Health and Online Cookbook App (We-Cook)

Team #12

Project Website: https://github.com/rtumalle0518/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

Contribution Matrix

Task	Randy Tumalle	Tyron Tucker	Randy Roque Sanchez	Ayo Obaisi	Charles Trangay	Muhammad Raza	Asim Malik	Memphis Chen	Naomie Popo	Arshad Vohra
Customer Statement	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Glossary of Terms	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
System Requirements	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Functional Requirement Specification	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Effort Estimation	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Domain Model Analysis	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Interaction Diagram	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Class Diagram and Interface Specification	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
System Architecture & System Design	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Algorithm & Data Structure	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%

User Interface & Implementation	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Design of Tests	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
History of Work, Current Status, Future Works	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Project Management	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%

Table of Contents

Summary of Changes

Functional Table of Contents: Went more in depth on many topics such as the system architecture and design, added possible future innovations and talked about the implementation of many features such as a chatbox. Ultimately we made the table of contents easier to understand and a lot more organized and detailed.

Detailed References by topic: Added a references section with references for the following categories: problem statement, system architecture and hardware requirements. Credited the proper sources with URLs. Topics include understanding nutrition, discussing architectural styles v architectural patterns, web browsing bandwidth requirements, etc.

More specific and detailed use-cases: Showed which use cases are being covered with each test case. Also added more details regarding both use cases and test cases and how they are covered within our application.

Added more diagrams: Updated and added more diagrams and elaborated on existing ones. Showed the actual operations that will be used with the interactions, and the return types of these operations. Addressed how the different classes are interacting with one another.

Changed UI renders and visualization: Elaborated upon the navigation path through the UI pages. Also estimated worst case scenario when navigating the page for UI. Associated the UI implementation with its specific use case.

1. Customer Problem Statement

In this day and age, an average person may find it increasingly difficult to maintain a proper healthy lifestyle. Complications may occur that make it easy to forget to take care of your health. We plan to help with one of the major components of a healthy lifestyle - diet. Ignoring one's diet will have dire consequences in the near future, as we can see in the following statistics.

In the United States, 3 out of the top 10 leading causes of death are heart disease, cancer, and diabetes and most of these deaths are actually linked to poor nutrition. According to the CDC, 40% of the adult population and 19% of young people aged 2 - 19 have obesity. Being in a state of obesity puts people at risk for heart disease, specific types of cancers, and type 2 diabetes. Additionally, two of the leading causes of heart diseases are high blood pressure and cholesterol, which occur as the result of an unbalanced diet. All of these harmful, life-threatening conditions can be prevented and treated with watching how you eat and maintaining healthy nutritional habits.

People want to be the best version of themselves and having a healthy diet is an essential part in doing so. However, we understand that starting a diet is a daunting task. Nowadays, we have so many resources and information available to use that we don't ultimately know what to process. This can lead into frustration and potentially lead a user to stop their diet and take harmful steps towards their dream diet.

However, with We Cook, you can avoid these dilemmas as our app is extremely straightforward. All we need from the user is their height, weight, age, and some other varying

factors. Our app will create a platform for you to monitor your diet and learn more about nutrition and how to lead a healthy life. Our vision is to make a good healthy lifestyle a standard. We can help you start your journey to a healthier lifestyle.

The app offers a centralized platform to look up recipes within your diet for easy meal planning, store personalized recipes that you may have created and share/provide a rating on other person's uploaded recipes you have tried. Here we go into depth into the different functions:

"We-Cook" offers a convenient way to stay on track with living a healthy life while exposing you to a world-class cooking experience. During a hectic day, it can become time consuming thinking or looking for meal ideas that fit a user's diet restrictions. The app supports many of the popular dietary preferences of users such as gluten-free, pescatarian, vegetarian, and a vegan plant-based diet. To even better tailor to individuals personalised eating habits, popular allergies such as peanuts, shellfish, soy, tree nuts, eggs, wheat and milk are considered before creating a meal plan. Not taking into account allergies and dietary preferences can hinder a user's ability to properly diet, so we make sure to include these important preferences. From a user's weight and height, the app can take into account a regular calorie intake for persons looking to lose weight, gain weight or maintain their weight.

If a user is looking to utilise ingredients already in their kitchen, the app can generate simple meals that use those ingredients offering a great solution for last minute meal ideas. By taking in a user's preferences for diet and weight goals, the app will be able to provide different healthy meals that fulfill the user's dietary requirements ensuring that the user obtains key

macronutrients and calories to sustain them. Having easy to follow recipes, this functionality allows users to expand their cooking options and try new and exciting meals.

Having a place where users can share their recipes/meals they cook can make the application be very interactive and increase user activity. It can also inspire other people on their diet to keep going and living that healthy lifestyle. We live in a world where social media has become part of our daily lives and it's widely available. This technology allows users to connect with others and share things they do in their lives. Similarly, creating a social media instance in our app will make those users feel familiar with the technology and create new ways of sharing their experiences. This application will let them rate specific recipes, look for new meals to cook, as well as share their recipes they cook at home. This can also open a lot of doors to those who have a passion for cooking and would like to share the world with their gift, just like Tik Tok does for those who like to make videos about dances, trends, or specific topics.

When you first open the app, the layout would be more mostly simple at the beginning. Questions would be asked with the input of a keyboard. These questions would be simple at first and ask for things like height and weight. Later, more complex questions will be asked and that will start to form an individualized profile for the user. The profile section of the app will have most of the information that the user typed along with a button to click for recommendation. More information answered by the user will help make the recommendations more unique to the user. More detailed personalization information will be on a separate page called the recommendation page. With this information organized in the website for the user, it will help save a tremendous amount of time.

1.1 Problem Decomposition

The problem that is being addressed can be decomposed as follows

1. The average person does not keep track of nutritional information of the foods they consume.
    - Solution: Big database of meal recipes, with ingredients and nutritional information
    - Assignees: Charles Trangay, Ayo Obaisi, Randy Tumalle
2. One needs a systematic approach to planning meals for periods of time, with instructions on how to prepare these meals
    - Solution: Meal Plan functionality
    - Assignees: Asim Malik, Arshad Vohra, Tyron Tucker, Naomie Popo
3. People want to see general public opinion about certain foods -- whether they help to accomplish a goal and taste good
    - Solution: User Feedback (Ratings + Comments)
    - Assignees: Randy Roque Sanchez, Muhammad Raza, Memphis Chen

2. Glossary of Terms

Obesity - the condition of being grossly fat or overweight.

Nutrition - the process where an organism uses energy from food as fuel to support its life

Diet - the usual food and drink consumed by an organism. Can also refer to a specific restriction one is going to to reach a certain bodily goal.

BMI (Body Mass Index) - a value derived by calculating the weight of a person (kgs) divided by their height (m). Used to screen for different weight categories.

User Profile - An individualized outline for the user, based on their height, weight, dietary restrictions, allergens, etc.

Pescataraian - a person who doesn't eat any meat except fish and seafood

Gluten free - a diet that excludes food that contains the protein gluten.

Vegetarian - a person who doesn't eat meat

Vegan - a person who doesn't eat animal products - includes meat, eggs, dairy products, etc

Allergies - a condition when the immune system reacts abnormally to a potentially harmful or foreign antigen.

3. System Requirements

3.1 Functional Requirements

Identifier	Priority Weight (Low 1 - 5 High)	Requirement Description

REQ-1	4	The system should allow users to be able to create a personalizer profile with weight, height, and other parameters.
REQ-2	5	The system inputs recipes that the user creates and plans a week of food and makes sure you are not deficient in any vitamins/nutrients.
REQ-3	4	The system creates a plan for the week of food options based on the user's personal preferences
REQ-4	2	The system tells the user when you should eat based on a calendar .
REQ-5	2	The system handles a Potential Budget Feature to help you financially plan your meals.
REQ-6	3	The system features a Social Media for Cooks where you will be able to share and rate recipes.
REQ-7	3	The system shall create a grocery list based on a weekly meal plan.
REQ-8	4	The system should give dish recommendations based on the ingredients the user has available.
REQ-9	3	The system will allow users to be able to make recommendations based on

		recipes that the user creates in the app.
REQ-10	3.	The system will provide users with a given meal on a day if they have no clue of what he or she should eat based on diet, allergies, and other parameters.
REQ-11	4	The system will offer a username/password recovery process and encryption.
REQ-12	3	Meal tag and identification to avoid duplicate recipes and identify recipes based on user input.
REQ-13	3	The system will allow the user to save and edit(add/remove) recipes in their profile. And provide a filter recipe system.

3.2 Non-Functional Requirements

Identifier	Priority Weight (Low 1 - 5 High)	Requirement Description
REQ-11	5	The system should allow the user to edit or update their profile, once a profile is established.
REQ-12	2	The system will allow users to be opted to confirm his or her profile and email once registration and profile are complete
REQ-13	5	The system will notify the user if they are deficient or have excess vitamin/nutrient intake based on what has been consumed throughout each week.
REQ-14	4	The system should give ingredient information for

		each meal and filter meals based on your preferences(keto,protein, vegan, etc..)
REQ-15	3	The system running time can vary because of the various broad profiles or very in detail profiles.
REQ-16	2	The system will send tips/directions to have users navigate the program easily.
REQ-17	3	The system is able to offer an email option whenever a bug has been detected or user issue has occurred.
REQ-18	2	The system will provide a goal checklist that can be developed and followed by the user throughout each week based on user input and preferences.
REQ-19	5	The system will control the user's diet by providing ways of eating healthier, tracking the users' macros and micros, and networking with others.
REQ-20	4	The system will be more of a benefit to users if the user is more engaged and specific with personalizing their profile.
REQ-21	3	The system will mainly be desktop based.
REQ-22	4	The system should be able to detect and encrypt user data.
REQ-23	3	The system will store the encrypted data in a database.

3.3 User Interface Requirements

Identifier	Priority Weight (Low 1 - 5 High)	Requirement Description
REQ-22	5	Users should be able to create an account and he or she must list some generic information such as First Name, Last Name,

		Email, Mobile Number, Date of Birth
REQ-23	5	Users should be able to log in to his or her account, and the user needs to enter the correct email and password
REQ-24	5	Users should be able to enter his or her email to reset through a reset password button.
REQ-25	1	Users should be able to enter both email and password in order for the user to click the “login” button, otherwise the login button will be shaded a gray color to show that the button is not executable.
REQ-26	3	Users should be able to interact with a bubble-like survey for meal/diet preferences
REQ-27	3	Users should be able to login to their account and update or edit their information under the profile tab
REQ-28	5	Users should be able to choose what category of meals he or she would be comfortable with in order to view what sub meals would come next
REQ-29	4	Users should be able to verify his or her account through the system from either the user’s email or phone number.
REQ-30	4	Users should be able to have the ability to change his or hers category of meals by updating and selecting a different category in the profile tab.
REQ-31	3	Users should be able to utilize the “add meal to cookbook” button for users who want to add ingredients to the cookbook database
REQ-32	3	Users should be able to select 3 general sections that he or she can enter upon entering the program.

3.3.1 User Interface Requirements Diagram

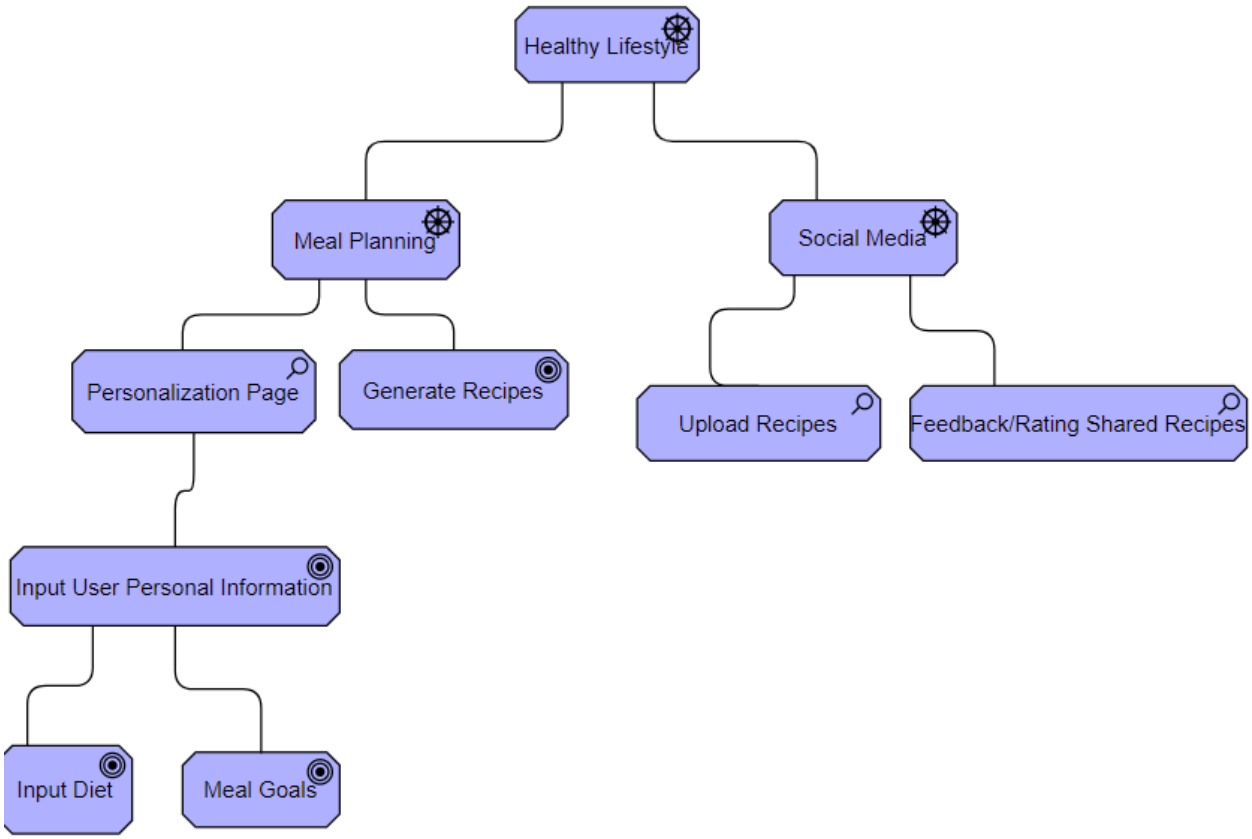


Figure 3-3 User Interface Requirements Diagram

4. Functional Requirements Specification

4.1 Stakeholders

Stakeholders are the individuals or organizations interested in the growth and development of our application. Our stakeholders are our users, organizations, and private employers, such as nutritionists and gymnasiums. Organizations and private employers will contribute to the promotion and sponsoring of our application in order to increase the amount of users. Our users

will contribute to the development and efficiency of the application through feedback and usage of the app.

4.2 Actors and Goals

Actor	**Goal**	**Use Cases**
User (Initiating)	Find meal that fits user's preferences	UC4, UC5,
User (Initiating)	Personalize utilization of website through use of an account	UC1, UC2, UC3
User (Initiating)	Meal specification through filter	UC5
User (Initiating)	Share meal recipes and plans.	UC9
User (Initiating)	Rate/Comment any meal recipe or plan.	UC9
User (Initiating)	Keep a track of nutrition information.(micros and macros included)	UC4, UC11
User (Participating)	Random meal recipe	UC5
System (Initiating)	Provide user with meals based on filter selections such as calorie counts, ratings, allergies, etc.	
System (Initiating)	The software will encrypt user information such as email and password.	UC10
System (Initiating)	Meal plan/recipe additions, removals, and edits stored in database	UC4, UC6,UC7,UC8
System (Initiating)	Calculate and display nutrition information	UC11
System (Participating)	Meal type tags and identification	UC8 UC12

4.3 Use Cases

Use Cases	Description	REQS
UC-1: Registration	User creates an account for the website/User signs into website using existing account	REQ-1
UC-2: Username/Password Recovery	If the user forgot their username/password they will initiate a username/password recovery process	REQ-11
UC-3: Personalization	User personalizes account with info like height, weight, and goals (weight loss/gain)	REQ-1, REQ-2, REQ-3,
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 new meal plan with parameters, ingredients, and recipes to meal plan database through their account	REQ-2, REQ-3, REQ-4,REQ-5,REQ-13
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.	REQ-10,REQ-13
UC-6: New Meal submissions	User submits a meal with parameters, ingredients, and recipe to the meal database through their account.	REQ-6

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 assigns a meal tag to the meal.	REQ-6, REQ-12
UC-8: Meal Tag	A unique identifier that the system uses to identify certain types meals in both the Cookbook and Meal Plan	REQ-12
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.	REQ-6, REQ-13
UC-10: System Security	User’s email address and password are encrypted before being stored	REQ-11
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	REQ-7,REQ-8,
UC-12: Recipe ID	Unique System assigned identification numbers to avoid duplicates in one day	REQ-12

4.3.1 Use Cases Diagram

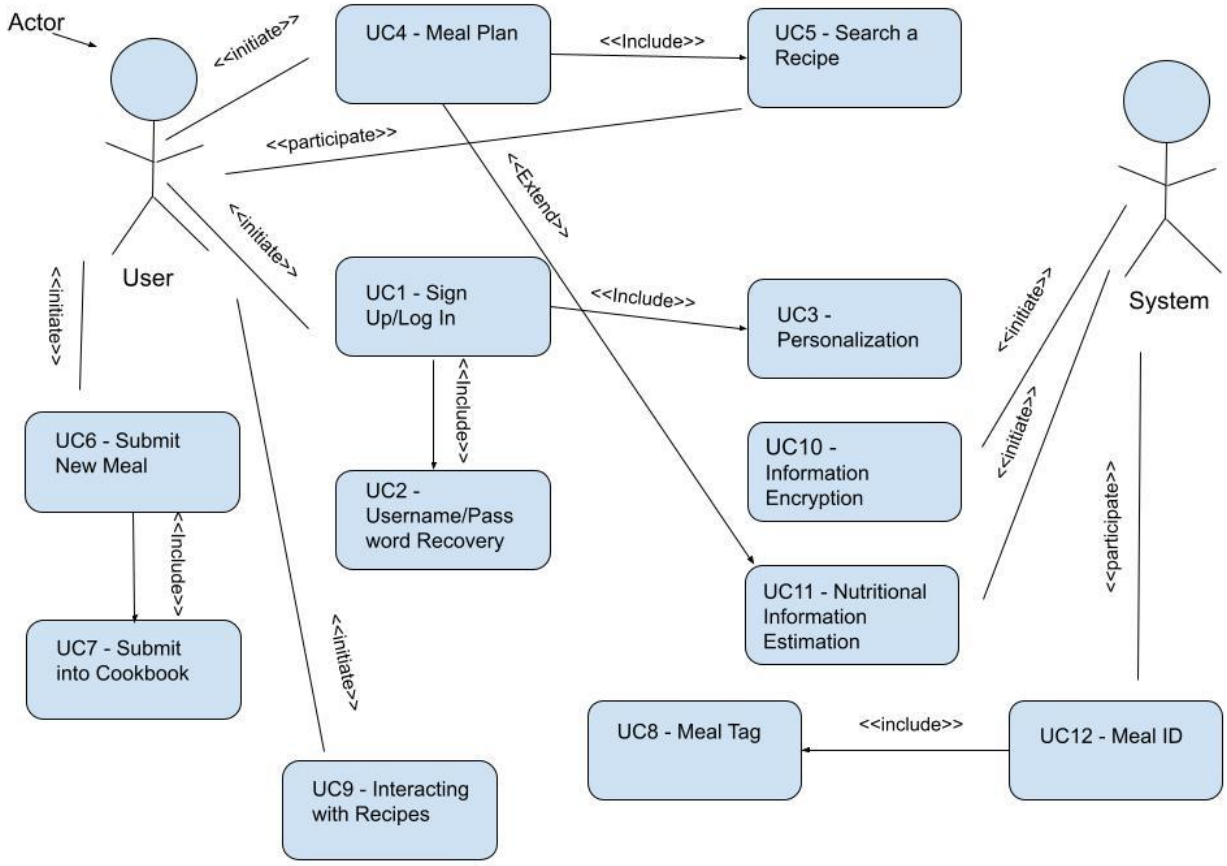


Figure 4-3 Use Cases Diagram

4.4 Traceability Matrix

REQ	PW 0-5	UC-1	UC -2	UC -3	UC -4	UC -5	UC -6	UC-7	UC -8	UC -9	UC-10	UC-11	UC-12
1	4	X		X									
2	5			X	X								

3	**4**	**X**			**X**								
4	**2**				**X**								
5	**2**												
6	**3**						**X**	**X**		**X**			
7	**3**											**X**	
8	**4**					**X**						**X**	
9	**3**												
10	**3**					**X**							
11	**4**		**X**								**X**		
12	**3**							**X**	**X**				**X**
13	**3**				**X**	**X**				**X**			
Max PW		**4**	**4**	**5**	**5**	**4**	**3**	**3**	**3**	**3**	**4**	**4**	**3**
Total PW		**8**	**4**	**9**	**14**	**10**	**3**	**6**	**3**	**6**	**4**	**7**	**3**

4.5 Fully Dressed Description

Use Case :	UC-3 - Personalization
Related Requirements:	REQ-1, REQ-11, REQ-20
Initiating Actor:	Health Enthusiast (User)
Actor's Goal:	To input personal information about height, weight and health goals (losing, gaining or maintaining)
Participating Actors:	System, Database, User
Preconditions:	The user is logged in
Postconditions:	The user is able to edit/update personalized information to stay aligned with goals
Flow of Events for	1.The user creates a profile and logs in

Main Success Scenario:	2.On the profile page the user answers the series of questions 3.The system saves the related answers for each question and utilizes it for generating meal plans 4.The user is able to see their answers and edit whenever they choose.

Use Case:	UC-4 : Meal Plan
Related Requirements:	REQ-2 , REQ-3, REQ-9 , REQ-10, REQ-19
Initiating Actor:	System, Database, User
Actor's Goal:	Generates a meal plan for the user using information obtained from the personalization page, allergies, dietary and other preferences input by the user. The user can also edit meals in the plan and submit their own meal plan.
Participating Actors:	User
Preconditions:	-The user is logged in -The user has entered their allergies, weight, height, goals, dietary obligations
Postconditions:	-The user needs to be able to edit the generated meal plan by either removing a meal or adding a meal
Flow of Events for Main Success Scenario:	1.The user clicks on the generate a meal plan button 2.The system filters and selects various meals from the database utilising the user's preferences as filters 3.The system displays the meal plan suggested 4.The user accepts the meal plan and saves it to be used

Use Case:	UC-5 : Search a Recipe
Related Requirements:	REQ-6 , REQ-14
Initiating Actor:	User
Actor's Goal:	The user should be able to search and filter through meals either

	from the system or ones uploaded by other users to find a recipe.
Participating Actors:	System, Database
Preconditions:	-The various meal recipes are accurately tagged to facilitate filtering -Recipes exist in the system
Postconditions:	-The ability to select and view a meal -The user should be able to save the recipe -The user should be able to share the recipe
Flow of Events for Main Success Scenario:	1.The user goes into the search bar and types in a tag 2.The system filters through the recipes in the data pool and lists them as suggestions. 3.The user clicks on a meal to open the recipe 4.The user saves the recipe

Use Case:	UC-6 : Submit New Meal Entry
Related Requirements:	REQ-8 , REQ-31
Initiating Actor:	User
Actor's Goal:	The user should be able to upload a new recipe for a meal into the system.
Participating Actors:	Other Users, System
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
Flow of Events for Main Success Scenario:	1.The user clicks on the submit new meal button 2.The user fills out the Ingredients and Recipe part of the template 3.The user previews and accepts that the recipe is accurately tagged 4.The system publishes the recipe and accepts it into its database 5.The recipe is saved to the my uploaded recipes folder for the user 6.The recipe comes up when searched

	7.The user / other users can view or save the recipe

4.6 Implementation of FireBase

The usage of Firebase was utilized in order to help with user functionality in the program. Firebase was the backbone for the sign up and login functionality. Along with that it helped with storing the user information through a database called firestore.In firebase, it would store the username and password of the user, with the addition of the email address and phone number that he or she provided. This is all kept in a database with Firebase. If there is ever a problem with the user's information being corrupted, the administrators or the higher authorities of the program have the power to simply delete the user's account within Firebase, and when this happens, it erases all of the user's information and everything is wiped. Firebase controls the power to allow a user to reach out for another chance at password changeability when they simply click "Forgot password" in order to reset it. Firebase also controls the way a user signs his or her account up and is stored in the database.

4.7 Implementation of Formik

Formik is a React component which works as a hook for building forms in React and it was implemented on the program. Formik was utilized on the "Add a recipe" page in order to show an easier and simpler way for the user to input his or her information in adding a new recipe to the database. On the page, the user has the ability to add a recipe title, dish type, serving size, cooking time, description, ingredients, and the directions to his or her new recipe. Formik has a unique feature to it, for example, in the serving size textbox, if a word or letters were entered in the textbox say "dog", Formik would detect it, and it would present an error saying "servings must be a 'number' type, but the final value was: 'NaN" (cast from the value "dog") in the color red, and an exclamation point with a circle around it will be brought up to show the indication of an error with the addition of the textbox being highlighted red to show indication of an error as well. In regards to this, Formik will ask the user to input numbers instead of letters. If the appropriate text is imputed and correctly placed, then the textbox would be highlighted green with the addition of a green check mark to show that what the user has imputed is valid and makes sense to be stored in the database. Also, Formik allows the user to upload a recipe image by clicking on the icon button that we have installed on the page, and will make sure it's an image file in regards to JPEG and PNG files. Once everything in the textboxes is checked green, the user will have the ability to click on the "Submit Recipe!" button which was utilized through Formik, and Formik will then send this information to the database.

5. Effort Estimation using Use Case Points

To calculate the Use Case Points for the project the following formula is used:

UCP=(UUCW + UAW) * TCF

The abbreviations above are:

UCP: Use Case Point

UUCW:Unadjusted Use Case Weight

UAW: Unadjusted Actor Weight

TCF: Technical Complexity Factor

Unadjusted Actor Weight

Actors	Description	Complexity	Weight
User	Health enthusiasts interact with the application by submitting and viewing information.	Complex	3
Web Application	The web application to host the code for functions.	Complex	3
Database	The database stores all the saved data for a user as well as holds the recipes.	Average	2

Unadjusted Actor Weight (UAW)=3+3+2=8

Unadjusted Use Case Weight

Use Cases	Complexity	Weight
UC1: Sign Up & Login	Average	10
UC2: Username/Password Recovery	Average	10

UC3: Personalization	Average	10
UC4: Meal Plan	Complex	15
UC5: Search a Recipe	Complex	15
UC6: Submit a New Meal	Average	10
UC7: Submit into Cookbook	Average	10
UC8: Meal Tag	Simple	5
UC9: Interacting with Recipes	Average	10
UC10: Information Encryption	Complex	15
UC11: Nutritional Information Estimation	Complex	15
UC12: Meal ID	Simple	5

Unadjusted Use Case Weight (UUCW)= 2(5) + 6(10) + 4(15) = 130

Technical Complexity Factor

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor
T1	Distributed System; Web-based System	2	3	6
T2	Good response and webpage loading times expected	1	1	1
T3	End-user efficiency is moderate	1	2	2
T4	Moderate to complex internal processing	1	4	4
T5	Ease of install is almost non existent, simply requires a web browser	0.5	0	0
T6	Very easy to use	0.5	1	0.5
T7	Reusable code would be nice but no prior projects in class	1	5	5

	exist			
T8	No portability concerns	2	0	0
T9	Ease of change necessary for website updates and maintenance	1	3	3
T10	Security concern for user login and personal info	1	4	4
T11	Concurrent use required	1	4	4
T12	No direct access for third parties	1	0	0
T13	Minimal User Training	1	1	1
Total				30.5

Technical Complexity Factor (TCF)= 0.6+(0.01*30.5) = 0.905

Use Case Points= (130+8)*0.905 = 124.89

Project Duration = UCP*PF

=124.89 * 28

= 3496.92hours

The PF value of 28 is used as there were no previous projects to base the productivity factor on.

With a team of 10 persons this gives us 349.692 hours per person.

6. Domain Analysis

6.1 Domain Model

Responsibility	Type	Concept
R1: Let the new user create an account and answer the initial screening questions	Action	Account Processing System Survey System

R2: Collect the data from the server that corresponds with the correctly inserted user information	Action	Account Processing System 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

6.2 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	User’s request to search database and	Recipe Reviews

Interaction System	leave feedback on recipes	

6.3 Attribute Definitions

Concept	Attribute	Attribute Description
Account	AccountInteracion	The user can enter information or the server can retrieve information that the user inputed
Viewing Interface	foodInterace	This will differentiate the viewing of the cookbook or the meal plan
Personal Interaction	saveValue	Save value will correspond to savings recipes to their account
	reviewValue	review value will correspond to rating recipes

6.4 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**

6.5 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:

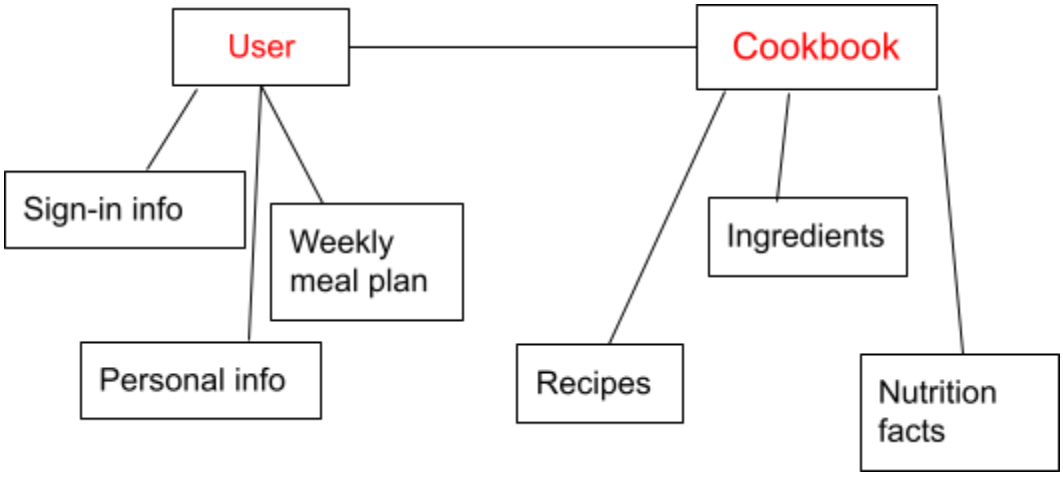
Operation	Search()
Cross References	UC-5: Search Recipe
Preconditions:	The various meal recipes are accurately tagged to facilitate filtering. Also the user can utilize different parameters like calorie limit or food restrictions.
Postconditions:	The system selects specific meals based on filters and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

6.6 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>;

7. Interaction Diagrams

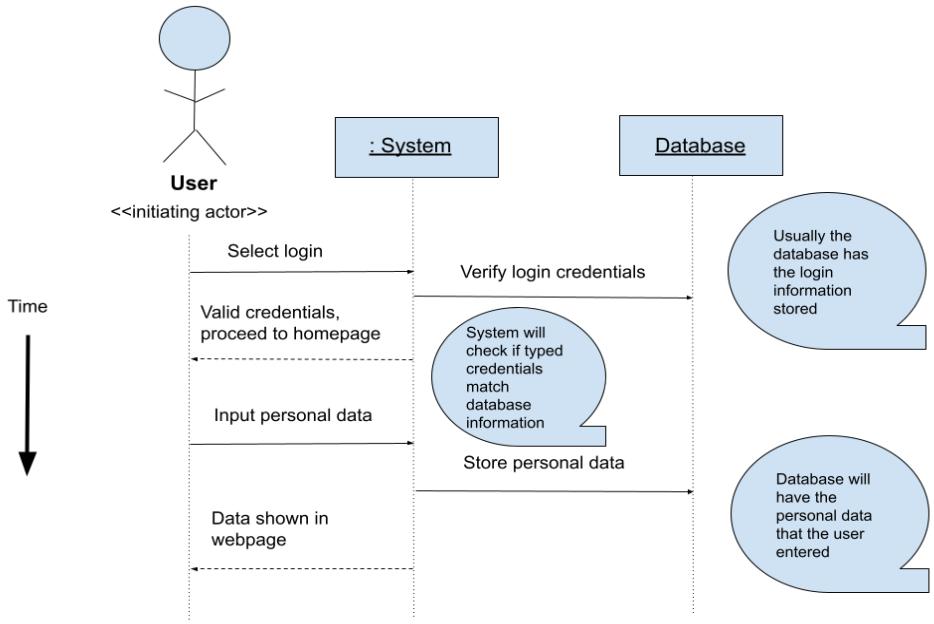


Figure 7-1 Interaction Diagram of Login/Sign Up

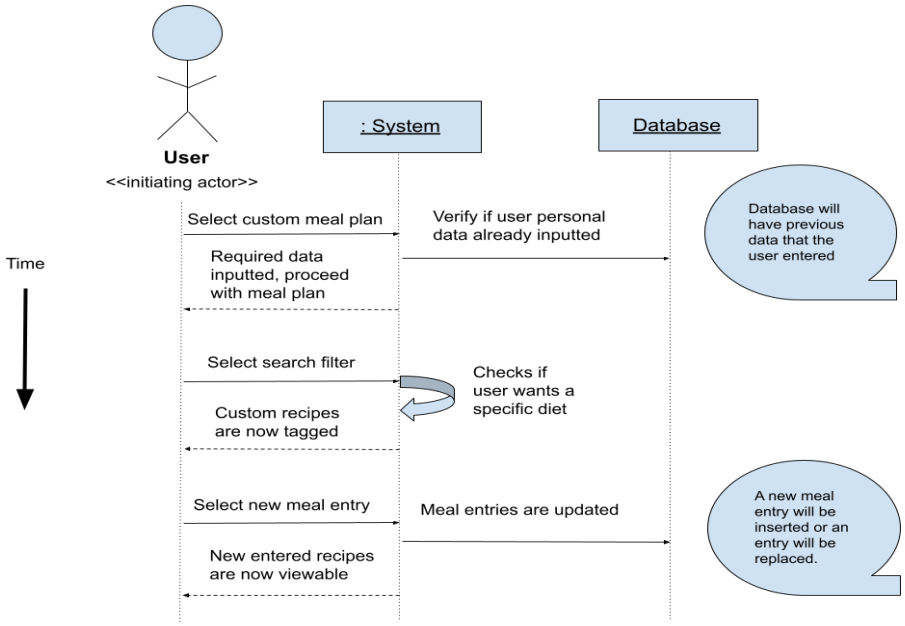
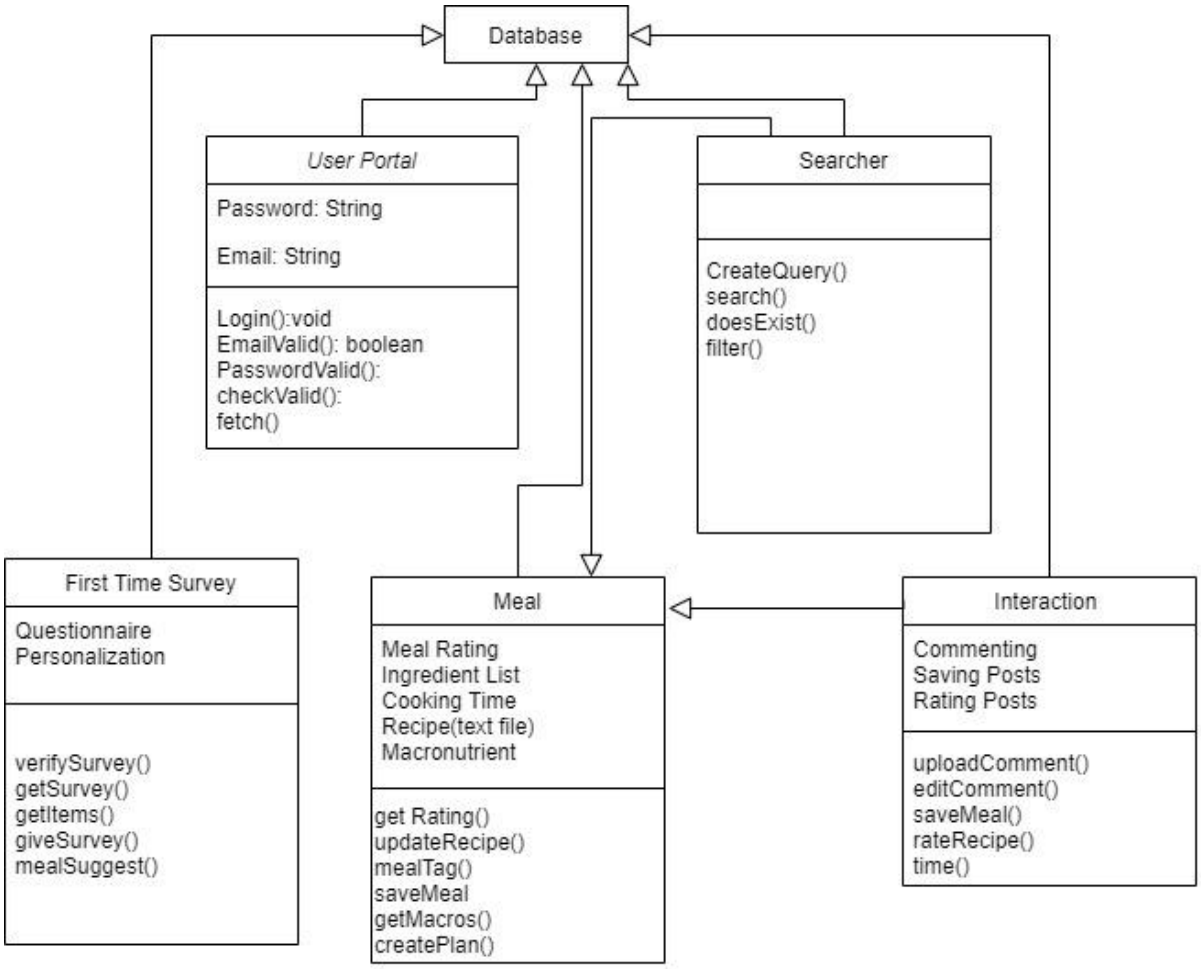


Figure 7-2 Interaction Diagram of custom meal plan

8. Class Diagram and Interface Specification

8.1 Class Diagram



8.2 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

8.3 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.

9. System Architecture and System Design

9.1 Identifying Subsystems

This subsystem will specify our login and register protocol:

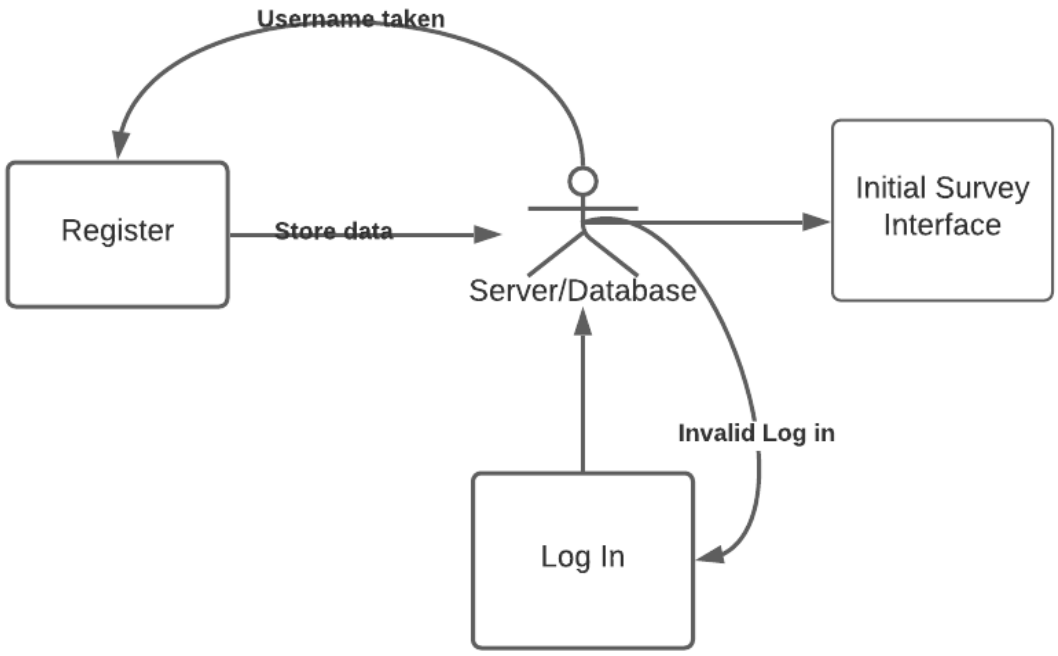


Figure 9-1 Login and Register Subsystem Diagram

This subsystem shows general idea of the path the user takes after entering the home screen as well as how the users actions interacts with the database:

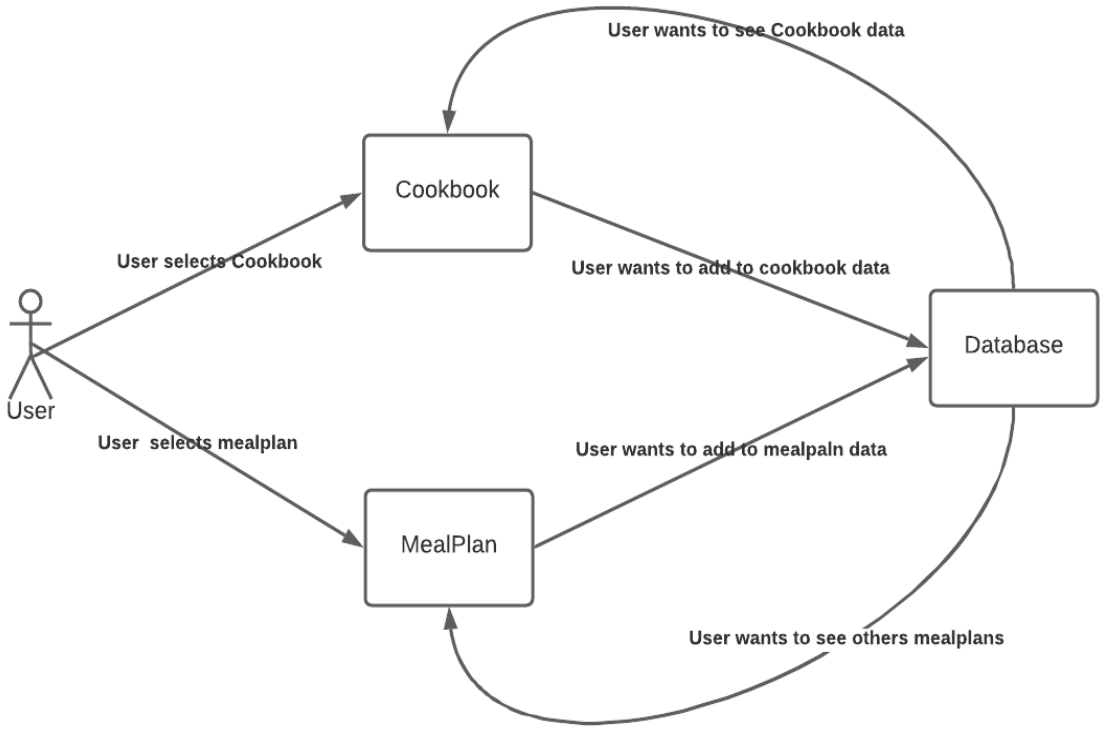


Figure 9-2 User Path Subsystem Diagram

9.2 Architectures Styles

A layered architecture style will be used for the design. The resource access layer in our design will often communicate with some database. It will need to do this to get stored customer information or the various food information. Another architecture style that will be used is an event-driven architecture style. The event producer will detect an event like a mouse click to filter information and then it will be sent back to the event consumer. The event will be processed in between the detection and retrieval stage. This process will occur in multiple parts of the project.

9.3 Mapping Subsystems to Hardware

Our project does not need hardware. Since we are making a website we would need to find a host for our website and hardware that we will need is more server side. Heroku will allow us to host our website. The website will have low to medium traffic. We have Firebase hosting all the data the user needs to read and write to. Firebase charges based on the amount of reads and writes we perform and the storage that is used. Initially our system would do a lot of writes because it needs to be populated with recipes. However after that there would only need to be reads to get information about recipes and calorie counts and it should not cost too much.

9.4 Connectors and Network Protocols

Since We-Cook is a website, it will be implemented using the HTTP (hypertext transfer protocol) network protocol. In our system, the user has to create a personal account and log onto it to access the software user interface that We-Cook provides. It is a website with a web design so we use HTTP so users can navigate the website on any computer as long as they have their login credentials.

9.5 Global Control Flow

Execution orderliness

The application can be procedure driven as well as event driven. The application can be procedure driven because users can create an account, input their personal information, have personal recommendations given to them, create a weekly meal plan, follow that plan for the week, and finally post their ratings for each meal on the social media if wanted. The application can also be event driven because users don’t necessarily need to follow the procedures mentioned in order to take advantage of what we offer. Users can create their accounts, look at

recipes, rate some of them, post their own recipes, create meal plans, everything without having to follow an order.

Time dependency

The application is of the event-response type. Every aspect of the application does not have a concern for real time. Users should be able to use all the tools available at all times and not have to depend or follow a deadline. The only possible time dependency will be the user's weekly meal planner which depends on the users themselves.

9.6 Hardware Requirements

The user will need a screen display to view the application. A desktop computer or laptop will also be needed with peripherals like a mouse and keyboard. The exact requirements will be described below:

Minimum specifications: 1280 x 720 screen resolution. 5 gigabytes of hard disk space. Bandwidth of 2 megabytes. Intel Core i3 processor(year 2012 or newer). 2 Gigabytes of ram.

Recommended specifications: 1920 x 1080 screen resolution. 8 gigabytes of hard disk space. Bandwidth of 3 megabytes. Intel Core i3 processor(year 2016 or newer). 4 Gigabytes of ram.

The minimum specification will make the application run. For a slightly more smoother experience the recommended specification is listed. The specification only pertains to the We Cook application.

9.7 Navigation Path

The navigation path of the user interface is very simple and coherent to understand. Once the user is brought to the program, he or she will be first brought to the home page. After this, the user can look through the other pages on the header tab and explore the different options if he or she would want to view. For example, if the user wanted to create a meal plan, the user can click on the recipes tab on the header and it will transport the user to the meal plan page in order to create a meal plan. However, the user will not be able to go to a different page, since the user must register or login by clicking on the “Log In” or “Sign Up” button on the top right of the program. Once the user clicks this, the user will be directed to the page where the user must log in if the user already created an account beforehand, or the signup page.

9.8 Worst Case Scenario (User)

If we were to come up with an estimate, the worst case user effort when navigating the page would have to be when the user must select a meal plan of his or choice when navigating through the meal plan page. This is because when the user decides on how he or she wants the meal to be constructed, the recipes from the API will take some time to generate in a functional order, and the algorithms will be searching for the right recipes, rather than just randomly selecting ones from the index.

10. Algorithms and Data Structures

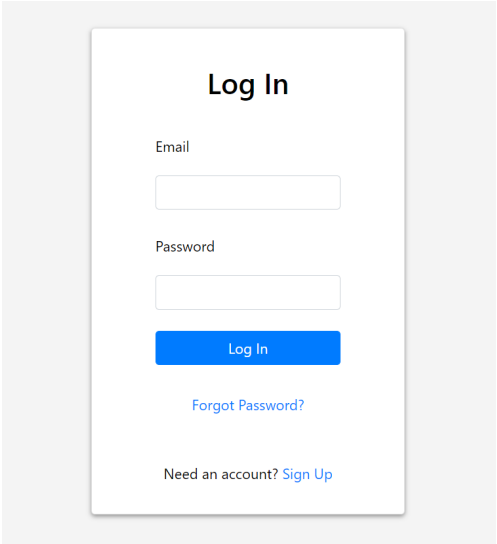
Arrays will be the main source of our data structure. This was decided based on the flexibility that arrays allow. If performance becomes an issue then something like a linked list or hash tables would be used instead.

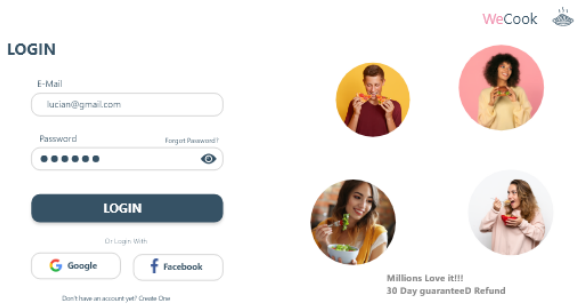
JSON files are another format that is helping us keep consistency transmitting data objects between our database, frontend, and backend. This format also allows us to easily query information requested by the user and will later on give us the ability to give meals specific IDs or tags and will also help us provide users the ability to upload their own recipes.

JSON files are retrieved by using the fetch method on an API from a website called Edamam (Edamam has a database of meals and supplementary information, such as ingredients and recipes. Their API is publicly available to developers). The JSON file is a big string with parameters and values attached to those parameters. Once the JSON file is fetched, a data.json() method is used to parse the data within the JSON file, which makes it usable by the rest of the code. Our code utilizes a JSON file retrieved from edamame -- in an example of a single meal, the queries represent individual meals, and the data such as nutritional information or ingredients can be retrieved after the data.json() method is called.

11. User Interface Design and Implementation

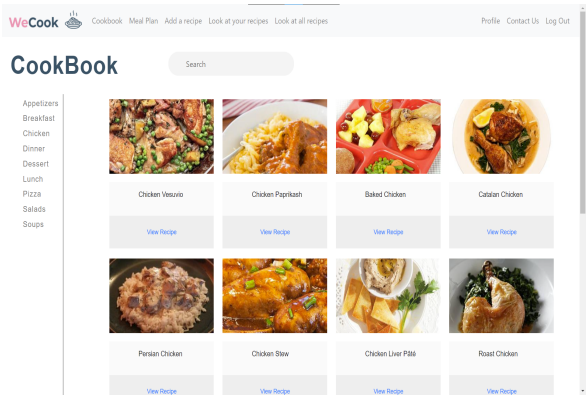
11.1 User Interface Design and Implementation





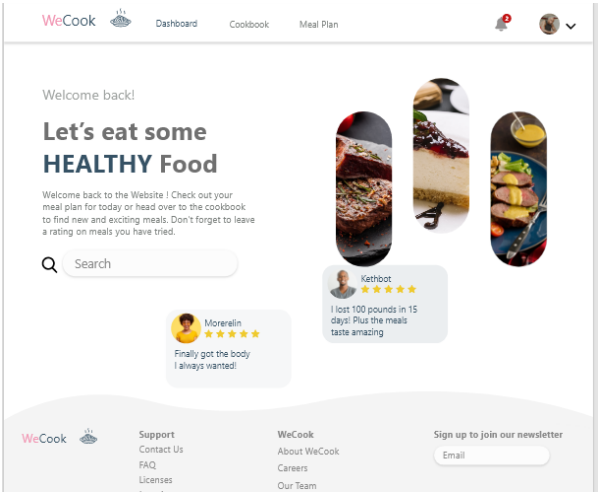
Login/Authentication

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.



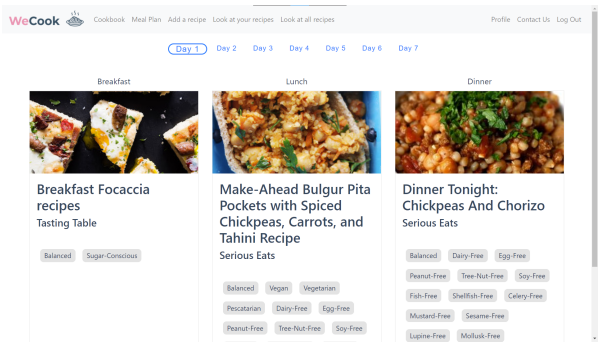
Cookbook page

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.



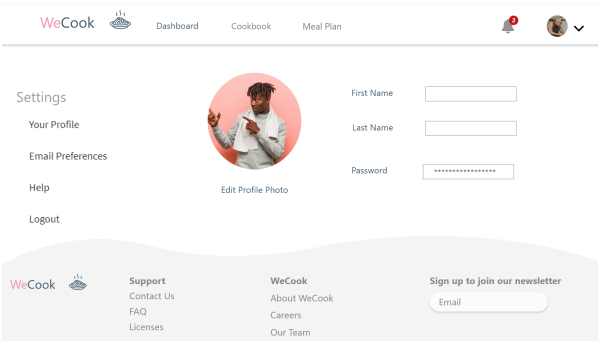
Dashboard

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



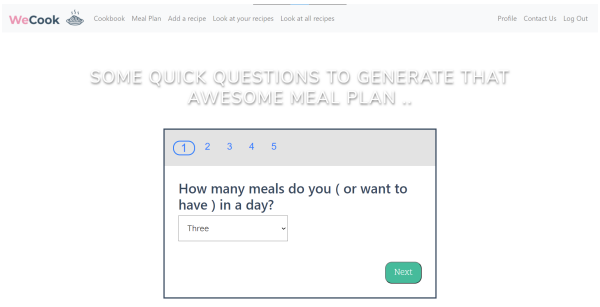
Meal Plan

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.



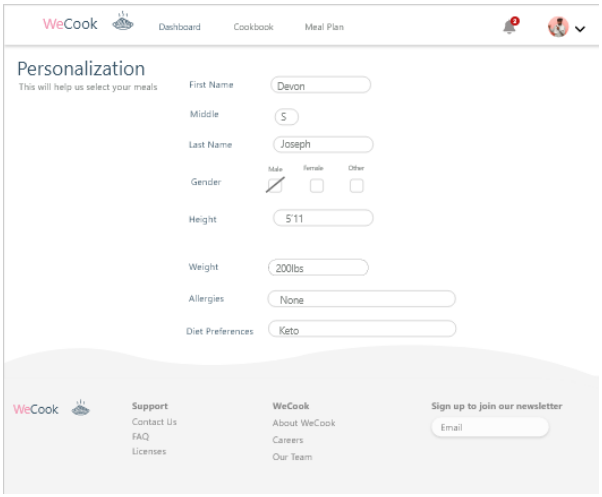
Settings

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



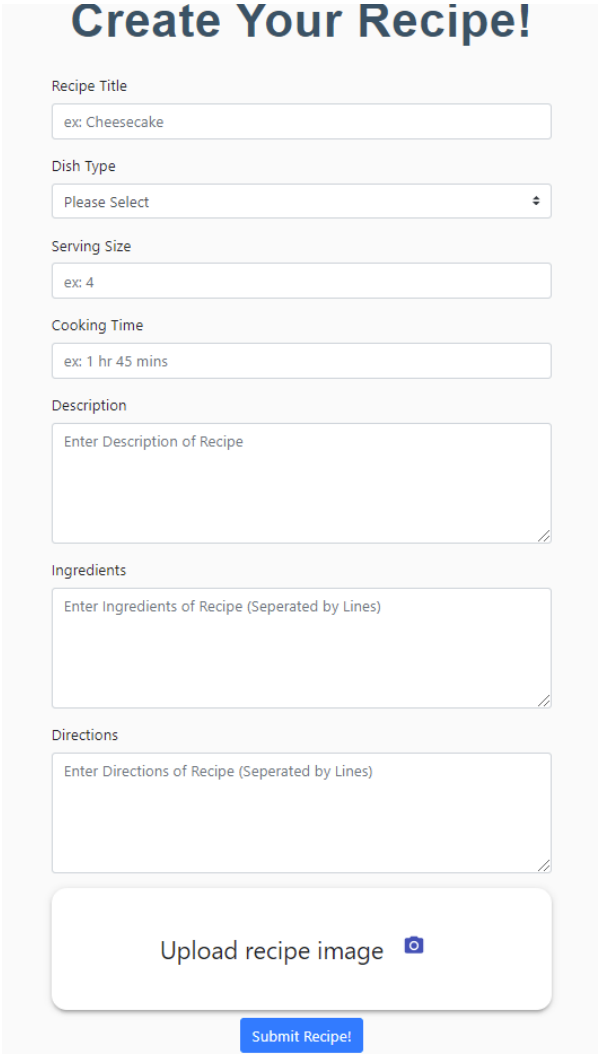
Survey

The user will be able to answer a set of questions and get a specialized meal plan



Personalization

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



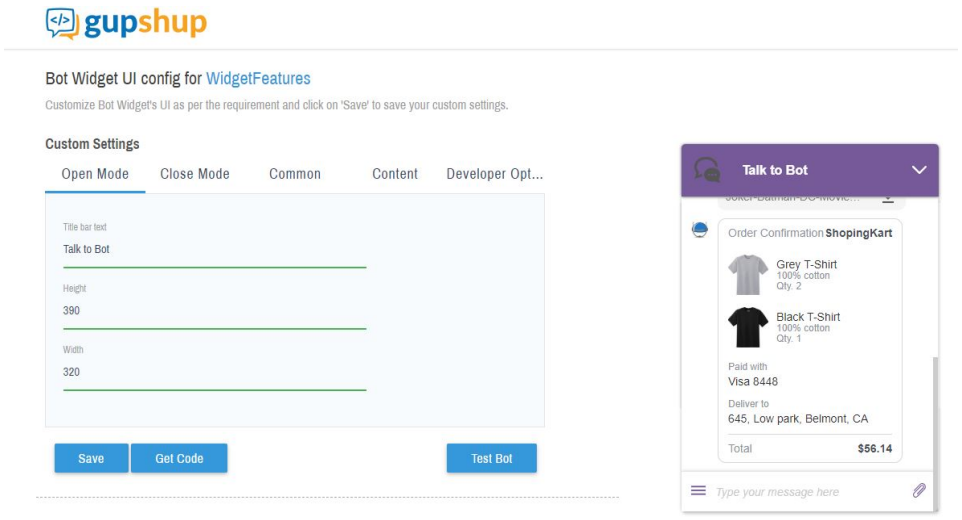
Create/View a Recipe

The user will be able to enter and add information about their own special recipes. We also created a separate page to see all user added recipes.

11.2 Implementation of ChatBot

The ContactUs page that was implemented in the program was a useful utility for the user to gain contact with the higher authorities in case he or she needed help, whether it would be in regards to recipes, nutrition, or something else that they would love to have more information on, or if they generally find any problems in the application and don't know how to go about solving it. However, a different solution for this that could have been implemented which would be the idea of a ChatBot. The ChatBot would have been implemented on the program on the bottom right corner of the program. There would be CSS on the ChatBot which would show the ChatBot as a

small rectangular tab which the user can click and then the whole ChatBot would pop up. In the ChatBot, it would first ask the user “Do you need help with anything?”, and the ChatBot will need a response from the user in order to navigate to the next command. The user will be able to answer and click on the “Yes" or “No” buttons in order to continue further. The ChatBot will keep asking the user questions such as “Do you need help finding a lower calorie meal?” or even “Do you need help finding a more satisfying recipe?”. These are all questions which would help solve the user’s problems most of the time. If there is a question that the ChatBot is unable to help with, then it would simply ask or even tell the user to personally send an email to the higher authorities, and that they would get back to the user in a reasonable time. Also, the ChatBot should be running for multiple users at the same time; therefore, users would not have to wait in line or queue in order to get help/support. If the company was generating a lot of money from this program, then they could even hire a representative for live customer support using the same ChatBot, in order to help the user in a more efficient and straight up way. The benefits of using a ChatBot is first, it is ultimately great for buisness. The ChatBot increases customer engagement, significantly cuts costs, automates multiple business processes, available around the clock, improves customer satisfaction, and handles multiple tasks simultaneously. With a ChatBot, we can program it to reply within a certain time if the range of questions is in its capability and we can have full control over the ChatBot’s action, where not the same can be said if a real person were to answer the user’s emails. An example of what the ChatBot would look like is shown below.



12. 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. - 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 Homepage rendering and interaction	- Formatting of the homepage should be correct in all aspects across all desktops. - If homepage is rendered correctly - If log out is functional - If the user can access the dashboard
3	Dashboard	- Test if user can log out - Test if logout can redirect user to login page - 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

		- Test for correct file input - 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)
		  - All messed up fields
		  - Some messed up fields
		- Test if user can create profile that fit constraints
		- Test if user can create profile with not every field
		- Test if user can create profile with not all fields
		- 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 - Test database status after profile changes are made - Test removal of fields - Test error statement if changes are invalid - Invalid field entries (does not fit constraints)

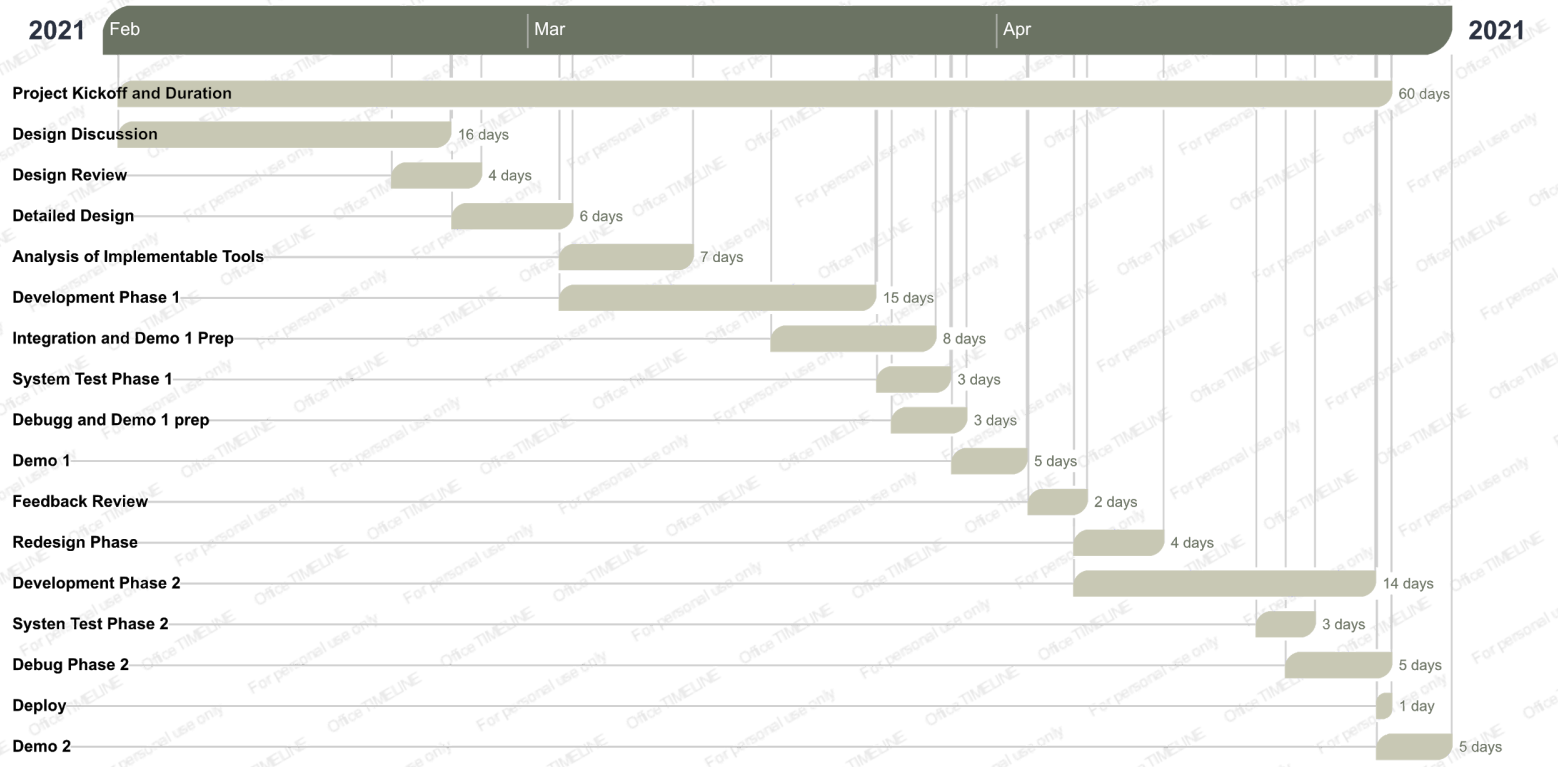
13. History of Work, Current Status, and Future Work

13.1 History of Work

In reports #1 and #2, Team #12 had set a lot of different goals that we wanted to achieve with this project. As we have dived deeper and closer to finishing the project, we have realized that some adjustments needed to be made.

So far, Team #12 has a fully functional and interactive website. In this website, users can create an account to use for login, look for all kinds of recipes, or take a personal survey that will be able to give you recommendations. However, at the beginning of the project we simply started brainstorming what kind of app we wanted to develop, and after a couple days, we all agreed to work on a cooking app. Giving that this idea is not overall completely new, we still wanted to implement some aspects that will make it unique. Some of these features are the social media aspect, so you can rate and upload recipes, and create meal plans for users, so they can experience new foods within their dietary restrictions. Putting all of these features together as well as a friendly and pretty user interface will give users a great experience that will make them want to keep coming back to our application. Around report #2 is when we started to put our ideas into practice and created our application, this gave us a perspective of how achievable goals were. Getting ideas from paper to actual code is harder than what it looks but this didn't stop us from still doing the best we can and so far we have achieved a lot of the goals. Another great source of feedback was demo #1 because it provided us with a live session to showcase our project and immediately receive feedback on things that we should retake a look at or new implementations that we could add. This helped keep our project focused on the initial proposal and challenged us to do things outside-the-box. With some of the main features still on the works, we believe that it's possible to complete this project and we will work hard to do it before the deadline. With that said, everyday we are implementing new features, adding new tools, making the website more user-friendly, and increasing performance overall, giving us all an opportunity to work on different aspects of the development and documentation of this application.

Below is the gantt chart diagram which enlists detailed task breakdown and the corresponding estimated start date and end date:



13.2 Current Status

The current status/completed parts of our application:

- Implemented website to hold all the tools
- Implemented login and signup to give users personal accounts
- Implemented a cookbook API that provides all kinds of recipes
- Created a user friendly dashboard interface for easy access
- Implemented data encryption for information saved in database
- Created a survey to help give personalized recommendations to users

13.3 Future Work

The future plans to complete for this project are the following:

- Implement user reviews on dishes/recipes
- Implement a like and favorite system, so the user can store Recipes they enjoy and liking a recipe will filter all bad recipes.
- Implement a comment section under each recipe
- Implement a Chatbot
- Optimize the process for adding recipes (autofill)

14. References

Problem Statement:
[1]https://www.cdc.gov/chronicdisease/resources/publications/factsheets/nutrition.htm#:~:text=Overweight%20and%20Obesity&text=In%20the%20United%20States%2C%2019,system%20%24147%20billion%20a%20year.

[2]https://www.medicalnewstoday.com/articles/282929

System Architecture:

[3]https://herbertograca.com/2017/07/28/architectural-styles-vs-architectural-patterns-vs-design-patterns/

[4] https://www.redhat.com/en/topics/integration/what-is-event-driven-architecture

Hardware Requirements:

[5]https://gobrolly.com/amount-data-and-bandwidth-required-web-browsing/#:~:text=How%20much%20data%20does%20web,download%20in%20seconds%20or%20less