GitHub Username: aayush287

Movie Review

Description

Movie Review app allows users to check popular movies, Upcoming movies, or Top Rated moves. It shows ratings, Poster, Cast, Similar Movies, Trailers, etc. App provides the user to Mark their favorite movie and share movies with their friends. The Movie Review app completely is written in Java. It uses TmDb API to request movie Data.

Intended User

The app specifically targets users who are very busy in their life and has no idea what is going on in the entertainment world. This app allows them to take a quick sneak of top movies or upcoming movies and mark their favorite movies as a favorite.

Features

List the main features of the app. For example:

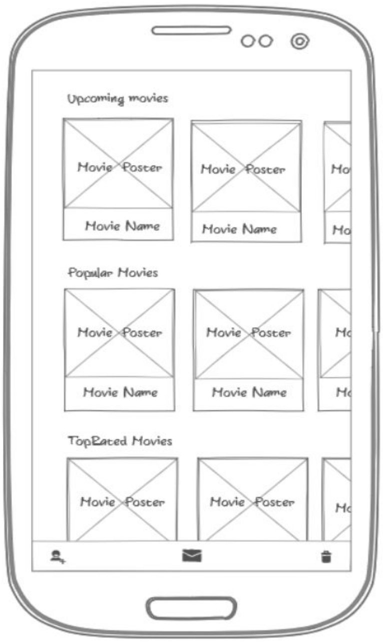
- Trailers of upcoming movies.

- Mark as my favorite.

- Search for specific movies.

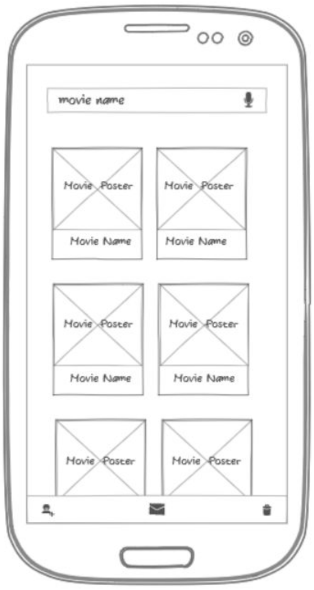
User Interface Mocks

Screen 1



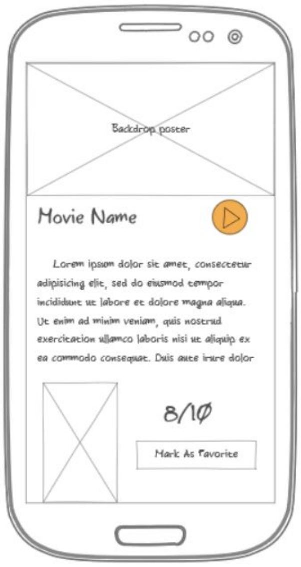
This is the main screen of the app where all movies are shown using different recycler views. On the main screen, there are three different fragment one for all types of movies as shown above.

Screen 2



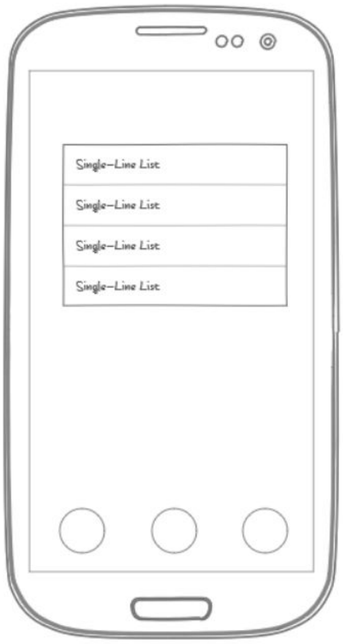
This is Search activity where users can search for any movie by name. For searching the movies we will be using AsyncTask.

Screen 3


Backdrop poster
Movie Name
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
8/10
Mark As Favorite

This is Detail Activity where users find all the information about the movie and can watch trailers of that movie. Apart from that users can see more images cast and similar movies of the selected movies.

Screen 3



This is the widget of the app where users can see a list of favorite movies and selecting a movie opens to detail activity of that movie.

Key Considerations

How will your app handle data persistence?

Data is handled using Room library and Retrofit to call the API. We will be using TmDb API to request movie data.

Describe any edge or corner cases in the UX.

We will be using ViewModel and LiveData for querying the data from API as well as from the database. So if the user exits detail activity after pressing the favorite button then it will be handled by ViewModel as it is a lifecycle aware component.
Also, we will create different layout for differents different screen sizes.
This app uses all common string in string.xml file. This app will support RTL switching and App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

Describe any libraries you'll be using and share your reasoning for including them.

- I will be using a stable version of the Room library (2.2.5) for data storage.
- Picasso (2.71828) for image loading and caching.
- Retrofit2 (2.5.0) for networking calls.
- Lifecycle extension(2.2.0) for Architecture components.

Describe how you will implement Google Play Services or other external services.

- Admob(19.1.0) to show ads.
- Analytics to get brief data.

Next Steps: Required Tasks

Task 1: Project Setup

The first task is to set up the project. To set up the project following steps are followed

- Adding stable libraries.
- Creating MVVM architecture for clean code.
- Creating a Repository to make networks call.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Detail Activity
- Build UI for Search fragment.

Task 3: Adding ViewModel and LiveData

- Adding ViewModels and LiveData
- Check for network requests i.e. if we are getting the right data or not.

Task 4: Integrate Network data to UI

Integrate the data received by network call to UI.

Task 5: Adding Room logic to store favorite movies

Adding database functionality to store favorite movies.

Task 6: AdMob and Analytics

Adding Admob and analytics.

Task 7: Creating a widget

Creating a widget that shows users' favorite movies.