## Food Waste Reduction Platform

## 1. Version History (Should include Version #, Name of the Author and Date)

version: 1.0

author: Tianshu Liu 041120553

Sadegh Houshyar 041077975

Tianyang Huang 041103184

Feiling Xie 041104728

Date: 2024-03-18

# 2. Introduction (About the system and what does the document is about)

The Food Waste Reduction Platform (FWRP) is a comprehensive web-based application designed to address the global challenge of food waste. This platform serves as a crucial bridge connecting food retailers, consumers, and charitable organizations. By facilitating the efficient redistribution of surplus food, the FWRP aims to promote sustainability, reduce hunger, and enhance the resilience of food ecosystems. The platform is structured to enable users to register as retailers, consumers, or charitable organizations, each with distinct functionalities tailored to their roles. Retailers can manage inventory and identify surplus food items, consumers can purchase discounted food items, and charitable organizations can claim available food donations.   
This High-Level Design document aims to outline the architectural framework, design considerations, and functional components of the FWRP. The document is intended to provide a comprehensive overview of the system's structure, the technologies utilized, and the interaction between various modules and user types. It serves as a roadmap for the development team, stakeholders, and reviewers, offering insights into the platform's core functionalities, database design, and the integration of different layers of the application.

# 3. Targeted Audience (Who are the targeted audience for this document)

Project managers, Development Team, Quality Assurance (QA) Team, Stakeholders and Clients.

# 4. Scope (What is in and out of the scope of this document)

In Scope:

The HLDD will encompass high-level descriptions of the application architecture, business architecture, detailed design components, data architecture, security considerations, deployment architecture, and testing models for the Food Waste Reduction Platform (FWRP).

Out of Scope:

The document will not include detailed implementation specifics, project management aspects, user interface design details, or operational procedures. These elements will be addressed in separate documents or manuals.

## 5. Application Architecture (High level architecture/overview of entire system/ main component etc.)

***5.1 System Overview***

#### ***5.1.1 Purpose and Scope:*** Define the objective of the Food Waste Reduction Platform, emphasizing its role in reducing food waste and connecting retailers, consumers, and charitable organizations.

#### ***5.1.2 Users and Stakeholders:*** Describe the different users of the system (retailers, consumers, charitable organizations) and any other stakeholders involved (e.g., regulatory bodies, environmental organizations).

#### ***5.1.3 High-Level Workflow:*** Provide a general overview of how the system works from the perspectives of different users. For example, retailers listing surplus food, consumers and charities viewing and claiming items, and the process of receiving notifications.

***5.2 Component Architecture***

***5.2.1 Presentation Layer (Frontend):***

* Technologies Used: Detail the use of HTML, CSS, and JSP for developing the user interface.
* User Interaction: Explain how users will interact with the system through this layer, including the registration, listing, claiming, and purchasing processes.
* Responsiveness and Accessibility: Mention how the design caters to various devices and user needs.

***5.2.2 Business Logic Layer (Backend):***

* + Servlets and Java Classes: Discuss the use of Servlets for handling business logic and processing user requests, including user authentication, inventory management, and transaction processing.
  + Design Patterns: Highlight any design patterns used (Singleton for database connections, Factory for creating objects).
  + Security Measures: Outline how the system ensures security, such as password hashing, input validation, and session management.

***5.2.3 Data Access Layer (Database Management):***

* + Database Technology: Specify the use of JDBC for database connectivity and the choice of SQL-based RDBMS (e.g., MySQL, PostgreSQL).
  + Schema Design: Briefly mention the database schema, including key tables and relationships (refer to a detailed schema in the database design section).
  + Data Security and Integrity: Describe measures for protecting data, such as secure connections, backup strategies, and integrity constraints.

6. Business Architecture (Use Case diagrams along with the description)   
A diagram of food waste reduction platform

Description automatically generated

Figure 1 Use Case Diagram

Description:

* Retailers are central to the platform's operations, managing inventory and identifying surplus food items. They list these items for donation or sale at discounted rates, contributing directly to waste reduction.
* Consumers engage with the platform by purchasing discounted items, thereby helping to decrease food waste. They can also subscribe to alerts for surplus food, ensuring they are informed when opportunities to purchase discounted items arise.
* Charitable Organizations, such as food banks, play a crucial role by claiming surplus food items listed by retailers, aiding in the distribution of food to those in need.
* The System is responsible for backend processes, updating inventories after transactions, and sending out automated notifications to subscribers about surplus food availability.

User Stories:

**For Retailers:**

* As a retailer, I want to manage my inventory and identify surplus food items, so that I can list them for donation or sale at discounted rates and contribute to waste reduction.

**For Consumers:**

* As a consumer, I want to purchase discounted items, so that I can help decrease food waste.
* As a consumer, I want to subscribe to alerts for surplus food, so that I am informed when opportunities to purchase discounted items arise.
* As a consumer, I want to post surplus food on the message board, so that it can be redistributed to those who need it rather than going to waste.

**For Charitable Organizations:**

* As a charitable organization, I want to claim surplus food items listed by retailers, so that we can aid in the distribution of food to those in need.
* As a charitable organization, I want to monitor the message board for surplus food posted by consumers, so that we can receive and distribute this food effectively.

7. Detailed Design (Class diagrams, Component diagrams etc.)

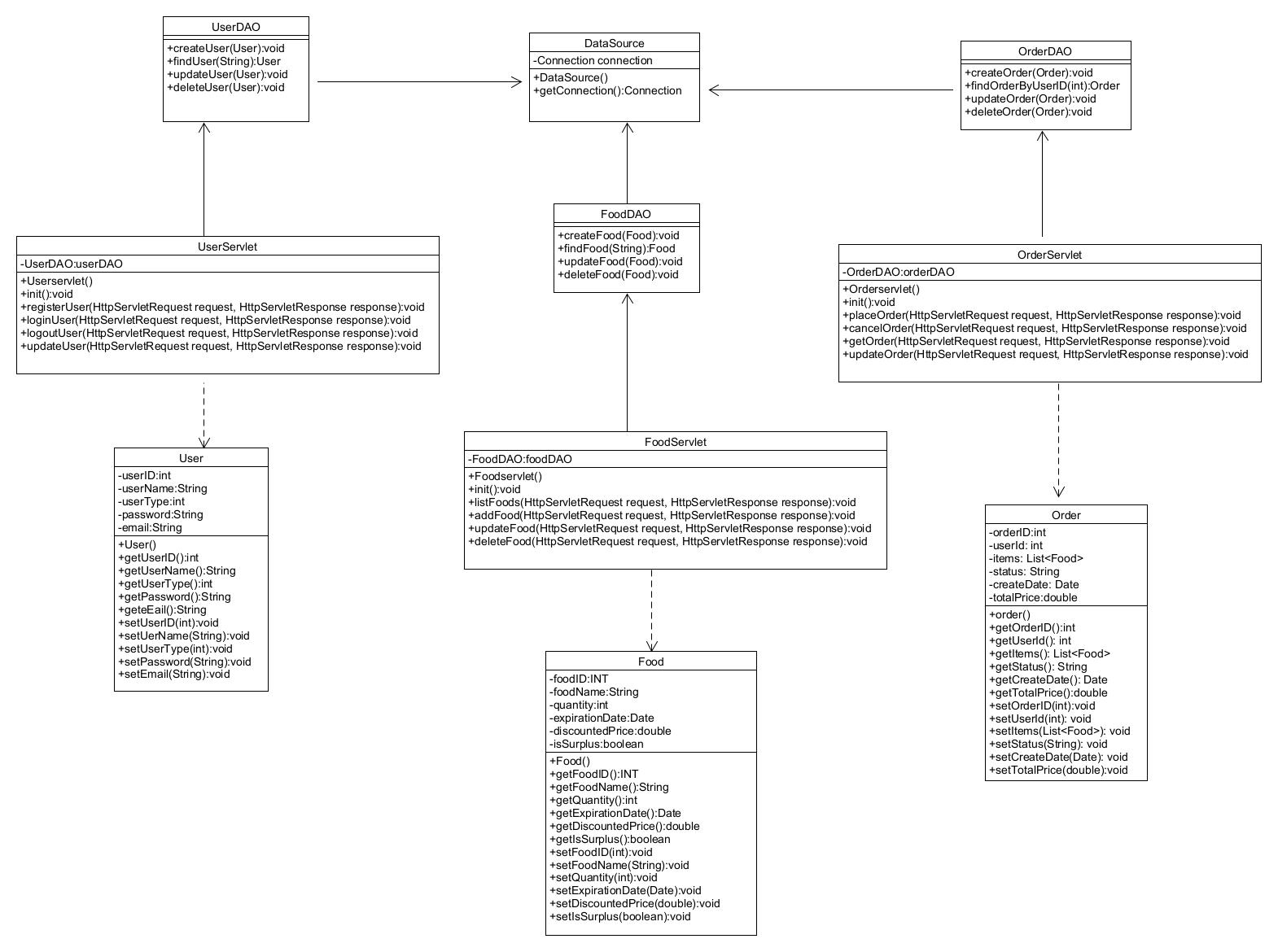


Figure 2 Class Diagram

8. Data Architecture (Database structures, ERD, Physical/Logical Data Model)

***8.1 Logical Data Model***

Users

Attributes: UserID, Name, Email, Password, UserType

Retailers

Attributes: RetailerID (links to Users), StoreName, Location

Charitable Organizations

Attributes: OrganizationID (links to Users), OrganizationName, Location

Food Items

Attributes: ItemID, RetailerID (links to Retailers), Name, Quantity, ExpiryDate, IsSurplus

Transactions

Attributes: TransactionID, ItemID (links to Food Items), UserID (links to Users), Quantity, TransactionType, TransactionDate

Subscriptions

Attributes: SubscriptionID, UserID (links to Users), Preferences, Location, ContactMethod

***8.2 Physical Data Model***

Users Table:

UserID: INT, Primary Key, Auto-Increment

Name: VARCHAR(255)

Email: VARCHAR(255), Unique

Password: VARCHAR(255)

UserType: ENUM('Retailer', 'Consumer', 'Charitable Organization')

Retailers Table:

RetailerID: INT, Primary Key, Foreign Key (Users.UserID)

StoreName: VARCHAR(255)

Location: VARCHAR(255)

Charitable Organizations Table:

OrganizationID: INT, Primary Key, Foreign Key (Users.UserID)

OrganizationName: VARCHAR(255)

Location: VARCHAR(255)

Food Items Table:

ItemID: INT, Primary Key, Auto-Increment

RetailerID: INT, Foreign Key (Retailers.RetailerID)

Name: VARCHAR(255)

Quantity: INT

ExpiryDate: DATE

IsSurplus: BOOLEAN

Transactions Table:

TransactionID: INT, Primary Key, Auto-Increment

ItemID: INT, Foreign Key (Food Items.ItemID)

UserID: INT, Foreign Key (Users.UserID)

Quantity: INT

TransactionType: ENUM('Purchase', 'Donation')

TransactionDate: DATETIME

Subscriptions Table:

SubscriptionID: INT, Primary Key, Auto-Increment

UserID: INT, Foreign Key (Users.UserID)

Preferences: TEXT

Location: VARCHAR(255)

ContactMethod: ENUM('Email', 'Phone')

8.3 ERD diagram

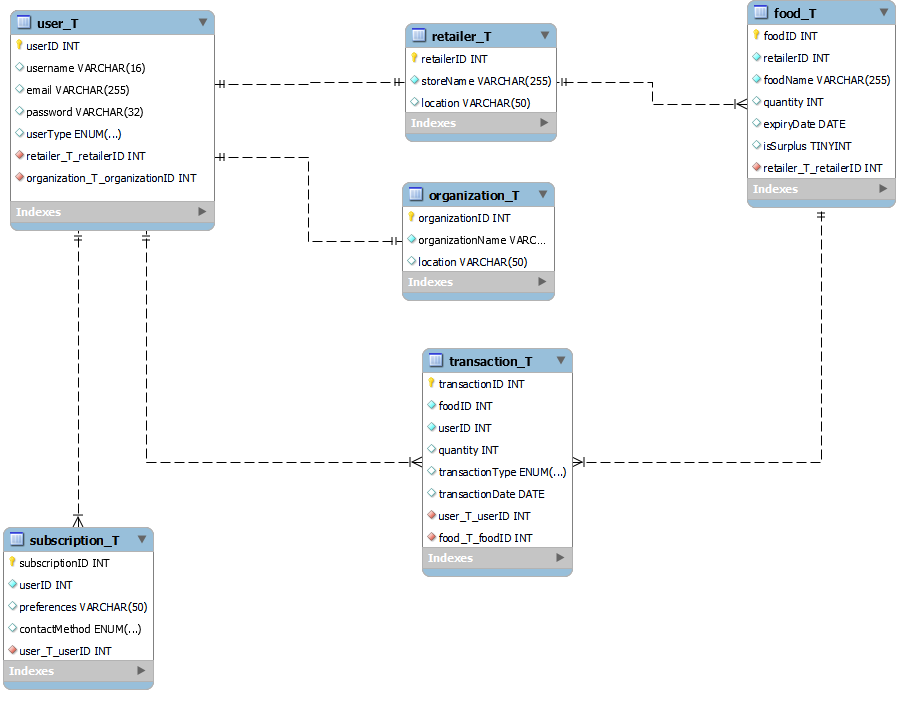


Figure 3 ERD Diagram

9. Security Architecture (What are the security consideration in your designs)

***9.1 Authentication and Authorization***

* Detail the mechanisms for secure registration, login, and logout processes, including password storage (e.g., password hashing and salting) and session management.
* Discuss the implementation of user roles and permissions, ensuring retailers, consumers, and charitable organizations have access only to appropriate features and data.
* Describe any third-party services for authentication

***9.2 Data Protection and Privacy***

* Explain the encryption of sensitive data both in the database and in transit.
* Discuss how the platform will handle personal data and the mechanisms for ensuring privacy, such as anonymization techniques.

10. Deployment Architecture (Infrastructure, deployment model etc.)

***10.1 Infrastructure***

* Describe a straightforward infrastructure setup that could include a basic web server for hosting the JSP pages and servlets, and an application server where the business logic runs.
* Outline a single-instance database server that will host the FWRP schema and handle transactions.

***10.2 Deployment model***

* Discuss a basic deployment process which includes pulling the latest stable version of the application from a version control system (like GitHub), building the application on a development machine, and then transferring the build artifacts to the server.

11. Testing Model

***11.1 JUnit Testing***

Outline the strategy for unit testing with JUnit, specifying the coverage goals and how the tests will integrate with the CI/CD pipeline.

***11.2 API Testing***

Outline the approach for API testing, emphasizing the validation of RESTful endpoints for correctness, reliability, and performance.

Describe the use of API testing tools such as Postman or automated scripts with frameworks like REST-assured for testing GET, POST, UPDATE, and DELETE operations.

Discuss the procedures for testing request and response structures, error handling, and edge cases to ensure APIs respond correctly under various conditions.

***11.3 Manual Testing***

Detail the process for conducting manual test cases and scenarios that might not be covered by automated tests, such as complex user interactions, design and layout consistency, and overall user experience.

# 12. References

# 13. Acronyms/Abbreviation

* FWRP: Food Waste Reduction Platform
* HLDD: High-Level Design Document
* HTML: Hypertext Markup Language
* CSS: Cascading Style Sheets
* JSP: JavaServer Pages
* RDBMS: Relational Database Management System
* JDBC: Java Database Connectivity
* ERD: Entity-Relationship Diagram

# List of Figures

[Figure 1 Use Case Diagram 4](#_Toc161841343)

[Figure 2 Class Diagram 5](#_Toc161841344)

[Figure 3 ERD Diagram 8](#_Toc161841345)