СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc57574900)

[1. ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ 4](#_Toc57574901)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 5](#_Toc57574902)

[2.1. Постановка задачи 5](#_Toc57574903)

[2.1.1. Постановка задачи 5](#_Toc57574904)

[2.1.2. Подробные требования к проекту 5](#_Toc57574905)

[2.2. Внешняя спецификация 6](#_Toc57574906)

[2.2.1. Описание задачи 6](#_Toc57574907)

[2.3. Результат работы программы 6](#_Toc57574908)

[2.4. Инструментальные средства 6](#_Toc57574909)

[2.5. Характеристика программы 7](#_Toc57574910)

[3. ВЫВОД 8](#_Toc57574911)

[СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 9](#_Toc57574912)

ПРИЛОЖЕНИЕ А. Исходный текст программы

ПРИЛОЖЕНИЕ Б. Руководство пользователя

# ВВЕДЕНИЕ

Необходимо разработать мобильное приложение «PerfectMovie». Данные необходимо получать, используя The Movie Database API (<https://www.themoviedb.org/documentation/api>). Приложение разработать, используя IDE Android Studio и язык программирования Kotlin.

# ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ

Для прохождения учебной практики использовался персональный компьютер Dell G3 3579, обладающий характеристиками, представленными в Таблице 1.

Таблица 1 - Технические средства разработки

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ноутбук Dell G3 3579 | | |
| 1 | Центральный процессор: | Intel Core i7-8750H |
| 2 | Видеоадаптер: | Intel UHD Graphics 630, 64+1632 МБ |
| 3 | Оперативная память: | DDR4 16GB 2666MHz |
| 4 | Твердотельный накопитель: | 128 GB SSD |

Для разработки приложения и технической документации использовались программные средства и среды, описанные в Таблице 2.

Таблица 2 – Программные средства

| № | Тип средства | Название средства | Назначение |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | Инструментальное средство разработки программных решений | Android Studio 4.1 | Разработка мобильного приложения |

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

### Постановка задачи

Необходимо разработать мобильное приложение «PerfectMovie». Данные необходимо получать, используя The Movie Database API (<https://www.themoviedb.org/documentation/api>). Приложение разработать, используя IDE Android Studio и язык программирования Kotlin.

### Подробные требования к проекту

Требования к проекту:

1. Данные необходимо хранить в моделях;

2. Необходимо использовать сериализацию в моделях;

3. Базовая ссылка должна храниться в классе константы;

4. API key должны храниться в Gradle;

5. Все методы API должны быть описаны в интерфейсе;

6. Шрифты должны быть адаптивными;

7. Использовать стили для схожих элементов со схожими свойствами;

8. Необходимо использовать анимации;

9. Вовремя подгрузки данных отобразить пользователю, данный процесс следующим образом (Рисуок 1).



Рисунок 1

Функционал:

В приложении должен быть BottomNavigation c 2 пунктами:

* Ожидаемые. Отображается список ожидаемых фильмов. Каждый элемент списка состоит из названия фильма, обложки, рейтинга и даты выхода. Нажав на элемент списка, можно перейти к экрану с информацией о фильме (Название, дата выхода, описание, обложка, рейтинг);
* Топ рейтинг. Отображается список топ фильмов. Каждый элемент списка состоит из названия фильма, обложки и рейтинга. Нажав на элемент списка, можно перейти к экрану с информацией о фильме (Название, дата выхода, описание, обложка, рейтинг).

## Внешняя спецификация

### Описание задачи

Визуальное оформление.

## Результат работы программы

На рисунке 4 представлен результат работы программы.

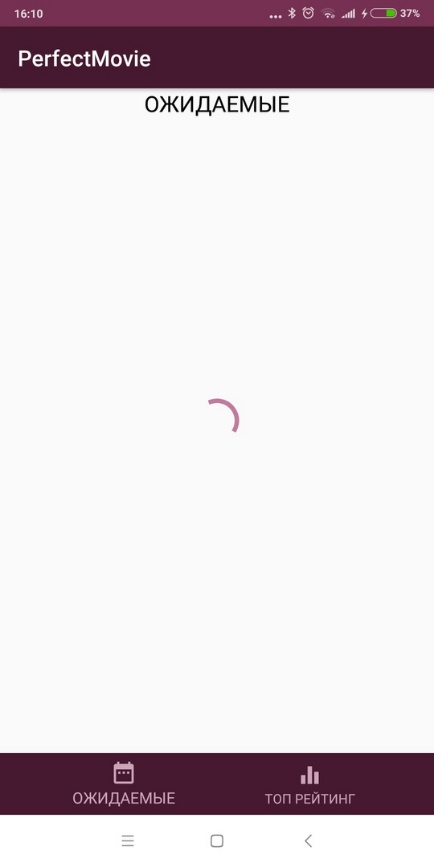


Рисунок 2 - Загрузка данных.

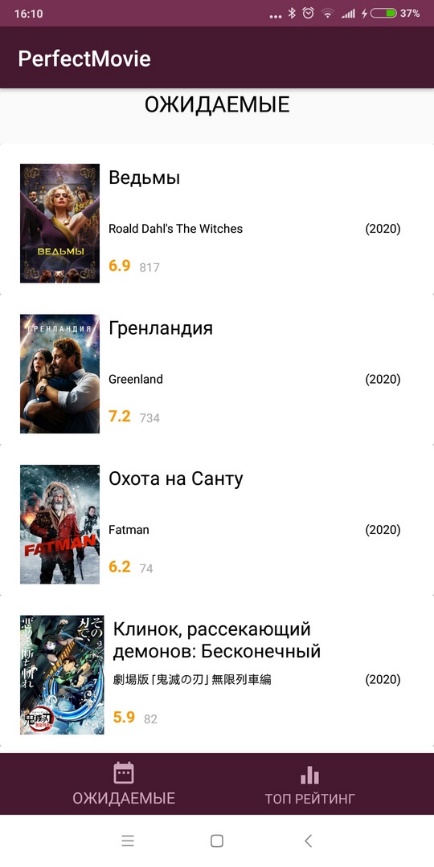


Рисунок 3 - Ожидаемые.

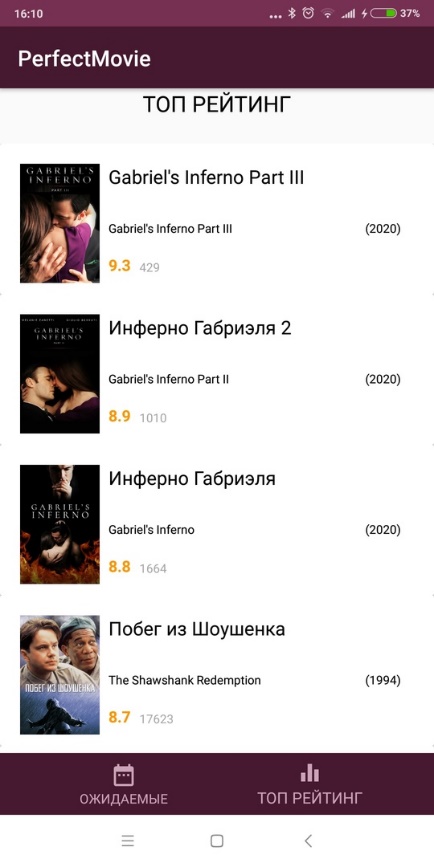


Рисунок 4 - Топ рейтинг.

## Инструментальные средства

Плюсы:

* гибкость среды разработки;
* больший набор функций;
* процесс разработки, который подстраивается под разработчика.
* редактор кода, с которым удобно работать;
* позволяет разрабатывать приложения не только для смартфонов/планшетов, а и для портативных ПК, приставок для телевизоров Android TV, устройств Android Wear, новомодных мобильных устройств с необычным соотношением сторон экрана;
* тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе;

Минусы:

* невозможность написать серверные проекты на языке Java для ПК, Android устройств.

## Характеристика программы

Таблица 3 – Характеристика программы

| Номер | Наименование модуля программы | Строчек кода | Размер, КБ |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | MainActivity.kt | 38 | 4 |
| 2 | TopFragment.kt | 94 | 5 |
| 3 | ReleasesFragment.kt | 92 |  |
| 4 | DetailActivity.kt | 42 |  |
| 5 | MovieAdapter.kt | 60 |  |
| 6 | Movies.kt | 17 |  |
| 7 | activity\_main.xml | 32 |  |
| 8 | activity\_detail.xml | 131 |  |
| 9 | films\_list.xml | 102 |  |
| 10 | fragment\_releases.xml | 37 |  |
| 11 | fragment\_top.xml | 40 |  |
| 12 | nav\_menu.xml | 14 |  |

# ВЫВОД

В ходе выполнения данной работы я усовершенствовала свои навыки работы с API и Picasso. Научился работать с различными элементами в Android studio. Углубил свои познания в языке программирования Kotlin.

# СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

<https://developer.android.com/docs>

<https://developers.themoviedb.org/3/getting-started/introduction>

ПРИЛОЖЕНИЕ А. Исходный текст программы

АННОТАЦИЯ

В приложении «Текст программы» продемонстрирован код приложения всех модулей.

Приложение состоит из нескольких разделов.

В разделе «Текст программы» представлен исходный текст каждого модуля системы.

Текст программы

MainActivity.kt

package com.example.perfectmovie  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import androidx.fragment.app.Fragment  
import com.example.perfectmovie.frames.ReleasesFragment  
import com.example.perfectmovie.frames.TopFragment  
import kotlinx.android.synthetic.main.activity\_main.\*  
  
class MainActivity : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
  
  
 val releasesFragment = ReleasesFragment()  
 val topFragment = TopFragment()  
  
 makeCurrentFragment(releasesFragment)  
 bottom\_navigation.setOnNavigationItemSelectedListener **{** when (**it**.*itemId*) {  
 R.id.*newReleases* -> makeCurrentFragment(releasesFragment)  
 R.id.*topFilms* -> makeCurrentFragment(topFragment)  
 }  
 true  
 **}** }  
  
  
 private fun makeCurrentFragment(fragment: Fragment) =  
 *supportFragmentManager*.beginTransaction().*apply* **{** replace(R.id.*frameSw*, fragment)  
 commit()  
 **}**}

TopFragment.kt

package com.example.perfectmovie.frames  
  
import android.content.Intent  
import android.os.Bundle  
import android.util.Log  
import androidx.fragment.app.Fragment  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ProgressBar  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.example.perfectmovie.\*  
  
import kotlinx.android.synthetic.main.fragment\_top.\*  
import okhttp3.OkHttpClient  
import okhttp3.Request  
import org.json.JSONObject  
import java.io.IOException  
  
*/\*\*  
 \* A simple [Fragment] subclass.  
 \*/*class TopFragment : Fragment() {  
  
 val URL = "https://api.themoviedb.org/3/movie/top\_rated?api\_key=${BuildConfig.API\_KEY1}&language=ru&page=1"  
  
 lateinit var progress\_Bar: ProgressBar  
 lateinit var recyclerview: RecyclerView  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
  
 ): View? {  
 // Inflate the layout for this fragment  
 var viewMain:View = inflater.inflate(R.layout.fragment\_top, container, false)  
 progress\_Bar = viewMain.findViewById<ProgressBar>(R.id.progressBar1)  
 progress\_Bar.visibility = View.VISIBLE  
 return viewMain  
  
 }  
  
 override fun onActivityCreated(savedInstanceState: Bundle?) {  
 super.onActivityCreated(savedInstanceState)  
  
  
 get()  
 }  
  
  
 fun get(){  
  
  
 val request: Request = Request.Builder().url(URL).build()  
 OkHttpClient().newCall(request).enqueue(object : okhttp3.Callback {  
 override fun onFailure(call: okhttp3.Call, e: IOException) {  
 Log.e("Error", e.toString())  
 }  
  
 var moviesList: ArrayList<Movies> = arrayListOf()  
 override fun onResponse(call: okhttp3.Call, response: okhttp3.Response) {  
 val json = (JSONObject(response.body()!!.string()))  
 activity?.runOnUiThread **{** val moviesArray = json.getJSONArray("results")  
 for (i in 0..moviesArray.length() - 1) {  
 var movie = moviesArray.optJSONObject(i)  
 var title = movie.getString("title")  
 var poster\_path = movie.getString("poster\_path")  
 var release\_date = movie.getString("release\_date")  
 var vote\_average = movie.getDouble("vote\_average")  
 var original\_title = movie.getString("original\_title")  
 var vote\_count = movie.getString("vote\_count")  
 var overview = movie.getString("overview")  
 var backdrop\_path = movie.getString("backdrop\_path")  
 moviesList.add(Movies(title, poster\_path,release\_date, vote\_average,original\_title,vote\_count,overview,backdrop\_path))  
 }  
 topReView.layoutManager = LinearLayoutManager(activity)  
 topReView.setHasFixedSize(true)  
 topReView.adapter = MovieAdapter(context,moviesList )**{** val intent = Intent(context, DetailActivity:: class.java)  
 intent.putExtra("OBJECT\_INTENT",it)  
 startActivity(intent)  
  
 **}** progress\_Bar.visibility = View.INVISIBLE  
 **}** }  
  
 })  
 }  
  
}

ReleasesFragment.kt

package com.example.perfectmovie.frames  
  
import android.content.Intent  
import android.os.Bundle  
import android.util.Log  
import androidx.fragment.app.Fragment  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ProgressBar  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.example.perfectmovie.\*  
  
import kotlinx.android.synthetic.main.fragment\_releases.\*  
import okhttp3.OkHttpClient  
import okhttp3.Request  
import org.json.JSONObject  
import java.io.IOException  
  
*/\*\*  
 \* A simple [Fragment] subclass.  
 \*/*class ReleasesFragment : Fragment() {  
  
 val URL = "https://api.themoviedb.org/3/movie/upcoming?api\_key=${BuildConfig.*API\_KEY1*}&language=ru&page=1"  
  
 lateinit var progress\_Bar: ProgressBar  
 lateinit var recyclerview: RecyclerView  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 // Inflate the layout for this fragment  
 val mainView:View = inflater.inflate(R.layout.*fragment\_releases*, container, false)  
 recyclerview = mainView.findViewById<RecyclerView>(R.id.*releasesReView*)  
 progress\_Bar = mainView.findViewById<ProgressBar>(R.id.*progressBar*)  
 progress\_Bar.*visibility* = View.*VISIBLE* return mainView  
 }  
  
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(view, savedInstanceState)  
 get()  
 }  
  
 fun get(){  
  
 val request: Request = Request.Builder().url(URL).build()  
 OkHttpClient().newCall(request).enqueue(object : okhttp3.Callback {  
 override fun onFailure(call: okhttp3.Call, e: IOException) {  
 Log.e("Error", e.toString())  
 }  
  
 var moviesList: ArrayList<Movies> = *arrayListOf*()  
 override fun onResponse(call: okhttp3.Call, response: okhttp3.Response) {  
 val json = (JSONObject(response.body()!!.string()))  
 *activity*?.runOnUiThread **{** val moviesArray = json.getJSONArray("results")  
 for (i in 0..moviesArray.length() - 1) {  
 var movie = moviesArray.optJSONObject(i)  
 var title = movie.getString("title")  
 var poster\_path = movie.getString("poster\_path")  
 var release\_date = movie.getString("release\_date")  
 var vote\_average = movie.getDouble("vote\_average")  
 var original\_title = movie.getString("original\_title")  
 var vote\_count = movie.getString("vote\_count")  
 var overview = movie.getString("overview")  
 var backdrop\_path = movie.getString("backdrop\_path")  
 moviesList.add(Movies(title, poster\_path,release\_date, vote\_average,original\_title,vote\_count,overview,backdrop\_path))  
  
  
  
 }  
 releasesReView.*layoutManager* = LinearLayoutManager(*activity*)  
 releasesReView.setHasFixedSize(true)  
 releasesReView.*adapter* = MovieAdapter(*context*,moviesList )**{** val intent = Intent(*context*, DetailActivity:: class.*java*)  
 intent.putExtra("OBJECT\_INTENT",**it**)  
 startActivity(intent)  
  
 **}** progress\_Bar.*visibility* = View.*INVISIBLE* **}** }  
  
 })  
 }  
  
}

DetailActivity.kt

package com.example.perfectmovie  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.ImageView  
import android.widget.TextView  
import com.squareup.picasso.Picasso  
  
class DetailActivity : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_detail*)  
  
  
 val item = *intent*.getParcelableExtra<Movies>("OBJECT\_INTENT")  
 val title = findViewById<TextView>(R.id.*tltFilm2*)  
 val original\_title = findViewById<TextView>(R.id.*orgTltFilm2*)  
 val release\_date = findViewById<TextView>(R.id.*dtFilm2*)  
 val vote\_average = findViewById<TextView>(R.id.*rtFilm2*)  
 val poster\_path = findViewById<ImageView>(R.id.*imgFilm2*)  
 val vote\_count = findViewById<TextView>(R.id.*votes2*)  
 val overview = findViewById<TextView>(R.id.*overview*)  
 val backdrop\_path = findViewById<ImageView>(R.id.*baner*)  
  
  
 title.*text* = item.title  
 release\_date.*text* = item.release\_date  
 original\_title.*text* = item.original\_title  
 vote\_average.*text* = item.vote\_average.toString()  
 vote\_count.*text* = item.vote\_count  
 overview.*text* = item.overview  
  
 Picasso.get().load("https://image.tmdb.org/t/p/w500/${item.backdrop\_path}").fit().centerInside()  
 .placeholder(R.drawable.*ic\_launcher\_background*)  
 .error(R.drawable.*ic\_launcher\_foreground*).into(backdrop\_path)  
 Picasso.get().load("https://image.tmdb.org/t/p/w500/${item.poster\_path}").fit().centerInside()  
 .placeholder(R.drawable.*ic\_launcher\_background*)  
 .error(R.drawable.*ic\_launcher\_foreground*).into(poster\_path)  
 }  
}

MovieAdapter.kt

package com.example.perfectmovie  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.view.animation.AnimationUtils  
import android.widget.ImageSwitcher  
import android.widget.ImageView  
import android.widget.TextView  
import androidx.cardview.widget.CardView  
import androidx.recyclerview.widget.RecyclerView  
import com.squareup.picasso.Picasso  
import kotlinx.android.synthetic.main.films\_list.view.\*  
  
class MovieAdapter(  
 private val context : Context?,  
 private val movies: List<Movies>,  
 val listener : (Movies) -> Unit  
): RecyclerView.Adapter<MovieAdapter.MovieViewHolder>() {  
 class MovieViewHolder (view:View) : RecyclerView.ViewHolder(view) {  
  
  
 val title = view.findViewById<TextView>(R.id.*tltFilm*)  
 val release\_date = view.findViewById<TextView>(R.id.*dtFilm*)  
 val vote\_average = view.findViewById<TextView>(R.id.*rtFilm*)  
 val poster\_path = view.findViewById<ImageView>(R.id.*imgFilm*)  
 val original\_title = view.findViewById<TextView>(R.id.*orgTltFilm*)  
 val vote\_count = view.findViewById<TextView>(R.id.*votes*)  
 val card\_view = view.findViewById<CardView>(R.id.*cardView*)  
  
 fun bindView(movies: Movies, listener: (Movies) -> Unit){  
 title.*text* = movies.title  
 release\_date.*text* = "(${movies.release\_date.*substring*(0,4)})"  
 vote\_average.*text* = movies.vote\_average.toString()  
 original\_title.*text* = movies.original\_title  
 vote\_count.*text* = movies.vote\_count  
  
 itemView.setOnClickListener **{** listener(movies) **}** }  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MovieViewHolder =  
 MovieViewHolder(LayoutInflater.from(context).inflate(R.layout.*films\_list*, parent, false))  
  
 override fun getItemCount(): Int = movies.size  
  
 override fun onBindViewHolder(holder: MovieViewHolder, position: Int) {  
  
 holder.card\_view.*animation* = AnimationUtils.loadAnimation(holder.card\_view.*context*,R.anim.*anim\_item*)  
  
 var currentItem = movies.get(position)  
 Picasso.get().load("https://image.tmdb.org/t/p/w500/${currentItem.poster\_path}").fit().centerInside()  
 .placeholder(R.drawable.*ic\_launcher\_background*)  
 .error(R.drawable.*ic\_launcher\_foreground*).fit().into(holder.poster\_path)  
  
 holder.bindView(movies[position], listener)  
 }  
}

Movies.kt

package com.example.perfectmovie  
  
import android.os.Parcelable  
import kotlinx.android.parcel.Parcelize  
  
@Parcelize  
data class Movies(  
 val title: String,  
 val poster\_path: String,  
 val release\_date : String,  
 val vote\_average: Double,  
 val original\_title : String,  
 val vote\_count : String,  
 val overview : String,  
 val backdrop\_path : String  
) : Parcelable

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".MainActivity"  
 >  
  
  
  
 <FrameLayout  
 android:id="@+id/frameSw"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_above="@id/bottom\_navigation" />  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottom\_navigation"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:background="#471930"  
 app:itemIconTint="#cca5b9"  
 app:itemTextColor="#cca5b9"  
 app:menu="@menu/nav\_menu" />  
  
  
  
</RelativeLayout>

activity\_detail.xml

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView 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=".DetailActivity">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
  
  
 <ImageView  
 android:id="@+id/imgFilm2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:scaleType="centerCrop"  
 tools:srcCompat="@tools:sample/avatars"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginTop="8dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="VectorDrawableCompat"  
 />  
 <TextView  
 android:id="@+id/tltFilm2"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/film\_name"  
 style="@style/title\_film"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="16dp"  
 app:layout\_constraintStart\_toEndOf="@id/imgFilm2"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:ems="12"  
 android:lines="2"  
 />  
  
 <TextView  
 android:id="@+id/orgTltFilm2"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="10dp"  
 android:text="@string/origin\_film\_name"  
 style="@style/org\_title\_film\_and\_date"  
 app:layout\_constraintStart\_toEndOf="@id/imgFilm2"  
 app:layout\_constraintTop\_toBottomOf="@id/tltFilm2"  
 app:layout\_constraintEnd\_toStartOf="@id/dtFilm2"  
  
 />  
 <TextView  
 android:id="@+id/dtFilm2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="20dp"  
 android:text="@string/date\_release"  
 style="@style/org\_title\_film\_and\_date"  
 app:layout\_constraintStart\_toEndOf="@id/orgTltFilm2"  
 app:layout\_constraintTop\_toBottomOf="@id/tltFilm2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 />  
 <TextView  
 android:id="@+id/rtFilm2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/orgTltFilm2"  
 app:layout\_constraintStart\_toEndOf="@id/imgFilm2"  
 android:layout\_marginBottom="7dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="20dp"  
 android:text="@string/rate\_film"  
 style="@style/rate"  
 />  
 <TextView  
 android:id="@+id/votes2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/orgTltFilm2"  
 app:layout\_constraintStart\_toEndOf="@id/rtFilm2"  
 android:layout\_marginBottom="7dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="24dp"  
 style="@style/vote\_count"  
 android:text="@string/votes"  
 />  
 <TextView  
 android:id="@+id/overview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/imgFilm2"  
 android:layout\_marginTop="10dp"  
 android:gravity="fill"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginStart="8dp"  
 android:padding="8sp"  
 android:text="@string/overview"  
 style="@style/overview"  
 android:lineSpacingMultiplier="1.25"  
 />  
  
 <ImageView  
 android:id="@+id/baner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginBottom="10dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/overview"  
 tools:ignore="VectorDrawableCompat"  
 tools:srcCompat="@tools:sample/avatars"  
 android:adjustViewBounds="true"  
 />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</ScrollView>

films\_list.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id = "@+id/cardView"  
 >  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 >  
  
 <ImageView  
 android:id="@+id/imgFilm"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:scaleType="centerCrop"  
 tools:srcCompat="@tools:sample/avatars"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="7dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="VectorDrawableCompat"  
 />  
 <TextView  
 android:id="@+id/tltFilm"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/film\_name"  
 style="@style/title\_film"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="16dp"  
 app:layout\_constraintStart\_toEndOf="@id/imgFilm"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:ems="12"  
 android:lines="2"  
 />  
  
 <TextView  
 android:id="@+id/orgTltFilm"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="10dp"  
 android:text="@string/origin\_film\_name"  
 style="@style/org\_title\_film\_and\_date"  
 app:layout\_constraintStart\_toEndOf="@id/imgFilm"  
 app:layout\_constraintTop\_toBottomOf="@id/tltFilm"  
 app:layout\_constraintEnd\_toStartOf="@id/dtFilm"  
  
 />  
 <TextView  
 android:id="@+id/dtFilm"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tltFilm"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="20dp"  
 android:text="@string/date\_release"  
 style="@style/org\_title\_film\_and\_date"  
 app:layout\_constraintStart\_toEndOf="@id/orgTltFilm"  
 app:layout\_constraintTop\_toBottomOf="@id/tltFilm"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 />  
 <TextView  
 android:id="@+id/rtFilm"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toEndOf="@id/imgFilm"  
 android:layout\_marginBottom="7dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="10dp"  
 android:text="@string/rate\_film"  
 style="@style/rate"/>  
  
 <TextView  
 android:id="@+id/votes"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toEndOf="@id/rtFilm"  
 android:layout\_marginBottom="7dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="10dp"  
 style="@style/vote\_count"  
 android:text="@string/votes"  
 />  
  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</androidx.cardview.widget.CardView>

fragment\_releases.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".frames.ReleasesFragment">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/fragment\_release\_title"  
 style="@style/title\_views"  
 />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/releasesReView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="50sp"  
  
 />  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:visibility="visible" />  
 </RelativeLayout>  
  
</FrameLayout>

fragment\_top.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".frames.TopFragment">  
  
 <RelativeLayout  
 android:id="@+id/relLay"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/fragment\_top\_title"  
 android:layout\_centerHorizontal="true"  
 style="@style/title\_views"  
 />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/topReView"  
 android:layout\_marginTop="50sp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
  
 />  
  
 <ProgressBar  
 android:id="@+id/progressBar1"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:visibility="visible" />  
 </RelativeLayout>  
  
</FrameLayout>

ПРИЛОЖЕНИЕ Б. Руководство пользователя

АННОТАЦИЯ

В данном документе приведено руководство пользователя веб-приложения для построения и вычисления графов.

Руководство пользователя показывает все возможности пользователя при тех или иных действиях.

В разделе «Использование» описан весь доступный пользователям функционал с подробным описанием его использования.

Использование

1. При нажатии на кнопку «Ожидаемые» происходит загрузка данных.

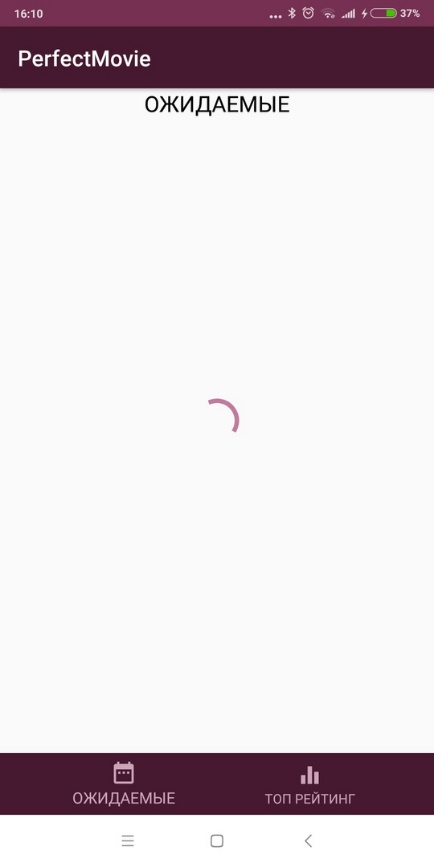


Рисунок 5 - Загрузка "Ожидаемые".

1. Появляются фильмы, которые ожидаются в ближайшее время.

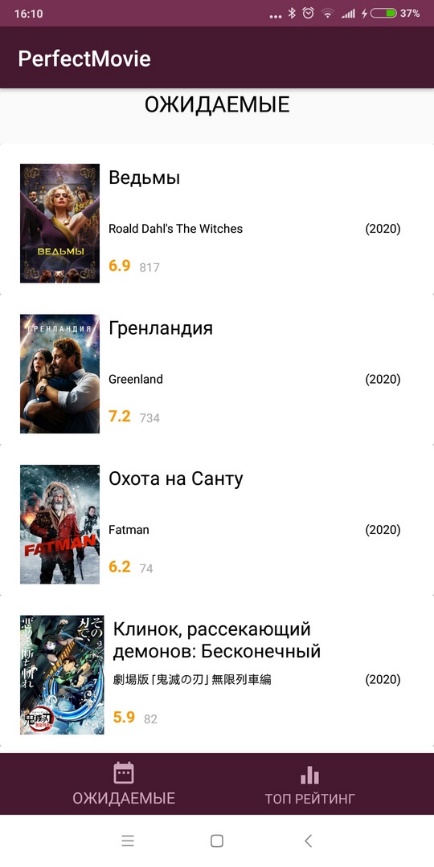


Рисунок 6 - Ожидаемые фильмы.

1. При нажатии на Один из фильмов высвечивается вся информация про фильм.

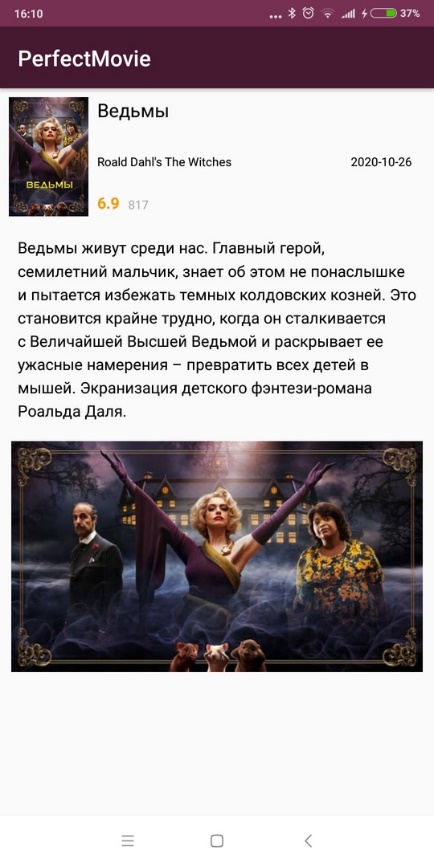


Рисунок 7 - Информация о фильме.

1. При нажатии на кнопку «Топ рейтинг» происходит загрузка данных.

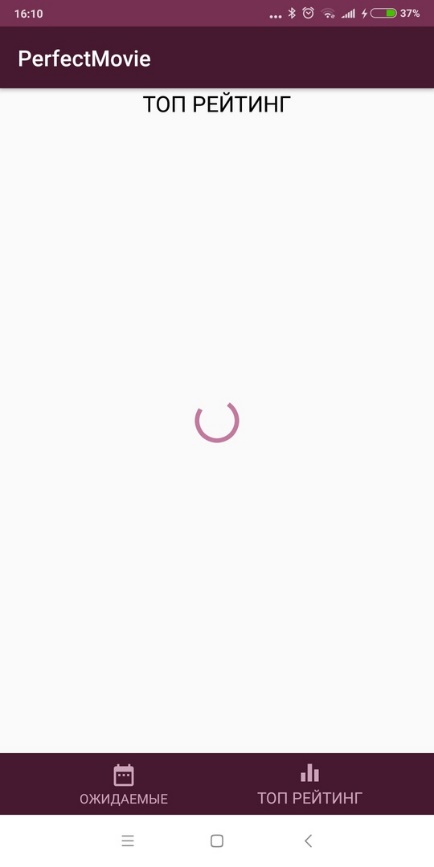


Рисунок 8 - Загрузка "Топ рейтинг".

1. Появляются фильмы, которые находятся в топ лучших.

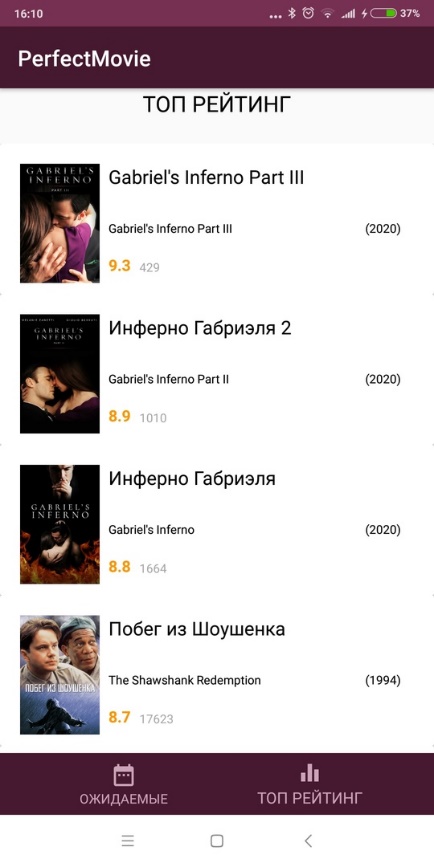


Рисунок 9 - Фильмы в топ рейтинге.

1. При нажатии на один из фильмов высвечивается вся информация о нем.

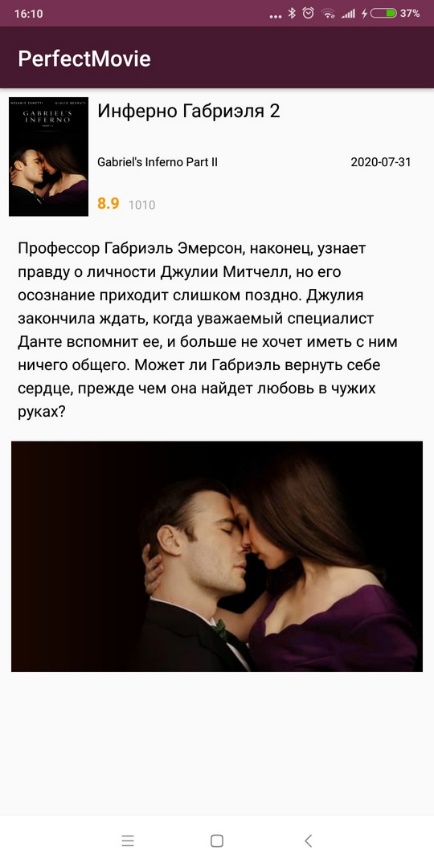


Рисунок 10 - Информация о фильме.