CAPSTONE PROJECT REQUIREMENTS

GitHub Username: *saiswaroopallurl*

CAPSTONE PROJECT REQUIREMENTS

Favorite Photos

Description

This app lets the users to search for photos of a particular region by dropping pins on the map. Users can mark their favorite photos of that region and also enables them to share the link or photo to their freinds or family.

Intended User

This app is for everyone who wants to search for beautiful pictures of a particular place and share. The app is retricted to **safe search only.**

CAPSTONE PROJECT REQUIREMENTS

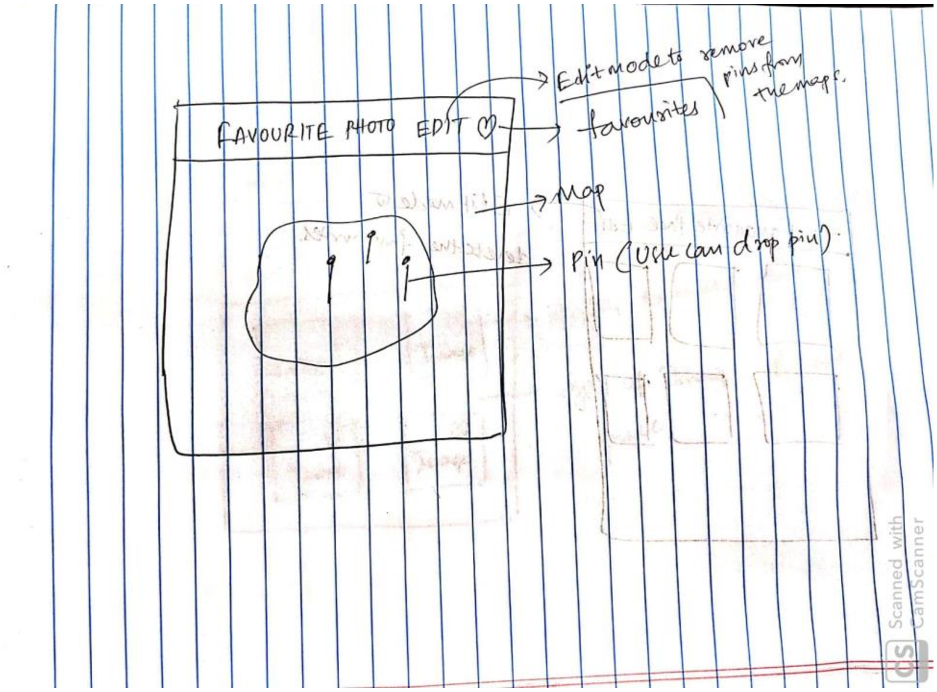
Features

This app has features like:

- Ability to search for beautiful features by dropping pins on the map.
- Ability to mark them as favorites.
- Ability to share the link or photos.

User Interface Mocks

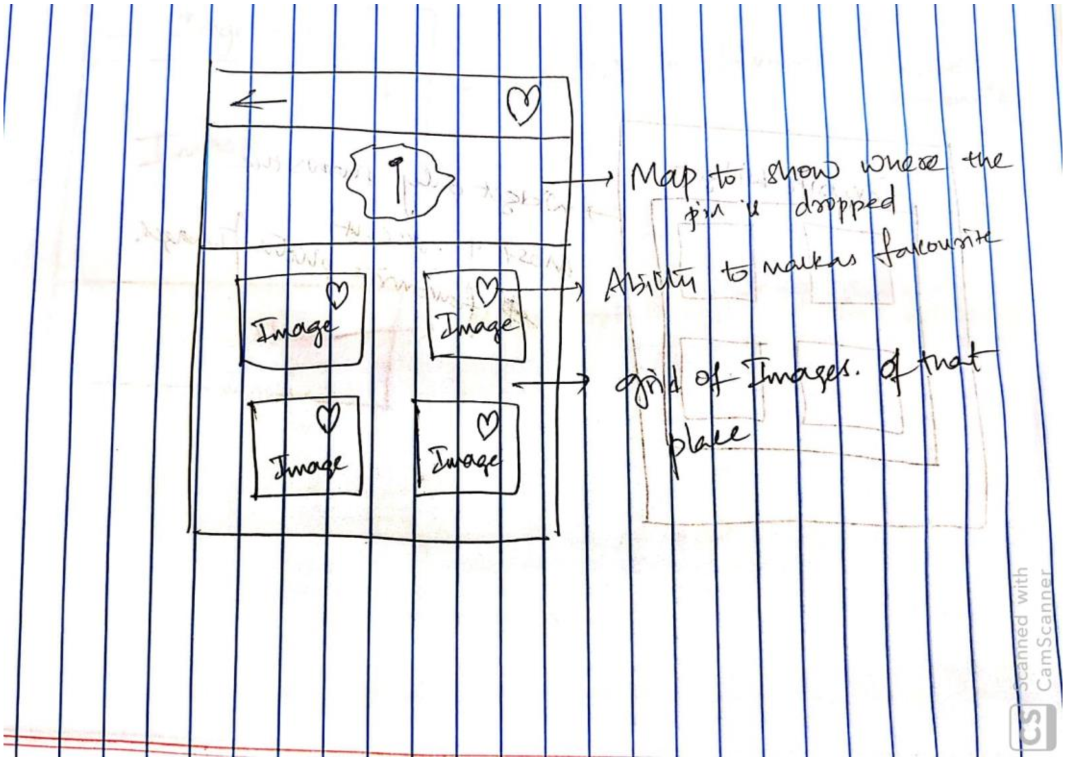
Main Screen - Main Activity (Map)



This is the main screen with the map where the user has the ability to drop pins on the map to retrieve photos of that area. It also has the edit mode to remove pins on the map if the user wants to do that. When the user clicks on any of the pin it takes them to next screen. It also has favorites button at the top right corner where the user can track of his favorite images.

CAPSTONE PROJECT REQUIREMENTS

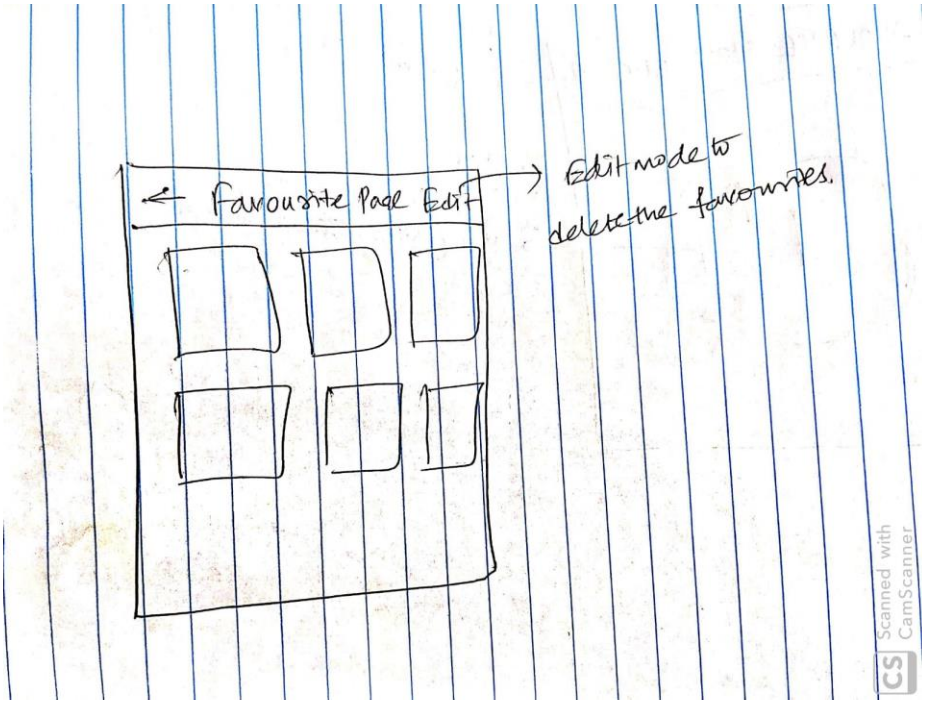
Place Images Screen:


Map to show where the pin is dropped
Ability to mark as favourite
grid of Images. of that place
Image
Image
Image
Image

So, when the user clicks on the pin the user lands on this page. The top portion just shows the where the user has dropped the pin on the map and the bottom it shows the user a list of photos of that place. Each photo has the ability to be marked as favorite.

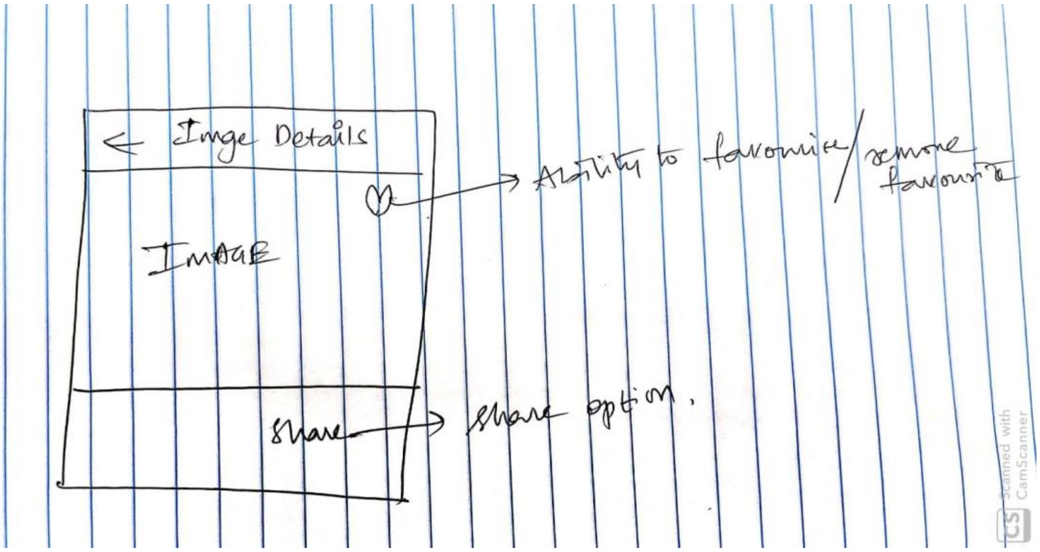
CAPSTONE PROJECT REQUIREMENTS

Favorites Images Screen:



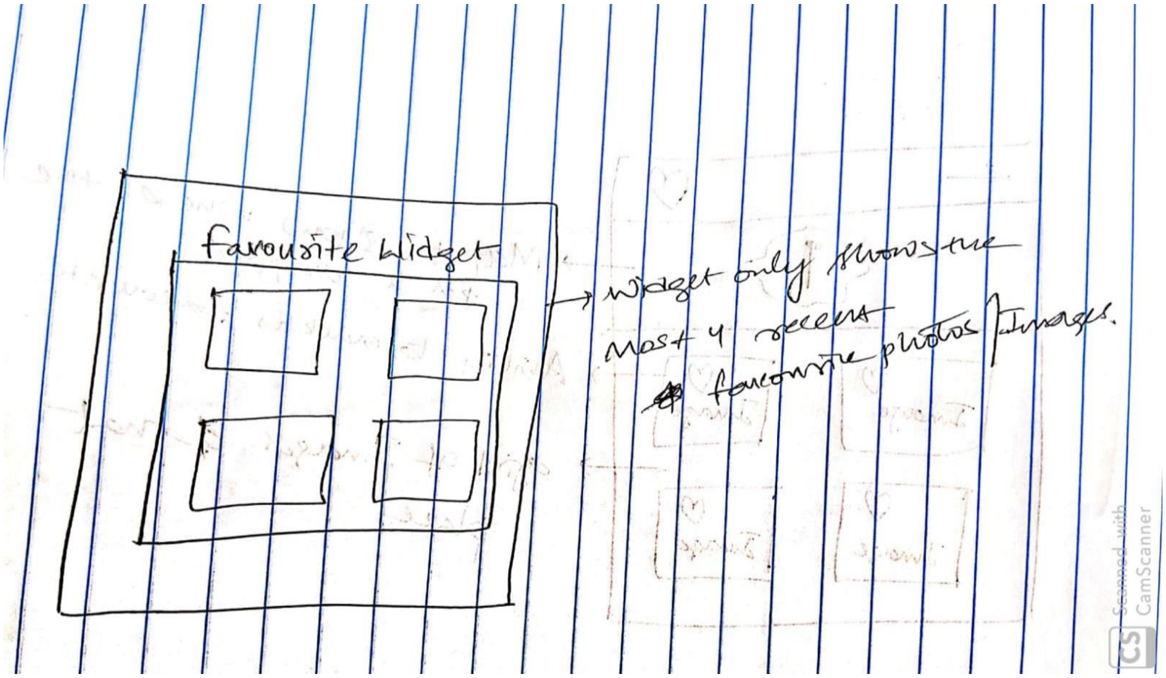
In this page the user can see all his favorite pictures and has the ability to remove from his favorites.

Detail Image Screen:



When the user clicks on any photo the user lands on the detailed page. The user has the ability to share the photo. An **Ad-mob** is displayed at the bottom of the page on this screen.

Widget Screen:


favourite widget
widget only shows the most 4 recent favourite photos / images.
Scanned with CamScanner

When a user selects a photo as favorite, the most recently selected four favorite photos will be shown on the widget.

Key Considerations

How will your app handle data persistence?

This app uses a content provider with SQLite Database to store the necessary information of the photo which are marked as favorite like the image URL, image id, etc. Also the pins data: lat and long where the users have dropped it on the map.

Describe any edge or corner cases in the UX.

When there is no network connection the app will show the already downloaded content which is basically cached. If no photo is downloaded by the time the network gets disconnected then listing will be shown with placeholder images.

CAPSTONE PROJECT REQUIREMENTS

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

1. Picasso: For downloading and caching the images.
    a. com.squareup.picasso:picasso:2.71828

2. Butterknife: For Binding android views inside the code.
    a. com.jakewharton:butterknife:8.8.1

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

1. Admob: For displaying ads on the detail screen.
    a. com.google.android.gms:play-services-ads:15.0.0
2. Google Map: For displaying the map and having the users to drop pins on the map.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new project.
- Add necessary libraries inside build.gradle file and make sure everything is in sync and no errors are present.
- This project uses **"Java"** as the programming language.
- Uses **Android Studio 3.1.4**
- Uses **gradle version: 4.4+** in conjunction with **android plugin: 3.1.0+**

Task 2: Resource Managment

- All Images necessary for the app will be placed accrodingly in drawable folders (dpi,xhdpi etc)
- All the colors will be placed unders **colors.xml (**Color codes not yet decided for the current app**)** of **values folder.**
- The custom theme will be placed in **styles.xml** of **values folder.**

Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity: which basically contains a map
- Build UI for PhotosActivity: which basically contains a list photos of a dropped pin.
- Build UI for FavoritesActivity: which basically contains a list of photos that are marked as favorites.
- Build UI for DetailActivity: which contains single enlarged image to view and share.
    - Main acivity will be having a map fragment which can be used for phone and tablet.

    - Photos fragment can be useful for displaying the photos of a dropped pin or favorite images.

Task 4: Persistance Store

- Build a Db Helper class for creating the tables: Pin Table, Favorites Table.
- Build a contract class.
- Build a content provider for accessing the pins and favorites.
- Will be using the Loader for content provider.

Task 5: Implement Network Task.

- Will be using AsyncTaskLoader for making network calls.

Task 6: Implement Miscellaneous.

- Implement **RTL** (Right-to-left) layout switch and test everything works accordingly and all the strings are stored in **strings.xml**
- Implement Dpad so that user can control on the screen.

Task 7: Implement Google Play Services.

- Implement Ad-mob service

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**"