**Министерство общего и профессионального образования  
Ростовской области  
государственное бюджетное профессиональное образовательное учреждение**

**Ростовской области**

**«Волгодонский техникум информационных технологий, бизнеса и дизайна**

**имени В.В. Самарского»**

|  |  |
| --- | --- |
|  | **Утверждаю:** |
|  | И.о. зам. директора по УР |
|  | Н.В. Погорелова |
|  | 2024 г. |

**Дипломный проект**

**На тему** Разработка веб-приложения корпоративного управления проектами для ИП «Ромашов Софт»

**Специальность** 09.02.07 Информационные системы и программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студентка |  |  |  | К.Ю. Еременко |
|  |  |  |
|  |  | (подпись) |  |  |
| Руководитель проекта |  |  |  | Р.В. Ромашов |
|  |  |  |
|  |  | (подпись) |  |  |
| Консультант по экономической части |  |  |  | Е.А. Галицына |
|  |  | (подпись) |  |  |
| Нормоконтроль |  |  |  | И.Н. Власенко |
|  |  |
|  |  | (подпись) |  |  |

|  |  |
| --- | --- |
| Защищен с оценкой |  |
| Протокол № |  |
| от 2024 г. |  |

Волгодонск

2024

**Министерство общего и профессионального образования  
Ростовской области  
государственное бюджетное профессиональное образовательное учреждение**

**Ростовской области**

**«Волгодонский техникум информационных технологий, бизнеса и дизайна**

**имени В.В. Самарского»**

|  |  |
| --- | --- |
| **Одобрено:** | **Утверждаю:** |
| цикловой комиссией профессионального  информационного цикла | И.о. зам. директора по УР |
| Протокол № 6 от «15» января 2024 г. | \_\_\_\_\_\_\_\_ / Н.В. Погорелова / |
|  |  |
| Председатель ЦК \_\_\_\_\_\_\_\_ / И.Н. Борзенкова / |  |

**Задание  
на выполнение выпускной квалификационной работы**

|  |  |  |
| --- | --- | --- |
| Студентки Еременко Кристина Юрьевна | | |
|  |  | |
| Специальность 09.02.07 Информационные системы и программирование | | |
|  |  | |
| 1. Тема: Разработка веб-приложения корпоративного управления проектами для ИП «Ромашов Софт»  утверждена приказом по техникуму №147 (ИСП-19)/148 (ИСП-19К) от «8» февраля 2024 г. | | |
|  |  | |
| 2. Срок сдачи законченной работы: 10 июня 2024 г. | | |
|  | | |
| 3. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов): | | |
| * анализ предметной области; * разработка технического задания; * проектирование программного продукта; * разработка программного продукта. | | |
|  | | |
| 4. Перечень графического материала (с точным указанием обязательных чертежей) | | |
|  | | |
|  | | |
| 5. Руководитель: | | преподаватель, Ромашов Р.В. |
|  | | (должность, фамилия, инициалы, подпись) |

|  |  |  |
| --- | --- | --- |
| Задание получила «1» марта 2024 г. |  | / К.Ю. Еременко / |
|  | (подпись) |  |

|  |  |
| --- | --- |
| **Одобрено:** | **Утверждаю:** |
| цикловой комиссией профессионального  информационного цикла | И.о. зам. директора по УР |
| Протокол № 6 от «15» января 2024 г. | \_\_\_\_\_\_\_\_ / Н.В. Погорелова / |
|  |  |
| Председатель ЦК \_\_\_\_\_\_\_\_ / И.Н. Борзенкова / |  |

**Календарный план выполнения  
выпускной квалификационной работы**

СтуденткиЕременко Кристина Юрьевна

Специальность 09.02.07 Информационные системы и программирование

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование мероприятия** | **Срок выполнения** | **Отметка о выполнении** |
| 1 | Получение задания на ВКР | 01.03.2024 |  |
| 2 | Подбор и проведение анализа источников специальной литературы по теме работы | 15.03.2024 |  |
| 3 | Подбор литературы и материалов о деятельности организации (предприятия) | 22.03.2024 |  |
| 4 | Выполнение исследования по теме ВКР | 29.03.2024 |  |
| 5 | Литературное изложение разделов. Работа над введением: актуальность, цель, задачи и пр. | 05.04.2024 |  |
| 6 | Первый просмотр руководителем ВКР | 08.04.2024 |  |
| 7 | Работа над теоретической главой | 13.04.2024 |  |
| 8 | Второй просмотр руководителем ВКР | 20.04.2024 |  |
| 9 | Работа над аналитической главой | 27.04.2024 |  |
| 10 | Работа над практической главой | 29.04.2024 |  |
| 11 | Описание практической значимости работы. Предложения по внедрению мероприятий | 04.05.2024 |  |
| 12 | Третий просмотр руководителем ВКР | 06.05.2024 |  |
| 13 | Форматирование работы в соответствии с требованиями нормоконтроля | 11.05.2024 |  |
| 14 | Формулировка выводов. Заключение. Оформление списка литературы | 13.05.2024 |  |
| 15 | Форматирование работы в соответствии с требованиями нормоконтроля | 16.05.2024 |  |
| 16 | Четвертый просмотр руководителем ВКР | 20.05.2024 |  |
| 17 | Техническое оформление работы | 27.05.2024 |  |
| 18 | Представление работы с отзывом и рецензией | 01.06.2024 |  |
| 19 | Подготовка защитного слова, оформление раздаточного материала для комиссии, презентации ВКР | 10.06.2024 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель |  | / Р.В. Ромашов / | «1» марта 2024 г. |
|  | (подпись) |  |  |
|  |  |  |  |
| Студентка |  | / К.Ю. Еременко / | «1» марта 2024 г. |
|  | (подпись) |  |  |

**Министерство общего и профессионального образования  
Ростовской области  
государственное бюджетное профессиональное образовательное учреждение**

**Ростовской области**

**«Волгодонский техникум информационных технологий, бизнеса и дизайна**

**имени В.В. Самарского»**

**Отзыв  
на дипломный проект**

**Студентки** *Еременко Кристина Юрьевна*

**Тема дипломного проекта** *Разработка веб-приложения корпоративного управления проектами для ИП «Ромашов Софт»*

**1. Актуальность темы.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.*

**2. Оценка содержания дипломного проекта.** *Vel orci porta non pulvinar neque laoreet suspendisse interdum. Sagittis orci a scelerisque purus semper eget duis at tellus. Sit amet cursus sit amet.*

**3. Качество теоретического и расчетного обоснования принятых в дипломном проекте решений (положительные стороны работы, замечания и недостатки).** *Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Semper viverra nam libero justo laoreet sit. Senectus et netus et malesuada fames ac turpis egestas sed.*

**4. Практическая значимость дипломного проекта.** *Vel orci porta non pulvinar neque laoreet suspendisse interdum.*

**5. Рекомендации по внедрению результатов дипломного проекта.** *Vel orci porta non pulvinar neque laoreet suspendisse interdum.*

**6. Заключение.** *Данный дипломный проект заслуживает оценки «Отлично»*

|  |  |  |
| --- | --- | --- |
| Руководитель |  | / Р.В. Ромашов / |
|  | (подпись) |  |
| С отзывом ознакомлена |  | / К.Ю. Еременко / |
|  | (подпись) |  |
| **«10» июня 2024 года** |  |  |

**Министерство общего и профессионального образования  
Ростовской области  
государственное бюджетное профессиональное образовательное учреждение**

**Ростовской области**

**«Волгодонский техникум информационных технологий, бизнеса и дизайна**

**имени В.В. Самарского»**

**Рецензия  
на дипломный проект**

**Студентки** *Еременко Кристина Юрьевна*

**Тема дипломного проекта** *Разработка веб-приложения корпоративного управления проектами для ИП «Ромашов Софт»*

**1. Актуальность, новизна.** *Тема дипломного проекта актуальна в связи с необходимостью взаимодействия контрагентов между собой посредством электронного документооборота.*

**2. Оценка качества выполнения каждой главы дипломного проекта.** *Все разделы дипломного проекта содержат необходимую информацию согласно заданию. Описание разделов подробное, сопровождается рисунками и таблицами.*

**3. Отличительные особенности дипломного проекта.** *Программный продукт позволит упростить работу с электронными документами в существующей системе предприятия.*

**4. Недостатки дипломного проекта.** *Не выявлено.*

**5. Практическое значение дипломного проекта и рекомендации по ее внедрению.** *Программный продукт, разработанный в рамках дипломного проекта, позволит работникам предприятия работать с счет-фактурой в более удобном формате. При определенных доработках может быть внедрен на предприятии.*

**6. Рекомендуемая оценка.** *Данный дипломный проект заслуживает оценки «Хорошо»*

|  |  |  |
| --- | --- | --- |
| Рецензент |  | / Р.В. Ромашов / |
|  | (подпись) |  |
| С рецензией ознакомлена |  | / К.Ю. Еременко / |
|  | (подпись) |  |
| **«10» июня 2024 года** |  |  |

Содержание

[Введение 6](#_Toc167020022)

[1 Технико-экономическая характеристика объекта 7](#_Toc167020023)

[1.1 Общие сведения о предприятии 7](#_Toc167020024)

[1.2 Организационная структура предприятия 7](#_Toc167020025)

[2 Сбор, анализ и формирование требований к информационной системе 9](#_Toc167020026)

[2.1 Цели, назначение и функции программного продукта 10](#_Toc167020027)

[2.2 Требования к функциональности и дизайну программного продукта 11](#_Toc167020028)

[2.3 Системные требования к оборудованию 12](#_Toc167020029)

[3 Проектирование информационной системы 13](#_Toc167020030)

[3.1 Разработка и описание сценариев использования программного продукта 13](#_Toc167020031)

[3.2 Потоки данных в информационной системе 15](#_Toc167020032)

[3.3 Выбор архитектуры разрабатываемого программного продукта 16](#_Toc167020033)

[3.4 Проектирование реляционной базы данных 17](#_Toc167020034)

[3.5 Прототипирование пользовательского интерфейса 22](#_Toc167020035)

[4 Разработка программного обеспечения 31](#_Toc167020036)

[4.1 Обоснование выбора инструментальных средств разработки программного продукта 31](#_Toc167020037)

[4.2 Описание алгоритма программы 36](#_Toc167020038)

[4.3 Описание интерфейса программного продукта 42](#_Toc167020039)

[4.4 Инструкция по эксплуатации программного продукта 51](#_Toc167020040)

[5 Экономическая часть 58](#_Toc167020041)

[6 Охрана труда и техники безопасности 59](#_Toc167020042)

[Заключение 62](#_Toc167020043)

[Список использованных источников 63](#_Toc167020044)

[Приложение А 64](#_Toc167020045)

# Введение

В настоящее время все большую роль играет эффективное корпоративное управление проектами. Стремительное развитие информационных технологий и все более высокие требования бизнеса к оперативности и точности принимаемых решений делают необходимым наличие надежной системы управления проектами.

Основной целью данной работы является создание удобного веб-приложения для управления проектами. Предлагаемая разработка позволит оптимизировать взаимодействие между сотрудниками, улучшить контроль над выполнением задач и снизить вероятность ошибок на различных этапах проекта. В результате введения данной системы управления проектами, ИП «Ромашов Софт» сможет значительно повысить свою конкурентоспособность на рынке.

Актуальность и необходимость новой разработки обусловлены не только конкурентным рынком, но и растущим спросом на качественные и инновационные решения в области управления проектами. Создание собственного веб-приложения позволит ИП «Ромашов Софт» предложить рынку уникальное решение, которое будет отвечать всем требованиям и потребностям клиентов.

Одной из основных предпосылок для разработки данного веб-приложения является существующая потребность в оптимизации и автоматизации процессов управления проектами в компании. Для этого необходимо провести анализ конкурентных проектов на рынке, оценить их функциональность и эффективность, чтобы определить основные требования и преимущества нашего веб-приложения.

Предполагаемый экономический и социальный выигрыш, который может быть получен в результате данной работы, заключается в том, что веб-приложение будет позволять компаниям существенно сократить время и ресурсы, затраченные на управление проектами. Это позволит повысить эффективность работы компаний, сократить издержки и улучшить конкурентоспособность на рынке. Кроме того, разработанное приложение будет способствовать более эффективному использованию ресурсов и повышению качества проектов.

# 1 Технико-экономическая характеристика объекта

?

## 1.1 Общие сведения о предприятии

Предприятие помогает компаниям и частным лицам решать задачи, автоматизируя бизнес-процессы, создавая новые продукты и сервисы.

Виды программного обеспечения, которые разрабатываются на предприятии:

* веб-приложения: сайты, веб-сервисы, мобильные приложения;
* десктопные приложения: программы для Windows, macOS, Linux;
* мобильные приложения: iOS, Android.
* Так же предприятия оказывает услуги:
* доработка и модернизация существующего ПО;
* техническая поддержка ПО;
* консультационные услуги.

При разработке программного обеспечения необходимо выполнить следующие этапы:

* определение целей и задач ПО, выявление функциональных и нефункциональных требований, составление технического задания;
* разработка архитектуры ПО, проектирование интерфейсов пользователя и структуры базы данных;
* написание кода ПО, его тестирование и отладка;
* установка и настройка ПО, обучение пользователей;
* техническая поддержка ПО, исправление ошибок, доработка ПО.

## 1.2 Организационная структура предприятия

Успешная реализация проекта по разработке программного обеспечения требует создания слаженной команды, каждый член которой будет играть определенную роль.

В данном разделе будут рассмотрены оптимальная структура команды, функции каждого ее члена:

Структура команды и функции каждого ее члена:

* руководитель осуществляет общее руководство командой, определяет стратегию развития компании, привлекает клиентов, ведет переговоры и заключает договоры;
* менеджер проекта обеспечивает координацию работы команды, контролирует ход выполнения проекта, отчитывается перед клиентом о статусе проекта;
* разработчики осуществляют разработку ПО в соответствии с требованиями заказчика, проводят тестирование и отладку ПО, готовят документацию;
* дизайнер разрабатывает интерфейсы пользователя, готовит графические материалы;
* тестировщик осуществляет тестирование ПО, выявляет ошибки, готовит отчеты о тестировании;
* администратор сайта занимается мониторингом входящих заявок и управлением контентом веб-сайта.

Структура предприятия представлена на рисунке 1.1.



Рисунок 1.1 – Организационная структура предприятия

# 2 Сбор, анализ и формирование требований к информационной системе

Процесс начинается с установления контакта с заказчиком и другими заинтересованными сторонами. Основные этапы включают:

1. Сбор требований:

* встреча с заказчиком и заинтересованными сторонами для выявления их потребностей и ожиданий;
* получение полного списка функциональных (что система должна делать) и нефункциональных (требования к производительности, безопасности, удобству использования и т.д.) требований.

1. Анализ требований:

* систематизация и классификация требований по приоритетности;
* проверка требований на соответствие бизнес-целям компании;
* выявление и устранение противоречий или недостатков в требованиях.

1. Формирование требований

* создание документа с функциональными и нефункциональными требованиями, описанием пользовательских сценариев и интерфейсов;
* утверждение документа с заказчиком и заинтересованными сторонами.

1. Дополнительные шаги

* проведение обсуждений и презентаций для уточнения требований и получения обратной связи;
* внесение корректировок на основе обратной связи;
* обеспечение полного понимания требований всеми участниками проекта.

Результатом этого процесса является документ, служащий основой для дальнейшей разработки веб-приложения, создания архитектуры, проектирования интерфейса и реализации функциональности, отвечающей потребностям компании в управлении проектами.

## 2.1 Цели, назначение и функции программного продукта

Цели проекта:

* эффективное управление проектами: создание инструмента для эффективного управления проектами с минимальными затратами времени и ресурсов;
* оптимизация процессов: планирование, выполнение и мониторинг проектов для достижения лучших результатов в установленные сроки и бюджет;
* улучшение коммуникации: улучшение обмена информацией между участниками проектов и сокращение времени на передачу информации.

Назначение программного продукта:

* централизованное управление проектами: централизованное управление задачами, сроками и ресурсами;
* планирование и контроль: планирование проектов, установка сроков и бюджетов, контроль выполнения в реальном времени;
* улучшенная командная работа: обеспечит эффективную командную работу, путем предоставления возможности совместного доступа к информации.

Функции программного продукта:

* управление задачами и проектами: создание, назначение и отслеживание выполнения задач и проектов;
* отслеживание прогресса: наблюдение за текущим состоянием задач и проектов;
* доступ к документам: возможность хранить и делиться документами и файлами, необходимыми для проекта.

Удобства использования:

* простота использования: интуитивно понятный интерфейс, который позволит пользователям быстро освоить приложение без необходимости специальной подготовки.

## 2.2 Требования к функциональности и дизайну программного продукта

Простой интерфейс:

* интуитивно понятные кнопки и меню, легко находящиеся на экране и понятные пользователю;
* четкое оформление страниц с удобным размещением элементов управления для быстрого доступа к функциям;
* удобен для работы как новичков, так и опытных пользователей, не требующий дополнительного обучения.

Функциональность:

* создание и назначение задач для проектов с возможностью быстро создавать задачи, устанавливать сроки и назначать ответственных;
* приложение должно позволять пользователю видеть текущий статус выполнения задач и проектов, чтобы он мог оценить степень их завершенности.

Дизайн:

* приятный и современный внешний вид с использованием современных дизайнерских трендов;
* простое оформление страниц с четко разделенными и оформленными элементами управления и информационными блоками;
* адаптивный дизайн для корректного отображения на различных устройствах.

Безопасность:

* приложение должно защищать конфиденциальные данные пользователей и информацию о проектах с помощью шифрования и других мер безопасности;
* приложение должно включать механизмы аутентификации и авторизации пользователей для контроля доступа к своим функциям и защиты от несанкционированного использования.

## 2.3 Системные требования к оборудованию

Соблюдение системных требований к оборудованию имеет ключевое значение, поскольку это гарантирует, что разработчик сможет эффективно работать с инструментами и окружением разработки, обеспечивая удобство и производительность. Основные аспекты, которые следует учитывать в этом контексте:

* процессор: многоядерный процессор с высокой тактовой частотой для плавной работы и быстрой компиляции кода;
* оперативная память: минимум 8 ГБ для комфортной работы, в зависимости от объема и сложности приложения;
* жесткий диск: 500 ГБ свободного места для инструментов разработки, зависимостей и проектов;
* операционная система: поддерживаются Windows, macOS и Linux. Рекомендуется использовать последние версии с обновлениями;
* браузеры: обновленный веб-браузер (Google Chrome, Mozilla Firefox или Microsoft Edge) для разработки и тестирования React-приложений;
* среда разработки (IDE): поддержка TypeScript (Visual Studio Code, WebStorm или Atom);
* Node.js и npm: для управления зависимостями и сборки проекта (Webpack или Parcel);
* Git: система контроля версий для отслеживания изменений и управления исходным кодом проекта.

# 3 Проектирование информационной системы

Проектирование информационной системы – это процесс разработки плана или модели, которая описывает структуру, компоненты и функциональность системы. Этот процесс включает в себя определение требований, создание архитектуры, разработку интерфейсов, проектирование базы данных, обеспечение безопасности и другие аспекты, необходимые для создания эффективной и функциональной информационной системы.

В ходе проектирования информационной системы учитывается специфика бизнес-процессов или задач, которые система должна поддерживать, а также потребности и ожидания пользователей. Целью проектирования является создание системы, которая эффективно решает поставленные задачи, обеспечивает удобство использования, масштабируемость, безопасность и надежность.

Проектирование информационной системы предшествует ее реализации и играет ключевую роль в определении архитектуры и функциональности системы. Успешное проектирование обеспечивает более простое и эффективное развертывание и внедрение системы, а также повышает вероятность успешного достижения поставленных целей и ожиданий пользователей [1].

## 3.1 Разработка и описание сценариев использования программного продукта

Диаграмма прецедентов или диаграмма вариантов использования представляет собой графическое изображение того, как пользователи взаимодействуют с системой, чтобы достичь определенных целей. Она помогает идентифицировать основные функциональные возможности системы и отображает, какие действия могут выполнять пользователи, чтобы достичь своих целей через систему [2].

Использование диаграммы прецедентов помогает команде проекта лучше понять требования пользователей и улучшить процесс проектирования и разработки системы, обеспечивая ее соответствие потребностям и ожиданиям пользователей.

На рисунке 3.1 изображена диаграмма вариантов использования.



Рисунок 3.1 – Диаграмма вариантов использования

На предоставленном изображении показана схема вариантов использования системы управления проектами. Действующими лицами являются пользователь и администратор.

Пользователь представляет собой людей, которые в основном используют систему управления проектами для управления своими задачами. Их взаимодействие с системой сосредоточено на:

* управление задачами: пользователи могут создавать, редактировать и просматривать задачи, связанные с проектами. Они могут помечать задачи как выполненные, а также отслеживать ход их выполнения;
* взаимодействие с проектом: пользователи могут просматривать информацию о проектах. Эта информация включает данные о самом проекте, а также списки задач, которые необходимо выполнить в рамках этого проекта.

Администратор представляет лиц, ответственных за управление всей системой и учетными записями пользователей. Их взаимодействие с системой сосредоточено на:

* управление этапами проекта: администраторы могут контролировать деление проектов на этапы, определяя задачи, сроки выполнения и конечные результаты;
* управление пользователями: администраторы могут создавать, редактировать и удалять пользователей. Они также могут определять роли пользователей, устанавливая соответствующие права доступа и привилегии;
* управление проектами: администраторы осуществляют контроль и управление всеми аспектами проекта. Это включает в себя создание новых проектов, изменение параметров проекта, удаление проектов, а также назначение ответственных лиц.

Представленная схема вариантов использования системы управления проектами демонстрирует взаимодействие пользователей и администраторов с системой. Пользователи могут эффективно управлять задачами, а администраторы обладают полным контролем над системой и учетными записями пользователей, обеспечивая стабильность работы системы.

## 3.2 Потоки данных в информационной системе

Потоки данных в информационной системе представляют собой направленное движение информации или данных от одного компонента или процесса к другому. Они играют важную роль в обмене информацией и обеспечивают функционирование системы. Рассмотрим основные типы потоков данных в информационной системе:

* внешние сущности, такие как пользователи, другие системы и клиенты, взаимодействующие с системой;
* процессы, представляющие собой последовательность действий для достижения цели, включая взаимодействие с внешними сущностями;
* потоки данных, описывающие передачу информации между внешними сущностями, процессами и хранилищем данных;
* хранилище данных, где информация сохраняется и доступна для использования.

Эти компоненты взаимодействуют для обработки, хранения и передачи данных, поддерживая бизнес-процессы и цели организации [3].

На рисунке 3.2 изображена диаграмма потоков данных.

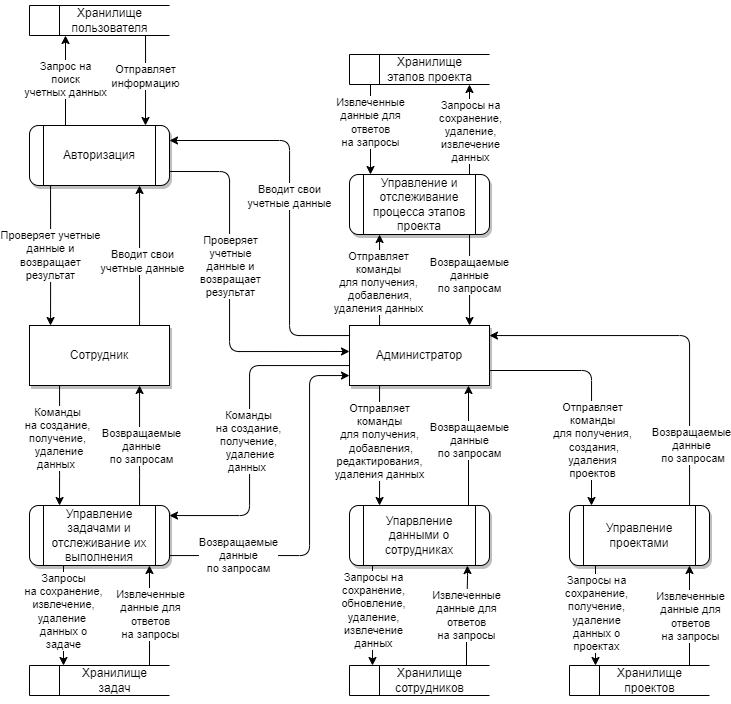


Рисунок 3.2 – Диаграмма потоков данных

На диаграмме изображены внешние сущности «Сотрудник» и «Администратор», которые взаимодействуют с различными модулями системы через отправку команд и получение данных. Каждый модуль системы обменивается данными с соответствующими хранилищами для получения или сохранения необходимой информации. Весь процесс направлен на обеспечение плавного функционирования системы управления проектами, задачами и сотрудниками, с единым контролем со стороны администратора.

Диаграмма потоков данных демонстрирует, как информация течет через систему, какие модули задействованы в процессе, и как данные обрабатываются и хранятся для выполнения различных задач.

## 3.3 Выбор архитектуры разрабатываемого программного продукта

Архитектура программного продукта – это набор ключевых решений и принципов, определяющих структуру, взаимодействие и поведение компонентов программного обеспечения [4].

Архитектура клиент-серверного приложения с модульной монолитной структурой представляет собой подход к разработке программного обеспечения, при котором приложение разделено на клиентскую и серверную части, а сама серверная часть организована как монолитное приложение, состоящее из отдельных модулей.

Клиентская часть представляет собой пользовательский интерфейс, с которым взаимодействует пользователь. Она отвечает за отображение данных, взаимодействие с пользователем и отправку запросов на сервер для получения или обновления информации. Эта часть построена с использованием React, популярной библиотеки для создания динамичных и интерактивных пользовательских интерфейсов.

Серверная часть, организованная как модульный монолит, выполняет все основные вычисления, обработку данных и бизнес-логику. Она состоит из различных модулей, каждый из которых отвечает за конкретную область функциональности. Несмотря на разделение на модули, все они работают в рамках одного процесса и одной кодовой базы, что упрощает управление и развертывание приложения.

Архитектура клиент-серверного приложения с модульной монолитной структурой предоставляет баланс между структурированной организацией кода и простотой управления, подходя для множества сценариев использования. Выбор данной архитектуры должен основываться на потребностях конкретного проекта и его долгосрочных целей.

## 3.4 Проектирование реляционной базы данных

Концептуальная модель данных – это абстрактное представление о данных в информационной системе, которое описывает сущности (или объекты), их атрибуты и взаимосвязи между ними. Она служит основой для проектирования базы данных, представляя собой высокоуровневую картину о данных, которые будут использоваться в системе [5].

На рисунке 3.3 изображена концептуальная модель базы данных.



Рисунок 3.3 – Концептуальная модель базы данных

На схеме показаны ключевые сущности, участвующие в системе управления проектами, и их взаимосвязи. Представленная концептуальная модель данных служит основой для дальнейшего проектирования и реализации базы данных.

Концептуальная модель данных позволяет лучше понять структуру и взаимосвязи данных в системе управления проектами, обеспечивая более эффективное и удобное управление проектами и ресурсами.

Следующим этапом необходимо сформировать логическую модель данных.

Логическая модель данных – это абстракция данных, отображающая их структуру, связи и ограничения. Этот инструмент проектирования баз данных и информационных систем помогает понять структуру данных, их взаимосвязи и учитывает требования к производительности, безопасности и доступности системы [6].

В данной базе данных будут храниться сведения о проектах и сотрудниках. Информация о проектах будет храниться в таблицах 3.1-3.9, а информация о сотрудниках в таблице 3.10-3.13.

Таблица 3.1 – Атрибуты статусов проекта

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Статусы проекта | Идентификационный код  Название |

Таблица 3.2 – Атрибуты этапов

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Этапы | Идентификационный код  Название  Описание |

Таблица 3.3 – Атрибуты статусов задач

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Статусы задач | Идентификационный код  Название |

Таблица 3.4 – Атрибуты проектов

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Проекты | Идентификационный код  Название  Описание  Идентификационный код статусов проекта  Дата начала  Дата завершения  Идентификационный код куратора |

Таблица 3.5 – Атрибуты участников проекта

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Участники проекта | Идентификационный код проекта  Идентификационный код сотрудника |

Таблица 3.6 – Атрибуты этапов проекта

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Этапы проекта | Идентификационный код  Идентификационный код проекта  Идентификационный код этапа  Дата начала  Дата завершения  Срок выполнения |

Таблица 3.7 – Атрибуты статусов задач для этапа проекта

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Статусы задач этапа проекта | Идентификационный код этапа проекта  Идентификационный код статусов задачи |

Таблица 3.8 – Атрибуты задач

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Задачи | Идентификационный код  Название  Описание  Идентификационный код этапа проекта  Идентификационный код статусов задачи  Идентификационный код создателя |

Таблица 3.9 – Атрибуты исполнителей задач

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Исполнители задач | Идентификационный код сотрудника  Идентификационный код задачи  Дата начала  Дата завершения |

Таблица 3.10 – Атрибуты статусов сотрудника

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Статусы сотрудника | Идентификационный код  Название |

Таблица 3.11 – Атрибуты должностей

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Должности | Идентификационный код  Название |

Таблица 3.12 – Атрибуты подразделений

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Подразделения | Идентификационный код  Код  Название |

Таблица 3.13 – Атрибуты сотрудников

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Сотрудники | Идентификационный код  Фамилия  Имя  Отчество  Пол  Табельный номер  Идентификационный код должности  Идентификационный код статуса сотрудника  Идентификационный код подразделения |

На рисунках 3.4-3.5 представлена схема базы данных.



Рисунок 3.4 – Схема базы данных

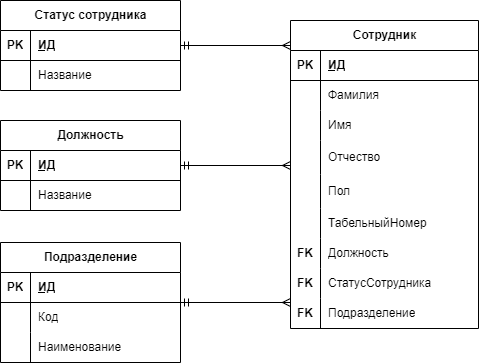


Рисунок 3.5 – Схема базы данных

Следующим этапом необходимо сформировать физическую модель данных.

Физическая модель представляет собой конкретное описание структуры данных в базе данных, включая таблицы, столбцы, типы данных, ограничения и связи между таблицами. Она является реализацией логической модели данных, которая определяет сущности, их атрибуты и отношения между ними [7].

На рисунке 3.4 представлена физическая модель.

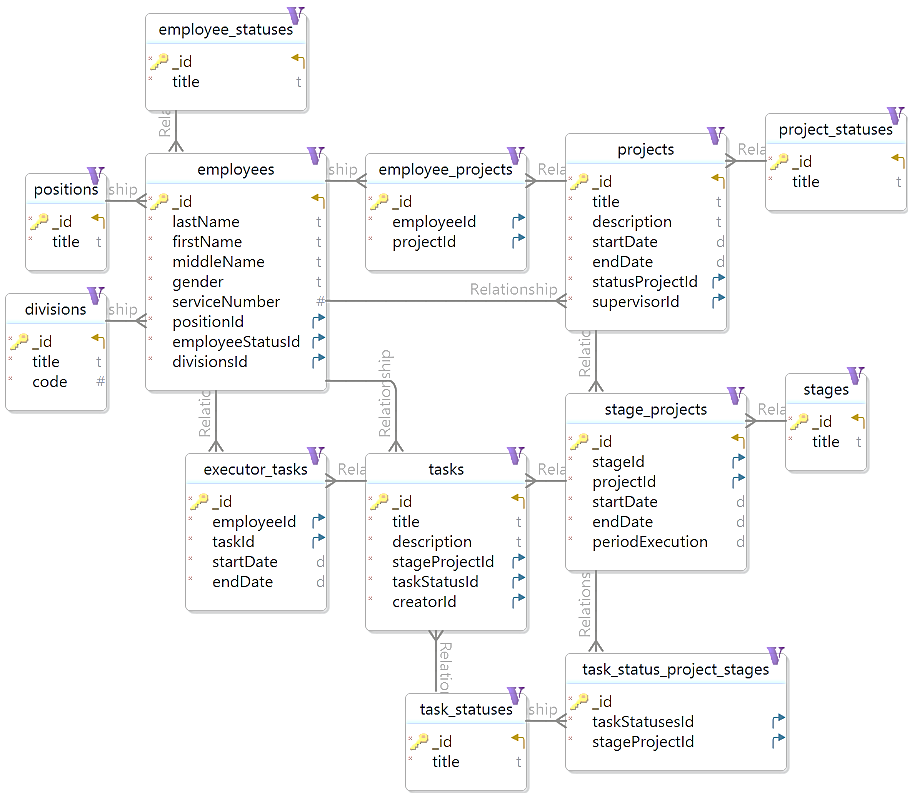


Рисунок 3.4 – Физическая модель

## 3.5 Прототипирование пользовательского интерфейса

Прототипирование представляет собой набор черновых макетов или схем, которые используются в процессе разработки веб-сайтов или приложений для визуализации структуры и расположения элементов интерфейса. Эти макеты содержат только базовые элементы, такие как блоки контента, кнопки, поля ввода, навигационные панели и т.д., и не включают в себя дизайн и декоративные элементы [8].

На рисунке 3.5 изображен прототип авторизации.

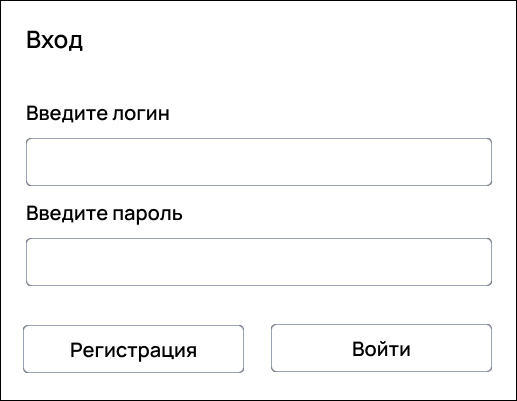


Рисунок 3.5 – Прототип авторизации

Прототип авторизации позволяет пользователям входить в систему.

На рисунке 3.6 изображен прототип регистрации.

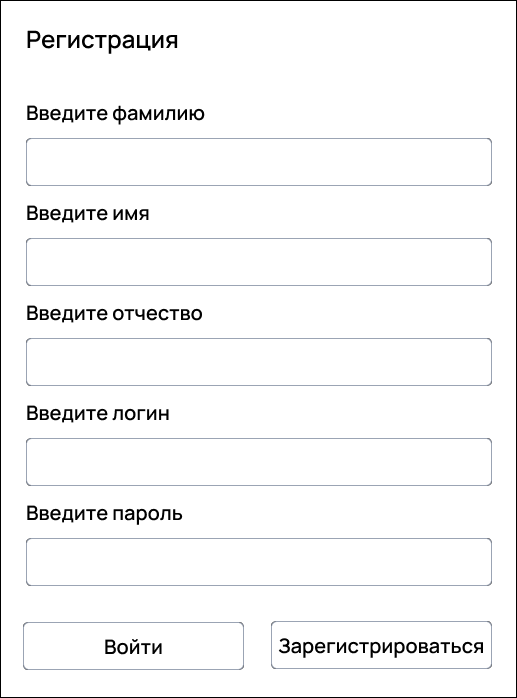


Рисунок 3.6 – Прототип регистрации

Прототип регистрации предоставляет простой и удобный способ создания учетной записи в системе.

На рисунке 3.7 изображен прототип страницы сотрудников.

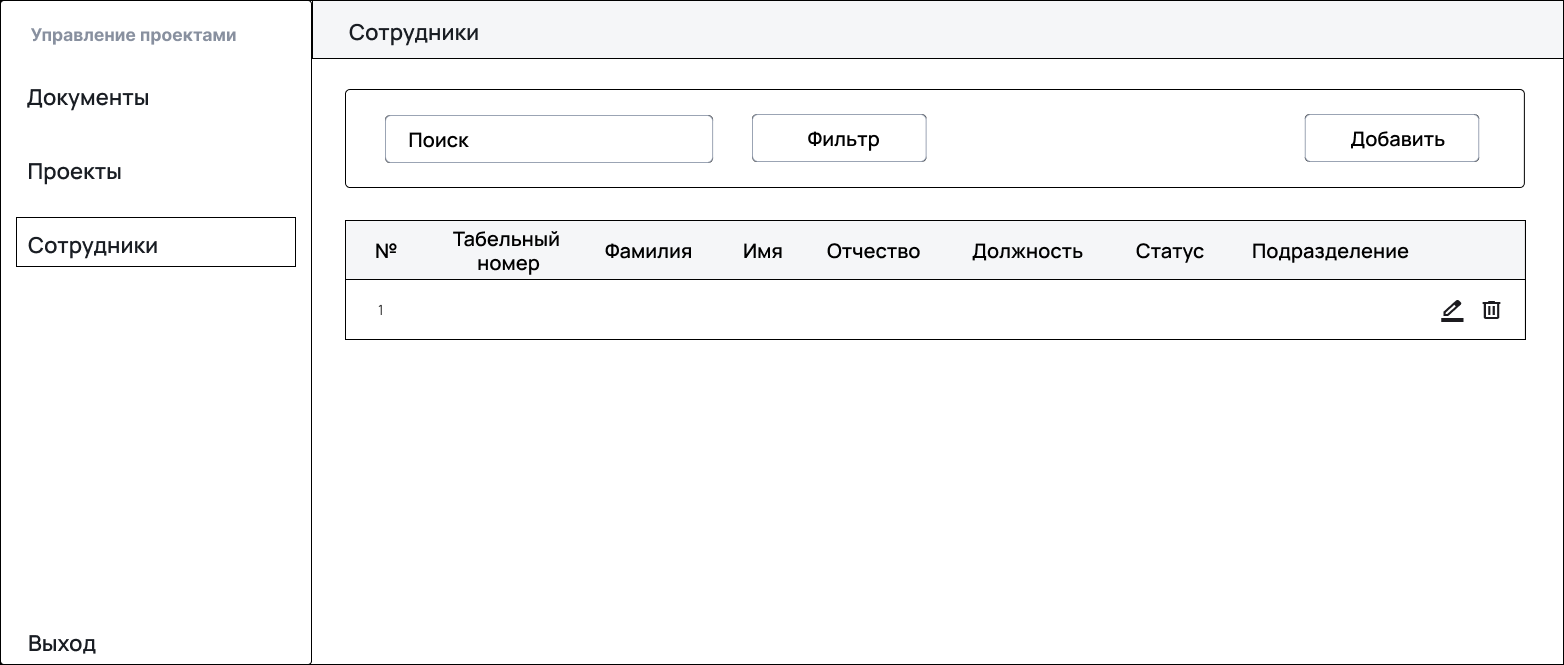


Рисунок 3.7 – Прототип страницы сотрудников

Прототип страницы сотрудников – это веб-страница для управления информацией о сотрудниках. В меню слева пользователи могут переходить к разделам, таким как документы, проекты и сотрудники. Основной раздел содержит таблицу с данными о сотрудниках: табельный номер, ФИО, должность, статус, подразделение. Справа есть кнопки для добавления, редактирования и удаления сотрудников.

На рисунке 3.8 изображен прототип формы добавления сотрудника.

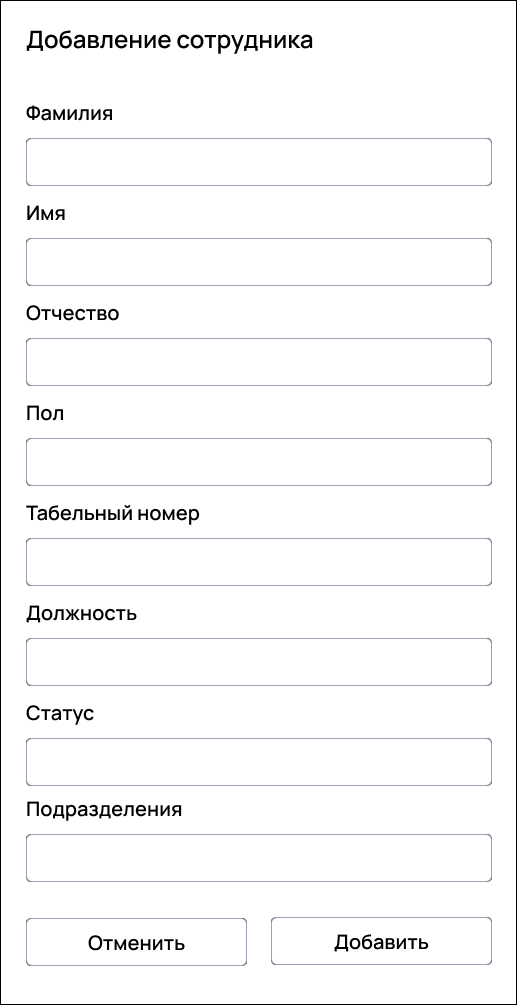


Рисунок 3.8 – Прототип формы добавления сотрудника

На представленном изображении показан прототип формы добавления сотрудника – веб-интерфейса, предназначенного для сбора и хранения информации о новых сотрудниках. Он представляет собой структурированный макет с четкими полями и интуитивно понятными элементами управления, что способствует эффективному вводу данных.

На рисунке 3.9 изображен прототип страницы проектов.

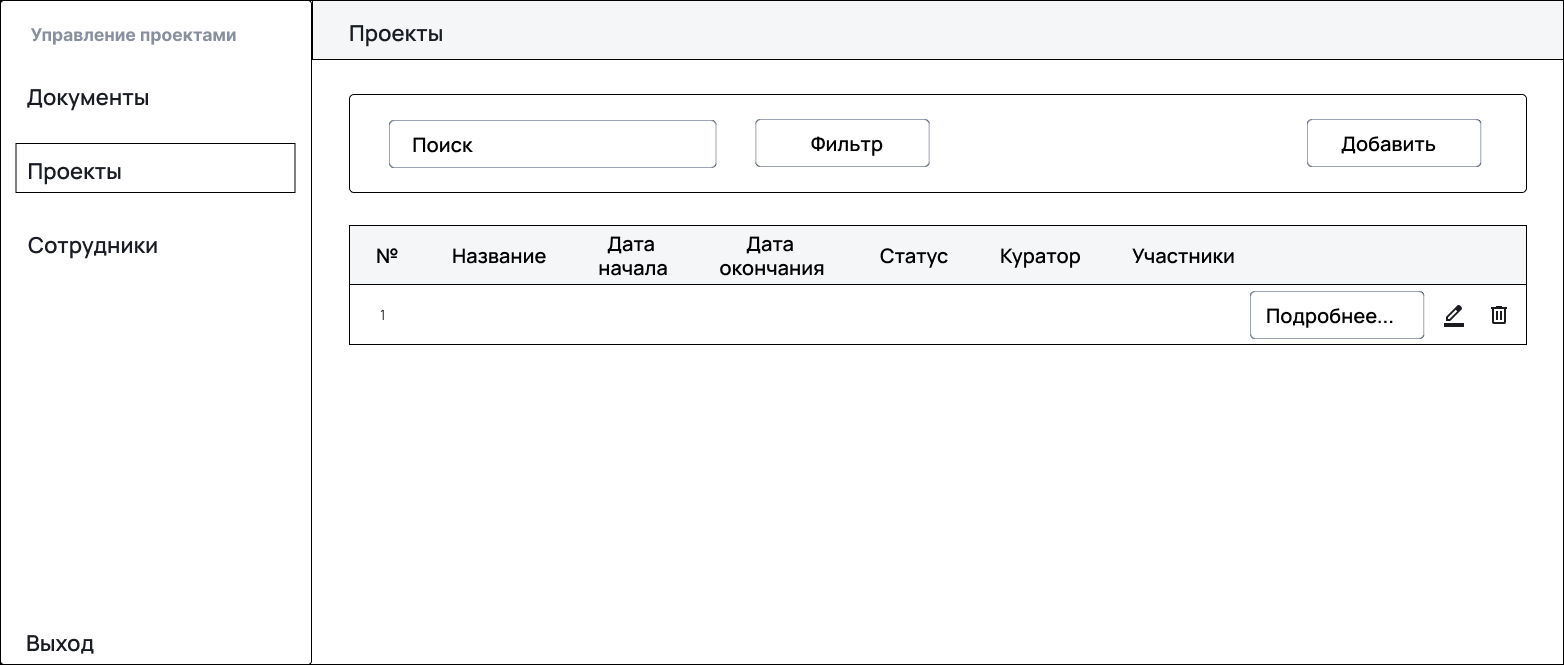


Рисунок 3.9 – Прототип страницы проектов

Прототип страницы проектов представляет собой удобный интерфейс, предназначенный для упрощения управления и организации проекта. Ключевые особенности:

* централизованная таблица проектов для быстрого доступа к основным сведениям о проекте;
* возможность использования дополнительных функций, таких как функции поиска, возможности фильтрации, страницы сведений о проекте и отслеживание хода выполнения.

На рисунке 3.10 изображен прототип формы добавления проекта.

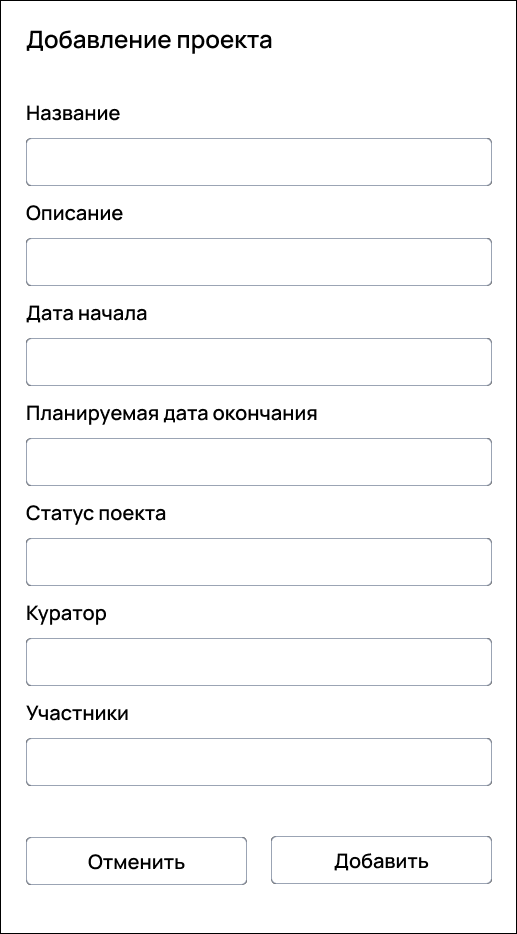


Рисунок 3.10 – Прототип формы добавления проекта

Прототип формы для добавления проекта обеспечивает удобный пользовательский интерфейс, который способствует упрощению процесса создания новых проектов. Его хорошо структурированный дизайн, интуитивно понятные поля и ясные инструкции помогают пользователям определить параметры проекта, обеспечивая эффективное начало работы над проектом.

На рисунке 3.11 изображен прототип страницы участников проекта.

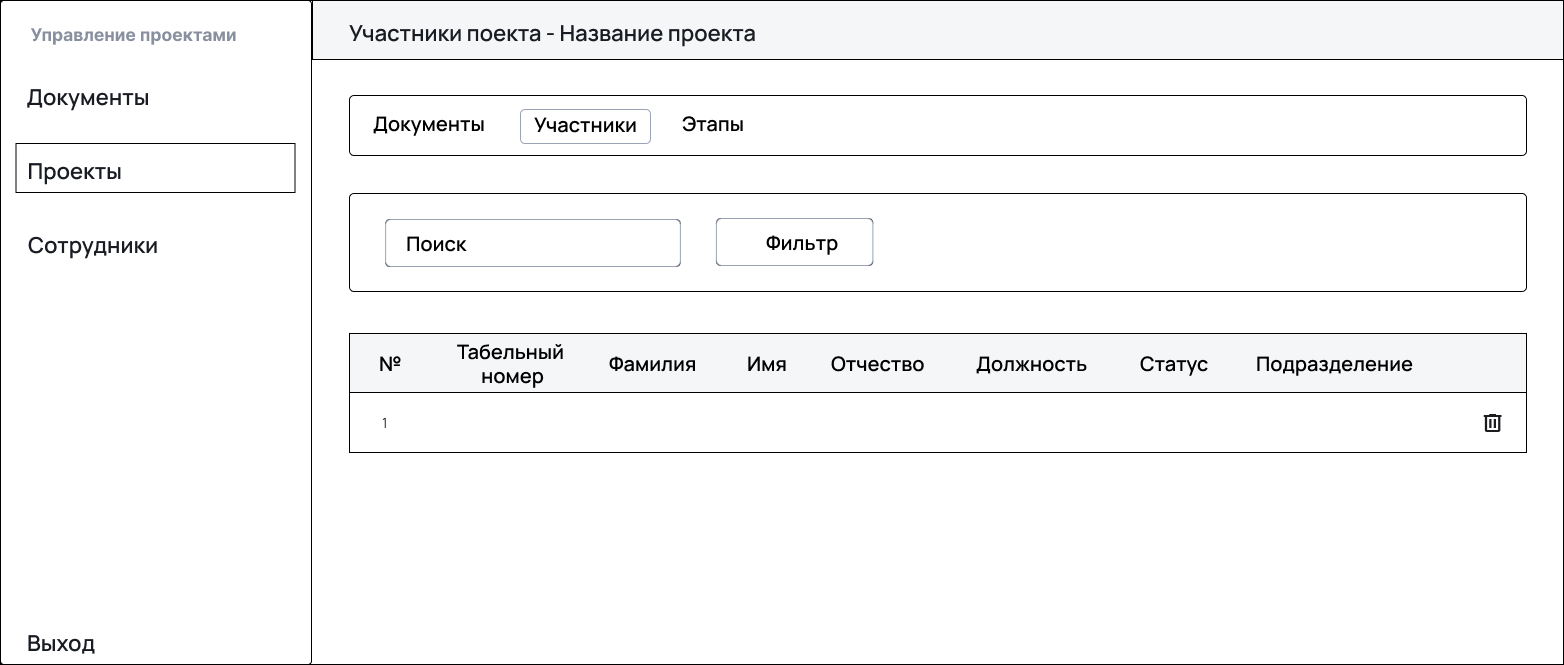


Рисунок 3.11 – Прототип страницы участников проекта

Прототип участников проекта служит централизованным хранилищем информации о лицах, участвующих в проекте, обеспечивая четкое представление об их обязанностях. Этот прототип включает в себя таблицу, в котором представлены сведения для каждого участника проекта.

На рисунке 3.12 изображен прототип страницы этапов проекта.

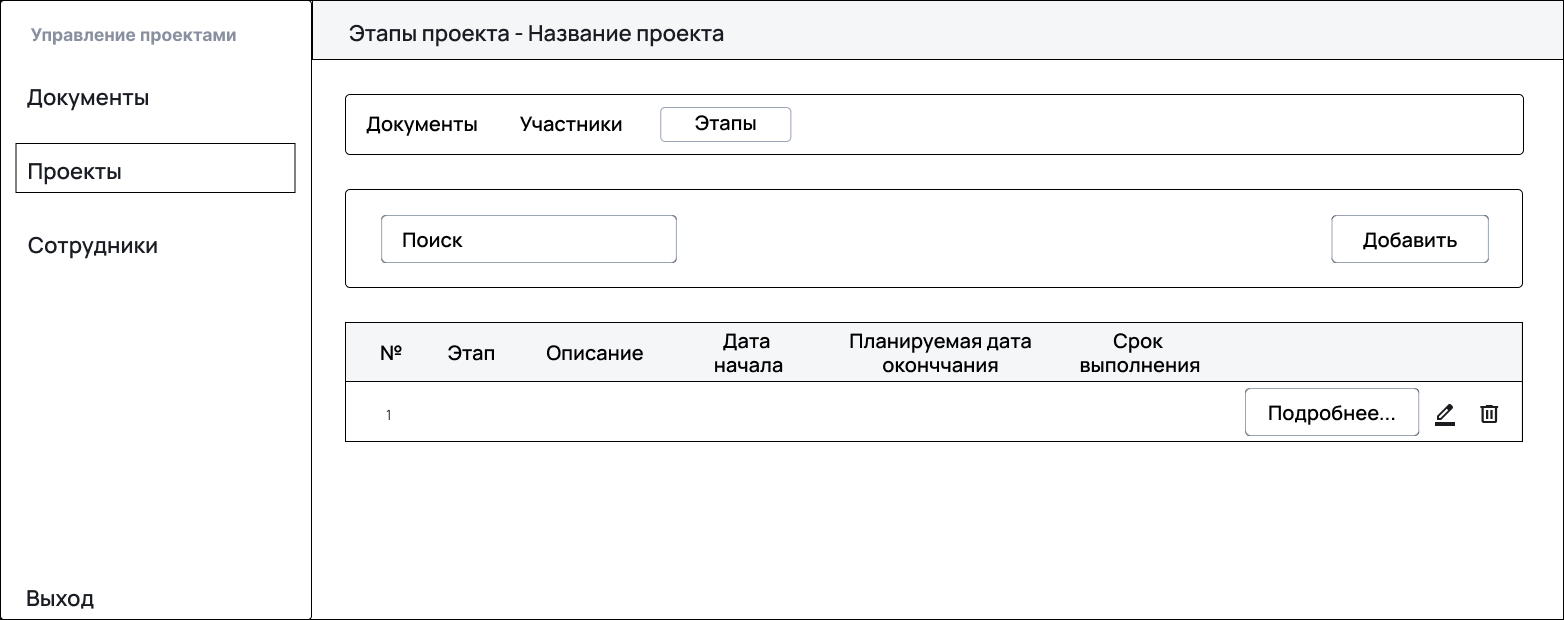


Рисунок 3.12 – Прототип страницы этапов проекта

Представленное изображение демонстрирует прототип страницы, которая является информационным ресурсом о различных этапах конкретного проекта. Этот веб-интерфейс разработан с целью обеспечить пользователям всесторонний обзор каждого этапа, включая его описание, продолжительность, ключевые события, участников и другую важную информацию.

На рисунке 3.13 изображен прототип формы добавления этапа проекта.

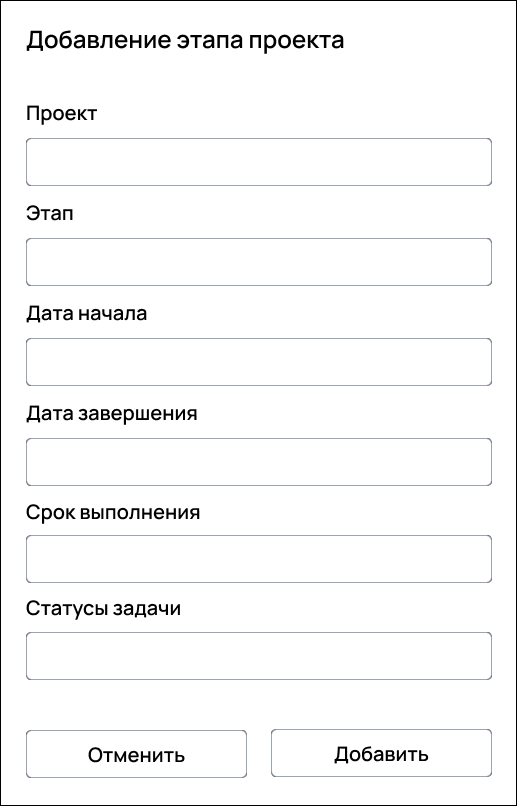


Рисунок 3.13 – Прототип формы добавления этапа проекта

На представленном изображении отображается прототип формы для добавления этапов в проект – это веб-интерфейс, разработанный для упрощения процесса создания новых этапов в системе управления проектами.

На рисунке 3.14 изображен прототип страницы подробности об этапе.

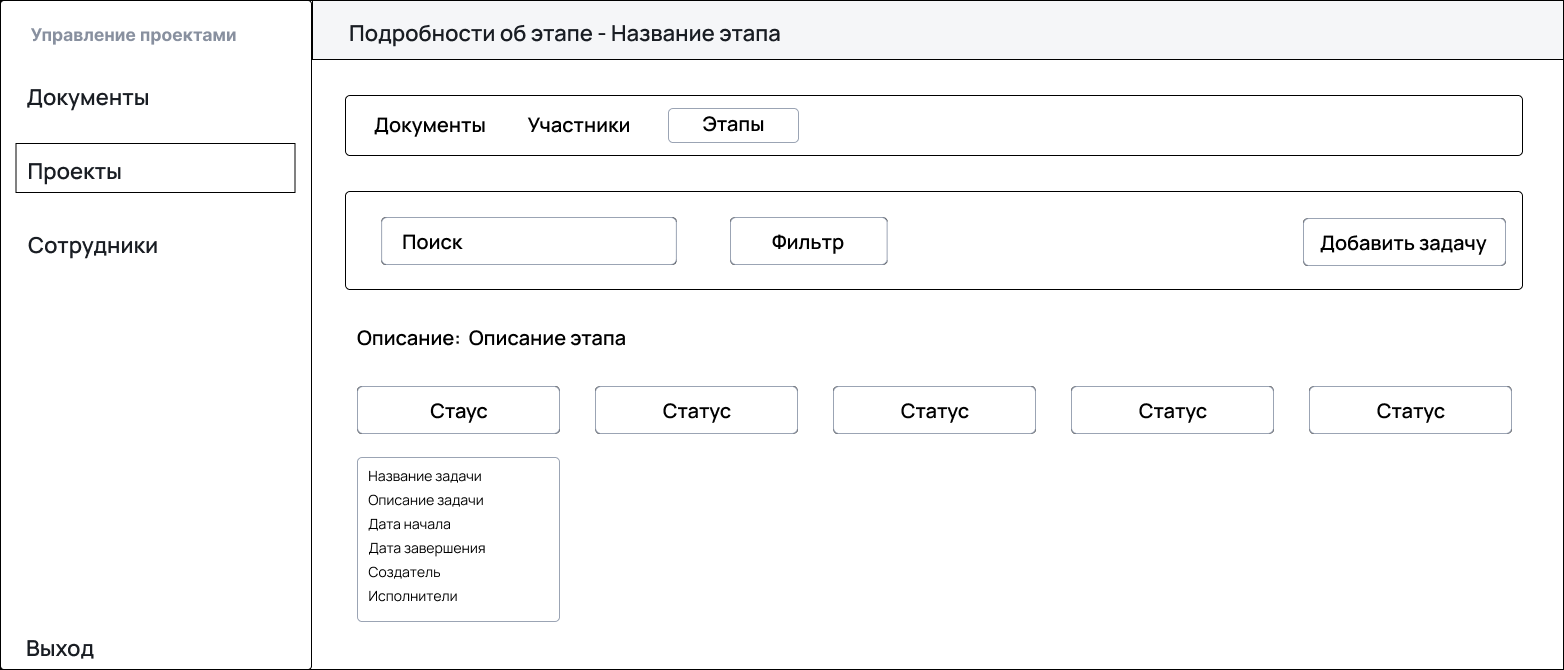


Рисунок 3.14 – Прототип страницы подробности об этапе

На данном изображении демонстрируется прототип инструмента, специально разработанного для эффективного управления проектами. Этот инструмент предоставляет возможность добавлять различные статусы к задачам, связанным с конкретными этапами проекта. Кроме того, этот инструмент позволяет пользователям создавать новые задачи в рамках каждого этапа проекта и назначать им соответствующие статусы.

На рисунке 3.15 изображен прототип формы добавления задачи.

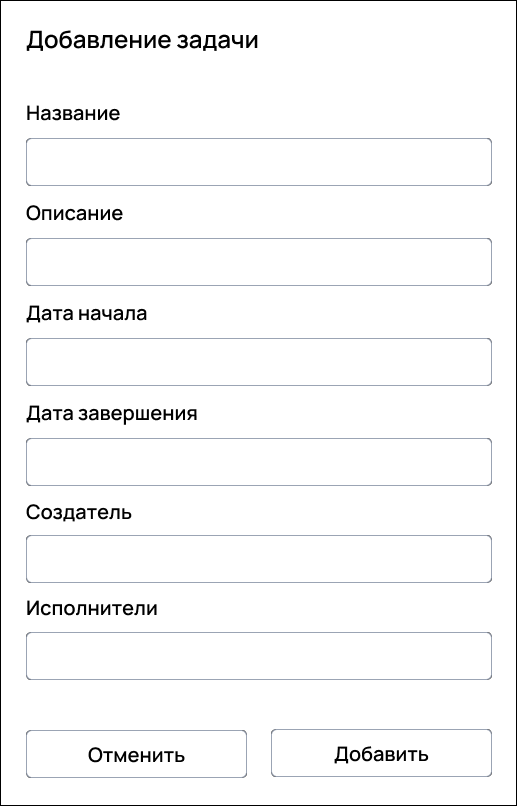


Рисунок 3.15 – Прототип формы добавления задачи

Изображенный прототип формы предназначен для добавления задач и обеспечивает удобство и эффективность при создании новых задач в системе управления проектами. Форма разработана таким образом, чтобы быть удобной и интуитивно понятной, что позволяет пользователям быстро вводить необходимые детали задачи без каких-либо затруднений или недопониманий.

На рисунке 3.16 изображен прототип информации о задаче.

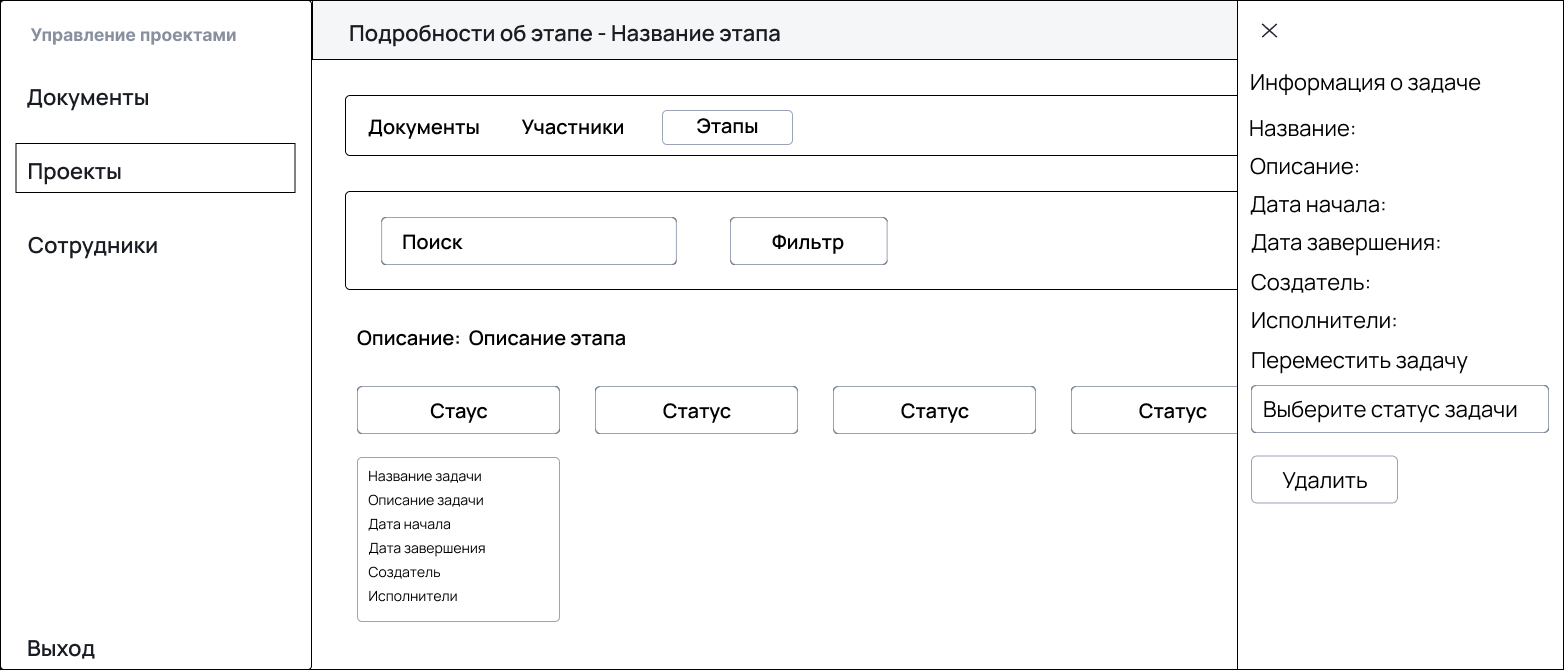


Рисунок 3.16 – Прототип информации о задаче

На представленном рисунке показан прототип для отображения информации о задачах в системе управления проектами. Дизайн нацелен на представление данных о задачах в понятной форме, что упрощает навигацию пользователей и обеспечивает легкий доступ к необходимой информации, что облегчает процесс управления задачами в рамках проекта.

Прототипирование пользовательского интерфейса играет ключевую роль в разработке, помогая определить структуру и функциональность продукта, а также обеспечивая единое понимание требований и целей проекта.

# 4 Разработка программного обеспечения

Разработка программного обеспечения – это процесс создания компьютерных программ и приложений, начиная от идеи и заканчивая готовым продуктом. Она включает в себя тщательное планирование, написание кода, тестирование и развертывание программы для использования конечными пользователями.

Каждый этап разработки программного обеспечения требует внимательного подхода и учета различных факторов, таких как потребности пользователей, технические требования, сроки и бюджет. Разработчики используют различные методологии и инструменты для облегчения процесса и повышения эффективности.

Цель разработки программного обеспечения – создать программу, которая эффективно решает поставленные задачи, легко используется конечными пользователями и соответствует ожиданиям заказчика. Это может быть веб-приложение, мобильное приложение, системное программное обеспечение или любое другое приложение, которое помогает улучшить работу и повысить эффективность бизнеса или организации [9].

## 4.1 Обоснование выбора инструментальных средств разработки программного продукта

Выбор языка TypeScript для разработки данного приложения был обусловлен несколькими ключевыми причинами. TypeScript представляет собой расширение языка JavaScript, которое добавляет статическую типизацию и множество других продвинутых функций. Это делает TypeScript мощным инструментом для разработки крупных и сложных приложений, таких как корпоративные веб-приложения.

Одним из основных преимуществ TypeScript является возможность выявления ошибок на этапе компиляции, что позволяет избежать множества ошибок времени выполнения и повысить надежность кода. Статическая типизация делает код более понятным и предсказуемым, облегчая сопровождение и масштабирование приложения.

Кроме того, TypeScript обладает широкой поддержкой инструментов разработки, таких как интегрированные среды разработки (IDE) и средства отладки, что делает процесс разработки более комфортным и продуктивным. Благодаря своей популярности и активному сообществу разработчиков, TypeScript предоставляет обширную документацию, множество сторонних библиотек и инструментов, что делает его привлекательным выбором для создания современных веб-приложений.

Таким образом, использование TypeScript в разработке данного приложения обеспечивает высокую производительность, надежность и удобство в разработке, что является ключевым фактором для успешной реализации корпоративного веб-приложения [10].

Выбор Visual Studio Code в качестве среды разработки для данного проекта был обусловлен множеством факторов, которые делают этот инструмент идеальным выбором для создания корпоративного веб-приложения на основе TypeScript и React.

Visual Studio Code (VS Code) – это бесплатный и открытый исходный код редактора кода. Он предоставляет обширный набор функций и инструментов для удобной и эффективной разработки программного обеспечения. Вот некоторые ключевые особенности и возможности Visual Studio Code:

* интерфейс пользователя: интерфейс Visual Studio Code дружелюбен к пользователю и легок в освоении. Он состоит из нескольких основных элементов, таких как редактор кода, боковая панель с файловым деревом и инструментами, строка поиска и навигации, а также верхнее меню с различными командами и инструментами;
* редактор кода: одним из ключевых элементов VS Code является его мощный редактор кода. Он поддерживает подсветку синтаксиса для множества языков программирования, автозавершение кода, быстрое переходы к определению переменных и функций, а также множество других функций, повышающих производительность при написании кода;
* расширения и плагины: система расширений Visual Studio Code позволяет добавлять новые функции и интеграции с другими инструментами и сервисами. В магазине расширений можно найти тысячи плагинов для поддержки различных языков программирования, интеграции с системами контроля версий, средств отладки, сборки и многое другое;
* интеграция с Git: Visual Studio Code имеет встроенную поддержку системы контроля версий Git. Это позволяет разработчикам эффективно работать с репозиториями Git, отслеживать изменения, выполнять коммиты и слияния, а также смотреть историю изменений прямо из редактора кода;
* отладка: редактор Visual Studio Code обеспечивает возможности отладки приложений на различных языках программирования. Он интегрируется с различными средствами отладки, позволяя разработчикам отслеживать и исправлять ошибки в своем коде;
* интеграция с облачными сервисами: Visual Studio Code интегрируется с различными облачными сервисами, такими как Microsoft Azure. Это обеспечивает возможность разработки и развертывания приложений в облаке прямо из редактора кода;
* кроссплатформенность: редактор кода поддерживает операционные системы Windows, macOS и Linux, что делает его универсальным инструментом для разработки на различных платформах;
* сообщество и поддержка: Visual Studio Code имеет активное сообщество пользователей и разработчиков. Это означает, что всегда можно найти помощь, руководства и решения проблем через форумы, блоги, социальные сети и другие ресурсы [11].

React – это инновационная библиотека JavaScript, созданная компанией Facebook, которая стала стандартом для разработки современных веб-приложений. Ее применение в разработке корпоративного веб-приложения обусловлено рядом фундаментальных преимуществ и особенностей, которые делают ее одним из лучших выборов для создания динамичных пользовательских интерфейсов:

* компонентный подход: основной концепцией React является компонентный подход к разработке интерфейса. Весь пользовательский интерфейс разбивается на множество небольших, автономных компонентов, каждый из которых отвечает за свою часть интерфейса. Это упрощает структурирование кода, повышает его читаемость и делает приложение более масштабируемым и поддерживаемым;
* виртуальный DOM: React использует виртуальное представление DOM для эффективного обновления пользовательского интерфейса. Вместо того чтобы непосредственно изменять реальный DOM при каждом обновлении, React сравнивает виртуальное представление с реальным DOM и применяет только необходимые изменения. Это повышает производительность приложения и снижает нагрузку на браузер;
* односторонний поток данных: данные в React передаются сверху вниз по иерархии компонентов через пропсы. Это позволяет легко отслеживать поток данных в приложении и упрощает его разработку и отладку. Родительские компоненты могут передавать данные своим дочерним компонентам, обеспечивая единый и надежный источник данных;
* JSX: JSX – это синтаксическое расширение JavaScript, которое позволяет объединять код JavaScript и разметку HTML внутри компонентов React. Это делает код более декларативным и понятным, а также упрощает разработку пользовательского интерфейса, позволяя разработчикам создавать компоненты в виде чистого и выразительного кода;
* активное сообщество и экосистема: React обладает огромным и активным сообществом разработчиков, что способствует его постоянному развитию и улучшению. Существует множество сторонних библиотек, инструментов и расширений, разработанных сообществом, которые облегчают разработку приложений на React и расширяют его возможности.

Использование React в разработке корпоративного веб-приложения обеспечивает высокую производительность, гибкость и удобство в разработке. Благодаря его инновационным концепциям и функциональным возможностям, React остается одним из самых востребованных инструментов в сфере веб-разработки [12].

MongoDB – это гибкая и мощная система управления базами данных типа NoSQL, которая была выбрана для хранения данных в разрабатываемом корпоративном веб-приложении. В отличие от традиционных реляционных баз данных, MongoDB использует гибкую схему данных в формате JSON, что позволяет разработчикам легко масштабировать и модифицировать структуру данных в соответствии с потребностями приложения.

Основные преимущества MongoDB включают в себя:

* гибкая схема данных: MongoDB не требует заранее определенной схемы данных, что позволяет хранить документы различной структуры в одной коллекции. Это делает MongoDB идеальным выбором для приложений, где структура данных может изменяться во времени или зависит от пользовательских потребностей;
* высокая производительность: MongoDB обеспечивает высокую производительность при работе с большими объемами данных. Благодаря горизонтальному масштабированию и распределенной архитектуре, MongoDB способен обрабатывать огромные нагрузки и обеспечивать быстрый доступ к данным даже при высокой нагрузке;
* поддержка запросов и агрегации: MongoDB предоставляет мощные средства для выполнения запросов и агрегаций данных, включая различные операторы, функции и индексы, что обеспечивает эффективный доступ к данным и обработку сложных запросов;
* гибкий язык запросов: MongoDB использует язык запросов, основанный на JavaScript, что делает его более понятным и удобным для разработчиков. Он поддерживает широкий спектр операторов и функций для выполнения различных операций с данными, включая фильтрацию, сортировку, агрегацию и многое другое;
* масштабируемость и отказоустойчивость: MongoDB поддерживает горизонтальное масштабирование, что позволяет легко увеличивать объемы данных и обрабатывать большие нагрузки. Кроме того, MongoDB обеспечивает отказоустойчивость и репликацию данных, что гарантирует надежное хранение и доступ к данным даже в случае сбоев системы.

В целом, MongoDB является идеальным выбором для хранения данных в корпоративном веб-приложении благодаря своей гибкой схеме данных, высокой производительности, масштабируемости и надежности. Его использование позволяет разработчикам эффективно управлять данными и обеспечить надежное функционирование приложения в условиях высокой нагрузки и изменяющихся потребностей пользователей [13].

## 4.2 Описание алгоритма программы

Описание алгоритма программы – это краткое изложение последовательности шагов, необходимых для выполнения определенной задачи программой [14].

Данная блок-схема описывает процесс запуска программы и авторизации.

Шаг 1. Запуск программы.

Шаг 2. Открытие окна авторизации.

Шаг 3. Ввод логина и пароля.

Шаг 4. Вопрос: Данные корректны? Если нет, то переходим к шагу 3. Если да, то переходим к шагу 5.

Шаг 5. Вопрос: Роль пользователя администратор? Если нет, то переходим к шагу 6. Если да, то открываем окно администратора и переходим к шагу 10.

Шаг 6. Выбор действия для сотрудника.

Шаг 7. Просмотр информации о сотрудниках. Если да, то переходим к шагу 8, если нет, то переходим к шагу 16.

Шаг 8. Фильтрация информации о сотрудниках. Если да, то вывод информации, если нет, то переходим к шагу 9.

Шаг 9. Поиск информации о сотруднике. Если да, то вывод информации.

Шаг 10. Выбор действия для администратора.

Шаг 11. Просмотр информации о сотрудниках. Если да, то переходим к шагу 12, если нет, то переходим к шагу 20.

Шаг 12. Фильтрация информации о сотрудниках. Если да, то вывод информации, если нет, то переходим к шагу 13.

Шаг 13. Поиск информации о сотруднике. Если да, то вывод информации, если нет, то переходим к шагу 14.

Шаг 14. Добавление нового сотрудника. Если да, то переходим к заполнению формы, если нет, то переходим к шагу 15.

Шаг 15. Редактирование информации о сотруднике. Если да, то переходим к редактированию формы.

Шаг 16. Просмотр информации о проекте. Если да, то переходим к шагу 17, если нет, то переходим к шагу 6.

Шаг 17. Фильтрация информации о проектах. Если да, то вывод информации, если нет, то переходим к шагу 18.

Шаг 18. Поиск информации о проекте. Если да, то вывод информации, если нет, то переходим к шагу 19.

Шаг 19. Подробнее о проекте. Если да, то переходим к шагу 25.

Шаг 20. Просмотр информации о проекте. Если да, то переходим к шагу 21, если нет, то переходим к шагу 10.

Шаг 21. Фильтрация информации о проектах. Если да, то вывод информации, если нет, то переходим к шагу 22.

Шаг 22. Поиск информации о проекте. Если да, то вывод информации, если нет, то переходим к шагу 23.

Шаг 23. Добавление нового проекта. Если да, то переходим к заполнению формы, если нет, то переходим к шагу 24.

Шаг 24. Подробнее о проекте. Если да, то переходим к шагу 28.

Шаг 25. Просмотр информации об участниках проекта. Если да, то переходим к шагу 26, если нет, то переходим к шагу 31.

Шаг 26. Фильтрация информации об участниках проекта. Если да, то вывод информации, если нет, то переходим к шагу 27.

Шаг 27. Поиск информации об участниках проекта. Если да, то вывод информации.

Шаг 28. Просмотр информации об участниках проекта. Если да, то переходим к шагу 29, если нет, то переходим к шагу 38.

Шаг 29. Фильтрация информации об участниках проекта. Если да, то вывод информации, если нет, то переходим к шагу 30.

Шаг 30. Поиск информации об участниках проекта. Если да, то вывод информации.

Шаг 31. Просмотр информации об этапах проекта. Если да, то переходим к шагу 32, если нет, то переходим к шагу 6.

Шаг 32. Поиск информации об этапах проекта. Если да, то вывод информации, если нет. То переходим к шагу 33.

Шаг 33. Подробнее об этапе проекта. Если да, то переходим к шагу 34.

Шаг 34. Просмотр этапа проекта. Если да, то переходим к шагу 35.

Шаг 35. Фильтрация информации об этапе проекта. Если да, то вывод информации, если нет, то переходим к шагу 36.

Шаг 36. Поиск информации об этапе проекта. Если да, то вывод информации, если нет, то переходим к шагу 37.

Шаг 37. Добавление задачи. Если да, то переходим к заполнению формы.

Шаг 38. Просмотр информации об этапах проекта. Если да, то переходим к шагу 39, если нет, то переходим к шагу 10.

Шаг 39. Поиск информации об этапах проекта. Если да, то вывод информации, если нет. То переходим к шагу 40.

Шаг 40. Добавление этапа проекта. Если да, то переходим к заполнению формы, если нет, то переходим к шагу 41.

Шаг 41. Подробнее об этапе проекта. Если да, то переходим к шагу 42.

Шаг 42. Просмотр этапа проекта. Если да, то переходим к шагу 43.

Шаг 43. Фильтрация информации об этапе проекта. Если да, то вывод информации, если нет, то переходим к шагу 44.

Шаг 44. Поиск информации об этапе проекта. Если да, то вывод информации, если нет, то переходим к шагу 45.

Шаг 45. Добавление задачи. Если да, то переходим к заполнению формы.

Шаг 46. Вопрос: Выход из учётной записи? Если нет, то переходим к шагу 6, 10. Если да, то переходим к шагу 47.

Шаг 47. Закрытие программы.

Блок-схема алгоритма представлена на рисунках 3.1-3.4.

Код программного модуля приведен в приложении А.

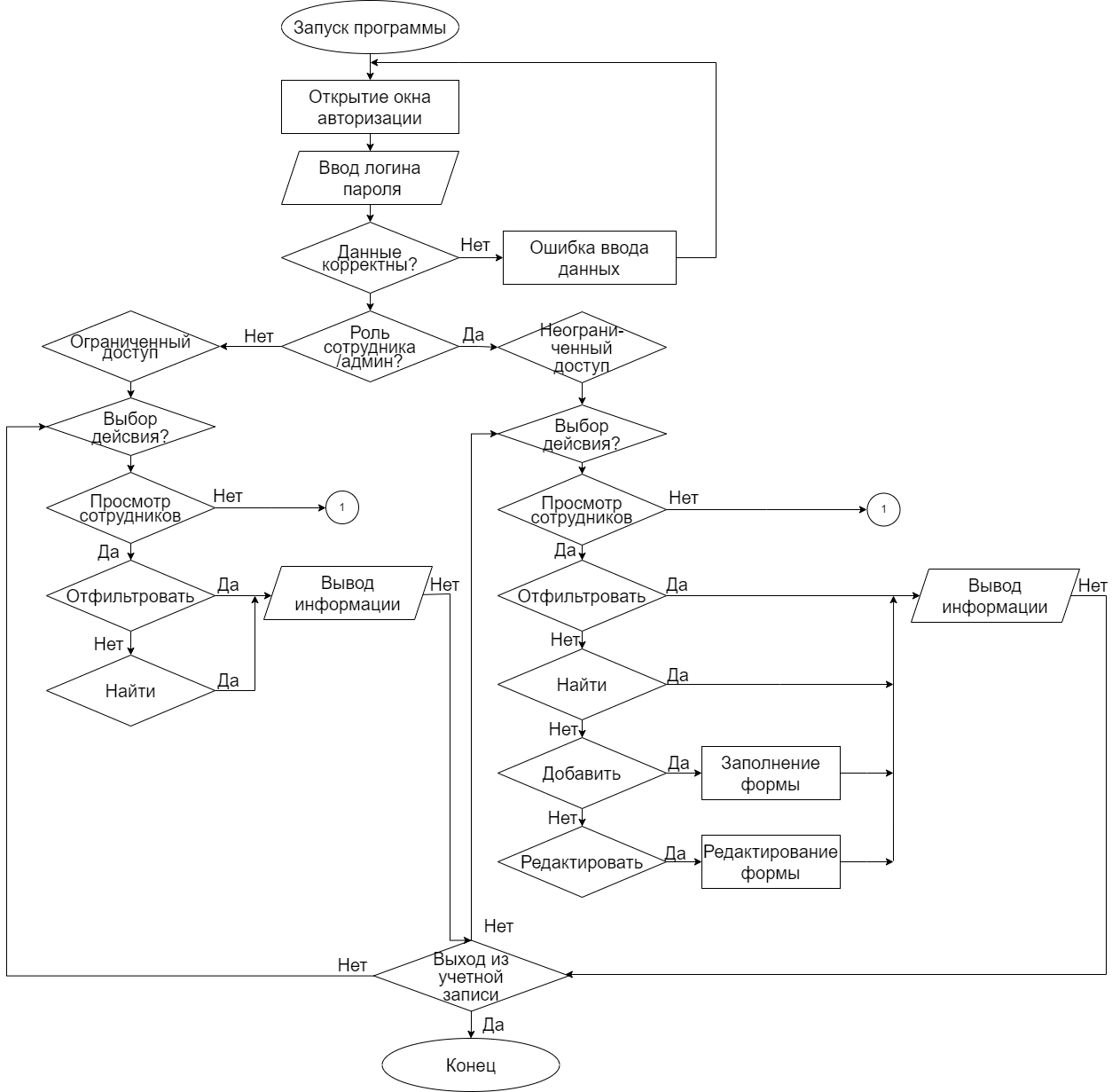


Рисунок 4.1 – Блок-схема веб-приложения

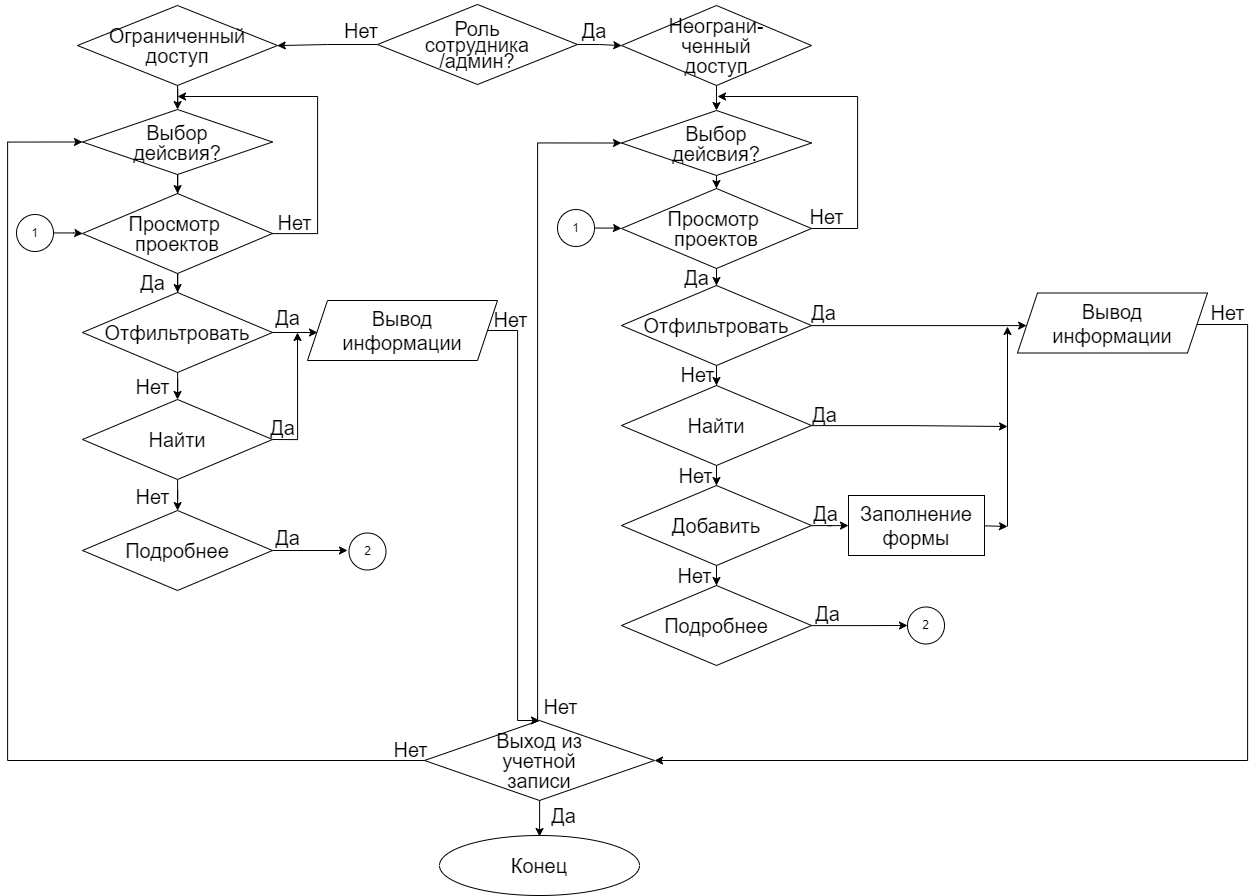


Рисунок 4.2 – Блок-схема веб-приложения

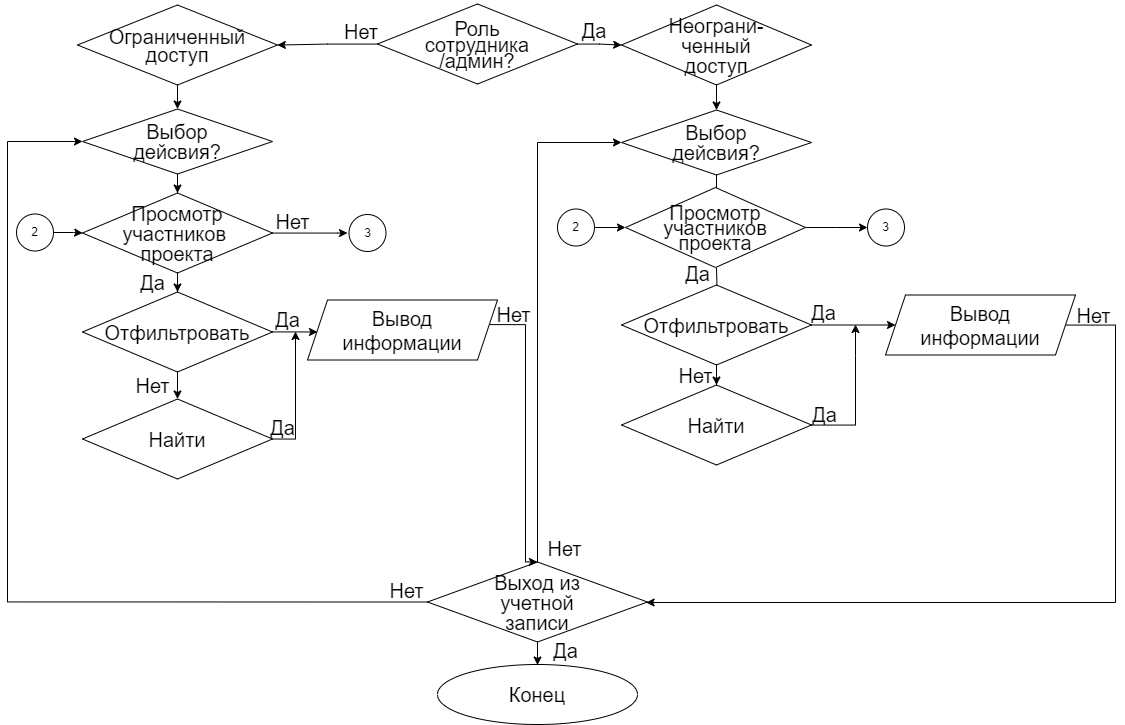


Рисунок 4.3 – Блок-схема веб-приложения

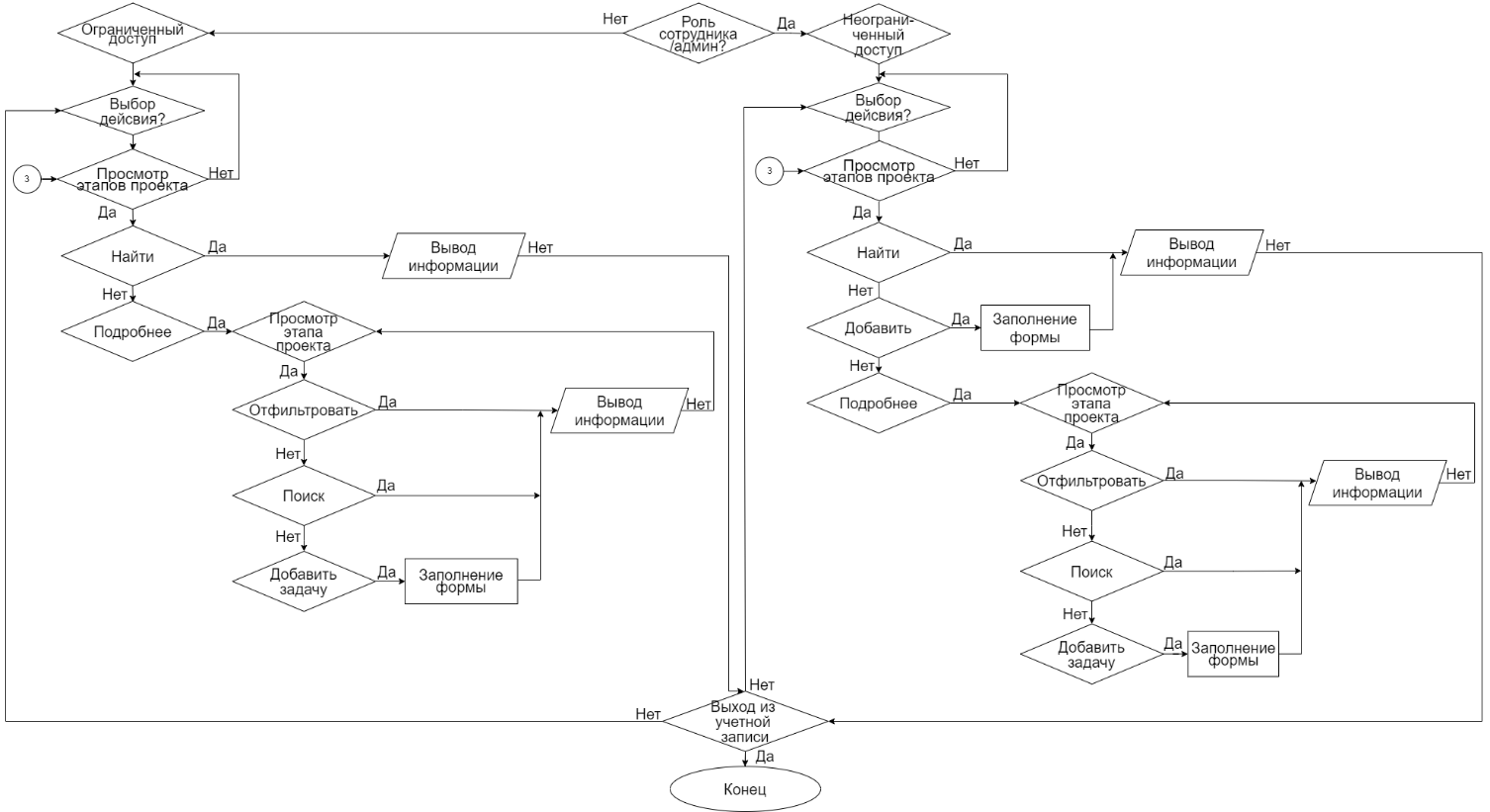


Рисунок 4.4 – Блок-схема веб-приложения

## 4.3 Описание интерфейса программного продукта

Интерфейс программного продукта – это совокупность элементов и функций, предназначенных для взаимодействия пользователя с программой. Он обеспечивает способ ввода и вывода информации, управления приложением, а также обеспечивает пользователю доступ к функциональности программы [15].

На рисунке 4.5 представлен интерфейс авторизации.

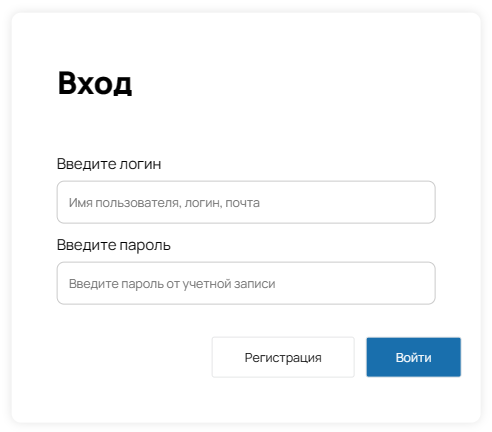


Рисунок 4.5 – Интерфейс авторизации

Интерфейс авторизации предоставляет простой способ входа в систему.

На рисунке 4.6 представлен интерфейс регистрации.

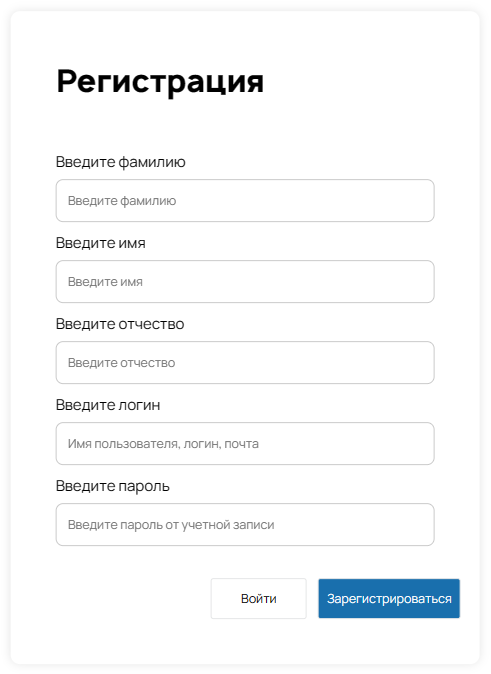


Рисунок 4.6 – Интерфейс регистрации

Интерфейс регистрации предоставляет простой и удобный способ создания учетной записи в системе.

На рисунке 4.7 представлен интерфейс страницы сотрудники.

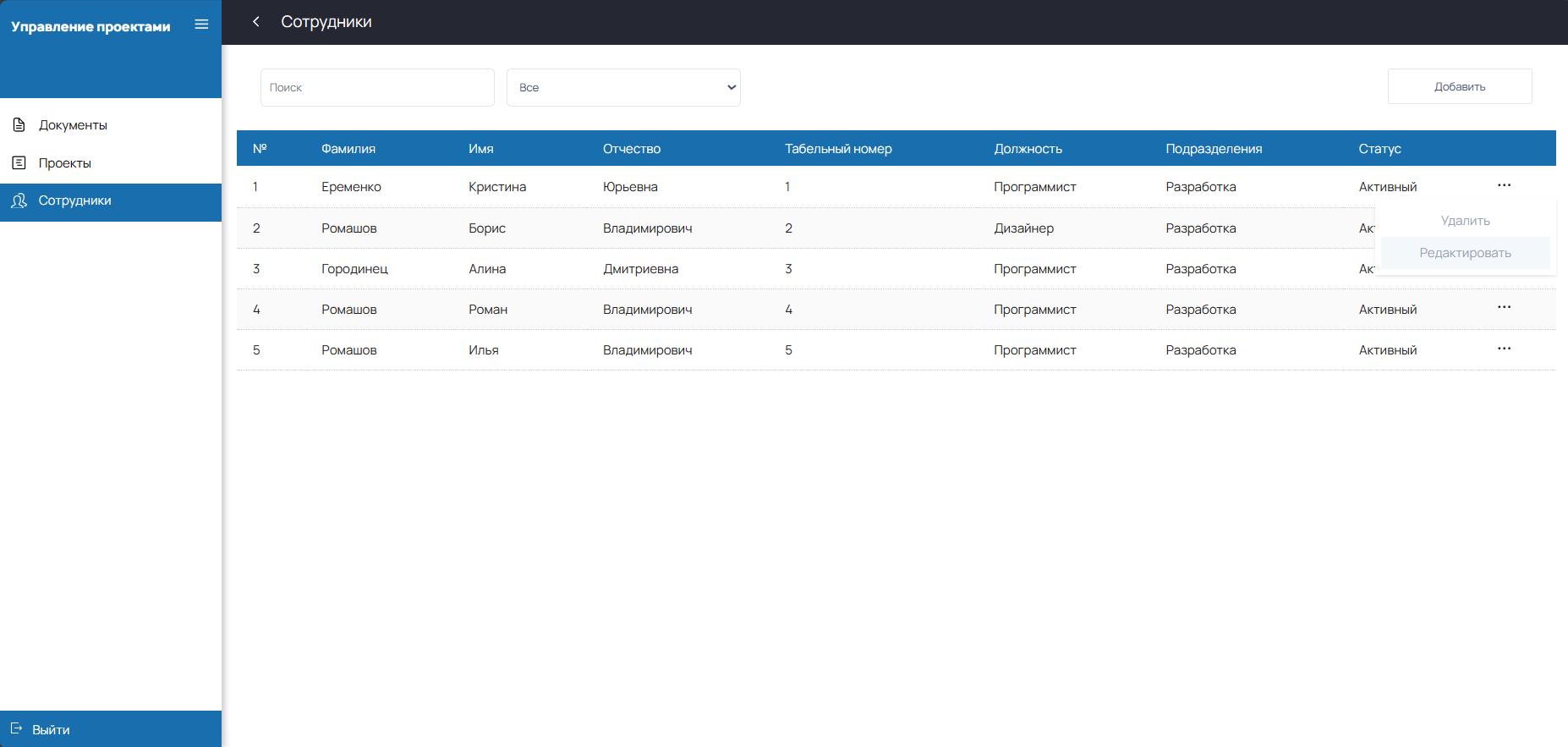


Рисунок 4.7 – Интерфейс страницы сотрудников

Интерфейс страницы сотрудников отображает список всех работников и предоставляет следующие функциональности пользователям:

* поиск сотрудников по имени, фамилии и отчеству;
* фильтрация сотрудников по их должности;
* просмотр полного списка сотрудников;
* добавление новых сотрудников в список;
* редактирование информации о существующих сотрудниках;
* удаление сотрудников.

Представленный интерфейс страницы сотрудников обеспечивает удобный доступ к информации о персонале и позволяет пользователям управлять списком сотрудников, включая поиск, фильтрацию, добавление, удаление и редактирование данных.

На рисунке 4.8 изображено окно добавления сотрудника.

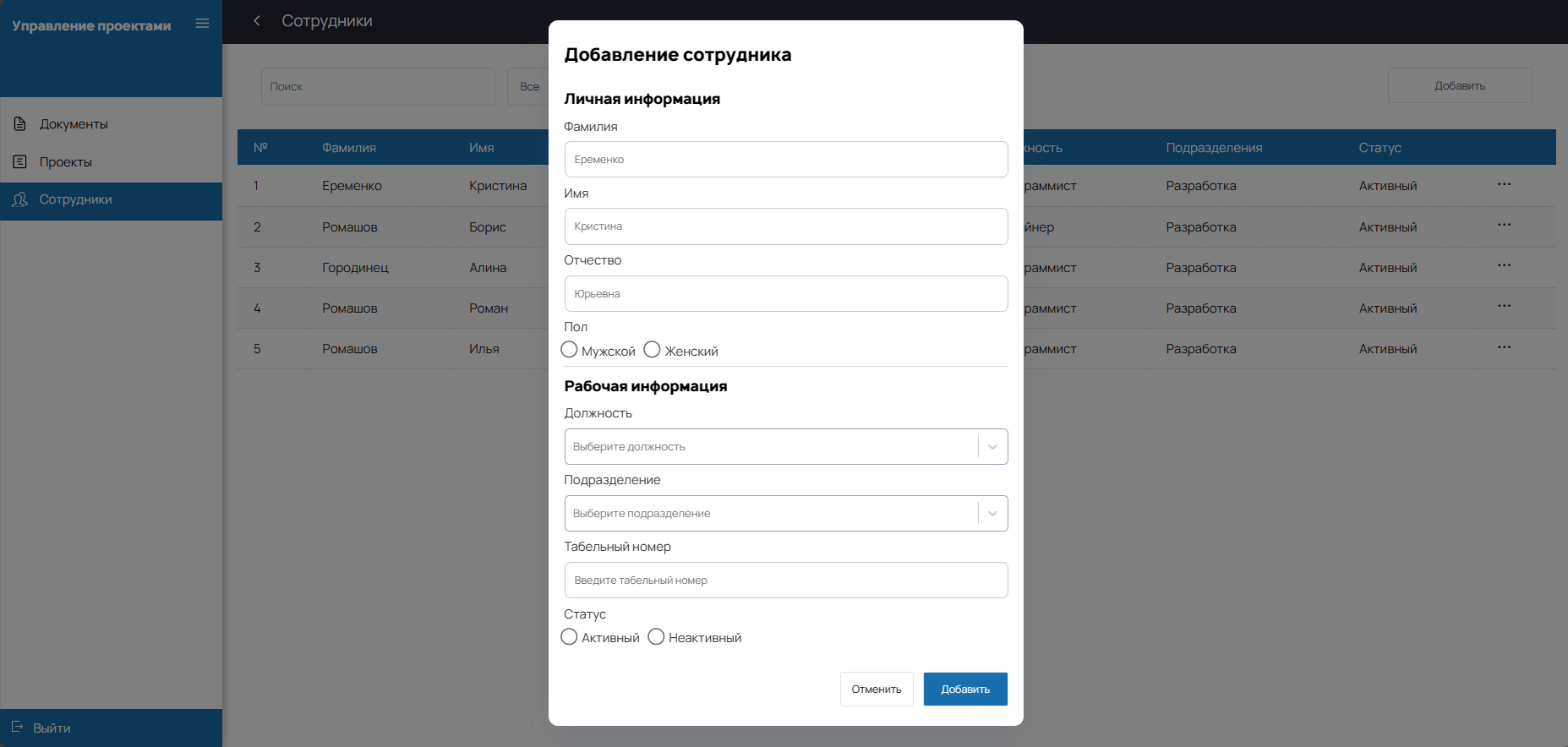


Рисунок 4.8 – Окно добавления сотрудника

На изображении показан интерфейс добавления сотрудника. Интерфейс представляет собой форму, которую необходимо заполнить для добавления нового сотрудника в список.

На рисунке 4.9 изображено окно редактирования сотрудника.

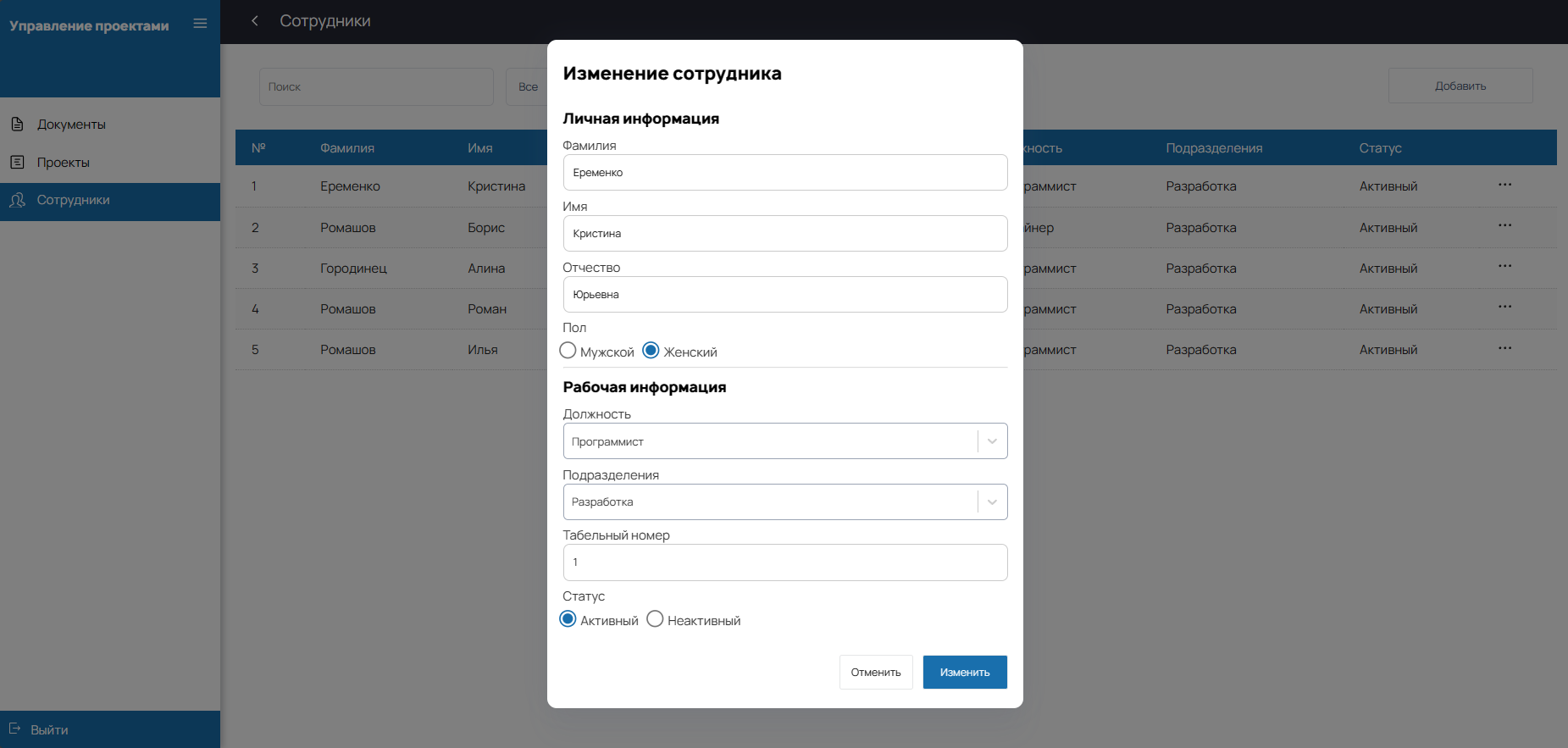


Рисунок 4.9 – Окно редактирования сотрудника

Интерфейс редактирования информации о сотруднике, изображенный на рисунке, дает возможность внесения изменений в данные о существующем работнике в системе.

На рисунке 4.10 представлен интерфейс страницы проектов.

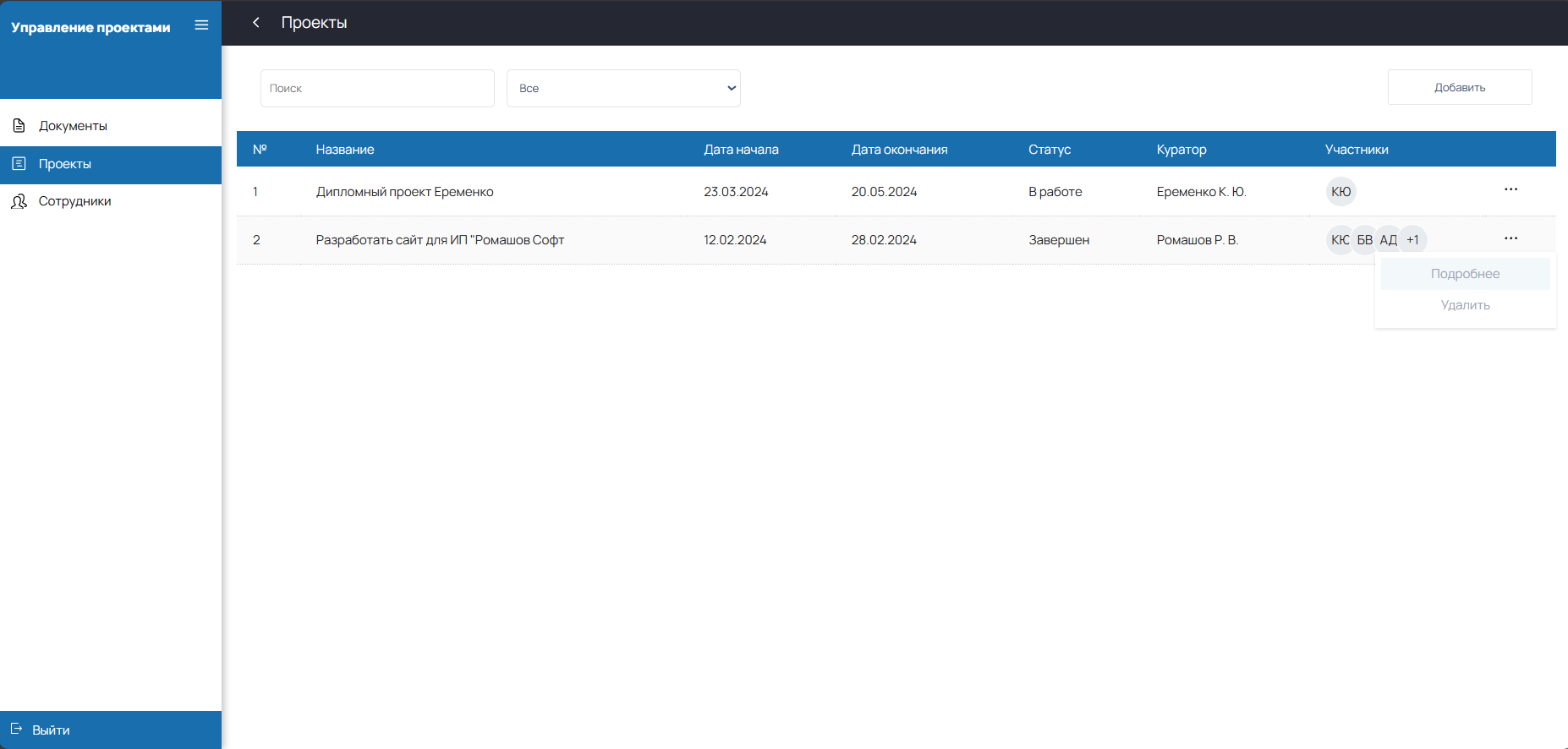


Рисунок 4.10 – Интерфейс страницы проекты

Страница проектов предоставляет пользователю удобный доступ ко всем проектам, имеющимся в системе, и обладает следующими возможностями:

* просмотр списка всех проектов, созданных в системе;
* поиск проектов по названию;
* фильтрация проектов по их статусу;
* добавление новых проектов в список;
* удаление ненужных проектов;
* переход на страницу с подробной информацией о конкретном проекте.

Этот интерфейс обеспечивает удобное управление проектами, позволяя пользователям легко находить нужные проекты, добавлять новые, удалять ненужные и получать подробную информацию о каждом из них.

На рисунке 4.11 изображено окно добавления проекта.

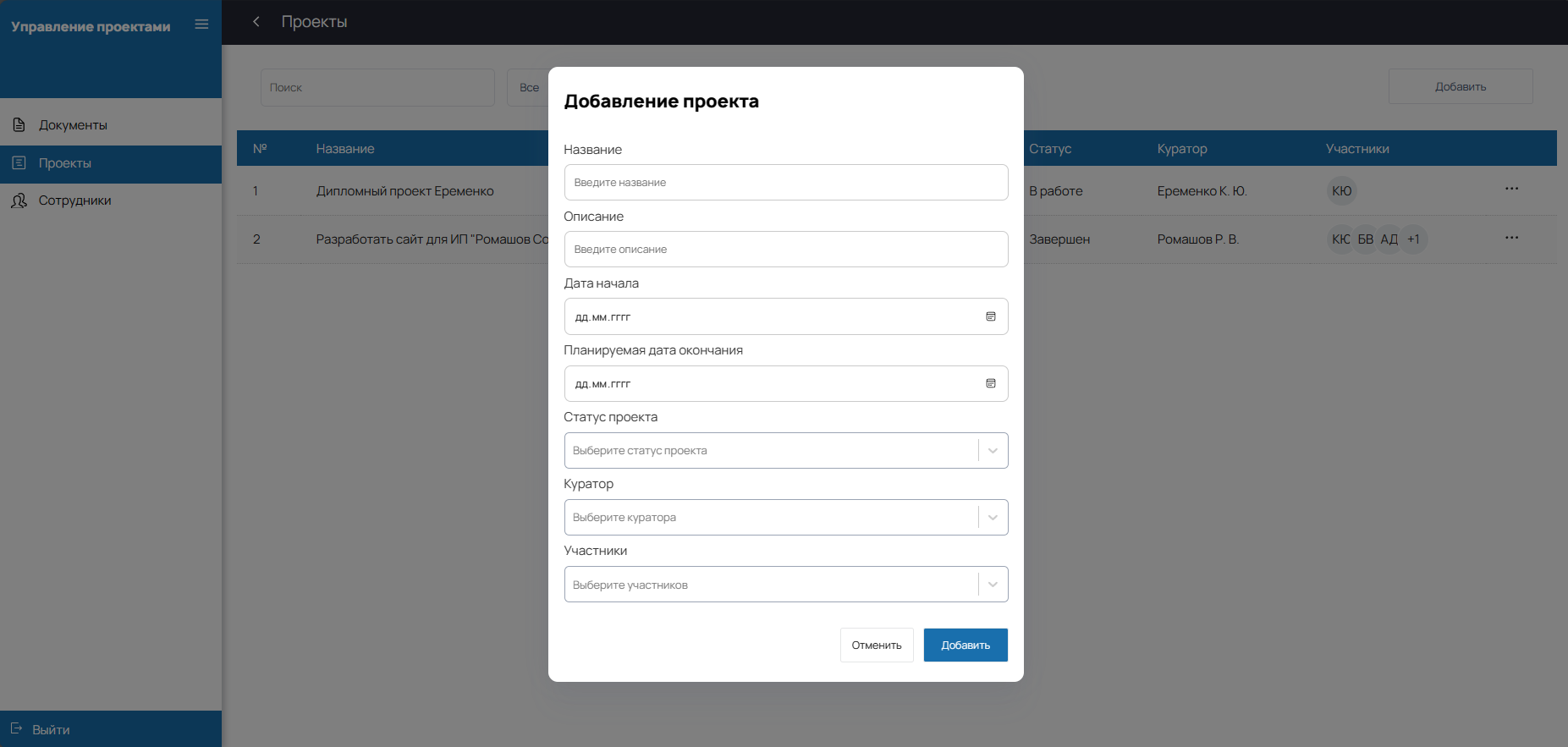


Рисунок 4.11 – Окно добавления проекта

Интерфейс окна добавления проекта прост, понятен и функционален. Он позволяет пользователю легко и быстро создать новый проект.

На рисунке 4.12 изображен интерфейс страницы участников конкретного проекта.

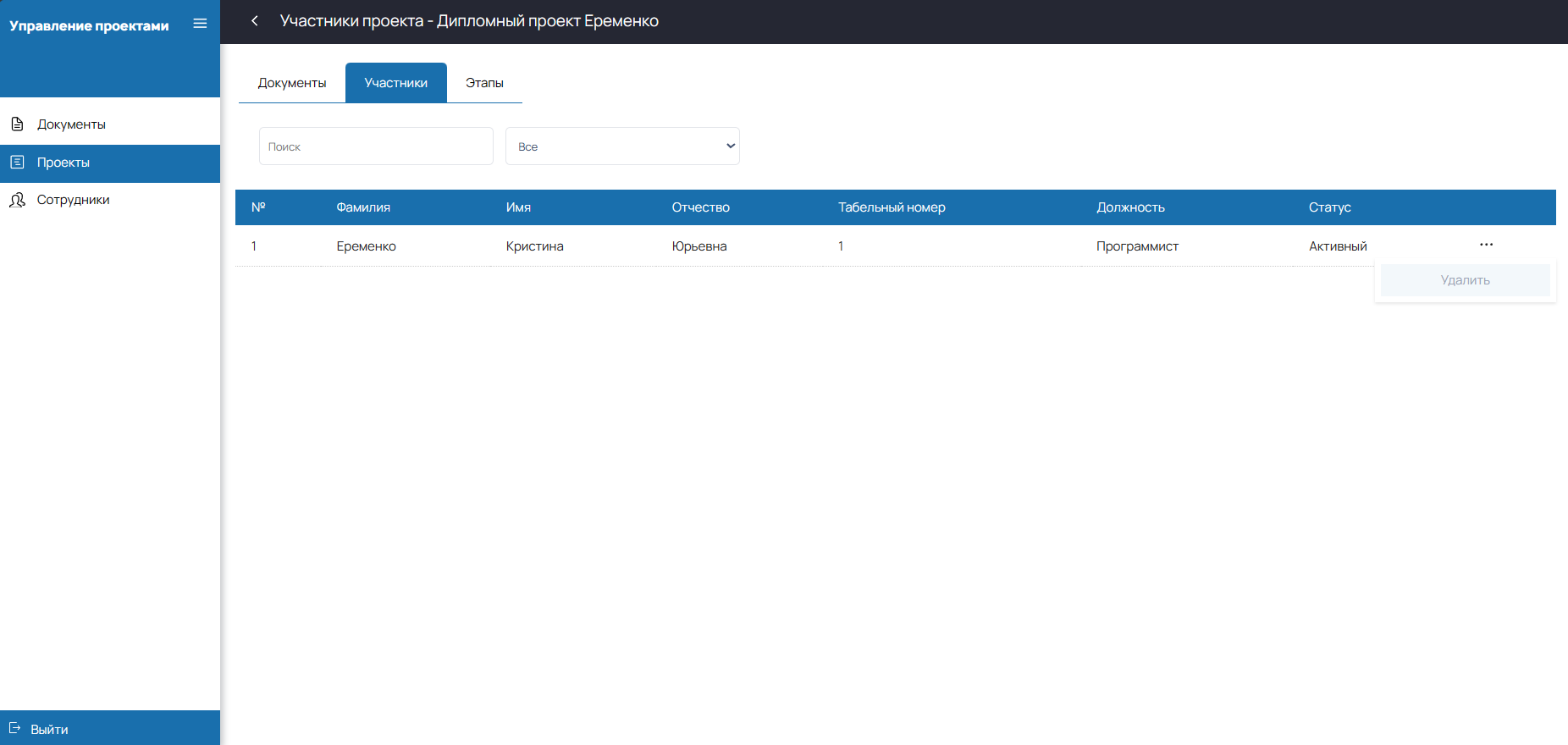


Рисунок 4.12 – Интерфейс страницы участников проекта

Интерфейс страницы участников конкретного проекта представлен простым, понятным и функциональным способом. Он обеспечивает пользователю удобный просмотр списка участников проекта и позволяет легко удалять их из проекта.

На странице участников проекта пользователи могут:

* просматривать список участников проекта;
* удалять участников из проекта;
* фильтровать участников по должности;
* осуществлять поиск сотрудников по имени, фамилии и отчеству.

Такой интерфейс обеспечивает эффективное взаимодействие пользователей с информацией о составе участников проекта и позволяет легко настраивать отображение данных в соответствии с потребностями пользователей.

На рисунке 4.13 изображен интерфейс страницы этапов конкретного проекта.

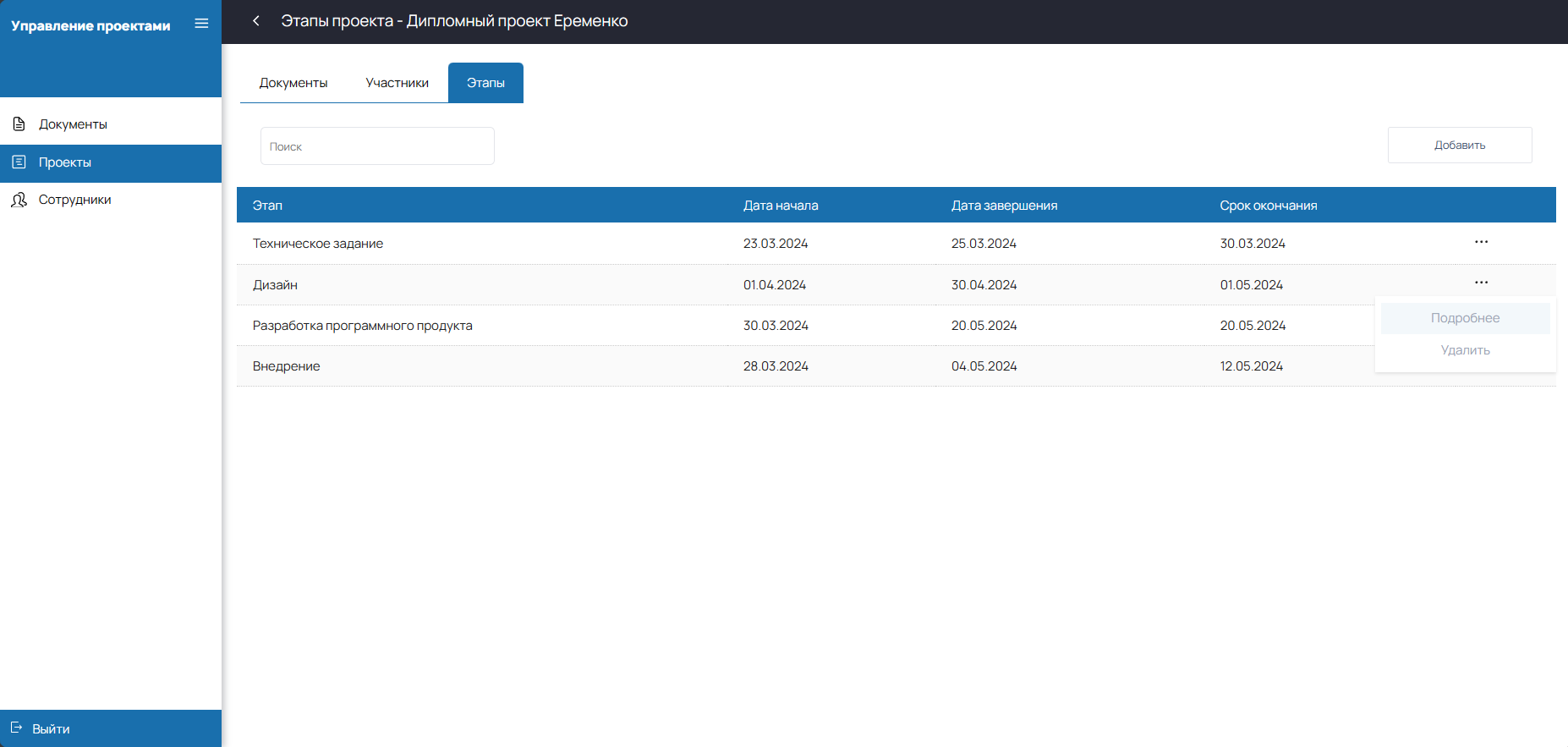


Рисунок 4.13 – Интерфейс страницы этапов проекта

Страница этапов проекта, изображенная на экране, предоставляет пользователям удобную возможность просмотра списка этапов проекта и добавления новых этапов.

На данной странице пользователи могут:

* просматривать список этапов проекта;
* добавлять новые этапы в проект;
* удалять этапы из проекта;
* осуществлять поиск по названию этапа;
* переходить на страницу с подробной информацией о конкретном этапе проекта.

Такой интерфейс обеспечивает удобное взаимодействие с информацией о этапах проекта, позволяя пользователям легко управлять этапами и настраивать их в соответствии с потребностями проекта.

На рисунке 4.14 изображено окно добавления этапа проекта.

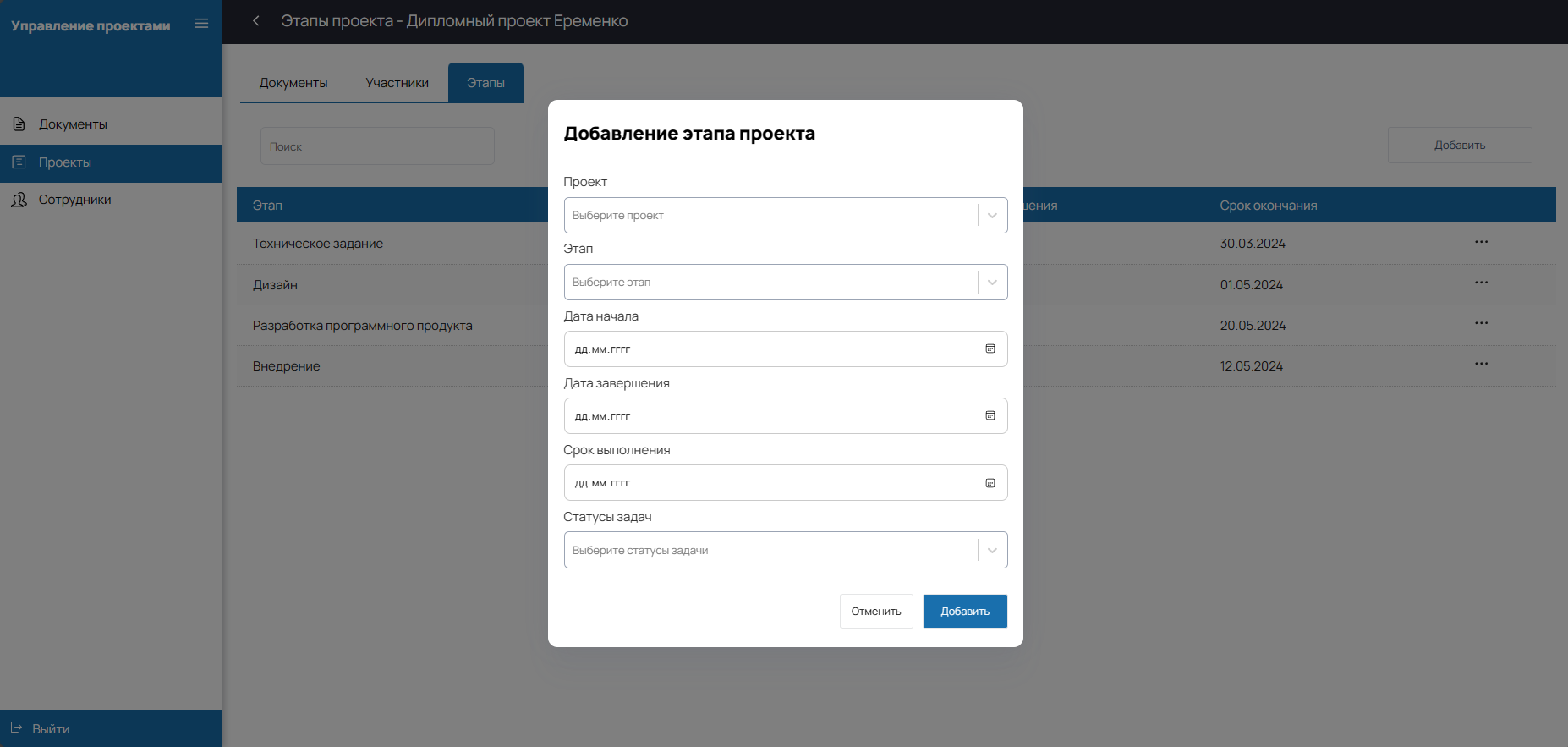


Рисунок 4.14 – Окно добавления этапа проекта

Окно добавления этапа проекта, представленное на изображении, позволяет пользователям создавать новые этапы в выбранном проекте.

На рисунке 4.15 изображен интерфейс страницы подробности этапа конкретного проекта.

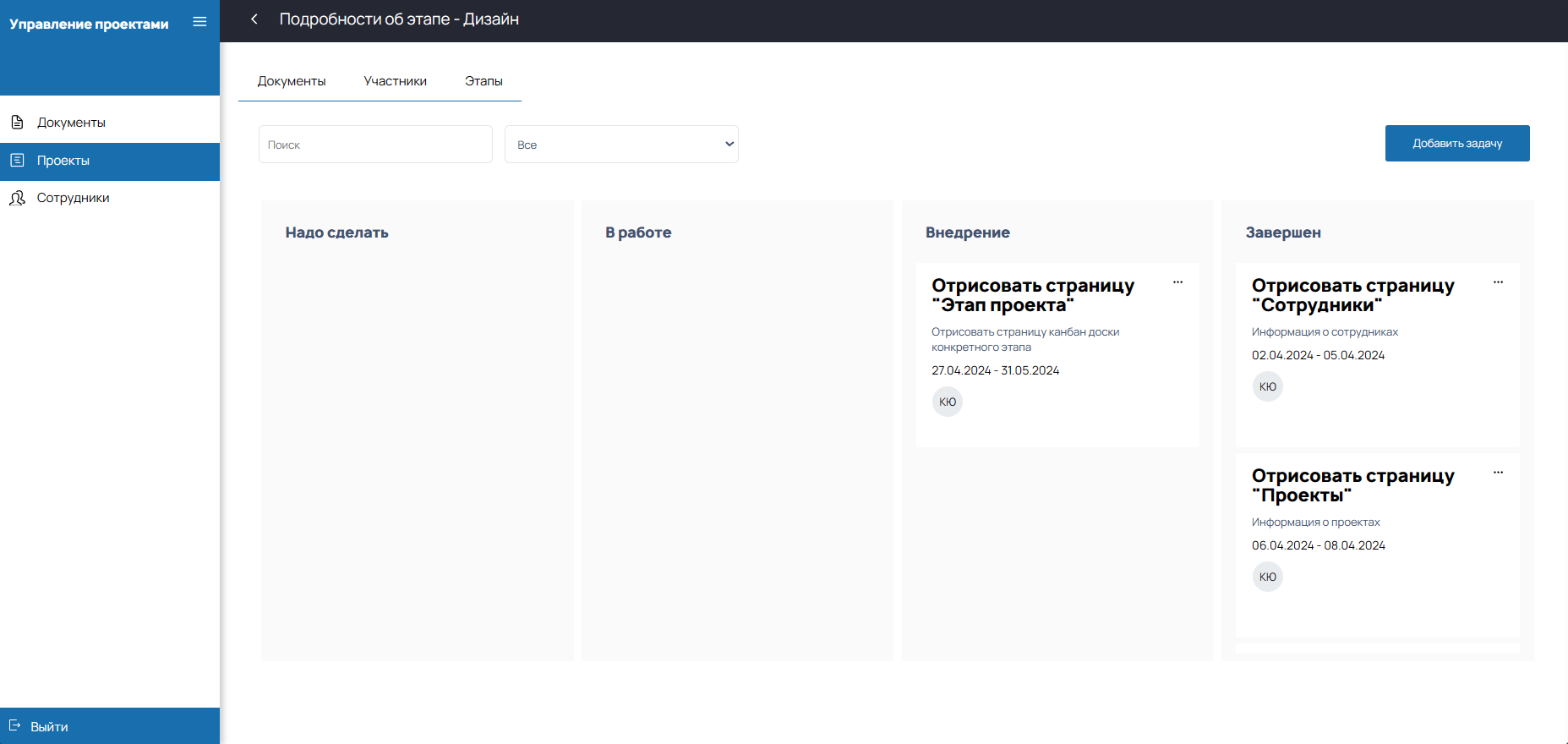


Рисунок 4.15 – Интерфейс страницы подробности этапа проекта

Страница с подробностями этапа проекта, отображаемая на изображении, предоставляет пользователям возможность просмотра информации о выбранном этапе проекта.

На данной странице пользователи могут:

* просматривать информацию о текущем этапе проекта;
* добавлять новые задачи в выбранный этап;
* фильтровать задачи по их статусу;
* осуществлять поиск задач по их названию.

Такой интерфейс обеспечивает удобное взаимодействие с информацией о конкретном этапе проекта, позволяя пользователям эффективно управлять задачами внутри этапа и настраивать их в соответствии с требованиями проекта.

На рисунке 4.16 изображен интерфейс страницы подробности о выбранной задаче.

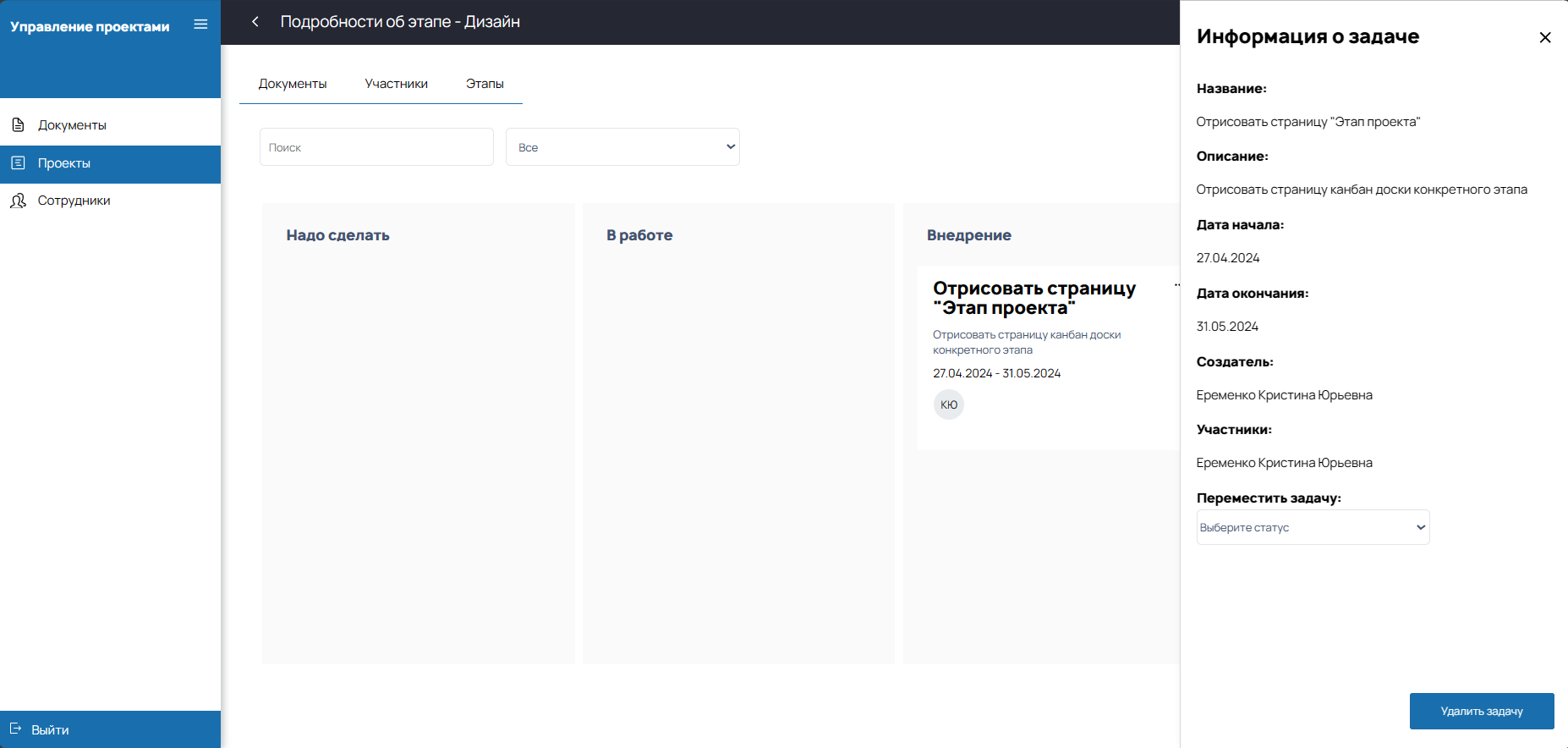


Рисунок 4.16 – Интерфейс страницы подробности о задаче

Страница с подробной информацией о задаче, изображенная на экране, предоставляет пользователям возможность ознакомиться с информацией о выбранной задаче.

На этой странице пользователи могут:

* просматривать подробную информацию о задаче;
* удалять задачу в случае необходимости;
* перемещать задачу между различными статусами в зависимости от ее текущего состояния.

Такой интерфейс обеспечивает удобство в работе с задачами, позволяя пользователям эффективно управлять процессом выполнения задачи и следить за ее статусом в рамках проекта.

На рисунке 4.17 изображено окно добавления задачи.

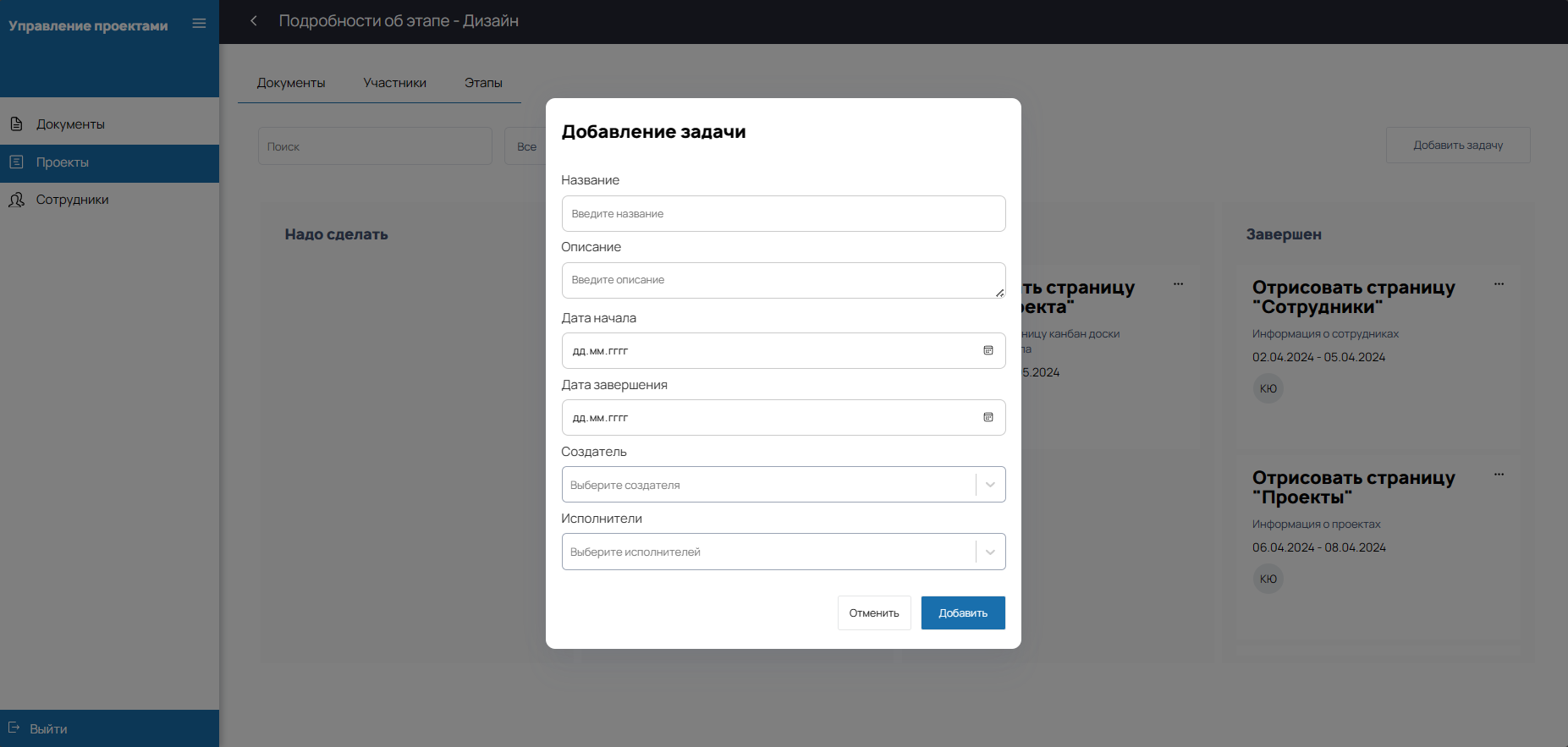


Рисунок 4.17 – Окно добавления задачи

Окно добавления задачи, представленное на изображении, позволяет пользователям создавать новые задачи в выбранном проекте.

## 4.4 Инструкция по эксплуатации программного продукта

Инструкция по эксплуатации программного продукта является ключевым документом, который предоставляется пользователям для ознакомления с функциональностью, использованием и настройкой программного продукта. Вот как может выглядеть инструкция по эксплуатации для корпоративного веб-приложения управления проектами:

При попытке входа в систему с данными, которые не существуют, появляется сообщение об ошибке, которое изображено на рисунке 4.18.

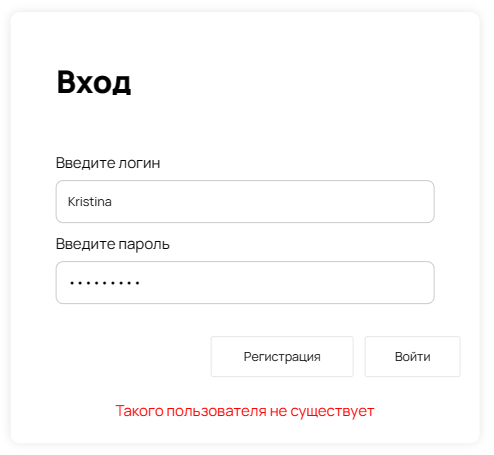


Рисунок 4.18 – Сообщение об ошибке

На рисунке 4.19 будет показано сообщение, которое появится при попытке регистрации нового пользователя с уже существующими данными, указывающее на то, что пользователь с такими данными уже существует.

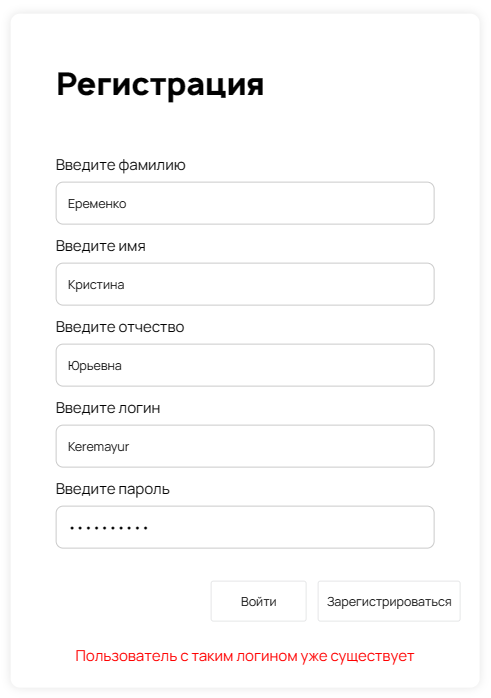


Рисунок 4.19 – Сообщение об ошибке

При попытке входа не заполнив поля, пользователь увидит другие сообщения, изображенные на рисунке 4.20.

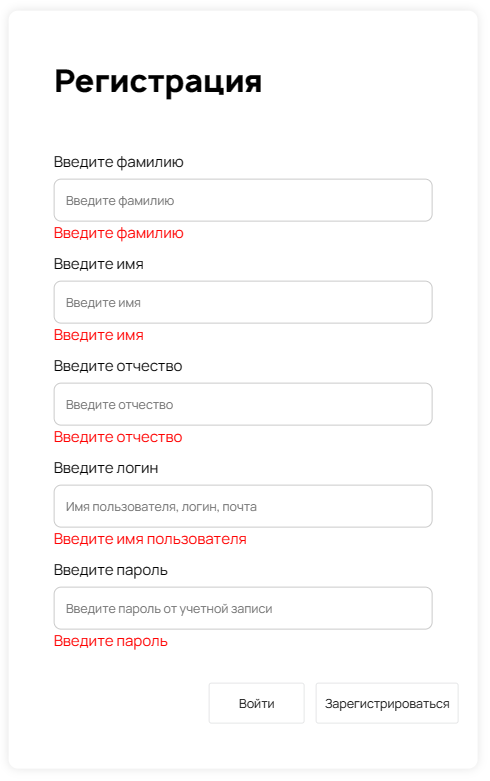


Рисунок 4.20 – Сообщение об ошибке

На рисунке 4.21 изображена фильтрация данных.



Рисунок 4.21 – Фильтрация данных

При использовании данного веб-приложения вы можете воспользоваться функцией фильтрации данных. Для этого перейдите в соответствующий раздел приложения и найдите опцию фильтрации. Здесь вы сможете выбрать различные критерии для фильтрации. После применения выбранных фильтров приложение будет отображать только данные, соответствующие вашим параметрам. Это позволит вам быстро находить необходимую информацию и эффективно управлять вашими проектами.

На изображении 4.22 представлена функция поиска данных.



Рисунок 4.22 – Функция поиска данных

При использовании веб-приложения вы можете воспользоваться функцией поиска данных. Для этого найдите поле поиска в соответствующем разделе приложения. Введите ключевые слова или фразы, которые вы хотите найти, и запустите поиск. Приложение отобразит результаты поиска, соответствующие вашему запросу. Функция поиска поможет вам быстро находить нужную информацию в данных проекта и упростит взаимодействие с приложением.

На изображении 4.23 изображена кнопка для добавления данных.



Рисунок 4.23 – Кнопка добавления данных

Для добавления данных в приложение перейдите в соответствующий раздел, где предусмотрена возможность добавления. Нажмите на кнопку «добавить». Затем заполните необходимые поля или выберите опции в соответствии с вашими требованиями. После завершения заполнения данных нажмите кнопку «Добавить», чтобы добавить информацию в систему. После успешного добавления данных они будут доступны для просмотра и использования в других частях приложения.

На рисунке 4.24 изображена кнопка для редактирования данных.

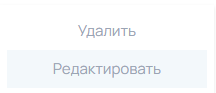


Рисунок 4.24 – Кнопка редактирования данных

Для внесения изменений в данные в приложении перейдите в соответствующий раздел, где предусмотрена функция редактирования. Найдите элемент данных, который вы хотите отредактировать, и выберите опцию «Редактировать». Появится форма с текущей информацией, которую вы можете изменить по своему усмотрению. Внесите необходимые коррективы и нажмите кнопку «Изменить», чтобы сохранить внесенные изменения. После успешного редактирования данные будут обновлены в системе и отображены с учетом внесенных изменений.

На рисунке 4.25 изображена кнопка удаления данных.

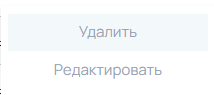


Рисунок 4.25 – Кнопка удаления данных

Для удаления данных из приложения перейдите в соответствующий раздел, где предусмотрена функция удаления. Найдите элемент данных, который вы хотите удалить, и выберите опцию «Удалить». Появится запрос на подтверждение удаления. Подтвердите свое намерение удалить данные, нажав на кнопку «Удалить». После этого выбранные данные будут удалены из системы. Обратите внимание, что операция удаления обычно необратима, поэтому будьте внимательны при удалении данных из приложения.

На рисунке 4.26 изображен запрос на подтверждение удаления.

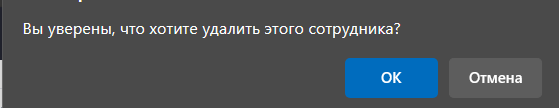


Рисунок 4.26 – Окно подтверждения удаления

На изображении 4.27 показана кнопка, предоставляющая дополнительную информацию.

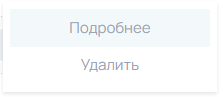


Рисунок 4.27 – Кнопка подробнее

При нажатии на кнопку «Подробнее» пользователь может осуществить переход на страницу с более подробной информацией о выбранном элементе. На этой странице могут быть представлены дополнительные сведения, которые помогут пользователю более полно понять элемент данных или процесс, с которым он работает. Такой подробный просмотр позволяет пользователям получить дополнительную информацию и лучше понять контекст, в котором они работают.

На рисунке 4.28 изображена карточка задачи.

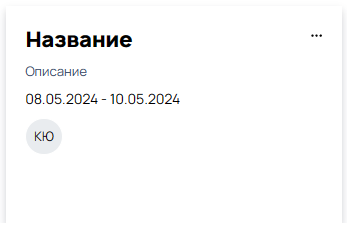


Рисунок 4.28 – Карточка задачи

В карточке задачи присутствует иконка, представленная в виде трех точек, которая предоставляет доступ к дополнительным действиям. При ее нажатии открывается боковая панель, где отображается дополнительная информация о данной задаче. В этой панели можно ознакомиться с основными деталями задачи. Кроме того, в боковой панели можно производить различные действия с задачей, такие как удаление и перемещение задачи по статусам, что позволяет удобно отслеживать ее прогресс и текущее состояние в рамках проекта. Иконка с тремя точками предоставляет удобный способ получения доступа к дополнительной информации и управления задачей прямо из основного интерфейса.

На рисунке 4.29 изображена боковая панель с информацией о конкретной задаче.

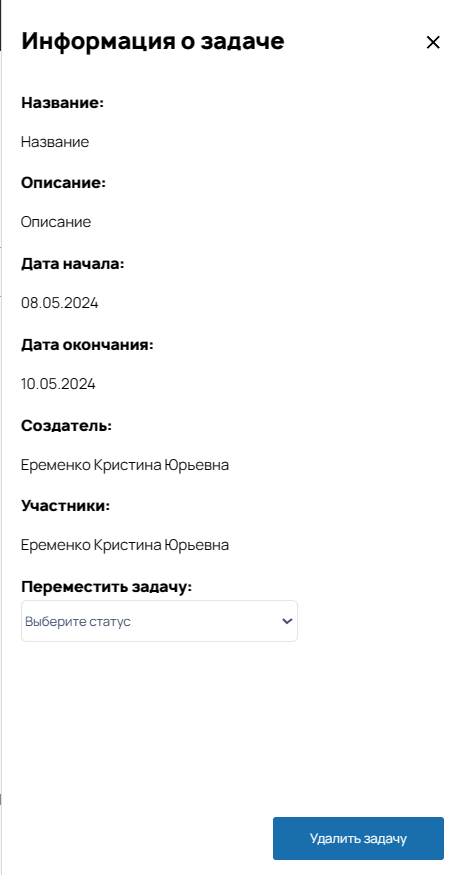


Рисунок 4.29 – Боковая панель

# 5 Экономическая часть

# 6 Охрана труда и техники безопасности

Охрана труда – это способ организации мер, включающих в себя правовые, лечебно-профилактические, реабилитационные, экономические, санитарно-гигиенические, организационно-технические и другие мероприятия направленные на сохранения жизни и здоровья людей в ходе трудовой деятельности, улучшение трудовой дисциплины, развитие их работоспособности, сохранение денежных средств и оборудования предприятия, увеличение прибыли компании. Дело в том, что большинство проблем и неприятностей, в том числе, несчастных случаев на производстве, происходит чаще всего из-за незнания существующих законодательных требований охраны труда [16].

Техника безопасности – система организационных мероприятий и технических средств, направленных на предотвращение воздействия на работающих опасных производственных факторов, являющихся причиной травм или внезапного резкого ухудшения здоровья. Техника безопасности является частью охраны труда и включает такие мероприятия, как обучение и инструктаж работающих по вопросам безопасности труда, поддержание в технически безопасном состоянии зданий и сооружений, оснащение вновь создаваемого и эксплуатируемого производственного оборудования защитными и предохранительными устройствами, разработку средств коллективной и индивидуальной защиты работающих от воздействия опасных и вредных производственных факторов, а также организацию обеспечения этими средствами рабочих и служащих [17].

Охрана труда и техника безопасности – это область, которая заботится о создании безопасной и здоровой рабочей среды для всех работников. Вот основные аспекты охраны труда и техники безопасности:

* оценка рисков: первый шаг в обеспечении безопасности на рабочем месте – это оценка всех потенциальных опасностей и рисков, связанных с выполняемыми задачами. Это включает в себя идентификацию опасных материалов, оборудования и процессов, а также оценку вероятности и последствий возможных несчастных случаев;
* разработка политики безопасности: компании должны разработать и внедрить политику безопасности, которая описывает их обязательства по обеспечению безопасности на рабочем месте. Это может включать в себя требования к использованию средств защиты, процедуры обучения и обучения персонала, а также процедуры для обращения с опасными материалами;
* обучение: работники должны быть обучены правильным методам и процедурам работы, которые гарантируют их безопасность и здоровье. Это может включать в себя обучение по правилам пожарной безопасности, использованию личной защиты, безопасной эксплуатации оборудования и так далее;
* использование средств защиты: работники должны использовать необходимые средства защиты, такие как шлемы, защитные очки, наушники, маски и другие, в соответствии с требованиями безопасности на рабочем месте;
* обслуживание оборудования: оборудование должно регулярно проверяться, обслуживаться и ремонтироваться для обеспечения его безопасной работы. Работники должны быть обучены распознавать потенциальные проблемы с оборудованием и сообщать о них своему руководству;
* соблюдение законодательства: компании должны соблюдать все соответствующие законы и нормативные акты, касающиеся охраны труда и техники безопасности, в том числе стандарты безопасности на рабочем месте, требования к маркировке и обозначению опасных материалов и оборудования.

Обеспечение безопасности на рабочем месте – это общее усилие работодателей, работников и соответствующих органов, и все заинтересованные стороны должны активно участвовать в этом процессе.

Мероприятия по охране труда и ТБ направлены на предотвращение травм работников и исключение ситуаций, следствием которых может стать несчастный случай или авария. При этом на различных предприятиях требования техники безопасности и комплекс необходимых мер могут существенно различаться в связи с отраслевыми особенностями. В целом же можно выделить общие требования. Приведем основные из этих требований.

Охрана труда и техника безопасности на предприятии – это, прежде всего, зона ответственности работодателя и соответствующих служб организации.

Работодатель обязан разработать внутреннюю нормативную документацию, проводить инструктажи и проверки знаний в соответствии с требованиями законодательства, информировать работников обо всех обстоятельствах, от которых зависит безопасность на производстве.

Также работодатель обязан создать для работников безопасные условия труда. Для этой цели предусматривается комплекс требований:

* использование оборудования и конструкций, соответствующих требованиям стандартов и другой нормативной документации;
* соблюдение сроков периодических ремонтов и обслуживания оборудования;
* соблюдение требований пожарной и электробезопасности при оснащении производственных и офисных помещений;
* установка необходимых защитных приспособлений и конструкций;
* обеспечение достаточной освещенности, вентиляции, поддержание оптимального температурного режима на рабочих местах;
* своевременное устранение пыли и отходов производства;
* обеспечение работников специальной одеждой и обувью, а также другими средствами индивидуальной защиты в соответствии со спецификой производства;
* обеспечение работников актуальными инструкциями по ТБ, наглядными материалами;
* создание на рабочих местах и в производственных помещениях всех необходимых систем сигнализации, размещение знаков безопасности и так далее.

# Заключение

В рамках проекта было разработано корпоративное веб-приложение, которое обеспечивает:

* эффективное управление проектами;
* оптимизацию процессов;
* улучшение коммуникации.

Веб-приложение обладает следующими преимуществами:

* удобный и понятный интерфейс: приложение имеет удобный интерфейс, что облегчает его использование и сокращает время обучения сотрудников;
* широкий функционал: приложение обладает широким функционалом, позволяющим эффективно управлять всеми аспектами проекта, включая планирование, создание, и отслеживание выполнения задач;

Предполагаемый экономический и социальный выигрыш от использования веб-приложения:

* сокращение времени и ресурсов: автоматизация рутинных задач с помощью приложения позволяет сотрудникам компании сосредоточиться на важных задачах, что сокращает временные затраты и повышает производительность труда;
* повышение эффективности работы: улучшение контроля над выполнением проектов с помощью приложения сокращает сроки выполнения задач и повышает качество работы, что является ключевым для успеха компании;
* повышение конкурентоспособности: эффективное управление проектами с помощью приложения повышает конкурентоспособность компании, позволяя ей быстрее и качественнее реагировать на изменения и требования клиентов.

В целом, веб-приложение для управления проектами представляет собой ценный инструмент, способный помочь компаниям повысить свою эффективность и конкурентоспособность, обеспечивая более эффективное управление ресурсами и процессами.

# Список использованных источников

Список использованной литературы

1. Вашистов, И. А., & Калугин, В. А. (2020). Проектирование информационных систем: учебное пособие. М.: Издательский центр «Академия».
2. Романовский, И. В. (2018). Основы UML. Разработка программных систем. М.: Издательский дом «Питер».

Интернет ресурсы

1. Диаграмма потоков данных. [Электронный ресурс] – Режим доступа: <https://practicum.yandex.ru/blog/diagramma-potokov-dannyh-dfd/>.
2. Архитектура программного продукта. [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/blog/kak-vybrat-arkhitekturu-po>.
3. Концептуальная модель данных. [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/blog/modelirovanie-dannykh-v-rdbms>.
4. Логическая модель данных. [Электронный ресурс] – Режим доступа: <https://translated.turbopages.org/proxy_u/en-ru.ru.c9d416d2-664b0c09-cd234824-74722d776562/https/en.wikipedia.org/wiki/Logical_schema>.
5. Физическая модель данных. [Электронный ресурс] – Режим доступа: <https://aws.amazon.com/ru/compare/the-difference-between-logical-and-physical-data-model/>.
6. Прототипирование. [Электронный ресурс] – Режим доступа: <https://www.unisender.com/ru/glossary/chto-takoe-prototipirovanie-i-zachem-ono-nuzhno/>.
7. Разработка программного обеспечения. [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/blog/razrabotka-programmnogo-obespecheniia>.
8. TypeScript. [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/blog/preimushchestva-typescript-veb-razrabotka>.
9. Visual Studio Code. [Электронный ресурс] – Режим доступа: <https://translated.turbopages.org/proxy_u/en-ru.ru.f71b2df9-664b1592-ac089bcf-74722d776562/https/code.visualstudio.com/Docs/editor/codebasics>.
10. React.js. [Электронный ресурс] – Режим доступа: <https://ru.react.js.org/docs/getting-started.html>.
11. MongoDB. [Электронный ресурс] – Режим доступа: <https://metanit.com/nosql/mongodb/?ysclid=lwerh3cqgy976902217>.
12. Алгоритм. [Электронный ресурс] – Режим доступа: <https://dzen.ru/a/ZUKwiJ8PzVHI4asg>.
13. Интерфейс программного продукта. [Электронный ресурс] – Режим доступа:<https://old.bigenc.ru/technology_and_technique/text/4426494?ysclid=lwernwq9ig271342600>.
14. Охрана труда. [Электронный ресурс] – Режим доступа: <https://dagmintrud.ru/press/55826?ysclid=lweruu0rqd338562553>.
15. Техника безопасности. [Электронный ресурс] – Режим доступа: <https://studfile.net/preview/4168836/page:9/>.

# Приложение А

(обязательное)

Листинг программы