МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной

безопасности мобильных систем»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА

по дисциплине «Программирование мобильных систем»

Тема «Приложение для просмотра коротких видео»

Исполнитель

студентка 3 курса 7 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. М. Рациборская

подпись, дата

Руководитель

асс. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. И. Уласевич

преподавателя

должность, ученая степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. И. Уласевич

подпись дата инициалы и фамилия

Содержание

[Введение 3](#_Toc197167749)

[1. Постановка задачи и анализ аналогичных решений 4](#_Toc197167750)

[1.1 Постановка задачи 4](#_Toc197167751)

[1.2 Актуальность задачи 4](#_Toc197167752)

[1.3 Обзор аналогичных решений 4](#_Toc197167753)

[1.3.1 TikTok 4](#_Toc197167754)

[1.3.2 Likee 5](#_Toc197167755)

[1.4 Вывод 6](#_Toc197167756)

[2. Проектирование мобильного приложения 7](#_Toc197167757)

[2.1 Диаграмма вариантов использования 7](#_Toc197167758)

[2.2 Проектирование базы данных 7](#_Toc197167759)

[2.3 Вывод 9](#_Toc197167760)

[3. Программная реализация приложения 10](#_Toc197167761)

[3.1 Технические средства для разработки 10](#_Toc197167762)

[3.2 Разработка глобальной базы данных 10](#_Toc197167763)

[3.3 Аутентификация пользователей в системе 11](#_Toc197167764)

[3.4 Разработка мобильного приложения 11](#_Toc197167765)

[3.5 Вывод 16](#_Toc197167766)

[4. Анализ информационной безопасности приложения 18](#_Toc197167767)

[4.1 Защита пользовательских данных 18](#_Toc197167768)

[4.2 Вывод 18](#_Toc197167769)

[5. Тестирование приложения 19](#_Toc197167770)

[5.1 Тестирование валидации данных 19](#_Toc197167771)

[5.2 Вывод 25](#_Toc197167772)

[6. Руководство пользователя 26](#_Toc197167773)

[6.1 Пользователь 26](#_Toc197167775)

[6.2 Администратор 37](#_Toc197167776)

[6.3 Вывод 40](#_Toc197167777)

[Заключение 41](#_Toc197167778)

[Список используемых источников 42](#_Toc197167779)

[ПРИЛОЖЕНИЕ А: Диаграмма вариантов использования пользователя 43](#_Toc197167780)

[ПРИЛОЖЕНИЕ Б: Диаграмма вариантов использования администратора 44](#_Toc197167781)

[ПРИЛОЖЕНИЕ В: Схема базы данных 45](#_Toc197167782)

# **Введение**

В настоящее время стремительно набирают популярность мобильные приложения, которые предоставляют пользователю все необходимые услуги. Это возможно благодаря стремительно развивающимся технологиям, которые дают доступ пользователю к учебе, развлечениям или просто к нужным услугам. Разработка мобильного приложения для просмотра коротких видеороликов даёт возможность его пользователям отвлечься и всегда быть в курсе современных трендов. Такие видео удобно просматривать в любом месте, где бы не находился пользователь: дом, транспорт, очередь. Данное приложение поможет скоротать время. Алгоритмы персонализации удерживают внимание пользователей, предлагая контент по их интересам.

Кроме того, пользователи имеют возможность не только потреблять контент, но и создавать его. Каждый пользователь может вести свой блог, загружая собственные видеоролики. Это создает сообщество активных участников, которые могут обмениваться идеями, делиться опытом и взаимодействовать друг с другом.

Целью данного курсового проекта является разработка мобильного приложения для просмотра коротких видео, где пользователи могут не только просматриваться ленту с видео, но и публиковать свои собственные. Будут исследованы различные аспекты данного приложения такие как функциональные возможности, способы взаимодействия с базой данных, обработка ошибок и валидация данных, пользовательский интерфейс.

Курсовой проект должен быть завершен выводами по каждому разделу работы и заключением, включающий вывод по проделанной работе.

В конечном итоге, создание мобильного приложения для просмотра коротких видеороликов не только предоставляет пользователям развлекательную платформу, но и создает возможности для самовыражения и взаимодействия, что делает его актуальным и востребованным в современном цифровом мире.

# **Постановка задачи и анализ аналогичных решений**

## **1.1 Постановка задачи**

Целью курсового проекта является разработка программного средства, представляющего собой мобильное приложение, которое должно предоставлять возможности просмотра и публикации коротких видеороликов. Функционально должны быть выполнены следующие задачи:

− определение ролей (администратор, пользователь);

− добавление видео;

− возможность комментировать видео и ставить лайки;

− поиск пользователей;

− подписка на пользователей;

− просмотр статистики профиля.

## **1.2 Актуальность задачи**

В последние годы наблюдается значительный рост интереса к мобильным приложениям для просмотра коротких видеороликов. Современные пользователи, особенно молодое поколение, предпочитают получать информацию и развлечение в формате коротких видео, что связано с изменением потребительских привычек и увеличением времени, проводимого на мобильных устройствах.

Основными факторами, способствующими популярности таких приложений, являются:

* удобство доступа: мобильные приложения позволяют пользователям просматривать видео в любое время и в любом месте, что делает их идеальным инструментом для быстрого потребления контента;
* персонализация контента: алгоритмы, предлагающие пользователям контент на основе их интересов, значительно повышают вовлеченность;
* **социальное взаимодействие**: возможности для создания и обмена контент способствуют формированию активного сообщества и взаимодействию между пользователями.

## **1.3 Обзор аналогичных решений**

Первым этапом разработки программного продукта выступает исследование существующих аналогов. Современный цифровой рынок предлагает обширное разнообразие приложений с аналогичным функционалом. Особого внимания заслуживает сегмент мобильных платформ, специализирующихся на демонстрации кратких видеороликов, где можно выделить ряд лидирующих решений.

### 1.3.1 TikTok

«TikTok»– это социальная сеть, которая позволяет пользователям делиться и просматривать короткие видеоролики. Данная социальная сеть позволяет пользователю загружать видео, вести прямые эфиры, подписываться на других пользователей, оставлять лайки и комментарии. Интерфейс приложения «TikTok» представлен на рисунке 1.1.

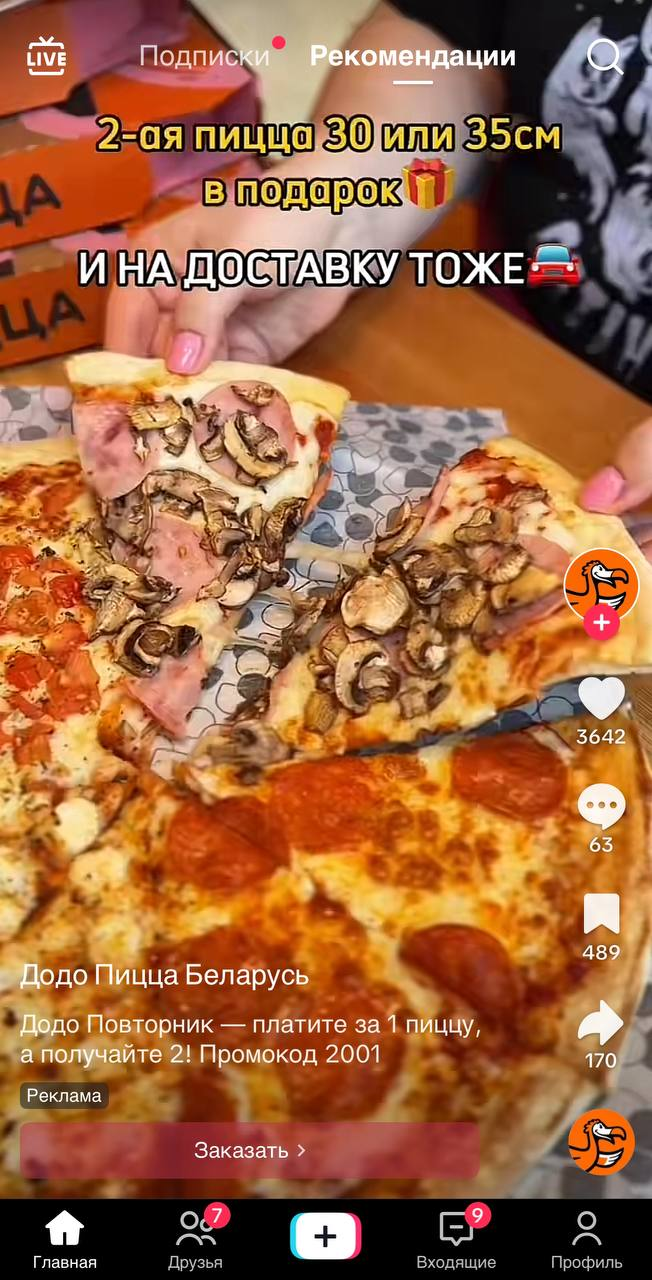


Рисунок 1.1 – Интерфейс приложения «TikTok»

«TikTok» предоставляет пользователю удобный интерфейс для просмотра видеороликов. Приложение оснащено удобным и интуитивно понятным меню внизу страницы для переключения между вкладками. Пользователь может легко переключаться по вкладкам для перехода на нудные страницу приложения. Важным преимуществом данного приложения является кроссплатформенность. Пользователь может установить и использовать приложение на различных платформах таких как Windows, MacOS, Android и IOS. Среди недостатков приложения можно выделить наличие рекламы, что может мешать пользователю просматривать интересующий его контент.

### 1.3.2 Likee

«Likee» – второй аналог приложения по просмотру коротких видеороликов. Приложение «Likee» стало значимой частью рынка платформ для обмена короткими видеороликами и привлекло внимание пользователей благодаря своим уникальным функциям и возможностям. Оно позволяет пользователям создавать видеоролики различной продолжительности, используя широкий выбор фильтров, эффектов и звуков, что делает контент более привлекательным. «Likee» активно поощряет взаимодействие между пользователями, предлагая возможность подписываться на любимых авторов, оставлять комментарии и лайки, что способствует формированию сообщества. Кроссплатформенность приложения обеспечивает доступность на разных устройствах, позволяя пользователям оставаться на связи и создавать контент в любое время и в любом месте. Интерфейс приложения «Like» представлен на рисунке 1.2.

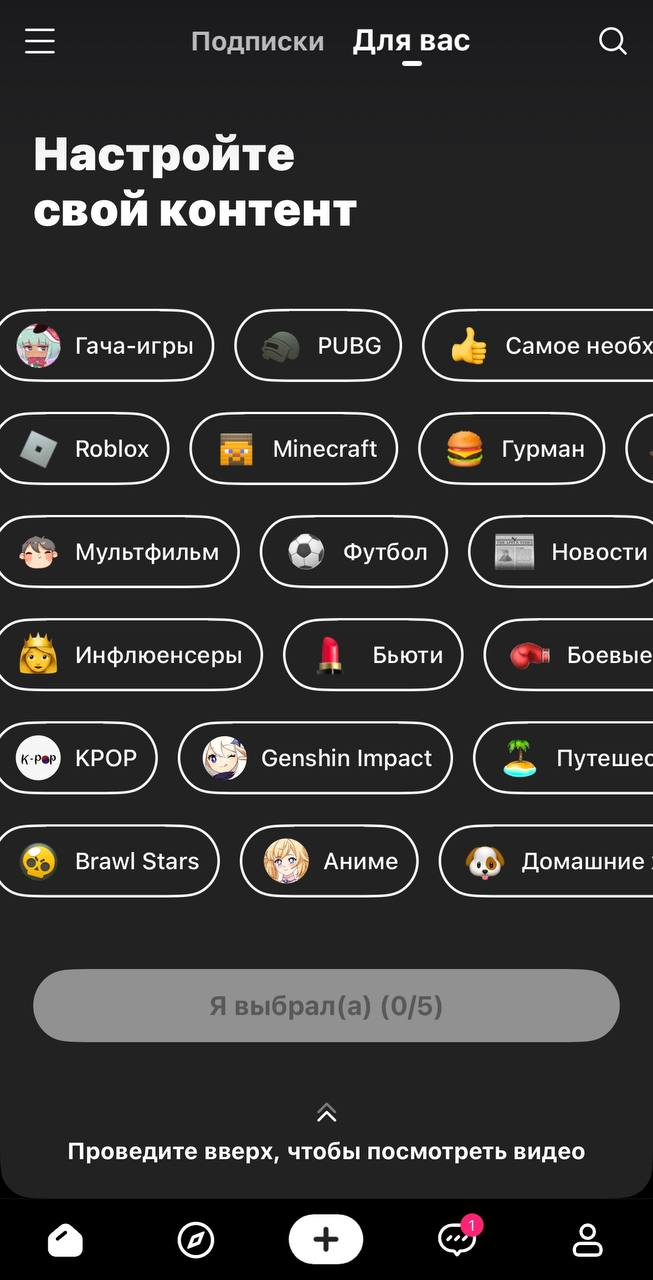


Рисунок 1.2 – Интерфейс приложения «Likee»

Данное приложение также оснащено удобным нижним меню для навигации между страницами. Также важным преимуществом приложения является возможность настроить контент, который пользователю будет интересно просматривать. Данное приложение предоставляет возможность регистрации и авторизации различными способами, например с помощью почты или по номеру телефона, что явно является преимуществом. К недостаткам данной платформы можно отнести отсутствие просмотра подробной статистики профиля.

## **1.4 Вывод**

В ходе анализа прототипов и литературных источников выявлены основные задачи и функциональные требования проекта. Необходимо разработать мобильное приложение для просмотра коротких видеороликов, предоставляющее пользователям следующие функции:

* простая и быстрая регистрация и авторизация;
* интуитивно понятный и ненагруженный интерфейс;
* возможность оставлять лайки и комментарии на видео;
* возможность публиковать свои собственные видео;
* просмотр статистики профиля.

Разработка данного приложения направлена на создание удобной платформы для просмотра коротких видеороликов, которая позволит пользователям скоротать время, просматривая интересующий их контент.

# **Проектирование мобильного приложения**

Проектирование является критическим этапом в процессе разработки программного обеспечения, которым часто пренебрегают неопытные разработчики. Это чревато отсутствием четкого плана действий, что может привести к тому, что проект будет отложен на неопределенное время.

## **2.1 Диаграмма вариантов использования**

Обычно при проектировании разработчики представляют систему в графическом виде, так как этот формат легко воспринимается человеком. Именно поэтому вместо того, чтобы писать объемные тексты о каждой функции будущей программы, разработчики создают различные диаграммы для описания своих систем.

Диаграмма вариантов использования (use-case diagram) – это диаграмма, которая описывает, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

В ходе разработки программного средства будут реализованы два типа пользователей: пользователь и администратор.

Диаграмма вариантов использования для пользователя и администратора представлена в приложении А и в приложении Б.

## **2.2 Проектирование базы данных**

Прежде чем приступить к разработке программного средства необходимо спроектировать базу данных.

Для достижения поставленных задач была разработана база данных с использованием Supabase. **Supabase** - это платформа с открытым исходным кодом, которая предоставляет набор инструментов для разработки приложений.

Для реализации функционала мобильного приложения для просмотра коротких видеороликов будет создано 5 таблиц.

Таблица «User» будет содержать информацию о пользователях. Структура представлена в таблице 2.1.

Таблица 2.1 – Описание таблицы User

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ID | uuid | Идентификатор пользователя |
| User\_name | varchar | Имя пользователя |
| Created\_at | timestamptz | Время создания пользователя |
| Updated\_at | timestamptz | Время обновления пользователя |
| email | text | Почта пользователя |
| Profile\_image\_url | text | Фото профиля пользователя |
| description | text | Описания профиля |
| role | text | Роль пользователя |
| password | text | Пароль пользователя |
| gender | text | Гендер пользователя |

Таблица «Videos» отвечает за хранение информации о видео. Перечень и описание ее столбцов приведены в таблице 2.2.

Таблица 2.2 – Описание таблицы Videos

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ID | uuid | Идентификатор видео |
| Username | text | Имя автора видео |
| Created\_at | timestamptz | Время создания видео |
| Uid | uuid | Идентификатор автора видео |
| likes | Int8 | Количество лайков на видео |
| commentCount | Int8 | Количество комментариев на видео |
| caption | text | Описание видео |
| videoUrl | text | Ссылка на видео |
| likedBy | uuid | Пользователи, которым понравилось видео |

Таблица «Reports» отвечает за хранение информации о жалобах на пользователей, комментарии и видео. Перечень и описание ее столбцов приведены в таблице 2.3.

Таблица 2.3 – Описание таблицы Reports

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ID | uuid | Идентификатор жалобы |
| Reporter\_id | uuid | Идентификатор кто пожаловался |
| Reported\_user\_id | uuid | Идентификатор на кого жалоба |
| reason | text | Причина жалобы |
| Created\_at | timestamptz | Время создания жалобы |
| Reported\_video\_id | uuid | Идентификатор видео на которое жалоба |
| Reported\_comment\_id | uuid | Идентификатор комментария на который жалоба |

Таблица «Followers» отвечает за хранение информации о подписках и подписчиках. Перечень и описание ее столбцов приведены в таблице 2.4.

Таблица 2.4 – Описание таблицы Followers

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ID | uuid | Идентификатор записи |
| Follower\_id | uuid | Идентификатор пользователя, на кого подписаны |
| Following\_id | uuid | Идентификатор пользователя, кто подписан |
| Created\_at | timestamptz | Время, когда подписались. |

Таблица «Comments» отвечает за хранение информации о комментариях. Перечень и описание ее столбцов приведены в таблице 2.5.

Таблица 2.5 – Описание таблицы Comments

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ID | uuid | Идентификатор комментария |
| video\_id | uuid | Идентификатор видео |
| user\_id | uuid | Идентификатор пользователя |
| Created\_at | timestamptz | Время создания комментария |
| Commet\_text | text | Текст комментария |

Для понимания структуры разрабатываемой базы данных была создана логическая схема, представленная в приложении В.

## **2.3 Вывод**

В данном разделе была подчеркнута важность тщательной проработки этапа проектирования, которая позволяет избежать хаоса в процессе разработки. Создание диаграмм вариантов использования, продемонстрированная в приложении А и приложении Б, помогает визуализировать функционал системы и облегчает понимание требований. Проектирование базы данных, выполненное с использованием Supabase, стало ключевым шагом в реализации приложения. Разработанные таблицы, такие как «User», «Videos», «Reports», «Followers» и «Comments», обеспечивают хранение необходимой информации и поддержку всех запланированных функций приложения. Логическая схема базы данных, представленная в приложении В, позволяет лучше понять связи между таблицами и структуру данных, что является основой для дальнейшей разработки.

Таким образом, проделанная работа закладывает фундамент для создания функционального и стабильного приложения для просмотра коротких видеороликов. Тщательное проектирование позволяет не только реализовать текущие требования, но и обеспечивает гибкость для масштабирования и внедрения нового функционала в будущем. Использование современных инструментов разработки, таких как Supabase, повышает эффективность и надежность системы.

# **Программная реализация приложения**

## **3.1 Технические средства для разработки**

Для реализации курсового проекта использовались Android Studio, язык программирования Dart и фреймворк Flutter.

Flutter – это кросс-платформенный фреймворк с открытым исходным кодом, разработанный компанией Google, который предоставляет обширную библиотеку готовых виджетов для создания интерфейсов приложений.

Dart – объектно-ориентированный язык программирования, также разработанный Google, который служит основным языком при разработке мобильных приложений с использованием Flutter.

В качестве базы данных была выбрана платформа Supabase, которая предлагает функционал для управления данными, аутентификации пользователей и хранения файлов.

Таким образом, приложение будет построено с использованием стека технологий: Android Studio, Dart, Flutter и Supabase, а также других вспомогательных библиотек.

## **3.2 Разработка глобальной базы данных**

Для разработки глобальной базы данных использовался Supabase, что упростило процесс разработки и управления данными. Были созданы необходимые таблицы для хранения информации о пользователях и видео.

Supabase автоматически создает RESTful API на основе разработанной схемы базы данных.

Для подключения к Supabase в приложении используется следующий код, представлен в листинге 3.1.

await Supabase.initialize(

url: 'https://ehdliqszkyprsiurkjwf.supabase.co',

anonKey: 'eyJhbGciOiJI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImVoZGxpa3lwcnNpdXJrandmIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDE5NjAzNTcsImV4cCI6MjA1NzUzNjM1N30.wpIIX2VRKDPQAd0J9VdADf4-BlngPM6XHrkLfXbo0',);

Листинг 3.1 – Код строки подключения к Supabase

Перед добавлением вышенаписанного код, в файл pubspec.yaml. были добавлены следующие зависимости для работы в Supabase, продемонстрированные в листинге 3.2

supabase\_flutter: ^2.8.4

Листинг 3.2 – Библиотеки в файле pubspec.yaml

Добавление данной зависимости обеспечивает доступ к методам Supabase, необходимым для аутентификации, работы с базой данных и хранилищем.

## **3.3** **Аутентификация пользователей в системе**

Программное средство подразумевает использование пользователями, а также администратором. Для этого необходима регистрация и аутентификация в системе. Supabase предоставляет встроенные функции аутентификации, которые были интегрированы в приложение. Это позволяет пользователям безопасно регистрироваться и входить в систему, а также управлять своими профилями. В системе может зарегистрироваться каждый пользователь. Данные, введенные пользователями, должны проходить валидацию на пустую строку и длину. Так же, когда пользователей регистрируется в приложение, ему приходить письмо на почту, указанную при регистрации, для подтверждения email. Для этого использовался встроенный механизм Supabase, а именно «Confirm signup».

## **3.4** **Разработка мобильного приложения**

При разработке мобильного приложения для просмотра коротких видеороликов использовался Flutter.

Мобильное приложение построено с использованием нескольких папок, каждая из которых выполняет определенную функцию и обеспечивает структурированное и логичное разделение кода. Файловая структура представлена на рисунке 3.1.

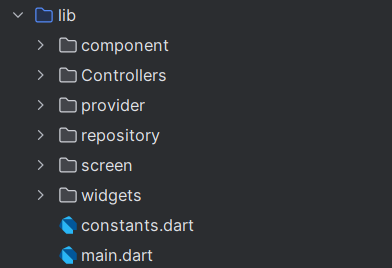


Рисунок 3.1 – Файловая архитектура мобильного приложения

В решении проекта содержатся директории, каждая из которых несет в себе определенное назначение.

* Component включает в себя отдельные элементы интерфейса, например нижнее навигационное меню.
* Controllers содержит компоненты, которые обеспечивают работу приложения с базой данных, такие как поиск пользователей, превью видео, подписка на пользователей, проигрывание видео.
* Provider содержит компоненты для управления состоянием, например класс, обеспечивающий централизованное управление состоянием навигации.
* Screen содержит классы, которые определяют страницы приложения и их функциональность, например страница авторизации, регистрации и страница пользователя.
* Widgets предназначена для хранения пользовательских виджетов, которые используются в приложении, например класс, который определяется реализацию текстового поля ввода.
* Repository содержит модель данных, такие как users.

В качестве примера в листинге 3.3 представлен код модели пользователей. Модель CustomUser представляет собой структуру данных, которая используется для хранения информации о пользователях в приложении. Метод fromJson предоставляет удобный способ создания экземпляра CustomUser из JSON-объекта. Это нужно для получения данных от API.

class CustomUser

{

final String id;

final String userName;

final String email;

final DateTime createdAt;

final DateTime updatedAt;

final String? profileImageUrl;

CustomUser

(

{

required this.id,

required this.userName,

required this.email,

required this.createdAt,

required this.updatedAt,

this.profileImageUrl,

});

factory CustomUser.fromJson(Map<String, dynamic> json)

{

return CustomUser(

id: json['id'],

userName: json['user\_name'],

email: json['email'],

createdAt: DateTime.parse(json['created\_at']

),

updatedAt: DateTime.parse(json['updated\_at']

),

profileImageUrl: json['profile\_image\_url'],

);

}

}

Листинг 3.3 – Модель пользователей.

В листинге 3.4 представлен пример компонента TiktokNavItemWidget. Этот компонент представляет собой интерактивные кнопки для навигации по различным разделам приложения.

import 'package:flutter/material.dart';

class TiktokNavItemWidget extends StatelessWidget {

const TiktokNavItemWidget({

super.key,

this.icon,

required this.label,

this.imgPath,

this.onTab,

required this.isSelect,

});

final IconData? icon;

final String label;

final String? imgPath;

final Function()? onTab;

final bool isSelect;

@override

Widget build(BuildContext context) {

return InkWell(

onTap: onTab,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

icon == null

? (imgPath != null

? Image.asset(

imgPath!,

width: 50.0,

height: 50.0,

): SizedBox())

: Icon(icon, color: isSelect ? Colors.red : Colors.white, size: 27.0),

Text(

label,

style: Theme.of(context).textTheme.bodyMedium?.copyWith(color: Colors.white

),

),

],

),

);}

Листинг 3.4 – Пример компонента TiktokNavItemWidget

Для создания превью видео был разработан класс VideoThumbnailWidget. Этот класс предназначен для генерации и отображения миниатюры видеофайла. Он принимает путь к видео в качестве параметра и автоматически создает миниатюру при инициализации. Код данного класса представлен в листинге 3.5.

class VideoThumbnailWidget extends StatefulWidget {

final String videoPath;

const VideoThumbnailWidget({Key? key, required this.videoPath}) : super(key: key);

@override

\_VideoThumbnailWidgetState createState() => \_VideoThumbnailWidgetState();}

class \_VideoThumbnailWidgetState extends State<VideoThumbnailWidget> {

String? \_thumbnailPath;

@override

void initState() {

super.initState();

\_generateThumbnail();}

Future<void> \_generateThumbnail() async {

final thumbnail = await VideoThumbnail.thumbnailFile(

video: widget.videoPath,

imageFormat: ImageFormat.JPEG,

quality: 50,);

setState(() {

\_thumbnailPath = thumbnail.path;});

}

@override

Widget build(BuildContext context) {

return \_thumbnailPath != null

? Image.file(File(\_thumbnailPath!), fit: BoxFit.cover)

: const Center(child: CircularProgressIndicator());}

}

Листинг 3.5 – Код класса VideoThumbnailWidget

Класс NavProvider используется для управления навигацией в приложении для переключения между страницами с помощью нижнего навигационного меню. Код класса NavProvider представлен в листинге 3.6.

class NavProvider extends ChangeNotifier{

int \_pageIndex=0;

int get pageIndex=> \_pageIndex;

void onChangePage(int index){

\_pageIndex=index;

notifyListeners();

}}

Листинг 3.6 – Код класса NavProvider

Класс TiktokPage отвечает за отображение основной структуры экрана. Также реализует нижнюю навигационную панель, позволяя пользователю переключаться между различными страницами приложения. Код класса TiktokPage представлен в листинге 3.7.

class TiktokPage extends StatelessWidget {

final String userId;

const TiktokPage({super.key, required this.userId});

@override

Widget build(BuildContext context) {

return Scaffold(

body: Consumer<NavProvider>(

builder: (context, value, child) => IndexedStack(

index: value.pageIndex,

children: [

RealScreen(currentUserId: userId),

SearchScreen(),

AddVideoScreen(),

ProfileStatsScreen(uid: userId),

ProfileScreen(uid: userId),

],),),

bottomNavigationBar: Consumer<NavProvider>(

builder: (context, value, child) =>

TiktokBottomNavigation(selectIndex: value.pageIndex),

),

);

}

}

Листинг 3.7 – Код класса TiktokPage

Класс TiktokHomeScreen представляет собой виджет, который инициализирует и предоставляет доступ к состоянию навигации через NavProvider. Он принимает userId в качестве параметра и передает его дочернему виджету TiktokPage, обеспечивая таким образом передачу данных между экранами. Код класса TiktokHomeScreen представлена на листинге 3.8.

class TiktokHomeScreen extends StatelessWidget {

final String userId;

const TiktokHomeScreen({super.key, required this.userId});

@override

Widget build(BuildContext context) {

return MultiProvider(

providers: [

ChangeNotifierProvider(create: (context) => NavProvider()),],

child: TiktokPage(userId: userId),);

}

}

Листинг 3.8 – Код класса TiktokHomeScreen

Метод \_saveVideoToDatabase сохраняет видео и всю информацию о нем в базу данных с использованием Supabase. Код метода saveVideoToDatabase представлен в листинге 3.9.

Future<void> \_saveVideoToDatabase(User user, String videoUrl, String caption) async {

try {

final supabase = Supabase.instance.client;

final profile = await supabase.from('user').select('user\_name').eq('id', user.id).single();

final username = profile['user\_name'] ?? 'Аноним';

final response = await supabase.from('videos').insert({

'uid': user.id.toString(),

'username': username,

'videoUrl': videoUrl,

'caption': caption,

'created\_at': DateTime.now().toIso8601String(),

'likes': 0,

'commentCount': 0,

'shareCount': 0,

}

).select();

} catch (e) {

print("Ошибка сохранения в базу: $e");

}

}

Листинг 3.9 – Код метода \_saveVideoToDatabase

Метод \_fetchUserData выполняет загрузку данных пользователя из базы данных Supabase. Код метода fetchUserData представлен в листинге 3.10.

Future<void> \_fetchUserData() async {

try

{

final response = await Supabase.instance.client

.from('user')

.select('user\_name, profile\_image\_url, description, gender')

.eq('id', widget.uid)

.maybeSingle();

setState(() {

userName = response?['user\_name'] ?? 'User';

profileImageUrl = response?['profile\_image\_url'] ?? 'https://example.com/default-avatar.png';

userDescription = response?['description'] ?? 'No description yet';

userGender = response?['gender'] ?? 'Не указан';});

} catch (e)

{

print("Error fetching user data: $e");}

}

Листинг 3.10 – Код метода \_fetchUserData

Правильная организация файлов помогает легче ориентироваться в проекте и облегчает управление кодом.

## **3.5 Вывод**

В данной главе подробно рассмотрен процесс разработки всех компонентов приложения. Были выявлены технические средства для разработки мобильного приложения, а именно бэкенд–решение Supabase и Android Studio как среда разработки. Авторизация и регистрация пользователей осуществляется с помощью встроенных функций, которые предоставляет Supabase. После выявления технических средств приступили к разработке структуры базы данных, определив ключевые таблицы и связи между ними для обеспечения целостности и эффективности работы с данными. Затем перешли к разработке мобильного приложения на Flutter.

Структурирование приложения на Flutter по папкам также сыграло ключевую роль в поддержании чистоты и организованности кода. Этот подход облегчил командную разработку, упростил поиск и исправление ошибок, а также позволил легко добавлять новые функции и масштабировать приложение в будущем. Были использованы общепринятые паттерны проектирования и лучшие практики разработки на Flutter, что обеспечило высокую производительность и стабильность приложения.

Однако, помимо перечисленных этапов, стоит отметить важность выбора Supabase в качестве бэкенд-решения. Supabase предоставляет не только готовую инфраструктуру для аутентификации, но и мощные инструменты для работы с базой данных, хранения файлов и реализации real-time функциональности, что значительно ускорило процесс разработки и позволило сосредоточиться на пользовательском интерфейсе и логике приложения.

Таким образом, в данной главе детально описан весь процесс разработки мобильного приложения для просмотра коротких видеороликов.

# **Анализ информационной безопасности приложения**

При разработке мобильного приложения ключевым аспектом является обеспечение информационной безопасности. Приложение должно защищать данные пользователей от несанкционированного доступа и взлома. Необходимо реализовать меры защиты данных пользователей, включая использование пароля для доступа к приложению и ограничение прав доступа. Важно точно определить и ограничить возможности каждой роли в приложении.

## **4.1 Защита пользовательских данных**

Для хранения и аутентификации пользовательских данных в Supabse применяются современные методы защиты, обеспечивающие высокий уровень конфиденциальности. Основным механизмом аутентификации является встроенный механизм Authentication. Чтобы обеспечить безопасность аккаунта, пароль шифруется с использованием алгоритма хеширования Bcrypt, который базируется на симметричном блочном шифре Blowfish. Этот шифр отличается сложной фазой подготовки ключей шифрования и использует ключевой фактор, регулирующий стоимость хеширования, что является ключевой функцией Bcrypt. Алгоритм хеширования паролей:

* генерация соли – случайной строки, добавляемой к паролю перед хешированием;
* хеширование пароля;
* возвращение хеша.

Hash-функция работает только в одну сторону, что обеспечивает безопасность пароля, так как восстановить исходный пароль по hash невозможно. При вводе пароля система преобразует его в hash и сравнивает с сохраненным в базе данных. Если значения совпадают, доступ предоставляется. Хранение паролей в виде хешей добавляет дополнительный уровень защиты. Даже если злоумышленник получит доступ к базе данных, он увидит лишь непонятные строки хешей, а не сами пароли.

Код, реализующий хеширование пароля, продемонстрирован на листинге 4.1.

final hashedPassword = BCrypt.hashpw(password, BCrypt.gensalt());

Листинг 4.1 – Код, реализующий хеширование пароля

Хеширование с помощью BCrypt позволяет надёжно защитить пароли благодаря использованию соли и адаптивности алгоритма

## **4.2 Вывод**

В данном разделе был проведен анализ информационной безопасности разрабатываемого мобильного приложения. Были выявлены ключевые аспекты: защита данных, ограничение прав доступа пользователей. Были представлены методы защиты пользовательских данных с использованием алгоритма bcrypt, который обеспечивает надежное хеширование паролей пользователей.

# **Тестирование приложения**

Тестирование приложений является важнейшей частью разработки программного обеспечения. Оно позволяет выявлять ошибки, проверять функциональность и качество приложения перед его запуском.

Основная цель тестирования заключается в обнаружении дефектов, оценке функциональности, а также в обеспечении надежности и стабильности его работы. Тестирование помогает удостовериться, что приложение соответствует требованиям и ожиданиям пользователей, выполняет свои функции без сбоев и обеспечивает безопасность и защиту данных.

Тестирование играет важную роль в снижении рисков, связанных с выпуском некачественного программного обеспечения. Это позволяет избежать финансовых потерь, рисков и недовольства пользователей

## **5.1 Тестирование валидации данных**

При использовании мобильного приложения пользователь много взаимодействует с данными. Одним из примеров взаимодействия пользователя с данными является регистрация пользователя. При регистрации пользователю требуется ввести свое имя, почту и пароль. В случае, если пользователь оставит поля ввода пустыми, возникнут проблемы хранения данных и нарушения зависимостей в базе данных. Для того, чтобы исключить такие случаи, требуется проводить валидацию данных.

Рассмотрим пример валидации данных на примере регистрации пользователя.

На рисунке 5.1 продемонстрирована форма для регистрации пользователя в мобильном приложении для просмотра коротких видеороликов.

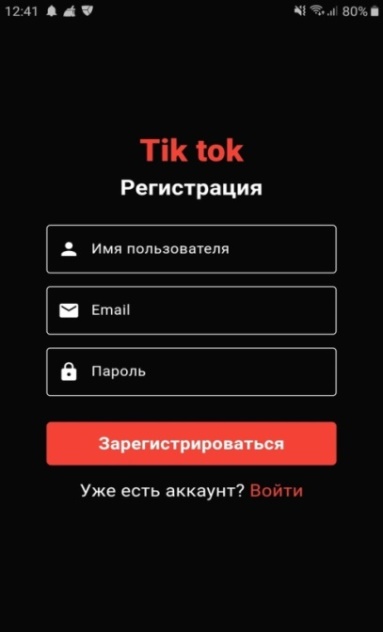


Рисунок 5.1 – Форма для регистрации

В случае, если пользователь не введет в поля ввода для имени пользователя, email и пароля ничего и нажмет на кнопку регистрации, то ему будет выведено следующее уведомление об ошибке, которое представлено на рисунке 5.2.

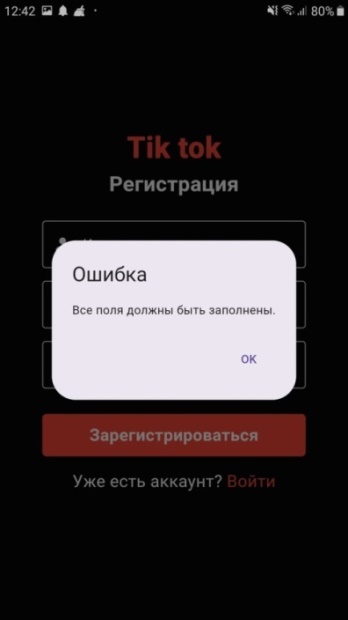


Рисунок 5.2 – Уведомление об ошибке при регистрации, связанное с пустыми полями

Аналогичное сообщение пользователь получит в случае если оставит хотя бы одно поле пустым.

Также предусмотрена валидация на пароль. Пароль должен содержать от 6 до 12 символов включая цифры. Если введенный пользователем пароль не соответствует данным требованиям, он получит предупреждение, продемонстрированное на рисунке 5.3

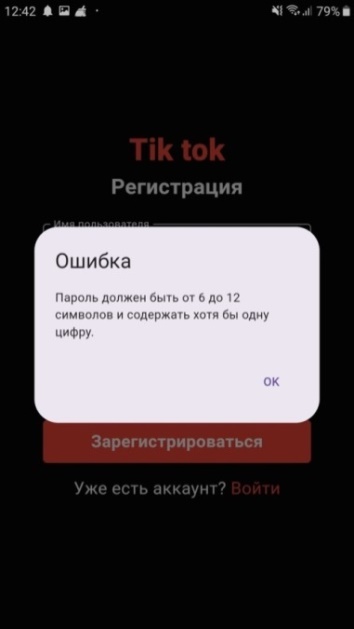


Рисунок 5.3 – Уведомление об ошибке при регистрации, связанное с некорректным паролем

Также, нельзя чтобы в базе приложения существовали пользователи с одинаковой почтой и именем. Для этого создана валидация на имя пользователя и почту. В случае если пользователь вводит имя или почту, под которыми уже зарегистрированы, пользователь получит соответствующие уведомления, продемонстрированные на рисунках 5.4 и 5.5

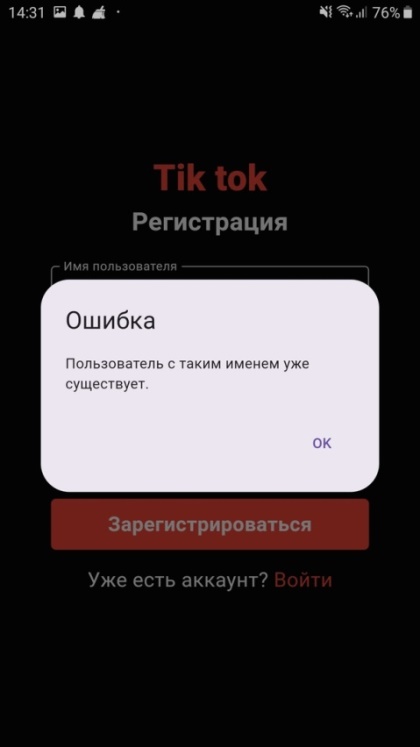


Рисунок 5.4 – Уведомление об ошибке при регистрации, связанное с уже существующим именем пользователя

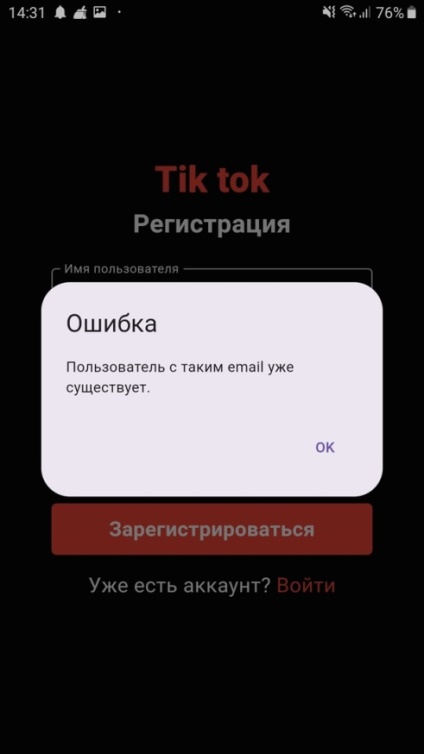


Рисунок 5.5 – Уведомление об ошибке при регистрации, связанное с уже существующей почтой пользователя

Также нередко встречается, что при авторизации пользователь может некорректно вводить данные по разным причинам. Для того, чтобы пользователю была понятна причина, по которой он не может войти в свой аккаунт, разработана валидация при авторизации.

На рисунке 5.6 продемонстрирована форма авторизации.

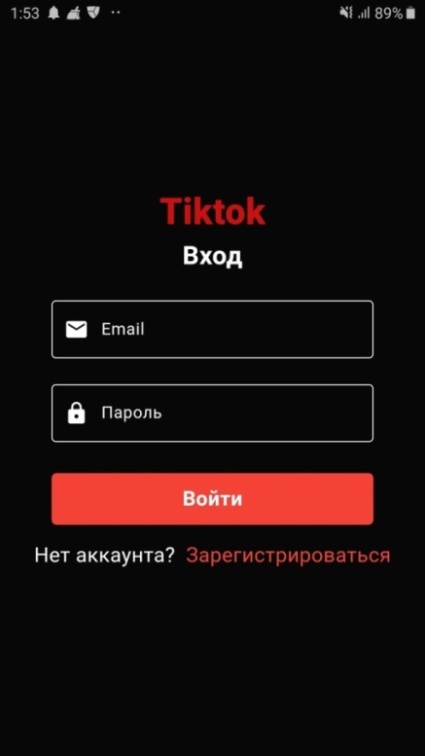


Рисунок 5.6 – Форма авторизации

Если пользователь оставит поля пустыми, ему высветится следующее уведомление, продемонстрированное на рисунке 5.7

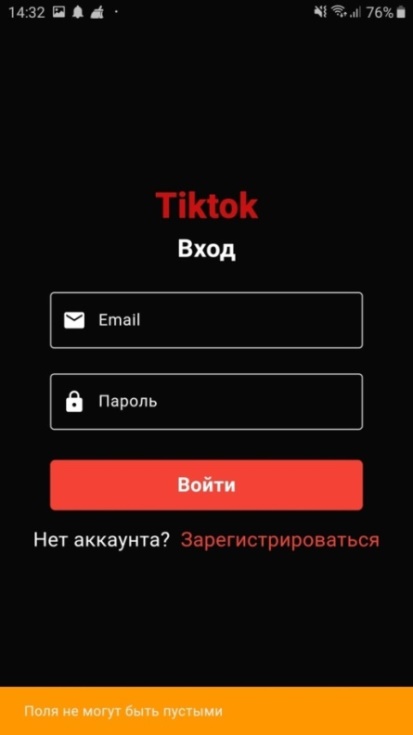


Рисунок 5.7 – Уведомление об ошибке при авторизации, связанное с пустыми полями

В случае, если пользователь ввел неверные данные для входа в систему, ему высветится следующее уведомление об ошибке, продемонстрированное на рисунке 5.8. Это сообщение четко указывает на необходимость проверки введенной информации и предлагает возможность повторного ввода данных.

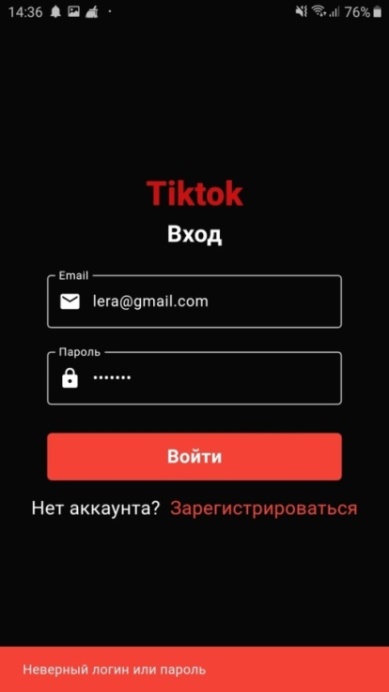


Рисунок 5.8 – Уведомление об ошибке при авторизации, связанное с неверными данными

После регистрации и авторизации пользователь попадает на главную страницу приложения, а именно ленту видео.

Также пользователь на странице профиля может при желании изменить информацию о профиле, такую как имя пользователя, описание профиля, фото профиля и пароль. Для изменения пароля предусмотрена валидация. Первая валидация для пароля это непосредственно количество символов в пароле, а именно от 6 до 12 используя цифры. На рисунке 5.9 представлена валидация на количество символов в пароле.

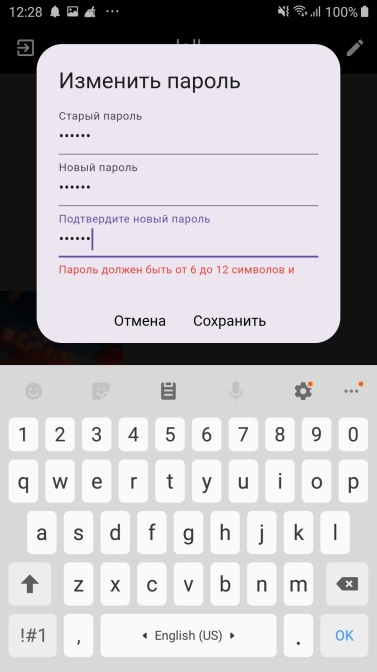


Рисунок 5.9 – Валидация на количество символов

Следующая валидация, предусмотренная при изменении пароля, заключается в проверке совпадения введённого нового пароля с подтверждённым. Это необходимо для предотвращения ошибок, связанных с опечатками при вводе, и гарантирует, что пользователь действительно указал тот пароль, который хотел установить. На рисунке 5.10 продемонстрирован процесс валидации поля подтверждения пароля: если введённые значения не совпадают, система уведомляет пользователя об ошибке и не позволяет продолжить, пока оба поля не будут идентичны.

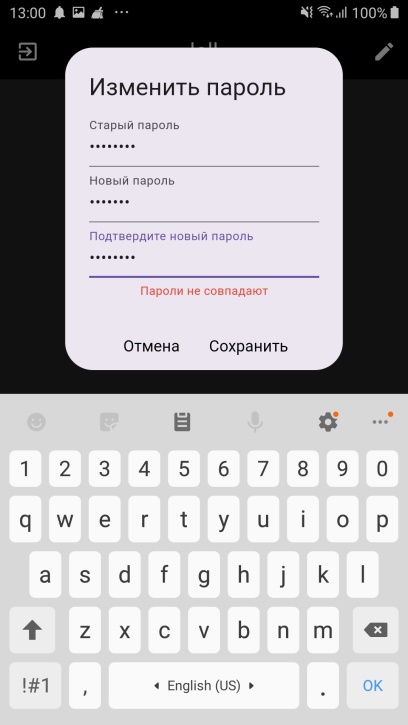


Рисунок 5.10 – Валидация подтверждения пароля

Также предусмотрена валидация на старый пароль, то есть перед тем как изменить старый пароль на новый, пользователю нужно ввести правильный старый пароль. На рисунке 5.11 продемонстрирована валидация на старый пароль.

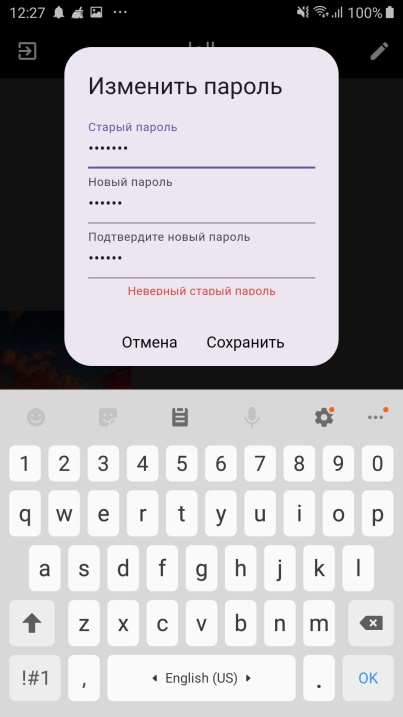


Рисунок 5.11 – Валидация на правильность старого пароля

И последним этапом валидации является проверка, не совпадает ли новый пароль со старым. Валидация совпадения нового пароля и старого продемонстрирована на рисунке 5.12.

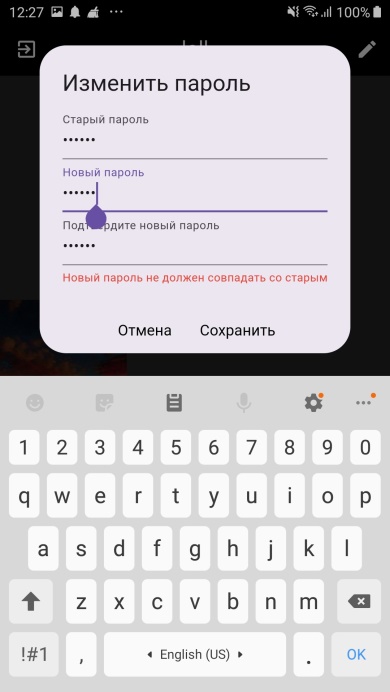


Рисунок 5.12 – Валидация совпадения нового пароля и старого

## Данные этапы валидации обеспечивают безопасность данных пользователя, предотвращая несанкционированный доступ, утечку конфиденциальной информации и возможные попытки взлома.

## **5.2 Вывод**

В данном разделе была подробно рассмотрена реализация валидации данных в мобильном приложении для просмотра коротких видеороликов. Валидация данных — один из ключевых элементов обеспечения корректной и безопасной работы приложения, особенно в тех местах, где пользователь вводит или изменяет информацию, такую как регистрация, авторизация и редактирование профиля.

Особое внимание уделено валидации при изменении пароля, где предусмотрено несколько уровней защиты: проверка длины нового пароля, его совпадения с полем подтверждения, корректность введённого старого пароля, а также контроль на совпадение нового и старого пароля. Эти меры направлены на усиление безопасности пользовательского аккаунта и предотвращение случайных или вредоносных изменений.

# **6. Руководство пользователя**

## В данном разделе описывается руководство для зарегистрированных пользователей, а именно для пользователя и администратора. У каждой роли есть свой функционал, что позволяет эффективно управлять системой и обеспечивает удобство в использовании.

## **6.1 Пользователь**

Для того, чтобы пользователь мог получить доступ к основному функционалу приложения, а именно просмотр видео, создание своего собственного контента, взаимодействие с другими пользователями посредством лайков и комментариев, просмотр статистики профиля, сперва ему нужно авторизоваться либо же зарегистрироваться. Страница с авторизацией представлена на рисунке 6.1.

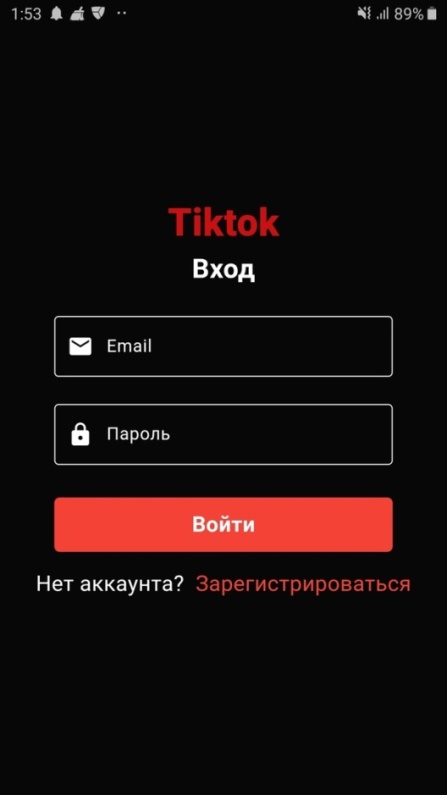


Рисунок 6.1 – Страница авторизации

В случае, если у пользователя еще нет аккаунта, он может нажать на кнопку «Зарегистрироваться» и быть перенаправленным на страницу регистрации, которая продемонстрирована на рисунке 6.2. Процесс регистрации состоит из трех этапов, на каждом из которых пользователю необходимо ввести определенные данные.

На первом этапе пользователю нужно ввести свое имя, что позволит создать персонализированный профиль. Далее, на втором этапе, требуется указать уникальный логин, который будет использоваться для входа в систему. На третьем этапе пользователю необходимо придумать надежный пароль. Рекомендуется использовать комбинацию букв, цифр и специальных символов для повышения безопасности.

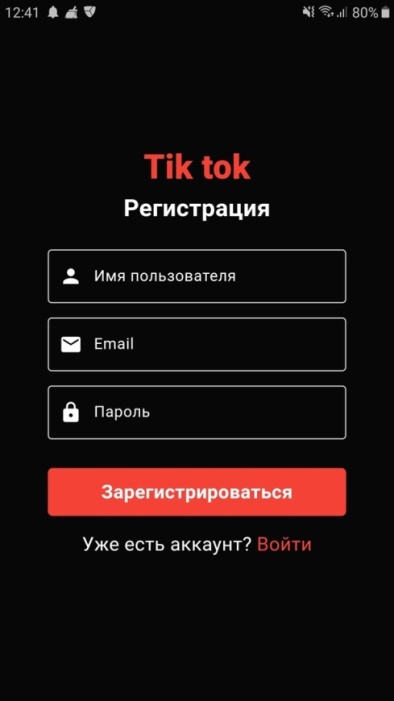


Рисунок 6.2 – Страница регистрации

После успешного завершения всех этапов регистрации пользователь получит сообщение о том, что ему нужно подтвердить свой email. Сообщение для подтверждение email придет на указанную пользователем почту. На рисунке 6.3 представлено само сообщение для подтверждения email.

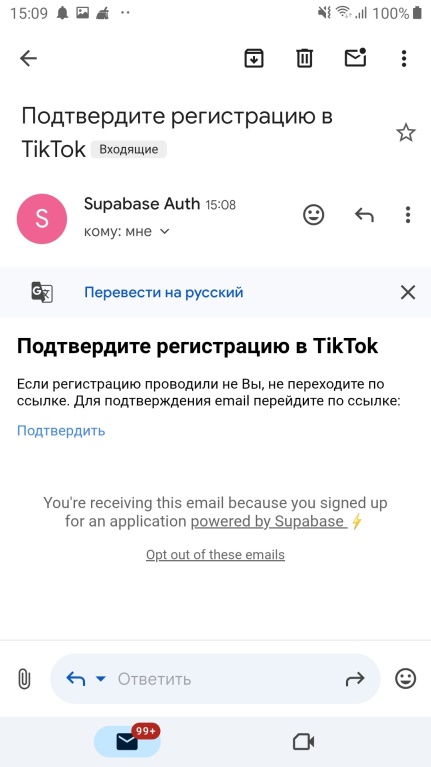


Рисунок 6.3 – Сообщение для подтверждение email

Для подтверждения email пользователю следует перейти по ссылке, указанной в сообщении. В случае, если пользователь не регистрировал свой email, сообщение следует проигнорировать. После перехода по ссылке пользователь перенаправиться на страницу, продемонстрированную на рисунке 6.4.



Рисунок 6.4 – Страница с сообщением о подтверждении email

После того, как пользователь зарегистрировался и авторизовался, он получает доступ к функционалу приложения.

При входе в аккаунт пользователь по умолчанию перенаправляется на страницу ленты с видео. На данной странице пользователь может просматривать видео других пользователей, ставить лайки и комментировать видео. Данная страница продемонстрирована на рисунке 6.5.



Рисунок 6.5 – Страница ленты с видео

Как описывалось выше, каждое видео можно лайкать и комментировать, а также просматривать комментарии других пользователей к видео. При необходимости также существует возможность оставлять жалобы на комментарии и видео. Страница с комментариями продемонстрирована на рисунке 6.6.



Рисунок 6.6 – Страница с отображением комментариев

Если текущий пользователь находит комментарии оскорбительными, он может пожаловаться на них нажав на «Пожаловаться» и выбрав причину жалобы. Данный функционал представлен на рисунках 6.7 и 6.8.

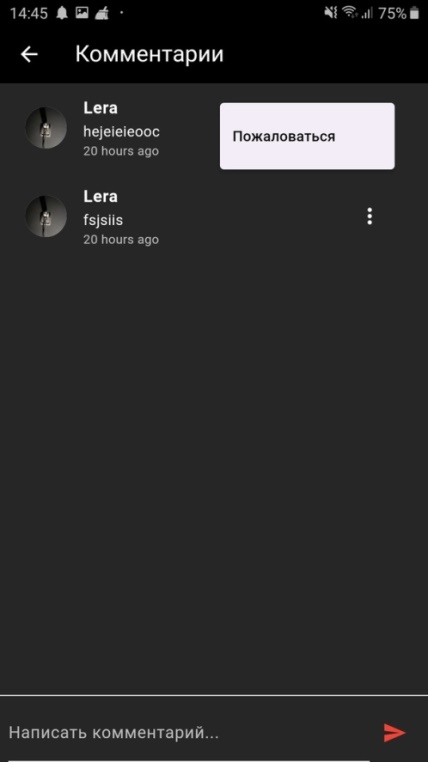


Рисунок 6.7 – Демонстрация функционала жалобы на комментарий

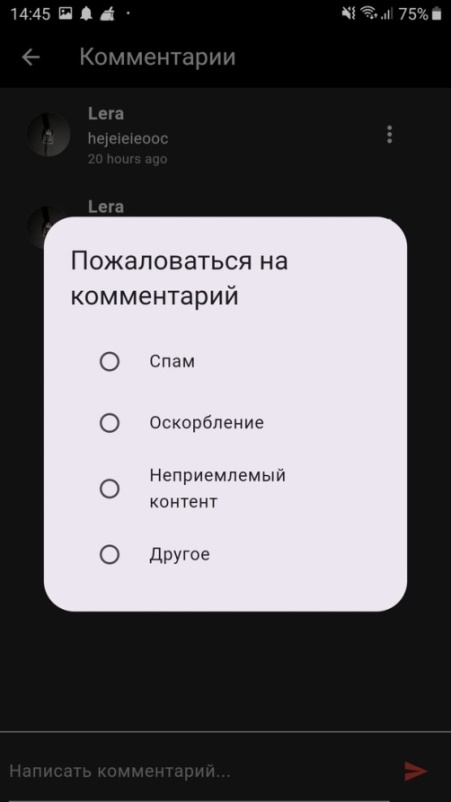


Рисунок 6.8 – Демонстрация функционала жалобы на комментарий

Для перехода на другие страницы приложение оснащено нижним навигационным меню. При нажатии на вкладку «Поиск», пользователь перенаправляется на страницу поиска, где он может найти других пользователей данного приложения. Страница поиска продемонстрирована на рисунке 6.9.

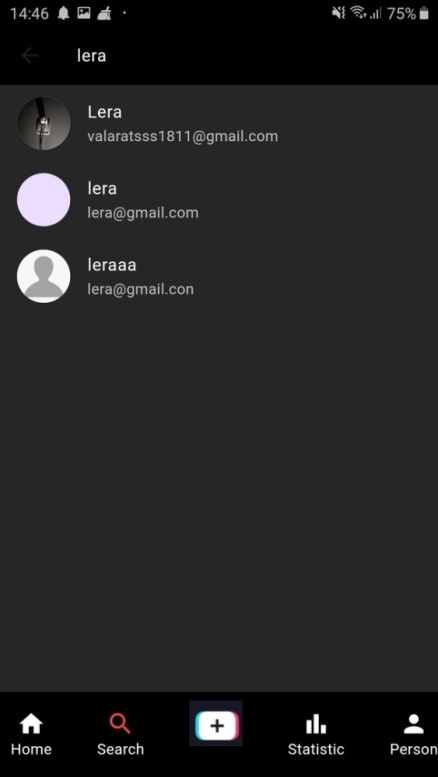


Рисунок 6.9 – Страница «Поиск»

Если пользователь нашел интересующего его пользователя, он может нажать по нему и перейти на личную страницу с отображением информации о выбранном пользователе. Также пользователь может подписаться или отписаться на другого пользователя нажав на соответствующую кнопку посреди страницы Данная страница продемонстрирована на рисунке 6.10

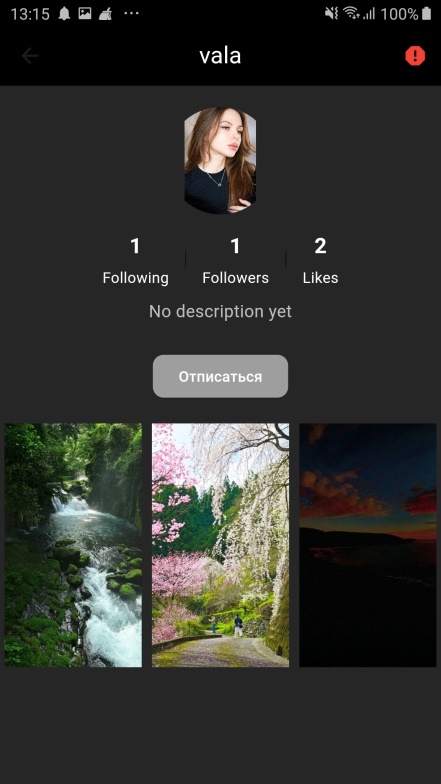


Рисунок 6.10 – Страница пользователя

Также как и с комментариями, текущий пользователь может пожаловаться на другого пользователя, нажав на иконку жалобы, далее он может выбрать причину жалобы, затем отправить ее. Данный функционал представлен на рисунке 6.11.

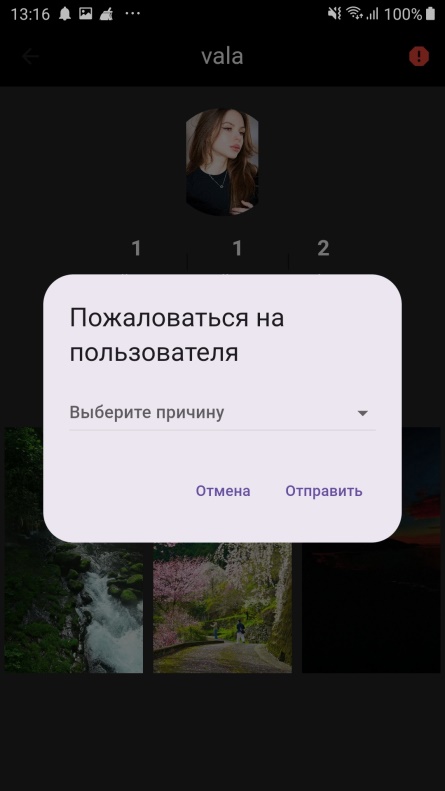


Рисунок 6.11 – Демонстрация функционала жалобы на пользователя

На странице пользователя можно увидеть видео, которые он публиковал. Для просмотра видео текущий пользователь может нажать на интересующее его видео, затем это видео отобразиться у него на весь экран с подробной информацией такой как описание, лайки и комментарии. Данная страница с видео продемонстрирована на рисунке 6.12.



Рисунок 6.12 – Страница видео

При желании пользователь может поделиться видео, нажав на троеточие, а затем нажав на «Поделиться». Данный функционал представлен на рисунке 6.13 и 6.14.

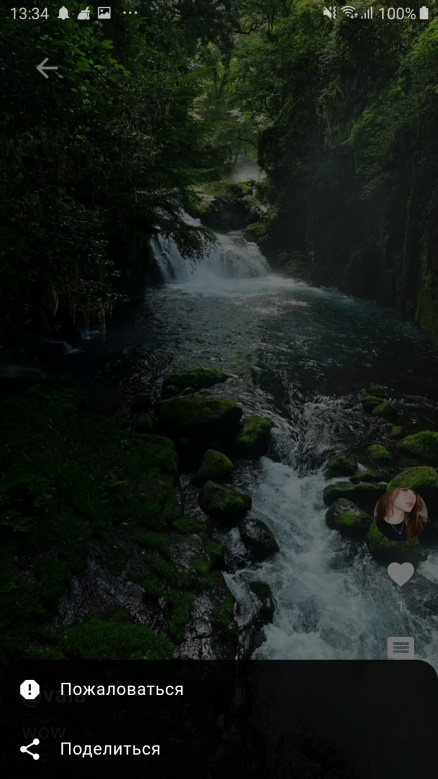


Рисунок 6.13 – Функционал «Поделиться» видео

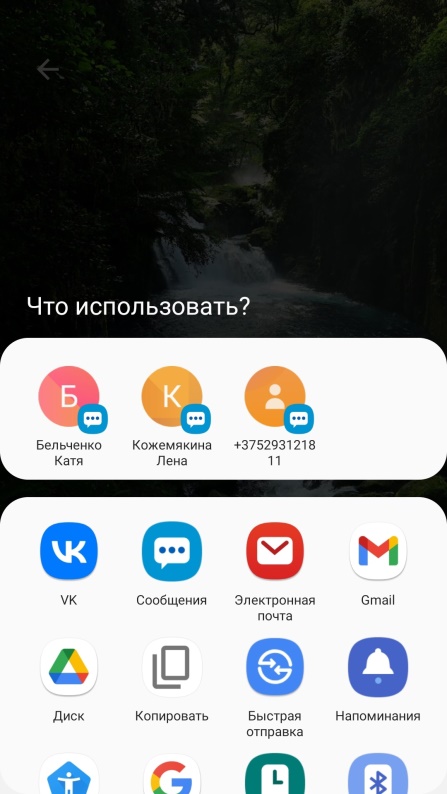


Рисунок 6.14 – Функционал «Поделиться» видео

Если пользователь захочет опубликовать свое собственное видео, он может с помощью нижнего навигационного меню переместиться на страницу добавления видео, где он может загрузить видео как с камеры, так и с телефона. Страница добавления видео продемонстрирована на рисунке 6.15.



Рисунок 6.15 – Страница добавления видео

После того как пользователь выбрал из какого ресурса он будет загружать видео, а именно с камеры или с галереи, ему требуется придумать и написать описание к видео, которое будет отображаться всем пользователям. Добавление описания к видео продемонстрировано на рисунке 6.16.

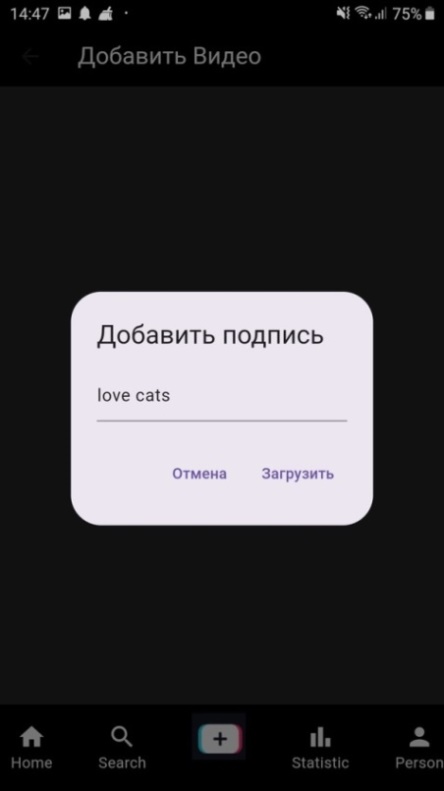


Рисунок 6.16 – Страница добавления видео

С помощью навигационного меню пользователь может перейти на свой профиль, где будет отображена информация о его профиле и видео, которые он загрузил. Страница профиля пользователя продемонстрирована на рисунке 6.17

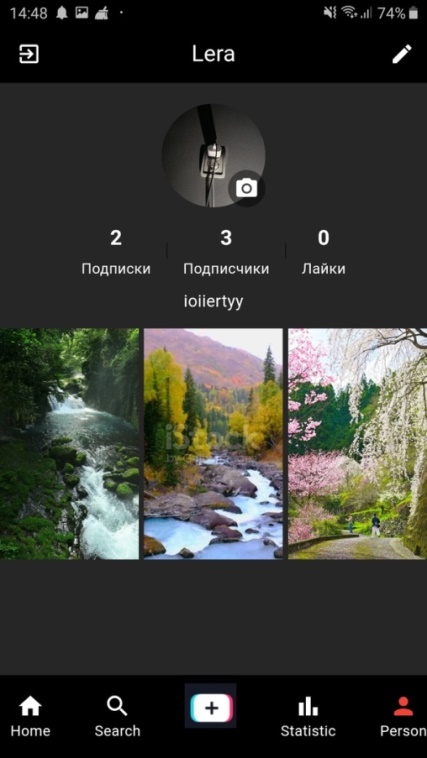


Рисунок 6.17 – Страница пользователя

При желании пользователь может изменить свое имя, описание профиля, пароль, фото профиля нажав на иконку с правом верхнем углу. Перед ним откроется форма для ввода новых данных. Данная форма продемонстрирована на рисунке 6.18.

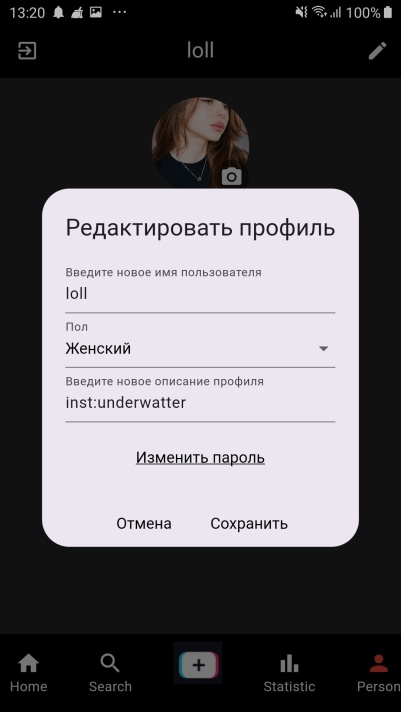


Рисунок 6.18 – Форма для ввода новых данных

Также пользователь при желании изменить пароль может нажать на «Изменить пароль» и перед ним откроется форма для изменения пароля, которая продемонстрирована на рисунке 6.19.

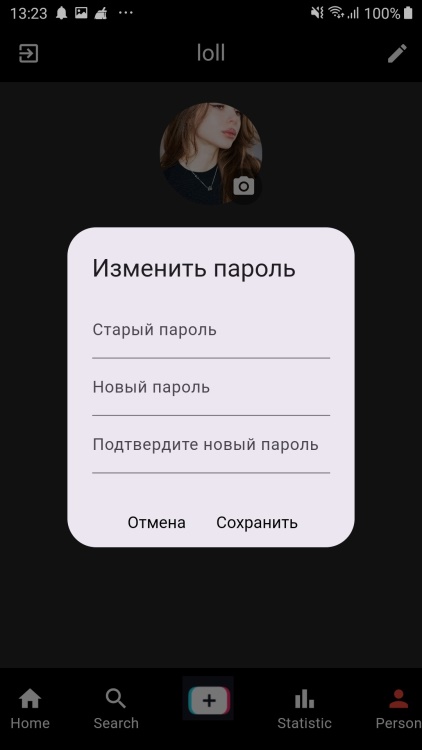


Рисунок 6.19 – Форма для изменения пароля

На странице пользователя расположено число подписчиков и подписок. При нажатии на соответствующие числа подписок/ подписчиков, пользователь перенаправляется на страницу, где отображаются аккаунты, на которые текущий пользователь подписан. Данная страница продемонстрирована на рисунке 6.20. Аналогичная страница есть и числа подписчиков.

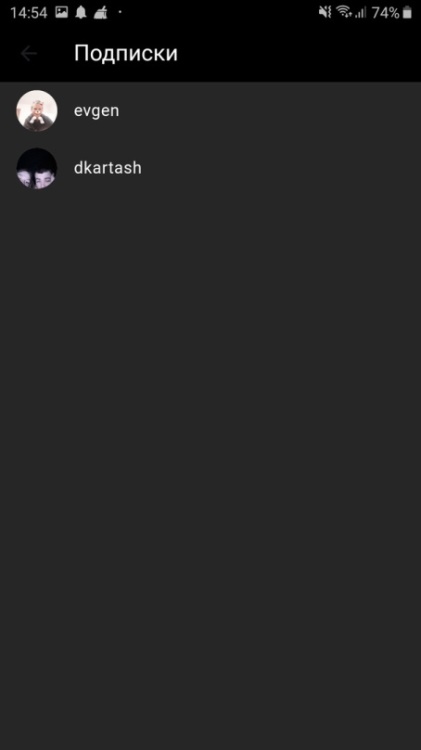


Рисунок 6.20 – Страница с подписками пользователя

Также пользователь может с помощью навигационного меню перейти на страницу статистики, где отображено самое популярное видео данного пользователя, также отображено количество пользователей, подписавшееся на него за день/неделю/месяц, а также круговая диаграмма которая отображает статистику гендера подписчиков. Страница статистики продемонстрирована на рисунках 6.21 и 6.22.

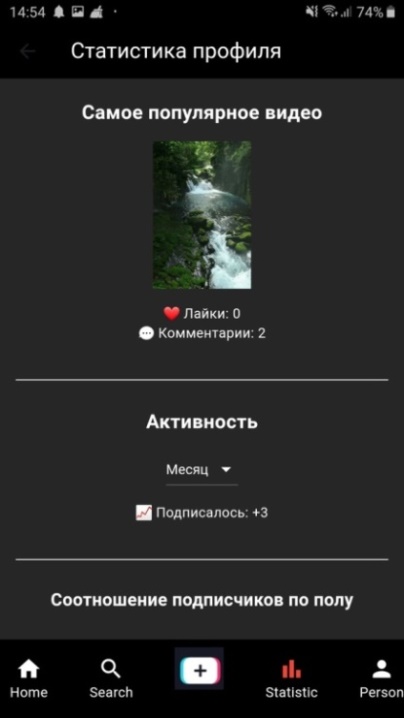


Рисунок 6.21 – Страница статистики

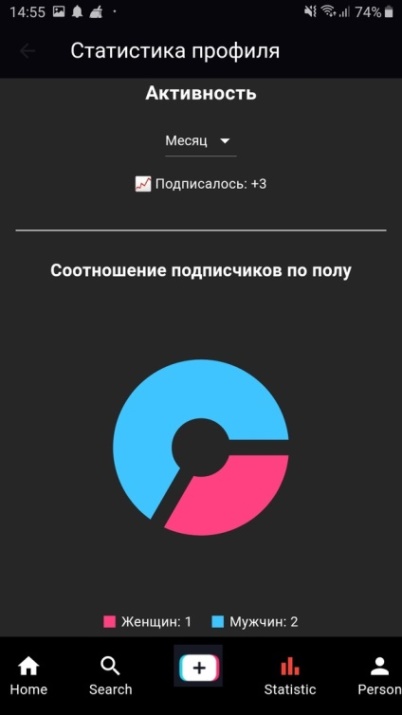


Рисунок 6.22 – Страница статистики

Для того чтобы пользователю выйти из учетной записи ему нужно нажать на иконку в левом верхнем углу на странице пользователя.

## **6.2 Администратор**

Курсовой проект разрабатывался с использованием фреймворка Flutter, что позволяет приложению быть кроссплатформенным. Для администратора была разработана веб-часть. Чтобы продолжить работу в приложении как администратор, пользователь должен иметь специальную роль и войти в приложение под соответствующей учетной записью. Пользователь с такой ролью имеет доступ к ленте с видео также, как и обычный пользователь, за исключением возможности удалять любое видео. Страница администратора с лентой видео продемонстрирована на рисунке 6.23.

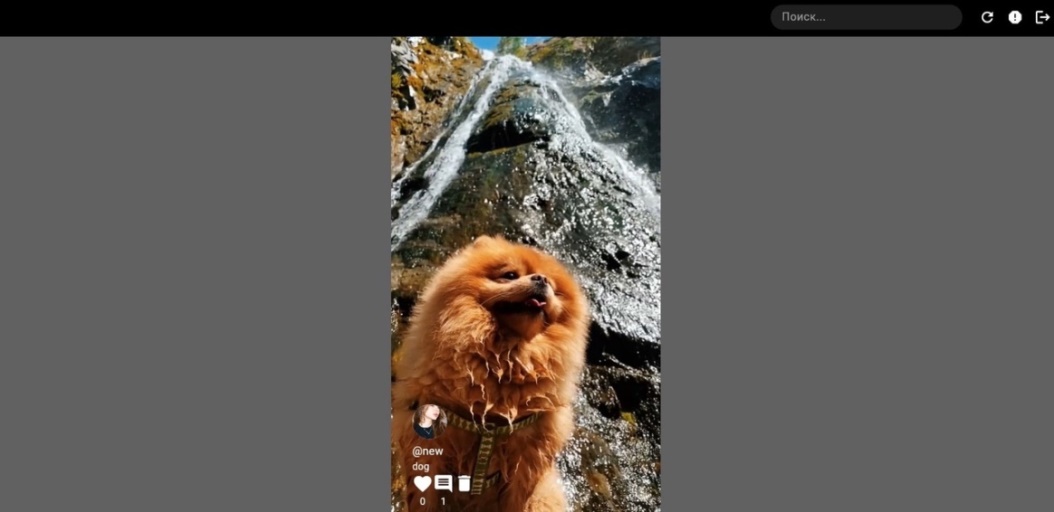


Рисунок 6.23 – Страница ленты видео для администратора

Вверху главной страницы находится меню, с помощью которого модно осуществлять поиск пользователей, иконка для перехода на страницу жалоб, а также иконка для выхода из учетной записи.

Для осуществления поиска администратору нужно ввести имя пользователя в текстовое поле для поиска. Функционал продемонстрирован на рисунке 6.24.

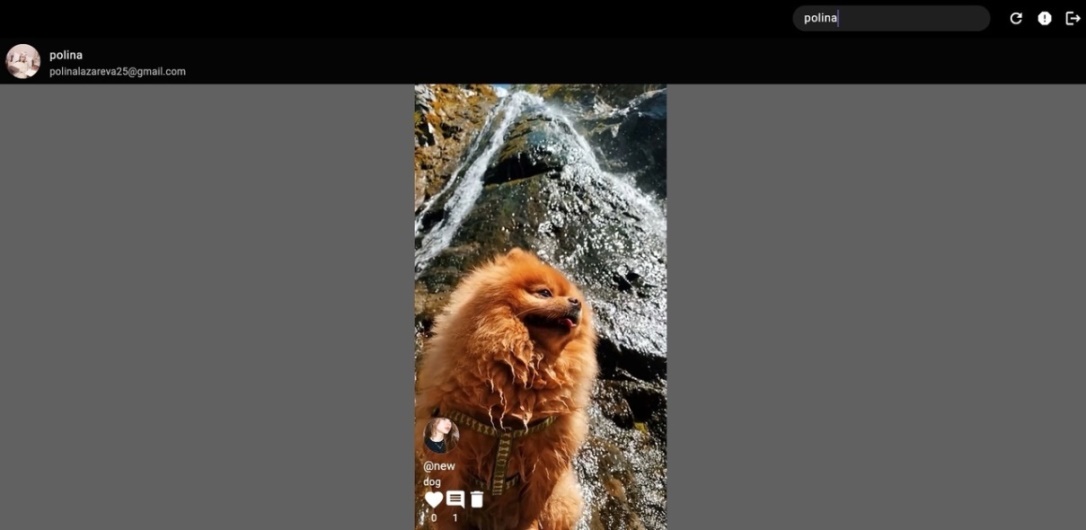


Рисунок 6.24 – Функционал поиска для администратора

При нажатии на иконку жалоб в верхнем меню администратор перенаправляется на страницу жалоб, где жалобы рассортированы по категориям, а именно жалобы на пользователей, на видео и на комментарии. В жалобах указано кто пожаловался, на кого пожаловались, когда пожаловались и по какой причине жалоба была отправлена. Также администратор может удалить любую жалобу, нажав на иконку корзины. Данная страница продемонстрирована на рисунках 6.25, 6.26 и 6.27.

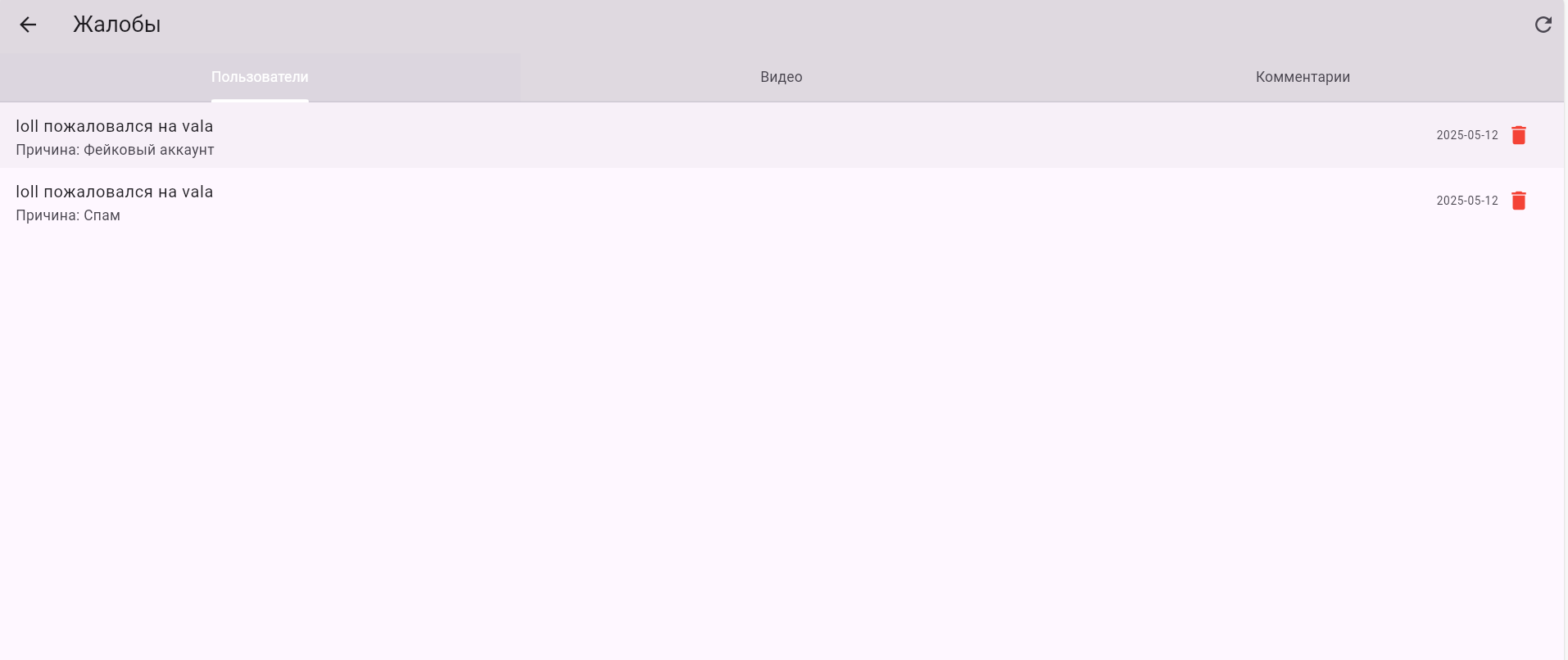


Рисунок 6.25 – Страница жалоб категории пользователи для администратора

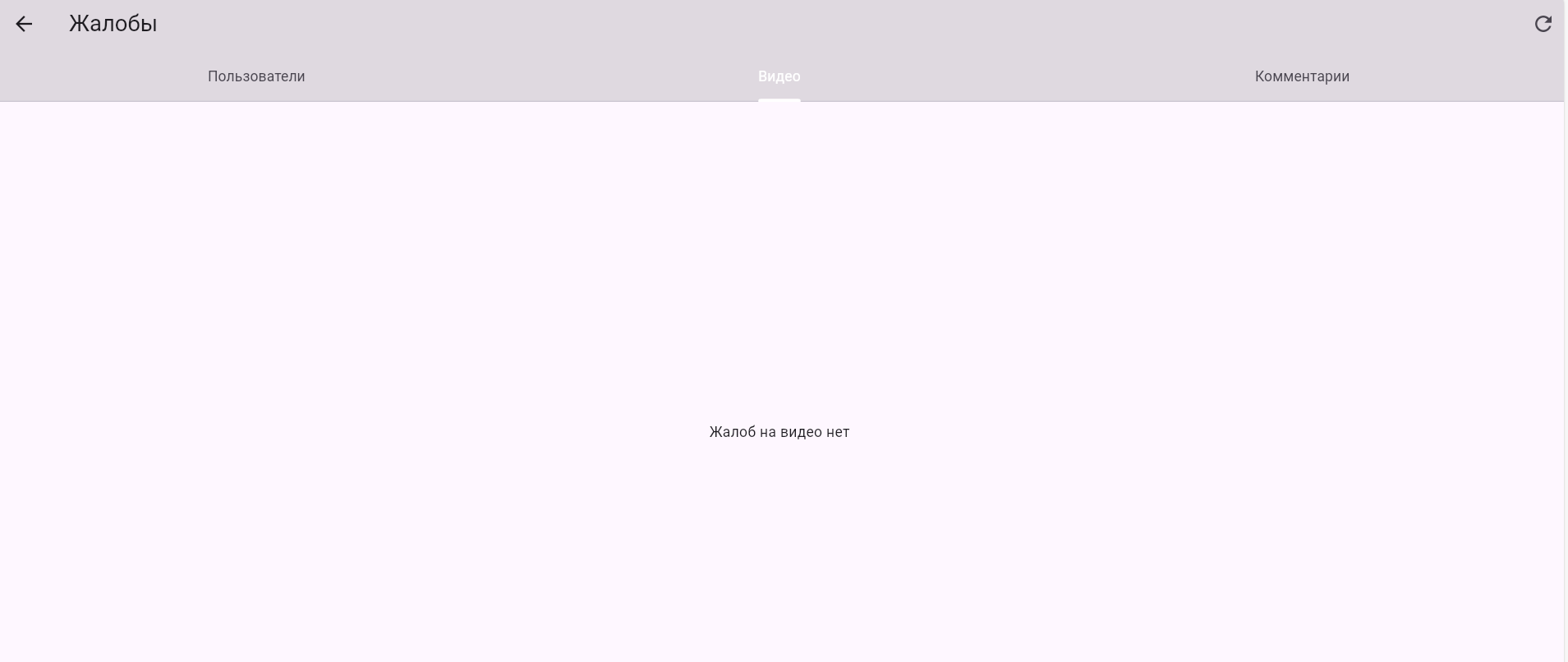


Рисунок 6.26 – Страница жалоб категории видео для администратора

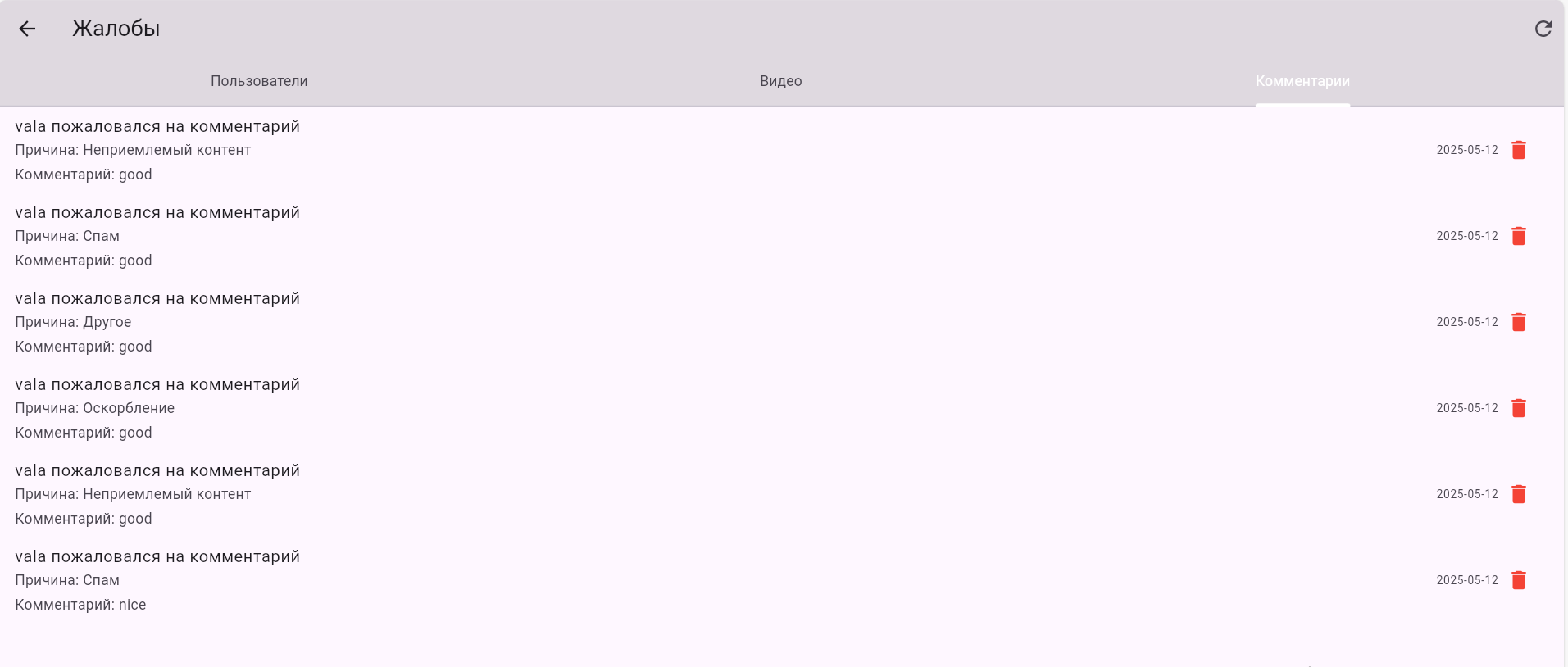


Рисунок 6.27 – Страница жалоб категории комментарии для администратора

При нажатии администратором на жалобу, он перенаправляется к соответствующем материалу, на который пожаловались, к примеру при жалобе на пользователя администратор перенаправляется на страницу пользователя, где при необходимости, может его удалить. Страница пользователя для администратора продемонстрирована на рисунке 6.28.

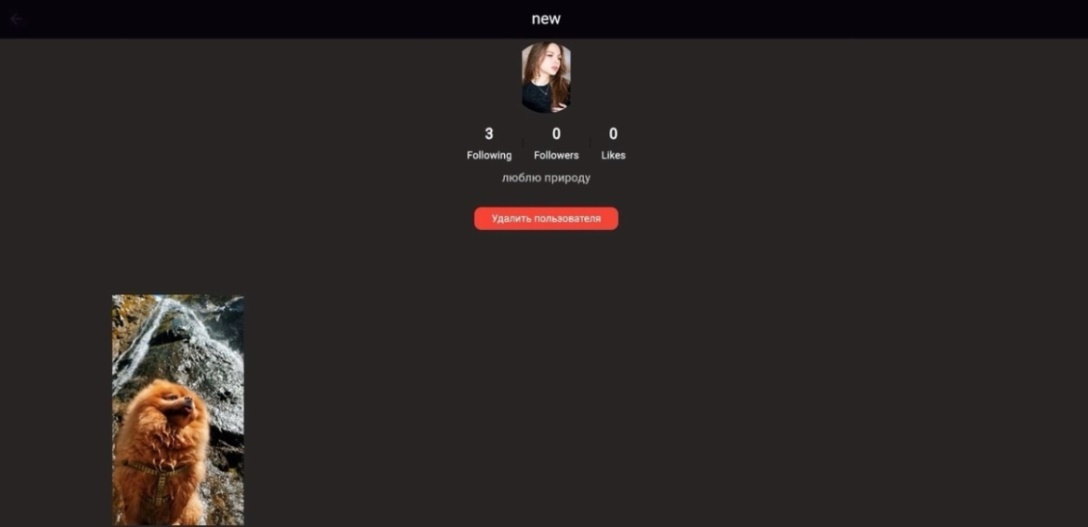


Рисунок 6.28 – Страница пользователя для администратора

Для выхода из учетной записи администратору нужно нажать на иконку выхода на главной странице.

## **6.3 Вывод**

Цель данного раздела заключалась в создании руководства пользователя, соответствующего реализованному функционалу мобильного и веб-приложения. В разделе были подробно рассмотрены ключевые аспекты взаимодействия пользователя и администратора с интерфейсом приложения.

Было представлено пошаговое описание процесса регистрации нового пользователя, включая все необходимые этапы – от ввода имени до создания безопасного пароля.

После регистрации и авторизации пользователь получает доступ ко всем функциям, включая просмотр и публикацию видео, комментирование, лайки, подписки, жалобы на контент, а также управление своим профилем. Особое внимание уделено интерактивности и удобству использования – реализована система жалоб на комментарии, видео и пользователей.

Навигация между основными экранами интуитивно понятна благодаря нижнему навигационному меню.

Также реализован раздел статистики, предоставляющий пользователю аналитику по популярности его контента, числу новых подписчиков и распределению по полу. Это позволяет автору лучше понимать свою аудиторию и адаптировать свой контент под её интересы.

Для пользователей с административными правами разработана отдельная веб-часть, которая предоставляет расширенные возможности управления системой: просмотр и удаление видео, пользователей и комментариев, а также обработка жалоб с возможностью перехода к объекту жалобы и принятия решений по нему.

# **Заключение**

В ходе курсового проекта было разработано мобильное приложение для просмотра коротких видеороликов.

Перед началом разработки были выявлены функциональные задачи мобильного приложения, рассмотрена актуальность данной темы, а также был произведен анализ аналогов приложений подобной тематики.

Функционально выполнены следующие задачи:

− определение ролей (администратор, пользователь);

− добавление видео;

− возможность комментировать видео и ставить лайки;

− поиск пользователей;

− подписка на пользователей;

− просмотр статистики профиля.

На следующем этапе разработки было выполнено проектирование, а именно реализована диаграмма вариантов использования (use-case diagram), которая описывает, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. Также, с использование платформы Supabase, была разработана база данных, состоящая из 6 таблиц. Для данной базы данных также была создана логическая схема, демонстрирующая связи между таблицами.

В качестве технических средств для разработки мобильного приложения использовались бэкенд–решение Supabase и Android Studio как среда разработки. Авторизация и регистрация пользователей осуществляется с помощью встроенных функций, которые предоставляет Supabase.

В ходе разработки мобильного приложения был проведен анализ информационной безопасности разрабатываемого мобильного приложения. Были выявлены ключевые аспекты, которые необходимо учесть при разработке приложения: защита данных, проверка на уязвимости, регулярное обновление мобильного приложения и ограничение прав доступа пользователей. Были представлены методы защиты пользовательских данных с использованием алгоритма bcrypt, который обеспечивает надежное хеширование паролей пользователей.

Неотъемлемой частью является валидация данных приложения для просмотра коротких видеороликов. Были рассмотрены сценарии регистрации и авторизации, при которых пользователь предоставляет недостаточно данных или эти данные указаны в неверном формате.

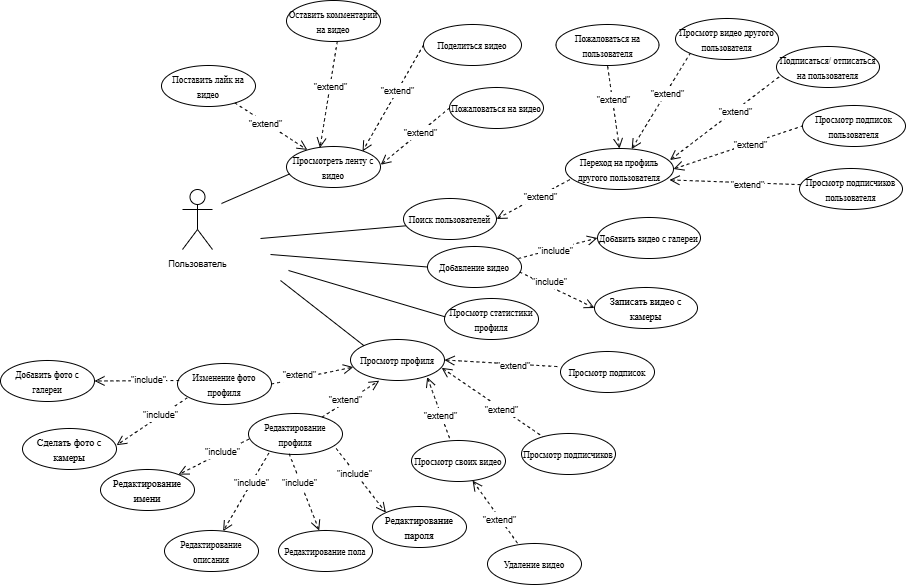
Также было рассмотрено руководство для пользователя и администратора, где подробно рассматриваются сценарии использования мобильного приложения для просмотра коротких видео.

В ходе курсового проектирования были закреплены, расширены и углублены полученные теоретические знания в области мобильной разработки с использованием фреймфорка Flutter и платформы Supabase, приобретены практические навыки самостоятельной работы, выработаны умения применять их при решении конкретных вопросов и задач.

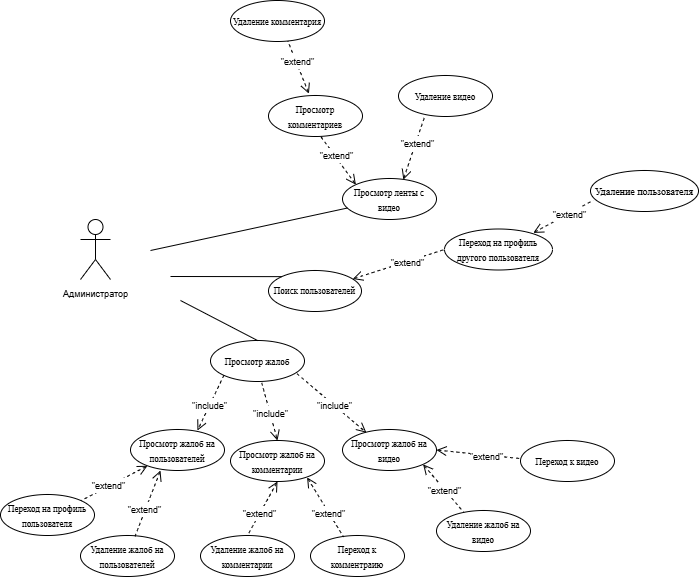
# **Список используемых источников**

1. Приложение TikTok [Электронный ресурс]. – Режим доступа https://apps.apple.com/by/app/tiktok-videos-music-live/id835599320 – Дата доступа: 10.03.2025.
2. Приложение Likee [Электронный ресурс]. – Режим доступа https://apps.apple.com/by/app/likee-video-live-chat/id1251790681– Дата доступа: 10.03.2025.
3. Документация Supabase [Электронный ресурс]. – Режим доступа: https://supabase.com/docs – Дата доступа: 14.03.2025.
4. Документация Flutter [Электронный ресурс]. – Режим доступа: https://docs.flutter.dev/– Дата доступа: 14.03.2025.
5. Документация BCrypt [Электронный ресурс]. – Режим доступа: https://pub.dev/documentation/bcrypt/latest/– Дата доступа: 10.04.2025.

ПРИЛОЖЕНИЕ А: Диаграмма вариантов использования пользователя



ПРИЛОЖЕНИЕ Б: Диаграмма вариантов использования администратора



ПРИЛОЖЕНИЕ В: Схема базы данных

