Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по УП

Задание3\_2. Приложение «Онлайн-кинотеатр» для Smart TV

Выполнила: Форрат Лия Константиновна

Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2024

**Описание задачи:**

****

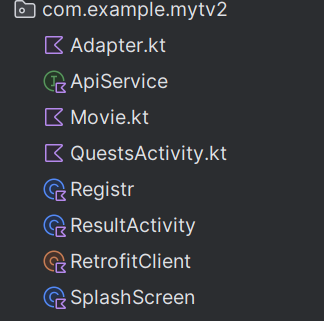
**Входные данные:** login - логин

password -пароль

seachEditText - поисковой запрос

**Выходные данные:** список фильмов

**Структура проекта:**

****

**Листинг программы:activity\_quests.xml**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/fon">  
  
 <!-- Строка поиска -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="8dp">  
  
 <EditText  
 android:id="@+id/searchEditText"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:hint="Введите название фильма"  
 android:padding="10dp"  
 android:background="@color/black"  
 android:textColor="@color/white"  
 android:textColorHint="@color/gray"  
 android:inputType="text" />  
  
 <Button  
 android:id="@+id/searchButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Поиск"  
 android:background="@drawable/button"  
 android:layout\_marginStart="8dp" />  
 </LinearLayout>  
  
 <!-- Спиннер с жанрами -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="8dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Жанры:"  
 android:textSize="18sp"  
 android:textColor="@color/white"  
 android:layout\_marginEnd="8dp" />  
  
 <Spinner  
 android:id="@+id/genreSpinner"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
 </LinearLayout>  
  
 <!-- Список фильмов -->  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <GridLayout  
 android:id="@+id/gridLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:columnCount="3"  
 android:padding="8dp" />  
 </ScrollView>  
</LinearLayout>

**activity\_registr.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/fon"  
 tools:context=".Registr">  
  
 <Button  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:background="@drawable/button"  
 android:onClick="onQui"  
 android:text="Sign In"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.465"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.773">  
  
 </Button>  
  
 <EditText  
 android:id="@+id/login"  
 android:layout\_width="340dp"  
 android:layout\_height="50dp"  
 android:backgroundTint="@color/gray"  
 android:hint="Login"  
 android:textColor="@color/white"  
 android:textColorHint="@color/gray"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.458"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.436">  
  
 </EditText>  
  
 <EditText  
 android:id="@+id/password"  
 android:layout\_width="340dp"  
 android:layout\_height="50dp"  
 android:backgroundTint="@color/gray"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:textColor="@color/white"  
 android:textColorHint="@color/gray"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.458"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.559">  
  
 </EditText>  
  
 <TextView  
 android:layout\_width="762dp"  
 android:layout\_height="113dp"  
 android:text="@string/lkdfjdf"  
 android:textAlignment="center"  
 android:textColor="@color/white"  
 android:textSize="80sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.387"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.107" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="70dp"  
 android:text="@string/some\_interesting\_films"  
 android:textColor="@color/white"  
 android:textSize="33sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.445"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.312" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_splash\_screen.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/fon"  
 tools:context=".SplashScreen">  
  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="200dp"  
 android:layout\_height="174dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.175"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.278"  
 tools:src="@drawable/kryg" />  
  
 <TextView  
 android:layout\_width="491dp"  
 android:layout\_height="98dp"  
 android:text="@string/lkdfjdf"  
 android:textColor="@color/white"  
 android:textSize="90dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.707"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.233" />  
  
 <TextView  
 android:layout\_width="529dp"  
 android:layout\_height="59dp"  
 android:text="@string/some\_interesting\_films"  
 android:textColor="@color/white"  
 android:textSize="50dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.742"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.417" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**ApiService.kt**

interface ApiService {  
 @GET("your\_endpoint\_here") // Замените на нужный эндпоинт  
 fun getMovieDetails(@Query("id") movieId: String): Call<Movie>  
}

**Register.kt**

class Registr : AppCompatActivity() {  
  
 lateinit var login: EditText  
 lateinit var password: EditText  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_registr*)  
  
 login = findViewById(R.id.*login*)  
 password = findViewById(R.id.*password*)  
 }  
  
 fun onQui(view: View) {  
 if (login.*text*.toString().*isNotEmpty*() && password.*text*.toString().*isNotEmpty*()) {  
 if(login.*text*.toString() == "ects" && password.*text*.toString() == "ects2024") {  
 val intent = Intent(this, QuestsActivity::class.*java*)  
 startActivity(intent)  
 }  
 else {  
 val alert = AlertDialog.Builder(this)  
 .setTitle("Ошибка")  
 .setMessage("Неверный логин или пароль!")  
 .setPositiveButton("ОК") **{** dialog, \_ **->** dialog.dismiss() **}** .create()  
 alert.show()  
 }  
 } else {  
 val alert = AlertDialog.Builder(this)  
 .setTitle("Ошибка")  
 .setMessage("Заполните оба поля!")  
 .setPositiveButton("ОК") **{** dialog, \_ **->** dialog.dismiss() **}** .create()  
 alert.show()  
 }  
 }  
}

**SplashScreen.kt**

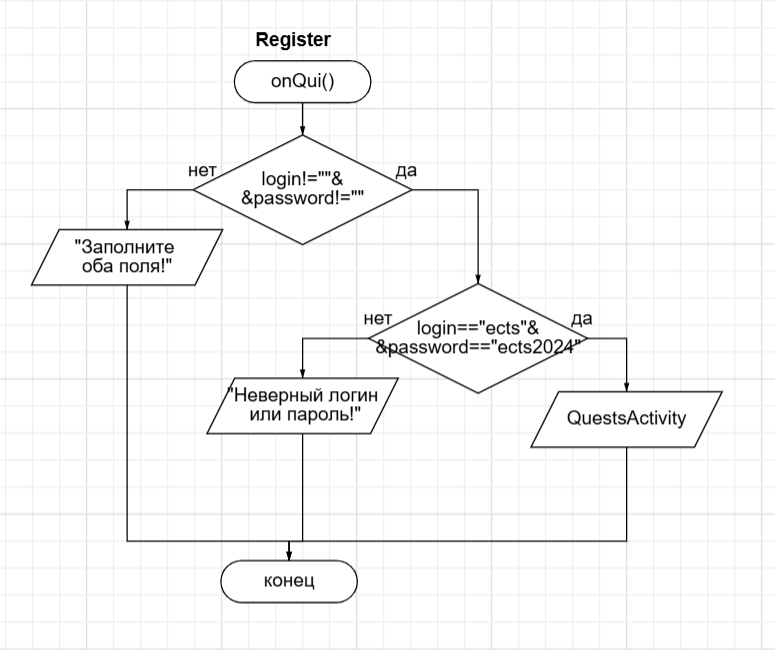
class SplashScreen : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_splash\_screen*)  
  
 val timer = object : CountDownTimer(3008, 1000) {  
 override fun onTick(millisUntilFinished: Long) {  
 }  
  
 override fun onFinish() {  
 val intent = Intent(this@SplashScreen, Registr::class.*java*)  
 startActivity(intent)  
 finish()  
 }  
 }  
 timer.start()  
 }  
}

**QuestsActivity.kt**

data class MovieResponse(  
 @SerializedName("docs") val docs: List<Movie>  
)  
  
data class Movie(  
 @SerializedName("alternativeName") val alternativeName: String,  
 @SerializedName("year") val year: Int,  
 @SerializedName("genres") val genres: List<Genre>,  
 @SerializedName("poster") val poster: Poster  
)  
  
data class Genre(  
 @SerializedName("name") val name: String  
)  
  
data class Poster(  
 @SerializedName("url") val url: String  
)  
  
class QuestsActivity : AppCompatActivity() {  
 private val client = OkHttpClient()  
 private val gson = Gson()  
 private lateinit var gridLayout: GridLayout  
 private lateinit var searchEditText: EditText  
 private lateinit var searchButton: Button  
 private lateinit var genreSpinner: Spinner  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_quests*)  
  
 gridLayout = findViewById(R.id.*gridLayout*)  
 searchEditText = findViewById(R.id.*searchEditText*)  
 searchButton = findViewById(R.id.*searchButton*)  
 genreSpinner = findViewById(R.id.*genreSpinner*)  
  
 // Устанавливаем жанры  
 val genres = *listOf*("драма", "комедия", "аниме")  
 val adapter = ArrayAdapter(this, android.R.layout.*simple\_spinner\_item*, genres)  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*)  
 genreSpinner.*adapter* = adapter  
  
 // Обработчик выбора жанра  
 genreSpinner.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(parent: AdapterView<\*>, view: View?, position: Int, id: Long) {  
 val selectedGenre = genres[position]  
 fetchMovies(selectedGenre)  
 }  
  
 override fun onNothingSelected(parent: AdapterView<\*>) {}  
 }  
  
 // Обработчик кнопки поиска  
 searchButton.setOnClickListener **{** val query = searchEditText.*text*.toString()  
 if (query.*isNotEmpty*()) {  
 searchMovie(query)  
 } else {  
 Toast.makeText(this, "Введите название фильма", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** }

private fun fetchMovies(selectedGenre: String) {  
 gridLayout.removeAllViews() // Очистка GridLayout перед добавлением новых элементов  
  
 val itemWidth = 1200 / 2 // Ширина для 3 колонок  
 val itemHeight = 600// Высота для 2 строк  
  
 val request = Request.Builder()  
 .url("https://api.kinopoisk.dev/v1.4/movie/search?page=1&limit=10&genres.name=${selectedGenre}") // Добавление жанра в запрос  
 .get()  
 .addHeader("accept", "application/json")  
 .addHeader("X-API-KEY", "YF53VNS-HXAMCHH-GYG66Q5-GZ7JKZD")  
 .build()  
  
 client.newCall(request).enqueue(object : Callback {  
 override fun onFailure(call: Call, e: IOException) {  
 Log.e("QuestsActivity", "Failed to fetch data", e)  
 }  
  
 override fun onResponse(call: Call, response: Response) {  
 if (response.isSuccessful) {  
 val responseData = response.body?.string()  
 Log.d("QuestsActivity", "Response: $responseData") // Логирование ответа  
  
 // Десериализация JSON в объект MovieResponse  
 val movieResponse = gson.fromJson(responseData, MovieResponse::class.*java*)  
  
 runOnUiThread **{** // Проверка на наличие фильмов и фильтрация по жанрам  
 var rows = 0  
 var columns = 0  
  
 movieResponse.docs.*forEach* **{** movie **->** Log.d("QuestsActivity", "Response: ${movie.genres}")  
  
 if (selectedGenre == "All" || movie.genres.*any* **{  
 it**.name.*equals*(  
 selectedGenre,  
 ignoreCase = true  
 )  
 **}**) {  
 val linearLayout = LinearLayout(this@QuestsActivity).*apply* **{** *orientation* = LinearLayout.*VERTICAL  
 layoutParams* = GridLayout.LayoutParams().*apply* **{** rowSpec = GridLayout.spec(rows)  
 columnSpec = GridLayout.spec(columns)  
 **}** setPadding(16, 16, 16, 16)  
 setBackgroundResource(R.drawable.*ramka*)  
 val layoutParams = *layoutParams* as GridLayout.LayoutParams  
 layoutParams.setMargins(8, 8, 8, 8)  
 **}** val imageView = ImageView(this@QuestsActivity).*apply* **{** *layoutParams* = LinearLayout.LayoutParams(itemWidth, itemHeight)  
 *scaleType* = ImageView.ScaleType.*FIT\_CENTER* movie.poster?.*let* **{** // Используйте movie.poster напрямую  
 Picasso.get().load(**it**.url)  
 .into(this) // Загрузка изображения с помощью Picasso  
 **}  
 }** linearLayout.addView(imageView)  
  
 val titleTextView = TextView(this@QuestsActivity).*apply* **{** *text* = movie.alternativeName // Используйте alternativeName  
 setTextColor(Color.parseColor("#FFFFFF"))  
 *layoutParams* =  
 LinearLayout.LayoutParams(itemWidth, itemHeight / 10)  
 *gravity* = Gravity.*CENTER* **}** val yearTextView = TextView(this@QuestsActivity).*apply* **{** *text* = movie.year.toString()  
 setTextColor(Color.parseColor("#FFFFFF"))  
 *layoutParams* =  
 LinearLayout.LayoutParams(itemWidth, itemHeight / 10)  
 *gravity* = Gravity.*CENTER* **}** linearLayout.addView(titleTextView)  
 linearLayout.addView(yearTextView)  
  
 gridLayout.addView(linearLayout)  
  
 columns += 1  
 if (columns == 3) {  
 rows += 1  
 columns = 0  
 }  
 }  
 **}  
 }** } else {  
 Log.e("QuestsActivity", "Response error: ${response.code}")  
 }  
 }  
 })  
}  
private fun searchMovie(query: String) {  
 gridLayout.removeAllViews()  
  
 val request = Request.Builder()  
 .url("https://api.kinopoisk.dev/v1.4/movie/search?page=1&limit=1&name=$query")  
 .get()  
 .addHeader("accept", "application/json")  
 .addHeader("X-API-KEY", "YF53VNS-HXAMCHH-GYG66Q5-GZ7JKZD")  
 .build()  
  
 client.newCall(request).enqueue(object : Callback {  
 override fun onFailure(call: Call, e: IOException) {  
 Log.e("QuestsActivity", "Failed to fetch data", e)  
 }  
  
 override fun onResponse(call: Call, response: Response) {  
 if (response.isSuccessful) {  
 val responseData = response.body?.string()  
 val movieResponse = gson.fromJson(responseData, MovieResponse::class.*java*)  
  
 runOnUiThread **{** if (movieResponse.docs.*isNotEmpty*()) {  
 /\*displayMovies(movieResponse.docs)\*/  
 } else {  
 Toast.makeText(this@QuestsActivity, "Фильм не найден", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** } else {  
 Log.e("QuestsActivity", "Response error: ${response.code}")  
 }  
 }  
 })  
}

**Блок схемы:**

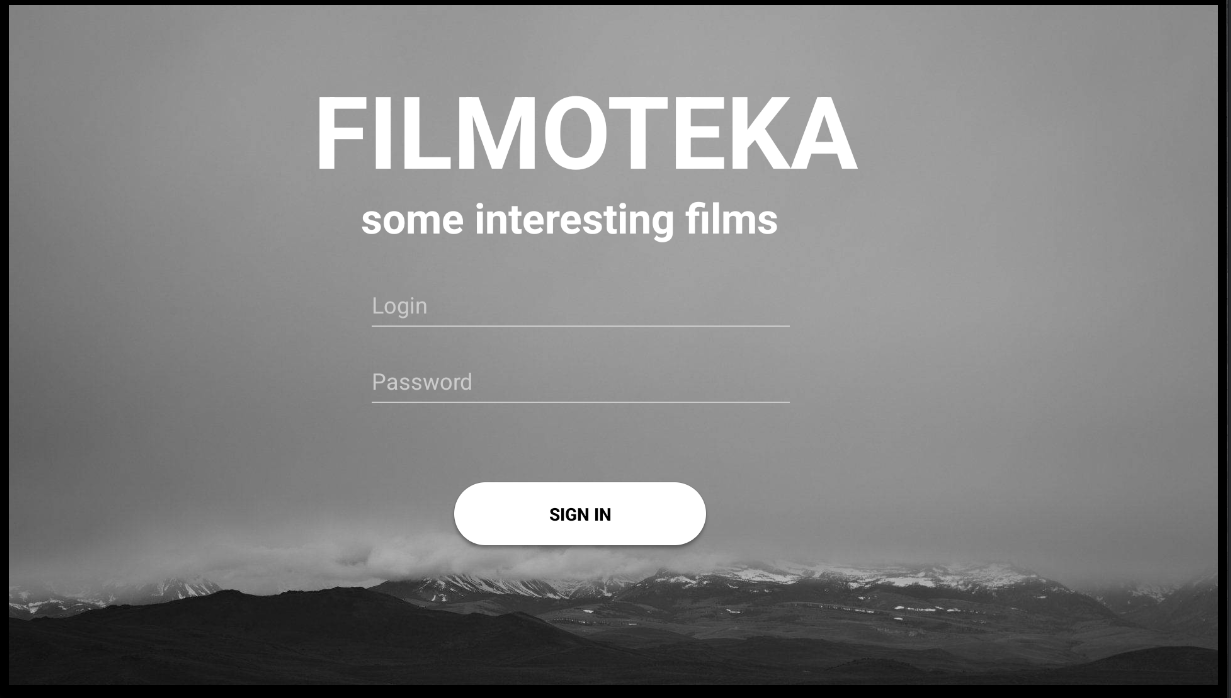
****

**Интерфейс:**

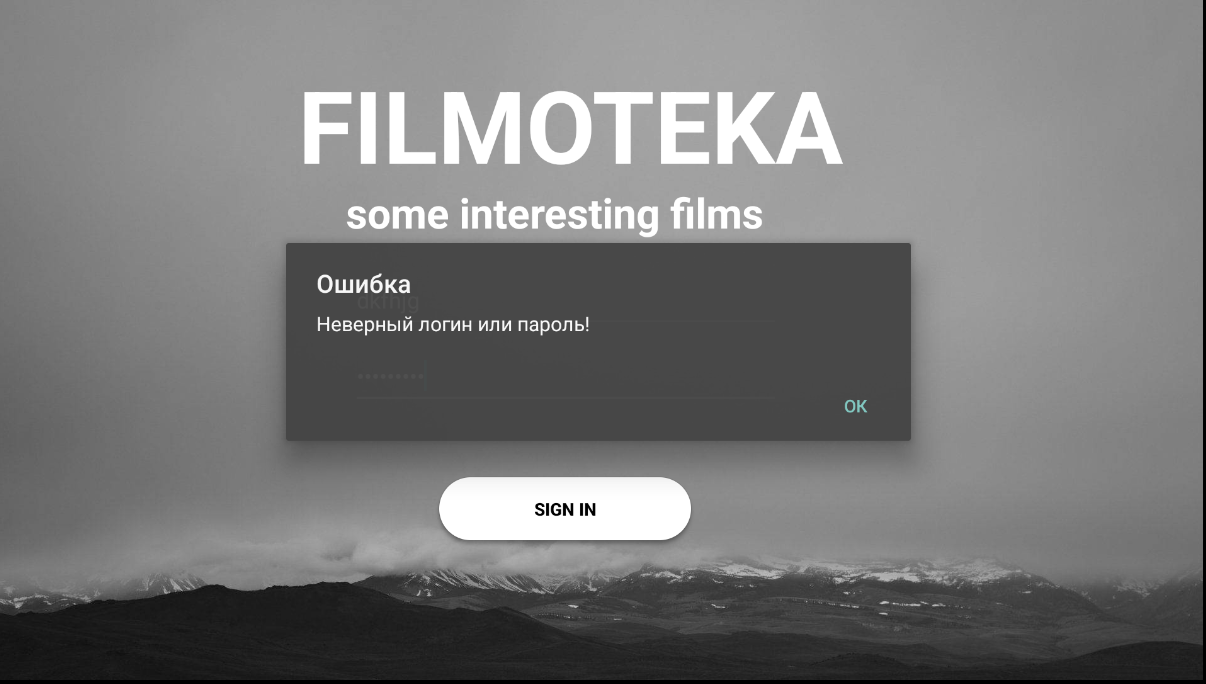
Экран при запуске, горит пару секунд потом исчезает:



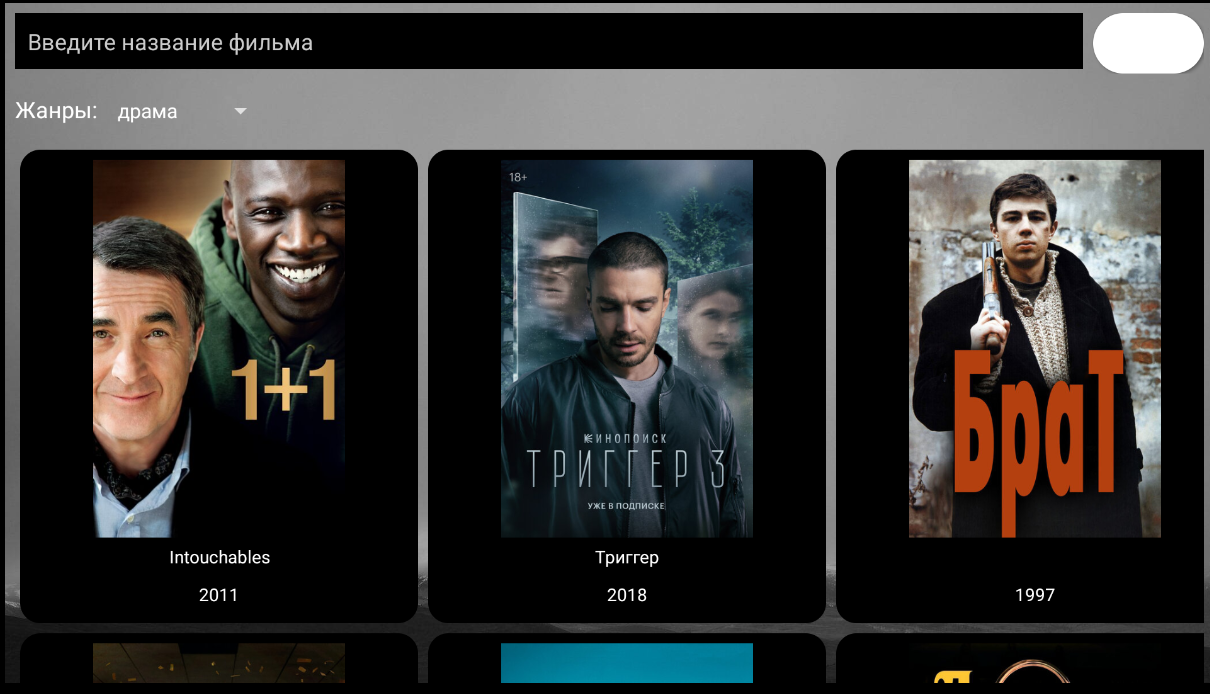
Экран авторизации:



При вводе неверных данных появляется ошибка:



Экран с поиском и выбором жанра фильмов или сериалов:



**Иконка** 

**Используемые библиотеки:**

|  |  |
| --- | --- |
| android.annotation.SuppressLint | Эта аннотация используется для подавления предупреждений компилятора о потенциальных проблемах в коде |
| android.graphics.Color | Этот класс предоставляет константы и методы для работы с цветами |
| androidx.appcompat.app.AppCompatActivity | AppCompatActivity — это базовый класс для активностей, который обеспечивает поддержку старых версий Android и позволяет использовать компоненты Material Design |
| android.os.Bundle | Класс Bundle используется для передачи данных между активностями и сохранения состояния активности |
| android.widget.GridLayout | GridLayout — это контейнер, который позволяет размещать дочерние элементы в виде сетки |
| android.widget.ImageView | Этот класс представляет элемент пользовательского интерфейса, который отображает изображения. |
| android.widget.LinearLayout | LinearLayout — это контейнер, который размещает дочерние элементы в одном направлении (горизонтально или вертикально) |
| android.widget.TextView | Этот класс представляет собой элемент пользовательского интерфейса для отображения текста |
| com.google.gson.Gson | Gson — это библиотека для работы с JSON в Java и Kotlin. Она позволяет легко сериализовать и десериализовать объекты, преобразуя их в JSON и обратно |
| com.squareup.picasso.Picasso | Picasso — это библиотека для загрузки и кэширования изображений для Android. Она упрощает процесс загрузки изображений из интернета в ImageView |
| okhttp3 | OkHttp — это мощный HTTP-клиент для приложений Android и Java. Он используется для отправки и получения сетевых запросов на основе HTTP |

**Используемые инструменты:**

Android Stusio, Kotlin