PROJECT REPORT
ON “ DIET PLANNER ” JAVA PROJECT

BY:

1. **SHADEV KHAREL, (H10006790)**
   sk61242@uowmail.edu.au
   Coding the functions + Database and Error Handling

2. **IP TSUN TING, (H10006777)**
   tti15515@uowmail.edu.au
   File I/O+ User Interface Management+ Sourcing

Handed To: Mr. Ka Man Pang

(AST10106, Tutor)

CONTENTS OF THIS REPORT

Topic: Page:

A. Objective of the Project

The Project “Nutritional Planner” came up as an extraordinary high-scope application-based project idea for our Java project, that we had to demonstrate for Semester A. The primary objective of this project design is to produce an accurate plan for food diet for an individual, according to their preferences and personalization. By relying on calculations through various mathematical formulas and statistically analyzing the required calorie intake for an individual, a balanced and healthy suggestion of diet was our priority. Not only does the project serves this primary purpose, but there are also numerous ranges of other application, including “Workout Planner”, Personal Health Tips, Info Tracker, and many more. In addition to this, the project encompasses a multiple range of data handling, which makes it possible for the user to retrieve their past credentials and data effectively. The database for diets, and workouts, is designed in a way for much larger scalability in the future. The efficiency of the user’s interaction was strictly taken into account. The UI, too, was prioritized, amid the limited text-based resources offered by the Java JDK. The user profile is also secured throughout the run, and the option to delete their credentials is left to the user, for security purposes. All in all, the project is designed in an organized manner, focusing on the accuracy of the information being catered to the user, as the user would want to focus solely on his/her weight loss/gain journey.

- Kharel Shadev,

- Ip Tsun Ting

B. Algorithm of the Project.

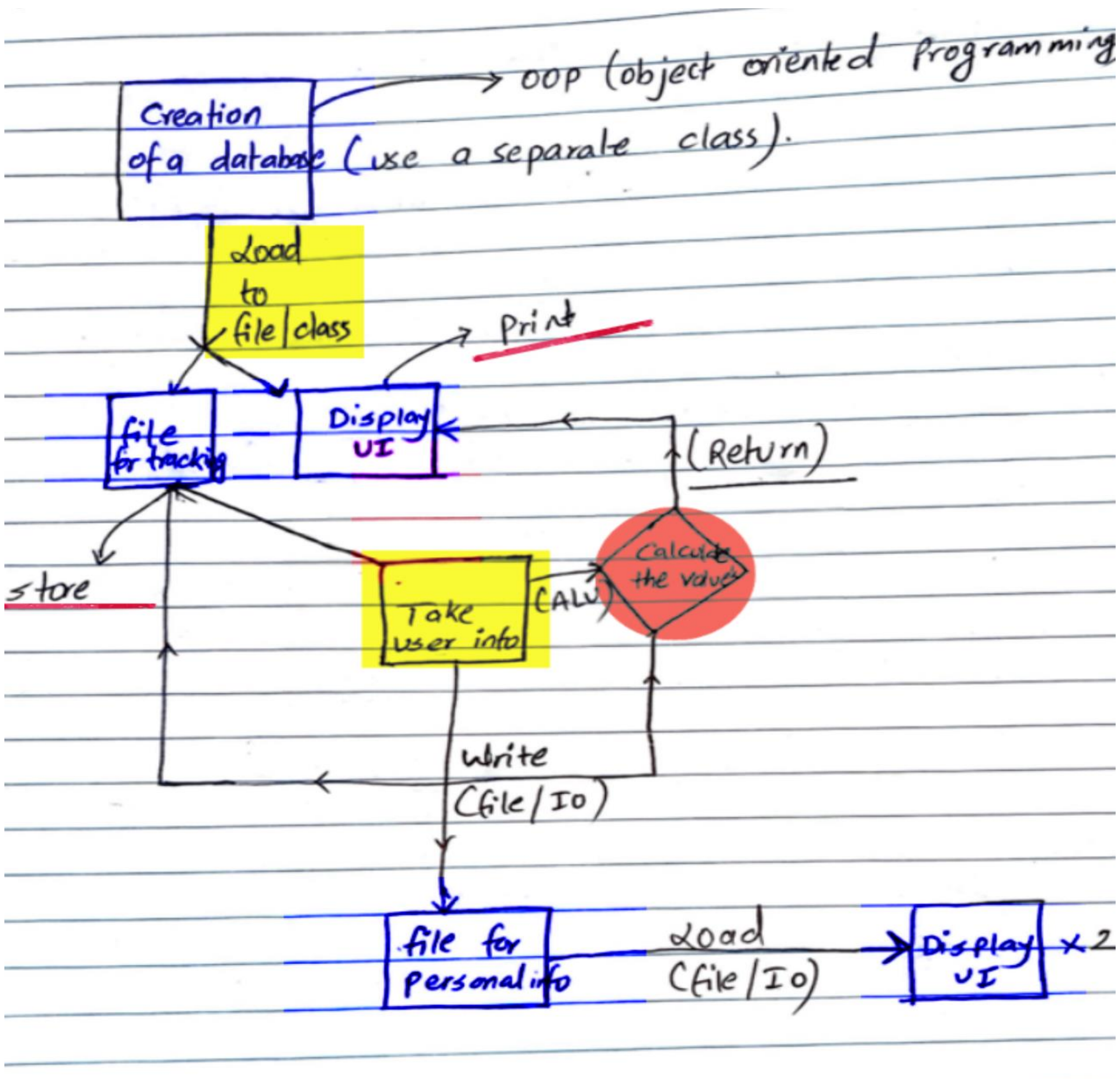
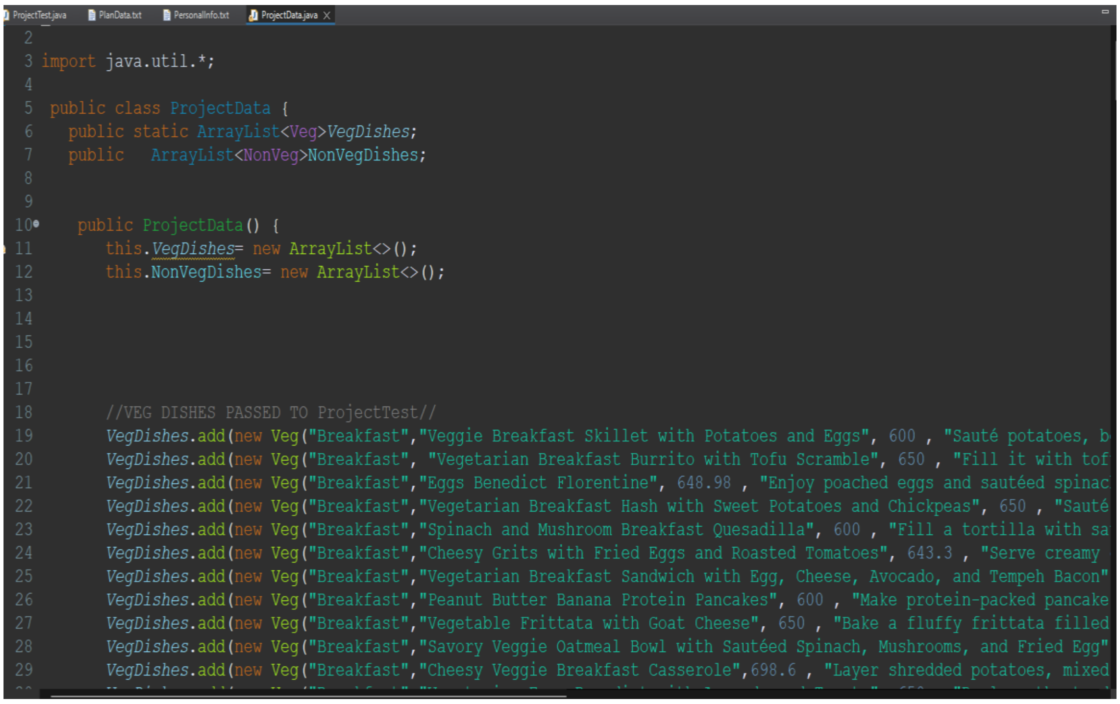


fig: Algorithmic flowchart of the code

C. Database of the Project.

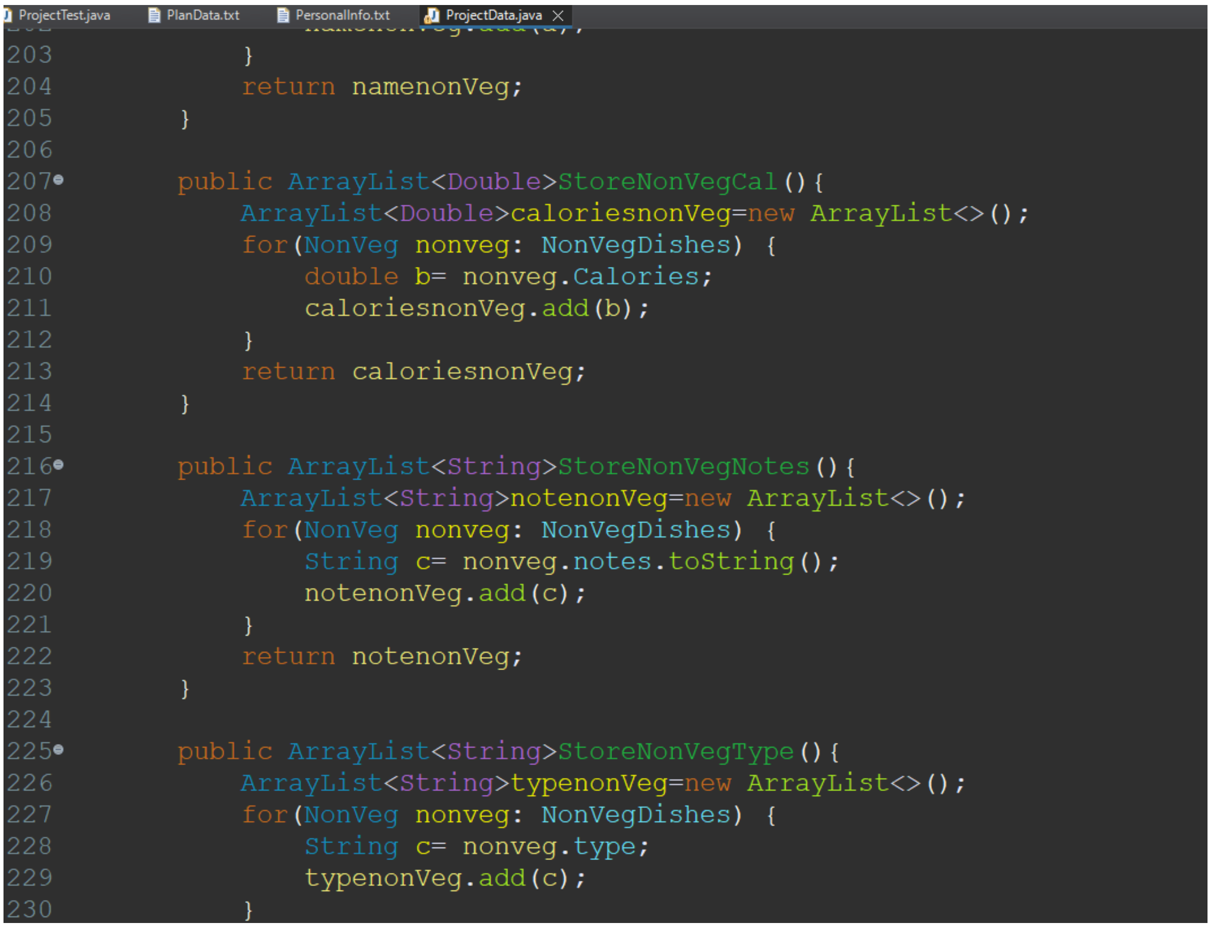
Our Programming project as mentioned before, applies the concept of Object-Oriented Programming(OOP) to handle the database. In the project file, there is a class named “ProjectData” which has the required database of diets and workouts, and it is imported into the main class file for operations and display purposes.



```java

import java.util.*;

public class ProjectData {
  public static ArrayList<Veg>VegDishes;
  public   ArrayList<NonVeg>NonVegDishes;


   public ProjectData() {
      this.VegDishes= new ArrayList<>();
      this.NonVegDishes= new ArrayList<>();





      //VEG DISHES PASSED TO ProjectTest//
      VegDishes.add(new Veg("Breakfast","Veggie Breakfast Skillet with Potatoes and Eggs", 600 , "Sauté potatoes, b
      VegDishes.add(new Veg("Breakfast", "Vegetarian Breakfast Burrito with Tofu Scramble", 650 , "Fill it with tof
      VegDishes.add(new Veg("Breakfast","Eggs Benedict Florentine", 648.98 , "Enjoy poached eggs and sautéed spinac
      VegDishes.add(new Veg("Breakfast","Vegetarian Breakfast Hash with Sweet Potatoes and Chickpeas", 650 , "Sauté
      VegDishes.add(new Veg("Breakfast","Spinach and Mushroom Breakfast Quesadilla", 600 , "Fill a tortilla with sa
      VegDishes.add(new Veg("Breakfast","Cheesy Grits with Fried Eggs and Roasted Tomatoes", 643.3 , "Serve creamy
      VegDishes.add(new Veg("Breakfast","Vegetarian Breakfast Sandwich with Egg, Cheese, Avocado, and Tempeh Bacon"
      VegDishes.add(new Veg("Breakfast","Peanut Butter Banana Protein Pancakes", 600 , "Make protein-packed pancake
      VegDishes.add(new Veg("Breakfast","Vegetable Frittata with Goat Cheese", 650 , "Bake a fluffy frittata filled
      VegDishes.add(new Veg("Breakfast","Savory Veggie Oatmeal Bowl with Sautéed Spinach, Mushrooms, and Fried Egg"
      VegDishes.add(new Veg("Breakfast","Cheesy Veggie Breakfast Casserole",698.6 , "Layer shredded potatoes, mixed
```

(Fig. ProjectData class file)

Initially, there is a lead class named “ProjectData” which has two objects constructed named “VegDishes” and “NonVegDishes”. Each of them takes a sub-class Veg and NonVeg which has various objects necessary to specify what sort of diet it is. We tried to encapsulate all the diet's names, the time of meals i.e Breakfast, Lunch, Snacks, etc, Name of the dish, Calories, and important information about the diet, inside the same class. The ProjectData class file also has various methods to return the name, calories, notes, and workouts efficiently. The name of the diets and the calorie per serving was generated and collected through various online databases. While it is true that per serving calorie is a subjective quantity, we tried our best to extract the average calorie for that diet.



ProjectTest.java PlanData.txt PersonalInfo.txt ProjectData.java

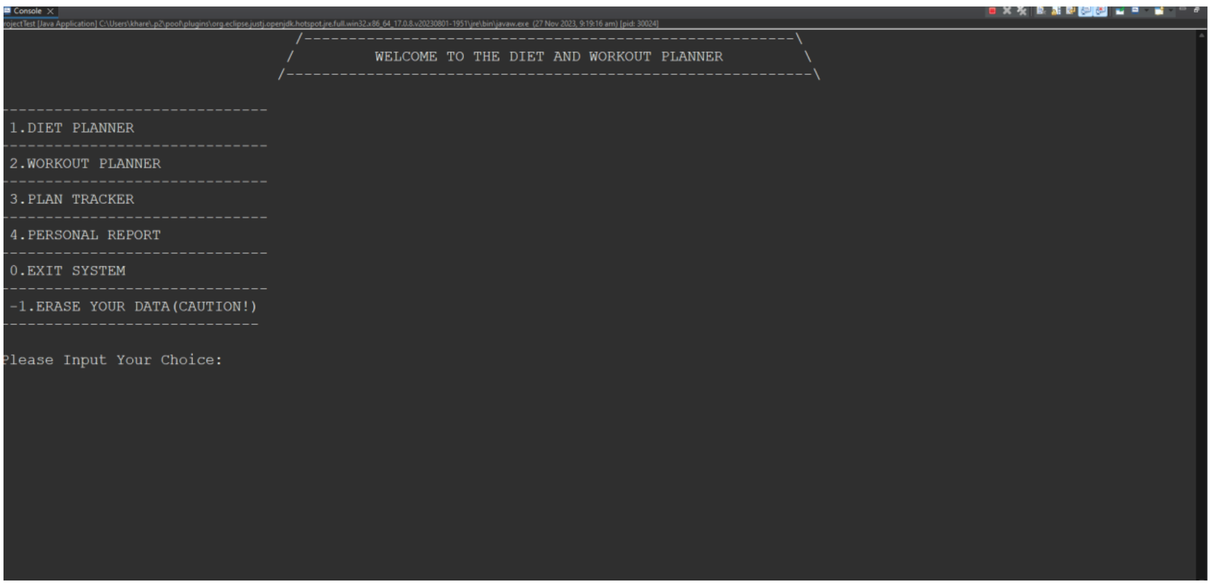
```
            }
            return namenonVeg;
        }

        public ArrayList<Double>StoreNonVegCal(){
            ArrayList<Double>caloriesnonVeg=new ArrayList<>();
            for(NonVeg nonveg: NonVegDishes) {
                double b= nonveg.Calories;
                caloriesnonVeg.add(b);
            }
            return caloriesnonVeg;
        }

        public ArrayList<String>StoreNonVegNotes(){
            ArrayList<String>notenonVeg=new ArrayList<>();
            for(NonVeg nonveg: NonVegDishes) {
                String c= nonveg.notes.toString();
                notenonVeg.add(c);
            }
            return notenonVeg;
        }

        public ArrayList<String>StoreNonVegType(){
            ArrayList<String>typenonVeg=new ArrayList<>();
            for(NonVeg nonveg: NonVegDishes) {
                String c= nonveg.type;
                typenonVeg.add(c);
            }
```

(Fig: Return methods for various entities)

D. Explaining Diet Planner.

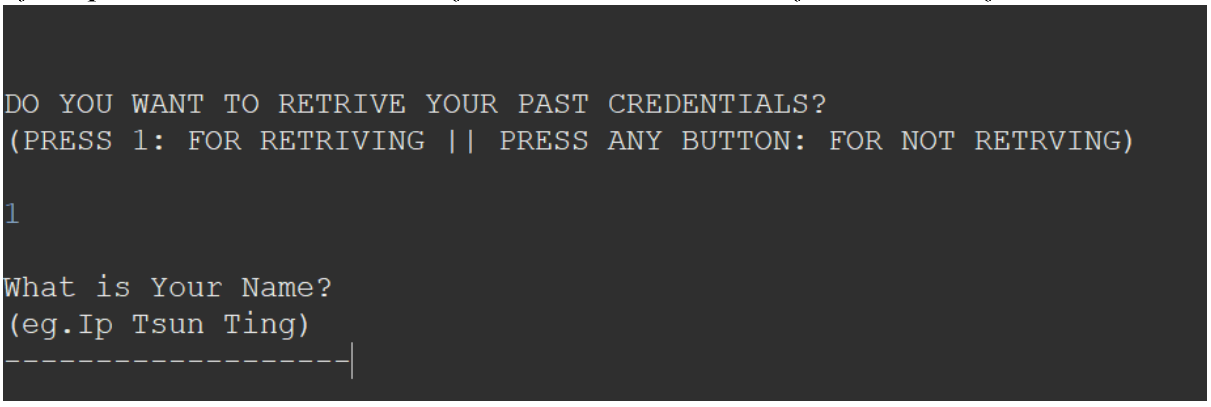
While we explored possible user-based applications that could have been catered to abiding by the concept of the diet planner, we did not/ or did not have enough access or understanding of the resources required for the undergoing those complex functions. For example: Access to Macronutrients and varieties of other factors like recipe came into our consideration but it would consume a lot of the memory space, and deeper database handling would have been required for it. Therefore, we simplified concepts like “Login Page”, “Diet Preferences”, and “Personalized Diet” to “Data retrieval page”, “Veg/Non-Veg option”, and “Modification section” respectively. This simplicity adds a whole new complexion to our project.

To begin with, in the given UI, when the user switches the option 1, the project displays the user information page required for calculations and stores it inside a file. **STEP 1:**



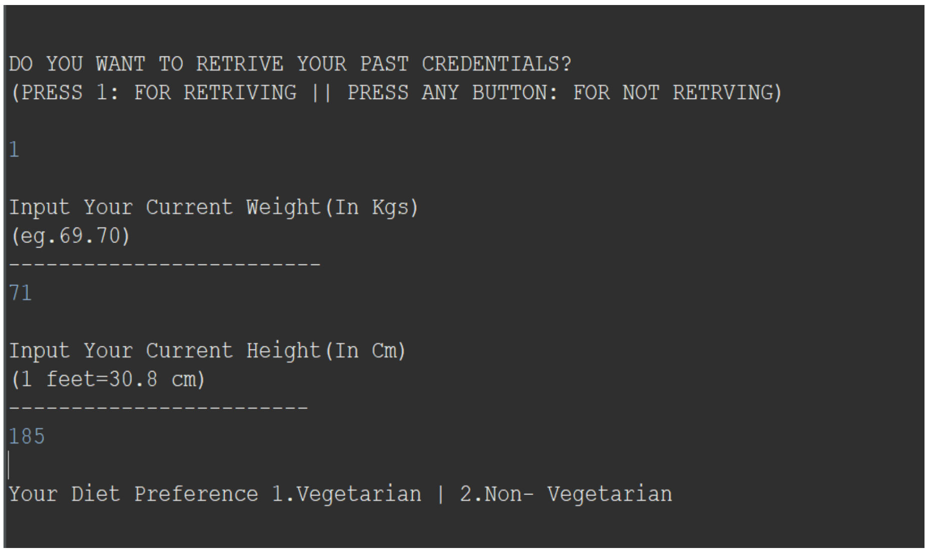
```
Console ✕
rojectTest [Java Application] C:\Users\khare\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230801-1951\jre\bin\javaw.exe (27 Nov 2023, 9:19:16 am) [pid: 30024]
                                  /-------------------------------------------------------\
                                 /          WELCOME TO THE DIET AND WORKOUT PLANNER        \
                                /-----------------------------------------------------------\

------------------------------
 1.DIET PLANNER
------------------------------
 2.WORKOUT PLANNER
------------------------------
 3.PLAN TRACKER
------------------------------
 4.PERSONAL REPORT
------------------------------
 0.EXIT SYSTEM
------------------------------
 -1.ERASE YOUR DATA(CAUTION!)
-----------------------------

Please Input Your Choice:
```

If no previous credentials are found, the user should fetch their information.



```
DO YOU WANT TO RETRIVE YOUR PAST CREDENTIALS?
(PRESS 1: FOR RETRIVING || PRESS ANY BUTTON: FOR NOT RETRVING)

1

What is Your Name?
(eg.Ip Tsun Ting)
-------------------
```

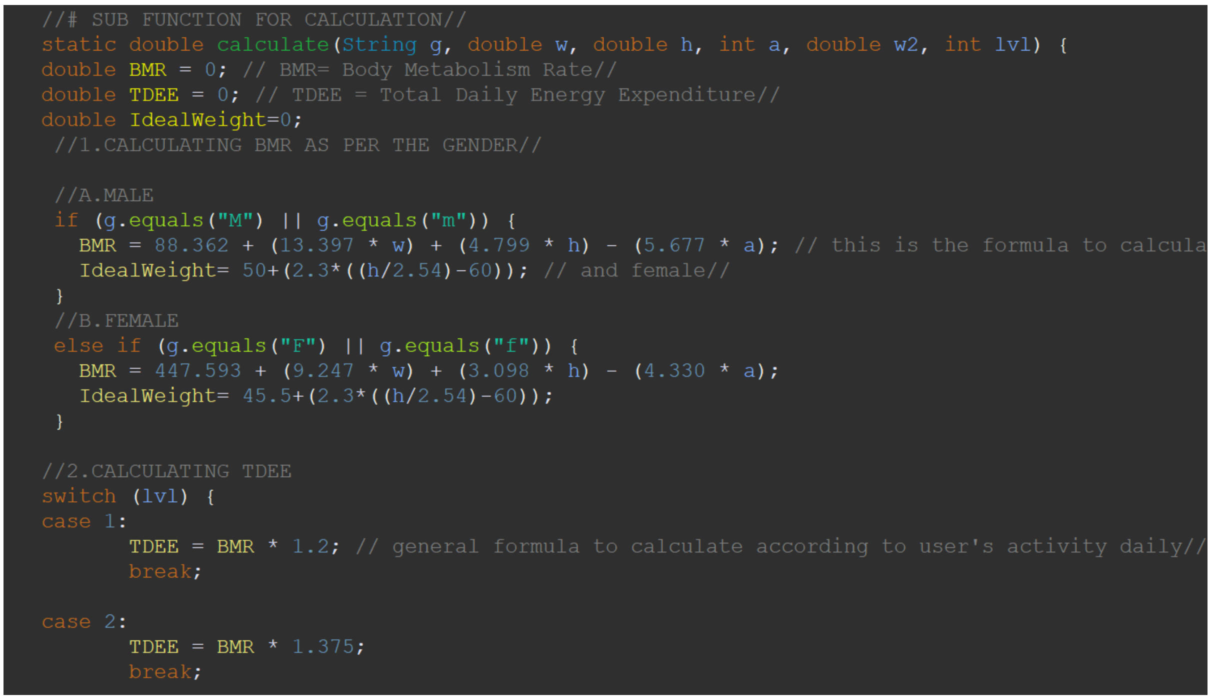
STEP 2: If the user has already set a profile, and wants to continue from that, options to retrieve data are provided. However, since weight and height change with time, to track them we let the user input their current weight.



```
DO YOU WANT TO RETRIVE YOUR PAST CREDENTIALS?
(PRESS 1: FOR RETRIVING || PRESS ANY BUTTON: FOR NOT RETRVING)

1

Input Your Current Weight(In Kgs)
(eg.69.70)
-------------------------
71

Input Your Current Height(In Cm)
(1 feet=30.8 cm)
------------------------
185

Your Diet Preference 1.Vegetarian | 2.Non- Vegetarian
```

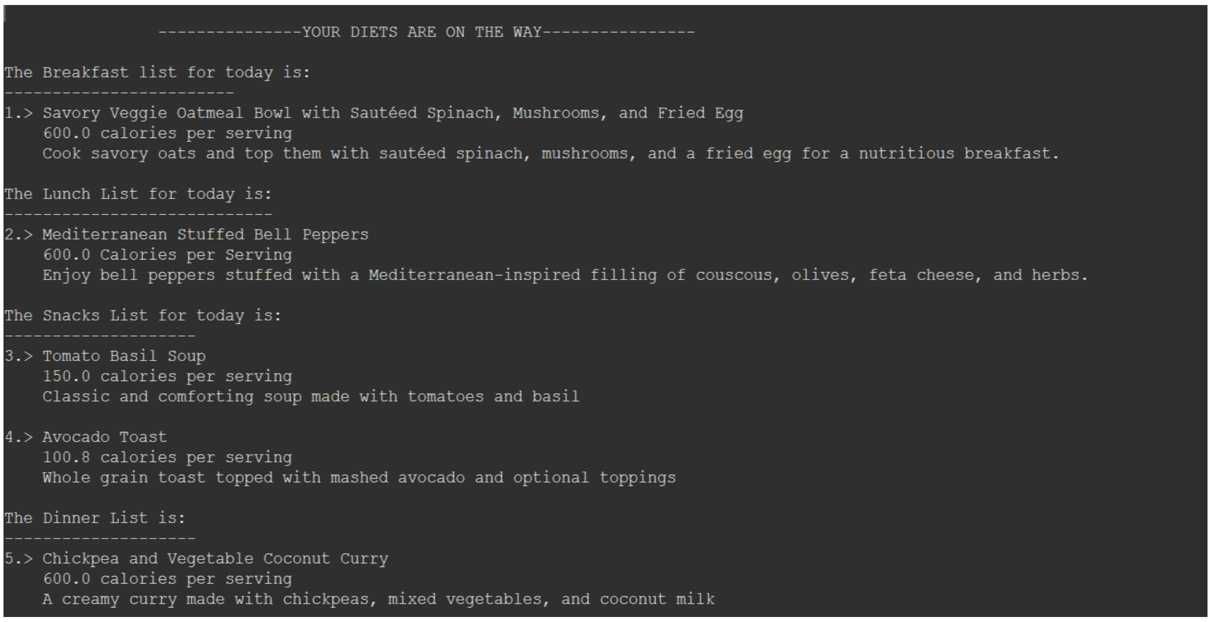
Step 3: After getting all the required credentials, we give the user their diet after many mathematical calculations in the Calculation Method.

BMR means the Basal Metabolism Rate and it is calculated using the "Harris-Benedict Formula", consequently TDEE by Taylor's formula is calculated and hence finally the required Daily calorie is obtained. The BMR of everyday input changes, so there is dynamic handling of the data throughout the code.



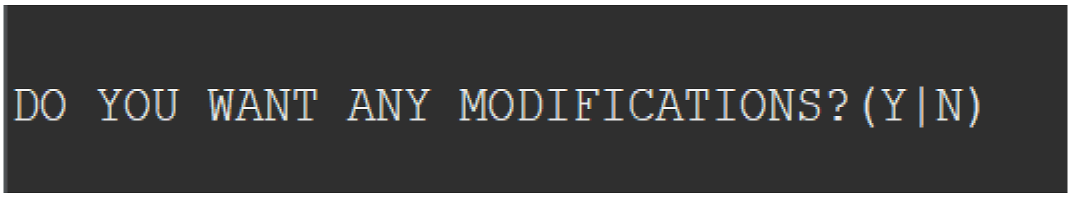
```
   //# SUB FUNCTION FOR CALCULATION//
   static double calculate(String g, double w, double h, int a, double w2, int lvl) {
   double BMR = 0; // BMR= Body Metabolism Rate//
   double TDEE = 0; // TDEE = Total Daily Energy Expenditure//
   double IdealWeight=0;
    //1.CALCULATING BMR AS PER THE GENDER//

    //A.MALE
    if (g.equals("M") || g.equals("m")) {
      BMR = 88.362 + (13.397 * w) + (4.799 * h) - (5.677 * a); // this is the formula to calcula
      IdealWeight= 50+(2.3*((h/2.54)-60)); // and female//
    }
    //B.FEMALE
    else if (g.equals("F") || g.equals("f")) {
      BMR = 447.593 + (9.247 * w) + (3.098 * h) - (4.330 * a);
      IdealWeight= 45.5+(2.3*((h/2.54)-60));
    }

   //2.CALCULATING TDEE
   switch (lvl) {
   case 1:
          TDEE = BMR * 1.2; // general formula to calculate according to user's activity daily//
          break;

   case 2:
          TDEE = BMR * 1.375;
          break;
```

Step 4: After all, finally, the diets are displayed in order of Breakfast, Lunch, Snack, and Dinner for a day. The user has options to modify (add, remove) the diets as per their preferences too.

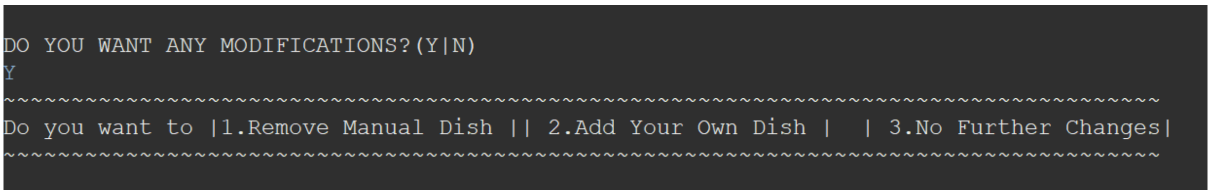


```
                --------------YOUR DIETS ARE ON THE WAY---------------

The Breakfast list for today is:
------------------------
1.> Savory Veggie Oatmeal Bowl with Sautéed Spinach, Mushrooms, and Fried Egg
    600.0 calories per serving
    Cook savory oats and top them with sautéed spinach, mushrooms, and a fried egg for a nutritious breakfast.

The Lunch List for today is:
----------------------------
2.> Mediterranean Stuffed Bell Peppers
    600.0 Calories per Serving
    Enjoy bell peppers stuffed with a Mediterranean-inspired filling of couscous, olives, feta cheese, and herbs.

The Snacks List for today is:
--------------------
3.> Tomato Basil Soup
    150.0 calories per serving
    Classic and comforting soup made with tomatoes and basil

4.> Avocado Toast
    100.8 calories per serving
    Whole grain toast topped with mashed avocado and optional toppings

The Dinner List is:
--------------------
5.> Chickpea and Vegetable Coconut Curry
    600.0 calories per serving
    A creamy curry made with chickpeas, mixed vegetables, and coconut milk
```

Modification section:

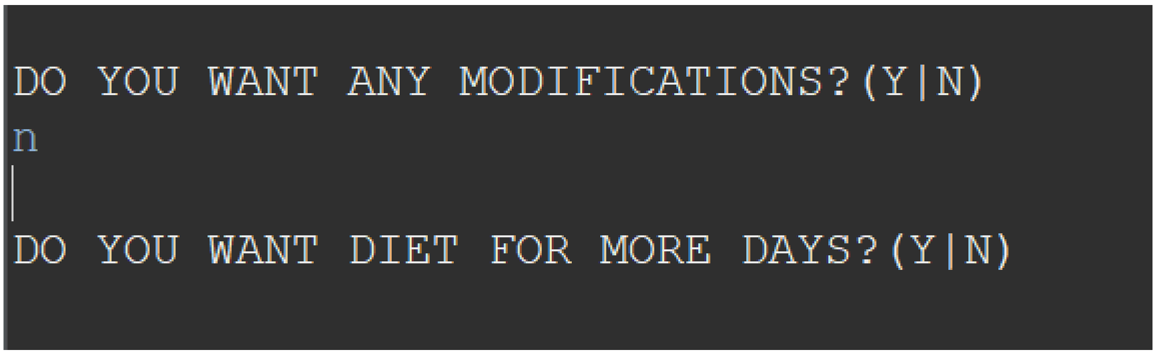


```
DO YOU WANT ANY MODIFICATIONS?(Y|N)
```



```
DO YOU WANT ANY MODIFICATIONS?(Y|N)
Y
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Do you want to |1.Remove Manual Dish || 2.Add Your Own Dish |  | 3.No Further Changes|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

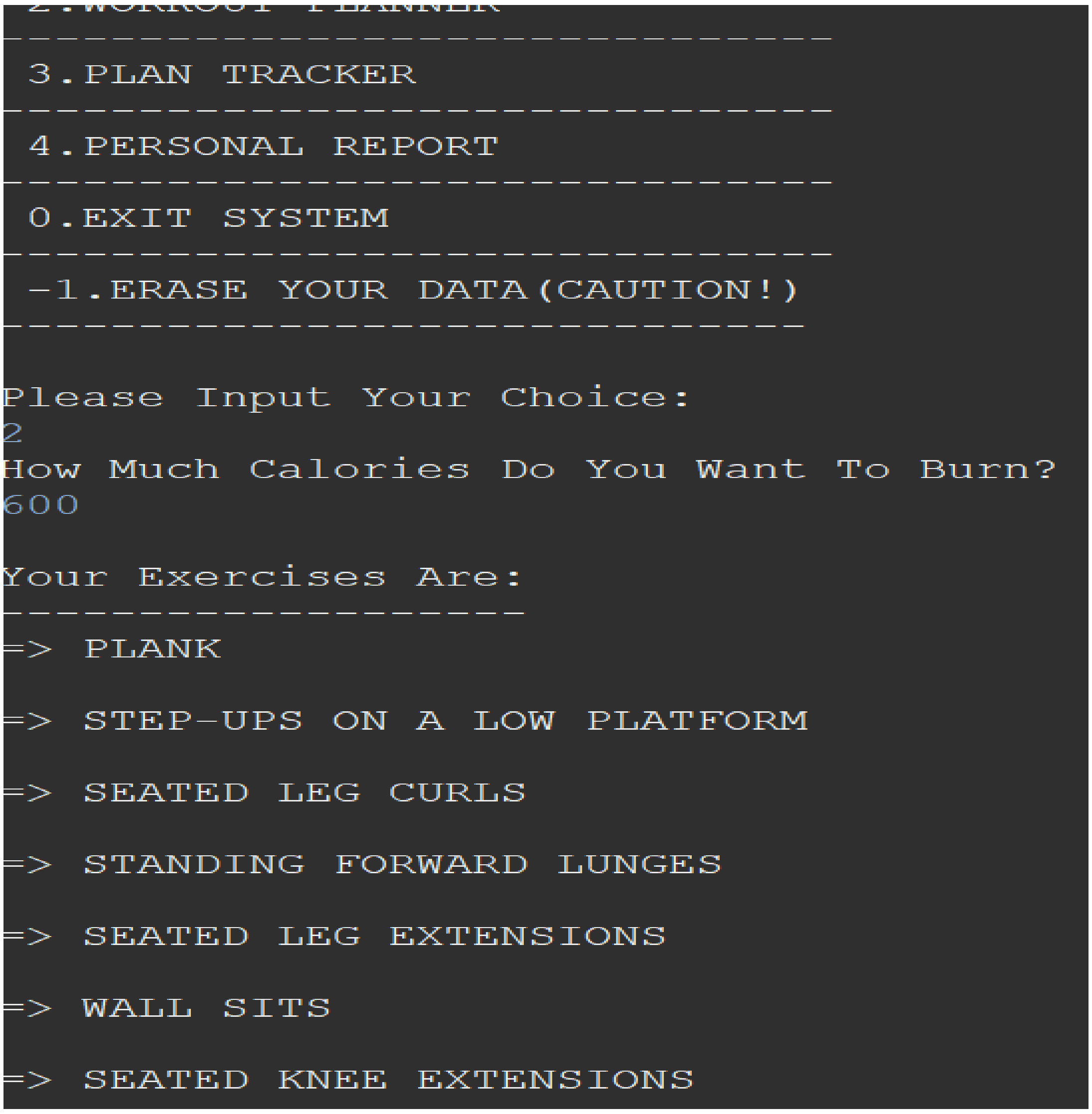
Step 5: After modifications, the user is given the option to create a diet plan for the next day too, while all the data of day 1 are stored inside a file safely.



```
DO YOU WANT ANY MODIFICATIONS?(Y|N)
n
|
DO YOU WANT DIET FOR MORE DAYS?(Y|N)
```

E. Explaining Workout Planner.

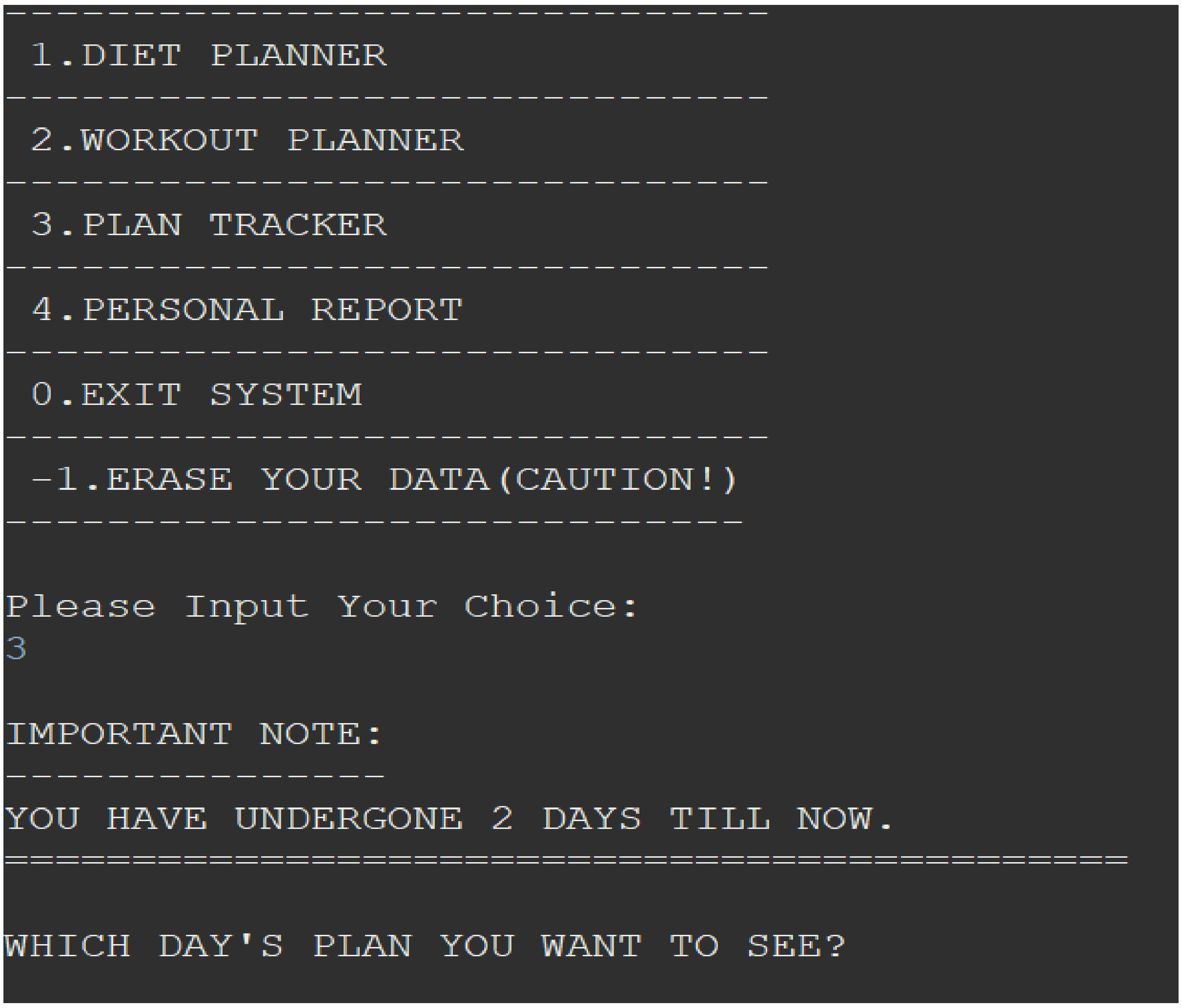
Workout Planner was included as **a secondary resource** in our project (which specifically focused on diet). However, the chain of diet and workout planners are interconnected. This means, that after the calculation of daily calories that the user needs throughout the day, we ask the user if they underwent the workout or not, if the user inputs "Yes", the calorie burnt by the user via. Workout is added, to the required calories, and then successively displayed them the diets. In the workout planner, the database from projectData is retrieved. Following are the steps in the Workout Planner method:



```
-----------------------------
 3.PLAN TRACKER
-----------------------------
 4.PERSONAL REPORT
-----------------------------
 0.EXIT SYSTEM
-----------------------------
 -1.ERASE YOUR DATA(CAUTION!)
-----------------------------

Please Input Your Choice:
2
How Much Calories Do You Want To Burn?
600

Your Exercises Are:
-------------------
=> PLANK

=> STEP-UPS ON A LOW PLATFORM

=> SEATED LEG CURLS

=> STANDING FORWARD LUNGES

=> SEATED LEG EXTENSIONS

=> WALL SITS

=> SEATED KNEE EXTENSIONS
```

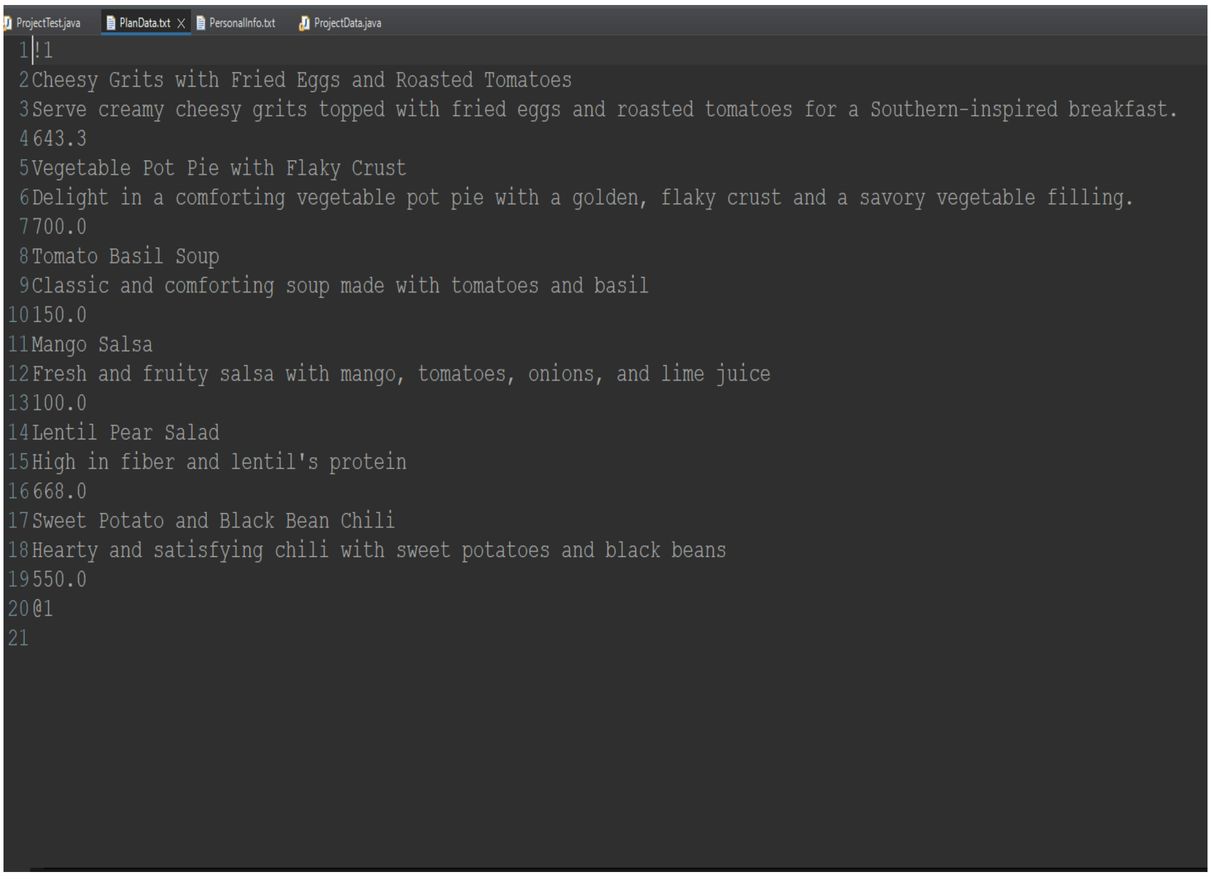
F. Explaining User Report.

Since we planned to build a personal system for planning and suggesting to the user of their diet requirement, there was a need to provide the required info's throughout their weight loss/gain journey. To serve that purpose, **a "Plan Tracker" was added** to our project. This feature enables the user **to access the diet plan they underwent each day of using our application for their reference.** Not only the diet plan, but user is also given to access the workout names they underwent on that particular day.



```
------------------------------
 1.DIET PLANNER
------------------------------
 2.WORKOUT PLANNER
------------------------------
 3.PLAN TRACKER
------------------------------
 4.PERSONAL REPORT
------------------------------
 0.EXIT SYSTEM
------------------------------
 -1.ERASE YOUR DATA(CAUTION!)
-----------------------------

Please Input Your Choice:
3

IMPORTANT NOTE:
---------------
YOU HAVE UNDERGONE 2 DAYS TILL NOW.
============================================

WHICH DAY'S PLAN YOU WANT TO SEE?
```

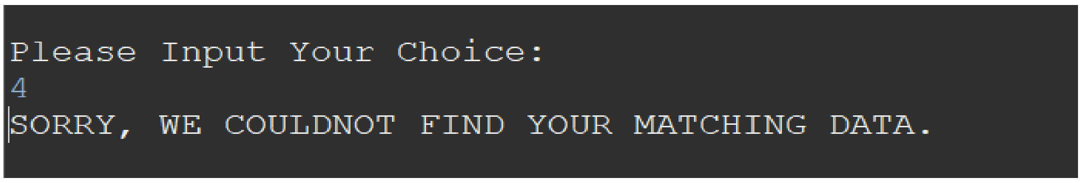
File I/O helps us to fulfil this feature, and until the user erases their data from our file, we keep on storing the diet they were suggested in the file “PlanData.txt”. By doing so, the user can compare their past diets with their current diets, if applicable they would also be able to follow the same diet they ate the day before, too. **Also, the user has access to their health facts like body Mass Index(BMI), daily Water Intake, Ideal Weight, and Weight Tracker since the day of commencement inside the report.** Below shows how the data is stored in the external file.



```
!1
Cheesy Grits with Fried Eggs and Roasted Tomatoes
Serve creamy cheesy grits topped with fried eggs and roasted tomatoes for a Southern-inspired breakfast.
643.3
Vegetable Pot Pie with Flaky Crust
Delight in a comforting vegetable pot pie with a golden, flaky crust and a savory vegetable filling.
700.0
Tomato Basil Soup
Classic and comforting soup made with tomatoes and basil
150.0
Mango Salsa
Fresh and fruity salsa with mango, tomatoes, onions, and lime juice
100.0
Lentil Pear Salad
High in fiber and lentil's protein
668.0
Sweet Potato and Black Bean Chili
Hearty and satisfying chili with sweet potatoes and black beans
550.0
@1

```

(fig. data stored in the external file)

If there is no data found in the external file, it will prompt the user no data.

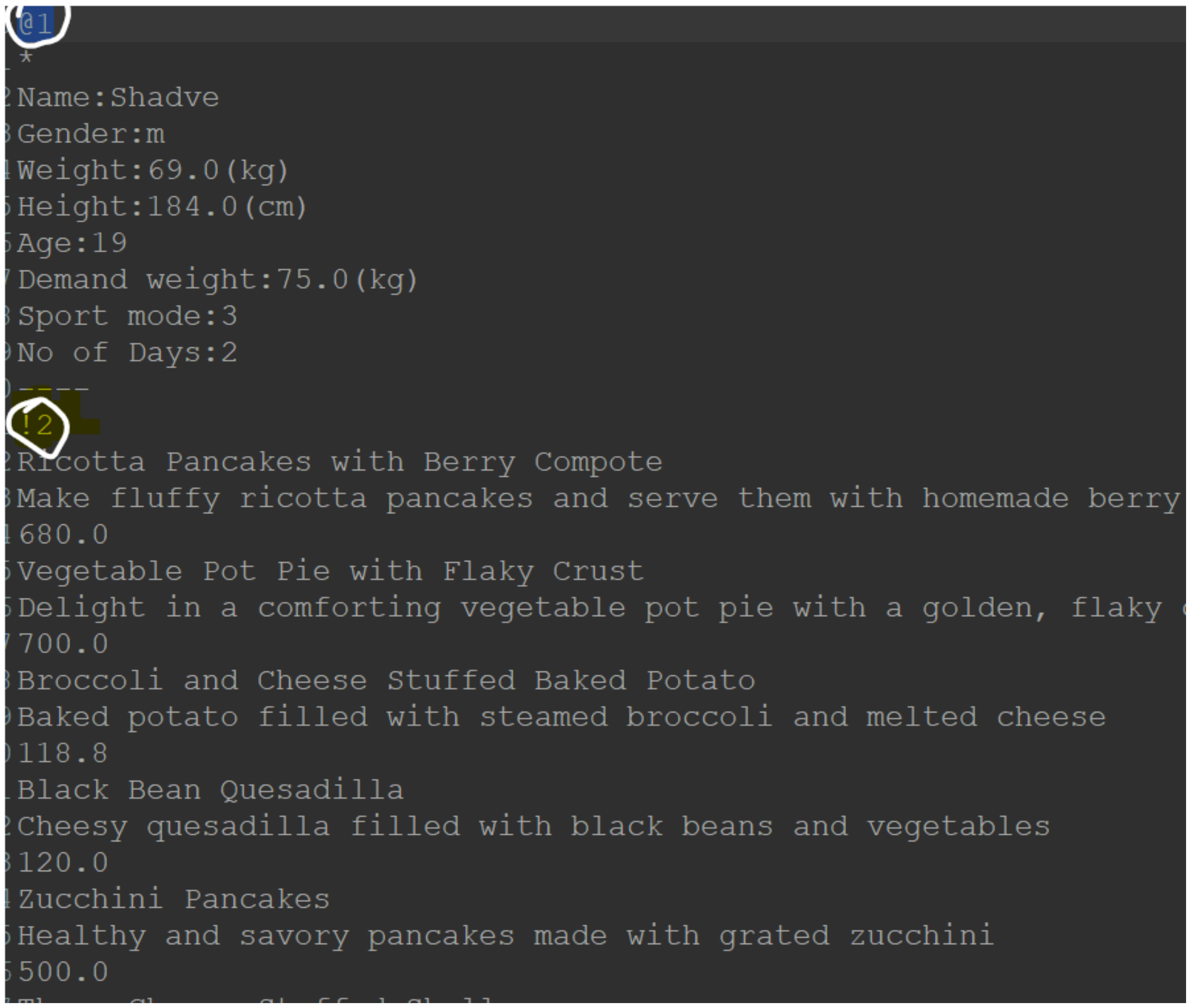


```
Please Input Your Choice:
4
SORRY, WE COULDNOT FIND YOUR MATCHING DATA.
```

G. Explaining Information Handling

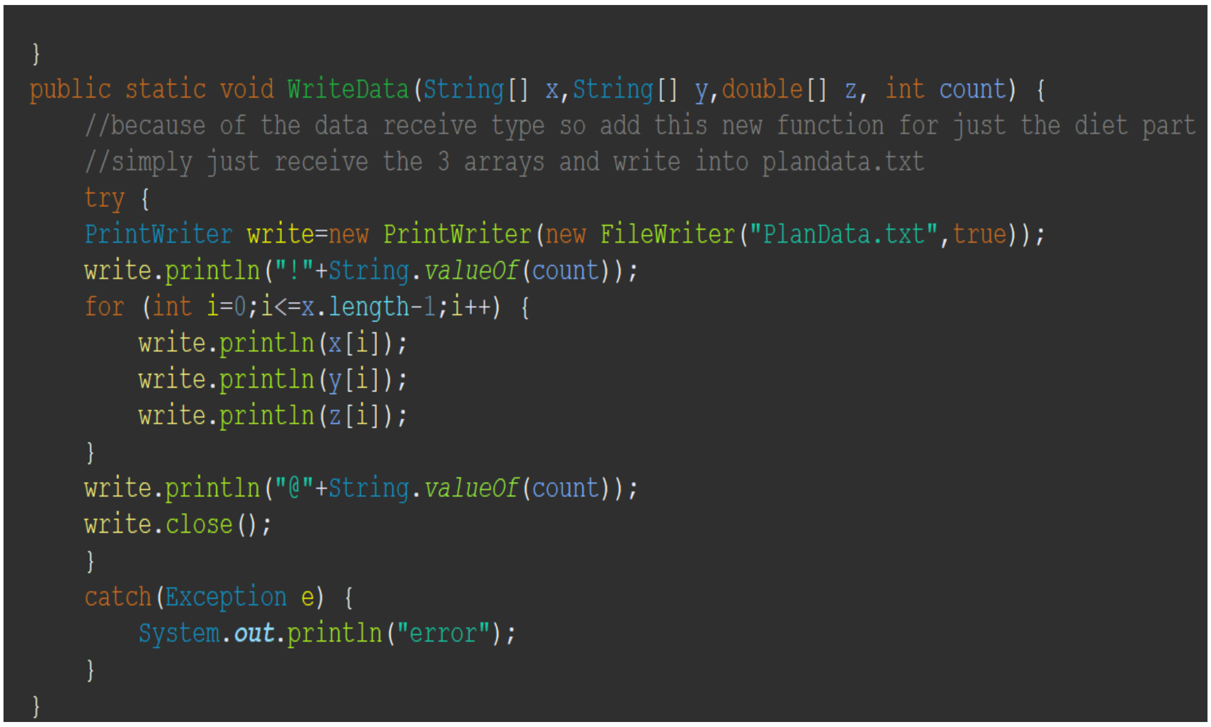
The concepts of File I/O were utilized to enable the information handling effectively. Firstly, we created two files, one for the tracking system and other health information that could be extracted using the personal information the user provided, and the second named **“planData.txt”** which was used to store the major information.

To build a proper flow of the storage and avoid overlapping of the data each data segment was labelled with various characters like !, *,@, #, etc.



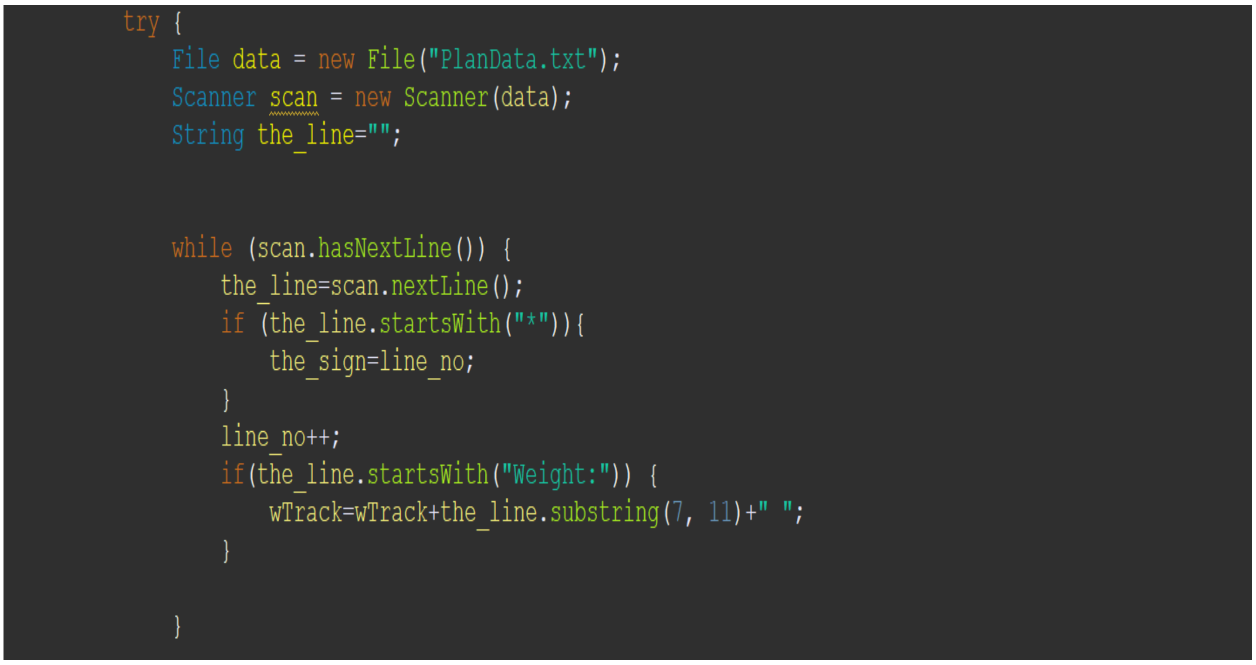
```
@1
*
Name:Shadve
Gender:m
Weight:69.0(kg)
Height:184.0(cm)
Age:19
Demand weight:75.0(kg)
Sport mode:3
No of Days:2
----
!2
Ricotta Pancakes with Berry Compote
Make fluffy ricotta pancakes and serve them with homemade berry
680.0
Vegetable Pot Pie with Flaky Crust
Delight in a comforting vegetable pot pie with a golden, flaky
700.0
Broccoli and Cheese Stuffed Baked Potato
Baked potato filled with steamed broccoli and melted cheese
118.8
Black Bean Quesadilla
Cheesy quesadilla filled with black beans and vegetables
120.0
Zucchini Pancakes
Healthy and savory pancakes made with grated zucchini
500.0
```

File Writing:



```
}
public static void WriteData(String[] x,String[] y,double[] z, int count) {
    //because of the data receive type so add this new function for just the diet part
    //simply just receive the 3 arrays and write into plandata.txt
    try {
    PrintWriter write=new PrintWriter(new FileWriter("PlanData.txt",true));
    write.println("!"+String.valueOf(count));
    for (int i=0;i<=x.length-1;i++) {
        write.println(x[i]);
        write.println(y[i]);
        write.println(z[i]);
    }
    write.println("@"+String.valueOf(count));
    write.close();
    }
    catch(Exception e) {
        System.out.println("error");
    }
}
```

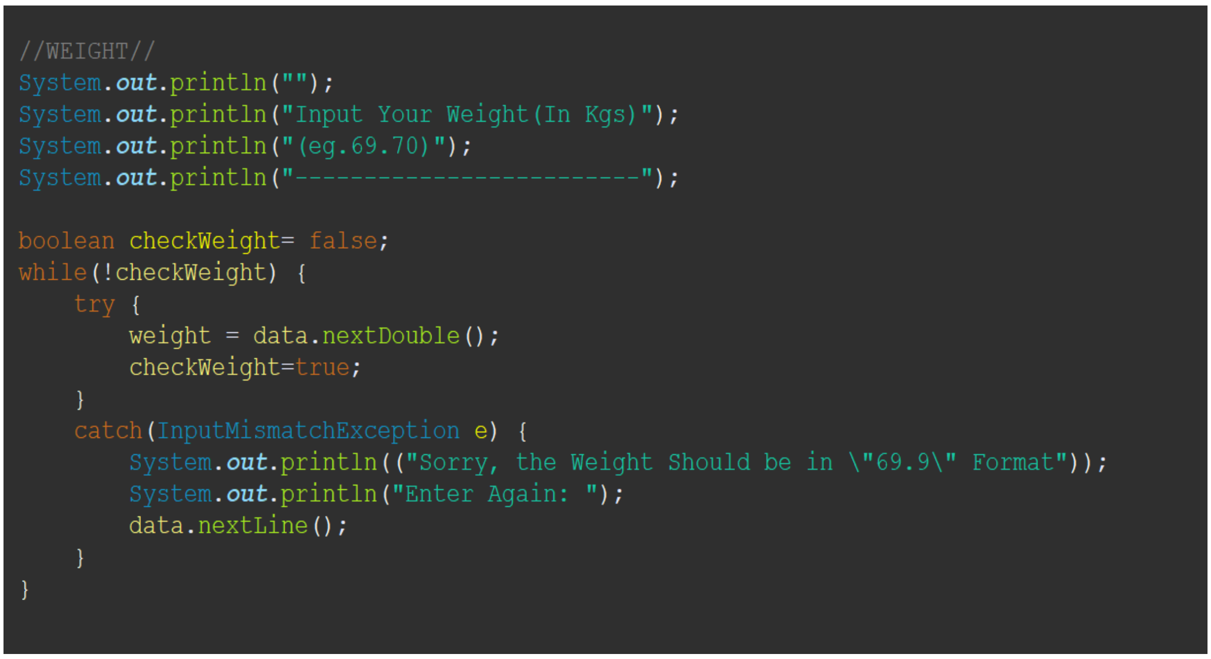
File Reading:



```
        try {
            File data = new File("PlanData.txt");
            Scanner scan = new Scanner(data);
            String the_line="";


            while (scan.hasNextLine()) {
                the_line=scan.nextLine();
                if (the_line.startsWith("*")){
                    the_sign=line_no;
                }
                line_no++;
                if(the_line.startsWith("Weight:")) {
                    wTrack=wTrack+the_line.substring(7, 11)+" ";
                }

            }
```

H. Explaining Error Handling

Because every function and method were chained together for a better flow of information throughout the program, various exceptions and out-of-bounds errors were highly interrupting in various cases. However various test cases were trialed multiple times, and because of it, probable conditions where there could be runtime errors in the code were caught. To avoid that, there is a strict feature in Java named try…catch, which was beneficial for us to catch the error without interrupting the code progression. To be specific, for example, in the given diagram, the try...catch is used to effectively get the error that the user is prompting. Here, if the user inputs weight indifferently than the suggested method, then the console will be bombarded with the prompt "Enter Again" and give the format for entering the data. This feature is utilized in numerous instances throughout the project, where data input should be done from the user.



```java
//WEIGHT//
System.out.println("");
System.out.println("Input Your Weight(In Kgs)");
System.out.println("(eg.69.70)");
System.out.println("------------------------");

boolean checkWeight= false;
while(!checkWeight) {
    try {
        weight = data.nextDouble();
        checkWeight=true;
    }
    catch(InputMismatchException e) {
        System.out.println(("Sorry, the Weight Should be in \"69.9\" Format"));
        System.out.println("Enter Again: ");
        data.nextLine();
    }
}
```

(Fig: try. catch used in error handling)

Apart from that, we've also used an effective utilization of while looping and if_else if looping conditions to check the possible errors. Various String methods like. equals, IgnoreUpperCase was also utilized in our program to compare objects. Similarly, every switch case also uses a default value case to break the switch when an unusual input is entered.

Self-reflection of Shadev Kharel:

The coding of Diet Planner was undoubtedly a valuable learning opportunity for me. To begin with, During the coding of the project, there were multiple hindrances and errors which disturbed the logical flow of the code. While I initially wrote code in a good flow, I forgot to constantly monitor the possibility of errors. It was when after around 300-500 lines, I realized there were multiple runtime errors in the code. Now, even if the codes were logically organized, it became tough for me to spot the location where the syntax was being violated. However, I gave my utmost effort to deal with it. Similarly, while we chose a very constrained topic, which required a large database set, it was difficult to manually enter every dish, and calorie, and find the accurate sources for it. Despite all these challenges, the project taught me how to be user-oriented and interactive.

Self-reflection of Ip Tsun Ting:

While developing this project I deal with exporting and importing the data, proofreading, UI maintenance, and multiple sourcing of the data. During the project, I assisted my group mate in various unity requiring tasks like data entry of raw data, and online sourcing. Which in the process, I also learned better to deal with the data of the object in a class also the File I/O is one of the problems while we trying to save the data and take out to load the progress of the user, which sometimes the data can’t be written in the data file. So, this is a good experience for me to learn more about using the file I/O function to save data.

Conclusion and Remarks.

In conclusion, the Java project was a valuable opportunity for us to hone our programming skills and become more familiar with various underlying Java complexities. Because, we have managed to pull off database handling, effectively securing the user’s information, we believe we have got a vivid image of the practical approach required in building a project.

Overall, the “Diet and Workout Planner” project showcased the significance of proper data handling in creating a successful application. While the topic we chose was constrained to a very limited domain of application, the ultimate focus given to the extraction of as many resources out of the given data paid off in building this project. By adhering to a user-friendly, customizable, and reliable application for the user to manage their food routine daily, the project was thoroughly coded. Despite all these, we also believe the project can further scale in fore coming days, and since, there is still room for improvements, both scalability and error handling should go hand in hand. While, this is a lower-scale program, with no use any JSON or graphical library, we believe this is and can be one of the true applications in the nutritional and health-conscious modern world.

References

1. Editor. (2019, January 15). *BMR calculator calculates your basal metabolic rate is the number of calories required to keep your body functioning at rest, also known as your metabolism.* Diabetes. https://www.diabetes.co.uk/bmr-calculator.html