

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка системы электронного документооборота для торговой
организации»

Обучающийся

В.Д. Гончаров

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Данная выпускная квалификационная работа посвящена разработке системы электронного документооборота (СЭД) для торговой организации. Актуальность темы обусловлена необходимостью цифровизации бизнес-процессов, повышением требований к юридической значимости электронных документов и стремлением к ускорению и упрощению обработки информации. Цифровой документооборот позволяет автоматизировать создание, согласование, хранение и передачу документов, обеспечивая соответствие требованиям законодательства Российской Федерации, включая использование электронной подписи и защиту конфиденциальных данных.

Основной целью работы является создание надёжной, масштабируемой и безопасной СЭД, адаптированной под нужды конкретной торговой компании. Для достижения этой цели выполнены: анализ предметной области, обзор программных аналогов, определение требований к системе, проектирование архитектуры, реализация программного обеспечения и его тестирование. В работе особое внимание уделяется вопросам информационной безопасности, включая аутентификацию, разграничение прав доступа и шифрование.

Структурно работа включает введение, три главы и заключение. В первой главе представлены теоретические основы, анализ аналогов и требования к системе. Вторая глава посвящена проектированию архитектуры и моделей данных. В третьей главе описаны этапы разработки и тестирования.

Результаты исследования могут быть применены для внедрения СЭД в торговых организациях и развития цифровой среды документооборота.

Оглавление

Введение	4
Глава 1 Постановка задачи на разработку программного обеспечения.....	7
1.1 Анализ и описание предметной области	7
1.2 Обзор и анализ аналогов программного обеспечения.....	15
1.3 Разработка требований к программному обеспечению	23
Глава 2 Проектирование программного обеспечения	30
2.1 Выбор методологии проектирования программного обеспечения.....	30
2.2 Логическое моделирование программного обеспечения	39
2.3 Моделирование данных системы	49
Глава 3 Реализация и тестирование программного обеспечения	58
3.1 Реализация программного обеспечения	58
3.2 Тестирование программного обеспечения.....	73
Заключение	82
Список используемых источников	85

Введение

В условиях стремительно развивающегося цифрового общества эффективное управление информационными потоками становится ключевым фактором успеха предприятий, особенно в торговом секторе. Современные компании сталкиваются с необходимостью быстрой адаптации к изменениям: увеличение объёма обрабатываемой информации, рост требований к прозрачности процессов, безопасности и соблюдению нормативов. В этих условиях важную роль играет цифровизация документооборота – переход от бумажных носителей к электронным форматам, автоматизация обработки деловых документов и интеграция с другими бизнес-системами.

История развития электронного документооборота (ЭДО) в России насчитывает более двух десятилетий. Первоначально СЭД использовались преимущественно в государственных учреждениях и крупных корпорациях. Однако с развитием облачных технологий, появлением стандартов юридически значимого ЭДО и широким распространением электронных подписей, такие системы стали востребованы и среди малого и среднего бизнеса. Сегодня развитие СЭД продолжается в направлении повышения автоматизации процессов, интеграции с внешними платформами, обеспечения безопасности данных и пользовательского удобства.

Актуальность выбранной темы обусловлена необходимостью модернизации процессов документооборота в организациях торговой сферы. Многие предприятия до сих пор используют разрозненные, устаревшие или неунифицированные системы, что приводит к потере данных, дублированию операций, снижению управляемости и росту рисков. Разработка современной системы электронного документооборота, адаптированной к нуждам торговой организации, является актуальной и практически значимой задачей.

В данной предметной области существует ряд родственных решений, таких как «1С:Документооборот», Directum, EOS for SharePoint, а также

международные продукты (DocuWare, OpenText и др.). Однако многие из них ориентированы либо на крупный бизнес, либо требуют значительных затрат на внедрение и сопровождение. Это открывает возможность для создания более лёгкого, гибкого и адаптируемого решения, ориентированного на нужды конкретной организации.

Объектом исследования являются процессы документооборота торговой организации, заинтересованной в цифровизации внутренних процессов. Предмет исследования – методы автоматизации документооборота с использованием современных информационных технологий.

Цель настоящей выпускной квалификационной работы – разработка программного обеспечения для системы электронного документооборота, отвечающего современным требованиям безопасности, масштабируемости и интеграции, а также учитывающего специфику деятельности торговой организации.

Для достижения цели были поставлены следующие задачи:

- провести анализ предметной области и существующих решений в сфере электронного документооборота;
- определить функциональные и нефункциональные требования к создаваемой системе;
- разработать архитектуру программного обеспечения, обеспечивающую отказоустойчивость, масштабируемость и безопасность;
- реализовать основные модули системы: регистрация, подписание, хранение документов и защита информации;
- провести тестирование системы в условиях, приближённых к реальной эксплуатации.

В работе использовались методы анализа, синтеза, моделирования и проектирования программных систем. Реализация программной части включала использование технологий Java Spring, MinIO, JWT, Apache

PDFBox, Bouncy Castle и других компонентов современной экосистемы разработки.

Научно-практическая значимость работы заключается в том, что разработанная система может быть адаптирована для использования в различных организациях малого и среднего бизнеса. Её архитектура предполагает возможность интеграции с бухгалтерскими системами, CRM и платформами юридически значимого ЭДО. Это делает систему перспективной для дальнейшего развития.

Структура работы включает обзор предметной области, анализ существующих программных решений, проектирование и реализацию программного обеспечения, а также тестирование и оценку полученного результата.

Выпускная квалификационная работа состоит из 87 страниц и содержит 23 рисунка, 2 таблицы, 30 источников.

Глава 1 Постановка задачи на разработку программного обеспечения

1.1 Анализ и описание предметной области

В современном мире, который характеризуется стремительным развитием информационных технологий и цифровых инструментов, особое значение приобретает вопрос эффективной организации документооборота в коммерческих и торговых структурах. Электронный документооборот, или, как его часто называют в деловой среде, СЭД (система электронного документооборота), представляет собой совокупность программных и технических решений, направленных на автоматизацию и оптимизацию процессов создания, обработки, согласования, хранения и передачи различных видов документов между участниками бизнес-процессов.

В условиях современной российской действительности, когда торговые компании сталкиваются с постоянным ужесточением требований со стороны регулирующих органов, ростом конкуренции на рынке, увеличением объёмов данных и документов, циркулирующих внутри организации, вопросы рационального управления документооборотом становятся особенно актуальными. Эффективно налаженный документооборот позволяет не только сократить время обработки заявок и ускорить процессы принятия управленческих решений, но и значительно повысить прозрачность деятельности компании, снизить вероятность ошибок, связанных с человеческим фактором, и обеспечить надлежащий уровень сохранности корпоративной информации.

Необходимо также отметить, что система электронного документооборота должна не только соответствовать внутренним требованиям предприятия, но и учитывать внешние регламенты, устанавливаемые государственными органами Российской Федерации. В

частности, внедрение СЭД должно осуществляться с учётом норм таких законодательных актов, как:

- федеральный закон № 63-ФЗ от 6 апреля 2011 года «Об электронной подписи», определяющий юридическую силу и применение электронной подписи при обмене документами;
- федеральный закон № 149-ФЗ от 27 июля 2006 года «Об информации, информационных технологиях и о защите информации», регулирующий вопросы правовой охраны и обработки информации;
- федеральный закон № 402-ФЗ от 6 декабря 2011 года «О бухгалтерском учёте», обязывающий компании обеспечивать надлежащее документальное оформление хозяйственных операций.

Таким образом, грамотное и всестороннее внедрение СЭД позволяет не только автоматизировать внутренние процессы компании, но и обеспечить соответствие текущему законодательству, что особенно важно в контексте взаимодействия с государственными структурами и внешними контрагентами.

Документооборот в торговых организациях, особенно работающих на российском рынке, обладает рядом характерных черт, которые определяются спецификой сферы торговли, особенностями внутреннего устройства компании, а также внешними факторами, включая экономические, правовые и технологические условия.

Прежде всего, следует отметить, что торговая деятельность предполагает постоянный обмен различными видами документов, как внутри компании, так и с внешними участниками рынка. Среди наиболее распространённых категорий документов, активно используемых в торговых организациях, можно выделить следующие:

- документы, связанные с поставками товаров (заказы, накладные, счета-фактуры, акты приёма-передачи);
- договорные документы, регулирующие взаимоотношения с поставщиками, дистрибьюторами и клиентами;

- внутренние документы, касающиеся кадровых вопросов, распоряжений, приказов, служебных записок;
- финансово-бухгалтерская документация, включая отчёты, первичные документы, ведомости;
- документация, связанная с логистикой и складским учетом;
- документы, связанные с налоговой отчётностью, включая формы, отчёты, декларации и переписку с налоговыми органами.

Следует понимать, что в рамках одного торгового предприятия одновременно может обрабатываться значительное количество документов, циркулирующих между различными структурными подразделениями: отделами продаж, закупок, логистики, бухгалтерии, юридическим департаментом, а также руководящим составом. Каждый из этих отделов выполняет свою роль в процессе обработки документации, что приводит к необходимости точной координации и выстраивания чётких маршрутов согласования.

Особое внимание следует уделить необходимости строгого контроля над сроками прохождения документов, так как нарушение графиков поставок, оплат или согласований может привести к убыткам, потерям клиентов и ухудшению репутации компании на рынке. В условиях жёсткой конкуренции подобные издержки крайне нежелательны.

Нельзя не отметить и тот факт, что работа с большим объёмом бумажной документации сопряжена с рядом трудностей, включая высокие трудозатраты на хранение и поиск, риски потери документов, а также сложности с обеспечением информационной безопасности. Все эти проблемы становятся особенно ощутимыми при масштабировании бизнеса, открытии новых филиалов и увеличении штата сотрудников [26].

В связи с этим внедрение системы электронного документооборота представляется необходимым и оправданным шагом, направленным на обеспечение стабильной, надёжной и эффективной работы предприятия в

условиях растущих требований к скорости обработки информации и качеству внутренних процессов.

Документы, циркулирующие в торговой организации, можно классифицировать по различным признакам, каждый из которых отражает определённые аспекты их использования и обработки. Такая классификация необходима для более точного построения логики функционирования системы электронного документооборота, а также для определения специфики маршрутов движения документации и назначения ролей пользователям системы [17].

В первую очередь документы можно подразделить по функциональному назначению, что является наиболее распространённой и прикладной классификацией в рамках делопроизводства:

- организационно-распорядительные документы – включают приказы, распоряжения, регламенты, положения и инструкции. Эти документы регулируют внутренние процессы предприятия, устанавливают правила, обязательные к выполнению, и направлены на организацию эффективной работы персонала;
- финансово-учётные документы – включают накладные, счета, акты, платёжные поручения, отчёты и другие документы, отражающие хозяйственные операции, проводимые в организации. Эти документы играют ключевую роль в бухгалтерском учёте, налогообложении и взаимодействии с контрагентами;
- кадровые документы – охватывают весь спектр документации, связанной с управлением персоналом: приказы о приёме и увольнении, заявления сотрудников, трудовые договоры, табели учёта рабочего времени и т.д.;
- договорные документы – включают в себя все формы договоров, соглашений, приложений и дополнительных соглашений,

оформляющих юридические отношения между организацией и её партнёрами, поставщиками, клиентами;

- служебная переписка – представляет собой внутренние и внешние письма, уведомления, запросы и прочие формы деловой коммуникации, как в бумажном, так и в электронном виде;
- отчётная документация – предназначена для представления результатов деятельности отдельных подразделений или компании в целом. Может быть как внутренней, так и предназначенной для внешних пользователей (например, отчёты перед налоговыми органами).

Также возможна классификация документов по направлению движения: входящие, исходящие и внутренние. Такая классификация важна для построения маршрутов прохождения документов, настройки прав доступа и фильтрации информации в пользовательском интерфейсе системы.

Кроме того, документы могут отличаться по формату представления: бумажные, электронные (в формате PDF, DOCX, XML и др.), а также гибридные (отсканированные копии бумажных оригиналов). В условиях цифровизации акцент всё больше смещается в сторону полного отказа от бумажных носителей.

Таким образом, для эффективной работы СЭД необходимо предусмотреть обработку всех указанных типов документов, их маршрутизацию, шаблонизацию и унификацию форматов, что будет способствовать улучшению качества документооборота.

На текущий момент многие торговые организации сталкиваются с рядом проблем и ограничений, затрудняющих эффективное управление документооборотом. Эти проблемы носят как технический, так и организационный характер и в совокупности могут серьёзно снижать эффективность деятельности предприятия [1].

Одной из наиболее острых проблем остаётся избыточная бюрократизация процессов, когда каждое действие по созданию, согласованию или утверждению документа требует множественных согласований, зачастую дублирующих друг друга. Отсутствие автоматизированного маршрута движения документа приводит к потере времени, увеличению операционных издержек и росту человеческого фактора.

Второй серьёзной проблемой является низкая прозрачность процессов. При бумажной или частично автоматизированной системе нередко невозможно точно определить, на каком этапе находится документ, кто несёт ответственность за задержку, и какие действия были предприняты. Это, в свою очередь, негативно влияет на контроль исполнения и снижает управляемость процессами [27].

Также существенным ограничением является отсутствие единого информационного пространства. Отдельные подразделения могут использовать разные методы и форматы хранения документов, что делает невозможной централизованную обработку и поиск информации. Возникает необходимость в повторном вводе данных, что увеличивает вероятность ошибок.

Немаловажным фактором остаются проблемы информационной безопасности. Отсутствие централизованной системы разграничения доступа, шифрования и контроля версий создаёт угрозы утечек данных, несанкционированного доступа к конфиденциальной информации, что особенно критично для финансовых и юридических документов.

Отдельно стоит отметить трудности, связанные с интеграцией с внешними системами – например, платформами ЭДО, бухгалтерским ПО (1С), CRM-системами и государственными сервисами. В условиях отсутствия единых протоколов обмена данными интеграция становится сложной задачей [4].

Устранение указанных проблем возможно только путём внедрения современной, комплексной системы электронного документооборота, которая будет учитывать как внутренние потребности организации, так и внешние требования.

В условиях стремительного роста объёмов информации и ужесточения требований к скорости обработки документов, автоматизация процессов становится не просто желательной, а насущно необходимой. Компании, стремящиеся сохранить конкурентоспособность и повысить эффективность своей деятельности, неизбежно приходят к необходимости внедрения СЭД.

Автоматизация документооборота позволяет:

- значительно сократить временные затраты на обработку документов за счёт устранения ручного труда, оптимизации маршрутов согласования и использования шаблонов;
- повысить прозрачность и управляемость внутренних процессов путём мониторинга состояния каждого документа в реальном времени;
- снизить издержки, связанные с бумажным документооборотом (печать, доставка, архивирование), а также избежать штрафов и санкций, связанных с нарушением сроков;
- обеспечить надёжное хранение информации с учётом сроков хранения и соблюдением нормативных требований;
- повысить качество обслуживания клиентов и партнёров, ускоряя цикл от запроса до подписания и выполнения обязательств [25].

Более того, автоматизация документооборота открывает возможности для глубокой аналитики и построения отчётности, выявления узких мест в бизнес-процессах и формирования управленческих решений на основе объективных данных.

Таким образом, внедрение СЭД – это не просто инструмент, а стратегический шаг на пути цифровой трансформации бизнеса.

Российский рынок предъявляет к системам электронного документооборота целый ряд требований, связанных как с нормативно-правовыми аспектами, так и с практическими ожиданиями со стороны бизнеса.

Во-первых, обязательным условием является соответствие законодательству РФ, в частности, требованиям по использованию квалифицированной электронной подписи (КЭП), электронных архивов, и соблюдение сроков хранения согласно нормативам.

Во-вторых, российские компании ожидают от системы высокой степени адаптируемости — возможность легко настраивать маршруты документов, шаблоны, роли пользователей, не прибегая к доработкам кода или вмешательству программистов.

Также важными требованиями являются:

- интеграция с 1С и системами ЭДО;
- поддержка гибкой настройки прав доступа и разграничения ролей;
- работа в условиях распределённых офисов и филиалов, в том числе с ограниченной связью;
- удобный пользовательский интерфейс, не требующий длительного обучения;
- возможность локального или облачного размещения, в зависимости от политики безопасности организации.

Рынок также предъявляет повышенные требования к скорости обработки, надёжности, и способности масштабироваться с ростом компании. Всё это делает задачу разработки СЭД для российского рынка технически и организационно сложной, но при этом крайне актуальной [3].

Исходя из проведённого анализа предметной области, можно сформулировать ключевые цели разработки системы электронного документооборота для торговой организации. Главная цель – создание эффективной, надёжной и масштабируемой системы, обеспечивающей

автоматизацию документооборота с учётом требований российского законодательства и специфики торговой отрасли.

Для достижения этой цели необходимо решить следующие задачи:

- разработать архитектуру системы, включающую удобный интерфейс, отказоустойчивое хранилище данных и гибкую модель маршрутизации;
- реализовать механизмы регистрации, согласования, подписания и хранения документов, обеспечивающие прозрачность всех этапов жизненного цикла;
- интегрировать систему с внешними сервисами, включая платформы ЭДО, бухгалтерские системы, электронную почту;
- обеспечить высокий уровень безопасности, включая аутентификацию, авторизацию, шифрование данных и контроль доступа;
- создать механизм уведомлений и отслеживания статусов документов, повышающий оперативность и снижая риск задержек;
- обеспечить поддержку версионности и истории изменений документов, что особенно важно в юридически значимом документообороте.

Таким образом, проектируемая система должна стать не просто инструментом автоматизации, а основой для построения современного цифрового офиса в рамках торговой организации.

1.2 Обзор и анализ аналогов программного обеспечения

Разработка собственной системы электронного документооборота невозможна без тщательного и всестороннего изучения существующих на рынке решений, как отечественных, так и зарубежных. Подобный анализ представляет собой важный и обязательный этап в исследовательской и

проектной деятельности, позволяющий определить текущее состояние предметной области, выявить тенденции в развитии технологий, изучить принятые в индустрии подходы, а также — понять, какие задачи и проблемы уже решены, а какие ещё остаются открытыми. Только на основании глубокого и разностороннего анализа аналогов возможно создать конкурентоспособное программное обеспечение, ориентированное на реальные потребности целевой аудитории.

Кроме того, анализ существующих программных решений позволяет не только оценить функциональность и архитектурные особенности тех или иных систем, но и провести определённую работу по «переосмыслению» текущих рыночных стандартов. Это важно в условиях, когда от разрабатываемого ПО требуется не только соответствие базовым требованиям, но и наличие уникальных конкурентных преимуществ, позволяющих занять свою нишу среди уже представленных решений.

Учитывая, что данная бакалаврская работа ориентирована на реалии российского рынка и должен соответствовать требованиям отечественного законодательства и практик документооборота, в данном разделе основной акцент будет сделан именно на российские системы. Тем не менее, в целях полноты картины и для получения представления о международных трендах, будут также кратко рассмотрены наиболее известные зарубежные аналоги.

Российские аналоги систем электронного документооборота. Следует отметить, что в последние десятилетия в Российской Федерации сложился достаточно насыщенный и активно развивающийся рынок программных продуктов в области автоматизации делопроизводства и управления документами. Это обусловлено как растущими требованиями к цифровизации управленческих процессов в организациях, так и необходимостью соответствия государственным стандартам и нормативным актам в части документооборота, хранения и юридической значимости электронных документов [14].

СЭД «ДЕЛО» (ЭОС).

Система «ДЕЛО», разработанная компанией ЭОС, на протяжении многих лет занимает лидирующие позиции среди российских систем электронного документооборота. Это зрелый и проверенный временем продукт, активно применяемый как в органах государственной власти, так и в коммерческом секторе. Популярность данной системы объясняется в первую очередь её функциональной полнотой, соответствием российским стандартам (включая ГОСТы, требования Минцифры и другие нормативы), а также наличием широкой сети внедренцев и специалистов [21].

Программный комплекс охватывает весь цикл документооборота – от регистрации и создания документа, его согласования, утверждения и подписания, до передачи в архив и обеспечения длительного хранения. Важным преимуществом является модульность и возможность интеграции с другими продуктами компании, такими как «Архивное ДЕЛО», «Канцелярия» и др.

Интерфейс системы, несмотря на некоторое визуальное устаревание, остаётся понятным для пользователей, привыкших к традиционным схемам работы с документами. Однако в ряде случаев может потребоваться обучение сотрудников и адаптация внутренних процессов под особенности платформы.

СЭД «ТЕЗИС» (Haulmont).

Система электронного документооборота «ТЕЗИС» представляет собой гибкое и современное решение, ориентированное на корпоративный сегмент. В основе продукта лежит открытая Java-платформа CUBA Platform, что позволяет глубоко кастомизировать систему под индивидуальные потребности заказчика [28].

Одним из ключевых преимуществ «ТЕЗИСа» является интуитивно понятный web-интерфейс, а также наличие мобильных приложений, что особенно важно для крупных торговых организаций с территориально распределённой структурой. Кроме того, система предлагает широкие

возможности по настройке маршрутов согласования, что позволяет автоматизировать даже сложные управленческие процессы [29].

Немаловажно и то, что разработчик активно развивает продукт, регулярно выпускает обновления и адаптирует систему под изменения в законодательстве и нормативных актах.

Directum RX.

Directum RX – это одна из наиболее технологичных и универсальных платформ в сегменте СЭД, разработанная с учётом современных требований к цифровой трансформации бизнеса. Система позиционируется как единая цифровая среда для управления контентом, взаимодействиями и процессами, что делает её подходящей не только для документооборота, но и для реализации более широких задач – от договорной работы до управления персоналом [24].

Среди характерных черт Directum RX можно выделить наличие встроенного процессного движка, поддержку сценариев с применением роботов, мощную систему прав доступа и модульную архитектуру. Всё это делает продукт масштабируемым и адаптируемым к различным бизнес-контекстам. Однако за такую функциональную насыщенность приходится платить – как в буквальном смысле, так и в плане сложности внедрения.

DocsVision.

DocsVision – ещё один яркий представитель российских СЭД, широко применяемый в крупных компаниях и госкорпорациях. Система реализует концепцию модульности, что позволяет наращивать функциональность по мере роста потребностей организации.

Продукт тесно интегрирован с Microsoft Office, что делает его особенно удобным для пользователей, уже знакомых с данной экосистемой. Наличие развитого API позволяет реализовывать нестандартные сценарии, а модуль документооборота – настроить детальные маршруты согласования.

ПИЛОТ:Документооборот.

Решение «ПИЛОТ:Документооборот» ориентировано на малый и средний бизнес, предлагая базовые функции СЭД по доступной цене. Несмотря на относительную простоту, система обеспечивает все необходимые процессы: регистрацию, маршрутизацию, согласование и хранение документов, а также – базовые возможности поиска и отчётности.

Хотя основное внимание при выборе СЭД в условиях российского рынка сосредоточено на отечественных продуктах, нельзя не учитывать и достижения международных разработчиков. Зарубежные системы зачастую обладают более продвинутыми интерфейсами, удобной мобильной работой и высокими стандартами производительности. Однако при этом возникает ряд проблем, связанных с юридической значимостью документов, соответствием локальным требованиям и трудностями интеграции [5].

DocuWare. Система DocuWare пользуется спросом в Европе и США, предлагая широкий спектр решений для управления документами в цифровой форме. Высокий уровень защищённости данных, поддержка облачных хранилищ и масштабируемость делают её подходящей для предприятий любого масштаба. В то же время, отсутствие официальной локализации и поддержки российских форматов ЭЦП делает её менее привлекательной для отечественных компаний [2].

OpenText. OpenText – канадская компания, выпускающая ЕСМ-решения, предназначенные для корпоративного использования в международных организациях. Продукты OpenText характеризуются высокой степенью надёжности, развитой инфраструктурой, но крайне сложно поддаются внедрению в среде малого и среднего бизнеса. Кроме того, стоимость владения такими решениями чрезвычайно высока.

Microsoft SharePoint.

SharePoint – универсальная платформа от Microsoft, активно применяемая для внутреннего документооборота, хранения информации и совместной работы. Благодаря тесной интеграции с продуктами Office, она

пользуется популярностью во всём мире. Однако использование SharePoint в качестве полноценной СЭД требует существенной настройки, а поддержка российской ЭЦП и интеграция с государственными сервисами – вовсе невозможна без доработок [20].

Alfresco.

Система с открытым исходным кодом, предоставляющая гибкость и возможность развертывания в любых условиях. Популярна в IT-среде и среди энтузиастов. Тем не менее, для серьёзной эксплуатации в российских компаниях требует значительных ресурсов на кастомизацию и настройку.

С целью получения наиболее наглядного представления о различиях между рассмотренными программными решениями, в настоящем подразделе приведена обобщённая сравнительная таблица 1. Данная таблица основывается на таких ключевых параметрах, как функциональность, масштабируемость, удобство пользовательского интерфейса, наличие мобильной версии, возможности интеграции, стоимость внедрения и эксплуатации, а также соответствие требованиям российского законодательства.

Таблица 1 – Сравнительная характеристика аналогов СЭД

Наименование системы	Функциональность	Интерфейс и UX	Мобильность	Интеграции	Законодательная адаптация	Стоимость	Уровень кастомизации
ДЕЛО (ЭОС)	Высокая	Средний	Ограничена	Да	Полная	Средняя	Средний
ТЕЗИС	Высокая	Высокий	Да	Да	Полная	Средняя	Высокий
Directum RX	Очень высокая	Высокий	Да	Да	Полная	Высокая	Очень высокий

Продолжение таблицы 1

DocsVision	Высокая	Средний	Частично	Да	Полная	Средняя	Высокий
ПИЛОТ:Документооборот	Средняя	Простой	Нет	Нет	Частичная	Низкая	Ограниченный
DocuWare	Высокая	Высокий	Да	Да	Отсутствует	Высокая	Средний
OpenText	Очень высокая	Высокий	Да	Да	Отсутствует	Очень высокая	Высокий
SharePoint	Средняя	Высокий	Да	Да	Ограниченная	Высокая	Средний
Alfresco	Средняя	Средний	Да	Да	Ограниченная	Низкая	Очень высокий

Как видно из таблицы выше, наиболее сбалансированными по соотношению функциональности, стоимости и адаптации к российским реалиям являются такие системы, как ТЕЗИС, Directum RX, и ДЕЛО. В то же время, зарубежные решения, несмотря на продвинутый интерфейс и высокую технологичность, зачастую проигрывают отечественным аналогам по критически важному показателю – соответствию законодательству РФ и поддержке российской электронной подписи.

На основании проведённого анализа можно сделать несколько ключевых выводов, имеющих значение для принятия архитектурных и проектных решений при разработке собственной системы электронного документооборота.

Во-первых, рынок программного обеспечения в сфере СЭД в России сформировался и представляет собой достаточно зрелую отрасль с устойчивыми игроками. Продукты компаний ЭОС, Directum, DocsVision и Haulmont успешно решают широкий спектр задач, связанных с управлением

документами, и в большинстве случаев полностью соответствуют требованиям государственных стандартов и нормативных актов.

Во-вторых, отечественные системы, как правило, более адаптированы к российским реалиям, включая работу с электронной подписью, интеграцию с госуслугами, СМЭВ, нормативными справочниками и другими обязательными компонентами документооборота. Также они демонстрируют лучшее понимание специфики работы российских организаций, в частности, структуры подчинённости, маршрутов согласования и особенностей кадрового документооборота.

В-третьих, несмотря на технологическую продвинутость многих иностранных решений, их использование в российском контексте сопряжено с рядом трудностей. Прежде всего, это касается соответствия локальному законодательству, поддержки отечественных форматов ЭЦП, а также ограниченной локализации. Кроме того, важным сдерживающим фактором является высокая стоимость внедрения, особенно в условиях ограниченных бюджетов [22].

В-четвёртых, все рассмотренные СЭД имеют как сильные, так и слабые стороны. Например, Directum RX отличается исключительной гибкостью и расширяемостью, но требует значительных ресурсов на внедрение и обучение. Система «ДЕЛО», в свою очередь, является стабильным и понятным решением, но несколько отстаёт в плане интерфейсных решений и мобильности.

Наконец, необходимо отметить, что ни одна из рассмотренных систем не является универсальным решением, способным удовлетворить абсолютно все потребности любого предприятия без дополнительной настройки или доработки. Именно поэтому многие организации делают выбор в пользу разработки собственного программного обеспечения или глубокой кастомизации существующих решений.

С учетом всего вышесказанного, можно прийти к обоснованному выводу о целесообразности разработки собственной системы электронного документооборота, ориентированной на специфику работы конкретной торговой организации. Такой подход позволит создать программный продукт, максимально соответствующий текущим и перспективным потребностям бизнеса, обладающий необходимой гибкостью, масштабируемостью и совместимостью с другими элементами информационной инфраструктуры предприятия.

1.3 Разработка требований к программному обеспечению

Создание информационной системы, в частности – системы электронного документооборота (СЭД) для торговой организации, невозможно без чётко сформулированных требований, отражающих как технические, так и бизнес-аспекты функционирования будущего программного продукта. Разработка требований представляет собой один из ключевых этапов жизненного цикла программного обеспечения и во многом определяет успешность проекта в целом. Именно на этапе формализации требований происходит трансформация разрозненных потребностей пользователей и организационных задач в структурированный и технически реализуемый набор характеристик системы.

Разработка требований является неотъемлемой частью инженерии требований – дисциплины, направленной на системный подход к выявлению, анализу, спецификации и валидации требований. Под требованиями к системе понимаются описания свойств и характеристик, которым должна удовлетворять система для успешного выполнения своей функции в рамках конкретной предметной области [6].

В соответствии с общепринятой практикой в области системного и программного инжиниринга, требования к информационной системе подразделяются на следующие основные категории:

- функциональные требования – описывают, что должна делать система, то есть конкретные функции, операции, бизнес-процессы, которые поддерживает программный продукт;
- нефункциональные требования – включают в себя характеристики качества системы: надёжность, производительность, удобство использования, безопасность, масштабируемость и другие параметры;
- ограничения (ограничивающие условия) – определяют рамки разработки, включая платформенные, аппаратные, нормативные и правовые ограничения;
- пользовательские требования – включают в себя пожелания конечных пользователей, нередко выраженные в менее формальном виде, но критически важные с точки зрения опыта взаимодействия с системой;
- бизнес-требования – отражают стратегические цели организации и ожидаемые эффекты от внедрения системы (например, повышение скорости документооборота, снижение числа ошибок, повышение прозрачности процессов и т.п.).

Каждое из этих направлений требует тщательной проработки, поскольку недостаточное внимание к любому из перечисленных аспектов может привести к необходимости дорогостоящих доработок, либо вовсе к провалу проекта.

Формулируя требования к разрабатываемому программному обеспечению, необходимо придерживаться следующих принципов:

- полнота – требования должны охватывать все значимые аспекты функционирования системы;
- однозначность – каждое требование должно быть трактуемо только одним способом;

- проверяемость – должно быть возможно однозначно определить, выполнено ли то или иное требование;
- прослеживаемость – должно быть понятно, откуда возникло каждое требование и на какой стадии разработки оно реализуется;
- согласованность – требования не должны противоречить друг другу;
- актуальность – требования должны соответствовать текущим реалиям и целям организации [19].

Соблюдение этих принципов особенно важно в условиях, когда система создаётся под конкретную организацию с учётом её внутренних процессов, структуры, уровня цифровой зрелости и специфики бизнес-деятельности. Функциональные требования определяют, какие задачи должна решать разрабатываемая система. Они вытекают напрямую из анализа бизнес-процессов и структуры документооборота, изложенных ранее. Ниже приведены основные функциональные возможности, которыми должна обладать система:

- регистрация документов – возможность ввода в систему новых документов, включая автоматическое формирование регистрационного номера, фиксацию даты и времени регистрации, указание автора и подразделения;
- согласование и подписание – организация маршрутов согласования и подписания документов с возможностью настройки логики маршрутов в зависимости от типа документа и структуры организации;
- редактирование и возврат на доработку – возможность изменения документов в статусах «Черновик» или после возврата с обоснованными замечаниями от согласующих лиц;
- контроль сроков исполнения – отслеживание установленных сроков исполнения и уведомление ответственных лиц о приближении или нарушении дедлайнов;

- хранение и архивирование – организация хранилища документов с поддержкой поиска, фильтрации, категоризации и перевода документов в архив с возможностью дальнейшего доступа;
- работа с шаблонами документов – использование унифицированных шаблонов для ускорения и стандартизации процесса подготовки документов.
- интеграция с внешними системами – возможность подключения к корпоративной почте, системам бухгалтерского учёта, HR-системам, а также государственным сервисам (например, СМЭВ, Госуслуги);
- ролевая модель доступа – разграничение прав доступа в зависимости от должностных обязанностей, уровня доступа и текущего статуса документа;
- система уведомлений и оповещений – отправка уведомлений пользователям о наступлении событий, требующих их внимания (например, запрос на согласование, изменение статуса, комментарии);
- система комментирования – возможность оставлять текстовые замечания и обсуждать документы в процессе согласования и доработки.

Все указанные функции должны быть реализованы в виде интуитивно понятного пользовательского интерфейса, не требующего длительного обучения для сотрудников с базовым уровнем компьютерной грамотности. Нефункциональные требования относятся не столько к тому, что делает система, сколько к тому, как она это делает. Данные характеристики определяют качество функционирования системы в условиях реального использования, а также влияют на удобство и надёжность её эксплуатации. Для разрабатываемой СЭД в контексте торговой организации на российском рынке основными нефункциональными требованиями являются следующие:

Надёжность. Система должна обеспечивать устойчивое функционирование при стандартных и пиковых нагрузках. Любые сбои в

работе должны быть минимизированы, а при их возникновении – задействованы механизмы аварийного восстановления и сохранения данных.

Производительность. Отклик пользовательского интерфейса на основные операции (открытие документа, сохранение, фильтрация) не должен превышать 2 секунд при стандартной нагрузке. Система должна поддерживать одновременную работу не менее 100 пользователей без существенного снижения скорости работы.

Масштабируемость. Архитектура программного обеспечения должна предусматривать возможность масштабирования – как вертикального (улучшение оборудования), так и горизонтального (добавление серверов). Это позволит расширять систему при росте компании и количества обрабатываемых документов.

Безопасность. СЭД должна соответствовать требованиям по информационной безопасности, включая:

- аутентификацию и авторизацию пользователей;
- шифрование хранимых и передаваемых данных;
- журналирование действий пользователей;
- защиту от SQL-инъекций, XSS и других распространённых уязвимостей;
- поддержку усиленной электронной подписи (УЭП) и средств криптографической защиты информации (СКЗИ), сертифицированных в РФ [18].

Удобство интерфейса (юзабилити).

Интерфейс должен быть интуитивно понятным, адаптированным под русскоязычную аудиторию, минимизировать количество действий, необходимых для выполнения повседневных задач. Важно предусмотреть краткое обучение, всплывающие подсказки и систему помощи.

Кроссплатформенность. Предпочтительно, чтобы система поддерживалась в браузере без необходимости установки дополнительного

программного обеспечения. Это позволит пользователям работать с СЭД как на стационарных ПК, так и на мобильных устройствах.

Реализация программного обеспечения всегда осуществляется в условиях множества внешних ограничений, которые могут существенно повлиять на технические и организационные решения [7].

Аппаратные ограничения. Система будет развёрнута на существующей IT-инфраструктуре торговой организации. В связи с этим необходимо учитывать текущие серверные мощности, каналы связи, а также технические характеристики пользовательских рабочих станций. Финансовые ограничения. Разработка и внедрение системы должны быть максимально экономически обоснованными. Недопустимы дорогостоящие лицензии, требующие ежегодных крупных затрат, а также избыточно сложные технологии, требующие привлечения внешних специалистов.

Временные ограничения. Разработка программного обеспечения должна быть завершена в установленные сроки, что определяет необходимость выбора проверенных архитектурных решений и применения гибкой методологии разработки (например, Agile-подхода) [12].

Нормативные ограничения. СЭД обязана соответствовать требованиям российского законодательства, включая:

- федеральный закон № 63-ФЗ «Об электронной подписи»;
- ГОСТ Р 7.0.8–2013, ГОСТ Р ИСО 15489–1–2007 и другие стандарты;
- инструкции по делопроизводству, принятые в организации [30].

Технические допущения. Предполагается, что пользователи обладают базовыми навыками работы с ПК. Кроме того, наличие стабильного доступа к внутренней сети и интернету рассматривается как обязательное условие корректного функционирования системы. Разработка требований к программному обеспечению является важнейшим этапом проектирования системы электронного документооборота и представляет собой основу для

последующего моделирования архитектуры, проектирования интерфейсов и реализации логики работы приложения.

На основании проведённого анализа были сформулированы:

- полный перечень функциональных требований, отражающих особенности документооборота в торговой организации;
- комплекс нефункциональных требований, включающих параметры надёжности, производительности, безопасности и удобства использования;
- основные ограничения, обусловленные внешними факторами: организационными, техническими и правовыми.

Соблюдение всех указанных требований позволит создать программное обеспечение, обладающее необходимым уровнем качества, соответствующее ожиданиям пользователей и адаптированное под специфику российского делопроизводства.

Глава 2 Проектирование программного обеспечения

2.1 Выбор методологии проектирования программного обеспечения

Одним из определяющих шагов в процессе проектирования является выбор методологии, то есть подхода, принципов и техник, которые будут положены в основу разработки архитектуры, моделей данных, компонентов и взаимодействия между ними. Методология проектирования определяет общий стиль мышления команды разработчиков, способствует систематизации процесса и служит ориентиром для принятия решений в условиях ограниченности ресурсов, времени и информации.

Современные условия разработки программного обеспечения, особенно в рамках корпоративных систем, таких как система электронного документооборота для торговой организации, предъявляют повышенные требования к гибкости, надёжности и адаптируемости. В условиях постоянно изменяющейся бизнес-среды, необходимости интеграции с внешними сервисами, а также учёта специфики нормативного регулирования в Российской Федерации, становится очевидной важность выбора методологии, способной учесть весь спектр требований.

Одним из наиболее эффективных и признанных подходов к описанию архитектуры и поведения информационной системы является использование унифицированного языка моделирования UML (Unified Modeling Language). Он позволяет представить структуру и логику системы в виде формализованных диаграмм, понятных как разработчикам, так и представителям бизнеса. UML охватывает все аспекты проектирования: от поведения отдельных компонентов до взаимодействия пользователей с системой [16].

Для проектирования системы электронного документооборота в торговой организации были выбраны следующие виды UML-диаграмм:

- диаграмма вариантов использования (Use Case Diagram);

- диаграмма классов (Class Diagram);
- диаграмма активности (Activity Diagram);
- диаграмма последовательности (Sequence Diagram).

Диаграмма вариантов использования (Use Case Diagram). Диаграмма вариантов использования позволяет зафиксировать основные сценарии взаимодействия пользователей (акторов) с системой. В рамках проектируемой СЭД были определены следующие акторы:

- сотрудник – внутренний пользователь системы, который создаёт, редактирует и отправляет документы на согласование, также он отслеживает статусы согласования и получает уведомления;
- руководитель – лицо, ответственное за согласование документов, может утверждать или отклонять документы, а также оставлять комментарии;
- поставщик – внешний контрагент, получающий документы, подписывающий их и возвращающий в систему, его роль критически важна при реализации внешнего документооборота;
- система – автоматический актор, генерирующий уведомления, изменяющий статусы и ведущий системный журнал.

Для каждого актора определены варианты использования (use cases), соответствующие бизнес-логике документооборота. Например, сотрудник может не только создать документ, но и инициировать его повторную отправку после возврата с замечаниями. Руководитель, в свою очередь, получает уведомления о поступивших на согласование документах, изучает их, оставляет комментарии, а затем принимает решение. Поставщик может взаимодействовать с системой через внешнюю интеграцию или личный кабинет.

Такое представление обеспечивает наглядность и позволяет заказчику увидеть полную картину функциональности будущей системы до этапа её

реализации. Общая диаграмма вариантов использования СЭД представлена на рисунке 1.

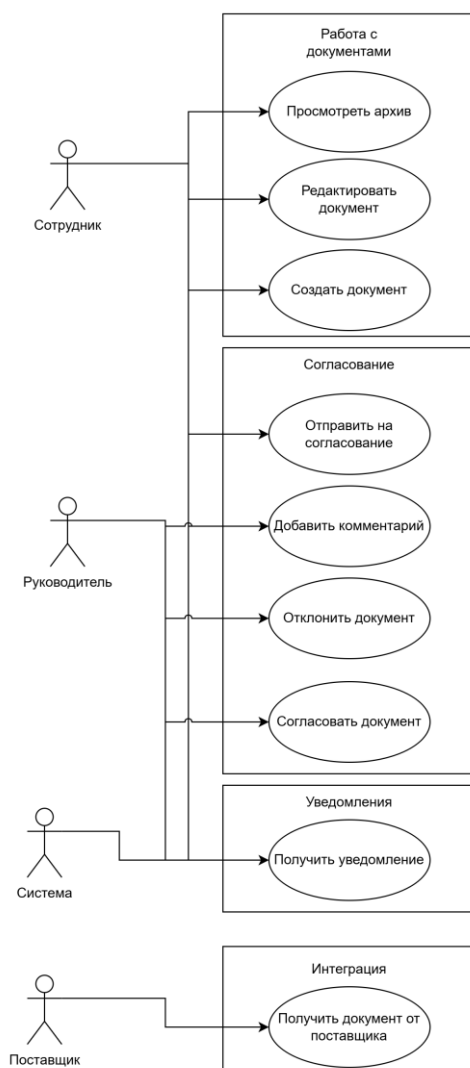


Рисунок 1 – Диаграмма вариантов использования

Диаграмма классов (Class Diagram). Диаграмма классов играет ключевую роль в описании структуры предметной области. Она позволяет определить основные сущности системы, их атрибуты и взаимосвязи [15].

На этапе проектирования системы электронного документооборота (СЭД) особое внимание уделяется точному моделированию бизнес-объектов,

таких как пользователь, документ, файл, согласование, комментарии и уведомления.

Основные классы системы:

- user – пользователь системы, каждый пользователь обладает уникальными идентификаторами, контактными данными и ролью, роли могут быть следующими: сотрудник, руководитель, поставщик;
- document – представляет документ, являющийся основным объектом в системе, документ проходит через стадии создания, редактирования, согласования, утверждения, подписания и архивирования;
- approval – отражает этап согласования, фиксирует кто согласовывает документ, когда и с каким результатом;
- comment – используется для добавления замечаний, предложений и пояснений к документу на любом этапе жизненного цикла;
- documentFile – представляет физическое вложение документа (например, PDF-файл), содержащее реквизиты и путь к файлу;
- notification – уведомление, отправляемое пользователям о событиях, связанных с документами;
- role – справочник ролей, определяющий доступ к функциональности;
- userRole – связь многие-ко-многим между пользователями и ролями.

Все классы связаны через отношения «один-ко-многим» или «многие-ко-многим», обеспечивая гибкость и масштабируемость модели.

Обоснование выбора классов:

- в системе предусмотрена поддержка множественных ролей, то есть один и тот же пользователь может быть и руководителем, и поставщиком в разных контекстах, это позволяет строить гибкую ролевую модель;
- документы часто имеют несколько вложений, особенно при взаимодействии с внешними контрагентами – сканы, проекты, подписанные версии;

- уведомления и комментарии являются критически важными для своевременного принятия решений и соблюдения сроков;
- механизм многоступенчатого согласования реализуется через сущность Approval, которая позволяет точно фиксировать, кто и когда согласовал, отклонил или перенаправил документ.

Диаграмма классов представлена на рисунке 2.

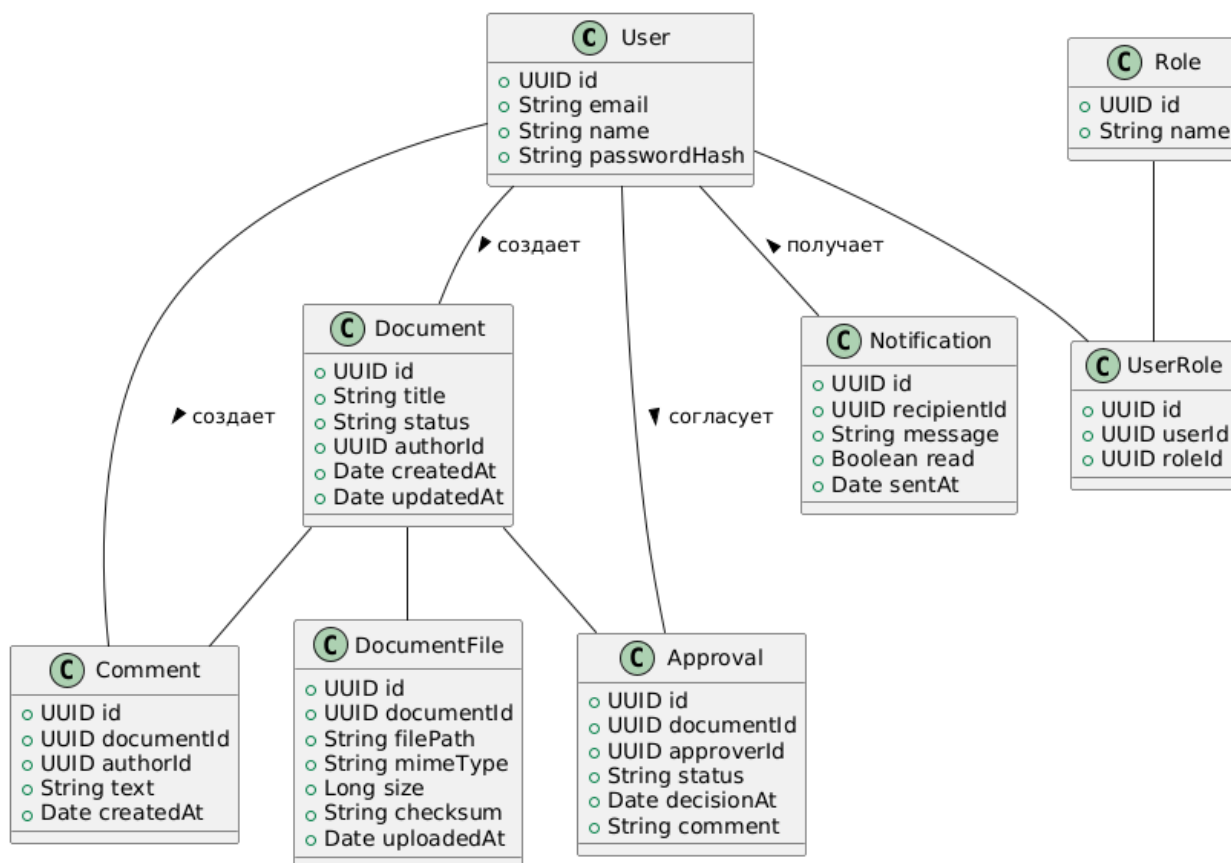


Рисунок 2 – Диаграмма классов

Каждый из классов в диаграмме выполняет строго определённую функцию. Например, сущность Approval позволяет реализовать гибкую модель маршрутов согласования, где документы могут согласовываться несколькими лицами в зависимости от бизнес-логики. В свою очередь,

Comment предоставляет возможность ведения деловой переписки внутри платформы, что особенно важно при возврате документа на доработку.

Диаграмма активности (Activity Diagram). Диаграмма активности предназначена для моделирования алгоритма бизнес-процесса в виде последовательности действий, ветвлений, условий и параллельных операций. В рамках проектируемой системы электронного документооборота диаграмма активности позволяет наглядно представить полный жизненный цикл обработки документа: от его создания до архивирования или отклонения.

Данная диаграмма охватывает все возможные состояния документа, отражает типичные действия сотрудников и руководителей, а также вмешательство системы в виде автоматических уведомлений. Это особенно важно, так как документооборот в торговой организации зачастую предполагает строгие регламенты, соблюдение сроков и ответственность участников за принятые решения [23].

Описание процесса:

- создание документа – сотрудник инициирует процесс, задаёт название, тип и содержание документа;
- редактирование – до момента отправки документ находится в статусе «Черновик», доступен для изменений;
- отправка на согласование – после подготовки документ переходит в состояние ожидания решения;
- согласование руководителем;
- подписание поставщиком – актуально для внешних документов (например, договоров), поставщик получает доступ, подписывает документ, возвращает его в систему;
- архивирование – завершённый документ попадает в архив для последующего хранения и доступа;
- уведомления – система автоматически уведомляет участников процесса о смене статуса, комментариях и необходимости действий.

Такой алгоритм позволяет минимизировать ошибки, повысить прозрачность действий и обеспечить прослеживаемость всех изменений. Диаграмма активности представлена на рисунке 3.



Рисунок 3 – Диаграмма активности

Диаграмма последовательности (Sequence Diagram). Диаграмма последовательности позволяет смоделировать взаимодействие между участниками процесса в виде упорядоченного обмена сообщениями. Это особенно полезно при проектировании модулей взаимодействия между

фронтом и бэкендом, между внутренними и внешними компонентами системы.

В рассматриваемой системе документооборота основными участниками последовательности являются:

- сотрудник – инициирует создание и отправку документа;
- система – принимает и обрабатывает действия пользователя, отправляет уведомления;
- руководитель – получает уведомление, просматривает документ и принимает решение;
- поставщик – получает документ на подпись при необходимости;
- база данных – условный участник, обеспечивающий постоянство данных и их хранение.

Сценарий:

- сотрудник создаёт документ;
- система сохраняет документ в статусе «Черновик»;
- сотрудник отправляет документ на согласование;
- система уведомляет руководителя;
- руководитель открывает документ, оставляет комментарий и принимает решение;
- в случае необходимости документ отправляется поставщику;
- поставщик подписывает и возвращает документ;
- система переводит документ в архив и уведомляет всех участников о завершении процесса.

Диаграмма последовательности представлена на рисунке 4.

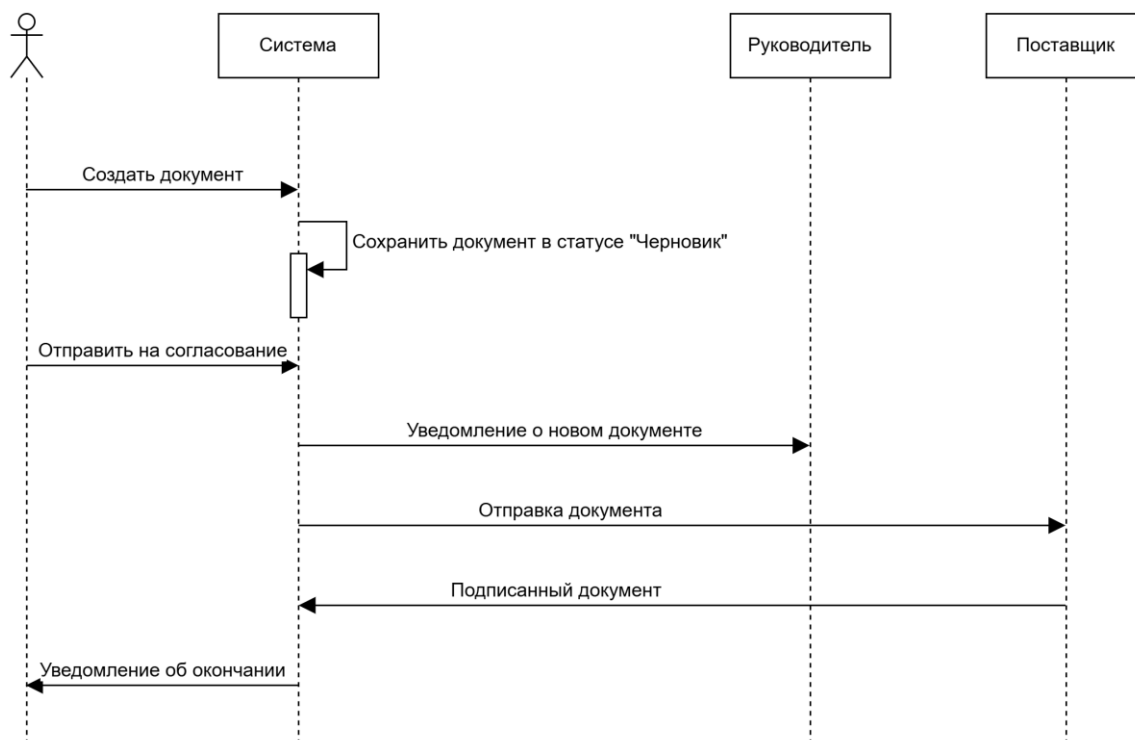


Рисунок 4 – Диаграмма последовательности

Использование UML-диаграмм позволило формализовать структуру и динамику проектируемой системы документооборота. Сценарии взаимодействия пользователей, бизнес-процессы согласования и логика обработки документов были подробно смоделированы при помощи наиболее уместных нотаций: диаграммы вариантов использования для идентификации требований, диаграммы классов для проектирования предметной области, диаграммы активности для описания рабочих процессов, и диаграммы последовательности для отражения коммуникации между участниками.

Такой подход обеспечивает чёткую и прозрачную архитектуру, улучшает взаимодействие между заказчиком и командой разработчиков, позволяет заранее выявить потенциальные узкие места и формализовать требования, минимизируя риски при последующей реализации. Внедрение системы, построенной на основе тщательно проработанной UML-модели, обеспечит

надёжную автоматизацию документооборота и адаптацию под требования современной торговой организации.

2.2 Логическое моделирование программного обеспечения

Логическое моделирование программного обеспечения является важнейшим этапом проектирования информационных систем, обеспечивающим основу для последующей реализации бизнес-логики, взаимодействия компонентов и хранения данных. В отличие от физического моделирования, на логическом уровне отсутствует привязка к конкретным механизмам хранения и реализации (например, к конкретной СУБД или форматам сериализации), что позволяет сосредоточиться на концептуальной структуре системы.

В рамках проекта, посвящённого разработке системы электронного документооборота (СЭД) для торговой организации, логическое моделирование позволяет определить состав сущностей, необходимых для обеспечения жизненного цикла документа, их атрибуты и взаимосвязи. Также формализуются ролевые взаимодействия между участниками процесса: обычными сотрудниками, руководителями отделов, администраторами, а в перспективе – внешними контрагентами.

Далее представлено подробное описание логической структуры проектируемой СЭД, включая анализ основных сущностей, их атрибутов, взаимосвязей и правил обработки данных.

Модель формирует фундамент для разработки надёжной, масштабируемой и управляемой системы, которая соответствует требованиям современной организации и позволяет автоматизировать рутинные операции, минимизировать ошибки и обеспечить юридическую значимость документооборота [13].

При разработке логической модели использовался реляционный подход, в основе которого лежит представление данных в виде набора взаимосвязанных таблиц. Этот подход был выбран по следующим причинам:

- высокая степень формализации;
- хорошая теоретическая проработанность (на базе теории множеств и логики предикатов);
- широкая поддержка в современных СУБД, таких как PostgreSQL;
- возможность строгого контроля целостности и согласованности данных;
- эффективная реализация запросов при помощи языка SQL.

Основным принципом моделирования стала нормализация структуры данных до 3-й нормальной формы (3НФ), что позволило устранить избыточность, минимизировать аномалии вставки, обновления и удаления, а также упростить сопровождение модели [11].

Дополнительно учитывались следующие факторы:

- возможность расширения модели (например, добавление новых типов документов, участников процесса или этапов согласования);
- поддержка аудита и версионирования;
- безопасность доступа к данным в зависимости от ролевой модели;
- соответствие внутренним регламентам организации и требованиям законодательства РФ в области документооборота.

На следующей диаграмме представлена визуализация логической структуры данных системы, отражающая связи между сущностями. Диаграмма подготовлена с использованием нотации UML и представлена на рисунке 5.

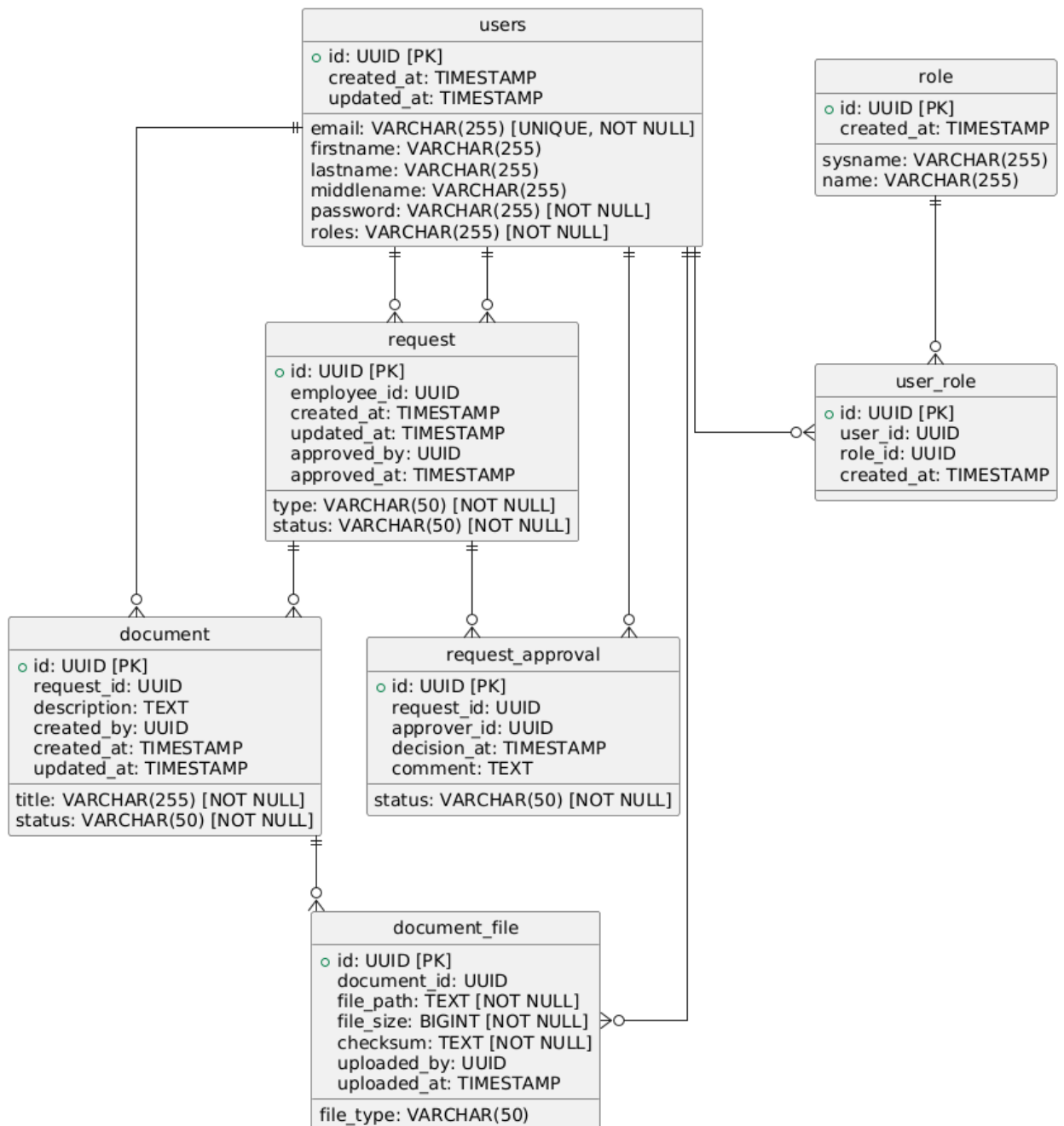


Рисунок 5 – Диаграмма логической структуры данных

Пользователь – ключевой субъект системы, от имени которого инициируются запросы, утверждаются документы, выполняются загрузки и другие действия. Структура таблицы users содержит следующие поля:

- id: первичный ключ UUID-формата, позволяющий гарантировать уникальность без риска коллизий даже в распределённой системе;

- email: уникальный идентификатор аутентификации;
- firstname, lastname, middlename: атрибуты, формирующие ФИО пользователя;
- password: хэшированный пароль, хранимый согласно требованиям безопасности;
- roles: строка, содержащая агрегированную информацию о ролях (реализация устаревшая, заменена сущностью user_role);
- created_at, updated_at: временные метки для отслеживания изменений.

Данная сущность обеспечивает базовую идентификацию пользователей и является основой для формирования персонализированного интерфейса и журналов событий.

Сущность role содержит перечень ролей, доступных в системе. Роли могут включать:

- обычного сотрудника (user);
- руководителя подразделения (supervisor);
- администратора (admin);
- кадрового специалиста (hr);
- юриста (lawyer);
- внешнего контрагента (external);
- системные роли (например, подпись или автоматическая проверка).

Сущность user_role реализует связь многие-ко-многим между пользователями и ролями. Это позволяет пользователю исполнять несколько ролей одновременно, например, быть одновременно сотрудником и руководителем, что нередко встречается в небольших отделах.

Запрос представляет собой бизнес-операцию, инициируемую сотрудником, требующую подписания или согласования. Основные атрибуты:

- employee_id: ID пользователя, создавшего запрос;
- type: тип запроса (например, «договор купли-продажи», «служебная записка»);

- status: состояние запроса (draft, pending, approved, rejected и т.д.);
- approved_by: ID пользователя, который утвердил запрос;
- approved_at: временная метка утверждения.

Поле status может принимать следующие значения:

- draft – черновик, редактируется;
- pending – отправлено на согласование;
- rejected – отклонено;
- approved – утверждено;
- completed – завершено.

Связи:

- employee_id → users.id;
- approved_by → users.id.

Каждый request может содержать один или несколько документов, отражающих суть запроса. Структура:

- request_id: внешний ключ, связывающий документ с запросом;
- title: наименование документа;
- description: краткое описание;
- status: статус обработки (на подписи, подписан, утвержден);
- created_by: автор документа;
- created_at, updated_at: временные метки.

Документы могут переходить из одного состояния в другое в процессе жизненного цикла, что должно фиксироваться в статусе.

Связи:

- request_id → request.id;
- created_by → users.id.

Документы в системе сопровождаются файлами, например, в формате PDF, DOCX, XLSX. Структура document_file:

- document_id: внешний ключ к document;

- file_type: MIME-тип (например, application/pdf);
- file_path: путь к файлу в файловой системе или хранилище;
- file_size: размер файла;
- checksum: контрольная сумма для верификации;
- uploaded_by: пользователь, загрузивший файл;
- uploaded_at: дата загрузки.

Это обеспечивает учет и контроль цифровых документов, а также возможность проверки подлинности и целостности.

Согласование (request_approval) отражает этапы рассмотрения запроса одним или несколькими уполномоченными лицами. Эта сущность позволяет реализовать как простые цепочки согласования (один утверждающий), так и сложные маршруты (многоступенчатое согласование).

Атрибуты:

- request_id: ссылка на исходный запрос;
- approver_id: пользователь, отвечающий за согласование;
- status: статус согласования (pending, approved, rejected, escalated и т.д.);
- decision_at: дата и время принятия решения;
- comment: пояснение к решению (например, причина отказа).

Эта таблица критически важна для отслеживания хода бизнес-процесса. Также она служит юридическим доказательством участия конкретного лица в рассмотрении документа.

Пример статусов:

- pending – ожидает рассмотрения;
- approved – согласован;
- rejected – отклонен;
- skipped – этап пропущен (например, автоматически);
- forwarded – передан другому сотруднику;
- escalated – передано вышестоящему руководителю.

Эта сущность поддерживает расширение до матрицы согласования, в которой учитываются различные критерии: тип документа, сумма, подразделение и т.д.

Система строится на четко определённых связях между сущностями. Ниже на рисунке 6 представлена текстовая ER-диаграмма логического уровня (ключевые связи).

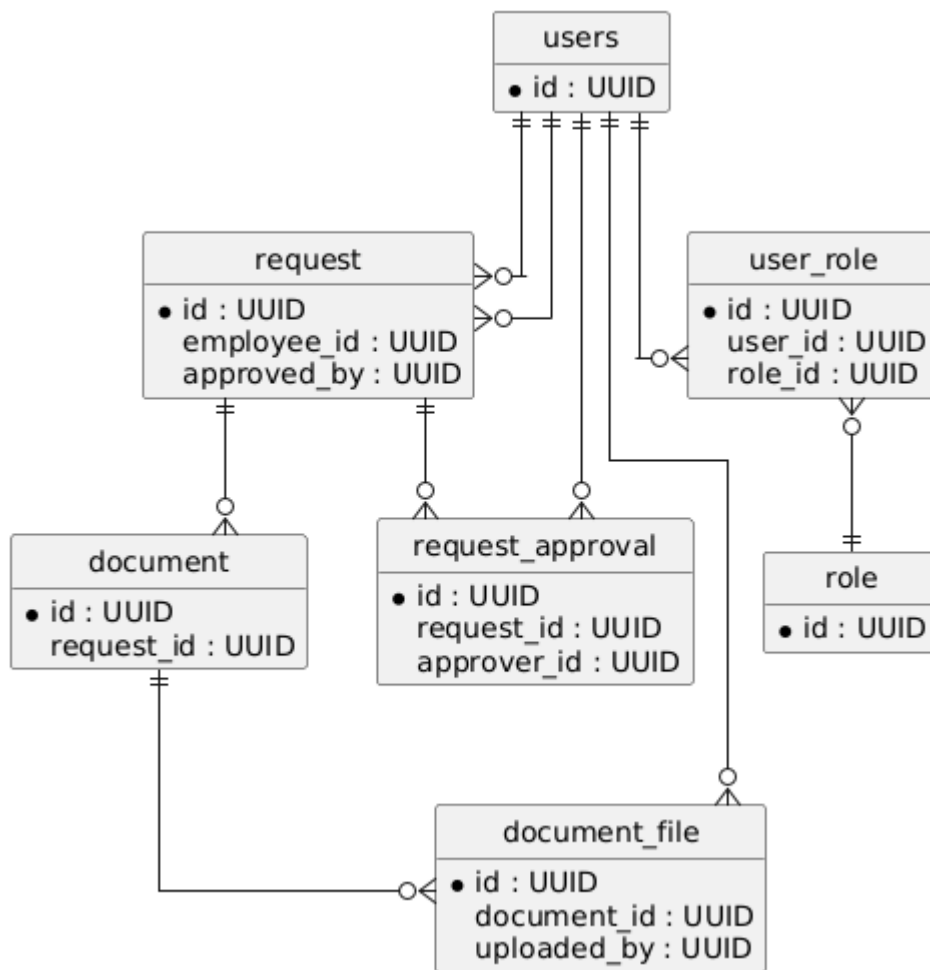


Рисунок 6 – ER-диаграмма логического уровня

Такая структура позволяет гибко обрабатывать отношения между сущностями и быстро масштабировать систему под новые потребности.

Основные сценарии использования и логическая трактовка.

Создание запроса:

- пользователь авторизуется;
- инициирует новый request, указывает тип (например, «договор поставки»);
- создает один или несколько document, прикладывает описания;
- загружает файлы к каждому документу через document_file;
- запрос сохраняется в статусе draft.

Отправка запроса на согласование:

- запрос переводится в статус pending;
- в request_approval создаются записи с идентификаторами согласующих;
- уведомления рассылаются соответствующим пользователям.

Согласование:

- согласующий переходит в раздел «На согласование»;
- изучает содержимое запроса, может оставить комментарий;
- принимает решение: approved, rejected;
- в request обновляется status, approved_by, approved_at при финальном согласовании.

Завершение:

- после всех этапов запроса статус меняется на completed;
- документы переводятся в архив, создается лог изменений;
- возможна интеграция с внешними системами (например, бухгалтерией или системой ЭЦП).

Предложенная модель поддерживает гибкое масштабирование:

- добавление новых ролей (например, юридический отдел, служба безопасности) – через role и user_role;
- поддержка многоуровневых маршрутов согласования – за счёт усложнения request_approval;
- интеграция с внешними пользователями – возможно расширение users с атрибутом external;

- версионирование документов – через отдельную сущность `document_version`;
- электронная подпись – можно добавить таблицу `document_signature`.

Кроме того, можно ввести логирование операций (аудит), например, в виде таблицы `audit_log` с указанием действия, исполнителя и времени.

Особое внимание в логической модели уделяется целостности:

- все связи построены на внешних ключах с каскадным удалением или `nullification`;
- контроль уникальности реализован через уникальные индексы (например, `email`);
- обязательные поля (`NOT NULL`) задают базовую валидацию;
- логика удаления заменяется «мягким удалением» через статус или флаг `deleted_at` (в перспективе);
- защита персональных данных (например, паролей) обеспечивается только хэшами без возможности восстановления.

Логическое моделирование программного обеспечения в рамках разработки системы электронного документооборота позволило:

- формализовать структуру бизнес-объектов и их взаимосвязи;
- обеспечить консистентность и расширяемость данных;
- задать основу для проектирования физической модели в PostgreSQL;
- предусмотреть поддержку ролевого доступа, многоуровневого согласования и электронных документов;
- спроектировать модульную, масштабируемую архитектуру, удобную для последующей реализации, сопровождения и интеграции.

Логическое моделирование представляет собой важный этап проектирования системы, в рамках которого формализуются сущности предметной области, их свойства, а также связи между ними. Результатом этого этапа является логическая модель данных, определяющая структуру

хранения, обработки и взаимодействия информации в пределах автоматизируемых процессов [10].

В рамках разработки системы электронного документооборота в торговой организации были выделены ключевые сущности, отражающие бизнес-процессы оформления и согласования документов, включая учет пользователей, маршрутизацию заявок, хранение прикрепленных файлов и распределение ролей.

Основные сущности и их описание:

- пользователи (users) – это сотрудники организации, которые взаимодействуют с системой, каждому пользователю может быть присвоено несколько ролей (например, «сотрудник», «руководитель отдела»), что позволяет реализовать гибкую систему авторизации и разграничения доступа;
- заявки (request) – основной объект инициирования документооборота, сотрудник создает заявку, указывает её тип, прикрепляет документы, а затем заявка отправляется на согласование;
- документы (document) – материалы, прикрепленные к заявке, они проходят этапы подписания и согласования, а документы могут быть многоуровневыми и содержать описание, статус и дату создания;
- файлы документов (document_file) – это физические представления документов (скан-копии, PDF-файлы и т.д.), которые прикрепляются к логическим документам, для обеспечения безопасности и контроля изменений используется контрольная сумма (checksum);
- согласование заявок (request_approval) – процесс, в котором участвуют руководители, здесь фиксируется, кто именно принял решение по заявке, когда это произошло и с каким результатом;
- роли (role) – определяют, какие действия доступны пользователю, например, одна роль может разрешать создание заявок, другая – их утверждение;

- связь пользователей и ролей (user_role) – обеспечивает возможность назначения нескольким пользователям разных ролей, а также динамическое изменение прав без модификации основной таблицы пользователей.

Представленные диаграммы позволяют наглядно представить структуру логической модели данных и отвечают требованиям современных торговых организаций, нуждающихся в юридически значимом документообороте с полной прозрачностью, отчетностью и безопасностью на всех этапах жизненного цикла запроса и документа.

2.3 Моделирование данных системы

Проектирование структуры данных является одной из ключевых стадий разработки программного обеспечения, особенно в системах, связанных с документооборотом, где требуется обеспечение целостности, достоверности и согласованности больших объемов информации. Этап моделирования данных определяет, каким образом информация будет храниться, обрабатываться и взаимодействовать внутри информационной системы, обеспечивая тем самым не только корректную реализацию бизнес-логики, но и масштабируемость, надежность и безопасность.

Предлагаем рассмотреть процесс моделирования данных системы электронного документооборота для торговой организации. Затем описать принципы нормализации данных, обосновать выбор ключевых сущностей, их взаимосвязи, а также рассмотреть особенности применения реляционной модели с использованием системы управления базами данных PostgreSQL.

Моделирование данных в рамках разработки информационной системы преследует цель создания логической и физической структуры хранения данных, которая максимально полно и непротиворечиво отражает предметную область.

Под моделированием данных понимается процесс формализации информации, циркулирующей в предметной области, в виде совокупности взаимосвязанных сущностей, атрибутов и отношений между ними. В процессе моделирования используется набор методологических подходов, таких как ER-моделирование (Entity-Relationship), нормализация данных, построение UML-диаграмм и логических схем баз данных [9].

При моделировании данных системы электронного документооборота ключевую роль играет правильное представление взаимодействия между пользователями, заявками, документами и процессами согласования. Для описания логической структуры данных была выбрана классическая ER-модель, реализуемая на уровне реляционной базы данных PostgreSQL.

Основные сущности и их взаимосвязи.

В процессе анализа требований к системе и предметной области были выделены следующие ключевые сущности:

- пользователь (users);
- заявка (request);
- документ (document);
- файл документа (document_file);
- согласование заявки (request_approval);
- роль (role);
- привязка пользователя к роли (user_role).

Каждая из перечисленных сущностей играет определенную роль в функционировании системы, и их взаимодействие отражает реалии документооборота в торговой организации.

Сущность users представляет зарегистрированных пользователей системы. В контексте документооборота пользователи могут выступать как инициаторами заявок, так и их согласующими лицами. Каждый пользователь может обладать несколькими ролями (например, быть как обычным сотрудником, так и руководителем), что реализовано через связующую

таблицу `user_role`. Атрибуты сущности включают уникальный идентификатор, адрес электронной почты, ФИО, хеш пароля, а также дату создания и обновления записи. Поле `roles`, как обобщенное представление роли, по сути, устарело и заменено более гибкой моделью через отдельную таблицу.

Сущность `request` реализует представление заявки, инициированной пользователем. Заявка содержит тип (например, «договор купли-продажи»), статус, идентификаторы инициатора и утверждающего лица. Это позволяет отслеживать весь жизненный цикл заявки: от черновика до финального утверждения. Заявки напрямую связаны с прикрепленными к ним документами (`document`) и процессами согласования (`request_approval`).

Сущность `document` представляет собой документы, прикрепленные к заявкам. Каждый документ связан с конкретной заявкой, имеет наименование, описание, статус и идентификатор пользователя, его создавшего. Данная сущность позволяет хранить не только метаданные о документе, но и управлять версионностью и статусами обработки.

Сущность `document_file` содержит информацию о файлах, прикрепленных к документам. Каждый файл имеет тип, путь к файлу, размер, контрольную сумму и дату загрузки. Такая структура обеспечивает возможность хранения нескольких файлов на один документ, контроля целостности (через `checksum`), а также отслеживания, кто именно загрузил тот или иной файл.

Сущность `request_approval` реализует механизм согласования заявок. Она хранит информацию о каждом согласующем, статусе согласования (например, «утверждено», «отклонено», «ожидание»), дате принятия решения и комментарии. Это позволяет реализовать сложные маршруты согласования, а также проводить аудит решений.

Сущности `role` и `user_role` реализуют распределение прав и обязанностей в системе. Таблица `role` содержит справочник доступных ролей, в то время как `user_role` отображает, какие пользователи какими ролями обладают. Это

позволяет легко масштабировать систему, добавлять новые роли и обеспечивать ролевую модель управления доступом.

Процесс нормализации базы данных направлен на устранение избыточности, обеспечение логической согласованности и предотвращение аномалий при вставке, обновлении или удалении данных. В рамках разрабатываемой системы были соблюдены основные нормы нормализации до третьей нормальной формы включительно.

Первая нормальная форма.

Все таблицы системы спроектированы таким образом, что каждый атрибут содержит атомарные значения, а каждая строка таблицы уникальна и идентифицируется первичным ключом. Например, в таблице users каждый пользователь имеет уникальный идентификатор, а атрибуты, такие как email, firstname, lastname, являются неделимыми.

Вторая нормальная форма.

Каждый неключевой атрибут зависит от всего первичного ключа. Это особенно актуально для таблиц со сложными (составными) первичными ключами, которых, однако, в данной системе не предполагается. Поэтому соблюдение 2NF достигается за счет корректного проектирования отношений «один ко многим», например, между users и request.

Третья нормальная форма.

Каждый неключевой атрибут не зависит транзитивно от первичного ключа. Например, информация о ролях пользователей вынесена в отдельную таблицу role, а связь реализуется через user_role, что предотвращает дублирование данных и упрощает управление ролевой моделью.

В системе активно применяются внешние ключи для обеспечения связности данных между таблицами. Например, request.employee_id ссылается на users.id, document.request_id – на request.id, а request_approval.approver_id – на users.id. В PostgreSQL для этого применяются ограничения FOREIGN KEY,

которые гарантируют, что ссылки указывают только на существующие записи, предотвращая тем самым появление «висячих ссылок».

Все каскадные операции (удаление, обновление) были спроектированы с осторожностью. Удаление пользователей, участвовавших в процессе согласования или создания заявок, блокируется за счёт ограничения ON DELETE RESTRICT. Это позволяет сохранить историю и юридическую значимость документов, связанных с конкретным пользователем.

Сценарии использования и трансляция бизнес-процессов в структуру данных. Важным этапом моделирования данных является трансляция типичных бизнес-процессов организации в конкретные реляционные структуры [8]. Рассмотрим, как через предложенную модель реализуются ключевые сценарии документооборота.

Регистрация заявки.

Когда сотрудник инициирует новую заявку, в таблицу request добавляется новая запись с указанием employee_id и типом заявки. При этом в таблицу document может быть добавлено несколько документов, каждый из которых связан с заявкой через request_id.

Загрузка документов и файлов.

Каждый документ может включать несколько файлов (например, PDF, сканы, сопроводительные листы). Эти файлы хранятся в document_file, а привязка осуществляется через document_id. Указание пользователя, загрузившего файл (uploaded_by), необходимо для внутренней отчетности и отслеживания действий.

Согласование заявок.

Для маршрутизации заявок по согласующим используется таблица request_approval. Каждый этап согласования — это отдельная запись с указанием пользователя (approver_id), статусом и временем принятия решения. Такая реализация дает гибкость: можно строить цепочки

согласования, задавать разные маршруты в зависимости от типа заявки или организационной структуры.

Управление доступом.

Система реализует ролевую модель управления доступом через связку users, user_role и role. Таким образом, добавление нового типа пользователя (например, аудитора) требует лишь создания новой роли и назначения ее соответствующим пользователям. Это повышает гибкость и масштабируемость системы.

Физическая реализация и производственные решения.

При выборе СУБД для хранения данных был сделан выбор в пользу PostgreSQL, как надежного, функционального и активно развивающегося решения. Среди причин выбора стоит отметить:

- поддержка строгой ссылочной целостности;
- наличие типов UUID и массивов;
- расширения для полнотекстового поиска и JSON;
- высокая скорость обработки запросов на выборку и агрегацию.

Особое внимание было уделено типам данных. Например, идентификаторы всех сущностей реализованы через UUID, что предотвращает угадывание ID, повышает безопасность и упрощает распределенное масштабирование. Для хранения текстов используется TEXT, для дат – TIMESTAMP, а для хешей паролей – VARCHAR с достаточной длиной.

Индексация и производительность.

Для повышения эффективности работы с данными в системе предусмотрена реализация целевого индексирования. Индексы особенно актуальны для колонок, которые участвуют в операциях поиска, фильтрации и соединения таблиц. Например:

- для таблицы users индексируется поле email, что ускоряет аутентификацию;

- в `request` – поле `employee_id`, позволяющее быстро получать заявки по пользователю;
- в `request_approval` – поле `approver_id` для извлечения всех заявок, находящихся на согласовании конкретного сотрудника;
- для `document_file` – поле `document_id`, ускоряющее доступ к файлам конкретного документа.

Кроме того, составные индексы могут быть добавлены для ускорения сложных запросов с фильтрацией по нескольким полям, например: (`status`, `created_at`) в таблице `request`.

При проектировании структуры также учтено снижение риска чрезмерного индексирования, которое может повлиять на производительность операций вставки и обновления. Индексы создаются по мере необходимости и на основе анализа профиля запросов в реальной эксплуатации.

Одним из важнейших требований при проектировании архитектуры базы данных является обеспечение безопасности как на логическом, так и на физическом уровне.

Пароли пользователей хранятся в зашифрованном виде, с использованием стойких хеш-функций (например, Argon2 или bcrypt). В системе отсутствует возможность восстановления пароля в открытом виде — только сброс по подтверждённому e-mail.

Для всех ключевых таблиц предусмотрены поля `created_at` и `updated_at`, что позволяет отслеживать изменения записей. Дополнительно, в таблицах `document_file` и `request_approval` присутствуют поля, фиксирующие, кто выполнил загрузку или согласование.

В будущем может быть внедрена отдельная таблица аудита с логированием операций изменения и удаления, а также аудит доступа к документам.

Права доступа к данным реализуются не только на уровне приложения, но и на уровне модели данных. Например:

- структура ролей и привязок (role, user_role) позволяет централизованно управлять полномочиями;
- запросы данных строятся с учётом ограничений видимости (например, менеджер видит только заявки своего отдела).

Дополнительно планируется реализация политик на уровне row-level security в PostgreSQL, что даст возможность декларативно ограничивать доступ к строкам таблиц в зависимости от пользователя.

Масштабируемость и подготовка к росту нагрузки.

Система проектируется с прицелом на будущий рост – как по числу пользователей, так и по объёму обрабатываемых данных. В связи с этим, при моделировании были приняты следующие меры:

- использование UUID в качестве ключей позволяет избежать конфликтов при распределённой архитектуре;
- отказ от хранения файлов в базе данных (используется ссылка file_path), что снижает нагрузку на СУБД;
- возможность разделения базы на логические схемы или физически – на несколько баз данных при необходимости масштабирования;
- использование денормализованных представлений (VIEW) для отчетности, не влияющих на основную модель;
- поддержка архивации и партиционирования данных (например, старых заявок) для оптимизации производительности.

Резервное копирование и отказоустойчивость.

Поскольку система хранит юридически значимые документы и критически важные данные, особое внимание уделяется вопросам резервного копирования и восстановления. Предусмотрено:

- регулярное создание бэкапов базы данных;
- отдельное резервирование файлов из хранилища (например, S3, FTP или локального диска);

- возможность репликации БД для повышения отказоустойчивости и быстрого переключения в случае сбоя.

В системе также предусмотрен журнал транзакций, позволяющий восстанавливать состояние до определённого момента времени.

Поддержка интеграций.

Проектируемая модель данных предусматривает интеграцию с внешними системами, такими как:

- корпоративная почта (для отправки уведомлений о новых заявках и решениях);
- системы кадрового учета (для автоматического создания пользователей и связей);
- электронная подпись (связанная с документами);
- архивные и юридические системы (например, госреестры или бухгалтерское ПО).

Для этого модель предусматривает наличие технических полей (`external_id`, `signature_hash`, `source_system`), которые могут быть добавлены в релевантные таблицы при необходимости.

В результате моделирования данных была разработана структурированная, логичная и масштабируемая модель реляционной базы данных, которая адекватно отражает бизнес-процессы торговой организации. Архитектура БД поддерживает ключевые функции электронного документооборота: создание заявок, хранение документов и файлов, согласование, управление пользователями и ролевой моделью.

Соблюдение принципов нормализации, обеспечение безопасности, поддержка масштабируемости и интеграций с внешними системами делает предложенную модель надёжным фундаментом для дальнейшей разработки, внедрения и эксплуатации системы электронного документооборота в корпоративной среде.

Глава 3 Реализация и тестирование программного обеспечения

3.1 Реализация программного обеспечения

Предлагаем начать с архитектурного подхода и общей структуры проекта. В ходе реализации электронного документооборота для торговой организации был выбран фреймворк Spring Boot, предоставляющий обширный набор инструментов для быстрого и структурированного создания корпоративных приложений. Архитектура проекта построена в соответствии с принципами модульности и разделения ответственности, что способствует масштабируемости, сопровождению и тестированию программного кода. Основные модули включают регистрацию пользователей, создание и обработку заявок, подписание и архивирование документов, применение штампов и визуализацию состояния обработки. Каждый модуль реализует строго ограниченную предметную область, что упрощает поддержку и развитие системы.

Проект организован в виде многомодульного Spring Boot-приложения с использованием архитектурного подхода, предполагающего разделение на уровни:

- контроллеры (controller) – обработка HTTP-запросов, валидация входных данных и возврат результатов пользователю;
- сервисный уровень (service) – реализация бизнес-логики;
- репозиторийный слой (repository) – доступ к данным через Spring Data JPA;
- модели данных (model, entity, dto) – классы, отражающие доменные объекты и структуры передачи данных;
- конфигурации (config) – настройки безопасности, работы с файлами, авторизации и прочих системных аспектов.

Такое разделение повышает читаемость и упрощает модульное тестирование компонентов. В рамках разработки были реализованы как REST-контроллеры для взаимодействия с клиентской частью, так и вспомогательные компоненты – фильтры безопасности, обработчики ошибок и классы для сериализации/десериализации данных.

Выбор технологий и библиотек.

Для реализации системы были использованы современные и проверенные временем технологии Java-экосистемы:

- Java 17 – современная LTS-версия языка с поддержкой нововведений (например, sealed classes, records);
- Spring Boot – основной фреймворк для создания REST-приложений;
- Spring Security – обеспечение безопасности и контроль доступа;
- JPA (Hibernate) – работа с базой данных на уровне объектов;
- Flyway – управление миграциями базы данных;
- JWT (JJWT) – реализация авторизации через токены;
- MinIO SDK – интеграция с хранилищем объектов для документов;
- Apache PDFBox и Bouncy Castle – работа с PDF и цифровой подписью документов;
- Lombok – сокращение шаблонного кода (геттеры, сеттеры, конструкторы);
- PostgreSQL – в качестве основной реляционной СУБД;
- JUnit и Mockito – для модульного и интеграционного тестирования.

Каждая из указанных библиотек играет важную роль в реализации функциональных и нефункциональных требований системы. Например, использование MinIO позволило абстрагироваться от конкретных поставщиков облачного хранилища и применить объектное хранение для архивируемых документов. JWT обеспечивает удобную и масштабируемую модель аутентификации и авторизации, особенно в условиях REST-архитектуры.

Реализация ключевых модулей. К основным модулям относятся: модуль регистрации и авторизации, обработка заявок и документов, реализация согласования.

Модуль регистрации и авторизации.

Регистрация пользователей реализована через отдельный контроллер `AuthController`, обеспечивающий конечные точки для регистрации новых пользователей и получения токена. При регистрации происходит валидация входных данных, хеширование пароля (с использованием `BCryptPasswordEncoder`) и сохранение данных в таблицу `users`.

Аутентификация пользователей построена на основе JWT-токенов. После успешного входа клиент получает JWT, который используется в заголовках последующих запросов. Токены содержат информацию о пользователе, срок действия и цифровую подпись. Для валидации токенов используется фильтр `JwtAuthenticationFilter`, интегрированный в цепочку `Spring Security`.

Обработка заявок и документов.

Основной функционал системы связан с созданием и обработкой заявок. Пользователь может создать заявку, загрузить документ, отредактировать данные и отправить её на согласование. В зависимости от текущего статуса заявки (`Draft`, `OnReview`, `Rejected`, `Approved`), система разрешает или запрещает определённые действия. Это поведение реализовано через бизнес-логику в `RequestService`.

Каждая заявка связывается с загруженным PDF-документом, который хранится в `MinIO`. Для подписания документа используется библиотека `PDFBox` совместно с `Bouncy Castle`: реализована логика наложения цифровой подписи и синей печати на файл, после чего он сохраняется в неизменяемом виде.

Реализация согласования.

Процесс согласования реализован с участием руководства и сторонних контрагентов. Система позволяет накладывать подписи, оставлять комментарии, возвращать документы на доработку или утверждать их. Для этого был реализован модуль комментариев, связанный с заявками, с возможностью редактирования при определённых статусах. Пользователи получают уведомления о действиях через email или интеграции с внешними системами (если настроено).

Далее рассмотрим конфигурацию безопасности и авторизации. Безопасность является одним из ключевых требований в разработке электронных систем документооборота, особенно если речь идёт о хранении, подписании и передаче юридически значимых документов. В рамках реализуемого программного обеспечения защита обеспечивается на нескольких уровнях: от разграничения прав доступа до шифрования данных в передаче и хранении.

Использование Spring Security позволило гибко настроить механизм аутентификации и авторизации пользователей. Все REST-эндпоинты защищены через механизмы фильтрации запросов с использованием `OncePerRequestFilter`, проверяющего наличие и валидность JWT-токена. При его отсутствии или истечении срока действия пользователь получает соответствующий ответ с кодом 401.

Реализация кастомного `UserDetailsService` позволила настроить загрузку информации о пользователе из базы данных, включая роли и полномочия. Это стало особенно важно для разграничения прав между сотрудниками, HR-отделом и руководителями. Благодаря аннотациям `@PreAuthorize` и `@Secured`, удалось настроить точечную защиту методов сервисного слоя.

Также произведена конфигурация CORS и HTTPS. Система нацелена на эксплуатацию как во внутренней корпоративной сети, так и через веб-интерфейс. Поэтому необходимо было учесть параметры настройки CORS (Cross-Origin Resource Sharing) для работы с фронтендом, расположенным на

другом домене. Это реализовано через глобальный CorsConfigurationSource, разрешающий конкретные домены, методы и заголовки.

Дополнительно система предполагает возможность работы по HTTPS с использованием сертификатов. Все чувствительные данные, включая JWT и загрузку документов, передаются в зашифрованном виде.

Следующим рассмотрим работу с MinIO и подпись документов. Для хранения документов в формате PDF используется MinIO – высокопроизводительное распределённое объектное хранилище с интерфейсом, совместимым с Amazon S3. Это позволяет легко масштабировать систему хранения и использовать преимущества отказоустойчивости. В рамках системы каждый загруженный документ сохраняется в уникальном бакете, связанный с идентификатором заявки.

Загрузка и извлечение файлов осуществляется через MinIO SDK. Документы шифруются на уровне бакета и имеют ограниченный доступ – только уполномоченные пользователи (авторы заявки, согласующие, администраторы) могут их просматривать или скачивать. Для обеспечения однозначной идентификации применяется UUID, исключая коллизии имён файлов.

Цифровая подпись и синие печати. Одним из важнейших аспектов реализуемой системы является возможность подписания документов. Используются библиотеки Apache PDFBox для модификации PDF и Bouncy Castle как криптографический провайдер. Подпись создаётся на основе закрытого ключа, хранящегося в зашифрованном виде. Перед подписанием система проверяет пользователя и его полномочия.

В дополнение к подписи реализовано наложение синей печати (blue stamp), выполненное с помощью графической вставки изображения в заданную область PDF-файла. Это делает итоговый документ юридически значимым и визуально понятным для дальнейшей обработки или печати.

Все подписанные документы перезаписываются в MinIO в неизменяемом виде (опционально можно активировать versioning).

Примеры реализации компонентов можно увидеть ниже. Например, контроллер для создания заявки (RequestController) представлен на рисунке 7.

```
@PostMapping("/requests")
@PreAuthorize("hasRole('EMPLOYEE')")
public ResponseEntity<RequestResponseDto> createRequest(@RequestBody
@Valid RequestCreateDto dto) {
    RequestResponseDto result = requestService.createRequest(dto);
    return ResponseEntity.status(HttpStatus.CREATED).body(result);
}
```

Рисунок 7 – Контроллер для создания заявки

Сервис для создания заявки (RequestService) представлен на рисунке 8.

```
public RequestResponseDto createRequest(RequestCreateDto dto) {
    Request request = new Request();
    request.setTitle(dto.getTitle());
    request.setAuthor(currentUserProvider.getCurrentUser());
    request.setStatus(RequestStatus.DRAFT);
    requestRepository.save(request);
    return mapper.toResponseDto(request);
}
```

Рисунок 8 – Сервис для создания заявки

Репозиторий для создания заявки (RequestRepository) представлен на рисунке 9. Реализация интерфейса в данном случае генерируется автоматически с помощью Spring Boot Data JPA.

```
public interface RequestRepository extends JpaRepository<Request,
UUID> {
    List<Request> findByUser(User user);
}
```

Рисунок 9 – Репозиторий для создания заявки

Такая структура упрощает тестирование и соблюдает принципы SOLID.

Давайте более подробно рассмотрим реализацию таких модулей как:

- модуль регистрации и аутентификации пользователей;
- модуль создания и обработки заявок;
- модуль согласования и подписания документов.

Реализация модуля регистрации и аутентификации пользователей.

Модуль регистрации и аутентификации пользователей является ключевым компонентом системы, обеспечивающим как безопасность, так и удобство взаимодействия с программным обеспечением. Его реализация осуществлялась с использованием Spring Security – мощного и гибко настраиваемого фреймворка, интегрируемого в экосистему Spring Boot. Для хранения учетных записей и авторизационной информации использовалась реляционная СУБД PostgreSQL, обеспечивающая надежность и масштабируемость.

Процесс регистрации пользователей предусматривает передачу формы с данными (ФИО, адрес электронной почты, пароль, роль и т. д.) на backend, где данные проходят валидацию. При помощи библиотеки Bean Validation (Hibernate Validator) проверяется корректность формата электронной почты, длина пароля и другие параметры. После успешной валидации данные поступают в сервис регистрации, который хеширует пароль с использованием алгоритма BCrypt и сохраняет пользователя в базе данных.

Для аутентификации используется механизм JWT (JSON Web Token). После успешной авторизации пользователю выдается токен, содержащий информацию о его идентификаторе, роли и сроке действия. Этот токен при каждом последующем запросе прикрепляется в HTTP-заголовке Authorization, и его подлинность проверяется посредством секретного ключа. Такая архитектура позволяет реализовать stateless-аутентификацию, избавляя систему от необходимости хранить сессионные данные на сервере.

В целях повышения безопасности также реализована поддержка черного списка токенов и автоматическое их истечение, а при необходимости – принудительное завершение сессии. Особое внимание уделялось защите от атак типа CSRF, XSS и brute-force.

Модуль создания и обработки заявок. Следующим важным компонентом системы является модуль обработки заявок. Он позволяет сотрудникам создавать, редактировать и отправлять заявки на рассмотрение. Каждая заявка проходит через определенный жизненный цикл, включающий состояния: «Черновик», «На рассмотрении», «Отклонена», «Одобрена» и «Завершена».

Создание заявки осуществляется через REST-интерфейс, предоставляющий эндпоинт POST /requests. На этапе создания пользователю необходимо заполнить форму, содержащую описание запроса, категорию, вложения (документы), а также указать предполагаемых согласующих. Backend обрабатывает входные данные, валидирует их и сохраняет в базе, создавая сущности Request и RequestStatusHistory, последняя из которых служит для отслеживания изменений состояния заявки.

Важно, что редактирование заявки разрешено только в состоянии «Черновик» или в случае, если заявка была возвращена на доработку ответственным лицом. Эта логика реализуется на уровне сервисного слоя, где производится проверка текущего состояния заявки перед её обновлением.

Модуль согласования и подписания документов. Ключевым элементом системы электронного документооборота является модуль согласования и подписания документов. Он реализует процесс поэтапного согласования заявок и прикрепленных к ним документов различными участниками: начальниками отделов, юристами и другими заинтересованными сторонами.

Процесс согласования выстроен на основе цепочки действий. При создании заявки пользователь указывает последовательность согласующих лиц, которая сохраняется в отдельной таблице `approval_steps`. Каждый шаг согласования может находиться в одном из состояний: «Ожидается», «Одобрено», «Отклонено», «Пропущено».

Подписание документов реализуется с использованием библиотеки Bouncy Castle и Java Cryptography Architecture (JCA). Каждый документ, прикрепленный к заявке, может быть подписан электронной подписью (ЭП), созданной на основе закрытого ключа сотрудника. Проверка подлинности подписей осуществляется при открытии документа, благодаря чему достигается юридическая значимость процесса.

Для создания и наложения штампов (так называемой «синей печати») используется Apache PDFBox. Эта библиотека позволяет работать с PDF-документами на низком уровне, что дало возможность автоматически добавлять визуальный элемент печати, а также подписи на последний лист документа. Пример такой подписи представлена на рисунке 10.

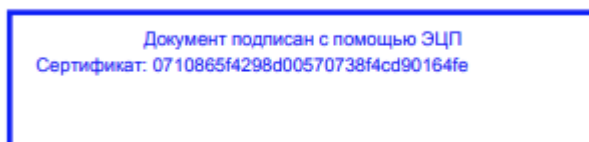


Рисунок 10 – Штамп с информацией об ЭЦП

Система реализует хранение оригиналов и подписанных копий документов в MinIO – объектном хранилище, совместимом с Amazon S3. Это решение позволило упростить управление бинарными файлами и повысить отказоустойчивость системы хранения.

Далее рассмотрим меры, принятые для повышения безопасности приложения. Безопасность является одним из ключевых аспектов разработки электронного документооборота (EDMS). Для защиты данных пользователей и документов в системе были реализованы несколько уровней безопасности, включая аутентификацию, авторизацию, шифрование данных и защиту от различных типов атак.

Аутентификация и авторизация. Для аутентификации в системе использовалась JWT (JSON Web Token) технология, которая позволяет обеспечивать безопасность пользовательских сессий с помощью токенов. Токен генерируется на сервере при успешной авторизации пользователя (по логину и паролю) и отправляется на клиентскую сторону для использования при последующих запросах.

На серверной стороне используется Spring Security для защиты конечных точек API. Для каждого запроса проверяется наличие и корректность JWT в заголовках (Authorization). Все маршруты, связанные с административными операциями или конфиденциальными данными, защищены с помощью соответствующих ролей и разрешений.

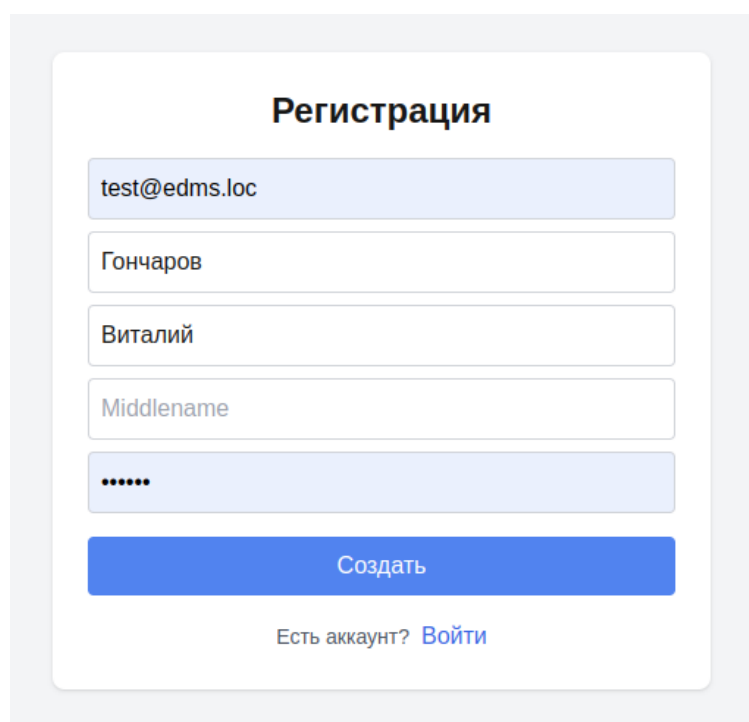
Шифрование данных. Вся конфиденциальная информация, включая персональные данные пользователей и содержимое документов, шифруется с использованием современных стандартов безопасности. Все передаваемые данные между клиентом и сервером защищены с помощью протокола HTTPS, что предотвращает перехват информации злоумышленниками в процессе передачи данных через сеть.

Защита от атак. Для защиты от распространённых атак, таких как SQL-инъекции, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) и других, в приложении были реализованы различные меры:

- SQL-инъекции предотвращены использованием подготовленных запросов (prepared statements) в JPA и Hibernate;
- XSS-атаки предотвращаются с помощью экранирования всех входных данных, поступающих от пользователей;
- CSRF-атаки защищены с использованием CSRF-токенов для каждого POST-запроса, что предотвращает возможность выполнения несанкционированных действий через фальсифицированные формы.

Далее рассмотрим модули авторизации, регистрации, создания запросов, подписания документов.

Рассмотрим регистрацию и аутентификацию пользователей. Для тестирования успешного кейса при регистрации вводим корректные данные. Процесс представлен на рисунке 11.



The image shows a registration form with the following fields and elements:

- Header:** "Регистрация" (Registration)
- Email:** "test@edms.loc"
- Surname:** "Гончаров" (Goncharov)
- First Name:** "Виталий" (Vitaliy)
- Middlename:** "Middlename"
- Password:** "*****" (masked)
- Submit Button:** "Создать" (Create)
- Footer:** "Есть аккаунт? Войти" (Have an account? Log in)

Рисунок 11 – Ввод корректных данных при регистрации

После успешной регистрации пользователь становится автоматически аутентифицированным и авторизованным для дальнейших действий. Далее пользователь попадает на страницу со всеми запросами. Эту страницу можно увидеть на рисунке 12.

id	Тип	Статус	Дата создания	Дата обновления	Утверждено	Дата утверждения	Действия
871297d5-254f-4e4a-a1a2-555546349128	Договор купли-продажи товаров	Черновик	2025-05-03T00:30:59.61092	N/A	N/A	N/A	Изменить Удалить
bd97bb8a-9571-47f6-8778-73d08ee6f26a	Договор купли-продажи товаров	Черновик	2025-05-19T16:41:41.150899	N/A	N/A	N/A	Изменить Удалить

Рисунок 12 – Страница со списком всех запросов

Если у пользователя уже есть созданный аккаунт, то он может воспользоваться формой входа. При корректно введенных данных, пользователь будет перенаправлен на список запросов. Успешный вариант авторизации представлен на рисунке 13.

Авторизация

Нет аккаунта? [Зарегистрироваться](#)

Рисунок 13 – Вариант успешной авторизации

Для создания запроса пользователь должен нажать на кнопку «Создать» на странице запросов. Местоположение кнопки представлено на рисунке 14.

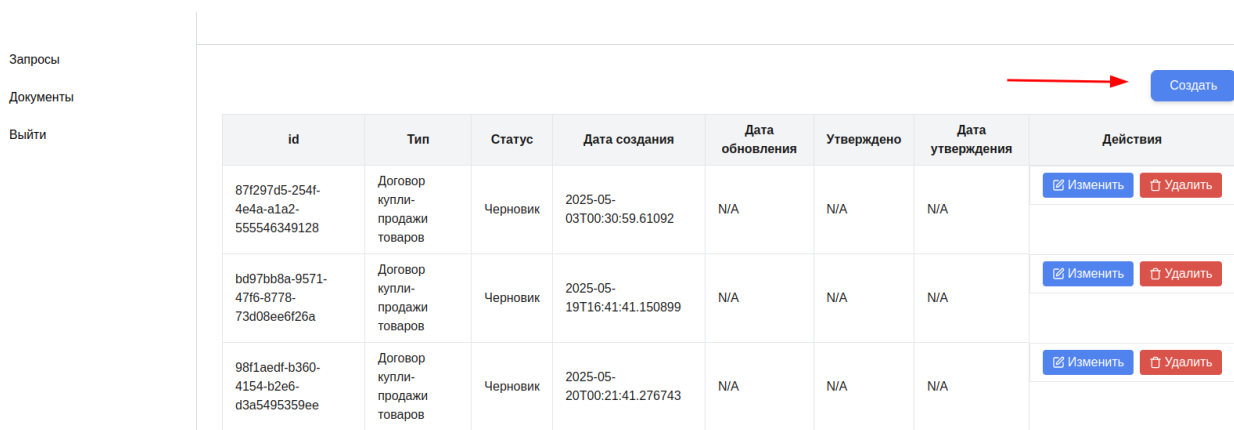


Рисунок 14 – Нахождение кнопки «Создать» для запроса

При нажатии на кнопку «Создать» пользователя перенаправляет на страницу создания запроса. На этой странице в форме необходимо указать тип запроса и прикрепить файл, который будет подписан электронной подписью.

После прикрепления файла, данные о файле отображены ниже в форме. При необходимости есть возможность удалить файл и прикрепить другой. Удаление представлено на рисунке 15.

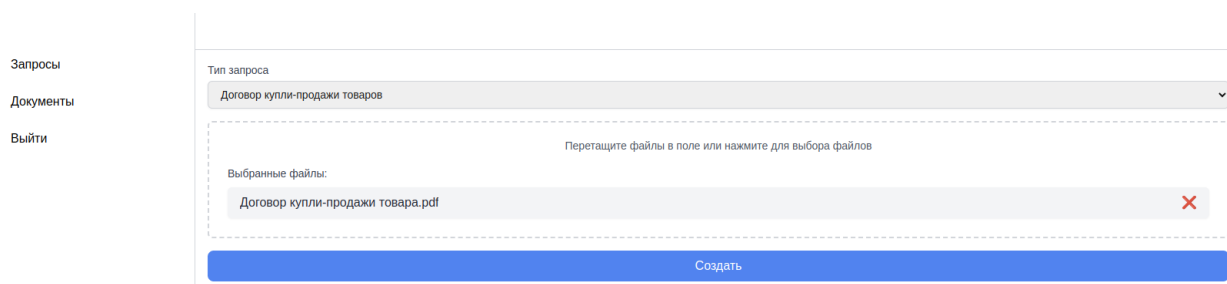


Рисунок 15 – Удаление файла в форме создания запроса

После успешного создания запроса пользователя перенаправляет на страницу с документами, прикрепленными к запросу. Пример на рисунке на 16.

Запросы
Документы
Выйти

Документы в запросе 98f1aedef-b360-4154-b2e6-d3a5495359ee

Наименование	Описание	Статус	Дата создания	Действия
Договор купли-продажи товара.pdf		DRAFT		Просмотреть

Рисунок 16 – Страница с документами, прикрепленными к запросу

При нажатии на кнопку «Просмотреть» открывается модальное окно с документом, представленным на рисунке 17.

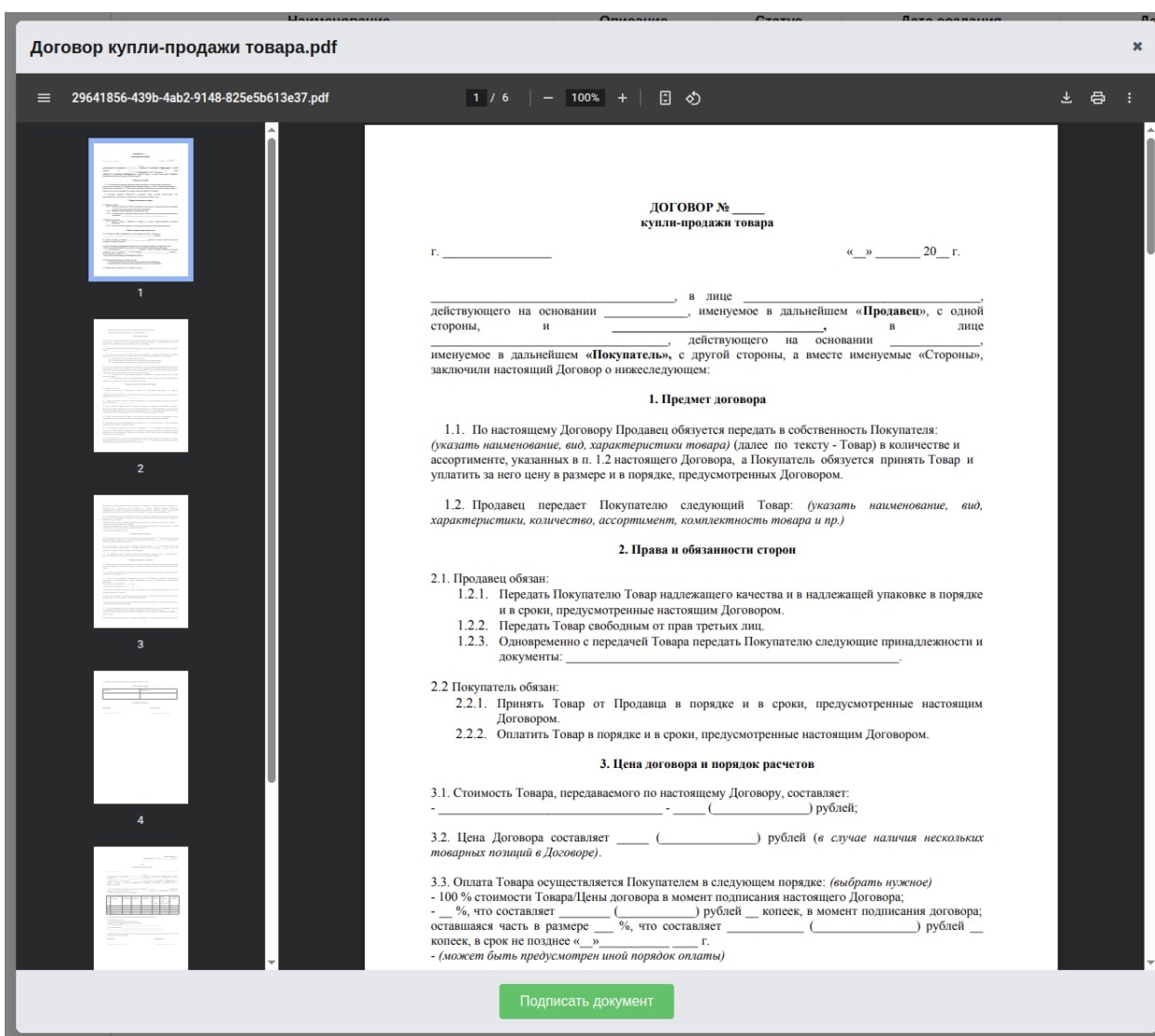


Рисунок 17 – Модальное окно с документом

Как мы видим на рисунке выше, документ можно подписать. После подписания документа кнопка «Подписать документ» скрывается и можно заметить синюю печать внизу документа с информацией об электронной подписи. Пример представлен на рисунке 18.

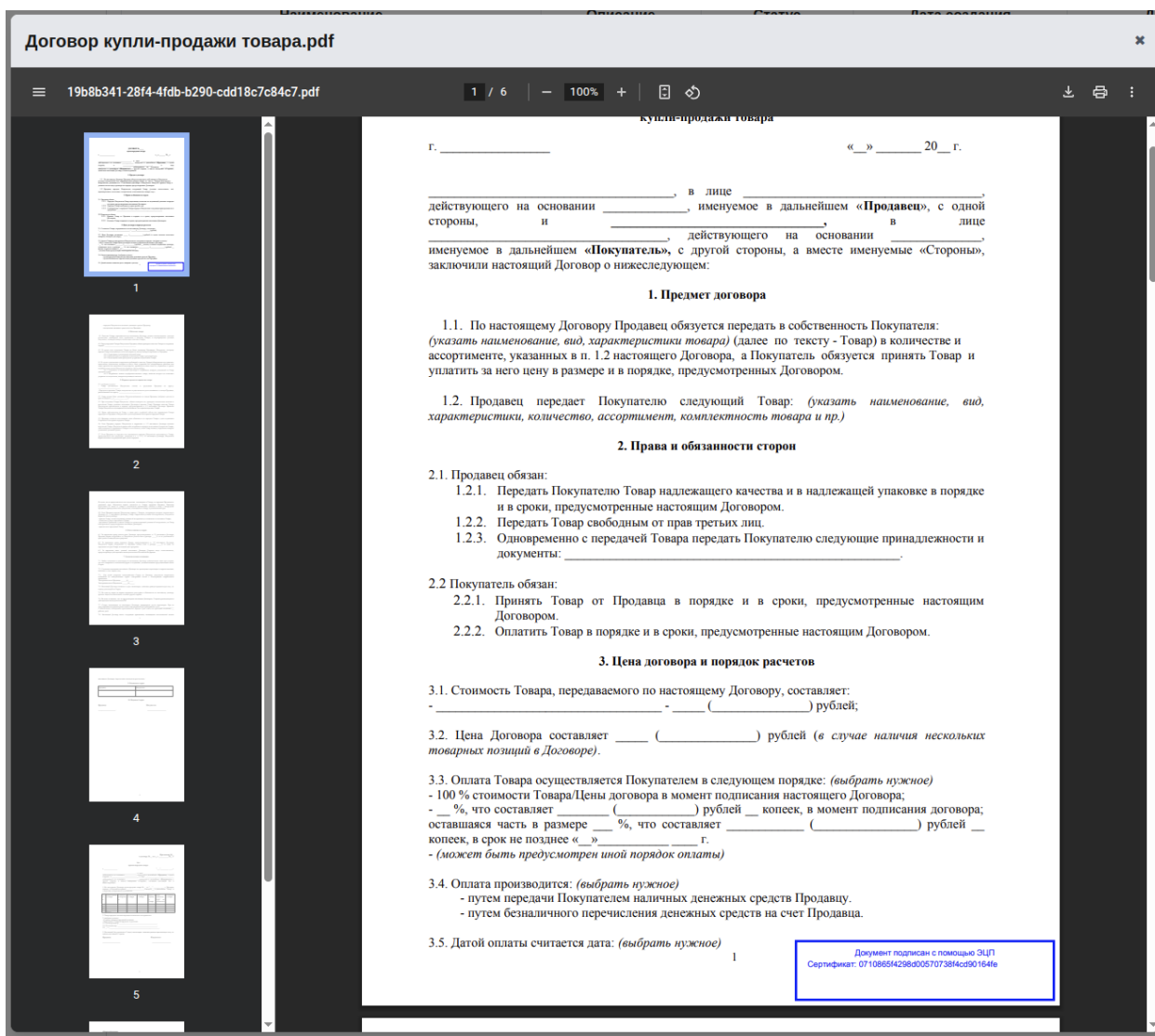


Рисунок 18 – Синяя печать с информацией об ЭЦП

Таким образом мы продемонстрировали комплексный подход к реализации СЭД с учётом архитектурных, функциональных и интеграционных аспектов, соответствующих современным требованиям к корпоративным информационным системам.

3.2 Тестирование программного обеспечения

Теперь рассмотрим тестирование системы. Процесс тестирования был организован на нескольких уровнях:

- модульное тестирование – проводится с использованием JUnit 5 и Mockito, охватывает бизнес-логику сервисов;
- интеграционные тесты – проверяют взаимодействие между слоями: контроллер → сервис → репозиторий, используется @SpringBootTest с поднятием контекста;
- ручное тестирование – охватывает пользовательские сценарии, тесты на удобство интерфейса и поведение в различных условиях.

Модульное тестирование проводится для конкретных классов сервисов. Например, в сервисе запросов мы тестируем создание запроса с типом документа «SALES_CONTRACT» и делаем заглушку для получения текущего пользователя. Пример модульного теста представлен на рисунке 19.

```
@Test
void shouldCreateRequestInDraftStatus() {
    RequestCreateDto dto = new RequestCreateDto("SALES_CONTRACT");
    when(userProvider.getCurrentUser()).thenReturn(testUser);
    RequestResponseDto result = requestService.createRequest(dto);
    assertEquals(RequestStatus.DRAFT, result.getStatus());
}
```

Рисунок 19 – Пример модульного теста

Для тестирования контроллеров применяется интеграционное тестирование. Особенностью данного тестирования является то, что имитируется реальный вызов к контроллеру. Пример интеграционного теста представлен на рисунке 20.

```

@Test
void createRequest_shouldReturnCreated() throws Exception {
    mockMvc.perform(post("/api/requests")
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"type\": \"SALES_CONTRACT\"}")
        .header("Authorization", "Bearer " + jwtToken))
        .andExpect(status().isCreated());
}

```

Рисунок 20 – Пример интеграционного теста

Далее рассмотрим ручное тестирование. Оно занимает важное место в жизненном цикле разработки, особенно в контексте внедрения бизнес-критичных решений, таких как система электронного документооборота в торговой организации. Несмотря на быстрое развитие инструментов автоматизации и повсеместное внедрение CI/CD-подходов, ручное тестирование остаётся незаменимым на ряде этапов, где требуется гибкость, контекстуальная интерпретация действий пользователя, оценка удобства интерфейса, а также оперативная проверка нестандартных сценариев использования.

В процессе ручного тестирования непосредственное взаимодействие с системой позволяет выявлять такие категории дефектов, которые могут быть упущены при автоматизированной проверке. Это, прежде всего, визуальные несоответствия, логические расхождения в сценариях поведения, сложности в навигации и использовании, неочевидные ошибки, возникающие при работе на границе допустимых значений входных данных. Кроме того, ручное тестирование способствует формированию целостного восприятия приложения и выявлению проблем, связанных с удобством использования (юзабилити).

В рамках реализации системы электронного документооборота ручное тестирование проводилось по следующим тест кейсам, представленным в таблице 2.

Таблица 2 – Тест кейсы для системы ЭДО

Название теста	Предусловия	Шаги	Ожидаемый результат
Регистрация пользователя с корректными данными	Пользователь не зарегистрирован	<ul style="list-style-type: none"> – открыть форму регистрации – ввести корректные данные – нажать «Зарегистрироваться» 	Регистрация успешна, переход на форму входа
Авторизация зарегистрированного пользователя	Пользователь зарегистрирован	<ul style="list-style-type: none"> – открыть форму входа – ввести корректный email и пароль – нажать «Войти» 	Переход в личный кабинет с отображением списка заявок
Создание запроса с прикреплением файла	Пользователь авторизован	<ul style="list-style-type: none"> – перейти в «Заявки» – нажать «Создать» – заполнить поля – прикрепить файл – сохранить 	Запрос отображается в списке, файл доступен для просмотра
Подписание документа	Документ прикреплен	<ul style="list-style-type: none"> – открыть карточку запроса – нажать «Просмотреть» – нажать «Подписать документ» 	Отображается штамп ЭЦП, подтверждающий подписание
Предпросмотр документа	Документ прикреплен	<ul style="list-style-type: none"> – нажать «Просмотреть» 	Открывается встроенный просмотрщик
Проверка интерфейса на разных устройствах	Различные экраны (ПК, ноутбук, терминал)	<ul style="list-style-type: none"> – открыть систему на каждом устройстве – проверить корректность интерфейса 	Интерфейс адаптирован, все функции доступны
Восстановление файла после сбоя загрузки	Имитирован сбой при загрузке	<ul style="list-style-type: none"> – повторить попытку загрузки файла 	Файл успешно загружен, отображается в системе

Продолжение таблицы 2

Одновременная работа с документом несколькими пользователями	Два пользователя авторизованы	<ul style="list-style-type: none"> – открыть документ с разных аккаунтов – попробовать редактировать 	Система блокирует одновременное редактирование или обрабатывает изменения корректно
Валидация при регистрации: некорректные данные	–	<ul style="list-style-type: none"> – ввести некорректный email или короткий пароль – оставить обязательные поля пустыми – нажать «Зарегистрироваться» 	Ошибки отображаются, регистрация невозможна
Валидация при авторизации: некорректные данные	–	<ul style="list-style-type: none"> – ввести несуществующий email или неверный пароль – нажать «Войти» 	Отказ в авторизации, сообщение об ошибке
Создание запроса без прикрепления файла	Пользователь авторизован	<ul style="list-style-type: none"> – нажать «Создать запрос» – заполнить поля – не прикреплять файл – нажать «Сохранить» 	Система не сохраняет запрос, сообщает об обязательности прикрепления файла

ТС-1. Регистрация пользователя с корректными данными. Предусловием является отсутствие зарегистрированного пользователя в системе. Пользователь открывает форму регистрации, вводит корректные данные (ФИО, email, пароль) и нажимает кнопку «Зарегистрироваться». Ожидается успешная регистрация и перенаправление на форму входа.

ТС-2. Авторизация зарегистрированного пользователя. Пользователь вводит корректный email и пароль в форму входа и нажимает «Войти». В случае успеха происходит переход к личному кабинету с отображением списка заявок.

ТС-3. Создание запроса с прикреплением файла. Пользователь, находясь в системе, нажимает «Создать запрос», выбирает тип, прикрепляет файл, вводит описание и сохраняет. Новый запрос отображается в списке, файл доступен для просмотра.

ТС-4. Подписание документа. Пользователь открывает документ, прикрепленный к запросу, и нажимает «Подписать». После этого отображается штамп или пометка о подписи, подтверждающая завершение действия.

ТС-5. Предпросмотр документа. Пользователь нажимает «Просмотреть» возле прикрепленного файла. Ожидается открытие документа во встроенном просмотрщике без скачивания.

ТС-6. Проверка интерфейса на разных устройствах. Система открывается на разных экранах: ПК, ноутбук, терминал. Интерфейс должен адаптироваться, все элементы оставаться доступными.

ТС-7. Восстановление файла после сбоя загрузки. При сбое в процессе загрузки файла пользователь повторяет попытку. Файл успешно загружается, отображается в карточке запроса.

ТС-8. Одновременная работа с документом несколькими пользователями. Два пользователя одновременно открывают документ. Система предотвращает конфликты, применяя блокировку или синхронизацию изменений.

ТС-9. Валидация данных при вводе некорректных данных при регистрации. Пользователь вводит некорректный email (например, без символа «@»), слишком короткий пароль или оставляет обязательные поля пустыми. После нажатия кнопки «Зарегистрироваться» система должна отобразить сообщения об ошибках и не допустить продолжения процесса. Данная валидация представлена на рисунке 21.

The image shows a registration form titled "Регистрация". It contains several input fields: "Email" (with "test@example.com" and a red error message "Email уже используется."), "Firstname", "Lastname" (with a red error message "Имя обязательно."), "Middlename", and a password field (with a red error message "Фамилия обязательна." and "Пароль должен содержать минимум 6 символов."). A blue "Создать" button is at the bottom, and a link "Есть аккаунт? Войти" is below it.

Рисунок 21 – Ошибки валидации при регистрации пользователя

ТС-10. Валидация данных при вводе некорректных данных при авторизации. Пользователь вводит несуществующий email, неверный пароль или оставляет поля пустыми. При попытке авторизации система должна сообщить о недействительных данных и отказать в доступе без перехода в личный кабинет. Пример представлен на рисунке 22.

The image shows an authorization form titled "Авторизация". It contains two input fields: "Email" (with "test@edms.loc1") and a password field (with "....."). A red error message "Неверный email или пароль." is displayed above the fields. A blue "Войти" button is at the bottom, and a link "Нет аккаунта? Зарегистрироваться" is below it.

Рисунок 22 – Ввод некорректных данных при авторизации

ТС-11. Создание запроса без прикрепления файла. Пользователь создаёт запрос, указывая тип и описание, но не прикрепляет файл. При нажатии кнопки «Сохранить» система должна отобразить сообщение об ошибке и не разрешить сохранить запрос без документа. Пример можно увидеть на рисунке 23.

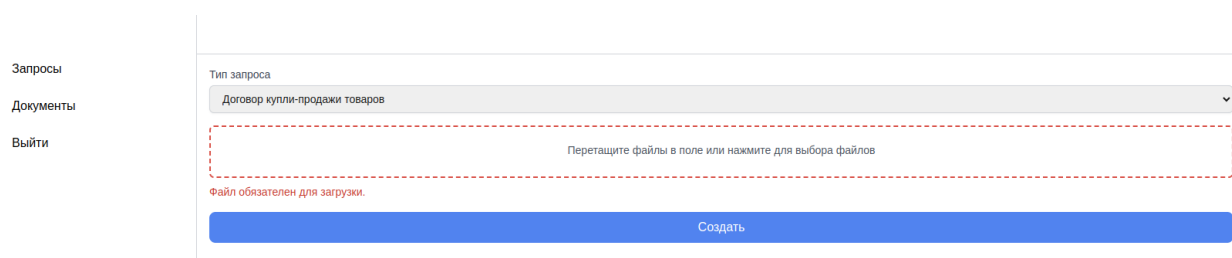


Рисунок 23 – Ошибка валидации при создании запроса

Каждый из вышеперечисленных сценариев вручную тестировался с применением различных комбинаций параметров ввода, действий и последовательностей, что позволило выявить множественные аспекты, требующие доработки. Например, в ходе проверки создания и редактирования заявок были обнаружены ошибки, связанные с валидацией полей, недоступностью кнопок в определённом состоянии заявки, а также некорректной обработкой неочевидных пользовательских действий.

Одним из важнейших направлений ручного тестирования стала проверка негативных сценариев. Именно эти случаи чаще всего не охватываются при написании автоматических тестов, но на практике они представляют собой критически важный элемент. В рамках тестирования рассматривались следующие нестандартные и потенциально опасные действия:

- ввод недопустимых символов, пустых значений и чрезмерно длинных строк в поля форм;

- загрузка файлов неподдерживаемого формата или превышающих лимит размера;
- принудительное обновление страницы в процессе сохранения или отправки;
- открытие одной и той же заявки несколькими пользователями;
- попытка выполнения действий, недопустимых по текущему статусу документа;
- манипуляции с URL-адресами с целью доступа к чужим данным;
- использование устаревших токенов авторизации.

Эти тесты позволили получить представление о степени устойчивости системы к ошибкам пользователя, потенциальным попыткам злоупотребления, а также к внешним сбоям. Кроме того, большое значение имели тесты на многопользовательское взаимодействие и состояние конкурентного доступа, при которых несколько пользователей выполняли действия с одними и теми же объектами системы. Это позволило убедиться в корректной работе механизмов блокировки и предотвращения конфликтов данных.

Все найденные ошибки и отклонения фиксировались вручную с указанием последовательности действий, вызывавших проблему, и предоставлялись разработчикам для анализа и устранения. Такой подход позволил наладить продуктивную обратную связь между тестированием и программной доработкой. Некоторые найденные ошибки носили системный характер и требовали изменения логики обработки данных, что невозможно было бы выявить при поверхностной автоматической проверке.

Также следует отметить, что ручное тестирование включало в себя проверку работы интерфейса на разных устройствах и разрешениях экрана. Это важно в условиях, когда пользователи могут работать с системой как с персонального компьютера, так и с ноутбуков, корпоративных терминалов и, возможно, мобильных браузеров. Проверка адаптивности интерфейса,

сохранения читаемости, корректности отображения таблиц, кнопок, всплывающих окон и форм — все эти аспекты также покрывались в процессе ручного тестирования.

Отдельного внимания заслуживает взаимодействие системы с внешними компонентами, такими как хранилище MinIO, библиотека PDFBox, механизм электронной подписи и авторизации. Ручные проверки позволили убедиться в корректности загрузки файлов, подписи, отображения метаданных, применения штампа, а также восстановления документов в случае неудачной загрузки.

Такая комплексная стратегия позволяет обеспечить высокое качество и надёжность системы.

В ходе разработки и тестирования системы электронного документооборота были учтены все критичные аспекты, такие как безопасность, интеграция с внешними системами и соблюдение нормативных требований. Архитектура приложения была спроектирована с учётом масштабируемости, надёжности и защищённости данных, что является основой для дальнейшего внедрения и использования системы в реальной производственной среде.

Процесс тестирования включал как ручное тестирование, так и автоматизированные юнит- и интеграционные тесты, что позволило выявить и устранить все значимые ошибки до релиза системы. Использование инструментов мониторинга и логирования обеспечило оперативное выявление и устранение возникающих проблем.

Система зарекомендовала себя как надёжное и безопасное решение для обработки и хранения электронных документов, что делает её идеальной для использования в организации, работающей в сфере торговли и других сферах, где важна безопасность и скорость обработки документов.

Заключение

Современные реалии стремительно трансформируют подходы к организации документооборота в коммерческих структурах. Электронный документооборот, ранее воспринимаемый как вспомогательная технология, сегодня становится критически важным элементом корпоративной инфраструктуры. В условиях цифровизации, роста информационного потока, ужесточения требований к юридической значимости документации и повышения уровня конкуренции, организация эффективной, безопасной и автоматизированной системы управления документами становится необходимостью, а не опцией.

В рамках данной бакалаврской работы была реализована информационная система электронного документооборота (СЭД), ориентированная на нужды торговой организации. В процессе выполнения проекта были выполнены всесторонние исследования, охватывающие как технические, так и организационные аспекты внедрения СЭД, что позволило получить комплексное представление о процессе цифровизации документооборота в корпоративной среде.

Поставленная цель работы – разработка современной, гибкой и юридически значимой системы электронного документооборота для торговой компании – была достигнута в полном объеме. Система охватывает все ключевые этапы работы с документами: создание, редактирование, регистрация, подписание, шифрование, передача, хранение и архивирование. Это позволило сформировать замкнутый цифровой цикл обработки документов, в котором минимизировано участие бумажных носителей и исключены потери информации.

В ходе выполнения проекта были решены следующие основные задачи:

- проведен анализ существующих систем электронного документооборота, включая как коммерческие, так и открытые

решения, что позволило выявить тенденции и лучшие практики отрасли;

- сформулированы требования к системе, исходя из специфики торговой деятельности, включая необходимость высокой скорости обработки, отказоустойчивости и интеграции с другими бизнес-сервисами;
- спроектирована архитектура программной системы на базе современных технологий: Java Spring Framework, микросервисный подход, использование REST API, JSON Web Token (JWT) для авторизации и MinIO для хранения файлов;
- разработаны модули регистрации и учета документов, реализации маршрутов согласования, интеграции электронной подписи, безопасного хранения и контроля версий;
- внедрены механизмы криптографической защиты данных с использованием библиотеки Bouncy Castle, реализована проверка подлинности документа и его неизменности;
- проведено функциональное и нагрузочное тестирование, по результатам которого подтверждена надежность и производительность системы в условиях, приближенных к эксплуатационным.

Система обладает рядом ключевых преимуществ. Она легко масштабируется и адаптируется под потребности конкретной организации, обеспечивая гибкость при внедрении и дальнейшем развитии. В ней реализован продуманный механизм маршрутизации документов, что позволяет настраивать и оптимизировать внутренние процессы. Поддержка электронной цифровой подписи и ведение подробного журнала действий обеспечивают высокий уровень безопасности и прозрачности документооборота. Интерфейс системы интуитивно понятен и ориентирован на обычных сотрудников, не обладающих специальной IT-подготовкой, что

способствует быстрой адаптации пользователей и эффективной работе с системой. Проведенные исследования показали, что автоматизация документооборота дает значительный экономический эффект. Она позволяет сократить временные затраты на обработку документов до 50-70%, снизить количество ошибок, связанных с человеческим фактором, обеспечить единое хранилище информации и прозрачность всех операций. Это особенно критично для торговых организаций, где скорость и точность бизнес-процессов напрямую влияют на прибыль и конкурентоспособность.

Разработанная система может быть успешно адаптирована к предприятиям малого и среднего бизнеса, а в перспективе – и для крупных организаций при условии масштабирования инфраструктуры. Внедрение решения в производственную эксплуатацию требует подготовки персонала, формализации регламентов обработки документов, а также интеграции с внутренними сервисами организации.

Также стоит отметить возможности дальнейшего развития проекта. Среди перспективных направлений:

- внедрение искусственного интеллекта и машинного обучения для автоматической классификации и маршрутизации документов;
- использование блокчейн-технологий для обеспечения полной прослеживаемости и неизменности истории документа;
- расширение поддержки мобильных платформ и удаленного доступа;
- интеграция с государственными системами ЭДО (например, СФЕ, ОФД, ГИС МТ и др.).

Таким образом, данная бакалаврская работа не только имеет теоретическую и практическую значимость, но и представляет собой готовую к внедрению систему, способную существенно повысить эффективность документооборота в торговой организации. Полученные знания, навыки и результаты работы могут быть использованы как основа для дальнейших разработок в области цифровизации корпоративного управления.

Список используемых источников

1. Баланов А. Н. Цифровизация в розничной и оптовой торговле. Разработка, интеграция и внедрение технологических решений. – Санкт-Петербург: Лань, 2024. – 350 с.
2. Бобылева М. П. Управленческий документооборот: от бумажного к электронному. Вопросы теории и практики. – Москва: Инфра-М, 2023. – 256 с.
3. Бондарев А. В. 1С:Документооборот. 250 вопросов и ответов. – Москва: ЛитРес, 2023. – 350 с.
4. Васильева И. Н. Криптографические методы защиты информации: учебник и практикум для вузов. – Москва: Юрайт, 2020. – 349 с.
5. Гарант. Информационно-правовой портал [Электронный ресурс]. – Режим доступа: <https://www.garant.ru/> (дата обращения: 04.05.2025).
6. Гатченко Н. А., Исаев А. С., Яковлев А. Д. Криптографическая защита информации. – Санкт-Петербург: НИУ ИТМО, 2012. – 142 с.
7. Глотова С. А., Бондарев А. В. Управление документами в современной организации. – Москва: ЛитРес, 2023. – 320 с.
8. Диамандис П., Котлер С. Изобилие. Будущее лучше, чем вы думаете. – Москва: Эксмо, 2020. – 400 с.
9. Доэрти П., Уилсон Д. Человек + машина. Новые принципы работы в эпоху искусственного интеллекта. – Москва: Альпина Паблишер, 2020. – 280 с.
10. Зараменских Е. П. Информационные системы в бизнесе: учебник и практикум для среднего профессионального образования. – 2-е изд., перераб. и доп. – Москва: Юрайт, 2023. – 470 с.
11. Издательство Лань. Сопровождение документов в системе электронного документооборота [Электронный ресурс]. – Режим доступа: <https://lanbook.com/catalog/discipline/soprovozhdenie-dokumentov-v-sisteme-elektronnogo-dokumentoooborota/> (дата обращения: 04.05.2025).

12. Издательство Лань. Информационные технологии в экономике [Электронный ресурс]. – Режим доступа: <https://lanbook.com/catalog/discipline/informatsionnye-tekhnologii-v-ekonomike>

13. Книги о системах электронного документооборота [Электронный ресурс]. – Режим доступа: https://www.doc-online.ru/tags/knigi_o_sed/ (дата обращения: 04.05.2025).

14. КонсультантПлюс. Справочная правовая система [Электронный ресурс]. – Режим доступа: <https://www.consultant.ru/> (дата обращения: 04.05.2025).

15. Коржук В. М., Попов И. Ю., Воробьева А. А. Защищенный документооборот. Часть 1: Учебно-методическое пособие. – Санкт-Петербург: Университет ИТМО, 2021. – 67 с.

16. ЛитРес. Электронные и аудиокниги по электронному документообороту [Электронный ресурс]. – Режим доступа: <https://www.litres.ru/tags/elektronnyy-dokumentoorot/> (дата обращения: 04.05.2025).

17. Лихтенштейн Р. Информационные технологии в бизнесе. Практикум: применение системы Decision. – Москва: Лабиринт, 2023. – 300 с.

18. Михайлов А. ИТ-стратегия и стратегия цифровой трансформации бизнеса: что это такое, варианты разработки, основные разделы. – Москва: Инфо-Стратегия, 2023. – 314 с.

19. Музагафаров А. Шифрованный мир: азы криптографии. Просто, понятно и увлекательно. – Москва: ЛитРес, 2018. – 250 с.

20. Нараян Ш. Agile IT Organization Design: для цифровой трансформации и непрерывной доставки. – Москва: Эксмо, 2020. – 300 с.

21. Рис Э. Бизнес с нуля. Метод Lean Startup для быстрого тестирования идей и выбора бизнес-модели. – Москва: Манн, Иванов и Фербер, 2020. – 320 с.

22. Система электронного документооборота «1С:Документооборот» [Электронный ресурс]. – Режим доступа: <https://v8.1c.ru/docflow/> (дата обращения: 04.05.2025).

23. Система электронного документооборота «Контур.Диадок» [Электронный ресурс]. – Режим доступа: <https://www.diadoc.ru/> (дата обращения: 04.05.2025).

24. Система электронного документооборота «Битрикс24» [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.ru/> (дата обращения: 04.05.2025).

25. Система электронного документооборота «КриптоПро» [Электронный ресурс]. – Режим доступа: <https://www.cryptopro.ru/> (дата обращения: 04.05.2025).

26. Система электронного документооборота «Тензор» [Электронный ресурс]. – Режим доступа: <https://tensor.ru/> (дата обращения: 04.05.2025).

27. Ульянцева С. Э. Управление документами: быстро, эффективно, своими силами. На примере «1С:Документооборота 8». – Москва: Рарус, 2015. – 148 с.

28. Федеральный закон от 06.04.2011 № 63-ФЗ «Об электронной подписи» [Электронный ресурс]. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_112701/ (дата обращения: 04.05.2025).

29. Хамбл Д., Молески Дж., О'Рейли Б. Бережливое предприятие: как высокоэффективные организации внедряют инновации в масштабе. – Москва: Эксмо, 2020. – 350 с.

30. Хворенков С. Г. Система Электронного Документооборота (СЭД) на платформе 1С:Предприятие 8.2. – Санкт-Петербург: Лань, 2021. – 200 с.