Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**Разработка мобильной версии видеохостинга**

Пояснительная записка к курсовому проекту

ПМ.01Разработка модулей программного обеспечения для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.211200.043.000ПЗ

Разработал:

студент группы ПР-315

Климова А.Д.

2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ 4

1.1 Описание предметной области 4

1.2 Образ клиента 4

1.3 Сценарии 5

1.4 Сбор и анализ прототипов 5

2ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 7

2.1 UI/UXдизайн приложения 7

2.2 Выбор технологии, языка и среды программирования 10

3РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ 11

3.1 Разработка базы данных 11

3.2 Разработка мультимедийного контента 12

3.3 Описание используемых плагинов 14

3.4 Описание разработанных процедур и функций 15

4 ТЕСТИРОВАНИЕ 20

ЗАКЛЮЧЕНИЕ 27

СПИСОК ИСТОЧНИКОВ 28

ПРИЛОЖЕНИЕ А. Техническое задание 29

# ВВЕДЕНИЕ

Большую часть жизни современных людей занимает медиапространство, выступающих, как пространство для поиска и просмотра медиафайлов. Человеку необходимо множество различных хостингов, чтобы у него всегда были выбор и альтернатива, в случае непригодности того или иного хостинга. В актуальности аниме-культуры в мире можно убедиться, посмотрев на статистику одного из самых популярных стриминговых сервисов, Netflix: по заявлению продюсера Кохэй Обары в 2021 году более половины подписчиков сервиса смотрели картины жанра аниме. Актуальность приложения, выполняющего функции личной библиотеки просмотренных и избранных сериалов, полнометражных и короткометражных фильмов, исходя из указанной информации, очевидна. Возможность просмотреть трейлер и прочитать описание мультсериала или полнометражного мультфильма являются полезным функционалом приложения.

Приложение предназначено для создания собственного списка просмотренных сериалов и полнометражных фильмов. Пользователь, не обращаясь к браузеру, в быстром доступе может посмотреть трейлер и прочитать подробную информацию об интересующем его аниме.

Целью курсового проекта является создание карманной библиотеки японской анимации. Для достижения цели поставлены следующие задачи:

* исследовать предметную область;
* спроектировать макет приложения;
* спроектировать базу данных;
* разработать приложение по макету;
* протестировать полученный продукт.

Объект – видеохостинг.

Предмет – организация ресурсов видеохостинга на мобильном устройстве.

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ
   1. Описание предметной области

Каждый современный человек пользуется видеохостингами. Видеохостинги несут огромную пользу обществу, будь то образовательный контент, культурный или развлекательный. Видеоконтента в жизни современного человека стало так много, что держать в голове список просмотренного материала человек попросту не сможет. В этом ему и поможет личная библиотека.

Личная библиотека – это инструмент, который помогает формировать заметки просмотренного и избранного контента. Это могут быть сериалы и мультфильмы в жанре аниме. В качестве личной библиотеки можно использовать как бумажный блокнот, так и заметки на мобильном устройстве, но функционал и удобство приложения гораздо лучше других вариантов. Блокнот не всегда будет при себе, а в приложении записи вести удобнее.

Удобство приложения заключается в том, что в нём пользователь всегда сможет посмотреть краткое описание сериала или мультфильма, а также его трейлер.

Приложение «AnimeLib» – программный продукт, который всегда будет под рукой, так как телефон у современного человека всегда с собой. Приложение выполняет функцию личной библиотеки без необходимости авторизации. Все данные о пользователе хранятся локально.

1.2 Образ клиента

Приложение предназначено для людей разных возрастов. Основной возрастной контингент от 10 до 30 лет. Пол и семейное положение пользователей не важны. Функционал приложения особенно привлекает аудиторию, уже привыкшую вести записи просмотренных фильмов и сериалов жанра японской анимации, а также развивает такую привычку у тех, у кого не было такого опыта раньше. Приложение предназначено для русскоязычной аудитории. Приложение удобно и полезно как для людей, только начинающих познавать мир японской анимации, так и для её постоянных зрителей.

1.3 Сценарии

Алексей давно смотрит аниме, и у него накопился большой список просмотренного контента. Всё время он записывал то, что посмотрел в заметках своего мобильного устройства. Но Алексею было бы гораздо легче, если бы у него на мобильном устройстве было установлено приложение с возможностью вести список просмотренных аниме с разделением их на разделы «Просмотренное» и «Избранное».

У Сергея хобби писать рецензии к фильмам и мультфильмам. Познакомившись с миром аниме, он решил писать рецензии к сериалам и полнометражным произведениям этого жанра. Сергею было бы удобнее вести отчётность, если бы у него на мобильном устройстве было установлено приложение, в котором можно просмотреть краткую информацию и составить списки просмотренных и избранных аниме.

1.4 Сбор и анализ прототипов

На данный момент существует не так много приложений, в которых можно вести список просмотренных фильмов и сериалов, а также читать информацию и просматривать трейлеры. Из предложенных в GooglePlay рассмотрим «AnixartLite». Из предложенных в AppStore – «4 Anime». Оба приложения имеют функционал для создания списка просмотренного контента. «AnimeLib» – разрабатываемое приложение.

Сравнение по основным критериям данных приложений представлено в таблице 1.

Таблица 1 – Сравнение приложений

|  |  |  |
| --- | --- | --- |
| Параметры | AnixartLite | 4 Anime |
| Стоимость | Бесплатно | Бесплатно |
| Язык | Русский | Английский |
| Просмотр трейлеров | Отсутствует | Отсутствует |
| Возможность вести список просмотренного контента | Есть | Есть |
| Подробное описание к контенту | Есть | Отсутствует |
| Наличие поиска | Есть | Есть |
| Выбор темы интерфейса | Есть | Есть |
| Наличие анимаций | Есть | Есть |
| Оценки | Есть | Есть |
| Возможность просмотра эпизодов напрямую в приложении | Отсутствует | Есть |
| Разделение контента на разделы | Есть | Есть |

Рассмотрев несколько приложений, выполняющих похожие задачи, было решено написать приложение, которое имело бы простой интерфейс без лишних функций и с акцентом внимания на важных для пользователя элементах, в котором можно просматривать краткую информацию, с возможностью просмотра трейлера.

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1 UI/UX дизайн приложения

2.1.1 Цветовая схема приложения и логотип

Проектирование дизайна и прототипа приложения производилось в графическом редакторе «Figma».

Определены две основные цветовые схемы приложения. Первая цветовая схема выполнена в более ярких тонах (Рисунок 1).

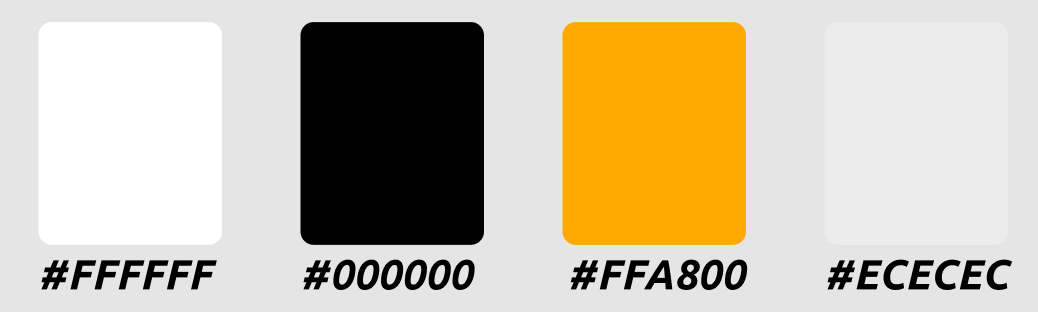


Рисунок 1 – Светлая цветовая схема

Вторая палитра цветов более тёмных, серых оттенков (Рисунок 2). Разделение на тёмную и светлую тему в приложении необходимо, так как пользователю может быть приятнее использовать тёмную тему ночью вместо светлой.



Рисунок 2 – Тёмная цветовая схема приложения

На основе цветовых схем разработан логотип (Рисунок 3). Логотип указывает на индивидуальность приложения и это первое, что видит пользователь перед скачиванием. Логотип состоит из заглавных букв названия приложения «AnimeLib». Для того, чтобы пользователь не смотрел на пустой белый экран, на начальном экране загрузки размещён логотип приложения.



Рисунок 3 – Логотип приложения

* + 1. Дизайн приложения

После определения цветовой палитры и создания логотипа разрабатывался полный дизайн, состоящий из пяти экранов:

* экран загрузки (splash screen);
* главный экран (main screen);
* экран информации (information screen);
* экран просмотренного контента (viewed screen);
* экран избранного контента (favorites screen).

На рисунке 4 представлен дизайн с использованием светлой палитры цветов.

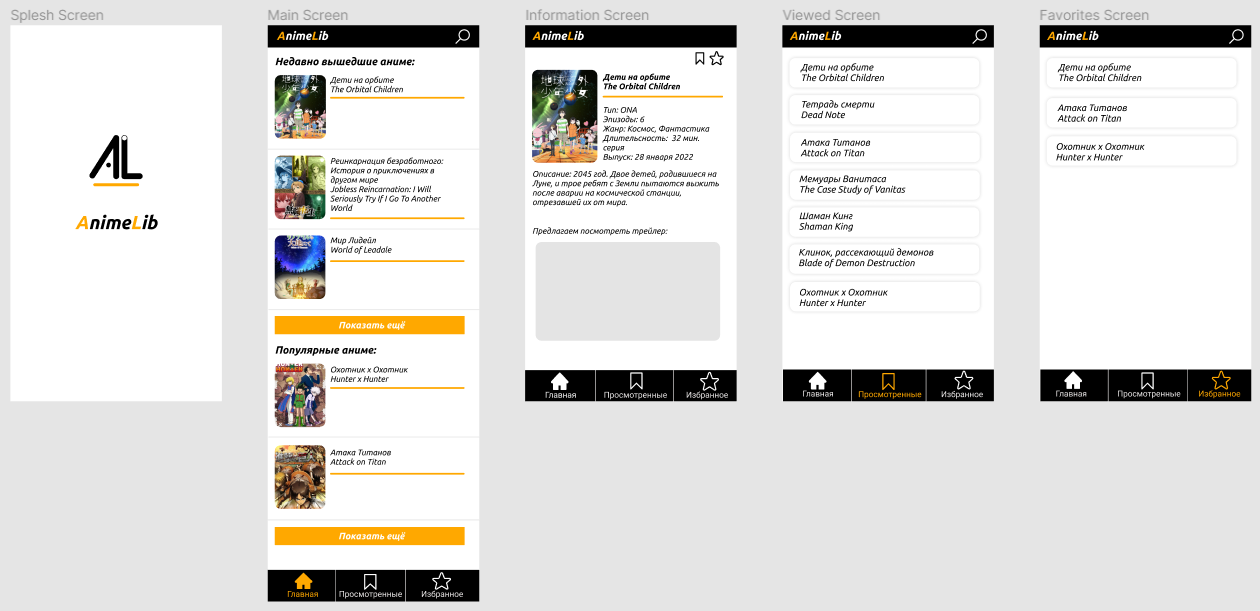


Рисунок 4 – Дизайн приложения в светлой теме

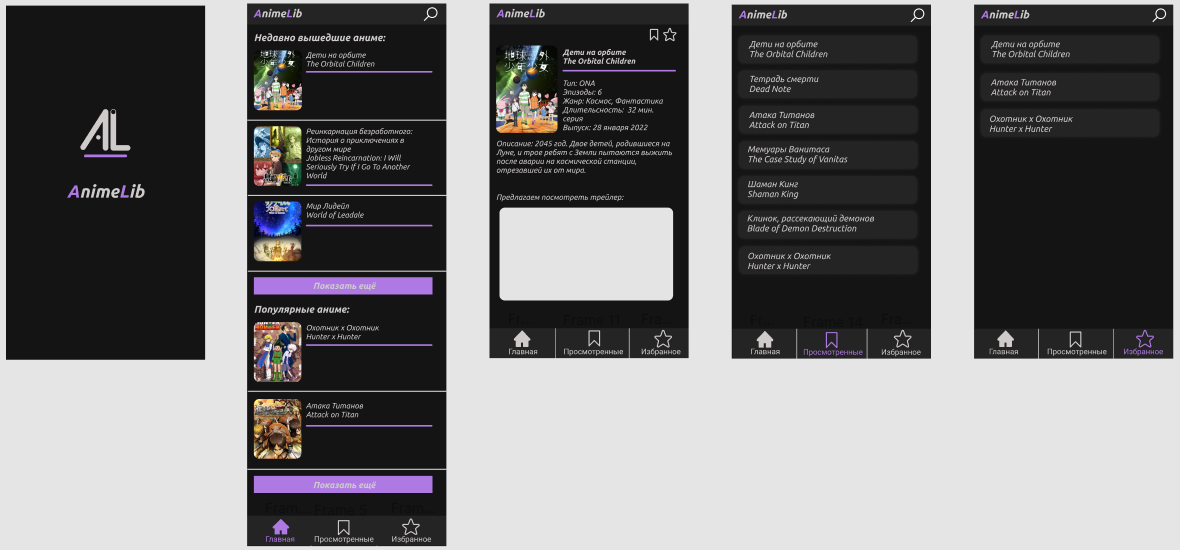
На рисунке 5 представлен дизайн разработанных экранов в тёмной теме.

Рисунок 5 – Дизайн приложения в тёмной теме

На рисунке 6 показано перемещение пользователя в приложении.

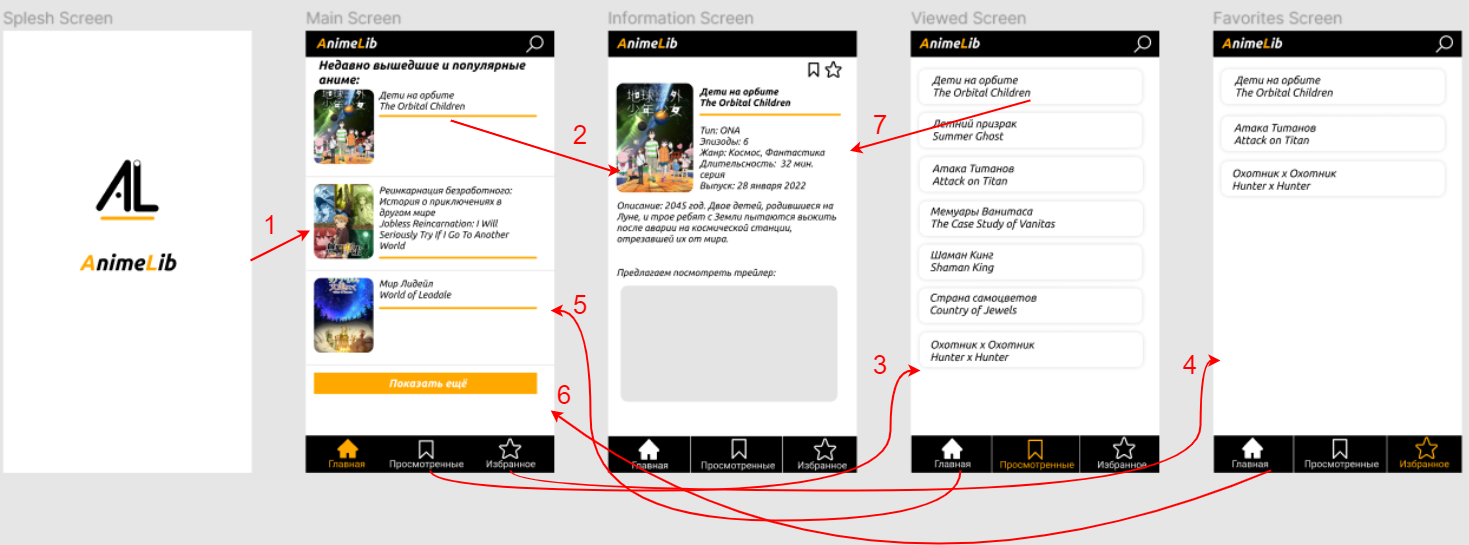


Рисунок 6 – Перемещение пользователя

При запуске приложения показывается экран загрузки, после чего пользователь попадает на главный экран (1). При нажатии на элемент списка на главном экране пользователь переходит на экран информации об аниме (2). С главного экрана можно перейти на экран просмотренного контента, либо на экран избранного контента (3;4). С экрана «Просмотренные» и с экрана «Избранное» можно перейти на главный экран (5;6). При нажатии на элемент списка на экране, например, просмотренного контента пользователь перейдет на экран информации об аниме, на экране избранного контента аналогично (7).

## 2.2 Выбор технологии, языка и среды программирования

Разработка мобильного приложения происходила в среде разработки Android Studio с использованием языка программирования Java, который по функционалу даёт большие возможности в сфере разработки.

Смартфоны, планшеты и другие устройства на базе Android доступны практически каждому. При этом язык Java в разработке для Android считается фундаментальным. Рынок требует все больше мобильных приложений, которые автоматизируют повседневные задачи пользователей. Количество пользователей Android растёт, поэтому разрабатывать программы для самой популярной мобильной платформы очень выгодно.

3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

3.1 Разработка базы данных

В качестве разрабатываемой базы данных выбрана облачная база данных Firebase. База данных интегрируется в Android Studio с помощью специальных встроенных инструментов. Для того чтобы подключить базу данных необходимо открыть вкладку Tools в Android Studio и выбрать Firebase, в открывшемся окне выбрать базу Realtime Database.

Поскольку Firebase является NoSQL-базой данных (не табличной базой данных), то данные хранятся в формате JSON и синхронизируются в реальном времени для каждого клиента. Для того чтобы создать JSON-файл с данными необходимо перейти в консоль разработчика Firebase и добавить новое хранилище данных. Пример хранения данных в базе представлен на рисунке 7.

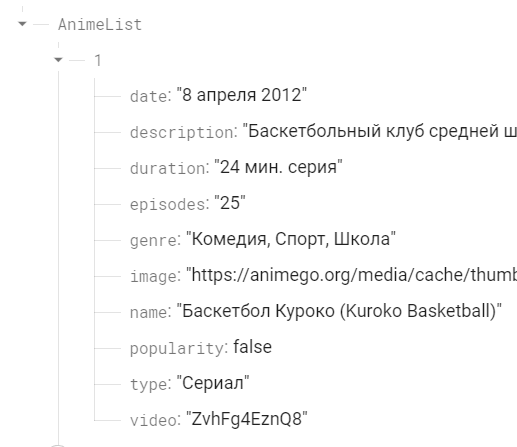


Рисунок 7 – Хранение данных в базе

База данных имеет древовидную структуру, где ID (уникальный идентификатор) является вершиной дерева и каждому идентификатору соответствует набор информации о конкретном аниме. Ниже на рисунке 8 представлена часть структуры базы данных в виде дерева.

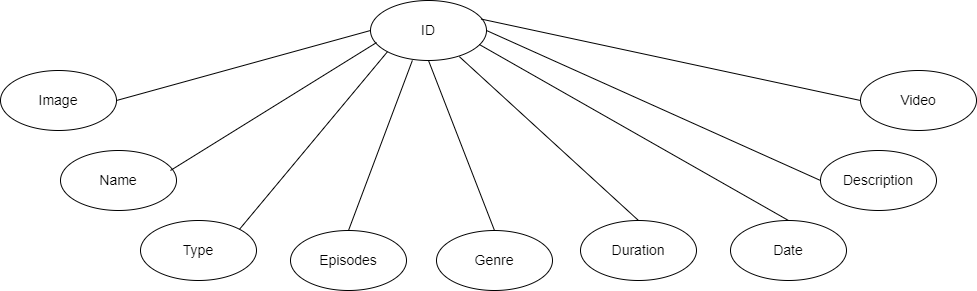


Рисунок 8 – Дерево базы данных

В базе данных приложения необходимо хранить информацию по каждому аниме: название, тип (сериал, фильм, ONA), количество эпизодов, жанр, длительность фильма/серии сериала, дату выпуска, описание, ссылки на трейлеры и ссылки на картинки.

3.2 Разработка мультимедийного контента

Весь мультимедийный контент разрабатывался с помощью языка разметки XML. Вёрстка выполнялась полностью в соответствии с ранее разработанным дизайном, но по мере разработки приложения в дизайн были внесены некоторые коррективы.

Все иконки и другие визуальные ресурсы были импортированы из Figma в проект в виде XML-файлов (Рисунок 9). Данный подход снижает общий вес APK-файла приложения и проекта, также отсутствуют потери в качестве мультимедийного контента. Данный тип ресурсов хранится в директории «drawable».

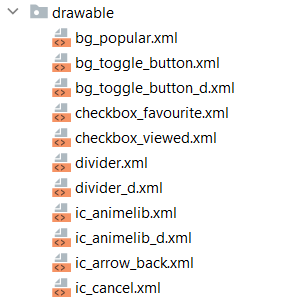
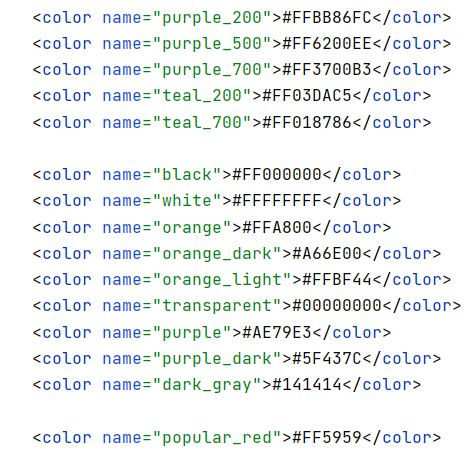


Рисунок 9 – Используемые иконки и визуальные ресурсы

Поскольку в приложении используются две цветовые схемы, в файл colors.xml занесены обе палитры (Листинг 1).

Листинг 1 – Светлая и тёмная палитра цветов в ресурсах



Ссылки на эти цвета используются для описания цветов темы приложения. Палитра для светлой темы была описана в файле themes.xml, для тёмной в themes.xml (night). На листинге 2 представлены светлая и тёмная темы.

Листинг 2 – Светлая и тёмная тема приложения





Выбор между тёмной и светлой темой – достойная черта приложения. При выборе светлой или тёмной темы приложений решения пользователей мобильных устройств разнятся, так как у кого-то могут болеть глаза из-за светлой темы, а кому-то недостаточно видно, если установлена тёмная тема. Также многие пользователи отдают своё предпочтение тёмной теме в любое время суток. Именно поэтому в разрабатываемом мобильном приложении пользователям предоставляется выбор.

3.3 Описание используемых плагинов

В проекте используются библиотеки с различными компонентами и элементами компоновки. Список библиотек с их описанием представлен в таблице 2.

Таблица 2 – Список используемых плагинов

|  |  |
| --- | --- |
| Разработчик/наименование | Описание |
| androidx.navigation:navigation-fragment:2.4.2  androidx.navigation:navigation-ui:2.4.2 | Популярная библиотека для навигации между "пунктами назначения" в приложении Android |
| files('libs/YouTubeAndroidPlayerApi.jar') | Библиотека, которая позволяет использовать методы для загрузки и проигрывания видеороликов и плейлистов, а также управлять видео в нужном режиме |
| 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0' | Компонент, с помощью которого можно обновить данные на странице свайпом вниз |
| 'com.google.firebase:firebase-database:20.0.5'   'com.google.firebase:firebase-core:21.0.0'   'com.firebaseui:firebase-ui-database:6.0.2' | Библиотеки для работы с базой данных Firebase |
| 'com.google.android.flexbox:flexbox:3.0.0' | Многофункциональный контейнер для компоновки элементов внутри |

Для того чтобы подключить вышеперечисленные плагины необходимо их указать в файле build.gradle. Пример подключения представлен ниже.

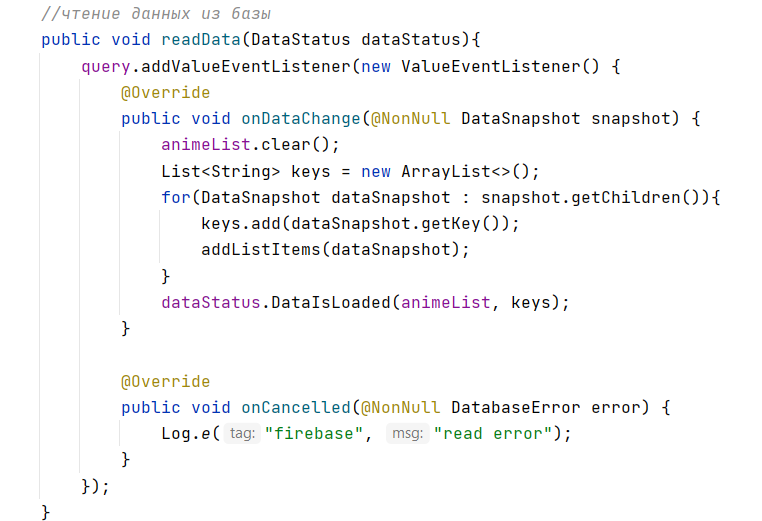
implementation 'com.google.firebase:firebase-database:20.0.5'

3.4 Описание разработанных процедур и функций

После вёрстки всех экранов реализовывались функции для навигации с помощью нижнего меню.

На главном экране, экране просмотренного и избранного контента, списки выводятся с помощью компонента для отображения элементов списка RecyclerView. Данные в RecyclerView заносятся с помощью функции, представленной на листинге 3.

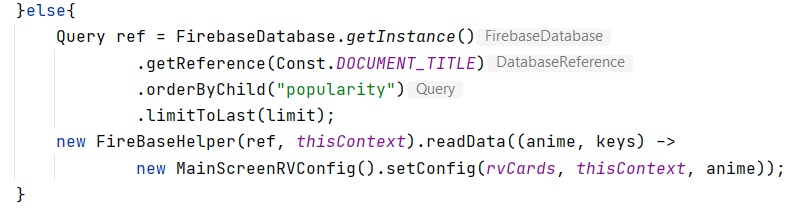
Листинг 3 – Чтение данных из базы



Функционал для главного экрана содержит такие функции, как раскрытие полного списка при нажатии на кнопку и просмотр списка аниме прокруткой (скролом). Список аниме выводится в виде списка в RecyclerView (компонент для отображения элементов списка).

Функция раскрытия полного списка при нажатии на кнопку выполняет запрос в базу данных. Код запроса представлен на листинге 4.

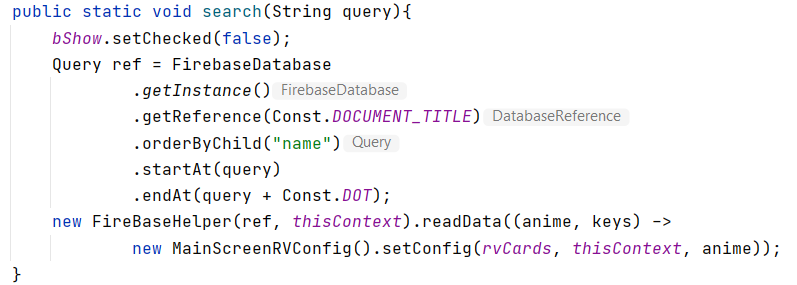
Листинг 4 – Запрос в базу данных приложения



Функция просмотра полного списка скролом – NeastedScrollView (компонент, позволяющий прокручивать иерархию представлений, размещенную в нём).

Функция поиска присутствует в MainActivity, но задействована на трёх экранах: главный экран, экран просмотренных и экран избранных. В зависимости от выбранного экрана отправляется запрос в базу данных на поиск (Листинг 5).

Листинг 5 – Запрос на поиск



Реализована функция для добавления в избранное. Добавление происходит при помощи внутреннего хранилища SharedPreference (Листинг 6).

Листинг 6 – Функция добавления в избранное



Функция добавления в просмотренные работает аналогично (Листинг 7).

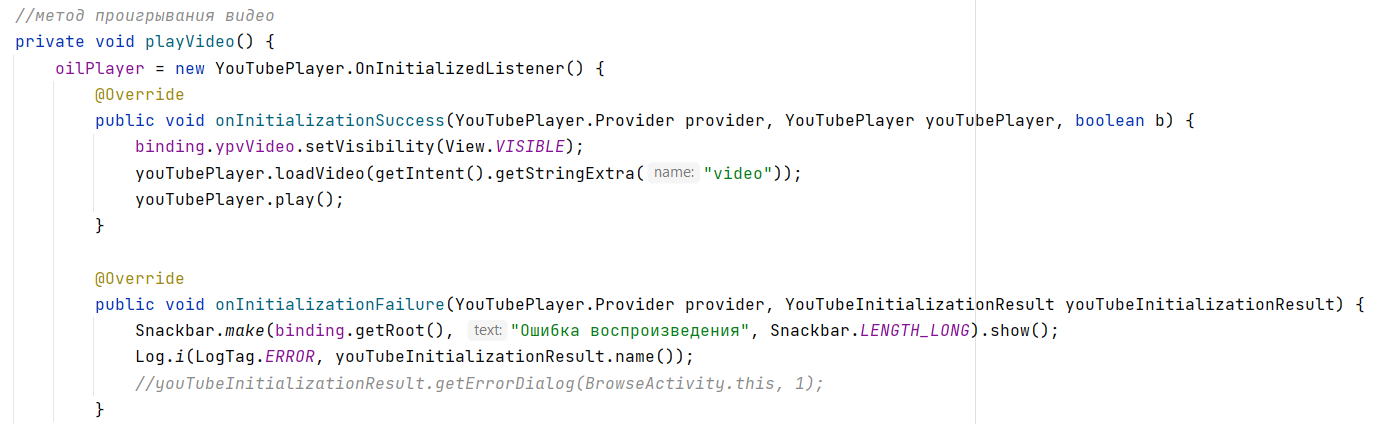
Листинг 7 – Функция добавления в просмотренные



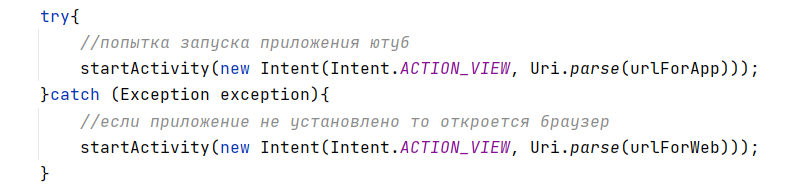
На экране информации к фильмам и сериалам реализована функция воспроизведения видео, добавления в просмотренные/избранные.

Функция воспроизведения видео реализована через YouTubeAPI. Функция на вход принимает ID видео и воспроизводит его. На листинге 8 представлен метод проигрывания видео.

Листинг 8 – Функция воспроизведения видео



Также на экране подробной информации реализована функция открытия видео либо в приложении YouTube, если оно установлено на мобильном устройстве пользователя, либо в браузере. Данная функция работает через запрос представленный на листинге 9.

Листинг 9 – Функция воспроизведения трейлера вне приложения

Таким образом, при использовании компонента для отображения элементов списка – RecyclerView, происходит отображение полного списка фильмов и сериалов на главном экране. Сохранение в избранное или просмотренное осуществляется с помощью локального хранилища SharedPreference на мобильном устройстве пользователя. Воспроизведение трейлеров происходит при помощи использования библиотеки YouTubeAPI.

# 4 ТЕСТИРОВАНИЕ

Тестирование дизайна приложения проводится на самом минимальном (Android SDK 24) и на более позднем (Android SDK 31) с различной диагональю экранов для проверки разметки страниц и вёрстки приложения.

Примеры проверок отображения элементов на экране представлены на рисунках 10, 11, 12, 13, 14, 15, 16.

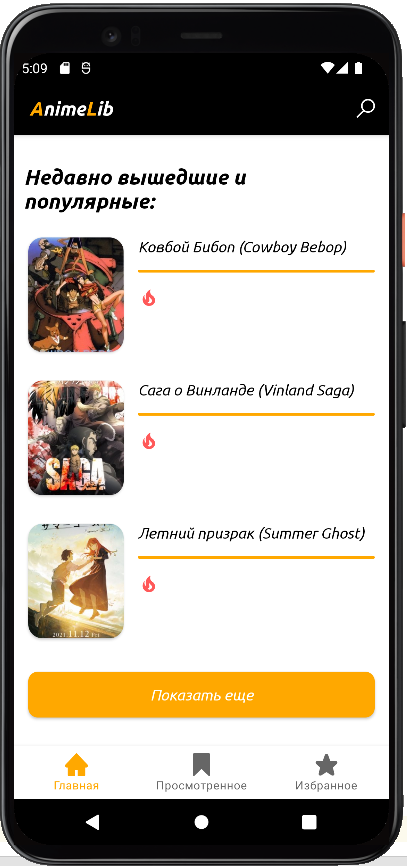
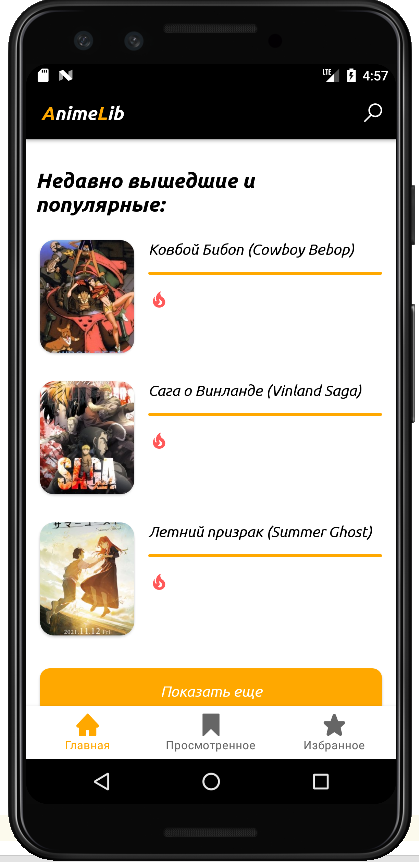


Рисунок 11 – Главный экран на 29 и 31 API соответственно

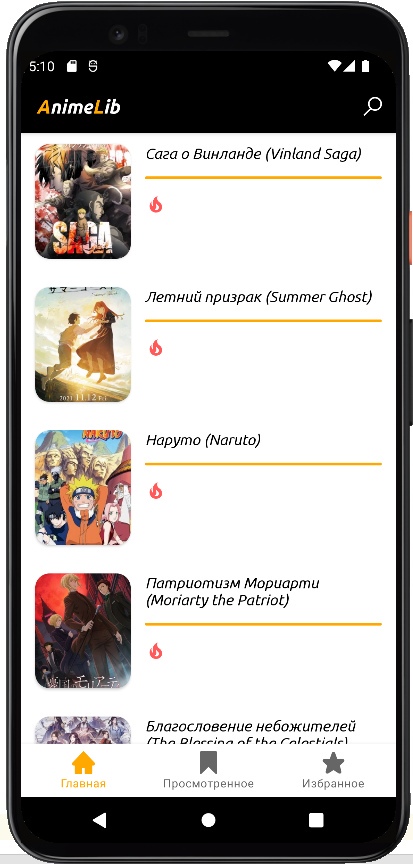
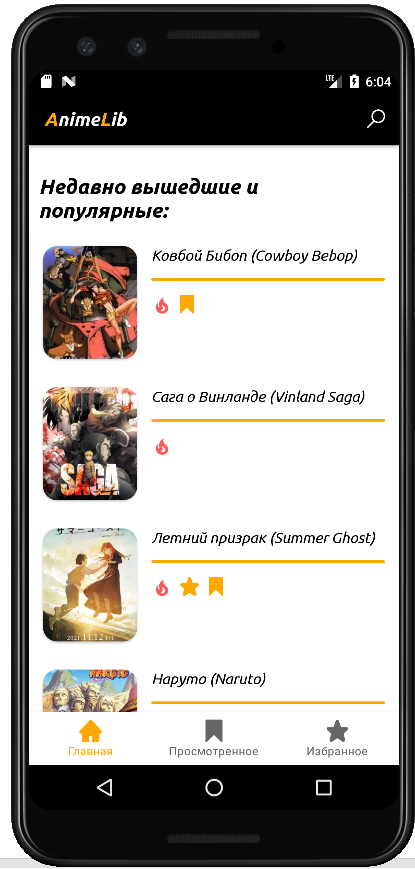


Рисунок 12 – Раскрытие полного списка на 29 и 31 API соответственно

Вывод элементов списка выполняется корректно в обоих случаях. Никаких ошибок не обнаружено.



Рисунок 13 – Экран информации на 29 и 31 API соответственно

Изображение и вся информация по аниме отображаются из базы данных на обоих устройствах корректно. Обложка к сериалу не меняет свой масштаб и не съезжает, никаких ошибок не обнаружено.

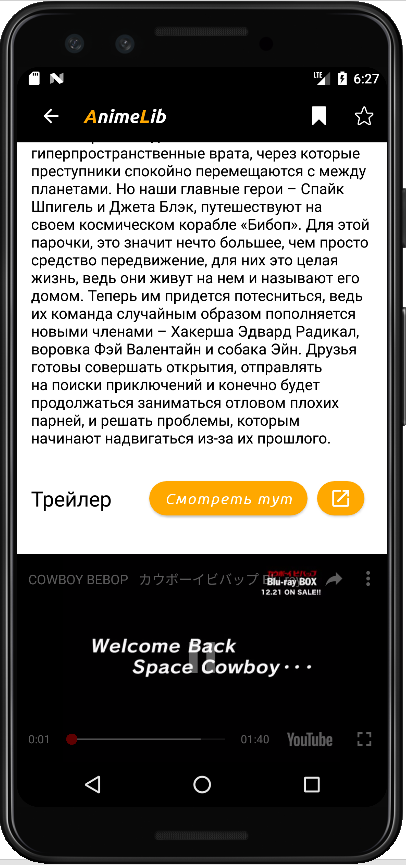


Рисунок 14 – Воспроизведение трейлера на 29 и 31 API соответственно

Отображение трейлера выполняется корректно в обоих случаях. Никаких ошибок не обнаружено.

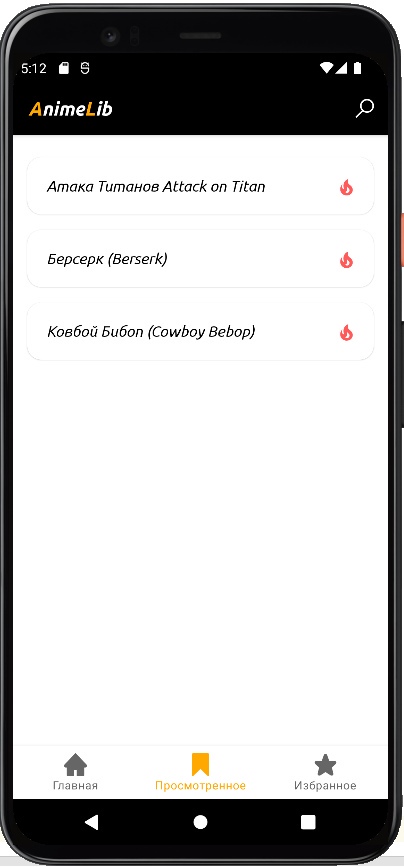
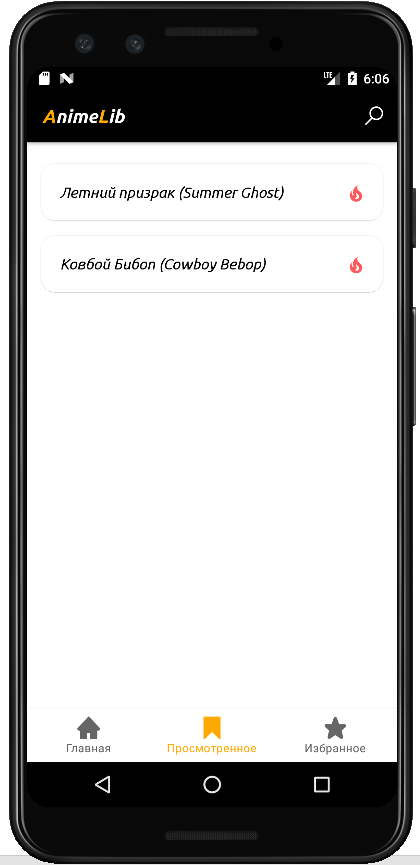


Рисунок 15 – Экран просмотренного контента на 29 и 31 API соответственно

На обоих устройствах список «Просмотренные» выводится корректно, при многократном нажатии на значок соответствующего экрана аниме добавилось в список лишь один раз, никаких ошибок не обнаружено.

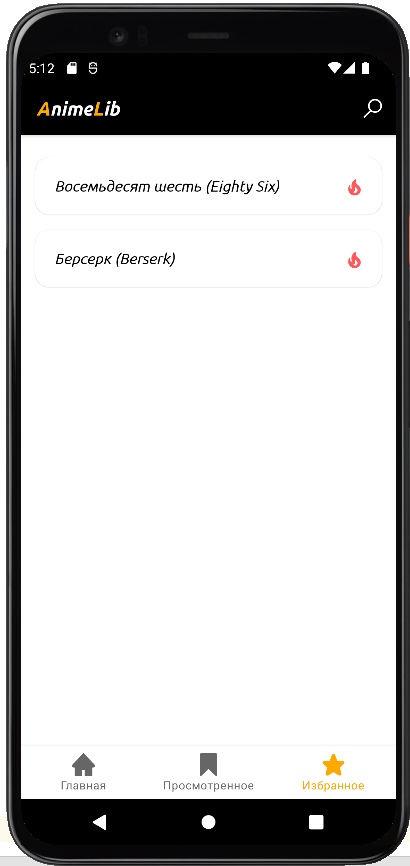
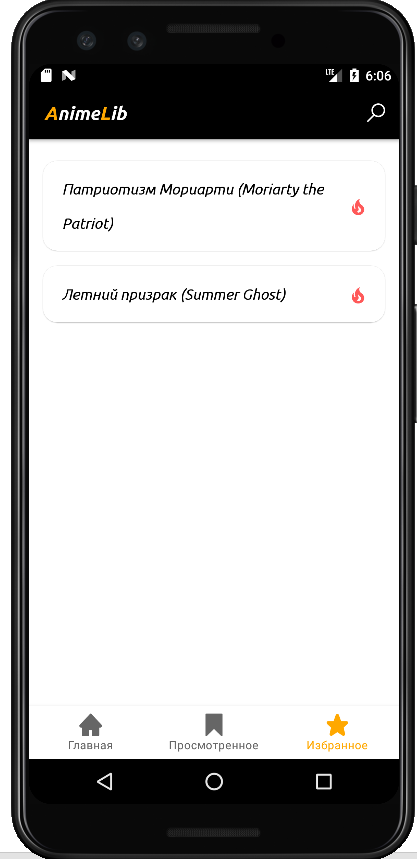


Рисунок 15 – Экран избранного контента на 29 и 31 API соответственно

На обоих экранах список «Избранные» выводится корректно. Как и было задумано, при многократном нажатии на значок соответствующего экрана в список аниме добавляется только один раз. Никаких ошибок при тестировании не обнаружено.

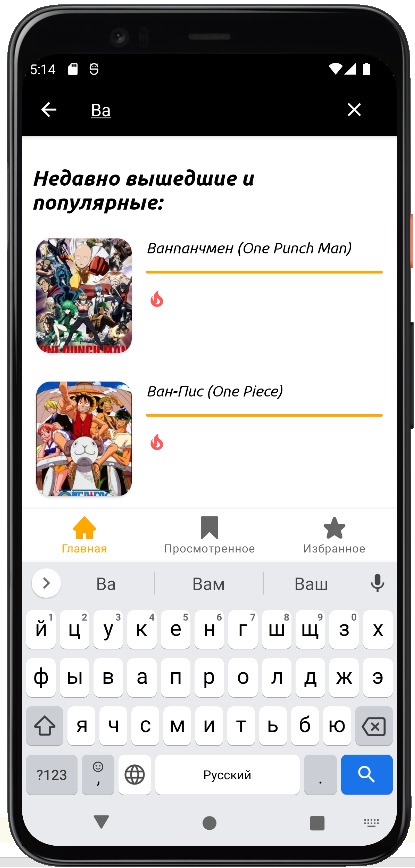
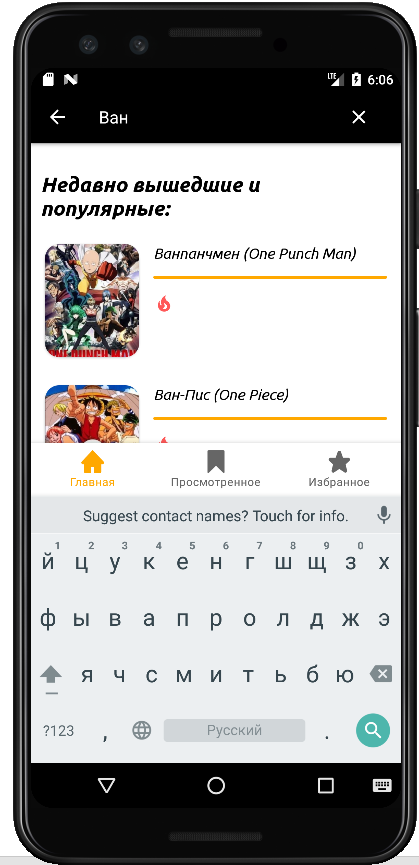


Рисунок 16 – Поиск на главном экране на 29 и 31 API соответственно

При поиске все элементы отображаются корректно. Поиск выполняется правильно. Никаких ошибок не обнаружено.

В процессе тестирования не выявлено ошибок в дизайне приложения и его логике. Каждый экран прошёл проверку на разных API на корректное отображение элементов соответствующих экранов.

ЗАКЛЮЧЕНИЕ

В ходе разработки мобильного приложения было выявлено, что аниме является популярным жанром у многих людей. Также, учитывая количество контента в заявленном жанре, функция ведения списка просмотренного контента является актуальной для многих людей. Именно на этом было акцентировано внимание при создании приложения.

Основные экраны приложения спроектированы таким образом, чтобы не нагружать пользователя лишней информацией. Весь интерфейс реализован так, чтобы пользователь достиг своей цели, выполнив минимальное количество действий. Также добавлены полезные функции, такие как просмотр трейлеров, создание собственных списков просмотренного и наиболее понравившегося контента.

Просмотр трейлеров является одной из главных задумок приложения, ведь пользователю важно понимать, что он собирается посмотреть. Это очень полезная функция и в таких ситуациях, когда пользователь посмотрел фильм или сериал очень давно, чтобы освежить в памяти произведение, достаточно посмотреть трейлер.

Ведение списков просмотренного и избранного контента также является важной задумкой приложения. Так у пользователя всегда под рукой список того, что он когда-то смотрел.

В дальнейшем предлагается усовершенствовать приложение, добавив функцию смены языка, а также рейтинга и оценки к фильмам и сериалам.

В целях проверки корректности работы приложения проведено тестирование дизайна. Работа приложения успешно прошла проверку. Таким образом поставленная цель курсовой работы достигнута.

СПИСОК ИСТОЧНИКОВ

1 Moluch [Электронный ресурс]: Феномен аниме и манга в современной массовой культуре – Режим доступа: <https://moluch.ru/archive/360/80423/>

2 Steppe [Электронный ресурс]: Почему аниме так популярно – Режим доступа: <https://the-steppe.com/razvitie/pochemu-anime-tak-populyarno-i-kak-ono-stalo-indikatorom-yaponskoy-kultury>

3 Yasno [Электронный ресурс]: Мобильный интерфейс: разработка дизайна приложений и проектирование макетов на примерах – Режим доступа к руководству: <https://yasno.mobi/blog/mobilnyy-interfeys-razrabotka-dizayna-prilozheniy-i-proektirovanie-maketov-na-primerakh/>

4 DevelopersAndroid [Электронный ресурс]: Полная документация по Android–Режим доступа к руководству: <https://developer.android.com/>

5 METANIT [Электронный ресурс]: Документация по языку программирования Java – Режим доступа к руководству: <https://metanit.com/java/tutorial/>

6 itProger [Электронный ресурс]: Уроки по Androidразработке – Режим доступа: <https://itproger.com/course/android/9>

7 Webkyrs [Электронный ресурс]: Создание базы данных Firebase – Режим доступа к руководству: <https://webkyrs.info/page/kak-sozdat-bazu-dannykh-firebase-realtime-database>

8 Epochta [Электронный ресурс]: Как составить портрет клиента – Режим доступа: <https://www.epochta.ru/blog/portret-klientov>

9 wtftime [Электронный ресурс]: Netflix – Режим доступа: <https://wtftime.ru/articles/123084/netflix-v-2021-godu-polovina-podpischikov-smotrela-anime/>

Приложение А

(информационное)

Техническое задание

ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку мобильного приложения «Разработка мобильной версии видеохостинга», используемого для просмотра трейлеров к фильмам и сериалам, ведения списков просмотренного и избранного контента.

Наименование программы: «AnimeLib». Далее по тексту Приложение.

Краткая характеристика области применения: мобильное приложение предоставляет возможность просмотра подробной информации к фильмам и сериалам жанра японской анимации с возможностью просмотра трейлера, а также ведения собственных списков просмотренного и наиболее понравившегося контента.

1 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основание для проведения разработки является Приказ № Уч-041/4 от 17.03.2022 года.

Наименование темы разработки − «Разработка мобильной версии видеохостинга».

Условное обозначение темы разработки – «AnimeLib».

2 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основное назначение приложения заключается в:

* обеспечении удобного и понятного интерфейса приложения для пользователя;
* предоставление большой библиотеки аниме.

Лица, которые могут работать с данной системой:

администратор – управляет полностью всем приложением, следит за его работоспособностью, обновляет информацию в базе данных приложения;

пользователь приложения – может просматривать подробную информацию к интересующим его фильмам и сериалам жанра японской анимации, а также вести списки просмотренного и наиболее понравившегося контента.

3 ТРЕБОВАНИЕ К ПРИЛОЖЕНИЮ

3.1 Требования к функциональным характеристикам

Система должна обеспечивать возможность выполнения нижеперечисленных функций, описанных в таблице А.1.

Таблица А.1 – Выполняемые функции приложения

|  |  |
| --- | --- |
| Номер | Функция |
|  | Просмотр списка фильмов и сериалов |
|  | Просмотр трейлеров |
|  | Поиск фильмов и сериалов |
|  | Добавление фильма или сериала в список просмотренных/избранных |
|  | Смена темы приложения |

3.2 Условия эксплуатации

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

3.3 Требования к составу и параметрам технических средств

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android не ниже версии 7.0.

3.4 Требования к информационной и программной совместимости

Разработка приложения ведется на языке программирования Java.

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android с версией SDK не ниже 24.

3.5 Требования к защите информации

Требования не представляются.

4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации должен включать в себя:

* техническое задание;
* пояснительная записка.

5 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Таблица A.2 − Стадии разработки

|  |  |  |
| --- | --- | --- |
| Этапы разработки КП | Срок выполнения | Отчетность |
| Определение цели и задач, объекта и предмета исследования | 26.03.2022 | Пояснительная записка |
| Описание предметной области | 02.04.2022 | Пояснительная записка |
| Выбор технологии, языка и среды программирования | 09.04.2022 | Пояснительная записка |
| Оформление технического задания | 16.04.2022 | Пояснительная записка |
| Проектирование UI/UX дизайна | 23.04.2022 | Пояснительная записка |
| Разработка мобильного приложения | 30.04.2022 | Пояснительная записка |

Окончание таблицы А.2

|  |  |  |
| --- | --- | --- |
| Отладка и тестирование приложения | 14.05.2022 | Пояснительная записка |
| Оформление документации | 21.05.2022 |  |
| Защита | 28.05.2022 |  |

7 ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт.