

---

# **Software Requirements Specification**

**for**  
**ChompTrack**

**Version 1.0 approved**

**Prepared By**  
**Lee Choonggi**  
**Daven Poon Siu Wing**  
**Sheng Jia-Qi**  
**Yeo Jie Sheng**  
**Ng Jia Xin**  
**Chester Chiow Yong Xun**

**Nanyang Technological University**  
**17/11/2024**

# Table of Contents

|   |           |
|---|-----------|
| <b>1. Introduction</b>                              | <b>1</b>  |
| 1.1 Purpose   | 1         |
| 1.2 Document Conventions                            | 1         |
| 1.3 Intended Audience and Reading Suggestions       | 1         |
| 1.4 Product Scope                                   | 2         |
| 1.5 References                                      | 2         |
| <b>2. Overall Description</b>                       | <b>3</b>  |
| 2.1 Product Perspective                             | 3         |
| 2.2 Product Functions                               | 4         |
| 2.3 User Classes and Characteristics                | 6         |
| 2.4 Operating Environment                           | 6         |
| 2.5 Design and Implementation Constraints           | 7         |
| 2.6 User Documentation                              | 7         |
| 2.7 Assumptions and Dependencies                    | 8         |
| <b>3. External Interface Requirements</b>           | <b>9</b>  |
| 3.1 User Interfaces                                 | 9         |
| 3.2 Hardware Interfaces                             | 24        |
| 3.3 Software Interfaces                             | 25        |
| 3.4 Communications Interfaces                       | 26        |
| <b>4. System Features</b>                           | <b>27</b> |
| 4.1 Registration                                    | 27        |
| 4.2 Create Profile                                  | 28        |
| 4.3 View suggested estimated nutrition daily intake | 29        |
| 4.4 Login   | 30        |
| 4.5 My Plan   | 31        |
| 4.6 Discover  | 34        |
| 4.7 Grocery List                                    | 38        |
| 4.8 History   | 39        |
| <b>5. Other Nonfunctional Requirements</b>          | <b>42</b> |
| 5.1 Performance Requirements                        | 42        |
| 5.2 Safety Requirements                             | 42        |
| 5.3 Security Requirements                           | 42        |
| 5.4 Software Quality Attributes                     | 43        |
| 5.5 Business Rules                                  | 43        |
| <b>6. Other Requirements</b>                        | <b>45</b> |
| 6.1 Internationalisation Requirements               | 45        |
| 6.2 Legal Requirements                              | 45        |
| 6.3 Reuse Objectives                                | 45        |
| 6.4 Accessibility Requirements                      | 45        |

|  |           |
|--|-----------|
| <b>Appendix A: Glossary</b>              | <b>46</b> |
| <b>Appendix B: Analysis Models</b>       | <b>50</b> |
| <b>Appendix C: To Be Determined List</b> | <b>57</b> |

## Revision History

| Name | Date | Reason For Changes | Version |
|------|------|--------------------|---------|
|      |      |                    |         |
|      |      |                    |         |

# 1. Introduction

## 1.1 Purpose

This document details the software requirements for ChompTrack, version 1.0. ChompTrack enables users to generate tailored recipes based on their budget, nutritional needs, and preferred cuisines, offering features that support personalized meal planning and cost management. The scope of this SRS includes the core functionalities of ChompTrack, such as including user account management, meal scheduling, and automatic recipe and grocery list generation. It also covers integration with the external API for recipe suggestions and grocery prices to enhance the user experience. This SRS serves as a foundation for the development, testing, and maintenance of the system, ensuring alignment with project's objectives.

## 1.2 Document Conventions

This document follows standard documentation conventions.

- Font Styles: ‘Times New Roman’
- Font Size: ‘18, Bold’ for main header, ‘14, bold’ for subsection header, ‘11’ for body text
- Highlighting: Bold for key terms
- Requirement Prioritisation: Higher-level requirements have inherited priority levels, cascading down to detailed requirements. Each requirement statement is explicitly labeled with a priority level (High, Medium, Low).
- Requirement Identification: Each requirement is uniquely identified for easy reference

## 1.3 Intended Audience and Reading Suggestions

This document is intended for various stakeholders involved in the development and management of the ChompTrack application, version 1.0. All readers to read begin with the introduction to understand the document's purpose and product scope, providing essential context. The primary audiences include:

- **Project Managers and Stakeholders:** Refer to overall description for a high-level overview of the ChompTrack system.
- **Developers:** Refer to External Interface Requirements for technical specifics needed for system interoperability and System Features for detailed descriptions of system's functionalities.
- **Testers:** Refer to System Features to develop functional test cases. Additionally, review the Other Nonfunctional Requirements to ensure that performance, security, and other quality standards are met during testing.

- **User Experience Designers:** Refer to the User Interface portions within the External Interface Requirements to gain insight into user interactions with system.
- **Technical Writers:** Entire document is relevant, with emphasis on User Documentation requirements and Assumptions and Dependencies to support the creation of clear and accurate user guides and help documentation.
- **Marketing Staff:** Refer to System Features and Other Nonfunctional Requirements to familiarize themselves with ChompTrack's main functionalities for promotional and user engagement efforts.

This document is organized to guide readers through a logical flow, from a high-level overview to detailed technical requirements, with each section building on the previous. Readers are encouraged to review the document in full for a cohesive understanding of system objectives and requirements.

## 1.4 Product Scope

ChompTrack is a meal planning web application designed to address the challenges of maintaining balanced nutrition and budgeting within Singapore's fast-paced lifestyle. Many individuals struggle to track their diets and meet daily nutritional requirements, often opting for convenience foods that lack essential nutrients. Rising grocery costs further complicate budgeting efforts, as prices fluctuate frequently. Hence, the goal of ChompTrack is to simplify meal planning by offering a wide range of easy-to-follow recipes, tracking nutrition intake which helps users meet dietary requirements and staying within budget by reflecting current grocery prices. This aligns with broader corporate goals of promoting healthier living and enhancing daily life through technology.

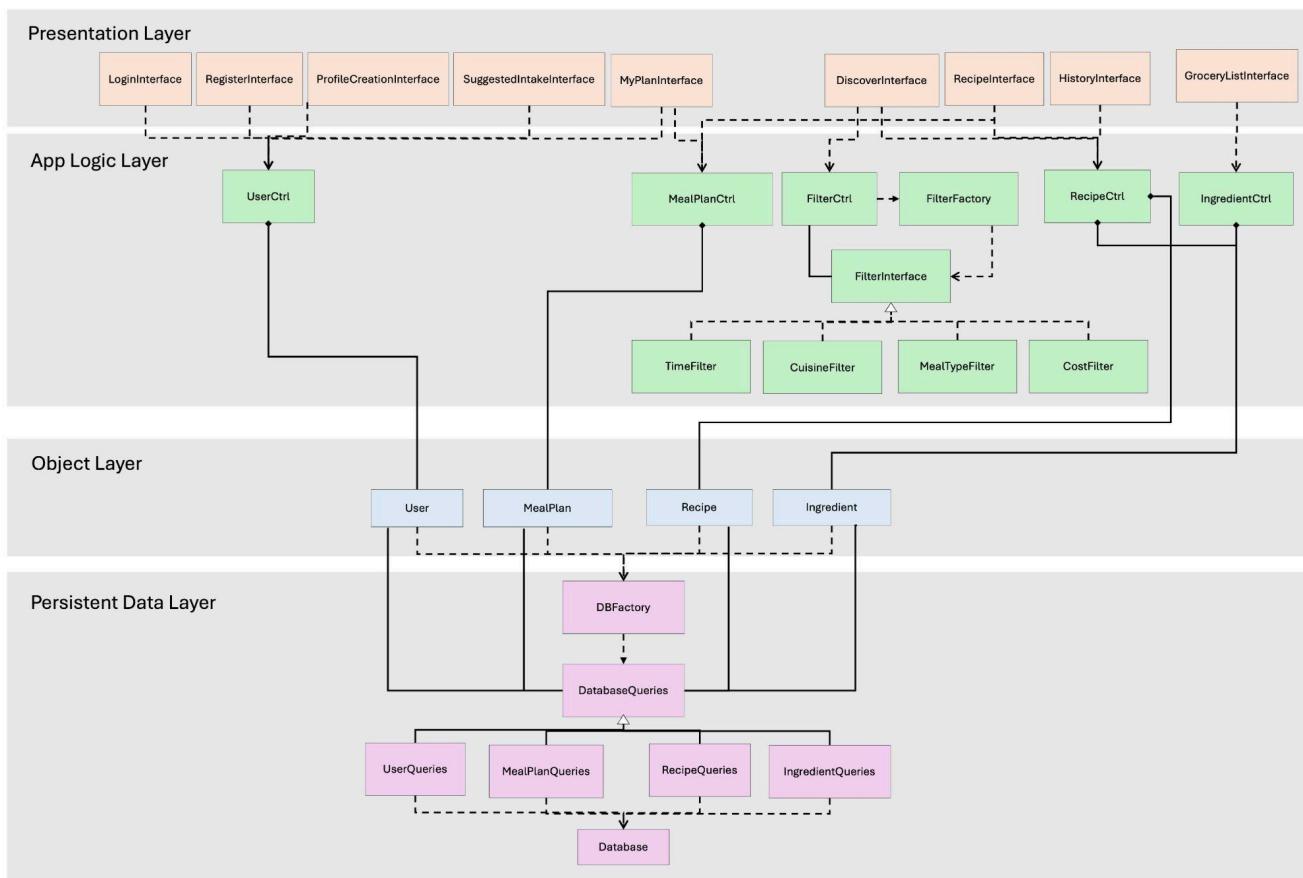
## 1.5 References

- I. Spontacular API: <https://spoonacular.com/food-api>
- II. Flask : <https://flask.palletsprojects.com/en/stable/>
- III. MySQL Workbench: <https://www.mysql.com/products/workbench/>

## 2. Overall Description

### 2.1 Product Perspective

This SRS outlines the requirements for ChompTrack, a standalone product developed to tackle issues related to meal planning, nutrition tracking, and food budgeting. It is not a follow-on member of a product family nor a replacement for existing systems, but instead a novel solution to enhance dietary habits and streamline food budgeting. While ChompTrack functions as an independent platform, it integrates with other technologies through APIs, specifically for recipe generation. A system architecture diagram (Figure 1.) is provided below to illustrate the major components, subsystem interconnections, and external interfaces.

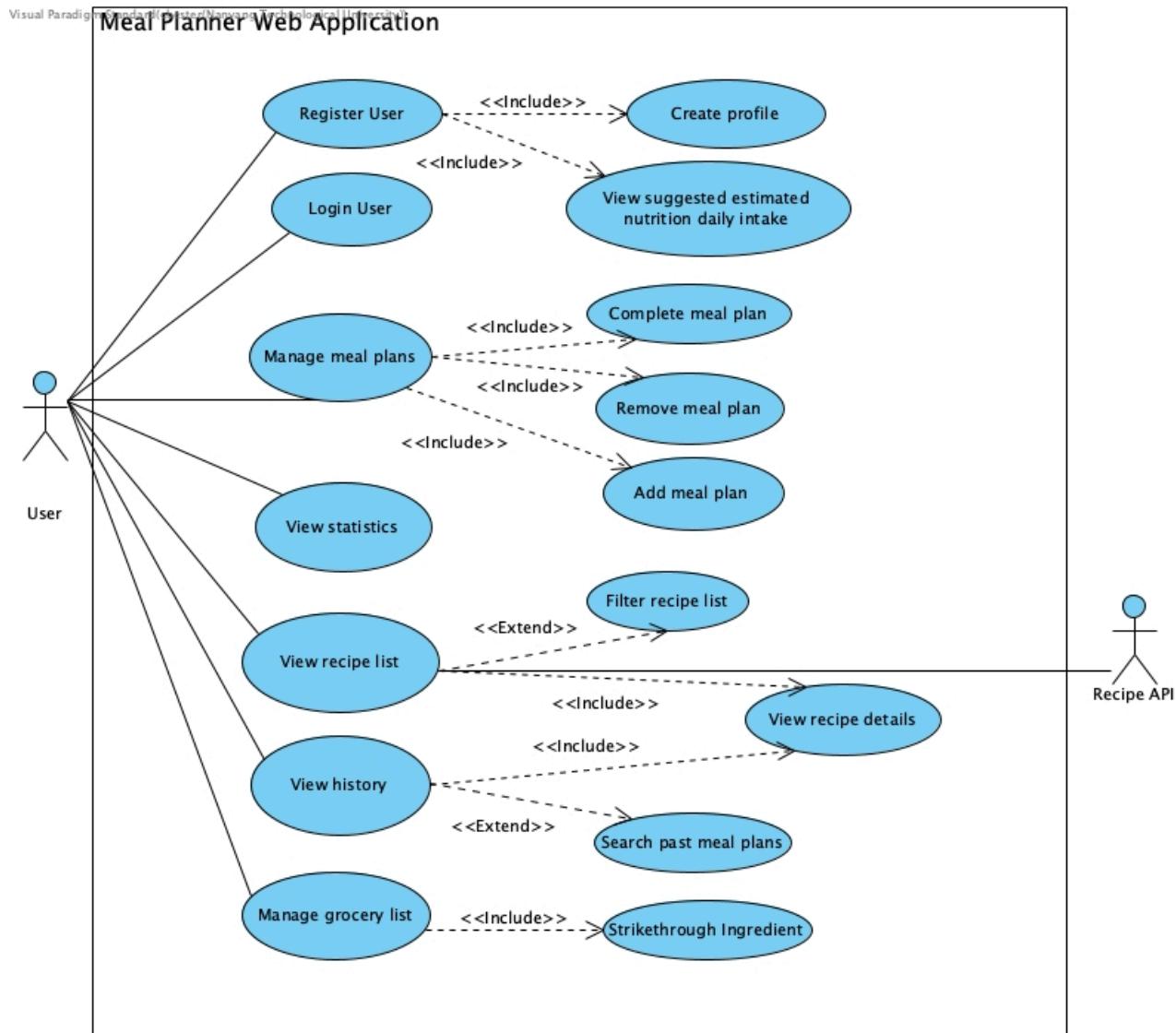


(Figure 1. System Architecture Diagram)

## 2.2 Product Functions

### 2.2.1. Use Case Diagram

The following use case diagram (Figure 2.) shows the key product functionalities that ChompTrack includes. There are 2 primary users of ChompTrack – User and Recipe API. Each type of user interacts with



ChompTrack to carry out their role-specific functions.

(Figure 2. Use Case Diagram)

### 2.2.2. Major Product Functions

ChompTrack will enable users to carry out the following functions:

## 1. Authentication

- Register
    - Create Profile
    - View suggested estimated daily nutrition intake
  - Login

## 2. User Functions

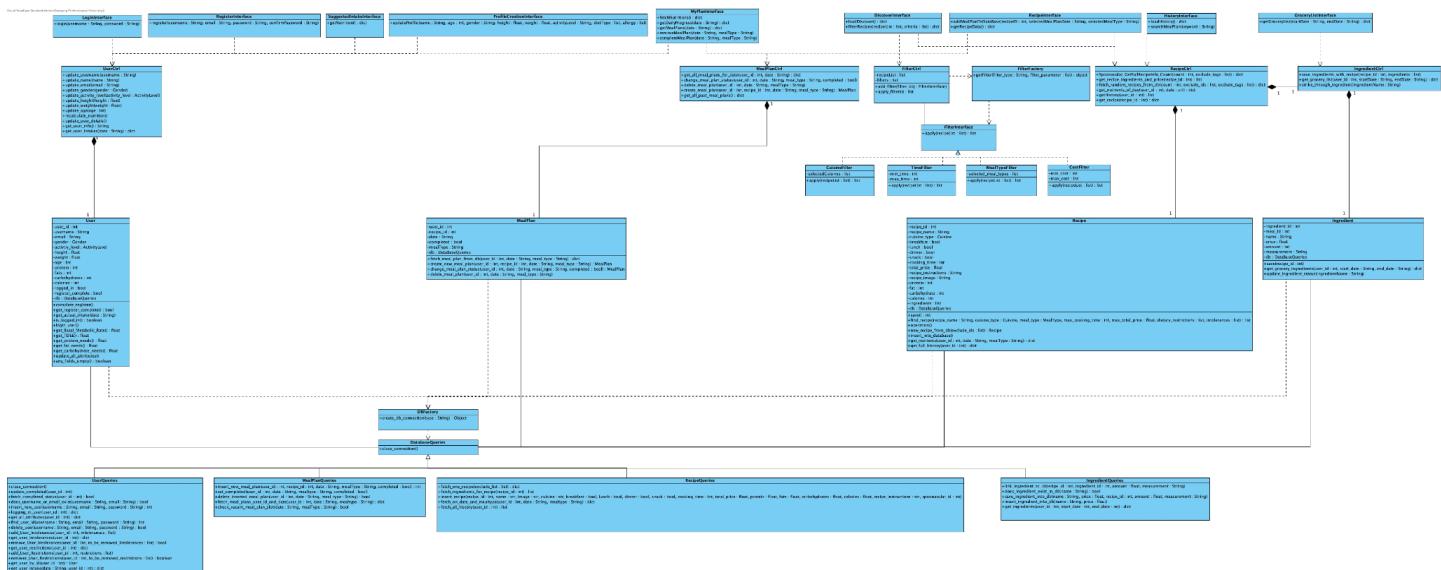
- Meal Planning
    - Add, Complete and Remove Meal Plan
  - Nutrition Tracking (Users can view the total accumulated values of each nutrient from all the meals in a specific day)
  - Recipe Selection
    - View and Filter Recipe List
  - Recipe Details
    - View Recipe Details
  - Grocery List
    - View Grocery List
    - Strikethrough Ingredient (after ingredients have been purchased)
  - View Past Recipes
    - Search Past Meal Plans (filter by name or date)

### 3. Recipe API Functions

- View Recipe List (Recipe API obtains a list of recipes that is passed back to the system to display to the users)

### 2.2.3. Class Diagram

The following Class Diagram (Figure 3.) shows how the different classes interact with each other, as well as the key methods that ChompTrack needs to implement.



(Figure 3. Class Diagram)

## 2.3 User Classes and Characteristics

Users of the platform are categorised into:

- **Recipe Users:** Home cooks, nutritionists, and culinary professionals who seek a variety of recipes. They require easy access to recipe search, nutritional details, ingredient customization, and saving favourites, with minimal technical expertise needed for intuitive navigation.
- **Meal Planners:** Individuals or families planning weekly meals, dietitians creating client meal plans, or fitness enthusiasts tracking nutrition. They require a calendar for meal scheduling, nutritional tracking, customization for dietary restrictions, and grocery list generation, all within a straightforward interface.
- **General Users:** Account holders and occasional visitors who interact with the application. They use functions like account creation, profile management, dietary preference setting, and privacy/security settings, with an interface that caters to various levels of technical expertise.

## 2.4 Operating Environment

The platform will be web-based, operating on the latest versions of Chrome, Firefox, Safari, and Edge.

| Development Environment  | Description   |
|--|---|
| Front-end: HTML,CSS,JS   | The application's user interface (UI) is developed utilizing vanilla HTML, CSS, and JavaScript. It runs in a web browser and uses RESTful API for handling requests and responses.  |
| Back-end:<br>Python<br>Core Libraries:<br>Flask, Requests, MySql.Connector | <ol style="list-style-type: none"> <li>1. Flask, a lightweight web framework in Python, handles RESTful API requests from Front-end.</li> <li>2. The application calls live APIs utilising Request such as Spoonacular to fetch up-to-date data, enabling real-time functionality.</li> <li>3. MySql library also allows seamless connection between backend and database.</li> </ol> |
| Database: MySQL  | Relational Database utilised to store user data to ensure data persistence between uses.<br>It stores data such as recipes, user data and ingredients.  |

## 2.5 Design and Implementation Constraints

The development of ChompTrack will be subject to several constraints that will guide the design and implementation of the software.

### 2.5.1 Frontend Constraints

#### 2.5.1.1 Framework (*Frontend*)

The user interface will be developed using Vanilla HTML, CSS and JavaScript. No Pre-Built or Open-Source Frameworks will be utilised.

### 2.5.2 Backend Constraints

#### 2.5.2.1 Framework (*Backend*)

The backend server will be developed using Flask and RESTful in Python.

#### 2.5.2.2 Database

The database will utilize SQL since it is well-suited for managing structured and relational data. SQL will serve as the primary language for interacting with the database, focusing on ensuring robustness and efficiently handling complex queries.

### 2.5.3 Language Constraints

All software documentation, interfaces, and user interactions will be presented in English. This decision is based on English's widespread use as a primary or secondary language among the target user demographics and to ensure uniformity throughout the platform.

## 2.6 User Documentation

ChompTrack will include a comprehensive set of user documentation to facilitate smooth onboarding and efficient use for developers and end-users alike. Documentation components include:

### 2.6.1 README File

A README file in the main repository will provide a project overview, including architecture details, development setup, and a high-level guide to the repository structure.

### 2.6.2 Code Comments

The source code across frontend and backend will be well-commented to clarify logic, functions, and methods. Complex sections will have additional inline documentation, explaining purpose and usage, and tagging TODOs and FIXMEs for future maintenance.

### 2.6.3 Pydoc

Docstrings are written for each class and method, serving as inline documentation. Python's Pydoc utilizes these docstrings to automatically generate comprehensive documentation for modules, classes, functions, and methods in HTML format.

## 2.7 Assumptions and Dependencies

### 2.7.1 Assumptions

- The web application ChompTrack depends on an external API Spoonacular to obtain relevant information , which has a maximum number of calls we can perform in 1 day, and there will always be enough calls for all our users to interact with in 1 day.
  - If we exceed the number of maximum calls, Spoonacular API will return error code, which will cause Recipes to load unsuccessfully.
- We assume that the data obtained from the API is accurate in nutritional values.
  - Inaccuracies in nutritional values on API side will result in wrong calculations for user's daily consumption

### 2.7.2 Dependencies

Core Libraries and Framework:

- Flask
- Mysql

If functions are deprecated and can no longer be in use, then the Backend could raise an error.

## 3. External Interface Requirements

### 3.1 User Interfaces

#### 3.1.1 Style Guides

##### 3.1.1.1 *Theme Colours*

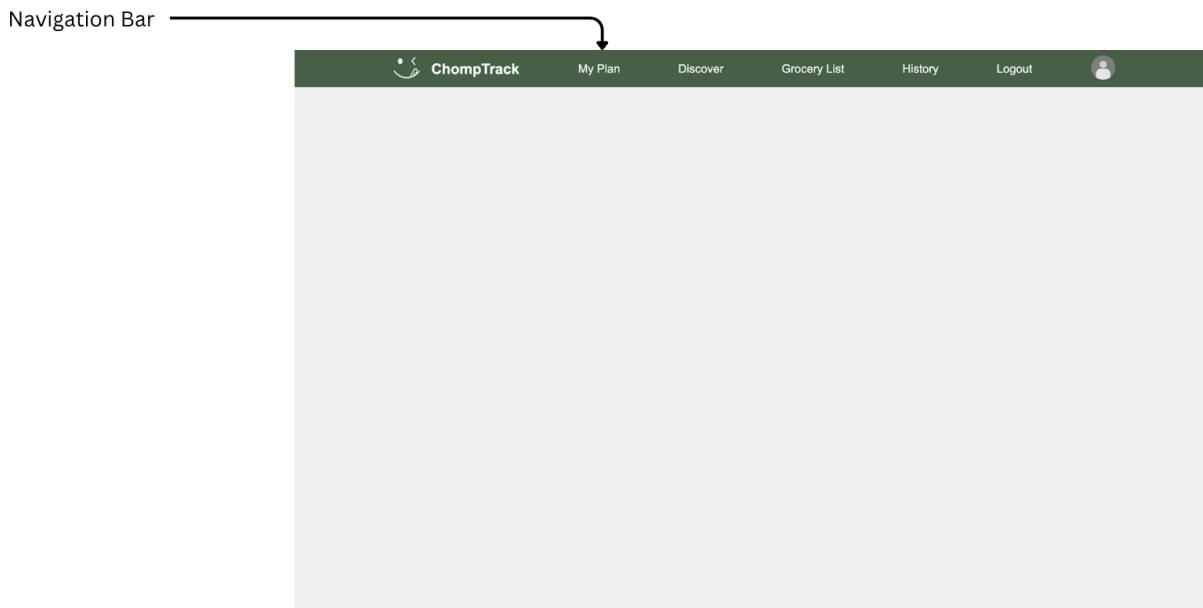
---

|                        |                        |                             |                       |
|------------------------|------------------------|-----------------------------|-----------------------|
| Neutral<br>0xF8F9FA    | Primary<br>0x3F6248    | Primary(Active)<br>0x234127 | Dark<br>0xF8F9FA      |
| Pie Chart<br>0x78ABA8  | Pie Chart<br>0xC8CFA0  | Pie Chart<br>0xEF9C66       | Pie Chart<br>0xFCFC94 |
| Light Gray<br>0xF3F3F3 | Light Gray<br>0xF0F0F0 | Gray<br>0CCCCCC             | Dark Gray<br>0x999999 |

### 3.1.1.2 Typography

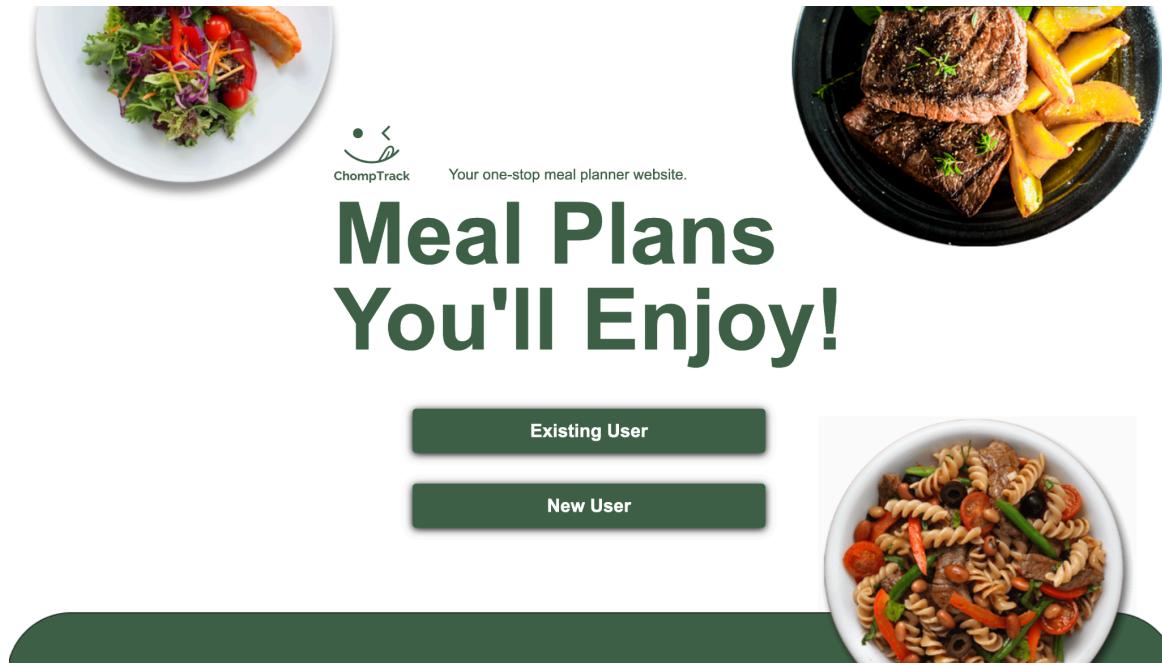
|               |      |       |
|---------------|------|-------|
| Heading       | 32px | Arial |
| SubHeading    | 20px | Arial |
| Text (Large)  | 18px | Arial |
| Text (Medium) | 14px | Arial |
| Text (Small)  | 12px | Arial |
| Text (XS)     | 11px | Arial |

### 3.1.1.2 Screen Template

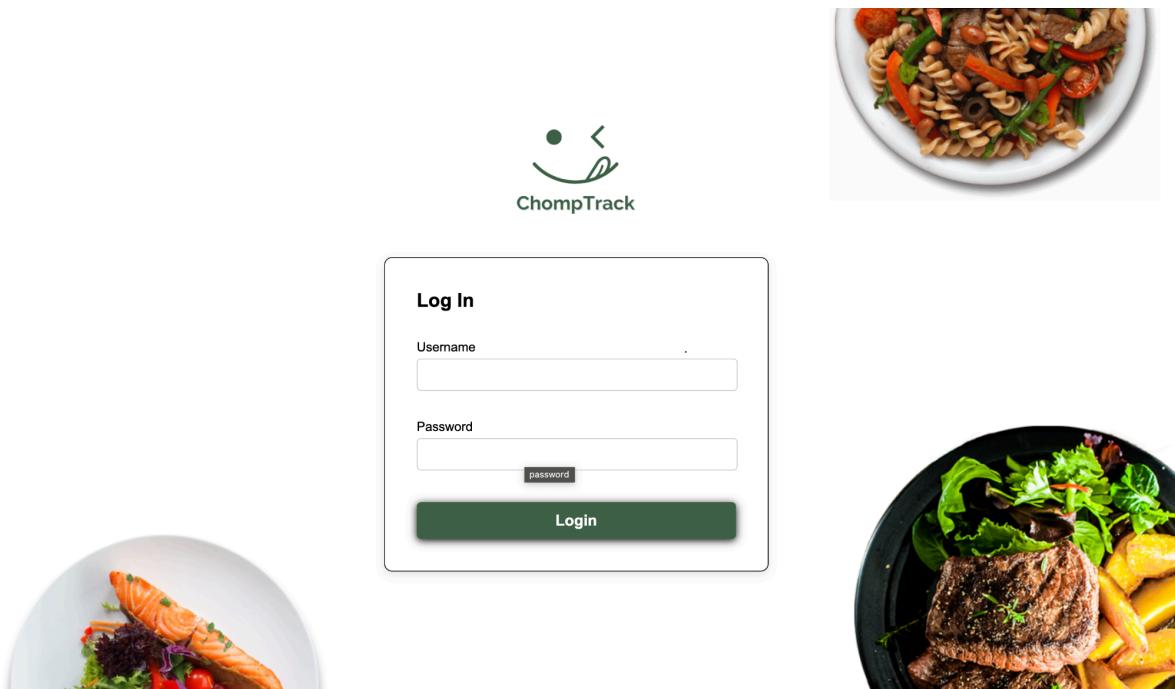


### 3.1.2 UI Mockups

#### 3.1.2.1 Landing Page



#### 3.1.2.2 Login Page



### 3.1.2.3 Registration Page



**Register**

Username

Email

Password

Confirm Password

cfmpassword  
**Register**



### 3.1.2.4 Profile Creation Page



**Complete Your Profile**

Name

Age

Gender

Height (cm)

Weight (kg)

Activity level

Diet Type (Optional)

Allergy (Optional)

**Next**



### 3.1.2.5 Suggested Daily Estimated Nutrition Intake Page

Your Suggested Daily Estimated Nutrition Intake

|                |          |
|----------------|----------|
| Calories:      | 1920kcal |
| Carbohydrates: | 264g     |
| Protein:       | 90g      |
| Fats:          | 50g      |

**Let's Start!**

### 3.1.2.6 My Plan Page

Hello, user!

Date: Sunday, November 10, 2024

Your Suggested Estimated Daily Nutrients Intake

|                      |
|----------------------|
| Calories: 1920 kcal  |
| Carbohydrates: 264 g |
| Protein: 90 g        |
| Fats: 50 g           |

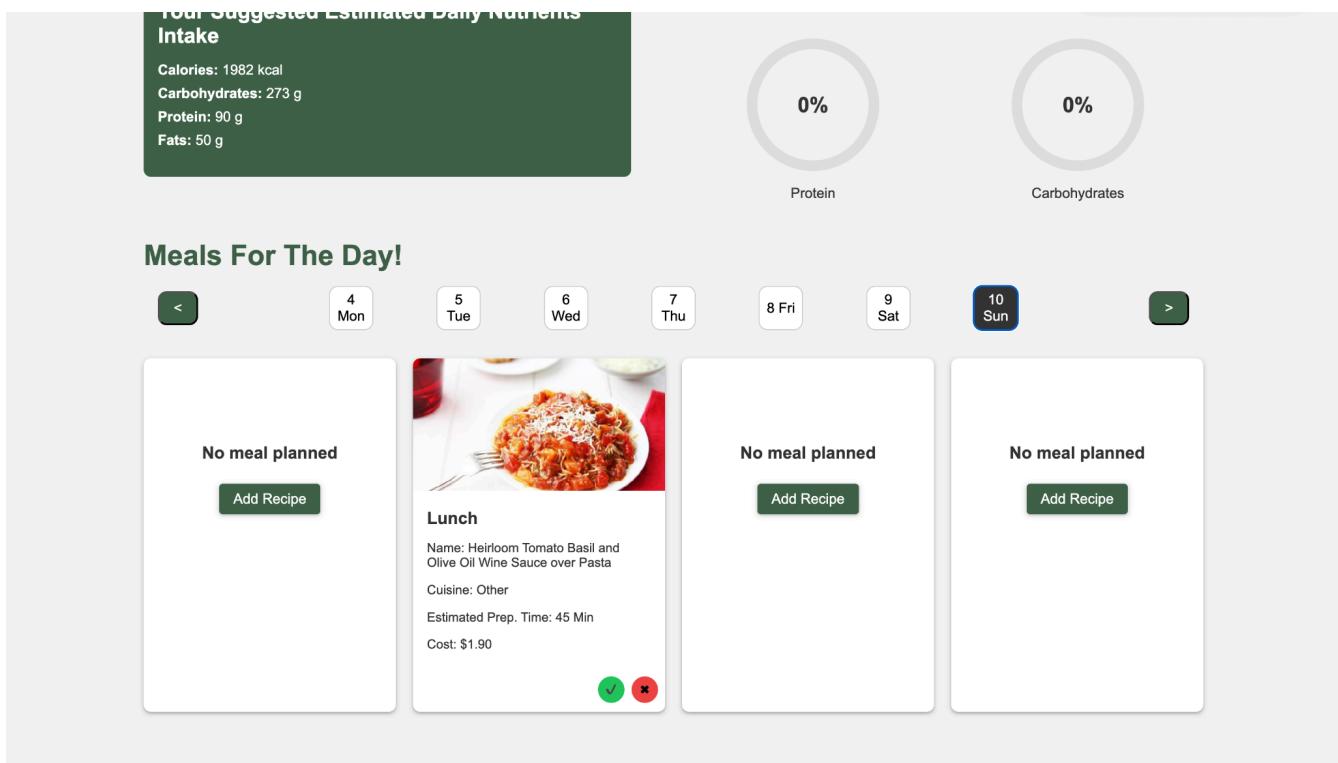
**Daily Progress**

- Calories: 0%
- Fats: 0%
- Protein: 0%
- Carbohydrates: 0%

**Meals For The Day!**

| 4 Mon           | 5 Tue           | 6 Wed           | 7 Thu           | 8 Fri | 9 Sat | 10 Sun |
|-----------------|-----------------|-----------------|-----------------|-------|-------|--------|
| No meal planned | No meal planned | No meal planned | No meal planned |       |       |        |

### 3.1.2.6.1 Interactive Calendar



### 3.1.2.7 Discover Page

The screenshot shows the 'Discover' page of the ChompTrack application. The top navigation bar includes links for 'My Plan', 'Discover', 'Grocery List', 'History', 'Logout', and a user profile icon. The main content area is titled 'Discover' and features a 'Filter' sidebar on the left with options for 'Cost' (Min \$0, Max \$100), 'Estimated Preparation Time' (Min 0 minutes, Max 120 minutes), and 'Cuisine' (checkboxes for American, Mediterranean, Southern, Italian, Indian, Mexican, Japanese, Korean, Chinese, and Other). To the right of the filter are four categories: 'Breakfast', 'Lunch', 'Dinner', and 'Snacks', each with a corresponding image and a list of recipes. The 'Breakfast' category shows 'Guinness Braised Corned Beef and Cabbage' (European cuisine, 45 min prep, \$2.90). The 'Lunch' category shows 'Heirloom Tomato Basil and Olive Oil Wine Sauce over Pasta' (Other cuisine, 45 min prep, \$1.90). The 'Dinner' category shows 'Salmon Butternut Squash Corn Chowder' (Other cuisine, 45 min prep, \$3.12). The 'Snacks' category shows 'Chocolate Nutella Walnut Cake' (Other cuisine), 'Bacon Ice Cream' (Other cuisine), and 'Lemony Zucchini Fritters' (Other cuisine).

### 3.1.2.7.1 Cost, Estimated Preparation Time, Cuisine Filter

**Filter**

**Cost**

Min \$ 0

Max \$ 3

**Estimated Preparation Time**

Min (minutes) 0

Max (minutes) 120

**Cuisine**

- American
- Mediterranean
- Southern
- Italian
- Indian
- Mexican
- Japanese
- Korean
- Chinese
- Other

### 3.1.2.7.2 Course Type Filter Buttons



### 3.1.2.8 Recipe Page

The screenshot shows the ChompTrack application interface. At the top, there is a dark green header bar with the logo "ChompTrack" and navigation links: "My Plan", "Discover", "Grocery List", "History", "Logout", and a user profile icon.

The main content area features a title "Heirloom Tomato Basil and Olive Oil Wine Sauce over Pasta" above a large image of the dish. To the left of the image, there is a sidebar with the following details:

- Cost:** \$1.90
- Cuisine:** Other
- Estimated Preparation time:** 45min
- View recipe instructions**

Below the sidebar is a green button labeled "Add to Meal Plan".

To the right of the image, there are two sections: "Ingredients:" and "Nutrition:". The "Ingredients:" section lists the following items:
 

- 1 16 oz Bag Of OG Pasta
- 5 Organic Heirloom Tomatoes
- 1/3 cup Organic Fresh Basil
- 3 large or 4 small cloves of Organic Garlic
- 1 Tablespoon plus 1/2 cup of Organic Extra Virgin Olive
- 1/3 cup of Dry White Wine
- 1 Teaspoon Sea Salt or to taste

The "Nutrition:" section provides the following nutritional information:
 

- Calories: 501kcal
- Carbohydrates: 92g
- Protein: 16g
- Fat: 6g

#### 3.1.2.8.1 Add Meal Plan Pop-Up

This screenshot shows the same ChompTrack Recipe Page as the previous one, but with a modal dialog box overlaid. The dialog is titled "Select Date and Course for Meal Plan" and contains the following fields:
 

- A date input field labeled "dd/mm/yyyy" with a calendar icon.
- A dropdown menu labeled "Select course".
- A green "Confirm" button at the bottom.

The rest of the page content, including the dish image, sidebar details, and nutrition information, remains visible beneath the modal.

### 3.1.2.9 Grocery List Page

The screenshot shows the 'Grocery List' page of the ChompTrack application. At the top, there is a navigation bar with links for 'My Plan', 'Discover', 'Grocery List' (which is highlighted in green), 'History', and 'Logout'. Below the navigation bar, the title 'Grocery List' is displayed. A date range selector shows 'Nov 4, 2024 - Nov 10, 2024' with arrows to change the dates. The main content area displays a table of items with their quantities and prices:

| Item                       | Quantity | Price  |
|----------------------------|----------|--------|
| og pasta (453.59g)         | 1        | \$0.97 |
| heirloom tomatoes (5)      | 1        | \$4.77 |
| fresh basil (8g)           | 1        | \$0.31 |
| extra virgin olive (1Tbsp) | 1        | \$0.17 |
| dry white wine (80ml)      | 1        | \$1.08 |
| sea salt or (1tsp)         | 1        | \$0.03 |

**Total Cost: \$7.33**

In the bottom left corner of the screenshot, there is a URL: 127.0.0.1:5000/grocery-list.

### 3.1.2.10 History Page

The screenshot shows the 'History' page of the ChompTrack application. At the top, there is a navigation bar with links for 'My Plan', 'Discover', 'Grocery List', 'History' (which is highlighted in green), and 'Logout'. Below the navigation bar, the title 'History' is displayed. On the left, there is a search form with fields for 'Search' (containing 'Search for meals...') and 'Select Date' (containing 'dd/mm/yyyy'). On the right, there is a card for a specific meal entry:

**Heirloom Tomato Basil and Olive Oil Wine Sauce over Pasta**

Recipe used on: 10 November 2024  
Cuisine: Other  
Meal Category: lunch  
Estimated Prep. Time: 45 Min  
Cost: \$1.90

The card also features an image of a plate of pasta with tomato sauce and basil.

### 3.1.3 Error Messages

#### 3.1.3.1 Registration

##### 3.1.3.1.1 Empty Fields

**Register**

Username

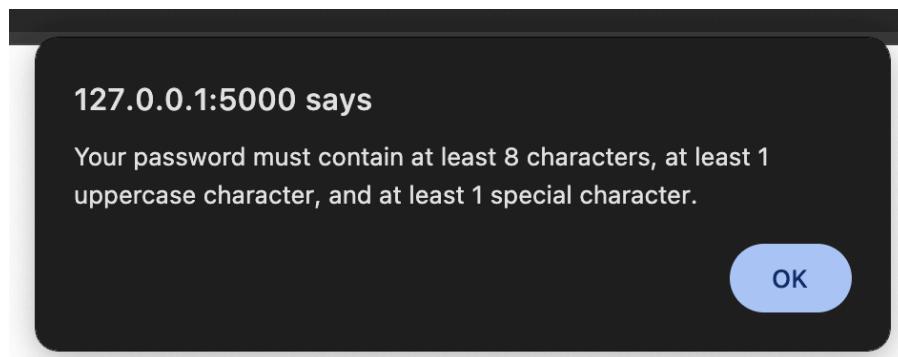
Email  
 ! Please fill in this field.

Password

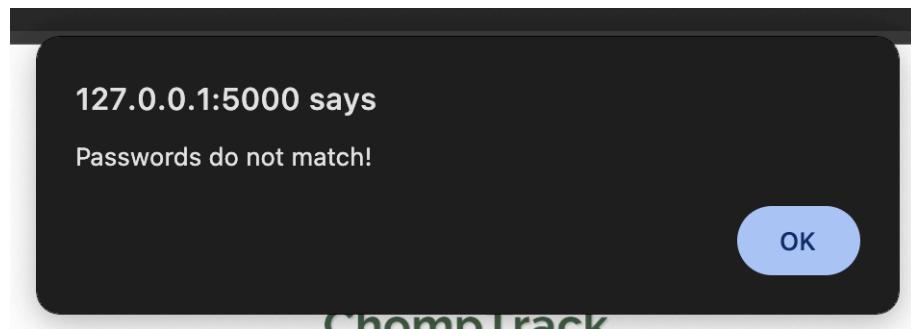
Confirm Password

**Register**

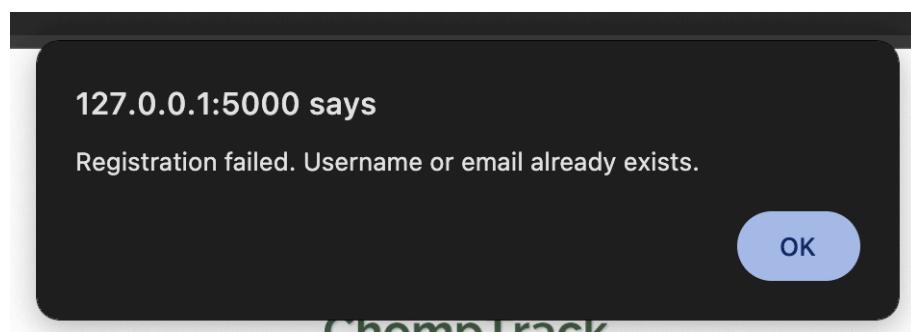
##### 3.1.3.1.2 Password Requirement Not Met



3.1.3.1.3 Password Does Not Match

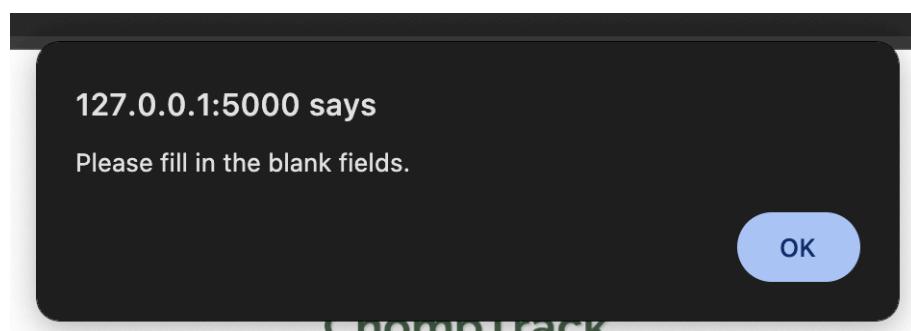


3.1.3.1.4 Existing Username or Email



3.1.3.2 *Profile Creation Page*

3.1.3.2.1 Empty Fields



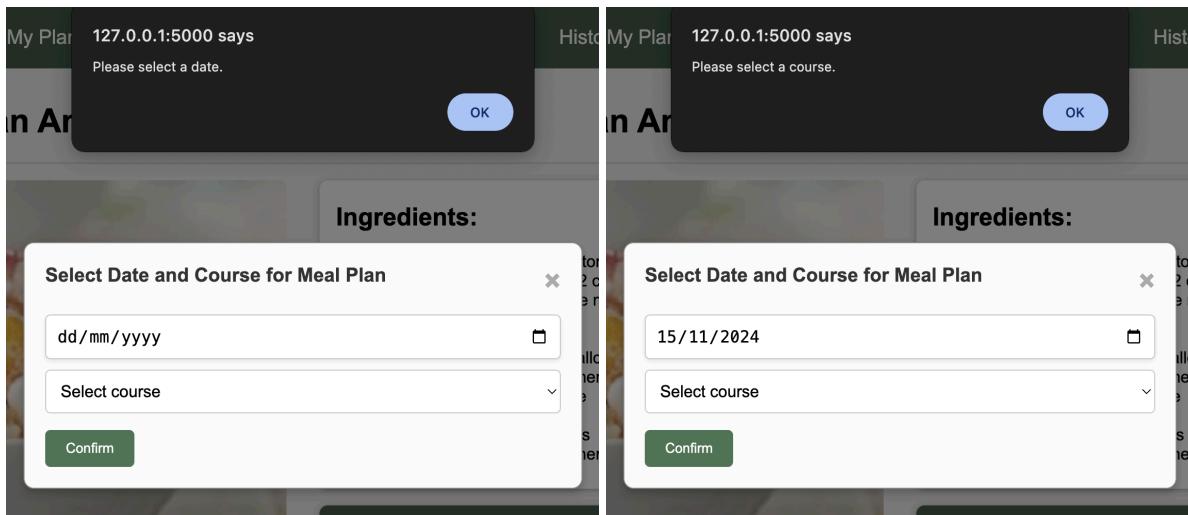
### 3.1.3.2.2 Invalid Values (Age, Weight, Height)

**Complete Your Profile**

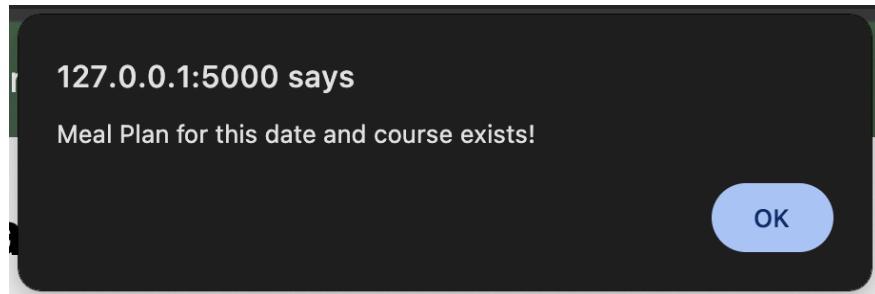
|  |     |        |
|--|-----|--------|
| Name   | Age | Gender |
| user   | -1  | Female |
| Height (cm) <span style="border: 1px solid orange; padding: 2px;">!</span> Value must be greater than or equal to 1. |     |        |
| 150  | 60  |        |
| Activity level   |     |        |
| Moderately active  |     |        |
| Diet Type (Optional)   |     |        |
|  |     |        |
| Allergy (Optional)   |     |        |
|  |     |        |
| <b>Next</b>  |     |        |

### 3.1.3.3 Recipe Page

#### 3.1.3.3.1 Empty Fields

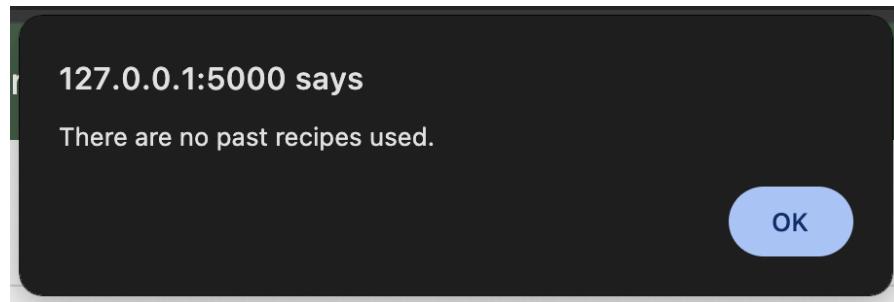


### **3.1.3.3.1 Existing Meal Plan**



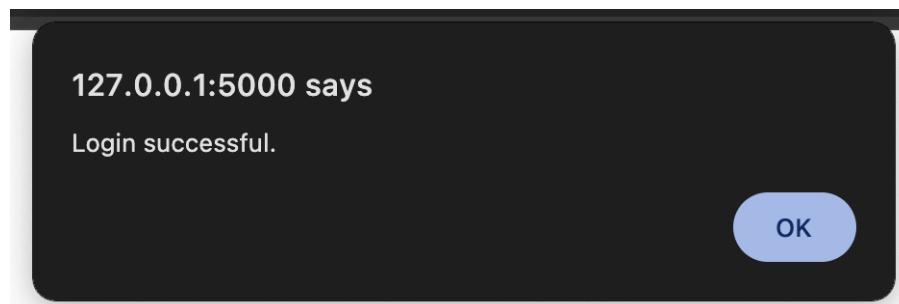
### **3.1.3.4 History Page**

#### **3.1.3.4.1 No Existing Past Recipes**

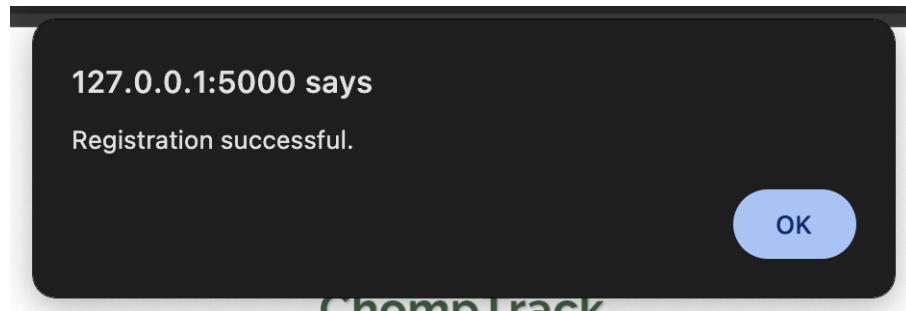


### **3.1.4 Success Message**

#### **3.1.4.1 Login**



### 3.1.4.2 Registration



## 3.1.5 Common Components

### 3.1.5.1 Navigation bar



### 3.1.5.2 Date Buttons



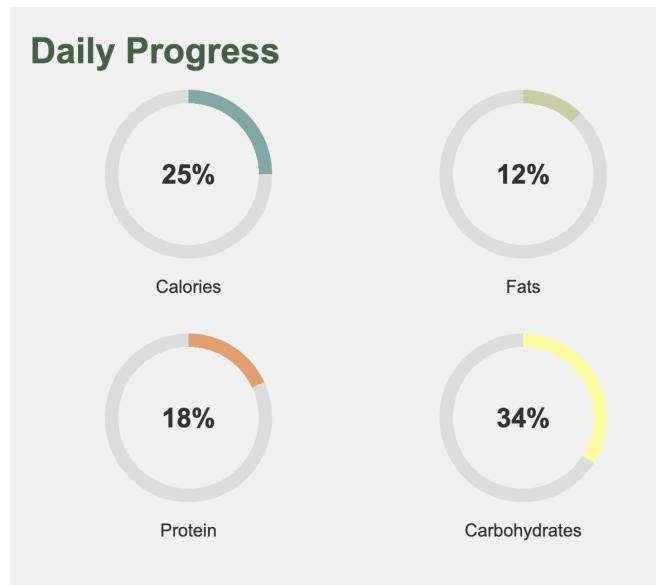
### 3.1.5.4 Meal Cards

The image shows three meal cards displayed side-by-side. The first card on the left is titled "No meal planned" and features a large "Add Recipe" button. The second card in the middle is titled "Dinner" and shows a photo of roast chicken with apples and rosemary. It includes details: Name: Roast Chicken with Apples and Rosemary, Cuisine: Other, Estimated Prep. Time: 45 Min, Cost: \$2.36, and two small circular buttons at the bottom right (green with a checkmark and red with an X). The third card on the right is titled "Lunch" and shows a photo of heirloom tomato basil pasta. It includes details: Name: Heirloom Tomato Basil and Olive Oil Wine Sauce over Pasta, Cuisine: Other, Estimated Prep. Time: 45 Min, Cost: \$1.90, and two small circular buttons at the bottom right (green with a checkmark and red with an X).

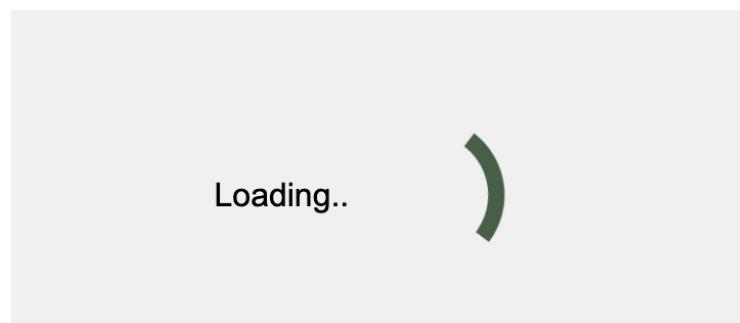
| Meal Type       | Item Description  | Preparation Time | Cost   | Status    |
|-----------------|---|------------------|--------|-----------|
| No meal planned | Add Recipe  | -                | -      | -         |
| Dinner          | Roast Chicken with Apples and Rosemary                    | 45 Min           | \$2.36 | ✓ (Green) |
| Lunch           | Heirloom Tomato Basil and Olive Oil Wine Sauce over Pasta | 45 Min           | \$1.90 | ✓ (Green) |

### **3.1.6 Miscellaneous**

#### **3.1.6.1 Daily Progress Chart**



#### **3.1.6.3 Page Loading**



### 3.1.6.3 *Strikethrough*

| Grocery List                |          |        |
|-----------------------------|----------|--------|
| Item                        | Quantity | Price  |
| venison stew meat (907.10g) | 4        | \$7.04 |
| onion (1)                   | 4        | \$0.24 |
| baby carrots (4)            | 1        | \$0.01 |
| celery (4stalks)            | 1        | \$0.60 |
| canned mushrooms (1can)     | 1        | \$1.23 |
| green bell pepper (1)       | 1        | \$0.36 |
| flour and 3 tbsp (125g)     | 1        | \$0.17 |
| beef bouillon (2Tbps)       | 1        | \$0.63 |
| <b>Total Cost: \$3.01</b>   |          |        |

## 3.2 Hardware Interfaces

### 3.2.1 Logical Characteristics

1. **Supported Devices:**
  - o Desktop: Optimised for modern web browsers (e.g., Chrome, Firefox).
2. **Data Interactions:**
  - o User actions (e.g., selecting preferences) send HTTP/HTTPS requests; responses return as formatted HTML/CSS.
3. **Protocols:**
  - o Uses HTTP/HTTPS for secure communication and JSON for structured data exchange.

### 3.2.2 Physical Characteristics

1. **Requirements:**
  - o **Client Devices:** Standard desktops or smartphones with modern browsers.
2. **Data Flow:**
  - o User devices interact with servers, which process requests, query databases, and fetch data from Spoonacular.
3. **Security:**
  - o HTTPS encryption and API authentication utilising API Key to protect data.

### 3.3 Software Interfaces

#### 3.3.1 Operating System

Any OS capable of running a modern web browser (Windows, macOS, Linux).

#### 3.3.2 Web Browsers

Latest versions of Chrome, Firefox, Safari, Edge, and others capable of supporting HTML5, CSS3, and JavaScript.

#### 3.3.3 Databases

MySQL-based database systems, accessed through backend services.

#### 3.3.4 Back-end tools

- **Flask (Python)** – Framework for developing and running the server-side application, handling requests, and serving responses.
- **MySQL (via MySQL Workbench)** – Relational database for managing and storing user data, recipes, and meal plans.

#### 3.3.5 Front-end tools

- HTML5
- CSS3
- JavaScript

#### 3.3.6 APIs

- **Spoonacular API** – External API used to retrieve recipe and nutrition data, providing a comprehensive meal-planning experience for users.

#### 3.3.7 Data Sharing

JSON format is used for all data exchange between frontend and backend. Data persistence is maintained through the SQL database with specific defined data models.

### 3.4 Communications Interfaces

- **HTTP/HTTPS:** Used for secure communication between the frontend (JavaScript) and backend (Flask server). These protocols allow the exchange of information such as retrieving recipes, meal plans, and user data over the web. All data transmission is secured using HTTPS to protect sensitive information.
- **REST API:** The Flask backend exposes a RESTful API, providing structured endpoints for client-server communication. This allows the frontend to perform operations such as retrieving data (GET), submitting data (POST), updating data (PUT/PATCH), and deleting data (DELETE) in a standardised way.
- **AJAX/Fetch API:** The frontend (JavaScript) uses AJAX and the Fetch API to send asynchronous requests to the Flask REST API. This allows for real-time data updates on the user interface without requiring a page reload, enhancing the user experience by maintaining dynamic and interactive content.

## 4. System Features

Please refer to the '[Use Case Model](#)' folder and '[Sequence Diagrams](#)' folder for the full use case diagram, detailed use case descriptions, and sequence diagrams for the main use cases.

### 4.1 Registration

#### 4.1.1 Description and Priority

Description:

- First time users can register for an account in ChompTrack. The system will create an account after verifying the information entered (username, email and password) is sufficient and valid.

Priority:

- High

#### 4.1.2 Stimulus/Response Sequences

1. User clicks the 'New user' button in the 'Landing' page.
2. System prompts user to enter username, email address and password.
3. User inputs the required information and clicks the 'Register' button.
4. System verifies the information entered, ensuring that it is sufficient and valid.
5. System creates an account and stores these details in the database.
6. User is logged in.

#### 4.1.3 Functional Requirements

REQ-1: The user must be able to register for an account on the system by clicking the 'New user' button in the 'Landing' page.

REQ-1.1: The system must display text fields for the user to enter his information in the "Register" page.

REQ-1.1.1: The text fields must consist of username.

REQ-1.1.2: The text fields must consist of email addresses.

REQ-1.1.3: The text fields must consist of passwords.

REQ-1.2: The user must fill in all the fields before clicking the 'Register' button.

REQ-1.2.1: If the fields are not filled in and the "Register" button is clicked, an error message, "Please fill in this field." must be displayed.

REQ-1.3: The system must verify the fields filled in by the user before creating an account.

REQ-1.3.1: The username entered must be unique and never been registered in the system.

REQ-1.3.1.1: If the username entered has been registered in the system, an error message, "Registration failed. Username or email already exists" must be displayed.

REQ-1.3.2: The password entered must contain at least 8 characters and at least 1 uppercase character and 1 special character.

REQ-1.3.2.1: If the password entered does not meet the criterias, an error message, “Your password must contain between 8 to 16 characters, at least 1 uppercase character and at least 1 special character.” must be displayed.

REQ-1.4: The system must create an account for the user upon verification.

## 4.2 Create Profile

### 4.2.1 Description and Priority

Description:

- After registering an account, first time users have to create a profile by entering information - name, height, weight, age, gender, activity level, diet type and allergy. The information will be used to calculate their estimated suggested daily nutrition intake.

Priority:

- High

### 4.2.2 Stimulus/Response Sequences

- User clicks the 'Register' button in the 'Register' page.
- System prompts user to enter his name, height, weight, age, gender, activity level, diet type and allergy.
- User inputs the required information and clicks the 'Next' button.
- System verifies the information entered, ensuring that it is sufficient and valid.
- System updates the profile and stores these details in the database.
- User is directed to the 'My Plan' page.

### 4.2.3 Functional Requirements

REQ-1: The user must be able to create a profile after registering an account.

REQ-1.1: The system must display text fields for the user to enter his information in the “Register” page.

REQ-1.1.1: The text fields must consist of the user's height, at least 1 cm.

REQ-1.1.2: The text fields must consist of the user's weight, at least 1 kg.

REQ-1.1.3: The text fields must consist of the user's age, at least 1 years old.

REQ-1.2: The system must display choice fields allowing the user to select from predefined options.

REQ-1.2.1: The choice fields must include gender categories (Male, Female).

REQ-1.2.2: The choice fields must include activity level categories (Sedentary, Lightly active, Moderately active, Very active, Super active)

REQ-1.2.3: The choice fields must include diet type categories (Vegetarian, Vegan, Gluten free).

REQ-1.2.4: The choice fields must include allergy categories (dairy, egg, gluten, grain, peanut, seafood, sesame, shellfish, soy, sulfite, tree nut, wheat).

REQ-1.3: The user must fill in all the fields before clicking the ‘Next’ button.

REQ-1.3.1: If the fields are not filled in and the “Next” button is clicked, an error message, “Please fill in the blank fields.” must be displayed.

## 4.3 View suggested estimated nutrition daily intake

### 4.3.1 Description and Priority

Description:

- After completing their profile, users will be able to view the suggested estimated nutrition daily intake. The estimated nutrition daily intake is determined by calculating the user's basal metabolic rate (BMR) and total daily energy expenditure (TDEE).

Priority:

- High

### 4.3.2 Stimulus/Response Sequences

1. When a user clicks the ‘Next’ button in the profile creation page, the system will display the suggested daily intake of calories, proteins, fats and carbohydrates based on the data the user has entered during profile creation.
2. The system will display the suggested estimated daily intake for each nutrient.

### 4.3.3 Functional Requirements

REQ-1: When clicking the ‘Next’ button, the system must display suggested daily estimated nutrition intake for the user.

REQ-1.1: The system must calculate the user's basal metabolic rate (BMR).

REQ-1.2: The system must calculate the user's total daily energy expenditure (TDEE) which is the estimated daily calories intake, displayed in kcal/day and rounded off to the nearest whole number .

REQ-1.3: The system must calculate the user's estimated daily protein intake and display it in grams/day, rounded off to the nearest whole number.

REQ-1.4: The system must calculate the user's estimated daily fats intake and display in grams/day, rounded off to the nearest whole number.

REQ-1.5: The system must calculate the user's estimated daily carbohydrates intake and display it in grams/day, rounded off to the nearest whole number.

REQ-2: When the “Next” button is clicked, the system must display the “My Plan” page.

## 4.4 Login

### 4.4.1 Description

Description:

- After successful registration, the user must be able to log back into the system by inputting the correct username and password in order to access the web application.

Priority:

- High

### 4.4.2 Stimulus/Response Sequences

1. User clicks the 'Existing user' button in the 'Landing' page.
2. The system displays the 'Login' page.
3. The user inputs username and password.
4. The user clicks on the 'Login' button.
5. System verifies the user's input with the user's data from the database.
6. System displays the "My plan" page.

### 4.4.3 Functional Requirements

REQ-1: The user must be able to log into the system by clicking the 'Existing user' button in the 'Landing' page.

REQ-1.1: The system must display text fields for the user to enter his information in the "Login" page.

REQ-1.1.1: The text fields must consist of username.

REQ-1.1.2: The text fields must consist of password.

REQ-1.2: The user must input the login credentials (username and password) in the corresponding text fields before clicking "Login" button.

REQ-1.3: The system must verify the login credentials input by the user.

REQ-1.3.1: The system must find the username in the database of the system.

REQ-1.3.2: The system must ensure that the password entered matches the password associated with the specified username.

REQ-1.4: If the username entered is found in the database of the system and the password entered matches the password associated with the specified username, the login credentials are verified.

REQ-1.4.1: The system must log the user into the "My Plan" page of the system.

REQ-1.5: If the username entered is not found in the database of the system, the login credentials are not verified.

REQ-1.5.1: The system must display the message "Login failed: Invalid username or password."

REQ-1.6: If the username entered is found in the database of the system but the password entered does not match the password associated with the username, the login credentials are not verified.

REQ-1.6.1: The system must display the message “Login failed: Invalid username or password.”

REQ-1.7: The user must be allowed to return to the ‘Landing’ page.

## 4.5 My Plan

### 4.5.1 Calendar

#### 4.5.1.1 View Meal Plans

##### 4.5.1.1.1 Description

Description:

- Users can view their selected recipes for each course of the day.

Priority:

- High

##### 4.5.1.1.2 Stimulus/Response Sequences

1. The user clicks on ‘My plan’ in the navigation bar.
2. The system displays an interactive weekly calendar.
3. The user can toggle between weeks by clicking on the left or right arrow.
4. The user clicks on a date on the calendar to view the meals planned for that day.

##### 4.5.1.1.3 Functional Requirements

REQ-1: The system displays a personalized greeting (e.g., "Hello [Username]") at the top of the page.

REQ-2: The system will provide an interactive weekly calendar for users to select dates easily.

REQ-2.1: The user must be able to toggle between weeks by clicking on the left and right arrows.

REQ-2.2: The user must be able to switch between days within the selected week by clicking on the date.

REQ-2.2.1: The system must display the 4 recipes planned for each course (breakfast, lunch, dinner and snack).

REQ-2.2.1.1: Each planned recipe must display basic information of the recipe.

REQ-2.2.1.1.1: The meal card must display the image of the meal.

REQ-2.2.1.1.2: The meal card must display the name of the meal.

REQ-2.2.1.1.3: The meal card must display the cost of the meal.

REQ-2.2.1.1.4: The meal card must display the cuisine of the meal.

REQ-2.2.1.2: If no recipes are planned for a course, the message, ‘No course planned’ and an ‘Add recipe’ button will be displayed.

REQ-2.2.1.3: When the ‘Add recipe’ button is clicked, the system will direct the user to the ‘Discover’ page.

#### 4.5.1.2 Complete Meal Plan

##### 4.5.1.2.1 Description

Description:

- The user can mark a course as completed for that particular day.

Priority:

- High

##### 4.5.1.2.2 Stimulus/Response Sequences

1. The user clicks on the tick icon next to a meal.
2. The system marks the course as completed.
3. The system updates the statistics of the different nutrients and displays it under ‘Daily Progress’.

##### 4.5.1.2.3 Functional Requirements

REQ-1: The system must display the 4 recipes planned for each course (breakfast, lunch, dinner and snack).

REQ-1.1: The system must display a tick icon.

REQ-1.1.1: When the user completes the meal, the user can click the tick icon to represent completion.

#### 4.5.1.3 Remove Meal Plan

##### 4.5.1.3.1 Description

Description:

- The user can remove a course from the meal plan for that particular day.

Priority:

- High

##### 4.5.1.3.2 Stimulus/Response Sequences

1. The user clicks on the cross icon next to a meal.
2. The system removes the meal from that course on the selected day.
3. The system displays 'No meal planned' and an 'Add recipe' button will be displayed.
4. The system removes the ingredients required to prepare the meal from the grocery list.

#### **4.5.1.3.3 Functional Requirements**

REQ-1: The system must display the 4 recipes planned for each course (breakfast, lunch, dinner and snack).

REQ-1.1: The system must display a cross icon.

REQ-1.1.1: When the cross icon is clicked, the system will remove the recipe from that course of that particular day.

REQ-1.1.2: When the cross icon is clicked, the system will remove the ingredients needed for the recipe from the grocery list.

### **4.5.2 Statistics**

#### **4.5.2.1 View Statistics**

##### **4.5.2.1.1 Description**

Description:

- Users can view the total accumulated values of each nutrient from all the meals consumed for the day. The user's actual daily nutrient intake will be displayed as a percentage of the suggested daily intake of each nutrient, in the form of pie charts.

Priority:

- High

##### **4.5.2.1.2 Stimulus/Response Sequences**

1. The user clicks on the 'tick' icon of a recipe to mark it as complete
2. The system displays the total accumulated values of each nutrient from the meals marked as completed for the day.
3. The system displays the pie charts for each nutrient, showing the accumulated daily values and the progress toward the recommended intake. The percentage of the progress is also shown in the centre of the pie chart, rounded to the nearest whole number.

##### **4.5.2.1.3 Functional Requirements**

REQ-1: The system must display the overall nutrition statistics of a particular day.

REQ-1.1: The system must display the total accumulated values of each nutrient from all the meals planned throughout the day.

REQ-1.1.1: The system must calculate the total accumulated calories intake from all the meals of that day and display in kcal, rounded to the nearest whole number.

REQ-1.1.2: The system must calculate the total accumulated protein intake from all the meals of that day and display in grams, rounded to the nearest whole number.

REQ-1.1.3: The system must calculate the total accumulated fats intake from all the meals of that day and display in grams, rounded to the nearest whole number.

REQ-1.1.4: The system must calculate the total accumulated carbohydrate intake from all the meals of that day and display in grams, rounded to the nearest whole number.

REQ-1.2: The system must display pie charts for each nutrient, showing the accumulated daily values and the progress toward the recommended intake. The percentage of the pie chart filled represents this progress.

REQ-1.2.1: The pie chart must display the progress of reaching the suggested intake for each nutrition in percentages to the nearest whole number.

REQ-1.2.2: The pie chart must have different designated colours for each nutrition to visually distinguish them.

## 4.6 Discover

### 4.6.1 View Recipe List

#### 4.6.1.1 Description

Description:

- The system will draw the recipe list from the AP and users can find the different recipes that suit their preference.

Priority:

- High

#### 4.6.1.2 Stimulus/Response Sequences

1. The user clicks on ‘Discover’ in the navigation bar.
2. The system calls the recipeAPI to obtain a list of recipes
3. The system displays the recipes to the users
4. If the user uses the filter, the system displays a list of recipes that satisfies the conditions.

#### 4.6.1.3 Functional Requirements

REQ-1: The system must display a list of recipes that matches the user’s profile.

REQ-1.1: The system calls the recipeAPI and obtains the list of recipes.

REQ-1.2: Each recipe in the “Discover” page must display basic information about the recipe.

    REQ-1.2.1: The meal card should display the image of the meal.

    REQ-1.2.2: The meal card should display the name of the meal.

    REQ-1.2.3: The meal card should display the cost of the meal.

    REQ-1.2.4: The meal card should display the estimated preparation time of the meal.

    REQ-1.2.5: The meal card should display the cuisine of the meal.

## 4.6.2 Filter Recipe List

### 4.6.2.1 Description

Description:

- User is able to filter the list of recipes based on the cost, cuisine, course and estimated preparation time.

Priority:

- High

### 4.6.2.2 Stimulus/Response Sequences

1. The system will display a list of filters (cost, cuisine, course, estimated preparation time).
2. User inputs the options to filter the list of recipes.
3. System updates the list of recipes displayed.

### 4.6.2.3 Functional Requirements

REQ-1: The system must display a list of filters.

    REQ-1.1: The list of filters shall consist of the cost filter.

        REQ-1.1.1: The system displays a text field for the user to enter the minimum and maximum price, in dollars.

        REQ-1.1.2: Recipes that fall within that specified price range will be displayed.

    REQ-1.2: The list of filters shall consist of the cuisine filter.

        REQ-1.2.1: The system displays a choice field of cuisines available.

        REQ-1.2.2: Recipes matching the cuisines selected will be displayed.

    REQ-1.3: The list of filters shall consist of the estimated preparation time filter.

        REQ-1.3.1: The system displays a text field for the user to enter the minimum and maximum estimated preparation time, in minutes.

        REQ-1.3.2: Recipes that fall within that specified estimated preparation time range will be displayed.

    REQ-1.4: The list of filters shall consist of the course filter.

REQ-1.4.1: The system displays a choice field of courses.

REQ-1.4.2: Recipes matching the courses selected will be displayed.

### 4.6.3 Recipe

#### 4.6.3.1 View Recipe Details

##### 4.6.3.1.1 Description

Description:

- The user can view detailed information about a recipe.

Priority:

- High

##### 4.6.3.1.2 Stimulus/Response Sequences

1. The user clicks on the recipe's image from the 'Discover' page.
2. The system displays the recipe's details.

##### 4.6.3.1.3 Functional Requirements

REQ-1: When clicking on the recipe's image, the system will display the recipe's page.

REQ-1.1: The recipe's page must display detailed information about the recipe.

    REQ-1.1.1: The page must display the image of the recipe

    REQ-1.1.2: The page must display the name of the recipe

    REQ-1.1.3: The page must display the cost of the recipe

    REQ-1.1.4: The page must display the course of the recipe

    REQ-1.1.5: The page must display the estimated preparation time.

    REQ-1.1.6: The page must display the ingredients needed for the recipe

    REQ-1.1.7: The page must display the nutrition of the recipe

    REQ-1.1.8: The page must display the link to a website with instructions.

#### 4.6.3.2 Add Meal Plan

##### 4.6.3.2.1 Description

Description:

- The user adds a recipe to their meal plan by selecting a date and course (e.g., breakfast, lunch, dinner, snack) and the recipe is then reflected under the 'My Plan' page.

Priority:

- High

#### **4.6.3.2.2 Stimulus/Response Sequences**

1. The user clicks on the ‘Add to Meal Plan’ button on the recipe’s page.
2. The system displays a popup with fields to input the date and course.
3. The user selects the date and course for the meal.
4. The system checks the selected date and course
5. The system adds the recipe to the meal plan for the selected date and course.

#### **4.6.3.2.3 Functional Requirements**

REQ-1: When clicking on the recipe’s image, the system will display the recipe’s page.

REQ-1.1: The recipe page must have a “Add to Meal Plan” button to add the recipe to the user’s meal plan.

REQ-1.1.1: When the button is clicked, the system will display a popup.

REQ-1.1.1.1: The popup consists of a choice field that prompts users to input the date the meal will be prepared.

REQ-1.1.1.2: The popup consist of a choice field that prompts users to input the course (breakfast, lunch, dinner, snack)

REQ-1.1.1.3: When users click on the ‘Confirm’ button, the recipe will be added to the meal plan for the specific date and course selected.

REQ-1.1.1.4: If the date and course selected already has a meal planned, the system will display the message, “Meal Plan for this date and course exists!”

REQ-1.1.1.5: If the date is not selected, the system will display the message, “Please select a date”

REQ-1.1.1.6: If the course is not selected, the system will display the message, “Please select a course.”

## 4.7 Grocery List

### 4.7.1 View Grocery List

#### 4.7.1.1 Description

Description:

- The user can view the consolidated grocery list for a particular week. The price and quantity required of each ingredient and the total cost of all ingredients will be reflected as well. Ingredient prices, updated to reflect current market rates, are retrieved via the recipeAPI.

Priority:

- High

#### 4.7.1.2 Stimulus/Response Sequences

1. The user clicks on ‘Grocery List’ in the navigation bar.
2. The system displays a list of ingredients to purchase for the meal plans from a particular week.
3. The system calls the recipeAPI to fetch the price of each ingredient.
4. The system displays the price and quantity required of each ingredient
5. The system calculates the total cost by summing up all the prices of the ingredients.

#### 4.7.1.3 Functional Requirements

REQ-1: The system must display the grocery list of a particular week, displaying all ingredients needed to make all the meals for that particular week.

REQ-1.1: The system must display the name of the ingredient.

REQ-1.2: The system must display the quantity required for each ingredient.

REQ-1.3: The system must display the prices of each ingredient.

REQ-1.3.1: The system calls recipe API to fetch the price of each ingredient.

REQ-1.3.2: The price of each ingredient in the grocery list must be displayed in SGD, to 2 decimal places of accuracy.

REQ-1.4: The system must display the total cost of the grocery list for a particular week.

REQ-1.4.1: The system calculates the total price by summing up all the prices of the ingredients.

REQ-1.4.2: The total price of the grocery list for a particular week must be displayed in SGD, to 2 decimal places of accuracy.

## 4.7.2 Strikethrough Ingredient

### 4.7.2.1 Description

Description:

- The user can strike through the ingredients in the grocery list to mark them as purchased.

Priority:

- High

### 4.7.1.2 Stimulus/Response Sequences

1. The user clicks on the name of the ingredient.
2. The system displays the name of the ingredient with a strikethrough.
3. The system recalculates the total cost of all ingredients.
4. The system displays the new total cost.

### 4.7.1.3 Functional Requirements

REQ-1: The system must display the grocery list of a particular week, displaying all ingredients needed to make all the meals for that particular week.

REQ-1.1: The user can strike through an ingredient from the grocery list by clicking on the text of the ingredient to mark them as purchased.

REQ-1.1.1: The system recalculates the total cost of all ingredients.

REQ-1.1.2: The system displays the new total cost.

## 4.8 History

### 4.8.1 View History

#### 4.8.1.1 Description

Description:

- User can find and filter past recipes used.

Priority:

- Medium

#### 4.8.1.2 Stimulus/Response Sequences

1. User navigates to the 'History' page.
2. The system displays a list of all the past recipes.

#### **4.8.1.3 Functional Requirements**

REQ-1: The user must be able to view a list of past recipes.

REQ-1.1: The system must display the list of recipes the user has previously used.

REQ-1.1.1: The meal card must display the recipe name.

REQ-1.1.2: The meal card must display the dates the recipes were used.

REQ-1.1.3: The meal card must display the cuisine of the recipe.

REQ-1.1.4: The meal card must display the images of the recipe.

REQ-1.1.5: The meal card must display the estimated prep time of the recipe.

REQ-1.1.6: The meal card must display the cost of the recipes, in SGD, displayed to 2 decimal places of accuracy.

#### **4.8.2 Search Past Meal Plans**

##### **4.8.2.1 Description**

Description:

- User can search for specific past recipes used.

Priority:

- Medium

##### **4.8.2.2 Stimulus/Response Sequences**

1. The user enters keywords into the search bar.
2. The system displays past recipes with the keyword in the recipe's details.

##### **4.8.2.3 Functional Requirements**

REQ-1: The user must be able to view a list of past recipes.

REQ-1.1: The user must be able to filter past recipes by selecting a specific date.

REQ-1.1.1: The system displays past recipes used on the specific date.

REQ-1.2: The user must be able to search for a specific recipe.

REQ-1.2.1: The system must provide a text field search bar for users to enter keywords related to the recipe.

REQ-1.2.2: The search criteria text must be minimally 1 character long.

REQ-1.2.3: The search criteria text must be maximally 50 characters long.

REQ-1.2.4: The search criteria can contain any ASCII recognised characters.

- REQ-1.2.5: The system must return recipes that match the search criteria.
- REQ-1.3: When the user clicks on the image of the recipe, the system will display the recipe's page.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- Each page should load in no more than 3 seconds, including logging in and authentication.
- Errors that could arise from synchronisation issues like two or more consumers request for the same API call at the same time should be kept to a maximum of no more than 100 times per month.
- The web application should be scalable enough to support at least 1000 users at the same time while maintaining optimal performance.
- The system should be available 24/7 with no more than 1 day of downtime per month

### 5.2 Safety Requirements

#### 5.2.1 Safe Handling Reminders

The application shall provide reminders and tips on safe food handling and storage practices. This will help reduce the likelihood of food spoilage or contamination and enhance food safety awareness among users.

#### 5.2.2 User Privacy and Data Security

ChompTrack shall not require users to share personal information, including real names, addresses, or contact details, with other non-admin users, even for recipe sharing or messaging functions. This ensures user privacy and protects users from potential misuse of personal data.

### 5.3 Security Requirements

- The user's personal information shall be kept confidential and shall not be disclosed to other users of the application or the public, in line with the PDPA.
- There should be zero tolerance for data leaks or compromise and the database should be stored securely by limiting access to the database to key developers and admins of the application.
- The system should implement password hashing for storing the users' passwords to protect users' accounts.
- SQL injection preventive measures must be in place.

## 5.4 Software Quality Attributes

### 5.4.1 Reliability

- The system shall perform a complete backup procedure every 48 hours, ensuring all data can be recovered in the event of system failure
- Maximum allowed time for data recovery shall not exceed 30 minutes
- Automated system health checks will automatically detect and address software issues

### 5.4.2 Usability

- The system UI shall be designed such that a minimum of 90% of first time users can navigate from the home page to the intended feature within 5 minutes of first visiting the website.

### 5.4.3 Supportability

- The system should log errors together with performance metrics and user activity to allow future analysis and troubleshooting.
- System design and implementation should be modular to allow individual components to be serviced without affecting the entire system
- The code will always be accompanied with the relevant code comments indicating at least the purpose of the following block of code, to allow for interpretability, code reusability and future maintenance.

## 5.5 Business Rules

### 5.5.1 Nutritional and Allergen Compliance

- All recipes added to the app must have fields for nutritional information and allergen indicators. Users can be required to fill out these fields for shared recipes, ensuring consistent information and compliance with safety standards.

### 5.5.2 Data Retention and Deletion

- ChompTrack will retain user data (such as meal plans, recipes, and pantry items) as long as the user has an active account. Users can delete their data upon request, or if the account is inactive for a specific period, with prior notification, to comply with data retention policies.

### **5.5.3 Third-Party Data Integration**

- Any third-party data used for recipes, nutritional information, or pricing (e.g., Spoonacular API) must be accurately sourced and credited. This ensures compliance with external data policies and maintains data reliability within the app.

## 6. Other Requirements

### 6.1 Internationalisation Requirements

- ChompTrack will initially support only the English language. Future updates may include multi-language support, including dialects like Hokkien and Cantonese, to better serve a broader audience in Singapore, particularly senior citizens who may not understand English.
- Date and time representations should be in the Singapore timezone (GMT+8).

### 6.2 Legal Requirements

- Compliance with PDPA when handling user's sensitive data.

### 6.3 Reuse Objectives

- The platform's components should be modular and reusable, ensuring flexibility within the project and for future applications.
- APIs will follow industry standards, such as REST, to enable seamless external integration and usage.

### 6.4 Accessibility Requirements

- Must meet WCAG 2.1 Level AA accessibility standards to ensure it is usable by people with disabilities.

## Appendix A: Glossary

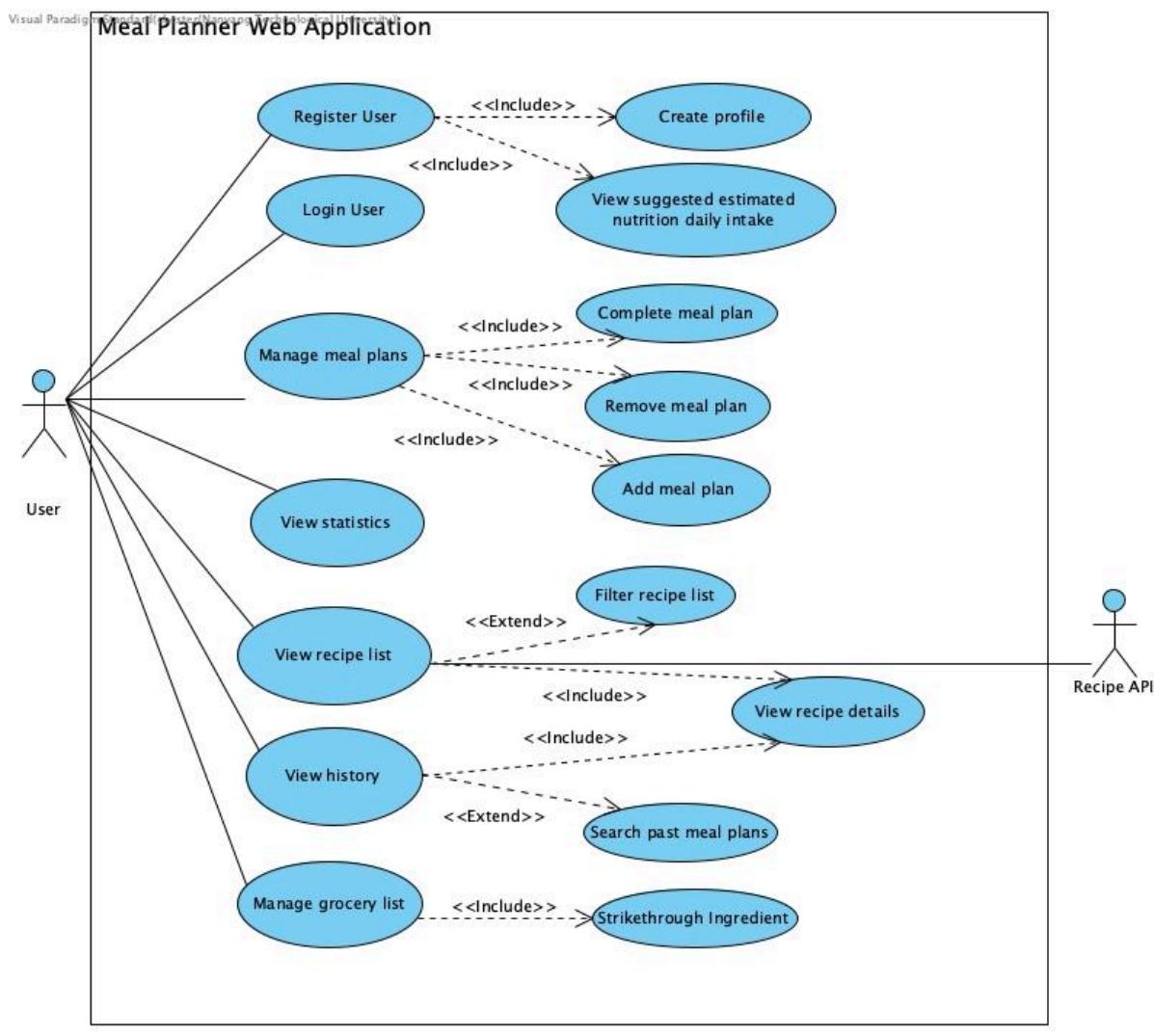
| Term                        | Description   |
|-----------------------------|---|
| User                        | Refers to the party operating the web application   |
| Username                    | A unique identifier that was chosen by the user during registration.  |
| Password                    | A secret String of characters used for user authentication. This must be a minimum of 8 characters long, including an uppercase character and a special character   |
| Age                         | The user's age measured in years old, provided from the "create profile" page.  |
| Height                      | The user's height measured in centimetres, provided from the "create profile" page  |
| Weight                      | The user's weight measured in kilograms, provided from "create profile" page  |
| Gender                      | A choice field that the user can choose from either "female" or "male".   |
| Activity Level              | A choice field on "create profile" page where user selects their level of physical activity. The categories are sedentary (little to no exercise), Lightly active (light exercise/sports 1-3 days/week), Moderately active (moderate exercise/sports 3-5 days/week), Very active (hard exercise/sports 6-7 days a week), Super active (very hard exercise & physical job or 2x training). |
| Diet Type                   | A choice field on "create profile" page where user selects the type of nutritional plan they are on based on a specific criteria. The categories are anything, vegetarian, vegan, dairy free, gluten free, paleo).  |
| Allergy Categories          | A choice field on "create profile" page where user selects the types of food they are allergic to. The categories are gluten, peanuts, eggs, fish, dairy, soy, shellfish).  |
| 'Login' Page                | The page where users enter their username and password to access their account. This page displays text fields for username and password  |
| 'My Plan' Page              | The page that users are redirected to after successfully logging in. This is the main dashboard where the users personalised informations such as meal plan is displayed  |
| Course                      | A course is a meal that is either breakfast, lunch, dinner or snack.  |
| Interactive weekly calendar | A visual calendar that allows users to select dates and navigate between weeks. Users can also switch between days within a selected week by selecting on the specific date.  |

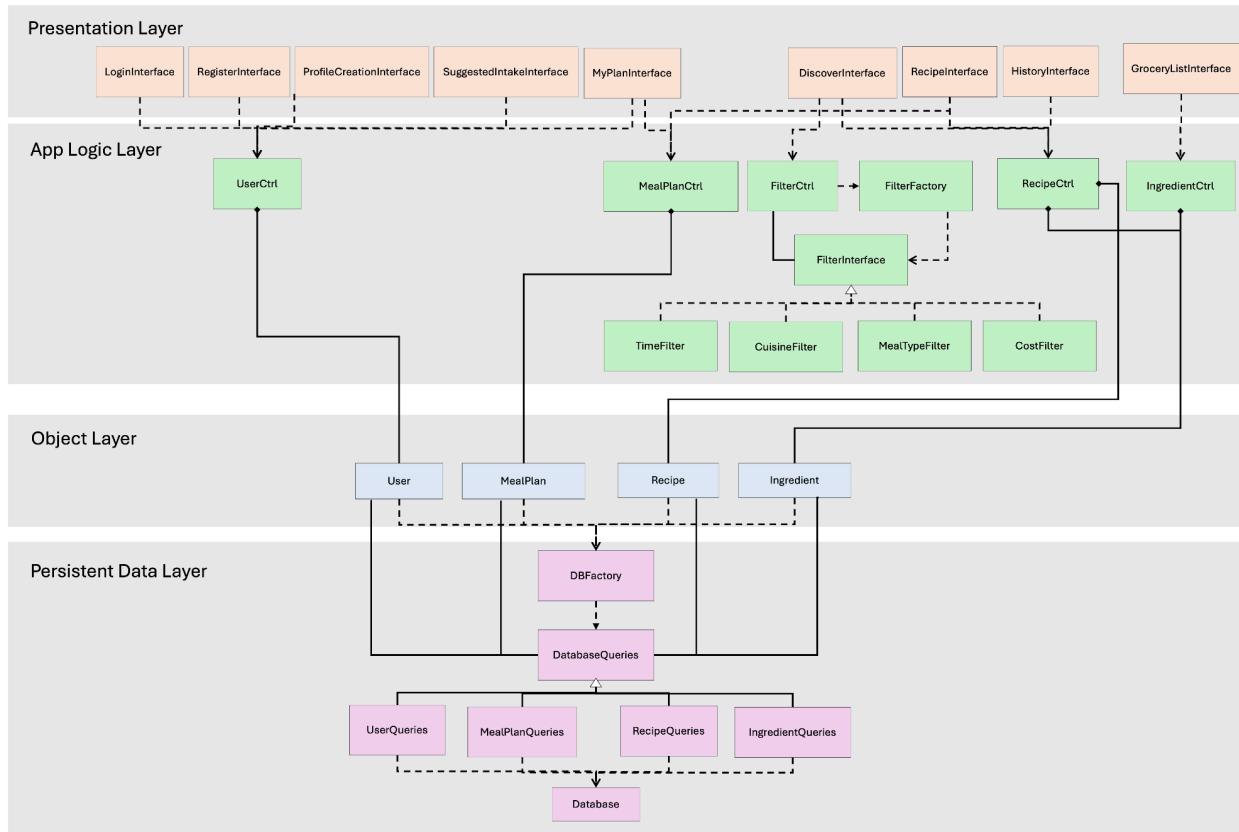
|                                      |   |
|--------------------------------------|---|
| Tick icon                            | An icon used to mark a meal as completed.   |
| Cross icon                           | An icon used to remove a recipe from the planned menu for a particular course.  |
| Basal Metabolic Rate(BMR)            | <p>BMR represents the number of calories the body needs to maintain its current weight without any additional activity.</p> <p>Calculation of BMR is as follows:</p> <p>Men: <math>(10 \times \text{weight in kg}) + (6.25 \times \text{height in cm}) - (5 \times \text{age in years}) + 5</math><br/>     Women: <math>(10 \times \text{weight in kg}) + (6.25 \times \text{height in cm}) - (5 \times \text{age in years}) - 16</math></p>   |
| Total Daily Energy Expenditure(TDEE) | <p>TDEE gives the estimated number of calories we need to maintain our current weight based on our activity level.</p> <p>Calculation of TDEE is as follows:</p> <p>Sedentary (little to no exercise): BMR x 1.2<br/>     Lightly active (light exercise/sports 1-3 days/week): BMR x 1.375<br/>     Moderately active (moderate exercise/sports 3-5 days/week): BMR x 1.55<br/>     Very active (hard exercise/sports 6-7 days a week): BMR x 1.725<br/>     Super active (very hard exercise &amp; physical job or 2x training): BMR x 1.9</p>  |
| Calories                             | Calories refer to the energy we get from the food we consume. Displayed in grams, up to the nearest whole number.   |
| Protein                              | <p>Protein are molecules that are required for the structure, function and regulation of the body's tissues and organs, found mostly in seafood and meat. Displayed in grams, up to the nearest whole number.</p> <p>Calculation of daily protein needs is as follows:</p> <p>For sedentary/lightly active individuals: 1-1.2g/kg/day<br/>     For moderately active - extremely active: 1.4-2.2g/kg/day</p> <p>Convert body weight in pounds to kg's (round to the nearest 10th). Multiply weight in kilograms by the range that best fits your activity levels.</p> <p><i>Example:</i></p> $150\text{lbs} / 2.2 = 68.2\text{kg}$ $68.2\text{kg}(1g) = 68\text{g}$ |
| Fats                                 | Fats is a source of essential fatty acids which help the body absorb Vitamin A, D and E. Displayed in grams, up to the nearest whole number.  |

|                               |   |
|-------------------------------|---|
|                               | <p>Calculation of daily fat needs is as follows:</p> <p>Convert body weight in pounds to kg's (round to the nearest 10th). Multiply weight in kilograms by 1.</p> <p><i>Example:</i></p> <p><math>150\text{lbs} / 2.2 = 68.2\text{kg}</math><br/> <i>68g of fat needed per day</i></p>  |
| Carbohydrates                 | <p>Carbohydrates is being broken down into glucose, which is the main source of energy for the body's cells, tissues and organs. Displayed in grams, up to the nearest whole number.</p> <p>Calculation of daily carbohydrates needs is as follows:</p> <p>Multiply daily calorie requirements by 0.45 &amp; 0.65 to obtain calories from carbohydrates.</p> <p><i>Example:</i></p> <p>a. <math>0.45(2000) = 900 \text{ calories}</math><br/> b. <math>0.65(2000) = 1300 \text{ calories}</math></p> <p>Divide answers in the above step by 4 since there are 4 calories per 1 gram of carbohydrate</p> <p><i>Example:</i></p> <p>a. <math>900/4 = 225\text{g}</math><br/> b. <math>1300/4 = 325\text{g}</math></p> |
| Nutrition progress percentage | The percentage displayed on the pie chart showing the progress towards the recommended intake for each nutrient, rounded to the nearest whole number.   |
| RecipeAPI                     | An external service or API that provides a list of recipes. The system queries this API to obtain recipe suggestions.   |
| 'Discover' Page               | A page where users can view recipes that match their profile. This page displays recipes with basic information and allows users to filter the list based on various criteria.  |
| Cost                          | The total price of ingredients needed to make the meal using the recipe. Displayed in SGD and up to 2 decimal places.   |
| Cuisine                       | The type of meal. The categories are western, italian, chinese, thai, etc.  |

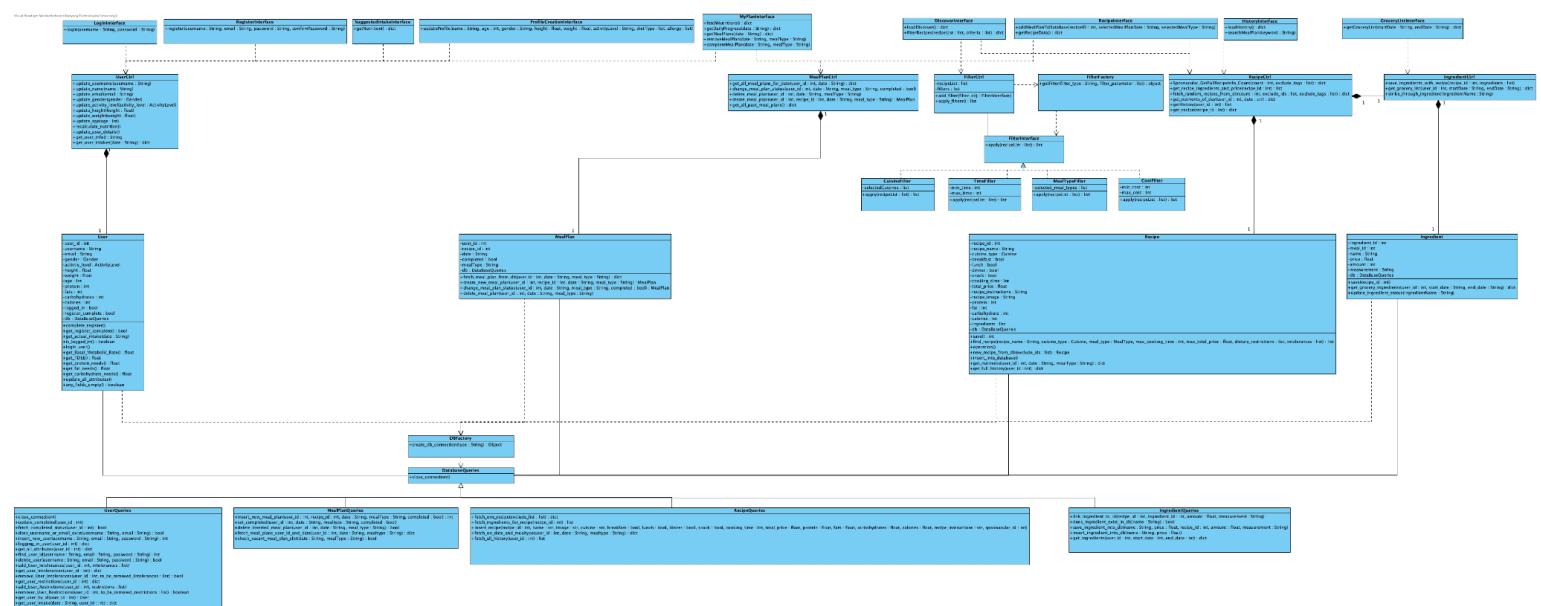
|                            |   |
|----------------------------|---|
| Estimated Preparation Time | It is the duration needed to make the meal according to the recipe given. Displayed in minutes.   |
| 'Recipe' Page              | A detailed page that displays comprehensive information about a specific recipe. This includes the basic data (image, name, cost, cuisine), estimated preparation time, nutrition breakdown, ingredient list with prices, and instructions.                 |
| 'Add to Meal Plan' Button  | A button on the recipe page that allows users to add the selected recipe to their meal plan. Clicking this button opens a popup for users to select the date and course for the meal.   |
| Popup                      | A dialog that appears when users click the "Add to Meal Plan" button. It includes choice fields for selecting the date and course (breakfast, lunch, dinner, snack) for the meal. It also has a 'Confirm' button to finalize the addition to the meal plan. |
| Ingredient List            | A compiled list of all ingredients needed to make the food from the recipe.   |
| Grocery List               | A compiled list of all ingredients needed and to be purchased for meals planned for a particular week. It includes the total number of items for each ingredient.   |
| Ingredient Price           | The cost of each ingredient displayed in SGD (Singapore Dollars), rounded to 2 decimal places of accuracy. This price is fetched using the Web Scraper API.   |
| Past Recipes List          | A display of recipes that the user has previously used. It includes the recipe names, creation dates, cuisine types, images, and costs, displayed in SGD to 2 decimal places of accuracy.   |
| Search Bar                 | A text field provided for users to enter keywords related to a recipe. The search criteria must be between 1 and 50 characters long, contain any ASCII characters, and will return recipes matching the search criteria.                                    |
| Sort and Filter Options    | The functionalities that allow users to organize and narrow down the list of past recipes. Users can sort by criteria such as date, cuisine type, cost, estimated preparation time, and date, and apply filters using a choice field.                       |
| Recipe Page Navigation     | The functionality that allows users to click on a recipe image to be directed to the recipe's detailed page, where more comprehensive information about the recipe is displayed.  |

## Appendix B: Analysis Models

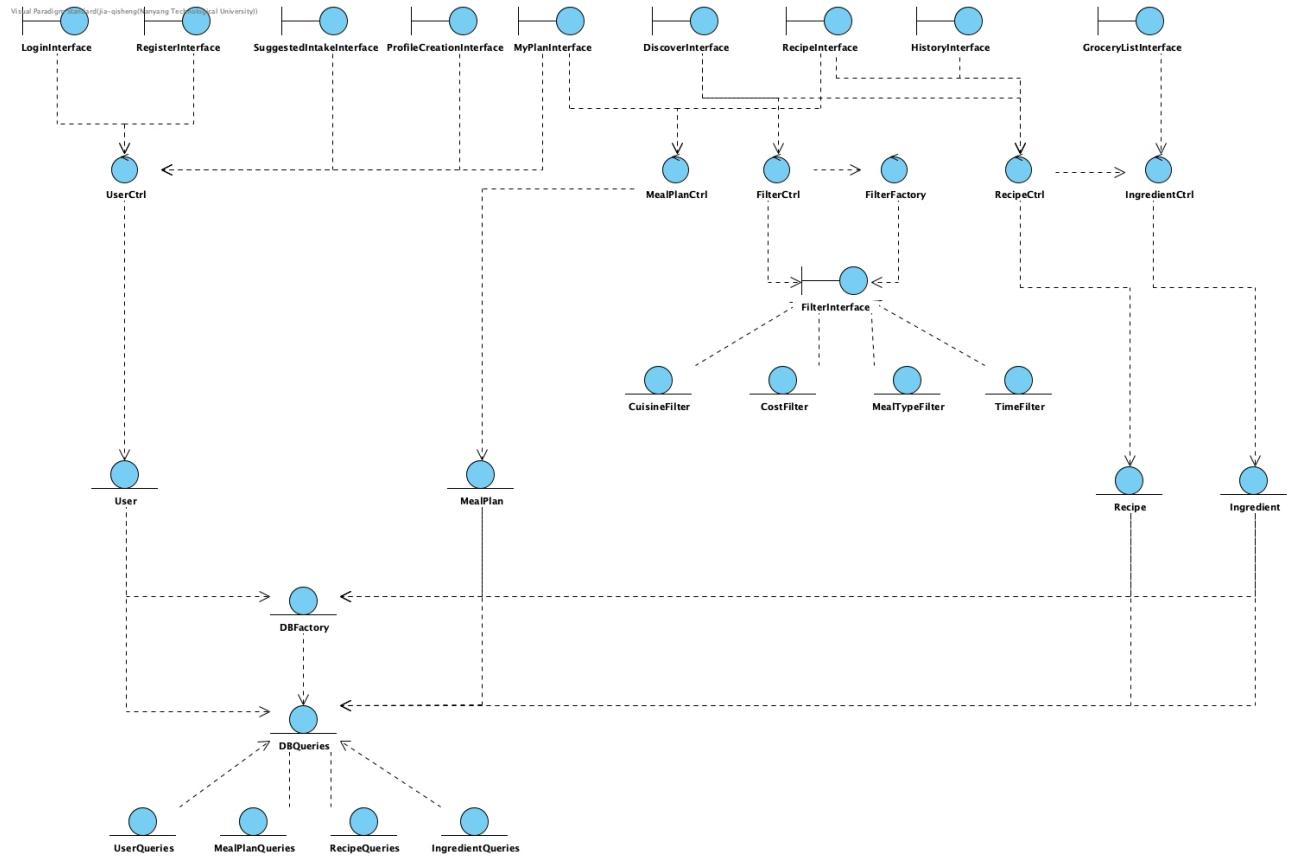




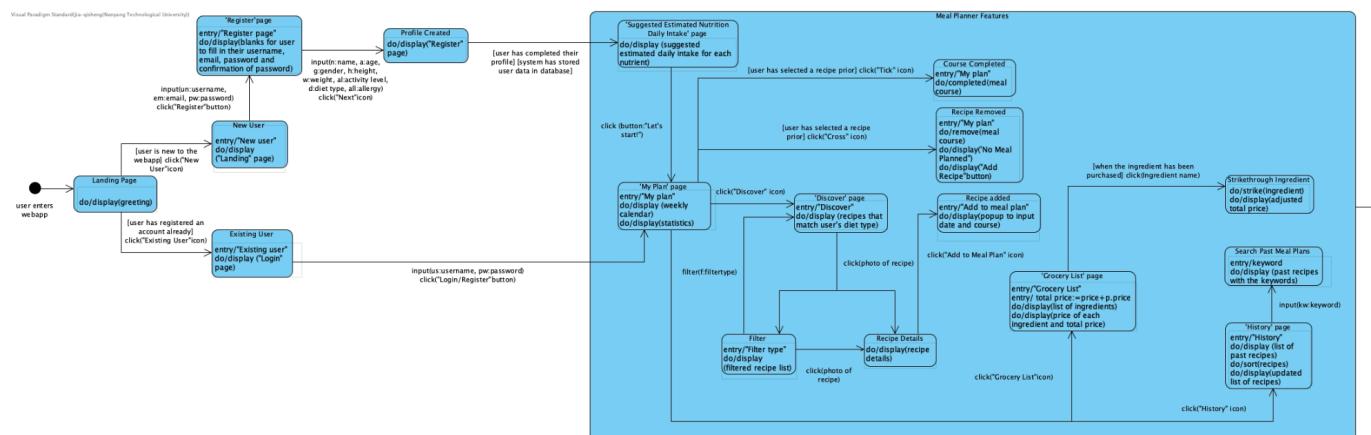
## System Architecture



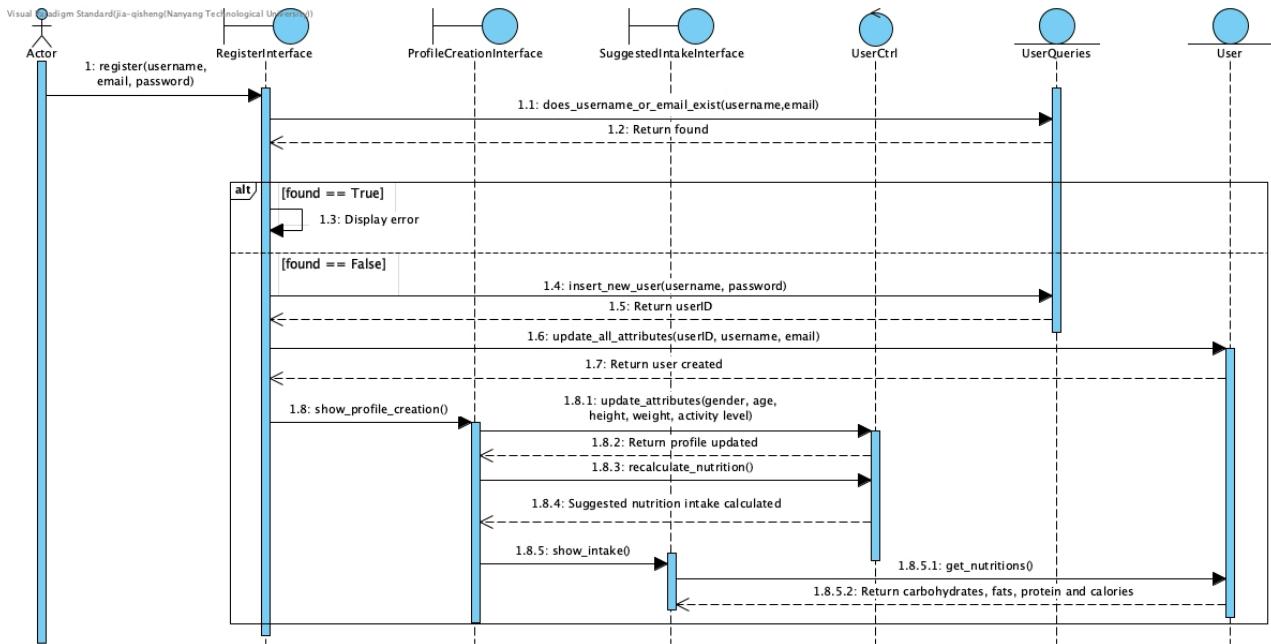
Class Diagram



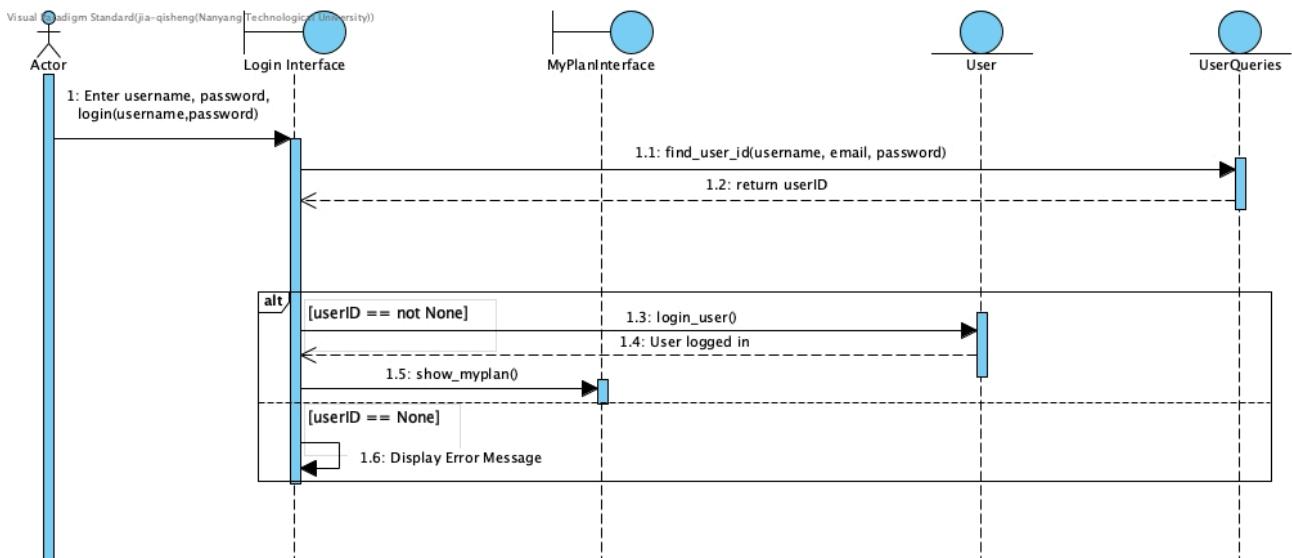
Key Boundary and Control Class Diagram



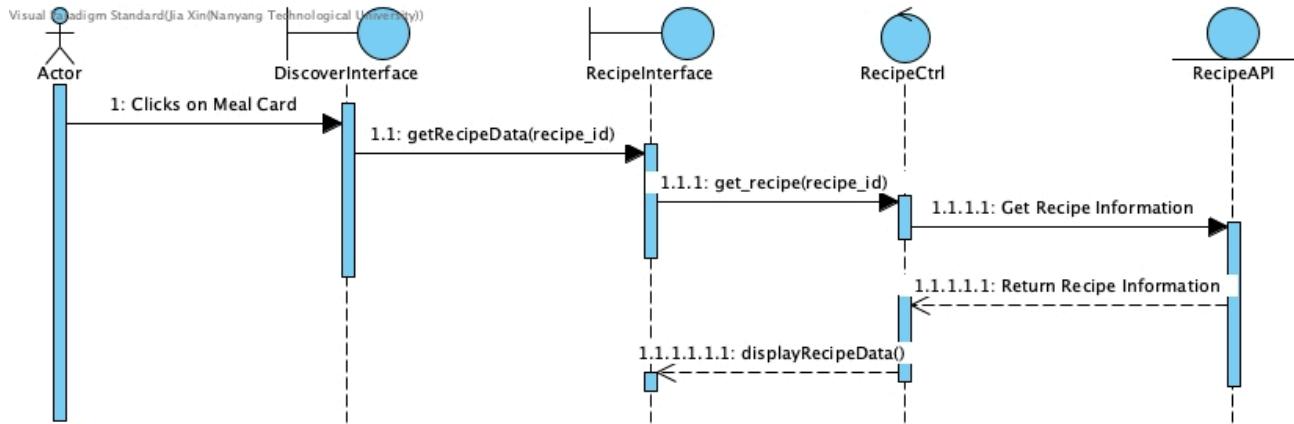
Dialog Map



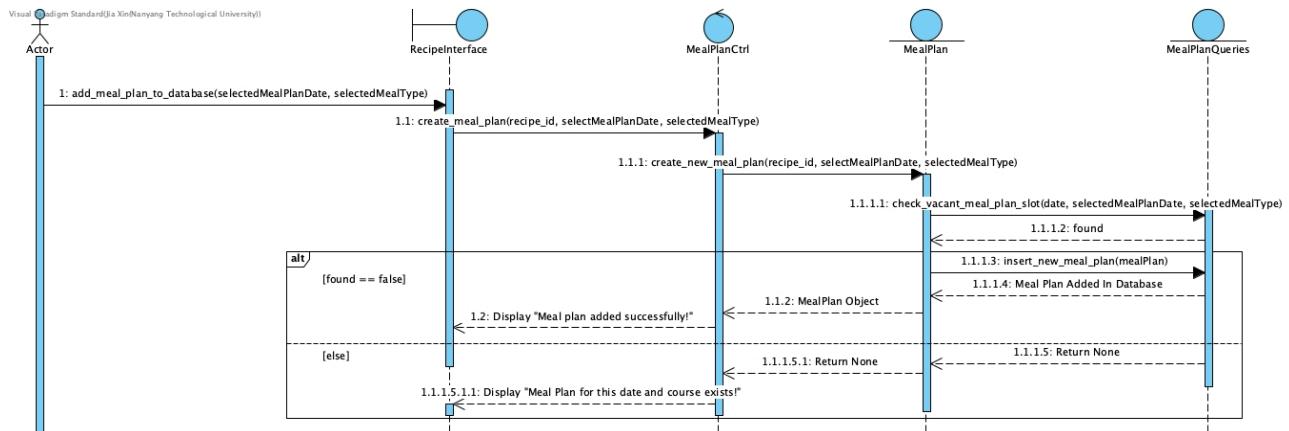
Registration and Create Profile Sequence Diagram



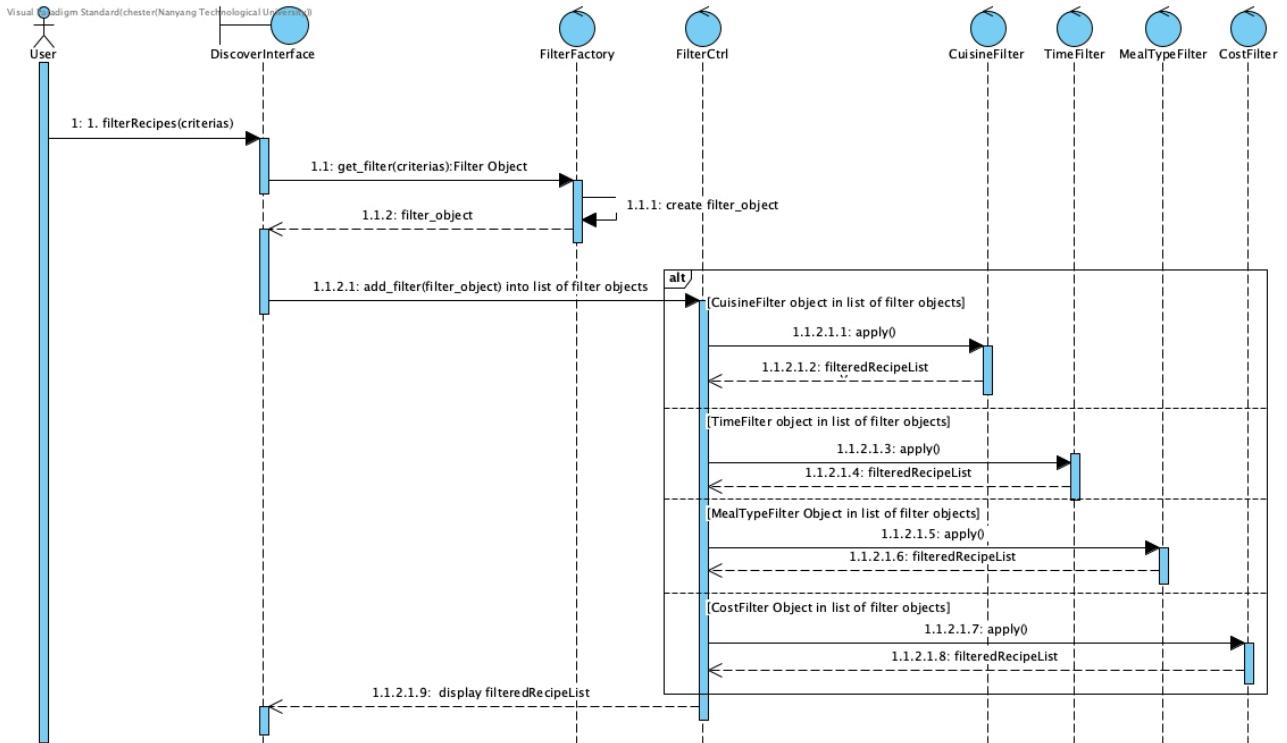
Login Sequence Diagram



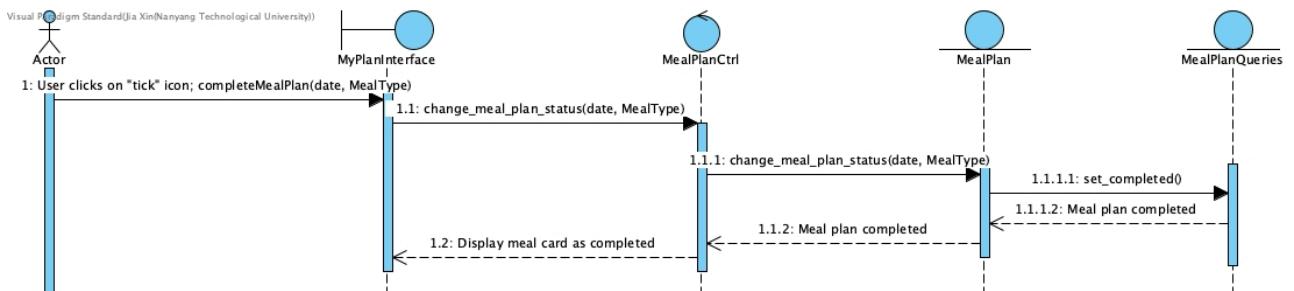
View Recipe Details Sequence Diagram



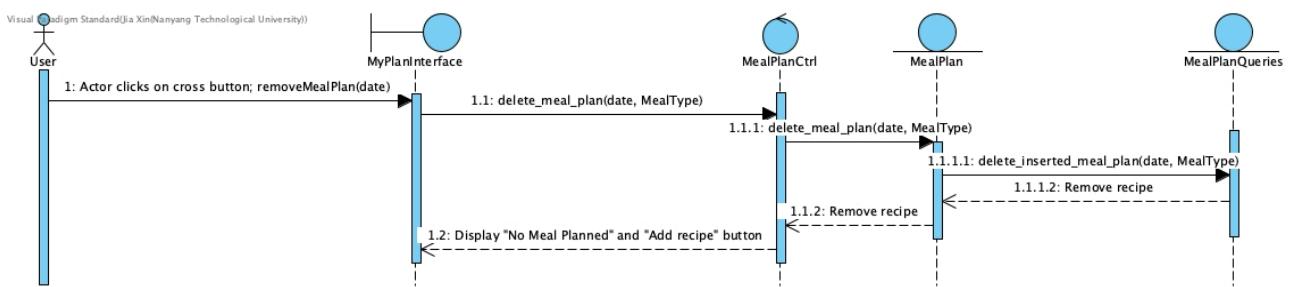
Add Meal Plan Sequence Diagram



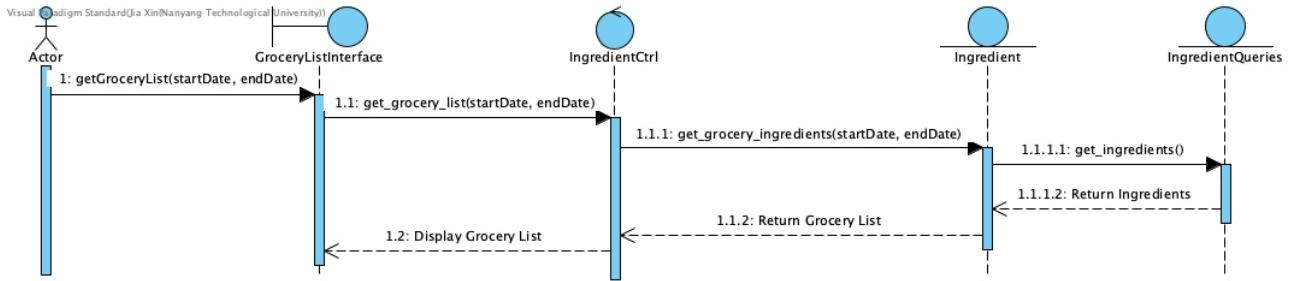
Filter Recipes Sequence Diagram



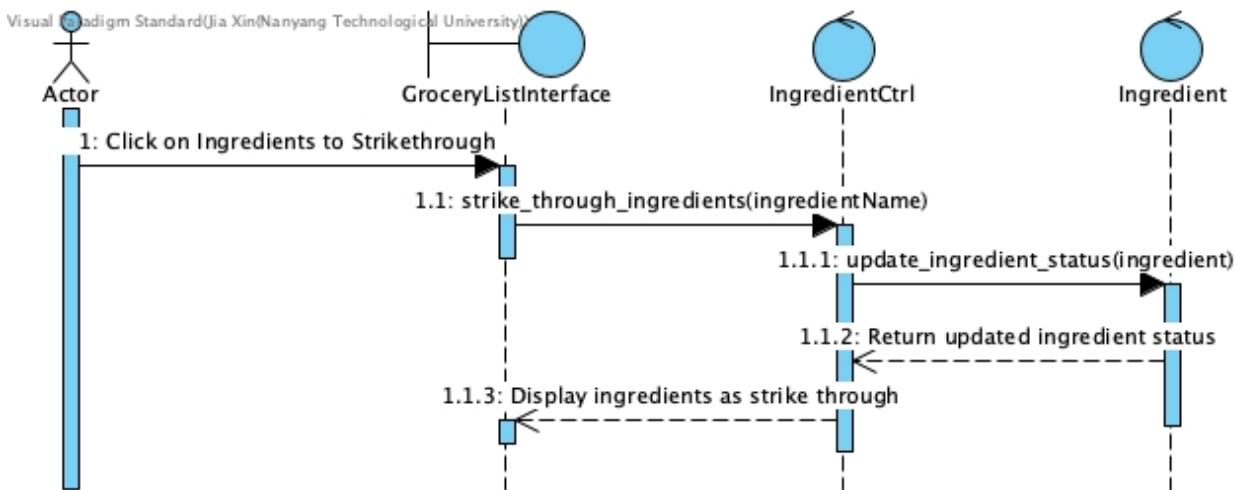
Complete Meal Plan Sequence Diagram



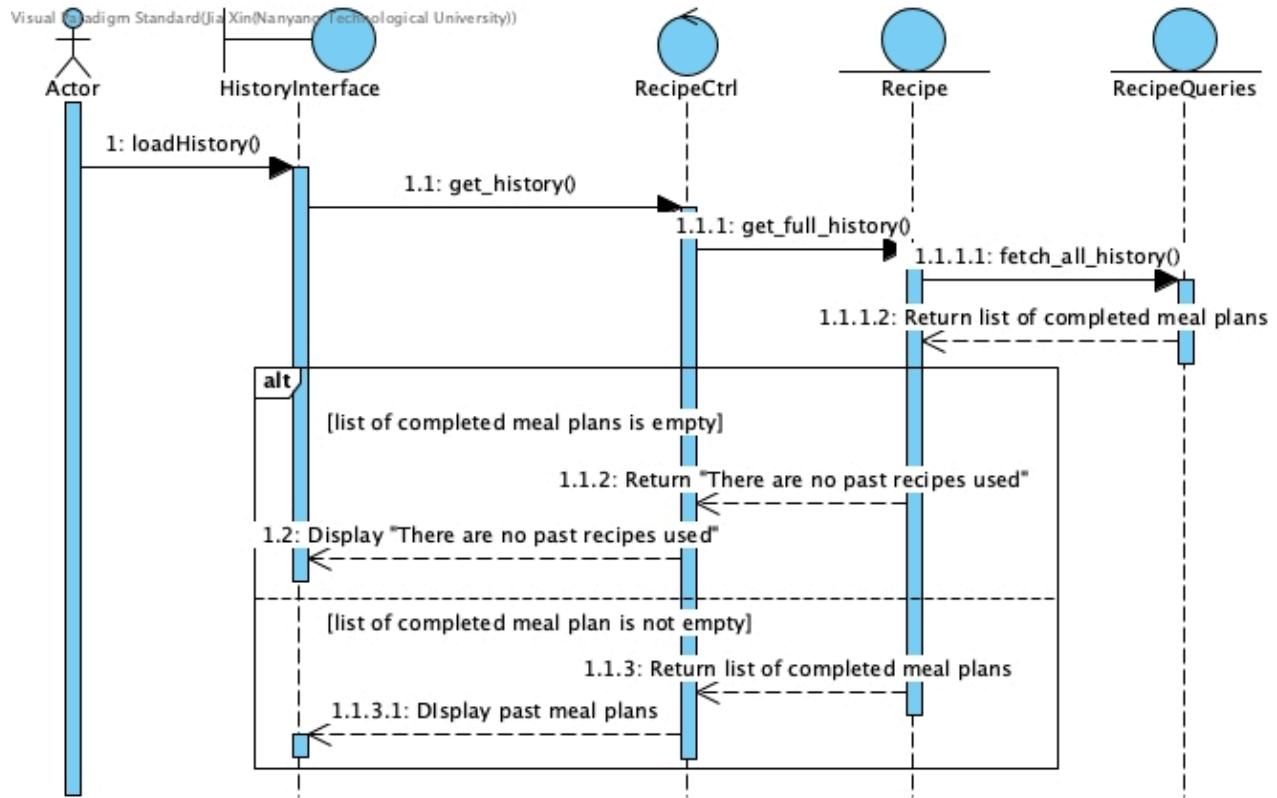
Remove Meal Plan Sequence Diagram



View Grocery List Sequence Diagram



Strikethrough Ingredients Sequence Diagram



View History Sequence Diagram

## Appendix C: To Be Determined List

There are no TBD items for this version of Software Requirement Specification.