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: Рубцова Диана Алексеевна

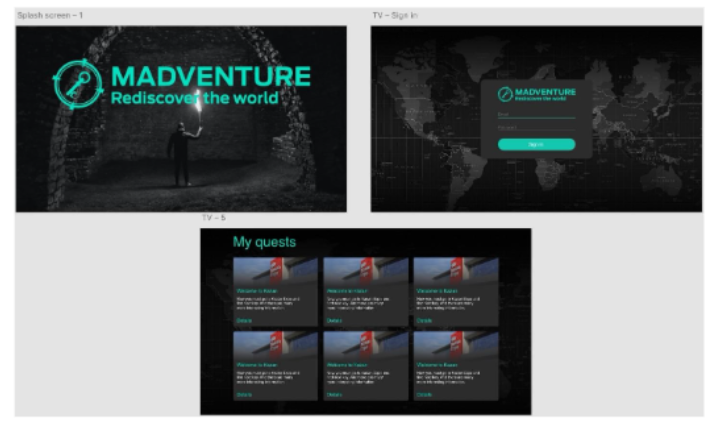
Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2024

**Описание задачи:**

****



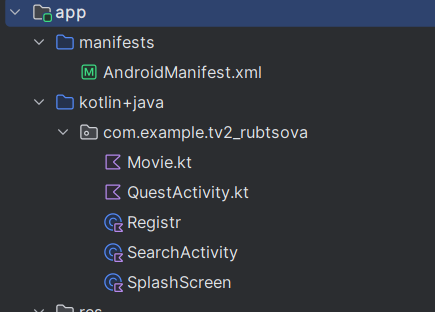
**Входные данные:** login - логин

password -пароль

searchBar- поисковой запрос

**Выходные данные:** список фильмов

**Структура проекта:**



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

<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/map"  
 tools:context=".QuestsActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="269dp"  
 android:layout\_height="49dp"  
 android:text="My films"  
 android:textSize="40dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.057"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.032" />  
  
 <ScrollView  
 android:id="@+id/scrollView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="450dp"  
 android:fillViewport="true"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="1.0">  
  
 <GridLayout  
 android:id="@+id/gridLayout"  
 android:layout\_width="930dp"  
 android:layout\_height="wrap\_content"  
 android:columnCount="10"  
 android:rowCount="10"  
 tools:ignore="MissingConstraints" />  
  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_registr.xml**

<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"  
 tools:context=".Registr"  
 android:background="@drawable/map">  
 <LinearLayout  
 android:background="@drawable/ll"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="60dp"  
 android:layout\_marginTop="10dp"  
 android:src="@drawable/log" />  
 <EditText  
  
 android:id="@+id/login"  
 android:textSize="12dp"  
 android:layout\_width="200dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="60dp"  
 android:layout\_marginTop="10dp"  
 android:hint="Login"  
 android:textColorHint="#3A3A3A"/>  
 <EditText  
 android:id="@+id/password"  
 android:textSize="12dp"  
 android:layout\_width="200dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="60dp"  
 android:layout\_marginTop="10dp"  
 android:hint="Password"  
 android:textColorHint="#3A3A3A"/>  
 <Button  
  
 android:layout\_width="200dp"  
 android:layout\_height="30dp"  
 android:text="Sign in"  
 android:textColor="#FFFFFF"  
 android:background="@drawable/bt"  
 android:gravity="center"  
 android:onClick="onQui"  
 android:layout\_marginLeft="60dp"  
 android:layout\_marginTop="10dp"/>  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_splash\_screen.xml**

<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/map"  
 tools:context=".SplashScreen">  
  
  
 <ImageView  
 android:layout\_width="376dp"  
 android:layout\_height="172dp"  
 android:src="@drawable/log"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.143" />  
  
 <TextView  
 android:layout\_width="354dp"  
 android:layout\_height="87dp"  
 android:text="FILMS.RU"  
 android:textColor="#FFFFFF"  
 android:textSize="60dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.549" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_search.xml**

<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"  
 tools:context=".SearchActivity"  
 android:background="@drawable/map">  
 <LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 >  
  
 <EditText  
 android:id="@+id/searchBar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Введите название фильма"  
 android:background="@color/lb\_tv\_white"  
 android:padding="12dp"  
 android:textColor="#000000"  
 android:textColorHint="#000000"  
 android:layout\_marginBottom="16dp" />  
  
 <Spinner  
 android:id="@+id/categorySpinner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/lb\_tv\_white"  
 android:textColorHint="#000000"  
 android:textColor="#000000"  
  
 android:layout\_marginBottom="16dp" />  
  
 <Button  
 android:id="@+id/searchButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Найти"  
 android:background="@color/lb\_tv\_white"  
 android:textColor="#000000"  
 android:textColorHint="#000000"  
 android:layout\_marginBottom="16dp" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:id="@+id/resultsLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical" />  
 </ScrollView>  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

**adapter.xml**

<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="#333333"  
 android:padding="8dp"  
 android:layout\_margin="4dp"  
 android:clipToPadding="false">  
  
 <ImageView  
 android:id="@+id/ivPoster"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:scaleType="centerCrop" />  
  
 <TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="16sp"  
 android:paddingTop="8dp" />  
  
 <TextView  
 android:id="@+id/tvYear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="#AAAAAA"  
 android:textSize="14sp" />  
</LinearLayout>

**Movie.kt**

import com.google.gson.annotations.SerializedName  
data class MovieResponse(  
 val docs: List<Movie>  
)  
  
  
data class Movie(  
 @SerializedName("alternativeName") val title: String,  
 @SerializedName("year") val year: Int,  
 @SerializedName("poster") val poster: Poster  
)  
  
data class Poster(  
 @SerializedName("url") val image: String? // URL постера может быть null  
)

**Registr.kt**

import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.widget.EditText  
  
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**

import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.os.CountDownTimer  
  
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**

import android.graphics.Color  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.view.Gravity  
import android.widget.GridLayout  
import android.widget.ImageView  
import android.widget.LinearLayout  
import android.widget.TextView  
import com.google.gson.Gson  
import com.google.gson.annotations.SerializedName  
import com.squareup.picasso.Picasso  
import okhttp3.Call  
import okhttp3.Callback  
import okhttp3.OkHttpClient  
import okhttp3.Response  
import java.io.IOException  
  
import android.content.Intent  
import android.widget.Button  
import okhttp3.\*  
  
  
class QuestsActivity : AppCompatActivity() {  
  
 private val stringArray: Array<String> =  
 *arrayOf*("Inception", "The Matrix", "Interstellar", "The Shawshank Redemption", "Pulp Fiction", "The Godfather")  
  
 lateinit var gridLayout: GridLayout  
 private val client = OkHttpClient()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_quest*)  
 gridLayout = findViewById(R.id.*gridLayout*)  
  
 // Добавление кнопки поиска фильмов  
 val searchButton = Button(this).*apply* **{** *text* = "Поиск фильмов"  
 setOnClickListener **{** val intent = Intent(this@QuestsActivity, SearchActivity::class.*java*)  
 startActivity(intent)  
 **}  
 }** // Добавляем кнопку в GridLayout  
 val buttonLayoutParams = GridLayout.LayoutParams().*apply* **{** rowSpec = GridLayout.spec(0) // Первая строка  
 columnSpec = GridLayout.spec(0, 3) // Занимает 3 колонки  
 **}** searchButton.*layoutParams* = buttonLayoutParams  
 gridLayout.addView(searchButton)  
  
 val itemWidth = 1800 / 3  
 var rows: Int = 1 // Начинаем со второй строки, так как первая занята кнопкой  
 var columns: Int = 0  
  
 stringArray.*forEach* **{** query **->** val request = okhttp3.Request.Builder()  
 .url("https://api.kinopoisk.dev/v1.4/movie/search?page=1&limit=10&query=$query&language=ru") // Added language parameter  
 .get()  
 .addHeader("accept", "application/json")  
 .addHeader("X-API-KEY", "Q5R3EPC-BRZMKJ2-Q7H0HKQ-D5MPGJ4")  
 .build()  
  
 client.newCall(request).enqueue(object : Callback {  
 override fun onFailure(call: Call, e: IOException) {  
 // Обработка ошибки  
 Log.e("QuestsActivity", "Failed to fetch data", e)  
 }  
  
 override fun onResponse(call: Call, response: Response) {  
 if (response.isSuccessful) {  
 val responseData = response.body?.string()  
 val movie = parseMovieJson(responseData)  
  
 runOnUiThread **{** if (movie != null) {  
 val linearLayout = LinearLayout(this@QuestsActivity).*apply* **{** *orientation* = LinearLayout.*VERTICAL  
 layoutParams* = GridLayout.LayoutParams().*apply* **{** rowSpec = GridLayout.spec(rows)  
 columnSpec = GridLayout.spec(columns)  
 **}** *gravity* = Gravity.*CENTER* **}** val imageView = ImageView(this@QuestsActivity).*apply* **{** *layoutParams* = LinearLayout.LayoutParams(300, 550)  
 *scaleType* = ImageView.ScaleType.*FIT\_CENTER* movie.poster.image?.*let* **{** Picasso.get().load(**it**).into(this)  
 **}  
 }** linearLayout.addView(imageView)  
  
 val titleTextView = TextView(this@QuestsActivity).*apply* **{** *text* = movie.title // Title in Russian  
 setTextColor(Color.*WHITE*)  
 *layoutParams* = LinearLayout.LayoutParams(itemWidth, LinearLayout.LayoutParams.*WRAP\_CONTENT*)  
 *gravity* = Gravity.*CENTER* **}** val yearTextView = TextView(this@QuestsActivity).*apply* **{** *text* = movie.year.toString()  
 setTextColor(Color.*WHITE*)  
 *layoutParams* = LinearLayout.LayoutParams(itemWidth, LinearLayout.LayoutParams.*WRAP\_CONTENT*)  
 *gravity* = Gravity.*CENTER* **}** linearLayout.addView(titleTextView)  
 linearLayout.addView(yearTextView)  
  
 gridLayout.addView(linearLayout)  
  
 columns += 1  
 if (columns == 3) {  
 rows += 1  
 columns = 0  
 }  
 }  
 **}** }  
 }  
 })  
 **}** }  
  
 private fun parseMovieJson(jsonString: String?): Movie? {  
 if (jsonString.*isNullOrEmpty*()) {  
 return null  
 }  
 val gson = Gson()  
 val movieResponse = gson.fromJson(jsonString, MovieResponse::class.*java*)  
 return movieResponse.docs.*firstOrNull*()  
 }  
  
 data class MovieResponse(  
 @SerializedName("docs") val docs: List<Movie>  
 )  
}

**SearchActivity.kt**

import android.graphics.Color  
import android.os.Bundle  
import android.widget.\*  
import androidx.appcompat.app.AppCompatActivity  
import com.google.gson.Gson  
import com.google.gson.annotations.SerializedName  
import okhttp3.\*  
import java.io.IOException  
import com.squareup.picasso.Picasso  
import android.widget.LinearLayout  
import android.widget.TextView  
import android.widget.ImageView  
import android.view.Gravity  
class SearchActivity : AppCompatActivity() {  
  
 private lateinit var searchBar: EditText  
 private lateinit var categorySpinner: Spinner  
 private lateinit var searchButton: Button  
 private lateinit var resultsLayout: LinearLayout  
 private val client = OkHttpClient()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_search*)  
  
 // Инициализация элементов интерфейса  
 searchBar = findViewById(R.id.*searchBar*)  
 categorySpinner = findViewById(R.id.*categorySpinner*)  
 searchButton = findViewById(R.id.*searchButton*)  
 resultsLayout = findViewById(R.id.*resultsLayout*)  
  
 // Настройка категорий  
 val categories = *arrayOf*("Все", "Боевик", "Комедия", "Драма", "Ужасы", "Фантастика")  
 val adapter = ArrayAdapter(this, android.R.layout.*simple\_spinner\_item*, categories)  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*)  
 categorySpinner.*adapter* = adapter  
  
 // Обработка нажатия на кнопку поиска  
 searchButton.setOnClickListener **{** val query = searchBar.*text*.toString()  
 val category = categorySpinner.*selectedItem*.toString()  
 searchMovies(query, category)  
 **}** }  
  
 private fun searchMovies(query: String, category: String) {  
 // Формирование URL для запроса  
 val url = "https://api.kinopoisk.dev/v1.4/movie/search?page=1&limit=10&query=$query&language=ru"  
 val request = Request.Builder()  
 .url(url)  
 .get()  
 .addHeader("accept", "application/json")  
 .addHeader("X-API-KEY", "Q5R3EPC-BRZMKJ2-Q7H0HKQ-D5MPGJ4")  
 .build()  
  
 // Выполнение запроса  
 client.newCall(request).enqueue(object : Callback {  
 override fun onFailure(call: Call, e: IOException) {  
 runOnUiThread **{** Toast.makeText(this@SearchActivity, "Ошибка загрузки данных", Toast.*LENGTH\_SHORT*).show()  
 **}** }  
  
 override fun onResponse(call: Call, response: Response) {  
 if (response.isSuccessful) {  
 val responseData = response.body?.string()  
 val movies = parseMoviesJson(responseData)  
  
 runOnUiThread **{** if (movies.isEmpty()) {  
 // Если фильмы не найдены, показываем сообщение  
 Toast.makeText(this@SearchActivity, "Фильмы не найдены", Toast.*LENGTH\_SHORT*).show()  
 } else {  
 displayMovies(movies)  
 }  
 **}** } else {  
 runOnUiThread **{** Toast.makeText(this@SearchActivity, "Ошибка загрузки данных", Toast.*LENGTH\_SHORT*).show()  
 **}** }  
 }  
 })  
 }  
  
 private fun parseMoviesJson(jsonString: String?): List<Movie> {  
 if (jsonString.*isNullOrEmpty*()) {  
 return *emptyList*()  
 }  
 val gson = Gson()  
 val movieResponse = gson.fromJson(jsonString, MovieResponse::class.*java*)  
 return movieResponse.docs  
 }  
  
 private fun displayMovies(movies: List<Movie>) {  
 resultsLayout.removeAllViews()  
 movies.*forEach* **{** movie **->** // Создаем линейный макет для каждого фильма  
 val movieLayout = LinearLayout(this).*apply* **{** *orientation* = LinearLayout.*VERTICAL* setPadding(8, 8, 8, 8)  
 *gravity* = Gravity.*CENTER* // Устанавливаем gravity для всего макета  
 **}** // Добавляем обложку фильма  
 val imageView = ImageView(this).*apply* **{** *layoutParams* = LinearLayout.LayoutParams(300, 450) // Установите размеры по вашему усмотрению  
 *scaleType* = ImageView.ScaleType.*FIT\_CENTER* movie.poster.image?.*let* **{** Picasso.get().load(**it**).into(this) // Загрузка изображения с помощью Picasso  
 **}  
 }** movieLayout.addView(imageView)  
  
 // Добавляем название фильма  
 val titleTextView = TextView(this).*apply* **{** *text* = "${movie.title} (${movie.year})"  
 setTextColor(Color.*WHITE*)  
 *textSize* = 16f  
 *gravity* = Gravity.*CENTER* // Устанавливаем gravity для текста  
 **}** titleTextView.*layoutParams* = LinearLayout.LayoutParams(  
 LinearLayout.LayoutParams.*WRAP\_CONTENT*,  
 LinearLayout.LayoutParams.*WRAP\_CONTENT* ).*apply* **{** gravity = Gravity.*CENTER* // Устанавливаем gravity для текста  
 **}** movieLayout.addView(titleTextView)  
  
 // Добавляем линейный макет в основной layout  
 resultsLayout.addView(movieLayout)  
 **}** }  
  
 data class MovieResponse(  
 @SerializedName("docs") val docs: List<Movie>  
 )  
}

**Интерфейс:**

Экран при запуске, горит пару секунд потом исчезает:

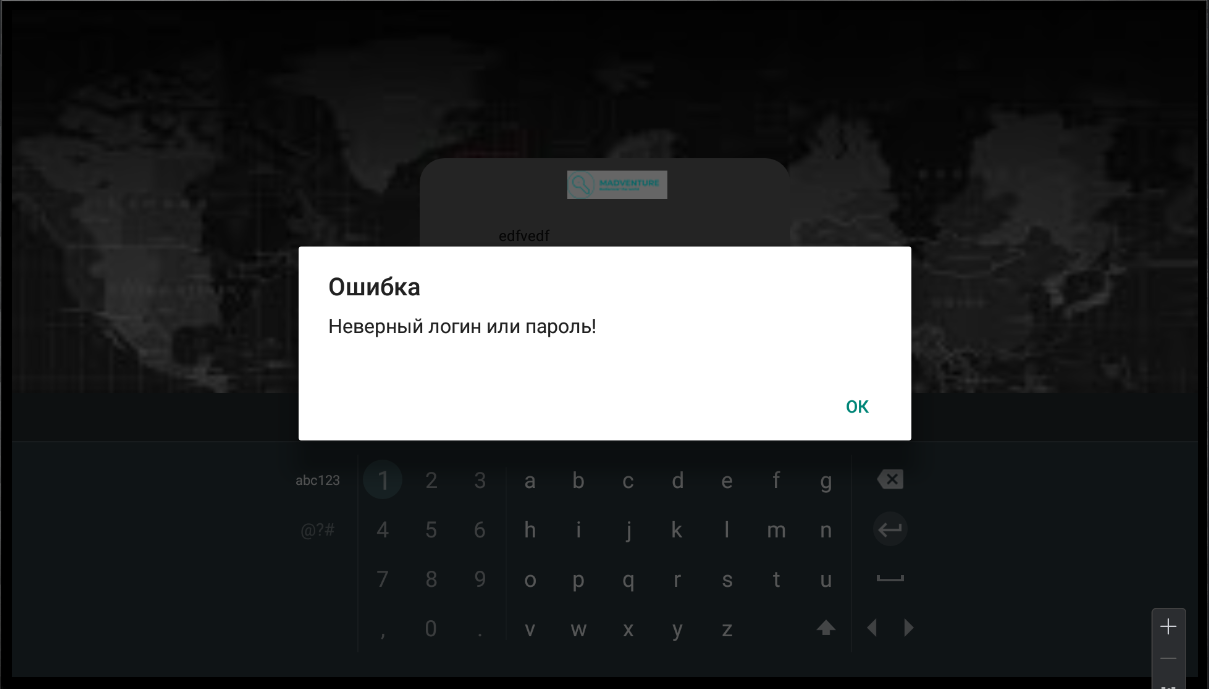


Экран авторизации:

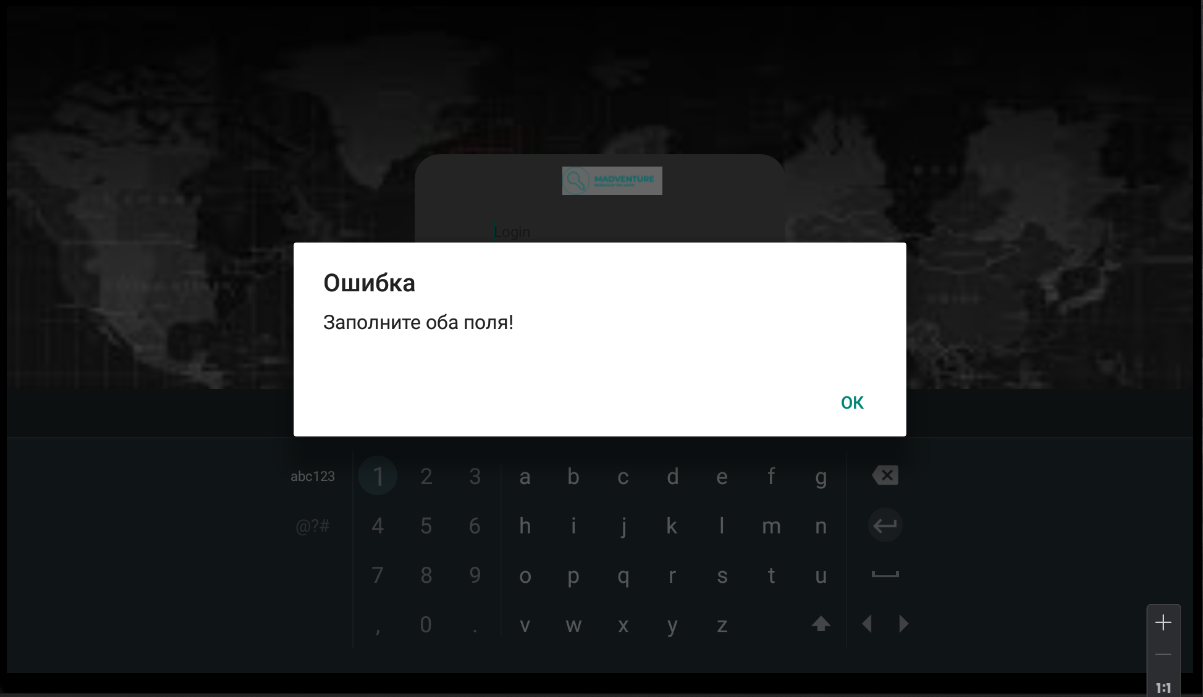


При вводе неверных данных появляется ошибка

(логин у нас ects , а пароль - ects 2024) :



Если какое -то из полей или оба не заполненные:



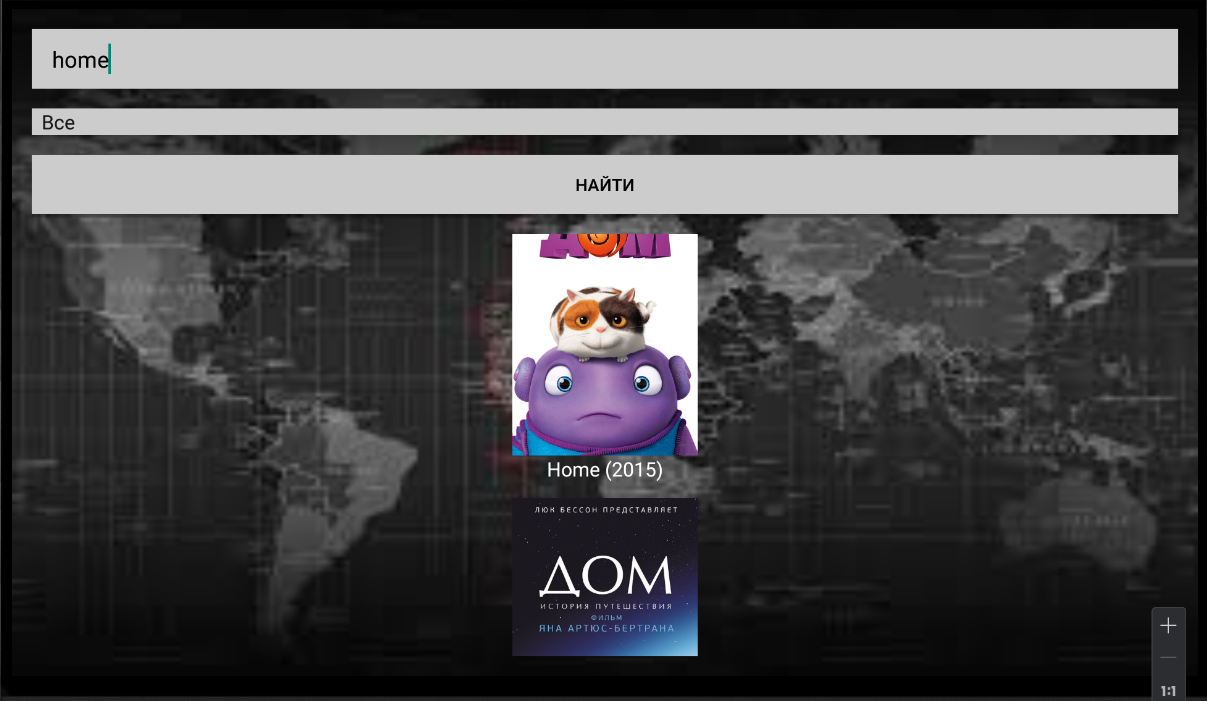
При переходе на следующий экран пользователь увидит список фильмов:



Далее, что бы найти фильм по определённым критериям , пользователь нажимает на кнопку «Поиск фильмов»







**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