**S24 CST8288 040/041   
High Level Design Document**

**Group 1**

Loy Yee Ko

Adam Hoddinott

Josh Bennett

Tony Nguyen

Theory Professor: Sazzad Hossain

Lab Professor: Abul Qasim

Course: CST8288 040

Lab Section: 041

Algonquin College,

Computer Programming

Sunday July 21st, 2024

Table of Contents

[1. Version History 3](#_Toc172367660)

[2. Introduction 3](#_Toc172367661)

[3. Target Audience 3](#_Toc172367662)

[4. Scope 3](#_Toc172367663)

[5. Application Architecture 4](#_Toc172367664)

[5.1 Architecture Breakdown 4](#_Toc172367665)

[5.1.1 Presentation Layer 4](#_Toc172367666)

[5.1.2 Business Layer 4](#_Toc172367667)

[5.1.3 Data Access Layer 4](#_Toc172367668)

[6. Business Architecture 5](#_Toc172367669)

[6.1 User Authentication & Registration 5](#_Toc172367670)

[6.2 Retailer 6](#_Toc172367671)

[6.3 Charitable Organization 6](#_Toc172367672)

[3.4.5 Consumer 7](#_Toc172367673)

[6.4 Subscription & Notification 7](#_Toc172367674)

[7. Detailed Design 8](#_Toc172367675)

[7.1 Goals 8](#_Toc172367676)

[7.2 General Constraints 8](#_Toc172367677)

[7.3 Class Diagrams 9](#_Toc172367678)

[8. Data Architecture 12](#_Toc172367679)

[9. Security Architecture 12](#_Toc172367680)

[10. Deployment Architecture 13](#_Toc172367681)

[11. Testing Model 13](#_Toc172367682)

[12. References 13](#_Toc172367683)

[12.1 Tools 13](#_Toc172367684)

[13. Acronyms/Abbreviations 13](#_Toc172367685)

## Version History

Version: 1.0

## 2. Introduction

Food waste is a large issue that we face daily, although we consume food there are still a lot of products that unfortunately go to waste. To combat this problem, we created the Food Waste Reduction Project. This group project is a dynamic java web project that utilizes Java, JSP, Servlet to create the necessary classes, structure of application and the user interface.

This document will delve into more details regarding the inner workings of this program, from the backend design, user interface and connection to our server hosting the web application. The goal is to breakdown our project in a manner that allows a developer and user to understand the functionality of the program.

## 3. Target Audience

There are two groups we look to target with this project, users of the application and developers.

As mentioned earlier, this project is used as a solution to food waste. Retailers, consumers and charitable organizations are the main target audience since they will be using the application. Out of all the users, retailers bare the most weight as they are ones providing inventory to be bought/donated to the other two users.

We are also looking to target developers as this could open a door to further collaboration. In the future this would be helpful as the team could expand once the application goes onto a larger scale.

## 4. Scope

* Overall architecture: diagram of how users interact with the system in general
* Business Architecture: User interaction with the system: the use case diagram in specific operation
* Detailed Design: Class UML diagram describe the shape of the entities
* Database design: the physical model of the database
* Goals: listing the key features and functionalities

## 5. Application Architecture

A diagram of a business

Description automatically generated

## 5.1 Architecture Breakdown

With role-based authorization, the user can access different services such as adding new items or claiming items.

### 5.1.1 Presentation Layer

* User Application Interface (JSP)
* Consumer Controller
* Retailer Controller
* Charitable Organization Controller
* User Controller
* Subscription Controller

### 5.1.2 Business Layer

* User Registration and Authentication Service
* Food Item Inventory Management Service
* Email notification Service
* Phone (SMS) notification Service

### 5.1.3 Data Access Layer

* DAO - update and query persistent data
* RDBMS with MySQL- storing persistent data
* SQL - Query language for data retrieval and updates to the database table.

## 6. Business Architecture

## 6.1 User Authentication & Registration

Use Case Diagram for User.

A diagram of a system

Description automatically generated

* The User must input name, email, password, and user type
* The system provides the login and logout.

## 6.2 Retailer

Use Case Diagram

A diagram of a system

Description automatically generated

* Retail able to list surplus and non-surplus items
* Update the quantities
* Set discount rate
* Set expiry date
* Set items as donations or discounted goods
* Update after food is purchased or donated

## 6.3 Charitable Organization

A diagram of a system

Description automatically generated

* Claim food items listed for donation by retailers
* View list of donations

## 3.4.5 Consumer

A diagram of a system

Description automatically generated

* View regular food items
* View surplus food items
* Purchase from the retailer

## 6.4 Subscription & Notification

A diagram of a system

Description automatically generated

* Consumers and Charitable Organizations will receive notifications based on location and food preferences.
* The retailer updates the food item status as surplus; an email or phone notification will be sent if the user subscribes.

## 7. Detailed Design

## 7.1 Goals

A list of functional requirements of the application.

* User
  + Registrations: name, email, password, phone(optional), user type
  + Authentication: login, logout
* Retailer
  + View all items: if the item is 1 week before expiry, it should show as surplus.
  + View all surplus
  + Create, update, and delete items
  + Change item status as donated, purchased
  + Set if the item is for “donation”
  + Set quantities
  + Set discount rate
* Charitable Organization
  + Claim the food items
  + View the “donation” food items
* Consumer
  + View regular food items
  + View surplus food items
  + Purchase from the retailer
* Subscription
  + Consumer, Charitable Organizations subscribe to surplus food alerts
    - Based on location
    - Communication
      * Email
      * Phone (either or)
    - Food preferences
* Automated Notification
  + Email or Phone alerts when retailers list surplus food items on the platform.
* Extra functionalities

## 7.2 General Constraints

The application must be user-friendly, users can operate based on their user type. Automated notification is required, minimum steps for users to perform any operations, and the users do not need to know the detailed software implementation and architecture.

## 7.3 Class Diagrams

Controllers Class Diagram

A screenshot of a computer

Description automatically generated

Services Class Diagram

A screenshot of a computer program

Description automatically generated

Models Class Diagram

A screenshot of a computer

Description automatically generated

DAO Class Diagram

A diagram of a computer program

Description automatically generated

## 8. Data Architecture

A screenshot of a computer

Description automatically generated

* Simple database design with three tables, user, item, order, subscription, and notification
* Items can have zero or many users consuming/selling/donating items
* Users can consume/purchase/donation zero or many items
* Users, when registering on the registry page can select what type of user they are.
* Order table will have user\_id and item\_id as foreign keys

## 9. Security Architecture

* Role – Based Authorization with Servlet Filter interface
* Password Authentication filter with Servlet Filter
* Implement HTTPS with CertBot

## 10. Deployment Architecture

* MVC Architecture
* Cloud Deployment
  + Azure Virtual Machine – Single Instance for running the application
  + MySQL Cloud Database – Relational Data
  + Blob Storage – Image Storage
* Jenkins / GitHub Action for CICD

## 11. Testing Model

* JUnit 5 for Unit Testing
* Mockito for Mock
* JMeter for performance testing
* Selenium for frontend testing

## 12. References

* Azure Documentation. [Online]. Available: <https://learn.microsoft.com/en-us/azure/?product=popular>. [Accessed: Jul. 17, 2024].
* MySQL 8.0 Reference Manual. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/>. [Accessed: Jul. 17, 2024].
* Junit 5 Guide. [Online]. Available: <https://junit.org/junit5/docs/current/user-guide/>. [Accessed: Jul. 17, 2024].
* Mockito. [Online]. Available: <https://javadoc.io/doc/org.mockito/mockito-core/latest/org/mockito/Mockito.html>. [Accessed: Jul. 17, 2024].
* Let’s Encrypt for HTTPS. [Online]. Available: <https://letsencrypt.org/docs/>. [Accessed: Jul. 17, 2024].
* JavaMail Guide. [Online]. Available: <https://javaee.github.io/javamail/>. [Accessed: Jul. 17, 2024].

### 12.1 Tools

* Draw.io - UMLs creation.
* Tomcat will compile JSP pages into servlets.
* Apache the web server handles the application.
* Servlet handles the lifecycle of the application.

## 13. Acronyms/Abbreviations

* FWRP - food waste reduction project
* JDBC - Java Database Connection between the application and the database
* JSP - Java Server Page is the main presentation technology for the application
* Tomcat - Web Server handles the lifecycle of the application
* Servlet - The Application Server handles requests and responses and the lifecycle
* ERD - Entity Relation Diagram between database tables
* MySQL - Open-sourced relational database
* Use Case Diagram - Overview of the user interaction with the specific operation
* RDBMS - Relational Database Management System
* DAO - Data Access Object
* DTO - Data Transform Object