|  |  |
| --- | --- |
| **­­** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ДОМАШНЕМУ ЗАДАНИЮ***

***НА ТЕМУ:***

***\_\_\_\_Групповая отправка файла в мессенджере\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_ИУ5-55Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Расулов А.Н.\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

# АННОТАЦИ**Я**

Расчётно-пояснительная записка содержит 30 страниц. С приложениями объем составляет 45 страниц. Работа включает в себя 8 диаграмм и 21 изображение системы. В процессе выполнения было использовано 13 источников.

Объектом разработки является система групповой отправки файлов в мессенджере. Данная система позволяет автоматизировать процессы учета получателей, формирования и обработки отправок файлов, а также улучшить взаимодействие между отправителями и получателями.

Цель работы заключается в разработке эффективной системы групповой отправки файлов, которая включает функциональные возможности для добавления и редактирования данных о получателях, формирования отправок, их подтверждения или отклонения, а также отслеживания статуса отправок.

В ходе работы была спроектирована архитектура системы, разработан интерфейс для взаимодействия отправителей и получателей с системой, развернут веб-сервер и приложения, предоставляющие доступ к функционалу системы через нативное и прогрессивное веб-приложения.

Пояснительная записка содержит 2 приложения.

СОДЕРЖ**АНИ**Е

[АННОТАЦИЯ 1](#_Toc185863538)

[СОДЕРЖАНИЕ 2](#_Toc185863539)

[ВВЕДЕНИЕ 3](#_Toc185863540)

[1 ПРЕДМЕТНАЯ ОБЛАСТЬ 5](#_Toc185863541)

[2 СРАВНЕНИЕ АНАЛОГОВ 9](#_Toc185863542)

[3 АРХИТЕКТУРА 13](#_Toc185863543)

[4 АЛГОРИТМЫ 17](#_Toc185863544)

[5 ОПИСАНИЕ ИНТЕРФЕЙСА 19](#_Toc185863545)

[ЗАКЛЮЧЕНИЕ 28](#_Toc185863546)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc185863547)

[ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ 30](#_Toc185863548)

[1. Введение 31](#_Toc185863549)

[2. Назначение разработки 31](#_Toc185863550)

[3. Стадии и этапы разработки 31](#_Toc185863551)

[4. Требования к функциональным характеристикам 32](#_Toc185863552)

[5. Требования к составу и параметрам технических средств 36](#_Toc185863553)

[6. Требования к информационной и программной совместимости 37](#_Toc185863554)

[ПРИЛОЖЕНИЕ Б СПИСОК HTTP 38](#_Toc185863555)

ВВЕДЕНИЕ

В современном мессенджере важное значение имеет скорость и удобство отправки файлов нескольким пользователям. В условиях высокой конкуренции и развития функционала, внедрение системы персонализированных комментариев к файлам становится неотъемлемой частью эффективного общения. Одной из таких функций является групповая отправка файлов с индивидуальными сообщениями для каждого получателя. Этот процесс должен быть максимально прозрачным, удобным для отправителя и понятным для получателей.

Целью данной работы является разработка системы, предоставляющей интерфейс для учета отправок файлов с комментариями, и включающей в себя веб-сервис, веб-приложение и десктопное приложение Tauri.

Эта система предназначена для пользователей мессенджера, желающих отправить один файл нескольким контактам с индивидуальными комментариями для каждого. Пользователи могут выбирать получателей из списка контактов и добавлять к файлу персональные сообщения. Система обеспечивает процесс создания, отслеживания и управления отправками, предоставляя возможность просматривать историю отправленных файлов и комментарии к ним. Получатели могут просматривать, отклонять или подтверждать отправки файлов, обеспечивая безопасность и соответствие правилам платформы.

Система автоматизирует процесс отправки файлов нескольким получателям, обеспечивая удобство для отправителя и персонализированное взаимодействие с каждым получателем. Реализован доступ к подробной информации об отправленных файлах и комментариях, что позволяет вести историю переписки и управлять отправленными файлами (например, удалять). Доступ получателя к информации о файлах и комментариях позволяет контролировать контент и обеспечивать безопасность платформы.

Нефункциональные требования к разрабатываемой системе:

1. Должна поддерживаться кроссплатформенность.
2. Интерфейс системы и текст ошибок должны быть на русском языке.

В ходе работы необходимо выполнить следующие задачи:

* 1. Создать MVP и базовый дизайн на основе vk.com
  2. Создать базу данных для хранения информации о получателях, отправках файлов и отправителях
  3. Создать веб-сервис на языке Python с использованием Django Rest Framework.
  4. Реализовать авторизацию и хранение сессий в Redis
  5. Разработать базовый SPA на React для гостя
  6. Внедрить адаптивность, менеджер состояний Redux Toolkit, PWА, разработать Tauri приложение
  7. Завершить разработку интерфейса отправителя на React, использовать для обращений к методам веб-сервиса Axios
  8. Реализовать интерфейс получателя в React
  9. Разработать десктопное приложение Tauri
  10. Развернуть приложение при помощи GitHub Pages
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм
  12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код работы
  13. Привести анализ сервисов-аналогов.

1. ПРЕДМЕТНАЯ ОБЛАСТЬ

Существует множество проблем в области эффективного обмена информацией внутри команд и рабочих групп. Далеко не каждый мессенджер предусматривает удобный и контролируемый механизм для групповой отправки файлов с персонализированными комментариями. Зачастую возникают ситуации, когда необходимо отправить один и тот же файл нескольким коллегам, добавив к каждому файлу индивидуальное пояснение или инструкцию. Необходима система, которая автоматизирует этот процесс и обеспечит контроль контента, что особенно важно в корпоративной среде. Подобный функционал частично реализован в некоторых мессенджерах, например, в Slack [1], который позволяет упоминать отдельных пользователей в сообщениях с файлами. Однако, отсутствует возможность добавления расширенных комментариев и централизованной модерации. В KiloGram мы стремимся предоставить более полный и удобный инструмент для групповых рассылок файлов.

В системе KiloGram пользователи могут прикреплять файлы к сообщениям, адресованным группе пользователей. В случае необходимости добавления индивидуального комментария к файлу для каждого получателя, предусмотрен специальный интерфейс. Таким образом, все получатели увидят один и тот же файл, но с персонализированным сообщением. Это позволяет избежать путаницы и улучшить коммуникацию внутри команды. Пользователи KiloGram, смогут отслеживать историю отправленных файлов и комментариев. При разработке системы учитывались лучшие практики в области юзабилити и безопасности, опираясь на опыт существующих мессенджеров и рекомендации экспертов [2, 3].

Пользователи KiloGram имеют возможность создавать черновики отправок, добавляя комментарии для каждого получателя. Когда отправитель будет готов отправить отправку, он формирует ее, после чего она направляется на модерацию. После этого отправку уже нельзя редактировать. Можно также просмотреть историю своих заявок. Получатели могут принимать заявки или же отклонять их, например, в случае нарушения политики безопасности или распространения конфиденциальной информации. Благодаря гибкой системе модерации, KiloGram обеспечивает контроль над контентом и помогает предотвратить утечку данных. В случае необходимости, получатели также могут редактировать комментарии к файлам перед их отправкой получателям. Таким образом, поддерживается баланс между удобством использования для отправителей и контролем контента со стороны получателей.

Функции пользователей с различными ролями описаны на диаграмме прецедентов (рисунок 1).

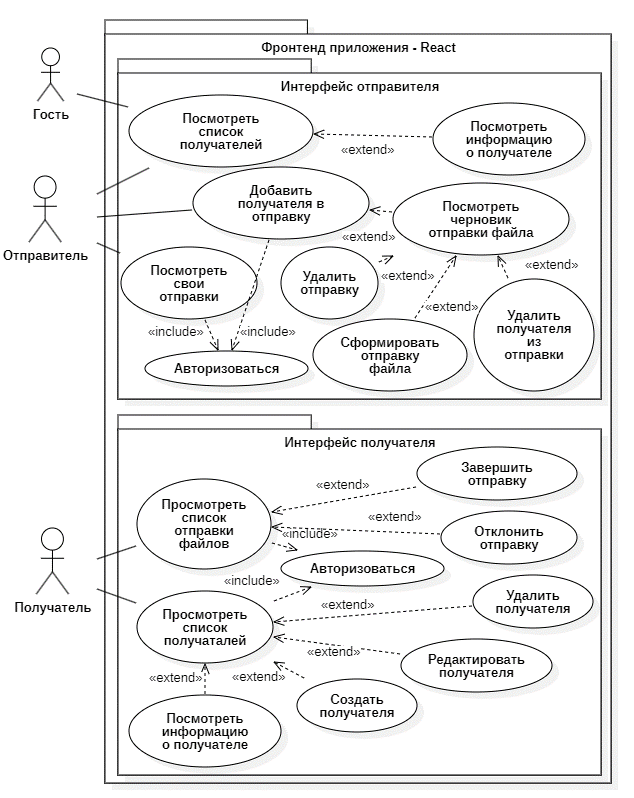


Рисунок 1 – Диаграмма прецедентов

Гостям доступен просмотр списка получателей. Гости, прошедшие этап регистрации, являются отправителями. Отправители могут добавлять получателей в отправку файла, формировать её, а также просматривать список своих отправок.

Отправитель выбирает получателей, которым намеревается отправить файл, затем на основе выбранного списка формирует отправку. После чего получатель может просмотреть её, а также согласовать или отклонить. Возможные состояния статуса отправки отображены на диаграмме состояний (рисунок 2)

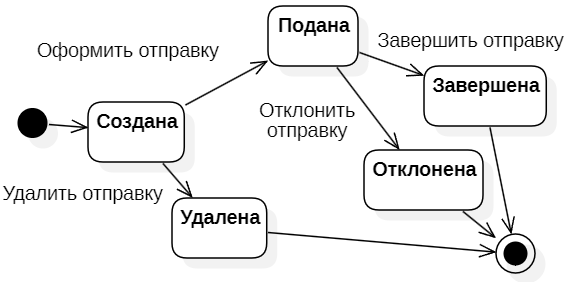


Рисунок 2 – Диаграмма состояний заявок

Заявки обрабатываются получателями. В результате обработки отправки ее либо одобряют, либо отклоняют. Получателю также доступны операции для работы со списком получателей: просмотр всех получателей, редактирование, создание и удаление получателей, а также просмотр списка всех получателей в табличном виде.

Процесс оформления отправки файла отражен на диаграмме деятельности (рисунок 3).

В начале взаимодействия с системой оформления отправок файла отправитель запрашивает список доступных для отправки получателей. При необходимости отправитель может добавить получателей в текущую черновую отправку. После чего может продолжить выбор получателей из каталога или перейти к оформлению заявки. В случае, если отправитель выбрал оформление отправки файла, ему необходимо выбрать файл, который будет отправлен всем получателям отправки. По нажатии кнопки «Оформить» отправка файла формируется и становится доступна в интерфейсе получателя.

Получатель имеет возможность просматривать отправки отправителей, а также принимать решение об одобрении или отклонении отправки.

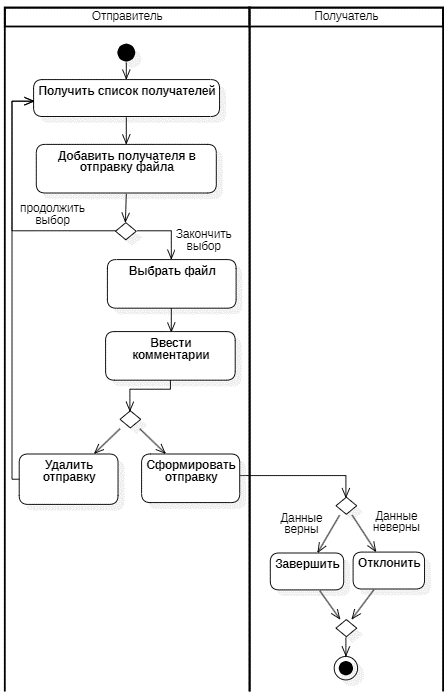


Рисунок 3 – Диаграмма деятельности

1. СРАВНЕНИЕ АНАЛОГОВ

Telegram – это мессенджер, известный своей высокой скоростью работы и надёжной защитой данных. Приложение поддерживает отправку файлов различных типов с максимальным размером до двух гигабайт, что делает его удобным инструментом для работы с крупными вложениями [4]. Telegram предоставляет возможность добавлять общие комментарии к файлам, однако функции индивидуальных комментариев для каждого получателя не предусмотрено.

На рисунке 4 представлен интерфейс Telegram с открытым каналом

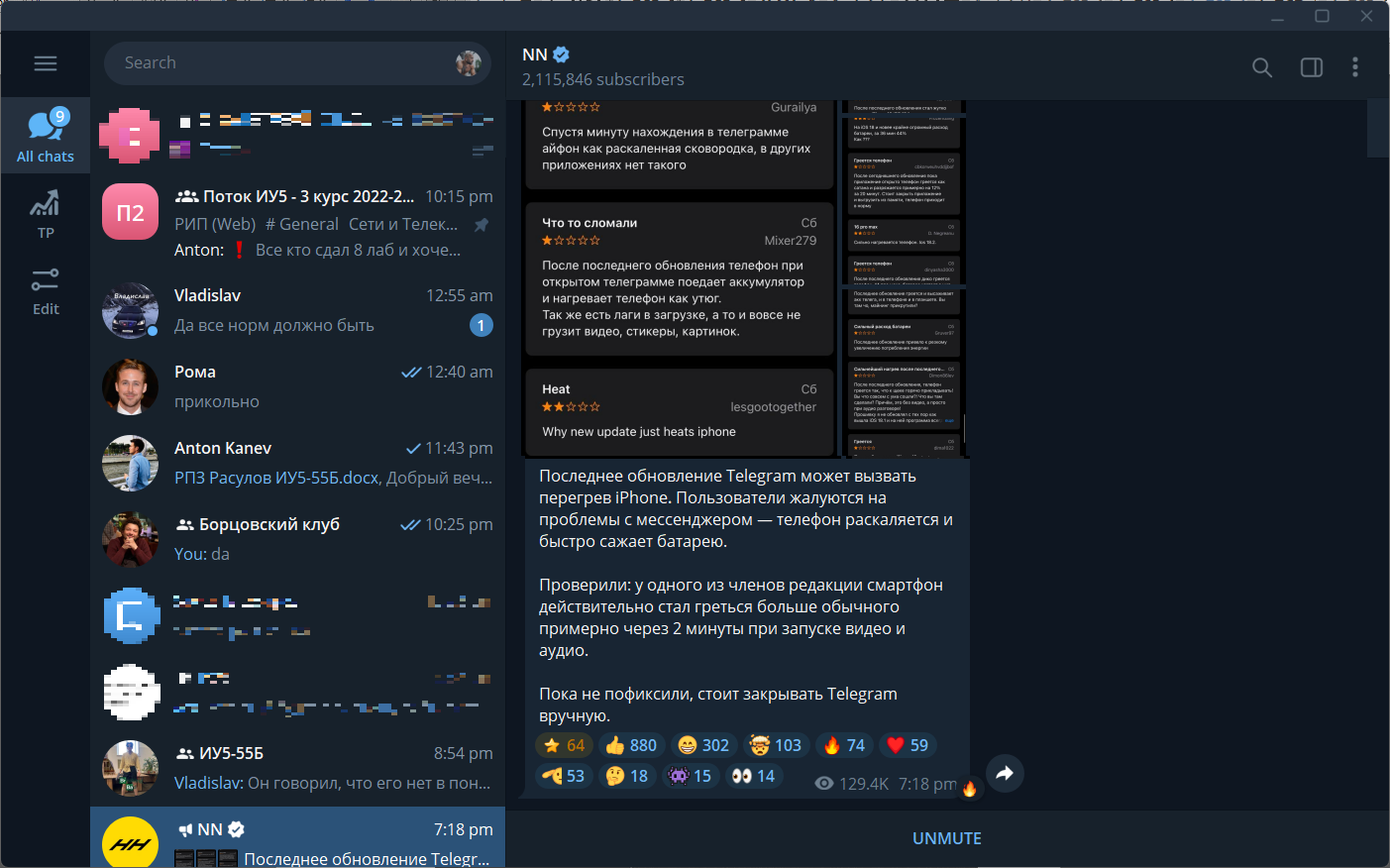


Рисунок 4 – Окно Telegram со списком получателей и открытым каналом

WhatsApp – мессенджер, широко используемый для личного и делового общения. Он позволяет отправлять файлы различных форматов, таких как документы, изображения и видео, но размер файлов ограничен 2 ГБ для документов и 100 МБ для видео [5]. Приложение поддерживает функцию списков рассылки, что позволяет отправлять сообщения сразу нескольким получателям, однако индивидуальные комментарии к файлам добавить невозможно.

На рисунке 5 представлен интерфейс WhatsApp с открытым чатом

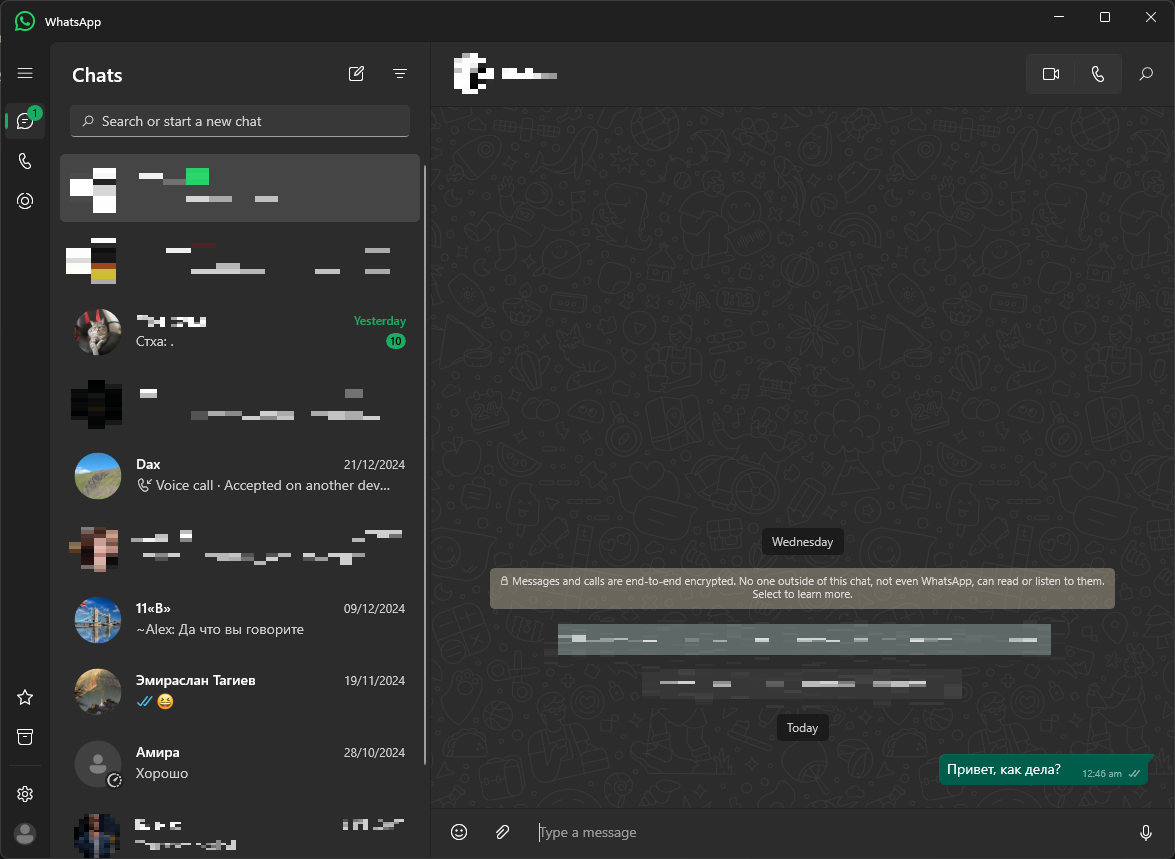


Рисунок 5 – Окно WhatsApp со списком получателей и открытым чатом.

Discord – мессенджер, изначально разработанный для геймеров, но теперь активно используемый для различных сообществ. Приложение позволяет отправлять файлы в текстовых каналах и личных сообщениях, но в бесплатной версии ограничение на размер файла составляет 25 мегабайт (до 100 мегабайт в версии Nitro) [6]. Discord также поддерживает добавление общих комментариев к файлам, но не предоставляет возможности добавления индивидуальных комментариев для каждого получателя.

На рисунке 6 представлен интерфейс Discord с отправленным файлом

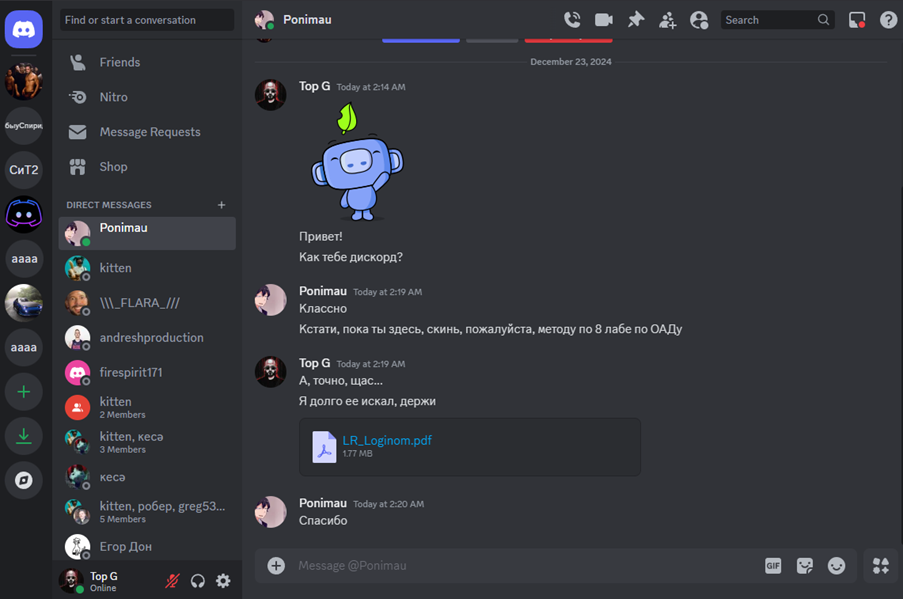


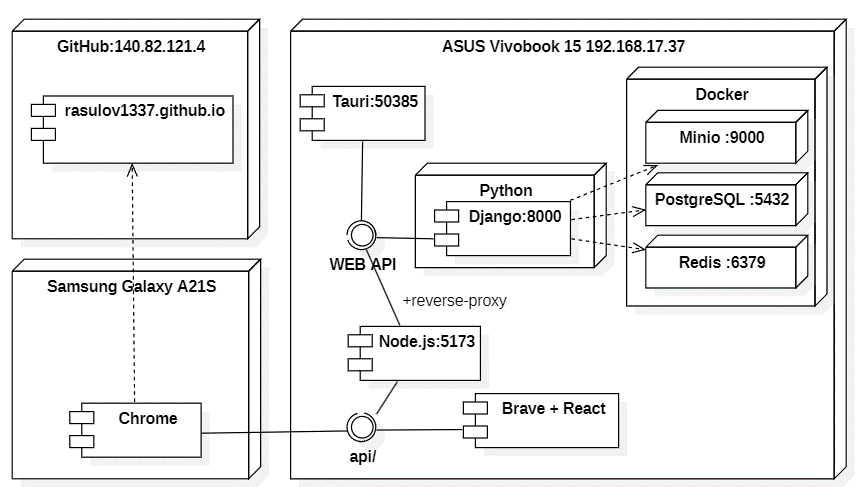
Рисунок 6 – Окно Discord со списком серверов и получателей (друзей) и открытым чатом.

Для наглядного представления различий между рассмотренными мессенджерами был составлен сравнительный анализ в виде таблицы. В таблице представлены ключевые характеристики каждого из приложений, такие как поддерживаемые типы файлов, ограничения на их размер, возможность добавления комментариев и другие параметры. Это позволит чётко увидеть преимущества и недостатки каждого мессенджера и сделать выводы о наиболее подходящем инструменте для массовой отправки файлов.

Таблица 1 – сравнительная таблица сервисов Telegram, WhatsApp и Discord

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Критерий** | **Telegram** | **WhatsApp** | **Discord** |
| 1 | Целевая аудитория | Широкая, включая все возрастные группы | Широкая, включая все возрастные группы | Геймеры, сообщества, стримеры, создатели контента |
| 2 | Ограничение на размер | 2 ГБ | 2 ГБ (видео до 16 МБ) | 25 МБ (по подписке Nitro Classic – 50 МБ, Nitro - 100 МБ) |
| 3 | Возможность добавлять комментарии | Присутствует, комментарии только общие | Присутствует, комментарии только общие | Присутствует, комментарии только общие |
| 4 | Возможность отмены отправки | Присутствует на ограниченное время | Присутствует (с пометкой «удалено») на ограниченное время | Присутствует |
| 5 | Модерация | Базовые функции | Базовые функции | Расширенные функции с ролями |
| 6 | Поддерживаемые форматы файлов | Любые форматы файлов | Изображения, видео, аудио, документы | Любые форматы файлов |
| 7 | Кроссплатформенность | Windows, macOS, Linux, Android, iOS, Web | Windows, macOS, Android, iOS, Web (ограниченная функциональность) | Windows, macOS, Linux, Android, iOS, Web |
| 8 | Максимальное число участников чата, число человек | 200 000 | 1024 | 250 000 |

1. АРХИТЕКТУРА

Архитектура системы отображена на диаграмме развертывания (рисунок 7). Рисунок 7 – Диаграмма развертывания

Веб-сервис, реализованный на языке Python [7] с использованием фреймворка Django Rest Framework [8] связан с серверами Redis [9], Minio [10], и СУБД PostgreSQL [11]. В Redis хранятся идентификаторы сессий пользователей, добавление которых происходит при входе пользователя в систему.

Язык программирования Python был выбран благодаря его хорошей производительности и простоте синтаксиса. Использование фреймворков Django и Django Rest Framework позволяет создавать надежные и эффективные веб-сервисы. Веб-сервис на Django является общим для веб-сервера и десктопного приложения Tauri [12].

Данные хранятся в СУБД PostgreSQL. Она была выбрана, является стандартом современной индустрии разработки.

Структура данных отражена на ER диаграмме (рисунок 8). Модель получателей представляет собой набор полей, отражающих параметры получателя. Данные о получателе хранятся в таблице Получатель. Для хранения в одной отправке нескольких договоров используется промежуточная таблица связи М-М. Таблица «Процесс отправки файла» представляет собой список отправок файла. Данные о пользователях системы (отправителях и получателях) хранятся в таблице «Пользователь».

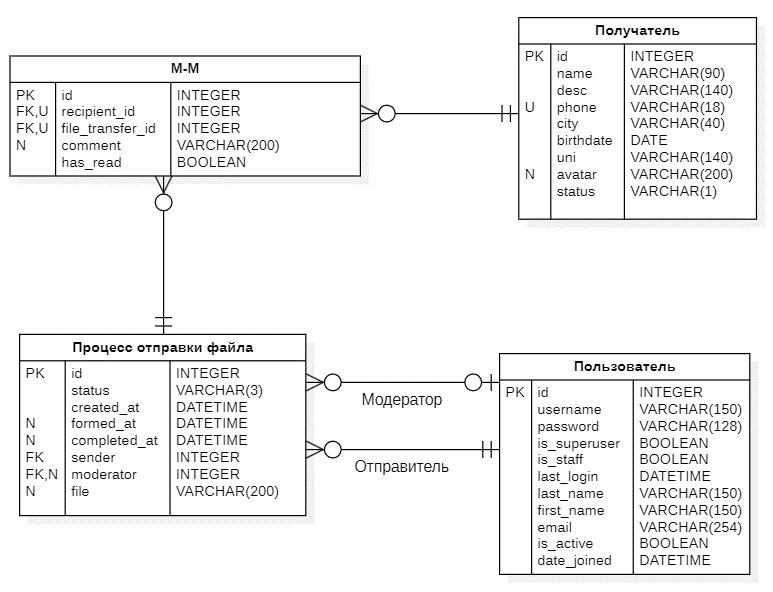


Рисунок 8 – ER диаграмма.

Веб-сервер реализован с использованием языка TypeScript и фреймворка React [13]. Устройство бэкенд-приложения разработанной системы приведено на диаграмме классов бэкенда (рисунок 9). Пользователи взаимодействуют с доменами. Домены связаны с моделями. Модели имеют связи с таблицами в базе данных.

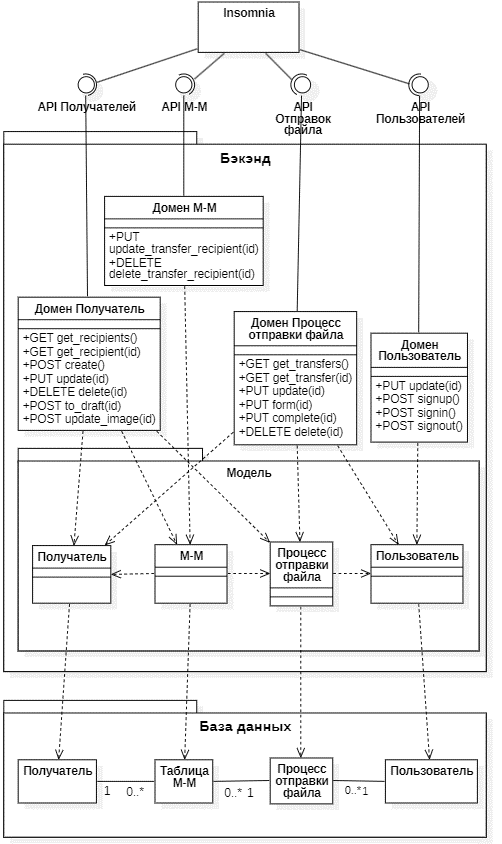


Рисунок 9 – Диаграмма классов бэкенда

Связь фронтенда и бэкенда отражена на диаграмме классов фронтенда (рисунок 10). Каждая страница связана с API, которое используется для взаимодействия с данными на соответствующей странице.

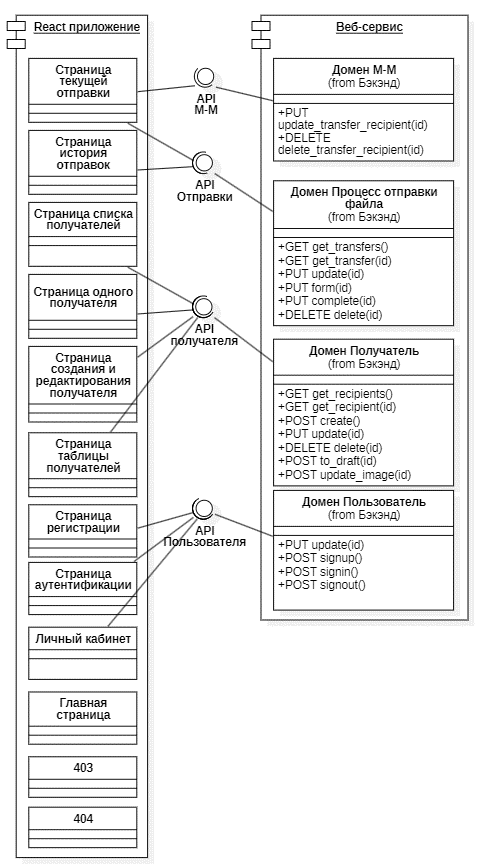


Рисунок 10 – Диаграмма классов фронтенда

1. АЛГОРИТМЫ

Алгоритм работы разработанной системы отображен на диаграмме последовательности (рисунок 11). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: пользователи, отправки, получатели, м-м. Методы следуют правилам REST API.

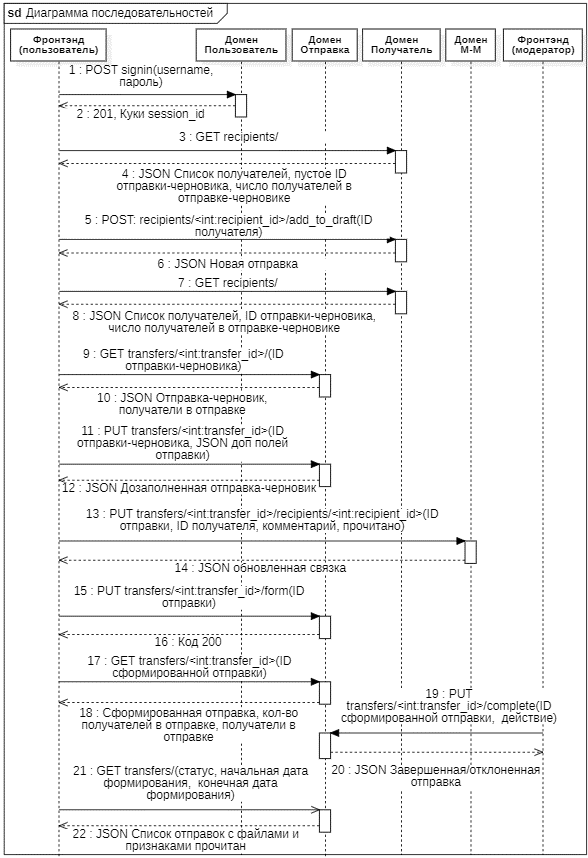


Рисунок 11 – Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. Через графический интерфейс гость вводит логин и пароль для доступа к системе. Если учетная запись с указанными данными существует в базе, сервис возвращает информацию о пользователе и идентификатор сессии в ответе. Если же учетной записи с введённым логином не существует или пароль неверен, пользователь получает сообщение об ошибке. В таком случае пользователю предлагается пройти регистрацию или повторно ввести корректные данные. На этом этапе система также определяет роль пользователя — отправитель или получатель.

После успешной аутентификации отправитель через графический интерфейс запрашивает у веб-сервиса список доступных получателей, который возвращается в формате JSON. Отправитель может выбрать нужного получателя и добавить его в отправку файла. Для этого он нажимает кнопку «Добавить» в графическом интерфейсе. Процесс выбора получателей и их добавления в черновую отправку может повторяться несколько раз.

Когда отправитель завершает выбор получателей и указывает необходимые данные для отправки файла (файл для отправки получателям), он нажимает кнопку «Сформировать отправку». Приложение отправляет на веб-сервис запрос на формирование отправки. После этого отправитель может отслеживать статус своих отправок на специальной странице интерфейса, где отображаются все его созданные отправки с указанием текущего статуса.

1. ОПИСАНИЕ ИНТЕРФЕЙСА

Главное меню приложения включает пункты, которые доступны в зависимости от роли пользователя (рисунки 12, 13, 14).

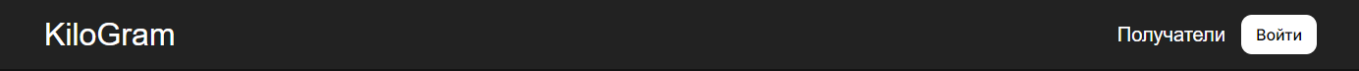


Рисунок 12 – Меню приложения (для гостя)

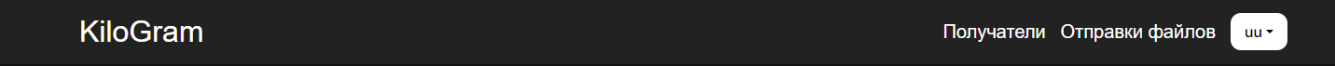


Рисунок 13 – Меню приложения (для отправителя)

На странице с формой регистрации (рисунок 14) гость может создать новый аккаунт.

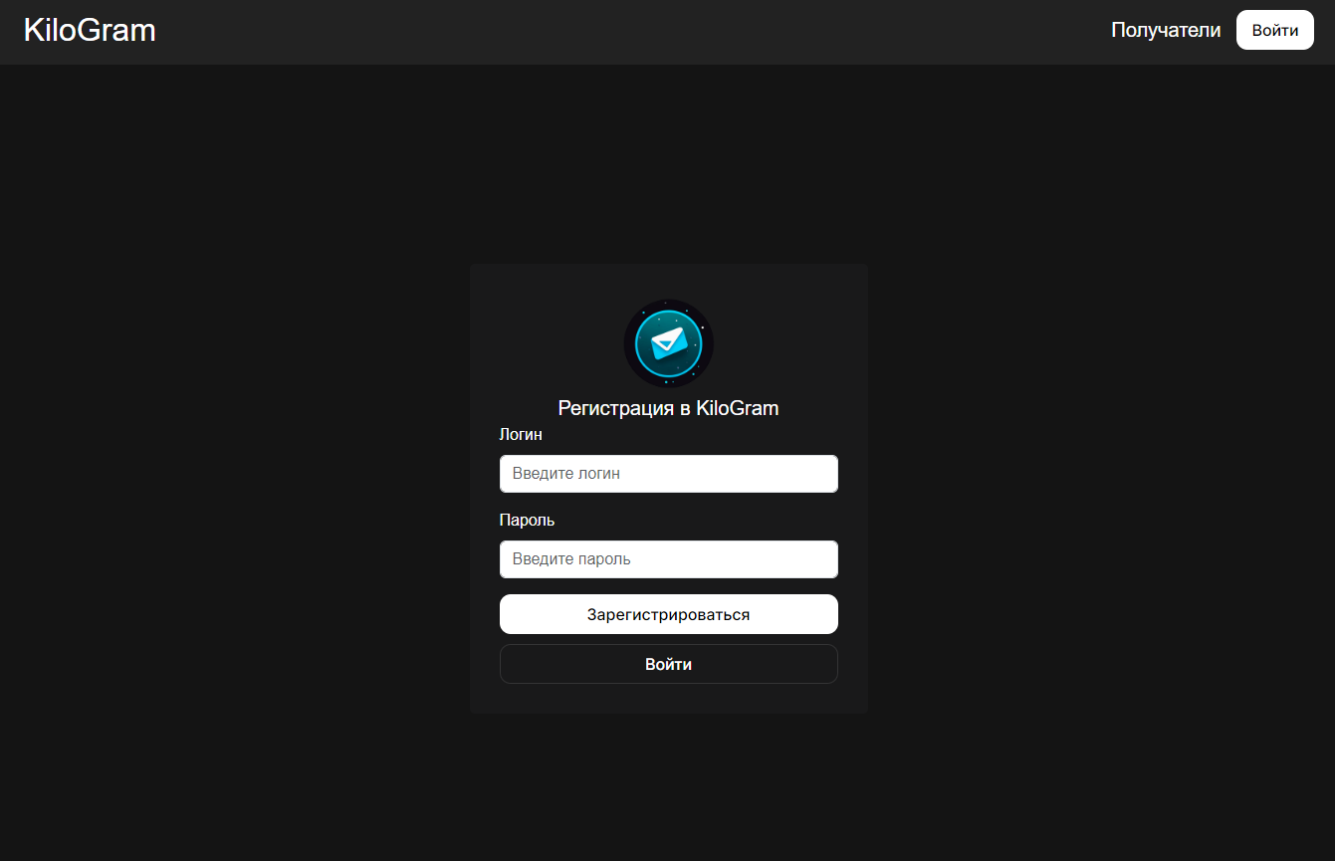


Рисунок 14 – Страница регистрации

На странице с формой входа (рисунок 15) отправитель может войти в свой аккаунт. При успешном вводе данных аккаунта на клиент приходит идентификатор сессии, который хранится в cookie браузера.

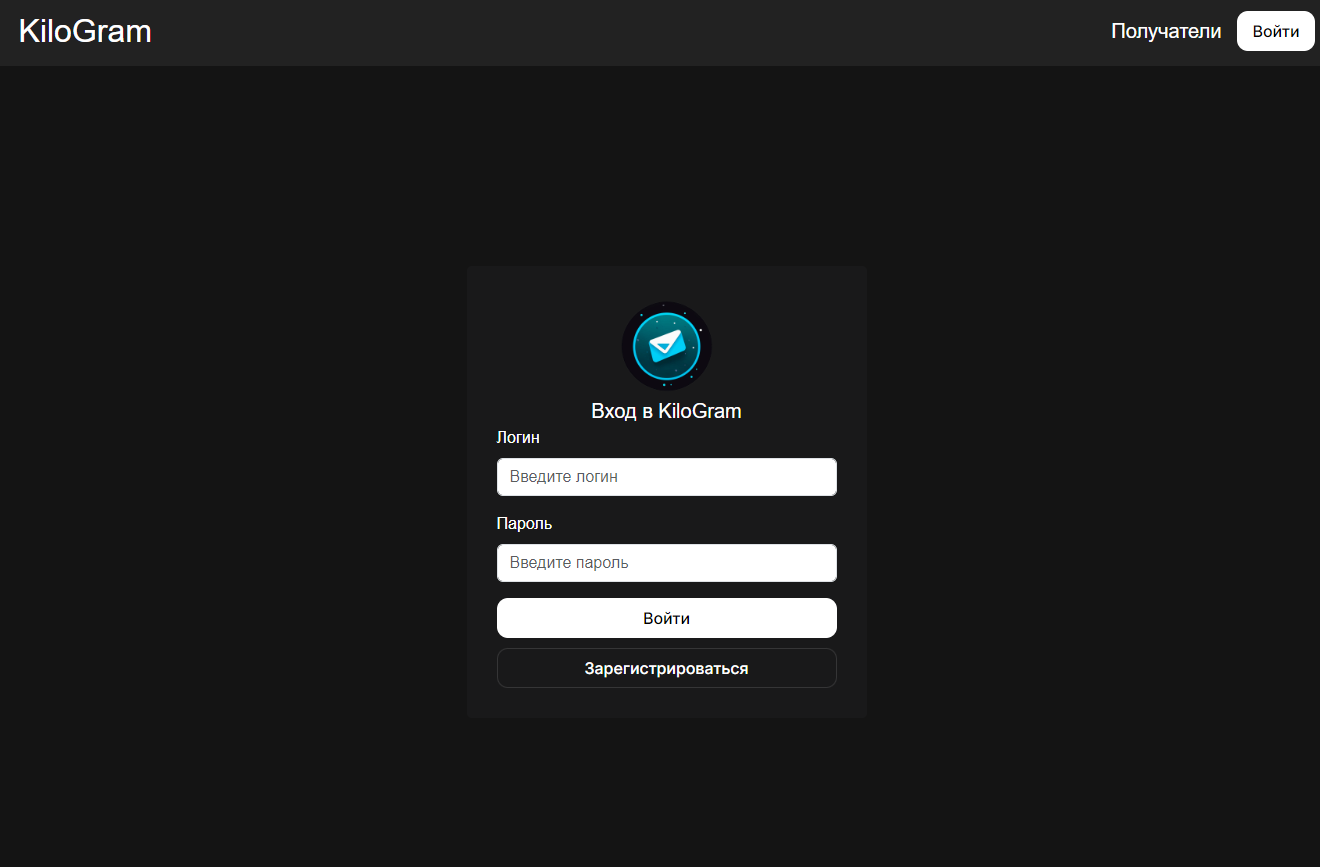


Рисунок 15 – Страница входа

После входа в аккаунт отправитель попадает на главную страницу (рисунок 16). На этой странице располагается описание разработанной системы.

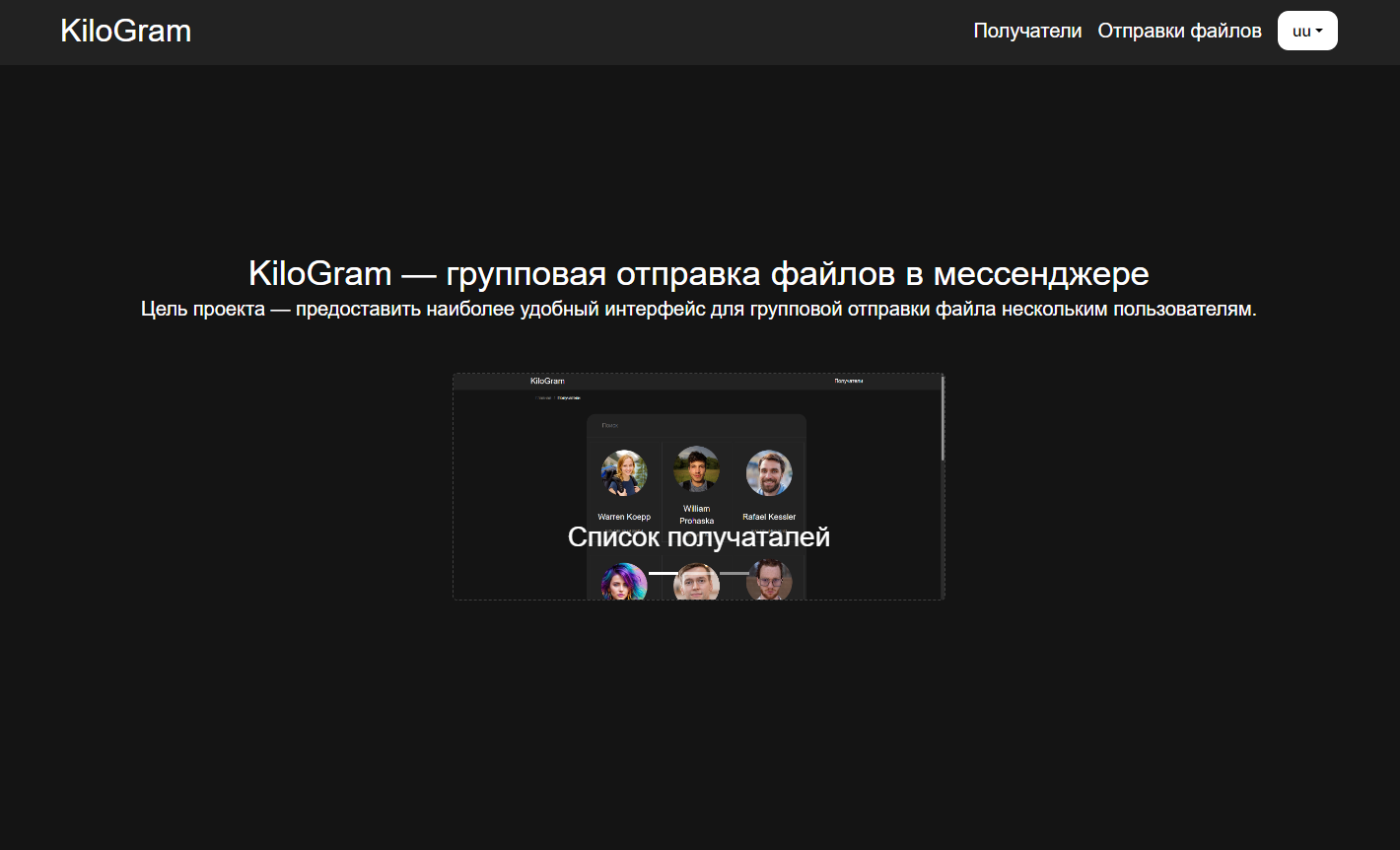


Рисунок 16 – Главная страница

Страница списка получателей (рисунок 17) содержит список доступных к добавлению в отправку получателей. При нажатии на имя получателя на любой карточке, открывается страница с подробной информацией о получателе. Также с помощью нажатия на кнопку «Добавить» пользователи могут добавлять получателей в черновую отправку файла. Вверху страницы находится строка для поиска получателей по имени.

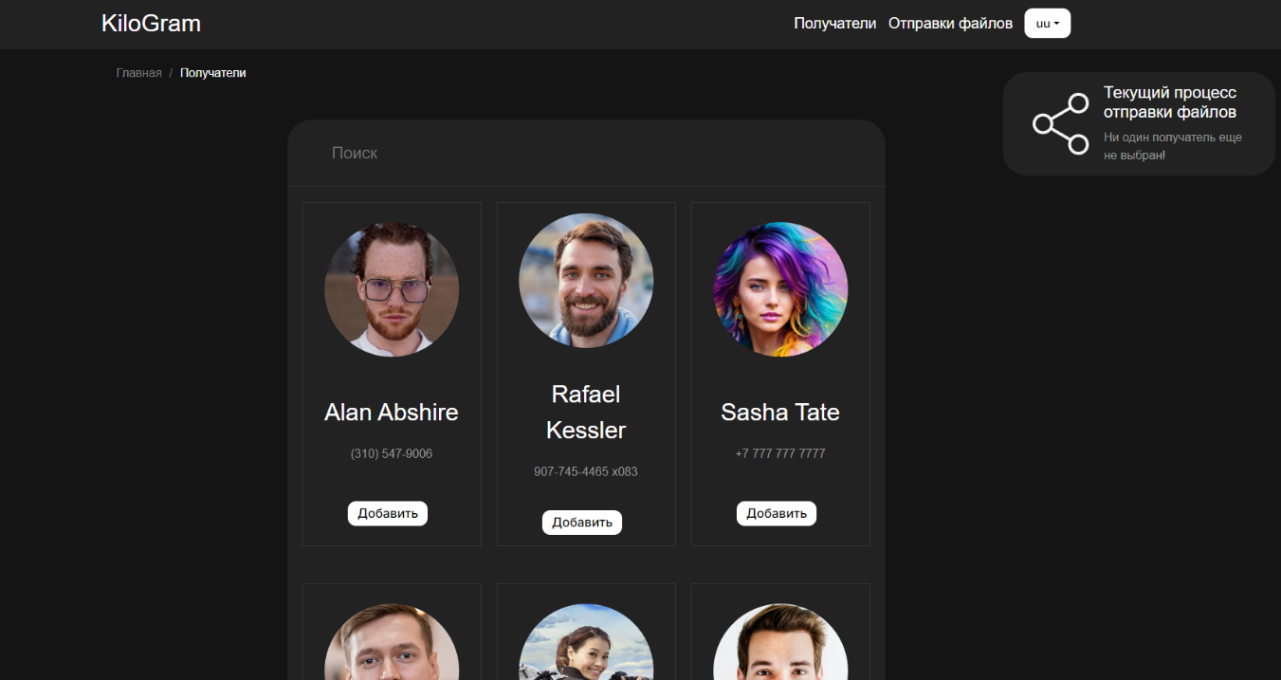


Рисунок 17 – Страница списка получателей

Страница с подробной информацией о получателе выглядит следующим образом (рисунок 18).

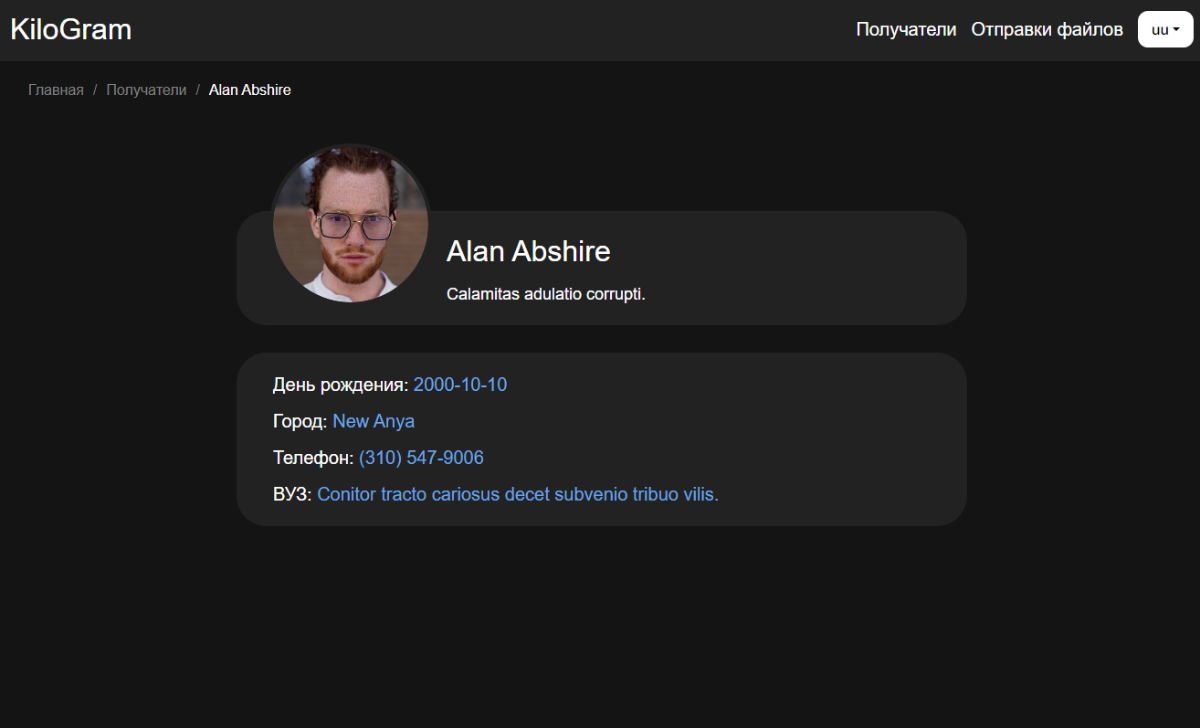


Рисунок 18 – Страница с подробной информацией о получателе

На странице одной отправки (рисунок 19) отображается текущая отправка файла, есть возможность удалить получателя из отправки, выбрать файл для отправки, указать индивидуальный комментарий для получателя, а также отправить отправку файла на проверку или удалить ее. Редактирование и удаление оформленных отправок запрещено.

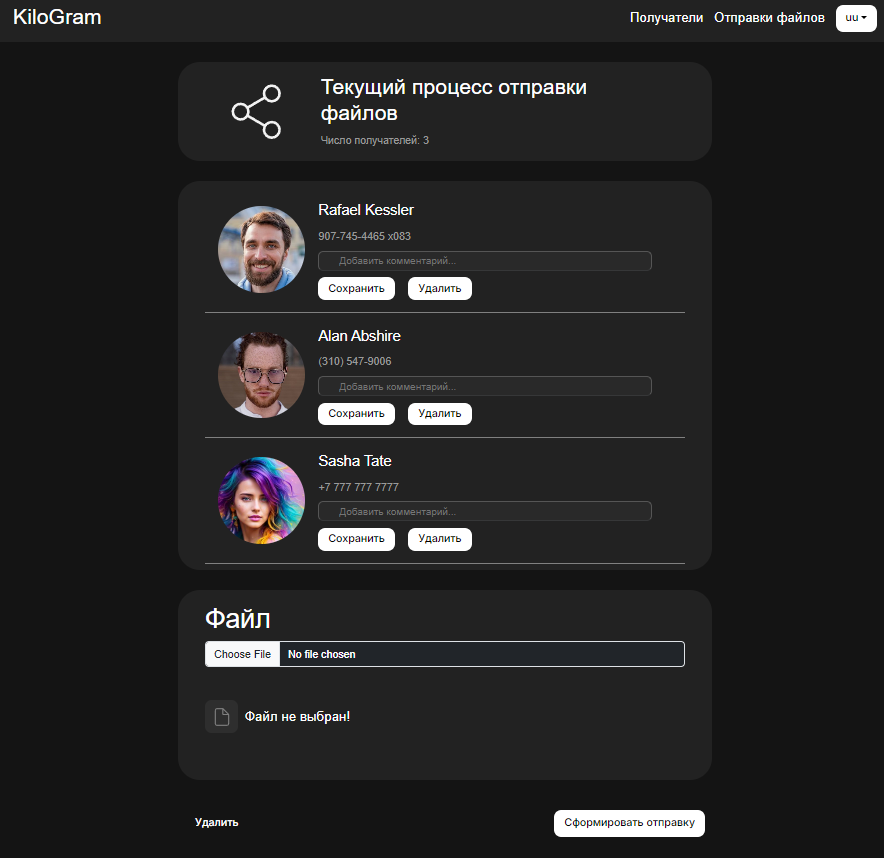


Рисунок 19 – Страница отправки файла

На странице списка заявок (рисунок 20) клиенты могут просматривать созданные ими отправки. На этой странице можно посмотреть подробную информацию об отправке, нажав на интересующую строку таблицы.

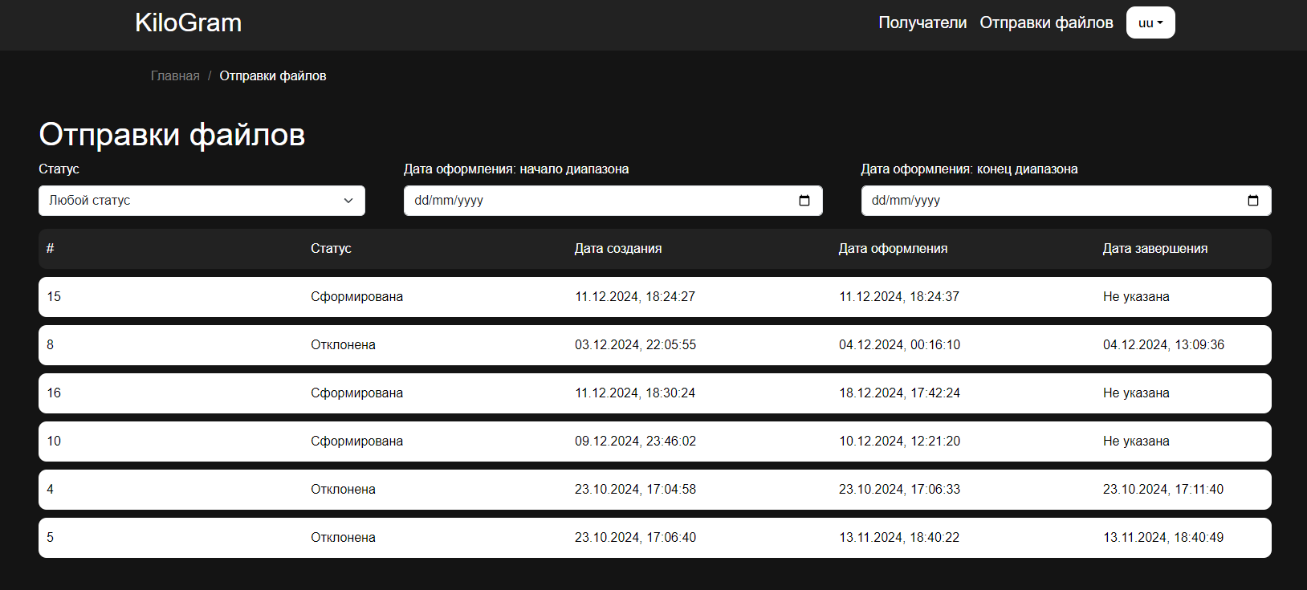


Рисунок 20 – Страница списка отправок файлов

Пользователь может изменить свой логин и пароль на странице редактирования данных пользователя (рисунок 21). Она открывается при нажатии на кнопку «Изменить профиль», которая находится в выпадающем меню, которое активируется нажатием на логин пользователя в меню.

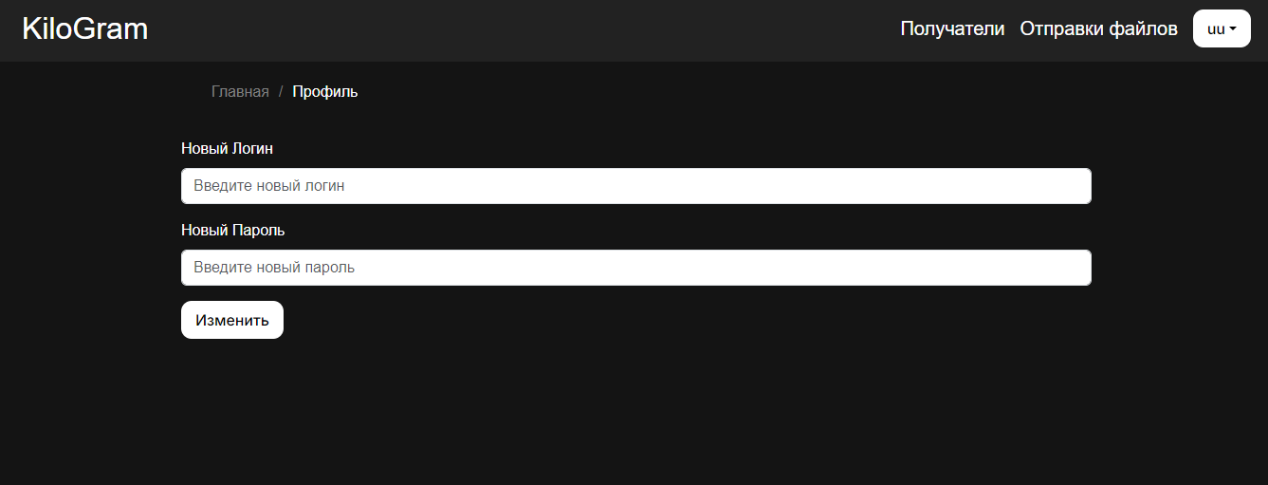


Рисунок 21 – Страница редактирования информации о пользователе

В случае, если пользователь запрашивает несуществующую страницу, он перенаправляется на страницу ошибки 404 (рисунок 22).

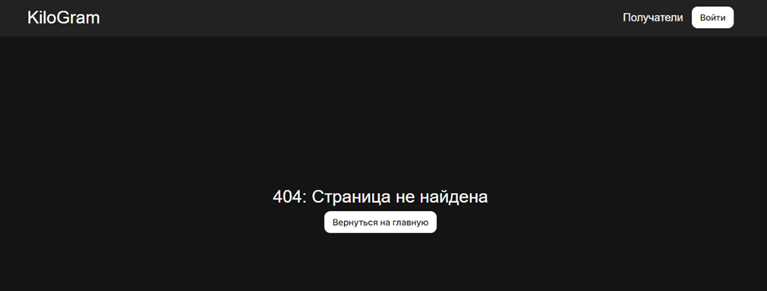


Рисунок 22 – Страница 404

В случае, если пользователь пытается получить доступ к страницам, для просмотра которых ему не хватает прав, он перенаправляется на страницу ошибки 403 (рисунок 23).

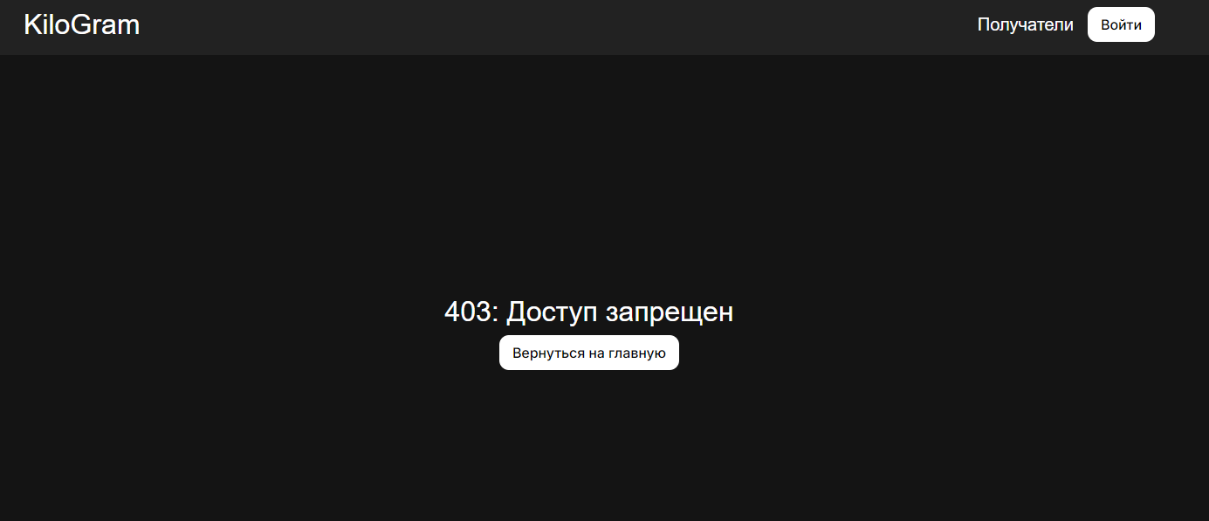


Рисунок 23 – Страница 403

Приложение поддерживает возможность работы как прогрессивное веб-приложение (PWA), что позволяет пользователю устанавливать его на устройство как нативное приложение. Интерфейс адаптирован для мобильных устройств, сохраняя функциональность основной версии, и состоит из трех страниц: главная страница PWA (рисунок 24, *а*), страница списка получателей (рисунок 24, *б*) и страница с подробной информацией о получателе (рисунок 24, *в*).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| *а* | *б* | | *в* |

Рисунок 24 – Интерфейс PWA

*а –* главная страница*; б* – страница списка получателей; *в* – страница с подробной информацией о получателе

Была разработана десктопная версия приложения на основе Tauri. Интерфейс состоит из трех страниц: главная страница (рисунок 25), страница списка получателей (рисунок 26) и страница с подробной информацией о получателе (рисунок 27).

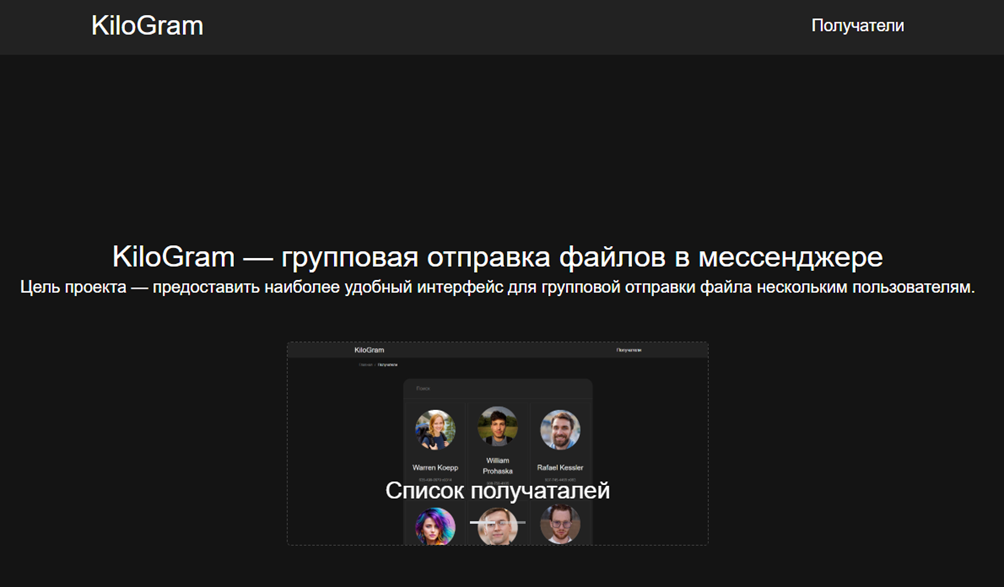


Рисунок 25 – Главная страница Tauri

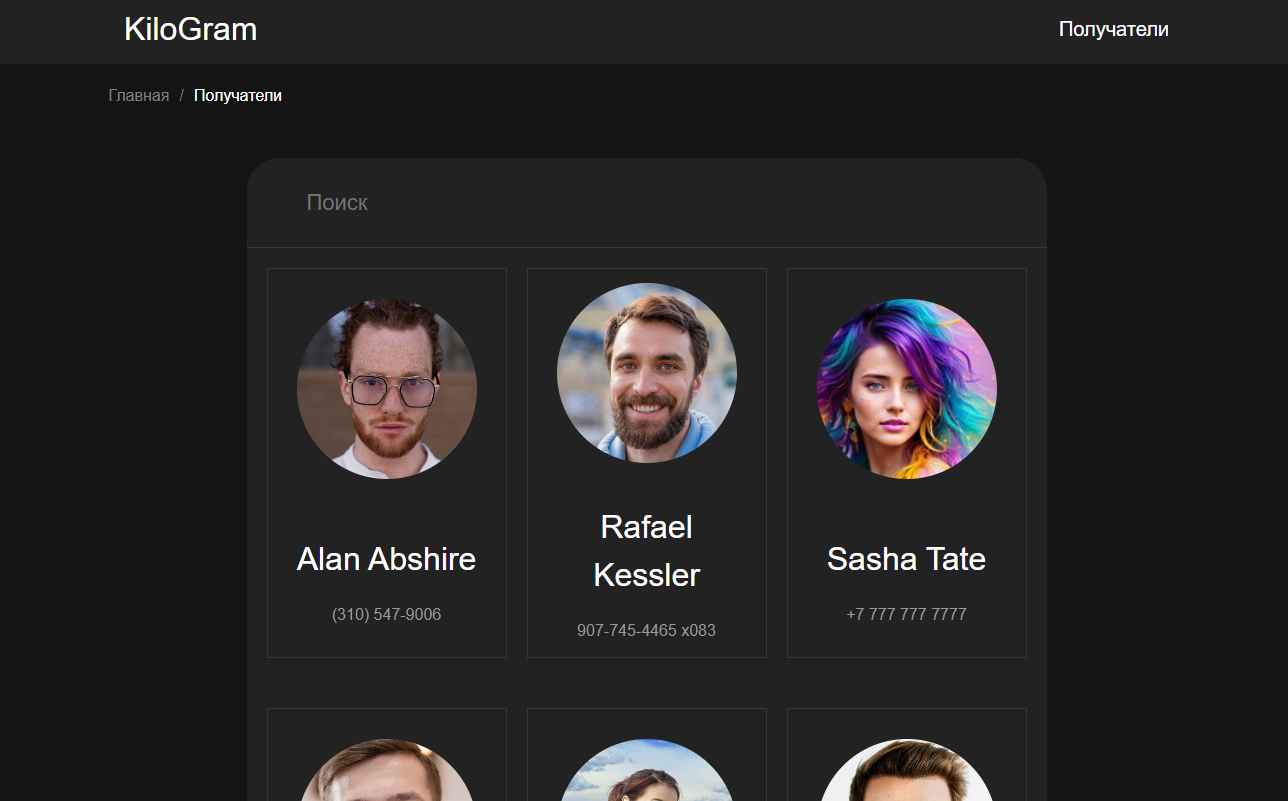


Рисунок 26 – Страница списка получателей Tauri

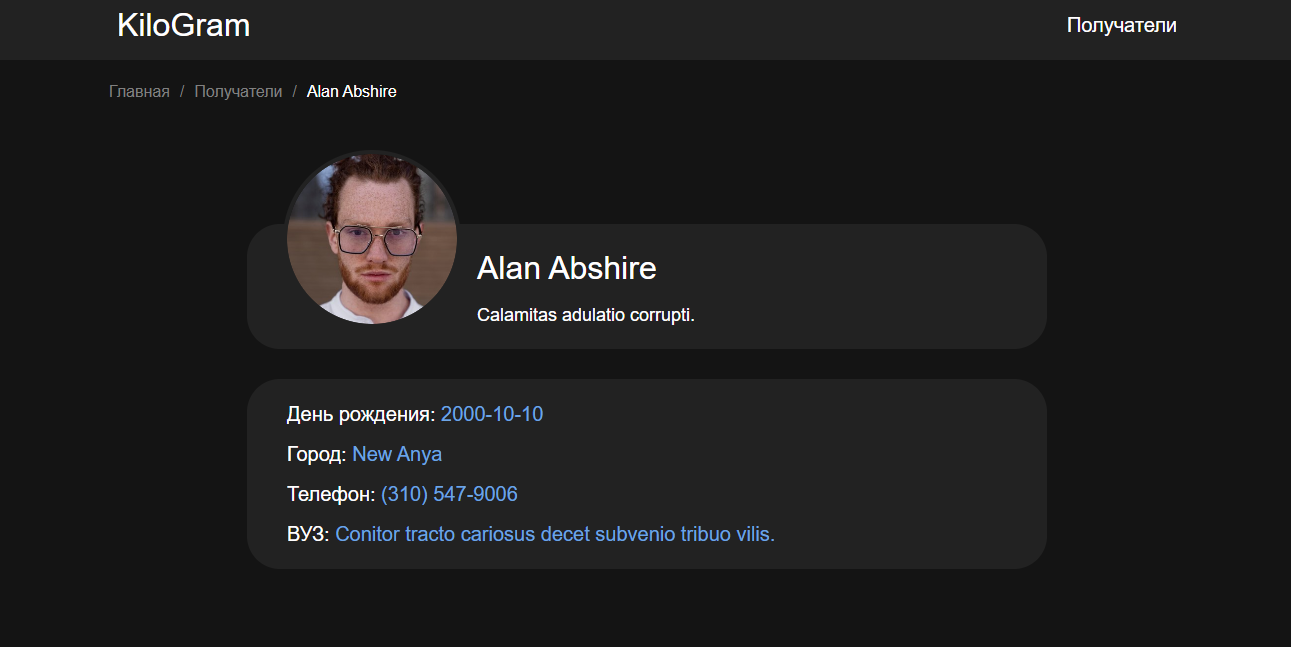


Рисунок 27 – Страница с подробной информацией о получателе Tauri

ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты следующие результаты:

* 1. Разработан дизайн приложения в Figma на основе vk.com. После ознакомления с разработкой бэкенда с использованием фреймворка Django создан MVP.
  2. Спроектирована и создана база данных PostgreSQL для хранения информации о получателях, отправках файлов и пользователей, а затем подключена к бэкенду.
  3. Создан веб-сервис на Django.
  4. Реализована авторизация и хранение сессий в Redis.
  5. Разработан базовый интерфейс приложения для гостя на React.
  6. Внедрена адаптивность, менеджер состояний Redux Toolkit, PWА, разработано Tauri приложение.
  7. Завершена разработка интерфейса пользователя в React, для обращений к методам веб-сервиса использован Axios.
  8. Реализован React интерфейс получателя, внедрен Real-time web.
  9. Реализовано десктопное приложение Tauri.
  10. Приложение развернуто при помощи GitHub Pages и доступно по ссылке: https://rasulov1337.github.io/kilogram-frontend/.
  11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
  12. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код работы: https://github.com/rasulov1337/kilogram-frontend.
  13. Произведен анализ сервисов-аналогов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация по Slack [Электронный ресурс]. // Slack URL: https://api.slack.com/docs (дата обращения: 28.10.2024);
2. Каким должен быть безопасный мессенджер [Электронный ресурс]. // Protectimus URL: https://www.protectimus.com/blog/ru-secure-messenger/ (дата обращения: 18.12.2024);
3. Как создать свой мессенджер: все, что нужно знать [Электронный ресурс]. // Лайв Тайпинг URL: https://livetyping.com/ru/kak-sozdat-prilozhenie-messenger (дата обращения: 18.12.2024);
4. Telegram FAQ [Электронный ресурс]. // Telegram FAQ URL: https://telegram.org/faq#q-what-makes-telegram-groups-cool (дата обращения: 23.12.2024);
5. Как отправлять медиафайлы, контакты или местоположение? [Электронный ресурс]. // WhatsApp FAQ URL: https://faq.whatsapp.com/591300876797857/?locale=ru\_RU (дата обращения: 23.12.2024);
6. Как отправлять медиафайлы, контакты или местоположение? [Электронный ресурс]. // Discord FAQ URL: https://support.discord.com/hc/en-us/articles/25444343291031-File-Attachments-FAQ (дата обращения: 23.12.2024);
7. Документация по Python [Электронный ресурс]. // Python URL: https://docs.python.org/3/ (дата обращения: 28.10.2024);
8. Документация по Django REST Framework [Электронный ресурс] // Django REST Framework. URL: https://www.django-rest-framework.org/ (дата обращения: 28.10.2024);
9. Документация по Redis [Электронный ресурс] // Netlify. URL: https://master--redis-doc.netlify.app/docs/ (дата обращения: 10.10.2024);
10. Документация Minio [Электронный ресурс] // Min. URL: https://min.io/docs/minio/kubernetes/upstream/index.html (дата обращения: 09.09.2024);
11. Документация PostgreSQL [Электронный ресурс] // Postgresql. URL: https://www.postgresql.org/docs/ (дата обращения: 15.09.2024);
12. Документация по Tauri [Электронный ресурс] // Tauri. URL: https://v2.tauri.app/develop/ (дата обращения: 20.11.2024);
13. Документация по React [Электронный ресурс] // React. URL: https://react.dev/learn (дата обращения: 01.11.2024);

ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Групповая отправка файла в мессенджере**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

7

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Расулов Арсен Низамудинович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. Введение

Система «KiloGram» представляет собой веб-сервис, включающий в себя веб-приложение и нативное приложение для получения информации о получателях и групповой отправки файла.

1. Назначение разработки

Данная система предназначена для хранения информации о получателях, файлах и отправках файла и отправки файлов нескольким получателям. Для этого отправитель может создать процесс отправки файла через автоматизированную систему, указав выбранных получателей и файл для отправки. Система осуществляет создание, учет и отслеживание статуса процессов отправок файла. Отправители могут загружать файл для последующей отправки. Получатели имеют возможность принимать или отклонять отправки файла, а также вносить изменения как в уже существующих получателей и добавлять новых, так и в уже существующих процессах отправки файлов.

1. Стадии и этапы разработки
   1. Создание MVP и базового дизайна на основе vk.com
   2. Создание базы данных для хранения информации о получателях и отправках файла в PostgreSQL
   3. Создание веб-сервиса на DRF
   4. Реализация авторизацию и хранение отозванных сессий в Redis
   5. Разработка SPA на React для гостя
   6. Внедрение адаптивности, менеджера состояний Redux Toolkit, PWА, разработка Tauri приложения
   7. Завершение разработки интерфейса отправителя на React
   8. Добавить в приложение интерфейс получателя на React
   9. Разработка десктопного приложения на Tauri
   10. Развертывание приложения при помощи GitHub Pages
   11. Подготовка всей документации (РПЗ, ТЗ и набор диаграмм)
   12. Оформление git-репозитория на сервисе GitHub
   13. Анализ сервисов-аналогов
2. Требования к функциональным характеристикам
   1. Методы HTTP
      1. GET Получение всех получателей
      2. POST Добавление получателя
      3. GET Получение получателя
      4. PUT Изменение получателя
      5. DELETE Удаление получателя
      6. POST Добавление получателя в процесс отправки файла
      7. POST Изменение/добавление картинки получателя
      8. GET Получение всех отправок файлов
      9. GET Получение отправки
      10. PUT Изменение дополнительных полей отправки
      11. DELETE Удаление получателя из отправки
      12. PUT Формирование отправки
      13. PUT Завершение отправки
      14. DELETE Удаление отправки
      15. PUT Изменение комментария для получателя в отправке
      16. POST Регистрация пользователя
      17. PUT Изменение данных пользователя (Личный кабинет)
      18. POST Аутентификация
      19. POST Деавторизация
   2. Меню
      1. KiloGram - перенаправляет на страницу 4.6
      2. Получатели – перенаправляет на страницу 4.7
      3. Отправки файлов – перенаправляет на страницу 4.10
      4. Выпадающее меню
         1. Изменить профиль – перенаправляет на страницу 4.5
         2. Выход – вызывается метод 4.1.19
   3. Регистрация
      1. Доступно только гостям
      2. Отображает форму регистрации
         1. Поле логина пользователя
         2. Поле пароля
      3. Действия
         1. Регистрация пользователя – (вызывается метод 4.1.16)
         2. Войти – перенаправляет на страницу 4.4
   4. Аутентификация
      1. Доступно только гостям
      2. Отображает форму аутентификации
         1. Поле логина
         2. Поле пароля
      3. Действия
         1. Войти – (вызывается метод 4.1.18)
         2. Регистрация – перенаправляет на страницу 4.3
   5. Личный кабинет
      1. Доступно аутентифицированному пользователю
      2. Действия
         1. Изменить данные пользователя – (вызывается метод 4.1.17)
   6. Главная
      1. Доступна всем
      2. Отображается статическая информация о сервисе
         1. Назначение сервиса
   7. Список получателей
      1. Доступна всем
      2. Отображаются элементы карточек с получателями, вызывается метод 4.1.1
         1. Имя получателя
         2. Номер телефона
      3. Действия
         1. Поиск – перенаправляет на страницу 4.7, (используется метод 4.1.1), с фильтрующем параметром
         2. Имя получателя – перенаправляет на страницу 4.8 (используется метод 4.1.3)
         3. Добавить – добавляет получателя в отправку-черновик, (вызывается метод 4.1.2), только аутентифицированные пользователи.
         4. Текущий процесс отправки файлов – перенаправляет на страницу 4.9, только аутентифицированные пользователи.
   8. Один получатель
      1. Доступна только авторизованным пользователям
      2. Отображается подробная информация о выбранном получателе, (вызывается метод 4.1.3)
   9. Одна отправка
      1. Доступно только аутентифицированным посетителям
      2. Отображает текущий заказ-черновик пользователя, (метод 4.1.9)
         1. Список выбранных получателей
         2. Поле комментария к отправке
         3. Поле с файлом к отправке
      3. Действия, доступны только в случае, если статус «черновик»
         1. Удалить получателя – удалят получателя из отправки, (вызывается метод 4.1.11)
         2. Сохранить – сохраняет комментарий для получателя, (вызывается метод 4.1.15)
         3. Удалить отправку – удаляет отправку-черновик, (вызывается метод 4.1.14)
         4. Заполнить – вносит данные отправки (вызывается метод 4.1.10)
         5. Изменить файл к отправке (вызывается метод 4.1.10)
   10. Список отправок
       1. Доступно аутентифицированному посетителю
       2. Отображается список отправок (метод 4.1.8)
          1. Только отправки, созданные данным отправителем, если он не получатель
          2. Все оформленные отправки в противном случае
       3. Действия
          1. Фильтрация – фильтрует отправки файла по диапазону даты формирования или статусу, вызывается (метод 4.1.8)
          2. Сформировать – формирует отправку, выполняется метод 4.1.13, доступно только отправителю
          3. Отклонить – отклоняет отправку, вызывается метод 4.1.13, доступно только получателю
          4. Посмотреть подробную информацию об отправке – перенаправляет на страницу 4.9 (вызывается метод 4.1.9)
   11. Список получателей таблицей
       1. Доступно только получателю
       2. Отображаются все существующие в системе получатели (вызывается метод 4.1.1)
       3. Действия
          1. Удалить – удаляет получателя (вызывается метод 4.1.5)
          2. Редактирование/создание – переход на страницу 4.12
   12. Редактирование/создание получателя
       1. Доступно только получателю
       2. Отображается информация об изменяемой/добавляемом получателе (вызывается метод 4.1.3)
          1. Имя
          2. Город
          3. Дата рождения
          4. Университет
          5. Номер телефона
          6. Описание
          7. Изображение
       3. Действия
          1. Сохранить – сохранение изменений в существующем получателе (вызывается метод 4.1.4)
          2. Добавить – добавление нового получателя (вызывается метод 4.1.2)
   13. 404
       1. Доступно всем
       2. Отображается в случае отсутствия ресурса
       3. Действия.
          1. Вернуться на главную – переход на страницу 4.6.
   14. 403
       1. Доступно всем
       2. Отображается в случае отсутствия прав на использование ресурса
       3. Действия.
          1. Вернуться на главную – переход на страницу 4.6.
3. Требования к составу и параметрам технических средств
   1. Сервер
      1. Процессор AMD Ryzen 5
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
   2. Клиент
      1. Процессор AMD Ryzen 5
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
4. Требования к информационной и программной совместимости
   1. Сервер
      1. ОС Ubuntu (24.10 и выше)
      2. Minio (RELEASE.2023-12-14T18-51-57Z)
      3. Redis (7.2)
      4. Node JS (1.23)
      5. PostgreSQL (17)
      6. Python 3.12
      7. Доступ к интернету
   2. Клиент
      1. Браузер (Safari 16.5.2, Firefox 121.0, Chrome 119.0.6045, Yandex 24.6.3.729, Opera 105.0.4970.16)

ПРИЛОЖЕНИЕ Б СПИСОК HTTP **МЕТОДОВ**

Таблица 2 – HTTP методы разрабатываемого веб-сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тип** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| 4.1.1 | GET | /recipients/ | Возвращает всех получателей  Доступно всем | recipient-name=string | [  {  id: int,  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  }  ],  draftId: int,  draftRecipientsLen: int |
| 4.1.2 | POST | /recipients/ | Добавляет получателя,  доступно получателю | {  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  } | {  id: int,  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  } |
| 4.1.3 | GET | /recipients/{pk}/ | Получает данные получателя,  Доступно только авторизованным пользователям | pk: int | {  id: int,  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  } |
| 4.1.4 | PUT | /recipients/{pk}/ | Обновляет данные получателя (кроме картинки), доступно менеджеру | pk: int  {  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  } | {  id: int,  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  } |
| 4.1.5 | DELETE | /recipients/{pk} | Удаляет получателя, доступно получателю | pk: int | {  status: string;  } |
| 4.1.6 | POST | /recipients/{pk}/draft/ | Добавляет получателя в отправку, доступно аутентифицированному пользователю | pk: int | {  status: string;  } |
| 4.1.7 | POST | /recipients/{pk}/avatar/ | Обновляет изображение получателя, доступно получателю | pk: int  {  avatar: Image } | {  id: int,  name: string,  desc: string,  phone: string,  city: string,  birthdate: string,  avatar: string,  uni: string  } |
| 4.1.8 | GET | /transfers/ | Получает список отправок файла, доступно отправителю и получателю | formed-at-range=string  status=string | [  {  id: int,  name: string  desc: string  phone: string,  city: string,  birthdate: string  avatar: string,  uni: string,  has\_read: boolean,  },  {  draftId: int,  draftRecipientsLen: int  }] |
| 4.1.9 | GET | /transfers/{pk} | Получает процесс отправки файла, доступно любому аутентифицированному пользователю | pk: int | {  id: int,  status: string,  created\_at: string,  formed\_at: string,  completed\_at: string,  sender: string,  moderator: string,  file: string,  recipients: [{}],  has\_read: boolean;  } |
| 4.1.10 | PUT | /transfers/{pk} | Обновляет поля отправки, доступно отправителю | pk: int  {  table\_number: date,  } | {  id: int,  status: string,  created\_at: string,  formed\_at: string,  completed\_at: string,  sender: string,  moderator: string,  file: string,  recipients: [{}],  } |
| 4.1.11 | DELETE | /transfers/{pk} | Удаляет заказ, доступно любому аутентифицированному посетителю | pk: int | {  status: string;  } |
| 4.1.12 | PUT | /transfers/{pk}/form | Формирует заказ, доступно отправителю | pk: int | {  status: string;  } |
| 4.1.13 | PUT | /transfers/{pk}/complete | Завершает отправку, доступно получателю | pk: int  {  status: string } | {  status: string;  } |
| 4.1.14 | DELETE | /transfers/{pk\_transfer}/dishes/ {pk\_recipient} | Удаляет получателя из отправки, доступно отправителю и получателю. | pk\_dinner: int,  pk\_dish: int | {  message: string,  } |
| 4.1.15 | PUT | /transfers/{pk\_transfer}/recipients/ {pk\_recipient} | Обновляет данные по М-М (поле comment). Доступно отправителю и получателю | pk\_transfer: int,  pk\_recipient: int  {  comment: string } | {  id: int,  comment: string,  sent\_at: string,  file\_transfer: int,  recipient: int  } |
| 4.1.16 | POST | /user / | Регистрирует пользователя, доступно всем | {  username: string,  password: string,  is\_staff: bool,  is\_superuser: bool } | {  status : string } |
| 4.1.17 | PUT | /users/profile/ | Обновляет данные пользователя. Доступно пользователю. | {  username: string,  password: string,  is\_staff: bool,  is\_superuser: bool } | {  message: string,  username: {  email: string,  password: string,  } } |
| 4.1.18 | POST | /signin/ | Аутентификация пользователя. Доступно всем | {  email: string,  password: string } | {  status: string  } |
| 4.1.19 | POST | /signout/ | Деавторизация, доступна дюбому аутентифицированному пользователю |  | {  status: string  } |