EXCESS FOOD MANAGEMENT SYSTEM

Software Requirements Specification

S.SAHANA
2022503049
ssahana2114@gmail.com

S.V.LOGARATHAN
2022503047
techdare7@gmail.com

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the Excess Food Management System. This system will connect food donors (such as hotels and events) with recipients (such as homeless individuals and charitable organizations) to reduce food waste and address food insecurity.

1.2 Scope

The system will facilitate the logging of excess food, matching donors to recipients, coordinating delivery, and tracking food redistribution. It will be scalable to accommodate a growing number of users and ensure safe, timely, and reliable food redistribution.

1.3 Definitions, Acronyms, and Abbreviations

- *Donor*: A hotel, restaurant, or event organizer with excess food.
- *Recipient*: Homeless individuals or organizations (e.g., shelters) that receive the food.
- *Volunteer/Delivery Service*: Individuals or services responsible for transporting food from donors to recipients.
- *System Administrator*: The user responsible for managing the system.

1.4 References

On our own thought process.

1.5 Overview

This SRS document outlines the functional and non-functional requirements for the system, along with constraints, assumptions, and dependencies.

2. General Description

//

2.1 Product Perspective

The system is a web-based application designed to connect food donors with recipients. It will use a centralized database for food logging, matching, and tracking deliveries. The system will be accessible via web browsers, with a user-friendly interface for donors, recipients, and volunteers.

2.2 Product Features

- Log excess food with details like type, quantity, and pickup time.
- Automatically match donors to recipients based on location, food type, and recipient needs.
- Real-time notifications for donors, recipients, and volunteers.
- Pickup and delivery scheduling and tracking.

- Account management for all users.
- Reports on food redistribution, impact, and feedback.

2.3 User Characteristics

- *Donors*: Non-technical users such as hotel/event staff who will log excess food.
- *Recipients*: Non-technical users from organizations or homeless individuals who will receive food.
- *System Administrator*: Technical user responsible for managing system operations and troubleshooting.

2.4 General Constraints

- The system must comply with local food safety and handling regulations.
- The system should be operational 24/7 to handle real-time food logging and matching.
- System performance should not degrade as the number of users increases.

2.5 Assumptions and Dependencies

2.5.1 Availability of Excess Food

Assumption: There will be consistent availability of excess food with the donors. This project depends on the regular supply of surplus food in order to work properly.
Justification: This project depends on the regular availability of this excess surplus food in order to capabilities.

2.5.2 Stakeholder Cooperation

Assumption: All stakeholders constituently food donors, food recipients will collaborate and take part as envisaged.
Justification: Cooperation is paramount in gathering, classifying and disseminating food.

2.5.3 Food Donation Partnerships

Dependency: Collaborations with donors (i.e. restaurants, grocery retailers) are essential with regards to obtaining extra food supply.
Impact: In the absence of these partnership, there will be inadequate food to manage and distribute.

2.5.4 Technology Infrastructure

Dependency: There must be functional software for tracking donated items, managing inventory and logistics.
Impact: Technology failure will result in operational disruptions and reduced efficiency.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- The system will have a web-based interface with responsive design to ensure compatibility with various screen sizes (desktop, tablet, mobile).
- Forms for logging food donations and requesting food should be simple and easy to fill out.

3.1.2 Hardware Interfaces

- The system will interact with delivery services via mobile devices for real-time tracking and scheduling.

3.1.3 Software Interfaces

- The system should be integrated with GPS services to track food pickups and deliveries.
- The system may integrate with external APIs for push notifications and messaging services.

3.1.4 Communications Interfaces

- The system will utilize our java application for managing donations and coordinating logistics, complemented by email for real-time updates and coordination with donors, recipients, and volunteers.

3.2 Functional Requirements

3.2.1 User Registration and Authentication:

- The system must allow donors, recipients, and volunteers to create and manage their accounts.
- Users must be authenticated using a username and password before accessing system features.

3.2.2 Log Excess Food:

- Donors should be able to log excess food, specifying type, quantity, pickup time, and any special handling instructions.
- The system must store these details in the database for matching with recipients.

3.2.3 Match Donors to Recipients:

- The system must automatically match food donations with recipients based on proximity, food type, and recipient needs.
- The system should allow donors and recipients to confirm or reject matches.

3.2.4 Pickup and Delivery Coordination:

- The system should allow volunteers to coordinate and schedule pickups from donors and deliveries to recipients.
- Real-time tracking of food delivery should be available to all users.

3.2.5 Notifications:

- The system must send real-time notifications to donors, recipients, and volunteers regarding the status of food donations and deliveries.
- Notifications must include food expiration reminders and pickup confirmations.

3.2.6 Feedback and Reporting:

- Recipients should be able to provide feedback on the quality and quantity of food received.
- The system should generate reports for donors and recipients, summarizing food redistribution and its impact.

3.3 Use Cases

3.3.1 Use Case #1

Log Excess Food: Donors log details of available surplus food, including type, quantity, and pickup time.

3.3.2 Use Case #2

Match Donors to Recipients: The system automatically matches available food with suitable recipients based on proximity, needs, and food type.

3.3.3 Use Case #3

Track and Monitor Deliveries: Real-time tracking of food delivery from donor to recipient to ensure timely and safe distribution.

3.3.4 Use Case #4

Manage User Accounts: Allows donors, recipients, and volunteers to create and manage their accounts.

3.4 Non-Functional Requirements

3.4.1 Performance

There should be no time lag from the system in recording donation and matching it to the persons in need. The system should support the connection of 10,000 users at the same time without any performance degradation.

3.4.2 Usability

The system should be simple with an easy-to-use interface for all the different categories of users comprising of donors, recipients and volunteers. Users must be assisted with proper instructions and simple error messages to avoid frustration and hopelessness..

3.4.3 Availability

The system must be running all the time as food donation and its matching is expected to be done in real time. The system should provide its users with 99.9% computer availability with scheduled downtime communicated to the users quite in advance.

3.4.4 Security

The system must ensure secure user authentication and data encryption for sensitive information. Only authorized users (donors, recipients, and volunteers) should have access to specific functions in the system.

3.4.5 Maintainability

The system should be constructed using a modular architecture that encompasses a number of different components so as to facilitate upgrades as well as repairs. The amount of documentation done for the codebase should have been ample to accommodate future code maintenance and enhancements.

3.4 Logical Database Requirements

-The system uses MySQL database for storing the hotel, individual users credentials.
-The system stores the offers placed and transcation histories in a secure and encrypted environment.

4. Analysis Models

4.1 Use Case Diagram

Illustrates interactions between users (donors, recipients, volunteers) and the system, focusing on food logging, matching, and delivery coordination.

4.2 Class Diagram

Illustrates the core system entities, such as Donor, Recipient, FoodItem, Delivery, and their relationships.