HEALTHCARE APPOINTMENT BOOKING

NAME: LAKSHMI PRAMODE

ROLL NO: 48

COMPUTER SCIENCE

DATE: 17-07-2024

INTRODUCTION

In the fast-paced world of healthcare, effective management of patient information and appointment scheduling is crucial for delivering quality care. Administrative tasks, such as booking appointments, maintaining medical records, and sending reminders, are essential yet time-consuming. This project focuses on developing a console-based healthcare management system using the C programming language. The system aims to simplify and automate the management of patient appointments and medical records for healthcare providers. By providing a straightforward, menu-driven interface, this system ensures that users can efficiently perform administrative tasks without the need for complex operations...

Brief overview of the project : This code implements a simple healthcare management system in C. It allows users to perform the following tasks via a text-based menu:

Book an Appointment: Collects and stores appointment details including appointment ID, patient ID, provider ID, provider name, specialization, date, and time into a file called appointments.txt.

Manage Medical Records: Collects and stores patient details including patient ID, name, contact number, and medical record into a file called patients.txt .

Send Appointment Reminders: Reads the appointments.txt file and displays reminders for all the appointments.

Display Appointments: Reads and displays all appointments from the appointments.txt file.

Problem statement : Develop a healthcare management system in C that allows administrators to book appointments, manage patient records, send appointment reminders, and display appointment details. The system should use file handling to store and retrieve data, ensuring efficient healthcare service management

Objectives

Efficiency: Streamline the process of managing appointments and patient records to save time and reduce administrative burden.

Data Persistence: Ensure that all entered data is stored in files, allowing information to persist between program executions.

User-Friendly Interface: Provide an intuitive, menu-driven interface that is easy to navigate for users with varying levels of technical expertise.

Safe Data Handling: Implement practices to prevent errors such as buffer overflows and ensure that data is entered correctly.

Scalability: Design the system to handle a growing number of appointments and patient records as the healthcare practice expands.

SYSTEM REQUIREMENTS

Hardware requirements

Processor: - Minimum: 1 GHz processor.

- Recommended: 2 GHz or faster processor.

Memory (RAM): - Minimum: 512 MB.

- Recommended: 1 GB or more.

Storage: - Minimum: 10 MB free disk space for the program and data files.

- Recommended: 100 MB free disk space to accommodate more records and future expansions.

Display: - A monitor capable of displaying text-based console output.

Software requirements

Operating system - Windows (XP or later)
- Linux (any distribution)
- macOS

Compiler - GCC (GNU Compiler Collection) for Linux and macOS.
- MinGW or TDM-GCC for Windows.
- Clang for various platforms.

Development Environment (optional): - Code::Blocks
- Dev-C++
- Visual Studio Code
- Eclipse CDT

Text Editor: - Notepad++
- Vim
- Sublime Text
- Atom

DESIGN AND DEVELOPMENT

Program logic: The healthcare management system program is designed to handle key administrative tasks such as booking appointments, managing medical records, sending reminders, and displaying appointments.

Main Menu Loop: - Display a menu of options for the user to choose from.

- Options include: booking an appointment, managing medical records, sending appointment reminders, displaying appointments, and exiting the program.

- Read the user's input and invoke the appropriate function based on the choice.

Book Appointments: - Collect appointment details from the user, including appointment ID, patient ID, provider ID, provider name, provider specialization, date, and time.

- Store these details in a file (appointments.txt) to maintain a record of all booked appointments.

- Confirm to the user that the appointment has been booked successfully.

Manage Medical Records: - Collect patient details from the user, such as patient ID, name, contact number, and medical record.

- Store these details in a file (patients.txt) to maintain an up-to-date record of patients.

- Confirm to the user that the medical record has been updated successfully.

Send Appointment Reminders: - Read the appointment details from the appointments.txt file.

-Display reminders for all upcoming appointments, showing details like appointment ID, date, and time.

Display Appointments: - Read the appointment details from the appointments.txt file.

- Display all the details of each appointment, including appointment ID, patient ID, provider ID, provider name, specialization, date, and time.

PSEUDOCODE

```
Initialize:
- Define structures: patient and
appointment

Main Loop:
while true:
    Display Menu:
    - "HEALTHCARE MANAGEMENT
SYSTEM"
    - Options:
        1. Book an appointment
        2. Manage medical records
        3. Send appointment
reminders
        4. Display appointments
        5. Exit

    Prompt user for choice (opt)
```

```
    Switch (opt):
        Case 1:
            Call book_appointments()
            break
        Case 2:
            Call
manage_medical_records()
            break
        Case 3:
            Call
send_appointment_reminders()
            break
        Case 4:
            Call display_appointments()
            break
        Case 5:
            Exit program
            break
        Default:
            Display "Invalid choice"
```

```
Function book_appointments():
    Input:
    - Prompt for appointment ID, patient ID, provider ID, provider name, specialization, date, time

    Output:
    - Append appointment details to "appointments.txt"
    - Display "Appointment booked successfully"

Function manage_medical_records():
    Input:
    - Prompt for patient ID, name, contact number, medical record

    Output:
    - Append patient details to "patients.txt"
    - Display "Medical record updated successfully"
```

```
Function send_appointment_reminders():
    Input:
    - Read appointments from "appointments.txt"

    Output:
    - Display reminders for each appointment: ID, date, time

Function display_appointments():
    Input:
    - Read appointments from "appointments.txt"

    Output:
    - Display all appointment details: ID, patient ID, provider ID, provider name, specialization, date, time
```

Test cases

Book an Appointment

Input- Choose option 1 from the main menu

- Enter valid appointment details.

Expected Output: - Message: "Appointment booked successfully..."

Validation:- Check appointments.txt to ensure the details are correctly appended.

Manage Medical Records

Input:- Choose option 2 from the main menu.

- Enter valid patient details:

Expected Output:- Message: "Medical record updated successfully..."

Validation:- Check patients.txt to ensure the details are correctly appended.

Send Appointment Reminders

Input:- Choose option 3 from the main menu.

Expected Output:- Display reminders for each appointment in appointments.txt, showing appointment ID, date, and time.

Validation:- Verify that reminders are displayed correctly on the console.

Display Appointments

Input:- Choose option 4 from the main menu.

Expected Output:- Display all appointment details from appointments.txt, including appointment ID, patient ID, provider ID, provider name, specialization, date, and time.

Validation:- Ensure all appointment details are correctly displayed on the console.

Invalid Choice

Input:- Enter an invalid option (e.g., 6) from the main menu.

Expected Output:- Message: "Invalid choice“

Validation:- Verify that the program handles invalid inputs gracefully and prompts the user again.

Each test case should validate both the expected output on the console and the correctness of data stored in appointments.txt and patients.txt.

Ensure that edge cases, such as empty files or large datasets, are also considered during testing to confirm robustness and reliability of the system.

By executing these test cases, you can ensure that the healthcare management system functions correctly across its various functionalities and handles user inputs and data storage effectively.

Executing these test cases will verify that the healthcare management system functions correctly in terms of data input, storage, retrieval, and user interaction. It ensures that the system operates as intended, maintaining data integrity and providing expected functionalities for managing appointments and medical records effectively.

OUTPUT OR RESULT

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\c programing> gcc -o project project.c
PS C:\c programing> ./project.exe
HEALTHCARE MANAGEMENT SYSTEM
1-book an appointment
2-manage medical records
3-send appointment reminders
4-display appointments
5-exit
1
enter appointment ID: 1
enter patient ID: 2345
enter provider ID: 50
enter provider name: Dr.Jayakumar
enter provider specialization: cardiologist
enter date: 23/04/2015
enter time: 10:00am
appointment booked successfully...
```

```
HEALTHCARE MANAGEMENT SYSTEM
1-book an appointment
2-manage medical records
3-send appointment reminders
4-display appointments
5-exit
2
enter patient ID: 2345
enter patient name: Lakshmi
enter patient contact number: 9700000000
enter medical record of patient: angiogram
medical record updated successfully...
HEALTHCARE MANAGEMENT SYSTEM
1-book an appointment
2-manage medical records
3-send appointment reminders
4-display appointments
5-exit
3
appointment reminders...
REMINDER-appointment ID 1 on 23/04/2015 at 10:00am
```

```
HEALTHCARE MANAGEMENT SYSTEM
1-book an appointment
2-manage medical records
3-send appointment reminders
4-display appointments
5-exit
4
appointments...
appointment ID:1
patient ID:2345
provider ID:50
provider name:Dr.Jayakumar
specialization:cardiologist
date:23/04/2015
Time:10:00am
HEALTHCARE MANAGEMENT SYSTEM
1-book an appointment
2-manage medical records
3-send appointment reminders
4-display appointments
5-exit
5
PS C:\c programing>
```

project.c | appointments.txt X

appointments.txt

```
1 2345 50 Dr.Jayakumar cardiologist 23/04/2015 10:00am
2
```

project.c | patients.txt X

patients.txt

```
2345 Lakshmi 1110065408 angiogram
```

CONCLUSION

Summary: The healthcare management system project effectively streamlines administrative tasks such as booking appointments, managing medical records, sending reminders, and displaying appointment details. It enhances operational efficiency and improves patient care management through its user-friendly interface and structured data handling. Overall, the system contributes to better resource utilization and streamlined processes within healthcare facilities.

Future enhancements: -Integration with Electronic Health Records (EHR).

-Appointment Scheduling Algorithms.

-Patient Portal and Mobile App.

-Automated Billing and Payment Processing.

-Data Analytics and Reporting.

-Enhanced Security and Compliance.

-Feedback Mechanisms.

THANK YOU...