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**"

GitHub Username: vane101

Meal Split

Description

Meal Split is an app that aims to solve a simple problem that many restaurant waiters face in their day to day job which is taking orders and splitting bills of a large table. Meal Split makes it easy for waiters to add, delete and split restaurant bills so that every member of the table pays their required share.

Intended User

Meal Split is intended to be used by restaurant waiters, mostly restaurants where school or campus kids frequent in groups.

Features

- Creation of tables.
- Create custom menus with categories
- Adding customers and taking their orders.
- Calculating bill totals.
- Splitting bills, combining bills and transferring bills from one customer to the other.
- Keep track of open and closed tables.
- Take picture of menu items

User Interface Mocks

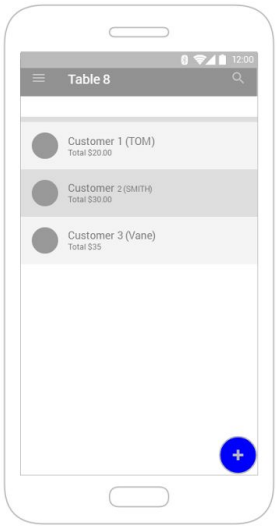
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

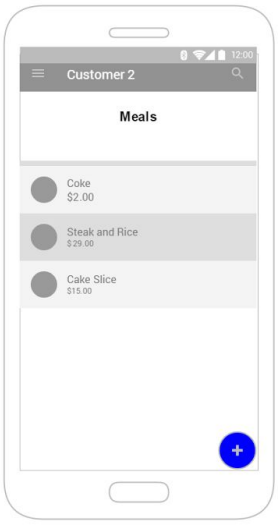
Home Page, shows open tables

Screen 2

Creating new Table Screen



List of Customers in one table



List of meals ordered by a customer

Tablet design will be a Parent child layout of Tables Open and Customers as well as Customers and Meals. I am also thinking of creating a Parent child layout for the Menu and its Details.

Key Considerations

How will your app handle data persistence?

Meal Split will utilise a SQLite database to save data as it will not be connecting to the internet. The following tables will be created:

- Table
- Customer
- Menu_Category
- Menu
- Orders
-

A Content Provider will be used to interact with the database and its tables and a Loader to transfer the data to the UI.

Describe any corner cases in the UX.

Some of the corner cases to consider would be the number of images of menu items a user can save based on their phone memory capabilities. This means there will be a limit on the number of menu images a user can add on their phone. A function to add/remove menu images will be provided.

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

Picasso will be used to load images from memory to the UX, I have used Picasso in a number of projects and it has proven itself very reliable.

I will also be using the "stated-fragment_support" library by "thecheeselibrary" to get data from external intents to the fragment that called it. This will be useful for Camera intent called via a fragment.

I will also be utilising a roudedimageview library to improve the way images are being displayed.

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

Major steps that will be followed:

- Update Android Studio IDEA
- Update Android SDK
- Add required libraries to Gradle.
- Enable development mode on mobile phone
- Think about what Google Play Service to add to the Application

Task 2: Implement UI for Each Activity and Fragment

Subtasks for this project:

- Setup Navigation Drawer for the MainActivity
- Create fragment for the Main Activity that shows open tables;
- Create Customer Fragment for the Table Activity,
- Create Orders Fragment
- Create Menu Activity with Fragment

Task 3: Implement Data Models and Content Provider

- Design database structure
- Create data model classes
- Use SQLite to create database and implement CRUD functions
- Create Content Providers;
- Implement Loaders.

Task 4: Implement Meal Menu

- Implement ability for user to capture images of food items

Task 5: Material design

Add material design colours and elements to application

Task 5: Implement Widget

Implement widget that shows the user number of tables still opened.

Task 5: Testing

Test application and catch any exceptions

- Test on phone
- Test on Tablet

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**"