**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«Чувашский государственный университет имени И.Н. Ульянова»**

**(ФГБОУ ВО «ЧГУ им. И.Н. Ульянова»)**

**Факультет информатики и вычислительной техники**

**Кафедра вычислительной техники**

ОТЧЕТ

О ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

(ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКЕ)

на базе Общества с ограниченной ответственностью «Бокус»

|  |  |  |
| --- | --- | --- |
| Обучающийся 3 курса, направление подготовки «Информатика и вычислительная техника», группа ИВТ-41-20 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, дата | Московкина Е.А. |
|  |  |
|  | | |
| Руководитель,  ст. преподаватель кафедры  вычислительной техники | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, дата | Юрьева Е.В. |
|  |  |
|  |  |  |
| Руководитель от профильной организации,  исполнительный директор | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, дата | Иванов Д.С. |
|  |  |
|  |  |  |
| Заведующий кафедрой  вычислительной техники,  к.п.н., доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, дата | Щипцова А.В. |
|  |  |
|  |  |  |

Чебоксары 2023

СОДЕРЖАНИЕ

[1 Теоретическая часть 6](#_Toc171367494)

[1.1 Информация о деятельности организации 6](#_Toc171367495)

[1.2 Информация о средствах разработки 6](#_Toc171367496)

[1.2.1 Средство кроссплатформенной разработки Flutter 6](#_Toc171367497)

[1.2.2 Язык программирования Dart 8](#_Toc171367498)

[1.2.3 Среда разработки Visual Studio Code 9](#_Toc171367499)

[1.2.4 Эмулятор Android Studio 10](#_Toc171367500)

[1.2.5 Вспомогательные библиотеки 11](#_Toc171367501)

[2 Практическая часть 14](#_Toc171367502)

[2.1 Техническое задание 14](#_Toc171367503)

[2.2 Анализ поставленной задачи и выбор пути ее решения 16](#_Toc171367504)

[2.3 Экраны 1.1 и 1.2. Авторизация и Регистрация 17](#_Toc171367505)

[2.4 Экран 2. Главный 18](#_Toc171367506)

[2.5 Экран 4. С детальной информацией 20](#_Toc171367507)

[2.6 Экран 3. Поиск 23](#_Toc171367508)

[2.7 Экран 5. Избранное (коллекция) 24](#_Toc171367509)

[2.8 Экран 6. Профиль 25](#_Toc171367510)

[ЗАКЛЮЧЕНИЕ 27](#_Toc171367511)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 28](#_Toc171367512)

[ПРИЛОЖЕНИЕ А 29](#_Toc171367513)

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей работе применяются следующие сокращения и обозначения.

|  |  |
| --- | --- |
| Фреймворк | программная платформа, определяющая структуру программной системы;  программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта |
| API | Application Programming Interface — программный интерфейс приложения — описание способов взаимодействия одной компьютерной программы с другими |
| BLoC | Business Logic Component — компонент бизнес-логики — класс, отделяющий бизнес-логику приложения от пользовательского интерфейса |
| БД | База Данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных |

ВВЕДЕНИЕ

База практики – общество с ограниченной ответственностью «Бокус».

Цели практики:

– получение профессиональных умений и опыта технологической (проектно-технологической) деятельности;

– закрепление, расширение и углубление теоретических и практических знаний умений и навыков, полученных ранее при изучении дисциплин учебного плана.

Задачи практики задачи:

– ознакомление с базой практики (профильной организацией), выпускаемой продукцией, структурой исследовательских, проектно-конструкторских и проектно-технологических подразделений, их ролью, задачами и взаимосвязями с другими подразделениями;

– ознакомление с научной организацией труда в исследовательских, проектно-конструкторских и проектно-технологических подразделениях профильной организации;

– изучение технологии создания программных средств вычислительной техники и автоматизированных систем: Flutter – набор инструментов разработки программного обеспечения для создания кроссплатформенных приложений с использованием языка программирования Dart;

– приобретение и закрепление навыков проектно-технологической работы (технологии проектирования программного обеспечения, программных интерфейсов, баз данных): верстка экранов мобильного приложения, построение и управление свойствами виджетов, работа со стилями, навигацией и архитектурой, разработка стейт-менеджмента и управление жизненным циклом приложения, установка и применение библиотек для реализации основного функционала, работа с API и http-запросами, внедрение механизмов авторизации и регистрации, а также управления базой данных и обработки ошибок;

– ознакомление с технологиями и методами обеспечения и оценки качества разрабатываемого программного обеспечения;

– приобретение навыков разработки и оформления программной документации;

– оформление отчета по практике в соответствии с рекомендациями п.п. 6,7 программы практики.

Планируемый результат: приложение для создания пользовательской коллекции фильмов, включающее возможности авторизации и регистрации, просмотра различных топов, поиска и получения детальной информации о конкретном фильме, а также функции добавления или удаления фильмов из коллекции и редактирования данных пользователя на странице профиля.

# **Теоретическая часть**

* 1. Информация о деятельности организации

IT-компания "Бокус" — многопрофильная компания, специализирующаяся на разработке программных продуктов и ИТ-решений для частного бизнеса и государственных структур.

Направления деятельности:

– разработка мобильных приложений;

– создание веб-приложений (внешние и внутренние корпоративные порталы);

– системная интеграция;

– разработка программного обеспечения на заказ;

– создание сайтов.

Компания "Бокус" включена в перечень малых ИТ-предприятий, аккредитованных Министерством цифрового развития, связи и массовых коммуникаций Российской Федерации.

* 1. Информация о средствах разработки
     1. Средство кроссплатформенной разработки Flutter

Flutter — это современный инструмент для разработки кроссплатформенных мобильных приложений, созданный компанией Google. Он позволяет создавать мобильные приложений под Android и iOS, веб-приложения, а также настольные приложения под Windows, macOS и Linux, используя один и тот же код.

Flutter — фреймворк c открытым исходным кодом, разработанный на языке программирования Dart — статически типизированном языке, также созданным Google. Flutter использует Dart для создания пользовательского интерфейса и управления всеми аспектами приложения.

Преимущества Flutter:

– кроссплатформенность: с помощью Flutter можно разрабатывать приложения для различных платформ с использованием одного и того же кода. Это значительно упрощает разработку и снижает затраты ресурсов;

– быстрая разработка: Flutter предоставляет множество готовых виджетов и инструментов для создания интерфейса, что ускоряет процесс разработки, позволяя быстро создавать приложения;

– гибкие настройки дизайна: Flutter предлагает мощные инструменты для создания красивых пользовательских интерфейсов. Можно настроить каждый аспект дизайна, чтобы приложение получилось уникальным;

– высокая производительность: Flutter использует собственный движок для рендеринга, что обеспечивает высокую производительность и плавную анимацию даже на устаревших устройствах;

– сообщество и поддержка: у Flutter активное сообщество разработчиков, которое предоставляет множество ресурсов и библиотек для облегчения разработки.

Основные принципы работы с Flutter:

– виджеты: основной строительный блок приложения в Flutter — это виджет. Все в Flutter является виджетами, даже само приложение и его компоненты. Виджеты могут быть составными или примитивными и представляют собой элементы пользовательского интерфейса — кнопки, текстовые поля, изображения и многое другое. При этом виджеты можно комбинировать, чтобы создавать более сложные пользовательские интерфейсы;

– горячий перезапуск: Flutter предоставляет функцию горячего перезапуска, которая позволяет мгновенно отображать в приложении все вносимые в код изменения, что значительно ускоряет процесс разработки;

– однонаправленный поток данных: Flutter рекомендует к использованию однонаправленный поток данных для управления состоянием приложения. Это означает, что данные всегда движутся от верхнего уровня к нижнему, что делает управление состоянием более предсказуемым и легко поддерживаемым;

– компиляция в нативный код: Flutter компилирует приложение в нативный код для каждой платформы, что обеспечивает высокую производительность и нативный вид приложения на каждой из платформ.

* + 1. Язык программирования Dart

Dart — это язык программирования, который был разработан компанией Google и является одним из ключевых инструментов для создания приложений, основанных на фреймворке Flutter.

Dart — статически типизированный язык программирования, который объединяет в себе множество инновационных концепций. Он был создан с целью улучшения производительности и разработки мобильных, настольных и веб-приложений. Dart поддерживает как клиентскую, так и серверную разработку, что делает его полезным и универсальным инструментом.

Преимущества Dart:

– статическая типизация: Dart предоставляет статическую типизацию, что позволяет выявлять ошибки на ранних этапах разработки и улучшает читаемость кода;

– эффективная сборка мусора: Dart использует сборку мусора, упрощая управление памятью и предотвращая утечки памяти;

– многозадачность: Dart поддерживает асинхронное программирование, что позволяет создавать быстродействующие приложения и обрабатывать множество запросов одновременно;

– простой синтаксис: синтаксис Dart интуитивно понятен и легко читаем, благодаря чему язык является доступным для начинающих программистов;

– обширная стандартная библиотека: Dart имеет стандартную библиотеку, включающую множество готовых инструментов и классов для решения различных задач.

Основные принципы работы с Dart:

– статическая типизация: одной из ключевых особенностей Dart является статическая типизация. Это означает, что каждая переменная и каждое значение имеют определенный тип данных, который проверяется на этапе компиляции. Это помогает предотвратить множество ошибок на ранних этапах разработки и улучшает надежность кода;

– асинхронное программирование: Dart предоставляет мощные инструменты для асинхронного программирования. Это особенно полезно при работе с сетью или вводом-выводом;

– классы и объекты: Dart поддерживает объектно-ориентированное программирование, что позволяет создавать классы и объекты, определять для них методы и поля и использовать наследование и интерфейсы;

– стрелочные функции: Dart позволяет использовать стрелочные функции, что делает код более кратким и читаемым, особенно при работе с функциональным программированием.

* + 1. Среда разработки Visual Studio Code

Visual Studio Code (VS Code) — это интегрированная среда разработки, разработанная компанией Microsoft, которая на данный момент является одним из наиболее популярных инструментов для создания приложений на Flutter/Dart.

VS Code — это бесплатная, с открытым исходным кодом, кроссплатформенная IDE, предназначенная для разработки различных типов приложений. Она предоставляет обширный набор функциональных возможностей, расширений и инструментов для удобной и эффективной разработки.

Преимуществами VS Code при работе с Flutter/Dart:

– расширения: VS Code предоставляет множество расширений, созданных сообществом и разработчиками, специализирующимися для работы с Flutter/Dart;

– интегрированная отладка: VS Code обеспечивает удобную интегрированную среду для отладки приложений. Она позволяет устанавливать точки останова, отслеживать переменные и мгновенно видеть результаты изменений в коде;

– автодополнение и подсказки: встроенное автодополнение и подсказки в VS Code значительно упрощают процесс написания кода на Flutter/Dart. IDE предлагает шаблоны для написания определенных фрагментов кода, что позволяет сэкономить много времени и предотвратить ошибки;

– интеграция с Git: VS Code интегрирован с системой контроля версий Git, что позволяет управлять и отслеживать изменения в коде.

* + 1. Эмулятор Android Studio

При разработке мобильных приложений на Flutter/Dart, как правило, используется несколько инструментов для создания, тестирования и отладки. Одним из важных компонентов этого процесса является среда разработки Android Studio и предоставляемые ей эмуляторы Android.

Android Studio — это интегрированная среда разработки, предназначенная, в первую очередь, для создания приложений под платформу Android. Однако при работе с Flutter/Dart, Android Studio может служить инструментом для обеспечения эффективной разработки и тестирования.

Одной из наиболее ценных функций Android Studio является возможность создавать и управлять виртуальными эмуляторами – смартфонами и планшетами с разными версиями Android. Они позволяют запускать и тестировать Flutter-приложение в режиме реального времени.

При этом основной IDE для разработки приложения на Flutter/Dart может быть Visual Studio Code (VS Code), а Android Studio легко интегрироваться в этот процесс. Таким образом, разработка и отладка осуществляются в VS Code, а Android Studio используется для создания и запуска эмуляторов, которые обеспечивают удобное тестирование мобильного приложения.

* + 1. Вспомогательные библиотеки

Библиотеки Flutter-проекта играют важную роль в разработке. Они представляют собой совокупность предопределённых функций, классов и компонентов, которые могут быть использованы для упрощения и ускорения процесса создания приложений.

Используемые вспомогательные библиотеки:

– cupertino\_icons: пакет иконок, предназначенных для использования в приложениях, разрабатываемых для платформы Flutter, в стиле, вдохновленном дизайном операционной системы iOS;

– fluttertoast: пакет, который позволяет показывать в приложении всплывающие уведомления — небольшие информационные сообщения, которые появляются на экране и автоматически исчезают через короткий период времени;

– ndialog: пакет, предназначенный для упрощения создания и управления в приложениях диалоговыми окнами — всплывающими интерфейсными элементами, которые используются для взаимодействия с пользователем, отображения информации или запроса подтверждения для различных действий;

– intl: пакет, предназначенный для обработки международной локализации и форматирования текста в приложении. Локализация — это процесс адаптации приложения к различным языкам, регионам и культурам, чтобы оно могло быть успешно использовано в разных частях мира. Данный пакет обеспечивает инструменты для перевода текста, форматирования чисел, дат и времени, а также для работы с другими аспектами, связанными с международной поддержкой;

– dio: пакет, который представляет собой библиотеку для работы с HTTP-запросами и взаимодействия с внешними веб-серверами и API. Обеспечивает разработчиков инструментами для выполнения различных сетевых операций;

– flutter\_bloc: библиотека для управления состоянием экранов в приложениях с использованием архитектурного паттерна BLoC. Этот паттерн помогает разделять бизнес-логику от пользовательского интерфейса;

– get\_it: библиотека для управления зависимостями в приложении. Она предоставляет механизмы для создания, регистрации и получения экземпляров объектов, что упрощает организацию кода, уменьшает связанность и делает приложение более легко тестируемым;

– equitable: инструмент, который упрощает сравнение объектов и определение равенства между ними. Широко используется при работе с классами, представляющими данные;

– talker\_flutter: инструмент обработки исключений и логирования для приложений Flutter/Dart. Представляет из себя систему для хранения и распространения отладочной информации приложения;

– talker\_dio\_logger: дополнение talker\_flutter для совместной работы с пакетом dio;

– talker\_bloc\_logger: дополнение talker\_flutter для совместной работы с пакетом bloc;

– path\_provider: пакет, который предоставляет инструменты для доступа к различным директориям в файловой системе устройства. Позволяет приложению получать доступ к папкам локального хранилища, кэша, документов и другим для сохранения, чтения или удаления файлов;

– image\_picker: пакет, предназначенный для упрощения выбора изображений с устройства пользователя с использованием камеры или галереи.

Следующие пакеты обеспечивают интеграцию с облачной платформой Firebase, которая предоставляет такие сервисы и инструменты для разработки мобильных и веб-приложений, как аутентификация, база данных в реальном времени, хранение файлов и другие функции:

– firebase\_core: пакет, необходимый для инициализации Firebase в приложении. Он обеспечивает основную инфраструктуру, которая позволяет приложению взаимодействовать с другими службами Firebase;

– firebase\_auth: пакет, предоставляющий инструменты для аутентификации пользователей. Позволяет различными методами авторизоваться в приложении, например, с использованием электронной почты и пароля, а также подключать для входа социальные сети;

– firebase\_database: пакет, позволяющий создавать базы данных в режиме реального времени;

– firebase\_storage: пакет, предоставляющий возможность загружать и хранить файлы в облаке. Можно использовать для хранения изображений, видео, аудио и других медиафайлов, а также для обмена ими между пользователями.

# **Практическая часть**

* 1. Техническое задание

Разработать приложение-проект, состоящее из нескольких экранов:

1. Экраны авторизации и регистрации. Все осуществляемые пользователем операции на этих экранах должны проходить через платформу Firebase. После успешной авторизации последующие запуски приложения должны вести на Главный экран;

2. Главный экран приложения должен содержать заголовок, списки из лучших, популярных и ожидаемых на данный момент фильмов, а также адаптивное боковое меню, обеспечивающее интуитивно понятную навигацию. Все фильмы на данном экране должны быть представлены постером, названием и годом выпуска. Меню должно включать кнопки для перехода на экран профиля, поиска и избранного (коллекции), а также для выхода из учетной записи;

3. Экран поиска, который должен содержать строку поиска и отображать список найденных фильмов по запросу пользователя. Все фильмы на данном экране должны быть представлены постером, названием, годом выпуска и списком жанров;

4. Детальная страница фильма. По нажатию на фильм с любого экрана должен происходить переход на детальную страницу фильма, где указаны постер, название, год, страны и жанры, рейтинг и прочая информация по желанию. Через экран с детальной информацией должна иметься возможность добавлять и удалять фильмы из избранного (коллекции);

5. Экран избранного должен содержать список фильмов, добавленных пользователем в коллекцию. У каждого элемента списка должна быть кнопка для удаления из списка;

6. Экран профиля, на котором должны отображаться аватар, имя и адрес электронной почты пользователя, а также кнопки для изменения соответствующих данных профиля.

Требования:

– авторизация в приложении должна происходить через платформу Firebase;

– получение всей информации о фильмах должно производится через открытое API Kinopoisk API Unofficial;

– коллекция из сохраненных фильмов должна храниться в Firebase;

– все экраны, за исключением экрана входа и главного экрана, должны иметь кнопку «назад» в шапке страницы;

– все состояния, которые записываются из веб-запросов, должны хранится в BLoC;

– для запросов необходимо использовать библиотеку Dio;

– должна присутствовать обработка исключений;

– дизайн может быть в свободной форме, прототипы экранов представлены на рисунках 1 и 2.

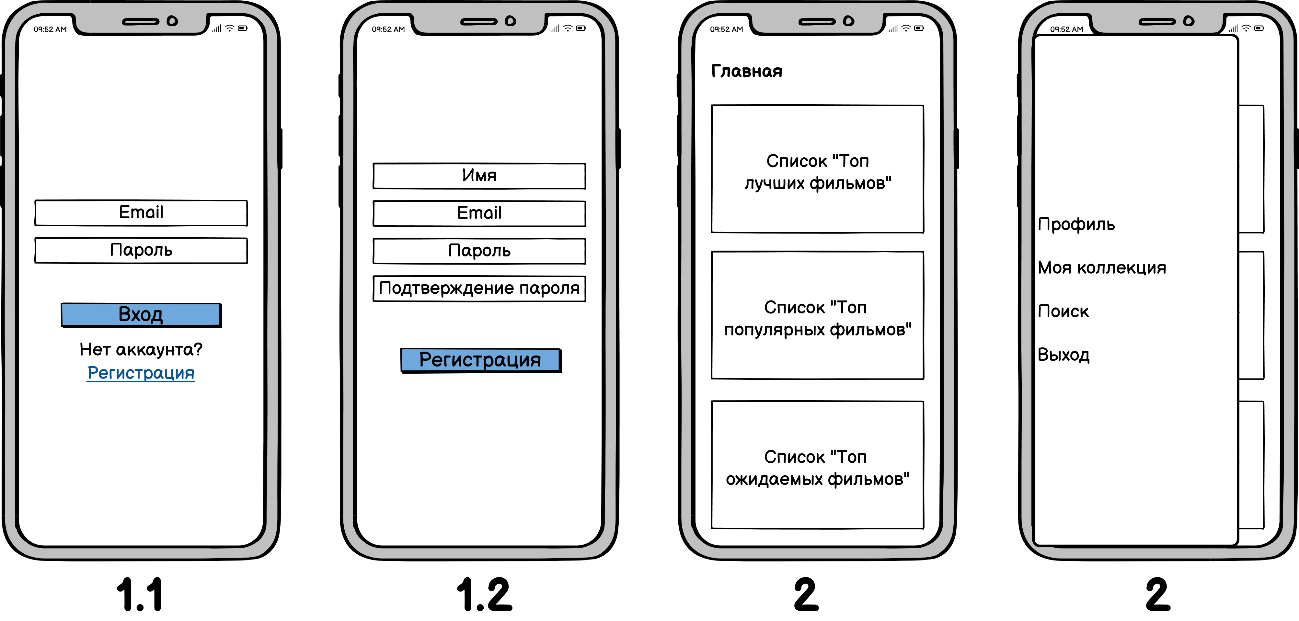


Рисунок 1 – Прототипы экранов с 1.1 по 2

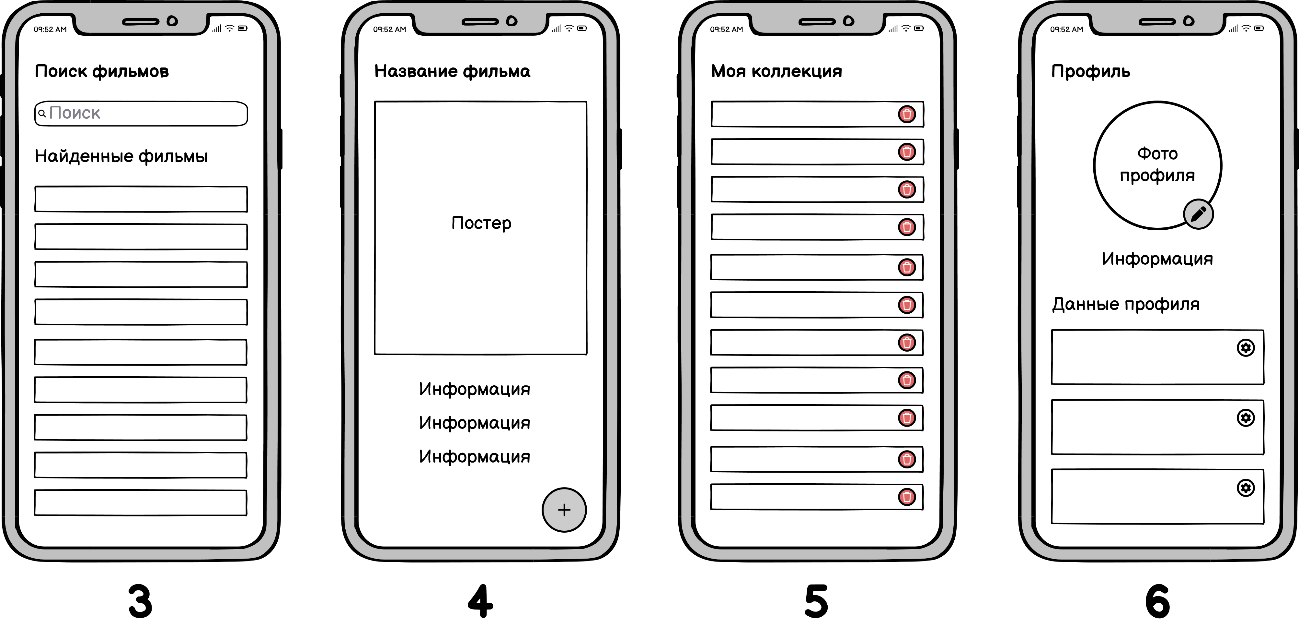


Рисунок 2 – Прототипы экранов с 3 по 6

* 1. Анализ поставленной задачи и выбор пути ее решения

На данном этапе был проведен анализ поставленного технического задания, результатом которого стал следующий план работ:

1. Верстка и реализация функционала экранов 1.1 и 1.2 – авторизации и регистрации соответственно;
2. Верстка и реализация функционала и стейт-менеджмента экрана 2 – главного;
3. Верстка и реализация функционала и стейт-менеджмента экрана 4 – с детальной информацией;
4. Верстка и реализация функционала и стейт-менеджмента экрана 3 – поиска;
5. Верстка и реализация функционала и стейт-менеджмента экрана 5 – избранного;
6. Верстка и реализация функционала и стейт-менеджмента экрана 6 – профиля.
   1. Экраны 1.1 и 1.2. Авторизация и Регистрация

Открывая приложение впервые после его установки на мобильное устройство, пользователь видит перед собой экран входа. Если ранее он уже пользовался приложением и у него имеется учетная запись, он может ввести данные в соответствующие поля и выполнить вход. В ином случае он должен перейти по ссылке на экран регистрации и завести новую учетную запись.

Оба экрана авторизации и регистрации представлены на рисунке 3.

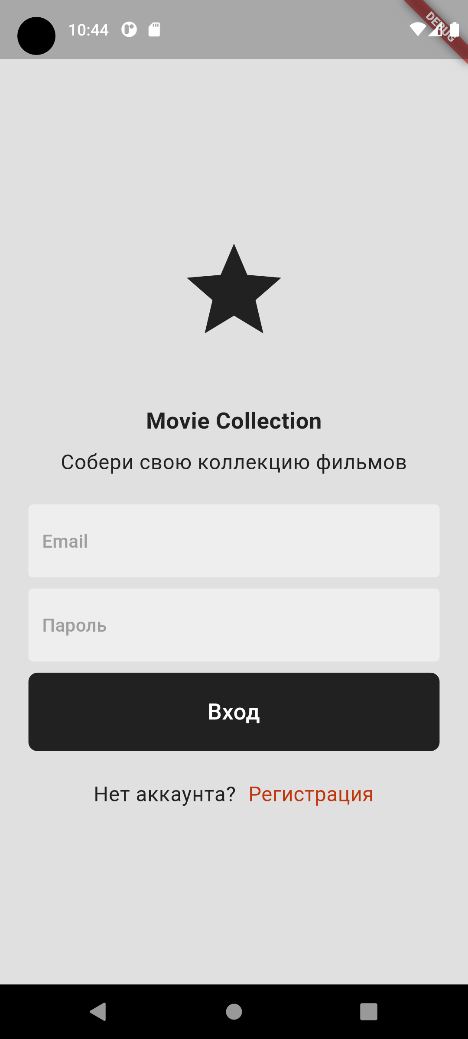
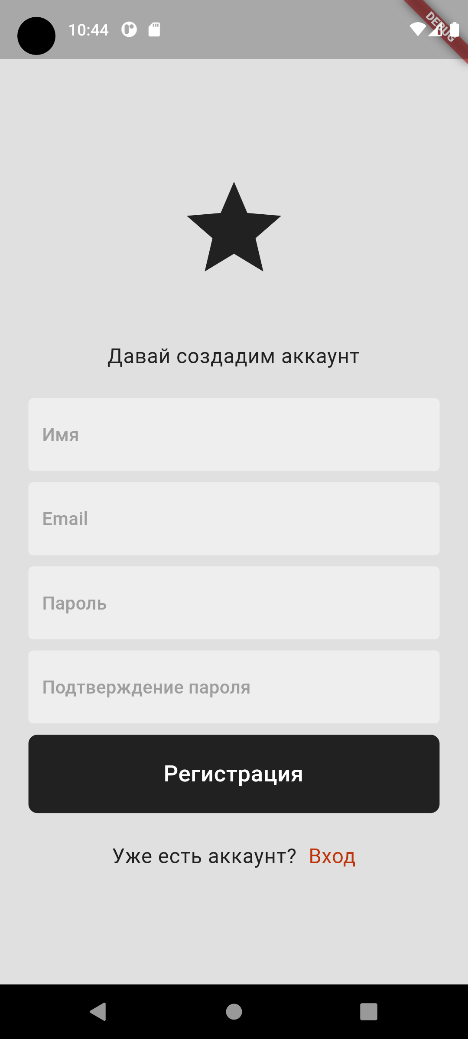
 

Рисунок 3 – Экраны авторизации и регистрации

Заполнение всех полей на каждой из страниц является обязательным. При входе осуществляется проверка на соответствие введенных данных тем, что хранятся в БД, при регистрации – на надежность и корректность выбранного пароля и уникальность адреса электронной почты. На один адрес нельзя регистрировать более одной учетной записи.

Если регистрация прошла успешно, новые данные добавляются в соответствующую таблицу БД. Далее пользователь будет возвращен на экран входа для авторизации. Если и она пройдет успешно, он попадет на главный экран, который будет открываться в первую очередь при последующих запусках приложения.

В случае ошибок и непредвиденных ситуаций на экране появится всплывающее окно с соответствующим сообщением для пользователя.

Авторизация и регистрация происходят через облачную платформу Firebase с использованием библиотек firebase\_auth и firebase\_database, которые предоставляют соответствующие инструменты для аутентификации и работы с БД.

* 1. Экран 2. Главный

Если вход в учетную запись был выполнен, при запуске приложения пользователь видит перед собой главный экран, который имеет следующие состояния:

1. Экран загружается. Переход в это состояние осуществляется автоматически при открытии экрана, за исключением случаев, когда был выполнен возврат с помощью кнопки «назад». На самом экране отображается анимированное колесо загрузки;

2. Экран загружен. Переход осуществляется, если с сервера были получены корректные и целостные данные. После этого на экране отображаются списки лучших, популярных и ожидаемых фильмов в данный период времени.

Каждый из них можно прокручивать по горизонтали до тех пор, пока не будет достигнут конец списка. При этом изначально они являются неполными и содержат ограниченное число элементов, полученных с сервера. Для последующего заполнения списков каждый был оснащен контроллером прокрутки, который активирует загрузку новых элементов, как только предыдущие были просмотрены пользователем. Когда с сервера будут получены все данные, список становится конечным.

Каждый фильм в списке представлен постером, названием и годом выпуска.

В левом верхнем углу экрана расположена кнопка, при нажатии на которую на экране отображается адаптивное боковое меню для навигации по приложению. У пользователя имеется возможность перехода на другие экраны, а также выхода из своей учетной записи.

Пример корректно загруженного главного экрана можно увидеть на рисунке 4;

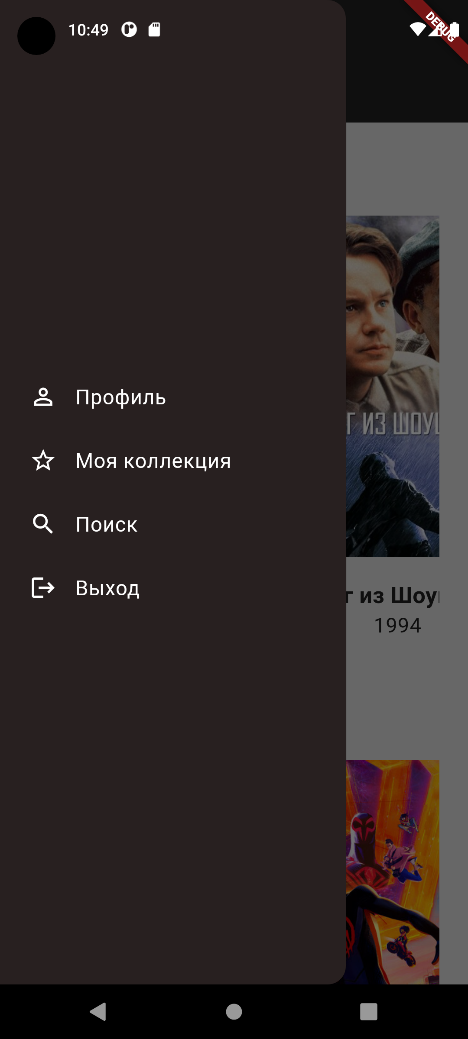
 

Рисунок 4 – Главный экран

3. Ошибка загрузки экрана. Переход осуществляется, ели при загрузке данных с сервера произошла ошибка или непредвиденная ситуация. В этом случае на экране отображается сообщение об ошибке и кнопка с возможностью повторной загрузки.

Примеры состояний, когда экран находится в процессе загрузки или во время этого процесса произошла ошибка, представлены на рисунке 5. Для других экранов эти состояния будут выглядеть аналогично.

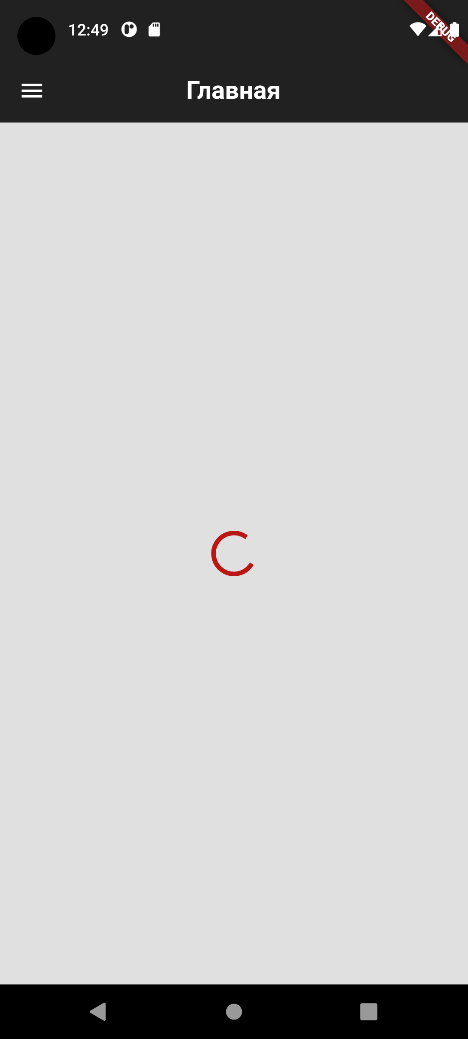
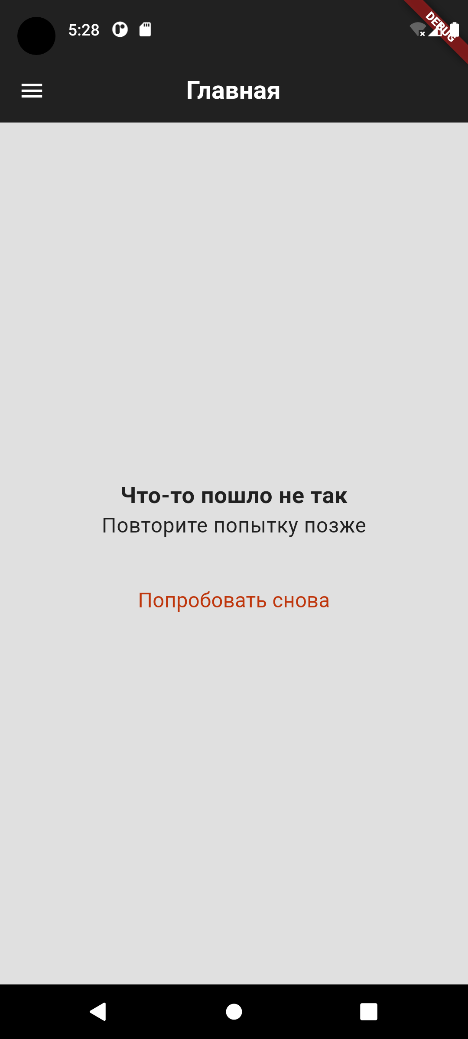
 

Рисунок 5 – Экран загружается и ошибка загрузки экрана

* 1. Экран 4. С детальной информацией

С любого экрана, содержащего виджет с фильмами, можно перейти на экран с детальной информацией о фильме по нажатию на соответствующий элемент виджета. Данный экран имеет следующие состояния:

1. Экран загружается;

2. Экран загружен. Переход в это состояние осуществляется, если с сервера были получены корректные и целостные данные. После этого на экране отображается детальная информация о выбранном пользователем фильме: постер, название на русском, название на языке оригинала, год выпуска, рейтинг, страны и жанры, а также краткая аннотация.

В правом нижнем углу экрана расположена кнопка для добавления или удаления фильма из избранного (коллекции).

Пример корректно загруженного экрана с детальной информацией можно увидеть на рисунке 6;

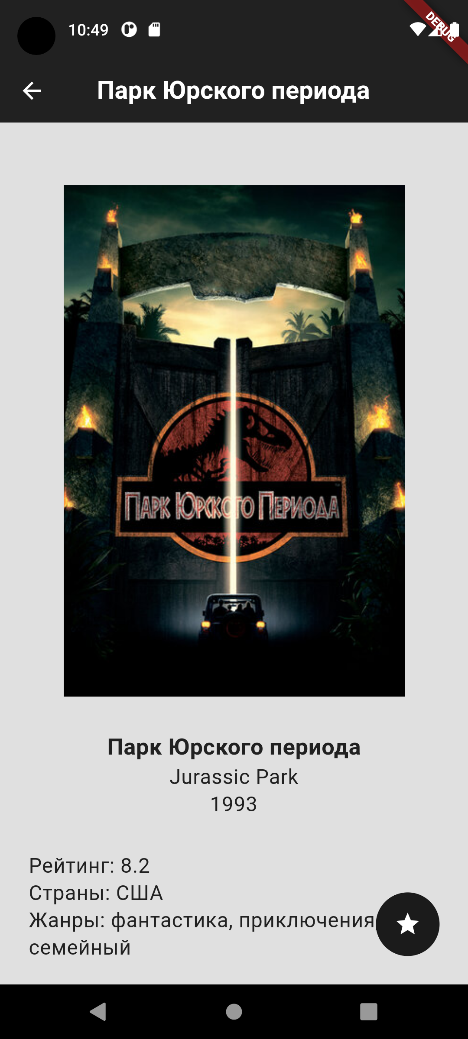
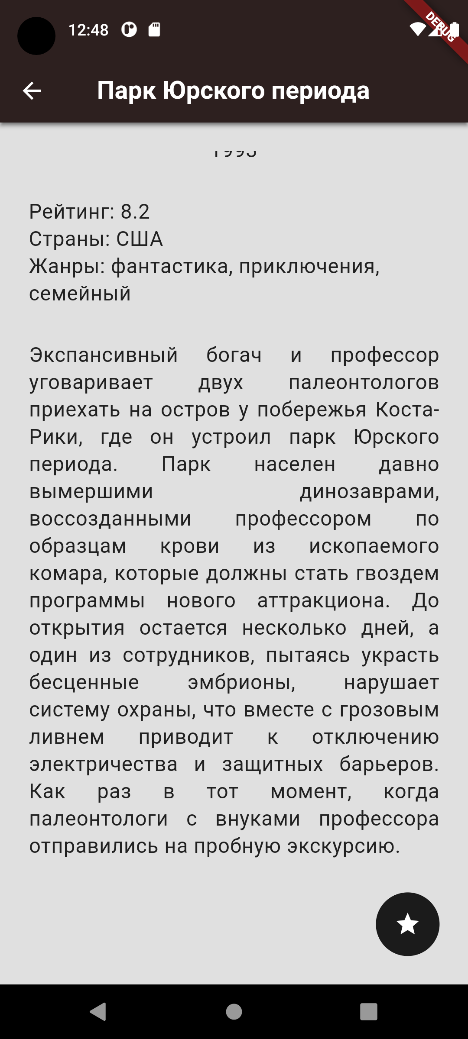
 

Рисунок 6 – Экран с детальной информацией

3. Ошибка загрузки экрана.

При нажатии на кнопку, выполняющую операцию добавления или удаление из избранного (коллекции), возможен переход экрана в следующие состояния:

1. Операция с фильмом выполняется. Переход в это состояние осуществляется сразу после нажатия на кнопку, при этом на ней отображается анимированное колесо загрузки. Пример данной кнопки представлен на рисунке 7;

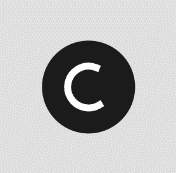


Рисунок 7 – Кнопка с анимированным колесом загрузки

2. Операция с фильмом выполнена. Переход осуществляется, если добавление или удаление фильма из БД прошло успешно, при этом на кнопке отображается соответствующая иконка, информирующая о том, находится ли фильм в избранном (коллекции) или нет. Примеры возможных вариантов представления данной кнопки изображены на рисунке 8;

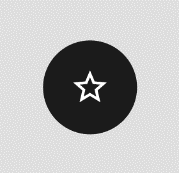
 

Рисунок 8 – Кнопки с иконками, которые меняются в зависимости от того, добавлен ли фильм в избранное (коллекцию) или нет

3. Ошибка выполнения операции с фильмом. Переход осуществляется, если при добавлении или удалении фильма из БД произошла ошибка или иная непредвиденная ситуация, при этом на кнопке отображается иконка ошибки. Пример данной кнопки представлен на рисунке 9.



Рисунок 9 – Кнопка с иконкой ошибки

* 1. Экран 3. Поиск

При нажатии на кнопку «Поиск» в меню навигации пользователя направляет на соответствующий экран, который содержит строку поиска – поле ввода, куда пользователь может ввести свой запрос, а также кнопку, выполняющую операцию поиска и активирующую переход экрана в следующие состояния:

1. Экран загружается;

2. Экран загружен. Переход в это состояние осуществляется, если с сервера были получены корректные и целостные данные. После этого на экране отображается список найденных фильмов.

Как и на главном экране, поначалу список содержит ограниченное число элементов и дополнятся по мере его прокручивания, на этот раз в горизонтальном направлении, пока с сервера не будут получены все данные.

Каждый фильм в списке представлен постером, названием, годом выпуска и списком его жанров.

Пример корректно загруженного главного экрана можно увидеть на рисунке 10;

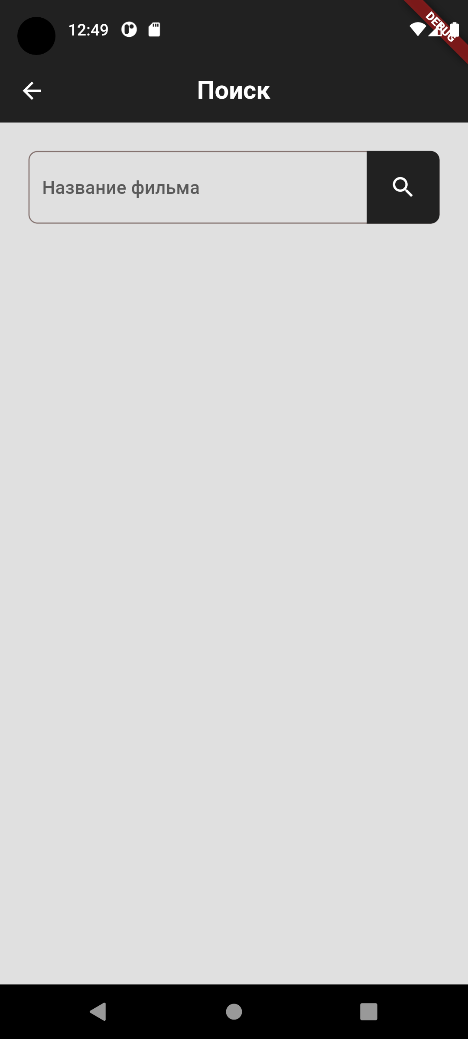
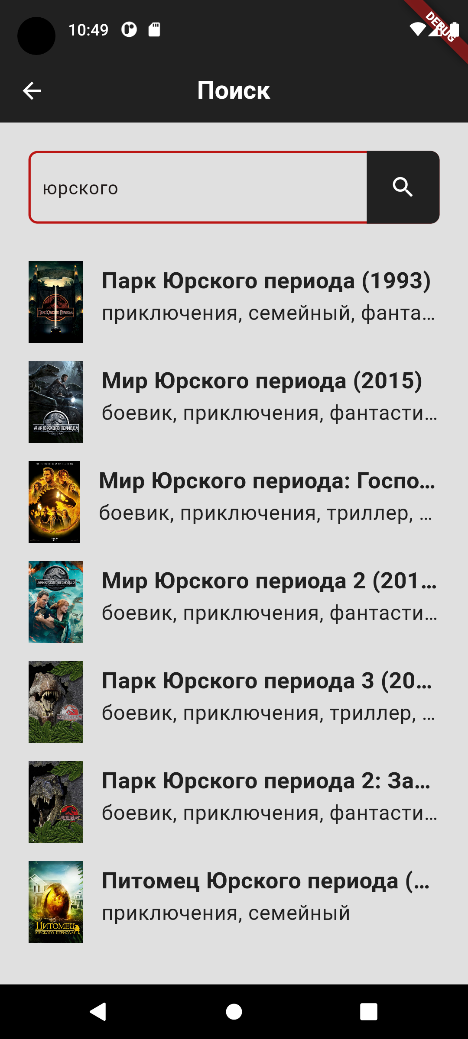
 

Рисунок 10 – Экран поиска

3. Ошибка загрузки экрана.

* 1. Экран 5. Избранное (коллекция)

При нажатии на кнопку «Моя коллекция» в меню навигации пользователя направляет на соответствующий экран, который имеет следующие состояния:

1. Экран загружается;

2. Экран загружен. Переход осуществляется, если с сервера были получены корректные и целостные данные. После этого на экране отображается список избранных пользователем фильмов.

Каждый фильм представлен постером и названием, а также имеет сопутствующую кнопку для удаления из списка.

Пример корректно загруженного главного экрана можно увидеть на рисунке 11;

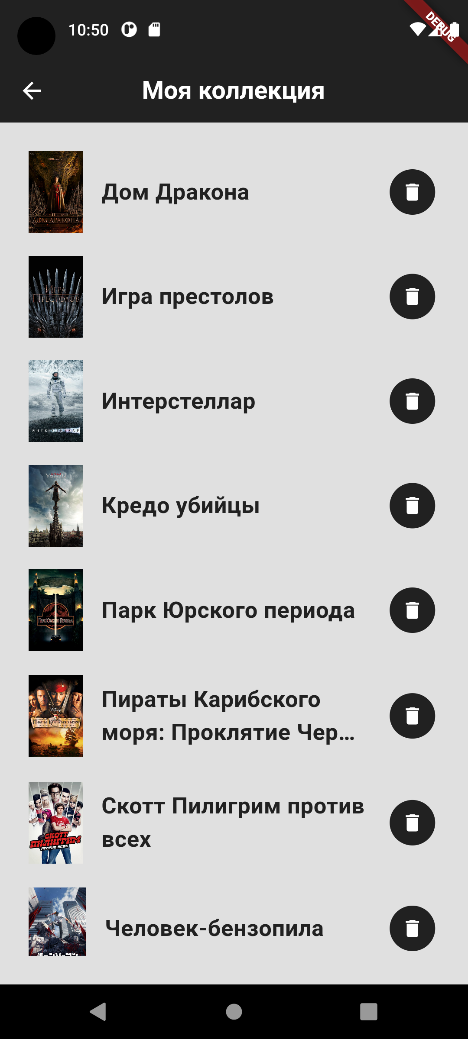


Рисунок 11 – Экран избранного (коллекции)

3. Ошибка загрузки экрана.

* 1. Экран 6. Профиль

При нажатии на кнопку «Профиль» в меню навигации пользователя направляет на соответствующий экран, который имеет следующие состояния:

1. Экран загружается;

2. Экран загружен. Переход осуществляется, если с сервера были получены корректные и целостные данные. После этого на экране отображаются регистрационные данные профиля.

В правой нижней части аватара расположена кнопка для замены изображения, при нажатии на которую открывается галерея мобильного устройства, где пользователь может выбрать новый аватар.

У графы «Имя пользователя» так же имеется сопутствующая кнопка для редактирования данных профиля. При нажатии на нее появляется всплывающее окно с полем ввода, где пользователь может указать новое имя и сохранить внесенные изменения.

Пример корректно загруженного главного экрана можно увидеть на рисунке 12;

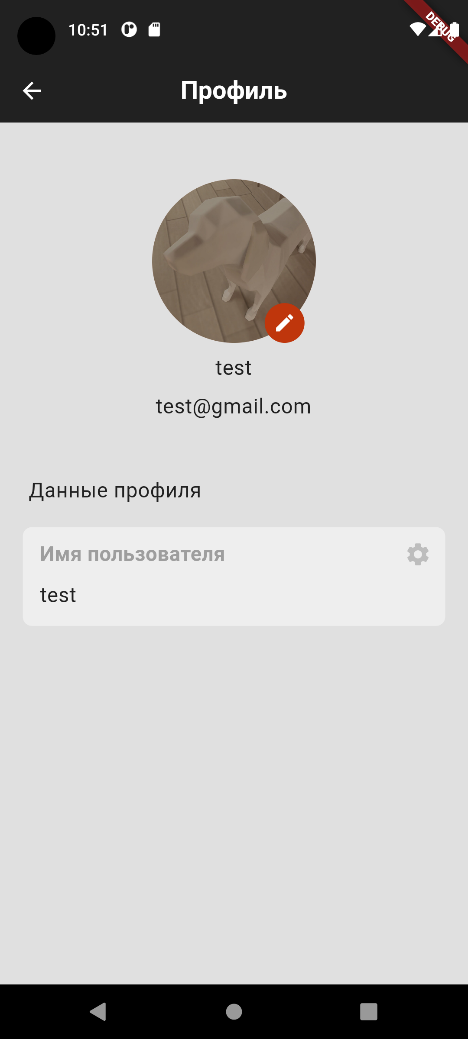
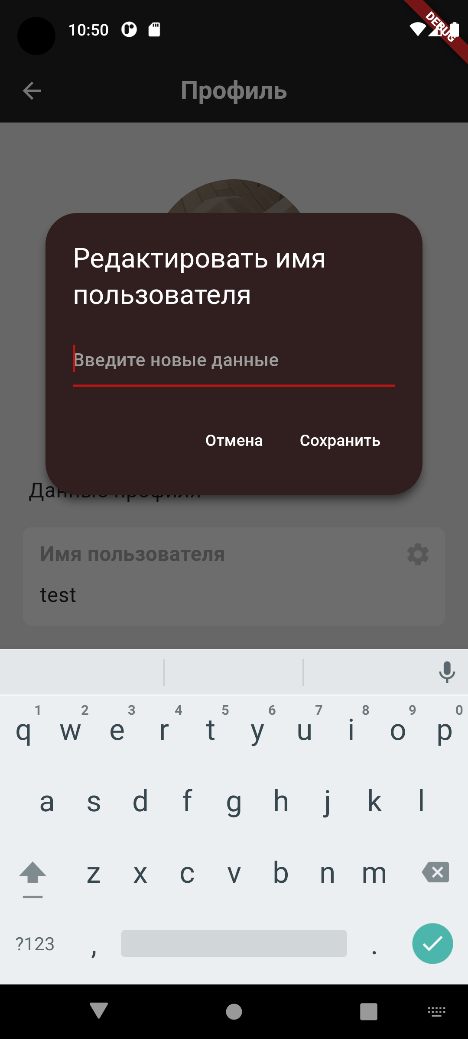
 

Рисунок 12 – Экран профиля

3. Ошибка загрузки экрана.

ЗАКЛЮЧЕНИЕ

В ходе прохождения учебной практики был приобретен опыт разработки кроссплатформенных приложений с использованием фреймворка Flutter и языка программирования Dart.

В ходе разработки приложения был проведен анализ, проектирование и интеграция с внешними API, активно использовались средства тестирования и отладки.

Работа над проектом поспособствовала улучшению навыков планирования рабочего дня, а также эффективного распределения задач и ресурсов.

Прохождение практики позволило увидеть, какие реальные задачи стоят перед разработчиками мобильных приложений в настоящее время.

Техническое задание было реализовано в полном объеме. Требования к заданию были учтены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Flutter [Электронный ресурс]. URL: <https://docs.flutter.dev/> (дата обращения 17.07.2021).
2. Документация Dart [Электронный ресурс]. URL: <https://dart.dev/guides> (дата обращения 17.07.2021).
3. Документация Visual Studio Code [Электронный ресурс]. URL: <https://code.visualstudio.com/docs> (дата обращения 17.07.2021).
4. Документация Android Studio [Электронный ресурс]. URL: <https://developer.android.com/docs> (дата обращения 17.07.2021).
5. Документация Firebase [Электронный ресурс]. URL: <https://firebase.google.com/docs?hl=ru> (дата обращения 17.07.2021).
6. Документация Flutter Bloc [Электронный ресурс]. URL: <https://bloclibrary.dev/#/> (дата обращения 17.07.2021).
7. Документация Kinopoisk Unofficial API [Электронный ресурс]. URL: <https://kinopoiskapiunofficial.tech/documentation/api/> (дата обращения 17.07.2021).

ПРИЛОЖЕНИЕ А

Название приложения (при наличии)