

**Experiment -3**

**Waste Food Management & Donation App**

**Software Engineering**

**By**

**Vulasala Sujan (BU22CSEN0101959)**

**Meti Chaitanya (BU22CSEN0101523)**

**Maraka Ganesh (BU22CSEN0101803)**

**J Bhargav Reddy (BU22CSEN0101198)**

**Under the Guidance of**

**Kerenalli Sudarshana (700542)**

**Gandhi Institute of Technology and Management**

**(DEEMED TO BE UNIVERSITY)**

**BENGALURU, KARNATAKA, INDIA**

**Academic Year 2024-25**

**INDEX**

* **Introduction**
* **Overall Description**
* **Functional Requirements**
* **Non – Functional Requirements**
* **Design Constraints**
* **Appendices**

**1. Introduction:**

1.1 Purpose  
 The purpose of this document is to define the functional, non-functional, and system requirements for the Waste Food Management and Donation App. The app will provide a platform for food donors and recipients to connect, ensuring efficient management of surplus food and its distribution to those in need.

1.2 Scope  
 The Waste Food Management and Donation App is a mobile and web application designed to reduce food waste and combat hunger. The system facilitates food donations, ensures proper management of surplus food, and connects donors to recipients such as individuals, shelters, and food banks. Key features include donor registration, real-time food listing, geolocation services, and food safety guidelines.

1.3 Definitions, Acronyms, and Abbreviations

* Donor: An individual or organization providing surplus food.
* Recipient: An individual or entity receiving donated food.
* Admin: An administrator managing users and ensuring compliance with guidelines.
* SRS: Software Requirements Specification.

1.4 References

* IEEE SRS Standard 830-1998.
* Food Donation Management Guidelines Document, Version 1.0.

1.5 Overview  
This document outlines the system requirements, behavior, constraints, and user expectations for the development of the Waste Food Management and Donation App.

**2. Overall Description:**

2.1 Product Perspective  
The app replaces manual food donation processes, creating a digital platform for efficient surplus food management, transparency, and scalability.

2.2 Product Features

* User Management: Registration, login, and profile management for donors and recipients.
* Food Listing: Donors can list surplus food with details like quantity and expiration.
* Geolocation: Match donors with nearby recipients.
* Food Pickup Scheduling: Coordinate pickups via the app.
* Admin Dashboard: Manage users, food listings, and ensure compliance with safety standards.

2.3 User Classes and Characteristics

* Admin: Oversees app operations, user management, and guideline compliance.
* Donor: Lists and donates surplus food items.
* Recipient: Searches for and claims donated food.

2.4 Operating Environment

* Platforms: Mobile (iOS, Android) and Web browsers (Chrome, Firefox, Safari).
* Server: Cloud-based server with MySQL and Node.js.

2.5 Design and Implementation Constraints

* Secure Transactions: Encrypted communication using SSL/TLS.
* Technology Stack: React Native for mobile and Django for backend.

2.6 Assumptions and Dependencies

* Users have access to stable internet connections.
* Integration with third-party mapping and notification services.

**3. Functional Requirements:**

3.1 User Management

* Users must register with email or social login.
* Profile management for updating contact information.

3.2 Food Listing and Search

* Donors can list surplus food with details like type, quantity, and expiration date.
* Recipients can search for available food by location or type.

3.3 Geolocation and Notifications

* Match donors and recipients based on proximity.
* Notify users about new food listings or status updates.

3.4 Food Pickup Scheduling

* Enable donors and recipients to coordinate pickup timings.

3.5 Admin Features

* Manage user registrations and food listings.
* Monitor food safety compliance and resolve disputes.

**4. Non-Functional Requirements:**

4.1 Performance Requirements

* Support up to 500 concurrent users.
* Load food search results within 3 seconds.

4.2 Security Requirements

* Encrypt sensitive data like user credentials using SHA-256.
* Comply with local food safety and donation regulations.

4.3 Usability Requirements

* Accessible on both mobile and desktop platforms.
* Intuitive interface for all user classes.

4.4 Reliability

* System uptime of 99.5% annually.

**5. Design Constraints:**

* Adherence to Material Design Guidelines (Android) and Human Interface Guidelines (iOS).
* Must integrate with GPS and real-time messaging services.

**6. Appendices:**

6.1 Appendix A: References

* React Native Documentation.
* Django Framework Documentation.

6.2 Appendix B: Glossary

* SSL/TLS: Secure Sockets Layer / Transport Layer Security.
* Geolocation: The process of determining a user’s physical location.

6.3 Appendix C: Future Enhancements

* Integrating blockchain for enhanced transparency.
* AI-driven analytics for predicting food demand and supply.