DIET MANAGER/ RELEASE

Diet Manager Design Document

Group1

Matija Selak <mxs4350@g.rit.edu>

Ivan Susnjara <ixs1924@g.rit.edu>

Leon Grubišić <lxg6892[@g.rit.edu](mailto:jtoa@rit.edu)>

Boris Fjorović <bxf4579@g.rit.edu>

Lehady Sani-Agatha <lxs4287@g.rit.edu>

# Project Summary

This project will consist of several stages planning, designing, development, and implementation. The planning stage main goal is to develop the Diet Manager application following the specifications given to us in my courses. Diet manager will be used to keep track and calculate the food intake by the user. The project development team consists of 5 members that will work closely hand in hand. The team will be using GitHub for better coordination of development stage. In the design, the stage team will create UML sequence and class diagrams. Discuss class names and applications overall design. Development stage will consist of solidifying the ideas into code. The team members will use the design principles and patterns that they all agreed upon. In the implementation stage, the team will be testing, modify and refactoring the application.

# Design Overview

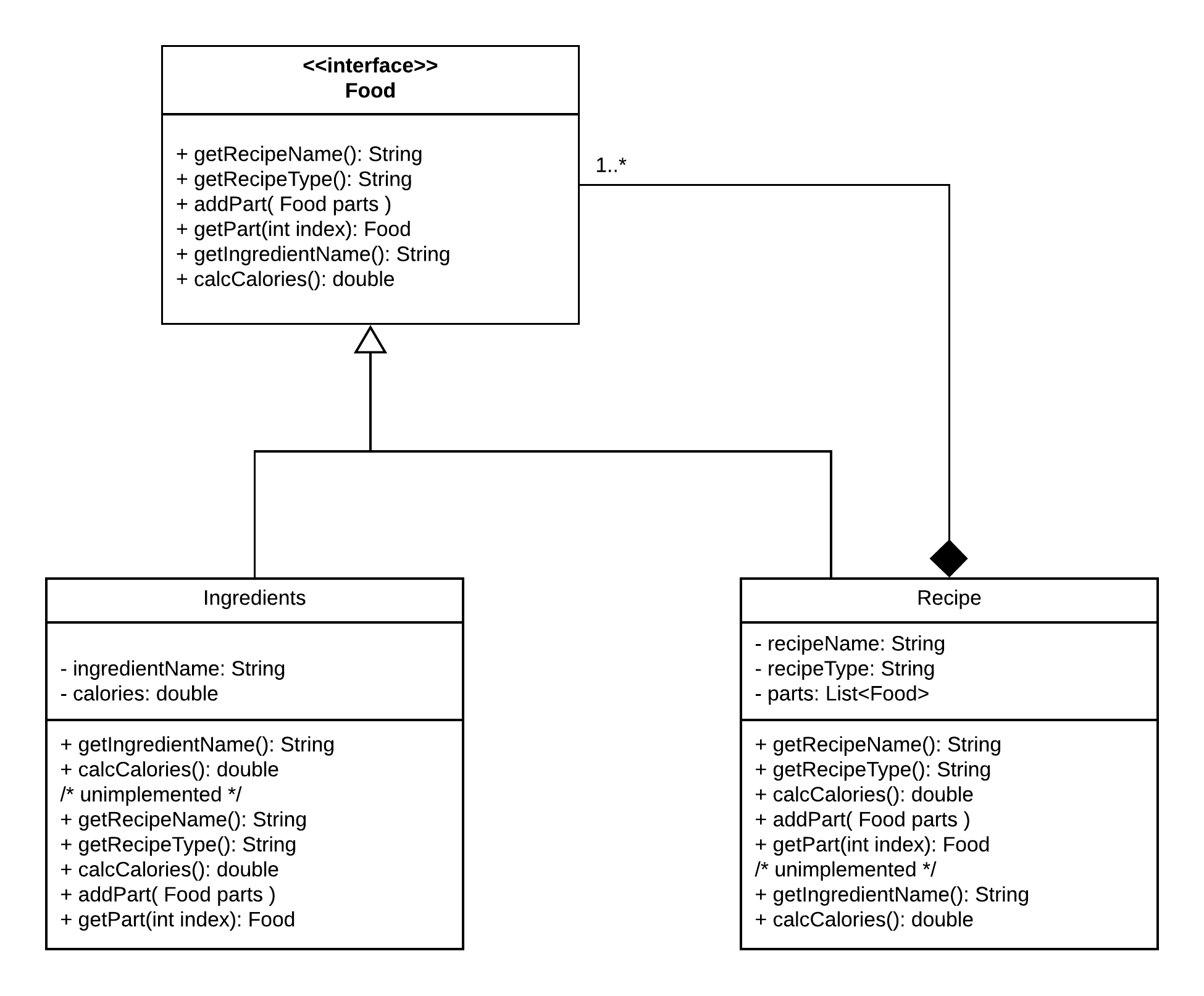
We divided our application into several modules and add an abstract layer at the top of each module. Such as food interface that holds methods which are used in both food and ingredient class. For this project, we will depend on the abstraction layer. To make our application open for future extensions. We will hold duplicate functionalities in our composite pattern to make it accessible through the whole application. This will make our modification a lot easier. We will give each class, method, and module a single responsibility for each of them in order to minimize regressions. Each module knows what another module does, but it will not know its functionalities.

# Subsystem Structure

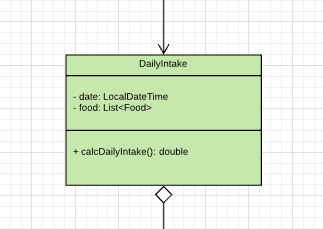
# 

# Subsystems

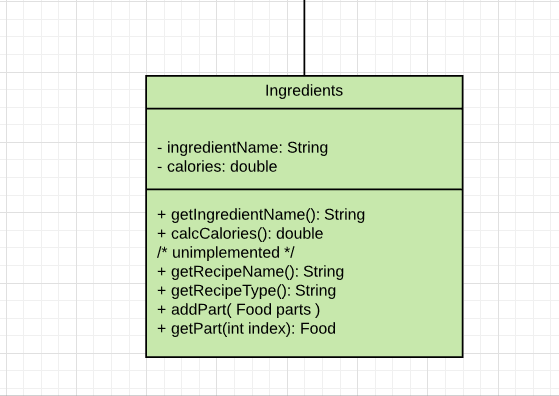
**Subsystem: Food**



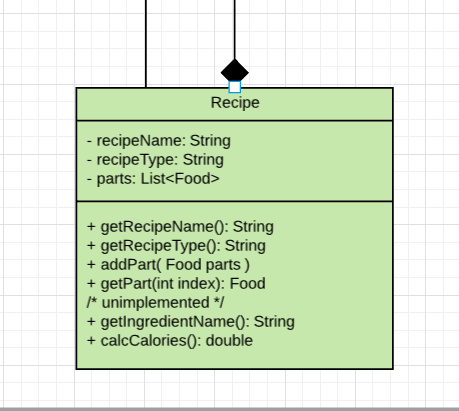
|  |  |
| --- | --- |
| **Interface** Food (interface) | |
| **Responsibilities** | Provides a generic interface to Ingredients and Recipes.  Can be ingredients or recipe. Hold all necessary methods that Ingredients and Recipe needs to implement. Defines their behaviour. |
| **Collaborators**  **(implements)** | Ingredients - will use getIngredientName() and calcCalories() methods.  Recipe - will use getRecipeName(), getRecipeType(), calcCalories(), addPart( )  and getPart() methods. |

****

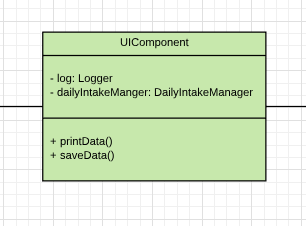
|  |  |
| --- | --- |
| **Class** DailyIntake | |
| **Responsibilities** | Keeps track of calories intake during daily period. Will also hold the list of all the foods consumed. |
| **Collaborators**  **(uses)** | Food - will calculate its calories and store food names. |



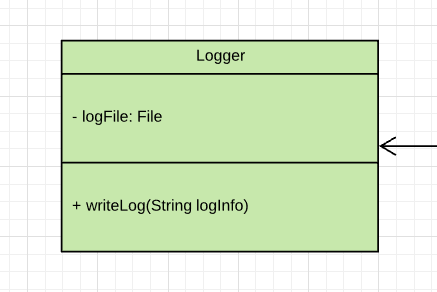
|  |  |
| --- | --- |
| **Interface** Ingredients | |
| **Responsibilities** | Will hold info about Ingredients. Each ingrediente instance will also have its calories property. |
| **Collaborators**  **(implements)** | Food - |

****

|  |  |
| --- | --- |
| **Class** Recipe | |
| **Responsibilities** | Will hold info about Recipes. It will also use List of Foods that will hold all the different ingredient parts. |
| **Collaborators**  **(implements)** | Food interface - |



|  |  |
| --- | --- |
| **Class** UIComponent | |
| **Responsibilities** | Is responsible for all the visual interface that users will use to interact with the application. |
| **Collaborators**  **(uses)** | DailyIntake  Logger  FoodReportManager |



|  |  |
| --- | --- |
| **Class** Logger | |
| **Responsibilities** | Will log all the activities that application makes into a file. |
| **Collaborators**  **(uses)** | UIComponent |

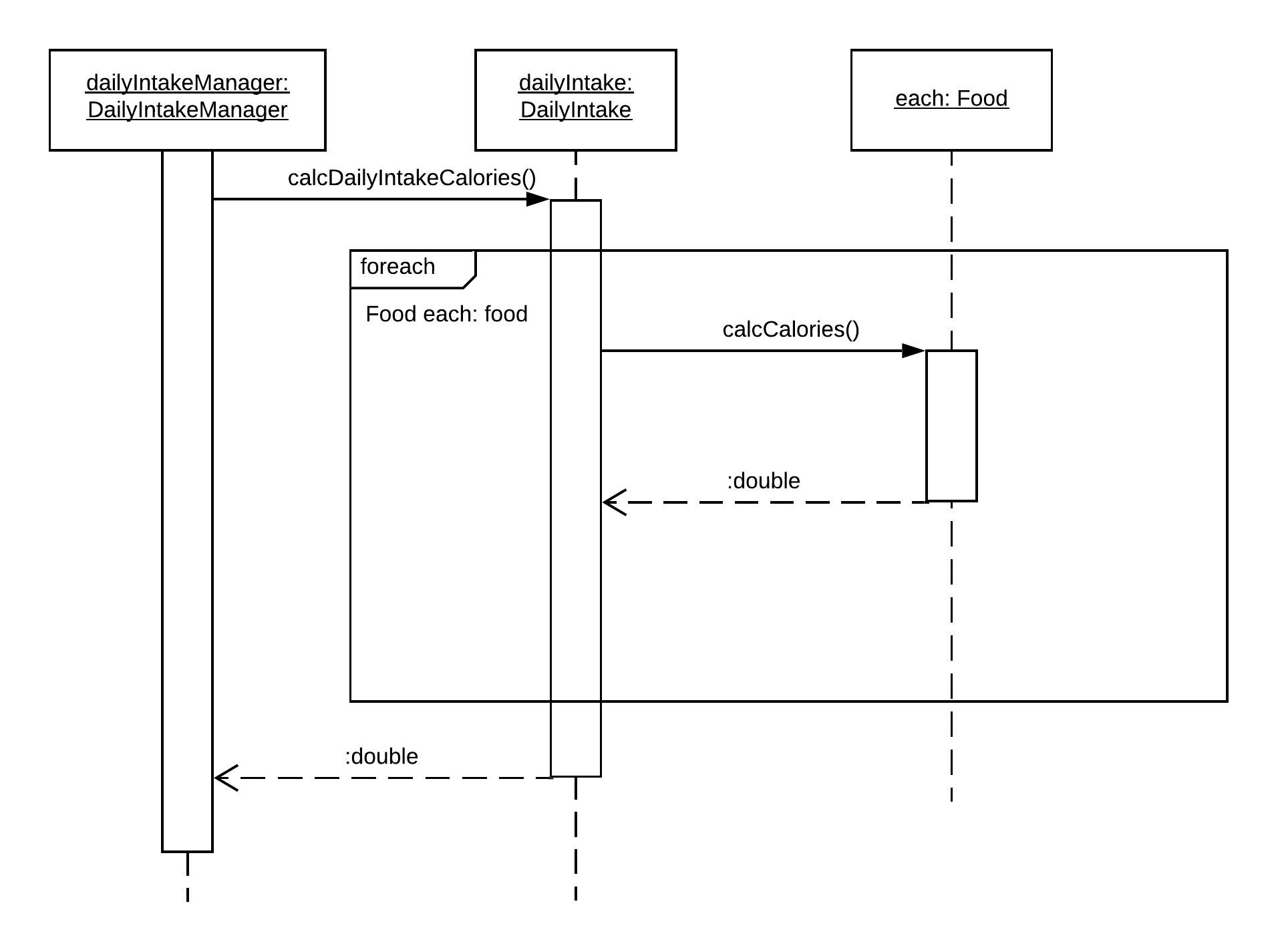
# 

|  |  |
| --- | --- |
| **Class** DailyIntakeManager | |
| **Responsibilities** | This class is responsible for intake calculation on given time and date pe |
| **Collaborators**  **(uses)** | UIComponent |

# Sequence Diagrams

## **Sequence diagram 1.**

This diagram shows calculation of calories within the daily intake



## **Sequence diagram 2**

## This diagram shows the calculation of total intake calories in a day.

## 

## **Pattern Usage**

Currently in this phase of development we are using one pattern and it is composite pattern.

Pattern #1 Composite pattern

|  |  |
| --- | --- |
| **Composite pattern** | |
| **Leaf(s)** | Ingredients |
| **<interface> Component** | Food |
| **Component** | Recipe |

