

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 документооборота для ИП
«Ромашов Софт»

Специальность 09.02.07 Информационные системы и программирование

Студент

(подпись)

А.Д. Городинец

Руководитель проекта

(подпись)

Р.В. Ромашов

Консультант по
экономической части

(подпись)

Е.А. Галицына

Нормоконтроль

(подпись)

И.Н. Власенко

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:

цикловой комиссией профессионального
информационного цикла
Протокол № 6 от «15» января 2024 г.

Председатель ЦК _____ / И.Н. Борзенкова /

УТВЕРЖДАЮ:

И.о. зам. директора по УР

_____ / Н.В. Погорелова /

**ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Студента Городинец Алины Дмитриевны

Специальность 09.02.07 Информационные системы и программирование

1. Тема: Разработка веб-приложения корпоративного документооборота для ИП «Ромашов Софт»

утверждена приказом по техникуму №147 (ИСП-20)/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 получил «1» марта 2024 г.

(подпись)

/ А.Д. Городинец /

ОДОБРЕНО:

цикловой комиссией профессионального
информационного цикла
Протокол № 6 от «15» января 2024 г.

УТВЕРЖДАЮ:

И.о. зам. директора по УР

_____ / Н.В. Погорелова /

Председатель ЦК _____ / И.Н. Борзенкова /

**КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Студента Городинец Алины Дмитриевны

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

_____ (подпись)

/ Р.В. Ромашов /

«1» марта 2024 г.

Студент

_____ (подпись)

/ А.Д. Городинец /

«1» марта 2024 г.

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**

Студента Городинец Алины Дмитриевны

Тема дипломного проекта Разработка веб-приложения корпоративного документооборота 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

(подпись)

/ Р.В. Ромашов /

С отзывом ознакомлен

(подпись)

/ А.Д. Городинец /

«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 Городинец Алины Дмитриевны

Тема дипломного проекта Разработка веб-приложения корпоративного документооборота 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

(подпись)

/ И.О. Фамилия /

С рецензией ознакомлена

(подпись)

/ И.О. Фамилия /

«10» июня 2024 года

СОДЕРЖАНИЕ

| | |
|-------------------------------------------------------------------------------------------|----|
| Введение..... | 5 |
| 1 Техничко-экономическая характеристика объекта | 6 |
| 1.1 Общие сведения о предприятии | 6 |
| 1.2 Организационная структура предприятия..... | 6 |
| 2 Сбор, анализ и формирование требований к информационной системе | 8 |
| 2.1 Цели, назначение и функции программного продукта | 8 |
| 2.2 Требования к функциональности и дизайну программного продукта..... | 8 |
| 2.3 Системный требования к оборудованию..... | 10 |
| 3 Проектирование информационной системы | 12 |
| 3.1 Разработка и описание сценариев использования программного продукта.... | 13 |
| 3.2 Потоки данных в информационной системе | 15 |
| 3.3 Выбор архитектуры разрабатываемого программного продукта | 16 |
| 3.4 Проектирование реляционной базы данных | 16 |
| 3.5 Прототипирование пользовательского интерфейса | 23 |
| 4 Разработка программного обеспечения | 29 |
| 4.1 Обоснование выбора инструментальных средств разработки программного продукта | 30 |
| 4.2 Описание алгоритма программы | 35 |
| 4.3 Описание интерфейса программного продукта | 35 |
| 4.4 Инструкция по эксплуатации программного продукта | 42 |
| 5 Экономическая часть | 45 |
| Охрана труда и техника безопасности | 46 |
| Заключение | 49 |
| Список использованных источников | 50 |
| Приложение | 51 |

| | | | | | | | | | | | |
|-----------|--|-----------------|---------|------|------------------------------------------------------------------------------------------|----------------------------------|---|------|---|--------|----|
| | | | | | ДП.09.02.07.24.??ПЗ | | | | | | |
| | | | | | | | | | | | |
| | | № докум. | Подпись | Дата | | | | | | | |
| Разраб. | | Городинец А.Д. | | | Разработка веб-приложения корпоративного документооборота для ИП «Ромашов Софт» | Лит. | | Лист | | Листов | |
| Провер. | | Ромашов Р.В. | | | | | у | | 4 | | ?? |
| | | | | | | ГБПОУ РО «ВТИТБид» гр. ИСП-20 | | | | | |
| Н. контр. | | Власенко И.Н. | | | | | | | | | |
| Утв. | | Погорелова Н.В. | | | | | | | | | |

Введение

Современные организации сталкиваются с растущим объемом документации, что создает необходимость в эффективном управлении документами для обеспечения плавного хода бизнес-процессов. Однако традиционные методы управления документами, такие как бумажные архивы или файловые системы, часто становятся устаревшими и неэффективными в условиях современного цифрового мира. Даже среди передовых технологий, многие корпорации сталкиваются с проблемами в процессе документооборота, что приводит к задержкам, ошибкам и потере ценного времени и ресурсов. В ответ на эти вызовы, разработка современного корпоративного веб-приложения для документооборота становится необходимостью.

Целью дипломного проекта является разработка корпоративного веб-приложения для документооборота, которое будет решать текущие проблемы и улучшать процессы управления документами в корпоративной среде. Это приложение предназначено для использования в различных отраслях, начиная от малых компаний и заканчивая крупными корпорациями, и направлено на оптимизацию рабочих процессов и повышение производительности персонала.

Проблемы, с которыми сталкиваются компании при управлении документами, включают в себя сложности в организации и поиске документов, риски утери или повреждения ценной информации, задержки в передаче документов между отделами или сотрудниками, а также проблемы с обеспечением безопасности и конфиденциальности данных.

Веб-приложение будет предоставлять широкий спектр функциональности, включая возможность создания, редактирования, подписания и отправки документов, а также автоматизацию определенных рабочих процессов, уведомления о сроках и напоминания о задачах, аналитические отчеты о документообороте и многое другое.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.??.ПЗ | Лист |
| | | | | | | 5 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

1 Технико-экономическая характеристика объекта

1.1 Общие сведения о предприятии

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

Виды программного обеспечения, которые разрабатываются на предприятии:

- веб-приложения: сайты, веб-сервисы, мобильные приложения;
- десктопные приложения: программы для Windows, macOS, Linux;
- мобильные приложения: iOS, Android.

Так же предприятия оказывает услуги:

- доработка и модернизация существующего ПО;
- техническая поддержка ПО;
- консультационные услуги.

При разработке программного обеспечения необходимо выполнить следующие этапы:

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

1.2 Организационная структура предприятия

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

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.??.ПЗ | Лист |
| | | | | | | 6 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

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.

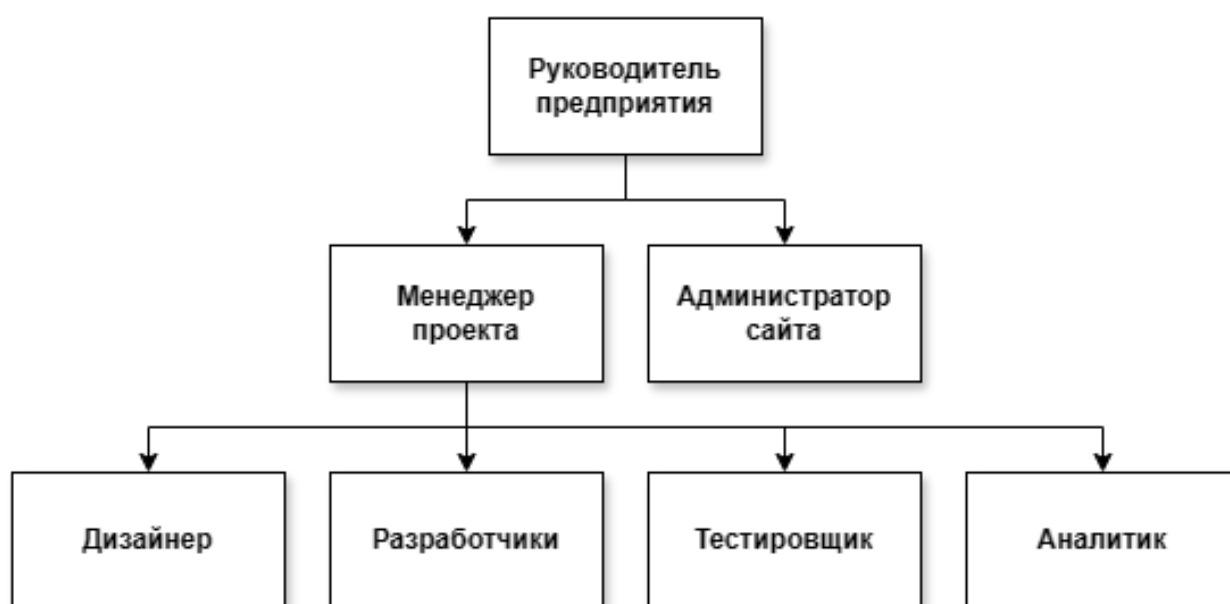


Рисунок 1.1 – Организационная структура предприятия

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 7 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

2 Сбор, анализ и формирование требований к информационной системе

На основе анализа требований можно определить следующие спецификации для программного модуля документооборота:

- язык программирования – TypeScript;
- база данных – MongoDB;
- интерфейс пользователя – веб-приложение;
- доступ к системе должен быть защищен паролем и логином;
- программный модуль должен поддерживать многопользовательскую работу с возможностью управления правами доступа.

2.1 Цели, назначение и функции программного продукта

2.2 Требования к функциональности и дизайну программного продукта

Требования к функциональности документооборота могут быть разнообразными, в зависимости от конкретных потребностей и особенностей бизнес-процессов организации. Вот несколько основных категорий требований:

- управление документами: возможность загрузки, хранения, просмотра и редактирования документов в различных форматах (текстовые документы, таблицы и т. д.). Организация документов по категориям, тегам или другим атрибутам для удобства поиска и навигации;

– электронное подписание и утверждение: возможность электронного подписания документов с использованием сертификатов. Механизмы утверждения документов, в том числе с возможностью установки сроков и напоминаний;

- автоматизация бизнес-процессов: настройка прав доступа и ролей пользователей для автоматического регулирования рабочих процессов;

- безопасность и конфиденциальность: защита конфиденциальных данных и обеспечение безопасности передачи информации. Аудит доступа к документам

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.??.ПЗ | Лист |
| | | | | | | 8 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

и действий пользователей для обеспечения соответствия законодательным требованиям.

Эти требования могут быть адаптированы и дополнены в соответствии с конкретными потребностями и целями организации, использующей программное обеспечение для документооборота.

Требования к дизайну для документооборота играют ключевую роль в обеспечении удобства использования, эффективности работы пользователей и достижении поставленных бизнес-целей. Вот несколько основных аспектов, которые следует учитывать при определении требований к дизайну:

- интуитивно понятный интерфейс: дизайн должен быть интуитивно понятным для пользователей всех уровней навыков, даже для тех, кто впервые использует программу. Это включает в себя понятную структуру меню, понятные иконки и подсказки, которые помогают пользователям быстро ориентироваться;

- простота и минимализм: дизайн должен быть простым и минималистичным, без избыточных элементов и информации. Основная функциональность должна быть легко доступна и не затеряна среди лишних деталей;

- совместимость: элементы дизайна должны быть консистентными по всему приложению, что обеспечивает единый стиль и согласованный пользовательский опыт. Это включает в себя единые шрифты, цветовые схемы, типы элементов управления;

- адаптивность и отзывчивость: дизайн должен быть адаптивным и отзывчивым, чтобы обеспечить корректное отображение на различных устройствах и разрешениях экранов. Это важно для удобства использования приложения на различных устройствах, включая компьютеры, планшеты и мобильные телефоны;

- эргономичность: дизайн должен быть эргономичным, учитывая привычные для пользователей паттерны взаимодействия и удобство выполнения повседневных задач. Это включает в себя удобство навигации, расположение элементов управления и логическую организацию контента;

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 9 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– визуальная привлекательность: дизайн должен быть визуально привлекательным и современным, чтобы привлечь внимание пользователей и создать положительное впечатление о продукте. Это включает в себя использование качественных графических элементов, анимации и эффектов;

– доступность: для всех пользователей, включая людей с ограниченными возможностями. Это включает в себя соблюдение стандартов доступности веб-контента, использование соответствующих цветовых схем, шрифтов и размеров элементов для обеспечения удобства использования для всех пользователей.

Учитывая эти аспекты, проектирование дизайна для документооборота поможет создать удобное, эффективное и привлекательное приложение, которое будет соответствовать потребностям пользователей и бизнес-задачам организации.

2.3 Системные требования к оборудованию

Системные требования к оборудованию для разработки React-приложений на TypeScript зависят от размера и сложности проекта, а также от инструментов и технологий, используемых в процессе разработки. Вот общие рекомендации:

– процессор: рекомендуется многоядерный процессор с тактовой частотой не менее 2.0 ГГц. Это обеспечит достаточную производительность при компиляции кода, запуске среды разработки и выполнении других ресурсоемких задач;

– оперативная память: минимальный объем оперативной памяти для разработки React-приложений на TypeScript составляет 4 ГБ. Однако, для работы с крупными проектами рекомендуется 8 ГБ или более, чтобы обеспечить плавную работу среды разработки и выполнение нескольких задач одновременно;

– хранилище: для хранения исходного кода, зависимостей и временных файлов проекта потребуется свободное место на жестком диске или SSD. Рекомендуемый объем – не менее 50 ГБ;

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 10 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– графический процессор: для разработки веб-приложений на React с использованием TypeScript не требуется мощного графического процессора. Однако, если в проекте используется работа с графикой или анимацией, GPU может повысить производительность работы;

– операционная система: поддерживаемые операционные системы: Windows, macOS, Linux. Рекомендуется использовать операционную систему, с которой вы знакомы и которая поддерживает необходимые инструменты разработки;

– браузеры: для тестирования и отладки приложений рекомендуется иметь установленные и актуальные версии популярных веб-браузеров, таких как Google Chrome, Mozilla Firefox, Microsoft Edge;

– интернет-соединение: для загрузки зависимостей, обновлений и доступа к онлайн-ресурсам необходимо стабильное и быстрое интернет-соединение;

– Node.js: предоставляет среду выполнения JavaScript на сервере, что позволяет разрабатывать и запускать серверную часть React-приложений. Он также включает в себя npm, который является крупнейшим репозиторием библиотек и инструментов для JavaScript. С его помощью разработчики могут управлять зависимостями проекта, устанавливать необходимые пакеты, выполнять скрипты сборки и многое другое;

– Git: является распределенной системой управления версиями, которая позволяет отслеживать изменения в коде, работать с ветками, управлять историей изменений и сотрудничать с другими разработчиками. Для React-приложений Git позволяет эффективно управлять и поддерживать кодовую базу, осуществлять разработку в команде, проводить ревью кода и разрешать конфликты при слиянии изменений.

Эти системные требования обеспечат комфортную и эффективную работу над React-приложениями на TypeScript.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 11 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

3 Проектирование информационной системы

Проектирование информационной системы – это процесс создания системы, которая позволяет эффективно управлять документами в организации, начиная с их создания и заканчивая хранением и уничтожением. Вот основные этапы проектирования такой системы:

- анализ потребностей и требований: важно понять потребности и цели организации относительно документооборота. Это включает выявление типов документов, потоков работы, ролей пользователей и требований к безопасности и доступу к документам;

- проектирование информационной модели: на основе анализа потребностей создается информационная модель, которая определяет структуру данных, используемых в системе. Это включает определение сущностей (например, документы, пользователи, отделы), их атрибутов и связей между ними;

- проектирование функциональных возможностей: определяются основные функции и процессы, которые должны поддерживаться в системе документооборота. Это может включать создание, редактирование, подпись, хранение и поиск документов;

- выбор технологий и инструментов: на основе требований к системе выбираются подходящие технологии и инструменты для реализации проекта. Это может быть использование готовых решений, таких как электронный документооборот, или разработка собственной системы с использованием определенных языков программирования и фреймворков;

- проектирование пользовательского интерфейса: создается удобный и интуитивно понятный пользовательский интерфейс, который позволит пользователям легко взаимодействовать с системой. Это включает разработку макетов, форм, списков, меню и других элементов интерфейса;

- разработка системы и интеграция: на этом этапе реализуется разработка самой системы на основе предыдущих этапов проектирования. Это включает

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 12 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

создание базы данных, разработку бизнес-логики, реализацию пользовательского интерфейса и интеграцию с другими системами, если необходимо;

– тестирование и отладка: после разработки система проходит тестирование, чтобы убедиться в ее правильной работе и соответствии требованиям. В случае обнаружения ошибок или недочетов проводится отладка и корректировка;

– внедрение и обучение пользователей: после успешного завершения тестирования система внедряется в организацию, а пользователи проходят обучение по ее использованию. Обычно проводится поддержка и консультации в начальный период использования системы;

– поддержка и сопровождение: после внедрения системы она поддерживается и сопровождается, включая регулярное обновление, резервное копирование данных, обеспечение безопасности и решение проблем, которые могут возникнуть в процессе эксплуатации.

3.1 Разработка и описание сценариев использования программного продукта

Диаграмма прецедентов или диаграмма вариантов использования – это графическое представление функциональных требований к системе, которое описывает, как различные пользователи (актеры) взаимодействуют с системой для достижения определенных целей.

Основные компоненты диаграммы прецедентов:

– актеры: представляют роли или типы пользователей, которые взаимодействуют с системой. Это могут быть конечные пользователи, другие системы или внешние устройства. На диаграмме актеры обычно изображаются в виде простых фигур, обозначающих их роль в системе;

– сценарии использования: диаграмма описывает конкретные действия или задачи, которые выполняются пользователем или системой для достижения определенной цели. Каждая диаграмма представляет собой отдельный

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 13 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

функциональный элемент системы. Они представляются в виде эллипсов с текстом, описывающим функциональность;

- отношения между актерами и диаграммой прецедентов: стрелки между актерами и диаграммой показывают взаимодействие между ними. Если актер использует диаграмму прецедентов, стрелка направлена от актера к диаграмме вариантов использования. Это показывает, что актер инициирует или участвует в выполнении;

- альтернативные пути: в диаграмме можно также показать альтернативные пути выполнения, которые могут возникнуть в результате различных условий или действий пользователя. Это помогает учесть различные сценарии использования системы.

Диаграмма вариантов использования является полезным инструментом для анализа требований и понимания того, как пользователи будут взаимодействовать с системой в различных сценариях использования. Они помогают выделить ключевые функциональности и определить, как система должна вести себя в различных ситуациях.

Ниже представлена диаграмма, описывающая основные функциональные возможности приложения и роли пользователей:

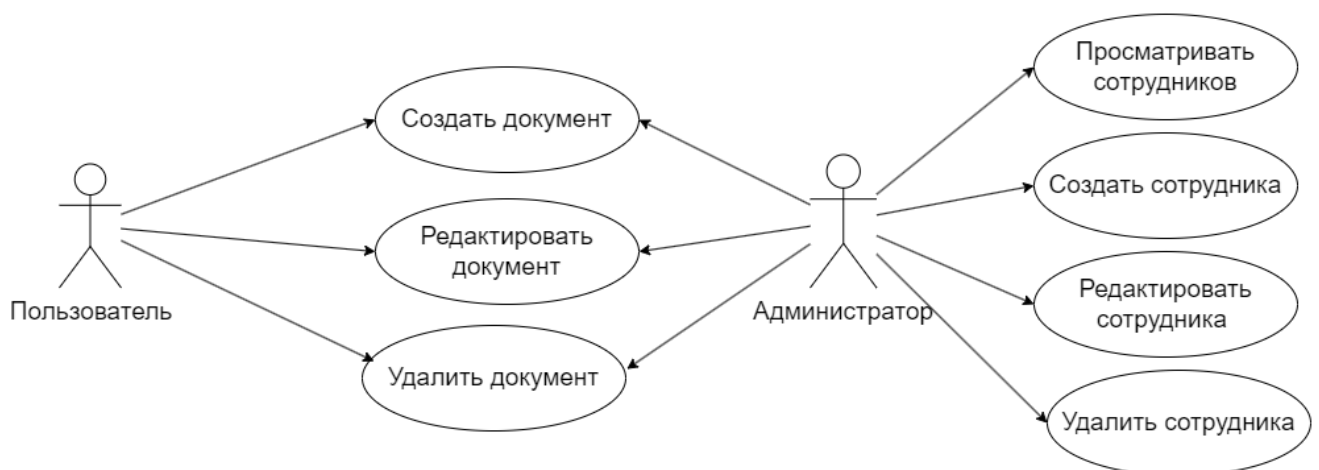


Рисунок 3.1 – Диаграмма вариантов использования

Актеры:

- Пользователь: сотрудник, который работает с документами в системе.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 14 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– Администратор: сотрудник, который имеет дополнительные возможности по работе с документами и сотрудниками.

Взаимодействия:

– Пользователь может создать новый документ, заполнив соответствующую форму.

– Пользователь может редактировать существующий документ, внося изменения в его текст.

– Пользователь может удалить существующий документ.

– Пользователь может просматривать существующие документы.

– Администратор может настраивать параметры системы, войдя в режим администрирования.

– Администратор может создавать сотрудников, заполнив соответствующую форму

– Администратор может редактировать сотрудников, внося изменения в тексте

– Администратор может удалить существующий документ.

– Администратор может просматривать всех сотрудников.

3.2 Потоки данных в информационной системе

Диаграмма потоков данных – это графическое представление потоков данных и процессов в информационной системе. Она используется для иллюстрации потоков информации от входных источников к выходным получателям через различные процессы обработки. Вот ключевые элементы диаграммы потоков данных:

– сущности: представляют внешние источники или получателей данных, с которыми информационная система взаимодействует. На диаграмме они представляются в виде прямоугольников с названием, например, «Пользователь», «Внешняя система»;

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 15 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– процессы: обозначают операции, выполняемые над данными в системе. Они могут включать в себя обработку, хранение, передачу или преобразование данных. Процессы обычно представлены в виде кругов с названием внутри, например, «Аутентификация пользователя», «Генерация отчета», «Обновление базы данных»;

– потоки данных: представляют передачу информации между сущностями и процессами. Они показывают направление движения данных через систему. Потоки данных изображаются стрелками, указывающими направление передачи данных, с названием на стрелке, описывающим передаваемую информацию;

– хранилища данных: обозначают места, где данные сохраняются в системе, например, базы данных или файловые системы. Хранилища данных обычно изображаются в виде параллелограммов с названием внутри, например, «База данных пользователей», «Локальное хранилище».

В диаграмме потоков данных каждый процесс обрабатывает входные данные и создает выходные данные, которые затем передаются другим процессам или сущностям. Это позволяет визуализировать поток данных через систему и выявить взаимосвязи между различными элементами системы. Она может быть представлена на разных уровнях детализации, начиная от общего представления системы на высоком уровне до более подробного описания отдельных процессов и потоков данных. Это позволяет лучше понять структуру и функционирование информационной системы, а также выявить потенциальные узкие места и области для оптимизации.

3.3 Выбор архитектуры разрабатываемого программного продукта

Для разрабатываемого программного продукта был выбран модульный монолит в качестве наиболее подходящей архитектуры для решаемых задач.

3.4 Проектирование реляционной базы данных

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 16 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Концептуальная модель

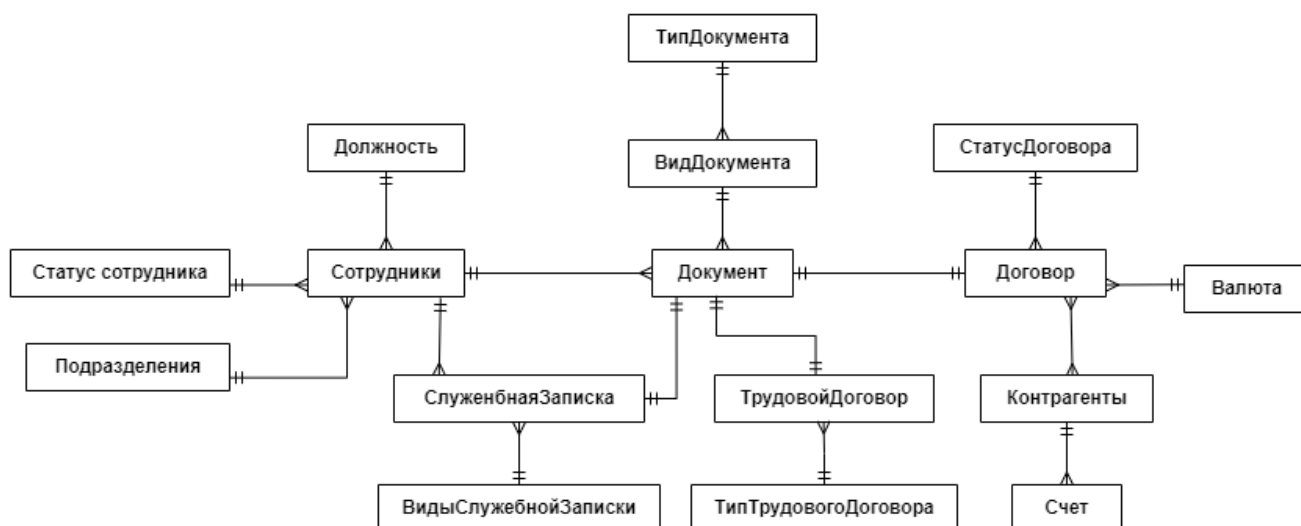


Рисунок 3.3 – Концептуальная модель

Описание

Следующим этапом необходимо сформировать логическую модель данных.

Логическая модель данных – это абстрактное представление структуры данных в базе данных без учета конкретных технических реализаций или ограничений конкретной СУБД.

Логической моделью данных называют усовершенствованную версию концептуальной модели. В ней схематически представлены ограничения данных, она описывает отношения между сущностями (таблицами) и их атрибутами, а также правила целостности данных, которые нужно будет реализовать независимо от платформы. Логическая модель данных определяют структуру, организацию и правила обработки данных, которые смогут обеспечить эффективность хранения, извлечения и обработки данных [3.4].

Объект – это сущность или явление, о котором хранятся данные. Например, клиент, продукция, заказ и т.д. Каждый объект имеет свой уникальный идентификатор, который позволяет однозначно идентифицировать запись в базе данных.

Атрибут – это характеристика объекта, которая описывает ее свойства или параметры. Например, для объекта «сотрудник» атрибутами могут быть имя, фамилия, отчество и т.д.

Объектом в данной базе данных будет информация о сотрудниках в таблице

В данной базе данных будут храниться сведения о сотрудниках и документов. Информация о сотрудниках будет храниться в таблицах 3.1-3.4, а информация о документах в таблице 3.5-3.18.

Таблица 3.1 – Атрибуты сотрудников

| Объект | Атрибуты |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Сотрудник | Идентификационный код Фамилия Имя Отчество Табельный номер Идентификационный код должность Идентификационный код статус сотрудника Идентификационный код подразделения Пол |

Таблица 3.2 – Атрибуты должности

| Объект | Атрибуты |
|-----------|-----------------------------------|
| Должность | Идентификационный код Название |

Таблица 3.3 – Атрибуты статуса сотрудника

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

Таблица 3.4 – Атрибуты подразделения

| Объект | Атрибуты |
|---------------|---------------------------------------|
| Подразделение | Идентификационный код Наименование |

Таблица 3.5 – Атрибуты документа

| Объект | Атрибуты |
|----------|-------------------------------------------------------------------------------------------|
| Документ | Идентификационный код Наименование Номер Дата Идентификационный код создателя |

Таблица 3.6 – Атрибуты типа документа

| Объект | Атрибуты |
|---------------|-----------------------------------|
| Тип документа | Идентификационный код Название |

Таблица 3.7 – Атрибуты вида документа

| Объект | Атрибуты |
|---------------|-------------------------------------------------------------------------------------------|
| Вид документа | Идентификационный код Название Идентификационный код код тип документа |

Таблица 3.8 – Атрибуты договора

| Объект | Атрибуты |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Договор | Идентификационный код Дата начала Дата окончания Идентификационный код валюты Стоимость Идентификационных код статуса Предмет договора |

Таблица 3.9 – Атрибуты статуса договора

| Объект | Атрибуты |
|-----------------|-----------------------------------|
| Статус договора | Идентификационный код Название |

Таблица 3.10 – Атрибуты валюты

| Объект | Атрибуты |
|--------|--------------------------------------------------------------------|
| Валюта | Идентификационный код Цифровой код Буквенный код Название |

Таблица 3.11 – Атрибуты контрагентов

| Объект | Атрибуты |
|-------------|------------------------------------------------------------------------------------------------------------------------------------|
| Контрагенты | Идентификационный код Наименование ИНН Телефон Электронная почта Юридический адрес Почтовый адрес Комментарий |

Таблица 3.12 – Атрибуты счета

| Объект | Атрибуты |
|--------|----------|
|--------|----------|

| | |
|------|----------------------------------------------------------------------------|
| Счет | Идентификационный код БИК Номер Идентификационный код контрагента |
|------|----------------------------------------------------------------------------|

Таблица 3.13 – Атрибуты контрагенты договора

| | |
|--------|---------------------------------------------------------------------------------------------------------------|
| Объект | Атрибуты |
| Счет | Идентификационный код договора Идентификационный код контрагента Идентификационный код стороны договора |

Таблица 3.14 – Атрибуты стороны договора

| | |
|------------------|-----------------------------------|
| Объект | Атрибуты |
| Стороны договора | Идентификационный код Название |

Таблица 3.15 – Атрибуты служебной записки

| | |
|-------------------|-----------------------------------------------------------------------------------------------------|
| Объект | Атрибуты |
| Служебная записка | Идентификационный код Идентификационный код вида Содержание Идентификационный код адресата |

Таблица 3.16 – Атрибуты виды служебной записки

| | |
|------------------------|-----------------------------------|
| Объект | Атрибуты |
| Виды служебной записки | Идентификационный код Название |

Таблица 3.17 – Атрибуты трудового договора

| | |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Объект | Атрибуты |
| Трудовой договор | Идентификационный код Фамилия Имя Отчество Должность Оклад Серия Номер Кем выдан Код подразделения Дата начала Дата окончания Идентификационный код типа трудоого договора |

Таблица 3.18 – Атрибуты типа трудового договора

| | |
|------------------------|-----------------------------------|
| Объект | Атрибуты |
| Тип трудового договора | Идентификационный код Название |

Инфологическая модель – это абстрактная модель данных, которая описывает структуру и отношения между данными в информационной системе без привязки к конкретной технологии или реализации базы данных. Она представляет собой логическое описание информации, которая используется в системе, включая сущности, их атрибуты и связи между ними. Основная цель инфологической модели – это описать информационные потребности бизнеса и определить структуру данных, не вдаваясь в технические детали хранения данных.

Инфологическая модель не зависит от конкретной реализации базы данных и часто используется на ранних стадиях проектирования для описания структуры данных и их отношений в системе. Это помогает разработчикам понять информационные потребности бизнеса и определить основные концепции, которые будут использоваться в дальнейшем проектировании базы данных.

Инфологическая модель базы данных представлена на рисунках 3.3-3.4.

| | | | | | | |
|------|------|----------|---------|------|---------------------|------|
| | | | | | ДП.09.02.07.24.??ПЗ | Лист |
| | | | | | | 21 |
| Изм. | Лист | № докум. | Подпись | Дата | | |



Рисунок 3.4 – Инфологическая модель базы данных модель

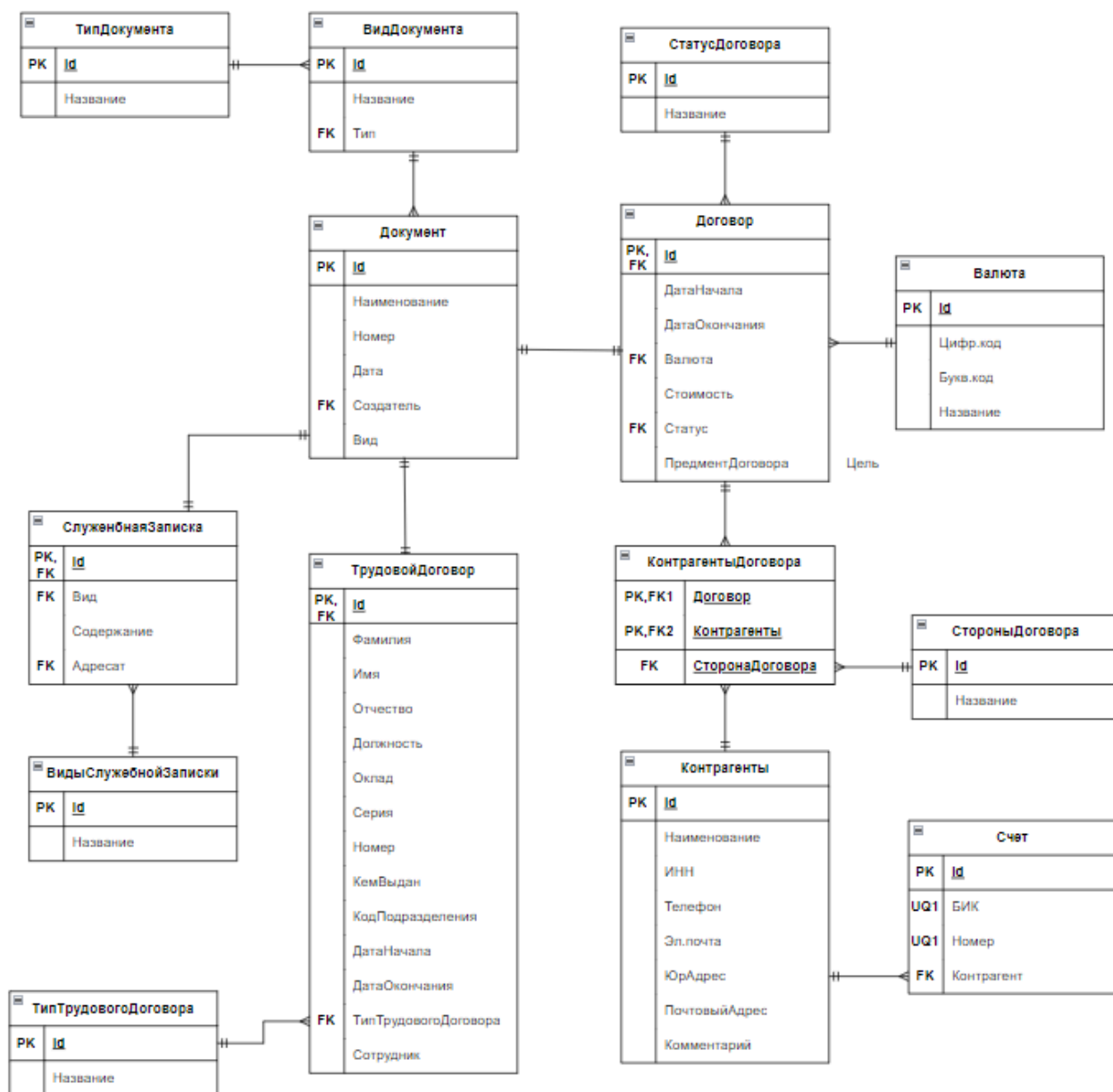


Рисунок 3.4 – Инфологическая модель базы данных

3.5 Прототипирование пользовательского интерфейса

Прототипирование пользовательского интерфейса – это процесс создания предварительной модели или макета продукта, или системы, который демонстрирует его основные характеристики, функции и взаимодействие с пользователем. Этот процесс позволяет представить концепцию продукта в виде простого прототипа до начала полноценной разработки. Прототипирование – это важный этап в процессе разработки продукта, который позволяет участникам

проекта лучше понять его концепцию и функциональность, а также получить обратную связь от пользователей до начала полноценной разработки.

Прототип может быть создан на разных этапах жизненного цикла проекта и может иметь разный уровень детализации. В зависимости от целей и потребностей проекта, прототип может быть простым эскизом на бумаге, дизайн-макетом в графическом редакторе, интерактивным цифровым прототипом или даже функциональным прототипом с базовым набором функций.

Прототипирование позволяет:

- проверить концепцию: путем создания прототипа можно быстро оценить жизнеспособность идеи или концепции продукта;

- собрать обратную связь: прототип может быть использован для сбора обратной связи от заинтересованных сторон, таких как заказчики и конечные пользователи, что помогает выявить недостатки и улучшить концепцию продукта;

- определить требования: прототипирование позволяет более точно определить требования к продукту, включая его функциональность, интерфейс и взаимодействие с пользователем;

- сэкономить время и ресурсы: обнаружение и исправление ошибок на ранних этапах разработки с помощью прототипа можно существенно сократить затраты времени и ресурсов на исправление ошибок на более поздних этапах проекта.

Таким образом, прототипирование играет важную роль в процессе разработки продукта, помогая команде разработки более точно понять и представить концепцию продукта и обеспечить его успешное развитие.

На рисунке 3.5 изображен прототип боковой панели.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 24 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

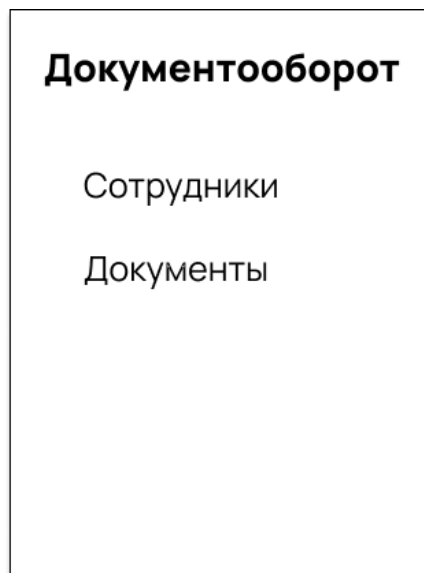


Рисунок 3.5 – Прототип боковой панели

Боковая панель – это элемент интерфейса программного приложения или веб-сайта, который обычно располагается сбоку экрана. Она содержит основное навигационное меню, которое позволяет пользователю перемещаться между различными разделами или функциями приложения. Боковая панель содержит следующие кнопки:

– Документооборот: эта кнопка переводит пользователей на главную страницу проекта, где они могут увидеть обзор всех документов и рабочих процессов.

– сотрудники: эта кнопка открывает пользователям список всех сотрудников компании;

– документы: эта кнопка открывает пользователям список всех документов.

На рисунке 3.6 изображен прототип сотрудников.

| | | | | | | |
|------|------|----------|---------|------|-----------------------|------|
| | | | | | ДП.09.02.07.24.?? .ПЗ | Лист |
| | | | | | | 25 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Документооборот

Сотрудники
Документы

Поиск...

Фильтр

Добавить сотрудника

| № | Фамилия | Имя | Отчество | Табельный номер | Должность | Статус | Подразделение |
|---|---------|-----|----------|-----------------|-----------|--------|---------------|
| 1 | | | | | | | |

✎ 🗑

Рисунок 3.6 – Прототип страницы сотрудников

Прототип страницы сотрудников – это веб-страница, которая представляет информацию и управление сотрудниками. Большая часть раздела включает в себя список сотрудников в виде таблицы. Каждая строка содержит сведения об одном сотруднике, такие как фамилия, имя, отчество, табельный номер, должность, статус и подразделение. В правой части таблицы в строке есть кнопки, которые позволяют пользователям редактировать и удалять сотрудников. Также выше таблицы есть поиск и фильтр, который позволяет пользователям искать сотрудников по должности. Правее находится кнопка, которая позволяет добавить нового сотрудника в таблицу.

На рисунке 3.7 изображен прототип формы добавления сотрудника.

Добавить сотрудника

Фамилия

Имя

Отчество

Пол

Должность

Табельный номер

Статус

Подразделение

Отмена

Добавить

Рисунок 3.7 – Прототип формы добавления сотрудника

На изображении представлен прототип формы добавления сотрудника – это форма, которая используется для добавления информации о новом сотруднике в систему компании. Она включает в себя поля для ввода фамилии, имени, отчества, пола, должности, табельного номера, подразделения и статуса. Ниже кнопка «Добавить» используется для добавления информации о новом сотруднике в систему. Кнопка «Отмена» используется для отмены операции добавления нового сотрудника.

На рисунке 3.8 изображен прототип страницы документов.

Документооборот

Сотрудники

Документы

Поиск...

Фильтр

Создать документ

| № | Наименование | Дата | Создатель |
|---|--------------|------|-----------|
| 1 | | | |

Рисунок 3.7 – Прототип страницы документов

Сотрудники

Документы

Создать документ

Трудовой договор

Служебная записка

Договор

Загрузить документ

Рисунок 3.8 – Прототип страницы создания документов

4 Разработка программного обеспечения

Разработка программного обеспечения – это процесс создания компьютерных программ и приложений с использованием программных языков, инструментов и методологий, которые решают определенные задачи или обеспечивают определенную функциональность. Она начинается с понимания потребностей пользователей или бизнеса и заканчивается внедрением и поддержкой готового программного продукта.

Разработка программного обеспечения документооборота представляет собой процесс создания программных решений, направленных на управление потоком документов в организации. Этот процесс начинается с анализа потребностей и требований организации к системе документооборота. Здесь определяются основные функциональные и нефункциональные требования, включая типы документов, процессы их создания, редактирования и хранения, а также требования к безопасности и защите информации.

После анализа приступаем к проектированию архитектуры программного продукта. В этом этапе определяется структура системы, её компоненты, интерфейсы, а также выбираются технологии и инструменты разработки. Архитектура должна быть гибкой и масштабируемой, чтобы адаптироваться к изменяющимся потребностям организации и объему документов.

После проектирования приступаем к созданию программного кода. Они используют выбранные языки программирования и инструменты для написания функциональных модулей, реализации **бизнес-логики** системы, создания пользовательского интерфейса и интеграции с другими системами организации.

После завершения разработки программное обеспечение проходит тестирование, включая модульное, интеграционное и системное тестирование.

После успешного завершения всех этапов разработки программное обеспечение внедряется в организацию. Затем осуществляется обучение пользователей, настройка и адаптация системы под специфику работы организации.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 29 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Важной частью процесса разработки является последующая поддержка и сопровождение программного обеспечения. Это включает в себя регулярное обновление, исправление ошибок, добавление новых функций и поддержку пользователей.

4.1 Обоснование выбора инструментальных средств разработки программного продукта

Для разработки данного приложения был выбран язык TypeScript – это язык программирования, который является строго типизированным надмножеством JavaScript. Он добавляет статическую типизацию, классы, интерфейсы и другие возможности, которых нет в чистом JavaScript, что делает его более мощным и надежным инструментом для разработки крупных и сложных приложений. Вот несколько ключевых особенностей TypeScript:

- статическая типизация: TypeScript позволяет определять типы данных переменных, параметров функций, возвращаемых значений и других элементов кода. Это помогает выявлять ошибки на этапе компиляции, предотвращает типовые ошибки и улучшает читаемость кода;
- надмножество JavaScript: TypeScript является надмножеством JavaScript, что означает, что весь действующий код JavaScript является также допустимым кодом TypeScript. Это позволяет постепенно внедрять TypeScript в существующие проекты и использовать современные возможности языка JavaScript;
- объектно-ориентированный подход: TypeScript поддерживает объектно-ориентированный стиль программирования, включая классы, интерфейсы, наследование, абстрактные классы, полиморфизм и другие концепции, которые делают код более структурированным и понятным;
- широкая поддержка и экосистема: TypeScript имеет широкую поддержку в сообществе разработчиков и интеграцию с популярными инструментами и фреймворками, такими как React, Angular, Vue.js и Node.js. Есть также множество

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 30 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

дополнительных инструментов и библиотек, разработанных специально для TypeScript.

TypeScript предоставляет множество преимуществ для разработчиков, помогая создавать более надежные, чистые и масштабируемые приложения на JavaScript. Он позволяет эффективно работать над крупными проектами и снижает количество ошибок в коде благодаря статической типизации и другим возможностям языка.

React – это JavaScript библиотека для создания пользовательских интерфейсов (UI), разработанная компанией Facebook. Она позволяет разрабатывать мощные, динамичные и масштабируемые веб-приложения с использованием компонентного подхода. Вот несколько ключевых особенностей React:

- компонентный подход: React основан на концепции компонентов, которые являются независимыми, переиспользуемыми и модульными блоками интерфейса. Компоненты могут быть составлены из других компонентов, что упрощает создание и поддержку сложных пользовательских интерфейсов;

- виртуальный DOM: React использует виртуальный DOM для эффективного обновления пользовательского интерфейса. Вместо обновления DOM напрямую, React создает виртуальное представление DOM в памяти, сравнивает его с реальным DOM и обновляет только те части, которые изменились. Это позволяет значительно повысить производительность и отзывчивость приложения;

- JSX(TSX): это расширение языка JavaScript, которое позволяет описывать структуру пользовательского интерфейса в виде XML-подобного синтаксиса. JSX упрощает создание и чтение компонентов React, делает код более выразительным и понятным;

- однонаправленный поток данных: в React данные передаются через иерархию компонентов в одном направлении - от родительских компонентов к дочерним. Это делает управление состоянием приложения более прозрачным и предсказуемым;

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 31 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– богатая экосистема: React имеет обширную экосистему инструментов, библиотек и фреймворков, которые облегчают разработку, тестирование и развертывание приложений на React. Например, Redux для управления состоянием, React Router для маршрутизации, и многие другие;

– кроссплатформенность: React может использоваться для разработки не только веб-приложений, но и мобильных приложений с использованием React Native, а также десктопных приложений с использованием Electron. Это позволяет создавать единое пользовательское интерфейсное решение для различных платформ.

В качестве среды для разработки приложения была выбрана программа Visual Studio Code – это кроссплатформенный редактор кода от компании Microsoft, предоставляющая широкий набор инструментов и функций для комфортной работы разработчика. Вот некоторые из основных инструментов, доступных в Visual Studio Code:

– расширения: Visual Studio Code поддерживает огромное количество расширений, которые расширяют его функциональность и добавляют новые возможности. Расширения могут предоставлять поддержку различных языков программирования, интеграцию с фреймворками и библиотеками, а также дополнительные инструменты для работы с кодом;

– интегрированный терминал: Visual Studio Code включает в себя интегрированный терминал, который позволяет выполнять команды напрямую из среды разработки, без необходимости переключения на внешний терминал или командную строку;

– отладчик: встроенный отладчик Visual Studio Code позволяет выполнять отладку кода напрямую из редактора. Он поддерживает различные языки программирования и фреймворки, а также обеспечивает мощные возможности отслеживания значений переменных, установки точек останова и анализа стека вызовов;

– интеграция с Git: Visual Studio Code интегрируется с системой контроля версий Git, обеспечивая возможность выполнения всех основных операций Git

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 32 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

непосредственно из редактора. Это включает в себя коммиты, ветвление, слияния, решение конфликтов и многое другое;

- подсказки: редактор кода в Visual Studio Code предоставляет поддержку автодополнения и подсказок, что существенно упрощает написание кода и уменьшает количество ошибок. Он также поддерживает различные языки программирования и фреймворки, что обеспечивает точное и контекстно-ориентированное автодополнение;

- интеграция с различными языками и фреймворками: предоставляет поддержку для множества языков программирования и фреймворков, включая JavaScript, TypeScript, Python, Java, C#, PHP, и многие другие. Это позволяет разработчикам работать с разнообразными технологиями, не выходя из среды VS Code;

- конфигурация и настройка: предоставляет широкие возможности для настройки и персонализации среды разработки. Разработчики могут настраивать темы оформления, раскладки редактора, горячие клавиши, сниппеты кода и многое другое, чтобы адаптировать среду разработки под свои нужды;

- интеграция с системами сборки и автоматизации: Visual Studio Code интегрируется с различными инструментами для автоматизации сборки и развертывания приложений, такими как npm, yarn и другие. Это позволяет автоматизировать рутинные задачи разработки, такие как компиляция, тестирование и сборка проекта;

- расширяемость и сообщество: Visual Studio Code является очень расширяемой средой разработки благодаря своему открытому и расширяемому API. Существует огромное сообщество разработчиков, которые создают и поддерживают разнообразные расширения, темы оформления и инструменты, которые улучшают работу среды Visual Studio Code;

- интеграция с облачными сервисами: Visual Studio Code интегрируется с различными облачными сервисами, такими как Azure, AWS, Google Cloud и другие, что обеспечивает удобное взаимодействие с облачными ресурсами прямо из редактора кода;

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 33 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– кроссплатформенность: редактор кода поддерживает операционные системы Windows, macOS и Linux, что делает его универсальным инструментом для разработки на различных платформах.

Эти функции и возможности делают Visual Studio Code одним из наиболее мощных и гибких инструментов для разработки программного обеспечения, который позволяет разработчикам ускорить процесс разработки, повысить качество кода и улучшить опыт разработки в целом.

В качестве базы данных была выбрана MongoDB – это реляционная база данных, которая широко используется для хранения и управления большими объемами данных. Она относится к категории NoSQL баз данных, что означает, что она не использует табличный способ организации данных, характерный для традиционных реляционных баз данных, таких как SQL. Вместо этого MongoDB использует гибкую модель документов, которая позволяет легко и быстро обрабатывать данные различных типов и структур.

Основные особенности MongoDB:

– документная модель данных: в MongoDB данные хранятся в виде документов JSON-подобного формата. Каждый документ представляет собой объект, содержащий пары «ключ-значение», что позволяет хранить данные в удобочитаемом формате и легко адаптировать их под различные структуры данных;

– гибкость и масштабируемость: MongoDB позволяет добавлять новые поля в документы без необходимости изменения структуры базы данных. Это делает MongoDB особенно подходящей для проектов с часто изменяющимися требованиями. MongoDB поддерживает горизонтальное масштабирование, что позволяет распределять данные по множеству серверов и обеспечивать высокую доступность и производительность;

– высокая производительность: MongoDB обеспечивает высокую скорость чтения и записи данных благодаря своей архитектуре и использованию встроенного механизма индексации. Это позволяет эффективно работать с большими объемами данных в режиме реального времени;

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 34 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

– репликация и отказоустойчивость: MongoDB поддерживает репликацию данных через Replica Sets, что позволяет автоматически дублировать данные на несколько серверов. Это обеспечивает высокую доступность данных и защиту от потерь данных в случае сбоя одного из серверов;

– запросы и агрегирование данных: MongoDB предлагает мощный язык запросов, который позволяет выполнять сложные операции выборки и обновления данных. Также имеется механизм агрегации, который позволяет выполнять операции обработки и анализа данных, такие как фильтрация, сортировка и группировка;

– интеграция с различными языками программирования: MongoDB имеет официальные драйверы и библиотеки для множества языков программирования, включая JavaScript, Python, Java, C#, Node.js и многие другие.

4.2 Описание алгоритма программы

Шаг 1. Запуск программы.

Шаг 2. Открытие окна авторизации.

Шаг 3. Ввод логина пароля.

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

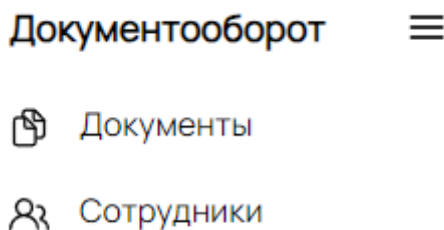
Шаг 5. Вопрос: Роль пользователя диспетчер? Если нет, то переходим к шагу 6. Если да, то открываем окно диспетчера и переходим к шагу 7.

4.3 Описание интерфейса программного продукта

Интерфейс программного продукта – это совокупность элементов, средств взаимодействия и визуальных компонентов, которые предоставляют пользователю доступ к функциональности программы. Это включает в себя различные элементы пользовательского интерфейса, такие как меню, кнопки, поля ввода, окна, диалоговые окна, панели инструментов, элементы управления и т. д. Цель

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 35 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

интерфейса программного продукта – обеспечить удобство использования и эффективное взаимодействие пользователя с программой.



На рисунке 4.2 изображен интерфейс страницы сотрудников.

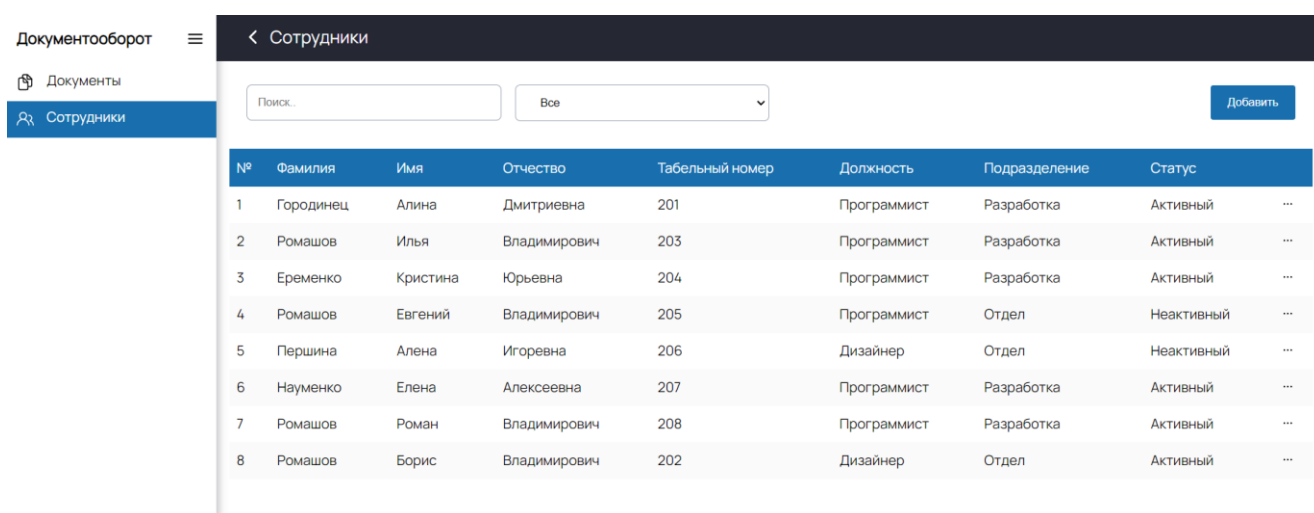


Рисунок 4.2 – Интерфейс страницы сотрудников

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

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

На рисунке 4.3 изображен интерфейс модального окна добавления сотрудника.

Рисунок 4.3 – Интерфейс формы добавления сотрудника

На изображении показан интерфейс добавления сотрудника. Форма имеет два раздела:

1) Личная информация:

- фамилия: это поле является обязательным и должно быть введено в виде фамилии сотрудника;
- имя: это поле является обязательным и должно быть введено в виде имени сотрудника;
- отчество: это поле является необязательным и может быть введено в виде отчества сотрудника;
- пол: это поле является обязательным и должно быть выбрано из раскрывающегося меню. Возможные варианты: «Мужской» и «Женский».

2) Рабочая информация:

- должность: это поле является обязательным и должно быть выбрано из раскрывающегося меню. Параметры заполняются в зависимости от названий должностей в организации;
- подразделение: это поле является обязательным и должно быть выбрано из раскрывающегося меню. Параметры заполняются в зависимости от подразделений организации;

– статус: это поле является обязательным и должно быть выбрано из раскрывающегося меню. Возможные варианты: «Активен» и «Неактивен».

Также форма имеет две кнопки «Отмена» она отменяет форму и возвращает пользователя на предыдущую страницу. «Добавить» сохраняет форму и добавляет нового сотрудника в проект.

На рисунке 4.4 изображен интерфейс модального окна редактирования сотрудника.

The screenshot displays a web interface for managing employees. A modal window titled "Редактирование сотрудника" is open, allowing for the editing of an employee's details. The modal is divided into two sections: "Личная информация" (Personal information) and "Рабочая информация" (Work information). The "Личная информация" section includes fields for "Фамилия" (Surname), "Имя" (Name), and "Отчество" (Patronymic), each with a text input field. Below these is a "Пол" (Gender) section with two radio buttons: "Мужской" (Male) and "Женский" (Female), with "Женский" currently selected. The "Рабочая информация" section includes a "Табельный номер" (Employee ID) field, a "Должность" (Position) dropdown menu, a "Подразделение" (Department) dropdown menu, and a "Статус" (Status) section with two radio buttons: "Активный" (Active) and "Неактивный" (Inactive), with "Активный" currently selected. At the bottom of the modal are two buttons: "Отменить" (Cancel) and "Изменить" (Save). The background shows a list of employees with columns for "№" (ID), "Фамилия" (Surname), "Имя" (Name), "Подразделение" (Department), and "Статус" (Status). The list contains 8 entries, with the first entry being "1 Еременко Кристина".

Рисунок 4.4 – Интерфейс формы редактирования сотрудника

– На изображении представлен интерфейс модального окна редактирования сотрудника – это форма, которая используется для редактирования информации дает возможность внесения изменений в данные о существующем работнике в системе. Ниже кнопка «Изменить» используется для добавления измененной информации в систему. Кнопка «Отмена» используется для отмены операции редактирования сотрудника.

На рисунке 4.5 изображен интерфейс страницы документов.

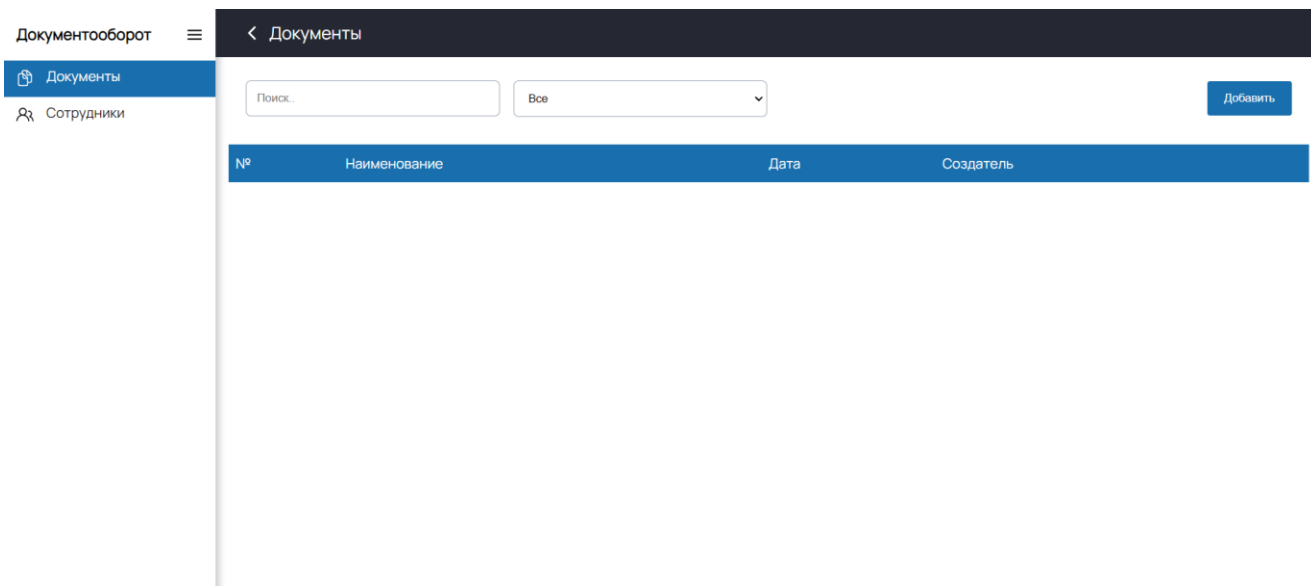


Рисунок 4.5 – Интерфейс страницы документов

На рисунке 4.6 изображен интерфейс страницы добавления трудового договора.

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

- поиск сотрудников по наименованию;
- фильтрация документов по наименованию;
- просмотр полного списка документов;
- добавление новых документов в список;
- редактирование информации о существующих документах;
- удаление документов.

Рисунок 4.6 – Интерфейс страницы добавления трудового договора

Рисунок 3.7 – Интерфейс страницы добавления трудового договора

На изображении показана страница, которая позволяет пользователям вводить информацию о трудоустройстве для создания трудового договора. Форма разделена на два раздела личная информация и персональные данные. Включает в себя поля для ввода фамилии, имени, отчества, пола, должности, подразделения, серия и номер паспорта, кем выдан, дата выдачи, код подразделения, оклад.

На рисунке 4.7 изображен интерфейс страницы добавления служебной записки.

Документооборот < Документы / Служебная записка

Документы

Сотрудники

Трудовой договор Служебная записка Договор Загрузить файл

ФИО

Выберите себя

Кому

Кому адресована

Вид служебной записки

Выберите вид

Содержание

Содержание, что Вас беспокоит

Отменить Добавить

Рисунок 4.7 – Интерфейс страницы добавления служебной записки

На изображении представлен интерфейс страницы добавления служебной записки – это страница, которая используется для добавления информации о новом документе в систему компании. Она включает в себя поля для выбора фамилия, имя, отчество кто создает документ и кому адресован, вид служебной записки и поле для ввода содержания документа. Ниже кнопка «Добавить» используется для добавления информации о новом документе в систему. Кнопка «Отмена» используется для отмены операции добавления нового документа.

На рисунке 4.8 изображен интерфейс добавления договора.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 40 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Рисунок 4.8 – Интерфейс добавления договора

На изображении представлен интерфейс страницы добавления договора. Она включает в себя поля для ввода информации о договоре предмет договора, дата начала, дата окончания, валюта, стоимость, статус, контрагенты – пользователь может добавить одного или нескольких контрагентов по договору, наименование, ИНН, телефон, электронная почта, юридический адрес, почтовый адрес.

На рисунке 4.9 изображен интерфейс добавления файлов из компьютера.

Рисунок 4.9 – Интерфейс добавление файлов

На изображении показана страница веб-сайта, предназначенная для загрузки файлов. Эта страница позволяет пользователю добавить еще один файл для загрузки. Пользователь может легко выбрать файл, который он хочет загрузить, и загрузить его на сервер. Файлы могут загружаться только в форматах

PDF, DOC и XLS. Максимальный размер файла - 5 МБ. Загруженные файлы будут сохранены в папке "Документы" на сервере.

4.4 Инструкция по эксплуатации программного продукта

Инструкция по эксплуатации программного продукта является важным документом, который предоставляет пользователям информацию о том, как использовать программное обеспечение для достижения своих целей.

На рисунке 4.9 изображена фильтрация данных.



Рисунок 4.9 – 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 рисунке 4.10 изображен поиск данных.



Рисунок 4.10 – Поиск 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 4.11 изображена кнопка добавления данных.

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 42 |
| Изм. | Лист | № докум. | Подпись | Дата | | |



Рисунок 4.11 – Кнопка добавления данных

Кнопка «Добавить» представляет собой элемент интерфейса, предназначенный для внесения новых данных или элементов в систему. При нажатии на эту кнопку пользователь открывает форму или модальное окно, где может ввести необходимую информацию о новом элементе, такую как название, описание, дата и другие параметры. Кнопка обеспечивает удобный и быстрый способ добавления новых данных в проект.

На рисунке 4.12 изображено модальное окно с кнопкой редактирования данных.

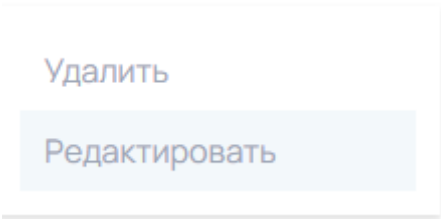


Рисунок 4.12 – Кнопка редактирования данных

Кнопка редактирования представляет собой элемент интерфейса, который позволяет пользователям вносить изменения в уже существующие данные или элементы. Пользователь выбирает элемент, который хочет изменить, а затем нажимает на кнопку редактирования. После этого открывается форма или модальное окно, где можно внести нужные изменения, например, изменить текст, исправить ошибки или обновить информацию. Кнопка редактирования обеспечивает удобный способ обновления данных в проекте.

На рисунке 4.13 изображено модальное окно с кнопкой удаления данных.

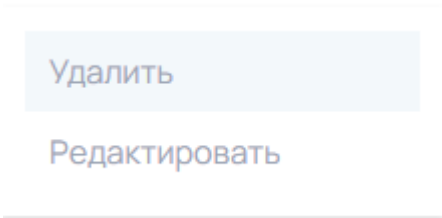


Рисунок 4.13 – Кнопка удаления данных

В проекте, кнопка удаления представляет собой элемент интерфейса, предназначенный для удаления данных или элементов из системы. Пользователь

выбирает элемент, который хочет удалить, и нажимает на кнопку удаления. После подтверждения удаления элемента система проводит соответствующие операции и удаляет выбранный элемент из базы данных или списка. Кнопка удаления предоставляет простой способ удаления ненужных данных из проекта.

На рисунке 4.14 изображено подтверждающее окно.

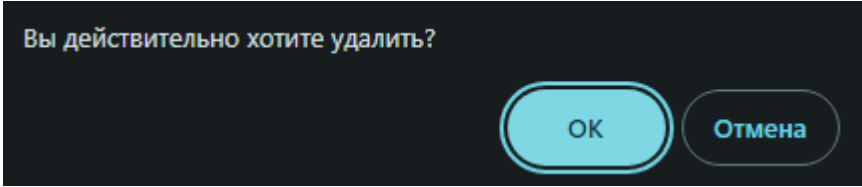


Рисунок 4.14 – Подтверждающее окно

Подтверждающее окно – это всплывающее диалоговое окно, которое появляется на экране после того, как пользователь совершил определенное действие, например, нажал кнопку удаления или другое действие, которое может иметь серьезные последствия. Оно запрашивает у пользователя подтверждение намерений перед выполнением этого действия.

5 Экономическая часть

| | | | | | | |
|------|------|----------|---------|------|-----------------------|------|
| | | | | | ДП.09.02.07.24.?? .ПЗ | Лист |
| | | | | | | 45 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Охрана труда и техника безопасности

Охрана труда – это способ организации мер, включающих в себя правовые, лечебно-профилактические, реабилитационные, экономические, санитарно-гигиенические, организационно-технические и другие мероприятия направленные на сохранения жизни и здоровья людей в ходе трудовой деятельности, улучшение трудовой дисциплины, развитие их работоспособности, сохранение денежных средств и оборудования предприятия, увеличение прибыли компании. Дело в том, что большинство проблем и неприятностей, в том числе, несчастных случаев на производстве, происходит чаще всего из-за незнания существующих законодательных требований охраны труда [6].

Техника безопасности – система организационных мероприятий и технических средств, направленных на предотвращение воздействия на работающих опасных производственных факторов, являющихся причиной травм или внезапного резкого ухудшения здоровья. Техника безопасности является частью охраны труда и включает такие мероприятия, как обучение и инструктаж работающих по вопросам безопасности труда, поддержание в технически безопасном состоянии зданий и сооружений, оснащение вновь создаваемого и эксплуатируемого производственного оборудования защитными и предохранительными устройствами, разработку средств коллективной и индивидуальной защиты работающих от воздействия опасных и вредных производственных факторов, а также организацию обеспечения этими средствами рабочих и служащих [6].

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

– оценка рисков: первый шаг в обеспечении безопасности на рабочем месте – это оценка всех потенциальных опасностей и рисков, связанных с выполняемыми задачами. Это включает в себя идентификацию опасных материалов,

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 46 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

оборудования и процессов, а также оценку вероятности и последствий возможных несчастных случаев;

– разработка политики безопасности: компании должны разработать и внедрить политику безопасности, которая описывает их обязательства по обеспечению безопасности на рабочем месте. Это может включать в себя требования к использованию средств защиты, процедуры обучения и обучения персонала, а также процедуры для обращения с опасными материалами;

– обучение: работники должны быть обучены правильным методам и процедурам работы, которые гарантируют их безопасность и здоровье. Это может включать в себя обучение по правилам пожарной безопасности, использованию личной защиты, безопасной эксплуатации оборудования и т.д.

– использование средств защиты: работники должны использовать необходимые средства защиты, такие как шлемы, защитные очки, наушники, маски и другие, в соответствии с требованиями безопасности на рабочем месте;

– обслуживание оборудования: оборудование должно регулярно проверяться, обслуживаться и ремонтироваться для обеспечения его безопасной работы. Работники должны быть обучены распознавать потенциальные проблемы с оборудованием и сообщать о них своему руководству;

– соблюдение законодательства: компании должны соблюдать все соответствующие законы и нормативные акты, касающиеся охраны труда и техники безопасности, в том числе стандарты безопасности на рабочем месте, требования к маркировке и обозначению опасных материалов и оборудования.

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

Мероприятия по охране труда и ТБ направлены на предотвращение травм работников и исключение ситуаций, следствием которых может стать несчастный случай или авария. При этом на различных предприятиях требования техники безопасности и комплекс необходимых мер могут существенно различаться в

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 47 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

связи с отраслевыми особенностями. В целом же можно выделить общие требования. Приведем основные из этих требований.

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

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

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

- использование оборудования и конструкций, соответствующих требованиям стандартов и другой нормативной документации;
- соблюдение сроков периодических ремонтов и обслуживания оборудования;
- соблюдение требований пожарной и электробезопасности при оснащении производственных и офисных помещений;
- установка необходимых защитных приспособлений и конструкций;
- обеспечение достаточной освещенности, вентиляции, поддержание оптимального температурного режима на рабочих местах;
- своевременное устранение пыли и отходов производства;
- обеспечение работников специальной одеждой и обувью, а также другими средствами индивидуальной защиты в соответствии со спецификой производства;
- обеспечение работников актуальными инструкциями по ТБ, наглядными материалами;
- создание на рабочих местах и в производственных помещениях всех необходимых систем сигнализации, размещение знаков безопасности и т.д. [6.1].

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.22.ПЗ | Лист |
| | | | | | | 48 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Заключение

| | | | | | | |
|------|------|----------|---------|------|-----------------------|------|
| | | | | | ДП.09.02.07.24.?? .ПЗ | Лист |
| | | | | | | 49 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Список использованных источников

| | | | | | | |
|------|------|----------|---------|------|-----------------------|------|
| | | | | | ДП.09.02.07.24.?? .ПЗ | Лист |
| | | | | | | 50 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Приложение
(обязательное)
Листинг программы

| | | | | | | |
|------|------|----------|---------|------|----------------------|------|
| | | | | | ДП.09.02.07.24.??.ПЗ | Лист |
| | | | | | | 51 |
| Изм. | Лист | № докум. | Подпись | Дата | | |