МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-98 01 03 “Программное обеспечение информационной

безопасности мобильных систем”

Специализация Инженер-программист

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

Мобильное приложение мессенджер

Выполнил студент Каракозова Регина Витальевна

(Ф.И.О.)

Руководитель проекта ассистент Уласевич Н. И. (учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой ст. преподаватель Блинова Е. А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: ассистент Уласевич Н. И. (учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: ассистент Уласевич Н. И. (учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2024

**СОДЕРЖАНИЕ**

[Введение](#_Toc153981770)

[1 Обзор аналогов приложения 5](#_Toc153981771)

[1.1 Аналог «WhatsApp» 5](#_Toc153981772)

[1.2 Аналог «Telegram» 6](#_Toc153981773)

[1.3 Аналог «KakaoTalk» 9](#_Toc153981774)

[1.4 Выводы по главе «Обзор аналогов приложения»Аналог 10](#_Toc153981774)

[2 Проектирование приложения 11](#_Toc153981775)

[2.1 Диаграмма вариантов использования 11](#_Toc153981776)

[2.2 Архитектура приложения 12](#_Toc153981777)

[2.3 Проектирование базы данных 13](#_Toc153981778)

[2.4 Выводы по главе «Проектирование приложения» 14](#_Toc153981779)

[3 Реализация приложения 15](#_Toc153981780)

[3.1 Разработка базы данных 15](#_Toc153981781)

[3.2 Подключение базы данных 16](#_Toc153981784)

[3.3 Разработка мобильного приложения 16](#_Toc153981788)

[3.4 Разработка пользовательского интерфейса 18](#_Toc153981791)

[3.5 Выводы по главе «Реализация приложения» 19](#_Toc153981791)

[4 Тестирование приложения 20](#_Toc153981792)

[4.1 Тестирование ограничения доступа 20](#_Toc153981793)

[4.2 Тестирование интерфейса 22](#_Toc153981795)

[4.3 Выводы по главе «Тестирование приложения» 23](#_Toc153981796)

[5 Руководство пользователя 24](#_Toc153981797)

[5.1 Мобильное приложение 24](#_Toc153981798)

[5.2 Выводы по главе «Руководство пользователя» 34](#_Toc153981800)

[Заключение 35](#_Toc153981801)

[Список использованных источников 36](#_Toc153981802)

[ПРИЛОЖЕНИЕ А. Листинг Диаграммы вариантов использования 37](#_Toc153981803)

[ПРИЛОЖЕНИЕ Б. Листинг Структура базы данных 38](#_Toc153981804)

[ПРИЛОЖЕНИЕ В. Листинг Диаграмма компонентов 39](#_Toc153981805)

# Введение

В современном информационном обществе использование мобильных приложений становится все более распространенным и неотъемлемым элементом нашей повседневной жизни. Одной из самых популярных категорий мобильных приложений являются мессенджеры, которые позволяют нам общаться и обмениваться сообщениями с другими людьми в режиме реального времени.

Мобильные приложения для обмена сообщениями, такие как WhatsApp, Telegram, Facebook Messenger, WeChat и многие другие, предоставляют возможность мгновенного общения с друзьями, семьей и коллегами, независимо от расстояния и времени. Они стали незаменимым инструментом для коммуникации, как в личных, так и в деловых целях.

Исследование и разработка мобильного приложения для обмена сообщениями представляет собой актуальную задачу, которая имеет большое значение в современном информационном обществе. Оно позволяет нам более эффективно общаться, сокращая расстояние и время, и создает новые возможности для социальной и деловой коммуникации.

Целью данной курсовой работы является изучение и разработка мобильного приложения для обмена сообщениями. Мы будем исследовать различные аспекты такого приложения, включая функциональные возможности, архитектуру, пользовательский интерфейс и производительность.

В заключение курсовой работы мы представим разработанное мобильное приложение для обмена сообщениями и оценим его функциональность, удобство использования и эффективность.

# 1 Обзор аналогов приложения

## 1.1 Аналог «WhatsApp»

На рынке существует множество мобильных приложений для обмена сообщениями, предлагающих различные функциональные возможности, уникальные особенности и уровень безопасности. Примерами таких приложений являются WhatsApp, Telegram и Twitter.

Первым рассматриваемым проектом среди конкурентов является «WhatsApp». WhatsApp – мессенджер номер 1 в мире, который завоевал сердца более миллиарда людей по всему миру. Его преимущества заключаются в том, что он абсолютно бесплатный, имеет простой интерфейс и обладает невероятно высокой скоростью передачи данных, и даже при слабом интернете.

Макет мобильного приложения представлен на рисунке 1.1.

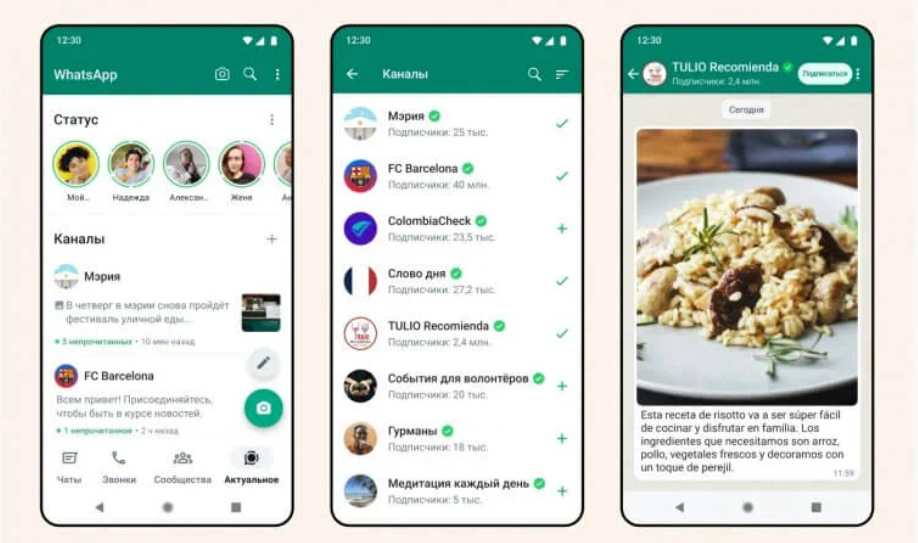


Рисунок 1.1 – Макет приложения «WhatsApp»

Приложение дает нам возможность общаться со своими близкими или знакомыми в приватных или открытых чатах, а количество людей в них, может достигать до 250 пользователей. Обмениваясь голосовыми сообщениями, видео контентом или просто текстовыми переписками, все наши данные проходят сквозное шифрование и не могут быть доступны третьим лицам, в том числе и мошенникам.

В данном приложении есть возможность скрытого чтения сообщений. Отправитель не узнает, прочитали ли вы его сообщение, — у него будет видна только одна серая галочка. Так же в данном приложении можно отправлять исчезающее сообщение. Когда адресат прочитает сообщение, оно исчезнет из переписки для обоих собеседников.

Интересной особенностью данного приложения является возможность отформатировать текст сообщения. Мессенджер позволяет выделить текст жирным, курсивом или зачеркнуть его. На рисунке 1.2 показаны примеры отформатированных сообщений.

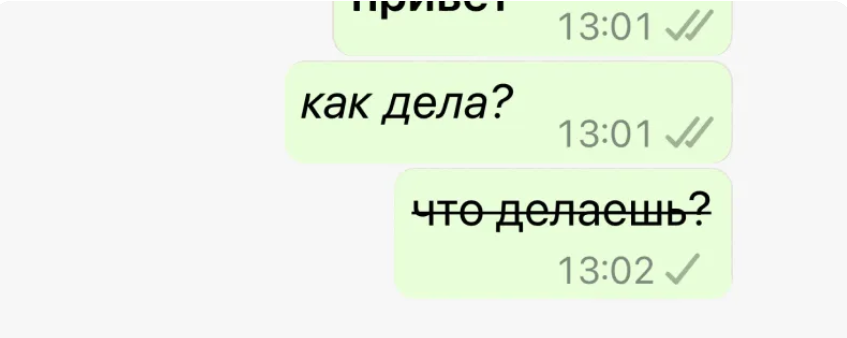


Рисунок 1.2 – Отформатированные сообщения

Так же у данного приложения есть ряд недостатков. Во-первых, войти в учётную запись мессенджера можно только с одного смартфона. Во-вторых, нельзя подтвердить вход с компьютера один раз. Каждый раз при активации мессенджера с десктопа необходимо подтверждать авторизацию с помощью смартфона, наведя камеру на QR-код. В-третьих, когда вы входите в аккаунт с другого устройства, по умолчанию ваши переписки не восстанавливаются.

В целом приложение имеет довольно понятный пользователю интерфейс и удобно для использования.

## 1.2 Аналог «Telegram»

Среди конкурентов рассматривается второй проект - «Telegram». Telegram является одним из ведущих мессенджеров на сегодняшний день, обладающим огромной популярностью и широким спектром функций. Он предоставляет пользователям удобный и безопасный способ общения, а также обмена файлами и мультимедийными сообщениями.

Одной из ключевых особенностей Telegram является возможность обмена аудио- и видео-сообщениями. Пользователи могут записывать голосовые сообщения или снимать видео и отправлять их другим пользователям в режиме реального времени. Это особенно удобно, когда нужно быстро и эффективно передать информацию или поделиться впечатлениями.

Важно отметить, что Telegram имеет простой и интуитивно понятный пользовательский интерфейс, что делает его доступным для широкого круга пользователей. Мессенджер доступен на различных платформах, включая мобильные устройства и компьютеры, и позволяет синхронизировать чаты и данные между ними.

Макет мобильного приложения представлен на рисунке 1.3.

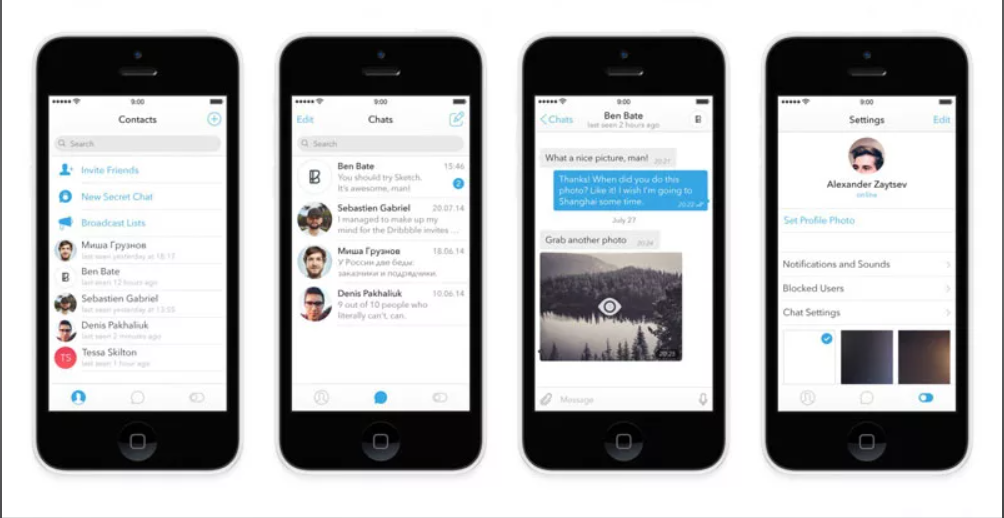


Рисунок 1.3 – Макет приложения «Telegram»

Если говорить о преимуществах, то в первую очередь стоит отметить конфиденциальность данных. Таким образом, кому-либо будет достаточно трудно получить информацию о конкретном человеке через представителей команды Telegram.

Важной возможностью данного приложения является блокировка приложения паролем. Для этого можно установить код, без ввода которого нельзя пользоваться мессенджером и просматривать историю чатов. Страница для представления ввода пароля представлена на рисунке 1.4.



Рисунок 1.4 – Страница ввода пароля

Так же Telegram реализовал функционал голосовых чатов и с тех пор постоянно улучшает эту опцию. Голосовой чат доступен для администраторов любых групп и каналов. К чату могут присоединиться любые желающие, пока он активен. Голосовой чат также можно запланировать на определенное время, в этом случае участники группы получат уведомление, когда чат начнется. Пример данного чата представлен на рисунке 1.5.

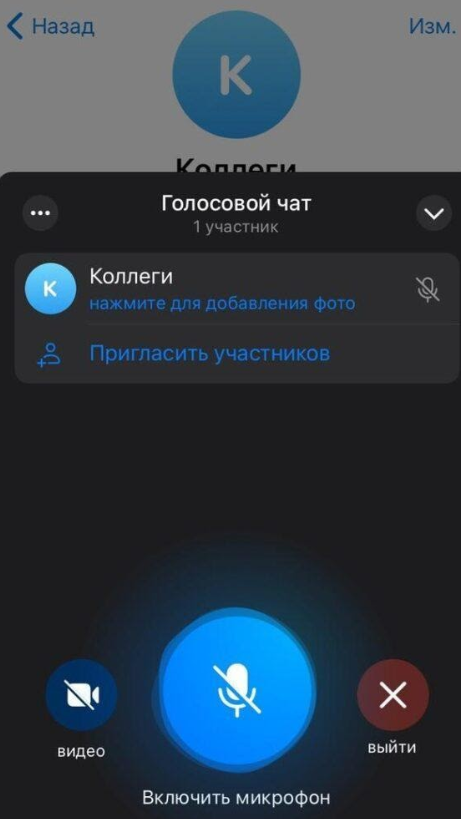


Рисунок 1.5 – Голосовой чат

Кроме того, Telegram предлагает возможность создания каналов, которые являются публичными страницами для публикации и получения информации. Каналы могут быть использованы для распространения новостей, обновлений, полезных материалов или просто для общения с аудиторией.

Суммируя все вышесказанное, Telegram представляет собой мощный и многофункциональный мессенджер, который удовлетворяет потребности пользователей в обмене сообщениями, передаче файлов, голосовых и видеозвонках, создании Telegram является одним из лидирующих мессенджеров в настоящее время, пользующимся широкой популярностью и предоставляющим множество функций. Он обеспечивает удобное и безопасное общение, а также обмен файлами и мультимедийными сообщениями.

**1.3** **Аналог «KakaoTalk»**

Третим рассматриваемым проектом среди конкурентов является «KakaoTalk». KakaoTalk – это многофункциональное приложение мессенджера, которое предоставляет пользователям удобный способ общения и обмена информацией. Оно разработано компанией Kakao Corporation и является одним из самых популярных мессенджеров в Южной Корее и ряде других стран.

Макет мобильного приложения представлен на рисунке 1.6.

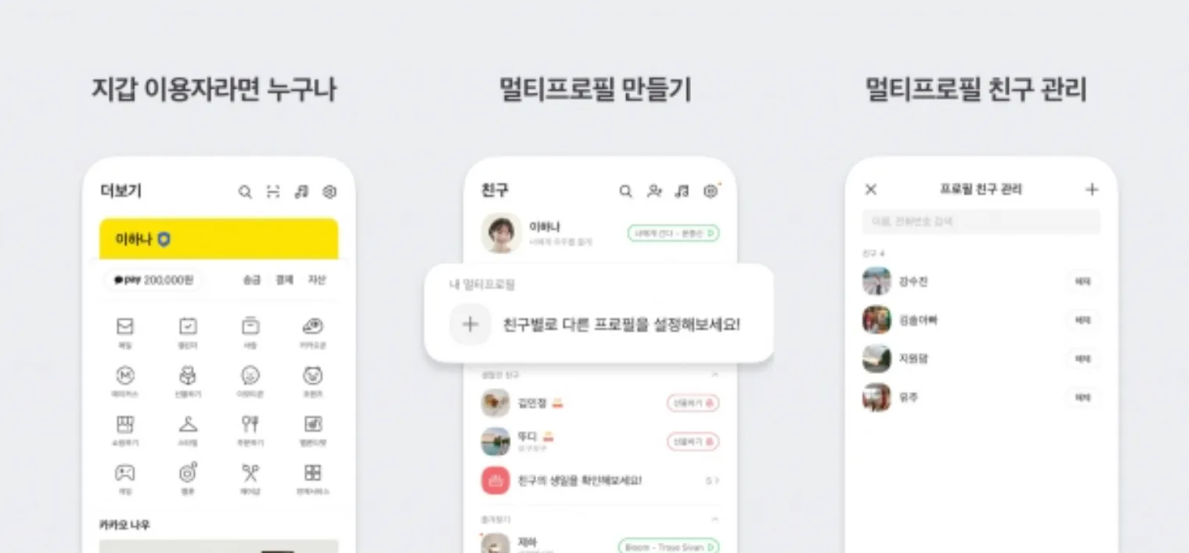


Рисунок 1.6 – KakaoTalk

KakaoTalk, появившись на мобильных прилавках в марте 2010-ого, оказался первым мессанджером, разработанным корейской командой. Главными особенностями продукта решили сделать: высокую лояльность к пользователю, прямую социализацию и наличие игр — главного развлекательного элемента.

Одной из уникальных особенностей KakaoTalk является наличие большого количества стикеров и эмодзи, которые пользователи могут использовать в своих сообщениях. Стикеры добавляют эмоциональность и экспрессивность к сообщениям, позволяя выразить свои чувства и настроение визуально.

Кроме того, KakaoTalk предлагает дополнительные функции, такие как расширения и приложения-партнеры. Расширения позволяют пользователям расширить возможности приложения, добавляя функции, такие как расписание, погода, игры и многое другое. Приложения-партнеры предоставляют доступ к различным сервисам и контенту, таким как онлайн-магазины, новостные ресурсы, музыкальные платформы и прочее.

Данное приложение имеет расширение Kakao Mail. Kakao Mail дает возможность получить доступ к электронной почте. Удобнее обмениваться электронными письмами с Kakao Mail, где вы можете получать уведомления по почте. Также в данном приложение можно менять темы интерфейса.

Примеры темы представлены на рисунке 1.7.

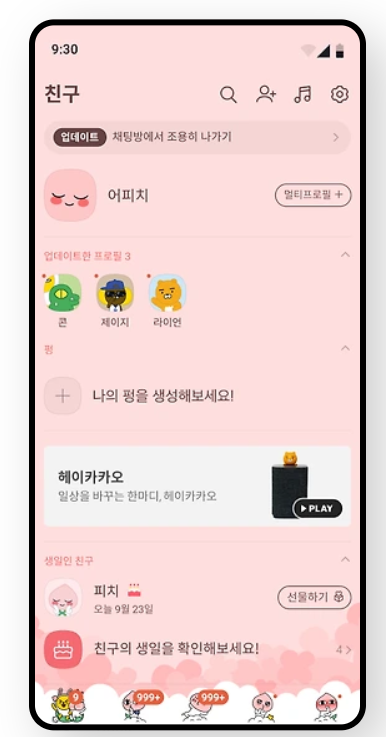


Рисунок 1.7 – KakaoTalk тема

KakaoTalk - это универсальное приложение мессенджера, которое объединяет удобство текстового общения, обмен мультимедийными файлами, голосовыми и видеозвонками, а также разнообразными дополнительными функциями. Оно предоставляет пользователям возможность поддерживать связь с друзьями, семьей и коллегами, делиться важной информацией и наслаждаться интерактивным общением в удобной и безопасной среде.

## 1.4 Выводы по главе «Обзор аналогов приложения»

В данной главе были представлены три аналога приложений мессенджеров, которые служат хорошими примерами для разрабатываемого в ходе курсового проекта приложения.

Каждое из этих приложений имеет свои преимущества в зависимости от потребностей и предпочтений пользователя. WhatsApp отличается широким охватом и простотой использования, Telegram фокусируется на безопасности и расширенных функциях, а KakaoTalk является популярным выбором в Южной Корее с множеством дополнительных возможностей.

# 2 Проектирование приложения

## 2.1 Диаграмма вариантов использования

Диаграмма вариантов использования (Use Case Diagram) служит для визуализации функциональности системы с точки зрения ее пользователей. В рамках данного приложения выделяются две ключевые роли: администратор и пользователь, каждая из которых обладает своими характерными функциями. Рассмотрим каждую роль по отдельности.

Администратор может имеет одну важную возможность. А именно возможность управление базой данных, которая в свою очередь представляет собой просмотр информации о пользователях, добавление пользователей, удаление пользователей и изменение пользователей.

Диаграмма вариантов использования для администратора представлена на рисунке 2.1.

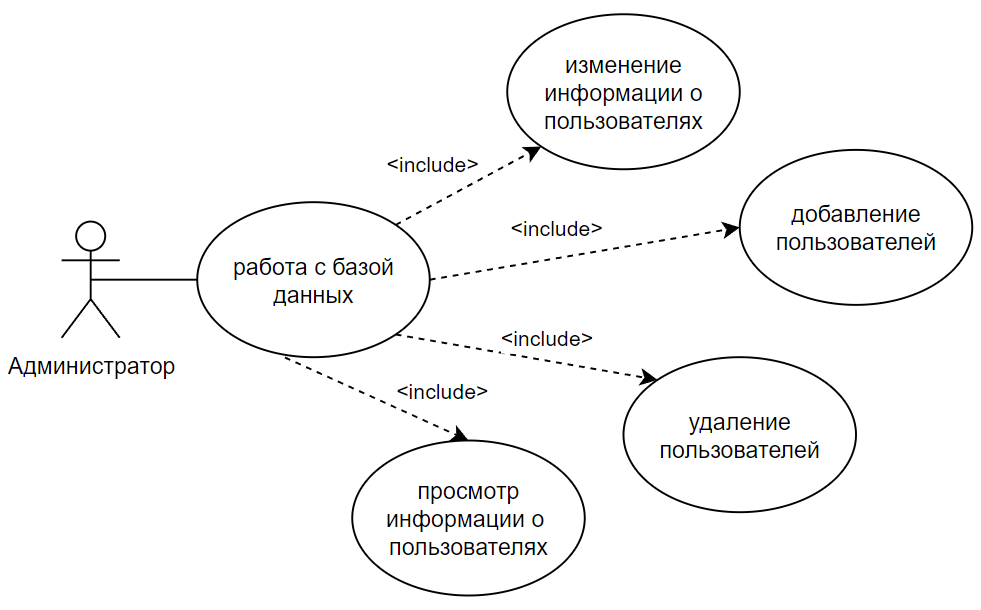


Рисунок 2.1 – Диаграмма вариантов использования (администратор)

Пользователь приложения обладает более широким спектром функций. Он может управлять своим аккаунтом: создавать, изменять и удалять. Также пользователь имеет возможность производить поиск друзей и отправлять им сообщения, создавать групповые чаты и отправлять в них сообщения. И также пользователь может регистрироваться и авторизовываться.

Диаграмма вариантов использования для пользователя представлена на рисунке 2.2.

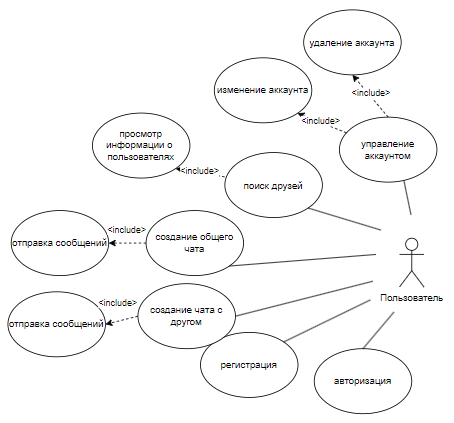


Рисунок 2.2 – Диаграмма вариантов использования (пользователь)

Таким образом с помощью двух диаграмм использования для роли администратора и для роли пользователя был определен весь необходимый доступный функционал.

## 2.2 Архитектура приложения

Архитектура приложения представляет собой само приложение базу данных. Облачная платформа Firebase будет использоваться как база данных.

Для разделения логики приложение отдельные функции для работы с чатом, групповым чатом, регистрации и авторизации были реализованы отдельных классах. На рисунке 2.3 представлена схема архитектуры.

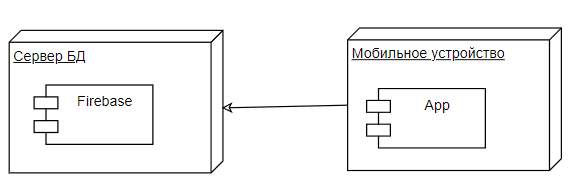


Рисунок 2.3 – Архитектура приложения (диаграмма компонентов)

Представленная архитектура позволит реализовать весь определенный ранее функционал приложения.

## 2.3 Проектирование базы данных

Для разработки мобильного приложения в рамках курсового проекта понадобилась база данных с необходимой конфигурацией сущностей для хранения всей необходимой информации.

Для реализации базы данных была выбрана серверная база данных Firebase. Firebase – это облачная платформа разработки приложений, предоставляемая компанией Google. Она предлагает разработчикам комплексный набор инструментов и сервисов для создания и управления приложениями.

Структуру базу данных, ограничения целостности, связи и поля представлены в приложении А и на рисунке 2.5.

Созданная база данных содержит в себе 4 коллекций: GroupChats, Group\_messages, Users и Messages.

В состав коллекции GroupChats входят следующие столбцы:

* Поле groupName. Хранит имя группового чата.
* Поле users. Хранит список пользователей чата.
* Поле createdAt. Хранит дату и время создания чата.

Данная коллекция используется для хранения информации о групповых чатах. Все поля коллекции должны содержать какое-либо значение.

В состав коллекции Group\_messages входят следующие столбцы:

* Поле content. Хранит текст сообщения.
* Поле groupName. Хранит имя чата, в которому принадлежит сообщение.
* Поле senderEmail. Хранит емаил пользователя, который отправил сообщение.
* Поле createdAt. Хранит дату и время создания и отправки сообщения.

Данная коллекция ипользуется для хранениея сообшений для конкретного чата.

В состав коллекции Users входят следующие столбцы:

* Поле uid. Хранит уникальный идентификатор.
* Поле fullname. Хранит имя пользователя.
* Поле password. Хранит пароль пользователя.
* Поле age. Хранит возраст пользователя.
* Поле country. Хранит название страны пользователя.
* Поле email. Хранит емаил пользователя.
* Поле imageURL. Хранит url изображения пользователя.

Данная коллекция ипользуется для хранениея информации о пользователях.

В состав коллекции Messages входят следующие столбцы:

* Поле message. Хранит текст сообщения.
* Поле receiverID. Хранит имя чата, в которому принадлежит сообщение.
* Поле senderEmail. Хранит емаил пользователя, который отправил сообщение.
* Поле senderID. Хранит дату и время создания и отправки сообщения.
* Поле timestamp. Хранит дату и время создания и отправки сообщения.

Данная коллекция ипользуется для хранениея информации о пользователях.

Таким образом были рассмотрены все сущности базы данных, используемые в разрабатываемом приложении.

## 2.4 Выводы по главе «Проектирование приложения»

В данном разделе произведено проектирование функциональности приложения с использованием диаграмм вариантов использования (Use Case Diagrams). Выделены две ключевые роли - администратор и пользователь, каждая из которых обладает определенными функциональностями. Были определены сущность БД и архитектура приложения.

# 3 Реализация приложения

Далее рассмотрим разработку базы данных и приложения мобильного.

## 3.1 Разработка базы данных

Для создания базы данных было необходимо установить все необходимые библиотеки и создать проект в firebase. На рисунке 3.1 представлен проект для базы данных.

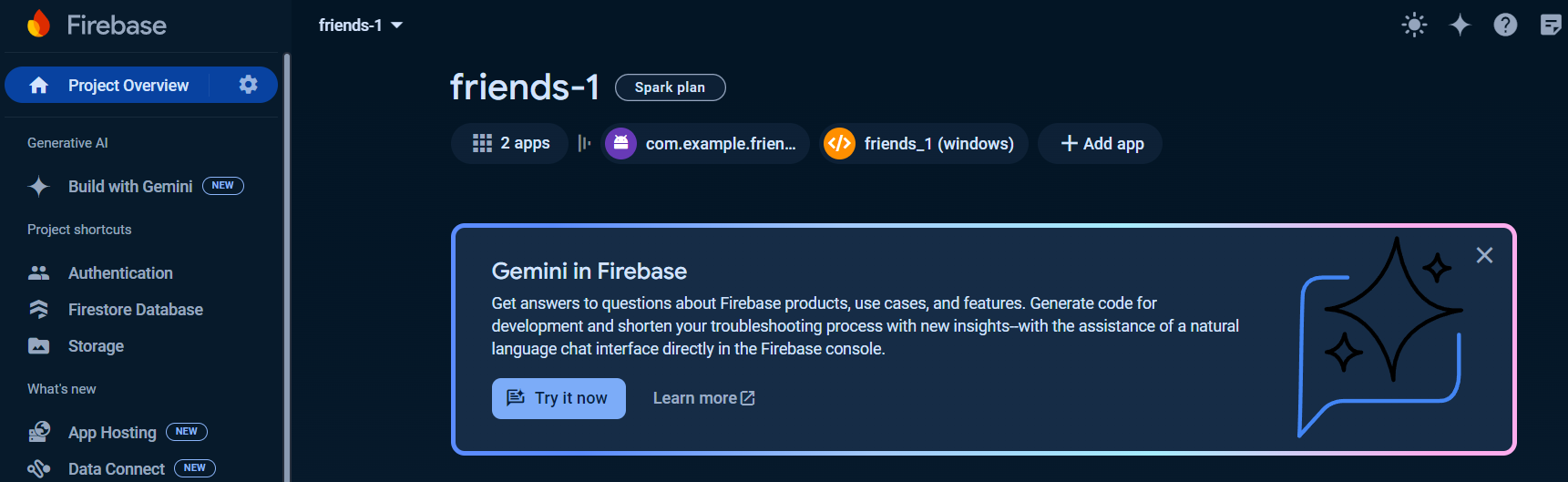


Рисунок 3.1 – Проект для базы данных

Далее было необходимо настроить Authentication и создать пользователей через почту электронную и пароль для проверки авторизации. На рисунке 3.2. представлен список пользователей.

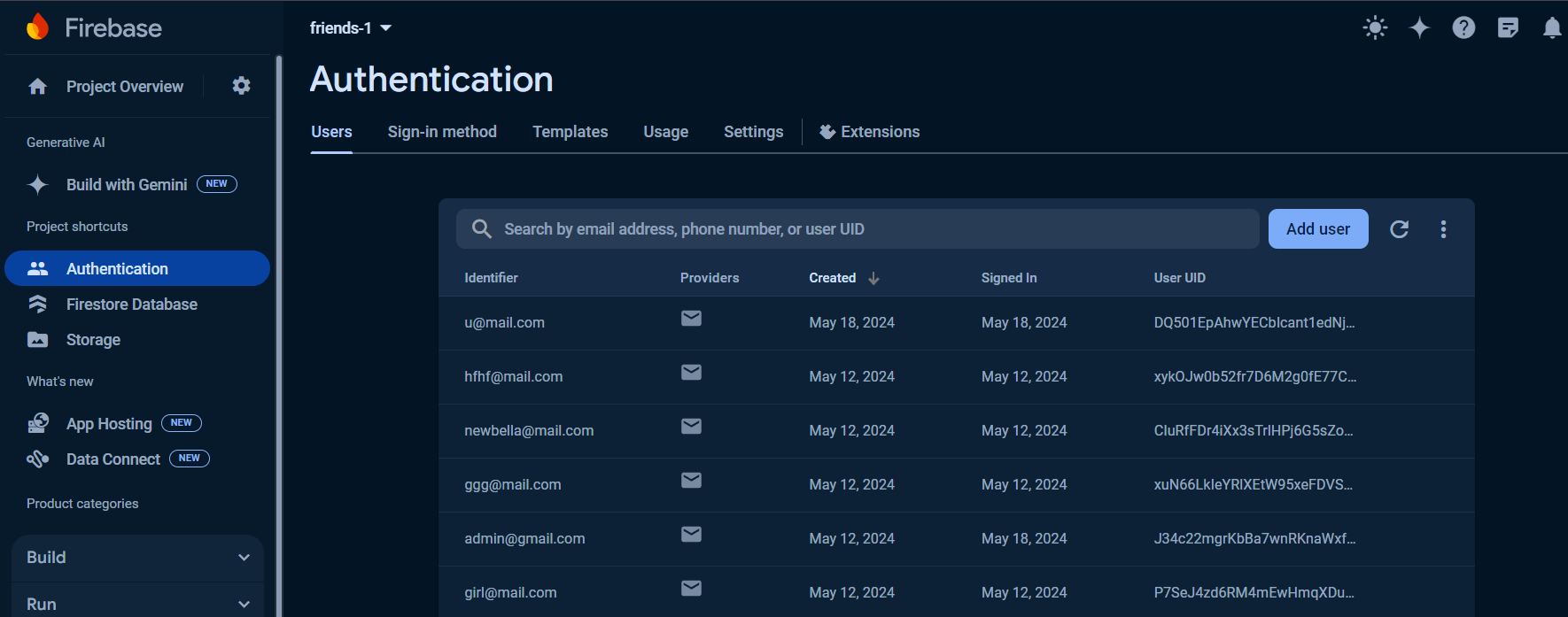


Рисунок 3.2 – Страница Authentication

Далее было необходимо настроить Cloud Firestore и создать коллекции для последующего использования. На рисунке 3.3. представлен список коллекций.

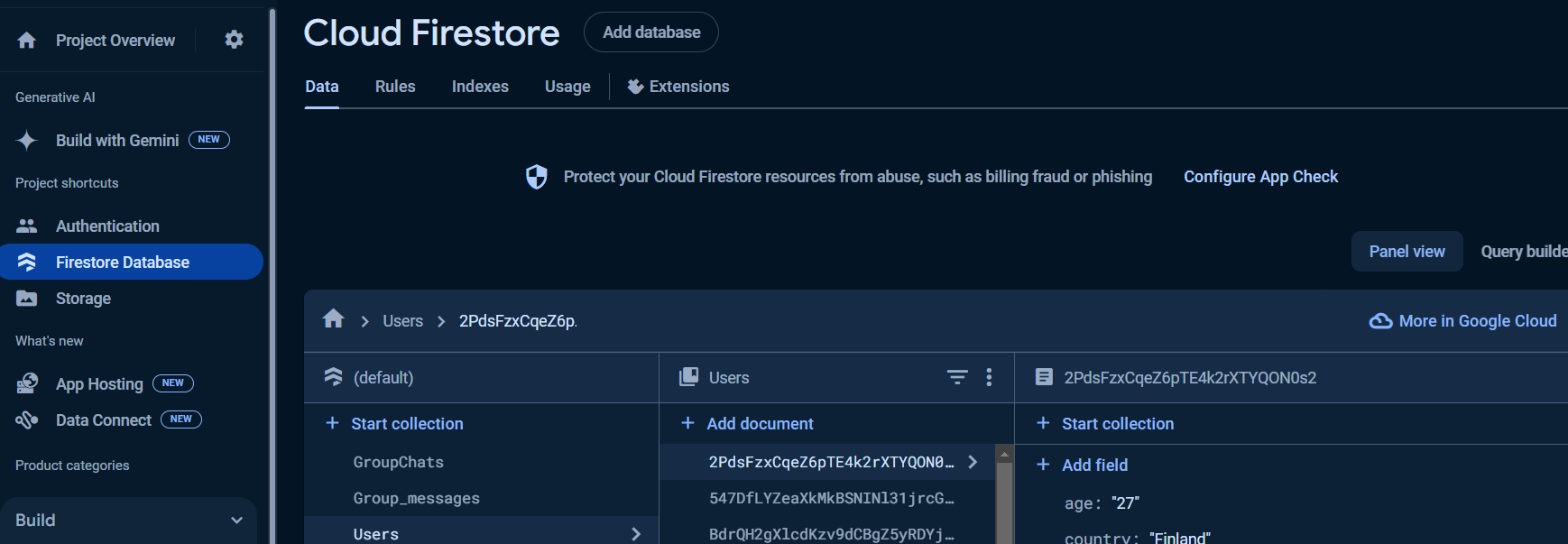


Рисунок 3.3 – Страница Cloud Firestore

На данном этапе было настроено FireBase, Authentication и Cloud Firestore для дальнейшего использования.

### 3.2 Подключение базы данных

Для работы с базой данных в внутри проекта нужно было добавить следующи код в файл main.dart.

|  |
| --- |
| Future<void> main() async {  WidgetsFlutterBinding.ensureInitialized();  await Firebase.initializeApp( );  runApp(const MyApp());  } |

Листинг 3.1 – Добавленный код

Перед добавления вышенаписанного кода в файл pubspec.yaml. Здесь прописано необходимые библиотеки для работы с Firebase, использования коллекций и использования аутентификации.

|  |
| --- |
| firebase\_core: ^2.30.1  firebase\_auth: ^4.19.4  cloud\_firestore: ^4.17.2 |

Листинг 3.2 – Библиотеки в файле pubspec.yaml

При помощи данных библиотек и кода, написанного выше, была подключения база данных Firebase для разрабатываемого приложения.

## 3.3 Разработка мобильного приложения

Клиентское приложение разрабатывается на языке программирования Dart с использованием фреймворка Flutter и дополнительных библиотек, опирается на архитектуру приложения.

На рисунке 3.4 представлена структура проекта.

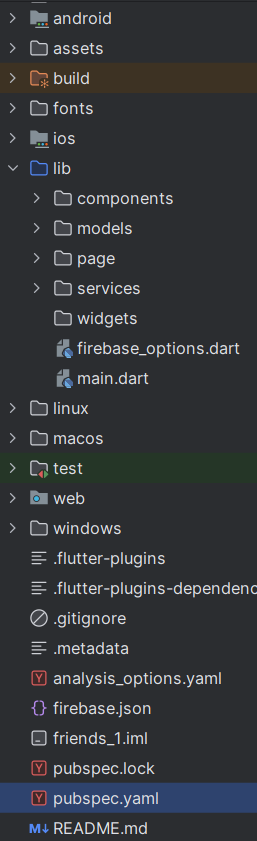


Рисунок 3.4 – Файловая структура приложения

Рассмотрим структуру приложения подробнее. В директории «components» представлены следующие компоненты приложения:

* ChatBubble – компонент для реализации сообщение;
* UserAdminTile – компонент для элемента списка друзей на странице администратора;
* UserTile – компонент для элемента списка друзей;

Директория «models» хранит в себе классы для моделей таблиц для коллекций в FireStore. Данная директория содержит следующие модели: модуль для сообщения (Message), модуль для пользователя (Users), модуль для сообщения в групповом чате (GroupMessage).

### 3.4 Разработка пользовательского интерфейса

Разметка всех экранов приложения создается с помощью языка программирования Dart. Во фреймворке Flutter каждый экран и каждый элемент пользователского интерфейса является так называемым виджетом (Widget). При этом каждый виджет может как хранить состояние так и нет.

В листинге 3.5 представлен фрагмент разработанного интерфейса для экрана авторизации.

|  |
| --- |
| @override  Widget build(BuildContext context) {  return Scaffold(  body: Container(  child:SingleChildScrollView(  child: Column(  children: [  SizedBox(  height: 40,  ),  Row(  mainAxisAlignment: MainAxisAlignment.start,  children: [  SizedBox(width: 20,),  FloatingActionButton(  onPressed: () {  Navigator.pop(context);  },  backgroundColor: Colors.transparent,  elevation: 0,  mini: true,  heroTag: null,  child: Icon(Icons.arrow\_back,  size: 35,  color: Colors.black38,),  ),]  ), |

Листинг 3.5 – Виджет экрана авторизации

Все остальные элементы приложения разработаны подобным образом: целый экран, являющийся виджетом, состоит из виджетов поменьше. Для отрисовки конкретного виджета вызывается конструктор с параметрами, описывающими свойства данного элемента. Таким образом можно изменять размер, форму, цвет любых элементов и компоновать интерфейс с помощью специальных виджетов-контейнеров.

## 3.5 Выводы по главе «Реализация приложения»

В данной главе мы подробно исследовали процесс создания различных компонентов нашего приложения

Базы данных были успешно созданы с использованием FireBase. Это обеспечивает удобный и высокоуровневый способ взаимодействия с базой данных.

В результате весь заранее спроектированный функционал был успешно реализован, и приложение готово к следующим этапам тестирования и доработки.

Кроме того, были уделены усилия созданию дружественного и интуитивно понятного пользовательского интерфейса, что способствует приятному взаимодействию пользователя с приложением. Важным аспектом является также поддержка мобильных устройств с использованием технологий Flutter, что расширяет охват аудитории и обеспечивает удобство использования в различных сценариях.

# 4 Тестирование приложения

Тестирование программного обеспечения играет ключевую роль в обеспечении высокого качества программных продуктов. Оно направлено на проверку соответствия фактических результатов работы программы ожидаемым. Главная цель тестирования - убедиться, что программа соответствует заявленным требованиям, выявить и исправить ошибки до того, как они окажутся замечены конечными пользователями.

Для проверки работоспособности приложение были выполнены вручную ряд тестов.

Для список тестируемых элементов прелдставлен в таблице 4.1 список тестов.

Таблица 4.1 – Список тестов

|  |  |  |
| --- | --- | --- |
| Описание теста | Ожидаемый результат | Статус |
| 1 | 2 | 3 |
| Вход в учетную запись | Перенаправление на домашнюю страницу | Успешно |
| Создание нового группового чата | Создание чата и преход на страницу чата | Успешно |
| Добавление нового пользователя | Создание аккаунта | Успешно |
| Изменение существующего пользователя | Изменение информации о пользователе | Успешно |
| Удаление пользователя | Удаление пользователя и переход на домашную страницу | Успешно |
| Отправление сообщение другу | Отображение отправленного сообщения на странице чата | Успешно |
| Отправление сообщение в групповой чат | Отображение отправленного сообщения на странице группового чата | Успешно |
| Поиск друзей | Отображение друга | Успешно |
| Выход из учетной записи | Перенаправление на страницу входа | Успешно |

## 4.1 Тестирование ограничения доступа

В данном подразделе была поставлена задача в проверки выполонения валидации. Были произведены ряд проверок приложения на выполнения валидации.

Например, была произведена проверка на попытку входа без заполнения требуемых полей. На рисунке 4.1 представлена обработка данной ситуации в приложения. Были не заполнены следующие поле: электронная почта пользователя и пароль пользователя. При попытка входа было показано два сообщение под полями ввода.



Рисунок 4.1 – Страница для авторизации пользователя

Также была произведена попытка входа с неправильными входными данными. На рисунке 4.2 представлена обработка такой ситуации с помощью отправки сообщение о неправильном вводе данных пользователем.

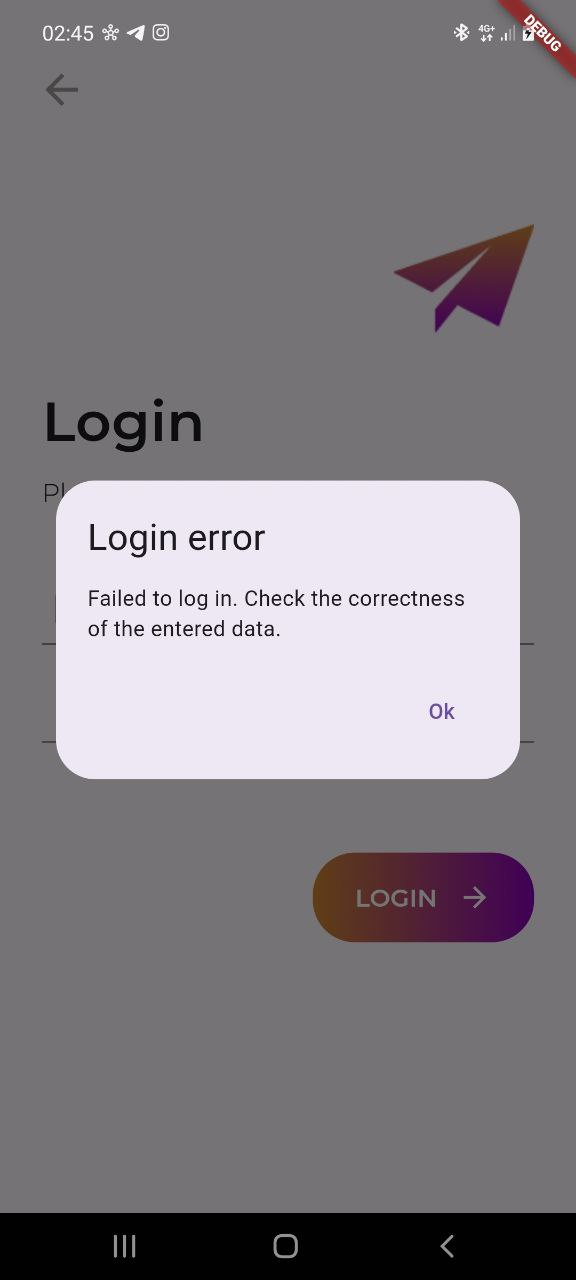
 

Рисунок 4.2 – Страница авторизации

## 4.2 Тестирование интерфейса

Протестируем адаптивность мобильного приложения. Для этого сравним то, как выглядит приложение при различных ориентациях экрана.

На рисунке 4.3 представлен страница чата сдругом в вертикальной ориентации.



Рисунок 4.3 – Страница чата с другом в вертикальной ориентации

На рисунке 4.4 представлен страница чата сдругом в горизонтальной ориентации.

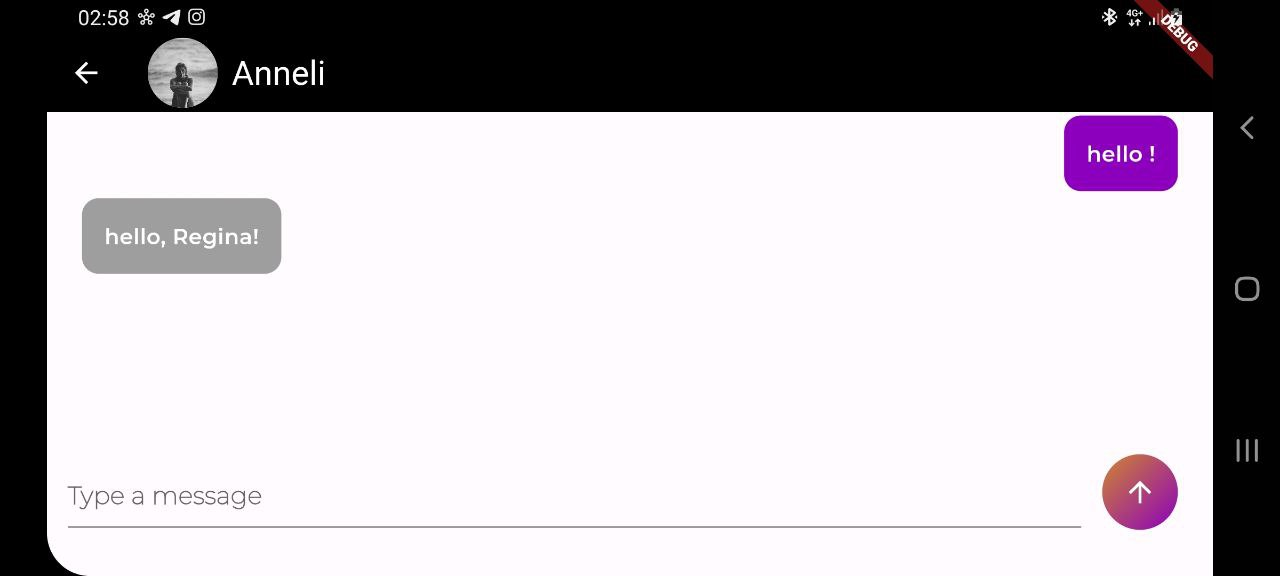


Рисунок 4.4– Страница чата с другом в горизонтальной ориентации

Стоит отметить, что при изменении ориентации приложение не изменяет или обновляет свое состояние. Все элементы интерфейса адаптивны и подстраиваются под изменение параметров экрана.

На представленном выше рисунке видно, что интерфейс также прекрасно адаптируется и в горизонтальной ориентации устройства.

## 4.3 Выводы по главе «Тестирование приложения»

В данной главе было проведено тестирование компонентов приложения. Все тесты были пройдены успешно. Также было протестировано разграничение функционала приложения по ролям.

После тестирования мобильного приложения можно сделать вывод о том, что приложение сохраняет свое состояние независимо от ориентации экрана. Пользовательский интерфейс очень точно адаптирован для устройств с разной диагональю экрана.

# 5 Руководство пользователя

В данном разделе рассмотрим, как использовать мобильное приложение.

## 5.1 Мобильное приложение

При запуске приложение пользователю отображается страница входа в приложение. На ней показаны два способа входа: регистрация и авторизация. Данная страница представлена на рисунке 5.1.

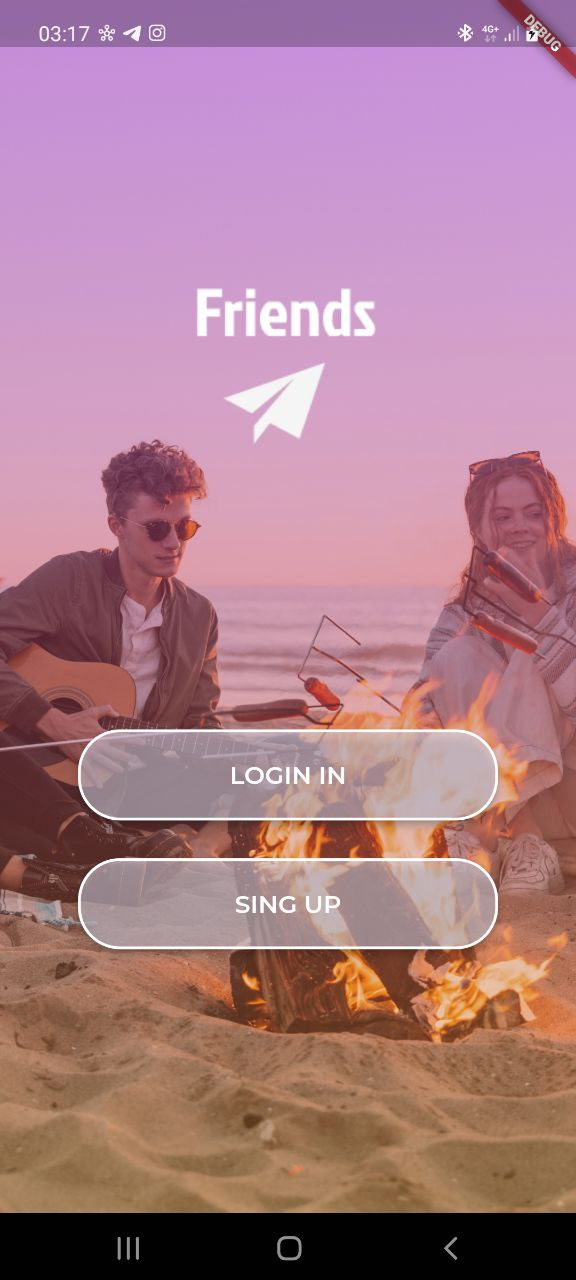


Рисунок 5.1 – Начальная страница

Если учетной записи пользователя нет, то пользователю следует перейти при помощи кнопки «Sing up» на страницы регистрации пользователя. На странице регистрации представлены поля для ввода информации о пользователе. Пользователь может выбрать фото из галереи телефоны и загрузить её в приложения, после чего это изображение будет использоваться как изображение пользователя в аккаунте. На рисунке 5.2 представлена страница для регистрации пользователя и страница регистрации с заполненными полями и выбранным изображением.

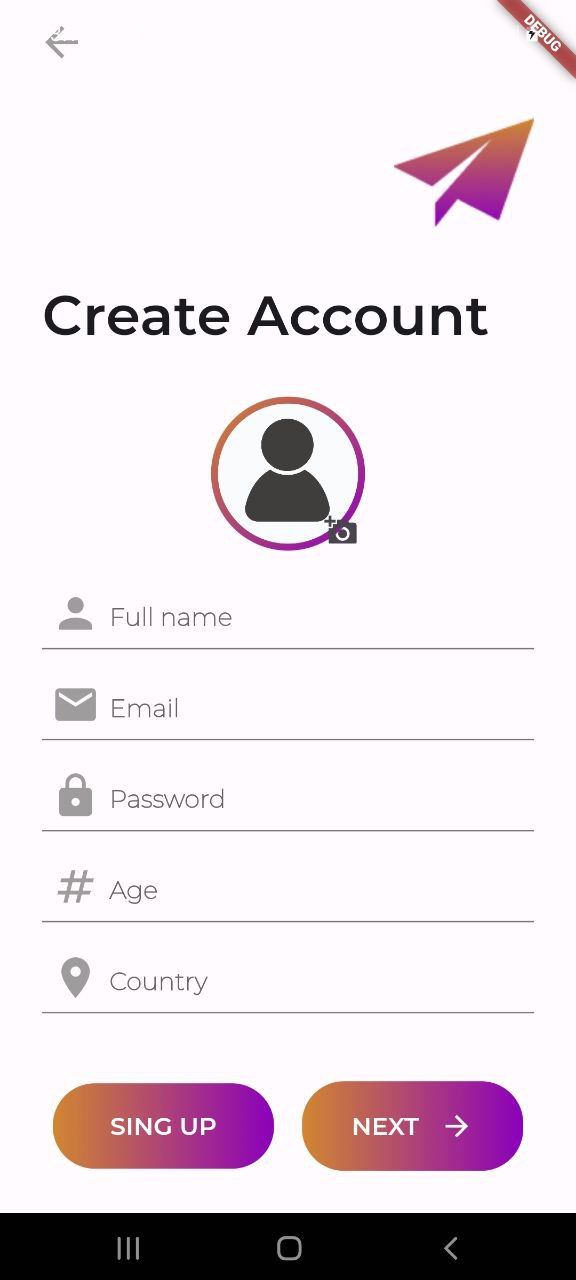
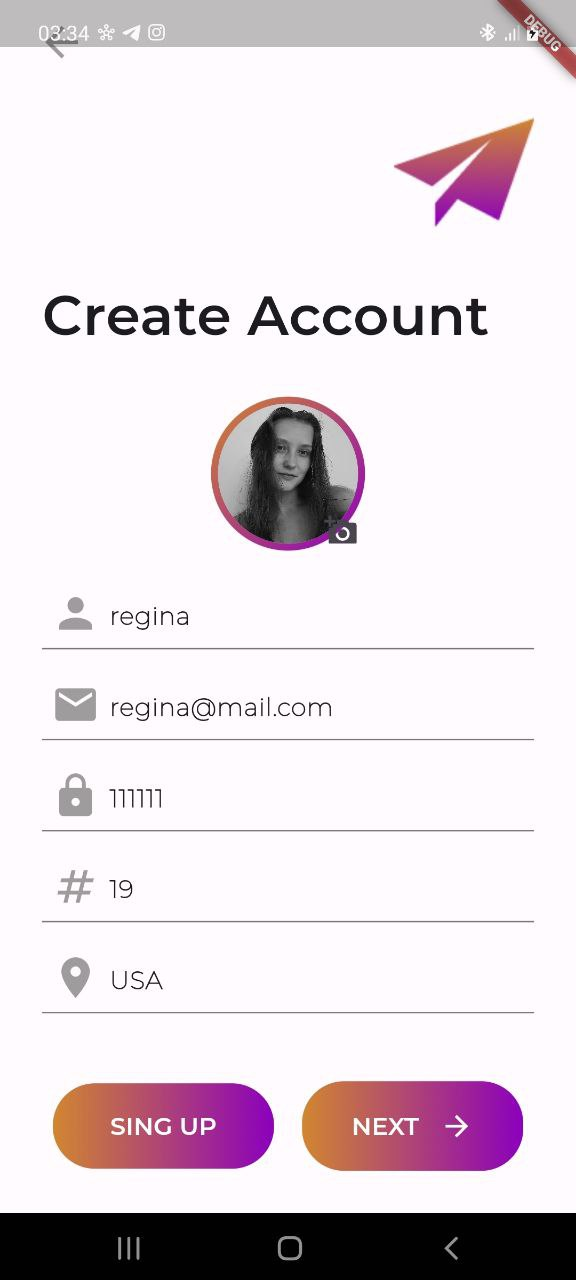
 

Рисунок 5.2 – Страница регистрации

Если все поля заполнены и выбрано изображение, то пользователь необходимо нажать на кнопку «Sing up», после чего появится сообщение о успешном создании пользователя. Данное сообщение представлено на рисунке 5.3.

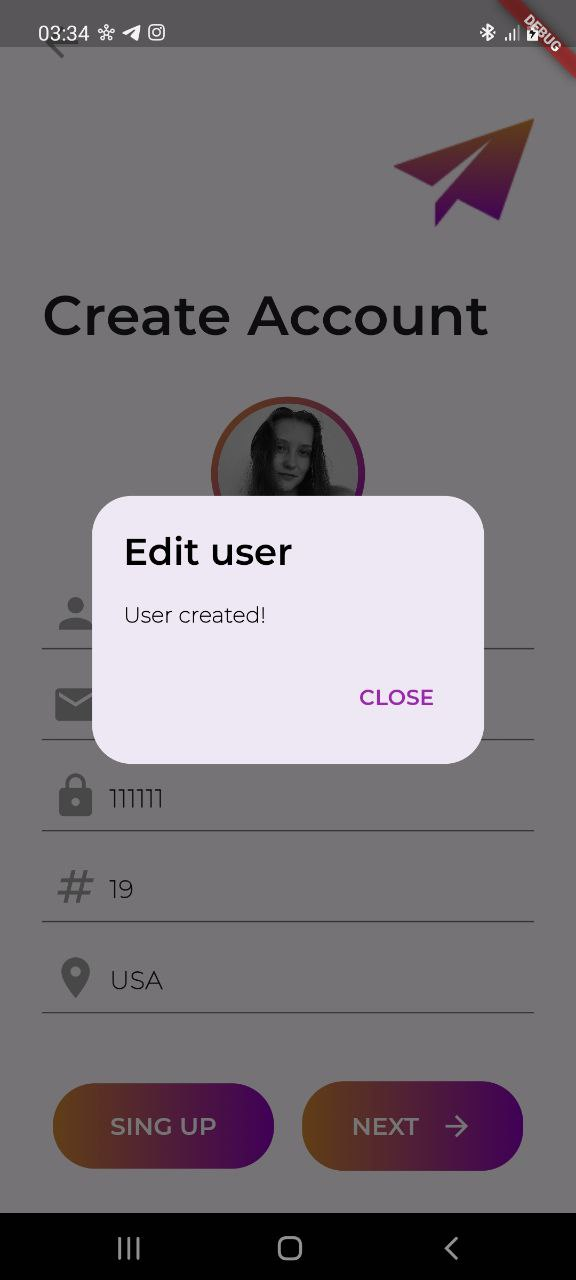


Рисунок 5.3 – Сообщение о успешном создании пользователя

После пользователю необходимо нажать на кнопку «Next», после чего будет открыта главная страница приложения. Данная страница представлена на рисунке 5.4.



Рисунок 5.4 – Главная страница

При нажатии на бургер меню будет отображать меню приложение. Меню представлено на рисунке 5.5.

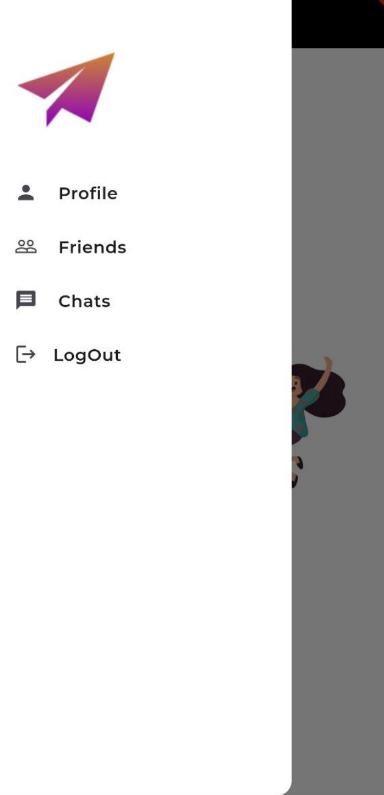


Рисунок 5.5 – Меню

При нажатии на элемент списка «Profile» пользователь переходит на страницу профиля пользователя. На данной странице отображается информация о пользователе. Страница представлена на рисунке 5.6.

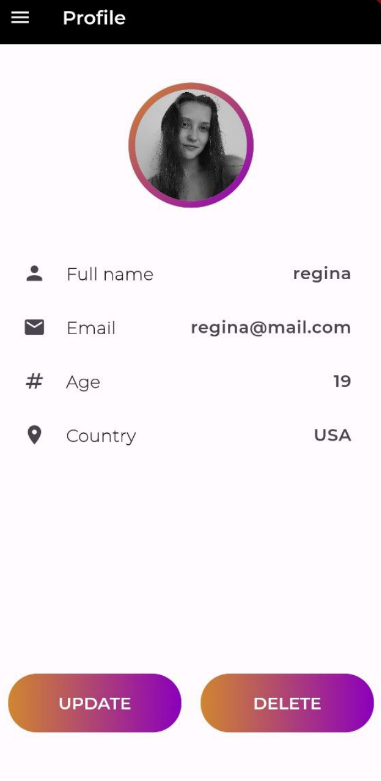


Рисунок 5.6 – Страница профиля

При нажатии на кнопки «Update» пользователь может изменить информацию о себе. Для этого после нажатия появляется окно ввода с полями для изменения информации и для загрузки фотографии. Окно представлено на рисунке 5.7.

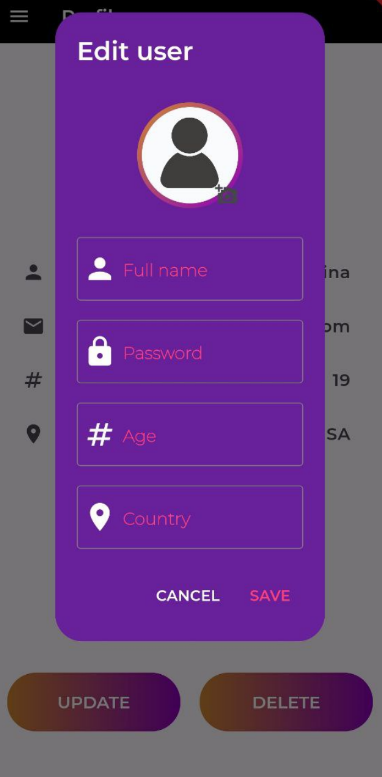


Рисунок 5.7 – Изменение пользователя

Если пользователь хочет удалить свой аккаунт, то ему необходима нажать на кнопку «Delete». При нажатии пользователю открывается окно предупреждения перед удалением аккаунта. Для продолжения процесса удаления пользователю нужно нажать на «Ок». Предупреждение представлено на рисунке 5.8.

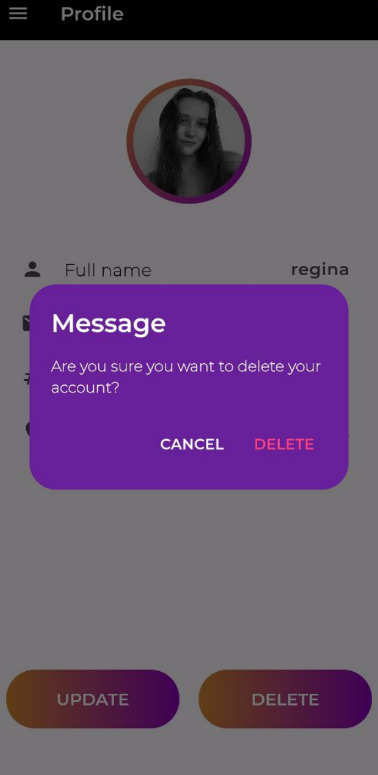


Рисунок 5.8 – Предупреждение об удалении

При нажатии на элемент списка «Friends» пользователь переходит на страницу со списком пользователей. На данной странице можно производить поиск друга, также можно производить переход на страницу чата с выбранным пользователем и н на страницу с информацией о пользователе. Страница все представлена на рисунке 5.9.

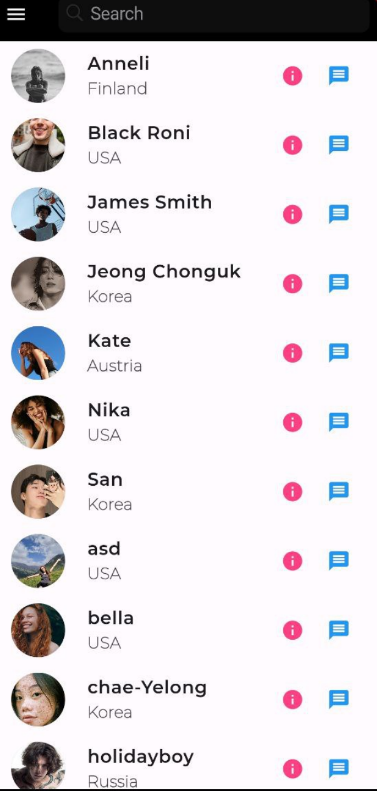


Рисунок 5.9 – Список друзей

На рисунке 5.10 представлен результат поиска друга по его имени.



Рисунок 5.9 – Поиск

На рисунке 5.11 представлена страница с информацией о друге.

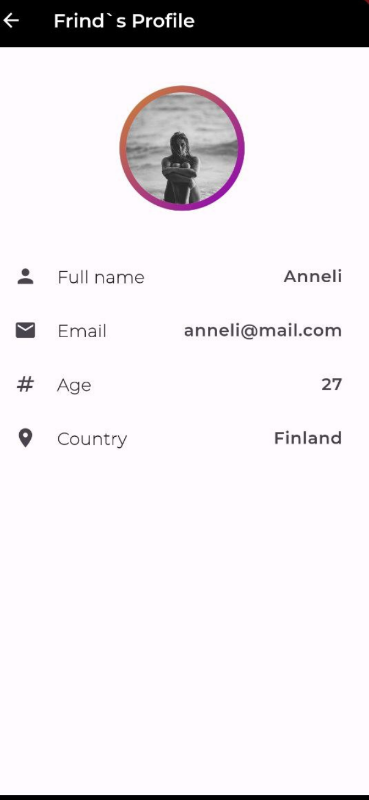


Рисунок 5.11 – Страница с информацией о друге

На рисунке 5.12 представлена страница с чатом.



Рисунок 5.12 – Страница чата с друге

При нажатии на элемент списка «Chats» пользователь переходит на страницу со списком групповых чатов. На данной странице можно также создать чат. Страница со списком чатов представлена на рисунке 5.13.

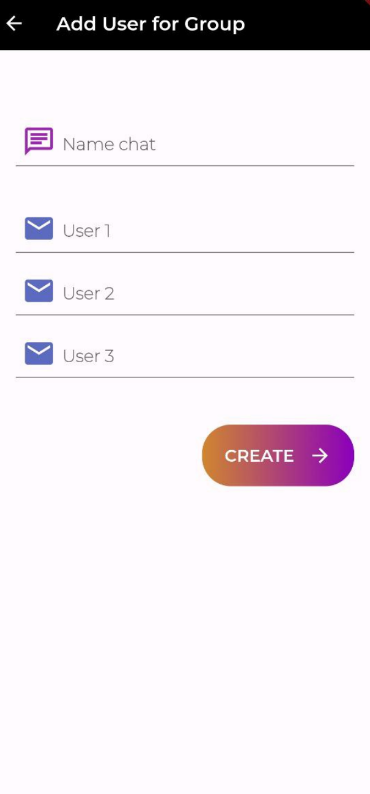


Рисунок 5.13 – Страница групповых чатов

При нажатии на иконку в виде стрелки пользователь переходит на страницу чата. На рисунке 5.14 представлена данная страница.



Рисунок 5.14 – Страница группового чата

Теперь рассмотрим часть приложения для роли администратора. При входе в приложение как администратор пользователю открывается страница со списком всех пользователей.

На рисунке 5.15 представлена данная страница.

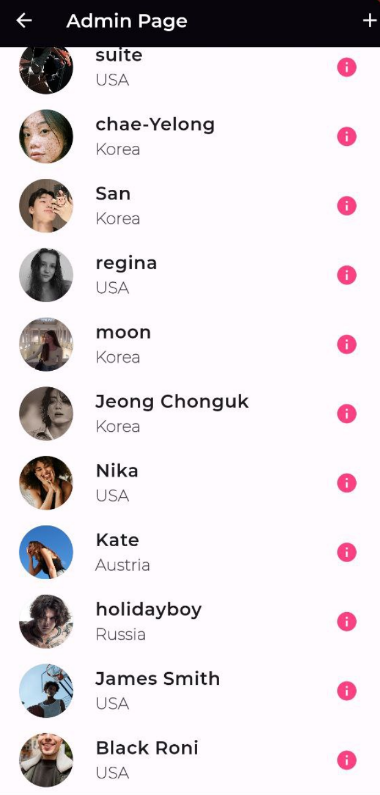


Рисунок 5.15 – Страница администратора

При нажатии на иконку «плюс» пользователь переходит на страницу добавления пользователя. Для добавления пользователя необходимо сначала нажать на кнопку «Create», а потом кнопку «Back» чтобы вернуться на страницу администратора.

На рисунке 5.16 представлена данная страница и список пользователе после добавления нового пользователя.

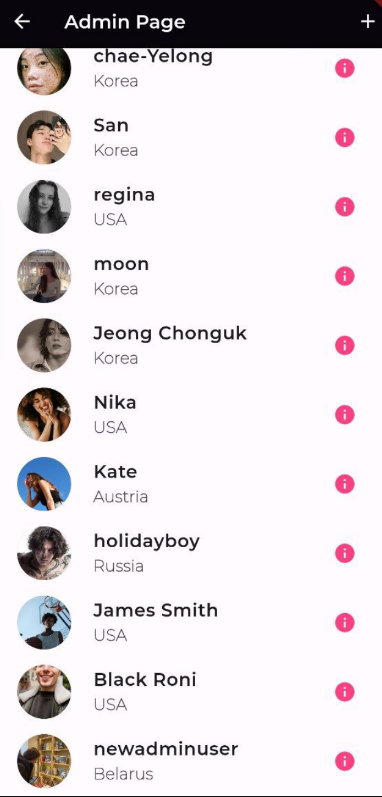
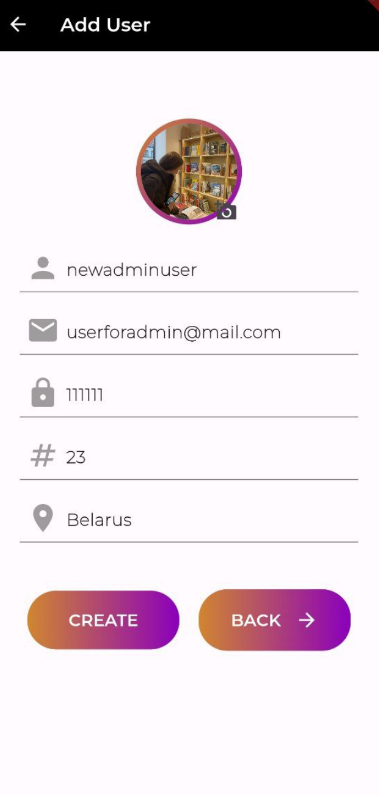


Рисунок 5.16 – Страница добавления пользователя

При нажатии иконку «информация», администратор переходит на страницу, где находится все информация о определенном пользователе.

На рисунке 5.17 представлена данная страница.

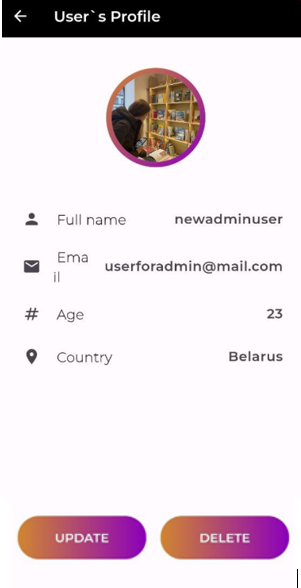
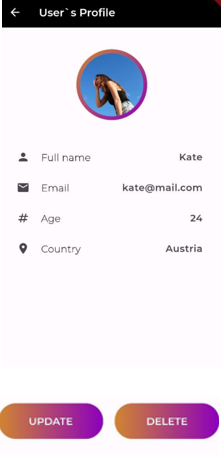
 

Рисунок 5.17 – Страница с информацией о пользователе

При нажатии кнопку «Update» администратор пользователь может изменить информацию о пользователе. Для этого после нажатия появляется окно ввода с полями для изменения информации и для загрузки фотографии. Окно представлено на рисунке 5.18.

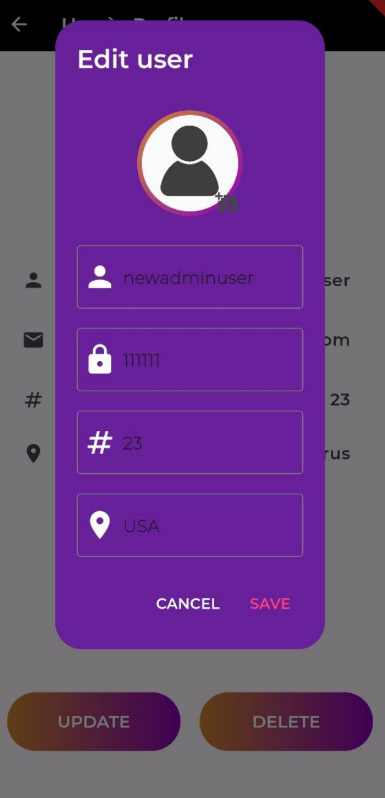


Рисунок 5.17 – Изменение информации о пользователе

При нажатии кнопку «Delete» администратор удаляет пользователя. При переходе на страницу с информацией о пользователе, админ может изменять информацию о пользователе.

На рисунке 5.19 представлена данная операция.

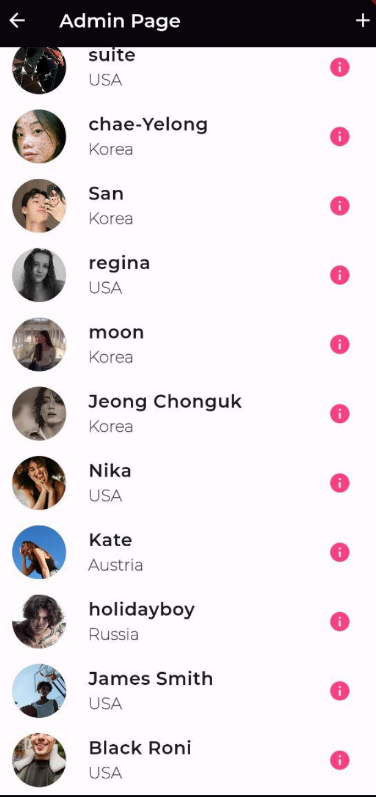
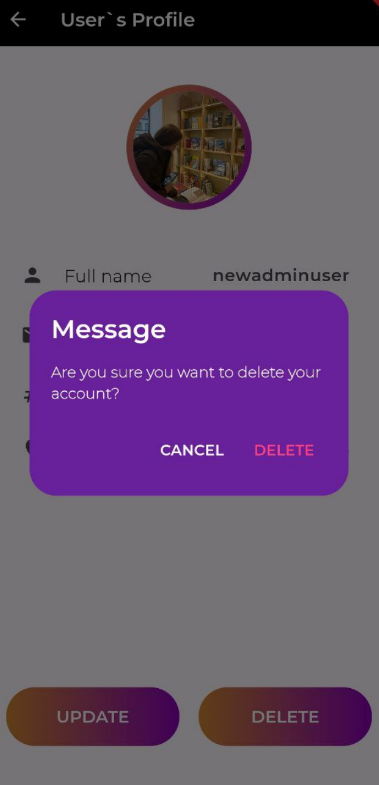


Рисунок 5.19 – Удаление пользователя

## 5.2 Выводы по главе «Руководство пользователя»

В данной главе представлено подробное описание функционала мобильного приложений, предназначенных для администраторов и пользователей. Важными аспектами являются удобство входа через адрес электронной почты, а также разнообразие функциональности приложения.

Мобильное приложение обеспечивает участникам простой вход и навигацию, взаимодействие с другими пользователями.

# Заключение

В ходе выполнения курсового проекта было разработано приложение мессенджер. Перед выполнением курсового задания были просмотрены и проанализированы три аналога приложения.

В результате написания курсового проекта была использована база данных Firebase и кросс-платформенный фреймворк Flutter.

При разработке были выполнены следующие функциональные требования:

* авторизация и регистрация аккаунта;
* управление аккаунтом (добавление, удаление, изменение);
* поиск друзей;
* создание чата с другом;
* создание общего чата.

Наконец, важно отметить, что разработанное приложение не только соответствует техническим требованиям проекта, но и ориентировано на удовлетворение потребностей конечных пользователей, обеспечивая им удобный и эффективный опыт в использовании приложения для проведения олимпиад по базам данных.

# Список использованных источников

1. Разработка приложения на Flutter с нуля [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/594963/ – Дата доступа: 02.03.2024.
2. Документация по компонентам MaterialUI [Электронный ресурс]. – Режим доступа: https://m3.material.io/components. – Дата доступа: 02.04.2024.
3. Документация о Firebase Authentication [Электронный ресурс]. – Режим доступа: https://firebase.google.com. – Дата доступа: 16.03.2024.
4. Статья о использовании Firestore [Электронный ресурс]. – Режим доступа: https://medium.com/firebase-tips-tricks. – Дата доступа: 30.03.2024.
5. Flutter Documentation [Электронный ресурс]. – Режим доступа: https://docs.flutter.dev/. – Дата доступа: 01.04.2024.
6. Firebase Documentation [Электронный ресурс]. – Режим доступа: https://firebase.google.com/. – Дата доступа: 29.04.2024.

**ПРИЛОЖЕНИЕ А.** **Диаграммы вариантов использования**

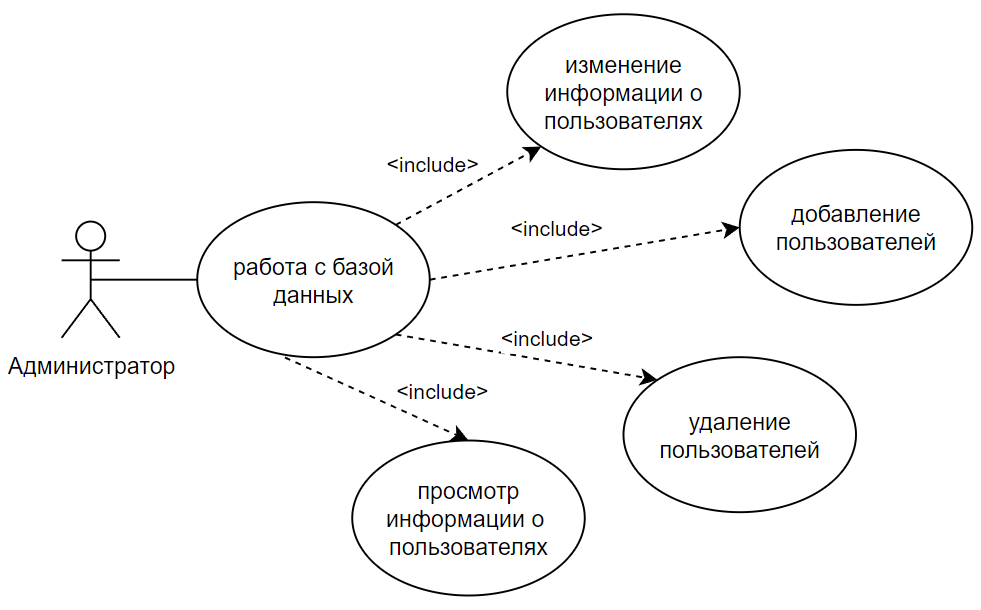


Рисунок 1 – Диаграмма вариантов использования (администратор)

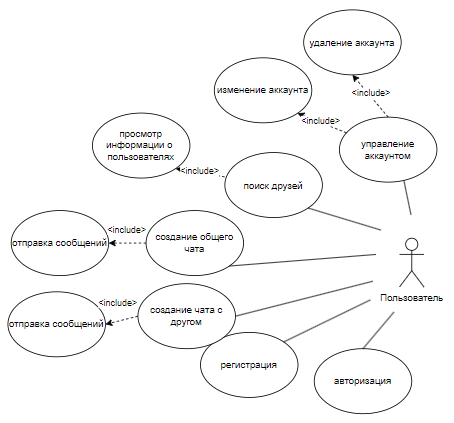


Рисунок 2 – Диаграмма вариантов использования (пользователь)

# ПРИЛОЖЕНИЕ Б. Структура базы данных

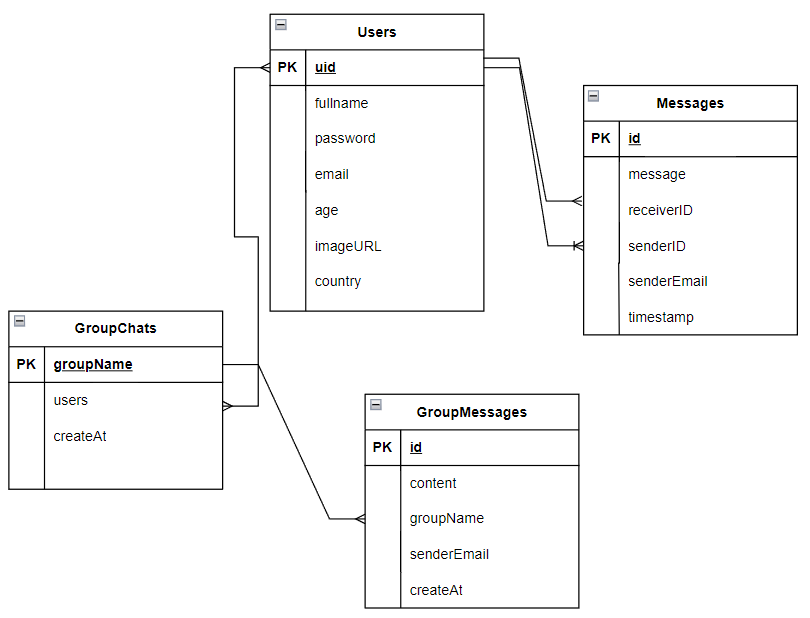


Рисунок 1 – Структура базы данных

# ПРИЛОЖЕНИЕ B. Диаграмма компонентов

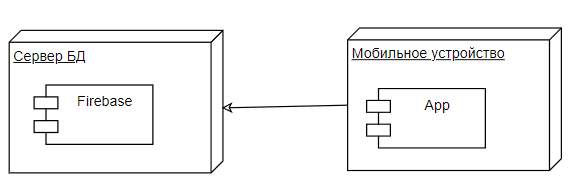


Рисунок 1 – Архитектура приложения (диаграмма компонентов)