Software Requirements Specification

for

MealMate

Version 1.0 approved

Prepared by:
Chew Zhen Yi (U2122697E)
Kunthamas Donchada (U2123242J)
Ng Ding Hei Ryan (U2121023D)
Tar Sreeja (U2123104B)
Yeoh Wei Yang (U2121112A)

Nanyang Technological University

20.02.2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Our website MealMate aims to serve the following three purposes:

1. It allows users to search for recipes, bookmark them and add their own recipes.
2. It allows users to search for nearby supermarkets.
3. It allows users to track their calorie consumption and maintain a healthy diet.

1.2 Document Conventions

Software Requirement Specification Format: This document follows the IEEE standard for SRS documentation.
Headers:
Font: Times New Roman
Font size: 18 pt. bold
Sub-headers:
Font: Times New Roman
Font size: 14 pt. bold
Content:
Font: Times New Roman
Font size: 12 pt.
Terminology conventions can be found in **Appendix A: Data Dictionary**.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for all stakeholders including the following:

1. Team Code Crafters - Team Code Crafters are the developers and testers of this website, as well as the documentation writers. Therefore, the SRS is intended for this team, for reference during the development and testing of the application. They will also use the SRS to ensure that the intended functional and non-functional requirements are satisfied by the final product.
2. Users - The SRS is intended for the users of this website to understand the various features offered by the website.

This document comprises the purpose of this website, the various functional and non-functional requirements to be satisfied by its features, analysis models as well as user-interface prototypes for better visualisation and comprehension of the product and its features.
The document must be read in sequence by all readers, with greater focus on sections 4 and 5, i.e., the functional and non-functional requirements to be fulfilled by the website. Readers must also take note of the analysis models and prototypes of the website in the appendix.

1.4 Product Scope

Mealmate is a web application that helps the user to track their caloric intake. Our web application also makes it easy for the user to plan and build a healthier diet with its built in

functionality to search for recipes and supermarkets nearby. The purpose of the web application is to promote and assist users in developing a healthier diet and empower then to take ownership of their own health.

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained product. This product idea originated from wanting an easy and convenient way to keep track of one's diet in order to lead a healthier lifestyle.

2.2 Product Functions

The user must be able to:

- Register for an account
- Log in and sign out of their accounts
- Edit their profiles when logged into their accounts
- Track their daily caloric intake via their profiles
- Search for recipes
- Search for supermarkets
- Submit feedback

2.3 User Classes and Characteristics

User classes include Guest and Registered users. Guest users will mainly use the website to search for recipes and/or nearby supermarkets. Registered users will mainly access these functions, as well as creating recipes and tracking calorie statistics. Registered users are frequent users who are health conscious, keen on monitoring consumption and calorie count patterns over extended periods of time through the website.

2.4 Operating Environment

MealMate will operate as a website which obtains information from Application Programming Interfaces (APIs) including Data.gov.sg, Spoonacular and MailChimp APIs.

2.5 Design and Implementation Constraints

MealMate will require the implementation of databases to operate, storing information of Registered users, Meals and Recipes. Team Code Crafters will have to craft the website using programming languages new to the team, and will be responsible for delivering and maintaining the software.

2.6 User Documentation

There will be a user guide for new and future developers to gain information on MealMate's functionality and database scheme. As MealMate will be implementing a RESTful API in order to access its own internal database, the manual will also include the API routes (i.e. GET, POST, PUT, DELETE) available to be called and any respective parameters needed to be included in the request if needed.

2.7 Assumptions and Dependencies

As mentioned in the Operating Environment section, MealMate will be making use of various external APIs. As we will only utilise the free plans of these APIs, there may be limitations in terms of the number of daily API calls and/or request type.

3. External Interface Requirements

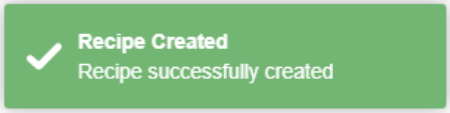
3.1 User Interfaces

MealMate will utilise several open-source resources:

- The sources of fonts and icons to be used in the web application will be from Google Fonts and Font Awesome.
- Screen layout constraints are dependent on the type of device the end-user is accessing MealMate from and will be determined by using open-source CSS framework, BootStrap, in order to achieve a responsive design.

Every page will have the navigation bar, where it will give the user easy access to all pages (i.e., Home, Recipes, Store Finder, Login) as such:

Temporary messages such as error/success messages will be displayed in the form of toast messages as such:



3.2 Hardware Interfaces

Mealmate will be usable on all types of operating system that offers web browsers.

3.3 Software Interfaces

MealMate is designed to work on any web browser. The system queries Spoonacular and Data.gov.sg database for information on recipes and supermarkets respectively. Allowing users

to search for their food recipes by inputting the food's name or its ingredients, and for supermarkets nearby their input postal code.

3.4 Communications Interfaces

MealMate is accessible over the Internet and will be deployed on the Hypertext Transfer Protocol Secure (HTTPS) while making use of Representational State Transfer (REST) API. HTTPS would secure data transfer and communication between the user's browser and the website. The REST API would then help query the database for the information that the end-user is requesting for.

4. System Features

Green - Actors
Blue - functionality
Underline - Highlights information for ease of reading

1. Main Menu

1.1. The system must allow the both guest and registered users to use the 2 search functions
  1.1.1. Search for recipes
  1.1.2. Search for nearby supermarkets

1.2. The system must allow the guest user to register their account.

1.3. The system must allow the registered user to login to their account.

1.4. The system must allow the registered user to view information about their history and daily caloric intake.

2. Registration

2.1. The guest user must be able to register for an account via the system.
  2.1.1. The guest user must provide the following information:
    2.1.1.1. Username
    2.1.1.2. Password
    2.1.1.3. Email Address

2.2. The system must validate all the required information the guest user has entered.
  2.2.1. Username of length 6-18 characters

2.2.2. Password of length 6-18 characters with at least one special character
2.2.3. Valid Email Address

2.3. The system must validate the following against its current registered users:
2.3.1. Username must be unique.
2.3.2. Email Address must not have been utilised before.

2.4. After validation, the system must inquire the newly registered user for their height and weight.
2.4.1. The system must calculate the newly registered user's BMI.
2.4.2. With the calculated BMI, the system must calculate the newly registered user's recommended daily caloric intake.

3. Login

3.1. The registered user must be able to login with their username and password.

3.2. The system must validate that all the required information has been filled up.

3.3. The system must check against the list of registered users and validate that the username and password are correct.
3.3.1. If both are valid, the registered user must be allowed to access additional features (i.e. add recipe, favourite recipe, view daily caloric intake).
3.3.2. If either one is invalid, the system must display an error message.

4. Search for Recipe

4.1. Both guest and registered users must be able to search for recipes via 2 methods:
4.1.1. Input up to 5 ingredients to obtain a random recipe
4.1.2. Input a dish name

4.2. The system must be able to display the recipe information.
4.2.1. If no recipes are found, the system must display an error message.

4.3. The registered user must be able to bookmark their favourite recipe.

5. Filter Recipes

5.1. Both guest and registered users must be able to filter for recipes via 2 methods:
5.1.1. Calories
5.1.2. Diet Type

5.2. The system must display recipes matching the guest/registered user's request

5.2.1. If no matching recipes are found, the system must display an error message.

5. Add a recipe

5.1. The system must display an option to add a recipe.

5.2. The registered user must be logged in to add a recipe.

5.3. The registered user must input the following details:

5.2.1. Name of Recipe
5.2.2. Preparation Time
5.2.3. Serving Portions
5.2.4. Calories
5.2.5. Description
5.2.6. Ingredients and their required measurements
5.2.7. Labelled step by step instructions.

6. Search for Supermarket

6.1. Both guest and registered users must be able to input their current location (postal code).

6.2. The system must be able to display nearby supermarkets within a 5km radius. The system must search for namely:

6.2.1. Giant
6.2.2. NTUC
6.2.3. Cold Storage
6.2.4. Sheng Shiong

6.3. Upon selecting a supermarket, the system must display a route to the supermarket on a map.

7. User Information

7.1. The system must be able to display the registered user's caloric intake for the past 31 days.

7.2. The system must display the registered user's current caloric intake for the day.

7.2.1. The system must display the recommended caloric intake based on the registered user's BMI.

7.2.2. The system must display how many calories the registered user has already consumed.
7.2.3. The system must display how far the registered user's calorie consumption is away from his recommended intake. 7.2.4. The system must reset the caloric intake for the registered user at 12am.

7.3 The registered user must be able to enter their caloric intake. This can be done via:
7.3.1. The registered user manually enters their caloric intake.
7.3.2. The registered user queries the system with the food that they consumed. The system must estimate the caloric intake and update the tracker.

7.4. The registered user must be able to view their own recipes.

8. Submit Feedback

8.1. Both guest and registered users must be able to submit feedback containing a message which is sent to the company.

8.2. The guest/registered user must be able to input their email address and a message.
8.2.1. If either of the fields are empty, the guest/registered user must not be able to submit the form.

8.3. The guest/registered user must be able to receive a copy of their feedback message that will be sent to their email.

9. Information to be processed

9.1. Height
8.1.1. Height format must be measured in cm, with no decimal places.

9.2. Weight
8.2.1. Weight format must be measured in kg, with an accuracy of up to 2 decimal places.

9.3. Calorie
8.3.1. Calorie format must be measured in kcal, with an accuracy of up to 2 decimal places.

10. Interface with other systems

10.1. Spoonacular API for recipes/food nutrition

10.2 Data.gov.sg API for supermarkets

10.3 MailChimp API for query/feedback submission

5. Nonfunctional Requirements

* 'User' in this section refers to both guest and registered users.

Performance	- The system must not crash when the user opens the web page. - The landing page's response time must be less than 4 seconds or less, including text and image rendering over an LTE connection, for all desktop browsers. - When recipes or nearby supermarkets are searched, the system must detect it and display the result within 2 seconds.
Portability	- The system must be able to run on all desktop/mobile browsers without any change in its behaviour and performance.
Reliability	- The system must perform without failure in 95 percent of use cases. - The system must not break due to the user's erroneous actions.
Maintainability	- The mean time to restore system following system failure must not be greater than 10 minutes - Maintenance must be conducted monthly to ensure that the system is always up to date - 3 days prior to maintenance, the system must inform the user of the upcoming maintenance and updates, stating the date and time accurately.
Security	- The personal data of users must not be disclosed without permission. - The password must be more than 8 characters and includes at least one special character. - The password must be encrypted.
Localisation	- The date format must be as follows: date.month.year.
Usability	- The system design is intuitively illustrated for easy navigation.

	- All features of the application are clearly displayed.
	- The system must offer informative feedback.
	  - To provide necessary feedback to the user when invalid inputs are detected.
	  - To display an appropriate error message when a certain process fails.
	- All features of the application will only be supported in an English language.

Appendix A: Data Dictionary

Term	**Definition**
User	A user can be of two types: 1. A guest user is a person using the website to search for recipes or supermarkets. 2. A registered user is a person using the website to track their caloric intake to maintain good health.
System	System refers to the 'MealMate' website.
Caloric intake	Caloric intake is the amount of energy consumed via food and beverage. It is measured in kilocalories (kcal). - Daily caloric intake is the amount of energy consumed via food and beverage **in a day**. - Recommended caloric intake is the daily caloric intake recommended by the website based on the user's BMI.
BMI	BMI stands for Body Mass Index. BMI = weight (kg) / (height x height) (sq. m) BMI is a metric used to determine if a person's body weight is healthy. BMI value ranges are classified into different categories: - BMI < 18.5 => underweight - 18.5 <= BMI <= 24.9 => healthy weight - 25 <= BMI <= 29.9 => overweight - BMI >= 30 => obese
Weight	Weight refers to a user's body weight. It will be input by the user in kilograms (kg).
Height	Height refers to a user's body height. It will be input by the user in centimetres (cm). Conversion to metres(m) for BMI calculation will be made by the system.

Recipe	A recipe is a set of instructions for preparing a particular dish, including a list of the ingredients required.
Ingredients	Ingredients are food items or substances that are combined to make a particular dish. E.g., vegetables, spices, etc.
Supermarket / Store	A supermarket is a self-service shop offering a wide variety of food, beverages and household products, organised into sections. The terms 'supermarket' and 'store' are used interchangeably.
GPS	GPS stands for Global Positioning System. It is a worldwide satellite-based navigation system that will aid in detecting a user's current location, suggesting nearby supermarkets

Appendix B: To Be Determined List

Source:
http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc