Software Requirements Specification

for

Food Management System

Prepared by

Group-7

Arun B chandran-14
Jyothis S -37
Deepak P-76
Asa Shijil-75

TKM College of Engineering

Table of Contents

1. Introduction

The food management system is an online platform designed to reduce food wastage and promote sustainability by connecting people who have excess food with those who need it. The system focuses on the hospitality industry, specifically hostels, catering areas, and hotels, and provides users with affordable food options.

1.1 Purpose

Reduce food wastage in the hospitality industry by providing a platform for hostels, catering areas, and hotels to list excess food and for users to purchase available food.

1.2 Intended Audience

1. Hostels, catering areas, and hotels: The system is designed to help these organizations reduce food wastage and promote sustainability by connecting them with people who need excess food.
2. Users: The system provides users with affordable food options by allowing them to purchase available food from hostels, catering areas, and hotels.

1.3 Product Scope

The product scope of the food management system is to create an efficient and user-friendly online platform that reduces food wastage and promotes sustainability by connecting people who have excess food with those who need it.

1.4 References

https://spdload.com/blog/how-to-develop-a-web-application/

2. Overall Description

2.1 Product Perspective

The product perspective of the food management system is to provide an efficient and user-friendly online platform that connects people who have excess food with those who need it. The system will enable hostels, catering areas, and hotels to reduce food wastage and promote sustainability, while also providing affordable food options for users.

2.2 Product Features

1. User account management
2. Search functionality
3. Food listing
4. Order management
5. Notification through email
6. Delivery management
7. Payment gateway integration
8. Availability notifications
9. Feedback system

2.3 User Classes

1. *Hostel/catering/hotel staff:* This user class will consist of staff members who will manage the food listings and delivery services. They will have the technical knowledge required to manage the system and will need to be able to input, modify, and delete food listings, manage orders, and track delivery services.

2. *Food buyers:* This user class will consist of individuals who will use the system to search for and purchase available food. They will be non-technical users who will need a simple and user-friendly interface to search for available food, purchase food, and track their orders.

3. *Delivery personnel:* This user class will consist of delivery personnel who will deliver the purchased food to the buyers. They will have technical knowledge and will need to be able to access the delivery information through the system and update the delivery status.

2.4 Operating Environment

It will work on any web browser. It must require Wi-Fi or mobile data to work.

2.5 Design and Implementation Constraints

- Hardware constraints
  - It works only on Wi-Fi or Mobile data
- Tool specifications
  - IDE: VS Code
-
  - Framework: Django
- Language requirements
  - Front-end: HTML, CSS
  - Back-end: Python

2.6 User Documentation

The document will be distributed to customers and developers as soft-copy in pdf format as as well as a manual. The end users (target) would be provided a tutorial video available in the Website while registering.

3. System Features

1. User Registration:
   Users can create an account on the system by providing their personal details such as name, address, and contact information. Users will also be required to set a username and password to access the system.

2. Listing of Food:
   Hostels, catering areas, or hotels can list the extra food available on the system. The food details will include the type of food, quantity, and price.

3. Food Buying:
   Users can browse the available food listings and select the food they want to buy. Users will be required to provide their contact details and payment information to complete the transaction.

4. Food Delivery:
   The system will facilitate food delivery by assigning delivery boys to pick up the food from the hostel, catering area, or hotel and deliver it to the user's location. The user will be able to track the delivery boy's location in real-time.

5. Order Tracking:
   Users can track their food orders in real-time. They will receive notifications on the order status, and delivery updates such as the delivery boy's location.

6. Payment Integration:
   The system will integrate with payment gateway services such as PayPal or Stripe to facilitate secure and easy payment transactions. Users will be able to pay for the food using their preferred payment method.

7. User Reviews and Ratings:
   Users will be able to provide reviews and ratings for the food and the delivery service. This feedback will help other users to make informed decisions about which food to buy and which delivery boys to choose.

8. User Dashboard:
   Users will have access to a dashboard that displays their order history, payment history, and delivery status. The dashboard will also include features such as order tracking and cancellation options.

9. Admin Dashboard:
   Admins will have access to a dashboard that displays food listings, user profiles, and transaction history. The admin dashboard will include features such as user management, food management, and delivery boy management.

10. Integration with Payment Gateway:
The system will integrate with a payment gateway to facilitate secure and easy payment transactions.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

The performance of the Website is quite fast. A simple design is chosen as the UI for easy understanding and less storage consumption

4.2 Security Requirements

the security requirements of the food management system are critical to ensure that users' personal and financial information is protected from unauthorized access and to maintain the trust of the users.

4.3 Software Quality Attributes

- Usability: The system should be easy to use and navigate, with a user-friendly interface and intuitive workflows. It should be designed with the end-user in mind and provide clear instructions and guidance.
- Maintainability: The system should be easy to maintain, with a modular and scalable architecture that allows for easy upgrades and updates. It should also be well-documented to facilitate future maintenance and development.
- Portability: The system should be designed to run on different platforms and environments, with minimal modifications required. It should also be compatible with different browsers and operating systems.
- Performance: The system should be designed to be efficient and fast, with a quick response time and minimal downtime. It should also be able to handle a large number of users and transactions without slowing down.
- Reliability: The system should be reliable, with minimal downtime and high availability. It should also be designed to handle errors and exceptions gracefully, with minimal disruption to the user experience.
- Security: The system should be secure, with a robust security architecture that protects user data and prevents unauthorized access. It should also comply with industry standards and regulations.