GitHub Username: Farhan Bakht

Movie Pocket

Description

A pocket helper for all movie’s lovers! The application contains a plenty of features which can help users to manage their personal movie collection, browse the movies for details, and all the info with thier corresponding Trailer and track the release dates of all up-coming movies. Also, Movie Pocket has a search feature which allows users to load and browse movies. It also contains give Rating Feature and Comment on a certain movie and you can also see other user’s people's opinion.

Intended User

This app is useful for all the people who wants keep track of their movies collection and get the details of Upcoming movies.

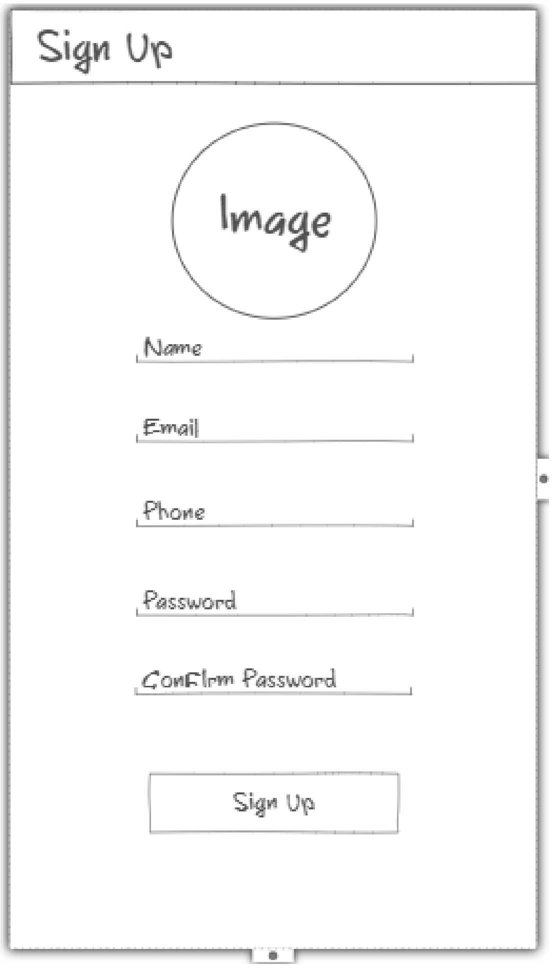
Features

List the main features of your app. For example:

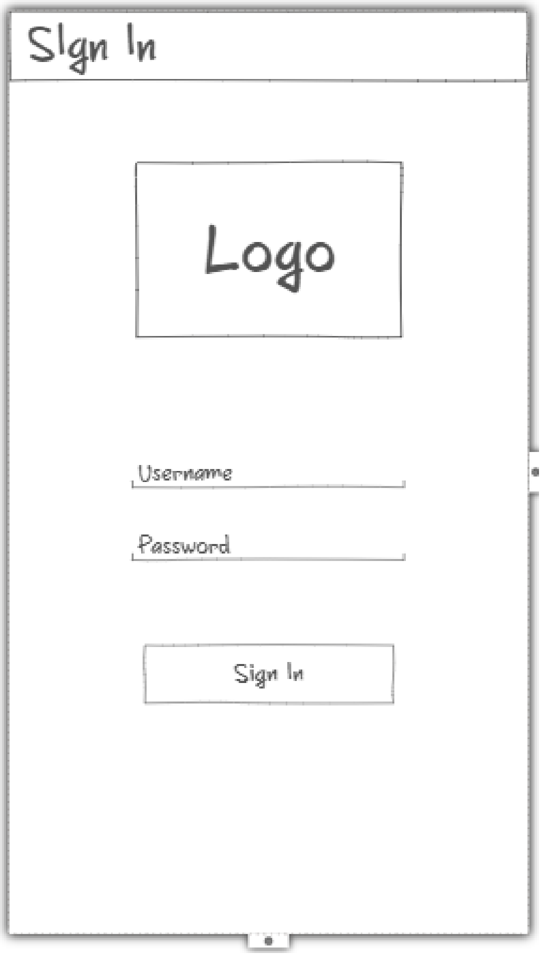
- Movie Full Details
- Your Profile
- Favorite Movies
- Movie Rating
- User Comments
- Sign in and Sign Up

User Interface Mocks

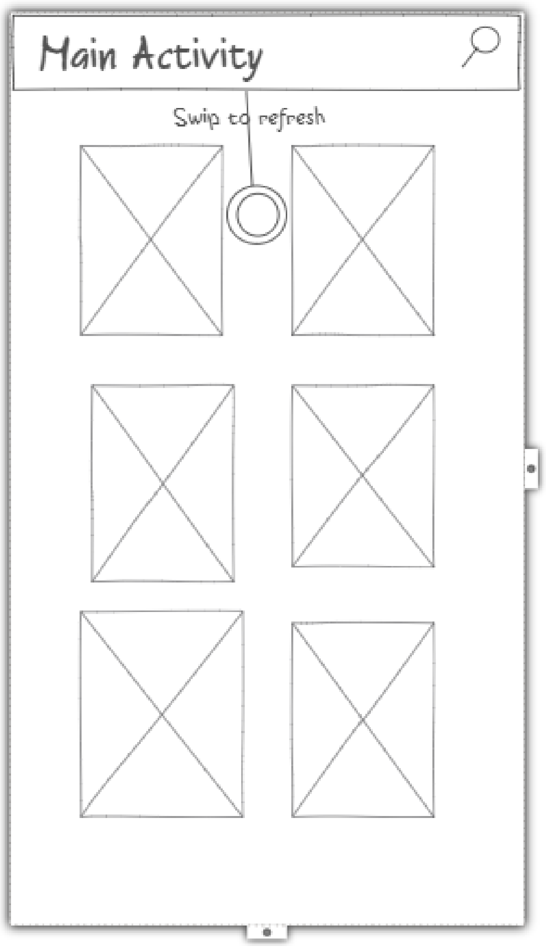
Screen 1



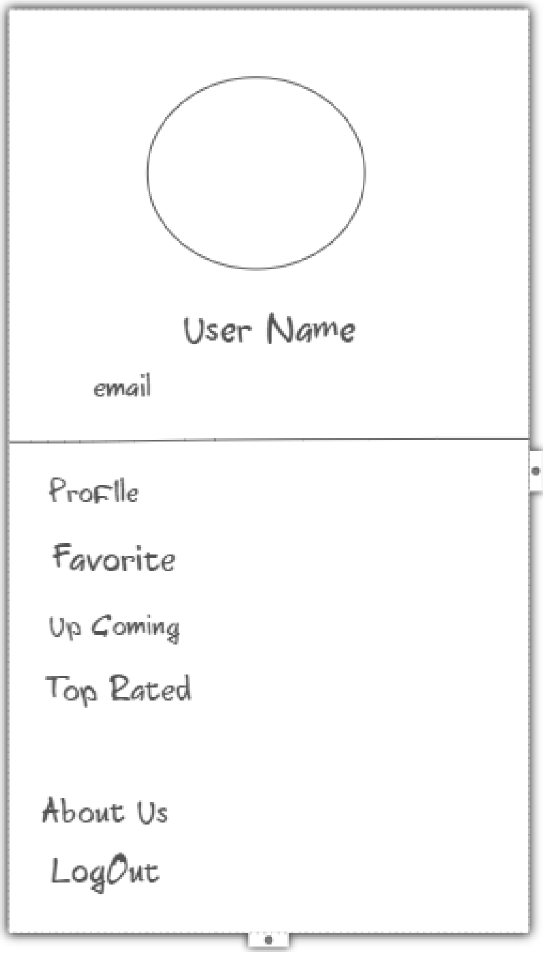
Screen 2



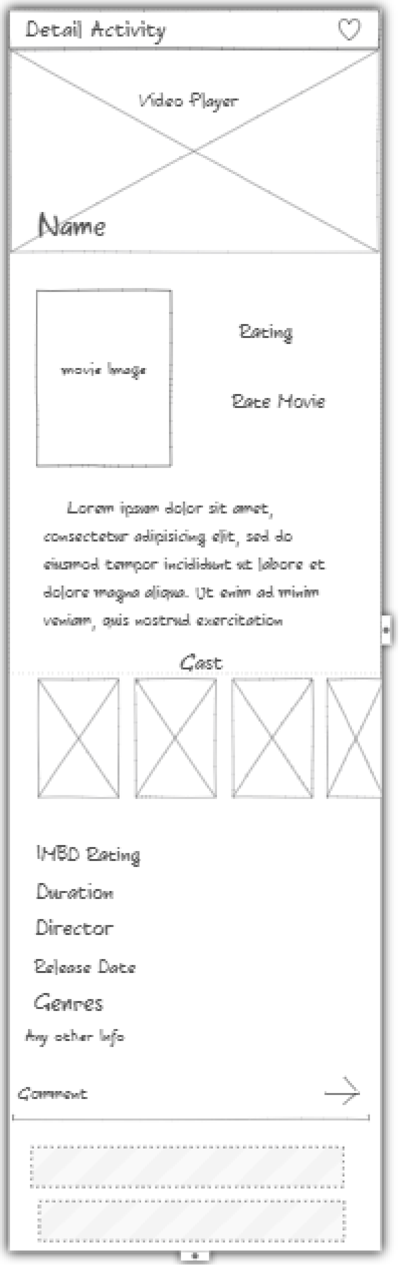
Screen 3



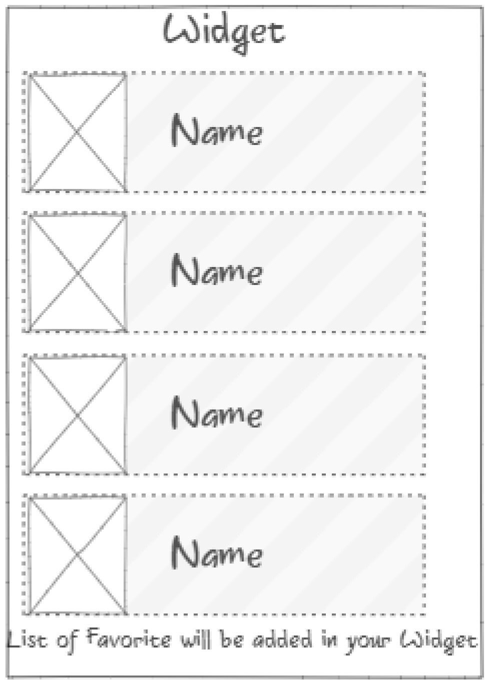
Screen 4


User Name
email
Profile
Favorite
Up Coming
Top Rated
About Us
LogOut

Screen 5


Detail Activity
Video Player
Name
movie Image
Rating
Rate Movie
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
Cast
IMBD Rating
Duration
Director
Release Date
Genres
Any other Info
Comment

Screen 6(Widget)



Key Considerations

App Development Feature's?

- App is written in the Java Programming Language
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts
- App will pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), app will use an Intent Service to do so.

How will your app handle data persistence?

App will use a Room and a Shared Preferences to maintain the local data.

Describe any edge or corner cases in the UX.

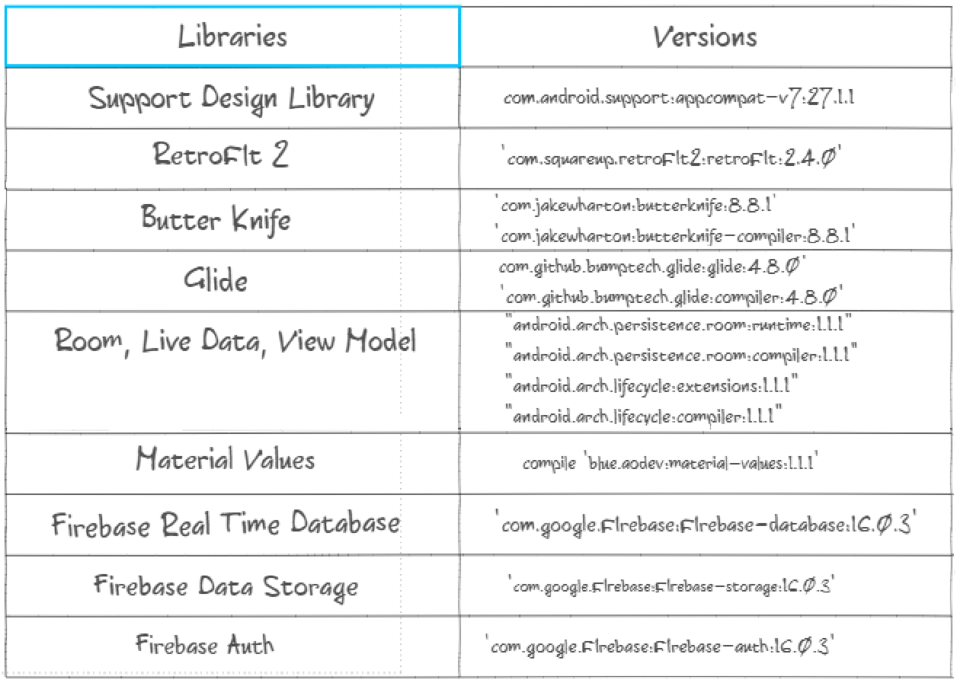
- Unstable or missed network connection: the application must not crash in that cases it will show data from database
- Device orientation change: the application must handle all long-running operations correctly considering possible configuration changes
- UI freezes: the application must not use the main thread for any resource consuming operations

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

- Firebase Crash Reporting
- Firebase Real Time Database: To Store data and show it changes real time
- Firebase Auth: To Sign in and Sign Up
- Firebase Storage: To store images

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

- Support Design Library for Material Widget's, Recycler view and Card view
- Retrofit 2: for network API requests
- Butter Knife: for boilerplate code reducing
- Glide: for images loading
- Room, Live Data, View Model
- Material Values: for handy Material Design dimensions access
- Firebase Auth: for User Login/ Sign Up
- Firebase Real Time Database: For Real time comments and rating
- Firebase Data Storage: To Store User Images



Libraries	Versions
Support Design Library	com.android.support:appcompat-v7:27.1.1
Retrofit 2	'com.squareup.retrofit2:retrofit:2.4.0'
Butter Knife	'com.jakewharton:butterknife:8.8.1' 'com.jakewharton:butterknife-compiler:8.8.1'
Glide	com.github.bumptech.glide:glide:4.8.0' 'com.github.bumptech.glide:compiler:4.8.0'
Room, Live Data, View Model	"android.arch.persistence.room:runtime:1.1.1" "android.arch.persistence.room:compiler:1.1.1" "android.arch.lifecycle:extensions:1.1.1" "android.arch.lifecycle:compiler:1.1.1"
Material Values	compile 'blue.aodev:material-values:1.1.1'
Firebase Real Time Database	'com.google.firebase:firebase-database:16.0.3'
Firebase Data Storage	'com.google.firebase:firebase-storage:16.0.3'
Firebase Auth	'com.google.firebase:firebase-auth:16.0.3'

Android Studio Version – 3.1.4

Gradle Version – 4.4

Required Tasks

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, Fragment and Item View’s

- Build UI for Sign in and Sign Up
- Build UI for Main Activity
- Call API data to fetch and show in Main Activity
- Add Drawer in main Activity
- Build UI for Detail Activity
- Add Custom video player or YouTube player
- Recycler view Item views

Task 3: Firebase Auth Sign in and Sign up

- Use Firebase Auth Integration
- Add Logics for Sign in and Sign up Activities

Task 4: Make Model’s and Recycler View Adapter’s

- Make Model Classe’s
- Make Recycler view adapter’s

Task 5: Call API and Set data and its Handling

- Add Main Activity data handling by calling API and show in Recycler view
- Handel No Network Connection
- Set data in Detail Activity
- Add video checks and its handling
- Add Rating logic
- Add Comment’s Logics

Task 6: Data persistence

- Add a Room and shared preferences helper class to handle all locally stored data.
- Crate Dao and Database Helper
- Add Favorite Logic’s and load Favorite when there is not net work

Task 7: Widget

- In Last Add Widget in App