TAYLOR'S UNIVERSITY

Wisdom · Integrity · Excellence

IT'S IN YOU TO **OWN THE FUTURE**

ITS63304 Object Oriented Programming

Group Project (50%)

Semester March 2021

Instructions to students:

- Complete this cover page and attach it to your assignment – this should be your first page.

Student declaration:		
I declare that: • **I understand what is meant by plagiarism** • **The implication of plagiarism have been explained to us by our lecturer** **This project is all our work and I have acknowledged any use of the published or unpublished works of other people.**		
Names of Group Members		
No	**Student ID**	**Student Name**
1	0347555	Lim Kang Wei
2	0348473	Khoo Tze Wei
3	0348321	Ng Wan Rou
4	0348584	Heng Yuen Lam
5	0348476	Chan Jia QI

Read the description and requirements below carefully. If you have any questions, please see your lecturer(s) for further clarification.

Table of Content

Marking Rubric

Criteria	Excellent (8-10)	Good (6-7)	Average (4-5)	Poor (0-3)	Your Score
User Interface	Very appealing and user-friendly	Moderate appealing and user-friendly	Average appealing and user-friendly	Not appealing nor user-friendly	
Features	Unique and very practical	Moderate practical	Average practical	Not practical	
Usability	Run smoothly and bug free	Less smoothly and some bugs	Lagging and same bug	Lagging and very buggy	
Source code	Very logical and well organized	Moderate logical	Average logical	Not logical	
Report	Very detail and complete content	Detail and good content	Less detail and average content	Not detail and not complete	
				TOTAL	_____

NOTE: Total marks will be adjusted to a maximum of 50% allocated for this assignment.

Application Description and Rational

According to Kaur (2019), among all the Asian countries, Malaysia has the highest obesity and overweight rate with a percentage of 64% and 65% for each male and female. Moreover, this result in an increment of diabetes among the adult aged 18 and above from 11.6% to 17.5% in 2006 to 2015. According to the current situation, most people doesn't know how to maintain a healthy lifestyle, so we decided to create an application related to healthy lifestyle to reduce the obesity rate and underweight cases. The application that we created is called Kang Wei's Smart Diet Management Plan.

Kang Wei's Smart Diet Management application helps users make bite-sized bodyweight changes. Our application gathers input in a user-friendly way. By following our recipes cut out for the user's calorie intake, they are able to achieve their dream bodyweight easily. We have extensive logic gates to ensure that any input mistakes can be detected immediately and will prompt corrective measurements to user.

Role and Responsibility of Each Group Member

Group Member Name	Role and Responsibility
Lim Kang Wei	- Leader of the group - Code the Main class - Researcher - Find one daily recipe for each gain and lose weight section - Debugger
Khoo Tze Wei	- Documentarian: Make the report - Code the BMI class - Find one daily recipe for each gain and lose weight section
Ng Wan Rou	- Code the Array List part - Find one daily recipe for each gain and lose weight section - PPT Maker
Heng Yuen Lam	- Code the CaloriesConsume class - Find one daily recipe for each gain and lose weight section - Debugger
Chan Jia Qi	- Code the Array List part - Find one daily recipe for each gain and lose weight section - Proofread the codes - Debugger

Application Features and Usefulness

Application Features

- Calculate user's BMI and show their BMI status (Samsukha, n.d.)
- Calculate the user's BMR to recommend daily calories intake (Samsukha, n.d.)
- Provide user a 5 days diet recipe according to their calorie intake (Ruchir, n.d.)
- User able to input their daily calories intake and view their daily calorie consumption after a week
- Collect daily weight using ArrayList, so that users can review their overall progress.

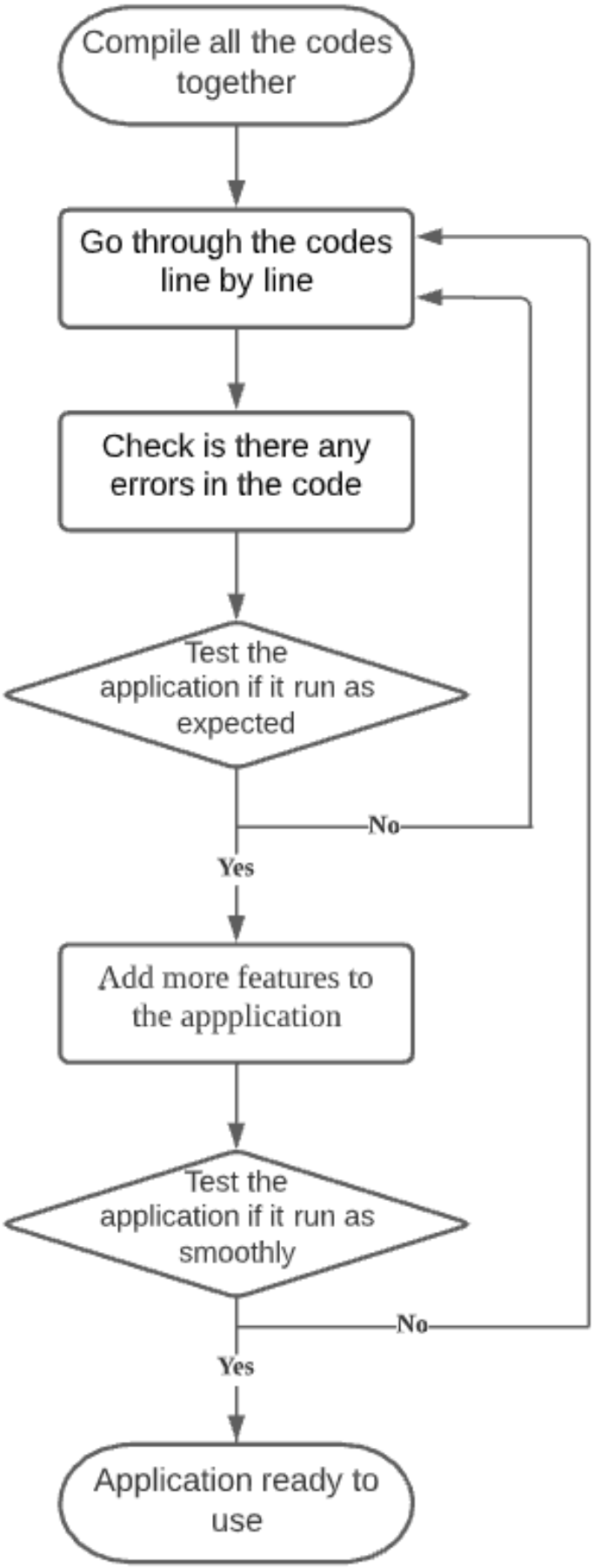
Usefulness

- Achieved the minimum requirement for a developed smart diet management application.
- Basic and user-friendly interface.
- Able to guide user to gain or lose weight effectively.
- Provide considerate recipe
- Require simple input from user

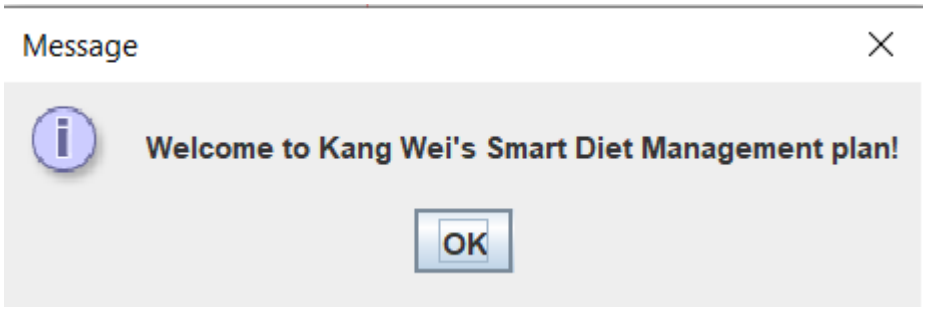
Testing Plan and Implementation

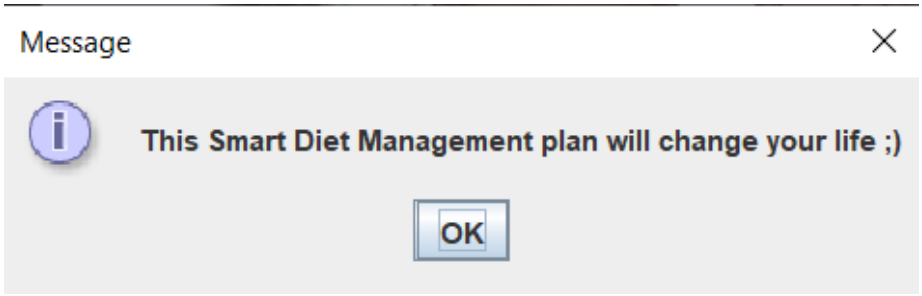
Testing Plan

1. Requirement *1st cycle* - Check the project requirement *2nd cycle* - Check once again if we had met all the requirement	**2. Design** *1st cycle* - Choose a topic to develop - Discuss the requirement and feature that need to include in the application - Distribute the work among the team mates *2nd cycle* - Add more features like restricting user to input ridiculous value to make the application to look more professional
4. Testing *1st cycle* - Test on the codes if it works as expected - Debug the error that appear in the application *2nd cycle* - Test the application again after adding in more features - Debug once again to ensure no bug or error in the application	**3. Development** *1st cycle* - Start to code the application according to the first discussion *2nd cycle* - Code the add-on features


Compile all the codes together
Go through the codes line by line
Check is there any errors in the code
Test the application if it run as expected
No
Yes
Add more features to the appplication
Test the application if it run as smoothly
No
Yes
Application ready to use

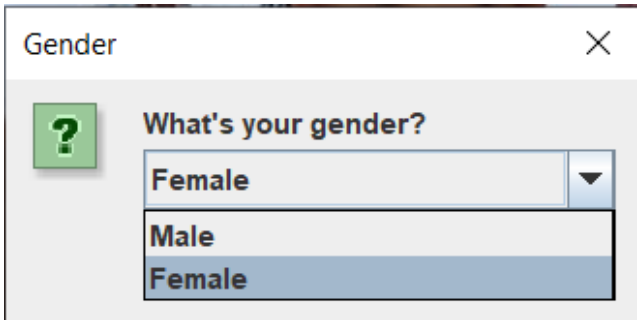
Implementation





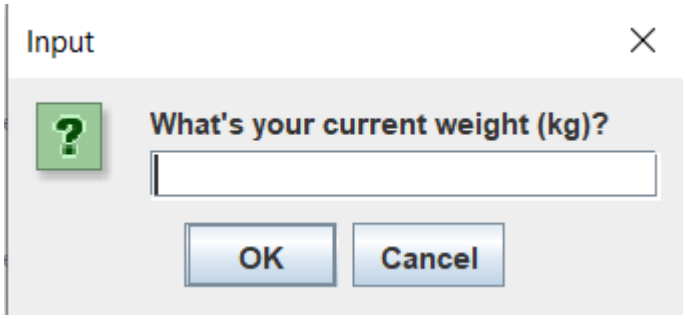
1. Welcome message to the user who use this application.

2. User get to input their name.



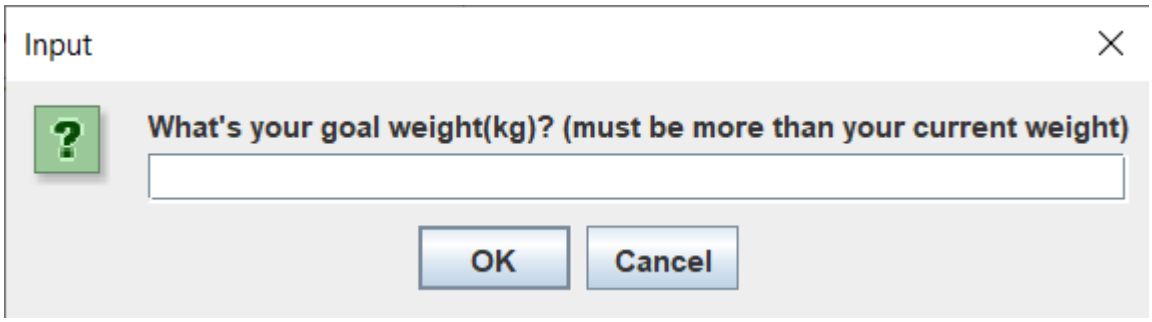
3. User get to select their gender.

4. User get to input their age.

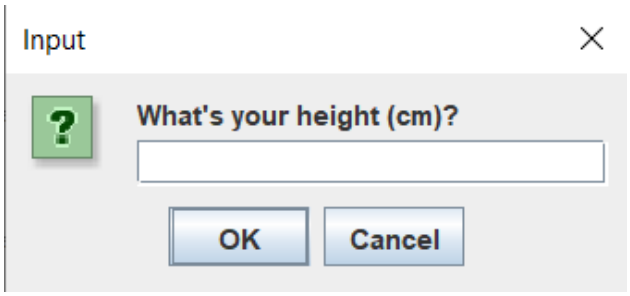


5. User get to input their weight.

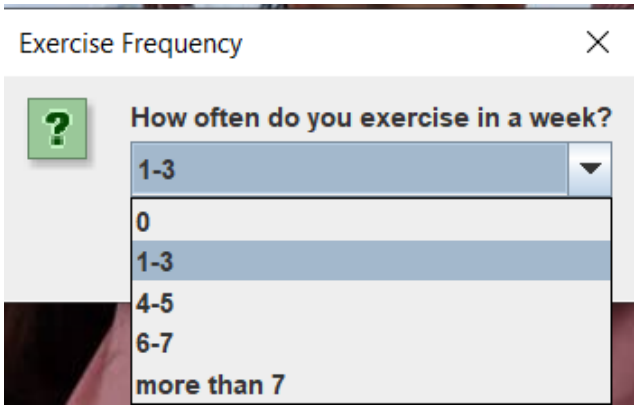
6. User get to choose to gain or lose weight.



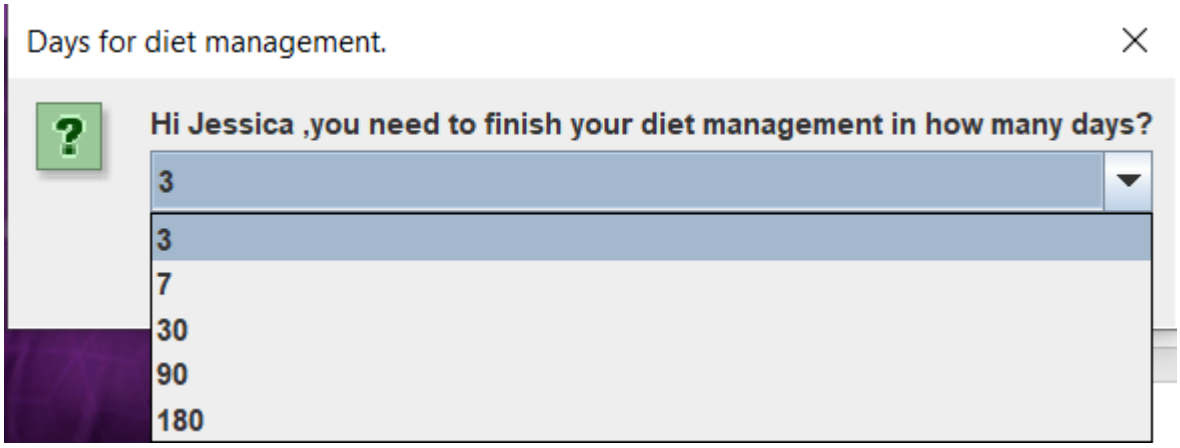
7. User will get to input their ideal weight.



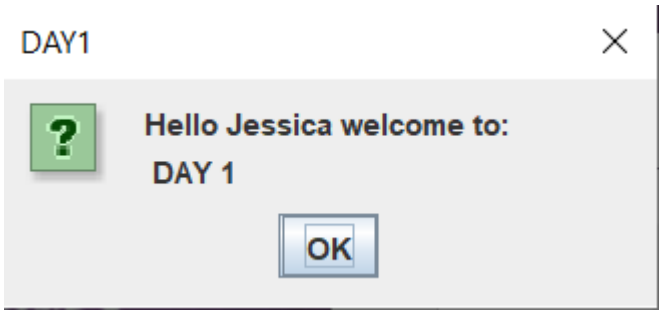
8. User get to input their height.



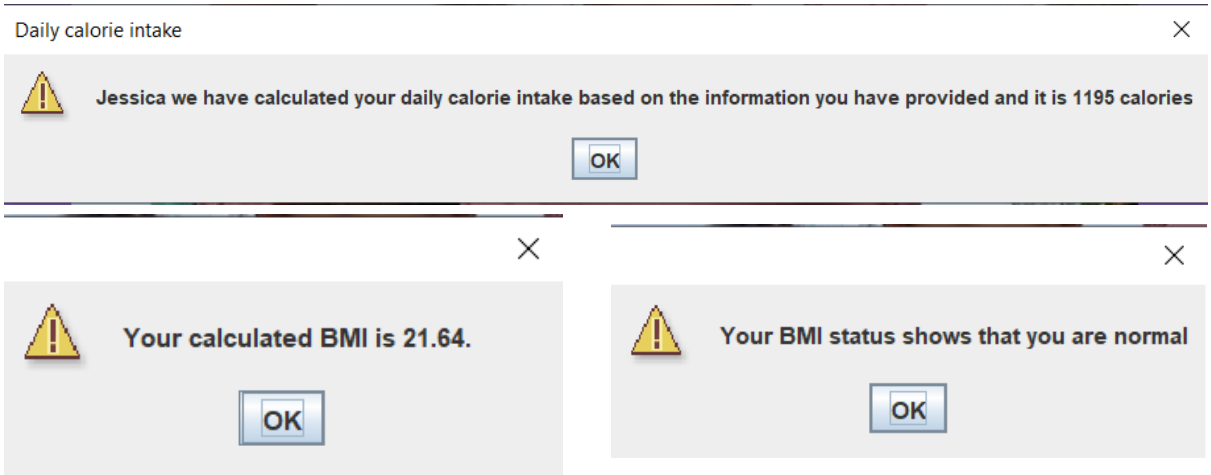
9. User will need to input their exercise frequency.



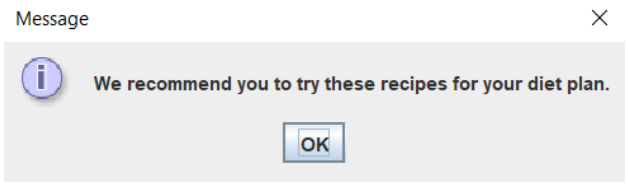
10. User are able to select how many days do they want to achieve their goals.

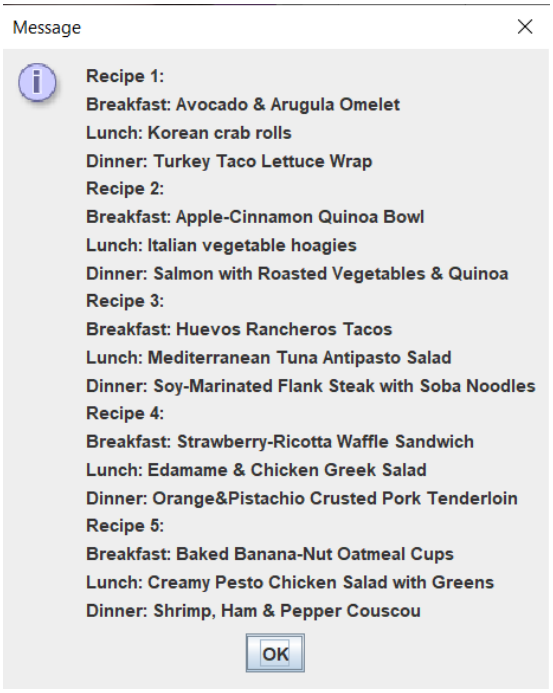


11. A greeting message will pop up.

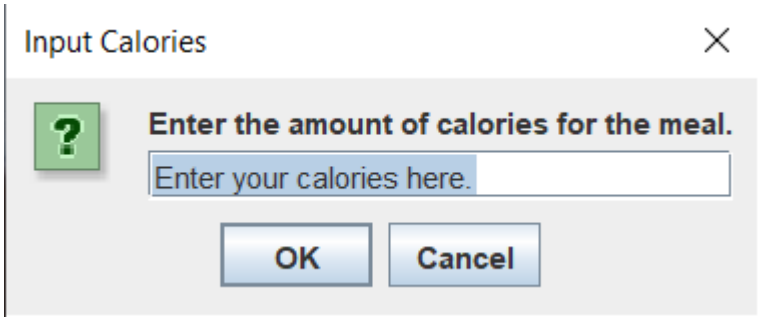


12. 3 messages will pop out. First is regarding to user's recommended calories intake (Check Your Health, n.d.). Different gender have different formula for calculating it (Mucha, Pawlik & Rain, 2020). Second is the calculated BMI (Diabetes Canada, n.d.). Last but not least is the BMI status of the user will be shown out (Centers for Disease Control and Prevention, 2020).

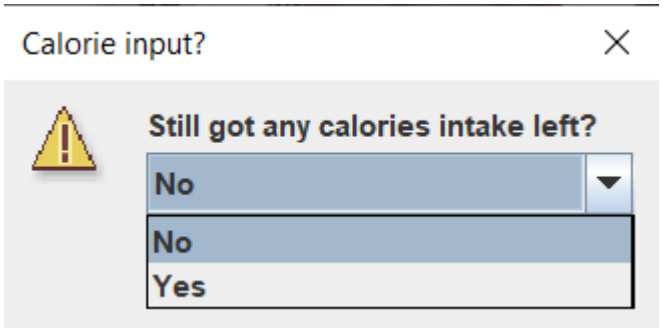




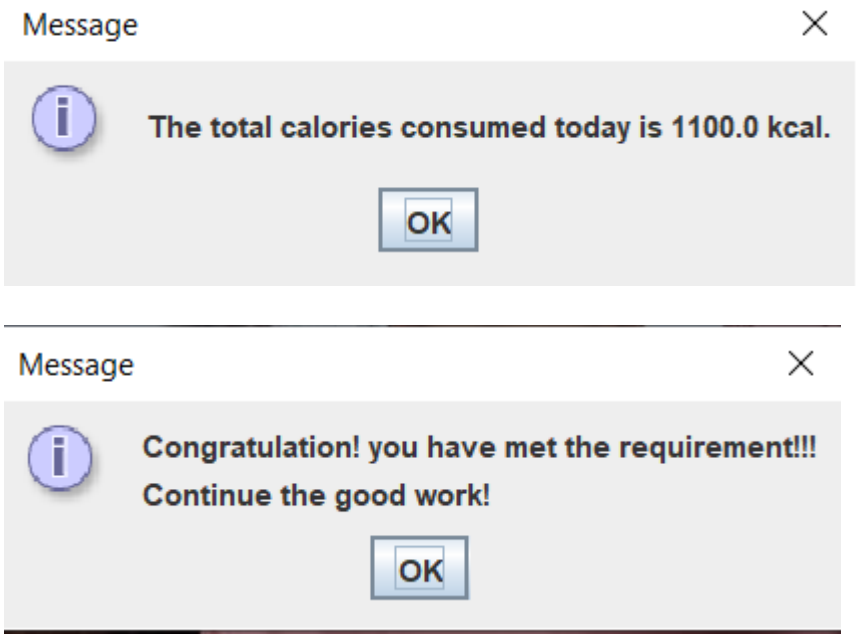
13. A recommended 5 days diet plan will be shown out according to the recommended daily calories intake (Delicious, n.d.; Delish, n.d.; Our Grilling Guide Will Turn You into a Pro, n.d.;).



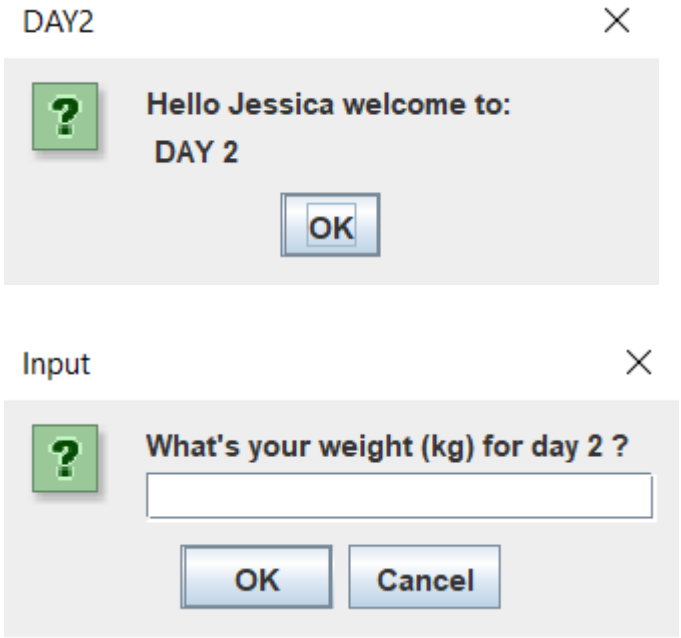
14. User get to input the calories of their meal that the user has taken.

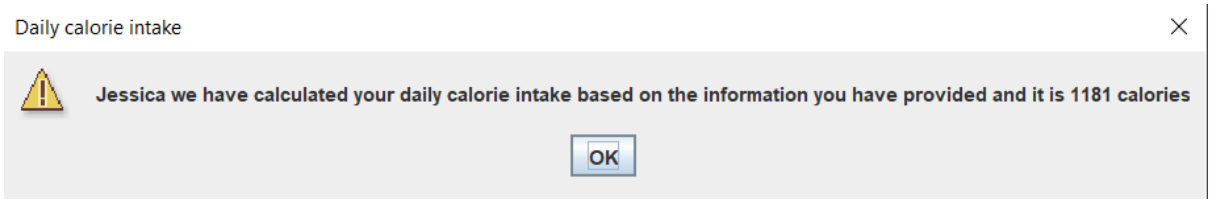


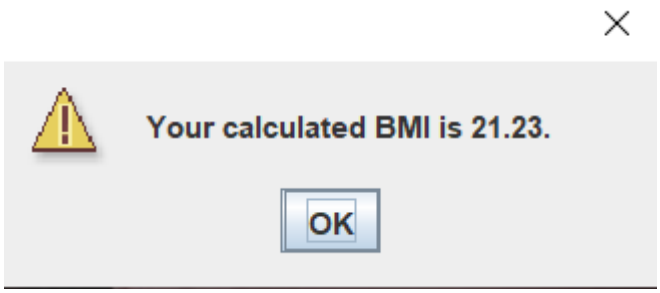
15. If there is additional calories intake that have not been input, user is able to choose the option “Yes” to input their calories intake.

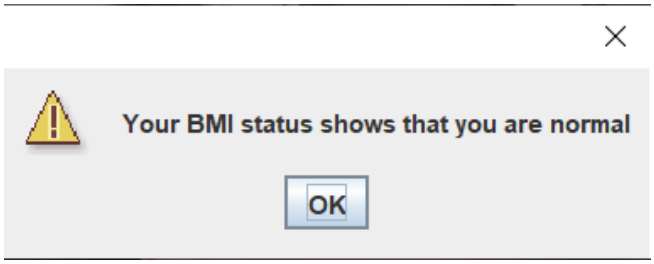


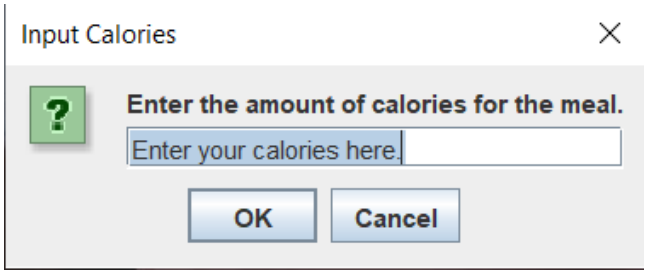
16. A message regarding to the total daily calories consumed will be shown.

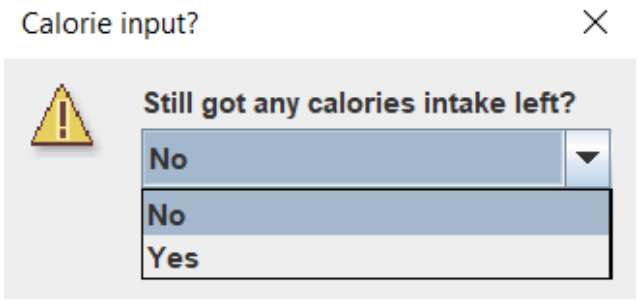


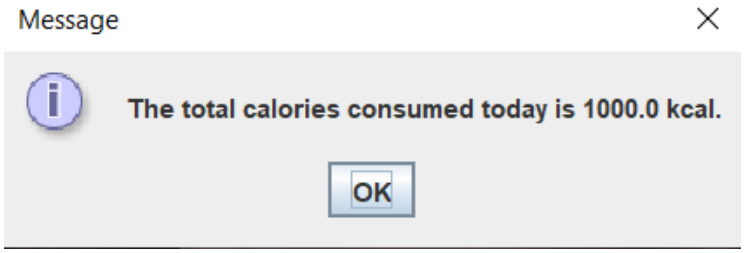

Daily calorie intake
Jessica we have calculated your daily calorie intake based on the information you have provided and it is 1181 calories
OK

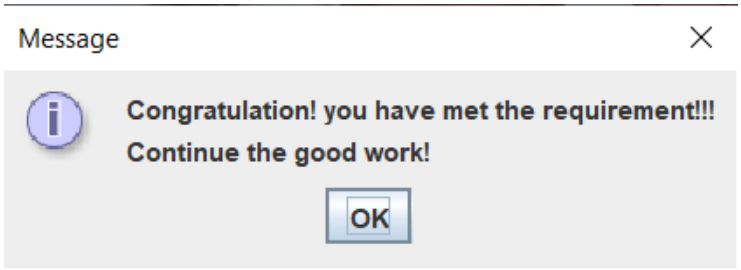

Your calculated BMI is 21.23.
OK


Your BMI status shows that you are normal
OK

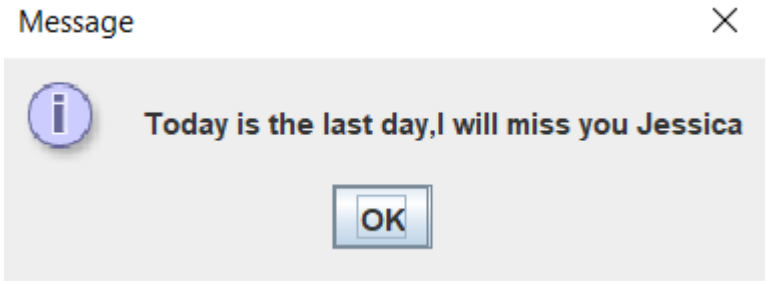

Input Calories
Enter the amount of calories for the meal.
Enter your calories here.
OK
Cancel


Calorie input?
Still got any calories intake left?
No
No
Yes

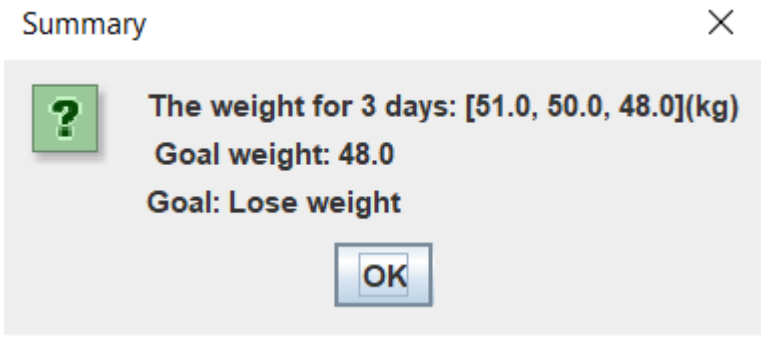

Message
The total calories consumed today is 1000.0 kcal.
OK



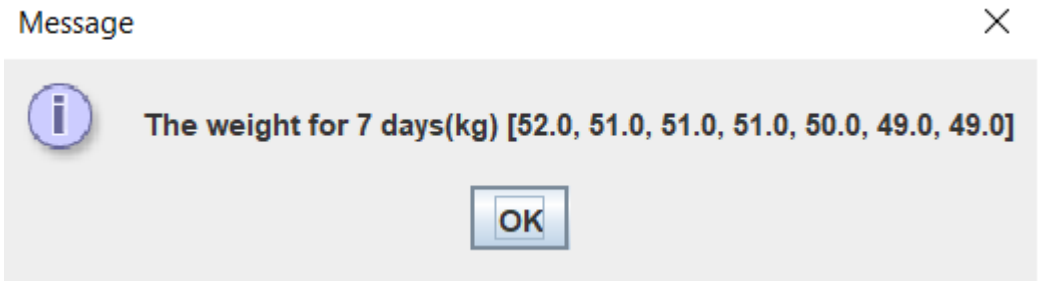
17. All these will be looped until the day was selected by user to achieve his or her goal.



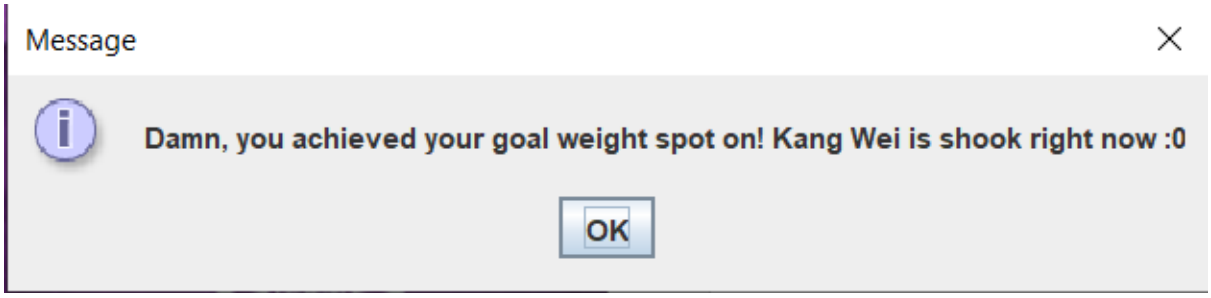
18. On the last day, a message will pop out.

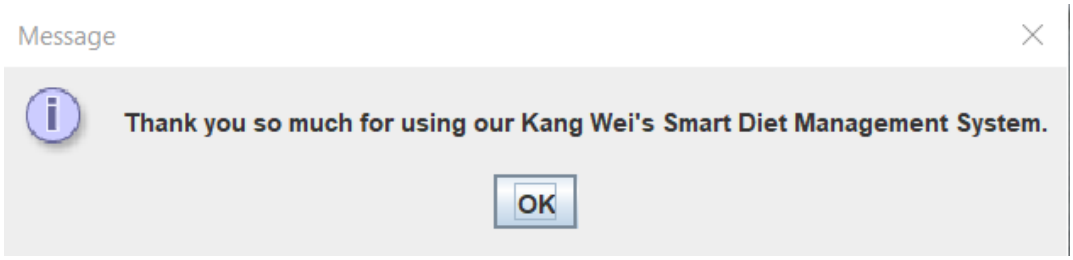


19. A summary will pop out.



20. After day 7, user is able to view the changes in weight for the pass 7 days.





21. After user is done using the application, a message will be shown to thanks them.

References

Centers for Disease Control and Prevention. (2020). Assessing Your Weight. Retrieved from https://www.cdc.gov/healthyweight/assessing/index.html#:~:text=If%20your%20BMI%20is%20less,falls%20within%20the%20obese%20range.

Check Your Health. (n.d). How Many Calories Do You Need? Retrieved from https://www.checkyourhealth.org/eat-healthy/cal_calculator.php

Delicious. (n.d.). Retrieved from Recipes, Restaurants, Travel, Food News & More - delicious.com.au

Delish. (n.d.). Retrieved from Recipes, Party Food, Cooking Guides, Dinner Ideas - Delish.com

Diabetes Canada. (n.d.). Body Mass Index (BMI) Calculator. Retrieved from https://www.diabetes.ca/managing-my-diabetes/tools---resources/body-mass-index-(bmi)-calculator

Kaur, N. (2019). Malaysia and WHO call for more investment in primary health care the 21st century. Retrieved from https://www.who.int/malaysia/news/detail/08-04-2019-malaysia-and-who-call-for-more-investment-in-primary-health-care-the-21st-century

Mucha, M., Pawlik, K., & Rain, R., (2020). BMR Calculator (Basal Metabolic Rate, Mifflin St Jeor Equation). Retrieved from https://www.omnicalculator.com/health/bmr

Our Grilling Guide Will Turn You into a Pro. (n.d.). Retrieved from Recipe Ideas, Product Reviews, Home Decor Inspiration, and Beauty Tips - Good Housekeeping

Ruchir, C. (n.d.). Must-Have Features to Add When Making a Diet and Nutrition Tracking App. Retrieved from https://www.cisin.com/coffee-break/Enterprise/must-have-features-to-add-when-making-a-diet-and-nutrition-tracking-app.html

Samsukha, A. (n.d.). Diet & Nutrition Tracking App Development Cost & Features. Retrieved from https://www.emizentech.com/blog/diet-nutrition-tracking-app-development.html

Appendix A: SDMInput

```
package sdm.input;

import java.util.ArrayList;

import javax.swing.JOptionPane;


public class SDMInput {


    public static void main(String[] args) {

        String name;

        String gender;

        int age = 0;

        String goal;

        double currentWeight;

        JOptionPane.showMessageDialog(null, "Welcome to Kang Wei's Smart Diet Management plan!");

        JOptionPane.showMessageDialog(null, "This Smart Diet Management plan will change your life ;)");

        name = JOptionPane.showInputDialog("What's your name?");

        Recipe rec = new Recipe();

        String[] acceptableValues = {"Male", "Female"};

        gender = (String) JOptionPane.showInputDialog(null,

                "What's your gender?",

                "Gender",

                3,

                null,
```

```
        acceptableValues,
        acceptableValues[1]);

    age = Integer.parseInt(JOptionPane.showInputDialog("What's your age? "));
    while (age < 15 || age > 60) {
       JOptionPane.showMessageDialog(null,
            "Your age isn't suitable to go on a diet.",
            "Invalid input.",
            0);

       age = Integer.parseInt(JOptionPane.showInputDialog("What's your age? "));
    }

    currentWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your
                    current weight (kg)?"));

    while (currentWeight >= 200.0 || currentWeight <= 30.0) {
       JOptionPane.showMessageDialog(null,
            "Your weight isn't suitable for this Smart Diet Management.",
            "Invalid input.",
            0);

       currentWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your
                       current weight (kg)?"));
    }
```

```
String[] acceptableValues2 = {"Gain weight", "Lose weight"};
goal = (String) JOptionPane.showInputDialog(null,
        "What's your goal?",
        "Goal",
        3,
        null,
        acceptableValues2,
        acceptableValues2[1]);

double goalWeight;
if(goal.equals("Gain weight")){
    goalWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your goal
                weight(kg)? (must be more than your current weight)"));
    while (goalWeight < currentWeight || goalWeight >(currentWeight + 20.0) ) {
   JOptionPane.showMessageDialog(null,
            "Please input a valid goal weight according to your goal.",
            "False input.",
            0);
   goalWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your goal
                weight(kg)? "));
}
}else{
    goalWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your goal
                weight(kg)? (must be less than than your current weight)"));
    while (goalWeight > currentWeight || goalWeight<(currentWeight - 20.0)) {
```

```
    JOptionPane.showMessageDialog(null,
            "Please input a valid goal weight according to your goal.",
            "False input.",
            0);
    goalWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your goal
                    weight(kg)?"));
}
}

boolean goalCheck = goal.equals("Gain weight");

double heightCm = Double.parseDouble(JOptionPane.showInputDialog("What's your
                        height (cm)? "));
while (heightCm <= 120.0 || heightCm >= 250.0) {
    JOptionPane.showMessageDialog(null,
            "Your height isn't suitable for this Smart Diet Management.",
            "Invalid input.",
            0);

    heightCm = Integer.parseInt(JOptionPane.showInputDialog("What's your height (cm)?
                    "));
}
double heightM = heightCm / 100;

String[] acceptableValues3 = {"0", "1-3", "4-5", "6-7", "more than 7"};
String exerciseFrequency = (String) JOptionPane.showInputDialog(null,
```

```
        "How often do you exercise in a week?",
        "Exercise Frequency",
        3,
        null,
        acceptableValues3,
        acceptableValues3[1]);

ArrayList<Double> todayweight = new<Double> ArrayList();
double dailyWeight;

String[] acceptableValues4 = {"3","7","30","90","180"};
String oldday =(String)(JOptionPane.showInputDialog(null,
        "Hi " +name+" ,you need to finish your diet management in how many days?",
        "Days for diet management.",
        3,
        null,
        acceptableValues4,
        acceptableValues4[0]));
int day = Integer.parseInt(oldday);
for(int days= 1; days<day+1; days++){

rec.Welcome(name, days);
if(days == day){
   JOptionPane.showMessageDialog(null, "Today is the last day,I will miss you "+name);
}
```

```
if(days == 1){
   dailyWeight = currentWeight;
}
else{
   dailyWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your
                 weight (kg) for day " + days + " ? "));
   while (dailyWeight >= 200.0 || dailyWeight <= 0.0) {
   JOptionPane.showMessageDialog(null,
         "Your weight isn't suitable for this Smart Diet Management.",
         "False input.",
         0);
   dailyWeight = Double.parseDouble(JOptionPane.showInputDialog("What's your
                 weight (kg) today? "));
   }
}

todayweight.add(dailyWeight);

double bmrStandard = ((10 * dailyWeight) + (6.25 * heightCm) - (5 * age));
//calculate the standard bmr before addition or subtraction based on the gender.

boolean genderCheck = gender.equals("Male");
double bmrGender;
if (genderCheck == true) {
   bmrGender = (bmrStandard + 5);
} else {
```

```
    bmrGender = (bmrStandard - 161);
}
//+5 if gender is male, -161 if gender is female.
//Check variable "gender" string content based on user input and boolean statement for
    T/F result.

double bmrExerciseFrequency = 0;
switch (exerciseFrequency) {
    case "0":
        bmrExerciseFrequency = (bmrGender * 1.2);
        break;

    case "1-3":
        bmrExerciseFrequency = (bmrGender * 1.375);
        break;

    case "4-5":
        bmrExerciseFrequency = (bmrGender * 1.55);
        break;

    case "6-7":
        bmrExerciseFrequency = (bmrGender * 1.725);
        break;

    case "more than 7":
        bmrExerciseFrequency = (bmrGender * 1.9);
```

```
        break;
    }
    //using switch statement to compare user input for variable "exerciseFrequency"
    //then multiply it with variable "bmrGender" with corresponding value for each exercise
      frequency

    double bmrGoal;
    if (goalCheck == true) {
        bmrGoal = (bmrExerciseFrequency + 500);
    } else {
        bmrGoal = (bmrExerciseFrequency - 500);
    }
    //+500 if variable "goal" = "Gain weight". Check using boolean, add 500 by using if else
        statement

    int bmrFinal = (int) bmrGoal;

    //display user daily calorie intake based on their gender, exerciseFrequency, and goal.

    BMI calc = new BMI();

    calc.PrintBMR(name,bmrFinal);

    JOptionPane.showMessageDialog(null,
            "Your calculated BMI is " + calc.findBMI(dailyWeight, heightM) + ".",
            "",
```

```
        2);

double finalbmi = calc.findBMI(dailyWeight, heightM);

JOptionPane.showMessageDialog(null,
        "Your BMI status shows that you are " + calc.bmiStatus(finalbmi),
        "",
        2);

rec.PrintRecipe(bmrGoal,name,days);
//If else statement and a range of values for the calorie intake
//matching the calorie intake we will print out the daily meal plan
//using a for loop we will keep on print out 5 days worth of meal plan, day by day.

double total = 0.0;

CaloriesConsume cal = new CaloriesConsume(); //other class

for (int meal = 0; meal < 10; meal++) {
    double InCal = cal.GetCalories();   //go to 'cal' class, execute GetCalories method
    if (InCal > 0) {
        total = cal.GetSum(InCal);  //go to 'cal' class, execute GetSum method by inputting
        InCal
    }
    if(InCal < 0){
        JOptionPane.showMessageDialog(null, "Calorie input is invalid.");
```

```
    }
    String resume = cal.NeedCon(); //go to 'cal' class, execute NeedCon method
    if (resume.equals("No")) {
        meal = 10;
    }
}

JOptionPane.showMessageDialog(null, "The total calories consumed today is " + total+"
 kcal.");
cal.printMess(total, bmrFinal, goalCheck); //go to 'cal' class, execute printMess method
}
rec.summary(day,todayweight,goalWeight,goal);

if(todayweight.get(day-1) == goalWeight){
  JOptionPane.showMessageDialog(null,"Damn, you achieved your goal weight spot on!
    Kang Wei is shook right now :0");
}
else if((todayweight.get(day-1) < goalWeight && goalCheck == true) ||
 (todayweight.get(day-1) > goalWeight && goalCheck == false)){
  JOptionPane.showMessageDialog(null,"Unfortunately, you have failed your diet plan.
 Kang Wei is crying at the corner right now.");
}
else{
  JOptionPane.showMessageDialog(null,"Congratulations, you have succeeded with your
  diet plan! Kang Wei will buy you a meal.");
}
```

```
        JOptionPane.showMessageDialog(null,
                "Thank you so much for using our Kang Wei's Smart Diet Management System.");
    }
}
```

Appendix B: BMI

```
package sdm.input;


import javax.swing.JOptionPane;


public class BMI {


    public double bmi;
    public String status;


    public double findBMI(double dailyWeight, double height) {
        bmi = (dailyWeight / Math.pow(height, 2));
        bmi = Math.round(bmi * 100.0) / 100.0;
        return bmi;
    }


    public String bmiStatus(double a) {
        if (a < 18.5) {
            status = "underweight";
        } else if (a >= 18.5 && a <= 24.9) {
            status = "normal";
        } else if (a >= 25 && a <= 29.9) {
            status = "overweight";
        } else if (a >= 30 && a <= 34.9) {
            status = "obese";
```

```
        } else if (a >= 35) {
            status = "extremely obese";
        }
        return status;
    }

    public void PrintBMR(String name,int bmr){
        JOptionPane.showMessageDialog(
                null,
                name + " we have calculated your daily calorie intake based on the information you have provided and it is " + bmr + " calories",
                "Daily calorie intake",
                2);
    }
}
```

Appendix C: CaloriesConsume

```
package sdm.input;


import javax.swing.JOptionPane;


public class CaloriesConsume {

   public double sum = 0.0;

   public String message1 = ("Unfortunately you didn't meet the requirement,you need to
                              consume more calories.\nTry harder for tomorrow.");

   public String message2 = ("Congratulation! you have met the requirement!!!\nContinue the
                              good work!");

   public String message3 = ("You have exceed the recommended calories you need to
                              consume less calories.\nTry to reduce you food intake!!!");

   public String message4 = ("Meh nothing changed\nGive me some better result!");


   public double GetCalories() {

      double Inputcalories1 = Double.parseDouble ((String)
                              JOptionPane.showInputDialog(null,

            "Enter the amount of calories for the meal.",

             "Input Calories",

             3,

             null,

             null,

             "Enter your calories here."));

      while(Inputcalories1<0 || Inputcalories1>6000){
```

```
        Inputcalories1 = Integer.parseInt(JOptionPane.showInputDialog("Your calories is invalid.\nPlease enter again."));
        }
        return Inputcalories1;
    }

    public double GetSum(double input) {
        sum = sum + input;
        return sum;
    }

    public String NeedCon() {
        String[] anws = {"No", "Yes"};
        String choice = (String) JOptionPane.showInputDialog(null,
                "Still got any calories intake left?",
                "Calorie input?",
                2,
                null,
                anws,
                anws[0]);
        return choice;
    }

    public void printMess(double total, double limit, boolean goal) {
        if (total < limit && goal == true) {
```

```
            JOptionPane.showMessageDialog(null, message1);

        } else if (total > limit && goal == false) {
            JOptionPane.showMessageDialog(null, message3);

        } else if (total == limit) {
            JOptionPane.showMessageDialog(null, message2);

        } else if (total > limit && goal == true && total <= (limit + 300)) {
            JOptionPane.showMessageDialog(null, message2);

        } else if (total < limit && goal == false && total >= (limit - 300)) {
            JOptionPane.showMessageDialog(null, message2);

        } else {
            JOptionPane.showMessageDialog(null, "Your calories consumption is
            unhealthy.\nPlease make some changes tommorrow.");
        }
    }
}
```

Appendix D: Recipe

```
package sdm.input;

import java.util.ArrayList;

import javax.swing.JOptionPane;


public class Recipe {

    public String a =("Recipe 1:\nBreakfast: Avocado & Arugula Omelet\nLunch: Korean crab rolls\nDinner: Turkey Taco Lettuce Wrap\n"

    +"Recipe 2:\nBreakfast: Apple-Cinnamon Quinoa Bowl\nLunch: Italian vegetable hoagies\nDinner: Salmon with Roasted Vegetables & Quinoa\n"

    +"Recipe 3:\nBreakfast: Huevos Rancheros Tacos\nLunch: Mediterranean Tuna Antipasto Salad\nDinner: Soy-Marinated Flank Steak with Soba Noodles\n"

    +"Recipe 4:\nBreakfast: Strawberry-Ricotta Waffle Sandwich\nLunch: Edamame & Chicken Greek Salad\nDinner: Orange&Pistachio Crusted Pork Tenderloin\n"

    +"Recipe 5:\nBreakfast: Baked Banana-Nut Oatmeal Cups\nLunch: Creamy Pesto Chicken Salad with Greens\nDinner: Shrimp, Ham & Pepper Couscou\n");


    public String b =("Recipe 1:\nBreakfast: Herbed ricotta & tomato toast\nLunch: Shaved carrot and radish salad\nDinner: Garlicky fried rice with crispy pork\n"

    +"Recipe 2:\nBreakfast: Yoghurt Parfait\nLunch: Super green mushroom & Orzo soup\nDinner: Green tea noodles with Sticky sweet chili salmon\n"

    +"Recipe 3:\nBreakfast: Apple vanilla greek yoghurt pancakes\nLunch: Vegan Caesar salad\nDinner: Italian sausage stuffed zucchini\n"

    +"Recipe 4:\nBreakfast: Smoked salmon cucumber wraps\nLunch: Tomato Panzanella\nDinner: Pan-fried tilapia\n"

    +"Recipe 5:\nBreakfast: Ham, Egg, and Avocado burrito\nLunch: Grilled leek potato salad\nDinner: Chicken Parm stuffed peppers\n");
```

```
public String c =("Recipe 1:\nBreakfast: Steel cut oatmeal\nLunch: Spinach and Gruyere potato casserole\nDinner: Caprese zoodles\n"

+"Recipe 2:\nBreakfast: Chocolate chip oatmeal cookie smoothie\nLunch: Roasted cauliflower pizza\nDinner: Taco tomatoes\n"

+"Recipe 3:\nBreakfast: Cardamom & peach quinoa porridge\nLunch: Tomato, peach and basil salad\nDinner: Skinny Alfredo\n"

+"Recipe 4:\nBreakfast: Pistachio nut & spiced apple bicher muesli\nLunch: Mediterranean cod\nDinner: Blackened tilapia\n"

+"Recipe 5:\nBreakfast: Creamy mustard mushrooms on toast\nLunch: Tofu pad Thai\nDinner: Tuscan butter roast chicken\n");


public String d =("Recipe 1:\nBreakfast: Welsh rarebit muffins\nLunch: Coconut rice and peas\nDinner: Primavera stuffed chicken\n"

+"Recipe 2:\nBreakfast: Hash browns with mustard & smoked salmon\nLunch: Creamy lemon pasta with chicken\nDinner: Honey walnut shrimp\n"

+"Recipe 3:\nBreakfast: Waffle with spinach, tomato & feta\nLunch: Poke bowl\nDinner: Zucchini lattice lasagna\n"

+"Recipey 4:\nBreakfast: Avocado toast with banana\nLunch: Vietnamese caramel pork with pickled carrot salad\nDinner: Cauliflower grilled cheese\n"

+"Recipe 5:\nBreakfast: Blueberry-banana-nut smoothie\nLunch: Crispy tofu bowl\nDinner: Cilantro lime salmon bowls\n");


public String e =("Recipe 1:\nBreakfast: Shakshuka\nLunch: Kale and roasted cauliflower salad\nDinner: Lasagna stuffed zucchini\n"

+"Recipe 2:\nBreakfast: Mushroom & zucchini quiche\nLunch: Korean pineapple beef lettuce wraps\nDinner: Zucchini lasagna roll-ups\n"

+"Recipe 3:\nBreakfast: Sweet potato and kale frittata\nLunch: Summer pesto pasta\nDinner: Chicken taco avocado\n"
```

```
+"Recipe 4:\nBreakfast: Chilled overnight chia\nLunch: Fried avocado tacos\nDinner: Maple rosemary pork tenderloin\n"

+"Recipe 5:\nBreakfast: Curry-avocado egg toast\nLunch: Sriracha meatball hoagies\nDinner: Vegan pizza\n");


public String f =("Recipe 1:\nBreakfast: Spiced plum & quinoa muffins\nLunch: Feta-dill greek Caesar salad\nDinner: Loaded cauliflower bake\n"

+"Recipe 2:\nBreakfast: Tropical smoothie bowl\nLunch: Prosciutto-melon panini\nDinner: Chicken adobo\n"

+"Recipe 3:\nBreakfast: Mexican chilaquiles\nLunch: Buffalo chicken Cobb salad\nDinner: Blackened shrimp bowls\n"

+"Recipe 4:\nBreakfast: Crustless quiche Lorraine\nLunch: Spicy tuna sandwiches\nDinner: Garlicky shrimp zucchini pasta\n"

+"Recipe 5:\nBreakfast: Cherry berry oats\nLunch: Thai beef and veggie stir-fry\nDinner: Chicken teriyaki pineapple bowls\n");


public void PrintRecipe(double bmr,String name,int day){

   if (day == 1){

   JOptionPane.showMessageDialog(null,"We recommend you to try these recipes for your diet plan.");


   if ((bmr >= 1000) && (bmr < 1500))

     JOptionPane.showMessageDialog(null,a);


   else if ((bmr >= 1500) && (bmr < 2000))

       JOptionPane.showMessageDialog(null,a);


   else if ((bmr >= 2000) && (bmr < 2500))
```

```
        JOptionPane.showMessageDialog(null,b);

    else if ((bmr >= 2500) && (bmr < 3000))
        JOptionPane.showMessageDialog(null,c);

    else if ((bmr >= 3000) && (bmr < 3500))
        JOptionPane.showMessageDialog(null,d);

     else if ((bmr >= 3500) && (bmr < 4000))
        JOptionPane.showMessageDialog(null,e);

     else
        JOptionPane.showMessageDialog(null, name + ", we are not capable of
        recommending you meal plans for your calorie intake. Please seek a dietician for
        help.");
    }
}
public void summary(int day,ArrayList todayweight,double goalWeight,String goal){
    JOptionPane.showMessageDialog(null,
        "The weight for "+day+" days:"+" " +todayweight+"(kg)"
              +"\n Goal weight: "+goalWeight+"\nGoal: "+ goal
        ,"Summary"
        ,3);
}

public void Welcome(String name,int days){
```

```
        JOptionPane.showMessageDialog(null,
                "Hello " + name + " welcome to: \n DAY "+days,
                "DAY" +days,
                3);

    }
}
```