FoodWaste

Student: Craciun Sergiu

**Group:30442**

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

Design and implement an application that helps users manage food waste.

Once a user is authenticated he can input grocery lists and see reports of how much food is wasted weekly and monthly. A grocery list item has a name and a quantity as well as a calorie value, purchase date, expiration date and consumption date.

The system also allows users to track goals and minimize waste by sending reminders if waste levels are too high based on ideal burndown rates.

The ideal burndown rate for 100 calories worth of groceries due to expire in 5 days is 20 calories worth of groceries per day.

The system should provide you with options to donate excess food to various local food charities and soup kitchens and notify you of them prior to item expiration.

# Functional Requirements

The application should use authentication. In order to access the application, the user needs to provide valid credentials, username and password.

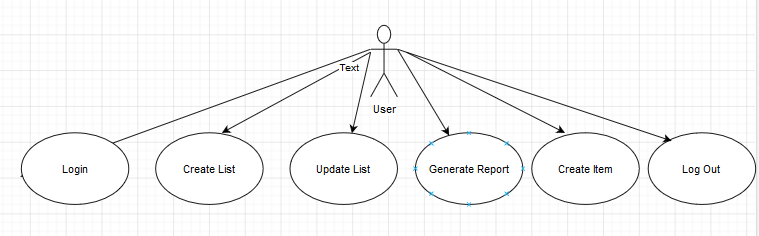
Once authenticated, the user can perform the following operations:

1. Perform CRUD on grocery lists (view, create, delete, update)
2. Perform CRUD on specific items inside a grocery list
3. Generate reports (monthly and weekly reports)
4. Donate excess food to charitable organizations
5. Set a goal and get notified whenever the number of calories is exceeded.

# Non-functional Requirements

* To use an ORM and a DI Container
* To use any OOP language you like
* To use a layered architecture
* To use the abstract factory pattern for creating weekly/monthly reports
* The data will be stored in a database

2. Use-Case Model



Use case: create new account

Level: user goal

Primary actor: User

Main success scenario: The User that wishes to create a new account needs to enter the login page and click on the “Create new user”. A new page will appear and the user will need to input the desired username and password, after which they will need to save the information.

Extensions: a fail case for this would be that the user could input an already used username, in which case they will be prompted to change the password

3. System Architectural Design

**3.1 Architectural Pattern Description**

For this project I have used a layered architecture , it having the following packages: applicationseed for creating users, configuration for configuring the application, controllers to ensure the functions of the app are working, generators for generating the reports, models for all the entities in the app, repositories for storing the reports and services that make use of the repository packages.

**3.2 Diagrams**

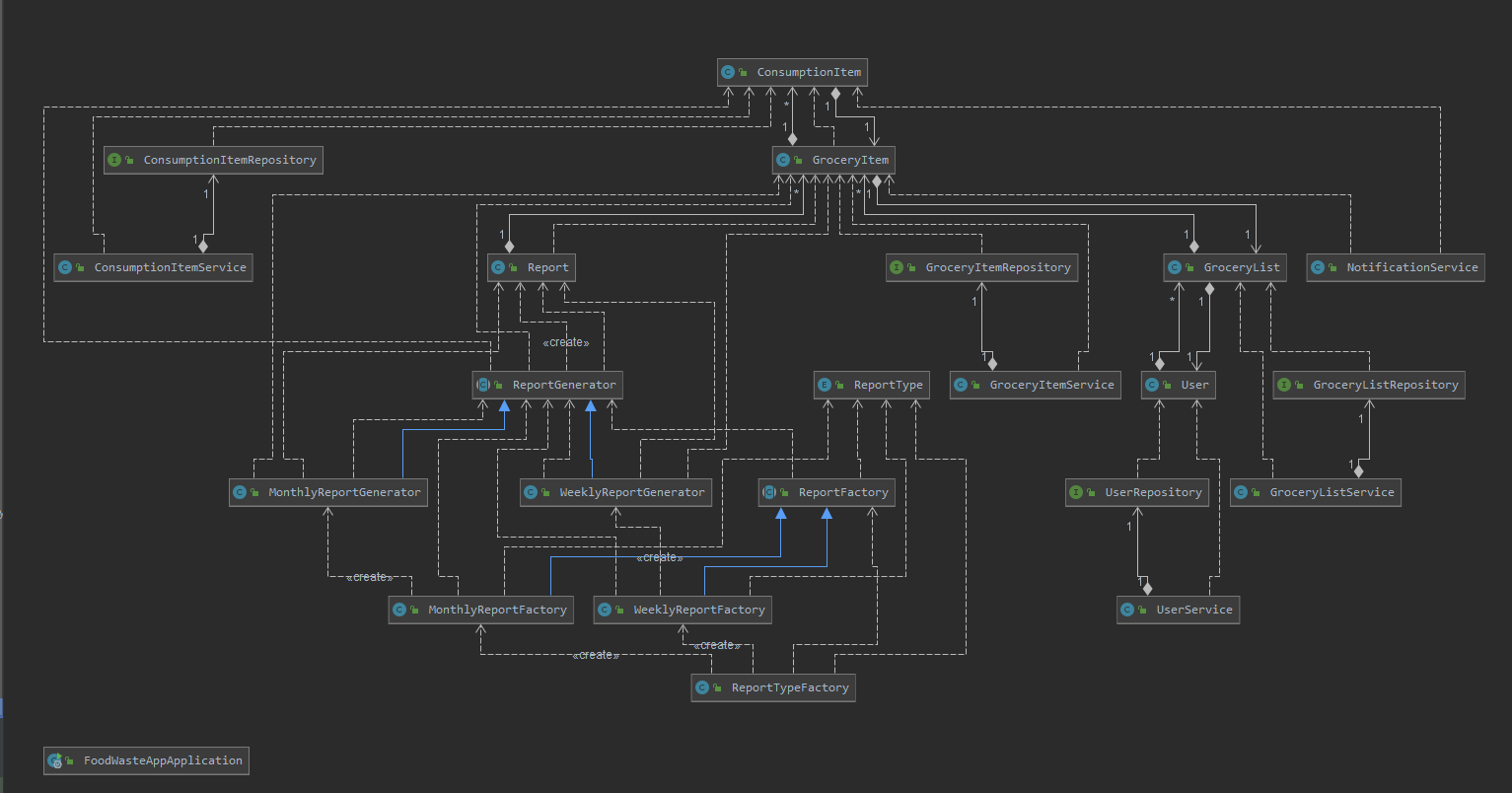
4. UML Sequence Diagrams

5. Class Design

**5.1 Design Patterns Description**

The design pattern used in this assignment is the Abstract Factory Pattern and the Observer. The first is basically a factory of factories. This was used to create two different kinds of reports, a weekly report and a monthly report of wasted food, as well as eaten food. Since the requirements only specify two types of reports, there are two Factories, one that generates weekly reports, and another that generates monthly reports. Both these factories make use of another class, ReportFactory that basically decides which type of report to be implemented. There is also an enum that only contains the two types of reports that can be implemented, weekly or monthly.

**5.2 UML Class Diagram**



6. Data Model

7. System Testing

I created a simple check test which will fail if the application context will not start.

8. Bibliography

<https://start.spring.io/>

<https://spring.io/guides/gs/testing-web/>

stackoverflow

Wikipedia

Google