How to Use this Template

1. Make a copy [File → Make a copy…]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: Github.com/Akshit94

MOVIELIB

Description

Problem:

With the increase in the development of different kinds of films it's hard to keep track of the movies that we really love or would love to see eventually. A movie fanatic may want to see a particular genre movie or show but don't want to search for hours on internet to get the best

movie he/she could see. A fan may want to see only those movies in which there is a particular famous personality that he/she loves but is not able to keep track of that famous personality's upcoming movies. A user who wants to watch a movie but could not decide whether it will be worth his/her time and money as he/she is not able to get accurate and honest movie reviews. There are many other problems that user is faced with like watching trailers on the go, writing your own reviews for a movie, searching for old movies, unable to see movie information without internet connectivity and the list goes on.

Proposed Solution:

MOVIELIB is a platform where a user is served with popular and upcoming movies around the world. A user can search for movies using their names, actors and actresses, genres etc. A user can also save or bookmark his/her favourite movies to view offline even without internet connectivity. They can see trailers, read and write reviews, see the movie's plot and characters, look out for popular and upcoming movies on the go, search for famous personalities, see movie's different posters and images and many more. MOVIELIB can also be used as a personal movie scrapbook and can be used to save movies. A user could also share the movie or show he/she loves with anyone through any application and spread the word about how excited and crazy he/she is about that movie.

Intended User

MOVIELIB is especially made for all those movie lovers and fanatics that can’t live a day without watching movies. MOVIELIB can also be used by all those who wants information about some movie or wants to watch trailers or read or write reviews about a movie they have already watched or/and are going to watch. Users of all age groups are wholeheartedly welcome.

Features

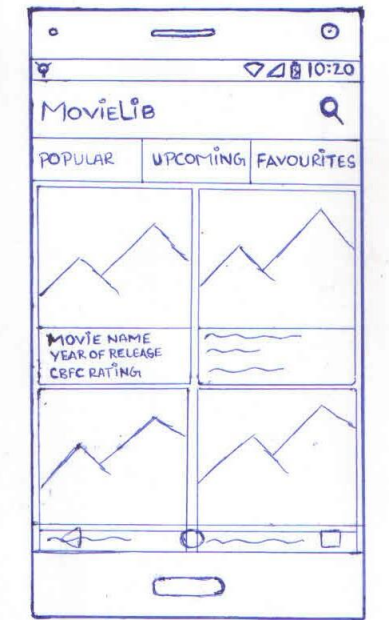
MOVIELIB includes the following features:

- Saves movie information for viewing offline.
- Watch movie trailers on YouTube.
- Share movies with friends and families.
- Read or write reviews.
- Search for movies.
- View upcoming and popular movies.
- Filter movies on the basis of genres, rating etc.

User Interface Mocks

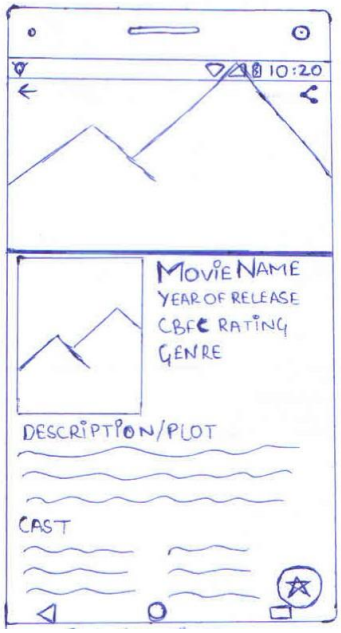
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



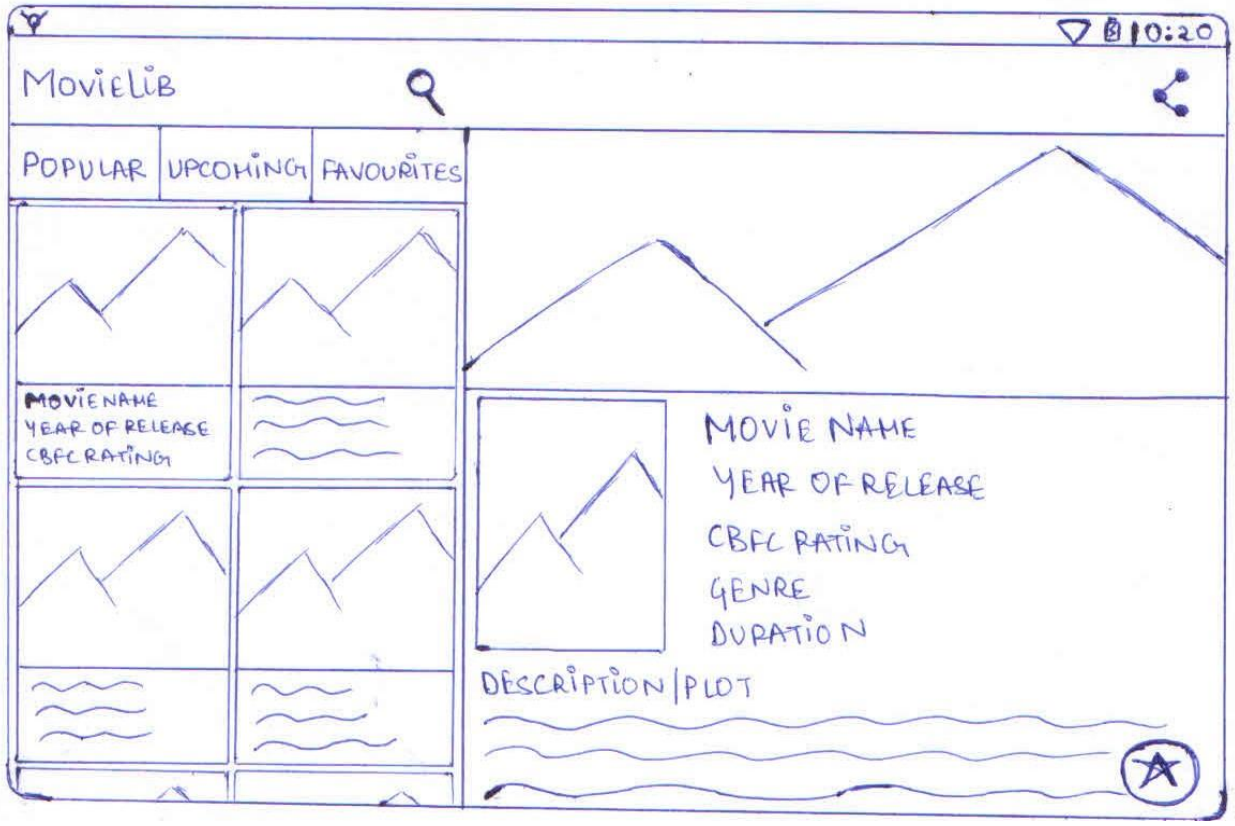
- This screen will be first displayed to the user when he/she first opens the app.
- This will show the user a grid of the most Popular movies chosen by the TMDB.
- The user can navigate to different screens such as the new upcoming movies and favourite movies marked by them using the ViewPager Tab.
- Clicking on each grid item will navigate the user to the Details Activity of that specific movie.
- The user can also search for movies using the search Action button on top.

Screen 2



- This activity will be shown whenever the user touches a movie on the main activity.
- This will show the users all the remaining information of the movie that cannot be seen on the main activity such as duration of the movie, description/plot of the movie, cast of the movie, genres of the movie, trailers and reviews.
- The user can watch trailers of the movie inside the app itself.
- A user can read and write reviews about a movie.
- A user can mark the movie as favourite to view it offline using the floating action button.
- A user can share certain information about the movie using the share action button on the toolbar.

Tablet Layout



- This is what a user will see when using the app on a tablet.
- To compensate for the white space and ease of use both details and the main activity have been added to the same screen as fragments.
- The buttons actions are same as they are on the phone.
- Selecting a grid item will change the movie details on the detail fragment.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

My app will handle data persistence using a content provider that I would manually create to store the movies data for viewing for offline purpose. I will also be using shared preferences to store user settings locally.

Describe any corner cases in the UX.

My app implementation will try consider any corner cases that the user might encounter in order to avoid strange behaviours such as:

- In case of empty MainActivity, the user will be treated with the correct information about why the user is not getting data.
- In case of image not loading, the ImageView will be filled with placeholder images.
- In case of no network availability, the saved favourites data would still be available to the user and can easily navigate to them using the ViewPager.

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

I will be using the Picasso library for loading images via the web into the image views. I will also be using the TheMovieDb API (TMDB) in order to get the movies data from their database. animations to Some material design libraries such as the "Design Support Library" will also be used to add transitions and animations to the app

Describe how you will implement Google Play Services.

Google Play Services that I will be using will include:

1. YouTube Data API: I will be using the YouTube Data API to play movies and TV shows trailers inside the app. Thus, preserving the user experience, making the user spend more time in my app rather than navigating to the YouTube to see the video.

2. Google Analytics API: I will be using the Google Analytics API to track the user experience and analyse the app crashes to better make the app.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Search and setup libraries that may be required for development.
- Implement requirement analysis and design the flow of the project.
- Download the required SDK components that will be required for the project such as keeping the Google Support Repository up-to-date.
- Setup the TMDB API and get an API key.
- Setup the YouTube Data API and get an API key.
- Setup the Google Analytics API and get an API key.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity.
- Implement the ViewPager tabs and their respect activities UI.
- Add empty view and its logic to change in case of no data to the MainActivity's grid.
- Build UI for DetailsActivity.
- Implement the layouts for landscape mode.

Task 3: Implement the logic for the TMDB API and create Database.

- Make constants to pull out the data from the JSON response.
- Implement an AsyncTask to pull out the data from the TMDB on a background thread using HttpsUrlConnection class.
- Check the status codes returned from the TMDB to display the correct result to the user in case of data not being displayed.
- Create a database contract and implement the database helper methods in order to store the movies data offline.

Task 4: Implement the Tablet UI.

- Design the container UI for tablets in order to display the Main and Details layout together.
- Check the app's compatibility with different sized tablet like 7inch and 10inch.
- Compensate for the extra white space accordingly for different sized devices using the Resource Folder Qualifiers.

Task 5: Add logic to all the action buttons and FAB.

- Add the share intent to the share action button on the DetailsActivity to share the movie's information with your friends and families via the supported apps in the device.
- Add the offline save feature to the Floating Action Button in order to display whether the following movie is a favourite or not.

Task 6: Implement libraries.

- Implement the YouTube Data API to watch the movie trailers.
- Implement the Google Analytics API the analyse and record the activity of the users that are using my app.
- Implement the Picasso library to load images from the web.
- Implement the components of the android design support library if any used in the app.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"