**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 |  |  |  | А.А. Аникина |
|  |  |
|  |  | (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 «9» января 2024 г. | \_\_\_\_\_\_\_\_ / Н.В. Погорелова / |
|  |  |
| Председатель ЦК \_\_\_\_\_\_\_\_ / И.Н. Борзенкова / |  |

**Задание  
на выполнение выпускной квалификационной работы**

|  |  |  |
| --- | --- | --- |
| Студентки Еременко Кристины Юрьевны | | |
|  |  | |
| Специальность 09.02.07 Информационные системы и программирование | | |
|  |  | |
| 1. Тема: Разработка веб-приложения корпоративного управления проектами для ИП «Ромашов Софт»  утверждена приказом по техникуму №148 от «6» февраля 2024 г. | | |
|  |  | |
| 2. Срок сдачи законченной работы: 10 июня 2024 г. | | |
|  | | |
| 3. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов): | | |
| * технико-экономическая характеристика объекта; * сбор, анализ и формирование требований; * проектирование информационной системы; * разработка программного продукта. | | |
|  | | |
| 4. Перечень графического материала (с точным указанием обязательных чертежей) | | |
|  | | |
|  | | |
| 5. Руководитель: | | преподаватель, Ромашов Р.В. |
|  | | (должность, фамилия, инициалы, подпись) |

|  |  |  |
| --- | --- | --- |
| Задание получила «01» марта 2024 г. |  | / К.Ю. Еременко / |
|  | (подпись) |  |

|  |  |
| --- | --- |
| **Одобрено:** | **Утверждаю:** |
| цикловой комиссией профессионального  информационного цикла | И.о. зам. директора по УР |
| Протокол № 6 от «9» января 2024 г. | \_\_\_\_\_\_\_\_ / Н.В. Погорелова / |
|  |  |
| Председатель ЦК \_\_\_\_\_\_\_\_ / И.Н. Борзенкова / |  |

**Календарный план выполнения  
выпускной квалификационной работы**

СтуденткиЕременко Кристины Юрьевны

Специальность 09.02.07 Информационные системы и программирование

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

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель |  | / Р.В. Ромашов / | «01» марта 2024 г. |
|  | (подпись) |  |  |
|  |  |  |  |
| Студентка |  | / К.Ю. Еременко / | «01» 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 Кристины Юрьевны*

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

**1. Актуальность темы.** *Тема дипломного проекта актуальна, так как веб-приложение для корпоративного управления проектами может помочь ИП «Ромашов Софт» улучшить эффективность и контроль за проектами.*

**2. Оценка содержания дипломного проекта.** *Все разделы дипломного проекта содержат необходимую информацию согласно заданию.*

**3. Качество теоретического и расчетного обоснования принятых в дипломном проекте решений (положительные стороны работы, замечания и недостатки).**

*В проекте хорошо обоснованы принятые решения, с глубоким анализом теории и точными расчетами. Решения соответствуют современным тенденциям и обладают практической значимостью.*

**4. Практическая значимость дипломного проекта.** *Дипломный проект практически значим, так как его программный продукт улучшает управление проектами, повышая эффективность работы предприятия.*

**5. Рекомендации по внедрению результатов дипломного проекта.** *Оценить готовность, обучить персонал, выполнить пробное внедрение с анализом, обеспечить поддержку пользователей.*

**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 Кристины Юрьевны*

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

**1. Актуальность, новизна.** *Тема дипломного проекта актуальна, так как веб-приложение для корпоративного управления проектами может помочь ИП «Ромашов Софт» улучшить эффективность и контроль за проектами.*

**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 управление проектами 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 сотрудникам предприятия эффективно управлять проектами. При определенных доработках он может быть успешно внедрен на предприятии.*

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

|  |  |  |
| --- | --- | --- |
| Рецензент |  | / А.А. Зацепин / |
|  | (подпись) |  |
|  | **М.П.** |  |
| С рецензией ознакомлена |  | / К.Ю. Еременко / |
|  | (подпись) |  |
| **«10» июня 2024 года** |  |  |

Содержание

[Введение 5](#_Toc168146019)

[1 Технико-экономическая характеристика объекта 6](#_Toc168146020)

[1.1 Общие сведения о предприятии 6](#_Toc168146021)

[1.2 Организационная структура 7](#_Toc168146022)

[2 Сбор, анализ и формирования требований к информационной системе 9](#_Toc168146023)

[2.1 Цели и назначение создания автоматизированной системы 10](#_Toc168146024)

[2.2 Характеристика объектов автоматизации 10](#_Toc168146025)

[2.3 Требования к автоматизированной системе 11](#_Toc168146026)

[3 Проектирование информационной системы 12](#_Toc168146027)

[3.1 Разработка и описание сценариев использования информационной системы 12](#_Toc168146028)

[3.2 Потоки данных в информационной системе 14](#_Toc168146029)

[3.3 Выбор архитектуры разрабатываемой информационной системы 16](#_Toc168146030)

[3.4 Проектирование базы данных 17](#_Toc168146031)

[3.5 Формирование основных бизнес-процессов информационной системы 20](#_Toc168146032)

[3.6 Прототипирование пользовательского интерфейса 22](#_Toc168146033)

[4 Разработка программного продукта 30](#_Toc168146034)

[4.1 Средства разработки программного продукта 30](#_Toc168146035)

[4.2 Реализация физической модели данных программного продукта 34](#_Toc168146036)

[4.3 Описание интерфейса программного продукта 36](#_Toc168146037)

[4.4 Инструкция по эксплуатации программного продукта 44](#_Toc168146038)

[5 Экономическая часть 50](#_Toc168146039)

[6 Охрана труда и техники безопасности 54](#_Toc168146040)

[Заключение 57](#_Toc168146041)

[Список использованных источников 58](#_Toc168146042)

[Приложение А 61](#_Toc168146043)

# 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.

Предполагаемый экономический и социальный выигрыш, который может быть получен в результате данной работы, заключается в том, что веб-приложение будет позволять ИП «Ромашов Софт» существенно сократить время и ресурсы, затраченные на управление проектами. Это позволит повысить эффективность работы, 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

ИП «Ромашов Софт» – это компания, специализирующаяся на разработке программного обеспечения и предоставлении широкого спектра услуг в области информационных технологий. Она занимается как созданием программных продуктов, так и поддержкой уже существующих информационных систем. Кроме того, компания активно занимается разработкой сайтов и мобильных приложений, предлагая клиентам инновационные и эффективные решения для различных сфер бизнеса.

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 программного обеспечения;
* техническая поддержка программного обеспечения;
* 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.

Таким образом, предприятие обеспечивает полный цикл разработки и сопровождения программного обеспечения, что позволяет эффективно удовлетворять потребности клиентов и поддерживать высокое качество предоставляемых услуг.

## 1.2 Организационная структура

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

Организационная структура предприятия, отраженная на рисунке 1.1, служит основой для эффективной работы команды.

# 2 Сбор, анализ и формирования требований к информационной системе

В процессе разработки информационной системы ключевым этапом является сбор, анализ и формирование требований. Этот процесс начинается с установления контакта с заказчиком и другими заинтересованными сторонами, чтобы полноценно понять их потребности и ожидания. Основные этапы процесса:

* сбор требований включает в себя встречу с заказчиком и заинтересованными сторонами для выявления их потребностей и ожиданий, а также получение полного списка функциональных и нефункциональных требований, описывающих желаемое поведение системы и её характеристики;
* при анализе требований проводится систематизация и классификация их по приоритетности, проверяется их соответствие бизнес-целям компании, а также выявляются и устраняются противоречия или недостатки в требованиях;
* в процессе формирования требований осуществляется создание документа, включающего в себя функциональные и нефункциональные требования, описание пользовательских сценариев и интерфейсов, а затем его утверждение с заказчиком и заинтересованными сторонами;
* в дополнение к основным этапам, предусмотрены следующие дополнительные шаги: проведение обсуждений и презентаций для уточнения требований и получения обратной связи, внесение корректировок на основе этой обратной связи, а также обеспечение полного понимания требований всеми участниками проекта.

В результате этого процесса формируется документ, который является основой для последующей разработки веб-приложения. Этот документ включает в себя подробное описание функциональных и нефункциональных требований, пользовательских сценариев и интерфейсов. Он служит руководством для создания архитектуры приложения, проектирования интерфейса и реализации функциональности, которая соответствует потребностям компании в управлении проектами.

## 2.1 Цели и назначение создания автоматизированной системы

Цели и назначение проекта представляют собой важные аспекты, определяющие его стратегическую направленность и ценность.

Цели проекта:

* разработка инструмента для эффективного управления проектами с минимальными временными и ресурсными затратами;
* оптимизация процессов планирования, выполнения и мониторинга проектов;
* улучшение обмена информацией между участниками проектов и сокращение времени на передачу данных, связанных с выполнением задач.

Назначение:

* создание централизованного хранилища данных с упрощенным доступом для всех участников проекта;
* автоматизация управленческих процессов, включая создание, назначение и отслеживание задач и этапов проектов;
* управление ресурсами и задачами, включая возможность контроля за списком задач и их выполнением.

В целом, проект стремится к созданию системы, которая значительно упростит процессы управления проектами, повысит их эффективность и снизит нагрузку на участников проектов. Предложенное назначение проекта ориентировано на улучшение рабочего процесса и обеспечение более эффективного использования ресурсов.

## 2.2 Характеристика объектов автоматизации

Объекты автоматизации включают в себя следующие основные компоненты и процессы, которые будут охвачены разработанной автоматизированной системой:

* создание и удаление проектов, определение их этапов, сроков выполнения и назначения ответственных лиц, а также мониторинг прогресса выполнения проектов и их этапов;
* постановка задач для сотрудников, контроль за их выполнением и отслеживание сроков и статусов задач;
* сбор и хранение личных данных сотрудников, а также обновление информации о них, их должностных обязанностях и статусе активности.

Разработанная система автоматизации упростит управление проектами и персоналом, повысит эффективность работы и сократит временные затраты.

## 2.3 Требования к автоматизированной системе

Задачи веб-приложения для корпоративного управления:

* аутентификация и авторизация, которые обеспечивают безопасный доступ, проверяя учетные данные пользователей и предоставляя доступ к системе в зависимости от их ролей;
* управление проектами, которое включает создание проектов, определение этапов и контроль их выполнения, назначение ответственных лиц и участников, а также мониторинг прогресса и статусов этапов;
* управление задачами и отслеживание их выполнения, предусматривающее постановку задач сотрудникам и контроль их выполнения;
* управление данными о сотрудниках, включающее сбор, хранение и обновление информации о сотрудниках, таких как личные данные, должностные обязанности и статус активности.

В целом, разработанное веб-приложение предоставляет инструменты для эффективного управления проектами и персоналом, обеспечивая безопасность и прозрачность процессов в корпоративной среде.

# 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 [6].

## 3.1 Разработка и описание сценариев использования информационной системы

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 [7].

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 выполняет следующие функции:

* создание, редактирование и просмотр задач, связанных с проектами. Пользователь может помечать задачи как выполненные и отслеживать их выполнение;
* просмотр информации о проектах, включая данные о самом проекте и списки задач, которые необходимо выполнить в рамках данного проекта.

Администратор выполняет следующие функции:

* контроль деления проектов на этапы, определение задач, сроков выполнения и конечных результатов;
* создание, редактирование и удаление пользователей, а также определение ролей пользователей с установкой соответствующих прав доступа и привилегий;
* осуществление контроля и управления всеми аспектами проекта, включая создание новых проектов, изменение параметров, удаление проектов и 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. Основные типы потоков данных в информационной системе:

* внешние сущности, такие как пользователи, другие системы и клиенты, взаимодействующие с системой;
* процессы, представляющие собой последовательность действий, которые система выполняет для достижения определенной цели. В процессах также могут участвовать внешние сущности. Потоки данных внутри процессов обеспечивают передачу информации между этапами процесса и взаимодействие с внешними сущностями;
* потоки данных, описывающие передачу информации между внешними сущностями, процессами и хранилищем данных. Эти потоки обеспечивают передачу данных от одного узла системы к другому, обеспечивая необходимую информационную связь;
* хранилище данных, где информация сохраняется и доступна для использования.

Эти компоненты взаимодействуют между собой, создавая структуру, которая позволяет эффективно обрабатывать, хранить и передавать данные. Взаимодействие между внешними сущностями, процессами, потоками данных и хранилищами данных обеспечивает поддержку бизнес-процессов и достижение целей организации [11].

На рисунке 3.2 изображена диаграмма потоков данных.

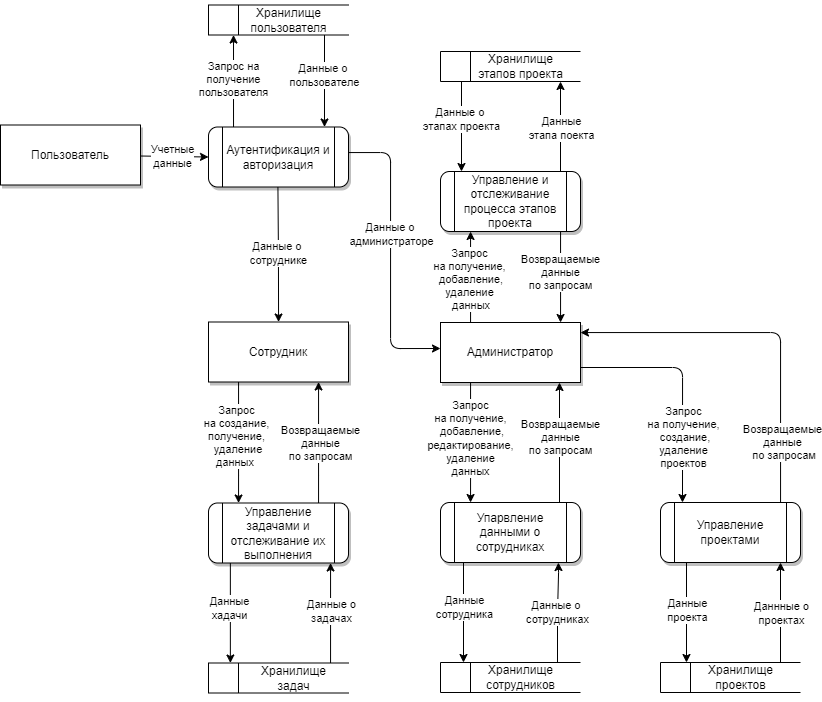


Рисунок 3.2 – Диаграмма потоков данных

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 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 разрабатываемой информационной системы

Архитектура программного продукта – это набор ключевых решений и принципов, определяющих структуру, взаимодействие и поведение компонентов программного обеспечения [12].

Архитектура клиент-серверного приложения 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.

На рисунке 3.3 представлена схема клиент-сервер.



Рисунок 3.3 – Схема клиент-сервер

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 программного продукта, основанная на принципах клиент-серверного подхода, представляет собой эффективный способ организации приложения, где клиентская часть обеспечивает интерфейс пользователя, а серверная часть отвечает за обработку данных и бизнес-логику. Использование модульного монолита для серверной части позволяет легко масштабировать и управлять приложением, обеспечивая высокую производительность и надежность.

## 3.4 Проектирование базы данных

Концептуальная модель данных – это абстрактное представление о данных в информационной системе, которое описывает сущности и взаимосвязи между ними. Она служит основой для проектирования базы данных, представляя собой высокоуровневую картину о данных, которые будут использоваться в системе [13].

На рисунке 3.4 изображена концептуальная модель базы данных.



Рисунок 3.4 – 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 [8].

Объектами в данной базе данных будут данные, содержащиеся в таблицах 3.1-3.13.

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 сотрудников

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

Таблица 3.11 – Атрибуты статусов сотрудника

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

Таблица 3.12 – Атрибуты должностей

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

Таблица 3.13 – Атрибуты подразделений

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

На рисунке 3.5 представлена логическая модель данных.

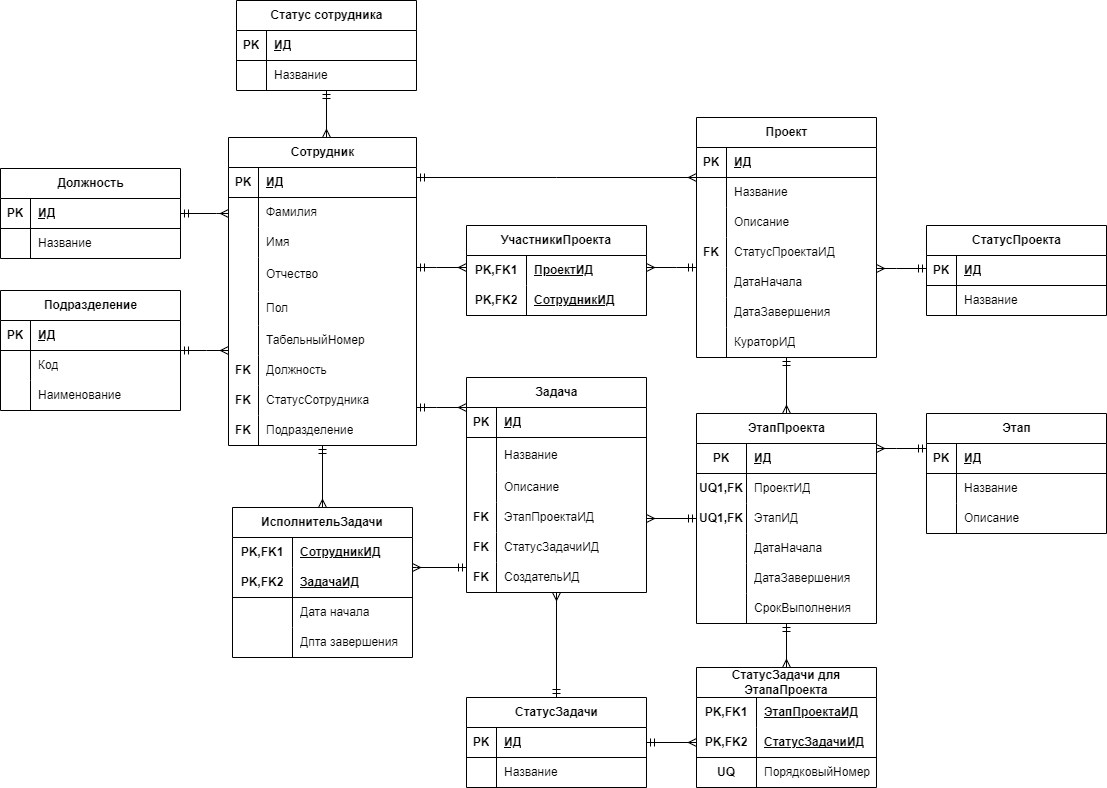


Рисунок 3.5 – Логическая модель данных

## 3.5 Формирование основных бизнес-процессов информационной системы

Методология моделирования бизнес-процессов представляет собой стандарт для графического отображения бизнес-процессов. Она создана для упрощения понимания и анализа процессов в различных организациях, а также для улучшения коммуникации между бизнес-аналитиками, разработчиками и менеджерами. Этот стандарт применяется для описания шагов, участников, действий и решений, которые происходят при выполнении бизнес-процессов [10].

На рисунке 3.6 изображена диаграмма модели и нотации бизнес-процессов, иллюстрирующая процесс авторизации.

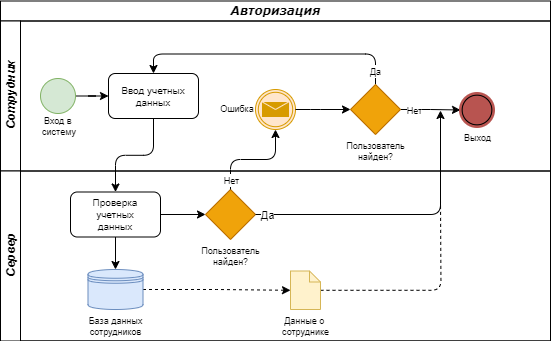


Рисунок 3.6 – Процесс авторизации

Диаграмма модели и нотации бизнес-процессов представляет процесс авторизации сотрудника в системе. Эта диаграмма иллюстрирует пошаговый процесс авторизации, демонстрируя взаимодействие между сотрудником и сервером, а также возможные сценарии возникновения ошибок и их обработку.

На рисунке 3.7 изображена диаграмма модели и нотации бизнес-процессов, иллюстрирующая процесс выполнения задачи.

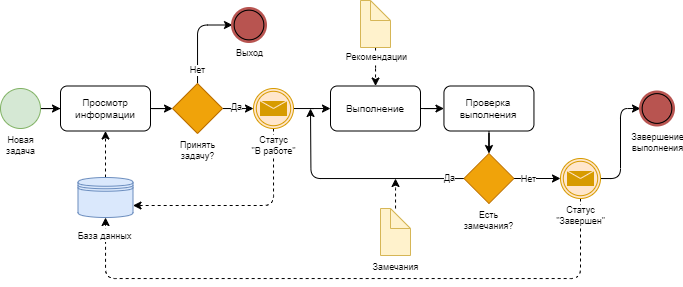


Рисунок 3.7 – Процесс выполнения задачи

Данная диаграмма подробно описывает все последовательные шаги выполнения задачи, начиная с получения новой задачи и заканчивая ее завершением. Она демонстрирует различные возможные сценарии, включая принятие решений о начале работы над задачей, процесс выполнения, проверку результатов, а также обработку и исправление замечаний, если таковые имеются.

## 3.6 Прототипирование пользовательского интерфейса

Прототипирование представляет собой создание набора черновых макетов или схем, которые используются в процессе разработки веб-сайтов или приложений. Этот этап разработки позволяет визуализировать структуру и расположение элементов интерфейса до начала полноценного проектирования, и кодирования.

Прототипы включают в себя только базовые элементы интерфейса, такие как блоки контента, кнопки, поля ввода, навигационные панели и другие функциональные компоненты. Важно отметить, что на этом этапе не уделяется внимание дизайну и декоративным элементам, таким как цвета, шрифты, изображения и стили. Основная цель прототипирования заключается в создании понятного и функционального каркаса будущего продукта, который поможет определить, как различные части интерфейса будут взаимодействовать между собой и с пользователями [14].

На рисунке 3.8 изображен прототип авторизации.

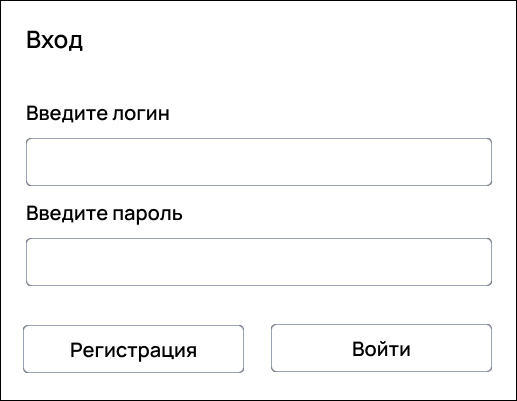


Рисунок 3.8 – Прототип авторизации

Прототип авторизации позволяет пользователям входить в систему.

На рисунке 3.9 изображен прототип регистрации.

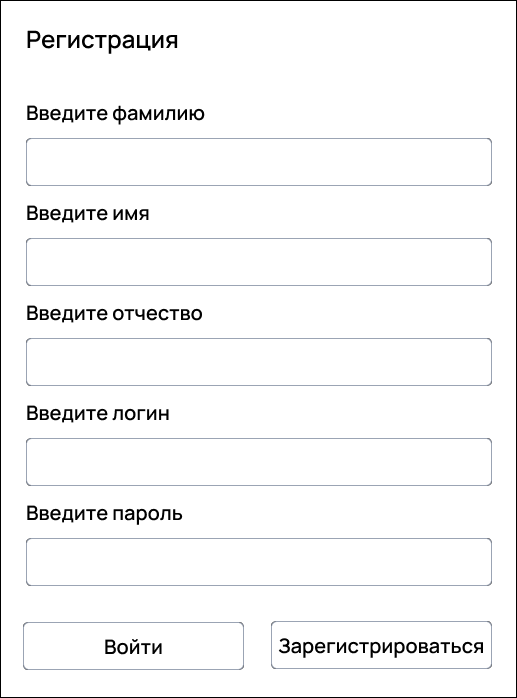


Рисунок 3.9 – Прототип регистрации

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

На рисунке 3.10 изображен прототип страницы сотрудников.

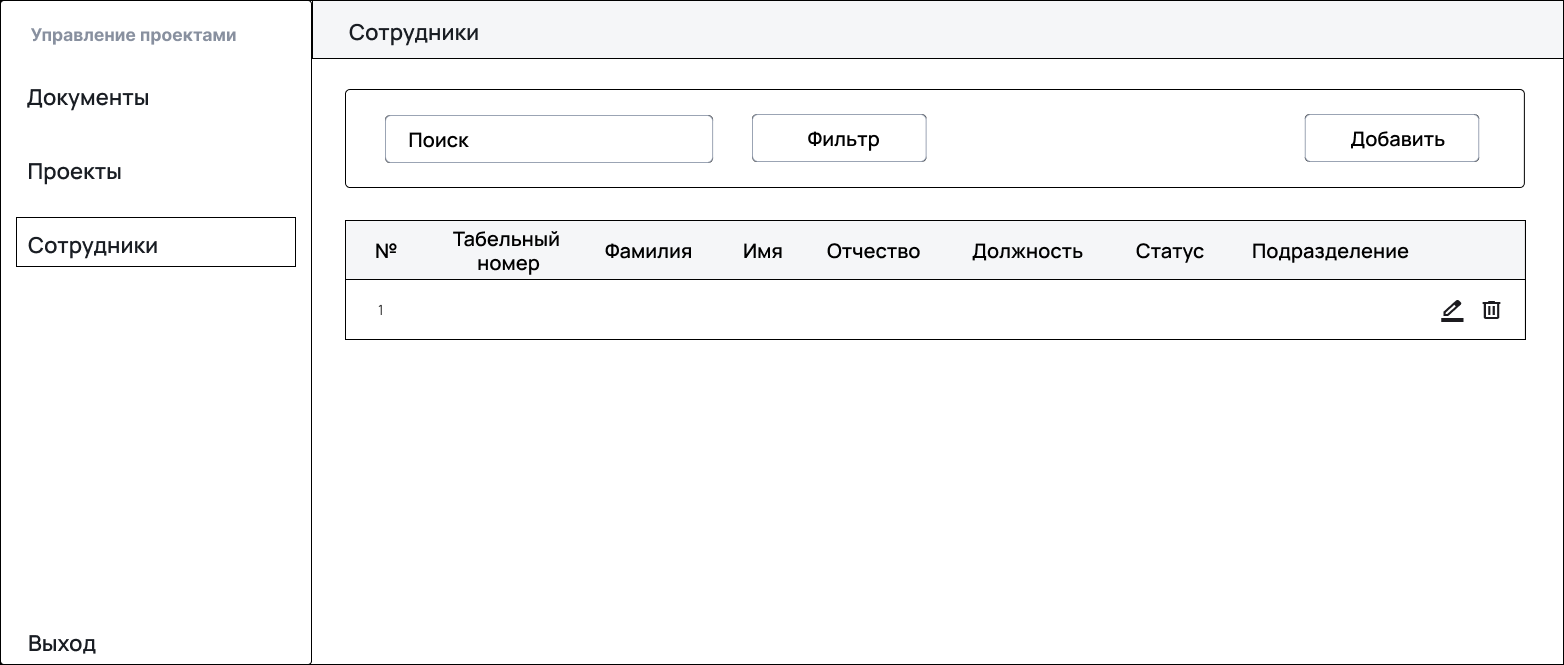


Рисунок 3.10 – Прототип страницы сотрудников

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 3.11 изображен прототип формы добавления сотрудника.

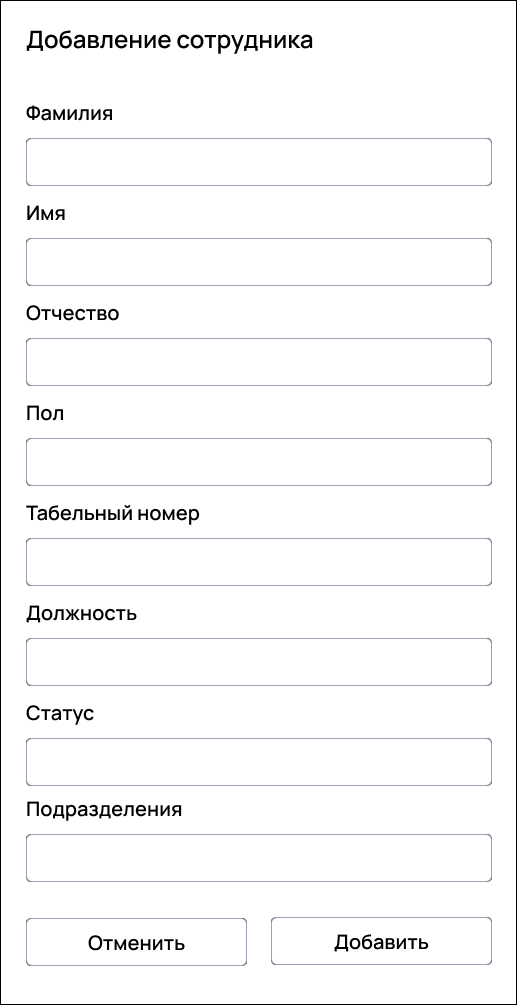


Рисунок 3.11 – Прототип формы добавления сотрудника

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 3.12 изображен прототип страницы проектов.

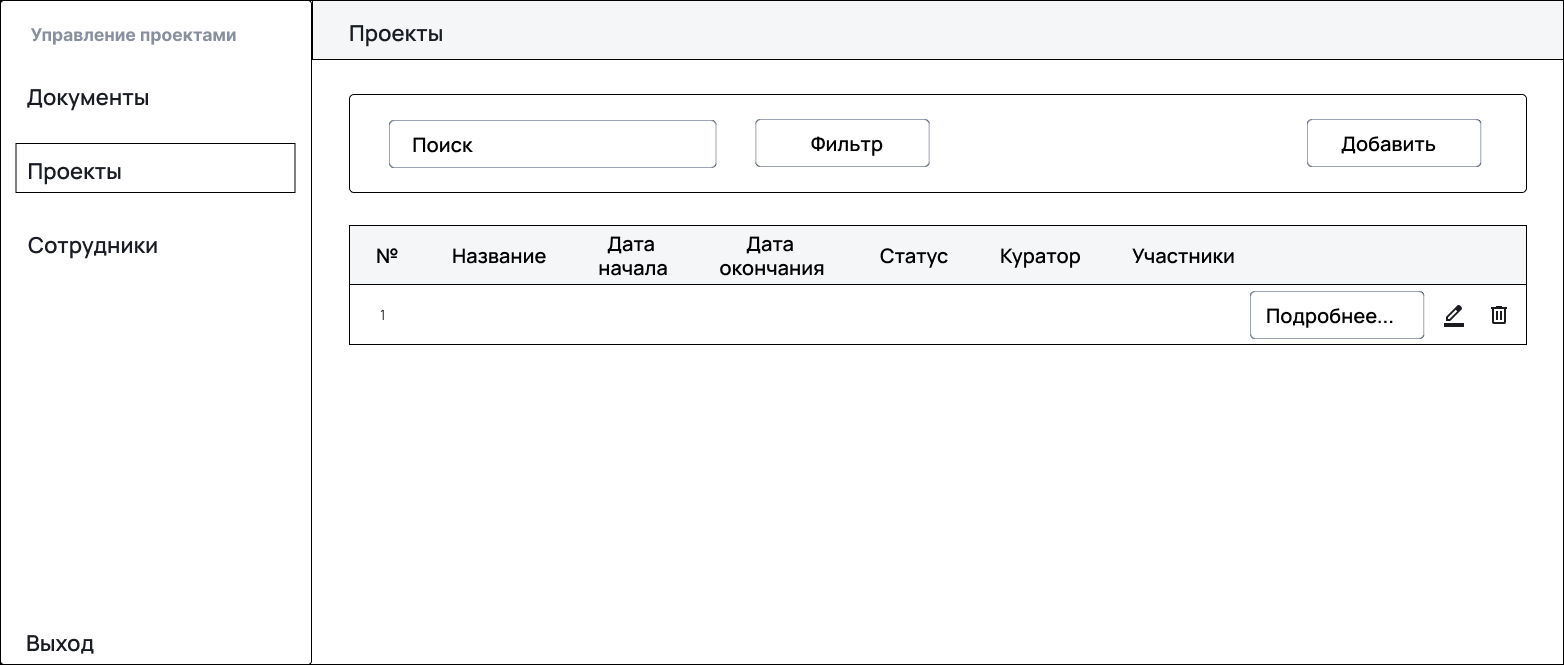


Рисунок 3.12 – Прототип страницы проектов

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 3.13 изображен прототип формы добавления проекта.

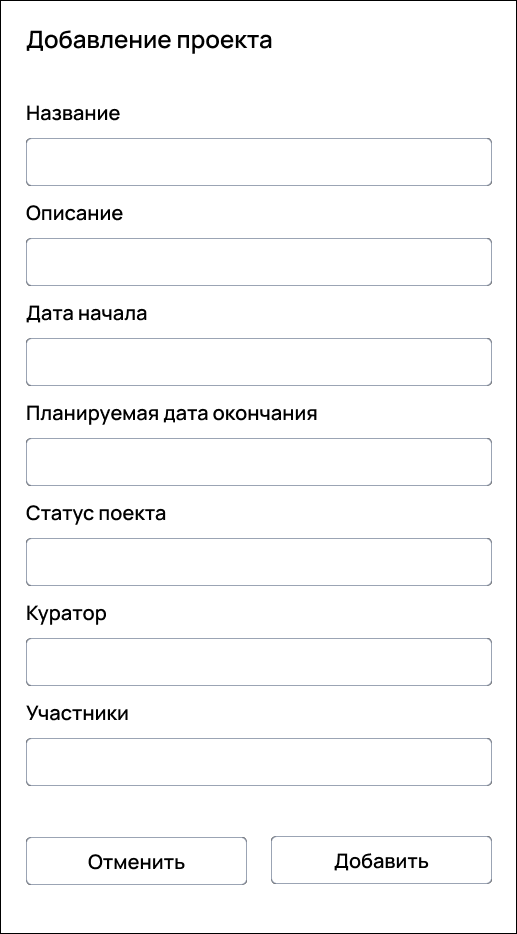


Рисунок 3.13 – Прототип формы добавления проекта

Прототип формы для добавления проекта предоставляет удобный и функциональный пользовательский интерфейс, направленный на упрощение и оптимизацию процесса создания новых проектов. Этот прототип демонстрирует хорошо структурированный дизайн, который включает в себя интуитивно понятные поля ввода и ясные инструкции, что значительно облегчает пользователям процесс ввода необходимых данных.

На рисунке 3.14 изображен прототип страницы участников проекта.

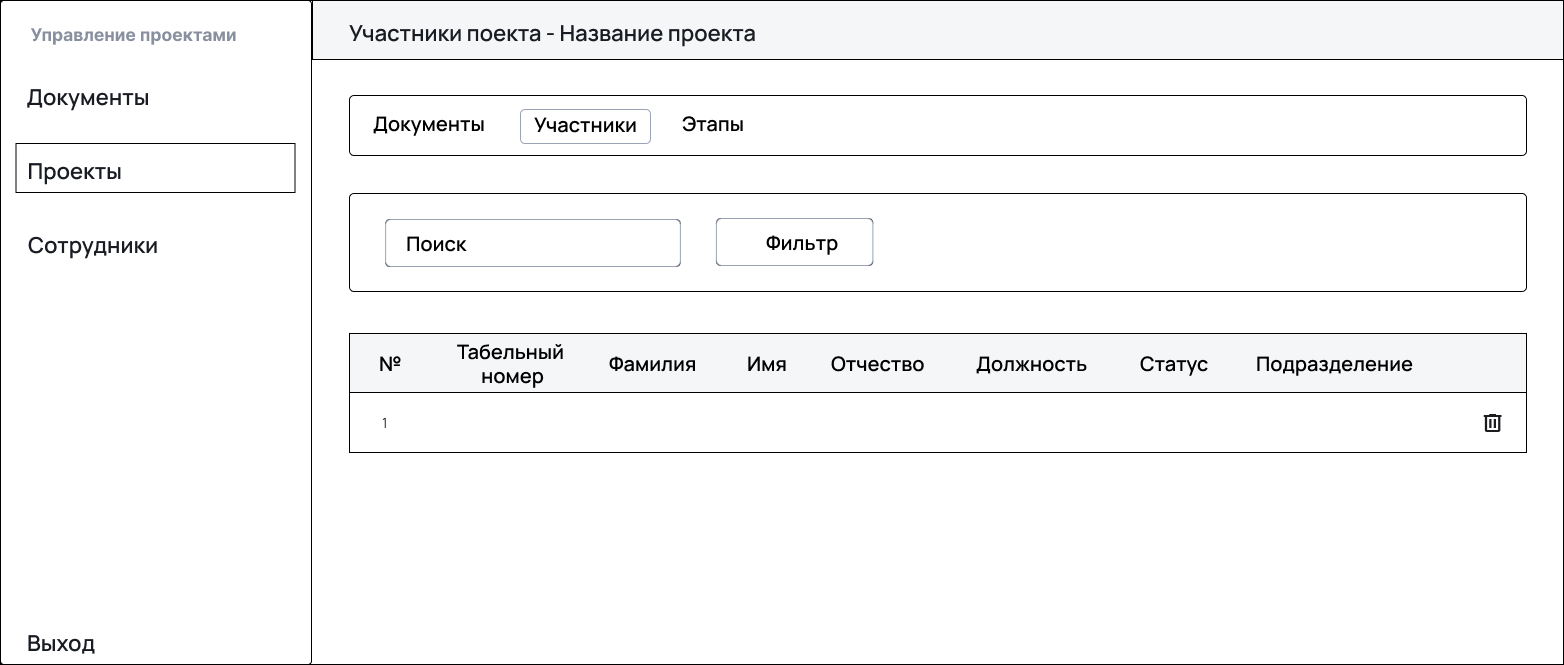


Рисунок 3.14 – Прототип страницы участников проекта

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 3.15 изображен прототип страницы этапов проекта.

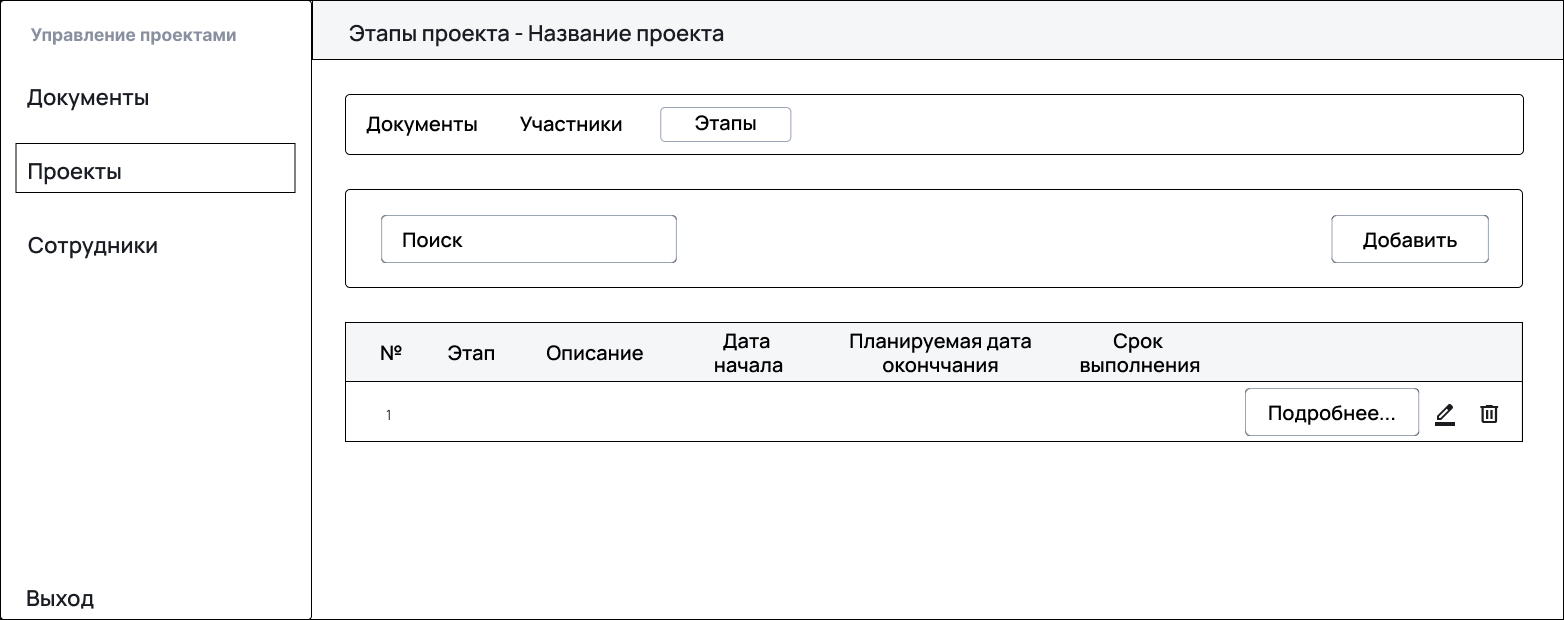


Рисунок 3.15 – Прототип страницы этапов проекта

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 3.16 изображен прототип формы добавления этапа проекта.

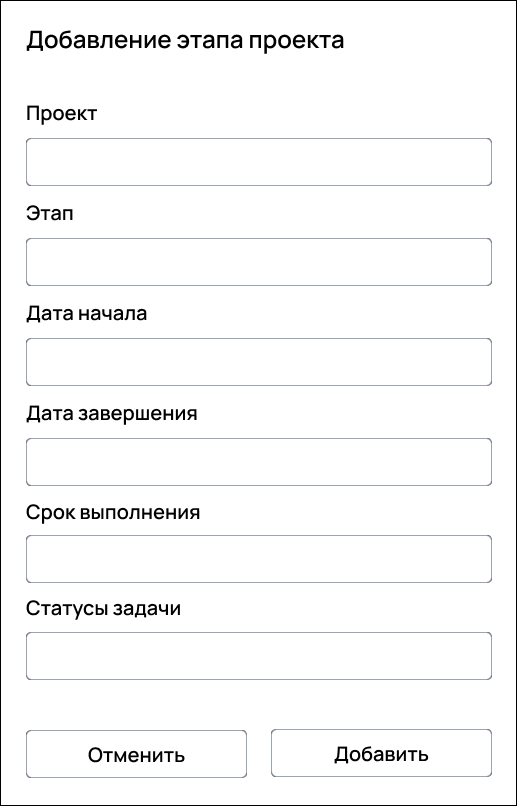


Рисунок 3.16 – Прототип формы добавления этапа проекта

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 3.17 изображен прототип страницы подробности об этапе.

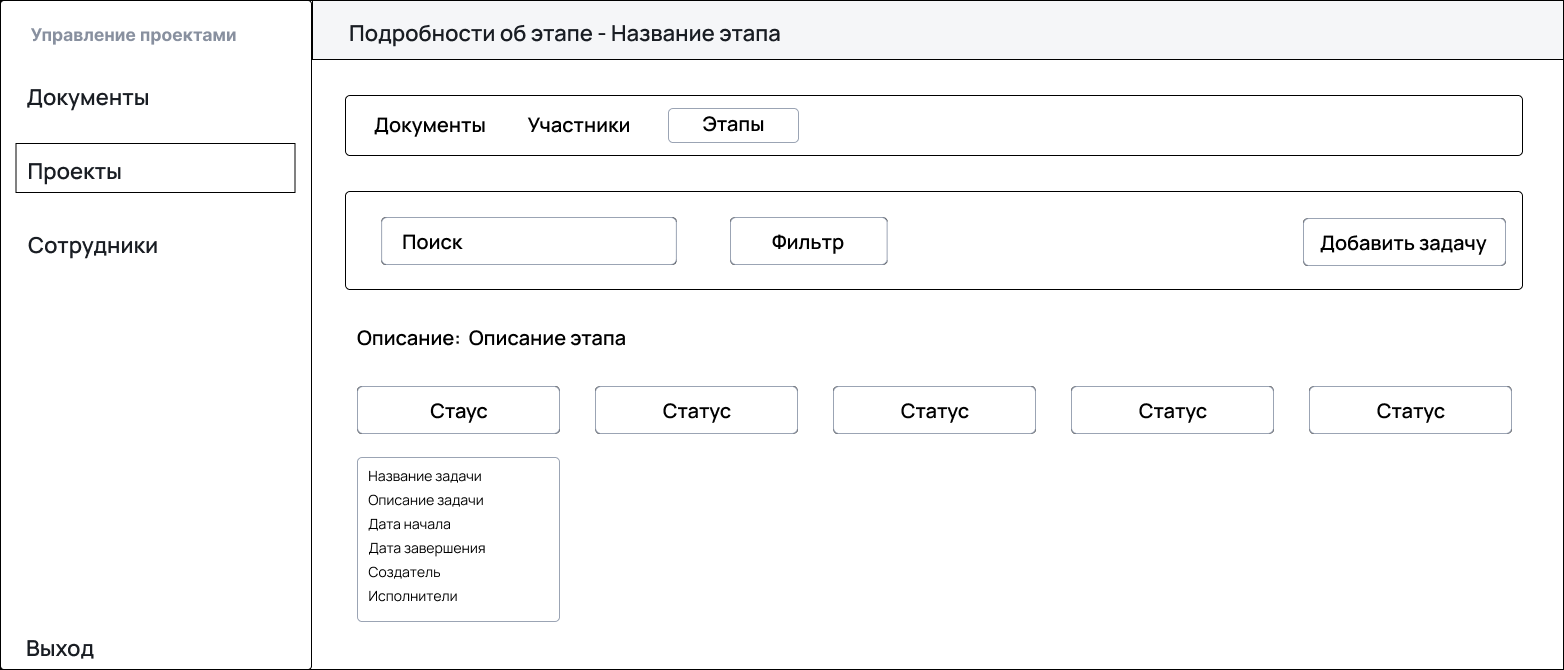


Рисунок 3.17 – Прототип страницы подробности об этапе

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 3.18 изображен прототип формы добавления задачи.

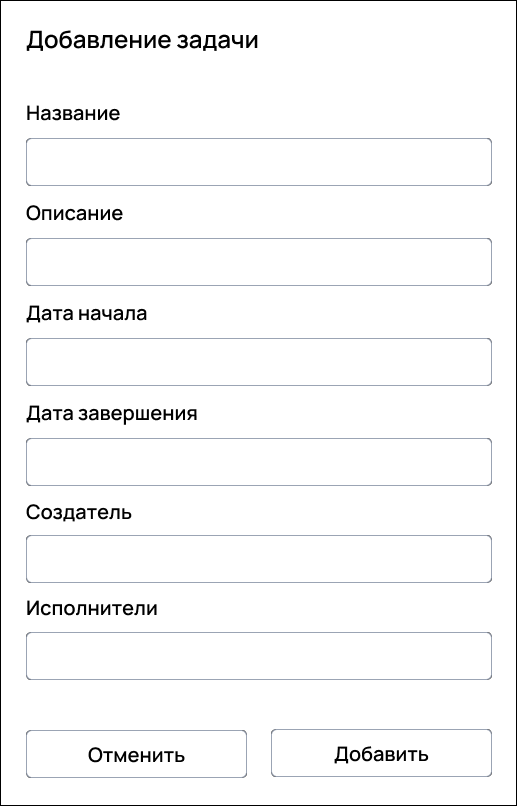


Рисунок 3.18 – Прототип формы добавления задачи

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 3.19 изображен прототип информации о задаче.

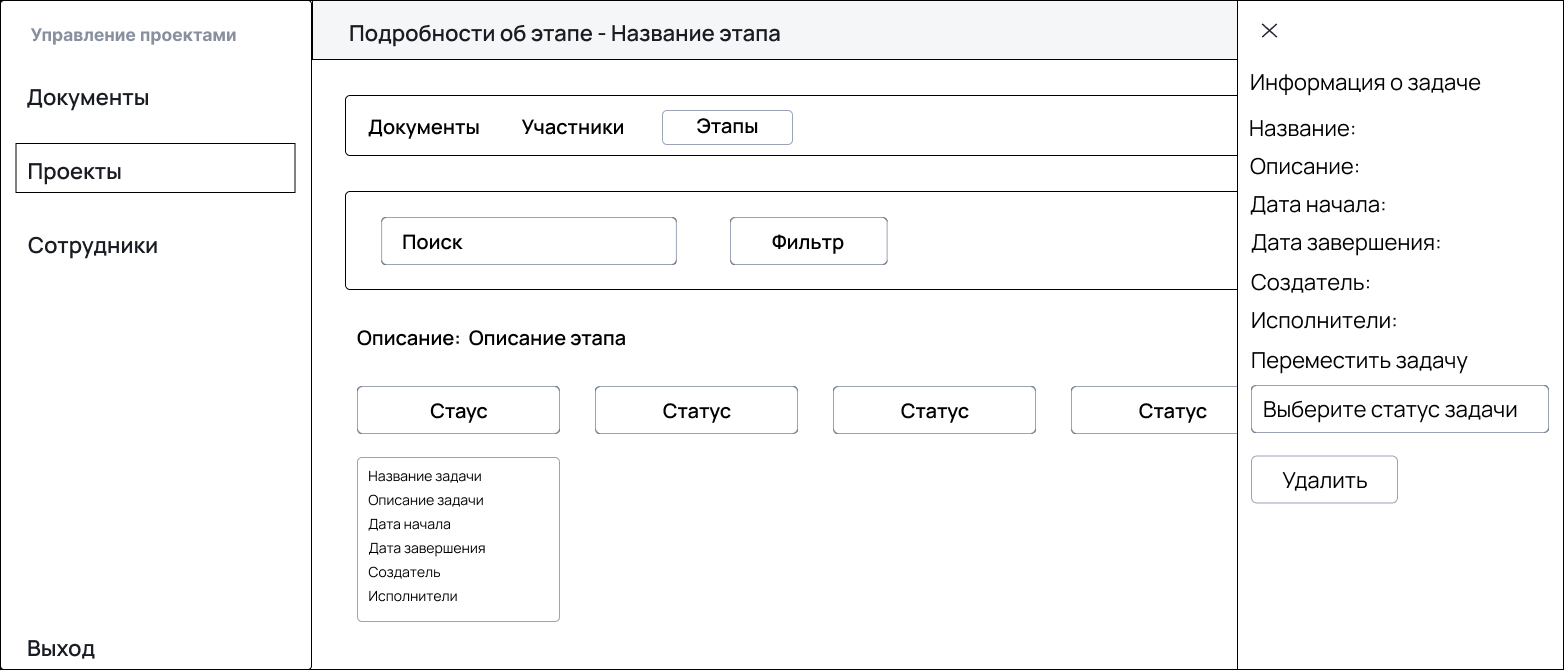


Рисунок 3.19 – Прототип информации о задаче

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 продукта

Разработка программного продукта – 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 продукта 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 продукта – 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 Средства 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 [1].

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 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;
* 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;
* 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;
* 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;
* 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;
* 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;
* редактор кода поддерживает операционные системы Windows, macOS и Linux, что делает его универсальным инструментом для разработки на различных платформах;
* 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 [15].

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:

* 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;
* 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;
* 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 – 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;
* 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 [2].

Node.js этот инструмент применяется для создания серверной части приложения. Node.js обеспечивает высокую производительность и масштабируемость [3].

Express.js является минималистичным и гибким фреймворком для создания веб-приложений на Node.js. Он обеспечивает удобный механизм для обработки маршрутов и запросов [4].

## 4.2 Реализация физической модели данных программного продукта

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 [8].

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:

* 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;
* 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;
* 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;
* 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;
* 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 [5].

На рисунке 4.1 представлена физическая модель данных, которая реализована с помощью MongoDB.

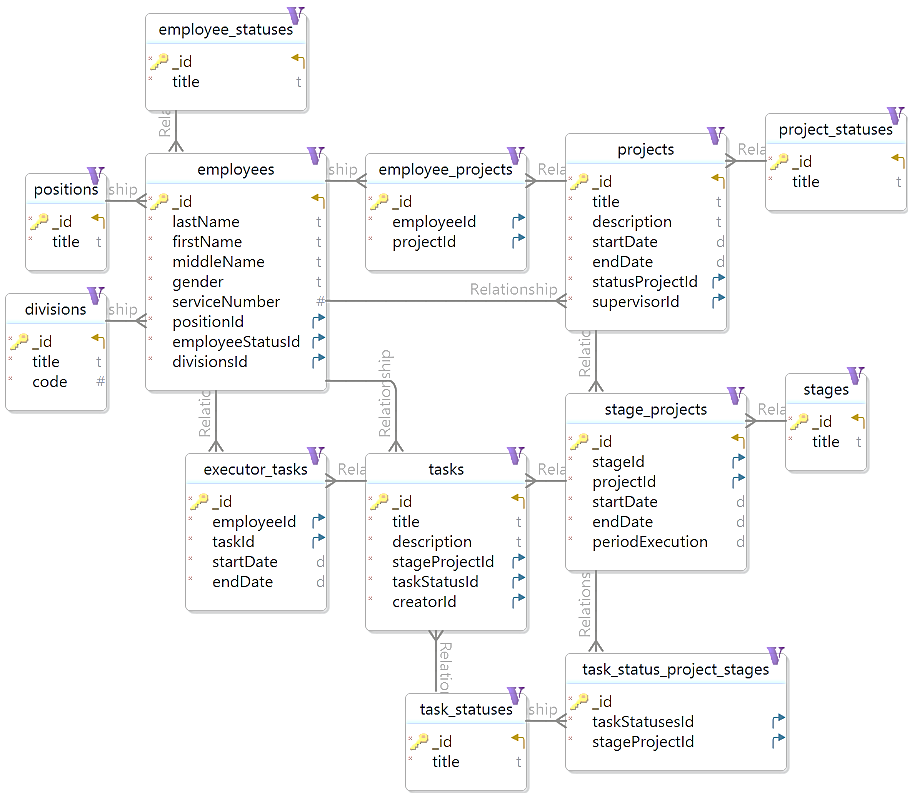


Рисунок 4.1 – Физическая модель

На рисунке представлена физическая модель данных, разработанная для системы управления проектами и задачами. Эта модель включает следующие основные таблицы и их взаимосвязи: таблицу сотрудников, должностей, подразделений, статусов сотрудников, проектов, статусов проектов, связи сотрудников с проектами, задач, связи сотрудников с задачами, статусов задач, этапы проектов, этапы и связи статусов задач с этапами проектов.

## 4.3 Описание интерфейса программного продукта

Интерфейс программного продукта представляет собой комплекс элементов и функций, разработанных для обеспечения взаимодействия пользователя с программным обеспечением. Он включает в себя различные компоненты, такие как кнопки, меню, окна, и другие интерактивные элементы, которые позволяют пользователю вводить данные, получать информацию и управлять приложением. Основная задача интерфейса заключается в том, чтобы сделать взаимодействие с программой интуитивно понятным и удобным, обеспечивая легкий доступ ко всем функциональным возможностям продукта [16].

На рисунке 4.2 представлен интерфейс авторизации.

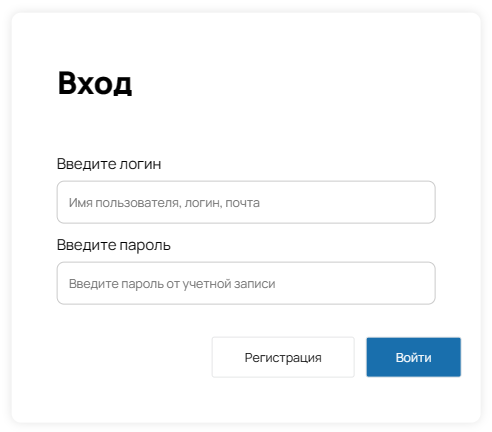


Рисунок 4.2 – Интерфейс авторизации

Интерфейс авторизации на изображении представляет собой форму входа в систему, где пользователю предлагается ввести свои учетные данные для доступа к аккаунту.

На рисунке 4.3 представлен интерфейс регистрации.

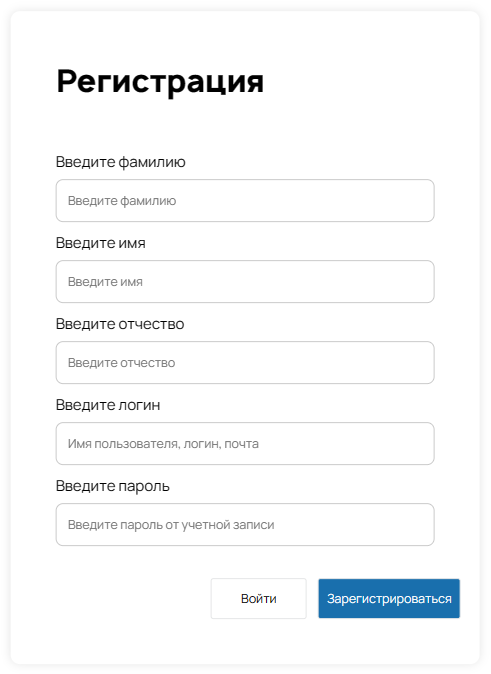


Рисунок 4.3 – Интерфейс регистрации

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

На рисунке 4.4 представлен интерфейс страницы сотрудники.

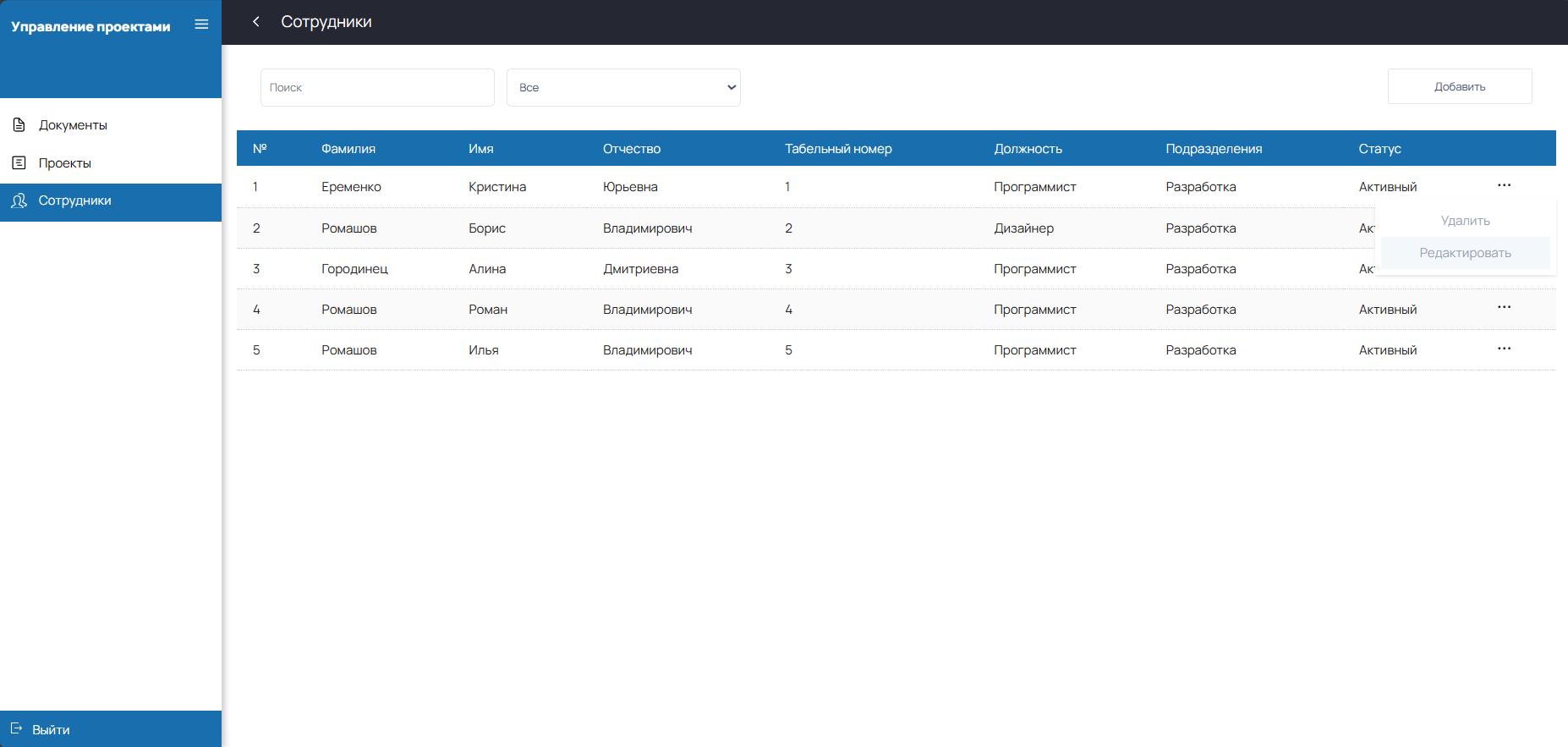


Рисунок 4.4 – Интерфейс страницы сотрудников

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 4.5 изображено окно добавления сотрудника.

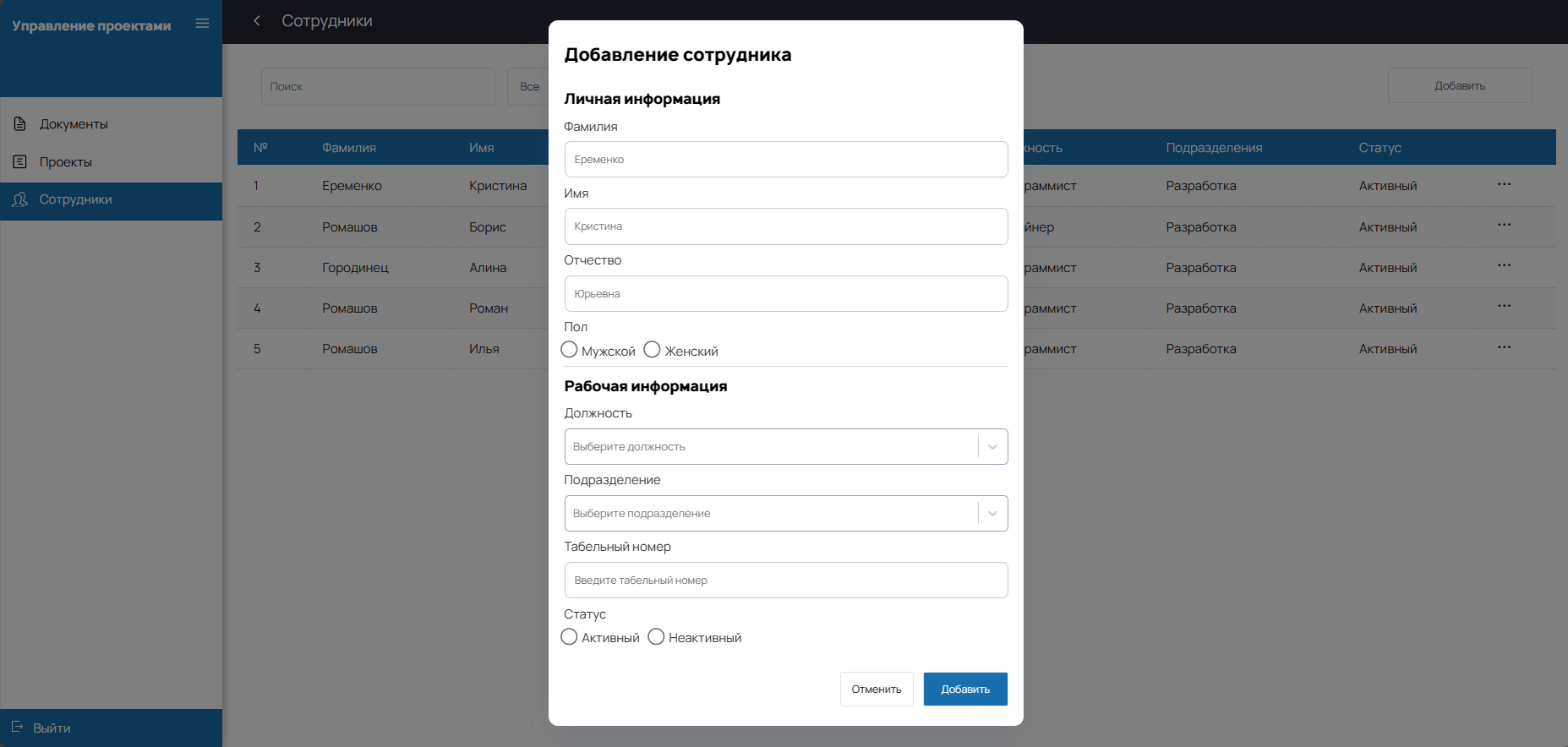


Рисунок 4.5 – Окно добавления сотрудника

На изображении представлен интерфейс добавления нового сотрудника в систему управления проектами. Этот интерфейс оформлен в виде модального окна, которое накладывается поверх основной страницы. В модальном окне содержится форма, разделенная на два основных блока: «Личная информация» и «Рабочая информация». Этот интерфейс предназначен для удобного и быстрого добавления новых сотрудников, обеспечивая интуитивно понятный способ ввода необходимых данных и управления персоналом в системе управления проектами.

На рисунке 4.6 изображено окно редактирования сотрудника.

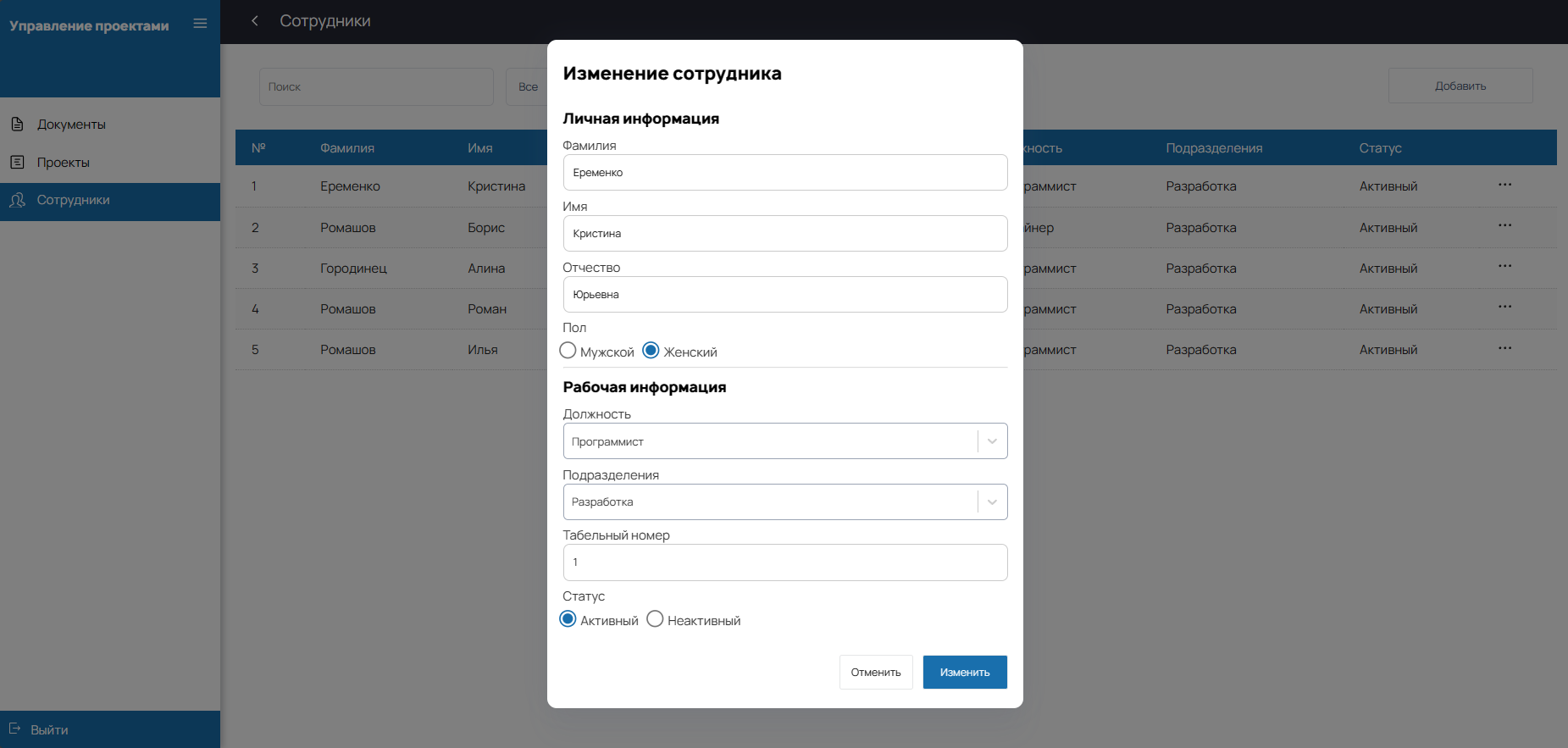


Рисунок 4.6 – Окно редактирования сотрудника

Интерфейс редактирования информации о сотруднике, изображенный на рисунке, предоставляет пользователю возможность вносить изменения в данные о существующем работнике в системе. Этот интерфейс оформлен в виде модального окна, которое накладывается поверх основной страницы. В модальном окне содержится форма, разделенная на два основных блока: «Личная информация» и «Рабочая информация».

На рисунке 4.7 представлен интерфейс страницы проектов.

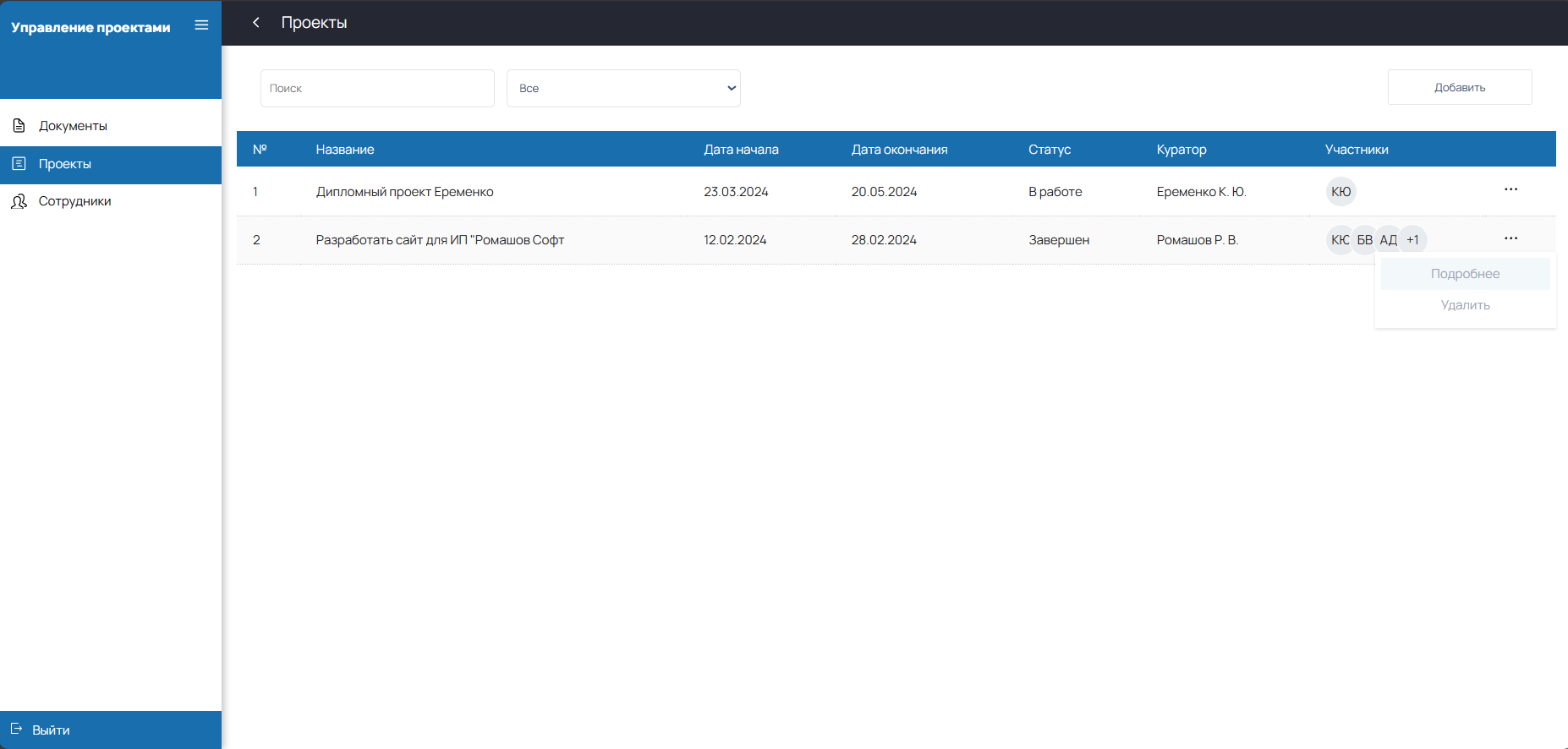


Рисунок 4.7 – Интерфейс страницы проекты

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 4.8 изображено окно добавления проекта.

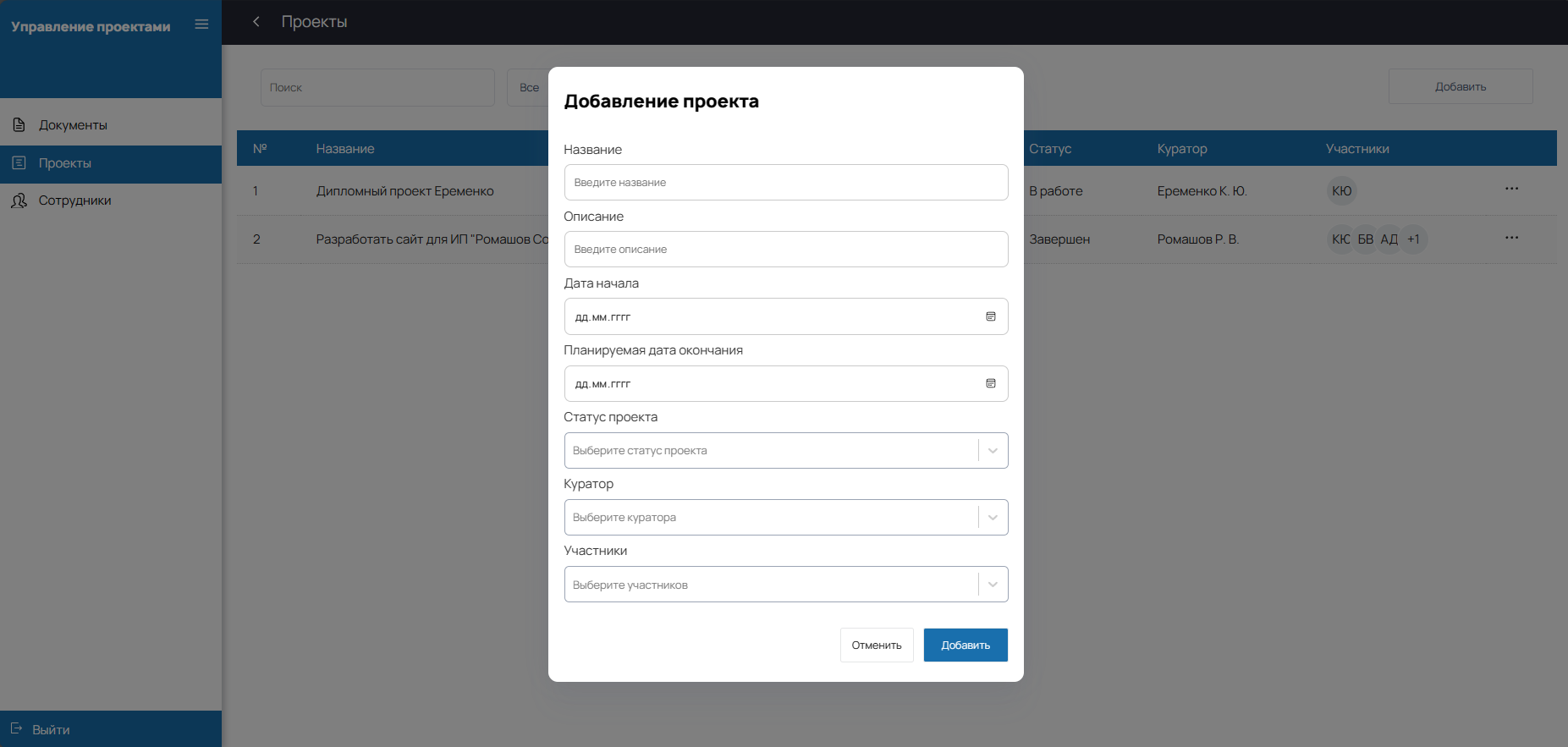


Рисунок 4.8 – Окно добавления проекта

На изображении показан интерфейс окна добавления проекта, являющийся частью системы управления проектами. Этот интерфейс оформлен в виде модального окна, которое накладывается поверх основной страницы, и содержит форму для ввода данных нового проекта. Интерфейс разделен на несколько блоков, каждый из которых предназначен для ввода определенной информации о проекте.

На рисунке 4.9 изображен интерфейс страницы участников конкретного проекта.

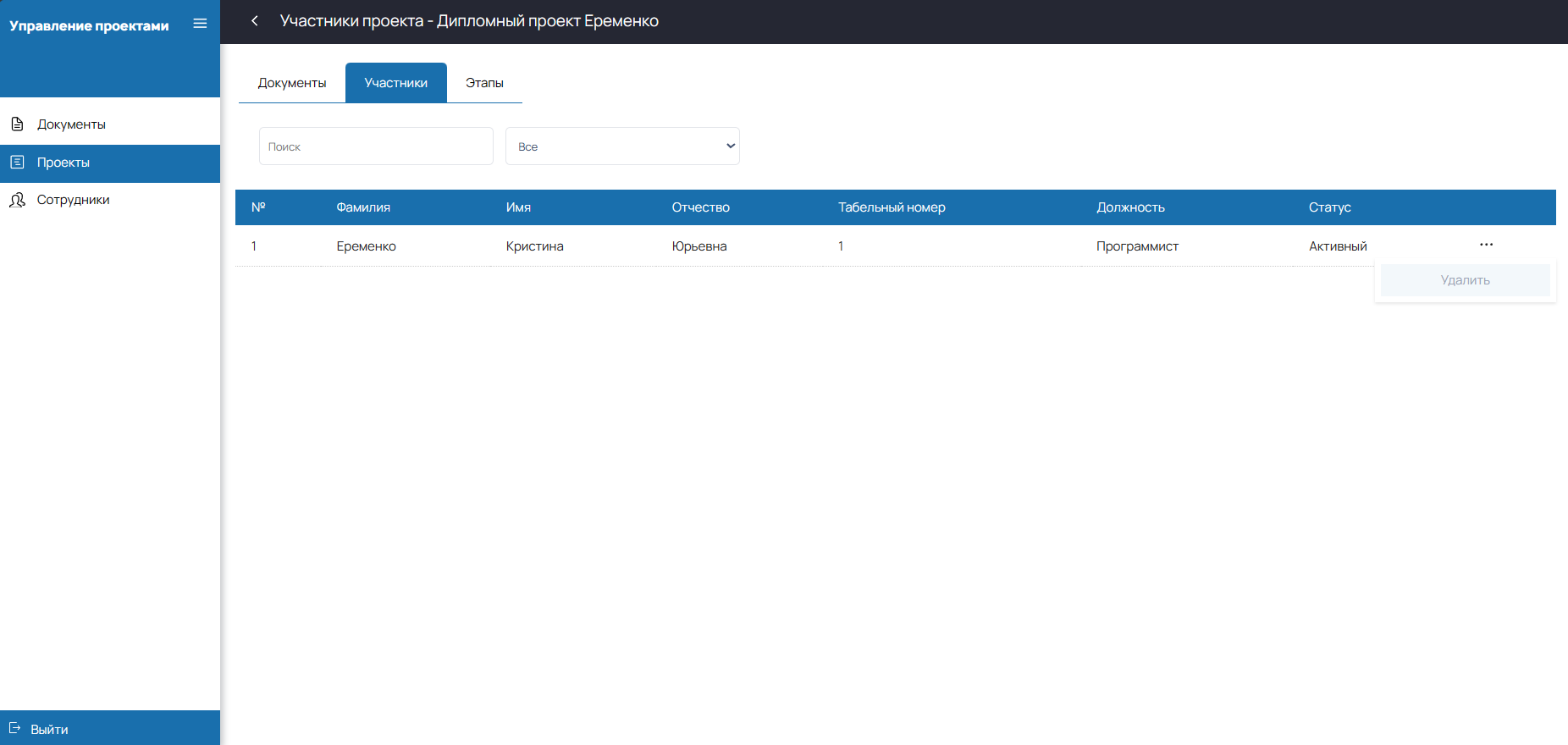


Рисунок 4.9 – Интерфейс страницы участников проекта

Интерфейс страницы участников проекта является простым, понятным и функциональным. Пользователь может легко просматривать список участников, выполнять их удаление, применять фильтры по должности и осуществлять поиск по имени, фамилии и отчеству.

На рисунке 4.10 изображен интерфейс страницы этапов конкретного проекта.

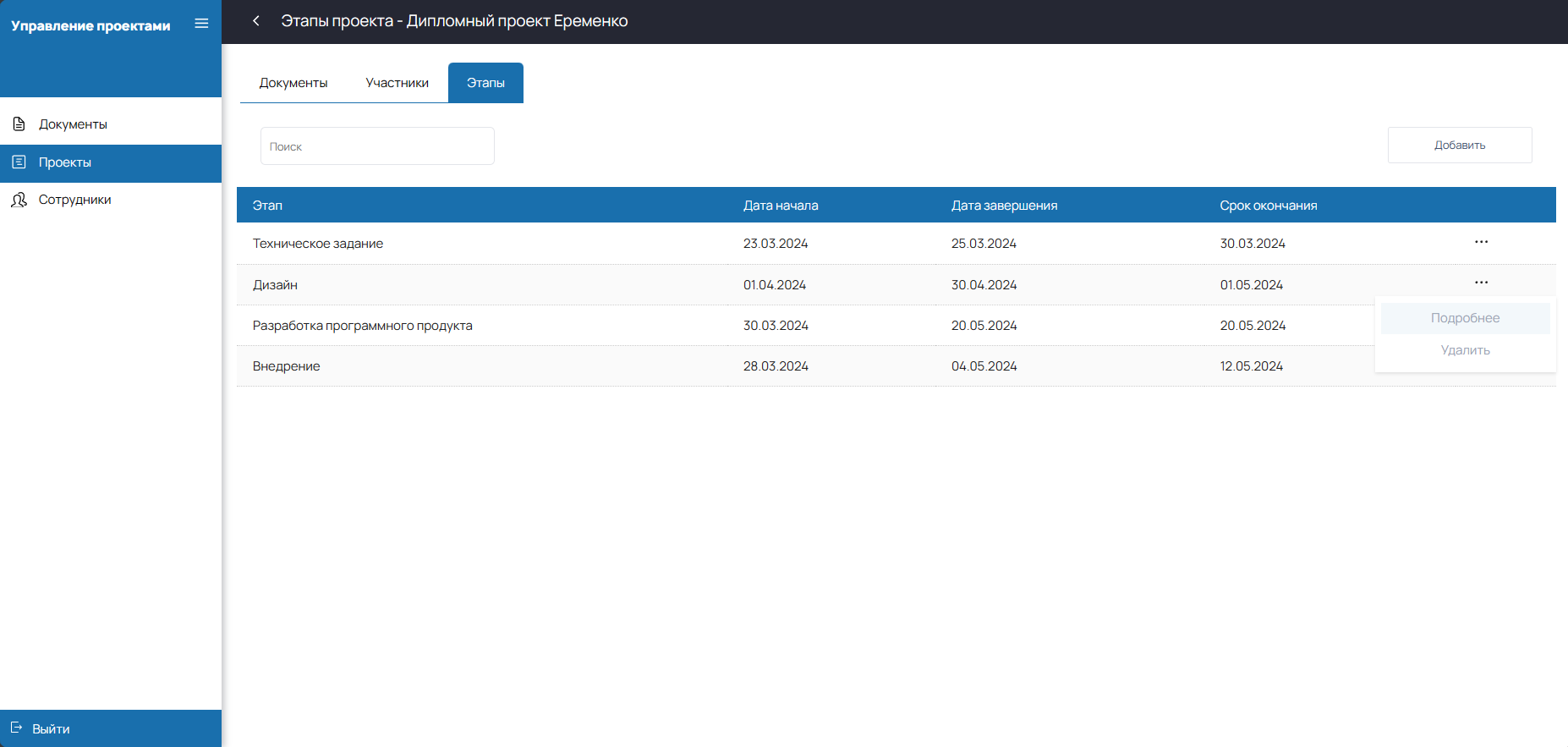


Рисунок 4.10 – Интерфейс страницы этапов проекта

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 4.11 изображено окно добавления этапа проекта.

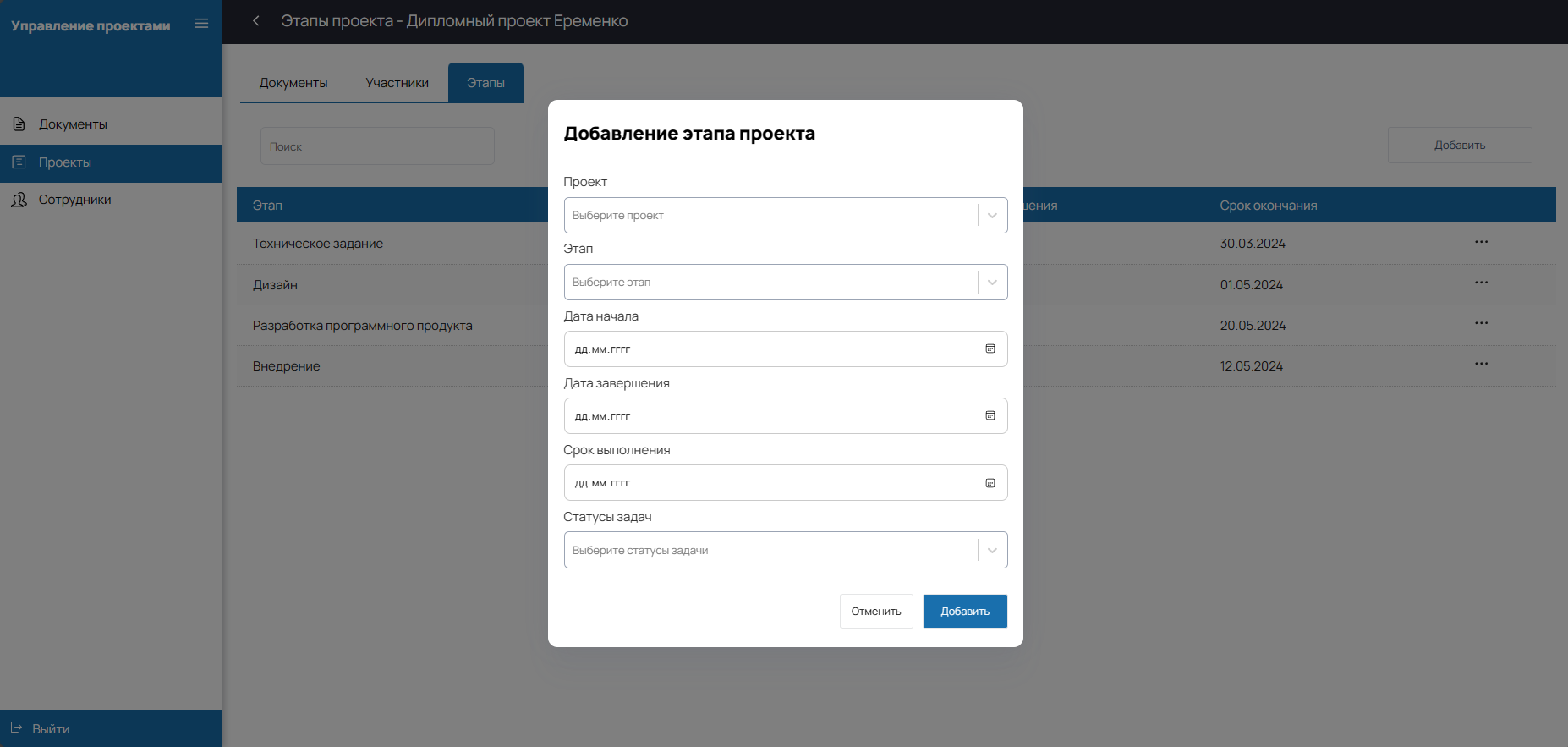


Рисунок 4.11 – Окно добавления этапа проекта

Окно добавления этапа проекта, представленное на изображении, позволяет пользователям создавать новые этапы в выбранном проекте. Этот интерфейс оформлен в виде модального окна, которое накладывается поверх основной страницы, и содержит форму для ввода данных нового этапа.

На рисунке 4.12 изображен интерфейс страницы подробности этапа конкретного проекта.

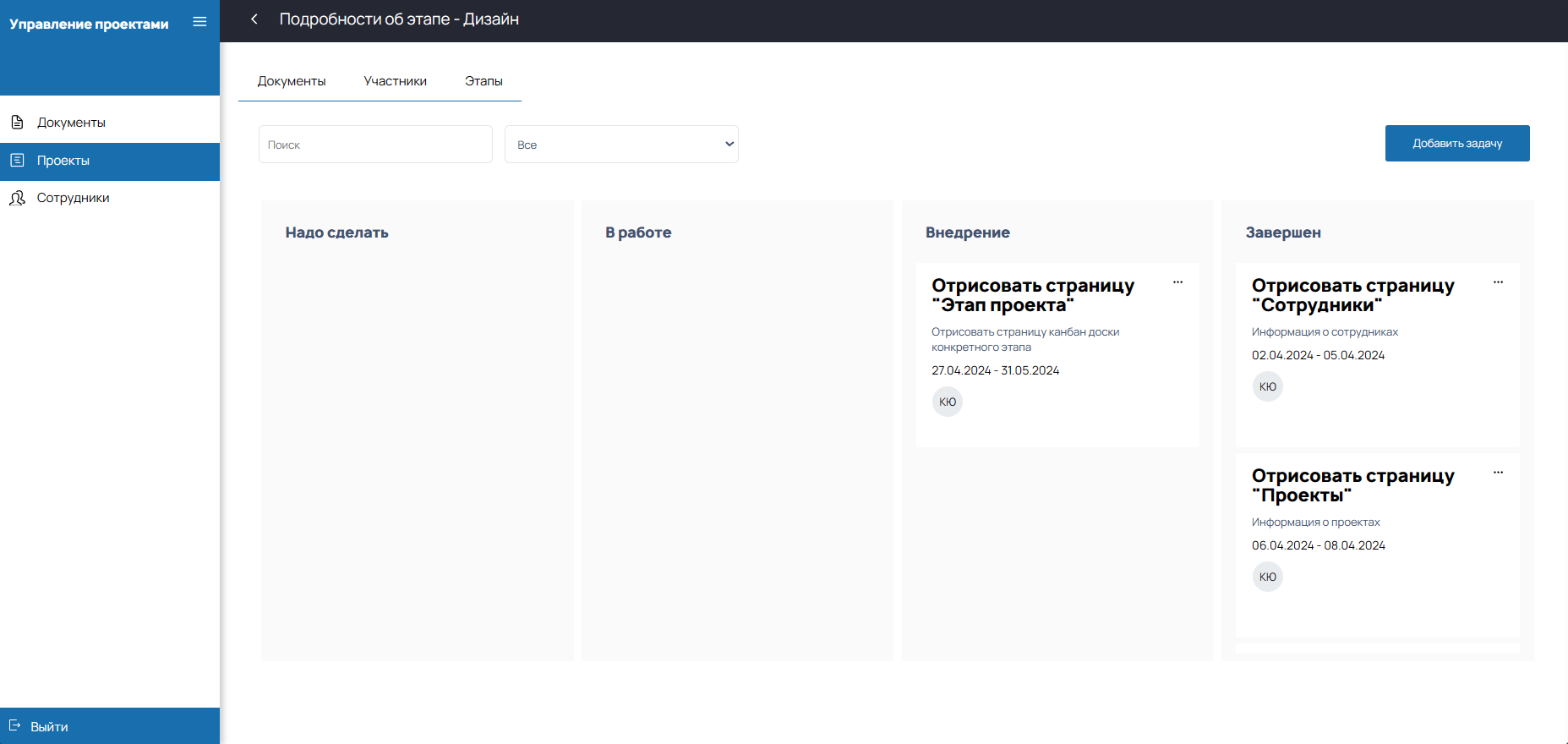


Рисунок 4.12 – Интерфейс страницы подробности этапа проекта

Страница с подробностями этапа проекта позволяет пользователям просматривать информацию о текущем этапе, добавлять новые задачи, фильтровать их по статусу и осуществлять поиск по названию. Этот интерфейс обеспечивает удобное управление задачами и настройку их в соответствии с требованиями проекта.

На рисунке 4.13 изображен интерфейс страницы подробности о выбранной задаче.

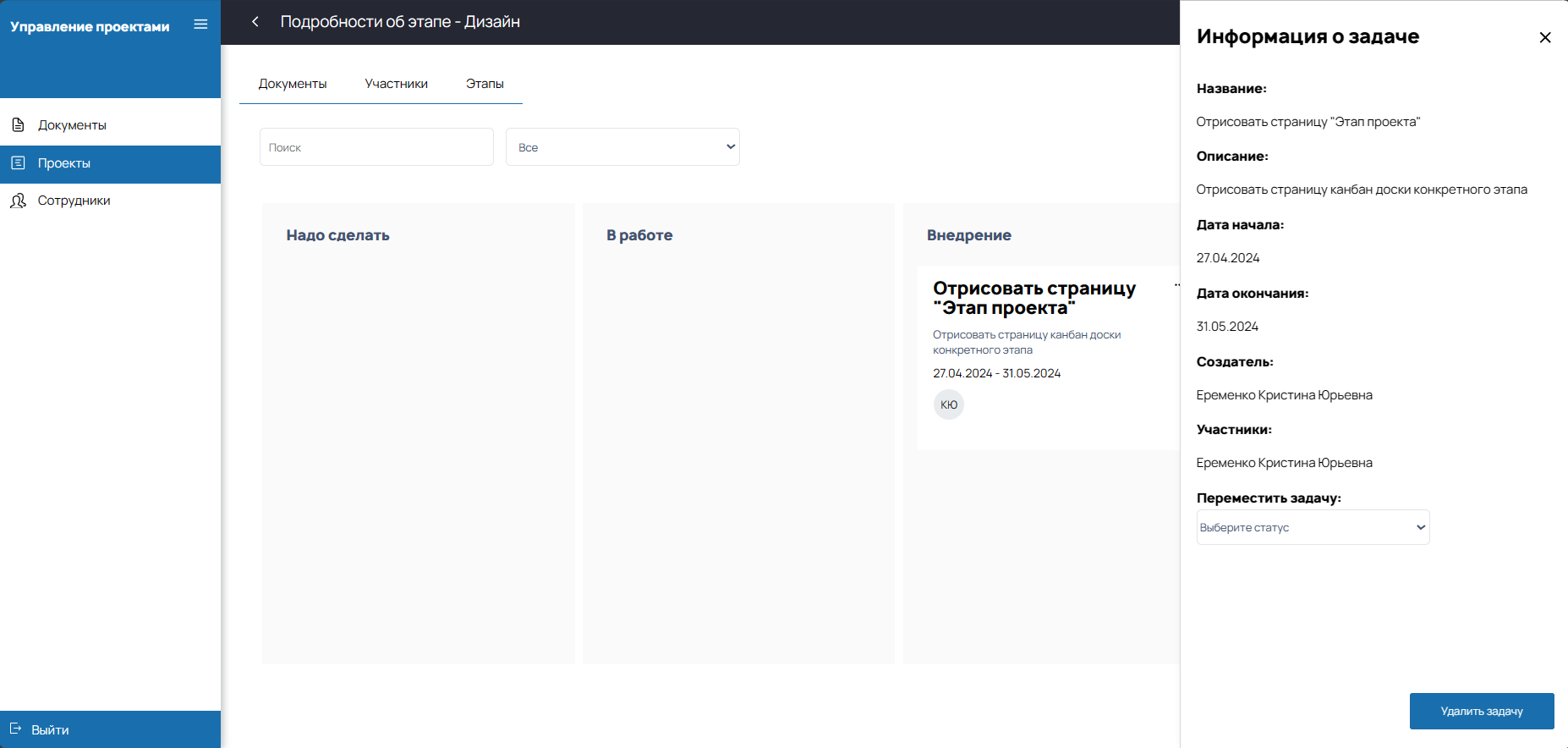


Рисунок 4.13 – Интерфейс страницы подробности о задаче

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 4.14 изображено окно добавления задачи.

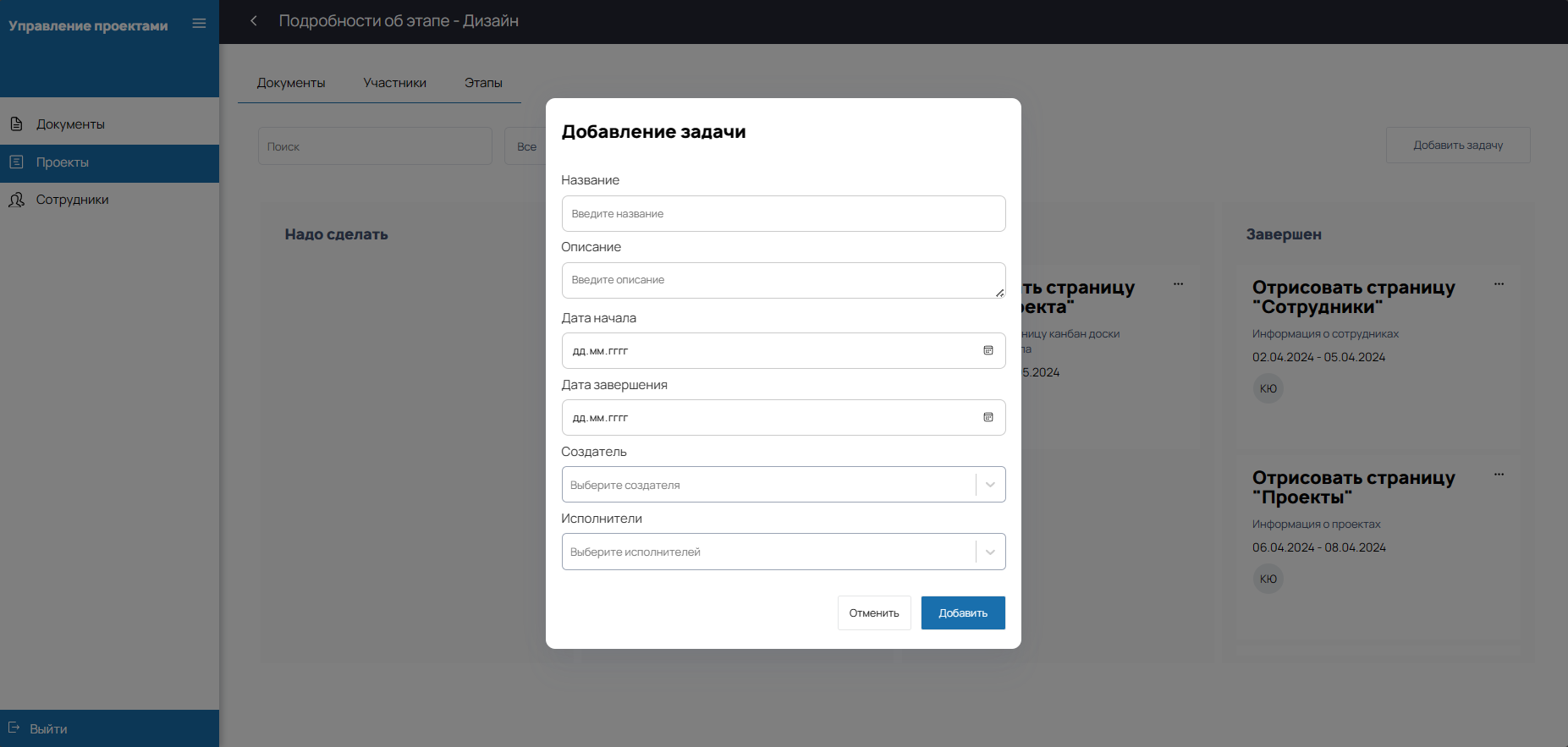


Рисунок 4.14 – Окно добавления задачи

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 4.15.

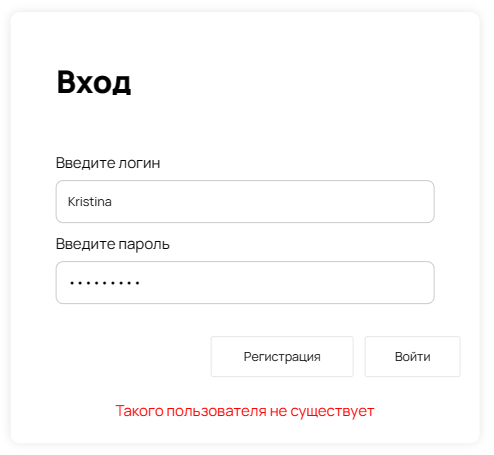


Рисунок 4.15 – Сообщение об ошибке

На рисунке 4.16 будет показано сообщение, которое появится при попытке регистрации нового пользователя с уже существующими данными, указывающее на то, что пользователь с такими данными уже существует.

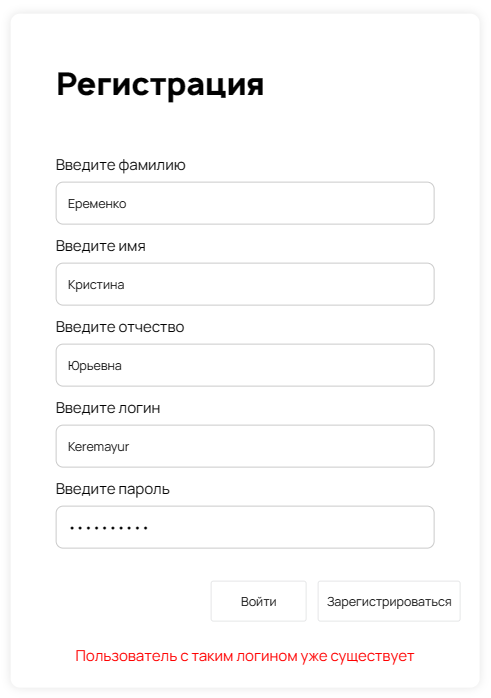


Рисунок 4.16 – Сообщение об ошибке

При попытке входа без заполнения полей, пользователь увидит другие сообщения, изображенные на рисунке 4.17.

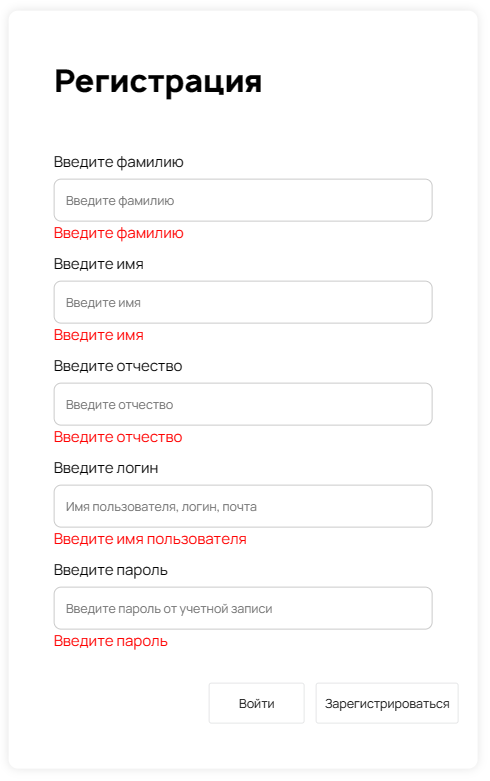


Рисунок 4.17 – Сообщение об ошибке

Эти сообщения являются уведомлениями об ошибках, которые предупреждают пользователя о необходимости заполнения всех обязательных полей для успешного входа в систему.

На рисунке 4.18 изображена фильтрация данных.



Рисунок 4.18 – 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 4.19 представлена функция поиска данных.



Рисунок 4.19 – 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 4.20 изображена кнопка для добавления данных.



Рисунок 4.20 – 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 4.21 изображена кнопка для редактирования данных.

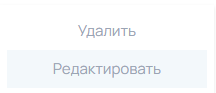


Рисунок 4.21 – Кнопка редактирования данных

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 4.22 изображена кнопка удаления данных.

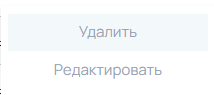


Рисунок 4.22 – Кнопка удаления данных

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 4.23 изображен запрос на подтверждение удаления.

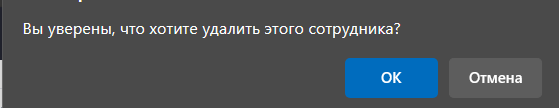


Рисунок 4.23 – Окно подтверждения удаления

На изображении 4.24 показана кнопка, предоставляющая дополнительную информацию.

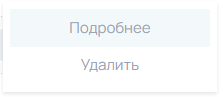


Рисунок 4.24 – Кнопка подробнее

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 4.25 изображена карточка задачи.

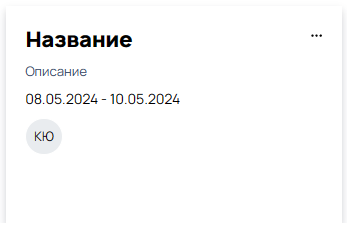


Рисунок 4.25 – Карточка задачи

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.

# 5 Экономическая часть

В расчёте экономической части дипломного проекта используются следующие статьи затрат:

* материальные затраты – материалы, требуемые при выполнении дипломного проекта, включающие свою стоимость в готовую продукцию;
* электроэнергия – затраты, связанные с потреблением электричества оборудованием используемыми при выполнении дипломного проекта;
* заработная плата – плата за фактически отработанное время;
* прочие затраты.

Расчет технологического процесса в написании программного продукта приведен в таблице 5.1.

Таблица 5.1 – Этапы создания программы

|  |  |
| --- | --- |
| Виды работ | Количество времени (ч.) |
| Получение задания | 2 |
| Подбор материалов | 2 |
| Написание программы | 92 |
| Отладка программы | 24 |
| Тестирование | 24 |
| Написание пояснительной записки | 16 |
| Итого: | 160 |

Расчет материальных затрат.

В процессе создания программного продукта были осуществлены материальные затраты, приведенные в таблице ниже, учитываются, как покупные, так и вспомогательные материалы. Расчеты приведены в таблице 5.2.

Таблица 5.2 ­– Расчет материальных затрат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Единицы измерения | Кол-во (шт.) | Цена (руб.) | Сумма (руб.) |
| Ручка | шт. | 1 | 30 | 30 |
| Оплата интернета | мес. | 1 | 400 | 400 |
| Папка | шт. | 1 | 200 | 200 |
| Бумага | уп. | 1 | 200 | 200 |
| Печать на принтере | лист | 78 | 10 | 780 |
| Итого: | | | | 1610 |

Расчет амортизационных отчислений.

Расчет амортизационных отчислений производится по годовым нормам амортизации исходя из первоначальной стоимости оборудования по формуле 1:

A = S × N, (1)

где S – первоначальная стоимость;

N – годовая норма амортизации.

Расчет суммы годовой амортизации выполняется по формуле 1.

А(стол) = 5000 × 10% = 500,00 руб.;

А(компьютер) = 60000 × 33,3% = 20000,00 руб.;

А(стул) = 2500 × 10% = 250,00 руб.

Расчет месячной суммы амортизации проводится по формуле 2:

, (2)

где Амес – месячная сумма амортизации, руб.;

Агод – годовая сумма амортизации, руб.;

12 – количество месяцев в году, мес.

Проведем расчет месячной суммы амортизации по формуле 2:

А(мес) = 20750 / 12 = 1729,17 руб.

Расчет амортизационных отчислений во времени фактического создания программного продукта производится по формуле 3:

A = ((S × N) / Rd) × t, (3)

где S – первоначальная стоимость, руб.;

N – годовая норма амортизации;

Rd – количество рабочего времени, ч.;

t – фактически затраченное время на работу, ч.

Далее, рассчитаем амортизацию на время разработки программного продукта по формуле 3:

А(вр) = (1729,17 / 160) × 40 = 432,29 руб.

Расчет амортизации основных фондов показан в таблице 5.3.

Таблица 5.3 – Расчет амортизации основных фондов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование основных фондов | Первоначальная стоимость (руб.) | Норма  амортизации | Годовая сумма амортизации (руб.) | Амортизация на выполнение проекта (руб.) |
| Стол | 5000,00 | 10% | 500,00 | 432,29 |
| Компьютер | 60000,00 | 33,3% | 20000,00 |
| Стул | 2500,00 | 10% | 250,00 |
| Итого: | 67500,00 |  | 20750,00 |

Расчеты расходов на электроэнергию показаны в таблице 5.4.

Таблица 5.4 – Расходы на электроэнергию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Источники  потребления | Потребляемая мощность (кВт/ч) | Время работы (час) | Цена за 1 кВт/ч (руб.) | Сумма (руб.) |
| Компьютер | 0,5 | 160 | 4,81 | 384,8 |
| Эл. лампочка | 0,1 | 25 | 4, 81 | 12,03 |
| Итого: | | | | 396,83 |

Расчет заработной платы.

Расчет заработной платы производится с учетом фактически использованного времени согласно техническому процессу и средней заработной платы, насчитанной в регионе на момент выполнения дипломного проекта.

Заработная плата за день работы приведена в формуле 4:

, (4)

где ZД. – зарплата за 1 день,

ZМЕС. – зарплата за 1 месяц,

D *–* количество рабочих дней в месяце.

Расчет заработной платы приведен в таблице 5.5.

Таблица 5.5 – Расчет заработной платы

|  |  |  |  |
| --- | --- | --- | --- |
| Тарифная ставка | Кол-во часов в мес. | Кол-во часов потраченных на написание ПП | Итоговая сумма (руб.) |
| 220 | 176 | 160 | 35200 |

Расчет коммунальных услуг показан в таблице 5.6.

Сумма затрат на коммунальные услуги рассчитывается за время фактического использования рабочего места в течение написания дипломного проекта и действующих тарифов.

Таблица 5.6 – Расходы на коммунальные услуги

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование услуг | Единица измерения | Тариф (руб.) | Количество | Сумма (руб.) |
| Горячая вода | Метр кубический | 33,44 | 1,5 | 50,16 |
| Холодная вода | Метр кубический | 31,58 | 3 | 94,74 |
| Вывоз ТБО | С человека | 134,92 | 1 | 134,92 |
| Итого: | | | | 279,82 |

Расчет себестоимости (затрат) выполнения дипломного проекта с выполнением указанного задания показан в таблице 5.7.

Таблица 5.7 – Себестоимость по прямым затратам

|  |  |
| --- | --- |
| Наименование статей затрат | Сумма (руб.) |
| Материальные затраты | 1610,00 |
| Заработная плата | 35200,00 |
| Амортизация основных фондов | 432,29 |
| Расходы на электроэнергию | 396,83 |
| Прочие затраты | 279,82 |
| Итого: | 37 918,94 |

Калькуляция рассчитана по прямым затратам.

Цена – это денежное выражение стоимости программного продукта. Для расчета цены принимаем прибыль 15% и рассчитываем по формулам 5 и 6:

П = (С × 15) / 100, (5)

Ц = С + П, (6)

где П – прибыль;

С – себестоимость;

Ц – цена.

Рассчитаем цену программного продукта по формулам 5 и 6.

П = (37 918,94 × 15) / 100 = 5 687,84 руб.;

Ц = 37 918,94 + 5 687,84 = 43 606,78 руб.

Исходя из приведенных расчетов и сведений, что подобная лицензионная программа, написанная на заказ, обходится предприятиям в несколько раз дороже, можно сделать вывод что, созданный экономически выгоднее для предприятия. В этом и состоит эффективность этого программного продукта.

# 6 Охрана труда и техники безопасности

Требования к помещениям для эксплуатации персональных компьютеров:

* площадь на одно рабочее место пользователей на базе плоских дискретных экранов составляет 4,5 кв.м. Помещение с ПК должно быть оборудовано системами отопления, кондиционирования или эффективной вентиляции воздуха [17].

Требования к микроклимату:

* в производственных помещениях, в которых работа на ПК является основной, должны обеспечиваться оптимальные параметры микроклимата;
* для повышения влажности воздуха в помещениях с ПК следует применять увлажнители воздуха.

Требования к освещению помещений и рабочих мест с персональным компьютером:

* помещения с ПК должны иметь естественное и искусственное освещение;
* устанавливать компьютер на рабочем месте таким образом, чтобы солнечные лучи не падали на экран [18].

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ в помещениях с источниками вредных производственных факторов должны размещаться в изолированных кабинах с организованным воздухообменом.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 - 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений [19, 20].

Требования к организации режима труда и отдыха при работе с персональным компьютером:

* для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей, на протяжении работы должны устанавливаться регламентированные перерывы;
* продолжительность непрерывной работы с ПК без регламентированного перерыва не должна превышать 2 часов;
* при работе на ПК для предупреждения развития переутомления необходимо осуществлять комплекс профилактических мероприятий.

При работе необходимо соблюдать правила пожарной безопасности, знать места расположения первичных средств пожаротушения.

Конструкция ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана ВДТ. Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность с коэффициентом отражения 0,4 - 0,6 и не иметь блестящих деталей, способных создавать блики.

Мощность экспозиционной дозы мягкого рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана и корпуса ВДТ (на электронно-лучевой трубке) при любых положениях регулировочных устройств не должна превышать 1 мкЗв/ч (100 мкР/ч) [21, 22].

Концентрации вредных веществ, выделяемых ПЭВМ в воздух помещений, не должны превышать предельно допустимых концентраций (ПДК), установленных для атмосферного воздуха.

В производственных помещениях, в которых работа с использованием ПЭВМ является вспомогательной, температура, относительная влажность и скорость движения воздуха на рабочих местах должны соответствовать действующим санитарным нормам микроклимата производственных помещений.

В производственных помещениях, где работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а и 1б согласно действующим санитарно-эпидемиологическим нормативам. На других рабочих местах микроклимат должен поддерживаться на допустимом уровне. Документация на проектирование, изготовление и эксплуатацию ПЭВМ должна соответствовать этим санитарным правилам [23].

# Заключение

Разработанное веб-приложение для управления проектами полностью соответствует требованиям, изложенным в задании. В процессе выполнения проекта удалось достичь всех поставленных целей, включая оптимизацию взаимодействия между сотрудниками, улучшение контроля над выполнением задач и снижение вероятности ошибок на различных этапах проекта.

Основные выводы по результатам выполненного проекта:

* разработанная система обеспечивает безопасный доступ к данным, гарантируя защиту конфиденциальности и предотвращая несанкционированный доступ;
* веб-приложение позволяет создавать, управлять и контролировать выполнение проектов, определяя их этапы, назначая ответственных и участников, а также отслеживая прогресс и статусы каждого этапа. Это способствует более эффективному распределению ресурсов и повышению прозрачности в процессе работы над проектами;
* приложение обеспечивает эффективное управление задачами, позволяя назначать задачи сотрудникам и отслеживать их выполнение. Это способствует улучшению организации работы, сокращению времени на выполнение задач и повышению производительности команды;
* система позволяет собирать, хранить и обновлять информацию о сотрудниках, включая их личные данные, должностные обязанности и статус активности, обеспечивая централизованное хранение данных;
* пользовательский интерфейс веб-приложения интуитивно понятен и удобен, что снижает время обучения сотрудников и ускоряет процесс адаптации к новому инструменту.

Таким образом, можно заключить, что разработанное веб-приложение эффективно решает поставленные задачи и способствует значительным улучшениям в управлении проектами, что положительно отражается на работе команды и общем уровне удовлетворенности клиентов.

Список использованных источников

Список использованной литературы

1. Адам Фримен. Основы TypeScript – Питер, 2024. – 576 с.
2. Бэнкс Алекс, Порселло Ева. React: современные шаблоны для разработки приложений. 2-е изд. – Питер, 2022. – 320 с.
3. Дейли Брэд, Дейли Брендан, Дейли Калеб. Разработка веб-приложений с помощью Node.js – Вильямс, 2020. – 656 с.
4. Скотт Адам Д., Пауэрс Шелли, Макдональд Мэтью. JavaScript Рецепты для разработчиков. 3-е изд. – Питер, 2023. ­– 528 c.
5. Шеннон Брэдшоу, Йон Брэзил, Кристина Ходоров. MongoDB: полное руководство. Мощная и масштабируемая система управления базами данных / пер. с англ. Д. А. Беликова – ДМК Пресс, 2020. – 540 с.
6. Грекул, В. И. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. – 2-е изд. – Москва: Издательство Юрайт, 2024. – 423 с. – (Профессиональное образование). – ISBN 978-5-534-17836-4. – Текст: электронный // Образовательная платформа Юрайт.
7. Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML: учебное пособие для вузов / Э. Г. Галиаскаров, А. С. Воробьев. – Москва: Издательство Юрайт, 2024. – 125 с. – (Высшее образование). – ISBN 978-5-534-14903-6. – Текст: электронный // Образовательная платформа Юрайт.
8. Стружкин, Н. П. Базы данных: проектирование: учебник для вузов / Н. П. Стружкин, В. В. Годин. – Москва: Издательство Юрайт, 2024. – 477 с. – (Высшее образование). – ISBN 978-5-534-00229-4. – Текст: электронный // Образовательная платформа Юрайт.
9. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения: учебное пособие для вузов / С. А. Чернышев. — Москва: Издательство Юрайт, 2024. – 176 с. – (Высшее образование). – ISBN 978-5-534-14383-6. – Текст: электронный // Образовательная платформа Юрайт.
10. Долганова, О. И. Моделирование бизнес-процессов: учебник и практикум для вузов / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова; под редакцией О. И. Долгановой. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 322 с. – (Высшее образование). – ISBN 978-5-534-17914-9. – Текст: электронный // Образовательная платформа Юрайт.

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. Прототипирование [Электронный ресурс] – Режим доступа: <https://www.unisender.com/ru/glossary/chto-takoe-prototipirovanie-i-zachem-ono-nuzhno/>.
5. 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>.
6. Интерфейс программного продукта [Электронный ресурс] – Режим доступа:<https://old.bigenc.ru/technology_and_technique/text/4426494?ysclid=lwernwq9ig271342600>.
7. Основные требования для помещений [Электронный ресурс] – Режим доступа: <https://ohranatruda.ru>.
8. Требования международных стандартов [Электронный ресурс] – Режим доступа: <https://technormativ.ru>.
9. Требования к рабочему месту [Электронный ресурс] – Режим доступа: <http://docs.cntd.ru>.
10. Конструкция рабочего места [Электронный ресурс] – Режим доступа: <https://studwood.ru>.
11. ПЭВМ [Электронный ресурс] – Режим доступа: <http://prom-nadzor.ru>.
12. Работа с ПЭВМ [Электронный ресурс] – Режим доступа: <http://mirzna-nii.com>.
13. Производственные факторы [Электронный ресурс] – Режим доступа: <https://legalacts.ru>.

# Приложение А

(обязательное)

Листинг программы

Index.tsx

import React from "react";

import ReactDOM from "react-dom/client";

import App from "./App";

const root = ReactDOM.createRoot(

document.getElementById("root") as HTMLElement

);

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

App.tsx

import React from "react";

import "./App.css";

import { BrowserRouter, Route, Routes } from "react-router-dom";

import EmployeesPage from "./views/employees/EmployeesPage";

import ProjectsPage from "./views/projects/ProjectsPage";

import ParticipantsPage from "./views/projects/projectDetails/participantsPage/ParticipantsPage";

import StagesPage from "./views/projects/projectDetails/stagesPage/StagesPage";

import StageDetailsProject from "./views/projects/projectDetails/stagesPage/stageProjectDetails/StageDetailsProject";

import Login from "./views/auth/login/Login";

import Register from "./views/auth/register/Register";

function App() {

return (

<>

<BrowserRouter>

<Routes>

{/\* Авторизация \*/}

<Route path="/" element={<Login />} />

<Route path="/register" element={<Register />} />

{/\* Сотрудники \*/}

<Route path="/employeesPage" element={<EmployeesPage />}></Route>

{/\* Проекты \*/}

<Route path="/projectsPage" element={<ProjectsPage />}></Route>

{/\* Участники проекта \*/}

<Route

path="/projectsPage/projectDetails/:projectId/participants"

element={<ParticipantsPage />}

></Route>

{/\* Этапы проекта\*/}

<Route

path="/projectsPage/stageDetails/:projectId/:stageId/stages"

element={<StagesPage />}

></Route>

{/\* Подробнее об этапе проекта \*/}

<Route

path="/projectsPage/stageDetails/:projectId/:stageId/stages/:stageProjectId"

element={<StageDetailsProject />}

/>

</Routes>

</BrowserRouter>

</>

);

}

export default App;

ProjectInterfaces.ts

export interface EmployeeStatus {

\_id: string;

title: string;

}

export interface Position {

\_id: string;

title: string;

}

export interface Employee {

\_id: string;

lastName: string;

firstName: string;

middleName: string;

gender: String;

serviceNumber: number;

positionId: Position;

employeeStatusId: EmployeeStatus;

}

export interface ProjectStatus {

\_id: string;

title: string;

}

export interface Project {

\_id: string;

title: string;

description: string;

startDate: string;

endDate: string;

statusProjectId: ProjectStatus;

supervisorId: Employee[];

}

export interface EmployeeProject {

\_id: string;

employeeId: Employee[];

projectId: Project;

}

ProjectsPage.tsx

import React, { useEffect, useState } from "react";

import Sidebar from "../../components/sidebar/Sidebar";

import { menuItems } from "../../components/sidebar/DataSidebar";

import axios from "axios";

import { Link } from "react-router-dom";

import { MdArrowBackIos } from "react-icons/md";

import { HiEllipsisHorizontal } from "react-icons/hi2";

import "./style.css";

import { Employee, EmployeeProject } from "./ProjectInterfaces";

import AddProjectForm from "../../components/project/addProjectForm/AddProjectForm";

const URL = process.env.REACT\_APP\_URL;

const ProjectsPage: React.FC = () => {

const [searchQuery, setSearchQuery] = useState<string>("");

const [filterStatus, setFilterStatus] = useState<string>("");

const [employeeProjects, setEmployeeProjects] = useState<EmployeeProject[]>(

[]

);

const [filteredProjects, setFilteredProjects] = useState<EmployeeProject[]>(

[]

);

const [openPopoverId, setOpenPopoverId] = useState<string | null>(null);

const [isModalOpen, setIsModalOpen] = useState<boolean>(false);

useEffect(() => {

fetchEmployeeProjects();

}, []);

const fetchEmployeeProjects = async () => {

try {

const response = await axios.get<EmployeeProject[]>(

`${URL}/get/employeeProject`

);

setEmployeeProjects(response.data);

} catch (error) {

console.error("Ошибка:", error);

}

};

useEffect(() => {

const filteredData = employeeProjects.filter((project) =>

project.projectId.title.toLowerCase().includes(searchQuery.toLowerCase())

);

if (filterStatus !== "") {

const filteredByStatus = filteredData.filter(

(project) => project.projectId.statusProjectId.title === filterStatus

);

setFilteredProjects(filteredByStatus);

} else {

setFilteredProjects(filteredData);

}

}, [employeeProjects, filterStatus, searchQuery]);

const handleDelete = (id: string) => {

if (window.confirm("Вы уверены, что хотите удалить этот проект?")) {

axios

.delete(`${URL}/employeeProject/${id}`)

.then(() => {

setEmployeeProjects(

employeeProjects.filter((project) => project.\_id !== id)

);

})

.catch((err) => console.log(err));

}

};

const togglePopover = (id: string) => {

setOpenPopoverId(openPopoverId === id ? null : id);

};

useEffect(() => {

const handleClickOutside = (event: any) => {

if (!event.target.closest(".HiEllipsisHorizontal"))

setOpenPopoverId(null);

};

document.addEventListener("click", handleClickOutside);

return () => {

document.removeEventListener("click", handleClickOutside);

};

}, []);

const handleAdd = () => {

setIsModalOpen(true);

};

const handleCloseModal = () => {

setIsModalOpen(false);

};

return (

<>

<Sidebar items={menuItems}>

<div className="header">

<div className="divArrowBackIos">

<MdArrowBackIos className="MdArrowBackIos" />

</div>

<p>Проекты</p>

</div>

<div className="container">

<div className={"container\_search\_filter"}>

<div className={"div\_input\_search"}>

<input

type="text"

className={"input\_search"}

placeholder="Поиск"

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

/>

</div>

<div className={"div\_filter"}>

<select

className={"filter"}

value={filterStatus}

onChange={(e) => setFilterStatus(e.target.value)}

>

<option value="">Все</option>

{Array.from(

new Set(

employeeProjects.map(

(project) => project.projectId.statusProjectId.title

)

)

).map((statusProjectId, index) => (

<option key={index} value={statusProjectId}>

{statusProjectId}

</option>

))}

</select>

</div>

<div className={"containet\_btn\_add"}>

<button className={"btn\_add"} onClick={handleAdd}>

Добавить

</button>

</div>

</div>

<div>

<table>

<thead>

<tr>

<th>№</th>

<th>Название</th>

<th>Дата начала</th>

<th>Дата окончания</th>

<th>Статус</th>

<th>Куратор</th>

<th>Участники</th>

<th></th>

</tr>

</thead>

<tbody>

{filteredProjects.map((project, index) => {

return (

<tr key={project.\_id}>

<td>{index + 1}</td>

<td>{project.projectId.title}</td>

<td>

{new Date(

project.projectId.startDate

).toLocaleDateString()}

</td>

<td>

{new Date(

project.projectId.endDate

).toLocaleDateString()}

</td>

<td>{project.projectId.statusProjectId.title}</td>

<td>

{project.projectId.supervisorId.map((employee) => {

return `${

employee.lastName

} ${employee.firstName.charAt(

0

)}. ${employee.middleName.charAt(0)}.`;

})}

</td>

<td>

<div className="avatar-container">

{project.employeeId.length > 0 ? (

project.employeeId

.slice(0, 3)

.map((employee: Employee, index) => (

<div key={index} className="avatar">

<div className="avatar-letter">

{employee.firstName.charAt(0)}

{employee.middleName

? employee.middleName.charAt(0)

: ""}

</div>

</div>

))

) : (

<span>Нет данных</span>

)}

{project.employeeId.length > 3 && (

<span>

<div className="avatar">

{" "}

<span>+{project.employeeId.length - 3}</span>

</div>

</span>

)}

</div>

</td>

<td>

<HiEllipsisHorizontal

className="HiEllipsisHorizontal"

onClick={() => togglePopover(project.\_id)}

/>

{openPopoverId === project.\_id && (

<div className="popover-content">

<div className="div-popover-content">

<div className="div\_edit">

<Link

to={`/projectsPage/stageDetails/${project.\_id}/${project.projectId.\_id}/stages`}

>

<p>Подробнее</p>

</Link>

</div>

<div

onClick={() => handleDelete(project.\_id)}

className="div\_delete"

>

<p>Удалить</p>

</div>

</div>

</div>

)}

</td>

</tr>

);

})}

</tbody>

</table>

</div>

</div>

<AddProjectForm isOpen={isModalOpen} onClose={handleCloseModal} />

</Sidebar>

</>

);

};

export default ProjectsPage;

AddProjectForm.tsx

import React, { useState, useEffect } from "react";

import axios from "axios";

import { Link } from "react-router-dom";

import Select from "react-select";

import { ProjectFormData, Employee, ProjectStatus } from "../../../views/projects/ProjectInterfaces";

import {style} from '../../../components/ui/StyleSelect';

const URL = process.env.REACT\_APP\_URL;

function AddProjectWithEmployee({ isOpen, onClose }: any) {

const [projectData, setProjectData] = useState<ProjectFormData>({

title: "",

description: "",

startDate: "",

endDate: "",

statusProjectId: "",

supervisorId: "",

employeeId: [],

});

const [employees, setEmployees] = useState<Employee[]>([]);

const [statusProjectId, setProjectStatuses] = useState<ProjectStatus[]>([]);

useEffect(() => {

fetchEmployees();

fetchProjectStatuses();

}, []);

const fetchEmployees = async () => {

try {

const response = await axios.get<Employee[]>(

`${URL}/get/employees`

);

setEmployees(response.data);

} catch (error) {

console.error("Ошибка:", error);

}

};

const fetchProjectStatuses = async () => {

try {

const response = await axios.get<ProjectStatus[]>(

`${URL}/get/projectStatuses`

);

setProjectStatuses(response.data);

} catch (error) {

console.error("Ошибка:", error);

}

};

const handleChange = (

e: React.ChangeEvent<HTMLInputElement | HTMLSelectElement>

) => {

const { name, value } = e.target;

setProjectData({ ...projectData, [name]: value });

};

const handleStatusChange = (selectedOption: any) => {

if (selectedOption) {

const statusProjectId = selectedOption.value;

setProjectData({ ...projectData, statusProjectId });

}

};

const handleSupervisorChange = (selectedOption: any) => {

if (selectedOption) {

const supervisorId = selectedOption.value;

setProjectData({ ...projectData, supervisorId });

}

};

const handleEmployeeIdChange = (selectedOptions: any) => {

if (selectedOptions) {

const employeeIds = selectedOptions.map((option: any) => option.value);

setProjectData({ ...projectData, employeeId: employeeIds });

}

};

const handleSubmit = async (e: React.FormEvent<HTMLFormElement>) => {

e.preventDefault();

try {

const response = await axios.post(

`${URL}/addProjectWithEmployee`,

projectData

);

window.location.reload();

onClose();

console.log(response.data.message);

} catch (error) {

console.error("Ошибка:", error);

}

};

return (

<>

{isOpen && (

<div className="pade">

<div className="wrapper">

<form onSubmit={handleSubmit}>

<div className="title-add">

<h3>Добавление проекта</h3>

</div>

<div className="container-data-form">

<div className="input\_div">

<label htmlFor="title">Название</label>

<div className="container\_input">

<input

type="text"

name="title"

value={projectData.title}

onChange={handleChange}

placeholder="Введите название"

className={"form\_control"}

required

/>

</div>

</div>

<div className="input\_div">

<label htmlFor="description">Описание</label>

<div className="container\_input">

<input

type="text"

name="description"

value={projectData.description}

onChange={handleChange}

placeholder="Введите описание"

className={"form\_control"}

required

/>

</div>

</div>

<div className="input\_div">

<label htmlFor="startDate">Дата начала</label>

<div className="container\_input">

<input

type="date"

name="startDate"

value={projectData.startDate}

onChange={handleChange}

className={"form\_control"}

required

/>

</div>

</div>

<div className="input\_div">

<label htmlFor="endDate">Планируемая дата окончания</label>

<div className="container\_input">

<input

type="date"

name="endDate"

value={projectData.endDate}

onChange={handleChange}

className={"form\_control"}

required

/>

</div>

</div>

<div className="input\_div">

<label htmlFor="status">Статус проекта</label>

<div className="container\_input">

<Select

// Опции для выбора, сформированные из отфильтрованного списка сотрудников

options={statusProjectId.map((status) => ({

value: status.\_id,

label: `${status.title}`,

}))}

// Функция обратного вызова, вызываемая при изменении выбранного куратора

onChange={handleStatusChange}

// Возможность очистить выбранное значение

isClearable

// Возможность поиска куратора по имени

isSearchable

placeholder="Выберите статус проекта"

styles={style}

required

/>

</div>

</div>

<div className="input\_div">

<label htmlFor="selectedEmployees">Куратор</label>

<div className="container\_input">

<Select

// Опции для выбора, сформированные из отфильтрованного списка сотрудников

options={employees.map((employee) => ({

value: employee.\_id,

label: `${employee.lastName} ${employee.firstName} ${employee.middleName}`,

}))}

// Функция обратного вызова, вызываемая при изменении выбранного куратора

onChange={handleSupervisorChange}

// Возможность очистить выбранное значение

isClearable

// Возможность поиска куратора по имени

isSearchable

placeholder="Выберите куратора"

styles={style}

required

/>

</div>

</div>

<div className="input\_div">

<label htmlFor="selectedEmployees">Участники</label>

<div className="container\_input">

<Select

// Опции для выбора, сформированные из списка всех сотрудников

options={employees.map((employee) => ({

value: employee.\_id,

label: `${employee.lastName} ${employee.firstName} ${employee.middleName}`,

}))}

// Функция обратного вызова, вызываемая при изменении выбранных участников

onChange={handleEmployeeIdChange}

// Возможность выбора нескольких участников одновременно

isMulti

placeholder="Выберите участников"

styles={style}

required

/>

</div>

</div>

</div>

<div className={"action\_buttons"}>

<div className="buttons">

<div>

<Link to={"/projectsPage"}>

<button className={"button\_add\_cancel"} onClick={onClose}>

Отменить

</button>

</Link>

</div>

<div>

<button className={"button\_add"}>Добавить</button>

</div>

</div>

</div>

</form>

</div>

</div>

)}

</>

);

}

export default AddProjectWithEmployee;