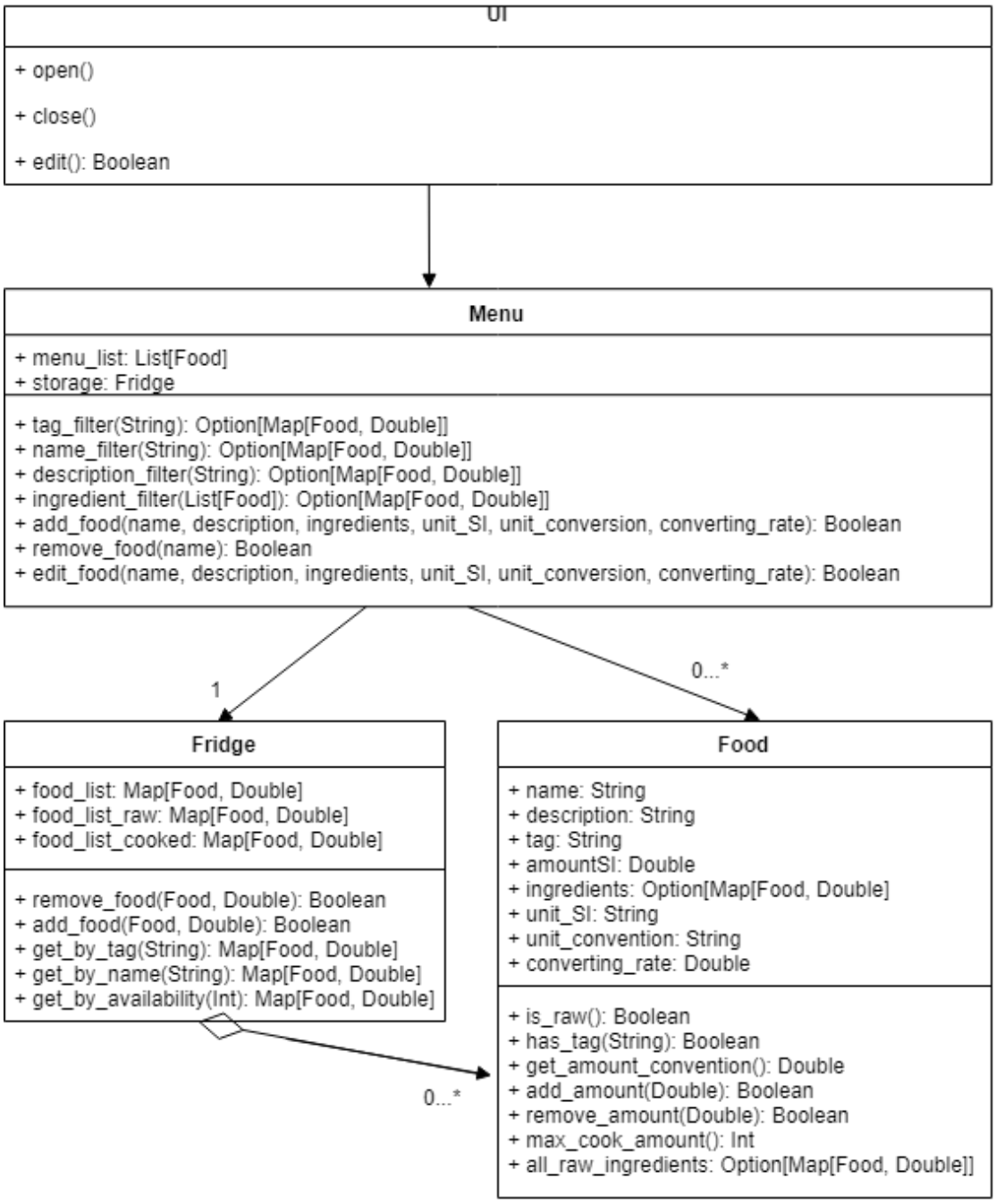
Personal information

Title: 232 Intelligent recipe book
Name: Taige Wang
Student number: 713672
Program: Aalto Bachelor's Programme in Science and Technology
Year of studies: 1st year
Data: 11th Feb, 2019

1. Class structure

The structure of my program is going to have a similar structure as the following diagram. Under the user interface, the Menu class in charge of various functions in the user interface, such as filtering, searching, editing/deleting/adding a food. The Fridge class provides the storage information, including classifying food to categories, provide methods for Menu class. Finally, the food class provides relevant information on foods, including allergenic information, amount, units and so on. This class also find the maximum amount available for cooking, so for each individual Food the amount can be tested in this way.



2. Use case description

The user opens the program. If the user has saved the file previously, the program opens the file from default path. Then the program shows a simplified food list based on the availability of foods. For this part, some methods in Menu are used. After user searching the keyword or placing selections on filtering criteria, all the methods combine together and perform the outcome on the user interface, the rest of the methods in Menu class get called at the same time. If one ingredient or more is missing, the list will be ranked based on availability so a user can either accept a menu, change a menu or quit.

3. Algorithms

The algorithm used in this program is quite simple, the availabilities of foods are checked individually. The program checks the available ingredients, if some of them are missing, the program tries to make those from raw ingredients if possible, or return the maximum value if failed. To avoid repetitive use of ingredients, availabilities are checked by performing calculations without using the availabilities of some other ingredients.

4. Data structures

Most of the data used in this program will be performed as Maps because the amount of food is closely related to food itself. The stored file may store in .csv, JSON or XML format depends on the simplicity.

5. Schedule

Demo	18.2	-	22.2
Algorithm Phase 1	23.2	-	6.3
Improvements / Debugging	7.3	-	10.3
Algorithm Phase 2	11.3	-	20.3
Improvements / Debugging	21.3	-	25.3
UI & IO	26.3	-	10.4
Improvements / Debugging	11.4	-	15.4
Debugging	16.4	-	18.4
Validating documentation	19.4	-	22.4

6. Unit testing plan

The core of unit testing in my project is the algorithm for the maximum available amount of food. The recursively defined menu is a huge problem, therefore, it should be avoided by providing tests for those. The unit test tries to provide several valid scenarios and check the validity of the algorithm. Tests will focus on the repetitive usage of ingredients and the amount of food.

7. References and links

Scala documents, Scala APIs, support for external libraries (e.g. CSV reader/writer), UI Libraries, Image processing libraries

8. Appendixes

/