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

GitHub Username: HaroldRidley

Meme_Me

Description

Create your own custom memes using custom or stock images and share them with your friends.

Intended User

Anyone who likes to share memes with their friends.

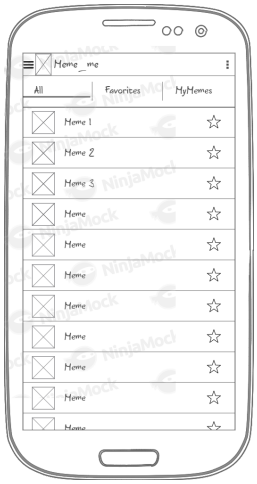
Features

- Saves information
- Shares images with other apps
- Creates custom images

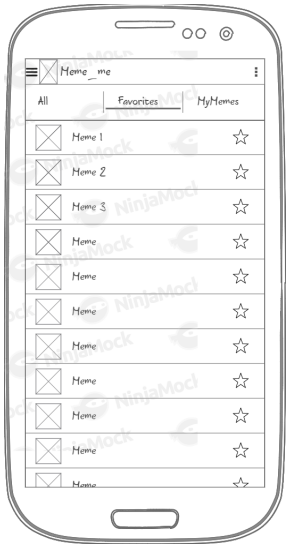
User Interface Mocks

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

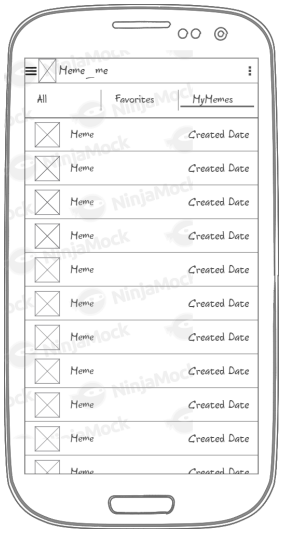


Screen 2

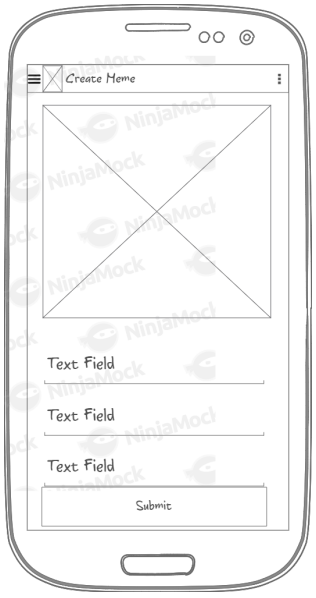


Screen 3

Screen 4



Screen 5



Screen 6

Key Considerations

How will your app handle data persistence?

Firebase Realtime DB.
I will be storing simple strings, urls of images and such.
Firebase makes this very simple

Describe any edge or corner cases in the UX.

A meme will have multiple text boxes.In cases that the number of text boxes is greater than 2 The view for creating a meme will need to be dynamic to accommodate the appropriate number of text boxes.

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

Glide- loading / manipulating images
Moshi - serialization / deserialization of Json
Timber - Logging
Butterknife - make binding views easier
RxJava - handle multiple asyc calls and merge results
Espresso - UI testing
Junit - Unit testing

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

Google Play services - small banner ads on views
Firebase - Realtime DB - storing simple strings

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

- Configure libraries
- Create empty activity
- Set up AppTheme
- Configure colors
- Create Common Styles
- Import Libraries
- Configure third party libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Create RecyclerViews with mock data
- Build UI for ImageDetailActivity
- Build UI for CreateMemeActivity
- Implement banner ads on Mainactivity.

Task 3: Implement Library to call API

- Setup Firebase RealTime DB
- Implement Firebase SDK
- Implement Google Play Services for Ads

Task 4: Create Library to call Imgflip api

- Create method calls to get all images from api
- Create method calls to POST data and create meme
- Handle Error case
- Handle success case - store results to Firebase DB
- Ensure network calls are not made on main thread

Task 5:Call API to populate MainActivity

- Call Method to get all Memes
- Create Adapter
- Pass Json response into adapter
- Render Memes list

Task 6: Implement Search Functionality on MainActivity

- Create Methods to filter on adapter based on strings
- Modify adapter
- Render new list

Task 7: Implement Favorites functionality

- Implement onclick listener on star
- On click move meme object to favorites array
- When star is unclicked remove from favorites array

Task 8: Pass selected meme to createimageview

- Render selected image on view
- Generate appropriate number of text boxes
- On submit, call method to POST to API

Task 9: Handle response from Creating Memes

- On success, display imagedetailactivity
- On failure, stay on current activity
- Display toast with error message

Task 10: Implement Share functionality

- Create intent to share
- Launch intent

Task 11: Implement shared preferences in settings

- Create font style shared preference
- Capitalization shared preference
- Create font size shared preferences

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**"