

## How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: “**Capstone\_Stage1**”
3. Replace the text in green

---

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any 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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** josephx86

# MemeFactory

## Description

MemeFactory allows a user to quickly make memes using pictures from a mobile device or the camera. Memes can be shared on platforms like Twitter and Facebook.

## Intended User

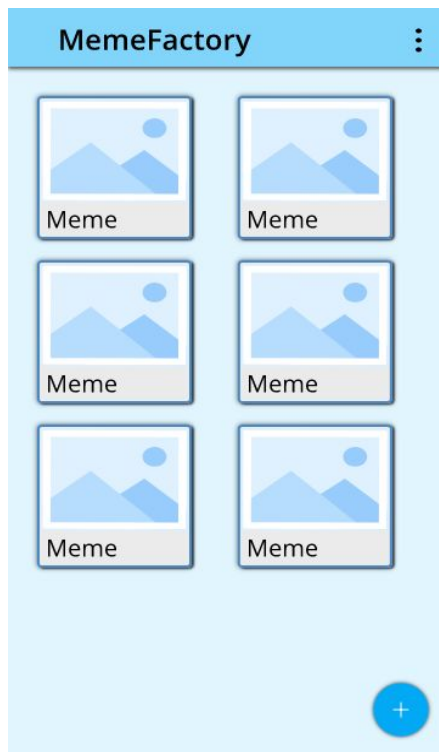
Anyone who enjoys jokes and memes.

## Features

- Creates memes from pictures saved in device or from camera
- Saved pictures to local storage
- Shares memes to social networks

## User Interface Mocks

### Screen 1



App home/library screen: Shows all the memes created in the app. From this screen, the user can also create a new meme.

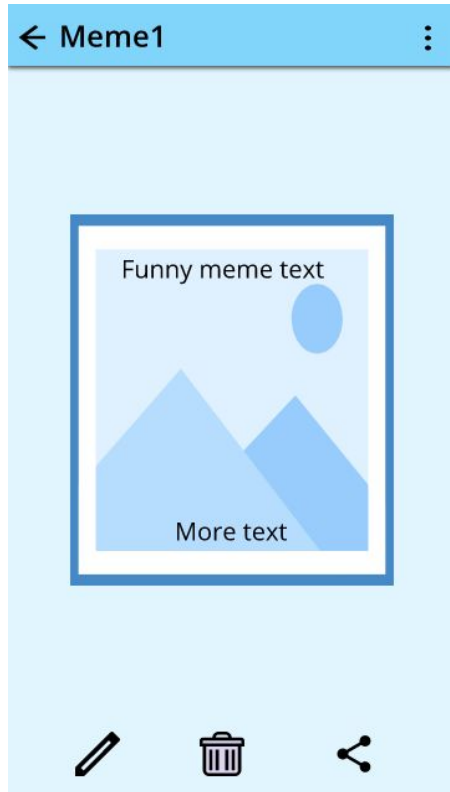
### Meme Widget



Meme will be displayed on the home screen as widgets in a recyclerview. The meme widget will be made from a CardView found in the Android Support Library. The Cardview will have the following child views:

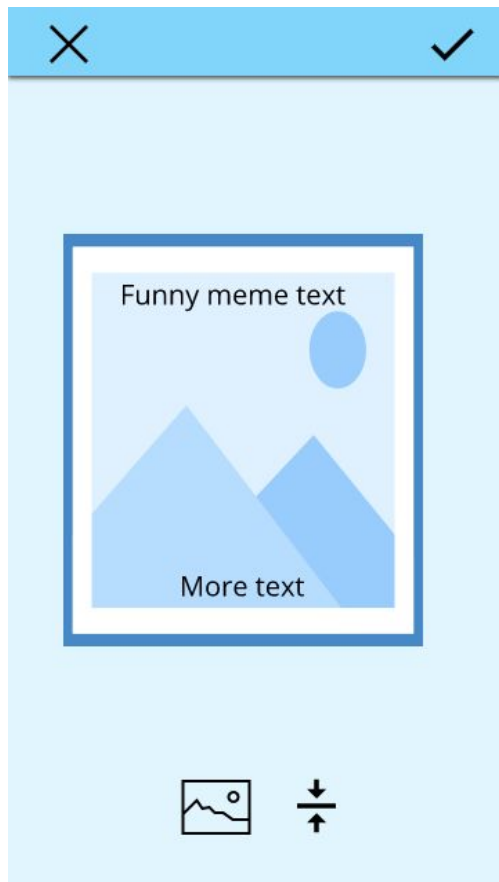
- ImageView - will show meme preview
- TextView - will show name of meme

## Screen 2



When a meme is selected, the app will allow the user to edit, delete, or share the meme.

### Screen 3



When adding or editing a meme, the user can set the background using a picture saved on the device or from a picture taken by the camera. The use can also toggle between 1 and 2 lines of text.

## Key Considerations

### How will your app handle data persistence?

The app will use a local SQLite database that will be accessed through a content provider. The app will also use Firebase Realtime Database for data persistence so that a user's meme data will be available on different devices after sign in.

### **Describe any edge or corner cases in the UX.**

If a user uses the camera to take a picture, the resulting image might be very large in size such that it slows the app's performance. It is also the case that in memes, users do not really need a high resolution image. To take care of this issue, the app will resize the image to a default of 320px x 320px.

On a tablet, the app will take advantage of the screen space and use master detail view to manage memes.

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

Picasso: Loading images

Schematic: Content provider and SQLite database.

Android Support Library: For cards and recyclerviews

Unity Ads: For ads in the ad version of the app.

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

The app will use Firebase Realtime Database and Firebase Authentication to save a user's meme data. Google Mobile Ads library will be used in the ad-version of the app to display banner ads at the bottom of the screen. Please refer to Task 1 below for the versions used for each of the libraries.

## **Next Steps: Required Tasks**

### **Task 1: Project Setup**

The versions of plugins and libraries used will be updated if newer versions become available. The project will be setup as follows:

- Development environment will be:
  - Android Studio - v 3.1.4
  - Gradle plugin - v 4.10
  - App will be written solely in the Java programming language
- Configure libraries:
  - Picasso - v 2.71828
  - Butterknife - v 8.8.1
  - Schematic - v 0.7.0
  - Android Support Library - v 27.1.1
  - Google Mobile Ads - v 15.0.1
  - Firebase Authentication - v 16.0.3

- Firebase Realtime Database - v 16.0.1
- Create vector graphics/images used by app
- Create color scheme and save then in xml as part of the default style.
- Save string in strings.xml and colors in colors.xml
- Configure app flavors - ad-version and ad-free version.

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

List the subtasks. For example:

- Build UI for
  - home/library screen
  - new/edit meme screen
  - manage meme screen
- Build UI for login/signup and implement Firebase Authentication
- Create menu xml file. Menu will have: 1. About app, 2. Get ad-free version

## Task 3: Meme widget implementation

The app will load meme into a recyclerview on the home screen. The recyclerview will use widgets which are created using a CardView widget. The CardView will have

- An image view that shows a preview of the meme
- A TextView with the name of the meme

## Task 4: Database configuration and access

Use Schematic to setup access to SQLite database through a content provider.

Database will have the following columns:

- ID - int, auto\_increment
- BackgroundImage - blob/image
- TextLine1 - string
- TextLine2 - string
- Meme - blob/image

## Task 5: Load memes on home screen

Now that the database has been configured, app will load memes on home screen

- Use content provider to load memes from database into a recyclerview
- Use floating action button to open activity that creates a new meme

## Task 6: New meme

When creating a new meme

- allow setting background using image from local storage or camera
  - resize image if it is too big (a side is larger than 720 pixels)
- allow use to toggle between 1 and 2 lines of text for the meme
- the user can type text
- if user cancels creating meme, go back to home screen
- if user saves meme
  - generate meme by combining image and text into a new image.
  - save meme data to database
  - go to manage meme screen

## Task 7: Manage meme

When a user selects a meme from the home screen, open activity that allows user to either edit, delete or share a meme.

- Edit - open new meme screen. It can also be used to edit a meme.
- Delete
  - delete database record
  - go back to home screen
- Share
  - Meme is an image. It will be shareable by any app that can handle/accept posting images.

## Task 8: Implement Firebase Realtime Database

In order to allow a user's data to be available across different devices/reinstall

When home screen is loaded and user is signed in, check if all data in Firebase Realtime Database is available in the local SQLite database. A timestamp will be used to determine last sync time.

When a new meme is created and saved to the local database, also save it in the Firebase Realtime Database.

## Task 9: Implement Google Mobile Ads

- App will display banner ads at the bottom of the screen in the ad-version of the app.

---

### 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**"