PHYSICAL HEALTH MONITORING : (3-DAYS) TRACK PROGRAM

A Final Project Presented to the
Faculty of the
Information and Communication Technology
STI College of Davao

In Partial Fulfillment
Of the Requirements for the Degree of Bachelor
Of Science in Information Technology

Tindoy Rhammzthienne Rurac
Christian Lloyd D. Pantinople
Marchelle Chaelyn Nerosa
Raniel Josh Larrobis
Almar M. Bacus

December 2024

EXECUTIVE SUMMARY

Project Context

Physical Health monitoring programs are becoming essential in today's society, as maintaining a healthy lifestyle is key to preventing chronic diseases and enhancing overall well-being. This project demonstrates the use of Java programming to create a physical health monitoring program that collects and evaluates vital health information from users. The program focuses on three key metrics: daily step count, sleep duration, and heart rate in beats per minute (BPM). It analyzes the data to provide feedback and recommendations when health indicators, such as heart rate and sleep hours fall outside the normal range.

The Physical Health Monitoring 3-Day Track Program, was programmed using Visual Studio Code (VS Code), because of its lightweight, quick, and highly configurable environment. Writing, testing, and debugging programs is made easier with Visual Studio Code's intelligent code completion and built-in debugging tools. The application may be built and tested on a variety of operating systems thanks to its cross-platform compatibility, and commands can be executed with ease thanks to the integrated terminal. Because of these characteristics, VS Code is the best option for effectively creating and managing the health tracking application. To sum up, VS Code was selected for the Physical Health Monitoring 3-Day Track Program due to its quickness, adaptability, user-friendliness, and capacity to effectively assist the development process.

It is the perfect option for creating an intuitive and useful health tracking program because of its strong features, which include intelligent code suggestions, debugging tools, Git integration, and cross-platform compatibility. VS Code guarantees clean code, increases productivity, and frees us up to concentrate on the program's primary purpose of assisting users in tracking their health over a three-day period.

TABLE OF CONTENTS

Page

INTRODUCTION

Project Context

In today's society, when leading a healthy lifestyle is critical for preventing chronic diseases and enhancing general well-being, physical health monitoring programs have grown in importance. This program demonstrates the utilization of Java programming in the creation of a physical health monitoring program. With a particular focus on three key health metrics: daily step count, sleep duration, and heart rate in beats per minute (BPM). The Physical Health Monitoring Program is a quick and easy tool to monitor your health over the following three days.

According to Baig & Gholamhosseini, Over the past 20 years, Physical health monitoring systems have advanced quickly and have the potential to transform the current delivery of healthcare. We will first ask you for some basic information in order to get started. Then, for the next three days, we'll ask you to enter your heart rate, sleep duration, and steps. Following each submission, you will receive a brief assessment to determine whether you are on track or need to make adjustments. We will request your height and weight on the last day in order to determine your BMI and provide beneficial health advice based on your findings. It all comes down to providing you with the means to better understand your body and make minor adjustments that can have a significant effect. After completing your three-day health tracking, you can also decide to set up a medication reminder to assist you remember to take any prescribed medications. At the end of the process, we will ask if you would like to retake the test. If you want to continue tracking your health, we'll restart.

In addition, you can choose to create a medicine reminder after finishing your three-day health tracking, which will help you remember to take any necessary medical treatments. We will ask if you would like to retake the test at the conclusion of the procedure. We'll restart if you would like to keep tracking your health. We'll acknowledge your dedication to enhancing your health and thank you for utilizing the program after you're finished.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

In conclusion, by concentrating on important indicators like heart rate, sleep length, and daily step count, the Physical Health Monitoring Program provides a useful tool for people looking to monitor and enhance their health. The application gives users a simple and convenient way to track their health over a three-day period by utilizing Java programming. The application assists users in assessing their progress, making required modifications, and even calculating their BMI for more health insights through routine evaluations. Additionally, the ability to schedule medicine reminders demonstrates the program's dedication to promoting general health. By enabling users to retake the test and continue monitoring their progress, this tool not only helps people understand their health better but also promotes ongoing participation.

The program collects user information and tracks five health categories: steps taken, sleep hours, heart rate, and on the final day, height and weight. It calculates BMI based on height and weight, provides daily feedback on whether the user's metrics are healthy, and offers health tips based on the BMI. Users can also set and manage medication reminders. After the 3-day tracking period, the program asks if the user wants to retake the test, and either restarts or ends with a thank you message.

Recommendations

The Java program for the 3-day Physical Health Monitor is a useful tool for tracking health metrics, but it could benefit from several enhancements. Implementing input validation would help ensure that users provide accurate data, particularly for age, height, and weight. Additionally, breaking the code into smaller, reusable methods would improve readability and maintainability. According to S. Asthana, et al., A person's well-being could be better maintained and major diseases could be avoided with proactive physical health monitoring. Numerous wearable technological gadgets have been developed to track and measure various health parameters in the current Internet of Things (IoT) environment.

APPENDICES

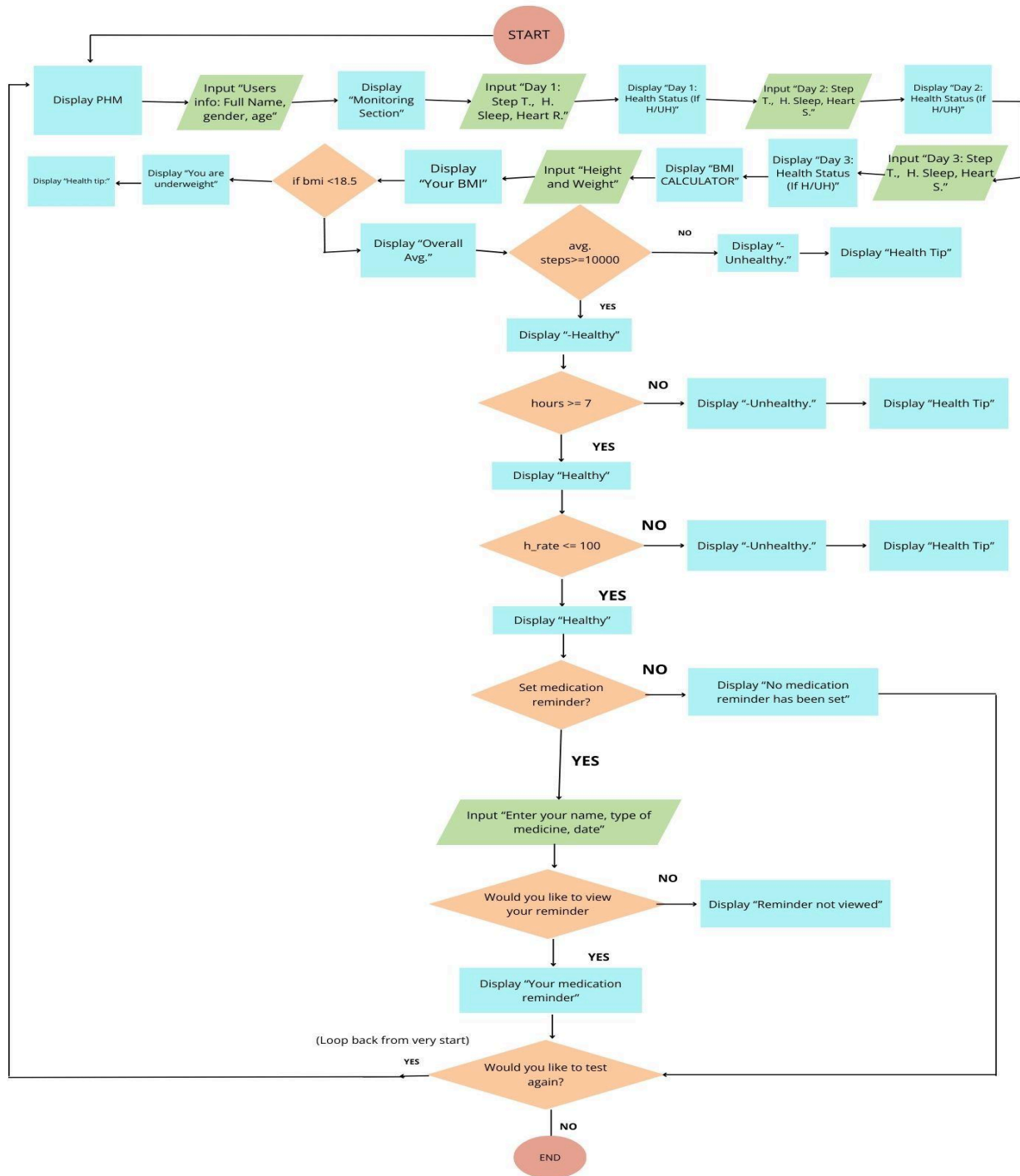The following pages contain the diagrams, references, and sample input/output.

Figure 1. Flow Chart

REFERENCES

Baig, M.M., Gholamhosseini, H. Smart Health Monitoring Systems: An Overview of Design and Modeling.

https://link.springer.com/article/10.1007/s10916-012-9898-z#citeas

S. Asthana, A. Megahed and R. Strong, "A Recommendation System for Proactive Health Monitoring Using IoT and Wearable Technologies," 2017 IEEE International Conference on AI & Mobile Services (AIMS), Honolulu, HI, USA, 2017, pp.14-21,doi:10.1109/AIMS.2017.11.keywords:{Monitoring;Biomedicalmonitoring;Disease;Wearablesensors;Temperaturesensors;IoT;Healthcare;Monitoring;WearableTechnologies;Analytics;Text Mining;Optimization},

https://ieeexplore.ieee.org/abstract/document/8027264

Input 1: Enter Personal Information , First name, Last name, Gender, & Age.

```
=== Welcome to the 3-day Physical Health Monitor ===
First name: Ram
Last name: Skibidi
Gender (Male/Female): Male
Age: 18
```

Output 2: Display the User Information Data.

```
=== User Information ===
Full name: Ram Skibidi
Gender: Male
Age: 18
```

Input 3: Proceed to monitoring section , The program asks to input steps taken, hours of sleep, & average heart rate for Day 1.

```
=== Monitoring Section ===

Day 1:

Enter steps taken: 10000
Enter hours of sleep: 7
Enter average heart rate (bpm): 90
```

Output 4: Display the Health Status for Day 1. If it's Healthy or Unhealthy.

```
Health Status for Day 1:
Steps: 10000 (Healthy)
Sleep: 7 hours (Healthy)
Heart Rate: 90 bpm (Healthy)
```

Input 5: Next the program asks the same question but for day 2.

```
=== Monitoring Section ===

Day 2:

Enter steps taken: 900
Enter hours of sleep: 5
Enter average heart rate (bpm): 120
```

Output 6: Display the Health Status for day 2.

```
Health Status for Day 2:
Steps: 900 (Unhealthy)
Sleep: 5 hours (Unhealthy)
Heart Rate: 120 bpm (Unhealthy)
```

Input 7: Lastly, The program asks the same question for day 3.

```
=== Monitoring Section ===

Day 3:


Enter steps taken: 11000
Enter hours of sleep: 8
Enter average heart rate (bpm): 95
```

Output 8: The program displays the health status for day 3.

```
Health Status for Day 3:
Steps: 11000 (Healthy)
Sleep: 8 hours (Healthy)
Heart Rate: 95 bpm (Healthy)
```

Input 9: After that, the program will proceed to ask the user for the user's height & weight to calculate the user's BMI at the last day of tracking.

```
=== BMI Calculator ===

Enter your height in meters (e.g., 1.75): 1.79
Enter your weight in kg (e.g., 70.5): 50
```

Output 10: The program will display the result of BMI and give a direct feedback of it.

```
=================
Your BMI is: 15.6
=================

== You are underweight. ==
Health Tip: Consider incorporating nutrient-dense foods into your diet,
such as lean proteins, whole grains, fruits, and vegetables to help reach a healthy weight.
```

Output 11 : The program displays the 3-day average health summary.

```
=== 3-Day Health Summary ===
Average Steps: 7300
Average Sleep: 6hours
Average Heart Rate: 101BPM
```

Output 12 : Display the overall average of Steps status and Display also the label if "Healthy", or "Unhealthy".

```
Overall Average:
Overall Steps status: Unhealthy!
Health Tip: Aim to increase your daily steps
```

Output 13 : Display the overall average of Sleep status and Display also the label if "Healthy", or "Unhealthy". And also give Health Tips if the result is Unhealthy.

```
Overall Sleep status: Unhealthy!
Health Tip: Consider increasing your sleep to 7-9 hours per night
better health.
```

Output 14 : Display the overall average of Heart rate status and Display also the label if "Healthy", or "Unhealthy". And also give Health Tips if the result is Unhealthy.

```
Overall Heart Rate status: Unhealthy!

Your heart rate seems high. Consider consulting a healthcare provider.
```

Input 15 : After that, the program will ask the user if he wants to set medication reminder.

```
Would you like to set a medication reminder? (yes/no)yes
```

Input 16 : Next, ask the user to enter what type of medicine and what day.

```
Please enter the name/type of your medicine:Bioflu
Please enter the date for the reminder (e.g., YYYY-MM-DD):
2024-01-15
```

Output 17 : The program displays the user type of medicine and schedule of the day.

```
=== Medication Reminder Set!===


Medicine: Bioflu
Reminder Date: 2024-01-15
```

Input 18 : Next, The program asks the user if the user would like to view the medicine reminder.

```
Would you like to view your reminder? (yes/no)
yes
```

Output 19: The program displays the user Medication Reminder.

```
Your Medication Reminder:
Medicine: Bioflu
Reminder Date: 2024-01-15
```

Input 20 : Lastly, The program asks the user if the user wants to test again/ try again to monitor the user's physical health.

```
Do you want to test again? (yes/no): yes
```

Output 21 : If the user chooses to test again the program will loop back from the very start to test/monitor new physical health data.

```
=== Welcome to the 3-day Physical Health Monitor ===
First name: █
```

Output 22 : If the user doesn't want to test again the program displays appreciation of using the program itself to monitor and track the user's physical health in 3-days.

```
Do you want to test again? (yes/no): no█
```

```java
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Scanner;

public class aynagComment {

        public static void main(String[] args) {

    boolean again = true;

    while (again){

      Scanner scanner  = new Scanner (System.in);

        System.out.println("\n=== Welcome to the 3-day Physical Health Monitor ===");
        System.out.print("First name: ");
        String f_name= scanner.nextLine(); // Store first name
        System.out.print("Last name: ");
        String s_name = scanner.nextLine(); // Store last name

        // Collect gender and age
        System.out.print("Gender (Male/Female): ");
        String genders = scanner.nextLine(); // Store gender
        System.out.print("Age: ");
        int aged = scanner.nextInt(); // Store age

        // Consume the leftover newline character
        scanner.nextLine();  // This is to clear the buffer after reading the integer
        //Display user information
        System.out.println("\n=== User Information ===");
        System.out.println("Full name: " +f_name + " " + s_name);
        System.out.println("Gender: " + genders);
        System.out.println("Age: " + aged);

      //Initialize arrays for 3-day tracking
      int[] stepst = new int [3];
```

```java
int[] s_hours = new int [3];
int [] h_rate = new int [3];

//collective data
for (int day = 0;  day < 3; day++) {
    System.out.println("\n=== Monitoring Section ===");
    System.out.println("\nDay " + (day +1) + ":");

    System.out.print("\nEnter steps taken: ");//q1
    stepst[day] = scanner.nextInt();

    System.out.print("Enter hours of sleep: ");//q2
    s_hours[day] = scanner.nextInt();

    System.out.print("Enter average heart rate (bpm): ");//q3
    h_rate[day] = scanner.nextInt();

    // Evaluate Health based on the input for each day
    System.out.println("\nHealth Status for Day " + (day + 1) + ":");

    // Check steps
    if (stepst[day] >= 10000) {
        System.out.println("Steps: " + stepst[day] + " (Healthy)");
    } else {
        System.out.println("Steps: " + stepst[day] + " (Unhealthy)");
    }

    // Check sleep
    if (s_hours[day] >= 7 && s_hours[day] <= 9) {
        System.out.println("Sleep: " + s_hours[day] + " hours (Healthy)");
    } else {
        System.out.println("Sleep: " + s_hours[day] + " hours (Unhealthy)");
    }

    // Check heart rate
    if (h_rate[day] >= 60 && h_rate[day] <= 100) {
        System.out.println("Heart Rate: " + h_rate[day] + " bpm (Healthy)");
    } else {
        System.out.println("Heart Rate: " + h_rate[day] + " bpm (Unhealthy)");
    }
```

```java
    }
    //Collect users data
    System.out.println("\n=== BMI Calculator ===");
    System.out.print("\nEnter your height in meters (e.g., 1.75): ");//q4
    double height = scanner.nextDouble();
    System.out.print("Enter your weight in kg (e.g., 70.5): ");//q5
    double weight = scanner.nextDouble();

     // Calculate BMI
    double bmi = weight / (height * height);
    DecimalFormat df = new DecimalFormat("#.0");  // Format to 1 decimal place

    // Output the BMI result
    System.out.println("\n================");
    System.out.println("Your BMI is: " + df.format(bmi));
    System.out.println("================");

    // BMI Categories
    // Check the BMI category and provide a health tip
    if (bmi < 18.5) {
        System.out.println("\n== You are underweight. ==");
        System.out.println("Health Tip: Consider incorporating nutrient-dense foods into your
diet,");
        System.out.println("such as lean proteins, whole grains, fruits, and vegetables to help
reach a healthy weight.");
    } else if (bmi < 24.9) {
        System.out.println("\n== You have a normal weight. ==");
        System.out.println("Health Tip: Keep up the good work! Continue maintaining a
balanced diet and regular physical activity");
        System.out.println("to stay healthy.");
    } else if (bmi < 29.9) {
        System.out.println("\n== You are overweight. ==");
        System.out.println("Health Tip: Focus on a balanced diet with a mix of whole foods
and exercise regularly");
        System.out.println(" to help achieve a healthier weight.");
    } else {
        System.out.println("\n== You are obese. ==");
        System.out.println("Health Tip: It's important to consult with a healthcare professional
for a tailored plan,");
```

```java
        System.out.println("which may include diet changes and physical activity, to work
toward a healthier weight.");
    }

    System.out.println("\n----------------------------------------------------------");
    // Calculate weekly averages and calories burned
    double avgSteps = 0;
    double avgSleep = 0;
    double avgHeartRate = 0;

    // Sum values for each array
    for (int i = 0; i < stepst.length; i++) {
        avgSteps += stepst[i];   // Sum steps
        avgSleep += s_hours[i];    // Sum sleep hours
        avgHeartRate += h_rate[i]; // Sum heart rate
    }

    // Calculate the averages (since there are 3 days)
    avgSteps /= stepst.length;      // Average steps
    avgSleep /= s_hours.length;        // Average sleep
    avgHeartRate /= h_rate.length; // Average heart rate


    // Display 3-days summary
    System.out.println("\n=== 3-Day Health Summary ===");
    System.out.println("Average Steps: " + (int) avgSteps);
    System.out.println("Average Sleep: " + (int) avgSleep + "hours");
    System.out.println("Average Heart Rate: " + (int) avgHeartRate + "BPM");

    // Overall health status
    System.out.println("\nOverall Average:");
    if (avgSteps >= 10000) {
        System.out.println("Overall Steps status: Healthy");
    }
    else {
        System.out.println("Overall Steps status: Unhealthy!");
        System.out.println("Health Tip: Aim to increase your daily steps");
    }

    if (avgSleep >= 7) {
```

```java
            System.out.println("\nOverall Sleep status: Healthy");
        }
        else {
            System.out.println("\nOverall Sleep status: Unhealthy!");
            System.out.println("Health Tip: Consider increasing your sleep to 7-9 hours per
night");
            System.out.println("better health.");

        }
        if (avgHeartRate < 100) {
            System.out.println("\nOverall Heart Rate status: Healthy");
        }
        else {
            System.out.println("\nOverall Heart Rate status: Unhealthy!");
            System.out.println("\nYour heart rate seems high. Consider consulting a healthcare
provider.");
        }
        scanner.nextLine();

        // Ask the user if they want a medication reminder
        System.out.print("\nWould you like to set a medication reminder? (yes/no)");
        String response = scanner.nextLine().toLowerCase();

        if (response.equals("yes")) {
            // Ask for the medication details
            System.out.print("\nPlease enter the name/type of your medicine:");
            String medicineName = scanner.nextLine();

            System.out.println("Please enter the date for the reminder (e.g., YYYY-MM-DD):");
            String reminderDate = scanner.nextLine();

            // Confirm and display the reminder
            System.out.println("\n=== Medication Reminder Set!===");
            System.out.println("\nMedicine: " + medicineName);
            System.out.println("Reminder Date: " + reminderDate);

            // Ask if the user wants to view the reminder
            System.out.println("\nWould you like to view your reminder? (yes/no)");
            String viewReminder = scanner.nextLine().toLowerCase();
```

```java
            if (viewReminder.equals("yes")) {
                // Display the stored reminder
                System.out.println("\nYour Medication Reminder:");
                System.out.println("Medicine: " + medicineName);
                System.out.println("Reminder Date: " + reminderDate);
            } else {
                System.out.println("!!! Reminder not viewed. !!!");
            }
        }
        else {
            System.out.println("! No medication reminder has been set. !");
        }

        System.out.print("\nDo you want to test again? (yes/no): ");
        String test = scanner.nextLine();
        if (test.contains("yes")){
            again = true;
        }
        else {
            again = false;
            System.out.println("\nGreat job taking charge of your health and using the program to
track your progress!");

        }
    }
  }
}
```