Documentation for Meal Planner Programming Project

Name: Jonathan Setiawan

Student ID: 26002404663

Project Purpose and Functionality

The purpose of this project is to build a useful logger to keep track of the nutrient intake for someone who wants to either go on a diet, build muscle, or fulfill any of their body goals. Initially, this project was made due to my current interest in keeping track of the amount of calories and protein I take on a daily basis as I have been going to the Ritsumeikan gym for the past couple of months so this can be beneficial for students who would like to do the same as I plan to which is to keep track of my nutrition to maximize my progress and recovery.

Explanation of Each Class and Method

1. **INTERFACE**
    a. *Saveable*: An interface that simulates a saveable characteristic and has the methods below:
        i. *saveToFile(String fileName)*: Saves the data to a specified file

2. **ABSTRACT CLASS**
    a. *Logger*: An abstract class that provides the most basic or fundamental foundations for logging and implements the Saveable interface, the functions include:
        i. Fields
            1. *Map<String, StringBuilder> logs*: A TreeMap that stores logs. Each key represents a date, and the corresponding value is a StringBuilder containing the log entries for that date.
        ii. Constructor
            1. *Logger()*: Initializes the “logs,” map
        iii. Methods
            1. *abstract void log(String key, String value)*: Abstract method that requires subclasses to implement logging logic.
            2. *String getAllLogs()*: Returns all the logs as a formatted string.

3. *saveToFile(String fileName)*: Saves all logs to the specified file by writing the key and value pairs from the logs map.

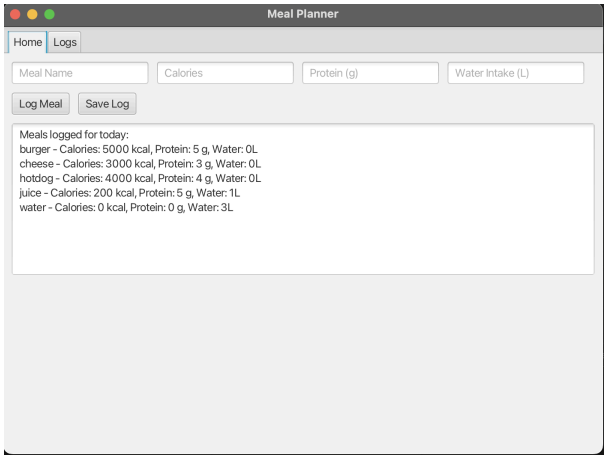
3. <u>CONCRETE CLASSES</u>

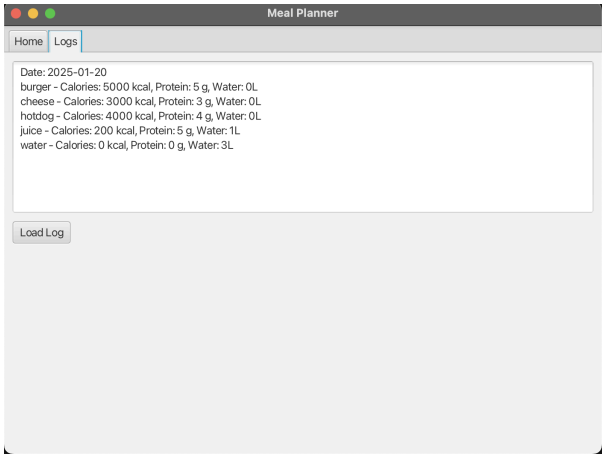
a. *DailyLogger*: A class that extends Logger and implements its abstract methods.
   i. Methods:
      1. *log(String date, String value)*: Adds a log entry under the specified date. If there are no entries for the date then it creates a new entry.
      2. *saveToFile(String fileName)*: Writes the logs data to the specified file using the logic provided in the Logger class.

<u>GUI (No Main.java file since the testing is already included and considering it's based on particular user's input)</u>

b. *MealPlannerGUI*: A class for the graphical user interface of meal planner
   i. Fields:
      1. *TextField mealNameField*: Input text field for meal name.
      2. *TextField calorieField*: Input text field for calories.
      3. *TextField proteinField*: Input text field for protein.
      4. *TextField waterIntakeField*: Input text field for water intake.
      5. *TextArea logArea*: Displays logged meals.
      6. *TextArea summaryArea*: Displays a summary of meals for the current day.
      7. *DailyLogger dailyLogger*: An instance of DailyLogger to manage logs.
   ii. Methods:
      1. *void logMeal()*: Logs a meal's details under the current date using dailyLogger.
      2. *void displayDailyLogs()*: Updates the logArea with all logs from dailyLogger.
      3. *void saveLogToFile()*: Saves the dailyLogger logs to a file named meals.txt.

Example Output:





Future Plans for Enhancement

Initially, I designed my project in two parts, the first is to get a rough visual idea of what I want to develop and the second is a dream design concept. Due to the fact that I was not able to develop the rough visual idea that I had sketched out initially due to complexity issues, we could try implementing a design that tackles similar features or implement the dream design concept itself. Fig. 1 shows the initial sketch of what I wanted to develop which could be the first part of a future plan for enhancement and Fig. 2 shows the dream design concept that I initially wanted to pursue as the ultimate goal of this project, which is to inevitably create a working application. Fig. 3 shows the initial diagram I made for the project before I incorporated any GUI into it which subsequently caused me to have to change the entire layout into what is seen in Fig. 4 due to the nature of GUI in java.

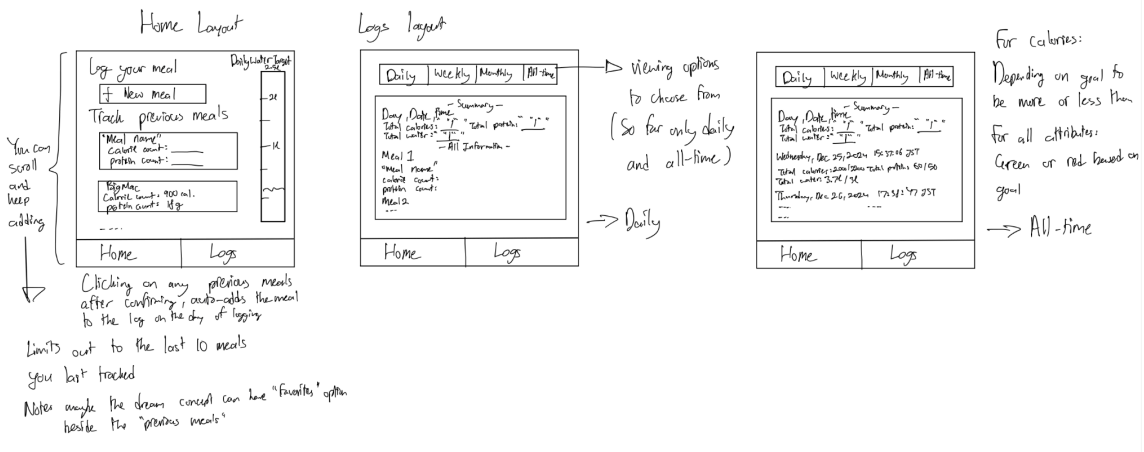


Figure 1. Initial Draft of Project Concept Design

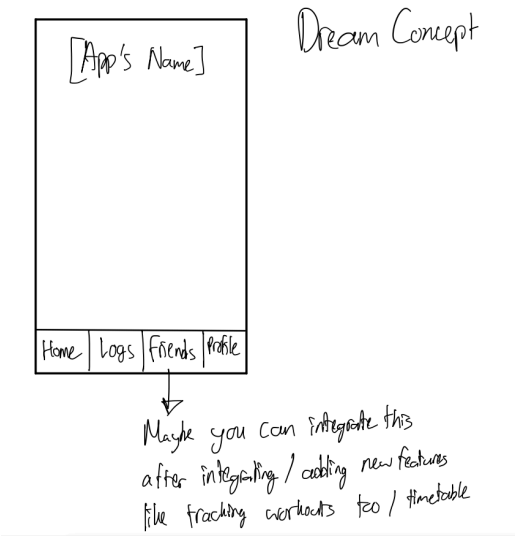


Figure 2. Rough Sketch of Dream Concept Design

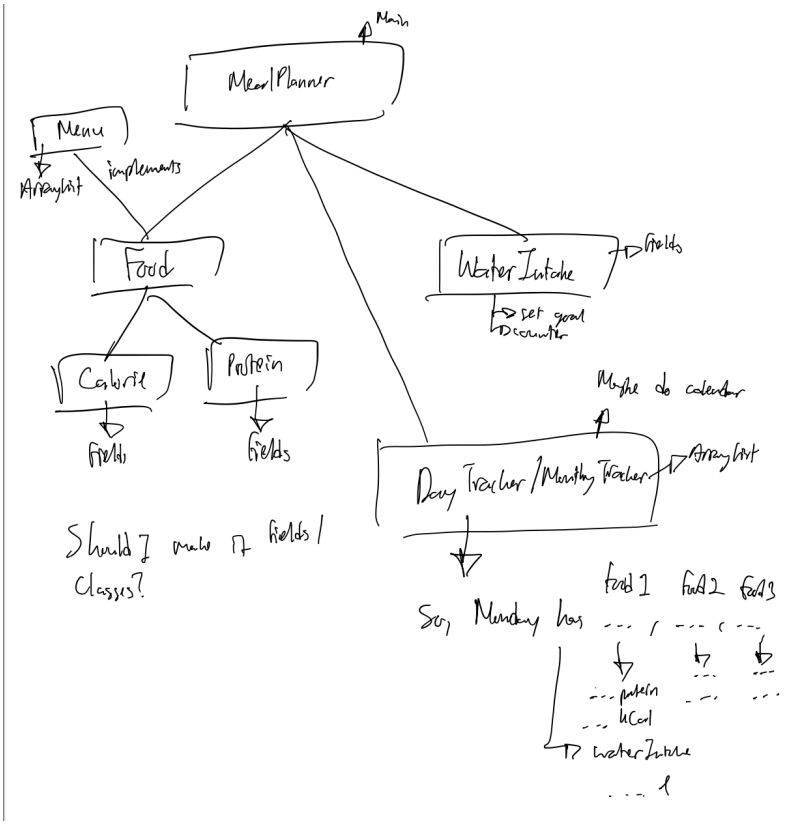


Figure 3. First Design Conceptualization Diagram

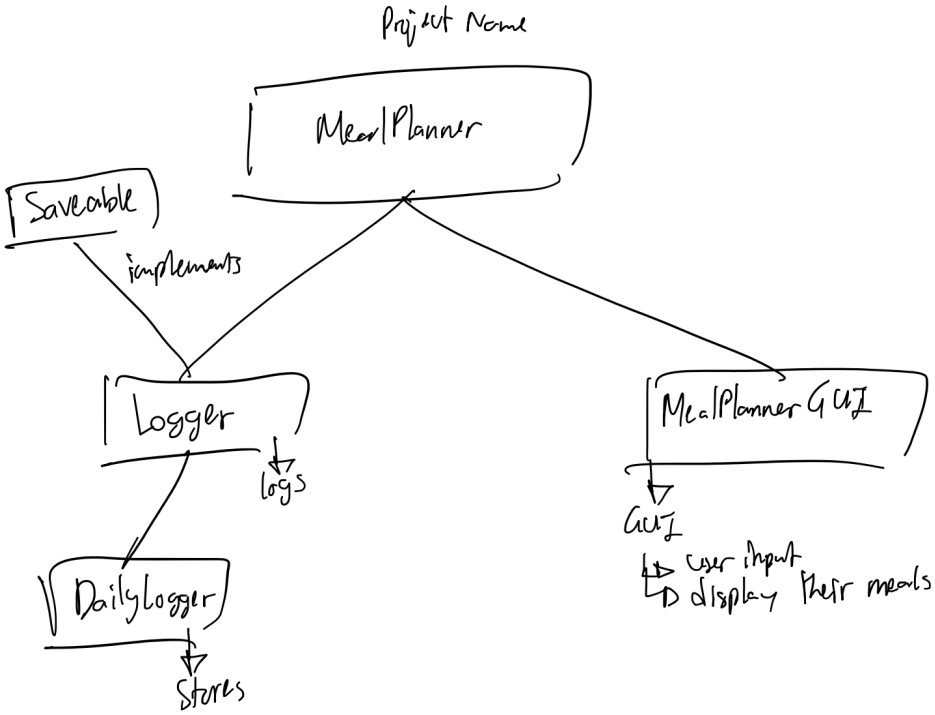


Figure 4. Final Design Conceptualization Diagram