Software Requirements Specification

for

Random Meal Generator

Prepared by

ARULSURIYA S-23CSR022

ASWIN E-23CSR027

DILSHAN S-23CSR053

KONGU ENGINEERING COLLEGE

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Project Scope 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies…………………………………………………………………...3

3. System Features 3

3.1 User Authentication and Role-based Access Description………………………………………….3

3.2 Product Inventory Management Description………………………………………………………3

3.3 Order Placement and Processing Description……………………………………………………...4

3.4 Low Stock Notifications Description………………………………………………………………4

3.5 Sales Analytics Dashboard Description……………………………………………………………4

3.6 Supplier Database Integration Description………………………………………………………...4

4. External Interface Requirements…………………………………………………………...4

4.1 User Interfaces 4

4.2 Hardware Interfaces 5

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Non-functional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Requirements 5

6. Other Requirements 6

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements for the Random Meal Generator application. This application will suggest meals based on user preferences and available ingredients. The platform will include features such as user authentication, meal suggestions, recipe details, meal history tracking, and customizable meal plans.

## Intended Audience and Reading Suggestions

This document is intended for:

* **frontend and backend developers**: working on the system's implementation, including the app's functionality and data processing.**.**
* **Database Developer:** Manages meal data, user preferences, and ingredient availability.
* **UI/UX Designer:** Ensures an intuitive and engaging user interface for seamless meal selection.
* **Developers:** Understand system requirements to implement features such as meal suggestion algorithms and user preference handling.

## Project Scope

The Random Meal Generator is an application designed to suggest meal options based on user preferences and available ingredients at home. The system will generate random meal recommendations while considering dietary restrictions, allergies, and ingredient availability.

**1.4** **References**

* MongoDB Documentation: https://www.mongodb.com/docs/
* React Documentation: https://reactjs.org/docs/
* Node.js Documentation: https://nodejs.org/en/docs/

# Overall Description

## Product Perspective

* The system will be a **web-based and mobile-friendly application** for generating random meal suggestions based on user preferences and available ingredients.
* It will provide an **interactive and personalized experience**, ensuring users receive relevant meal recommendations.
* The app will integrate a **recipe database**, allowing users to explore meals with detailed instructions.

## Product Functions

* **User Authentication and Profile Management**: Users can create accounts, save preferences, and manage their ingredient inventory.
* **Random Meal Suggestions**: The app generates meals based on dietary preferences and available ingredients.
* **Ingredient Inventory Management**: Users can add, update, and delete ingredients they have at home.
* **Dietary and Allergy Preferences**: Users can specify dietary restrictions (e.g., vegan, gluten-free) and excluded ingredients.
* **Nutritional Information Display**: Provides details like calorie count and macronutrients for suggested meals.
* **API Integration**: Optionally integrates with external databases for more diverse recipe recommendations.
* **Meal History & Favorites Meals**: Lists past suggestions and saved meals.

## User Classes and Characteristics

* **General Users:** Can explore random meal suggestions, manage ingredient lists, and save favorite meals.Have access to personalized recommendations based on saved preferences and dietary needs.
* **Admin:** Manages the recipe database, verifies meal suggestions, and ensures the system runs smoothly.

**2.4 Operating Environment**

* **Frontend**: React.js with Tailwind CSS for a modern, responsive user interface.
* **Backend**: Node.js with Express.js for handling API requests and business logic.
* **Database**: MongoDB (NoSQL) for efficient storage of user preferences, ingredients, and recipes.
* **Compatibility**: Optimized for Chrome, Firefox, Edge, and mobile browsers.

**2.5 Design and Implementation Constraints**

* The system must provide real-time meal suggestions based on user inputs.
* Data security must be ensured, especially for user authentication and saved preferences.
* The platform must be responsive, supporting both desktop and mobile devices.
* The meal database must be maintained and periodically updated for accurate suggestions.

## User Documentation

* A user manual with step-by-step guidance on using the application.
* Online tutorials explaining features such as meal suggestions, ingredient management, and preference settings.

## Assumptions and Dependencies

* A stable internet connection is required for retrieving and updating recipes.
* External APIs, if used, must be available and functional for enhanced meal suggestions.

# System Features

## User Authentication

## Description: Users can register, log in, and manage their profiles.

## Input: Email, password, and optional profile details (e.g., dietary preferences).

## Output: Authentication success/failure message.

## Preconditions: The user must provide a valid email and password.

## Postconditions: A user session is created, allowing access to personalized features.

## User Interaction: Users fill out a registration or login form and submit details.

## 3.2 Ingredient Management

## Description: Users can add, update, or delete ingredients from their pantry to get relevant meal suggestions.

## Input: Ingredient name, quantity, and optional expiry date.

## Output: Updated pantry list reflecting changes.

## Preconditions: The user must be logged in.

## Postconditions: The ingredient list is updated in the database.

## User Interaction: Users enter ingredients manually, select from a predefined list, or scan barcodes (if implemented).

## Meal Suggestion System

## Description: The system randomly suggests a meal based on available ingredients and user preferences.

## Input: User's ingredient list, dietary preferences, and allergy restrictions.

## Output: A suggested meal with recipe details, including preparation steps and estimated cooking time.

## Preconditions: The user has added ingredients to their pantry.

## Postconditions: A meal is displayed based on the provided inputs.

## User Interaction: Users click "Generate Meal", and the system provides a random meal suggestion.

## Advanced Search and Filtering

* **Description:** Users can search for meals based on specific criteria for a more refined meal suggestion experience.
* **Inpu**t: Filters such as cooking time, number of ingredients, meal type (breakfast, lunch, dinner), cuisine, difficulty level.
* **Output:** A list of meals that match the selected criteria.
* **Preconditions:** The user must enter at least one search filter.
* **Postconditions:** A filtered list of meals is displayed.
* **User Interaction:** Users input search criteria and receive a list of relevant meals.

## Meal Planning and Favorites

## Description: Users can save favorite meals and create meal plans for future reference.

## Input: Selected meal(s) to add to favorites or weekly meal plan.

## Output: Updated meal plan or favorites list.

## Preconditions: The user must be logged in.

## Postconditions: The meal is saved to the user's meal plan or favorites list.

## User Interaction: Users click "Save to Favorites" or "Add to Meal Plan" for future access.

## Nutritional Information Display

* **Description:** Displays the nutritional breakdown (calories, macronutrients) for each suggested meal.
* **Input:** Meal selection.
* **Output:** A breakdown of calories, proteins, carbs, fats, and other nutritional details.
* **Preconditions:** The system must have nutritional data for the selected meal.
* **Postconditions:** The nutritional information is displayed to the user.
* **User Interaction:** Users view nutrition details on the recipe page.

# External Interface Requirements

## User Interfaces

* **Home Screen**: Displays a welcome message and quick meal generation option.
* **Dashboard**: Shows user preferences, saved meals, and pantry inventory.
* **Meal Suggestion Screen**: Displays randomly suggested meals with recipe details.
* **Ingredient Management Screen**: Allows users to add, update, and delete ingredients from their pantry.
* **Meal Planning Screen**: Enables users to create and manage weekly meal plans.
* **Search & Filter Screen**: Provides options to search for meals based on dietary needs, cuisine, and cooking time.
* **User Profile Screen**: Allows users to manage preferences, dietary restrictions, and saved meals.

## Hardware Interfaces

* No specialized hardware is required

## Software Interfaces

 **Database Integration**: MongoDB (NoSQL) for storing user data, ingredients, and recipes.

 **Third-Party APIs**: Integration with external recipe databases for expanded meal suggestions.

 **Authentication**: OAuth for Google/Facebook login options.

**4.4 Communication Interfaces**

* Frontend communicates with the backend via RESTful APIs.
* **API Integration**: Fetches additional recipes and nutritional data from external services.

# Other Nonfunctional Requirements

## Performance Requirements

* The system should support up to **1,000 concurrent users** without performance degradation.
* Meal suggestions should be generated within **2 seconds** of user input.
* The ingredient management system should update in **real-time** when users add, modify, or delete ingredients.

## Security Requirements

* **Secure data transmission** via HTTPS to prevent unauthorized access.
* **User authentication** using JWT tokens for secure session management.
* **Data encryption** for storing sensitive information such as user preferences and saved recipes.
* **Regular backups** of user data to prevent loss of saved meals and preferences.

## Usability Requirements

* **Simple and intuitive UI** for easy meal selection and ingredient management.
* **Multi-platform support**, ensuring compatibility across desktops, tablets, and mobile devices.
* **Offline mode** allowing users to view saved recipes and meal plans without an internet connection.

## Software Quality Attributes

* **User-friendly interface** with clear navigation and interactive design.
* **Reliable performance** under high loads, ensuring smooth meal suggestion generation.
* **Scalability** to support an increasing number of users and recipe expansions.
* **Maintainability** with clean and modular code for easy updates and feature enhancements.

# Other Requirements

* The platform must be compatible with modern browsers and mobile devices.
* Secure session management is required to prevent unauthorized access.