
GitHub Username: ritesh-singh

Dekho

Description

Dekho makes it easier for people to find popular,top rated, now playing and upcoming movies.

The app is mobile friendly and easy to use with following feature-
1.Explore movies.
2.Information of selected movies.
3.Built in trailers for movies.

Intended User

This app is for everyone for who are into Movies.

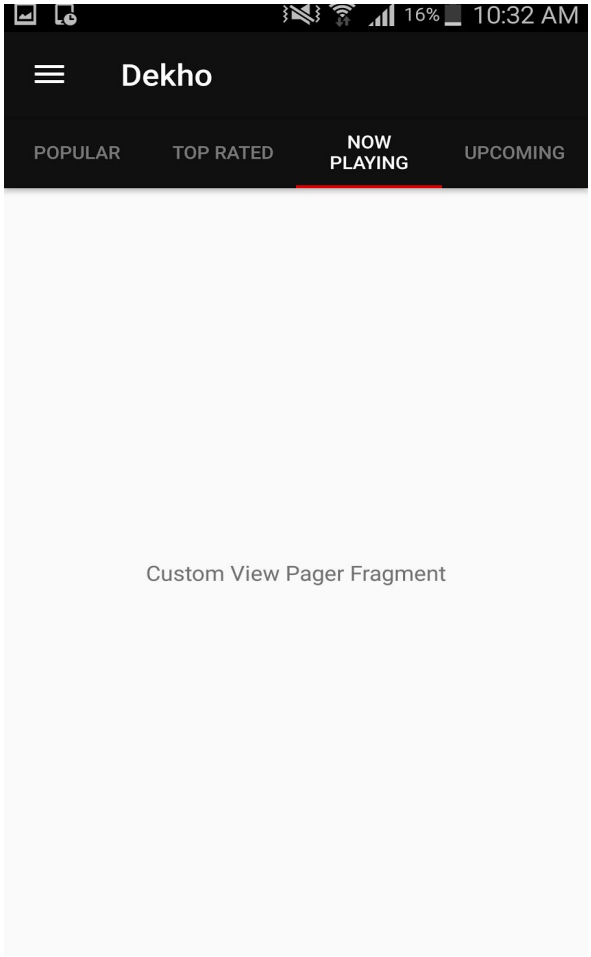
Features

Main Features -

- Explore movies by different categories.
- See the detailed information on click.
- See the trailer and information like rating.

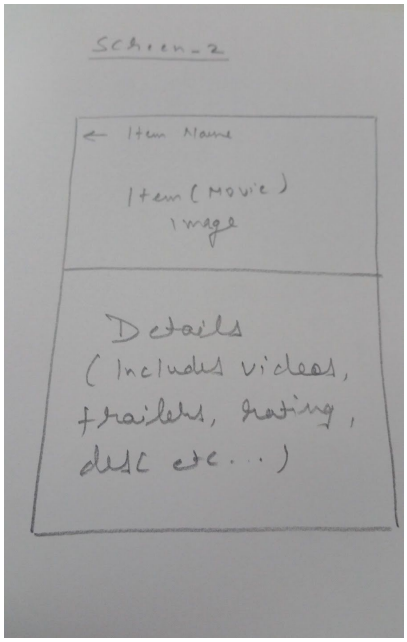
User Interface Mocks

Screen 1



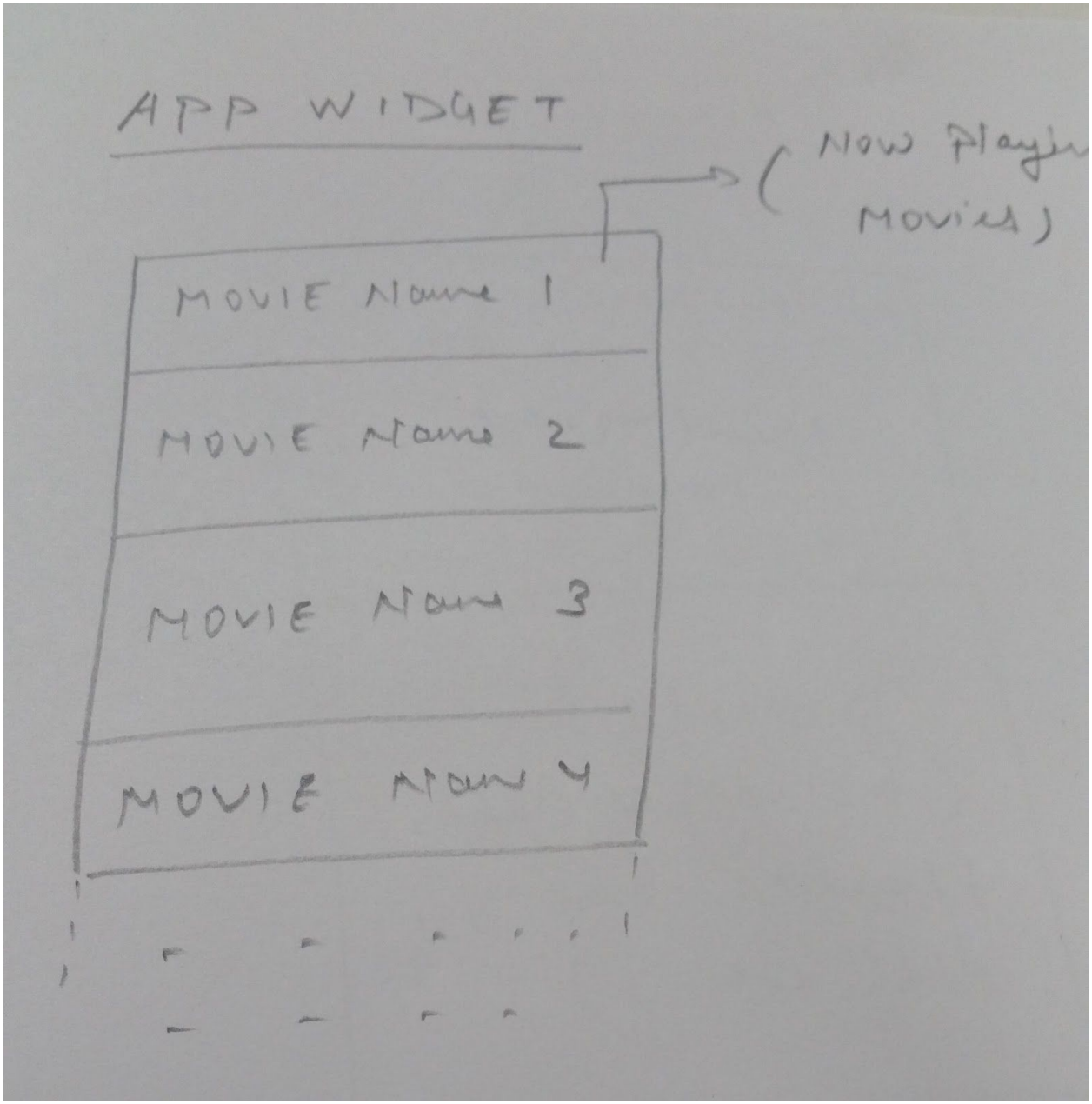
This is the different versions of the home screen with 4 different categories.For each selected category user will be able to see the details.

Screen 2



User is navigated to this screen once, he selects any item from movies (popular, top rated, nowplaying, upcoming) from home screen.
This screen holds the selected item image and it’s information based on selected item type.

App Widget



App widget, this will show the now playing movies list.

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.

I will also try to add, TV series and Celebs options in navigation drawer.

Key Considerations

How will your app handle data persistence?

App will use Firebase Real Time Database for data persistence.

App will use JobScheduler to schedule data update from web service to firebase cache.

And this firebase cached data will be displayed to the user only when he is offline (as in no internet connection).

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.
- App uses a Loader to moves it data to its views.

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**"