High Level Design

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

   Version #: 1.0
   Author: Meet Maheta
   Date: 22/03/2024

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

   The Food Waste Reduction Platform aims to address the global issue of food waste by providing a comprehensive solution that connects food retailers, consumers, and charitable organizations. This document outlines the architecture and design considerations for the platform.

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

   This document is intended for software developers, architects, project managers, and stakeholders involved in the development and deployment of the Food Waste Reduction Platform.

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

In Scope:

- Design and architecture considerations for the Food Waste Reduction Platform.
- High-level overview of the application architecture, business architecture, detailed design, data architecture, security architecture, deployment architecture, and testing model.

Out of Scope:

- Detailed implementation code.
- Specific hardware or infrastructure requirements.

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

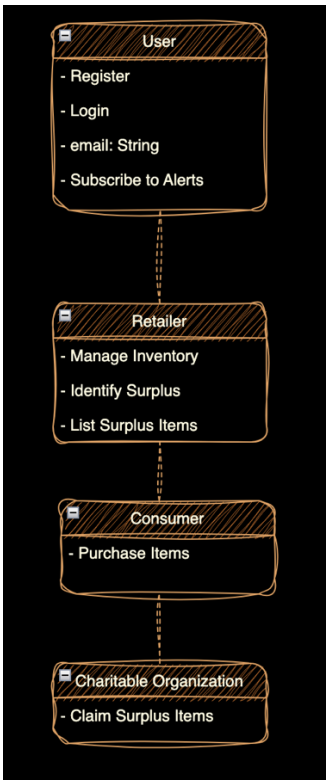
Overview
The application architecture of the Food Waste Reduction Platform consists of the following main components:

- Presentation Layer
- Business Layer
- Database Layer

Subheading 1: Presentation Layer
- Handles user interactions and displays data to users.
- Built using HTML, CSS, JavaScript for front-end development.

Subheading 2: Business Layer
- Contains the business logic and functionalities of the platform.
- Developed using Java/J2EE, utilizing design patterns for efficient implementation.

6. Business Architecture (Use Case diagrams along with the description)



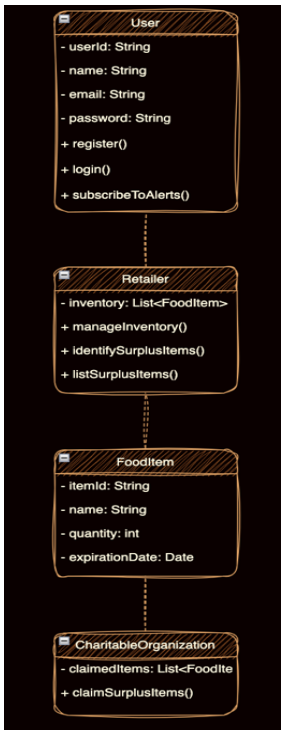
Explanation:

- User: Represents individuals who interact with the platform. They can register, login, and subscribe to alerts for surplus food items.
- Retailer: Represents food retailers who have surplus food items. They can manage their inventory, identify surplus items, and list them on the platform.
- Consumer: Represents individuals who purchase food items from retailers on the platform.
- Charitable Organization: Represents organizations such as food banks or missions.
- They can claim surplus items listed by retailers for redistribution.

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

Class Diagrams:

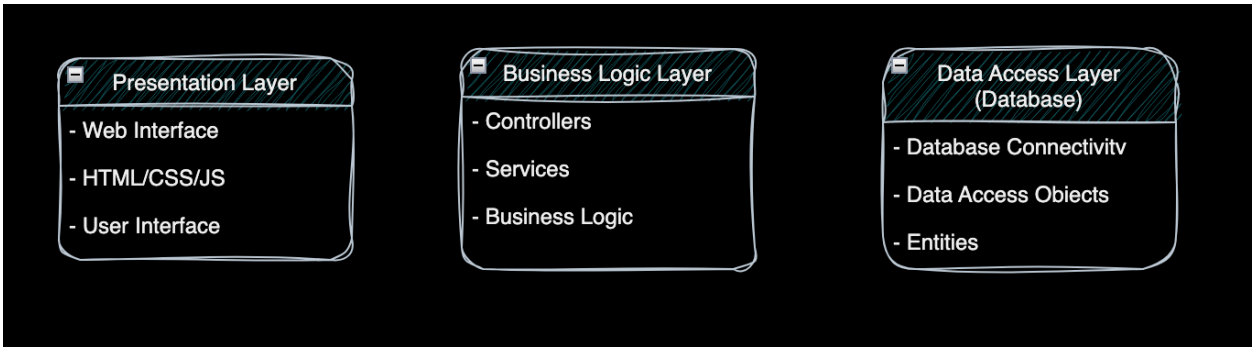
The class diagram illustrates the relationships between classes in the system.



Include class diagrams illustrating the relationships between classes in the system.

Component Diagrams:

The component diagram shows the high-level components and their interactions in the system.



Include component diagrams showing the high-level components and their interactions.

Explanation:

- Presentation Layer: Handles user interactions and displays data to users. It consists of web interfaces built using HTML/CSS/JS and user interfaces for different types of users.

- Business Logic Layer: Contains the business logic and functionalities of the platform. It includes controllers, services, and business logic components responsible for processing user requests and implementing platform functionalities.

- Data Access Layer (Database): Manages the interaction with the database. It includes components for database connectivity, data access objects (DAOs) for accessing database entities, and entity classes representing database tables.

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

Database Structures:

The Food Waste Reduction Platform utilizes a relational database management system (RDBMS) to store data.

Tables:

User Table:
userId (Primary Key)
name
email
password
userType (Retailer, Consumer, Charitable Organization)

Retailer Table:
retailerId (Primary Key, Foreign Key referencing User Table)
inventoryId (Foreign Key referencing Inventory Table)

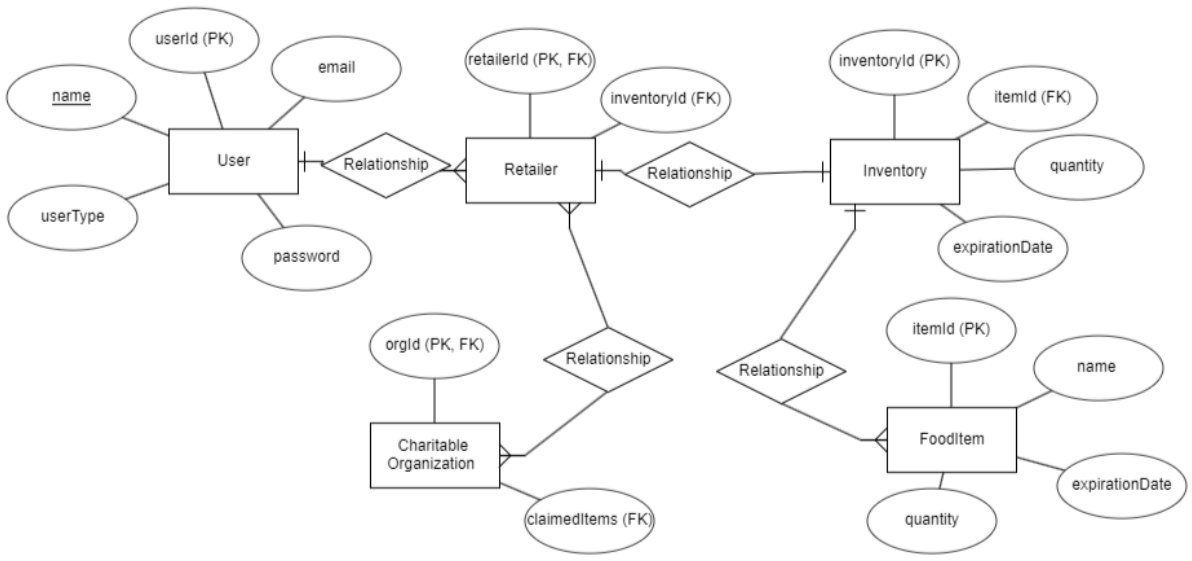
Inventory Table:
inventoryId (Primary Key)
itemId (Foreign Key referencing FoodItem Table)
quantity
expirationDate

FoodItem Table:
itemId (Primary Key)
name
quantity
expirationDate

CharitableOrganization Table:
orgId (Primary Key, Foreign Key referencing User Table)
claimedItems (Foreign Key referencing FoodItem Table)

Entity-Relationship Diagram (ERD)

The ERD depicts the entities and their relationships in the Food Waste Reduction Platform.



Include the ERD depicting the entities and their relationships.

Explanation:

- User: Represents all users of the system, including retailers, consumers, and charitable organizations. Each user has a unique userId.
- Retailer: Extends the User entity and represents retailers. Each retailer has a unique retailerId and can have an associated inventory.
- Inventory: Represents the inventory of a retailer, containing various food items. Each inventory has a unique inventoryId and can contain multiple food items.
- FoodItem: Represents individual food items with details such as name, quantity, and expiration date. Each food item has a unique itemId.
- CharitableOrganization: Extends the User entity and represents charitable organizations. Each organization has a unique orgId and can claim surplus food items from retailers.

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

The security architecture of the Food Waste Reduction Platform is designed to ensure the confidentiality, integrity, and availability of data, as well as protect against common security threats. Key security considerations and implementations include:

Authentication and Authorization:

Example: Implementing username/password authentication and role-based access control (RBAC) for user authentication and authorization.

Data Encryption:

Example: Encrypting sensitive user data such as passwords using bcrypt hashing algorithm before storing in the database.

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

Infrastructure

For deploying the Food Waste Reduction Platform, the following infrastructure components are required:

- **Server Environment:**

Utilize virtual or dedicated servers capable of running Java EE applications.

- **Database Server:**

Deploy a database server to host the relational database management system (RDBMS) for storing platform data.

- **Application Server:**

Set up an application server compatible with Java EE specifications to host the platform's web applications.

- **Networking Infrastructure:**

Establish networking infrastructure to facilitate communication between clients, servers, and databases.

Deployment Model

The deployment model for the Food Waste Reduction Platform is a cloud-based deployment model for its scalability, reliability, and cost-effectiveness.

11. Testing Model (How are you testing your application. Junit, API testing etc)

Unit Testing with Junit

- **Unit Testing of Business Logic Components:**

Write unit tests to validate the functionality of business logic components such as inventory management, surplus food identification, and listing of surplus food items.

Use JUnit framework to create test cases for individual methods and classes.

- **Unit Testing of API Endpoints:**

Write unit tests to validate the functionality of API endpoints such as user registration, food item listing, and donation claiming.

- **Integration Testing of Service Layer:**

Write integration tests to validate the interaction between service layer components such as InventoryService, UserService, and DonationService.

12. Acronyms/Abbreviation

- RDBMS - Relational Database Management System
- ERD - Entity-Relationship Diagram
- DAO - Data Access Object
- HTML - HyperText Markup Language
- CSS - Cascading Style Sheets
- JS - JavaScript
- J2EE - Java 2 Platform, Enterprise Edition
- RBAC - Role-Based Access Control
- JUnit - A unit testing framework for the Java programming language
- OAuth - Open Authorization (a protocol for authorization)