Contents

GitHub Username: Lepatlamass

LLMOVIES

Description

LLMOVIES is remaster version of the popular movie app using Google material design principle to show case Movies, TV Show and search them, its main aim is to help users search and view their best movies and series.

Intended User

Our users are people passionate about film and TV Shows.

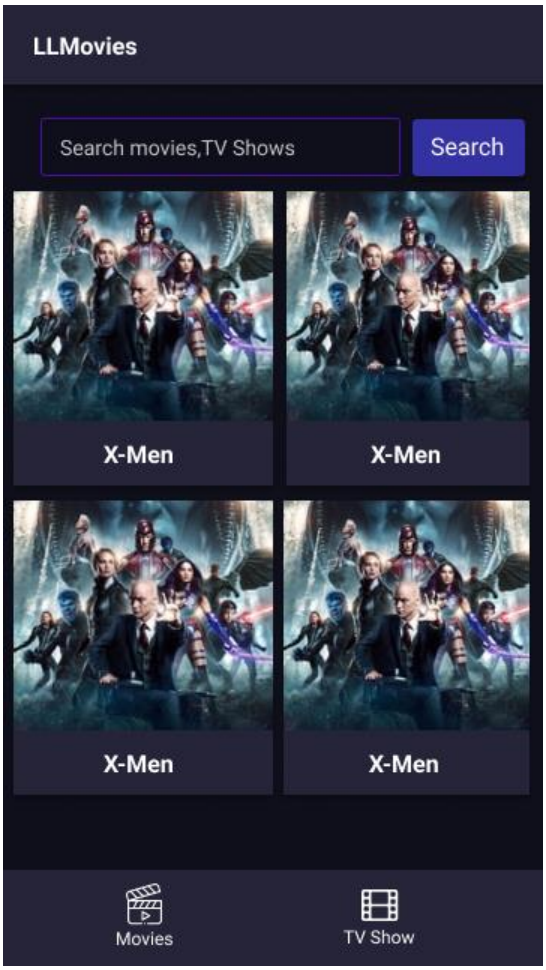
Features

List the main features of your app. For example:

- Search Movies and TV Shows.
- View Movies and TV Shows.
- Rate Movies and TV Shows.
- View Reviews.

User Interface Mocks

Screen 1

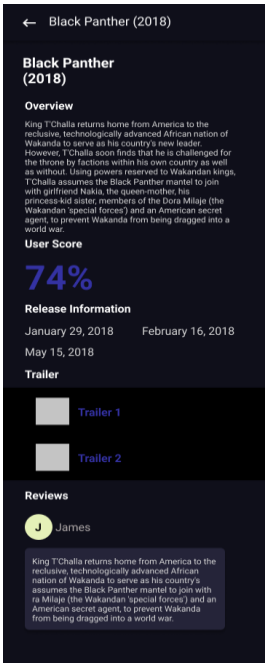


When you open the app it will display the latest films, TV shows according to which tab you clicked and also give you the possibility to search movies or films.

Screen 2

Shows all the details of TV show or Movie

Screen 3



Key Considerations

How will your app handle data persistence?

Expenses data will be persisted using Room Database, use also Shared Preferences to store other necessary information.

Describe any edge or corner cases in the UX.

The app will always load latest movies on launching.

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

- **Room Database**: to store our expenses data without SQLite boilerplate code. (version: 1.1.1)
- **Lifecycle**: to make UI automatically responds to lifecycle events. (version: 1.1.1)
- **Glide**: to load and cache our images. (version: 4.8.0)
- **Retrofit:** to turn Http API into java interface. (version: 2.4.0)
- **Material Component**: to use google material design component. (version: 1.0.0)
- **Butterknife**: to bind views. (version: 10.0.1)
- **Constraint-layout**: to well organized are layout. (version: 1.1.3)
- **Support Library**: Android support library (version: 28.0.0)

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

Describe which Services you will use and how.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create the project on Android Studio
- Add gradle Dependencies for third-party libraries
- Create the app structure by adding packages for app components.

Task 2: Configure themoviedb APikey

- Create account as a developer at themoviedb
- Create and application
- Get the Apikey
- Configure your gradle Apikey.

Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity to search and display latest films or Series
- Build UI to display a Movie or TV Show
- Build UI to display search movie and TV Show.

Task 4: Setup API Calls and Interface

- Create class for Api calls
- Create Interface for the services.

Task 5: Setup Data layer

- Create Data classes

- Create Room database components: DAO, database
- Create expense data repository to expose data to our viewModel.

Task 6: Setup UI layer and Uses-cases components

- Create all UI classes and their related xml layouts
- Add related uses cases in viewModel classes.

Task 7: Polish UI

- Use material components to make the app beautiful.