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: devikamehra

Memenized

Description

Create your customized Meme through this Meme Creator app. This app will assist you in trolling your friends and family. This application allows you to choose from 1000+ meme, add the caption in various fonts and share with everyone. The app has a simple interface, professional material design and lots of fun. The memes will be created using a 3rd party library.

Intended User

The app is intended for the people of all age groups who wish to troll their friends and family.

Features

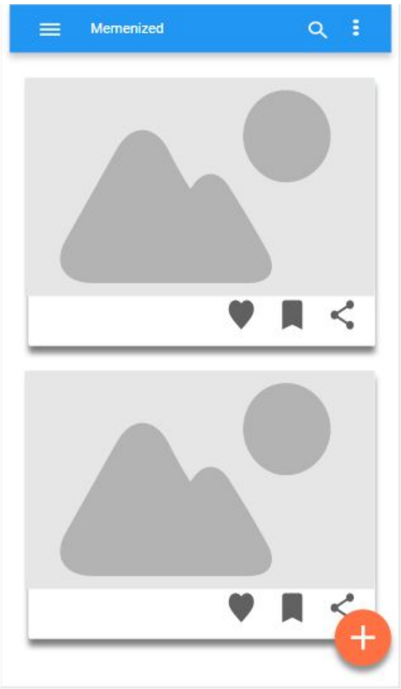
List the main features of my app.

- Choose from 1000+ meme
- Use as many as 125+ fonts
- Widget

User Interface Mocks

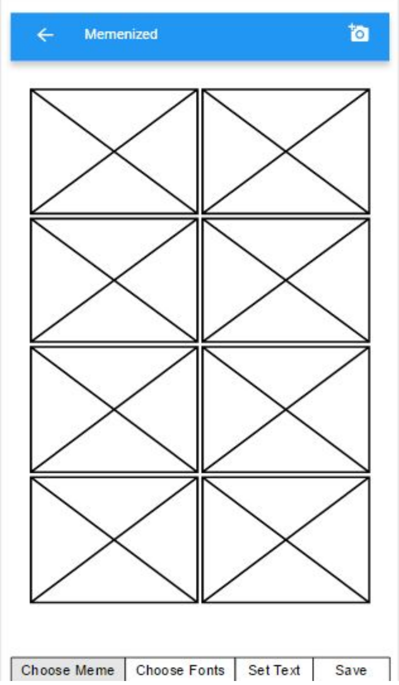
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



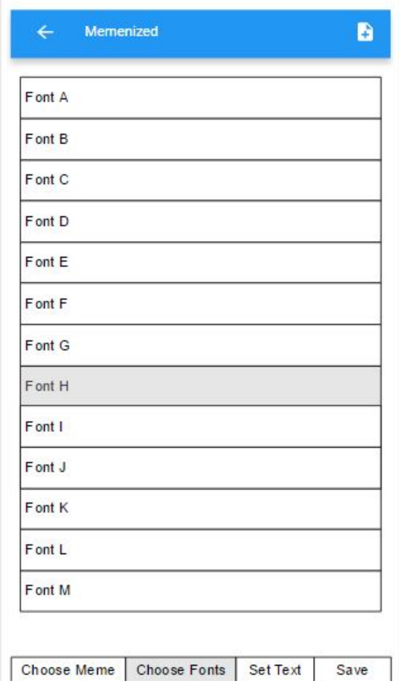
1. This screen displays the Memes user has already created.
2. The floating action button will allow us to create a new Meme.

Screen 2



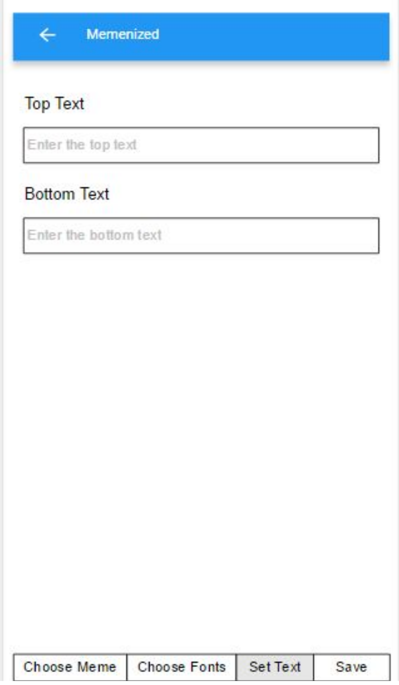
This screen displays all the available Memes.

Screen 3



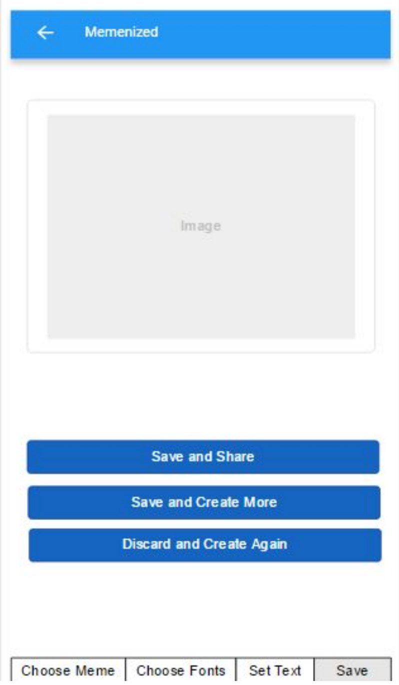
This screen displays all the available Fonts.

Screen 4



This screen will allow the user to add top and bottom text for the Meme.

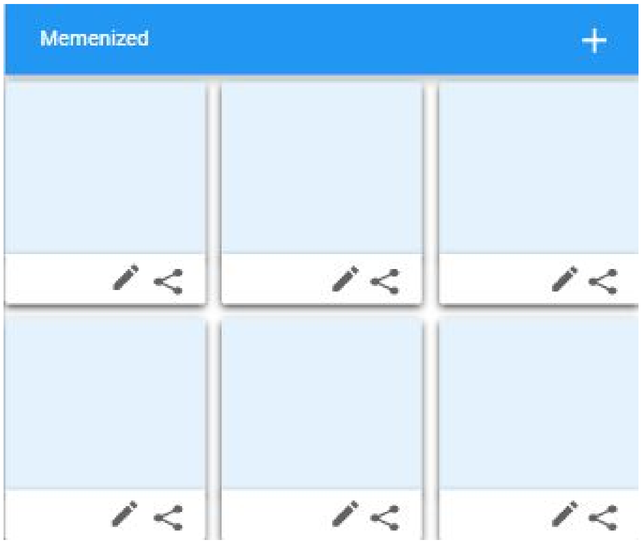
Screen 5



This screen will be the final screen and would send the api call and receive the final Meme. Three options are provided

1. **Save and Share:** The meme would be saved in a folder called Memenized. This folder will be created when first image is to be saved. To share Meme, share intent will be used.
2. **Save and Create More:** The meme would be saved in a folder called Memenized. This folder will be created when first image is to be saved. To create more meme, screen 2 will be displayed.
3. **Discard and create again:** The meme made so far will be discarded and wouldn't find it's space in Memenized folder. The app will lead to screen 2 to lead user design the meme again.

Screen 6



This screen shows the view of the widget. It will show the memes already created. New one can be created from the 'Add' icon.

Key Considerations

How will your app handle data persistence?

The app will store its data with the help of a new Mobile Database build using firebase. All the base meme images and the meme created will be stored using Firebase Storage.

Describe any corner cases in the UX.

While creating the meme, if the user presses the back button, a dialog will pop up to caution the user about the un-saved work and only if user agrees to discard it, the user will return to screen1.

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

1. Picasso to handle the loading and caching of images
2. Retrofit for network calls
3. Firebase for maintaining database
4. Design Library

Describe how you will implement Google Play Services.

Google Play Services that will be used are

1. **Admob:** This will be used to add advertisement. One small banner on screen 1 and one larger banner after every 3 Meme generation.
2. **Firebase:** To maintain all the necessary data.

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

- Update Android Studio, SDK, etc
- Configure libraries
- Setup Firebase
  - Sync Adapter/IntentService/AsyncTask will not be required with Firebase.
  - Loaders are not required as results will appear through callbacks. The view will be updated onSuccess() of the respective callback.
- Setup Gradle Dependencies
- Setup basic third party library
- Organising app in MVC Architecture.

Task 2: Implementing Database

- Setup API calls
- Setup database
- Storing memes in firebase on first run

Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity with
  - Recyclerview
  - Placeholder for handling empty list
  - Floating action button

- Build UI for creating Meme activity with
  - 4 fragments as shown
  - Adding all necessary UI components
  - Saving Memes
  - Adding share intent

Task 4: Implementing Canvas

In this step

- The canvas will be implemented with the help of the said library.
- Build UI for viewing the selected meme and drawing over it.

Task 5: Implementing Google Play Services

- Admob

Task 6: Others

- Widget
  - Building screen1 as a grid.
  - Building all necessary managing code files.
- Accessibility
- RTL

Task 7: Build Variants

- Free and paid flavours in gradle
- Changing activity and fragment.

Task 9: Error Handling

- Rotation
- Enhancing performance
- Testing app for various plausible errors.
- Debugging app for all the errors

Task 10: Publishing App

The app will be published in this step.

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**”