Software Requirements Specification

for

Food Waste Reducer

Version <1.1>

Prepared by

Group Name: EcoMunch

Prachi Sarda	SE22UARI130	se22uari130@mahindrauniversity.edu.in
G Aniketh	SE22UARI211	se22uari211@mahindrauniversity.edu.in
P Mohith Krishna	SE22UARI138	se22uari138@mahindrauniversity.edu.in
Sneha Sharma	SE22UARI162	se22uari162@mahindrauniversity.edu.in
Shreya Patil	SE22UARI207	se22uari207@mahindrauniversity.edu.in
Shilpa	SE22UARI152	se22uari152@mahindrauniversity.edu.in

Instructor:	Arun Avinash Chauhan
Course:	Software Engineering
Lab Section:	*IT lab*
Teaching Assistant:	*Sri Venkata Surya Phani Teja*
Date:	10 /03/ 20255

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
SRS 2	Prachi Sarda, G Aniketh, P Mohith Krishna, Sneha Sharma, Shreya Patil, Shilpa	The SRS2 documents includes updates to the use case model and modifications to the appendices. The remodeled use case diagram enhances clarity, while the appendix revisions improve documentation accuracy.	**10 March, 2025**
SRS 1	Prachi Sarda, G Aniketh, P Mohith Krishna, Sneha	The SRS1 document outlines the functional and non-functional requirements of a software system. It typically includes an	**04 March, 2025**

Version	Primary Author(s)	Description of Version	Date Completed
SRS 2	Prachi Sarda, G Aniketh, P Mohith Krishna, Sneha Sharma, Shreya Patil, Shilpa	The SRS2 documents includes updates to the use case model and modifications to the appendices. The remodeled use case diagram enhances clarity, while the appendix revisions improve documentation accuracy.	**10 March, 2025**
	Sharma, Shreya Patil, Shilpa	introduction with project objectives, scope and definitions along with system features, user requirements and interface specifications. Functional requirements define system behavior, while non-functional aspects like performance, security, and usability constraints are also specified.	

1 Introduction

*The **Food Waste Reducer** is a software solution designed to help individuals and households minimize food waste through smart tracking, meal planning, and local food-sharing initiatives. By allowing users to log food items, receive expiry alerts, and access proper storage recommendations, the system aims to promote sustainable consumption habits. Additionally, it connects users with local food-sharing and composting programs, ensuring that surplus food is repurposed rather than wasted. Through an intuitive interface, the Food Waste Reducer provides actionable insights and fosters a more responsible approach to food consumption.*

1.1 Document Purpose

This Software Requirements Specification (SRS) document defines the functional and non-functional requirements for the "Food Waste Reducer" software application. The system aims to help individuals and households track and plan meals, provide guidance on proper food storage techniques, and connect users with food-sharing or composting programs to minimize food wastage.

1.2 Product Scope

The "Food Waste Reducer" application is designed to:

- Enable users to track and plan meals efficiently.
- Provide best practices for food storage to reduce spoilage.
- Connect users with local food-sharing or composting programs.
- Offer a waste analytics dashboard to track and optimize food consumption.

This product will be a user-friendly application accessible via web and mobile platforms, supporting sustainable consumption habits to contribute to global food waste reduction efforts

1.3 Intended Audience and Document Overview

This document is intended for developers, project managers, clients (Mahindra University Software Engineering Course), testers, and users who will interact with the system. It covers the overall product description, specific functional and non-functional requirements, and system constraints.

1.4 Definitions, Acronyms and Abbreviations

- **SOW**: Statement of Work

- **SRS**: Software Requirements Specification
- **UI**: User Interface
- **UX**: User Experience
- **API**: Application Programming Interface
- **DBMS**: Database Management System

1.5 Document Conventions

Text Formatting:

- *Section headings are numbered (e.g., 1.1, 2.3).*
- *Important terms and definitions are italicized or bolded.*
- *Bullet points are used for lists.*

Terminology:

- *Acronyms such as SRS (Software Requirements Specification), UI (User Interface), API (Application Programming Interface), etc., are defined in the Definitions section.*
- *Technical terms are used consistently throughout.*

Numbering and Referencing:

- *Sections follow a structured numbering format.*
- *Requirements are labelled systematically (e.g., F1 for Functional Requirements).*
- *Figures and tables are labelled appropriately*

2 Overall Description

2.1 Product Overview

The Food Waste Reducer is a new, self-contained software solution designed to address the global issue of food waste by integrating food tracking, meal planning, and food-sharing functionalities. Unlike traditional meal planners or inventory management systems, this application focuses specifically on reducing food wastage by providing expiration alerts, personalized storage recommendations, and seamless integration with local food-sharing and composting initiatives.

The system operates independently but can integrate with third-party food donation platforms and community-based food-sharing networks. It is designed for mobile and web platforms, ensuring accessibility for users across different devices. The solution consists of several interconnected modules, including a User Dashboard, Food Inventory System, Notification Engine, Analytics Dashboard, and Food-Sharing Interface. These components interact with external systems such as food banks, community kitchens, and composting centers to maximize food utilization.

2.2 Product Functionality

The system will include:

1. **Meal Tracking & Planning**: Users can log their meals, input expiration dates, and receive timely reminders.
2. **Storage Tips & Expiry Alerts**: System will provide personalized storage recommendations and sends notifications before food items expire.

3. **Local Food-Sharing & Composting**: Connects users to local surplus food-sharing networks and composting initiatives.
4. **Waste Analytics Dashboard**: Offers detailed insights into food consumption patterns, wastage levels, and cost savings.
5. **Multi-Platform Accessibility**: Available on web, iOS, and Android with a consistent user experience.

2.3 Design and Implementation Constraints

- The system must be compatible with web, iOS, and Android platforms.
- It must ensure secure storage and protection of user data using encryption methods.
- The system should comply with food safety and data privacy regulations.
- The software must integrate with third-party APIs for local food-sharing initiatives.
- The database should support high availability and scalability.

2.4 Assumptions and Dependencies

- Users will have access to smartphones or computers.
- Local food-sharing initiatives and composting programs will be available for integration.
- Data security measures will be implemented to protect user information.
- The software will require a stable internet connection for real-time updates and synchronization.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Interface Features:

1. ***Home Dashboard*** *– Displays food inventory, expiration alerts, and quick meal planning options.*
2. ***Food Inventory Management –*** *Users can add, edit, and remove food items by scanning barcodes or manually entering details.*
3. ***Notification Centre*** *– Provides alerts for food expiry and reminders for meal planning.*
4. ***Food Sharing Section –*** *Allows users to find and connect with local food-sharing and composting programs.*
5. ***Analytics Dashboard*** *– Displays insights on food consumption and waste reduction trends.*

3.1.2 Hardware Interfaces

1. ***Touchscreen & Click-based Navigation*** *– Users can tap or click on icons and menus to access different sections.*
2. ***Search & Filter Options*** *– Users can quickly locate food items or filter based on expiry date and category.*
3. ***Barcode Scanning (Mobile) –*** *Allows quick food item entry by scanning barcodes.*
4. ***Interactive Graphs & Charts –*** *Used in the analytics dashboard to provide visual insights.*

3.1.3 Software Interfaces

1. ***Mobile App ↔ Backend Server***
   - *The mobile app communicates with the backend using secure API endpoints.*
   - *Users can add food items, edit inventory, and receive expiration alerts in real time.*
2. ***Mobile App ↔ Notification System***
   - *The backend pushes real-time expiry alerts and meal reminders to the mobile app using Firebase Cloud Messaging (FCM) for Android/iOS.*
3. ***Mobile App ↔ Food-Sharing Network***
   - *The app connects users to local food banks and composting programs via integrated APIs.*
   - *Users can find nearby donation centres and schedule food pickups.*
4. ***Mobile App ↔ Analytics Dashboard***
   - *The app retrieves waste analytics data from the backend.*
   - *Users can view consumption trends, track food wastage, and optimize their meal planning.*

3.2 Functional Requirements

3.2.1 F1: Food Inventory Management

- The system shall allow users to **add, edit, and delete** food items in their inventory.
- The system shall support **barcode scanning** to quickly add food items.
- The system shall enable users to **categorize food items** (e.g., dairy, vegetables, meat).
- The system shall track **expiration dates** for all stored food items.

3.2.2 F2: Expiry Alerts & Notifications

- The system shall send **push notifications** when food items are approaching their expiration date.
- The system shall provide **customizable alert settings** (e.g., remind 3 days before expiry).
- The system shall notify users of expired food items and suggest disposal or donation options.

3.2.3 F3: Meal Planning Assistance

- The system shall recommend **meal plans** based on the user's food inventory.
- The system shall allow users to **create and save custom meal plans**.
- The system shall provide **nutritional information** for suggested meals.

3.2.4 F4: Food-Sharing & Donation Integration

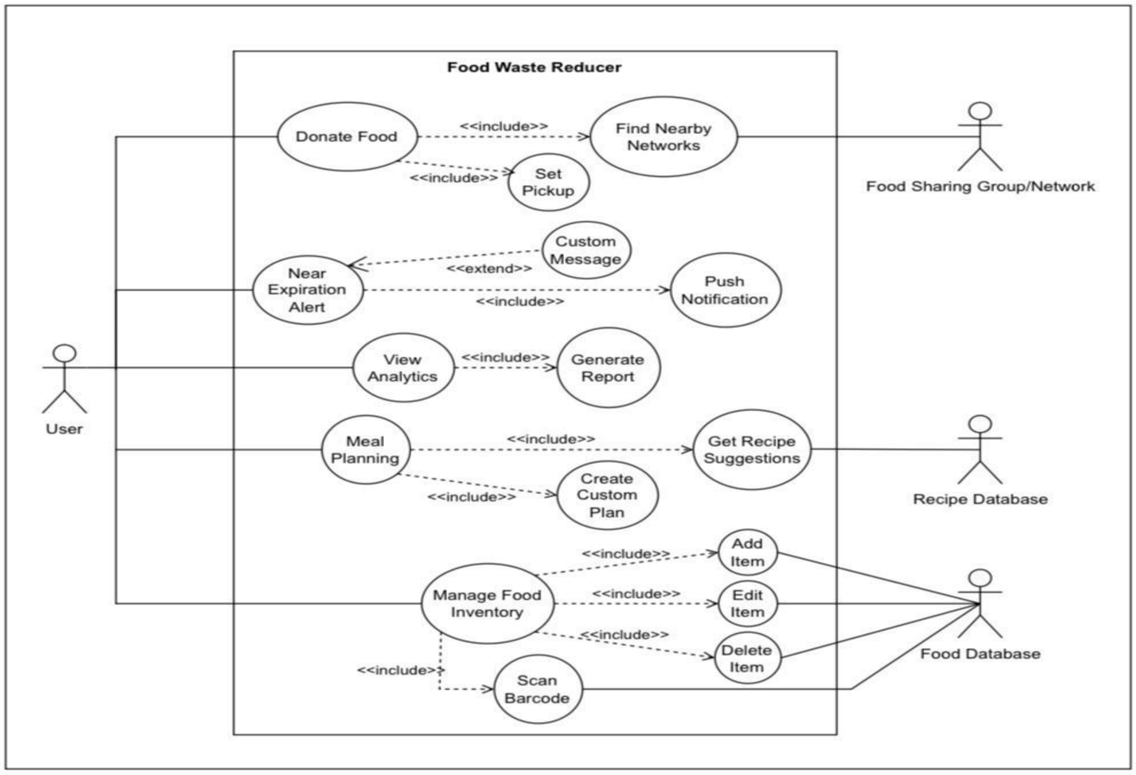
- The system shall allow users to **list surplus food** for donation.
- The system shall connect users with **local food banks and composting programs**.
- The system shall provide **location-based search** for nearby food-sharing initiatives.

3.2.5 F5: Analytics & Insights

- The system shall generate **waste analytics reports**, showing food consumption trends.

- The system shall provide **recommendations on reducing food waste** based on user data.
- The system shall display **interactive charts** for tracking waste reduction progress.

3.3 Use Case Model



3.3.1 Use Case #1 (U1): Manage Food Inventory

Author

Purpose - Allow users to efficiently track and manage their food items, including adding new items, updating existing ones, and removing consumed or discarded items to maintain an accurate inventory

Requirements Traceability – F1 (Food Inventory Management)

Priority - High - Core functionality of the system

Preconditions –

User has created an account and is logged into the system

User has access to the food inventory section of the application

Post conditions –

Food inventory is updated with accurate information

System tracks expiration dates for all food items

Actors –

User (primary)

Food Database (for barcode scanning functionality)

Flow of Events

1. Basic Flow

   User navigates to the food inventory section

   User selects option to add, edit, or remove food items

   - For adding items:
     - User can manually enter food item details or scan barcode
     - User inputs expiration date, quantity, and category
     - System stores the information in the inventory

- For editing items:
    - User selects existing item from inventory
    - User modifies details as needed
    - System updates the item information
- For removing items:
    - User selects item(s) to be removed
    - System prompts for confirmation
    - System removes the item(s) from inventory

2. Alternative Flow

    If barcode scanning is selected:

    - System activates device camera
    - User scans product barcode
    - System queries food database for product information
    - System pre-fills item details for user to confirm

3. Exceptions

    If barcode is not recognized:

    System notifies user

    Prompts for manual entry

    If required fields are missing:

    System displays validation error

    Highlights missing information

Includes U1.1 (Scan Barcode)

Notes/Issues

The system should support bulk operations for adding or removing multiple items

Mobile and web interfaces should maintain feature parity for inventory management

3.3.2 Use Case #2 (U2): Receive Expiry Alerts

Author: Based on SRS document
Purpose: Notify users about food items approaching expiration to reduce wastage through timely consumption or redistribution.
Requirements Traceability: F2 (Expiry Alerts & Notifications)
Priority: High - Critical for waste reduction goal
Preconditions:

- User has food items with expiration dates in the inventory
- User has enabled notifications for the application

Post conditions:

- User receives timely alerts about expiring food
- User can take action based on notifications

Actors:

- User (primary)
- Notification System

Flow of Events:

1. **Basic Flow:**
   - System regularly checks expiration dates of inventory items
   - When an item approaches expiration (based on user preferences):
     - System generates an alert
     - Notification is sent to user's device
     - User receives notification with item details and suggested actions
     - User can mark the notification as read or take immediate action

2. **Alternative Flow:**
   - User accesses notification center within app:
   - System displays all pending expiry alerts
   - User can view details and take action from notification center

3. **Exceptions:**
   - If notification delivery fails:
     - System logs the failure
     - Attempts to deliver again at next scheduled check
   - If user has disabled notifications:
     - Alerts are still visible in the tab’s notification centre

Notes/Issues:

Notification preferences should be customizable (timing, frequency)
Different alert levels may be implemented (urgent for items expiring soon)
Consider bundling notifications to prevent alert fatigue

3.3.3 Use Case #3 (U3): Plan Meals

Author: Based on SRS document

Purpose: Help users create efficient meal plans that prioritize soon-to-expire ingredients to reduce food waste while maintaining balanced nutrition.

Requirements Traceability: F3 (Meal Planning Assistance)

Priority: Medium - Important supporting functionality

Preconditions:

- User has food items in inventory
- User has accessed the meal planning section

Post conditions:

- User has created a meal plan
- System updates expected usage of inventory items

Actors:

- User (primary)

Flow of Events:

1. Basic Flow:
    - User navigates to meal planning section
    - System analyses inventory and suggests recipes prioritizing soon-to-expire items
    - User selects meals to include in plan
    - User assigns meals to specific days/times
    - System saves the meal plan
    - System updates expected consumption dates for inventory items
2. Alternative Flow:

    - User creates custom meal:
        - User selects ingredients from inventory
        - User provides meal details (name, portions, etc.)
        - System adds custom meal to plan
3. Exceptions:
    - If no suitable recipes are found:
        - System suggests alternatives or basic usage ideas for expiring items

Notes/Issues:

- Integration with recipe databases could enhance functionality
- Consider dietary restrictions and preferences in meal suggestions
- Provide nutritional information for suggested meals

3.3.4 Use Case #4 (U4): Share/Donate Food

Author: Based on SRS document

Purpose: Enable users to connect with local food-sharing initiatives to donate surplus food instead of discarding it.

Requirements Traceability: F4 (Food-Sharing & Donation Integration)

Priority: Medium - Important for community impact

Preconditions:

- User has surplus food items to share
- User has accessed the food-sharing section

Post conditions:

- User's donation is listed or scheduled
- Receiving organization is notified of available donation

Actors:

- User (primary)
- Food-Sharing Network

Flow of Events:

1. Basic Flow:
    - User navigates to food-sharing section
    - User selects items from inventory to donate
    - System displays nearby donation options (food banks, community fridges, etc.)
    - User selects preferred donation method
    - User schedules pickup/dropoff if applicable
    - System confirms donation listing
2. Alternative Flow:
    - For direct community sharing:
        - System lists item in community marketplace
        - Other users can request the item
        - Original user confirms the recipient
3. Exceptions:
    - If no local options are available:
        - System suggests composting or alternative disposal methods
    - If item is expired:
        - System flags item as unsuitable for human consumption
        - Suggests composting options instead

Includes: Location-based search for nearby donation centers

Notes/Issues:

- Consider verification methods for food safety
- Include clear guidelines on acceptable donation items

- Implement rating/feedback system for donation process

3.3.5 Use Case #5 (U5): View Analytics

Author: Based on SRS document

Purpose: Provide users with insights about their food consumption patterns and waste reduction progress to encourage sustainable habits.

Requirements Traceability: F5 (Analytics & Insights)

Priority: Low - Enhances user experience but not critical for core functionality

Preconditions:

- User has been using the system for some time (data available)
- User has accessed the analytics dashboard

Post conditions:

- User views personalized analytics
- System provides actionable recommendations

Actors:

- User (primary)

Flow of Events:

1. Basic Flow:
    - User navigates to analytics dashboard
    - System processes historical data and generates visualizations
    - Dashboard displays:
        - Waste reduction trends over time
        - Most frequently wasted food categories
        - Cost savings from reduced waste
        - Environmental impact metrics

   - System provides personalized recommendations

2. Alternative Flow:
   - User filters analytics by time period:
     - System recalculates metrics for selected timeframe
     - Updates visualizations accordingly
3. Exceptions:
   - If insufficient data is available:
     - System displays placeholder information
     - Explains that more data is needed for accurate analytics

Notes/Issues:

- Consider gamification elements to encourage engagement
- Provide comparison with community averages if possible
- Include exportable reports for user reference

4 Other Non-functional Requirements

4.1 Performance Requirements

- The system should support up to 10,000 concurrent users.
- Data retrieval for meal tracking should be completed in under 2 seconds.
- Notifications should be delivered within 5 seconds of triggering an alert.

4.2 Safety and Security Requirements

- User data must be encrypted and stored securely using AES-256 encryption.
- The application should comply with GDPR and other data privacy laws.
- User authentication should be secured with multi-factor authentication (MFA).
- API access should be protected using OAuth 2.0 authentication.

4.3 Software Quality Attributes

- **Reliability**:
  - The system should ensure minimal downtime and be available 99.9% of the time.
- **Usability**:
  - Intuitive design for users of all age groups, including accessibility features for visually impaired users.
- **Scalability**:
  - Should support increased user load over time and ensure smooth performance even under high demand.
- **Maintainability**:
  - Code should be modular and well-documented for ease of future development.
- **Interoperability**:
  - Should be compatible with various third-party APIs and external food databases.