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

**Ростовской области**

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

**имени В.В. Самарского»**

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

**Дипломный проект**

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

**Специальность** 09.02.07 Информационные системы и программирование

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

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

Волгодонск

2024

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

**Ростовской области**

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

**имени В.В. Самарского»**

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

**Задание  
на выполнение выпускной квалификационной работы**

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

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

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

**Календарный план выполнения  
выпускной квалификационной работы**

СтуденткиЕременко Кристина Юрьевна

Специальность 09.02.07 Информационные системы и программирование

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

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

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

**Ростовской области**

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

**имени В.В. Самарского»**

**Отзыв  
на дипломный проект**

**Студентки** *Еременко Кристина Юрьевна*

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

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

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

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

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

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

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

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

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

**Ростовской области**

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

**имени В.В. Самарского»**

**Рецензия  
на дипломный проект**

**Студентки** *Еременко Кристина Юрьевна*

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

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

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

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

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

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

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

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

Содержание

[1 Технико-экономическая характеристика объекта 7](#_Toc165588943)

[2 Сбор, анализ и формирование требований к информационной системе 8](#_Toc165588944)

[2.1 Цели, назначение и функции программного продукта 9](#_Toc165588945)

[2.2 Требования к функциональности и дизайну программного продукта 10](#_Toc165588946)

[2.3 Системные требования к оборудованию 11](#_Toc165588947)

[3 Проектирование информационной системы 13](#_Toc165588948)

[3.1 Разработка и описание сценариев использования программного продукта 13](#_Toc165588949)

[3.2 Потоки данных в информационной системе 15](#_Toc165588950)

[3.3 Выбор архитектуры разрабатываемого программного продукта 17](#_Toc165588951)

[3.4 Проектирование реляционной базы данных на основе инфологической модели 17](#_Toc165588952)

[3.5 Прототипирование пользовательского интерфейса 21](#_Toc165588953)

[4 Разработка программного обеспечения 30](#_Toc165588954)

[4.1 Обоснование выбора инструментальных средств разработки программного продукта 30](#_Toc165588955)

[4.2 Описание алгоритма программы 35](#_Toc165588956)

[4.3 Описание интерфейса программного продукта 35](#_Toc165588957)

[4.3 Инструкция по эксплуатации программного продукта 35](#_Toc165588958)

[5 Экономическая часть 37](#_Toc165588959)

[6 Охрана труда и техники безопасности 38](#_Toc165588960)

[Заключение 39](#_Toc165588961)

[Список использованных источников 40](#_Toc165588962)

[Приложение А 41](#_Toc165588963)

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.

Существующие на рынке решения имеют определенные недостатки или ограничения, которые препятствуют эффективному использованию инструментов управления и вовлечению всех участников проекта. Для ИП «Ромашова Софт» разработка веб-приложения корпоративного управления проектами станет актуальной задачей, которая поможет решить множество проблем и повысить производительность компании.

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

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

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

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

Во введении была приведена оценка современного состояния решаемой задачи, представлены примеры существующих конкурентных проектов на рынке, показана актуальность и необходимость новой разработки, приведены основные предпосылки и исходные данные для разработки темы, а также показан предполагаемый экономический или социальный выигрыш, который может быть получен в результате данной работы.

# 1 Технико-экономическая характеристика объекта

# 2 Сбор, анализ и формирование требований к информационной системе

Сбор, анализ и формирование требований к информационной системе являются ключевыми этапами в разработке любой IT-системы, включая корпоративные веб-приложения. Этот процесс начинается с установления контакта с заказчиком, в данном случае, с ИП «Ромашов Софт», и другими заинтересованными сторонами. Основные этапы этого процесса включают:

1. Сбор требований:

* на этом этапе осуществляется встреча с заказчиком и другими заинтересованными сторонами для выявления их потребностей и ожиданий от будущей системы. Это может включать в себя проведение интервью, конференций или других форм коммуникации;
* результатом этого этапа является получение полного списка требований, включая функциональные (что система должна делать) и нефункциональные (требования к производительности, безопасности, удобству использования и т.д.).

1. Анализ требований:

* на этом этапе собранные требования систематизируются и классифицируются на основе их приоритетности и важности для бизнеса ИП «Ромашов Софт»;
* требования проверяются на соответствие бизнес-целям и стратегии компании;
* возможные противоречия или недостатки в требованиях выявляются и устраняются.

1. Формирование требований

* на этом этапе создается документ с требованиями к информационной системе. В этом документе содержатся функциональные и нефункциональные требования, описание пользовательских сценариев, интерфейсов и прочее;
* документ с требованиями утверждается и согласовывается с заказчиком и другими заинтересованными сторонами;

1. Дополнительные шаги

* проведение обсуждений и презентаций для уточнения требований и получения обратной связи от заказчика и других участников проекта;
* внесение корректировок в требования в соответствии с полученной обратной связью;
* обеспечение полного понимания требований всеми участниками проекта.

В результате этого процесса формируется документ, который будет использоваться в дальнейшей разработке веб-приложения. Он представляет собой основу для создания архитектуры приложения, проектирования интерфейса пользователя и реализации функциональности, отвечающей потребностям ИП «Ромашов Софт» в управлении проектами.

## 2.1 Цели, назначение и функции программного продукта

Цели проекта:

* эффективное управление проектами: создание инструмента, который позволит организациям эффективно управлять своими проектами, от начала до конца, с минимальными затратами времени и ресурсов;
* оптимизация процессов: планирование, выполнение и мониторинг проектов должны быть оптимизированы для достижения лучших результатов в установленные сроки и рамки бюджета;
* улучшение коммуникации: улучшение обмена информацией между участниками проектов, сокращение времени, требуемого для передачи этой информации, и улучшение понимания процесса работы над проектами.

Назначение программного продукта:

* централизованное управление проектами: приложение предоставит возможность централизованного управления всеми аспектами проекта, включая задачи, сроки и ресурсы;
* планирование и контроль: позволит планировать проекты, устанавливать сроки и бюджеты, а также контролировать их выполнение в режиме реального времени;
* улучшенная командная работа: обеспечит эффективную командную работу, путем предоставления возможности совместного доступа к информации, обсуждения задач и обмена файлами.

Функции программного продукта:

* управление задачами и проектами: создание, назначение и отслеживание выполнения задач и проектов;
* отслеживание прогресса: способность наблюдать за текущим состоянием задач и проектов, чтобы иметь представление о том, насколько они приближаются к завершению;
* доступ к документам: возможность хранить и делиться документами и файлами, необходимыми для проекта;

Удобства использования:

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

## 2.2 Требования к функциональности и дизайну программного продукта

Простой интерфейс:

* интуитивно понятные кнопки и меню, которые легко находятся на экране и понятны для пользователя;
* четкое оформление страниц с удобным размещением элементов управления для быстрого доступа к функциям;
* интерфейс должен быть удобным для работы как новичков, так и опытных пользователей, не требующим дополнительного обучения.

Функциональность:

* создание и назначение задач для проектов: пользователи должны иметь возможность быстро создавать новые задачи, определять их приоритеты, устанавливать сроки выполнения и назначать ответственных;
* отслеживание прогресса задач и проектов: приложение должно предоставлять пользователю возможность видеть текущий статус выполнения задач и проектов, что поможет понять, насколько они приближаются к завершению.

Дизайн:

* приятный и современный внешний вид: дизайн интерфейса должен быть привлекательным для пользователя, с использованием современных дизайнерских трендов;
* простота оформления страниц: элементы управления и информационные блоки должны быть четко разделены и оформлены для легкости восприятия информации;
* адаптивность: дизайн приложения должен быть адаптирован для корректного отображения на различных устройствах, таких как компьютеры, планшеты и смартфоны.

Безопасность:

* защита данных: приложение должно обеспечивать защиту конфиденциальных данных пользователей и информации о проектах с помощью шифрования данных и других средств безопасности;
* механизмы аутентификации и авторизации: приложение должно иметь механизмы аутентификации пользователей для контроля доступа к функциональности приложения и защиты от несанкционированного доступа.

## 2.3 Системные требования к оборудованию

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

* характеристики процессора: рекомендуется использовать многоядерный процессор с высокой тактовой частотой для обеспечения плавной работы и быстрой компиляции кода;
* оперативная память (RAM): размер оперативной памяти зависит от объема и сложности приложения. Обычно рекомендуется иметь не менее 8 ГБ оперативной памяти для комфортной работы;
* жесткий диск (HDD/SSD): обычно достаточно 500 ГБ свободного места на диске для установки необходимых инструментов разработки, зависимостей и хранения проектов;
* операционная система: React и TypeScript поддерживаются на различных операционных системах, включая Windows, macOS и различные дистрибутивы Linux. Рекомендуется использовать последние версии операционных систем с обновленными пакетами;
* браузеры: для разработки и тестирования React-приложений может потребоваться установленный и обновленный веб-браузер, такой как Google Chrome, Mozilla Firefox или Microsoft Edge;
* среда разработки (IDE): разработка на TypeScript обычно проводится в среде разработки с поддержкой TypeScript, такой как Visual Studio Code, WebStorm или Atom;
* Node.js и npm (Node Package Manager): для управления зависимостями и сборки проекта с помощью инструментов типа Webpack или Parcel, необходимо установить Node.js и npm;
* Git: рекомендуется использовать систему контроля версий Git для отслеживания изменений в коде и управления исходным кодом проекта.

# 3 Проектирование информационной системы

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

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

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

## 3.1 Разработка и описание сценариев использования программного продукта

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

На диаграмме прецедентов представлены как различные типы пользователей или внешние системы, которые могут взаимодействовать с системой. Прецеденты же представляют собой конкретные действия, которые пользователи могут выполнить с помощью системы. Эти действия обычно описываются в терминах того, что пользователь хочет сделать с системой, а не как система это делает.

Диаграмма прецедентов помогает определить функциональные требования к системе, описывая ее в терминах взаимодействия между пользователями и системой. Она является основой для дальнейшего анализа и проектирования системы, включая определение архитектуры, проектирование интерфейсов пользователя и определение процессов бизнес-логики.

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

На рисунке 3.1 изображена диаграмма вариантов использования.



Рисунок 3.1 – Диаграмма вариантов использования

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

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

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

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

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

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

## 3.2 Потоки данных в информационной системе

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

* внешние сущности: представляют собой объекты или субъекты, которые находятся за пределами информационной системы, но взаимодействуют с ней. Это могут быть пользователи, другие информационные системы, клиенты, поставщики и т. д. Важно учитывать требования и потребности этих внешних сущностей при проектировании и разработке информационной системы;
* процессы: представляют собой серию шагов или действий, которые выполняются в информационной системе для достижения определенной цели. Например, это может быть процесс обработки заказа, процесс регистрации пользователя, процесс анализа данных и т. д. Каждый процесс может включать в себя взаимодействие с внешними сущностями, обработку данных и генерацию выходных результатов;
* потоки данных: потоки данных представляют собой направленное движение информации или данных от одной точки к другой в информационной системе. Они описывают, как данные передаются от внешних сущностей к системе, как обрабатываются в процессах и как сохраняются в хранилище данных. Потоки данных могут быть как входящими, так и исходящими, а также могут проходить через различные этапы обработки и преобразования;
* хранилище данных: хранилище данных представляет собой место, где данные сохраняются и доступны для последующего использования. Это может быть база данных, файловая система, кэш-память и т. д. Хранилище данных обеспечивает удобный доступ к информации, а также обеспечивает ее сохранность и целостность. Важно разработать структуру хранилища данных, которая соответствует требованиям системы и обеспечивает эффективное управление информацией.

Эти компоненты взаимодействуют между собой, образуя целостную информационную систему, которая обеспечивает обработку, хранение и передачу данных для поддержки бизнес-процессов и достижения целей организации.

## 3.3 Выбор архитектуры разрабатываемого программного продукта

## 3.4 Проектирование реляционной базы данных на основе инфологической модели

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

Основные компоненты концептуальной модели данных включают:

* сущности (или объекты): это основные объекты, которые описываются в системе;
* атрибуты: каждая сущность имеет свои характеристики, называемые атрибутами;
* взаимосвязи: это связи между сущностями, определяющие, как они связаны друг с другом.

Цель концептуальной модели данных – предоставить понятное и независимое от технических деталей представление о данных в системе. Она используется на начальном этапе проектирования информационных систем для определения основных концепций и требований к базе данных. Концептуальная модель данных обычно разрабатывается совместно с заказчиком или конечными пользователями, чтобы удостовериться, что она правильно отражает их потребности и бизнес-процессы.

На рисунке 3.3 изображена концептуальная модель базы данных.



Рисунок 3.3 – Концептуальная модель базы данных

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

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

Следующим этапом необходимо сформировать логическую модель данных.

Логическая модель данных – это абстрактное представление данных, используемых в системе, которое отображает их связи, структуру и ограничения. Она позволяет описать данные, используемые в системе, независимо от физического хранения и способа доступа к ним. Логическая модель данных является важным инструментом при проектировании баз данных и информационных систем, так как она помогает разработчикам понять структуру данных и определить связи между ними. При создании логической модели данных учитываются требования к производительности, безопасности и доступности системы.

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

Таблица 3.1 – Атрибуты статусов проекта

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

Таблица 3.2 – Атрибуты этапов

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

Таблица 3.3 – Атрибуты статусов задач

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

Таблица 3.4 – Атрибуты проектов

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

Таблица 3.5 – Атрибуты участников проекта

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

Таблица 3.6 – Атрибуты этапов проекта

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

Таблица 3.7 – Атрибуты статусов задач для этапа проекта

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

Таблица 3.8 – Атрибуты задач

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

Таблица 3.9 – Атрибуты исполнителей задач

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

Таблица 3.10 – Атрибуты статусов сотрудника

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

Таблица 3.11 – Атрибуты должностей

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

Таблица 3.12 – Атрибуты подразделений

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

Таблица 3.13 – Атрибуты сотрудников

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

На рисунке 3.4 представлена физическая модель.

## 3.5 Прототипирование пользовательского интерфейса

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

Прототипирование создается для того, чтобы продемонстрировать распределение элементов на странице и их функциональные взаимосвязи. Это помогает команде разработки и заказчику лучше понять структуру будущего продукта и согласовать его основные характеристики еще на ранних этапах проекта.

Основная цель – определить расположение элементов интерфейса и их функциональные возможности без учета дизайна. Он позволяет сосредоточиться на пользовательском опыте и функциональности продукта, а также упрощает процесс коммуникации между разработчиками, дизайнерами и заказчиками.

Прототипирование пользовательского интерфейса обычно создается в начале проекта после того, как сформулированы основные требования и цели. Это позволяет установить общую структуру продукта и обозначить основные функциональные возможности до того, как приступить к более детальной разработке и дизайну.

На рисунке 3.1 изображен прототип страницы сотрудников.

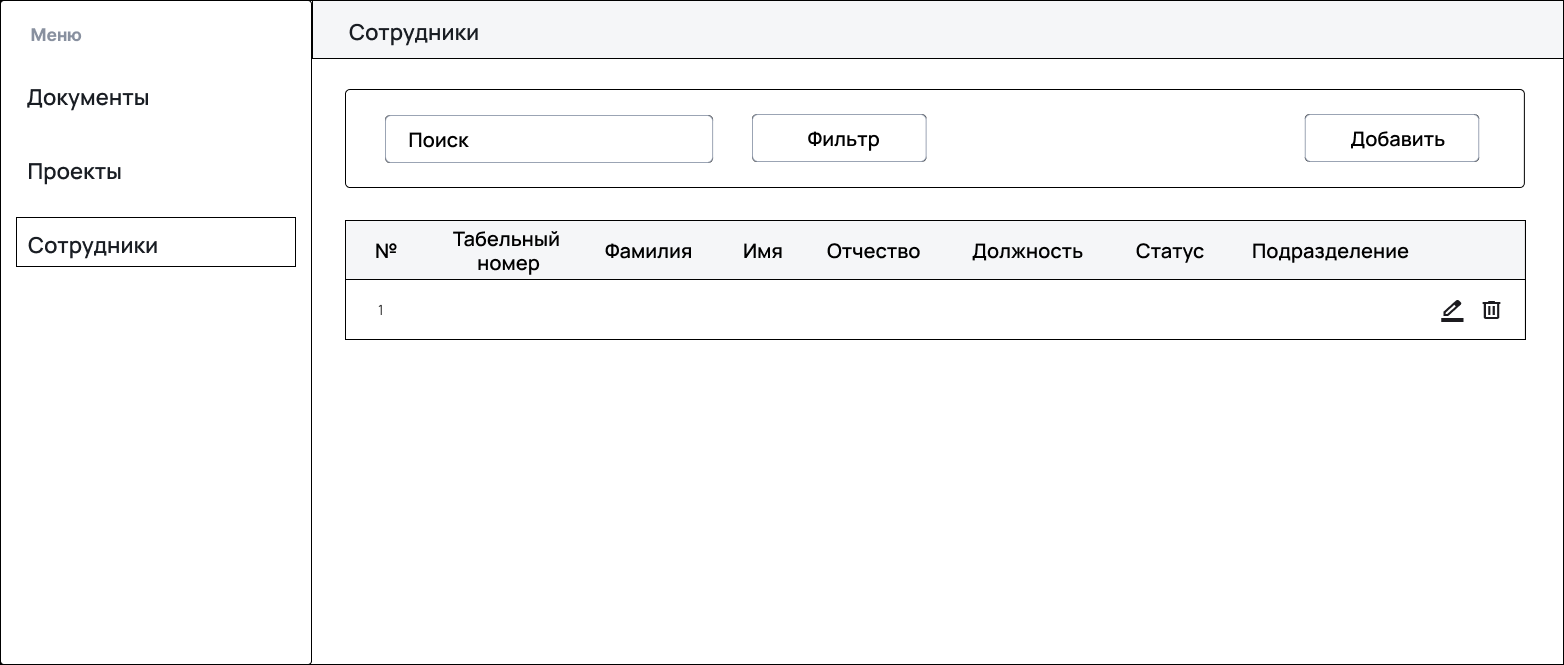


Рисунок 3.1 – Прототип страницы сотрудников

Прототип страницы сотрудников – это веб-страница, предназначенная для управления информацией о сотрудниках. В левой части страницы есть меню, которое позволяет пользователям переходить к различным разделам, таким как документы, проекты и сотрудники. Основной раздел страницы представляет собой таблицу, в которой перечислены все сотрудники компании. Каждая строка в таблице содержит сведения об одном сотруднике, такие как табельный номер, фамилия, имя, отчество, должность, статус, подразделение. В правой части страницы также есть кнопки, которые позволяют пользователям добавлять, редактировать и удалять сотрудников.

На рисунке 3.2 изображен прототип формы добавления сотрудника.

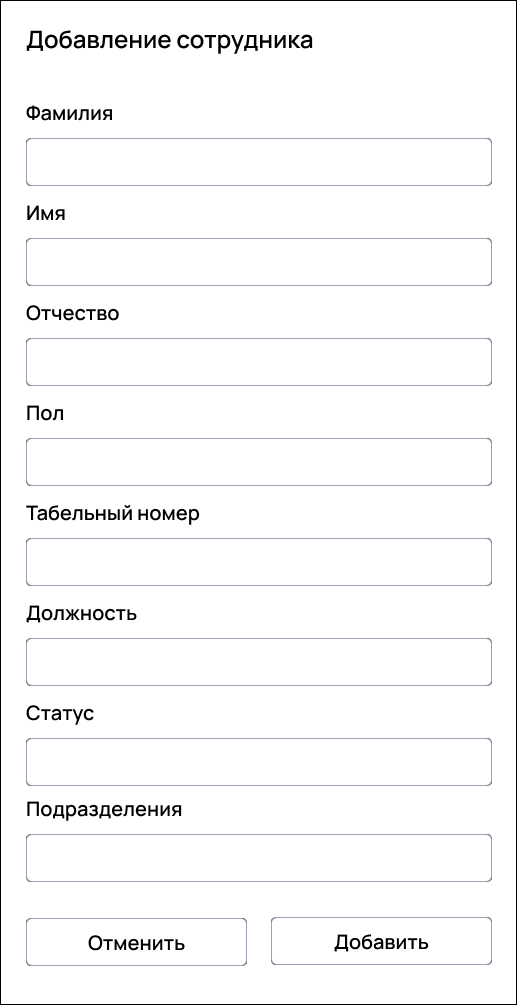


Рисунок 3.2 – Прототип формы добавления сотрудника

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 3.3 изображен прототип страницы проектов.

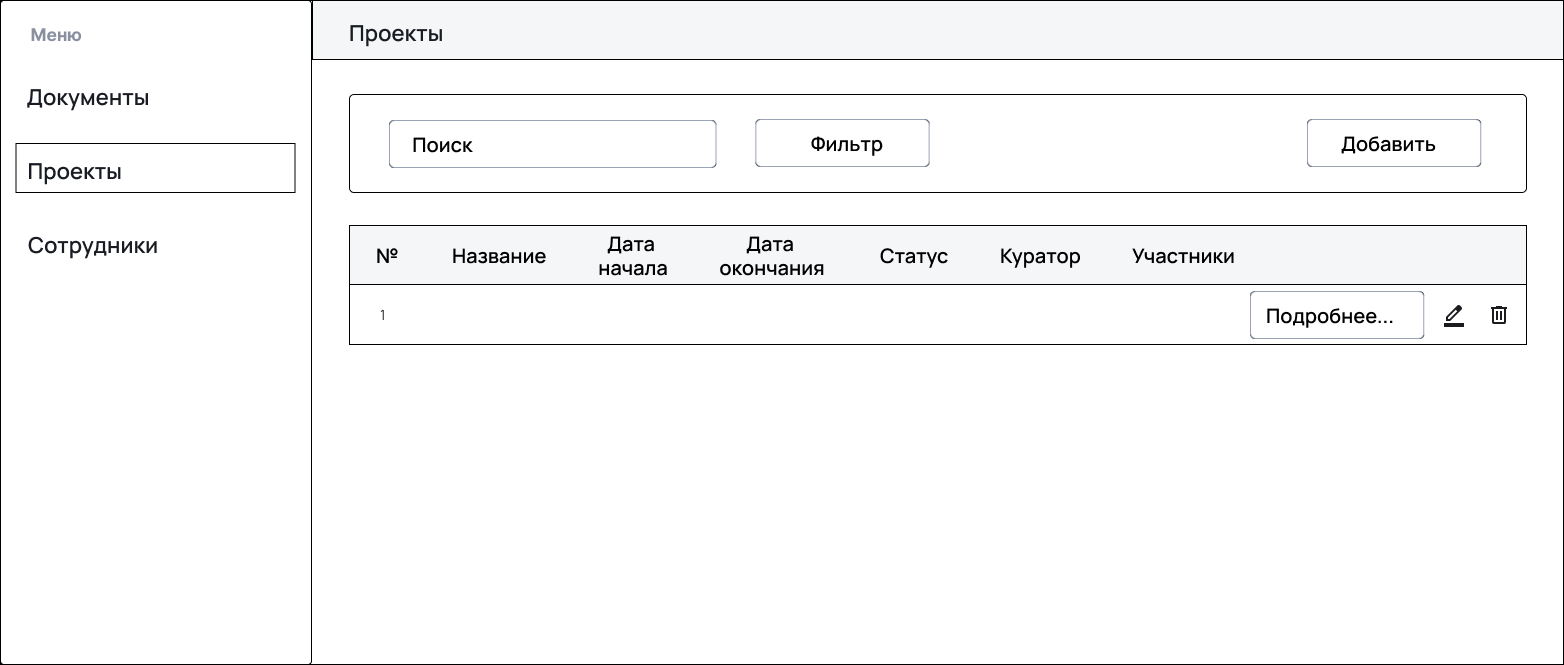


Рисунок 3.3 – Прототип страницы проектов

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 3.4 изображен прототип формы добавления проекта.

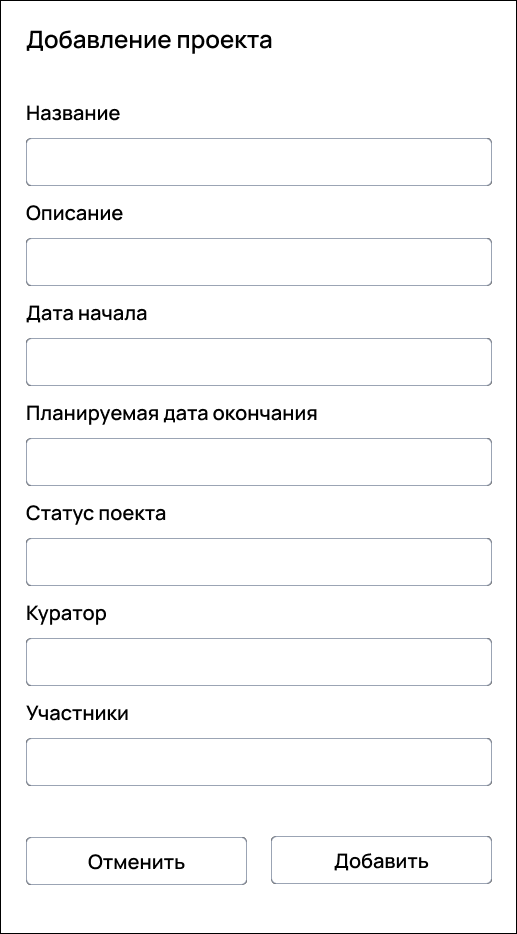


Рисунок 3.4 – Прототип формы добавления проекта

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

На рисунке 3.5 изображен прототип страницы информации о проекте.

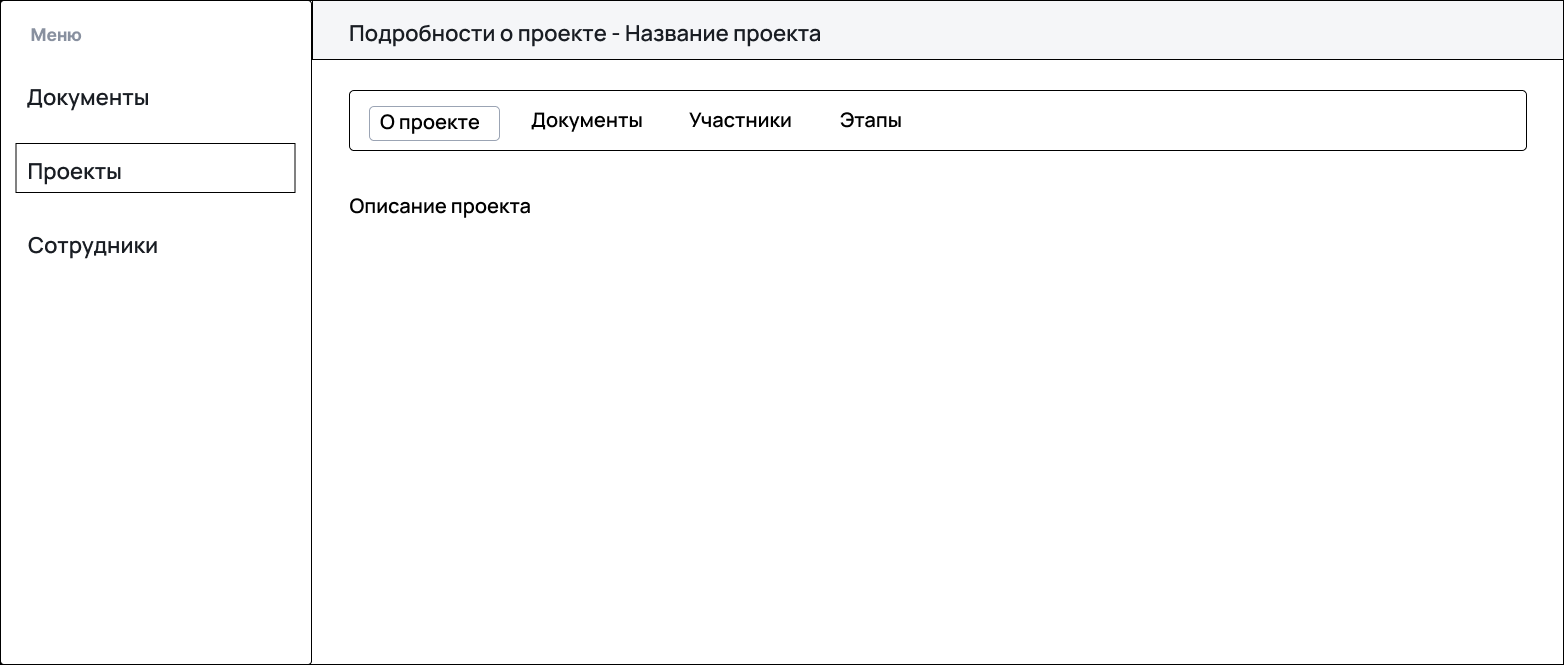


Рисунок 3.5 – Прототип страницы информации о проекте

На этом изображении вы можете увидеть пример информационной страницы проекта, веб-интерфейса, который создан для того, чтобы предоставить подробный обзор определенного проекта. Эта страница, как правило, включает в себя различные разделы, которые в совокупности дают полное представление о проекте.

На рисунке 3.6 изображен прототип страницы участников проекта.

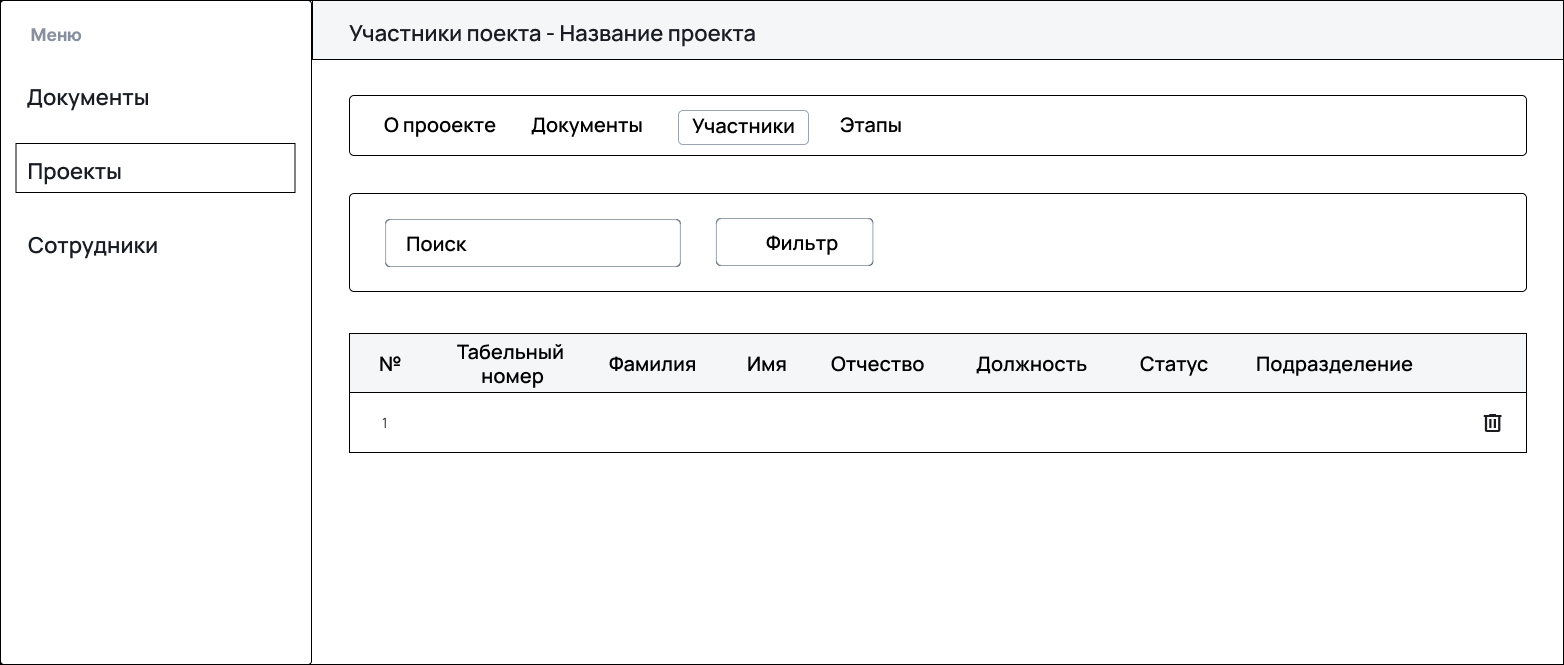


Рисунок 3.6 – Прототип страницы участников проекта

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 3.7 изображен прототип страницы этапов проекта.

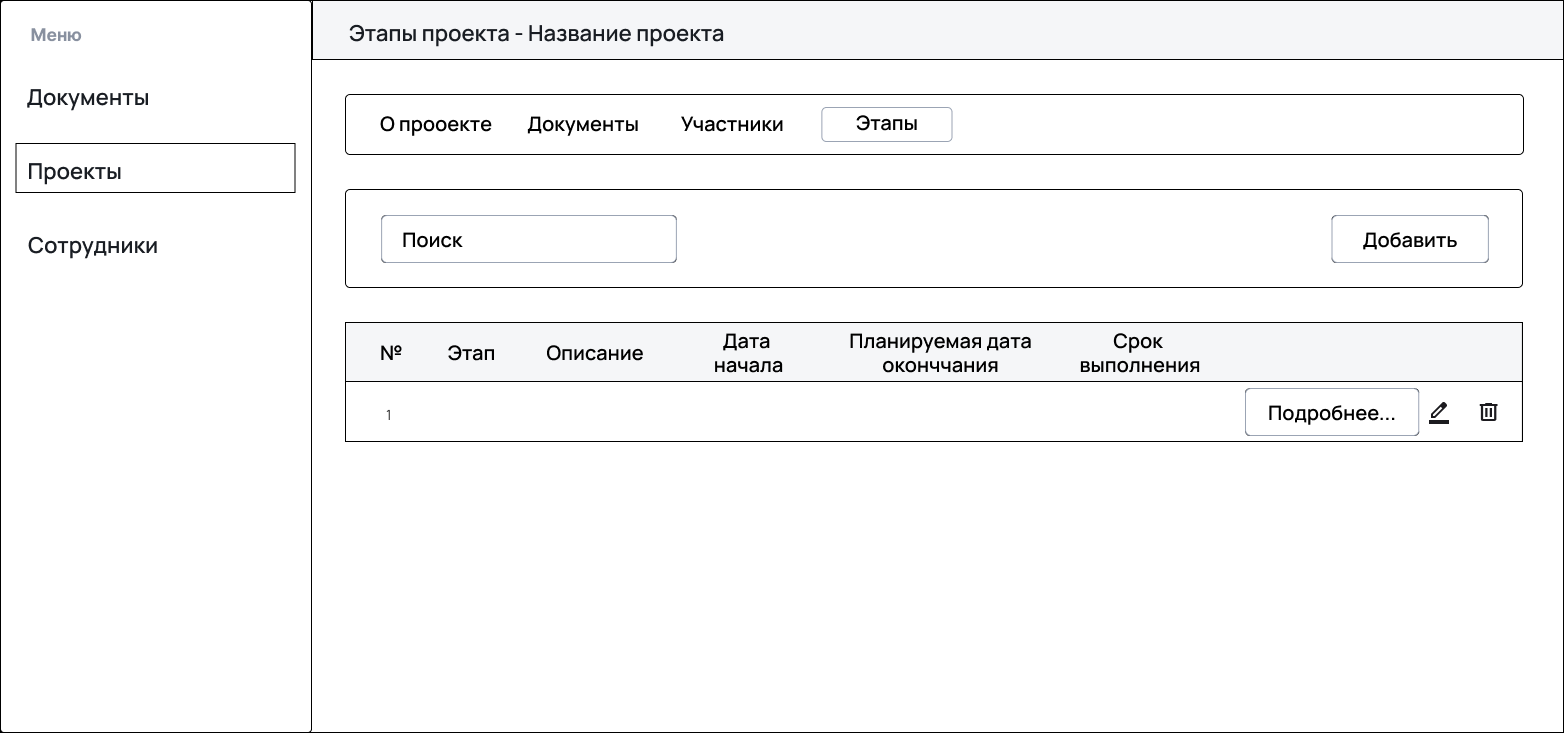


Рисунок 3.7 – Прототип страницы этапов проекта

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 3.8 изображен прототип формы добавления этапа проекта.

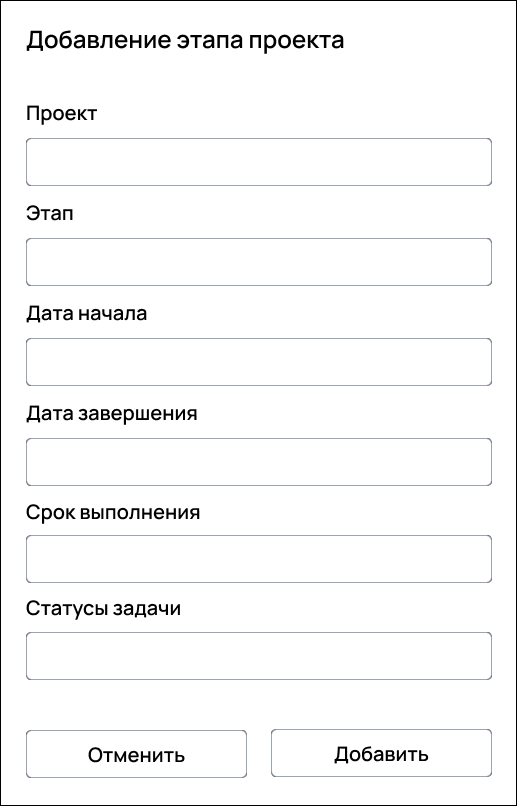


Рисунок 3.8 – Прототип формы добавления этапа проекта

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 3.9 изображен прототип страницы подробности об этапе.

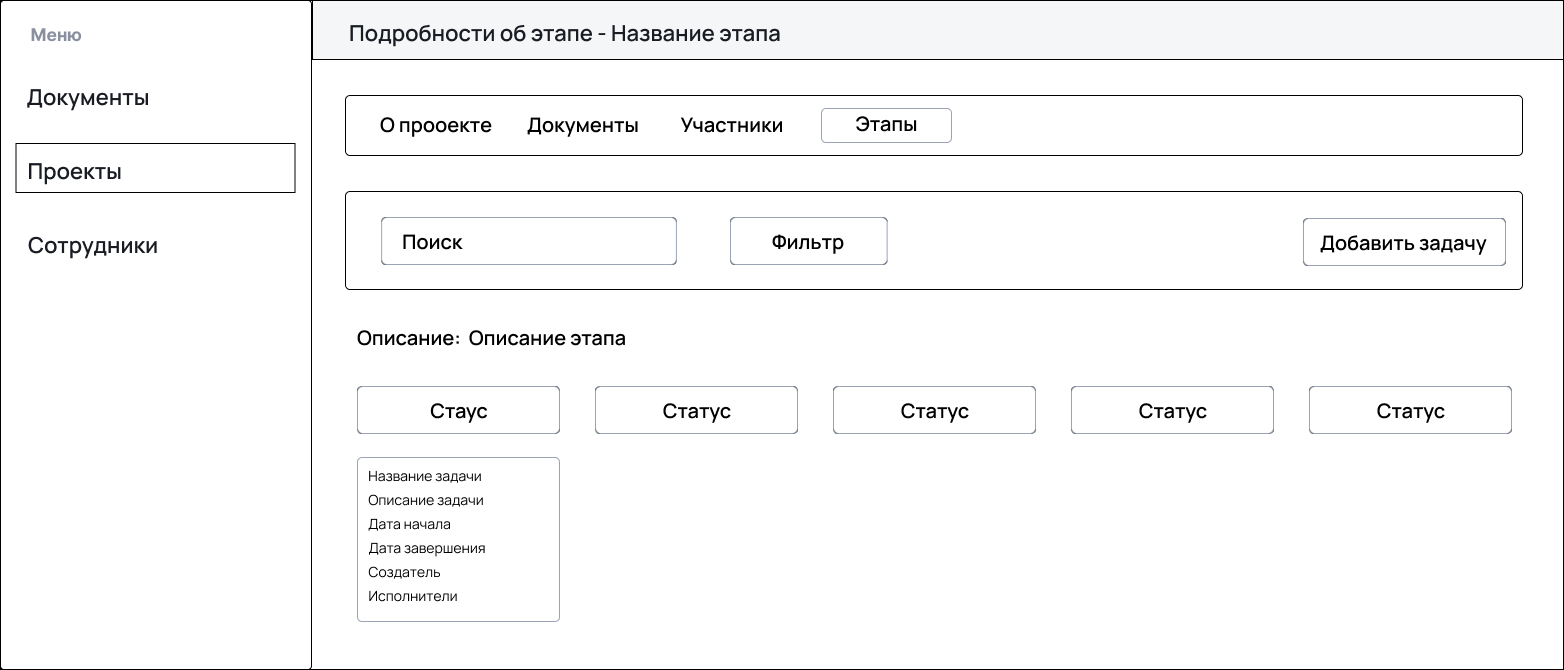


Рисунок 3.9 – Прототип страницы подробности об этапе

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 3.10 изображен прототип формы добавления задачи.

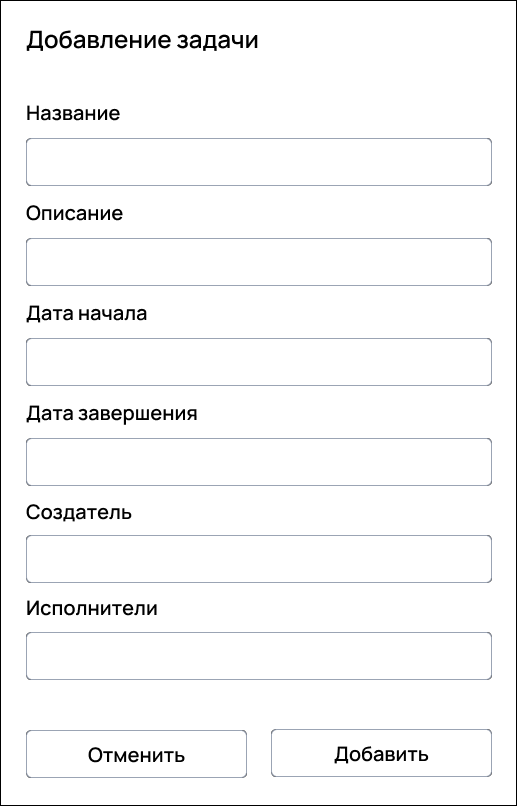


Рисунок 3.10 – Прототип формы добавления задачи

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 3.11 изображен прототип информации о задаче.

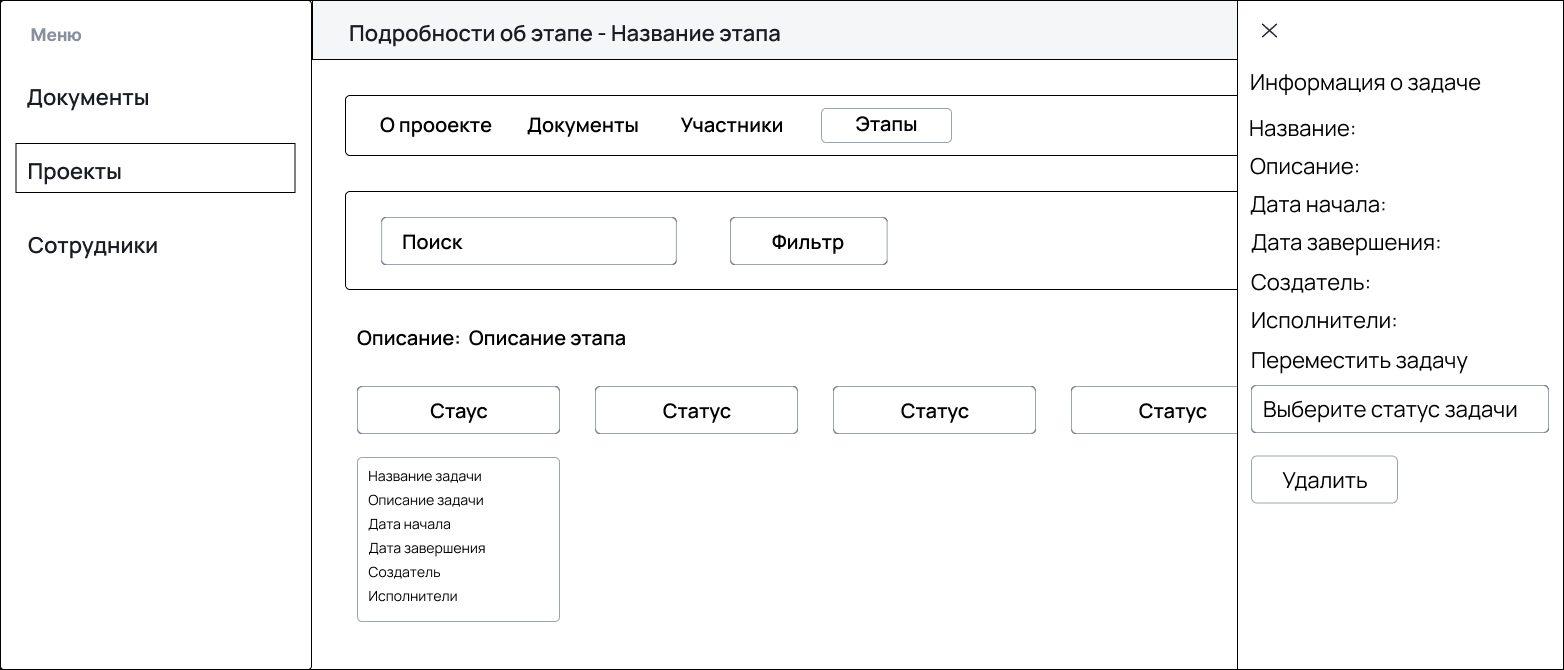


Рисунок 3.11 – Прототип информации о задаче

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

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

# 4 Разработка программного обеспечения

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

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

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

## 4.1 Обоснование выбора инструментальных средств разработки программного продукта

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

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

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

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

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, разработанный и поддерживаемый компанией Microsoft. Он предоставляет обширный набор функций и инструментов для удобной и эффективной разработки программного обеспечения. Вот некоторые ключевые особенности и возможности Visual Studio Code:

* интерфейс пользователя: интерфейс Visual Studio Code дружелюбен к пользователю и легок в освоении. Он состоит из нескольких основных элементов, таких как редактор кода, боковая панель с файловым деревом и инструментами, строка поиска и навигации, а также верхнее меню с различными командами и инструментами;
* редактор кода: одним из ключевых элементов VS Code является его мощный редактор кода. Он поддерживает подсветку синтаксиса для множества языков программирования, автозавершение кода, быстрое переходы к определению переменных и функций, а также множество других функций, повышающих производительность при написании кода;
* расширения и плагины: система расширений Visual Studio Code позволяет добавлять новые функции и интеграции с другими инструментами и сервисами. В магазине расширений можно найти тысячи плагинов для поддержки различных языков программирования, интеграции с системами контроля версий, средств отладки, сборки и многое другое;
* интеграция с Git: Visual Studio Code имеет встроенную поддержку системы контроля версий Git. Это позволяет разработчикам эффективно работать с репозиториями Git, отслеживать изменения, выполнять коммиты и слияния, а также смотреть историю изменений прямо из редактора кода;
* отладка: редактор Visual Studio Code обеспечивает возможности отладки приложений на различных языках программирования. Он интегрируется с различными средствами отладки, позволяя разработчикам отслеживать и исправлять ошибки в своем коде;
* интеграция с облачными сервисами: Visual Studio Code интегрируется с различными облачными сервисами, такими как Microsoft Azure. Это обеспечивает возможность разработки и развертывания приложений в облаке прямо из редактора кода;
* кроссплатформенность: редактор кода поддерживает операционные системы Windows, macOS и Linux, что делает его универсальным инструментом для разработки на различных платформах;
* сообщество и поддержка: Visual Studio Code имеет активное сообщество пользователей и разработчиков. Это означает, что всегда можно найти помощь, руководства и решения проблем через форумы, блоги, социальные сети и другие ресурсы [4].

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

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

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

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

Основные преимущества MongoDB включают в себя:

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

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

## 4.2 Описание алгоритма программы

## 4.3 Описание интерфейса программного продукта

Интерфейс программного продукта – это совокупность элементов и функций, предназначенных для взаимодействия пользователя с программой. Он обеспечивает способ ввода и вывода информации, управления приложением, а также обеспечивает пользователю доступ к функциональности программы [5].

## 4.3 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:

# 5 Экономическая часть

# 6 Охрана труда и техники безопасности

# Заключение

# Список использованных источников

# Приложение А