# Health and Online Cookbook App (We-Cook)

**Team #12**

**Project Website:** https://rtumalle0518.github.io/WeCook/

**Software Engineering 14:332:452**

**Group Members:**


Randy Roque Sanchez
Randy Tumalle
Muhammad Raza
Charles Trangay
Naomie Popo
Asim Malik
Ayo Obaisi
Tyron Tucker
Memphis Chen
Asim Malik
Arshad Vohra

# Table of Contents

# 1. Section 1: *Analysis and Domain Modeling*

## 1.1 Domain Model

### 1.1.1 Conceptual Models

| **Responsibility** | **Type** | **Concept** |
|---|---|---|
| R1: Let the new user create an account and answer the initial screening questions | Action | Account Processing System<br>Survey System |
| R2: Collect the data from the server that corresponds with the correctly inserted user information | Action | Account Processing System<br>Server Communication System |
| R3: Ask first time Meal Plan users, a set of survey questions to figure out a suitable diet | Action | Survey System |
| R4: Users will be able to see and interact with the online cookbook section of the application | Action | Cook Interface |
| R5: Users will be able to see and interact with the online Meal Plan | Action | MealPlan Interface |
| R6: Users will be able to “like” and leave reviews on published recipes | Action | Online Interaction System |
| R7: Users will be able to save recipes and meal plans to their account | Action | Account Processing System |

### 1.1.2 Association Definitions

| Concept Pair | Association Description | Association Name |
|---|---|---|

| | | |
|---|---|---|
| Account Processing System ⇔Survey System | User will request to sign up for an account. Account Processing System will create account for user gathering data from the survey system | Sign Up |
| Account Processing System ⇔Server Communication System | Login request made by user. User password and username verified. User's data fetched from database | Login |
| Survey System⇔MealPlan Interface | Obtains user information to personalise meal plans | Personalisation |
| MealPlan Interface ⇔Server Communication System | Display meal plans and allow user requests to edit | Meal Plan |
| Cook Interface⇔Server Communication System | Fetches stored recipes data to display to the user. Also allows users to submit recipes. | Recipes |
| Cook Interface⇔Online Interaction System | User's request to search database and leave feedback on recipes | Recipe Reviews |

## 1.1.3 Attribute Definitions

One example attribute will be the accountInteraction. The user can enter information or the server can retrieve information that the user inputed. Another attribute will be a foodInterace. This will differentiate the viewing of the cookbook or the meal plan. The personalInteration attribute will have a save value or a review value. A save value will correspond to savings recipes to their account and a review value will correspond to rating recipes.

## 1.1.4 Traceability Matrix

| **RES** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** | **UC** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **X** | | **X** | | | | | | | **X** | | |
| **2** | | | | **X** | | | | | | **X** | **X** | |
| **3** | | | **X** | **X** | | **X** | | | | | **X** | |
| **4** | | | | **X** | **X** | **X** | **X** | **X** | **X** | | **X** | **X** |
| **5** | | | | **X** | | **X** | | | **X** | | | |
| **6** | | | | | | | | | **X** | | | |
| **7** | | | | | | | | | | | **X** | **X** |

# 1.2 System Operations Contracts

**ContractOP1:**

| | |
|---|---|
| Operation | Personalization() |
| Cross References | UC-3: Personalization |
| Preconditions: | The user has registered their account and inputted their personalized information |
| Postconditions: | The user edit/update their personalized information |

**ContractOP2:**

| | |
|---|---|
| Operation | MealPlan() |
| Cross References | UC-4: Meal Plan |
| Preconditions: | The user has entered their allergies, weight, height, goals, dietary obligations, and available ingredients. |
| Postconditions: | The user is assigned a meal plan and the user can edit their meal plan. |

**ContractOP3:**

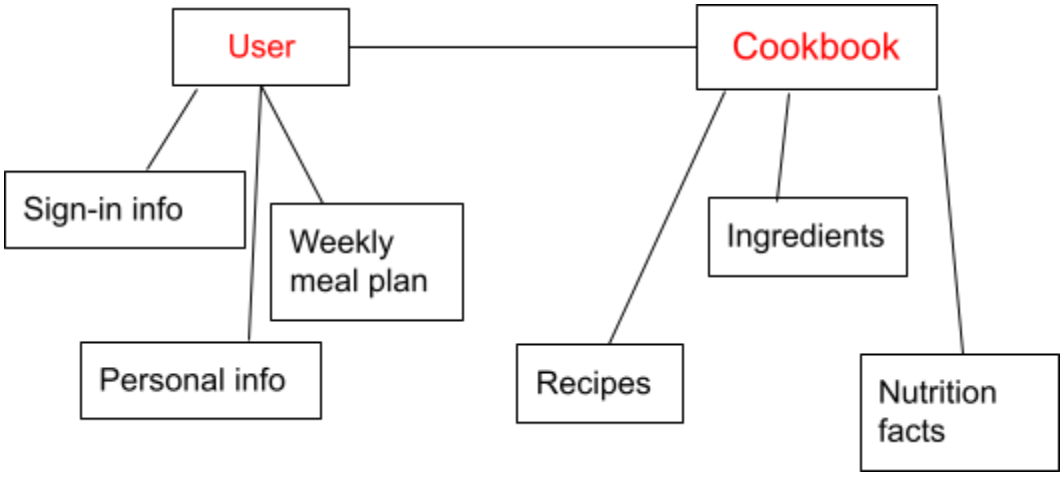
| Operation | Search() |
|---|---|
| Cross References | UC-5: Search Recipe |
| Preconditions: | The various meal recipes are accurately tagged to facilitate filtering. Also the user can utilize different parameters like calorie limit or food restrictions. |
| Postconditions: | The system selects specific meals based on filters and parameters. The user is granted with a random meal. |

**ContractOP4:**

| Operation | SubmitEntry() |
|---|---|
| Cross References | UC-6: Submit New Meal |
| Preconditions: | The user has properly quantified the ingredients in the recipe. The database should be able to hold a large amount of recipes |
| Postconditions: | The user should be able to keep track of their uploaded recipes and edit them. The recipe should be able to be viewed,saved or shared by other users |

# 1.3 Data Model and Persistent Data Storage

Our system will need to save data that needs to outlive a single execution of the system. This is because some of the data will be shared across multiple users of the application. Therefore, we will be using a data storage system to save some information generated from users to be shared across the platform as well as user's weekly plans.




To save all this information we will be using databases. Given unique identifications through their sign-in information, they will be able to access their personal information, such as their characteristics/physical attributes as well as their personalized weekly meal plan. Another instance of a database will need to be implemented for the cookbook to save all the data about recipes, ingredients, and nutritional facts. Both of these will be connected and users will be able to interact with the information from the cookbook's database.

Some SQL statements (or commands):

CREATE TABLE <table-name> (<field-name-1> <domain>,...);

INSERT INTO <table-name> (<field-name-1), <field-name-2>, …)

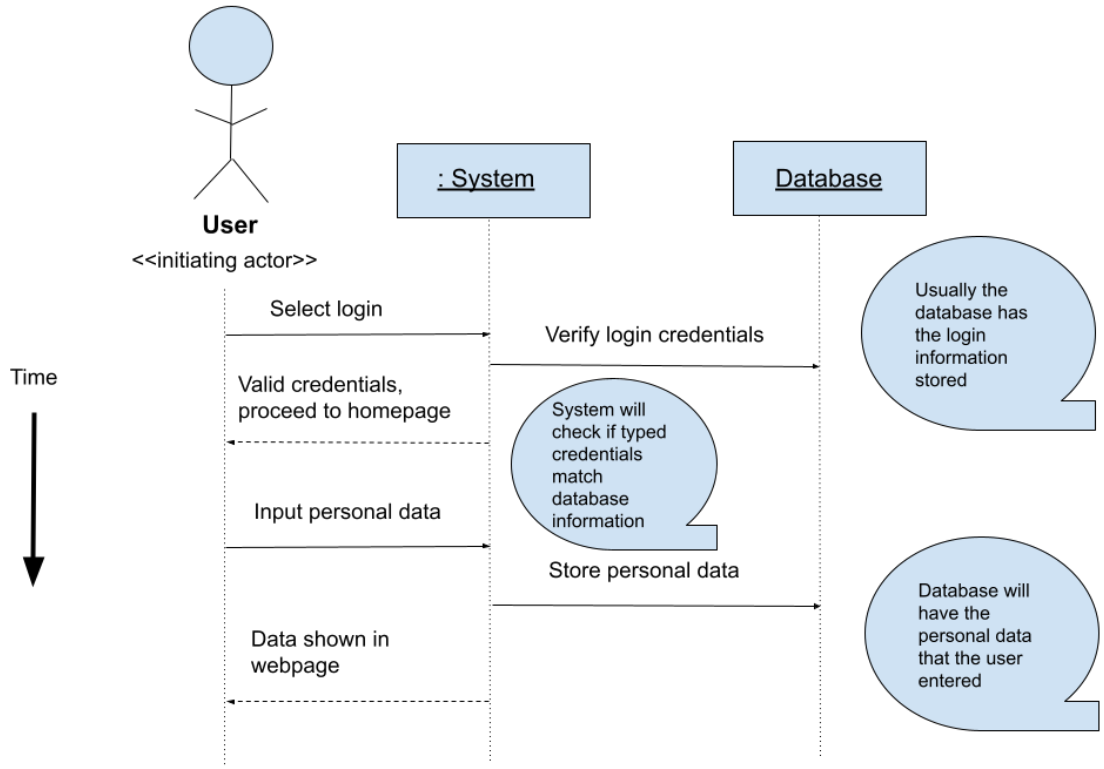
VALUES (<field-value-1>, <field-value-2>, …);

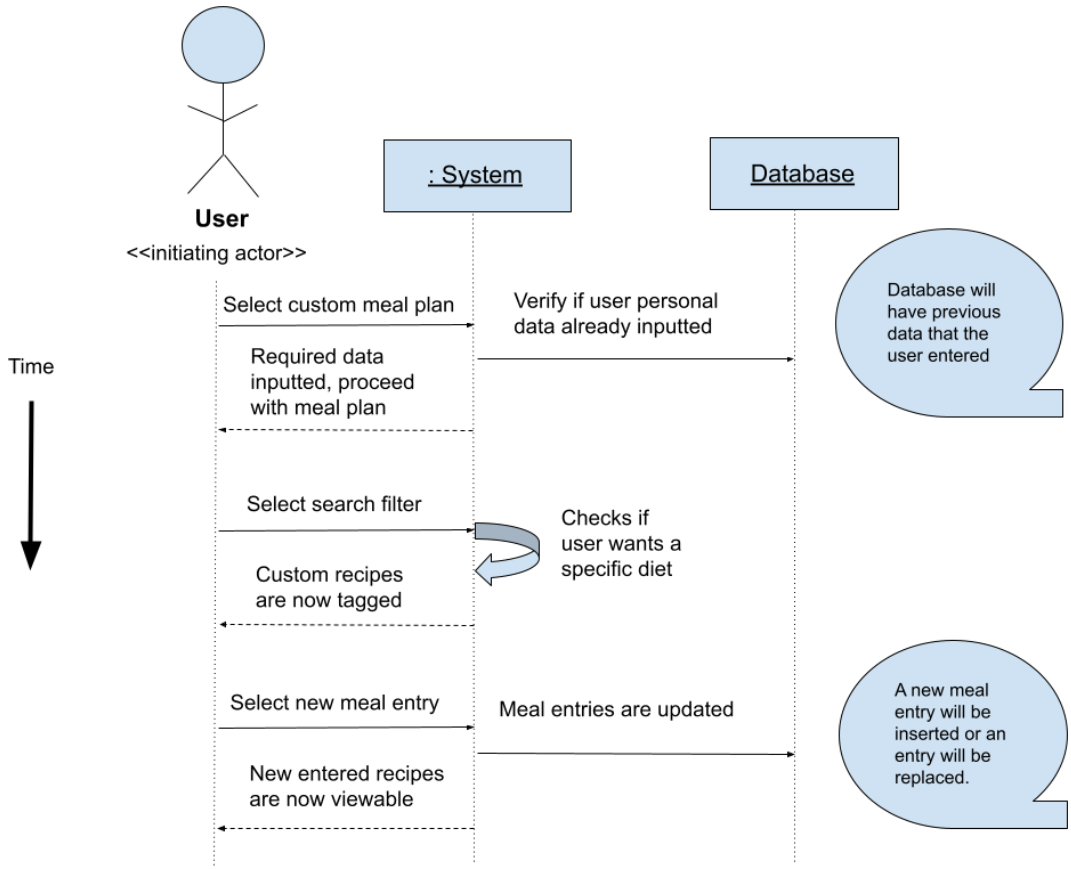
SELECT (<field-name-1>, <field-name-2>, …) FROM <table-name> WHERE <condition>;

UPDATE <table-name> SET <field-name> = <value> WHERE <condition> ;

DELETE FROM <table-name> WHERE <condition>;

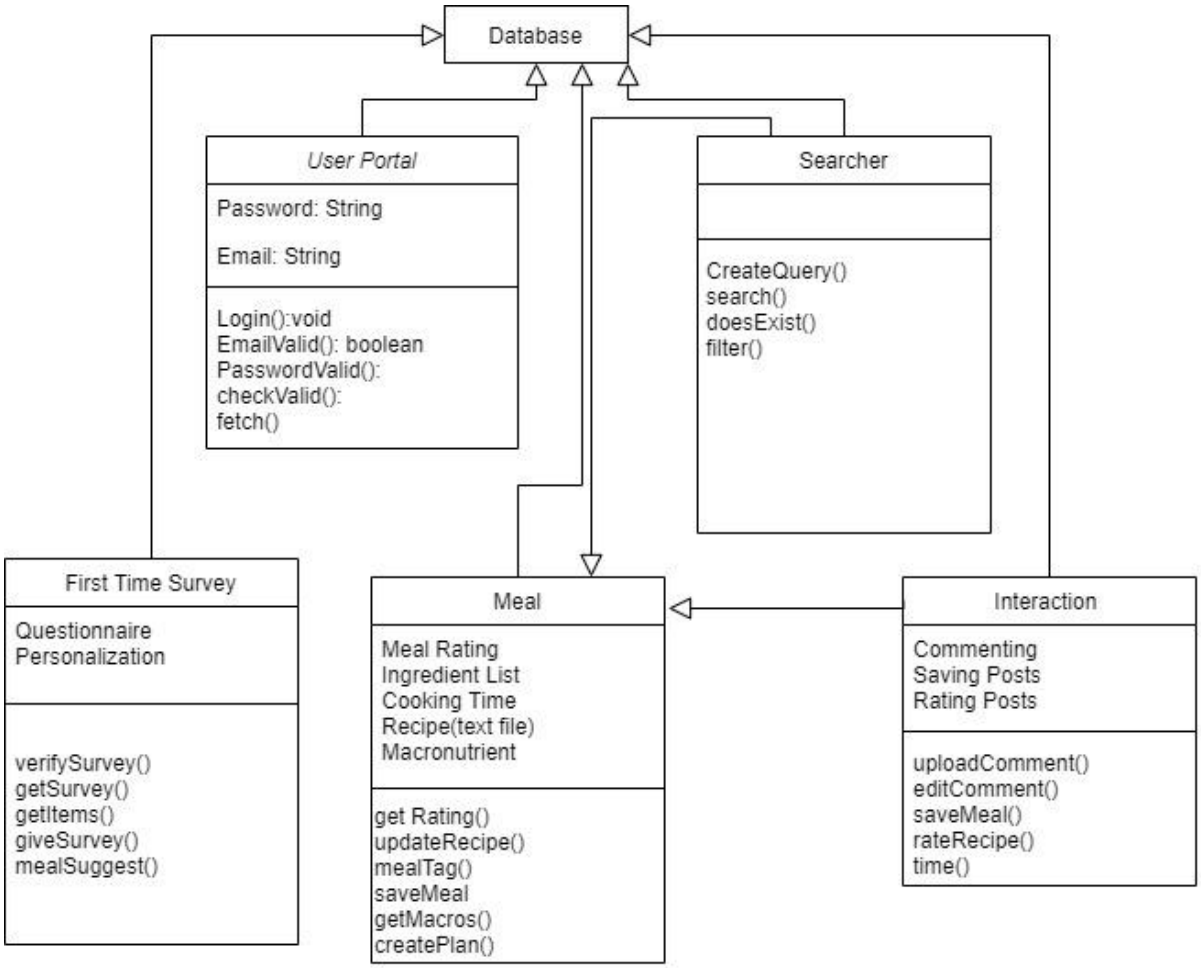
# 2. Section 2: *Interaction diagram*






# 12. Class Diagram and Interface Specification

# 12.1 Class Diagram




# 12.2 Data Types and Operation Signatures

**UserPortal Class** - This class handles the login

**Attributes:** Password: String, Email: String

**Login():** void Allows the user to effectively call on the server to login with inputted information

**EmailValid() :** boolean Checks if the email is valid Checks if email contains illegal special characters or not in proper format

**PasswordValid()** : boolean Checks if the password is valid

**checkValid()** : boolean Checks if name is <=0 Checks if password is longer than 8 Checks if passwords match

**fetch():** Checks if database console accepts the email and password and proceeds to interface

**Searcher Class**

**CreateQuery():** Requests for information from the database that matches the users search query.

**search() -** displays the result of the search query

**doesExist() -** checks if the search query matches the data inside our database and returns a boolean value

**filter() -** filters from the selection of meals

**Meal Class**

**Attributes:** Meal Rating, Ingredient List, Cooking Time, Recipe(text file), Macronutrient Information(Object with Proteins, carbs, and fats)

**Functions:**

**getRating():** retrieves rating from Meal object

**updateRecipe(string recipe):** Replaces Meal.recipe with given file argument

**mealTag():** Assigns a unique identifier to a meal that will be used for making meal plans

**getMacros():** calculates and retrieves the macros from one meal

**createPlan():** creates a meal plan with User's specific restrictions

**First Time Survey Class:**

**Attributes:** Questionnaire, Personalization

**verifySurvey() -** method that verifies user input

**getSurvey()** - method that builds the survey for each user

**getItems() -** method that holds all possible survey questions

**giveSurvey() -** method that displays the survey for each user

**mealSuggest() -** a method that exports a set of information to be processed by the meal class to suggest a premade meal plan

**Interaction Class**

Attributes: Commenting, Saving and rating Posts

**uploadComment():** user can upload a comment under a recipe

**editComment():** allows the user to change the post string of input

**saveMeal():** Saves the selected meal under your profile

**rateRecipe():** rate recipe

**time():** the time that the comment was posted

## 12.3 Traceability Matrix

| Domain Concept | Derived Class | Explanation |
|---|---|---|
| Account Processing System | User Portal Class | The user portal class is to allow any person to create a new account with your personal email and password. While also verifying that your email and password are valid for registration. This class acts as the controller of the registration. |
| Survey System | Survey Class | The survey class supports the personalization of your entire profile. With asking certain questions for the system to provide with you the best meals possible whether that is random or part of your filter as a client. This class has a UI and back end to act as a profile personalizer. |
| Cook/MealPlan Interface | Meal Class | The meal class is derived from multiple functions such as creating the meal plan based on user profile personalization. Updating recipes by replacing certain ingredients and or meals. Tracking your macro/micronutrients from each meal that is being consumed. This class acts as the heart of the user's meal plan . |
| Online Interaction System | Interaction Class | With the interaction class, its main focus is social media. Rating, editing, and uploading comments to different meals acts as a great |

| | | |
|---|---|---|
| | | tool for users to participate in expressing their personal opinions about specific meals. |

# 13. Project Management

There were no glaring issues for part 2 of the report for our team members.

The use cases that have been implemented are still in progress. Functionality part of the project will be addressed within the following week.

By the following Monday one use case should be fully functional. Two more use cases should be done by Wednesday. Next Friday should be the day where at least four cases or more should be functional. Worse case scenario would be next Saturday for four use cases if the Friday deadline was not obtainable.

# 14. References

# 15 Section 2: Algorithms and Data Structures

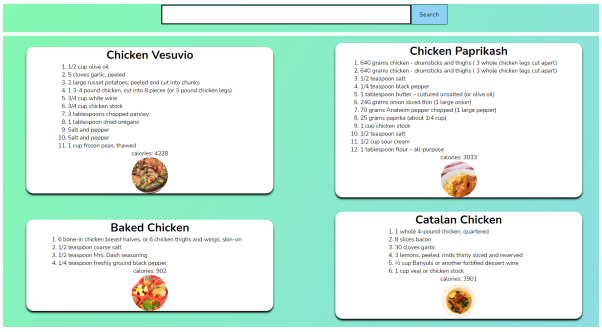
## 15.1 Data Structures

At the moment arrays will be the main source of our data structure. This was decided based on the flexibility that arrays allow. If performance becomes an issue then something like a linked list or hash tables would be used instead.

# 16. User Interface Design and Implementation

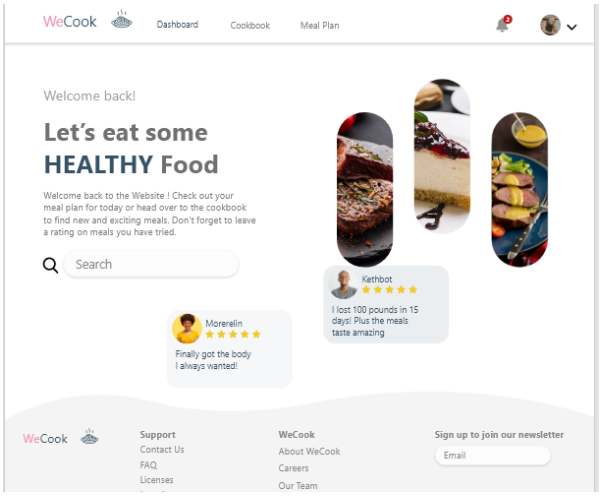

## Login/Authentication

This will be the first page that the user can see prior to logging in. If they do not have an account, they can create one. In the next iteration of this home page (WIP), we will have a “forgot password” function that will authenticate via email.



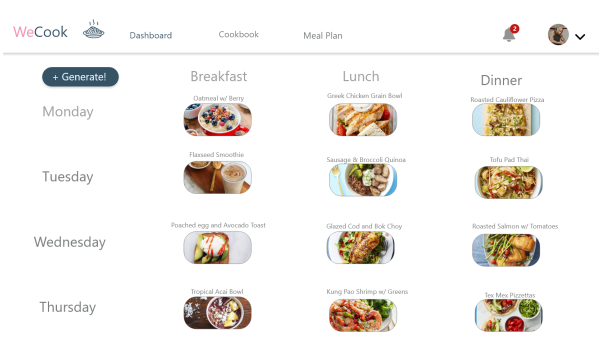

## Cookbook page

Here, users will be able to access the cookbook database. In future iterations you will be able to add meals to your meal plan, use the random meal generator, and add personal meals to the database.



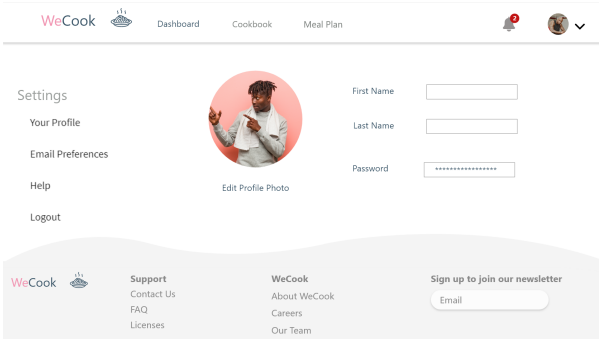

# Dashboard

The user will be able to navigate to the different pages from this screen. It is the first screen the user sees after login.




# Meal Plan

By clicking the generate button the user will be able to auto generate a meal plan for the entire week. Each photo leads to the recipe of the dish.

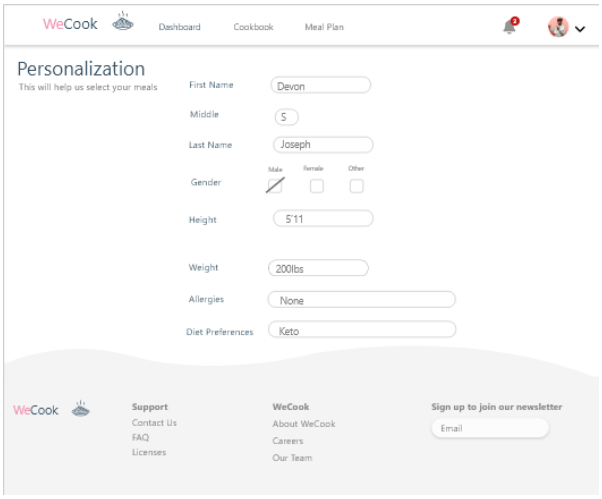



# Settings

The user will be able to edit their name and password from the settings option. They will also be able to logout.


# Recipe

Once a user clicks on a recipe, a picture, ratings, ingredients and directions will be displayed. The user will also have the option to save the recipe to their personal folder MyRecipes.




# Personalization

The user will be able to enter and edit information about their weight, height, gender, allergies, dietary preferences.

## 17. Design of Tests

Unit Testing

TC-1-: Sign up/Login Functionality

TC2-: User is able to view the Home page screen once logged in.

TC3-: Test the functionality of the Dashboard page

TC4-: Test user's ability to interact with the recipe page(edit,add,remove,etc.)

TC5-: Test user's ability to create their profile

TC6-: Test user's ability to personalize their profile

| TC | Tests | Test Coverage |
|---|---|---|
| 1 | Sign up numerous emails(school emails,gmail,etc) with different passwords | - Test if any type of email results in an issue for sign up or logging in.<br>- Test if any passwords result in an issue for sign up or logging in. |
| 2 | Home Page should be fully viewable on the user's desktop/laptop<br><br>Homepage rendering and interaction | - Formatting of the homepage should be correct in all aspects across all desktops.<br>- If homepage is rendered correctly<br>- If log out is functional<br>- If the user can access the |

| | | |
|---|---|---|
| | | dashboard |
| 3 | Dashboard | - Test if user can log out<br>- Test if logout can redirect user to login page<br>- Test if user can access cookbook page from there |
| 4 | Test if user can create new recipe, view recipe | - Test error alert if incorrect file input<br>- Test for correct file input<br>- Test if download of correct input recipe functions |
| 5 | Test if user can create profile | - Test if user can create profile with incorrect profile info (does not fit constraints)<br>  - All messed up fields<br>  - Some messed up fields<br>- Test if user can create profile that fit constraints<br>- Test if user can create profile with not every field<br>- Test if user can create profile with not all fields<br>- Test database changes before and |

| | | |
|---|---|---|
| | | after profile creation |
| 6 | Test if user can personalize their profile | - Test if user can change one then many fields of profile<br>- Test database status after profile changes are made<br>- Test removal of fields<br>- Test error statement if changes are invalid<br>  - Invalid field entries (does not fit constraints) |

# 18. Project Management

## 18.1 Merging the Contributions from Individual Team Members

To develop the report, all team members worked on a google document that helped us keep everything organized and in the same place. Before starting each report, we make sure to write the different sections needed so everyone has their space to work. After that, we create a to-do list where everyone writes their names on the sections they will work on.

Consistency is key, therefore, we make sure that we all follow the same protocols and keep everything the same for the most part. We also have meetings before the reports are due to make sure we completed all the requirements and that we are ready to submit.

## 18.2 Project Coordination and Progress Report and Plan of Work

| Use Cases | Description | Start | End | Status |
|---|---|---|---|---|
| UC-1: Registration | User creates an account for the website/User signs into website using existing account | 03/11/21 | 03/14/21 | Completed |
| UC-2: Username/Password Recovery | If the user forgot their username/password they will initiate a username/password recovery process | 03/22/21 | 03/24/21 | On-progress |
| UC-3: Personalization | User personalizes account with info like height, weight, and goals (weight loss/gain) | 03/23/21 | 03/27/21 | On-progress |
| UC-4: Meal Plan | User selects a meal plan that is generated based on their personalization, if the user does not like a meal, the user can remove it from their meal plan. The user can also submit a | 03/27/21 | 03/30/21 | On-progress |

| | | | | |
|---|---|---|---|---|
| | new meal plan with parameters, ingredients, and recipes to meal plan database through their account | | | |
| UC-5: Search filter | Users can use filters to find meals using parameters such as calorie limit, food restrictions, and ratings in Cookbook. Also, if the user does not know what to eat the randomizer will pick a meal from the meal ID pool. | 03/13/21 | 03/14/21 | Completed |
| UC-6: New Meal submissions | User submits a meal with parameters, ingredients, and recipe to the meal database through their account. | 03/24/21 | 03/26/21 | On-progress |
| UC-7: Cookbook submissions | User gets asked a series of questions about the contents inside the meal such as, if it contains any animal products. Based on these answers the app | 03/29/21 | 04/01/21 | On-progress |

| | | | | |
|---|---|---|---|---|
| | assigns a meal tag to the meal. | | | |
| UC-8: Meal Tag | A unique identifier that the system uses to identify certain types meals in both the Cookbook and Meal Plan | 03/25/21 | 03/27/21 | On-progress |
| UC-9:Recipe Interaction | Users have the ability to Rate a meal from 1-5 stars , Comment under a shared recipe to give constructive criticism, and Save recipes into their profile. | 04/02/21 | 04/04/21 | On-progress |
| UC-10: System Security | User’s email address and password are encrypted before being stored | 03/11/21 | 03/14/21 | Completed |
| UC-11: Nutrition Information | Based on users' ingredients the system will calculate the calorie count of the meal per serving, and the user can keep a track of their micros and macros | 04/05/21 | 04/09/21 | Completed |

## 18.3 Plan of Work

| Team Member | Module/Class | Integration | Integration Testing |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |