Web based food waste management solution

Author(s): Mohamed inamul hussain N

Version: 0.1

Date: March 27, 2024

1. Abstract:

The Food Waste Management Web Application aims to address the critical issue of food wastage by connecting donators with excess food to volunteers willing to distribute it to those in need. The application serves as a platform for efficient coordination and management of food donations, reducing waste while contributing to community welfare. The system facilitates user roles such as Donators, who can add and manage food donations, and Volunteers, who can view available donations and request specific items for distribution. The integration of maps allows users to visualize the locations of donators and volunteers, enhancing coordination and logistics in the donation process. Key features of the application include user registration and authentication, donation management with CRUD operations, map integration using Leaflet.js and OpenStreetMap, and educational content related to food waste reduction and sustainable practices. The system ensures data security through password hashing, access control mechanisms, and session management for authenticated users.

2.Objective and Scope

Objective of the Food Waste Management Web Application is to create a platform that connects donators and volunteers to efficiently distribute excess food and reduce food waste. The main goals of the application include:

- Facilitating the donation process: Allow donators to easily add information about surplus food items they want to donate, including item details, quantities, and expiry dates.
- Enabling volunteers to request and distribute food: Provide volunteers with access to available food donations, allowing them to request specific items for distribution to those in need.
- Enhancing visibility and coordination: Display the locations of donators and volunteers on a map to improve visibility and coordination in food distribution efforts.
- Promoting sustainability: Educate users about the importance of reducing food waste and encourage sustainable practices through the platform.

The scope of the Food Waste Management Web Application aims to connect donators and volunteers efficiently to distribute excess food and reduce waste. It includes user registration and authentication for Volunteers and Donators, with dashboards for managing donations and requesting items. Map integration using Leaflet.js and OpenStreetMap enhances visibility of locations. Educational content on reducing food waste is also provided. Excluded are advanced analytics, real-time tracking, payment processing, and mobile apps, focusing on core functionalities for a sustainable food distribution system.

3.Project End Users

There are two primary end user roles: Volunteers and Donators.

Volunteers: These users are individuals or organizations who actively participate in the distribution process by collecting and delivering excess food to those in need.

Role in the System:

- View available food donations: Volunteers can browse through the list of available food items donated by users.
- Request specific items: Volunteers have the option to request specific food items from Donators based on their availability and need.
- Interact with Donators: Volunteers can communicate with Donators regarding the pickup or delivery of food items.
- Access to map integration: Volunteers can view the geographic locations of Donators and plan their distribution routes efficiently using the integrated maps.

Donators: These users are individuals, businesses, or organizations that have excess food and want to donate it to reduce waste and contribute to community welfare.

Role in the System:

- Add new food donations: Donators can add details about surplus food items they wish to donate, including item name, quantity, expiry date, and location.
- Manage existing donations: Donators can edit or remove food donations from their list as needed.
- Respond to volunteer requests: Donators can view requests from Volunteers and coordinate the pickup or delivery of donated items.
- Access to map integration: Donators can see the geographic locations of other Donators and Volunteers, facilitating coordination and communication.

4.Module Description

4.1 User Management Module

Registration and Authentication:

The module facilitates user registration for both Volunteers and Donators, allowing individuals to create accounts on the platform. During registration, users provide necessary information such as name, email, and choose a password. The registration process validates user inputs and ensures that each user has a unique account on the system. Once registered, users can use their credentials to log in to the application.

Login Functionality:

Registered users can access the application using the login functionality provided by the User Management Module. This process involves verifying user credentials (email and password) against stored data in the database. Successful authentication grants users access to their respective dashboards and features within the application.

User Roles Management:

The module also handles user roles management by differentiating between Volunteer and Donator roles. Each role comes with specific privileges and access rights within the application. For example, Volunteers may have permissions to view available food donations and request items, while Donators can add new donations and manage existing ones. User roles management ensures that users interact with the application based on their designated roles, contributing to a structured and efficient user experience.

4.2 Donator Dashboard Module

Add New Food Donations:

The module allows Donators to add new food donations to the platform. Donators are prompted to input essential details related to each donation, including:

- Item Details: Description or name of the food item being donated (e.g., "Fresh fruits," "Canned goods," "Bakery items," etc.).
- Quantity: The amount or quantity of the food item being donated (e.g., number of kilograms, units, etc.).

- Expiry Date: Information about the expiration date of the donated food item, ensuring transparency and helping volunteers manage distributions efficiently.

The interface for adding new donations typically includes input fields, dropdown menus, and date pickers to capture these details accurately. Once a new donation is added, it becomes visible to Volunteers who can then request or distribute the items based on need and availability.

View and Manage Existing Donations (CRUD Operations):
The Donator Dashboard also provides functionalities to view and manage existing food donations. This includes performing CRUD (Create, Read, Update, Delete) operations on donation entries:

- **View Donations:** Donators can view a list or grid view of all their existing donations, displaying relevant details such as item name, quantity, and expiry date. This overview helps Donators keep track of their contributions.
- **Edit Donations:** Donators have the option to edit the details of their existing donations. They can modify item descriptions, quantities, or update expiry dates as needed. This flexibility ensures accurate and up-to-date information about the donated items.
- **Delete Donations:** In case a donation is no longer available or needs to be removed from the system, Donators can delete the donation entry. This action removes the donation from the platform and updates the database accordingly.

4.3 Volunteer Dashboard Module

View Available Food Donations:
Volunteers can access a comprehensive list or grid view of available food donations contributed by Donators. This view typically includes details such as item names, quantities, expiry dates, and sometimes additional information provided by Donators (e.g., food category, pickup instructions). Having visibility into available donations helps Volunteers identify items that match community needs or requests.

Request Specific Food Items from Donators:
Volunteers have the ability to request specific food items from Donators based on their availability and the needs of the community or beneficiaries. The request process involves selecting desired items from the available donations and sending a formal request to the respective Donator. This feature promotes targeted and efficient distribution of food

resources, ensuring that donated items align with actual demand.

View Locations of Donators and Volunteers on the Map:
One of the key functionalities of the Volunteer Dashboard is the integration of maps, such as Leaflet.js and OpenStreetMap (OSM), to display the geographic locations of both Donators and Volunteers. This visual representation on the map allows Volunteers to:

Identify Donator Locations:
Volunteers can see markers or pins on the map indicating the locations of Donators who have made food donations. This information helps Volunteers plan their pickup or delivery routes efficiently, reducing travel time and optimizing resource utilization.

4.4 Map Integration Module

4.4.1 The Map Display:

Initialize and Display Maps:
The Map Integration Module utilizes Leaflet.js, a popular JavaScript library, to initialize and display interactive maps within the web application. Leaflet.js provides a user-friendly interface for rendering maps and adding layers, markers, and other interactive elements.

Utilize OpenStreetMap (OSM) Tiles:
The module leverages OpenStreetMap (OSM) tiles for map visualization. OSM is a collaborative mapping project that provides free and open-source geographic data, including map tiles that can be used to create custom maps in web applications. By utilizing OSM tiles, the application can display detailed and customizable maps without the need for proprietary map services.

4.4.2 Location Visualization:

Fetch and Display Donator and Volunteer Locations:

- The Map Integration Module dynamically fetches and displays the locations of Donators and Volunteers on the map. This functionality involves:

- Retrieving location data from the database or API endpoints associated with user profiles.

- Plotting markers or icons on the map to represent the geographic coordinates of Donators and Volunteers.

- Updating the map in real-time or periodically to reflect changes in user locations or activities.

4.5 Database Management Module

User Data Storage:

Store User Information: The Database Management Module is responsible for storing user-related data such as:

Username: User's chosen name or identifier.

Email: User's email address used for communication and login.

Password Hash: Securely hashed version of the user's password for authentication.

Roles: Information about the user's role in the system (e.g., Volunteer, Donator).

Donation Data Storage:

Store Food Donation Details: The module also manages the storage of donation-related information, including:

Item Name: Description of the donated food item.

Quantity: Amount or quantity of the donated food item.

Expiry Date: Information about the expiration date of perishable items.

Location Coordinates: Geographic coordinates (latitude and longitude) indicating the location of the donation (for mapping purposes).

4.6 Authentication and Authorization Module

User Authentication:

Authenticate Users:

Comparing entered login credentials (email and password) with stored data in the database.
Using server-side verification techniques to ensure the authenticity of user identities.
Granting access to authenticated users based on successful verification.

Access Control:

Authorize Feature Access: The module implements access control mechanisms to authorize user access to specific features or functionalities based on their roles. For example: Donators may have access to features related to adding and managing donations. Volunteers may have access to features related to viewing available donations, requesting items, and interacting with the map interface.

5 Functional and Non-Functional Requirements

5.1 Functional Requirements:

User Management:

- Users can register as Volunteers or Donators.
- Users can log in and log out of their accounts.
- Volunteers can view available food donations.
- Volunteers can request specific food items from Donators.
- Donators can add new food donations with details like item name, quantity, and expiry date.
- Donators can manage existing food donations (CRUD operations).

Map Integration:

The application must display locations of Donators and Volunteers on an interactive map. Donators and Volunteers must be able to see their respective locations on the map. The map should update dynamically to reflect real-time locations.

Dashboard Features:

Volunteers should have a dashboard to manage their requested items and view donation history. Donators should have a dashboard to manage their donated items and view donation history.

Data Management:

User data (username, email, password hash, roles) and donation data (item name, quantity, expiry date, location coordinates) must be stored securely in the database. CRUD operations must be available for managing user and donation data.

5.2 Non- Functional Requirements:

To ensure optimal performance, security, user experience, reliability, scalability, and compatibility. Performance standards mandate that user actions like login and donation requests must elicit responses within 2 seconds, while the map integration should load swiftly and display locations without delays. Security measures encompass securely hashing user passwords using robust algorithms such as bcrypt and implementing strict access controls for sensitive data based on user authentication and authorization. The user interface is expected to be intuitive and user-friendly for both Volunteers and Donators, with a particular emphasis on an interactive and easy-to-navigate map interface. Reliability considerations necessitate an uptime of at least 99.9% and regular data backups to mitigate risks of data loss during system failures. Scalability features involve efficiently handling large user volumes, donations, and requests without compromising performance, utilizing techniques like load balancing and database optimization. Lastly, the application must be compatible across modern web browsers, including Chrome, Firefox, Safari, and Edge, as well as mobile devices, ensuring a seamless and responsive user experience across different screen sizes and resolutions. These requirements collectively ensure a robust, secure, and user-centric food waste management platform.

6 Lower-level design:

The lower-level design of the Food Waste Management Web Application involves several key components and technologies across the frontend and backend layers. On the frontend, the user interface (UI) will be developed using HTML, CSS, and JavaScript to ensure a user-friendly experience for both Volunteers and Donators. This includes creating forms for user registration, login, donation submissions, and request actions. Additionally, a map interface will be implemented using Leaflet.js to display the locations of Donators and Volunteers dynamically. In backend, a web server will be built using Node.js with Express.js or Python with Flask/Django to handle HTTP requests and responses. The authentication module will use JWT (JSON Web Tokens) for secure login sessions, while user data such as usernames, emails, hashed passwords, and donation details will be stored in a MySQL database. RESTful APIs will be created for CRUD operations on user accounts, donations, and map data, with business logic implemented for donation management, user roles, and authorization checks. Integration with external services such as SMTP for email notifications and Google Analytics for tracking user interactions and system performance will be incorporated.

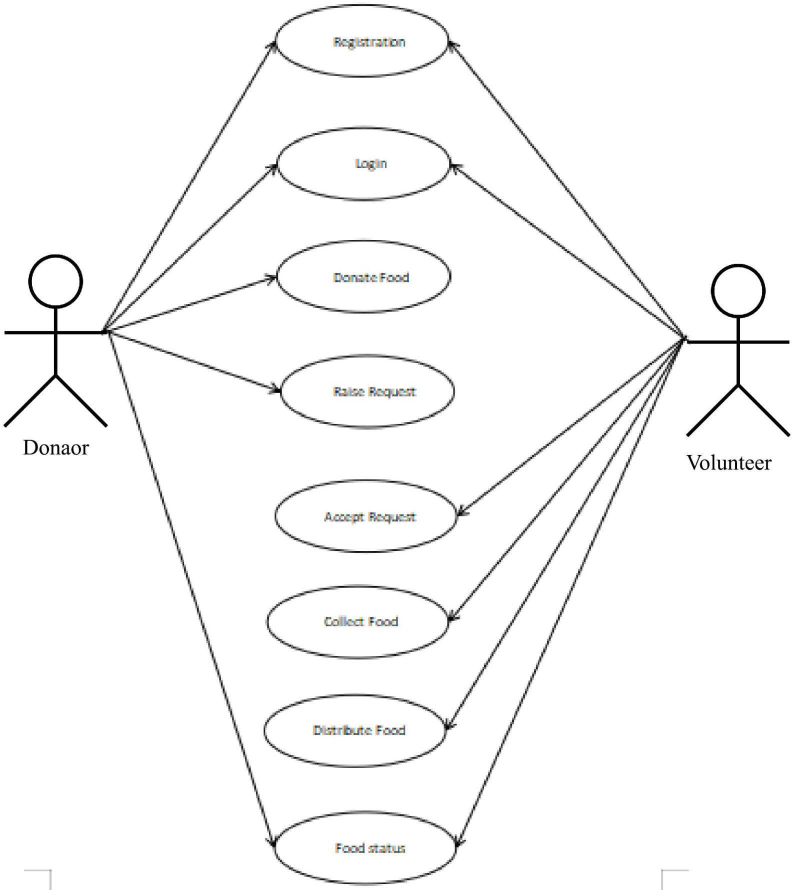
7. Higher-level design:

The pages are created using HTML, CSS, and JavaScript, ensuring compatibility across different devices and screen sizes. On the backend, Node.js or Python with frameworks like Express.js or Django will be used to handle server-side logic, including user authentication, authorization, and database interactions via RESTful APIs. Data will be stored in a suitable database system such as MySQL or MongoDB, with optimized schemas for user data, donation details, and geographical information. The application will integrate mapping functionality using libraries like Leaflet.js or Google Maps API to visualize the locations of Donators and Volunteers, alongside geocoding services for accurate map rendering. Security measures will include password hashing, role-based access control (RBAC), and

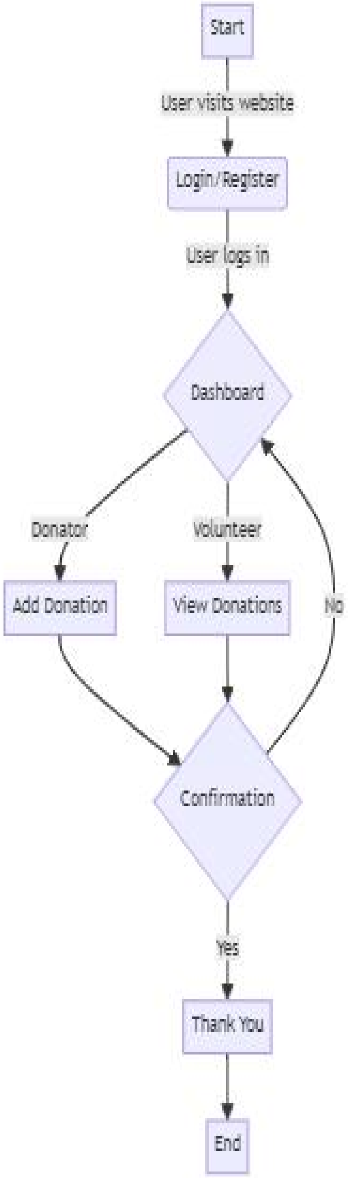
HTTPS encryption to safeguard user data and ensure secure communication. External services such as email providers and analytics tools will be integrated for functionalities like notifications and performance monitoring. Load balancing and auto-scaling mechanisms will be configured to manage varying levels of traffic and ensure optimal performance during peak usage. Extensive testing, including unit testing, integration testing, and user acceptance testing (UAT), will be conducted to validate functionality, performance, and user experience before deployment. This comprehensive approach ensures the development of a reliable, secure, and scalable Food Waste Management Web Application that meets user needs and industry standards.

8. Diagram:

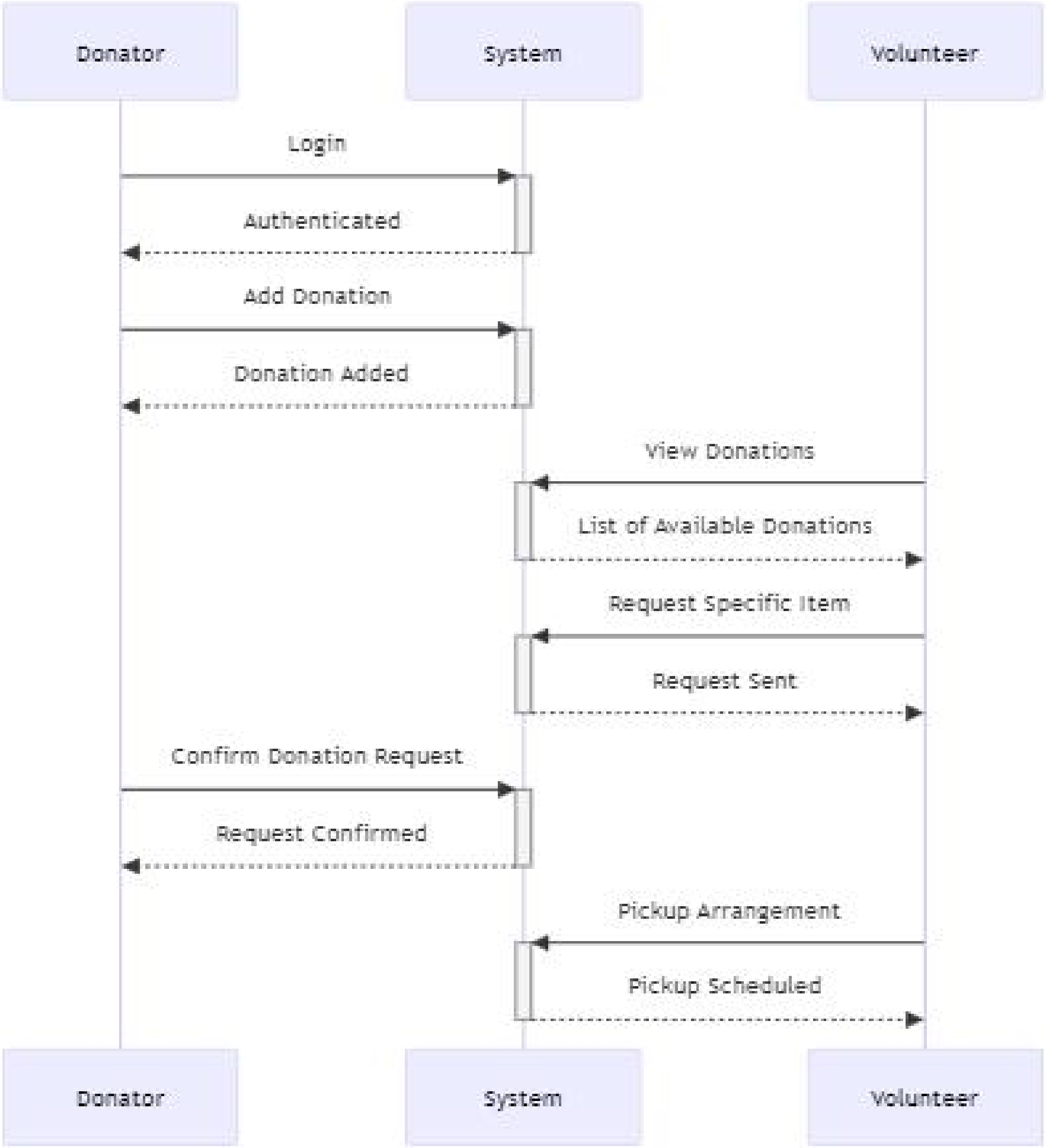
8.1 Use Case Diagram:



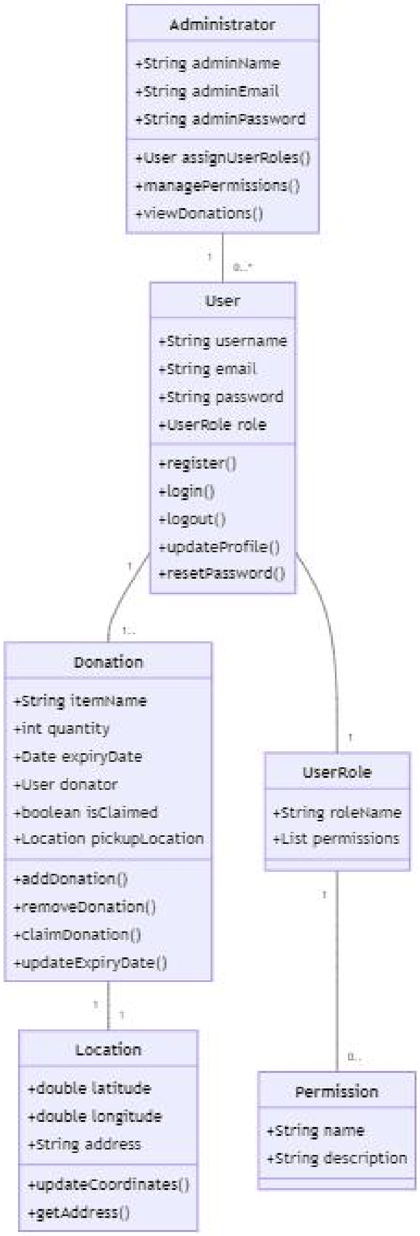
8.2 Flow chart:



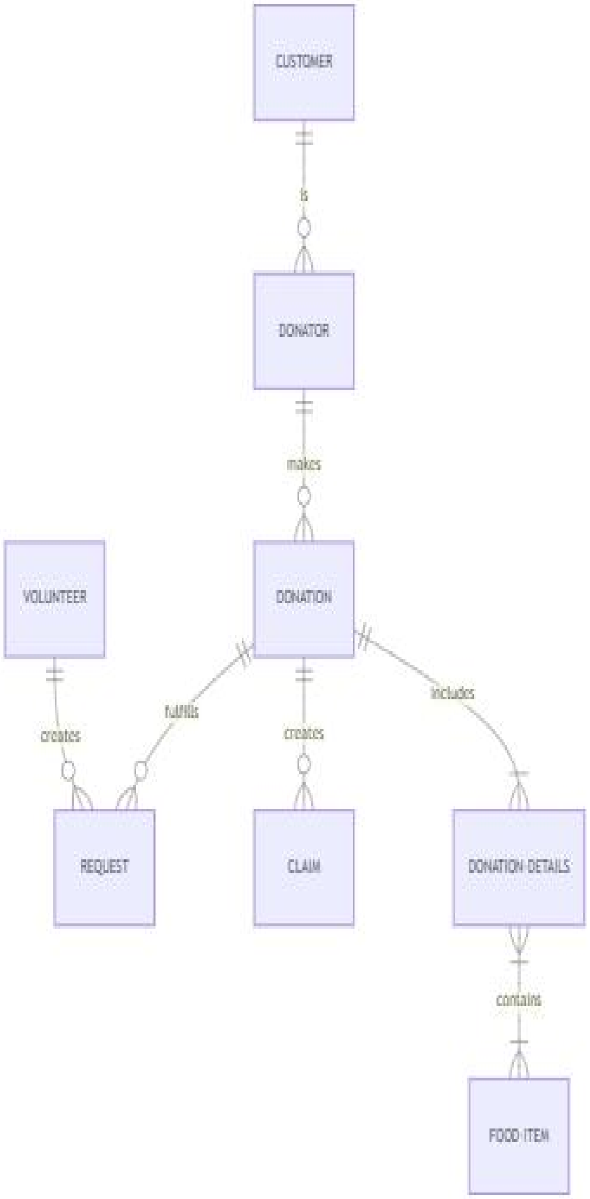
8.3 Sequence Diagram:



8.4 Class diagram:



9. ER Diagram:


CUSTOMER
is
DONATOR
makes
VOLUNTEER
DONATION
fulfills
creates
creates
includes
REQUEST
CLAIM
DONATION-DETAILS
contains
FOOD ITEM

10. Test Cases:

Test Cases	Test Purpose	Test condition	Expected outcome	Actual Result
User Registration	To verify that a user can successfully register on the platform.	User provides valid registration details (username, email, password).	User should be registered and redirected to the login page.	Registration successful, user redirected to login page.
User Login	To validate user login functionality.	User enters correct username and password.	User should be logged in and redirected to the dashboard.	Login successful, user redirected to dashboard.
Add Donation	To test the ability to add a new food donation.	Donator provides valid details for the donation (item name, quantity, expiry date).	Donation should be successfully added to the system.	Donation added successfully.
View Available Donations	To check if a volunteer can view available food donations.	Volunteer navigates to the available donations section.	Volunteer should see a list of available donations.	Volunteer can view available donations.
Request Food Item	To verify the process of requesting a specific food item.	Volunteer selects a donation and requests a food item.	Request should be sent successfully to the donator.	Request sent and acknowledged.

Map Integration	To ensure that the map displays donator and volunteer locations accurately.	Donator and volunteer locations are available in the system.	Map should show the locations of donators and volunteers.	Map displays locations correctly.
Update Profile	To verify that users can update their profile information.	User modifies their profile details such as email or password.	Profile information should be updated successfully.	Profile updated without any issues.
Reset Password	To test the password reset functionality.	User initiates a password reset request and follows the reset process.	User should be able to reset the password and log in with the new password.	Password reset successful, user logs in with new password.
View Donation Details	To ensure that users can view detailed information about a donation.	User selects a donation and views its details.	Donation details such as item name, quantity, and expiry date should be displayed accurately.	Donation details displayed correctly.
Search Functionality	To test the search feature for finding specific donations or users.	User enters search criteria (e.g., donation name, username) in the search bar.	System should return relevant search results based on the entered criteria.	Search functionality returns accurate results.