Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА Мобильного приложения «киноафиша»

Пояснительная записка к курсовому проекту

МДК.01.03 Разработка мобильных приложений

НАТКиГ.200100.010.000ПЗ

Разработал:

студент группы ПР-21.101

Барсук К.Д.

**Содержание**

[Введение 3](#_Toc165895864)

[1 Исследовательский раздел 5](#_Toc165895865)

[1.1 Описание предметной области 5](#_Toc165895866)

[1.2 Образ клиента 5](#_Toc165895867)

[1.3 Сценарии 6](#_Toc165895868)

[1.4 Сбор и анализ прототипов 7](#_Toc165895869)

[2 Проектирование приложения 10](#_Toc165895870)

[2.1 UI/UX дизайн приложения 10](#_Toc165895871)

[2.2 Выбор технологии, языка и среды программирования 12](#_Toc165895872)

[3 Разработка мобильного приложения 13](#_Toc165895873)

[3.1 Разработка базы данных 13](#_Toc165895875)

[3.2 Разработка мультимедийного контента 13](#_Toc165895876)

[3.3 Описание используемых плагинов 15](#_Toc165895877)

[3.4 Описание разработанных процедур и функций 16](#_Toc165895878)

[4 Тестирование 18](#_Toc165895879)

[4.1 Протокол тестирования дизайна приложения 18](#_Toc165895881)

[4.2 Протокол тестирования функционала приложения 20](#_Toc165895882)

[Заключение 23](#_Toc165895883)

[Библиография 24](#_Toc165895884)

[Приложение А (Обязательное) Техническое задание 25](#_Toc165895885)

[Приложение Б 32](#_Toc165895886)

Введение

С ростом интереса к кинематографу и увеличением количества выходящих фильмов, возникает потребность в надёжном и удобном источнике информации о кино. Активное развитие технологий и распространение мобильных устройств, которые все чаще используются для получения информации и развлечений.

Сегодня существует множество сайтов и мобильных приложений, предоставляющих информацию о фильмах. Однако, многие из них не адаптированы для мобильных устройств, имеют неудобный интерфейс, неполную или устаревшую информацию. Кроме того, часто эти ресурсы перегружены рекламой и не обеспечивают комфортного пользовательского опыта.

Разработка мобильного приложения «Киноафиша» позволит предоставить пользователям надежный и удобный источник информации о фильмах, адаптированный для мобильных устройств. Это приложение будет содержать полную и актуальную информацию о фильмах, их рейтингах и отзывах, а также предоставит возможность фильтровать фильмы по различным критериям, таким как жанр, год выпуска, рейтинг. Кроме того, приложение будет иметь интуитивно понятный интерфейс, который обеспечит комфортное использование приложения.

Таким образом, создание мобильного приложения «Киноафиша» является актуальной задачей, которая отвечает потребностям современных пользователей в быстром и удобном доступе к информации о кино. Реализация этого проекта позволит улучшить пользовательский опыт и предоставить качественный и надежный информационный ресурс о кино для мобильных устройств.

Целью курсового проекта является создание мобильного приложения «Киноафиша» для предоставления информации о фильмах.

Задачами проекта в связи с указанной целью являются:

* анализ существующих информационных ресурсов;
* проектирование пользовательского интерфейса, адаптированного для мобильных устройств;
* разработка функционала приложения;
* создание базы данных фильмов;
* тестирование полученного продукта.

Реализуемый продукт – мобильное приложение, предназначенное для просмотра информации о фильмах.

# 1 Исследовательский раздел

## Описание предметной области

Киноафиша — это настоящая сокровищница интереснейших кинопроизведений, в которой каждый киноман может найти что-то по своему вкусу. Она предлагает подробный обзор того, что можно увидеть на большом экране. А это значит, что вы сможете узнать о последних релизах и самых ожидаемых премьерах.

В киноафишах вы найдете разнообразие фильмов - от захватывающих блокбастеров до трогательных драм, от смешных комедий до мрачных триллеров. Каждый жанр представлен в полном объеме, удовлетворив желания самых взыскательных зрителей. Вы легко сможете найти информацию о дате выхода ваших долгожданных премьер, а также узнать подробности о фильмах, которые могут стать для вас настоящим открытием.

Кроме описаний самих фильмов, в афишах вы найдете данные об актерском составе, режиссерах и сценаристах. Это поможет вам понять, какие любимые актеры или режиссеры стоят за тем или иным проектом, и решить, стоит ли его посмотреть. Ведь зачастую именно имя режиссера или актера становится залогом качества и интереса фильма.

Также в афишах часто публикуются трейлеры, которые позволяют получить первое впечатление о фильме, его атмосфере и главных героях. Это как короткий пролог к самому фильму, заманчивое приглашение в кино.

Киноафиша — это настоящий гид в мире кино для тех, кто хочет быть в курсе последних тенденций и не пропускать важнейшие релизы. Она информирует, вдохновляет и предлагает невероятные истории, которые ждут своего рассказа на большом экране.

## Образ клиента

Образ клиента может включать в себя различные категории пользователей, такие как:

Киноманы: энтузиасты и любители кино, стремящиеся всегда быть в курсе последних новинок кинопроката и получать подробную информацию о фильмах.

Студенты и молодёжь: молодые люди, интересующиеся кино как частью культурного досуга и ищущие информацию о разнообразных жанрах фильмов для организации своего свободного времени.

Любители домашнего кинопросмотра: пользователи, предпочитающие смотреть фильмы в комфортной домашней обстановке и ищущие интересные кинопроизведения для просмотра в кругу семьи или друзей.

## Сценарии

Студент Иван часто обсуждает последние кинопремьеры в социальных сетях и любит делиться своими впечатлениями о просмотренных фильмах с друзьями. Ранее он тратил много времени на поиск информации о новых фильмах на различных сайтах и форумах. Ему было бы намного удобнее, если бы существовало мобильное приложение, где можно легко находить актуальные данные о новинках кино, включая детальные описания, трейлеры, рейтинги и дату выхода. Такое приложение позволило бы ему быстро получать интересующую информацию и очень облегчило поиск этой информации.

Виталий активно интересуется кинематографом, но сталкивается с проблемой запоминания не просмотренных фильмов. Для решения этой проблемы идеально подошло бы мобильное приложение, предлагающее возможность добавления фильмов в раздел «Избранное». Эта функция позволяет отмечать не просмотренные фильмы, создавая организованный список, и планировать что, посмотреть в будущем, облегчая процесс выбора. Такое приложение стало бы незаменимым помощником для забывчивого кинолюбителя, делая его кинопросмотры более осмысленными и организованными.

## Сбор и анализ прототипов

В GooglePlay и AppStore существует два приложения, выполняющих роль киноафиши. Первое – Кинопоиск, второе – Okko. Оба этих приложения имеют возможность просматривать информацию о фильмах и сохранять их в список избранного.

Оба этих приложения имеют главный экран (рисунок 1).

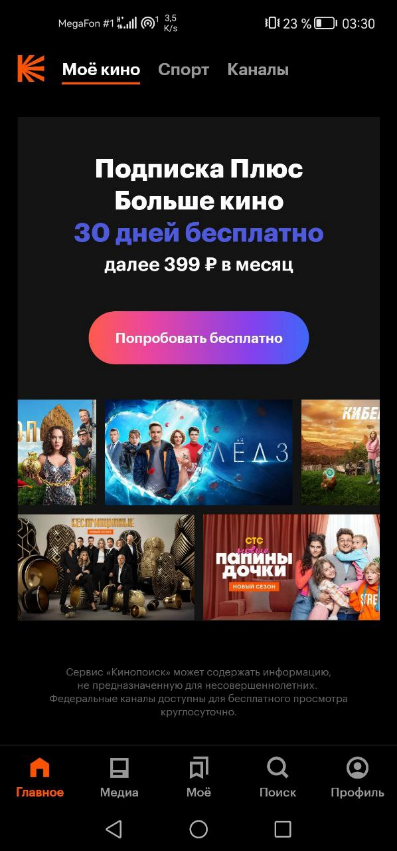
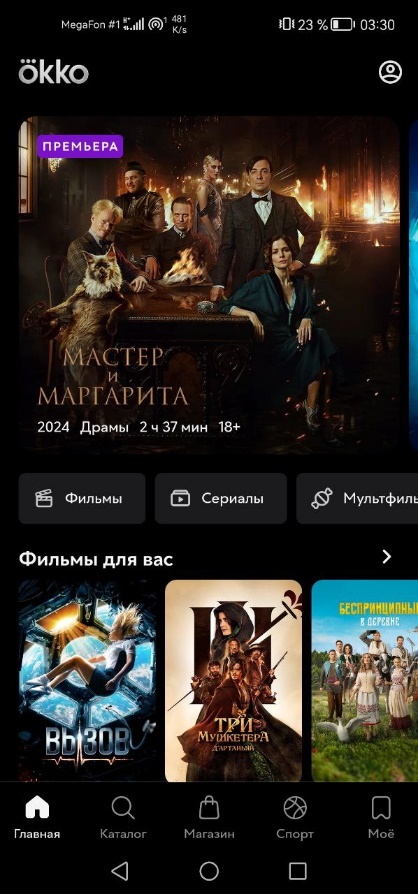
 

Рисунок 1 – Главный экран приложений

В приложениях на главном экране можно увидеть нижнюю и верхнюю панель, отвечающие за навигацию по приложению. На верхней панели в обоих приложении «Кинопоиск» кнопки:

* моё кино;
* спорт;
* каналы.

У Okko только кнопка профиля.

Нижняя панель отличается только элементами, к которым она перенаправляет, но общими являются:

* главная;
* поиск;
* моё.

Вывод видеороликов отличается. В приложении «Okko» – видео выводятся по группам, а в «Кинопоиск» – на главном экране вообще нет списка фильмов, пока не оформлена подписка. Логотип находится в одинаковом месте – в верхнем левом углу.

Так же в них реализована функция добавления видеороликов в «Избранное» (рисунок 2):

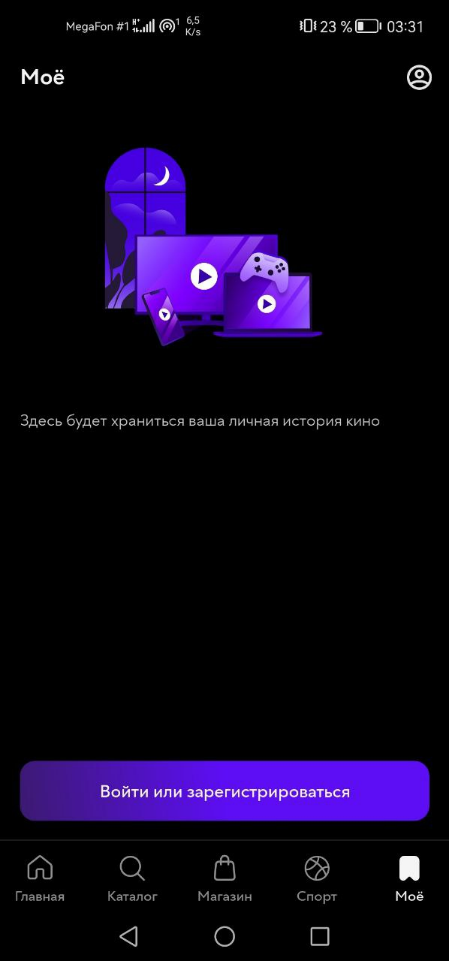
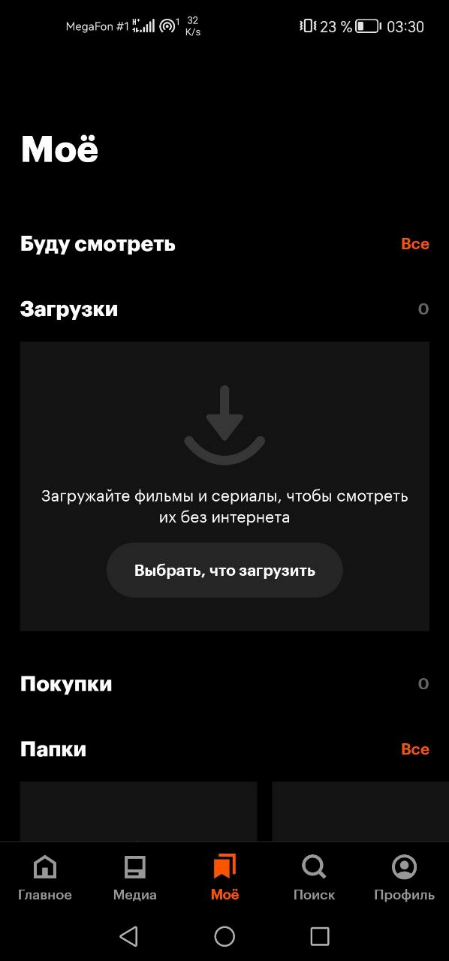
 

Рисунок 2 – Экран с избранными фильмами

Все добавленные в «Избранное» фильмы, отображаются на отдельной вкладке. Их так же можно оттуда удалить или посмотреть повторно. Однако в приложении «Okko» добавлять фильмы в избранное можно только если авторизироваться, что не очень удобно.

Сравнение по основным критериям данных приложений представлено в таблице 1.

Таблица 1 – Сравнение приложений

|  |  |  |
| --- | --- | --- |
| Параметры | Кинопоиск | Okko |
| Стоимость | Бесплатно | Бесплатно |
| Основной язык | Русский | Русский |
| Просмотр информации о фильмах | Есть | Есть |
| Возможность сохранять фильмы в «Избранное» | Есть | Есть |
| Подробное описание к контенту | Есть | Есть |
| Наличие поиска | Есть | Есть |
| Выбор темы интерфейса | Есть | Есть |
| Список просмотренного | Есть | Есть |
| Рейтинг фильмов | Есть | Есть |

Рассмотрев самые популярные приложения, выполняющие похожие задачи, решено написать приложение, которое имело бы простой интерфейс и с акцентом внимания на важных для пользователя элементах, в котором можно смотреть информацию о фильмах и добавлять их в «Избранное». За основу будут взяты разные элементы двух приложений, так как каждое из них имеет как свои плюсы, так и минусы.

# 2 Проектирование приложения

## UI/UX дизайн приложения

Дизайн проекта разработан в программе Figma, онлайн-редактор для разработки интерактивных макетов.

Для темы приложения определена следующая цветовая схема. Тема реализуется в тёмных тонах, поэтому в ней основными цветами являются: синий, черный, розовый (Рисунок 3).



Рисунок 3 – Цветовая схема

Стоит заметить, что данные цвета, разбавленные изображениями, не выглядят скучно, а весьма лаконично.

Логотип приложения представляет собой кино хлопушку, так же в основных цветах приложения.

Ниже на рисунке 4 представлен логотип киноафиши.



Рисунок 4 – Логотип видеохостинга

Цветовая схема логотипа состоит из: розового и черного (Рисунок 5). Это броские цвета, которые могут привлечь внимание пользователя к приложению.



Рисунок 5 – Цветовая схема логотипа

Определившись с цветовой схемой приложения и создав его логотип, был разработан дизайн следующих экранов:

* главный экран с фильмами;
* жанры;
* подробная информация о фильме;
* избранное.

Ниже на рисунке 6 представлен дизайн приложения.

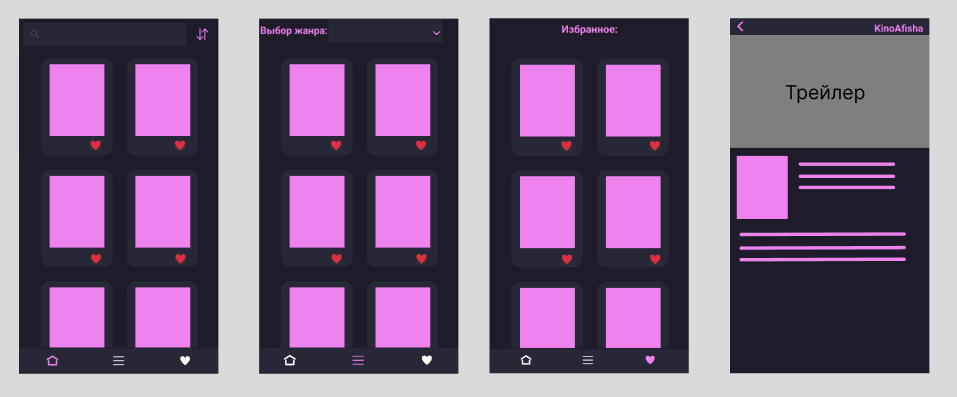


Рисунок 6 – Дизайн приложения

При запуске приложения пользователю открывается главный экран со списком фильмов. На этом экране предоставлен список фильмов с некоторый информацией о них. В верхней части экрана есть строка поиска и кнопка сортировки. С каждого экрана можно перейти в любую часть приложения. При нажатии на иконку сердца пользователь добавит фильм в избранное, и он будет отображаться на экране избранных. Нажимая на кнопку «Жанры» откроется экран жанров, на котором пользователь может отфильтровать список фильмов по нужному ему жанру. При нажатии на фильм, пользователь попадает на экран с трейлером этого фильма и подробной информацией о нём.

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio, так как она довольно удобна для создания приложения и является довольно популярной, поэтому в случае возникновения ошибки, легче найти способ её решения. Языком программирования является Java, так как это зрелый и широко используемый язык программирования, обладающий богатым набором библиотек и фреймворков, что делает его идеальным выбором для разработки надежных и масштабируемых приложений.

Используемый сервис для базы данных – PgAdmin 4. Это мощная, открытая система управления реляционными базами данных, которая отлично подходит для решения поставленных задач. Она обладает высокой производительностью и расширяемостью. PgAdmin 4 также предоставляет хорошие инструменты для работы с большими объемами данных и надежные механизмы безопасности.

# 3 Разработка мобильного приложения



## Разработка базы данных

В качестве разрабатываемой базы данных выбрана облачная база данных PostgreSQL, соединенная с Android Studio через API.

Пример хранения данных в базе представлен на рисунке 7.

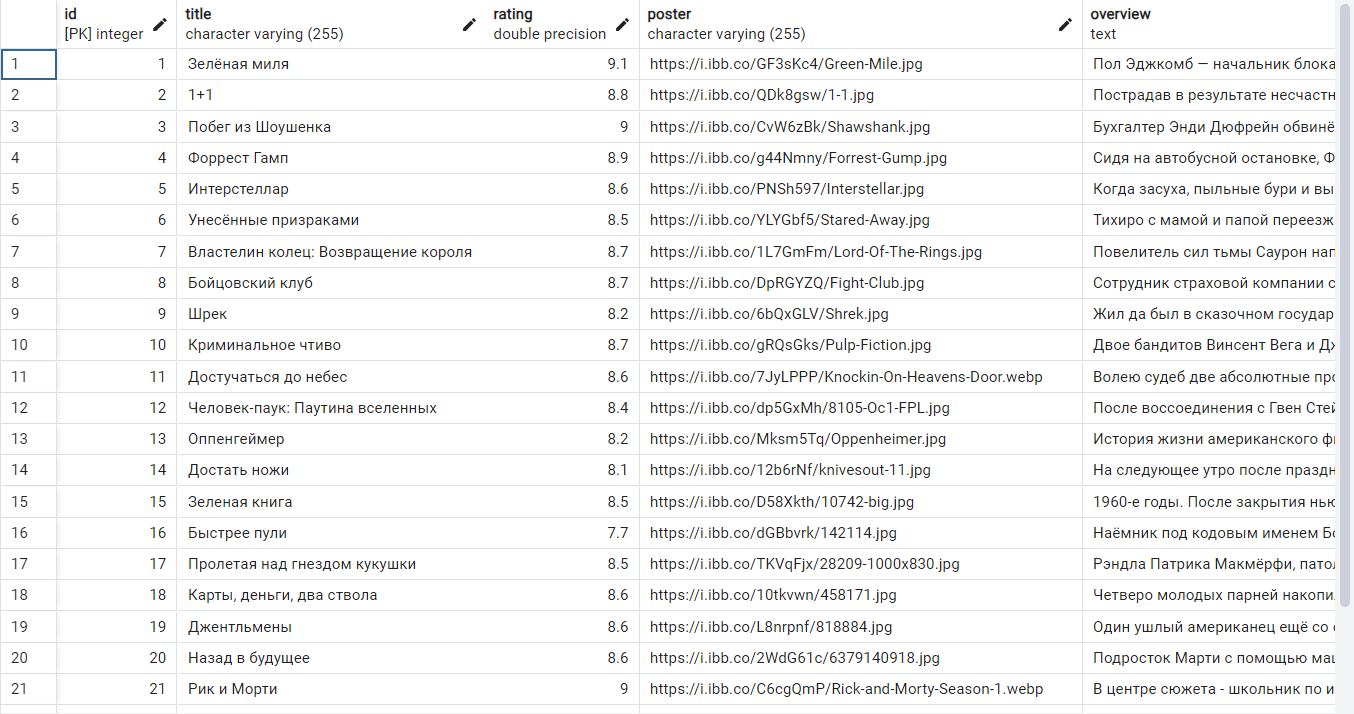


Рисунок 7 – Таблица в базе данных с информацией о фильмах

База данных организована в виде реляционных таблиц, где каждая запись представлено уникальным идентификатором (ID), который служит первичным ключом таблицы. Каждому ID соответствует запись, содержащая всю информацию о фильме.

В базе данных хранится вся информация о фильмах: название, дата выхода, постер, описание, жанры и рейтинг.

## Разработка мультимедийного контента

Весь мультимедийный контент разрабатывался с помощью языка разметки XML. Вёрстка выполнялась по дизайну, разработанному ранее в приложении Figma. Однако, по мере разработки, в дизайн были внесены изменения.

Иконки, кнопки и другие элементы приложения, были импортированы в проект в виде XML-файлов (Рисунок 7). Такой способ хранения уменьшает вес приложения, а также, избавляет от проблем с потерей качества мультимедийного контента.

В моем курсовом проекте для Android Studio я использую специальный сервер, на котором хранится база данных с информацией о фильмах. В этой базе данные о трейлерах и постерах к фильмам хранятся в виде символов, то есть ссылок на соответствующие файлы.

Для получения этих даннах в свое приложение, я создаю специальный интерфейс, который называется API (Application Programming Interface). Это своего рода мостик между моим приложением и базой данных на сервере.

Когда приложение запускается, оно отправляет запрос на этот API. Сервер обрабатывает этот запрос, достает нужные данные из базы и отправляет их обратно в приложение в специальном формате, который называется JSON.

Приложение получает этот JSON. Оно извлекает оттуда названия фильмов, описания, рейтинги и, конечно же, ссылки на постеры и трейлеры.

Для отображения постеров приложение использует специальную библиотеку, которая умеет загружать изображения по ссылкам и показывать их на экране. Для трейлеров есть другая библиотека, которая может воспроизводить видео, также используя ссылки из базы данных.

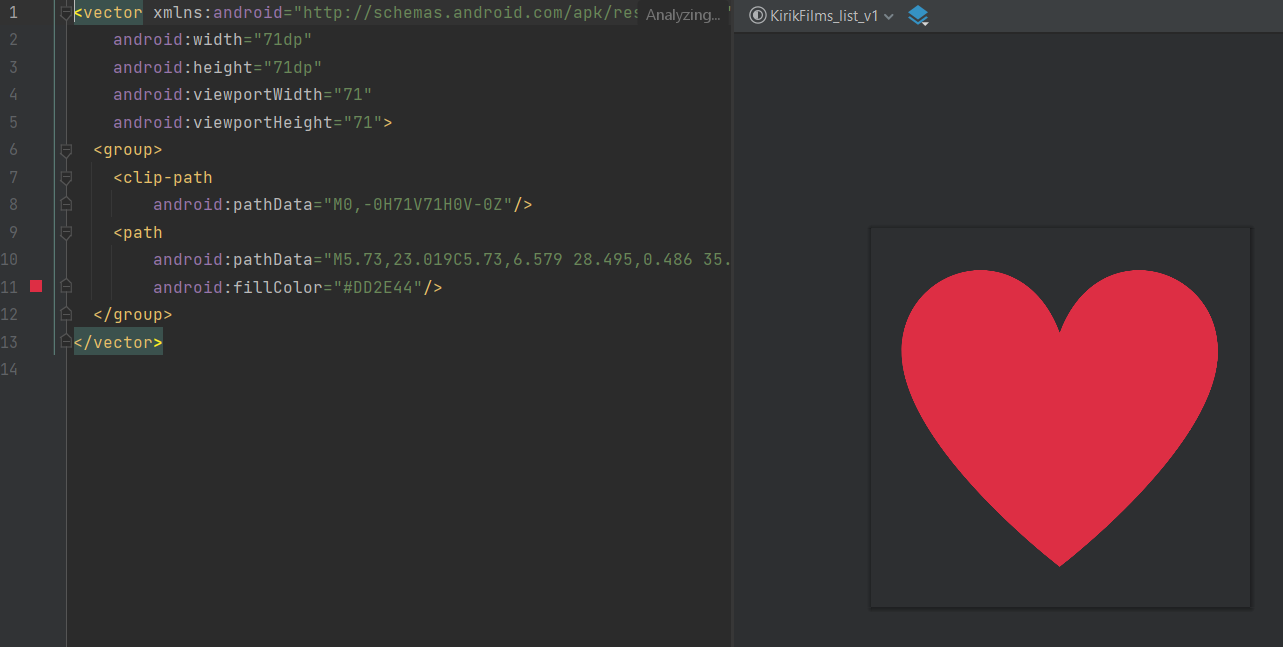


Рисунок 7 – Мультимедийный контент

Все основные цвета приложения хранятся в файле colors.xml

## Описание используемых плагинов

В проекте используются библиотеки с различными компонентами и функциями. Список всех библиотек, а также их описание представлен в таблице 2.

Таблица 2 – Библиотеки и их описание

|  |  |
| --- | --- |
| Библиотека | Описание |
| junit:junit:4.13.2 | Библиотека для написания и запуска модульных тестов в Java. |
| androidx.appcompat:appcompat:1.6.1 | Библиотека, обеспечивающая обратную совместимость с новыми функциями и возможностями Android для более старых версий платформы. |
| com.google.android.material:  material:1.11.0 | Библиотека, содержащая Material Design компоненты пользовательского интерфейса. |
| androidx.constraintlayout:  constraintlayout:2.1.4 | Библиотека для создания и управления экраном загрузки (splashscreen) в приложении. |
| androidx.appcompat:appcompat:1.6.1 | Библиотека, предоставляющая инструменты для создания сложных и гибких макетов в приложениях Android. |
| androidx.test.ext:junit:1.1.5 | Библиотека, расширяющая возможности JUnit для тестирования Android-приложений. |
| androidx.test.espresso:espresso-core:3.5.1 | Библиотека для автоматизированного UI-тестирования Android-приложений с использованием Espresso фреймворка. |
| com.github.bumptech.glide:  glide:4.16.0 | Библиотека для эффективной загрузки, кэширования и отображения изображений в Android приложениях. |
| com.github.bumptech.glide:  compiler:4.16.0 | Annotation processor для Glide, который генерирует необходимый код для оптимальной работы с изображениями. |

Продолжение таблицы 2.

|  |  |
| --- | --- |
| com.android.volley:volley:1.2.1 | Библиотека для управления сетевыми запросами, анализа ответов и кэширования в Android приложениях. |
| com.squareup.picasso:picasso:2.71828 | Библиотека для загрузки и кэширования изображений в Android приложениях, позволяющая легко отображать изображения из интернета или локальных источников. |

Данные библиотеки являются неотъемлемой частью приложения, обеспечивая его правильное функционирование и реализацию всех задуманных возможностей. Без этих библиотек приложение не сможет работать в полной мере, так как они предоставляют необходимые инструменты и функции, которые необходимы для его работы.

## Описание разработанных процедур и функций

В приложении реализованы следующие классы (таблица 3):

Таблица 3 – Классы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| DetailActivity | Класс DetailActivity отвечает за отображение подробной информации о выбранном фильме. Он получает данные о фильме, такие как название, описание, рейтинг, дата выхода, ссылка на трейлер, постер и список жанров, из Intent, который был передан при запуске этой активности. |
| GenresActivity | Класс GenresActivity отвечает за отображение фильмов, сгруппированных по жанрам. |
| MainActivity | Класс MainActivity отображает список фильмов с возможностью поиска по названия и различными сортировками. |
| Movie | Класс Movie представляет собой модель данных для фильма. Он содержит информацию о фильме, такую как название, постер, описание, рейтинг, дата выхода, URL трейлера и список жанров. |
| MovieAdapter | Класс Movie Adapter представляет собой адаптер для RecyclerView, который отвечает за отображение списка фильмов в пользовательском интерфейсе. Он связывает данные о фильмах с соответствующими элементами списка и обрабатывает взаимодействие пользователя с элементами списка. |
| TopActivity | Класс TopActivity отображает список избранных фильмов. |

В приложении реализованы следующие методы (таблица 4):

Таблица 4 – Методы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| onBackPressed() | Обрабатывает нажатие кнопки назад устройства |
| filter(String text) | Фильтрует фильмы по введенному тексту |
| fetchMovies() | Загружает список фильмов с сервера |
| showSortOptions() | Показывает опции сортировки фильмов |
| sortByRatingDescending() | Сортирует фильмы по убыванию рейтинга |
| sortByRatingAscending() | Сортирует фильмы по возрастанию рейтинга |
| sortByReleaseDateDescending() | Сортирует фильмы по убыванию даты выхода |
| sortByReleaseDateAscending() | Сортирует фильмы по возрастанию даты выхода |
| sortByOriginalOrder() | Сортирует фильмы в исходном порядке. |
| loadFavorites() | Загружает избранные фильмы из предпочтений пользователя |
| filterFavorites(List<Movie> allMovies) | Фильтрует избранные фильмы из полного списка |
| updateMovieList(List<Movie> movies) | Обновляет список фильмов в адаптере |
| getMovies() | Возвращает список всех фильмов |

# 4 Тестирование



## Протокол тестирования дизайна приложения

Тестирование дизайна приложения проводится на устройствах с различной диагональю экранов для проверки разметки страниц и вёрстки приложения.

Примеры проверок отображения элементов на экране представлены на рисунках 8, 9, 10, 11, где левый телефон Android SDK 34, а правый Android SDK 24.

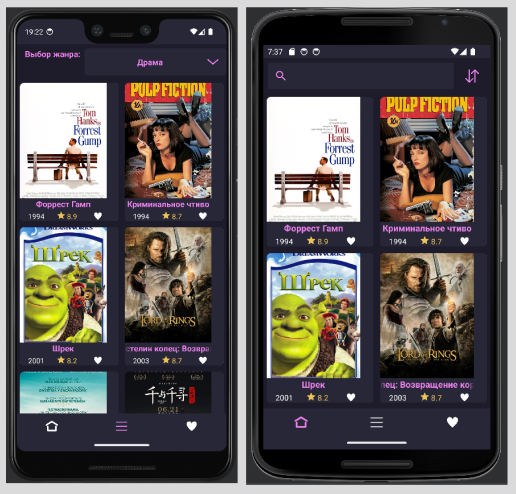


Рисунок 8 – Главный экран

Главный экран так же отображается корректно на обоих устройствах, все элементы интерфейса расположены на своих местах.

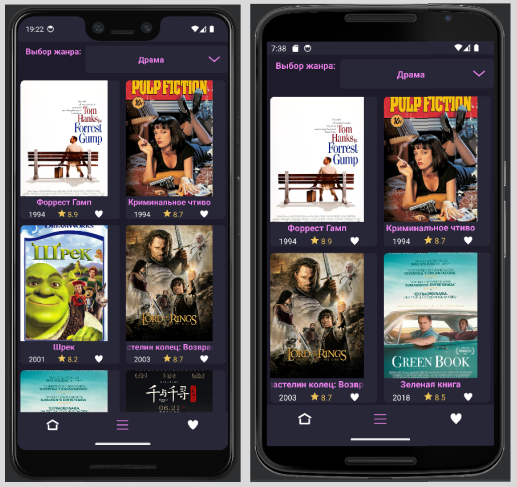


Рисунок 9 – Экран жанров

Экран жанров тоже отображается корректно. Верхняя и нижняя панель нормальных размеров, элементы на них не сдвинуты.

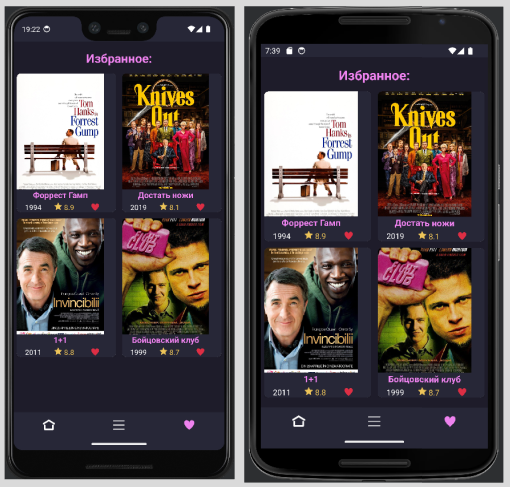


Рисунок 10 – Экран избранного

На экране избранного так же не обнаружено ошибок. Масштаб элементов сохранен. Верхняя панель находится на своём месте и имеет нормальные размеры.



Рисунок 11 – Экран детальной информации

Элементы экрана детальной информации отображаются корректно. Постер не поменял свои масштабы. Текст не растянут и не выходит за пределы экрана.

## Протокол тестирования функционала приложения

Так же необходимо проверить функционал приложения. Для этого, для каждой функции были разработаны TestCase.

В таблице 5 представлено тестирование функции вывода трейлера на экран.

Таблица 5 – Тестирование функции загрузки постеров

|  |  |  |
| --- | --- | --- |
| **Название**: | Киноафиша | |
| **Функция**: | Загрузка постера | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «Киноафиша» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Все постеры загрузились и отобразились | пройден |

В таблице 6 представлено тестирование функции добавления фильма в «Избранное».

Таблица 6 – Тестирование функции добавление фильма в «Избранное».

|  |  |  |
| --- | --- | --- |
| **Название**: | Киноафиша | |
| **Функция**: | Добавление видео в «Избранное» | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «Киноафиша» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Все постеры загрузились и отобразились | пройден |
| Нажать на кнопку добавления фильма в избранное | На устройстве сохранилась информации о списке избранных фильмов | пройден |
| Перейти на экран «Избранное» | Добавленный фильм отобразится в списке | пройден |

В таблице 7 представлено тестирование функции поиска фильмов.

Таблица 7 – Тестирование функции поиска фильмов

|  |  |  |
| --- | --- | --- |
| **Название**: | Киноафиша | |
| **Функция**: | Поиск фильмов | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «Киноафиша» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Все постеры загрузились и отобразились | пройден |
| Ввести в строку поиска название фильма | Список фильмов обновился и отобразил искомый фильм | пройден |

В таблице 8 представлено тестирование функции сортировки списка фильмов.

Таблица 8 – Тестирование функции сортировки списка фильмов

|  |  |  |
| --- | --- | --- |
| **Название**: | Киноафиша | |
| **Функция**: | Сортировка списка фильмов | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «Киноафиша» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Все постеры загрузились и отобразились | пройден |
| Нажать на кнопку сортировки | Отображается меню с вариантами сортировки | пройден |
| Выбрать требуемый вариант сортировки | Список фильмов отсортирован в соответствии с выбранным вариантом | пройден |

В таблице 9 представлено тестирование функции фильтрации по жанрам.

Таблица 9 – Тестирование функции фильтрации по жанрам

|  |  |  |  |
| --- | --- | --- | --- |
| **Название**: | Киноафиша | | |
| **Функция**: | Фильтрация по жанрам | | |
| **Действие** | **Ожидаемый** **результат** | | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | | |
| Запустить приложение «Киноафиша» | Приложение открылось корректно |  | |
|  |  |  | |
| **Шаги** **теста (positive)**: |  | | |
| Перейти на экран жанров | Все постеры загрузились и отобразились | пройден | |
| Выбрать жанр для фильтрации | Список фильмов обновился в соответствии с выбранным жанром | пройден | |

Разработанные TestCase демонстрируют корректную работу функций приложений. Так же 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 для просмотра информации о фильмах учтены предпочтения и потребности пользователей, которые интересуются кинематографом. Функция ведения списка избранных фильмов успешно реализована, позволяя пользователям отслеживать свои любимые фильмы и легко вернуться к уже просмотренной информации.

Интерфейс приложения разработан с учетом удобства использования и минимального количества действий для достижения пользовательской цели. Добавлены полезные функции, такие как просмотр трейлеров и возможность создания собственного списка избранных фильмов.

Важным элементом концепции приложения является возможность быстрого доступа к информации о фильмах, что позволяет пользователям узнавать о новинках кинематографа в дороге на работу, учебу или в перерывах между делами. Это удобный способ получать актуальную информацию на ходу.

Важным этапом в разработке приложения было тестирование дизайна, которое подтвердило его корректность и успешное функционирование. Это гарантирует, что разработанное приложение соответствует поставленным целям и требованиям.

В целом, разработка мобильного приложения для просмотра информации о фильмах позволила создать функциональное и удобное приложение. Проект достиг поставленной цели курсового проектирования и является основой для дальнейшего развития и улучшения приложения.

Библиография

Нормативно-правовые акты:

1. ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва: Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1. Android Developers [Электронный ресурс]. – Документация Android Studio. – URL: https://developer.android.com/docs/ (дата обращения: 07.03.2024)
2. Java HTML Parser [Электронный ресурс]. – Вытягивание данных с страницы. – URL: https://jsoup.org/ (дата обращения: 08.03.2024)
3. Material Design [Электронный ресурс]. – Bottom Navigation. – URL: https://material.io/components/bottom-navigation/android (дата обращения: 10.03.2024)
4. Figma [Электронный ресурс]. – Прототипирование проектов. – URL: https://www.figma.com/ (дата обращения: 02.03.2024)

Приложение А

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

РАЗРАБОТКА Мобильного приложения «киноафиша»

Техническое задание

НАТКиГ.200100.010.000ПЗ

Выполнил:

Студент группы ПР-21.101

Барсук К.Д.

2024

**Содержание**

Введение 25

1 Назначения разработки 26

2 Требования к мобильному приложению 26

2.1 Требования к функциональным характеристикам 26

2.2 Требования к надёжности 26

2.3 Условия эксплуатации 27

2.4 Требования к составу и параметрам технических средств 27

2.5 Требования к информационной и программной совместимости 27

2.6 Требования к защите информации 27

2.7 Требования к маркировке и упаковке 27

3 Требования к программной документации 27

4 Технико-экономические показатели 28

5 Стадии и этапы разработки 28

6 Порядок контроля и приёмки 29

**Введение**

Настоящее техническое задание распространяется на разработку мобильного приложения «Киноафиша», используемого для просмотра информации о фильмах и ведения списков избранного.

Наименование приложения: «Кинофиша».

Краткая характеристика области применения: мобильное приложение предоставляет возможность просмотра информации о фильмах, находящихся в базе данных, а также ведения списка избранных фильмов самого пользователя.

Основанием для проведения разработки является Протокол №6 от 21 февраля 2024 года.

Наименование темы разработки − «Разработка мобильного приложения Киноафиша».

Условное обозначение темы разработки – «Кинофиша».

**1 Назначение разработки**

Основное назначение приложения заключается в:

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

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 информацию о фильмах, загруженных в базу данных, а также вести список избранных фильмов.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* просмотр информации о фильмах;
* просмотр списка избранных фильмов;
* просмотр трейлеров фильмов;
* сортировка и фильтрация списка фильмов;

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация стабильного интернет-соединения.

**2.3 Условия эксплуатации**

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

**2.4 Требования к составу и параметрам технических средств**

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

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Java.

**2.6 Требования к защите информации**

Доступ к информации БД предоставляется только администратору базы данных.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* пояснительная записка.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

График курсового проектирования представлен в таблице 1.

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Этапы разработки КП** | **Сроки выполнения** | **Отчётность** |
| 1 | Определение цели и задач, объекта и предмета исследования | 24.02.2024 | Пояснительная записка |
| 2 | Описание предметной области | 25.02.2024 | Пояснительная записка |
| 3 | Выбор технологии, языка и среды программирования | 27.02.2024 | Пояснительная записка |
| 4 | Оформление технического задания | 29.02.2024 | Техническое задание |
| 5 | Проектирование UI/UX дизайна | 02.03.2024 | Спецификации программного обеспечения |
| 6 | Разработка мобильного приложения | 07.03.2024 | Схема структурная системы и спецификации компонентов |
| 7 | Разработка базы данных | 16.03.2024 | Программный  продукт |
| 8 | Отладка и тестирование приложения | 18.03.2024 | Тексты программных компонентов |
| 9 | Оформление документации | 20.03.2024 | Программная  документация |
| 10 | Защита | 29.04.2024 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

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

# Приложение Б

Метод для загрузки списка фильмов

private void fetchMovies() {

String url = "http://192.168.43.169:5000/movies";

JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(Request.Method.GET, url, null, new Response.Listener<JSONArray>() {

@Override

public void onResponse(JSONArray response) {

try {

movieList.clear();

for (int i = 0; i < response.length(); i++) {

JSONObject jsonObject = response.getJSONObject(i);

int id = jsonObject.getInt("id");

String title = jsonObject.getString("title");

String poster = jsonObject.getString("poster");

String overview = jsonObject.getString("overview");

double rating = jsonObject.getDouble("rating");

int date = jsonObject.getInt("release\_date");

String trailerUrl = jsonObject.getString("trailerurl");

// Извлечение массива жанров

List<String> genresList = new ArrayList<>();

JSONArray genresArray = jsonObject.getJSONArray("genres");

for (int j = 0; j < genresArray.length(); j++) {

genresList.add(genresArray.getString(j));

}

// Передача списка жанров в конструктор Movie

Movie movie = new Movie(id, title, poster, overview, rating, date, trailerUrl, genresList);

movieList.add(movie);

}

// Обновление адаптера RecyclerView

recyclerView.getAdapter().notifyDataSetChanged();

originalMovieList = new ArrayList<>(movieList); // Обновление списка для поиска и сортировки

} catch (JSONException e) {

e.printStackTrace();

Toast.makeText(MainActivity.this, "Error parsing JSON", Toast.LENGTH\_SHORT).show();

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

Toast.makeText(MainActivity.this, "Error fetching data", Toast.LENGTH\_SHORT).show();

}

});

recyclerView.setAdapter(new MovieAdapter(this, movieList));

requestQueue.add(jsonArrayRequest);

}