1. INTRODUCTION

1.1 Purpose

This SRS document describes the functional and non-functional requirements for the Community Food Sharing Platform - a web-based application that connects food donors (restaurants, households) with recipients (individuals, families) to reduce food waste and address food insecurity.
The purpose of this document is to:

- Define all system requirements clearly
- Serve as a reference for development and testing
- Ensure all stakeholders understand the project scope
- Guide the development process

1.2 Intended Audience

This document is intended for:

- Student Developer: To understand what to build
- Project Guide/Supervisor: To review and approve requirements
- Evaluators: To assess project scope and implementation
- Testers: To create test cases based on requirements

1.3 Intended Use

This SRS document should be used to:

- Guide the development of the web application
- Create test plans and test cases
- Validate that all requirements are implemented
- Resolve any disputes about functionality

1.4 Product Scope

The Community Food Sharing Platform is a web application designed to:

- Reduce food waste by connecting surplus food with those in need
- Provide easy access to free/low-cost food for individuals facing food insecurity
- Build community connections between food donors and recipients
- Promote sustainable practices and social responsibility

What this product IS:

- A responsive web application accessible via browser
- A platform for posting and claiming food
- A communication tool between donors and recipients

What this product IS NOT:

- A mobile app (iOS/Android)
- A delivery service
- A payment processing system

- A food quality verification system

1.5 Definitions and Acronyms

Term	Definition
Donor	User who posts available food
Recipient	User who searches for and claims food
Listing	A food post created by a donor
Claim	Action of a recipient reserving food
SRS	Software Requirements Specification
API	Application Programming Interface
UI	User Interface
CRUD	Create, Read, Update, Delete
GPS	Global Positioning System
MVP	Minimum Viable Product

---

2. OVERALL DESCRIPTION

2.1 Product Perspective

This is a new, standalone web application built as an academic project. It will be developed using:

- Frontend: HTML, CSS, JavaScript
- Backend: Node.js with Express framework
- Database: MongoDB
- APIs: Google Maps API for location services

The system will consist of:

- Web interface (accessible via browser)
- Backend server for business logic
- Database for data storage
- Third-party map integration

2.2 User Needs

Primary Users:

1. Food Donors (Restaurants, Cafes, Households)

Need: Quick and easy way to post surplus food
Problem: Food going to waste at end of day
Solution: Simple form to list available food with pickup details
Goal: Reduce waste, help community, feel good about contribution

2. Food Recipients (Individuals, Families, Students)
Need: Find free/affordable food nearby
Problem: Food insecurity, tight budget
Solution: Search nearby food by location, see available options
Goal: Access nutritious food, save money

3. System Administrator
Need: Monitor and manage platform
Problem: Need to remove inappropriate content and spam
Solution: Admin dashboard with user/listing management
Goal: Keep platform safe and functional

2.3 Assumptions and Dependencies

Assumptions:

- Users have access to internet and a device (computer/smartphone)
- Users will provide accurate information about food
- Donors and recipients will coordinate pickup honestly
- Local regulations allow peer-to-peer food sharing
- Users understand basic web navigation

Dependencies:

- Google Maps API - For location services and map display
- Internet Connection - Required for all users
- Modern Web Browser - Chrome, Firefox, Safari, or Edge
- Email Service - For sending notifications (NodeMailer)

Technical Dependencies:

- Node.js runtime environment
- MongoDB database service
- Express.js framework
- Web hosting service (Netlify/Render)

---

3. SYSTEM FEATURES AND REQUIREMENTS

3.1 Functional Requirements

FR1: User Registration
Description: The system allows new users to create an account.

• The system shall display a registration form.
• The system shall validate user inputs.
• The system shall store user details securely.

```
Use Case: Register User
Actor: User
Given the user is on the registration page
When the user enters valid details and submits
Then the system creates a new account
```

FR2: User Login
Description: Registered users can log into the system.
• The system shall authenticate user credentials.
• The system shall allow access only to valid users.

```
Use Case: Login User
Actor: User
Given the user has a registered account
When the user enters correct login details
Then the system grants access to the dashboard
```

FR3: Post Food Listing
Description: Donors can post available food.
• The system shall allow donors to add food details.
• The system shall save listings in the database.

```
Use Case: Post Food
Actor: Donor
Given the donor is logged in
When the donor submits food details
Then the food listing is created
```

FR4: Browse and Search Food
Description: Recipients can browse available food listings.
• The system shall display all available food.
• The system shall allow searching by location.

FR5: Claim Food
Description: Recipients can claim food listings.
• The system shall mark food as claimed.
• The system shall notify the donor.

FR6: Messaging System
Description: Users can communicate after a claim.
• The system shall allow message exchange.

3.2 Non-Functional Requirements

Performance:
• The system shall load pages within 3 seconds.

Security:
• Passwords shall be encrypted.
• Unauthorized access shall be prevented.
Usability:
• The system shall be easy to understand.
• The interface shall be responsive.

3.3 External Interface Requirements

User Interface:
• Simple and user-friendly design

Software Interface:
• Database system
• Location service

3.4 System Features

• User authentication
• Food listing management
• Food claiming system
• Messaging system

---

4. OTHER REQUIREMENTS

4.1 Database Requirements

The system shall store:
• User details
• Food listings
• Claims
• Messages

4.2 Legal and Regulatory Requirements

• The platform is not responsible for food quality
• User data should be kept private

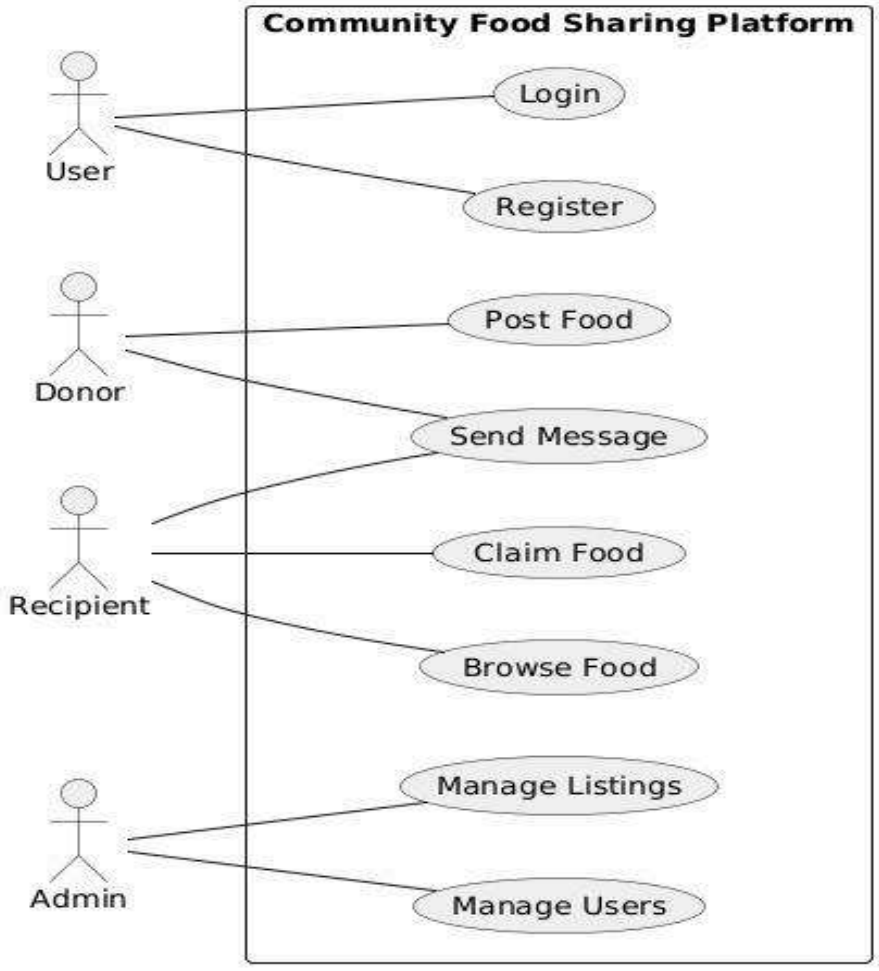
4.3 Risk Management (FMEA)

Risk	Impact	Solution
Fake food posts	High	Admin monitoring
Food not picked up	Medium	Claim confirmation
Server issues	High	Error

---

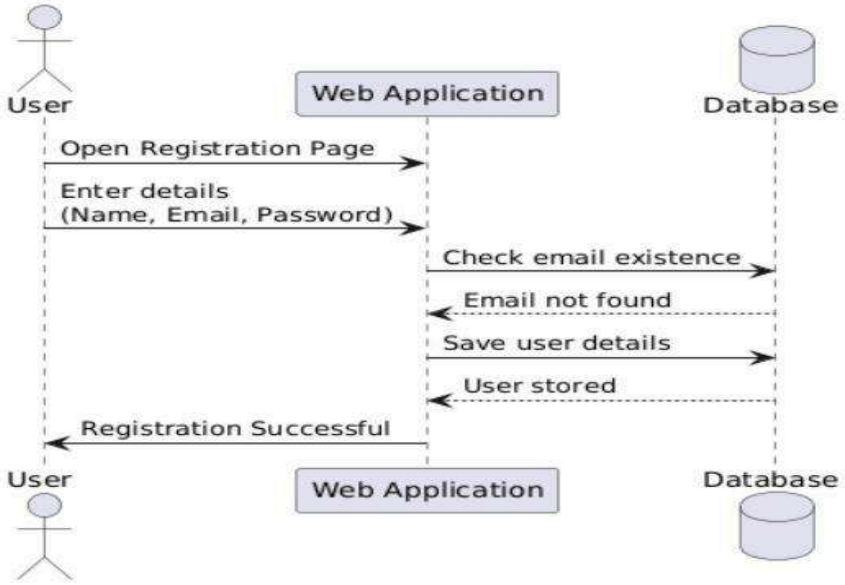
5. APPENDICES

5.1 Use Case Diagram

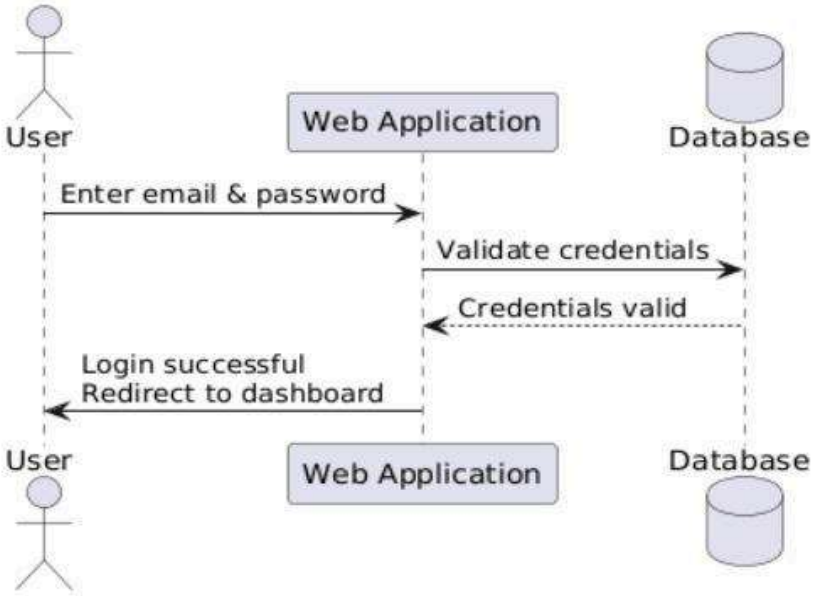


5.2 Sequence Diagram

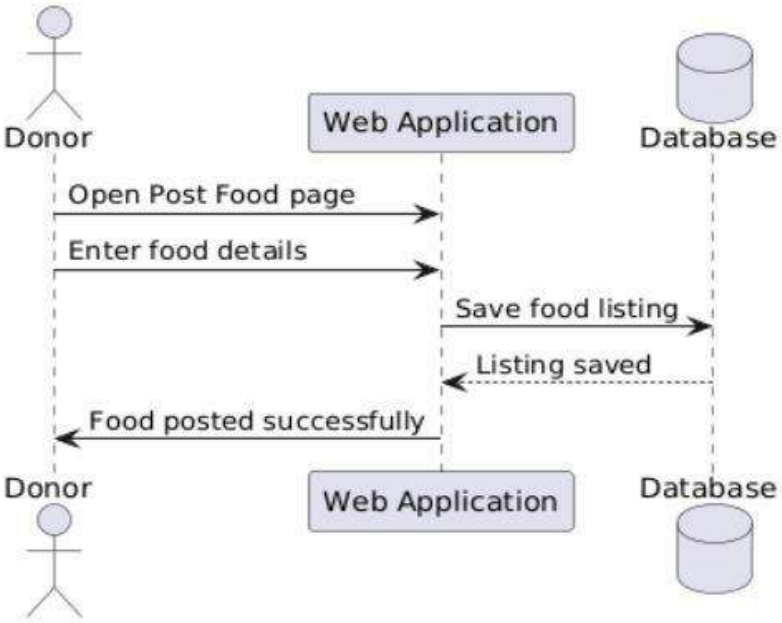
1. User Registration



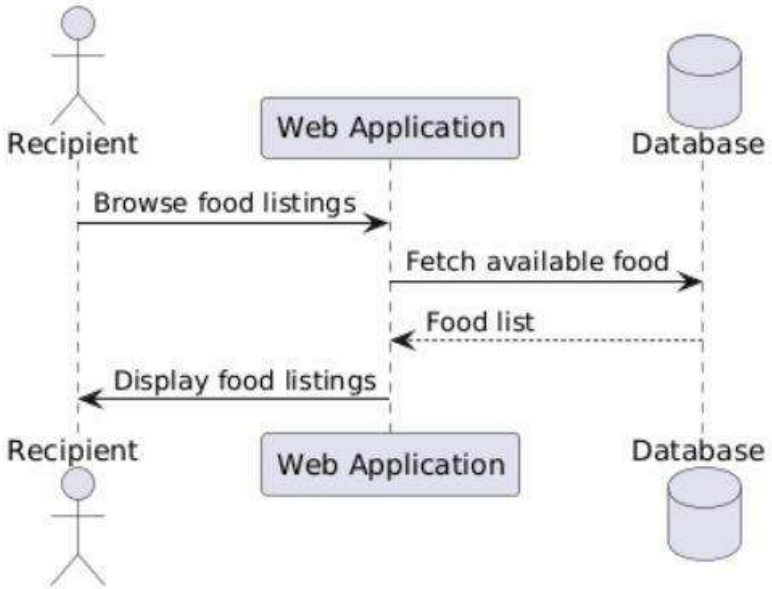
2. User Login



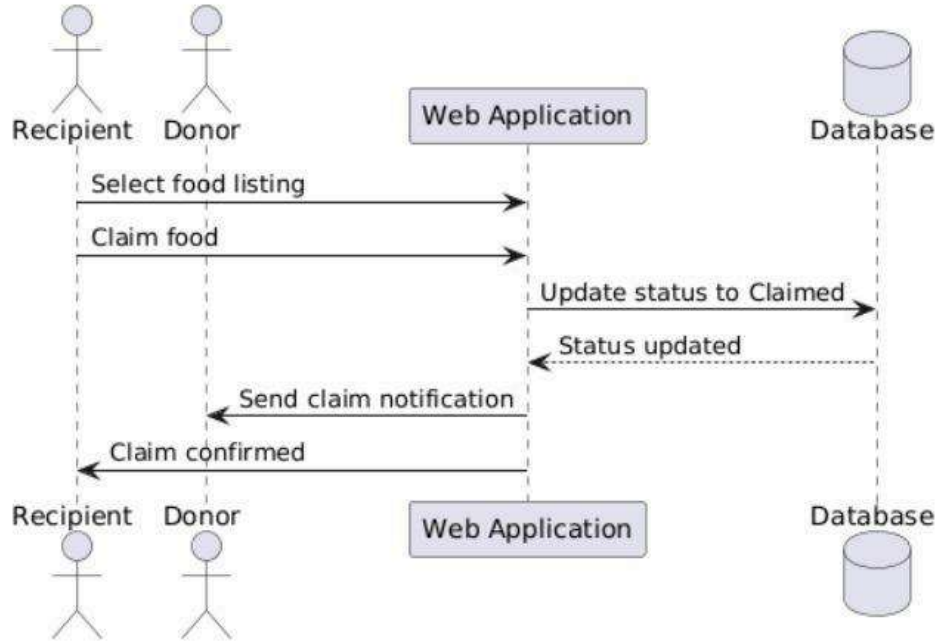
3. Post Food



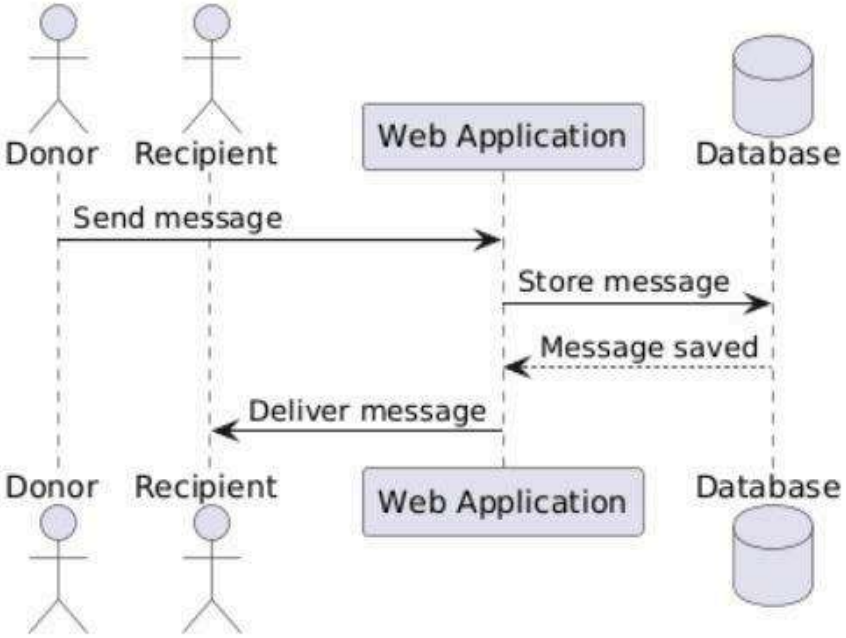
4. Browse Food



5. Claim Food



6. Send Message



5.3 To Be Determined (TBD)

- Image upload
- Rating system
- SMS alerts

END OF DOCUMENT