

Description

Intended User

Features

User Interface Mocks

Screens

Key Considerations

How will your app handle data persistence?

Describe any edge or corner cases in the UX.

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

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

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Setting response classes

Task 3: Setup network requests to the api

Task 4: Setup Room database & Viewmodel

Task 5-8: Implement UI for Each Activity and Fragment

Task 9-12: Image Details Activity & Fragment UI

Task 13: implement other activities/fragments

Task 14: Implement Homescreen Widget

Task 15: Implement Google play service

Task 16: Optional: handle multi-screen support

GitHub Username: medyas

WallBay

Description

The play store is filled with images/wallpaper apps, but they mostly lack the material design, filled with bugs and do not provide enough unique images.

My idea is to create an app using Pixabay, pexels and Unsplash API. Combining these 3 APIs will provide around 3 Million royalty free photos released under Creative Commons CC0: Free

for commercial use, no attribution required. The app will allow users to search and find free images which they can be downloaded, edited or use as wallpaper.

The app will be written in Java. It will also keep all strings in a strings.xml file and enables RTL layout switching on all layouts.

The app will include support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

Intended User

The app intended for users who want to discover and find new images/wallpapers for free and observe and watch relaxing images.

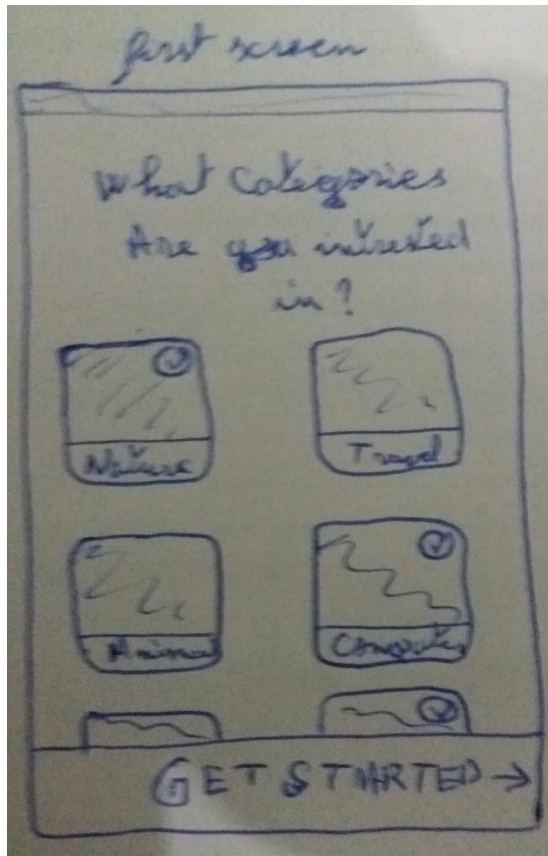
Features

The app main features are:

- Browse and search over 3 million unique images
- Display images statistic
- Add images to favorite
- Download raw images using background service (no UI blocking - IntentService)
- Multiple images select to add to favorite or download
- In app image editing
- Share images to social media or with link
- Set images as wallpaper from the app
- Optional: Images slideshow with the optional audio
- Optional: play relaxing audio

User Interface Mocks

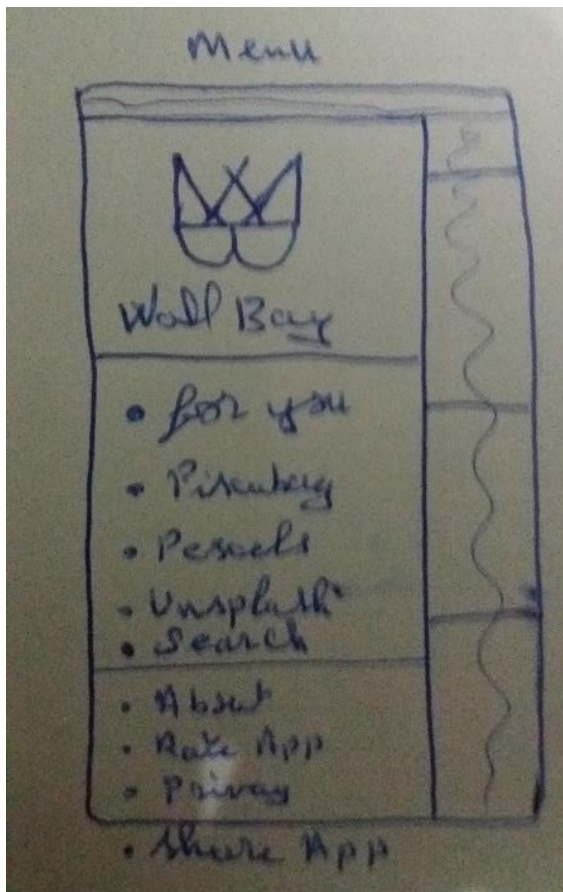
Screens



Getting started activity, only shows on first app launch. The activity allows users to select categories that they are interested in, and the data is shown in the 'For You' fragment.

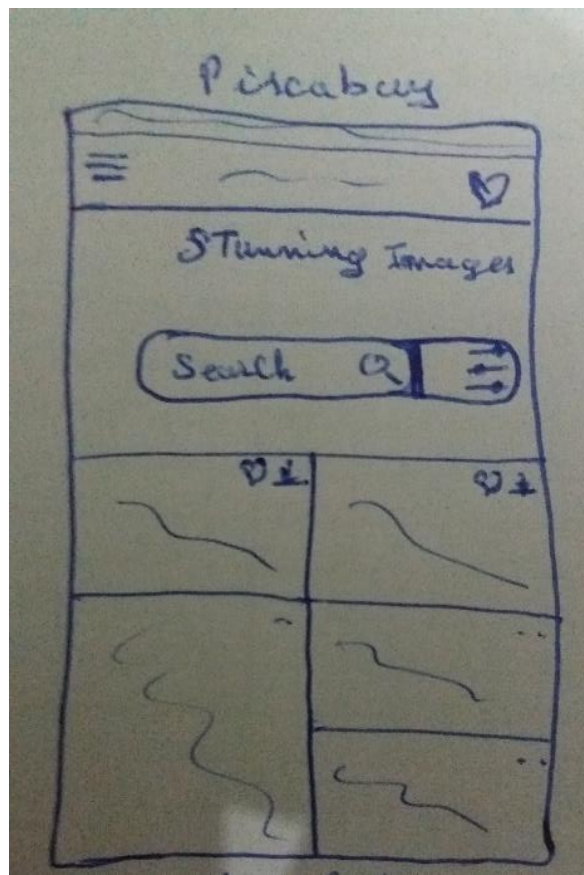


For you fragment shows the user images based on their selected categories. The toolbar has a menu and a favorite icon to navigate to the fav activity.

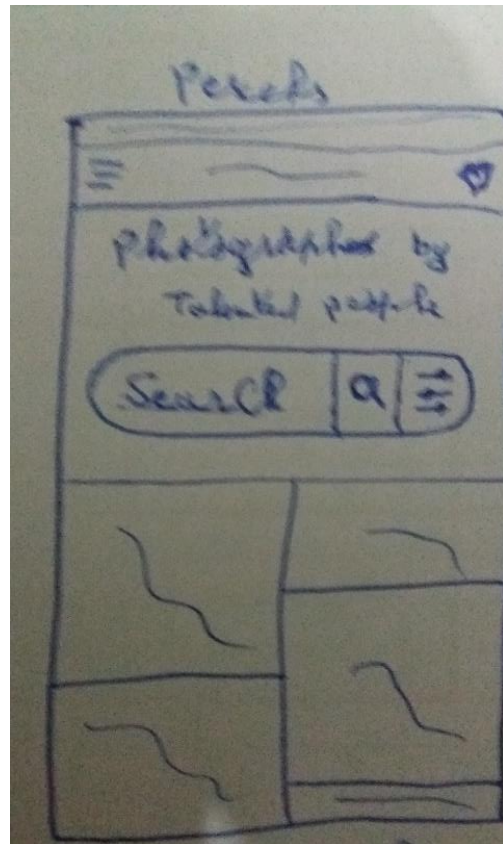


The navigation menu allows the user to navigate between fragment/activities in the app. The menu consists of 5 main fragments: For you, Pixabay, Pexels, Unsplash and search.

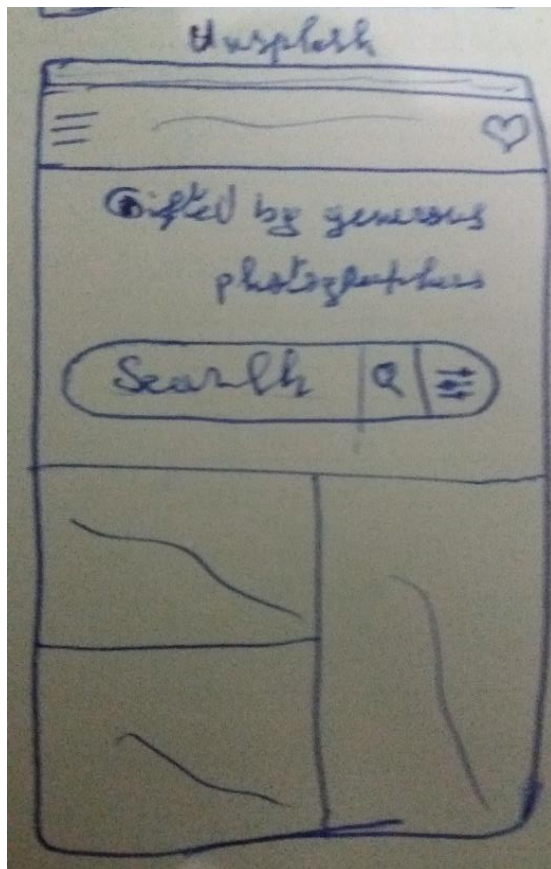
The second part of the menu is the about, privacy, rating and sharing the app.



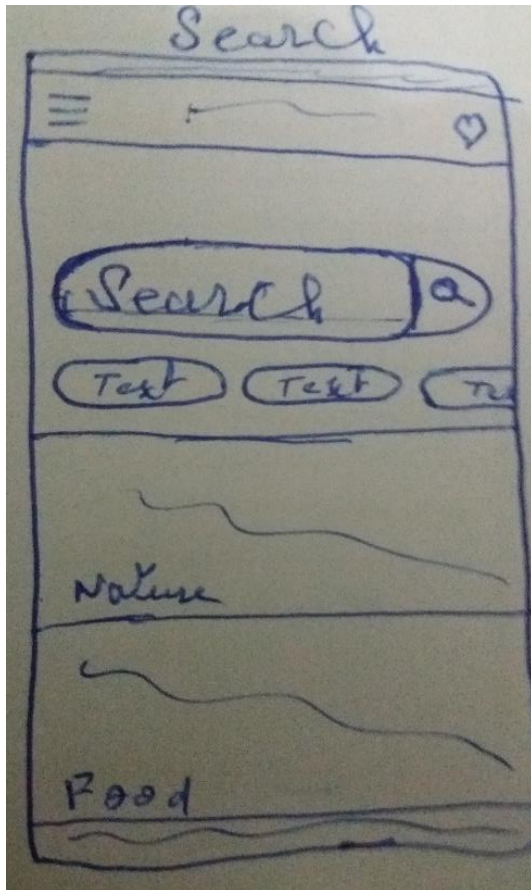
Pixabay fragment shows the most popular images by default and allows the user to search and find images in the Pixabay website.



Pexels fragment shows the most popular images by default and allows the user to search and find images in the Pexels website.



Unsplash fragment shows the most popular images by default and allows the user to search and find images in the Unsplash website.



Search fragment allows the user to find images in all 3 websites.

By default, the fragment displays a list of pre-searched terms and list of categories.

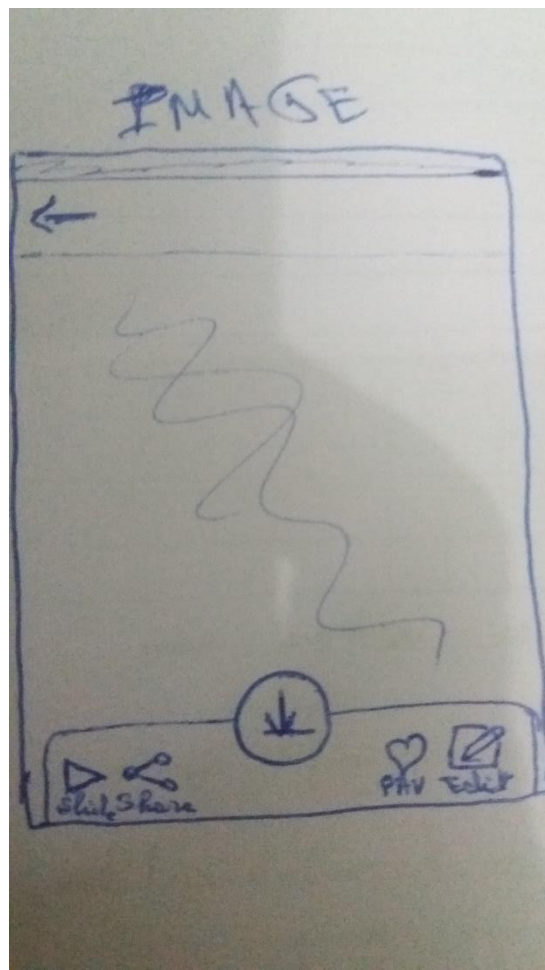
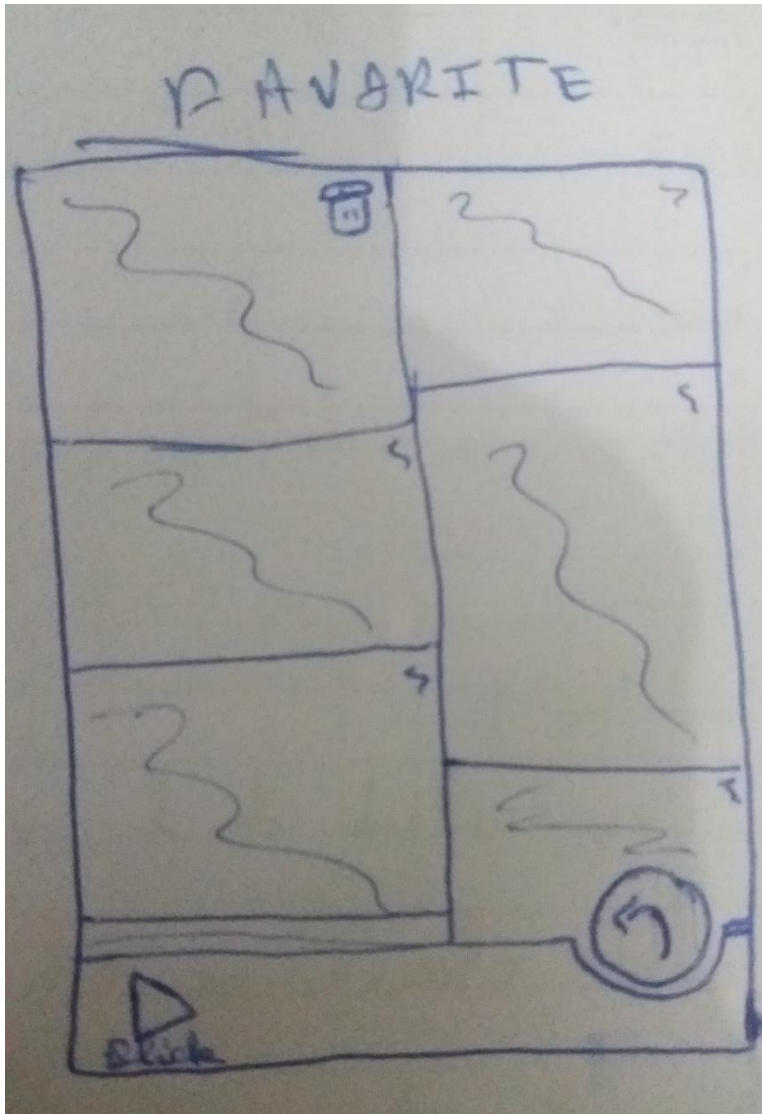
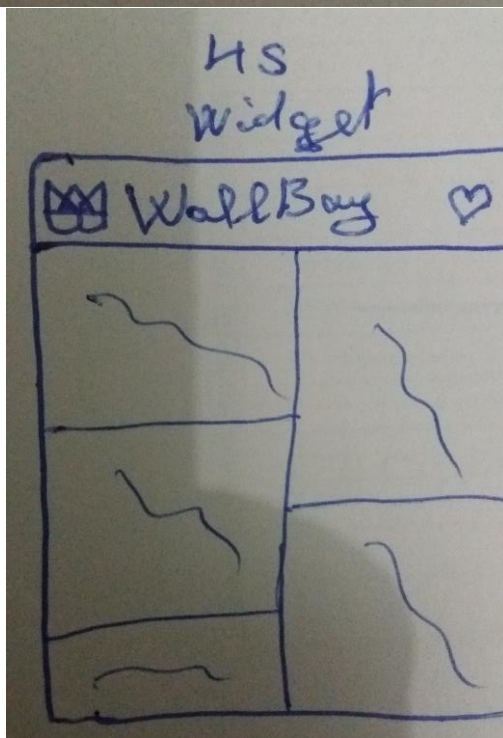


Image details activity with partially transparent toolbar with back button. A fab with download icon and bottom sheet displaying the rest of the info and icons: share, edit, add to fav, slideshow...



Favorite activity displaying the list of images added by the user to favorite and stored in the DB. The user will be able to press back and start an images slideshow, also they can delete images from the list.



Home Screen widget to display user favorite images. The widget will contain a listview of images, and onclick the images details activity will start.

Key Considerations

How will your app handle data persistence?

I will be using API endpoints, and still thinking about caching the response using room for offline availability but this is still an optional thing.

For the favorite images, Room will be used to save and retrieve the data.

Describe any edge or corner cases in the UX.

The app will comply with the Android and Udacity guidelines. Mostly the app will be in a single activity so pressing back will result in exiting the app. If the user navigates to another activity or a sub fragment, the back button will be shown and allows the user to navigate up as provided by the guidelines.

As I will be using Retrofit to fetch data from the api endpoints, I don't need to implement JobDispatcher, IntentService or AsyncTask. But the IntentService will be used in downloading the images from the download url using the DownloadManager.

Also, the app will use Room with ViewModels and LiveData.

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

Android Support library V28

Gradle: 4.6

Constraint-layout:1.1.3

Lifecycle: ViewModel and LiveData: 1.1.1

Room: 1.1.1

Glide: image download and cache V4.8

PhotoView: imageView with zoom in and out V2.1.4 - <https://github.com/chrisbanes/PhotoView>

PhotoEditor: images editing library V0.3.3 - <https://github.com/burhanrashid52/PhotoEditor>

Retrofit: fetching the data from API endpoints V2.5

Lottie: displaying adobe animation V2.7 - <https://github.com/airbnb/lottie-android>

Won't be using Butterknife as I want to try Android Data Binding

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

The app will have ads and will implement the admob service, also firebase analytics will be implemented for analytics and understanding user's needs.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Create a new android project, and start implementing the required dependencies, adding endpoints auth keys and setting git.

Task 2: Setting response classes

Creating java classes that matches the json response from endpoints. There are 3 endpoints, plus there will be a common class for all endpoints to be used in favorite (find matching elements to save in this class)

Task 3: Setup network requests to the api

Setting up communication to the api endpoints

- Creating the requests with retrofit
- Settings Repo and Viewmodal

Task 4: Setup Room database & Viewmodal

Setting up Favourites DB with its Viewmodal

Optional: setup caching db with room and viewmodal to cache api response – allow app to work offline.

Task 5-8: Implement UI for Each Activity and Fragment

Create First Screen UI & and caching user pref

Create fragments UI and communication with viewmodal (livedata) for :

- For You
- Pixabay
- Pexels

- Unsplash
- Search

MainActivity :

- Creating the menu
- Setting the navigation between fragments

Task 9-12: Image Details Activity & Fragment UI

Create the activity

Create ImageDetailsFragment UI & Setup:

- DownloadManager
- Add to favorite
- Image Palette: show the images color palette
- Edit, share and set as wallpaper
- Slideshow and audio – optional

Add animation

Task 13: implement other activities/fragments

About, Rate app, Privacy and share app.

Task 14: Implement Homescreen Widget

Create the ui and appwidget class to handle the widget.

Task 15: Implement Google play service

Implement admob and firebase analytics.

Task 16: Optional: handle multi-screen support

Add tablet layout for each fragment/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”**