GitHub Username: ritesh-singh

Dekho

Description

Dekho makes it easier for people to find their favourite Tv shows, movies and celebrities.
It helps user by showing them the popular, top rated etc.. basically movies and tv shows in different categories and also the information about the same.
The app is mobile friendly and easy to use with following feature-
1.Explore movies, tv series and celebrities.
2.Information of selected movies, series and celebs.
3.Built in trailers for movies and tv series.
4.Celebs info.
5.Gives user an option to mark as fav and unfav.

Intended User

This app is for everyone for who are into Movies and Tv - Series.

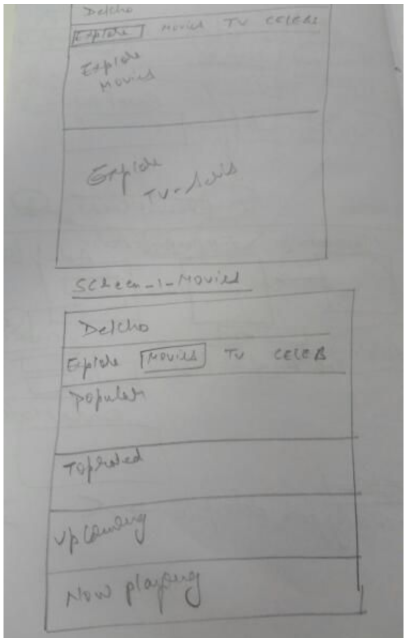
Features

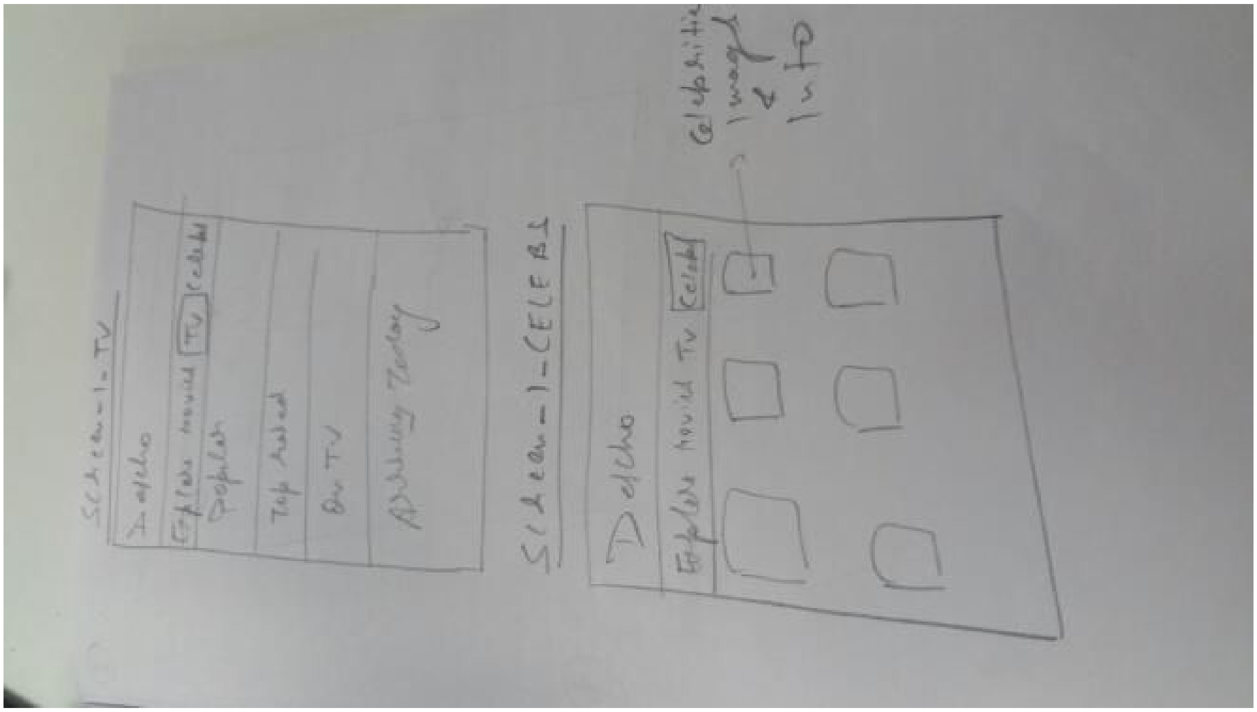
Main Features -

- Explore movies, Tv series and celebs by categories.
- See the detailed information on click.
- See the trailer and information like rating.
- Option to mark it as favourite.

User Interface Mocks

Screen 1

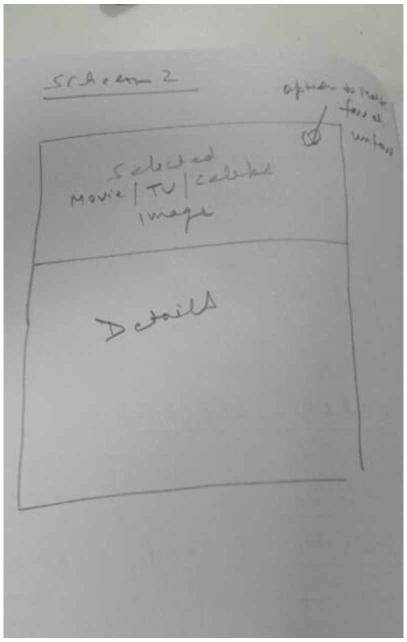




This is the different versions of the home screen with 4 different categories.For each selected category user will be able to see the details.

For example, if user selects the movies section, the below list will be populated with popular, top rated, upcoming and now playing cards.

Screen 2


Screen 2
option to mark fav or unfav
Selected Movie | TV | Celebs Image
Details

User is navigated to this screen once, he selects any item(movies, tv shows, or celebs) from home screen.
This screen holds the selected item image and it's information based on selected item type.

Since, this is the early stage of app, i will try to add more features like show data based on region, user will be able to upvote and downvote and read or write comments and watch videos in an another screen.

Key Considerations

How will your app handle data persistence?

The app will use RxJava for network call and pagination for the same. Will use Rxjava cache feature.
The app will also use database to store fav items.

Describe any corner cases in the UX.

No, corner use cases.

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

- Picasso for image loading.
- Dagger 2 for dependency injection.
- RxJava for network calls and common operations.
- Retrofit as a rest client.
- Stetho as an interceptor.
- Android design support libraries.

Describe how you will implement Google Play Services.

Interstitial Ads when user lands on the home screen.

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

- Setup MVP architecture.
- Setup libraries like Dagger 2 and Rxjava.
- Setup different layers, like data layer.
- Setup Network layer inside data layer.
- Design the flow of project.
- Implement Google Ad Services.

Task 2: Implement UI for Each Activity and Fragment

- Build splash screen.
- Build UI for each activity.
- Build fragment(view) for each activity.

Task 3: Implement Google Ad service

- Implement Interstitial Ad in UI.

Task 4: Design for Tablets

- Create detail activity layout for tablet.
- Enable layout mirroring.
- Check compatibility with older devices.

Task 5: Testing and Debugging

- Design Test cases and perform each test case accordingly.
- Analyze the bug.

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**"