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.

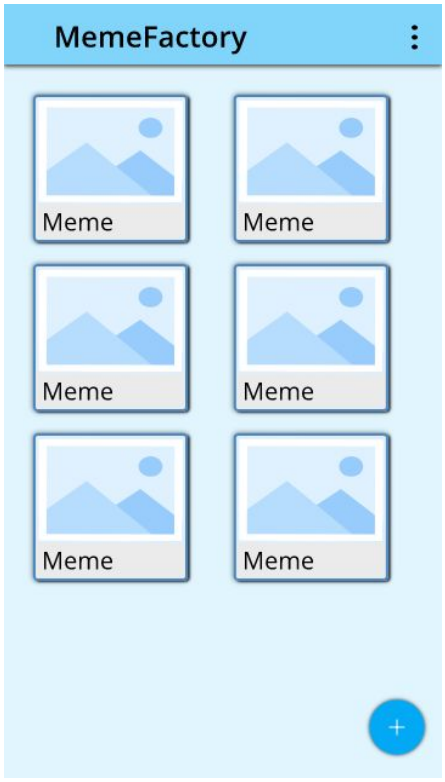
Features

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

User Interface Mocks

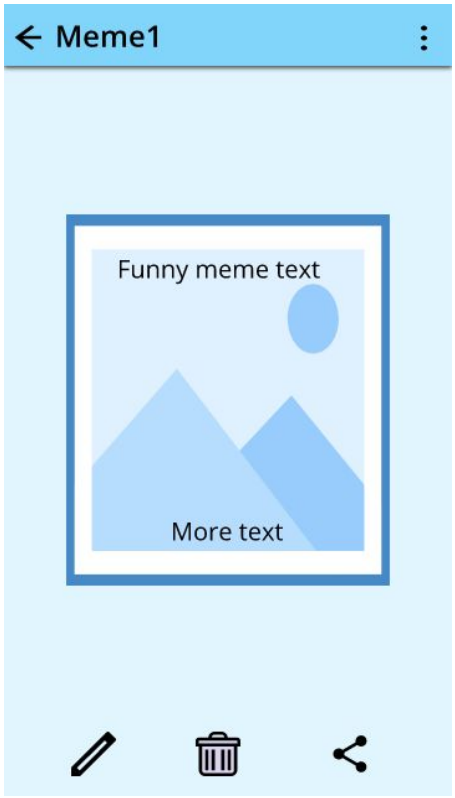
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



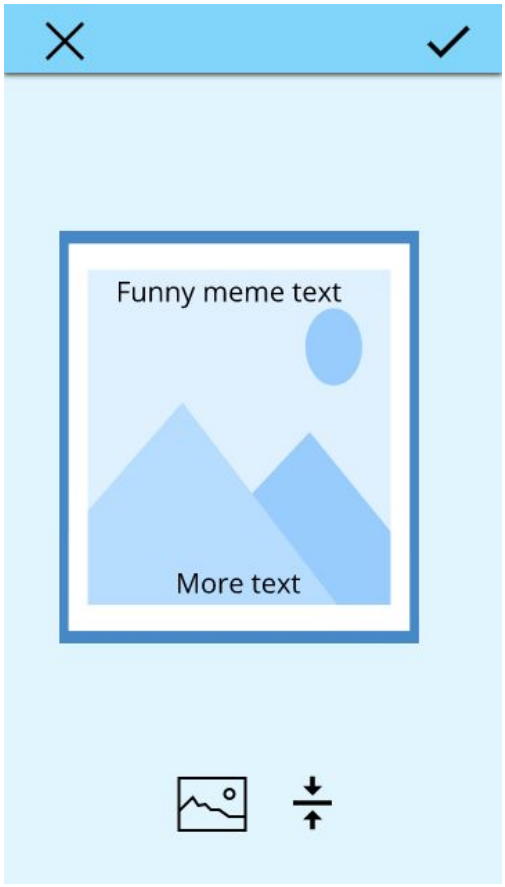
App home/library screen: Shows all the memes created in the app. From this screen, the user can also create a new meme.

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.

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 slowly 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 not use Google Play Services. For the ad version of the app, Unity Ads will be used. Other apps may opened apps when sharing memes.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Configure libraries: Picasso, Schematic, Android Support Library, Unity Ads
- Create vector graphics/images used by app
- Create color scheme and save then in xml as part of the default style.

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
- Create menu xml file. Menu will have: 1. About app, 2. Get ad-free version

Task 3: 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 4: 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 5: 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 5: 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 6: Implement Unity Ads

To make ads less annoying to the user, only show an ad under the following conditions;

- At least 3 minutes have passed since last ad was shown
- When navigating to new/edit screen

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**"