How to Use this Template
1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

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**"

GitHub Username: NeerajMoudgil

GIF ZONE

Description

App allows users to see, download and share various gif images. It allows to search gifs and share gifs on social/messaging apps. Users can express emotions using different small gifs.

Intended User

Everyone who love expressing emotions using images/gifs or stickers. Gifs can be shared over social apps.
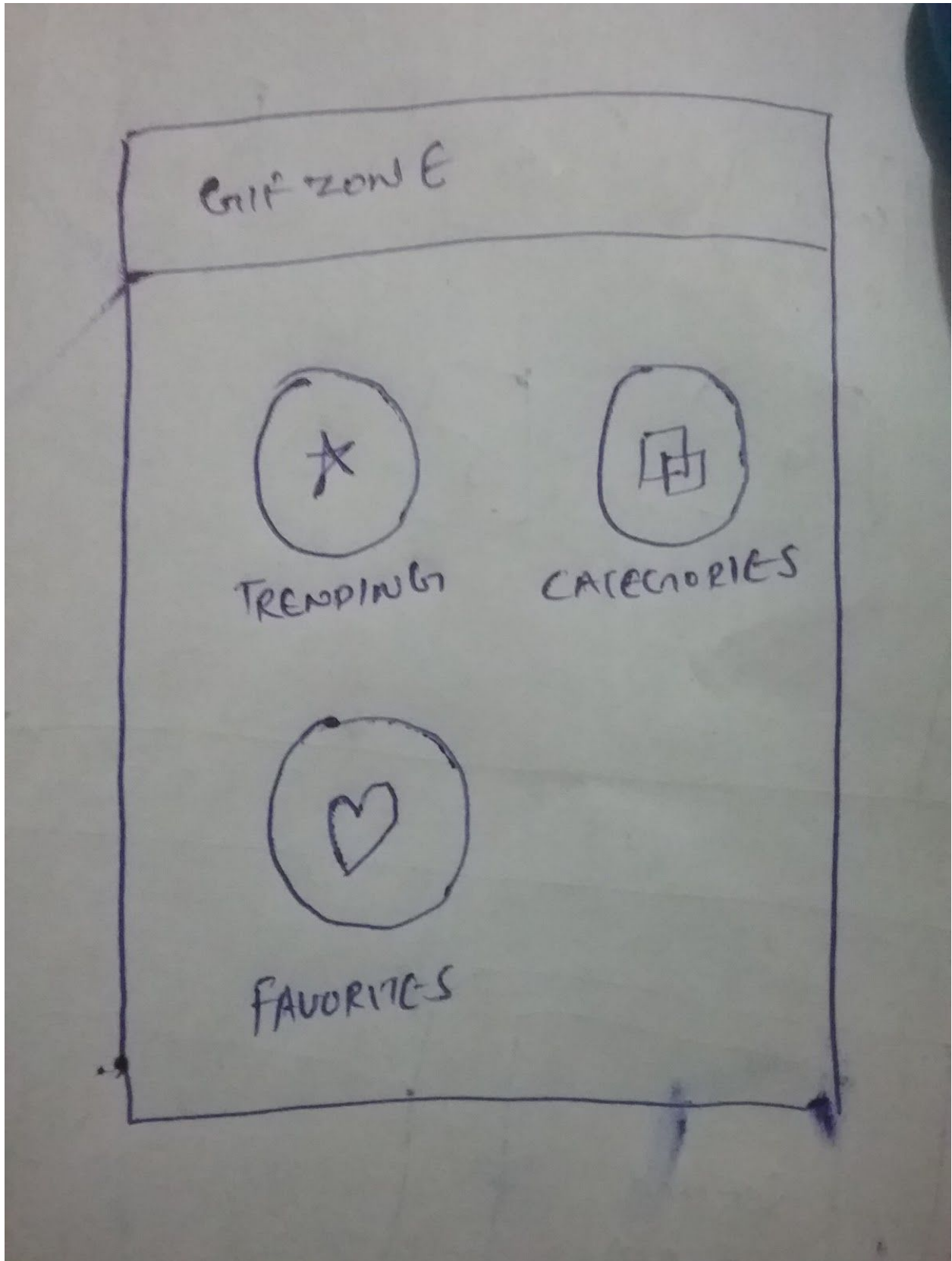
Features

- Show trending Gifs
- Download and share Gifs
- Save favorites gifs in local DB

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

This is the Home Screen where we have buttons to link to other screens. Trending Will navigate to the screen showing trending gifs, similarly categories will show famous categories and favorites will navigate to user favorited images.
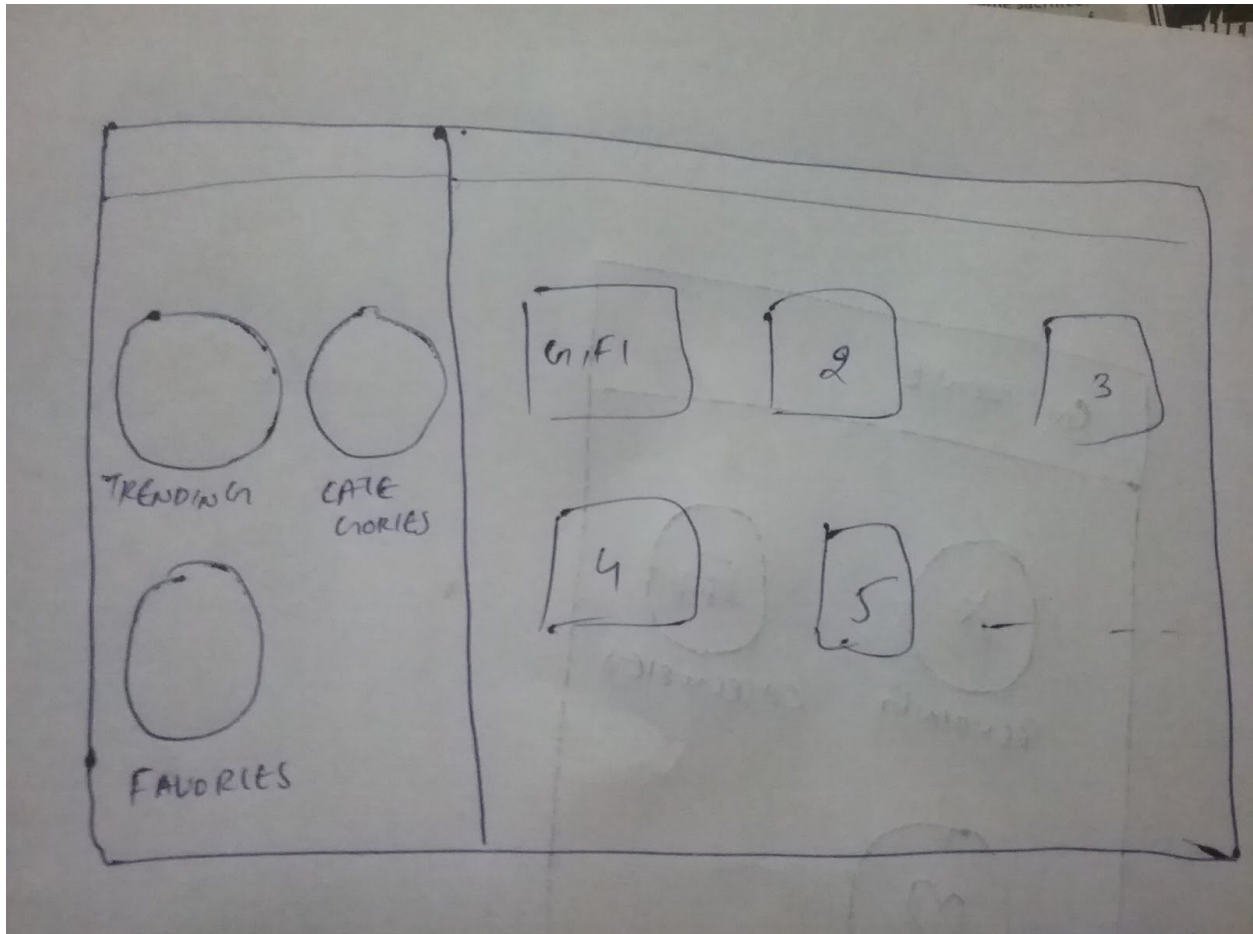
Screen 2
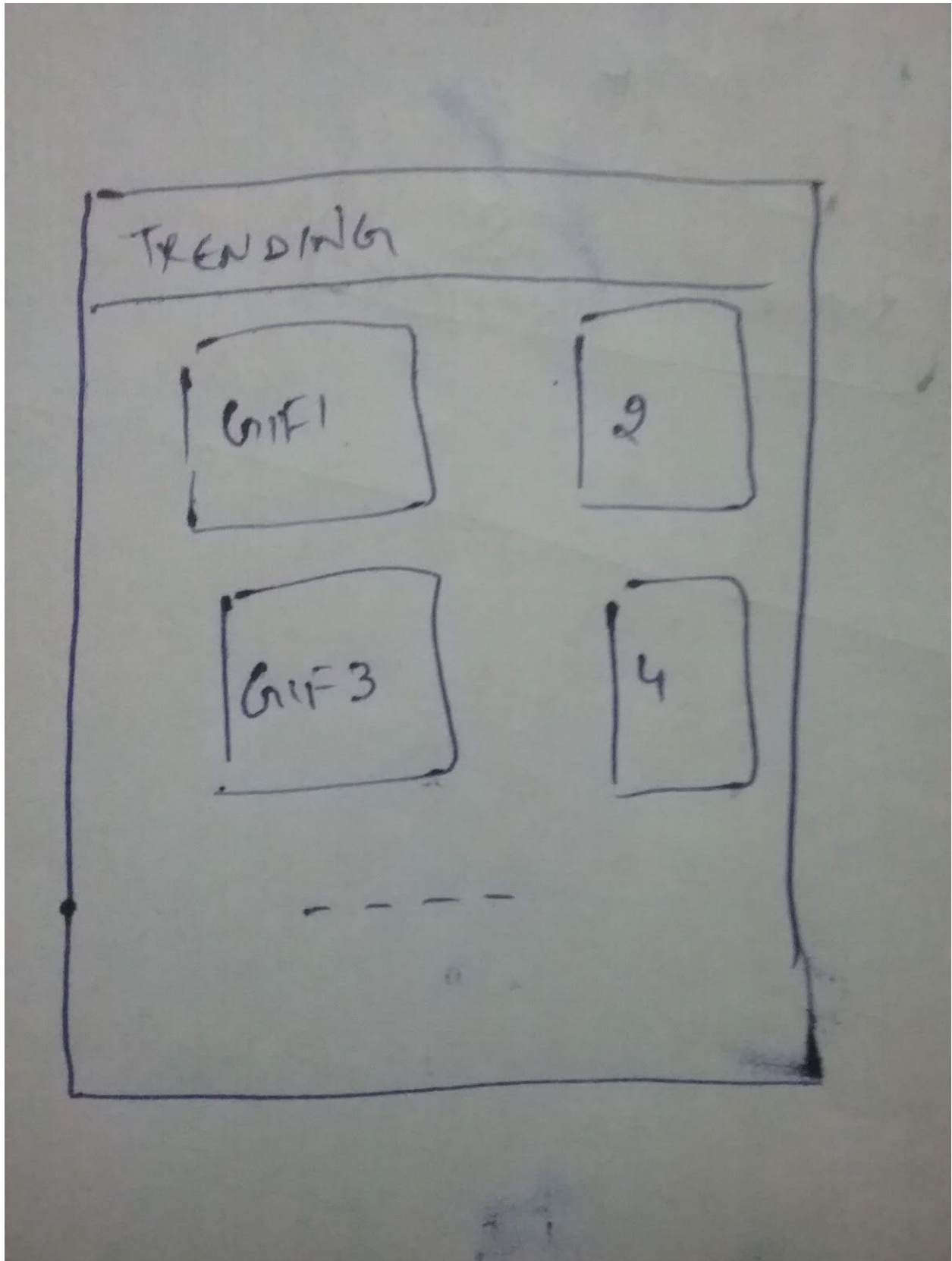
When user clicks on Trending icon in home screen, this screen comes with trending images. Images will be populated using GridView in recyclerView.

Screen 3

DETAIL

GIFI
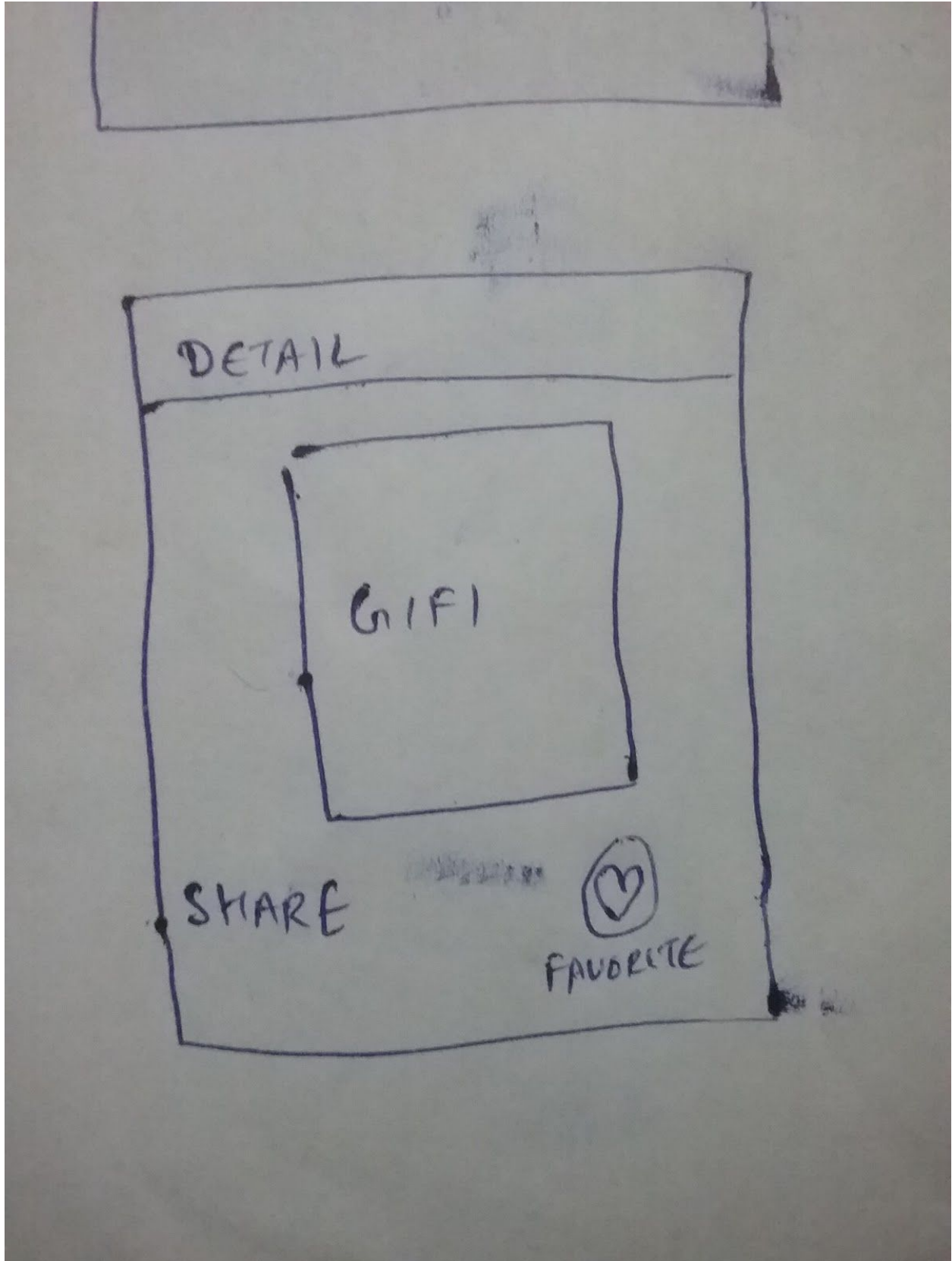
SHARE

FAVORITE

Detail Page of image is opened when any image is clicked in previous screen with options to share on different social apps and adding to favorites.

Screen 4

Tab Layout - Home fragment will be at left and corresponding detail fragment screen will be shown on the right based on the selection done.

Key Considerations

How will your app handle data persistence?

App will only save the gifs marked as favorites in Content Provider or downloaded gifs in Gallery o user's mobile.

Describe any corner cases in the UX.

SavedInstanceState or LoaderManager Callbacks will be used to prevent activity reload data on activity recreate when rotated or back pressed.

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

Volley will be used to make calls to Giphy API to get Gifs
Glide will be used to load Gif images into ImageView
ButterKnife will be used for View Bindings

Describe how you will implement Google Play Services.

Admob for showing ads..
Analytics for app analytics.

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

- Creating project in Android Studio.
- Target SDK.
- Adding dependencies to Gradle for libraries usage

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Fragments
- Build UI for detail Image of Selected Gif

- Build UI for Tablet Screen
- Build UI for Widget

Task 3: Implement Google Play Services

Implementing Analytics and Admob.

Task 4: Implementing Giphy API

- API key for API
- Calling endpoint using Volley and retrieving response

Task 5: Implementing Widget

- Create Widget Provider.
- Create Service.

Task 6: Handling Errors

- Handling network errors..
- Empty Views
- Progress Loaders.

Task 7: Content Providers

- Implementing Sqlite Tables/local data using content providers.

Task 8: RecyclerView

- Most of the views will be using REcyclerview to show number of images.
- Glide will be used to load images.

Task 9: Sharing Images

- Share Gifs to different social apps.

Task 10: Shared Transitions

- Implementing Shared transitions to detail page of image.

Task 11: Downloading Images

- Downloading Images to local memory.

Task 10: Creating Build Variants

- Creating release and debug variants.
- Creating signed release apk.

Add as many tasks as you need to complete your app.

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**"