How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: sergeD97

Movie App

Description

This application that allows you to see the posters of different movies. The app offers the possibility to the user to be able to store his favorite posters, see the comments on the film and see the trailers on youtube. Application will use the movie db API to fetch movie information.

Intended User

Movie fans

Features

App features:

- Show Movie Poster
- Save favorite movie poster
- Authentification using firebase
- Show movie comment

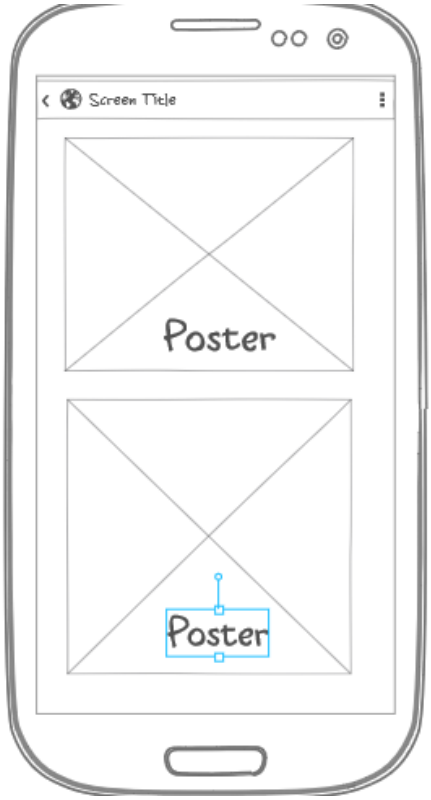
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1

The login activity : Allow user to access to the home page of the app. Authetication is made with google account.

Screen 2



The home page : Display the movie poster according to the sort criteria.

Screen 3

Detail Movie activity : Display the selected movie details, and a button that allows user to mark movie as “favorite”

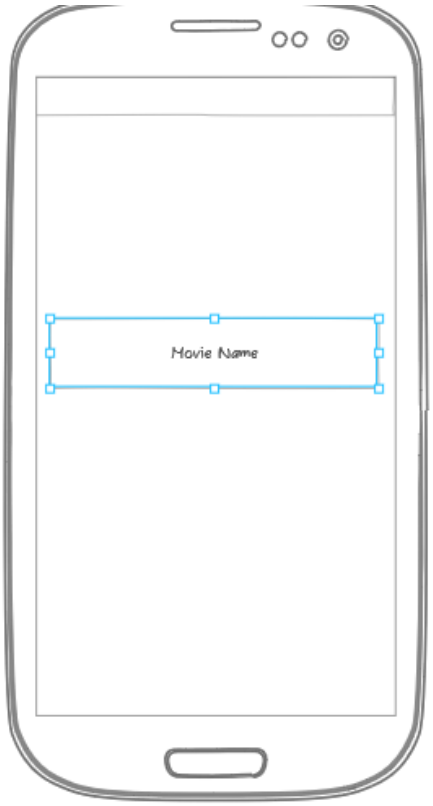
Screen 4

This activity display the list of reviews of a selected movie.

Screen 5

This Activity display a screen setting that allow user to select the sort criteria(sort criteria can be : Favorite, Most Popular, and More Rated)

Screen 6



The App widget that display the movie that the user have see. User can click on it and launch the details screen

Key Considerations

How will your app handle data persistence?

- App will use SharedPrefences to persiste information about the sort criteria : There is three kind of sort criteria :
- Favorite : to make the app only display the Movie that are stored in the device.
- Most Popular : to make the app display the most popular movie.
- More Rated : to make the app display the more rated movie.

- App will use room library to persist the favorite movie.
- App will use AsynctaskLoader to make a request in a separate thread and to temporally cache data(for example when the device configuration change)

Describe any edge or corner cases in the UX.

Unstable or missed network connection: the application must not crash in that cases

• **Device orientation change**: the application must handle all long running operations correctly

considering possible configuration
changes
• **UI freezes:** the application must not use the main thread for any
resource consuming operations

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

- Picasso (com.squareup.picasso:picasso:2.5.2): to handle the loading and caching of images.
- Butter Knife(com.jakewharton:butterknife:8.5.1): for boilerplate code reducing
- Material Values(version 27.1.1): for handy Material Design dimensions access
- Firebase (com.google.firebase:firebase-auth:16.0.1, com.google.firebase:firebase-core:16.0.0, com.google.firebase:firebase-auth:16.0.1): for user Authentication
- Room (com.jakewharton:butterknife:8.5.1) : for local data storage
- Google Ads (com.google.android.gms:play-services-ads:11.8.0): To show testing ads

There is other technology that we will use to build the app :

- Java 1.8 : the programming langage that we will use to build app.
- Android Studio 3.0.1 : the IDE.
- Gradle 4.1.

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

- The application will use Firebase Authentication to identify the user.
- Google ads mods to show a test ads.

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 and setup a new project. This task includes:

- creating a new project in Android Studio
- configuring libraries by adding all necessary dependencies.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build Fragment

Task 3: Data model classes

Create data classes which help to handle all response data provided by MovieDB API calls.
Required classes:

- Movie
- Review
- Trailer

Task 4: Data Persistence

Add room and shared preferences helper class to handle all locally stored data.

- Create AppdataBase file
- Create DAO file for each model

Task 5: Google Play Service

Implement chosen services (Firebase Authentification and Google Ads Mobs).

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

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”