UDACITY

PROJECT SPECIFICATION

Popular Movies, Stage 2

Common Project Requirements

MEETS SPECIFICATIONS
App is written solely in the Java Programming Language.
App conforms to common standards found in the Android Nanodegree General Project Guidelines.
App utilizes stable release versions of all libraries, Gradle, and Android Studio.

User Interface - Layout

MEETS SPECIFICATIONS
UI contains an element (e.g., a spinner or settings menu) to toggle the sort order of the movies by: most popular, highest rated.

MEETS SPECIFICATIONS
Movies are displayed in the main layout via a grid of their corresponding movie poster thumbnails.
UI contains a screen for displaying the details for a selected movie.
Movie Details layout contains title, release date, movie poster, vote average, and plot synopsis.
Movie Details layout contains a section for displaying trailer videos and user reviews.

User Interface - Function

MEETS SPECIFICATIONS
When a user changes the sort criteria (most popular, highest rated, and favorites) the main view gets updated correctly.
When a movie poster thumbnail is selected, the movie details screen is launched.
When a trailer is selected, app uses an Intent to launch the trailer.

MEETS SPECIFICATIONS

In the movies detail screen, a user can tap a button (for example, a star) to mark it as a Favorite. Tap the button on a favorite movie will unfavorite it.

Network API Implementation

MEETS SPECIFICATIONS

In a background thread, app queries the `/movie/popular` or `/movie/top_rated` API for the sort criteria specified in the settings menu.

App requests for related videos for a selected movie via the `/movie/{id}/videos` endpoint in a background thread and displays those details when the user selects a movie.

App requests for user reviews for a selected movie via the `/movie/{id}/reviews` endpoint in a background thread and displays those details when the user selects a movie.

Data Persistence

MEETS SPECIFICATIONS

MEETS SPECIFICATIONS

The titles and IDs of the user’s favorite movies are stored in a native SQLite database and exposed via a ContentProvider
OR
stored using Room.

Data is updated whenever the user favorites or unfavorites a movie. No other persistence libraries are used.

When the "favorites" setting option is selected, the main view displays the entire favorites collection based on movie ids stored in the database.

Android Architecture Components

MEETS SPECIFICATIONS

If Room is used, database is not re-queried unnecessarily. LiveData is used to observe changes in the database and update the UI accordingly.

If Room is used, database is not re-queried unnecessarily after rotation. Cached LiveData from ViewModel is used instead.

Suggestions to Make Your Project Stand Out!

- Extend the favorites database to store the movie poster, synopsis, user rating, and release date, and display them even when offline.

- Implement sharing functionality to allow the user to share the first trailer’s `YouTube` URL from the movie details screen.

Student FAQ