Wasteless 2

Student:Cioban Dumitru-Darius

**Group:30431**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

The application provides the option of a user to manage their food waste, by tracking it using this application, by tracking how food is wasted, and providing notifications when food waste is imminent, giving the user the option to donate.

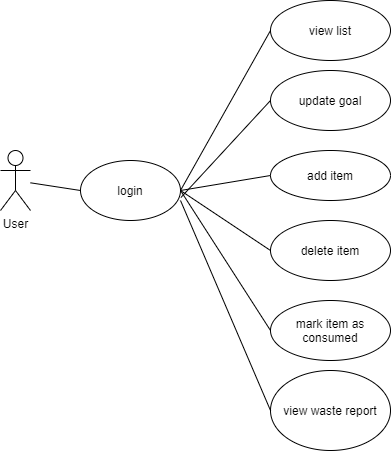
# Functional Requirements

*The functional requirements are adding and viewing the grocery list, viewing reports regarding the food waste, and notifications providing the donation option.*

# Non-functional Requirements

*The non-functional requirements are using a client-server architecture, using an observer pattern and a database to store the data.*

2. Use-Case Model

**

*Use case: add item*

*Level: user-goal level*

*Primary actor: user*

*Main success scenario: The user inputs the data of the new grocery item they want to add. It passes through the validator, and if the data is valid, a new item is added to their list.*

*Extensions: The validator returns an invalid value, signifying that the data is not according to the standard. An error message is displayed, giving info on how the data should look like. Another possibility is the case where the item wasn’t successfully added to the database, in which case the update won’t take place and nothing will be changed.*

3. System Architectural Design

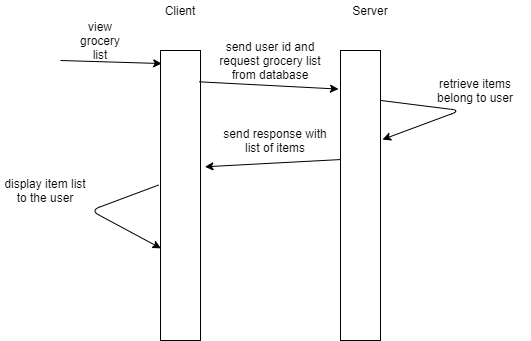
**3.1 Architectural Pattern Description**

*The architectural pattern used is the client-server architecture. In this architecture, the application is divided in two, the server side and the client side. The server side usually manages the data (like a database if it exists). The client side is what the user uses. They communicate through requests and responses and by establishing a connection.*

**3.2 Diagrams**

*[Create the system’s conceptual architecture; use architectural patterns and describe how they are applied. Create package, component and deployment diagrams]*

4. UML Sequence Diagrams

**

5. Class Design

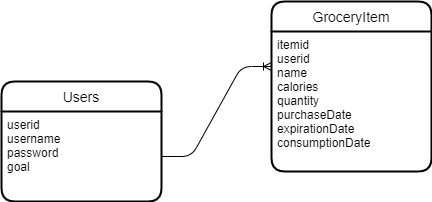
**5.1 Design Patterns Description**

*The used designs pattern is the Observer pattern. The idea of this pattern is that one objects may be observers or observable objects. When an observable object does something that needs sent to the observer, it will notify all of its observers. This design pattern is mainly used in event-driven architectures. The classes for an Abstract Factory pattern also exist, though they are not used in this version.*

**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

6. Data Model

**

*The data is represented by 2 tables. One for the users, where their id is the primary key. Username and password are used for authentication and goal for functionality. The grocery item table has an item id as primary key, an user id (establishes a one to many relationship between users and items), a name, calorie count, quantity and 3 dates representing the purchase date, the expiration date and the consumption date.*

7. System Testing

*The testing was done by feeding the system both valid and invalid data, seeing how it reacts. However, there are some scenarios which haven’t been tested or have been tested but haven’t been updated (for example the need of inputing only numbers in number fields).*

8. Bibliography