1. Описание бизнес-логики мобильного приложения

Бизнес-процессы мобильного приложения «Университетский мессенджер» обеспечивают полный цикл взаимодействия пользователя с приложением. Реализация этих процессов предусматривает высокую степень защиты данных, гибкость в управлении чатами и контактами, а также удобство интерфейса. Представленная на рисунке 1 диаграмма описывает ключевые действия пользователя, которые реализуются через клиент-серверное взаимодействие. Данный подход обеспечивает надёжную и эффективную работу приложения в образовательной среде университета.

Пользователь начинает работу с приложением с этапа регистрации или авторизации. При регистрации он вводит свои данные: имя, email, пароль. После успешной регистрации пользователь переходит к авторизации. Авторизация осуществляется путём ввода email и пароля. Сервер проверяет учетные данные и генерирует токен, который используется для последующих запросов. Если пользователь уже зарегистрирован, он может сразу перейти к авторизации, минуя этап регистрации.

После входа в систему приложение проверяет наличие новых уведомлений. Уведомления могут быть связаны с новыми сообщениями, добавлением в групповые чаты, изменениями в контактах.

После авторизации и проверки уведомлений пользователь попадает на главный экран, где может выбрать одно из следующих действий: открыть личный чат, открыть групповой чат, найти нужный контакт, выбрать файл для отправки, создать новый чат, удалить существующий чат.

При выборе личного или группового чата пользователь переходит в окно переписки. Личный чат открывается с конкретным собеседником, а групповой — с несколькими участниками. В обоих случаях доступны следующие функции: отправка и получение текстовых сообщений, обмен файлами, редактирование и удаление сообщений, получение уведомлений о прочтении сообщений другими участниками.

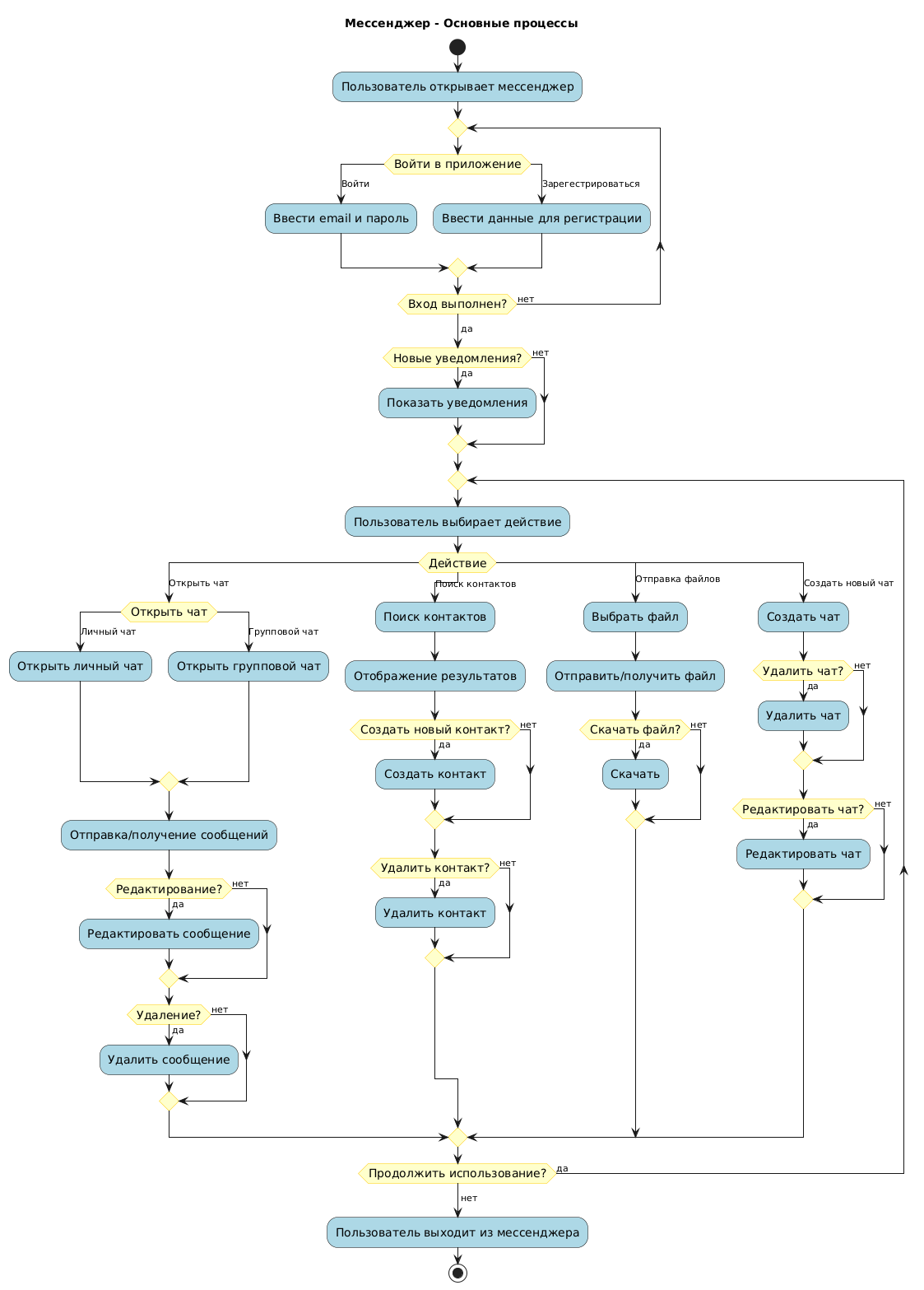


Рисунок 1 – Диаграмма, описывающая основные бизнес процессы

Поиск контактов. Для создания нового чата или начала диалога с другим пользователем предусмотрена функция поиска контактов. Найденные контакты отображаются в списке. Пользователь может выбрать нужный контакт и начать общение.

Функция обмена файлами позволяет пользователю прикреплять документы и другие поддерживаемые форматы к сообщению. Пользователь выбирает файл с устройства, после чего он загружается на сервер. Сервер проверяет тип и размер файла. Файл становится доступен для всех участников чата.

Пользователь может создать как личный, так и групповой чат. При создании группового чата необходимо выбрать не менее двух участников. После этого сервер создаёт запись в таблице Chats и добавляет выбранных пользователей в список участников.

Пользователь может удалить чат, если он является его администратором.

Если пользователь имеет право на удаление, чат помечается как удалённый, и данное изменение отражается у всех участников. Если у пользователя нет прав, ему показывается соответствующее уведомление.

Пользователь может редактировать или удалять свои сообщения и файлы. Редактирование доступно в течение определённого времени (например, 1 час). Удаление возможно только для собственных сообщений и файлов. После удаления элемент помечается как удалённый, и у всех участников отображается соответствующая метка.

Во время работы приложения пользователь получает push-уведомления о новых сообщениях, добавлении в чаты и других событиях. Также предусмотрена возможность настройки уведомлений: пользователь может отключить их полностью или для конкретного чата.

Пользователь может выйти из аккаунта в любое время. При выходе все активные сессии закрываются, статус пользователя меняется на «оффлайн». Это позволяет соблюдать конфиденциальность и предотвращать несанкционированный доступ к данным.

1. Основные функции приложения

Основные функции приложения (в порядке убывания важности для пользователя):

1. Отправка и получение сообщений: открывается окно чата, пользователь пишет и отправляет сообщение. Условия: пользователь должен быть зарегистрирован. Результат: сообщение доставляется собеседнику.
2. Групповые чаты: Пользователь создаёт новый чат и добавляет участников, все участники могут писать в чат. Условия: не менее 2 участников. Результат: чат создан, все могут обмениваться сообщениями.
3. Уведомления: приходит новое сообщение и уведомление, пользователь может открыть чат через уведомление. Условия: подключение к интернету. Результат: пользователь получает информацию о новом сообщении.
4. История сообщений: пользователь видит все предыдущие сообщения чата. Условия: пользователь имеет доступ к чату. Результат: полная история сообщений загружена, можно пролистывать назад.
5. Поиск контактов. Условия: база данных содержит данные пользователей. Результат: найденные пользователи отображаются, можно начать чат или добавить в контакты.
6. Отправка файлов: пользователь нажимает кнопку "Прикрепить файл", выбирает файл с устройства, файл загружается и отправляется в чат. Условия: поддерживаемые форматы: PDF, DOCX и т.д. Ограничение на размер файла. Результат: файл успешно передан, получатель может его скачать.
7. Статус онлайн/оффлайн. Условия: поддержка WebSocket или фоновых запросов. Результат: индикатор корректно отражает состояние пользователя.
8. Редактирование и удаление сообщений: пользователь редактирует на сообщение, изменения отображаются у всех участников чата. Условия: сообщение не старше 1 часа, только автор может редактировать/удалять своё сообщение. Ожидаемый результат: сообщение изменено или удалено, изменения синхронизированы у всех.
9. Процесс синхронизации данных между клиентом и сервером
   1. API - методы

API мобильного приложения «Университетский мессенджер» включает широкий набор методов, обеспечивающих полноценное взаимодействие пользователя с системой: регистрация и авторизация реализуются через методы POST /api/auth/register и POST /api/auth/login, а получение данных пользователя — через GET /api/users/{userId}. Для работы с чатами предусмотрены методы создания (POST /api/chats), редактирования (PUT /api/chats/{chatId}), удаления (DELETE /api/chats/{chatId}), просмотра одного чата (GET /api/chats/{chatId}) и получения списка чатов пользователя (GET /api/chats?userId={userId}). Обмен сообщениями осуществляется с помощью методов отправки (POST /api/messages), получения (GET /api/messages/{messageId}), редактирования (PUT /api/messages/{messageId}) и удаления (DELETE /api/messages/{messageId}) сообщений. Управление контактами включает добавление (POST /api/contacts), удаление (DELETE /api/contacts/{contactId}), просмотр конкретного контакта (GET /api/contacts/{contactId}) и вывод списка контактов (GET /api/contacts?userId={userId}).

* 1. ER – диаграмма сущностей

ER-диаграмма, представленная на рисунке 2, описывает структуру базы данных мессенджера и включает основные сущности: USER (пользователи), CHAT (чаты), MESSAGE (сообщения), FILE (файлы), CONTACT (контакты), NOTIFICATION (уведомления) и USER\_CHAT (участники чатов). Связи между сущностями обеспечивают хранение истории переписки, управление контактами, отслеживание участников чата и доставку уведомлений. Каждый пользователь может иметь несколько чатов и сообщений, а каждый файл привязан к конкретному сообщению.

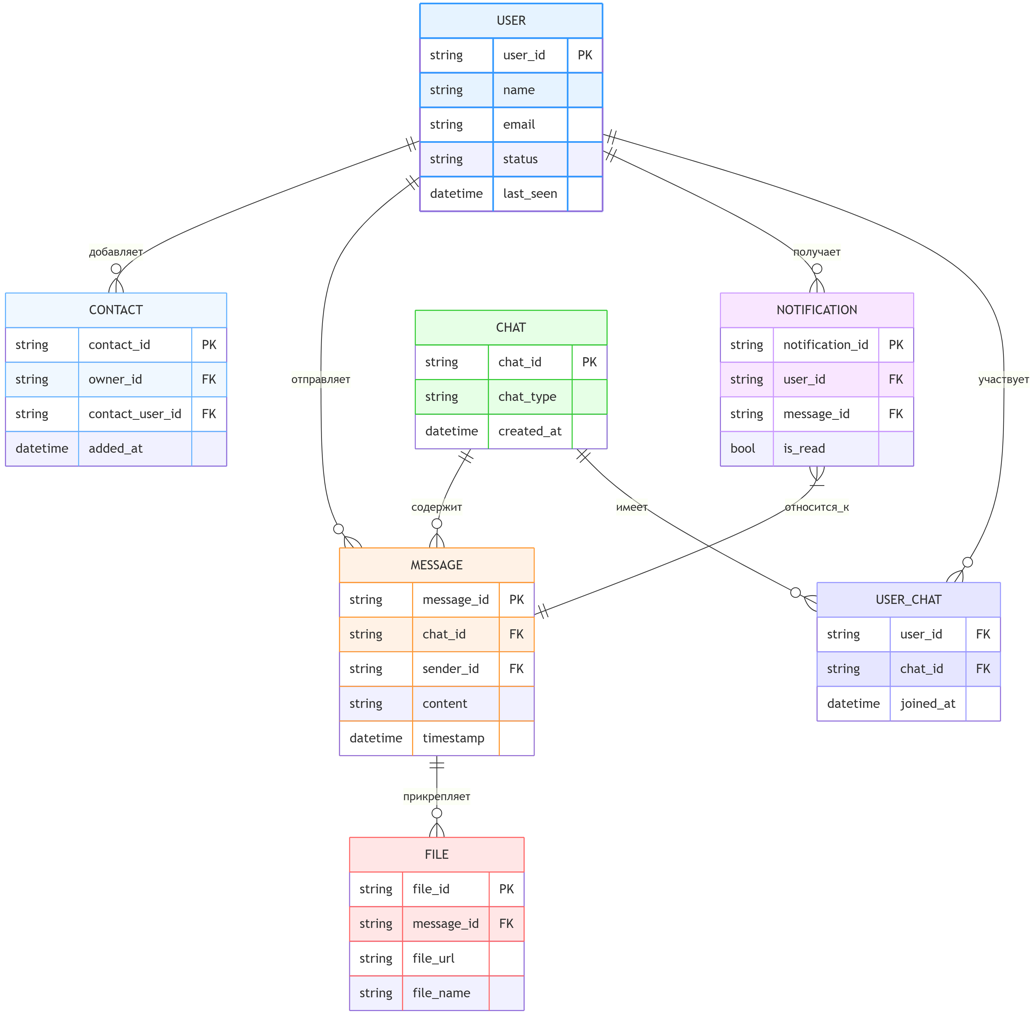


Рисунок 2 – ER диаграмма сущностей

* 1. Диаграмма последовательности

Диаграмма последовательности, представленная на рисунке 3, демонстрирует основные операции мессенджера, включая регистрацию и авторизацию пользователя, создание чатов и контактов, просмотр списков чатов и контактов, отправку сообщений и файлов, а также удаление и редактирование данных. Пользователь регистрируется с помощью имени, email и пароля, после чего получает токен авторизации и обновляет свой статус на "online". Затем пользователь может создавать новые чаты и контакты, просматривать их списки, отправлять текстовые сообщения и прикреплять файлы. При отправке сообщения сервер сохраняет его и возвращает message\_id, а для файлов — URL. Для управления данными пользователь может удалять чаты, контакты и сообщения, а также редактировать существующие сообщения. Сервер проверяет права доступа перед выполнением операций и уведомляет участников о любых изменениях.

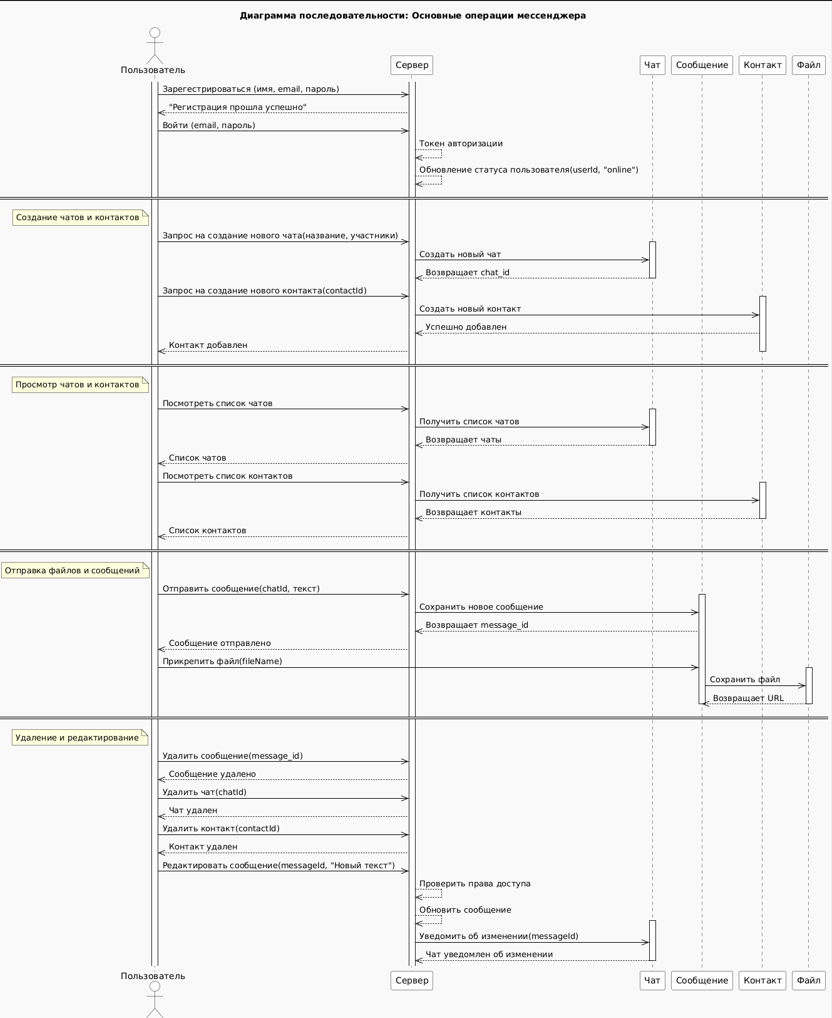
****

Рисунок 3 – Диаграмма последовательности

1. Прототип экрана мобильного приложения

Макет экрана, показывающий расположение чатов, представлен на рисунке 4.

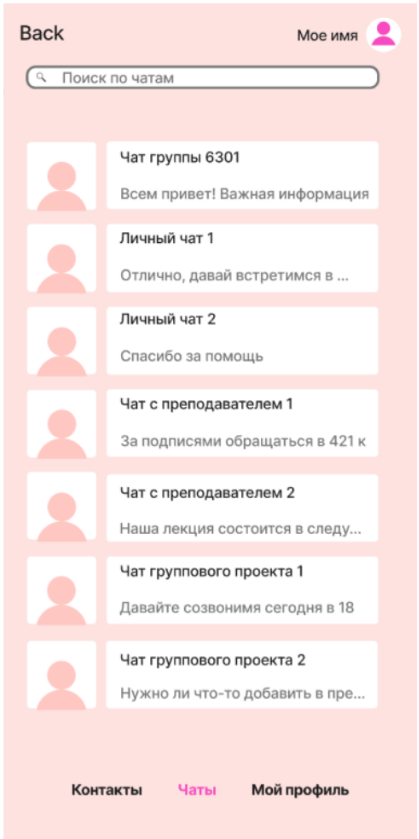


Рисунок 4 – Прототип экрана

Описание поведения пользовательского интерфейса:

Верхняя панель

* Нажатие на кнопку "Back" возвращает пользователя к предыдущему экрану (например, главному меню или профилю).
* Пользователь может ввести текст в поле поиска.
* Если результатов поиска нет, показывается сообщение: "Чаты не найдены" .
* Клик по аватарке/иконке профиля открывает экран с информацией о пользователе (например, имя, статус, настройки).

Список чатов

* Отображается аватарка собеседника/группы.
* Название чата (например, "Личный чат", "Чат группы").
* Последнее сообщение в чате.
* Переход в экран чата выбранного диалога.
* Если чат содержит непрочитанные сообщения, состояние изменяется на прочитанное после перехода.
* Если у пользователя нет чатов, отображается сообщение: "У вас пока нет чатов. Создайте новый!"

Нижняя панель

* Клик по каждому табу переключает содержимое экрана:
* Контакты: Отображаются контакты пользователя.
* Чаты: Отображаются текущие чаты (текущее состояние).
* Мой профиль: Открывается экран профиля пользователя.
* Если данные ещё не загружены: отображается индикатор загрузки (например, крутящийся круг или текст "Загрузка..." ).
* Если произошла ошибка при получении данных: показывается сообщение: "Ошибка получения данных. Попробуйте обновить страницу."

1. Подробное описание функции работы с файлами
   1. Сценарии использования

Перед отправкой файла пользователь должен быть авторизован в приложении, а сервер должен быть доступен и готов к обработке запросов; также должно быть установлено интернет-соединение, файл должен соответствовать ограничениям по формату и размеру. Пользователь выбирает файл из галереи или файлового менеджера, после чего клиент начинает его передачу на сервер, где происходит проверка формата и размера файла, далее он сохраняется во временном хранилище, а в базе данных создаётся соответствующая запись. В случае, если файл превышает допустимый размер, пользователю показывается сообщение об ошибке.

Перед тем как пользователь может получить или скачать файл от собеседника, он должен быть авторизован в приложении, сервер должен быть доступен и готов обрабатывать запросы, интернет-соединение должно быть установлено, пользователь должен иметь права на просмотр/скачивание файла (в зависимости от политики доступа), а на устройстве должно быть достаточно места для сохранения файла. Пользователь нажимает на файл в сообщении, клиент отправляет запрос на сервер по ID файла через API, сервер проверяет права доступа пользователя, после чего возвращает ссылку на файл. В случае, если файл уже был удалён, показывается сообщение "Файл недоступен", а если у пользователя нет прав на скачивание, выводится предупреждение "Доступ запрещён".

Для удаления файла пользователь должен быть авторизован, файл должен находиться в списке сообщений, пользователь должен быть его автором и сервер должен быть доступен. Пользователь нажимает кнопку "Удалить" рядом с файлом, клиент отправляет запрос на сервер. Сервер проверяет, существует ли файл и является ли пользователь его автором; если условия выполнены, файл физически удаляется. Если пользователь пытается удалить чужой файл, сервер возвращает ошибку и клиент показывает сообщение «Невозможно удалить чужой файл».

* 1. Диаграмма вариантов использования

Диаграмма вариантов использования на рисунке 5 иллюстрирует основные функциональные сценарии работы с файлами в мессенджере. Пользователь может выполнять следующие действия: отправить файл, получить файл, просмотреть список файлов, удалить файл.

Диаграмма также показывает взаимодействие между пользователями, сервером и файловым хранилищем, а операция "Выбрать файл для отправки" включена в процесс "Отправить файл". Это обеспечивает чёткую структуру бизнес-процессов по работе с файлами в приложении.

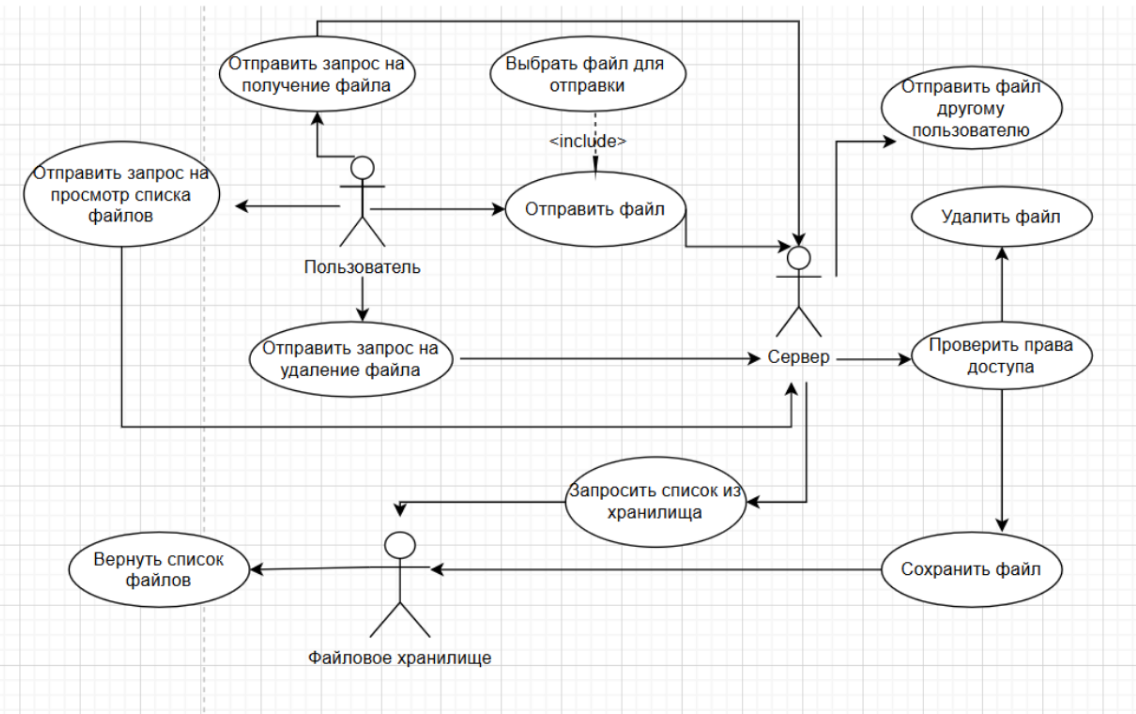


Рисунок 5 – Диаграмма вариантов использования

* 1. UML диаграмма последовательности

Диаграмма последовательности на рисунке 6 демонстрирует основные операции с файлами в мессенджере, включая отправку, получение, удаление и просмотр списка файлов.

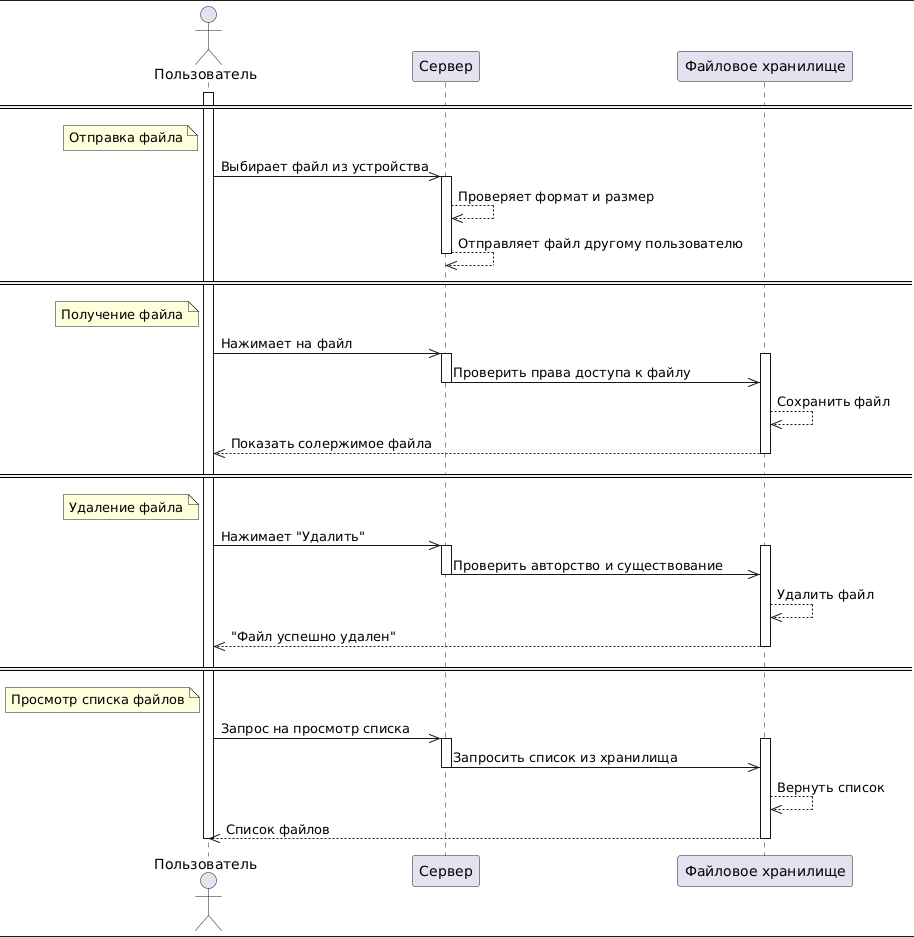


Рисунок 6 – Диаграмма последовательности для функции работы с файлами

* 1. API – методы

В таблице 1 представлены основные API – методы, которые используются при работе с файлами. Также подробно расписаны тела запросов и успешные ответы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод | URL | Назначение | Тело запроса | Ответ |
| POST | /api/files | Отправка файла | curl -X POST https://server.example/api/files  \  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  -H "Content-Type: multipart/form-data" \  -F "file=@report.pdf" \  -F "chatId=chat\_001" \  -F "senderId=user\_123" | {  "fileId": "file\_456",  "fileName": "report.pdf",  "url": "https://server.example/files/file\_456",  "uploadedAt":  "2025-04-05T10:00:00Z"  } |
| GET | /api/files/{fileId} | Получение файла | curl -X GET https://server.example/api/files/file\_456  \  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." | {  "fileId": "file\_456",  "fileName": "report.pdf",  "size": 123456,  "downloadUrl": "https://server.example/files/file\_456/download",  "uploadedAt": "2025-04-05T10:00:00Z"  } |
| DELETE | /api/files/{fileId} | Удаление файла | curl -X DELETE https://server.example/api/files/file\_456  \  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." | {  "status": "deleted",  "fileId": "file\_456",  "message": "Файл успешно удалён"  } |
| GET | /api/chats/  {chatId}/files | Просмотр всех файлов чата | curl -X GET https://server.example/api/chats/chat\_001/files  \  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." | [  {"fileId": "file\_456",  "fileName": "report.pdf",  "senderId": "user\_123",  "senderName": "Иван Иванов",  "size": 123456,  },{  "fileId": "file\_789",  "fileName": "photo.jpg",  "senderId": "user\_456",  "senderName": "Мария Смирнова",  "size": 456789,  }] |

Таблица 1 – Основные API-методы для работы с файлами

* 1. Процесс хранения файлов в системе

Файлы в мессенджере хранятся во временном хранилище после проверки формата и размера, при этом в базе данных сохраняются их метаданные — имя, ID, ссылка для скачивания и статус; доступ к файлам возможен только у тех пользователей, которые участвуют в соответствующем чате и имеют на это права, а каждый файл может находиться в одном из состояний: загрузка (в процессе отправки), готов (успешно загружен и доступен для скачивания), ошибка (файл не прошёл проверку или не был загружен) или удалён (помечен как удалённый и скрыт из интерфейса).

1. Сложности разработки и вопросы заказчику

Разработка клиент-серверного мобильного мессенджера связана с рядом технических и функциональных сложностей. Одной из ключевых задач является обеспечение надежной и быстрой синхронизации данных между клиентом и сервером, особенно при работе с большими объемами сообщений и файлов. Обмен файлами сопряжен с необходимостью проверки форматов, ограничения размера, а также организации безопасного хранения и предоставления к ним доступа. Также возникают трудности с обеспечением безопасности данных — требуется продумать механизмы авторизации, шифрования и контроля доступа.

Список уточняющих вопросов для заказчика:

1. Нужна ли поддержка групповых звонков (аудио/видео)? Если да, то требуется ли запись или трансляция?
2. Должны ли быть постоянный чат для каждой учебной группы или пользователи могут создавать чаты самостоятельно?
3. Нужно ли хранить файлы локально или использовать облачное хранилище (например, Firebase Storage, AWS S3)?
4. Какие платформы необходимо поддерживать (Android, iOS, Web)?
5. Есть ли ограничения по времени разработки , которые могут повлиять на объём реализуемого функционала?
6. Есть ли ограничения по стилю интерфейса (например, корпоративный стиль, цветовая палитра)?
7. Нужна ли поддержка темного режима ?
8. Как часто пользователи будут обмениваться файлами? Нужна ли возможность поиска файлов по чату ?
9. Какие данные о пользователях должны быть доступны другим (статус онлайн/оффлайн, дата последнего входа)?
10. Какое поведение должно быть у чата при выходе пользователя — должен ли он оставаться в списке или архивироваться?
11. Создание таблиц и SQL – запросов

В таблице 2 показана структура хранения информации и чатах, где:

* Id: Уникальный идентификатор чата.
* UserId: Идентификатор пользователя (владельца чата).
* ChatName: Название чата.
* MessageCount: Количество сообщений в чате.

Таблица 2 - Чаты

|  |  |  |  |
| --- | --- | --- | --- |
| ID | USERID | CHATNAME | MESSAGECOUNT |
| 1 | 1 | Чат группы 6301 | 5 |
| 2 | 2 | Личный чат 1 | 10 |
| 3 | 3 | Личный чат 2 | 7 |
| 4 | 4 | Чат с преподавателем 1 | 3 |
| 5 | 5 | Чат с преподавателем 2 | 8 |
| 6 | 6 | Чат группового проекта 1 | 15 |
| 7 | 7 | Чат группового проекта 2 | 9 |

Таблица 3 демонстрирует структуру хранения инфорации и пользователях, где:

* Id: Уникальный идентификатор пользователя.
* UserName: Имя пользователя.

Таблица 3 - Пользователи

|  |  |
| --- | --- |
| ID | USERNAME |
| 1 | Иван Петров |
| 2 | Мария Смирнова |
| 3 | Сергей Иванов |
| 4 | Екатерина Козлова |
| 5 | Антон Сидоров |
| 6 | Ольга Васильева |
| 7 | Петр Волков |

Основные SQL-запросы для мессенджера представлены в таблице 4.

Таблица 4 – SQL-запросы

|  |  |
| --- | --- |
| Название запроса | Запрос на языке SQL |
| Общее количество сообщений для каждого пользователя | SELECT      Users.UserName AS Автор,      SUM(Chats.MessageCount) AS Общее\_количество\_сообщений  FROM      Chats  JOIN      Users ON Chats.UserId = Users.Id  GROUP BY      Users.UserName  ORDER BY      Общее\_количество\_сообщений DESC; |
| Пользователи, которые отправили больше 10 сообщений | SELECT      Users.UserName AS Автор,      SUM(Chats.MessageCount) AS Общее\_количество\_сообщений  FROM      Chats  JOIN      Users ON Chats.UserId = Users.Id  GROUP BY      Users.UserName  HAVING      SUM(Chats.MessageCount) > 10; |
| Вывести пользователей с количеством чатов | SELECT      Users.UserName AS Автор,      COUNT(Chats.Id) AS Количество\_чатов  FROM      Users  LEFT JOIN      Chats ON Users.Id = Chats.UserId  GROUP BY      Users.UserName; |
| Пользователи без чатов | SELECT      Users.UserName AS Автор  FROM      Users  LEFT JOIN      Chats ON Users.Id = Chats.UserId  WHERE      Chats.Id IS NULL; |