

ПЕРМСКИЙ  
ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# АКТУАЛЬНЫЕ ПРОБЛЕМЫ МАТЕМАТИКИ, МЕХАНИКИ И ИНФОРМАТИКИ 2022

Сборник статей по материалам  
студенческой конференции



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 2022**

Сборник статей по материалам студенческой конференции

(г. Пермь, ПГНИУ, 25 мая – 10 июня 2022 г.)



Пермь 2022

УДК 51+531+004.8](082)

ББК 22+32.81

A437

**Актуальные** проблемы математики, механики и информатики 2022 [Электронный ресурс] : сборник статей по материалам студенческой конференции (г. Пермь, 25 мая – 10 июня 2022 г.) / под редакцией А. В. Черникова, М. М. Бузмаковой ; Пермский государственный национальный исследовательский университет. – Электронные данные. – Пермь, 2022. – 5,61 Мб; 147 с. – Режим доступа: <http://www.psu.ru/files/docs/science/books/sborniki/aktualnye-problemy-matematiki-mekhaniki-informatiki-2022.pdf>. – Заглавие с экрана.

ISBN 978-5-7944-3908-3

В сборнике представлены материалы студенческой конференции «Актуальные проблемы математики, механики и информатики 2022», которая проводилась 25 мая – 10 июня 2022 г. в г. Перми. В состав сборника вошли работы студентов, выполненные под руководством ученых Пермского государственного национально исследовательского университета. Работы обладают актуальностью и научной новизной, часто они имеют междисциплинарный характер. Большой блок публикаций связан информационными технологиями, так как это направление в настоящее время активно развивается, и в ПГНИУ имеются научные школы, связанные с информационными технологиями.

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

УДК 51+531+004.8](082)

ББК 22+32.81

*Издается по решению ученого совета механико-математического факультета  
Пермского государственного национального исследовательского университета*

*Рецензенты:* доцент кафедры информационных технологий в бизнесе НИУ-ВШЭ – Пермь, канд. физ.-мат. наук, доцент **Л. В. Шесткова**

доцент кафедры прикладной математики и информатики Астраханского государственного университета им. В. Н. Татищева, канд. физ.-мат. наук, доцент **В. А. Черкасова**

ISBN 978-5-7944-3908-3

© ПГНИУ, 2022

## СОДЕРЖАНИЕ

**Беляков К.В.**

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ РЕШЕНИЙ  
АЛГОРИТМИЧЕСКИХ ЗАДАЧ В СРЕДЕ SCRATCH .....6

**Васильев И.Д., Василюк Н.Н.**

ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ  
СИСТЕМЫ УЧЕТА ВРЕМЕНИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ  
СТУДЕНТОВ В КОМПЬЮТЕРНОМ КЛАССЕ .....11

**Васильевых М.Г.**

АНАЛИЗ ВЕРОЯТНОСТИ ВЫХОДА НА СДЕЛКУ  
ДЛЯ КАЖДОЙ БРОНИ НА ПОКУПКУ НЕДВИЖИМОСТИ .....16

**Воронова К.Д., Плаксин М.А.**

«СТРОЙКА» – КОМПЬЮТЕРНАЯ ИГРА ДЛЯ ЗНАКОМСТВА  
С ПАРАЛЛЕЛЬНЫМ ПРОГРАММИРОВАНИЕМ .....25

**Главатских К.Е.**

ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА  
ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОНЛАЙН-  
КОНСУЛЬТИРОВАНИЯ .....30

**Кармазин В.И., Кнутова Н.С.**

ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ  
ЕДИНОЙ ТОЧКИ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ .....36

**Корзников А.О., Дацун Н.Н.**

РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ РАСЧЕТА МЕТРИК КОДА  
НА ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ЯЗЫКЕ  
ПРОГРАММИРОВАНИЯ ..... 40

**Котлов М.А., Соловьева Т.Н.**

ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ УЧЁТА  
АКТИВОВ ПРЕДПРИЯТИЯ .....46

**Летовальцев Д.Д., Кнутова Н.С.**

ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ  
СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ АЛГОРИТМОВ  
ДИАЛОГОВЫХ ИНТЕРФЕЙСОВ ДЛЯ ВНЕШНИХ СИСТЕМ .....52

**Лукьянов А.М.**

РАЗРАБОТКА ОНТОЛОГИЧЕСКИ УПРАВЛЯЕМОГО  
ПРОТОКОЛА СЕТЕВОЙ САМОИДЕНТИФИКАЦИИ УСТРОЙСТВ  
ИНТЕРНЕТА ВЕЩЕЙ .....58

<b>Маргин Е.Р., Соловьева Т.Н.</b>	
ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМ ДЛЯ СВОМЕСТНОЙ РАЗРАБОТКИ ПРОЕКТОВ .....	63
<b>Миргалеев А.А., Ознобихина М.С.</b>	
ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ КАРТ КОНДИТЕРСКИХ ИЗДЕЛИЙ ДЛЯ КОНДИТЕРА .....	73
<b>Мосеева Ю.С., Огурецкая А.И.</b>	
ЭЛЕКТРОННАЯ КУЛИНАРНАЯ КНИГА .....	78
<b>Мустафина Н.И., Плаксин М.А.</b>	
ОТНОШЕНИЕ СТУДЕНТОВ К ПРИМЕНЕНИЮ ДЕЛОВЫХ ИГР И КОМПЬЮТЕРНЫХ СИМУЛЯТОРОВ В ИЗУЧЕНИИ ПРОГРАММНОЙ ИНЖЕНЕРИИ. СРАВНЕНИЕ ПЕРМСКОГО ГОСУНИВЕРСИТЕТА И ПЕРМСКОГО ФИЛИАЛА ВЫСШЕЙ ШКОЛЫ ЭКОНОМИКИ.....	83
<b>Пичкалев М.В., Василюк Н.Н.</b>	
ПРОЕКТИРОВАНИЕ И ПРОТОТИПИРОВАНИЕ ИС СООТНЕСЕНИЯ КОМПЕТЕНЦИЙ, ФОРМИРУЕМЫХ ФГОС (СУОС) ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ, И ТРЕБОВАНИЙ ПРОФЕССИОНАЛЬНЫХ СТАНДАРТОВ.....	89
<b>Пономарев А.Ф., Кнутова Н.С.</b>	
ПРОЕКТИРОВАНИЕ СИСТЕМЫ СБОРА WHOIS ДАННЫХ ДОМЕННЫХ ИМЕН И ИХ АНАЛИЗА.....	96
<b>Проскуряков В.С.</b>	
СОЗДАНИЕ БЛОКА КОЭФФИЦИЕНТОВ КОРРЕЛЯЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ВИБРОДИАГНОСТИКИ.....	100
<b>Сычев И.А., Степанов В.А.</b>	
ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ СБОРА И ОБРАБОТКИ ИЗОБРАЖЕНИЙ ДЛЯ МАШИННОГО ОБУЧЕНИЯ ВНЕШНИХ СИСТЕМ .....	112
<b>Тляшева И.И., Огурецкая А.И.</b>	
ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЗАЦИИ ПРИНЯТИЯ И ОБРАБОТКИ ЗАКАЗОВ В КАФЕ И РЕСТОРАНАХ.....	121

<b>Тудвасев И.В., Кнутова Н.С.</b>	
ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ ДЛЯ СОЗДАНИЯ, ХРАНЕНИЯ И ВОССТАНОВЛЕНИЯ РЕЗЕРВНЫХ КОПИЙ БАЗ ДАННЫХ MYSQL.....	128
<b>Угринов В.А., Кушев В.О.</b>	
ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ «ПОДАРИЮ – ЗАБЕРУ».....	134
<b>Шабahов Е.М.</b>	
ШИФРОВАНИЕ В СОЦИАЛЬНОЙ СЕТИ ВКОНТАКТЕ БРАУЗЕРНЫХ РАСШИРЕНИЙ.....	141

## **АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ РЕШЕНИЙ АЛГОРИТМИЧЕСКИХ ЗАДАЧ В СРЕДЕ SCRATCH**

*Беляков Кирилл Викторович*

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

Олимпиады являются хорошим методом развития навыков учащихся. Однако при проведении олимпиад по программированию среди младших и средних классов возникают трудности в освоении языков программирования, решением которых является использование языка визуального программирования Scratch. При использовании Scratch возникает проблема тестирования из-за отсутствия автоматического запуска и работы с файлами. Решением является транспилирование программ на Scratch в делегаты языка C# и использование их в системе автоматизированного тестирования. В данной работе представлены проектирование, реализация и тестирование системы автоматизированного тестирования программ на языке Scratch, а также проектирование, реализация и тестирование библиотеки классов, конвертирующей Scratch-проекты в делегаты языка C# посредством использования интерфейса деревьев выражений.

Ключевые слова: автоматизированное тестирование, языки программирования, методы трансляции, транспилирование.

### **Необходимость автоматизации процесса тестирования**

Основным методом выполнения тестирования при проведении небольших школьных олимпиад, с небольшим числом участников является метод ручного тестирования. Однако, если рассматривать более крупные соревнования, то данный подход абсолютно не уместен из-за огромных временных затрат.

Решающим данную проблему методом является автоматизированная проверка решений. Использование системы тестирования сокращает к минимуму нагрузку на проверяющего, ограничивая его работу заполнением системы необходимыми данными. Хотя данный метод и теряет в гибкости по сравнению с ручной проверкой, он значительно облегчает проверку.

### **Описание принципов транспилирования**

Транслятор – преобразование программы, представленной на одном языке программирования, в программу на другом языке [1]. Транспайлер – один из видов транслятора, принимающий на вход исходный код одного высокоуровневого языка и возвращающий эквивалентный исходный код того же или другого языка [2].

Среди способов применения транспилирования можно выделить: межверсионный перенос исходного кода; перенос или обработка исходного кода в целях оптимизации скорости работы; использование для написания кода языка близкого к исходному, который включает в себя дополнительные функции, и дальнейший перенос полученного кода в код исходного языка.

Этапы трансляции включают в себя: лексический, синтаксический и семантический анализ, создание промежуточной модели и генерация кода.

Структура программы на языке Scratch [3] представляет собой заархивированное абстрактное синтаксическое дерево, хранящееся в формате JSON [4]. В то же время целевой программный интерфейс Expression [5] языка C# [6] представляет собой дерево выражений, что сводит процесс транпилирования к этапам создания промежуточной модели и генерации кода.

### Проектирование

Выделены функциональные требования: создание, изменение, удаление участников, задач, решений и тестов; сохранение и восстановление сессии; сохранение результатов тестирования; использование параллельных вычислений при выполнении тестирования; отображение процесса выполнения тестирования; масштабируемость элементов управления.

Выделены нефункциональные требования: производительность – проверка 2000 решений по 10 тестов на каждое не должна занимать больше 2 минут; отказоустойчивость – продолжение работы при возникновении ошибок в решениях во избежание потери результатов работы.

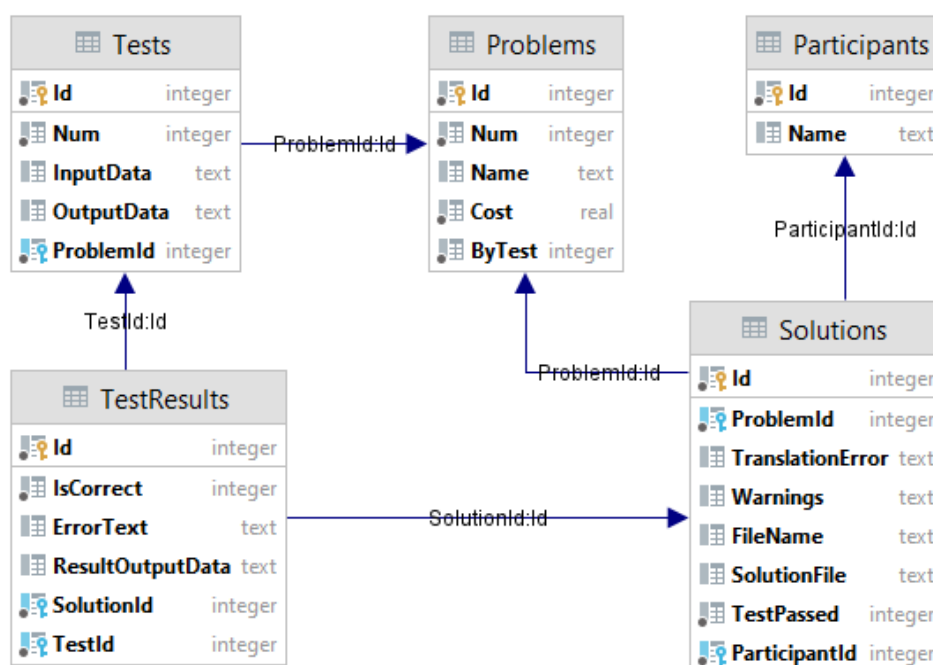


Рис. 1. Диаграмма базы данных

Выделены классы сущностей системы (см. рис. 1), которые будут храниться в базе данных: Participant – участник, Problem – задача, Solution – решение задачи, Test – тест, TestResult – результат теста.

Выделены классы библиотеки-конвертера и системы автоматизированного тестирования.

### Реализация

На основе спроектированных классов реализованы компоненты библиотеки конвертера и системы тестирования (см. рис. 2).



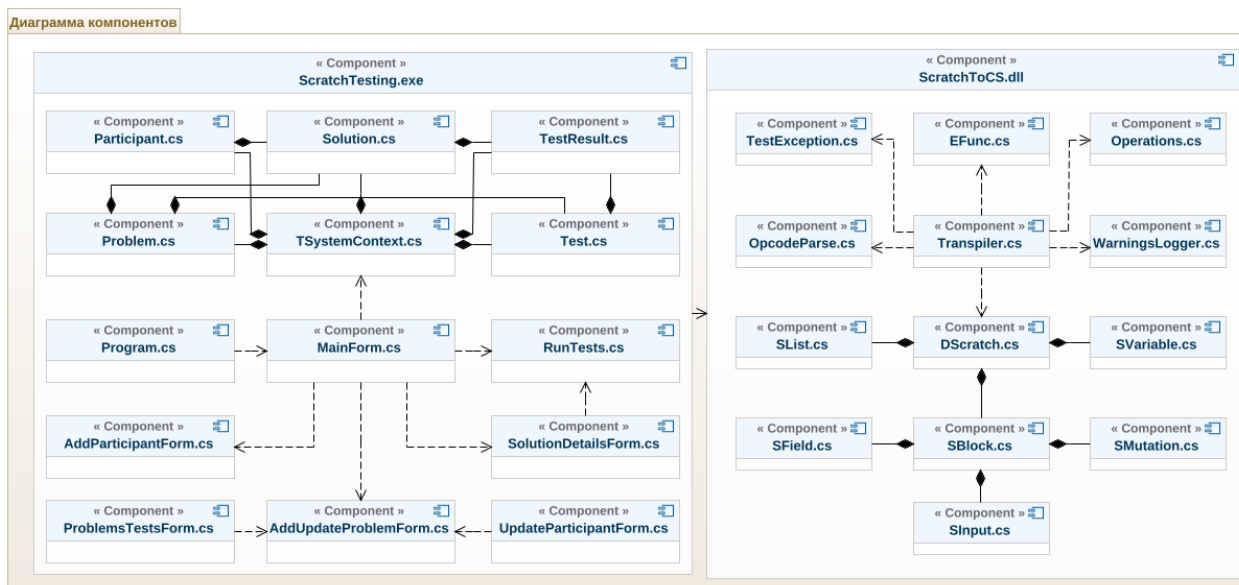


Рис. 2. Диаграмма компонентов

Система реализована с использованием программного интерфейса Windows Forms [7].

### Тестирование

Проведено общее тестирование библиотеки-конвертера на наборе разных решений одной олимпиадной задачи, а также тестирование конкретных частей реализации на различных наборах блоков, включающих все реализованные. Результаты библиотеки соответствуют результатам клиентского приложения Scratch.

Проведено сравнение времени выполнения реализации алгоритма сортировки пузырьком в клиентском приложении Scratch и в реализованной библиотеке (см. рис. 3).

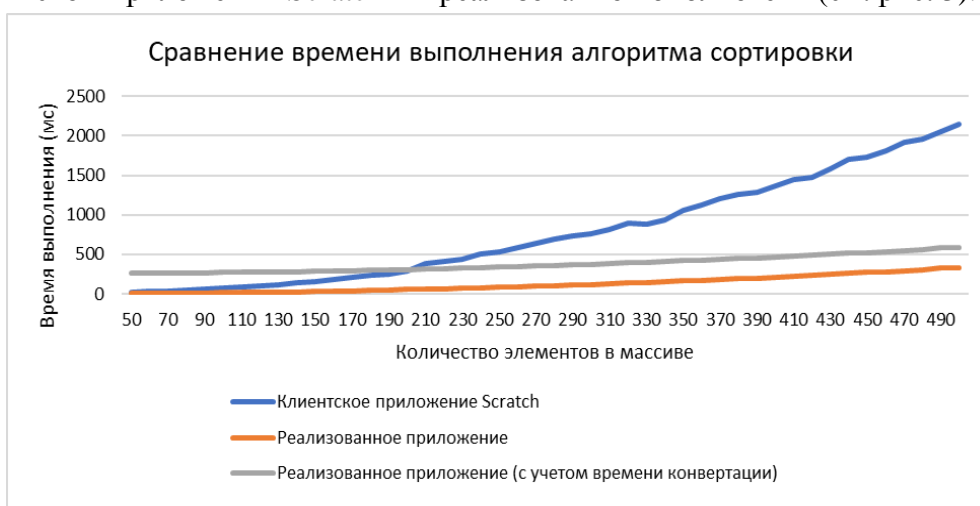


Рис. 3. Диаграмма сравнения времени выполнения алгоритма в разных приложениях

Проведено автоматическое модульное тестирование реализованной системы средствами среды Visual Studio [8] и библиотек Appium.WebDriver [9] и Windows Application Driver [10].

Проведено сравнение времени выполнения параллельных и не параллельных алгоритмов распаковки и тестирования решений (см. рис. 4, 5).

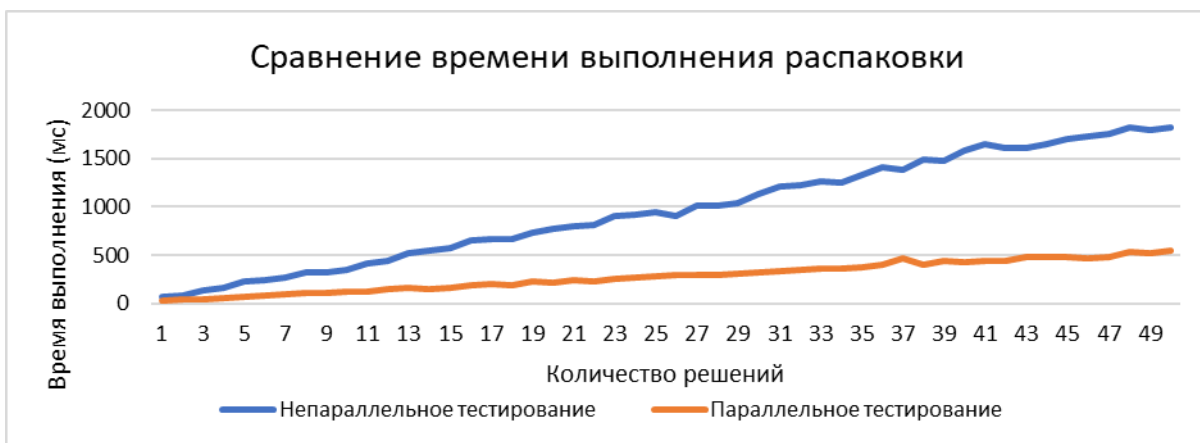


Рис. 4. Диаграмма сравнения времени выполнения распаковки с параллельной обработкой и без

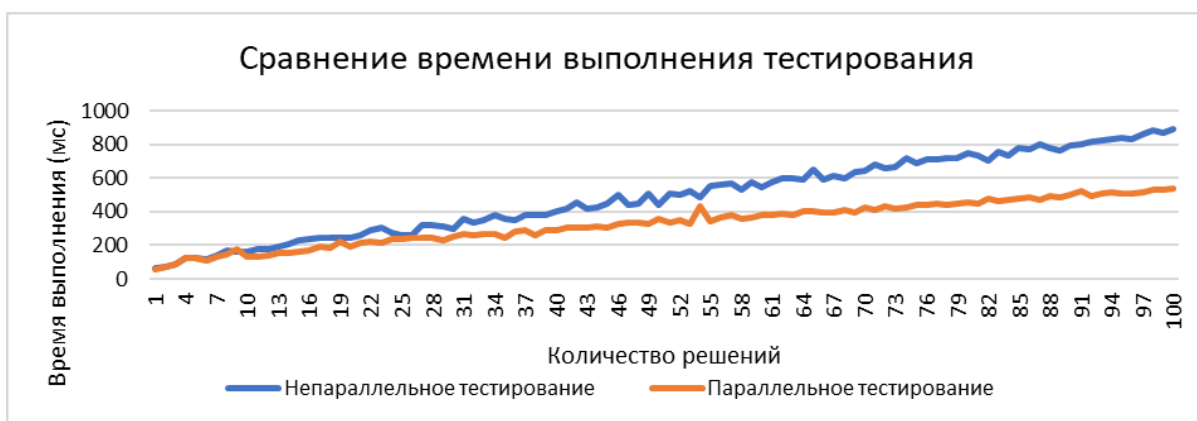


Рис. 5. Диаграмма сравнения времени выполнения тестирования с параллельной обработкой и без

### Заключение

По результатам работы были спроектированы, реализованы и протестированы библиотека классов, позволяющая преобразовывать исходный код на языке Scratch в делегаты языка C#, и система автоматизированного тестирования, позволяющая за небольшой промежуток времени выполнять проверку большого числа решений олимпиадных задач.

Данная система использовалась при тестировании работ участников заочного и очного туров краевого конкурса «Мартовский заяц». Были внедрены все предложения по доработке программы, поступившие от организационного комитета после её использования.

### Библиографический список

1. Дроздов А.В., Мансурова Т.П., Блинников А.В. Технологии трансляции языков программирования высокого уровня // Наука, технологии, общество – НТО-2021: материалы всерос. науч. конф. – Красноярск: ОУ «ККДГИТ», 2021. – С. 88-92.
2. Types of compilers [Электронный ресурс] UPL: <http://www.compilers.net/paedia/compiler/index.htm/> (дата обращения 07.01.22).
3. Алейникова Т.Г., Шербах А.И. Scratch-Программирование – Инструмент развития вычислительного мышления: Примеры решения задач в среде Scratch [Текст] / Т.Г. Алейникова., А.И. Шербах // Народная асвета. – 2019. – № 3. – С. 15-19.
4. Формат JSON [Электронный ресурс] UPL: <https://learn.javascript.ru/json> (дата обращения 07.01.22).

5. Деревья выражений | Microsoft Docs (C#) [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/expression-trees/> (дата обращения 07.01.22).
6. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. – М.: Вильямс, 2015. – 486 с.
7. Петцольд, Чарльз Программирование с использованием Microsoft Windows Forms / Чарльз Петцольд // Питер, Русская Редакция. – 2006. – 432 с.
8. Visual Studio: IDE и редактор кода для разработчиков и групп, работающих с программным обеспечением [Электронный ресурс] URL: <https://visualstudio.microsoft.com> (дата обращения 28.05.22).
9. Appium: Mobile App Automation Made Awesome [Электронный ресурс] URL: <https://appium.io> (дата обращения 28.05.22).
10. GitHub – microsoft/WinAppDriver: Windows Application Driver [Электронный ресурс] URL: <https://github.com/microsoft/WinAppDriver> (дата обращения 28.05.22).

## **AUTOMATION OF TESTING SOLUTIONS TO ALGORITHMIC PROBLEMS IN THE SCRATCH ENVIRONMENT**

*Belyakov Kirill V.*

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

Student competitions are a good method of developing students' skills. However, when conducting programming competition among junior and middle classes, difficulties arise in mastering programming languages. The solution is the use of the Scratch visual programming language. When using Scratch, there is a testing problem due to the lack of autorun and working with files. The solution is to transpile Scratch programs into C# delegates and use them in an automated testing system. This paper presents the design, implementation and testing of a system for automated testing of programs in the Scratch language, as well as the design, implementation and testing of a class library that converts Scratch projects into C# delegates by using the expression tree interface.

Keywords: automated testing, programming languages, translation methods, transpiling.

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

*Васильев Игорь Дмитриевич, Василюк Надежда Николаевна*

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

В данной статье рассмотрены наиболее популярные аналоги информационных систем учета самостоятельной работы студентов в компьютерном классе, проведен сравнительный анализ этих систем, и на его основе выявлены и определены функциональные требования к проектируемой информационной системе (ИС). ИС планируется использовать для дистанционной записи студентов, для хранения данных о компьютерах, студентах и времени их работы и т.д. Для моделируемой ИС определена соответствующая методология и набор инструментальных средств, а также инструменты для прототипирования интерфейса. Работа содержит описание логики работы информационной системы, а также примеры возможного интерфейса пользователя. Спроектированная информационная система может быть использована студентами и сотрудниками университета, отвечающими за работу компьютерного класса. Возможно применение в образовательных учреждениях другого уровня образования.

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

### **Введение**

Как правило, в настоящее время студенту, чтобы записаться на самостоятельную работу в компьютерный класс, необходимо каждый раз лично подходить к сотруднику, ответственному за компьютерный класс, который по разным причинам может отсутствовать на рабочем месте. В свою очередь, сотрудник компьютерного класса должен понимать, сколько студентов записалось и в какое время, за какими компьютерами, они будут работать, сколько времени потребуется для решения их задач, и с какой целью им потребуется компьютер.

Проектирование ИС с поддержкой таких задач является актуальной проблемой, которая позволит упростить доступ студентов к записи и облегчит анализ отчетов. Таким образом, объектом исследования является учет времени самостоятельной работы студентов. Предметом исследования является автоматизация учета времени самостоятельной работы студентов в компьютерном классе.

Цель настоящей работы – проектирование и документирование ИС учета времени студентов для самостоятельной работы в компьютерном классе, для автоматизации работы сотрудника, отвечающего за компьютерный класс.

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

Рассмотрим примеры некоторых информационных систем учета времени.

1С: Университет [1] является обширным программным обеспечением, основной задачей которого является анализ и обработка всех данных студентов. К достоинствам системы можно отнести хранение и обработка сведений о контингенте студентов вуза, учет посеще-

ния студентов, создание почасовых и суточных отчетов по посещениям студентов. Наиболее значимые недостатки системы – это долгое внедрение и развертывание.

Информационная система «Электронный журнал» [2] предназначена для ведения учета и анализа учебной деятельности студентов в образовательных организациях. Основным функционалом ИС включает: осуществление гибкой системы настроек структуры учебного года, просмотр и редактирование данных в электронном журнале в соответствии с уровнем доступа и обеспечение контроля посещаемости студентов. Данный продукт не может быть интегрирован в другие системы, кроме колледжей Подмосквья.

Программа Yclients [3] предназначена для учета клиентов различных предприятий и всевозможных сфер услуг. Эта система позволяет записать людей на определенный день и время, предоставляет возможность онлайн записи клиентов, ведение статистики по дням, месяцам, кварталам и годам.

Универсальная система учета [4] представляет собой программу, в которой администратор может оперативно принимать заявки от клиентов и вести запись в электронном формате.

Рассмотренные выше системы, являются многофункциональными и имеют как достоинства, так и ряд недостатков. Многие из представленных оболочек не ориентированы на учет времени самостоятельной работы студента, что является критическим в выборе программного продукта. Особенно важной чертой продукта должна быть запись студента через сеть Интернет, что присутствует только в 2-х рассмотренных системах.

На основе анализа существующих систем сформулированы следующие функциональные требования к проектируемой ИС:

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

### **Выбор методологии моделирования, case-средства и инструмента прототипирования интерфейса**

Крайне важно выбрать подходящую методологию для проектирования для того, чтобы в достаточной степени описать протекающие процессы внутри системы. В конечном итоге именно от этого зависит, достаточно ли наглядной, удобной и понятной окажется модель.

При анализе основных методологий проектирования информационных систем, таких как IDEF [5], BPMN [6], UML [7]. Было принято решение использовать UML, так как эта методология обладает простыми правилами моделирования и интуитивно понятными элементами диаграмм. Схемы, построенные в UML, могут понимать все заинтересованные лица, и могут использоваться как средство общения между людьми.

В качестве case-средства была выбрана среда разработки StarUML [8], потому что она обладает бесплатной версией, высокую степень расширяемости, на хорошем уровне настраивается с требованиями пользователя, а также поддерживает обратный инжиниринг. В каче-

стве «конкурентов» были и другие редакторы UML-диаграмм такие как, Violet UML Editor [9], Diagram Designer [10], ArgoUML [11], но у них были существенные недостатки, по сравнению с StarUML.

Для прототипирования интерфейса, была выбран графический редактор Figma, так как имеет ряд достоинств:

- простой и удобный интерфейс;
- возможность работать через браузер;
- созданные макеты хранятся в облаке и можно удобно контролировать версии.

### Проектирование информационной системы

Доступ к ИС для лаборанта будет производиться со стационарного устройства, а для студентов через интернет-сайт. Для достижения целей визуального проектирования с использованием нотации UML строится диаграмма вариантов использования. Диаграмма вариантов использования – это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки.

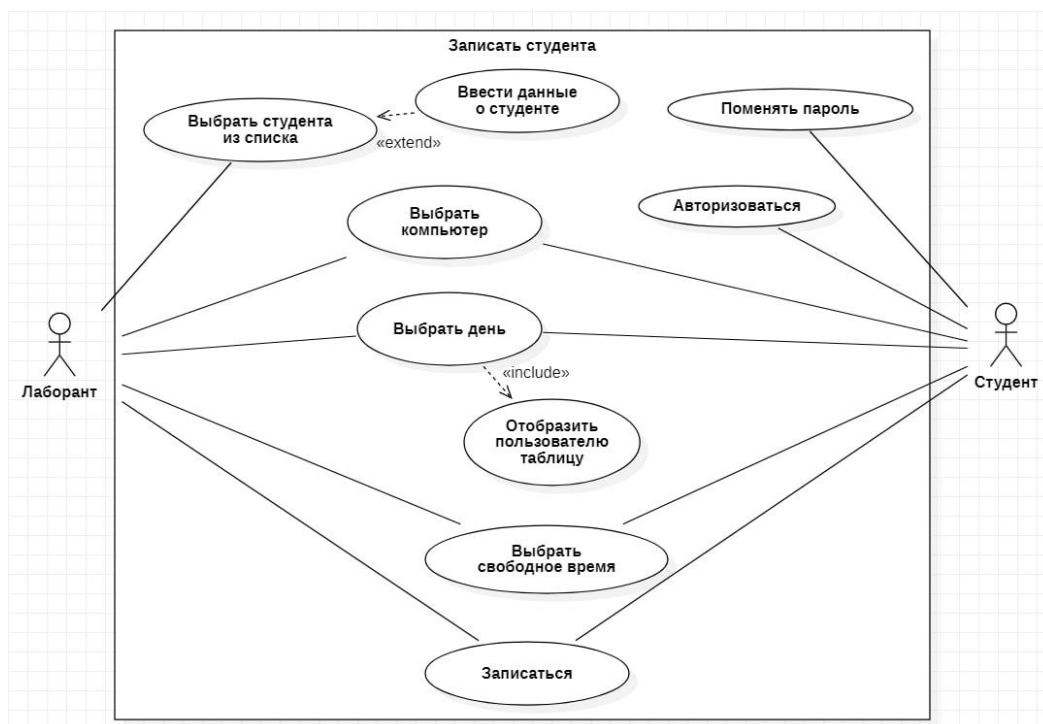


Рис. 1. Диаграмма прецедентов

Рассмотрим диаграмму прецедентов (рис. 1). На ней изображены два актера: студент и лаборант. ИС подразумевает, что для записи студента на свободный компьютер для лаборанта, необходимо проделать следующие операции, в том порядке, который указан далее:

- 1) выбрать пришедшего студента, по его номеру студенческого билета, а если студента нет в системе, указать информацию о студенте, а именно: имя, фамилию, номер студенческого билета, пароль сгенерируется автоматически;
- 2) выбрать компьютер, соответствующий требованиям и пожеланиям студента;
- 3) выбрать интересующий день, для работы в компьютерном классе;
- 4) выбрать свободное время;
- 5) сохранить запись.

В свою очередь, для записи в компьютерный класс будучи студентом, необходимо сначала авторизоваться, чтобы у ИС была точная информация, какой студент хочет записаться, а также для того чтобы защитить информацию от злоумышленников. Еще у студента есть возможность поменять свой пароль, так как он выдается лаборантом при первой регистрации. Таким образом, у студента последовательность действий точно такая же, как у лаборанта, только первый пункт заменяется авторизацией.

Также рассмотрим интерфейс системы (рис. 2), для записи на определенный день и время. Интерфейс должен быть понятен пользователю, и не слишком перегружен излишней информацией.

Computer\_6  
Выберите подходящий день и время, когда вы сможете прийти

Часы записи	Записаться
9:00 - 10:00	<input type="checkbox"/>
10:00 - 11:00	<input type="checkbox"/>
11:00 - 12:00	<input checked="" type="checkbox"/>
12:00 - 13:00	<input checked="" type="checkbox"/>
13:00 - 14:00	<input checked="" type="checkbox"/>
14:00 - 15:00	<input type="checkbox"/>
15:00 - 16:00	<input type="checkbox"/>
16:00 - 17:00	<input type="checkbox"/>
17:00 - 18:00	<input type="checkbox"/>

Рис. 2. Интерфейс записи студента

### Заключение

Разработанная модель информационной системы позволит вести учет времени работы студентов в компьютерном классе, будет хранить данные о компьютерах, студентах и времени их работы, будет предоставлять дистанционный доступ к записи для студентов и обладает удобным интерфейсом, а также обладает минимально необходимой защитой от злоумышленников.

Данная информационная система может быть использована для записи на самостоятельную работу в компьютерном классе учащихся среднего, высшего и профессионального образования.

### Библиографический список

1. 1С:Университет. Возможности продукта [Электронный ресурс]. – 2022. – URL: <https://solutions.1c.ru/catalog/university/features> (дата обращения 03.03.2022)
2. Платформа «Цифровой колледж Подмосковья». Руководство пользователя [Электронный ресурс]. – 2022. – URL: [https://e-learning.tspk-mo.ru/seo/help/module\\_2\\_2\\_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAABAI%3D](https://e-learning.tspk-mo.ru/seo/help/module_2_2_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAABAI%3D) (дата обращения 05.03.2022)

3. Yclients. Описание продукта [Электронный ресурс]. – 2022. – URL: <https://www.yclients.com/ru> (дата обращения 05.03.2022)
4. Универсальная система учета [Электронный ресурс]. – 2022. – URL: [usu.kz](http://usu.kz) (дата обращения 05.03.2022)
5. Основные методологии обследования организаций. Стандарт IDEF0. [Электронный ресурс]. – 2022. – URL: <https://www.cfin.ru/vernikov/idef/idef0.shtml> (дата обращения 05.03.2022)
6. Нотация BPMN 2.0: ключевые элементы и описание [Электронный ресурс]. – 2021. – URL: <https://www.comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/> (дата обращения 05.03.2022)
7. Бабич А. Введение в UML [Электронный ресурс]. – 2022. – URL: <https://intuit.ru/studies/courses/1007/229/info> (дата обращения 10.03.2022)
8. StarUML documentation [Электронный ресурс]. – 2022. – URL: <https://docs.staruml.io/> (дата обращения 10.03.2022)
9. Classic Violet Review [Электронный ресурс]. – URL: <https://horstmann.com/violet/> (дата обращения 10.03.2022)
10. Diagram Designer [Электронный ресурс]. – 2020. – URL: <https://logicnet.dk/DiagramDesigner/> (дата обращения 10.03.2022)
11. Проектирование UML (ArgoUML) [Электронный ресурс]. – 2022. – URL: <https://studwood.net/1707669/informatika/proektirovanie> (дата обращения 10.03.2022)

## **DESIGNING AND DOCUMENTING AN INFORMATION SYSTEM FOR RECORDING THE TIME OF INDEPENDENT WORK OF STUDENTS IN A COMPUTER CLASSROOM**

*Vasilev Igor D., Vasiluk Nadezhda N.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [vasilevid1111@mail.ru](mailto:vasilevid1111@mail.ru)

In this article, the most popular analogues of information systems for accounting for students' independent work in a computer classroom are studied, a comparative analysis of these systems is carried out, and on its basis, and functional requirements for the designed system are identified and determined. The system plans remote recording for students, storing data about computers, students and their working hours, and other necessary functions. The corresponding methodology and a set of tools, as well as tools for prototyping the interface, are defined for the simulated information system. The work contains a description of the logic of the information system, as well as a possible user interface. The designed information system can be used for an employee of a computer class and for students who want to enroll in a computer class for independent work.

Keywords: information system, time accounting, student's independent work, computer class, uml diagrams.



## АНАЛИЗ ВЕРОЯТНОСТИ ВЫХОДА НА СДЕЛКУ ДЛЯ КАЖДОЙ БРОНИ НА ПОКУПКУ НЕДВИЖИМОСТИ

*Васильевых Михаил Геннадьевич*

Пермский государственный национальный исследовательский университет,  
614990, Россия, г. Пермь, ул. Букирева, 15, vasilevyh00@gmail.com

В статье представлено описание разработки системы для прогнозирования вероятности выхода на сделку брони, на основе данных о бронях строительной компании. Прогнозирование вероятности выхода брони на сделку может оказаться полезным для менеджеров отдела продаж, а также людям, ответственным за планирование экономических показателей. В ходе работы данные были предварительно очищены и разбиты на два множества, тестовое и обучающее. Далее были рассмотрены различные методы прогнозирования, такие как классификация, единичные модели и ансамбли и кластеризация. Выбраны модели knn, svm, k-means, случайный лес, решающее дерево, стекинг, нейронная сеть. Также были выбраны методы оценивания моделей, на основании которых был проведен сравнительный анализ, в результате которого выяснилось, что наиболее точной моделью оказался случайный лес.

Ключевые слова: искусственный интеллект, прогнозирование, случайный лес, беггинг, стекинг, бустрепинг, knn, svm, k-means, случайный лес, решающее дерево, нейронная сеть, Data mining.

### **Введение**

Рынок недвижимости один из самых крупных, развитых и прибыльных на данный момент. Однако прибыль можно увеличить, правильно используя современные технологии.

Для многих застройщиков важной проблемой является то, что не все брони выходят на сделку. На это может повлиять множество факторов, а именно: цена объекта, его расположение, способ оплаты, время брони и тому подобное. Если заранее предугадывать поведение потенциальных покупателей, то можно предлагать своевременные скидки на объекты или игнорировать «случайные» брони, тем самым сэкономить время работы менеджера отдела продаж, более точно планировать финансовые показатели.

Для анализа подобных данных и прогнозирования поведения покупателей можно использовать средства Data mining. Для решения описанной ранее проблемы могут помочь такие методы классификация, кластеризация, ансамбли, применяемые в Data mining [6].

Кластеризация позволяет разбить совокупность брони разными способами. Но нас интересует выделение 2 конкретных классов таким образом, чтобы метки классов совпадали с тем, выходит бронь на сделку или нет.

Классификация позволяет отнести объект к тому или иному заранее определенному классу исходя из его свойств.

Ансамбли позволяют использовать несколько моделей одновременно и агрегировать их результаты, для понижения смещения и разброса в предсказании при использовании одной модели.

## **Предобработка данных**

Предварительная обработка данных должны проводиться до того, как данные будут использованы в обучении. В сырых данных достаточно часто встречаются ошибки и неточности и пропущенные значения. Что зачастую приводит к некорректным результатам моделирования. Эти задачи являются частью процесса обработки и анализа данных и обычно подразумевают первоначальное изучение набора данных, используемого в моделировании.

Строго определенной методики не существует, однако есть множество разных приемов для очистки данных и в разных исследованиях для разных данных применяются разные, зачастую уникальные приёмы.

Исходные данные в решаемой задаче содержали характеристики 5519 броней, из которых 1945 вышли на сделку, 3364 отказались от сделки и 210 находятся в резерве.

Были исключены брони, находящиеся в резерве, так как по ним мы не сможем оценить качество прогноза, поскольку на момент исследования неизвестно вышли они на сделку или нет.

Еще несколько записей были исключены по причине выбросов, данных существенно отличающихся от других наблюдений. Они могут соответствовать реальным отклонениям, но могут быть и просто ошибками.

Так же помимо удаления данных необходимо обработать отсутствующие значения, если это возможно.

Характеристики «Этаж», «СкидкаНаКвартиру», «ВариантОплатыДоп» заменялись значением по умолчанию, для «Этаж» и «СкидкаНаКвартиру» это 0, а для «ВариантОплатыДоп» – «нет». Значения характеристик «Тип» и «Продаваемая площадь» заменяли медианой по всему набору данных.

Были выявлены и исключены из общего набора броней те, в которых не указан тип объекта и город, в котором он бронируется, а также брони, где не был указан вариант оплаты.

После очистки данных осталось 4775 броней, из них 1683 вышедших на сделку.

Так же часть параметров была нормализована, то есть приведена к диапазону единичной длины. В данной работе применялся метод минимаксной нормализации.

В рассматриваемом наборе 11 из 21 характеристик были категориальные. Поэтому возникла необходимость кодировать данный вид данных. Для этого применялся метод OneHot [5].

Полученная выборка была разделена в отношении 9 к 1 на обучающий и тестовый наборы.

## **Методы оценивания**

Одной из главных задач принятия решения о выборе того или иного алгоритма и обоснованности его результатов является оценка точности. Существует 2 метода оценивания алгоритмов, внутренняя и внешняя оценки [2].

В качестве методов оценивания использовались Ассигасу, F1 мера, матрица ошибок и сумма квадратов расстояний внутри кластера.

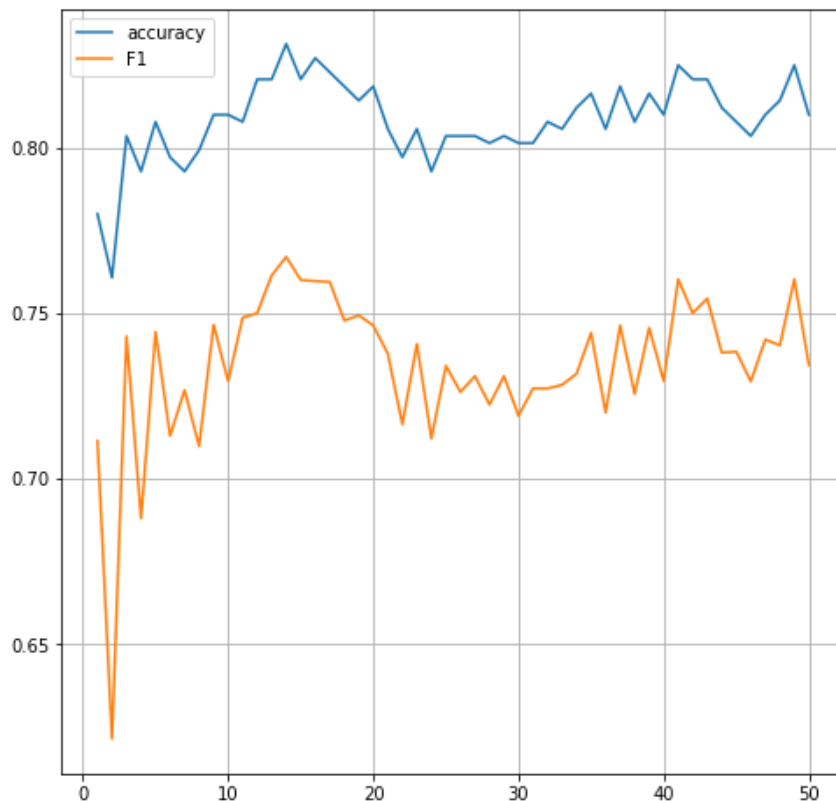
## **Обучение алгоритмов**

Для того, чтобы построить эффективную модель, необходимо правильно подобрать параметры. Для этого строились графики зависимости точности и F1 меры от перебираемого

параметра. Графики строятся при помощи библиотеки matplotlib и seaborn. На основе исследования графика нужно выявить наилучшее значения параметра и по нему построить модель.

### **KNN**

Для того чтобы определить наилучшее значение параметра  $k$  – количества соседей, был построен график (рис. 1) зависимости точности и F1 меры от значения параметра  $k$ . Из него видно, что наиболее оптимальное значение равно 15.



**Рис. 1. График зависимости точности и F1 меры от количества соседей в алгоритме KNN**

В результате чего для наилучшего набора параметров получена точность 82%, F1 мера 76%.

### **Решающее дерево**

Аналогично описанному выше, из анализа графика был сделан вывод, что наиболее оптимальное значение глубины дерева равно 8. В результате чего для наилучшего набора параметров получена точность 88%, F1 мера 85%. Данный алгоритм позволяет добиться более точного прогноза чем, KNN.

### **Случайный лес**

Бэггинг – это ансамблевый метод [3], в котором независимо и параллельно обучаются однородные модели, а результаты их работы объединяют, следуя некоторому детерминированному процессу. Формирование выборки для обучения каждой отдельной модели происходит при помощи бустрепинга. Бустрепинг – метод генерации выборок заданного размера из исходного набора данных путем случайного выбора элементов с повторениями.

Частный случай использования бэггинга – это случайный лес, где в роли базовой модели используется решающее дерево [4].

Так как лес является ансамблевым методом, использующим в качестве базового метода дерево, было принято решение использовать параметры наилучшего единичного дерева с глубиной 8.

Описанным ранее способом, из графика был сделан вывод, что наиболее оптимальное значение количества деревьев равно 8.

В результате чего для наилучшего набора параметров получена точность 90%, F1 мера 87%.

### **SVM**

Алгоритм SVM (опорных векторов) – базируется на идее разделения выборки некоторой гиперплоскостью оптимальным образом. Алгоритм достаточно неустойчив к выбросам. Данный алгоритм требует указания ядра для разделяющей гиперплоскости.

Параметры использовались по умолчанию (было взято базовое ядро) так как оно давало точность не хуже, чем у других используемых моделей.

В результате чего для наилучшего набора параметров получена точность 87%, F1 мера 83%.

По сравнению с результатами предыдущих алгоритмов не удалось превзойти случайный лес и решающее дерево, однако, данный алгоритм все же надежнее, чем KNN.

### **Нейронная сеть**

Была построена нейронная сеть с одним скрытым слоем, содержащим 40 нейронов.

Описанным ранее способом, из графика был сделан вывод, что наиболее оптимальное количество эпох равно 300.

В результате чего для наилучшего набора параметров получена точность 86%, F1 мера 82%.

Нейронная сеть также не смогла превзойти предыдущие алгоритмы кроме KNN.

### **Стекинг**

Стекинг подразумевает параллельное обучение разнородных алгоритмов, на результатах работы которых будет обучаться мета алгоритм, прогнозирующий класс исследуемого объекта.

В процесс обучения применяется метод блендинга. Блендинг – это ансамблевый метод, в котором выборку заранее делят на 2 части. На первой половине учат базовые алгоритмы, затем обученным моделям подают 2 половину выборки, в результате чего получаем новый набор мета признаки, основанный на результатах работы базовых моделей. Далее на основе новой выборки обучаем мета модель.

В качестве базовых моделей были взяты решающее дерево, KNN, SVM, а в качестве мета модели – нейронная сеть. Параметры для базовых моделей были взяты те же, что и при использовании их отдельно, а именно: KNN  $k = 15$ , решающее дерево глубина 8, SVM базовое ядро. Мета модель представляет собой нейронную сеть, имеющую 1 скрытый слой из 6 нейронов. Ранее упомянутым способом было получено, что оптимальное число эпох равно 550.

В результате чего для наилучшего набора параметров получена точность 89%, F1 мера 86%.

Стекинг же в свою очередь, даёт результаты, примерно сравнимые, со случайным лесом и решающим деревом.

### **K-means**

В результате анализа было получено что оптимальное число эпох обучения при бинарном разбиении равно 600.

В результате чего для наилучшего набора параметров получена точность 51%, F1 меру 44%.

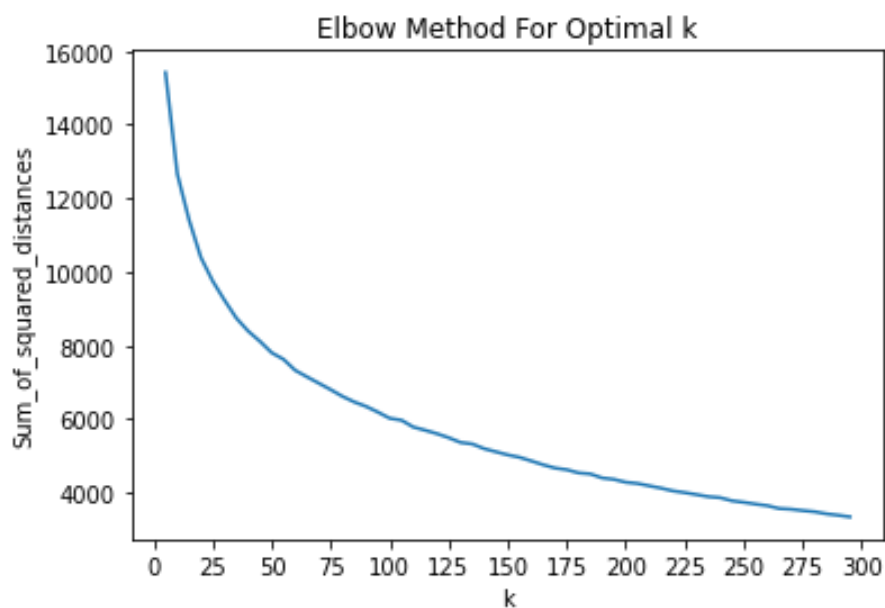
На данном этапе это самый худший результат среди всех представленных моделей.

Также был рассмотрен случай, при котором происходит разделение на оптимальное число кластеров.

Нам не всегда задано число кластеров, поэтому возникает вопрос, как определить число кластеров наилучшим образом. Существует множество методов, но один из самых точных и распространенных – это метод локтя.

Суть метода локтя заключается в том, чтобы многократно циклически выполнять алгоритм кластеризации, увеличивая количество кластеров и вычисляя для каждой итерации баллы кластеризации. Как правило, баллы вычисляются при помощи меры качества классификации, и выбирается то количество кластеров, где происходит наибольший скачок качества кластеризации.

Для определения оптимального количества кластеров пришлось прибегнуть к методу локтя, ниже представлен график зависимости суммы квадратов расстояний внутри каждого кластера от количества кластеров (рис. 2).



**Рис. 2. График зависимости суммы квадратов расстояний внутри каждого кластера от количества кластеров**

Было выбрано 50 кластеров, так как в этой точке, наблюдается резкое уменьшение скорости изменения суммы квадратов расстояний внутри кластеров.

На основании данного разбиения были рассчитаны доли вышедших на сделку броней по кластерам (рис. 3), где каждый столбец представляет собой кластер, в котором синим цветом изображаются брони, вышедшие на сделку, а красным – освободившие брони.

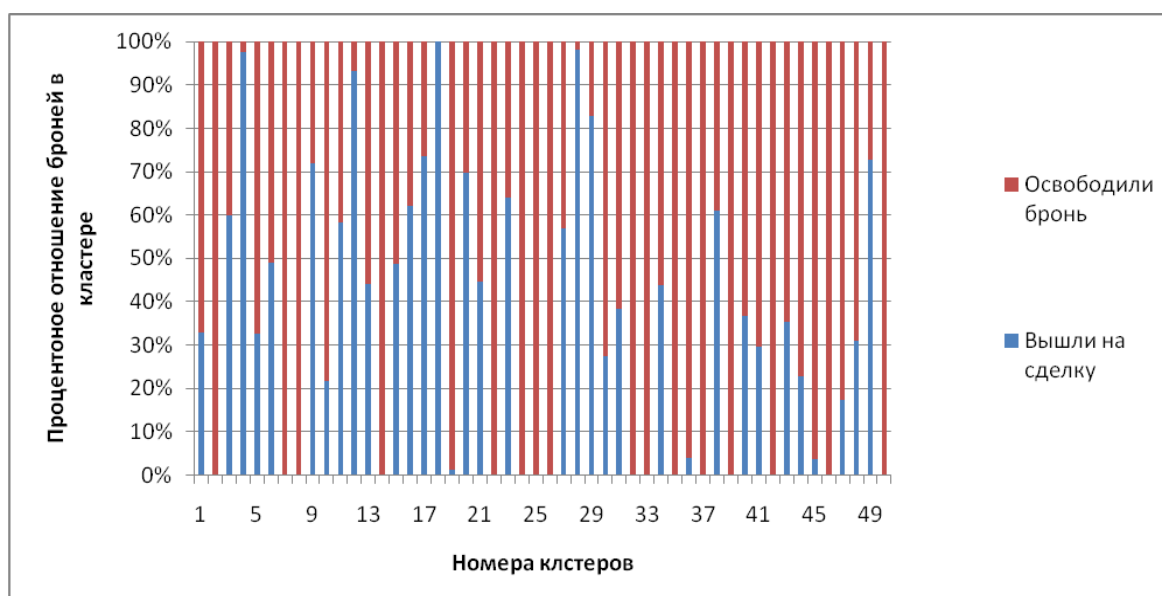


Рис. 3. Результат кластеризации.

По полученным кластерам можно давать прогноз о выходе брони на сделку с определенной вероятностью. Для этого необходимо определить объем вышедших на сделку броней в кластере, достаточный, чтобы отнесение объекта к этому кластеру считалось принятием его как вышедшего на сделку. Используя ранее описанный метод, для наилучшего количества кластеров и порогового отсечения была получена точность 80%, F1 мера 72%, что значительно лучше, чем при бинарной классификации, однако все же хуже, чем при использовании классификаторов и ансамблевых методов.

### Вывод

В результате проведенного исследования были получены модели и их оценки точности. Была построена таблица сравнения результатов обучения алгоритмов (таблица 1).

Анализируя данную таблицу, можно заметить, что наиболее точный прогноз даёт случайный лес, точность которого 90% и F1 мера 87%.

Таблица 1. Сравнение результатов обучения алгоритмов

	Точность	F1 мера
<b>KNN</b>	82%	76%
<b>Решающее дерево</b>	88%	85%
<b>Случайный лес</b>	90%	87%
<b>SVM</b>	87%	83%
<b>Нейронная сеть</b>	86%	82%
<b>Стекинг</b>	89%	86%
<b>K-means (бинарное разбиение)</b>	51%	44%
<b>K-means (мультикластерное разбиение)</b>	80%	72%

Была построена тепловая карта прогнозирования моделей на тестовой выборке, (таблица 2), где зеленым обозначено наибольшее значение среди всех методов, а красным – наименьшее.

Анализируя данную таблицу можно заметить, что наиболее распространенная ошибка – это ошибочное отнесение брони к группе вышедших на сделку. Это обусловлено тем, что в данных преобладают брони, не вышедшие на сделку, поэтому алгоритмы чаще всего ошибаются в прогнозировании данного вида броней.

**Таблица 2. Тепловая карта прогнозирования моделей на тестовой выборке**

	KNN	Решающее дерево	Случайный лес	SVM	Нейронная сеть	Стекинг	K-means (2 кластера)	K-means (50 кластеров)
<b>Правильно определенные не вышедшие на сделку брони</b>	251	264	267	260	256	259	148	250
<b>Ошибочно отнесенные брони к группе вышедших на сделку</b>	47	34	31	38	42	39	150	48
<b>Ошибочное отнесенные брони к группе не вышедших на сделку</b>	37	19	13	21	22	12	79	45
<b>Правильно определенные вышедшие на сделку брони</b>	133	151	157	149	148	158	91	125

### Библиографический список

1. Кан К.А. Нейронные сети. Эволюция ЛитРес: Самиздат 2018.
2. Метрики в задачах машинного обучения [Электронный ресурс] URL: <https://habr.com/ru/company/ods/blog/328372/> (дата обращения 20.01.2022).
3. Ensemble methods: bagging, boosting and stacking [Электронный ресурс] URL: <https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205> (дата обращения 14.02.2022).
4. Случайный лес (Random Forest) [Электронный ресурс] URL: <https://dyakonov.org/2016/11/14/%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9-%D0%BB%D0%B5%D1%81-random-forest/> (дата обращения 14.02.2022).
5. Stop One-Hot Encoding Your Categorical Variables [Электронный ресурс] URL: <https://medium.com/analytics-vidhya/stop-one-hot-encoding-your-categorical-variables-bbb0fba89809> (дата обращения 14.02.2022).
6. МЕТОДЫ DATA MINING: ОБЗОР И КЛАССИФИКАЦИЯ [Электронный ресурс] URL: <https://hsbi.hse.ru/articles/metody-data-mining-obzor-i-klassifikatsiya/> (дата обращения 14.02.2022).
7. Что такое обработка данных в машинном обучении [Электронный ресурс] URL: <https://deepcloud.ru/articles/chto-takoe-obrabotka-dannyh-v-mashinnom-obuchenii/> (дата обращения 14.02.2022).
8. Ансамблевые методы: беггинг, бустинг и стэкинг [Электронный ресурс] URL: <https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/> (дата обращения 14.02.2022).
9. Моем датасет: руководство по очистке данных в Python [Электронный ресурс] URL: <https://proglib.io/p/moem-dataset-rukovodstvo-po-ochistke-dannyh-v-python-2020-03-27> (дата обращения 14.02.2022).
10. Федотов М.Х. Предиктивная диагностика оборудования тепловоза на основе интеллектуального анализа данных
11. Мисюра В.В., Пирко Д.В. АНСАМБЛИРОВАНИЕ РЕГРЕССИОННЫХ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ НА ПРИМЕРЕ ЗАДАЧИ HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES С РЕСУРСА KAGGLE.COM / МОЛОДОЙ ИССЛЕДОВАТЕЛЬ ДОНА, 2019. – 54-62 с.
12. Шадрин А.Ю., Городилов А.Ю. ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ПОВЕДЕНИЯ ИГРОКОВ В МНОГОПОЛЬЗОВАТЕЛЬСКИХ ONLINE-ИГРАХ / МАТЕМАТИКА И МЕЖДИСЦИПЛИНАРНЫЕ ИССЛЕДОВАНИЯ / ПГНИУ. – Пермь 2019. 427-431 с.
13. Коваленко М.В., Самарев Р.С. МЕТОДЫ ОЧИСТКИ ДАННЫХ РЕЗЮМЕ / ТЕХНОЛОГИИ ИНЖЕНЕРНЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ, 2021. – 85-91 с.
14. Подготовка данных для расширенного машинного обучения [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/azure/architecture/data-science-process/prepare-data> (дата обращения 14.02.2022).



## **ANALYSIS OF THE PROBABILITY OF GOING TO A DEAL FOR EACH RESERVATION TO PURCHASE REAL ESTATE**

*Vasilevykh Mikhail G.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, vasilevyh00@gmail.com

The article presents a description of the development of a system for predicting the probability of entering a booking deal, based on data on the bookings of a construction company. Forecasting the probability of booking a deal can be useful for sales managers, as well as people responsible for planning economic indicators. The data were preliminarily cleaned and divided into two sets, test and training. Next, various forecasting methods were considered, such as classification, single models and ensembles, and clustering. Selected models are knn, svm, k-means, random forest, decision tree, stacking, neural network. Methods for evaluating models were also chosen, on the basis of which a comparative analysis was carried out, as a result of which it turned out that the most accurate model was a random forest.

Keywords: artificial intelligence, prediction, random forest, bagging, staking, boostrapping, sampling, knn, svm, k-means, random forest, decision tree, neural network

## «СТРОЙКА» – КОМПЬЮТЕРНАЯ ИГРА ДЛЯ ЗНАКОМСТВА С ПАРАЛЛЕЛЬНЫМ ПРОГРАММИРОВАНИЕМ

*Воронова Кристина Дмитриевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, christinavoronova12@gmail.com

*Плаксин Михаил Александрович*

Национальный исследовательский университет Высшая школа экономики (Пермский филиал), 614070, Россия, г. Пермь, ул. Студенческая, 38, mapl@list.ru

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15

Бурное развитие параллельных вычислительных технологий делает актуальным включение пропедевтики параллельных вычислений в школьный курс информатики. Поскольку эта тема еще не вошла в школьную программу, сделать это можно через внеурочную деятельность, в частности, через Интернет-конкурсы. С 2013 г. задания на параллельные вычисления стали обязательной частью конкурса «ТРИЗформашка». Среди них всегда есть задания на исполнение и составление алгоритмов для игровых параллельных исполнителей. В статье описывается один из таких исполнителей – «Стройка». В течение нескольких лет он использовался в конкурсе в безмашинном варианте. В 2022 г. появилась его компьютерная версия.

Ключевые слова: информатика, начальная школа, средняя школа, методика преподавания, параллельное программирование, параллельные вычисления, ТРИЗформашка.

### **Введение**

В связи с бурным развитием computer science параллельные вычисления становятся неотъемлемой частью современных технологий на различных уровнях (многоядерные процессоры, многопроцессорные системы). Это делает актуальным включение пропедевтики параллельных вычислений в школьный курс информатики. Но в настоящее время в школьной программе для этого не предусмотрены часы, отсутствуют методические материалы.

Одним из способов решения проблемы является включение заданий на параллельные вычисления в программы различных конкурсов по информатике, которые проводятся в рамках внеурочной деятельности. В качестве такового конкурса с 2013 г. выступает конкурс «ТРИЗформашка» – ежегодный межрегиональный Интернет-конкурс по информатике, системному анализу и ТРИЗ (теории решения изобретательских задач) для школьников и студентов. [1, 2] Весной 2022 г. конкурс состоялся в 22-й раз. Возраст участников – от первого класса до четвертого курса. Среднее количество команд – около 70, максимальное – 202 (более шестисот человек). Сайт конкурса [www.trizformashka.ru](http://www.trizformashka.ru).

Данная статья посвящена описанию одного из параллельных исполнителей, использовавшихся на конкурс «ТРИЗформашка» – компьютерной игры «Стройка».

## История исполнителя «Стройка»

Впервые исполнитель «Стройка» (тогда еще бескомпьютерный) появляется в конкурсе «ТРИЗформашка» в 2013 г. Программных исполнителей тогда еще не было. Задания для «Стройки» были даны двух типов: на выполнение заданного алгоритма и на составление алгоритма для построения заданного «здания».

Исполнитель «Стройка» действовал по следующим правилам.

Стройку ведут 3 бригады, работающие одновременно. Все они совершенно равнозначны: могут выполнять одни и те же действия за одно и то же время. Можно считать, что работа ведется тактами. За каждый такт каждая бригада выполняет одно действие.

Строительство ведется из балок. Каждая балка имеет уникальный номер.

Каждая балка может быть уложена горизонтально или установлена вертикально.

Уложить балку можно на землю между двумя указанными точками, краями на две вертикальные балки, серединой на вершину вертикальной балки.

Установить балку можно на землю (в указанную точку), на вершину другой стоящей балки, на середину, на левый или правый край лежащей балки.

Точки на земле, в которые может быть установлена балка, обозначены латинскими буквами. Расстояние между соседними точками равно длине балки.

Так как параллельные алгоритмы на конкурсе «ТРИЗформашка» появились впервые, задания были простейшими. В результате выполнения алгоритма, предложенного участникам конкурса, должно было получиться «здание», изображенное на рис. 1.

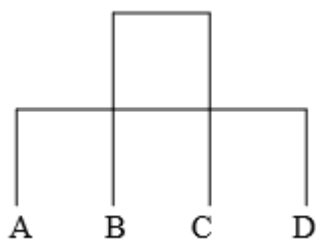


Рис. 1. Пример задания для исполнителя «Стройка»

На рис. 2 представлены варианты результатов выполнения этого алгоритма, присланные участниками конкурса.

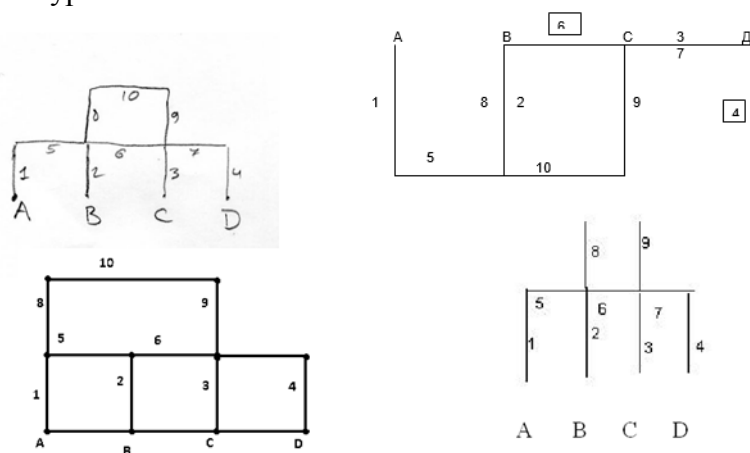


Рис. 2. Ответы участников конкурса «ТРИЗформашка-2013»

В 2014 г. на конкурсе появилась компьютерная игра «Танковый экипаж» [3]. Для исполнителя «Стройка» осталось только задание на выполнение заданного алгоритма. Но по сравнению с 2013 г. задание усложнилось.

В 2015 г. «Стройка» сохранила задание на выполнение алгоритма, но сложность задания существенно увеличилась. Вместо одного были даны три алгоритма. Все они выдавали один и тот же результат (описывали процесс возведения одного и того же строения). Но это факт участнику конкурса еще надо было установить. Один из алгоритмов содержал ошибки. Надо было определить ошибочный алгоритм и указать ошибочные команды. После этого надо было сравнить два оставшихся алгоритма по эффективности с точки зрения сроков и стоимости строительства.

В следующем 2016 г. «Стройка» становится ещё более сложной. Балку в этой версии можно было расположить не только горизонтально или вертикально, но и наклонно. Нижним концом наклонная балка должна была опираться на край или середину горизонтальной балки, верхний край вертикальной балки, стык наклонных и/или вертикальных балок. Верхний конец наклонной балки мог опираться на край горизонтальной балки, верхний край вертикальной балки или верхний край другой наклонной балки. Верхний стык двух наклонных балок мог служить опорой при установке вертикальных и горизонтальных балок.

На рис. 3 приведен пример здания, построенного с использованием наклонных балок. Это здание было результатом исполнения алгоритма, который был задан участникам конкурса «ТРИЦформашка-2016».

Наклонные балки значительно повышали разнообразие «зданий», которые могли быть результатами конкурсных алгоритмов. Но их применение существенно усложняло систему команд. Количество команд возросло втрое, с шести до девятнадцати. Десять из них были предназначены для установки наклонных балок и были значительно более громоздки и сложны для понимания, чем для горизонтальных или вертикальных. Вместо «Уложить (N) серединой на (M)» или «Установить (N) слева на (M)» для горизонтальной и вертикальной балок для наклонной приходилось писать «Установить (N) вправо вверх на стык (M, N)», «Установить (N) влево вверх на середину(M)». Все это привело к тому, что в последующих конкурсах исполнитель «Стройка» не участвовал. Вернулся он только в 2022 г уже в виде компьютерной игры.

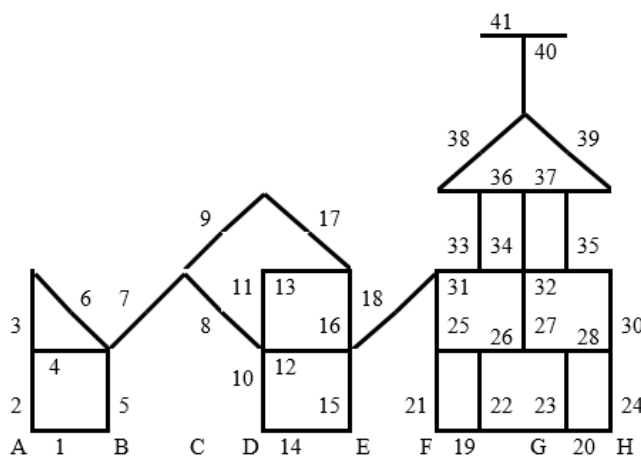


Рис. 3. Пример задания конкурса «ТРИЦформашка-2016»

### Компьютерная игра «Стройка»

Для компьютерной реализации данного исполнителя была выбрана версия без наклонных балок. Система команд повторяет версии «Стройки» 2013-2015 годов. Но задания, которые нужно выполнить участнику, также изменили тип. Теперь это задания не на исполнение заданного алгоритма, а на составление алгоритма для программного исполнителя.

Игра состоит из двух программ: конструктора заданий и конструктора алгоритмов.

Конструктор заданий позволяет вручную прорисовать здание, которое должно быть результатом выполнения алгоритма, составленного участниками конкурса. Схема этого здания служит подсказкой для игрока в процессе отладки алгоритма и позволяет оценить его правильность: указать балки, установленные верно, не верно или не установленные вообще.

При запуске конструктора алгоритмов открываются два окна: окно составления алгоритма и окно «строительной площадки», где происходит пошаговая отрисовка строящегося здания, по мере выполнения алгоритма.

На рис. 4 изображено окно составления алгоритмов. Оно рассчитано на три бригады. Команды выбираются из меню, их параметры (номера балок и точек) вводятся с клавиатуры.

Первая бригада					Вторая бригада					Третья бригада				
№ балки	Действие	I номер	II номер		№ балки	Действие	I номер	II номер		№ балки	Действие	I номер	II номер	
1	положить на землю между точками	A	B		2	положить на землю между ...	C	D		1	пауза			
2	поставить на левый край балки	1			3	поставить на левый край б...	2			2	поставить на правый кра...	2		
3	поставить на правый край балки	1			4	положить между балками	4	5		3	положить серединой на б...	6		
4	поставить на среднюю балки	3			5	поставить на левый край б...	9			4	поставить на правый кра...	9		
5	положить между балками	7			14	положить между балками	10	12		5	поставить на правый кра...	11		
7	положить серединой на балку													
8	положить на землю между точками													
	пауза													

Рис. 4. Окно алгоритма игры «Стройка»

На рис. 5 показано окно «строительной площадки». В нем расположено меню для загрузки задания, загрузки и сохранения алгоритма, управления выполнением алгоритма (целиком или пошагово), а также ползунки, позволяющие менять размер изображения строительной площадки и скорость анимации при выполнении алгоритма.

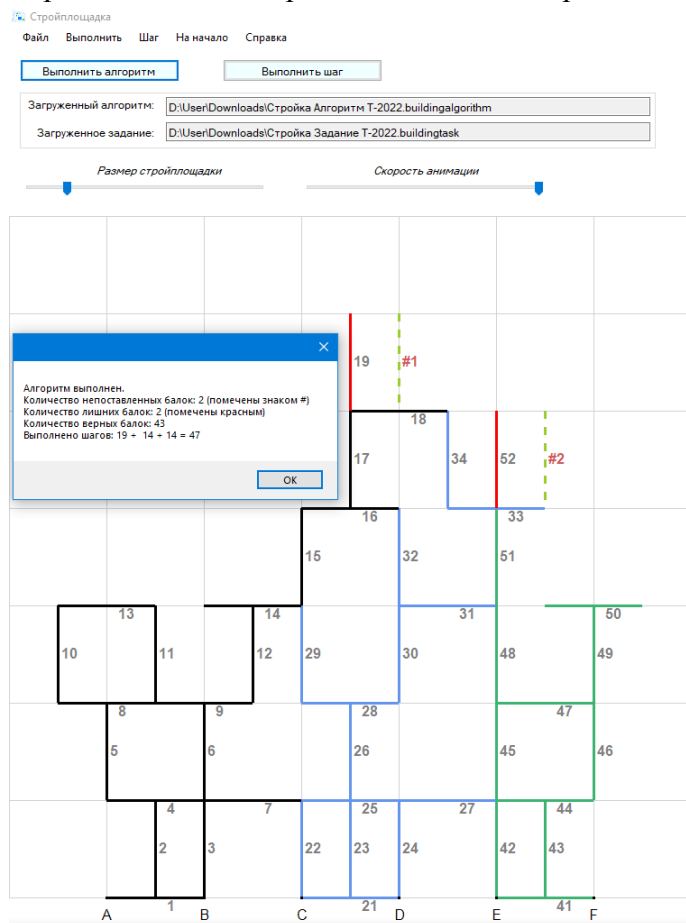


Рис. 5. Окно строительной площадки игры «Стройка»

## Заключение

Игра «Стройка» была использована в конкурсе «ТРИЗформашка-2022». Участники конкурса оценили интересность и полезность игры на 9 баллов из 10. В мае 2022 г. Игра была представлена на Международной научно-методической конференции в Минске [4].

## Библиографический список

1. *Плаксин М.А.* О пропедевтике параллельных вычислений в школьной информатике. //Информатика и образование. 2016. № 10 (279). С. 27-36.
2. *Плаксин М.А., Иванова Н.Г., Русакова О.Л.* Набор заданий для знакомства с параллельными вычислениями в конкурсе «Тризформашка». В сборнике «Преподавание информационных технологий в Российской Федерации. Материалы Тринадцатой открытой Всероссийской конференции. Пермский государственный национальный исследовательский университет. 2015. С. 232-234.
3. *Кучев А.Д., Плаксин М.А.* Пропедевтика параллельных вычислений в школьной информатике. Компьютерная игра «Танковый экипаж». //Информатика в школе. №9 (122), ноябрь 2016. С.42-48.
4. *Воронова К.Д., Плаксин М.А.* Знакомство с параллельными вычислениями в рамках дистанционного конкурса «ТРИЗформашка-2022». // Дистанционное обучение – образовательная среда XXI века : материалы XII Междунар. науч.-метод. конф. (Республика Беларусь, Минск, 26 мая 2022 года). С. 163.

## "CONSTRUCTION" IS A COMPUTER GAME TO INTRODUCE PARALLEL PROGRAMMING

*Voronova Kristina D.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, christinavoronova12@gmail.com

*Plaksin Mikhail A.*

National Research University Higher School of Economics (Perm branch), 614070, Russia, Perm, st. Student, 38, mapl@list.ru

Perm State University, 15, Bukireva st., Perm, 614990, Russia

Abstract. The rapid development of parallel computing technologies makes it urgent to include propeaedeutics of parallel computing in the school computer science course. Since this topic is not yet included in the school curriculum, it can be done through extracurricular activities, in particular, through Internet contests. Since 2013, parallel computing tasks have become a compulsory part of the "TRIZformashka" contest. Among them, there are always tasks for executing and composing algorithms for game parallel performers. This article describes one of such performers – "Construction". For several years, it was used in the contest in a machine-less version. In 2022 its computer version appeared.

Keywords: computer science, elementary school, high school, teaching methodology, parallel programming, parallel computing, TRIZformashka.

## ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОНЛАЙН-КОНСУЛЬТИРОВАНИЯ

*Главатских Ксения Елизаровна*

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

В работе рассмотрена проблема доступности бесплатных и эффективных систем взаимодействия клиентов и менеджеров небольших компаний с помощью виджета чата на их сайт. В результате анализа были определены требования к информационной системе, на основании которых она была спроектирована и разработана. Область применения разрабатываемой системы – самозанятость, микробизнес и малый бизнес. В этом случае управляющим лицам важна доступность используемых средств, а также оперативность в общении с клиентами, с чем может помочь кроссплатформенная система мгновенного обмена сообщениями Telegram, имеющая бесплатное и свободное распространение. Система позволяет общаться менеджерам и клиентам привычными для них способами. Первый отправляет и получает сообщения в Telegram, а второй видит всю переписку привычным окном чата на веб-странице. У представителя компании есть возможность войти не только под паролем менеджера, но и администратора, которому доступна настройка интерфейса чата и конфигурации системы. Клиент имеет возможность отправлять как текстовые сообщения, так и файлы. Система в изначальном варианте предлагает посмотреть демоверсию, а уже после настроить её у себя, устанавливая при этом файлы исходного кода на собственный домен. Работа будет интересна людям, работающим в сфере онлайн консультирования, а также тем, кому необходим простой способ общения с клиентами на сайте с небольшим охватом с помощью мессенджера Telegram и желающим получить self-hosted и open source решения, исключающие внезапное прекращение обслуживания предлагаемой услуги.

Ключевые слова: онлайн-консультирование, виджет чата, чат Telegram, чат-бот, веб-сайт, онлайн-чат.

Удовлетворение потребностей клиентов является приоритетом номер один для многих компаний. Особенно для предприятий, которые зависят от постоянного дохода от удержания клиентов.

Хороший способ достичь этой цели — использовать виджет чата на вашем веб-сайте, который позволяет вам либо общаться с посетителями веб-сайта в режиме реального времени, чтобы ответить на их вопросы как можно скорее, либо использовать бота (ИИ) для ответа на их вопросы и/или получить дополнительный ответ от ваших человеческих команд.

Значимость создания чата онлайн-консультирования заключается в том, что клиент получит гибкую настройку пользовательского интерфейса и основных функций, требующихся при использовании данного чата, а также исходный код – основу, с которой в последующем можно работать.

На рынке ИС достаточно много подобных реализаций, которые разрабатывались не один год и конкурировать с ними очень сложно, в большинстве случаев они пользуются спросом у твердо стоящих на ногах компаний. Однако можно создать не менее эффективную

систему, ориентированную на небольшие организации, которым важны не столько функциональные показатели посетителей сайта, как быстрота и качество удовлетворения новых клиентов, что влечет за собой повышение потенциальных потребителей.

В связи с этим было принято решение использовать в качестве платформы взаимодействия с пользователем сайта один из популярных мессенджеров, который уже адаптирован под настольные ПК, мобильные устройства и планшеты.

Информационную систему онлайн-консультирования представим в виде веб-сервера, состоящего из БД и API и платформы одного из мессенджеров, которую мы настроим для менеджеров через веб-сервер.

Для написания API был выбран язык программирования для web – PHP. Он хорошо документирован, имеет встроенные функции для работы с HTTP, прост в освоении и отладке. Также он имеет высокую степень переносимости и хорошо работает под управлением различных ОС. Интерфейс нашей системы будет разработан при помощи языка разметки HTML и CSS. Связь интерфейса с API будет осуществляться через скрипты, написанные на JavaScript. Также потребуется связь баз данных и SQL. Сценарий нашего использования субд – небольшое веб-приложение, которое не требует больших запросов и объемов данных, для чего подойдет простой в настройке и более гибкий MySQL.

В отчёте Digital на начало 2022 года определено, что в России насчитывается около 129 млн активных интернет-пользователей, то есть 89% от всего населения. В топ-5 самых популярных сервисов вошли WhatsApp (80% пользователей в возрасте 16-64 лет), VK (76,4%), Instagram (63,7%) – принадлежит компании Meta и является экстремистской организацией, деятельность которой запрещена в России, Telegram (50,8%) и TikTok (46,6%). [1]

В качестве платформы для общения нам необходим мессенджер – приложение, которое специализировано на быстром и надежном обмене сообщениями в режиме реального времени. Рассмотрим в качестве вариантов WhatsApp и Telegram. Ключевым фактором для выбора мессенджера является возможность беспрепятственного создания бота, который поможет в дальнейшем принимать и отправлять сообщения, – в Telegram доступны официальные и сторонние чат-боты, а WhatsApp же боты всё ещё находятся на стадии бета-тестирования. Помимо этого, важна кроссплатформенность – Telegram синхронизирует файлы и чаты, чтобы использовать одну учетную запись на разных устройствах, WhatsApp же не поддерживает такой опции: войдя в свою учетную запись с другого устройства, вы автоматически выйдете из своего предыдущего устройства. На основе этих критериев выбираем Telegram.

Создание веб-приложения, который работает в реальном времени, — достаточно непростая задача, так как нам необходимо задуматься, как мы будем отправлять данные с сервера клиенту. Технологии, позволяющие активно связываться, есть довольно давно и ограничены двумя общими подходами: клиентское извлечение либо серверное извлечение. Некоторые способы их реализации: Long/short polling (client pull), WebSockets (server push), Server-Sent Events (server push) [2].

Для разрабатываемого чат-приложения было выбрано использование HTTP/2 + SSE и WebSocket, между которыми можно будет выбрать. Поскольку SSE основан на HTTP, он естественно сочетается с HTTP/2 и может быть объединен, чтобы получить лучшее из обоих: HTTP/2 обрабатывает эффективный транспортный уровень на основе мультиплексированных потоков, а SSE предоставляет API для приложений. Итак, получаем мультиплексирование через HTTP/2 из коробки. Клиент и сервер информируются об обрыве соединения. Под-



держивая уникальный идентификатор с сообщениями, сервер может видеть, что клиент пропустил несколько сообщений, и отправлять список пропущенных сообщений при повторном подключении. Также WebSocket, характеризующийся полнодуплексной связью по одному TCP-соединению и постоянно поддерживающий соединения между клиентом и сервером, является альтернативой. Что использовать из предложенных вариантов, может решить сам пользователь при установке системы.

Разместить сервер с чат-ботом и тестируемым веб-сайтом необходимо либо на хостинге, либо на виртуальном сервере. Воспользуемся вторым вариантом в сочетании с CentOS Stream — это платформа разработки Linux, где члены сообщества с открытым исходным кодом могут вносить свой вклад в Red Hat Enterprise Linux в тандеме с разработчиками Red Hat. CentOS Stream получает небольшие, более регулярные по сравнению с CentOS, хорошо протестированные обновления. Остается выбрать веб-сервер, где размещаются программы и данные и который отвечает на клиентские запросы веб-страниц или других веб-служб. Распространенными веб-серверами являются Apache, Microsoft Internet Information Services (IIS) и Nginx. Второй вариант не подойдет для CentOS Stream. Если говорить о скорости и о том, сколько клиентов можно обслужить при высокой нагрузке, Nginx превосходит над Apache. Это делает Nginx значительно легче и отлично подходит для серверных ресурсов.

Мы реализуем на сайте виджет чата, бэкэндом для которого является мессенджер Telegram. Пользователи отправляют сообщение в чат, оно принимается нашим виртуальным сервером, где запускается бот, который в свою очередь направляет запрос на отправку сообщения на сервер Telegram. Информировается оператор и после ответа с помощью установленного веб-перехватчика (получаем обновления мы с помощью WebHook) получаем от сервера Telegram новое сообщения на наш виртуальный сервер. Это даёт сигнал боту о том, что произошло обновление в чате. Тогда бот отправляет сообщение в виджет чата.

Для более наглядного представления изобразим схему описанного выше на рисунке 1.

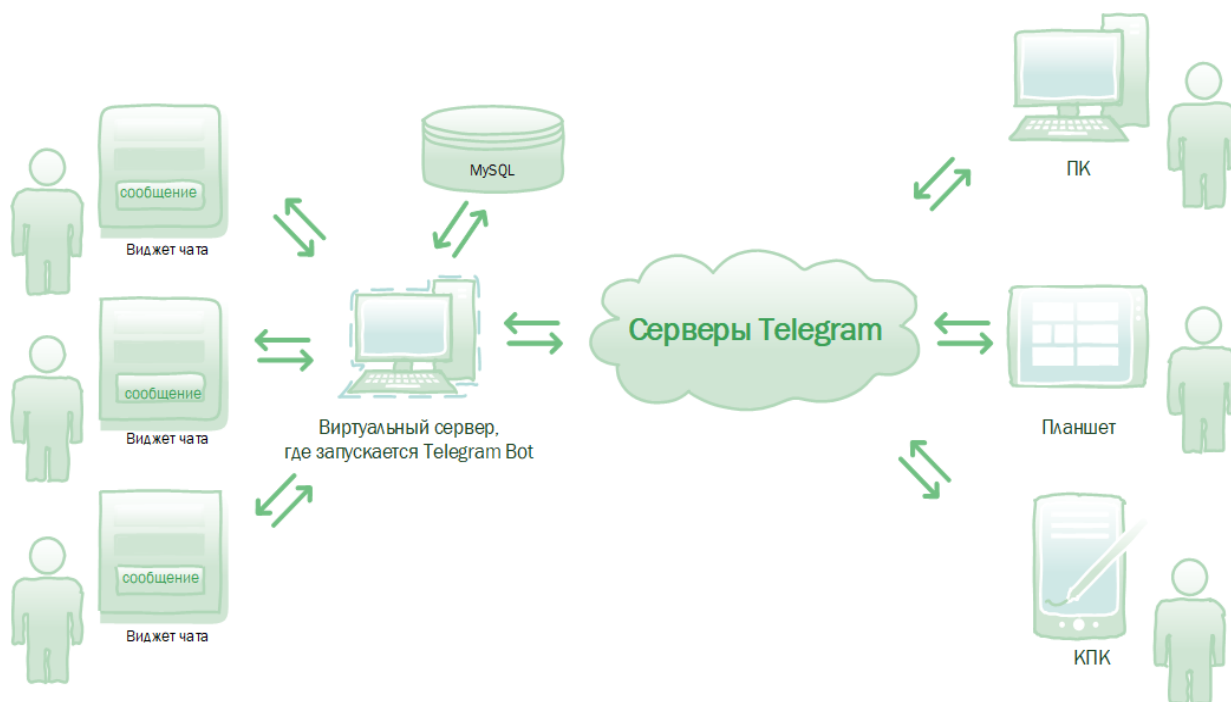


Рис. 1. Структура ИС

Существуют два способа принятия обновления для бота – `getUpdates()`-метод и `WebHook`. Входящие обновления (JSON-сериализованные объекты `Update`) сохраняются на сервере Telegram до того момента, пока бот не получит их, но они не хранятся более одних суток.

В случае использования метода `getUpdates()` приходится реализовывать в коде постоянное обращение этой команды на сервер с ботом, что делается в циклическом режиме, а это может привести к зависаниям и багам. `WebHook` также держит постоянное соединение, но отличается тем, что запросы отправляются только при свершившемся событии.

В системе будет 3 типа пользователей: оператор, менеджер операторов и пользователь виджета чата на сайте. Последнему предоставляется только общение в чате. Первому доступно также общение в чате, только не с виджета чата, а посредством мессенджера. Менеджер операторов имеет самый обширный доступ, ему предоставляются все возможности оператора, к которым добавляются настройка системы через чат-бота-настройщика.

Демоверсия веб-страницы с вариантом развернутого окна чата представлена на рисунке 2. В качестве кнопки можно разместить любую фигуру, указав `id` объекта из `html`-файла в файле с настройками виджета. Перед тем, как новый пользователь начнёт общение, ему необходимо будет нажать кнопку с согласием на использование `cookie`-файлов. В отсутствие менеджеров и администраторов пользователю приходит уведомление о том, что сейчас нет доступных консультантов. Дополнительной возможностью является отправка картинок и документов.

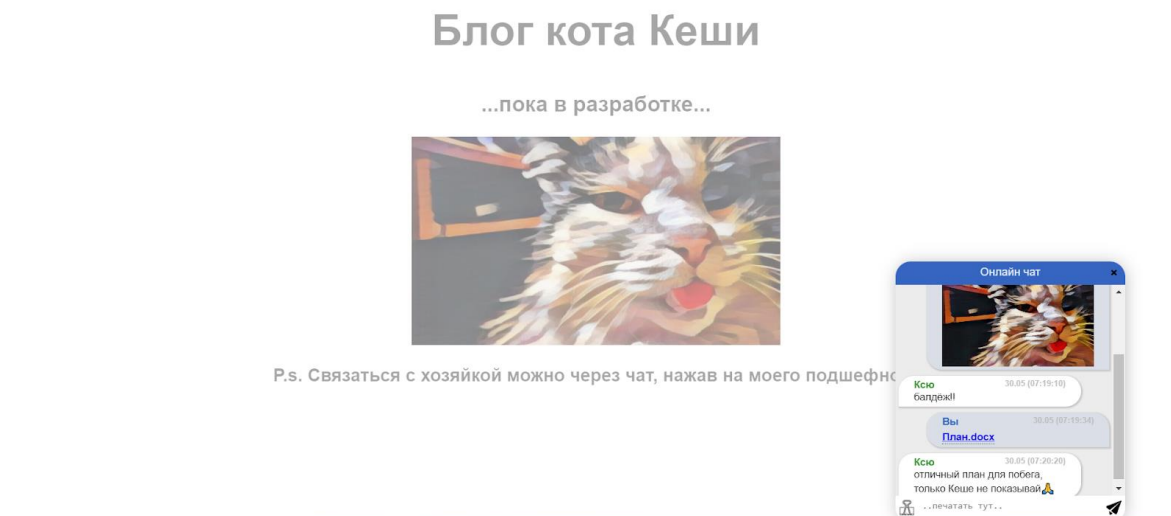


Рис. 2. Демоверсия веб-страницы с открытым окном чата

Для того, чтобы будущим установщикам виджета чата наглядно было понятно, с чем они будут работать, предоставим им возможность войти в чат-бот под паролями менеджера и администратора, которые будут указаны в сообщении после нажатия кнопки «Инфо». Здесь же кнопка ниже «Инструкция по установке» со держит требования и порядок действий в виде инлайн кнопок (рис. 3) для установки. Общаться с пользователем окна чата консультант может только при переходе в конкретный чат. Уведомление о новом клиенте приходит при нажатии кнопки о начале общения в виджете чата и при статусе «онлайн». При статусе «офлайн» уведомления приходят только от тех чатов, которые были выбраны.

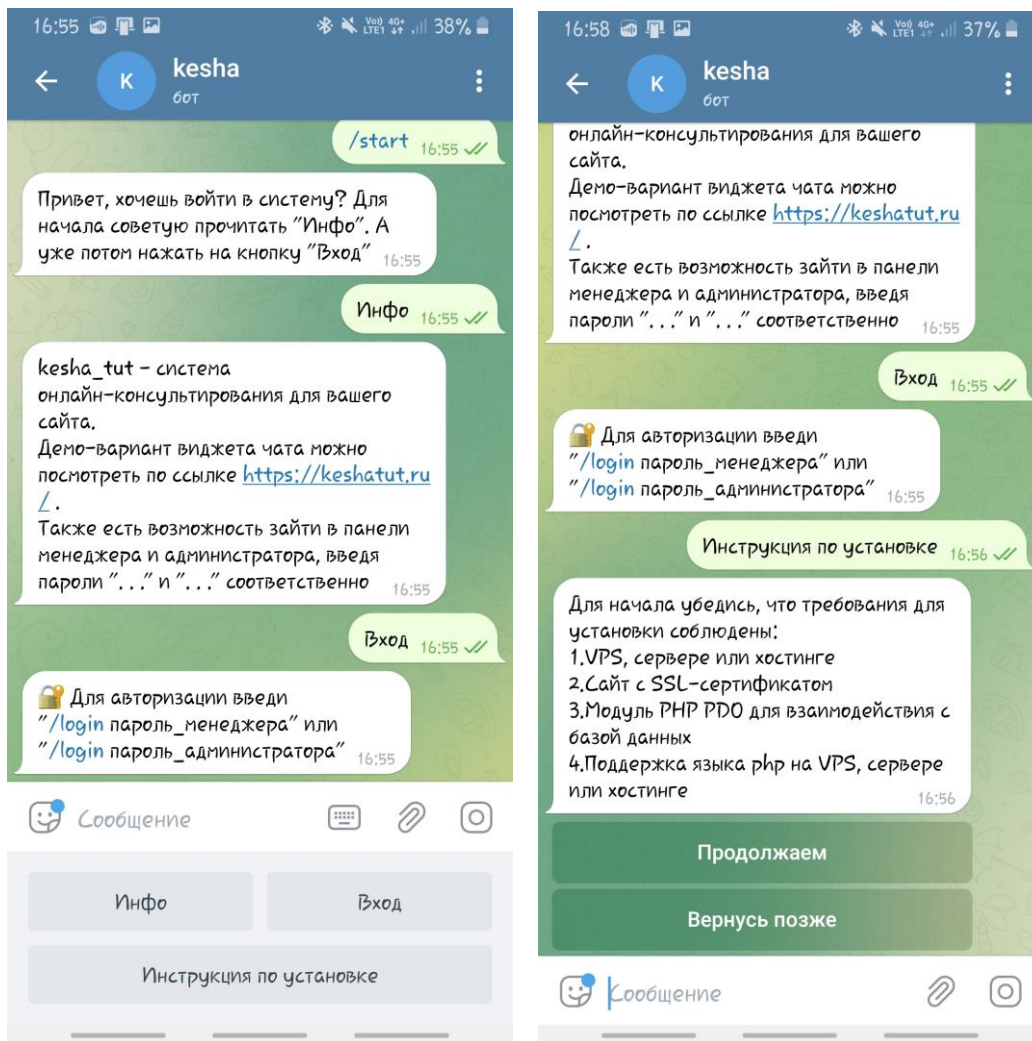


Рис. 3. Чат-бот в Telegram. Обычные кнопки и инлайн кнопки

Каждому менеджеру и администратору доступны кнопки «О нас», «Все команды», «Выход», «Онлайн», «Оффлайн», а также общие команды, список которых можно получить при нажатии «Все команды». Второму в качестве привилегий доступна «Настройка виджета»: «Конфигурационный файл», «Внешний вид виджета чата».

В результате проделанной работы создана ИС для онлайн консультирования с помощью мессенджера Telegram, которую можно установить на свой хостинг или сервер, следуя инструкции и попробовав перед этим демонстрационный вариант с полным функционалом.

### Библиографический список

1. Чуранов Е. Интернет в России в 2022 году: самые важные цифры и статистика. WebCanape [Электронный ресурс] URL: <https://www.web-canape.ru/business/internet-v-rossii-v-2022-godu-samy-e-vazhnye-cifry-i-statistika/> (дата обращения: 15.05.2022)
2. Bharathvaj Ganesan Polling vs SSE vs WebSocket— How to choose the right one. coderust.io [Электронный ресурс] URL: <https://codeburst.io/polling-vs-sse-vs-websocket-how-to-choose-the-right-one-1859e4e13bd9> (дата обращения: 17.05.2022)

## **DESIGN, DOCUMENTATION AND DEVELOPMENT OF INFORMATION SYSTEM FOR ONLINE CONSULTANCY**

*Glavatskikh Kseniya E.*

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

The paper considers the problem of the availability of free and effective systems for the interaction of customers and managers of small companies using a chat widget on their website. As a result of the analysis, the requirements for the information system were determined, on the basis of which it was designed and developed. The system allows managers and clients to communicate in the ways they are used to. The first sends and receives messages in Telegram, and the second sees all correspondence with the usual chat window on the web page. A company representative has the opportunity to log in not only with the manager's password, but also with the administrator's password, who can configure the chat interface and system configuration. The client has the ability to send both text messages and files. The system in its original version offers to watch the demo version, and after that, set it up for yourself, while installing the source code files on your own domain. The work will be of interest to people working in the field of online consulting, as well as those who need an easy way to communicate with clients on a site with a small coverage using the Telegram messenger.

Keywords: online consultancy, chat widget, Telegram chat, chat bot, website, online chat.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ ЕДИНОЙ ТОЧКИ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ

*Кармазин Владислав Игоревич, Кнутова Наталия Сергеевна*

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

В настоящее время практически у любой компании, учреждения или организации есть свои ресурсы. Например, электронные дневники в школах и вузах или электронные библиотеки, а также блоги и форумы. Имея множество таких ресурсов в рамках одной организации, пользователям этих сервисов приходится запоминать пароли каждой учетной записи, либо устанавливать одинаковые и, как правило, простые, а значит подверженные к атакам, таким как подбор по словарю. В тоже время появляется необходимость повторной аутентификации при переходе на каждый из ресурсов по отдельности. Еще одна проблема, связанная с наличием множества учетных записей пользователей на каждом ресурсе отдельно, заключается в нагрузке на системных администраторов. Например, при поступлении нового студента приходится регистрировать его на каждом ресурсе, а при его отчислении закрывать доступ, либо удалять учетную запись так же на каждом ресурсе. В связи с этим появилась технология единой точки авторизации. Её внедрение позволяет пользователям ресурса проходить процедуру авторизации единожды, даже при переходе на другие ресурсы в рамках одной системы, а также иметь одну общую учетную запись.

Ключевые слова: система единой точки авторизации пользователей, авторизация, аутентификация.

Существует множество готовых реализаций технологии единой аутентификации, которые можно легко интегрировать в свою систему. У них есть, как свои недостатки, так и преимущества. Из них были рассмотрены такие как: Ory Hydra, Avanpost FAM, IBM Security Access Manager и Oracle Access Manager.

Реализация этой технологии в корпоративных масштабах требует гораздо больших усилий, чем разработка учебной системы или решения для маленькой компании. Поэтому и существуют готовые решения систем единой авторизации, предназначенные для крупномасштабных корпоративных проектов.

На примере четырёх рассмотренных решений было отмечено, что все они имеют практически одинаковую структуру системы, наличие компонентов, отвечающих за определенный функционал, а различия обычно объясняются желанием добиться дополнительных возможностей и гибкости системы. Так же отдельные компании, например, Oracle, придумывают новые идеи вокруг системы единой авторизации, такие как Oracle Enterprise Single Sign-On например, который предлагает концепцию, в соответствии с которой пользователь вводит пароль единственный раз при авторизации в операционной системе своего рабочего компьютера. После этого в процессе работы он не замечает никаких других процедур авторизации в

используемых им приложениях и информационных ресурсах, поскольку Oracle Enterprise Single Sign-On берет на себя выполнение этих процедур.

Внедрение системы единой точки авторизации способствует решению следующих задач[1]:

- оптимизация работы администраторов, связанной с управлением доступом к ресурсам;
- повышение безопасности системы;
- облегчение менеджмента аккаунтов для пользователей;
- создание единых правил предоставления доступа вне зависимости от ресурсов.

За время существования рассматриваемой технологии, в её реализации было использовано множество различных протоколов. На данный момент системы единой авторизации пользователей преимущественно разрабатываются с использованием протокола OAuth2.0 и протокол SAML, но ранее также популярными были протоколы LDAP, Kerberos, OpenID.

SAML возник в ИТ-индустрии для упрощения процесса авторизации, когда пользователям необходимо было получить доступ к нескольким независимым веб-приложениям в разных доменах. До появления SAML была возможна технология единого входа, но она основывалась на файлах cookie, но эти файлы cookie работают только в одном домене. SAML, с другой стороны, представляет собой протокол взаимодействия между провайдером идентификации и сервис-провайдером, который перенаправляет пользователя к провайдеру идентификации тогда, когда в его браузере отсутствуют необходимые данные для авторизации.

Основным недостатком протокола SAML является его ограниченность областью веб-ресурсов, которая, несомненно, не исчерпывает всего множества информационных ресурсов, которыми обладает современное предприятие.

OAuth 2.0 – протокол авторизации, который позволяет предоставлять одной службе права доступа к пользовательским ресурсам другой службы. Протокол устраняет необходимость доверять приложению логин и пароль, а также позволяет предоставлять ограниченный набор прав, а не все сразу.

OAuth 2.0 определяет четыре варианта для получения токена доступа. Эти варианты называются грантами. Решение о том, какой из них подходит для вашего случая, зависит в основном от типа вашего приложения:

- Вариант кода авторизации: используется веб-приложениями, выполняющимися на сервере. Это также используется мобильными приложениями с использованием метода Proof Key for Code Exchange (PKCE).
- Неявный вариант с публикацией формы: используется приложениями, ориентированными на JavaScript (одностраничные приложения), которые выполняются в браузере пользователя.
- Вариант пароля владельца ресурса: используется приложениями с высоким уровнем доверия.
- Вариант учетных данных клиента: используется для связи между компьютерами.

В свою очередь, OpenID также не потерял актуальность, т.к. его последняя версия OpenID Connect основана на протоколе OAuth2.0 и в неё интегрированы все его возможности.[2]

В своей реализации было принято решение не выбирать OAuth как основу, а постараться спроектировать всё с нуля. Это обосновано тем, что позволит лучше понять, как работает

система во всех аспектах, начиная от самобытного способа взаимодействия сервера и клиента, заканчивая проработкой диаграмм.

Для реализации данной системы может подойти любой серверный язык программирования, например: PHP, Python, Java, C#, JS с использованием NodeJS.

Все языки имеют свои преимущества и свои недостатки, связанные с тем, какие имеют фреймворки, парадигмы и правила написания на языке.

Язык, в данном случае, это инструмент, и не так важен его выбор, сколько фундаментальное понимание работы системы. Поэтому выбор был остановлен на Java, как на языке программирования, который основан на объектно-ориентированной парадигме и имеет отличный веб-фреймворк Spring.

Для разработки прототипа внешнего вида интерфейса информационной системы была выбрана программа Photoshop. Это не совсем целевое приложение, так как его главное назначение – это редактирование фотографий. Однако Photoshop не ограничивает пользователя своими формами и функционалом, позволяя создавать любой прототип, однако уступает во времени его создания другим средствам.

Для хранения данных пользователей на сервере необходима база данных и система позволяющая управлять ей. Для рассмотрения была выбрана «большая четверка»: Oracle, MS SQL Server, MySQL, PostgreSQL. Из которой в последующем был выбран PostgreSQL. Именно эта СУБД является частым выбором в паре с Java. Также, проведя анализ был замечен факт, что многие компании использовавшие СУБД Oracle пытаются провести миграцию именно на PostgreSQL. Связано это больше с лицензионными соглашениями.

Для того чтобы контактировать с базой данных с помощью языка Java есть два варианта: JDBC или ORM (Hibernate).

Согласно [3], ключевым преимуществом Hibernate по сравнению с JDBC является тот факт, что разработчики могут писать Java-приложения на основе объектно-ориентированных принципов и лучших практик, не беспокоясь о семантике базы данных. В результате разработка может протекать быстрее, особенно если разработчики программного обеспечения не имеют четкого представления о SQL и реляционных базах данных. Кроме того, поскольку взаимодействие между базой данных и Java-программой осуществляется через тщательно протестированную и надежную структуру, вы, скорее всего, увидите снижение количества ошибок на уровне отображения базы данных.

Подводя итоги, можно сказать, что система единой точки авторизации является востребованной в наше время, а также имеет большое количество подходов при её реализации, являясь очень гибкой технологией, при этом, не изменяя свою основную задачу.

### **Библиографический список**

1. Юданов Ф.Н., Федотов А. М., Сейткасым Р.С. Технологии единой авторизации и организации единой точки доступа к информационным ресурсам сети.
2. Authentication Protocols: LDAP vs Kerberos vs OAuth2 vs SAML vs RADIUS [Электронный ресурс] URL: <https://www.getkisi.com/blog/authentication-protocols-overview>
3. Difference between JDBC and Hibernate in Java [Электронный ресурс] URL: <https://www.geeksforgeeks.org/difference-between-jdbc-and-hibernate-in-java/>

## DESIGNING AND DOCUMENTING SINGLE SIGN-ON TECHNOLOGY

*Karmazin Vladislav I., Knutova Nataliya S.*

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

Currently, almost any company, institution or organization has its own resources. For example, electronic diaries in schools and universities or electronic libraries, as well as blogs and forums. Having many such resources within the same organization, users of these services have to remember the passwords of each account, or set the same and, as a rule, simple, and therefore susceptible to attacks, such as dictionary matching. At the same time, there is a need for re-authentication when switching to each of the resources separately. Another problem associated with the presence of multiple user accounts on each resource separately is the load on system administrators. For example, when a new student enters, they have to register him on each resource, and when he is expelled, close access or delete the account on each resource as well. In this regard, the technology of a single point of authorization has appeared. Its implementation allows users of the resource to go through the authorization procedure once, even when switching to other resources within the same system, and also have one common account.

Keywords: system of a single point of user authorization, authorization, authentication.



## РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ РАСЧЕТА МЕТРИК КОДА НА ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ЯЗЫКЕ ПРОГРАММИРОВАНИЯ

*Корзников Артем Олегович, Дацун Наталья Николаевна*

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

Работа посвящена автоматизации получения формальных характеристик программного кода. Целью является создание приложения для получения значений метрик кода программных продуктов на языке объектно-ориентированной парадигмы (ООП). Рассмотрены подходы к расчету метрик и существующие приложения, предоставляющие возможность их вычисления. На основании этого анализа принято решение об использовании набора метрик, характеризующих программный код в целом с помощью 16 метрик и с точки зрения ООП при использовании 11 метрик. С помощью форм Бэкуса-Наура выполнено формальное описание подмножества языка ООП C#, которое определяет структуры языка, допустимые в коде измеряемых компилируемых программ. Выполнено проектирование и реализация на языке C# приложения расчета метрик кода на объектно-ориентированном языке программирования, которые в дальнейшем могут быть использованы для комплексной оценки свойств программных продуктов. Тестирование приложения выполнено на различных проектах, включая код разработанного приложения.

Ключевые слова: метрика, программа, расчет метрик, объектно-ориентированная парадигма, объектно-ориентированный язык, программный код, проектирование приложения, реализация приложения, тестирование приложения, оценка свойств программы.

В настоящее время сохраняется и усиливается тенденция всеобщей информатизации. Электронные устройства прочно закрепились во всех сферах жизни современного общества. Но для обеспечения эффективной, корректной, надежной работы программного обеспечения и возможности сопровождения, необходимы усилия целой команды разработчиков. Процесс разработки не является детерминированным, поэтому весьма сложен в осуществлении. Одним из способов формализации может быть получение значимых показателей, на основании которых, специалист может обратить внимание на недостатки программы и улучшить ее.

Кроме того, процесс создания программного продукта отличается от большинства отраслей производства. Чтобы определить экономические характеристики проекта, недостаточно знать трудоемкость и усредненную производительность труда программиста. Они находятся в нетривиальной зависимости от объема проделанной работы и, зачастую, сложность определяется неверно.

На данный момент метрики являются неотъемлемой частью практики программной инженерии. Отраслевые стандарты, например, ISO 9000 [1] и отраслевые модели, такие как СММ [2] и СММІ [3], также включают показатели и измерения.

Метрика программного продукта – это численная мера, которая отражает показатель некоторого свойства программного продукта [4]. Обычно с ее помощью производится оценка характеристик некоторого участка кода.

Работа была проделана с целью предоставить разработчикам и менеджерам проектов инструмент для выявления недостатков в разрабатываемой системе и комплексной оценки ее свойств. С помощью реализованного приложения были получены метрики для его кода и кода VibeCheck, программы, реализующей проверку форматирования документов по ГОСТ.

На основании результатов анализа существующих решений можно сделать выводы:

- программ, позволяющих выполнять аналогичные или похожие задачи, небольшое количество (MS Visual Studio, Source Stat, NLOC),
- все программы, решающие аналогичные или похожие задачи, анализируют узкий круг метрик (3-6 метрик).

Был сформирован набор метрик, в котором используются технические метрики – «показатели, служащие для определения того, насколько хорошо структурирован программный код, адекватно программное обеспечение, полна, корректна и актуальна документация» [5].

В этот набор вошли стандартно используемые метрики кода:

- количество физических строк кода,
- количество логических строк кода,
- оценка стилистики программы.

Также в набор включены метрики Холстеда, являющиеся независимыми от языка программирования и его парадигмы:

- длина программы,
- словарь программы,
- объем программы,
- уровень качества программирования,
- трудоемкость кодирования программы,
- теоретическая длина программы,
- теоретический словарь программы,
- теоретический объем программы,
- оценка уровня качества программирования,
- сложность,
- работа по программированию,
- время программирования,
- оценка количества ошибок.

Для оценки программы с точки зрения объектно-ориентированной парадигмы были взяты метрики Чидамбера и Кемерера:

- взвешенные методы на класс,
- высота дерева наследования,
- количество дочерних классов,
- сцепление между классами,
- отклик класса,
- число аргументов метода,
- недостаток связности в методах.

Также для этой цели был добавлен ряд метрик Лоренца и Кидда:

- размер класса,

- высота дерева наследования,
- переопределяемые подклассом операции,
- добавляемые подклассом операции,
- индекс специализации.

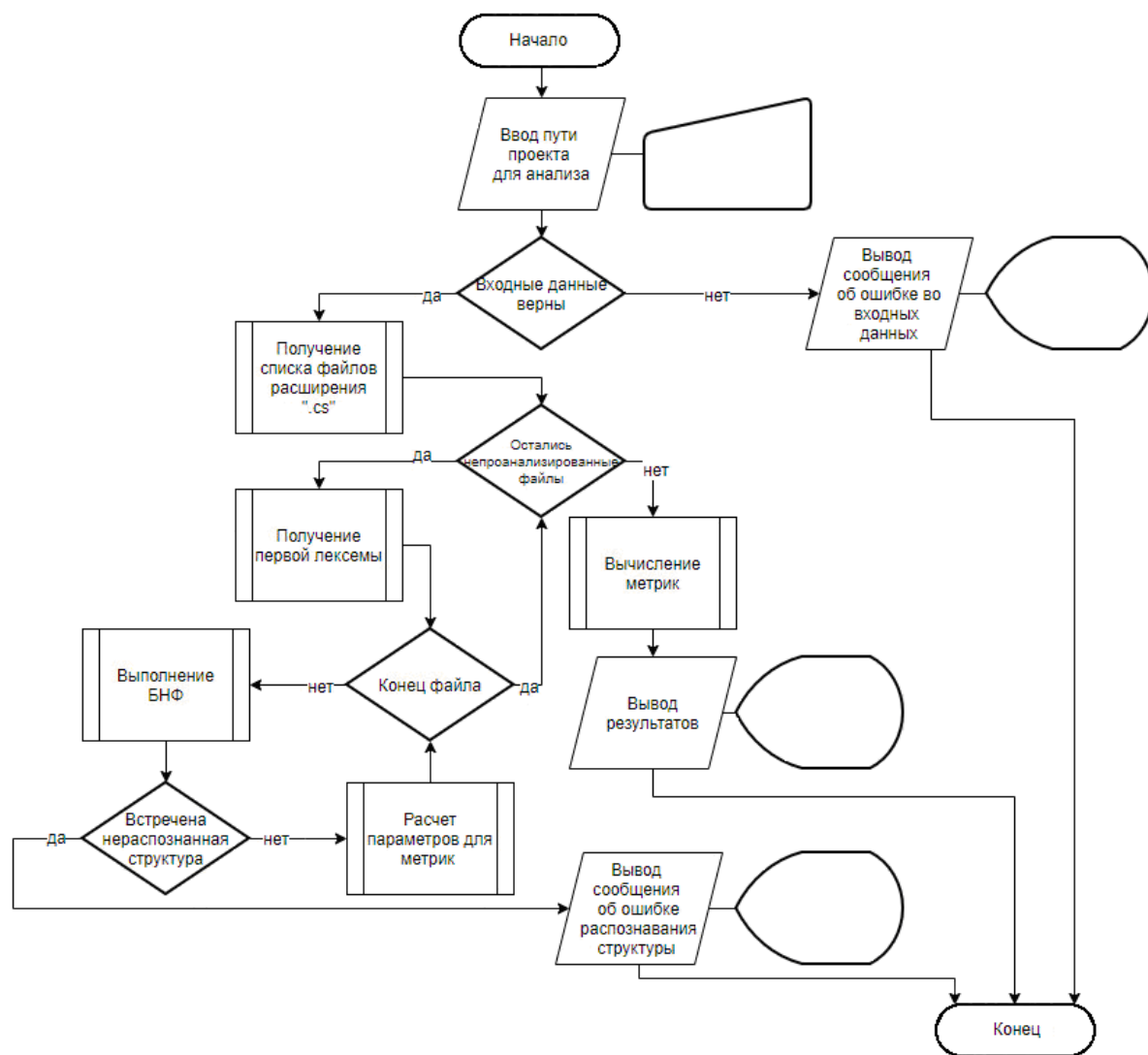
На основе результатов анализа и определения набора метрик была сформулирована постановка задачи разработки приложения расчета метрик. Входными данными являются путь к проекту и множество файлов, содержащих код на языке C#. В качестве ограничений выступают корректный путь до проекта и наличие анализируемых файлов. Также код должен быть компилируемым, принадлежать к описанному в работе подмножеству языка ООП (см. рис. 1). В результате работы приложения формируется список наименований метрик и их значений; в случае, если расчет не был произведен, выводится сообщение о соответствующей ошибке.

При расчете метрик используется подмножество конструкций языка C# [6]. Для описания его синтаксиса были составлены формы Бэкуса-Наура (БНФ), подмножество которых приведено на рис. 1.

<b>&lt;программа&gt; ::= {&lt;оператор добавления ресурса&gt;}{&lt;пространство имен&gt;   &lt;класс&gt;   &lt;перечисление&gt;}</b>
<b>&lt;пространство имен&gt; ::= &lt;заголовок пространства имен&gt; "{" {&lt;элемент пространства имен&gt; } "}"</b>
<b>&lt;элемент пространства имен&gt; ::= [&lt;атрибуты&gt;] [&lt;модификаторы&gt;] (&lt;класс&gt;   &lt;пространство имен&gt;   &lt;перечисление&gt;)</b>
<b>&lt;класс&gt; ::= &lt;заголовок класса&gt; "{" {&lt;элемент класса&gt; } "}"</b>
<b>&lt;элемент класса&gt; ::= [&lt;атрибуты&gt;] [&lt;модификаторы&gt;] (&lt;класс&gt;   &lt;конструктор&gt;   &lt;метод&gt;   &lt;поле&gt;)</b>
<b>&lt;метод&gt; ::= &lt;объявление идентификатора&gt; &lt;блок кода&gt;</b>
<b>&lt;блок кода&gt; ::= (&lt;заголовок конструктора&gt;   &lt;заголовок блока кода&gt;) (&lt;составной оператор&gt;   ";"</b>
<b>составной оператор&gt; ::= "{" {&lt;оператор&gt;   &lt;составной оператор&gt;} "}"</b>
<b>&lt;оператор&gt; ::= (while "("&lt;выражение&gt;")"   for &lt;параметры&gt; &lt;элемент оператора&gt;   do &lt;составной оператор&gt; while "("&lt;выражение&gt;")" ";"   if "("&lt;выражение&gt;")" &lt;элемент оператора&gt; else &lt;элемент оператора&gt;   try &lt;элемент оператора&gt; catch "("&lt;тип&gt;[&lt;имя&gt;]"") &lt;элемент оператора&gt;   switch "("&lt;выражение&gt;")" &lt;выбирающий оператор&gt;   foreach "("&lt;объявление идентификатора&gt; in &lt;выражение&gt; ")" &lt;элемент оператора&gt;   using "("&lt;объявление идентификатора&gt;&lt;оператор присваивания&gt; ")" &lt;элемент оператора&gt;   throw &lt;выражение&gt;   &lt;составной оператор&gt;   &lt;простой оператор&gt;   ";"</b>
<b>&lt;выражение&gt; ::= &lt;операнд&gt; {&lt;операция&gt;&lt;выражение&gt;} [&lt;условный оператор&gt;]</b>
<b>&lt;операнд&gt; ::= "("&lt;выражение&gt; ")" &lt;часть сложного идентификатора&gt;   &lt;сложный идентификатор&gt;   &lt;вызов метода&gt;   &lt;первичный оператор&gt;   &lt;унарный оператор&gt;   &lt;переменная без идентификатора&gt;   &lt;создание нового объекта&gt;   &lt;константа&gt;   &lt;имя&gt;</b>

Рис. 1. Используемые БНФ

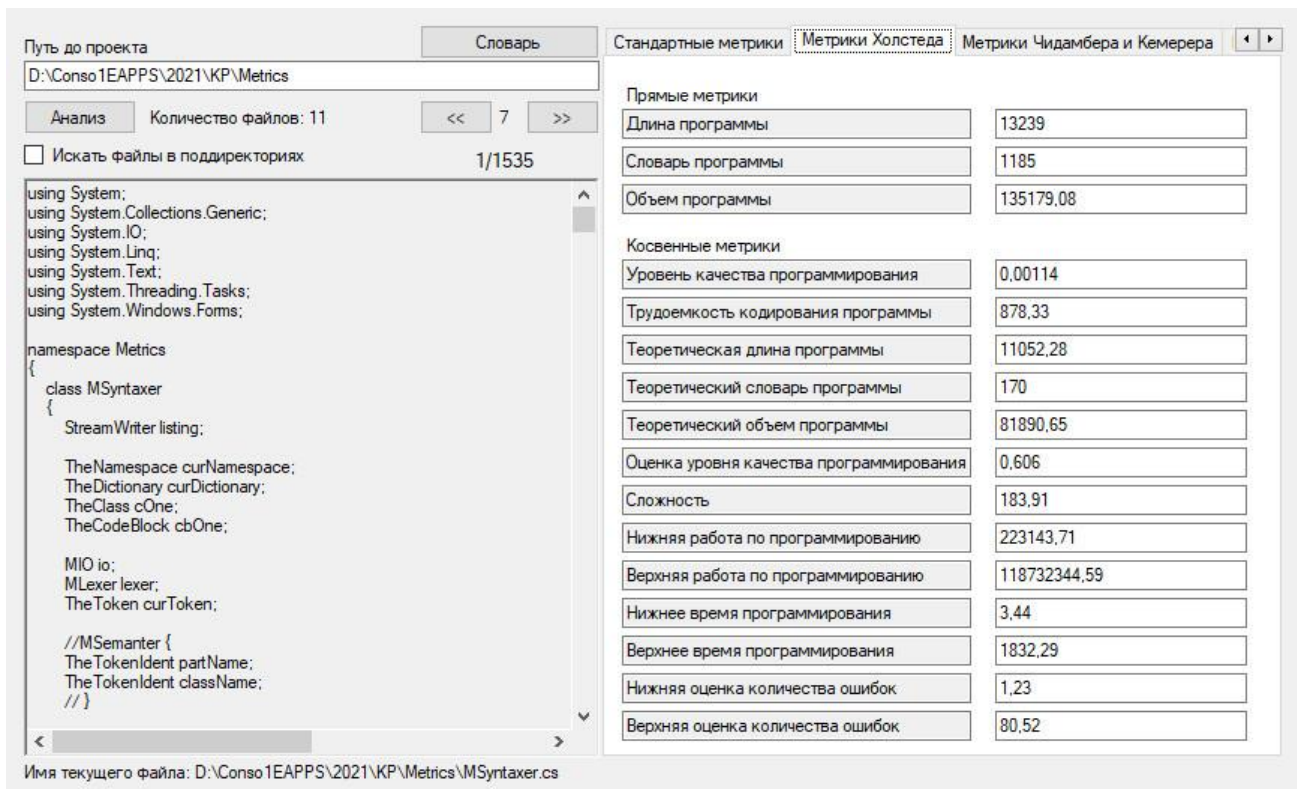
Для хранения информации о коде объектно-ориентированного языка созданы классы: блок кода (хранит основную информацию: название, список полей, словарь лексем), класс (содержит список блоков кода и дочерних классов, уровень в иерархии), корневой класс (корень собственной иерархии классов), пространство имен (агрегирует списки подпространств имен и классов). Алгоритм работы приложения представлен на рис. 2.



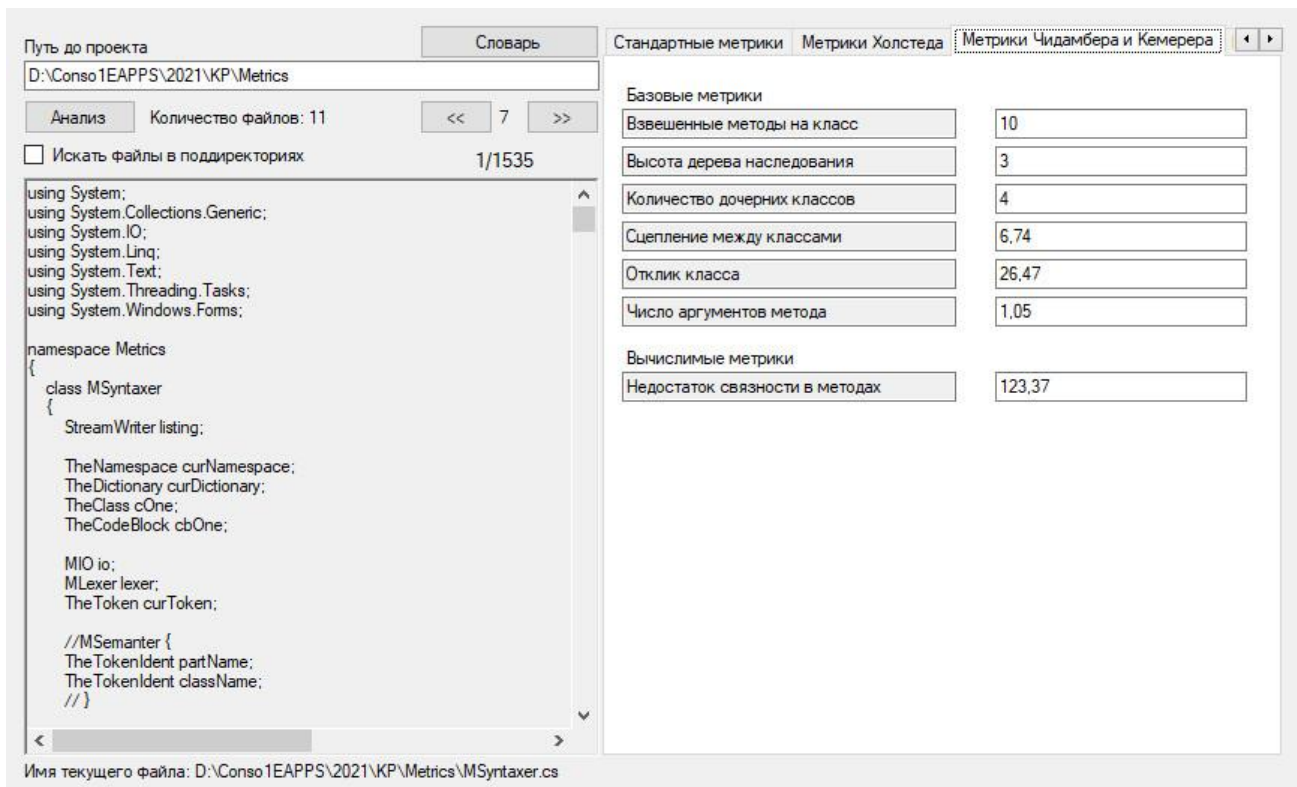
**Рис. 2. Алгоритм работы приложения**

При проектировании приложения были выделены его структурные элементы: уровень бизнес-логики (анализатор кода [7], классы структур ООП, модуль вычисления итоговых значений метрик) и уровень представления (интерфейс пользователя). В свою очередь модуль анализатора реализует фазы лексического и синтаксического-анализа, а также включает классы лексем, словари ключевых слов и специальных символов. Классы, описывающие код объектно-ориентированного языка, выполняют расчет метрик этой парадигмы и содержат словари операторов и операндов, необходимые для других метрик.

В результате вычисляются 11 метрик для объектно-ориентированных программ и 16 метрик для всех стилей программирования с помощью 19 классов. Тестирование разработанного приложения было проведено, в том числе, на его собственном коде. Итоги теста для метрик, не зависящих от парадигмы программирования, представлены на рис. 3, для относящихся к ООП – на рис. 4.



**Рис. 3. Метрики Холстеда для кода программы разработанного приложения**



**Рис. 4. Метрики Чидамбера и Кемерера для кода программы разработанного приложения**

Результатом работы является проектирование, реализация и тестирование приложения для получения метрик программного продукта на языке C#, способное получать набор значений, предоставляемых метриками, для их дальнейшего анализа специалистом, что позволит комплексно оценить программу.

## Библиографический список

1. ISO 9000 family. Quality management – URL: <https://www.iso.org/iso-9001-quality-management.html> (дата обращения: 10.04.2022).
2. CMM. Capability Maturity Model – URL: <https://www.geeksforgeeks.org/software-engineering-capability-maturity-model-cmm/> (дата обращения: 10.04.2022).
3. CMMI. Capability Maturity Model Integration – URL: <https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops> (дата обращения: 10.04.2022).
4. *Звездин С.* Метрики как средство управления качеством // Открытые системы. СУБД – 2009. – № 08. – с. 36-40.
5. Ming-Chang Lee, To Chang. Software Measurement and Software Metrics in Software Quality // International Journal of Software Engineering and Its Applications – Vol. 7, No. 4, July, 2013. – p. 15-34.
6. *Албахари Б., Албахари Дж.* С# 7.0. Справочник. Полное описание языка. М.: Вильямс – 2018. – 1024 с.
7. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции // М.: Мир – 2012. – Т. 1-2.

## PROGRAM REALIZATION FOR CODE METRICS CALCULATION IN OBJECT-ORIENTED PROGRAMMING LANGUAGE

*Korznikov Artem O., Datsun Natalya N.*

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

The work is devoted to automatization of receiving formal values for code properties. The purpose is to create the program for calculating code metrics values of program products in object-oriented language. We investigated approaches of metrics calculation and existing tools which provide it. Based on the analysis, the decision to use the set of metrics able to characterize program code in general using 16 metrics and as object-oriented one by 11 metrics was made. Using Backus–Naur forms a formal description of object-oriented language C# subset was created. It defines allowed language structures in compilable programs, which code can be measured. Program for calculating metrics set of code in object-oriented language was designed and created using C# language. Further it can be used for comprehensive estimation of software properties. The program was tested on various projects, including the code of the program itself.

Keywords: metric, software, metric calculation, object-oriented paradigm, object-oriented language, program code, software design, software development, software testing, estimation of software properties.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ УЧЁТА АКТИВОВ ПРЕДПРИЯТИЯ

*Котлов Максим Александрович*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, max6kotlov@gmail.com

*Соловьева Татьяна Николаевна*

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

Рассматриваются основные средства для проектирования и документирования информационной системы для учёта активов предприятия. Определён предмет и объект исследования, сформулирована актуальность разработки данной информационной системы, а также поставлены задачи, которые необходимо решить в ходе проектирования. В ходе исследования проведен анализ существующих информационных систем, которые выполняют похожие функции. Определены функциональные возможности информационной системы. Произведен выбор средств проектирования и разработки. В качестве средств проектирования рассматриваются инструменты для UML-моделирования [1]. В результате анализа средств проектирования было выбрано инструментальное программное обеспечение проектирования информационной системы CASE-средство Software Ideas Modeler для проектирования системы, и было выбрано приложение PaintTool SAI для разработки прототипа интерфейса приложения [2, 3]. В результате проектирования описываемой информационной системы было описано поведение системы и взаимодействие с ней бухгалтера. На основе построенных диаграмм был спроектирован интерфейс информационной системы.

Ключевые слова: учёт активов, актив, бухгалтерия, проектирование, документирование.

В настоящее время популярность организационно-правовых форм бизнеса только растёт. Предприятия открываются, предоставляют услуги, продают товары, но большинство так или иначе в конце прекращают свой жизненный цикл (ЖЦ), не задерживаясь надолго на рынке. Для увеличения ЖЦ предприятия и нужны программно-аппаратные решения, которые улучшат качество предоставляемых услуг или товара, и при этом уменьшат издержки, что позитивно отразится на влиянии предприятия на экономику. Потому следует считать данную тему актуальной.

Объектом исследования в данной работе является человек и предприятие, предметом – компьютеризация и систематизация данных активов предприятия.

Целью работы было проектирование и документирование программного средства, систематизирующее данные об активах предприятия.

Для выполнения цели выделены следующие задачи:

- Теоретический анализ существующих ПО на схожую тематику;
- Анализ и сравнение ПО для проектирования UML моделей;
- Анализ и сравнение ПО для разработки кода;
- Проектирование моделей UML логики приложения и базы данных;

– Разработка прототипа UI системы.

Логика работы системы построена в Software Ideas Modeler с помощью диаграмм прецедентов и представлена ниже (см. рис. 1.).

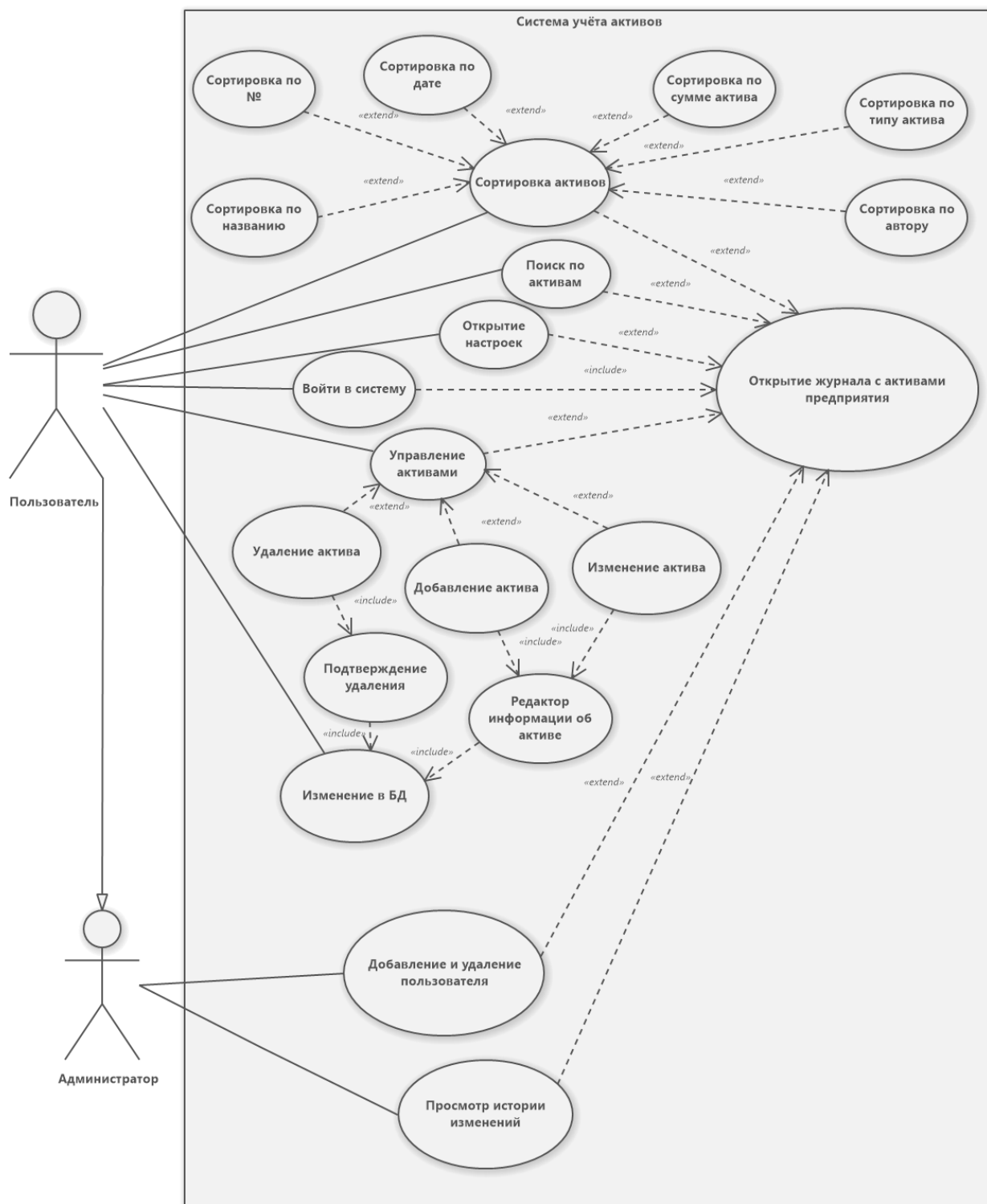


Рис. 1. Диаграмма прецедентов пользователя и администратора

Разберёмся, кем являются администраторы и пользователи. Учёт активов – это один из разделов бухгалтерского учёта, потому данной системой будут пользоваться сотрудникам бухгалтерии. Управляющие бухгалтерского отдела: главные бухгалтеры бухгалтерии, главные бухгалтеры различных отделов, заместители вышеперечисленных сотрудников и техни-



ческие специалисты. Пользователями являются остальные сотрудники бухгалтерии и производственного, материального и расчётного отделов, в зависимости от его структуры.

В отличие от пользователя администратор имеет полный доступ к БД и возможность заниматься менеджментом состава бухгалтерии, т.е. добавлять или удалять пользователей. Пользователи, как и администраторы имеют из функций сортировку по атрибутам актива, удаление, добавление и изменение данных об активе, а также проведение настройки приложения через одноимённое меню.

Отдельное внимание было уделено фактическому развёртыванию системы в реальных условиях. Учтя опыт аналогов, был проведён процесс моделирования при помощи диаграммы развёртывания. Эта диаграмма показывает возможности установки системы учета активов предприятия на реальное предприятие, а также показывает физическое развёртывание артефактов на узлах. Проще говоря, визуализирует, какие аппаратные компоненты существуют, какие программные компоненты работают на каждом узле, и как различные части этого комплекса соединяются друг с другом (рис. 2).

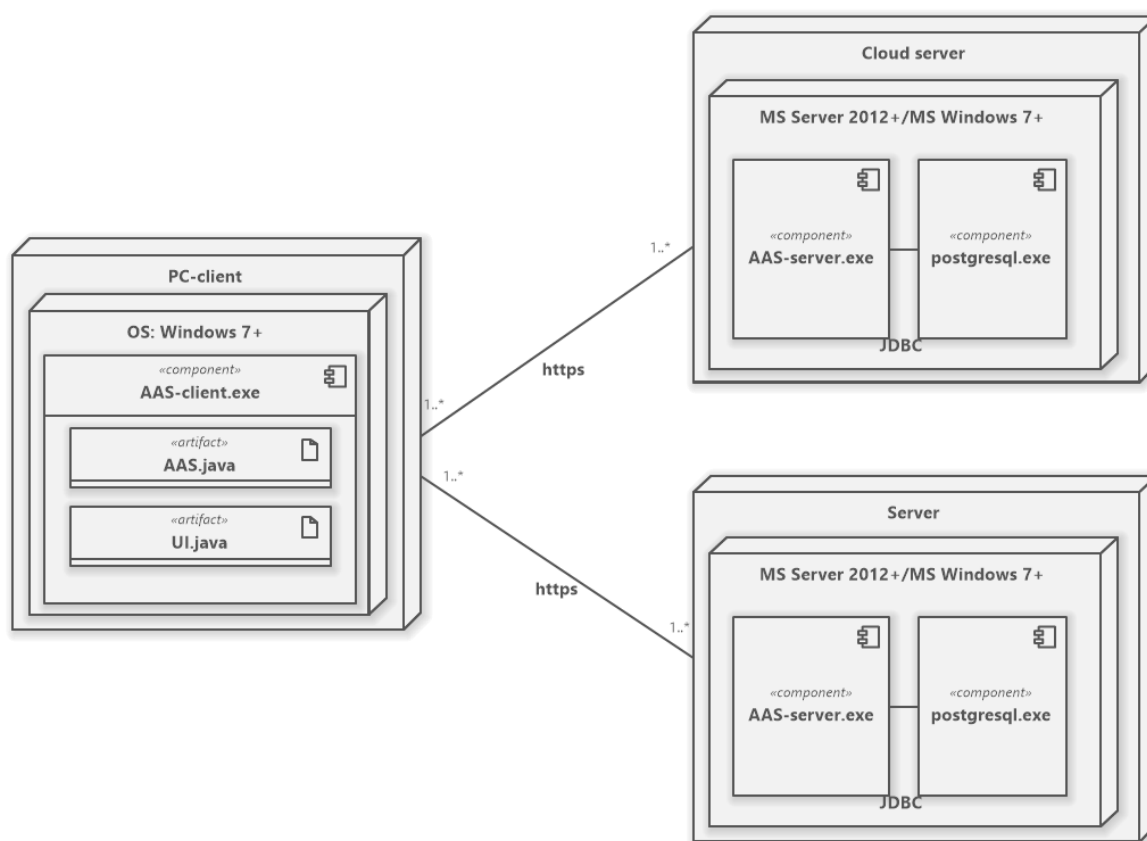


Рис. 2. Диаграмма развёртывания

Таким образом, на диаграмме реализованы https соединения клиентов либо с частным внутриорганизационным сервером, либо работа с облачным сервером. На рабочих станциях развёрнуты клиентские версии системы с файлами расширения типа .java и .exe, первые отвечают за код и пользовательский интерфейс программы, вторые – исполняемые файлы программы. На серверах стоят серверные версии системы, состоящие из приложения системы и системы управления базами данных. Отметим, что соединение на сервере проходит через независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД – JDBC.

Также здесь система учёта активов предприятия переименована в AAS, название является аббревиатурой перевода системы на английский язык.

Любая ИС, используемая в любой среде должна иметь интуитивно понятный пользовательский интерфейс – это уменьшает трудовые и временные затраты на обучение сотрудников.

После авторизации пользователя встречает главный журнал с активами. Сверху журнала расположены заголовки атрибутов актива. При клике на столбец данные об активе будут отсортированы в алфавитном порядке по выбранному атрибуту, при повторном клике – будут отсортированы в обратном алфавитном порядке, при последующем, третьем клике вернётся прежний вид журнала (рис. 3). Журнал можно пролистать вверх и вниз, для этого справа экрана расположен ползунок, или при помощи колесика мыши.

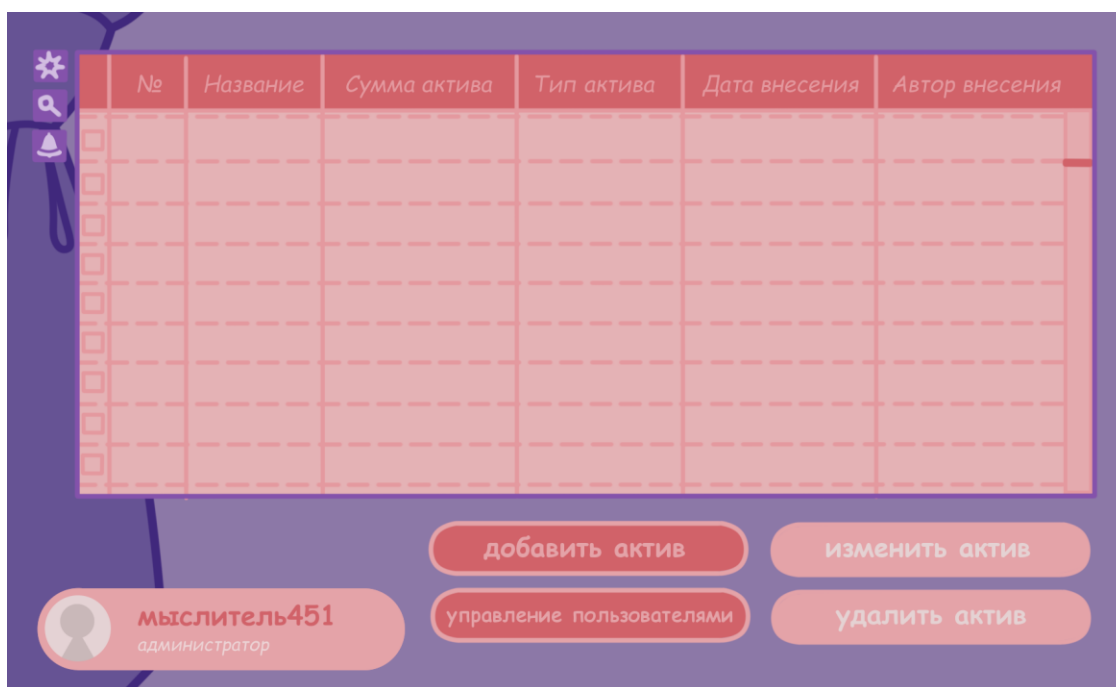


Рис. 3. Главное меню системы

При нажатии на настройку открывается меню настроек, состоящее из проверки обновлений, авторизации, персонализации и настройки подключения к серверу. При настройке подключения видны некоторые настройки такого подключения (**Ошибка! Источник ссылки не найден.**).

В меню «персонализация» возможна настройка шрифта и размера текста, а также цвета интерфейса.

Меню «авторизация» является единственной возможностью выйти из аккаунта или поменять пользователя.

При переключении в меню обновлений, пользователь может проверить ПО на наличие обновлений, если таковых не имеется, то кнопка «обновить» неактивна

При нажатии на кнопку лупы открывается меню поиска, введя ключевое слово, в которое, выводится результат.

При нажатии на колокольчик открывается «история изменений», где отображаются все последние изменения, совершенные пользователями для администраторов (рис. 12). У пользователей такой кнопки нет. Также все изменения записываются в таблицу «история изменений» в БД.

Кнопка «информация» создана для справки и обучения новых пользователей.

При нажатии на кнопку «добавление актива» или «изменение актива» открывается окно с активными для ввода или изменения полями атрибутов актива. Перемотка вниз или вверх происходит при помощи колесика мыши. В данном случае показан пример с изменениями данных об активе (см. рис. 4).

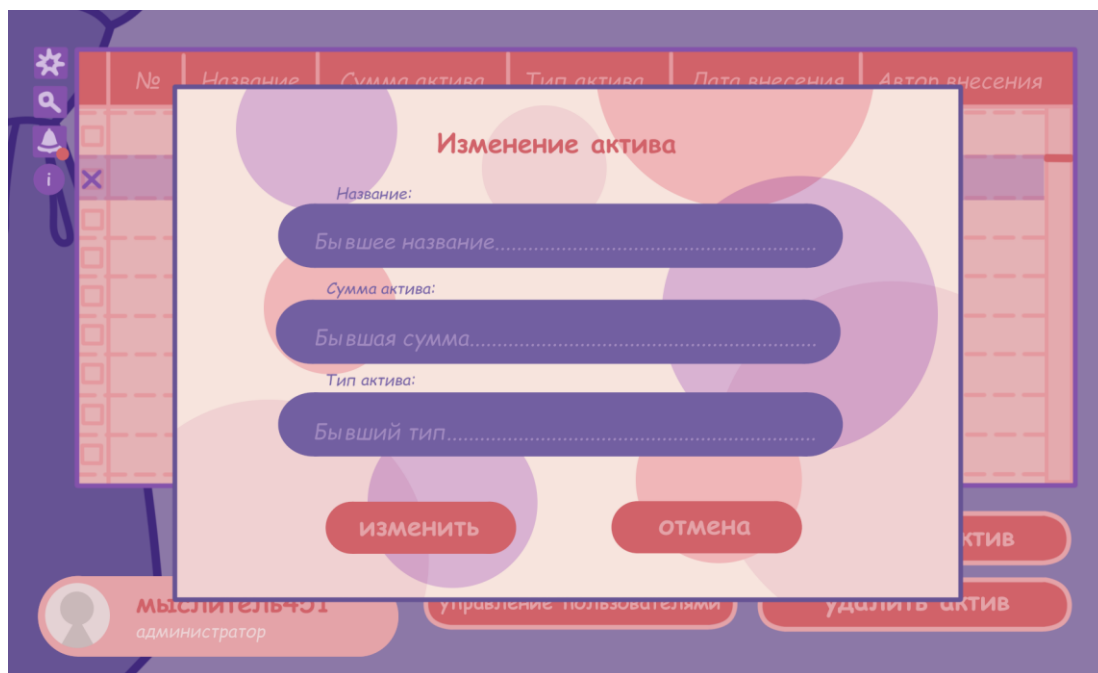


Рис. 4. Страница «Технологические карты»

Для добавления пользователя нажимается противоположная кнопка, после чего открывается окно с вводом информации о пользователе (рис. 5).

Поля ФИО, логина и пароля являются обязательными для ввода, остальные заполняются по желанию. Функция изменения данных не предусмотрена.

Такая возможность есть только через изменение полей таблиц СУБД, что разрешено делать администраторам.

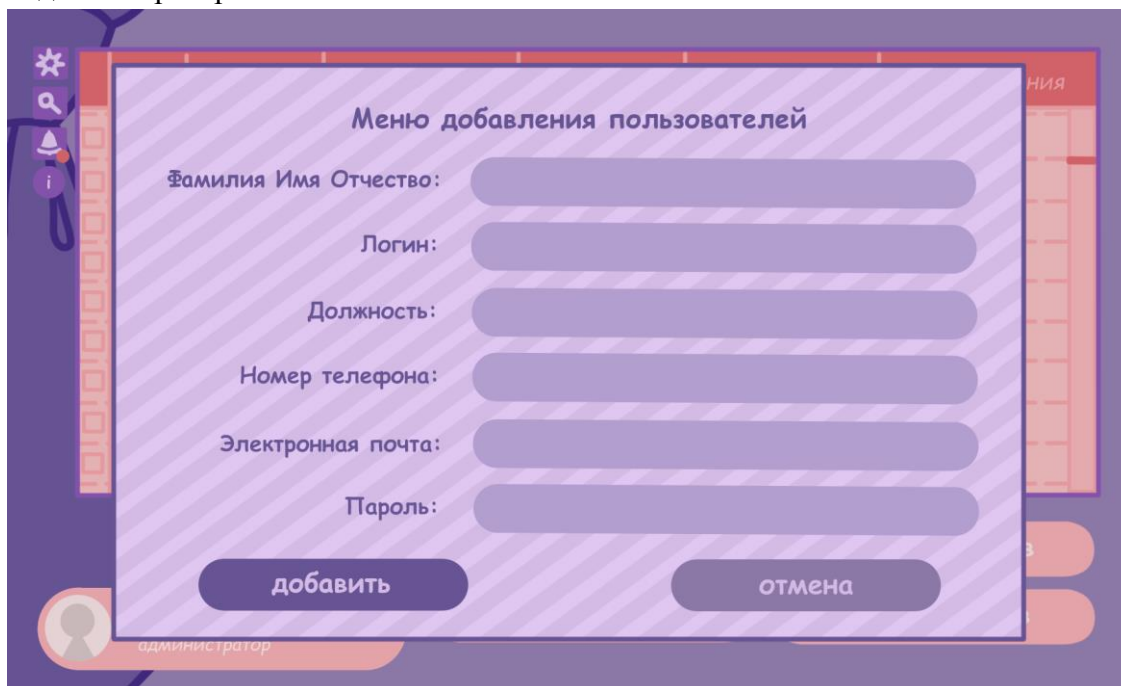


Рис. 5. Добавление пользователя

## Библиографический список

1. *Крэг Ларман* Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. – 3-е изд. – М.: Вильямс, 2006. – 736 с.
2. Features Overview: [Электронный ресурс] // Diagram CASE Tool for Software Modeling & Analysis – UML, BPMN, ERD. – URL: <https://www.softwareideas.net/en/features> (дата обращения: 17.05.2022)
3. PaintTool SAI [Электронный ресурс] // PaintTool SAI: high quality and lightweight painting software. – URL: <http://www.systemax.jp/en/sai/> (дата обращения: 17.05.2022)

## DESIGNING AND DOCUMENTING AN ENTERPRISE ASSET ACCOUNTING SYSTEM

*Kotlov Maxim A.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
artem.mirgaleev@mail.ru

*Solovyova Tatiana N.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
SolovevaTN@ya.ru

The main means for designing and documenting an information system for accounting for the assets of an enterprise are considered. The subject and object of research are defined, the relevance of the development of this information system is formulated, and the tasks that need to be solved during the design are set. In the course of the study, an analysis of existing information systems that perform similar functions was made. The functionality of the information system is defined. The choice of design and development tools is made. Tools for UML modeling are considered as design tools. As a result of the analysis of the design tools, the CASE information system design software was selected—the Software Ideas Modeler tool for system design, and the PaintTool SAI application was selected to develop a prototype of the application interface. As a result of the design of the described information system, the behavior of the system and the interaction of the accountant with it were described. The interface of the information system was designed on the basis of the constructed diagrams.

Keywords: asset accounting, asset, bookkeeping, design, documentation.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ АЛГОРИТМОВ ДИАЛОГОВЫХ ИНТЕРФЕЙСОВ ДЛЯ ВНЕШНИХ СИСТЕМ

*Летовальцев Даниил Дмитриевич, Кнутова Наталья Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, daniil-letovalcev@yandex.ru

В работе рассматриваются основные средства для проектирования и документирования информационной системы для автоматизации алгоритмов диалоговых интерфейсов для внешних систем. В ходе исследования были проанализированы: преимущества и недостатки аналогов, спрос на подобные информационные системы и экспериментально доказана востребованность на рынке систем для автоматизации алгоритмов диалоговых интерфейсов. При анализе аналогичных информационных систем были выявлены основные недостатки, которые требуется устранить в реализуемой информационной системе: слабая интеграция с ВКонтакте, отсутствие адаптивной мобильной версии, отдельная регистрация на электронном ресурсе, перегруженный интерфейс. В результате обзора инструментального ПО было выбрано инструментальное программное обеспечение информационной системы: CASE-средство Diagrams.net для моделирования системы [1]; сервис Figma для прототипирования интерфейса приложения; React.JS, HTML5, CSS3, MobX, VK Bridge для клиентской части информационной системы; Node.js как платформа для реализации серверной части информационной системы; MongoDB как СУБД на серверной части. В результате проектирования описываемой информационной системы было описано поведение системы и взаимодействие с ней пользователей. На основе построенных диаграмм был спроектирован интерфейс информационной системы, интерфейс был спроектирован как под десктопные, так и под мобильные устройства. На Android и iOS были соблюдены принципы дизайна в интерфейсе для этих операционных систем [2], [3]. Так же были рассмотрены предреализованные процессы: тестирование через VK Testers [4], настройка логирования и организация CI/CD процессов. Запущенная информационная система набрала 87.000 пользователей. Результаты запуска отображают востребованность проектируемой информационной системы на рынке систем для автоматизации алгоритмов диалоговых интерфейсов, демонстрируют спрос и актуальность системы.

Ключевые слова: средства для проектирования и документирования, конструктор чат-ботов, чат боты, автоматизация диалоговых интерфейсов, ВКонтакте, VKUI, адаптивный дизайн, диалоговые интерфейсы, автоматизация алгоритмов, MERN.

### **Введение**

В эпоху социальных сетей, когда одним из крупнейших каналов связи между представителями услуг и потребителями являются те самые социальные сети, автоматизация диалоговых интерфейсов становится актуальна как никогда. Помимо представителей услуг в потребности автоматизации диалоговых интерфейсов например являются:

– студенты и школьники (например: отображение расписание образовательного учреждения);

- администраторы страниц в социальных сетях (например: отображение актуальной информации о ценах, времени работы);
  - представители сферы бизнеса (например: отображение справочной информации о предприятии);
- автоматизация диалоговых интерфейсов позволяет решать множество бизнес-задач начиная от отображения общей информации о бизнесе заканчивая приемом и оплатой услуг.

### **Анализ предметной области**

Информационная система для автоматизации алгоритмов диалоговых интерфейсов для внешних систем включает в себя процесс, при котором пользователь ИС может создать чат-бота, а также настроить и развернуть его. Основные процессы, которые пользователь должен выполнить при ручном создании чат-бота:

- написание программного кода;
- Тестирование и отладка программного кода;
- настройка Callback API ВКонтакте;
- создание токенов сообщества и их настройка;
- аренда сервера для развертывания;
- развертывание чат-бота;
- настройка сервера;

дополнительные процессы, которые возможно потребуется реализовать пользователю в зависимости от его требований:

- хранение информации о пользователях чат-бота
- проведение рассылки в чат-боте
- создание панели администрирования чат-бота
- управление доступом в панель администрирования
- гибкая настройка чат-бота, через панель администрирования
- логирование событий в чат-боте для дальнейшего анализа
- формирование статистики чат-бота

### **Требования**

Должна быть спроектирована информационная система для автоматизации алгоритмов диалоговых интерфейсов для внешних систем. Система будет предназначена для создания чат-ботов для сообществ социальной сети ВКонтакте, которую стоит использовать в силу её большой аудитории и доступности API для интеграции решений.

К требованиям системы можно отнести:

- предоставление пользователям возможности создания, редактирования и управления чат-ботом;
- развертывание созданного пользователем чат-бота на сервере;
- обеспечение конфиденциальности, целостности и доступности информации;
- интеграция с ВКонтакте;
- адаптировать систему под мобильные устройства для использования с комфортом;
- разработать приятный и комфортный в использовании UI / UX;

- обеспечить возможность генерации отчетов с экспортом в актуальные форматы данных;
- ведение логов о событиях в информационной системе для дальнейшего анализа;
- использовать современные технологии в реализации, обеспечить гибкость разработки системы и её масштабируемость.

## Обзор инструментального ПО

В качестве языка программирования выбран JavaScript, одно из главных его преимуществ – возможность использования платформы Node.js, которая даст нам выигрыш в скорости работы реализуемой ИС, больший, чем, например, Java. Также учитывая, что наша клиентская составляющая будет так же на JavaScript, то один и тот же язык для клиентской и серверной составляющей является хорошим плюсом в нашем выборе. Node.js — платформа, позволяющая использовать JavaScript в серверных приложениях. Главный плюс Node.js в его скорости и возможности масштабируемости, чем он, например, выигрывает перед Java. В Node.js нет модели параллелизма на основе потоков. Когда в Java на каждый запрос на веб-сервера создается поток, то при большом количестве запросов количество создаваемых потоков соответственно будет расти и процессор сервера при большом количестве пользователей будет занят только переключением потоков. В Node.js же не создается новый поток, в Node.js один поток, однако он будет выигрывать в скорости перед Java в рамках данной задачи. На сервер будет идти большое количество HTTP-запросов при запросе пользователей к ботам ИС, при реализации на Node.js не будет создаваться поток на каждого пользователя, что существенно ускорит работу нашего сервера и с точки зрения бизнеса снизит расходы на аппаратное обеспечение сервера. В Node.js реализована событийно-ориентированная архитектура, благодаря которой снижается потребление памяти, упрощается реализация серверной логики. Так же учитывая, что проектируется веб-сервис, то придется использовать HyperText Markup Language 5 (HTML5), Cascading Style Sheets (CSS3).

В клиентской части помимо использования HTML5, CSS3 и React.js используется библиотека MobX – менеджер состояний. С помощью данной библиотеки можно управлять состоянием в приложении. Основное преимущество – это уменьшение количества рендеров и повышение читаемости кода.

Поток в React таков, что родительские компоненты передают входные параметры дочерним элементам, из-за чего при повышении кодовой базы понижается читаемость кода. Именно поэтому был использован менеджер состояний. Можно удобно хранить текущее состояние и использовать его на разных страницах приложения без потребности заново их запрашивать с сервера.

Стек клиентской части:

- React.js
- HTML5
- CSS3
- VKUI
- VK Bridge
- MobX

В качестве СУБД будет использоваться MongoDB. MongoDB — документоориентированная СУБД. MongoDB не требует описания схемы таблиц. Причины почему MongoDB подходит к стеку технологий реализуемой ИС.

- Простые предписанные решения
- Масштабирование
- Концепция шардинга для масштабирования системы
- Гибкость
- JSON формат

В рамках распределения процессов MongoDB будет вынесен отдельно в облачное решение. Благодаря этому будет понижен шанс утраты данных и повышена безопасность. А та же атомизированы обработчики отдельно от базы данных. Для обеспечения безопасности подключение к базе данных разрешено только с IP-адреса сервера приложения, а так же используются корневые сертификаты для подключения.

Для проектирования и прототипирования интерфейса будет использован: Figma. Для создания диаграмм будет использован: Diagrams.net. В целом получается известный стек (набор технологий) MERN (MongoDB, Express.js, React.js, Node.js), который хорошо зарекомендовал себя и является популярным решением при создании веб-сервисов.

### **Моделирование информационной системы**

В информационной системе будет реализовано клиент-серверное взаимодействие через API. Так как наша система будет интегрирована с ВКонтакте, более глубоко, чем аналоги, то регистрация пользователей для создания бота нам будет не нужна. Нужно будет только проверять подпись пользователя, о том, что это действительно он послал запрос. На каждый прецедент нам нужен будет отдельный метод обработчик на сервере. При открытии страницы будет посылаться запрос на сервер для получения данных, требуемых для отображения на странице приложения. На сервере будет проверяться подпись пользователя, после этого будет получен список данных пользователя из базы данных и будет возвращен ответ в клиентскую часть приложения. Клиентская часть в зависимости от ответа будет отображать: либо заглушку, о том, что у пользователя ещё нет данных; либо отображать список данных.

### **Прототип пользовательского интерфейса**

В проектируемом интерфейсе будут отображены элементы управления для пользователей информационной системы. Интерфейс должен быть простым и понятным для пользователей и в нем должны быть учтены привычные для пользователя паттерны поведения. С помощью интерфейса рядовой пользователь интуитивно и без затруднений должен понять как создать чат-бота для своего сообщества и разобраться в его управлении. Ведь в интерфейсе реализуемой информационной системы должно быть учтено поведение пользователей не имеющих технических навыков и быть понятной им, потому что проектируемая система рассчитана на массовую аудиторию и чтобы любой пользователь смог создать себе чат-бота, который бы покрывал его потребности.



## Тестирование

Одним из ключевых этапов в жизненном цикле информационной системы является тестирование и релиз этой системы. Тестирование помогает исправить недочеты в приложении, чтобы реальные пользователи при использовании приложения не столкнулись с ошибками во время работы. В тестировании продукта приняли участие 175 тестировщиков, было заведено 214 отчетов, 41 из них были отклонены, 128 исправлены, 15 из них были пожеланиями, остальные баги в зависимости от статусов были отложены.

Также из этих отчетов 5 были уязвимостями, 15 с критическим приоритетом, 31 с высоким, 58 с средним и 101 с низким. Каждая новая версия приложения проходит тестирование, что помогает избегать ошибок, возникающих у пользователей приложения.

## CI/CD

В данной системе процесс реализован следующим образом. Создается ветка для разработки в GitHub и при написании кода переключается на ветку, в которой добавляется новая функциональность. После того как весь планируемый код написан изменения отправляются в ветку и происходит запуск автоматических тестов для нового кода. Если автоматические тесты пройдены успешно, то код начинает развертываться на свою среду. Для серверной части это публикация кода на тестовый сервер и запуск всех нужных процессов на серверной части. Для клиентской части идет публикация на хостинг статических файлов. Все это реализуется через GitHub Actions.

Все ключи доступа к серверам и хостингу статических файлов записаны в GitHub Secret, где после записи никоим образом нельзя прочитать это значение, а только его удалить, что повышает безопасность, потому что GitHub Actions сам берет эти ключи доступа и подставляет их в указанные в процессах команды для подключения к серверам.

В данной информационной системе для серверной части процессы запускаются, если изменения отправляются на ветку master или stage. После этого запускаются автотесты и запускается исполняемый файл сервера, если возникла ошибка при запуске исполняемого файла сервера (например, возникла опечатка в коде из-за незакрытой скобки и файл не может запуститься) или при прохождении автотестов, то возникает ошибка и она отображается в GitHub Actions. Если же ошибка не возникла, то код вливается в ветку в репозитории и на сервере обновляется код и перезапускаются процессы и уже обновленный код спустя пару минут уже запущен на сервере и не нужно делать лишнюю работу по его доставке, развертыванию и прочим процессам, что существенно упрощает и ускоряет разработку.

В клиентской же части все аналогично: при публикации кода в ветку запускается процесс прохождения автоматических тестов и после успешного прохождения код публикуется на указанную, в зависимости от ветки, среду и уже готов к использованию.

Также в зависимости от того в какую ветку публикуется код запускаются процессы с разными параметрами: для публикации кода на пользователей или на тестовое окружение для дальнейшего ручного тестирования.

Таким образом в системе организовано CI/CD, что намного сокращает затраченное время на ручные действия и не приходится вручную переносить код на каждый из серверов и запускать его вручную, а автоматизирует весь этот процесс.

## Библиографический список

1. Введение в UML: Информация [Электронный ресурс] // Национальный Открытый Университет «ИНТУИТ». URL: <https://www.intuit.ru/studies/courses/1007/229/info> (дата обращения: 01.06.2022);
2. Design for Android [Электронный ресурс] // Google Developers. URL: <https://developer.android.com/design> (дата обращения 01.06.2022);
3. Human Interface Guidelines [Электронный ресурс] // Apple Developers. URL: <https://developer.apple.com/design/human-interface-guidelines/> (дата обращения 01.06.2022);
4. VK Testers | ВКонтакте [Электронный ресурс]. URL: <https://vk.com/testers> (дата обращения 14.06.2022)

## DESIGNING AND DOCUMENTING AN INFORMATION SYSTEM FOR AUTOMATING DIALOG INTERFACE ALGORITHMS FOR EXTERNAL SYSTEMS

*Letovaltsev Daniil D., Knutova Natalia S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, daniil-letovalcev@yandex.ru

**Abstract.** The paper considers the main tools for designing and documenting an information system for automating algorithms of dialog interfaces for external systems. The study analyzed the advantages and disadvantages of analogs, the demand for such information systems and experimentally proved the market demand for systems for automating algorithms of dialog interfaces. When analyzing similar information systems, the main shortcomings that need to be eliminated in the implemented information system were identified: weak integration with VKontakte, lack of an adaptive mobile version, separate registration on an electronic resource, an overloaded interface. As a result of the review of the tool software, the tool software of the information system was selected: CASE-tool Diagrams.net for system modeling [1]; Figma service for prototyping the application interface; React.JS, HTML5, CSS3, MobX, VK Bridge for the client side of the information system; Node.js as a platform for implementing the server side of an information system; MongoDB as a DBMS on the server part. As a result of the design of the described information system, the behavior of the system and the interaction of users with it were described. Based on the diagrams constructed, the interface of the information system was designed, the interface was designed for both desktop and mobile devices. On Android and iOS, the design principles in the interface for these operating systems were observed [2], [3]. Pre-release processes were also considered: testing via VK Testers [4], logging setup and organization of CI/CD processes. The launched information system has gained 87,000 users. The results of the launch reflect the demand for the designed information system in the market of systems for automating algorithms of dialog interfaces, demonstrate the demand and relevance of the system.

**Keywords:** design and documentation tools, chatbot constructor, chatbots, chat bots, VKontakte, automating dialog interfaces, VKUI, adaptive design, dialog interfaces, MERN.

## РАЗРАБОТКА ОНТОЛОГИЧЕСКИ УПРАВЛЯЕМОГО ПРОТОКОЛА СЕТЕВОЙ САМОИДЕНТИФИКАЦИИ УСТРОЙСТВ ИНТЕРНЕТА ВЕЩЕЙ

*Лукьянов Александр Михайлович*

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

В статье рассматривается совместное использование методов онтологического управления устройствами и протоколов самоидентификации, позволяющих этим устройствам передавать в сети идентифицирующую их информацию и за счёт этого находить друг друга – а именно объединение процессов поиска устройств и передачи между ними онтологий. Представлены результаты анализа существующих протоколов самоидентификации как базовых протоколов для объединения вышеуказанных процессов. Описаны процессы создания и дальнейшего использования программного модуля, реализующего протокол самоидентификации SSDP для микроконтроллеров. Данный модуль включает в себя механизмы дополнения протокола, позволяющие легко расширить его передачей онтологий.

Ключевые слова: интернет вещей, протоколы самоидентификации, поиск устройств, онтологический инжиниринг, микроконтроллеры, периферийные вычисления.

Использование сетей нулевой конфигурации (англ. Zero Configuration Networking) позволяет быстро и автоматически создавать сети устройств без их ручной настройки. Наибольшую пользу использование таких сетей приносит в случаях, когда создание сети требуется от людей, далёких от информационных технологий, либо когда состав сети необходимо изменять за максимально короткий промежуток времени.

Примером обоих таких случаев является использование сетей нулевой конфигурации в научных целях – для быстрой организации экспериментов, а также для визуализации и обработки их результатов. Возможности для организации и автоматизации этих процессов предоставляет платформа визуальной аналитики SciVi[1-5], разработанная сотрудниками ПГНИУ. Данная платформа позволяет описывать алгоритмы работы с данными и выполнять эти алгоритмы на различных устройствах, объединённых одной сетью, в т.ч. устройствах Интернета вещей (англ. internet of things, IoT). Однако, в данный момент для платформы остаётся нерешённой главная задача создания сетей нулевой конфигурации – самоидентификация устройств и их дальнейший поиск. Этот процесс затруднён тем, что для описания алгоритмов и их передачи на внешние устройства для исполнения, а также для описания состава самих этих устройств, платформа использует онтологии. Таким образом, возникает задача объединения процессов поиска устройств в сети и передачи между ними онтологий. На сегодняшний день не существует общедоступных программных средств, позволяющих удобно и полноценно решить эту задачу. Созданию таких средств и посвящена данная статья.

Необходимо также уточнить, что платформа SciVi для передачи онтологий устройствам IoT использует сжатие в специальный бинарный формат EON (англ. Embedded or Edge ONtology)[3], и вес сжатой онтологии, как правило, составляет всего несколько сотен байт.

На первом этапе работы был проведён анализ существующих протоколов самоидентификации и поиска устройств с целью выбора базового протокола для реализации и последующего дополнения. Были рассмотрены следующие протоколы:

1. NetBIOS (Network Basic Input/Output System) [6]
2. DNS-SD (DNS Service Discovery) [7]
3. mDNS (multicast DNS) [8]
4. SSDP (Simple Service Discovery Protocol) [9]
5. SLP (Service Location Protocol) [10]

Основные результаты анализа представлены в таблице 1 ниже.

**Таблица 1. Сравнение протоколов самоидентификации**

	NetBIOS	DNS-SD	mDNS	SSDP	SLP
1) Возможность обмена дополнительной информацией	2) +	3) -	4) -	5) +	6) -
7) Возможность поиска по именам	8) +	9) +	10) +	11) +	12) +
13) Возможность поиска по типам	14) -	15) +	16) -	17) +	18) +
19) Возможность поиска с помощью предикатов	20) -	21) -	22) -	23) -	24) +
25) Возможность поиска не только устройств, но и сервисов	26) -	27) +	28) -	29) +	30) +
31) Наличие дополнительных механизмов защиты	32) -	33) -	34) -	35) -	36) +
37) Наличие механизмов уведомления о вхождении устройства в сеть	38) +	39) -	40) -	41) +	42) +
43) Бинарный протокол представления данных	44) -	45) +	46) +	47) -	48) +
49) Год появления актуальной редакции протокола	50) 1987	51) 2013	52) 2013	53) 2015	54) 1999

На основании анализа данной таблицы, а также ряда дополнительных требований платформы SciVi, для дальнейшей реализации был выбран протокол SSDP. Этот протокол соответствует всем поставленным требованиям, прост в расширении передачи дополнительных данных и не содержит особенностей, не являющихся необходимыми для достижения цели работы, но реализация которых при этом обязательна для соответствия протоколу.

Протокол SSDP для передачи данных использует транспортный протокол UDP. Алгоритм поиска устройств протокола SSDP основан на отправке UDP-пакетов на специальный мультивещательный (англ. Multicast) адрес. Устройство, осуществляющее поиск, отправляет на этот адрес UDP-пакет, содержащий описание искомых устройств. Такое описание может заключаться в уникальном идентификаторе устройства, его типе или имени сервиса, который должно предоставлять искомое устройство. Затем каждое устройство определяет, соответствует ли оно этому описанию, и, если соответствует, отправляет ответный пакет. Пакет-ответ содержит адрес отправившего его устройства и его описание. Отправляются такие пакеты напрямую устройству, осуществляющему поиск – по адресу, указанному в поисковом пакете. Помимо этого, все устройства обязаны сообщать данные о себе и своих сервисах при

вхождении в новую сеть и повторять эти сообщения через определённые промежутки времени, а также сообщать о своём уходе из сети перед тем, как это произойдёт. По протоколу SSDP данные передаются в текстовом виде в формате запросов HTTP, но без тела запроса. Т.е. каждый передаваемый пакет состоит из стартовой строки, описывающей этот пакет, и набора пар имя-значение (заголовков HTTP). Документация протокола описывает обязательные заголовки, которые должны содержаться в пакете, оставляя возможность добавлять любые другие, если только их имена уникальны внутри одного пакета[9].

В соответствии со стандартом протокола, дополнительно устройства могут иметь в своём составе отдельные встроенные устройства, каждое со своим набором сервисов, и поиск может производиться также по встроенным устройствам и их сервисам. Однако в данной статье рассматривается реализация протокола для устройств IoT, которые обычно не имеют встроенных устройств, поэтому возможность такого поиска в работе не исследовалась и не была реализована.

В качестве устройств IoT в SciVi в основном используются микроконтроллеры семейства ESP8266, поэтому в ходе работы разрабатывалась реализация протокола в первую очередь для них. Для разработки использовались инструменты специального дополнения к среде разработки Arduino IDE, позволяющее создавать программы для этих микроконтроллеров на языке C++. [11]. Это дополнение включает в себя библиотеки функций для сетевого взаимодействия, в том числе для приёма и отправки UDP пакетов на обычные и мультивещательные адреса. Таким образом, в реализации нуждаются только сами алгоритмы поиска и самоидентификации, описанные выше. В том же наборе библиотек уже существует реализация протокола SSDP[12], однако она является неполной, реализует устаревшую версию протокола и не предоставляет возможности для расширения.

Для решения поставленной задачи была разработана библиотека классов для языка программирования C++, содержащая класс, реализующий протокол SSDP. Диаграмма рабочего процесса объекта этого класса представлена на рис. 1.

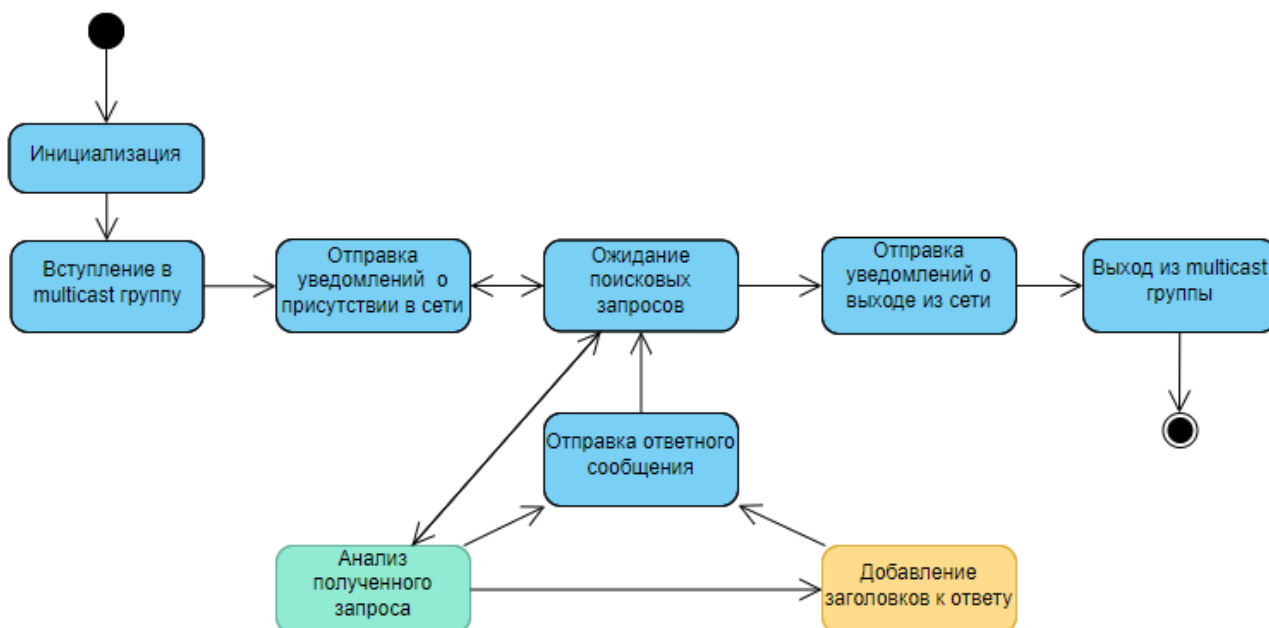


Рис. 1. Диаграмма работы объекта, реализующего протокол

Большинство этапов работы сводится к использованию примитивов сетевой коммуникации, реализованных в упомянутом выше наборе библиотек, а особого внимания требуют этапы, выделенные цветом.

Анализ получаемых поисковых запросов является задачей синтаксического анализа, и для её решения используется машина состояний, диаграмма состояний которой представлена на рис. 2 (каждое состояние, кроме состояния обработки ошибок, означает прочтение советующей лексемы). В процессе анализа также принимается решение, соответствует ли устройство поисковому запросу. В случае соответствия запросу и корректности поискового пакета, после окончания анализа производится отправка пакета-ответа устройству, осуществляющему поиск.

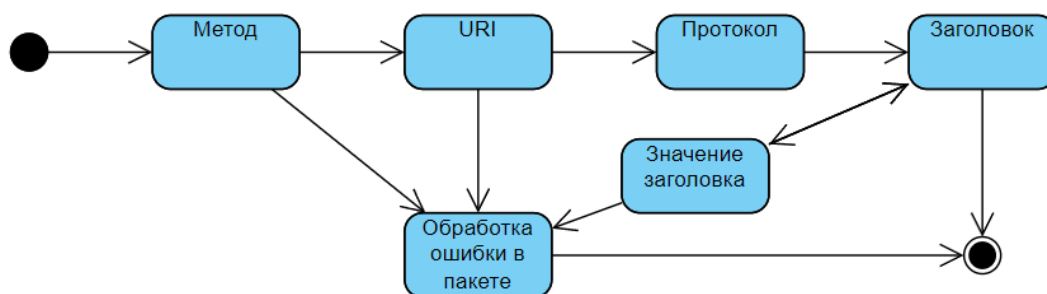


Рис. 2. Диаграмма состояний синтаксического анализатора пакетов

Добавление заголовков к ответу на поисковый запрос является главной особенностью данной реализации. Разработанный класс позволяет, используя механизм наследования, переопределить специальные методы, вызываемые автоматически перед отправкой пакета-ответа. Структура класса позволяет вызовом ещё одного метода из переопределённых методов добавлять к пакету произвольные заголовки. Именно за счёт этого механизма данную реализацию протокола SSDP можно использовать для передачи онтологий. Важно отметить, что итоговый размер пакета не должен превышать 64 кб, при этом обязательные заголовки занимают порядка нескольких сотен байт, что оставляет достаточно места для дополнительных заголовков, соответственно и для передачи в них больших онтологий, сжатых средствами EON.

Также созданный класс позволяет задавать и считывать дополнительные атрибуты устройства, не используемые в процессе поиска, но помогающие подробнее описать устройство, такие как человекочитаемое имя устройства, название модели, сайт производителя и т.д.

Описанная реализация протокола SSDP была успешно протестирована и в данный момент внедряется в платформу SciVi. Исходный код библиотеки, а также примеры её использования и инструмент для автоматического тестирования, находятся в открытом доступе и доступны для использования [13].

### Библиографический список

1. Ryabinin K., Chuprina S. Adaptive Scientific Visualization System for Desktop Computers and Mobile Devices // *Procedia Computer Science*. – Elsevier, 2013.– Vol.18.– PP. 722–731.
2. Ryabinin K., Chuprina S. Development of Ontology-Based Multiplatform Adaptive Scientific Visualization System // *Journal of Computational Science*. – Elsevier, 2015. – Vol. 10. – PP. 370–381.
3. Ryabinin K., Chuprina S. Ontology-Driven Edge Computing // *Lecture Notes in Computer Science*. – Springer, 2020. – Vol. 12143. – P. 312–325.

4. *Ryabinin K., Kolesnik M.* Scientific Visualization System on a Chip with Tangible User Interface // Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). Part 2. – 2020. – Vol. 2744.
5. *Ryabinin K., Chuprina S., Labutin I.* Tackling IoT Interoperability Problems with Ontology-Driven Smart Approach // Lecture Notes in Networks and Systems. – Springer, 2021. – Vol. 342. – PP. 77–91.
6. PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: CONCEPTS AND METHODS, 1987 [Электронный ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc1001>
7. *Cheshire S., Krochmal M.* DNS-Based Service Discovery, Apple Inc., 2013 [Электронный ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc6763>
8. *Cheshire S., Krochmal M.* Multicast DNS, Apple Inc., 2013 [Электронный ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc6762>
9. UPnP Device Architecture 2.0, Open Connectivity Foundation Inc., 2020 [Электронный ресурс] URL: <https://openconnectivity.org/upnp-specs/UPnP-arch-DeviceArchitecture-v2.0-20200417.pdf>
10. *Guttman E., Perkins C., Sun Microsystems, Veizades J.* Service Location Protocol, Version 2, 1999 [Электронный ресурс]: <https://www.ietf.org/rfc/rfc2608.txt>
11. *Grokhotkov I. u др.* Arduino core for ESP8266 WiFi chip [Электронный ресурс] URL: <https://github.com/esp8266/Arduino>
12. *Gochkov H., Sallemi F.* ESP8266SSDP library [Электронный ресурс] URL: <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266SSDP>
13. *Lukianov A.* almilukESP8266SSDP library [Электронный ресурс] URL: <https://github.com/almiluk/almilukESP8266SSDP>

## **DEVELOPMENT OF AN ONTOLOGY-DRIVEN NETWORK SELF-IDENTIFICATION PROTOCOL FOR INTERNET OF THINGS DEVICES**

*Lukianov Aleksandr M.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [almiluk@yandex.ru](mailto:almiluk@yandex.ru)

**Abstract.** We research the combined use of ontology-based device management techniques and self-identification protocols that allow these devices to transmit identifying information to the network and, by doing so, find each other. Specifically, combining the processes of finding devices and transferring ontologies between them. Existing self-identification protocols as basic protocols for unification of that processes are analyzed. The creating and further using the program module implementing the SSDP self-identification protocol for microcontrollers are described. This implementation includes mechanisms for extending the protocol to allow it to be easily extended by the transfer of ontologies.

**Keywords:** internet of things, self-identification protocols, device searching, ontology engineering, microcontrollers, edge computing

## **ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМ ДЛЯ СВОМЕСТНОЙ РАЗРАБОТКИ ПРОЕКТОВ**

*Маргин Ефим Романович, Соловьева Татьяна Николаевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, pro100efim@gmail.com

Цель данной работы состоит в проектировании, документировании и разработке системы для совместной разработки проектов. В работе проведен анализ аналогов информационных систем, выявлены их достоинства и недостатки. По результатам анализа поставлены задачи на проектирование и составлены требования к разрабатываемой системе. В соответствии с предъявленными требованиями и проведенным сравнительным анализом по ряду критериев и параметров выбраны средства моделирования и реализации информационной системы. Проведен анализ состояния текущих бизнес-процессов и предложен вариант оптимизации. На основании этого спроектирована модель системы. Реализован прототип информационной системы сбора и обработки изображений для машинного обучения внешних систем.

Ключевые слова: модульность, совместная разработки, информационная система.

С каждым годом наша жизнь все больше и больше взаимосвязана с информационными технологиями и информационными системами, что ведет к увеличению интереса разработчиков программного обеспечения к созданию крупномасштабных it-проектов, для реализации которых требуется отлаженное взаимодействие между членами команды, документация проделанной деятельности и визуализация рабочих задач. Информационная система должна быть открытой и гарантировать конфиденциальность задокументированной информации. А также доступ к этой информации в любой момент времени, так как это может на прямую влиять на построение рабочего процесса команды

Разрабатываемая информационная система в данной работе направлена на замещение существующих иностранных аналогов, дабы избежать возможность блокировок доступа разработчикам к их рабочей области тем самым нарушая логику взаимодействия внутри команды, оптимизацию процесса создания и документирования разрабатываемых информационных систем, посредством обеспечения доступа всей команды разработчиков к перечню текущих задач, возможности редактирования и создания новых задач, а также возможности присваивания статусов, которые будут видны другим членам команды, например, «Задача – Оптимизация модуля сортировки. Статус – в работе». Это позволит командам разработчиков выполнять более высокие объемы задач, повысит уровень логики и менеджмента внутри проекта без значительных трудозатрат со стороны штата разработчиков. Также данное решение сможет повысить уровень понимания общих целей и задач проекта у каждого сотрудника, так как предоставит возможность дополнительно визуализировать весь процесс, который выполняется в процессе разработки продукта. Еще одним преимуществом данной информа-



ционной системы может стать повышение скорости интеграции новых сотрудников в проект и повышение общего качества разработки.

Под проектом будет подразумеваться совместно разрабатываемая ИС, для которой существуют сформулированные требования, а конечный результат понятен<sup>11</sup> [1].

Рассмотрим несколько представителей систем по совместной разработке

– Trello – облачная программа для управления проектами небольших групп, разработанная FogCreek Software. Trello использует парадигму для управления проектами, известную как канбан метод.

– Notion – приложение, которое предоставляет такие компоненты, как базы данных, доски канбан, вики, календари и напоминания. Пользователи могут подключать эти компоненты для создания собственных систем управления знаниями, ведения заметок, управления данными, управления проектами и другими

– Jira – коммерческая система, предназначена для организации совместной разработки проектов. Имеет веб-интерфейс, а так же обширный набор подключаемых модулей, которые могут быть настроены при необходимости команды.

– Rugus – отечественный представитель сегмента. Система для организации совместной разработки проектов посредством таких процессов как: общение, управление задачами, согласование заявок и документов и автоматизация бизнес-процессов.

Основным достоинством является то, что каждая из взятых на рассмотрение систем выполняет свою основную задачу – позволяет заниматься совместной разработкой проектов.

Несмотря на модульность Notion главным недостатком системы, по мнению пользователей, является перегруженность и громоздкость, на организацию рабочего пространства, часто уходит больше времени, чем на саму деятельность, отсутствие возможности выгружать отчетность, тяжеловесность.<sup>12</sup>

Функционал Trello напротив – слишком упрощён, так как отсутствуют инструменты для аналитики масштабных проектов, выглядит как инструмент для разработки небольших проектов, а никак универсальное решение.

Из недостатков Jira – отсутствие возможности назначать нескольких исполнителей на одну и ту же задачу, так же имеется огромное количество настроек для каждого тикета задач, что весьма перегружает рабочий процесс и делает его сложным для пользователя. Основной же минус заключается в том, что система ограничивает доступ к системе по принципу географического расположения, не гарантирует безопасность и сохранность данных, относящихся к разработке.

Недостатками Rugus является слишком упрощенный дизайн, по мнению некоторых пользователей, не позволяющий быстро взаимодействовать с системой. Весьма ограниченный спектр выгружаемой отчетности

---

<sup>11</sup> <https://research-journal.org/social/chto-takoe-proekt-opredelenie-i-priznaki/>

<sup>12</sup> <https://habr.com/ru/post/539308/>

В первую очередь необходимо определить общие требования к функционалу проектируемой ИС. С этой целью будут использоваться диаграммы прецедентов (вариантов использования), в них можно увидеть связь между акторами и прецедентами системы.

На рисунке 1 представлена диаграмма прецедентов (использования), показывающая прецеденты связанные с акторами данной системы, а именно пользователем и администратором. Пользователь сможет пользоваться функционалом разрабатываемой системы. Администратор будет обладать всем тем же функционалом, что и пользователь, но, кроме этого, будет иметь возможность редактировать, добавлять и удалять учетные записи обычных пользователей.

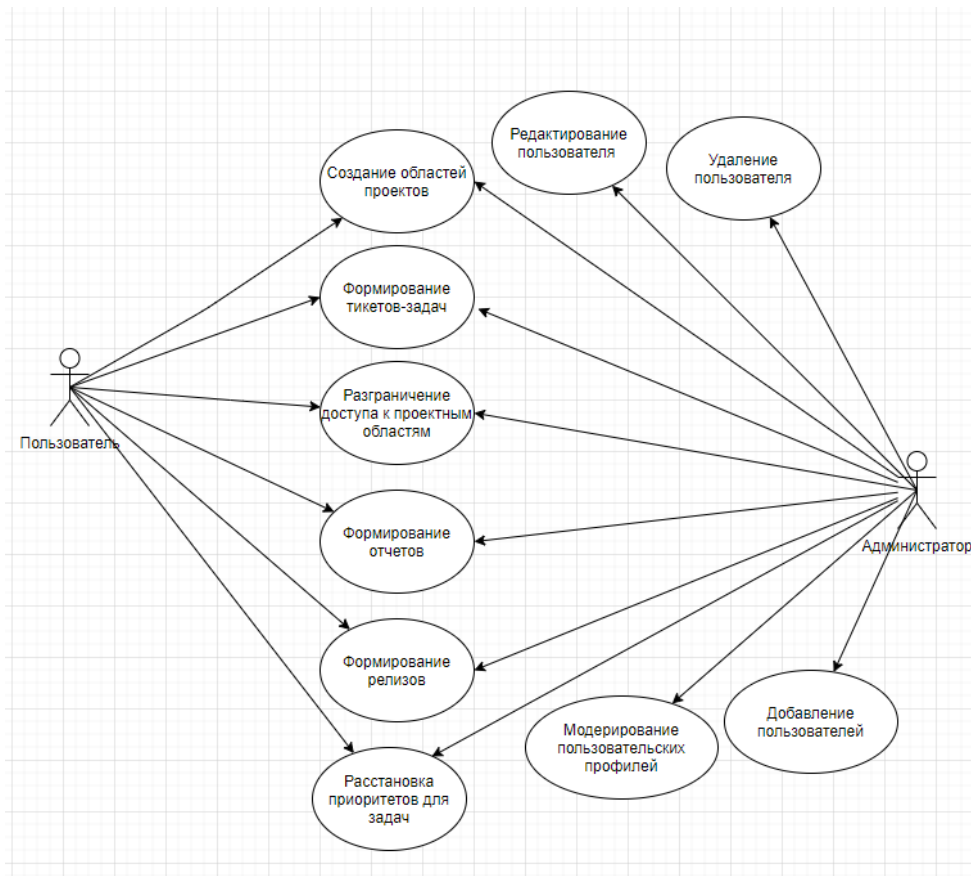
- Возможности пользователя:
- Создание областей проектов.
- Формирование тикетов-задач.
- Разграничение доступа к проектным областям.
- Формирование отчетов.
- Формирование релизов<sup>13</sup>.
- Расстановка приоритетов для задач.
- Приглашение в свою рабочую область других пользователей.

Администратор может:

- Приглашение в свою рабочую область других пользователей.
- Весь функционал доступный пользователю.
- Добавление пользователей.
- Редактирование пользователей.
- Удаление пользователей.
- Моделирование пользовательских профилей.

---

<sup>13</sup> <https://habr.com/ru/company/hygger/blog/358204/>



**Рис. 1. Диаграмма прецедентов**

Диаграмма последовательностей описывает поведенческие аспекты системы и рассматривает взаимодействие объектов. На рис 2 представлена схема, по которой можно увидеть, как пользователь может взаимодействовать с данной системой. Взаимодействия пользователя и системы начинается с авторизации. После ввода этих данных пользователем, они проверяются на корректность. Далее пользователь получает доступ ко всему функционалу разрабатываемой системы. В схеме представлен примерный подход при использовании сервиса.

Пользователь формирует рабочее пространство и добавляет туда других пользователей. Далее каждый из пользователей может создавать тикеты с задачи внутри рабочего пространства. Задачи видны и доступны для редактирования всем пользователем. Пользователь создающий тикет может назначить того, кто будет выполнять его. Кроме того, существует возможность формировать и выгружать отчетность, касающуюся рабочего процесса внутри рабочей области. Отчет может содержать внутри себя следующую информацию:

Все тикеты, выполненные за заданный промежуток времени.

Все пользователи, выполнявшие хотя бы одну задачу внутри заданного временного промежутка.

Количество выполненных задач для каждого пользователя, попавшего в отчет.

Суммарное время, затраченное на выполнение задач.

Данный функционал предусмотрен с целью анализа рабочего процесса для выявления проблемных моментов в менеджменте команды. На основе этой информации будут приниматься решения, касательно оптимизации данного процесса.

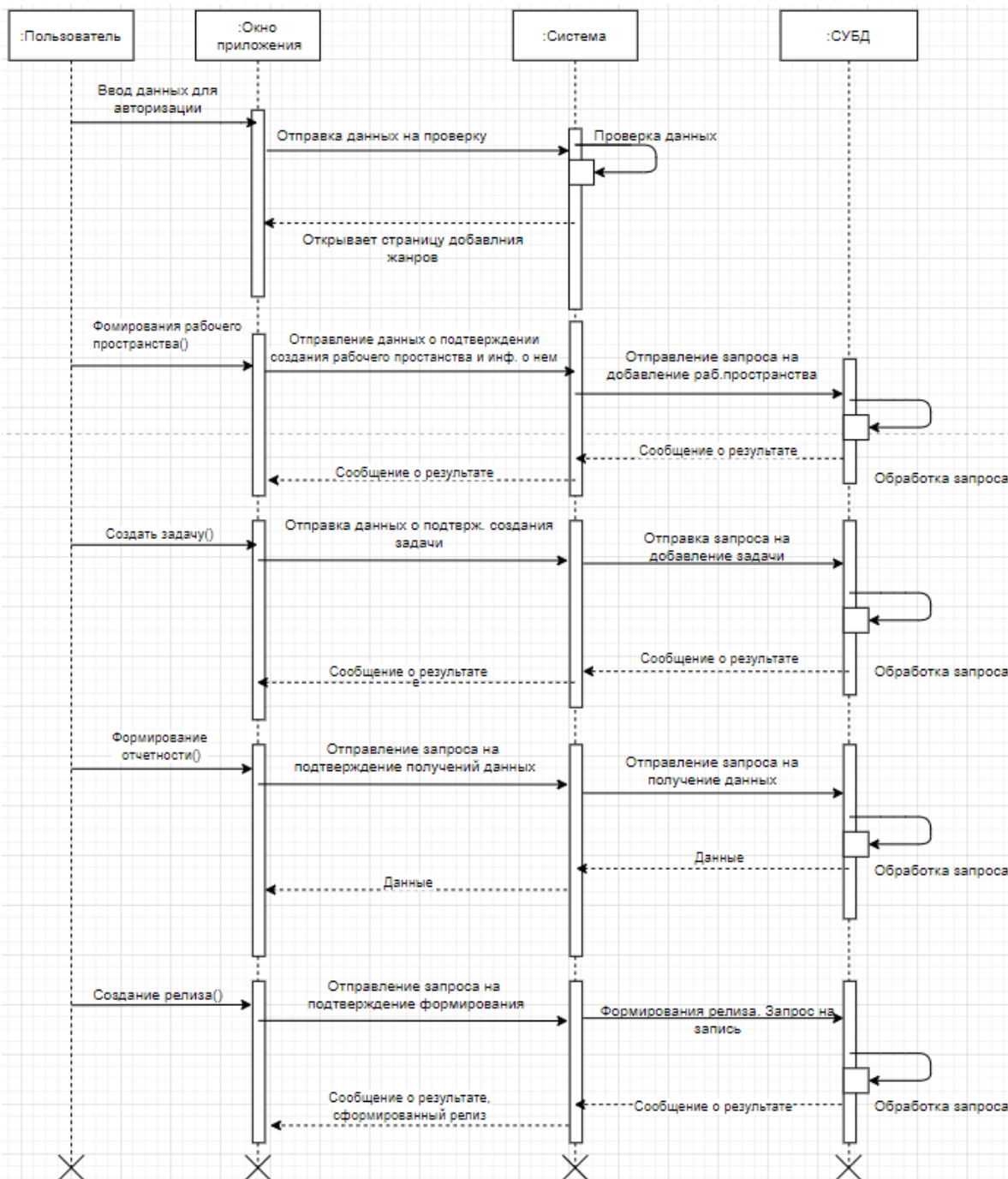


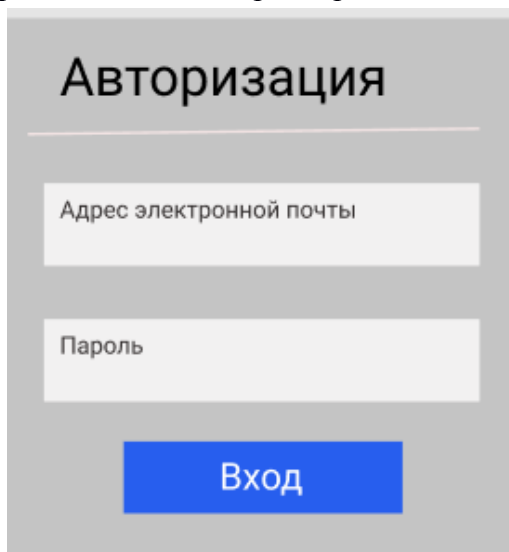
Рис. 2. Диаграмма последовательностей

Основное окно представляет собой панель навигации по проектам и задачам, кнопку для создания рабочей области. После того, как рабочая область создана, пользователь имеет возможность взаимодействовать с ней, с помощью кнопок, расположенных в левой части экрана, а именно:

- Окно регистрации
- Окно регистрации
- Посмотреть доску с текущими задачами
- Посмотреть список всех задач

- Сформировать отчет по проекту
- Оформить релиз

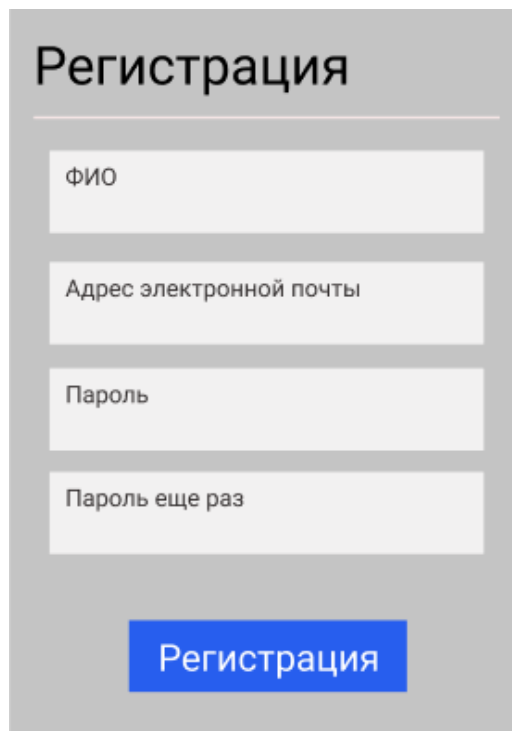
Окно авторизации представлено в абсолютно классическом виде, вход в систему происходит по адресу электронной почты и паролю рис 3.



Прототип окна авторизации. В центре экрана на сером фоне находится белая панель. В верхней части панели крупным черным шрифтом написано «Авторизация». Ниже этой заголовка находится горизонтальная линия. Далее следуют два белых текстовых поля: первое с надписью «Адрес электронной почты», второе с надписью «Пароль». В нижней части панели расположен синий прямоугольный элемент с белым текстом «Вход».

**Рис. 3. Прототип окна авторизации**

При регистрации пользователь вводит ФИО, адрес своего почтового ящика и дважды пароль рис. 4.



Прототип окна регистрации. В центре экрана на сером фоне находится белая панель. В верхней части панели крупным черным шрифтом написано «Регистрация». Ниже этой заголовка находится горизонтальная линия. Далее следуют четыре белых текстовых поля: первое с надписью «ФИО», второе с надписью «Адрес электронной почты», третье с надписью «Пароль», четвертое с надписью «Пароль еще раз». В нижней части панели расположен синий прямоугольный элемент с белым текстом «Регистрация».

**Рис. 4. Прототип окна регистрации**

На стартовой странице отображаются все доступные рабочие пространства, попасть на эту страницу можно, нажав на кнопку «проекты».

Кнопка «проекты» и «задачи» оформлены в виде выпадающего списка. При наведении на первую из них можно так же увидеть список всех доступных пользователю списков

рабочих областей. В случае наведения на кнопку «задачи», пользователь сможет увидеть список из 5-ти самых свежих тикетов внутри текущей рабочей области. При нажатии на нее, пользователь перейдет к полному списку, где будут отображены все существующие задачи.

Пользователь может переключаться между ними, эта функция позволяет логически разделять работу над разными проектами внутри одной или нескольких компаний.

В разделе «доска задач» отображены все существующие задачи, а также их расположение в соответствующей колонке, которая характеризует статус рабочего процесса.

Рис. 5.



Рис. 5. Прототип страницы рабочей области

Логикой приложения подразумевается возможность взаимодействия с карточками задач и их перетаскивание и между колонками, с целью изменения статуса рабочего процесса.

На данном этапе проектирования заложено 4 доступных статуса разработки.

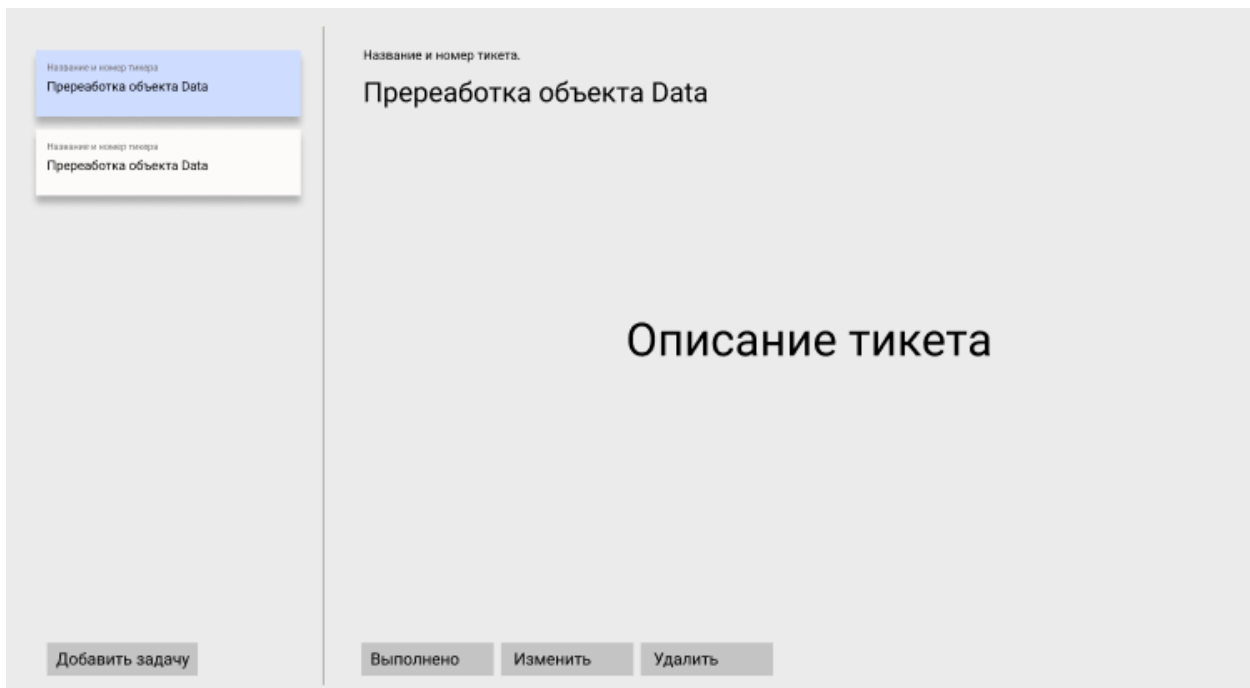
Требуется сделать – задача находится в ожидании пользователя, который будет вести над ней работу. Если задача находится в этом разделе – работа над ней еще не начата.

В работе – пользователь начал заниматься решением задачи из этого раздела.

Ревью – задача завершена и находится на ревью. После проведения ревью существует два варианта развития события. Либо все выполнено отлично и задаче присваивается статус «выполнено», либо ревьюер добавляет комментарии и возвращает задачу в раздел «Требуется сделать».

Выполнено – задача полностью завершена и готова к включению в основной проект.

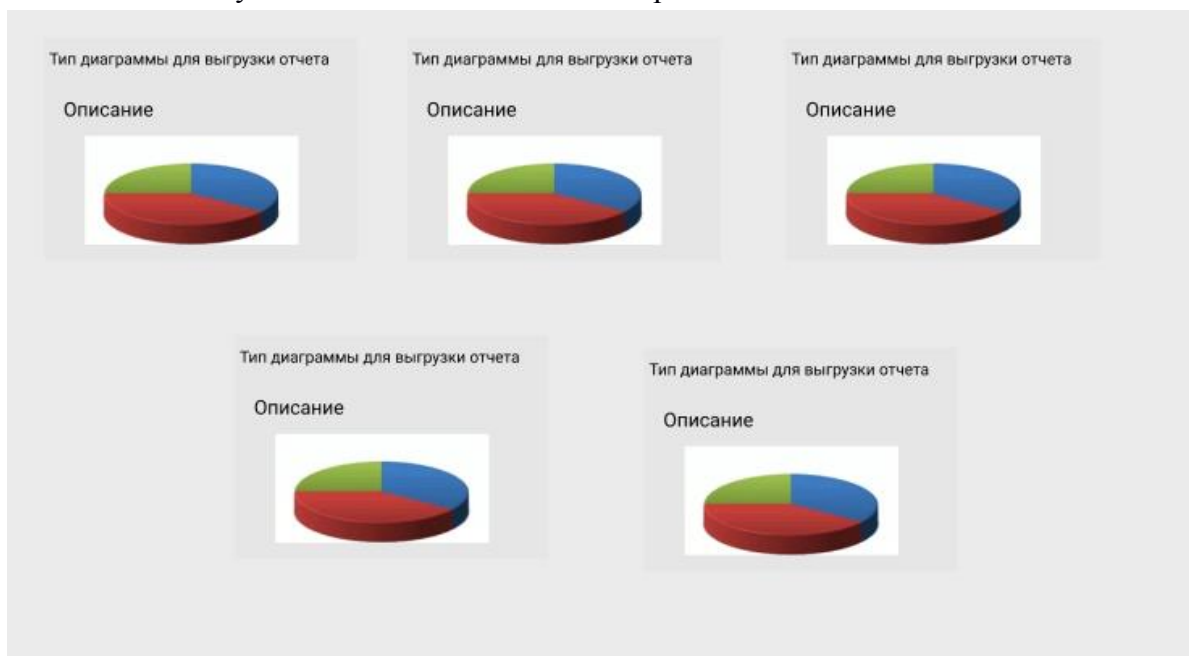
На рисунке 6 отображает раздел «задачи». Смысл этого раздела — показать все существующие на данный момент задачи из всех разделов в виде списка. При нажатии на любую из задач пользователь сможет ознакомиться с развернутой информацией по этой задаче.



**Рис. 6. Прототип страницы задач**

Кроме этого, в этом разделе имеется возможность присвоить задаче статус «выполнено», отредактировать ее или удалить. В случае редактирования задачи, пользователь сможет переназначить человека, ответственного за выполнения, изменить название и описание, а также сменить приоритет.

В разделе «Отчеты» (рис. 7) в виде карточек представлены варианты формирования отчетности с последующей возможностью его экспорта.



**Рис. 7. Прототип страницы отчетов**

На текущем этапе проектирования было принято решение не углубляться в аналитику и не закладывать какие-то конкретные варианты формирования отчетности. Отчетность формируется с целью анализа рабочего процесса, выявления слабых мест и устранения их.

Ключевыми различиями между вариантами отчета заключаются в тех данных, которые будут выгружаться, а также вид отображения этих данных(различные виды диаграмм).

На рисунке 8 отображен раздел, посвященный формированию и взаимодействию с релизами.



Прототип страницы формирования релизов. Вверху заголовок "Релизы". Ниже таблица с пятью строками. Каждая строка имеет три столбца: "Номер", "Название" и "Дата".

Номер	Название	Дата
Номер	Название	Дата
Номер	Название	Дата
Номер	Название	Дата
Номер	Название	Дата
Номер	Название	Дата

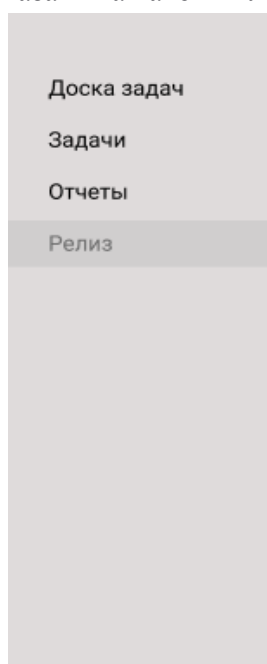
**Рис. 8. Прототип страницы формирования релизов**

Релизы отображаются в виде списка с возможностью получения более подробной информации при нажатии на них.

Каждый из релизов имеет уникальный номер, название и дату, когда был сформирован.

Внутри релиза можно увидеть все задачи, которые входят в него, пользователей, которые занимались их выполнением и другую информацию.

Для перемещения по системе предусмотрена навигационная панель (рис. 9). Текущая вкладка подсвечивается, дабы явно указать на какой вкладке находится пользователь.



**Рис. 9. Прототип навигационной панели**



В рамках данной работы разработан прототип системы, реализовывающий основные функции, такие как:

- аутентификация и авторизация пользователей;
- создание проекта;
- создание задач в проекте;
- редактирование задач в проекте
- удаление задач в проекте
- назначение статусов задач

### **Библиографический список**

1. *Макото Накаяма* Влияние инструментов управления проектами на оценки и выгоды проекта, с. 1-2
2. *Иванов А.Н.* Моделирование интерфейса полнофункциональных Web-приложений, интенсивно работающих с данными /Иванов Александр Николаевич, Кознов Дмитрий
3. *Васильков А.В.* Информационные системы и их безопасность: Учебное пособие / А.В. Васильков, А.А. Васильков, И.А. Васильков. – М.: Форум, 2017. – 528 с

### **DESIGN, DOCUMENTATION AND DEVELOPMENT OF SYSTEMS FOR JOINT PROJECT DEVELOPMENT**

*Margin Efim R., Soloveva Tatyana N.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
pro100efim@gmail.com

The purpose of this work is to design, document and develop a system for collaborative project development. The paper analyzes analogues of information systems, reveals their advantages and disadvantages. Based on the results of the analysis, design tasks were set and requirements for the system being developed were drawn up. In accordance with the requirements and the comparative analysis carried out for a number of criteria and parameters, the means of modeling and implementing the information system were selected. An analysis of the state of current business processes was carried out and an optimization option was proposed. Based on this, a system model was designed. A prototype of an information system for collecting and processing images for machine learning of external systems has been implemented.

Keywords: modularity, joint development, information system.

## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ КАРТ КОНДИТЕРСКИХ ИЗДЕЛИЙ ДЛЯ КОНДИТЕРА**

*Миргалеев Артем Александрович*

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

*Ознобихина Мария Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, ozmasha-90@mail.ru

Рассматриваются основные средства для проектирования и документирования информационной системы для автоматизации работы кондитера с технологическими картами кондитерских изделий. Определён предмет и объект исследования, сформулирована актуальность разработки данной информационной системы, а также поставлены задачи, которые необходимо решить в ходе проектирования. В ходе исследования, сделан анализ существующих информационных систем, которые выполняют похожие функции. Определены функциональные возможности информационной системы. Производится выбор средств проектирования и разработки. В качестве средств проектирования рассматриваются инструменты для UML-моделирования. [1] В результате анализа средств проектирования было выбрано инструментальное программное обеспечение проектирования информационной системы CASE-средство StarUML для проектирования системы, и был выбран сервис Figma.ru для разработки прототипа интерфейса приложения. В результате проектирования описываемой информационной системы было описано поведение системы и взаимодействие с ней кондитера. На основе построенных диаграмм был спроектирован интерфейс информационной системы.

Ключевые слова: автоматизация, технологические карты, кондитер, кондитерские изделия.

В настоящее время сложно определить, в какой момент возникло кондитерское дело. Однако мастера по изготовлению сладких изделий ценились всегда и везде. Как правило, кондитеры готовят изделия по так называемым технологическим картам.

Определение технологической карты по ГОСТ 31987-2012: Технологическая карта – документ, содержащий рецептуру и описание технологического процесса изготовления продукции, оформления и подачи изделия. [2] По закону требуется составлять технологическую карту для кондитерских или кафе.

Актуальность работы обусловлена, во-первых, необходимостью учёта расходного материала (сырья) для оптимизации процесса продаж и списания сырья со склада, во-вторых, необходимостью автоматизации создания и заполнения технологических карт.

Объектом исследования является автоматизация формирования технологических карт в кулинарии.

Предмет исследования автоматизация формирования технологических карт кондитерских изделий.

Целью исследования является проектирование информационной системы для создания автоматических технологических карт кондитерских изделий.

Если использовать распечатанные технологические карты, то возникает ряд проблем. Все эти проблемы решаются автоматизированной системой для технологических карт. Ведя технологические карты в системе автоматизации, становится проще:

- отслеживать ингредиенты. При производстве изделий будет удобно контролировать, сколько потратилось и сколько осталось ингредиентов на складе;
- оценить разницу между себестоимостью и конечной ценой. Так можно заметить выгодные и невыгодные позиции меню;
- выявлять кражи и нарушения со стороны сотрудников.

Выделим задачи, которые должна решать информационная система:

1. Авторизация;
2. Просмотр, редактирование, добавление и удаление продуктов;
3. Просмотр, редактирование, добавление и удаление технологических карт;
4. Печать технологических карт по требованию кондитера;
5. Списание продуктов со склада при каждом изготовлении изделия;
6. Контроль количества продуктов на складе и уведомление кондитера в случае, если продуктов может не хватить для приготовления изделия;
7. Хранение данных о продуктах и технологических карт.
8. Вывод технологических карт по определенному запросу, например, по виду теста на торт.

На основе технологических карт высчитывать калькуляционные карточки, которые включают итоговую стоимость кондитерского изделия.

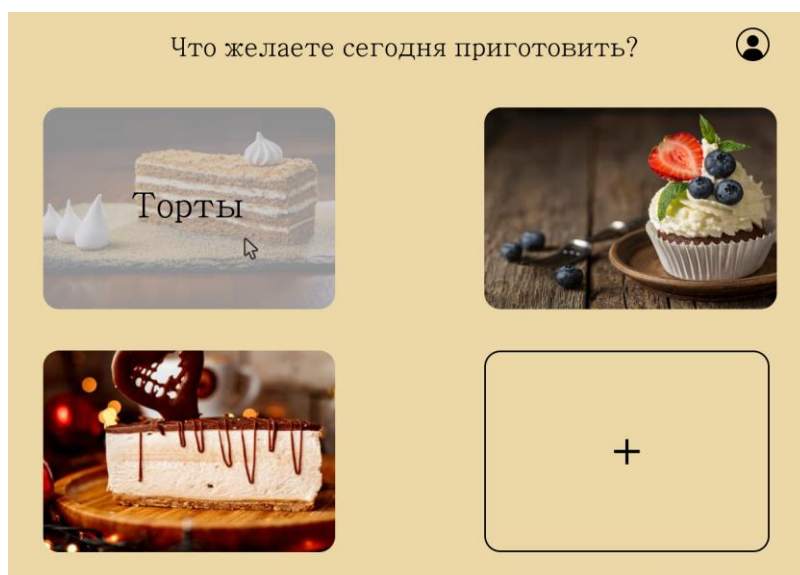
Логика работы системы построена в StarUML с помощью диаграмм вариантов использования и представлена на диаграмме вариантов использования (см. рис. 1).



Рис. 1. Диаграмма вариантов использования для кондитера

Заметим, что при редактировании цены на продукт, который уже есть в системе, цена на него изменяется во всех технологических картах, в которых этот продукт используется. Под печатью понимается то, что кондитер может выбрать технологическую карту и нажать на кнопку «печать», тем самым вызвав окно печати. При нажатии на «Приготовление» происходит списание того количества продуктов, которое записано в технологической карте. Так же кондитер может посмотреть итоговую цену изделия, и она же ему будет показываться при выводе технологических карт, запрос по которым кондитер может сделать в главном меню.

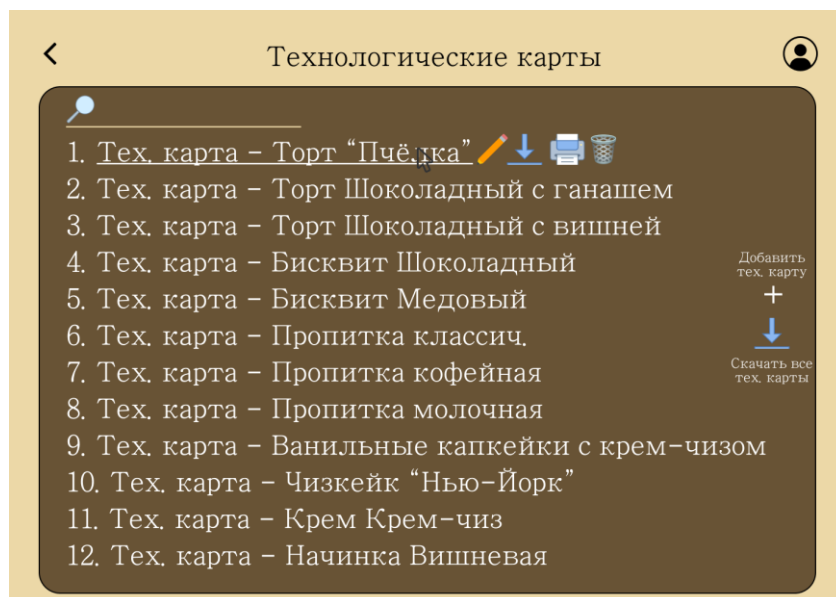
Любому кондитеру должно быть просто взаимодействовать с информационной системой, поэтому система должна иметь понятный интерфейс. Рассмотрим интерфейс некоторых функций системы. После авторизации кондитера встречается главное меню системы (см. рис. 2). В нем можно выбрать какое кондитерское изделие необходимо приготовить. Изначально даётся выбор: торты, капкейки или чизкейки, однако имеется возможность добавить новую группу изделий. При наведении на фото с изделием показывается подсказка, какая группа изделий будет показана после перехода



**Рис. 2. Главное меню системы**

При переходе на страницу «Тех. карты», кондитер попадает на страницу со списком всех технологических карт, которые есть в системе и добавлены кондитером (см. рис. 3).

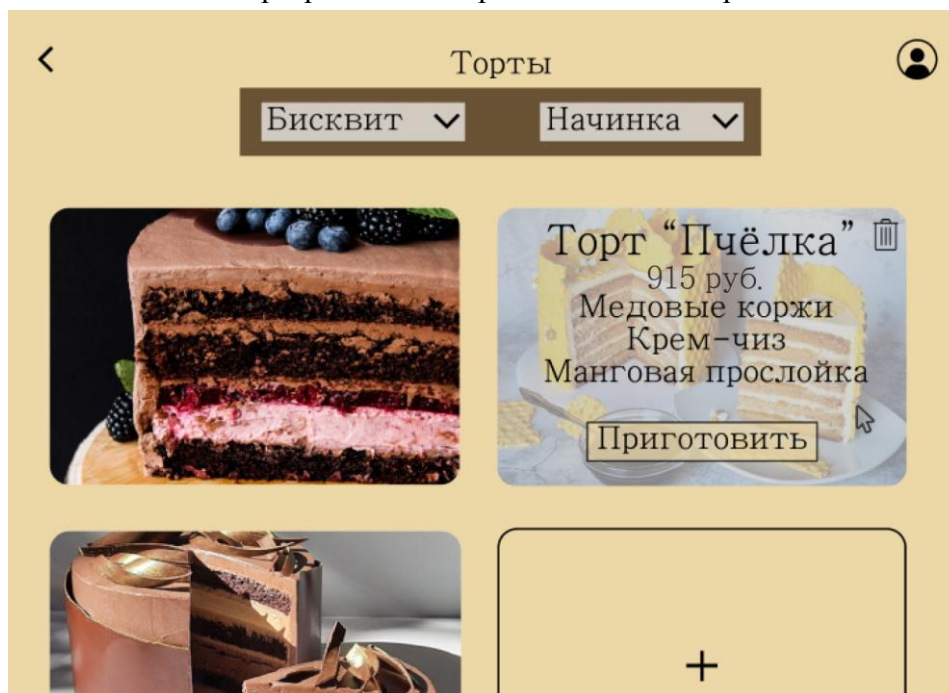
При наведении на технологическую карту изделия появляются иконки редактирования, сохранения, печати и удаления технологической карты. Справа имеются дополнительные иконки, при наведении на «+» показывается подсказка, нажав на эту иконку, кондитер может добавить новую технологическую карту. Нажав на нижнюю иконку, кондитер может скачать сразу все технологические карты в формате .doc. Сверху имеется поиск по всем технологическим картам, это удобно при большом объеме данных.



**Рис. 3. Страница «Технологические карты»**

При переходе в группу изделий "Торты", из главного меню, открывается страница со всеми тортами, на которые составлена технологическая карта (см. рис. 4). При наведении на фото торта показывается название торта, его себестоимость и краткое описание того, из чего состоит торт. На этой странице можно выбрать, какой бисквит и начинку надо приготовить, по выбранным параметрам произойдет отбор технологических карт, и кондитер увидит только те торты, в которых есть выбранный бисквит или начинка.

Также можно добавить новые торты, нажав на "+". При добавлении торта нужно выбрать бисквит и начинку торта. Дополнительно можно добавить, например, пропитку для бисквита. При наборе текста выводятся возможные варианты ингредиента. Также нужно выбрать фото торта. После заполнения кондитер нажимает на кнопку "Добавить" и остается только ввести название торта, и новая технологическая карта на торт будет создана автоматически, если заполнены поля разработчик и организация в настройках.



**Рис. 4. Страница «Торты»**

## Библиографический список

1. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие / Леоненков А.В. — Москва : Интернет-Университет Информационных Технологий, 2006. — 317 с.

2. ГОСТ 31987-2012 Услуги общественного питания. Технологические документы на продукцию общественного питания. Общие требования к оформлению, построению и содержанию URL: <https://docs.cntd.ru/document/1200103473> (дата обращения: 06.06.2022)

### **DESIGN AND DOCUMENTATION OF THE INFORMATION SYSTEM AUTOMATION OF TECHNOLOGICAL CHARTS OF CONFECTIONERY PRODUCTS FOR THE CONDITIONER**

*Mirgaleev Artem A.*

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

*Oznobikhina Maria S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, ozmasha-90@mail.ru

Abstract. The main means for the design and documentation of the information system for the automation of the confectioner's work with technological charts of confectionery products are considered. Defined the subject and object of research, formulated the relevance of the development of this information system, as well as tasks to be solved during the design. During the study, an analysis of existing information systems that perform similar functions was made. The functionality of the information system is defined. The choice of design and development tools is made. As design tools are considered tools for UML-modeling. As a result of the analysis of tools, the information system design software CASE tool StarUML for system design was selected, and the service Figma.ru was selected to develop a prototype of the application interface. As a result of the design of the described information system, the behavior of the system and the interaction of the confectioner with it were described. Based on the built diagrams the interface of the information system was designed.

Keywords: automation, technological charts, confectioner, confectionery products.

## ЭЛЕКТРОННАЯ КУЛИНАРНАЯ КНИГА

*Мосеева Юлия Сергеевна, Огурецкая Анна Игоревна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, super.xblegje2014@yandex.ru

Рассматриваются вопросы проектирования информационной системы «Кулинарная книга» с функциями расчета ингредиентов по калорийности и количеству и размеру порций, а также подсчет нормы калорийности пользователя и учет его суточной калорийности на основании потребления изготовленных рецептов и иных приемов пищи. Работа включала обзор существующих решений, рассмотрение теории кулинарных книг, способов хранения и передачи рецептов, вопросов, связанных формулами, по которым производится расчет ингредиентов и нормы калорийности. Для проектирования системы были отобраны средства, наиболее соответствующие потребностям. Для данной системы произведена постановка задачи с учетом особенностей, выявленных на этапе обзора систем, построены диаграммы *idef0* и *uml*, описывающие поведение, функциональные особенности и создан прототип пользовательского интерфейса.

Ключевые слова: информационная система «Кулинарная книга», расчет ингредиентов, подсчет нормы калорийности, учет суточной калорийности.

В настоящее время люди стали активнее проявлять интерес к здоровому образу жизни, в том числе и к питанию. Это можно связать и со временем пандемии, с переходом на удаленную работу, когда люди стали меньше двигаться и начали переедать из-за стресса. В этом может помочь электронная кулинарная книга, в которой представлены рецепты полезных, сбалансированных по БЖУ, блюд, учет калорий. В свободном доступе в сети Интернет можно найти множество рецептов, но очень сложно выбрать и найти нужное блюдо, рассчитать количество ингредиентов для конкретного количества порций.

Кулинарная книга представляет собой справочник, содержащий различные рецепты приготовления пищи. Обычно ее структура состоит из видов блюд или категорий и рецептов.

Перед тем, как рассмотреть основные функции информационной системы, приведем диаграмму вариантов использования (см. рис. 1), на которой изображены варианты взаимодействия пользователя и системы.

При написании текста курсовой были использованы различные материалы, в особенности книга А. В. Леоненкова «Объектно-ориентированный анализ и проектирование с использованием UML и IBM RATIONAL ROSE» [1], а также курсы, созданные в рамках проекта «Школа здорового питания». [2]

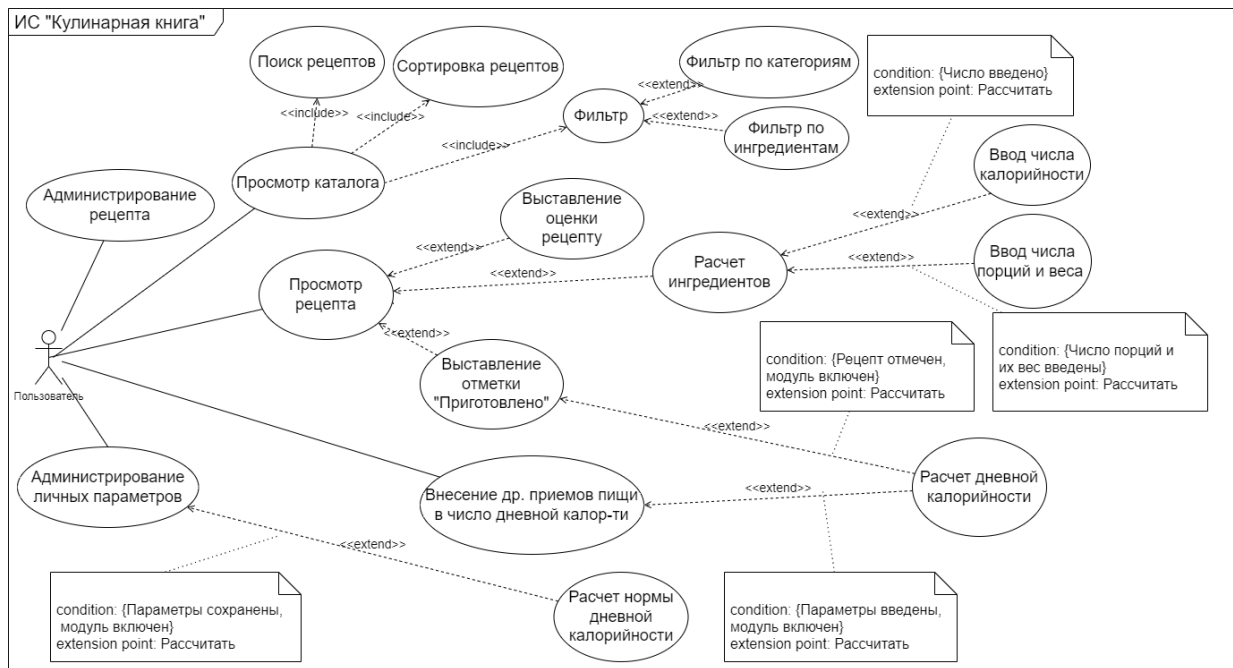


Рис. 1. Диаграмма вариантов использования

### Каталог рецептов

При открытии каталог представляет топ-3 самых популярных рецептов за неделю. При необходимости найти рецепт пользователь может воспользоваться фильтрацией по ингредиентам и отметить нужные элементы в списке или фильтрацией по названию и ввести название в поле. При переходе в категорию становится доступна сортировка рецептов по рейтингу (см. рис. 2).

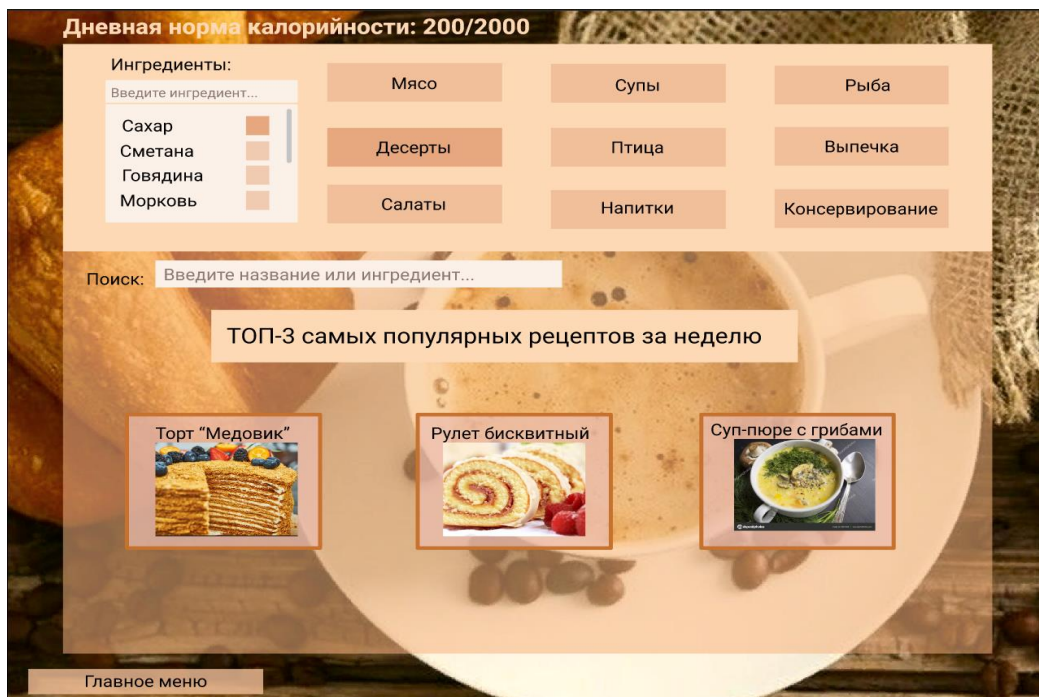


Рис. 2. Каталог рецептов

### Учет калорий

Если пользователь хочет получить свою норму калорийности и рассчитывать дневную калорийность, то при переходе в модуль «Учет калорий» необходимо включить его и запол-



нить форму с личными параметрами. Ниже расположена форма для учета дополнительных приемов пищи, куда пользователь может вводить название и общую калорийность блюда. В правом нижнем углу расположена кнопка «Просмотреть историю», с помощью которой пользователь может увидеть все дополнительные приемы пищи и приготовленные рецепты. Рецепты приложения отмечены иконкой «Книга» и имеют возможность удаления, иные приемы пищи отмечены иконкой «Человек» и имеют возможность изменения и удаления (см. рис. 3).

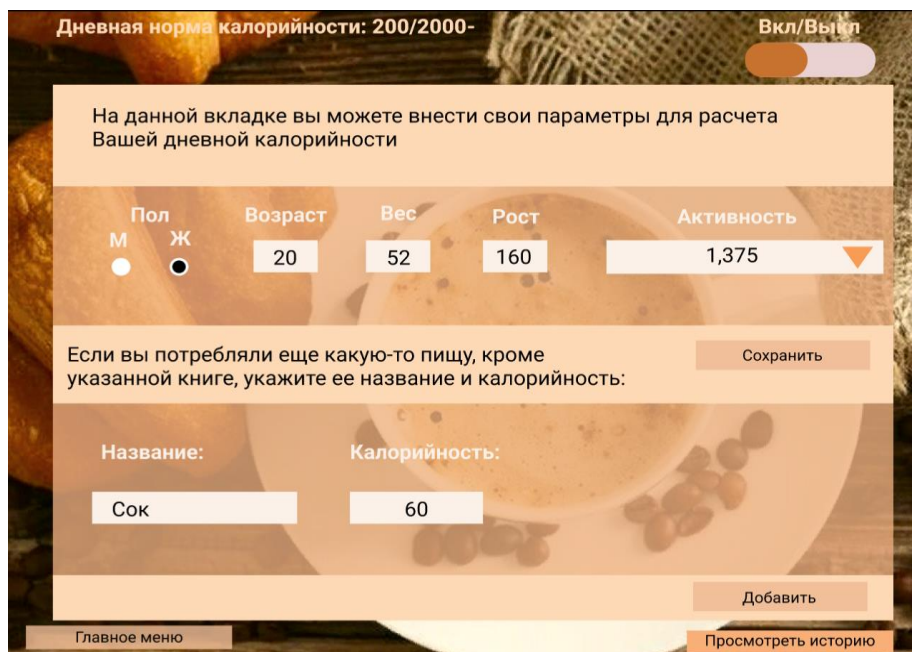


Рис. 3. Учет калорий

Для расчета нормы дневной калорийности могут использоваться различные методы. Например, один из способов предполагает расчет КБЖУ. Он подстроен под конкретного человека, для подсчета используют следующие данные: пол, вес, рост, возраст, коэффициент активности. Для этого используются формула Миффлин-Сан Жеора для расчета калорий:

$$K = (10 * \text{вес(кг)} + 6.25 * \text{рост(см)} - 5) * \text{возраст(г)} * A - \text{ для мужчин (1),}$$

$$K = (10 * \text{вес(кг)} + 6.25 * \text{рост(см)} - 5 * \text{возраст(г)} - 161) * A - \text{ для женщин (2),}$$

Коэффициент А показывает активность человека:

- 1,2 – люди с сидячим образом жизни, не занимающиеся спортом.
- 1,375 – сидячий образ жизни с тренировками 1-3 раза в неделю или образ жизни, предполагающий активную ходьбу.
- 1,55 – занятий спортом 3-4 раза в неделю.
- 1,7 – спортсмены, строители, сельхозработники.
- 1,9 – спортсмены с ежедневными многократными тренировками, шахтеры.

### Расчет ингредиентов

При переходе в каталог рецептов и выборе рецепта можно рассчитать количество ингредиентов по калорийности или порциям, тогда экземпляр рецепта отобразится во вкладке «Мои рецепты».

Формулы, с помощью которых производится расчет, представлены ниже.

Количество ингредиентов рецепта зависит от количества порций и размера порции (задаются пользователем).

Расчетные формулы, учитывая вес и размер порций:

$$V_{общ} = \sum (p_1 + p_2 + \dots + p_n) \quad (3),$$

где  $V_{общ}$  – общий вес,  $p_n$  – вес n-ой порции.

$$k_{пересч} = \frac{V_{общ}}{V_{исх}} \quad (4),$$

где  $k_{пересч}$  – коэффициент пересчета ингредиентов,  $V_{исх}$  – исходный вес по рецепту.

$$P_{инг} = P_{ингИсх} * k_{пересч} \quad (5),$$

где  $P_{инг}$  – вес отдельного ингредиента,  $P_{ингИсх}$  – исходный вес ингредиента по рецепту

$$K_{реци} = K_{рециИсх} + k_{пересч} \quad (6),$$

где  $K_{реци}$  – калорийность полученного рецепта,  $K_{рециИсх}$  – калорийность исходного рецепта

Расчетные формулы, учитывая калорийность:

$$k_{пересч} = \frac{K_{треб}}{K_{рециИсх}} \quad (7),$$

где  $K_{треб}$  требуемая калорийность.

Диаграмма idef0 [3] по расчету ингредиентов представлена на рис. 4.

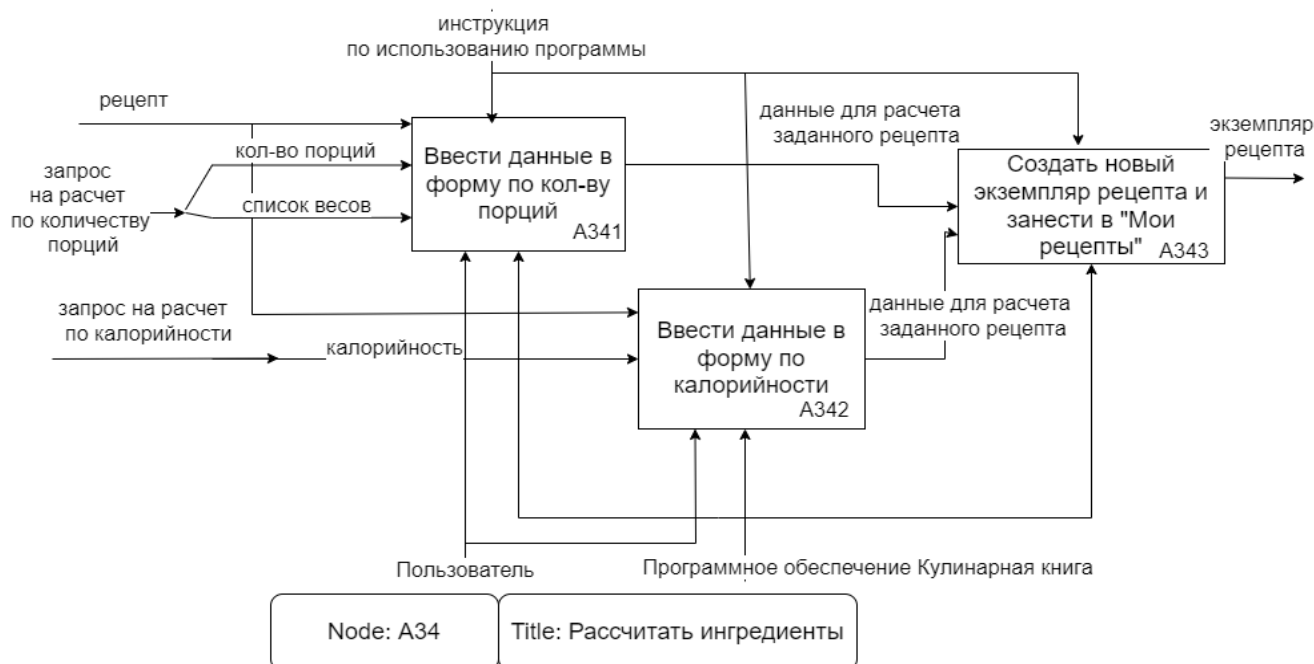


Рис. 4. Диаграмма «Расчет ингредиентов»

### Добавление рецепта

Пользователь может внести свой рецепт при помощи кнопки «Добавить рецепт» и заполнения соответствующих полей (см. рис. 5).

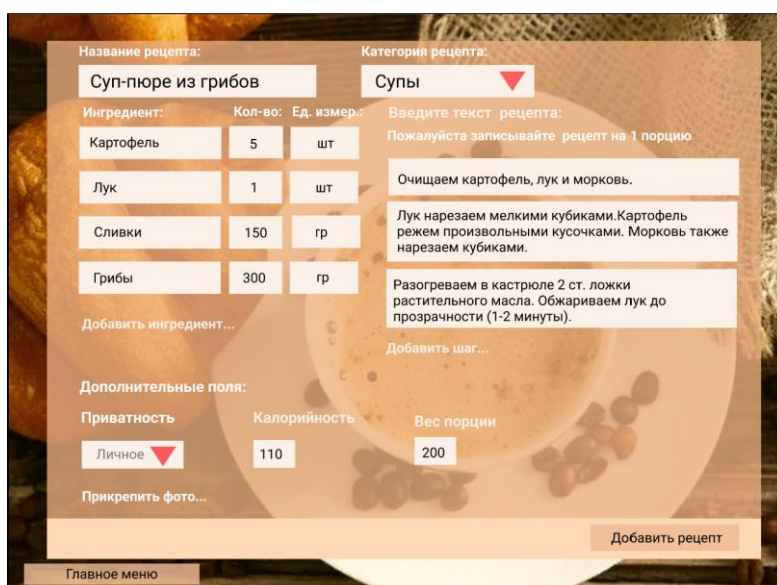


Рис. 5. Добавление рецепта

### Библиографический список

1. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие / Леоненков А.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 317 с. — ISBN 978-5-4497-0667-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/97554.html> (дата обращения: 07.01.2022). — Режим доступа: для авторизир. Пользователей
2. Курсы, созданные в рамках проекта : «Школа здорового питания» РФ. Национальный проект «Демография», федеральный проект «Укрепление общественного здоровья» © «Федеральная служба по надзору в сфере защиты прав потребителей и благополучия человека», 2021
3. IDEF0. Знакомство с нотацией и пример использования [Электронный ресурс] URL: <https://infostart.ru/1c/articles/640962/> (дата обращения: 25.11.2021)

### "ELECTRONIC COOKBOOK"

*Oguretskaya Anna I., Moseeva Yulia S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, ivanov@email.ru

The issues of designing the information system "Cookbook" with the functions of calculating ingredients by calorie content and the number and size of portions, as well as calculating the user's calorie intake and accounting for his daily calorie intake based on the consumption of recipes and other meals are considered. The work included a review of existing solutions, consideration of the theory of cookbooks, methods of storing and transmitting recipes, issues related to formulas used to calculate ingredients and calorie norms. For the design of the system, the means most appropriate to the needs were selected. For this system, the task was formulated taking into account the features identified at the stage of the system review, ideo and uml diagrams describing behavior, functional features were constructed, and a prototype of the user interface was created.

Keywords: information system "Cookbook", calculation of ingredients, calculation of caloric content, accounting of daily caloric content.

## **ОТНОШЕНИЕ СТУДЕНТОВ К ПРИМЕНЕНИЮ ДЕЛОВЫХ ИГР И КОМПЬЮТЕРНЫХ СИМУЛЯТОРОВ В ИЗУЧЕНИИ ПРОГРАММНОЙ ИНЖЕНЕРИИ. СРАВНЕНИЕ ПЕРМСКОГО ГОСУНИВЕРСИТЕТА И ПЕРМСКОГО ФИЛИАЛА ВЫСШЕЙ ШКОЛЫ ЭКОНОМИКИ**

*Мустафина Назгуль Ибрагимовна*

Национальный исследовательский университет Высшая школа экономики (Пермский филиал), 614070, Россия, г. Пермь, ул. Студенческая, 38, nazgul-2003@mail.ru

*Плаксин Михаил Александрович*

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

Исследовано отношение студентов ряда компьютерных специальностей двух вузов (Пермского государственного университета и Пермского филиала Высшей школы экономики) к применению деловых игр и компьютерных симуляторов в ходе изучения программной инженерии. Форма деловой игры оценивается всеми студентами как интересная и полезная, превосходящая по эффективности традиционные лекционно-практические занятия, заслуживающая более широкого применения. В оценке студентами разных вузов одних и тех же игр есть некоторая (непринципиальная) разница. В ПГУ оценка интересности и полезности несколько ниже, чем во ВШЭ. Но отклонение объясняется не разницей вузов, а разницей года обучения. Из-за разницы учебных планов в ПГУ эти игры проводятся на третьем курсе, а во ВШЭ – на первом.

Ключевые слова: деловая игра, компьютерный симулятор, учебный процесс, университет, программная инженерия, agile, MSF, scrum.

### **Введение**

Изучение программной инженерии требует обязательного подкрепления теории практикой. Одних лекций совершенно недостаточно. Среди прочего это относится к знакомству студентов с современными технологиями программирования, этапами разработки компьютерных систем, набором специальностей, задействованных в производстве программных продуктов. Лучше всего было бы знакомиться со всем этим на практической работе в настоящей фирме, но в реальной жизни организовать такую практику для большого числа студентов невозможно. Кроме того, цели реального производственного процесса отличаются от целей процесса учебного. Зачастую, в учебном процессе имеет смысл создавать ситуации, которые в реальном производственном процессе нежелательны.

Поэтому в Пермском филиале Высшей школы экономики и в Пермском государственном университете в качестве замены реального производственного процесса активно используются деловые игры и компьютерные симуляторы. После каждой игры студенты пишут рефлексии, в которой оценивают интересность проведенной игры и ее полезность с точки зрения знакомства с computer science. В данном исследовании были проанализированы рефлексии студентов двух

ВУЗов по шести деловым играм за 4 года. Цель исследования – выяснить отношение студентов к применению деловых игр в учебном процессе и сравнить отношение студентов разных вузов.

### **Материалы и методы**

В качестве материалов использовались рефлексии студентов двух вузов: Пермского филиала Высшей школы экономики (ВШЭ) и Пермского национального исследовательского университета (ПГНИУ). Во ВШЭ это были студенты первого курса направления «Программная инженерия», всего 220 человек за 4 года, дисциплина «Введение в программную инженерию». В ПГНИУ – студенты третьего курса направлений «Прикладная математика», «Информационные технологии», «Компьютерная безопасность», всего 130 человек за 4 года, дисциплина «Методы коллективной разработки программных систем». Таким образом, общее количество участников исследования составило 350 человек, продолжительность исследования – 4 года.

Рефлексии касались шести учебных заданий. А именно: «Технология Microsoft Solution Framework for Agile. Этапы Envisioning & Planning»[1], «Технология MSF. Этап Stabilizing», «Технология MSF. Этап Deploying», «Технология Scrum: ритуалы и артефакты»[2], «Руководство ИТ-проектом», «Преобразование программы в программный продукт». Все эти задания знакомят студентов с современными технологиями программирования, этапами разработки компьютерных систем, набором специальностей, задействованных в производстве программных продуктов. Два задания («Envisioning & Planning» и «Scrum») представляют собой безмашинные деловые игры. Одно («Руководство ИТ-проектом») состоит в работе с компьютерным симулятором. Еще два («Stabilizing» и «Deploying») включают работу с компьютерными симуляторами на отдельных этапах. В задании «Преобразование программы в программный продукт» используется обычная система программирования (Visual Studio). Все задания – командные. Размер команды варьируется от двух до девяти человек. В играх «Stabilizing» и «Deploying» действует одна и та же команда. В остальных четырех случаях команда создается для одной конкретной игры.

В каждой рефлексии студенты оценивали проведенную игру по параметрам «Интересность» и «Полезность» по шкале от 0 до 10, а также отвечали на ряд вопросов по содержанию и организации игры. Для анализа вычислялись статистические характеристики оценок интересности и полезности: среднее арифметическое, медиана, мода и др.

Разница в отношении к аудиторному и дистанционному формату игр обсуждалась на Международной конференции по дистанционному обучению в Минске в мае 2022 г.[3].

### **Исследование**

На рис. 1 и 2 приведены средние значения характеристик «Интересность» и «Полезность» по отдельным заданиям для каждого из вузов и для двух вузов вместе. (Для обозначения игр используются очевидные сокращения). В табл.1 представлены обобщенные значения по всем заданиям: средняя оценка, медианная, модальная, минимальная и максимальная. В табл.2 – информация о частоте появления разных оценок. На рис. 3 и 4 – разделение оценок интересности и полезности на низкие (от 0 до 3-х), средние (от 4-х до 7-и) и высокие (от 8-и до 10-и).

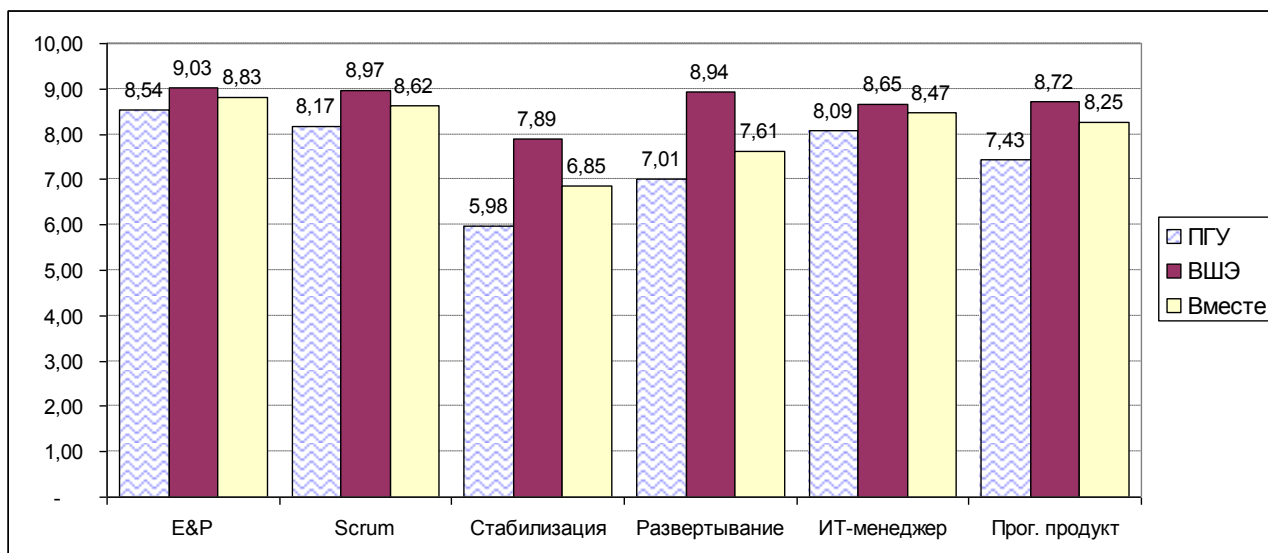


Рис. 6. Средняя оценка интересности разных заданий.

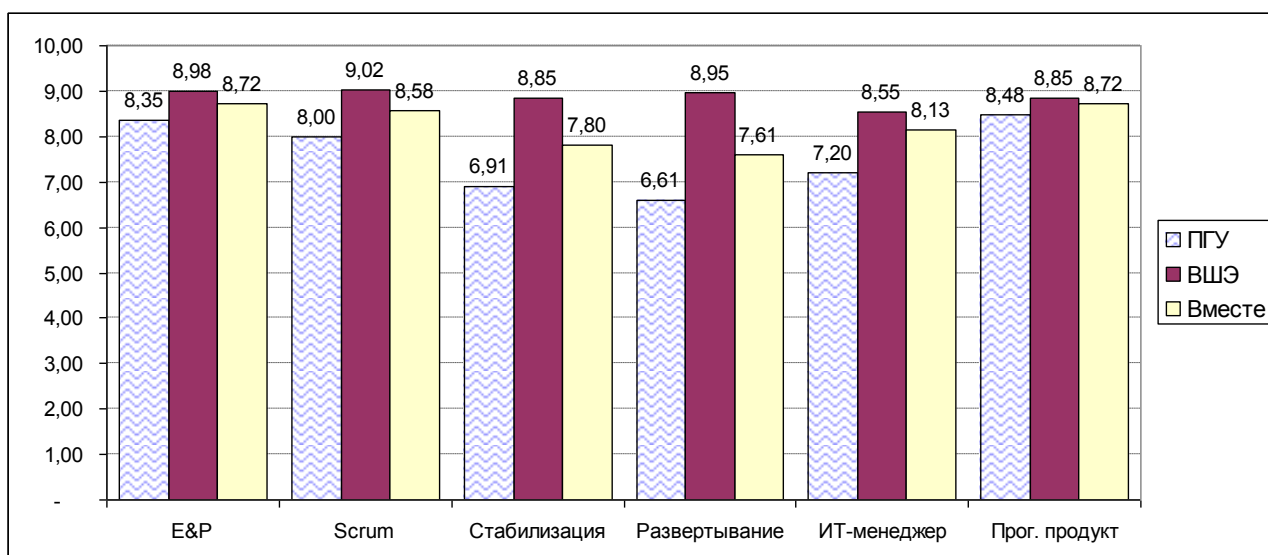


Рис. 2. Средняя оценка полезности разных заданий.

Таблица 1. Обобщенные оценки по всем заданиям

Параметр	Интересность			Полезность		
	ПГУ	ВШЭ	Вместе	ПГУ	ВШЭ	Вместе
Среднее значение	7.45	8.7	8.14	7.56	8.85	8.27
Медиана	8	9	9	8	9	9
Мода	8	10	10	8	10	10
Минимум	0	2	0	0	2	0
Максимум	10	10	10	10	10	10

Таблица 2. Частота появления оценок по всем играм, %

Оценка	Интересность			Полезность		
	ПГУ	ВШЭ	Вместе	ПГУ	ВШЭ	Вместе
0	0,87	0,00	0,39	0,87	0,00	0,39
1	1,01	0,00	0,45	0,29	0,00	0,13
2	2,02	0,35	1,09	1,45	0,35	0,84
3	3,61	0,23	1,74	2,75	0,46	1,48
4	3,90	1,16	2,38	4,49	0,46	2,25
5	7,37	2,78	4,82	6,67	2,08	4,11
6	7,66	4,51	5,91	8,12	3,69	5,66
7	16,47	9,03	12,34	15,65	8,65	11,76
8	21,24	19,44	20,24	23,91	16,26	19,67
9	14,60	20,95	18,12	13,77	22,15	18,44
10	21,24	41,55	32,52	22,03	45,91	35,35

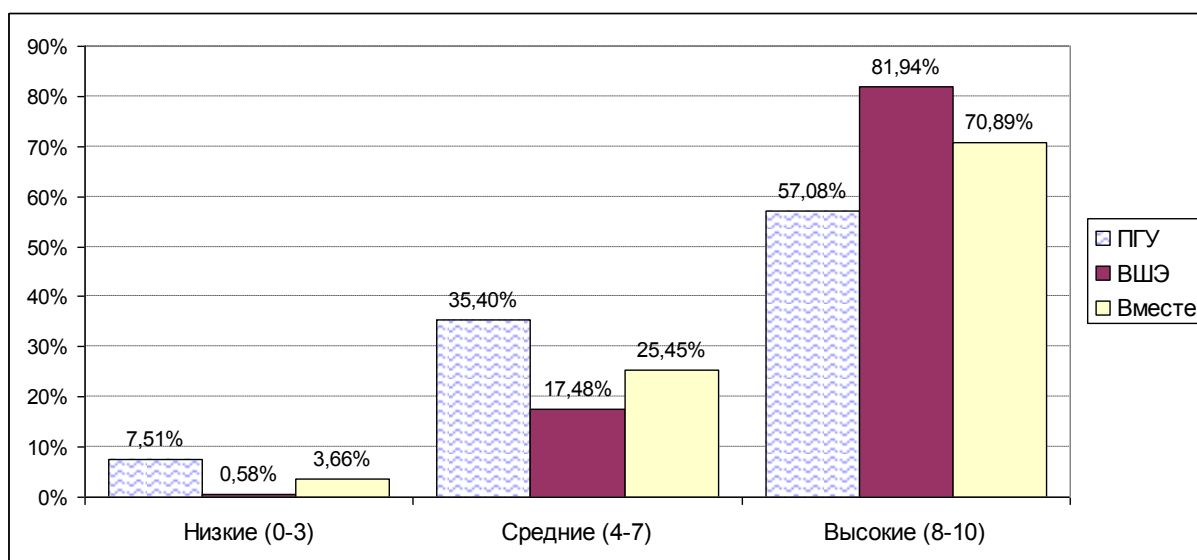


Рис. 3. Разделение оценок интересности на низкие, средние и высокие.

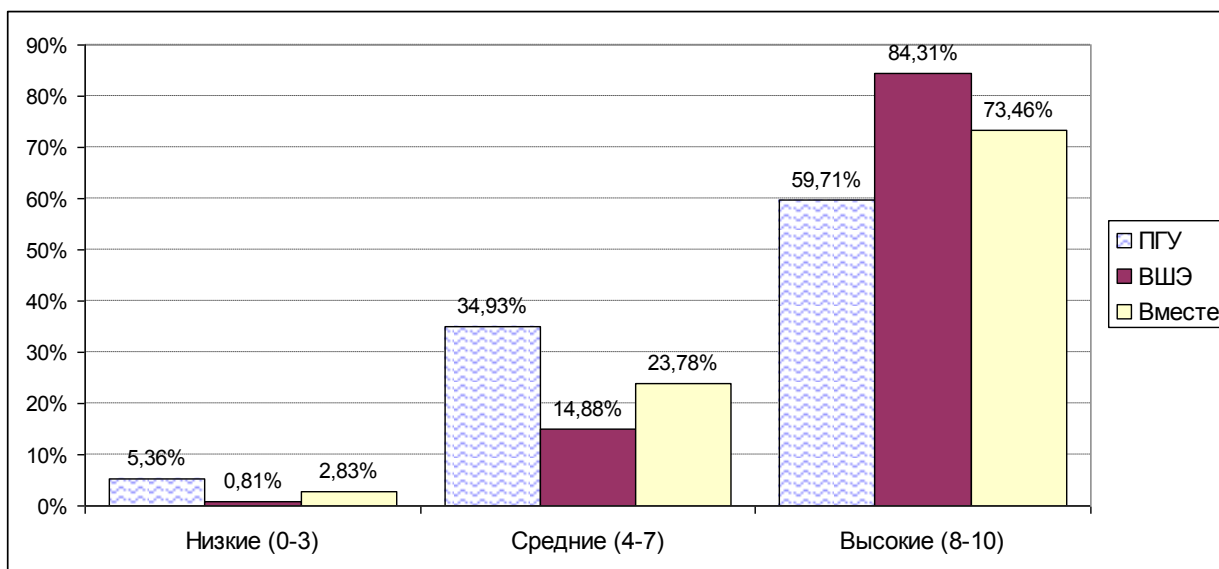


Рис. 4. Разделение оценок полезности на низкие, средние и высокие.

### Обсуждение

1. Из всех приведенных выше таблиц и диаграмм видно, что студенты обоих вузов оценивают форму деловых игр как интересную и полезную. Средняя оценка интересности по всем играм составляет 8.14, полезности – 8.27. Медианное значение для обеих характеристик составляет 9, модальное – 10. Модальное значение в обоих случаях равно «официальному» максимальному (на самом деле, были студенты поставившие и 12). В комментариях деловые игры были оценены как форма, превосходящая по эффективности традиционные лекционно-практические занятия, заслуживающая более широкого применения.

2. Оценки студентов ПГНИУ стабильно ниже оценок студентов ВШЭ примерно на 17% (1.3 балла). В НИУ ВШЭ средняя оценка интересности равна 8.70, полезности – 8.85, в ПГНИУ интересности – 7.45, полезности – 7.56. Объяснение было найдено в комментариях. Отношение к одним и тем же играм зависит от года обучения. Из-за разницы в учебных планах одни и те же игры в НИУ ВШЭ проводятся со студентами первого курса, а в ПГНИУ – третьего. Для первокурсников они были совершенно новыми, а потому более интересными и более сложными. Третьекурсники (которые зачастую уже имеют опыт практической работы) чаще отмечали, что материал был им в какой-то мере уже знаком.

3. Высокие средние, медианные и модальные оценки сопровождаются «выбросами вниз» вплоть до нуля. Доля низких оценок (от нуля до трех) составляют 5.5-7.5% для студентов ПГУ и 0,5-0.8% для студентов ВШЭ. Объяснения опять были найдены в комментариях. Это не принципиальные возражения против игр, а частные претензии: к качеству компьютерных симуляторов, к уровню сложности заданий, к времязатратности, к недостатку теоретических знаний и др. Несмотря на малочисленность, «выбросы вниз» отслеживаются и предпринимаются меры к ликвидации их причин (обновляются компьютерные симуляторы, совершенствуются формы проведения занятий и пр.).

### Заключение

Цель данного исследования – выяснить отношение студентов ВШЭ и ПГУ к применению деловых игр в учебном процессе. Форма деловой игры оценивается студентами как интересная и полезная, превосходящая по эффективности традиционные лекционно-



практические занятия, заслуживающая более широкого применения. Несколько более низкие оценки студентов ПГУ объясняются разницей в годе обучения.

### **Библиографический список**

1. Microsoft Corporation. Microsoft Solutions Framework. MSF Project Management Discipline v.1.1.1. 2002.
2. *Рубин К.С.* Основы Scrum: практическое руководство по гибкой разработке ПО. // К.С. Рубин. – Диалектика-Вильямс. 2013.
3. *Мустафина Н.И., Плаксин М.А.* О влиянии дистанционного режима на восприятие студентами деловых игр. // Дистанционное обучение – образовательная среда XXI века: материалы XII Междунар. науч.-метод. конф. (Республика Беларусь, Минск, 26 мая 2022 года) – Минск: БГУИР, 2022. – С.37.

## **STUDENTS' ATTITUDE TO THE APPLICATION OF BUSINESS GAMES AND COMPUTER SIMULATORS IN THE STUDY OF SOFTWARE ENGINEERING. COMPARISON OF THE PERM STATE UNIVERSITY AND THE PERM BRANCH OF THE HIGHER SCHOOL OF ECONOMICS**

*Mustafina Nazgul I.*

National Research University Higher School of Economics (Perm branch), Student st. 38, Perm, Russia, 614070, nazgul-2003@mail.ru

*Plaksin Mikhail A.*

Perm State National Research University, Bukireva st. 15, Perm, Russia, 614990, mapl@list.ru

**Abstract.** The attitude of students of a number of computer specialties of two universities (Perm State University and the Perm branch of the Higher School of Economics) to the use of business games in the course of software engineering was studied. The form of the business game is assessed by all students as interesting and useful, superior in efficiency to traditional lectures and practical classes, and deserving of wider application. There is some difference in the evaluation of the same games by students of different universities. At PSU, the rating of interestingness and usefulness is somewhat lower than at the Higher School of Economics. But judging by the comments of the students themselves, the deviation is explained not by the difference in universities, but by the difference in the year of study. Due to the difference in curricula at PSU, these games are held in the third year, and at the HSE – in the first.

**Keywords:** business game, computer simulator, educational process, university, software engineering, agile, MSF, scrum.

## **ПРОЕКТИРОВАНИЕ И ПРОТОТИПИРОВАНИЕ ИС СООТНЕСЕНИЯ КОМПЕТЕНЦИЙ, ФОРМИРУЕМЫХ ФГОС (СУОС) ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ, И ТРЕБОВАНИЙ ПРОФЕССИОНАЛЬНЫХ СТАНДАРТОВ**

*Пичкалев Михаил Викторович, Василюк Надежда Николаевна*

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

В статье описан процесс проектирования и прототипирования пользовательского интерфейса информационной системы соотнесения компетенций, формируемых ФГОС (СУОС) по направлению подготовки, и требований профессиональных стандартов. Исследуются Федеральный государственный образовательный стандарт (ФГОС) и профессиональные стандарты из Трудового кодекса РФ, их сходства и различия, а также возможности соотнесения. Система позволит хранить, обрабатывать и представлять пользователю данные о сравнении компетенций, формируемых ФГОС (СУОС) по направлению подготовки и требований профессиональных стандартов, предполагает наличие разных ролей пользователей в системе (пользователь и администратор). Она имеет потенциал для использования составителями учебных планов и разработчиками ФГОС и профессиональных стандартов. Приводятся результаты проектирования информационной системы и прототипирования интерфейса информационной системы.

Ключевые слова: ФГОС, профессиональные стандарты, информационная система, соотнесение стандартов.

В современном мире высшее образование играет важную роль при устройстве на работу. Но ещё более важную роль при трудоустройстве играет соответствие кандидата на место требованиям профессиональных стандартов, которые должны быть сформированы у него по итогам обучения в вузе. Однако, хоть в законе «Об образовании в Российской Федерации» сказано, что ФГОС формируются на основе требований профессиональных стандартов, на практике часто оказывается, что компетенции, формируемые ФГОС по направлению подготовки, не соответствуют требованиям профессиональных стандартов. А соответствующих систем, позволяющих соотнести компетенции ФГОС и профессиональные стандарты, не существует.

Чтобы узнать о сопоставлении ФГОС и профессиональных стандартов, рассмотрим документ «Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов». В документе указано, что «в процессе сопоставления требований ФГОС СПО и профессиональных стандартов необходимо учитывать различия их терминологии, связанные с предметом описания. В профессиональных стандартах описывается деятельность, для этого используются термины "вид профессиональной деятельности", "обобщенные трудовые функции", "трудовые функции", "трудовые действия". В образовательных стандартах приводятся характеристики выпускника, владеющего деятельностью, в качестве ключевого используется термин "компетенция" [4]. Таким образом, при програм-

мировании ИС нужно будет учитывать различия в формулировке компетенций во ФГОС и требований профессиональных стандартов и выполнять соотнесение по ключевым словам и словосочетаниям.

В другом абзаце этого документа упоминается, что «кроме профессиональных компетенций по видам деятельности в качестве результата образования ФГОС СПО фиксируют общие компетенции. Соответствующий им элемент в профессиональных стандартах отсутствует, поскольку в них изначально зафиксирован уровень образования, необходимый для выполнения ОТФ и ТФ, что, в свою очередь, подразумевает наличие у выпускника общих компетенций, соответствующих этому уровню» [4]. Следовательно, при проектировании баз данных (БД) нам нужно разделять общие компетенции и профессиональные компетенции, и при соотнесении ФГОС и профессиональных стандартов выбранной профессии их не выводить, однако всё равно их следует хранить в БД при выводе конкретных ФГОС, чтобы у пользователя не возникало вопросов при виде отсутствия общих компетенций.

Также в документе говорится, что «с учетом различия подходов, использованных при составлении характеристики профессиональной деятельности и определении требований к результатам освоения образовательной программы в ФГОС ВО и описании квалификации в профессиональных стандартах, в каждом конкретном случае разработчики должны самостоятельно определить, какие единицы профессионального стандарта и как корреспондируют с единицами ФГОС» [4]. Из этого следует, что не существует чёткого алгоритма сравнения компетенций и требований профессиональных стандартов, и нам следует разработать этот алгоритм самостоятельно.

Нам следует рассмотреть практическое соотнесение ФГОС и профессиональных стандартов. Для примера возьмём «Экспертное заключение по результатам независимой экспертизы федерального государственного образовательного стандарта по направлению подготовки 08.03.01 «Строительство» (уровень бакалавриата) (утвержден приказом Министерства образования и науки РФ от «12» марта 2015 г. №201) в части оценки его соответствия профессиональному стандарту 16.025 «Организатор строительного производства» (№ 244; утвержден приказом Министерства труда и социальной защиты России от «21» ноября 2014 г. №930н), представленной Советом по профессиональным квалификациям в строительстве».

Одним из критериев соотнесения являлся пункт: «В разделе «Виды профессиональной деятельности» ФГОС учтены все обобщенные трудовые функции в ПС, выполнение которых возлагается на работника данной квалификации» [5]. Экспертное заключение именно по этому пункту выявило несоответствие ФГОС профессиональным стандартам и требование от анализатора дополнить ФГОС следующей задачей профессиональной деятельности: «Участие в процессах материально-технического обеспечения строительного производства на участке строительства» на основе следующей формулировки трудовой функции в ПС, реализация которой возлагается на работника данной квалификации: «Материально-техническое обеспечение строительного производства на участке строительства», а также требования изменений некоторых формулировок ФГОС.

Также в этом экспертном заключении описан критерий «Требования к профессиональным компетенциям отражают требования ПС к выпускнику организации высшего профессионального образования в полном объеме». Здесь также выявлены несоответствия ФГОС и профессионального стандарта, например, «Не учтено требование к профессиональным компетенциям, связанное с трудовой функцией «Материально-техническое обеспечение строи-

тельного производства на участке строительства» и соответствующими ей трудовыми действиями».

Из приведённых экспертных заключений следует, что для специальностей среднего профессионального образования требуется полное соответствие ФГОС и профессиональных стандартов. Такое же соответствие требуется и для высшего образования, хотя бы в рамках профессиональных компетенций. Также для ИС потребуются администраторы, корректирующие ФГОС для каждой образовательной организации.

Необходимо разработать проект соотнесения компетенций, формируемых ФГОС (СУОС) по направлению подготовки, и требований профессиональных стандартов. Система должна обеспечивать работу со списками ФГОС по направлениям подготовки и требований образовательных стандартов. В списке ФГОС по направлениям подготовки хранятся следующие данные: название ФГОС, общепрофессиональные компетенции, общекультурные компетенции, профессиональные компетенции. В списке требований профессиональных стандартов содержится следующая информация: наименование профессии, необходимые умения.

Система должна обеспечивать добавление, удаление и изменение данных в списках ФГОС по направлениям подготовки и требований образовательных стандартов (по мере необходимости). ИС позволит осуществлять поиск профессий и направлений подготовки для просмотра компетенций. Также ИС должна давать пользователю возможность сравнить выбранное направление подготовки и профессиональные стандарты по списку всех компетенций и требований. Система должна выводить следующую информацию:

- 1) Недостающие во ФГОС требования для этой профессии.
- 2) Не требующиеся для этой профессии компетенции, которые присутствуют в ФГОС.

Далее мы разобьём всю систему на функции, которые должны быть реализованы, и данные, которые привязаны к ним. Это поможет нам выделить основные классы для диаграммы классов. Функции:

1. Ведение списка ФГОС (название ФГОС, общепрофессиональные компетенции, общекультурные компетенции, профессиональные компетенции).
  - a. Добавление данных.
  - b. Удаление данных.
  - c. Изменение данных.
2. Ведение списка требований профессиональных стандартов (наименование профессии, необходимые умения).
  - a. Добавление данных.
  - b. Удаление данных.
  - c. Изменение данных.
3. Осуществление поиска (название ФГОС, наименование профессии).
  - a. Поиск ФГОС в списке ФГОС.
  - b. Поиск профессий в списке профессий.
4. Соотнесение компетенций, формируемых ФГОС (СУОС) по направлению подготовки, и требований профессиональных стандартов.
  - a. Возможность выбора ФГОС и профессии.
  - b. Сравнение выбранных ФГОС и профессии.
  - c. Вывод недостающих во ФГОС требований для этой профессии.

d. Вывод не требующихся для этой профессии, но присутствующих во ФГОС компетенций.

e. Вывод профессии, наиболее соответствующей выбранному ФГОС.

После рассмотрения различных продуктов для проектирования, прототипирования и разработки ИС были выбраны следующие средства: UML, как методология проектирования, draw.io, как средство моделирования, Figma, как средство прототипирования, SQL, как средство проектирования баз данных и PHP, как язык разработки приложения.

Для проектирования системы важна диаграмма классов. На рисунке изображена иерархия классов пользователей, а также их функций. В ИС хранится список ФГОС и список профессий, у каждого ФГОС есть списки общепрофессиональных компетенций, профессиональных компетенций и общекультурных компетенций, у каждой профессии есть список необходимых умений. В списки можно добавлять данные и удалять из них данные. Другие данные можно изменять.

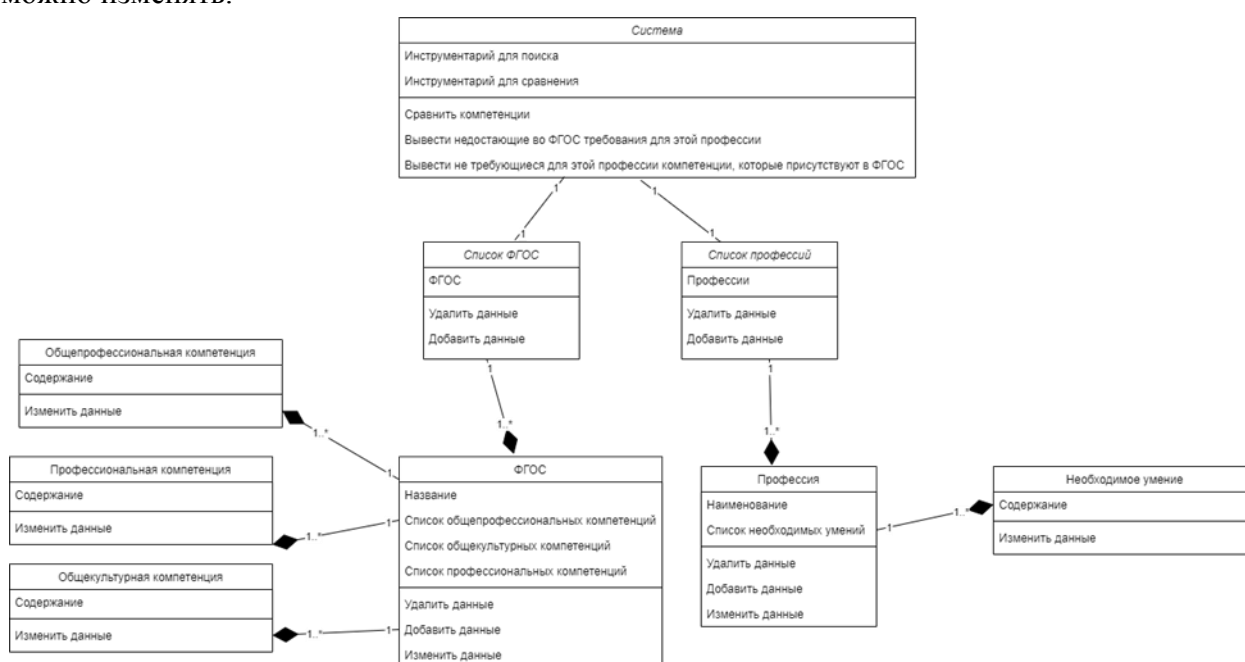
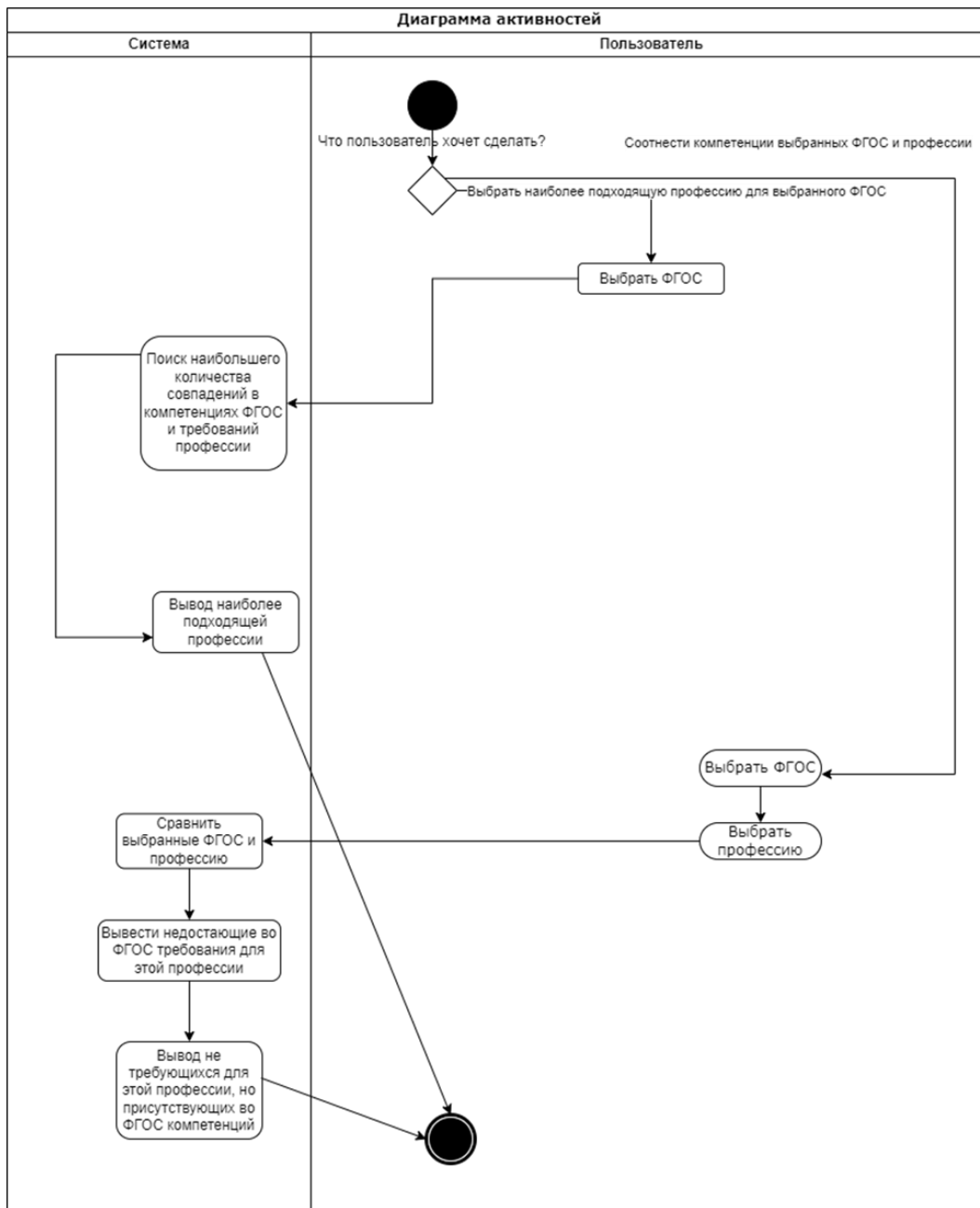


Рис. 1. Диаграмма классов

Чтобы показать динамические аспекты системы была создана диаграмма активностей. Она показывает взаимодействие пользователя с системой. Пользователь выбирает действие, которое ему нужно выполнить. Если он выбирает вариант «вывести наиболее подходящую профессию для выбранного ФГОС», то сначала задает ФГОС, затем система выполняет поиск наибольшего количества совпадений в компетенциях ФГОС и требований профессии, и выводит наиболее подходящую профессию. Если же пользователь выбирает соотнесение компетенций выбранных ФГОС и профессий, то он задает ФГОС, профессию, система производит их сравнение, а затем выводит недостающие во ФГОС требования для этой профессии и компетенции, которые присутствуют во ФГОС, но не требуются для этой профессии.



**Рис. 2. Диаграмма активностей**

На рисунках 3 и 4 представлены прототипы пользовательского интерфейса. В нём отражены некоторые варианты использования пользователем ИС из диаграмм прецедентов.

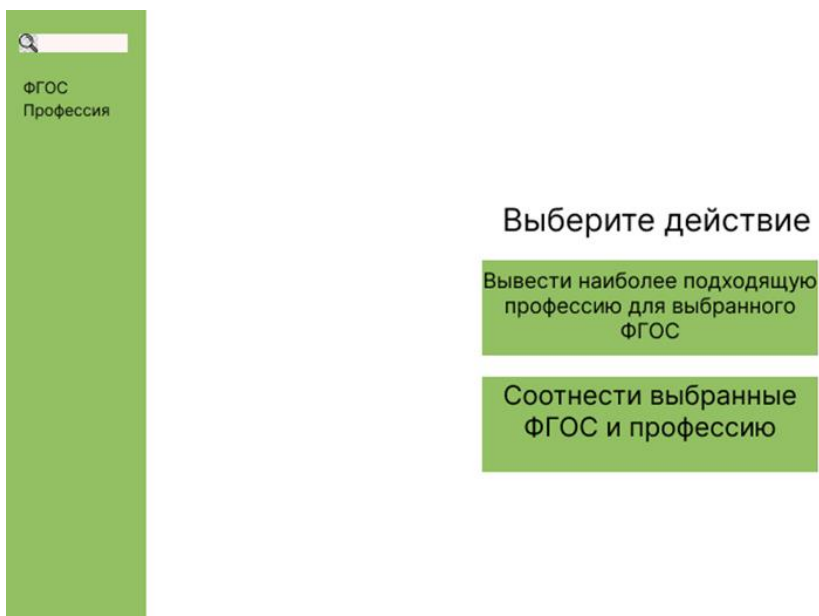


Рис. 3. Стартовый экран системы



Рис. 4. Результат соотнесения выбранных ФГОС и профессии

Имея результаты проектирования, можно приступить к реализации моделей системы на практике: к созданию базы данных ИС и программированию.

### Библиографический список

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации». Доступ из справ.-правовой системы КонсультантПлюс
2. Инструмент для сравнения двух списков. URL:: <https://bezbubna.com/free/compare.php> (дата обращения 06.05.2022)
3. 19 лучших инструментов прототипирования для дизайнеров UX/UI URL:: <https://medium.com/nuances-of-programming/19-лучших-инструментов-прототипирования-для-ux-и-дизайнеров-3d3ee8f56443> (дата обращения 04.05.2022)
4. Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующим

щих профессиональных стандартов, утвержденные Министром образования Российской Федерации Ливановым Д.В 22.01.2015 № ДЛ-01/05вн. Доступ из справ.-правовой системы fgosvo.ru

5. Экспертиза соответствия федеральных государственных стандартов высшего образования профессиональным стандартам, Москва, 2015

6. Приказ Министерства образования и науки РФ от 23 августа 2017 г. N 808 "Об утверждении федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии" (с изменениями и дополнениями) Редакция с изменениями N 1456 от 26.11.2020 Доступ из справ.-правовой системы fgosvo.ru

7. Приказ Министерства труда и социальной защиты РФ от 1 ноября 2016 г. N 598н "Об утверждении профессионального стандарта "Специалист по безопасности компьютерных систем и сетей" Доступ из справ.-правовой системы Гарант.

## **DESIGN AND PROTOTYPE OF THE INFORMATION SYSTEM OF CORRELATION OF COMPETENCES DEVELOPED BY FSES (SUOS) IN THE DIRECTION OF TRAINING AND THE REQUIREMENTS OF PROFESSIONAL STANDARDS**

*Pichkalev Mikhail V., Vasiluk Nadezhda N.*

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

The Federal State Educational Standard (FSES) and professional standards from the RF Labor Code, their similarities and differences, as well as possibilities of correlation are investigated. The article describes the process of designing and prototyping the user interface of the information system for correlation of the competencies formed by FSES and professional standards requirements. The system will allow storing, processing and presenting to the user the data on comparison of competences formed by FSES (FSES) in the field of training and requirements of professional standards, assumes different roles of users in the system (user, administrator). It has the potential to be used by curriculum planners and developers of FSES and professional standards. The paper presents the results of information system design and prototyping the interface of the information system described above.

Keywords: FSES, professional standards, information system, correlation of standards.



## ПРОЕКТИРОВАНИЕ СИСТЕМЫ СБОРА WHOIS ДАННЫХ ДОМЕННЫХ ИМЕН И ИХ АНАЛИЗА

*Пономарев Александр Федорович, Кнутова Наталия Сергеевна*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, airsunday2001@gmail.com

Ежедневно меняется количество размещенных на NS-серверах доменов. Это не является показателем успешности компании, размещающих доменные имена, но таким образом можно анализировать положение дел на рынке. Узнать регистраторов с небольшим количеством зарегистрированных доменов, узнать NS-сервера, где содержится наибольшее количество доменов, узнать распределение доменов по автономным системам – все это может оказать положительное влияние на регистрацию интересного доменного имени. Открытый доступ к данным, зарегистрированным в домене, обеспечивает исследовательская служба Whois, предоставляющая имена, адреса, а также техническую информацию о каждом домене. Так можно узнать не только данные для связи с ответственным лицом доменного имени, но и доступность определенного домена для покупки или регистрации при его освобождении.

Ключевые слова: домен, whois, анализ, информационная система, сбор данных.

Для навигации в интернете используются имеющиеся у каждого компьютера или сервера IP-адреса, благодаря которым можно обращаться к определенным серверам. Чтобы упростить обращение к IP-адресам, которые вызывают проблемы у пользователей с запоминанием несвязного набора чисел, была придумана DNS. Набирая в поисковой строке браузера доменное имя, система посылает запрос DNS-серверу, он обращается к одному из корневых NS-серверов, который в свою очередь передает список NS-серверов искомой зоны. DNS-сервер обращается к одному из NS-серверов зоны и получает список NS-серверов, на которых был делегирован искомый домен. Далее идет обращение к полученным NS-серверам, где храниться информация по домену, например IP-адрес.

Кроме IP-адреса, можно получить и другие данные по доменному имени, например данные регистратора, NS-сервера, даты регистрации и освобождения. За это отвечает WHOIS.

WHOIS представляет из себя сетевой протокол прикладного уровня, базирующийся на 43-ем порту протокола TCP. Эти сервисы активно используются потребителями, государственными учреждениями и правоохранительными органами для связи с правообладателем доменного имени и определения источника продуктов и услуг, продвигаемых в интернете, а также для определения статуса домена.

WHOIS сервис может располагать следующей информацией:

- Доменное имя;
- Данные сервиса DNS;
- Статус домена;

- Данные владельца домена и его контактные данные;
- Дата регистрации и окончание ее срока;
- Дата окончания периода преимущественного продления;
- Данные о регистраторе, поддерживающего сведения о доменном имени;
- Контактные данные регистратора.

Перечень информации может меняться в зависимости от конкретной зоны и ее правил, определяющих набор и содержание полей whois. В базе данных корневой зоны DNS хранится список доменов верхнего уровня, который был создан IANA [1], контролирующей процесс утверждения новых доменов верхнего уровня для ICANN [2]. IANA [1] выделяет шесть групп доменов верхнего уровня:

- ARPA (Advanced Research Project Agency): используются для управления технической сетевой инфраструктуры. Примером является домен .arpa;
- gTLD (Generic top-level domain): к ним относятся, например, такие домены: .com, .net или .org. Они открыты для использования в любых целях;
- grTLD (Generic-restricted top-level domain): в отличие от общих доменов верхнего уровня они имеют ограничения. Например, домен .biz используется для регистрации бизнеса спонсирующих его организаций, например Inc.
- sTLD (Sponsored top-level domain): спонсируемые домены верхнего уровня имеют спонсора, который представляет собой более узкое сообщество, ограничивающее возможность регистрации такого домена. Например, домен .aero спонсируется SITA (Societe Internationale de Telecommunications Aeronautiques), что ограничивает регистрацию только участниками авиатранспортной отрасли;
- ccTLD (Country code top-level domain): такие зоны обычно показывают принадлежность к стране. В некоторых национальных доменах могут регистрироваться только граждане страны, которой эта зона принадлежит, например .ru используется в России, так же как .su, оставшийся от Советского Союза.
- tTLD (Infrastructure top-level domain): используется при тестировании программного обеспечения. Такой домен нельзя зарегистрировать в глобальной системе доменных имен.

Доменное имя – важная составляющая сайта для привлечения и сохранения своих пользователей. Чтобы подобрать интересный домен, а также правильно его зарегистрировать, важно анализировать рынок, регистраторов и сервера, где чаще размещают домены. На данный момент существует очень мало систем предоставляющих данный функционал анализа зон рф\ru\su, и даже они имеют ряд недостатков от необходимости платить за использование возможностей системы, и до устаревшей информации, которая перестала обновляться. Сказанное подчеркивает необходимость создания удобной системы для сбора whois данных доменных имен и их анализа.

Ориентиром для функционала проектируемой системы стал приостановивший свою работу веб сервис firststat.ru [3], предоставляющий рейтинг провайдеров русских национальных зон рф\ru\su, данные автономных систем, а также содержащихся в них доменах. Данный сервис был разработан, основываясь на приостановившем работу некоммерческом проекте lstat.ru [4], где была представлена возможность сравнивать статистику роста количества доменных имен различных периодов с точностью месяц или

год, а также оценивать различные провайдеры между собой. Таким образом, были выделены требования для разрабатываемой системы:

- Возможность получения WHOIS данных из различных доменных зон;
- Удобное для пользователя представление этих данных;
- Реализация мониторинга освобождения домена;
- Удобный поиск доменных имен, по различным критериям;
- Проведения автоматического анализа по распределению доменов в зонах ru/рф/su;

Для реализации всего потенциала системы, лучший выбор – создание веб-приложения с использованием клиент-серверной архитектуры, чтобы реализовать работу с базами данных и передачу файлов посредством HTTP. Для управления информацией воспользуемся архитектурой REST, предназначенной для распределенных систем и являющейся очень простым решением для этих задач. В реализации REST и сервисной части будет уместнее всего выбрать Node JS. Данная программная платформа имеет возможность подключение необходимых библиотек для работы с сервером. Для разработки клиентской части будут использоваться: язык гипертекстовой разметки HTML5 (HyperText Markup Language), язык стилей CSS (Cascading Style Sheets) и язык программирования JavaScript. Чтобы облегчить создание пользовательского интерфейса будет использоваться JavaScript-фреймворк Vue.js. Для работы с базой данных будет использоваться язык программирования Python. Такой выбор подкреплен богатым функционалом данного языка и простой его использования. В качестве СУБД, был выбран MySQL, поддерживающую большое количество типов таблиц. Чтобы создать прототип информационной системы будет использоваться многофункциональный графический редактор Adobe Photoshop 2022. Существует много бесплатных средств, нацеленных на прототипирование интерфейса системы, например Figma. Однако, данные средства, пусть и содержат множество полезных функций для создания прототипа, они не обладают настолько широкими возможностями редактирования пользовательского интерфейса, имея ограничения в создании дизайна различных объектов, в отличии от Photoshop. Также немаловажным фактом в выборе программного средства стал опыт работы в данной программе и то, что была приобретена лицензионная версия данного продукта.

В процессе проведения работы было изучено 14 источников по теме исследования, проанализированы 4 аналога системы сбора whois данных доменов, выявлены и определены требования к разработке информационной системы, проведен анализ инструментов для моделирования информационной системы, определен набор инструментальных средств в соответствии с требованиями, спроектированы диаграммы для системе по методологии UML, разработано 5 эскизов пользовательского интерфейса.

Из вышесказанного можно сделать вывод, что система сбора whois данных доменных имен является незаменимым помощником в анализе рынка для регистрации уникального и интересного доменного имени.

Итогами работы стала соответствующая всем сформулированным требованиям модель системы сбора whois данных доменных имен и их анализа.

### **Библиографический список**

1. IANA: Internet Assigned Numbers Authority [Электронный ресурс] URL: <https://www.getkisi.com/blog/authentication-protocols-overview>;
2. ICANN: Internet Corporation for Assigned Names and Numbers [Электронный ресурс] URL: <https://www.icann.org/>
3. firststat.ru [Электронный ресурс]. URL: <http://firststat.ru>;
4. 1stat.ru [Электронный ресурс]. URL: <http://1stat.ru>.

### **DESIGNING A SYSTEM FOR COLLECTING WHOIS DOMAIN NAME DATA AND ANALYZING THEM**

*Ponomarev Alexander F., Knutova Nataliya S.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [airsunday2001@gmail.com](mailto:airsunday2001@gmail.com)

The number of domains hosted on NS servers changes daily, then it grows, then it falls. And let it not be an indicator of the success of companies hosting domain names, but in this way you can analyze the state of affairs in the market. To find out registrars with a small number of registered domains, to find out NS servers where the largest number of domains is contained, to find out the distribution of domains across autonomous systems – all this can have a positive impact on the registration of an interesting domain name. Open access to the data registered in the domain is provided by the Whois research service, which provides names, addresses, as well as technical information about each domain. This way you can find out not only the data for contacting the responsible person of the domain name, but also the availability of a certain domain for purchase or registration when it is released.

Keywords: domain, whois, analysis, information system, data collection.

## СОЗДАНИЕ БЛОКА КОЭФФИЦИЕНТОВ КОРРЕЛЯЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ВИБРОДИАГНОСТИКИ

*Проскуряков Владислав Сергеевич*

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

Статья посвящена анализу выявления аварий на производстве путем такого метода диагностики, как вибродиагностика, при помощи линейных и ранговых коэффициентов корреляции. Вибродиагностика позволяет выявлять самые разнообразные дефекты оборудования. На основании полученных данных, можно исключать неожиданный выход оборудования из строя, предотвращать аварии, таким образом оптимизировать расходы на ремонт, закупку запчастей и т.п. В настоящее время нет таких устройств, которые могли бы точно указывать на какие показатели нужно обращать внимание, чтобы предотвратить аварию. Коэффициенты корреляции указывают на связь показателей и силу между ними, также они могут показывать направление и тесноту корреляционной связи. Через коэффициенты корреляции возможно выявить аварию на производстве, таким образом узнав от каких показателей зависит данный дефект. На основе выявленного показателя производство может как исправить дефект, так и полностью избавиться от ненужного показателя.

Ключевые слова: вибродиагностика, линейный коэффициент корреляции, ранговый коэффициент корреляции, корреляционная связь.

Аналоговые сигналы изначально преобразуют в цифровые, что является основной функцией современных виброизмерительных приборов [1].

Число проб (выборка) будет содержать числа, равные степени 2:

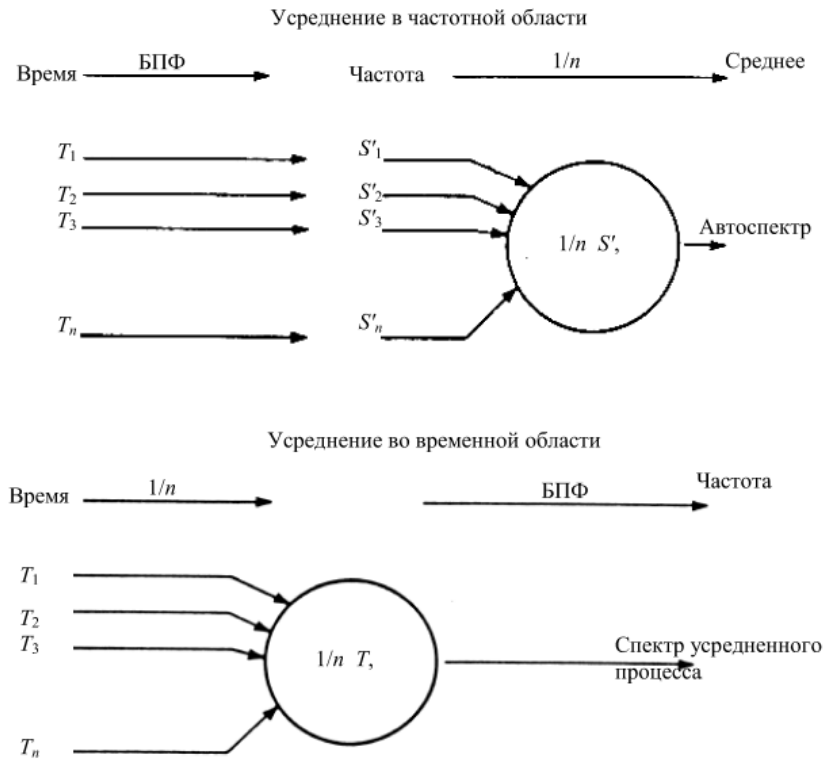
$$N=256,512,1024,2048,4096\dots \quad (1)$$

Под усреднением понимается обработка вибрационного сигнала за принятую длительность измерения для вычисления среднего значения.

Существует несколько видов усреднения :

1. Без усреднения
2. Линейное (нормальное)
3. Экспоненциальное
4. Пиковое

Рис. 1. показывает смысл усреднения в частотной и временной областях по данным.



**Рис. 1. Классификация компьютеров по уровню специализации**

Без усреднения – режим усреднения, когда процесс усреднения не приводится.

Линейное нормальное усреднение – проводится непрерывно или после  $n$  циклов измерений.

В обоих случаях усреднение приводится по формулам:

$$X_{iccg} = \sum_{i=(n-1)}^i X_{i/n} \quad (2)$$

Экспоненциальное усреднение происходит непрерывно время измерения, но усреднению подвергаются последние и реализации по формуле экспоненциального сглаживания. При этом используется рекуррентная зависимость [1, с. 15].

$$x_{et} = (1 - \alpha)x_{e,t-1} + \alpha x_{t-1}, \quad (3)$$

где  $x_{et}$  – экспоненциальная средняя в момент времени  $t$ ,  $0 < \alpha < 1$  – параметр сглаживания.

Пиковое усреднение. Усреднение пиковых значений спектра.

Зачастую используют нормальное усреднение, при этом экспоненциальное усреднение лучше сглаживает выбросы измерений.

Если в приборах, измеряющих диагностические показатели, не предусмотрено усреднение циклов измерений, то необходимо порядка 6 циклов измерений с интервалом не менее 5 с. Усреднение результатов проводится по формулам [2, с. 110]:

$$X = X + 1,05\sigma, \quad (4)$$

где  $X = \frac{\sum X_i}{6}$ ;  $\sigma = \sqrt{\frac{\sum (X_i - X)^2}{5}}$ ;  $X_i$  – результаты измерений.

Динамический диапазон определяет возможность виброизмерительной аппаратуры при измерении амплитуды вибрационного сигнала сохранения линейной связи между входом и выходом.

Динамический диапазон ограничен сверху его максимальным значением входного заряда, снизу – уровнем собственных шумов усилителя заряда.

Динамический диапазон лежит в пределах от 60 до 100дБ, иногда выше [2, с. 110].

Нормальное распределение является одним из наиболее применяемых в математической статистике. Для оценки отклонения эмпирического распределения от нормального используют коэффициенты асимметрии и эксцесс, которые служат для сравнения полигона теоретических частностей вариационного ряда с функцией плотности нормального распределения [3, с. 27].

Выборочный коэффициент эксцесса определяется по формуле:

$$\varepsilon_k^* = \frac{\mu_4^*}{\sigma_e^4} - 3 \quad (5)$$

Эксцесс характеризует крутизну подъема кривой распределения по сравнению с нормальной кривой.

Эксцесс определяет отклонение параметров от нормального распределения, свойственного исправному состоянию механизма.

Преимущества вибродиагностики заключаются в том, что применяемые методы позволяют: [4]

- находить скрытые дефекты;

- не требуют, как правило, сборки-разборки оборудования и значительного времени диагностирования;

- обнаруживать неисправности на этапе их зарождения;

- снижать риск возникновения аварийной ситуации при эксплуатации оборудования.

Однако вибродиагностика не лишена и недостатков, к которым можно отнести: [9]

- особые требования к способу крепления датчика вибрации;

- зависимость параметров вибрации от большого количества факторов и сложность выделения вибрационного сигнала, обусловленного наличием неисправности, что требует глубокого применения методов корреляционного и регрессионного анализа;

- точность диагностирования в большинстве случаев зависит от числа сглаженных параметров, например, числа оценок SPM.

В современном машиноведении значительное внимание уделяется задачам защиты машин, оборудования и аппаратуры от вибрационных внешних воздействий.

Цель виброзащиты в этом случае состоит в уменьшении силы  $F=F(t)$ , которая будет определяться деформацией и скоростью деформации виброизолятора (значением координаты  $y$  и ее производной  $y'$ ) [5]:

$$F = b\dot{y} + ky \quad (6)$$

Корреляционная зависимость – стохастическая зависимость между случайными величинами, при которой наблюдается функциональная зависимость между значениями одной величины и средними значениями другой величины.

Наличие корреляционной зависимости между пересеченными не всегда означает наличие непосредственной связи этих величин друг с другом: наблюдаемая связь часто существует благодаря другим переменным, а изучаемые величины могут быть связаны между собой через латентные (скрытые от исследователя) переменные.

Корреляционная связь в отличие от функциональной, показывает лишь тенденцию изменения одной величины под действием другой, следовательно, на основании корреляции можно утверждать лишь о степени между переменными, но не о существовании причинно-следственной зависимости между ними [6, с. 6].

Корреляционная связи по тесноте связи и количеству признаков.

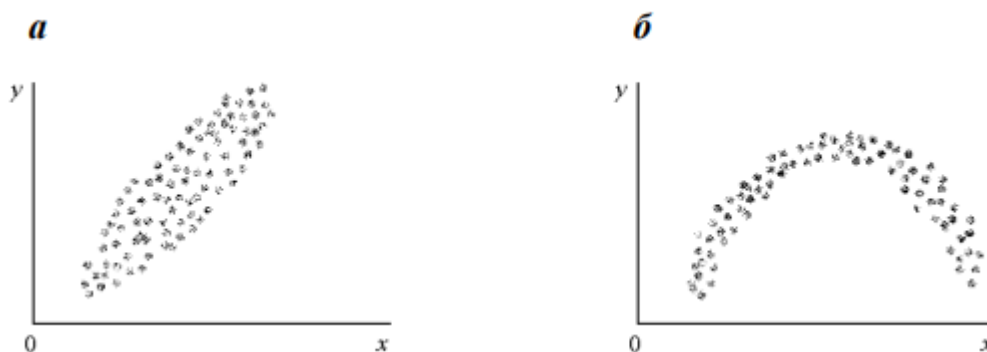
По тесноте корреляционной связи принято выделять:

1. функциональную
2. тесную (сильную)
3. среднюю (умеренную)
4. слабую
5. нулевую (отсутствующую)

По количеству признаков корреляция может быть парной и множественной.

Форма парной корреляции может быть линейной, описываемой линейной функцией регрессии, и нелинейной, описываемой нелинейными функциями.

На рисунке 2 показана линейная (а) и нелинейная (б) корреляционная связи.



**Рис. 2. Линейная (а) и нелинейная (б) корреляционные связи**

Известным примером нелинейной корреляции является первый закон Йеркса – Додсона: по мере увеличения интенсивности мотивации качество деятельности изменяется по колоколообразной кривой: сначала повышается, а затем постепенно снижается (рис. 2б). Другим примером нелинейной связи является закон Хика: скорость переработки информации пропорциональна логарифму от числа альтернатив.

Парная корреляция может быть как положительной, так и отрицательной. При возрастании одного признака в среднем увеличивается другой, в случае же отрицательной корреляции при возрастании одного признак как другой в среднем уменьшается [6 с. 7].

Коэффициент ковариации представляет собой математическое ожидание произведения отклонений величин от их математических ожиданий:

$$\text{cov}(X, Y) = M[(X - M(X))(Y - M(Y))]. \quad (7)$$



Коэффициент корреляции представляет собой безразмерную величину, изменяющуюся в пределах от -1 до 1. Значение коэффициента корреляции выражает лишь долю от максимально возможной ковариации, в чем и состоит его преимущество перед коэффициентом ковариации.

При  $\rho = \pm 1$  корреляционная связь представляет собой линейную функциональную зависимость, при этом все наблюдаемые значения располагаются на прямой (рис. 2 а). При  $\rho = 0$  линейная корреляционная связь отсутствует, корреляционное поле представляет собой круг. Чем ближе значение  $|\rho|$  к единице, тем с большим основанием можно считать, что изучаемые величины находятся в линейной зависимости.

Для независимых случайных величин коэффициент корреляции равен нулю. Однако равенство нулю коэффициента корреляции не всегда означает независимость случайных величин: оно свидетельствует лишь об отсутствии линейной корреляционной зависимости между изучаемыми величинами, но не корреляционной зависимости вообще. Коэффициент корреляции может быть равен нулю в случае нелинейной связи [6, с. 11].

Коэффициент линейной корреляции Пирсона используется для оценки тесноты (силы) связи между двумя переменными в случаях, если:

1. рассматриваемая связь линейная
2. обе переменные измерены в сильных шкалах

Коэффициент линейной корреляции Пирсона:

$$r_{XY} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (8)$$

где  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$  и  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$

Величина выборочного коэффициента линейной корреляции Пирсона, как и генерального, изменяется в пределах от -1 до +1.

Коэффициент ранговой корреляции Спирмена – это непараметрический метод, который используется с целью статистического изучения связи между явлениями. В этом случае определяется фактическая степень параллелизма между двумя количественными рядами изучаемых наблюдений и дается оценка тесноты установленной связи с помощью количественно выраженного коэффициента.

В принципе, коэффициент Спирмена в непараметрической статистике выполняет ту же роль, что и коэффициент корреляции Пирсона ( $r$ ), квадратом которого является широко известный коэффициент детерминации ( $R^2$ ), в параметрической статистике. Однако, для подтверждения статистической значимости на основании коэффициента корреляций Пирсона ( $F$  – критерий, критерий Фишера) данные должны быть распределены нормально, и/или выборка должна быть достаточно большой. Для корреляций Спирмена данные могут быть любыми [7].

Для расчёта коэффициента ранговой корреляции Спирмена необходимо:

1. Сопоставить каждому из признаков их порядковый номер (ранг) по возрастанию или убыванию.
2. Определить разности рангов каждой пары сопоставляемых значений (d).
3. Возвести в квадрат каждую разность и суммировать полученные результаты.
4. Вычислить коэффициент корреляции рангов по формуле:

$$\rho_{XY} = 1 - \frac{6 \sum_{i=1}^n (r_i - s_i)^2}{n(n^2 - 1)}, \quad (9)$$

Коэффициент конкордации Кендалла используется в случае, когда совокупность объектов характеризуется несколькими последовательностями рангов, а исследователю необходимо установить статистическую связь между этими последовательностями. Такие задачи возникают, например, при анализе экспертных оценок: несколько экспертов ранжируют одних и тех же испытуемых по определенному качеству, а психологу для проведения углубленного анализа ситуации и принятия обоснованного решения требуется определить степень согласованности мнений группы экспертов [8].

Коэффициент конкордации Кендалла определяется по формуле:

$$\tau = \frac{2(C - H)}{n(n - 1)}, \quad (10)$$

где С – число совпадений, Н – число инверсий или несовпадений, n – объем выборки, в знаменателе – число сравнений всего.

Этот коэффициент обычно не упоминают в литературе. Однако, если интересует не сумма случаев инверсий, а сумма фактов инверсий, то коэффициент можно вычислить по формуле [9]:

Коэффициент модифицированной конкордации Кендалла определяется по формуле:

$$\tau = \frac{C - H}{n - 1} \quad (11)$$

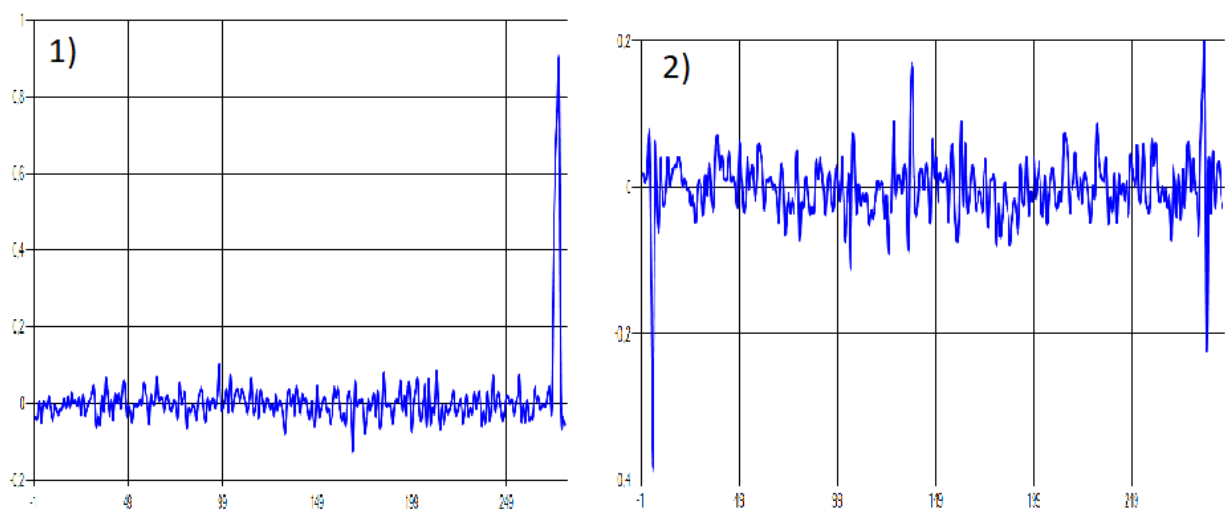
Коэффициент Фехнера, характеризует элементарную степень тесноты связи, которую целесообразно использовать для установления факта наличия связи, когда существует небольшой объем исходной информации. Он основан на сравнении поведения отклонений индивидуальных значений каждого признака x и y от своей средней величины. При этом во внимание принимаются не величины отклонений  $(x_i - \bar{x})$  и  $(y_i - \bar{y})$ , а их знаки («+» или «-»). Определив знаки отклонения от средней величины в каждом ряду, рассматривают все пары знаков и подсчитывают число их совпадений  $\sum C$  и несовпадений  $\sum H$ . Коэффициент Фехнера рассчитывается как отношение разности чисел пар совпадений и несовпадений знаков к их сумме, т.е. к общему числу наблюдаемых единиц [10]:

$$\tau = \frac{C - H}{C + H} \quad (12)$$

Коэффициент Фехнера, как и любой другой показатель тесноты связи, может принимать значения от -1 до +1.

По итогам всех перечисленных методов, единственным методом, по которому можно что-то утверждать является ранговый коэффициент корреляции Спирмена, причем который изменялся динамически.

Примером доказательства служит тот факт, что мы сравнивали нынешний показатель со следующим, соответственно, если эти два показателя будут иметь сильную корреляционную связь, то их нужно будет проверять тщательнее, чем остальные (см. рис. 3).

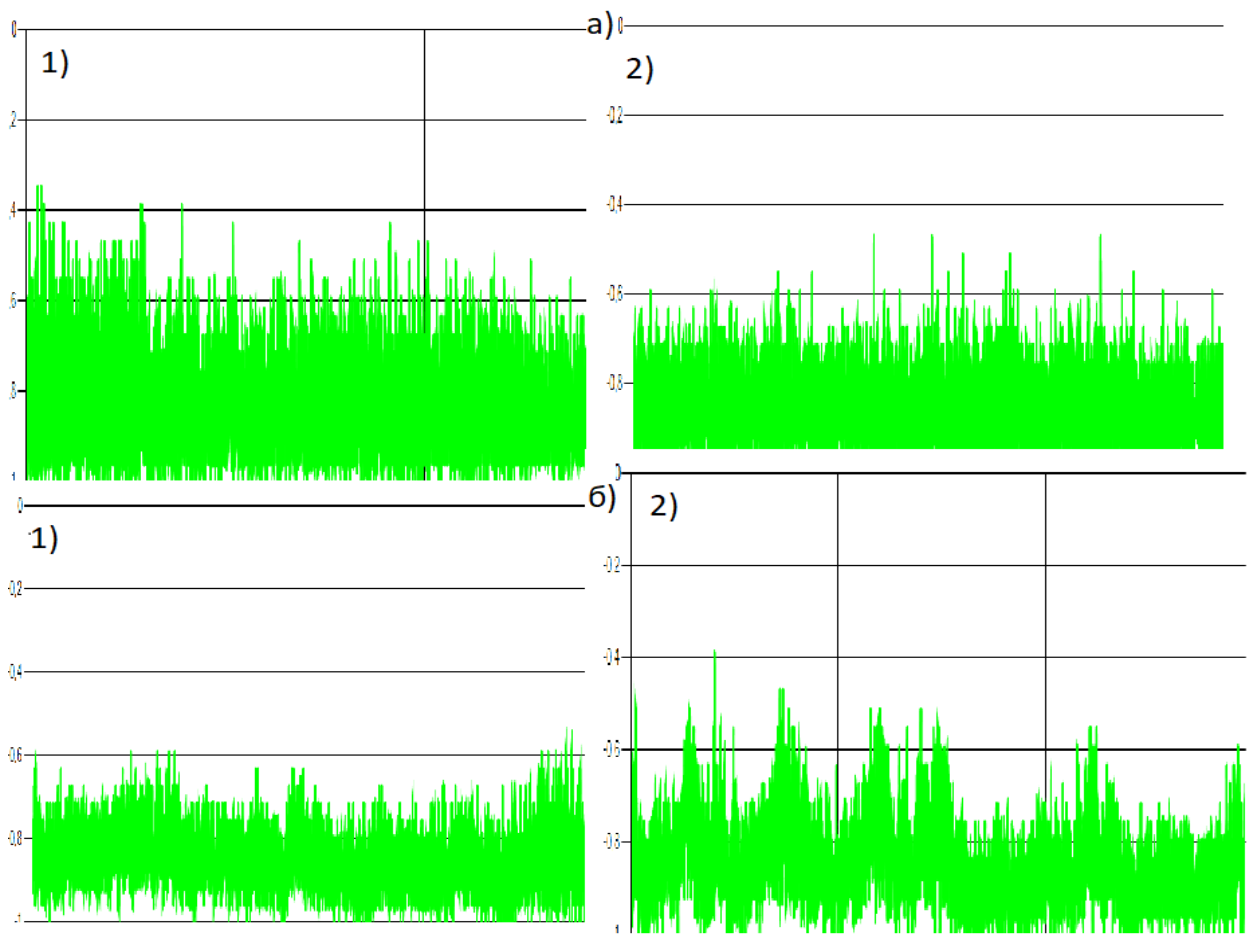


**Рис. 3. Ранговый коэффициент корреляции Спирмена при выборке равной 1000 на двух разных статистических данных где показатель меняется динамически**

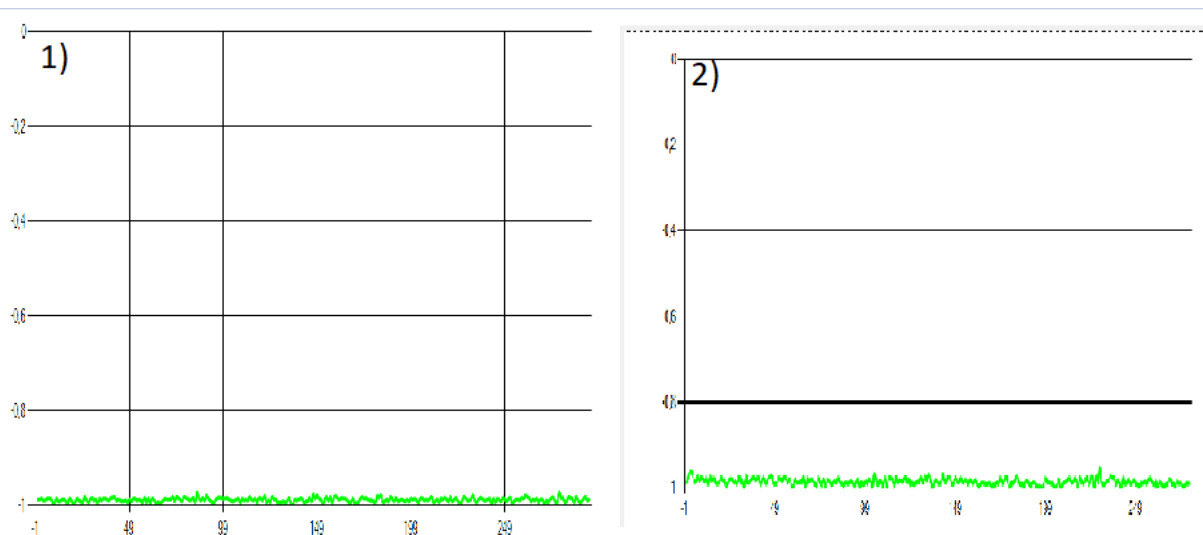
Если один показатель является «эталоном», сказать что-то однозначное нельзя, поэтому данный метод не такой действенный, как предполагалось. Но с другой стороны, если, например, нам нужно будет проверить именно этот показатель со всеми остальными, то, скорее всего, можно будет смотреть показатель, который по отношению к «главному» имеет наивысшую по модулю корреляционную связь, при этом, скорее всего, лучше использовать линейный коэффициент корреляции, т.к. он может показать силу связи.

После обработки и выяснения корреляционных связей каждым методом, описанным в, были выявлены следующие проблемы.

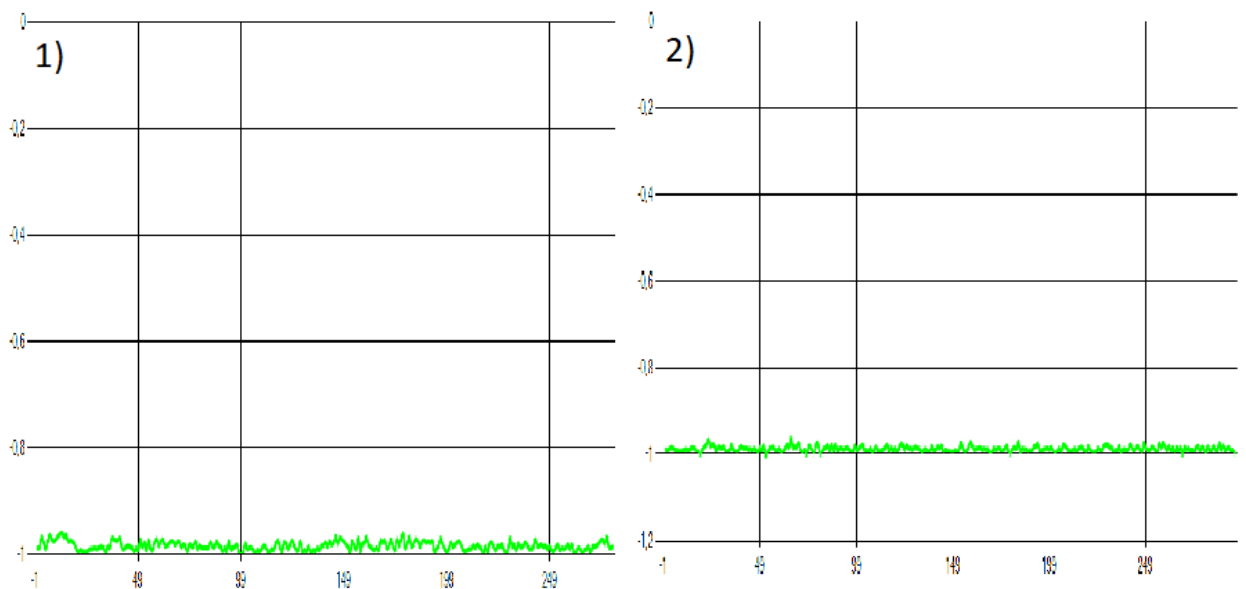
1) Для большой выборки модифицированный коэффициент Кэндалла крайне неэффективен, т.к. количество инверсий слишком велико, поэтому почти все коэффициенты близки к -1 (см рис. 4). Однако при небольшой выборке модифицированный коэффициент Кэндалла может быть полезен. Рисунок 4 показывает модифицированный коэффициент Кэндалла при тех же статистических данных, что и на рисунках 5 и 6, только с выборкой равной 50. Здесь мы уже можем говорить о связи между показателями.



**Рис. 4. Ранговый модифицированный коэффициент корреляции Кэндалла при выборке равной 50 на двух разных статистических данных, а) показатель является «эталоном» б) показатель изменяется динамически**

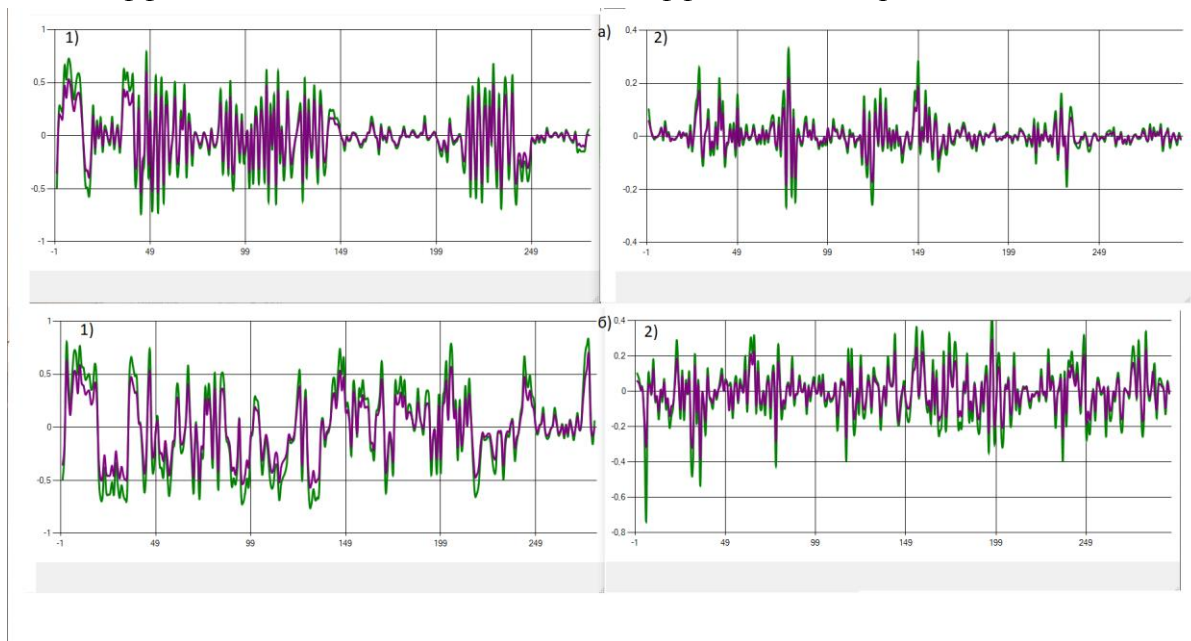


**Рис. 5. Ранговый модифицированный коэффициент корреляции Кэндалла при выборке равной 1000 на двух разных статистических данных, где показатель является «эталоном»**



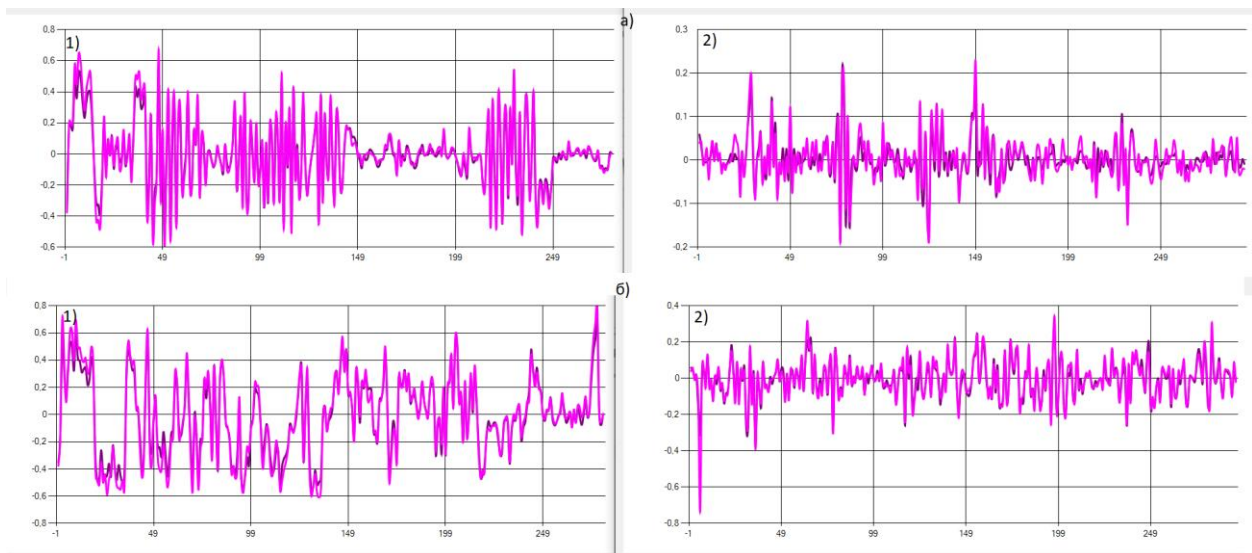
**Рис. 6. Ранговый модифицированный коэффициент корреляции Кэндалла при выборке равной 1000 на двух разных статистических данных, где показатель меняется динамически**

2) При большой выборке коэффициент Кэндалла имеет практически такую же корреляционную связь, что и коэффициент Пирсона, при этом связь переменных для каждого интервала меньше, чем у линейного коэффициента Пирсона. Рисунок 7 показывает разницу между ранговым коэффициентом Кэндалла и линейным коэффициентом Пирсона.



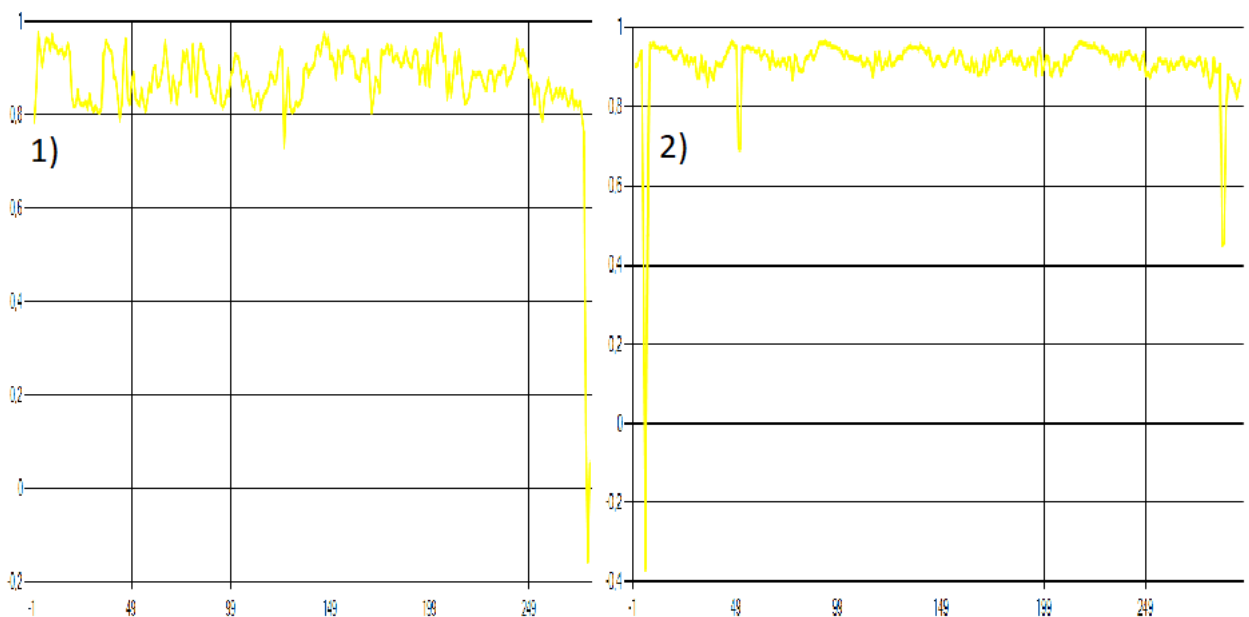
**Рис. 7. Ранговый коэффициент корреляции Кэндалла и линейный коэффициент корреляции Пирсона при выборке равной 1000 на двух разных статистических данных, а) показатель является «эталонном» б) показатель изменяется динамически**

3) Ранговые коэффициенты Кендалла и Фехнера очень близки (почти совпадают) при большой выборке. Рисунок 8 наглядно показывает данный факт.

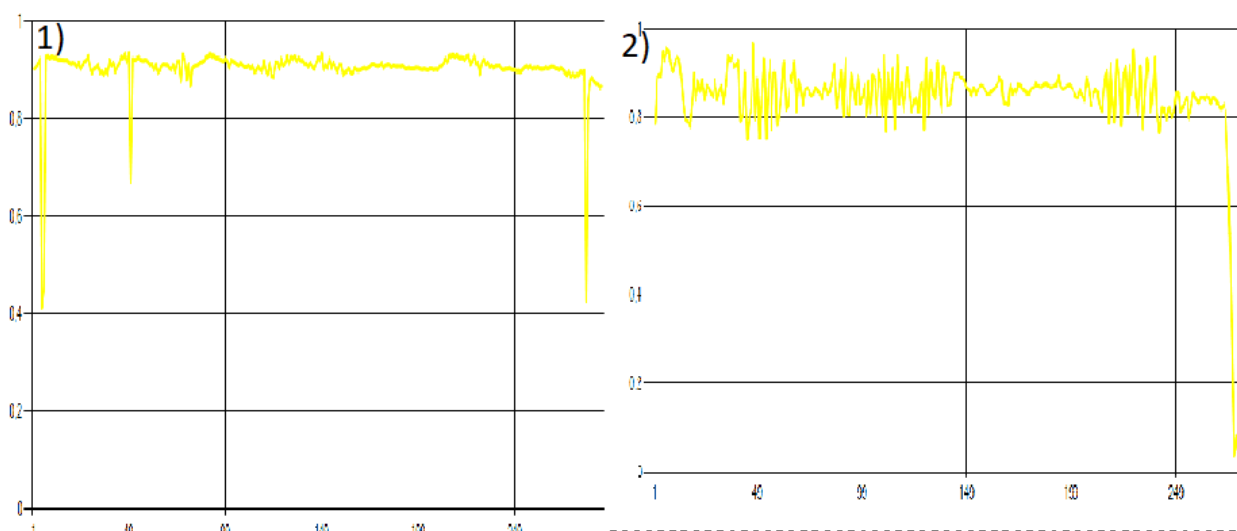


**Рис. 8. Ранговые коэффициенты корреляции Кэндалла и Фехнера при выборке равной 1000 на двух разных статистических данных, а) показатель является «эталоном» б) показатель изменяется динамически**

4) При построении графика корреляционных зависимостей методом линейного коэффициента Пирсона с шумом (см рис. 9 и 10) можно заметить, что на конечном этапе построения график резко убывает, что делает данный метод определения аварии недееспособным, т.к. при дальнейших исследованиях может случиться ситуация, что программа обнаружит аварию, а аварии может и не быть.



**Рис. 9. Линейный коэффициент корреляции Пирсона при шуме равном 0,5 при выборке равной 1000 на двух разных статистических данных, где показатель меняется динамически**



**Рис. 10.** Линейный коэффициент корреляции Пирсона при шуме равном 0,5 при выборке равной 1000 на двух разных статистических данных, где показатель является «эталоном»

### Библиографический список

1. *Кувайскова Ю.Е., Клячкин В.Н.* Статистические методы прогнозирования: учебное пособие. – Ульяновск: УлГТУ, 2019. — 197 с.
2. *Розенберг Г.Ш., Модорский Е.З., Голуб Е.С., Виницкий М.Л., Неелов А.Н., Поросенков Ю.В., Таджибаев А.И.* Вибродиагностика — СПб., ПЭИПК, 2003. — 284с.
3. *Агишева Д.К., Зотова С.А., Матвеева Т.А., Светличная В.Б.* Математическая статистика: учеб.пособие — ВПИ(филиал) ВолГТУ., Волгоград, 2010. — 27 – 29 с.
4. *Мокрушев А.А.* Оценка фактического технического состояния редукторов горных машин методами вибродиагностики – В сборнике: сборник материалов X всероссийской научно-практической конференции молодых ученых с международным участием «Россия молодая». 2018. С. 10306.1 – 10306.3
5. *Сухова К.М. Макарьянц Г.М.* Моделирование системы автоматического управления системы виброзащиты. В книге: международная молодежная научная конференция «XV королевского чтения», посвященная 100-летию со дня рождения Д.И.Козлова. тезисы докладов. 2019. С. 366-367.
6. *Харченко М.А.* Корреляционный анализ. – ВГУ, Воронеж, 2008. – 30 с.
7. *Слуцкий А.А.* Непараметрический критерий статистической значимости результата оценки – коэффициент ранговой корреляции Спирмена. – [Электронный ресурс] URL: <http://tmpro.su/wp-content/uploads/2021/03/Непараметрический-Критерий-Значимости-Результата-Оценки.pdf>. (дата обращения 23.04.2022)
8. *Maurice G., Kendall Rank Correlation Methods* — 1970. — P. 5-20.
9. *Лубенец Ю.В.* О модифицированном коэффициенте конкордации, учитывающем в большей степени согласованность лучших альтернатив — 2017. – 33-39с.
10. *Саадалов Т. Мырзаibraимов Р., Абдуллаева Ж.Д.* Методика расчета коэффициента корреляции Фехнера и Пирсона и их область применения – Бюллетень науки и практики. 2021. Т. 7. №10. С. 270-276.

## **CREATING A BLOCK OF CORRELATION COEFFICIENTS FOR SOLVING THE PROBLEMS OF VIBRO-DIAGNOSTICS**

*Proskuryakov Vladislav S.*

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

**Abstract.** The article is devoted to the analysis of detection of accidents in production by such a diagnostic method as vibrodiagnostics, using linear and rank correlation coefficients. Vibrodiagnostics allows you to identify a wide variety of equipment defects. Based on the data obtained, it is possible to exclude unexpected failure of equipment, prevent accidents, thus optimizing the cost of repairs, purchase of spare parts, etc. Currently, there are no such devices that could indicate exactly what indicators need to be paid attention to in order to prevent an accident. Correlation coefficients indicate the relationship of indicators and the strength between them, they can also show the direction and closeness of the correlation. Through the correlation coefficients, it is possible to identify an accident at work, thus finding out on which indicators this defect depends. Based on the identified indicator, production can both correct the defect and completely get rid of the unnecessary indicator.

**Keywords:** vibration diagnostics, linear correlation coefficient, rank correlation coefficient, correlation.



## **ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ СБОРА И ОБРАБОТКИ ИЗОБРАЖЕНИЙ ДЛЯ МАШИННОГО ОБУЧЕНИЯ ВНЕШНИХ СИСТЕМ**

*Сычев Иван Андреевич, Степанов Владимир Анатольевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Букирева, 15, sychov.ivan2000@yandex.ru

Цель данной работы состоит в проектировании, документировании и разработке информационной системы сбора и обработки изображений для машинного обучения внешних систем. В работе проведен анализ информационных систем сбора и разметки данных, выявлены их достоинства и недостатки. По результатам анализа поставлены задачи на проектирование и составлены требования к разрабатываемой системе. В соответствии с предъявленными требованиями и проведенным сравнительным анализом по ряду критериев и параметров выбраны средства моделирования и реализации информационной системы. Проведен анализ состояния текущих бизнес-процессов и предложен вариант оптимизации. На основании этого спроектирована модель системы. Реализован прототип информационной системы сбора и обработки изображений для машинного обучения внешних систем.

Ключевые слова: датасет, разметка, сбор данных, краудсорсинг, информационная система.

Развитие современного общества неразрывно связано с применением широкого спектра информационных технологий. Данные технологии создают новые возможности: навигатор строит маршрут лучше нас, системы компьютерного зрения точнее, чем человек определяют личность по внешности и т.д.

Машинное обучение являет собой самый простой вариант искусственного интеллекта. Оно предполагает, что можно квалифицировать или предсказывать любой объект, явление или событие с помощью различных методов на основе большого количества тренировочных данных, называемых датасетом.

Датасет – это структурированный набор данных, представленный в удобной форме для исследований и обучения систем.

Довольно часто нужного датасета не существует, либо лицензия запрещает его использование в коммерческих целях. Поэтому, чаще всего датасеты собираются индивидуально, для конкретной задачи, такими задачами могут быть, например классификация изображений или поиск объектов на фото.

Процесс создания датасета чаще всего состоит из двух основных этапов: сбор данных, аннотирование и маркировка данных.

Актуальность данной темы состоит в том, что существующие инструменты в основном узконаправлены, предназначены только для одного из этапов создания датасета, в следствии чего приходится интегрировать их между собой, производить какие-либо операции вручную или писать скрипты.

Цель данной работы состоит в проектировании информационной системы сбора и обработки изображений для машинного обучения внешних систем.

Объектом исследования является система сбора и обработки изображений.

Предметом исследования является автоматизация рутинных действий при сборе и разметке изображений для сокращения трудозатрат при создании датасета.

В датасетах, предназначенных для проектов машинного зрения, используются фото и видео данные. В зависимости от решаемой задачи исходные данные могут быть из различных источников, например поиск в интернете или создание собственного контента.

Существует множество различных типов и методов аннотации изображений. Метод аннотации изображений выбирается в соответствии с требованиями проекта. На основе изученной литературы [1,2] были выделены следующие универсальные методы:

- ограничивающие рамки – рамка вокруг объекта для атрибуции деталей, относящихся к конкретному объекту;
- ориентиры – множество точек, используются для выявления небольших объектов, распознавания лиц, жестов, позы и т.д.;
- полигоны – многоугольники, позволяющие выделить точные границы объектов, для получения наиболее чистых данных;
- линии – применяются для более легкого определения границ, чаще всего используются для выделения дорожной разметки.

Сбор контента, структуризация и разметка довольно трудоемкий процесс, который невозможен без участия человека, но при этом человек совершает множество рутинных действий, которые возможно автоматизировать. В связи с этим появляется потребность в системе позволяющей снизить трудозатраты работников. Собранный контент необходимо хранить в удобной форме с возможностью быстрого доступа и расчёта статистики. Так же нужно контролировать качество разметки и контента.

Автоматизируемые процессы:

- помощь при разметке и классификации – например, использование активного обучения для предварительной разметки;
- сбор статистики – автоматическое вычисление метрик позволяющих оценить эффективность процесса создания датасета;
- распределение подзадач между исполнителями;
- контроль качества.

Разберем примеры некоторых существующих в настоящее время информационных систем, предоставляющих функционал и услуги для создания датасетов изображений.

Яндекс.Толока<sup>21</sup> представляет собой краудсорсинговый проект, созданный с целью сбора, аннотации и анализа большого объема данных, использующихся для машинного обучения и улучшения поисковых алгоритмов.

Amazon SageMaker Ground Truth<sup>22</sup> – это сервис разметки контента с использованием человеческих ресурсов для создания высококачественных наборов данных для моделей ма-

---

<sup>21</sup> <https://toloka.ai/>

<sup>22</sup> <https://aws.amazon.com/ru/sagemaker/data-labeling/>

шинного обучения. Рабочей силой может являться команда заказчика, сторонние поставщики или Amazon Mechanical Turk<sup>23</sup>.

Computer Vision Annotation Tool<sup>24</sup> (CVAT) – это инструмент с открытым исходным кодом для разметки цифровых изображений и видео для задач машинного обучения. Для конечных пользователей CVAT – это web-приложение, работающее в браузере. Он поддерживает разные сценарии работы и может быть использован как для персональной, так и для командной работы. CVAT пригоден для всех основных задач машинного обучения с учителем в области обработки изображений, таких как детектирование объектов, классификация изображений и сегментация изображений.

К достоинствам всех рассмотренных сервисов можно отнести наличие статистики, кроссплатформенность, возможность одновременной работы нескольких пользователей.

Недостатком всех рассмотренных сервисов является невозможность сбора и разметки данных в рамках единого процесса.

Целью создания информационной системы сбора и обработки изображений для машинного обучения внешних систем является:

- сокращение трудозатрат человека;
- повышение качества обработки контента;
- упорядочивание процесса.

Назначением разрабатываемой системы является организация единого процесса создания датасета в соответствии с требованиями проекта.

Разрабатываемая информационная система должна удовлетворять следующим функциональным требованиям:

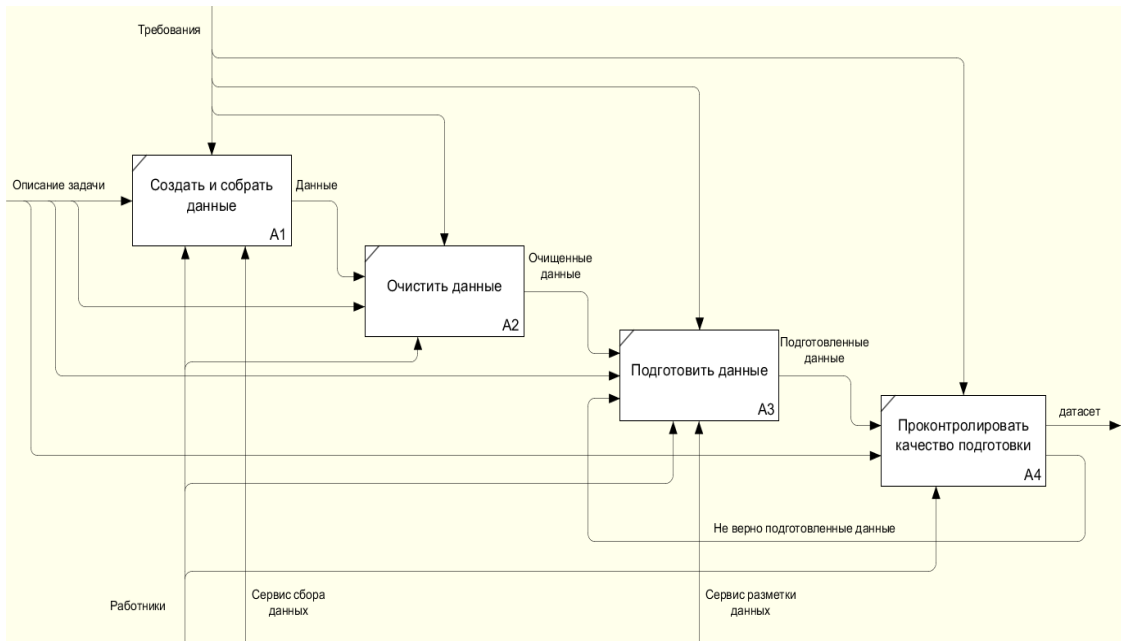
- возможность одновременного создания нескольких независимых датасетов;
- возможность построения единого процесса создания датасета;
- пользователь должен иметь возможность загрузки контента;
- пользователь должен иметь возможность просмотра статистики по собранному контенту;
- в системе должны быть реализованы правила разграничения доступа к данным;
- в системе должны быть реализованы встроенные инструменты контроля качества.

В начале для моделирования процесса сбора и обработки изображений для машинного обучения внешних систем применим методологию IDEF0. Диаграмма первого уровня состоит из 4 функциональных блоков. В функциональном блоке «Создать и собрать данные» работники обрабатывают контент из источников, а также создают собственный в соответствии с требованиями. Далее происходит очистка данных, собранных в предыдущем блоке. После этого очищенные данные передаются в блок подготовки. Под подготовкой данных, в зависимости от требований, подразумевается классификация изображений, детектирование объектов или сегментация изображений. В следующем блоке происходит контроль качества подготовки данных, не верно подготовленные данные отправляются обратно на подготовку. Диаграмма первого уровня представлена на рисунке 1.

---

<sup>23</sup> <https://www.mturk.com/>

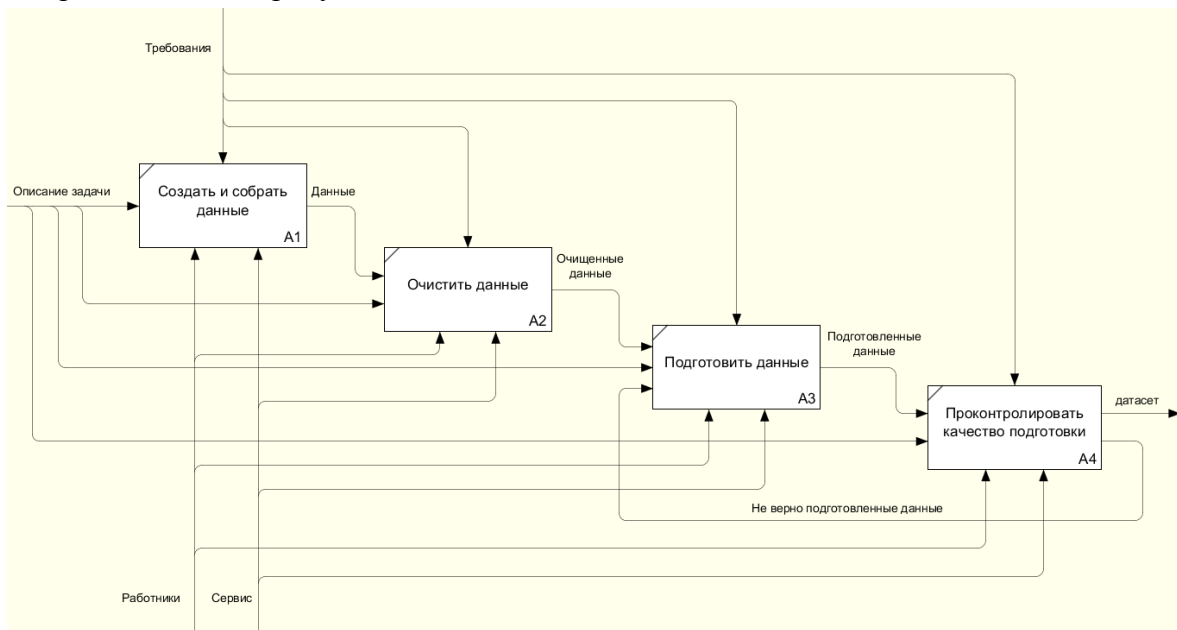
<sup>24</sup> <https://github.com/openvinotoolkit/cvat>



**Рис. 1. Первый уровень диаграммы «как есть»**

На данный момент для каждой части процесса используется свое ПО либо своя независимая задача на сервисе. Таким образом, можно оптимизировать процесс реализовав выполнение всех основных подпроцессов создания датасета в одной системе и автоматизировав передачу данных между ними.

Основное отличие от процесса «как есть» это использование одного сервиса в качестве механизма во всех этапах создания датасета. Декомпозиция основного процесса «как должно быть» представлена на рисунке 2.



**Рис. 2. Первый уровень диаграммы «как должно быть»**

Проведем теоретическую оценку ускорения процесса. Упростим процесс до двух этапов: сбор данных и аннотирование данных.

Введем следующие переменные:

- $N > 0$  – количество изображений, которое должен содержать итоговый датасет.
- $a > 0$  – время затраченное исполнителем на добавление изображения.

- $b \geq 0$  – разница между временем начала сбора данных и временем начала добавления изображения исполнителем.
- $c > 0$  – время затраченное исполнителем на разметку изображения.
- $d \geq 0$  – разница между временем начала процесса разметки данных и временем начала разметки изображения исполнителем.
- $T_1$  – затраченное время процесса «как есть»:
- $T_2$  – затраченное время процесса «как должно быть».

Поскольку изображения добавляются исполнителями не одновременно, в случае отсутствия единого процесса разметки собранных изображений начинается после окончания сбора. Временем, затраченным на сбор данных, можно считать разницу между началом сбора и добавлением последнего изображения. Временем, затраченным на разметку, можно считать разницу между временем начала разметки и разметкой последнего изображения. Соответственно общее затраченное время можно вычислить по формуле (1).

$$T_1 = \max \left( \left\{ \max \left( \{a_j + b_j\}_{j=1}^N \right) + c_i + d_i \right\}_{i=1}^N \right) \quad (1)$$

При наличии единого процесса разметка заданий может начаться сразу после их добавления в датасет. Общее затраченное время можно вычислить по формуле (2).

$$T_2 = \max(\{a_i + b_i + c_i + d_i\}_{i=1}^N) \quad (2)$$

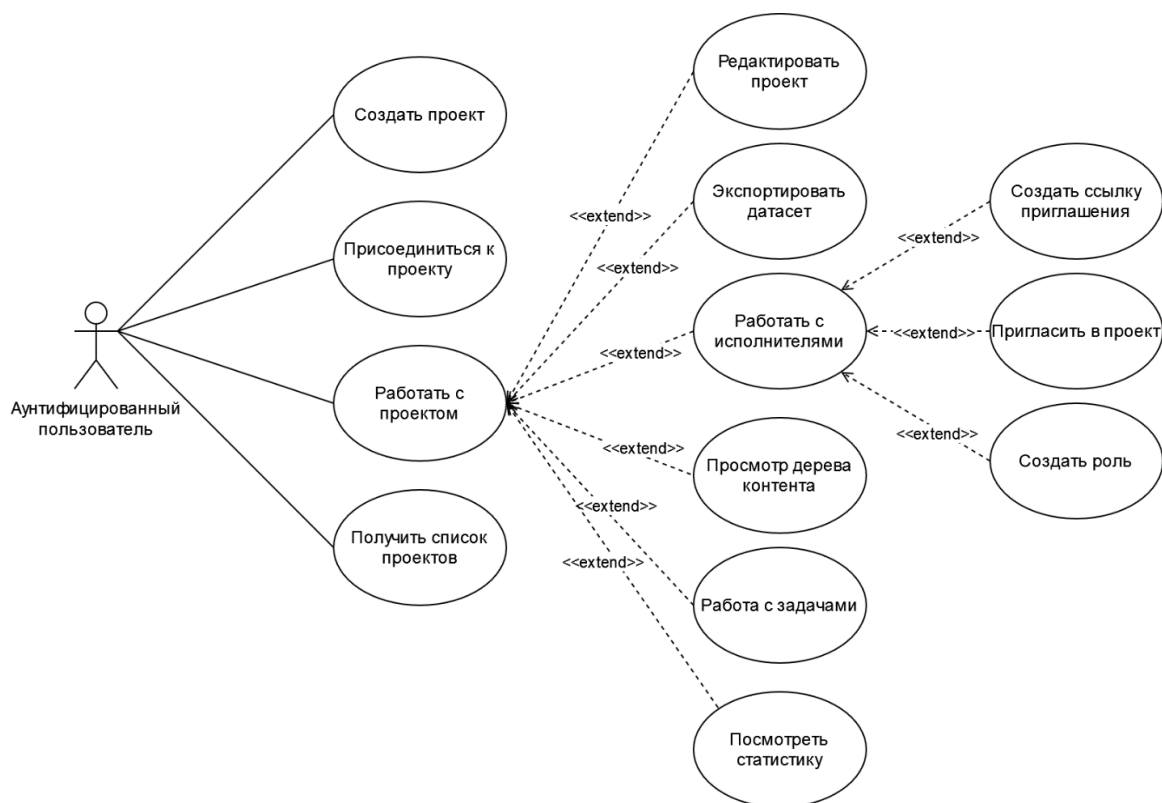
Из формул (1) и (2) следует что:

$$T_1 \geq T_2$$

Соответственно при увеличении числа этапов разница затраченного времени между процессами «как есть» и «как должно быть» будет увеличиваться.

Поскольку система должна контролировать права пользователей, перед началом работы с ней необходимо зарегистрироваться, а после этого войти в систему. В случае если пользователь забыл пароль ему необходима возможность восстановить его. Пароль восстанавливается с использованием кода, отправляемого на почту. Пользователь, вошедший в систему, может выйти из нее, либо изменить пароль.

После прохождения аутентификации пользователю так же становятся доступны основные функции системы. Он может создать проект, получить список доступных ему проектов, присоединиться к проекту по ссылке или работать с одним из проектов. При работе с проектом пользователь имеет возможность отредактировать его, экспортировать доступную на данный момент версию датасета, просмотреть дерево файлов, поработать с задачами, посмотреть статистику. Поскольку в системе требуется разделение прав пользователей, владелец проекта или тот, кто получил права на это может создавать роли, приглашать пользователей в проект, удалять из проекта, изменять права. При создании новой роли определяются права доступа к объектам проекта, таким как задачи, файлы, метки и т.д. Пользователь может иметь одновременно несколько ролей в проекте. Диаграмма основных функций представлена на рисунке 3.



**Рис. 3. Основные функции системы**

В системе может одновременно вестись работа над созданием нескольких независимых датасетов, для разделения этих процессов вводится понятие проект.

Процесс создания датасета в проекте декомпозируется на отдельные задачи, которые связаны между собой.

При добавлении новой задачи пользователь описывает требование к ней в инструкции по выполнению, выбирает тип задачи, добавляет метки, которые будут применяться к контенту исполнителями или автоматически и настраивает список исполнителей выбирая группы пользователей при помощи ролей, которым будет доступна создаваемая задача.

Задачи в системе делятся на подзадачи, это обусловлено тем, что одну и ту же задачу необходимо выполнить несколько раз. Например, в задаче разметки работа с каждым изображением будет считаться отдельной подзадачей. После того как подзадача взята в работу она становится недоступной другим исполнителям. Для исключения ситуаций с повисшими подзадачами в статусе «выполнение», ограничивается время выполнения задачи. В задаче может требоваться проверка результатов выполнения в этом случае подзадача, может быть, либо принята, либо отклонена.

В процессе выполнения задачи для удобства исполнителя и минимизации излишних действий после завершения подзадачи ему сразу выдается следующая. У пользователя есть возможность отказаться от выполнения подзадачи, после этого он так же получит следующую.

Для технической реализации приложения 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 документов в коллекциях базы данных [3].

Так же при разработке будет использоваться технология SPA которая позволяет выполнить большинство требований предъявляемых к системе. Для взаимодействия клиентской и серверной части API строится в соответствии с архитектурным стилем REST. Для хранения файлов было решено использовать объектное хранилище S3, а именно Minio<sup>25</sup>, но система совместима и с другими реализациями объектных хранилищ.

Всю разработку можно разделить на три этапа: разработка серверной части, разработка клиентской части и разработка БД.

Проведен анализ популярных средств проектирования и разработки, сделан выбор средств:

- Draw.io – моделирование диаграмм UML, IDEF0, ER;
- Figma – инструмент прототипирования интерфейса.
- Visual Studio Code – среда разработки;
- JavaScript – язык программирования для разработки;
- MongoDB – база данных.

Карта страниц пользовательского интерфейса представлена на рисунке 4.



Рис. 4. Карта страниц пользовательского интерфейса

При проектировании структуры базы данных применен метод семантического моделирования. Модель базы данных информационной системы, проектируемой в данной работе, представлена в виде ER-диаграммы на рисунке 5.

<sup>25</sup> <https://min.io/>



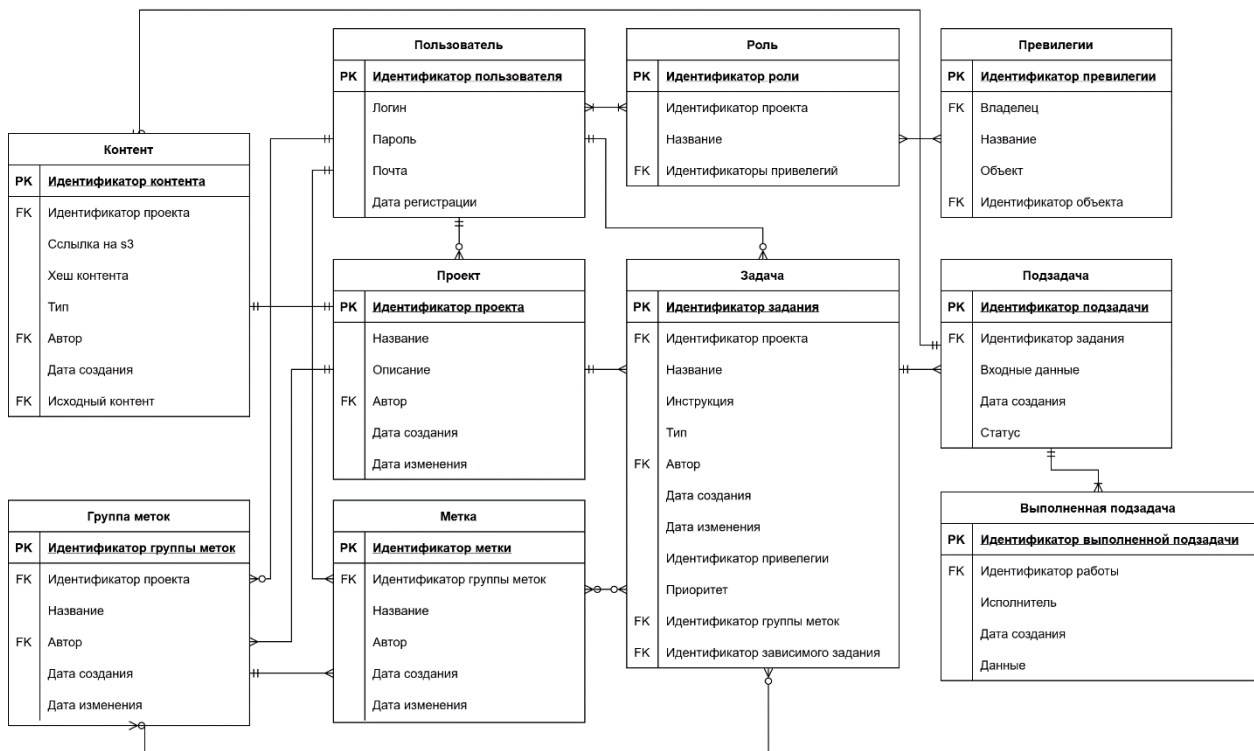


Рис. 5. Логическая модель данных

В рамках данной работы разработан прототип системы, реализовывающий основные функции, такие как:

- аутентификация и авторизация пользователей;
- создание проекта;
- создание задач в проекте;
- объединение задач в проекте в единый процесс;
- функционал выполнения задачи загрузки контента в файловое хранилище;
- функционал выполнения задачи аннотации при помощи ограничивающей рамки;
- файловый менеджер для просмотра создаваемого датасета.

### Библиографический список

1. *Druitsa A. u др.* Practice of efficient data collection via crowdsourcing at large-scale. // arXiv preprint arXiv:1912.04444 – 2019. – С. 1-9.
2. *Элканов, А.Г., Добаев А.З.* Подготовка исходных наборов данных для алгоритмов машинного обучения // Молодые ученые в решении актуальных проблем науки: Материалы VIII Международной научно-практической конференции., Владикавказ, 22 июня 2018 года. – Владикавказ: «Веста», 2018. – С. 90-92. – EDN KQIER.
3. *Беляева И.В.* Архитектура информационных систем: учебное пособие. – Ульяновск УлГТУ, 2019. – 192 с.
4. *Mikowski M., Powell J.* Single Page Web Applications. – Manning Publications, 2014. – 432 с.
5. 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 25.05.2022).



6. *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 с. – Текст: непосредственный.

7. *Симонова О.Н., Лясин Д.Н.* Шаблон проектирования MVC как эффективное средство построения архитектуры программной системы // Современные наукоемкие технологии. – 2014. – №5-2. – С. 96-97.

8. *Brown E.* Web Development with Node and Express. – Gravenstein Highway North, Sebastopo: O'Reilly Media, 2014. – 307 с.

**DESIGN, DOCUMENTATION AND DEVELOPMENT OF AN INFORMATION SYSTEM  
FOR IMAGE ACQUISITION AND PROCESSING FOR MACHINE LEARNING  
OF EXTERNAL SYSTEMS**

*Sychev Ivan A., Stepanov Vladimir A.*

Perm State National Research University, 15 Bukireva str., Perm, 614990, Russia,  
syhov.ivan2000@yandex.ru

The purpose of this work is to design, document and develop an information system for collecting and processing images for machine learning of external systems. The paper analyzes information systems for data collection and markup, identifies their advantages and disadvantages. Based on the results of the analysis, design tasks have been set and requirements for the system being developed have been compiled. In accordance with the requirements presented and the comparative analysis carried out on a number of criteria and parameters, the modeling and implementation tools of the information system were selected. The analysis of the state of current business processes is carried out and an optimization option is proposed. Based on this, a model of the system is designed. A prototype of an information system for collecting and processing images for machine learning of external systems has been implemented.

Keywords: dataset, markup, data collection, crowdsourcing, information system.

## ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЗАЦИИ ПРИНЯТИЯ И ОБРАБОТКИ ЗАКАЗОВ В КАФЕ И РЕСТОРАНАХ

*Тляшева Ильзиля Ильдаровна, Огурецкая Анна Игоревна*

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

Аннотация. В данной статье рассматривается решение по автоматизации ресторанов и кафе в виде электронного меню и его актуальность. Представлено проектирование системы и прототипирование интерфейса системы. В рамках методологии объектно-ориентированного проектирования, была описана диаграмма прецедентов. В рамках методологии моделирования бизнес-процессов была описана BPMN диаграмма. В качестве прототипов интерфейса системы были рассмотрены интерфейсы 3 модулей: для гостя, повара и менеджера.

Ключевые слова: обработка заказов в ресторане и кафе, автоматизация ресторана и кафе.

### **Проектирование информационной системы**

Актуальность темы:

- Система снижает контакты людей между собой во время пандемии;
- Система решает конфликтные ситуации, возникающие из-за человеческого фактора, когда официант неправильно услышал заказ или повара не поняли, что написал официант, что приводит к задержке выполнения заказа;
- Данная система может помочь ресторану или кафе увеличить заработок, обеспечив быструю работу персонала;
- Данная система экологична, так как обеспечена экономия бумаги, связано это с тем, что официанты и повара могут использовать телефоны или планшеты, что уменьшит затраты на покупку блокнотов и чековых принтеров;

На рисунке 1 представлена диаграмма прецедентов, которая изображает основной функционал системы. Из нее мы можем выделить 3 модуля: для гостей, поваров и менеджеров.

Диаграмма прецедентов была выполнена в приложении StarUML. StarUML – это бесплатный программный инструмент для моделирования систем и программного обеспечения [1].

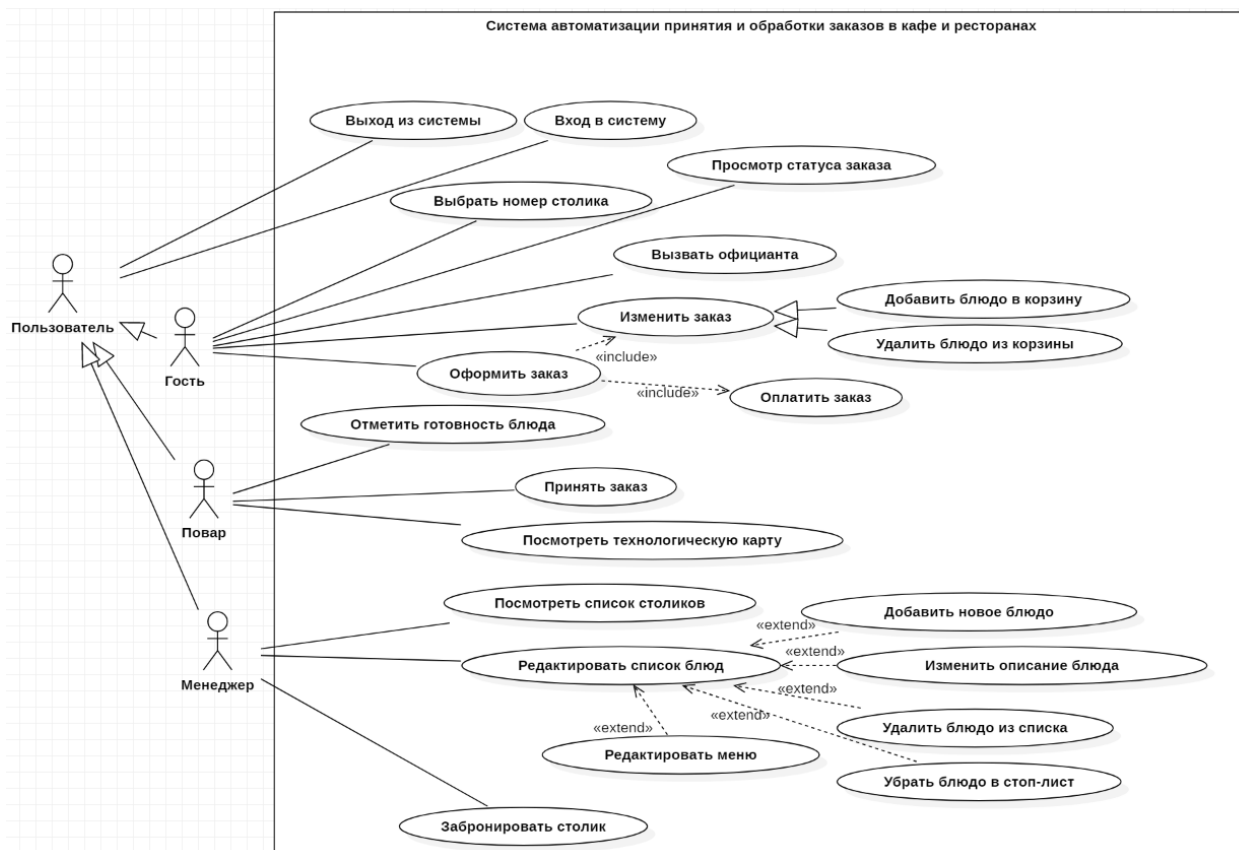


Рис. 1. Диаграмма прецедентов

В рамках методологии моделирования бизнес-процессов рассмотрим BPMN диаграмму, которая была выполнена с помощью приложения Camunda. Camunda – это платформа для моделирования бизнес процессов [2].

На диаграмме, изображенной на рисунке 2, был рассмотрен процесс обработки заказа. В данном процессе участвуют гость, официант и повар.

Процесс будет выполняться до тех пор, пока гость не получит все заказанные им блюда и не оплатит заказ.

Гость делает заказ удобным ему способом и ожидает его получения. Если гость вызывает официанта, то официанту приходит уведомление и он приходит к гостю и выполняет его просьбу. После того как гость или официант сделали заказ, повару приходит уведомление и как только он примет заказ начинается процесс готовки блюда, и он продолжается до тех пор, пока не будут приготовлены все блюда из заказа, по готовности блюда отправляется уведомление официанту, и он относит его гостю. Также в процессе готовки блюд при необходимости повар имеет возможность посмотреть технологическую карту блюда.

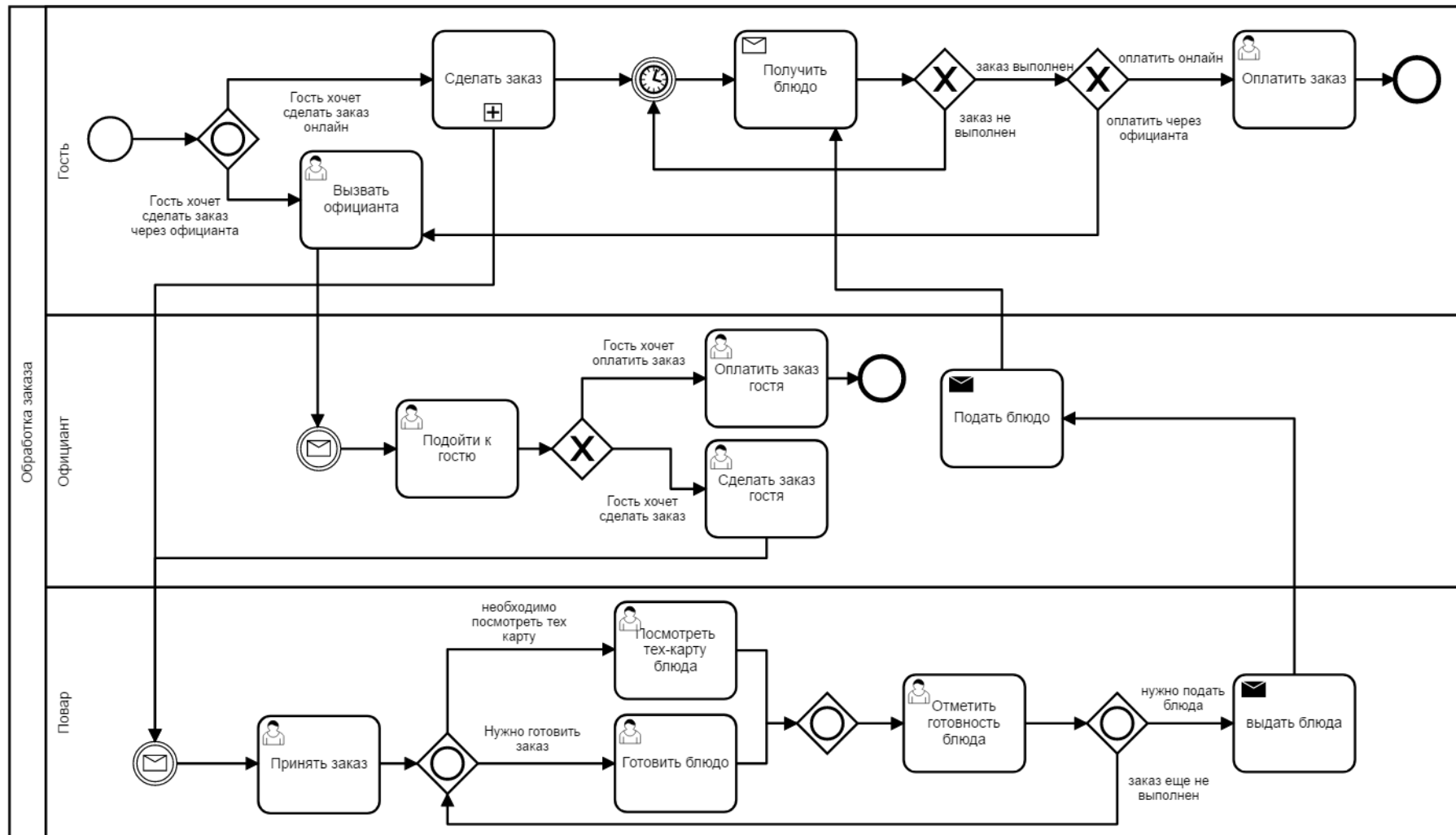


Рис. 2. BPMN диаграмма

### Прототипирование интерфейса системы

Рассмотрим прототипы интерфейсов системы для гостя, менеджера и повара. Прототипы были выполнены с помощью графического онлайн редактора Figma [3].

Опишем работу сайта для пользователя:

После сканирования QR-кода, открывается страница с просьбой ввести номер столика.

После ввода номера столика открывается главная страница, на которой предоставлен выбор категории блюд и представлены блюда из этой категории. При нажатии на блюдо открывается страница с его описанием, и дается возможность добавить его в корзину. При нажатии кнопки «Добавить в корзину» добавляется возможность изменить количество данного блюда в заказе. Интерфейсы перечисленных страниц представлены на рисунке 3.

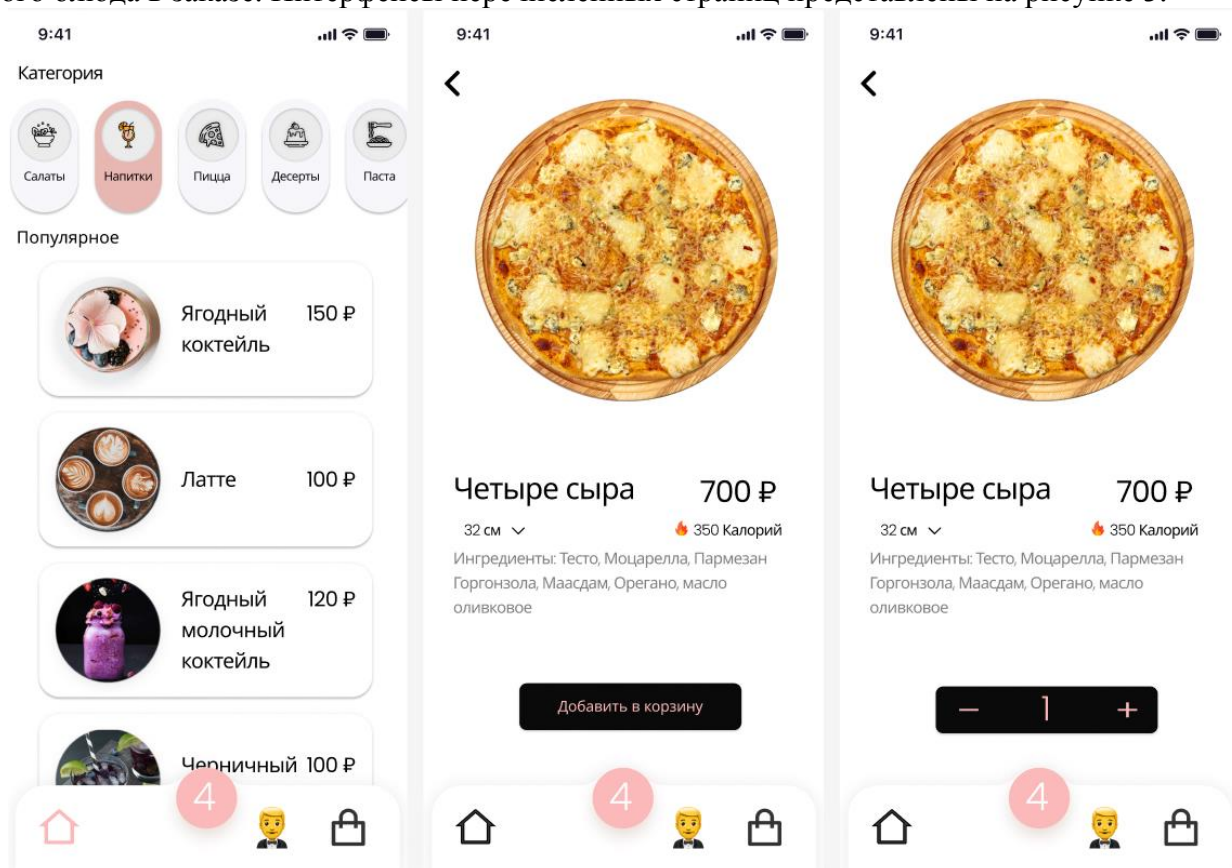


Рис. 3. Главная страница и описание блюда для гостя

При нажатии на кнопку корзины, расположенной в нижнем правом углу страниц, откроется страница с заказом, который можно оформить, нажав на кнопку «Оформить заказ». После оформления заказа открывается страница с его номером, статусом и с примерным временем приготовления заказа. Также на странице представлен список блюд и сумма заказа. Ниже списка блюд есть кнопка «Оплатить заказ», которая предоставляет возможность оплатить заказ в любой момент. Данные страницы изображены на рисунке 4.

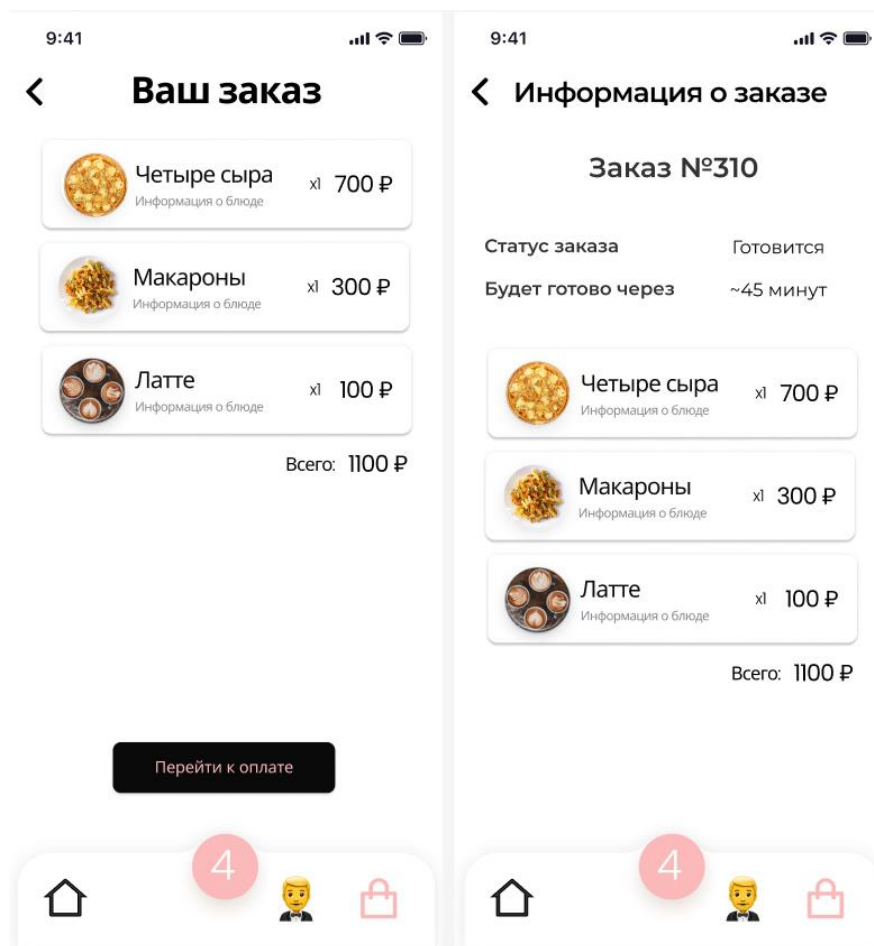


Рис. 4. Корзина и информация о заказе

Рассмотрим прототипы интерфейсов приложения для менеджера изображенные на рисунке 5. При открытии приложения менеджеру необходимо авторизоваться и выбрать действие, которое он хочет сделать.

Менеджер имеет возможность редактировать меню просматривая полный список блюд заведения, в который входит стоп-лист и действующее меню, либо просматривая только меню или только стоп-лист. Добавление и удаление блюда изменю происходит добавлением и удалением галочки напротив названия блюда с правой стороны. Изменить описание блюда можно нажав на кнопку, находящуюся между названием блюда и галочкой. При редактировании описания блюда можно также изменить картинку и технологическую карту. Также через страницу редактирования меню можно добавить новое блюдо нажав на плюс рядом со списками.

Для бронирования столика выбирается номер столика и к нему уже выбираются свободные дата и время, указывается количество человек, вписывается информация о госте и его пожелания.

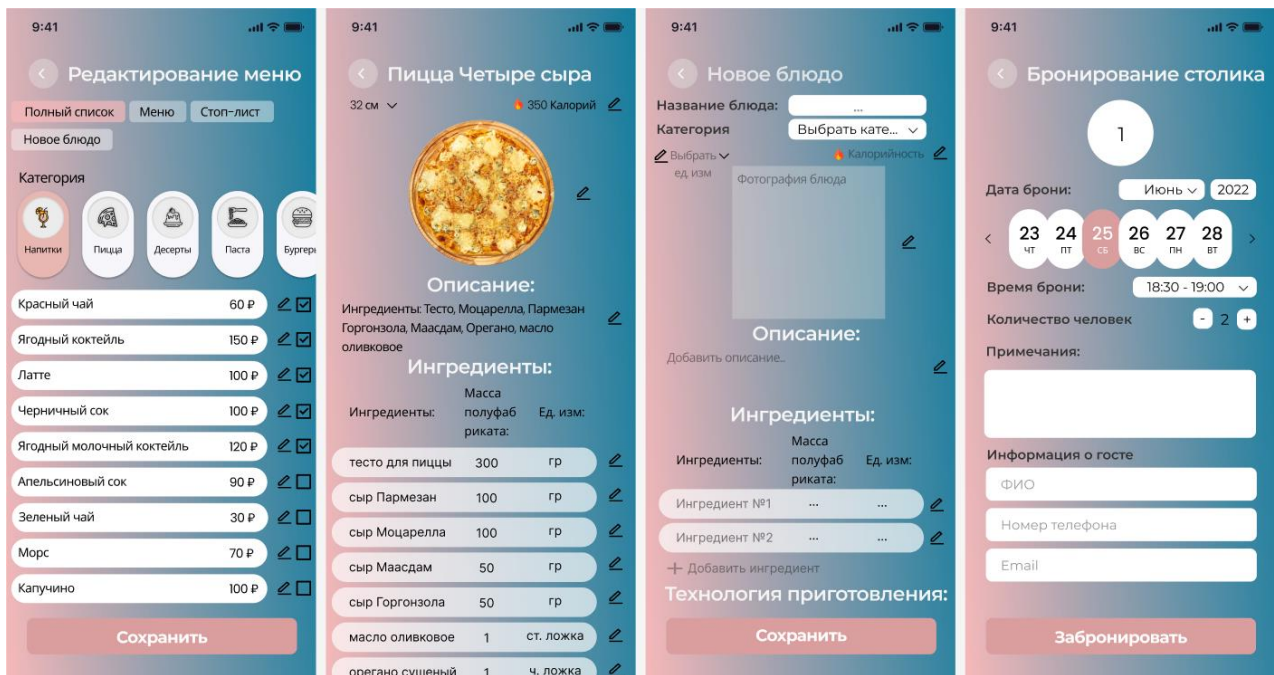


Рис. 5. Редактирование списка блюд и бронирование столика

Следующим разберем прототипы интерфейса приложения для поваров. Главный экран повара, изображенный на рисунке 6, показывает какие заказы сейчас готовятся, ожидают и уже готовы. Также есть возможность посмотреть заказы по стадиям их готовности. В правом нижнем углу указано количество заказов, которые необходимо приготовить. При нажатии на блюдо появляется страница с его технологической картой.



Рис. 6. Главная страница для повара

### Библиографический список

1. StarUML [Электронный ресурс] URL: <https://staruml.io/> (дата обращения: 10.09.2022)
2. Camunda [Электронный ресурс] URL: <https://camunda.com/bpmn/> (дата обращения: 10.09.2022)
3. Figma [Электронный ресурс] URL: <https://www.figma.com/> (дата обращения: 10.09.2022)

## **DESIGNING AND DOCUMENTING OF AN AUTOMATION SYSTEM FOR ACCEPTING AND PROCESSING ORDERS IN CAFES AND RESTAURANTS**

*Tlyasheva Ilzilya I., Oguretskaya Anna I.*

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

**Abstract.** This article discusses the solution for automating restaurants and cafes in the form of an electronic menu and its relevance. The system design and prototyping of the system interface are presented. As part of the methodology of object-oriented design, a precedent diagram was described. As part of the business process modeling methodology, a BPMN diagram was described. As prototypes of the system interface, the interfaces of 3 modules were considered: for the guest, the cook and the manager.

**Keywords:** order processing in a restaurant and cafe, restaurant and cafe automation.



## **ПРОЕКТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ ДЛЯ СОЗДАНИЯ, ХРАНЕНИЯ И ВОССТАНОВЛЕНИЯ РЕЗЕРВНЫХ КОПИЙ БАЗ ДАННЫХ MYSQL**

*Тудвасев Илья Васильевич, Кнутова Наталья Сергеевна*

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

В работе производится анализ предметной области, в результате которого определяется классификация резервного копирования. Также анализируются существующие системы, предназначенные для резервного копирования баз данных MySQL. На основе выделенных достоинств и недостатков существующих систем выдвигаются требования к собственной проектируемой системе. Производится выбор средств проектирования и разработки. В качестве средств проектирования рассматриваются CASE-средства для UML-моделирования. Перед выбором средств разработки сравниваются различные способы реализации. Далее рассматриваются готовые консольные решения для резервного копирования и восстановления баз данных. Сравниваются утилиты от MySQL Server. На следующем шаге производится проектирование системы. Выделяются типы пользователей. На основе ранее определенных требований и выделенных типов пользователей строится диаграмма прецедентов. Для объяснения некоторых прецедентов также строятся диаграммы последовательностей. Для графического представления структуры компонентов системы и взаимодействующих узлов строятся диаграмма развертывания и диаграмма компонентов. Далее осуществляется разработка пользовательского интерфейса с помощью ранее выбранных средств. Перед реализацией серверной части системы производится подготовка программной составляющей. В процессе разработки серверной части сравниваются и выбираются функции. Описываются используемые утилиты резервного копирования, SQL-запросы и решения для удаленного хранения копий.

Ключевые слова: проектирование и документирование системы, резервное копирование баз данных, хранение резервных копий баз данных, восстановление баз данных, облачное хранилище, MySQL, серверный скрипт, UML-моделирование, разработка интерфейса, разработка серверной части.

### **Введение**

На данный момент многими практикуется работа с базами данных. Повреждение данных на носителе или же сбой на сервере может привести к потере данных. В качестве решения этой проблемы были придуманы различные способы создания резервных копий. Но также существует проблема хранения резервных копий. Хранить резервные копии на том же сервере или носителе является небезопасным решением, так как сбой может привести как к потере самой базы данных, так и к потере ее резервных копий. Для решения этих проблем производится проектирование, документирование и разработка собственной системы резервного копирования.

### **Анализ предметной области**

В результате анализа резервного копирования баз данных MySQL выделяется классификация процесса. Резервное копирование классифицируется:

- по типу создаваемой копии;
- по методу создания копии;
- по полноте сохраняемой информации.

По типу создаваемой копии выделяются физическое и логическое резервное копирование. По методу создания копии – холодное и горячее. По полноте сохраняемой информации – полное, инкрементное и дифференциальное.

На следующем этапе анализируются существующие системы, предназначенные для резервного копирования баз данных MySQL. Рассматриваются следующие системы: phpMyAdmin [2], Sypex Dumper [3], Adminer [4], Handy Backup [5], HeidiSQL [6]. В результате анализа выделяются основные достоинства и недостатки существующих систем. На основе выделенных достоинств и недостатков выдвигаются требования к собственной проектируемой системе:

- интуитивно понятный интерфейс;
- поддержка Windows и Linux ОС;
- экономия памяти;
- возможность переключения между языками (русский/английский);
- возможность просмотра журнальных записей;
- предоставление функций создания резервной копии, восстановления БД из резервной копии;
- возможность выбора объектов для резервного копирования и восстановления;
- возможность выбора алгоритма сжатия;
- возможность делать резервные копии по расписанию автоматически;
- возможность прекратить процесс резервного копирования или восстановления;
- возможность добавления облачного сервера для хранения копий;
- использование технологии, позволяющей делать быстрые бэкапы;
- возможность редактирования программной составляющей.

### **Выбор средств проектирования и разработки**

После анализа предметной области производится выбор средств проектирования и разработки. В качестве средств проектирования рассматриваются CASE-средства для UML-моделирования [7]: StarUML, Diagrams.net, Umbrello, Microsoft Visio. Средства сравниваются по удобству интерфейса, функционалу, поддержке русского языка и способу распространения. В результате сравнения выбирается CASE-средство Diagrams.net. Перед выбором средств разработки сравниваются различные способы реализации, среди которых выбирается реализация в виде серверного скрипта. Для серверной части в качестве языка программирования выбирается PHP. Для клиентской части – язык гипертекстовой разметки HTML5, язык стилей CSS3, язык JavaScript и WEB-конструктор Nicepage.

Далее рассматриваются готовые консольные решения для резервного копирования и восстановления баз данных. Сравниваются решения от MySQL Server [8]: mysqldump, mysqlpump, mysqldumper, MySQL Shell Dump/Load. В результате сравнения выбирается решение MySQL Shell Dump/Load, так как его утилиты имеют наиболее высокую скорость работы, а также имеется поддержка функций, подходящих под ранее установленные требования к собственной системе.

После определения используемых средств разработки производится анализ сред разработки. В качестве сред разработки рассматриваются: Visual Studio, IntelliJ IDEA, PhpStorm. В

результате анализа выбирается PhpStorm, так как данная IDE может использоваться на бесплатной основе, включает в себя функции оптимизации процесса написания кода, а также поддерживается разработка на языках, выбранных ранее.

### Проектирование информационной системы

Проектирование информационной системы состоит из следующих этапов:

- моделирование поведения ИС;
- моделирование структуры ИС.

Перед построением диаграммы прецедентов определяются действующие лица. Для действующих лиц производится распределение прецедентов. Также выделяются вложенные прецеденты. В результате строится следующая диаграмма:

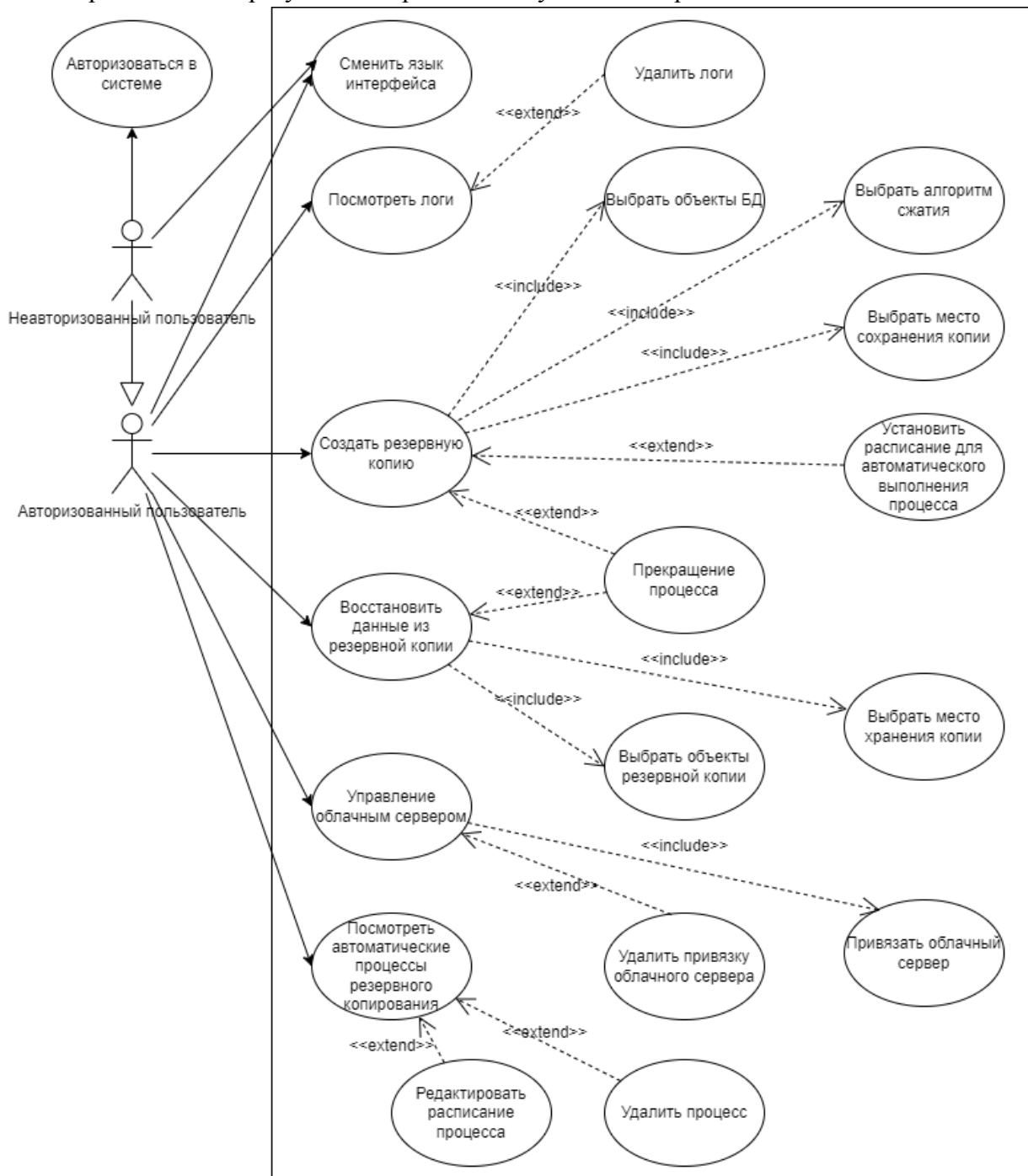


Рис. 11. Диаграмма прецедентов

Для объяснения прецедентов «Создать резервную копию», «Установить расписание для автоматического выполнения процесса», «Восстановить данные из резервной копии», «Посмотреть логи», «Удалить логи» и «Привязать облачный сервер» строятся диаграммы последовательностей. Для описания процессов создания резервной копии и восстановления БД из резервной копии строятся диаграммы взаимодействия. Для графического представления структуры компонентов системы и взаимодействующих узлов строятся диаграмма развертывания и диаграмма компонентов.

### **Разработка информационной системы**

Разработка информационной системы состоит из следующих этапов:

- разработка пользовательского интерфейса;
- подготовка программной составляющей для разработки ИС;
- разработка серверной части ИС.

Для разработки пользовательского интерфейса были задействованы ранее выбранные средства: WEB-конструктор Nicepage, языки HTML5, CSS3 и JavaScript. С помощью данных средств были построены:

- страница авторизации;
- страница выбора объектов для резервного копирования;
- всплывающее окно с информацией о таблице;
- всплывающее окно с настройкой расписания резервного копирования;
- страница резервного копирования в момент создания резервной копии;
- страница выбора места сохранения файла;
- страница просмотра и редактирования расписания резервного копирования;
- страница выбора объектов для восстановления БД;
- страница восстановления БД в момент процесса восстановления;
- страница журнальных записей;
- страница управления облачными серверами;
- всплывающее окно с вариантами облачных сервисов.

Перед переходом к процессу разработки серверной части ИС была произведена подготовка программной составляющей. В качестве локального сервера был установлен и настроен программный комплекс Open Server Panel. При настройке компонентов в качестве HTTP-сервера был выбран Apache 2.4, в качестве используемой версии PHP была выбрана версия 8.0, для СУБД – MySQL 8.0.

При разработке серверной части производилось сравнение функций исполнения внешних программ, в результате которого было принято решение использовать функцию `proc_open()`. Помимо этого, были рассмотрены опции, указываемые при открытии MySQL Shell, а также утилиты [9] для создания резервной копии и восстановления БД. В качестве опций были выделены `--host`, `--user` и `--password`, которые указываются для прохождения процесса авторизации при открытии MySQL Shell. В качестве утилит же были выделены `util.dumpInstance` – для создания полной резервной копии, `util.dumpTables` – для создания резервной копии таблиц и триггеров из БД при выборе части объектов, `util.dumpTables` – для создания резервной копии при выборе части объектов, среди которых есть процедуры/события, а также `util.loadDump` – для восстановления БД из резервной копии. Также были

определены SQL-запросы для извлечения необходимой информации из БД, используемые функции и технологии при проведении автоматического резервного копирования и ручного прекращения процесса восстановления и резервного копирования. Для реализации привязки и использования облачного хранилища было принято решение использовать готовые решения, представленные в документации Drive API [10] для Google Drive, на странице GitHub одного из разработчиков компании «Яндекс» [11] для Яндекс.Диск, в документации Dropbox-PHP API [12] для сервиса Dropbox. Установка библиотек производилась при помощи менеджера зависимостей Composer.

### **Заключение**

В результате выполнения работы выполнены следующие задачи:

- анализ предметной области;
- анализ существующих аналогов ИС резервного копирования БД MySQL;
- определение требований к проектируемой ИС;
- выбор средств для проектирования и разработки ИС;
- проектирование и документирование ИС резервного копирования БД MySQL;
- разработка прототипа пользовательского интерфейса;
- разработка прототипа ИС.

Разработанный прототип информационной системы позволяет за короткое время создавать резервные копии, восстанавливать БД MySQL из резервных копий, а также сохранять копии на облачном, серверном или локальном хранилище. Также система позволяет устанавливать и редактировать расписание для резервного копирования, изменять язык интерфейса, просматривать и удалять журнальные записи.

### **Библиографический список**

1. Глава 8. Резервирование и восстановление // RLDP – русская версия Linux Documentation Project, включающая в себя руководства по СУБД MySQL [Электронный ресурс]. URL: <http://www.rldp.ru/mysql/mysql80/backup.htm> (Дата обращения: 10.12.21)
2. phpMyAdmin Приложение для администрирования СУБД MySQL [Электронный ресурс]. URL: <https://www.phpmyadmin.net/> (Дата обращения: 15.12.21)
3. Syrex Dumper Программный продукт для создания и восстановления резервных копий баз данных MySQL [Электронный ресурс]. URL: <https://sypex.net/> (Дата обращения: 15.12.21)
4. Adminer – database management in a single PHP file [Электронный ресурс]. URL: <https://www.adminer.org/> (Дата обращения: 15.12.21)
5. Handy Backup Программа для резервного копирования и восстановления данных ПК [Электронный ресурс]. URL: <https://www.handybackup.ru/> (Дата обращения: 15.12.21)
6. HeidiSQL Free and open-source administration tool for MySQL and its forks [Электронный ресурс]. URL: <https://www.heidisql.com/> (Дата обращения: 15.12.21)
7. Введение в UML: Информация // Национальный Открытый Университет «ИНТУИТ» [Электронный ресурс]. URL: <https://www.intuit.ru/studies/courses/1007/229/info> (Дата обращения: 22.05.22)

8. MySQL Shell Dump & Load // MySQL Server Blog – news from MySQL Server Team [Электронный ресурс]. URL: <https://mysqlservertteam.com/mysql-shell-dump-load-part-1-demo/> (Дата обращения: 25.05.22)
9. MySQL Shell API: util // MySQL Developer Zone [Электронный ресурс]. URL: [https://dev.mysql.com/doc/dev/mysqlsh-api-javascript/8.0/group\\_\\_util.html](https://dev.mysql.com/doc/dev/mysqlsh-api-javascript/8.0/group__util.html) (Дата обращения: 25.05.22)
10. Drive API PHP Quickstart // Google Drive for Developers [Электронный ресурс]. URL: <https://developers.google.com/drive/api/quickstart/php> (Дата обращения: 25.05.22)
11. GitHub – jack-theripper/yandex // GitHub – веб-сервис для хостинга IT-проектов и совместной разработки [Электронный ресурс]. URL: <https://github.com/jack-theripper/yandex> (Дата обращения: 25.05.22)
12. Dropbox-PHP API Documentation // Dropbox client library for PHP [Электронный ресурс]. URL: <https://dropbox-php.github.io/dropbox-php/docs> (Дата обращения: 25.05.22)

## **DESIGN AND DOCUMENTING A SYSTEM FOR CREATING, STORING AND RESTORING BACKUPS OF MYSQL DATABASES**

*Tudvasev Ilya V., Knutova Natalia S.*

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

The paper analyzes the subject area, as a result of which the classification of backup is determined. The existing systems designed for backup of MySQL databases are also analyzed. Based on the identified advantages and disadvantages of existing systems, the requirements for the own designed system are put forward. The choice of design and development tools is made. CASE-tools for UML modeling are considered as design tools. Before choosing the development tools, different implementation methods are compared. Next, we consider ready-made console solutions for database backup and recovery. Utilities from MySQL Server are compared. The next step is the design of the system. The types of users are highlighted. A use case diagram is constructed based on previously defined requirements and selected user types. Sequence diagrams are also constructed to explain some use cases. Sequence diagrams are constructed to describe the processes of database backup and recovery. For a graphical representation of the structure of the system components and interacting nodes, a deployment diagram and a component diagram are built. Next, the user interface is developed using the previously selected tools. Before the implementation of the server part of the system, the software component is prepared. During the development of the back-end, features are compared and selected. Describes the backup utilities used, SQL queries, and remote backup solutions.

**Keywords:** system design and documentation, database backup, database backup storage, database recovery, cloud storage, MySQL, server script, UML modeling, interface development, server side development.

## ПРОЕКТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ «ПОДАРИЮ-ЗАБЕРУ»

*Угринов Валерий Александрович*

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

*Кушев Вадим Олегович*

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

В статье поэтапно разобран процесс разработки информационной системы, благодаря которой пользователи смогут безвозмездно обмениваться товарами в интернете. Определён предмет и объект исследования, сформулирована актуальность разработки данной информационной системы, а также поставлены задачи, которые необходимо решить в ходе разработки. Проведён анализ существующих интернет-сервисов размещения объявлений, определены задачи, которые должны обеспечиваться разрабатываемой информационной системой. Выбраны средства проектирования и разработки. Созданы различные диаграммы в нотации UML для описания функциональной части информационной системы. Также создана концептуальная диаграмма, описывающая то, как будут храниться данные. После чего был разработан пользовательский интерфейс, включающий в себя страницы и компоненты веб-сайта. Также описаны особенности реализации и тестирования информационной системы.

Ключевые слова: безвозмездный обмен, доска-объявлений, веб-сайт.

Жизнь людей стала намного проще, ведь стало легче отдавать и покупать вещи. Появилось множество компаний, предлагающих огромное количество услуг в данной отрасли. Многие интернет сервисы ставят огромную долю внимания на платные сделки, а хороших сервисов для проведения безвозмездных, бесплатных сделок очень мало. Большинство таких сделок проходит в социальных сетях ВКонтакте и Одноклассники, которые имеют ограниченный функционал для этого.

В понимании того, как устроен феномен сообществ онлайн дарообмена крайне полезна статья Е.В. Полухина и А.В. Стрельникова "От избыточного потребления – к устойчивому: феномен онлайн-дарообмена", в которой был проведён анализ безвозмездного обмена вещами в интернете, исследуются причины, структура последствия этого явления [1].

В связи с актуальностью темы безвозмездного обмена вещами появилась потребность в разработке информационной системе по обмену вещами на бесплатной основе.

Данный сервис будет иметь популярность, так как не будет перенасыщен отдельными платными объявлениями и будет иметь функционал, специально предназначенный для безвозмездного обмена вещами.

Таким образом, цель – проектирование, документирование и разработка приложения “Подарю-заберу”.

Задачи, которые необходимо решить в ходе проектирования и разработки информационной системы:

- 1) обзор и анализ существующих информационных систем, выполняющих схожие функции;
- 2) обзор и выбор средств проектирования и реализации информационной системы;
- 3) проектирование информационной системы;
- 4) реализация информационной системы.

Объект исследования – безвозмездный обмен вещами.

Предмет исследования – автоматизация предоставления информации для безвозмездного обмена вещами.

Разобраны такие интернет-сервисы для размещения объявлений как:

- avito;
- youla;
- irr;
- kupipro dai;
- gde;
- farpost;
- barahla;
- otdarom.

В ходе анализа были составлены критерии сравнения, которые позволили полноценно сравнить данные информационные системы и выделить наиболее важный для данной предметной области функционал.

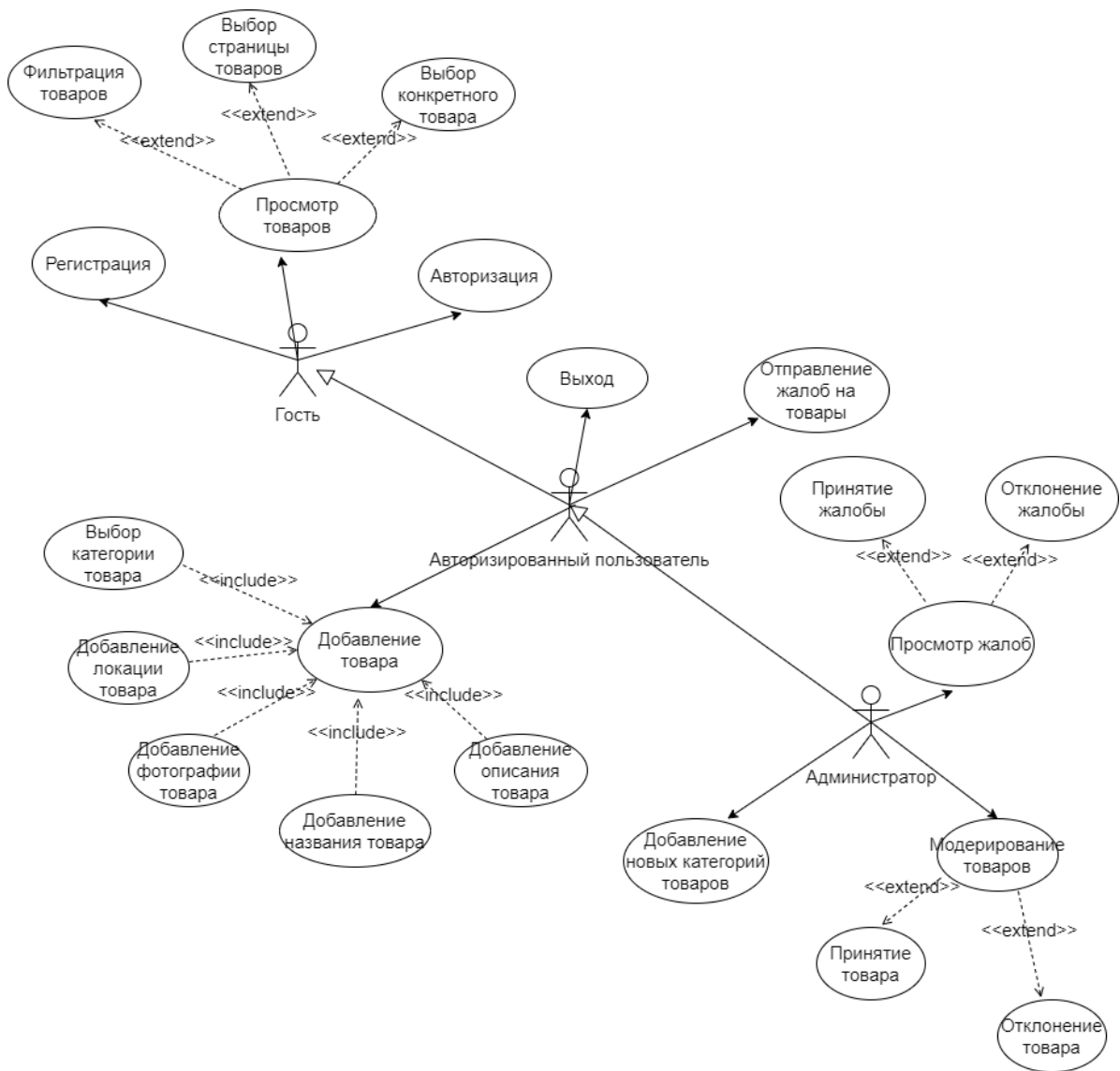
На основе анализа были сформулированы задачи, которые должны обеспечиваться разрабатываемой информационной системой:

- система фильтрации товаров;
- регистрация и авторизация через популярные социальные сети;
- система кармы для справедливости обмена;
- подбор объявлений через геолокацию;
- безвозмездный обмен вещами;
- система модерации.

Важной частью проектирования информационной системы является выбор средств проектирования. Для моделирования функциональной части сайта в нотации UML, а также для создания концептуальной модели базы данных было выбрано веб-приложение Draw.io. А для проектирования пользовательского интерфейса было выбрано приложение Figma.

Для описания разграничения групп пользователей по функционалу была выбрана диаграмма прецедентов (рисунок 1).





**Рис. 1. Диаграмма прецедентов**

Отличительная особенность концептуальной базы данных заключается в том, что при её создании не следует ориентироваться на какую-либо СУБД. Для описания концептуальной модели базы данных воспользуемся диаграммой сущность-связь (см. рисунок 2).

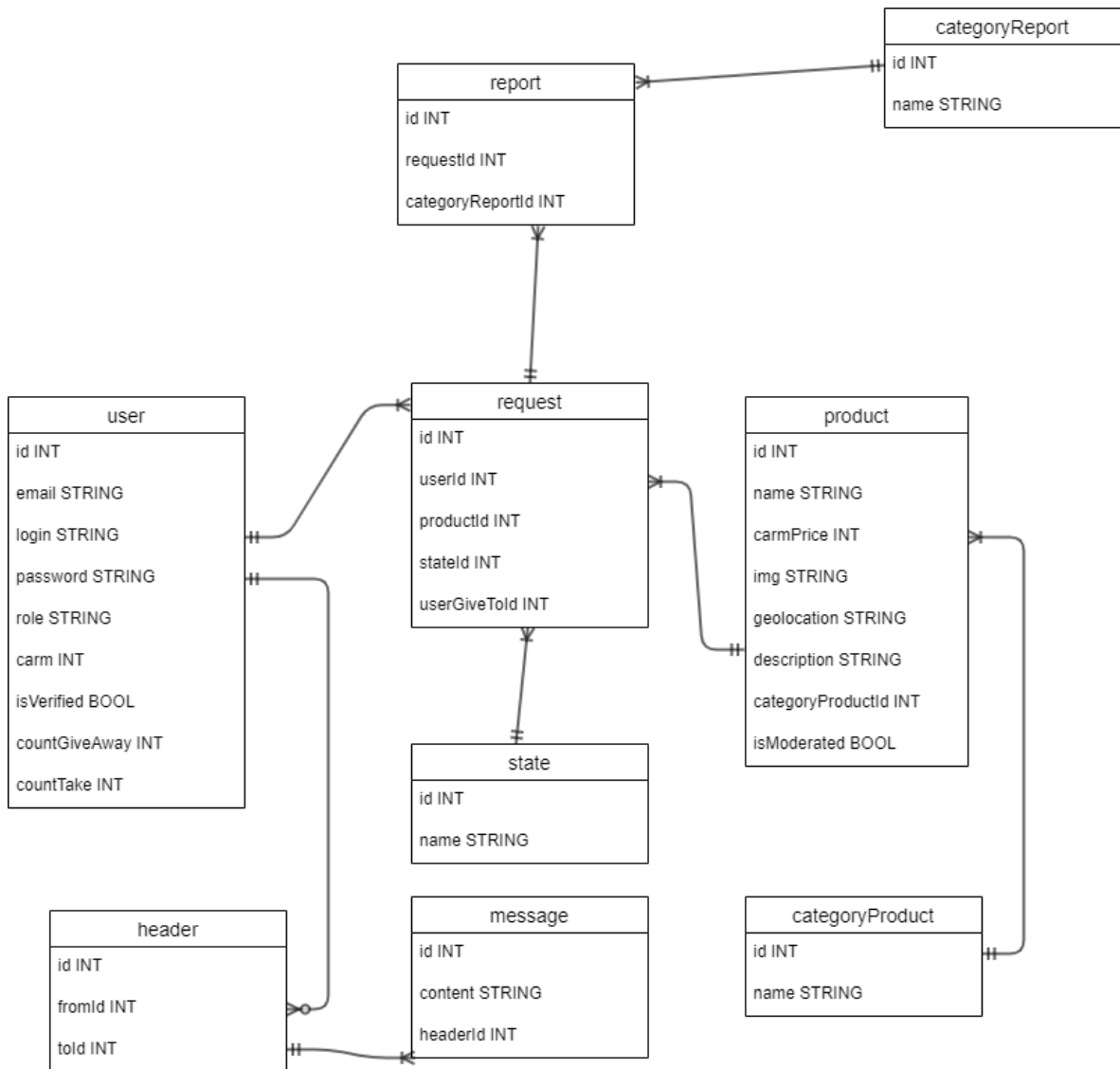


Рис. 2. Диаграмма “Сущность-связь”

При разработке информационной системы особенно важно, чтобы был интуитивно понятный и удобный интерфейс, ведь от этого зависит, будут ли пользователи посещать и пользоваться сайтом. Были разработаны компоненты системы в инструменте для проектирования интерфейсов Figma, который был до этого выбран.

При публикации товара, товар будет добавлен на сайт, где можно просмотреть содержимое объявления о выбранном товаре, так, содержимое будет включать: фотографию товара, геолокацию товара, название товара, категорию товара, стоимость товар, а также будет предоставлена возможность снять товар с публикации (см. рисунок 3)

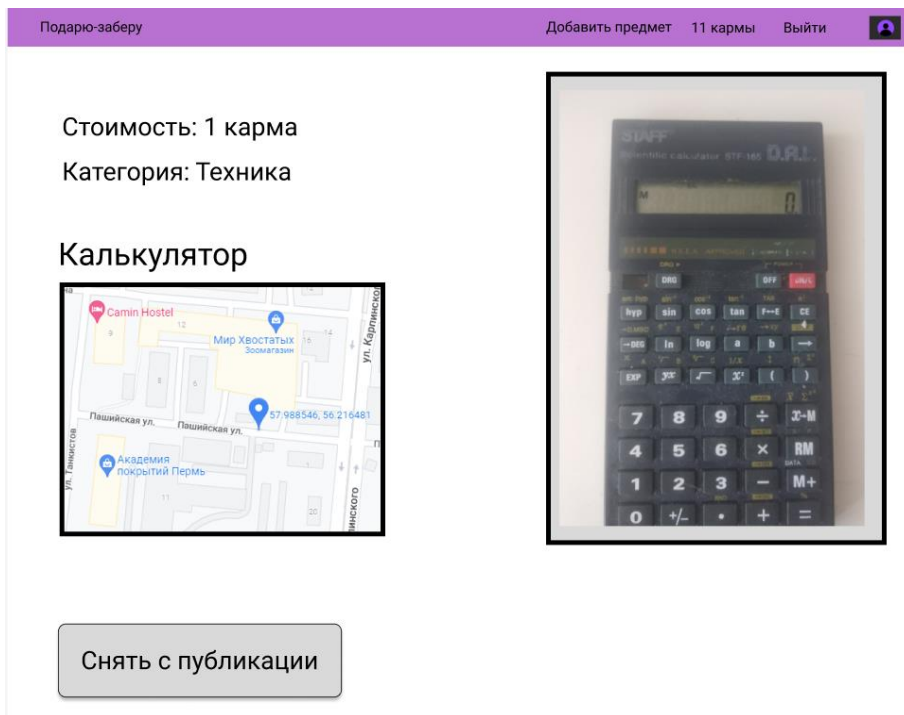


Рис. 3. Страница товара, опубликованного пользователем

Также был спроектирован чат между пользователями, так, он содержит в привычной форме сообщения. Заголовком чата будет являться название товара (рисунок 4).

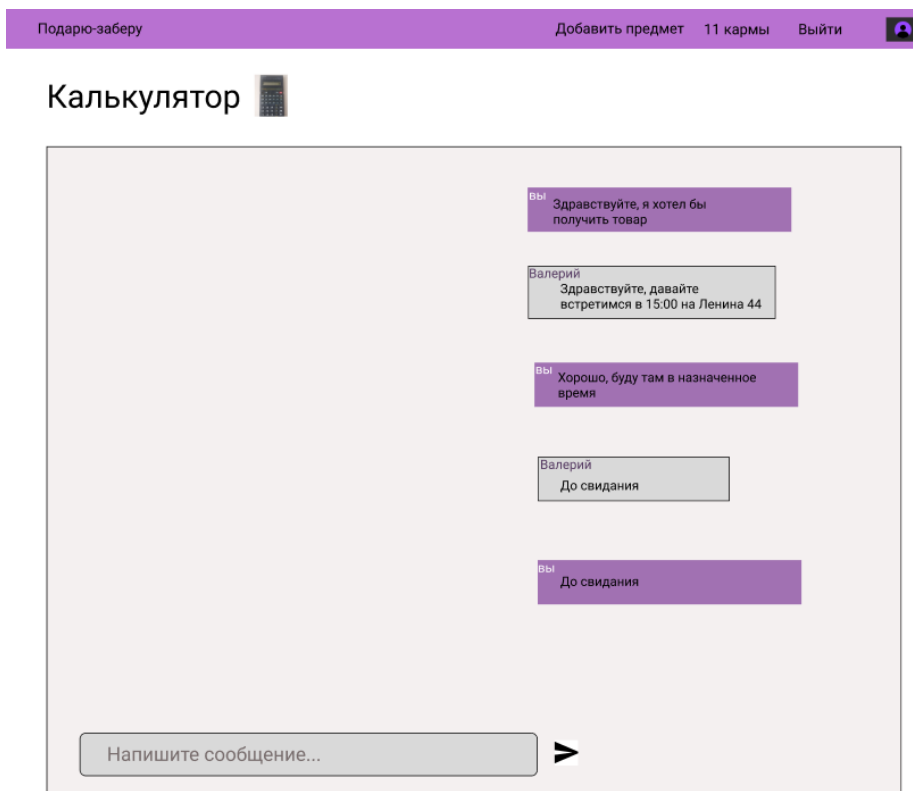


Рис. 4. Диалог между пользователями

Информационная система была разработана как веб-сайт. Клиентская сторона пользовательского интерфейса системы была разработана при помощи языка разметки HTML (HyperText Markup Language), CSS (Cascading Style Sheets) и языка программирования JavaScript.

Внутренняя часть веб-сайта (API) была разработана с помощью языка программирования JavaScript в среде выполнения Node JS, он хорошо документирован и прост в освоении, а также имеет крупное сообщество разработчиков, а так как пользовательская часть и API будут писаться на одном языке JavaScript. это делает процесс разработки намного удобнее и проще.

Для взаимодействия клиентской и серверной части была выбрана технология REST API (Representational State Transfer). Для того, чтобы управлять данными была выбрана схема разделения данных MVC (Model view controller). Для реализации клиентской части приложения было выбрано решение реализовывать SPA (single page application) приложение, то есть одностраничное приложение. Для управления состоянием приложения была выбрана библиотека MobX [2], она намного проще другой библиотеки для управления состоянием Redux, обладает понятной документацией, с этой библиотекой код становится намного понятнее и легче. Для связи с базой данных была использована ORM (Object-Relational Mapping) Sequelize.

При тестировании api веб-сайта была использована библиотека supertest, также было проведено Unit тестирование на различные функции системы и End-to-end для проведения поведенческого тестирования [3].

В дальнейшем развитии информационной системы планируется разработать приложение для мобильных устройств и добавить функционал к имеющемуся.

### **Библиографический список**

1. *Полухина Е.В., Стрельникова А.В.* От избыточного потребления – к устойчивому: феномен онлайн-дарообмена // Коллаборативное потребление. – 2014. – № 2. – С. 87-95.
2. *Mobx – управление состоянием вашего приложения [Электронный ресурс] // habr* Электрон. дан. – Режим доступа: <https://habr.com/ru/post/282578>, свободный (дата обращения: 30.05.22).
3. *Тестируем приложение nodejs [Электронный ресурс] // habr – Электрон. дан. – Режим доступа: https://habr.com/ru/post/162555*, свободный (дата обращения: 30.05.22).

## **DESIGN, DOCUMENTATION AND DEVELOPMENT OF THE “PODARU-ZABERU” APPLICATION**

*Ugrinov Valeriy A.*

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

*Kushev Vadim O.*

Perm State University, 15, Bukireva st., Perm, 614990, Russia, [vadimk@psu.ru](mailto:vadimk@psu.ru)

The article step-by-step analyzes the process of developing an information system through which users can exchange goods on the Internet for free. The subject and object of research are defined, the relevance of the development of this information system is formulated, and tasks that need to be solved during development are set. The analysis of existing Internet ad placement services is carried out, the tasks that should be provided by the information system being developed are defined. Design and development tools have been selected. Various diagrams have been created in UML notation to describe the functional part of an information system. A conceptual diagram describing how the data will be stored has also been created. After that, the user interface was developed, which includes the pages and components of the website. The features of the implementation and testing of the information system are also described.

Keywords: gratuitous exchange, bulletin board, website.

## ШИФРОВАНИЕ В СОЦИАЛЬНОЙ СЕТИ ВКОНТАКТЕ БРАУЗЕРНЫХ РАСШИРЕНИЙ

*Шабатов Есет Мамаевич*

Пермский государственный национальный исследовательский университет, 614990, Россия,  
г. Пермь, ул. Генкеля, 7, eset-2000@mail.ru

В данной работе исследуется проблема безопасного общения в социальной сети ВКонтакте. Проводится краткий теоретический обзор принцип обмена сообщениями ВКонтакте и алгоритмов шифрования. Обосновывается актуальность использования шифрования сообщений для ВКонтакте. Для исследования было изучено 5 существующих браузерных расширений, позволяющих шифровать сообщения ВКонтакте. Полученные данные подвергаются анализу. Осуществляется поиск браузерного расширения, которое позволяет быстро, удобно и эффективно шифровать сообщения. На основе полученных результатов написано расширение, которое устраняет недостатки изученных расширений, для браузера Chrome, позволяющее шифровать и расшифровывать сообщения ВКонтакте.

Ключевые слова: симметричное шифрование, алгоритм шифрования Магма, ГОСТ Р 34.12—2015

### **Введение**

В нашем мире на данный момент в сфере информационных технологий и их использовании достаточно актуальной остается проблема анонимности в социальных сетях. Хотят оставаться анонимными абсолютно все, начиная от малого бизнеса, в котором люди хотят обеспечить защиту коммерческой тайны при общении с рекламодателями, заканчивая простыми гражданами, которые используют социальные сети для того, чтобы удовлетворить потребность в общении. Это привело к тому, что с определенного момента времени стали появляться браузерные расширения для ВКонтакте, которые позволяют включать шифрование на стороне пользователей браузером и общаться в зашифрованном виде ВКонтакте.

Целью данной работы является разработка браузерного расширения для шифрования и расшифрования сообщений во ВКонтакте.

### **ВКонтакте, принцип обмена сообщениями и шифрование.**

На серверах ВКонтакте в основном используется операционная система Debian Linux, так как является одним из самых стабильных и проверенных дистрибутивов Linux.

Для балансировки нагрузки между серверами ВКонтакте используется вебсервер nginx, который работает в режиме reverse proxy. Сервер nginx должен поддерживать соединение с браузером пользователя и передавать запросы серверам ВКонтакте, которые отвечают за исполнение PHP-кода, а также они контролируют попадание результата запроса обратно в браузер пользователю.

Сервис позволяющий мгновенно обмениваться сообщениями внутри социальной сети реализован на node.js с использованием протокола XMPP.

Балансировка нагрузки между серверами происходит по многоуровневой схеме, включающая балансировку на уровне DNS и маршрутизацию запросов внутри системы, при этом разные сервера используются для разных типов запросов.

Для хранения используется собственная система управления базами данных ВКонтакте, написанная на языке программирования C, но подробностей насчет этой системы управления базами данных нет.

Сейчас во ВКонтакте отсутствует шифрование сообщений, а при использовании сквозного шифрования получится так, что переписка не будет храниться на серверах социальной сети. Во всех существующих мессенджерах, где применяется сквозное шифрование, история переписки хранится только на 7 устройствах и теряется, если удалить приложение или зайти в аккаунт мессенджера с другого устройства.

Если разработчики ВКонтакте начнут использовать такой вид шифрования, у пользователей перестанет синхронизироваться переписка между разными устройствами, это в свою очередь принесёт огромные неудобства пользователям ВКонтакте.

ВКонтакте также не может обеспечивать конфиденциальность общения ещё и из-за юридических тонкостей. ВКонтакте – российская социальная сеть, а в России принят пакет антитеррористических законов, обязывающий сервисы, подобные ВКонтакте, хранить переписку и записи звонков и предоставлять их специальным службам и полиции.

### **Алгоритмы шифрования**

Существует большое количество алгоритмов шифрования для шифрования сообщений. Существуют 2 метода шифрования:

1. Симметричное шифрование [1];
2. Ассиметричное шифрование [2].

В симметричном шифровании используется один и тот же ключ как для шифрования, так и для расшифровывания. К симметричному шифрованию относятся блочные шифры [3] и поточные шифры [4].

Блочные шифры обрабатывают информацию блоками определенной длины, применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки, называемыми раундами. Результатом повторения раундов является лавинный эффект — нарастающая потеря соответствия битов между блоками открытых и зашифрованных данных.

Одни из симметричных блочных шифров:

- DES;
- 3DES;
- AES;
- Магма [5];
- Kuznechik.

Поточные шифры обрабатывают информацию не блоками, а обрабатывают информацию над каждым битом исходного текста с использованием гаммирования.

Одни из популярных симметричных поточных шифров:

- RC4;
- SEAL;
- WAKE.

В асимметричном шифровании используется два ключа: для шифрования используется открытый ключ, который передается по незащищенному, доступному для наблюдения каналу и для проверки электронно-цифровой подписи; для расшифрования используется закрытый (секретный) ключ, который используется еще и для генерации электронной цифровой подписи.

Одни из популярных асимметричных шифров:

- RSA;
- DSA;
- Elgamal;
- Diffie-Hellman;
- ECDSA.

В ходе выполнения работы был выбран алгоритм шифрования Магма, так как он прост в реализации и имеет неплохую криптостойкость.

### **Исследование браузерных расширений для шифрования сообщений во ВКонтакте**

Для исследования были выбраны 5 браузерных расширений, которые по заявлениям авторов говорится, что они могут шифровать и расшифровывать сообщения во ВКонтакте: VxCrypt, messCrypter, VK Secure, VKEncrypt, The R2P для ВК.

Далее были исследованы каждые из этих пяти браузерных расширений. Единственный минус всех вышеперечисленных расширений состоит в том, что хоть и написано, что личные данные пользователей нигде не хранятся и не передаются третьим лицам, но никак это нельзя проверить, так как отсутствует возможность просмотра реализации этих расширений.

VxCrypt – браузерное расширение, которое позволяет шифровать/расшифровывать сообщения в ВКонтакте. Алгоритм, выбранный для шифровки и расшифровки сообщений – AES. Рядом с формой отправки вводится закрытый (секретный) ключ. После шифрования сообщения в начале сообщения задается маркер (метка) AESSTART, по которому и будет распознаваться, какие сообщения надо расшифровывать. Шифрованное сообщение отправляется, как и обычное. Далее после отправки сообщения браузерное расширение просматривает сообщения из диалога и по маркеру AESSTART расшифровывает сообщение и меняет фон сообщения на зеленый. Но на данный момент это расширение устарело и к текущей версии ВКонтакте ее применить нельзя (оно не будет работать).

messCrypter – браузерное расширение, которое позволяет шифровать/расшифровывать сообщения во ВКонтакте. Алгоритм, выбранный для шифровки и расшифровки сообщений – неизвестно. Рядом с адресной строкой есть кнопка, в которой вводится закрытый (секретный) ключ. Ключей можно задать не ограниченное количество, а текущий ключ задается нажатием над соответствующим ключом кнопки default. После шифрования сообщения в начале сообщения задается маркер (метка) messCrypter, по которому и будет распознаваться, какие сообщения надо расшифровывать. Шифрованное сообщение отправляется, как и обычное. Далее после отправки сообщения браузерное расширение просматривает сообщения из диалога и по маркеру messCrypter расшифровывает сообщение. Но на данный момент это расширение позволяет только шифровать сообщение, но отсутствует возможно расшифрования сообщения.

VK Secure – браузерное расширение, которое позволяет шифровать/расшифровывать сообщения во ВКонтакте. Является одним из самых удобных и эффективных браузерных



расширений для шифрования/расшифрования сообщений во ВКонтакте. Алгоритм, выбранный для шифровки и расшифровки сообщений – неизвестно. В поле с набором текстового сообщения есть кнопка, по нажатию которой вводится закрытый (секретный) ключ в диалоговом окне браузера. После шифрования сообщения в начале сообщения задается маркер (метка) `vk:secure`, по которому и будет распознаваться, какие сообщения надо расшифровывать. Шифрованное сообщение отправляется, как и обычное. Далее после отправки сообщения браузерное расширение просматривает сообщения из диалога и по маркеру `vk:secure` расшифровывает сообщение. На данный момент расширение полностью рабочее и загрузить бесплатно его может любой желающий из интернет-магазина Chrome. Единственный минус этого расширения состоит в том, что при вводе большого текстового сообщения, которое не влезает в рамки одного сообщения будет разделяться на 2 и более сообщения. И все части шифрованного сообщения, кроме первой, нельзя будет никак расшифровать.

VKEncrypt – браузерное расширение, которое позволяет шифровать/расшифровывать сообщения во ВКонтакте. Алгоритм, выбранный для шифровки и расшифровки сообщений – AES256. Под полем с набором текстового сообщения есть кнопка, по нажатию которой происходит переключение в режим шифрования без ввода секретного ключа. Секретный ключ генерируется на основе текущей даты, `seed'a`. И эта информация о секретном ключе содержится в шифрованном сообщении. После шифрования сообщения в начале сообщения задается маркер (метка) `Encrypted by VKEncrypt`, по которому и будет распознаваться, какие сообщения надо расшифровывать. Шифрованное сообщение отправляется, как и обычное. Далее после отправки сообщения браузерное расширение просматривает сообщения из диалога и по маркеру `Encrypted by VKEncrypt` расшифровывает сообщение, и помечает сообщение зеленым цветом. На данный момент расширение полностью рабочее и загрузить бесплатно его может любой желающий из интернет-магазина Chrome. Существенный минус этого расширения состоит в том, что при шифрованном сообщении присутствуют данные, необходимые для секретного ключа, то есть при желании со стороны разработчиков ВКонтакте можно попытаться расшифровывать сообщения пользователей, не имея при этом этого браузерного расширения.

The R2P для ВК – браузерное расширение, которое позволяет шифровать/расшифровывать сообщения во ВКонтакте. Алгоритм, выбранный для шифровки и расшифровки сообщений – AES и RSA. Справа от поля с набором текстового сообщения есть кнопка, по нажатию которой происходит переключение в режим шифрования без ввода секретного ключа. Публичный (открытый) ключ вводится после нажатия кнопки и выводится в диалоговое сообщение, после перезагрузки веб-страницы появляется в диалоговом сообщении закрытый (секретный) ключ. Маркер, которым помечаются сообщения для расшифрования, как в предыдущих расширениях, отсутствует. Шифрованное сообщение отправляется, как и обычное. Далее после отправки сообщения браузерное расширение просматривает сообщения из диалога и расшифровывает нужное сообщение, и помечает сообщение припиской (зашифровано). На данный момент расширение полностью рабочее и загрузить бесплатно его может любой желающий из интернет-магазина Chrome. Существенный минус этого расширения состоит в том, что при в диалоге фигурируют сообщения с открытым и секретным ключом, которые используются шифрования и расшифрования сообщений, то есть без опасная переписка между пользователями не происходит.

## Создание расширения для шифрования сообщений

Для того, чтобы создать аналогичное расширения для шифрования и расшифрования сообщений во ВКонтакте необходимо создать папку с расширением и предварительно создать в нем 2 файла: `manifest.json` и `vk_inject.js`. В файле `manifest.json` будет содержаться информация, необходимая для инициализации расширения в браузере. В нем содержится дескриптор расширения и ссылки на внедряемые файлы. В файле `vk_inject.js` будет содержаться непосредственно исполняемый код для браузера, написанный на языке программирования javascript. Написанный код будет отвечать за логику шифрования и расшифрования сообщений со стороны пользователя ВКонтакте. Также необходимой частью для создания удобного интерфейса являются картинки с обозначением включения/отключения шифрования. Эти картинки тоже будут содержаться в папке с расширением.

После того, как были описаны первичные файлы для расширения в ВК и было подключено это расширение в браузер, опишем следующий алгоритм для шифрования и расшифрования сообщений во ВКонтакте:

### 1. В файле `manifest.json`

- 1.1. указать имя расширения, описание расширения и версию расширения;
- 1.2. описать рабочую область расширения, в данном случае – “\*://vk.com/im\*”;
- 1.3. указать ссылку на исполняемый код, в данном случае – “vk\_inject.js”;
- 1.4. указать “run\_at”: “document\_end”;

### 2. В файле `vk_inject`

- 2.1. описать алгоритм шифрования Магма (ГОСТ Р 34.12—2015), который будет использоваться для шифрования/расшифрования сообщений;
- 2.2. добавить кнопку для включения/отключения шифрования рядом с кнопкой «Прикрепить фото или видео»;
- 2.3. подключить обработчик событий на кнопку включения/отключения шифрования;
- 2.4. при включении шифрования выводится браузерное диалоговое окно, в котором необходимо ввести ключ шифрования;
- 2.5. при включенном шифровании
  - 2.5.1. к зашифрованному сообщению добавляется метка «EncryptMessagesVK: »;
  - 2.5.2. при нажатии на кнопку «Отправить сообщение» происходит шифрование введенного сообщения и в поле для ввода сообщения происходит замена введенного сообщения на зашифрованное
    - 2.5.2.1. если зашифрованное сообщение превышает 4096 знаков, то выдать предупреждение, что сообщение слишком большое, и отменить отправку сообщения;
    - 2.5.2.2. сообщения, содержащие метку «EncryptMessagesVK: » должны расшифровываться;
- 2.6. при выключенном шифровании зашифрованные сообщения должны выводиться на экран пользователя в зашифрованном виде, то есть должна фигурировать приписка «EncryptMessagesVK: ».

## Заключение

В ходе выполнения данной работы было разработано и протестировано браузерное расширение для шифрования сообщений во ВКонтакте. Проведено сравнение браузерных расширений для шифрования сообщений в ВК и их критериев. Получен, реализован и протестирован алгоритм шифрования и расшифрования сообщений во ВКонтакте.

### Библиографический список

1. Шнайер Б. Типы алгоритмов и криптографические режимы. – 2002 г
2. Саломая А. Криптография с открытым ключом. – 1995 г. – с. 74-75
3. Shannon C., Company A. T. Communication Theory of Secrecy Systems (англ.) // Bell Syst. Tech. J. — Short Hills, N.J. – 1949 г. – с. 656-715
4. Шнайер Б. Глава 16. Генераторы псевдослучайных последовательностей и потоковые шифры // Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. — 2002 г. — 816 с.
5. ГОСТ Р34.12—2015. Национальный стандарт Российской Федерации – Информационная технология. КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Блочные шифры. – 2015 г. – с.9-10.

### ENCRYPTION IN THE SOCIAL NETWORK VKONTAKTE OF BROWSER EXTENSIONS

*Shabakhov Eset Mamaevich*

Perm State University, 7, Genkel st., Perm, 614990, Russia, eset-2000@mail.com

This paper investigates the problem of secure communication in the social network Vkontakte. A brief theoretical review of the principle of messaging in Vkontakte and encryption algorithms is carried out. The relevance of using message encryption for Vkontakte is substantiated. For the study, 5 existing browser extensions were studied that allow you to encrypt messages on Vkontakte. The data obtained are analyzed. A browser extension is being searched for that allows you to quickly, conveniently and efficiently encrypt messages. Based on the results obtained, an extension was written that eliminates the shortcomings of the studied extensions for the Chrome browser, which allows you to encrypt and decrypt messages on Vkontakte.

Keywords: symmetric encryption, Magma encryption algorithm, GOST R 34.12—2015

*Научное издание*

## **Актуальные проблемы математики, механики и информатики 2022**

Сборник статей по материалам студенческой конференции  
(г. Пермь, ПГНИУ, 25 мая – 10 июня 2022 г.)

Ответственные редакторы *А. В. Черникова, М. М. Бузмакова*

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 5,61 Мб  
Подписано к использованию 24.11.2022

---

Размещено в открытом доступе  
на сайте [www.psu.ru](http://www.psu.ru)  
в разделе НАУКА / Электронные публикации  
и в электронной мультимедийной библиотеке ELiS

Издательский центр  
Пермского государственного  
национального исследовательского университета  
614990, г. Пермь, ул. Букирева, 15