

Software Requirements Specification

For

OnlyFood

Version 1.0 approved

Prepared by SDAA-TEAM32

Nanyang Technological University

10 November 2024

Table Of Contents

1. Introduction.....	2
1.1. Purpose.....	2
1.2. Document Conventions.....	2
1.3. Intended Audience and Reading Suggestions.....	3
1.4. Product Scope.....	3
2. Overall Description.....	4
2.1. Product Perspective.....	4
2.2. Product Functions.....	4
2.3. User Classes and Characteristics.....	7
2.4. Operating Environment.....	9
2.5. Design and Implementation Constraints.....	9
3. External Interface Requirements.....	10
3.1 User Interfaces.....	10
3.2 Hardware Interfaces.....	11
3.3 Software Interfaces.....	11
3.4 Communications Interfaces.....	11
4. System Feature and Requirements.....	13
4.1. User Registration and Authentication.....	13
4.2. Profile Management.....	14
4.3. Post Creation and Sharing.....	15
4.4. Nutritional Tracking.....	16
4.5. Search and Explore.....	17
4.6. Notifications.....	17
4.7. Admin Controls.....	18
5. Other Non-Functional Requirements.....	19
5.1 Performance Requirements.....	19
5.2 Security Requirements.....	21
5.3 Usability Requirements.....	21
5.4 Reliability and Availability Requirements.....	22
5.5 Scalability Requirements.....	23
6. Appendices.....	23
6.1. Appendix A: Data Dictionary.....	24
6.2. Appendix B: Use Case Diagrams and Descriptions.....	28
6.2.1. Use Case Model.....	28
6.2.2. Use Case Descriptions.....	28

1. Introduction

1.1. Purpose

The purpose of this Software Requirement Specification (SRS) is to present a comprehensive description of OnlyFoods, a web-based social media application focused on creating a community for food enthusiasts to share, discover, and track meals and recipes. This platform aims to provide users with a personalized experience for posting recipes, tracking daily calorie and nutrient intake, and engaging with other users through interactions such as likes and comments.

This SRS outlines the full scope of the OnlyFoods system. It includes detailed explanations of product functionalities, operational requirements, design constraints, and interface specifications. Additionally, this document covers requirements for user authentication, profile management, post interaction, nutrition tracking, and administrative functions. This SRS serves as the foundation for the design, development, and implementation of the OnlyFoods application, ensuring that the system meets user needs and performance expectations.

1.2. Document Conventions

Level 1 Header - Font: Times New Roman, size: 20, style: Bold

Level 2 Header - Font: Times New Roman, size: 17, style: Bold

Level 3 Header - Font: Times New Roman, size: 12, style: Normal Text

Text - Font: Times New Roman, size: 12, style: Normal Text, align: Left align

Each requirement is uniquely identified for traceability, and high-level requirements inherit priority unless otherwise stated. This document adheres to IEEE Standard 830-1998 for Software Requirements Specification.

1.3. Intended Audience and Reading Suggestions

This SRS is intended for developers, testers, and customers involved in the OnlyFoods application. Developers will use it to understand detailed system requirements and behaviors, while project managers can leverage it to monitor feature scope and priorities. Testers will utilize this document to design and validate test cases ensuring functionality and user experience align with expectations. Customers can utilize this document to track and understand the software requirements of the development phase to check if it aligns with their own requirements. Readers are encouraged to focus on sections most relevant to their roles, such as functional requirements for developers and usability specifications for testers.

1.4. Product Scope

OnlyFoods is a social media application designed to connect food enthusiasts by enabling users to share, discover, and interact with food-related content. The platform allows users to create and share meal posts, track their daily nutritional intake, and set personal dietary goals. Users can also save posts, follow other users, and manage their profiles, creating a personalized and engaging food-sharing experience.

The key objectives of OnlyFoods are to foster a community centered around healthy eating, recipe sharing, and dietary goals while providing tools to monitor daily nutrition and connect with like-minded individuals. OnlyFoods also includes an administrative interface for managing flagged content and user accounts to ensure a safe and positive environment.

1.5. References

IEEE 830-1998: IEEE Recommended Practice for Software Requirements Specifications.

Package dependencies specified in package.json and package-lock.json for Express, Supabase, bcrypt, EJS, and other essential modules.

2. Overall Description

2.1. Product Perspective

OnlyFoods is an independent web-based platform designed as a social media application for food sharing, with a focus on culinary creativity and nutritional insights. OnlyFoods integrates with third-party services, such as Supabase for user authentication and database management, and uses multiple dependencies including Express, bcrypt, and EJS for server and front-end rendering support.

This platform allows users to create profiles, follow other users, and share food-related posts that include ingredients, photos, and nutritional information. OnlyFoods also features personalized tracking of dietary goals, making it a holistic space for food discovery, recipe sharing, and health management.

The application architecture is built on a Node.js backend, which utilizes RESTful API endpoints to handle user actions, and a responsive front-end that ensures a seamless user experience across different devices and screen sizes.

2.2. Product Functions

OnlyFoods provides the following core functionalities:

1. User Registration and Authentication:

- Users can register and log in using email and password, with bcrypt providing secure password hashing.
- The system verifies user accounts, manages sessions securely, and enforces email uniqueness to prevent duplicate accounts.
- Users can log out, and sessions are maintained to ensure secure access.

2. Profile Management:

- Users can update their profile, including details such as username, profile picture, and personal dietary goals like calorie targets.
- Profiles display user statistics, posts, and user details.
- Users can view their own posts as well as saved posts, allowing for easy access to previously liked or bookmarked content.
- Users can modify personal information, including height, weight, and age, with real-time updates on calorie targets based on these metrics.

3. Post Creation:

- Users can create posts with images, providing detailed information about ingredients and nutritional values, as well as their recipes.
- Each post allows the user to specify ingredients, quantities, descriptions and categories, which the system verifies to ensure all fields are filled and values are valid.
- Posts can be liked, commented on, saved, and flagged for inappropriate content.

4. Post Interaction:

- Users can interact with other posts by liking, commenting, saving, or flagging posts for review.
- Liked and saved posts reflect real-time feedback, updating the like count and save status accordingly.
- Flagging a post triggers a reporting form where users select a reason, activating the button upon submission.
- All interactions are persistent, allowing users to revisit posts with previous interactions accurately reflected.

5. Nutritional Tracking:

- OnlyFoods automatically calculates calorie and macronutrient content (e.g., protein, carbohydrates, fats) based on the ingredients selected in each post.
- Users can track daily and long-term dietary goals through their posts and profile.
- The system allows users to log daily meals, increasing daily calorie count and updating the progress bar accordingly.

6. Daily Meal Management:

- Users can add meals to their daily intake with specific ingredients and quantities.
- Empty field checks are in place, ensuring that all meal entries are valid and complete.
- Ingredients can be added or removed, allowing flexibility in meal composition.
- Daily meals reset after a set period, supporting refreshed daily tracking.

7. Search and Explore:

- Users can search for posts based on title names, descriptions, categories, or specific users.
- The search functionality enables users to discover a variety of food content and encourages community engagement around food and nutrition.

8. Notification Management:

- Users receive notifications for interactions on their posts, including likes, comments, and flags.
- Notifications update in real-time, allowing users to stay engaged with community interactions.
- Users can mark notifications as read or clear all notifications for a streamlined experience.

9. Admin Controls:

- Administrators can review flagged posts and manage user reports, applying suspensions or bans if necessary.
- Admins can filter and search flagged posts for efficient content moderation.
- Admins can issue warnings, apply temporary suspensions, or permanently ban users based on repeated violations.
- Upon review, administrators can mark reports as resolved or take action based on community guidelines.

10. Account Suspension and Ban Management:

- Users with flagged content or repeated violations may receive warnings, suspensions, or permanent bans, managed by the admin.
- Suspended accounts display a message upon login explaining the suspension duration, while banned accounts are restricted from logging in.
- After a suspension period, users can log in again, resuming their normal activity on the platform.

2.3. User Classes and Characteristics

The primary user classes for OnlyFoods include:

General Users: Individuals interested in exploring and sharing food content, tracking dietary habits, and following other users. They may have varying levels of technical familiarity but primarily seek an intuitive, easy-to-navigate interface.

Content Creators: Users who actively post food-related content, including recipes and meal preparation tips. They rely heavily on post creation tools, multimedia support, and interaction tracking (likes, shares, and comments) to engage their audience.

Diet-Conscious Users: These users focus on tracking their daily nutritional intake and may engage more with OnlyFoods' tracking features to set and meet personal dietary goals.

Admins: Users with elevated privileges who monitor platform content and user interactions to maintain a safe, supportive community. They have access to moderation tools for managing reports and enforcing community guidelines.

User Class	Features
General Users	<ul style="list-style-type: none">- Primary focus on exploring and engaging with food content.- Seek intuitive and easy navigation.
Content Creators	<ul style="list-style-type: none">- Frequently create and share food-related posts, including recipes and tips.- Rely on tools for multimedia, post creation, and audience engagement (likes, comments, shares).
Diet-Conscious Users	<ul style="list-style-type: none">- Focus on tracking daily nutritional intake.- Use dietary tracking features to set and meet personal health goals.
Admin	<ul style="list-style-type: none">- Maintain a safe and supportive community environment.- Have access to moderation tools for managing reports and enforcing guidelines.- Monitor and manage user content and interactions.

2.4. Operating Environment

OnlyFoods is a web-based application accessible on desktop and with responsive design. The application operates in the following environments:

Client-Side: Compatible with major browsers (Chrome, Firefox, Safari, Edge) and optimized for both desktop.

Server-Side: Hosted on a Node.js and Express environment, with a PostgreSQL database managed by Supabase. The backend requires an internet connection and access to the Supabase API for data storage and retrieval.

Security Dependencies: Utilizes bcrypt for password encryption and optionally supports two-factor authentication (2FA) for enhanced user security.

2.5. Design and Implementation Constraints

The system is subject to the following constraints:

Platform Dependency: OnlyFoods relies on the Supabase service for backend data management and authentication, which imposes constraints on scalability and storage as user numbers grow.

Security Standards: OnlyFoods must comply with data protection regulations, such as the PDPA, to ensure user data privacy and security. Passwords are encrypted, and data encryption is employed for sensitive information.

Performance Constraints: To enhance user experience, page load times must not exceed 5 seconds under average server load, with nutritional calculations processed within 1 second after a post is created.

2.6. Assumptions and Dependencies

This document makes the following assumptions and identifies dependencies:

Third-Party Services: OnlyFoods assumes uninterrupted access to Supabase for authentication and database storage. The system also depends on stable external services to maintain functionality.

User Authentication: It is assumed that email verification and password hashing using bcrypt are sufficient for ensuring basic security.

Client Environment: The application assumes that users will access OnlyFoods on a device with a stable internet connection and a modern browser that supports JavaScript.

Performance of External APIs: OnlyFoods relies on any additional APIs or data sources for ingredient information, calorie calculations, or nutritional data. Any disruptions or rate limitations from these sources could affect the application's functionality

3. External Interface Requirements

3.1 User Interfaces

OnlyFoods has several key user interface components, optimized for ease of navigation:

Login and Signup Pages: Allow users to create new accounts or sign in with existing credentials. The signup page includes fields for username, email, and password, with error prompts for invalid inputs.

Profile Page: Displays user information, including posts, daily meals, personal information and dietary targets. Users can edit their profile and access their dietary tracking data.

Main Page: The main interface for users to browse and view posts, enhancing the discovery of new recipes and users within the OnlyFoods community. Also contains the option for users to access the function to upload their own posts with a pop-up interface.

Notification Page: Allows users to view notifications from other user interactions on their own posts, as well as warnings received from the admin after violating certain terms.

3.2 Hardware Interfaces

OnlyFoods does not require specific hardware interfaces beyond general web accessibility. It is designed to be accessible on any device with internet connectivity, including desktop computers, tablets, and mobile devices.

3.3 Software Interfaces

OnlyFoods integrates with several software libraries and APIs:

Supabase API: For handling user authentication, database interactions, and data retrieval.

Express: Provides the server-side framework for handling HTTP requests.

Multer: Manages file uploads for post images.

bcrypt: Encrypts user passwords for secure storage.

EJS: A templating engine for rendering HTML with embedded JavaScript for dynamic content updates.

3.4 Communications Interfaces

OnlyFoods relies on secure HTTP and HTTPS protocols for data transmission. Sensitive data such as login credentials and personal information are encrypted to protect user privacy.

RESTful API Calls

1. Frontend to Backend Communication:

The frontend of OnlyFoods communicates with the backend using RESTful API calls. These calls facilitate various functionalities, including user authentication, profile updates, post creation, and nutritional tracking. OnlyFoods relies on Supabase as its backend, hosted locally during development at localhost:3000.

Endpoint	Method	Description
/login	POST	Authenticates the user using username and password, handling errors like incorrect password or account issues.
/profile	GET	Fetches and displays the user's profile details, posts, and daily meals.
/update-profile-pic	POST	Uploads a new profile picture and updates it in the database.
/updateProfile	POST	Updates user profile details like username, height, weight, target weight, and birthday.
/addMeal	POST	Adds a new daily meal with title, ingredients, calories, and meal type to the database.
/signup	GET	Renders the signup page.
/setup-profile/:user_id	GET	Renders the profile setup page for the specified user.
/setup-profile/:user_id	PUT	Updates user profile during initial setup, including uploading a profile picture if provided.
/main	GET	Displays the main page with a list of all posts ordered by creation date.
/addPost	POST	Allows users to add a new post with image, title, description, category and ingredients
/getPosts	GET	Fetches all posts in JSON format, optionally providing signed URLs for private images.

4. System Feature and Requirements

4.1. User Registration and Authentication

Functional Requirements

1. The users must be able to sign up for an account if they haven't yet
 - 1.1. The users must be able to key in their username
 - 1.2. The users must be able to key in an email
 - 1.3. The users must be able to set up their password for account
 - 1.4. The users must be able to confirm his password during sign up
 - 1.4.1.
2. The users must be able to be redirect to profile setup page upon successfully signup
 - 2.1. The users must be able to set up their account name
 - 2.2. The users must be able to set up their birthday
 - 2.3. The users must be able to set up their current weight and target weight
 - 2.4. The users must be able to set up their profile picture
 - 2.4.1. The users must be able to upload their profile picture in .png format.
3. The System must verify the account's legitimacy to allow login
 - 3.1. The System must verify the user's email to allow login
 - 3.1.1. If the user's email is invalid, the system should not allow the users to login
 - 3.1.2. When the user's email is invalid, the system shall display an error message
 - 3.2. If the user's account is suspended, the system should not allow the user to log in during the suspended period.
 - 3.2.1. If the user's account has passed the suspended period, he/she should be able to log in the account again.
 - 3.3. If the user's account is banned, the system should not allow the user to log in.
 - 3.4. If the account is an admin account, the system shall redirect the user to the admin page upon successful login.

- 3.5. If the account is not an admin account, the system shall redirect the user to the main user page upon successful login.

4.2. Profile Management

Functional Requirements

1. The user should be able to manage their profile and personal posts.
 - 1.1. The user should be able to view and interact with their own posts on their profile.
 - 1.1.1. The user shall be able to view a list of posts they have created in the "My Posts" section.
 - 1.1.1.1. The user shall be able to click on a post to expand it, displaying post details including food title, ingredients, total calories, and any additional description.
 - 1.2. The user should be able to view and interact with saved posts on their profile.
 - 1.2.1. The user shall be able to view a list of posts they have saved in the "Saved Posts" section.
 - 1.2.1.1. The user shall be able to click on a saved post to expand it and view its details.
 - 1.2.1.1.1. The user shall be able to dislike a post in their saved posts, and the system shall decrease the like count accordingly.
 - 1.2.1.1.2. The user shall be able to unsave a post, which removes it from the "Saved Posts" section.
2. The user should be able to modify their profile details.
 - 2.1. The user should be able to update their profile picture.
 - 2.1.1. Upon successful change, the system shall update and display the new profile picture on the profile.
 - 2.2. The user should be able to update their personal information on their profile.
 - 2.2.1. The user shall be able to change their birthday, height, and weight.

- 2.2.1.1. The system shall recalculate the user's age based on the updated birthday and display it on the profile.
- 2.2.1.2. The system shall recalculate the user's target calorie intake based on updated height and weight.
- 2.2.1.3. The system shall display updated information for age and target calories on the profile.

4.3. Post Creation and Sharing

Functional Requirements

1. The user must be able to create posts on the main page.
 - 1.1. The user must be able to add a unique title and description for the post.
 - 1.1.1. The system shall prevent the user from leaving the tile blank.
 - 1.2. The user must be able to select ingredients from a list or add custom ingredients with specified quantities.
 - 1.2.1. The system shall prevent the user from leaving the quantity blank.
 - 1.2.2. The system shall prevent the user from inputting negative values in the quantity field.
 - 1.3. The user must be able to select a category for the food (e.g., breakfast, lunch).
 - 1.3.1. The system shall prevent the user from leaving the category blank.
 - 1.4. The user must be able to upload a picture of the food.
 - 1.4.1. The system shall prevent the user from leaving the picture blank.
 - 1.5. Upon successful creation of post, the main UI must be able to auto render the post with correct post picture, title and total calories.
2. The user should be able to interact with other posts.
 - 2.1. The user should be able to like and do unlike posts.
 - 2.1.1. When the user likes a post, the like button shall respond visually (e.g., changing color or style) to indicate the post is liked.
 - 2.1.2. The like count shall automatically update to reflect the current number of likes.

- 2.1.3. When the user unlikes a post, the like button shall return to its unliked state, and the like count shall decrease accordingly.
 - 2.2. The user should be able to comment on posts.
 - 2.2.1. Comments shall be immediately visible on the post after submission.
 - 2.3. The user should be able to save posts to their collection.
 - 2.3.1. When a user saves a post, the save button shall visually indicate that the post is saved (e.g., changing color or displaying an icon).
 - 2.3.2. The save button shall toggle to an unsaved state if the user clicks to unsave the post.
 - 2.4. The user should be able to flag inappropriate posts and submit a report.
 - 2.4.1. When the user clicks the flag button, a report form shall appear, allowing the user to choose the type of report (e.g., spam, offensive content).
 - 2.4.2. After submitting a report, the flag button shall visually indicate that the post has been flagged (e.g., by changing color or displaying a checkmark).
- 3. The System should persist user interactions.

Saved, liked, and flagged states should persist upon reopening a post.

Post owner's details (profile picture, username) and post details (title, ingredients, calories) should be correctly rendered.

4.4. Nutritional Tracking

Functional Requirements

- 1. Users should be able to log their daily meals on their own profile page.
 - 1.1. The user must be able to add a unique title
 - 1.1.1. The system shall prevent the user from leaving the tile blank
 - 1.2. The user must be able to select ingredients from a list or add custom ingredients with specified quantities.
 - 1.2.1. The system shall prevent the user from leaving the quantity blank.
 - 1.2.2. The system shall prevent the user from inputting negative values in the quantity field.

- 1.3. The system should calculate and display total daily calorie intake based on inputs.
- 1.4. The user must be able to select a meal type for the food (e.g., breakfast, lunch).
 - 1.4.1. The system shall prevent the user from leaving the meal type blank.
- 1.5. Upon successful creation of a daily meal, the profile UI must be able to auto render the daily meal with correct title, ingredients, and total calories.
- 1.6. The daily meals should be automatically deleted by the system every 24 hours.
2. The profile page should display the user's daily calorie target based on their profile details (e.g. age, height, weight and target weight).
 - 2.1. Daily calorie targets should update in real time when profile details (e.g. age, height, weight and target weight) are modified.
3. The profile page should display a progress bar that visually indicates daily calorie intake relative to the target.
 - 3.1. The progress bar should update instantly when new meals are added or removed.

4.5. Search and Explore

Functional Requirements

1. The System must allow users to search and filter posts.
 - 1.1. The user should be able to filter posts by post category.
 - 1.2. The user should be able to search posts by post title.

4.6. Notifications

1. The System should notify users of interactions on their posts.
 - 1.1. The user should receive notifications upon likes on their posts.
 - 1.2. The user should receive notifications upon comments on their posts.
 - 1.3. The user should receive notifications upon flags on their posts.
 - 1.4. The user should be able to mark notifications as read.
 - 1.4.1. Once the notification has been marked as read, it should disappear in their notification session.

- 1.5. The user should have an option to clear all notifications at once.
 - 1.5.1. Once the notifications have been cleared, they should disappear in their notification session.

4.7. Admin Controls

Functional Requirements

- 1. The administrator should be able to view and moderate posts.
 - 1.1. The system shall provide admins with an Admin Dashboard where they can view user posts.
 - 1.2. The system shall highlight posts that have been flagged by users for admin attention.
 - 1.2.1. Admins shall be able to open and view details of any flagged post, including:
 - 1.2.1.1. Post content and images.
 - 1.2.1.2. The post owner's username and profile.
 - 1.2.1.3. Reasons for which users flagged the post.
 - 1.2.2. Admins shall be able to close the post details and return to the admin dashboard without taking action.
 - 1.3. While reviewing a flagged post, admins shall have the following options:
 - 1.3.1. Issue a Warning: Send a warning message to the post owner.
 - 1.3.2. Suspend Account: Temporarily suspend the post owner's account.
 - 1.3.3. Ban Account: Permanently ban the post owner's account.
 - 1.4. Admins shall be able to select the appropriate action based on the violation severity.
- 2. The administrator should be able to manage user accounts.
 - 2.1. The administrator should be able to monitor users who have had posts flagged.
 - 2.2. The administrator shall have the ability to perform the following actions on user accounts:
 - 2.2.1. Issue a Warning: Send a formal warning message.

- 2.2.2. Suspend Account: Temporarily disable the account.
- 2.2.3. Ban Account: Permanently disable the account.
- 2.3. The administrator should be able to suspend user accounts.
 - 2.3.1. Admins shall be able to suspend a user account for a specific duration, selectable or input by the admin.
 - 2.3.2. The system shall record the suspension duration and reason.
 - 2.3.3. During suspension, the user shall be prevented from logging in.
 - 2.3.4. If a suspended user attempts to log in, the system shall display a suspension message indicating the remaining duration.
- 2.4. The administrator should be able to ban a user account, with the option to set:
 - 2.4.1. An indefinite ban.
- 2.5. The system shall prevent banned users from accessing their accounts.
- 2.6. If a banned user attempts to log in, the system shall display a ban message with relevant details.

5. Other Non-Functional Requirements

5.1 Performance Requirements

Section	Performance Requirement
Web Client	
Page Load Time	The OnlyFoods website must load the homepage within 3 seconds on average for a smooth user experience. This includes pre-loading recent posts and profile data if the user is logged in.
Post Interaction	Liking, saving, or commenting on a post should trigger an immediate response on the interface, with feedback (e.g., like count increment or comment addition) appearing within 2 seconds.
Image Uploads	Uploading a new image for a post or profile picture should complete

	within 5 seconds on a stable internet connection.
Search and Filtering	When users search for posts by ingredients, dietary goals, or users, the search results must be displayed within 2 seconds.
Navigation Across Pages	Navigating between the main pages (e.g., Home, Profile, Admin Dashboard) should not exceed 3 seconds per page load.

Section	Performance Requirement
Backend Server	
Database Queries	The backend server should respond to user queries (e.g., fetching user profiles, retrieving posts, calculating nutritional content) within 5 seconds to avoid delays in the frontend.
Automated Nutritional Calculations	The system should calculate nutritional information (e.g., calories, macronutrients) for a post based on user-selected ingredients within 2 seconds after submission.
Data Synchronization	The backend should synchronize any user changes (e.g., new posts, profile updates) and update the relevant data across the application within 2 seconds.
Notification Delivery	Notifications, such as alerts for likes, comments, or flags, should be delivered to the recipient's notification center within 2 seconds of the triggering event.
Admin Operations	Admin actions, such as flagging content or managing user accounts, should be processed immediately, with updates reflected on the admin dashboard within 2 seconds.

5.2 Security Requirements

Section	Performance Requirement
Data Encryption	Data at Rest: OnlyFoods should be able to fully encrypt and store vital and sensitive user information such as passwords, and personal user information such as age.
Authentication and Authorization	Multi-Factor Authentication (MFA): Users should be required to use multi-factor authentication when accessing sensitive parts of the app such as changing account settings. Role-Based Access Control (RBAC): The app should have role-based access control to ensure that only authorized users (admins) can perform certain actions (user moderation).
Data Privacy	Data Minimization: Only necessary data should be collected, and it should not be retained longer than needed. Users should also have the option to delete their data completely.

5.3 Usability Requirements

Section	Performance Requirement
Responsiveness	OnlyFoods should be fully functional on mobile, tablet, and desktop devices, with layouts that adjust to screen size without sacrificing functionality. The interface should support touch interactions on mobile devices, and the website should be tested across major devices and browsers to ensure consistency.

Error Recovery	Clear, descriptive error messages should guide users to resolve issues independently. If a form submission fails, the system should retain the entered information so the user doesn't have to re-enter all details. A "Cancel" button should be available in all forms, allowing users to exit without saving changes.
Efficiency of Use	Frequent users should be able to accomplish core tasks, such as creating a post or tracking daily meals, within 2-3 minutes. Users should be able to add, edit, or delete ingredients in a post within 30 seconds.
Learnability	New users should be able to understand and navigate the basic functions of OnlyFoods (e.g., viewing posts, creating a post, liking, and saving posts) without requiring external guidance. An interactive tutorial or help section should be available on the first login, guiding users through primary features.

5.4 Reliability and Availability Requirements

Section	Performance Requirement
Uptime	The OnlyFoods platform must maintain a minimum uptime of 99.9%, allowing for no more than 8.77 hours of downtime per year. Scheduled maintenance should be minimized and performed during off-peak hours to reduce user impact.
Data Consistency	The system must ensure data consistency across different parts of the platform. For example, if a user likes or saves a post, the action should be reflected immediately across the user's profile, main feed, and notifications.
User Session	User sessions should be managed reliably, allowing users to remain

Management	logged in as expected, with appropriate session timeouts and reauthentication mechanisms. The system should automatically handle session expiration and display a clear message to the user if they need to log in again.
Recovery Time	In the event of a system failure or data corruption, the system should be fully recoverable within 2 hours, including restoring recent data backups to ensure minimal disruption for users.

5.5 Scalability Requirements

Section	Performance Requirement
Concurrent Users	The OnlyFoods platform must support up to 1,000 concurrent users without a significant increase in page load time (no more than a 10% increase in response times).
Database Scalability	The database should be able to handle large volumes of data, including user profiles, posts, images, and interactions (likes, comments, flags). OnlyFoods should be prepared to scale its database resources as the user base grows.
Data Caching	Frequently accessed data, such as user profiles, recent posts, and popular posts, should be cached to reduce load on the main database and improve response times for users.

6. Appendices

6.1. Appendix A: Data Dictionary

Term	Definition
Account	The combination of username, email, and password that allows a user to access and use the system's features.
Account Deletion	The permanent removal of a user's account and all associated data by an admin.
Admin	A user role with elevated privileges to manage the system, including reviewing, suspending, banning, or deleting user accounts and posts.
Authentication	The process of verifying the identity of a user, typically through credentials like a username and password.
Automatic Nutritional Calculation	A feature that calculates the nutritional content of a post based on the ingredients provided by the user.
Ban	Disciplinary actions taken by an admin to restrict or revoke a user's access to the system permanently.
Category	A classification for food posts, selected by the content creator when creating a post (e.g., "Breakfast," "Dinner").
Comment	A textual response from a user to a post. Comments are visible to other users interacting with the post.
Content Moderation	The process of reviewing and managing user-generated content to ensure it complies with guidelines.
Content Sharing	The ability for users to distribute posts or media to other platforms or social networks.

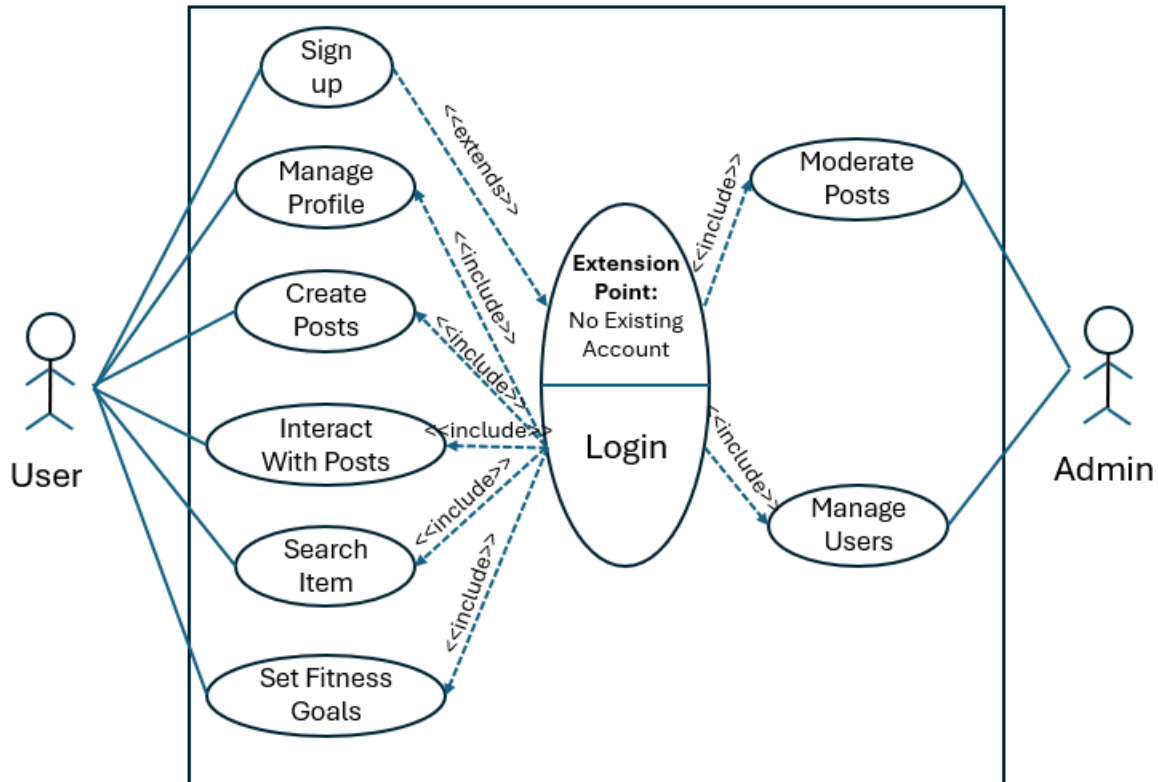
Daily Calorie Intake	The total number of calories consumed by a user in a day, which the system can help track.
Delete User Account	The removal of a user's account and all associated data from the system by an admin.
Diet Goals	Objectives related to nutrition and fitness, such as weight loss or muscle gain, which users can set and track.
Email Verification	A process that confirms the validity of a user's email address, often involving sending a confirmation link to the provided email.
Error Message	Clear and helpful messages provided to users when something goes wrong, guiding them on how to resolve the issue.
Flag	A process where users can report content that violates guidelines or is inappropriate, triggering a review by admins.
Follow	An action taken by a user to subscribe to updates from another user, allowing the follower to see the following user's posts in their feed.
Food Category	A classification of food items, such as desserts, main courses, or snacks, that can be selected when creating a post.
Height	A personal metric provided by the user that can be updated in their profile.
Ingredients	Food item that builds up to a dish in the post. They can be selected from the database or keyed in by a content creator when creating a post. Ingredients are used to calculate the nutritional content of a post.
Like	An action taken by a user to show appreciation for a post. The total number of likes is displayed on the post.
Login	Accessing of the user account on the website after authentication
Macronutrients	Essential nutrients required in large amounts, such as proteins, fats, and carbohydrates.

Micronutrients	Essential nutrients needed in smaller amounts, such as vitamins and minerals.
Nutritional Content	The calculated information about the calories, carbohydrates, fats, dietary fiber, and other nutrients in a post, based on the ingredients provided.
Passwords	Secret keys or codes used by users to authenticate themselves and gain access to the system.
Personal Information	Data related to an individual, such as name, email, height, weight, and dietary preferences.
Photos/Images	Visual content that users can upload to enhance their posts.
Post	Content created by a user (content creator) that includes text, images, videos, ingredients, and preparation steps. Posts can be liked, commented on, shared, saved and flagged.
Post Deletion	The removal of content by an admin, typically in response to guideline violations or user reports.
Post Submission	The process of creating and submitting content for publication on the system.
Post-Creation Template	A pre-designed format provided to users to help them structure their posts, including sections for text, images, and nutritional information.
Profile	The user's personal information stored in the system, including username, email, profile picture, height, weight, and target calorie intake.
Profile Management	The set of features that allows users to update their personal information, such as usernames, profile pictures, and contact details.
Profile Picture	An image uploaded by the user to visually represent their account. It can be updated by the user.
Report	A system-generated document sent to a user when their post is deleted, detailing the reasons for deletion.

Reporting System	A mechanism that allows users to flag inappropriate content for review by an admin.
Revoke Suspension/Ban	The action of lifting a suspension or ban, allowing the user to regain access to the system before the originally set duration ends.
Search	The action of finding specific posts, users, or ingredients within the system using keywords or filters.
Share	An action taken by a user to distribute a post to other social media platforms or users within the system.
Sign Up	The process of creating a new user account in the system.
Suspend	A temporary restriction placed on a user account by an admin, preventing the user from accessing their account for a specified duration.
Suspension Duration	The duration of time during which a user is suspended or banned from the system.
Target Calorie Intake	A user-defined goal for daily calorie consumption, based on their fitness objectives.
User Interface (UI)	The visual and interactive elements of the system that users interact with.
Username	A unique identifier chosen by the user during account creation. It is used for logging in and displaying user information across the platform.
Users	An individual who can interact with the system, including creating an account, managing their profile, and interacting with content. Users are categorized into general users and admins.
Weight	A personal metric provided by the user that can be updated in their profile.

6.2. Appendix B: Use Case Diagrams and Descriptions

6.2.1. Use Case Model



6.2.2. Use Case Descriptions

- Login

Use Case ID:	1		
Use Case Name:	Login		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	User, Admin		

Description:	Allows Users to log in to the application, providing access to personalized features and data. The system verifies the credentials, email, and account status (e.g., banned, suspended). Admins are redirected to the admin dashboard, while other users are directed to the main user page.
Preconditions:	User has an active, verified account with valid email and password. User is on the login page.
Postconditions:	User gains access to their account and is redirected based on role (Admin or standard User).
Priority:	High
Frequency of Use:	Frequently used by all Users and Admins.
Flow of Events:	<ol style="list-style-type: none"> 1. User enters email and password. 2. System verifies credentials and account status. 3. If verified and active, System logs the User in and redirects them to the Main page.
Alternative Flows:	<p>AF1: If credentials are incorrect, System prompts the User to re-enter details.</p> <p>AF2: If the account is suspended, System displays a message with the suspension reason and duration.</p> <p>AF3: If the account is banned, System notifies the User of the ban and provides details.</p> <p>AF4: If credentials belong to Admin, System redirects to Admin page.</p>
Exceptions:	If there is a server error during login, System displays an error message and allows the User to retry.
Includes:	This use case includes functions for credential verification, email validation, and role-based redirection.
Special Requirements:	System must encrypt User credentials and comply with security standards.
Assumptions:	User has stable internet connectivity and valid login credentials.

Notes and Issues:	Future enhancements may include two-factor authentication for added security.
-------------------	---

- Sign Up

Use Case ID:	2		
Use Case Name:	Sign Up		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	New User		
Description:	Allows new Users to create an account by providing necessary details, such as username, email, and password. Upon successful registration, Users are redirected to a profile setup page.		
Preconditions:	User is redirected from the login page to the sign-up page.		
Postconditions:	A new User account is created and saved in the database. User is redirected to the profile setup page.		
Priority:	High		
Frequency of Use:	Frequently used by new Users		
Flow of Events:	<ol style="list-style-type: none"> 1. User enters username, email, password, and confirms password. 2. System verifies that all required fields are complete and passwords match. 3. System creates a new account and redirects the User to the profile setup page. 		
Alternative Flows:	AF1: If email is invalid, System displays an error message. AF2: If passwords do not match, System prompts the User to re-enter them.		
Exceptions:	If there is a server error during registration, System displays an error message and allows the User to retry.		
Includes:	This use case includes validation functions for email format and password matching.		

Special Requirements:	System must support password encryption and store credentials securely.
Assumptions:	User has stable connectivity and unique username and email.
Notes and Issues:	Future updates may include CAPTCHA or reCAPTCHA for bot prevention.

- Manage Profile

Use Case ID:	3		
Use Case Name:	Manage Profile		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	User		
Description:	Allows users to update and manage profile details, including profile picture, birthday, height, and weight. Updates to height and weight recalculate the target calorie intake.		
Preconditions:	User is logged in. User is on the profile management page.		
Postconditions:	User profile information is updated in the database. Updated information, including recalculated target calories, is displayed on the profile.		
Priority:	Medium		
Frequency of Use:	Frequently used by Users		
Flow of Events:	<ol style="list-style-type: none"> 1. User navigates to profile management. 2. User updates profile details. 3. System validates and saves changes to the database. 4. System recalculates age and target calories, if applicable. 		
Alternative Flows:	AF1: If the system detects an invalid image, missing required fields or invalid inputs, it displays an error message and prompts the user to correct the input.		

Exceptions:	If there is a server error during profile update, System displays an error message and allows the User to retry.
Includes:	Functions for image upload, field validation, and recalculation of nutritional goals. User to be logged in under a valid account
Special Requirements:	Profile picture must be in .png, .jpg, .jpeg format.
Assumptions:	User has stable connectivity.
Notes and Issues:	Future enhancements may include batch updates or social media integrations.

- Create Posts

Use Case ID:	4		
Use Case Name:	Create Posts		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	User		
Description:	Allows users to search and filter posts based on ingredients, post title, or category.		
Preconditions:	The user is logged in. The user has launched the post creation template.		
Postconditions:	A new post is saved in the database, containing the user's input. The post is visible to other users on the platform.		
Priority:	High		
Frequency of Use:	Frequently used by active users.		
Flow of Events:	<ol style="list-style-type: none"> 1. User navigates to the "+ Add" button. 2. System displays a post creation template with fields for images, ingredients, and descriptions. 3. User uploads images and selects ingredients and their quantities. 		

	<ol style="list-style-type: none"> 4. User enters a description and any additional information. 5. System validates all fields (e.g., image format, text length). 6. System saves the post data to the database. 7. System displays a success message and redirects the user to their profile or the feed.
Alternative Flows:	<ol style="list-style-type: none"> 1. AF1: If the system detects an invalid image or missing required fields, it displays an error message and prompts the user to correct the input.
Exceptions:	If there is a server error during save, the system displays an error message and allows the user to retry.
Includes:	<p>This use case may call functions to handle image uploads and perform validation on fields.</p> <p>User to be logged in under a valid account</p>
Special Requirements:	Post creation fields must support text, image uploads, and a dynamic list of ingredients.
Assumptions:	The user has the necessary permissions and a stable connection for uploading images.
Notes and Issues:	Future versions may allow for bulk ingredient selection or templates for recurring posts.

- Interact with Posts

Use Case ID:	5		
Use Case Name:	Interact with Posts		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	User		
Description:	Enables users to engage with posts by liking, commenting, saving, and flagging them. Notifications are sent to post owners for likes, comments, and flags.		
Preconditions:	User is logged in.		

	User is viewing a post.
Postconditions:	User interactions are saved and visible to other users. Notifications are sent to post owners based on interactions.
Priority:	High
Frequency of Use:	Frequently used by active Users
Flow of Events:	<ol style="list-style-type: none"> 1. User likes, comments, saves, or flags a post. 2. User can delete only their own posts 3. System records the interaction and updates post metadata. 4. Notifications are sent to the post owner for each interaction.
Alternative Flows:	AF1: If user flags a post, the system displays a report form for further details. AF2: User can undo an interaction by clicking the icons again.
Exceptions:	If there is a server error while saving interactions, the system displays an error message and allows the user to retry.
Includes:	Functions for updating interactions and sending notifications. User to be logged in under a valid account
Special Requirements:	User has a stable connection and valid permissions for the action.
Assumptions:	User has stable connectivity.
Notes and Issues:	Future enhancements may include batch updates to increase rate which interactions are updated for Users

- Search Item

Use Case ID:	6		
Use Case Name:	Search item		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	User, Admin		
Description:	Allows users to search and filter posts based on post title, category or description.		

	Allows admins to search and filter flags based on violation type or username.
Preconditions:	User is logged in. User is has opened the search bar.
Postconditions:	Relevant posts are displayed in response to the search query.
Priority:	Medium
Frequency of Use:	Frequently used by users exploring specific ingredients or recipes. Frequently used by admins exploring flags.
Flow of Events:	1. User or Admin enters search terms or selects filters. 2. System retrieves and displays items with matching letters.
Alternative Flows:	AF1: If no matches are found, the system will not display any items.
Exceptions:	
Includes:	This use case may call functions to handle filtering by text. User to be logged in under a valid account
Special Requirements:	Search must be responsive within 2 seconds for efficient user experience.
Assumptions:	User has stable connectivity and valid search inputs.
Notes and Issues:	Future updates may include advanced filtering and sort options.

- Track Daily Nutrition Goals

Use Case ID:	7		
Use Case Name:	Track Daily Nutrition Goals		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	User		
Description:	Tracks users' daily nutritional intake based on profile data on daily meals uploaded by users, recalculating requirements when daily meals information changes.		
Preconditions:	User is logged in. User has logged meals or updated profile information.		

Postconditions:	User's nutrition goals are recalculated and displayed based on recent daily meals or profile updates.
Priority:	High
Frequency of Use:	Frequently used by users tracking dietary goals.
Flow of Events:	<ol style="list-style-type: none"> 1. User logs a meal or updates profile details. 2. System calculates updated daily intake and goals. 3. Updated information is displayed in the profile page.
Alternative Flows:	AF1: If User deletes daily meals logs, System will update profile details accordingly.
Exceptions:	If a server error occurs, the system displays an error message and allows the user to retry.
Includes:	<p>Functions for calorie tracking and goal calculation.</p> <p>User to be logged in under a valid account</p>
Special Requirements:	Nutritional data must update in real-time based on recent inputs.
Assumptions:	User has valid profile data and a stable connection.
Notes and Issues:	Future enhancements may include advanced nutritional analytics.

- Admin Moderate Users

Use Case ID:	8		
Use Case Name:	Admin Moderate Users		
Created By:	Chan Kee Qing	Last Updated By:	Isaac
Date Created:	3/9/2024	Date Last Updated:	10/11/2024
Actor:	Admin		
Description:	Enables admins to review flagged posts, and manage user accounts with actions like issuing warnings, suspensions, and bans.		
Preconditions:	<p>Admin is logged in.</p> <p>Admin is on the admin page.</p>		
Postconditions:	User accounts are moderated as per admin actions.		
Priority:	High		

Frequency of Use:	Moderately used by admins for community management.
Flow of Events:	<ol style="list-style-type: none"> 1. Admin accesses the dashboard and reviews flagged posts. 2. Admin selects actions (e.g., warn, suspend, ban) based on violation severity. 3. System updates the account status accordingly.
Alternative Flows:	AF1: If flagged content is deemed appropriate, admin dismisses the flag.
Exceptions:	If a server error prevents action, the system notifies the admin and retries.
Includes:	<p>Functions for flag review, user suspension, and ban management.</p> <p>User to be logged in under a valid account</p>
Special Requirements:	Administrative actions must update in real-time for user logins and notifications based on admin reviews.
Assumptions:	Admin has the necessary permissions and stable connectivity.
Notes and Issues:	Future versions may implement automation for flag review and reporting analytics.