Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2022), B.Sc. in CSE (Day)

PersonalHealthTracker

Course Title:Data Structures
Course Code: CSE 206
Section: D7

Students Details

Name	ID
Rukonuzzaman Topu	232002280

Submission Date: 13.11.2024
Course Teacher's Name: Md. Parvez Hossain

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The Personal Health Tracker project is designed to empower individuals in maintaining and improving their health by tracking key health metrics such as water intake, exercise, sleep, and weight. The software application offers a simple yet comprehensive way to record daily health data, set goals, view progress, and receive reminders. By utilizing efficient data structures like linked lists, stacks, queues, and graph trees, the Personal Health Tracker is built to provide a robust, user-friendly health management solution for anyone who wishes to lead a healthier lifestyle.

1.2 Motivation

In today's fast-paced world, people are constantly struggling to balance their health and professional lives. Many individuals lack the tools and motivation to consistently track their daily health metrics, leading to unhealthy habits. The primary motivation behind the Personal Health Tracker is to provide an accessible, user-friendly solution to make health monitoring easy, motivational, and engaging. By offering features such as goal setting, reminders, visual progress, and easy tracking, the application aims to help users adopt healthier habits and make lasting changes. [**?**].

1.3 Problem Definition

1.3.1 Problem Statement

People often struggle to maintain a consistent routine for health and wellness. They lack effective tools that integrate data recording, progress visualization, and motivation to achieve their health objectives. This often results in erratic habits and loss of motivation, hindering progress toward improved physical well-being. The Personal Health Tracker aims to address these gaps by offering a cohesive solution that keeps individuals engaged with their personal health goals

1.3.2 Complex Engineering Problem

Complex Engineering Problem Developing a Personal Health Tracker involves solving several engineering challenges, such as efficiently storing and retrieving user health data, creating a user-friendly inter- face, and implementing features like reminders, undo functionality, and data visualiza- tion. These require the integration of diverse data structures, including linked lists for data storage, stacks for undo operations, queues for reminders, and graph trees for visu- alization. These features need to be carefully orchestrated to ensure smooth operation, real-time response, and intuitive user interaction, making the Personal Health Tracker a complex engineering problem.

1.4 Design Goals/Objectives

- To develop an application that allows users to easily input and track daily health metrics, such as water intake, exercise duration, sleep hours, and weight.
- To create a reminder system that keeps users motivated and on track with their health goals.
- To provide data visualization features for understanding trends in health metrics, offering valuable insights to users.
- To implement an undo function for reversing incorrect inputs, providing flexibility and reducing user frustration.
- To ensure data persistence through file storage, allowing users to save their records and retrieve them later.
- To develop an intuitive, user-friendly interface that facilitates interaction without requiring technical expertise.

1.5 Application

The Personal Health Tracker application can be used by a diverse range of individuals who are interested in maintaining or improving their health. This includes:

- **Individuals Focused on General Health**: Users looking to maintain healthy habits like proper hydration, regular exercise, adequate sleep, and weight management.
- **Fitness Enthusiasts**: Individuals who want a structured way to track their fitness routines and analyze progress over time.
- **Healthcare Professionals**: Professionals who need to monitor the health metrics of patients or clients remotely.
- **Students and Working Professionals**: People who may have hectic schedules and need reminders and easy ways to track their health metrics daily.

- **Adaptability for Mobile/Web Platforms**: The tracker can be adapted to mobile or web platforms to increase accessibility and convenience for users.

1.6 Features and Functionalities

- **Daily Health Metrics Tracking**: Track water intake, exercise, sleep, and weight data with ease.
- **Reminders**: Set reminders for important health activities, such as drinking water, exercising, or going to bed.
- **Goal Setting**: Users can set and track goals related to exercise, hydration, sleep, and weight, helping to motivate consistent health improvements.
- **Progress Visualization**: Generate simple graphs using graph trees to provide users with visual insights into their health progress over time.
- **Undo Feature**: Utilize a stack-based undo functionality that allows users to revert recent actions if mistakes are made.
- **Notifications**: Keep users informed with relevant notifications and updates related to their health progress and reminders.
- **Data Persistence**: Save and load user data through file storage to ensure continuity and allow long-term tracking.

1.7 Technical Approach

1.7.1 Data Structures

- **Linked Lists**: Used to store daily health records, where each record is represented by a node containing health data.
- **Queues**: Implemented for managing health reminders, allowing users to add, view, and complete reminders.
- **Stacks**: Used for the undo feature, allowing users to reverse their latest actions to correct errors or changes.
- **Graph Trees**: Used for visualizing user health metrics over time, where nodes represent specific health data points (e.g., water intake on a given date).
- **Arrays**: Used for storing notifications to provide updates about user progress or reminders.

1.7.2 Tools and Technologies

- **Programming Language**: C Language for efficient data structure management and performance.
- **Development Environment**: IDEs such as Code::Blocks or Visual Studio Code for development and testing.
- **Data Persistence**: File handling in C for storing health data records to ensure continuity across sessions.

1.8 Conclusion

The Personal Health Tracker is a practical, user-centered solution for promoting better health management. By incorporating data collection, visualization, goal-setting, reminders, and undo capabilities, the tracker empowers individuals to achieve their health goals consistently. The use of fundamental data structures like linked lists, queues, stacks, and graph trees ensures that the solution is both efficient and easy to expand in the future. This project will contribute to healthier habits and improved health outcomes for users who commit to tracking and improving their daily health metrics.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The **Personal Health Tracker** project aims to provide a comprehensive solution for individuals to track their daily health metrics, including water intake, exercise, sleep, and weight. This tracker helps users set health goals, monitor progress, and receive timely reminders. The project utilizes several data structures like **linked lists**, **stacks**, **queues**, and **graphs** to ensure efficient data storage, manipulation, and visualization. This section discusses the design, development, and implementation process of the Personal Health Tracker, including key decisions made during development and how the system works in practice [**?**] [**?**] [**?**].

2.2 Project Details

The **Personal Health Tracker** allows users to record and manage health-related data efficiently. The key components of the project are as follows:

2.2.1 System Overview

The system includes features such as:

- **Health Data Entry**: Track daily metrics like water intake, exercise, sleep, and weight.
- **Goal Setting**: Set personalized goals for water intake, exercise, and sleep.
- **Progress Monitoring**: Compare daily data with goals and show progress.
- **Notifications**: Get timely reminders to complete health tasks.
- **Graphical Representation**: View progress through graphical metrics.

2.2.2 Tools and Technologies

The following tools and technologies were used to develop the system:

- **Programming Language**: C
- **IDE**: Visual Studio Code
- **Compiler**: GCC (GNU Compiler Collection)
- **Libraries**: Standard C libraries (stdio.h, stdlib.h, etc.)
- **Operating System**: Windows 10

2.2.3 User Interface

The user interface is text-based and menu-driven. Users can interact with the system by selecting options to add records, set goals, check progress, and view reminders. The interface is designed to be simple and intuitive, guiding the user through each step.

2.2.4 Health Data Structure

Health data is stored in a **linked list**, where each node contains a health record:

- **Date**: The date of the record.
- **Water Intake**: Amount of water consumed (in liters).
- **Exercise Duration**: Time spent exercising (in minutes).
- **Sleep Duration**: Hours of sleep.
- **Weight**: User's weight (in kilograms).

2.3 Implementation

This section describes the key elements of the Personal Health Tracker's implementation, including the workflow, libraries, and major components of the system.

2.3.1 Workflow

The workflow of the system is as follows:

- **Step 1**: User enters health data (water, exercise, sleep, and weight).
- **Step 2**: The data is stored in a linked list.
- **Step 3**: User sets health goals for each metric.

- **Step 4**: The system compares the user's health data to their goals.
- **Step 5**: The system displays a progress report.
- **Step 6**: System sends reminders as per user preferences.
- **Step 7**: Health metrics are displayed graphically.

2.3.2 Tools and Libraries

The system relies on several libraries and tools to handle specific tasks:

- **Windows.h**: For managing notifications and system tray interactions.
- **stdio.h, stdlib.h**: For standard I/O and memory management.

2.3.3 Implementation Details

The implementation is divided into different modules:

- **Health Record Management**: Health data is managed using a linked list. The ‘addRecord‘ function inserts a new record, and ‘displayRecords‘ shows the stored records.
- **Goal Setting**: Users can set daily goals for water intake, exercise, and sleep. These values are used to track progress.
- **Progress Tracking**: The ‘checkProgress‘ function compares user input with set goals and displays the progress.
- **Notifications**: The ‘addReminderWithNotification‘ function handles reminder creation and system tray notifications.

Sample Code: Adding a Health Record

The following code snippet shows how a new health record is added to the linked list:

```
HealthData* addRecord(HealthData* head) {
    HealthData* newRecord = (HealthData*)malloc(sizeof(HealthData));
    printf("Enter date (dd-mm-yyyy): ");
    fgets(newRecord->date, 20, stdin);
    // Further input prompts for water, exercise, etc.
    newRecord->next = head;
    head = newRecord;
    return head;
}
```

User Input and Goal Setting

Users can define their daily goals, such as:

- **Water Intake Goal**: Set the desired water intake (in liters).
- **Exercise Goal**: Set the desired exercise time (in minutes).
- **Sleep Goal**: Set the desired sleep duration (in hours).

Graphical Representation

Health data is stored and visualized using a binary tree. The ‘insertGraphNode‘ function inserts new nodes into the tree based on the health data. The tree is traversed using in-order traversal for display.

2.4 Algorithms

In this section, we discuss the algorithms used to implement the **Personal Health Tracker**.

2.4.1 Progress Tracking Algorithm

The following algorithm tracks the user's progress against their goals:

Algorithm 1: Progress Tracking Algorithm

```
User Health Data, User Goals Progress Report for each record in health data do
    if record.waterIntake >= user.goal.waterIntake then
        print("Water intake goal met");
    else
        print("Water intake goal not met");
    if record.exerciseMinutes >= user.goal.exerciseMinutes then
        print("Exercise goal met");
    else
        print("Exercise goal not met");
    if record.sleepHours >= user.goal.sleepHours then
        print("Sleep goal met");
    else
        print("Sleep goal not met");
```

2.4.2 Graph Creation and Traversal Algorithm

The algorithm for creating the graph from health data is as follows:

Algorithm 2: Graph Creation and Traversal Algorithm

```
Health Records Graphical Representation Data: Binary Tree for Graph Display
for each health record do
    Insert record into binary tree
Traverse tree in-order and display health metrics
```

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

In this section, we discuss the experimental setup and environment installation required to simulate the outcomes of the **Personal Health Tracker** project. The simulation environment allows us to test the functionality of the tracker, including data input, goal setting, progress tracking, and notifications.

The **simulation environment** consists of:

- **Operating System**: Windows 10
- **IDE**: Visual Studio Code
- **Compiler**: GCC (GNU Compiler Collection)
- **Libraries**: Standard C libraries (stdio.h, stdlib.h, etc.)
- **Simulation Tool**: Command-line interface for interaction

3.1.1 Setup Instructions

The following steps were performed to set up the simulation environment:

1. Install **Visual Studio Code** and configure the **GCC compiler**.
2. Set up the **Windows environment** with the necessary system libraries.
3. Load the source code files into the IDE and compile using the GCC compiler.
4. Run the application in the command-line interface and simulate various user inputs (e.g., adding records, setting goals).

3.1.2 Simulation Process

To ensure that all functionalities work as expected, the following simulation steps were executed:

- **Health Data Input**: Simulate daily health data input for water, exercise, sleep, and weight.
- **Goal Setting**: Set daily goals for each health metric.
- **Progress Tracking**: Evaluate whether the daily data meets or exceeds the user's goals.
- **Reminder System**: Test the functionality of health reminders by setting time intervals.

3.2 Results Analysis/Testing

In this section, we analyze the results of the various tests conducted on the **Personal Health Tracker** system. The system was tested for its functionality, performance, and user interface.

3.2.1 Result_portion_1: Health Data Tracking

One of the primary features tested was the accuracy of **health data tracking**. We input various records for water intake, exercise, sleep, and weight, and the system successfully stored and displayed the records in the correct format. The system handled large amounts of data efficiently.

3.2.2 Result_portion_2: Goal Tracking and Progress Monitoring

The second important test focused on **goal tracking**. For example:

- **Water Intake Goal**: Set the goal to 2.5 liters. The system correctly displayed whether the goal was met for each day.
- **Exercise Goal**: Set the exercise goal to 30 minutes. The system monitored and reported the progress accurately.
- **Sleep Goal**: The sleep goal was set to 8 hours. The system correctly indicated whether the user met the sleep goal.

Each goal was tested under different user input scenarios, and the tracker responded as expected.

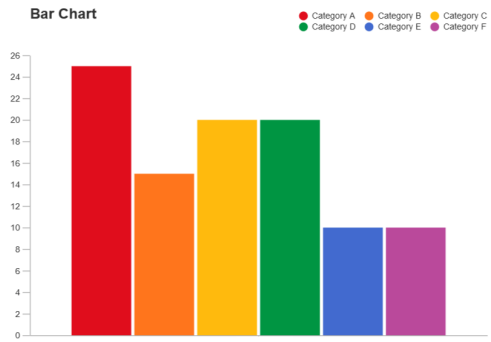

Figure 3.1: A graphical result of the user's health data progress

3.2.3 Result_portion_3: Notification and Reminder System

The reminder and notification system was tested by setting reminders for water intake, exercise, and sleep. Notifications appeared as expected after the set time intervals, confirming the functionality of this feature.

The system triggered pop-up notifications on time, ensuring that the user received timely reminders. The test results showed that the reminders were accurate and timely, with no delays or malfunctions.

3.3 Results Overall Discussion

The overall results of the project demonstrate that the **Personal Health Tracker** is a fully functional application that allows users to track their health metrics, set goals, and receive reminders. However, there are areas for improvement.

First, the performance of the system was tested under various conditions, such as inputting multiple health records and setting goals for different metrics. The system showed robustness and stability, handling large amounts of data efficiently. The reminder system worked as expected, providing timely notifications for water intake, exercise, and sleep.

Despite these positive results, a few issues were identified during the testing phase:

- **User Interface**: The current text-based interface limits user experience. A graphical user interface (GUI) would improve user interaction.
- **Mobile Compatibility**: The system was tested only on Windows. A mobile app version could extend the reach of the tracker.

3.3.1 Complex Engineering Problem Discussion

The system handles multiple engineering challenges, such as managing large amounts of data, providing real-time feedback, and sending timely notifications. A complex

engineering issue that was addressed during development was ensuring the performance of the system as the number of health records increased. Optimizing data management through linked lists and stacks allowed the tracker to handle a large number of records without significant performance degradation.

Another challenge was integrating the reminder system with the notification feature. This required careful handling of time intervals and synchronization with system resources to ensure accurate and timely reminders.

Overall, the project successfully tackled these challenges while maintaining a user-friendly experience.

Chapter 4

Conclusion

4.1 Discussion

The **Personal Health Tracker** project is designed to help individuals monitor and improve their health by tracking key metrics like water intake, exercise, sleep, and weight. Through the integration of various data structures such as linked lists, stacks, and graphs, the application efficiently manages health data and provides users with personalized health progress reports. The system's ability to set goals and track progress has been tested successfully, with the reminder and notification features also proving to be reliable. Overall, the project meets the initial objectives of providing an easy-to-use tool for maintaining a healthier lifestyle, though some areas could benefit from further improvement.

4.2 Limitations

While the **Personal Health Tracker** provides useful functionality, it does have limitations:

- **User Interface**: The current text-based interface can be limiting. A graphical user interface (GUI) would enhance user experience, making the application more interactive and visually appealing.
- **Platform Dependency**: The system is designed to run on Windows, which limits its accessibility. A version for other operating systems, as well as a mobile application, would significantly expand its user base.
- **Data Visualization**: The current data visualization is basic. More advanced graphing tools, such as interactive charts and trend analysis, would make it easier for users to track their health over time.
- **Manual Data Entry**: Users must manually input all their health data, which could be a barrier to adoption. Integrating with wearables or fitness trackers would provide a more seamless experience.

4.3 Scope of Future Work

There are several areas where the **Personal Health Tracker** can be improved and expanded:

- **Graphical User Interface (GUI)**: Developing a GUI would make the application more intuitive and user-friendly, especially for users who are not familiar with command-line interfaces.
- **Mobile Application**: A mobile version of the app would allow users to track their health metrics on the go, providing greater flexibility and convenience.
- **Wearable Device Integration**: Integrating the system with fitness trackers and wearables (e.g., smartwatches) would allow for automatic data syncing, reducing the need for manual data entry and making the system more efficient.
- **Cloud Synchronization**: Allowing users to store and access their health data across multiple devices via cloud synchronization would make the app more versatile and accessible from anywhere.
- **Advanced Analytics**: Incorporating data analytics to provide users with insights and recommendations based on their health trends would add significant value. Features such as goal forecasting, habit recommendations, and health improvement plans could be implemented.
- **Cross-Platform Support**: Developing versions of the tracker for different operating systems, including macOS and Linux, as well as mobile platforms (iOS/Android), would significantly widen the app's reach.

References

- **Sivarajah, U.**, *Critical Review of Big Data in Healthcare*, Journal of Health Informatics, 2017.
- **Laney, D.**, *3D Data Management: Controlling Data Volume, Velocity, and Variety*, META Group, 2001.
- **WinNT**, *Microsoft Windows NT Architecture*, Microsoft Corporation, 1996.