**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**: kienme

# ReacT

## Description

Looking for the perfect reaction gif? Look no further! Just share the text you received with ReacT and it will suggest the best gif for you.

# Intended User

This app is intended for anyone who wishes to express their reaction through gifs rather than text and emoji. Target audience includes students and meme browsers alike.

# Features

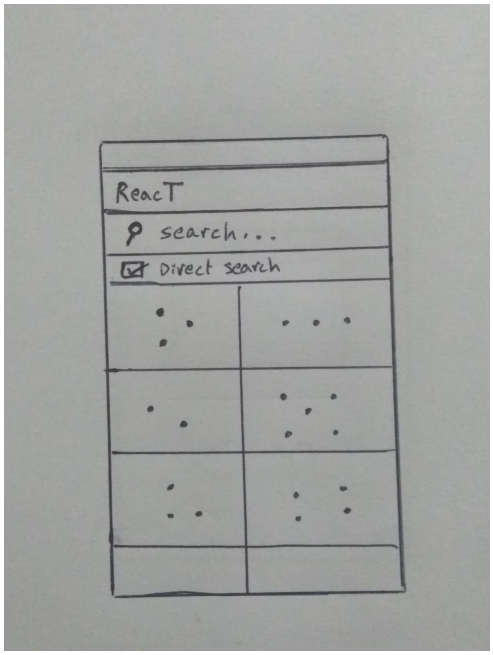
These are the main features

- Show suggested gifs
- Gifs based on keywords and emotion
- Share from the app

# User Interface Mocks

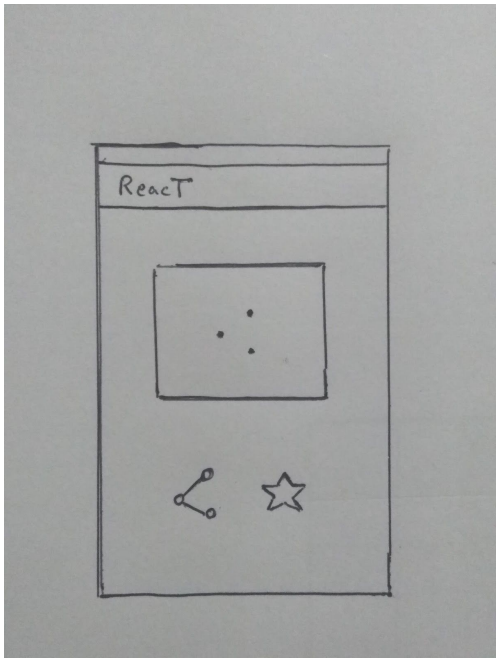
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

### Screen 1



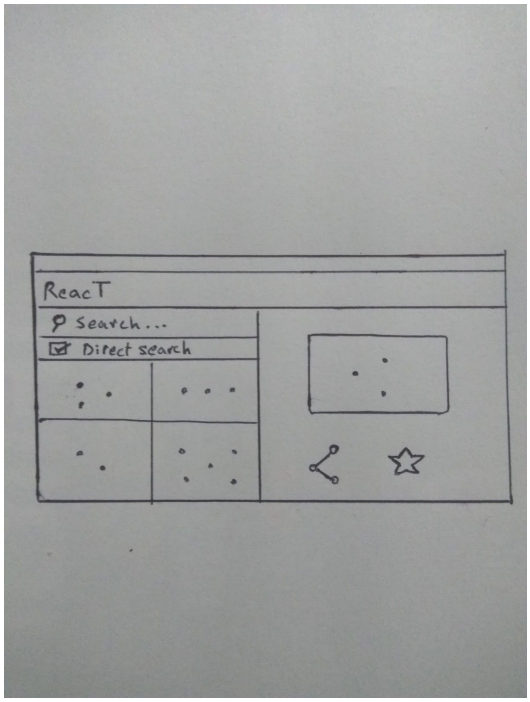

This will be the main activity for phone. It contains a tool bar and a search bar. Hitting search brings up the result in the grid view below.

## Screen 2



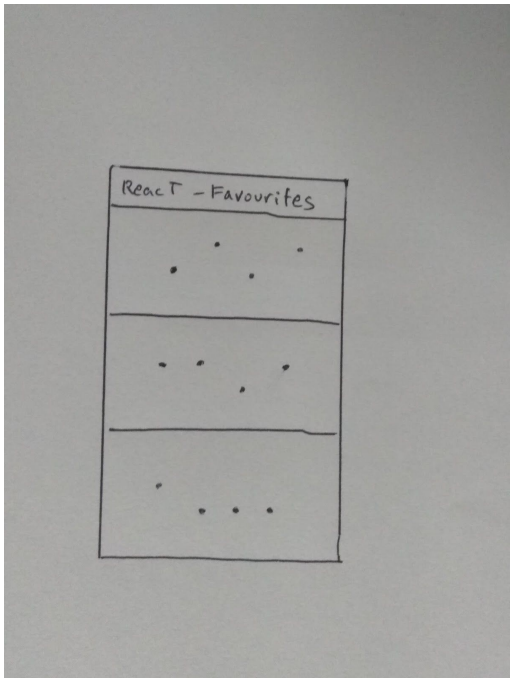

Clicking on an image brings up this activity where the image is in focus and options are provided to share and favourite the image.

## Screen 3




The tablet layout includes the above two fragments side by side.

### Screen 4




The widget will show a list of favourite gifs.

## Key Considerations

### How will your app handle data persistence?

The app will use a local SQL database for storing favourite gifs. This will be implemented using a ContentProvider. A CursorLoader will be used to load the data to the views.

### Describe any corner cases in the UX.

When the user directly types in the search bar, it performs a lookup verbatim. When text is shared to the app via an intent, it analyzes it for emotion/keywords and displays the result.

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

Synesketch for emotion detection, Glide to handle image loading and Giphy API for obtaining gifs.

**Describe how you will implement Google Play Services.**

The following Play Services will be used

1. AdMob for displaying ads
2. Analytics for… well, 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

The following tasks are involved in setting up

- Configure libraries
- Create activities and fragments

### Task 2: Implement UI for Each Activity and Fragment

Subtasks for building UI

- Build UI for MainActivity
- Build UI for DetailFragment

### Task 3: Emotion Recognition

When input text is obtained, it is required to check its dominant emotion. This task involves building a module for simplifying this using the Synesketch library.

### Task 4: Querying with the API

Once the emotion has been detected it must be passed to the API to return results. In this task the result is verified for the given input text. The calls will be made using an IntentService.

## Task 5: Updating UI

The main GridView is populated with the results using Glide. Clicking on an image allows users to share it or star for future use. For favourites, a CursorLoader loads the data from the ContentProvider.

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**"