию о пользователях и ресторанах, и обрабатывать запросы приложения.

Для того, чтобы хранить данные о зарегистрированных пользователях была составлена таблица «User». Благодаря ней пользователи смогут заходить в систему под своими данным. Чтобы это осуществить в ней должны содержаться данные о почте, номере телефона и пароле.

Пользователь должен просматривать и выбирать ресторан, который он хочет посетить. Чтобы хранить данные о ресторанах, была составлена «Restaurant». Пользователю о ресторанах должно быть известно его название, адрес и телефон. Так как при бронировании необходимо производить оплату, а банковский счет у каждого ресторана свой, то эта информация тоже должна храниться в этой таблице. Еще у пользователя должна быть возможность видеть график работы ресторана, чтобы знать когда его можно посетить. На каждый день и у каждого ресторана может быть свой график, поэтому необходимо было создать отдельную таблицу «Schedul». В графике работы необходимо указать день, начало и конец работы. Любая строка данных об одном дне может принадлежать одному ресторану, поэтому эта таблица связывается с таблицей «Restaurant» по уникальному ключу ресторана. Также в ресторане есть много посадочных мест. Это говорит о том, что данные о них нужно хранить в отдельной таблице. Для этого была составлена таблица «Place». О конкретном месте пользователь может видеть цену бронирования и количество свободных мест. Для удобства ресторана у каждого места есть свое имя. Как уже говорилось выше, одно место может принадлежать строго одному ресторану, поэтому в этой таблице должен еще уникальный ключ ресторана.

Все блюда в ресторане для удобства поделены на категории. Для хранения названий категорий составлена отдельная таблица «Category_dish». В одной категории находятся несколько блюд. Для хранения информации о блюдах создана таблица «Dish». О блюде поль-

зователю может быть известно его название и цена. Поскольку с категорией блюд необходимо связать несколько позицией, то в таблице «Dish» используется уникальный ключ. Но блюдо принадлежит не только категории, но продается в конкретном ресторане, для этой связи используется уникальный ключ из таблицы «Restaurant». Пользователь также может смотреть состав конкретных блюд, которые состоят точно не из одного ингредиента. Для хранения этой информации создана таблица «Ingredient». В ней должно храниться название ингредиента и информация о том, можно ли из блюда убрать ингредиент, ведь не всегда это возможно. Под каждым блюдом пользователь может оставлять отзывы и оценки или читать их, для хранения этой информации создана таблица «Review». Чтобы знать к какому блюду какой отзыв принадлежит она содержит уникальный ключ блюда.

Про каждую бронь необходимо хранить достаточно много информации, поэтому была создана еще одна таблица «Reservation». При выборе места пользователь указывает дату и время. Необходимо знать какой пользователь сделал бронь, поэтому храниться его уникальный ключ. При выборе мест, чтобы разные пользователи не оформляли заказ на одно место, нужно ввести переменную, которая будет отвечать за состояние брони. При бронировании мест за один раз можно занимать несколько столов. Для этого потребуется вспомогательная таблица, которая будет хранить уникальный ключ брони и места, для их сопоставления.

Пользователь также может заказать несколько блюд в один заказ, поэтому чтобы знать в какой заказ какое блюдо входит, составлена таблица «Order_dish», которая хранит уникальные ключи блюда и брони. При заказе блюда пользователь может изменять его состав. Для хранения информации о том какое блюдо заказали и какой ингредиент был убран, тоже составлена отдельная таблица, которая хранит уникальные ключи заказанного блюда и ингредиента, от которого пользователь отказался.

При просмотре меню или заказах можно выделять свои любимые блюда. Чтобы их где-то хранить, создана таблица «Favorite_dish». Чтобы знать чье это блюдо в таблице должно храниться уникальный ключ пользователя и блюда. Пользователь блюда может сохранить наборами, для этого необходимо записывать их название. Также понравившиеся блюда можно редактировать, для этого аналогично создана отдельная таблица «Favorite_dish_editor», которая хранит в себе уникальные номера понравившегося блюда и ингредиента.

Сервер и база данных содержат секретную информацию и являются ценным ресурсом. Потеря кража или изменение которых может повлечь серьезные последствия как для пользователей, так и для ресторанов, например, финансовые потери или повреждение репутации. Сервер и база данных является уязвимой для различных атак, которые могут привести к несанкционированному доступу или нарушению работы сервера и приложения. Поэтому важно принимать меры предосторожности и обеспечить защиту. Защита – это неотъемлемая часть

разработки мобильного приложения и сервера для него, особенно, когда речь идет о заказах и денежных переводах пользователя ресторану. Для его защиты [10] могут подойти:

- надежные пароли,
- использование двухфакторной аутентификации,
- фаервол,
- VPN,
- антивирусное программное обеспечение,
- шифрование данных,
- не хранить секретную информацию в самом мобильном приложении,
- разграничение доступа к базе данных,
- резервное копирование,
- мониторинг активности.

Внедрение систем бронирования становится все более распространенным в различных отраслях экономики. Начиная от гостиниц и туризма, заканчивая арендой частных квартир и покупкой билетов на самолет или поезд. Общество привыкает к удобству и эффективности таких сервисов. Однако, в некоторых специфических случаях, таких как бронирование мест в ресторане, все еще существуют определенные сложности. В данной статье была разработана информационная система для выбора и бронирования мест в ресторанах, которая будет способствовать улучшению качества обслуживания и повышению удовлетворенности клиентов.

В ходе исследования были проанализированы существующие системы бронирования, выделены их достоинства и недостатки, которые послужили основой для определения функциональных требований к разрабатываемой системе. Особое внимание было уделено разработке мобильного приложения, которое будет обеспечивать удобный и интуитивно понятный интерфейс для пользователей. Кроме того, была разработана архитектура базы данных, которая будет использоваться для хранения информации о ресторанах, столиках и бронированиях.

Одним из ключевых моментов в разработке системы бронирования мест в ресторане является обеспечение безопасности и конфиденциальности данных клиентов. В связи с этим, в рамках работы были реализованы меры по защите сервера с базой данных от несанкционированного доступа и злоумышленных действий.

В целом, результаты данной работы могут быть использованы для дальнейшего развития и совершенствования систем бронирования мест в ресторанах, а также для повышения конкурентоспособности ресторанного бизнеса в целом.

Библиографический список

1. LeClick [Электронный ресурс]. URL: <https://leclick.ru/> (дата обращения: 25.05.2024).
2. ToМесто [Электронный ресурс]. URL: <https://tomesto.ru/perm/categories/restoran> (дата обращения: 25.05.2024).
3. Bourgeois Boohomians – BOBO [Электронный ресурс]. URL: <https://bobospb.ru/#> (дата обращения: 25.05.2024).
4. Архитектура системы: понятие, виды [Электронный ресурс]. URL: <https://gb.ru/blog/arhitektuta-sistemy/> (дата обращения: 25.05.2024).
5. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. СПб.: Питер, 2021. 352 с.
6. О чём «говорят» приложения: 5 популярных языков для мобильной разработки [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/yazyki-programmirovaniya-dlya-mobilnoy-razrabotki/#sovet-eksperta> (дата обращения: 25.05.2024).
7. TIOBE Index for June 2024 [Электронный ресурс]. URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 25.05.2024).
8. Android Studio: среда разработки мобильных приложений [Электронный ресурс]. URL: <https://arduinoplus.ru/android-studio/> (дата обращения: 25.05.2024).
9. SQLite, MySQL и PostgreSQL: Сравниваем популярные реляционные СУБД [Электронный ресурс]. URL: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison> (дата обращения: 25.05.2024).
10. Какие способы помогут защитить ваш сервер [Электронный ресурс]. URL: https://www.nic.ru/help/kakie-sposoby-pomogut-zashitit6-vash-server_11773.html (дата обращения: 25.05.2024).

References

1. LeClick [Electronic resource]. URL: <https://leclick.ru/> (date of access: 25.05.2024). (In Russ.)
2. ToMesto [Electronic resource]. URL: <https://tomesto.ru/perm/categories/restoran> (date of access: 05/25/2024). (In Russ.)
3. Bourgeois Boohomians – BOBO [Electronic resource]. URL: <https://bobospb.ru/#> (date of access: 05/25/2024). (In Russ.)
4. Arkhitektura sistemy: ponyatiye, vidy [Electronic resource]. URL: <https://gb.ru/blog/arhitektuta-sistemy/> (date of access: 05/25/2024). (In Russ.)
5. Martin R. Chistaya arkhitektura. Iskustvo razrabotki programmno obespecheniya. SPb.: Piter. 2021. 352 p. (In Russ.)

6. О чем «govoryat» prilozheniya: 5 populyarnykh yazykov dlya mobilnoy razrabotki [Electronic resource]. URL: <https://practicum.yandex.ru/blog/yazyki-programmirovaniya-dlya-mobilnoy-razrabotki/#sovet-eksperta> (date of access: 05/25/2024). (In Russ.)
7. TIOBE Index for June 2024 [Electronic resource]. URL: <https://www.tiobe.com/tiobe-index/> (date of access: 05/25/2024).
8. Android Studio: sreda razrabotki mobilnykh prilozheniy [Electronic resource]. URL: <https://arduinoplus.ru/android-studio/> (date of access: 05/25/2024).
9. SQLite. MySQL i PostgreSQL: Sravnivayem populyarnyye relyatsionnyye SUBD [Electronic resource]. URL: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison> (date of access: 05/25/2024).
10. Kakiye sposoby pomogut zashchitit vash server [Electronic resource]. URL: https://www.nic.ru/help/kakie-sposoby-pomogut-zashchitit6-vash-server_11773.html (date of access: 05/25/2024).

DESIGN AND DOCUMENTATION OF A RESTAURANT TABLE RESERVATION INFORMATION SYSTEM

Diyakova Irina A.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, irinadyakova.mat@yandex.ru

Stepanov Vladimir A.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, svaperm@gmail.com

This article discusses the process of designing and documenting an information system for restaurant table selection and reservation. A review of similar reservation systems was conducted, highlighting both their positive and negative aspects. The conclusions and observations drawn from this analysis served as the basis for defining the functional requirements of the system being developed. As part of the research, the architecture of the mobile application and the necessary software were presented. Particular attention was paid to the development of the database, which will be used in conjunction with the mobile application. An essential part of the study was the implementation of security measures for the server hosting the database, an aspect critical to modern information technology. The overall significance of this article lies in its practical applicability and potential benefit for the further development of a mobile application based on a restaurant table selection and reservation system.

Keywords: mobile application, reservation.

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ ВЫЯВЛЕНИЯ ФЕЙКОВЫХ ИЗОБРАЖЕНИЙ МЕТОДАМИ АНАЛИЗА УРОВНЕЙ СЖАТИЯ СВЕРТОЧНОЙ НЕЙРОСЕТЬЮ И АНАЛИЗА МЕТАДААННЫХ

Золотарев Данил Сергеевич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, zolotariov.danil@yandex.ru

Работа посвящена применению метода нейросетевого анализа уровней сжатия и метода анализа метаданных для выявления фейковых изображений. Целью является разработка приложения, которое позволит пользователю провести анализ изображения без необходимости владения навыками выявления фейковых изображений. Предметом исследования является выявление фейковых изображений посредством анализа уровней сжатия и метаданных. Объектом исследования является признак артефактов сжатия и параметры метаданных цифрового изображения. Решенные задачи – анализ литературы по теме, выбор технологий для разработки системы, сбор изображений для обучения нейросети, обучение нейросети, проектирование архитектуры и интерфейса приложения, разработка приложения. В статье рассматриваются методы анализа изображения на наличие следов подделки, датасеты изображений для обучения нейросети, обучение нейросети, проектирование и разработка приложения. В результате, приложение было разработано и выложено в открытый доступ.

Ключевые слова: нейронная сеть, изображения, уровни сжатия, метаданные, проектирование системы, разработка приложения.

Введение

В современном мире информационные технологии распространены повсеместно. Практически в каждом доме есть персональный компьютер с инструментами для обработки информации. Стоит уделить особое внимание цифровым изображениям, т.к. данный вид информации может быть доказательством какого-либо важного факта. Из-за того, что в открытом доступе существуют инструменты для редактирования изображений, каждый человек, имеющий персональный компьютер, может редактировать изображение по своему усмотрению. Изображения, которые были отредактированы, являются фейковыми и могут быть использованы для доказательства ложных фактов. Поэтому необходим такой инструмент, который позволил бы убедиться в подлинности изображения.

Постановка задачи

Была поставлена цель – разработать приложение для проведения анализа изображений. Для достижения цели были выполнены следующие задачи:

1. Определен датасет для обучения нейросети;
2. Обучена нейросеть;
3. Разработано приложение.

Методы анализа изображения

EXIF (Exchangeable Image File Format) — это стандартный формат метаданных, который используется для хранения дополнительной информации о цифровых изображениях. Для анализа будут использованы такие данные как дата создания, дата изменения и программное обеспечение, которое в последний раз обрабатывало изображение.

ELA (Error Level Analysis) [1] — это метод анализа цифровых изображений, который позволяет обнаружить разницу в уровнях сжатия изображения. При наличии такой разницы можно сказать, что изображения подделано. У метода есть ограничение, он работает только с изображением того формата, который сохраняется с потерями в качестве, например jpg.

CNN (Convolutional Neural Network) – это специализированный тип нейронной сети, который используется для анализа изображений. Суть модели CNN заключается в том, что используются сверточные слои для извлечения признаков из изображений. Сверточные слои применяют фильтры к различным областям, чтобы выделить признак, а именно разницу уровней сжатия.

Результаты

Была обучена нейросеть на датасете Casia [2] и получена точность предсказания 89%, а показатель функции потерь 0,28. Процесс обучения изображен на рис. 1. Обученная модель нейросети была использована в разработанном приложении.

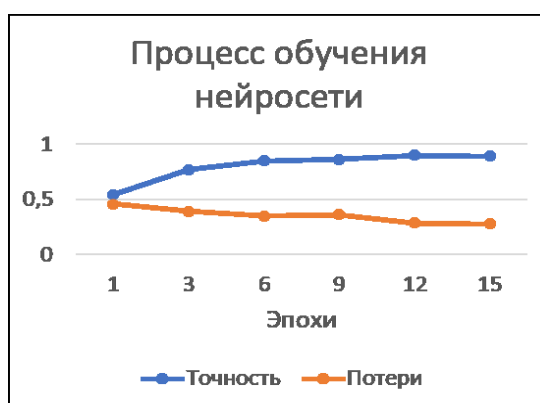


Рис. 1. Процесс обучения нейросети на датасете Casia

Было разработано настольное приложение для Windows. Интерфейс приложения изображен на рис. 2.

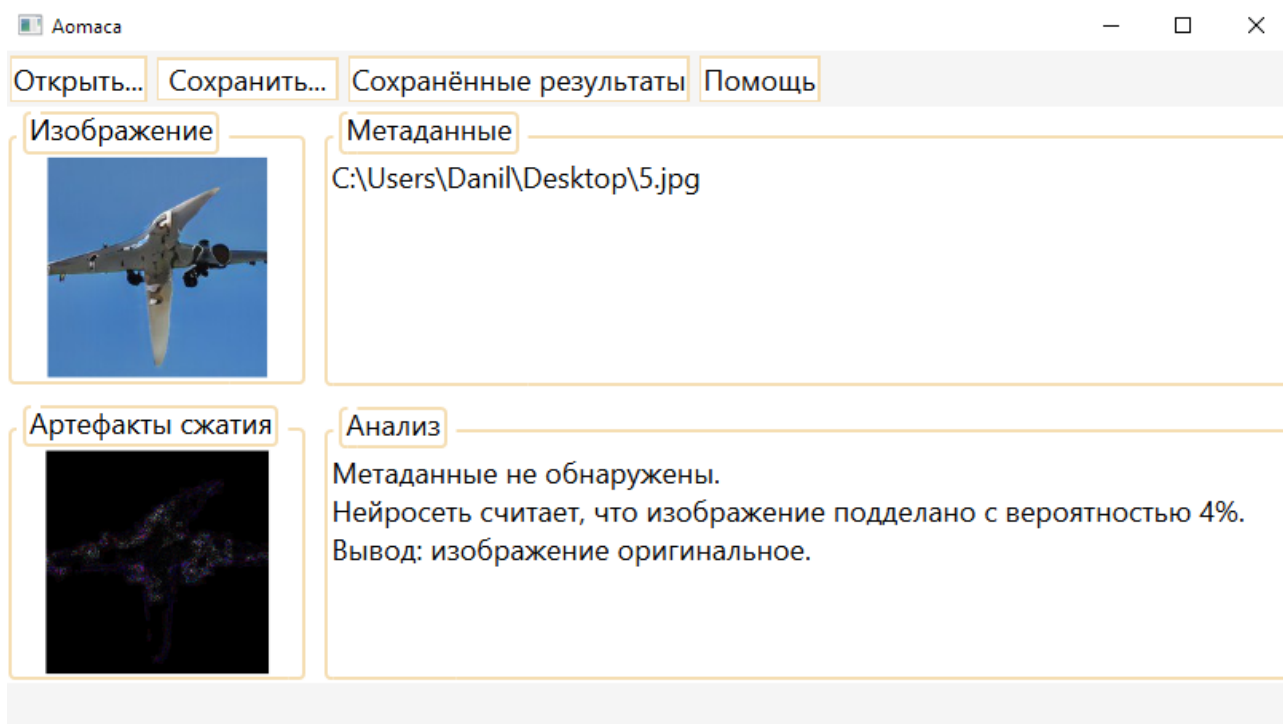


Рис. 2. Интерфейс разработанного приложения для анализа изображений

Для разработанного приложения был создан установщик и выложен в открытый доступ [3].

Заключение

Нейросеть была обучена с точностью предсказаний 89%, но данная точность справедлива только для изображений из датасета Casia, для изображений из сети Интернет данная точность не гарантируется. Приложение с пользовательским интерфейсом для нейросетевого анализа уровней сжатия и анализа метаданных было разработано. Приложение выложено в открытый доступ для скачивания. Данным комплексом может воспользоваться обычный пользователь без специальных навыков. А значит, поставленная цель была достигнута.

Библиографический список

1. Wang W., Dong J., Tan T. Tampered Region Localization of Digital Color Images Based on JPEG Compression Noise // Digital Watermarking: 9th International Workshop, IWDW 2010. Seoul, Korea. Pp. 120–133.
2. Casia dataset, Kaggle [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/sophatvathana/casia-dataset> (дата обращения 04.03.2024).
3. Aomaca, disk.yandex [Электронный ресурс]. URL: <https://disk.yandex.ru/d/tiXli0FQkei7hw> (дата обращения 26.05.2024).

References

1. Wang W., Dong J., Tan T. Tampered Region Localization of Digital Color Images Based on JPEG Compression Noise // Digital Watermarking: 9th International Workshop, IWDW 2010. Seoul, Korea. Pp. 120–133.
2. Casia dataset, Kaggle [Electronic resource]. URL: <https://www.kaggle.com/datasets/sophatvathana/casia-dataset> (date of access: 03/04/2024).
3. Aomaca, disk.yandex [Electronic resource]. URL: <https://disk.yandex.ru/d/tiXlI0FQkei7hw> (date of access: 05/26/2024).

DESIGN AND DEVELOPMENT OF A SYSTEM FOR DETECTING FAKE IMAGES USING METHODS OF ANALYSIS OF COMPRESSION LEVELS BY A CONVOLUTIONAL NEURAL NETWORK AND METADATA ANALYSIS

Zolotarev Danil S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, zolotariov.danil@yandex.ru

The work is devoted to the application of the method of neural network analysis of compression levels and the method of metadata analysis to identify fake images. The goal is to develop an application that will allow the user to analyze an image without having to have the skills to identify fake images. The subject of the study is the identification of fake images by analyzing compression levels and metadata. The object of the study is the characterization of compression artifacts and metadata parameters of a digital image. The tasks to be solved are the analysis of literature on the topic, the choice of technologies for system development, image collection for neural network training, neural network training, application architecture and interface design, application development. The article discusses methods of image analysis for detecting traces of forgery, image datasets for training neural networks, training neural networks, application design and development. As a result, the application was developed and became publicly available.

Keywords: neural network, images, error levels, metadata.

ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МЕТОДОВ ТЕОРИИ ПЕРКОЛЯЦИИ ДЛЯ ИССЛЕДОВАНИЯ КРУПНОМАСШТАБНОЙ СТРУКТУРЫ ВСЕЛЕННОЙ

Караваева Ксения Алексеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, kk.ksenia1@yandex.ru

В статье рассматривается возможность применения методов теории перколяции для изучения крупномасштабной структуры Вселенной. В работе представлено описание основных космологических моделей и определений, а также обсуждается современное состояние исследований, методы и математические подходы в данной области. Основное внимание уделено исследованию крупномасштабной структуры Вселенной с использованием методов теории перколяции. В результате исследования была разработана модель крупномасштабной структуры Вселенной с помощью теории перколяции и компьютерного моделирования, представляющая собой ценный инструмент для дальнейших исследований в области космологии. У модели есть ряд параметров, которые можно изменять для изучения ее поведения и свойств.

Ключевые слова: теория перколяции, перколяция, космология, космологические модели, крупномасштабная структура Вселенной, компьютерное моделирование, математическое моделирование.

Крупномасштабная структура Вселенной – один из наиболее сложных и фундаментальных вопросов современной космологии. Несмотря на множество исследований, многие аспекты формирования и эволюции этих структур до сих пор остаются недостаточно изученными. Актуальным является разработка новых моделей и методов изучения структуры Вселенной с целью прогнозирования ее дальнейшего поведения. Галактики взаимодействуют между собой благодаря гравитационным силам. Гравитационное взаимодействие приводит к тому, что галактики притягиваются друг к другу и образуют различные скопления и сверхскопления. Однако, помимо гравитационных сил, галактики могут подвергаться влиянию других процессов, таких как взаимодействие с темной материей, тепловые процессы, радиационное давление и другие.

Теория перколяции – это математический метод, который используется для изучения случайных сетей, включая распределение объектов в пространстве. В контексте исследования галактик теория перколяции может быть применена для изучения распределения галактик во Вселенной и определения структуры галактических кластеров [1].

Теория перколяции может быть применима для изучения иерархической модели Вселенной, так как теория перколяции применяется для изучения структурных свойств системы, состоящей из

множества взаимодействующих элементов, где возможно возникновение крупных кластеров или связанных участков [2]. Из-за своей способности моделировать процессы образования крупных структурных образований, теория перколяции действительно может быть полезной в исследовании крупномасштабной структуры Вселенной. Теория перколяции изучает теорию вероятности на графах, что может быть использовано в изучении процессов формирования и эволюции космической сети – совокупности галактик и их скоплений, распределённых в пространстве в виде сложной сети, где узлами служат скопления галактик, а связями – филаменты тёмной материи [3, 4]. В данной работе исследуется структура сформированного графа, в зависимости от связности, которая задается с помощью специального порога [5, 6]. Таким образом, основной целью исследования является смоделировать крупномасштабную структуру Вселенной, применяя методы теории перколяции.

Предложена перколяционная модель крупномасштабной структуры Вселенной. Для модели вводятся следующие обозначения и общие определения. Первичные материальные объекты – галактики, кластеры – скопления галактик. Входные параметры модели:

- `num_galaxies` – целое число равное количеству моделируемых галактик,
- `galaxies_masses` – массив масс для каждой галактики,
- `galaxies_position` – размер области для построения галактической структуры,
- `force_threshold` – порог расстояния для «связи» между галактиками,
- `mass` – средняя масса галактики (в кг),
- `radius` – радиус окружности, внутри которой формируются скопления галактик,
- `G` – гравитационная постоянная (в $\text{H}^*\text{m}^2/\text{кг}^2$).

Галактические нити лежат между войдами. Галактики с большей массой притягивают галактики с меньшей массой, образуя скопления – это будет продемонстрировано в данной модели. Также использован закон Всемирного тяготения Ньютона. Структура Вселенной может быть представлена в виде графа, где узлы представляют галактики, а ребра представляют гравитационные силы между ними. Также в модели учитывается масса галактики, в зависимости от нее определяется размер вершины графа – размер точки на рисунке. Имеется возможность регулировать масштаб, смотреть номер и массу каждой галактики. Это может понадобиться для дальнейших исследований. Таким образом, методами компьютерного моделирования и теории перколяции, возможно получение графа, который представляет собой крупномасштабную структуру Вселенной.

Было проведено моделирование крупномасштабной структуры вселенной с разными наборами значений входных параметров. Позиции галактик могут быть смоделированы в соответствии с нормальным распределением, а также с равномерным. Таким образом, позиции точек при нормальном распределении должны «собираться» в центре, а при равномерном –

однородно. Связность между галактиками можно регулировать с помощью изменения параметра `force_threshold`. Чем он больше – тем меньше связей будут появляться на графе, соответственно, мелких кластеров будет образовано больше. Также можно менять количество галактик. Был проведен ряд компьютерных экспериментов для следующих входных параметров: $mass = 6 \cdot 10^{42}$, $radius = 5 \cdot 3.09 \cdot 10^{22}$, $G = 6.67430 \cdot 10^{-11}$, $num_galaxies = 1000$. Позиции галактик имеют нормальное распределение. Для `force_threshold` с $2 \cdot 10^{29}$ до $5 \cdot 10^{31}$ вид структуры вселенной представлен на рис. 1.

Для `force_threshold` с $7 \cdot 10^{31}$ до $3 \cdot 10^{31}$ вид структуры вселенной представлен на рис. 2.

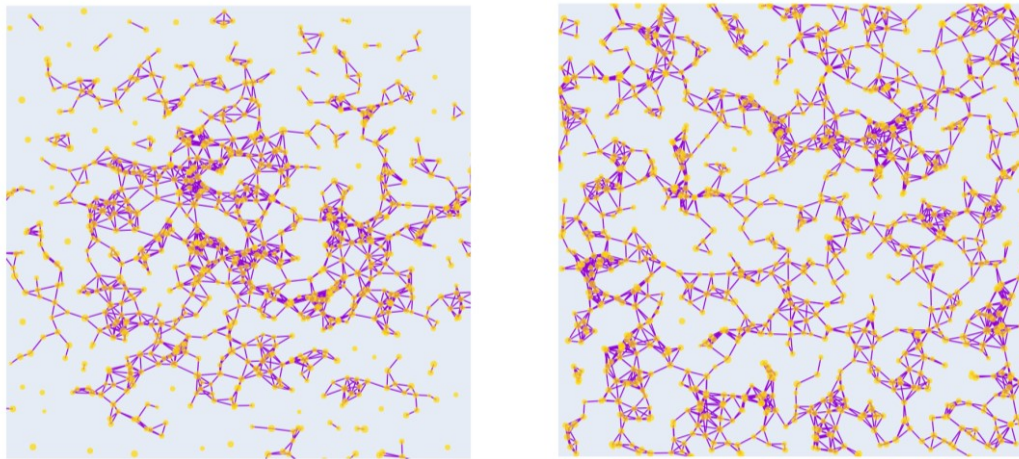


Рис. 1. Вид структуры вселенной `force_threshold` с $2 \cdot 10^{29}$ до $5 \cdot 10^{31}$

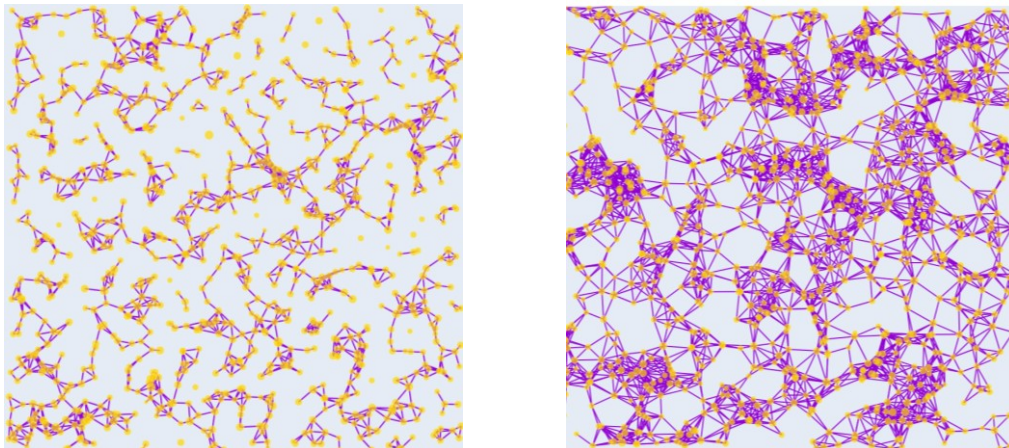


Рис. 2. Вид структуры вселенной `force_threshold` с $7 \cdot 10^{31}$ до $3 \cdot 10^{31}$

Для входных параметров $mass = 6 \cdot 10^{42}$, $radius = 5 \cdot 3.09 \cdot 10^{22}$, $G = 6.67430 \cdot 10^{-11}$, `force_threshold` = $3 \cdot 10^{31}$ и при изменении $num_galaxies$ с 100 до 500 вид структуры вселенной представлен на рис. 3.

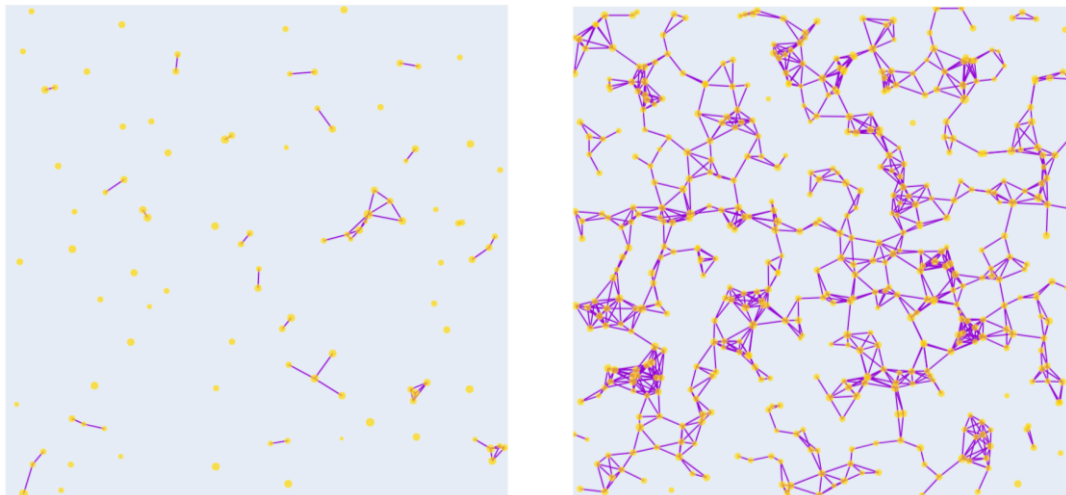


Рис. 3. Вид структуры вселенной $\text{force_threshold} = 3 \cdot 10^{31}$ и при изменении num_galaxies с 100 до 500

Таким образом, на рисунках видно, что предложенная перколяционная модель адекватно описывает модель крупномасштабной структуры Вселенной. В дальнейшем автором настоящей работы планируется исследование возможности применения разработанной модели (или ее модификации) для прогнозирования поведения Вселенной, в частности ее расширения.

Библиографический список

1. Космологические модели. URL: https://ru.wikipedia.org/wiki/Космологические_модели (дата обращения: 23.04.2024).
2. Теория перколяции. URL: https://ru.wikipedia.org/wiki/Теория_перколяции (дата обращения: 23.04.2024).
3. Schulman L. S., Seiden P. E. Percolation and Galaxies // Science. 1986. Vol. 233 (4762). P. 425–431.
4. Bhavsar S. P., Barrow J. D. Percolation analyses of observed and simulated galaxy clustering // Monthly Notices of the Royal Astronomical Society. 1983. Vol. 205. No 1. P. 61–66. URL: <https://doi.org/10.1093/mnras/205.1.61P> (дата обращения: 23.04.2024).
5. Абди Р., Ерохин М. Измерение характеристик алгоритма моделирования перколяции. URL: <https://metrology-bg.org/fulltextpapers/212.pdf> (дата обращения: 23.04.2024).
6. Тарасевич Ю. Ю. Перколяция: теория, приложения, алгоритмы: учеб. пособие. М.: Едиториал УРСС, 2002. 200 с.

References

1. Cosmological models. URL: https://ru.wikipedia.org/wiki/Cosmological_models (date of access: 04/23/2024). (In Russ.)

2. Zasov A. V., Postnov K. A. General astrophysics. Fryazino: Century 2, 2006. 496 p. (In Russ.)
3. Schulman L. S., Seiden P. E. Percolation and Galaxies // Science. 1986. Vol. 233 (4762). P. 425–431.
4. Bhavsar S. P., Barrow J. D. Percolation analyses of observed and simulated galaxy clustering // Monthly Notices of the Royal Astronomical Society. 1983. Vol. 205. No 1. P. 61–66. URL: <https://doi.org/10.1093/mnras/205.1.61P> (date of access: 23.04.2024).
5. Abdi R., Erokhin M. Measuring the characteristics of a percolation modeling algorithm. URL: <https://metrology.bg.org/fulltextpapers/212.pdf> (date of access: 04/23/2024). (In Russ.)
6. Tarasevich Yu. Yu. Percolation: theory, applications, algorithms: textbook. M.: Unified URSS, 2002. 200 p. (In Russ.)

POSSIBILITIES OF USING PERCOLATION THEORY METHODS TO STUDY THE LARGE-SCALE STRUCTURE OF THE UNIVERSE

Karavaeva Ksenia Alekseevna

Perm State University, 15, Bukireva st., Perm, 614068, Russia, kk.ksenia1@yandex.ru

The article considers the possibility of applying percolation theory methods to study the large-scale structure of the universe. The paper describes the main cosmological models and definitions, as well as discusses the current state of research, methods and mathematical approaches in this field. The main attention is paid to the study of the large-scale structure of the universe using percolation theory methods. As a result of the research, a model of the large-scale structure of the Universe was developed using percolation theory and computer modeling, which is a valuable tool for further research in the field of cosmology. The model has a number of parameters that can be changed to study its behavior and properties.

Keywords: percolation theory, percolation, cosmology, cosmological models, large-scale structure of the Universe, computer modeling, mathematical modeling.

ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ОНЛАЙН-СЕРВИСА С API ДЛЯ ЗАПОЛНЕНИЯ ДОКУМЕНТОВ ПО ШАБЛОНАМ

Колпащиков Максим Николаевич, Кнутова Наталия Сергеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, mxmblock@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 классов и компонентов системы, схему реляционной базы данных. На основании построенных моделей реализована и развёрнута система, написана документация.

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 персо-

нальных 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 по шаблонам 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 информационной системы, предназначенной для формирования документов по шаблонам (или иначе — системы заполнения шаблонов).

Данная работа использует ранее полученные результаты [2], расширяет и дополняет их.

Общий анализ программ заполнения шаблонов

Концептуально работа программы заполнения шаблонов сводится к тому, чтобы на основе 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 данной ситуации или к объекту действительности.

Если детализировать эту модель, то содержательные данные можно представлять как разнородную 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 набора значений с набором полей формы и их слиянию. Поскольку данные

имеют различную природу, сопоставлению может предшествовать этап преобразования, в ходе которого осуществляется приведение данных к однородному виду.

Результаты анализа существующих решений

В рамках данной работы было рассмотрено несколько различных систем, реализующих заполнение шаблонов: R6R [3], Eforms [4], DocWebService [5], Aspose Words Assembly [6], DocAssemble [7], Microsoft Word, LibreOffice Writer, Модуль заполнения шаблонов 1С: Документооборот.

Ниже приведена сводная классификация рассмотренных систем по разным признакам. В скобках указано число систем, попадающих в каждый класс, из общего числа рассмотренных систем.

По способу подготовки шаблона: вручную (6 из 8) или автоматически (2 из 8).

По способу заполнения: одна форма (2 из 8), серия форм, интервью (2 из 8), подготовка данных заполнения в структурированном формате (3 из 8), автоматически, без участия пользователя (1 из 8).

По наличию пакетного режима работы: поддерживается (6 из 8), не поддерживается (2 из 8).

По возможности настройки свойств полей: присутствует (5 из 8), отсутствует (3 из 8).

По наличию программного интерфейса: присутствует (5 из 8), отсутствует (3 из 8).

Наиболее распространенным поддерживаемым форматом документов является Office Open XML.

На основании вышеизложенного, можно сформулировать требования к новому решению.

Система должна: быть доступна как для личного использования, так и для развёртывания на предприятии; предлагать пользователям возможность сохранения шаблонов для повторного использования; поддерживать обработку пользовательских шаблонов; поддерживать формирование документов в пакетном режиме; включать гибкий механизм преобразования и проверки данных заполнения; поддерживать работу с документами в формате Office Open XML, как с наиболее распространенным форматом описания форматированных документов; быть выполнена в виде веб-приложения, иметь дружелюбный пользовательский интерфейс, а также базовый программный интерфейс, за счёт которого будет возможна интеграция с внешними системами.

Объектное моделирование системы

Для описания модели системы был выбран язык UML и построен ряд диаграмм.

Диаграмма прецедентов призвана отразить функциональность системы в наиболее обобщённом виде. Диаграмма прецедентов изображена на рис. 1.

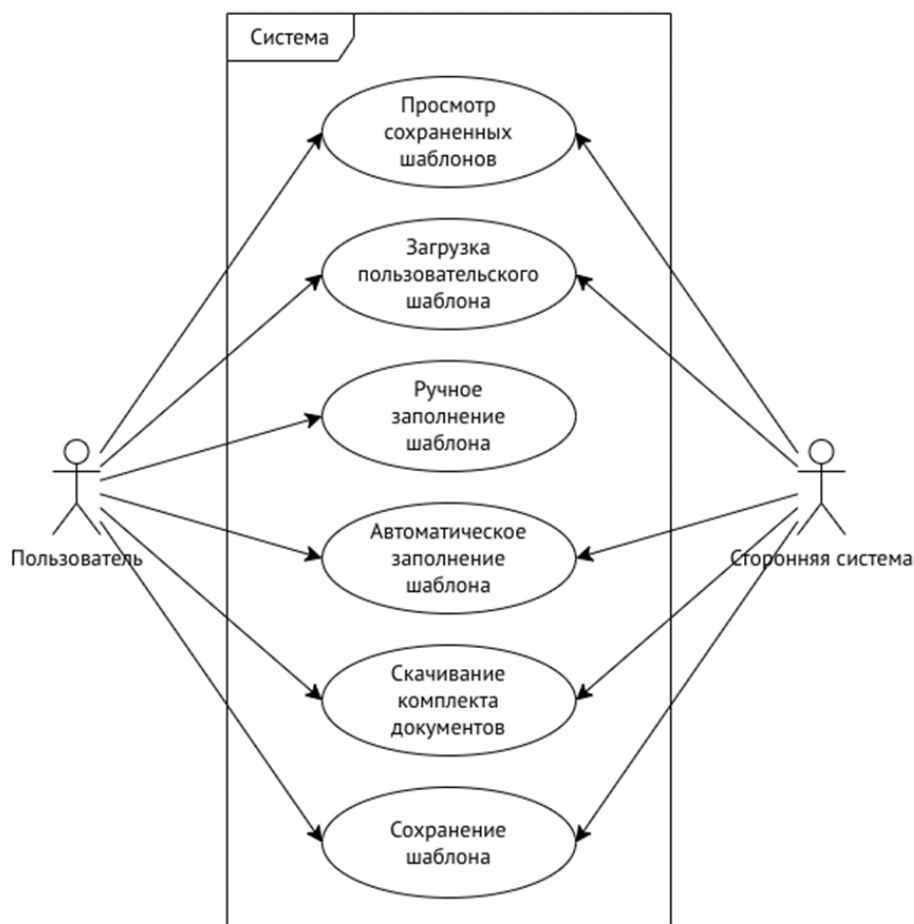


Рис. 1. Диаграмма вариантов использования

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 сократить количество операций обмена данными с сервером.

Наконец, система осуществляет заполнение шаблона, вследствие чего формируется комплект результирующих документов, который предоставляется для скачивания.

Диаграмма классов позволяет отразить структуру используемых классов и взаимосвязей между ними. На рисунке 2 представлена диаграмма классов для серверной части системы. Клиентская часть не использует объектно-ориентированный подход и не требует реализации данного вида диаграммы.

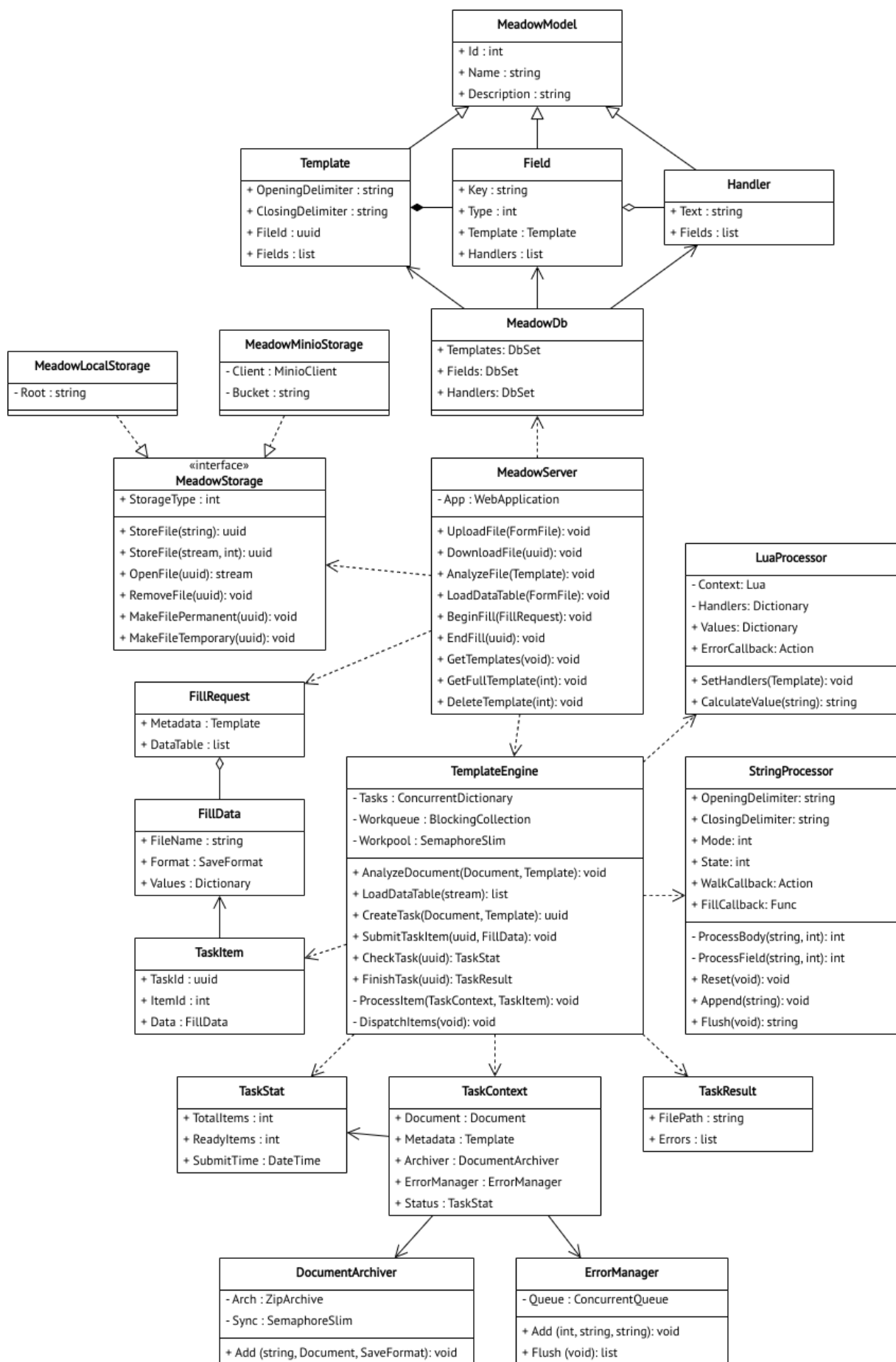


Рис. 2. Диаграмма классов

Все классы, входящие в представленную иерархию, можно разделить на две группы: модели и функциональные модули.

Модели – это классы, не обладающие поведением (не содержащие каких-либо методов). Модели используются для инкапсуляции свойств, описывающих какой-либо объект, с которым работают функциональные модули.

Функциональный модуль – это класс, реализующий какую-либо функцию в системе. Некоторые функциональные модули являются отражением описанных ранее компонентов. Например, класс `MeadowServer` соответствует компоненту «сервер API». Кроме того, система содержит вспомогательные функциональные модули, которые необходимы для работы основных компонентов. Например, класс `TemplateEngine`, который соответствует компоненту «менеджер задач», использует два вспомогательных класса: `StringProcessor` и `LuaProcessor`. Подробное описание всех функциональных модулей системы и соответствующих классов приведено в следующей главе данной работы.

Для работы с базой данных система использует `Entity Framework Core` и реализует интерфейсный класс `MeadowDb`. Данный класс представляет собой контекст, свойства которого автоматически отображаются на таблицы базы данных. Сущности реляционной модели, которая описана в следующем пункте данной главы, отображаются на строго типизированные классы, составляющие объектную модель системы.

Диаграмма компонентов позволяет взглянуть на структурную организацию системы. Принципиально в системе следует выделить два компонента: клиентскую и серверную части.

Клиентская часть реализует пользовательский интерфейс системы и состоит из экранов – веб-страниц с определенным набором элементов управления. Серверная часть включает в себя основные функциональные компоненты системы и реализует её программный интерфейс. Клиент-серверное взаимодействие организуется через три интерфейса: управление шаблонами, управление файлами и управление задачами.

Рассмотрим кратко экраны пользовательского интерфейса.

Стартовый экран предназначен для отображения списка сохраненных шаблонов, а также для загрузки пользовательского шаблона. Для выполнения этих функций экран использует интерфейсы управления шаблонами и управления файлами соответственно.

Экран настройки предназначен для установки метаданных шаблона. Предполагается, что данный экран будет вызываться сразу после загрузки пользовательского шаблона.

Экран заполнения предлагает визуальную форму для заполнения шаблона. Данный экран использует интерфейс управления задачами для того, чтобы инициировать процесс формирования комплекта документов на сервере.

Экран загрузки результата предназначен для получения результата заполнения с сервера через интерфейс управления файлами.

Далее рассмотрим компоненты, составляющие серверную часть системы.

За обработку всех входящих запросов отвечает сервер API. Его главной задачей является внутренняя маршрутизация данных и делегирование запросов другим компонентам системы. Так, запросы к интерфейсу управления шаблонами связаны с обращениями к базе данных и обрабатываются соответствующим компонентом.

Для хранения файлов-шаблонов и файлов-результатов система использует выделенное хранилище, которое также отвечает за обработку запросов к интерфейсу управления файлами.

Наконец, менеджер задач реализует основной функционал системы, связанный с заполнением шаблонов, а также обрабатывает запросы к соответствующему интерфейсу.

Полностью диаграмма компонентов представлена на рис. 3.



Рис. 3. Диаграмма компонентов

Как уже было сказано выше, одним из компонентов проектируемой системы является реляционная база данных. Роль базы данных в системе заключается в хранении метаданных о шаблонах. Сами шаблоны представляют собой обычные файлы и сохраняются в хранилище системы.

Реализация системы

Система использует библиотеку Aspose для работы с документами в формате Office Open XML. Серверная часть системы реализована на основе платформы Microsoft .NET и сопутствующего стека технологий, включая веб-фреймворк ASP.NET Core [8].

Клиентская часть системы написана с использованием HTML, CSS и JavaScript. Для написания программной части применялась минималистичная библиотека jQuery [9].

Для построения визуально привлекательного пользовательского интерфейса использовалась библиотека Bootstrap [10].

На самом нижнем уровне система оперирует так называемыми шаблонными строками. Шаблонная строка представляет собой расширение понятия строки. Формальное определение шаблонной строки представлено на рис. 4.

```
<шаблонная строка> ::= <строка> | <строка> <поле> <шаблонная строка>  
<поле> ::= <открывающий разделитель> <имя поля> <закрывающий разделитель>  
<имя поля> ::= <непустая строка>  
<открывающий разделитель> ::= <непустая строка>  
<закрывающий разделитель> ::= <непустая строка>
```

Рис. 4. Определение шаблонной строки в нотации BNF

Основная операция, применимая к шаблонной строке, это подстановка – замена содержащегося в ней поля на обычную строку по некоторому правилу. Серия подстановок может привести к полному удалению из шаблонной строки её переменной части, в результате чего шаблонная строка превращается в обычную.

Строковый процессор (StringProcessor) – это компонент рассматриваемой системы, который предназначен для обработки потока шаблонных строк. Для работы строковому процессору необходима информация о разделителях, используемых в данной шаблонной строке. Архитектура процессора позволяет обновлять эту информацию перед обработкой очередной строки из входящего потока.

Система использует микропрограммы (обработчики, LuaProcessor) на языке Lua для придания дополнительных свойств полям шаблона. Механизм обработчиков предлагает гибкий способ настройки правил проверки и преобразования значений полей, а использование языка Lua способствует облегчению освоения этого механизма.

Архиватор документов (DocumentArchiver) – это один из двух вспомогательных классов, предназначенных для организации работы ядра системы в условиях параллелизма. Все публичные члены этого класса являются потокобезопасными, если не указано иное. Назначение архиватора состоит в том, чтобы сохранять заполненные документы в заданном формате в общий архив ZIP.

Менеджер ошибок (ErrorManager) является ещё одним вспомогательным классом ядра. Его назначение состоит в потокобезопасном централизованном сборе ошибок заполнения от нескольких экземпляров процессора Lua.

Ядро (TemplateEngine) – это компонент системы, реализующий основную её функциональность. Ядро представляет собой многоканальную систему массового обслуживания с бесконечной очередью. Механизм его работы основан на управлении задачами, поэтому в тексте данной работы, а также в сопроводительной документации к системе, этот компонент может называться «менеджером задач».

Задача – это зарегистрированный в ядре запрос на заполнение комплекта документов.

С каждой задачей всегда связан по крайней мере один предмет. Если задача описывает для ядра комплект документов в целом, то каждый предмет описывает ровно один документ из этого комплекта.

Для организации работы с этими артефактами в системе предусмотрено хранилище. Хранилище предоставляет методы для сохранения, удаления, изменения файлов. Кроме того, хранилище реализует механизм самоочистки. В целях предотвращения возможных утечек, временные файлы автоматически удаляются из хранилища по истечении некоторого времени с момента их сохранения.

Система поддерживает интеграцию с объектным хранилищем MinIO [11] и позволяет использовать его как файловое хранилище. В основе реализации данного провайдера лежит оригинальный API MinIO, обеспечивающий взаимодействие по протоколу Amazon S3 [12].

Серверная часть системы реализует доступ к своим функциональным компонентам посредством веб-API. Такой подход позволяет единообразно организовать взаимодействие как с клиентской частью системы, так и с внешними системами.

Клиентская часть системы представляет собой независимый компонент. Следствием независимости клиентской части является ряд удобных архитектурных особенностей: клиент-серверное взаимодействие в системе происходит с использованием того же API, что и при взаимодействии с внешними системами; серверная часть является самодостаточной и может при необходимости работать без пользовательского интерфейса; предлагаемая реализация пользовательского интерфейса может быть заменена на другую, использующую тот же API.

Пользовательский интерфейс системы состоит из 4 веб-страниц: домашняя страница, страница настройки шаблона, страница заполнения шаблона и страница загрузки результата.

На рисунке 5 представлена иллюстрация страницы заполнения шаблона.

Мeadow Приказ о назначении ответственного по охране труда

✓ Завершить ↓ Сохранить ✎ Настроить ✕ Отменить

Сделайте нужные настройки

Придумайте имя для документа

Приказ

Символ # будет заменён на номер экземпляра документа

Выберите желаемый формат документа

Документ PDF (.pdf)

Не все преобразования форматов допустимы, попробуйте разные варианты

Загрузите данные из таблицы

Выберите файл Файл не выбран Отправить

Документы будут автоматически созданы на основе этих данных

Создайте несколько экземпляров документа

+ Добавить - Удалить ← Предыдущий → Следующий

Все документы будут сохранены в выбранном формате и упакованы в архив

Заполните форму - Новый экземпляр

Наименование организации

АНО ДПО "Учебный центр "ШИФТ"

Должность ответственного

системный инженер

ФИО ответственного

Колпациков М.Н.

Полное наименование организации

АНО ДПО "Учебный центр "ШИФТ"

Номер документа

123

Дата составления

15.05.2024

Рис. 5. Страница заполнения шаблона

Заключение

В рамках настоящей работы был решён ряд задач.

Во-первых, была изучена предметная область, связанная с обработкой документов и их шаблонов. Были рассмотрены существующие решения разного уровня и предназначенные для различных сегментов пользователей. На основе сравнения существующих решений были выдвинуты требования к разрабатываемому решению.

Во-вторых, был проведен обзор инструментальных средств проектирования и разработки, результатом которого послужил выбор стека технологий, который затем был задействован при проектировании и разработке решения.

В-третьих, была построена модель разрабатываемой системы в виде нескольких диаграмм на языке UML, описывающих систему с разных сторон и на разных уровнях.

Наконец, была проведена разработка системы с использованием приведённых выше средств. Структура и принцип действия всех функциональных элементов системы описаны в рамках этой работы.

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. Колпашиков М. Н., Кнутова Н. С. Проектирование онлайн-сервиса с API для заполнения документов по шаблонам // Актуальные проблемы математики, механики и информатики 2023 [Электронный ресурс]. Пермь: ПГНИУ, 2023. С. 178–187. URL: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematiki-mekhaniki-informatiki-2023.pdf> (дата обращения: 15.05.2024).
3. R6R.ru [Электронный ресурс]. URL: <https://r6r.ru> (дата обращения: 12.05.2024).
4. eForms [Электронный ресурс]. URL: <https://eforms.com/> (дата обращения: 12.05.2024).
5. DocWebService [Электронный ресурс]. URL: <https://docwebservice.ru/> (дата обращения: 12.05.2024).
6. Aspose Words Assembly [Электронный ресурс]. URL: <https://products.aspose.app/words/ru/assembly> (дата обращения: 12.05.2024).
7. Docassemble [Электронный ресурс]. URL: <https://docassemble.org/> (дата обращения: 12.05.2024).
8. ASP.NET Core [Электронный ресурс]. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата обращения: 12.05.2024).

9. jQuery [Электронный ресурс]. URL: <https://jquery.com/> (дата обращения: 12.05.2024).
10. Bootstrap [Электронный ресурс]. URL: <https://getbootstrap.com/> (дата обращения: 12.05.2024).
11. MinIO .NET Client API Reference [Электронный ресурс]. URL: <https://min.io/docs/minio/linux/developers/dotnet/API.html> (дата обращения: 15.05.2024).
12. Amazon Simple Storage Service [Электронный ресурс]. URL: <https://docs.aws.amazon.com/AmazonS3/latest/userguide//Welcome.html> (дата обращения: 14.05.2024).

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. Kolpashchikov M. N., Knutova N. S. Designing online service with API for filling out documents from templates: Current problems of mathematics, mechanics and computer science 2023 [Electronic resource]. Perm: PSU, 2023. Pp. 178–187. URL: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematiki-mekhaniki-informatiki-2023.pdf> (date of access: 05/15/2024). (In Russ.)
3. R6R.ru [Electronic resource]. URL: <https://r6r.ru> (date of access: 05/12/2024). (In Russ.)
4. eForms [Electronic resource]. URL: <https://eforms.com/> (date of access: 05/12/2024).
5. DocWebService [Electronic resource]. URL: <https://docwebservice.ru/> (date of access: 05/12/2024). (In Russ.)
6. Aspose Words Assembly [Electronic resource]. URL: <https://products.aspose.app/words/ru/assembly> (date of access: 05/12/2024).
7. Docassemble [Electronic resource]. URL: <https://docassemble.org/> (date of access: 05/12/2024).
8. ASP.NET Core [Electronic resource]. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (date of access: 05/12/2024).
9. jQuery [Electronic resource]. URL: <https://jquery.com/> (date of access: 05/12/2024).
10. Bootstrap [Electronic resource]. URL: <https://getbootstrap.com/> (date of access: 05/12/2024).
11. MinIO .NET Client API Reference [Electronic resource]. URL: <https://min.io/docs/minio/linux/developers/dotnet/API.html> (date of access: 05/15/2024).
12. Amazon Simple Storage Service [Electronic resource]. URL: <https://docs.aws.amazon.com/AmazonS3/latest/userguide//Welcome.html> (date of access: 05/14/2024).

DESIGNING, DEVELOPING AND DOCUMENTING ONLINE SERVICE WITH API FOR FILLING OUT DOCUMENTS FROM TEMPLATES

Kolpashchikov Maxim N., Knutova Nataliya S.

Perm State University, 15, Bukireva st., Perm, 614068, 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 classes and all system components, a relational database schema. Based on the constructed models, the system was developed and deployed, and documentation was written.

Keywords: filling out documents according to a template, document management, online service, API.

ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ANDROID-ПРИЛОЖЕНИЯ ДЛЯ КОНТРОЛЯ ЛИЧНЫХ ФИНАНСОВ И КУРСА ВАЛЮТ

Костицын Владислав Витальевич, Кнутова Наталья Сергеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, kosticyn.vladislav@gmail.com

С развитием технологий, управление личными финансами становится проще благодаря мобильным приложениям. Они помогают отслеживать доходы и расходы, планировать бюджет для финансовой стабильности. В условиях изменчивых валютных курсов, приложения для отслеживания и анализа валютных курсов становятся все более востребованными. Они предоставляют информацию о текущих курсах, позволяют устанавливать оповещения о изменениях, что помогает принимать финансовые решения. В статье рассматривается разработка и обеспечение безопасности данных мобильного приложения, а также процесс публикации в RuStore. Вместе с тем будут рассмотрены клиентская и серверная часть приложения.

Ключевые слова: разработка, приложение Android, разработка приложения.

В настоящее время наблюдается стремительное развитие мобильных технологий, что приводит к возрастающему интересу к мобильным приложениям и поиску более простых решений повседневных задач. Одной из таких задач является управление личными финансами, которое является неотъемлемой частью жизни современного человека. Кроме того, появление цифровых валют и необходимость управления цифровыми активами также требует упрощения процессов финансового менеджмента.

Для решения поставленной задачи разработано Android приложение для управления финансами. В данной работе будут рассмотрены детали разработки приложения на C# с использованием фреймворка Xamarin и приведены результаты разработки данного приложения.

Разработка клиентской части android приложения

Xamarin предоставляет доступ к API Android и обеспечивает кроссплатформенную функциональность, а также возможность повторного использования кода при разработке для других платформ [1, 2]. Мобильные приложения традиционно используют многоуровневую архитектуру с разделением уровней доступа к данным [3], бизнес-логики и отображения пользовательского интерфейса. Шаблон MVVM (Model-View-ViewModel) описывает взаимосвязь между представлением, моделью представления и моделью данных.

Данная архитектура является классической для Xamarin.Forms[4]. Каждый из этих слоев выполняет специфические функции:

- уровень представления (View) – управляет поведением и внешним видом компонентов пользовательского интерфейса;
- уровень модели представления (ViewModel) – реализует логику взаимодействия с пользователем и бизнес-сценарии, относящиеся к бизнес-логике;
- уровень модели (Model) – отвечает за логику получения, хранения и преобразования данных, что соответствует уровню доступа к данным;
- уровень платформы – включает функциональность, не связанную с пользовательским интерфейсом.

Многоуровневая архитектура позволяет создавать модульные, расширяемые и тестируемые приложения. Код, независимый от платформы, может быть повторно использован в других проектах, что значительно повышает эффективность разработки.

Регистрация и авторизация пользователя приложения

Для регистрации новых пользователей в системе необходимо пользоваться библиотеками, такими как кроссплатформенная библиотека Xamarin.Auth [5], которая помогает аутентифицировать пользователей через протокол OAuth (OAuth1 и OAuth2)

К тому же для аутентификации через сторонние сервисы, например Google необходимо использовать Google Cloud API & Services. Приложение использует классы OAuth2Authenticator и OAuth2Request, предоставляемые Xamarin.Auth, для реализации потока аутентификации на стороне клиента. После инициализации объекта Authenticator с необходимыми параметрами (client_id, redirect_url и т.д.), пользователю отображается браузер со страницей входа в аккаунт Google. После успешной аутентификации приложение получает маркер доступа для дальнейшего взаимодействия с API Google.

Аналогичным образом реализуется интеграция с ВКонтакте, но с использованием соответствующего API ВКонтакте для мобильных приложений. Приложение должно быть зарегистрировано в Standalone приложениях ВКонтакте, получить client_id и client_secret, которые используются при инициализации объекта Authenticator.

Курс валют

Для обновления и поддержания актуальности информации о курсе валют, приложению необходимо актуализировать данные из проверенных источников, таких как сайт Центрального банка России и финансовые биржи.

Получить информацию с данных сайтов можно при помощи парсинга данных. Это достигается с помощью использования библиотеки HtmlAgilityPack [6]. Эта библиотека является популярным инструментом для парсинга HTML-документов в среде .NET, позволяя разработчикам извлекать данные из DOM и выбирать необходимые элементы с помощью

XPath и других методов. Процесс включает загрузку HTML-контента страницы, анализ его структуры и извлечение нужных данных.

Кроме парсинга, некоторые ресурсы, такие как CoinLore, предоставляют API для доступа к актуальной информации о курсах валют. Использование API более надежный и эффективный способ получения данных, так как это позволяет избежать потенциальных проблем, связанных с изменениями в структуре HTML-страниц, которые могут повлиять на парсинг.

Разработка серверной части android приложения

Реализации серверной части приложения возможна на языке программирования PHP. PHP выполняет ключевую роль в обработке пользовательских запросов для работы с MySQL и выполнения операций с данными.

Для управления базой данных приложения подойдет MySQL.

Веб-интерфейс phpMyAdmin значительно упрощает администрирование баз данных, предоставляя пользователю удобные инструменты для выполнения следующих задач:

- создание, изменение и удаление таблиц;
- выполнение SQL-запросов;
- управление правами доступа пользователей.

Apache функционирует как веб-сервер, обрабатывающий входящие HTTP-запросы от клиентов и перенаправляющий их на обработку PHP. Он обеспечивает надежность работы приложения, эффективно управляет нагрузкой и поддерживает безопасность веб-сервера.

Процесс взаимодействия между PHP и MySQL включает несколько ключевых этапов.

Установление соединения: PHP-скрипт устанавливает соединение с базой данных MySQL[7], используя соответствующие функции и параметры. Это позволяет выполнять запросы к базе данных, включая операции SELECT, INSERT, UPDATE и DELETE для манипуляции данными.

Обработка запросов: Скрипт PHP обрабатывает запросы, поступающие от Android-приложения. На основе данных и параметров запроса формируется ответ в формате JSON или XML.

Возврат данных: Ответ отправляется обратно в Android-приложение, где он обрабатывается для отображения информации пользователю.

Обновление интерфейса: Полученные данные могут быть использованы для обновления пользовательского интерфейса и выполнения других операций, необходимых для обеспечения функциональности приложения, таких как поиск данных и взаимодействие с базой данных.

Для обеспечения устойчивого функционирования базы данных и удобного взаимодействия с PHP необходимо обеспечить наличие сервера, на котором будут размещены все

данные приложения. Это позволит обеспечить надежность, безопасность и высокую производительность веб-приложения, что является критически важным для удовлетворения потребностей пользователей.

Безопасность данных

Для клиент-серверного Android-приложения критически важно функционировать через защищенный протокол – https. Htpps, что расшифровывается как HyperText Transfer Protocol Secure, является улучшенной версией протокола http. Основная цель этого протокола - обеспечить безопасную передачу данных между сервером и клиентом в интернете.

Использование протокола https имеет существенные преимущества, включая шифрование передаваемых данных. Это гарантирует защиту информации от несанкционированного доступа, а также от изменения или подмены данных. Это важная мера предосторожности, которая помогает защитить приложение от возможных атак. Несмотря на все сложности, это значительно повышает безопасность передачи данных, что является приоритетом в любом веб-приложении.

Для эффективного использования протокола https, важно иметь собственное доменное имя, поскольку сертификаты ssl/tls выдаются только для определенных доменных имен. Сертификат ssl/tls содержит информацию о доменном имени и сервере, которому предназначены запросы к этому доменному имени. В данном случае, для регистрации доменного имени был выбран сервис reg.ru, благодаря его удобному интерфейсу, доступным ценам и предоставлению бесплатного ssl сертификата на домен в течение года. Важно обратить внимание на настройки сервера nginx. Все запросы, отправленные по протоколу http, автоматически перенаправляются на https. Это гарантирует, что вся коммуникация между клиентом и сервером будет происходить в зашифрованном виде. В настройках сервера nginx также подключен приобретенный ssl сертификат. Все запросы с https перенаправляются на сервер, где устанавливаются необходимые http-заголовки. Это обеспечивает бесперебойное функционирование приложения и его компонентов.

Кроме того, в системе используется процесс хеширования для обеспечения защиты критически важных данных пользователей. Это достигается путем использования алгоритма SHA-256. При регистрации нового пользователя система сохраняет хеш от его пароля. Если пользователь уже зарегистрирован в системе, то при его попытке входа в систему, введенный пароль хешируется и сравнивается с сохраненным на сервере хешем. Если полученный хеш совпадает с хранящимся на сервере, система распознает пароль пользователя как корректно введенный, обеспечивая безопасный доступ к его аккаунту. Эта мера безопасности обеспечивает дополнительный уровень защиты данных пользователя и исключает возможность несанкционированного доступа к аккаунту.

Таким образом, применение протокола https, использование доменных имен и сертификатов ssl/tls, настройки сервера nginx и процесс хеширования для защиты данных пользователя, являются ключевыми элементами безопасности при разработке и функционировании Android-приложения. Это включает в себя все аспекты, от обеспечения безопасного и надежного соединения до защиты критически важных пользовательских данных от несанкционированного доступа.

Публикация приложения в RuStore

RuStore – официальный российский магазин приложений от VK и ведущих IT-компаний, запущенный в 2022 г. Магазин предлагает широкий спектр приложений, включая те, которые больше не доступны в Google Play.

Прежде чем публиковать приложение в RuStore, приложение должно соответствовать требованиям, включая минимальную версию Android 4.4 KitKat, соответствие правилам RuStore и российскому законодательству.

Помимо этого, для публикации приложения необходимо создать аккаунт разработчика в RuStore, собрать APK-файл и подготовить описание, скриншоты и иконку приложения.

После загрузки приложение в RuStore и происходит процесс проверки, который включает в себя заполнение формы загрузки, отправку APK-файла и ожидание результатов проверки, которое обычно занимает до 7 рабочих дней.

Выводы

Разработка Android-приложения для управления личными финансами представляет собой важный шаг в упрощении финансового менеджмента в условиях стремительного развития мобильных технологий и цифровых валют.

Клиентская часть приложения, построенная на многоуровневой архитектуре, обеспечивает модульность и возможность повторного использования кода, что значительно повышает эффективность разработки. Интеграция с внешними сервисами, такими как Google и ВКонтакте, через протокол OAuth, а также использование API для получения актуальной информации о курсах валют, демонстрирует гибкость и функциональность приложения.

Серверная часть, реализованная на PHP с использованием MySQL, обеспечивает надежное управление данными, а также безопасное взаимодействие через протокол HTTPS. Применение современных методов безопасности, таких как хеширование паролей и использование SSL-сертификатов, гарантирует защиту пользовательских данных.

Публикация приложения в RuStore открывает новые возможности для его распространения, учитывая требования российского законодательства и специфику местного рынка.

В целом, разработанное приложение не только отвечает современным требованиям пользователей, но и способствует более эффективному управлению личными финансами в условиях цифровизации.

Библиографический список

1. Топ-10 языков программирования для разработки приложений под iOS и Android [Электронный ресурс]. URL: <https://forum-media.kz/programming-languages/top-10-yazykov-programmirovaniya-dlya/> (дата обращения: 22.05.2024).
2. Какой язык программирования выбрать для разработки мобильных приложений [Электронный ресурс]. URL: <https://it-vacancies.ru/blog/kakoi-iazyk-programmirovaniia-vybrat-dlia-razrabotki-mobilnyx-prilozenii/> (дата обращения: 22.05.2024).
3. Guide to app architecture [Электронный ресурс]. URL: <https://developer.android.com/topic/architecture> (дата обращения: 20.05.2024).
4. Xamarin.Forms Documentation [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/previous-versions/xamarin/xamarin-forms/> (дата обращения: 22.05.2024).
5. Xamarin Documentation [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/previous-versions/xamarin/get-started/what-is-xamarin> (дата обращения: 22.05.2024).
6. Html Agility Pack (HAP) [Электронный ресурс]. URL: <https://html-agility-pack.net/> (дата обращения: 20.05.2024).
7. MySQL [Электронный ресурс]. URL: <https://www.mysql.com/> (дата обращения: 22.05.2024).

References

1. Top-10 yazykov programmirovaniya dlya razrabotki prilozhenij pod iOS i Android [Electronic resource]. URL: <https://forum-media.kz/programming-languages/top-10-yazykov-programmirovaniya-dlya/> (date of access: 05/22/2024). (In Russ).
2. Kakoj yazyk programmirovaniya vybrat' dlya razrabotki mobil'nyh prilozhenij [Electronic resource]. URL: <https://it-vacancies.ru/blog/kakoi-iazyk-programmirovaniia-vybrat-dlia-razrabotki-mobilnyx-prilozenii/> (date of access: 05/22/2024). (In Russ.)
3. Guide to app architecture [Electronic resource]. URL: <https://developer.android.com/topic/architecture> (date of access: 05/20/2024).
4. Xamarin.Forms Documentation [Electronic resource]. URL: <https://learn.microsoft.com/ru-ru/previous-versions/xamarin/xamarin-forms/> (date of access: 05/22/2024).
5. Xamarin Documentation [Electronic resource]. URL: <https://learn.microsoft.com/ru-ru/previous-versions/xamarin/get-started/what-is-xamarin> (date of access: 05/22/2024).

6. Html Agility Pack (HAP) [Electronic resource]. URL: <https://html-agility-pack.net/> (date of access: 05/20/2024).
7. MySQL [Electronic resource]. URL: <https://www.mysql.com/> (date of access: 05/22/2024).

DESIGNING, DEVELOPING AND DOCUMENTING AN ANDROID APPLICATION FOR MONITORING PERSONAL FINANCES AND CURRENCY EXCHANGE RATES

Kostitsyn V.V. Knutova N.S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, kosticyn.vladislav@gmail.com

With the development of technology, personal finance management is becoming easier thanks to mobile applications. They help to track income and expenses, and plan a budget for financial stability. In an environment of volatile exchange rates, applications for tracking and analyzing exchange rates are becoming more and more in demand. They provide information about current rates, allow you to set alerts about changes, which helps you make financial decisions. The article discusses the development and data security of a mobile application, as well as the process of publishing in RuStore. At the same time, the client and server side of the application will be considered.

Keywords: development, android application, application development.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ФУДШЕРИНГА

Кузнецов Александр Максимович, Гасумова Светлана Евгеньевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, kuznetcov.alma@yandex.ru

В данной работе рассматривается проектирование информационной системы для фудшеринга. Показана актуальность проблемы бедности, голода и пищевых отходов в мире и в России. Описан фудшеринг и его история, начиная с первых банков еды в 1960-х годах и до современных движений. Проведены сравнение и анализ существующих российских платформ для фудшеринга, таких как Karmitt, ЕдаДома и DoggyBag, в виде таблицы. На основе анализа определены задачи, которые должна решать проектируемая система. Выбраны инструменты для моделирования поведения системы, создания прототипов пользовательского интерфейса и средства для её дальнейшей разработки. Работа содержит диаграмму прецедентов, отражающую взаимодействие акторов с системой, а также прототипы интерфейса.

Ключевые слова: фудшеринг, информационная система.

В современном мире проблемы голода и бедности проявляются в различной степени во всех странах. В России, хотя она и входит в группу стран с устойчивым развитием и довольно высоким уровнем индекса человеческого развития [1], тем не менее, значительная часть населения имеет доходы ниже прожиточного минимума. Так, доля населения, которая относится к этой группе, в первом квартале 2023 г. по данным Федеральной службы государственной статистики составила 13,5% от всех жителей страны, а именно 19,6 млн чел. [1]. Если же говорить про весь мир, то по данным ООН на 2022 г. более 1,2 млрд чел. в мире живут в условиях крайней нищеты [2].

Что же касается голода и пищевых отходов, то, по словам директора московского отделения Продовольственной и сельскохозяйственной организации ООН О.Ю. Кобякова, в России в 2023 г. голодали 400 тысяч чел., а исходя из результатов исследований ООН на 2022 г., около 2,3 млрд чел. испытывали умеренную или острую нехватку продовольствия [3, 4]. Также согласно докладу ООН в рамках программы по окружающей среде в 2019 г. был образован 931 млн тон пищевых отходов, а по данным Всемирной продовольственной программы ежегодно выбрасывается треть всех продуктов питания [5].

Таким образом, как в России, так и в мире присутствуют проблемы бедности, голода и в то же время пищевых отходов. Одним из решений данных проблем является фудшеринг.

Под фудшерингом понимается социально-экологическая технология перераспределения продуктов питания, задача которой – реализовать пригодную для употребления еду до истечения срока годности вместо того, чтобы её утилизировать [7].

Моментом появления данной технологии можно считать возникновение первых банков еды в США в 60-х гг. прошлого века. Само же понятие «фудшеринг» появилось в 2012 г. в Германии, в Берлине как общественное экологическое движение. Дословно термин переводится как «делиться едой» [8].

Сегодня в России для этого занятия используются социальные сети, где создаются сообщества, в которых люди обмениваются едой, и специализированные платформы, обзор которых приведён в статье. Однако функционал таких сообществ ограничен, а существующие платформы имеют некоторые недостатки. Поэтому для привлечения большего числа людей в реализацию этой концепции необходимо создать информационную систему (ИС) для фудшеринга.

Цель данной статьи – проектирование информационной системы для фудшеринга.

Объектом исследования является автоматизация деятельности фудшеринга в России, а предметом исследования выступает информационная система для решения задач фудшеринга.

Для достижения обозначенной цели необходимо выполнить следующие задачи: провести обзор литературы по теме исследования и сравнительный анализ существующих информационных систем для фудшеринга, построить логическую структуру и функциональную модель информационной системы, разработать прототип пользовательского интерфейса информационной системы.

В соответствии с задачами был проведен сравнительный анализ, организованный в виде таблицы (см. табл.), существующих российских систем для фудшеринга по нескольким критериям:

- 1) прозрачность объявления, позволяющая пользователю знать, что он получит;
- 2) работоспособность системы;
- 3) понятность интерфейса;
- 4) наличие проверки объявления перед публикацией;
- 5) стороны, которые взаимодействуют друг с другом при фудшеринге;
- 6) количество объявлений на территории Пермского края.

Первые три критерия были оценены по 5-бальной системе, где 1 – это низшая оценка, а 5 – высшая. Для них в таблице представлены результаты авторской оценки. В результате анализа были выделены как преимущества, так и недостатки рассмотренных платформ [9–11].

Исходя из результатов сравнительного анализа, были составлены задачи, которые должна решать проектируемая система:

- 1) обеспечить регистрацию в ИС как для обычных пользователей, так и для продуктовых заведений;
- 2) предоставить удобный функционал для создания, просмотра и удаления объявлений;

3) дать возможность обмениваться продуктами питания между участниками сообщества фудшеринга и продуктовыми заведениями, где могут возникать пищевые отходы.

Таблица

Сравнительный анализ российских информационных систем для фудшеринга

Критерии оценки	Karmitt	ЕдаДома	DoggyBag
Прозрачность объявлений	5	5	2
Работоспособность системы	5	3	5
Понятность интерфейса	5	5	5
Проверка объявления перед публикацией	Отсутствует	Присутствует	Отсутствует
Взаимодействующие стороны	Люди	Люди	Продуктовые заведения и люди
Количество объявлений на территории Пермского края, шт.	0	3	0

При создании этой системы можно будет заключить договоры о сотрудничестве с органами труда и социального развития Пермского края, социальными службами и общественными организациями, которые согласятся распространить социальную рекламу данной информационной системы, чтобы проинформировать о ней население региона (в первую очередь – малоимущих граждан).

Для моделирования процессов информационной системы были выбраны язык UML и графический редактор diagrams.net [6]. Результатом моделирования стала диаграмма прецедентов (см. Рис. 1). На ней можно увидеть всех акторов и их взаимодействие с системой. Акторами являются пользователь, продуктовая организация и администратор – лицо, контролирующее работу системы.

В качестве инструмента для создания прототипов был выбран Figma [7], так как он имеет бесплатный план, благодаря которому можно было создать необходимые прототипы, несмотря на малозначимые ограничения в отличие от остальных инструментов.

Сначала был сделан прототип главной страницы системы. На ней будут отображаться доступные объявления, кнопки навигации, которые позволят перейти к соответствующему функционалу (см.

Рис. 2).

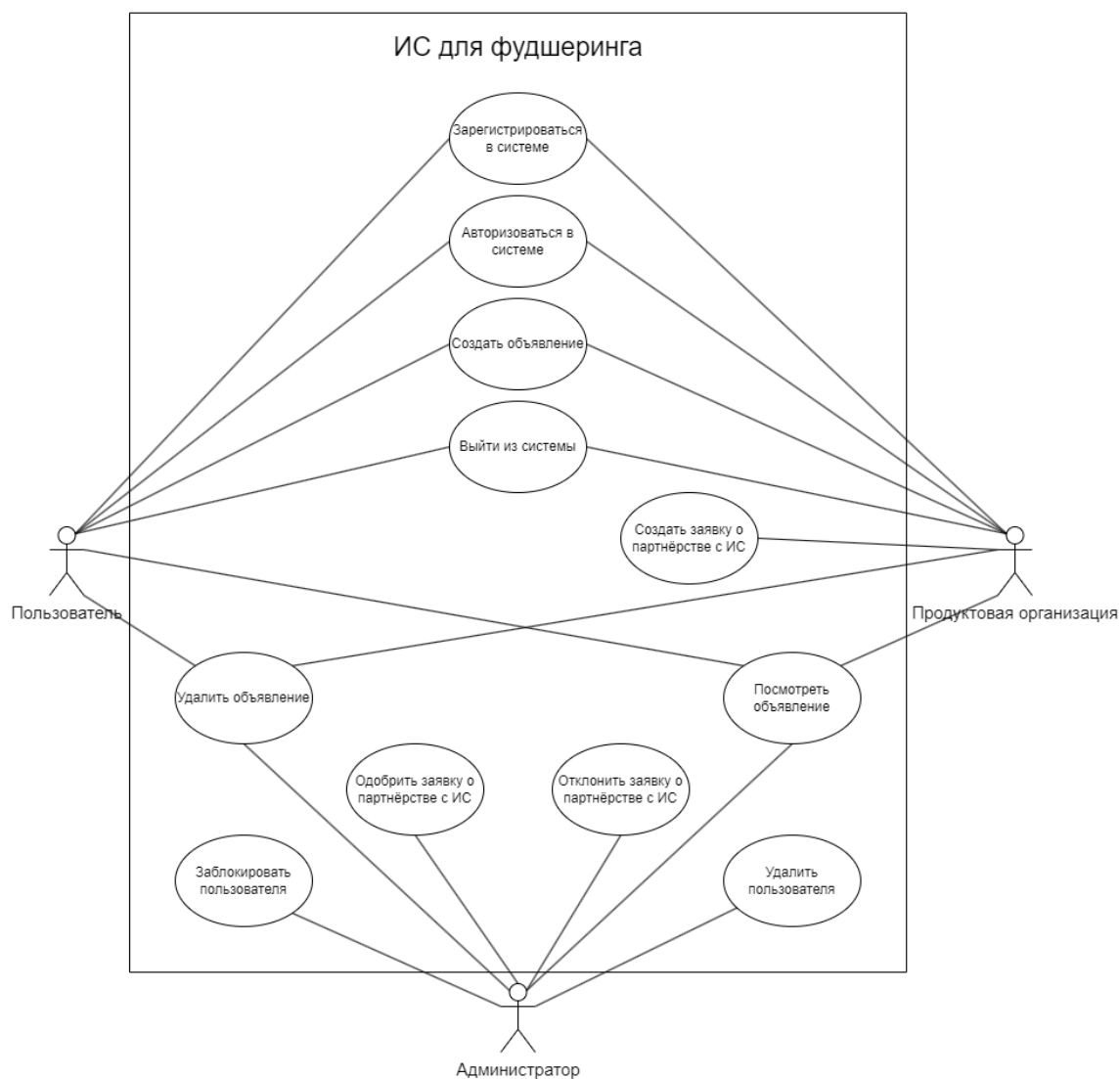


Рис. 4. Диаграмма прецедентов

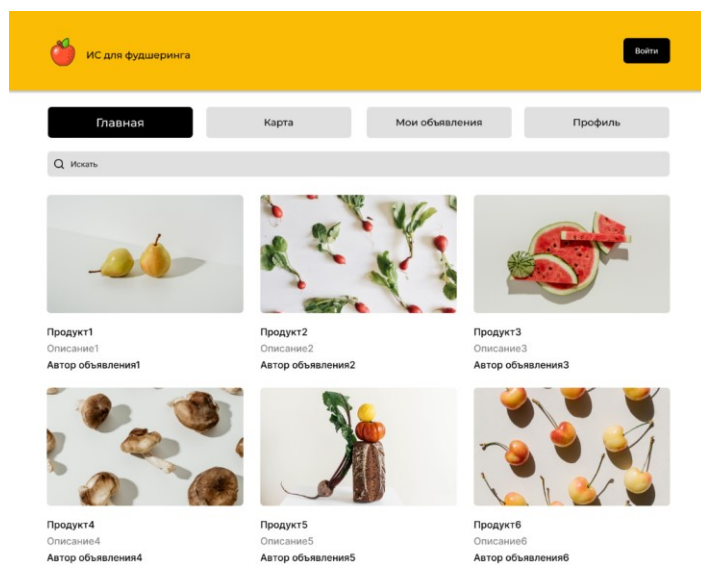


Рис. 5. Главная страница

Далее было спроектировано окно для регистрации пользователя в системе (см. Рис. 3).

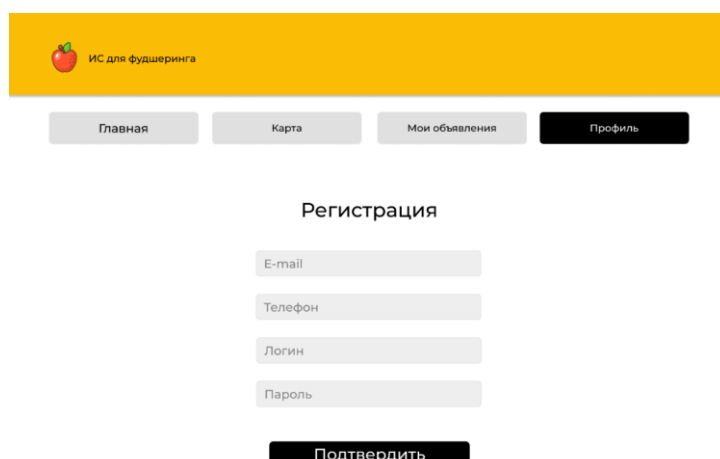


Рис. 6. Страница регистрации

Таким образом, исходя из существующих проблем бедности, голода и в то же время пищевых отходов, фудшеринг в России и в мире является актуальной потребностью для общества. Данную практику спасения пригодной для употребления еды целесообразнее всего реализовать с помощью цифровых технологий. Представленная спроектированная информационная система может быть оптимальным вариантом для решения выделенных в результате анализа задач. Она даст участникам фудшеринга возможность создавать объявления с едой, которой они хотят поделиться, просматривать объявления других пользователей и таким образом получать бесплатные продукты питания. Всё это будет способствовать сохранению пригодных для употребления продуктов питания и повышению качества жизни малоимущих слоёв населения.

Библиографический список

1. Human development insights / Human development reports [Электронный ресурс]. URL: <https://hdr.undp.org/data-center/country-insights#/ranks> (дата обращения: 23.02.2024).
2. Официальный сайт Федеральной службы государственной статистики // Росстат представил данные о численности населения с денежными доходами ниже границы бедности в I квартале 2023 года [Электронный ресурс]. URL: <https://rosstat.gov.ru/folder/313/document/208845> (дата обращения: 15.02.2024).
3. Более 1,2 млрд человек в мире живут в условиях крайней нищеты // Новости ООН [Электронный ресурс]. URL: <https://news.un.org/ru/story/2022/10/1433527> (дата обращения: 06.03.2024).

4. В продовольственной структуре ООН заявили о победе России над голодом // РБК [Электронный ресурс]. URL: <https://www.rbc.ru/business/16/06/2023/648c37189a79474633f9eaeef> (дата обращения: 09.03.2024).
5. ФАО, МФСР, ЮНИСЕФ, ВПП и ВОЗ. 2022 год. Положение дел в области продовольственной безопасности и питания в мире – 2022. Переориентация политики в области продовольствия и сельского хозяйства в интересах повышения экономической доступности здорового питания. Рим, ФАО. С. 260. URL: <https://doi.org/10.4060/cc0639ru> (дата обращения: 06.03.2024).
6. UNEP. 2021. Food Waste Index Report 2021. Nairobi, UNEP. С. 100. URL: <https://wedocs.unep.org/bitstream/handle/20.500.11822/35280/FoodWaste.pdf> (дата обращения: 10.03.2024).
7. Фудшеринг и бизнес: отдавать, продавать или выбрасывать продукты с истекающим сроком годности? // Retail.ru [Электронный ресурс]. URL: <https://www.retail.ru/articles/fudshering-i-biznes-otdavati-prodavati-ili-vybrasyvat-produkty-s-istekayushchim-srokom-godnosti/> (дата обращения: 17.03.2024).
8. Фонд «Банк еды «Русь» [Электронный ресурс]. URL: <https://foodbankrus.ru/> (дата обращения: 17.03.2024).
9. Karmitt [Электронный ресурс]. URL: <https://karmitt.com/> (дата обращения: 03.04.2024).
10. ЕдаДома [Электронный ресурс]. URL: <https://edadoma.ru/> (дата обращения: 03.04.2024).
11. DoggyBag [Электронный ресурс]. URL: <https://doggybag.club/> (дата обращения: 03.04.2024).
12. diagrams.net [Электронный ресурс]. URL: <https://www.diagrams.net/> (дата обращения: 16.04.2024).
13. Figma [Электронный ресурс]. URL: <https://www.figma.com/> (дата обращения: 17.04.2024).

References

1. Human development insights / Human development reports [Electronic resource]. URL: <https://hdr.undp.org/data-center/country-insights#/ranks> (date of access: 02/23/2024).
2. Official website of the Federal State Statistics Service // Rosstat predstavil dannye o chislennosti naseleniya s denezhnymi dohodami nizhe granicy bednosti v I kvartale 2023 goda [Electronic resource]. URL: <https://rosstat.gov.ru/folder/313/document/208845> (date of access: 02/15/2024). (In Russ.)

3. Bolee 1,2 mlrd chelovek v mire zhivut v usloviyah krajnej nishchety // UN News [Electronic resource]. URL: <https://news.un.org/ru/story/2022/10/1433527> (date of access: 03/06/2024). (In Russ.)
4. V prodovol'stvennoj strukture OON zayavili o pobede Rossii nad golodom // RBC [Electronic resource]. URL: <https://www.rbc.ru/business/16/06/2023/648c37189a79474633f9eaeef> (date of access: 03/09/2024). (In Russ.)
5. FAO, IFAD, UNICEF, WFP and WHO. 2022. Polozhenie del v oblasti prodovol'stvennoj bezopasnosti i pitaniya v mire – 2022. Pereorientaciya politiki v oblasti prodovol'stviya i sel'skogo hozyajstva v interesah povysheniya ekonomicheskoy dostupnosti zdorovogo pitaniya. Rome, FAO. P. 260. URL: <https://doi.org/10.4060/cc0639ru> (date of access: 03/06/2024). (In Russ.)
6. UNEP. 2021. Food Waste Index Report 2021. Nairobi, UNEP. P. 100. URL: <https://wedocs.unep.org/bitstream/handle/20.500.11822/35280/FoodWaste.pdf> (date of access: 03/10/2024).
7. Fudshering i biznes: otdavat', prodavat' ili vybrasyvat' produkty s istekayushchim srokom godnosti? // Retail.ru [Electronic resource]. URL: <https://www.retail.ru/articles/fudshering-i-biznes-otdavati-prodavati-ili-vybrasyvat-produkty-s-istekayushchim-srokom-godnosti/> (date of access: 03/17/2024). (In Russ.)
8. Fond «Bank Edy «Rus'»» [Electronic resource]. URL: <https://foodbankrus.ru/> (date of access: 03/17/2024). (In Russ.)
9. Karmitt [Electronic resource]. URL: <https://karmitt.com/> (date of access: 04/03/2024). (In Russ.)
10. EdaDoma [Electronic resource]. URL: <https://edadoma.ru/> (date of access: 04/03/2024). (In Russ.)
11. DoggyBag [Electronic resource]. URL: <https://doggybag.club/> (date of access: 04/03/2024). (In Russ.)
12. diagrams.net [Electronic resource]. URL: <https://www.diagrams.net/> (date of access: 04/16/2024).
13. Figma [Electronic resource]. URL: <https://www.figma.com/> (date of access: 04/17/2024).

DESIGNING AN INFORMATION SYSTEM FOR FOODSHARING

Kuznetsov Alexander Maksimovich, Gasumova Svetlana Evgenievna

Perm State National Research University, 15 Bukireva str., Perm, 614068, Russia,
kuznetcov.alma@yandex.ru

In this paper, the design of an information system for foodsharing is considered. The urgency of the problem of poverty, hunger and food waste in the world and in Russia is formulated. Foodsharing and its history are described, starting with the first food banks in the 1960s and up to modern movements. The comparison and analysis of existing Russian foodsharing platforms, such as Karmitt, EdaDoma and DoggyBag, are carried out in the form of a table. Based on the analysis, the tasks that the designed system should solve are determined. The tools for modeling the behavior of the system, creating prototypes of the user interface and tools for its further development are selected. The work contains a precedent diagram reflecting the interaction of actors with the system, as well as interface prototypes.

Keywords: foodsharing, information system.

ПРОЕКТИРОВАНИЕ И ПРОТОТИПИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ПОДДЕРЖКИ ПРОЕКТОВ ДЕТСКОГО ТЕХНОПАРКА «КВАНТОРИУМ ФОТОНИКА»

Лешехва Екатерина Андреевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, leshehvaea@yandex.ru

Кушев Вадим Олегович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, vadimk@psu.ru

Рассматриваются основные теоретические положения для проектирования и прототипирования автоматизированной системы с целью регистрации проектов учащихся детского технопарка «Кванториум-Фотоника». Определены предмет и объект исследования, сформулирована актуальность разработки информационной системы, а также поставлены цель и задачи, которые необходимо выполнить в ходе проектирования. Проведен анализ предметной области, определены понятия и их свойства. Проведена беседа с заказчиком, в ходе которой определены пользователи системы и функционал приложения. Проведен анализ существующих решений. В ходе исследования был определен набор средств проектирования и прототипирования. В работе рассматриваются инструменты UML-проектирования draw.io, CASE-средство DBDesigner для построения ER-диаграммы, и онлайн-редактор Figma в качестве средства прототипирования. В результате проектирования автоматизированной системы было описано взаимодействие пользователей с ней, описана архитектура приложения на программно-аппаратном уровне, была разработана ER-диаграмма на основе предметной области Кванториума, был спроектирован интерактивный интерфейс приложения.

Ключевые слова: детский технопарк «Кванториум Фотоника», Кванториум, регистрация проектов Кванториума, автоматизированная система.

Введение

Множество организаций нуждаются в программном обеспечении, которое дает возможность получать, накапливать, хранить и обрабатывать данные, представляя результат в виде наглядных документов. Одной из таких организаций является детский технопарк «Кванториум Фотоника». В настоящее время регистрация проектов учащихся технопарка осуществляется средствами электронных таблиц, где хранится неструктурированная база данных, содержащая в себе проекты учащихся, их научных руководителей, квантумы (площадки для реализации) и многое другое. Вследствие чего извлечение данных и их представление является труднодоступным. Для устранения трудностей, перечисленных ранее, была поставлена цель – проектирование приложения для регистрации проектов учащихся технопарка с целью получения отчетности в дальнейшем.

Постановка задачи

Прежде чем приступать к проектированию системы необходимо ознакомиться с требованиями заказчика. Было принято решение обратиться к директору технопарка Кренделю Павлу Ильичу (далее – Заказчик). В ходе обсуждения была поставлена задача: создать web-сайт, на котором будут храниться данные о проектах, созданных в пермском Кванториуме. Система должна обеспечивать вход для администратора. Администратор должен обладать функциями удаления, изменения, создания, чтения данных. А также в системе должен быть доступ для обычного пользователя, который может просмотреть этот регистр проектов.

Методы научного исследования

В ходе работы применены методы теоретического исследования: анализ и моделирование.

В результате анализа предметной области детского технопарка определяются основные понятия и их свойства.

Понятие «Участник» – учащийся Кванториума. Важные характеристики «Участника»: фамилия, имя, отчество, дата рождения, почта, школа, фото.

Понятие «Руководитель» – педагог Кванториума или консультант вне технопарка. Важные характеристики «Руководителя»: фамилия, имя, отчество, дата рождения, фото.

Понятие «Квантум» – площадка для реализации проекта. Важные характеристики «Квантума»: название, пиктограмма.

Понятие «Проект» – научная работа учащегося/ихся Кванториума. Важные характеристики «Проекта»: название, введение, проблема, актуальность, цель, задачи, этапы, результат, перспективы, дополнительная информация, документы (фото, презентация).

Понятие «Конкурс» – соревнование, в котором принимают участие проекты. Важные характеристики «Конкурса»: название.

После анализа предметной области осуществляется анализ существующих решений. Пермский технопарк использует электронные таблицы в качестве хранения данных [1]. В ходе анализа были выявлены недостатки применения этого способа представления данных: сложный поиск и сортировка, ошибки в написании дат, множество листов и документов. Также мы рассмотрели решение Томского технопарка, который является web-ресурсом [2]. Недостатки приложения томского технопарка: нет возможности сортировки или поиска данных для обычного пользователя. В ходе анализа было сформировано представление о конечном продукте.

Метод моделирования был использован в процессе проектирования и прототипирования автоматизированной системы.

Проектирование автоматизированной системы

Для проектирования ER-диаграммы было использовано CASE-средство DBDesigner [3], так как этот инструмент бесплатный, дает возможность заполнять в таблице тип данных, метку «нулевое значение», обозначения внешнего и первичного ключа, имеет цветовую палитру, а также этот инструмент представляет web-сайт, а не локальное приложение.

Для создания UML-диаграмм преимущественно использовано CASE-средство diagrams.net [4], так как оно предоставляет удобный интерфейс и бесплатный доступ.

Для проектирования концептуальной инфологической модели базы данных построение проводилось в рамках модели entity relation, так как она наиболее полно отражает сущности и их отношения.

На основе данных форм отчетов о деятельности технопарка была сформирована структура базы данных в нулевой нормальной форме и нормализована до 3 нормальной формы, в результате которой получилось 9 таблиц. Выделены классы сущностей, которые будут храниться в базе данных: зарегистрированный проект, конкурс, место в конкурсе, проект, руководитель, руководитель в квантуме, квантум, участник, школа.

Рассмотрев отношения между сущностями, была построена ER-диаграмма, представленная на рис. 1.

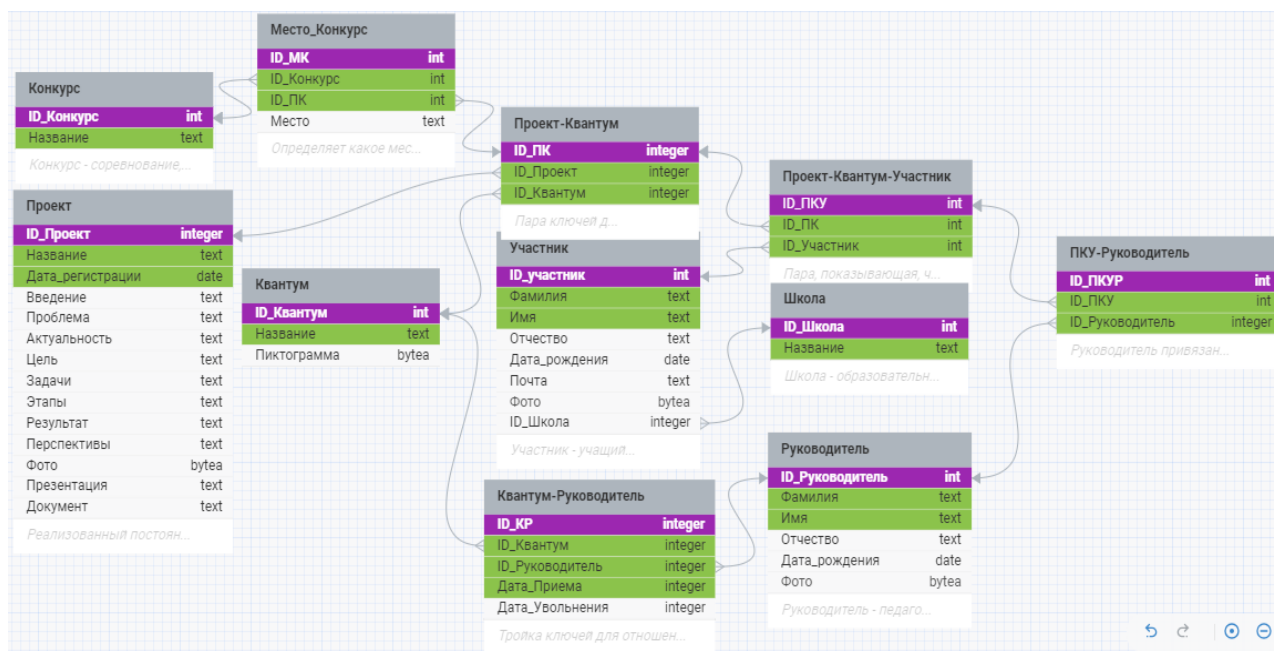


Рис. 1. ER-диаграмма

Взаимодействие пользователей с системой реализовано с помощью диаграммы прецедентов, представленная ниже на рис. 2.



Рис. 2. Диаграмма прецедентов

Администратор при взаимодействии с системой обладает следующими функциями: авторизация, редактирование, удаление, чтение, добавление и поиск данных. Авторизация – предоставление прав доступа к функциям администратора с помощью входа в систему по логину и паролю. Остальные функции предназначены для обновления и поддержки БД. Редактирование – частичное изменение записей в базе данных. Удаление – полное уничтожение данных без возможности восстановления. Чтение – извлечение данных из базы. Добавление – создание новых записей. Поиск – сортировка и фильтрация. Пользователь при взаимодействии с системой может просматривать web-страницу с данными и использовать сортировку.

Была построена диаграмма последовательности авторизации администратора, представленная на рис. 3.

Администратор переходит на форму авторизации, где вводит логин и пароль и нажимает на кнопку «Войти». Далее в системе осуществляется аутентификация. Если данные введены неверно, то выводится сообщение об ошибке, иначе администратор осуществляет вход в систему.

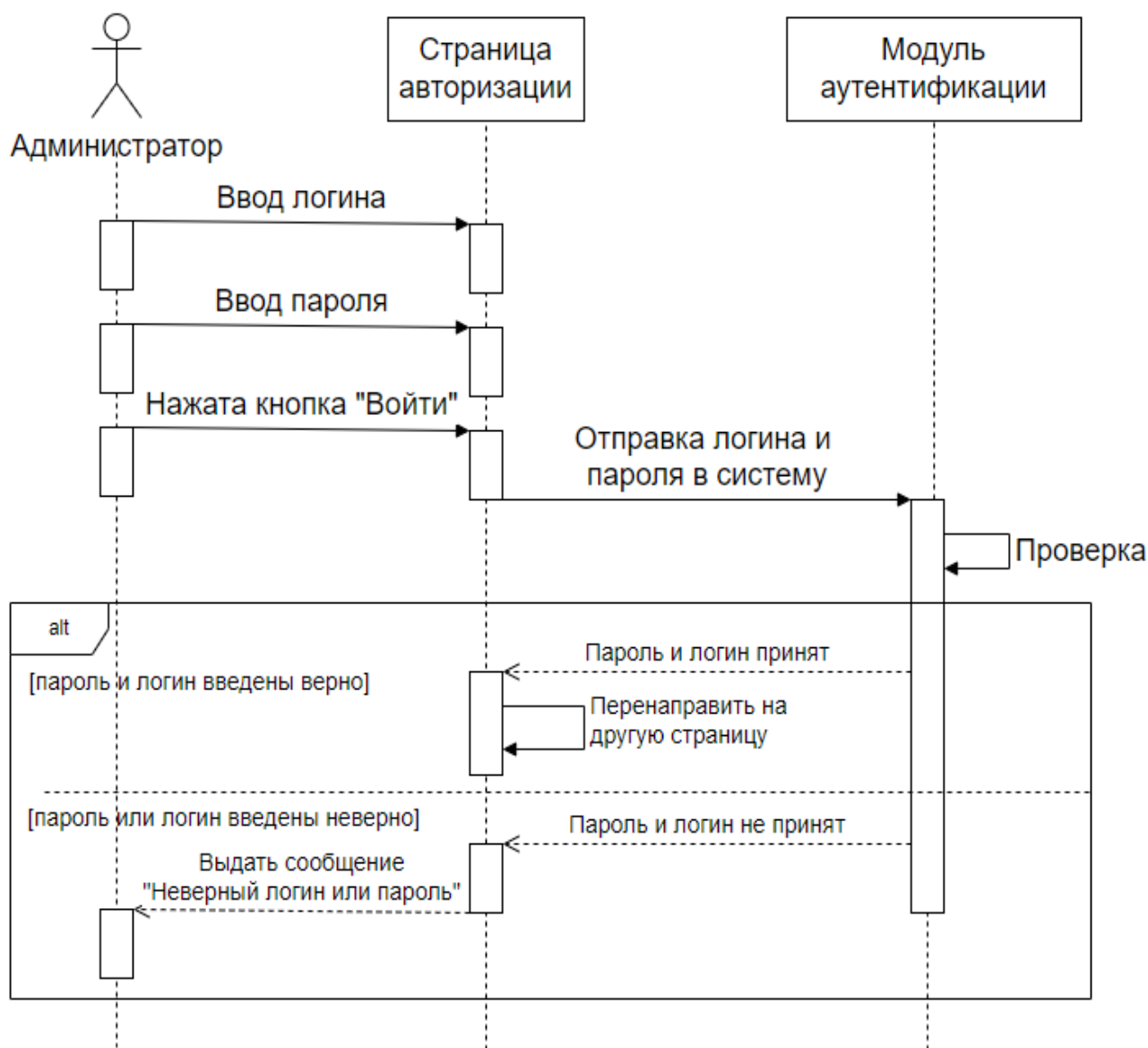


Рис. 3. Диаграмма последовательности авторизации администратора

Была построена диаграмма последовательности функции добавление (см. рис. 4). Администратор переходит на форму добавления данных и выбирает кнопку для добавления экземпляра класса в систему. Далее он переходит на форму заполнения сведений и вводит данные. Если идентичного кортежа в системе не обнаружено, то экземпляр успешно добавляется, иначе выводится сообщение об ошибке.

Аналогичным образом построены диаграммы последовательности редактирования и удаления данных в системе, которые представлены на рис. 5 и 6.

Была построена диаграмма последовательности функции чтения, а именно формирование отчетности (см. рис. 7). Администратор переходит на форму с выбором даты и параметров для фильтрации и при нажатии на кнопку «Запросить отчет» конвентируется документ в PDF или CSV формате, который отправляется на электронную почту.

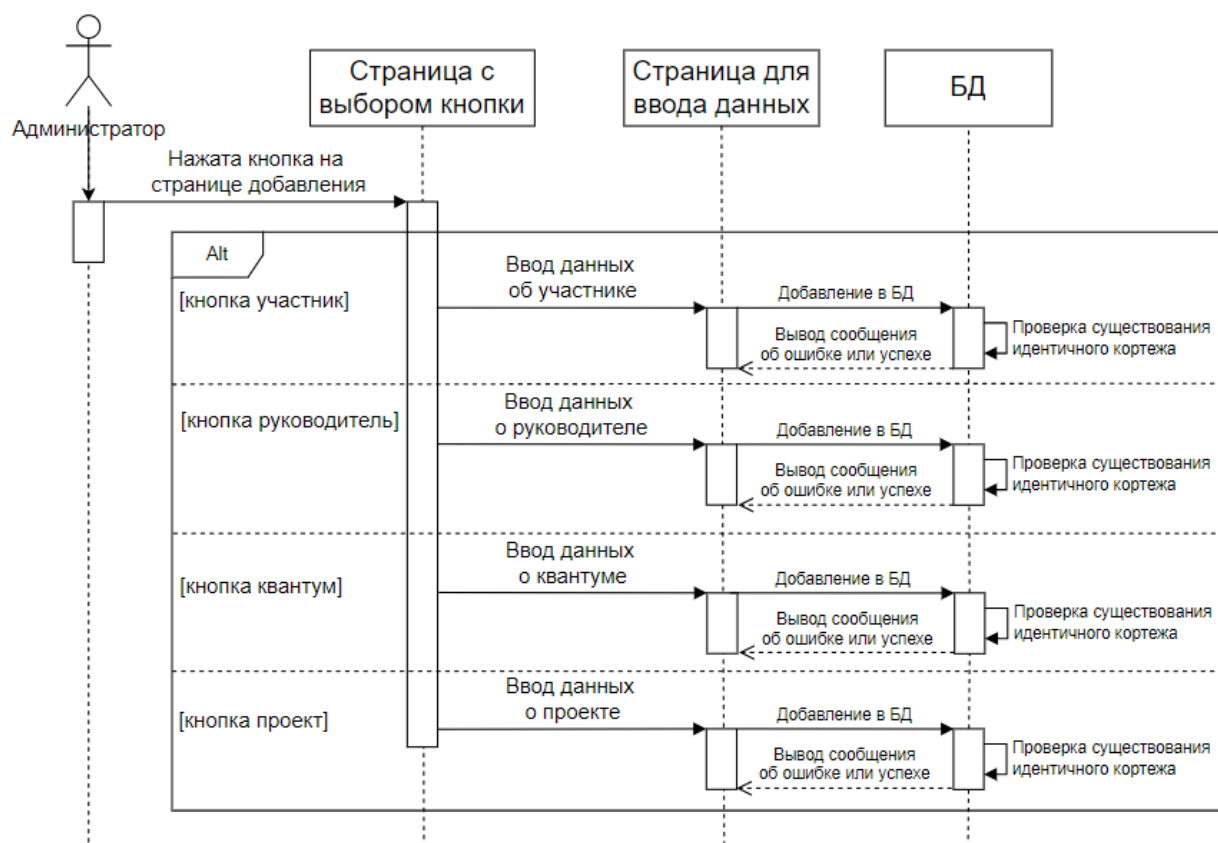


Рис. 4. Диаграмма последовательности функции добавление

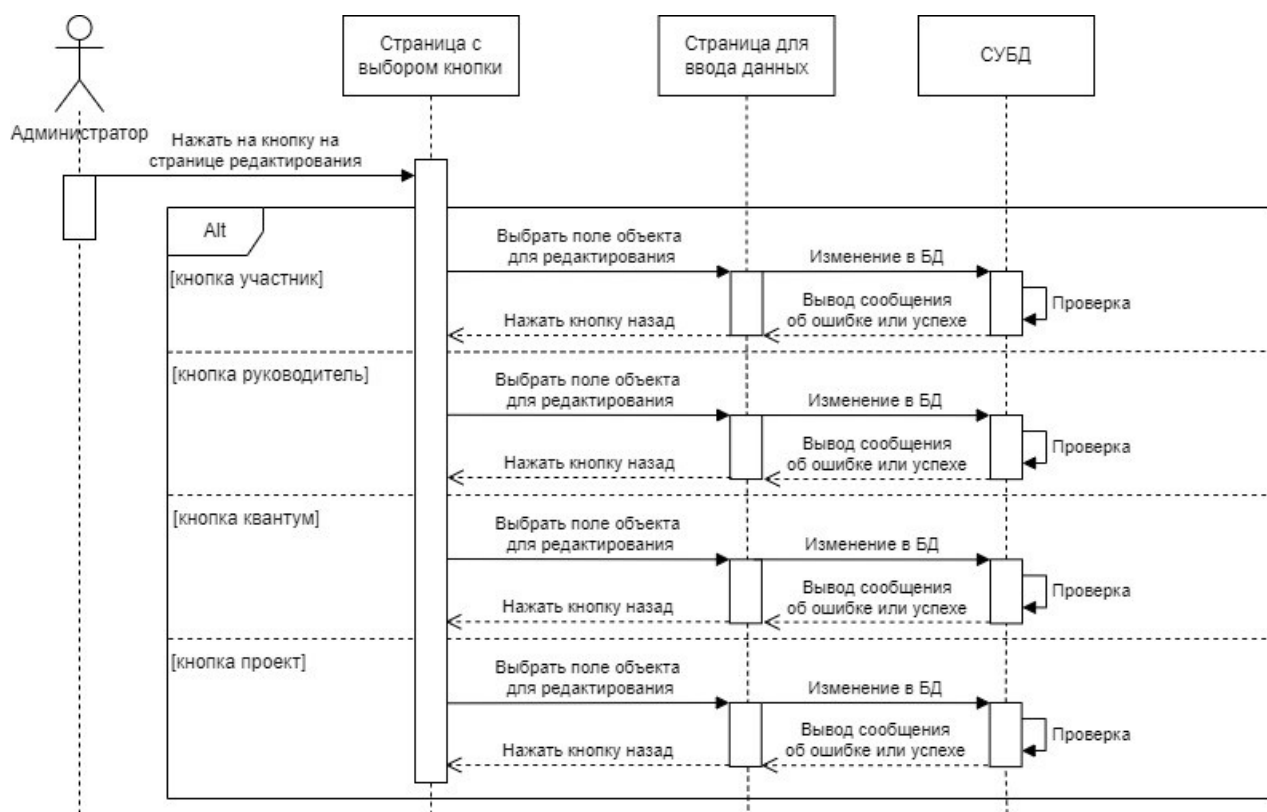


Рис. 5. Диаграмма последовательности функции редактирование

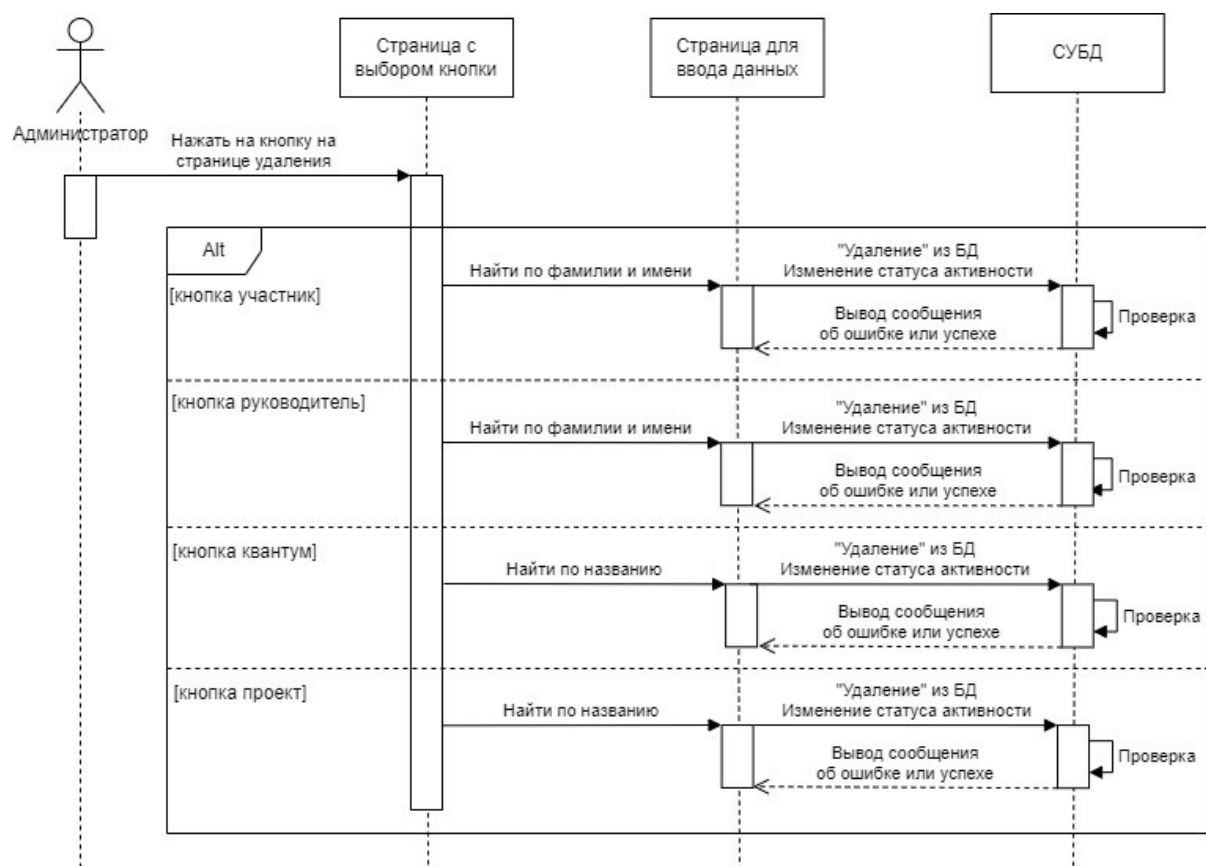


Рис. 6. Диаграмма последовательности функции удаление

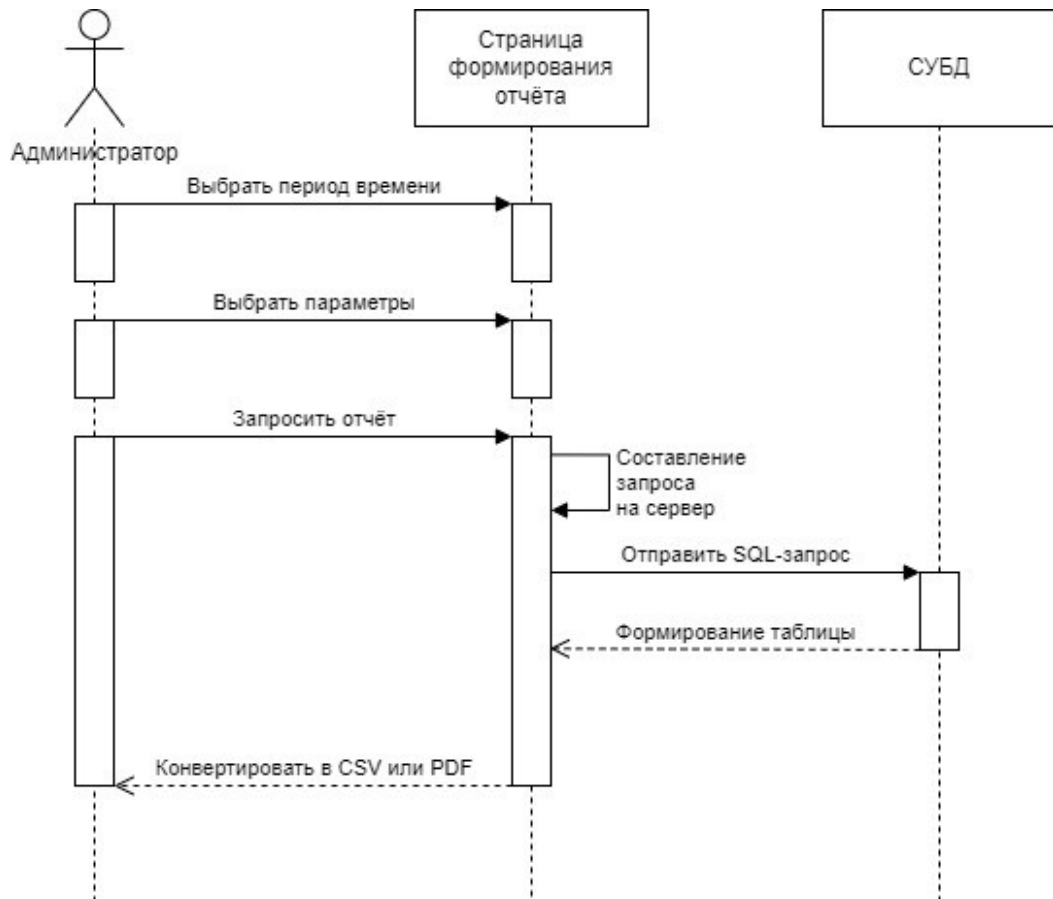


Рис. 7. Диаграмма последовательности функции чтение

Архитектура исполнения процессов на программно-аппаратном уровне представлена на диаграмме развертывания (см. рис. 8). На ней изображены 3 узла: клиент, сервер, СУБД. Клиент – слой представления данных. Сервер – слой управления функциями системы. СУБД – средство управления базой данных. Клиент взаимодействует с сервером с помощью запросов HTTP, а сервер посылает запросы к БД с помощью SQL-запросов и передает ответ клиенту.

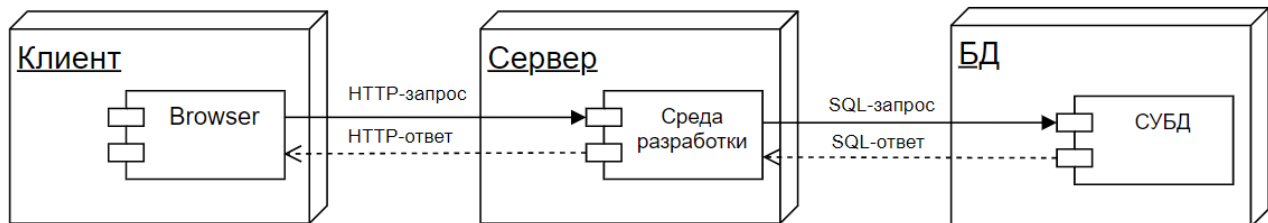


Рис. 8. Диаграмма развертывания

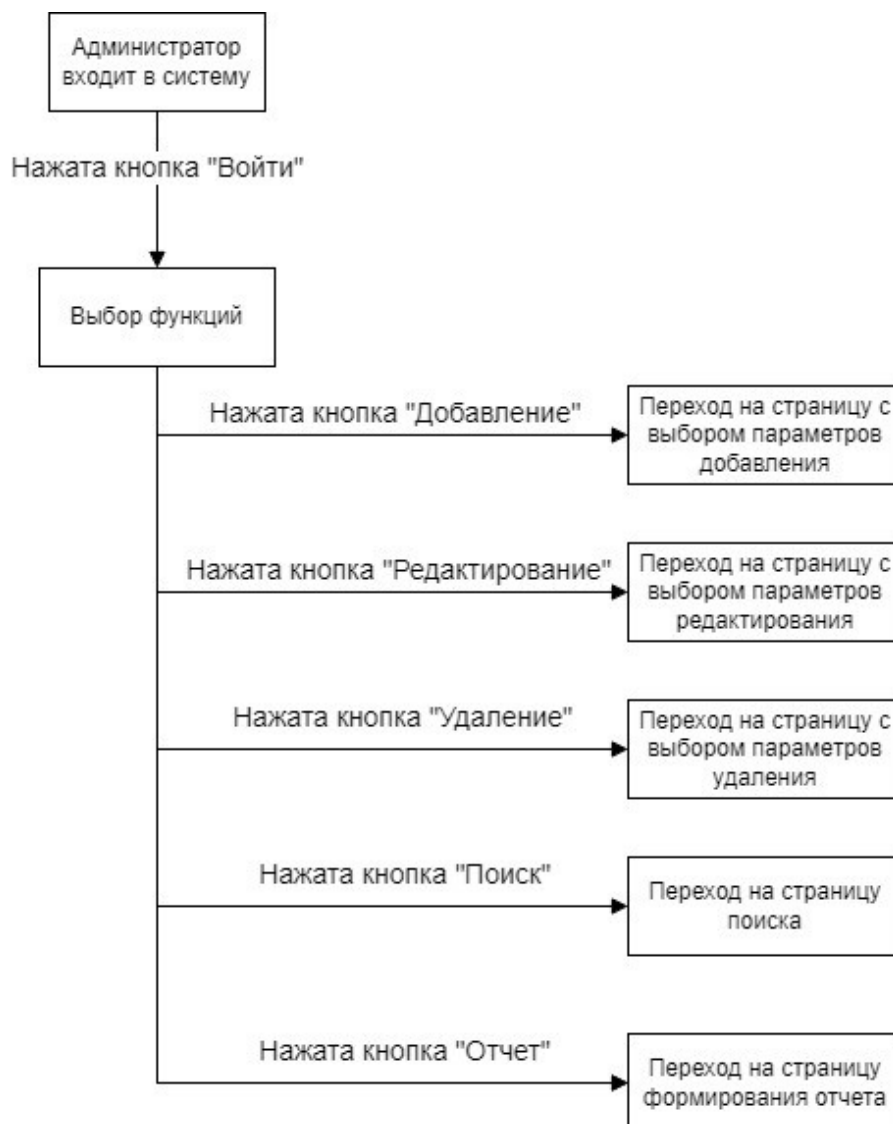


Рис. 9. Диаграмма переходов функций администратора

Были построены модели динамических аспектов, представленные на диаграмме переходов функций администратора и на диаграмме переходов функции добавление (см. рис. 9 и 10). Администратор нажимает на кнопку с выбором операции на странице с выбором функций и переходит на страницу соответствующей функции. Если он выбрал функцию добавление, то он переходит на страницу добавления данных и выбирает параметр, который необходимо добавить. Далее ему необходимо заполнить формы в соответствии с диаграммой переходов функции добавление.

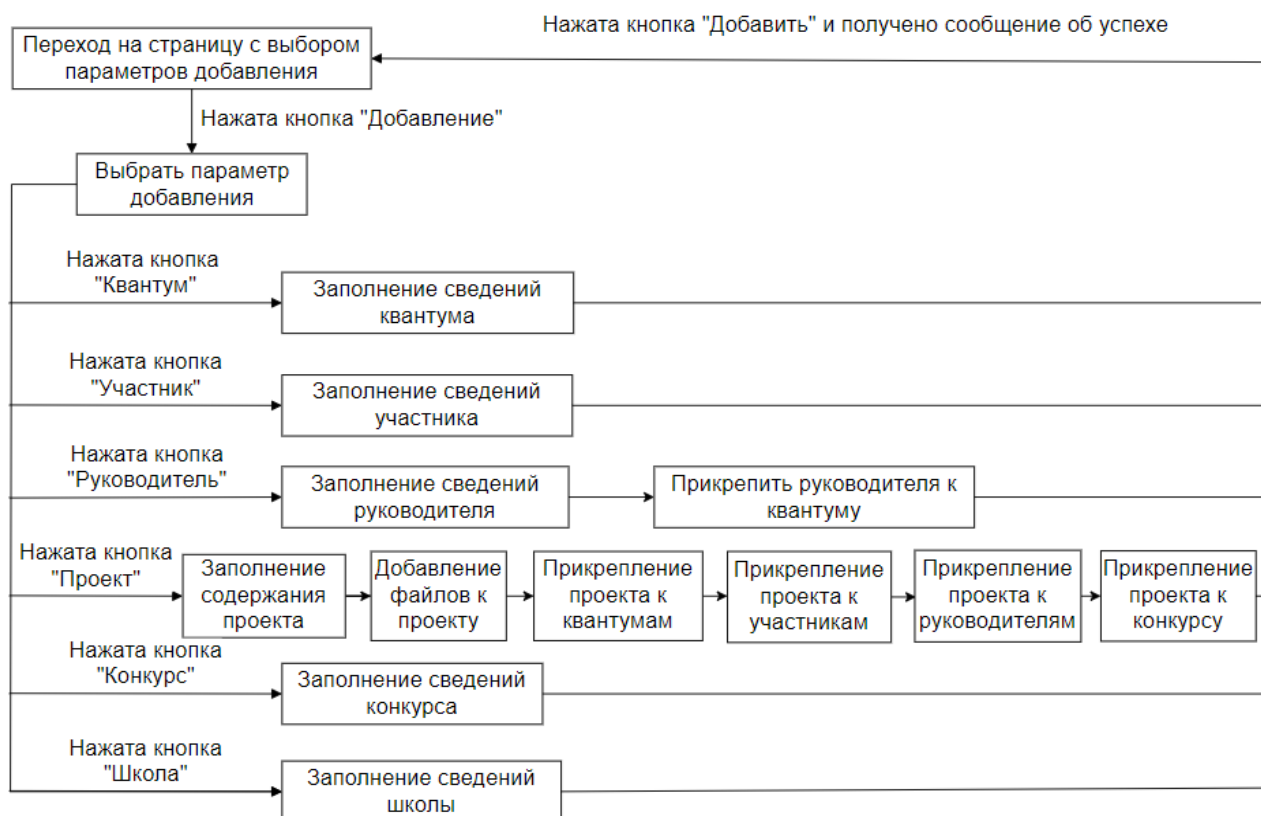


Рисунок 10 – Диаграмма переходов функции добавление

Прототипирование автоматизированной системы

Для создания прототипа, а именно для создания макетов пользовательского интерфейса и имитации функционала был использован онлайн-редактор FIGMA [5].

На основе анализа предметной области, разработанной базы данных и смоделированных диаграмм можно приступить к прототипу автоматизированной системы. В результате прототипирования были разработаны макеты главного экрана администратора и пользователя, макет входа в систему для администратора, а также макеты по осуществлению операции добавление. С помощью средств приложения Figma был создан интерактивный интерфейс для эмуляции взаимодействия пользователя с системой. Прототип может быть доработан на основе предпочтений заказчика.

Макет главного экрана администратора представлен на рис. 11.

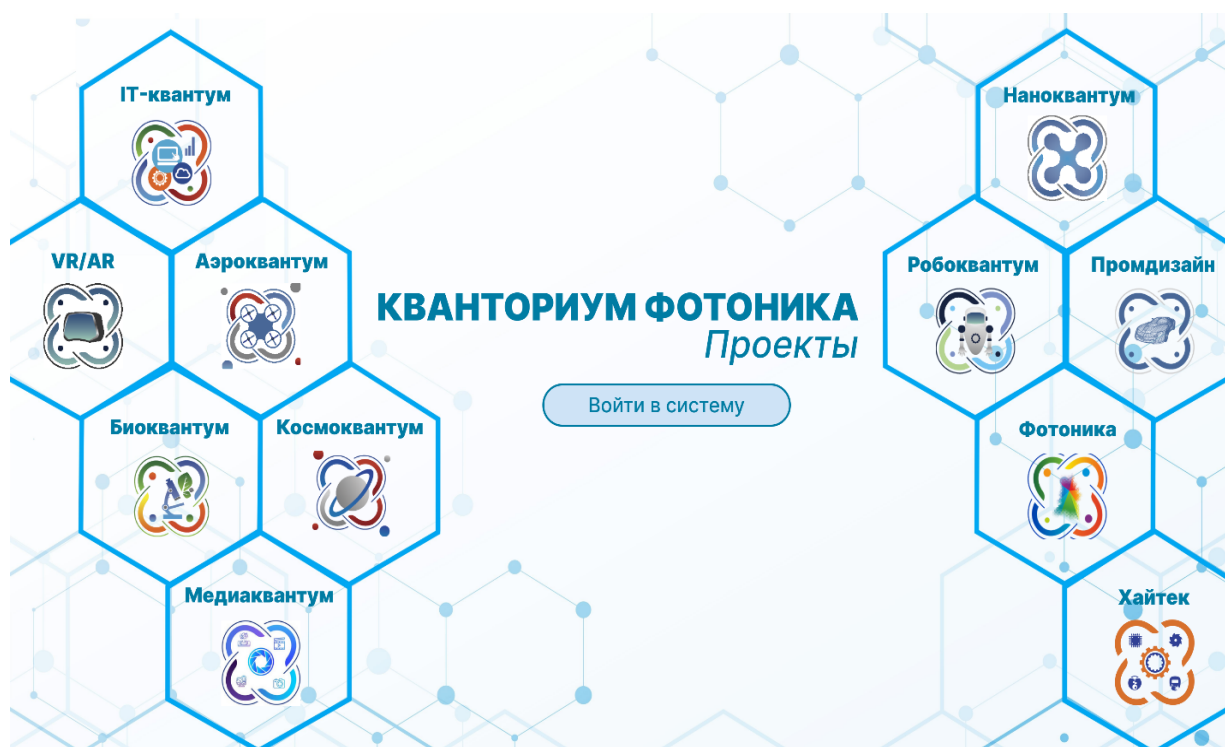


Рис. 11. Главный экран администратора

Макет страницы авторизации администратора представлен на рис. 12.



Рис. 12. Страница авторизации администратора

Макет старницы функций администратора представлен на рис. 13.

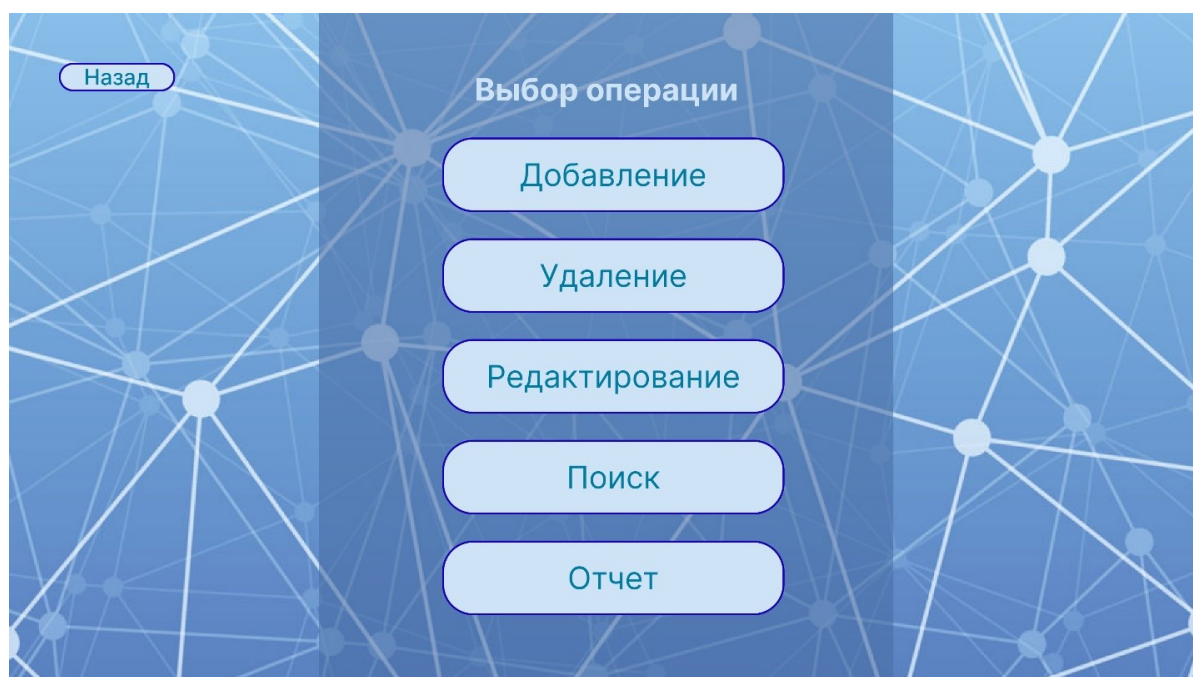


Рис. 13. Страница функций администратора

Заключение

Созданная модель автоматизированной системы позволит реализовать приложение по осуществлению регистрации проектов в детском технопарке «Кванториум Фотоника». В дальнейшем планируется разработка приложения, включающая в себя функционал по получению отчетности.

Библиографический список

1. Формы для заполнения отчетов по проектной деятельности, предоставленные директором детского технопарка «Кванториум Фотоника». URL: https://docs.google.com/spreadsheets/d/1VceJBXkhxg9NF_IoBwIJIQDyvofKn1gQ2rVcfQqKPQ4/edit?usp=sharing/ (дата обращения: 21.01.2024).
2. Проекты учащихся, открытая региональная ярмарка креативно-технологических проектов TOMSK 2022. URL: <https://kvantoriumproject.ru/projectsofopenfair2022/> (дата обращения: 21.01.2024).
3. DBDesigner [Электронный ресурс]. URL: <https://www.dbdesigner.net/> (дата обращения 10.01.24).
4. Diagrams.net [Электронный ресурс]. URL: <https://app.diagrams.net/> (дата обращения 10.01.24).
5. Figma [Электронный ресурс]. URL: <https://www.figma.com/> (дата обращения: 10.01.2024).

References

1. Forms for filling out reports on project activities provided by the director of the children's technopark "Quantorium Photonics". URL: https://docs.google.com/spreadsheets/d/1VceJBXkhxg9NF_IoBwIJIQDyvofKn1gQ2rVcfQqKPQ4/edit?usp=sharing/ (date of access: 01/21/2024).
2. Student projects, Open Regional fair of creative and technological projects TOMSK 2022. URL: <https://kvantoriumproject.ru/projects/openfair2022/> (date of access: 01/21/2024).
3. DBDesigner [Electronic resource]. URL: <https://www.dbdesigner.net/> (date of access: 01/10/2024).
4. Diagrams.net [Electronic resource]. URL: <https://app.diagrams.net/> (date of access: 01/10/2024).
5. Figma [Electronic resource]. URL: <https://www.figma.com/> (date of access: 01/10/2024).

DESIGNING AND PROTOTYPING AN APPLICATION TO SOLVE THE TASKS OF SUPPORTING THE PROJECTS OF THE CHILDREN'S TECHNOPARK «QUANTORIUM PHOTONICS»

Leshekhva Ekaterina A.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, leshehvaea@yandex.ru

Kushev Vadim O.

Perm State National Research University, 614068, Russia, Perm, 15, Bukireva str., vadimk@psu.ru

The basic theoretical provisions for the design and prototyping of an automated system for the purpose of registering projects of students of the children's technopark "Quantorium-Photonics" are considered. The subject and object of research are defined, the urgency of information system development is formulated, and the purpose and tasks to be performed in the course of design are set. The subject area was analyzed, concepts and their properties were defined. A conversation with the customer was conducted, during which the users of the system and the functionality of the application were determined. The analysis of existing solutions was carried out. A set of design and prototyping tools was identified during the study. The paper considers draw.io UML design tools, DBDesigner CASE tool for building ER diagram, and Figma online editor as a prototyping tool. As a result of designing the automated system, the interaction of users with it was described, the architecture of the application at the software and hardware level was described, an ER-diagram based on the subject area of the Quantorium was developed, and an interactive interface of the application was designed.

Keywords: children's technopark "Quantorium Photonics", Quantorium, registration of Quantorium projects, automated system.

ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНОЙ ИГРЫ В ЖАНРЕ ПЛАТФОРМЕР

Мальцев Владислав Леонидович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, vladmalcev6616@gmail.com

Кнутова Наталия Сергеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, talieknutova@gmail.com

В данной статье рассматривается процесс проектирования компьютерной игры в жанре платформер. Особое внимание уделяется игровым механикам и дизайну пользовательского интерфейса. Статья содержит обзор современных игр в жанре платформер и последующее выделение их ключевых особенностей. На основе анализа игр в жанре платформер предлагается Game Design Document, который содержит все ключевые аспекты игры для последующей её разработки. В статье также подробно рассматривается вопрос о выборе программного обеспечения которое поможет разработать проработанную и качественную игру, особенно внимание уделяется выбору игрового движка. Также в статье представлены прототипы пользовательского интерфейса и игрового процесса. Результаты, представленные в статье, могут быть полезны как для начинающих разработчиков компьютерных игр, так и для людей, которые просто играют в видеоигры и заинтересованы в процессе их разработки.

Ключевые слова: игра в жанре платформер, игровая механика, пользовательский интерфейс.

Первая игра в жанре платформер появилась в 1981 г. (Donkey Kong от Nintendo [1]) и, быстро завоевав популярность, определила развитие этого жанра на следующие десятилетия. К середине 90-х возможности техники позволили сделать игры более разнообразными с точки зрения механик и графики, появилась возможность использовать не только двухмерную, но и трёхмерную графику. Современные платформеры вобрали в себя многие черты других игровых жанров и обрели увлекательный динамичный игровой процесс. Исследования показывают [2], что игры подобного жанра могут быть полезны – они отлично развивают аналитическое и логическое мышление, зрительно-моторную координацию и помогают избавиться от негативных эмоций. Таким образом, разработка компьютерных игр в данном жанре является актуальной.

Для разработки игры были поставлены следующие задачи:

- рассмотреть игры жанра платформер,
- выделить основные черты жанра платформер,
- составить Game Design Document,

- придумать основные игровые механики,
- подобрать ПО для разработки.

Для проектирования были рассмотрены наиболее популярные и современные игры в жанре платформер и после была выделена основная особенность жанра.

Never song[3] – компьютерная игра в жанре 2D-платформер, разработанная студиями Atmos Games и Serenity Forge. Под управление игрока попадает мальчик Пит, по вине которого украли его подругу. Он винит себя, что не смог её уберечь и ставит своей целью спасти Рен. В игре нам предстоит решать различные головоломки с элементами простого платформинга, а также периодически сражаться с монстрами. Never song – небольшая игра, время её прохождения составляет 3 часа, что позволяет рассказать игроку интересную историю и не наскучить за время прохождения.

It Takes Two[4] – приключенческий боевик с элементами платформера. Игра разработана Hazelight Studios под изданием Electronic Arts. Играть предстоит за супружескую пару, которая близка к разводу. Чтобы сохранить брак, их дочь отправляет своих родителей в опасное приключение, которое они смогут пройти только объединив силы. Основной особенностью игры, как и прошлых игр Hazelight Studios является то, что в игре доступен только кооперативный режим и играть предстоит вдвоём. Работая в команде, игрокам предстоит решать головоломки и принимать участие в сражениях против обширного набора боссов, которые будут всячески мешать персонажам пройти дальше. В игре присутствует множество механик, которые вносят разнообразие в игровой процесс, например: полёты на спиннерах, отталкивание своего напарника при помощи магнита, катание на коньках и т.д. Хотя и среднее время прохождения игры составляет 14 часов, обширный набор механик не позволяет игрокам заскучать, а сюжет интересен до самого конца.

Trine 5: A Clockwork Conspiracy [5] – игра платформер-приключение, в которой нам выпадает роль героев, цель которых спасти мир Трины. На выбор нам даётся 3 героя, у каждого из них свои особенные способности. При помощи их комбинирования предстоит решать головоломки и поражать врагов. В игре доступно несколько игровых режимов: однопользовательский – игрок может свободно переключаться между героями и решать головоломки, многопользовательский – каждому игроку даётся определённый герой, тут уже надо будет действовать сообща чтобы продвигаться вперёд. Разработана игра студией Frozenbyte, а издателями выступили THQ Nordic.

Bread & Fred [6] – пиксельный 2d-платформер, в котором предстоит играть за двух пингвинов, цель которых взобраться на вершину горы. Персонажи связаны страховочной верёвкой, что затрудняет передвижение и заставляет игроков продумывать свои шаги наперёд и действовать сообща. Если один из игроков промедлит или поспешит с прыжком, то оба

персонажа рискуют упасть вниз и начать свой путь сначала. Игра поддерживает также и однопользовательский режим, тогда роль напарника сменит камень. В этом случае всё зависит от одного игрока, а камень можно использовать как груз для преодоления препятствий. Разработали игру SandCastles Studio под издательством Apogee Entertainment.

Проанализировав аналоги был сделан вывод, что управление персонажем является основной особенностью платформеров, так как весь игровой процесс завязан на передвижении персонажа по различным плоскостям. Поэтому необходимо что бы управление персонажем было отзывчивым и приятным. Для этого были разработаны базовые механики, которые помогут сделать управление персонажем лучше, а опыт игрока приятнее:

- раннее падение – игрок начинает падать, как только отпускается кнопка прыжка, завершая прыжок раньше времени. Благодаря этой механике, игрок может точно выбрать место для приземления, не снижая горизонтальной скорости;
- буфер прыжка – когда персонаж уже в воздухе, нажатие кнопки «прыжок» за мгновение до касания земли активирует прыжок, как только он приземлится. Преждевременное нажатие кнопки не приводит к тому, что входные данные не регистрируются контроллером;
- Coyote Time – прыжок можно запустить в течении нескольких кадров, после того, как персонаж вышел за край платформы. Данная механика прощает ошибки игрокам, которые хотят совершить прыжок на максимальную длину и откладывают нажатие кнопки прыжка до самого последнего момента.

При проектировании компьютерной игры был разработан Game Design Document. Последующее проектирование и разработка компьютерной игры будет осуществляться на основе данного документа. В Game Design Document можно выделить следующие основные пункты:

- название игры: Dream of two friends;
- жанр: Платформер, Инди, Приключение, 2D, Головоломка, СО-ОР (на одном компьютере);
- платформа: РС (клав./мышь, геймпад);
- время прохождения: 2–3 часа;
- целевая аудитория: Жители стран СНГ, Северной Америки, Европы;
- площадки публикации игры: Steam, VK Play.

Концепт игры

Игра «Dream of two Friends» расскажет историю про двух детей, которые покидают свой дом и отправляются в путешествие, чтобы спасти кота юной ведьмы из рук Лилит, которая похитила его. Только работая сообща юные путешественники смогут преодолеть все трудности и спасти кота Аббадона.

Игровой процесс

Основной особенностью игры от других платформеров является то, что как и в It Takes Two игрокам будет доступен только кооперативный режим. У каждого персонажа будут свои уникальные способности. Концепт персонажей представлен на рис. 1. Прототип игрового процесса на рис. 2.



Рис. 1. Концепт персонажей



Рис. 2. Прототип игры

При проектировании игры был разработан прототип меню игры, а именно «Главное меню» (рис. 3), «Новая Игра/Продолжить» (рис. 4). С его помощью игрок может перейти к основному игровому процессу, «Настройки» (рис. 5). В данном меню игрок может посмотреть управление персонажем или изменить настройки громкости.



Рис. 3. Главное меню



Рис. 4. Новая Игра/Продолжить

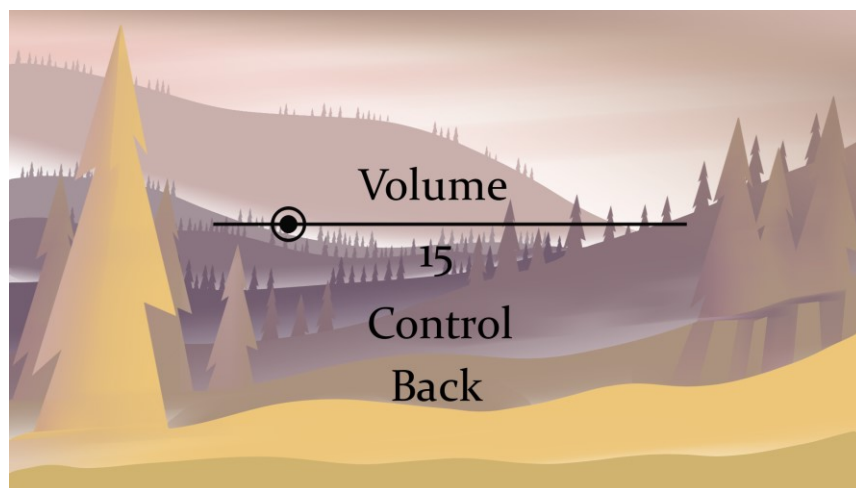


Рис. 5. Настройки

Для разработки игры было выбрано ПО, которое позволит быстро и качественно создавать игровые ассеты, анимации, механики и выполнять редактирование звуков:

- в качестве среды разработки компьютерных игр был выбран Unity [7]. Unity – это популярный игровой движок, который позволяет создавать игры для различных платформ, включая ПК, мобильные устройства, консоли и веб. Unity содержит множество инструментов для создания графики, анимации, звуков и игровой логики. Одними из главных преимуществ Unity являются большое количество обучающего материала и дружное комьюнити, к которому можно обратиться с вопросом;

- для создания графических элементов, таких как спрайты и фоновые изображения, был выбран графический редактор Krita [8]. Krita – бесплатный графический редактор, который предназначен для создания цифровых изображений, он поддерживает как растровую графику, так и векторную. Krita поддерживает множество форматов файлов, включая PSD, JPEG, PNG, и предоставляет широкий спектр инструментов для рисования, ретуширования и обработки изображений;

- для работы с игровыми звуками и музыкой выбор пал на Audacity [9]. Это бесплатный аудио редактор, который позволяет записывать, редактировать и обрабатывать звук. Audacity поддерживает множество форматов файлов, и предоставляет широкий выбор инструментов для обработки звука;

- в качестве среды для написания кода, была выбрана Visual Studio 2022 [10]. Visual Studio поддерживает множество языков программирования, включая C#, который является основным языком для разработки на Unity, и предоставляет широкий спектр для написания кода, отладки и тестирования.

- Spriter Pro [11] будет использоваться как программа для создания спрайтовых анимаций;

- для управления проектом Unity был выбран Github Desktop [12]. Это бесплатное приложение, которое позволяет загружать и выгружать проект в GitHub или другие службы размещения Git.

Компьютерных игр с каждым годом становится больше, а интерес к ним у игроков не уменьшается. Поэтому разработка игр является актуальной и востребованной. В данной статье представлен этап проектирования компьютерной игры в жанре платформер. Были рассмотрены игры аналоги на основе которых были выделены основные механики жанра платформер, составлен подробный дизайнерский документ, ссылаясь на который можно разработать игру. Также было выбрано программное обеспечение, которое необходимо для разработки качественного продукта. Все полученные наработки можно использовать в дальнейшей разработке.

Библиографический список

1. A brief history of the platformer [Электронный ресурс]. URL: <https://newsroom.activisionblizzard.com/p/a-brief-history-of-the-platformer> (дата обращения 14.05.2024).
2. Isabela Granic, Adam Lobel and Rutger C. M. E. Engels “The Benefits of Playing Video Games”.
3. Neversong [Электронный ресурс]. URL: <https://store.steampowered.com/app/733210/Neversong/> (дата обращения 15.05.2024).
4. It Takes Two [Электронный ресурс]. URL: <https://www.ea.com/en-gb/games/it-takes-two> (дата обращения 15.05.2024).
5. Trine 5: A Clockwork Conspiracy [Электронный ресурс]. URL: https://store.steampowered.com/app/1436700/Trine_5_A_Clockwork_Conspiracy/ (дата обращения 15.05.2024).
6. Bread & Fred [Электронный ресурс]. URL: https://store.steampowered.com/app/1607680/Bread_Fred/ (дата обращения 15.05.2024).
7. Unity [Электронный ресурс]. URL: <https://unity.com/> (дата обращения 16.05.2024).
8. Krita [Электронный ресурс]. URL: <https://krita.org/en/> (дата обращения 16.05.2024).
9. Audacity [Электронный ресурс]. URL: <https://www.audacityteam.org/> (дата обращения 16.05.2024).
10. Visual Studio 2022 [Электронный ресурс]. URL: <https://visualstudio.microsoft.com/ru/vs/> (дата обращения 16.05.2024).
11. Spriter Pro [Электронный ресурс]. URL: https://store.steampowered.com/app/332360/Spriter_Pro/?l=russian (дата обращения 16.05.2024).
12. Github Desktop [Электронный ресурс]. URL: <https://desktop.github.com/> (дата обращения 16.05.2024).

References

1. A brief history of the platformer [Electronic resource]. URL: <https://newsroom.activisionblizzard.com/p/a-brief-history-of-the-platformer> (date of access: 05/14/2024).
2. Isabela Granic, Adam Lobel and Rutger C. M. E. Engels “The Benefits of Playing Video Games”.
3. Neversong [Electronic resource]. URL: <https://store.steampowered.com/app/733210/Neversong/> (date of access: 05/15/2024).
4. It Takes Two [Electronic resource]. URL: <https://www.ea.com/en-gb/games/it-takes-two> (date of access: 05/15/2024).

5. Trine 5: A Clockwork Conspiracy [Electronic resource]. URL: https://store.steampowered.com/app/1436700/Trine_5_A_Clockwork_Conspiracy/ (date of access: 05/15/2024).
6. Bread & Fred [Electronic resource]. URL: https://store.steampowered.com/app/1607680/Bread__Fred/ (date of access: 05/15/2024).
7. Unity [Electronic resource]. URL: <https://unity.com/> (date of access: 05/16/2024).
8. Krita [Electronic resource]. URL: <https://krita.org/en/> (date of access: 05/16/2024).
9. Audacity [Electronic resource]. URL: <https://www.audacityteam.org/> (date of access: 05/16/2024).
10. Visual Studio 2022 [Electronic resource]. URL: <https://visualstudio.microsoft.com/ru/vs/> (date of access: 05/16/2024).
11. Spriter Pro [Electronic resource]. URL: https://store.steampowered.com/app/332360/Spriter_Pro/?l=russian (date of access: 05/16/2024).
12. Github Desktop [Electronic resource]. URL: <https://desktop.github.com/> (date of access: 05/16/2024).

COMPUTER GAME DESIGN IN THE GENRE OF PLATFORMER

Maltsev Vladislav L.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, vladmalcev6616@gmail.com

Knutova Natalia S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, talieknutova@gmail.com

This article discusses the process of designing a computer game in the platformer genre. Particular attention is paid to game mechanics and user interface design. The article contains an overview of modern platformer games and then highlights their key features. Based on the analysis of games in the platformer genre, a Game Design Document is proposed, which contains all the key aspects of the game for its subsequent development. The article also discusses in detail the issue of choosing software that will help develop a well-developed and high-quality game, especially attention is paid to the choice of a game engine. The article also presents prototypes of the user interface and gameplay. The results presented in the article can be useful both for beginning computer game developers and for people who simply play video games and are interested in the process of their development.

Keywords: platformer game, game mechanics, user interface.

ИСТОРИКИ ПЕРМСКОГО КРАЯ 19–20 вв.: ПРОЕКТ ПРОСОПОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Маткин Михаил Алексеевич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, matkinpmi@gmail.com

Аверин Сергей Игоревич

Пермский национальный исследовательский политехнический университет,
614000, Россия, г. Пермь, Комсомольский проспект, д. 29, averin-s-i@ya.ru

Работа описывает процесс проектирования, разработки и документирования информационной системы «Историки Прикамья 19–20 вв.». Определён предмет и объект исследования, сформулирована актуальность разработки информационной системы. Поставлены задачи, которые необходимо решить в ходе создания системы. Определены функциональные возможности, которыми должна обладать разрабатываемая информационная система. В результате проектирования и документирования информационной системы описано поведение системы и взаимодействие с ней пользователя. В результате разработки был описан принцип работы системы и взаимодействия её компонентов.

Ключевые слова: просопография, историко-ориентированные информационные системы, историки, Пермский край.

Историки Прикамья являются неотъемлемой частью многолетней истории Пермского края и истории как науки. Эти люди сыграли ключевую роль в исследовании и понимании исторического прошлого региона, что, в свою очередь, оказало влияние на решение современных проблем и планирование будущего. Создание информационной системы, систематизирующей исторические данные об этих людях и их работах, а также позволяющей создание коллективного портрета данного профессионального сообщества, сделает исследование истории Пермского края более удобным и доступным. Вышесказанное является аргументом актуальности разработки системы, посвящённой историкам Прикамья.

В рамках работы рассмотрено проектирование, разработка и документирование информационной системы, которая будет совмещать в себе информацию об историках Прикамья, об их биографии и научных достижениях.

Объект исследования: создание коллективных портретов профессиональных сообществ за счет разработки информационной системы. Предмет исследования – просопографические информационные системы, посвященные историкам.

Целью работы является проектирование, разработка и документирование информационной системы «Историки Прикамья 19–20 вв.».

Для достижения цели необходимо выполнить ряд задач:

- 1) изучить предметную область;
- 2) рассмотреть аналогичные информационные системы;
- 3) выбрать средство моделирования и разработки информационной системы;
- 4) смоделировать систему;
- 5) разработать систему;
- 6) написать документацию к системе.

Реализуемая система является информационно-справочной по характеру обработки данных, а также историко-ориентированной по сфере применения, в частности, просопографической. Просопография – это историческая дисциплина, которая изучает биографии групп людей, объединённых какими-либо общими чертами, такими как географическое положение, социальный статус, вероисповедание и т.д. В нашем случае изучаемой группой являются Историки Прикамья. Для наполнения системы данными использовались библиографические справочники, например, книга, вышедшая в 2016 г. «Профессора Пермского университета 1916–2016» [2], в которой кратко представлены биографии всех ученых университета. Из нее были отобраны биографии профессоров истории.

Всего в системе было выделено три роли пользователей: незарегистрированный пользователь, зарегистрированный пользователь и администратор. Любой пользователь в системе способен просматривать опубликованную информацию, проводить поиск по параметрам, а также проводить просопографическое исследование по найденной выборке. Зарегистрированный пользователь и администратор способны создавать, просматривать и отменять собственные заявки на изменение опубликованных каталогов. Администратор способен принимать и отклонять чужие заявки, а также менять остальные данные в системе, например, список стран, религий, учебных заведений и т.д. К этим данным также относится информация о пользователях. В системе не предусмотрена самостоятельная регистрация, администратор сам добавляет новых пользователей в систему.

Информация об историках и их работах представлена в виде карточек (рис. 1). При создании заявки, кроме основных данных, можно указать комментарий к заявке, что упростит процесс принятия новой информации. При отклонении заявки администратор может оставить ответный комментарий, который увидит заявитель. В системе происходит копирование информации в списке заявок для сохранения полной истории изменения данных через механизм заявок.

ФИО	Аверина Нина Фёдоровна	
Пол	жен	
Дата рождения	17-06-1935	
Место рождения	Ильинский, Пермская область	
Статус	Жив	
Страны	Россия СССР Австралия	
Образование	Место: ПермГУ	
Научная сфера	История Пермского края Краеведение Журналистика Литературоведение История	
Род деятельности	Историк Краевед Литературовед Журналист Писатель Поэт Книговед	
Проекты/Литература	название	История пермской книги
	Дата	1989 год
	О работе	Автор: Нина Аверина ISBN: 5-7625-0145-0 Издательство: Пермское книжное издательство Язык: Русский В 1792 году в Перми была напечатана первая на Урале книга. Автор воссоздает историю книгоиздания в Прикамье, анализирует традиции, которые развивает в своей деятельности основанное в 1939 году Пермское книжное издательство.
	Темы	Пермь Россия
<div>Изменить</div> <div>Удалить</div>		

Рис. 1. Пример карточки историка

Система, над которой велась работа, является веб-приложением, построенным на основе клиент-серверной архитектуры. В качестве сервера используется компьютер с установленной на нём операционной системы Windows 10. Диаграмма компонентов-развёртывания представлена на рис. 2.

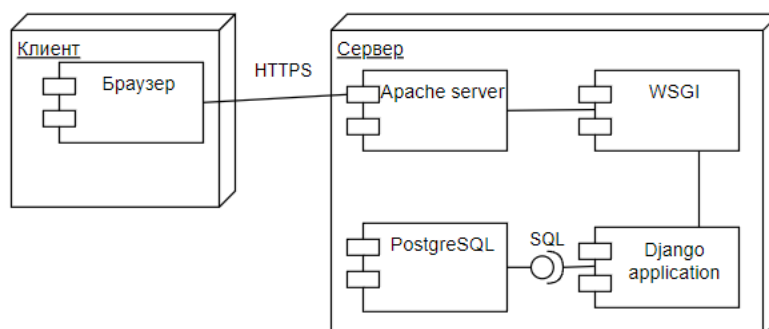


Рис. 2. Диаграмма компонентов-развёртывания

На стороне клиента пользователь взаимодействует с веб-приложением через браузер, отправляя запросы на сервер. Для разработки клиентской части приложения использовались языки HTML, CSS и JavaScript, без использования фреймворков.

Для создания серверной части были выбраны язык программирования Python, фреймворк Django, веб-сервер Apache и база данных PostgreSQL. Выбор языка Python обусловлен тем, что Python известен благодаря различным библиотекам, специализирующимся на анали-

зе данных, от чего дальнейшее развитие системы с добавлением сложной обработки данных будет более удобным и простым. Django, в свою очередь, предлагает встроенные инструменты для администрирования, функции для защиты веб-приложения, а также возможность представлять данные в системе через модели (Object Relational Mapping – объектно-реляционное отображение), позволяя работать с базами данных на более высоком уровне абстракций, не беспокоясь о SQL-запросах. Использование веб-сервера Apache для развёртывания Django приложения в операционной системе Windows хорошо зарекомендовал себя в интеграции с модулем `mod_wsgi`, который реализует интерфейс WSGI (Web Server Gateway Interface).

Ниже представлен полный процесс работы серверной части:

- 1) когда пользователь запрашивает страницу, запрос поступает на веб-сервер Apache;
- 2) Apache передаёт запрос модулю `mod_wsgi`, который в свою очередь передаёт его Django-приложению;
- 3) Django определяет, какое `view` (представление/функция) должно быть вызвано на основе URL, используя URL Dispatcher;
- 4) `view` обрабатывает запрос, взаимодействует с моделями, передаёт данные в шаблон (template), либо выполняет любые другие действия, описанные в функции;
- 5) шаблон генерирует HTML-ответ, используя данные, предоставленные `view`;
- 6) сгенерированный HTML-ответ возвращается обратно в Apache через модуль `mod_wsgi`;
- 7) Apache отправляет ответ, составленный из HTML страниц и статических файлов пользователю.

В качестве документации к системе прилагается руководство пользователя, описывающее пользование системой для каждой роли пользователя, а также руководство разработчика, описывающее использованные инструменты, инструкцию по настройке веб-сервера и запуска системы, а также список используемых маршрутов в системе и их описание.

В результате была проведена работа по проектированию, разработки и документированию информационной системы «Историки Прикамья 19–20 вв.». Работа включала в себя проектирование и документирование системы, а также описание принципа работы главных компонентов. Созданная система позволит не только собирать данные об историках Прикамья 19–20 вв., но также анализируя данные, создав на основе выбранных признаков 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. унт-т. Пермь, 2016. 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. P.80–85. (In Russ.)
2. Kosticyn V. I. Professora Permskogo universiteta 1916–2016. Perm State University. Perm, 2016. 336 p. (In Russ.)

HISTORIANS OF THE PERM KRAI OF THE 19TH AND 20TH CENTURIES: PROJECT OF PROSOPOGRAPHIC INFORMATION SYSTEM

Matkin Mikhail Alekseevich

Perm State University, 15, Bukireva st., Perm, 614068, Russia, matkinpmi@gmail.com

Averin Sergey Igorevich.

State National Research Politechnical University of Perm,
Komsomolsky Av. 29, Perm, Russia, 614000, averin-s-i@ya.ru

The work describes the process of designing, developing and documenting an information system “Historians of the Prikamye of the 19th and 20th centuries”. The subject and object of research are defined, the relevance of the information system development is formulated. The tasks that need to be solved during the creation of the system have been set. The functional capabilities that the information system under development should have are defined. As a result of designing and documenting an information system, the behavior of the system and the user's interaction with it are described. As a result of the development, the principle of operation of the system and the interaction of its components was described

Keywords: prosopography, historical-oriented information systems.

ПРОЕКТИРОВАНИЕ МОДУЛЯ «МУНИЦИПАЛЬНАЯ СОБСТВЕННОСТЬ» ИНФОРМАЦИОННОЙ СИСТЕМЫ ОТДЕЛА МУНИЦИПАЛЬНОГО ОБРАЗОВАНИЯ ДЕПАРТАМЕНТА ИМУЩЕСТВЕННЫХ ОТНОШЕНИЙ ГОРОДА ПЕРМИ

Моздакова Анастасия Александровна, Ракина Валерия Денисовна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, mozdakova.nastya@yandex.ru

В данной статье рассмотрены аналоги информационных систем Департаментов имущественных отношений, выделены преимущества и недостатки этих систем и определены функциональные требования к проектируемой информационной системе. Данные требования были согласованы с заказчиком и прописаны в техническом задании. Информационная система позволяет на сайте в виде карты посмотреть муниципальную собственность города Перми, которая по типу строения выделена разноцветными указателями. При нажатии на указатель пользователю выводится 3D-модель здания. Для моделируемой системы были определены средства для проектирования, язык программирования и среда разработки, а также средства для хранения данных и средства для 3D-моделирования. Работа содержит описание логики работы информационной системы, а также примерный прототип интерфейса. Данная система предназначена для аудитории, проживающей в Перми и Пермском крае.

Ключевые слова: информационная система, муниципальная собственность, Департамент имущественных отношений, веб-сайт.

Введение

Муниципальная собственность [1] – это огромный имущественный комплекс, в который входят различные виды имущества: земля, движимое и недвижимое имущество. Наличие эффективной информационной системы для управления муниципальным имуществом становится необходимостью. Такая система позволяет повысить качество управления собственностью, но и сделать доступной информацию для населения [2]. На данный момент, реестр муниципальной собственности представлен на сайте Департамента образования города Перми только в виде таблиц [1]. Эти данные являются важными для местной власти в качестве полезного инструмента для мониторинга и управления активами. Для граждан они представляют удобный доступ к информации, который позволяет легко выяснить кто является владельцем определённого земельного участка, что предоставляет доступность данных. Для инвесторов муниципальная собственность открывает возможность для инвестиций и сотрудничества с муниципалитетом.

Разработка модуля информационной система «Муниципальная собственность» является актуальной задачей, так как она направлена на систематизацию данных о муниципальном имуществе. Это позволит упорядочить информацию, а также визуализировать её, пред-

ставляя какие объекты являются муниципальными зданиями и что из них относится к недвижимой собственности, а что функционирует как муниципальное учреждение. Цель данной работы спроектировать модуль к информационной системе Департамента имущественных отношений города Перми для отдела о работе с муниципальной собственностью.

Методы исследования

В данном разделе рассмотрим существующие Департаменты с муниципальной собственностью, а также выберем программное обеспечение и язык программирования.

Веб-сайт Департамент управления муниципальной собственностью администрации Города Томска [3]. На сайте подробно изложены функции и задачи Департамента, обеспечивая ясность в управлении городскими активами. С помощью понятных карточек, классифицированных по разным категориям, пользователи могут легко находить необходимую информацию. С другой стороны, при наличии обширной информации, сайт сталкивается с ограничением представленной в единой таблице. Это дает неудобство для пользователей, которые стремятся к более глубокому изучению.

Веб-сайт Департамент имущественных отношений администрации муниципального образования городской округ город-курорт Сочи Краснодарского края [4]. Сайт представляет четкое представление о функциях и задачах Департамента, что способствует пониманию в управлении имущества курортного города. Особенностью сайта является интерактивная карта, которая доступна через специальную кнопку. Особый интерес представляет раздел «Объекты социальной сферы», где представлена информация о школах и детских садах. При нажатии на указатель открывается 3D-модель здания, где отображена краткая информация о каждом объекте. Несмотря на функциональность и визуальную привлекательность, сайт не предоставляет данные о реестре муниципального имущества.

Веб-сайт Департамент по управлению муниципальным имуществом (ДУМИ) города Екатеринбург [5]. На сайте подробно освещены функции и задачи Департамента. Сайт предлагает удобную вкладку муниципального имущества, где собраны все необходимые подразделы с данными. Одной из важных частей будет являться реестр имущественных отношений, представленный в формате Excel таблицы. Должным интересом представляет интерактивная карта, которая разделена на две части: одна демонстрирует детские сады, а другая – общеобразовательные учреждения. При нажатии на указатели появляется краткая информация о здании, а также панорама с помощью интеграции с Яндекс Картами.

Рассмотрев и проанализировав различные веб-сайты, можно установить основные задачи и требования к системе:

- пользователь при открытии веб-сайта должен видеть карту с указателями;

- указатели на карте должны иметь разные цвета, по которым можно определить тип здания;
- пользователю при нажатии на указатель должно отобразиться окно с 3-х мерным изображением здания;
- в окне с 3-х мерным изображением может быть построенное здание и строящееся;
- база данных системы должна хранить информация о зданиях: тип, координаты, название, адрес и 3D-модель.

В качестве средства моделирования было принято решение использовать Diagrams.net. Это технологический набор для создания приложений для построения диаграмм и наиболее широко используемое конечным пользователем программное обеспечение для построения диаграмм на основе браузера [6]. Причинами выбора Diagrams.net стало, что программа является бесплатной, имеет возможность ручного рисования и шаблоны диаграмм.

В качестве языка программирования был выбран JavaScript [7] – динамический язык, который используется для обеспечения интерактивности на веб-сайтах. В качестве средства разработки было выбран фреймворк React.js, который поддерживается JavaScript.

В системе будет использована база данных сайта Департамента образования города Перми, а пока будет создана тестовая база данных, и для ее создания была выбрана PostgreSQL [8] – это мощная объектно-реляционная система баз данных с открытым исходным кодом, которая использует и расширяет язык SQL в сочетании со множеством функций, позволяющих безопасно хранить и масштабировать самые сложные рабочие нагрузки с данными.

Blender [9] – профессиональное программное обеспечения, в основе которой лежит креативность и 3D-технологии. Главным фактором выбор стало, что программа является бесплатной, а также имеет обширный набор инструментов для моделирования и хорошую производительность.

Проектирование информационной системы

Рассмотрим систему, представленную в виде UML-диаграмм для визуализации структуры и взаимодействия. Роль в данной системе выполняет пользователь. Пользователь может взаимодействовать с системой тем, что имеет возможность «Просмотр карты» сделать «Выбор указателя на карте», а также «Просмотр 3D-модели дома». В данной системе областью применения является веб-сайт, на котором находится интерактивная карта. Диаграмма прецедентов, описывающий данную систему, представлена ниже (см. рис. 1).

Описание взаимодействия пользователя с веб-сайтом, на котором изображена интерактивная карта представлена в виде диаграммы деятельности (см. рис. 2). Пользователь начинает с официального сайта Департамента имущественных отношений города Перми и

затем переходит на наш сайт. Человек просматривает интерактивную карту на которой нанесены разноцветные указатели, представляющие различные здания. Затем он может выбрать интересующий его на карте указатель для просмотра интересующего его здания. После выбора точки сайт автоматически отправляет запрос на получение 3D-модели выбранного здания из базы данных. Как только модель загружается на сайт, она отображается пользователю в отдельном окне.

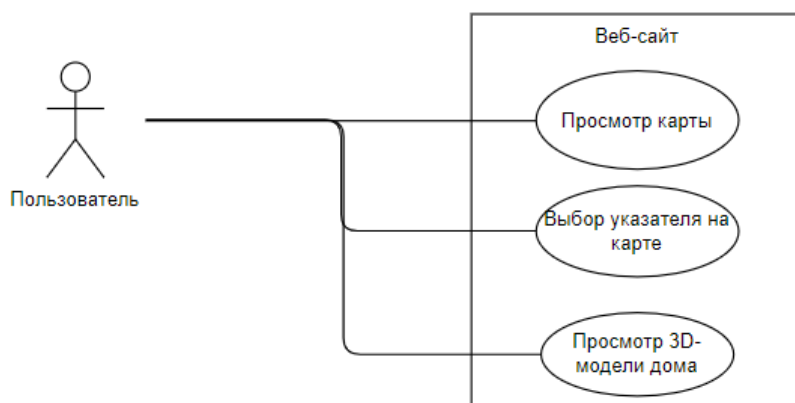


Рис. 1. Диаграмма прецедентов

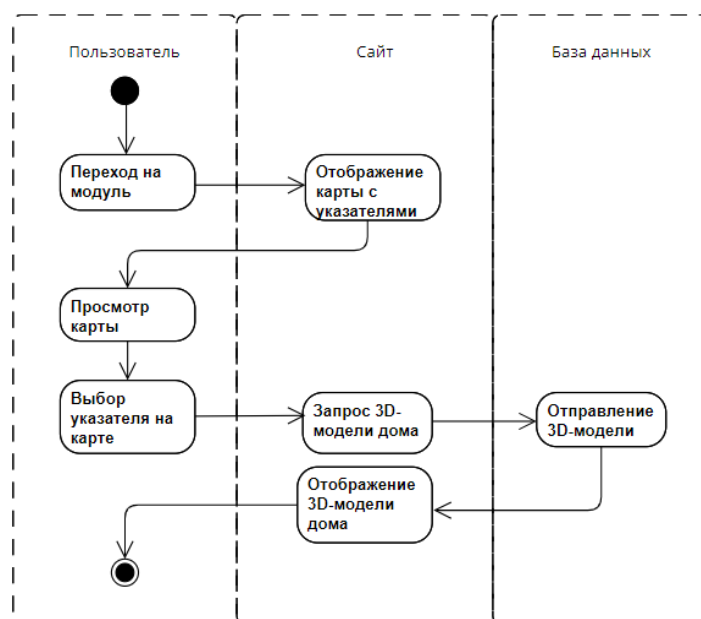


Рис. 2. Диаграмма деятельности

Логика базы данных представлена в виде ER-диаграммы и изображена ниже (см. рис. 3). Для этого было созданы 2 таблицы, которые отображают сущности для хранения в базе данных. Таблица Building_type. Представляет собой хранение данных о типах зданий. Содержит поля building_type_id – первичный ключ, name – название типа здания, color – цвет указателя. Таблица Building. Представляет хранение данных о всей информации о здании.

Содержит поля `building_id` – первичный ключ, `building_type_id` – ID типа здания, внешний ключ, который отсылается к таблице `Building_type`, `coordinates` – координаты нахождения здания на карте, `3d_model` – 3D-модель здания, которое отображает построенное здание или здание в процессе постройки, `name` – название здания, `adres` – адрес здания.

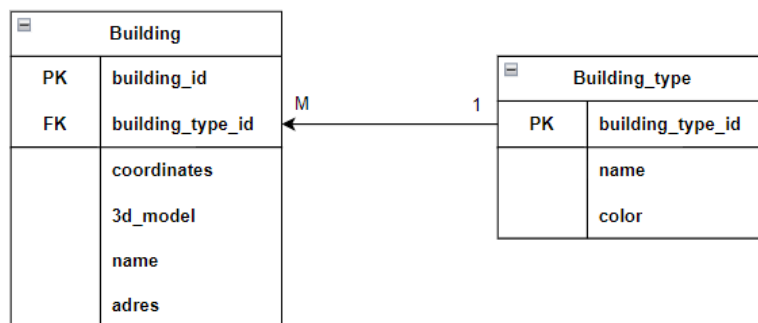


Рис. 3. ER-диаграмма базы данных

Ниже изображен прототип веб-сайта «Муниципальная собственность», на который мы перешли с сайта Департамента образования города Перми (см. рис. 4). На нем размещена интерактивная карта Перми с разноцветными указателями, указывающих на различные здания, находящиеся в муниципальной собственности. Цвет указателя показывает к какому типу относится данная постройка. Так точка оранжевого цвета показывает размещение образовательных учреждений. Указатель зеленого цвета представляет детские сады. Точка розового цвета показывает недвижимое имущество, в которое входят: жилые дома, жилые помещения, места общего пользования, нежилое здание и т.д.

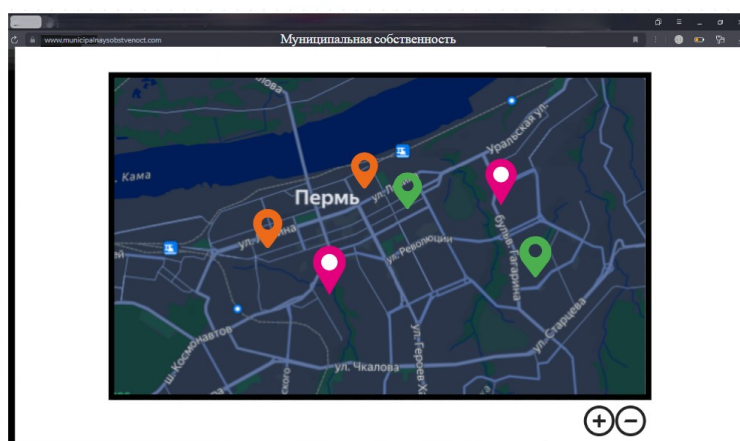


Рис. 4. Прототип веб-сайта

Ниже изображен прототип всплывающего окна, когда пользователь нажал на цветной указатель (см. рис. 5). В этом окне изображена 3D-модель построенного или здания, которое

находится в процессе постройки. Также показывается информация о том какое название имеет данное здание, выводит его координаты, адрес и тип здания.

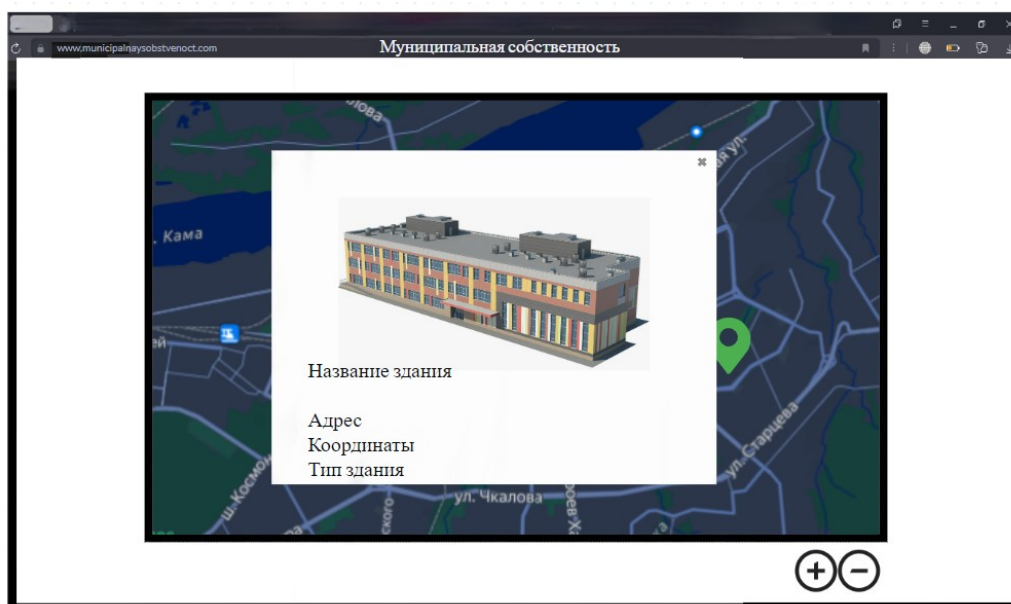


Рис. 5. Прототип всплывающего окна с 3D-моделью

Заключение

Спроектированная система будет полезна Департаменту имущественных отношений, а также людям, проживающим в Перми и Пермском крае, так как позволит наглядно посмотреть какие объекты являются муниципальными зданиями.

Библиографический список

1. Муниципальная собственность Пермь – Муниципальное образования город Пермь [Электронный ресурс]. URL: <https://www.gorodperm.ru/actions/property/> (дата обращения: 22.05.2024).
2. Дубровский В. Ж., Рожков Е. В. Использование новых цифровых технологий для управления земельными участками (на примере города Перми) // Информационные технологии в управлении и экономике. 2023. №.1. С. 58–71.
3. Официальный сайт муниципального образования 'Город Томск': Департамент управления муниципальной собственностью администрации Города Томска [Электронный ресурс]. URL: <https://admin.tomsk.ru/pgs/6c> (дата обращения: 22.05.2024).
4. Администрация города Сочи – Структура администрации [Электронный ресурс]. URL: <https://sochi.ru/gorodskaya-vlast/administratsiya-goroda/structure/> (дата обращения: 22.05.2024).

5. Власть Екатеринбурга – Официально – Официальный портал [Электронный ресурс]. URL: <https://екатеринбург.рф/официально/власть?item=1172> (дата обращения: 22.05.2024).
6. Diagrams.net [Электронный ресурс]. URL: <https://www.drawio.com/> (дата обращения: 25.05.2024).
7. Хавербек М. Выразительный JavaScript: 2-е изд. Сан-Франциско: No Starch Press, Inc., 2017. 436 с.
8. PostgreSQL [Электронный ресурс]. URL: <https://www.postgresql.org/about/> (дата обращения: 23.05.2024).
9. Blender [Электронный ресурс]. URL: <https://www.blender.org/> (дата обращения: 23.05.2024).

References

1. Munitsipal'naya sobstvennost' Perm' – Munitsipal'noe obrazovaniya gorod Perm' [Electronic resource]. URL: <https://www.gorodperm.ru/actions/property/> (In Russ.).
2. Dubrovsky V. Zh., Rozhkov E. V. Ispol'zovanie novykh tsifrovyykh tekhnologiy dlya upravleniya zemel'nymi uchastkami (na primere goroda Permi) // Informatsionnye tekhnologii v upravlenii i ekonomike. 2023. No.1. Pp. 58–71. (In Russ.)
3. Ofitsial'nyi sait munitsipal'nogo obrazovaniya 'Gorod Tomsk': Departament upravleniya munitsipal'noi sobstvennost'yu administratsii Goroda Tomsk [Electronic resource]. URL: <https://admin.tomsk.ru/pgs/6c> (In Russ.).
4. Administratsiya goroda Sochi – Struktura administratsii [Electronic resource]. URL: <https://sochi.ru/gorodskaya-vlast/administratsiya-goroda/structure/> (In Russ.).
5. Vlast' Ekaterinburga – Ofitsial'no – Ofitsial'nyi portal [Electronic resource]. URL: <https://екатеринбург.рф/официально/власть?item=1172> (In Russ.).
6. Diagrams.net [Electronic resource]. URL: <https://www.drawio.com/>.
7. Haverbeke M. Vyrazitel'nyi JavaScript: 2-e izd. San Francisco: No Starch Press, Inc., 2017. 436 p. (In Russ.)
8. PostgreSQL [Electronic resource]. URL: <https://www.postgresql.org/about/>.
9. Blender [Electronic resource]. URL: <https://www.blender.org/>.

**DESIGN OF THE "MUNICIPAL PROPERTY" MODULE
FOR THE INFORMATION SYSTEM OF THE MUNICIPAL EDUCATION
DEPARTMENT OF THE PROPERTY RELATIONS DEPARTMENT
OF THE CITY OF PERM**

Mozdakova Anastasia A., Rakina Valeria D.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, mozdakova.nastya@yandex.ru

The article examines analogs of information systems for the Departments of Property Relations, highlights the advantages and disadvantages of these systems and defines the functional requirements for the designed information system. These requirements were agreed upon with the customer and outlined in the technical specification. The information system allows users to view municipal property in the city of Perm on a website in the form of a map, with different colored indicators to represent the type of structure. When you click on an indicator, it displays a 3D model of the building. Specific tools for design, programming language, development environment, data storage means, and 3D modeling tools were determined for the modeling systems. The work includes a description of the logic of the information system's operation and a rough prototype interface. This system is intended for an audience residing in Perm and the Perm region.

Keywords: information system, municipal.

**ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ
ИНФОРМАЦИОННОЙ СИСТЕМЫ
«ЭКОЛОГИЧЕСКИЙ ТУРИЗМ ПЕРМСКОГО КРАЯ»***

Низаметдинов Ростислав Эдуардович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, rostislav.nizametdinov@mail.ru

Аверин Сергей Игоревич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, averin-s-i@yandex.ru

Рассматриваются этапы разработки информационной системы, позволяющей делиться информацией между туристами, занимающимся экологическим туризмом на территории Пермского края. Определены инструменты используемые в процессе создания веб-приложения. Разобран процесс создания интерактивной карты, которая используется как основной способ отображения информации, хранящейся в базе данных, к которой имеет доступ система. Подробно описана функция, которая позволяет зарегистрированным пользователям добавить в приложение информацию по туристическим маршрутам, которые они считают нужными для закрепления. В моменте описания данной функции, уточняется способ, по которому вычисляется примерная длина маршрута в километрах. В статье упоминается процесс организации модерации и разделение ролей разных пользователей на несколько групп, ограничивая возможности для каждой роли.

Ключевые слова: экологический туризм, туризм, Пермский Край, проектирование, разработка, документирование.

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 [5].

При этом у туристов нет возможности корректно делиться пройденными маршрутами, которые они разработали сами, или узнали из другого источника. Для обеспечения туристов

© Низаметдинов Р. Э., Аверин С. И., 2024

* 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).

специальным инструментом, который позволяет обмениваться знаниями было разработано веб-приложение.

В процессе создания данного предложения были составлены требования и в соответствии этим требованиям в последствии были спроектированы прототипы, описывающие функциональные возможности. [1] Для этого использовались инструменты StarUML, GIMP, Paint.NET [2, 3, 4].

При разработке логики веб-приложения использовались скриптовые языки PHP и JavaScript, для интерфейса пользователя был задействован язык разметки HTML. База данных, из которой приложение берет информацию, управляется с помощью реляционной СУБД MySQL. Взаимодействие между инструментами обеспечивается с помощью программной среды AMPPS.

Для работы в MySQL применялось система: phpMyAdmin, предоставляющее графический интерфейс, облегчающий работу при создании базы данных и таблиц.

Чтобы разработать интерактивную карту, используемую приложением, были задействованы инструменты:

- ArcMap – геоинформационная система, позволяющая открывать для просмотра карты, хранящиеся в формате шейпфайлов;
- NPM – менеджер пакетов, входящий в Node.js, используемый для установки нужных пакетов;
- ТороJSON – формат данных в котором хранится информация о границах Пермского Края, а также одноименный пакет, используемый для получение нужной информации из используемой карты;
- d3.js – библиотека на языке JavaScript, применяемая для визуализации данных;
- в качестве текстовых редакторов использовались редакторы Notepad++ и Visual Studio Code.

Разработка интерактивной карты состояла из нескольких этапов.

Для создания контура будущей карты использовались шейпфайлы, которые хранят в себе информацию об административном делении государств. Файлы, в которых хранятся эти данные, были взяты из открытого источника «Natural Earth».

Для отображения полученных данных в приложении используется библиотека d3.js, которая позволяет превратить информацию из шейпфайлов в карту, имеющую форму Пермского Края. Для удобства пользователей, эта карта центрируется на координатах, присущих городу «Пермь», 58.01 СШ и 56.22 ВД. Масштаб увеличивается в 5 000 раз от изначального. Для отображения используется проекция Меркатора.

Основной функцией в системе является модуль, позволяющий пользователям предлагать свои собственные туристические маршруты, которые затем принимаются или отклоняются администратором. Данная функция состоит из следующей последовательности действий:

- вначале пользователем вводится количество контрольных точек, желаемое название, а также описание маршрута;
- в случае если пользователь оставляет поле с именем маршрута пустым, маршруту автоматически дается имя вида «Маршрут: 'пользователь'», где 'пользователь' заменяется на логин человека, который предлагает маршрут;
- для вычисления длины каждого из отрезков применяется формула гаверсинуса, используемая для того, чтобы узнать длину отрезка на сфере:

$$d = 2R * \arcsin \sqrt{\sin^2\left(\frac{\varphi_1 - \varphi_2}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2\left(\frac{\lambda_1 - \lambda_2}{2}\right)}.$$

Чтобы предоставить посетителю сайта определённые права и возможности, используется модуль, отвечающий за возможность авторизоваться в приложении или создать новый аккаунт. Он состоит из двух форм, которые можно заполнить и которые в последствии передаются скрипту, написанному на языке PHP.

В случае если пользователь решит не авторизоваться в системе, то его возможности ограничены только просмотром карты, просмотром объектов и маршрутов в виде списков, а также изучение информации о каждом памятнике экологического туризма или экологической тропе на отдельной странице.

Роль «User» вместе с просмотром всей справочной информации, посвященной природному туризму Пермского края, дает возможность предложить свой собственный маршрут, который может быть принят администратором и в последствии добавлен на карту.

У администратора также есть право просмотреть информацию о предложенных маршрутах, которую решили добавить пользователи в систему. Вся информация представлена в виде списка, который проверяется администратором вручную.

Все страницы системы состоят из нескольких контейнеров, демонстрирующих данные в соответствии с нужным форматом.

В верхней части страницы расположена шапка веб-приложения. В левой части расположен флаг Пермского края, в правой части, расположена кнопка, используемая пользователем, чтобы авторизоваться в системе, или перейти на свою страницу профиля, если он уже авторизован. В случае если пользователь уже находится в профиле, то на этом месте расположена кнопка выхода из профиля.

Ниже кнопок расположено меню, представленное строкой, на которой находятся ссылки ведущие на определенные страницы приложения:

- карта – ведет на начальную страницу с интерактивной картой, на которой расположена все объекты и маршруты;
- объекты – ведет на страницу, на которой объекты представлены в виде списка;
- маршруты – ведет на страницу, на которой маршруты, добавленные пользователями и администратором представлены в виде списка.

Затем располагается основная часть. В левой части показываются данные авторизованного аккаунта. В центре находится основная информация, которая зависит от страницы, на которой в данный момент располагается пользователь. В правой части находятся кнопки, позволяющие перейти на страницы добавления маршрута или проверки предложенных маршрутов.

Разработанная система сможет существенно упростить туристам обмен информацией, позволяя наглядно демонстрировать маршруты на карте. Приложение также имеет возможность способствовать развитию туризма в крае.

Библиографический список

1. Низаметдинов Р. Э., Аверин С. И. Проектирование и документирование информационной системы «Экологический туризм пермского края» // Актуальные проблемы математики, механики и информатики 2023. С. 272–279.
2. StarUML 2 [Электронный ресурс]. URL: <https://staruml.io> (дата обращения: 05.05.2024).
3. Paint.NET – Free Software for Digital Photo Editing [Электронный ресурс]. URL: <https://getpaint.net/index.html> (дата обращения: 13.05.2024).
4. GIMP – GNU Image Manipulation Program [Электронный ресурс]. URL: <https://www.gimp.org> (дата обращения: 13.05.2024).
5. Правительство Пермского края: Распоряжение от 1 апреля 2022 года N82-рп. «Об утверждении Стратегии развития туризма в Пермском крае на период до 2035 года». С. 16–17.

References

1. Nizametdinov R. E., Averin S. I. Designing and documenting the information system "Ecological tourism of the Perm region" // Aktual'nye problemy matematiki, mekhaniki i informatiki 2023. P. 272–279. (In Russ.)
2. StarUML 2 [Electronic resource]. URL: <https://staruml.io>.
3. Paint.NET – Free Software for Digital Photo Editing [Electronic resource]. URL: <https://getpaint.net/index.html>.

4. GIMP – GNU Image Manipulation Program [Electronic resource]. URL: <https://www.gimp.org>.

5. Pravitel'stvo Permskogo Kraya: Rasporyazhenie ot 1 aprelya 2022 goda N82-rp. "Ob utverzhdenii Strategii razvitiya turizma v Permskom krae na period do 2035 goda". P. 16–17. (In Russ.)

DESIGNING, DEVELOPMENT AND DOCUMENTING AN INFORMATION SYSTEM « ECOLOGICAL TOURISM OF PERM REGION»

Nizametdinov Rostislav E.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, rostislav.nizametdinov@mail.ru

Averin Sergey I.

Perm State University, 15, Bukireva st., Perm, 614680, Russia, averin-s-i@yandex.ru

The stages of the development of an information system allowing the sharing of information between tourists engaged in ecological tourism in the Perm territory are considered. The tools used in the process of creating a web application are defined. The process of creating an interactive map is analyzed, which is used as the main way to display information stored in a database that the system has access to. The function is described in detail, which allows registered users to add information to the application on tourist routes that they consider necessary for fixing. At the moment of describing this function, the method by which the approximate route length in kilometers is calculated is specified. The article mentions the process of organizing moderation and dividing the roles of different users into several groups, limiting the possibilities for each role.

Keywords: ecological tourism, tourism, Perm Region, design, development, documentation.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ВОССТАНОВЛЕНИЕ ТРЕХМЕРНОЙ МОДЕЛИ СОСУДОВ СЕРДЦА ПО ДВУХМЕРНЫМ ИЗОБРАЖЕНИЯМ КОРОНАРОГРАФИИ»

Осипов Александр Денисович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, sashaosipovden2003@gmail.com

Барулина Марина Александровна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, mab@psu.ru

Рассматривается проектирование информационной системы для воссоздания 3D-модели сердца по двухмерным изображениям коронарографии. Проводится исследование существующих информационных систем с подобным назначением. На основе анализа этих систем формулируются основные функциональные требования к разрабатываемому настольному приложению. Производится обоснованный выбор средств для проектирования и реализации информационной системы, а также средств для прототипирования пользовательского интерфейса. Для проектирования системы используются программное CASE-средство, поддерживающие моделирование на UML – Visual Paradigm. Для создания прототипа интерфейса используется Figma. Для хранения данных в роли СУБД выступает PostgreSQL. Для реализации серверной части системы будут применяться языки программирования Python, C++, C# и фреймворк Django, также будет использоваться контейнеризация каждой части сервера, а взаимодействие с клиентом и распределение нагрузки будет осуществляться с помощью веб-сервера Nginx. Клиентская часть будет написана на фреймворке Flutter.

Ключевые слова: трехмерная модель, сосуды сердца, коронарография, информационная система, проектирование.

Введение

Коронарография и мультиспиральная компьютерная томография (МСКТ) сердца являются основными методами обследования, предоставляющими собой “золотой стандарт” диагностики проблем с сердечными артериями. Первый метод предоставляет двумерные рентгенографические изображения артериальных сосудов, в то время как второй метод дает трехмерное изображение. Трехмерная модель артериальных сосудов более информативна для диагностики и планирования хирургического вмешательства и дальнейшей реабилитации, но существует проблема засвечивания изображений МСКТ из-за скопления солей кальция в сердечных артериях. Это, очевидно, снижает диагностическую ценность МСКТ. Поэтому задача восстановления 3D моделей сосудов сердца по данным коронарографии имеет важное значение для повышения качества медицинских операций и перехода к персонифи-

цированной медицине. Таким образом, возникает необходимость разработки новой информационной системы, способной восстанавливать 3D модели сосудов сердца по результатам коронарографии и формулировать рекомендации по проведению хирургического вмешательства и послеоперационного восстановления пациента.

Объектом исследования настоящей работы является автоматизация процесса восстановления 3D моделей сосудов сердца и формирования рекомендаций.

Предмет исследования – информационная система для восстановления 3D моделей сосудов сердца, их анализа и формирования рекомендаций по лечению.

Целью работы является проектирование информационной системы для восстановления 3D моделей сосудов сердца по данным коронарографии, анализу этой модели и генерации рекомендаций по лечению.

Для достижения обозначенной цели были поставлены и решены следующие задачи:

- провести сравнительный анализ существующих информационных систем для восстановления 3D моделей сосудов сердца;
- провести обзор инструментального ПО;
- построить логическую структуру и функциональную модель информационной системы;
- разработать прототип пользовательского интерфейса информационной системы.

Анализ предметной области

Коронарография – это малоинвазивное исследование артерий сердца, при котором в них вводится контрастное йодсодержащее вещество, делающее артерии видимыми под рентгеном. Процедура проводится для оценки состояния сердечных артерий, выявления стенозов (сужений), их количества, протяженности и расположения (рис. 1) [1].



Рис. 1. Снимок коронарографии

По данным ВОЗ и Всемирной федерации сердца, ежегодно от сердечно-сосудистых заболеваний (ССЗ) умирают от 17 до 18,6 миллионов человек, что составляет около 31% всех случаев смерти в мире (рис. 2) [2, 3]. Снижение смертности от ССЗ является одной из основных задач. Коронарография играет важную роль в диагностике и лечении ССЗ, предоставляя критически важную информацию о состоянии сердечных артерий.

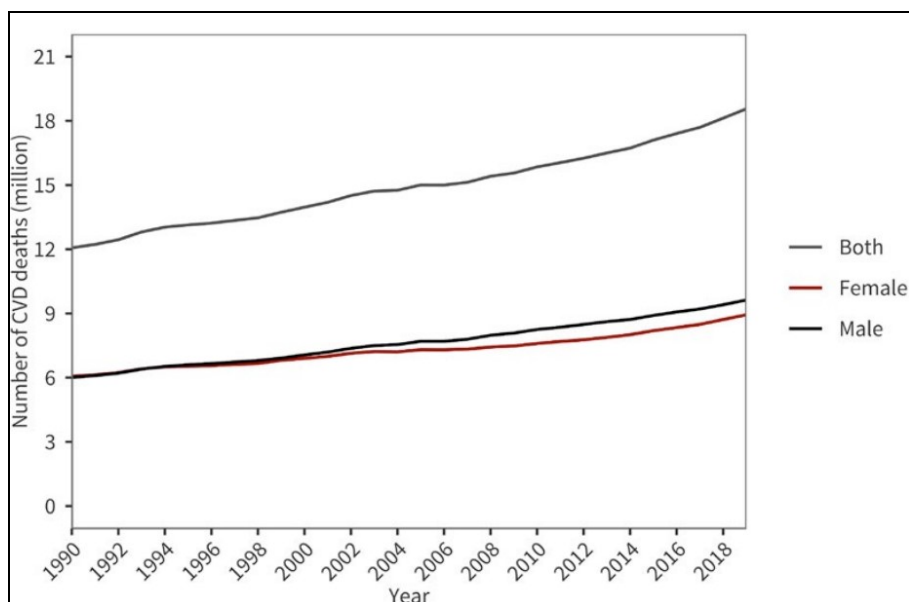


Рис. 2. Тенденция смертности от ССЗ

Снимки коронарографии, как и других медицинских исследований, хранятся в формате DICOM – медицинский стандарт данных, который содержит графическую информацию и метаданные к ней.

Анализ аналогов существующих информационных систем

Были рассмотрены частичные аналоги разрабатываемой системы их преимущества и недостатки, данные аналоги можно разделить на 3 типа:

1) приложения для исследования динамического состояния сосудов – Dynamic Coronary Roadmap (DCR) [4]. Предоставляет в режиме реального времени изображение артерий, что повышает эффективность процедур;

2) приложения для просмотра DICOM файлов с данными исследования сосудов сердца – Vidar Dicom Viewer [5]. Позволяет просматривать 2D и 3D медицинские изображения, производить простые расчеты и работать с цветокоррекцией;

3) приложения на основе Искусственного интеллекта (ИИ) для диагностики проблем с сердцем – CORONAROGRAPHY.AI [6]. С помощью ИИ делает прогноз патологий по анамнезу пациента и электрокардиограмм.

Преимущества данных программ позволяют им успешно использоваться в медицинской области. Однако, главным недостатком систем является отсутствие функционала для восстановления 3D-моделей сосудов сердца по изображениям коронарографии, что говорит о важности создания планируемой системы.

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

- 1) разработать функционал создания, сохранения и загрузки проекта – структурно-логически связанных исходных данных и результатов их обработки в различных компонентах, разрабатываемой ИС;
- 2) предоставить удобный функционал для работы с изображениями коронарографии одного пациента;
- 3) разработать функционал создания, сохранения, удаления, загрузки файлов проекта с 3D-моделями сосудов;
- 4) возможность моделирования гемодинамики, установки стента или шунта на модели;
- 5) предоставление информации о проблемных местах в сосудах и информации о возможном лечении;
- 6) Реализовать регистрацию врачей в ИС для работы с пациентами, и администраторов для контроля за всей системой.

Выбор средства для проектирования

Для проектирования было выбрано ПО Visual Paradigm. Этот инструмент поддерживает нотацию UML, генерацию кода для языков программирования, и скриптов для баз данных. Visual Paradigm удобен в использовании, поддерживает работу на разных устройствах и доступен бесплатно с небольшими функциональными ограничениями [7].

Выбор средства для прототипирования

Для прототипирования интерфейса информационной системы был выбран сервис Figma. Это онлайн-сервис, который поддерживает работу на всех компьютерных и мобильных ОС, а также предоставляет все необходимые функции для создания прототипов интерфейса, поддерживает работу в режиме реального времени и имеет бесплатный тарифный план с ограничениями по количеству проектов, но не по функциональным возможностям [8].

Языки программирования

Для реализации компонентов ИС были выбраны Python, C++, и C# [9, 10]. Эти языки отличаются высоким уровнем абстракции, читабельностью кода, обширным набором библиотек и фреймворков для работы. Они являются кроссплатформенными и хорошо подходят для задач, связанных с изображениями, графикой и нейронными сетями.

Фреймворк для серверной части

Для написания серверной части выбран фреймворк Django. Этот высокоуровневый веб-фреймворк на Python включает множество встроенных функций, таких как ORM для работы с БД, интеграция с PostgreSQL, поддержка REST API и WebSocket для реализации реального времени и асинхронного программирования [11].

Фреймворк для клиентской части

Для разработки клиентской части приложения был выбран фреймворк Flutter, разработанный Google как мультиплатформенный фреймворк. Flutter позволяет создавать производительные пользовательские интерфейсы для мобильных платформ, для веб-приложений и для десктопных операционных систем. Flutter поддерживает взаимодействие с серверной частью через REST API и WebSocket, а также обеспечивает современный и удобный интерфейс для пользователя. К преимуществам можно также отнести большое количество разработанных для этого фреймворка виджетов, и поддержку java-подобного языка программирования Dart [12].

Система управления базами данных (СУБД)

Для хранения данных выбрана PostgreSQL. Эта объектно-реляционная СУБД поддерживает сложные запросы и транзакции, имеет высокую масштабируемость, надежность и поддерживает различные типы данных. PostgreSQL обладает необходимым функционалом для работы с различными типами данных и поддержкой концепции ACID [13].

Docker

Для автоматизации развёртывания, масштабирования и управления приложениями используется Docker. Контейнеризация позволяет упаковать приложение и все его зависимости в единый переносимый пакет, который можно запустить в любой среде, обеспечивая единообразие и независимость от инфраструктуры [14].

Nginx

Nginx – это высокопроизводительный веб-сервер, а также обратный прокси-сервер для HTTP, HTTPS и других протоколов, который также может работать как балансировщик нагрузки и веб-сервер.

По принципу работы Nginx отличается от традиционных веб-серверов своей асинхронной и событийно-ориентированной архитектурой, которая позволяет ему обрабатывать множество соединений параллельно, делая его эффективным для высоконагруженных систем [15].

Проектирование информационной системы

Были определены виды пользователей информационной системы и доступные для них функциональные возможности. В системе будет два вида пользователей: хирург-кардиолог и администратор. Функциональные возможности для каждого вида пользователей, представлены с помощью диаграммы прецедентов (рис. 3).



Рис. 3. Диаграмма прецедентов

Для понимания того, как система будет реализована и размещена на физических устройствах, необходимо изучить её структуру и взаимосвязи. Диаграмма развёртывания (рис. 4) показывает взаимодействие и размещение компонентов системы.

ИС будет основана на клиент-серверной архитектуре. Клиентское приложение на Flutter взаимодействует с сервером через интернет, используя HTTPS и WebSocket.

На сервере установлен NGINX, выполняющий роль обратного прокси и балансировщика нагрузки. Серверное приложение на Django работает в Docker-контейнере, взаимодействуя с базой данных PostgreSQL, также в Docker. Сервер также включает файловое хранилище для сохранения и обработки файлов.

Структура базы данных системы моделирования сосудов сердца организована в виде ER-диаграммы (рис. 5). Она включает в себя ключевые сущности и их атрибуты, а также связи между ними. Использование такой базы данных позволит улучшить управление медицинскими данными и обеспечит необходимое функционирование системы.

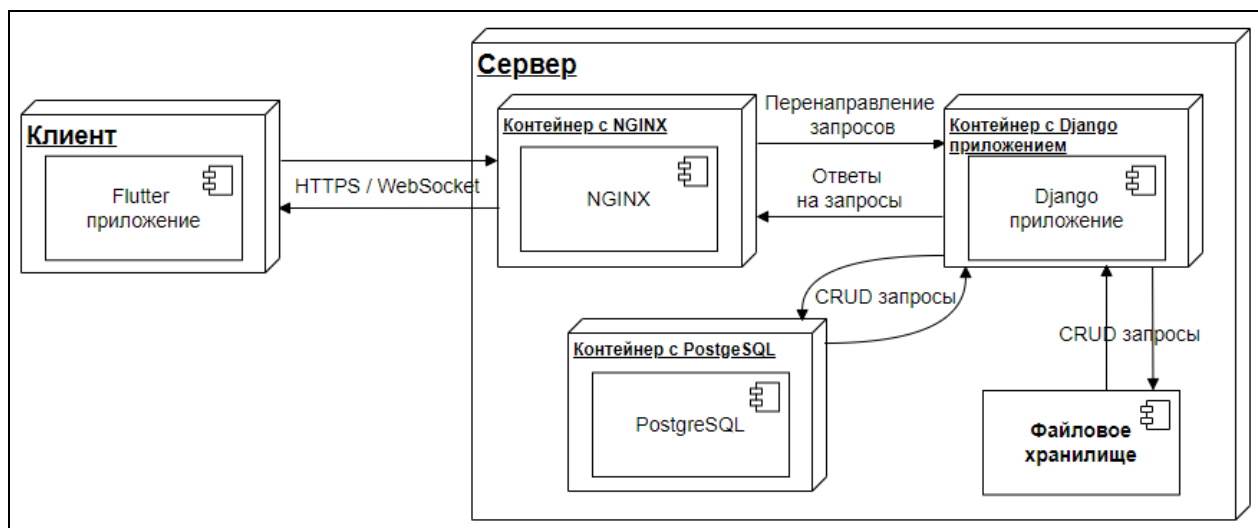


Рис. 4. Диаграмма развёртывания ИС

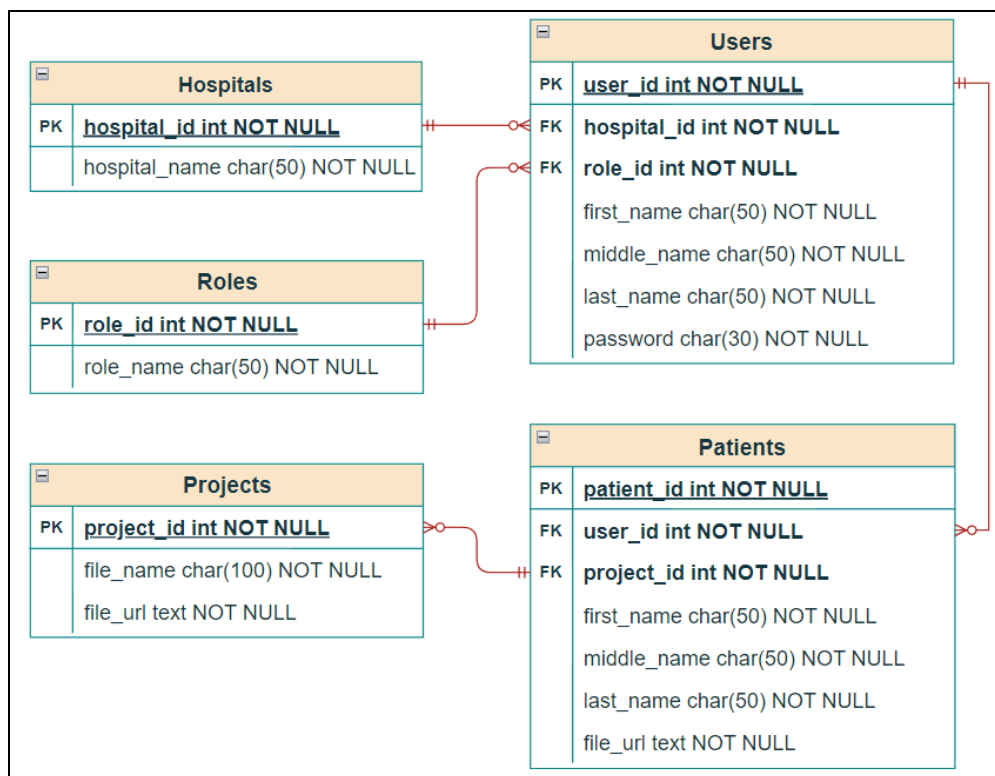


Рис. 5. Структура базы данных

Проектирование прототипа интерфейса

На основе построенных моделей был спроектирован пользовательский интерфейс настольного приложения ИС для 3D моделирования артерий. Создание происходило в Figma с использованием некоторых готовых шаблонов.

При запуске приложения пользователю показывается страница авторизации, где отображаются поля ввода логина и пароля, кнопка авторизации и переход на страницу регистрации. Успешный ввод данных приводит пользователя на главную страницу приложения. На рисунке 6 представлена визуализация страницы авторизации.

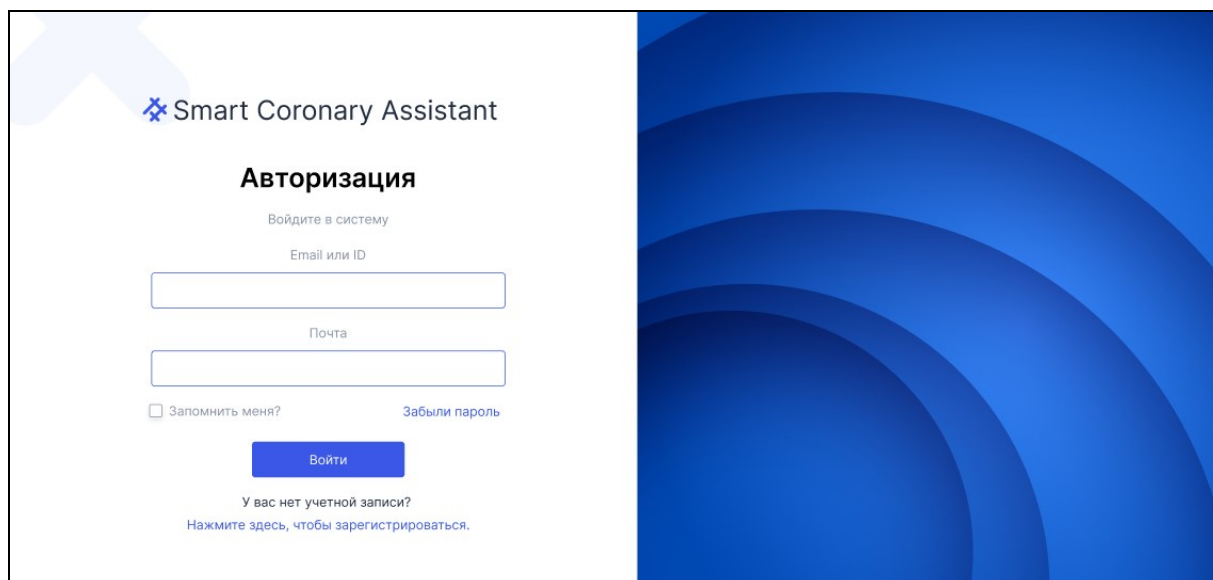


Рис. 6. Страница авторизации

Главная страница приложения отображает список пациентов в виде таблицы с указанием ФИО пациента, названия проекта, ФИО врача и индикатора завершенности проекта, в левой части расположено меню с разделами команд (см. рис. 7).

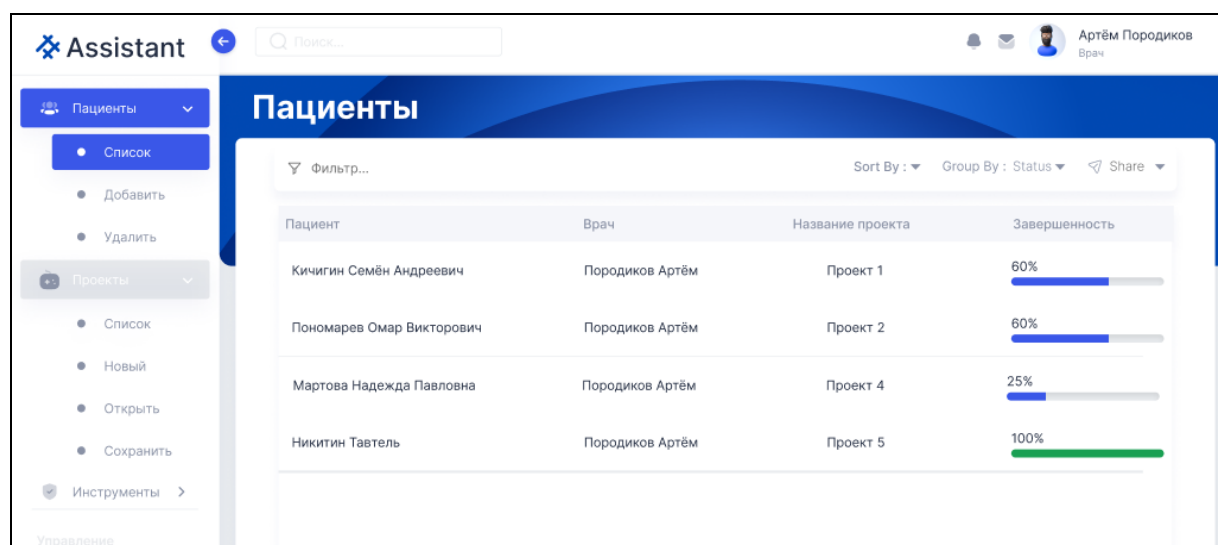


Рис. 7. Главная страница

Для каждого пациента может быть создано несколько проектов. В меню «Проекты» можно выбрать существующий проект или создать новый. В ходе работы над проектом его можно сохранить. Страница проекта включает меню инструментов, название проекта и данные о пациенте, а основную часть занимают модели проекта. Каждый проект состоит из моделей, каждая из которых объединяет 5 инструментов моделирования. Статус выполнения и редактирование инструмента отображены правее названия модели (см. рис. 8).



Рис. 8. Страница проекта

На странице инструмента 3D-модели пользователь может выбирать артерии, изменять их границы и создавать 3D-модель артерий. Он размечает лишь одну проекцию, на остальных с помощью ИИ дорисовываются соответствующие точки и границы артерий, и в итоге с помощью математических методов создается 3D-модель выбранных артерий. Данная страница показана на рис. 9.

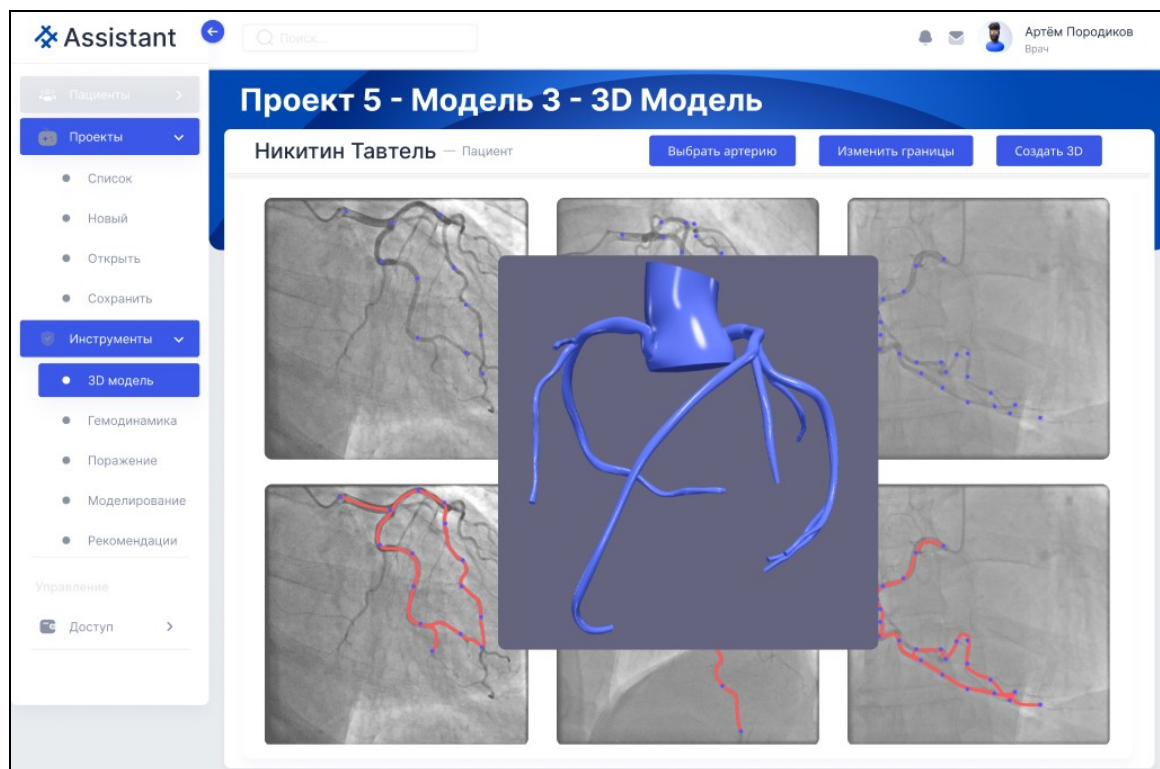


Рис. 9. Страница создания 3D-модели

Страница гемодинамики и поражения позволяет настраивать параметры моделирования и запускать модуляцию кровотока. Обработанные ИИ области поражения отображаются красным цветом и указывается процент функционирования данной области (см. рис. 10).

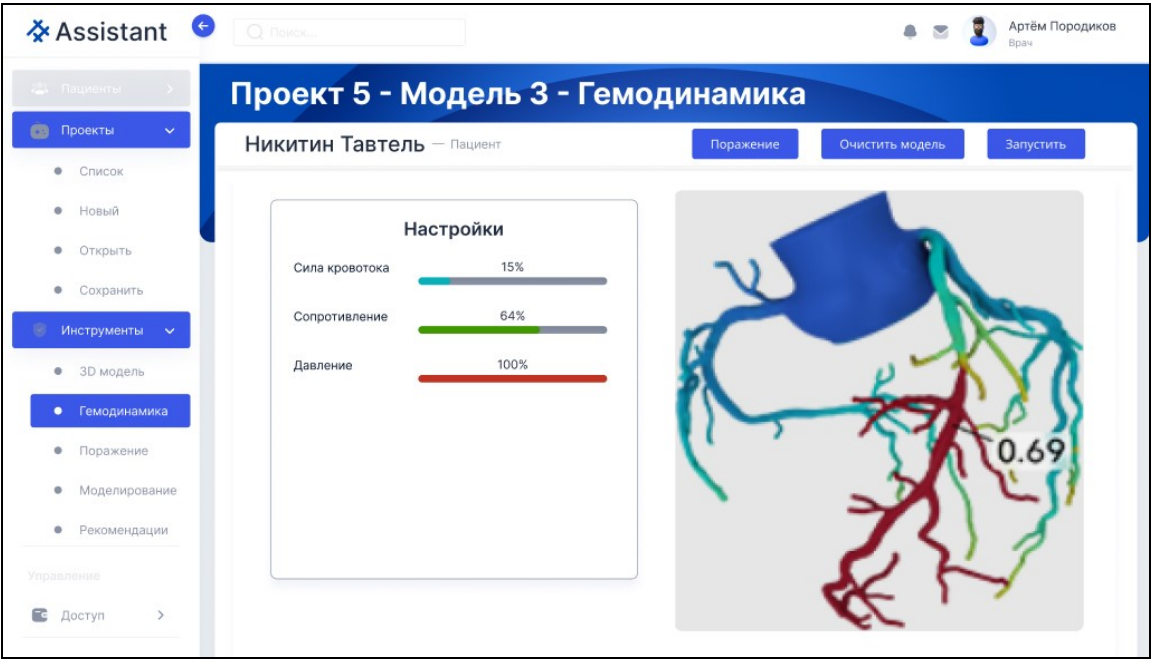


Рис. 10. Страница гемодинамики и поражения

Страница моделирования шунтов и стентов позволяет создавать шунты или стенты, очищать модель и запускать модуляцию с отображением возможных рисков при использовании конкретного метода. На рисунке 11 показана данная страница.

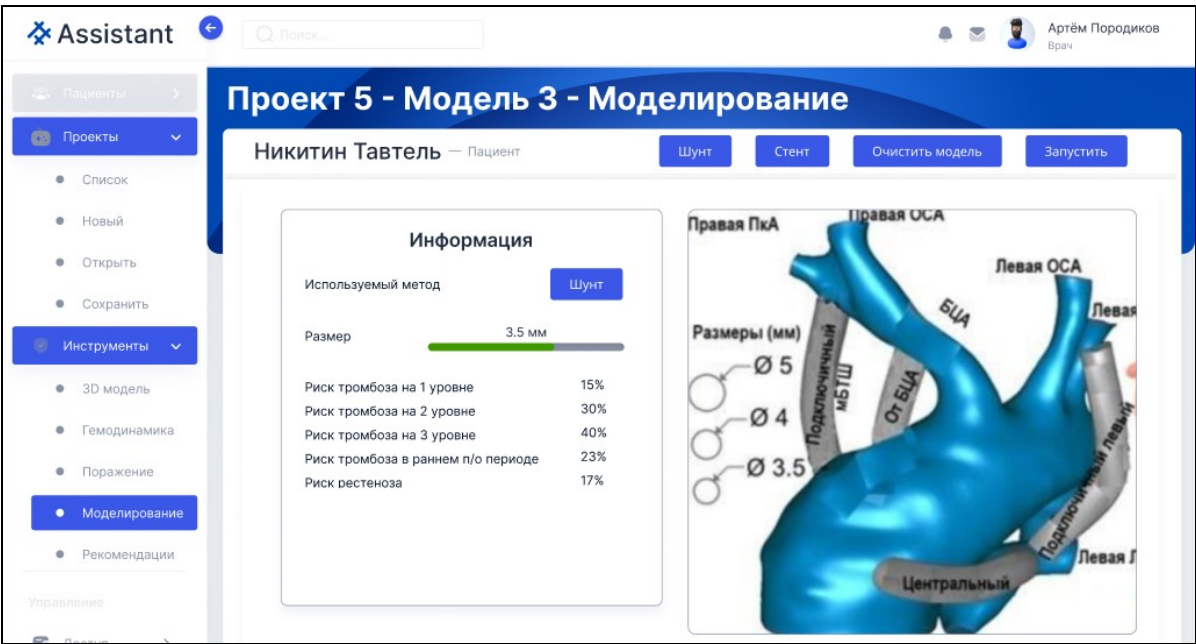


Рис. 11. Страница моделирования шунтов и стентов

Страница рекомендаций по лечению пациента включает текст, созданный ИИ на основе анализа полученного моделирования, где показаны все проделанные варианты, риски по ним и наилучший метод с наименьшими рисками. Также пользователь может распечатать или редактировать текст рекомендаций. Пример страницы показан на рис. 12.

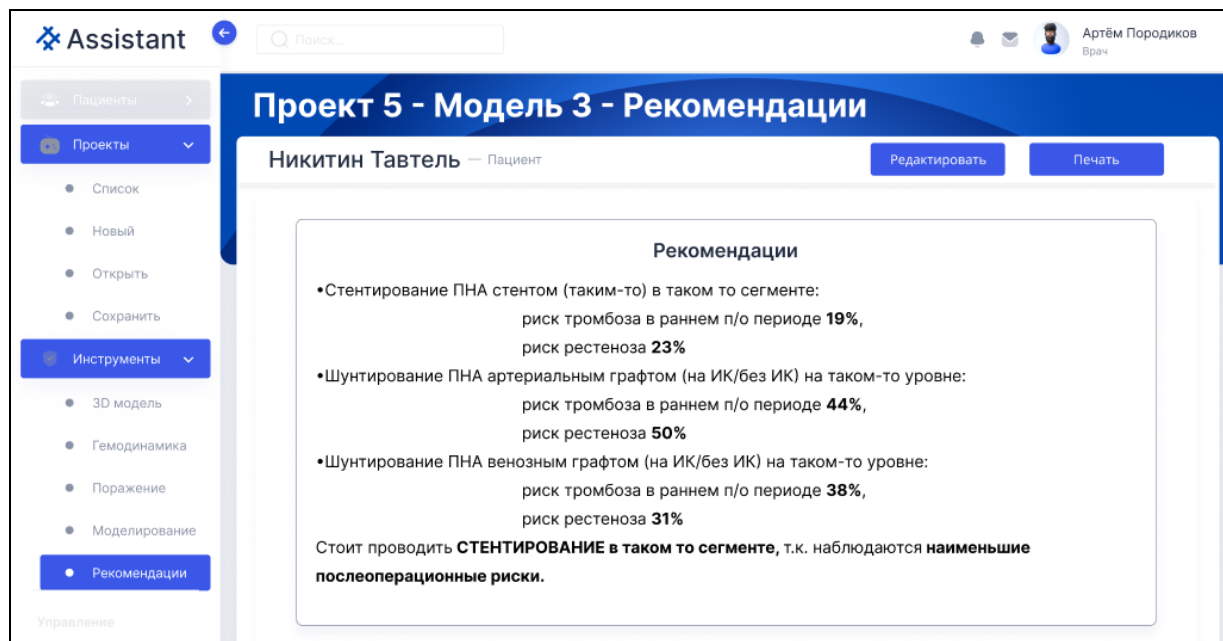


Рис. 12. Страница рекомендаций

Заключение

Таким образом, в работе было рассмотрен вопрос проектирования информационной системы для воссоздания 3D-модели сердца по двухмерным изображениям коронарографии. Использование разрабатываемой информационной системы позволит существенно улучшить диагностику и планирование лечения сердечно-сосудистых заболеваний, что в итоге будет способствовать снижению смертности и повышению качества медицинской помощи.

Был проведен анализ существующих на данный момент информационных систем, использующихся в кардиохирургии, и были сформулированы функциональные требования к разрабатываемому приложению. В работе произведен выбор средств для проектирования, прототипирования и реализации информационной системы, таких как Visual Paradigm, Figma, PostgreSQL, Python, C++, C# и фреймворк Django. Также будет использоваться контейнеризация с помощью Docker, а взаимодействие с клиентом и распределение нагрузки будет осуществляться с помощью веб-сервера Nginx. Клиентская часть будет написана на фреймворке Flutter.

Спроектированная информационная система готова к следующему этапу – реализации.

Библиографический список

1. Госмед. Коронарография сосудов сердца. [Электронный ресурс]. URL: <https://www.gosmed.ru/lechebnaya-deyatelnost/metody-dagnostiki/rentgenoendovaskulyarnye-metody-dagnostiki/koronarografiya-sosudov-serdtsa/> (дата обращения: 13.05.2024).
2. World Heart Federation. World Heart Report 2023: Confronting the World's Number One Killer. Geneva, Switzerland, 2023. [Электронный ресурс]. URL: <https://heartreport23.world-heart-federation.org/> (дата обращения: 13.05.2024).
3. Всемирная организация здравоохранения. Сердечно-сосудистые заболевания (ССЗ) [Электронный ресурс]. URL: [https://www.who.int/ru/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/ru/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) (дата обращения: 13.05.2024).
4. Philips. Dynamic Coronary Roadmap Documentation [Электронный ресурс]. URL: <https://www.philips.ie/healthcare/product/HCDRCR01/dynamic-coronary-roadmap-see-clearly-guide-confidently#documents> (дата обращения: 14.05.2024).
5. Повидар. DICOM Viewer v3 [Электронный ресурс]. URL: <https://povidar.ru/dicom-viewer/v3/> (дата обращения: 14.05.2024).
6. CORONAROGRAPHY.AI. Non-invasive predictive AI coronary angiography [Электронный ресурс]. URL: <https://coronagraphy.ai/> (дата обращения: 14.05.2024).
7. Visual Paradigm [Электронный ресурс]. URL: <https://www.visual-paradigm.com/> (дата обращения: 22.05.2024).
8. Figma. The Collaborative Interface Design Tool [Электронный ресурс]. URL: <https://www.figma.com/> (дата обращения: 16.05.2024).
9. ТЮБЕ. Индекс ТЮБЕ [Электронный ресурс]. URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 17.05.2024).
10. Python.org. О Python™ [Электронный ресурс]. URL: <https://www.python.org/about/> (дата обращения: 16.05.2024).
11. Django Documentation. [Электронный ресурс]. URL: <https://docs.djangoproject.com/en/5.0/> (дата обращения: 20.05.2024).
12. Flutter [Электронный ресурс]. URL: <https://github.com/flutter/flutter> (дата обращения: 22.05.2024).
13. PostgreSQL Documentation [Электронный ресурс]. URL: <https://www.postgresql.org/docs/> (дата обращения: 19.05.2024).
14. Docker Documentation [Электронный ресурс]. URL: <https://docs.docker.com/get-started/overview/> (дата обращения: 22.05.2024).
15. NGINX [Электронный ресурс]. URL: <https://nginx.org/ru/> (дата обращения: 22.05.2024).

References

1. Gosmed. Koronarografiya sosudov serdca [Electronic resource]. URL: <https://www.gosmed.ru/lechebnaya-deyatelnost/metody-dagnostiki/rentgenoendovaskulyarnye-metody-dagnostiki/koronarografiya-sosudov-serdtsa/> (date of access: 05/13/2024). (In Russ.)
2. World Heart Federation. World Heart Report 2023: Confronting the World's Number One Killer. Geneva, Switzerland, 2023. [Electronic resource]. URL: <https://heartreport23.world-heart-federation.org/> (date of access: 05/13/2024).
3. World Health Organization. Cardiovascular diseases (CVD) [Electronic resource]. URL: [https://www.who.int/ru/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/ru/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) (date of access: 05/13/2024).
4. Philips. Dynamic Coronary Roadmap Documentation [Electronic resource]. URL: <https://www.philips.ie/healthcare/product/HCDRCR01/dynamic-coronary-roadmap-see-clearly-guide-confidently#documents> (date of access: 05/14/2024).
5. Povidar. DICOM Viewer v3 [Electronic resource]. URL: <https://povidar.ru/dicom-viewer/v3/> (date of access: 05/14/2024).
6. CORONAROGRAPHY.AI. Non-invasive predictive AI coronary angiography [Electronic resource]. URL: <https://coronarography.ai/> (date of access: 05/14/2024).
7. Visual Paradigm [Electronic resource]. URL: <https://www.visual-paradigm.com/> (date of access: 05/22/2024).
8. Figma. The Collaborative Interface Design Tool [Electronic resource]. URL: <https://www.figma.com/> (date of access: 05/16/2024).
9. TIOBE. TIOBE index [Electronic resource]. URL: <https://www.tiobe.com/tiobe-index/> (date of access: 05/17/2024).
10. Python.org. About Python™ [Electronic resource]. URL: <https://www.python.org/about/> (date of access: 05/16/2024).
11. Django Documentation [Electronic resource]. URL: <https://docs.djangoproject.com/en/5.0/> (date of access: 05/20/2024).
12. Flutter [Electronic resource]. URL: <https://github.com/flutter/flutter> (date of access: 05/22/2024).
13. PostgreSQL Documentation [Electronic resource]. URL: <https://www.postgresql.org/docs/> (date of access: 05/19/2025).
14. Docker Documentation [Electronic resource]. URL: <https://docs.docker.com/get-started/overview/> (date of access: 05/22/2024).
15. NGINX [Electronic resource]. URL: <https://nginx.org/ru/> (date of access: 05/22/2024).

**DESIGN OF THE INFORMATION SYSTEM
“RESTORATION OF A THREE-DIMENSIONAL MODEL
OF THE HEART VESSELS FROM 2D CORONARY IMAGES”**

Osipov Alexander D.

Perm State University, 614068, Russia, Perm, st. Bukireva, 15, sashaosipovden2003@gmail.com

Barulina Marina A.

Perm State University, 614068, Russia, Perm, st. Bukireva, 15, mab@psu.ru

The design of an information system for recreating a 3D model of the heart from two-dimensional coronary angiography images is considered. A study of existing information systems with a similar functionality is being conducted. Based on the analysis of these systems, the basic functional requirements for the developed desktop application are formulated. A reasonable choice of tools is made for the design and implementation of the information system, as well as tools for prototyping the user interface. To design the system, a CASE software tool is used that supports UML modeling – Visual Paradigm. Figma is used to create a prototype of the interface. To store data, PostgreSQL acts as a DBMS. To implement the server part of the system, the programming languages Python, C++, C# and the Django framework will be used; containerization of each part of the server will also be used, and interaction with the client and load distribution will be carried out using the Nginx web server. The client part will be written in the Flutter framework.

Keywords: three-dimensional model, heart vessels, coronary angiography, information system, design.

ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ УЧЕТА ТОВАРООБОРОТА МАГАЗИНА

Парамонов Арсений Сергеевич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, paramosha228@gmail.com

Кушев Вадим Олегович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, vadimk@psu.ru

В статье рассматриваются основные теоретические положения для проектирования мобильного приложения для учета товарооборота в магазине «Сюрприз». Была проведена беседа с заказчиком, в ходе которой был выявлен список проблем, связанных с процессом учета товаров. Был сформирован список функциональных требований к мобильному приложению. Проведен анализ предметной области, анализ существующих решений и уточнен список требований к приложению. Был проведен анализ и выбор средств проектирования. Была спроектирована база данных, построена ER-диаграмма и физическая модель данных. Также были построены диаграммы классов и прецедентов для проектирования функционала системы и взаимодействия пользователей с системой учета товарооборота. В статье представлено проектирование пользовательского интерфейса мобильного приложения. В перспективе рассматривается полноценная разработка мобильного приложения для учета товарооборота магазина.

Ключевые слова: товарооборот, мобильное приложение, проектирование.

В современном мире торговли процесс учета товарооборота является неотъемлемой частью ведения успешного бизнеса. Однако, многие торговые предприятия малого бизнеса не могут позволить себе дорогостоящее ПО для учета товаров. Вместо этого многие используют устаревшие методы, такие как книги учета. Некоторые торговые предприятия обращаются к разработчикам с просьбой создать более доступную систему, которая позволила бы вести учета товаров без значительных финансовых затрат. Данная работа как раз является заказом от одного из таких предприятий.

Целью работы является разработка проектного решения мобильного приложения, которое позволит вести учет товарооборота, добавлять новые товары и отслеживать остатки на складе.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ предметной области.
2. Провести анализ существующих решений.

3. Провести обзор, анализ и выбор инструментального ПО.

4. Провести проектирование системы.

Магазин «Сюрприз» является торговым предприятием малого бизнеса и специализируется на продаже одежды для работы и отдыха.

В результате собеседования с ИП Золотаревой Т. М. был выявлен ряд проблем, связанные с процессом учета товарооборота в магазине «Сюрприз». В настоящее время магазин «Сюрприз» не может позволить себе приобрести дорогостоящее ПО для ведения учета товаров. Вместо этого, в магазине используются книги учета, которые негативно сказываются на ведении учета товаров. Ручная запись данных о товаре, закупках и продажах требует много времени, из-за чего заказчику часто не успевает заполнить всю информацию, и приходится брать книги учета домой после рабочего дня. Хранение большого количества книг учета требует много пространства, которое может быть использовано для других целей. Таким образом происходит процесс учета товаров в магазине «Сюрприз».

Исходя из проблем можно выявить требования для проектируемой системы. Мобильное приложение должно быть бесплатным, так как заказчик не может позволить себе дорогостоящее ПО для учета товаров. Приложение должно быть мобильным или переносимым для того, чтобы заказчик мог использовать приложение как на работе, так и дома. В проектируемом приложении должны быть функции добавления закупленного товара, уменьшения количества товара при его продаже, подсчета выручки, а также отслеживания текущего количества товара в магазине. Приложение должно работать локально на устройстве, так как заказчику важно использовать приложение без подключения к интернету.

Для более точного проектирования проведем анализ существующих решений в данной области.

Исходя из потребностей заказчика и проблем, связанных с учетом товара в магазине «Сюрприз» были определены требования, которые необходимо реализовать в разрабатываемом приложении.

Таковыми функциональными требованиями являются:

- добавление товара;
- удаление проданного товара;
- отслеживание количества товара в магазине;
- подсчет выручки;
- бесплатное приложение;
- локальное приложение;
- переносимое или мобильное приложение для системы Android;
- русскоязычное приложение.

Для того, чтобы понять, существуют ли решения, которые способны реализовать обозначенный функционал, и приемлемы для малого торгового предприятия с точки зрения стоимости, необходимо провести анализ существующих решений.

Одним из решений является «ForPOSt Супермаркет». Это программное обеспечение, которое позволяет автоматизировать процесс продажи товаров в торговом предприятии. Для магазина «Сюрприз» представленное решение может предложить возможности добавления и удаления товара, отслеживания количества товара в магазине и подсчет выручки. Однако, эта система требует больших финансовых затрат на обслуживание. Также, эта система не может функционировать локально, так как обязана иметь подключение к оператору фискальных данных. Также не предусмотрена возможность работы на мобильном телефоне. Не все требования возможно реализовать при помощи данного решения. Следовательно «ForPOSt Супермаркет» не подходит для магазина «Сюрприз».

Oracle ERP Cloud – это комплексное программное решение, которое объединяет в себе все бизнес процессы компании, включая учет продаж и выручки [1]. Данная система могла бы помочь не только в учете товаров, но и в подсчете выручки, а также и в других аспектах бизнеса. Однако, продукт «Oracle ERP Cloud» требует значительных ежемесячных денежных вложений, что невозможно для магазина «Сюрприз». Помимо этого, в данном решении отсутствует русская локализация, что может негативно сказаться на эффективности работы на торговом предприятии. Следовательно, решение Oracle ERP Cloud не подходит для магазина «Сюрприз».

Мобильное приложение «Учет товаров – простой склад». Это приложение, которое можно установить на смартфон или планшет и использовать для учета товаров. В данном мобильном приложении возможно реализовать все функциональные требования, начиная с добавления купленного товара, и, заканчивая подсчетом выручки. Помимо этого, данное приложение может полностью функционировать без подключения к интернету. Но данное приложение в своем бесплатном виде имеет ограниченный функционал, а для открытия полного доступа требует значительных денежных затрат, что непозволительно для малого торгового предприятия «Сюрприз». Отсюда можно сделать вывод, что мобильное приложение «Учет товаров – простой склад» не подойдет для магазина «Сюрприз», потому что для полноценного функционирования требует денежных вложений.

Исходя из особенностей каждого решения, можно уточнить функциональные требования к проектируемой системе. При реализации сформулированного функционала можно опираться на функционал мобильного приложения «Учет товаров – простой склад», потому что оно удовлетворяет большему количеству функциональных требований. Но, в отличие от него, проектируемое мобильное приложение для учета товарооборота будет бесплатным.

Также отсутствуют функции, которые можно добавить в список ранее сформулированных функциональных требований.

Теперь, определившись с требованиями к системе, можно осуществлять проектирование этой системы.

На этом этапе необходимо определить основные компоненты системы учета товарооборота магазина, их взаимосвязь и функциональность. Необходимо рассмотреть подходящую структуру для мобильного приложения. Также требуется разработать структуру базы данных, которая будет хранить необходимую информацию о товарах, поставках, заказах и других аспектах товарооборота магазина. Так же предполагается проектирование интерфейса.

Так как одним из требований заказчика является возможность ведения учета товарооборота при помощи мобильного телефона, было принято решение для проектирования системы применить архитектуру мобильного приложения.

Так как для предприятия важно, чтобы приложением можно было пользоваться в любом месте без подключения к интернету, было принято решение, что мобильное приложение должно быть нативного типа, то есть запускаться локально на устройстве. Проектироваться и разрабатываться данное нативное приложение должно конкретно для системы Android 10. Благодаря этому приложение будет запускаться в автономном режиме.

Мобильное приложение будет основано на трехслойной архитектуре. Первый слой будет представлять данные для пользователя. При создании первого слоя для приложения особое внимание будет уделяться дизайну пользовательского интерфейса (UI) и пользовательского опыта (UX). Интерфейс приложения будет полностью на русском языке. Благодаря этому приложение будет иметь комфортный для заказчика внешний вид и удобную навигацию. На втором слое располагается вся бизнес-логика системы. На данном слое будут производиться операция добавления купленного и удаления проданного товара, операция подсчета выручки за определенный период времени. На третьем слое будет располагаться данные для системы и база данных. В базе данных будет храниться информация о товаре, продажах товаров и закупках.

Для проектирования интерфейса приложения было решено использовать Figma, так этот инструмент бесплатный, предоставляет возможность работы в браузере и предлагает возможность совместной работы.

Для построения физической модели данных необходимо определиться с типом базы данных, выбрать СУБД и средство проектирования базы данных.

Чтобы хранить данные о товарах понадобится табличное представление данных, поэтому отличным выбором будет реляционный тип базы данных. Отсюда следует, что выбирать СУБД нужно из списка реляционных СУБД.

Существует множество различных реляционных СУБД, таких как PostgreSQL, MySQL, SQLite, Firebird, Microsoft SQL Server и другие. Для локального мобильного приложения для учета товарооборота важно, чтобы СУБД была ориентированная на кроссплатформенность, то есть могла использоваться для мобильных приложений. Так же важно, чтобы база данных не занимала большого пространства в памяти мобильного устройства и все данные находились локально на устройстве. Помимо этого, СУБД должна позволять работать на языке программирования Java, так как он является самым популярным языком программирования для мобильных устройств. Все перечисленные особенности заключены в СУБД SQLite.

Для построения концептуальной модели базы данных было решено использовать «DBDesigner», так как это приложение позволяет проектировать базу данных в терминах СУБД SQLite и позволяет экспортировать модель в СУБД.

Перед построением диаграмм для разрабатываемой системы, необходимо определиться с нотацией для моделирования системы. Нотация должна позволять проектировать программное обеспечение, объектов системы и процессов. Нотация должна быть понятна для программиста-разработчика, диаграммы построенные в данной нотации возможно реализовать на языке программирования. Для этих задач подойдет нотация UML.

Чтобы построить диаграммы для системы учета товарооборота, необходимо выбрать подходящую программу для моделирования. В этой программе должна быть возможность построения различных диаграмм в нотации UML. Для этих целей будет использоваться программа Draw.io, так как она одна из самых популярных программ для моделирования.

После того, как были выбраны все средства проектирования, можно приступить к процессу проектирования системы для учета товарооборота магазина одежды.

Для того, чтобы спроектировать базу данных для данной системы, необходимо построить для нее ER-диаграмму. Для построения ER-диаграммы необходимо сформировать все данные из предметной области в одну таблицу, а затем нормализовать эту таблицу с данными [2]. Также, предполагается построение физической модели данных для системы учета товарооборота.

Для начала необходимо определить все атомарные атрибуты в данной предметной области [3]. Ранее, были определены данные для реализации обозначенного функционала. Понадобится фиксировать факты продажи товара, факты закупки товара в магазин, факты размещения на продажу и информацию о товаре.

Была сформирована структура базы данных в нулевой нормальной форме и нормализована до 3 нормальной формы. В результате получилось 4 таблицы. Затем была построена ER-диаграмма, представленная на рис. 1.

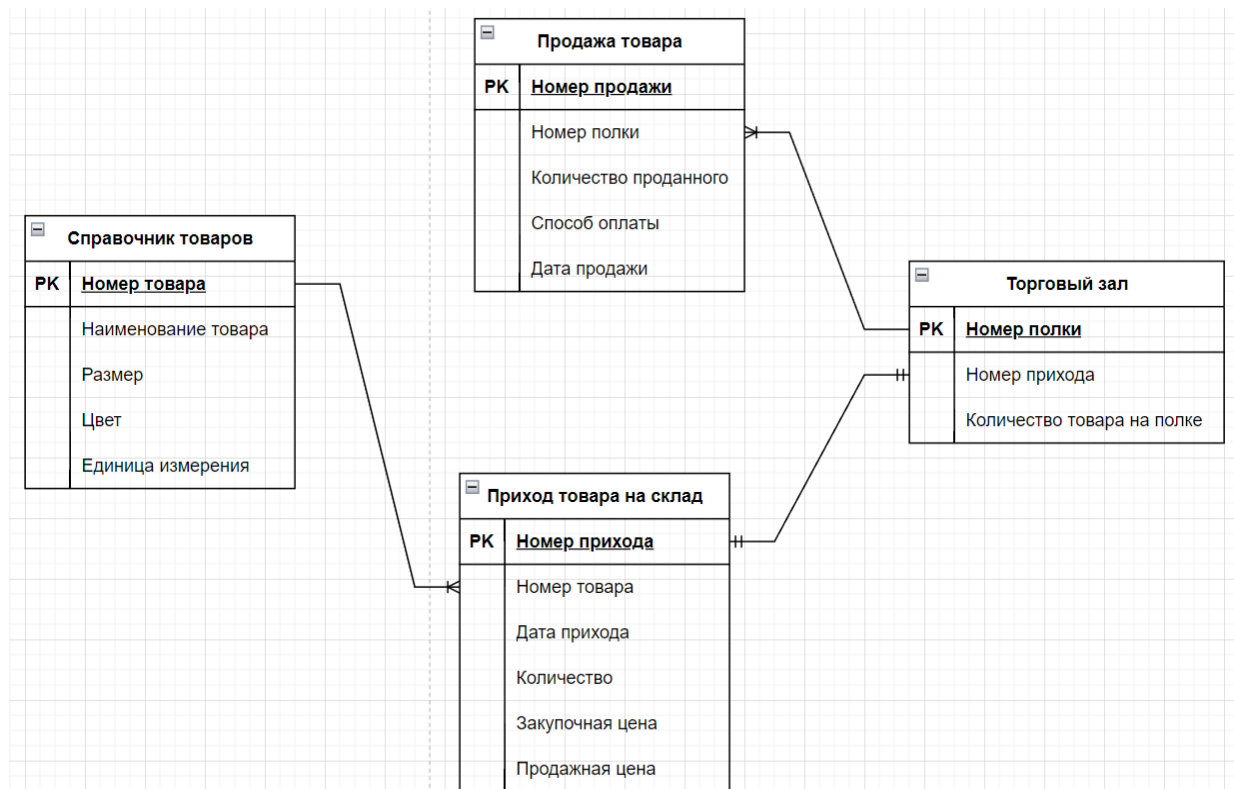


Рис. 1. ER-диаграмма

Далее была построена физическая модель данных. Использовался сервис «DBDesigner», модель была построена в терминах СУБД SQLite. Физическая модель данных изображена на рис. 2.

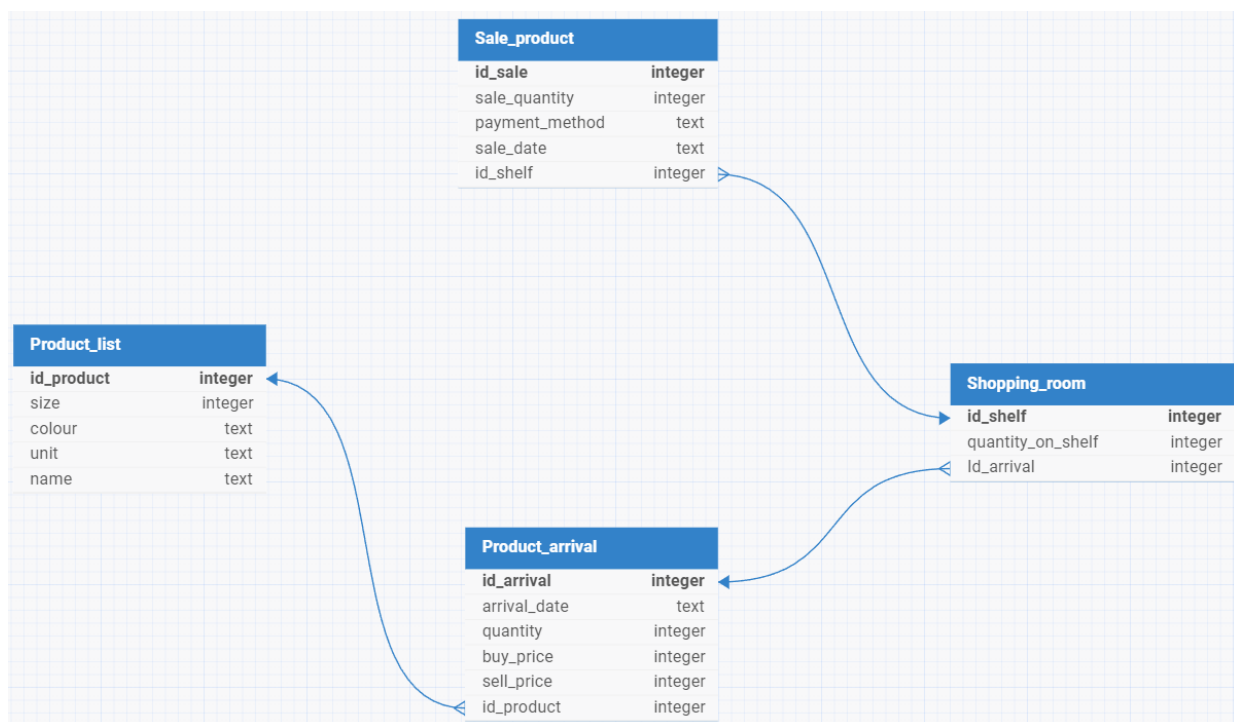


Рис. 2. Физическая модель данных

Для моделирования взаимодействия пользователя с системой была построена диаграмма прецедентов. С системой взаимодействует только сотрудник магазина. Сама диаграмма прецедентов изображена на рис. 3.



Рис. 3. Диаграмма прецедентов

Для моделирования функционала системы была построена диаграмма классов [4]. Для реализации обозначенного функционала понадобятся классы приход товара, продажа товара, список товаров и торговый зал. Сама диаграмма изображена на рис. 4.

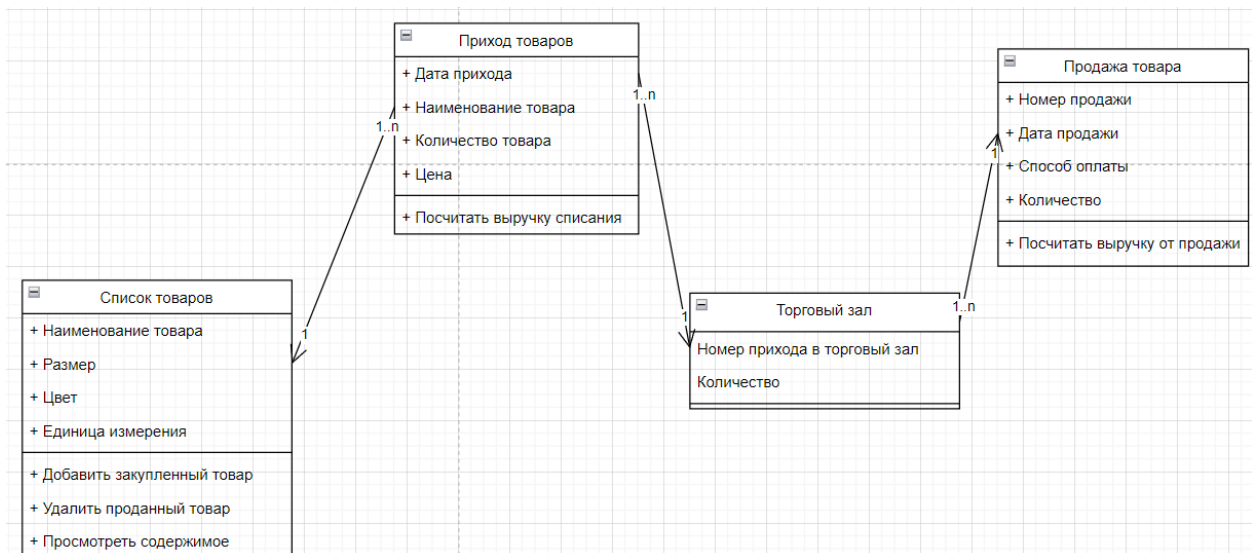


Рис. 4. Диаграмма классов

После того как была спроектирована база данных, взаимодействие пользователя с системой и функционал системы, можно приступать к проектированию интерфейса приложения.

В результате проектирования были созданы макеты главного экрана, экрана прихода товара, экрана добавления нового прихода, изображенные на рис. 5.



Рис. 5. Макеты главного экрана, прихода и добавления нового прихода

А также были созданы макеты окна продажи товара, добавления новой продажи и окна со списком всех товаров, изображенные на рис. 6.

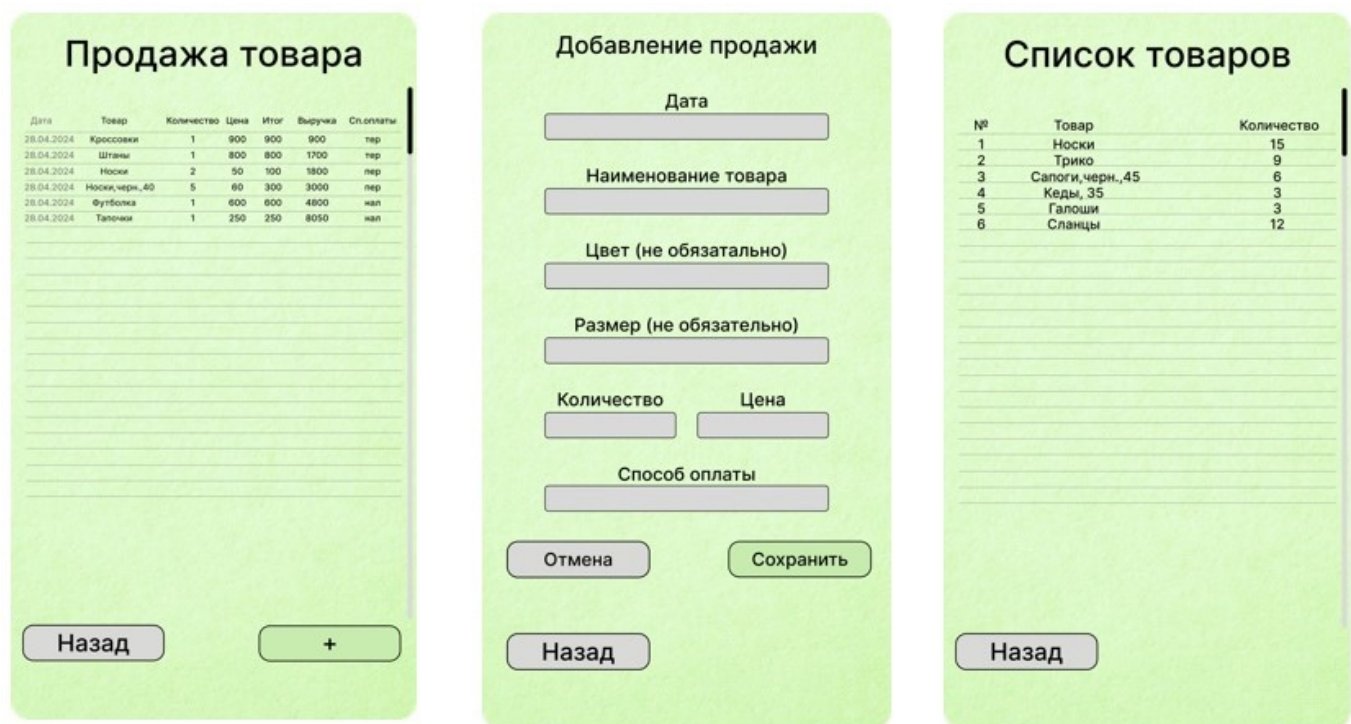


Рис. 6. Макеты окна продажи, добавления новой продажи и список всех товаров

В ходе выполнения работы были проведены анализ предметной области, обзор существующих решений и сформулированы функциональные требования к системе.

Было проведено проектирование системы для учета товарооборота магазина одежды. Была построена диаграмма классов, чтобы определить необходимые данные для реализации сформулированных функциональных требований. Была нормализована база данных, построена ER-диаграмма и физическая модель данных в терминах выбранной СУБД. Для моделирования взаимодействия пользователя и системы, была построена диаграмма прецедентов.

Был проведен процесс проектирования пользовательского интерфейса мобильного приложения для учета товарооборота, путем описания требований к каждой форме интерфейса, что позволит реализовать описанный интерфейс в будущем.

Библиографический список

1. Oracle Enterprise Resource Planning (ERP) [Электронный ресурс]. URL: <https://www.oracle.com/erp/> (дата обращения 11.01.2024).
2. Советов Б. Я. Базы данных: учебник для прикладного бакалавриата / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовский. 3-е изд., перераб. и доп. М.: Издательство Юрайт, 2019. 420 с. URL: <https://www.urait.ru/bcode/431947> (дата обращения: 03.03.2024).
3. Кузнецов С. Д. Введение в реляционные базы данных: учеб. пособие. 3-е изд. М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. 247 с. URL: <https://www.iprbookshop.ru/102002.html> (дата обращения: 15.03.2024).
4. UML Class Diagram Tutorial [Электронный ресурс]. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> (дата обращения: 02.03.2024).

References

1. Oracle Enterprise Resource Planning (ERP) [Electronic resource]. URL: <https://www.oracle.com/erp/> (date of access: 01/11/2024).
2. Sovetov B. Y. Bazy dannyh: uchebnik dlya prikladnogo bakalavriata / B. Y. Sovetov, V. V. Cehanovskiy, V. D. Chertovskiy. 3rd ed., revised and expanded. M.: Publishing Urait, 2019. 420 p. URL: <https://www.urait.ru/bcode/431947> (date of access: 03/03/2024). (In Russ.)
3. Kuznetsov S. D. Vvedenie v relyacionnie bazy dannyh: uchebnoe posobie. 3rd ed. M.: Internet University of Information Technologies (INTUIT), IPR Media, 2021. 247 p. URL: <https://www.iprbookshop.ru/102002.html> (date of access: 03/15/2024). (In Russ.)
4. UML Class Diagram Tutorial [Electronic resource]. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> (date of access: 03/02/2024).

DESIGNING A MOBILE APPLICATION FOR ACCOUNTING STORE TRADE TURNOVER

Paramonov Arseniy S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, paramosha228@gmail.com

Kushev Vadim O.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, vadimk@psu.ru

The article discusses the basic theoretical principles for designing a mobile application for recording turnover in the Surprise store. A conversation was held with the customer, during which a list of problems related to the process of accounting for goods was identified. A list of functional requirements for the mobile application was generated. The subject area was analyzed, existing solutions were analyzed and the list of requirements for the application was clarified. An analysis and selection of design tools was carried out. A database was designed, an ER diagram and a physical data model were built. Class and use case diagrams were also constructed to design the functionality of the system and user interaction with the turnover accounting system. The article presents the design of the user interface of a mobile application. In the future, a full-fledged development of a mobile application for recording store turnover is being considered.

Keywords: trade turnover, mobile app, designing.

УПРАВЛЕНИЕ НАБОРАМИ ДАННЫХ В NO-CODE СИСТЕМЕ АНАЛИЗА ДАННЫХ

Русаков Вадим Вячеславович, Мокунева Валерия Евгеньевна, Селетков Илья Павлович

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, iseletkov@gmail.com

В данной работе продолжен обзор построения no-code системы анализа данных. Подробнее рассматриваются функции загрузки, хранения, визуализации и предобработки пользовательских табличных наборов данных. Для управления списком наборов, предобработки данных предложены отдельные микросервисы на FastAPI, программный интерфейс которых рассмотрен в этой работе. Для отображения табличных данных в web-интерфейсе выбрана библиотека Handsontable, которая позволяет осуществлять визуализацию и предобработку данных в удобном и привычном пользователю виде.

Ключевые слова: обработка данных, аналитические модели, машинное обучение, клиент-серверная система, web-интерфейс, микросервис, pandas, scikit-learn, joblib, handsontable.

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, не требующей навыков программирования, является эффективным способом решения этой проблемы.

Обзор архитектуры системы

Архитектура системы, серверной части, клиентской части, используемые способы клиент-серверного взаимодействия рассмотрены в работе [1]. Общая архитектура приведена на рис. 1.

Сервис управления наборами данных

Сервис управления наборами данных является ключевым компонентом в системе. Его основная задача – преобразование загружаемых пользовательских файлов табличных форматов в стандартизированные наборы данных и их последующая запись в файловое хранилище. Помимо этого, сервис предоставляет функциональность для получения информации о поль-

зовательских наборах данных и отображения этой информации по запросу пользователя на клиентском интерфейсе.

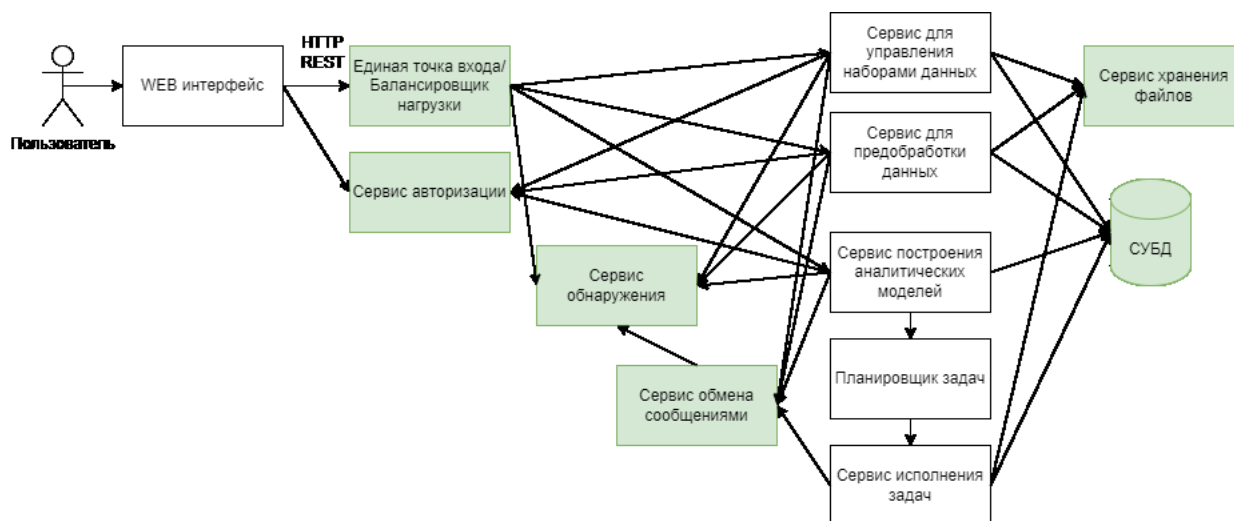


Рис. 1. Структура по-code системы анализа данных

Методы, представленные в данном сервисе (рис. 2):

upload: загрузка файла в различных табличных форматах (например, .csv, .xlsx), которые затем конвертируются в единый системный формат и сохраняются в файловое хранилище;

get_one: получение конкретного набора данных из файлового хранилища по запросу пользователя;

get_all: получение списка всех пользовательских наборов данных, это удобно для получения обзора всех доступных данных, загруженных пользователем;

delete: удаление набора данных;

update: переименование набора данных;

download: скачивание из системы набора данных с выбранным пользователем табличным форматом (например, .csv, .xlsx) на клиентский компьютер для дальнейшего использования.

datasets		
GET	/datasets/info	About
POST	/datasets/upload	Upload
GET	/datasets/get_all	Get All
GET	/datasets/get_one	Get One
DELETE	/datasets/delete	Delete
PUT	/datasets/update	Update
GET	/datasets/download	Download

Рис. 2. Методы микросервиса управления наборами данных

Представленные выше методы обеспечивают полный цикл управления данными — от их загрузки и хранения до получения, обновления и удаления, что делает данный сервис важнейшим инструментом для работы с наборами данных в системе.

Основная библиотека, используемая в этом сервисе, — это `joblib` [2]. Эта библиотека играет важную роль в процессе преобразовании данных в текстовый формат, что позволяет эффективно сохранять и восстанавливать наборы данных в Python. Когда пользователь загружает файлы в различных табличных форматах (например, `.csv`, `.xlsx`), сервис преобразует эти файлы в единый системный формат `dataframe` из библиотеки `Pandas` [3]. Затем данные преобразуются в текстовый вид с помощью библиотеки `joblib` и сохраняются в файловое хранилище. Это позволяет сохранить структуру данных и их состояние.

Сервис предобработки данных

Основная задача данного сервиса — подготовка наборов данных к последующему их использованию для построения аналитических моделей. Помимо этого, сервис предоставляет функциональность для получения информации о результатах обработки и отображения этой информации по запросу пользователя на клиентском интерфейсе.

Некоторые методы, представленные в данном сервисе (рис. 3):

- **`check_null`**: позволяет выявить и посчитать количество пропущенных значений в каждой колонке;
- **`delete_rows_with_nan`**: удаляет строки с пропущенными значениями (`NaN`), это позволяет пользователю получить чистый набор данных, готовый для дальнейшего анализа или обработки;
- **`change_rows_with_nan`**: заменяет пропущенные значения (`NaN`) на заданное значение или метод (например, среднее значение, медиану), это удобно для подготовки данных к анализу, обеспечивая целостность и полноту данных;
- **`create_column_scaler`**: создает и применяет масштабирование к указанной колонке данных. Всего доступно 4 вида скейлеров;
- **`change_column_encode`**: преобразует категориальные данные в числовые значения с помощью кодирования, это необходимо для подготовки данных к машинному обучению и другим видам анализа;
- **`one_hot_encode`**: преобразует категориальные значения в бинарные столбцы, это позволяет эффективно использовать категориальные данные в моделях машинного обучения.

Представленные выше методы в полной мере обеспечивают цикл обработки данных — от добавления новых значений и преобразования до проверки, очистки и кодирования данных.

preparation		
GET	/preparation/info	About
GET	/preparation/check_null	Check Null
POST	/preparation/change_cell	Change Cell
POST	/preparation/delete_rows_with_NAN	Delete Rows With Nan
POST	/preparation/change_rows_with_NAN	Change Rows With Nan
POST	/preparation/delete_row	Delete Row
POST	/preparation/delete_column	Delete Column
POST	/preparation/create_column_scaler	Create Column Scaler
POST	/preparation/change_column_encode	Change Column Encode
POST	/preparation/change_column_OneHotEncode	Change Column Onehotencode

Рис. 3. Методы микросервиса предобработки данных

Основная библиотека, используемая в этом сервисе – это Pandas [3]. Она играет важную роль в процессе предобработки данных, предоставляя мощные и гибкие инструменты для работы с табличными данными. Благодаря широкому набору функций для обработки данных Pandas становится незаменимым инструментом для обеспечения высокого качества и готовности данных для дальнейшего анализа.

Отображение табличных данных в web-интерфейсе

В работе [1] описан процесс дизайна и построения макетов страниц web-интерфейса системы с использованием инструментов Vue.js и vuetify. Рассмотрим детальнее отображение табличных данных и её реализацию.

Одним из инструментов, который позволяет встроить в web-интерфейс мощные и гибкие таблицы с возможностью редактирования данных, является библиотека Handsontable [4]. Установка библиотеки в проект Vue осуществляется вызовом команды `yarn install handsontable @handsontable/vue3`, под текущую версию Vue. Подключается библиотека в сам код при помощи команды `import { HotTable } from '@handsontable/vue3'`.

Вызов метода происходит следующим образом:

1. Создание ссылки на компонент HotTable: в шаблоне (template) добавить атрибут `ref` к компоненту HotTable. Это позволит получить доступ к экземпляру Handsontable через `this.$refs.hotTable.hotInstance`.
2. Добавление настроек для Handsontable: в объекте компонента добавить свойство `hotSettings`. Это свойство будет содержать начальные настройки для таблицы Handsontable, такие как данные, заголовки строк и столбцов, контекстное меню, фильтры и выпадающее меню.

3. Обработка загрузки файла: в методе `uploadFile`, который вызывается при загрузке файла, создайте объект `FormData` и добавьте в него загружаемый файл.

4. Загрузка данных в таблицу Handsontable: создать метод `loadDataToTable`, который будет принимать данные и использовать экземпляр Handsontable для обновления данных в таблице (рис. 4). Для этого нужно получить экземпляр Handsontable через `this.$refs.hotTable.hotInstance` и вызовите метод `loadData`, передав полученные данные.

Таким образом, при загрузке файла данные будут отправлены на сервер, а затем загружены в таблицу Handsontable с помощью метода `loadData`. Это позволяет интегрировать функции загрузки данных в таблицу Handsontable в Vue-компоненте.

```
73     },  
74     loadDataToTable(data) {  
75         const hotInstance = this.$refs.hotTable.hotInstance;  
76         hotInstance.loadData(data);  
77     },  
78 }
```

Рис. 4. Вызов метода обновления и загрузки данных

Отображение набора данных в виде таблицы вынесено на отдельную страницу web-интерфейса. Данная страница предназначена для эффективной работы с данными, предоставляя пользователю удобные инструменты для их управления и анализа. Страница состоит из двух основных частей: панели инструментов, расположенной слева, и таблицы данных, напоминающей Excel, расположенной справа.

Левая панель страницы содержит набор инструментов, которые облегчают работу с данными в таблице. Основные функциональные элементы панели инструментов включают:

1) **добавление и удаление строки**. Кнопка позволяет пользователю как добавить новую строку в таблицу данных, так и удалить ее. При нажатии на кнопку в таблицу добавляется новая строка, готовая для ввода данных. Это позволяет пользователю быстро расширять таблицу по мере необходимости. Пользователь выбирает строку, которую хочет удалить, и нажимает на кнопку. После подтверждения удаления выбранная строка удаляется из таблицы (рис. 5);

2) **сортировка данных**. Кнопка позволяет сортировать данные в таблице по выбранному столбцу. Пользователь выбирает столбец для сортировки и направление сортировки (по возрастанию или убыванию). Таблица обновляется, отображая данные в указанном порядке;

3) **экспорт данных**. Кнопка «Сохранить» предоставляет возможность экспорта данных из таблицы в различные форматы (например, CSV, Excel). При нажатии на кнопку от-

крывается меню с вариантами экспорта. Пользователь выбирает нужный формат, и данные из таблицы сохраняются в выбранном формате на локальное устройство пользователя.

Инструменты

ADD COLUMN

DELETE COLUMN

	A	B	C	D	E
1		Ford	Volvo	Toyota	Honda
2	2017	20	11	14	13
3	2018	30	15	12	13
4	2019	20	11	14	13
5	2020	30	15	12	13

Рис. 5. Таблица в полном виде

Правая панель страницы содержит таблицу данных, которая напоминает интерфейс Excel и предоставляет следующие возможности:

1) **редактирование ячеек.** Описание: Пользователь может редактировать данные в ячейках таблицы, вводя и изменяя значения. При двойном щелчке на ячейку она переходит в режим редактирования, позволяя пользователю ввести или изменить данные (рис. 6). После завершения редактирования данные сохраняются автоматически;

	A	B	C	D
1		Ford	Volvo	Toyota
2	2017	20	11	14
3	2018	30	15	12
4	2019	20	11	14
5	2020	30	15	12

Рис. 6. Редактирование ячеек таблицы

2) **вставка и удаление строк и столбцов.** Пользователь может добавлять или удалять строки и столбцы в таблице. В панели инструментов (рис. 7) доступны кнопки для вставки и удаления строк и столбцов. Это позволяет пользователю гибко управлять структурой таблицы в зависимости от потребностей.

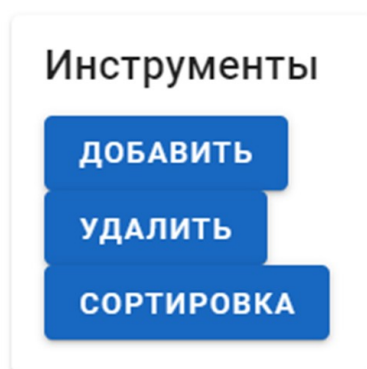


Рис. 7. Панель инструментов

Таким образом функционал данной web-страницы разработан для обеспечения максимального удобства и эффективности при работе с данными. Левая панель с инструментами предоставляет пользователю быстрый доступ к основным функциям управления данными, таким как добавление, удаление, сортировка и экспорт. Правая панель с таблицей данных, напоминающей интерфейс ПО MS Excel, обеспечивает интуитивно понятное и гибкое рабочее пространство для редактирования и анализа данных. Вместе эти компоненты создают мощный инструмент, который может быть полезен для различных задач, связанных с обработкой данных, таких как бухгалтерский учет, управление проектами, анализ данных и многое другое.

Подводя итог, можно сказать, что разработанная web-страница с панелью инструментов и таблицей данных представляет собой мощный и удобный инструмент для работы с данными. Интуитивно понятный интерфейс, гибкость и адаптивность, поддержка формул и функций, безопасность данных и возможности интеграции делают эту web-страницу незаменимым помощником для различных задач, связанных с обработкой и анализом данных.

Библиографический список

1. Русаков В. В., Мокунева В. Е., Селетков И. П. Разработка No-Code системы обработки и хранения данных // Актуальные проблемы математики, механики и информатики 2023: сб. ст. по материалам студ. конф. Пермский государственный национальный исследовательский университет / под ред. М. М. Бузмаковой. Пермь, 2023. С. 312–317. [Электронный ресурс]. URL: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematikimekhaniki-informatiki-2023.pdf> (дата обращения: 12.07.2024).
2. Joblib [Электронный ресурс]. URL: <https://joblib.readthedocs.io/en/stable/> (дата обращения: 12.07.2024).
3. Pandas [Электронный ресурс]. URL: <https://pandas.pydata.org/docs/> (дата обращения: 12.07.2024).
4. Handsontable [Электронный ресурс]. URL: <https://handsontable.com/docs/react-data-grid/vue3-installation/> (дата обращения: 12.07.2024).

References

1. Rusakov V. V., Mokuneva V. E., Seletkov I. P. Developing of no-code system for storing and analyzing data: Proceedings of student conference // Actualnie problemy matematiki, mehanici, informatiki. Perm, 2023. Pp. 312–317. [Electronic resource]. URL: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematikimekhaniki-informatiki-2023.pdf> (date of access: 07/12/2024). (In Russ.)

2. Joblib [Electronic resource]. URL: <https://joblib.readthedocs.io/en/stable/> (date of access: 07/12/2024).
3. Pandas [Electronic resource]. URL: <https://pandas.pydata.org/docs/> (date of access: 07/12/2024).
4. Handsontable [Electronic resource]. URL: <https://handsontable.com/docs/react-data-grid/vue3-installation/> (date of access: 07/12/2024).

MANAGING DATASETS IN NO-CODE DATA PROCESSING AND ANALYSIS SYSTEM

Rusakov Vadim V., Mokuneva Valeria E., Seletkov Ilya P.

Perm State University, 15 Bukireva St., 614068, Perm, iseletkov@gmail.com

This paper continues the review of the construction of a no-code data analysis system. The functions of loading, storing, visualizing and preprocessing custom tabular datasets are discussed in more detail. To manage the list of datasets and data preprocessing separate microservices on the FastAPI are proposed. This work considers the software interface of API of this microservices. To display tabular data in the web interface the Handsontable library is selected. It allows visualization and preprocessing of data in a convenient and familiar way to the user.

Keywords: data processing, analytical model, machine learning, frontend, backend, microservices, pandas, joblib, handsontable.

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ УПРАВЛЕНИЯ БИЗНЕСОМ В СФЕРЕ ОБЩЕСТВЕННОГО ПИТАНИЯ

*Сычев Иван Андреевич, Соколова Ольга Леонидовна, Яшичев Дмитрий Львович,
Соколов Андрей Валерьевич*

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, info@interprogram.ru

Цель данной работы состоит в проектировании и разработке информационной системы предзаказа и доставки. В работе проведен анализ существующих процессов и заинтересованных сторон. По результатам анализа поставлены задачи на проектирование и составлены требования к разрабатываемой системе. В соответствии с предъявленными требованиями спроектирована и реализована информационная система. Результаты данной работы могут быть интересны организациям общественного питания, которые хотят расширить рынок сбыта, покупателям, которые ищут удобные способы заказа еды, а также курьерам.

Ключевые слова: информационная система, оптимизация процессов, предзаказ, доставка, общественное питание.

В условиях ускоренного темпа жизни потребителей возрос спрос на удобные и эффективные способы приобретения товаров, продуктов питания, включая готовые блюда. Создание веб-платформы для онлайн заказа продуктов питания и готовых блюд направлено на удовлетворение этого спроса, предоставляя потребителям удобный доступ к широкому ассортименту товаров с возможностью предзаказа и доставки.

Функциональные требования

В рамках анализа предметной области было выделено четыре группы пользователей системы: покупатели, продавцы, курьеры и сотрудники поддержки.

У актора покупателя следующие функциональные требования:

Личный кабинет: создание профиля, указание контактных данных, привязка и удаление банковской карты, назначение по умолчанию.

Создание заказа: сбор корзины, выбор времени и способа оплаты, выбор формы получения заказа, бронирование стола.

Выбор товаров: доступ к спискам заведений, поиск по товарам и заведениям, добавление товара в корзину или избранное, фильтрация товаров и заведений по набору критериев.

Коммуникация: комментарии сборщику заказа, курьеру, вопросы в чат с продавцом, вопросы в техническую поддержку, в том числе до создания заказа и добавления товаров в корзину.

У актора сотрудника заведения выделены следующие функциональные требования:

- кабинет управления: регистрация, принятие оферты, авторизация, загрузка товарного знака, присвоение ролей, информация о заведении и филиалах;
- управление заказами: просмотр списка заказов, подтверждение принятия заказа, отказ от заказа, управление статусами заказа;
- управление товарами: загрузка фотографий, текстовое описание, стоп-лист;
- коммуникация: список чатов, просмотр сообщений в чате, отправление сообщения с прикреплением файлов, возможность просмотра истории диалога с покупателем со стороны заведения в случае, если доступ имеют несколько пользователей. Возможность связаться с технической поддержкой.

Для актора курьера предъявляются следующие функциональные требования:

- личный кабинет: указание обязательных данных, номер телефона, ФИО, гражданство, дата рождения, статус налогоплательщика, возможные способы доставки, загрузка документов для идентификации личности и транспортного средства (если используется);
- выход на линию: выбор типа транспорта, типа доставки, фото контроль;
- доставка заказов: возможность принять заказ, навигация и определение местоположения курьера и целевых точек, управление статусами доставки;
- дополнительный функционал: планирование рабочих смен, выбор района, финансовый учет, бонусная система.

У актора сотрудника поддержки следующие функциональные требования:

- кабинет управления: возможность управления заказами, просмотр списка заказов, получение информации по конкретному заказу и управление статусами заказов. Для структурирования обращений пользователей сотруднику поддержки необходима возможность создавать, просматривать, редактировать, назначать исполнителя и менять статус заявок.

Нефункциональные требования

Для обеспечения высокого качества и эффективности системы, учтены следующие нефункциональные требования: платформенная совместимость для обеспечения доступности системы на различных устройствах, удобство использования для обеспечения положительного пользовательского опыта, безопасность для защиты данных пользователей и обеспечения стабильной работы системы, надежность для обеспечения непрерывной работы и предотвращения потери данных, производительность для быстрой и эффективной работы системы, удобство доработки для облегчения процесса технического сопровождения и обновления системы, расширяемость для обеспечения возможности масштабирования и добавления новых функциональных возможностей.

Моделирование бизнес-процессов

Моделирование бизнес-процессов служит важной основой для дальнейших этапов проектирования и разработки системы. Рассмотрим более подробно наиболее сложные процессы в системе.

Для оплаты заказа покупателю необходимо привязать банковскую карту. Процесс привязки банковской карты показан на UML диаграмме на рис. 1.

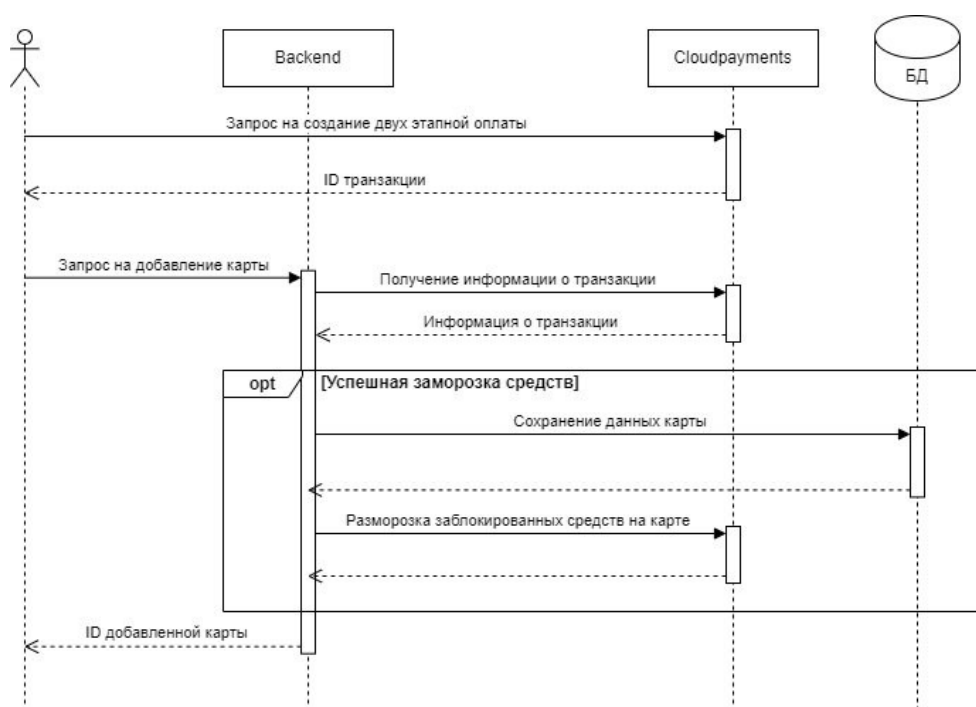


Рис. 1. Диаграмма последовательности процесса привязки карты

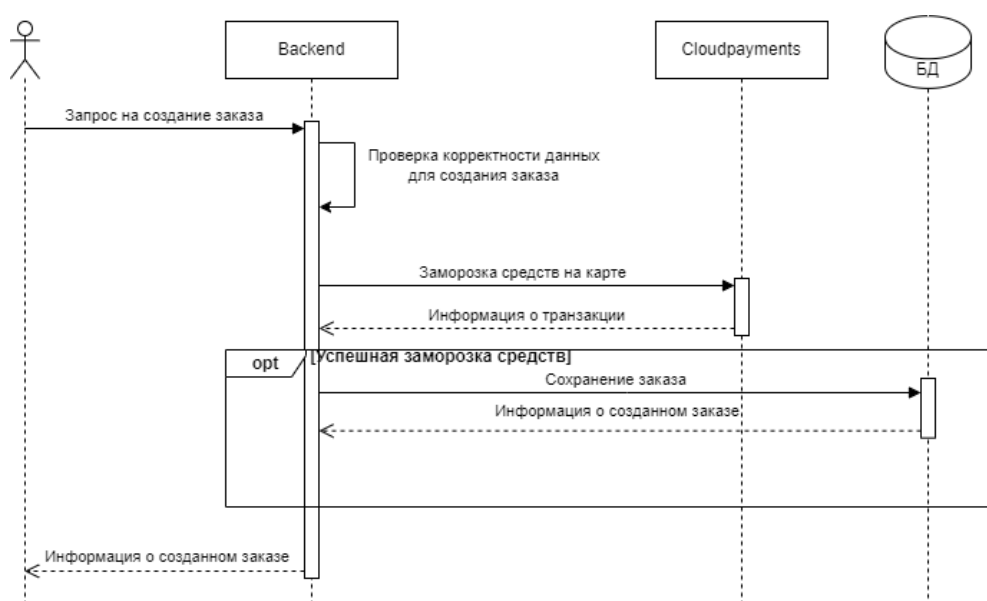


Рис. 2. Диаграмма последовательности процесса создания заказа

Для создания заказа пользователь добавляет товары в корзину, выбирает время получения, способ получения (самовывоз, в заведении или доставка), ранее привязанную банковскую карту для оплаты и отправляет запрос на создание заказа. Процесс создания заказа показан на рис. 2.

Диаграмма переходов заказа между статусами показана на рис. 3.

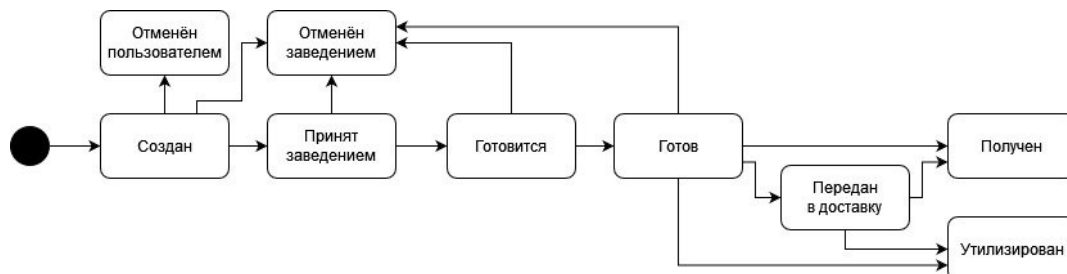


Рис. 3. Диаграмма статусов заказа

Так же заказ имеет статус оплаты. При создании заказа замораживаются средства на указанной банковской карте пользователя, в случае успешной заморозки устанавливается статус «средства заморожены», иначе «заморозка средств не удалась». Если заморозка средств не удалась покупатель может повторить попытку заморозки с той же карты или выбрать другую.

При отмене заказа покупателем или заведением до перехода заказа в статус «принят заведением» средства на банковской карте размораживаются. В случае ошибки разморозки попытка повторяется с определенным интервалом времени.

После принятия заказа в работу происходит подтверждение оплаты со списанием замороженных средств, а также отправка электронного чека на почту пользователя (процесс изображен на рис. 4).

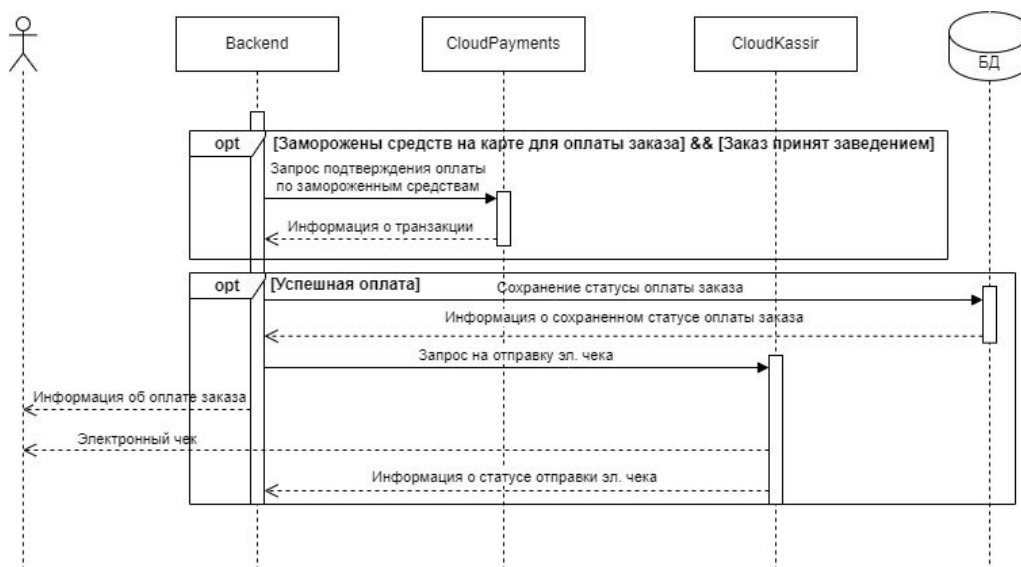


Рис. 4. Диаграмма последовательности оплаты заказа с отправкой эл. чека

В случае отмены заказа заведением средства возвращаются пользователю, если возврат средств произошел с ошибкой, он повторяется через определенный интервал времени. Диаграмма переходов между статусами оплаты заказа показана на рис. 5.

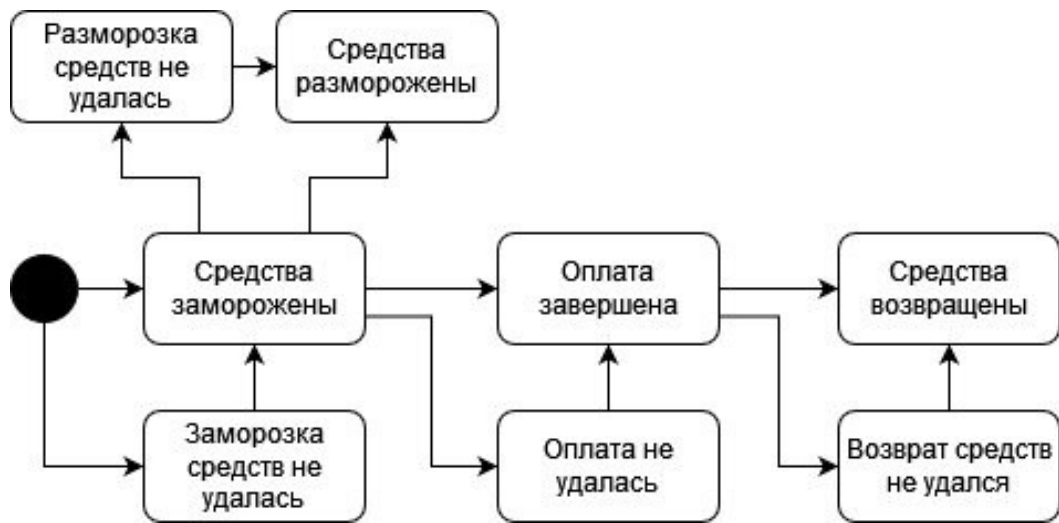


Рис. 5. Диаграмма статусов оплаты заказа

В случае заказа в заведении покупатель при желании может забронировать столик. Для этого он выбирает столик в заведении, и время начала и продолжительность бронирования. Процесс бронирования столика показан на рис. 6.

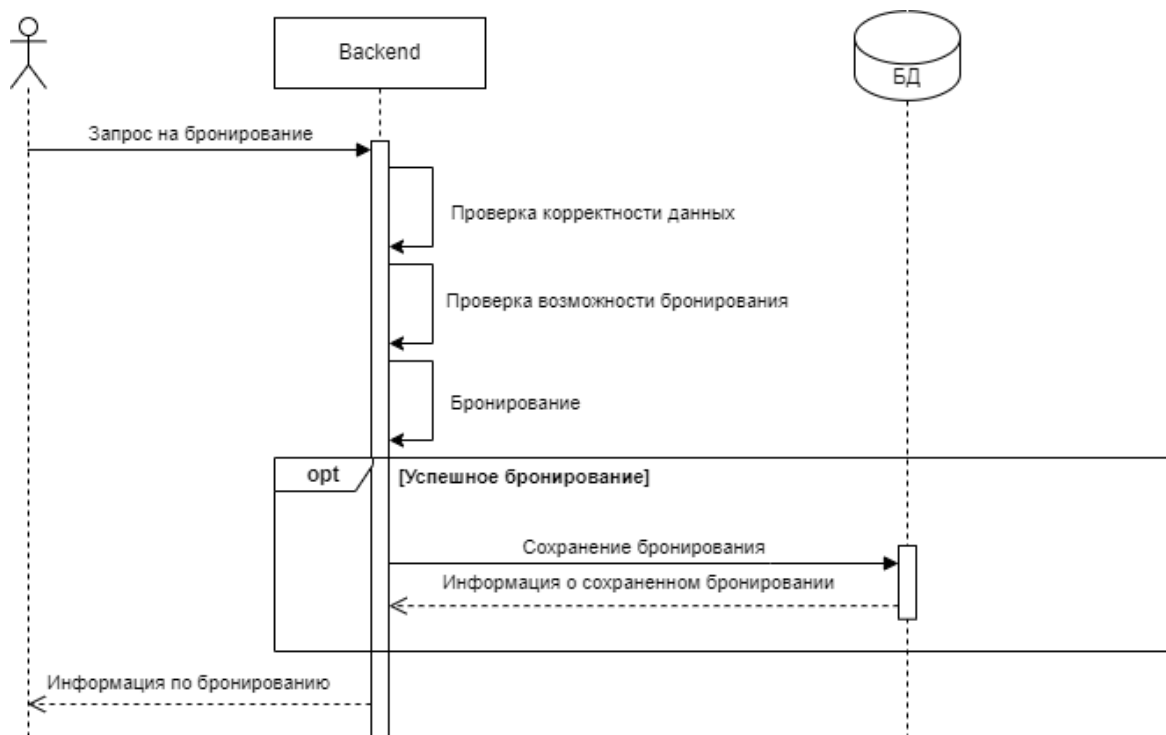


Рис. 6. Диаграмма последовательности процесса бронирования столика

Проектирование архитектуры

Архитектурные решения играют критически важную роль в обеспечении масштабируемости, надежности и гибкости системы, что в свою очередь влияет на ее долгосрочную эффективность и жизнеспособность.

Для технической реализации системы было решено использовать трехуровневую клиент-серверную архитектуру. В частности, сервер выступает в качестве посредника между клиентом и базой данных, обеспечивая безопасную запись, чтение, редактирование и удаление записей в таблицах базы данных.

Серверное приложение будет разрабатываться согласно архитектурному паттерну Model View Controller Service (MVCS). Применение MVCS позволит создать достаточно структурированную, гибкую и масштабируемую систему, что приведет к повышению качества разработки и удобству сопровождения.

Спроектирована отказоустойчивая архитектура (рис. 7), которая включает в себя следующие компоненты: балансировщик нагрузки, сервера и базы данных в репликации, а также мониторинг всех компонент системы. При отказе одного backend сервера, трафик автоматически будет переключен на резервные серверы, и работа системы не будет нарушена.

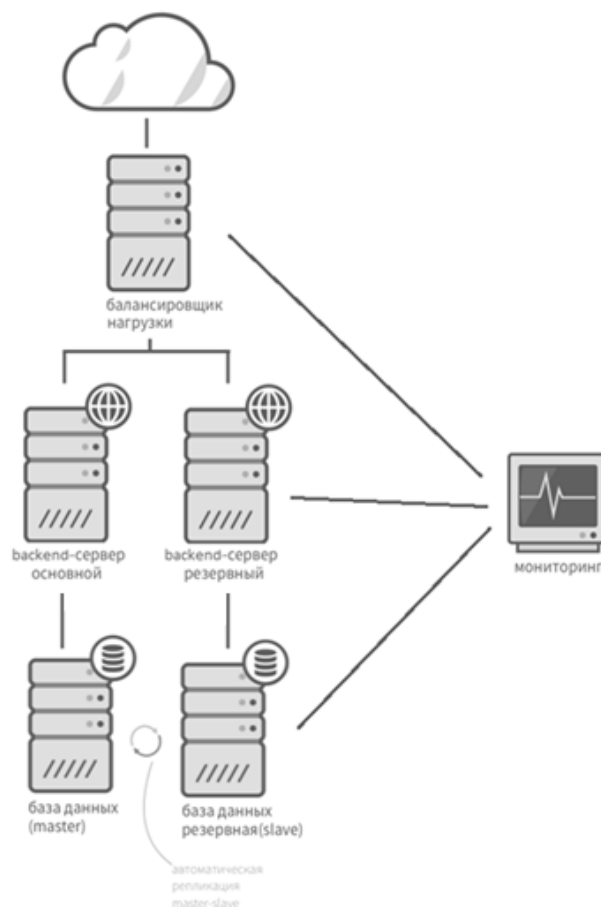


Рис. 7. Архитектура системы

Спроектирована база данных, выполнена нормализация до третьей нормальной формы. Модель базы данных представлена в виде ER-диаграммы (фрагмент на рис. 8).

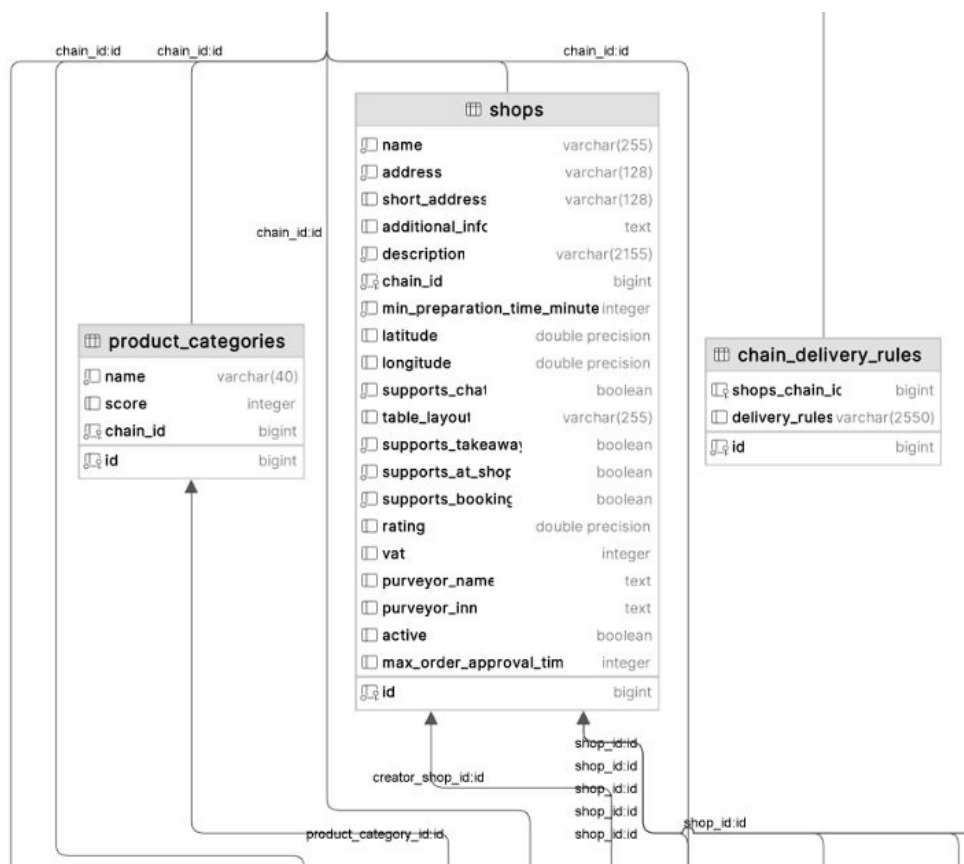


Рис. 8. Фрагмент ER-диаграммы

Для взаимодействия клиентской и серверной части API строится в соответствии с архитектурным стилем REST. Representational State Transfer (REST) – архитектурный стиль взаимодействия между компонентами, представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой системы [1].

При разработке веб-приложений решено использовать технологию SPA которая позволяет выполнить большинство требований, предъявляемых к системе.

Single page application (SPA) – концепция одностраничного веб-приложения, предполагающая использование архитектурного решения «Толстый клиент» (предоставляет собой функциональный интерфейс пользователя с выполнением бизнес-логики) и API серверной части для обмена данными [2].

В соответствии с концепцией SPA для рендеринга интерфейса применяется подход Client-Side Rendering. Клиентский рендеринг подразумевает рендеринг страниц в браузере с помощью JavaScript. Вся логика, получение данных, шаблонизация и маршрутизация обрабатываются на клиенте, а не на сервере [3].

Определение технологий и методологий для реализации проекта

В качестве системы управления версиями исходного кода выбран Git [4]. Для совместной разработки выбран GitLab [5]. Для автоматизации сборки и развертывания программного обеспечения будут использоваться Gitlab CI/CD пайплайны [6]. Для работы с окружениями и зависимостями будет использоваться технология контейнеризации Docker [7].

Для предоставления внешнего API выбран архитектурный стиль REST API с использованием стандарта HTTP и передачей данных в формате JSON. Это позволяет стандартизировать работу программных интерфейсов, сделать их более удобными и производительными. REST-подход к архитектуре позволяет сделать сервисы отказоустойчивыми, гибкими и производительными, а также снижает трудозатраты при масштабировании и внесении изменений [8].

Для асинхронного обмена данными через веб-сокеты используется библиотека Socket.IO.

Разработка серверной части ведется на языке программирования Java. Выбор обусловлен зрелостью языка, эффективностью и стабильностью, а также широкому распространению с активным сообществом и поддержкой.

Используемый Java-фреймворк – Spring Boot, выбран благодаря возможности упрощенной конфигурации и поддержке различных технологий (интегрируется с различными базами данных, инструментами кэширования, messaging-системами и др.).

Для автоматизации сборки проекта и управления зависимостями используется Apache Maven.

Используемый фреймворк для модульного и интеграционного автотестирования – JUnit.

В качестве системы управления базами данных выбран PostgreSQL. Обладает высокой надежностью, стабильностью и богатой функциональностью.

Балансировщиком нагрузки и прокси сервером выступает Nginx – свободно распространяемый веб-сервер. Обеспечивает высокую производительность, масштабируемость и возможность гибкой конфигурации.

Для разработки приложения был выбран фреймворк React Native. Выбор был сделан исходя из того, что этот фреймворк позволяет разрабатывать приложение одновременно под Android и iOS, что существенно сокращает время и ресурсы на разработку. React Native использует JavaScript, что упрощает интеграцию с существующими веб-приложениями и позволяет использовать одну и ту же кодовую базу для различных платформ, обеспечивая высокую производительность и нативный пользовательский интерфейс. Кроме того, благодаря широкому сообществу разработчиков и обилию готовых библиотек и компонентов, React Native позволяет быстро решать возникающие задачи и интегрировать новые функции.

Для реализации архитектуры Flux была выбрана библиотека Redux Toolkit. Это решение обосновано тем, что Redux Toolkit предоставляет упрощенные и стандартизированные способы работы с состоянием приложения, что позволяет сократить количество шаблонного кода и снизить вероятность ошибок. Это значительно облегчает разработку и поддержку сложных приложений, обеспечивая масштабируемость и предсказуемость управления состоянием.

Для реализации взаимодействия с сервером используется библиотека RTK-Query, которая является частью Redux Toolkit. RTK-Query обеспечивает мощные и гибкие средства для выполнения асинхронных запросов к серверу, управления кэшированием данных и обновлением состояния приложения на основе полученных данных. Благодаря интеграции с Redux, RTK-Query упрощает процесс синхронизации данных между клиентом и сервером, обеспечивая при этом высокую производительность и минимизацию количества сетевых запросов. Это особенно важно для приложений, работающих с большими объемами данных или требующих частого обновления информации в реальном времени.

Подготовка среды разработки

Для хранения исходного кода и конфигураций системы были созданы проекты в GitLab.

Далее выбраны и настроены следующие окружения разработки и развертывания программного обеспечения (рис. 9):

- local – компьютер(ы) разработчика(ов);
- development – сервер разработки, выступающий как песочница;
- testing – окружение для различных видов тестирования;
- staging – зеркало production, для финального тестирования;
- production – окружение, с которым непосредственно взаимодействуют конечные пользователи системы.

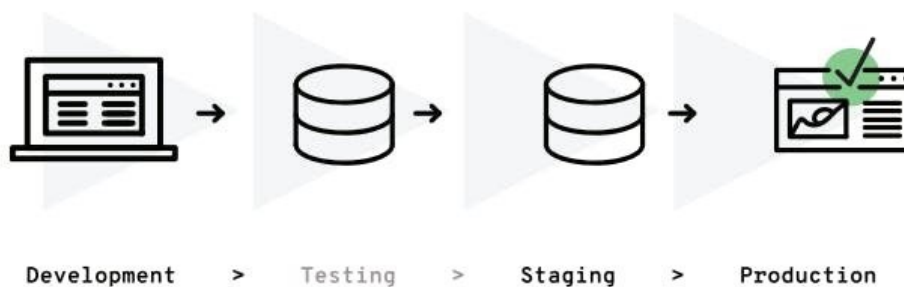


Рис. 9. Окружения развертывания программного обеспечения

Для разработки приложений под IOS, используя проект в GitHub – sickcodes/Docker-OSX [9], была запущена KVM виртуальная машина в Docker с операционной системой MacOS.

Реализованы Gitlab CI/CD пайплайны с шагами автотестирования, автоматической сборки и развертывания приложений.

UML диаграмма развертывания показана на рис. 10.

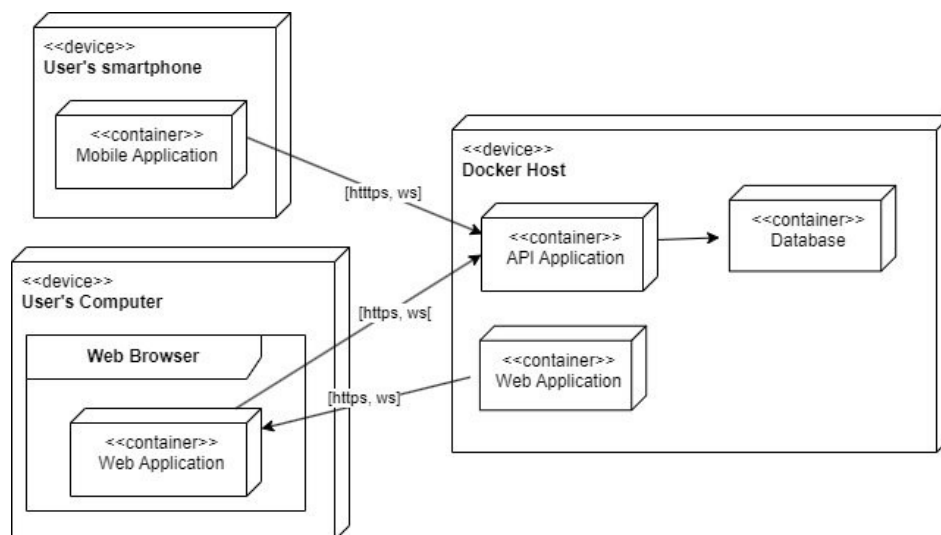


Рис. 10. Диаграмма развертывания

Программная реализация клиентской части

При разработке важно сохранять единый стиль кода, что обеспечивает повышение эффективности разработки, простоту восприятия и сопровождения. Для этого в проекте используются eslint8, stylelint9, prettier10. В конфигурации линтеров используется стиль от airbnb11 и плагины, рекомендованные разработчиками линтеров. Поскольку клиентская часть состоит из нескольких приложений, для удобства конфигурация линтеров была вынесена в отдельный npm пакет12, который был подключен во все проекты.

Для управления состояниями согласно идее Flux используется библиотека Redux Toolkit13.

Для сборки проектов мобильных приложений, реализуемых на React Native, используются несколько сборщиков:

Metro Bundler14 – интерпретирует JavaScript-код;

Gradle15 – компилирует Java-код для ОС android;

Xcode16 – компилирует Objective-C-код для ОС IOS.

Для сборки веб-приложений используется vite17.

Слабая типизация в JavaScript может сказаться на надежности системы, для решения данной проблемы было решено использовать статический типизатор Typescript18.

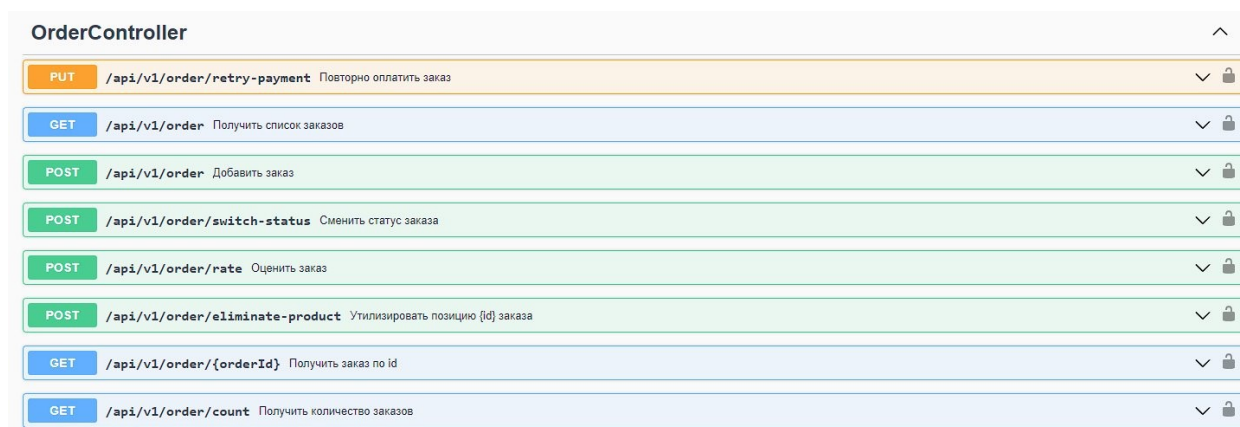
Для управления конфигурациями приложений применяется формат `env`. Для использования этого формата в мобильных приложениях применяется библиотека `react-native-config`¹⁹. В веб-приложениях в сборщик `vite` заложена поддержка такого формата, дополнительные библиотеки не требуются.

Программная реализация серверной части

При разработке серверной части на языке программирования Java был использован фреймворк Spring Boot.

База данных реализована в СУБД PostgreSQL согласно спроектированной ER-диаграмме. Заданы ограничения целостности, которые обеспечивают целостность и корректность данных, предотвращая ошибки, дубликаты и несогласованности в данных.

В ходе разработки реализован REST API, на текущий момент включающий более 70 обработчиков HTTP запросов. На рисунке 11 показан фрагмент из автогенерируемой спецификации openAPI/swagger.



OrderController		
PUT	/api/v1/order/retry-payment	Повторно оплатить заказ
GET	/api/v1/order	Получить список заказов
POST	/api/v1/order	Добавить заказ
POST	/api/v1/order/switch-status	Сменить статус заказа
POST	/api/v1/order/rate	Оценить заказ
POST	/api/v1/order/eliminate-product	Утилизировать позицию {id} заказа
GET	/api/v1/order/{orderId}	Получить заказ по id
GET	/api/v1/order/count	Получить количество заказов

Рис. 11. Фрагмент спецификации openAPI/swagger

Аутентификация в запросах реализована при помощи JSON Web Token (JWT). Получение токена происходит посредством подтверждения одноразового кода из звонка на номер телефона пользователя. Далее клиентское приложение взаимодействует с серверной частью с использованием JWT.

Валидация — это неотъемлемая часть бизнес-логики, поэтому, во всех обработчиках валидируются принимаемые параметры. В случае ошибок валидации в ответ возвращается понятное описание ошибки.

С целью обеспечения возможности аудита, все действия и ошибки в системе логируются с использованием следующей иерархии уровней событий (рис. 12). По событиям с уровнем выше ERROR настроены оповещения.

			x: Visible				
	FATAL	ERROR	WARN	INFO	DEBUG	TRACE	ALL
OFF							
FATAL	x						
ERROR	x	x					
WARN	x	x	x				
INFO	x	x	x	x			
DEBUG	x	x	x	x	x		
TRACE	x	x	x	x	x	x	
ALL	x	x	x	x	x	x	x

Рис. 12. Уровни логирования событий в системе

При работе с данными через API используются DTO (объекты передачи данных).

Данные отображаются из DTO в нужный формат модели предметной области и наоборот при помощи компонента Mapper на уровне представления (рис. 13). Это позволяет оптимизировать нагрузку при передаче данных и инкапсулировать модели предметной области от уровня представления.

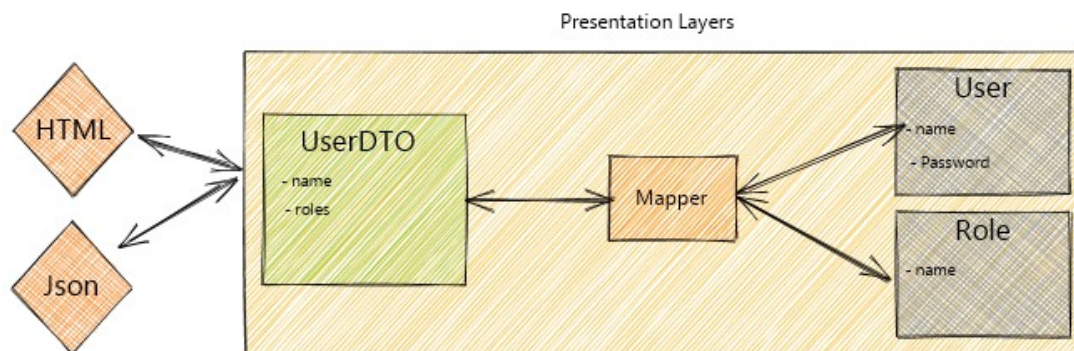


Рис. 13. Отображение данных DTO на уровне представления

Интеграция с внешними системами

Была произведена интеграция со следующими внешними системами.

Платформа Unibell [10] используется для аутентификации по одноразовому коду из звонка на номер телефона пользователя. Для заведений партнеров реализованы звонки с голосовыми уведомлениями о новых заказах.

В качестве эквайринга (сервиса для приема онлайн платежей) используется CloudPayments [11]. При осуществлении заказов используем двустадийную схему оплаты с заморозкой средств на карте клиента и последующим подтверждением списания средств или отменой платежа.

Облачная касса Kloud Kassir [12] используется для формирования онлайн чеков по заказам и отправки их на email.

Также, для решения проблемы нехватки курьеров была проведена интеграция с внешними сервисами доставки.

Разработка подсистемы автоматизированного SMM-маркетинга

Одной из ключевых проблем в сервисах агрегации является привлечение нового клиента. Сервис предзаказа и доставки не является исключением. Скорее наоборот, в данном направлении бизнеса цена за привлечение одного клиента еще более высокая. Чтобы адаптировать стратегию выхода на рынок к низким маркетинговым затратам разработана инновационная система автоматизированного SMM-маркетинга, в основе которой лежат несколько моделей машинного обучения, в том числе глубокое обучение (нейронные сети). Разработанный модуль позволяет автоматизировать часть воронки продаж, где цикл привлечения клиента, как в SMM-маркетинге, так и в других источниках интернет, выполняется с помощью нейронных сетей и уникальных алгоритмов. Система позволяет подобрать такие алгоритмы, которые позволят максимально эффективно привлечь новых пользователей в продукт, то есть с минимальным количеством вложенных в рекламу средств получить максимальное число новых лояльных пользователей. Данная система была подана на получение патента.

Тесты показали, что система позволяет снизить стоимость привлечения одного клиента в несколько раз. Укрупненно можно разделить систему на 3 составляющие:

- 1) определение наиболее лояльных целевых групп;
- 2) генерация максимально эффективных рекламных постов, на основе накопленного опыта конкурентов и смежных практик;
- 3) определение оптимальных мест для размещения готовых рекламных постов, способных обеспечить эффективный результат.

Система привлечения опробована в минимальном функциональном наполнении, получен положительный результат, ведется дальнейшая работа над расширением функционала системы, в том числе по накоплению исторических данных и их аналитике.

Заключение

В данной работе была описана разработка информационной системы. Был проведен анализ процессов, выдвинуты функциональные и нефункциональные требования. Проведено проектирование системы. Была детально рассмотрена подготовка среды разработки, программная реализация клиентской и серверной части системы, интеграция с внешними системами, разработка подсистемы автоматизированного SMM-маркетинга.

Библиографический список

1. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures: Dissertation submitted in partial satisfaction of the requirements for the degree of doctor of philosophy in information and computer science / Roy Thomas Fielding., University of California – Irvine, 2000. 162 p.
2. Mikowski M., Powell J. Single Page Web Applications. Manning Publications, 2014. 432 p.
3. Серверный или клиентский рендеринг на вебе: что лучше использовать у себя в проекте и почему [Электронный ресурс]. 2019. URL: <https://tproger.ru/translations/rendering-on-the-web/> (дата обращения 30.05.2024).
4. Git [Электронный ресурс]. URL: <https://git-scm.com/> (дата обращения: 15.05.2024).
5. GitLab Documentation [Электронный ресурс]. URL: <https://docs.gitlab.com/> (дата обращения: 15.05.2024).
6. Get started with GitLab CI/CD [Электронный ресурс]. URL: <https://docs.gitlab.com/ee/ci/> (дата обращения: 15.05.2024).
7. Docker: Accelerated Container Application Development [Электронный ресурс]. URL: <https://www.docker.com/> (дата обращения: 15.05.2024).
8. REST API: что это такое и как работает [Электронный ресурс]. URL: <https://skillbox.ru/media/code/rest-api-cto-eto-takoe-i-kak-rabotaet/> (дата обращения: 15.05.2024).
9. GitHub – sickcodes/Docker-OSX: Run macOS VM in a Docker! [Электронный ресурс]. URL: <https://github.com/sickcodes/Docker-OSX> (дата обращения: 15.05.2024).
10. Платформа от компании «Unibell» [Электронный ресурс]. URL: <https://www.unibell.ru/platform.html> (дата обращения: 30.05.2024).
11. Интернет-эквайринг CloudPayments [Электронный ресурс]. URL: <https://cloudpayments.ru> (дата обращения: 30.05.2024).
12. Облачная онлайн-касса CloudKassir [Электронный ресурс]. URL: <https://cloudkassir.ru> (дата обращения: 30.05.2024).

References

1. Fielding R.T. Architectural Styles and the Design of Network-based Software Architectures: Dissertation submitted in partial satisfaction of the requirements for the degree of doctor of philosophy in information and computer science / Roy Thomas Fielding., University of California – Irvine, 2000. 162 p.
2. Mikowski M., Powell J. Single Page Web Applications. Manning Publications, 2014. 432 p.
3. Servernyj ili klientskij rendering na vebe: chto luchshe ispol'zovat' u sebya v proekte i pochemu [Electronic resource]. 2019. URL: <https://tproger.ru/translations/rendering-on-the-web/> (date of access: 05/30/2024). (In Russ.)

4. Git [Electronic resource]. URL: <https://git-scm.com/> (date of access: 05/15/2024).
5. GitLab Documentation [Electronic resource]. URL: <https://docs.gitlab.com/> (date of access: 05/15/2024).
6. Get started with GitLab CI/CD [Electronic resource]. URL: <https://docs.gitlab.com/ee/ci/> (date of access: 05/15/2024).
7. Docker: Accelerated Container Application Development [Electronic resource]. URL: <https://www.docker.com/> (date of access: 05/15/2024).
8. REST API: что это такое и как работает [Electronic resource]. URL: <https://skillbox.ru/media/code/rest-api-cto-eto-takoe-i-kak-rabotaet/> (date of access: 05/15/2024).
9. GitHub – sickcodes/Docker-OSX: Run macOS VM in a Docker! [Electronic resource]. URL: <https://github.com/sickcodes/Docker-OSX> (date of access: 05/15/2024).
10. Platforma ot kompanii «Unibell» [Electronic resource]. URL: <https://www.unibell.ru/platform.html> (date of access: 05/30/2024). (In Russ.)
11. Internet-ekvajring CloudPayments [Electronic resource]. URL: <https://cloudpayments.ru> (date of access: 05/30/2024). (In Russ.)
12. Oblachnaya onlajn-kassa CloudKassir [Electronic resource]. URL: <https://cloudkassir.ru> (date of access: 05/30/2024). (In Russ.)

DESIGN AND DEVELOPMENT OF AN INFORMATION SYSTEM FOR BUSINESS MANAGEMENT IN THE FOOD SERVICE INDUSTRY

Sychev Ivan A., Sokolova Olga L., Yashichev Dmitry L., Sokolov Andrey V.

Perm State National Research University, 15 Bukireva str., Perm, 614068, Russia,
info@interprogram.ru

The aim of this work is to design and develop an information system for pre-ordering and delivery. The work includes an analysis of existing processes and stakeholders. Based on the analysis, tasks for the design were set and requirements for the developed system were formulated. In accordance with the specified requirements, an information system was designed and implemented. The results of this work may be of interest to food service organizations looking to expand their market, customers seeking convenient ways to order food, and couriers.

Keywords: information system, process optimization, pre-order, delivery, food service.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПРАКТИКА НА БИОЛОГИЧЕСКОМ ФАКУЛЬТЕТЕ ПГНИУ»

Уткина Александра Владимировна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, utkina22985@mail.ru

Кнутова Наталия Сергеевна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, talieknutova@gmail.com

Рассматриваются основные пункты проектирования и моделирования информационных систем. Проводится проектирование информационной системы для биологического факультета Пермского государственного национального исследовательского университета, которая предназначена для цифровизации осуществления измерений в разных областях биологических наук и систематизации хранения их результатов. Анализируются проблемные факторы обучения на биологическом факультете, выступающие частью определения задач проекта. Обозреваются инструменты и приложения, относящиеся к предметной области, в том числе лабораторные информационные системы, оцениваются средства проектирования и разработки. Определены требования к проектированию, выбрана мобильная платформа реализации. С целью построения базы данных показана диаграмма сущность-связь в нотации Crow's foot. Для описания ключевых функций использована диаграмма прецедентов. Для конкретизации последовательности действий и выполняемых операций продемонстрирована диаграмма деятельности. Прилагается пример интерфейса приложения.

Ключевые слова: информационные системы, проектирование, диаграммы UML, вёрстка приложения.

Биологический факультет ПГНИУ отвечает за формирование практических знаний студентов в многочисленных областях, однако в рамках исследования рассмотрены три. Физиология растений содержит в себе темы: физиология растительных клеток и тканей, устойчивость растений, метаболические системы, жизнедеятельность растений, диагностика устойчивости культур к экстремальным факторам внешней среды [1, с. 2]. Микробиология отвечает за: микроскопирование, культивирование, посев, идентификация и изучение клеток, выделение чистых культур микроорганизмов, выявление клеточных структур, использование микрометодов определения численности колониеобразующих микроорганизмов [2, с. 2]. Проблемой выступает дороговизна оборудования для перепроверки верности подсчёта выведенных колоний. На текущий момент студенты и научные сотрудники используют eCountTm (ручка-счётчик колоний микроорганизмов от Heathrow Scientific [3]). Полевые исследования осматривают: изучение биоразнообразия и определение видов высших растений, полевые

методы изучения растительных сообществ, основные полевые методы изучения растительности [4, с. 2]. Отсутствие опыта и развитого глазомера может привести к получению неверных данных, нелогичность которых формирует ложные выводы о местности. Для исключения человеческого фактора студенты могут использовать ручной датчик урожайности GreenSeeker [5], что приводит к дополнительным затратам на инструментарий, обладающий возможностью разрядки или поломки. Безусловно, специалист своей области обязан уметь использовать ручные методы замеров и исследований. Однако, студентам может потребоваться больше времени на изучение метода и примеров к нему, чем доступно во время практических занятий.

Подобное положение дел уменьшает приближённость проведённых лабораторных работ к действительности. В связи с чем целью исследования выступает проектирование информационной системы, направленной на предоставление эффективных инструментов для прохождения практик на биологическом факультете ПГНИУ.

Обзор существующих систем

Для регистрации и наблюдения за изменениями полученных данных могут быть использованы лабораторные информационные системы ЛИМС АИСТ [6] и ЛИС ULAB [7]. В таблице приведено сравнение двух указанных систем.

Таблица

Сравнение лабораторных информационных систем

Параметр	ЛИМС АИСТ	ЛИС ULAB
Минимальная стоимость лицензии на компанию до 12 чел.	от 1 600 000,00 Р	от 399 000,00 Р
Целевая аудитория, лаборатории	пробирного анализа; геологические; алюминиевого производства, цементного производства	независимые испытательные; ориентированные на внутреннего заказчика; строительные; пищевые; физических факторов; дорожные; нефтехимические эколого-аналитические
Возможность доработки по запросу клиента	Наличие персональных доработок системы	Наличие персональных доработок системы
Облачные хранения	Нет информации	Присутствует
Мобильная версия	В качестве отдельного модуля для просмотра и управления данными	В качестве отдельного модуля
Основные функции	Управление отбором проб: место отбора, материал, регистрация факта отбора, создание заданий. Управление лабораторными журналами, рабочими листами, протоколами отчётов, паспортами продукции и другими бланковыми отчётами. Универсальный отчёт исследования по всей информации в системе	Управление взаимоотношениями с клиентами. Ведение нормативной документации. Учёт оборудования. Контроль условий проведения испытаний. Формирование документации и отчётности. Возможность интеграции с внешними системами

Кроме профессионального программного обеспечения доступны офисные инструменты для хранения информации, к примеру, приложение «МойОфис Документы» [8].

Среди недостатков существующих систем выделяется дороговизна ИС предметной области, отсутствие унифицированного решения, неполнота функционала мобильных версий.

Обоснование выбора языка программирования и фреймворка

Основываясь на соображении о целесообразности внедрения средств машинного обучения, список выбираемых ЯП может быть ограничен следующими: Python [9], Java [10], R [11]. Помехой для использования R является узкая специализация языка на расчёты и анализ данных, а не на разработку ПО. Несмотря на сходства и достоинства Java и Python, для формирования ИС выбран язык Python. Его синтаксис и правила применения изучаются в ходе обучения, что позволяет вовлечь в производство модулей студентов университета и расширить функционал программы и область знаний программистов.

Сообщество языка программирования Python предлагает несколько разнотипных фреймворков, среди которых рассматриваются Kivy [12] и BeeWare [13]. Так как BeeWare не предоставляет возможности применения стилизованного интерфейса, итоговым выбором для разработки выступает фреймворк Kivy.

Обзор средств проектирования приложения

Среди наиболее популярных программ и веб-инструментов выступает сервис Lucidchart [14]. Однако, Lucidchart базируется на плановой подписке и требует подключение к сети Интернет. Платформой для быстрого командного моделирования выделяется Miro [15]. Недостатки: ограничение на количество досок и необходимость подключения платной подписки. В качестве основного средства проектирования выбрано приложение Diagrams.net [16]. В нём пользователю доступно облачное хранение созданных проектов, их скачивание, экспортирование png-файлов.

Обзор средств проектирования интерфейса

Основным поставщиком продуктов для дизайна и иллюстрации выступает компания Adobe [17]. Однако, Adobe не может предложить универсальное решение, подходящее для каждого проекта. Sketch – универсальный продукт для создания прототипов разных форматов для разных платформ [18]. Недостатком выступает доступ для пользователей операционной системы macOS. Выбранным инструментом для проектирования интерфейса выступает платформа figma [19]. Она предназначена для проектирования и генерации кода.

Проектирование логической части информационной системы

Проведено моделирование данных, продемонстрированное в диаграмме сущность-связь (ERD) нотации Crow's foot. На рисунке 1 представлены сущности ИС, которые концептуально можно поделить на три группы.

Главные сущности. Модуль – реализация инструмента проведения замера для той или иной области практики. Пользователь – зарегистрированный человек, который использует приложение. Замер – именованное использование инструмента-модуля. Замер содержит измерения, которые содержат параметры для изучения.

Вспомогательные сущности. Тип пользователя – сущность, необходимая для указания на целочисленный тип пользователя. Группа пользователей – это группа студентов, сформированная и зарегистрированная преподавателем. Измерение – сессия использования инструмента-модуля. Параметр – сущность измеренной величины. Картинка – сущность для хранения изображений, используемых в замерах.

Связующие сущности. Группа-модуль – сущность необходима для управления доступом группы к тому или иному модулю. Принадлежность к группе – сущность, используемая для закрепления пользователя-студента к учебной группе.

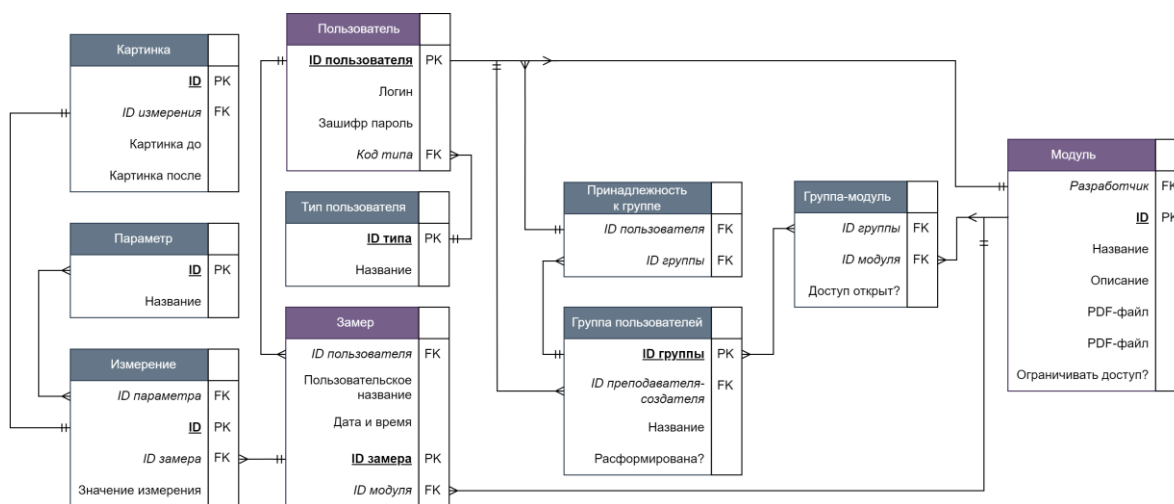


Рис. 1. ER-диаграмма нотации «Воронья лапка»

Диаграмма прецедентов ИС представлена в приложении А. Актор «Пользователь» введён для обобщения преподавателей, научных сотрудников и студентов. Для упрощения чтения схемы введено цветное разделение действий.

Диаграмма деятельности представлена в приложении Б.

В качестве пользователя рассматривается тип пользователя «преподаватель», в качестве серверной оболочки – взаимодействие с базой данных, в качестве локальной оболочки – само приложение, в качестве модуля – любой модуль-инструмент.

Первым блоком выступает блок запуска приложения, демонстрирующий взаимодействие между пользователем и оболочками. Блок входа пользователя оперирует вариациями ситуаций входа пользователя при разных начальных условиях. Блок использования приложения – последний этап диаграммы деятельности. Вне зависимости от типа подключения, основные действия реализуются локальной оболочкой без вмешательства серверной части.

Проектирование интерфейса

Дизайн оболочки ИС придерживается стиля графических элементов модулей. Приложение предполагает, что пользователь реализует цепочку последовательных действий, из-за чего на одну страницу приходится одна транзакция. Каждое окно может состоять из элементов, приведённых на рис. 2. Используется шрифт семейства Montserrat.

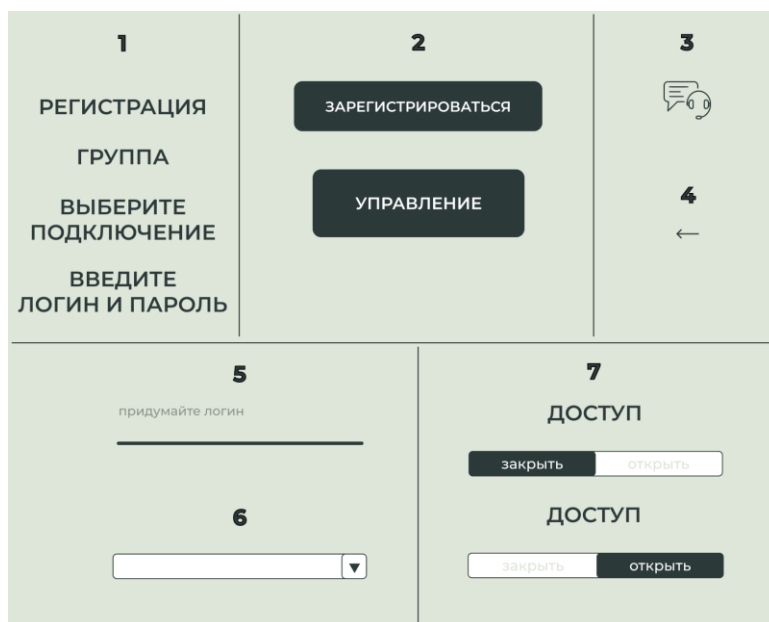


Рис. 2. Элементы приложения



Рис. 3. Пример окон загрузки

Кроме перечисленных элементов, в приложении могут быть добавлены иные объекты.

Для моментов, требующих ожидания пользователя, будет появляться окно загрузки (см. рис. 3), из которого также можно будет выйти на прошлую страницу, вызвать помощь или дождаться смены окружения.

Не все типы окон будут доступны каждому пользователю. Так, для типа пользователя «преподаватель» спроектирована отдельная категория окон (см. рис. 4).

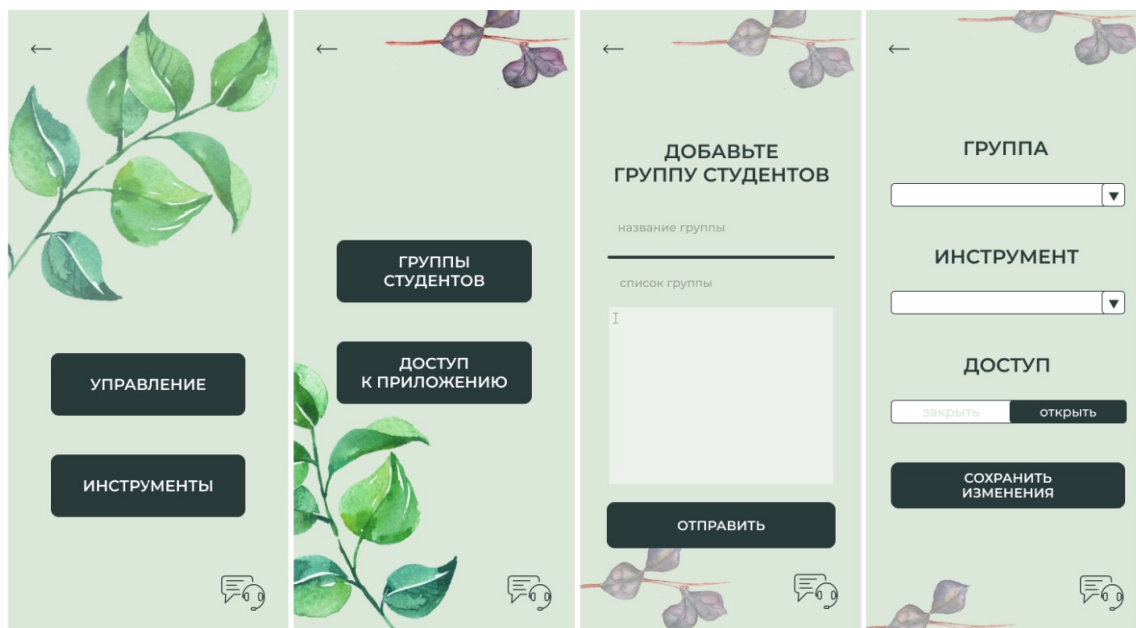


Рис. 4. Пример окон для ограниченного типа пользователей

Результаты

Спроектирована информационная система «Практика на биологическом факультете ПГНИУ». Указанный проект отвечает за формирование среды для предоставления инструментов анализа данных, собираемых за время прохождения практики или лабораторных исследований, определения и систематизации полученных характеристик, сохранения результатов вычислений в едином информационном пространстве. Доступность среды достигается использованием мобильного приложения и независимостью от подключения к глобальной сети.

Для реализации продукта выбраны следующие инструменты: язык программирования Python, кроссплатформенный фреймворк Kivy, библиотека компьютерного зрения OpenCV.

Для проектирования ИС использовались веб-сервисы diagrams.net и figma.com.

Благодаря внедрению планируемого проекта биологический факультет сможет познакомить студентов с примерами инструментов лабораторных и иных профессиональных систем. Также, уменьшится процент получения ложного опыта взаимодействия с изучаемыми темами за счёт применения машинного обучения. А благодаря контролю процессу использования вспомогательных средств при выполнении заданий факультет сможет минимизировать объём закупаемого профессионального оборудования.

Библиографический список

1. Четина О. А., Чудинова Л. А. Физиология растений. Лабораторные работы: учеб. пособие / Перм. гос. нац. исслед. ун-т. Пермь, 2020. 88 с.
2. Ившина И. Б. Микробиология. Рабочая тетрадь для лабораторно-практических занятий: практикум / И. Б. Ившина, Т. Н. Каменских, Е. А. Тюмина, А. А. Елькин. Перм. гос. нац. исслед. ун-т. Пермь, 2022. 100 с.
3. ECOUNT COLONY COUNTER [Электронный ресурс] / Heathrow Scientific. URL: <https://www.heathrowscientific.com/ecount-colony-counter/> (дата обращения: 03.12.2023).
4. Овеснов С. А. Биоразнообразие и экология высших растений: учеб. пособие по учеб. практике / Перм. гос. ун-т. Пермь, 2009. 129 с.
5. Ручной датчик урожайности GreenSeeker [Электронный ресурс] / PTx Trimble LLC. URL: <https://ru.agriculture.trimble.com/product/ручной-датчик-урожайности-greenseeker/> (дата обращения: 03.12.2023).
6. Система управления измерениями и испытаниями ЛИМС АИСТ [Электронный ресурс] / ГК «Термо Техно». М., 2020. URL: <https://lims-aist.ru> (дата обращения: 12.04.2024).
7. Система автоматизации деятельности испытательной лаборатории ULAB [Электронный ресурс] / ООО «НИИСТРОМ». Челябинск, 2019. URL: <https://niistrom.pro/#about> (дата обращения: 12.04.2024).
8. Мобильное приложение «МойОфис Документы» [Электронный ресурс] / ООО «Новые Облачные Технологии». Иннополис, 2013. URL: <https://myoffice.ru/apps/mobile-documents/> (дата обращения: 12.04.2024).
9. Python [Электронный ресурс] / Python Software Foundation. 2001. URL: <https://www.python.org> (дата обращения: 15.01.2024).
10. Java [Электронный ресурс] / Oracle. URL: <https://www.java.com/ru/> (дата обращения: 15.01.2024).
11. R [Электронный ресурс] / The R Foundation. URL: <https://www.r-project.org> (дата обращения: 15.01.2024).
12. Kivy [Электронный ресурс]. URL: <https://kivy.org> (дата обращения: 23.04.2024).
13. BeeWare [Электронный ресурс] / Russell Keith-Magee. URL: <https://beeware.org> (дата обращения: 23.04.2024).
14. Lucidchart [Электронный ресурс] / Lucid Software Inc. URL: <https://www.lucidchart.com/pages/ru> (дата обращения: 24.05.2024).
15. Miro [Электронный ресурс]. URL: <https://miro.com/ru/> (дата обращения: 24.05.2024).
16. Diagrams.net [Электронный ресурс] / JGraph Ltd. 2005. URL: <https://app.diagrams.net> (дата обращения: 20.01.2024).

17. Adobe [Электронный ресурс]. URL: <https://www.adobe.com/ru/> (дата обращения: 24.05.2024).

18. Sketch [Электронный ресурс] / Sketch B.V. URL: <https://www.sketch.com> (дата обращения: 24.05.2024).

19. Figma [Электронный ресурс] / Figma, Inc. URL: <https://www.figma.com> (дата обращения: 20.01.2024).

References

1. Chetina O.A., Chudinova L.A. Fiziologiya rastenij. Laboratorny`e raboty`: ucheb. posobie / Perm. gos. nacz. issled. un-t. Perm`, 2020. 88 p. (In Russ.)

2. Ivshina I. B. Mikrobiologiya. Rabochaya tetrad` dlya laboratorno-prakticheskix zanyatij : praktikum / I. B. Ivshina, T. N. Kamenskix, E. A. Tyumina, A. A. El`kin. Perm. gos. nacz. issled. un-t. Perm`, 2022. 100 p. (In Russ.)

3. ECOUNT COLONY COUNTER [Electronic resource] / Heathrow Scientific. URL: <https://www.heathrowscientific.com/ecount-colony-counter/> (date of access: 12/03/2023). (In Russ.)

4. Ovesnov S. A. Bioraznoobrazie i e`kologiya vy`sshix rastenij: ucheb. posobie po uchebnoj praktike / Perm. gos. un-t. Perm`, 2009. 129 p. (In Russ.)

5. Ruchnoj datchik urozhajnosti GreenSeeker [Electronic resource] / PTx Trimble LLC. URL: <https://ru.agriculture.trimble.com/product/ruchnoj-datchik-urozhajnosti-greenseeker/> (date of access: 12/03/2023). (In Russ.)

6. Sistema upravleniya izmereniyami i ispy`taniyami LIMS AIST [Electronic resource] / GK «Termo Texno». M., 2020. URL: <https://lims-aist.ru> (date of access: 12.04.2024). (In Russ.)

7. Sistema avtomatizacii deyatel`nosti ispy`tatel`noj laboratorii ULAB [Electronic resource] / OOO «NIISTROM». Chelyabinsk, 2019: URL: <https://niistrom.pro/#about> (date of access: 04/12/2024). (In Russ.)

8. Mobil`noe prilozhenie «MojOfis Dokumenty`» [Electronic resource] / OOO «Novy`e Oblachny`e Teknologii». Innopolis, 2013. URL: <https://myoffice.ru/apps/mobile-documents/> (date of access: 04/12/2024). (In Russ.)

9. Python [Electronic resource] / Python Software Foundation. 2001. URL: <https://www.python.org> (date of access: 01/15/2024).

10. Java [Electronic resource] / Oracle. URL: <https://www.java.com/ru/> (accessed: 01/15/2024).

11. R [Electronic resource] / The R Foundation. URL: <https://www.r-project.org> (date of access: 01/15/2024).

12. Kivy [Electronic resource]. URL: <https://kivy.org> (date of access: 04/23/2024).

13. BeeWare [Electronic resource] / Russell Keith-Magee. URL: <https://beeware.org> (date of access: 04/23/2024).
14. Lucidchart [Electronic resource] / Lucid Software Inc. URL: <https://www.lucidchart.com/pages/ru> (date of access: 05/24/2024).
15. Miro [Electronic resource]. URL: <https://miro.com/ru/> (date of access: 05/24/2024).
16. Diagrams.net [Electronic resource] / JGraph Ltd. 2005. URL: <https://app.diagrams.net> (date of access: 01/20/2024).
17. Adobe [Electronic resource]. URL: <https://www.adobe.com/ru/> (date of access: 05/24/2024).
18. Sketch [Electronic resource] / Sketch B.V. URL: <https://www.sketch.com> (date of access: 05/24/2024).
19. Figma [Electronic resource] / Figma, Inc. URL: <https://www.figma.com> (date of access: 01/20/2024).

**DESIGNING THE INFORMATION SYSTEM
«PRACTICE AT THE FACULTY OF BIOLOGY OF PSU»**

Utkina Alexandra V.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, utkina22985@mail.ru

Knutova Nataliya S.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, talieknutova@gmail.com

The main points of information systems design and modeling are considered. An information system is being designed for the Faculty of Biology of PSU, which is designed to digitalize measurements in various fields of biological sciences and systematize the storage of their results. The problematic factors of teaching at the Faculty of Biology, which are part of the definition of the project objectives, are analyzed. Tools and applications related to the subject area are reviewed, including laboratory information systems, and design and development tools are evaluated. Design requirements have been defined, and a mobile implementation platform has been selected. In order to build a database, an entity-relationship diagram in Crow's foot notation is shown. A use case diagram is used to describe the key functions. To specify the sequence of actions and operations performed, an activity diagram is demonstrated. An example of the application interface is attached.

Keywords: information systems, design, UML diagrams, application layout.

Диаграмма прецедентов

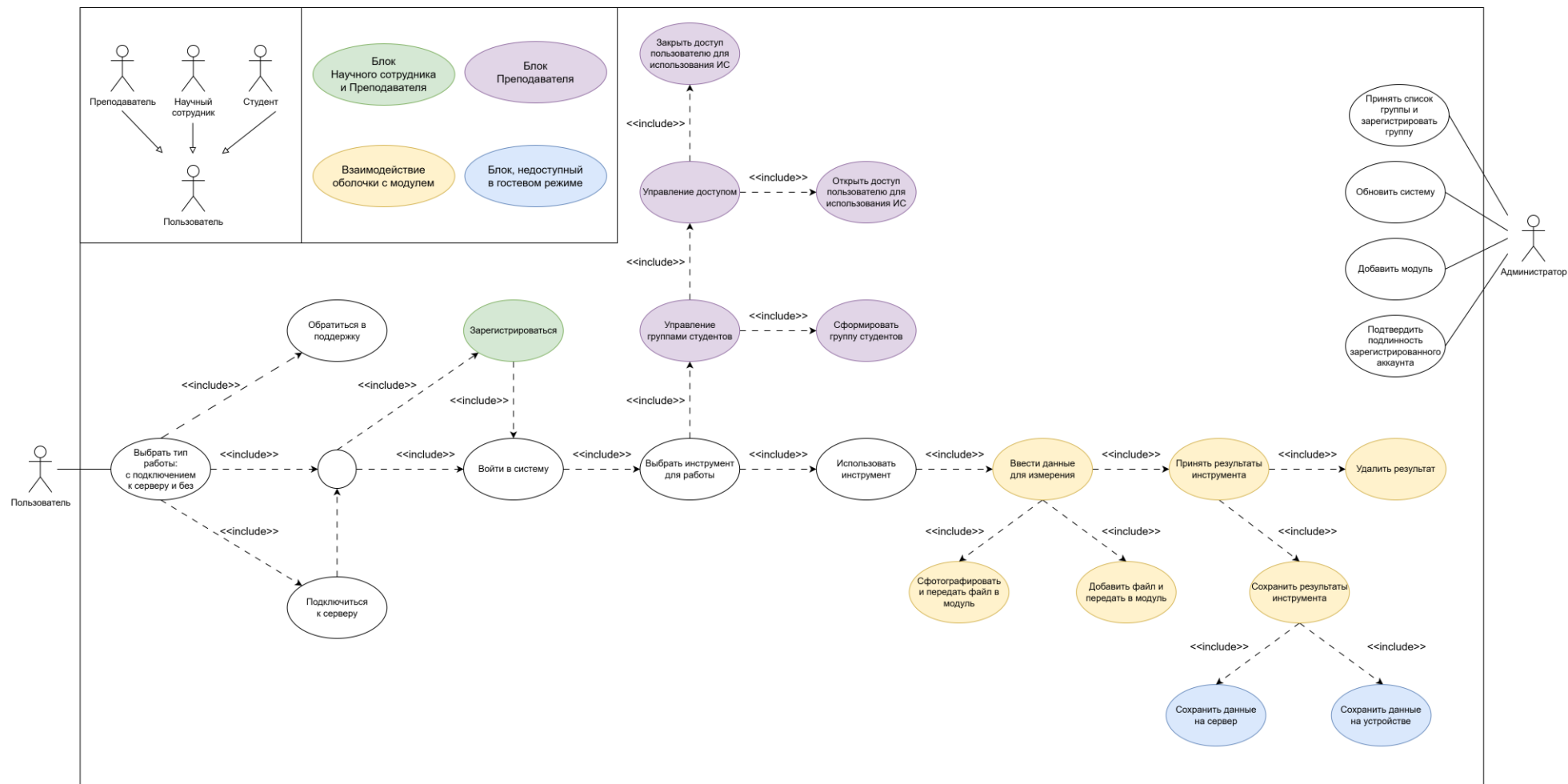


Диаграмма деятельности

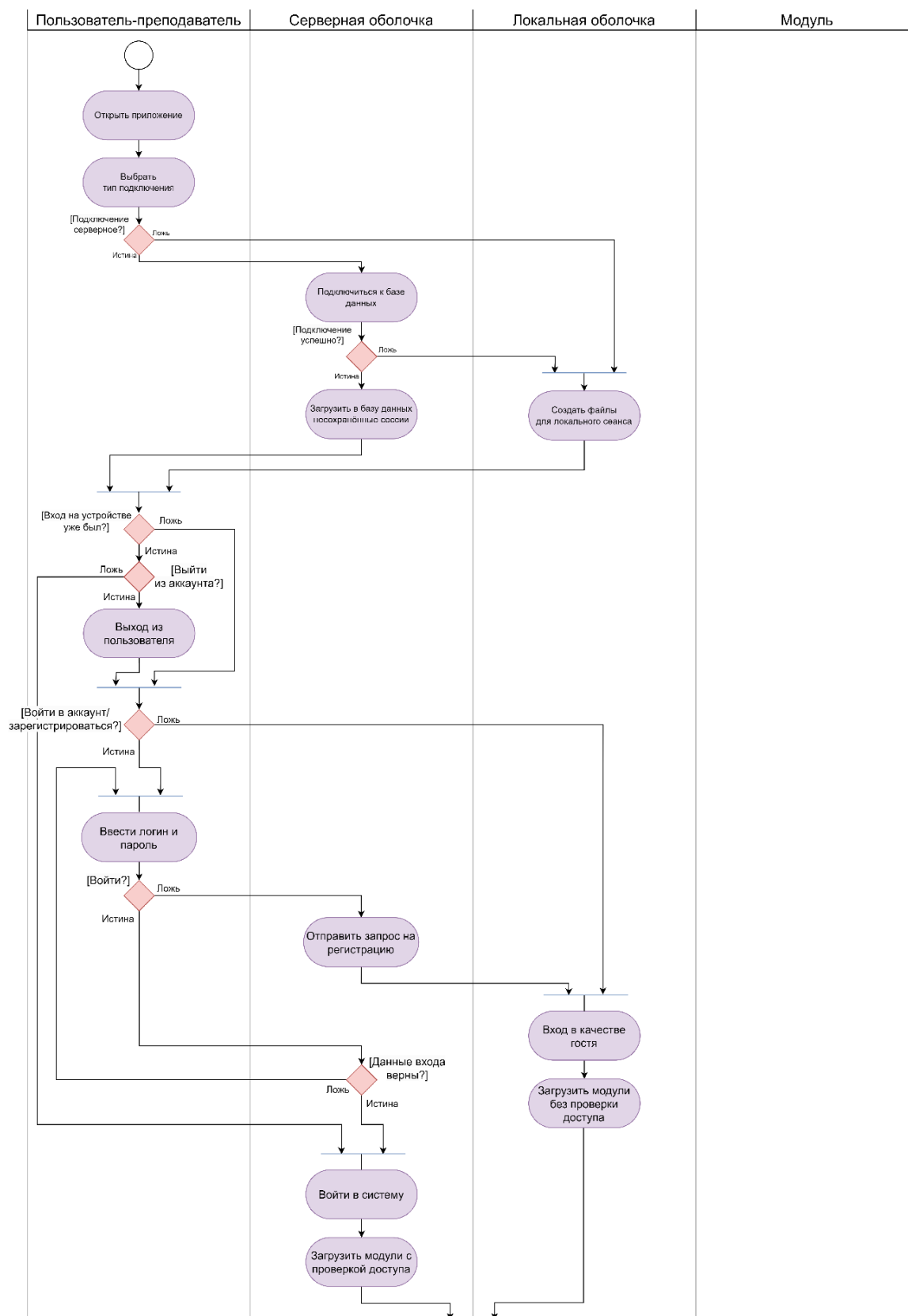


Рис. Б.1. Диаграмма деятельности, блоки запуска и входа

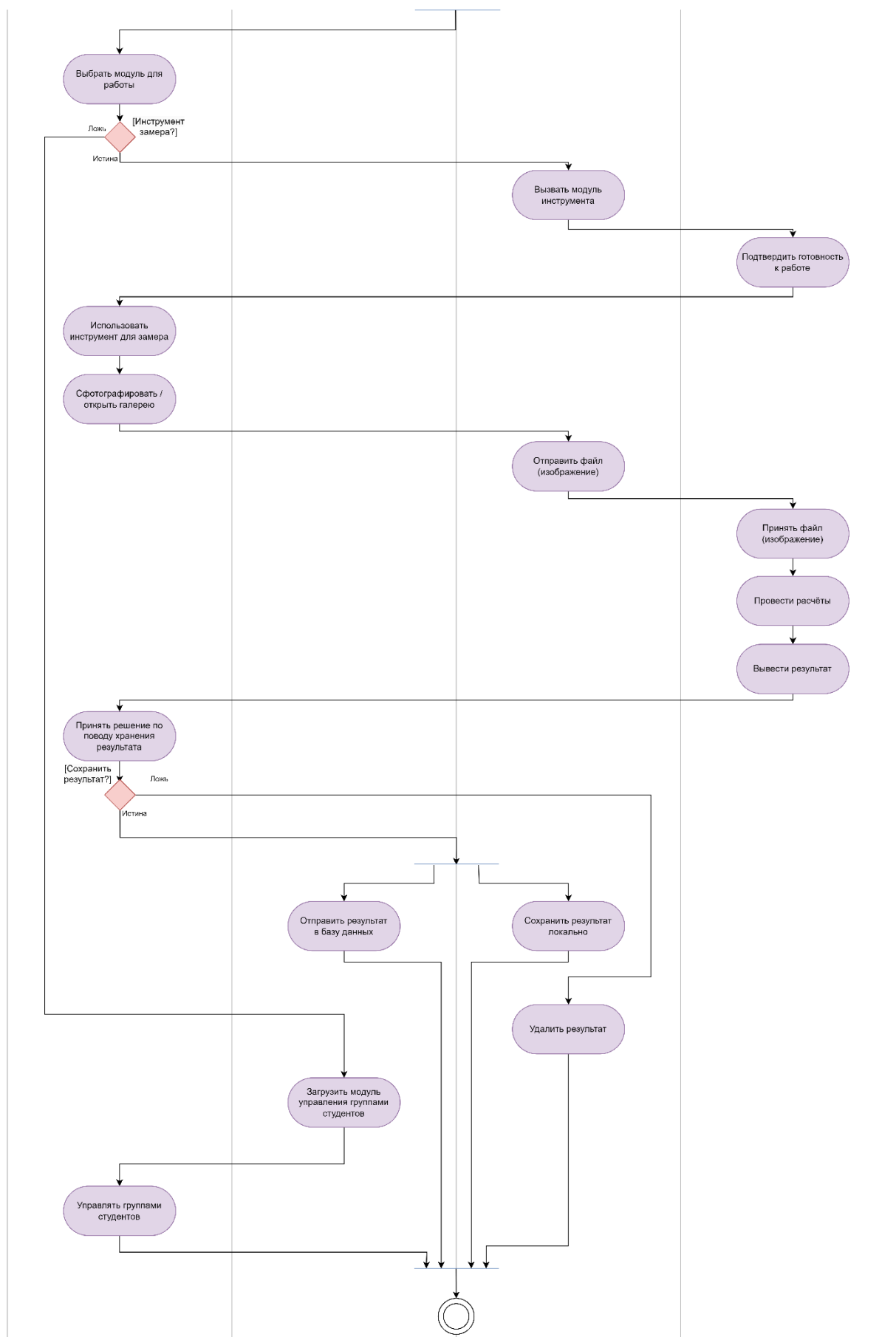


Рис. Б.2. Диаграмма деятельности, блок использования приложения

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРЕПОДАВАНИЯ БЛОКА АЛГОРИТМИЗАЦИИ В ДИСЦИПЛИНЕ «ИНФОРМАТИКА»

Шаров Роман Владиславович, Ракина Валерия Денисовна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, roma752275@mail.ru

В процессе обучения студентов ПГНИУ (Пермский государственный национально исследовательский университет) блоку алгоритмизации в рамках дисциплины “Информатика” используется приложение Schemes, предназначенное для формирования и развития алгоритмического мышления, а также для решения не сложных практических задач. Приложению требуется обновление, так как текущая версия устарела и имеет некоторые недостатки, делающие обучения не комфортным, что подтверждает анализ социологического исследования о необходимости обновления приложения. Для решения этой проблемы представлены результаты проектирования обновлённой системы Schemes. В ходе работы была изучена предметная область и аналогичные системы, определены требования к информационной системе, выбрано программное обеспечение и язык программирования для дальнейшей реализации системы. Для проектирования системы были построены UML диаграммы и ER модель. А также был создан макет пользовательского интерфейса.

Ключевые слова: алгоритмизация, алгоритм, блок-схема, информационная система, Schemes, проектирование.

Введение

В данный момент для обучения студентов ПГНИУ блоку алгоритмизации в рамках дисциплины «Информатика» используется приложение Schemes. Schemes представляет собой интерпретатор алгоритмов, представленных в виде блок-схем. Последняя версия данного приложения выпускалось в 2009 г. и более не обновлялась. Поэтому появилась мысль об обновлении данной программы. Актуальность данной работы подтверждают результаты опроса, проведённого среди студентов и преподавателей, работавших с Schemes. Результаты представлены в виде диаграмм на рис. 1–3.

Главной задачей ИС (информационной системы) Schemes помощь в обучение студентов алгоритмизации в рамках дисциплины информатики. Студенты выстраивают алгоритмы в виде блок-схем для решения заранее подготовленных задач и получают оценки.

Алгоритм должен удовлетворять требованиям, которые представлены в виде следующих свойств: дискретность, понятность, определённости, результативность, массовость [1, 2].

По типу используемого вычислительного процесса различают линейные (прямые), разветвляющиеся и циклические алгоритмы [3].

Считаете ли вы, что приложение устарело? – количество

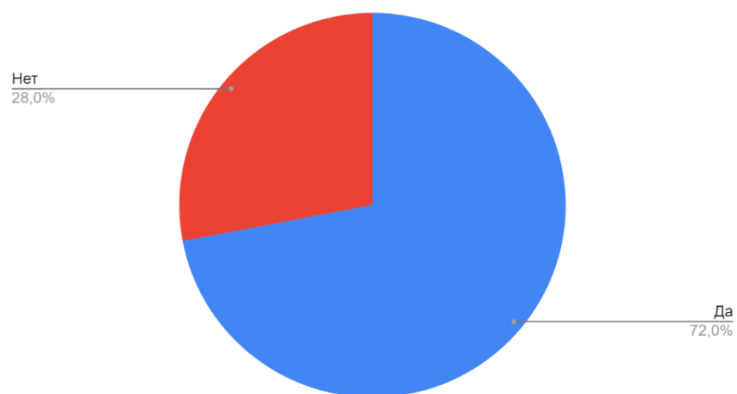


Рис. 1. Диаграмма ответов на первый вопрос

Удобно ли навигации по приложению? – количество

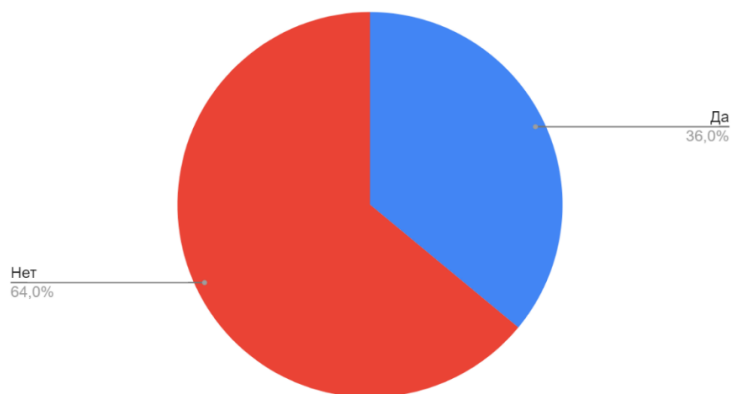


Рис. 2. Диаграмма ответов на второй вопрос

Интерфейс интуитивно понятный? – количество

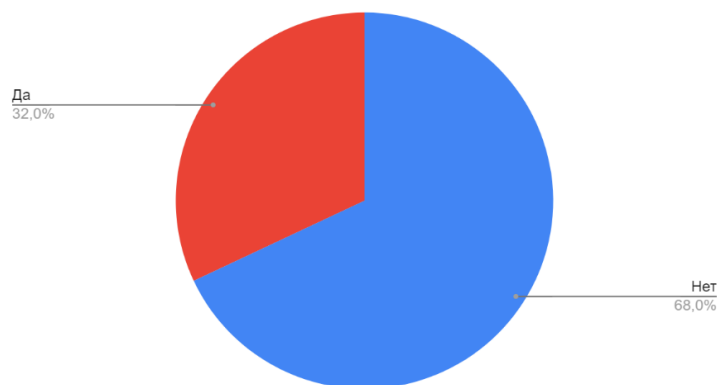
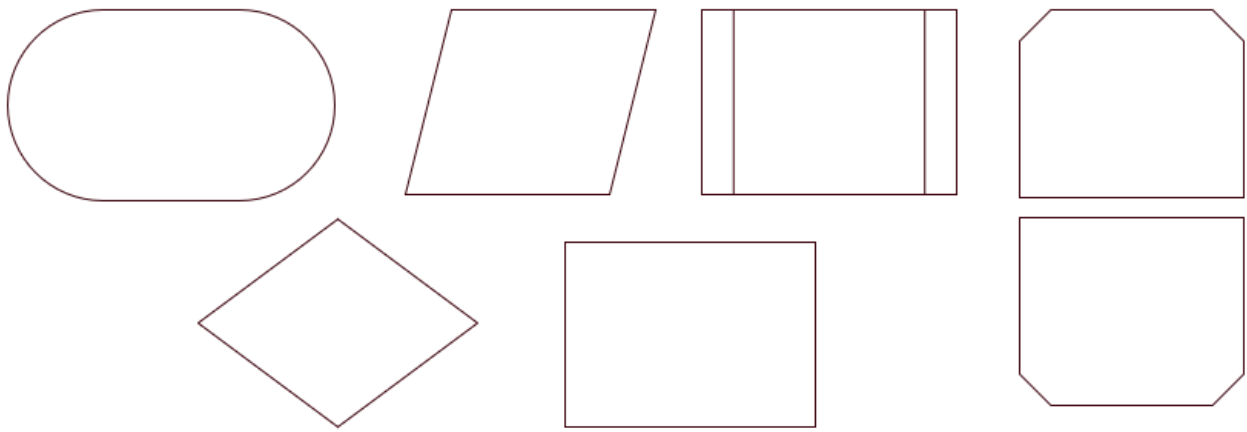


Рис. 3. Диаграмма ответов на третий вопрос

Для решения задач в Schemes студенты строят блок-схемы, то есть используют графическое описание.

На рисунке 4 представлены основные условные обозначения для представления алгоритма в графической записи, которые используются в приложении Schemes.



Рису. 4. Блоки используемы в приложение Schemes

Стандарт ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения» определяет блоки, используемые для создания блок-схем [4].

Постановка задачи

Необходимо спроектировать обновлённую версию приложения Schemes. Обновлённая версия должна удовлетворять следующим требованиям:

- ИС должна имеет следующие функции: разработка схем алгоритмов, откладка схем алгоритмов, контрольное тестирование разработанной схемы алгоритма с целью оценивания деятельности обучаемого;
- ИС должна контролировать продолжительность занятия, этапов его дидактической;
- структуры, всех видов работы и этапов решения задач;
- ИС должна содержать контекстно-зависимую справочную информацию, которая может быть использована и при ознакомлении с изучаемым материалом, и в качестве краткого конспекта;
- ИС должна проверять правильность выполнения заданий и формировать итоговую оценку;
- ИС должна иметь в базе данных набор заданий, который может быть пополнен или изменён администратором;
- Система должна случайным образом выдавать задания студентом из своей базы данных.

Сравнение аналогичных существующих систем

Рассмотрев возможности Lucidchart [5], Microsoft Visio [6], Gliffy [7] с Schemes, был сделан вывод, что современные существующие приложения не подходят для обучения алго-

ритмизации. Они, безусловно, предлагают обширные возможности для создания блок-схем, но не предоставляют при этом ни учебных материалов, ни возможностей выполнять алгоритмы и смотреть на результаты их выполнения.

Выбор программного обеспечения и языка программирования

Сравнив приложения для создания UML-диаграмм, было выбрано приложение Diagrams.net [8].

Для создания макета пользовательского интерфейса был выбран MarvelApp [9].

В качестве языка программирования, на котором будет разрабатываться Schemes, был выбран C# [10].

В качестве среды разработки была выбрана Microsoft Visual Studio [11].

Для управления базой данных был выбран PostgreSQL [12].

Проектирования информационной системы

На диаграмме прецедентов (см. рис. 5) представлены возможности студента и администратора. Администратором в данной системе является преподаватель.

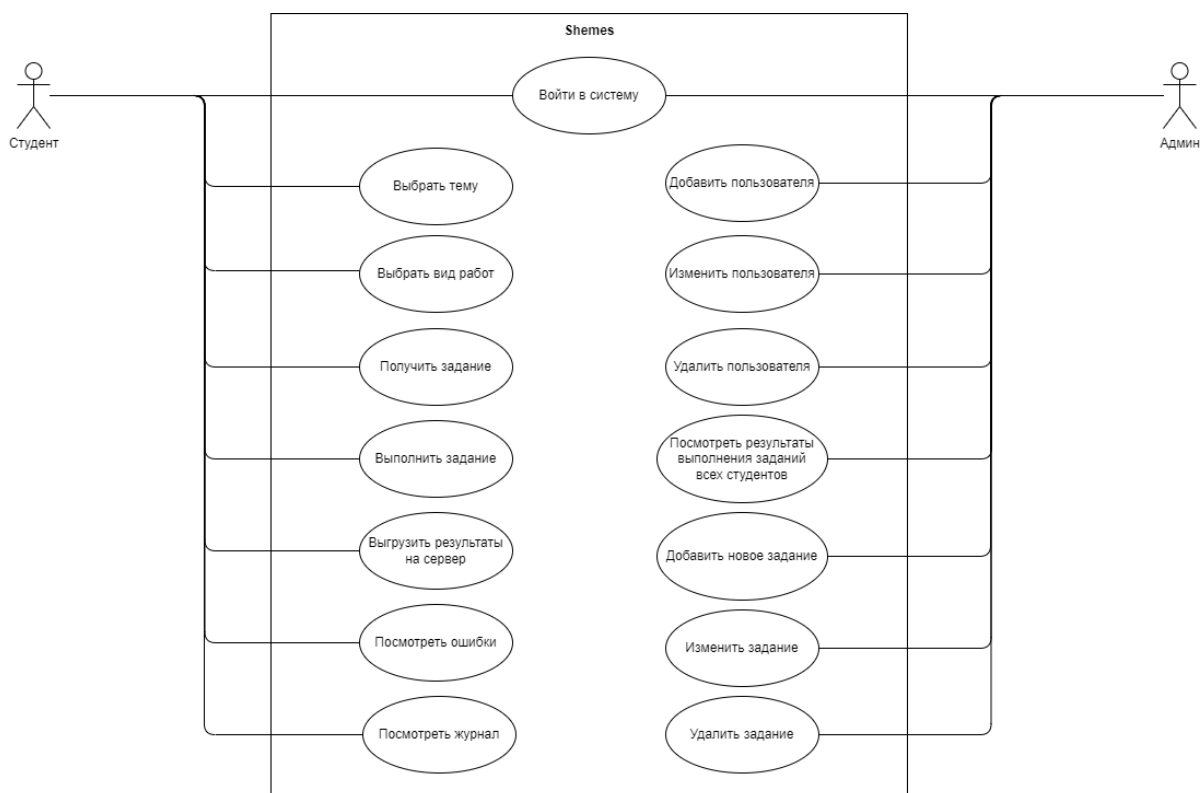


Рис. 5. Диаграмма прецедентов

Диаграмма состояний, изображённая на рис. 6, демонстрирует динамическую работу системы в зависимости от действий пользователя.

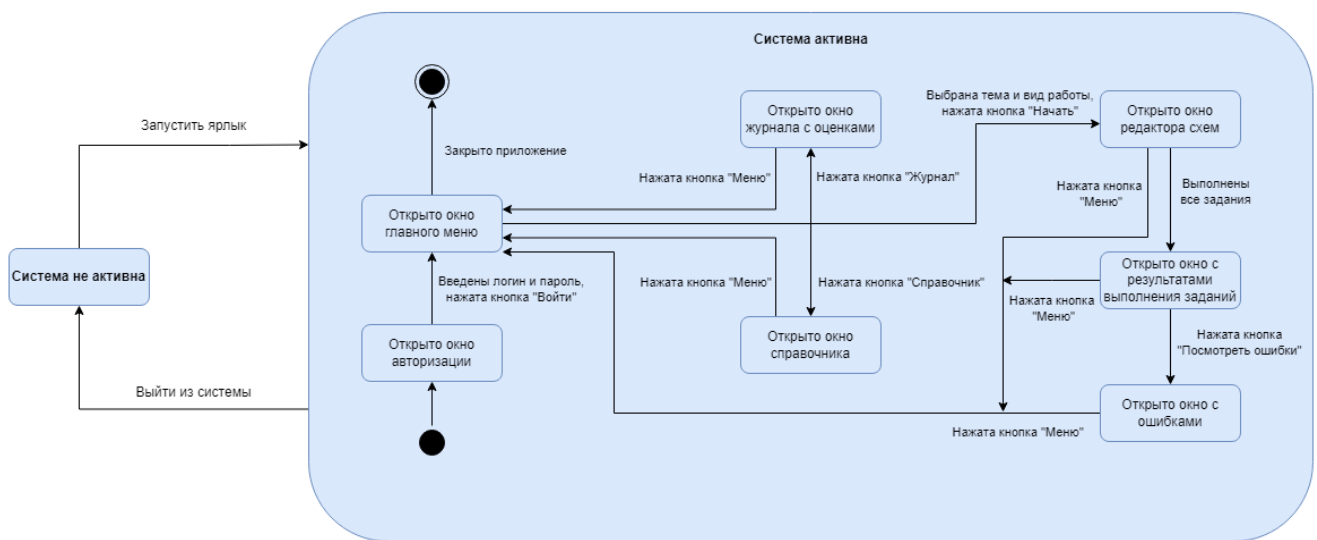


Рис. 6. Диаграмма состояний

На рисунке 7 изображена схема базы данных системы. В ней будут храниться данные о заданиях, студентах и их результатах

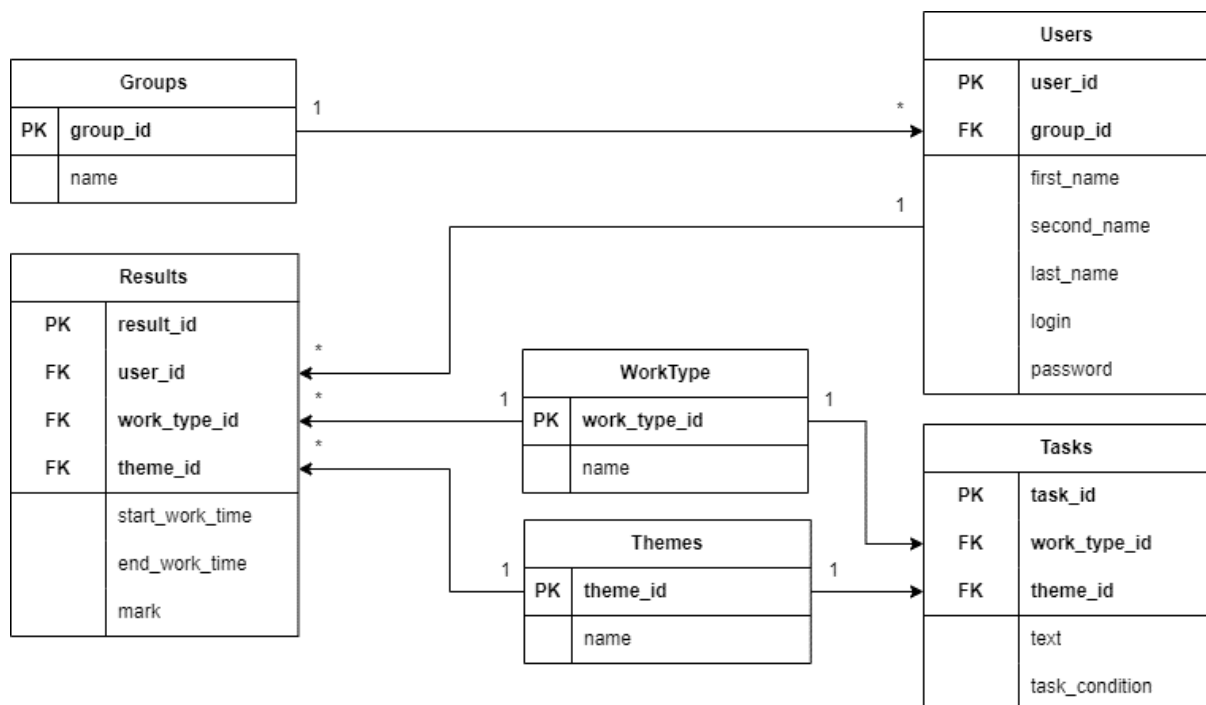


Рис. 7. Схема базы данных

В ходе работы был спроектирован макет пользовательского интерфейса.

Главное меню, в которое студент попадает после авторизации, представлено на рис. 8. Там он может выбирать тему занятия и вид работы, и приступить к выполнению заданий. Также студент в меню может получить справку по выбранной теме или виду работы.

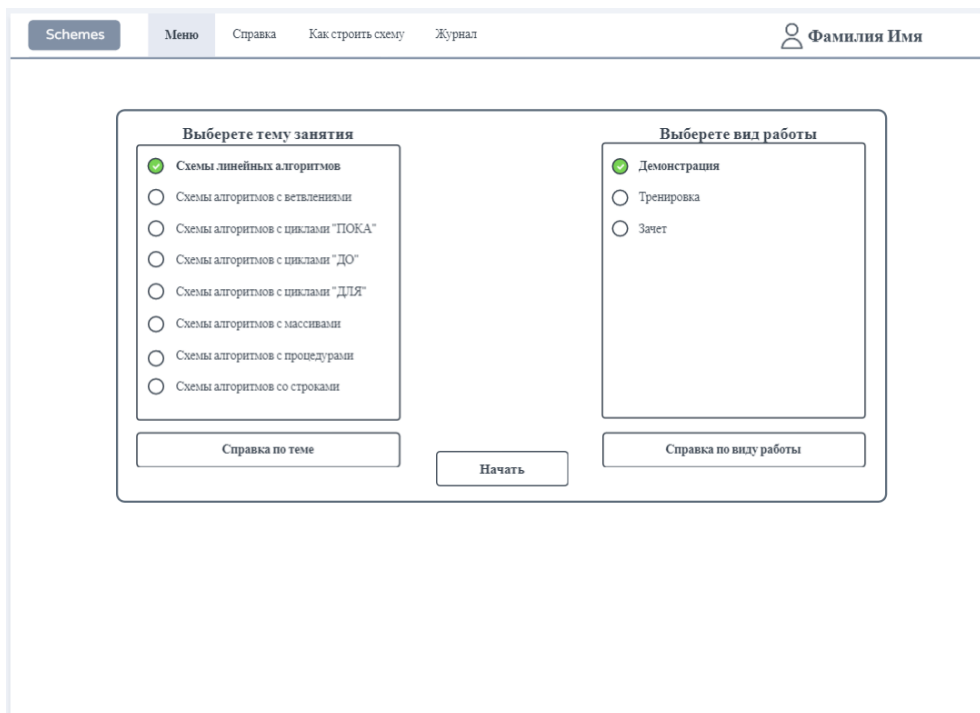


Рис. 8. Главное меню

Само выполнения заданий происходит в редакторе блок-схем, представленном на рис. 9. Сверху располагается номер задачи и её текст. Слева находятся иконки блоков. С помощью переноса данных иконок в рабочую область по середине, происходит построение блок-схемы. Справа располагаются кнопки для запуска алгоритма и отладочная информация.

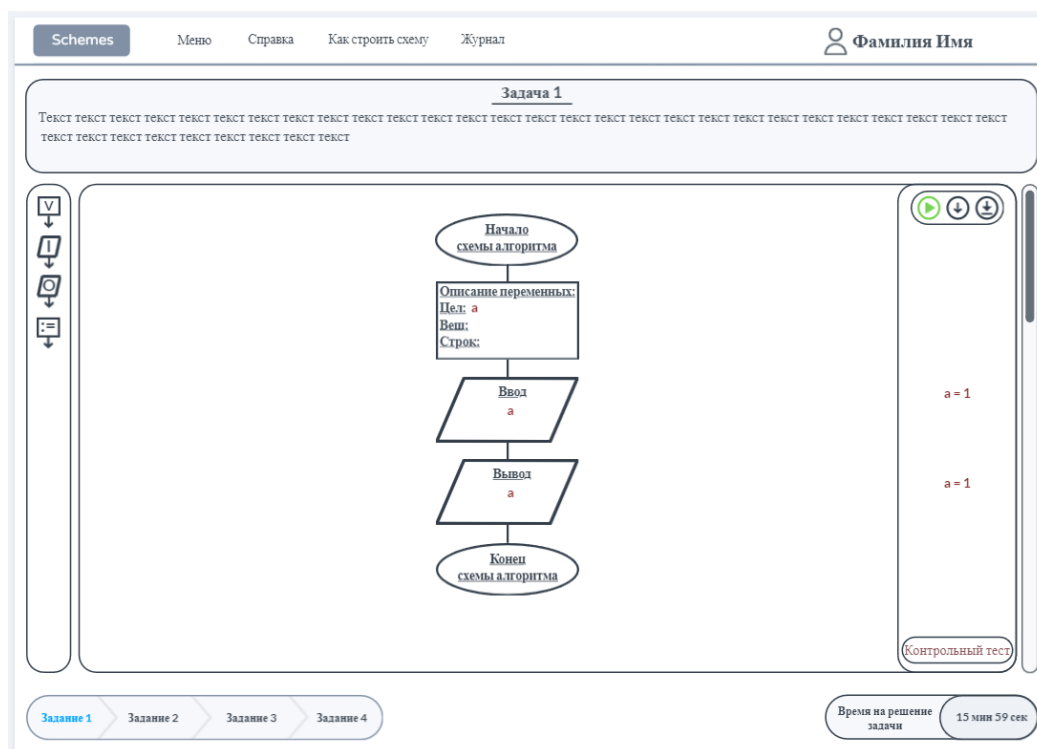


Рис. 9. Редактор блок-схем

В окне, представленном на рис. 10, пользователь, может получить справочную информацию о самом приложении или о темах занятий.

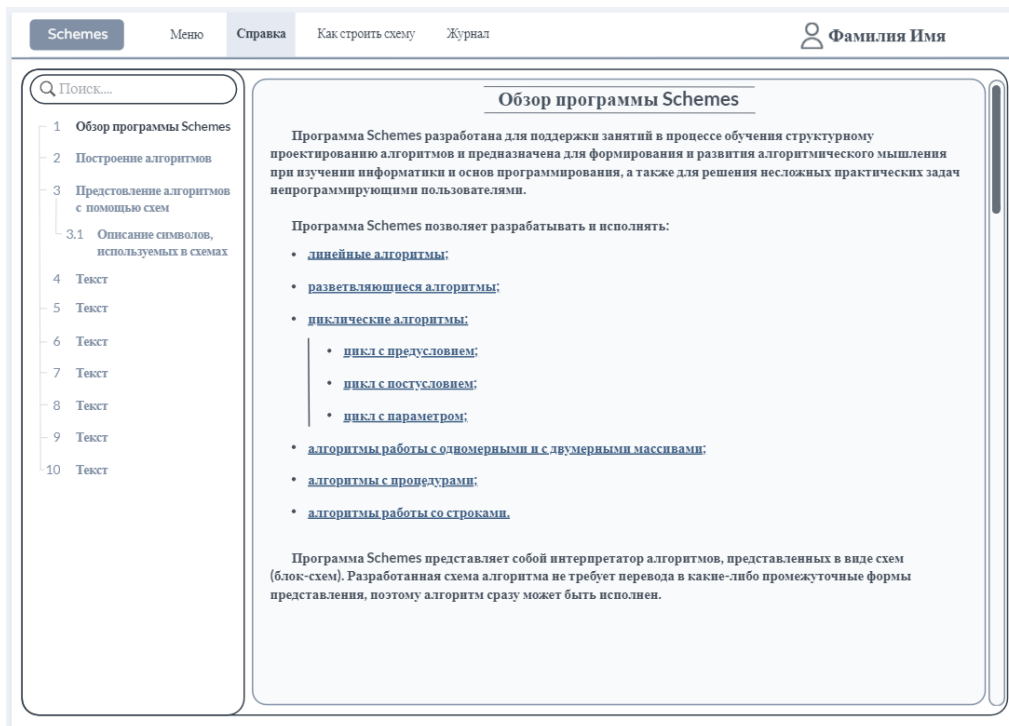


Рис. 10. Окно справки

В журнале (см. рис. 11) студент может посмотреть, какие работы он выполнял и какие у него результаты. А также здесь студент может выгрузить свои оценки на сервер.

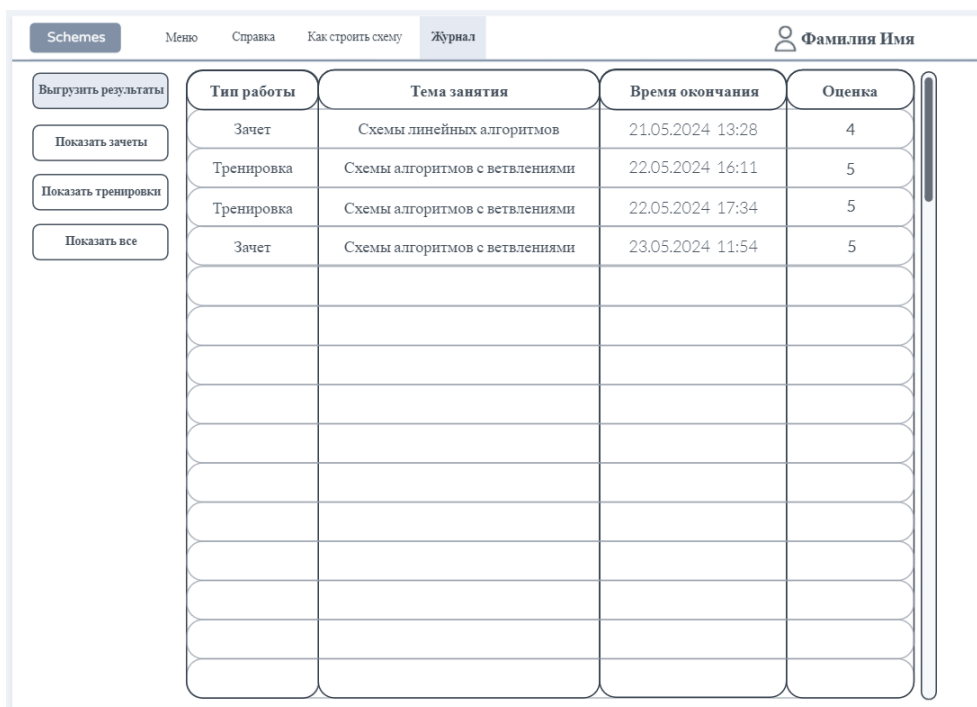


Рис. 11. Журнал студента

Администратор же в своём журнале (см. рис. 12) видит более детальную информацию о студентах и их результатах. А также в данном окне присутствует фильтр для быстрого поиска необходимой информации.

Рис. 12. Журнал администратора

В результате работы были рассмотрены ключевые аспекты предметной области, было выбрано программное обеспечение и язык программирования для проектирования и дальнейшей разработки системы, а также система была спроектирована: построены UML диаграммы прецедентов и развёртывания, ER диаграмма и создан макет пользовательского интерфейса.

4. ГОСТ 19.701–90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (действует с 01.01.1992).
5. Lucidchart.com. URL: <http://lucidchart.com/> (дата обращения 16.05.2024).
6. Microsoft Visio Flowchart Software. URL: <https://www.microsoft.com/ru-ru/microsoft-365/visio/flowchart-software> (дата обращения 16.05.2024).
7. Gliffy. URL: <https://www.gliffy.com/> (дата обращения 16.05.2024).
8. Diagrams.net. URL: <https://www.drawio.com/> (дата обращения 16.05.2024).
9. Marvelapp.com. URL: <https://marvelapp.com/> (дата обращения 16.05.2024).
10. Microsoft C#. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения 16.05.2024).
11. Microsoft Visual Studio. URL: <https://visualstudio.microsoft.com/ru/> (дата обращения 16.05.2024).
12. PostgreSQL.org. URL: <https://www.postgresql.org/> (дата обращения 16.05.2024).

References

1. Informatika: ucheb. posobie dlja stud. ped. vuzov / A. V. Mogilev, N. I. Pak, E. K. Henner; pod red. E. K. Hennera. 8-e izd., ster. M.: Izdatel'skij centr "Akademija", 2012. 848 p. (In Russ.)
2. Osnovy algoritmizacii i programmirovanija: uchebnik dlja srednego professional'nogo obrazovanija / V. V. Trofimov, T. A. Pavlovskaja; pod red. V. V. Trofimova. 4-e izd. M.: Izdatel'stvo Jurajt, 2024. 119 p. (In Russ.)
3. Osnovy algoritmizacii i programmirovanija: uchebno-metod. posobie dlja studentov vuzov spec. "Informacionnye sistemy i tehnologii izdatel'sko-poligraficheskij kompleks" / N. V. Pacej. Minsk: BGTU, 2010. 288 p. (In Russ.)
4. GOST 19.701–90 Edinaja sistema programnoj dokumentacii. Shemy algoritmov, programm, dannyh i sistem. Uslovnye oboznachenija i pravila vypolnenija (valid from 01.01.1992). (In Russ.)
5. Lucidchart.com. URL: <http://lucidchart.com/> (date of access: 05/16/2024).
6. Microsoft Visio Flowchart Software. URL: <https://www.microsoft.com/ru-ru/microsoft-365/visio/flowchart-software> (date of access: 05/16/2024).
7. Gliffy. URL: <https://www.gliffy.com/> (date of access: 05/16/2024).
8. Diagrams.net. URL: <https://www.drawio.com/> (date of access: 05/16/2024).
9. Marvelapp.com. URL: <https://marvelapp.com/> (date of access: 05/16/2024).
10. Microsoft C#. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (date of access: 05/16/2024).

11. Microsoft Visual Studio. URL: <https://visualstudio.microsoft.com/ru/> (date of access: 05/16/2024).
12. PostgreSQL.org. URL: <https://www.postgresql.org/> (date of accessed: 05/16/2024).

DESIGNING AN INFORMATION SYSTEM FOR TEACHING THE OF ALGORITHMIZATION BLOCK IN THE DISCIPLINE "COMPUTER SCIENCE"

Sharov Roman V, Rakina Valeria D

Perm State University, 15, Bukireva st., Perm, 614068, Russia, roma752275@mail.ru

In the process of teaching students at Perm State University, the block of algorithmization is used within the framework of the discipline "Computer Science". The Schemes application is designed to help students develop their algorithmic thinking and solve simple practical problems. The application needs to be updated, as the current version is outdated and has some disadvantages, making learning uncomfortable. This confirms the analysis of a sociological study that shows the need for an update. To solve this problem, the results of the design for the updated Schemes system are presented. In the process of the work, the subject area and similar systems were studied, the requirements for the information system were determined, software and programming language were selected for further implementation of the system. UML diagrams and an ER model were used to design the system. A user interface design has also been developed.

Keywords: algorithmization, algorithm, flowchart, information system, Schemes, design.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОНТРОЛЯ И ПРОГНОЗИРОВАНИЯ ФИНАНСОВ

Язык Мария Денисовна, Аверин Сергей Игоревич

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, yazik.mariya@mail.ru

Финансы играют одну из ключевых ролей в жизни любого человека. Для автоматизации и удобства контроля своего бюджета существуют PFM (Personal Finance Management) приложения - сервисы для управления личными финансами. Существующие методы финансового контроля часто оказываются недостаточно удобными в использовании, что приводит к потере времени и ресурсов. Для решения этой проблемы представлены результаты проектирования информационной системы контроля и прогнозирования финансов «Мои финансы». В ходе работы была изучена предметная область и аналогичные системы, выбрано программное обеспечение и язык программирования для проектирования и дальнейшей реализации системы. При моделировании системы были построены UML диаграммы и ER модель. А также был создан макет пользовательского интерфейса.

Ключевые слова: финансы, прогнозирование, информационная система, проектирование.

Эффективное управление своим бюджетом позволяет быстрее и грамотнее накопить на поставленные цели и имеющиеся желания, отследить ненужные траты. Однако существующие методы финансового контроля часто оказываются недостаточно удобными, что приводит к потере времени и ресурсов. Одним из способов решения этой проблемы является проектирование информационных систем – PFM-приложений (англ. Personal Finance Management) и внедрение их в нашу жизнь, что обеспечивает автоматизацию процессов управления финансами и позволяет получать оперативную информацию о бюджете. В нашем динамичном обществе люди все чаще прибегают к использованию подобных приложений для установки финансовых целей, планирования событий и контроля своего бюджета. Исходя из этого можем сделать вывод об актуальности данной работы.

Главной задачей ИС (информационной системы) помочь пользователям в контроле собственного бюджета. Пользователи отслеживают доходы и расходы, видят статистику и аналитику в виде диаграмм, просматривают будущие расходы в зависимости от запланированных мероприятий.

Рассмотрев функционал таких аналогичных приложений, как Дзен-мани [1], Coin-Keeper [2], и 1Money [3] был сделан вывод, что современные существующие приложения позволяют отслеживать расходы и доходы по категориям, ведут учет всех имеющихся

средств, имеют возможность подключить банк и поддерживают основной функционал онлайн-банков. Но ни одно из них не предоставляет пользователю информацию о предстоящих тратах.

Следующим этапом работы над системой стала разработка базы данных. На рисунке 1 представлена схема структуры базы данных.

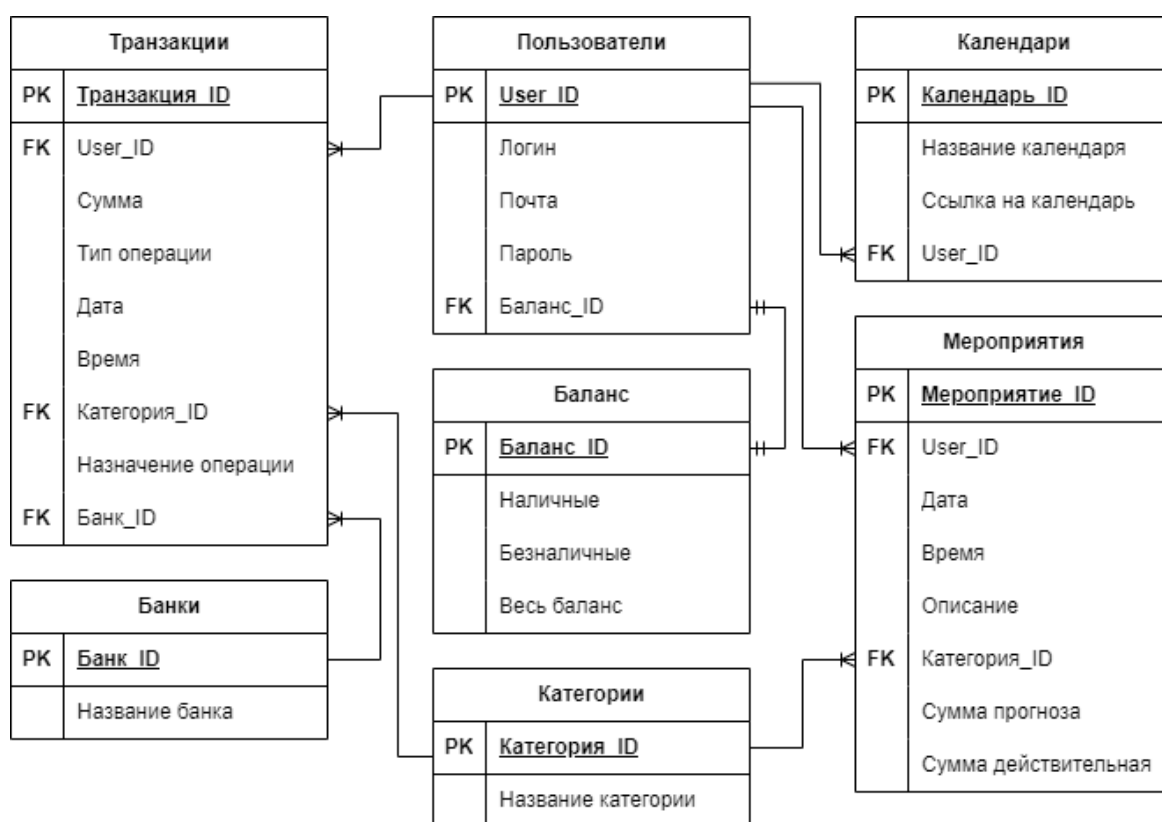


Рис. 1. Схема структуры базы данных

На диаграмме прецедентов (см. рис. 2) представлено взаимодействие пользователя и администратора с системой.

При первом открытии приложения, пользователя встречает экран регистрации, где необходимо ввести логин, пароль, почту и подтвердить пароль. Если пользователь уже зарегистрирован в системе, он может перейти на экран входа. После успешного прохождения регистрации, появляется доступ к приложению.

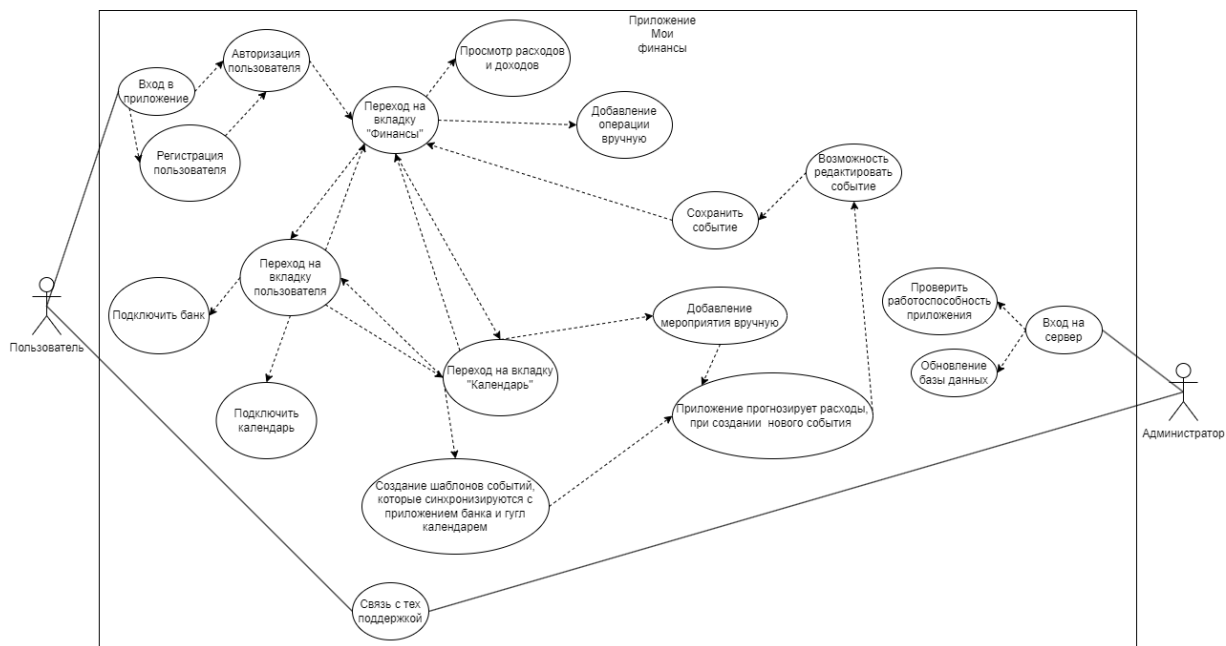


Рис. 2. Диаграмма прецедентов

На панели внизу экрана расположены три иконки: «Финансы», «Календарь», «Пользователь», нажатие на которые служит переходом между экранами.

Сверху на экране «Финансы» расположена сумма расходов или доходов за данный месяц, в зависимости от выбранного режима отображения. Рядом с надписью «Итого» есть стрелочка, при нажатии на которую выходит окно, на котором можно посмотреть все имеющиеся у пользователя средства на данный момент. Ниже расположена диаграмма расходов/доходов по категориям. При нажатии на кнопку плюс, можно самостоятельно добавить транзакцию. Под диаграммой указаны суммы всех трат по категориям, нажав на категорию можно увидеть список всех операций по ней. Снизу расположена кнопка, нажав на которую выводится список всех операций вне зависимости от категории (см. рис. 3).

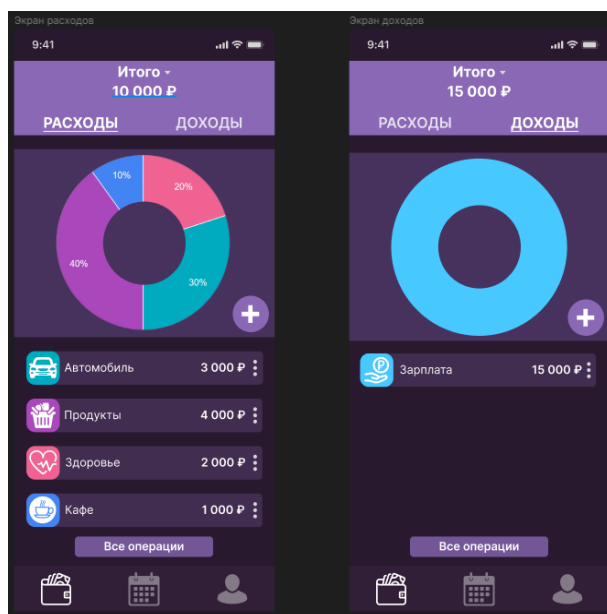


Рис. 3. Окна доходов и расходов

На экране «Календарь» можно нажать на дату и увидеть все запланированные или прошедшие мероприятия в этот день, если мероприятие прошло, то оно тускнеет. На экране «Пользователь» можно перейти во вкладки «Мои данные», где можно сменить пароль или почту, «Добавить банк» для синхронизации с банком, «Добавить календарь» для синхронизации с календарем, «Настройки уведомлений», а также «Связаться с поддержкой» (см. рис. 4).

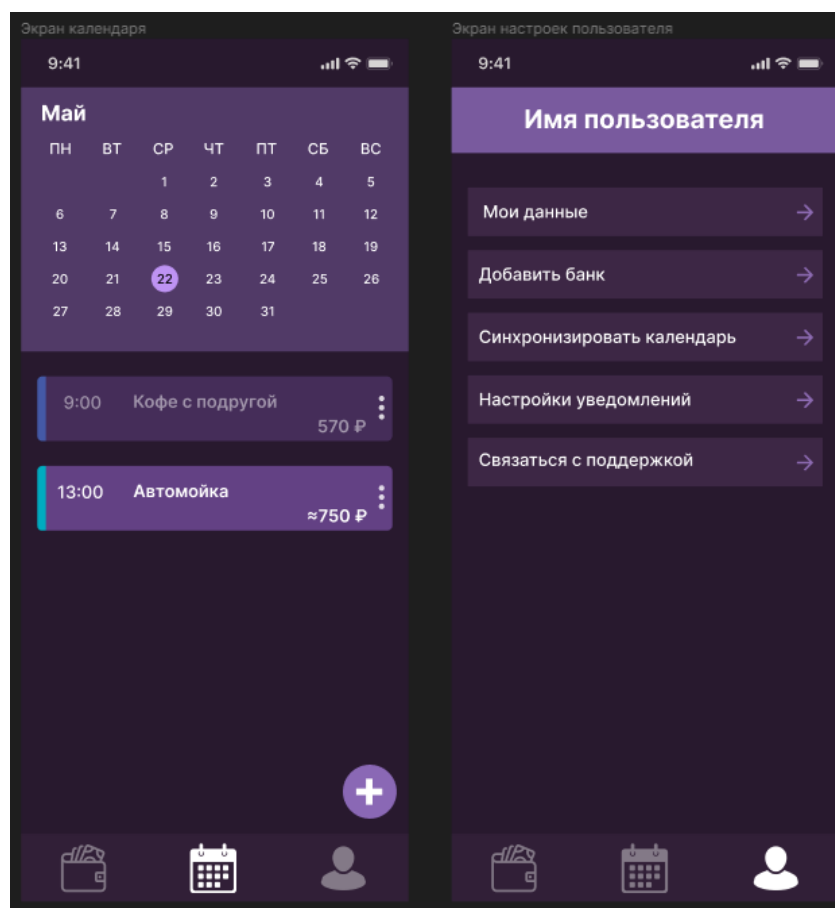


Рис. 4. Окна календаря и пользователя

В результате работы была изучена предметная область, были выбраны средства проектирования и разработки системы, а также спроектирована сама система: построена диаграмма прецедентов для лучшего понимания работы системы, ER-диаграмма и создан прототип пользовательского интерфейса.

Библиографический список

1. Дзенмани [Электронный ресурс]. URL: <https://zenmoney.ru/> (дата обращения 02.05.2024).
2. CoinKeeper [Электронный ресурс]. URL: <https://about.coinkeeper.me/> (дата обращения 02.05.2024).
3. 1Money [Электронный ресурс]. URL: <https://1moneyapp.com/> (дата обращения 02.05.2024).

4. Diagrams.net [Электронный ресурс]. URL: <https://app.diagrams.net/> (дата обращения 15.05.2024).
5. Figma [Электронный ресурс]. URL: <https://www.figma.com> (дата обращения: 15.05.2023).
6. Flutter [Электронный ресурс]. URL: <https://flutter.dev/> (дата обращения 15.05.2024).
7. PostgreSQL [Электронный ресурс]. URL: <https://www.postgresql.org/> (дата обращения: 15.05.2023).

References

1. Dzenmani [Electronic resource]. URL: <https://zenmoney.ru/> (date of access: 05/02/2024).
2. CoinKeeper [Electronic resource]. URL: <https://about.coinkeeper.me/> (date of access: 05/02/2024).
3. 1Money [Electronic resource]. URL: <https://1moneyapp.com/> (accessed on 02.05.2024).
4. Diagrams.net [Electronic resource]. URL: <https://app.diagrams.net/> (date of access: 05/15/2024).
5. Figma [Electronic resource]. URL: <https://www.figma.com> (date of access: 05/15/2023).
6. Flutter [Electronic resource]. URL: <https://flutter.dev/> (date of access: 05/15/2024).
7. PostgreSQL [Electronic resource]. URL: <https://www.postgresql.org/> (date of access: 05/15/2023).

DESIGNING AN INFORMATION SYSTEM FOR FINANCIAL MONITORING AND FORECASTING

Iazik Maria D, Averin Sergei I

Perm State University, 15, Bukireva st., Perm, 614068, Russia, yazik.mariya@mail.ru

Finance plays one of the key roles in the life of any individual. For the automation and convenience of budget control, there are Personal Finance Management (PFM) applications – services for managing personal finances. Existing methods of financial control are often not convenient to use, which leads to time and resource losses. To address this issue, the results of designing an information system for financial control and forecasting called "My Finance" are presented. The subject area and similar systems were studied during the work, software and programming language were selected for designing and further implementation of the system. UML diagrams and an ER model were constructed during the system modeling process. Additionally, a user interface mockup was created.

Keywords: finance, forecasting, information system, design.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОТСЛЕЖИВАНИЯ БРАЧНОГО ЛЁТА МУРАВЬЕВ*

Ясырев Михаил Евгеньевич, Ракина Валерия Денисовна

Пермский государственный национальный исследовательский университет,
614068, Россия, г. Пермь, ул. Букирева, 15, yasyrevmikhail@gmail.com

В статье представлен процесс проектирования информационной системы для отслеживания брачного лёта муравьев. Обоснована актуальность рассматриваемой темы, проведен анализ существующих систем с аналогичным функционалом, выделены их основные преимущества и недостатки, определены дополнительные требования. На основе анализа предметной области выделены основные характеристики. Установлены функциональные требования к системе. Произведен анализ средств проектирования и разработки, выбрана наиболее сбалансированная по ряду параметров. Спроектирована модель информационной системы с использованием ряда UML-диаграмм, таких как диаграмма прецедентов, последовательности, классов. Также была построена концептуальная модель данных, спроектирован и приведен макет для пользовательского интерфейса. В заключении приведены выводы о проделанной работе.

Ключевые слова: брачный лёт муравьев, отслеживание брачного лёта муравьев, проектирование информационной системы.

Введение

Мирмикиперы – это сообщество энтузиастов, увлеченных изучением мира муравьев. Они не только наблюдают за жизнью этих насекомых, но и активно участвуют в сборе данных о различных аспектах их поведения. Одним из ключевых моментов в жизненном цикле муравьев является их брачный лёт, который становится объектом особого внимания для мирмикиперов, стремящихся создать собственные колонии.

Целью данной работы является проектирование информационной системы, которая поможет мирмикиперам определить время и место для поиска маток муравьёв для создания собственных колоний. Объектом исследования являются характеристики брачного лёта муравьёв, включая время начала, продолжительность, местоположение и погодные условия. Предметом исследования является автоматизация процесса мониторинга брачного лёта муравьёв с использованием информационных технологий.

В настоящее время существует необходимость в создании эффективного инструмента, который позволит мирмикиперам оперативно получать информацию о брачном лёте муравьёв в их регионе. Предлагаемая информационная система будет предоставлять возможность

© Ясырев М. Е., Ракина В. Д., 2024

* 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).

пользователям добавлять записи о наблюдениях брачного лёта, указывать детали о времени, местоположении и других характеристиках события. Эти записи будут модерироваться и отображаться на интерактивной карте с возможностью фильтрации и поиска по различным параметрам.

Анализ предметной области

Брачный лёт муравьёв – это ключевой момент в жизненном цикле данных насекомых, когда особи определенного пола вылетают из муравейника для размножения и создания новых колоний. Успешная поимка маток требует точного знания времени и места вылета муравьёв, что в свою очередь требует системы наблюдения и учёта.

Цель настоящей работы состоит в проектировании информационной системы, предназначенной для мирмикперов, позволяющей отслеживать брачный лёт муравьёв и определять оптимальные места для их поимки с целью основания новых колоний.

Для определения будущих характеристик и возможностей проектируемой системы был проведен анализ наиболее популярных систем.

AntFlights.com это специализированный веб-сайт, предназначенный для глобальной регистрации и отслеживания брачного лёта муравьев [1]. Из достоинств: быстрая регистрация через социальные сети, генерация графиков статистики, удобный вид карточек наблюдений, вывод карты с отметками о наблюдениях. Недостатками же можно назвать спорные решения в форме данных о наблюдении, отсутствие необходимых фильтров, отсутствие сводной таблицы периодов лета видов.

Antclub.ru это всеобъемлющий веб-сайт, посвященный муравьям и мирмекологии, предлагающий широкий спектр ресурсов для энтузиастов [2]. Аналогом проектируемой системы является раздел «Таблица лёта», в котором в виде таблицы представлена информация о некоторых видах муравьев. Для каждого из них отмечены диапазоны брачного лёта, исходя из литературы, и отдельно отмечены диапазоны из наблюдений пользователей, опубликованных в отдельном разделе сайта. К достоинствам системы можно отнести представление совместных наблюдений пользователей и научных записей в виде единой таблицы, отметки видов, лёт которых идет сейчас и пометки у видов, лёт которых ожидается в ближайшее время. Недостатками же можно назвать ограниченный список видов, отсутствие возможности добавить к наблюдениям фотографии, отсутствие модерации записей, недружелюбный интерфейс.

Сайт muravdom.ru представляет собой интернет-магазин, специализирующийся на продаже муравьиных ферм, муравьёв и сопутствующих товаров для их содержания [3]. Аналогом является раздел «Таблица лёта муравьев по России». К достоинствам системы можно

отнести представление совместных наблюдений пользователей и научных записей в виде единой таблицы, приятный дизайн, удобное выделение строк таблицы. Недостатками же можно назвать ограниченный список видов, непонятный способ добавления наблюдений, отсутствие описания условий лёта.

Недостатком всех систем является отсутствие возможности подписаться на уведомления о начале лёта выбранных пользователем видов.

Выбор средств проектирования

CASE-средства – это методы и технологии, которые позволяют проектировать различные информационные системы (в частности, базы данных) и автоматизировать их создание [4]. Они будут оценены по следующим критериям: наличие бесплатной или студенческой версии, поддержка стандарта UML.

Сравним 5 наиболее распространённых из них:

1. StarUML. Инструмент для моделирования, поддерживающий стандарты UML, известен своей расширяемостью и поддержкой различных UML-диаграмм, в том числе диаграммы классов, последовательности и состояния. Распространяется на коммерческой основе, но доступна бесплатная версия для некоммерческого использования [5].

2. Rational Rose. Инструмент для объектно-ориентированного анализа и проектирования, разработанный компанией IBM. Предоставляет средства для создания UML-диаграмм и поддержки разработки ПО на основе моделей. Распространяется на коммерческой основе [6].

3. Astah Professional. Инструмент для проектирования, позволяет различные диаграммы, в том числе UML, ER, блок-схемы и схемы потока данных. Продукт распространяется на коммерческой основе, доступна ознакомительная версия, доступна пробная версия [7].

4. MS (Microsoft Office) Visio. Коммерческое программное обеспечение для создания диаграмм и векторных графиков, входящее в состав Microsoft Office. Доступна по подписке через Microsoft 365 или как самостоятельный продукт. Предлагает мощный набор инструментов для визуализации данных и проектирования систем, что делает его популярным в бизнесе и инженерных средах [8].

5. Silverrun. Семейство инструментов для моделирования данных и бизнес-процессов, включающее такие продукты, как Silverrun ModelSphere. Программа распространяется на коммерческой основе, на выбор предлагается несколько версий и лицензии для удовлетворения потребностей пользователей [9].

Поддержку стандарта UML имеют все рассмотренные продукты, а постоянную бесплатную версию только StarUML, поэтому для проектирования информационной системы был выбран именно этот инструмент.

Выбор средств разработки

Информационную систему «Отслеживание брачного лёта муравьев» можно представить в виде серверного приложения, состоящего из БД, API и агента, управляемого через сайт, который будет являться интерфейсом системы. Такая архитектура – один из вариантов реализации подобных систем.

Выбор СУБД

Для функционирования система необходима база данных. Были рассмотрены СУБД Oracle, MySQL, MS SQL Server и PostgreSQL.

В таблице 1 представлены значения критериев для выбора СУБД.

Таблица 1

Сравнение СУБД

СУБД	Поддержка работы с веб-сервером	Свободная форма распространения	Наличие полной документации
MySQL	+	+	+
Oracle	+	-	+
MS SQL Server	+	-	+
PostgreSQL	+	+	-

Для информационной системы «Отслеживание брачного лёта муравьев» СУБД должна иметь ряд черт, а именно: работать с веб-сервером, быть свободно распространяемой, иметь хорошую документацию. Из перечисленных СУБД была выбрана MySQL, потому как она удовлетворяет всем выбранным критериям оценки.

Выбор языка для разработки API

При выборе языка программирования для разработки API информационной системы важно учитывать множество факторов, таких как производительность, удобство использования, поддержка сообщества, наличие библиотек и инструментов, а также соответствие требованиям проекта.

Для информационной системы «Отслеживание брачного лёта муравьев» язык для API должен иметь ряд черт, а именно: производительность, удобство использования и скорость разработки, поддержка сообщества и экосистема, масштабируемость и управляемость, инструменты и библиотеки.

Для сравнения выбраны языки JavaScript (Node.js), Python (Django/Flask), Java (Spring Boot) и Ruby (Ruby on Rails) [10, 11, 12, 13]. По выбранным критериям JavaScript в среде Node.js является оптимальным выбором для разработки API информационной системы. Node.js предлагает наибольший баланс преимуществ для современных веб-приложений.

Выбор инструмента для фронтенд разработки

При выборе библиотеки для разработки фронтенда важно учитывать множество факторов, таких как производительность, удобство использования, поддержка сообщества, наличие компонентов и инструментов, а также соответствие требованиям проекта.

Для информационной системы «Отслеживание брачного лёта муравьев» инструмент для фронтенд разработки должен иметь ряд черт, а именно: производительность, компонентный подход, удобство использования, поддержка сообщества.

Для сравнения выбраны React, Angular и Vue [14, 15, 16]. По выбранным критериям React является оптимальным выбором для разработки фронтенда. Сравнивая React с альтернативами, можно увидеть, что он остается одним из наиболее популярных и предпочтительных выборов для разработки фронтенда благодаря своей производительности, удобству использования, поддержке сообщества и обширной экосистеме.

Проектирование информационной системы

Моделирование поведения ИС

Для начала проектирования информационной системы необходимо выделить основные типы пользователей и определить возможности для каждого из полученных типов. В системе «Отслеживание брачного лёта муравьев» будет 3 типа пользователей: гость, зарегистрированный пользователь и модератор. Все роли должны иметь возможность просматривать опубликованные наблюдения. Гость может зарегистрироваться. Зарегистрированный пользователь и модератор дополнительно могут изменять свои данные для авторизации, добавлять наблюдения и подписываться на уведомления о лёте. Модератор дополнительно может отклонить/опубликовать наблюдения, заблокировать пользователя.

Чтобы представить действия пользователя во времени можно воспользоваться диаграммой последовательностей. На рисунке 2 представлена последовательность действий пользователя при добавлении наблюдений в системе «Отслеживание брачного лёта муравьев».

Чтобы лучше понять порядок действий, лежащий за некоторыми прецедентами, составим для них диаграммы деятельности. Если данные были успешно сохранены в базе данных, добавленные наблюдения переходят на одобрение модератором. На рисунке 3 представлена диаграмма деятельности, поясняющая прецедент «Модерация наблюдений». На ней отображены взаимодействия между модератором, интерфейсом, API и базой данных.

Моделирование структуры ИС

Моделирование структуры информационной системы подразумевает описание составных частей системы и отношения между ними. Язык UML применяется как объектно-

ориентированный подход к моделированию систем. Объекты системы можно представить в виде классов. Для отображения классов и связей между ними в UML предусмотрена диаграмма классов. (см. рис. 4).

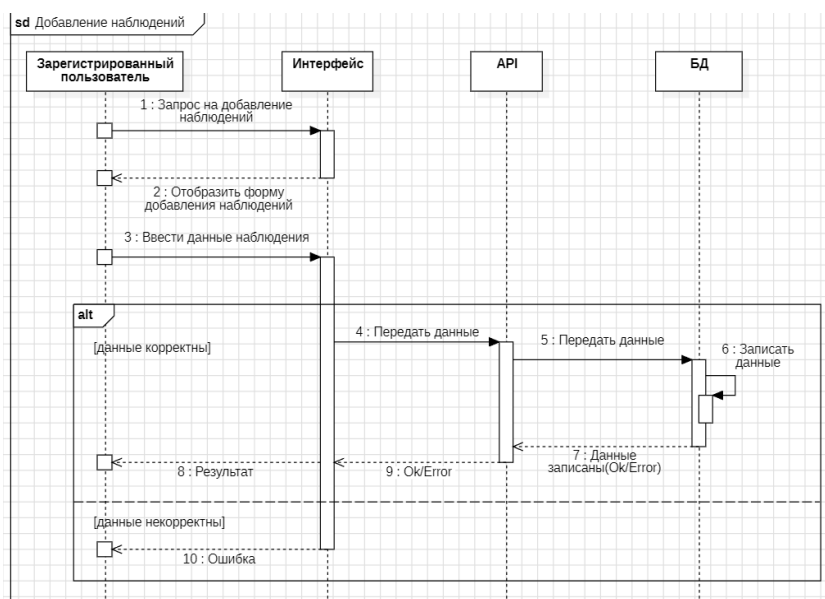


Рис. 2. Диаграмма последовательности действий пользователя при добавлении наблюдений

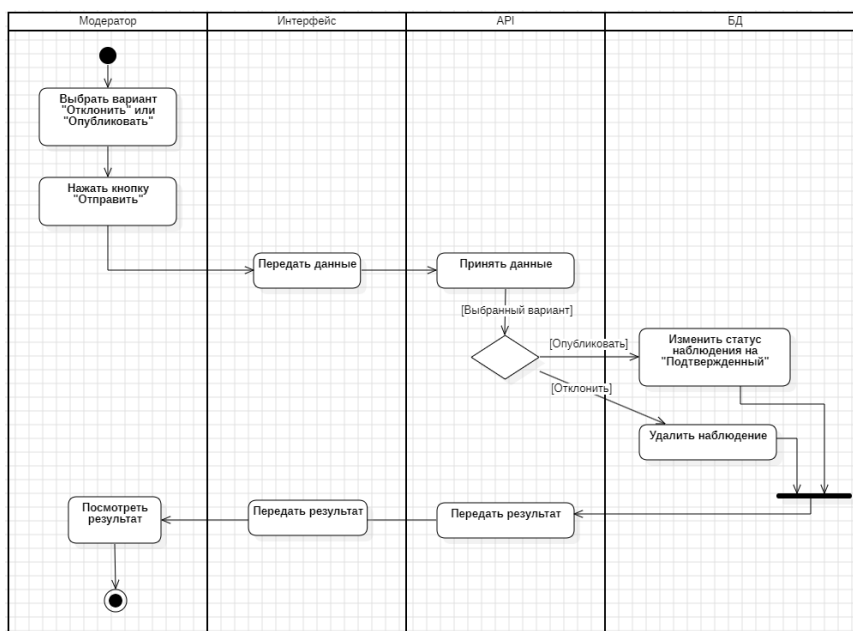


Рис. 3. Диаграмма деятельности прецедента «Модерация наблюдений»

На основе диаграммы классов, при помощи, ранее выбранной СУБД, была разработана база данных информационной системы «Отслеживание брачного лёта муравьев». Её структура представлена на рис. 5 в виде ER-диаграммы.

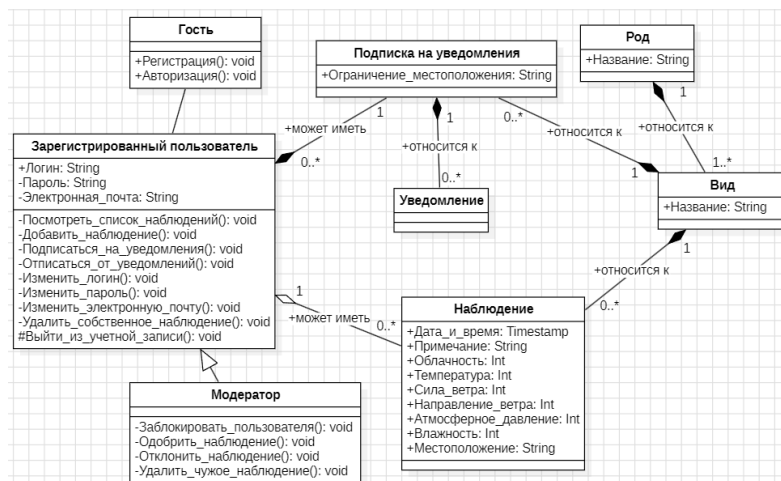


Рис. 4. Диаграмма классов

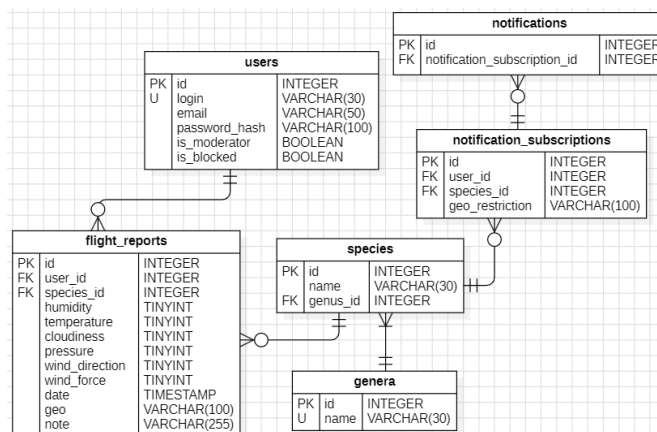


Рис. 5. ER диаграмма

Проектирование макета пользовательского интерфейса

Завершающим этапом проектирования информационной системы «Отслеживание брачного лёта муравьев» будет создание макета пользовательского интерфейса для базовых элементов приложения.

Для создания макета интерфейса был использован инструмент Figma. Это онлайн-платформа для создания интерфейсов и прототипов, которая поддерживает совместную работу в реальном времени. Доступ к сервису предоставляется по подписке, но имеется бесплатный тариф для одного пользователя. Также доступны офлайн-версии для Windows и macOS [17]. При выборе учитывались наличие бесплатной версии, полнота документации и размер сообщества, Figma оказалась лучшим выбором.

Далее рассмотрим главную страницу. (см. рис. 6). В шапке сайта размещено общее и меню для авторизованного пользователя. Первым блоком на странице идет карта с результатами, далее блок фильтров и сами карточки результатов. В карточке выведена фотография, название вида, тип события, комментарий и основные характеристики.

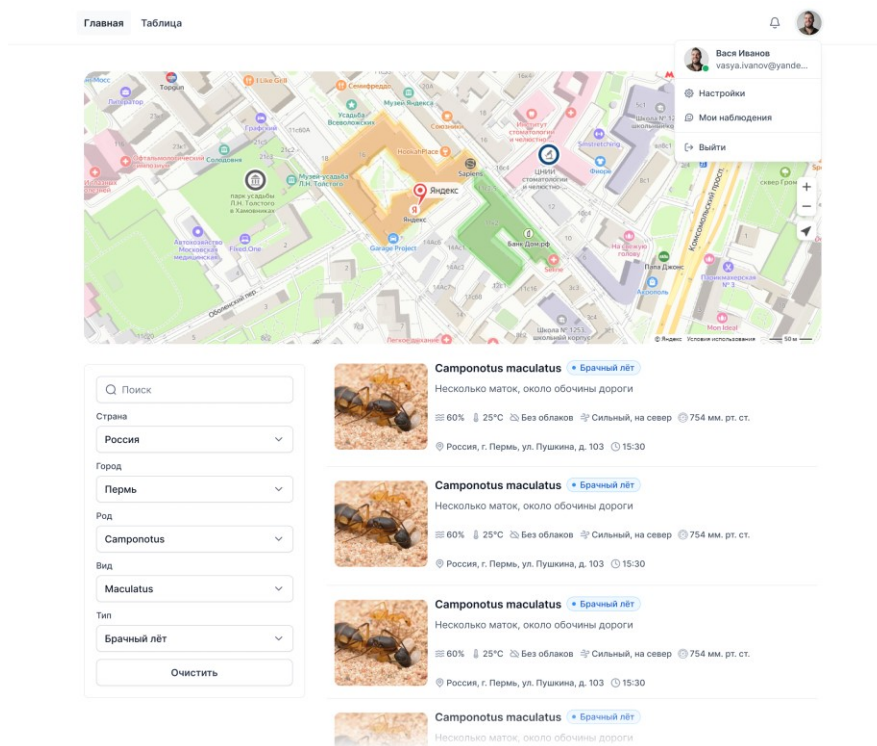


Рис. 6. Главная страница

Далее на рисунке 7 представлена страница добавления наблюдений. Доступен выбор местоположения на карте, загрузка изображений через специальное поле.

На рисунке 8 представлена страница с таблицей лёта. На ней можно увидеть шкалу градации цветов по количеству наблюдений, блок фильтров и саму таблицу. При клике на цветную ячейку таблицы открывается главная страница с предустановленными фильтрами.

Рис. 7. Страница «Добавление наблюдений»

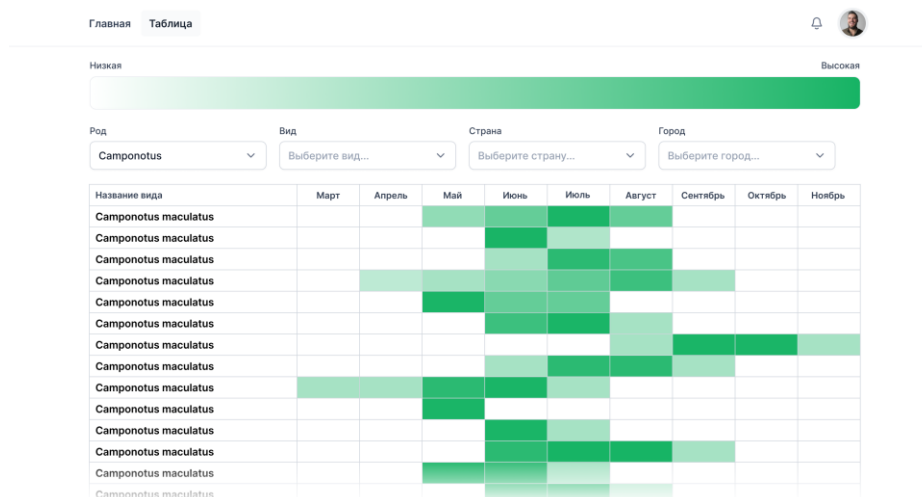


Рис. 8. Страница «Таблица лёта»

Заключение

В ходе выполнения работы были рассмотрены аналоги проектируемой системы «Отслеживание брачного лёта муравьев». Эти аналоги реализуют требуемые функции проектируемой системы, но имеют ряд недостатков. Исходя из темы работы были выявлены недостатки данных информационных систем, наиболее значимым из которых является: излишний функционал и неудобное, перегруженное настройками меню.

На основе анализа существующих реализаций системы были разработаны требования к проектируемой информационной системе и поставлена задача на проектирование.

Далее был проведен анализ средств проектирования и моделирования информационных систем, на основе которого был сделан выбор в пользу CASE-средства StarUML, СУБД MySQL, языка программирования JavaScript в среде выполнения Node.js и был установлен формат взаимодействия web-интерфейса и API.

Библиографический список

1. AntFlights : Flights and Travel Deals [Электронный ресурс] // AntFlights – веб-сайт, предоставляющий информацию о рейсах и туристических предложениях. URL: <https://antflights.com/> (дата обращения: 10.12.2023).
2. AntClub: Сообщество любителей муравьев [Электронный ресурс] // AntClub – веб-сайт, объединяющий любителей муравьев. URL: <http://antclub.ru/> (дата обращения: 10.12.2023).
3. Muravdom: Всё о муравьях [Электронный ресурс] // Муравьиный дом – веб-сайт, посвященный информации о муравьях и их содержании. URL: <https://muravdom.ru/> (дата обращения: 10.12.2023).
4. Case-средства проектирования баз данных [Электронный ресурс] // Высшая школа бизнеса НИУ ВШЭ – веб-сайт, предоставляющий информацию о средствах проектирования

баз данных. URL: <https://hsbi.hse.ru/articles/case-sredstva-proektirovaniya-baz-dannykh/> (дата обращения: 18.01.2024).

5. StarUML: UML моделер [Электронный ресурс] // StarUML – инструмент для моделирования UML. URL: <https://staruml.io/> (дата обращения: 18.01.2024).

6. Rational Rose: Моделирование и разработка [Электронный ресурс] // КПМ Системы – сайт, предоставляющий информацию о Rational Rose. URL: https://www.kpms.ru/Automatization/Rational_Rose.htm (дата обращения: 18.01.2024).

7. Astah Community: UML и другие диаграммы [Электронный ресурс] // Astah – продукт для создания UML и других диаграмм. URL: <https://astah.net/products/astah-community/> (дата обращения: 18.01.2024).

8. Microsoft Visio: Flowchart Software [Электронный ресурс] // Microsoft – программное обеспечение для создания диаграмм. URL: <https://www.microsoft.com/ru-ru/microsoft-365/visio/flowchart-software> (дата обращения: 18.01.2024).

9. CitForum: CASE технологии [Электронный ресурс] // CitForum – веб-сайт, содержащий информацию о CASE технологиях и базах данных. URL: http://citforum.ru/database/case/glava5_1_1.shtml (дата обращения: 18.01.2024).

10. Node.js: JavaScript runtime built on Chrome's V8 JavaScript engine [Электронный ресурс] // Node.js – официальный веб-сайт платформы. URL: <https://nodejs.org/en> (дата обращения: 20.02.2024).

11. Python: Python Programming Language [Электронный ресурс] // Python – официальный веб-сайт языка программирования. URL: <https://www.python.org/> (дата обращения: 20.02.2024).

12. Java: Technology for a Better Connected World [Электронный ресурс] // Java – официальный веб-сайт языка программирования. URL: <https://www.java.com/en/> (дата обращения: 20.02.2024).

13. Ruby on Rails: A web-application framework that includes everything needed to create database-backed web applications [Электронный ресурс] // Ruby on Rails – официальный веб-сайт фреймворка. URL: <https://rubyonrails.org/> (дата обращения: 20.02.2024).

14. React: A JavaScript library for building user interfaces [Электронный ресурс] // React – официальный веб-сайт библиотеки. URL: <https://react.dev/> (дата обращения: 20.02.2024).

15. Angular: The modern web developer's platform [Электронный ресурс] // Angular – официальный веб-сайт платформы. URL: <https://angular.dev/> (дата обращения: 20.02.2024).

16. Vue.js: The Progressive JavaScript Framework [Электронный ресурс] // Vue.js – официальный веб-сайт фреймворка. URL: <https://vuejs.org/> (дата обращения: 20.02.2024).

17. Figma: сервис для разработки интерфейсов и прототипирования [Электронный ресурс] // Figma – веб-сайт, предоставляющий возможности для создания интерфейсов и прототипов с поддержкой совместной работы. URL: <https://www.figma.com/> (дата обращения: 19.04.2024).

References

1. AntFlights: Flights and Travel Deals [Electronic resource] // AntFlights – a website providing information on flights and travel deals. URL: <https://antflights.com/> (date of access: 12/10/2023).
2. AntClub: Ant Enthusiast Community [Electronic resource] // AntClub – a website uniting ant enthusiasts. URL: <http://antclub.ru/> (date of access: 12/10/2023). (In Russ.)
3. Muravdom: All About Ants [Electronic resource] // Ant House – a website dedicated to information about ants and their care. URL: <https://muravdom.ru/> (date of access: 12/10/2023). (In Russ.)
4. Case Tools for Database Design [Electronic resource] // Higher School of Business, HSE – a website providing information on database design tools. URL: <https://hsbi.hse.ru/articles/case-sredstva-proektirovaniya-baz-dannykh/> (date of access: 01/18/2024). (In Russ.)
5. StarUML: UML Modeler [Electronic resource] // StarUML – a tool for UML modeling. URL: <https://staruml.io/> (date of access: 01/18/2024).
6. Rational Rose: Modeling and Development [Electronic resource] // KPM Systems – a website providing information on Rational Rose. URL: https://www.kpms.ru/Automatization/Rational_Rose.htm (date of access: 01/18/2024).
7. Astah Community: UML and Other Diagrams [Electronic resource] // Astah – a product for creating UML and other diagrams. URL: <https://astah.net/products/astah-community/> (date of access: 01/18/2024).
8. Microsoft Visio: Flowchart Software [Electronic resource] // Microsoft – software for creating diagrams. URL: <https://www.microsoft.com/ru-ru/microsoft-365/visio/flowchart-software> (date of access: 01/18/2024).
9. CitForum: CASE Technologies [Electronic resource] // CitForum – a website containing information on CASE technologies and databases. URL: http://citforum.ru/database/case/glava5_1_1.shtml (date of access: 01/18/2024). (In Russ.)
10. Node.js: JavaScript runtime built on Chrome's V8 JavaScript engine [Electronic resource] // Node.js – official website of the platform. URL: <https://nodejs.org/en> (date of access: 02/20/2024).
11. Python: Python Programming Language [Electronic resource] // Python – official website of the programming language. URL: <https://www.python.org/> (date of access: 02/20/2024).

12. Java: Technology for a Better Connected World [Electronic resource] // Java – official website of the programming language. URL: <https://www.java.com/en/> (date of access: 02/20/2024).
13. Ruby on Rails: A web-application framework that includes everything needed to create database-backed web applications [Electronic resource] // Ruby on Rails – official website of the framework. URL: <https://rubyonrails.org/> (date of access: 02/20/2024).
14. React: A JavaScript library for building user interfaces [Electronic resource] // React – official website of the library. URL: <https://react.dev/> (date of access: 02/20/2024).
15. Angular: The modern web developer's platform [Electronic resource] // Angular – official website of the platform. URL: <https://angular.dev/> (date of access: 02/20/2024).
16. Vue.js: The Progressive JavaScript Framework [Electronic resource] // Vue.js – official website of the framework. URL: <https://vuejs.org/> (date of access: 02/20/2024).
17. Figma: Interface Design and Prototyping Service [Electronic resource] // Figma – a website providing tools for interface design and prototyping with collaboration support. URL: <https://www.figma.com/> (date of access: 04/19/2024).

DEVELOPMENT OF AN INFORMATION SYSTEM FOR TRACKING THE NUPTIAL FLIGHT OF ANTS

Yasyrev Mikhail E., Rakina Valeria D.

Perm State University, 15, Bukireva st., Perm, 614068, Russia, yasyrevmikhail@gmail.com

The article presents the process of designing an information system for tracking the nuptial flight of ants. The relevance of the topic under consideration is justified. The analysis of existing systems with similar functionality is carried out. Their main advantages and disadvantages are highlighted, additional requirements are defined. On the basis of the analysis of the subject area the main characteristics are allocated. The functional requirements to the system are established. Design and development tools are analyzed, the most balanced by a number of parameters is chosen. The model of the information system was designed using a number of UML-diagrams, such as precedent, sequence, class diagram. A conceptual data model was also built, and a mockup for the user interface was designed and given.

Keywords: nuptial flight of ants, tracking the nuptial flight of ants, information system design.

Научное издание

Актуальные проблемы математики, механики и информатики 2024

Сборник статей по материалам студенческой конференции

г. Пермь, ПГНИУ,
22–26 апреля 2024 г. (1 сессия),
20–24 мая 2024 г. (2 сессия),
17–28 июня 2024 г. (3 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.

Издается в авторской редакции
Компьютерная верстка: *Д. Р. Бакирова*

Объем данных 12,6 Мб
Подписано к использованию 10.12.2024

Размещено в открытом доступе
на сайте www.psu.ru
в разделе НАУКА / Электронные публикации
и в электронной мультимедийной библиотеке ELiS

Управление издательской деятельности
Пермского государственного
национального исследовательского университета
614990, г. Пермь, ул. Букирева, 15