GitHub Username: kkyn

MyMovieApp

Description

This app. collates a list of top movies as rated by public opinion(other movie-goers).
For each movie, it provides a short storyline about the movie,
public reviews and accompanying trailer.
This app. should help movie-goers to decide the kind of movies they
wish to watch after looking through the reviews and other information.

Intended User

This app appeals to movie-goers who may wish to get the latest updates of movies.

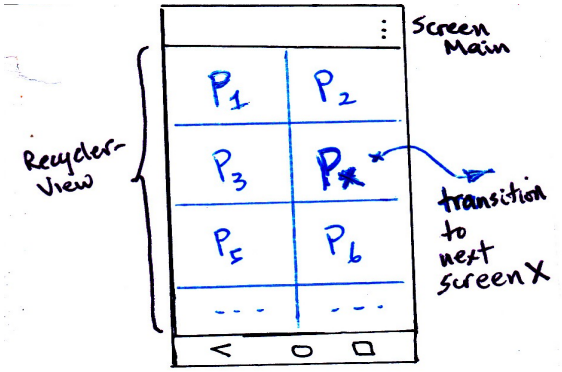
Features

The main feature about this app. are

(1) Allow users to see the list of popuar movies available.
(2) Enable the user to view short movie trailers.
(3) It allows the user to store their favorites for future access.
(4) It will have an immersive user experience using Design Material principles.

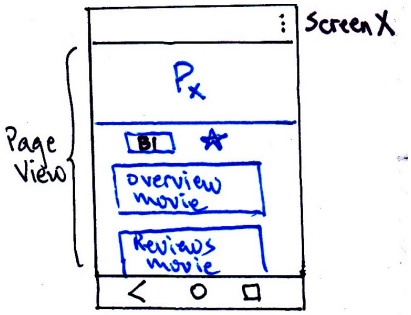
User Interface Mocks

Screen 1



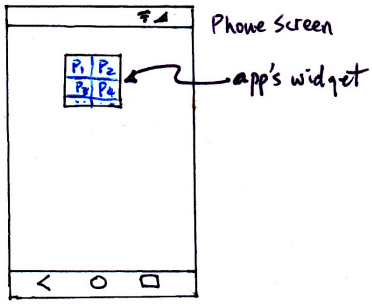
Screen Main appears when the app start.
The user can scroll the list and select which movies they wish to explore more.
Selecting the movie image will transition the app to the next screen X.

Screen 2



Screen X will show more information about the selected movie.
'Clicking' on the button **B1** it will open up a new app that will show the movie trailer.

Screen 3



Screen 3 shows how the widget's view will be like for my app.
It will show the same view as seen in **Screen 1**.

Key Considerations

How will your app handle data persistence?

I will design a content-provider structure to manage the SQLite-Database.
Data for this app will be retrieve from the server at 'https://www.themoviedb.org/'

As this app requires periodic updates, a SyncAdapter approach will be
used to get updates from the server,
and within the app. loader<cursor> will be used to manage
updates in SQLite-Database and UI.

Describe any corner cases in the UX.

When the user select the back button, the UI will return to the previous screen.
The back transition will be a smooth intuitive process.

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

(1) Use the support libraries so that older versions of Android can use the App.
Design the app for latest version of Android but it can still support the older version of Android.
(2) Volley's ImageLoader will be used to cache and load images.
Volley is used as it fast, efficient and cache size is customable.

Describe how you will implement Google Play Services.

I will implement Firebase-AdMob and Firebase-Analytics features.

Next Steps: Required Tasks

NOTE: After Task 1 (Project Setup), the other tasks are not neccesary done sequentially.
It reminds the designer the main components/considerations needed during the incremental designing process.

Task 1: Project Setup

(1) Initiate a project, follow the process wizard.
(2) Add libraries that you may think you need, especially the support libraries in the app's gradle file.
(3) Make sure the compile/android versions are what you want in the gradle file.
(4) Make sure the latest gradle version is used.

NOTE: the compile dependecies, details in the manifest file will grow as the project

details grow. Like when you need have permission for internet access.

Task 2: Implement UI for Each Activity and Fragment

(1) From UI-drawings, create UI layout xml files to model the UI-drawings
(2) Add codes in the Activity's file to refer to those vies in the xml files.
(3) Build a Activity with it's accompanying Fragment, to enhance function modularity.
(4) Incrementally and iteratively, add features to realise the final expected UI.

NOTE: In the beginning of design, target the design for Phone's form-factor.
When everything functions properly then enhance the app for Tablet's form-factor.

Task 3: Design the content-provider.

(1) Know where you will get your information from the 'cloud' to implement this app.
(2) Understand the call format, e.g. json format.
(3) Build the content-provider based on the information from (1) and (2).
(4) Design the data syncronization scheme between the server and the content-provider.

NOTE: Decide the type for syncronization scheme that fits the task for this app. (make trade-offs based on complexity, efficency). Always start simple to make the app functional first.

Task 4: Design RecyclerView, ViewPager Adapters

(1) RecyclerView is for the 1st UI to display the list of movies of interest.
(2) ViewAdapter is used by the 2nd UI, to display some introductory information for a selected movie.

Task 5: Design for quality

Constantly consider design trade-offs to achieve quality design.
(1) Power.
(2) Responsiveness.
(3) Usage of Materials Design principles.

Task 6: Design for the tablet form-factor (Future Work)

(1) Upgrade the code to for tablet.
(2) Upgrade the layout file for 2-pane, when at landscape orientation.