Образовательная автономная некоммерческая организация высшего образования

«МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ»

Направление ***09.03.03*** (код) **Направленность** ***Вычислительные машины, комплексы, системы и сети*** (наименование)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему: Разработка программного обеспечения для организации обмена данными в вычислительной сети организации АО «ФИНТЕХ»

Обучающийся ***Хомутов Сергей Андреевич*** (Ф.И.О. полностью) _____ подпись

Руководитель ***Быков Артем Александрович*** (Ф.И.О. полностью) _____ подпись

МОСКВА 2023 г.

Образовательная автономная некоммерческая организация высшего образования

«МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ»

ЗАДАНИЕ

на выпускную квалификационную работу

обучающемуся

1. Тема: Разработка программного обеспечения для организации обмена данными в вычислительной сети организации АО «ФИНТЕХ»
2. Срок сдачи обучающимся законченной работы «__» _____ 20___ г.
3. Исходные данные к ВКР: научная и учебная литература, интернет ресурсы и др.
4. Содержание ВКР (перечень подлежащих разработке вопросов):

ВВЕДЕНИЕ

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ДЛЯ РАЗРАБОТКИ ПО ИС

1.1 Изучение существующих методов и средств обмена данными в вычислительной сети организации

1.2 Определение общих потребностей и требований пользователей в области обмена данными

1.3 Анализ существующих проблем и вызовов при обмене данными в вычислительной сети организации

1.4 Характеристика исследуемой организации и обоснование потребности в проектировании системы обмена данными

Выводы по разделу 1

2. ВЫБОР ТЕХНОЛОГИЙ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС

2.1.1 Выбор типа приложения

2.1.2 Выбор языков программирования

2.1.3 Выбор системы управления базой данных

2.1.4 Выбор графического фреймворка

2.1.5 Выбор алгоритма шифрования файлов

2.2 Формирование требований к разработке ПО

Выводы по разделу 2

3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС

3.1 Реализация программного обеспечения для организации обмена данными в вычислительной сети организации

3.2 Пример функционирования программного обеспечения для организации обмена данными в вычислительной сети организации

3.3 Оценка экономической эффективности проекта

Выводы по разделу 3

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

4. Перечень графического материала ___таблиц, ___ рисунков.
5. Приложения ___.

Дата выдачи задания __.__.202__ г. Задание принял (дата) __.__.202__ г.

Подпись руководителя _____ Подпись обучающегося _____

СОДЕРЖАНИЕ

Введение.....5

1. Анализ предметной области для разработки ПО ИС.....7

1.1 Изучение существующих методов и средств обмена данными в вычислительной сети организации.....7

1.2 Определение общих потребностей и требований пользователей в области обмена данными.....16

1.3 Анализ существующих проблем и вызовов при обмене данными в вычислительной сети организации.....18

1.4 Характеристика исследуемой организации и обоснование потребности в проектировании системы обмена данными.....22

Выводы по разделу 1.....32

2. Выбор технологий и проектирование программного обеспечения ИС. 34

2.1.1 Выбор типа приложения.....34

2.1.2 Выбор языков программирования.....36

2.1.3 Выбор системы управления базой данных.....40

2.1.4 Выбор графического фреймворка.....41

2.1.5 Выбор алгоритма шифрования файлов.....42

2.2 Формирование требований к разработке ПО.....44

Выводы по разделу 2.....53

3. Реализация программного обеспечения ИС.....56

3.1 Реализация программного обеспечения для организации обмена данными в вычислительной сети организации.....56

3.2 Пример функционирования программного обеспечения для организации обмена данными в вычислительной сети организации.....61

3.3 Оценка экономической эффективности проекта.....67

Выводы по разделу 3.....71

Заключение.....72

Список использованных источников.....76

Приложение 1 – Программный код.....78

ВВЕДЕНИЕ

Разработка системы обмена файлами в локальной сети организации остается актуальной и имеет множество преимуществ и применений, несмотря на современные облачные и интернет-сервисы обмена.

Ниже представлены факторы, подтверждающих актуальность разработки и внедрения подобной системы в организации:

- Безопасность и контроль. Локальная система обмена файлами предоставляет организации полный контроль над данными. Важные и конфиденциальные файлы могут храниться и передаваться внутри сети, где организация может обеспечивать более строгие меры безопасности и контроля доступа.
- Высокая скорость. Обмен файлами в локальной сети обычно более быстр и эффективен, чем через интернет, что делает такую систему более подходящей для передачи больших объемов данных или файлов высокого разрешения.
- Контроль конфиденциальности. Система позволит хранить данные в пределах локальной сети, что очень хорошо для их безопасности.
- Легкость интеграции. Локальная система обмена файлами легче интегрировать с существующими внутренними приложениями и процессами, такими как серверы для автоматической обработки данных.
- Расходы. Локальная система обмена файлами может быть более экономичной, особенно если организация уже имеет необходимое оборудование и ресурсы.
- Управление пропускной способностью. Сетевой администратор может эффективно управлять пропускной способностью и приоритетами обмена файлами внутри локальной сети, что обеспечивает более стабильную производительность.

Важно отметить, что локальная система обмена файлами может хорошо дополнять облачные решения и использоваться для внутреннего обмена файлами

в организации, особенно в случаях, когда безопасность и скорость имеют приоритет.

Таким образом, ключевыми преимущества применения разработанной системы обмена файлами в организации должны быть высокая скорость, безопасность, простота использования, невысокая стоимость интеграции и обслуживания. Данные соображения обуславливают актуальность темы выпускной работы.

Целью выпускной работы является разработка программного обеспечения для организации обмена данными в вычислительной сети организации.

Объектом исследования выступает организация АО «ФИНТЕХ», которая является одним из ведущих разработчиков общего и специального программного обеспечения, специальных технических средств и комплексов в интересах государственных и корпоративных заказчиков.

Предметом исследования является процесс разработки программного обеспечения для организации обмена данными в вычислительной сети организации.

Для достижения поставленной цели надо решить задачи:

- выполнить анализ предметной области;
- выбрать технологии проектирования;
- осуществить проектирование программного обеспечения ИС;
- выполнить реализацию программного кода приложения;
- показать пример работы системы;
- обосновать экономический эффект для организации от внедрения данной системы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ДЛЯ РАЗРАБОТКИ ПО ИС

1.1 Изучение существующих методов и средств обмена данными в вычислительной сети организации

Обмен данными в компьютерной сети организации имеет решающее значение для общения, сотрудничества и обмена информацией между подразделениями и/или отдельными их сотрудниками.

Выбор метода и средств обмена данными в компьютерной сети организации зависит от различных факторов, включая характер данных, требования безопасности, сетевой архитектуры и конкретных потребностей организации. Часто комбинация этих методов обеспечивает комплексное решение для эффективного и безопасного обмена данными внутри организации.

Локальная сеть (LAN) – это фундамент для обмена данными в организации. Локальные сети используются для подключения компьютеров и устройств в пределах локализованной территории, например, офисного/учебного здания или центра обработки данных. Близость устройств сети обеспечивает быстрый и надежный обмен данными между ними.

Локальные сети (ЛВС) обычно обеспечивают высокую скорость передачи данных, часто варьирующуюся от 100 Мбит/с (Fast Ethernet) до 1 Гбит/с (Gigabit Ethernet) или 10 Гбит/с (10 GE), что делает пригодными для передачи больших файлов и обеспечивать поддержку бизнес-процессов с интенсивным использованием данных [17].

ЛВС могут быть реализованы с использованием проводных (Ethernet) или беспроводных (Wi-Fi) соединений. Проводные локальные сети обеспечивают более высокую скорость и безопасность, а беспроводные локальные сети обеспечивают мобильность и гибкость.

Обмен данными в компьютерной сети организации может происходить различными способами, каждый из которых имеет свои особенности и варианты

использования. Перечислим распространенные методы обмена данными внутри компьютерной сети организации:

1) Протоколы FTP и FTPS.

Классическим примером обмена данными в ЛВС является протокол FTP (FTPS). FTP (протокол передачи файлов) и FTPS (FTP Secure) – это методы обмена данными в компьютерной сети организации, предполагающие передачу файлов между устройствами.

FTP – это стандартный сетевой протокол, используемый для передачи файлов с одного хоста на другой по сети на основе протокола TCP. Он широко поддерживается и относительно прост в использовании для обмена файлами, не обеспечивает шифрование, а это означает, что данные, передаваемые с помощью FTP, не защищены. Это может стать проблемой при обмене конфиденциальной информацией, даже внутри ЛВС организации [12].

FTPS – это расширение FTP, которое добавляет уровень безопасности за счет шифрования данных во время передачи, снижая риск несанкционированного доступа или перехвата. Он использует протоколы SSL (Secure Sockets Layer) или TLS (Transport Layer Security) для защиты передачи данных и считается более безопасным вариантом, что делает его подходящим для обмена конфиденциальными данными внутри сети организации.

FTP и FTPS полезны для обмена файлами и данными внутри организации, причем FTPS является предпочтительным выбором для безопасной передачи данных, особенно когда речь идет о конфиденциальной информации.

Таким образом, локальные сети с протоколами FTP и FTPS являются фундаментальным и важным компонентом сетевой инфраструктуры организации. Они предоставляют высокоскоростные и надежные средства обмена данными, способствуя эффективному взаимодействию и сотрудничеству между пользователями и устройствами внутри организации.

2) Электронная почта.

Электронная почта – широко используемый метод обмена данными в компьютерной сети организации. Она включает отправку сообщений и файлов от одного пользователя другому через серверы электронной почты [22].

Пользователи могут отправлять текстовые сообщения, документы и мультимедийные файлы в виде вложений. Электронные письма имеют структурированный формат со строками темы, адресами отправителя и получателя, а также телами сообщений.

Доступ к электронной почте можно получить с различных устройств и мест, что делает ее очень удобным инструментом. К тому же службы электронной почты часто включают в себя шифрование и спам-фильтры для повышения безопасности информационного обмена.

Электронные письма можно архивировать и легко искать в них информацию в дальнейшем. Также электронную почту можно интегрировать с другими приложениями и службами, оптимизируя общение между сотрудниками. Пользователи могут обмениваться информацией, совместно работать над документами и общаться с коллегами.

Таким образом, электронная почта – универсальный и широко распространенный метод внутренней и внешней коммуникации внутри организаций.

3) Общие сетевые диски.

Общие сетевые диски – это метод обмена данными в компьютерной сети организации, предполагающий создание централизованных мест хранения, доступных авторизованным пользователям. Общие диски облегчают совместную работу, позволяя нескольким пользователям работать над одними и теми же файлами или документами в централизованном расположении на сервере организации [21].

Доступ к общим дискам контролируется механизмом авторизации, что гарантирует, что только авторизованные пользователи смогут просматривать, редактировать или удалять данные.

Файлы и данные организованы в иерархическую структуру, что упрощает категоризацию, поиск и управление информацией. К тому же общие диски можно масштабировать для размещения растущих объемов данных и числа пользователей по мере расширения организации.

Для данных, хранящихся на общих дисках, можно регулярно создавать резервные копии, а для предотвращения потери данных можно применять меры избыточности. Многие системы общих дисков также предлагают контроль версий, позволяя пользователям отслеживать изменения и возвращаться к предыдущим версиям файлов.

Таким образом, общие сетевые диски – это фундаментальный инструмент для обмена данными и совместной работы внутри организаций, гарантирующий централизованное хранение, безопасность и легкий доступ к файлам и данным авторизованным пользователям.

4) Облачные службы хранения и обмена файлами.

Службы облачного хранения и обмена файлами – это метод обмена данными в компьютерной сети организации, который предполагает использование облачных платформ для хранения, совместного использования и совместной работы над файлами и данными. Облачные сервисы, такие как Яндекс.Диск, Dropbox, Google.Диск, OneDrive и другие, предлагают удобные и масштабируемые методы обмена файлами и совместной работы над ними в облаке.

Ключевыми особенностями облачных сервисов являются [6]:

- Удаленный доступ. Доступ к данным можно получить из любого места, где есть подключение к сети Интернет, что делает их очень доступными для удаленных и мобильных пользователей.
- Масштабируемость. Облачное хранилище можно легко масштабировать для размещения растущих объемов данных, устраняя необходимость в разноплановом локальном оборудовании.

- Совместная работа. Несколько пользователей могут совместно работать над документами в режиме реального времени с помощью таких функций, как контроль версий и комментирование.
- Синхронизация файлов. Облачные сервисы часто включают автоматическую синхронизацию между устройствами, гарантируя актуальность данных на всех платформах.
- Функции безопасности. Облачные сервисы обычно предлагают возможности шифрования, контроля доступа и аудита для повышения безопасности данных.
- Резервное копирование и восстановление. Резервное копирование данных часто выполняется автоматически, что снижает риск потери данных из-за сбоев оборудования или непредвиденных ситуаций.
- Интеграция. Многие облачные сервисы интегрируются с другими приложениями и сервисами, оптимизируя рабочие процессы и обмен данными.
- Экономическая эффективность. Облачное хранилище снижает потребность в локальной инфраструктуре, что приводит к потенциальной экономии затрат.

Таким образом, службы облачного хранения и обмена файлами широко используются организациями благодаря своей гибкости, доступности и функциям совместной работы, что делает их удобным методом обмена данными и управления файлами.

5) Инструменты для совместной работы.

Инструменты для совместной работы – это метод обмена данными в компьютерной сети организации, который предоставляет командам платформу для общения, обмена информацией и эффективного взаимодействия.

Платформы для совместной работы, Битрикс24, Мегаплан, ЛидерТаск, Slack, Microsoft Teams и Trello, предлагают функции чата и обмена документами, которые облегчают обмен данными в режиме реального времени между членами команды, проводить видеоконференции и обмениваться мгновенными сообщениями.

Пользователи могут обмениваться документами, файлами и данными непосредственно внутри платформы, что облегчает совместную работу над проектами, помогая командам эффективно координировать и выполнять задачи.

Инструменты для совместной работы можно интегрировать с другими приложениями и сервисами, оптимизируя обмен данными и автоматизируя рабочие процессы. Также некоторые инструменты предлагают функции управления версиями для отслеживания изменений и исправлений в документах.

Пользователи могут контролировать, кто имеет доступ к общим данным, обеспечивая безопасность и конфиденциальность данных. К тому же они получают уведомления и оповещения о важных обновлениях и сообщениях, что позволяет им быть в курсе и быть вовлеченными.

Таким образом, инструменты для совместной работы ускоряют и повышают качество выполнения командных задач, оптимизируют обмен данными и повышают производительность внутри организаций, предоставляя центральную платформу для общения и совместной работы.

6) Службы обмена мгновенными сообщениями.

Службы обмена мгновенными сообщениями – это удобный и эффективный способ общения пользователей и обмена данными внутри организации, способствующий сотрудничеству в режиме реального времени и обмену информацией. Это метод обмена данными в компьютерной сети организации, который обеспечивает текстовую связь в реальном времени. Примерами могут служить мессенджеры Skype, Telegram, Viber, WhatsApp и другие.

Системы часто поддерживают групповые беседы, позволяя нескольким пользователям совместно обсуждать темы. Пользователи могут обмениваться файлами, документами и мультимедиа непосредственно в чатах.

Сотрудники получают уведомления о новых сообщениях, что обеспечивает своевременный ответ. Многие такие системы интегрируются с другими приложениями и службами, повышая производительность и обмен данными.

7) Службы удаленного рабочего стола (RDS).

RDS позволяет пользователям подключаться к удаленному компьютеру или серверу через ЛВС или Интернет, предоставляя им доступ к данным и приложениям, как если бы они физически присутствовали на удаленном компьютере.

Пользователи могут обмениваться данными между своим локальным устройством и удаленным компьютером или сервером, осуществляя загрузку, скачивание и обмен файлами.

RDS позволяет пользователям взаимодействовать с удаленной системой в режиме реального времени, облегчая совместную работу и обмен данными между пользователями, расположенными в разных физических местах. Этот обмен обычно безопасный, поскольку все сеансы шифруются для защиты данных во время передачи. Доступ к удаленной системе контролируется посредством механизма аутентификации и разрешений доступа [5].

Сотрудники могут совместно использовать ресурсы (принтеры и локальные диски) между локальной и удаленной системами, что упрощает обмен данными.

Службы удаленного рабочего стола также часто используются для управления и обслуживания удаленных серверов и рабочих станций, сохраняя при этом безопасность данных и централизованный контроль.

Особенностью является тот факт, что RDS может поддерживать несколько одновременных пользователей, что делает его подходящим для сценариев, когда нескольким сотрудникам одновременно требуется доступ к одной и той же системе и данным.

8) Интранет (Интрасеть).

Интранет – это эффективный инструмент, облегчающий внутренний обмен данными, способствующий взаимодействию и обеспечивающий легкий доступ сотрудников к информации, необходимой им для эффективного выполнения своих функций. Интранет обеспечивает безопасную и контролируемую среду для обмена данными внутри сети организации.

Это частный и внутренний метод обмена данными в компьютерной сети организации, безопасная и ограниченная сеть, которая предоставляет несколько ключевых особенностей для обмена данными внутри организации:

- Внутренние коммуникации. Интранет облегчает внутреннюю связь и обмен данными, предлагая протоколы электронной почты, чата и обмена сообщениями исключительно для сотрудников внутри организации.
- Совместное использование документов. Организации могут использовать Интранет для совместного использования и распространения документов, файлов и ресурсов среди сотрудников, способствуя сотрудничеству и обмену данными.
- Централизованное хранилище. Интрасети часто служат центральным хранилищем важной информации и корпоративных данных, что делает их легко доступными для всех сотрудников.
- Безопасность данных. Интрасети обычно защищены межсетевыми экранами и средствами контроля доступа, гарантируя, что конфиденциальные данные компании остаются в безопасности и недоступны для внешних сторон.
- Контроль доступа. Можно определить права доступа и роли пользователей для контроля того, кто может просматривать, редактировать или удалять данные, что повышает безопасность и конфиденциальность данных.
- Базы знаний с возможностью поиска. Интранет часто включает в себя базы знаний с возможностью поиска и внутренние поисковые системы, что позволяет сотрудникам легко находить необходимую им информацию.
- Порталы сотрудников. Интрасети могут предоставлять персонализированные порталы сотрудников с доступом к данным, ресурсам и приложениям, соответствующим роли каждого сотрудника.
- Инструменты для совместной работы. Инструменты для совместной работы, такие как общие календари, дискуссионные форумы и приложения для управления проектами, часто интегрируются в интрасети для облегчения совместной работы и обмена данными.

- Масштабируемость. Интранет можно масштабировать для удовлетворения растущих потребностей организации в обмене данными по мере ее расширения.

Таким образом, можно видеть, что Интранет представляет собой объединение других ранее перечисленных методов обмена данными в единую информационную среду организации.

Также следует кратко отметить, что обмен данными между серверами и рабочими станция также возможен такими средствами [15]:

- Одноранговый (Point-To-Point – P2P) обмен. Сети P2P позволяют осуществлять прямой обмен файлами между отдельными устройствами, хотя обычно это не рекомендуется для конфиденциальных или объемных данных.
- Одноранговый обмен с шифрованием (P2PE). Для безопасной передачи конфиденциальных данных организации могут использовать двухточечное шифрование для защиты данных во время передачи.
- Интеграция баз данных и приложений. Организации могут обмениваться данными между различными программными приложениями и базами данных с помощью API и шлюзов.
- Двухточечные соединения, например по технологии Ethernet, Wi-Fi или Bluetooth, позволяют обмениваться данными между устройствами, находящимися в непосредственной близости.
- NFC. Эта беспроводная технология используется для обмена данными на небольшом расстоянии между мобильными устройствами, устройствами Интернета вещей и периферийными устройствами.
- Репликация базы данных. Для согласованного обмена данными между базами данных, расположенными в разных отделах или рабочих местах, может использоваться репликация баз данных.
- Распределенные файловые системы. Распределенные файловые системы, напрмимер, Hadoop HDFS или GlusterFS, обеспечивают масштабируемое хранение данных и обмен ими по сети.

Таким образом, в данном пункте рассмотрены существующие возможные методы и средства обмена данными в вычислительной сети организации.

1.2 Определение общих потребностей и требований пользователей в области обмена данными

Потребности пользователей организации в обмене данными могут сильно различаться в зависимости от их ролей, обязанностей и характера самой организации. В результате анализа предметной области был выделен набор общих потребностей и ожиданий большинства пользователей в организации в отношении обмена данными:

- Эффективность и скорость. Пользователи ожидают, что методы обмена данными будут быстрыми и эффективными. Медленная передача данных может привести к разочарованию в процессе и снижению производительности работы сотрудников.
- Надежность. Пользователям нужны надежные и последовательные методы обмена данными. Частые сбои или прерывания в работе могут помешать эффективному выполнению бизнес-процессов.
- Безопасность. Безопасность данных является главным приоритетом. Пользователи должны быть уверенными в том, что их данные защищены во время обмена, и что конфиденциальная информация не будет скомпрометирована.
- Доступность. Пользователям необходим легкий доступ к инструментам и методам обмена данными, которые они используют, что включает в себя удобный интерфейс и минимальное время простоя.
- Масштабируемость. По мере роста организации пользователям нужны методы обмена данными, которые можно масштабировать для удовлетворения растущих объемов информации и количества персонала, а значит и рабочих станций.
- Взаимодействие. Функции совместной работы очень важны, особенно для командного режима взаимодействия. Пользователям часто

необходимо работать в режиме реального времени, совместно редактировать документы и беспрепятственно ими обмениваться.

- Совместимость. Сотрудникам нужны методы и инструменты обмена данными, совместимые с используемым ими программным и аппаратным обеспечением. Кроссплатформенная совместимость особенно важна.
- Контроль версий. Пользователям могут потребоваться функции контроля версий, чтобы отслеживать изменения в документах и гарантировать, что они работают с самыми актуальными документами.
- Управление данными. Эффективные инструменты организации и управления данных имеют решающее значение для пользователей, которые имеют дело с большими их объемами. Системы обмена данными в таких случаях должны обладать возможностями поиска, сортировки и категоризации документов.
- Мобильный доступ. В мире, который становится все более мобильным, пользователи часто ожидают, что смогут обмениваться данными не только с компьютеров (рабочих станций), но и со своих смартфонов и планшетов.
- Автоматизация. Функции автоматизации могут помочь оптимизировать процессы обмена данными, например, выполнять автоматическую синхронизацию файлов, настройку фильтров электронной почты или использование сценариев для повторяющихся задач.
- Резервное копирование и восстановление данных. Пользователи хотят быть уверены, что их данные будут зарезервированы и их можно будет восстановить в случае случайного удаления или потери.
- Обучение и поддержка. Организации требуется выделять ресурсы для обучения и поддержки пользователей, чтобы ее сотрудники понимали и эффективно использовали инструменты и методы обмена данными.
- Соответствие и нормативные требования. Пользователям, которые обрабатывают конфиденциальные данные, требуются методы обмена данными, соответствующие определенным нормативным актам, законам и правилам.

- Экономическая эффективность. Сотрудники могут учитывать стоимость методов обмена данными и ожидать, что они принесут пользу организации.
- Конфиденциальность. Пользователи ожидают, что их данные и сообщения останутся конфиденциальными, и они будут иметь контроль над тем, кто может получить доступ к ним.
- Отзывы пользователей и улучшения в работе системы. Сотрудники ценят возможность оставить отзыв об используемых методах обмена данных и хотят видеть постоянное улучшение параметров таких систем на основе их откликов.

Чтобы удовлетворить выделенные потребности, организация должна тщательно выбирать и внедрять решения по обмену данными, учитывая специфические требования различных групп пользователей (сотрудников) и учитывая баланс между безопасностью, удобством и эффективностью. Кроме того, постоянный мониторинг и адаптация методов обмена данными необходимы для обеспечения их соответствия меняющимся потребностям персонала.

В следующем пункте будет выполнен анализ проблем обмена данными.

1.3 Анализ существующих проблем и вызовов при обмене данными в вычислительной сети организации

Обмен данными в компьютерной сети организации может столкнуться с различными проблемами и сложностями. Понимание этих проблем имеет решающее значение для улучшения процессов обмена важной информацией и обеспечения бесперебойной работы инфокоммуникационной инфраструктуры предприятия.

Выполним анализ некоторых распространенных проблем и вызовов при обмене данными в вычислительной сети организации. Результаты сведены в таблицу 1.

Таблица 1 – Результаты анализа распространенных проблем и вызовов при обмене данными в вычислительной сети организации

Проблемы безопасности	Вызовы	Пути решения
Проблемы безопасности	Защита конфиденциальных данных от несанкционированного доступа или взлома является постоянной проблемой любой инфокоммуникационной сети предприятия	Использовать надежное шифрование, контроль доступа и политики безопасности для снижения рисков
Перегрузка сети	Большие объемы данных или неэффективная конфигурация сети могут привести к перегрузке сети, замедляя обмен данными	Оптимизировать сетевую инфраструктуру и внедрить меры качества обслуживания (qos) для определения приоритетов трафика данных
Несовместимость оборудования	Разные устройства, операционные системы или версии программного обеспечения могут быть несовместимы для беспрепятственного обмена данными	Стандартизировать форматы и протоколы или использовать промежуточное программное обеспечение для устранения несовместимости
Простои и перебои в обслуживании	Незапланированные сбои в работе сети или ее техническое обслуживание могут нарушить обмен данными	Запланировать резервирование, системы аварийного переключения и планировать обслуживание в периоды минимальной загрузки сетевой инфраструктуры
Потеря данных и повреждение	Потеря или повреждение данных во время обмена может произойти из-за проблем с сетью или ошибок программного обеспечения	Регулярно создавать резервные копии данных и внедрять механизмы проверки ошибок для обнаружения повреждений и восстановления после них
Отсутствие инструментов для совместной работы	Неэффективные инструменты совместной работы мешают обмену данными в режиме	Инвестировать в программное обеспечение для

	реального времени и успешной командной работе	совместной работы с такими функциями, как обмен документами, чат и контроль версий
Ограничения пропускной способности	Ограниченная пропускная способность может замедлить обмен данными и привести к возникновению узких мест в сетевой инфраструктуре организации	Обновить сетевую инфраструктуру и использовать инструменты управления пропускной способностью для эффективного распределения ресурсов
Проблемы управления данными	Неэффективная организация и управление данными может привести к проблемам с обменом данными, включая трудности с поиском файлов и доступом к ним	Внедрить передовые методы управления данными, такие как правильная категоризация и маркировка метаданных
Обучение и осведомленность пользователей	Пользователи вычислительной сети организации могут не знать о передовом опыте/разработках/технологиях или не иметь необходимой подготовки для эффективного использования инструментов обмена данными	Предоставить возможность обучения и ресурсы для изучения пользователями методов эффективного обмена данными
Соответствие нормативным требованиям	Обработка конфиденциальных данных без соблюдения правил их защиты данных может привести к проблемам в области соблюдения законодательства и нормативных актов	Убедиться, что методы обмена данными соответствуют нормативным актам государства и самого предприятия
Масштабируемость инфраструктуры	По мере роста организации существующие методы обмена данными могут неэффективно масштабироваться для удовлетворения растущих объемов данных или числа пользователей	Регулярно оценивать и обновлять системы обмена данными для удовлетворения потребностей в размере организации и ее инфраструктуры
Хранилища данных	Когда разные отделы или команды сотрудников используют разные методы обмена данными, могут	Поощрять межфункциональное взаимодействие между отделами и продвигать

	возникнуть хранилища данных различных форматов, что ограничивает обмен информацией между ними	унифицированные решения для обмена данными внутри всей организации
Зависимость от сторонних услуг	Чрезмерная зависимость от сторонних служб обмена данными может создать риски, если в этих службах возникнут сбои или нарушения работоспособности, например, вследствие санкционного давления (что очень актуально в наше время)	Диверсифицировать методы обмена данными, либо создать собственную внутрикорпоративную систему, а также разработать планы действий на случай непредвиденных обстоятельств
Отсутствие резервного копирования и восстановления данных	Неправильно настроенные либо отсутствующие механизмы резервного копирования и восстановления данных могут привести к потере информации и простоям в функционировании бизнес-процессов организации	Утвердить регулярные протоколы резервного копирования и восстановления данных и выполнять их периодическое тестирование
Управление затратами	Затраты на внедрение и поддержание методов обмена данными могут привести к перерасходу бюджета организации	Оптимизировать затраты, выбирая экономически эффективные решения и регулярно анализируя расходы

Выявление и решение описанных выше проблем и вызовов имеет важное значение для оптимизации обмена данными внутри компьютерной сети организации. Часто требуется сочетание передовых технологических решений, мер и политик безопасности, обучения пользователей и постоянного мониторинга сетевой инфраструктуры для обеспечения эффективных и безопасных процессов обмена данными.

1.4 Характеристика исследуемой организации и обоснование потребности в проектировании системы обмена данными

АО «ФИНТЕХ» имеет богатую тридцатилетнюю историю. Будучи правопреемником Главного вычислительного центра Министерства финансов СССР, компания смогла стать одним из ведущих разработчиков общего и специального программного обеспечения, специальных технических средств и комплексов в интересах государственных и корпоративных заказчиков [24].

Основные компетенции:

- разработка и техническая поддержка отечественной защищенной цифровой платформы «СинтезМ»;
- проектирование, внедрение и техническая поддержка защищенных центров обработки данных, ситуационных центров, автоматизированных (информационно-аналитических) систем, а также серверных, сетевых и телекоммуникационных решений;
- разработка и техническая поддержка программно-аппаратных комплексов пограничного сегмента ведомственного сегмента ФСБ России государственной системы миграционного и регистрационного учета, изготовления, оформления и контроля паспортно-визовых документов нового поколения (ГС «МИР»);
- разработка, внедрение и сопровождение автоматизированных и автоматических программных и программно-аппаратных систем и комплексов с использованием технологий биометрического контроля, в том числе автоматических пропускных модулей автоматизированной системы паспортного контроля в воздушных, морских и автомобильных пунктах пропуска через государственную границу Российской Федерации;
- проектирование, монтаж и обслуживание инженерных систем: контроля доступа, видеонаблюдения, кондиционирования;
- выполнение функций специализированной организации в области обеспечения информационной безопасности.

Примерами готовых решений компании являются [24]:

- Российская сертифицированная цифровая платформа «СинтезМ». Это комплекс программ, включающий отечественную сертифицированную операционную систему нового поколения ЗОС «СинтезМ» со встроенными средствами виртуализации и защиты.
- Автоматический пропускной модуль. АПМ АСПК – Автоматический пропускной модуль автоматизированной системы паспортного контроля предназначен для обеспечения самостоятельного прохождения пассажиром процедуры паспортного контроля в пунктах пропуска через государственную границу Российской Федерации.
- Портативное техническое средство паспортного контроля. ПТСПК – Портативное техническое средство паспортного контроля предназначено для автономного контроля лиц и транспортных средств, пересекающих государственную границу Российской Федерации в местах, не оснащенных стационарными техническими средствами паспортного контроля.
- Считыватель паспортно-визовых документов. Устройство «СПВ–7024М» предназначен для автоматизированного оптического сканирования и распознавания содержимого страниц и машиносчитываемых зон паспортно-визовых документов формата ID-1, ID-2, ID-3, а также считывания содержимого бесконтактных (RFID) интегральных схем, интегрированных в паспортно-визовые документы.
- Автоматизированное рабочее место специальной проверки документов. АРМ СПД – Автоматизированное рабочее место специальной проверки документов предназначено для исследования (экспертизы) документов, проверки их подлинности, автоматизированного учёта и документирования проверочных процедур.
- Автоматизированная дактилоскопическая информационная система «БИОМАРК». Система предназначена для проведения исследований образцов отпечатков пальцев рук человека, их верификация и идентификация для установления личности человека.

- Автоматизированная система распознавания лиц «БИОВИЗУМ». Предоставляет возможность бесконтактной идентификации лица человека.
- Автоматизированная информационная система «СКИФ-БП». Это программа для автоматизации управления общественными финансами, предназначенная для построения технологической цепочки сбора, обработки, доставки и анализа финансовой информации, поддержки процессов принятия оперативных и стратегических решений, планирования, прогнозирования и управления.

Организационная структура АО «ФИНТЕХ» является достаточно гибкой и способной к быстрой адаптации к изменяющимся рыночным требованиям и технологическим вызовам в сфере информационных технологий, что позволяет компании оставаться конкурентоспособной и инновационной на рынке информационных технологий (рисунок 1).

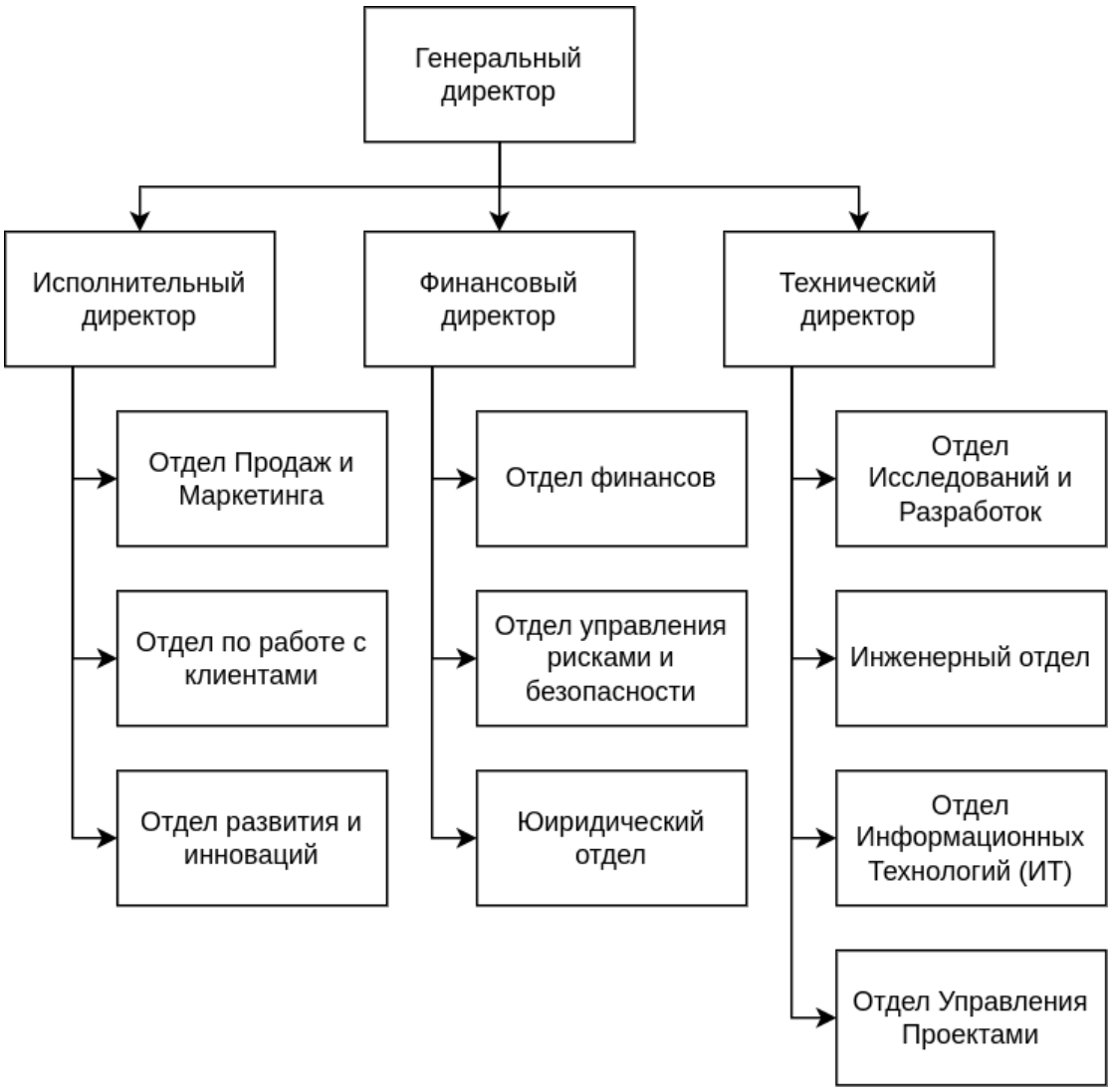

Рисунок 1 – Организационная структура АО «ФИНТЕХ»

Представленная организационная структура относится к классической функциональной, в которой подразделения организованы по функциональному принципу, где каждое из которых занимается определенным видом деятельности (маркетинг, разработка, финансы и т. д.).

Основные достоинства такой организационной структуры:

- Специализация. Каждое подразделение сосредоточено на выполнении определенных функций, что способствует хорошему качеству и производительности в данных областях.
- Четкость и упорядоченность. Организационная структура АО «ФИНТЕХ» имеет четкую и упорядоченную иерархию, что делает управление более прозрачным.

- Экономия ресурсов. Специализация приводит к экономии ресурсов, так как определенные функции выполняются в рамках одного подразделения.
- Легкость в управлении. Одному руководителю подразделения проще управлять задачами и персоналом, связанными с определенной функцией.

Организационная структура АО «ФИНТЕХ» также имеет и недостатки:

- Сложность координации. Иногда взаимодействие между функциональными подразделениями может быть сложным, особенно при решении межфункциональных задач.
- Бюрократия. Иерархическая структура может порождать бюрократию и замедлять процессы принятия решений.
- Ограничение гибкости. Функциональная структура может быть менее гибкой при необходимости быстро реагировать на изменения на рынке.
- Изоляция функций. Возможна изоляция функциональных подразделений, что может мешать коммуникации и совместной работе.

Опишем кратко функциональные обязанности сотрудников организации.

Генеральный директор руководит компанией и принимает стратегические решения.

Руководство компании состоит из таких должностей:

- Исполнительный директор. Отвечает за операционное управление компанией и осуществление стратегических планов.
- Финансовый директор. Управляет финансовыми аспектами, бюджетированием и финансовой отчетностью.
- Технический директор. Отвечает за технические аспекты, включая разработку и ИТ-инфраструктуру.

Ключевые отделы и подразделения включают в себя:

- Отдел исследований и разработок. Занимается инновациями и разработкой новых продуктов и технологий.
- Инженерный отдел. Отвечает за проектирование и инженерные решения.

- Отдел информационных технологий (ИТ). Управляет ИТ-инфраструктурой, сетями и обеспечивает ИТ-поддержку.
- Отдел продаж и маркетинга. Отвечает за маркетинг, продажи и взаимодействие с клиентами.
- Отдел управления проектами. Управляет проектами и обеспечивает их успешное выполнение.
- Юридический отдел. Обеспечивает юридическую поддержку и соблюдение законодательства РФ.
- Отдел поддержки и обслуживания клиентов. Осуществляет поддержку клиентов и управление взаимоотношениями с клиентами.
- Отдел управления рисками и безопасности. Занимается управлением рисками и обеспечением безопасности данных и операций.

Каждое подразделение включает в себя менеджеров и команды специалистов, которые работают над конкретными проектами и задачами.

Среднесписочная численность сотрудников исследуемой организации составляет 365 человек.

Что касается основных финансовых показателей работы предприятия, то они представлены в таблице 2 и рисунках 2-3 [25].

Таблица 2 – Основные финансовые показатели работы предприятия АО «ФИНТЕХ»

Показатель	Выручка, млрд. руб.	Прибыль, млн. руб.
2004	0,2	7,8
2005	0,1	4,8
2006	0,1	9,4
2007	0,1	13,6
2008	0,2	20,7
2009	0,4	14,3
2010	0,2	15,9
2011	0,5	41,8
2012	0,6	25,8
2013	0,2	-52,5
2014	0,4	-8,8
2015	0,3	22
2016	1,1	175,7

2017	1,6	89,1
2018	1,2	48,3
2019	3	55,4
2020	2,1	230,2
2021	0,7	39
2022	1,7	28,6



Рисунок 2 – Показатель «Выручка» (млрд. руб.)



Рисунок 3 – Показатель «Прибыль» (млн. руб.)

Из показателей видно, что они нестабильны, хоть и имеют положительную линию тренда. Поэтому следует предложить пути повышения эффективности деятельности организации АО «ФИНТЕХ». Они может, напрямую и не приведут к росту прибыльности предприятия, но, по крайней мере, будут ей способствовать.

Теперь следует рассмотреть существующую техническую архитектуру организации (рисунок 4).

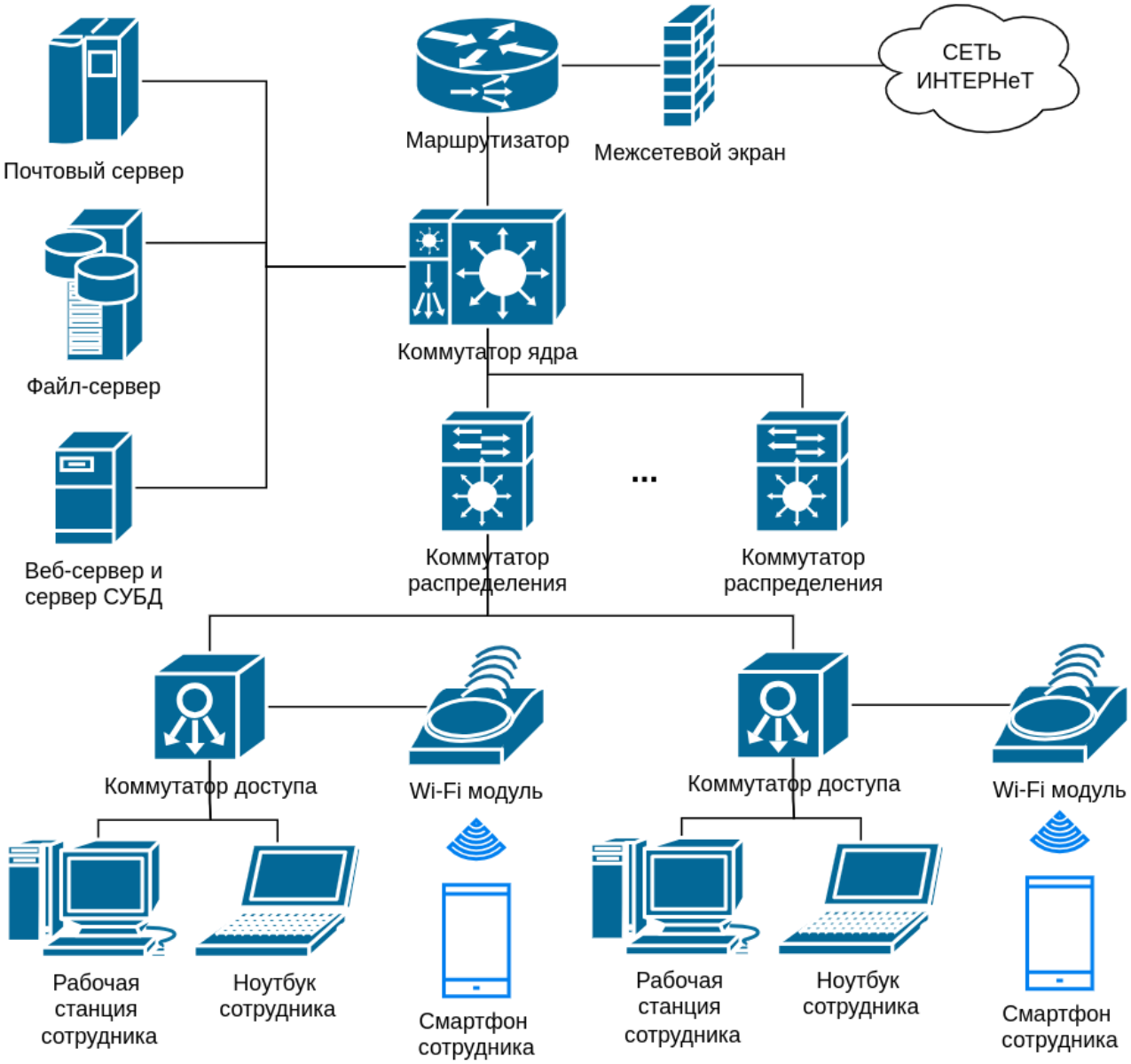

Рисунок 4 – Существующая техническая архитектура АО «ФИНТЕХ»

Архитектура имеет вид трехзвенной. Трехзвенная сетевая архитектура для построения LAN (локальной сети) включает в себя три основных компонента или уровня, которые обеспечивают функциональность и управление сетью. Кратко опишем эту архитектуру:

- Уровень доступа. Этот уровень представляет собой точку доступа для конечных устройств, таких как компьютеры, принтеры, телефоны и другие устройства. На уровне доступа используются коммутаторы для обеспечения

соединения между конечными устройствами и сетью. Здесь также реализуются механизмы контроля доступа, включая виртуальные LAN (VLAN) для разделения сегментов сети и безопасности.

- Уровень дистрибуции. Этот уровень выполняет роль агрегации и маршрутизации данных между уровнем доступа и уровнем ядра. На уровне дистрибуции установлены коммутаторы, которые обеспечивают маршрутизацию между VLAN, балансировку нагрузки, а также безопасность.
- Уровень ядра. Этот уровень представляет собой основную сетевую инфраструктуру, которая обеспечивает высокую пропускную способность и связность между различными уровнями доступа и дистрибуции, а также высокоскоростной доступ к серверам. На уровне ядра используются высокоскоростные коммутаторы, маршрутизаторы и другое оборудование для обеспечения быстрого и эффективного перемещения данных.

Компания АО «ФИНТЕХ» использует разнообразное программное обеспечение (рисунок 5).

Программная среда функционирует в пространстве операционной системы Astra Linux 1.7. На предприятии используется СУБД MySQL и PostgreSQL.

Бухгалтерия работает в рамках соответствующей системы от отечественной фирмы 1С.

Почтовый сервер организован системой RuPost (производство РФ).

FTP-сервер работает в рамках системы Samba.

Из программной архитектуры видно, что компания стремиться использовать отечественное или бесплатно распространяемое ПО.

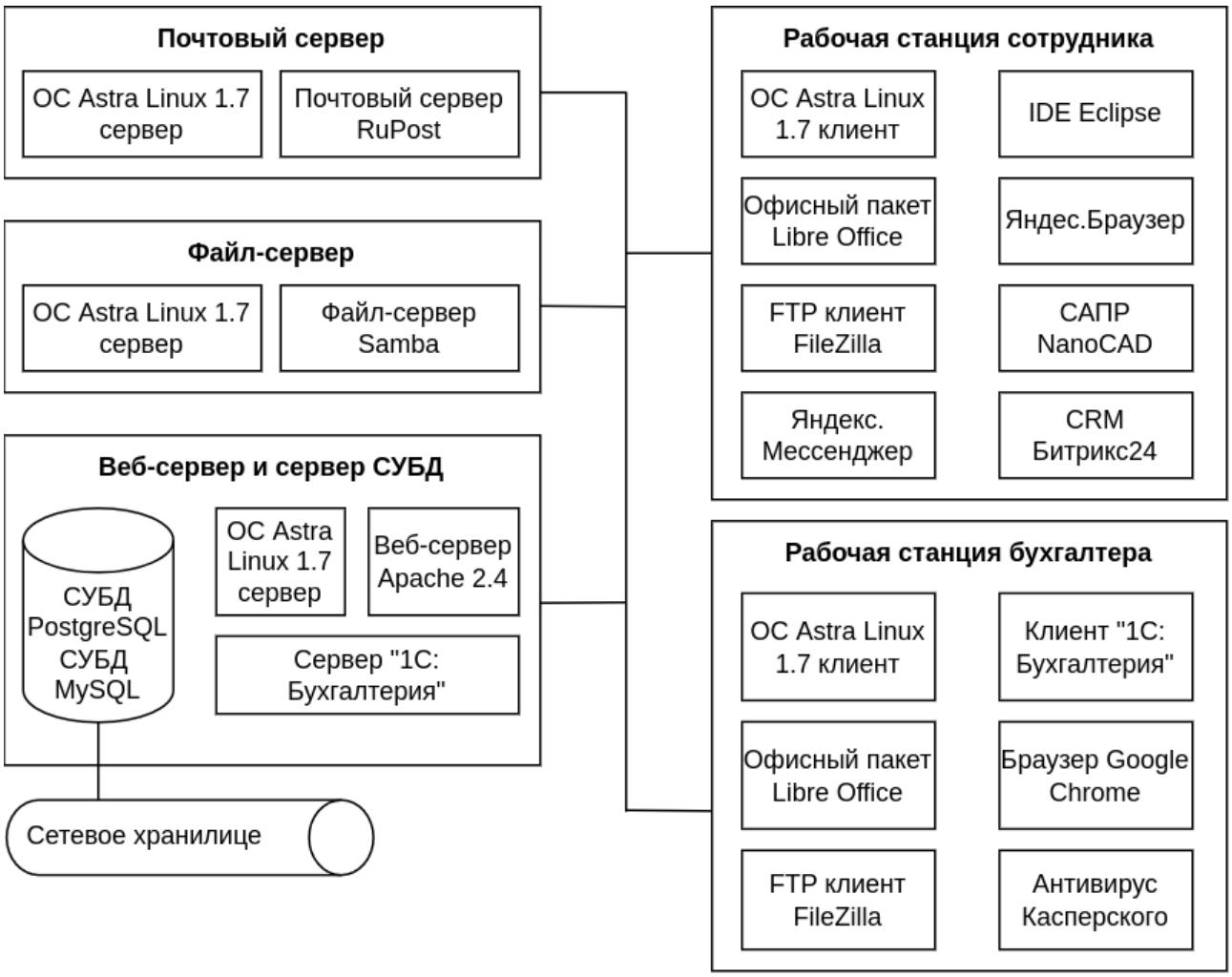

Рисунок 5 – Программное обеспечение АО «ФИНТЕХ»

Что касается обмена данными в вычислительной сети исследуемой организации, то он осуществляется через такие каналы:

- FTP-сервер Samba;
- электронная почта через сервер RuPost;
- мессенджер «Яндекс.Мессенджер»;
- файлообменники в сети Интернет;
- флеш-карты сотрудников.

Часто встает задача передачи файлов таким образом, чтобы гарантировать их шифрование, чтобы никто из сотрудников, кому они не предназначены, не могли их прочитать, даже если каким-то образом скачают.

К сожалению, текущий обмен с помощью разных технологий и отсутствие четкой стандартизации приводит к тому, что сложно обеспечить требуемый

уровень безопасности. Дело в том, что файлы на файловом сервере хранятся в незащищенном виде, и сотрудники, имеющие доступ к серверу, могут их читать, то есть получать доступ к информации, которая им может не предназначаться.

Некоторые сотрудники при передаче файлов их архивируют с паролем, однако зачастую эти пароли очень простые, и такие архивы легко взломать методом простого подбора слов/символов. То есть опять же, отсутствует четкая стандартизация информационного обмена, кто-то повышает безопасность обмена, а кто-то передает файлы через мессенджеры в группах сотрудниках, когда все, кто хочет может их скачать и просмотреть.

Поэтому следует предложить способ обмена файлами, который будет обеспечивать, с одной стороны, удобство, а, с другой стороны, безопасность.

Важным является то, что система должна быть корпоративной, то есть разработанной и используемой внутри организации, а не сторонний сервис (файлообменник) в сети Интернет, ведь только так можно будет действительно гарантировать безопасность конфиденциальной корпоративной информации.

Выводы по разделу 1

В первом разделе выполнен анализ предметной области.

Вначале рассмотрены существующие обмена данными в вычислительной сети организации: протоколы FTP и FTPS, электронная почта, общие сетевые диски, облачные службы хранения и обмена файлами, инструменты для совместной работы, службы обмена мгновенными сообщениями, службы удаленного рабочего стола.

Также были определены общие требования, которые могут предъявляться для систем обмена данным: эффективность и скорость, надежность, безопасность, доступность, масштабируемость, взаимодействие, совместимость, контроль версий, управление данными, мобильный доступ, автоматизация, резервное копирование и восстановление данных, обучение и поддержка, соответствие и нормативные требования, экономическая эффективность, конфиденциальность, отзывы пользователей и улучшения в работе системы.

Далее выполнен Анализ существующих проблем и вызовов при обмене данными в вычислительной сети организации.

В заключении дана характеристика исследуемой организации. Показаны ее услуги, описана организационная структура, которая является функиональной.

Из обзора финансовых показателей видно, что они нестабильны, хоть и имеют положительную линию тренда. Поэтому следует предложить пути повышения эффективности деятельности организации АО «ФИНТЕХ».

Архитектура вычислительной сети имеет вид трехзвенной. Трехзвенная сетевая архитектура для построения LAN (локальной сети) включает в себя три основных компонента или уровня, которые обеспечивают функциональность и управление сетью.

Программная среда функционирует в пространстве операционной системы Astra Linux 1.7.

Что касается обмена данными в вычислительной сети исследуемой организации, то он выполняется с помощью разных технологий и отсутствие четкой стандартизации приводит к тому, что сложно обеспечить требуемый уровень безопасности.

Поэтому следует предложить способ обмена файлами, который будет обеспечивать, с одной стороны, удобство, а, с другой стороны, безопасность.

2. ВЫБОР ТЕХНОЛОГИЙ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС

2.1 Выбор технологий для разработки программного обеспечения для организации обмена данными в вычислительной сети организации

2.1.1 Выбор типа приложения

Программное обеспечение для организации обмена данными в вычислительной сети организации АО «ФИНТЕХ» может быть таких типов:

- веб-приложение;
- десктопное (настольное) приложение;
- мобильное приложение.

Рассмотрим особенности каждого типа, а затем сравним их по критериям.

Веб-приложение – это программная система, которая доступна через веб-браузер. Она может размещаются на удаленном сервере, и пользователи будут получать к ней доступ через Интернет [1].

В рамках исследуемого предприятия веб-приложение может быть размещено на корпоративном веб-сервере, с доступом к ней по ЛВС компании.

Веб-приложение позволяет пользователям взаимодействовать с ним на различных устройствах без установки на само устройство.

Настольное приложение – это программная система, установленная и выполняемая непосредственно на локальном компьютере или устройстве. Она имеет более широкий доступ к ресурсам устройства и операционной системы, чем веб-приложение.

Настольное приложение может быть разработано для различных операционных систем, таких как Windows, macOS и Astra Linux и не сможет напрямую запускаться в другой ОС (сможет через виртуализацию).

Мобильное приложение – это программная система, разработанная специально для мобильных устройств, таких как смартфоны и планшеты. Оно

оптимизировано для сенсорных экранов и прочих мобильных особенностей, таких как, например, геолокация.

Сравнение данных типов приложений представлено в таблице 4.

Таблица 4 – Сравнение типов приложений для проектирования системы

Параметр	Веб-приложение	Настольное приложение	Мобильное приложение
Доступность	Доступно через браузеры на любом устройстве с доступом в ЛВС организации	Требует установки на конкретном компьютере или устройстве	Доступно только на мобильных устройствах
Доступ к ресурсам	Ограничено в доступе к ресурсам на клиентской стороне	Имеет более широкий доступ к ресурсам устройства, таким как файловая система и аппаратное обеспечение	Имеет доступ к мобильным функциям, таким как GPS, камера и различные датчики
Кросслатформен-ность	Может быть доступно на различных платформах, но может иметь ограничения	Может быть разработано для разных операционных систем, но трудозатраты повышаются	Разрабатывается для конкретных платформ (iOS, Android) и требует разных версий для разных систем
Установка и обновление	Не требует установки и автоматически обновляется на сервере	Требует установки и обновления через ручную переустановку	Требует установки и обновления вручную
Скорость и производитель-ность	Может зависеть от скорости ЛВС, но она высокая	Часто предоставляет более высокую производительность, так как работает непосредственно на устройстве	Обычно оптимизировано для мобильных платформ и имеет хорошую производительность
Разработка и обслуживание	Обычно более универсально и может быть доступно на разных браузерах	Требует разработки для каждой операционной системы и обновления отдельных версий	Требует разработки для каждой мобильной платформы, что повышает затраты

Исходя из анализа возможных альтернативных типов приложений, выбор пал на веб-приложение, потому что оно наиболее универсально при минимизации затрат на его разработку. Поскольку шифрование/дешифрование будет осуществляться на сервере, то для клиентского аппаратного обеспечения не будет предъявляться серьезных требований к производительности.

2.1.2 Выбор языков программирования

Система будет работать как на пользовательском устройстве, так и на сервере, поэтому следует выбрать технологии для работы в браузере на стороне сотрудника, а также на сервере.

Для веб-приложения базовыми технологиями являются [7]:

- HTML – язык гипертекстовой разметки веб-документов;
- CSS – язык описания стилей веб-документа.

Что касается клиентской стороны, то стандартом, по сути, на сегодняшний день, является Javascript. Это популярный и мощный язык для веб-приложений, в том числе предназначенных для обмена данными в компьютерной сети организации, по таким причинам [8, 14, 18]:

- Взаимодействие на стороне клиента. JavaScript запускается в веб-браузере пользователя, обеспечивая динамичный и интерактивный пользовательский интерфейс. Пользователи могут вводить, манипулировать и просматривать данные без необходимости частого обновления страниц.
- Асинхронная связь. JavaScript обеспечивает асинхронную связь с серверами с помощью таких технологий, как Fetch API или XMLHttpRequest, что позволяет обмениваться данными в режиме реального времени, не нарушая пользовательский интерфейс.
- Межбраузерная совместимость. JavaScript поддерживается всеми современными веб-браузерами, что делает его отличным выбором для создания веб-приложения обмена данными, которое будет одинаково работать на разных платформах.

- Богатая экосистема. JavaScript имеет обширную экосистему библиотек и фреймворков, таких как React, Angular и Vue.js, jQuery, которые упрощают разработку веб-приложений и расширяют их функциональность.
- Открытый исходный код. Природа JavaScript с открытым исходным кодом означает, что разработчики могут получать доступ к широкому спектру ресурсов и инструментов и вносить в них свой вклад.
- Сторонние API. JavaScript может взаимодействовать с различными сторонними API, обеспечивая интеграцию с внешними службами и источниками данных. Это имеет решающее значение для обмена данными и совместимости в сети организации.
- Безопасность. JavaScript обеспечивает функции проверки и безопасности на стороне клиента, помогая предотвратить распространенные уязвимости веб-приложений, такие как межсайтовый скриптинг (XSS) и подделка межсайтовых запросов (CSRF).
- Масштабируемость. JavaScript можно использовать для создания масштабируемых приложений, которые подходят для растущего числа пользователей и обмена данными.
- Поддержка сообщества. обширное сообщество разработчиков JavaScript и ресурсы доступны для помощи и обучения, что упрощает поиск решений проблем и доступ к лучшим практикам.
- Адаптивный дизайн. JavaScript можно использовать для создания адаптивных пользовательских интерфейсов, которые адаптируются к различным размерам экрана и устройствам, что важно для организации, где пользователи получают доступ к приложению с разных устройств.
- Модульность. JavaScript продвигает модульный подход к программированию, позволяя разбить приложение на управляемые компоненты, что упрощает его разработку и обслуживание.

Разработка серверной части ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» возможна на таких языках, как PHP, Python и Java. Выполним их сравнение по набору параметров (таблица 5) [10, 16, 23].

Таблица 5 – Сравнение языков PHP, Python и Java

Параметр	PHP	Python	Java
Веб-фреймворки	Имеет популярные веб-фреймворки, такие как Laravel и Symfony, которые упрощают веб-разработку и предлагают широкий спектр пакетов для обработки файлов и аутентификации.	Имеет такие фреймворки, как Django и Flask, которые выделяются своей гибкостью и безопасностью. В Python также есть библиотеки, такие как Django Storages, для управления хранилищем файлов.	Имеет такие платформы, как Spring и JavaServer Faces (JSF) для веб-разработки. Экосистема Java отлично подходит для крупномасштабных приложений, таких как службы хостинга и обмена файлами.
Обработка файлов	Имеет встроенные функции и библиотеки для загрузки, обработки и хранения файлов. Отлично хорошо подходит для обработки загрузки файлов.	Модули Python OS и Shutil могут управлять файловыми операциями. Такие библиотеки, как Boto, можно использовать для взаимодействия с сервисами хранения.	Java предоставляет классы в пакете java.io для операций с файлами. Он также хорошо подходит для интеграции с решениями облачного хранения через API.
Безопасность	Имеет историю уязвимостей безопасности, но в современных версиях и платформах PHP улучшены методы обеспечения безопасности.	Читабельность кода Python и надежная модель безопасности делают его менее подверженным уязвимостям безопасности при условии, что разработчики следуют передовым методах и подходам.	строгая типизация Java и надежная модель безопасности способствуют ее надежной безопасности. Однако безопасность во многом зависит от реализации приложения.
Масштаби-руемость	PHP-приложения можно масштабировать с помощью фреймворков и методов оптимизации.	Масштабируемость Python зависит от выбранной платформы и архитектуры. Он подходит для средних и крупных приложений.	Java может создавать масштабируемые приложения корпоративного уровня. Это хороший выбор для крупных систем.
Производи-	Простой в	Может обеспечить	Высокая

тельность	использовании и производительный.	хорошую производительность при оптимизации.	производительность и эффективность, особенно для крупномасштабных приложений.
Интеграция базы данных	Поддерживает различные базы данных через драйверы и ORM	Поддерживает базы данных через такие библиотеки, как SQLAlchemy и Django ORM.	Взаимодействует с базами данных через Hibernate для ORM и JDBC для прямого взаимодействия.
Сообщество и экосистема	Имеет большое и активное сообщество с широким спектром сторонних пакетов и фреймворков.	Имеет большое сообщество и обширные библиотеки для веб-разработки и обработки файлов.	Имеет зрелую и надежную экосистему, особенно подходящую для крупномасштабных и корпоративных приложений.
Развертывание и хостинг	PHP-приложения могут размещаться на широком спектре веб-серверов, часто с меньшими затратами.	Приложения Python можно развертывать на различных хостинговых платформах, включая облачные сервисы.	Приложения Java обычно размещаются на выделенных серверах или в облачной инфраструктуре из-за их требований к ресурсам.

В качестве языка программирования серверной части ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» выбран PHP, так как он производительный, простой в изучении и использовании, обеспечивает должный уровень безопасности и является стандартом для разработки веб-ориентированных систем.

2.1.3 Выбор системы управления базой данных

На сервере АО «ФИНТЕРХ» установлены такие СУБД: MySQL и PostgreSQL. Это популярные системы управления реляционными базами данных с открытым исходным кодом, каждая из которых имеет свои сильные и слабые стороны.

Для разработки ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» выбрана MySQL, потому что она имеет такие преимущества по сравнению с PostgreSQL [4, 20]:

- Простота использования. MySQL считается более простой в настройке и управлении для несложных приложений, которым является проектируемое ПО. Она имеет простой и удобный интерфейс командной строки и графические инструменты администрирования.
- Производительность для рабочих нагрузок с интенсивным чтением. MySQL известна своей производительностью в сценариях с интенсивным чтением, особенно при правильной настройке. Она отлично подходит для ситуаций, когда нужно быстро получить и отобразить данные.
- Большое сообщество. MySQL имеет большое и хорошо зарекомендовавшее себя сообщество пользователей, а это означает, что существует множество ресурсов, форумов и учебных пособий, которые помогут устранять проблемы и находить их решения.
- Механизм хранения по умолчанию. Механизм хранения MySQL по умолчанию, InnoDB, предлагает надежную поддержку соответствия ACID (атомарность, согласованность, изоляция, долговечность), что делает его подходящим для приложения, требующего поддержки транзакций.
- Репликация. Возможности репликации MySQL развиты и хорошо документированы, что позволяет относительно легко настроить репликацию «главный-подчиненный» или «главный-главный» для обеспечения высокой доступности и отказоустойчивости подсистемы хранения данных.

2.1.4 Выбор графического фреймворка

CSS-фреймворки в основном используются в веб-разработке для создания адаптивных и визуально привлекательных пользовательских интерфейсов. Хотя они напрямую не обеспечивают обмен данными, они могут значительно улучшить дизайн и удобство использования приложения для обмена данными.

Кратко рассмотрим возможные альтернативы при выборе фреймворка [9, 11, 19]:

- Bootstrap. Это один из наиболее широко используемых CSS-фреймворков. Он имеет адаптивную сетку, предварительно разработанные компоненты пользовательского интерфейса и различные стили CSS, идеально подходит для создания простого и современного пользовательского интерфейса для приложения обмена данными. Bootstrap также предлагает настраиваемые темы и широкий спектр сторонних расширений.
- Foundation. Это надежная CSS-инфраструктура, ориентированная на доступность. Она предлагает гибкую систему сеток, компоненты пользовательского интерфейса и различные варианты стилей. Foundation хорошо подходит для разработки удобных и доступных интерфейсов, что может быть важно для программного обеспечения для обмена данными.
- Bulma. Это легкий и современный CSS-фреймворк, основанный на Flexbox. Его легко освоить, и он предлагает простую и гибкую сеточную систему. Bulma идеально подходит для создания элегантных и минималистичных проектов, которые могут подойти для приложения обмена данными, где важна простота.
- Semantic UI. Он известен своей интуитивно понятной и удобной для пользователя структурой HTML, предоставляет широкий спектр компонентов пользовательского интерфейса и систему тем, которая позволяет легко настроить внешний вид приложения. Semantic UI может помочь в создании удобных и хорошо структурированных интерфейсов.
- Materialize CSS. Это CSS-фреймворк, соответствующий рекомендациям Google по дизайну материалов. Он предназначен для создания визуально привлекательных и единообразных пользовательских интерфейсов.
- Tailwind CSS. Он использует другой подход, предоставляя служебные классы, которые позволяют проектировать пользовательский интерфейс, применяя атрибуты класса непосредственно в HTML-разметке. Он

предлагает широкие возможности настройки и гибкость, что делает его подходящим для создания уникальных проектов.

- UIkit. Это легковесная и модульная платформа с упором на модульность и гибкость. Она предоставляет множество компонентов и возможностей настройки для создания индивидуального пользовательского интерфейса для приложения обмена данными.

Для разработки ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» выбран фреймворк Bootstrap, потому что он широко распространен, очень прост и понятен в изучении и использовании. Имеются множество форумов и ресурсов с шаблонами, которые можно применять для быстрой и эффективной разработки.

2.1.5 Выбор алгоритма шифрования файлов

Алгоритмы шифрования бывают симметричные и ассиметричные. Симметричное шифрование использует один и тот же ключ для шифрования и расшифровки, что весьма эффективно, но требует безопасного механизма распределения ключей, в то время как асимметричное шифрование универсально для безопасной связи и обмена ключами, но требует больших вычислительных затрат.

Будет использовано симметричное шифрование, так как нагрузка на сервер является важным параметром. А ключи надо обязать передавать через личные сообщения. При этом даже если сообщение будет перехвачено внешним по отношению к предприятию субъектом, то он еще должен будет получить доступ к файлу, а он на защищенном сервере внутри предприятия, доступ к которому можно получить только изнутри ЛВС организации.

Рассмотрим кратко популярные алгоритмы [2, 3, 13]:

- AES. Является одним из самых популярных алгоритмов симметричного шифрования и широко используется для шифрования файлов и данных. Он предоставляет различные размеры (128, 192 и 256 бит) для различной степени безопасности.

- DES. Был одним из первых алгоритмов симметричного шифрования, но в течение некоторого времени оставался открытым. Его улучшенная версия, Triple DES (3DES), все еще используется в некоторых технологиях для обеспечения совместимости, но не рекомендуется для новых приложений из-за слабой безопасности.
- Blowfish. Это алгоритм симметричного шифрования с относительно высокой скоростью выполнения. Он обеспечивает хороший баланс между безопасностью и производительностью и может использоваться для приложения шифрования файлов.
- Twofish. Twofish – это алгоритм, созданный как альтернатива DES и обеспечивает хорошую безопасность.
- Serpent. Это метод симметричного шифрования, который отличается высокой стойкостью и сравнительно медленной производительностью. Его можно использовать для шифрования файлов, где безопасность имеет приоритет над скоростью.
- CAST5. Это алгоритм с относительно низкой сложностью, который можно использовать для шифрования файлов со средним уровнем безопасности.

Выполним сравнение алгоритмов AES, Blowfish, CAST5 (таблица 6).

Таблица 6 – Сравнение алгоритмов AES, Blowfish, CAST5

Параметр	AES	Blowfish	CAST5
Криптостой-кость	Считается безопасным и пользуется доверием благодаря своей стойкости к различным атакам.	Разработан с учетом высоких требований безопасности и считается надежным.	Обеспечивает хорошую безопасность, но считается, что он имеет меньший запас безопасности по сравнению с AES и Blowfish.
Производи-тельность	Хорошая производительность, особенно при аппаратной	Имеет высокую скорость и эффективность, что делает его	Обеспечивает приемлемую производительность, что делает его

	реализации. Он эффективен и быстр, что делает его подходящим для различных приложений.	пригодным для различных приложений.	полезным в сценариях, где необходим баланс между безопасностью и производительностью.
Размеры ключей	Позволяет использовать три различных ключа длиной 128, 192 или 256 бит.	Поддерживает размеры ключей от 32 до 448 бит, что обеспечивает гибкость в выборе подходящего уровня безопасности.	Поддерживает размеры ключей от 40 до 128 бит.

Для реализации ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» будет использован алгоритм Blowfish, так как он имеет высокую скорость, безопасность, различные размеры ключей в широком диапазоне, что позволит при необходимости настраивать скорость его работы при масштабировании системы.

Писать алгоритм на языке PHP нет необходимости, достаточно взять готовую библиотеку, реализующую его [26], и подключить к проекту.

2.2 Формирование требований к разработке ПО

Формирование требований к ПО для организации обмена данными в вычислительной сети организации включает в себя описание функциональных и нефункциональных параметров системы.

Название проекта: «Служба безопасного хостинга файлов».

Цель проекта: разработать веб-хостинг с шифрованием файлов и несколькими вариантами продолжительности хранения, позволяющий также осуществлять однократную загрузку.

Функциональные требования:

- пользователи (сотрудники АО «ФИНТЕХ») могут загружать файлы в приложение;
- автоматическая генерация ключевого слова (секретного ключа), чтобы предотвратить случаи использования простых ключей;
- файлы должны быть зашифрованы при загрузке с использованием безопасного алгоритма шифрования;
- автоматическая генерация ссылки на доступ к файлу;
- помещение данных о файлах в централизованное хранилище (за исключением ключей шифрования);
- автоматизированная расшифровка файлов по переданному системе ключу шифрования, генерация ссылки для загрузки восстановленного файла на устройство получателя;
- если файл был скачан, либо истекло время его хранения в системе, файл должен быть безвозвратно удален.

Пользователи могут выбрать один из следующих вариантов продолжительности хранения:

- 7 дней;
- 14 дней;
- 30 дней;
- хранение до однократной загрузки файла.

Требования к безопасности:

- система должна использовать для шифрования/дешифрования криптостойкий алгоритм Blowfish;
- длина ключа должна быть не менее 128 бит;
- ключи должны быть случайными, чтобы предотвратить атаки перебором;
- ключи не должны храниться в базе данных.

Требования к пользовательскому интерфейсу:

- следует разработать веб-интерфейс для удобного управления файлами;

- пользовательские формы должны содержать четкие инструкции о том, как пользоваться услугой по обмену данными в вычислительной сети организации;
- формы интерфейса должны одинаковым образом отображаться в различных браузерах и операционных системах, на различных устройствах;
- формы должны автоматически подстраиваться под различные размеры и разрешения экранов;
- интерфейс должен быть реализован с применением кроссплатформенного фреймворка.

Требования к масштабируемости:

- служба должна быть спроектирована так, чтобы обрабатывать масштабируемое количество пользователей и файлов;
- архитектура построения веб-системы должна быть трехзвенная клиент-серверная.

Требования к производительности:

- система должна обеспечивать быструю и оперативную загрузку и скачивание файлов;
- время шифрования/дешифрования для документов:
  - объемом до 10 МБ должно быть не более 5 сек;
  - объемом до 100 МБ должно быть не более 30 сек.

Требования к надежности:

- сервис должен обеспечивать целостность и надежность данных;
- применяемые технологии должны поддерживать организацию резервного копирования базы данных и/или зашифрованных файлов.

Требования к программному обеспечению:

- технологии на стороне клиента:
  - язык гипертекстовой разметки документов HTML;
  - язык описания стилей CSS;
  - язык программирования сценариев JavaScript;

- графический фреймворк Bootstrap;

- технологии на стороне сервера:
  - язык программирования PHP;
  - реляционная СУБД MySQL.

Требования к развертыванию:

- развертывание веб-сервиса должно быть осуществлено на защищенном веб-сервере в локальной вычислительной сети организации АО «ФИНТЕХ».

Требования к сопровождению:

- следует регулярно обслуживать сервер и его программное окружение для решения проблем безопасности, обновлений программного обеспечения;
- в случае потребности следует проводить обучение и поддержку пользователей системы.

2.3 Проектирование программного обеспечения для организации обмена данными в вычислительной сети организации

Для проектирования системы воспользуемся возможностями, которые предоставляет визуальный язык моделирования UML. Это простой, но в то же время очень гибкий и мощный инструмент для описания бизнес-процессов и реализующих их информационных систем.

Язык содержит множество диаграмм, которые позволяют спроектировать те или иные особенности будущего приложения.

Начнем с диаграммы прецедентов, которая описывает функционал системы с общих позиций, выделяя ее базовые способности (действия). Диаграмма представлена на рисунке 6.

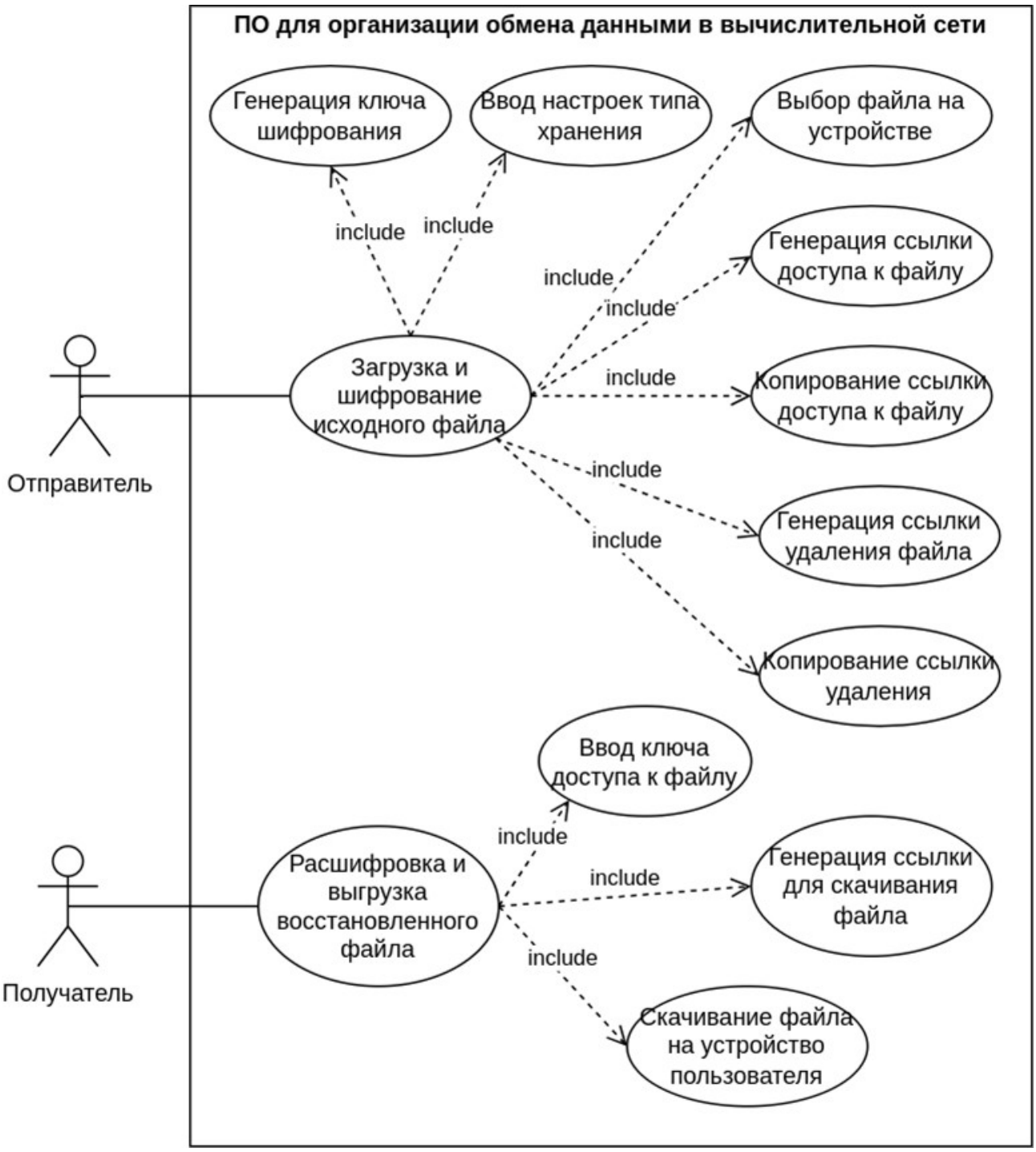

Рисунок 6 – Диаграмма прецедентов ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ»

Выделяют два типа пользователей:

- отправитель;
- получатель.

Неважно, из какого они отдела и какие у них должности. С точки зрения системы (и здравого смысла) в общем случае любому сотруднику предприятия

может потребоваться отправить тот или иной файл кому-то из персонала компании. Поэтому роли отделов здесь не выделяются, а только отправитель и получатель файла.

Отправитель должен осуществить загрузку и шифрование исходного файла посредством системы. Связь типа «include» на диаграмме говорит о том, что все соответствующие прецеденты обязательны к исполнению. То есть при загрузке и шифровании файла система осуществляет:

- генерацию ключа шифрования;
- ввод настроек типа хранения (одно скачивание, 7, 14 или 30 дней);
- выбор файла на устройстве;
- генерация ссылки доступа к файлу;
- копирование ссылки доступа к файлу.

Получатель файла производит расшифровку и выгрузку восстановленного файла. Этот процесс включает в себя такие операции:

- ввод ключа доступа к файлу;
- генерация системой ссылки на скачивание файла;
- переход по данной ссылке и скачивание расшифрованного файла на устройство пользователя.

Недостатком диаграммы вариантов использования является тот факт, что она показывает действия, но не показывает их четкую последовательность, которая в данной системе будет присутствовать.

Поэтому представим диаграмму последовательностей (рисунок 7). Она описывает, как следует из названия, последовательность операций в рамках процесса обмена данными между отправителем и получателем в рамках вычислительной сети организации.

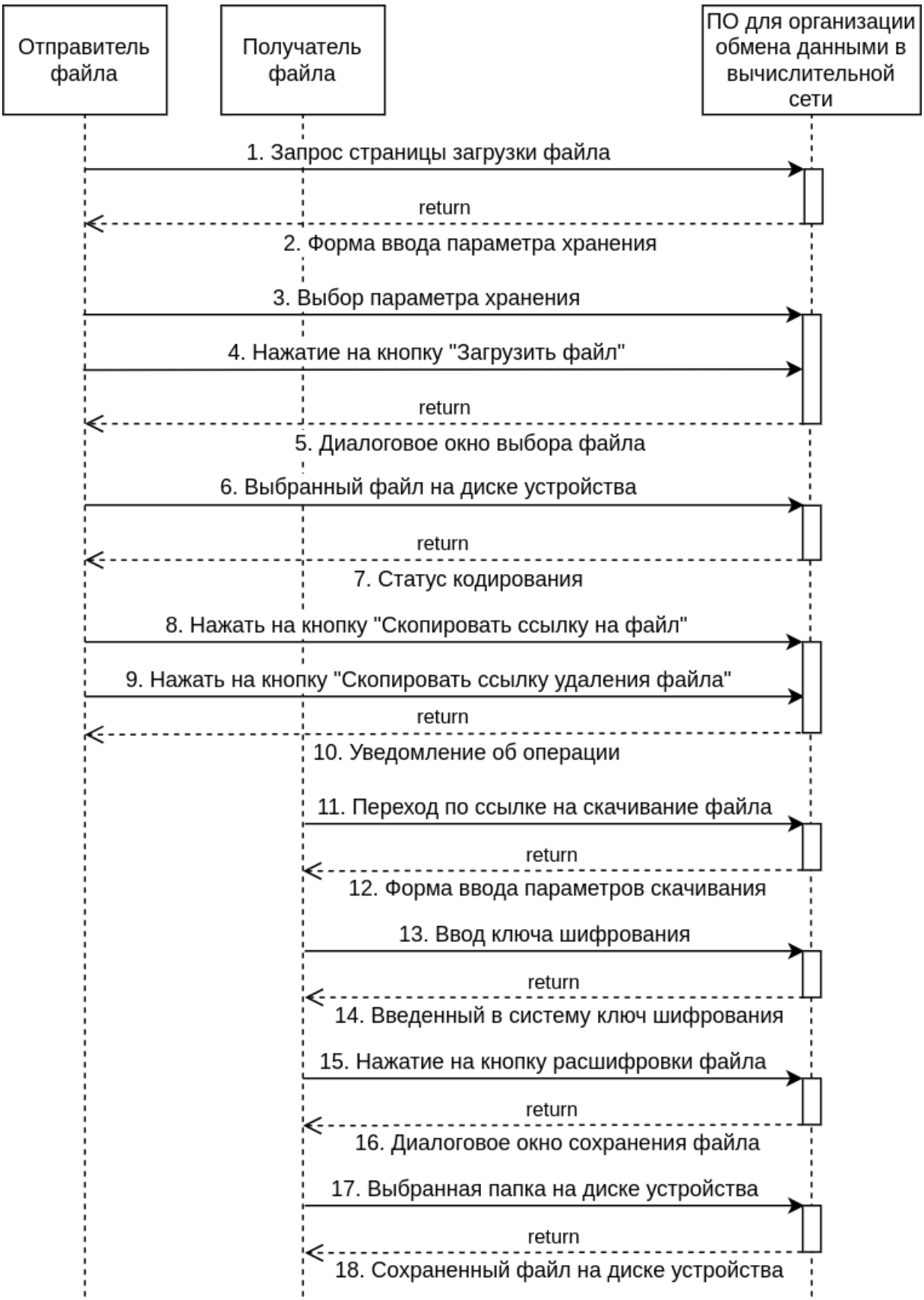

Рисунок 7 – Диаграмма последовательностей процесса обмена данными между отправителем и получателем в рамках вычислительной сети организации

Читать диаграмму нужно сверху-вниз и слева-направо. Половина диаграммы (если смотреть сверху-вниз) относится к процессу загрузки и шифрованию файла, а вторая – к его дешифрации и скачиванию на устройство получателя.

Также составим диаграмму компонентов, которая покажет архитектуру построения ПО для обмена файлами (рисунок 8).

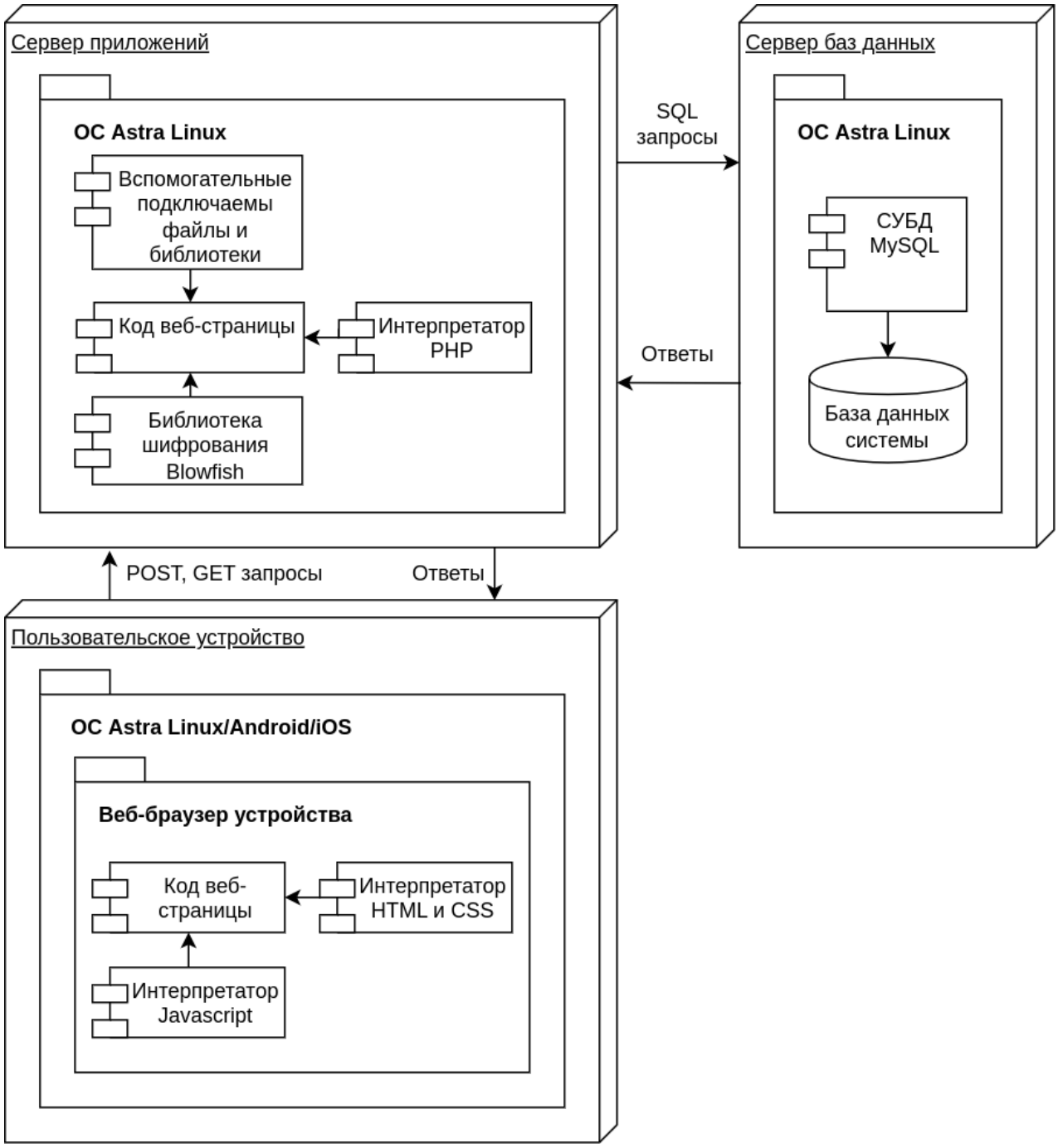

Рисунок 8 – Архитектура построения ПО для обмена данными между отправителем и получателем в рамках вычислительной сети организации

Проектируемое ПО реализует трехзвенную сетевую архитектуру – это тип архитектуры информационных систем, который разделяет систему на три основных компонента или слоя (звена).

Первое звено (первый уровень) – это клиентская часть системы, которая включает в себя интерфейс пользователя, веб-браузер. Пользователи (сотрудники) взаимодействуют с системой через это звено.

Второе звено (второй уровень) – представляет собой бизнес-логику и обработку данных. Этот слой содержит приложение на языке PHP, которое выполняет функции, необходимые для обработки запросов пользователей и выполнения бизнес-логики, то есть загрузку, шифрование/дешифрование и скачивание файлов.

Третье звено (третий уровень) – это слой данных, который включает в себя базу данных. На нем хранятся и управляются все данные, необходимые для функционирования системы.

Преимущества трехзвенной сетевой архитектуры:

- Модульность и гибкость. Каждое звено может быть разрабатывается, масштабируется и обновляется независимо, что упрощает управление и обслуживание системы.
- Разделение ответственности. Разделение на три звена позволяет четко определить, где выполняются различные функции, улучшая общую архитектуру системы.
- Устойчивость к нагрузкам. Путем масштабирования и распределения нагрузки между звеньями, система может легче справляться с высокими нагрузками.

Недостатки трехзвенной сетевой архитектуры:

- Сложность. Разработка и поддержание трехзвенной архитектуры может быть более сложным, но это характерно для больших и высоконагруженных систем, что к конкретной задаче не относится.

- Задержка. Каждый запрос пользователя должен проходить через все три звена, что может вызвать небольшую задержку в обработке. Это плата за гибкость и масштабируемость.
- Зависимость от сети. Передача данных между звеньями требует надежной вычислительной сети. Проблемы с сетью могут повлиять на работоспособность системы. Однако система как раз и задумывается для работы в ЛВС АО «ФИНТЕХ».

Что касается базы данных, то, поскольку в приложении не предусмотрена авторизация, а также какие-либо сложные настройки, то она будет состоять из одной таблицы (рисунок 9).

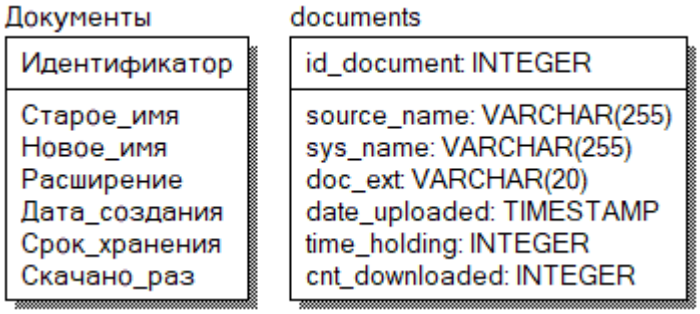

Рисунок 9 – Логическая и физическая модели БД ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ»

Логическая модель содержит имена таблицы и полей на кириллице, а в физической модели осуществлен переход к названиям на латинице, а также добавлены типы данных.

В следующем разделе будет осуществлена непосредственная реализация программного кода веб-приложения.

Выводы по разделу 2

Во втором разделе осуществлен выбор технологий, а также осуществлено проектирование ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ».

Исходя из анализа возможных альтернативных типов приложений, выбор пал на веб-приложение, потому что оно наиболее универсально при

минимизации затрат на его разработку. Поскольку шифрование/дешифрование будет осуществляться на сервере, то для клиентского аппаратного обеспечения не будет предъявляться серьезных требований к производительности.

В качестве языка программирования клиентской части выбран Javascript, так как он является стандартом для веб-приложений на сегодняшний день, он мощный и гибкий.

В качестве языка программирования серверной части выбран PHP, так как он производительный, простой в изучении и использовании, обеспечивает должный уровень безопасности и является стандартом для разработки веб-ориентированных систем.

Выбор СУБД был между MySQL и PostgreSQL, так как они уже функционируют на сервере предприятия. Выбрана MySQL, потому что она более проста в использовании, имеет высокую производительность на чтение, надежную поддержку соответствия ACID, поддерживает механизмы репликации.

Также выбран фреймворк Bootstrap, потому что он широко распространен, очень прост и понятен в изучении и использовании. Имеются множество форумов и ресурсов с шаблонами, которые можно применять для быстрой и эффективной разработки.

В качестве алгоритма шифрования выбран Blowfish, так как он имеет высокую скорость, безопасность, различные размеры ключей в широком диапазоне, что позволит при необходимости настраивать скорость его работы при масштабировании системы.

Далее сформированы функциональные и нефункциональные требования к проектируемому программному обеспечению.

Для проектирования системы были применены возможности, которые предоставляет визуальный язык моделирования UML.

Разработаны диаграммы:

- вариантов использования;
- последовательностей;

- компонентов.

Также составлены логическая и физическая модели базы данных, которая состоит из одной таблицы «Документы». Логическая модель содержит имена таблицы и полей на кириллице, а в физической модели осуществлен переход к названиям на латинице, а также добавлены типы данных.

3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС

3.1 Реализация программного обеспечения для организации обмена данными в вычислительной сети организации

Вначале представим структурную схему программных модулей ПО для обмена данными (рисунок 10).

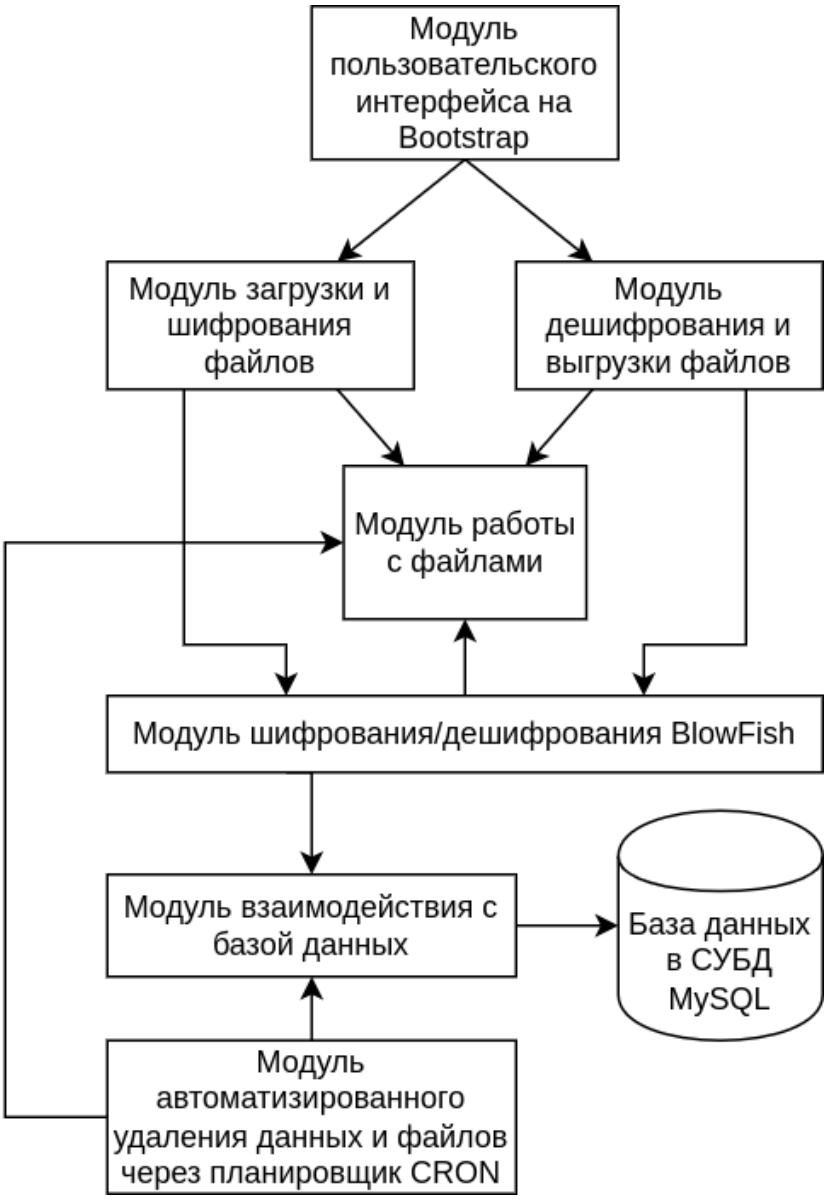

Рисунок 10 – Структурная схема ПО для обмена данными между отправителем и получателем в рамках вычислительной сети организации

Дадим некоторые пояснения. Данные модули – логические, то есть физически они могут присутствовать в разных файлах на диске.

Модуль работы с файлами реализуется средствами самой ОС и позволяет выбирать файл для загрузки и скачивания на устройстве пользователя.

Чтобы автоматически удалять файлы, у которых истек срок хранения, применяется CRON-планировщик, встроенный в веб-сервер. Он по расписанию (раз в час) запускает скрипт, который считывает из БД информацию о сроках хранения файлов. Для тех из них, у которых срок хранения исчерпан, выполняется:

- физическое удаление зашифрованного файла с диска сервера;
- удаление записи в БД о файле.

Для работы веб-приложения нужен файл конфигурации, в котором задаются базовые параметры настроек (рисунок 11).

config.php

```php
<?php
    define('F_DB','filesharing');
    define('F_USER','root');
    define('F_PASSWORD','');
    define('F_HOST','localhost');
    define('F_PREFIX','http://localhost/filesharing.fintech.ru/');
    define('F_PREF','filesharing.fintech.ru');
?>
```

Рисунок 11 – Файл конфигурации ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ»

Структура файлов проекта представлена на рисунке 12 и в таблице 7.

Таблица 7 – Описание физических файлов и директорий проекта

Тип	Название	Назначение
Директория	filesharing.fintech.ru	Папка с приложением
Директория	ajax	Папка для хранения модулей, которые вызываются динамически по технологи AJAX
Файл	file_decrypt.php	Модуль дешифрования файлов
Файл	file_upload.php	Модуль загрузки и шифрования файлов
Директория	css	Папка для хранения файлов описания стилей веб-страниц
Директория	files	Папка для хранения зашифрованных файлов
Директория	fonts	Папка для хранения подключаемых

		шрифтов
Директория	inc	Папка с модулями включения в проект
Файл	Biglnteger.php	Сторонний модуль реализации вычислений с большими числами
Файл	blowfish.php	Сторонний модуль реализации алгоритма симметричного шифрования Blowfish
Файл	config.php	Файл-конфигурация системы
Директория	js	Папка для скриптов Javascript
Файл	bootstrap.min.js	Служебный скрипт фреймворка Bootstrap
Файл	bootstrap.mini.js	Служебный скрипт фреймворка Bootstrap
Файл	file_upload.js	Скрипт для старта и контроля динамической загрузки файлов на сервер и их шифрования
Файл	jquery-3.3.1 .slim.min.js	Сторонняя подключаемая библиотека jQuery
Файл	jquery.min.js	Сторонняя подключаемая библиотека jQuery
Файл	popper.min.js	Служебная библиотека подсказок для фреймворка Bootstrap
Файл	popper.min1.js	Служебная библиотека подсказок для фреймворка Bootstrap
Файл	tether.min.js	Служебная библиотека позиционирования для фреймворка Bootstrap
Директория	tmp	Временная директория для файлов
Файл	cron.php	Модуль автоматизированной проверки срока хранения файлов и их удаления с сервера и базы данных
Файл	filesharing.sql	Скрипт создания БД в СУБД MySQL
Файл	index.php	Точка входа в приложение. Главный скрипт
Файл	финтех.png	Картинка для тестирования работы ПО

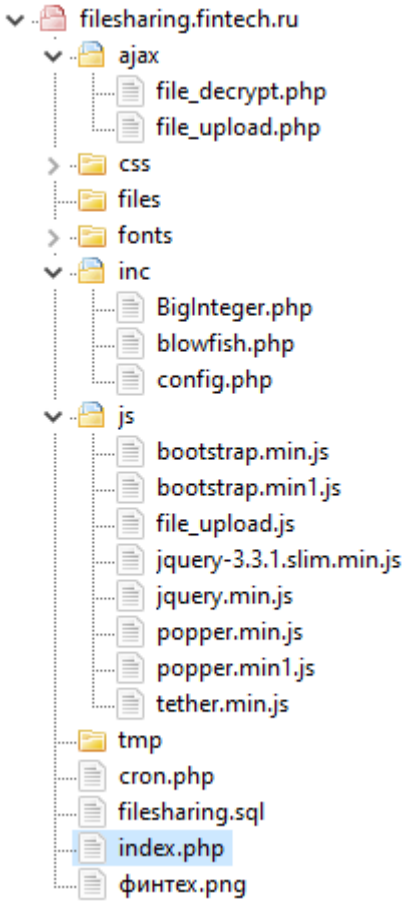

Рисунок 12 – Структура файлов проекта

Ниже представлен код генерации имени файла, его кодирования, а также записи в БД:

```
// Имя файла
while(true)
{
     $filename = uniqid('', true);//.'.'.$inf['doc_ext'];
     $path_file = $uploaddir.$filename;
     if (!file_exists($path_file))
          break;
}

// Шифрование файла посредством BlowFish
$source_symbols = file_get_contents($name);
$alg_key = trim($_POST['key']);
$alg_mode = Blowfish::BLOWFISH_MODE_EBC;
$alg_padding = Blowfish::BLOWFISH_PADDING_RFC;
$alg_iv = null;
$blow_object = new Blowfish($alg_key, $alg_mode, $alg_padding,
$alg_iv);
$encr_symbols = $blow_object->encrypt($source_symbols, $alg_key,
$alg_mode, $alg_padding, $alg_iv);
file_put_contents($path_file, $encr_symbols);
```

```
// Запись в БД
$connection = mysqli_connect(F_HOST, F_USER, F_PASSWORD, F_DB);
$query_string = «INSERT INTO documents(source_name, sys_name,
doc_ext, time_holding) values('$source_name', '$filename',
'$doc_ext', $time_holding);»;
$query_result = mysqli_query($connection, $query_string);
if(!$query_result)
{
     echo «Ошибка вставки в БД»;
     return false;
}
mysqli_close($connection);

echo F_PREFIX.$filename;
```

В модуле index.php, если в него был передан параметр «Путь к зашифрованному файлу», то запускается такой код:

```
if(isset($parts[2]) && trim($parts[2])!=””) // Присутствует
параметр
{
     $sys_name = trim($parts[2]);

     // Выяснение наличия файла в системе
     $connection = mysqli_connect(F_HOST, F_USER, F_PASSWORD,
F_DB);
     $query_string = «SELECT * from files WHERE sys_name like
'$sys_name'«;
    $query_result = mysqli_query($connection, $query_string);
    if(!$query_result)
     echo '<div class=«row»><div class=«col-md-12»><div
role=«alert» class=«alert alert-danger»>Ошибка подключени к
БД</div></div></div>';
    $query_data = mysqli_fetch_assoc($query_result);
     mysqli_close($connection);
     $id_document = $query_data['id_document'];
     $source_name = $query_data['source_name'];
     $doc_ext = $query_data['doc_ext'];
     $date_uploaded = $query_data['date_uploaded'];
     $time_holding = $query_data['time_holding'];
     $cnt_downloaded = $query_data['cnt_downloaded'];

     КОД ФОРМЫ СКАЧИВАНИЯ ФАЙЛА (ОПУЩЕН)
}
```

Ниже представлен код, который осуществляет декодирование файла:

```
// Декодирование файла
$source_symbols = file_get_contents($path_to_file);
$alg_mode = Blowfish::BLOWFISH_MODE_EBC;
```

```
$alg_padding = Blowfish::BLOWFISH_PADDING_RFC;
$alg_iv = null;
$blow_object = new Blowfish($alg_key, $alg_mode, $alg_padding,
$alg_iv);
$decoded_symbols = $blow_object->decrypt($source_symbols,
$alg_key, $alg_mode, $alg_padding, $alg_iv);

// Копирование в папку для временного хранения
while(true)
{
     $tmppath = uniqid('', true);
     if (!is_dir($tmppath))
     break;
}
mkdir($dir_base.'/tmp/'.$tmppath);
$decoded_symbols_file_1 = $tmppath.»/».iconv(«UTF-8»,
«Windows-1251», $source_name).».».$doc_ext;
file_put_contents($dir_base.'/tmp/'.$decoded_symbols_file_1,
$decoded_symbols);
$decoded_symbols_file_2 = $tmppath.»/».$source_name.».».$doc_ext;
echo $decoded_symbols_file_2;
```

Таким образом, разработан программный код, который использует стороннюю библиотеку симметричного шифрования Blowfish для защиты файлов в рамках процесса информационного обмена между сотрудниками организации АО «ИНТЕХ».

3.2 Пример функционирования программного обеспечения для организации обмена данными в вычислительной сети организации

Теперь перейдет к описанию примера работы ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ».

После обращения к адресу системы на сервере будет отображена страница загрузки файла (рисунок 13).

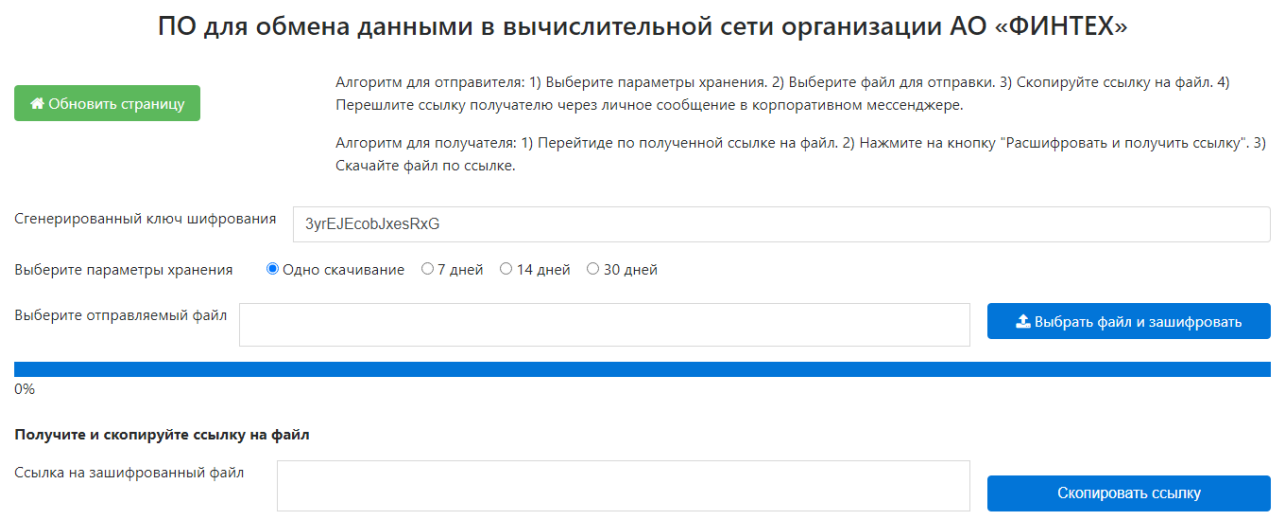

Рисунок 13 – Форма загрузки файла

На этой форме присутствует краткая инструкция для пользователей. Тут же автоматически генерируется ключ шифрования. Отправитель может выбрать один из параметров хранения файла, а также сам файл.

В качестве тестового файла выберем скрин веб-сайта компании АО «ИНТЕХ» (рисунок 14).

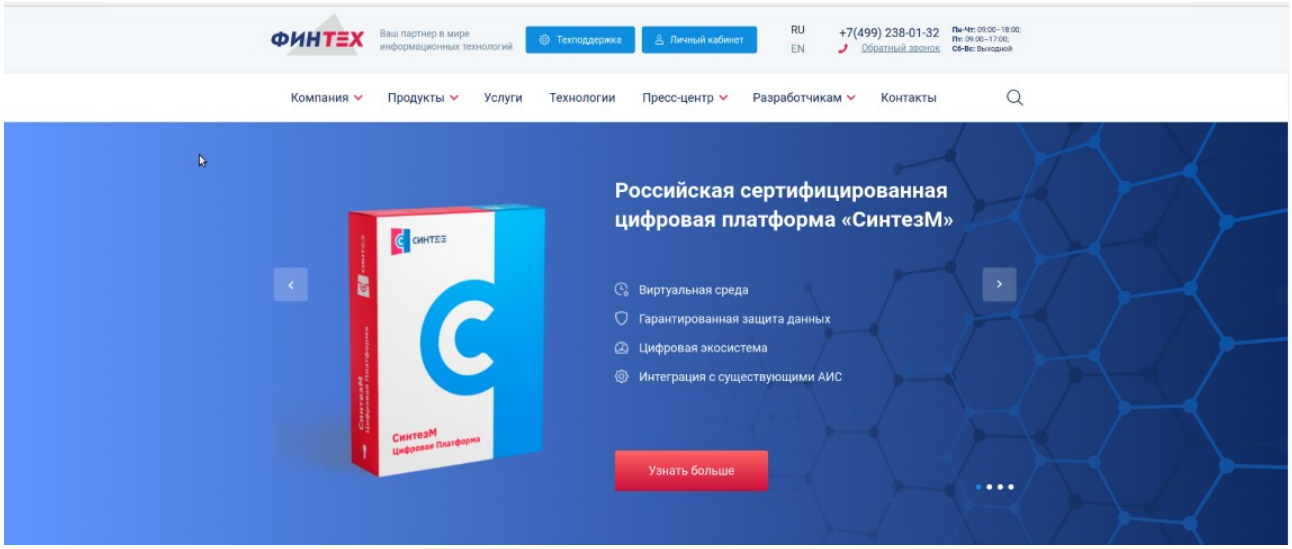

Рисунок 14 – Тестовый файл для загрузки

При нажатии на кнопку выбора файла открывается соответствующая форма (рисунок 15).

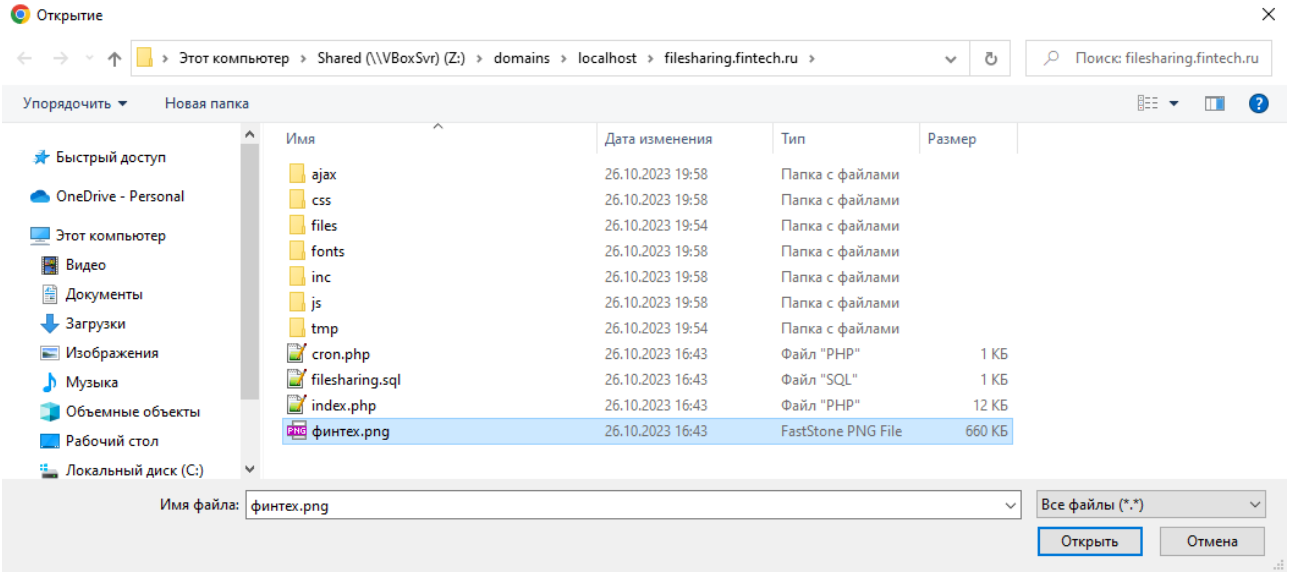

Рисунок 15 – Форма выбора файла на устройстве

После выбора он автоматически шифруется, и пользователю выводится ссылка и пароль на файл (рисунок 16), которые можно скопировать (рисунок 17).

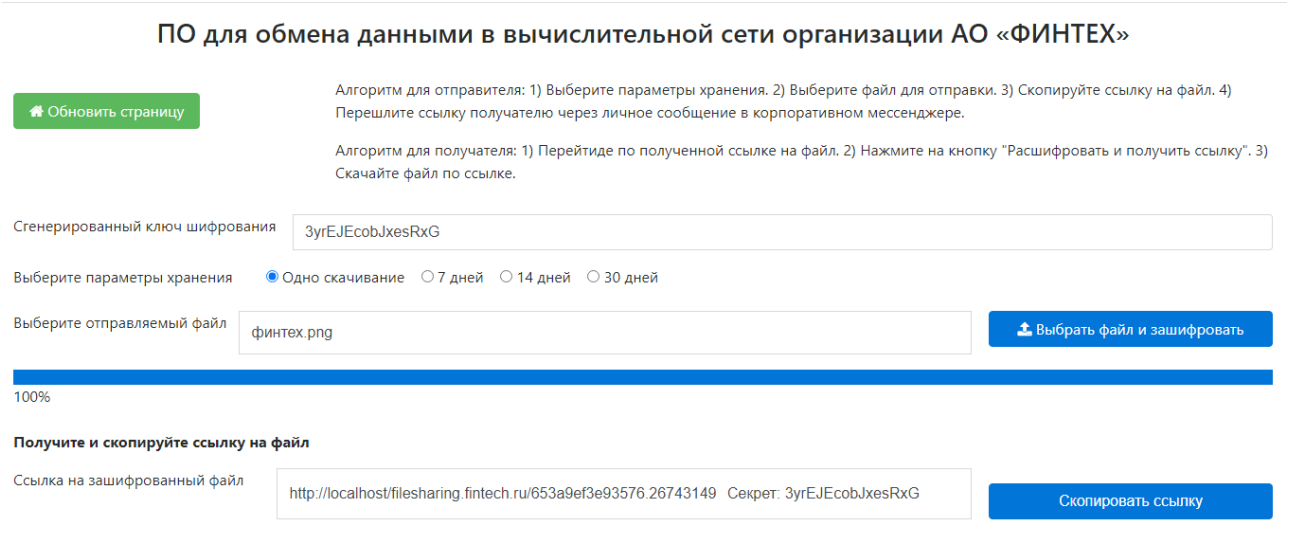

Рисунок 16 – Автоматически сгенерированная ссылка и секретный пароль к файлу

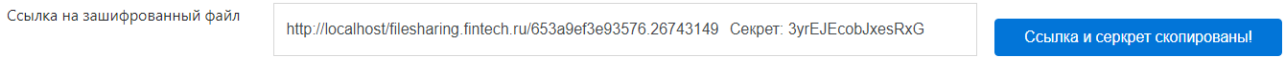

Рисунок 17 – Уведомление о копировании параметров в буфер обмена

Файл в зашифрованном виде можно просмотреть на диске (рисунки 18-19).

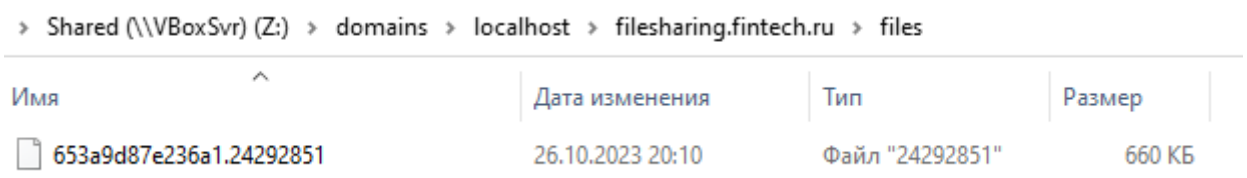

Рисунок 18 – Зашифрованный файл



Рисунок 19 – Зашифрованный файл в текстовом редакторе

При попытке открыть зашифрованную картинку ничего не получится (рисунок 20).

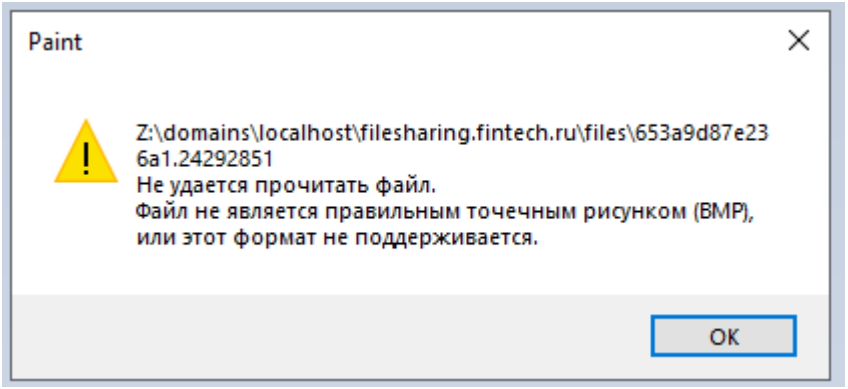

Рисунок 20 – Невозможность чтения зашифрованного файла как картинки

Получив ссылку, отправитель в личном сообщении через корпоративный мессенджер передает ссылку и пароль на скачивание получателю, который переходит по этой ссылке (рисунок 21).

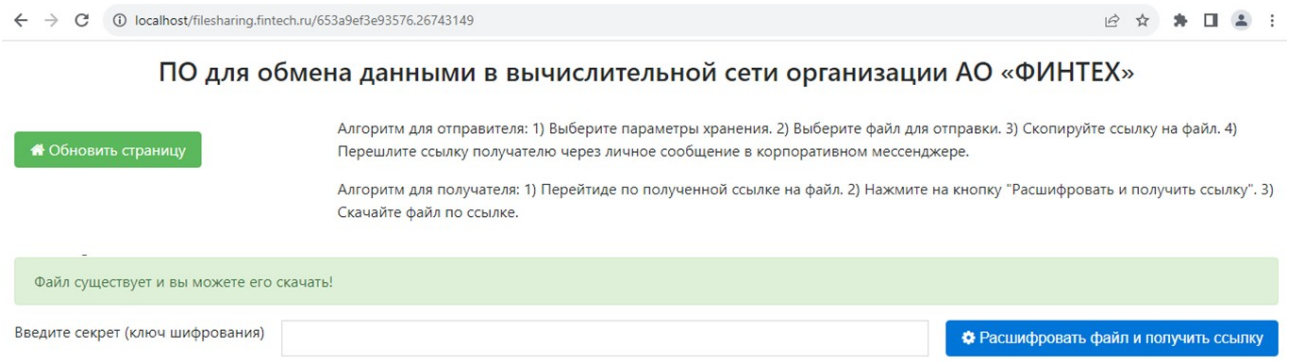

Рисунок 21 – Форма расшифровки файла

Далее отправитель вставляет пароль доступа к файлу и нажимает кнопку. Происходит расшифровка файла и формирование ссылки для его скачивания (рисунок 22).

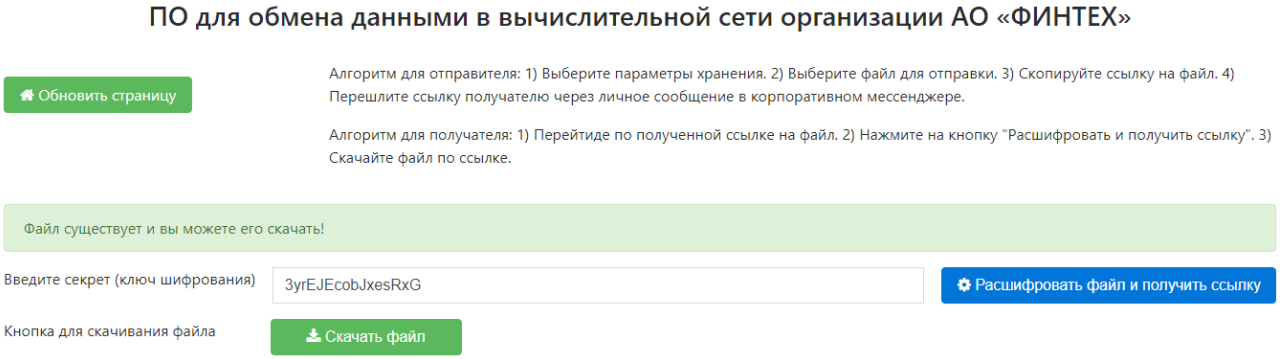

Рисунок 22 – Файл успешно расшифрован и сформирована ссылка на скачивание

Скачанный файл можно просмотреть (рисунок 23). Это та же исходная картинка. Если бы хотя бы один бит был бы поврежден, то картинка могла бы не отобразиться. Раз она отображается, то это говорит о работоспособности используемого алгоритма шифрования и его корректных настройках.

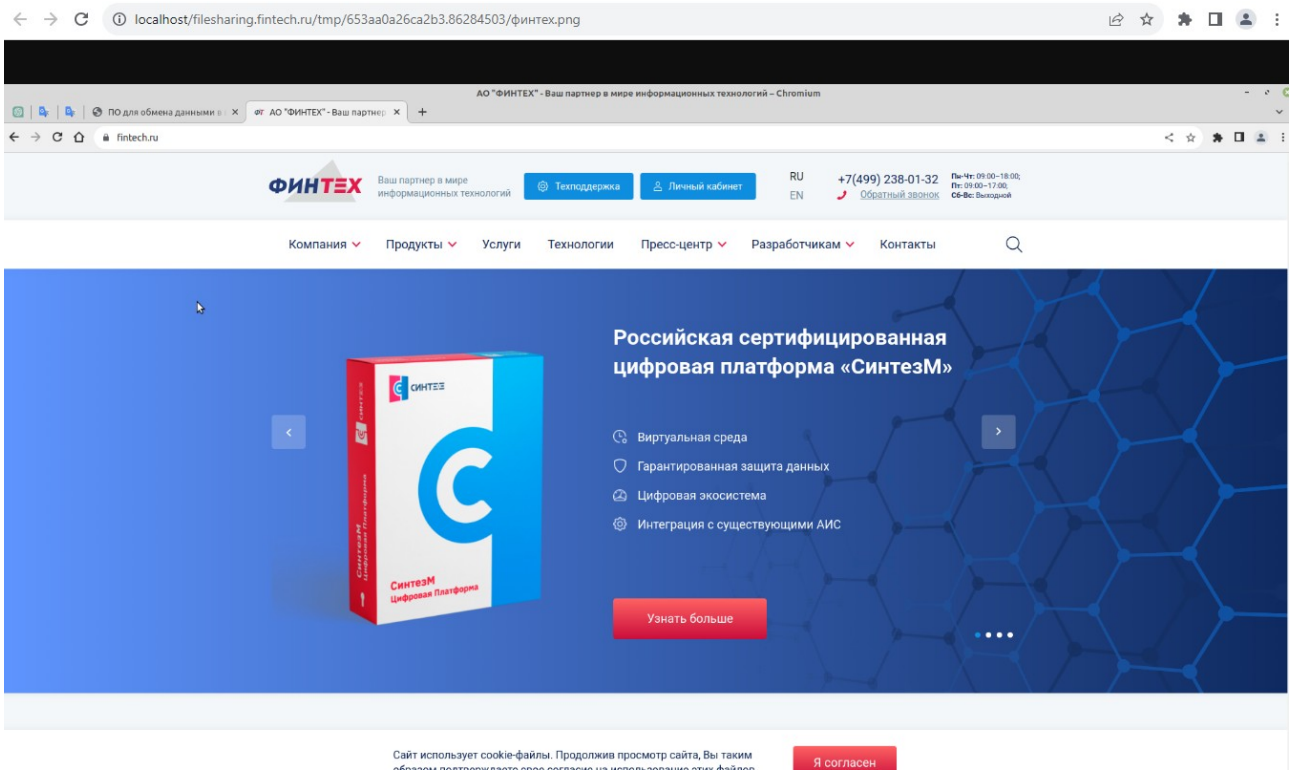

Рисунок 23 – Расшифрованный файл

Если же перейти по ссылке, для которой файла нет, то система это отследит и выведет уведомление (рисунок 24).

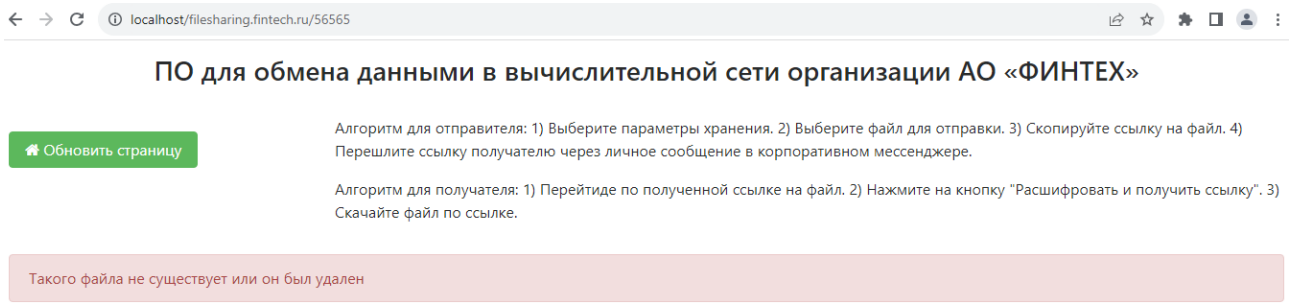

Рисунок 24 – Уведомление об отсутствии файла в системе

Таким образом, разработанное веб-приложение может быть использовано в компании АО «ИНТЕХ» для организации обмена данными в формате файлов/документов посредством их передачи через вычислительную сеть.

Следует отметить, что модульность архитектуры позволит в будущем улучшать функционал системы по мере необходимости, и это не будет сопряжено со значительными финансовыми и трудовыми расходами на содержание команды разработчиков.

3.3 Оценка экономической эффективности проекта

Себестоимость разработки ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» включает в себя набор показателей:

- время разработки;
- трудозатраты разработчика;
- заработная плата разработчика;
- затраты на расходные материалы;
- затраты на ЭВМ;
- затраты на электроэнергию, потребляемую ЭВМ.

В таблице 8 представлены этапы проекта, которые нужны для расчета его длительности в днях.

Таблица 8 – Расчет длительности проекта

№ этапа	Наименование работ	Количество дней
1	Исследование предметной области	4
2	Анализ и моделирование бизнес-процессов организации	3
3	Поставнока задачи на проект	2
4	Выбор технологий проектирования	3
5	Проектирование системы	5
6	Реализация системы	10
7	Внедрение на предприятии	2
Итого:		29

Длительность разработки и внедрения составляет 29 дней.

В проекте участвует один разработчик с часов ставкой в 300 рублей.

При продолжительности работ в 29 дней или 232 часа (при восьмичасовой рабочей смене) заработная плата соствит: Зп = 232 * 300 = 69 600 руб.

Взносы в социальные фонды РФ составляют 30,2%:

Зф = 69 600 * 0,302 = 21 019.2 рублей.

Для разработки был куплен б/у ПК стоимостью 23 000 рублей.

В таблице 9 показаны расходы на материалы.

Таблица 9 – Расходы на материалы

Затраты	Количество	Стоимость, руб.	Сумма, руб.
Бумага А4, упаковок	1	400	400
Флеш-карта 32 ГБ, штук	1	600	600
Итого			1000

Так как разработка велась на ПК, то следует рассчитать затраты на потребленную электроэнергию (таблица 10).

Таблица 10 – Затраты на потребленную электроэнергию

Изделие	Мощность, кВт	Количество часов	Количество кВт	Стоимость кВт/ч, руб.	Сумма, руб.
Б/у ПК	0,5	232	116	6,43	745,88
Итого:					745,88

Можно рассчитать себестоимость разработки (таблица 11, рисунок 25).

Таблица 11 – Себестоимость разработки и внедрения ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ»

Статьи затрат	Сумма, руб	Доля от общей суммы затрат
Заработная плата	69600,00	60,33%
Страховые взносы	21019,20	18,22%
Эксплуатационные расходы	745,88	0,65%
Затраты на расходные материалы	1000,00	0,87%
Стоимость технических средств разработки системы	23000,00	19,94%
Итого:	115365,08	100%

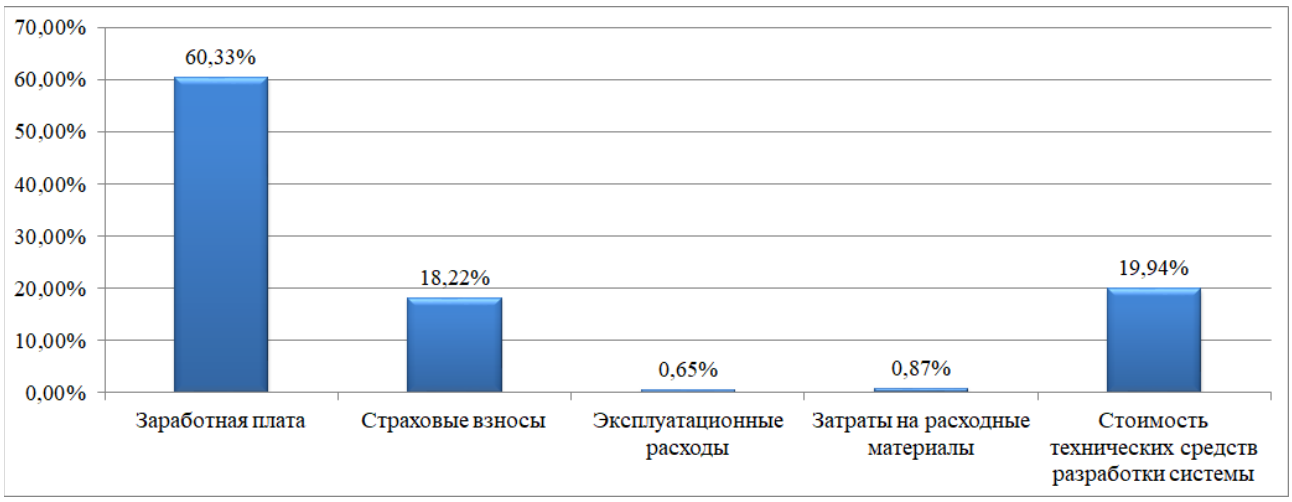

Рисунок 25 – Структура себестоимости разработки и внедрения ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ»

Себестоимость разработки и внедрения ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ» составляет 115 365,08 рублей.

Экономический эффект от внедрения заключается, в первую очередь, в увеличении скорости выполнения обмена данными внутри организации, а также в повышении безопасности, удобства обмена.

В таблице 12 представлен расчет экономии трудозатрат сотрудников.

Таблица 12 – Расчет экономии трудозатрат сотрудников на обмен данными

	Время на ручное выполнение, часов	Время на выполнение с программой, часов	Экономия, часов
Передача файла	0,25	0,025	0,225

В год осуществляется передача 100 000 файлов. Поэтому общая экономия составит 375 часов. При средней стоимости рабочего часа сотрудника в 400 рублей экономия составит 375 * 400 = 150 000 рублей в год.

Представим это наглядно (рисунки 26-27).

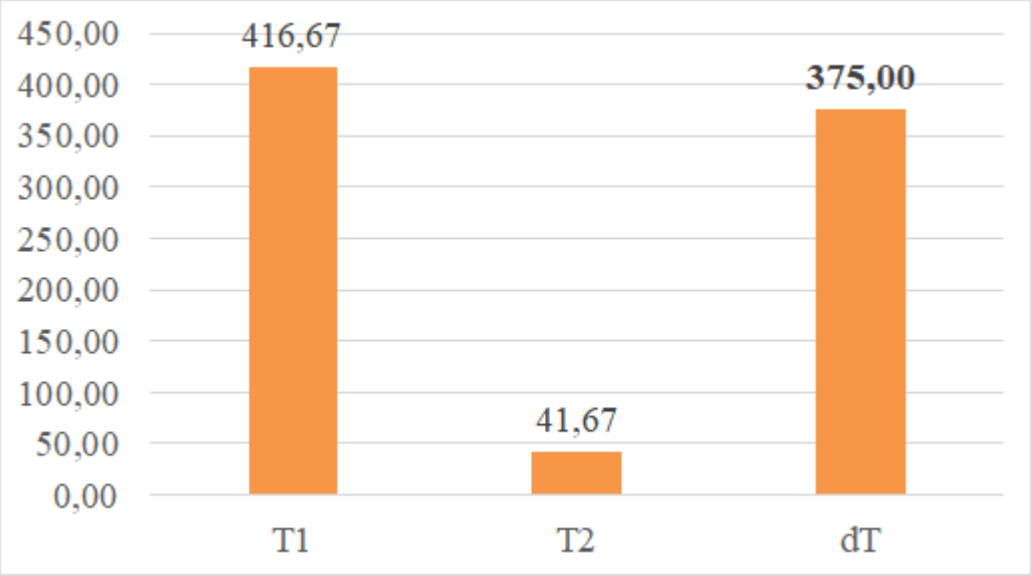

Рисунок 26 – Снижение трудозатрат на обмен данными

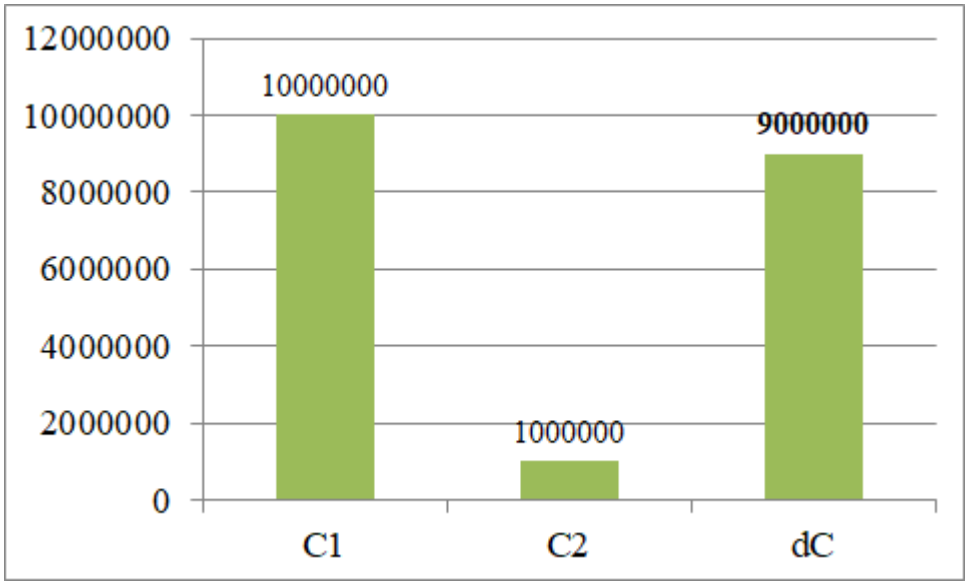

Рисунок 27 – Снижение финансовых затрат на обмен данными

Несложно посчитать срок окупаемости проекта:

Ток = 115 365,08 / 150 000 = 0,77 года.

Это значение является приемлемым для АО «ИНТЕХ». Поэтому можно говорить об эффективности проекта разработки и внедрения ПО для обмена данными в вычислительной сети организации.

Выводы по разделу 3

В третьем разделе выполнена программная реализация ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ».

Описана структурная схема веб-приложения.

Далее описан пример работы системы. Разработанное веб-приложение может быть использовано в компании АО «ИНТЕХ» для организации обмена данными в формате файлов/документов посредством их передачи через вычислительную сеть.

Модульность архитектуры позволит в будущем улучшать функционал системы по мере необходимости, и это не будет сопряжено со значительными финансовыми и трудовыми расходами на содержание команды разработчиков.

Расчет экономической эффективности показал, что:

- себестоимость проекта составляет 115 365,08 рублей;
- экономия трудозатрат составляет 375 часов в год;
- экономия финансовых затрат на обмен данными составляет 150 000 рублей в год;
- срок окупаемости проекта составит 0,77 года.

Это значение является приемлемым для АО «ИНТЕХ». Поэтому можно говорить об эффективности проекта разработки и внедрения ПО для обмена данными в вычислительной сети организации.

ЗАКЛЮЧЕНИЕ

В выпускной работе выполнено проектирования и реализация программного обеспечения для организации обмена данными в вычислительной сети организации.

В первом разделе выполнен анализ предметной области.

Вначале рассмотрены существующие обмена данными в вычислительной сети организации: протоколы FTP и FTPS, электронная почта, общие сетевые диски, облачные службы хранения и обмена файлами, инструменты для совместной работы, службы обмена мгновенными сообщениями, службы удаленного рабочего стола.

Также были определены общие требования, которые могут предъявляться для систем обмена данным: эффективность и скорость, надежность, безопасность, доступность, масштабируемость, взаимодействие, совместимость, контроль версий, управление данными, мобильный доступ, автоматизация, резервное копирование и восстановление данных, обучение и поддержка, соответствие и нормативные требования, экономическая эффективность, конфиденциальность, отзывы пользователей и улучшения в работе системы.

Далее выполнен Анализ существующих проблем и вызовов при обмене данными в вычислительной сети организации.

В заключении дана характеристика исследуемой организации. Показаны ее услуги, описана организационная структура, которая является функиональной.

Из обзора финансовых показателей видно, что они нестабильны, хоть и имеют положительную линию тренда. Поэтому следует предложить пути повышения эффективности деятельности организации АО «ФИНТЕХ».

Архитектура вычислительной сети имеет вид трехзвенной. Трехзвенная сетевая архитектура для построения LAN (локальной сети) включает в себя три основных компонента или уровня, которые обеспечивают функциональность и управление сетью.

Программная среда функционирует в пространстве операционной системы Astra Linux 1.7.

Что касается обмена данными в вычислительной сети исследуемой организации, то он выполняется с помощью разных технологий и отсутствие четкой стандартизации приводит к тому, что сложно обеспечить требуемый уровень безопасности.

Поэтому следует предложить способ обмена файлами, который будет обеспечивать, с одной стороны, удобство, а, с другой стороны, безопасность.

Во втором разделе осуществлен выбор технологий, а также осуществлено проектирование ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ».

Исходя из анализа возможных альтернативных типов приложений, выбор пал на веб-приложение, потому что оно наиболее универсально при минимизации затрат на его разработку. Поскольку шифрование/дешифрование будет осуществляться на сервере, то для клиентского аппаратного обеспечения не будет предъявляться серьезных требований к производительности.

В качестве языка программирования клиентской части выбран Javascript, так как он является стандартом для веб-приложений на сегодняшний день, он мощный и гибкий.

В качестве языка программирования серверной части выбран PHP, так как он производительный, простой в изучении и использовании, обеспечивает должный уровень безопасности и является стандартом для разработки веб-ориентированных систем.

Выбор СУБД был между MySQL и PostgreSQL, так как они уже функционируют на сервере предприятия. Выбрана MySQL, потому что она более проста в использовании, имеет высокую производительность на чтение, надежную поддержку соответствия ACID, поддерживает механизмы репликации.

Также выбран фреймворк Bootstrap, потому что он широко распространен, очень прост и понятен в изучении и использовании. Имеются множество

форумов и ресурсов с шаблонами, которые можно применять для быстрой и эффективной разработки.

В качестве алгоритма шифрования выбран Blowfish, так как он имеет высокую скорость, безопасность, различные размеры ключей в широком диапазоне, что позволит при необходимости настраивать скорость его работы при масштабировании системы.

Далее сформированы функциональные и нефункциональные требования к проектируемому программному обеспечению.

Для проектирования системы были применены возможности, которые предоставляет визуальный язык моделирования UML.

Разработаны диаграммы:

- вариантов использования;
- последовательностей;
- компонентов.

Также составлены логическая и физическая модели базы данных, которая состоит из одной таблицы «Документы». Логическая модель содержит имена таблицы и полей на кириллице, а в физической модели осуществлен переход к названиям на латинице, а также добавлены типы данных.

В третьем разделе выполнена программная реализация ПО для обмена данными в вычислительной сети организации АО «ФИНТЕХ».

Описана структурная схема веб-приложения.

Далее описан пример работы системы. Разработанное веб-приложение может быть использовано в компании АО «ИНТЕХ» для организации обмена данными в формате файлов/документов посредством их передачи через вычислительную сеть.

Модульность архитектуры позволит в будущем улучшать функционал системы по мере необходимости, и это не будет сопряжено со значительными финансовыми и трудовыми расходами на содержание команды разработчиков.

Расчет экономической эффективности показал, что:

- себестоимость проекта составляет 115 365,08 рублей;

- экономия трудозатрат составляет 375 часов в год;
- экономия финансовых затрат на обмен данными составляет 150 000 рублей в год;
- срок окупаемости проекта составит 0,77 года.

Это значение является приемлемым для АО «ИНТЕХ». Поэтому можно говорить об эффективности проекта разработки и внедрения ПО для обмена данными в вычислительной сети организации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Буткевич Д. Spring в действии. - ДМК Пресс, 2019. - 576 с.
2. ИВВ Тайны Шифрования: Взгляд в Мир Криптографии. Криптография: Магия Шифров. - Ridero, 2023. - 28 с.
3. Кит М. Криптография. Как защитить свои данные в цифровом пространстве. - Бомбора, 2022. - 368 с.
4. Копет Д. SQL. Полное руководство. - ДМК Пресс, 2020. - 384 с.
5. Кузин А.В. Компьютерные сети. Учебное пособие. - Форум, 2020. - 278 с.
6. Левицкий Н. Удаленный сервер своими руками. От азов создания до практической работы. - Наука и техника, 2021. - 400 с.
7. Макколл А. «HTML и CSS. Дизайн и верстка веб-сайтов.» Москва: Вильямс, 2019. – 354 с.
8. Макфарланд Д. JavaScript и jQuery. Интерактивная веб-разработка. - Вильямс, 2019. - 640 с.
9. Машнин Т. Bootstrap: Быстрое создание современных сайтов. - Ridero, 2020. - 232 с.
10. Никсон Робин Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 6-е изд. - Питер, 2023. - 832 с.
11. Никулина И. Настольная книга веб-дизайнера. Практический курс по веб-дизайну и проектированию сайтов. - Ridero, 2023. - 302 с.
12. Олифер В. Компьютерные сети. Принципы, технологии, протоколы: Юбилейное издание. - Питер, 2023. - 496 с.
13. Омассон Ж. О криптографии всерьез. - ДМК Пресс, 2022. - 328 с.
14. Резиг Джон. «Pro JavaScript Techniques» Birmingham: Apress, 2019.
15. Северанс Ч. Как работают компьютерные сети и интернет. - ДМК Пресс, 2021. - 116 с.
16. Скляр Д. PHP и Laravel. Создание сайтов. - Диалектика, 2019. - 528 с.
17. Таненбаум Э. Компьютерные сети. - Вильямс, 2019. - 960 с.
18. Терлецкий А. JavaScript. Подробное руководство. - Питер, 2019. - 832 с.

19. Тидвелл Д. Разработка интерфейсов. Паттерны проектирования. - Питер, 2022. - 560 с.
20. Тургенев А. Проектирование баз данных. - Питер, 2019. - 576 с.
21. Урбанович П. Компьютерные сети. Учебное пособие. - Инфра-Инженерия, 2022. - 385 с.
22. Уэзеролл Д. Компьютерные сети. 6-е изд. - Питер, 2023. - 992 с.
23. Шилдт Г. Java 10. Основы. - ДМК Пресс, 2019. - 672 с.
24. Веб-сайт компании АО «ФИНТЕХ» [Электронный ресурс]. - Режим доступа: https://fintech.ru/
25. Данные контрагента АО «ФИНТЕХ» [Электронный ресурс]. - Режим доступа: https://sbis.ru/contragents/7733022819/770601001
26. Blowfish encryption algorithm [Электронный ресурс]. - Режим доступа: https://www.schneier.com/academic/blowfish/download/

ПРИЛОЖЕНИЕ 1 – ПРОГРАММНЫЙ КОД

Код шаблона главной страницы

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>ПО для обмена данными в вычислительной сети организации
АО «ФИНТЕХ»</title>
    <meta name="description" content="ПО для обмена данными в
вычислительной сети организации АО «ФИНТЕХ»">
    <link href="css/bootstrap.min.css" rel="stylesheet">
  <link
href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>

    <div class="container-fluid">
  <div class="row">
    <div class="col-md-12">
      <h3 class="text-center mt-3 mb-3">ПО для обмена данными в
вычислительной сети организации АО «ФИНТЕХ»</h3>
    </div>
  </div>
  <div class="row pt-4 mb-4">
    <div class="col-md-3">
      <a class="btn btn-success mt-2" href="<?php echo F_PREFIX; ?
>" role="button"><i class="fa fa-home"></i> Обновить страницу</a>
    </div>
    <div class="col-md-9">
      <p>Алгоритм для отправителя: 1) Выберите параметры хранения.
2) Выберите файл для отправки. 3) Скопируйте ссылку на файл. 4)
Перешлите ссылку получателю через личное сообщение в корпоративном
мессенджере.</p>
      <p>Алгоритм для получателя: 1) Перейтиде по полученной
ссылке на файл. 2) Нажмите на кнопку "Расшифровать и получить
ссылку". 3) Скачайте файл по ссылке.</p>
    </div>
  </div>

<?php

// Парсинг строки URL
$address = preg_replace("@^".preg_quote(F_PREFIX)."@isU","",
$_SERVER['REQUEST_URI']);
```

```
$parts = explode('/', $address);
$sys_name = '';
if(isset($parts[2]) && trim($parts[2])!="") // Присутствует
параметр
{
  $sys_name = trim($parts[2]);

  // Выяснение наличия файла в системе
  $connection = mysqli_connect(F_HOST, F_USER, F_PASSWORD, F_DB);
    $query_string = "SELECT * from files WHERE sys_name like
'$sys_name'";
    $query_result = mysqli_query($connection, $query_string);
    if(!$query_result)
      echo '<div class="row"><div class="col-md-12"><div
role="alert" class="alert alert-danger">Ошибка подключени к
БД</div></div></div>';
    $query_data = mysqli_fetch_assoc($query_result);
  mysqli_close($connection);
  $id_document = $query_data['id_document'];
  $source_name = $query_data['source_name'];
  $doc_ext = $query_data['doc_ext'];
  $date_uploaded = $query_data['date_uploaded'];
  $time_holding = $query_data['time_holding'];
  $cnt_downloaded = $query_data['cnt_downloaded'];
  if($id_document > 0) // Файл присутствует
  {
    if($time_holding==1 && $cnt_downloaded>0) // Файл не найден
или удален
    {
      echo "<div class='row'><div class='col-md-12'><div
class='alert alert-danger' role='alert'>Файл не найден или
удален</div></div></div>";
    }
    else
    {
      echo '<div class="row"><div class="col-md-12"><div
class="alert alert-success" role="alert">Файл найден, и можено его
скачать!</div></div></div>';
      ?>
      <div class="row">
        <div class="col-md-12">
          <div class="form-group">
            <div class="input-group mb-4">
              <div class="input-group-prepend">
                <span class="input-group-text">Введите ключ
шифрования   </span>
              </div>
              <input id="key_decode" class="form-control"
type="text" aria-label="Введите ключ шифрования">
              <div class="input-group-append">
                   <button class="btn btn-primary"
type="button" link="" style="width:340px;" id="decode_btn"><i
```

```
class="fa fa-gear"></i> Расшифровать файл и получить
ссылку</button>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div id="dwnld_div" class="invisible row">
        <div class="col-md-12">
          <div class="form-group">
            <div class="input-group mb-4">
              <div class="input-group-prepend">
                <span class="input-group-text">Кнопка для
скачивания файла      </span>
              </div>
              <div class="input-group-append">
                   <button type="button" class="btn
btn-success" id="button_download" style="width:200px;"><i
class="fa fa-download"></i> Скачать файл</button>
              </div>
            </div>
          </div>
        </div>
      </div>
      <?php
    }
  }
  else // Файл не найден
  {
    echo '<div class="row"><div class="col-md-12"><div
class="alert alert-danger" role="alert">Файл был удален, либо его
не существует</div></div></div>';
  }
}
else
{
?>
  <div class="row">
    <div class="col-md-12">
      <div class="form-group">
        <div class="input-group mb-4">
          <div class="input-group-prepend">
            <span class="input-group-text">Сгенерированный ключ
шифрования   </span>
          </div>
          <input type="text" class="form-control" aria-
label="Сгенерированный ключ шифрования" id="key_encode">
        </div>
      </div>
    </div>
  </div>
  <div class="row">
```

```
    <div class="col-md-12">
      <div class="form-group">
        <div class="input-group mb-4">
          <div class="input-group-prepend">
            <span class="input-group-text">Укажите параметры
хранения      </span>
          </div>
          <label class="radio-inline" for="time_holding-0">
            <input type="radio" name="time_holding"
id="time_holding-0" value="1" checked="checked">
            Одно скачивание   
          </label>
          <label class="radio-inline" for="time_holding-1">
            <input type="radio" name="time_holding"
id="time_holding-1" value="2">
            7 дней   
          </label>
          <label class="radio-inline" for="time_holding-2">
            <input type="radio" name="time_holding"
id="time_holding-2" value="3">
            14 дней   
          </label>
          <label class="radio-inline" for="time_holding-3">
            <input type="radio" name="time_holding"
id="time_holding-3" value="4">
            30 дней   
          </label>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-12">
      <div class="form-group">
        <div class="input-group mb-4">
          <div class="input-group-prepend">
            <span class="input-group-text">Выберите отправляемый
файл   </span>
          </div>
          <input type="text" class="form-control" aria-
label="Выберите файл на своем устройстве" id="file_input">
          <div class="input-group-append">
               <label style="width:290px;"
class="btn btn-primary" ><i class="fa fa-upload"></i> Выбрать файл
и зашифровать<input type="file" hidden
id="choose_encode_btn"></label>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
```

```
    <div class="col-md-12">
      <div id="my_progress">
        <div class="progress-bar"></div>
        <div class="status">0%</div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-12">
      <p class="pt-4"><b>Получите и скопируйте ссылку на
файл</b></p>
      <div class="form-group">
        <div class="input-group mb-4">
          <div class="input-group-prepend">
            <span class="input-group-text">Ссылка на зашифрованный
файл      </span>
          </div>
          <input type="text" class="form-control" aria-
label="Ссылка на зашифрованный файл" id="file_output">
          <div class="input-group-append">
               <button type="button" class="btn
btn-primary mt-3" id="copy_link_btn"
style="width:290px;">Скопировать ссылку на файл</button>
          </div>
        </div>
      </div>
    </div>
  </div>
<?php
}
?>
```

Подсистема генерации имени файла, его кодирования, а также записи в БД

```
// Имя файла
while(true)
{
     $filename = uniqid('', true);//.'.'.$inf['doc_ext'];
     $path_file = $uploaddir.$filename;
     if (!file_exists($path_file))
          break;
}

// Шифрование файла посредством BlowFish
$source_symbols = file_get_contents($name);
$alg_key = trim($_POST['key']);
$alg_mode = Blowfish::BLOWFISH_MODE_EBC;
$alg_padding = Blowfish::BLOWFISH_PADDING_RFC;
$alg_iv = null;
```

```
$blow_object = new Blowfish($alg_key, $alg_mode, $alg_padding,
$alg_iv);
$encr_symbols = $blow_object->encrypt($source_symbols, $alg_key,
$alg_mode, $alg_padding, $alg_iv);
file_put_contents($path_file, $encr_symbols);

// Запись в БД
$connection = mysqli_connect(F_HOST, F_USER, F_PASSWORD, F_DB);
$query_string = «INSERT INTO documents(source_name, sys_name,
doc_ext, time_holding) values('$source_name', '$filename',
'$doc_ext', $time_holding);»;
$query_result = mysqli_query($connection, $query_string);
if(!$query_result)
{
     echo «Ошибка вставки в БД»;
     return false;
}
mysqli_close($connection);

echo F_PREFIX.$filename;
```

Подсистема декодирования файла

```
// Декодирование файла
$source_symbols = file_get_contents($path_to_file);
$alg_mode = Blowfish::BLOWFISH_MODE_EBC;
$alg_padding = Blowfish::BLOWFISH_PADDING_RFC;
$alg_iv = null;
$blow_object = new Blowfish($alg_key, $alg_mode, $alg_padding,
$alg_iv);
$decoded_symbols = $blow_object->decrypt($source_symbols,
$alg_key, $alg_mode, $alg_padding, $alg_iv);

// Копирование в папку для временного хранения
while(true)
{
     $tmppath = uniqid('', true);
     if (!is_dir($tmppath))
     break;
}
mkdir($dir_base.'/tmp/'.$tmppath);
$decoded_symbols_file_1 = $tmppath.»/».iconv(«UTF-8»,
«Windows-1251», $source_name).».».$doc_ext;
file_put_contents($dir_base.'/tmp/'.$decoded_symbols_file_1,
$decoded_symbols);
$decoded_symbols_file_2 = $tmppath.»/».$source_name.».».$doc_ext;
echo $decoded_symbols_file_2;
```

ПОСЛЕДНИЙ ЛИСТ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«08» Ноября 2023 г.



/ ***Хомутов С. А.***

(подпись) (Ф.И.О.)