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: jainkamini

MovieCollection

Description

Find largest collection of movies with popularity, star ratings, trailers, reviews, overview, movie genres, release date and all other information you need to know. App also shows the upcoming movies, popular movies, and top-rated movies and can create your favorite list of movies. App comes with latest UI and seamless user experience.
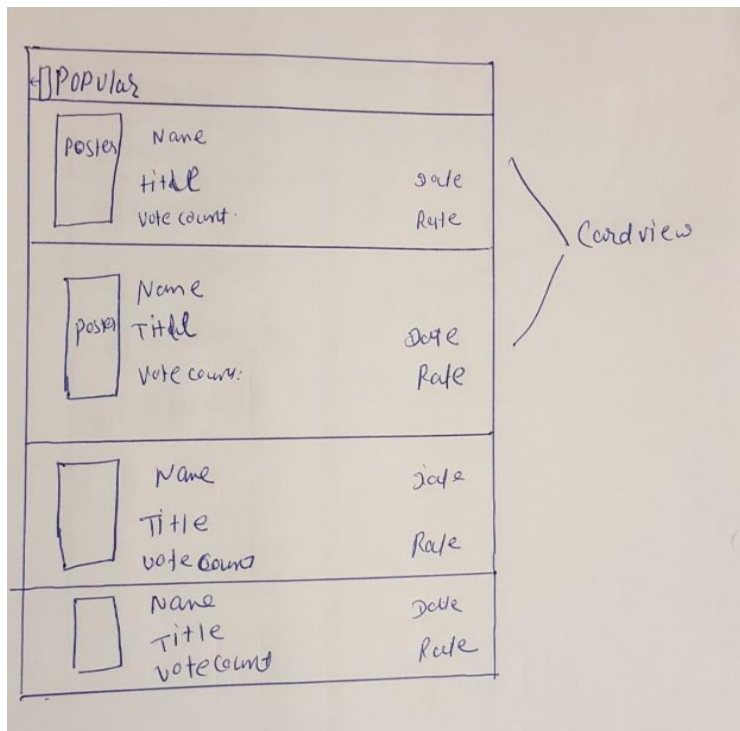
Intended User

Anyone who has interest in movies and want to know more about it.

Features

- Show popular/upcoming/top-rated Movies
- Save favorite movies
- It will display movie poster, description, status etc.
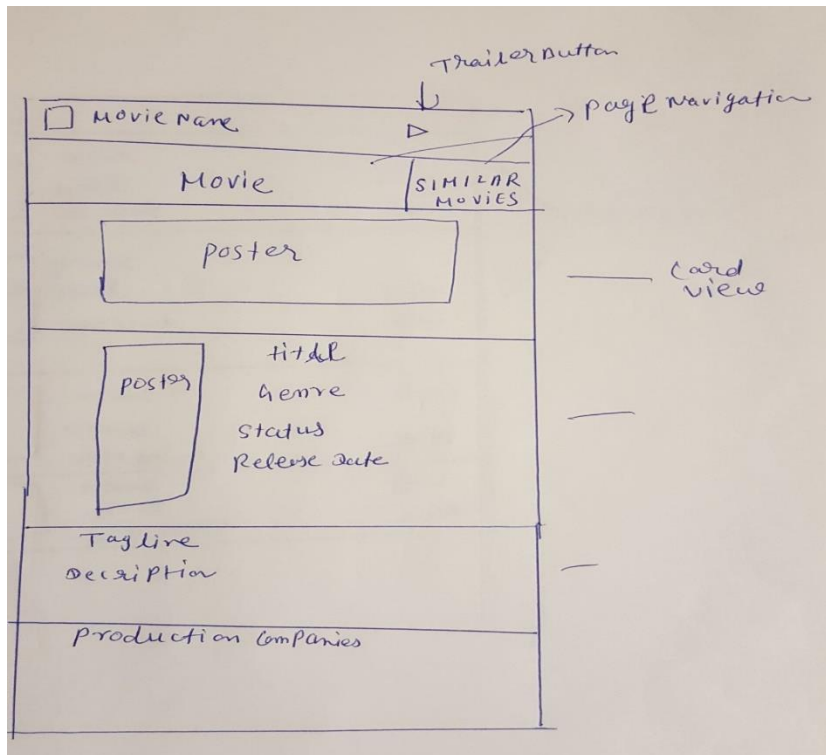- Watch trailer of movies
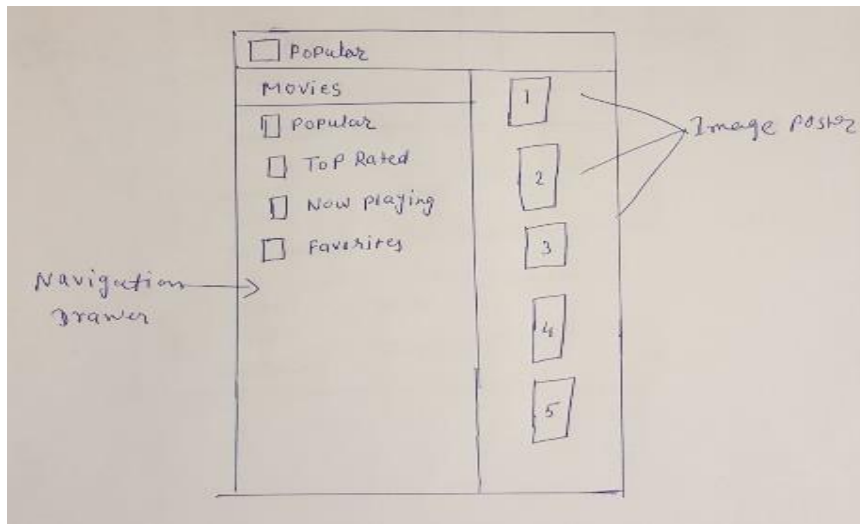
User Interface Mocks

Screen 1

Movie poster, name, title, vote count, date, rate will display on this Screen.

Screen 2

This screen gives the details about movie which was clicked on screen 1.
Also it will show two tab one for movie details and other for the similar movie list.
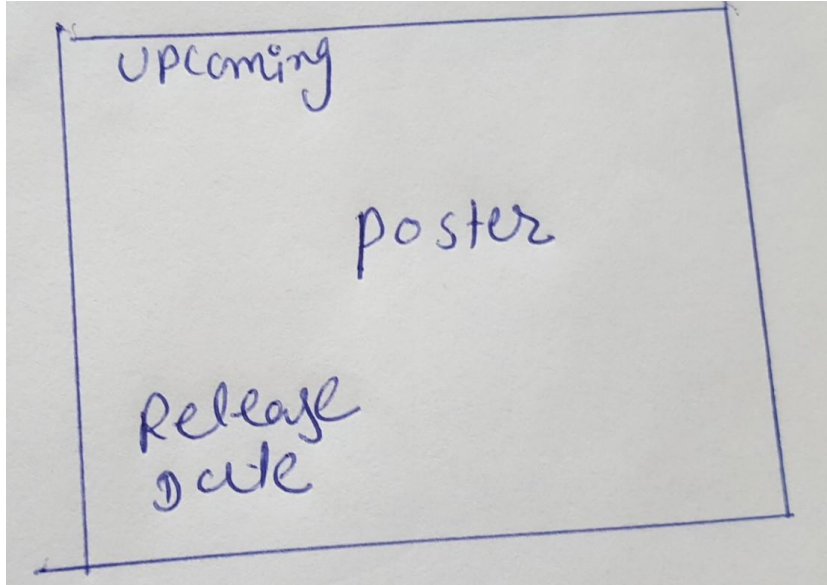Screen will have share button which is not displayed in image.

Screen 3

It will show the navigation drawer where user can select types of movies

Screen 4(Tablet View)
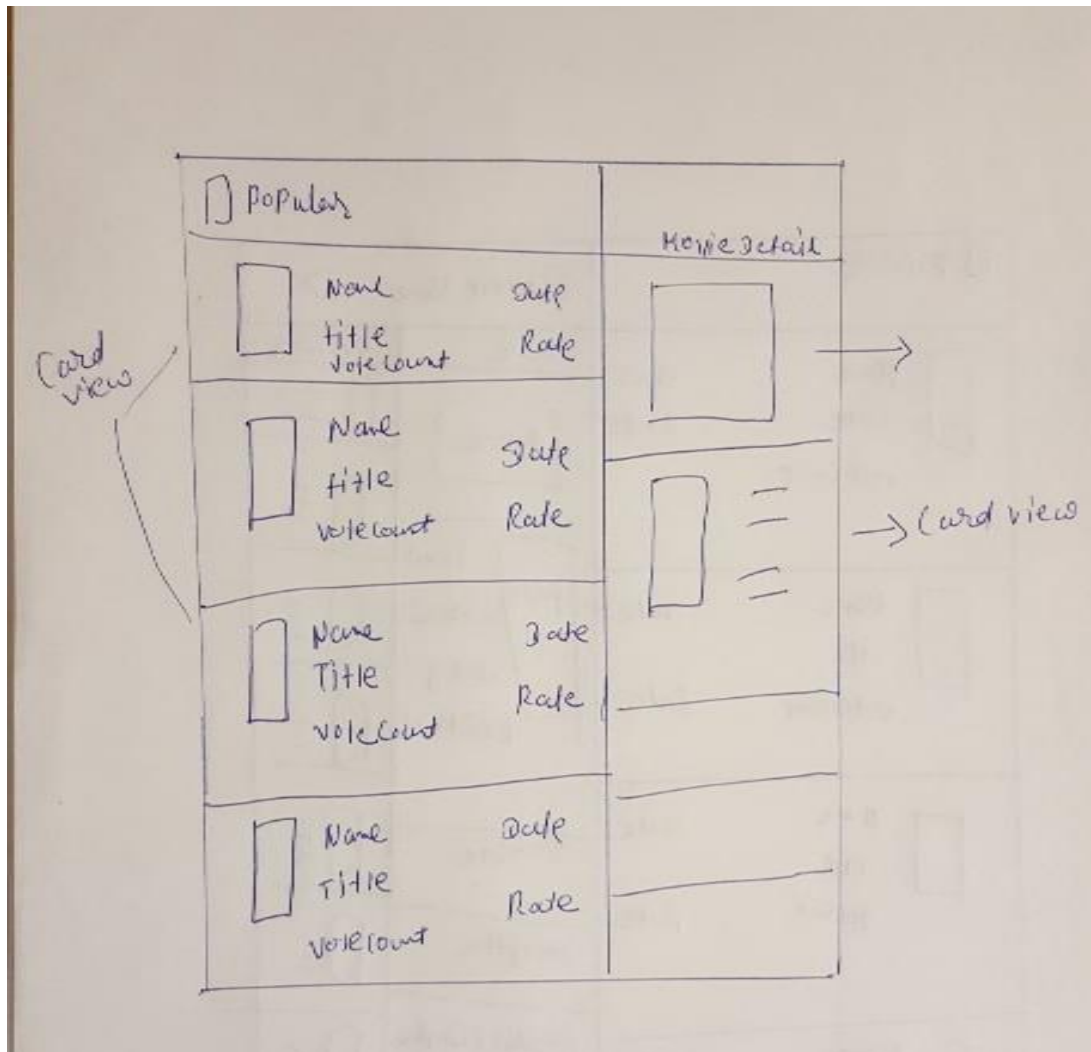
This is two pane tablet view one pane shows movie list and other pane will display detail of that movie

Screen 5

Widget screen will display upcoming movie poster title and Release data

Key Considerations

How will your app handle data persistence?

- Fetch the data from Api using intent service and save in local database
- App pull data from Api only once on per request basis
- App implements a ContntProvider to access locally store data.

Describe any corner cases in the UX.

- When user click on main screen item than it will be go to detail screen
- Hit back button on detail screen it will go on in main screen
- When hit left corner button it will open navigation drawer

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

- Picasso to handle the loading and caching images
- Retrofit for access data from server

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: Implement UI for Main screen

Build Layout for MainActivity
- Create card view for display movie list
- Set all the constants in value folder
- Configure Picasso library for Images

Task 2: Implement Data Persistence
- Configure Retrofit Library to get data
- Create all classes and interface for Retrofit Library
- Create intent service for network call
- Store data in local database
- Build content provider for fetch data in local data store

Task 3: Implement View
- Implement Load data form loader in view

Task 4: UI for Navigation drawer
Build NevigationDrawer
- Create layout for it.
- Change data according to select item on it

Task 5: UI for Detail screen
- Build Detail Screen
  ➢ Create view pager layout for this screen
- Create layout for first tab
  ➢ In first tab show detail data for selected item
  ➢ Save the data for favorite movie
- Create layout for second tab
- Fetch the data for similar item
- Use button for play trailer of movie

Task 6: Implement Accessibility and RTL Layout
Implement accessibility
Implement RTL Layout

Task 7: Implement Material Design

- Implement app bar and toolbars

Task 8: Implement Google Paly services

- Implement google play service for Admob
- Implement google play service for Analytics

Task 9: Implement widget

Create a Widget for this.

- Create layout for it
- Include classes for it

Task 10: Implement ShareActionProvider and transition

- Implement ShareActionProvider for share content
- Implement Shared element transition
- Implement parallax scrolling

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**"