

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)

А.А. Аникина

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 №148 от «6» февраля 2024 г.

2. Срок сдачи законченной работы: 10 июня 2024 г.

3. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов):

- технико-экономическая характеристика объекта;
- сбор, анализ и формирования требований к информационной системе;
- проектирование информационной системы;
- разработка программного обеспечения.

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**

Студентки 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

_____ / Р.В. Ромашов /
(подпись)

«1» марта 2024 г.

Студент

_____ / А.Д. Городинец /
(подпись)

«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 Технико-экономическая характеристика объекта	6
1.1 Общие сведения о предприятии	6
1.2 Организационная структура предприятия.....	7
2 Сбор, анализ и формирования требований к информационной системе	9
2.1 Цели и назначение создания автоматизированной системы	10
2.2 Характеристика объектов автоматизации	11
2.3 Требования к автоматизированной системе.....	11
3 Проектирование информационной системы	12
3.1 Разработка и описание сценариев использования информационной системы	12
3.2 Поток данных в информационной системе	14
3.3 Выбор архитектуры разрабатываемой информационной системы	16
3.4 Проектирование базы данных.....	17
3.5 Формирование основных бизнес-процессов информационной системы.....	21
3.6 Прототипирование пользовательского интерфейса	22
4 Разработка программного продукта.....	27
4.1 Средства разработки программного продукта.....	27
4.2 Реализация физической модели данных программного продукта.....	32
4.3 Описание интерфейса программного продукта	34
4.4 Инструкция по эксплуатации программного продукта	40
5 Экономическая часть	44
6 Охрана труда и техника безопасности	48
Заключение	51
Список использованных источников	52
Приложение	55

					ДП.09.02.07.24.05.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.

					ДП.09.02.07.24.05.ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

1 Технико-экономическая характеристика объекта

1.1 Общие сведения о предприятии

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

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

Виды программного обеспечения, которые разрабатываются на предприятии:

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

Так же предприятия оказывает услуги:

- доработка и модернизация существующего программного обеспечения;
- техническая поддержка 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;

					ДП.09.02.07.24.05.ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

- техническая поддержка ПО, исправление ошибок, доработка ПО.

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

1.2 Организационная структура предприятия

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

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

Структура команды и функции каждого ее члена:

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

					ДП.09.02.07.24.05.ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Структура предприятия представлена на рисунке 1.1.

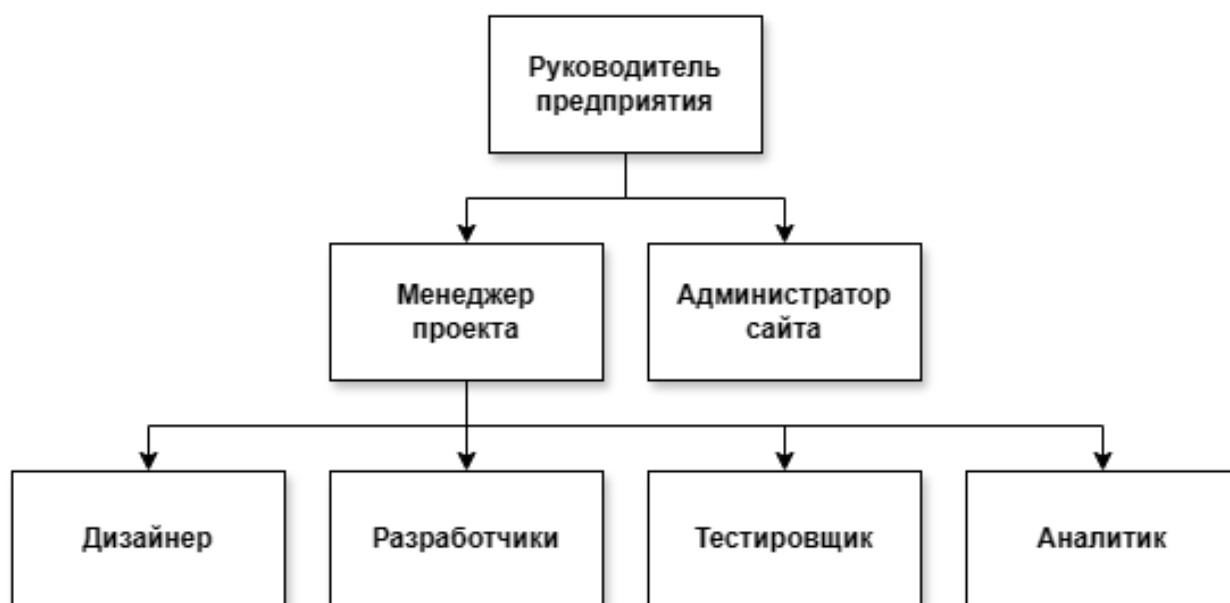


Рисунок 1.1 – Организационная структура предприятия

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

2 Сбор, анализ и формирования требований к информационной системе

Сбор, анализ и формирование требований к информационной системе для документооборота – это важные этапы разработки, которые обеспечивают создание потребностей пользователей программного продукта. Рассмотрим эти этапы более подробно:

- сбор требований включает в себя определение всех участников, которые будут использовать систему или влиять на ее разработку и внедрение. Проведение интервью, опросов и встреч с заинтересованными сторонами для выяснения их потребностей и ожиданий от системы документооборота. Изучение аналогичных систем, чтобы понять лучшие практики и возможные улучшения;

- анализ требований представляет определение приоритетов требований на основе их важности и влияния на бизнес-процессы. Разработка требований к пользовательскому интерфейсу системы для обеспечения удобства и эффективности работы пользователей. Это включает в себя требования к навигации, структуре экранов и взаимодействию с пользователем;

- в формировании требований проходит документирование всех собранных и проанализированных требований, например, в виде технического задания. Создание моделей бизнес-процессов, которые будут поддерживаться системой документооборота. Создание прототипов пользовательского интерфейса для получения ранней обратной связи от пользователей и проверки соответствия требований. Проверка требований на полноту, утверждение требований заинтересованными сторонами, чтобы гарантировать, что они полностью согласны с определенными.

Сбор, анализ и формирование требований являются ключевыми этапами разработки информационной системы для документооборота. Они помогают обеспечить, что конечный продукт будет соответствовать ожиданиям пользователей и требованиям бизнеса, улучшит рабочие процессы и повысит общую эффективность организации.

					ДП.09.02.07.24.05.ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

2.1 Цели и назначение создания автоматизированной системы

Основные цели и назначение документооборота заключаются в улучшении эффективности и надежности обработки документов, а также в снижении затрат на их управление.

Основные цели создания автоматизированной системы:

- автоматизация рутинных задач, таких как регистрация, хранение, поиск и обработка документов, позволяет значительно ускорить рабочие процессы;
- сокращение времени, затрачиваемого на обработку и поиск документов, повышает производительность сотрудников;
- обеспечение защиты конфиденциальной информации с помощью механизмов аутентификации, авторизации и шифрования данных;
- переход на электронные документы сокращает расходы на бумагу, печать и хранение.

Назначение автоматизированной системы:

- создание единого электронного архива для всех документов организации;
- обеспечение соответствия требованиям законодательства и внутренним нормативным актам в области документооборота;
- быстрый поиск и доступ к нужным документам из любой точки и в любое время.

Цели и назначение создания автоматизированной системы документооборота направлены на улучшение управления документами в организации. Внедрение способствует повышению эффективности и надежности процессов обработки документов, а также снижению затрат. Такая система играет ключевую роль в модернизации бизнес-процессов и способствует более эффективному использованию ресурсов, обеспечивая конкурентоспособность и устойчивое развитие организации.

					ДП.09.02.07.24.05.ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

2.2 Характеристика объектов автоматизации

Автоматизация документооборота в корпоративной среде включает в себя различные объекты, каждый из которых играет ключевую роль в упрощении и оптимизации процессов работы с документами. Основными объектами автоматизации являются:

- возможность быстрого нахождения документов по различным параметрам;
- администраторы и пользователи, которые имеют разные уровни доступа и возможности работы с документами;
- включение автоматических полей, например, дата создания, номер документа, название документа.

В результате, автоматизация документооборота способствует повышению общей эффективности работы организации, улучшению управления документами и обеспечению высокого уровня безопасности данных.

2.3 Требования к автоматизированной системе

Для автоматизации документооборота требуется выполнение ряда задач:

- создание и удаление документов различных типов;
- поиск по метаданным дата создания, автор, тип документа и т.д. Фильтрацию и сортировка результатов поиска по различным критериям;
- разграничение доступа к документам и функциям системы в зависимости от ролей пользователей;
- шифрование данных при авторизации;
- наличие справочной системы.

Эти требования обеспечивают комплексный подход к созданию автоматизированной системы документооборота, способной эффективно и безопасно обрабатывать документы, обеспечивая при этом высокую производительность и удобство использования для всех категорий пользователей.

					ДП.09.02.07.24.05.ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

3 Проектирование информационной системы

Проектирование информационной системы – это процесс создания плана или модели, которая определяет структуру, компоненты, интерфейсы и другие характеристики системы с целью удовлетворения определенных потребностей или решения определенных задач. Оно включает в себя анализ требований пользователей и бизнеса, разработку архитектуры системы, выбор технологий и инструментов, а также создание подробных спецификаций для разработки и внедрения системы. Проектирование информационной системы является ключевым этапом в жизненном цикле разработки программного обеспечения и влияет на ее функциональность, надежность, безопасность и эффективность [6].

В контексте документооборота проектирование информационной системы включает в себя определение основных компонентов системы, таких как система управления документами, интерфейсы пользователя, механизмы авторизации и аутентификации, а также интеграцию с другими системами и сервисами. Основная цель проектирования информационной системы документооборота – обеспечить управление документами, улучшить процессы работы с ними, снизить риски и повысить прозрачность и контроль над документами в организации.

3.1 Разработка и описание сценариев использования информационной системы

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

					ДП.09.02.07.24.05.ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

системы или внешние устройства. На диаграмме актеры обычно изображаются в виде простых фигур, обозначающих их роль в системе;

– диаграмма описывает конкретные действия или задачи, которые выполняются пользователем или системой для достижения определенной цели. Каждая диаграмма представляет собой отдельный функциональный элемент системы. Они представляются в виде эллипсов с текстом, описывающим функциональность;

– 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 [13].

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.

На рисунке 3.1 представлена диаграмма, описывающая основные функциональные возможности приложения и роли пользователей:

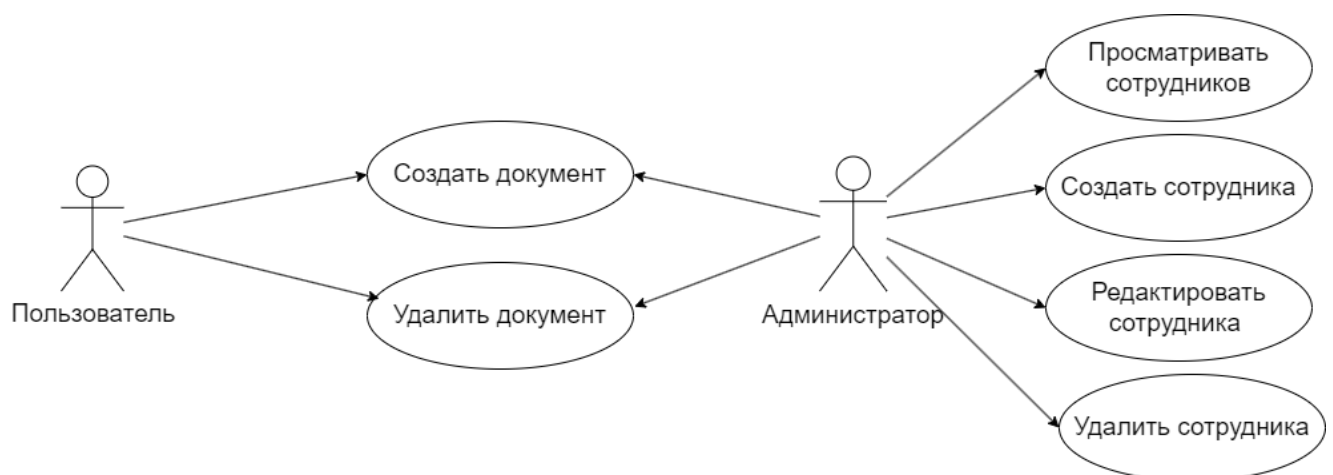


Рисунок 3.1 – Диаграмма вариантов использования

Актеры:

– пользователь: сотрудник, который работает с документами в системе;

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

Взаимодействия:

– пользователь может создать новый документ, удалять существующий документ, просматривать существующие документы.

– администратор может настраивать параметры системы, войдя в режим администрирования, создавать сотрудников, редактировать сотрудников, внося изменения, 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»;

– 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;

					ДП.09.02.07.24.05.ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

– 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» [11].

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.

На рисунке 3.2 изображена диаграмма потоков данных.

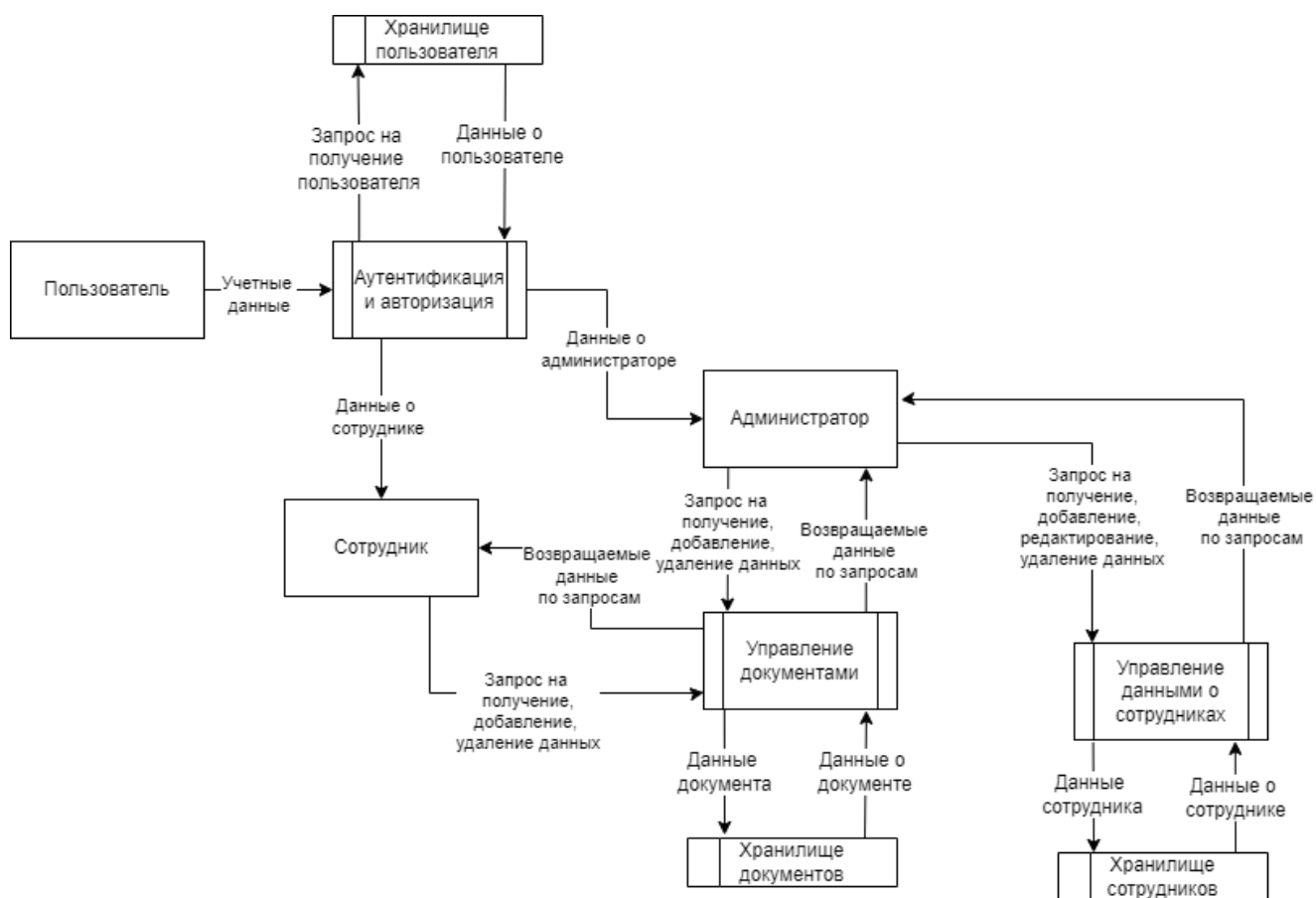


Рисунок 3.2 – Диаграмма потоков данных

Изм.	Лист	№ докум.	Подпись	Дата

3.3 Выбор архитектуры разрабатываемой информационной системы

Архитектура программного продукта – это разработанная структура программы, которая включает в себя определение взаимодействия компонентов интерфейса с внутренними процессами программы [12].

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

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

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

Серверная часть отвечает за обработку запросов от клиентской части, выполнение бизнес-логики приложения, доступ к базе данных и обеспечение безопасности данных.

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

					ДП.09.02.07.24.05.ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

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

3.4 Проектирование базы данных

Концептуальная модель – это абстрактное представление о данных в информационной системе, которое определяет основные сущности, их атрибуты и взаимосвязи между ними. Она служит основой для проектирования базы данных, представляя собой высокоуровневую картину о данных, которые будут использоваться в системе [10].

На рисунке 3.3 изображена концептуальная модель.



Рисунок 3.3 – Концептуальная модель

Концептуальная модель показывает основные компоненты системы и их взаимосвязи. Она служит основой для дальнейшего проектирования и реализации базы данных.

Представляет собой высокоуровневое описание сущностей и их взаимосвязей, необходимых для эффективного управления документами в организации. Модель помогает в дальнейшем разработке и реализации системы, обеспечивая базу для более детализированных технических решений.

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 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 [7].

В данной базе данных будут храниться сведения о сотрудниках и документах. Информация о сотрудниках будет храниться в таблицах 3.1-3.4, а информация о документах в таблице 3.5-3.12.

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 договора

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

Таблица 3.6 – Атрибуты статуса договора

Объект	Атрибуты
Статус договора	Идентификационный код Название

Таблица 3.7 – Атрибуты валюты

Объект	Атрибуты
Валюта	Идентификационный код Цифровой код Буквенный код Название

Таблица 3.8 – Атрибуты контрагентов

Объект	Атрибуты
Контрагенты	Идентификационный код Наименование ИНН Телефон Электронная почта Юридический адрес Почтовый адрес БИК Номер

Таблица 3.9 – Атрибуты контрагенты договора

Объект	Атрибуты
Контрагенты договора	Идентификационный код договора Идентификационный код контрагента

Таблица 3.10 – Атрибуты служебной записки

Объект	Атрибуты
Служебная записка	Идентификационный код Идентификационный код вида Наименование Дата Содержание Идентификационный код создателя Идентификационный код адресата

Таблица 3.11 – Атрибуты виды служебной записки

Объект	Атрибуты
Виды служебной записки	Идентификационный код Название

Таблица 3.12 – Атрибуты трудового договора

Объект	Атрибуты
Трудовой договор	Идентификационный код Наименование Дата Оклад Серия Номер Кем выдан Код подразделения Дата начала Дата окончания

На рисунке 3.4 представлена логическая модель данных.

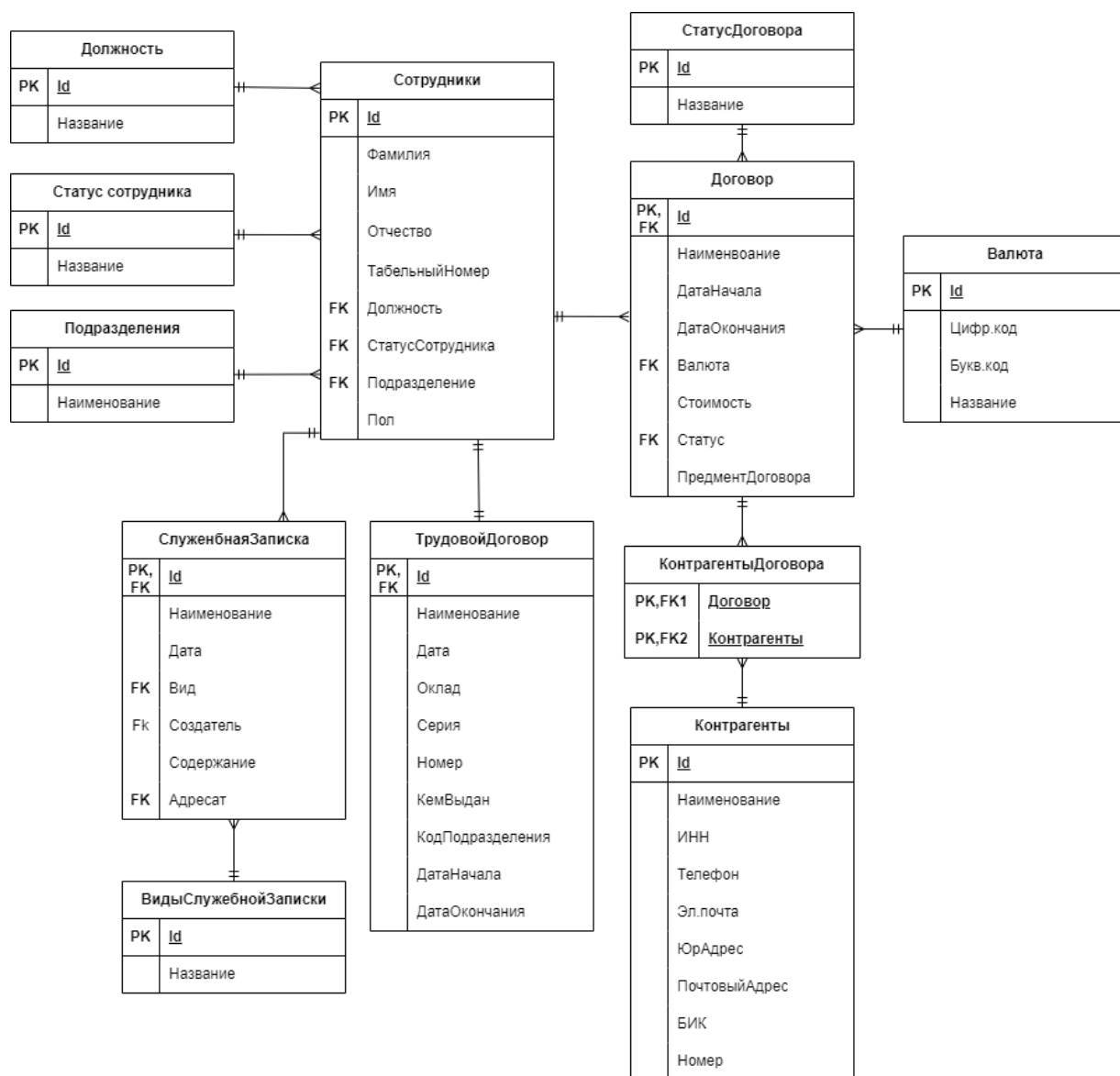


Рисунок 3.4 – Логическая модель данных

3.5 Формирование основных бизнес-процессов информационной системы

Методология моделирования бизнес-процессов – это язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией или визуализацией и воплощением бизнес-процесса. С помощью моделирования мы можем описать любые бизнес-процессы, и они могут выполняться в самых разных системах управления.

Методология моделирования бизнес-процессов является частью двух основных компонентов:

- моделирование бизнес-процессов – это среда, в которой вы непосредственно участвуете в моделировании. В одиночку или в команде;
- система моделирования бизнес-процессов – это инструменты для выполнения создаваемых вами моделей [14].

Методология моделирования бизнес-процессов представляет собой стандарт для графического отображения бизнес-процессов. Она создана для упрощения понимания и анализа процессов в различных организациях, а также для улучшения коммуникации между бизнес-аналитиками, разработчиками и менеджерами. Этот стандарт применяется для описания шагов, участников, действий и решений, которые происходят при выполнении бизнес-процессов [9].

На рисунке 3.5 представлена методология моделирования бизнес-процессов.

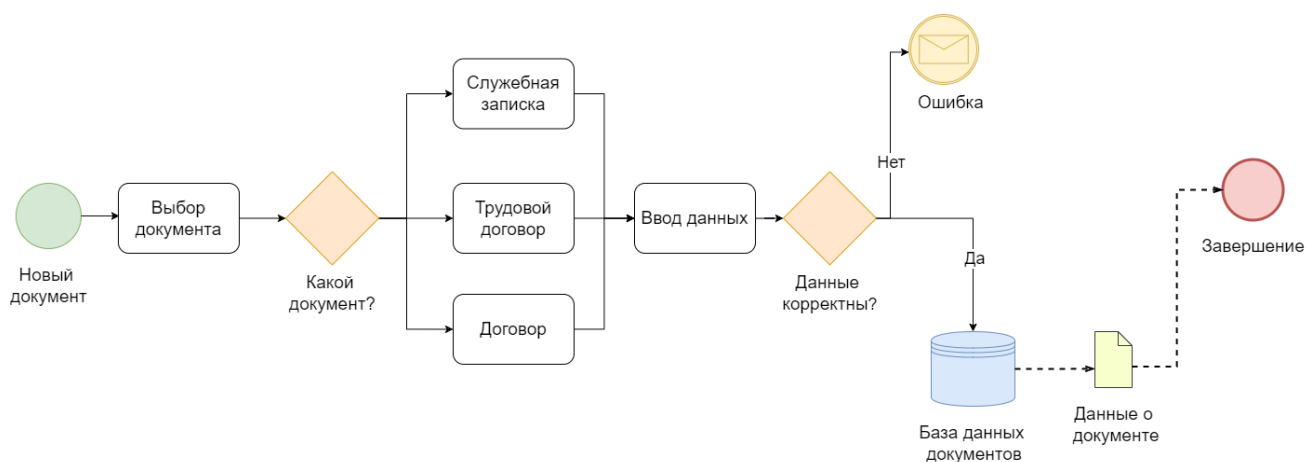


Рисунок 3.5 – Методолог моделирования бизнес-процессов

На изображении представлена диаграмма, которая описывает все шаги создания нового документа.

3.6 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;
- прототипирование позволяет более точно определить требования к продукту, включая его функциональность, интерфейс и взаимодействие с пользователем;
- сэкономить время и ресурсы на 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.

					ДП.09.02.07.24.05.ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

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 3.6 изображен прототип боковой панели.

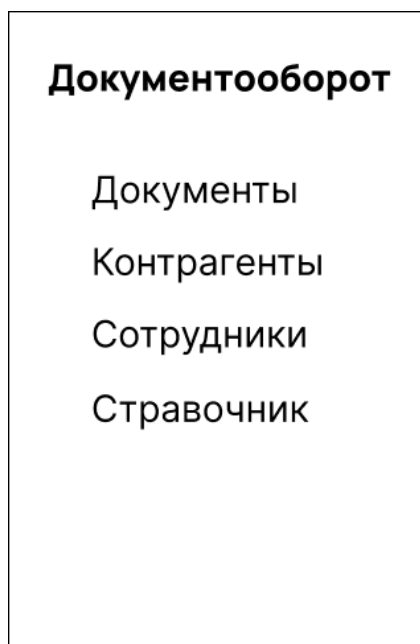


Рисунок 3.6 – Прототип боковой панели

Боковая панель – это важный и универсальный элемент пользовательского интерфейса, который обычно находится в левой или правой части основной области содержимого веб-приложения. Она содержит основное навигационное меню, которое позволяет пользователю перемещаться между различными разделами или функциями приложения [15]. Боковая панель содержит следующие кнопки:

- документы: эта кнопка открывает страницу пользователям список всех документов;
- контрагенты: эта кнопка открывает страницу пользователям список всех контрагентов;
- сотрудники: эта кнопка открывает страницу пользователям список всех сотрудников компании;
- справочник: эта кнопка открывает страницу пользователям дополнительную информацию.

Такая боковая панель удобна и практичная, что позволяет пользователям быстрее перемещаться по веб-приложению.

На рисунке 3.7 изображен прототип страницы трудового договора.

Рисунок 3.7 – Прототип страницы трудового договора

Прототип страницы трудового договора – это веб-сраница, которая представляет информацию о документе, которая содержится в виде таблицы, и она имеет поля, такие как номер документа, наименование, дату, ФИО, должность и подразделение. Также есть поиск по фамилии, имени, отчеству и фильтр, который позволяет пользователям искать документ по должности.

На рисунке 3.8 изображен прототип страницы служебной записки.

Рисунок 3.8 – Прототип страницы служебной записки

На изображении представлен прототип служебной записки и вся информация содержится в виде таблице, и она имеет поля, такие как номер документа, наименование, дату, вид, создателя, адресата. Также есть поиск по создателю и фильтр, который позволяет пользователям искать документ по виду.

На рисунке 3.9 изображен прототип страницы договора.

Рисунок 3.9 – Прототип страницы договора

На изображении представлен прототип договора. Таблица имеет поля, такие как номер, наименование, дату начала и окончания, контрагенты и статус.

На рисунке 3.10 изображен прототип страницы сотрудников.

Рисунок 3.10 – 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 3.11 изображен прототип формы добавления сотрудника.

Добавить сотрудника

Фамилия

Имя

Отчество

Пол

Должность

Табельный номер

Статус

Подразделение

Отмена Добавить

Рисунок 3.11 – 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.

4 Разработка программного продукта

Разработка программного продукта – это процесс создания программного обеспечения, который включает в себя несколько этапов, начиная от анализа требований и проектирования, и заканчивая тестированием, развертыванием и поддержкой. Основная цель разработки программного продукта - создание качественного и функционального программного решения, которое удовлетворяет потребности пользователей или бизнеса.

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.

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 [10].

4.1 Средства 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:

– позволяет определять типы данных переменных, параметров функций, возвращаемых значений и других элементов кода. Это помогает выявлять ошибки

					ДП.09.02.07.24.05.ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

на этапе компиляции, предотвращает типовые ошибки и улучшает читаемость кода;

- 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;

- поддерживает объектно-ориентированный стиль программирования, включая классы, интерфейсы, наследование, абстрактные классы, полиморфизм и другие концепции, которые делают код более структурированным и понятным;

- 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 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 [1].

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

- 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;

- 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

					ДП.09.02.07.24.05.ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

обновляет только те части, которые изменились. Это позволяет значительно повысить производительность и отзывчивость приложения;

- расширение JSX(TSX) 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;

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

- 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;

- 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 [2].

В качестве среды для разработки приложения была выбрана программа Visual Studio Code – это кроссплатформенный редактор кода, 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 в 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;

					ДП.09.02.07.24.05.ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

– 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;

– Visual Studio Code интегрируется с системой контроля версий Git, обеспечивая возможность выполнения всех основных операций Git непосредственно из редактора. Это включает в себя коммиты, ветвление, слияния, решение конфликтов и многое другое;

– 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;

– 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;

– 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;

– 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;

– Visual Studio Code является очень расширяемой средой разработки благодаря своему открытому и расширяемому API. Существует огромное сообщество разработчиков, которые создают и поддерживают разнообразные

					ДП.09.02.07.24.05.ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

расширения, темы оформления и инструменты, которые улучшают работу среды Visual Studio Code;

- 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;

- редактор кода поддерживает операционные системы Windows, macOS и Linux, что делает его универсальным инструментом для разработки на различных платформах [16].

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.

Node.js – это среда выполнения JavaScript, построенная на движке JavaScript V8 от Google Chrome. Она позволяет запускать JavaScript-код на сервере, что делает Node.js идеальным выбором для создания масштабируемых и высокопроизводительных веб-приложений. Несколько ключевых аспектов Node.js:

- поддерживает модульную архитектуру, что означает, что вы можете использовать модули из внешних источников, а также создавать свои собственные модули для повторного использования кода;

- Node.js поставляется с менеджером пакетов NPM, который является одним из самых больших репозиторий открытых исходных кодов в мире. С помощью NPM вы можете легко управлять зависимостями и подключать сторонние библиотеки к вашему проекту.

Node.js активно используется для создания веб-серверов, API, приложений реального времени, одностраничных приложений и других сценариев разработки. Он имеет широкую поддержку в сообществе разработчиков и продолжает развиваться, предоставляя новые возможности и инструменты для создания современных веб-приложений [4].

					ДП.09.02.07.24.05.ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

Express.js – это веб-фреймворк для Node.js, который предоставляет широкий спектр функций для создания веб-приложений и API. Express.js является одним из наиболее популярных фреймворков для Node.js благодаря своей простоте, гибкости и расширяемости. Он используется для создания различных типов веб-приложений, включая веб-серверы, API, одностраничные приложения и многое другое [5].

4.2 Реализация физической модели данных программного продукта

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:

- 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;
- 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;
- 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.09.02.07.24.05.ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

– 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;

– MongoDB имеет официальные драйверы и библиотеки для множества языков программирования, включая JavaScript, Python, Java, C#, Node.js и многие другие [3].

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

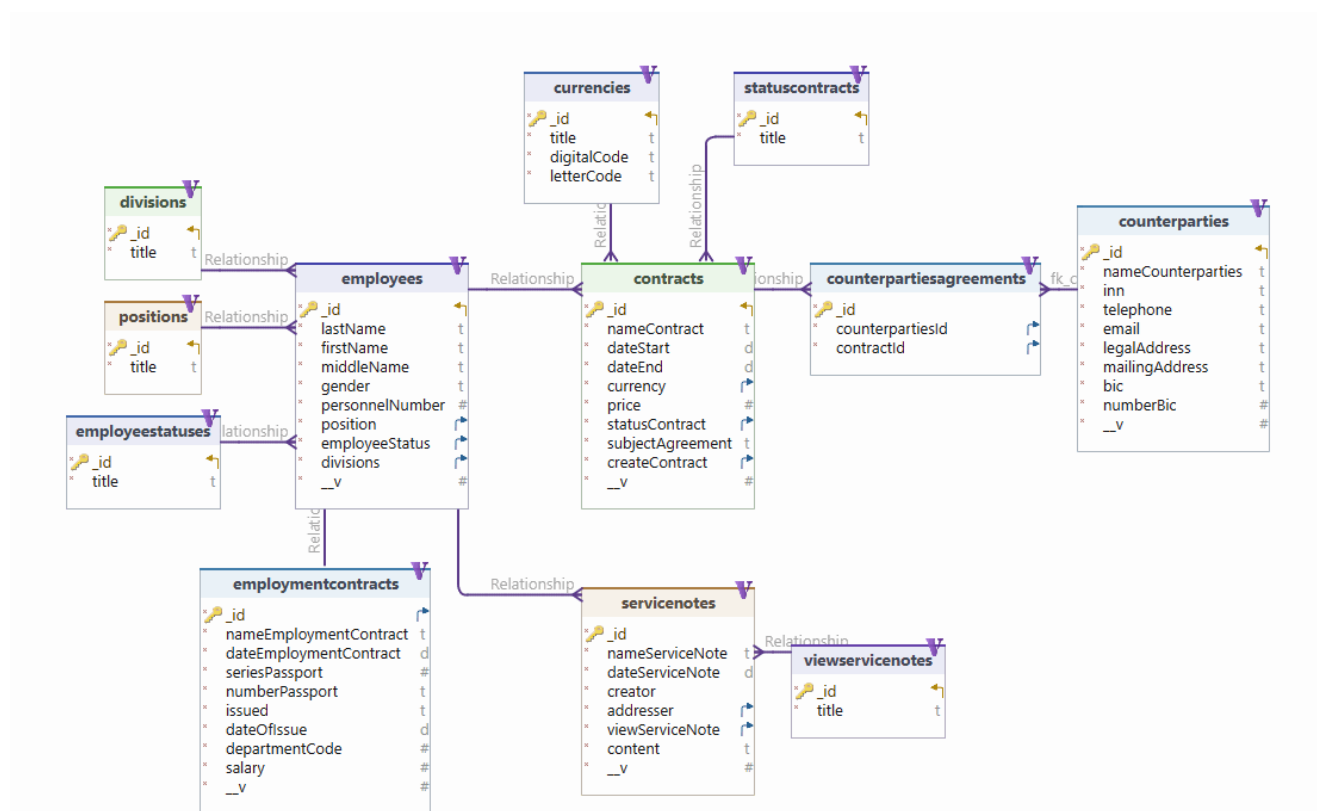


Рисунок 4.1 – Физическая модель данных

На изображении представлена физическая модель данных для системы документооборота. Эта модель включает таблицы: сотрудников, должностей, подразделений, статусов сотрудников, служебной записки, виды служебной записки, трудовой договор, договор, связи договора с контрагентами, валюта, статус договора.

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 интерфейса программного продукта – обеспечить удобство использования и эффективное взаимодействие пользователя с программой [17].

На рисунке 4.2 представлен интерфейс боковой панели.

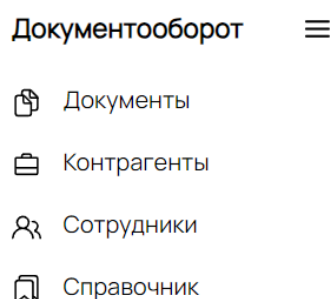


Рисунок 4.2 – Интерфейс боковой панели

На изображении представлена боковая панель, которая легко перемещаться между различными разделами «Документы» и «Сотрудники».

На рисунке 4.3 изображен интерфейс страницы сотрудников.

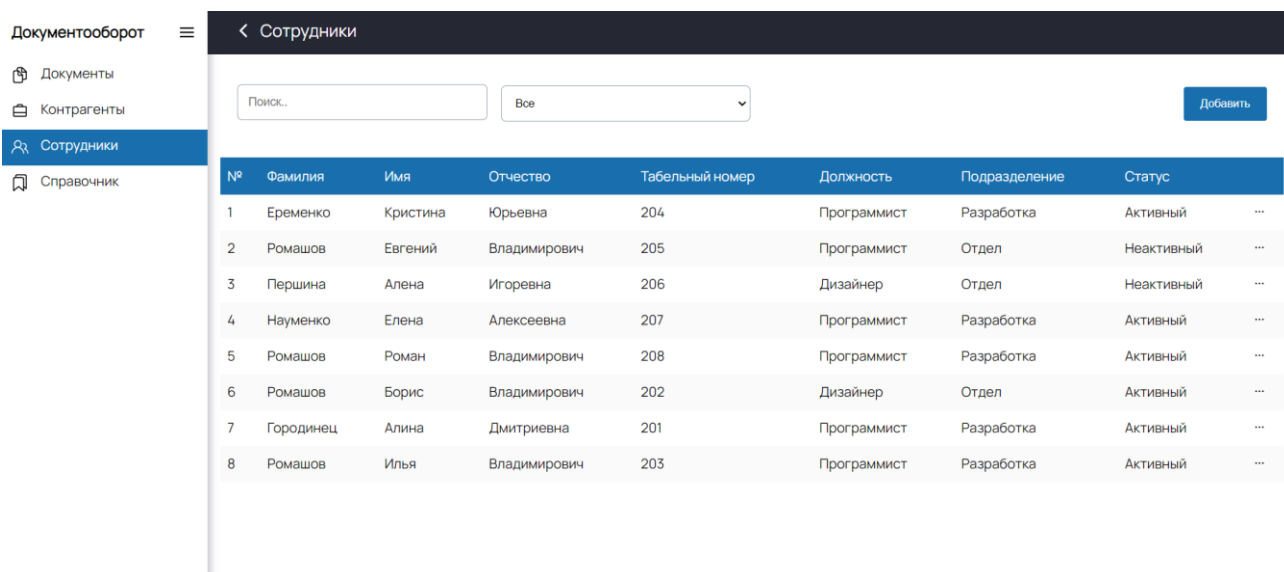


Рисунок 4.3 – Интерфейс страницы сотрудников

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.

Таким образом данный интерфейс обеспечивает удобное управление сотрудниками, позволяя легко находить нужного сотрудника, добавлять нового, редактировать существующего и удалять.

На рисунке 4.4 изображен интерфейс модального окна добавления сотрудника.

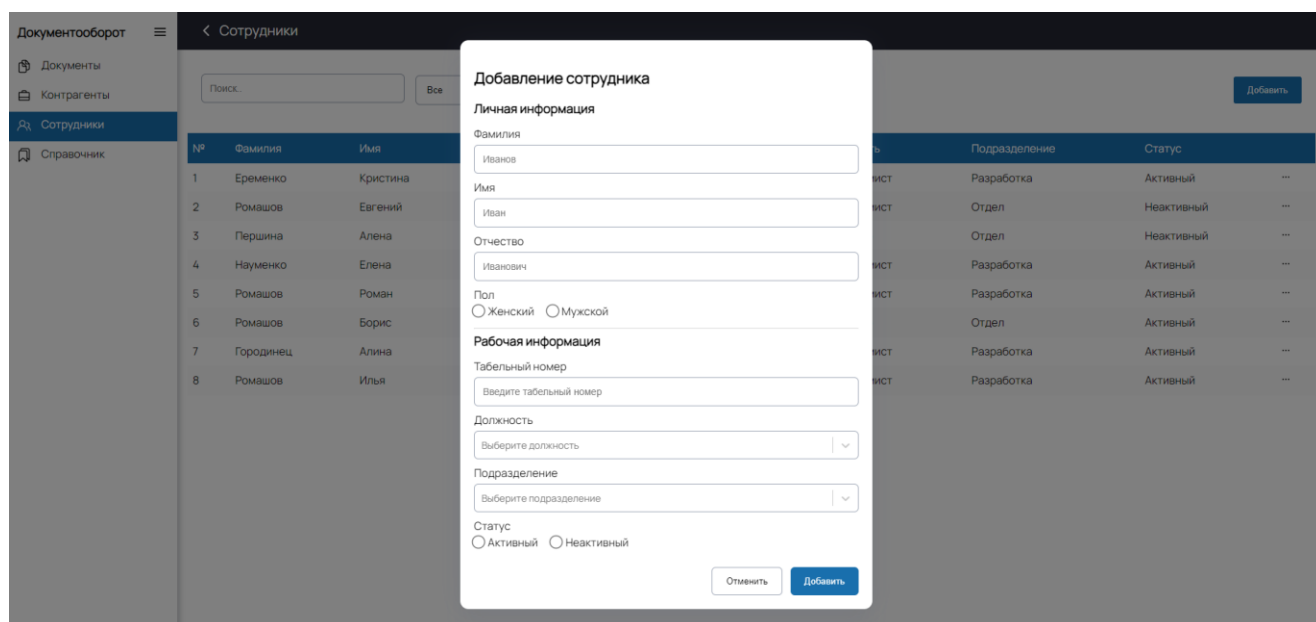


Рисунок 4.4 – Интерфейс модального окна добавления сотрудника

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

1) Личная информация:

- поле фамилия является обязательным и должно быть введено в виде фамилии сотрудника;

– поле имя является обязательным и должно быть введено в виде имени сотрудника;

– поле отчество является необязательным и может быть введено в виде отчества сотрудника;

– поле пол является обязательным и должно быть выбрано из раскрывающегося меню. Возможные варианты: «Мужской» и «Женский».

2) Рабочая информация:

– поле должность является обязательным и должно быть выбрано из раскрывающегося меню;

– поле подразделение является обязательным и должно быть выбрано из раскрывающегося меню;

– поле статус 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 4.5 изображен интерфейс модального окна редактирования сотрудника.

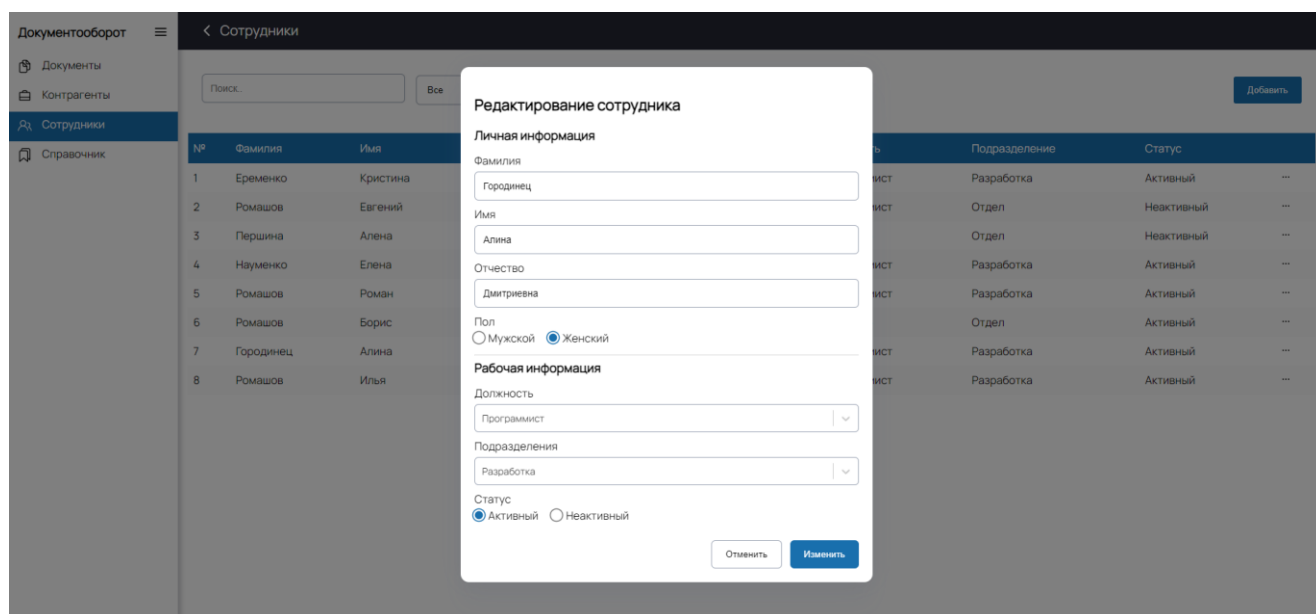


Рисунок 4.5 – Интерфейс модального окна редактирования сотрудника

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.

На рисунке 4.6 изображен интерфейс страницы трудового договора.

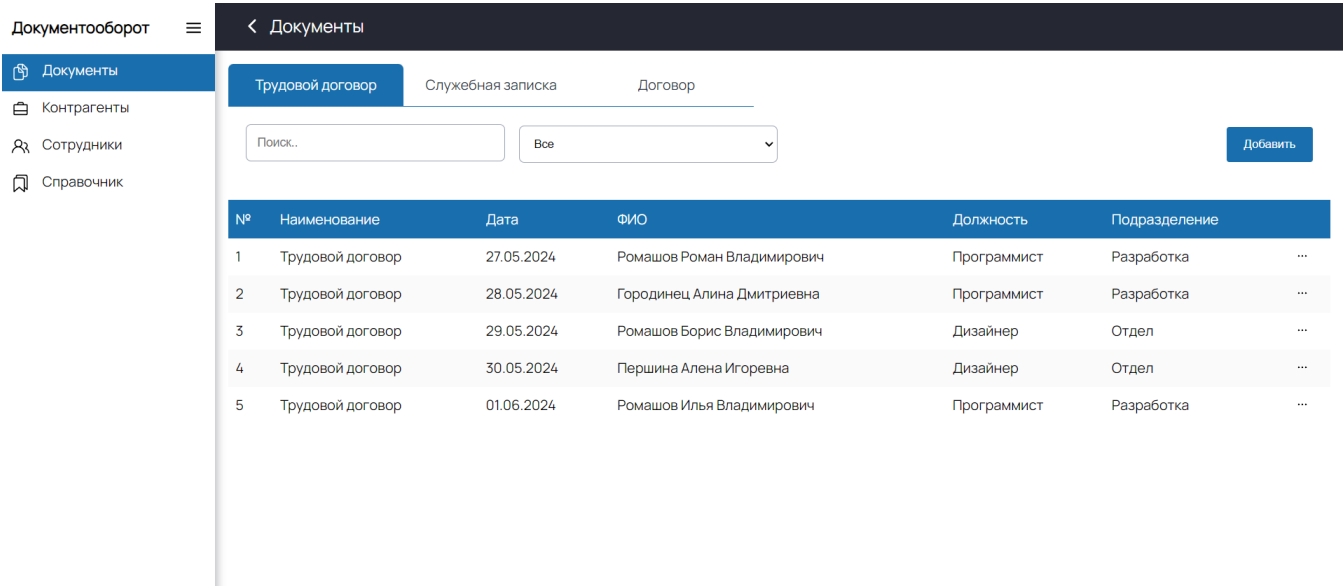


Рисунок 4.6 – Интерфейс страницы трудового договора

Интерфейс страницы трудового договора отображает список всех документов в виде таблицы и предоставляет следующие функциональности пользователям:

- просмотр полного списка;
- поиск трудового договора по фамилии, имени, отчеству;
- фильтрация по должности;
- добавление новых документов в список;
- удаление документов;
- просмотр полной информации трудового договора.

Интерфейс должен быть понятным, чтобы пользователи могли легко найти нужный им договор. При разработке интерфейса страницы трудового договора важно учитывать потребности пользователей и обеспечить им простой и удобный способ просмотра, поиска и управления трудовыми договорами.

На рисунке 4.7 изображен интерфейс страницы служебной записки.

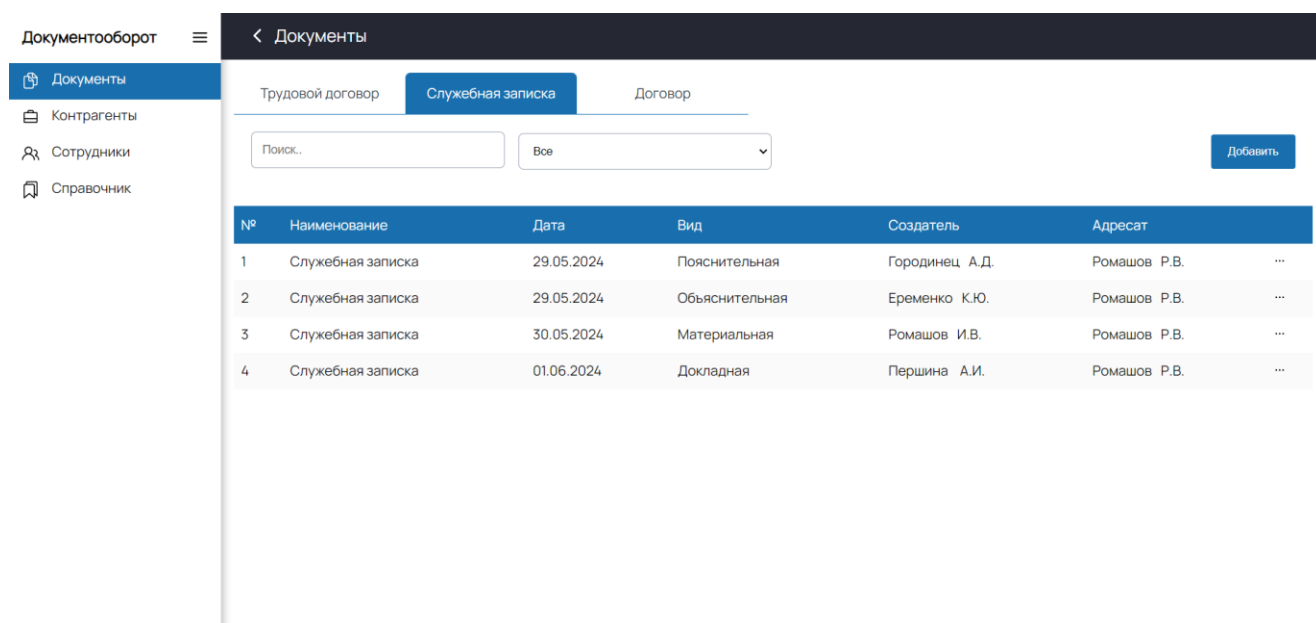


Рисунок 4.7 – Интерфейс страницы служебной записки

Интерфейс служебной записки предназначен для того, чтобы предоставить пользователям простой и удобный способ создания, просмотра и управления служебными записками. Он должен быть понятным, чтобы пользователи могли легко найти нужную информацию и выполнять необходимые действия. Веб-приложение страницы служебной записки отображает список всех документов в виде таблицы и предоставляет следующие функциональности пользователям:

- просмотр полного списка;
- поиск служебной записки по создателю;
- фильтрация по виду документа;
- добавление новых документов в список;
- удаление документов;
- просмотр полной информации документа.

Интерфейс страницы служебной записки является простым и удобным в использовании. Он позволяет пользователям легко находить, просматривать, создавать и управлять служебными записками.

На рисунке 4.8 изображен интерфейс страницы добавления трудового договора.

Рисунок 4.8 – 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 4.9 изображен интерфейс страницы добавления служебной записки.

Рисунок 4.9 – 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 4.10 изображен интерфейс добавления договора.

Рисунок 4.10 – 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.

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 4.11 изображена фильтрация данных.

Рисунок 4.11 – Фильтрация данных

					ДП.09.02.07.24.05.ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

При работе с веб-приложением вы можете использовать функцию фильтрации данных. Для этого откройте соответствующий раздел и найдите опцию фильтрации. Выберите нужные критерии для фильтрации данных. После применения фильтров приложение покажет только те данные, которые соответствуют вашим параметрам. Это поможет вам быстро находить нужную информацию и эффективно управлять проектами.

На рисунке 4.12 изображен поиск данных.



Рисунок 4.12 – Поиск данных

При работе с веб-приложением вы можете использовать функцию поиска данных. Для этого перейдите в соответствующий раздел и найдите поле для ввода поискового запроса. Введите ключевые слова или фразы, которые вы хотите найти. Приложение покажет результаты, соответствующие вашему запросу. Функция поиска облегчит вам быстрый доступ к необходимой информации и упростит работу с приложением.

На изображении 4.13 изображена кнопка добавления данных.

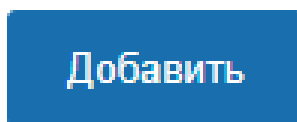


Рисунок 4.13 – Кнопка добавления данных

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 4.14 изображено модальное окно с кнопкой редактирования данных.

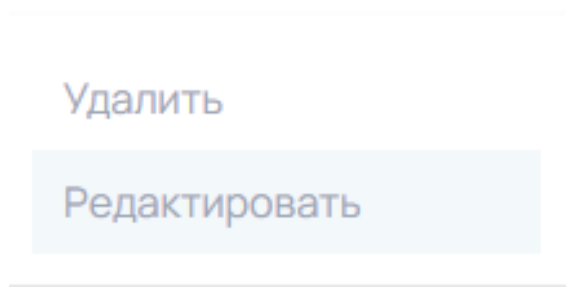


Рисунок 4.14 – Кнопка редактирования данных

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 4.15 изображено модальное окно с кнопкой удаления данных.

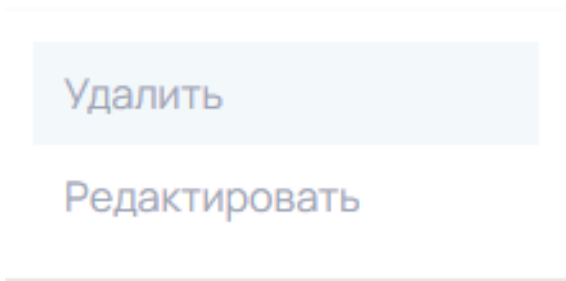


Рисунок 4.15 – Кнопка удаления данных

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 4.16 изображено модальное окно с кнопкой подробнее.

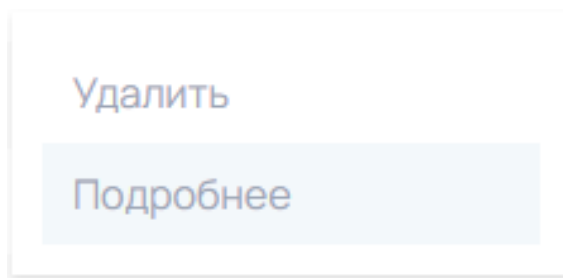


Рисунок 4.16 – Кнопка подробнее

Кнопка подробнее позволяет отобразить дополнительную информацию, которая не была выведена на основную страницу. Она является полезным инструментом, который позволяет пользователям быстро получать доступ к информации о документах.

На рисунке 4.17 изображено подтверждающее окно.

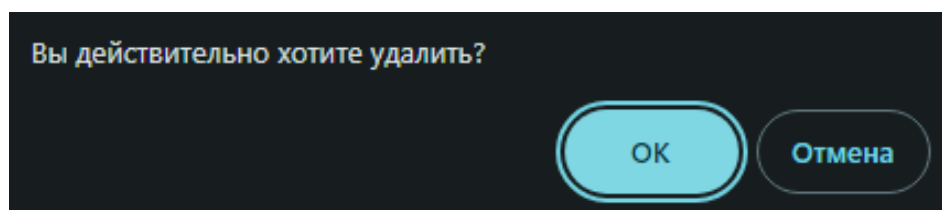


Рисунок 4.17 – Подтверждающее окно

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

В расчёте экономической части дипломного проекта используются следующие статьи затрат:

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

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

Таблица 5.1 – Этапы создания программы

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

Расчет материальных затрат.

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

Таблица 5.2 – Расчет материальных затрат

Наименование	Единицы измерения	Кол-во (шт.)	Цена (руб.)	Сумма (руб.)
Ручка	шт.	1	60	60
Оплата интернета	мес.	1	600	600
Папка	шт.	1	226	226
Бумага	уп.	1	199	199
Печать на принтере	лист	69	10	690
Итого:				1775

Расчет амортизационных отчислений.

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

$$A = S \times N, \quad (1)$$

где S – первоначальная стоимость;

N – годовая норма амортизации.

Расчет суммы годовой амортизации выполняется по формуле 1.

где S – первоначальная стоимость;

N – годовая норма амортизации.

Расчет суммы годовой амортизации выполняется по формуле 1.

$$A(\text{стол}) = 4000 \times 10\% = 400,00 \text{ руб.};$$

$$A(\text{компьютер}) = 61190 \times 33,3\% = 20376,27 \text{ руб.};$$

$$A(\text{стул}) = 2900 \times 10\% = 290,00 \text{ руб.}$$

Расчет месячной суммы амортизации проводится по формуле 2:

$$A_{\text{мес}} = \frac{A_{\text{год}}}{12}, \quad (2)$$

где $A_{\text{мес}}$ – месячная сумма амортизации, руб.;

$A_{\text{год}}$ – годовая сумма амортизации, руб.;

12 – количество месяцев в году, мес.

Проведем расчет месячной суммы амортизации по формуле 2:

$$A(\text{мес}) = 21066,27 / 12 = 1755,52 \text{ руб.}$$

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

$$A = ((S \times N) / Rd) \times t, \quad (3)$$

где S – первоначальная стоимость, руб.;

N – годовая норма амортизации;

Rd – количество рабочего времени, ч.;

t – фактически затраченное время на работу, ч.

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

$$A(\text{вр}) = (1755,52 / 160) \times 40 = 438,88 \text{ руб.}$$

					ДП.09.02.07.24.05.ПЗ	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

Расчет амортизации основных фондов показан в таблице 5.3.

Таблица 5.3 – Расчет амортизации основных фондов

Наименование основных фондов	Первоначальная стоимость (руб.)	Норма амортизации	Годовая сумма амортизации (руб.)	Амортизация на выполнение проекта (руб.)
Стол	4000	10%	400,00	438,88
Компьютер	61190	33,3%	20376,27	
Стул	2900	10%	290,00	
Итого:	68098		21066,27	

Расчеты расходов на электроэнергию показаны в таблице 5.4.

Таблица 5.4 – Расходы на электроэнергию

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

Расчет заработной платы.

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

Заработная плата за день работы приведена в формуле 4:

$$Z_{\text{д.}} = Z_{\text{мес.}}/D, \quad (4)$$

где $Z_{\text{д.}}$ – зарплата за 1 день,

$Z_{\text{мес.}}$ – зарплата за 1 месяц,

D – количество рабочих дней в месяце.

Расчет заработной платы приведен в таблице 5.5.

Таблица 5.5 – Расчет заработной платы

Тарифная ставка	Кол-во часов в мес.	Кол-во часов потраченных на написание ПП	Итоговая сумма (руб.)
190	176	160	30400

Расчет коммунальных услуг показан в таблице 5.6.

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

Таблица 5.6 – Расходы на коммунальные услуги

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

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

Таблица 5.7 – Себестоимость по прямым затратам

Наименование статей затрат	Сумма (руб.)
Материальные затраты	1775,00
Заработная плата	30400,00
Амортизация основных фондов	438,88
Расходы на электроэнергию	396,83
Прочие затраты	279,82
Итого:	33287,53

Калькуляция рассчитана по прямым затратам.

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

$$\Pi = (C \times 15) / 100, \quad (5)$$

$$Ц = C + \Pi, \quad (6)$$

где Π – прибыль;

C – себестоимость;

$Ц$ – цена.

Рассчитаем цену программного продукта по формулам 5 и 6.

$$\Pi = (33287,53 \times 15) / 100 = 4993,13 \text{руб.};$$

$$Ц = 33287,53 + 4993,13 = 38280,66 \text{руб.}$$

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

6 Охрана труда и техника безопасности

Требования к помещениям для эксплуатации персональных компьютеров:

– площадь на одно рабочее место пользователей на базе плоских дискретных экранов – 4,5 кв.м, помещение с компьютером должны оборудоваться системами отопления, кондиционирования воздуха или эффективной вентиляцией воздуха [18].

Требования к микроклимату:

– в производственных помещениях, в которых работа на ПК является основной, должны обеспечиваться оптимальные параметры микроклимата;

– для повышения влажности воздуха в помещениях с ПК следует применять увлажнители воздуха.

Требования к освещению помещений и рабочих мест с персональным компьютером:

– помещения с ПК должны иметь естественное и искусственное освещение;

– устанавливать компьютер на рабочем месте таким образом, чтобы солнечные лучи не падали на экран [19].

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

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

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

					ДП.09.02.07.24.05.ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

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

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

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

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений [21, 22].

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

- для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей, на протяжении работы должны устанавливаться регламентированные перерывы;

- продолжительность непрерывной работы с ПК без регламентированного перерыва не должна превышать 2 часов;

					ДП.09.02.07.24.05.ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

– при работе на ПК для предупреждения развития переутомления необходимо осуществлять комплекс профилактических мероприятий.

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

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

Мощность экспозиционной дозы мягкого рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана и корпуса ВДТ (на электронно-лучевой трубке) при любых положениях регулировочных устройств не должна превышать 1 мкЗв/ч (100 мкР/ч) [23, 24].

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

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

В производственных помещениях, где работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а и 1б согласно действующим санитарно-эпидемиологическим нормативам. На других рабочих местах микроклимат должен поддерживаться на допустимом уровне. Документация на проектирование, изготовление и эксплуатацию ПЭВМ должна соответствовать этим санитарным правилам [20].

					ДП.09.02.07.24.05.ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

Разработанное веб-приложение для документооборота успешно решает задачу создания удобного инструмента для управления документами в корпоративной среде. Целью данного проекта было разработать веб-приложение, которое бы упростило и оптимизировало процессы документооборота в организации, обеспечивая удобный доступ к документам, их хранение и обработку.

Основные результаты проекта включают:

- создание и хранение информации о сотрудниках, а также обновления личных данных;
- позволяет хранить документы в системе, создавать и сортировать;
- пользовательский интерфейс способствует быстрому освоению системы пользователями;
- веб-приложение значительно сократило время на создание, поиск и управление документами;
- обеспечение безопасности данных посредством авторизации и аутентификации пользователей;
- быстрый поиск и доступ к нужным документам из любой точки и в любое время;
- автоматизация рутинных процессов уменьшит нагрузку на сотрудников;

Проект показал, что разработанное веб-приложение для корпоративного документооборота значительно улучшил процессы управления документами, сделал их более эффективными и безопасными. Внедрение такой системы помогает организациям сократить время и трудности, повысить контроль над документами и обеспечить высокую степень защиты данных.

					ДП.09.02.07.24.05.ПЗ	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

Список использованной литературы

- 1 Фримен Адам. Основы TypeScript – Питер, 2024. – 576 с.
- 2 Бэнкс Алекс, Порселло Ева. React: современные шаблоны для разработки приложений. 2-е изд. – СПб.: Питер, 2022. – 320 с.
- 3 Шеннон Брэдшоу, Йон Брэзил, Кристина Ходоров. MongoDB: полное руководство. Мощная и масштабируемая система управления базами данных – ДМК Пресс, 2020. – 540 с.
- 4 Дейли Брэд, Дейли Брендан, Дейли Калев. Разработка веб-приложений с помощью Node.js – Вильямс, 2020. – 656 с.
- 5 Скотт Адам Д., Пауэрс Шелли, Мэтью Макдоналд. JavaScript рецепты для разработчика – Питер, 2023. – 528 с.
- 6 Проектирование информационных систем: учебник и практикум для вузов / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук; под общей редакцией Д. В. Чистова. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 293 с. – (Высшее образование). – ISBN 978-5-534-15923-3. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/536195> (дата обращения: 19.04.2024).
- 7 Советов, Б. Я. Базы данных: учебник для вузов / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – 4-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 403 с. – (Высшее образование). – ISBN 978-5-534-18479-2. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/535113> (дата обращения: 25.04.2024).
- 8 Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML: учебное пособие для вузов / Э. Г. Галиаскаров, А. С. Воробьев. – Москва: Издательство Юрайт, 2024. – 125 с. – (Высшее образование). – ISBN 978-5-534-14903-6. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/544559> (дата обращения: 21.04.2024).

					ДП.09.02.07.24.05.ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

9 Долганова, О. И. Моделирование бизнес-процессов: учебник и практикум для вузов / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова; под редакцией О. И. Долгановой. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 322 с. – (Высшее образование). – ISBN 978-5-534-17914-9. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/536465> (дата обращения: 28.04.2024).

10 Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения: учебное пособие для среднего профессионального образования / С. А. Чернышев. – Москва: Издательство Юрайт, 2024. – 176 с. – (Профессиональное образование). – ISBN 978-5-534-18705-2. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/545401> (дата обращения: 8.05.2024).

Интернет ресурсы

11 Диаграмма потоков данных. [Электронный ресурс] – Режим доступа: <https://practicum.yandex.ru/blog/diagramma-potokov-dannyh-dfd/>.

12 Архитектура программного продукта [Электронный ресурс] – Режим доступа: <https://wezom.com.ua/blog/arhitektura-programmnogo-obespecheniya>.

13 Концептуальная модель [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/blog/modelirovanie-dannykh-v-rdbms>.

14 Методология моделирования бизнес-процессов [Электронный ресурс] – Режим доступа: <https://habr.com/ru/companies/auriga/articles/667084/>

15 Боковая панель [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/glossary/bokovaia-panel>.

16 Visual Studio Code [Электронный ресурс] – Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.f71b2df9-664b1592-ac089bcf-74722d776562/https/code.visualstudio.com/Docs/editor/codebasics.

17 Интерфейс программного продукта [Электронный ресурс] – Режим доступа: <https://blog.skillfactory.ru/glossary/interface/>.

					ДП.09.02.07.24.05.ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

18 Основные требования для помещений [Электронный ресурс] – Режим доступа: <https://ohranatruda.ru>.

19 Требования международных стандартов [Электронный ресурс] – Режим доступа: <https://technormativ.ru>.

20 Производственные факторы [Электронный ресурс] – Режим доступа: <https://legalacts.ru>.

21 Требования к рабочему месту [Электронный ресурс] – Режим доступа: <http://docs.cntd.ru>.

22 Конструкция рабочего места [Электронный ресурс] – Режим доступа: <https://studwood.ru>.

23 ПЭВМ [Электронный ресурс] – Режим доступа: <http://prom-nadzor.ru>.

24 Работа с ПЭВМ [Электронный ресурс] – Режим доступа: <http://mirznaii.com>.

					ДП.09.02.07.24.05.ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение
(обязательное)
Листинг программы

```
const root = ReactDOM.createRoot(
  document.getElementById('root') as HTMLElement
);
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
import React from 'react';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import './App.css';
import EmployeesPage from './pages/employees/EmployeesPage';
import Contract from
'./pages/documents/contractPage/CreateContractPage';
import UploadingFile from
'./pages/documents/uploadingFilePage/UploadingFilePage';
import PDFServiceNote from
'./pages/documents/serviceNotePage/PDFServiceNote';
import ServiceNotePage from
'./pages/documents/serviceNotePage/ServiceNotePage';
import CreateServiceNote from
'./pages/documents/serviceNotePage/CreateServiceNotePage';
import EmploymentContractPage from
'./pages/documents/employmentContractPage/EmploymentContractPage';
import CreateEmploymentContract from
'./pages/documents/employmentContractPage/CreateEmploymentContractPage';
import ReferenceBook from './pages/referenceBook/ReferenceBook';
import Counterparties from
'./pages/counterparties/CounterpartiesPage';
```

					ДП.09.02.07.24.05.ПЗ	Лист
						55
Изм.	Лист	№ докум.	Подпись	Дата		

```

import PDFEmploymentContract from
'./pages/documents/employmentContractPage/PDFEmploymentContract';
import ContractPage from
'./pages/documents/contractPage/ContractPage';
function App() {
  return (
    <>
      <BrowserRouter>
        <Routes>
          <Route path='/documents'
element={<EmploymentContractPage/>}/>
          <Route
path='/documents/createDocument/employmentContract'
element={<CreateEmploymentContract/>}/>
          <Route
path='/documents/createDocument/employmentContract/pdf/:id'
element={<PDFEmploymentContract/>}/>
          <Route path='/documents/serviceNote'
element={<ServiceNotePage/>}/>
          <Route path='/documents/createDocument/serviceNote'
element={<CreateServiceNote/>}/>
          <Route
path='/documents/createDocument/serviceNote/pdf/:id'
element={<PDFServiceNote/>}/>
          <Route path='/documents/contract'
element={<ContractPage/>}/>
          <Route path='/documents/createDocument/contract'
element={<Contract/>}/>
          <Route path='/documents/createDocument/file'
element={<UploadingFile/>}/>
          <Route path='/counterparties'
element={<Counterparties/>}/>
          <Route path='/employees' element={<EmployeesPage/>}/>
          <Route path='/referenceBook'
element={<ReferenceBook/>}/>

```

					ДП.09.02.07.24.05.ПЗ	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		

```

        </Routes>
      </BrowserRouter>
    </>
  );
}
export default App;
export interface ServiceNote {
  _id: string;
  nameServiceNote: string;
  dateServiceNote: Date;
  creator: Employee;
  addresser: {
    _id: string;
    lastName: string;
    firstName: string;
    middleName: string;
  },
  viewServiceNote: {
    _id: string;
    title: string;
  };
  content: string;
}
export interface Employee {
  _id: string;
  lastName: string;
  firstName: string;
  middleName: string;
}
import React, { useEffect, useState } from "react";
import { ServiceNote } from "../DocumentInterface";
import { IoIosArrowBack } from "react-icons/io";
import { HiEllipsisHorizontal } from "react-icons/hi2";
import axios from "axios";
import { Link } from "react-router-dom";

```

					ДП.09.02.07.24.05.ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

```

import TabDocumentsPage from
"../../../components/tabDocumentsPage/TabDocumentsPage";
import "../documentsPage/Document_style.css";
import Sidebar from "../../../components/sidebar/Sidebar";
import { sidebarItems } from
"../../../components/sidebar/DataSidebar";
import { PDFDownloadLink } from "@react-pdf/renderer";
import PDFDocument from "../PDFDocument";
import "../../../style/Global_style.css"
const URL = process.env.REACT_APP_URL;
function ServiceNotePage() {
  const [dataServiceNote, setDataServiceNote] =
useState<ServiceNote[]>([]);
  const [openPopoverId, setOpenPopoverId] = useState<string |
null>(null);
  const [filteredServiceNote, setFilteredServiceNote] =
useState<ServiceNote[]>([]);
  const [searchQuery, setSearchQuery] = useState<string>("");
  const [filter, setFilter] = useState<string>("");
  useEffect(() => {
    axios
      .get(`${URL}/get/serviceNote`)
      .then((res) => {
        setDataServiceNote(res.data);
      })
      .catch((err) => console.log(err));
  }, []);
  const handleDelete = (id: string) => {
    if (window.confirm(`Вы действительно хотите удалить?`)) {
      axios
        .delete(`${URL}/delete/serviceNote/${id}`)
        .then((res) => {
          console.log(res);
          // Обновляем данные после удаления сотрудника
          setDataServiceNote(

```

					ДП.09.02.07.24.05.ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

```

        dataServiceNote.filter((serviceNote) => serviceNote._id
    !== id)

        );
    })
    .catch((err) => console.log(err));
}
};
useEffect(() => {
    const filterDocument = () => {
        let filteredData = dataServiceNote;
        if (filter !== "") {
            filteredData = filteredData.filter(
                (serviceNote) =>
                    serviceNote.viewServiceNote &&
                    serviceNote.viewServiceNote.title === filter
            );
        }
        if (searchQuery !== "") {
            filteredData = filteredData.filter((serviceNote) => {
                const fullName = `${serviceNote.creator.lastName}
${serviceNote.creator.firstName} ${serviceNote.creator.middleName}`;
                return
                fullName.toLowerCase().includes(searchQuery.toLowerCase());
            });
        }
        setFilteredServiceNote(filteredData);
    };
    filterDocument();
}, [dataServiceNote, filter, searchQuery]);
const togglePopover = (id: string) => {
    setOpenPopoverId(openPopoverId === id ? null : id);
};
useEffect(() => {
    const handleClickOutside = (event: any) => {
        if (!event.target.closest(".HiEllipsisHorizontal"))

```

					ДП.09.02.07.24.05.ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подпись	Дата		

```

        setOpenPopoverId(null);
    };
    document.addEventListener("click", handleClickOutside);
    return () => {
        document.removeEventListener("click", handleClickOutside);
    };
}, []);
return (
    <>
        <Sidebar items={sidebarItems}>
            <div className={"header_documents"}>
                <div className={"header_content"}>
                    <IoIosArrowBack className={"arrow_documents"} />
                    <p>Документы</p>
                </div>
            </div>
            <section>
                <TabDocumentsPage />
                <div className={"container_navigate"}>
                    <div className={"container_search_filter"}>
                        <div className={"search"}>
                            <input
                                type="text"
                                name="search"
                                placeholder="Поиск.."
                                value={searchQuery}
                                onChange={(e) => setSearchQuery(e.target.value)}
                            />
                        </div>
                        <div className={"container_filter"}>
                            <select
                                value={filter}
                                onChange={(e) => setFilter(e.target.value)}
                                className={"filter"}
                            >

```

					ДП.09.02.07.24.05.ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

```

        <option value="">Все</option>
        <option value="Докладная">Докладная</option>
        <option
value="Пояснительная">Пояснительная</option>
        <option
value="Объяснительная">Объяснительная</option>
        <option value="Материальная">Маретиальная</option>
    </select>
</div>
</div>
<div className={"btn_add_document"}>
    <Link to="/documents/createDocument/serviceNote">
        <button className={"add_document"}>Добавить</button>
    </Link>
</div>
</div>
<table>
    <thead>
        <tr>
            <th>№</th>
            <th>Наименование</th>
            <th>Дата</th>
            <th>Вид</th>
            <th>Создатель</th>
            <th>Адресат</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        {filteredServiceNote.map((serviceNote, index) => (
            <tr key={index} data-documentId={serviceNote._id}>
                <td>{index + 1}</td>
                <td>{serviceNote.nameServiceNote}</td>
                <td>

```

					ДП.09.02.07.24.05.ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

```

        {new
Date(serviceNote.dateServiceNote).toLocaleDateString())}
      </td>
      <td>{serviceNote.viewServiceNote.title}</td>
      <td>
        {serviceNote.creator.lastName}
        {serviceNote.creator.firstName.charAt(0)}.
        {serviceNote.creator.middleName.charAt(0)}.
        {/* {serviceNote.creator.map((creator) => {
return `${creator.lastName}
${creator.firstName.charAt(
0
)}}. ${creator.middleName.charAt(0)}.`;
}}) */}
      </td>
      <td>
        {serviceNote.addresser.lastName}
        {serviceNote.addresser.firstName.charAt(0)}.
        {serviceNote.addresser.middleName.charAt(0)}.
      </td>
      <td>
        <HiEllipsisHorizontal
          className="HiEllipsisHorizontal"
          onClick={() => togglePopover(serviceNote._id)}
        />
        {openPopoverId === serviceNote._id && (
          <div className="popup">
            <div className="popup_content">
              <div
                onClick={() =>
handleDelete(serviceNote._id)}
                className="button_delete"
              >
                <p>Удалить</p>

```

					ДП.09.02.07.24.05.ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подпись	Дата		


```

        </div>
        <div className="button_details">
            <Link
to={` /documents/createDocument/serviceNote/pdf/${serviceNote._id}`}
            >
                <p>Подробнее</p>
            </Link>
        </div>
        { /* <PDFDownloadLink
document={ <PDFDocument /> } fileName="ServiceNote.pdf">
            ({ { loading }) => (loading ? 'Loading
document...' : 'Download now!') }
            </PDFDownloadLink> */ }
        </div>
    </div>
    )}
</td>
</tr>
    )}
</tbody>
</table>
</section>
</Sidebar>
</>
    );
}

export default ServiceNotePage;
import React, { useEffect, useRef, useState } from "react";
import axios from "axios";
import { useParams } from "react-router-dom";
import { useReactToPrint } from "react-to-print";
import { ServiceNote } from "../DocumentInterface";
import "../ServiceNote_style.css";
const URL = process.env.REACT_APP_URL;
function PDFServiceNote({ serviceNoteId }: any) {

```

					ДП.09.02.07.24.05.ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

```

const { id } = useParams<{ id: string }>();
const [dataServiceNote, setDataServiceNote] = useState<ServiceNote
| null>(
  null
);
useEffect(() => {
  axios
    .get(`${URL}/get/serviceNote/${id}`)
    .then((res) => {
      setDataServiceNote(res.data);
    })
    .catch((err) => console.log(err));
}, []);
const component = useRef<HTMLDivElement>(null);
const handlePrint = useReactToPrint({
  content: () => component.current,
  documentTitle: "Служебная записка",
});
if (!dataServiceNote) {
  return <div>Загрузка...</div>;
}
return (
  <>
    <div ref={component} style={{ position: "relative", margin:
"35px" }}>
      <div>
        <p style={{ textAlign: "end" }}>
          Уважаемый(ая)
            {`${dataServiceNote.addresser.lastName}
${dataServiceNote.addresser.firstName}
${dataServiceNote.addresser.middleName}`}
        </p>
        <h3 style={{ textAlign: "center" }}>
          {dataServiceNote.nameServiceNote}
        </h3>

```

					ДП.09.02.07.24.05.ПЗ	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

```

        <p>{dataServiceNote.viewServiceNote.title}</p>
        <p style={{ textAlign: "justify", textIndent: "20px" }}>
            {dataServiceNote.content}
        </p>
        <p style={{ textAlign: "end" }}>
            `{dataServiceNote.creator.lastName}
            {dataServiceNote.creator.firstName}
            {dataServiceNote.creator.middleName}`
        </p>
    </div>
</div>
<div className={"container_btn_print"}>
    <button onClick={handlePrint} className={"print"}>
        Печатать
    </button>
</div>
</>
);
}
export default PDFServiceNote;

```

					ДП.09.02.07.24.05.ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		