MOVIE DATABASE

A Mini Project Report Submitted by

SHASHANK (4NM17CS163)
SHRIKESH I (4NM17CS177)

UNDER THE GUIDANCE OF

MS. Ankitha A Nayak
Assistant Professor
Department of Computer Science and Engineering

in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering in
Computer Science & Engineering
from

Visvesvaraya Technological University, Belagavi

NITTE
EDUCATION TRUST

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

May 2020

NITTE EDUCATION TRUST | **N.M.A.M. INSTITUTE OF TECHNOLOGY**
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

"MOVIE DATABASE"

is a bonafide work carried out by

Shashank (4NM17CS163)
Shrikesh I (4NM17CS177)

in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2019-2020.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, we express our deep sense of gratitude and indebtedness to our guide **Ms. Ankitha A Nayak**, Assistant Professor, Department of Computer Science and Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We sincerely thank **Dr. K.R. Udaya Kumar Reddy**, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

Our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We also thank all those who have supported us throughout the entire duration of our project.

Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

Shashank (4NM17CS163)
Shrikesh I (4NM17CS177)

ABSTRACT

There are lot of movies available to watch and nowadays people prefer to watch movies at home rather than going to theatres. So how are they going to figure out what’s good and what’s not-so-good? What if the Netflix recommendations aren’t working for them? The purpose of this app is to point the user in the direction of something worth checking out. This application allows users to discover the most popular movies, Top rated and Upcoming movies which are now running in theatres. After reading the synopsis and reviews, people get the flexibility to decide if they want to watch the movies or not. The project has been developed using different technologies Java, Android and SQLite and is integrated together with android libraries such as Picasso and Retrofit.

Movie Database

Table Of Contents:

CHAPTER 1

INTRODUCTION

1.1 Scope

It is a flexible, easy to use, compact designed to benefit the user. It is used to find the latest movies and new movies and fresh content when you open using RESET interface.

1.2 Importance

It is difficult to find which movies to watch and the Netflix recommendation may not be good for you .Instead of surfing the internet for the ratings and review we have all that in one place using the information from TMDB website.

1.3 Objective:

The objective of this project is a mobile application developed for viewing the most popular and top rated movies available and the rating and date of airing and make it easy for user to make their watchlist.

CHAPTER 2

LITERATURE SURVEY

2.1 Technical Background:

The developed application is based on the data available in TMdb [The Movie DB API], but can also incorporate new data sources. This application uses REST interface to pull fresh content each time it opens from the database

2.2 Proposed system

The proposed solution is ‘Moviedatabase’, an android mobile application using REST Interface [2], which provides movie lovers with the most updated movie information and this helps them save time and money. They can read the movie information and then decide if they want to watch movies at theatre or at home. This application helps users to find the most popular movies, top rated movies and upcoming movies which will be running in theatres. It also provides the list of movies based on different genres. Users will be able to read the ratings, movie synopsis, and release information about the latest movie. Moviedatabase is developed using different technologies Java, Android and SQLite and it is integrated together with android libraries such as Picasso and Retrofit. Movie box application is based on the data available in TMdb [The Movie DB API], but can also incorporate new data sources. My application uses REST interface to pull fresh content each time it opens from the database. In particular, my design has the following feature. I have organized the movie title in such a way that it helps people to go over the list of grid and discover the most popular movies. My idea is to keep it simple and provide a user friendly experience to the users. Overall, Moviedatabase is an attempt to demonstrate the integration of several latest technologies to create a useful real time app.

CHAPTER 2

SYSTEM REQUIREMENT AND SPECIFICATION

3.1 Introduction

Requirements are during early stages of a system development as a specification of what should be implemented or as a constraint of some kind of on the system. They may be a user level facility description, a detailed specification of expected system behavior, a general system property, a specific constraint on the system, and information on how to carry out some computation or a constraint on the development of the system. The end product of the requirement analysis phase is a requirement specification. The requirement specification is a reconstruction of the result of this analysis phase. Its purpose is to communicate this result to others. System requirements are more detailed descriptions of the user requirements. They may serve as the basis for a contract to the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point of system design. In principle, the system requirements should state what the system should do and not how it should be implemented. However, at the level of detail required to specify the system completely, it is virtually impossible to exclude all design information.

Natural language is often used to write system requirements specifications. Further problems with natural language can arise when it is used for more detailed specification:

1. Natural language understanding relies on the specification of the readers and writers using the same words for the same concept. This leads to misunderstandings because of the ambiguity of the natural language.
2. A natural language requirements specification is over-flexible. You can say the same thing in completely different ways. It is up to the reader to find out when requirements are same and when they are distinct.

3. There is no easy way to modularize natural language requirements. It may be difficult to find all the related requirements. To discover the consequence of a change, you may have to look at every requirement rather than just a group of related requirements.

3.2 Functional Requirements

The functional requirements are the statement of services the system should provide, how system reacts to particular inputs and how system should behave in particular situation. It describes the functionality that the system provides.

Our app requires:

I) Active internet connection.

3.3 User Requirements

Both student and lecturer requires active internet connection to use the app.

3.4 Software Requirements

I. Operating System: Windows 7/8/10 (32-bit or 64-bit)

II. Android SDK

III. III. Android Studio

IV. IV. Firebase account.

3.4.1 Android SDK

I. The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. The ADT bundle includes the essential Android SDK components and a version of the Eclipse IDE with built-in Android Developer Tools to streamline the Android app development. ADT bundle consists of following components for developing the application

II. Eclipse ADT plugin.

III. Android SDK Tools

IV. Android Platform-tools

V. The latest Android platform

VI. The latest Android system image for the emulator

3.4.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed

specifically, for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.3, which was released in January 2019.

3.4.3 Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps. Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage.

3.5 HARDWARE REQUIREMENTS

I. Intel i Series Processor.

II. Minimum 4 GB RAM (8GB recommended).

III. 5GB free disk space

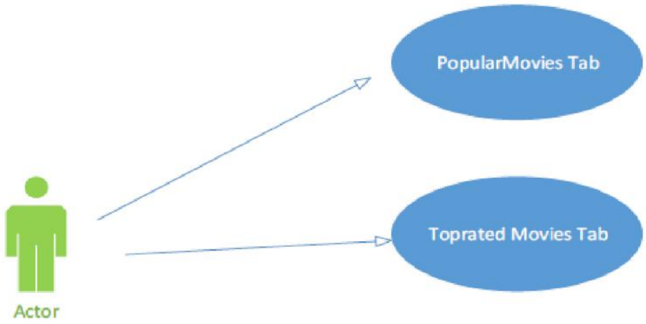
IV. USB 2.0 or higher.

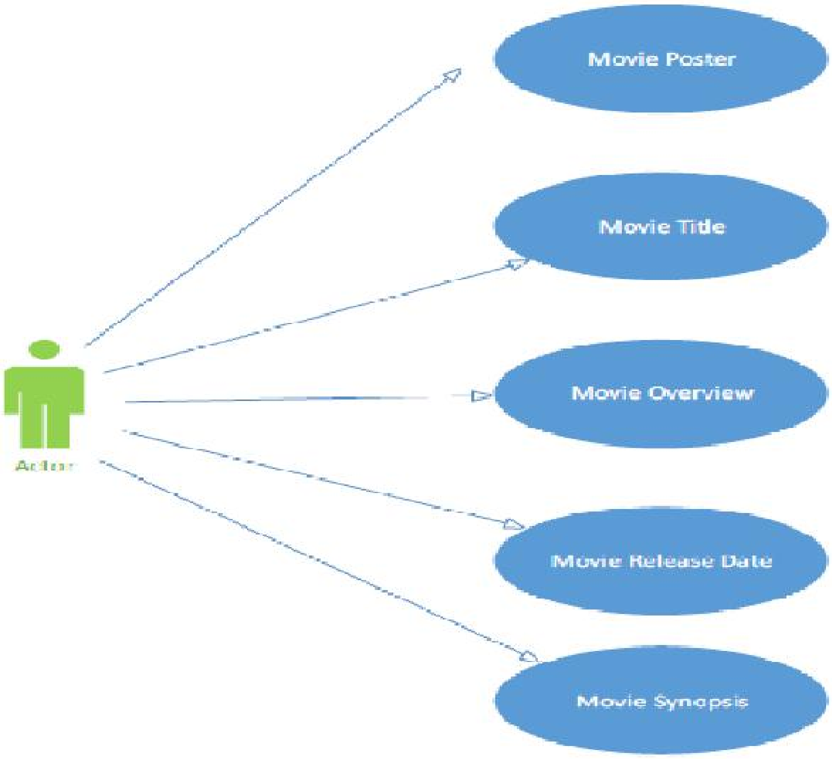
V. Android Device

CHAPTER 4

SYSTEM DESIGN

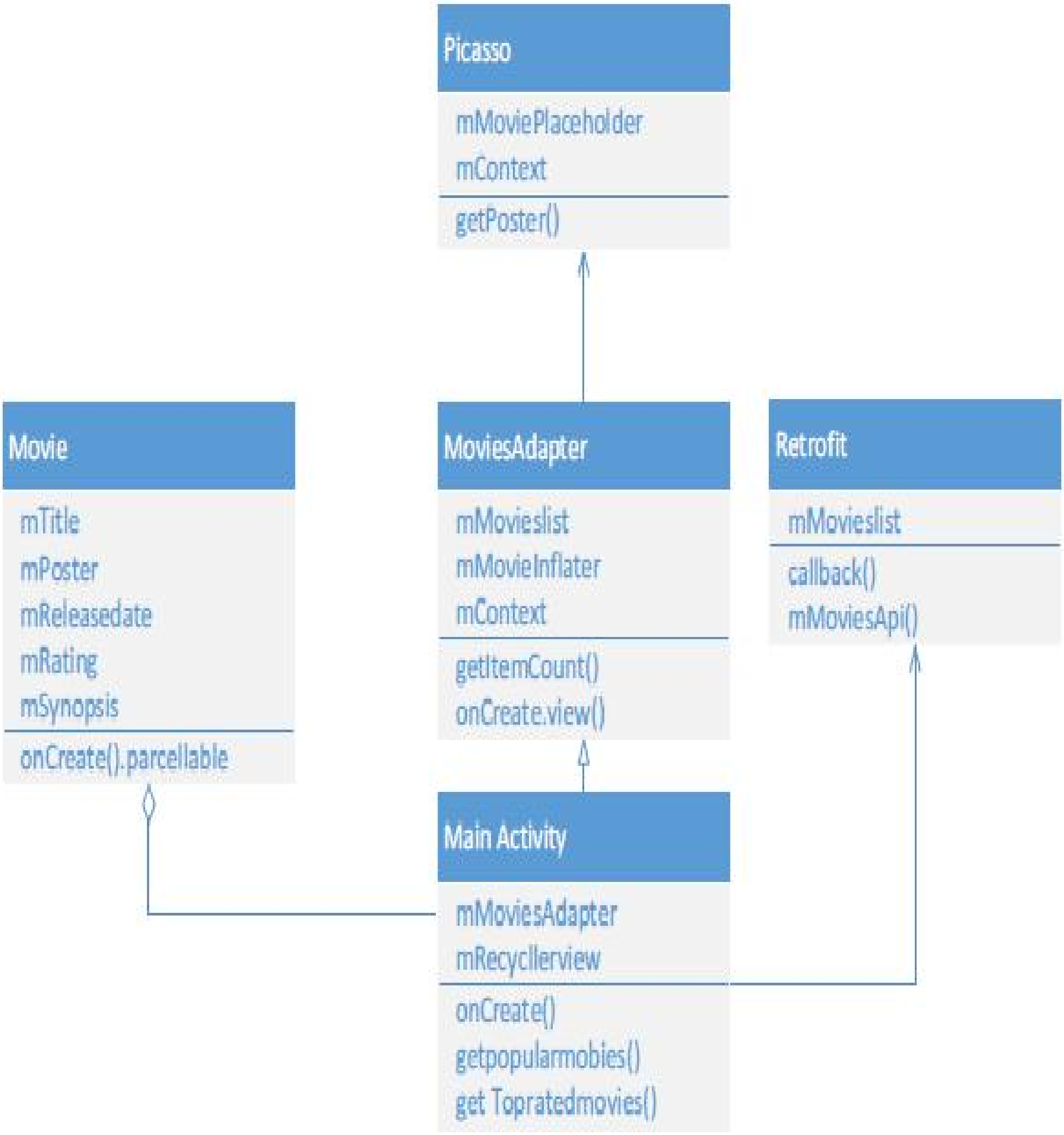
4.1 Use case model for Syncup





Use Case Diagram for Movie Detail Activity

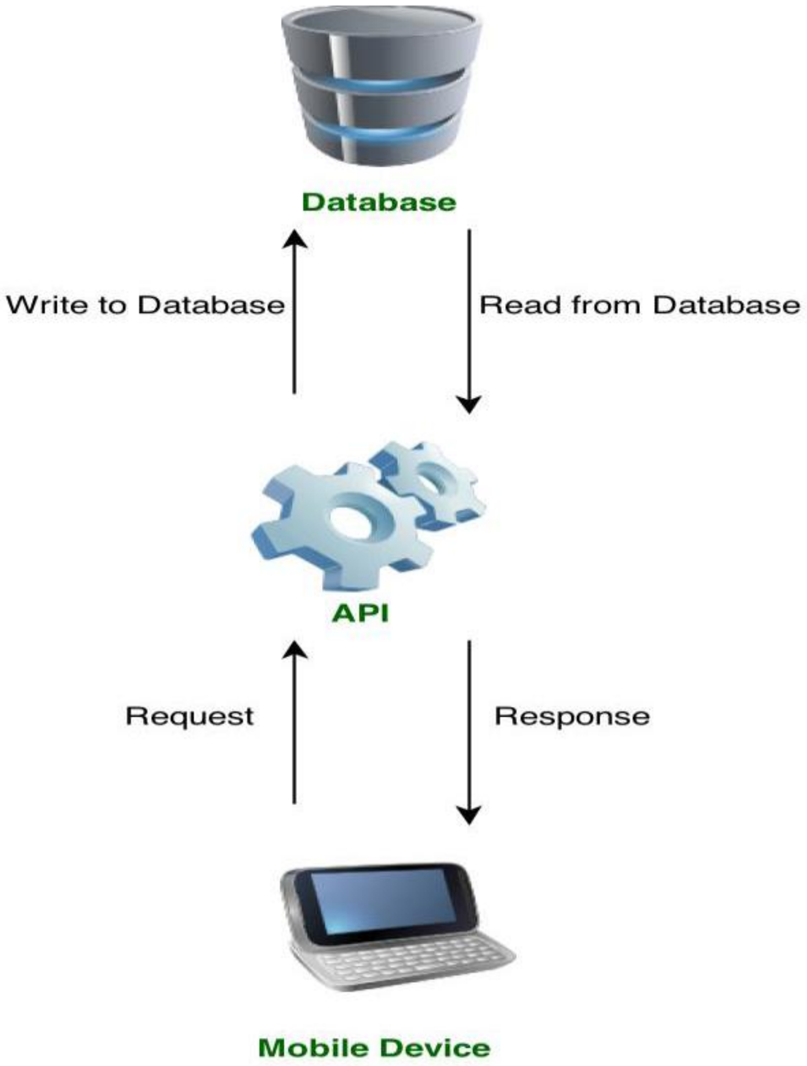
4.2 Dataflow Diagram



1. Lecturer logs into google firebase account, uploads circulars and can insert, update and delete the notices.
2. The students at the front end will be able to view circulars and timetables
3. Each user is given with a unique username and password.

4.4 Database Design

In this project, I have used SQLite database which is an open source database to sort the movies based on different order. Moviebox connects to the TMdb database using the REST API [24]. It uses HTTP request to GET the data from the database using the Web services [24]. The movie information is fetched using the JSON formats. In this project, when we click on a movie poster the information is sent via web services in JSON format and displayed to users from the Tmdb. The Figure 10 below illustrates this process .


Database
Write to Database
Read from Database
API
Request
Response
Mobile Device

CHAPTER 5

IMPLEMENTATION

User Interface Design for any mobile application should strive to be sufficiently easy such that a new user can use the app. The app should be user friendly and easily accessible. A good design approach is to have a three click rule where the user should be able to get all the information in just three clicks. This application has two main screens:

- Home screen
- Movie Detail screen

Home screen is where the movie posters are displayed across three different tabs, and the user should be able to view the movies based on Popular category, Top-rated Category and Upcoming Category. Movie Detail screen is where the users can select the movie posters and use the app to view the movie information.

Following Figure 9 is the main (home screen). As seen from the image, this screen has all the key features.

The next important feature implemented in the MainActivity is Recycler view [25]. The home screen displays scrollable movie posters. This feature can be implemented by using the object android.support.v7.widget.RecylerView.Recycler. This is responsible for managing item views for reuse. Here I have created a grid view of movie poster images by creating a row for these items, and used that layout which I created in the adapter. The Recycler view only creates required number of views to display the data needed at a time. For example, for a 2x2 grid which scrolls vertically, the Recyclerview creates 8 views. It displays 4 views at a time two above and two below are ready to be shown when scrolled. As the user scrolls, already created views are displayed by the Recyclerview which binds new data to them.

To make this job easier, Android developer site provides a smooth scrolling option with the view Holder. The Recycler view creates eight views to display the number of rows and columns and by doing so avoids creating new views whenever user scrolls..

By using this feature, I have created a class which references all the views which are needed to populate the layout. This reference is reused frequently by calling method findViewById(). This advantage greatly improves the performance of the app. The recycler 33 view uses the Adapter as the data source and I have used View Holders to keep reference in memory.

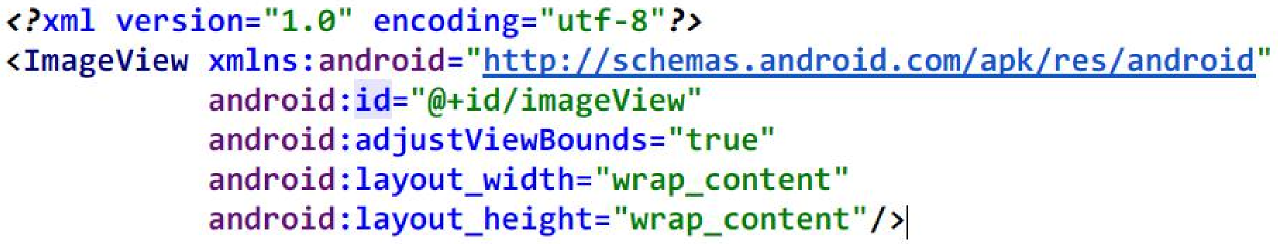
4.4.1 Using Picasso to load image into Image View

Picasso is an image library which is used for image loading and image processing [20]. The library handles every step from loading initial http request to caching of images. The below import statement should be added to the Main Activity class in order to use Picasso.

import com.squareup.picasso.Picasso;

For using Picasso, permissions should be added to the manifest file:

Then I added imageView to the layout file

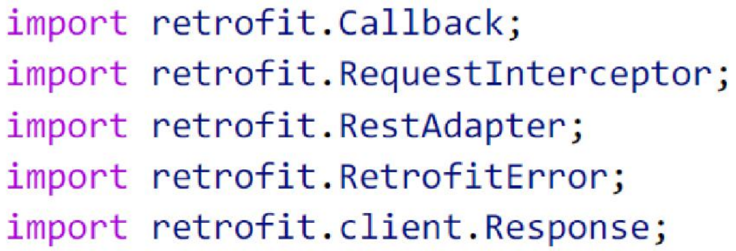


```
<?xml version="1.0" encoding="utf-8"?>
<ImageView xmlns:android="http://schemas.android.com/apk/res/android"
             android:id="@+id/imageView"
             android:adjustViewBounds="true"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"/>
```

I have used the Retrofit library, which is REST client for Android which is developed by Square [21]. It provides an easy way to retrieve the data and upload JSON file using a REST based service. I have implemented retrofit and used it to display movie pages from the TMdb API by following the steps below.

1. Downloaded Retrofit and installed or imported Retrofit by adding the below code to the build.gradle app file
Compile ‘com.squareup.retrofit:retrofit:1.9.0’

2. Imported the library in the MainActivity.java

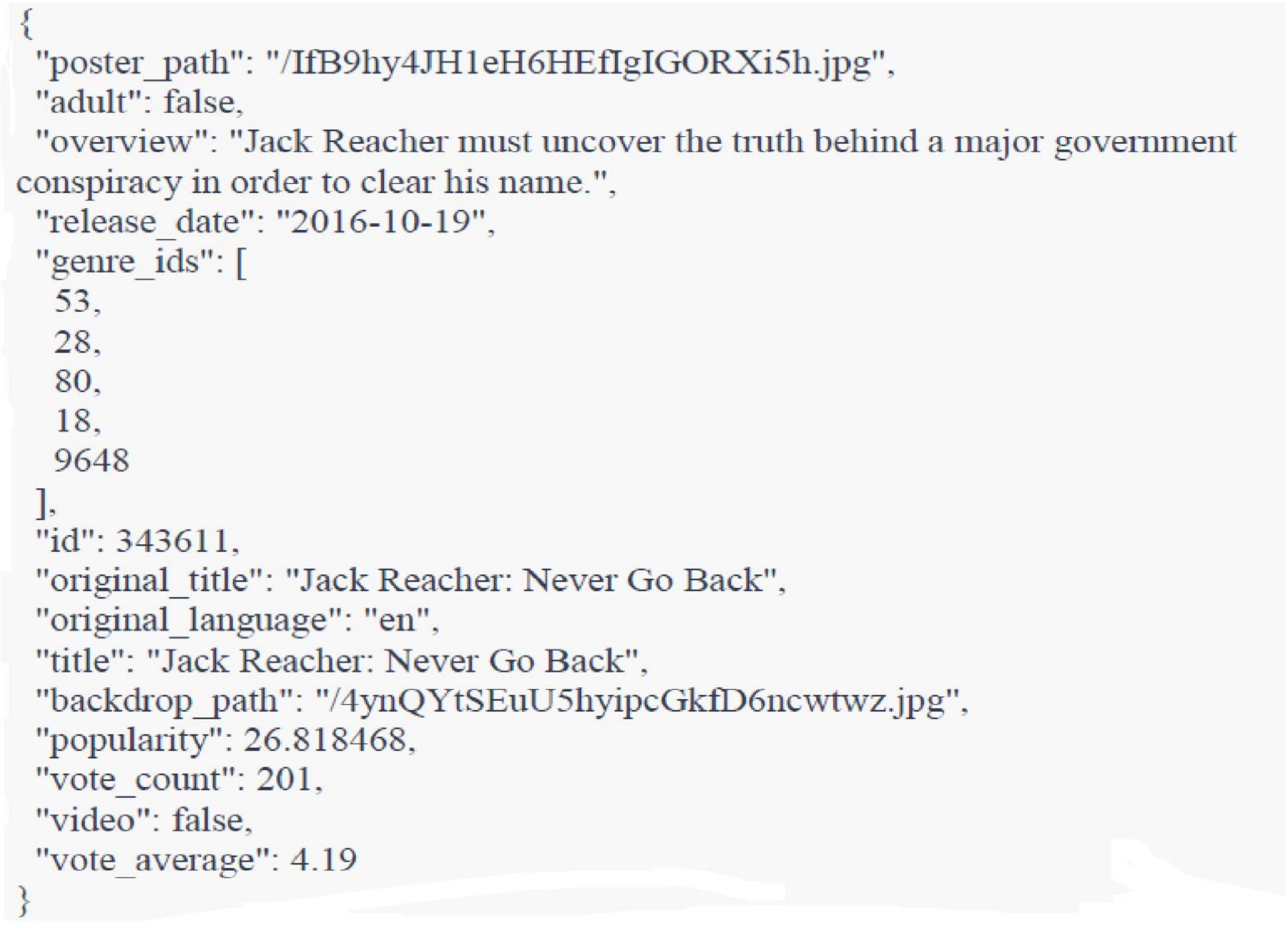


```
import retrofit.Callback;
import retrofit.RequestInterceptor;
import retrofit.RestAdapter;
import retrofit.RetrofitError;
import retrofit.client.Response;
```

3. Then created an interface which will represent the API. Here, I had set up an interface to the API, which expose calls on an API endpoint like
https://api.themoviedb.org/3/movie/popular?api_key=4e759e0ec75edf2f80e6d752
6aee7735&language=en-US&page=1

After setup is done, the API needs to be called, before which we need to initialize retrofit. First a rest adapter is created with all the configuration detail it needs in order to use the interface. The rest adapter sets the endpoint as the first part of the url for the API. A request interceptor gets called for every request made instead of adding API key each time. The log level informs the retrofit about the request and responses that it should display.
Now REST adapter is complete and we use the getPopularMovies() method using callback. Callback is a method used to pull data from web services and is executed by a different object. Success method will be called on success of the callback. I will have access to the movie list from the server using that method. I can then use that movie list and pass it to the adapter.

Below is a sample JSON array which contains the movie list objects of a single item



```
{
  "poster_path": "/IfB9hy4JH1eH6HEfIgIGORXi5h.jpg",
  "adult": false,
  "overview": "Jack Reacher must uncover the truth behind a major government
conspiracy in order to clear his name.",
  "release_date": "2016-10-19",
  "genre_ids": [
    53,
    28,
    80,
    18,
    9648
  ],
  "id": 343611,
  "original_title": "Jack Reacher: Never Go Back",
  "original_language": "en",
  "title": "Jack Reacher: Never Go Back",
  "backdrop_path": "/4ynQYtSEuU5hyipcGkfD6ncwtwz.jpg",
  "popularity": 26.818468,
  "vote_count": 201,
  "video": false,
  "vote_average": 4.19
}
```

4.5 MovieDetailInfo Activity

When the user clicks on the Movie image poster in Home screen, he should be navigated to the next screen where he should be able to view the movie detail below 49

- Movie Poster with Title
- Overview
- Release date
- Rating
- Synopsis

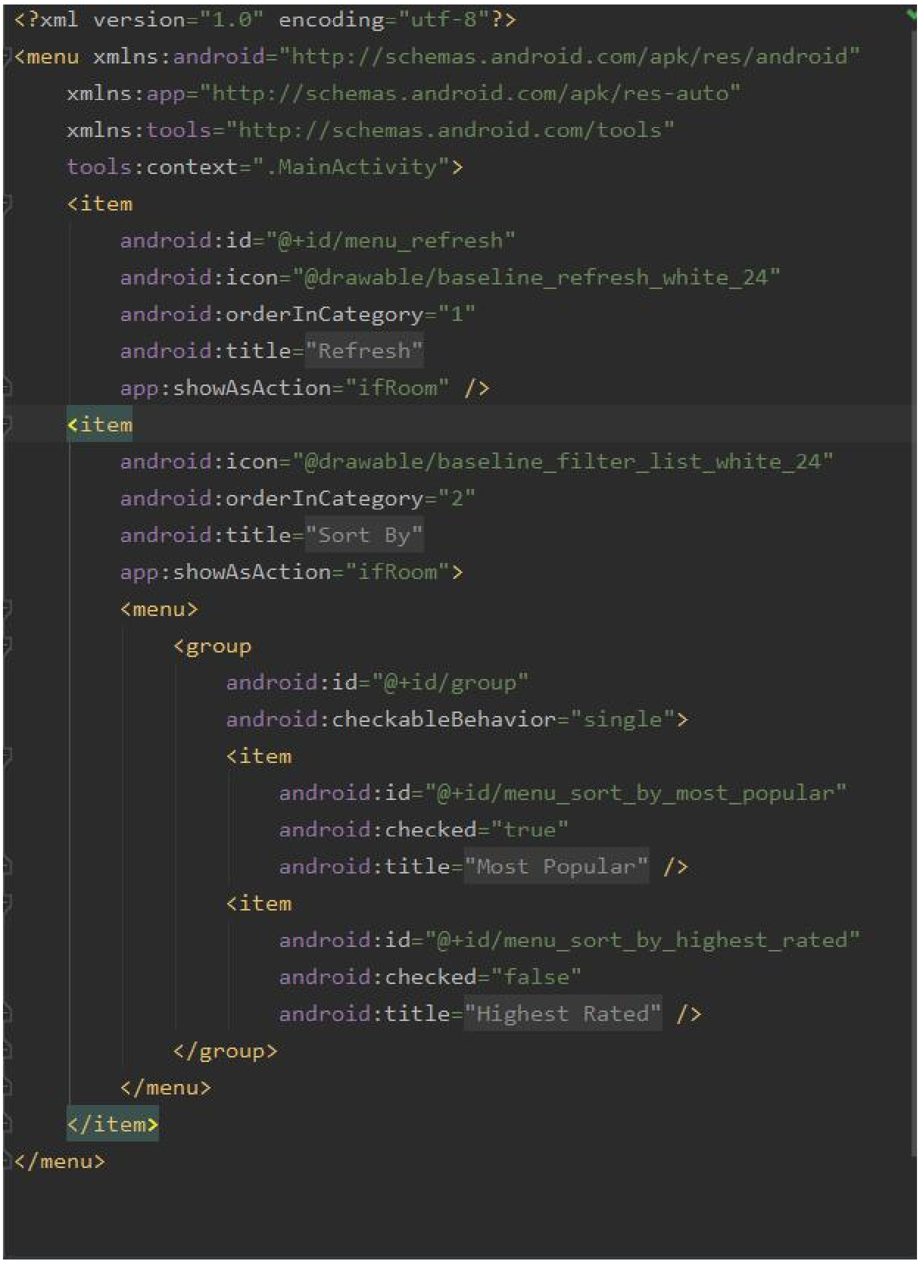
5.1.4 Java

There are several ways to create apps for Android devices, but the recommended method for most developers is to write native apps using Java and the Android SDK. Java for Android apps is both similar and quite different from other types of Java applications.

If you have experience with Java (or a similar language) then you'll probably feel comfortable diving right into the code and learning how to use the Android SDK to make your app run. But if you're new to programming or object-oriented languages then you'll probably want to get familiar with the syntax of the Java language and how to accomplish basic programming tasks before learning how to use the Android SDK.

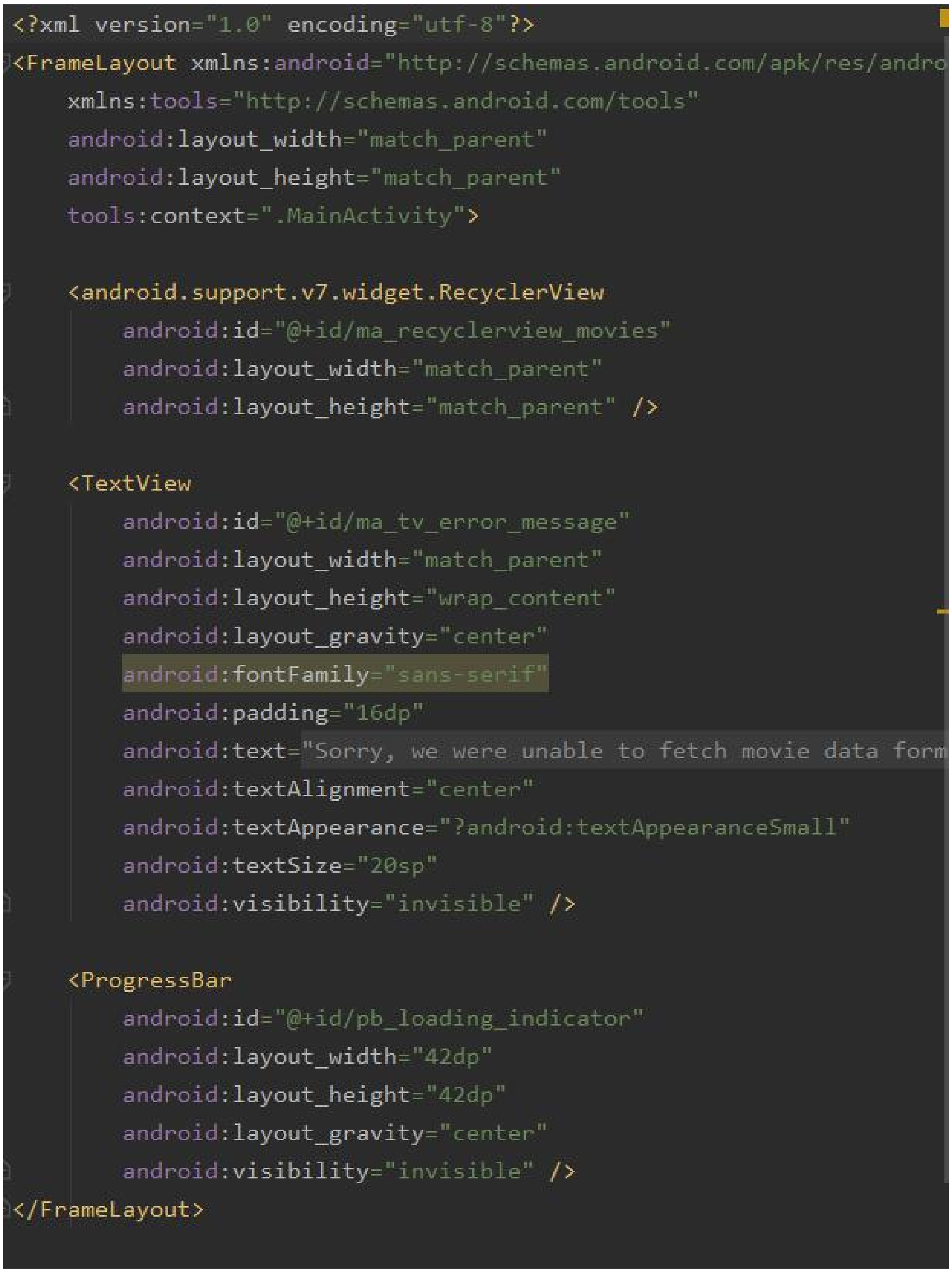
5.2 IMPLEMENTATION CODE

Main_menu.xml



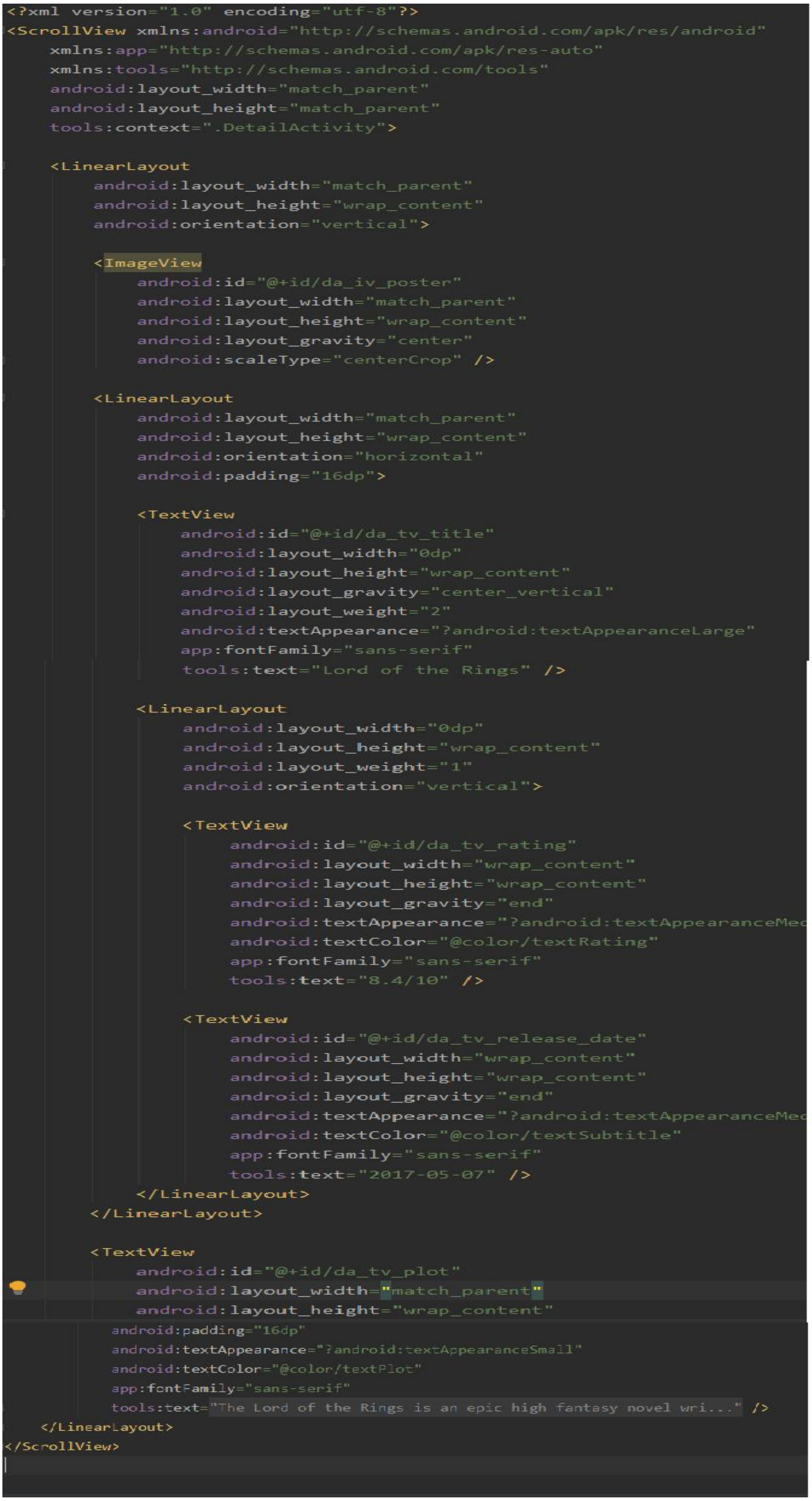
```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".MainActivity">
    <item
        android:id="@+id/menu_refresh"
        android:icon="@drawable/baseline_refresh_white_24"
        android:orderInCategory="1"
        android:title="Refresh"
        app:showAsAction="ifRoom" />
    <item
        android:icon="@drawable/baseline_filter_list_white_24"
        android:orderInCategory="2"
        android:title="Sort By"
        app:showAsAction="ifRoom">
        <menu>
            <group
                android:id="@+id/group"
                android:checkableBehavior="single">
                <item
                    android:id="@+id/menu_sort_by_most_popular"
                    android:checked="true"
                    android:title="Most Popular" />
                <item
                    android:id="@+id/menu_sort_by_highest_rated"
                    android:checked="false"
                    android:title="Highest Rated" />
            </group>
        </menu>
    </item>
</menu>
```

Activity_main.xml



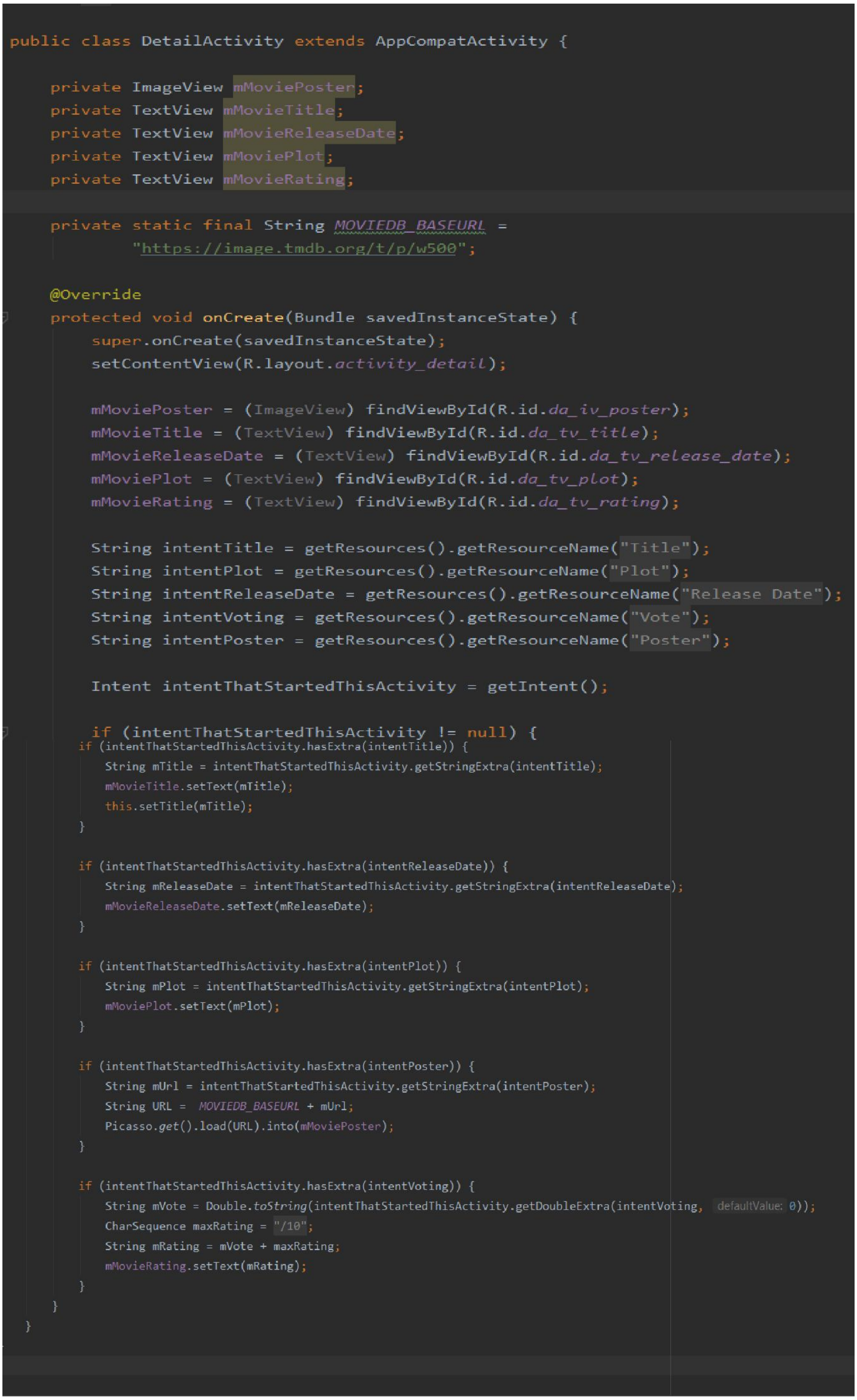
```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/andro
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/ma_recyclerview_movies"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <TextView
        android:id="@+id/ma_tv_error_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="sans-serif"
        android:padding="16dp"
        android:text="Sorry, we were unable to fetch movie data form
        android:textAlignment="center"
        android:textAppearance="?android:textAppearanceSmall"
        android:textSize="20sp"
        android:visibility="invisible" />

    <ProgressBar
        android:id="@+id/pb_loading_indicator"
        android:layout_width="42dp"
        android:layout_height="42dp"
        android:layout_gravity="center"
        android:visibility="invisible" />
</FrameLayout>
```

Activity_detail.xml



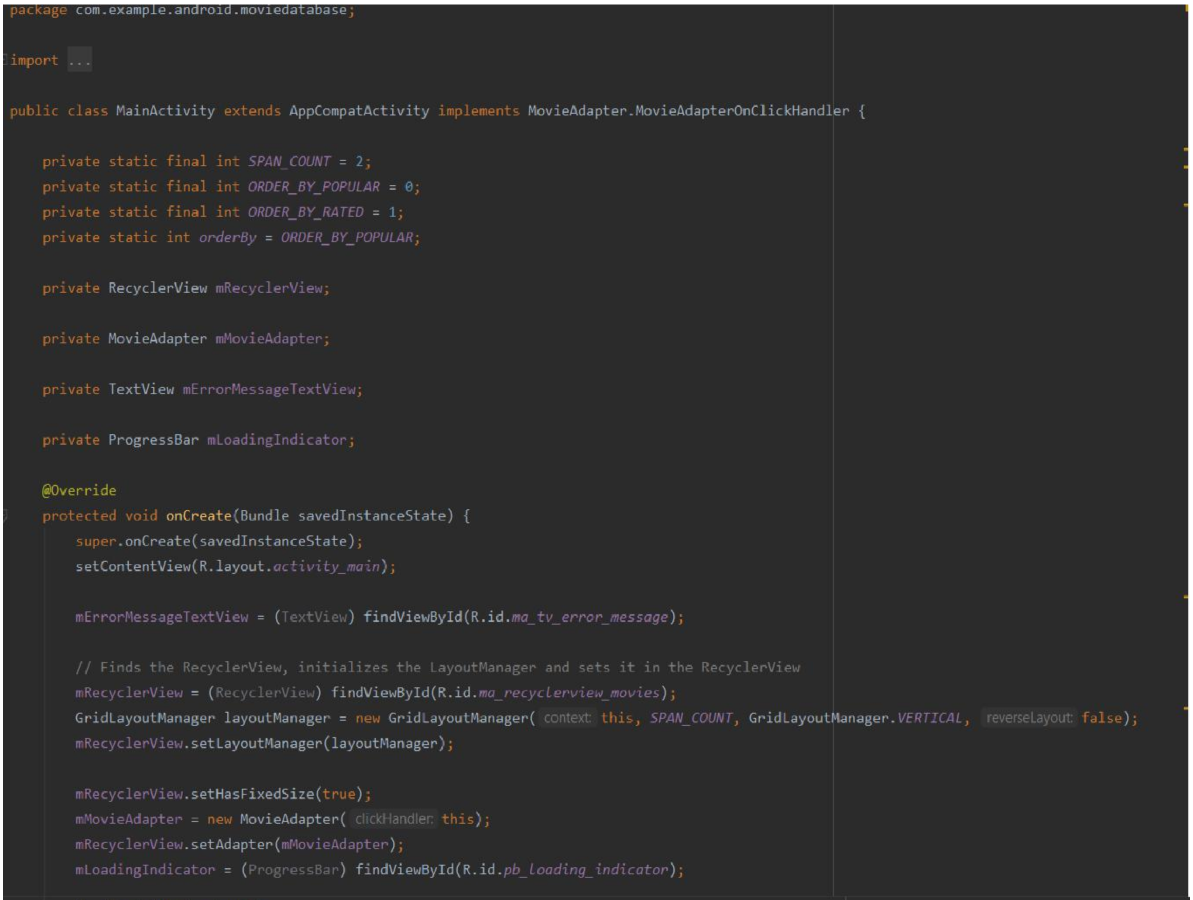
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DetailActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/da_iv_poster"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:scaleType="centerCrop" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:padding="16dp">

            <TextView
                android:id="@+id/da_tv_title"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:layout_weight="2"
                android:textAppearance="?android:textAppearanceLarge"
                app:fontFamily="sans-serif"
                tools:text="Lord of the Rings" />

            <LinearLayout
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/da_tv_rating"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_gravity="end"
                    android:textAppearance="?android:textAppearanceMed
                    android:textColor="@color/textRating"
                    app:fontFamily="sans-serif"
                    tools:text="8.4/10" />

                <TextView
                    android:id="@+id/da_tv_release_date"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_gravity="end"
                    android:textAppearance="?android:textAppearanceMed
                    android:textColor="@color/textSubtitle"
                    app:fontFamily="sans-serif"
                    tools:text="2017-05-07" />
            </LinearLayout>
        </LinearLayout>

        <TextView
            android:id="@+id/da_tv_plot"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp"
            android:textAppearance="?android:textAppearanceSmall"
            android:textColor="@color/textPlot"
            app:fontFamily="sans-serif"
            tools:text="The Lord of the Rings is an epic high fantasy novel wri..." />
    </LinearLayout>
</ScrollView>
```

DetailActivity.java

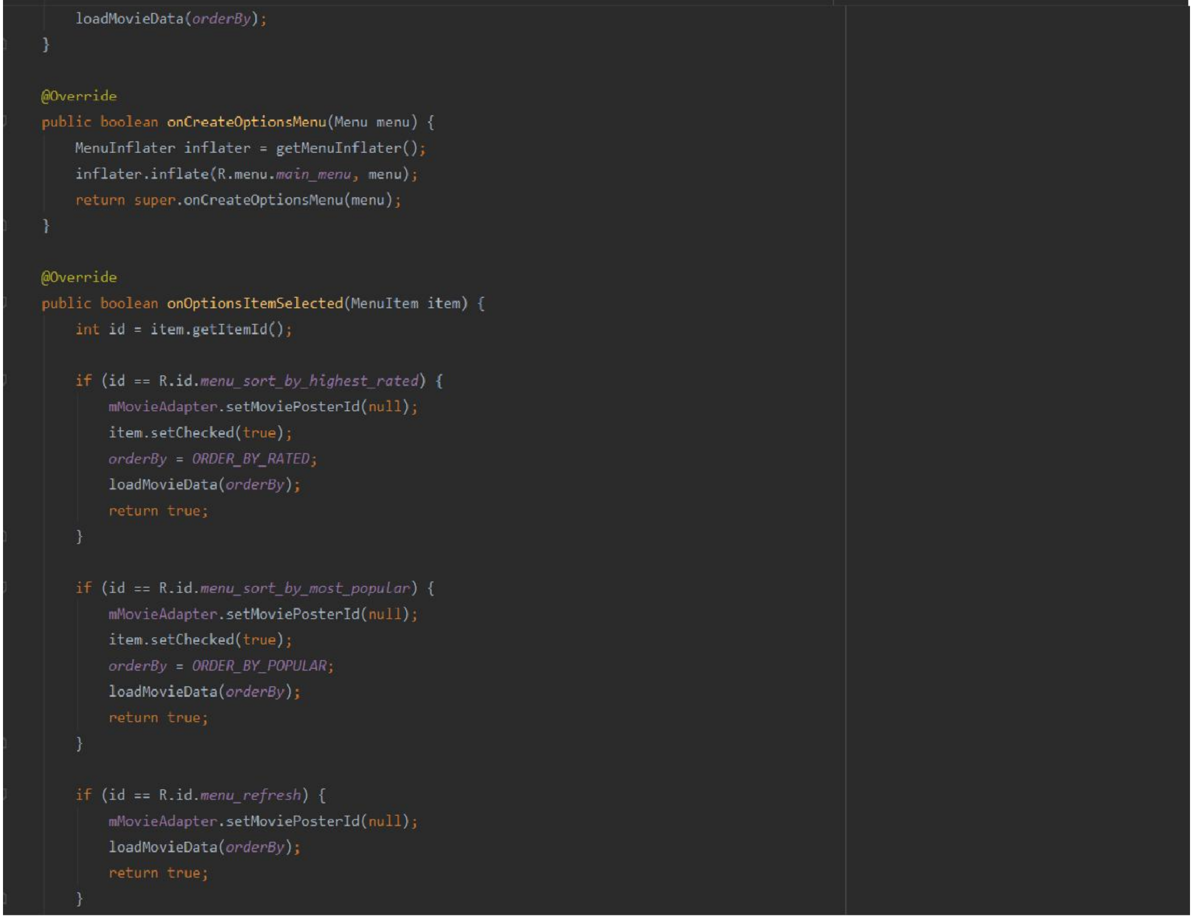


```
public class DetailActivity extends AppCompatActivity {

    private ImageView mMoviePoster;
    private TextView mMovieTitle;
    private TextView mMovieReleaseDate;
    private TextView mMoviePlot;
    private TextView mMovieRating;

    private static final String MOVIEDB_BASEURL =
            "https://image.tmdb.org/t/p/w500";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        mMoviePoster = (ImageView) findViewById(R.id.da_iv_poster);
        mMovieTitle = (TextView) findViewById(R.id.da_tv_title);
        mMovieReleaseDate = (TextView) findViewById(R.id.da_tv_release_date);
        mMoviePlot = (TextView) findViewById(R.id.da_tv_plot);
        mMovieRating = (TextView) findViewById(R.id.da_tv_rating);

        String intentTitle = getResources().getResourceName("Title");
        String intentPlot = getResources().getResourceName("Plot");
        String intentReleaseDate = getResources().getResourceName("Release Date");
        String intentVoting = getResources().getResourceName("Vote");
        String intentPoster = getResources().getResourceName("Poster");

        Intent intentThatStartedThisActivity = getIntent();

        if (intentThatStartedThisActivity != null) {
        if (intentThatStartedThisActivity.hasExtra(intentTitle)) {
            String mTitle = intentThatStartedThisActivity.getStringExtra(intentTitle);
            mMovieTitle.setText(mTitle);
            this.setTitle(mTitle);
        }

        if (intentThatStartedThisActivity.hasExtra(intentReleaseDate)) {
            String mReleaseDate = intentThatStartedThisActivity.getStringExtra(intentReleaseDate);
            mMovieReleaseDate.setText(mReleaseDate);
        }

        if (intentThatStartedThisActivity.hasExtra(intentPlot)) {
            String mPlot = intentThatStartedThisActivity.getStringExtra(intentPlot);
            mMoviePlot.setText(mPlot);
        }

        if (intentThatStartedThisActivity.hasExtra(intentPoster)) {
            String mUrl = intentThatStartedThisActivity.getStringExtra(intentPoster);
            String URL =  MOVIEDB_BASEURL + mUrl;
            Picasso.get().load(URL).into(mMoviePoster);
        }

        if (intentThatStartedThisActivity.hasExtra(intentVoting)) {
            String mVote = Double.toString(intentThatStartedThisActivity.getDoubleExtra(intentVoting,  defaultValue: 0));
            CharSequence maxRating = "/10";
            String mRating = mVote + maxRating;
            mMovieRating.setText(mRating);
        }
    }
}
```

MainActivity.java



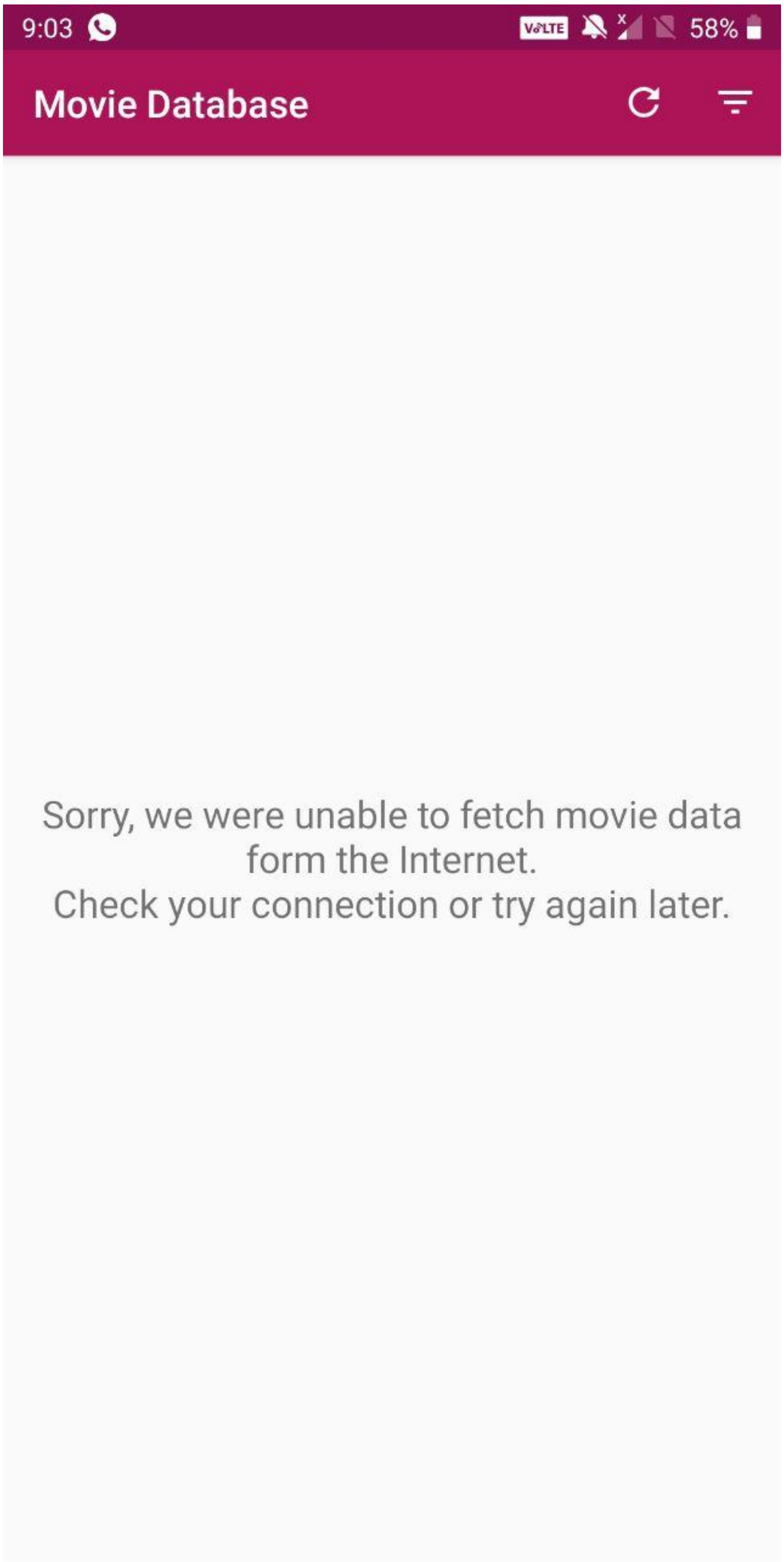
```
package com.example.android.moviedatabase;

import ...

public class MainActivity extends AppCompatActivity implements MovieAdapter.MovieAdapterOnClickHandler {

    private static final int SPAN_COUNT = 2;
    private static final int ORDER_BY_POPULAR = 0;
    private static final int ORDER_BY_RATED = 1;
    private static int orderBy = ORDER_BY_POPULAR;

    private RecyclerView mRecyclerView;

    private MovieAdapter mMovieAdapter;

    private TextView mErrorMessageTextView;

    private ProgressBar mLoadingIndicator;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mErrorMessageTextView = (TextView) findViewById(R.id.ma_tv_error_message);

        // Finds the RecyclerView, initializes the LayoutManager and sets it in the RecyclerView
        mRecyclerView = (RecyclerView) findViewById(R.id.ma_recyclerview_movies);
        GridLayoutManager layoutManager = new GridLayoutManager( context: this, SPAN_COUNT, GridLayoutManager.VERTICAL,  reverseLayout: false);
        mRecyclerView.setLayoutManager(layoutManager);

        mRecyclerView.setHasFixedSize(true);
        mMovieAdapter = new MovieAdapter( clickHandler: this);
        mRecyclerView.setAdapter(mMovieAdapter);
        mLoadingIndicator = (ProgressBar) findViewById(R.id.pb_loading_indicator);
```



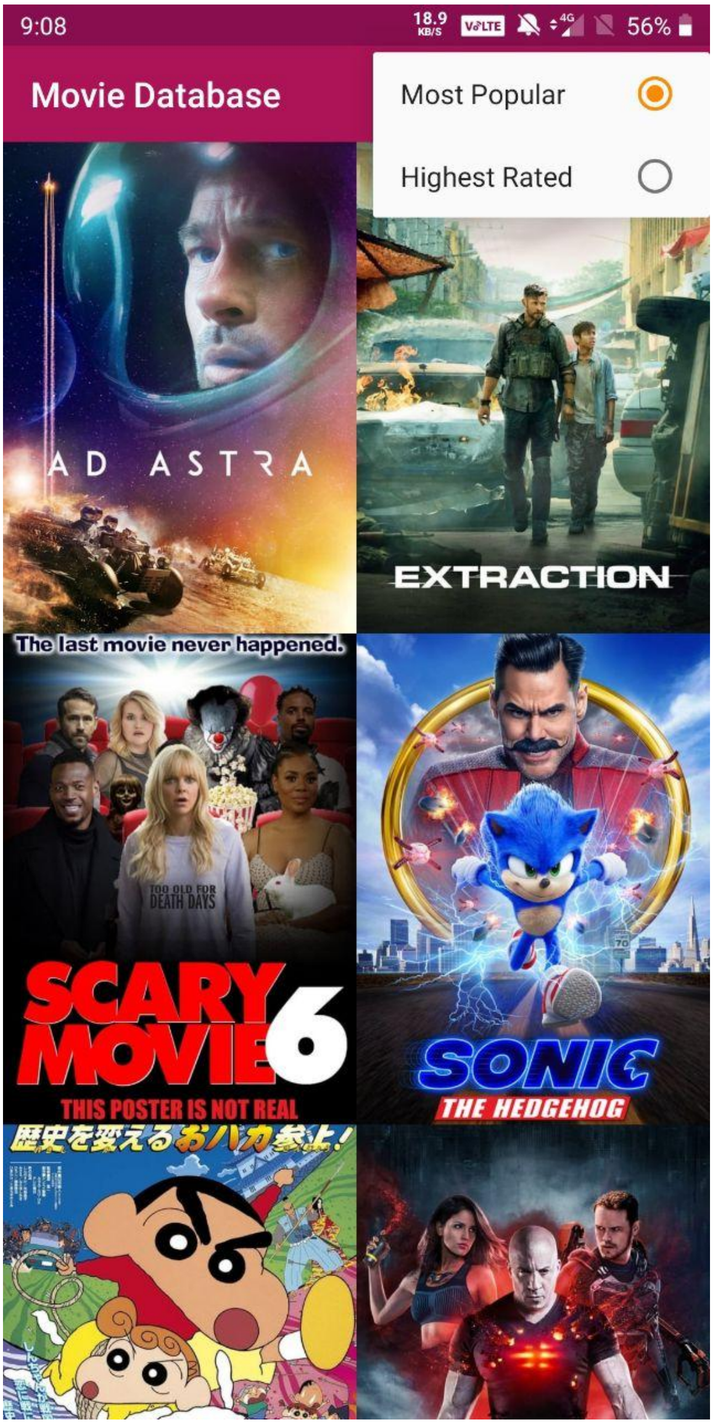
```
        loadMovieData(orderBy);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.main_menu, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.menu_sort_by_highest_rated) {
            mMovieAdapter.setMoviePosterId(null);
            item.setChecked(true);
            orderBy = ORDER_BY_RATED;
            loadMovieData(orderBy);
            return true;
        }

        if (id == R.id.menu_sort_by_most_popular) {
            mMovieAdapter.setMoviePosterId(null);
            item.setChecked(true);
            orderBy = ORDER_BY_POPULAR;
            loadMovieData(orderBy);
            return true;
        }

        if (id == R.id.menu_refresh) {
            mMovieAdapter.setMoviePosterId(null);
            loadMovieData(orderBy);
            return true;
        }
```

CHAPTER 7

SCREENSHOTS



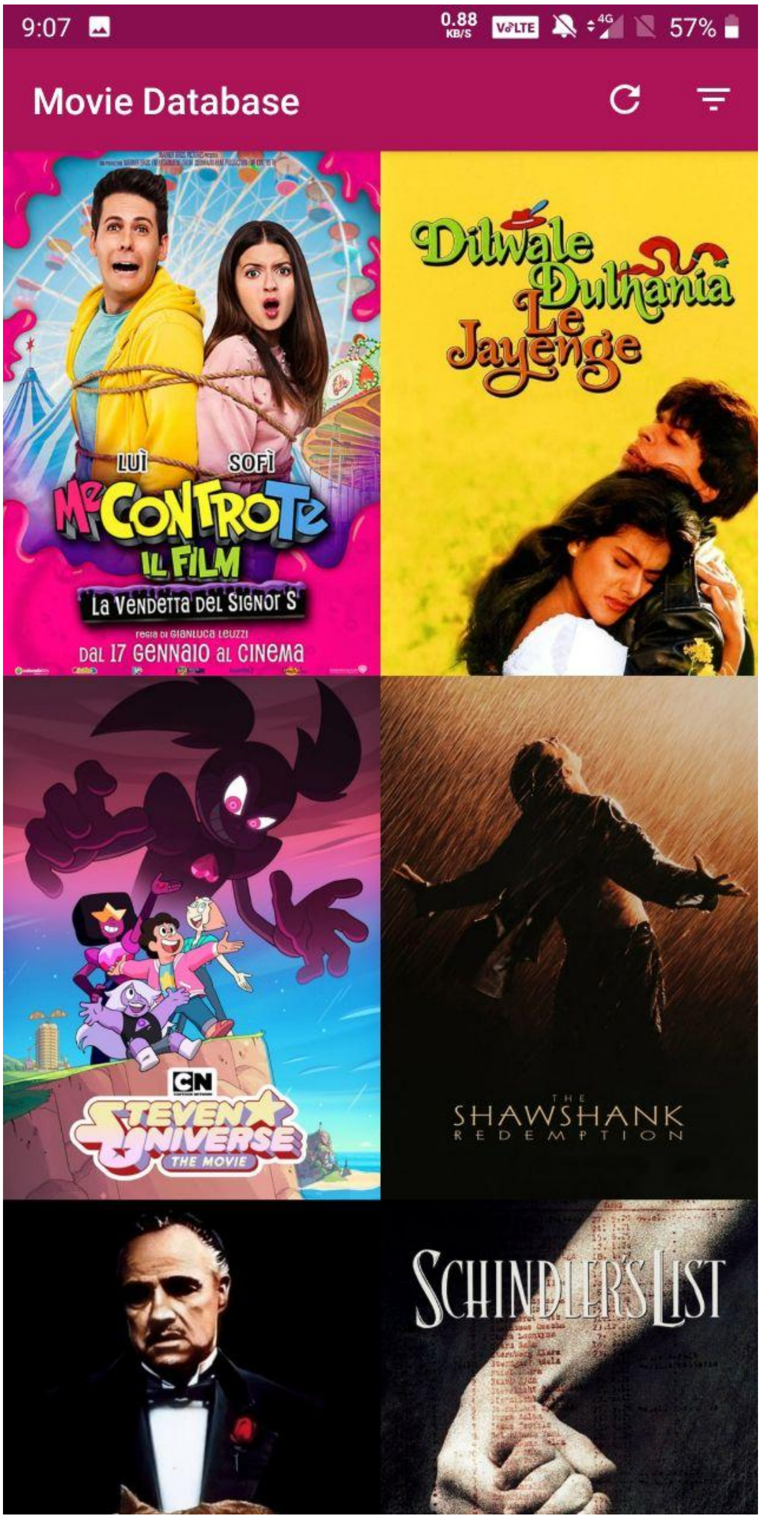
Sorry, we were unable to fetch movie data form the Internet.
Check your connection or try again later.

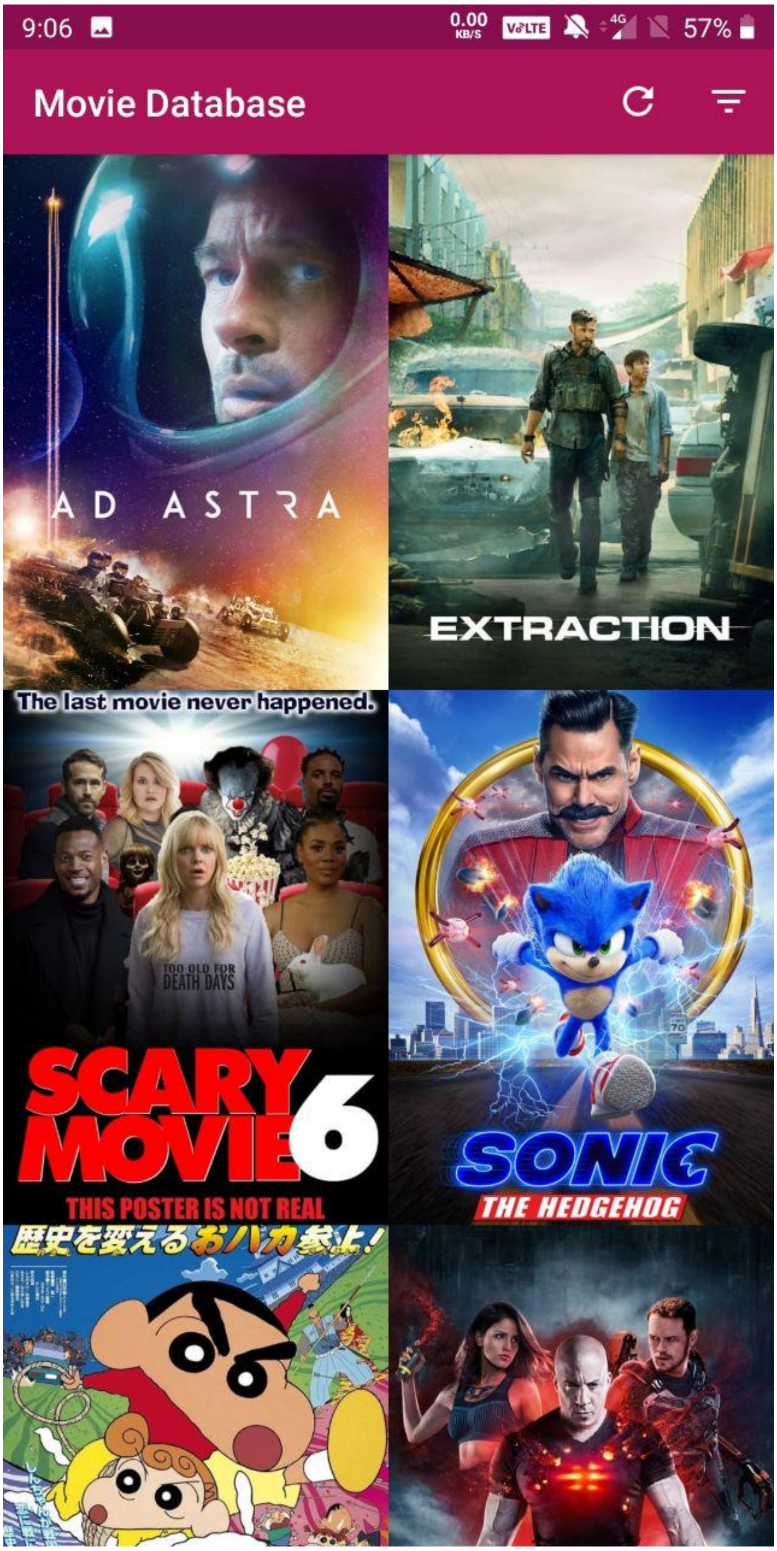


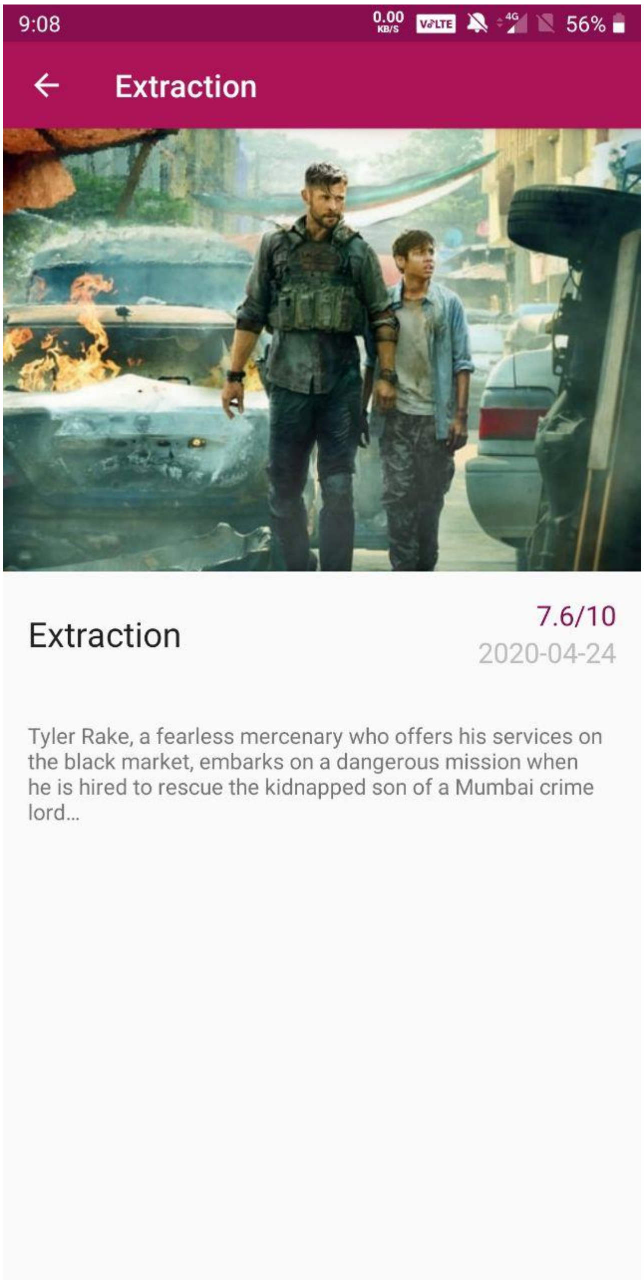
Movie Database

Most Popular ●

Highest Rated ○


9:07
0.88 KB/S
VoLTE
4G
57%
Movie Database
LUÌ
SOFÌ
Me Contro Te
IL FILM
La Vendetta del Signor S
regia di Gianluca Leuzzi
dal 17 gennaio al cinema
Dilwale Dulhania Le Jayenge
CN
Steven Universe
THE MOVIE
THE SHAWSHANK REDEMPTION
SCHINDLER'S LIST


9:06
0.00 KB/S
VoLTE
4G
57%
Movie Database
AD ASTRA
EXTRACTION
The last movie never happened.
TOO OLD FOR DEATH DAYS
SCARY MOVIE 6
THIS POSTER IS NOT REAL
SONIC
THE HEDGEHOG
歴史を変えるおバカ参上!



Extraction

Extraction

7.6/10

2020-04-24

Tyler Rake, a fearless mercenary who offers his services on the black market, embarks on a dangerous mission when he is hired to rescue the kidnapped son of a Mumbai crime lord...

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion:

Moviedatabase is an android mobile application developed using REST Interface [2], which provides movie lovers with the most updated movie information and this helps them save time and money. Moviedatabase provides the below features to users:

View the List of movies and their posters

Should be able to search for movies

Should be able to navigate between different tabs and view movies based on different categories like Popular, Top rated and Upcoming

Users should also be able to read the ratings, movie synopsis, and release information about the latest movie. Moviedatabase is developed using different technologies Java, Android and SQLite and it is integrated together with android libraries such as Picasso and Retrofit. This app is extremely useful for all people as they can view all the movie information at any place and time in their mobile. This in turn will help people save time and money as users can decide what movie they want to watch in theatres. The User Interface of this app is also so simple that any user can use this app with ease. It does not require any user account creation or payment and hence any user can use it for free.

8.3 References

- https://stackoverflow.com
- https://youtube.com
- https://www.google.com
- https://github.com