Manav Rachna International Institute of Research and Studies

SCHOOL OF COMPUTER APPLICATION

MCA - B 1st SEM

PROJECT REPORT

(Course Code: 6.0CA102C01H)

SUBMITTED BY:

TINKU JAIN 25/SCA/MCAN/071

ABHISHEK GUPTA 25/SCA/MCAN/072

SHIVAM KUMAR 25/SCA/MCAN/ 073

SUBMITTED TO:

Dr. Ritu Sachdeva

∇ HEALTH TRACKER

"Case Study on a Java-Based System for Personal Health Monitoring and Quality Improvement"

CODE: -

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

class HealthEntry {
   String date;      // YYYY-MM-DD
   int steps;
   int calories;
   float sleep;

   HealthEntry(String date, int steps, int calories, float sleep) {
      this.date = date;
      this.steps = steps;
      this.calories = calories;
      this.sleep = sleep;
   }
}

public class HealthTracker {

   static ArrayList<HealthEntry> data = new ArrayList<>();
   static Scanner sc = new Scanner(System.in);

   // Basic date validation: YYYY-MM-DD
   static boolean isValidDate(String date) {
      if (date.length() != 10) return false;

      for (int i = 0; i < 10; i++) {
         if (i == 4 || i == 7) {
            if (date.charAt(i) != '-') return false;
         } else {
            if (!Character.isDigit(date.charAt(i))) return false;
```

```
        }
    }
    return true;
}

static void addEntry() {
    System.out.print("Enter date (YYYY-MM-DD): ");
    String date = sc.nextLine();

    if (!isValidDate(date)) {
        System.out.println("Invalid date format.");
        return;
    }

    System.out.print("Enter steps: ");
    int steps;
    try {
        steps = Integer.parseInt(sc.nextLine());
        if (steps < 0) throw new Exception();
    } catch (Exception e) {
        System.out.println("Invalid steps value.");
        return;
    }

    System.out.print("Enter calories: ");
    int calories;
    try {
        calories = Integer.parseInt(sc.nextLine());
        if (calories < 0) throw new Exception();
    } catch (Exception e) {
        System.out.println("Invalid calories value.");
        return;
    }

    System.out.print("Enter sleep (hours, e.g., 7.5): ");
    float sleep;
    try {
        sleep = Float.parseFloat(sc.nextLine());
        if (sleep < 0) throw new Exception();
    } catch (Exception e) {
        System.out.println("Invalid sleep value.");
        return;
    }
```

```
    data.add(new HealthEntry(date, steps, calories, sleep));
    System.out.println("Entry added successfully!");
}

static void displayEntries() {
    if (data.isEmpty()) {
        System.out.println("No entries found.");
        return;
    }

    System.out.printf("\n%-12s %-10s %-12s %-10s\n",
            "Date", "Steps", "Calories", "Sleep");

    for (HealthEntry e : data) {
        System.out.printf("%-12s %-10d %-12d %-10.2f\n",
                e.date, e.steps, e.calories, e.sleep);
    }
}

static void showSummary() {
    if (data.isEmpty()) {
        System.out.println("No data available.");
        return;
    }

    int totalSteps = 0;
    int totalCalories = 0;
    float totalSleep = 0;

    int maxSteps = data.get(0).steps;
    int maxCalories = data.get(0).calories;
    float maxSleep = data.get(0).sleep;

    for (HealthEntry e : data) {
        totalSteps += e.steps;
        totalCalories += e.calories;
        totalSleep += e.sleep;

        if (e.steps > maxSteps) maxSteps = e.steps;
        if (e.calories > maxCalories) maxCalories = e.calories;
        if (e.sleep > maxSleep) maxSleep = e.sleep;
    }

    System.out.println("\n------- SUMMARY -------");
```

```
    System.out.println("Total days       : " + data.size());
    System.out.println("Total steps      : " + totalSteps);
    System.out.println("Average steps    : " + (totalSteps / data.size()));
    System.out.println("Total calories   : " + totalCalories);
    System.out.println("Average calories : " + (totalCalories / data.size()));
    System.out.println("Total sleep      : " + totalSleep + " hrs");
    System.out.println("Average sleep    : " + (totalSleep / data.size()) + " hrs");
    System.out.println("Max steps        : " + maxSteps);
    System.out.println("Max calories     : " + maxCalories);
    System.out.println("Max sleep        : " + maxSleep + " hrs");
}

static void visualizeData() {
    if (data.isEmpty()) {
        System.out.println("No data available to visualize.");
        return;
    }

    System.out.println("Visualize what?");
    System.out.println("1. Steps");
    System.out.println("2. Calories");
    System.out.println("3. Sleep");
    System.out.print("Enter choice: ");

    int choice;
    try {
        choice = Integer.parseInt(sc.nextLine());
    } catch (Exception e) {
        System.out.println("Invalid choice.");
        return;
    }

    float max = 0;

    for (HealthEntry e : data) {
        float value = 0;
        if (choice == 1) value = e.steps;
        else if (choice == 2) value = e.calories;
        else if (choice == 3) value = e.sleep;
        else {
            System.out.println("Invalid option.");
            return;
        }
        if (value > max) max = value;
```

```
    }

    if (max == 0) {
        System.out.println("All values are zero. Nothing to show.");
        return;
    }

    System.out.println("\n--- BAR CHART ---");
    for (HealthEntry e : data) {
        float value;
        if (choice == 1) value = e.steps;
        else if (choice == 2) value = e.calories;
        else value = e.sleep;

        int bars = (int)((value / max) * 30);

        System.out.print(e.date + " | ");
        for (int i = 0; i < bars; i++) {
            System.out.print("#");
        }

        if (choice == 1) System.out.println(" (" + e.steps + " steps)");
        else if (choice == 2) System.out.println(" (" + e.calories + " cal)");
        else System.out.println(" (" + e.sleep + " hrs)");
    }
}

public static void main(String[] args) {

    while (true) {
        System.out.println("\n=== PERSONAL HEALTH TRACKER ===");
        System.out.println("1. Add Entry");
        System.out.println("2. View Entries");
        System.out.println("3. Show Summary");
        System.out.println("4. Visualize Data");
        System.out.println("5. Exit");
        System.out.print("Enter choice: ");

        int choice;
        try {
            choice = Integer.parseInt(sc.nextLine());
        } catch (Exception e) {
            System.out.println("Invalid input.");
            continue;
```

```
            }

            switch (choice) {
                case 1:
                    addEntry();
                    break;
                case 2:
                    displayEntries();
                    break;
                case 3:
                    showSummary();
                    break;
                case 4:
                    visualizeData();
                    break;
                case 5:
                    System.out.println("Goodbye!");
                    return;
                default:
                    System.out.println("Choose between 1 - 5 only.");
            }
        }
    }
}
```

OUTPUT: -

```
=== PERSONAL HEALTH TRACKER ===
1. Add Entry
2. View Entries
3. Show Summary
4. Visualize Data
5. Exit
Enter choice: 3

------- SUMMARY -------
Total days       : 1
Total steps      : 3000
Average steps    : 3000
Total calories   : 2500
Average calories : 2500
Total sleep      : 8.0 hrs
Average sleep    : 8.0 hrs
Max steps        : 3000
Max calories     : 2500
Max sleep         : 8.0 hrs

=== PERSONAL HEALTH TRACKER ===
1. Add Entry
2. View Entries
3. Show Summary
4. Visualize Data
5. Exit
Enter choice: 4
Visualize what?
1. Steps
2. Calories
3. Sleep
Enter choice: 1

--- BAR CHART ---
2002-12-07 | ############################## (3000 steps)
```

```
=== PERSONAL HEALTH TRACKER ===
1. Add Entry
2. View Entries
3. Show Summary
4. Visualize Data
5. Exit
Enter choice: 1
Enter date (YYYY-MM-DD): 2002-12-07
Enter steps: 3000
Enter calories: 2500
Enter sleep (hours, e.g., 7.5): 8
Entry added successfully!

=== PERSONAL HEALTH TRACKER ===
1. Add Entry
2. View Entries
3. Show Summary
4. Visualize Data
5. Exit
Enter choice: 2

Date        Steps     Calories    Sleep
2002-12-07  3000      2500        8.00
```