Geekbrains

**РАЗРАБОТКА ПРОЕКТА ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ НА С#.**

IT-специалисты:

Разработка проекта

Деркач П.И.

Архангельск 2024

ОГЛАВЛЕНИЕ

[Введение 5](#_bookmark0)

1 [Разработка системного проекта](#_bookmark1) 7

1.1 [Назначение разработки](#_bookmark2) 7

1.1.1 [Описание деятельности предприятия](#_bookmark3) 8

1.1.2 [Анализ требований к проекту](#_bookmark4) 7

1.1.3 Анализ готовых программных решений 9

1.2 [Требования к функциональным характеристикам 1](#_bookmark5)1

1.2.1 [Функциональные требования 11](#_bookmark6)

1.2.2 [Архитектура разрабатываемого приложения](#_bookmark7) 12

1.3 [Требования к надежности и безопасности 1](#_bookmark8)4

1.5 [Требования к составу и параметрам технических средств 15](#_bookmark9)

1.6 [Требования к информационной и программной совместимости 1](#_bookmark10)6

2 [Разработка технического проекта 1](#_bookmark11)8

2.1 [[Анализ требований и определение спецификаций программного](#_bookmark12) [обеспечения 1](#_bookmark12)](#_bookmark11)8

2.2 [Проектирование модели данных 26](#_bookmark13)

2.3 [Детальное проектирование программного обеспечения 30](#_bookmark13)

3 [Реализация 3](#_bookmark15)8

[3.1. Обоснование выбора средств разработки 3](#_bookmark16)8

[3.2 Описание основных программных модулей 3](#_bookmark17)9

4 [Тестирование программного продукта 42](#_bookmark18)

[Заключение 44](#_bookmark19)

[Список использованных источников 45](#_bookmark20)

[Приложение А (обязательное) Техническое задание](#_bookmark21) 48

[Приложение Б (обязательное) Руководство пользователя 5](#_bookmark22)5

Приложение B (обязательное) Листинг кода 72

ВВЕДЕНИЕ

Современные компании, работающие с большим количеством информации и документами, сталкиваются с необходимостью автоматизации документооборота. Для компании, занимающейся обработкой и хранением данных, эффективная система управления документооборотом становится важным инструментом для повышения прозрачности рабочих процессов, оптимизации операций и улучшения контроля.

Актуальность данного проекта заключается в создании системы управления документооборотом, которая позволит автоматизировать работу с документами: от их создания и согласования до архивации и отслеживания состояния на каждом этапе. Такая система обеспечит контроль за жизненным циклом документов, минимизируя временные и трудозатратные издержки, а также повышая доступность и безопасность данных. Решение задач, связанных с автоматизацией документооборота, становится ключевым элементом для компаний, стремящихся оптимизировать работу сотрудников и снизить риски ошибок и потерь.

Целью проекта является разработка модуля информационной системы, которая упростит и ускорит управление документооборотом в компании, и повысит качество обмена информацией.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Создание структуры системы управления документооборотом, охватывающей процессы согласования, утверждения и мониторинга документов;
2. Разработка функций автоматизированного контроля и управления статусами документов на разных этапах их жизненного цикла;
3. Создание интерфейса для упрощения работы сотрудников с модулем системы и оперативного доступа к документам.

Объектом проекта в соответствии с выбранной темой и поставленной целью является деятельность предприятия по управлению документооборотом.

Предметом является автоматизация деятельности предприятия по управлению документооборотом.

В рамках проекта будут использоваться такие технологии, как объектно-ориентированное программирование для устойчивой архитектуры, язык программирования C# для создания интерфейса, платформа .NET для надежности и масштабируемости, а также база данных PostgreSQL для хранения и обработки информации.

Ожидается, что внедрение данного модуля системы позволит повысить эффективность работы сотрудников, улучшить контроль за документами и сократить время на их обработку.

**1 РАЗРАБОТКА СИСТЕМНОГО ПРОЕКТА**

**1.1 Назначение разработки**

Назначение разработки заключается в создании модуля информационной системы управления документооборотом для автоматизации процессов работы с документами в организации [4]. Основной целью модуля системы является упрощение и ускорение обработки и хранения документов, а также обеспечение контроля за их движением на всех этапах — от создания до архивирования. Система должна повысить прозрачность документооборота, минимизировать риски потери или несанкционированного доступа к информации, а также обеспечить быстрый доступ к необходимым данным.

**1.1.1 Описание деятельности предприятия**

Предприятие, для которого разрабатывается система управления документооборотом, специализируется на предоставлении широкого спектра товаров и услуг на рынке, таких как:

* Поставка оборудования и материалов: предприятие занимается продажей и арендой специализированного оборудования, а также поставкой необходимых материалов для строительной, производственной и сервисной отраслей. К этому направлению относятся логистика, контроль качества поставляемых товаров, обслуживание и сопровождение клиентов;
* Техническое обслуживание и ремонт: предприятие оказывает услуги по техническому обслуживанию и ремонту оборудования для своих клиентов. Это включает проведение плановых и внеплановых ремонтных работ, а также предоставление услуг по модернизации оборудования.

Основная деятельность предприятия включает заключение договоров, выполнение заказов клиентов, управление внутренними процессами и взаимодействие с контрагентами. Важным элементом всех этих процессов является работа с документами, такими как контракты, счета, накладные, служебные записки, отчеты и иные формы документации. Эти документы являются основой для учета обязательств, управления ресурсами и обеспечения прозрачности операций, что делает эффективный документооборот критически важным для стабильной и бесперебойной работы предприятия.

**1.1.2 Анализ требований к проекту**

Проблемы автоматизации документооборота предприятия:

* Рутинные задачи: обработка входящих и исходящих документов, согласование и утверждение, ведение архивов, что занимает много времени и ресурсов сотрудников;
* Отсутствие единой системы: информация о документах хранится в различных местах (бумажные носители, локальные папки, электронные почты), что затрудняет доступ к актуальным данным и их поиск;
* Недостаточная аналитика: неэффективное отслеживание статусов документов, отсутствие инструментов для анализа эффективности документооборота и выявления узких мест в процессах.

Первичные функциональные требования к разрабатываемому программному обеспечению:

* Возможность ведения базы данных документов: создание и редактирование записей о документах, их статусах, сроках исполнения, а также возможность хранения связанных файлов;
* Ведение отчетности: автоматическое формирование отчетов о состоянии документооборота, статистике обработки документов и анализе эффективности работы сотрудников;
* Интеграция с существующими системами: возможность взаимодействия с другими информационными системами предприятия для обеспечения обмена данными и синхронизации процессов.

**1.1.3 Анализ готовых программных решений**

**DocuWare:**

DocuWare – это система управления документами, которая предлагает функции для автоматизации обработки и хранения документов.

Основные возможности ПО:

* Сканирование и загрузка документов: возможность быстро сканировать бумажные документы и загружать их в систему;
* Управление рабочими процессами: автоматизация процессов согласования и обработки документов с использованием шаблонов;
* Хранение и организация документов: централизованное хранение файлов с возможностью организации по папкам и категориям;
* Поиск документов: возможность поиска документов по ключевым словам, однако поиск может быть ограничен по типу документа.

Преимущества DocuWare:

* Удобный интерфейс: интуитивно понятный интерфейс, позволяющий легко находить необходимые функции;
* Безопасность данных: поддержка защиты данных и соблюдения нормативных требований.
* Недостатки DocuWare:
* Ограниченные функции анализа: недостаток встроенных инструментов для глубокого анализа данных о документообороте;
* Высокая стоимость: лицензирование системы может быть дорогостоящим для малых и средних предприятий;
* Необходимость настройки: для полной адаптации к процессам компании может потребоваться значительное время на настройку.

**M-Files:**

M-Files – это система управления документами, ориентированная на автоматизацию управления контентом.

Основные возможности ПО:

* Организация документов по метаданным: документы организуются по метаданным, что упрощает доступ, но требует предварительной настройки;
* Процесс согласования документов: возможность создания рабочих процессов, но система не всегда обеспечивает прозрачный контроль статусов;
* Хранение версий: поддержка хранения разных версий документов, однако механизмы управления версиями могут быть недостаточно гибкими.

Преимущества M-Files:

* Удобный интерфейс: пользовательский интерфейс на основе метаданных помогает упростить поиск документов;
* Масштабируемость: подходит для организаций разного размера и масштаба.

Недостатки M-Files:

* Сложность интеграции: интеграция с другими системами может быть затруднена, что вызывает дополнительные расходы;
* Ограниченная аналитика: функции анализа данных и отчетности недостаточны для полного контроля над документооборотом;
* Стоимость: лицензирование может оказаться дорогим для организаций с ограниченным бюджетом;

Таблица 1 – Соответствие первичным функциональным требованиям

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | DocuWare | M-Files | Разрабатываемое ПО |
| Ведение базы данных документов | - | - | + |
| Поиск и фильтрация документов | + | + | - |
| Ведение отчетности | - | - | + |
| Интеграция с существующими системами | - | - | + |

В таблице 1 знаком «+» отмечен факт соответствия программы функциональному требованию, а знаком «-» обозначены недостатки.

Разрабатываемое ПО будет обладать более широким функционалом, чем DocuWare и M-Files, что позволит ему более эффективно решать задачи управления документооборотом в организации.

**1.2 Требования к функциональным характеристикам**

**1.2.1 Функциональные требования**

На основе исходных функциональных требований были разработаны следующие более подробные функциональные требования к программному обеспечению:

1.Организация хранения данных о документах:

* возможность создания новых записей для документов;
* возможность редактирования уже существующих записей;
* хранение информации о статусах документов и сроках их исполнения;
* возможность прикрепления связанных файлов к записям о документах.

2.Интеграция с существующими системами:

* обеспечение обмена данными с другими информационными системами, используемыми в организации;
* синхронизация процессов между различными системами для оптимизации документооборота.

3.Администрирование модуля системы:

* управление учетными записями пользователей: добавление, редактирование и удаление;
* назначение ролей и прав доступа для различных пользователей;

4.Ведение отчетности и использование шаблонов:

* возможность формирования различных отчётов, таких как отчёты о выполнении задач, статусе документов и сроках выполнения;
* в будущем планируется возможность загрузки шаблонов документов для их автоматического заполнения, что упростит создание типовых документов и отчётов.

**1.2.2 Архитектура разрабатываемого приложения**

Для разработки модуля информационной системы управления документооборотом используется клиент-серверная архитектура, которая обеспечивает разделение задач между клиентом и сервером базы данных.

Архитектура приложения будет включать следующие уровни:

* Клиентский уровень: пользовательский интерфейс приложения разрабатывается с использованием C# и WPF (Windows Presentation Foundation). Этот уровень предоставляет интерфейс для пользователей, позволяющий выполнять основные функции системы, такие как создание, редактирование и удаление документов, а также просмотр информации о задачах и статусах документов. Через клиентский уровень отправляются запросы для работы с данными, результаты которых отображаются пользователю;
* Уровень базы данных: в качестве системы управления базами данных используется PostgreSQL. Она хранит всю необходимую информацию, включая данные о пользователях, документах, задачах, их статусах и историю изменений. База данных выполняет функции по обеспечению надёжности хранения данных, высокопроизводительного доступа к ним и поддержки резервного копирования.

Для наглядной работы всех компонентов приложения представлена диаграмма развертывания (рисунок 1). На ней показаны ключевые компоненты системы и их взаимодействие, включая клиентский компьютер, которые работает с WPF-приложением, и сервер базы данных PostgreSQL.

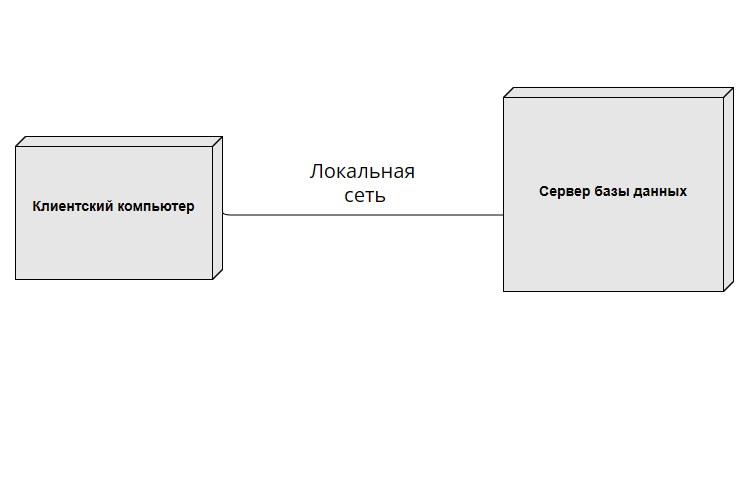


Рисунок 1 – Диаграмма развертывания

На рисунке 1 представлена диаграмма развертывания, демонстрирующая клиент-серверную архитектуру для модуля информационной системы управления документооборотом.

Диаграмма включает два ключевых компонента:

* Клиентский узел — персональный компьютер клиета, на которых установлен WPF-интерфейс, разработанный на C#. Эти узел взаимодействует с сервером базы данных через локальную сеть;
* Сервер базы данных — узел с PostgreSQL, где хранятся все данные документооборота, обеспечивающий прямое взаимодействие с клиентским узлом.

**1.3 Требования к надежности и безопасности**

Разрабатываемый модуль системы управления документооборотом должна обеспечивать высокий уровень надежности и безопасности для предотвращения потери данных и несанкционированного доступа. Для этого необходимо выполнить следующие требования [1]:

1. Надежность:

* Стабильность работы системы — система должна обеспечивать бесперебойную работу при высоких нагрузках и большом количестве одновременно работающих пользователей;
* Защита от сбоев — предусмотрена возможность резервного копирования данных для предотвращения их утраты в случае аппаратных или программных сбоев;
* Контроль целостности данных — необходимо использовать механизмы контроля целостности данных для предотвращения их повреждения или потери в процессе передачи или хранения.

2. Безопасность [14]:

* Аутентификация и авторизация пользователей — система должна предусматривать многоуровневую систему аутентификации и авторизации для разграничения прав доступа к данным и функциям системы;
* Регулярное обновление — необходимо предусмотреть механизм регулярного обновления системы безопасности для защиты от новых уязвимостей и угроз.

**1.4 Требования к составу и параметрам техническим средств**

Для обеспечения работы разрабатываемого программного обеспечения по управлению документооборотом необходимы определенные технические средства, которые должны соответствовать следующим минимальным и рекомендуемым требованиям.

Минимальные требования к клиентским устройствам:

* Операционная система: Windows 10 (x64);
* Процессор: Intel Core i3-6100 или AMD Ryzen 3 1200;
* Оперативная память: 8 ГБ;
* Свободное место на диске: 50 ГБ;
* Графическая карта: Intel HD Graphics 530 или AMD Radeon R7 250;
* Монитор: Разрешение 1366x768;
* Дополнительные устройства: Клавиатура, Мышь, Сканер документов (опционально).

Рекомендуемые требования к клиентским устройствам:

* Операционная система: Windows 11 (x64);
* Процессор: Intel Core i5-8500 или AMD Ryzen 5 3600;
* Оперативная память: 16 ГБ;
* Свободное место на диске: 100 ГБ (SSD предпочтителен);
* Графическая карта: NVIDIA GeForce GTX 1050 или AMD Radeon RX 570;
* Монитор: Разрешение 1920x1080;
* Дополнительные устройства: Клавиатура, Мышь, Сканер, Принтер (опционально).

Минимальные требования к серверу:

* Операционная система: Windows Server 2016 (x64);
* Процессор: Intel Xeon E5-2630 v3 или AMD EPYC 7301;
* Оперативная память: 16 ГБ;
* Свободное место на диске: 150 ГБ;
* Сеть: Gigabit Ethernet;
* Дополнительные устройства: Источник бесперебойного питания.

Рекомендуемые требования к серверу:

* Операционная система: Windows Server 2022 (x64);
* Процессор: Intel Xeon E5-2670 v3 или AMD EPYC 7401;
* Оперативная память: 32 ГБ;
* Свободное место на диске: 250 ГБ (с возможностью расширения);
* Сеть: 10 Gigabit Ethernet;
* Дополнительные устройства: Источник бесперебойного питания.

**1.5 Требования к информационной и программной совместимости**

Для обеспечения совместимости программного обеспечения с операционной системой и базой данных, используемой на предприятии, должны быть соблюдены следующие условия:

* Совместимость с операционными системами: Программное обеспечение должно корректно работать на клиентских устройствах с ОС Windows 10/11, а также на сервере с ОС Windows Server 2016/2022;
* Совместимость с базой данных: Приложение должно быть полностью совместимо с PostgreSQL версии 10 и выше, чтобы обеспечить корректное подключение и стабильное взаимодействие с базой данных;
* Совместимость с аппаратным обеспечением: Устройства должны поддерживать минимальные аппаратные требования, описанные выше, чтобы гарантировать надлежащую производительность. Кроме того, рекомендуется использовать SSD-накопители для улучшения скорости чтения и записи данных;
* Использование .NET платформы: Для работы приложения требуется установленная среда .NET 8, совместимая с архитектурой x64;
* Интеграция с офисными приложениями: Для работы с документами в различных форматах возможно использование библиотек, которые позволяют экспорт данных в PDF, DOCX и XLSX, что обеспечит совместимость с офисным программным обеспечением, таким как Microsoft Office или LibreOffice. Система должна предоставлять отчеты в указанных форматах для удобства использования и дальнейшей обработки данных.

**2 РАЗРАБОТКА ТЕХНИЧЕСКОГО ПРОЕКТА**

**2.1 Анализ требований и определение спецификаций программного**

**обеспечения**

Разрабатываемое программное обеспечение будет предоставлять следующие основные функции [3]:

Управление профилем пользователя:

* «Профиль» для отображения информации о пользователе;
* Возможность редактирования информации о пользователе для актуализации данных.

Управление задачами:

* «Задачи» для просмотра текущих задач, назначенных пользователю;
* Возможность редактирования статуса и данных по задачам;
* Удаление задач, если они неактуальны или завершены.

Управление документами:

* «Документы» для добавления новых документов;
* Возможность скачивания документов для дальнейшей работы или анализа;
* Удаление устаревших или ненужных документов.

Создание новых задач:

* «Создать задачу» для добавления новой задачи с указанием ответственного пользователя и описанием задания.

Отчетность:

* Возможность формирования отчетов о задачах, документах и профилях пользователей;
* Экспорт отчетов в форматы PDF, DOCX и XLSX для дальнейшей работы и анализа;
* Поддержка шаблонов отчётов, которые могут автоматически заполняться данными из системы, чтобы ускорить и упростить работу с отчётностью.

Программное обеспечение также будет отвечать следующим нефункциональным требованиям:

* Разработка интерфейса, удовлетворяющего требованиям удобства и понятности для пользователей;
* Обеспечение безопасности данных и прав доступа пользователей;
* Регулярное резервное копирование базы данных для предотвращения потери данных;
* Возможность масштабирования модуля системы для поддержки увеличения количества пользователей и объема данных.

На рисунке 2 представлена контекстная диаграмма, демонстрирующая работу модуля системы управления документооборотом. Диаграмма отображает взаимодействие пользователей с модулем системой, основные операции, а также входные и выходные данные, участвующие в процессе управления документооборотом



Рисунок 2 – Контекстная диаграмма

Модуль системы управления документооборотом предназначена для автоматизации процесса обработки документов, задач и информации о пользователях в организации.

Входные данные:

* Информация о пользователях (имя пользователя, фото, пароль и роль), включая их учётные записи, права доступа и профили;
* Документы, загружаемые пользователями для дальнейшей их обработки и хранения;
* Данные о задачах, а именно название, описание, исполнитель и статус задачи.

Выходные данные:

* Обновлённые профили пользователей с информацией об изменениях, внесённых в их учётные записи;
* Список задач для каждого сотрудника с возможностью редактирования и отметки статуса;
* Сформированная база данных документов, включающая все файлы и записи о действиях сотрудников.
* Сформированные отчеты по данным документов, задач и пользователей.

Пользователи:

* Сотрудники — это пользователи системы, которые взаимодействуют с документами и задачами;
* Менеджеры — пользователи с расширенными правами, которые могут назначать задачи, отслеживать их выполнение и просматривать отчёты;
* Администраторы — сотрудники, которые поддерживают систему, управляют правами доступа и отвечают за ее корректную работу.

Автоматизация происходит в соответствии с:

* Федеральным законом «О персональных данных» от 27.07.2006 N 152-ФЗ — устанавливает требования;
* ГОСТ Р 6.30-2003 «Унифицированные системы документации. Унифицированная система организационно – распорядительной документации. Требование к оформлению документов»;
* Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 N 149-ФЗ — регулирует защиту информации и требования к ее обработке в информационных системах;
* Нормативно-правовые актами, регулирующие документооборот, являются внутренние инструкции по ведению, хранению и архивированию документов.

Этот модуль системы позволяет сотрудникам компании эффективно управлять документами, задачами и профилями, повышая скорость обработки данных.

На рисунке 3 представлена диаграмма декомпозиции первого уровня, отображающая основные процессы модуля системы управления документооборотом. Диаграмма помогает структурировать последовательность ключевых операций, выполняемых пользователями при работе с документами, задачами и профилями, а также выделить основные функции модуля системы.

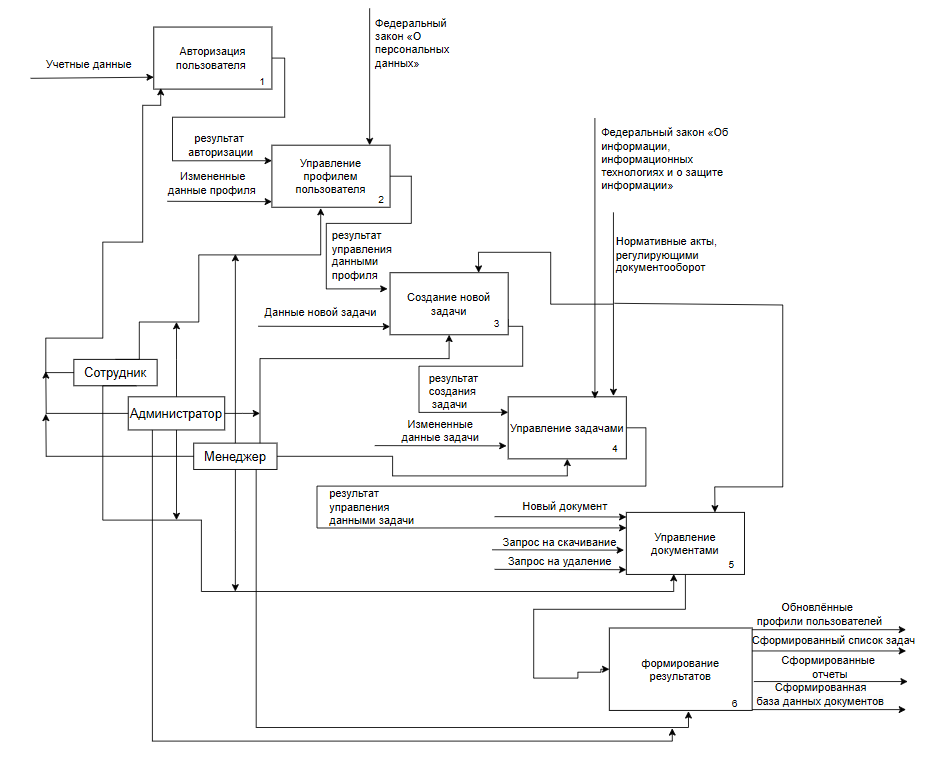


Рисунок 3 – Диаграмма декомпозиции первого уровня

Основные этапы работы включают:

* Управление профилем пользователя: включает просмотр информации о пользователе, а также редактирование профиля для обновления данных.
* Управление задачами: позволяет просматривать текущие задачи, редактировать их статус (например, "На рассмотрении" или "Закрыта"), а также удалять ненужные задачи.
* Управление документами: предусматривает добавление новых документов в систему, загрузку для ознакомления и редактирования, а также удаление устаревших или ненужных файлов.
* Создание новой задачи: обеспечивает создание задачи для выполнения определенных рабочих процессов или контроля документов, что упрощает отслеживание этапов работы.

Эта диаграмма позволяет представить взаимодействие пользователей с модулем системы, формируя наглядное представление о ключевых процессах и логической последовательности операций.

Диаграмма прецедентов, представленная на рисунке 4, иллюстрирует взаимодействие различных типов пользователей с модулем системы управления документооборотом.

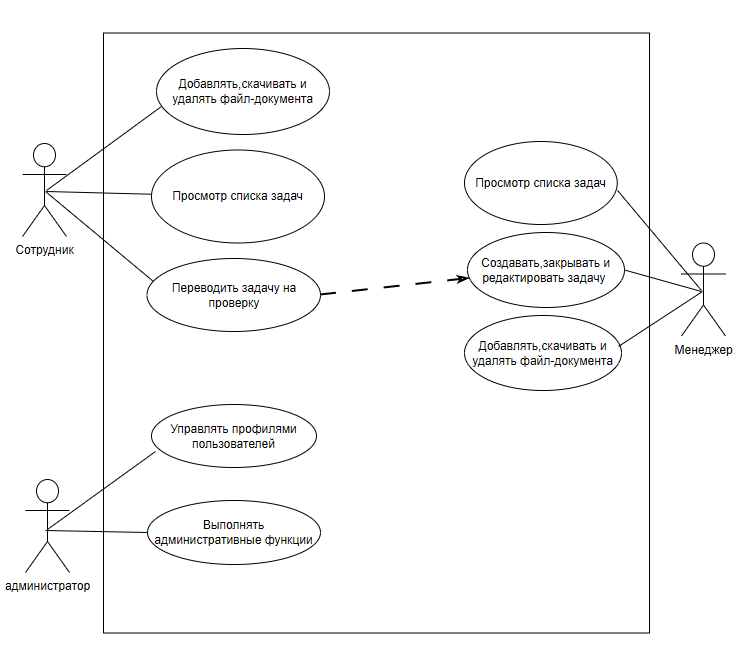


Рисунок 4 – Диаграмма прецендентов

На диаграмме показаны следующие действия [7]:

Сотрудник:

* Получать доступ к списку задач, а также редактировать их статус и удалять задачи;
* Просматривать список документов, загружать и скачивать документы, а также добавлять новые документы.

Администратор:

* Управлять профилями пользователей, добавлять и удалять учетные записи, редактировать информацию о сотрудниках;
* Выполнять административные функции, такие как создание и настройка ролей, а также распределение прав доступа.

Менеджер:

* Просматривать и контролировать выполнение задач, назначенных подчиненным сотрудникам;
* Подтверждать или отклонять выполненные задачи, отправленные на проверку;
* Создавать и назначать новые задачи сотрудникам.
* Просматривать список документов, загружать и скачивать документы, а также добавлять новые документы.

На рисунке 5 представлена диаграмма развертывания, демонстрирующая клиент-серверную архитектуру для модуля информационной системы управления документооборотом.

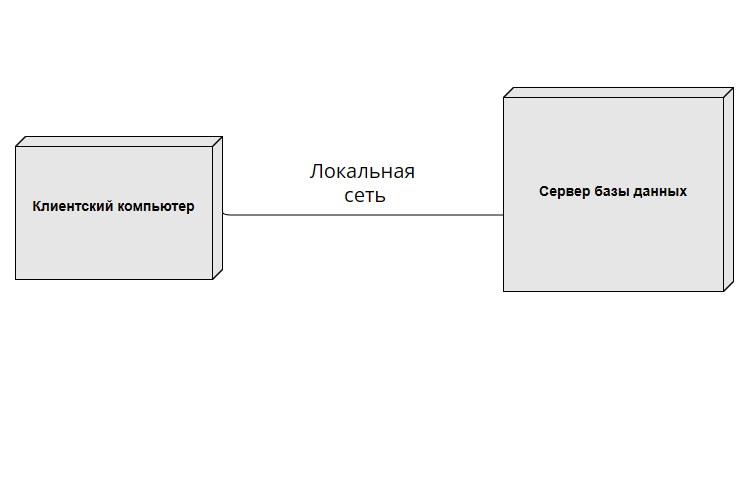


Рисунок 5 – Диаграмма развертывания

Диаграмма включает два ключевых компонента:

* Клиентский узел — персональный компьютер клиета, на которых установлен WPF-интерфейс, разработанный на C#. Эти узел взаимодействует с сервером базы данных через локальную сеть;
* Сервер базы данных — узел с PostgreSQL, где хранятся все данные документооборота, обеспечивающий прямое взаимодействие с клиентским узлом.

**2.2 Проектирование модели данных**

Модель данных модуля системы управления документооборотом включает в себя основные сущности, которые обеспечивают работу с пользователями, документами и задачами, а также определяет связи между этими сущностями для эффективного управления данными.

Описание сущностей и атрибутов

Для реализации функционала системы были спроектированы следующие сущности [16]:

1. Users - таблица, представляющая пользователей системы.

Основные атрибуты:

* Id (INT) — уникальный идентификатор пользователя.
* Username (VARCHAR) — имя пользователя, уникальное.
* Password (VARCHAR) — зашифрованный пароль пользователя.
* Role (VARCHAR) — роль пользователя, определяющая уровень доступа.
* IsAdmin (BOOLEAN) — флаг, указывающий, является ли пользователь администратором.

2. Tasks - таблица, представляющая задачи, назначенные пользователям.

Основные атрибуты:

* Id (INT) — уникальный идентификатор задачи.
* Title (VARCHAR) — название задачи.
* Description (TEXT) — описание задачи.
* Executor (VARCHAR) — исполнитель задачи.
* StartDate и EndDate (TIMESTAMP) — даты начала и завершения задачи.
* UserId (INT) — внешний ключ, ссылающийся на пользователя, создавшего задачу.
* Status (VARCHAR) — текущий статус задачи.

3. Documents - таблица, содержащая информацию о загруженных в систему документах.

Основные атрибуты:

* Id (INT) — уникальный идентификатор документа.
* DocumentName (VARCHAR) — название документа.
* Content (BYTEA) — содержимое документа.
* UploadedBy (INT) — внешний ключ, указывающий на пользователя, загрузившего документ.
* UploadDate (TIMESTAMP) — дата загрузки документа.

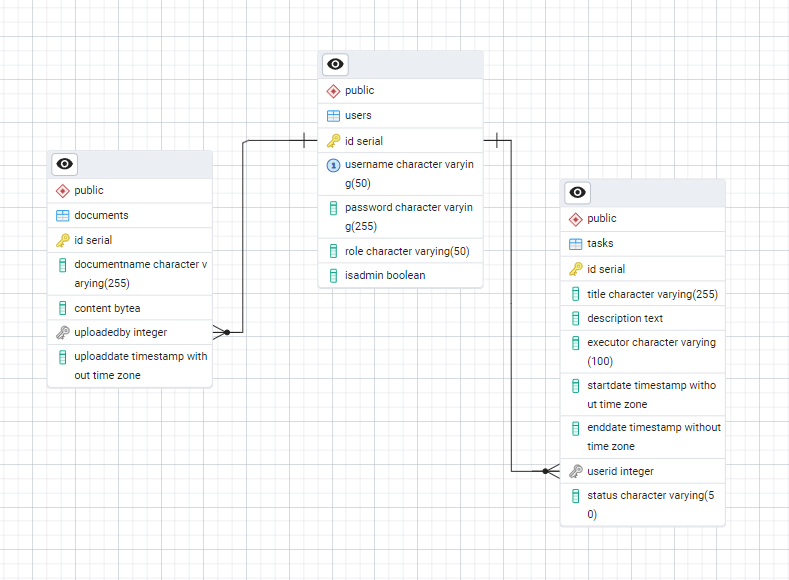


Рисунок 6 – Диаграмма «сущность – связь»

На рисунке 6 представлена диаграмма «сущность – связь» (ER-диаграмма), которая отображает связи между сущностями системы. ER-диаграмма позволяет наглядно представить основные взаимосвязи между таблицами и их атрибутами:

* Таблица Users связана с таблицей Tasks через атрибут UserId, что позволяет отслеживать, какие задачи были назначены и выполнены определенными пользователями.
* Таблица Users также связана с таблицей Documents через атрибут UploadedBy, что позволяет фиксировать, какие пользователи загрузили документы в систему.

Каждый атрибут в таблицах системы имеет свои характеристики, которые позволяют организовать целостное и безопасное хранение данных. Ниже в таблице 2 представлены основные характеристики атрибутов:

Таблица 2 – Характеристика атрибутов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Атрибут | Описание | Тип данных | Длина | Обязательное поле | Индексация | Значение по умолчанию |
| Users | | | | | | |
| Users.Id | Уникальный идентификатор пользователя | INT | Нет значения | Да | Да | Нет |
| Users.Username | Имя пользователя | VARCHAR | 50 | Да | Да | Нет |
| Users.Password | Пароль пользователя | VARCHAR | 255 | Да | Нет | Нет |
| Users.Role | Роль пользователя | VARCHAR | 50 | Нет | Нет | Нет |
| Users.IsAdmin | Администраторский статус | BOOLEAN | Нет значения | Нет | Нет | FALSE |
| Tasks | | | | | | |
| Tasks.Id | Уникальный идентификатор задачи | INT | Нет значения | Да | Да | Нет |
| Tasks.Title | Название задачи | VARCHAR | 255 | Да | Нет | Нет |
| Tasks.Description | Описание задачи | TEXT | Нет значения | Нет | Нет | Нет |
| Tasks.Executor | Исполнитель задачи | VARCHAR | 100 | Нет | Нет | Нет |
| Tasks.StartDate | Дата начала задачи | TIMESTAMP | Нет значения | Нет | Нет | Нет |
| Tasks.EndDate | Дата завершения задачи | TIMESTAMP | Нет значения | Нет | Нет | Нет |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tasks.UserId | ID пользователя, создавшего задачу | INT | Нет значения | Нет | Да | NULL |
| Tasks.Status | Статус задачи | VARCHAR | 50 | Нет | Нет | Нет |
| Documents | | | | | | |
| Documents.Id | Уникальный идентификатор документа | INT | Нет значения | Да | Да | Нет |
| Documents.DocumentName | Название документа | VARCHAR | 255 | Да | Нет | Нет |
| Documents.Content | Содержимое документа | BYTEA | Нет значения | Нет | Нет | Нет |
| Documents.UploadedBy | ID пользователя, загрузившего документ | INT | Нет значения | Нет | Да | NULL |
| Documents.UploadDate | Дата загрузки документа | TIMESTAMP | Нет значения | Нет | Нет | CURRENT\_TIMESTAMP |

Продолжение таблицы 2

**2.3 Детальное проектирование программного обеспечения (конструирование прототипа)**

Целью разработки программного средства является создание модуля системы управления документами, которая позволит автоматизировать процессы документооборота, управлять задачами и предоставлять пользователям быстрый доступ к документам и отчетам. Ниже представлено детальное проектирование программного обеспечения, включая его структурно-функциональную часть, а также описание и представление конструирования прототипа.

Структурно-функциональная часть:

Экран авторизации:

* Цель: обеспечить безопасность доступа в систему.

Функционал:

* Ввод учетных данных: имя пользователя и пароль.
* Проверка учетных данных с использованием базы данных.
* Переход на главный экран при успешной авторизации или вывод ошибки при неправильных данных.

Основные элементы интерфейса:

* Поля для ввода имени пользователя и пароля.
* Кнопка для авторизации.
* Текстовое сообщение об ошибке.

Главный экран приложения:

* Цель: представить основную информацию и обеспечить доступ ко всем функциональным модулям системы.

Функционал:

* Боковая панель навигации с доступом к разделам профиля пользователя, управления задачами, работы с документами и отчетов.
* Основная панель контента для отображения выбранного раздела (документы, задачи, профиль и отчеты).

Основные элементы интерфейса:

* Боковая панель с кнопками для навигации.
* Основная область контента для отображения выбранного функционального модуля.

Управление документами:

* Цель: обеспечить управление документацией — добавление, скачивание, удаление документов.

Функционал:

* Добавление нового документа, включая имя и дату загрузки.
* Скачивание и удаление документов.

Основные элементы интерфейса:

* Список документов.
* Кнопки для добавления, скачивания и удаления документа.

Управление задачами:

* Цель: управлять задачами, назначать их пользователям и отслеживать их статус.

Функционал:

* Создание задачи с указанием названия, описания, исполнителя и статуса.
* Редактирование и удаление задач, изменение статуса, например такие как «В процессе», «На рассмотрении», «Закрыта».

Основные элементы интерфейса:

* Список задач.
* Кнопки для создания, редактирования и удаления задачи.

Отчеты и статистика:

* Цель: предоставить статистическую информацию о документах и задачах.

Функционал:

* Общее количество документов, включая загруженные за последние 30 дней.
* Количество активных задач, завершенных задач и среднее время выполнения задач.
* Отчет по активности пользователей и документам за последний месяц.

Основные элементы интерфейса:

* Панель статистики с показателями по документам и задачам.

Профиль пользователя:

* Цель: обеспечить пользователю возможность просматривать и редактировать личные данные.

Функционал:

* Просмотр информации о пользователе, включая имя, роль и фото профиля.
* Редактирование данных профиля.

Основные элементы интерфейса:

* Форма для редактирования профиля.
* Кнопка для сохранения изменений.

Прототип представляет собой систему с основными элементами интерфейса и функциональными компонентами, ориентированную на предоставление всех ключевых возможностей разрабатываемого продукта.

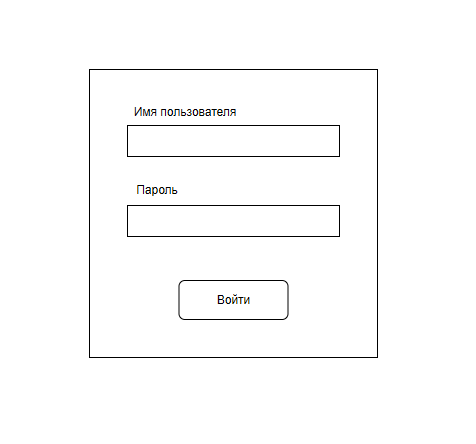


Рисунок 7 – Прототип экрана авторизации

Экран авторизации включает в себя (рисунок 7):

* Поля для ввода имени пользователя и пароля.
* Кнопка для входа.

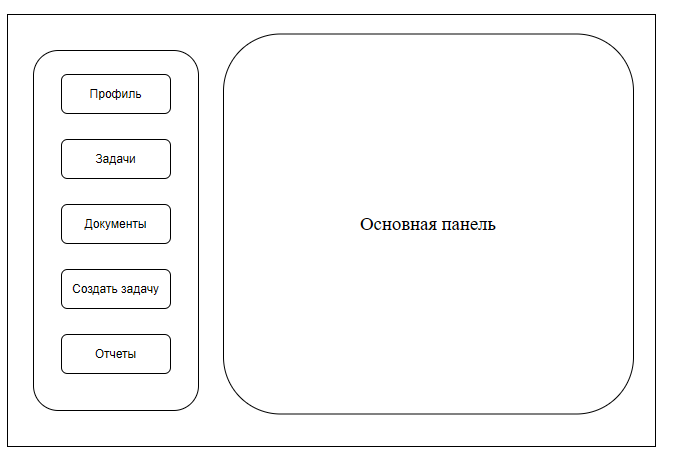


Рисунок 8 – Прототип главного экрана

Главный экран с панелью навигации включает в себя (рисунок 8):

* Боковая панель, содержащая кнопки для быстрого перехода между профилем, задачами, документами и отчетами.
* Основная панель контента для отображения выбранных модулей.

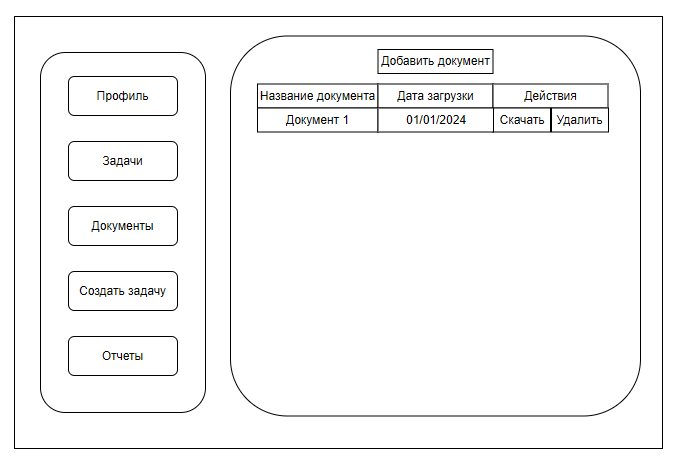


Рисунок 9 – Прототип экрана работы с документами

Экран для работы с документами включает в себя (рисунок 9):

* Список документов.
* Кнопки для добавления, скачивания и удаления документов.

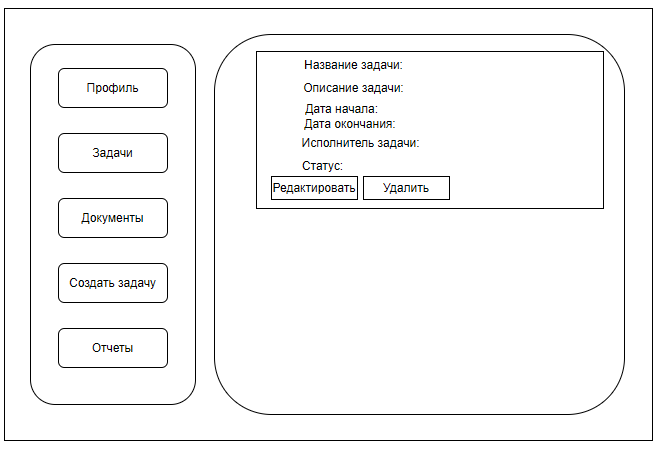


Рисунок 10 – Прототип экрана работы с задачами

Экран для работы с задачами включает в себя (рисунок 10):

* Список задач.
* Кнопки для создания, редактирования и удаления задач.

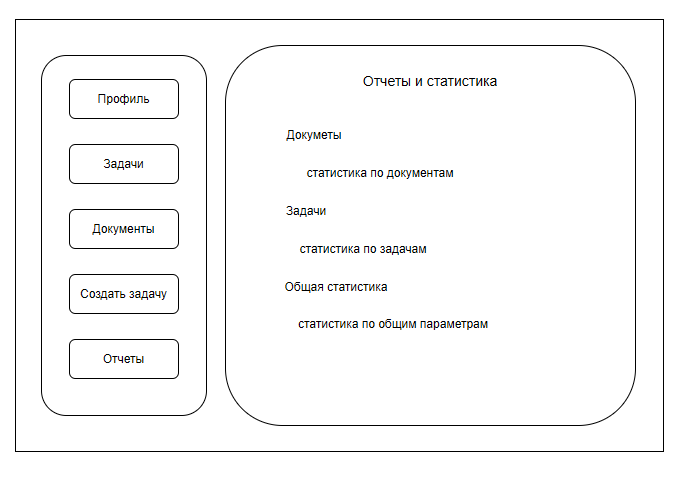


Рисунок 11 – Прототип экрана отчетов

Экран отчетов включает в себя (рисунок 11):

* Показатели статистики по документам, задачам и активности пользователей.

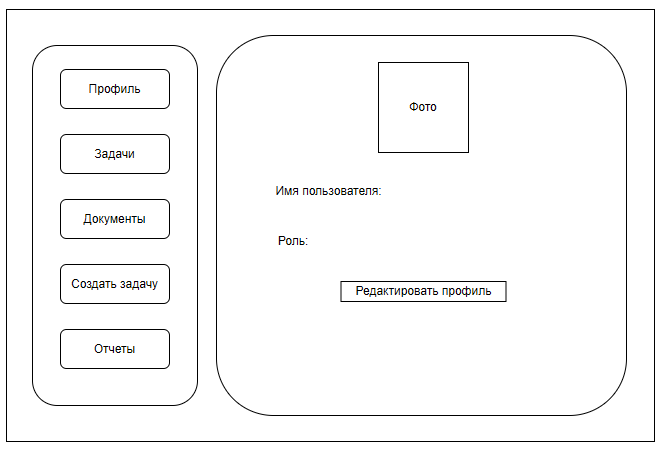


Рисунок 12 – Прототип экрана профиля пользователя

Экран профиля пользователя включает в себя (рисунок 12):

* Поля для отображения и кнопка для редактирования информации пользователя.

Каждый из экранов содержит кнопки, поля ввода, списки и другие элементы интерфейса, которые позволяют пользователям эффективно взаимодействовать с системой и выполнять все предусмотренные функции.

**3 РЕАЛИЗАЦИЯ**

**3.1. Обоснование выбора средств разработки**

Для реализации системы документооборота в проекте были выбраны язык программирования C#, среда разработки Visual Studio 2022 и СУБД PostgreSQL. Данный выбор обусловлен соответствием этих технологий требованиям к программному обеспечению, установленным в организации, и совместимостью с инструментами, уже используемыми заказчиком.

Разработка GUI и возможности интерфейсов: Язык C# и платформа .NET позволяют создавать графические интерфейсы с использованием WPF (Windows Presentation Foundation), обеспечивая поддержку для работы с интерфейсами системы документооборота с множеством элементов управления. Это решение также обеспечивает интеграцию с сервисами, совместимыми с экосистемой .NET, такими как SharePoint и Microsoft Exchange[10], что гарантирует взаимодействие с уже используемыми инструментами заказчика.

Средства разработки и отладки: Visual Studio 2022 предоставляет полный цикл разработки, включая написание, отладку и тестирование кода, что позволяет наладить взаимодействие с базой данных и минимизировать время на настройку и поддержку приложения. Поддержка интеграции с другими продуктами Microsoft, такими как Azure DevOps, обеспечивает синхронизацию кода и отчетов с другими корпоративными инструментами заказчика.

Хранение и управление данными: В проекте используется PostgreSQL как система управления базами данных, поддерживающая операции с данными, что важно для систем документооборота, где требуется безопасное хранение данных.

Проектирование и управление базой данных: Для управления и визуального проектирования базы данных используется pgAdmin, предоставляющий интерфейс для работы с PostgreSQL, выполнения запросов, визуализации структуры базы данных и управления её настройками, что обеспечивает управление данными и синхронизацию с другими базами данных заказчика.

Доступ к базе данных и взаимодействие с объектами: Библиотека Npgsql обеспечивает драйвер для доступа к PostgreSQL из приложений на C#, поддерживая совместимость с .NET. Использование Entity Framework Core [20] для работы с базой данных на уровне объектов позволяет выполнять операции с данными, поддерживая согласованность данных и синхронизацию с корпоративными сервисами, такими как Microsoft Dynamics и SharePoint.

Выбор C#, Visual Studio 2022 и PostgreSQL обусловлен их функциональностью и совместимостью с уже установленными корпоративными инструментами заказчика, что обеспечивает выполнение целевых задач проекта и взаимодействие с такими продуктами, как SharePoint, Microsoft Exchange, Azure DevOps и Microsoft Dynamics.

**3.2 Описание основных программных модулей**

Описание рабочих форматов:

* .cs — файлы исходного кода на языке C#, в которых реализована основная логика приложения, включая обработку событий интерфейса, взаимодействие с базой данных и обработку данных.
* .xaml — файлы разметки, используемые в WPF для создания интерфейсов приложения. Они содержат описание визуальных элементов, их стили и привязки данных, что позволяет разделить логику интерфейса и бизнес-логику.

Основные модули и функциональные экраны приложения:

* MainWindow — экран авторизации, позволяющий пользователям входить в систему, вводя свои учетные данные. Доступ к функционалу системы предоставляется только после успешной аутентификации.
* MainPage — основное окно приложения, содержащее боковую панель с кнопками для перехода к различным функциям системы, включая управление документами и задачами.
* UserProfileView — модуль профиля пользователя, отображающий информацию о текущем пользователе. Здесь можно просматривать и обновлять информацию профиля, включая изображение пользователя.
* DocumentsView — модуль для управления документами, включающий функционал добавления, редактирования, удаления и поиска документов. Экран предоставляет доступ к списку документов и возможностям их фильтрации по дате и категориям.
* TasksView — модуль управления задачами, предназначенный для создания, редактирования и удаления задач, а также для изменения их статусов (например, «В процессе», «Закрыта»).
* CreateTaskWindow — экран создания и редактирования задач, обеспечивающий возможность назначения задач пользователям и установку их статусов.
* ReportWindow — экран отчетов, где отображаются статистические данные, такие как количество загруженных документов, активные задачи и среднее время выполнения задач, а также другая информация по работе системы.
* DatabaseHelper — вспомогательный модуль для взаимодействия с базой данных PostgreSQL. Содержит методы для выполнения операций CRUD (создание, чтение, обновление, удаление) и получения статистики.

Описание структуры экранов представлено в таблице 3:

Таблица 3 – Описание структуры экранов

|  |  |  |
| --- | --- | --- |
| Название | Описание | Экран верхней границы |
| MainWindow | Экран авторизации |  |
| MainPage | Основное окно после авторизации | AuthorizationWindow |
| UserProfileView | Экран профиля пользователя | MainPage |
| DocumentsView | Экран управления документами | MainPage |
| TasksView | Экран управления задачами | MainPage |
| CreateTaskWindow | Экран создания и редактирования задач | TaskManagement |
| ReportWindow | Экран отображения статистики | MainPage |

**4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА**

Функциональное тестирования программного продукта представлено в таблице 4.

Таблица 4 – Функциональное тестирование

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Получаемый результат |
| 1. Ввести корректное имя пользователя и пароль, нажать кнопку «Войти» | Открывается главное окно программного продукта | Успешный вход, открывается главное окно |
| 2. Ввести некорректное имя пользователя или пароль, нажать кнопку «Войти» | Появляется сообщение "Неправильное имя пользователя или пароль" | Сообщение об ошибке отображается |
| 3. Нажать кнопку «Создать задачу» для пользователей с данными правами | Открывается окно создания задачи | Открывается окно создания задачи |
| 4. Нажать кнопку «Создать задачу» для пользователя без данных прав | Появляется сообщение "У вас нет прав для создания задачи" | Сообщение об ошибке отображается |
| 5. Нажать кнопку «Документы» на главной странице | Отображается список документов | Список документов отображается |
| 6. Нажать кнопку «Задачи» на главной странице | Отображается список задач | Список задач отображается |
| 7. Нажать кнопку «Профиль» | Отображается информация профиля текущего пользователя | Информация профиля отображается |
| 8. Нажать кнопку «Отчет» на главной странице | Отображается окно с отчетом по статистике документов и задач | Окно с отчетом отображается |
| 9. Нажать кнопку «Добавить документ» в списке документов | Появляется окно выбора для сохранения файла | Окно сохранения файла открывается |
| 10. Нажать кнопку «Редактировать профиль», изменить данные и сохранить | Появляется сообщение об успешном обновлении профиля | Сообщение отображается, профиль обновлен |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| 11. Нажать кнопку "Выход" в профиле пользователя | Приложение закрывается | Приложение закрывается |

Критерии на надежность и правильность работы разрабатываемого программного средства описаны в таблице 5.

Таблица 5 – Критерии на надежность и правильность работы разрабатываемого программного средства

|  |  |
| --- | --- |
| Критерий | Наличие/Описание |
| Правильность работы с данными (сохранение и извлечение) | +/Гарантированная корректность сохранения данных и их извлечения из базы данных. |
| Надежность обработки ошибок | +/Вывод информативных сообщений об ошибках в случае сбоев или некорректных действий пользователя. |
| Корректность отображения данных в интерфейсе | +/Правильное отображение всех данных в таблицах и элементах интерфейса |
| Унификация интерфейса (цвет, шрифт, размер) | +/Согласованный стиль всех элементов интерфейса, одинаковые шрифты и цвета для улучшения восприятия. |
| Выделение выбранного элемента | +/Правильное выделение и визуализация активных элементов интерфейса, таких как выбранные строки. |
| Производительность системы | +/Быстрая и стабильная работа программы, минимальное время отклика и высокая скорость обработки данных. |

ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта был разработан модуль системы управления документооборотом для компании, которая автоматизирует процессы работы с документами, обеспечивая их создание, согласование, хранение и отслеживание на каждом этапе жизненного цикла. В системе реализованы функции управления статусами документов, упрощенный интерфейс для пользователей, а также интеграция с базой данных для хранения и обработки информации.

Все поставленные задачи были успешно выполнены: проведен анализ предметной области, разработана структура системы, реализован функционал работы с документами, создан графический интерфейс для взаимодействия с пользователем, а также проведено тестирование программного обеспечения.

Ожидается, что внедрение данной системы управления документооборотом позволит значительно повысить эффективность работы сотрудников, улучшить контроль за документами и сократить время на их обработку. В будущем систему можно будет совершенствовать и расширять, добавляя новые функции для повышения производительности, улучшения пользовательского опыта и обеспечения более гибкого контроля над документами. Также планируется расширить возможности отчетности, добавить более детальную аналитику и улучшить функции безопасности данных.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 N 149-ФЗ (последняя редакция): [принят Государственной Думой 8 июля 2006 года: одобрен Советом Федерации 14 июля 2006 года]. – Текст: электронный // КонсультантПлюс: справочно-правовая система: [сайт]. – URL: http://www.consultant.ru/document/cons\_doc\_LAW\_61798/ (дата обращения: 05.11.2024).
2. Федеральный закон «О персональных данных» от 27.07.2006 N 152-ФЗ (последняя редакция): [принят Государственной Думой 8 июля 2006 года: одобрен Советом Федерации 14 июля 2006 года]. – Текст: электронный // КонсультантПлюс: справочно-правовая система: [сайт]. – URL: http://www.consultant.ru/document/cons\_doc\_LAW\_61801/ (дата обращения: 05.11.2024).
3. ГОСТ 34.936-91 (ИСО 10039-91). Информационная технология. Локальные вычислительные сети. Определение услуг уровня управления доступом к среде: государственный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Постановлением Комитета стандартизации и метрологии СССР от 29.12.91 N 2325: дата введения 1992-07-01 / подготовлен Министерством радиопромышленности СССР. – Текст: электронный // КОНСОРЦИУМ «КОДЕКС»: электронный фонд правовой и нормативно-технической документации: [сайт]. – URL: http://docs.cntd.ru/document/1200021831?section=text (дата обращения: 05.11.2024).
4. СТО 60-02.2.3-2018. Стандарт организации. Общие требования к оформлению и изложению документов учебной деятельности обучающихся: утвержден приказом ректора университета от 24.01.2018 № 36: дата введения 2018-02-01. – Архангельск: САФУ, 2018. – 142 с. – Текст: электронный // САФУ: [официальный сайт] / Северный (Арктический) федеральный университет имени М. В. Ломоносова. – URL: https://narfu.ru/upload/iblock/9a7/STO\_60\_02.2.3\_227\_018\_Obshchie\_trebovaniya\_k\_oformleniyu\_i\_izlozheniyu\_dokumentov\_uchebnoy\_deyatelnosti\_obuchayushchikhsya.pdf (дата обращения: 10.11.2024).
5. Intuit.ru [Электронный ресурс]. Точка доступа: https://intuit.ru/studies/courses/2195/55/lecture/1630?page=2 (Дата обращения: 07.11.2024) – Текст: электронный.
6. Habr.com [Электронный ресурс]. Точка доступа: https://habr.com/ru/articles/514364/ (Дата обращения: 07.11.2024) – Текст: электронный.
7. TestEngineer.ru [Электронный ресурс]. Точка доступа: https://testengineer.ru/use-case-diagram/ (Дата обращения: 07.11.2024) – Текст: электронный.
8. Mixtelecom.ru [Электронный ресурс]. Точка доступа: https://mixtelecom.ru/blog/top-10-samyh-populyarnyh-sistem-upravleniya-bazamidannyh (Дата обращения: 07.11.2024) – Текст: электронный.
9. Itelon.ru [Электронный ресурс]. Точка доступа: https://itelon.ru/blog/arkhitektura-klient-server/ (Дата обращения: 07.11.2024) – Текст: электронный.
10. Microsoft.com [Электронный ресурс]. Точка доступа: https://learn.microsoft.com/ruru/dotnet/desktop/winforms/overview/?view=netdesktop-8.0 (Дата обращения: 07.11.2024) – Текст: электронный.
11. Segezha-group.com [Электронный ресурс]. Точка доступа: https://segezha-group.com/about/factory/onezhskiy\_ldk/ (Дата обращения: 07.11.2024) – Текст: электронный.
12. Zaochnik.ru [Электронный ресурс]. Точка доступа: https://zaochnik.ru/blog/kak-pisat-vvedenie-v-kursovoj-rabote-chtoby-ne-popastvprosak/ (Дата обращения: 07.11.2024) – Текст: электронный.
13. Solutions.1c.ru [Электронный ресурс]. Точка доступа: https://solutions.1c.ru/catalog/woodwork-erp (Дата обращения: 07.11.2024) – Текст: электронный.
14. Scanner.narod.ru [Электронный ресурс]. Точка доступа: https://scanner.narod.ru/link/Safe/bezopasnost\_programmnogo\_obespecheniya1.htm (Дата обращения: 07.11.2024) – Текст: электронный.
15. Studfile.net [Электронный ресурс]. Точка доступа: https://studfile.net/preview/16375477/page:2/ (Дата обращения: 07.11.2024) – Текст: электронный.
16. Gb.ru [Электронный ресурс]. Точка доступа: https://gb.ru/blog/atributy-dannyh/ (Дата обращения: 07.11.2024) – Текст: электронный.
17. Practicum.yandex.ru [Электронный ресурс]. Точка доступа: https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/ (Дата обращения: 07.11.2024) – Текст: электронный.
18. Uchet-jkh.ru [Электронный ресурс]. Точка доступа: https://uchetjkh.ru/i/pocemu-c-tak-nazyvaetsya/ (Дата обращения: 07.11.2024) – Текст: электронный.
19. Online-convert.com [Электронный ресурс]. Точка доступа: https://www.online-convert.com/ru/file-format/cs (Дата обращения: 07.11.2024) – Текст: электронный.
20. Metanit.com [Электронный ресурс]. Точка доступа: https://metanit.com/sharp/entityframework/1.1.php (Дата обращения: 07.11.2024) – Текст: электронный.

ПРИЛОЖЕНИЕ А

(обязательное)

Техническое задание

|  |  |
| --- | --- |
| СОГЛАСОВАНО  должность руководителя и наименование организации/предприятия - заказчик  подпись, инициалы, фамилия  дата | СОГЛАСОВАНО  должность - разработчик подпись, инициалы, фамилия подпись, инициалы, фамилия подпись, инициалы, фамилия  дата |

наименование и условное обозначение изделия

ТЕХНИЧЕСКОЕ ЗАДАНИЕ ТЗ-ХХХХХХХХХХ-ХХ

шифр работы, год утверждения

УТВЕРЖДАЮ

должность руководителя и наименование

утверждающей организации

подпись, инициалы, фамилия

дата

Архангельск 2024 г.

ОГЛАВЛЕНИЕ

1. Общие сведения 50
2. Назначение и цели создания и развития системы 51
3. Характеристика объектов автоматизации 51
4. Требования к системе 51
5. Состав и содержание работ по созданию системы 53
6. Порядок контроля и приёмки 53
7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 53
8. Требования к документированию 53
9. Источники разработки 53

10 Выводы 54

**1. Общие сведения**

а) Полное наименование системы: Автоматизированная система документооборота для управления внутренней документацией компании.

б) Краткое наименование системы: АС «Документооборот».

в) Основания для разработки: Основанием для разработки системы является договор №123/24 от 10 ноября 2024 года между компанией «АйтиГум» и заказчиком — организацией ООО «ДокПро».

г) Наименование организации-заказчика: ООО «ДокПро».

д) Фактический адрес организации-заказчика: г. Москва, ул. Технологий, дом 5.

е) Основные виды деятельности организации-заказчика: ООО «ДокПро» занимается заключением договоров, выполнением заказов клиентов, управление внутренними процессами и взаимодействие с контрагентами.

ж) Плановые сроки начала и окончания работы:

Дата начала работ: 19 Октября 2024 г.

Окончание работ: 15 Ноября 2024 г.

з) Порядок оформления и предъявления заказчику результатов работы:

На каждом этапе разработки результаты работы будут предоставляться заказчику для проверки и утверждения, включая оценку промежуточных результатов, внесение изменений в ТЗ и документирование всех изменений в ходе разработки.

**2. Назначение и цели разработки**

Назначение: Система предназначена для автоматизации процессов управления документами в организации, система должна повысить прозрачность документооборота, минимизировать риски утраты документов или несанкционированного доступа к ним, а также обеспечить быстрый доступ к необходимой информации для пользователей.

Цели:

Снижение временных затрат на управление документами.

Обеспечение безопасности и контроля доступа к данным.

Ускорение обработки документов.

**3. Характеристика объектов автоматизации**

Основные объекты автоматизации:

Отдел управления документацией: для систематизации и структурированного хранения документов.

Отдел контроля доступа: для настройки прав доступа пользователей к документам и отчетам.

Административный отдел: для выполнения задач по анализу использования и хранения документации.

**4. Требования к системе**

**4.1 Функциональные характеристики**

Организация хранения данных о документах:

Возможность создания новых записей для документов.

Возможность редактирования уже существующих записей.

Хранение информации о статусах документов и сроках их исполнения.

Возможность прикрепления связанных файлов к записям о документах.

Интеграция с существующими системами:

Обеспечение обмена данными с другими информационными системами, используемыми в организации.

Синхронизация процессов между различными системами для оптимизации документооборота.

Администрирование модуля системы:

Управление учетными записями пользователей: добавление, редактирование и удаление.

Назначение ролей и прав доступа для различных пользователей.

Ведение отчетности и использование шаблонов:

Возможность формирования различных отчётов, таких как отчёты о выполнении задач, статусе документов и сроках выполнения.

В будущем планируется возможность загрузки шаблонов документов для их автоматического заполнения, что упростит создание типовых документов и отчётов.

**4.2 Надежность и безопасность**

Защита данных от несанкционированного доступа.

Регулярное обновление.

Контроль целостности данных

Обеспечение непрерывной работы системы при одновременном доступе пользователей.

**4.3 Технические требования**

Рабочие станции: Windows 10 и выше, процессор Intel Core i3-6100, 8 ГБ оперативной памяти, SSD объемом 100 ГБ.

Сервер: Windows Server 2016 и выше, процессор Intel Xeon E5-2670 v3, 32 ГБ оперативной памяти, cвободное место на диске: 150 ГБ.

**4.4 Программная совместимость**

Совместимость с операционными системами Windows 10 и выше. Поддержка СУБД PostgreSQL версии 10 и выше, совместимость с продуктами .NET.

**4.5 Нефункциональные требования**

Безопасность данных: Защита данных с использованием многоуровневой системы аутентификации.

Резервное копирование: Резервное копирование данных каждые 24 часа.

Масштабируемость: Возможность расширения системы для поддержки увеличения количества пользователей и объема данных.

**5. Состав и содержание работ по созданию системы**

Сбор требований и анализ текущих процессов — 3 дня.

Разработка технического задания — 5 дней.

Проектирование интерфейса и структуры базы данных — 7 дней.

Разработка алгоритмов и программирование — 15 дней.

Тестирование и отладка — 7 дней.

Подготовка документации для пользователей — 5 дней.

Внедрение системы и обучение пользователей — 3 дня.

Поддержка и сопровождение системы — по согласованию с заказчиком.

**6. Порядок контроля и приемки**

Предварительные испытания: Проверка системы на предмет соответствия ТЗ и выявление ошибок.

Приемочные испытания: Завершение всех тестов, включая проверку на готовность к эксплуатации и финальное согласование.

**7. Требования к подготовке объекта автоматизации к вводу системы в действие**

Подготовка оборудования и программного обеспечения.

Организация сетевого взаимодействия.

Подготовка и обучение персонала заказчика для работы с системой.

**8. Требования к документированию**

Программная документация включает:

Техническое задание.

Руководство пользователя.

Расчетно-пояснительная записка.

**9. Источники разработки**

Федеральный закон от 27.07.2006 №149-ФЗ «Об информации, информационных технологиях и о защите информации».

ГОСТ 34.602-89 «Комплекс стандартов на автоматизированные системы».

**10. Выводы**

Настоящее техническое задание определяет структуру и функциональные требования к разрабатываемой системе документооборота, а также порядок выполнения работ, контроля и приемки.

ПРИЛОЖЕНИЕ Б

(обязательное)

Руководство пользователя

ПРОГРАММНОЕ СРЕДСТВО

«Автоматизация работы документооборота»

Руководство пользователя

Листов 14

2024

ОГЛАВЛЕНИЕ

Введение 50

1 Системные требования 51

1.1 Требования к составу и параметрам технических средств………………………………58

1.2 Требования к функциональным характеристикам……………………………………….58

2 Инструкция по работе с программным обеспечением………………………………………..51

2.1 Запуск и вход в систему……………………………………………….………………….60

2.2 Главный экран приложения ……………………………………..61

2.3 Управление документами ……………………………………..62

2.4 Управление задачами ……………………………………..66

2.5 Отчеты и статистика ……………………………………..69

2.6 Профиль пользователя ……………………………………..70

**ВВЕДЕНИЕ**

Данное руководство пользователя предназначено для работы с программным обеспечением, разработанным для автоматизации процессов документооборота в организации. Система создана с целью оптимизации работы с документами, ускорения их обработки, хранения и контроля за движением на всех этапах — от создания до архивирования. В условиях роста объемов документации и требований к ее безопасности эффективное управление документооборотом становится неотъемлемой частью успешной деятельности любой организации.

Основной целью данного программного средства является упрощение и автоматизация процессов работы с документами, что позволяет снизить риски потери информации, улучшить контроль за сроками исполнения и повысить прозрачность всех этапов документооборота. Программное обеспечение предоставляет удобный инструмент для пользователей, позволяя повысить скорость обработки документов и улучшить взаимодействие между отделами.

Система включает следующие ключевые функциональные модули:

Организация хранения данных о документах — для создания, редактирования и хранения записей о документах, их статусах и сроках исполнения, а также для прикрепления связанных файлов.

Интеграция с существующими системами — для обмена данными с другими информационными системами организации и синхронизации процессов для повышения эффективности документооборота.

Администрирование модуля системы — для управления учетными записями пользователей, назначения ролей и прав доступа, а также для обеспечения безопасности данных.

Ведение отчетности и использование шаблонов — для автоматического формирования отчетов о выполнении задач и статусах документов, а также для планируемого использования шаблонов документов для их автоматического заполнения.

Это руководство подробно описывает функциональные возможности программы, предоставляет пошаговые инструкции по выполнению основных операций, а также примеры использования интерфейса для удобства пользователей.

**1 СИСТЕМНЫЕ ТРЕБОВАНИЯ**

**1.1 Требования к составу и параметрам технических средств**

Для работы системы необходимы следующие аппаратные средства:

Рабочие станции (пользователи системы):

Операционная система: Windows 10 (x64) или Windows 11 (x64).

Процессор: Intel Core i3-6100 или AMD Ryzen 3 1200 (минимум), Intel Core i5-8500 или AMD Ryzen 5 3600 (рекомендуется).

Оперативная память: 8 ГБ (минимум), 16 ГБ (рекомендуется).

Жесткий диск: 50 ГБ свободного места (минимум), 100 ГБ (рекомендуется, SSD предпочтителен).

Графическая карта: Intel HD Graphics 530 или AMD Radeon R7 250 (минимум), NVIDIA GeForce GTX 1050 или AMD Radeon RX 570 (рекомендуется).

Монитор: Разрешение 1366x768 (минимум), 1920x1080 (рекомендуется).

Дополнительные устройства: Клавиатура, Мышь, Сканер документов (опционально).

Серверные компоненты:

Операционная система: Windows Server 2016 (x64).

Процессор: Intel Xeon E5-2630 v3 или AMD EPYC 7301.

Оперативная память: 16 ГБ.

Свободное место на диске: 150 ГБ.

Сетевое подключение: Gigabit Ethernet.

Дополнительные устройства: Источник бесперебойного питания.

**1.2 Требования к функциональным характеристикам**

Система предназначена для автоматизации работы с документами и включает следующие функциональные характеристики:

Организация хранения данных о документах:

Возможность создания новых записей для документов.

Возможность редактирования уже существующих записей.

Хранение информации о статусах документов и сроках их исполнения.

Возможность прикрепления связанных файлов к записям о документах.

Интеграция с существующими системами:

Обеспечение обмена данными с другими информационными системами, используемыми в организации.

Синхронизация процессов между различными системами для оптимизации документооборота.

Администрирование модуля системы:

Управление учетными записями пользователей: добавление, редактирование и удаление.

Назначение ролей и прав доступа для различных пользователей.

Ведение отчетности и использование шаблонов:

Возможность формирования различных отчетов, таких как отчеты о выполнении задач, статусе документов и сроках выполнения.

В будущем планируется возможность загрузки шаблонов документов для их автоматического заполнения, что упростит создание типовых документов и отчетов.

**2** **ИНСТРУКЦИЯ ПО РАБОТЕ С ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ**

**2.1 Запуск и вход в систему**

Для начала работы с системой необходимо пройти процедуру авторизации. В окне авторизации (Рисунок 1) пользователю требуется ввести логин и пароль. Если введенные данные совпадают с теми, что содержатся в базе данных, то система откроет доступ к главному экрану. В случае ошибочного ввода будет выведено сообщение об ошибке (Рисунок 2). После успешной авторизации пользователь попадет на главный экран системы, с которого можно перейти к необходимым функциональным модулям.

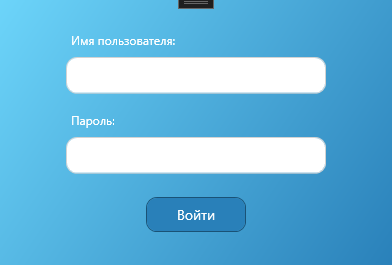


Рисунок 1 – Окно авторизации

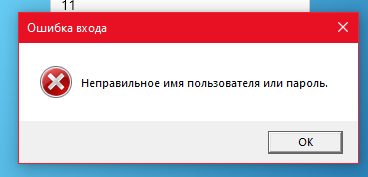


Рисунок 2 – Сообщение об ошибке

**2.2 Главный экран приложения**

На главном экране (Рисунок 3) пользователю доступна боковая панель навигации, через которую можно перейти к различным разделам: профиль пользователя, задачи, документы, отчеты. Основная панель контента отображает выбранный раздел. Это позволяет быстро и удобно перемещаться между функциональными модулями.

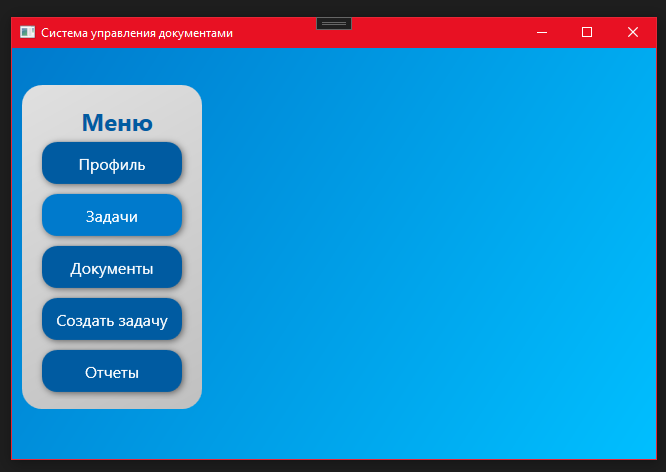


Рисунок 3 – Окно главного экрана приложении

**2.3 Управление документами**

Для работы с документами предусмотрена специальная страница (Рисунок 4), где пользователь может просматривать список всех загруженных документов. Здесь же можно добавлять новый документ, скачивать и удалять существующие. Для добавления нового документа необходимо нажать на кнопку «Добавить» и указать необходимые данные (Рисунок 5): название документа, выбрать необходимый файл документа, дата загрузки будет заполнена автоматически. Документ появится в списке, и пользователю будет доступна возможность скачать(Рисунок 6) или удалить(Рисунок 7) его.

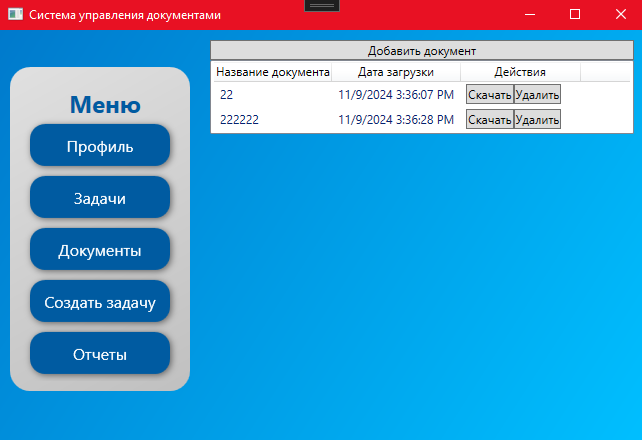


Рисунок 4 – Окно управления документами

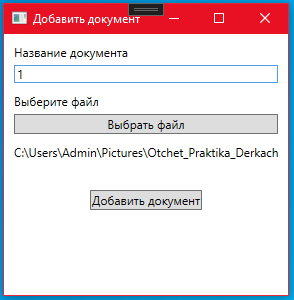


Рисунок 5 – Окно добавления нового документа

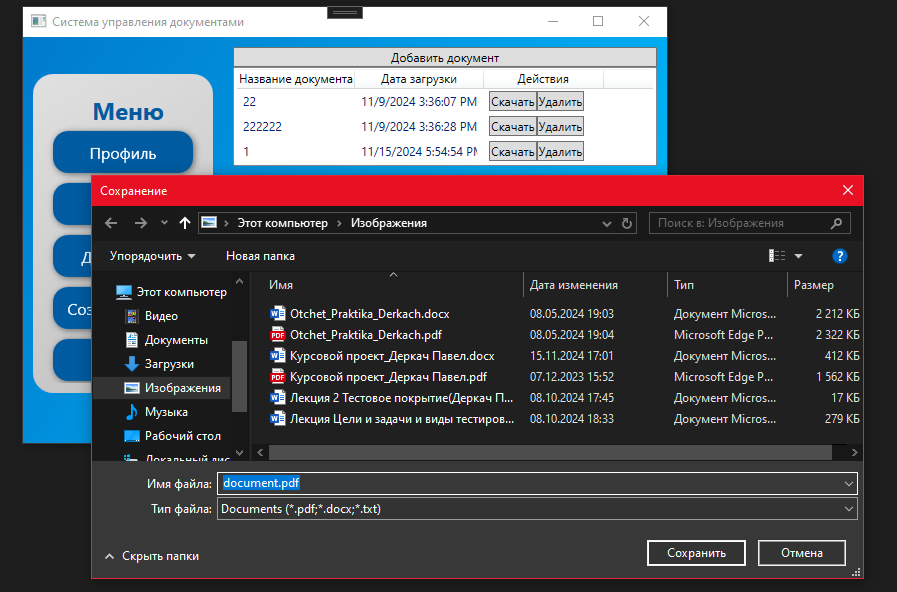


Рисунок 6 – Обновленный список документов и демонстрация функции скачивания

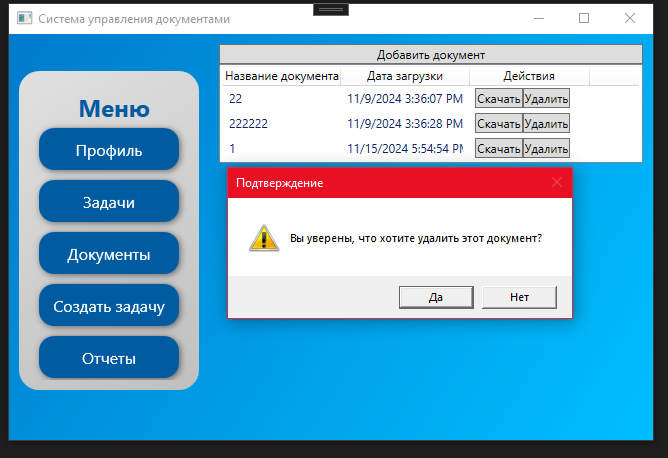


Рисунок 7 – Окно с подтверждения удаления документа

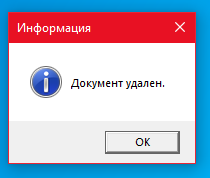


Рисунок 8 – Окно сообщения об удалении документа



Рисунок 9 – Окно управления документами с обновленным списком

**2.4 Управление задачами**

Страница «Задачи» (Рисунок 10) предназначена для управления задачами в системе. Здесь пользователь может создавать новые задачи, редактировать существующие, а также изменять их статус. Для создания задачи необходимо указать название, описание, назначить исполнителя и установить статус задачи (например, «В процессе», «На рассмотрении», «Закрыта») (Рисунок 11). Пользователь также может редактировать задачи (Рисунок 11) и удалять их при необходимости.

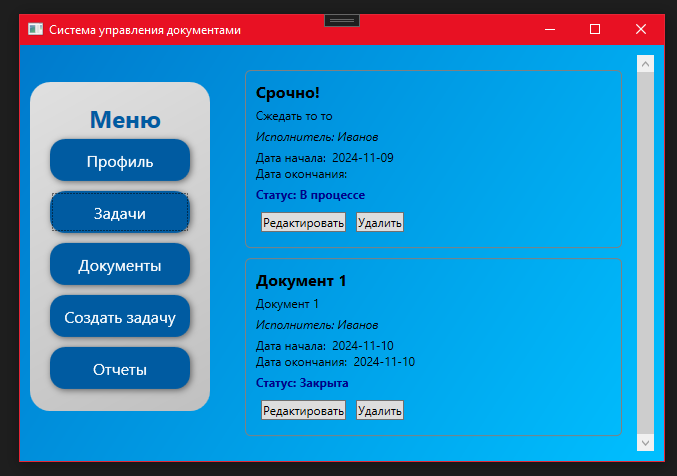


Рисунок 10 – Окно управления задачами

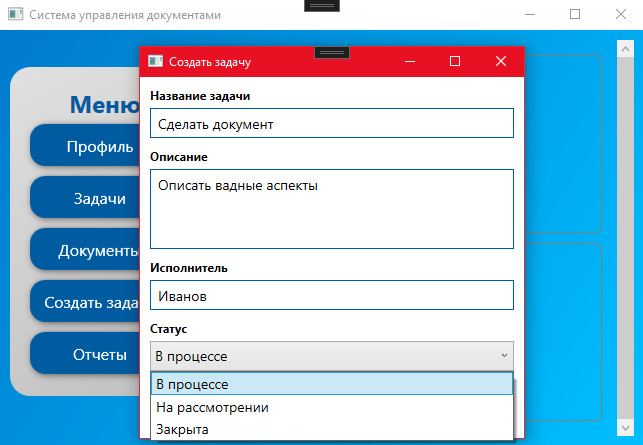


Рисунок 11 – Окно добавления и редактирования задач

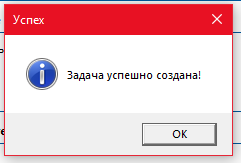


Рисунок 12 – Окно сообщения о созданной задачи

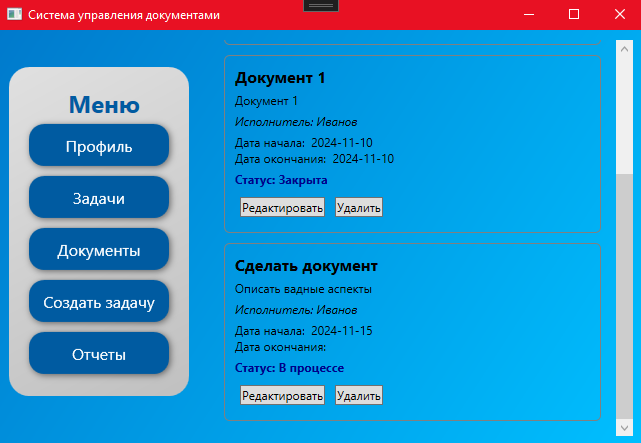


Рисунок 13 – Окно управления задач

**2.5 Отчеты и статистика**

На странице «Отчеты» (Рисунок 14) отображаются статистические данные по документам, задачам и активности пользователей. Пользователь может увидеть общее количество загруженных документов, количество завершенных задач, а также другие показатели эффективности. Эти данные помогут пользователю отслеживать текущую активность и принимать более информированные решения.



Рисунок 13 – Окно отчетов и статистики

**2.6 Профиль пользователя**

Страница профиля (Рисунок 14) позволяет пользователю просматривать и редактировать свою личную информацию. На этой странице отображаются данные пользователя, такие как имя, роль и фото профиля. Пользователь может внести изменения в свои данные(Рисунок 15), а затем сохранить их, а также выйти из системы нажав на кнопку «Выйти».

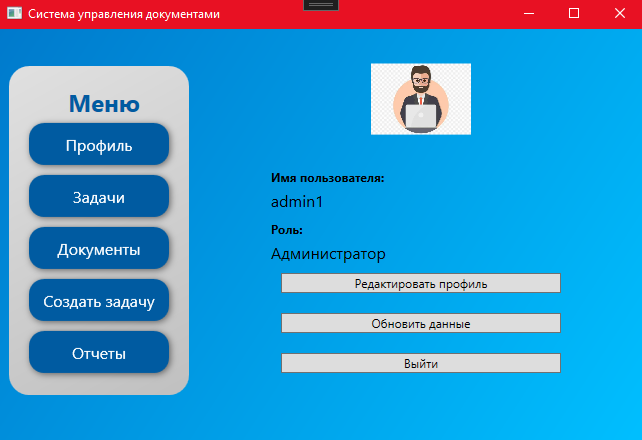


Рисунок 14 – Окно профиля пользователя

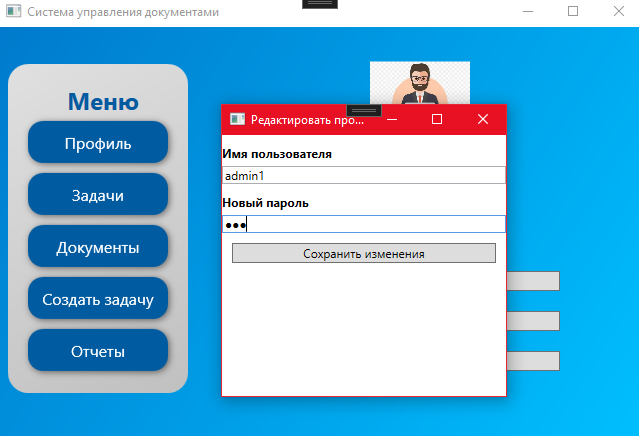


Рисунок 15 – Окно редактирования данных пользователя

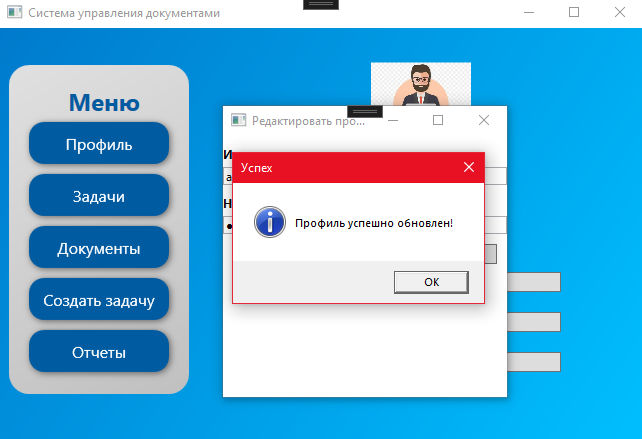


Рисунок 16 – Окно сообщения об успешном обновлении профиля

ПРИЛОЖЕНИЕ В

(обязательное)

Листинг кода

Листинг кода 1 – MainWindow

using System.Windows;

using System.Windows.Controls;

namespace AuthApp

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void LoginButton\_Click(object sender, RoutedEventArgs e)

{

// Получаем введенные имя пользователя и пароль

string username = UsernameTextBox.Text;

string password = PasswordBox.Password;

// Создаем экземпляр AuthService для аутентификации

var authService = new AuthService();

User currentUser = authService.Authenticate(username, password);

// Проверяем, был ли найден пользователь

if (currentUser != null)

{

// Если аутентификация успешна, открываем главное окно с задачами

var mainPage = new MainPage(currentUser.Id); // Передаем идентификатор пользователя

mainPage.Show();

// Закрываем окно входа

this.Close();

}

else

{

// Если аутентификация неудачна, выводим сообщение

MessageBox.Show("Неправильное имя пользователя или пароль.", "Ошибка входа", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

Листинг кода 2 – MainPage

using Npgsql;

using System.Windows;

namespace AuthApp

{

public partial class MainPage : Window

{

private int \_userId;

private DatabaseHelper \_dbHelper;

private User \_currentUser;

private string connectionString = "Host=localhost;Port=5432;Username=postgres;Password=postgres;Database=docudb";

public MainPage(int userId)

{

InitializeComponent();

\_userId = userId; // Сохраняем идентификатор пользователя

\_dbHelper = new DatabaseHelper(); // Инициализируем DatabaseHelper

\_currentUser = GetUserById(\_userId);

if (\_currentUser != null && \_currentUser.IsAdmin)

{

CreateTaskButton.Visibility = Visibility.Visible;

}

var userProfileView = new UserProfileView(\_currentUser);

}

private User GetUserById(int userId)

{

// Метод для получения пользователя из базы данных по userId

using (var connection = new NpgsqlConnection(connectionString))

{

connection.Open();

var command = new NpgsqlCommand("SELECT Id, Username, Password, IsAdmin FROM users WHERE Id = @Id", connection);

command.Parameters.AddWithValue("@Id", userId);

using (var reader = command.ExecuteReader())

{

if (reader.Read())

{

return new User

{

Id = reader.GetInt32(0),

Username = reader.GetString(1),

Password = reader.GetString(2),

IsAdmin = reader.GetBoolean(3)

};

}

}

}

return null;

}

private void LoadTasksView()

{

// Создаем экземпляр TasksView, передавая userId для фильтрации задач

var tasksView = new TasksView(\_userId);

Продолжение листинга кода 2

ContentArea.Content = tasksView; // ContentArea — это элемент, где отображается TasksView

}

private void OpenCreateTaskWindow()

{

if (\_currentUser.IsAdmin)

{

var createTaskWindow = new CreateTaskWindow(\_currentUser.Id);

createTaskWindow.ShowDialog();

LoadTasksView();

}

else

{

MessageBox.Show("У вас нет прав для создания задачи.", "Доступ запрещен", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void CreateTaskButton\_Click(object sender, RoutedEventArgs e)

{

OpenCreateTaskWindow();

}

private void TasksButton\_Click(object sender, RoutedEventArgs e)

{

LoadTasksView();

}

private void DocumentsButton\_Click(object sender, RoutedEventArgs e)

{

var documentsView = new DocumentsView(\_userId); // Передаем ID пользователя

ContentArea.Content = documentsView; // Отображаем список документов

}

private void ProfileButton\_Click(object sender, RoutedEventArgs e)

{

UserProfileView userProfileView = new UserProfileView(\_currentUser);

ContentArea.Content = userProfileView;

}

private void ReportButton\_Click(object sender, RoutedEventArgs e)

{

var reportWindow = new ReportWindow();

ContentArea.Content = reportWindow;

}

}

}

Листинг кода 3 – UserProfileView

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

namespace AuthApp

{

public partial class UserProfileView : UserControl

{

private User \_currentUser;

private DatabaseHelper \_dbHelper;

public UserProfileView(User user)

{

InitializeComponent();

\_dbHelper = new DatabaseHelper();

\_currentUser = user;

DisplayUserInfo();

}

private void DisplayUserInfo()

{

UsernameTextBlock.Text = \_currentUser.Username;

RoleTextBlock.Text = \_currentUser.IsAdmin ? "Администратор" : "Пользователь";

// Загружаем изображение профиля, если путь задан

if (!string.IsNullOrEmpty(\_currentUser.AvatarPath))

{

AvatarImage.Source = new BitmapImage(new Uri(\_currentUser.AvatarPath, UriKind.RelativeOrAbsolute));

}

else

{

AvatarImage.Source = new BitmapImage(new Uri("C:\\Users\\Admin\\Pictures\\i-\_2\_.png")); // Укажите путь к изображению по умолчанию

}

}

private void EditProfileButton\_Click(object sender, RoutedEventArgs e)

{

// Логика для редактирования профиля

var editProfileWindow = new EditProfileWindow(\_currentUser);

if (editProfileWindow.ShowDialog() == true)

{

\_currentUser = editProfileWindow.UpdatedUser;

DisplayUserInfo();

}

}

private void RefreshButton\_Click(object sender, RoutedEventArgs e)

{

// Обновляем информацию о пользователе из базы данных

\_currentUser = \_dbHelper.GetUserById(\_currentUser.Id);

DisplayUserInfo();

}

private void LogoutButton\_Click(object sender, RoutedEventArgs e)

{

// Логика для выхода из системы

Application.Current.Shutdown();

}

}

}

Листинг кода 4 – DocumentsView

using System.Collections.ObjectModel;

using System.Windows;

using System.Windows.Controls;

using System.IO;

namespace AuthApp

{

public partial class DocumentsView : UserControl

{

private DatabaseHelper \_dbHelper;

private int \_userId; // Поле для хранения ID пользователя

public ObservableCollection<Document> Documents { get; set; }

public DocumentsView(int userId)

{

InitializeComponent();

\_dbHelper = new DatabaseHelper();

\_userId = userId; // Сохраняем ID пользователя

Documents = new ObservableCollection<Document>();

DocumentsListView.ItemsSource = Documents;

LoadDocuments(); // Загружаем документы из базы

}

private void LoadDocuments()

{

var dbHelper = new DatabaseHelper();

var documents = dbHelper.GetDocuments(); // Получаем список документов

Documents.Clear();

foreach (var document in documents)

{

Documents.Add(document); // Добавляем документы в коллекцию

}

}

private void AddDocumentButton\_Click(object sender, RoutedEventArgs e)

{

var addDocumentWindow = new AddDocumentWindow(\_userId); // Передаем ID пользователя

if (addDocumentWindow.ShowDialog() == true)

{

LoadDocuments(); // Перезагрузка документов

}

}

private void DownloadButton\_Click(object sender, RoutedEventArgs e)

{

var button = sender as Button;

int documentId = (int)button.Tag;

// Получаем содержимое документа

byte[] content = \_dbHelper.GetDocumentContent(documentId);

if (content != null)

{

// Создаем диалог выбора файла для сохранения

var saveFileDialog = new Microsoft.Win32.SaveFileDialog

{

FileName = "document", // Имя файла по умолчанию

DefaultExt = ".pdf", // Расширение файла по умолчанию

Filter = "Documents (\*.pdf;\*.docx;\*.txt)|\*.pdf;\*.docx;\*.txt" // Разрешенные форматы файлов

};

Продолжение листинга кода 4

// Показываем диалог и проверяем, было ли нажато "OK"

if (saveFileDialog.ShowDialog() == true)

{

string filePath = saveFileDialog.FileName;

File.WriteAllBytes(filePath, content); // Сохраняем содержимое документа в выбранный файл

MessageBox.Show("Документ успешно загружен по адресу: " + filePath);

}

}

else

{

MessageBox.Show("Не удалось скачать документ.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void DeleteButton\_Click(object sender, RoutedEventArgs e)

{

var button = sender as Button;

int documentId = (int)button.Tag;

// Подтверждение удаления

var result = MessageBox.Show("Вы уверены, что хотите удалить этот документ?", "Подтверждение", MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes)

{

\_dbHelper.DeleteDocument(documentId);

MessageBox.Show("Документ удален.", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

LoadDocuments(); // Перезагрузите список документов

}

}

}

}

Листинг кода 5 – TasksView

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

namespace AuthApp

{

public partial class TasksView : UserControl

{

private int \_userId; // Идентификатор текущего пользователя

private DatabaseHelper \_dbHelper; // Вспомогательный класс для работы с базой данных

public TasksView(int userId)

{

InitializeComponent();

\_userId = userId; // Сохраняем userId

\_dbHelper = new DatabaseHelper(); // Инициализируем DatabaseHelper

LoadTasks(); // Загружаем задачи

}

// Метод для загрузки задач

private void LoadTasks()

{

List<Task> tasks = \_dbHelper.GetTasks(\_userId); // Получаем задачи для текущего пользователя

TasksItemsControl.ItemsSource = tasks; // Привязываем задачи к элементу управления

Продолжение листинга кода 5

}

private void EditButton\_Click(object sender, RoutedEventArgs e)

{

var task = (sender as Button)?.Tag as Task; // Получаем задачу из Tag кнопки

if (task != null)

{

var editTaskWindow = new CreateTaskWindow(\_userId, task); // Передаем userId и объект Task

if (editTaskWindow.ShowDialog() == true)

{

LoadTasks(); // Обновляем список задач после редактирования

}

}

}

private void DeleteButton\_Click(object sender, RoutedEventArgs e)

{

var button = sender as Button;

if (button?.Tag is int taskId && taskId != 0) // Проверяем, что Tag является int

{

\_dbHelper.DeleteTask(taskId); // Удаляем задачу из базы

LoadTasks(); // Обновляем список задач

}

}

}

}

Листинг кода 6 – CreateTaskWindow

using System;

using System.Windows;

using System.Linq;

using System.Windows.Controls;

namespace AuthApp

{

public partial class CreateTaskWindow : Window

{

private DatabaseHelper \_dbHelper;

private int \_adminId;

private Task \_task;

// Конструктор для создания новой задачи

public CreateTaskWindow(int adminId)

{

InitializeComponent();

\_dbHelper = new DatabaseHelper();

\_adminId = adminId;

// Очистить поля для новой задачи

ClearFields();

}

// Конструктор для редактирования существующей задачи

public CreateTaskWindow(int adminId, Task task) : this(adminId)

{

\_task = task; // Сохраняем ссылку на задачу

PopulateFields(task); // Заполняем поля данными задачи

}

private void ClearFields()

{

TitleTextBox.Clear();

DescriptionTextBox.Clear();

ExecutorTextBox.Clear();

Продолжение листинга кода 6

StatusComboBox.SelectedIndex = 0; // Установить статус по умолчанию

}

private void PopulateFields(Task task)

{

TitleTextBox.Text = task.Title;

DescriptionTextBox.Text = task.Description;

ExecutorTextBox.Text = task.Executor;

StatusComboBox.SelectedItem = StatusComboBox.Items

.OfType<ComboBoxItem>()

.FirstOrDefault(item => item.Content.ToString() == task.Status);

}

private void CreateTaskButton\_Click(object sender, RoutedEventArgs e)

{

string title = TitleTextBox.Text;

string description = DescriptionTextBox.Text;

string executor = ExecutorTextBox.Text;

string status = (StatusComboBox.SelectedItem as ComboBoxItem)?.Content.ToString() ?? "В процессе";

if (\_task == null) // Если задача новая

{

\_task = new Task

{

Title = title,

Description = description,

Executor = executor,

StartDate = DateTime.Now,

Status = status,

UserId = \_adminId // Связываем задачу с администратором, который ее создал

};

\_dbHelper.AddTask(\_task); // Метод для добавления задачи в базу данных

MessageBox.Show("Задача успешно создана!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

else // Если задача редактируется

{

\_task.Title = title;

\_task.Description = description;

\_task.Executor = executor;

\_task.Status = status;

// Если статус изменился на "Completed", устанавливаем EndDate

DateTime? endDate = null;

if (\_task.Status == "Закрыта" && \_task.EndDate == null)

{

endDate = DateTime.Now; // Устанавливаем текущую дату и время

}

// Обновляем задачу в базе данных

\_dbHelper.UpdateTask(\_task.Id, \_task.Title, \_task.Description, \_task.Executor, \_task.Status, endDate);

MessageBox.Show("Задача успешно обновлена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

this.Close(); // Закрываем окно

}

}

}

Листинг кода 6 – ReportWindow

using System;

using System.Windows;

using System.Windows.Controls;

namespace AuthApp

{

public partial class ReportWindow : UserControl

{

private DatabaseHelper \_databaseHelper;

public ReportWindow()

{

InitializeComponent();

\_databaseHelper = new DatabaseHelper();

LoadStatistics(1);

}

private void LoadStatistics(int userId)

{

// Получаем статистику по документам

var documentStats = \_databaseHelper.GetDocumentStatistics(userId);

TotalDocumentsText.Text = $"Общее количество документов: {documentStats.totalDocuments}";

RecentDocumentsText.Text = $"Загружено за последние 30 дней: {documentStats.documentsLast30Days}";

AccessibleDocumentsText.Text = $"Доступные документы: {documentStats.accessibleDocuments}";

// Получаем статистику по задачам

var taskStats = \_databaseHelper.GetTaskStatistics(userId);

CurrentTasksText.Text = $"Текущие задачи: {taskStats.currentTasks}";

CompletedTasksText.Text = $"Завершенные задачи: {taskStats.completedTasks}";

AverageCompletionTimeText.Text = $"Среднее время выполнения задач: {taskStats.avgCompletionTime:F0} дней";

// Получаем общую статистику

var generalStats = \_databaseHelper.GetGeneralStatistics();

ActiveUsersText.Text = $"Активные пользователи: {generalStats.activeUsers}";

DocumentUpdatesText.Text = $"Загрузки или обновления за последний месяц: {generalStats.documentUploadsLastMonth}";

}

}

}