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: amanjeetsingh150

Telelove

Description

Telelove is an application which allows users to get the information about the most popular and top rated TV Shows. It provides you all the posters, rating, release dates, quotes and other visual information including trailers and posters in a go.

Intended User

Telelove is for all the people who are crazy about the TV series and are very consistent watching the episodes. It will keep them engaged with all the information about the TV series they watch.

Features

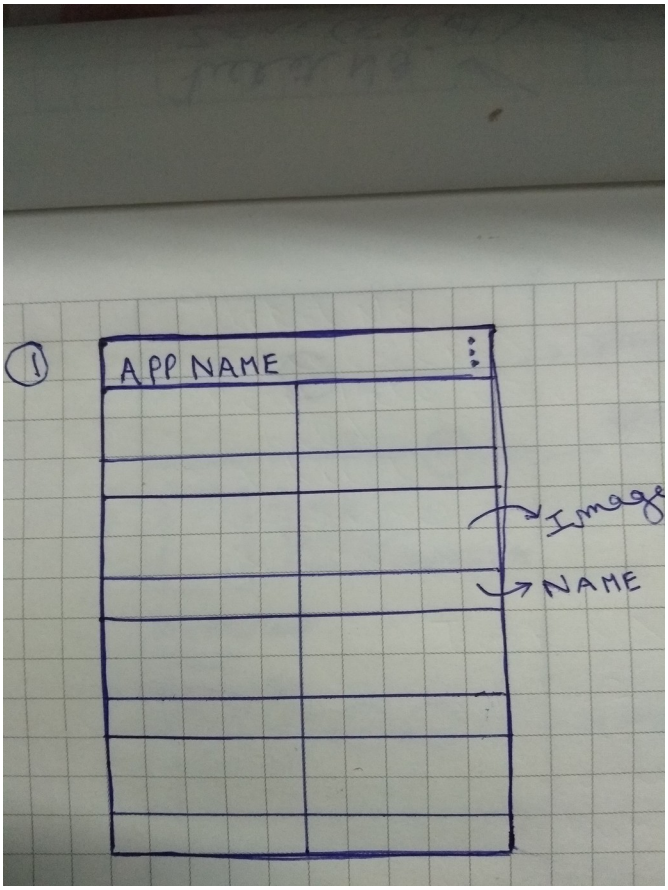
Features of Telelove:

- All the detailed information about a TV series from their overview, release, ratings.
- It also has features for viewing trailers of the TV series
- Giving periodic notifications containg quotes from different TV series.
- Having options for viewing similar shows to the viewed TV series
- Having options to view full cast

User Interface Mocks

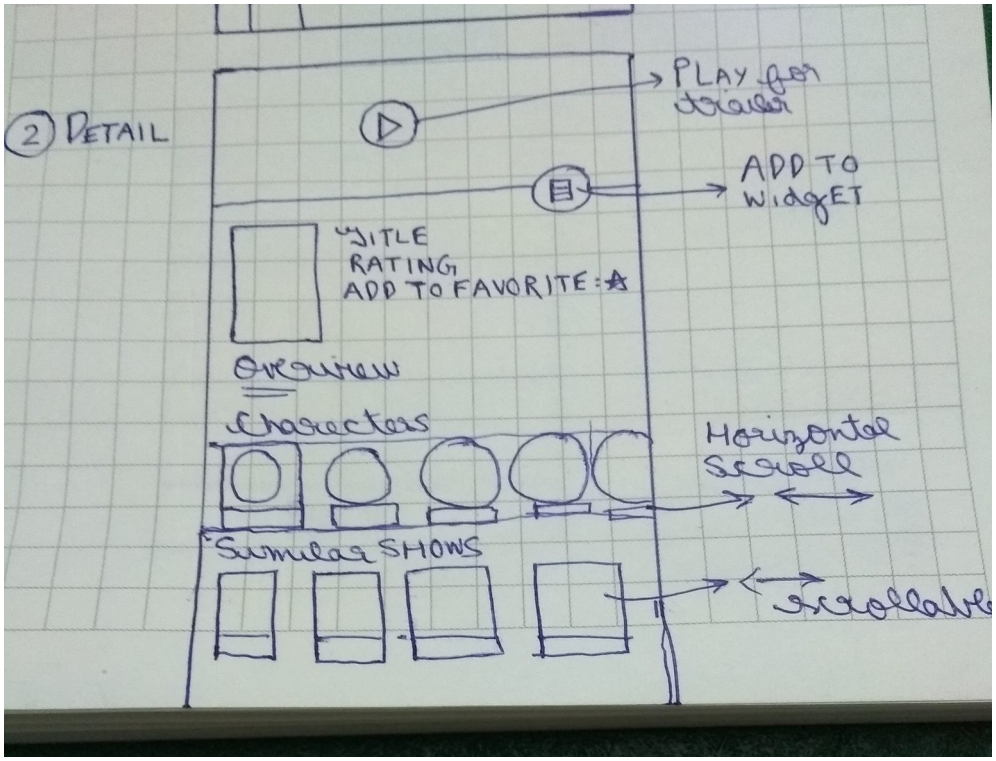
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



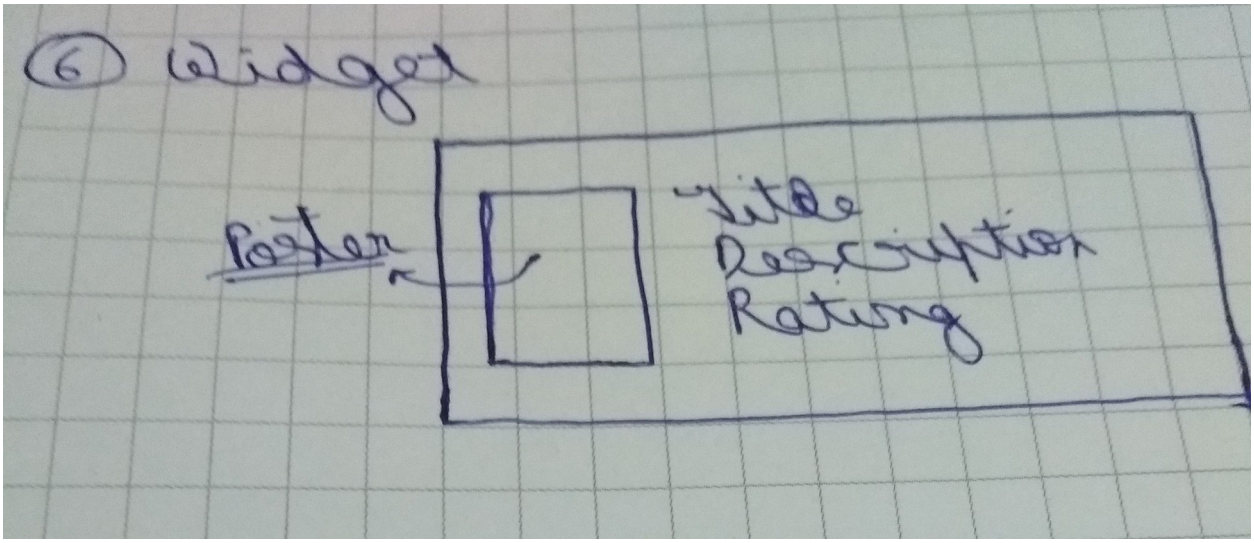
The screen shows the populated TV shows from APIs in grid.

Screen 2



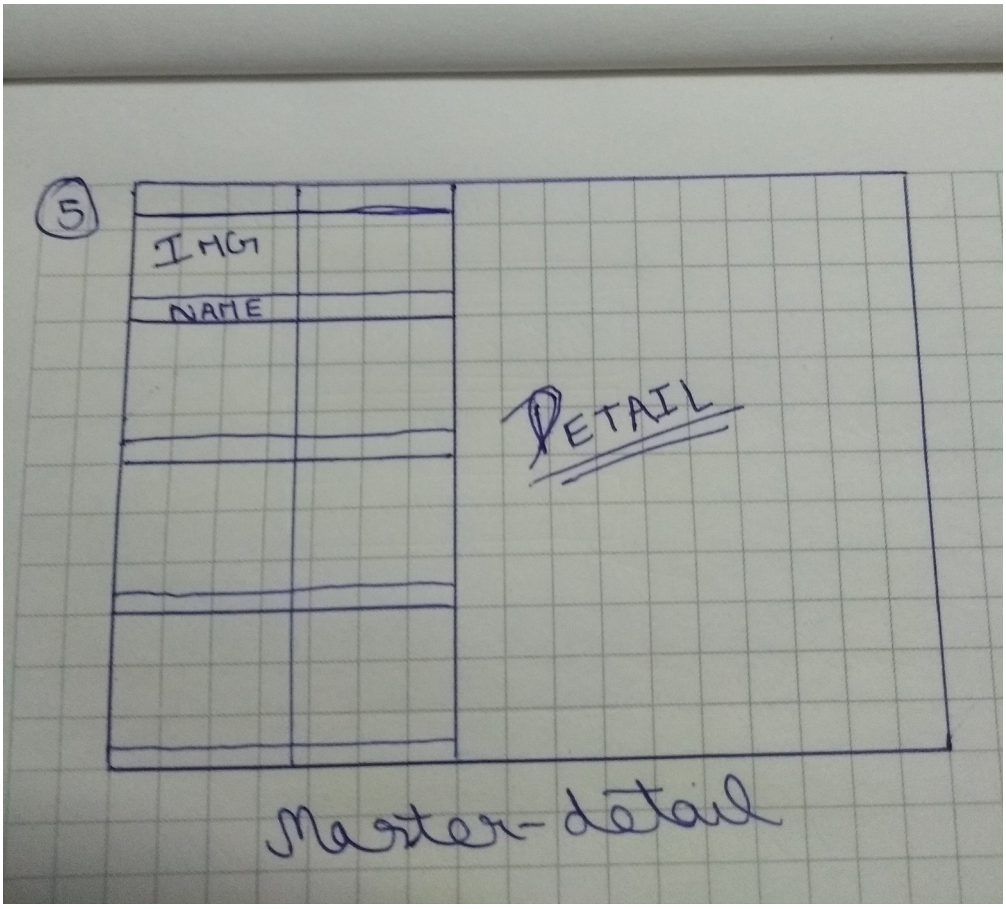
This is the Details Screen of App.

Screen 3



Widget will have these mention information abouth the TV show added by user.

Screen 4



Master-Detail layout for tabs.

Key Considerations

How will your app handle data persistence?

The data will be stored in SQLite Database (with help of Room) powered by a content provider.

Describe any edge or corner cases in the UX.

The edge case would be that users will be asked if they want the periodic notifications of quote the first time they open the app. By default these will be set to one day interval but the user can switch them off and also can customize the interval by going in settings.

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

1. Retrofit: For API calls
2. RxJava: Powered to Retrofit and helping in API calls
3. GSON: Serialization and Deserialization POJOs.
4. FirebaseJobDispatcher: For setting job periodically for geeting quotes in form of notfications.
5. Picasso: For loading of Images
6. Dagger2: As a dependency Injector
7. ButterKnife: As a view injector for avoiding findViewById.
8. Room: For having SQLite services for persistence

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

I will be using Firebase services for backend services like Firebase Realtime Database, Analytics.

Next Steps: Required Tasks

This is the place where the all required steps for completing the apps are declared by describing all the technical aspects in eleborated way.

Task 1: Project Setup

Before beginning the task I will write a scraper for getting quotes related to TV series and adding it to firebase. The first task will be adding all the required dependencies. To the studio project.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- MainActivity

a. Creating all the required widgets like recycler view
b. Declaring the important adapters related to view.
c. Setting on Click on photos for opening Details.

- Details Activity
  a. Creating the required layout by keeping all the widgets in layout.
  b. Providing options available like adding to widgets and adding to favorites.
- SettingsActivity setup by providing option for sortby most popular or Top Rated or Favorites and getting quotes options
- Setting the duration picker dialog for setting interval for quotes.

Task 3: Setting all the support utils

- Declaring all the required components for Content providers, SQLite Database components for room using all the support libraries like SimpleSqlProvider
- Creating the FirebaseJobDispatcher and setting up the notifications in a periodic interval according to the intervals.
- Creating the necessary API interfaces and completing the code for API calls throught the Retrofit and RxJava and implementing GSON for the POJO serialize and deserialization.
- Creating necessary services for updating the widget when the user add the tv series to widget.
- Setting up loaders for efficient loading in case of orientation changes.

Task 4: Setting the Firebase Realtime Database components

- Setting up instances for firebase instances to get quotes when ever job starts fetching it from databases and showing in notifications

Task 5: Setting up animations

- Setting meaning ful animations like shared element transitions and other while launching other activity.

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**"