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: dibellaa

WhatsInMyFridge

Description

WhatsInMyFridge helps you to do not waste your ingredients proposing to you simple recipes. You add what you have in the dispensation and WhatsInMyFridge will suggest to you the dish that does to your case!

Intended User

Family that want to give a refresh to their menus. Students that have to eat something :P

Features

- Keep a list of ingredients and keep track of it
- Propose recipes based on the ingredients
- Search recipes by name
- Keep a list of favorite recipes

Java language will be used for development.
Use Loaders with content provider.
Implement Intent Service for Widget.
App support accessibility (for example contentDescription).
All strings will be in res/values/strings.xml.
All images in res/drawable
App enables RTL layout switching.

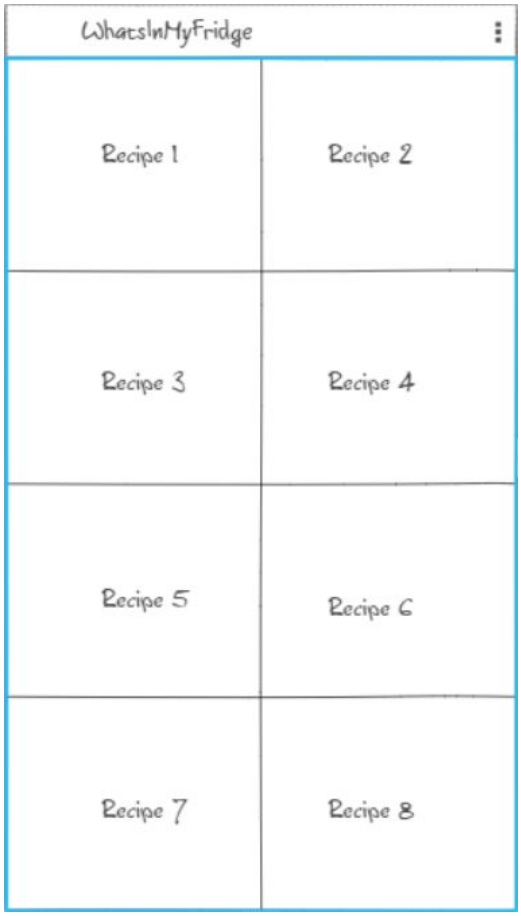
User Interface Mocks

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.

MainActivity 1

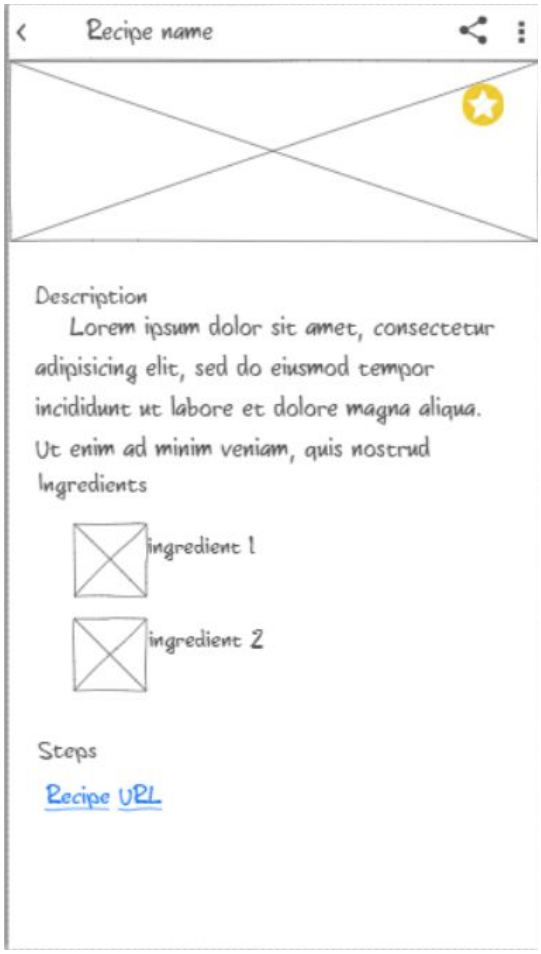
If there are not any ingredients added (first time app launched), this will be the screen. The button will open the Ingredients activity in order to add new ingredients.

MainActivity 2



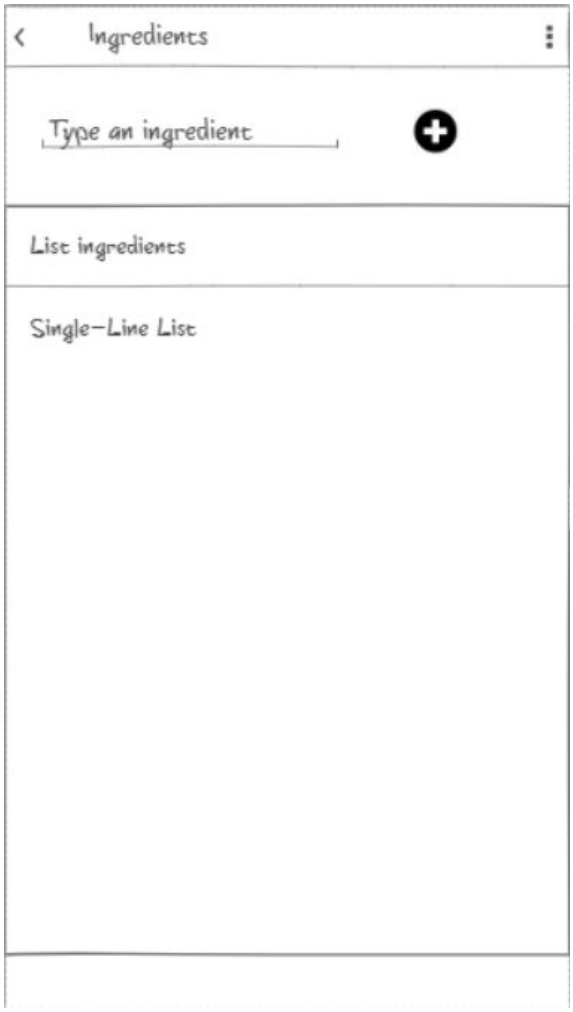
Grid layout with the recipes that the user can cook with the provided ingredients. Each item will open the Recipe activity of the relative recipe.
The Main activity can display only the favorite recipes (option in the menu for example using shared preferences).

RecipeActivity



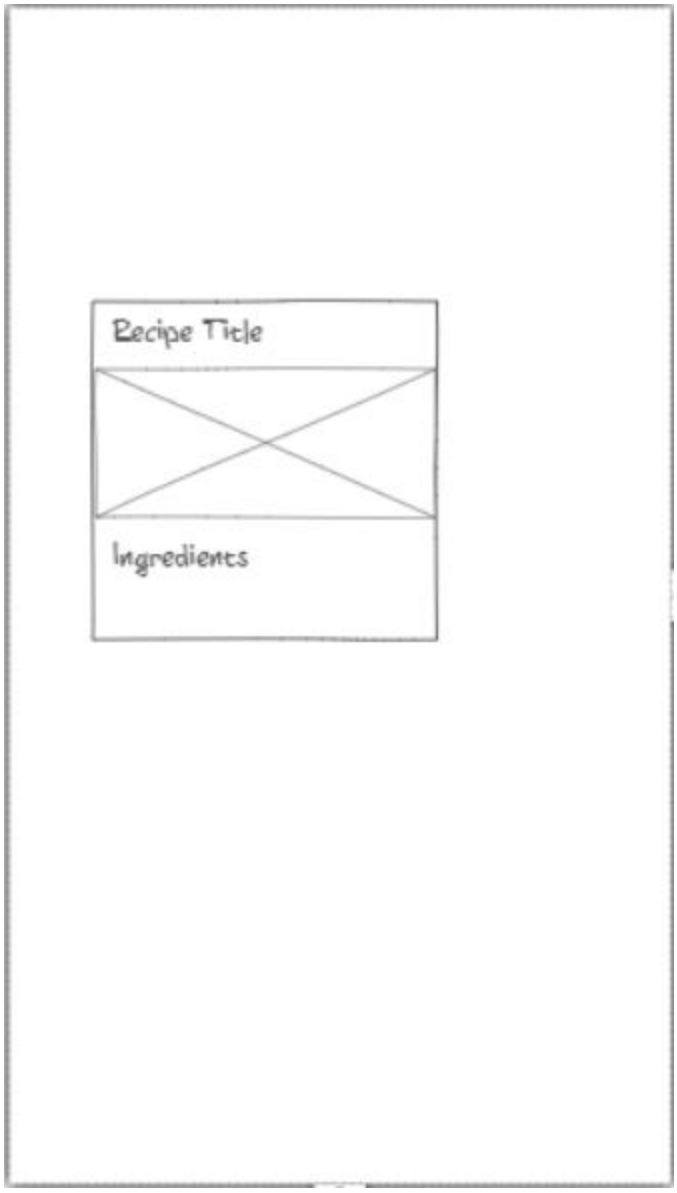
Recipe activity, with the description of the Recipe. The user can put as favorite, share it and open the URL of the recipe steps.
There will be a SQL database and its content provider for the favorite recipes.

IngredientsActivity



Add ingredients to the application. There will an SQL database and its content provider to maintain the list of ingredients.

Widget



Pin a favorite recipe with title, image and ingredients.

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

Key Considerations

How will your app handle data persistence?

All the sata in the application is stored in a local repository using SQL. A content provider will be used to connect the local repository and the main application. The main entities which will be stored are:

- Favorite Recipe
- Ingredient

Describe any edge or corner cases in the UX.

A widget will provide the favorite recipe and its ingredients. When the user taps on the widget it will open the application displaying the recipe.

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

- Spoonacular API: for Food API
- Picasso to load the recipe images: com.squareup.picasso:picasso:2.71828
- Gson to manage jsons: com.google.code.gson:gson:2.8.5
- Butterknife for binding layout: com.jakewharton:butterknife:8.8.1
- Timber for logs: com.jakewharton.timber:timber:4.7.1
- Gradle version: 3.1.1
- Android Studio: 3.1.1

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

AdMob, to show ads in free version.
Analytics API

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

- Create new project in Android studios.
- Adding project dependencies and third-party libraries (like spoonacular)
- Configure free and paid flavors

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity and other Activities
- Build the layouts

Task 3: Widget design

- create the widget

Task 4: Enhancement of design

- Beatify the application with transactions

Task 5: Test

- Create UI tes cases with Espresso

Add as many tasks as you need to complete your 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**"