Software Design Description

Meal Budget Plan (MBP)

V.1.2

February 02, 2024

https://docs.google.com/document/d/1bf8CZwRk0hHykRXEZdOg\_HPlujac6dNkhr\_CVDTAUQo/edit?usp=drive\_link

Biola University, Department of Math and Computer Science

**Revisions Page**

**Overview:**

The purpose of this document is to outline the design with a top down approach. The document will begin by giving the description of the different requirements through the use of a RACI table. The document will also provide additional in depth information about the Architecture of system in place used to design the application. Through this document we will also document the function of the components as well as designs of the user interface.

**Target Audience:**

This document is intended for the use of all computer science students part of this project. As such this document should be organized, understandable, and readable to all who seek to contribute and aid them in completing tasks in a timely manner, while maintaining a uniform organized plan and structure. Beyond the team this document will aid our professor have a clear understanding of our processes, organization, and schedule.

**Project Team Members:**

Rachel Liu, Jonathan Willey, Miguel Oh, Oscar Navarro

**Version Control History:**

The following table will include the version history of this document in order to log any changes/amendments. Each update should include a new version depending on the size of the update as well as the person responsible, description and date.

| **Version** | **Primary Author(s)** | **Description** | **Date Completed** |
| --- | --- | --- | --- |
| 1.0 | Oscar Navarro | Creation of initial document | 01.28.2024 |
| 1.1 | Rahcel Liu | Inclusion of wireframe for UI | 02.02.2024 |
| 1.2 | Miguel Oh | Database Entities & Business Rules | 02.02.2024 |
|  |  |  |  |
|  |  |  |  |

**Table Of Contents:**

[1.0 Introduction: 1](#_iolqnwqz8r8n)

[1.1 Design Overview: 1](#_auqc124bcisb)

[1.2 Requirement Traceability Matrix: 2](#_5rvhyzo94ivr)

[2.0 System Architecture Design 3](#_t4ao91x5u6jb)

[2.1 Chosen System Architecture 3](#_6rrag2zbmd6f)

[2.2 Discussion of Alternative Designs 3](#_m3q1f9gr57gg)

[2.3 System Interface Description 4](#_rc62c81i52d2)

[3.0 Detailed Description of Components 5](#_hpqzu5emm934)

[3.1 Component 1: Pseudocode for Application Functions 5](#_vhby5uksyoi)

[3.2 Component 2: Database Structure 8](#_hzt0n6dpq164)

[4.0 User Interface Design 10](#_ib2tc91i3sc)

[4.1 Descriptions of the UI 10](#_6fvhmqr1xpjr)

[4.1.1 Screen Images 11](#_strq6f8ijim2)

[4.1.2 Objects and Actions 15](#_wpxdzvn0clum)

[5.0 Additional Materials 18](#_ryvr6nt0uoqj)

[5.1 Definitions, Acronyms, and Abbreviations: 18](#_nxas4lvokocs)

## **1.0 Introduction:**

### **1.1 Design Overview:**

This mobile application will be designed using a three tier architecture which will include a user interface, an application, and a back-end database. The user interface will include a variety of screens which will be navigated through the use of buttons and a menu. The application level will be the home to the functions used to gather resources from the database and change information in the database. The database will be the home to our schema containing all the different entities for the customer information, meals, ingredients, recipes, and archive.

### 

### **1.2 Requirement Traceability Matrix:**

### 

| **Req. ID** | **Req. Type** | **Requirement Description** | **Source** | **Priority** |
| --- | --- | --- | --- | --- |
| UI-1.0 | Required | General App Design | Initial Meeting 01/11/24 | High |
| DB-1.0 | Required | Record user information into customer database | Initial Meeting 01/11/24 | High |
| A-1.0 | Required | Budget Options |  | High |
| A-1.1 | Functionality | Meal and Ingredient implementation |  | High |
| A-1.2 | Functionality | Meals will include a cost, and caloric information |  | Moderate |
| A-2.0 | Functionality | Grocery list function |  | Moderate |
| A-2.1 | Required | Grocery Cost | Initial Meeting 01/11/24 | High |
| A-3.0 | Required | User Settings |  | Low |
| A-1.4 | Functionality | Meal Flexibility | Initial Meeting 01/11/24 | Low |
| A-1.3 | Required | Recipe | Initial Meeting 01/11/24 | High |
| A-3.1 | Functionality | Help Settings |  | Low |
| A-3.2 | Required | Authenticate User |  | High |
| A-3.3 | Functionality | My Budget |  | Low |

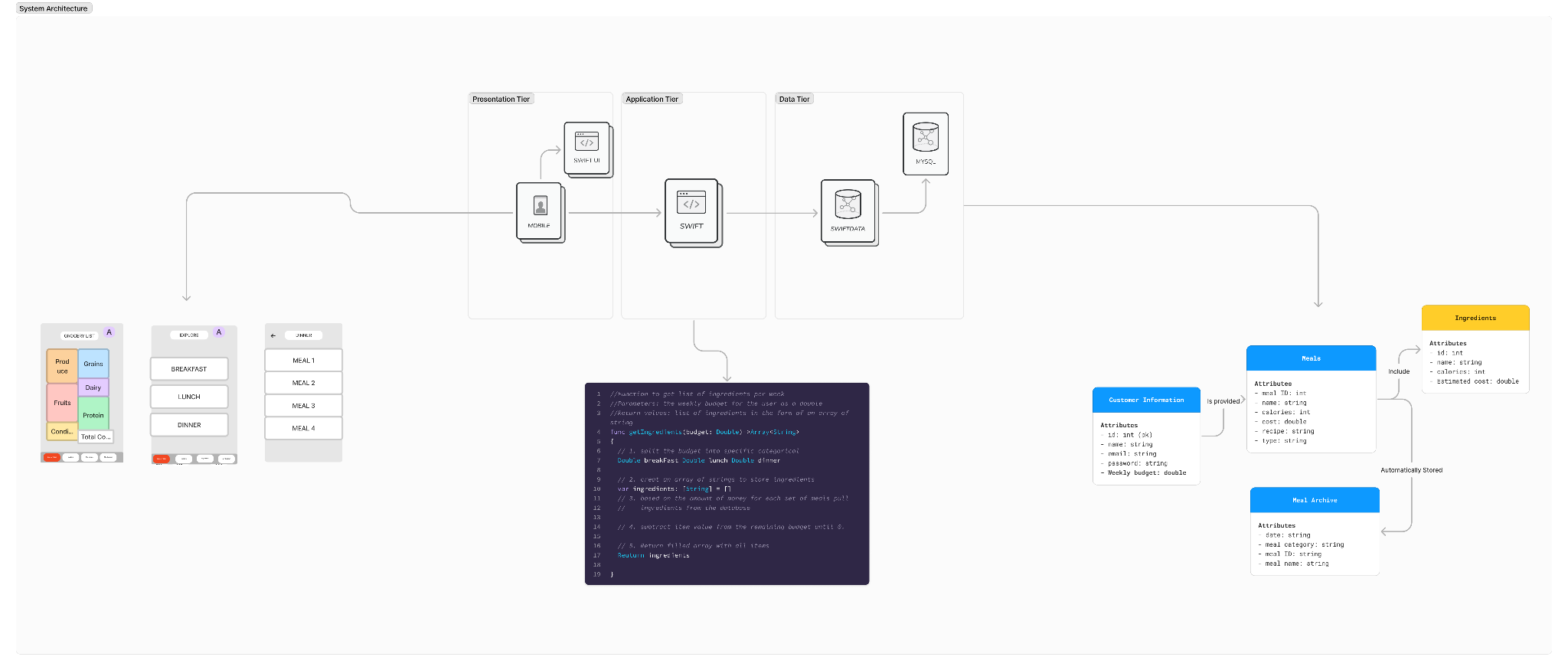
### 

## **2.0 System Architecture Design**

### 

### **2.1 Chosen System Architecture**

This application will use a three tier architecture splitting up its different components. The first layers of this architecture will be the presentation layer which will include that which has to do with the user interface as well as what is visible to the user. In this layer SwiftUI will be used to develop the user interface components. The user will be able to interact with the app through the use of a menu and clickable buttons. Upon the click of some buttons the application will be able to pull information from the database layer. The next layer of the architecture will be the application layer. In this layer the computational algorithms used by the application. This layer will take care of the functionality of the app including pulling and selecting meals for the user from a database based on the user configured settings. Beyond the application layer will be the data layer which will be the home to all the backend information needed for the application to function. This layer will be in constant communication with the application layer and through the use of SQL we will be able to have our algorithms pull from this data layer.



### **2.2 Discussion of Alternative Designs**

There was no open discussion of alternative architecture designs for our application. We saw that for the constraints of our schedule as well as to meet the necessary requirements a three tier architecture would be the most efficient while also being a complete option.

### 

### **2.3 System Interface Description**

Data Tier:

Stores data of customer and application essentials such as ingredients and meal recipes in order for the application tier to interact with.

Application Tier:

Program Development through Swift for Apple application of Meal Budget Plan. Will interact with the user UI and connect to the database for implementation of results.

Presentation Tier:

User Interface that the user will interact with on a given Apple device. The UI will be simple to use to efficiently interact with the software and have access to the information in the database.

## **3.0 Detailed Description of Components**

### **3.1 Component 1: Pseudocode for Application Functions**

Edit Settings:



Get grocery list



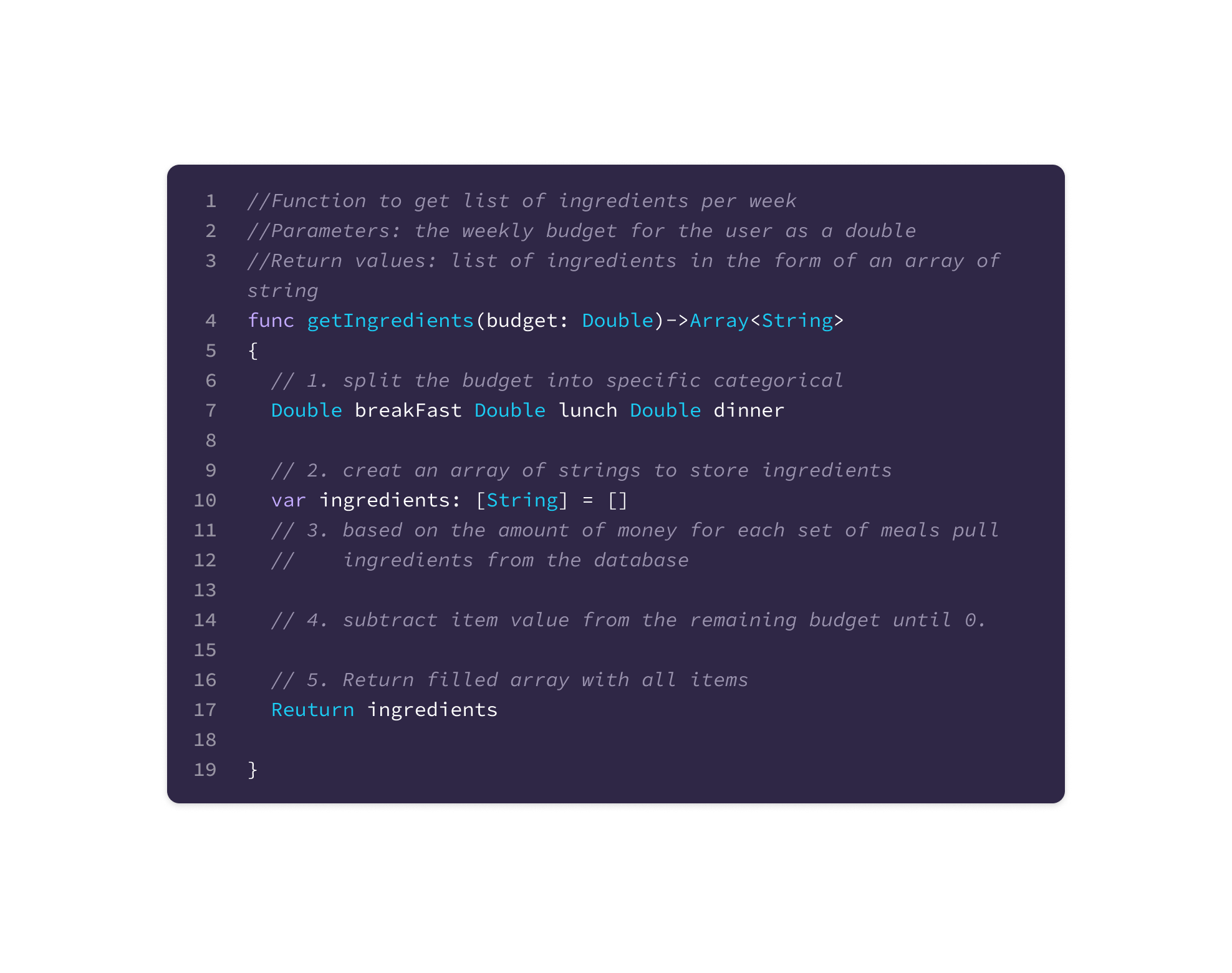
Get meals:



Get Recipe:



Get Ingredients:



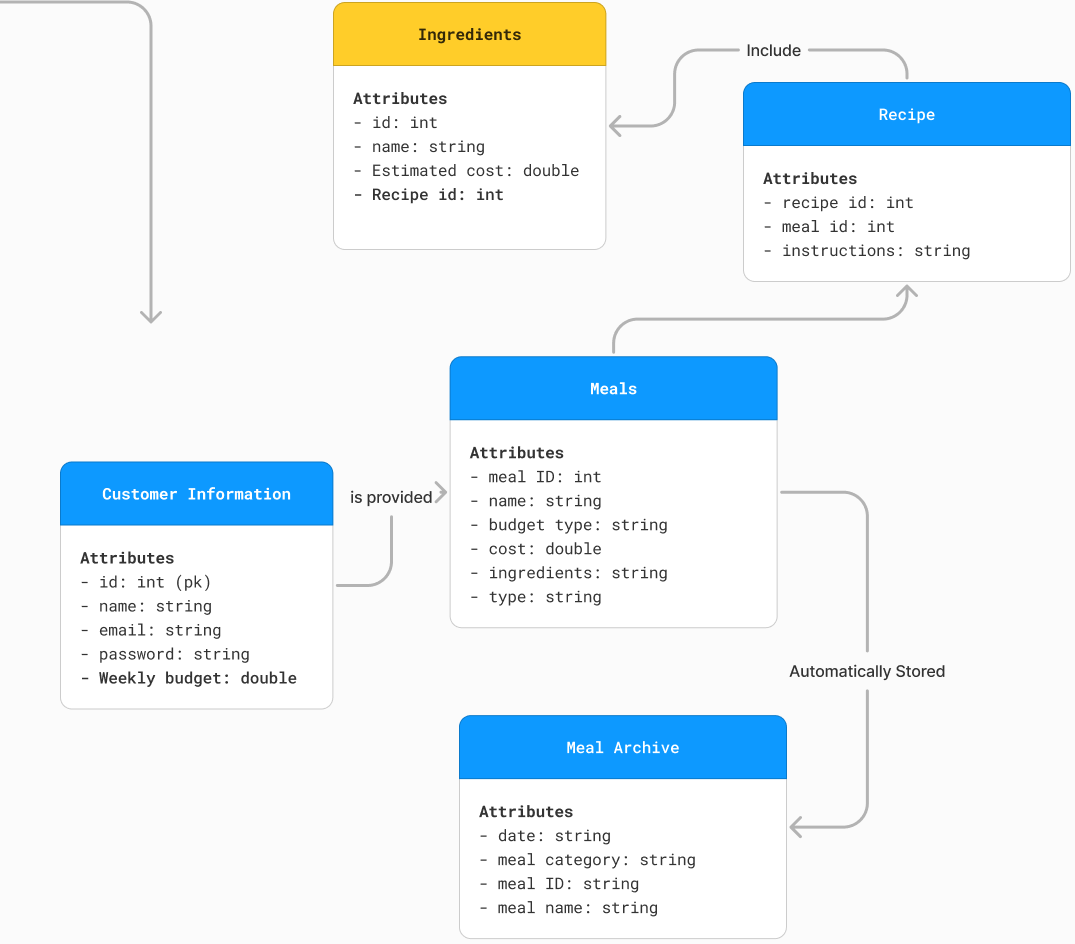
Select Meals:



The pseudo code presented represents the application layer components. The described functions include the main application functions of the app.

### 

### **3.2 Component 2: Database Structure**



## 

Data Field / Type

Customer Information -> CustomerID, Name, Email, Password, Budget

Meals -> MealID, MealName, BudgetType, Cost, Ingredients, Type

Ingredients -> IngredientID , IngredientName, Calories, EstCost

Meal Archive -> MealArchiveID, Date, Meal Category, *MealID*, Meal Name

Recipe -> RecipeID, MealID, Instruction

Entities:

* Customer Information - Stores information of the User ID, name, e-mail, password, and budget history

Attributes:

* + CustomerID - Int
  + Name - VarChar (50)
  + Email - VarChar (20)
  + Password - VarChar (30)
  + Weekly Budget - Double
* Meals - Stores Meal ID, name, calories, cost, recipe, and type of ingredient

Attributes:

* + MealID - Int
  + Meal Name - VarChar (50)
  + Budget Type - String
  + Cost - Double
  + Ingredients - VarChar (30)
  + Type - VarChar (30)
* Meal Archive - Stores date, meal category, meal name, and the Meal ID

Attributes:

* + MealArchiveID - Int
  + Date - Date
  + Meal Category - VarChar (30)
  + MealID - Int
  + Meal Name - VarChar (30)
* Recipe

Attributes:

* + RecipeID
  + MealID
  + Instruction - String
* Ingredients - Stores the Food ID, name, calories, and estimated cost

Attributes:

* + IngredientID - Int
  + IngredientName - VarChar (30)
  + Estimated Cost - Double
  + RecipeID

## 

## **4.0 User Interface Design**

### **4.1 Descriptions of the UI**

The UI will consist of a variety of screens which will allow the user to interact with all the functions of the application and get a visual representation of the stored data. The app will start for new users in the log in or create an account screen where the user will be able to create their account and customize their user settings. For returning users the application will launch to the main home page where the user will be shown the day and specific meals they have prepared for that day.

### 

#### 

#### 4.1.1 Screen Images

Account Log In:

### 

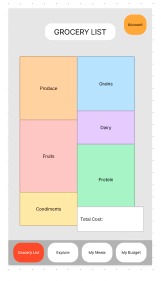
Meal Selection:

### 

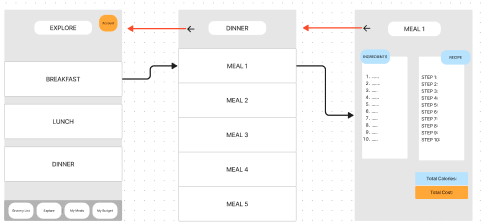
Navigation Bar:

### 

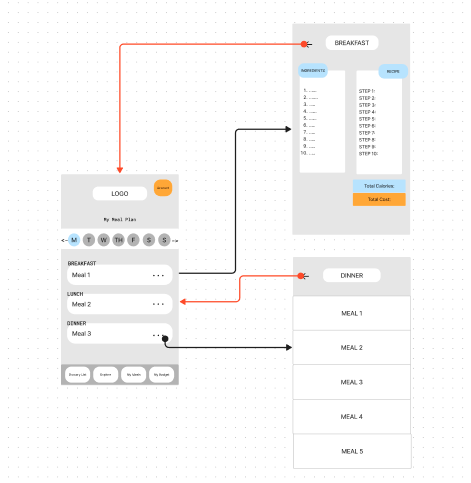
Grocery List:



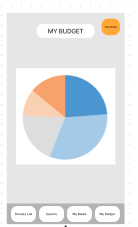
Explore:



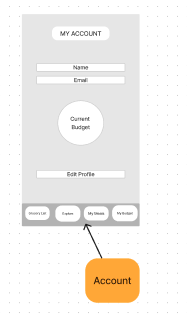
My Meals:



My Budget:



My account:



### 

#### 4.1.2 Objects and Actions

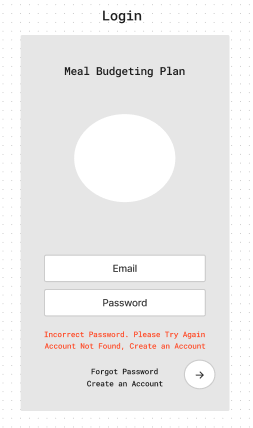
Navigation Bar:

### 

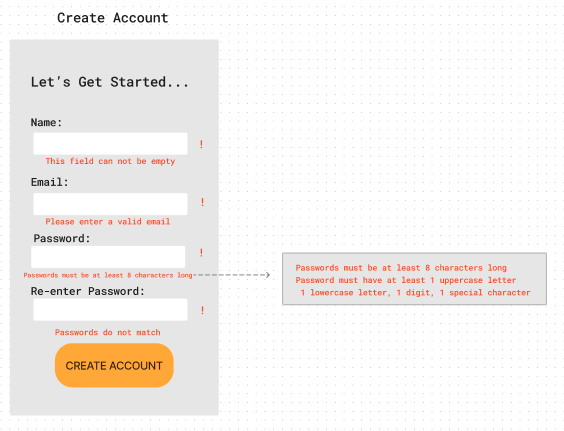
Account Button:



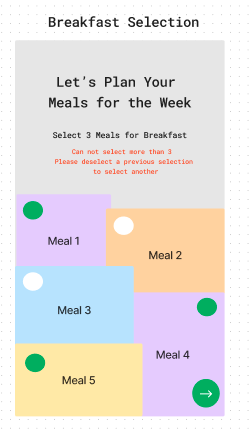
Error Message: Login



Error Message: Create Account



Error Message: Meal Selection



Edit Profile:

## 

Modify Meal:

## 

## **5.0 Additional Materials**

### **5.1 Definitions, Acronyms, and Abbreviations:**

SDD - software design description

Req - requirement

UI - User Interface

A - application

DB - database