Dayna Eidle

Professor Pablo Rivas

Software Development I

11 May 2017

Project Milestone

Abstract

The app I designed is called GetFit!. It is an easy-to-use caloric intake calculator that also tells you the status of your current weight. Although the final product is different from what I had originally hoped the app would turn out like, I am proud with the way it turned out and that I was able to incorporate things like a JFrame window even though I had never used it before. It is simple in that the user only needs to type in the requested information in order for the program to calculate how many calories they should be eating in a given day. It uses one file with several methods included in the code. Aside from my previous knowledge, there are a list of references at the end that I used to help me learn how to code the parts that I had never done before.

Introduction

The GetFit! app is designed to allow users to input information about themselves and then see nutritional information, like how many calories they should be eating each day and their weight status. The idea extremely conscious about what I eat and how much of everything I eat and so I thought it would be useful to create an app that allows

users to see how many calories they should be eating and what their current weight status is. I also like when apps have a nicer look to them and so I wanted to challenge myself to make part of the app in a graphical window.

I will go on to first discuss how the app works and the basics of the design of the program. Then, I will move into the requirements of the system and go into other apps that are similar to mine and the similarities and differences between them. There will be a description of how to use the app and what information it generates. Then, a conclusion will tie everything together and I will have a list of references I used for my research.

Detailed System Description

The GetFit! app is an easy to use nutritional program that calculates how many calories the user should eat over the course of a day and what their current weight status is. The user must first input their name, age, gender, current weight, and height and then choose how active they are on a daily basis and finally, choose if they'd like to lose weight, gain weight, or stay the same weight. Once they input all of their information, the amount of calories they should be eating each day and the status of their current weight (underweight, normal weight, overweight, or obese) appear in a separate graphical window. To exit the program, the user then just has to hit enter.

The program itself is made up of elements within the main method and some methods that are outside of the main. The paint method outside the main is the component of the code that creates the JFrame window. Then, there are three other

methods — calcFemaleRMR, calcMaleRMR, and calcBMI — all of which are used in other parts of the code to calculate the caloric intake and weight status. For various parts of the code, it was necessary to import the javax.swing class, the awt class, the Scanner class, and the Math class. (UML Diagram on page 6)

Requirements

The code itself for the app is not too complex. There is one class that contains methods outside of the main method. There are several global variables involved in the code including double rmr, boolean counter, String BMI, and String name. Outside of the main method are the methods used for calculating BMI and resting metabolic rate in the program. The program has some objects needed for various parts. A scanner object was created so that the user could input their information and have it be stored in variables. There is also a JFrame object so that part of the program could be displayed in a graphical window.

Literature Survey

There are plenty of other fitness apps out there from fitness trackers to calorie counters to workout apps. I chose the features that I did because they were things that I take the most interest in and that I also knew I would be able to program but would still have some difficulty with. With some other apps, if it has to deal with calories, it will usually tell you how many calories you should be eating each day and then allow the user to input what they are eating each day so that they can keep track of how many

calories they are eating. To do that, it would have involved me creating a database for the user's information. That was part of my original plan, however, after I started the project, I determined that with having no previous knowledge of databases, I wouldn't have enough time to incorporate it into my program. Most calorie counting programs wouldn't tell the user what their weight status is and so that is a feature that my program has that is original.

User Manual

The system has a very basic design. To use the app, the user must first launch it in a terminal window. Upon launching, an intro logo and instructions in both the terminal window as well as a JFrame window show up. The user then has to input their information — name, age, gender, current weight, and height — and say how active they are on a daily basis and if they want to lose weight, gain weight, or stay the same weight. With this portion of the app, it is helpful if the user inputs the correct data type for each given field. For example, if the program asks for the user's height in inches and they put in a letter then the program will crash. Also, if the user inputs value that are too high, the program will overload during runtime.

After the user inputs their information, their caloric intake and weight status will appear in the JFrame window next to the terminal window. After viewing the information, to exit the program, all the user has to do is hit enter.

My goal for the app was to make sure it had a simple and easy-to-use design so that wasn't too much effort to figure out how to use it. I feel like I achieved that by

making it so minimal instructions were needed and that there weren't too many things the user had to do to use the program.

Conclusion

In the process of designing and programming this app, it took many unexpected turns. It was originally supposed to be a fitness tracker that lets you store your weight progress and count calories. That turned out to be a little too ambitious and so that is why I decided to make it just a caloric intake calculator. I figured that wasn't challenging enough, though, and that is why I decided to program it to show up in a graphical JFrame window. The JFrame window proved to be my biggest challenge. Upon beginning this project, I had never worked with JFrames before. Because of this, I had to do a lot of research and figure out a good portion of the code on my own. It posed as an obstacle a few times, but in the end, I got everything to come together and work the way I wanted it to. The JFrame was something that seemed necessary to me because I wanted the app to have an extra layer to it so that it wasn't so simple. Although it wasn't the most fancy layout, it made it nicer to look at so that it wasn't just in the terminal window.

All in all, I was proud with the product I produced. It showed me that I am capable of a lot more than I think I am and am able to work out problems that I have with my code efficiently. My hope is that one day I will be able to add onto this program and make it into a fully functional fitness tracker app.

UML Diagram

References/Bibliography

Frey, Malia. "How Many Calories Should I Eat to Lose Weight?" *Verywell*. About Inc., 15 Mar. 2017. Web. 05 Apr. 2017.

"Graphics: drawString()." *Java2S*. Demo Source and Support, n.d. Web. 06 May 2017.

"How to Make Frames (Main Windows)." *How to Make Frames (Main Windows) (The Java™ Tutorials Creating a GUI With JFC/Swing Using Swing Components)*. Oracle, 2015. Web. 09 May 2017.

Magnuson, Bernadene. "How to Calculate Your Body Mass Index (BMI)." *EtoxNet*. University of Idaho, 1998. Web. 04 May 2017.

"Resting Metabolic Rate (RMR) Calculation." *WpCalc*. Wow-Company, 12 Mar. 2016. Web. 05 Apr. 2017.