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**”

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: hhprogram

FoodOnMyMind

Description

- Keep track of recipes you’ve made (either from a website or one you’ve created yourself) and document the outcome with notes, tags and pictures
- Search recipes based on keyword searches or cuisine type

- Uses maps / location to find nearby restaurants that most closely match the keyword search / cuisine type searches
- Save which restaurants you have gone to and make notes, tags and add pictures alongside document each of your experiences there

Intended User

People that really enjoy cooking / documenting what they cook or their experiences out in restaurants.

Features

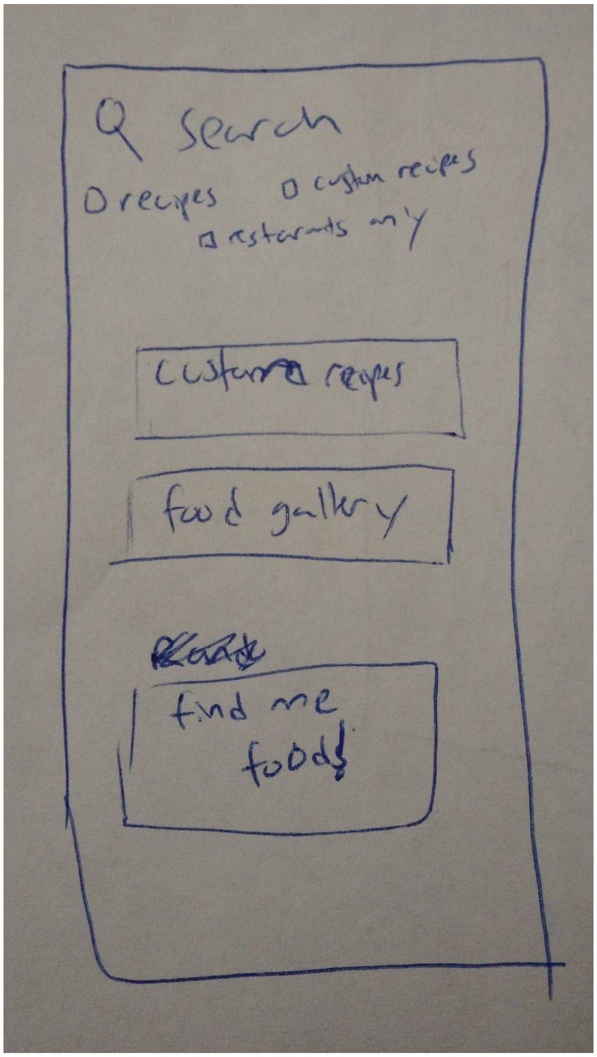
List the main features of your app. For example:

- Saves information
- Takes pictures
- Upload pictures from existing phone gallery
- Use GPS location and Maps to find nearby restaurants
- Search for restaurants
- Search for recipes

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.

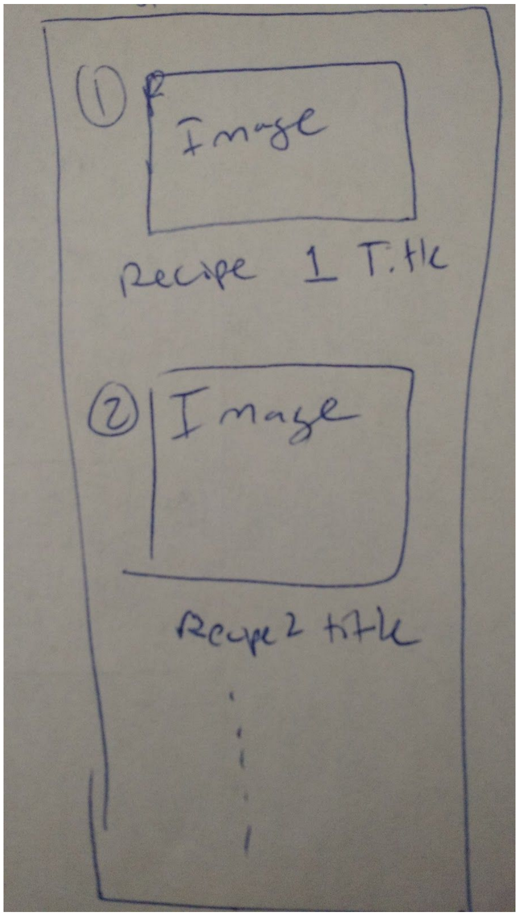
Screen 1



This will be the start screen.

- Will have the search bar at the top with 3 checkboxes (1) Website recipes (2) Custom entered recipes (3) Restaurants
- Custom Recipes (button)
- Food Gallery (button)
- Find Me Food! (button)

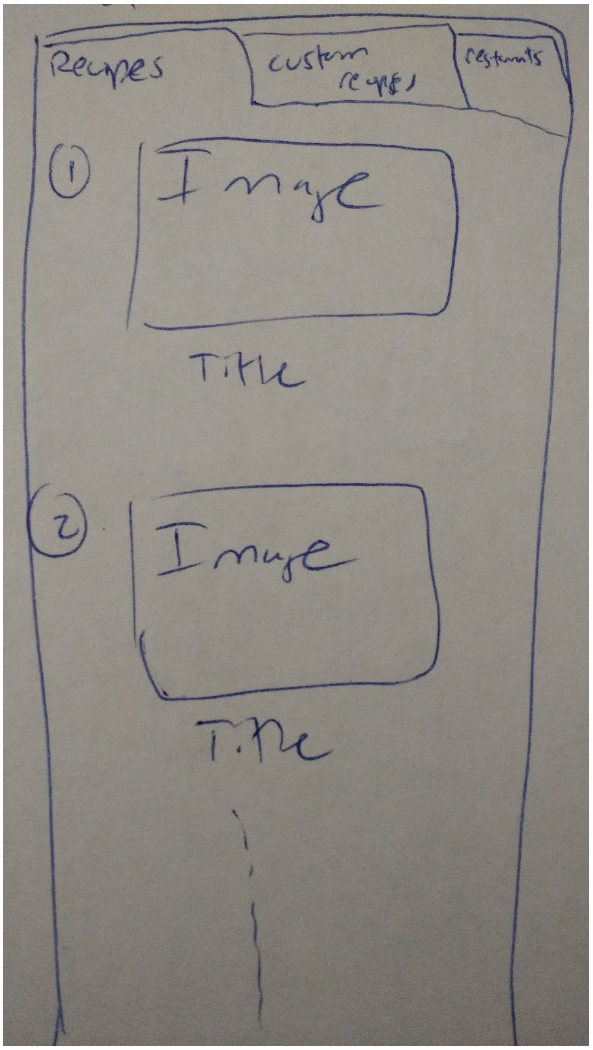
Screen 2



Resulting screen if either only one of the checkboxes is checked and a search is done

- A scrolling view that lists the relevant results
- Each item in the list will have an image and then the title associated with that image (whether it be recipe title or restaurant name)

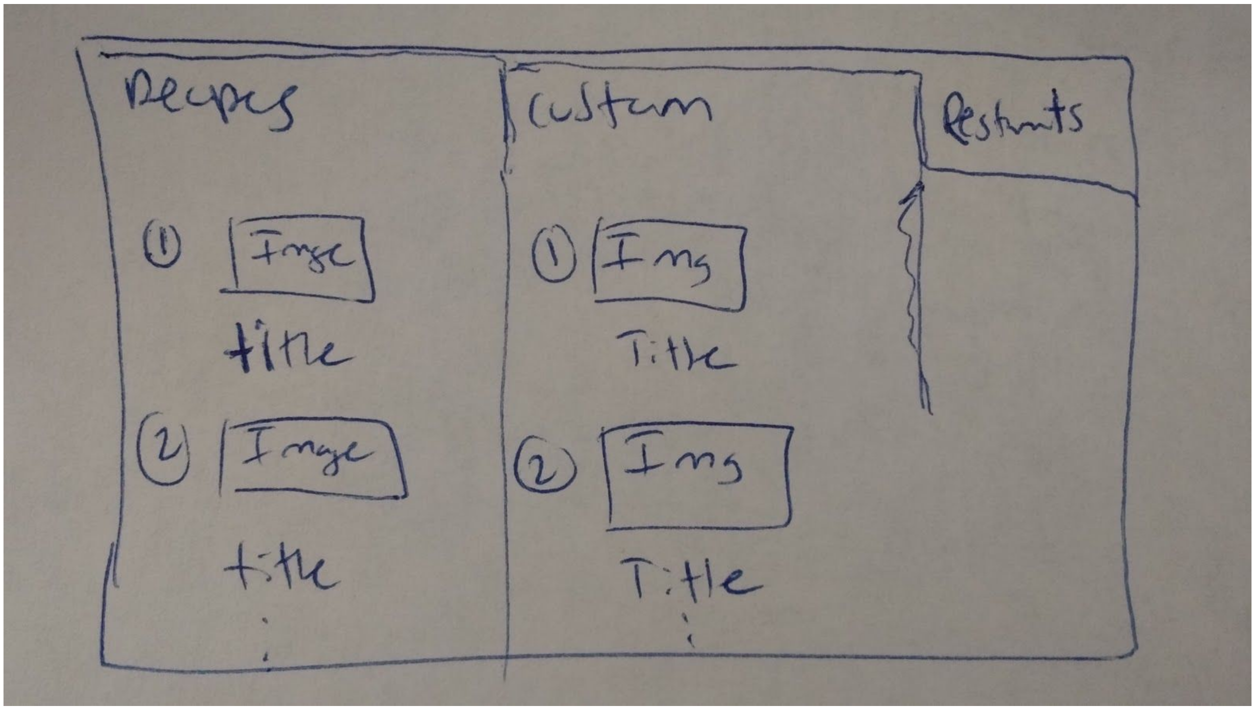
Screen 3



Resulting screen if at least 2 of the check boxes are checked and a search is done.

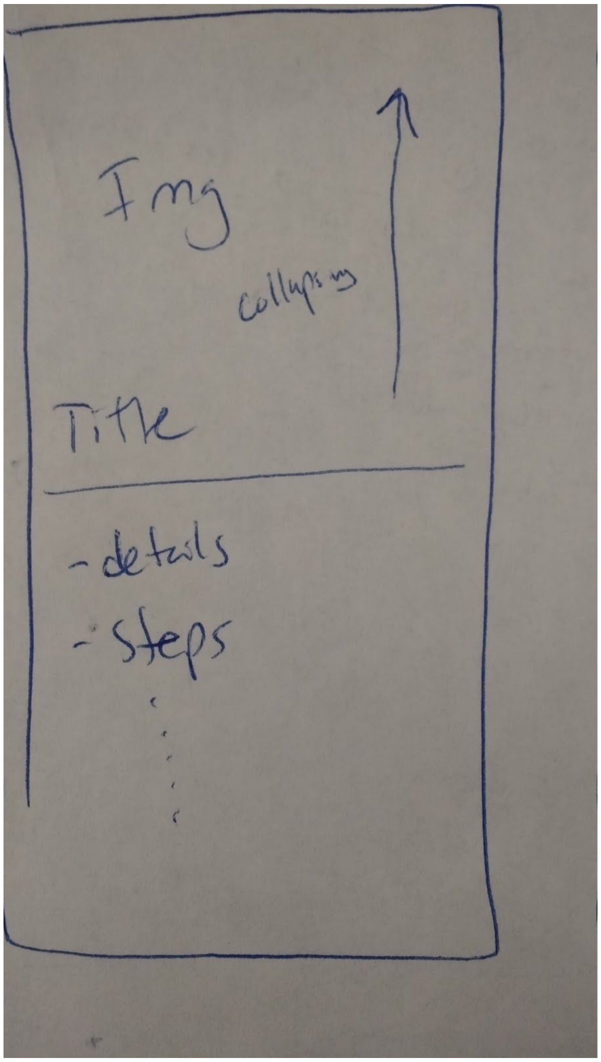
- Will allow for tabbing between result categories (one tab for each check box)
- Then within each 'tab view' it will be the same scroll view as screen 2

Screen 4



Same resulting screen as screen 3 just in landscape mode. If landscape mode then show resulting lists side by side. If even larger device (like a tablet) then show all 3 resulting lists side by side if applicable

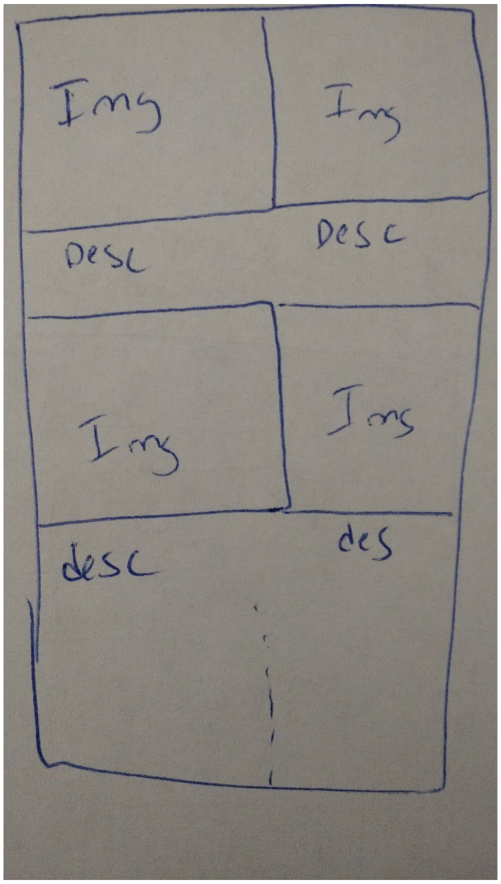
Screen 5



Resulting screen when you click on an item in the results list screen

- Collapsing toolbar with the ‘main’ image of the item as the background image of toolbar
- Details / steps under the collapsing toolbar

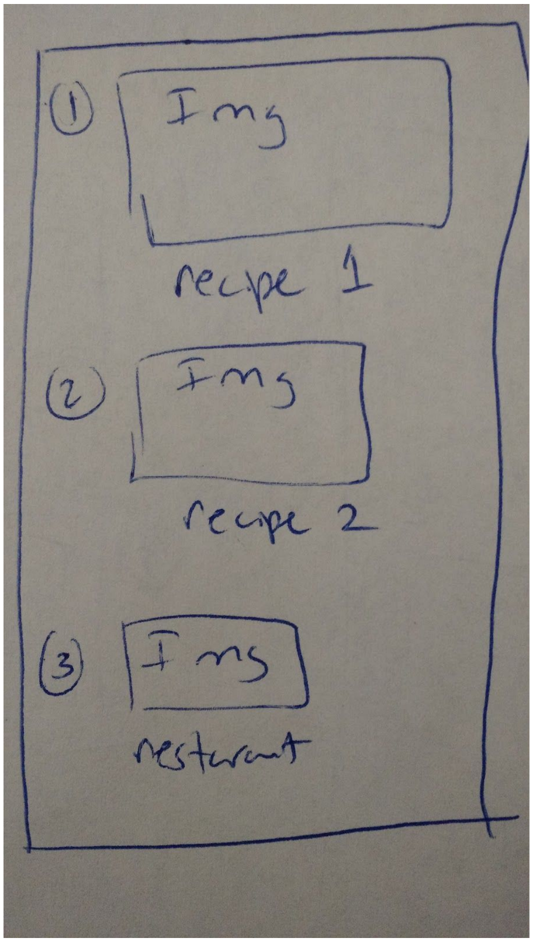
Screen 6



Resulting screen when click the “Food Gallery” button

- Will be a collage of images with small descriptions (user inputted) of each image
- Images will consist of either photos taken through app or loaded into the app from the phone gallery
- Clicking any of the pictures will just full screen the picture and show a longer description if available

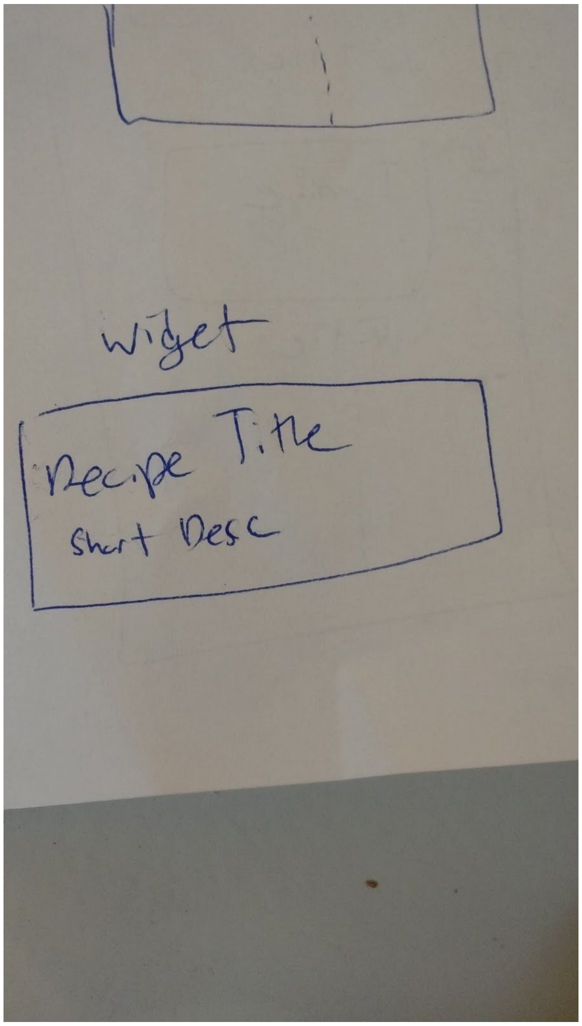
Screen 6



Resulting screen after pressing the “Find me Food!” button

- Will return 2 recipes that we will randomly get from recipe websites
- Return 1 random nearby restaurant

Screen 7



This is the widget UI mockup

- App will generate a ‘recipte of the day’ (either a random one from a website or a random one from one of the user entered custom recipes)

Add as many screens as you need to portray your app’s UI flow.

Key Considerations

How will your app handle data persistence?

Use a content provider for all my text needs, then couple that will external storage for saving images that the user has uploaded onto the app or taken with the app

Describe any corner cases in the UX.

For all screens that are more than just one step from the home page. Show a small ‘home icon’ to allow user to quickly go back to the home screen to allow for quicker navigation. (or do this by putting this button under the ‘menu’ section of toolbar

Will be possible to ‘associate’ multiple images with one custom recipe. Allow user to see these other than just the ‘main’ image (in the collapsing toolbar background. By either (1) when image is associate with recipe user can associate it with a ‘step’ and thus this image will be displayed in the scroll view once the step is displayed (2) put a gallery button in the ‘menu’ section of toolbar and elevate a photo gallery that is ‘above’ the recipe that user can flip through and interact with

If user wants to scroll through recipes or restaurants once within the details page support ‘paging’ to go to the next or previous item of the list with a side swipe motion

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

Use picasso for handling images

Describe how you will implement Google Play Services.

Will use admobs to show ads where necessary, Maps to show relevant info on restaurants nearby search

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

- Set up content provider for data persistence
- Set up AsyncTask class to help retrieve data during searches

- Set up an adapter to help easily show scrollable views (with their corresponding titles / images) for food gallery view, results view
- Set up external storage for app to allow user to build image collection

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Home screen
- Build UI for Results screen (for 1 - 3 boxes checked)
- Build UI for each individual item within the results screen
- Build UI for food gallery page
- Build UI for each individual item in the food gallery page
- Build UI for details page (which should be structured the same whether it be a custom recipe, website recipe or restaurant)
- Build UI for Find me Food! Button
- Build UI for the ‘gallery pop-up’ window when user wants to look at more pictures associated with recipe all at once and edit them

Task 3: Implement Google Play Services

- Create a layout to serve ads
- Set up correct gradle build configuration to allow for free (ads) and paid (no ads) versions

Task 4: Implement Maps

- Set up necessary maps dependencies
- Put in necessary code within Manifest file to use user location

Task 5: implement content provider

- Build the ‘contract’ file to define the persisting database
- Build the ‘helper’ file to make it easy to actually build the database when needed with the defined columns in the ‘contract’ file
- Build the ‘provider’ file to implement actual queries into the db and what results should be returned and other required db interactions

Task 6: Set up external storage

- Set up manifest to allow for external storage use
- Set up methods to allows for external storage / error checking in terms of available memory

Task 6: Implement a loader

- Implement a loader in the results list activity screen and the food gallery screen that loads the persistent user data and allows for easy population of views of the text values related to each recipe or user review of restaurants
- Loader ideally will also have access to some sort of key or way to reference the ‘externally’ stored images that the user has added to the app

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**”