

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет инженерно-экономический
Кафедра экономической информатики
Дисциплина «Программирование сетевых приложений»

«К ЗАЩИТЕ ДОПУСТИТЬ»
Руководитель курсового проекта
Ассистент кафедры ЭИ
_____. Ю.В. Сильванович
_____. _____. 2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему:
«СИСТЕМА УПРАВЛЕНИЯ ПЕРСОНАЛОМ НА ИТ-ПРЕДПРИЯТИИ»

БГУИР КП 1-40 05 01-02 019 ПЗ

Выполнил студент группы 272303
ПИВОВАРЧИК Егор Дмитриевич

(подпись студента)

Курсовая работа представлена на
проверку _____. 2024

(подпись студента)

Минск 2024

РЕФЕРАТ

БГУИР КП 1-40 05 01-02 019 ПЗ

Пивоварчик Е.Д. Разработка автоматизированной системы управления персоналом на IT предприятии: курсовой проект / Е. Д. Пивоварчик. – Минск: БГУИР, 2024, П. 3. – 35 с.

Пояснительная записка 35 с., ?? рис., 0 источников, 0 приложения

В данном курсовом проекте исследуются вопросы, связанные с оптимизацией, автоматизацией процессов в управленческой сфере, а именно автоматизацией процессов управления персоналом в IT-сфере. Основой этого проекта можно считать анализ существующих методов и технологий в области управления, выявлению наиболее подходящих и эффективных для оптимизации работы предприятия.

Объектом исследования является процесс автоматизации работы предприятия в IT-сфере с использованием специализированного программного обеспечения.

Предметом исследования являются методы расчета заработной платы, оптимизации затраченного времени, методы управления своим персоналом в современном мире.

Целью курсового проекта является разработка автоматизированной системы управления персоналом любой компанией в IT-сфере, предоставление качественного, удобного и работоспособного приложения для использования как подчиненными, так и нанимающими.

Задачи курсового проекта включают в себя ознакомление с существующими возможностями оптимизации работы людей, оценить все их достоинства и недостатки. Провести анализ потребностей пользователей, найти особенности в пользователях IT-сферы, проектирование автоматизированной системы, базы данных и алгоритмов, учитывающих различные факторы, такие как расписание сеансов, загруженность залов, предпочтения зрителей и другие параметры. Разработка автоматизированной системы, а также её тестирование и оценку эффективности.

Результаты курсового проекта: рассмотрены основные бизнес-процессы предметной области. В результате было создано клиент-серверное приложение, способное автоматизировать работу персонала IT-сфер, обеспечить оптимальное взаимодействие между подчиненными и нанимающими, имеющее простой и понятный дизайн, поддерживающееся легкому масштабированию и применению.

СОДЕРЖАНИЕ

Введение	6
1 Анализ и моделирование автоматизированной системы управления персоналом.....	8
1.1 Описание деятельности IT-предприятия	8
1.2 Разработка функциональной модели деятельности IT-предприятия	10
1.3 Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований.....	15
1.4 Разработка информационной модели IT-предприятия	16
1.5 UML-модели представления разработанной автоматизированной системы IT-предприятия и их описание.	18
2 Проектирование и конструирование программного средства	23
2.1 Постановка задачи	23
2.2 Архитектурные решения	24
2.3 Описание алгоритмов, реализующих ключевую бизнес-логику разрабатываемого программного средства	25
2.4 Проектирование пользовательского интерфейса	28
2.5 Обоснование выбора компонентов и технологий для реализации программного средства	30
3 Тестирование и проверка работоспособности программного средства	34

ВВЕДЕНИЕ

В современных условиях стремительного развития информационных технологий и роста конкуренции на рынке труда, вопрос эффективного управления персоналом приобретает особую значимость. Для IT-предприятий, успех которых напрямую зависит от квалификации и мотивации сотрудников, внедрение автоматизированных систем управления персоналом (далее HRM — Human Resource Management) становится ключевым фактором повышения эффективности управления человеческими ресурсами.

В 2024 году IT-предприятия должны иметь стабильный способ управлять своим персоналом с помощью имеющихся технологий. На Белорусском и Мировом рынках в данный момент не каждая компания может похвастаться унифицированным способом управления своими кадрами.

Традиционные методы управления персоналом с использованием бумажных носителей и электронных таблиц больше не отвечают требованиям современного бизнеса. Сложность учета кадров, необходимость оперативного принятия управленческих решений и соблюдение трудового законодательства требуют применения интегрированных решений, которые позволяют автоматизировать рутинные процессы, обеспечивать прозрачность кадровых процедур и повышать производительность сотрудников.

Целью данной курсовой работы является оптимизация процесса взаимодействия сотрудников внутри IT-предприятия, которая позволит оптимизировать процессы кадрового учета, найма и адаптации сотрудников, автоматизировать расчет заработной платы, премий и анализ производительности. Система должна обеспечить удобный доступ к информации для руководителей и сотрудников, повысить точность обработки данных, а также снизить временные затраты на выполнение административных задач.

Для достижения данной цели необходимо выполнить следующие задачи:

- анализировать предметную область, выявить конкурентов на рынке
- смоделировать основные процессы системы;
- спроектировать взаимодействие ролей системы, базу данных;
- протестировать работу системы.

Актуальность. Приложения для управления человеческими ресурсами остаются актуальными и востребованными для компаний любого размера в 2024 году. Компании стремятся сократить ручную работу HR-отделов, внедряя автоматизированные системы для расчета зарплат, учетов отпусков и больничных, оценки сотрудников и управления талантами. Большинство существующих на данный момент HRM приложений имеют интеграцию с различными ERP-системами (например, SAP или Oracle), системами управления проектами (Jira, Trello) и т.д. По меркам любой IT-компании в 2024 году приложения HRM остаются не только актуальными, но и необходимыми в условиях растущей цифровизации бизнеса. Компании, которые внедряют

такие системы, получают конкурентное преимущество благодаря оптимизации HR-процессов, улучшению опыта сотрудников и повышению эффективности управления талантами.

Объектом исследования является процесс автоматизации взаимодействия работников ИТ-компаний с использованием специализированного программного обеспечения.

Предметом исследования являются методы оценки требований бизнеса, конкуренции и рынка, создания отчетных документов и оптимизация этих процессов.

В ходе работы будут рассмотрены различные технические аспекты проектирования и реализации системы, включая выбор технологий и инструментов разработки, анализ требований пользователей и разработку алгоритмов. Также будет уделено внимание вопросам безопасности и надежности системы, чтобы обеспечить пользователей защищенными и эффективными сервисами.

При разработке особое внимание будет уделено созданию удобного и интуитивно понятного интерфейса, дающего пользователям возможность легкого взаимодействия с компонентами системы. Результатом данной работы станет создание автоматизированной системы, которая предоставит пользователям удобные инструменты для взаимодействия с сотрудниками внутри компании, оптимизирует процессы распределения обязанностей и задач, расчета заработной платы, расширения отделов. Система реализует все необходимые функции для организации работы предприятия, что положительно скажется на эффективности его работы.

Курсовой проект выполнен самостоятельно, проверен в системе «Антиплагиат». Процент оригинальности составляет Q%. Цитирования обозначены ссылками на публикации, указанными в «Списке использованных источников». Скриншот приведен в приложении А (рисунок А.1).

1 АНАЛИЗ И МОДЕЛИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПЕРСОНАЛОМ

1.1 Описание деятельности IT-предприятия

IT-предприятия играют ключевую роль в цифровой экономике, разрабатывая и внедряя технологические решения, которые автоматизируют и улучшают процессы в различных отраслях. Основные направления деятельности, такие как разработка программного обеспечения, техническая поддержка, интеграция и адаптация новых технологий, аутсорсинг и консалтинг, требуют значительного количества человеческих ресурсов. IT-компании сильно зависят от компетенций, опыта и мотивации своих сотрудников. Специфика деятельности IT-предприятий подразумевает тесное взаимодействие между командами разработчиков, тестировщиков, менеджеров проектов и других специалистов для решения задач и достижения целей.

В условиях стремительного роста компании и увеличения числа сотрудников на первый план выходят задачи, связанные с оптимизацией работы с человеческими ресурсами. При значительном количестве сотрудников HR-менеджмент становится все более трудоемким процессом.

На HR-менеджеров возложено много задач, начиная от подбора и найма персонала, отбора лучших кандидатов и онбординга до организации зарплатной политики новоиспеченных сотрудников и обратной связи с ними по различным вопросам. При постоянно увеличивающемся составе предприятия эти задачи становятся все более и более громоздкими и требуют практически невозможного, что может оказать влияние на продуктивность предприятия. От эффективного взаимодействия команд, быстроты найма и адаптации новых сотрудников зависит успешная работа компании. Рассмотрим основные причины, по которым предприятию необходимо внедрение системы управления персоналом.

Рост числа сотрудников и сложности в управлении.

С увеличением количества работников возрастает потребность в эффективном управлении кадровым составом. Ведение данных о сотрудниках в ручном режиме ведет к росту ошибок, дублированию информации и потере данных.

Улучшение обучения и повышения квалификации

IT-индустрия требует постоянного профессионального роста и освоения новых технологий. Система HRM может включать модуль для организации обучения и сертификации сотрудников. Это упрощает отслеживание навыков персонала, планирование тренингов и развитие сотрудников в соответствии с целями компании.

Оптимизация процесса рекрутинга.

IT-компании сталкиваются с высокой конкуренцией за квалифицированных специалистов. Автоматизация процесса найма позволяет ускорить обработку заявок, отслеживать стадии собеседований и взаимодействовать с кандидатами, что повышает эффективность подбора персонала.

Учет рабочего времени и отпусков.

В условиях гибкого графика работы, удаленного режима и разветвленной структуры предприятия, ручной учет рабочего времени и отпусков становится трудоемким и неэффективным процессом. Система управления персоналом позволит автоматизировать эти процессы и минимизировать ошибки.

Оценка производительности сотрудников.

Для IT-компаний критически важно оценивать вклад каждого сотрудника в общий успех проекта. Система управления персоналом позволяет отслеживать выполнение задач, устанавливать цели и анализировать показатели эффективности работы сотрудников.

Оценка производительности сотрудников.

Многие IT-компании имеют уникальные процессы работы с персоналом, связанные с гибким графиком, проектной занятостью или распределенной структурой. HRM-система может быть адаптирована под конкретные потребности компании, чтобы учитывать все её особенности и минимизировать проблемы в управлении.

Автоматизация расчета заработной платы.

В условиях, когда компания использует различные системы оплаты труда (почасовая, фиксированная, бонусная система), система управления персоналом помогает автоматизировать процессы расчета заработной платы, премий, налогов и других выплат. Это уменьшает вероятность ошибок и ускоряет процесс выплат.

Вовлечение руководителей предприятий в HRM-процесс.

С внедрением системы руководители могут без значимой траты временных ресурсов оценить состояние своей компании и производительность того или иного отдела предприятия, что является не очень прагматичным без внедрения такой системы. Руководители смогут давать HR-менеджерам различные задачи по оптимизации HRM-процессов внутри компании и делать это с большей уверенностью, так как система обозначит все пробелы и недочеты в работе того или иного отдела.

Таким образом, внедрение системы управления персоналом на данном IT-предприятии позволит не только автоматизировать рутинные кадровые процессы, но и повысить производительность сотрудников, улучшить качество управления проектами, а также снизить административную нагрузку на кадровую службу.

1.2 Разработка функциональной модели деятельности IT-предприятия

Рассмотрим функциональную модель процесса выполнения проекта предприятием выполненную с помощью средств моделирования функций IDEF0 (разработан в 1981 году в рамках программы автоматизации промышленных предприятий) – нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Для начала необходимо сделать контекстную модель информационной системы. Контекстная диаграмма – самая верхняя диаграмма, на которой объект моделирования представлен единственным блоком с граничными стрелками. Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой.

На рисунке 1.1. – представлена контекстная диаграмма процесса «Выполнить проект».

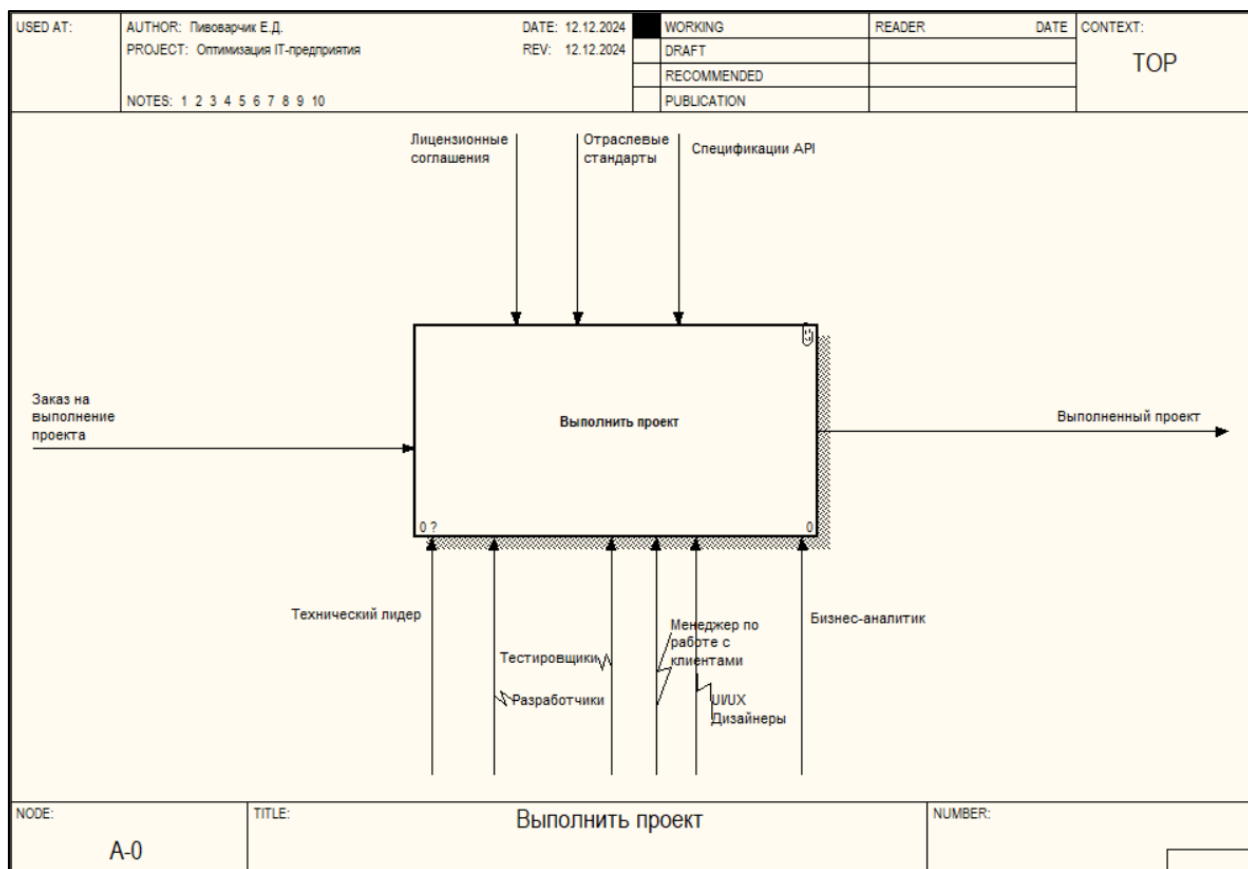


Рисунок 1.1. – Контекстная диаграмма «Выполнить проект»

Контекстная диаграмма содержит следующие параметры:

- входной параметр – заказ на выполнение проекта;
- выходной параметр – выполненный проект;

- управляющие параметры – лицензионные соглашения, отраслевые стандарты, спецификации API;
- исполнительные параметры – технический лидер, тестировщики, разработчики, менеджер по работе с клиентами, UI/UX дизайнеры, бизнес-аналитик.

На рисунке 1.2. представлена декомпозиция контекстной диаграммы процесса.

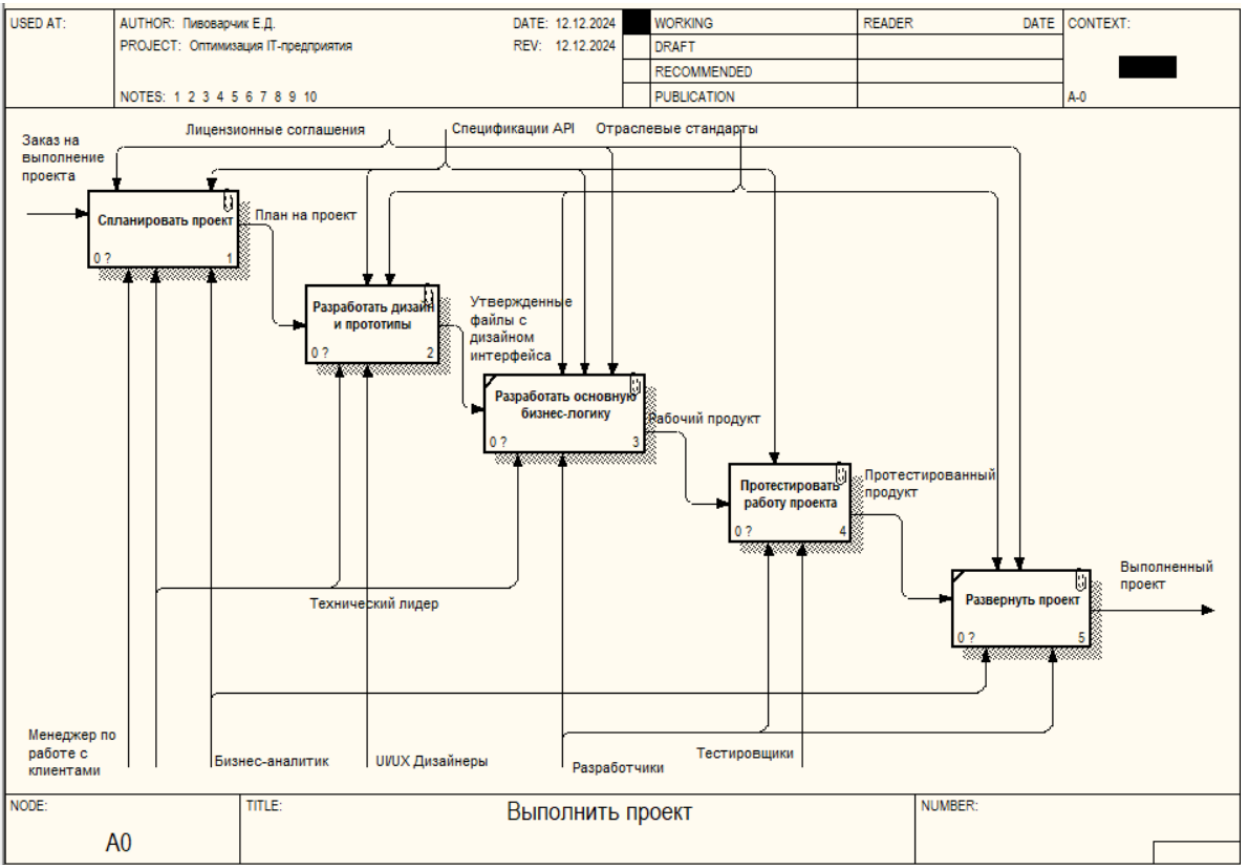


Рисунок 1.2. – Декомпозиция контекстной диаграммы

Декомпозиция контекстной диаграммы показывает, как распределяются задачи в процессе и последовательность их выполнения.

В декомпозиции показано как участники разработки (исполнители) распределяются по проекту. Чтобы выполнить проект его необходимо спланировать, разработать основной дизайн и прототипы, на их основе реализовать бизнес логику, протестировать работу проекта и потом выпустить или развернуть его.

Декомпозиция блока «Спланировать проект» представлена на рисунке 1.3.

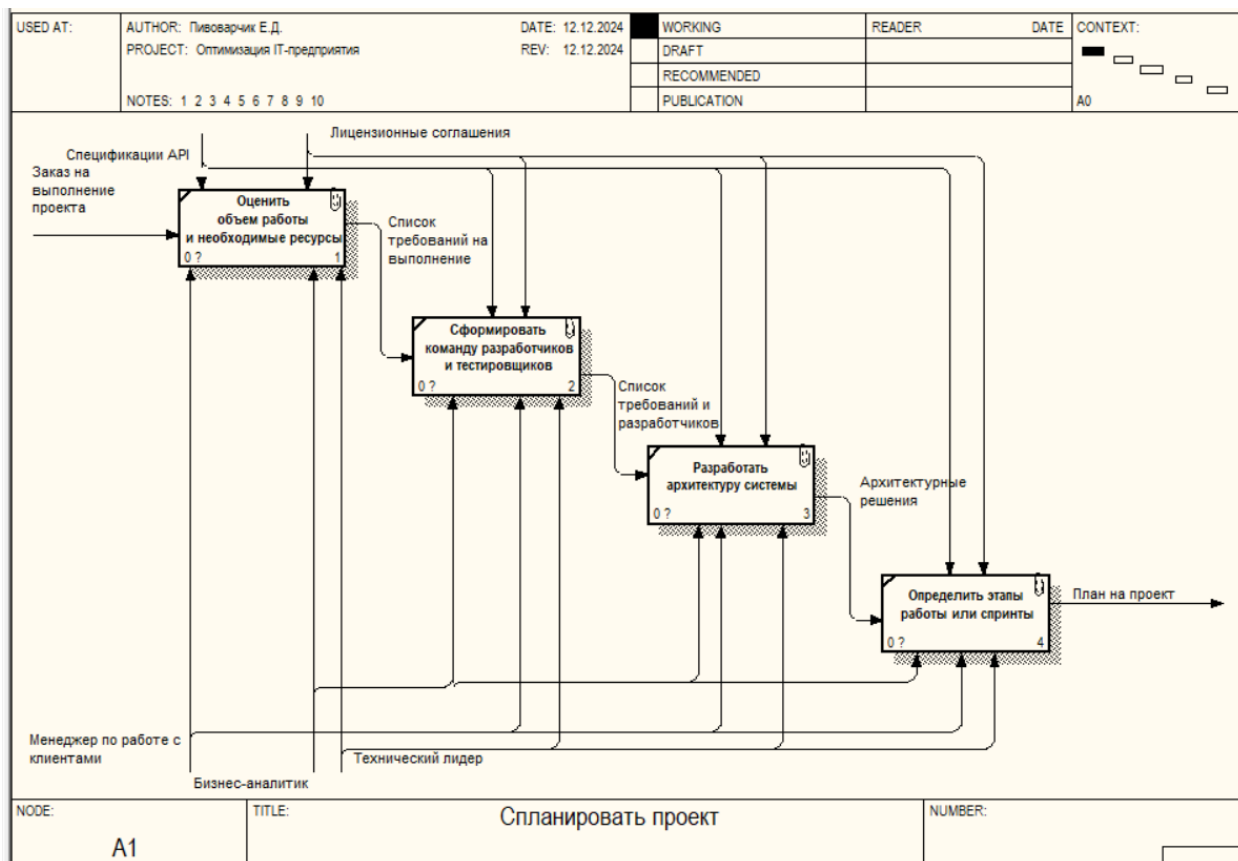


Рисунок 1.3. – Декомпозиция блока «Спланировать проект»

В результате декомпозиции получили 4 блока реализующих планирование проекта:

- оценить объем работы и необходимые ресурсы;
- сформировать команду разработчиков и тестировщиков;
- разработать архитектуру системы;
- определить этапы работы или спринты.

Чтобы разработать любой проект нужно оценить имеющиеся ресурсы и сформировать команду для реализации проекта, составить этапы работы чтобы уложиться в временные сроки реализации проекта.

Менеджер по работе с клиентами оценивает объем работы и необходимые ресурсы и обсуждает вместе с клиентом. Технический лидер участвует во всей планировке проекта, так как именно он разрабатывает основную архитектуру системы проекта.

Бизнес-аналитик оценивает необходимые ресурсы для выполнения проекта.

Декомпозиция блока «Разработать дизайн и прототипы» представлена на рисунке 1.4.

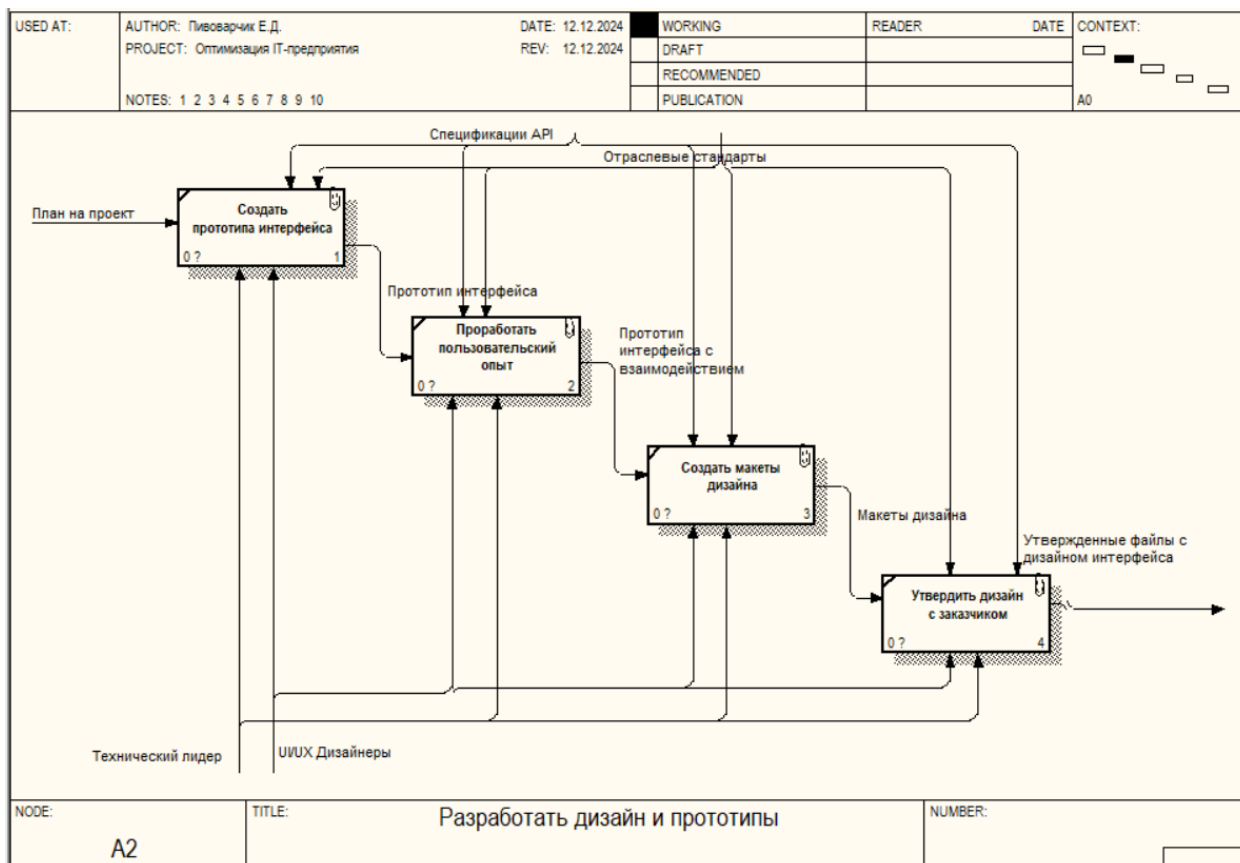


Рисунок 1.4. – Декомпозиция блока «Разработать дизайн и прототипы»

После планирования проекта начинается этап разработки основного дизайна проекта. Создание прототипов интерфейса требуется для реализации пользовательского опыта, а пользовательский опыт вместе с прототипами реализует основной пользовательский интерфейс. Процесс создания макетов дизайна включает создание визуального оформления, включая цветовую палитру, шрифты, иконки и прочие элементы. Результатом становится дизайн, который отображает финальный вид интерфейса. заказчик проверяет и утверждает итоговые макеты дизайна. Возможно внесение корректировок перед окончательным утверждением.

За разработку дизайна и прототипов почти полностью отвечают UI/UX дизайнеры, так как это является их основной ролью в создании проекта.

Декомпозиция блока «Протестировать работу проекта» представлена на рисунке 1.5.

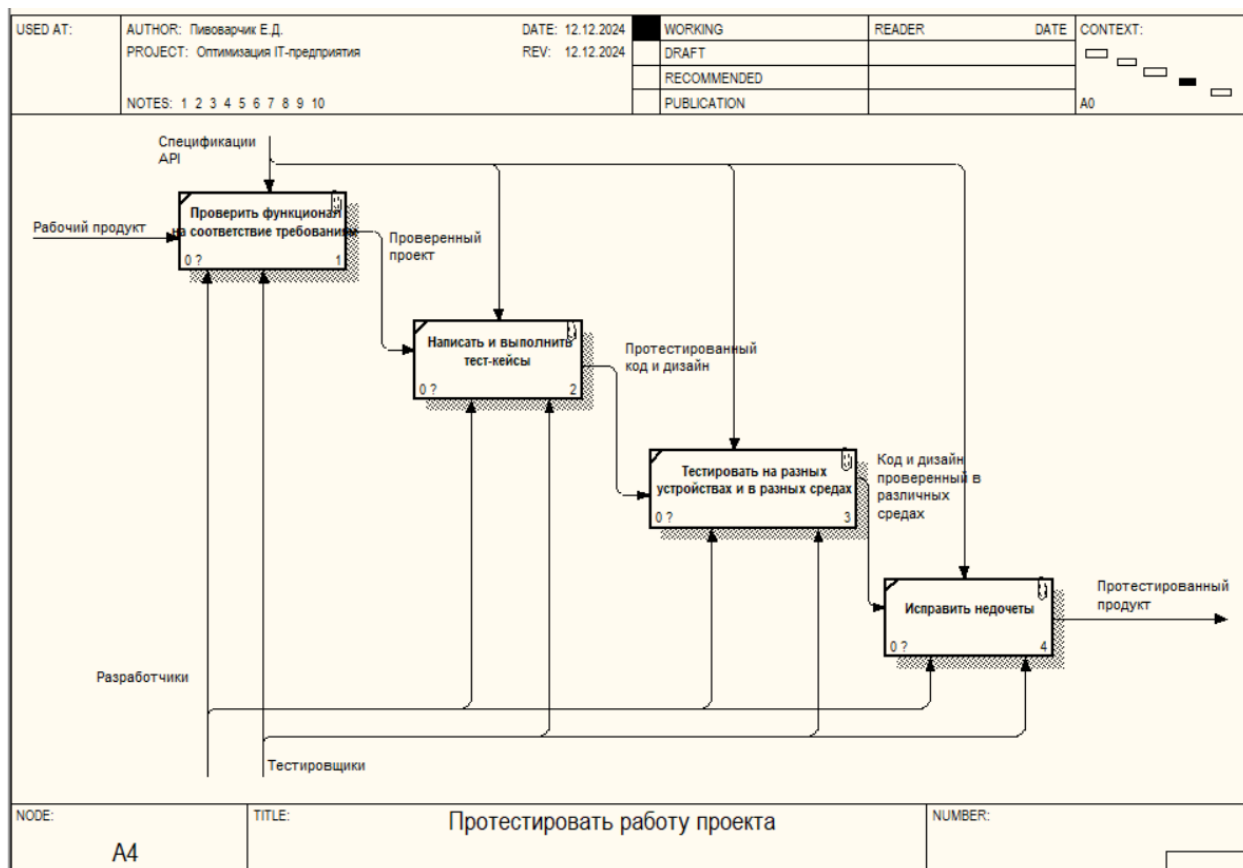


Рисунок 1.5. – Декомпозиция блока «Протестировать работу проекта»

В результате декомпозиции данного блока мы получили диаграмму, состоящую из 4-ёх процессов:

- проверить функционал на соответствие требованиям;
- написать и выполнить тест-кейсы;
- тестировать на разных устройствах и в разных средах;
- исправить недочеты.

Этот процесс является одним из ключевых процессов перед реализацией любого проекта, так как проект должен не допускать остановку работы базы данных или вылет сервера, так как это приведет к огромным проблемам при реализации.

Таким образом, была разработана функциональная модель процесса выполнения проекта в нотации IDEF0. Эта модель предоставляет комплексное представление ключевых этапов и элементов процесса реализации коммерческого заказа обеспечивая возможность оптимизации и повышения эффективности данного процесса. Использование нотации IDEF0 в разработке функциональных моделей позволяет более четко определить роли различных участников, структуру взаимодействия между ними и последовательность шагов в процессе покупки билета. Это способствует лучшему пониманию организации и улучшения этого процесса.

1.3 Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований

Перед тем, как приступить к созданию нашего программного продукта, необходимо провести тщательный анализ направления его разработки. Это ключевой этап, который поможет нам понять внутренние процессы и особенности разработки системы.

HRM-система должна быть универсальной, гибкой и масштабируемой, чтобы подходить для IT-компаний разного размера. Она должна включать автоматизацию рутинных процессов, поддержку удалённой работы и обеспечение высокой степени конфиденциальности данных сотрудников.

Система должна обеспечивать безопасность внутри своей компании, жесткий контроль авторизации и регистрации новых пользователей, так как она может хранить внутреннюю информацию о пользователях и юридических лицах внутри компании.

Хранение данных должно быть обеспечено на удаленном сервере с подключенной базой данных для безопасного доступа к данным и структурирования данных внутри приложения.

Задание состоит в создании функционирующего клиент-серверного GUI-приложения с самостоятельно установленным сервером, где должна быть предусмотрена возможность параллельной обработки запросов. Функциональные возможности серверной части должны насчитывать не менее 12 высокоуровневых вариантов использования, исключая тривиальные операции работы с базой данных (создание, добавление, удаление, редактирование записей). Клиентская часть представляет собой оконное приложение с использованием стандартных библиотек пользовательского интерфейса (JavaFX, SWING, AWT). Данные (логин, пароль), а также ресурсные данные и иная информация должны храниться в реляционной базе данных.

Исходные данные:

- версия JDK – 8 или выше;
- пользовательский интерфейс должен обеспечивать доступ ко всем функциям. Количество ролей – 2;
- операционная система Windows;
- СУБД MySQL;
- шифрование хранимых и передаваемых данных.

Применяя аналогичный подход, мы можем создать клиент-серверное приложение для автоматизации работы кинотеатра. Это приложение будет включать функции покупки и бронирования билетов на интересующий сеанс и выбора мест в зале, просмотр доступных сеансов, учета пользовательских комментариев. Благодаря использованию современных технологий и шифрования данных, наше приложение будет безопасным и надежным для всех пользователей.

1.4 Разработка информационной модели IT-предприятия

Информационная модель приложения представляет собой мнимое представление данных и их взаимосвязей в системе. Она отображает структуру информации, включая объекты, их свойства и отношения между ними, без привязки к конкретным технологиям хранения данных. Эта модель помогает разработчикам и аналитикам понять, как данные будут организованы, обрабатываться и взаимодействовать друг с другом внутри приложения. Логическая модель служит основой для проектирования базы данных и разработки бизнес-логики, обеспечивая согласованность и целостность данных. В конечном итоге, логическая модель помогает создать эффективное и надежное приложение, удовлетворяющее потребности пользователей и бизнес-задачи.

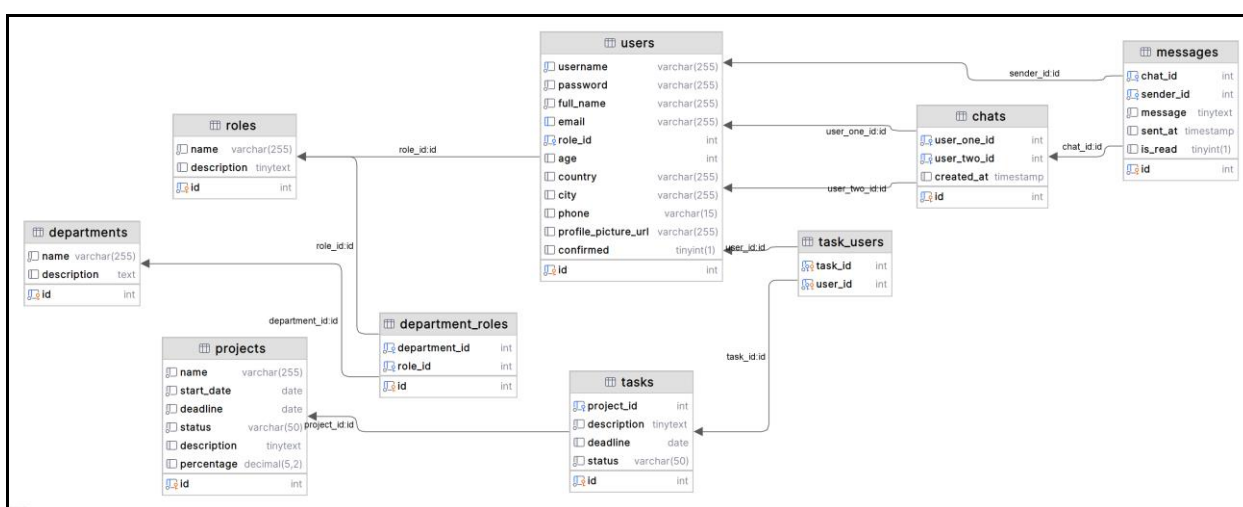


Рисунок 1.6. – Основная структура информационной модели

На рисунке 1.6. показана основная структура информационной модели приложения

Основные сущности, которые могут быть представлены в информационной модели в предметной области кинотеатра включают в себя:

- «projects» – сущность, которая содержит информацию о проектах компании;
- «roles» – сущность, которая содержит информацию о должностях IT-работников в компании;
- «users» – сущность, которая содержит информацию о пользователях приложения;
- «tasks» – сущность, которая содержит информацию о пользователях приложения;
- «chats» – сущность, которая содержит информацию о личных чатах пользователей приложения;

– «messages» – сущность, которая содержит информацию о всех сообщениях внутри приложения, ссылку на чат и участников чата;

– «departments» – сущность, которая содержит информацию о департаментах предприятия, их названия и описание.

Опишем связи между сущностями:

– «projects», «tasks» – связь один ко многим (один проект может включать множество задач) – осуществляется через поле `project_id` в таблице `tasks`, которое является внешним ключом, ссылающимся на `id` таблицы `projects`;

– «departments», «roles» – связь многие ко многим (один департамент может включать множество должностей, и одна должность может быть передана многим департаментам) – осуществляется через таблицу `department_roles`, содержащая внешние ключи `department_id` и `role_id`;

– «roles», «users» – связь один ко многим (одна роль может быть присвоена множеству пользователей) – осуществляется через поле `role_id` в таблице `tasks`, которое является внешним ключом, ссылающимся на `id` таблицы `roles`;

– «tasks», «users» – связь многие ко многим (одна задача может быть назначена нескольким пользователям, один пользователь может выполнять несколько задач) – осуществляется через таблицу `task_users`, содержащая внешние ключи `task_id` и `user_id`. Первичный ключ таблицы `task_users` – комбинация полей `task_id` и `user_id`, что предотвращает дублирование записей;

– «chats», «messages» – связь один ко многим (один чат может содержать множество сообщений) – осуществляется через поле `chat_id` в таблице `messages`, оно является внешним ключом;

– «messages», «users» – связь один ко многим (один пользователь может быть отправителем множества сообщений) – осуществляется через поле `sender_id` в таблице `messages`, оно является внешним ключом.

Рассмотрим представленные таблицы и выполним процесс нормализации, чтобы привести их к третьей нормальной форме.

Для соответствия 1-ой нормальной форме все атрибуты таблицы должны быть атомарными, т.е. каждый атрибут должен содержать только одно значение в каждом ряду. Также не допускаются повторяющиеся группы данных.

В представленной базе данных каждая ячейка содержит одно значение и нет повторяющихся групп данных, значит база данных уже находится в 1НФ.

Для соответствия 2-ой нормальной форме таблица должна быть в 1-ой нормальной форме и все не ключевые атрибуты должны зависеть от всего первичного ключа. То есть, если таблица имеет составной ключ, то каждый атрибут должен зависеть от всего ключа, а не от его части.

В таблице `task_users` поля `task_id` и `user_id` соответствуют внешним ключам, что значит, что она соответствует 2-ой нормальной форме.

База данных соответствует 2-ой нормальной форме, поскольку все не ключевые атрибуты зависят от первичного ключа и нет частичных зависимостей.

Для соответствия 3-ей нормальной форме таблица должна быть во 2-ой нормальной форме и не должна содержать транзитивных зависимостей, т.е. не ключевые атрибуты не должны зависеть от других не ключевых атрибутов.

Транзитивных зависимостей в базе данных нет.

Значит, база данных находится в 3-ей нормальной форме, так как в каждой таблице все не ключевые поля зависят только от ключевых.

Таким образом, информационная модель ИТ-предприятия помогает понять, какие сущности взаимодействуют между собой и какие связи между ними существуют, обеспечивая структурированное представление данных, которое является одним из ключевых требований к системе такого калибра.

1.5 UML-модели представления разработанной автоматизированной системы ИТ-предприятия и их описание.

UML (Unified Modeling Language) — это универсальный язык для визуализации, описания, проектирования и документирования систем программного обеспечения. Он представляет собой стандарт, который используется для моделирования различных аспектов систем, начиная от архитектуры и структуры до поведения и взаимодействия. UML помогает разработчикам, архитекторам и бизнес-аналитикам наглядно представить концепции и логику программных систем.

Основная цель UML — облегчить коммуникацию между всеми участниками процесса разработки, предоставляя понятный и структурированный способ передачи информации о системе. UML охватывает множество аспектов разработки, таких как моделирование классов, взаимодействий, процессов и состояний. Это позволяет проектировать систему до её реализации, выявлять потенциальные проблемы и упрощать внесение изменений.

Диаграмма вариантов использования (Use Case Diagram) — это инструмент из UML, который используется для описания взаимодействий между пользователями (акторами) и системой. Она помогает визуально представить, какие функции система должна выполнять с точки зрения пользователя. Основная цель такой диаграммы — показать, какие сценарии использования существуют в системе и как они связаны с различными ролями.

На рисунке 1.7. предоставлена диаграмма вариантов использования приложения.

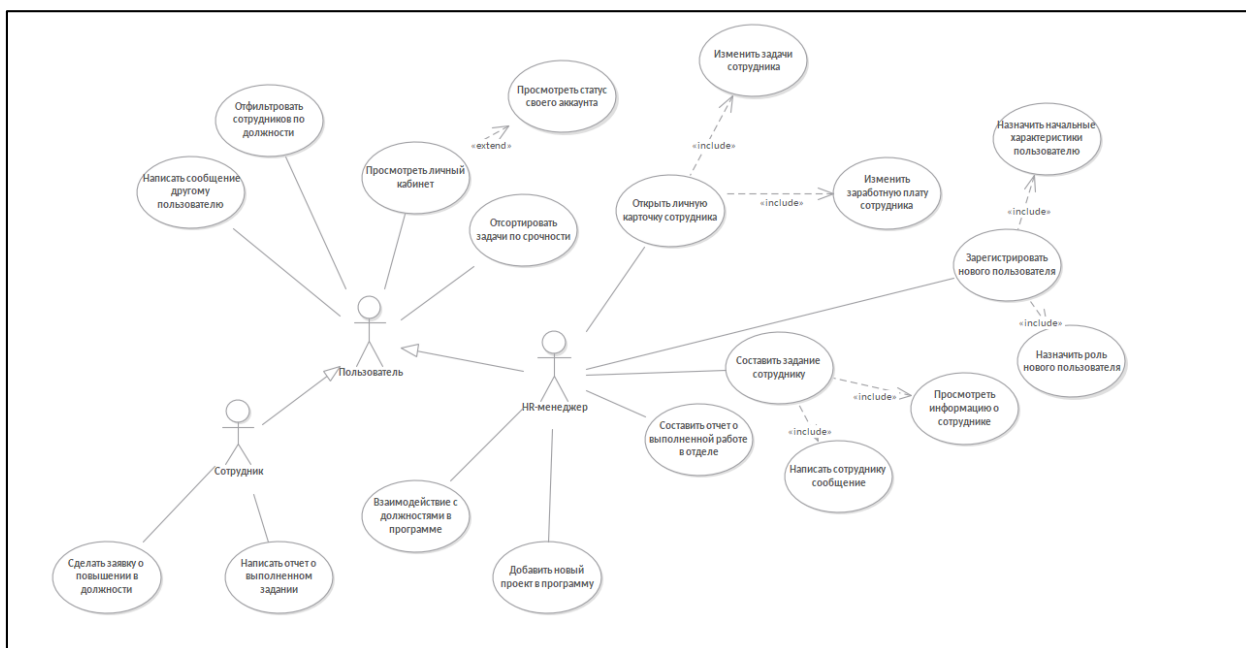


Рисунок 1.7. – Диаграмма вариантов использования

Варианты использования предназначены для определения функциональных требований к системе и управляют всем процессом разработки. Варианты использования, выступая в роли сценариев или кейсов, играют ключевую роль в выявлении потребностей конечных пользователей и становятся основой для формирования функциональных требований к системе. Они активно взаимодействуют с процессом разработки, ориентируя команду разработчиков на реальные потребности пользователей и обеспечивая более точное соответствие конечного продукта ожиданиям заказчика.

Актёрами на данной диаграмме являются: пользователь, HR-менеджер и обычный сотрудник. В роли администратора приложения будет выступать HR менеджер так как он является регулирующим звеном во взаимодействии сотрудников как внутри предприятия, так и в приложении.

HR менеджер и сотрудник компании наследуют свое поведение от пользователя который может написать сообщение другому пользователю, просмотреть свой личный кабинет, отфильтровать сотрудников по критерию, отсортировать свои задачи по срочности их выполнения.

HR менеджер обладает полными возможностями, начиная от CRUD операций с пользователями заканчивая назначением задач различным пользователям. Этот актёр может составлять отчеты о проделанной работе департаментами, составлять задачи сотрудникам, добавлять проекты в систему.

Сотрудник же, зарегистрировавшись, может реализовывать свои рабочие функции дистанционно, а именно:

- оставлять отметки о выполненных задачах и оконченных проектах;
- писать различным сотрудникам компании;

- следить за статусом своего аккаунта;
- написать запрос менеджеру о повышении заработной платы.

Таким образом HR менеджер может контролировать работу над проектом с помощью распределения обязанностей над проектом и общением с пользователями с помощью встроенного чата.

Пользователи при авторизации в системе будут записаны в лог авторизаций и HR менеджер может запросить статистику авторизаций в системе, что сможет упростить систему управления пользователями.

Диаграмма последовательности — это тип диаграммы в языке моделирования UML (Unified Modeling Language), который используется для визуального представления взаимодействий между объектами системы в процессе выполнения. На ней отображаются различные объекты, их сообщения и действия, происходящие между ними в рамках выполнения конкретной задачи или сценария. Основная цель диаграммы последовательности — показать, как объекты взаимодействуют между собой в определенном временном контексте, акцентируя внимание на последовательности вызовов методов, передаче данных и изменениях состояний.

В данном проекте на диаграмме последовательности удобно отразить логику процесса авторизации пользователя, так как диаграмма помогает увидеть какие функции и какие методы происходят на разных стадиях выполнения программы.

Диаграмма последовательности помогает анализировать логику работы системы и понять, как происходят взаимодействия в ответ на события. Она часто используется для описания сложных взаимодействий между компонентами, выявления возможных ошибок или улучшений в процессе взаимодействия, а также для планирования и документирования работы системы на разных уровнях.

Диаграмма последовательности представлена на рисунке 1.8.

На ней мы можем видеть актёра – пользователя который с помощью интерфейса AuthorizationLayout вызывает функцию login() которая отправляет через LoginRequest запрос к серверу на авторизацию пользователя.

В зависимости от ответа сервера реализуется логика авторизации пользователя в приложении. Если сервер по введенному логину пользователя находит его в базе данных, то пользователь может пройти дальше по логике программы, если же логин не виден в базе данных значит клиенту необходимо пройти ступень регистрации.

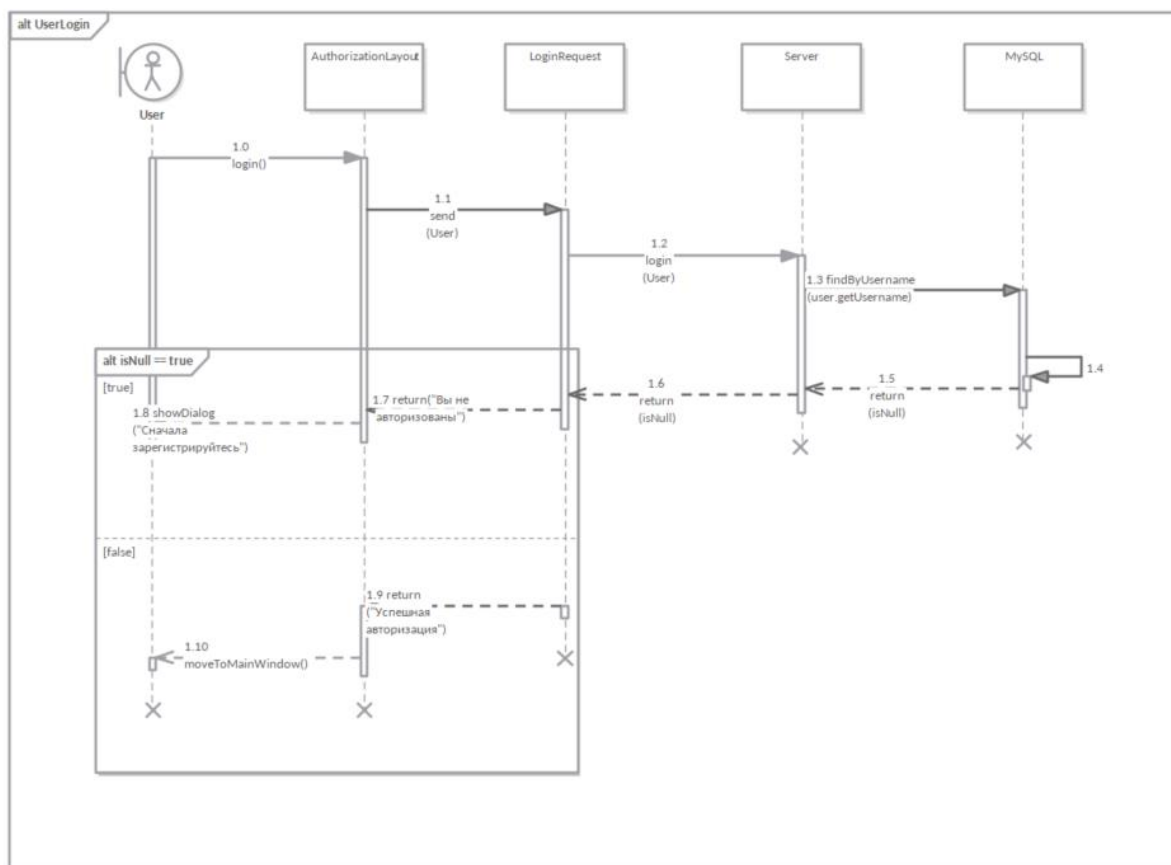


Рисунок 1.8. – Диаграмма вариантов использования

Диаграмма состояний (state machine diagram) — это один из видов диаграмм в UML, который используется для описания поведения объекта в системе через его состояния и переходы между ними. Она отражает, как объект изменяет своё состояние под воздействием различных событий.

Диаграмма состояний позволяет разработчикам и аналитикам глубже понять и спланировать сложные системы, в которых объект проходит через множество состояний, зависящих от событий и условий.

Для системы HRM хорошо подойдет описание состояния выполнения проекта которая представлена на рисунке 1.9.

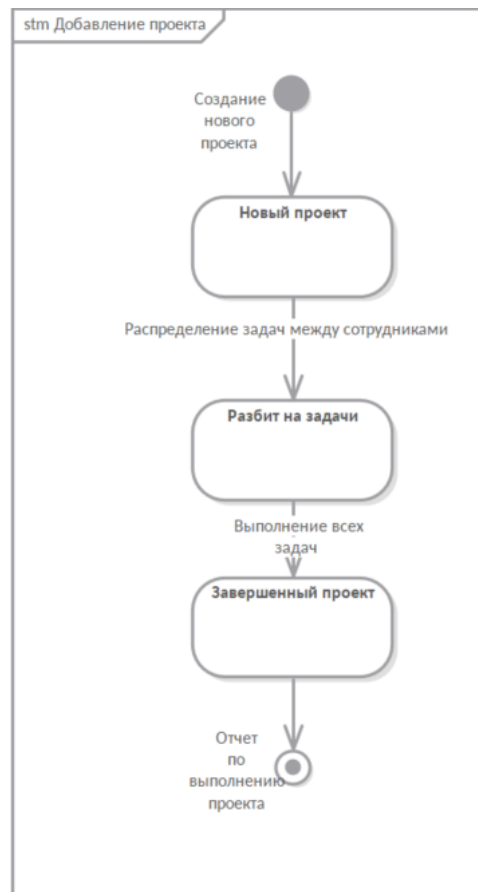


Рисунок 1.9. – Диаграмма состояний

Таким образом UML диаграммы играют ключевую роль в разработке программного продукта, обеспечивая визуальные средства для моделирования различных аспектов системы. Они позволяют разработчикам лучше визуализировать функционал системы, что приводит к намного менее трудоемкой разработке программного средства. Использование UML диаграмм повышает качество проектирования, снижает риски и повышает эффективность процесса разработки программного обеспечения.

2 ПРОЕКТИРОВАНИЕ И КОНСТРУИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

2.1 Постановка задачи

Система управления персоналом на IT-предприятии должна обеспечивать автоматизацию и оптимизацию процессов, связанных с управлением кадрами, их развитием и эффективным использованием. Она должна помогать в управлении информацией о сотрудниках, поддерживать процессы найма, адаптации, обучения и повышения квалификации. Важной задачей является обеспечение прозрачности и удобства работы с графиками, отпускными, больничными и другими аспектами рабочего времени.

Система должна учитывать потребности в оценке эффективности работы сотрудников, помогая руководителям устанавливать цели, отслеживать их выполнение и предоставлять обратную связь. Она должна упрощать коммуникацию внутри компании, обеспечивать доступ к документам и данным, связанным с трудовой деятельностью, а также предоставлять аналитические инструменты для оценки потребностей в кадрах и планирования ресурсов.

Кроме того, система должна обеспечивать соответствие законодательным требованиям в области трудового права, включая ведение документации, отчетности и соблюдение конфиденциальности данных.

Система внутри предприятия должна реализовывать механизм ограниченной авторизации. Так как это система внутри предприятия она не должна допускать влияния извне и авторизации пользователей, не связанных с этим предприятием. Должен быть реализован механизм авторизации пользователя через аккаунт администратора, и вторичная регистрация с аккаунта пользователя.

Система должна поддерживать возможность автоматического составления отчетной документации по проделанной работе за какой-то установленный период времени, чтобы автоматизировать этот процесс.

Должен быть реализован способ коммуникации между пользователями, для поддержания всестороннего взаимодействия внутри программного продукта. Должен быть реализован механизм динамического изменения данных, обновления информации в базе данных.

Важной функциональной особенностью является внедрение возможности ведения списка зарегистрировавшихся пользователей и составление отчетной документации по использованию системы за определенный период времени.

Все эти задачи направлены на создание интуитивно понятного, удобного и персонализированного интерфейса, который поможет работникам подбирать для себя удобное время для реализации проектных решений, уменьшить число часов нахождения в офисе, легкое взаимодействие со всей

инфраструктурой предприятия, и закладывание огромных возможностей к масштабированию.

2.2 Архитектурные решения

Для реализации курсовой работы по разработке системы управления персоналом (HRM) была выбрана комбинация технологий JavaFX, Hibernate, Gson и TCP/IP. Такое решение обусловлено их функциональными возможностями, которые наиболее полно отвечают требованиям проекта

JavaFX представляет собой современный фреймворк для разработки графических пользовательских интерфейсов. Он обеспечивает широкие возможности для создания интуитивно понятных, удобных и визуально привлекательных приложений. Поддержка CSS и FXML позволяет реализовать гибкое оформление и разделить бизнес-логику от представления, что упрощает разработку и последующее сопровождение системы. В рамках HRM-приложения JavaFX используется для построения интерфейса, включающего карточки сотрудников, формы добавления и редактирования данных, а также для взаимодействия с пользователем через модальные окна. Это повышает удобство работы пользователей, особенно в условиях интенсивного взаимодействия с большими объемами данных.

Hibernate был выбран как ORM-инструмент для работы с базой данных, так как он упрощает взаимодействие между объектами Java и реляционными базами данных. Вместо написания сложных SQL-запросов, разработчику достаточно определить классы-сущности и связи между ними, что позволяет сосредоточиться на бизнес-логике приложения. Hibernate обеспечивает поддержку транзакций, кэширования и автоматическую генерацию схемы базы данных, что критически важно для HRM-системы, которая должна обеспечивать надежное хранение и обработку информации о сотрудниках, их позициях, графиках работы и других данных. Кроме того, Hibernate помогает минимизировать риск ошибок, связанных с манипуляцией данными, благодаря встроенным механизмам валидации.

Технология Gson была выбрана для работы с форматом JSON, который часто используется для передачи данных между различными компонентами приложения. Gson позволяет легко сериализовать и десериализовать объекты Java в формат JSON и обратно, что упрощает интеграцию между клиентской и серверной частью системы. В контексте HRM-приложения Gson применяется для передачи данных о сотрудниках, ролях и других сущностях через сеть или для временного хранения конфигураций и пользовательских настроек. Легкость и гибкость Gson делают его идеальным выбором для работы с большими объемами структурированной информации.

TCP/IP используется для обеспечения сетевого взаимодействия между различными компонентами системы. Эта технология гарантирует надежность и стабильность передачи данных, что критично для корпоративных приложений. В HRM-системе, которая может включать серверную часть для

хранения базы данных и клиентскую часть для доступа к ней, TCP/IP позволяет реализовать обмен данными между клиентами и сервером, поддерживая как запросы на получение информации, так и выполнение операций, например, добавление или удаление данных. Использование TCP/IP обеспечивает возможность масштабирования системы, например, подключение удаленных пользователей или интеграцию с другими корпоративными приложениями.

В совокупности эти технологии позволяют создать функциональное, надежное и масштабируемое приложение для управления персоналом. JavaFX отвечает за удобство взаимодействия с пользователем, Hibernate обеспечивает надежную работу с базой данных, Gson позволяет легко и быстро передавать данные в JSON-формате, а TCP/IP гарантирует стабильность сетевого взаимодействия. Такой подход позволяет максимально эффективно решить задачи, поставленные в рамках курсовой работы, и реализовать систему, соответствующую современным требованиям разработки программного обеспечения.

2.3 Описание алгоритмов, реализующих ключевую бизнес-логику разрабатываемого программного средства

Блок-схемы играют важную роль в процессе проектирования и разработки программных систем, так как они предоставляют визуальное представление последовательности действий, алгоритмов или процессов. Использование блок-схем позволяет наглядно представить логику работы системы или её отдельных компонентов, что значительно облегчает понимание и анализ алгоритмов. Благодаря своей структурированности блок-схемы помогают разработчикам обнаружить возможные логические ошибки на ранних этапах проектирования, тем самым снижая затраты на исправление ошибок в будущем.

Кроме того, блок-схемы способствуют упрощению коммуникации внутри команды, так как визуальная форма подачи информации более интуитивна и понятна для всех участников проекта, независимо от их уровня технической подготовки. Они также являются важным инструментом документации, обеспечивая легкий доступ к описанию алгоритмов или процессов даже спустя длительное время после завершения разработки. Таким образом, блок-схемы играют ключевую роль в обеспечении эффективности, точности и прозрачности процессов проектирования и разработки.

На рисунке 2.1. представлена блок схема алгоритма добавления нового проекта в систему.

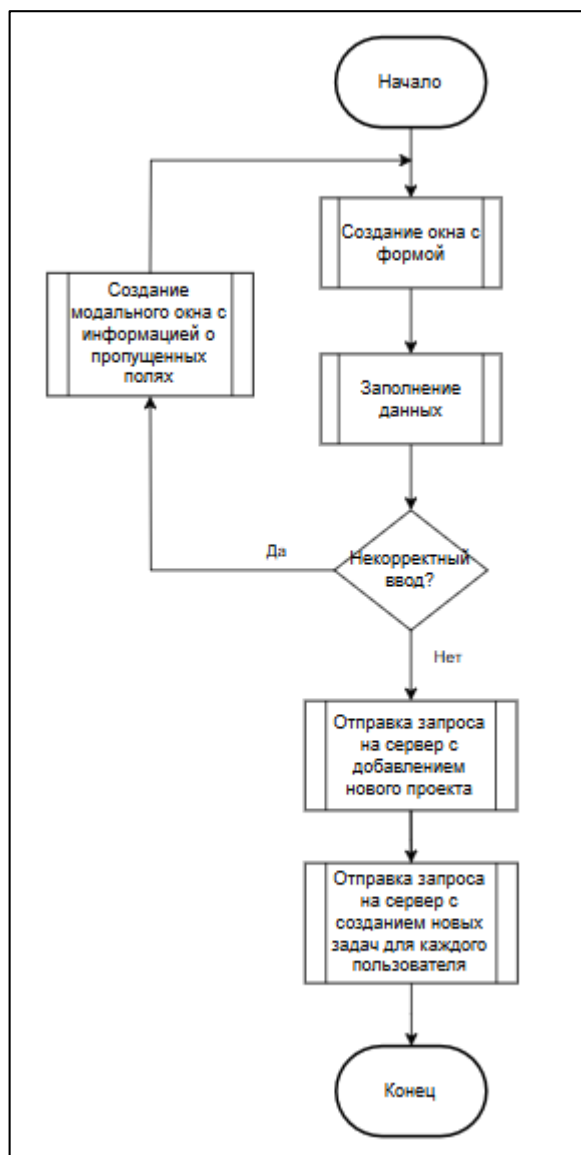


Рисунок 2.1. – Добавление нового пользователя в систему

На рисунке 2.2. показана блок схема алгоритма реализации интерфейса личной карточки пользователя

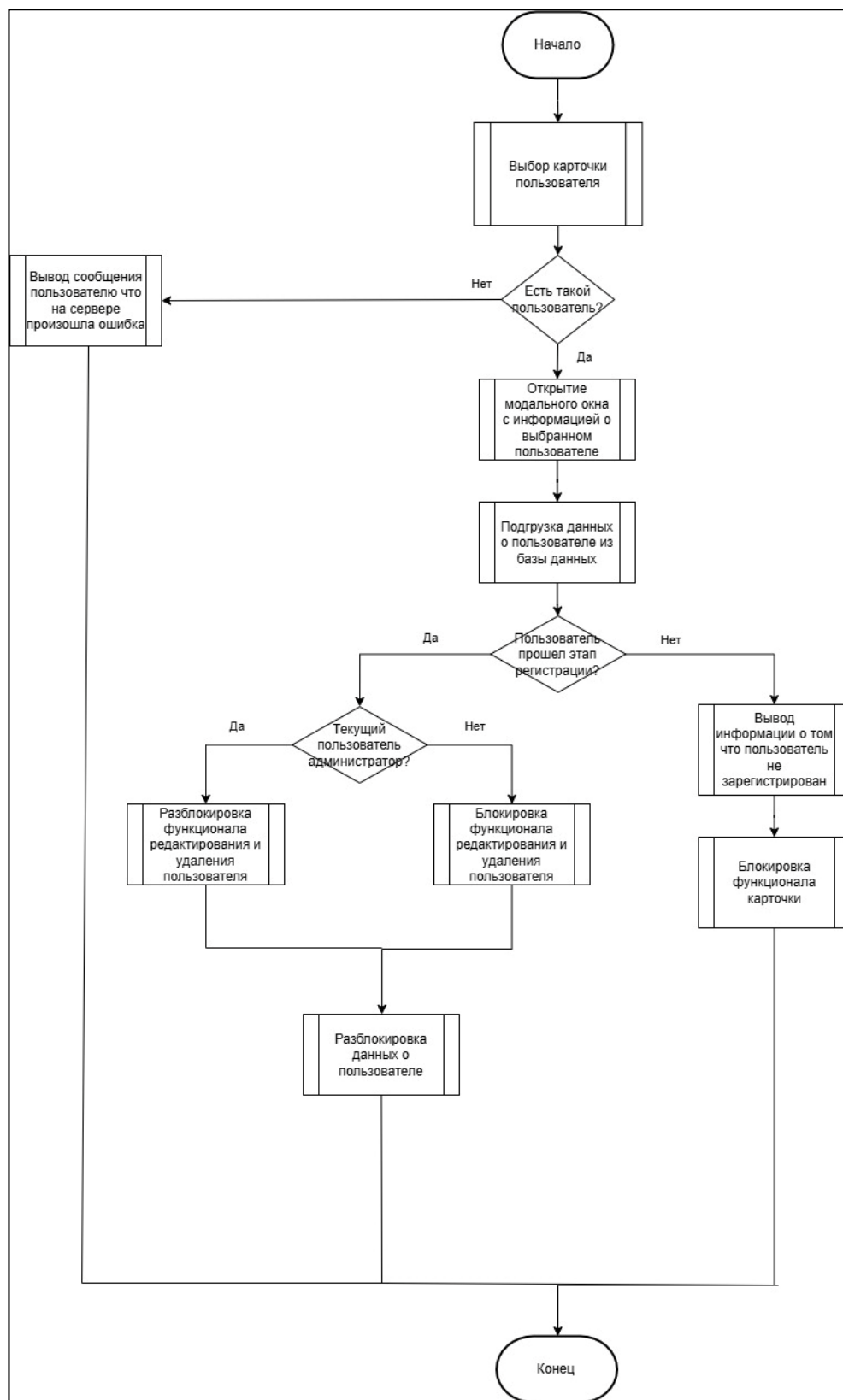


Рисунок 2.2. Блок схема алгоритма реализации интерфейса личной карточки пользователя

На рисунке 2.3. показана блок схема алгоритма реализации вторичной авторизации пользователя.

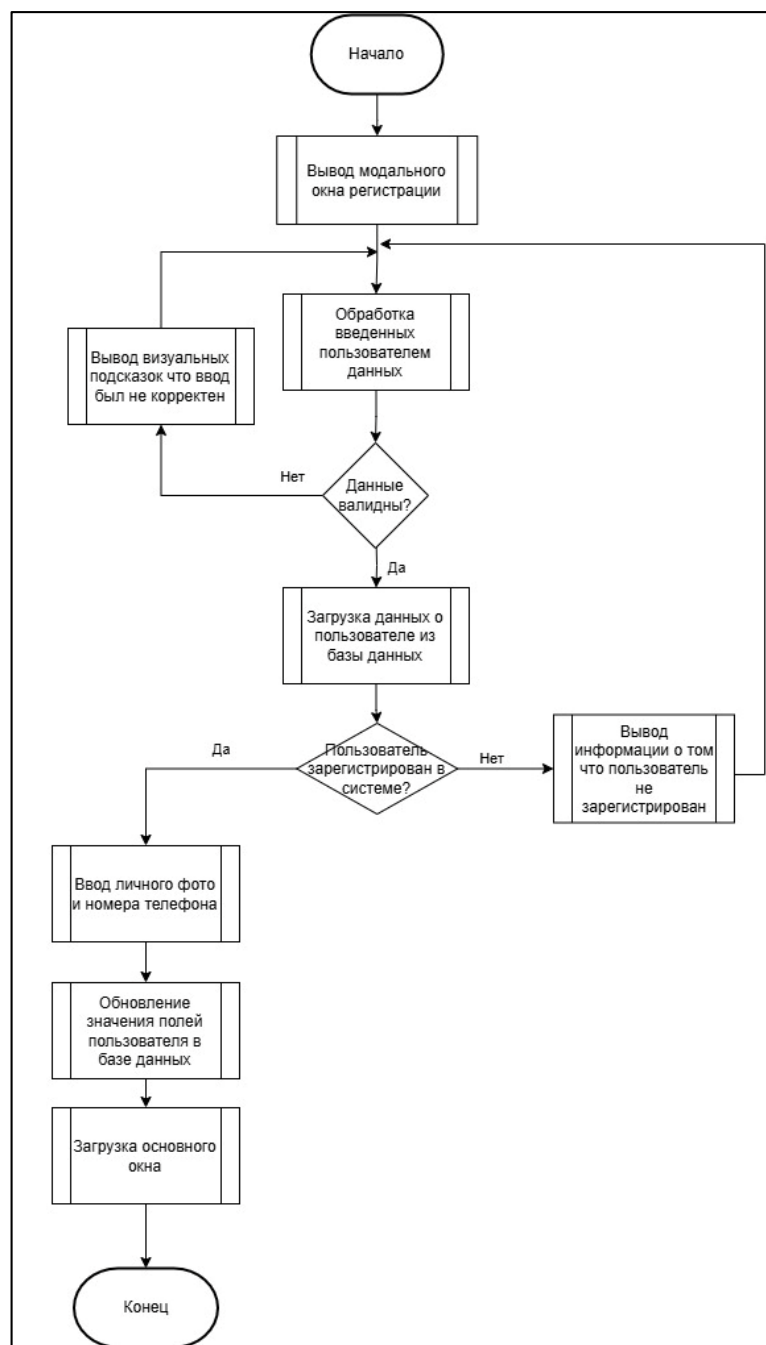


Рисунок 2.3. Блок схема алгоритма реализации вторичной авторизации пользователя

Таким образом создание блок схем намного облегчает дальнейшую реализацию проекта и минимизирует количество рисков и ошибок при проектировании программного обеспечения.

2.4 Проектирование пользовательского интерфейса

JavaFX представляет собой современный фреймворк для разработки графических пользовательских интерфейсов, который идеально подходит для

реализации системы управления персоналом (HRM). Он обеспечивает широкий набор инструментов для создания интуитивно понятного, удобного и визуально привлекательного интерфейса, что особенно важно для приложений, требующих работы с большими объемами данных и сложной логикой взаимодействия.

Одним из ключевых преимуществ JavaFX является возможность построения интерфейса с использованием FXML, что позволяет отделить логику программы от визуального представления. Это упрощает процесс разработки и сопровождения приложения, делая код более структурированным и читаемым. Например, в проекте HRM FXML используется для создания шаблонов интерфейса, таких как карточки сотрудников, формы добавления и редактирования данных. Разделение логики и представления также позволяет разработчикам сосредоточиться на функциональной части программы, не отвлекаясь на детали оформления.

Поддержка CSS в JavaFX предоставляет мощные средства для стилизации элементов интерфейса. Благодаря этому можно добиться современного и привлекательного дизайна без необходимости вносить изменения в основной код. В контексте HRM-приложения это позволило создать стилизованные таблицы, кнопки и модальные окна, улучшая визуальное восприятие и удобство работы для конечных пользователей.

Кроме того, JavaFX предлагает широкие возможности для работы с графикой и анимацией. Это позволяет создавать не только статические элементы интерфейса, но и добавлять интерактивные элементы, такие как всплывающие подсказки, анимированные переходы между экранами и визуализацию данных. Такие элементы повышают удобство использования системы и делают её более интуитивно понятной для пользователя.

Еще одной важной особенностью JavaFX является поддержка мультимедиа. Это может быть полезно для HRM-приложения, например, для интеграции изображений сотрудников или внедрения видеоматериалов, связанных с обучением персонала. Такая функциональность делает систему более многофункциональной и позволяет реализовать в ней новые сценарии использования.

Гибкость JavaFX позволяет легко масштабировать приложение и добавлять новые функции. Например, можно реализовать разные типы интерфейсов для различных ролей пользователей: менеджеров, сотрудников отдела кадров или рядовых работников. Это упрощает адаптацию системы под конкретные требования и делает её использование более удобным.

Встроенная поддержка обработки событий в JavaFX делает возможной реализацию сложной логики взаимодействия. Например, в HRM-приложении реализованы модальные окна, которые запрашивают подтверждение действий пользователя (например, удаление данных), а также обработка событий, таких как выбор записей в таблице или нажатие кнопок. Такая функциональность повышает удобство и безопасность работы с приложением.

JavaFX также хорошо интегрируется с различными библиотеками и инструментами, что делает его универсальным выбором для разработки настольных приложений. Благодаря этому HRM-приложение, разработанное с использованием JavaFX, может быть легко расширено или доработано в будущем.

Таким образом, использование JavaFX в проекте по разработке HRM-системы оправдано его широкими возможностями для создания качественного пользовательского интерфейса. Гибкость, мощность и удобство этого фреймворка делают его отличным выбором для реализации всех необходимых функций и обеспечения удобного взаимодействия пользователя с системой.

2.5 Обоснование выбора компонентов и технологий для реализации программного средства

Для разработки многопоточного оконного приложения для автоматизации работы IT-предприятия были выбраны компоненты и технологии, оптимальные для решения поставленных задач.

Клиент-серверное взаимодействие реализовано с помощью сокетов, которые позволяют установить устойчивое двустороннее соединение между клиентом и сервером, что особенно полезно для приложений, в которых важна оперативная передача данных.

Клиентская часть системы была реализована с использованием JavaFX, что обеспечило создание удобного и интуитивно понятного интерфейса для конечного пользователя. JavaFX предоставляет широкий набор инструментов для разработки современных пользовательских интерфейсов. В данном проекте с помощью JavaFX удалось реализовать полноценное клиентское приложение с различными экранами и компонентами, что улучшает восприятие системы пользователем и способствует более легкому освоению интерфейса.

Одним из ключевых преимуществ JavaFX является возможность построения интерфейса с использованием FXML, что позволяет отделить логику программы от визуального представления. Это упрощает процесс разработки и сопровождения приложения, делая код более структурированным и читаемым. В проекте используется FXML для создания шаблонов интерфейса, таких как экраны выбора сеансов, формы бронирования билетов и модальные окна для подтверждения действий.

Поддержка CSS в JavaFX предоставляет мощные средства для стилизации элементов интерфейса. Благодаря этому можно добиться современного и привлекательного дизайна без необходимости вносить изменения в основной код. Это позволило создать стилизованные таблицы расписания сеансов, кнопки управления и всплывающие подсказки, что улучшает визуальное восприятие и удобство работы для конечных пользователей. Важным аспектом системы является обеспечение безопасности данных, для чего была реализована функция шифрования

паролей перед их сохранением в базе данных. Такой подход повышает уровень безопасности пользовательских данных, предотвращая возможность их кражи даже при несанкционированном доступе к базе. Шифрование паролей является важным шагом для защиты системы от потенциальных угроз и является необходимым для соблюдения стандартов информационной безопасности.

Как база данных для проекта была выбрана база данных MySQL.

MySQL является одним из наиболее популярных выборов для реляционных баз данных благодаря своим преимуществам, которые делают его подходящим для разработки приложений, включая автоматизацию процессов кинотеатра.

Во-первых, MySQL обладает высокой производительностью и надежностью, что особенно важно для обработки больших объемов данных, таких как бронирования билетов, управление сеансами, учёт пользователей и статистика. Оптимизированные алгоритмы работы с запросами позволяют системе эффективно обрабатывать запросы даже при высокой нагрузке.

MySQL отличается простотой настройки и использования. Интуитивно понятный интерфейс управления и широкая поддержка в виде документации и сообществ разработчиков делают его удобным для разработчиков любого уровня опыта. Это позволяет быстрее начать работу с базой данных и сосредоточиться на разработке функциональности приложения.

Еще одно преимущество MySQL — это масштабируемость. Он поддерживает обработку данных как для небольших приложений, так и для сложных систем с тысячами пользователей. Благодаря этому, база данных может легко адаптироваться под растущие требования проекта.

Поддержка транзакций и обеспечение целостности данных через механизмы ACID гарантируют, что все операции с данными будут выполнены корректно. Это особенно важно для приложений, где ошибки в обработке данных могут привести к неудобствам для пользователей или финансовым потерям. В случае кинотеатра это может касаться таких операций, как бронирование билетов или расчёты.

MySQL также обеспечивает высокую совместимость с различными языками программирования и фреймворками, включая Java. Это упрощает интеграцию базы данных в проект и взаимодействие с клиентской частью приложения, что делает MySQL отличным выбором для использования в многопоточных Java-приложениях.

Кроме того, MySQL является бесплатной и открытой базой данных, что делает её особенно привлекательной для образовательных и стартап-проектов. При этом она также доступна в коммерческих версиях, предлагающих дополнительные функции и профессиональную поддержку.

Сборка и управление зависимостями осуществлялось с помощью Maven. Использование IntelliJ IDEA в качестве IDE обусловлено её мощными инструментами для разработки на Java, включая автоматизацию сборки, поддержку отладки и интеграцию с популярными библиотеками и

фреймворками, такими как JavaFX и Hibernate. Кроме того, IntelliJ IDEA обеспечивает удобную работу с базами данных, позволяет эффективно управлять зависимостями через Maven и предоставляет интеллектуальные подсказки, что значительно ускоряет процесс разработки.

Диаграмма компонентов в UML представляет физическую архитектуру программной системы, показывая основные компоненты и их взаимодействия. Компоненты, изображаемые в виде прямоугольников с иконками документа, представляют собой независимые модули, выполняющие определенные функции. Зависимости между компонентами изображаются пунктирными стрелками, указывающими на связь и взаимодействие между ними. Эта диаграмма помогает разработчикам и архитекторам визуализировать модульную структуру системы, определять границы и интерфейсы между модулями, а также планировать их развертывание и интеграцию.

Диаграмма компонентов автоматизированной системы представлена на рисунке 2.5.

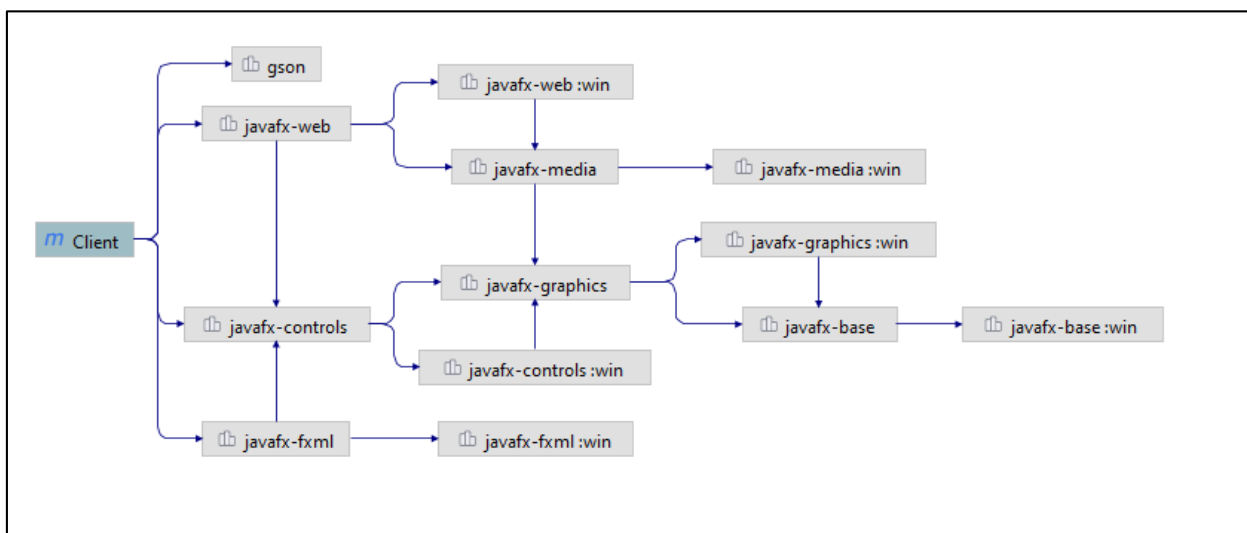


Рисунок 2.5 – Диаграмма компонентов клиентской стороны

На данной иллюстрации показаны зависимости и компоненты, которые требуются для клиентской стороны системы. Ниже представлено описание каждого этого компонента:

- `javafx.controls` – этот модуль предоставляет основные UI-компоненты для разработки графических приложений на JavaFX, такие как кнопки, таблицы, текстовые поля и другие элементы управления;

- `javafx.fxml` – модуль, поддерживающий использование FXML — XML-формата для описания интерфейсов JavaFX. Обеспечивает возможность загрузки FXML-файлов и их связывание с Java-кодом;

- `javafx-web` – Модуль `javafx-web` является частью JavaFX, который предоставляет поддержку для встраивания веб-контента в Java-приложения, позволяя отображать веб-страницы внутри JavaFX-приложений. Он

используется в основном для интеграции веб-технологий, таких как HTML, CSS и JavaScript, в десктопные приложения;

- gson – это библиотека Java, которая может использоваться для преобразования объектов Java в их JSON-представление. Она также может использоваться для преобразования строки JSON в эквивалентный объект Java. Gson может работать с произвольными объектами Java, включая уже существующие объекты, для которых у вас нет исходного кода;

- javafx.graphics – Определяет основные API графа сцены для инструментария пользовательского интерфейса JavaFX (такие как контейнеры компоновки, жизненный цикл приложения, фигуры, преобразования, холст, ввод, рисование, обработка изображений и эффекты), а также API для анимации, CSS, параллелизма, геометрии, печати и оконного взаимодействия;

- javafx.media – Определяет API для воспроизведения медиа- и аудио контента в рамках инструментария пользовательского интерфейса JavaFX, включая MediaView и MediaPlayer;

- javafx.base – Определяет базовые API для инструментария пользовательского интерфейса JavaFX, включая API для привязок, свойств, коллекций и событий.

В заключение, использование таких технологий и инструментов, как JavaFX, MySQL, Maven и IntelliJ IDEA, позволяет создать высокопроизводительное, безопасное и удобное приложение для автоматизации процессов в IT-предприятии.

3 ТЕСТИРОВАНИЕ И ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММНОГО СРЕДСТВА

Разработанное GUI-приложение включает в себя валидацию форм и полей, которая не позволяет пользователю ввести некорректные данные и записать их в базу данных.

При попытке авторизации у пользователя могут возникнуть следующие ошибки: ошибка сервера (клиент пытается зайти, но сервер не работает), ошибка ввода данных авторизации (пустые данные, некорректные данные, слишком короткий пароль, слишком длинный пароль, некорректный пароль но корректный логин и т.д.)

Данные ошибки изображены на рисунках 3.1-3.3.

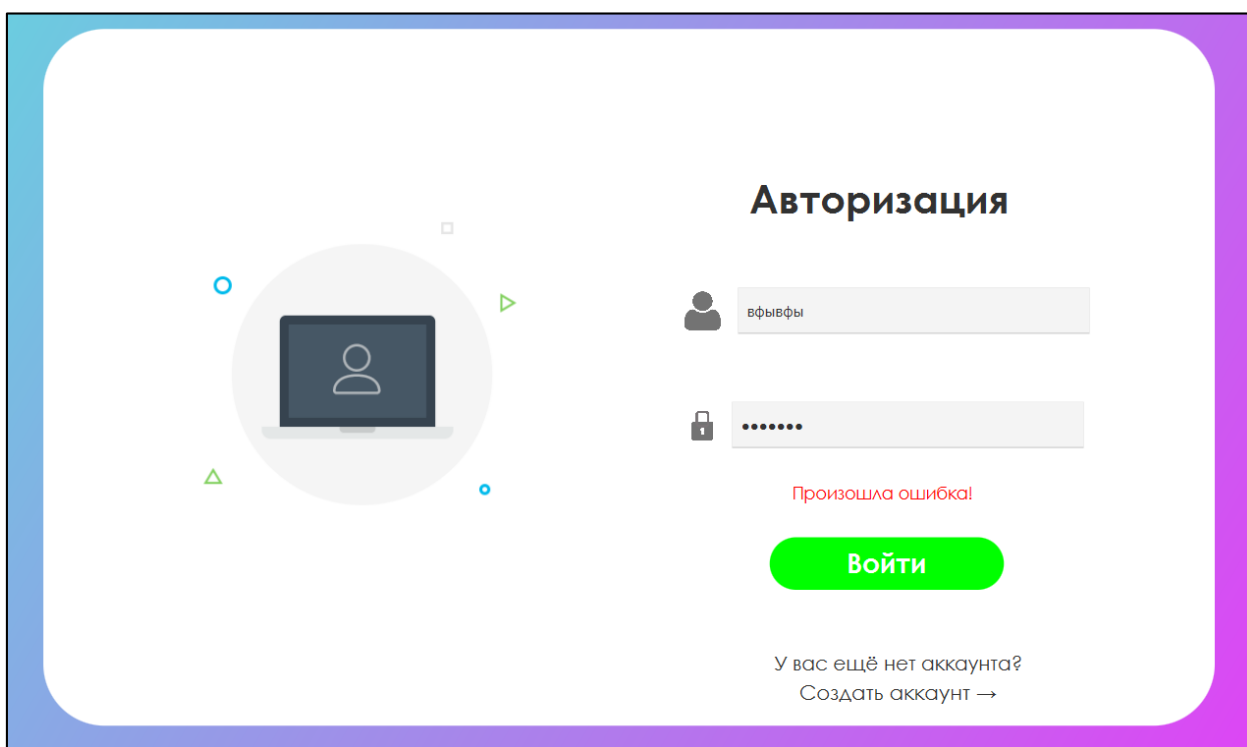


Рисунок 3.1 – Ошибка сервера

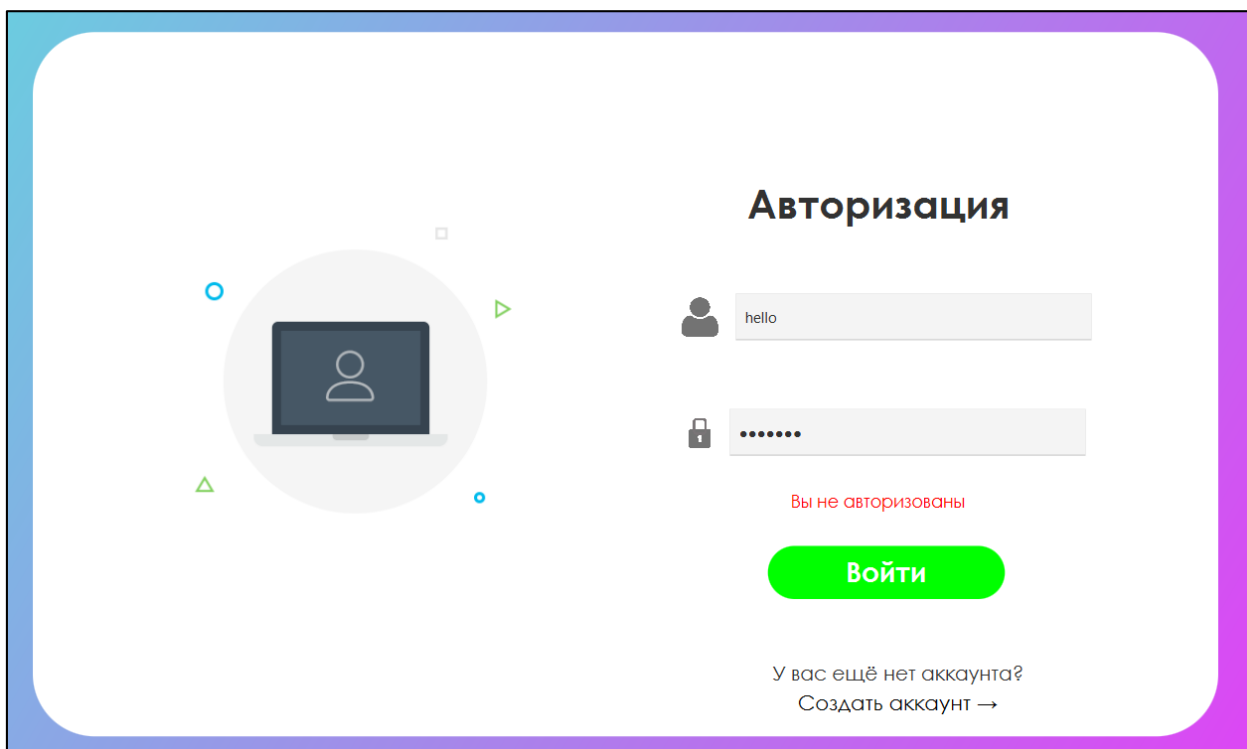


Рисунок 3.2. – Пользователь не авторизован

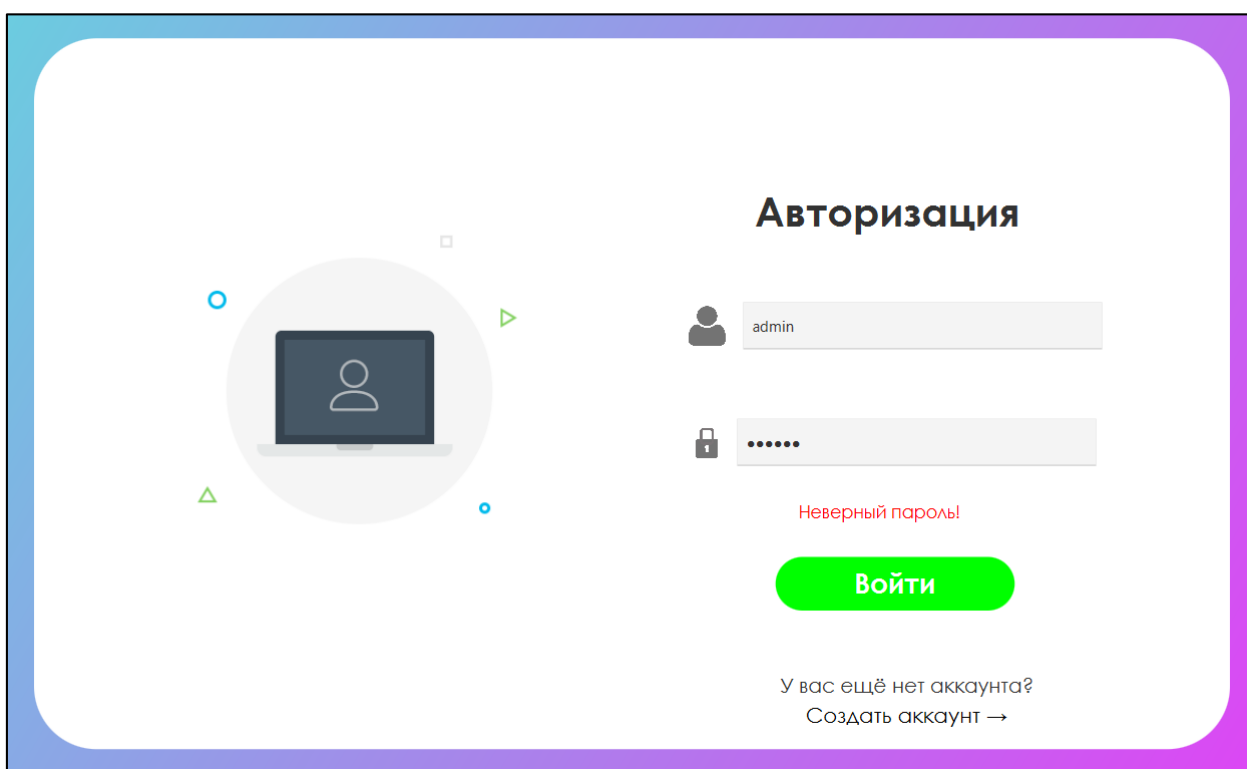


Рисунок 3.3. – Пользователь ввел некорректный пароль

При попытке регистрации введена валидация на корректный ввод значений в поля (с помощью regex), а также учитывая то, что в программе

реализована двухступенчатая регистрация, то введена серверная проверка на существование пользователя в базе данных.

Данные ошибки изображены на рисунке 3.4 – 3.6

The image shows a web registration form titled "Форма регистрации". It contains several input fields with validation feedback icons (red 'X' for error, green checkmark for success):

- Логин**: Input field contains "авывавы", followed by a red 'X' icon.
- Пароль**: Input field contains ".....", followed by a red 'X' icon.
- Имя**: Input field contains "авыавывавы", followed by a green checkmark icon.
- Фамилия**: Input field contains "выавывавы", followed by a green checkmark icon.
- Почта**: Input field contains "авывавыа", followed by a red 'X' icon.
- Возраст**: Input field contains "выавывавы", followed by a red 'X' icon.
- Страна**: Input field contains "авыавывавыа", followed by a green checkmark icon.
- Город**: Input field contains "авыавыа", followed by a green checkmark icon.

At the bottom of the form is a large green button with the text "ПОДТВЕРДИТЬ".

Рисунок 3.4. – Проверка ввода пользователя с использованием regex

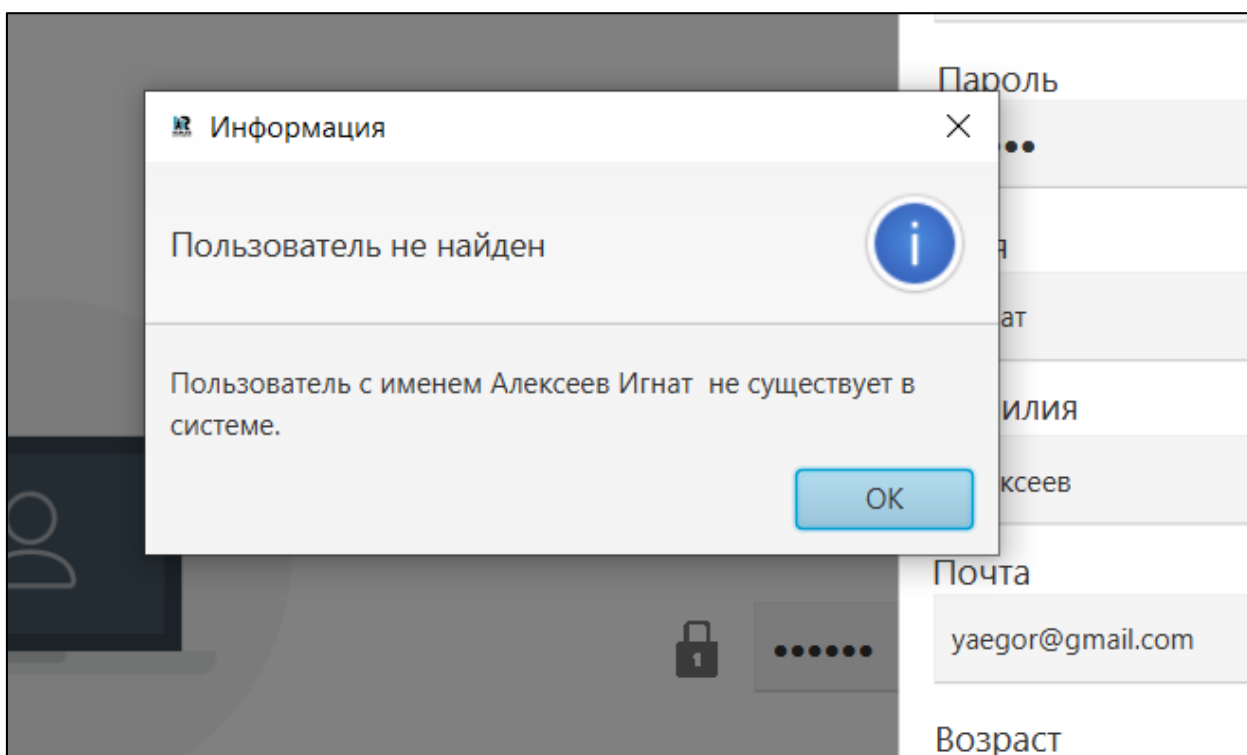


Рисунок 3.5. – Проверка ввода пользователя, который был ранее авторизован