Lanka Nippon Biztech Institute

Assignment Name – **Clinical Management System**

Course Code Course Name - IT11024 Programming Fundamentals

Name	: D.M.P.Janodh Dissanayaka
Index Number	: UGC0223015
Batch	: Software Engineering 02

CLINICAL MANAGEMENT SYSTEM

CONTENT

Introduction

Brief Overview

This Clinical Management System is a program designed to optimize the management of clinical information, patient records, appointments, and administrative tasks within health-care facilities. It offers a user-friendly interface accessible to both doctors and patients.

Objectives and Goals

1. **Efficient Appointment Scheduling:** The program aims to streamline the process of scheduling appointments for patients with available doctors. Patients can easily schedule, cancel and view their appointments.

2. **Comprehensive Patient Record Management:** This program allows doctors to update and access patient records securely. Through file handling , doctors can efficiently manage patient's diagnosis reports, medication reports and recommendation reports.

3. **Secure User Authentication with Hashed Passwords:** Ensuring the security and confidentiality of patient data is paramount. This program stores the user's passwords as hashed passwords to safeguard sensitive information.

4. **Effective Schedule Management for Doctors:** Doctors can easily keep track of their schedules and appointments using the program.

Problem Statement

Problem Statement

This Clinical Management System addresses the inefficiencies and challenges inherent in traditional paper-based systems and disjointed digital solutions within health care facilities. The primary problem being addressed is the lack of a unified, efficient, and secure platform for managing clinical information, patient records, appointments, and administrative tasks.

Importance

Inefficient appointment scheduling often leads to long wait times for patients, which can result in frustration, decreased patient satisfaction, and even compromised health outcomes due to delayed care. Furthermore, disparate patient record management systems hinder effective communication and collaboration among health care providers, leading to errors, redundancies, and gaps in patient care.

The security of patient data is a critical concern in the health care industry, as breaches can lead to severe consequences including identity theft, financial fraud, and compromised medical histories. Additionally, without effective schedule management tools, doctors may struggle to optimize their time, leading to overbooking, missed appointments, and burnout.

Therefore, the development of a comprehensive Clinical Management System is imperative to streamline processes, enhance communication, and safeguard patient information within health care settings.

Methodology

This Clinical Management System was developed using several methodologies when achieving the above mentioned objectives. The approach to the project was taken by creating several .cpp files (main.cpp,UserAuthenticationSystem.cpp, DoctorsFunctions.cpp, PatientsFunctions.cpp) , creating a header file for each .cpp file. And the files were connected to the main.cpp. The functions that can be achieved by a doctor were created in the DoctorsFunctions.cpp and the functions that can be achieved by a patient were created in the PatientsFunctions.cpp. And all the user authentication functions (Register, login, hashing the password) were created in the UserAuthenticationSystem.cpp.

Overview of Technologies, Tools, and Frameworks that were used

C++ Language: C++ was chosen as the primary programming language due to its efficiency, flexibility, and suitability for system-level programming tasks.

Variables, Data Types, Operators, and Control Flow Statements: Fundamental C++ syntax elements were extensively used throughout this project to define variables, manipulate data types, employ operators, and implement control flow logic to manage program execution paths.

Functions: Modeling of code was achieved through the definition and invocation of functions, enhancing code maintainability and re-usability.

Header Files: Header Files were used for organizing function prototypes, promoting code readability and structure.

File Handling: File input/output operations were employed to persistently store and retrieve patient and doctor information, ensuring data integrity and accessibility across sessions.

Data Structures: The Standard Template Library (STL) was utilized to implement data structures such as vectors, facilitating the storage and manipulation of collections of objects like patients, doctors, and appointments.

STL (Standard Template Library): The STL provided useful C++ templates, such as std::unordered_map, which efficiently stored user accounts, enhancing the program's performance and scalability.

User Input and Validation: Techniques for accepting user input and performing input validation were employed to handle potential errors and ensure the correct behavior of the program.

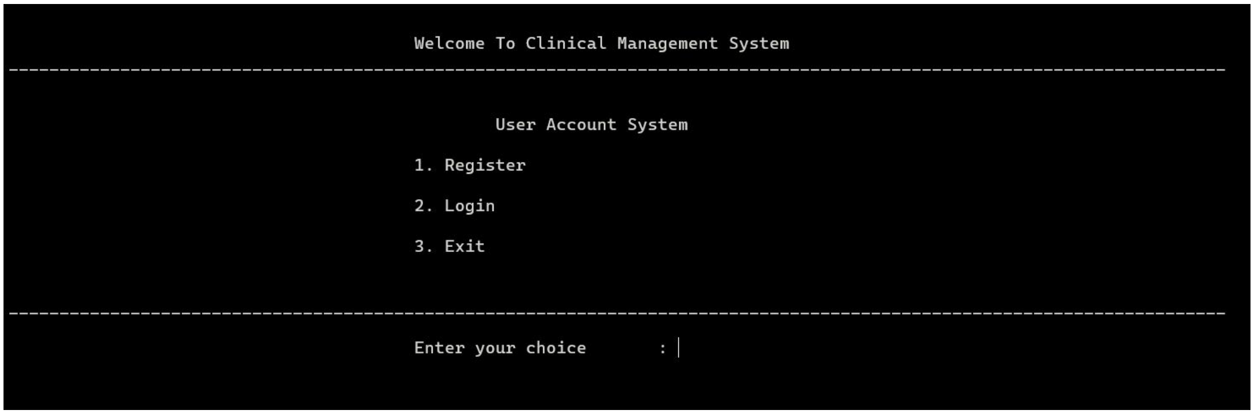
String Manipulation: String manipulation techniques were utilized to manage user-related data such as user names, passwords, and other textual information.

Memory Management: Direct control over memory allocation and de-allocation was exercised using new and delete operators, with careful attention paid to avoid memory leaks and optimize resource usage.

User Interface Design: A user-friendly console interface was designed, featuring appropriate prompts, menus, and clear instructions to guide users through the system, enhancing usability and accessibility.

EVIDENCE FOR THE SYSTEM IMPLEMENTATION

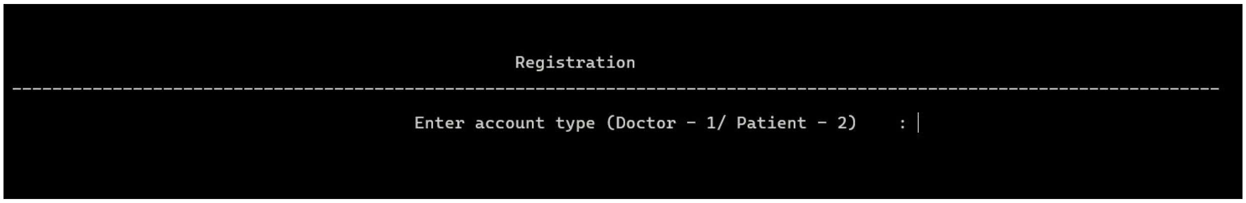
- Main Menu / User Account System



```
                                        Welcome To Clinical Management System
--------------------------------------------------------------------------------------------------------------------------

                                                User Account System
                                        1. Register
                                        2. Login
                                        3. Exit

--------------------------------------------------------------------------------------------------------------------------
                                        Enter your choice       : |
```

This is the main menu or the user account system of the clinical management system. Here there are 3 options; Register, Login and Exit. If the user (Either Doctor or Patient) , has not registered to the system before , select the choice 1 to register as a new user. If the user is a registered user of the system, select the choice 2 to login. And choice 3 is to exit the program.

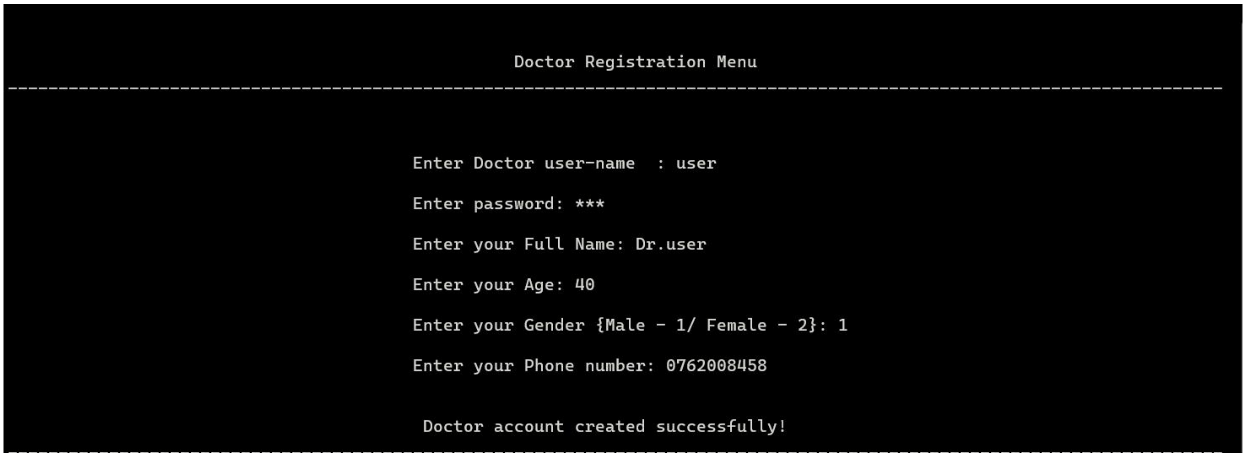
- Register Menu



```
                                                  Registration
--------------------------------------------------------------------------------------------------------------------------
                                        Enter account type (Doctor - 1/ Patient - 2)    : |
```

If the user is a unregistered doctor, enter 1 to go to the doctor registration   or if the user is a unregistered patient, enter 2 to go to the patient   registration.

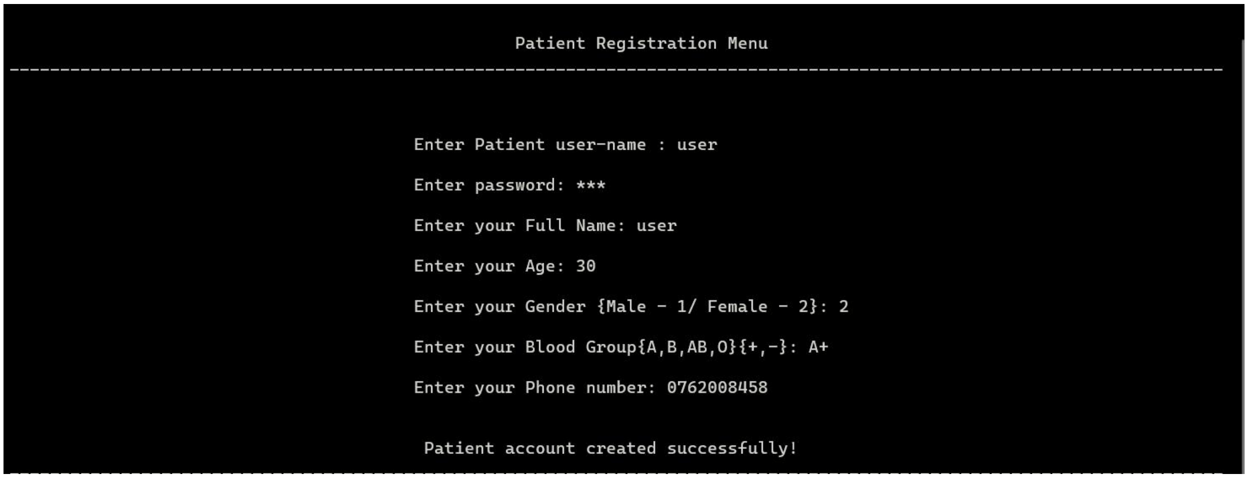
- Doctor Registration Menu



```
                                      Doctor Registration Menu
----------------------------------------------------------------------------------------------------------------------


                              Enter Doctor user-name  : user

                              Enter password: ***

                              Enter your Full Name: Dr.user

                              Enter your Age: 40

                              Enter your Gender {Male - 1/ Female - 2}: 1

                              Enter your Phone number: 0762008458

                               Doctor account created successfully!
----------------------------------------------------------------------------------------------------------------------
```

To register an unregistered doctor , the above information should be filled. A username and a password should be entered to login again to the system. Then the user can enter the full name, age, gender (by selecting 1 or 2) and the phone number.

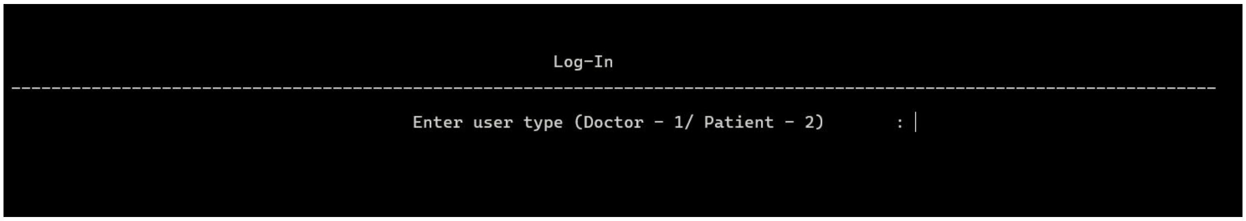
- Patient Registration Menu



```
                                      Patient Registration Menu
----------------------------------------------------------------------------------------------------------------------


                              Enter Patient user-name : user

                              Enter password: ***

                              Enter your Full Name: user

                              Enter your Age: 30

                              Enter your Gender {Male - 1/ Female - 2}: 2

                              Enter your Blood Group{A,B,AB,O}{+,-}: A+

                              Enter your Phone number: 0762008458

                               Patient account created successfully!
----------------------------------------------------------------------------------------------------------------------
```

To register an unregistered patient , the above information should be filled. A username and a password should be entered to login again to the system. Then the user can enter the full name, age, gender (by selecting 1 or 2), Blood Group and the phone number.

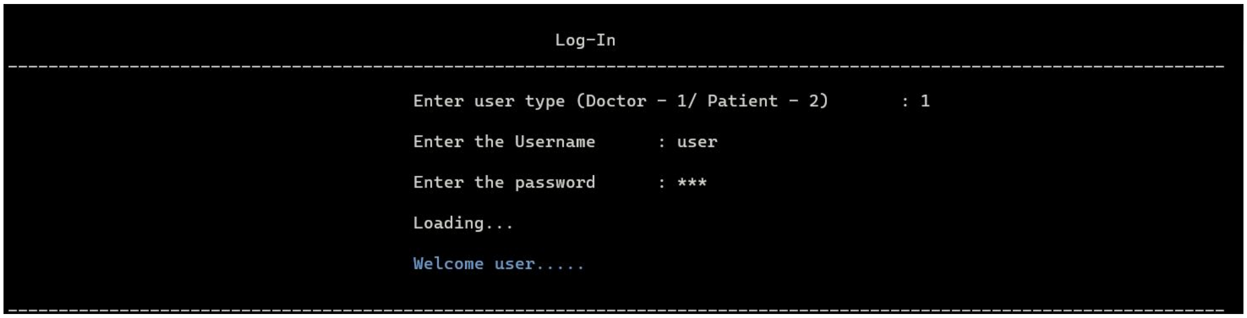
- Login



```
                                                   Log-In
----------------------------------------------------------------------------------------------------------------
                                      Enter user type (Doctor - 1/ Patient - 2)       : |
```

If the user is a registered doctor, enter 1 to go to login as a doctor or if the user is a registered patient, enter 2 to go to login as a patient.

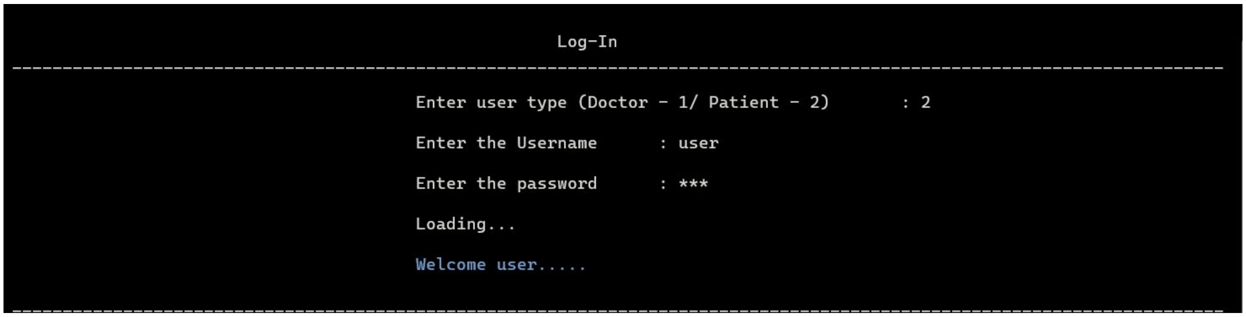
- Doctor Login



```
                                                   Log-In
----------------------------------------------------------------------------------------------------------------
                                      Enter user type (Doctor - 1/ Patient - 2)       : 1

                                      Enter the Username      : user

                                      Enter the password      : ***

                                      Loading...

                                      Welcome user.....

----------------------------------------------------------------------------------------------------------------
```

If a registered doctor wants to login to the system he or she should enter the username and the password that was given in the registration.

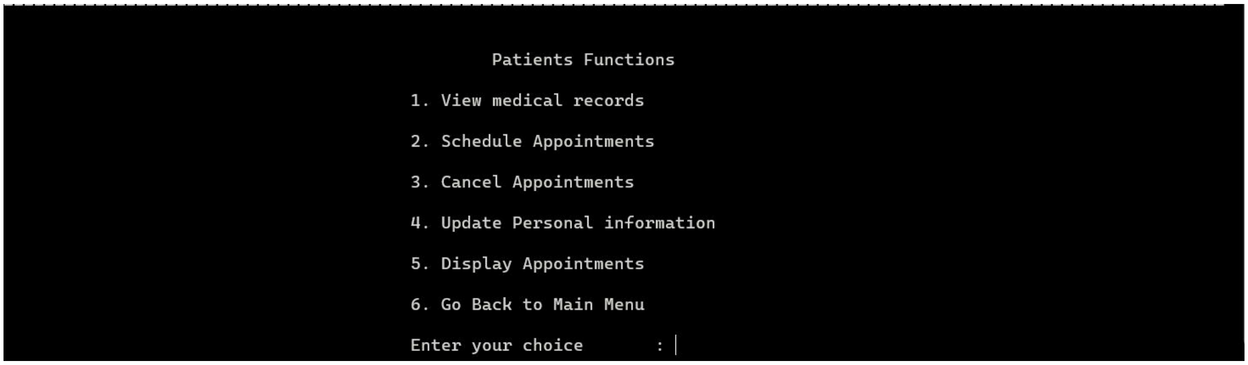
- Patient Login



```
                                        Log-In
----------------------------------------------------------------------------------------------------------------------
                                Enter user type (Doctor - 1/ Patient - 2)       : 2

                                Enter the Username      : user

                                Enter the password      : ***

                                Loading...

                                Welcome user.....

----------------------------------------------------------------------------------------------------------------------
```

If a registered patient wants to login to the system he or she should enter the username and the password that was given in the registration.

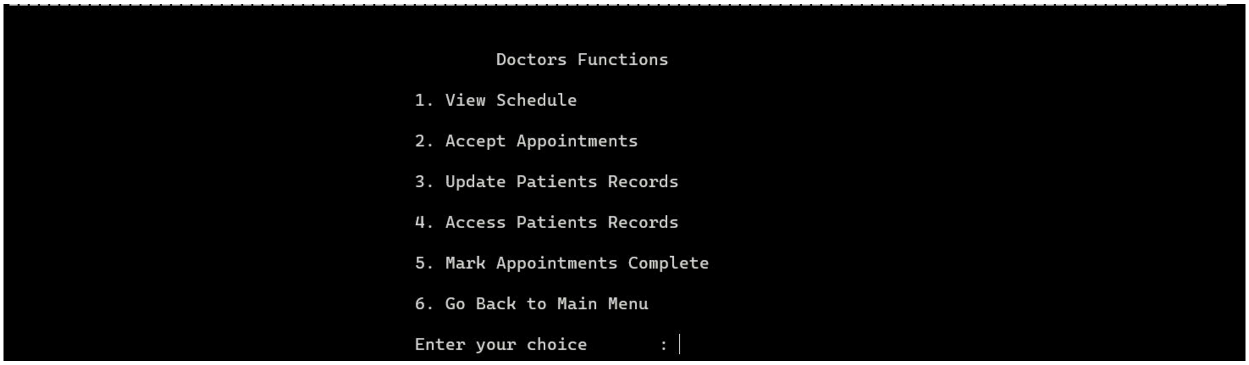
- Patient Menu



```
----------------------------------------------------------------------------------------------------------------------

                                        Patients Functions

                                1. View medical records

                                2. Schedule Appointments

                                3. Cancel Appointments

                                4. Update Personal information

                                5. Display Appointments

                                6. Go Back to Main Menu

                                Enter your choice       : |
```

When a patient is registered and logged in to the system the main functions or services that can be fulfilled from this program will be displayed in the above code. The user have to enter the choice for the relevant function.

- Doctor Menu

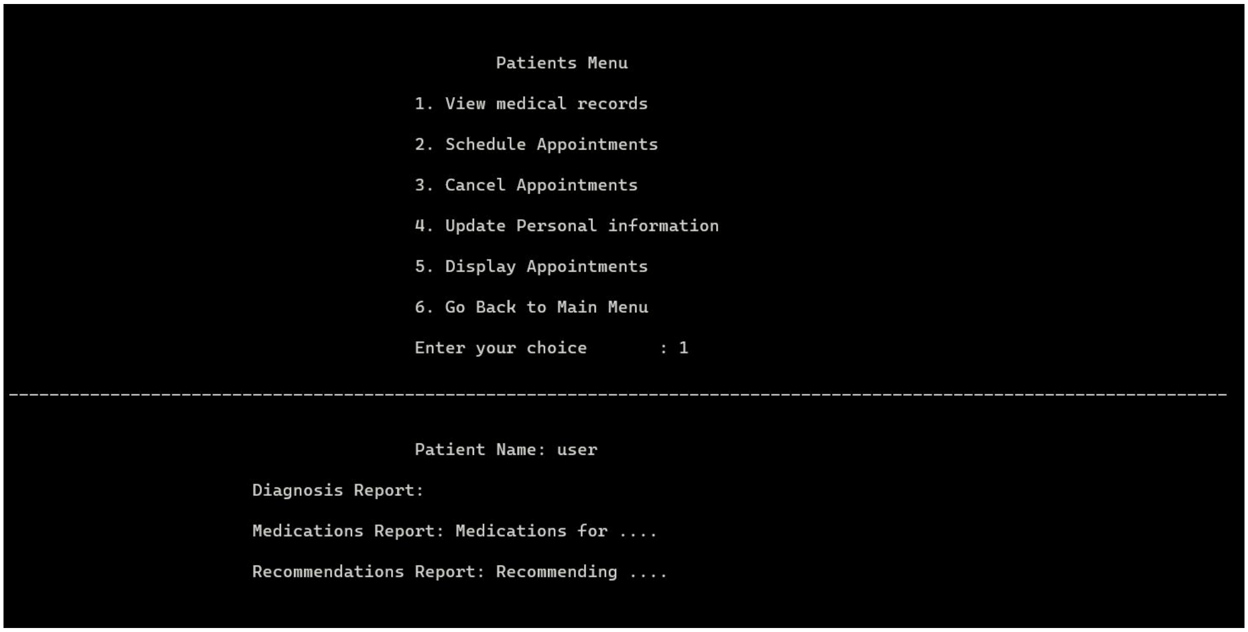


```
        Doctors Functions
1. View Schedule
2. Accept Appointments
3. Update Patients Records
4. Access Patients Records
5. Mark Appointments Complete
6. Go Back to Main Menu
Enter your choice       : 
```

When a doctor is registered and logged in to the system the main functions or services that can be fulfilled from this program will be displayed in the above code. The user have to enter the choice for the relevant function.

Patient Functions

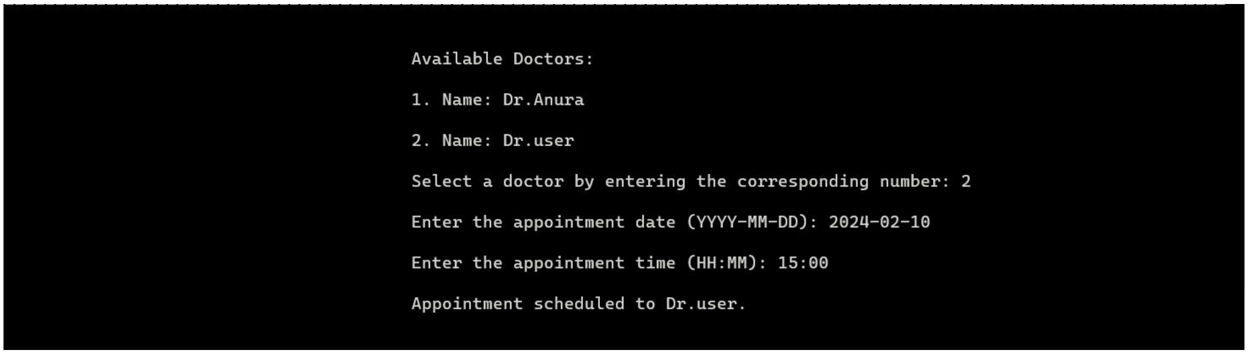
- View medical records



```
                                Patients Menu
                        1. View medical records
                        2. Schedule Appointments
                        3. Cancel Appointments
                        4. Update Personal information
                        5. Display Appointments
                        6. Go Back to Main Menu
                        Enter your choice       : 1
------------------------------------------------------------------------------------------------------------------------

                        Patient Name: user
                Diagnosis Report:
                Medications Report: Medications for ....
                Recommendations Report: Recommending ....
```

When a doctor update the patient’s medical records which include the Diagnosis report, Medications report and the Recommendations report the patient can view his or her medical records via this function.

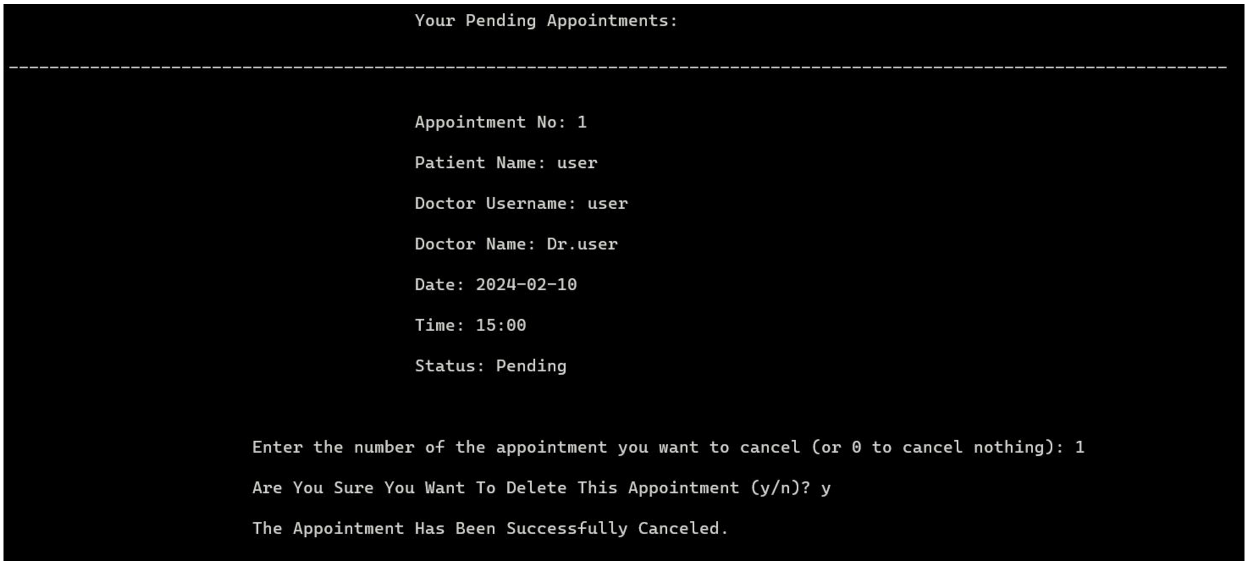
- Schedule Appointments



```
                        Available Doctors:
                        1. Name: Dr.Anura
                        2. Name: Dr.user
                        Select a doctor by entering the corresponding number: 2
                        Enter the appointment date (YYYY-MM-DD): 2024-02-10
                        Enter the appointment time (HH:MM): 15:00
                        Appointment scheduled to Dr.user.
```

To schedule an appointment for a doctor the patient should select the choice 2, then the function will show the registered and available doctors. The user can select a doctor by entering the corresponding number of the doctor, then the patient should enter the appointment date and time. The appointment with be stored as a pending appointment.

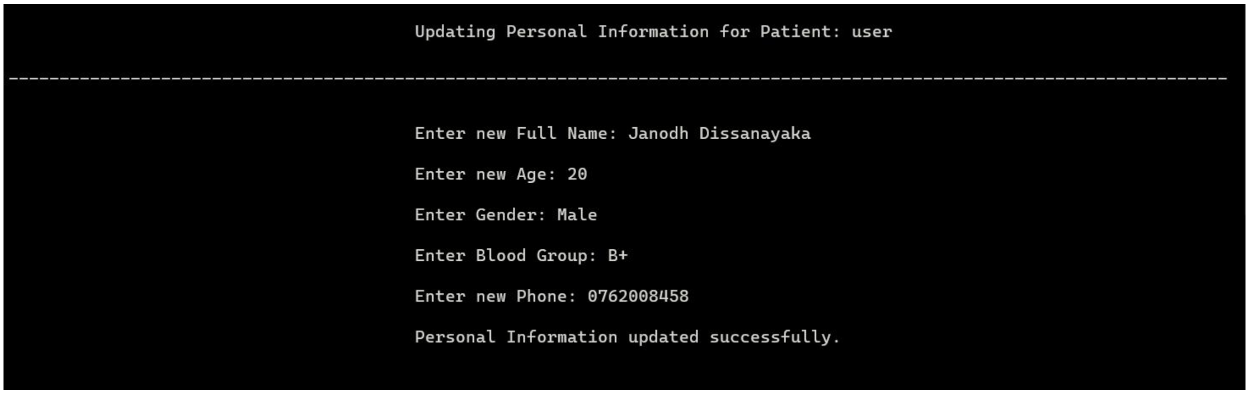
- Cancel Appointments



```
                                        Your Pending Appointments:

------------------------------------------------------------------------------------------------------------------------

                                        Appointment No: 1

                                        Patient Name: user

                                        Doctor Username: user

                                        Doctor Name: Dr.user

                                        Date: 2024-02-10

                                        Time: 15:00

                                        Status: Pending


                        Enter the number of the appointment you want to cancel (or 0 to cancel nothing): 1

                        Are You Sure You Want To Delete This Appointment (y/n)? y

                        The Appointment Has Been Successfully Canceled.
```

The patient can cancel a pending appointment via the choice 3, firstly the user should enter the number of the appointment that want to canceled , then to confirm the cancellation letter ‘y’ should be entered as a yes.

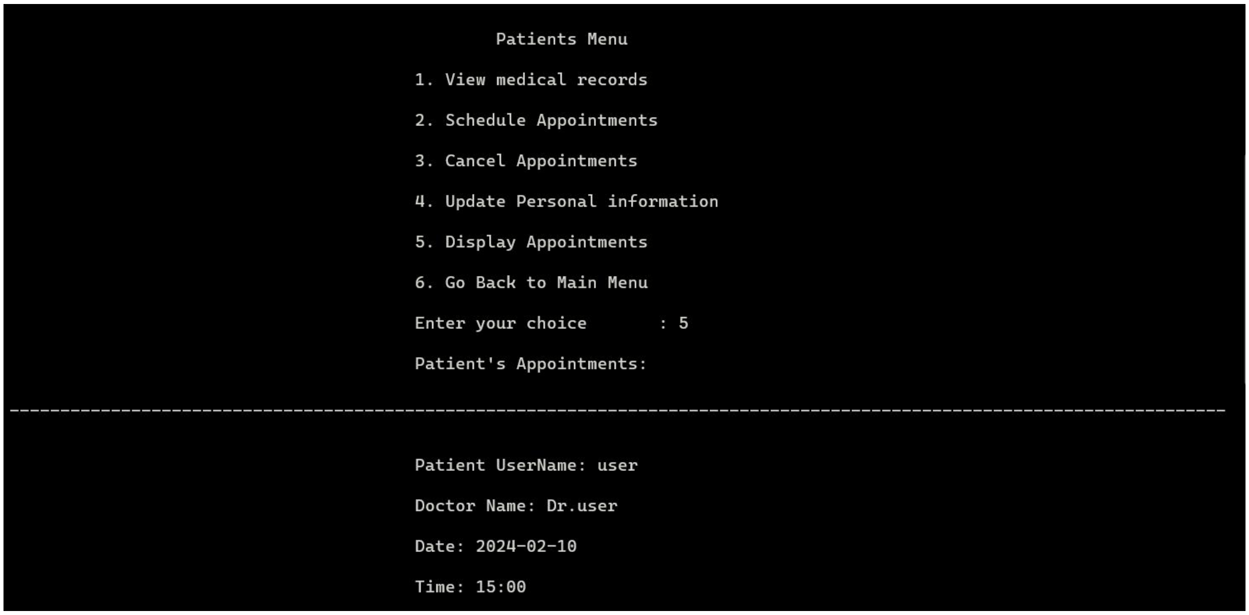
- Update Personal Information



```
                                        Updating Personal Information for Patient: user

------------------------------------------------------------------------------------------------------------------------

                                        Enter new Full Name: Janodh Dissanayaka

                                        Enter new Age: 20

                                        Enter Gender: Male

                                        Enter Blood Group: B+

                                        Enter new Phone: 0762008458

                                        Personal Information updated successfully.
```

By selecting the choice 4, the patient can update his or her personal information, such as name, age and phone number.

- Display Appointments

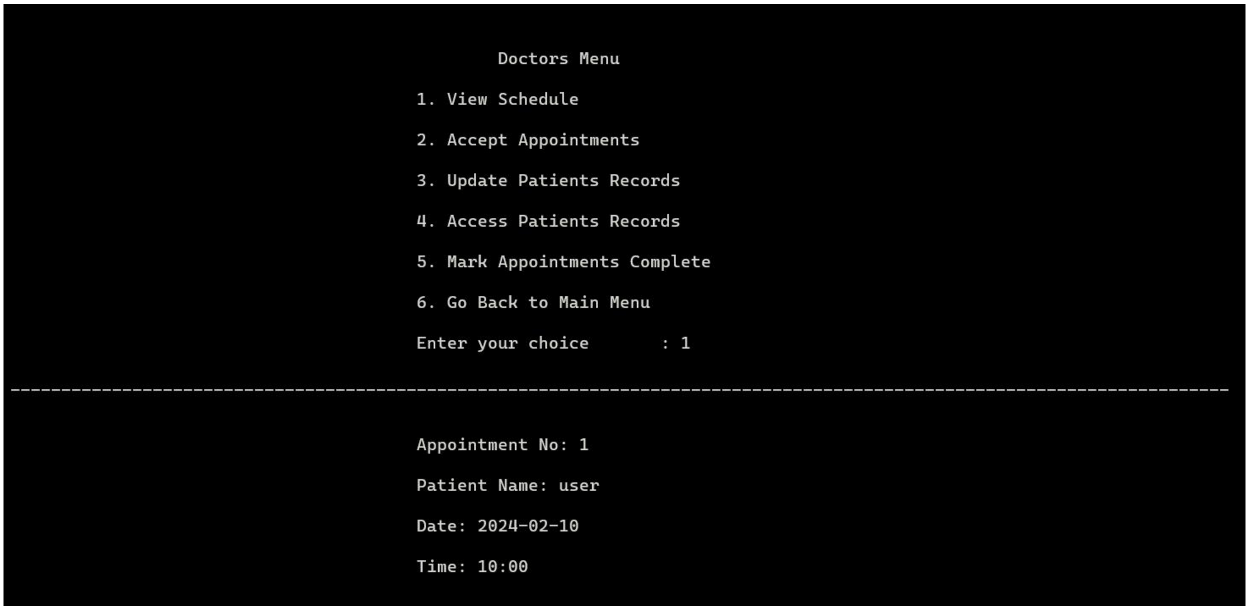


```
                                        Patients Menu
                                1. View medical records
                                2. Schedule Appointments
                                3. Cancel Appointments
                                4. Update Personal information
                                5. Display Appointments
                                6. Go Back to Main Menu
                                Enter your choice       : 5
                                Patient's Appointments:
------------------------------------------------------------------------------------------------------------------------

                                Patient UserName: user
                                Doctor Name: Dr.user
                                Date: 2024-02-10
                                Time: 15:00
```

The patient can view the appointments that has been accepted by the doctor using this function.

Doctor Functions

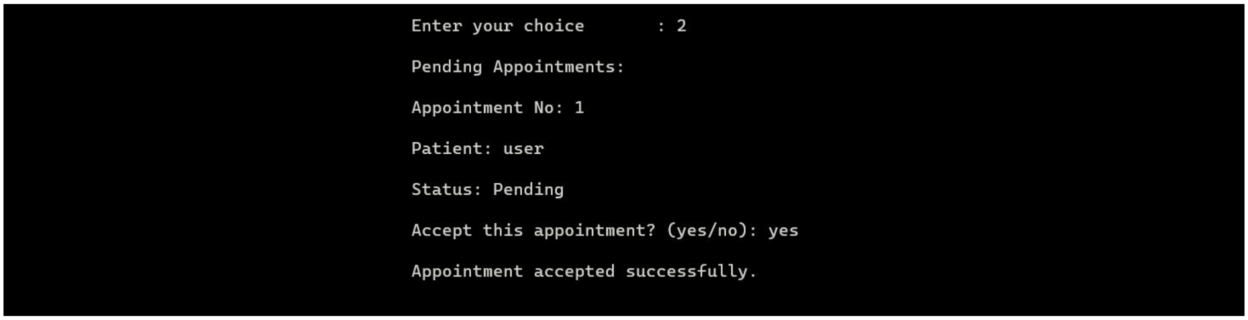
- View Schedule



```
                                               Doctors Menu
                                       1. View Schedule
                                       2. Accept Appointments
                                       3. Update Patients Records
                                       4. Access Patients Records
                                       5. Mark Appointments Complete
                                       6. Go Back to Main Menu
                                       Enter your choice       : 1
------------------------------------------------------------------------------------------------------------------------

                                       Appointment No: 1
                                       Patient Name: user
                                       Date: 2024-02-10
                                       Time: 10:00
```

By selecting the choice 1 , the doctor can view the accepted appointments, with the index , name of the patient , date and the time.

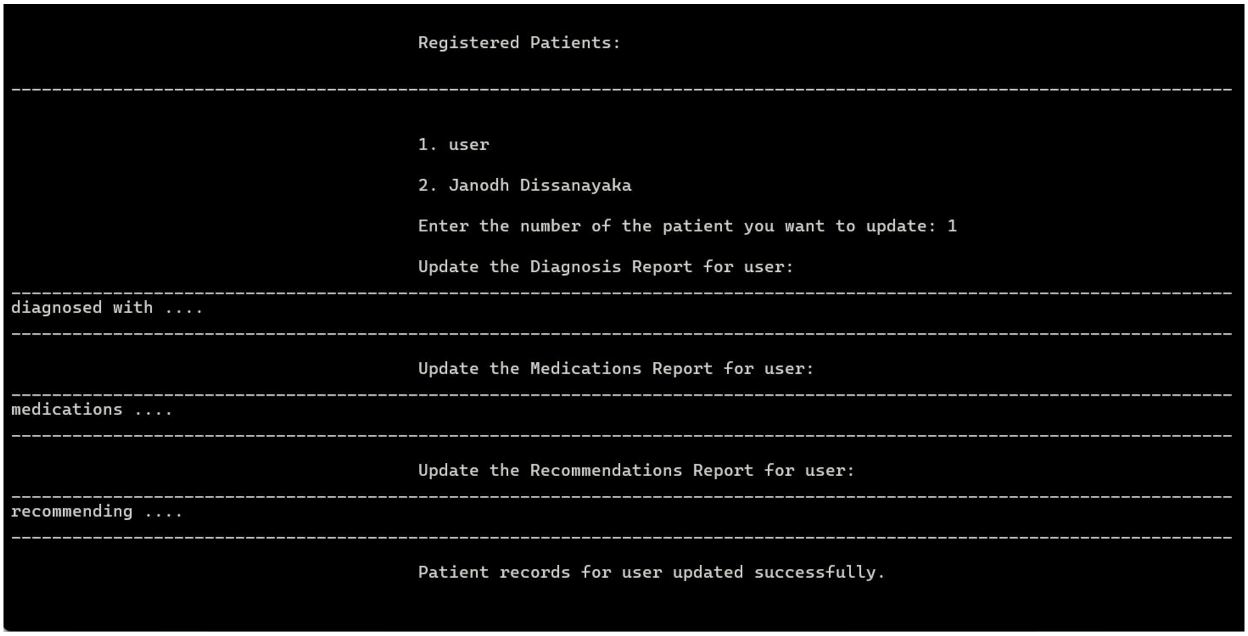
- Accept Appointments (Additional Feature)



```
                                       Enter your choice       : 2
                                       Pending Appointments:
                                       Appointment No: 1
                                       Patient: user
                                       Status: Pending
                                       Accept this appointment? (yes/no): yes
                                       Appointment accepted successfully.
```

Now the doctor can accept the appointments that are scheduled to them.

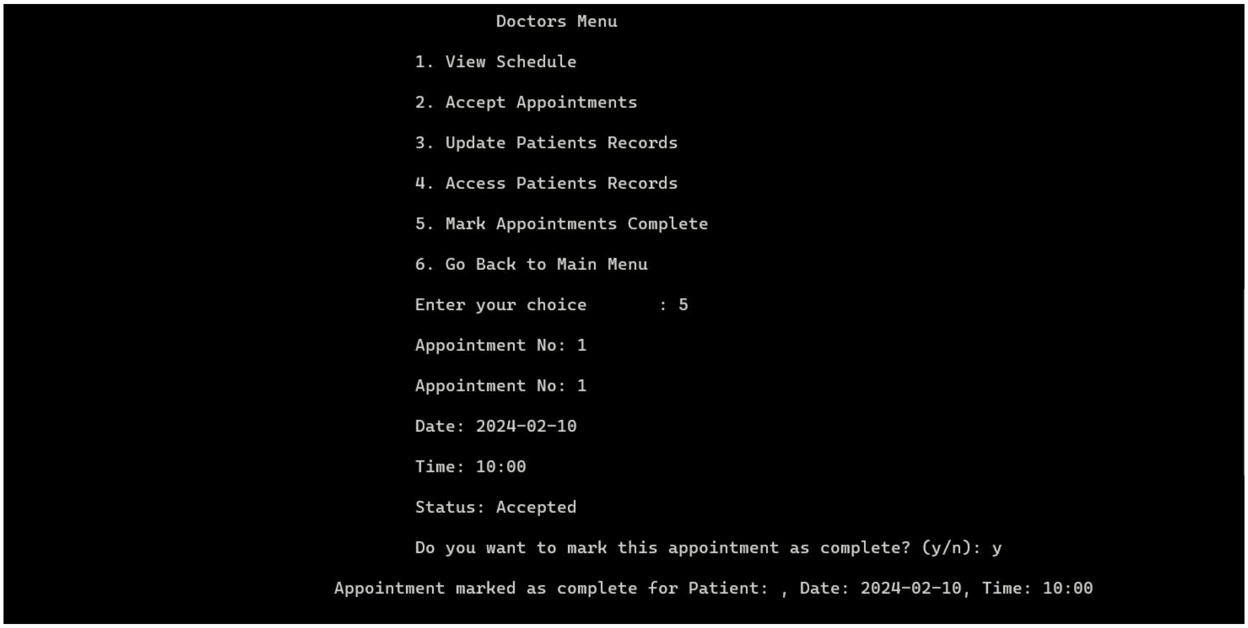
- Update Patients Records



```
                                        Registered Patients:

------------------------------------------------------------------------------------------------------------------------


                                        1. user

                                        2. Janodh Dissanayaka

                                        Enter the number of the patient you want to update: 1

                                        Update the Diagnosis Report for user:
------------------------------------------------------------------------------------------------------------------------
diagnosed with ....
------------------------------------------------------------------------------------------------------------------------

                                        Update the Medications Report for user:
------------------------------------------------------------------------------------------------------------------------
medications ....
------------------------------------------------------------------------------------------------------------------------

                                        Update the Recommendations Report for user:
------------------------------------------------------------------------------------------------------------------------
recommending ....
------------------------------------------------------------------------------------------------------------------------

                                        Patient records for user updated successfully.
```

The patient record contains three reports diagnosis report, medications report and the recommendations report. The doctor can update these , but first the doctor should select the patient by entering their index number.

- Mark Appointment Complete

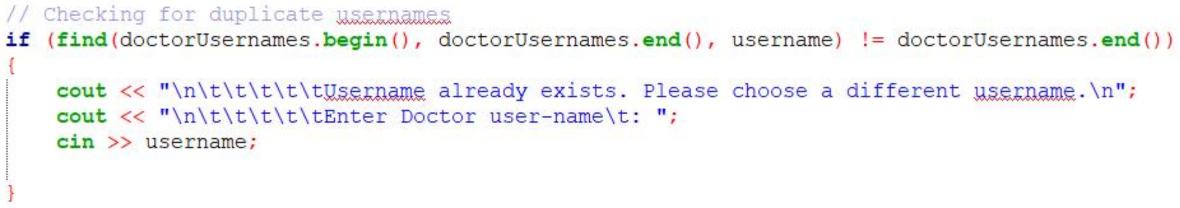


```
                            Doctors Menu
                    1. View Schedule
                    2. Accept Appointments
                    3. Update Patients Records
                    4. Access Patients Records
                    5. Mark Appointments Complete
                    6. Go Back to Main Menu
                    Enter your choice       : 5
                    Appointment No: 1
                    Appointment No: 1
                    Date: 2024-02-10
                    Time: 10:00
                    Status: Accepted
                    Do you want to mark this appointment as complete? (y/n): y
                Appointment marked as complete for Patient: , Date: 2024-02-10, Time: 10:00
```

When the appointment is completed the doctor can select the choice 5 and mark it as completed , then the appointment will be removed from the accepted file and store in the completed file.

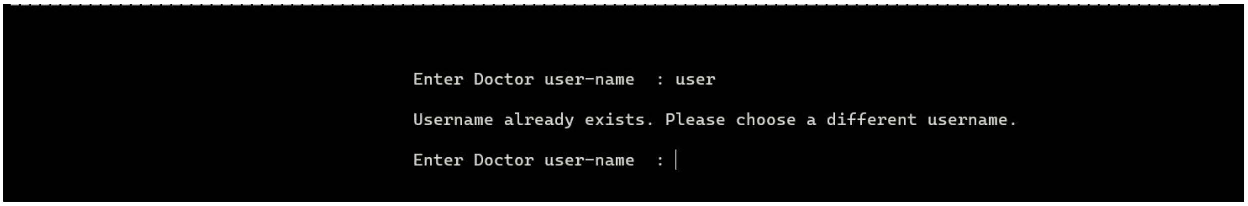
DATA VALIDATIONS AND ERROR HANDLING

- Checking for duplicate user names



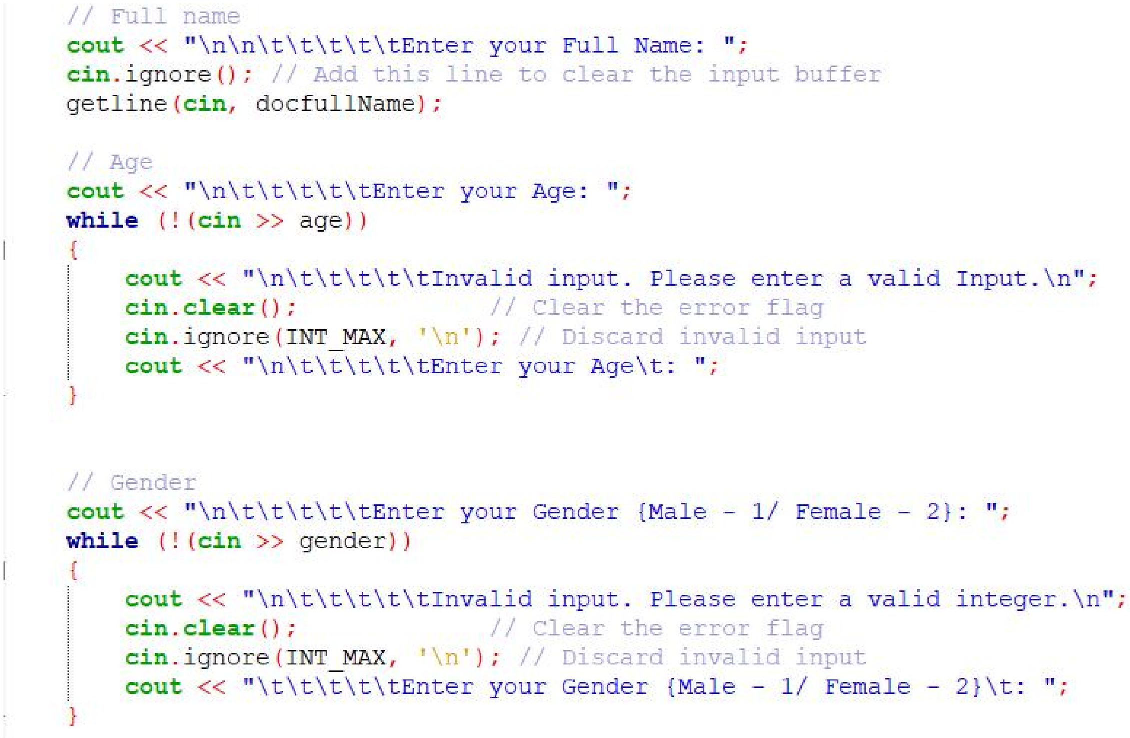
```
// Checking for duplicate usernames
if (find(doctorUsernames.begin(), doctorUsernames.end(), username) != doctorUsernames.end())
{
    cout << "\n\t\t\t\t\tUsername already exists. Please choose a different username.\n";
    cout << "\n\t\t\t\t\tEnter Doctor user-name\t: ";
    cin >> username;

}
```

The entering usernames are pushed back to a vector called "doctorUsernames", the vector is being searched from the beginning to the end whether there is a username same to the entering username. And if it exists a message will be displayed and you should enter another username.



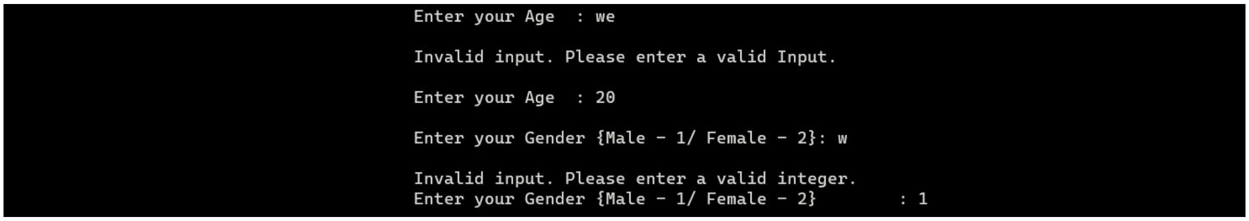
```
                                        Enter Doctor user-name  : user

                                        Username already exists. Please choose a different username.

                                        Enter Doctor user-name  : |
```

- <u>Data Validation in Registration</u>



```
// Full name
cout << "\n\n\t\t\t\t\tEnter your Full Name: ";
cin.ignore(); // Add this line to clear the input buffer
getline(cin, docfullName);

// Age
cout << "\n\t\t\t\t\tEnter your Age: ";
while (!(cin >> age))
{
    cout << "\n\t\t\t\t\tInvalid input. Please enter a valid Input.\n";
    cin.clear();              // Clear the error flag
    cin.ignore(INT_MAX, '\n'); // Discard invalid input
    cout << "\n\t\t\t\t\tEnter your Age\t: ";
}


// Gender
cout << "\n\t\t\t\t\tEnter your Gender {Male - 1/ Female - 2}: ";
while (!(cin >> gender))
{
    cout << "\n\t\t\t\t\tInvalid input. Please enter a valid integer.\n";
    cin.clear();              // Clear the error flag
    cin.ignore(INT_MAX, '\n'); // Discard invalid input
    cout << "\t\t\t\t\tEnter your Gender {Male - 1/ Female - 2}\t: ";
}
```

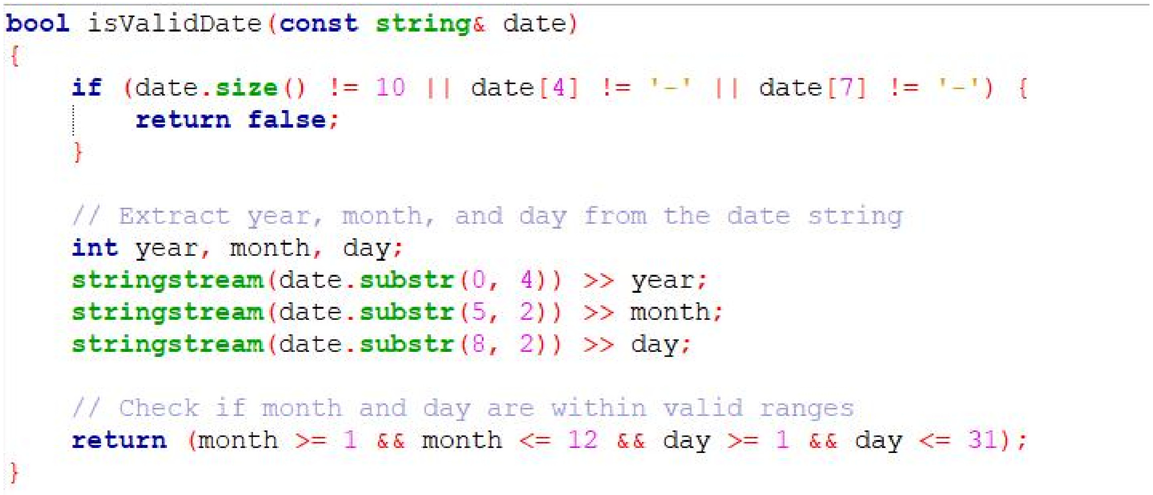
Data validation is provided for every detail that the user enter to the system.



```
Enter your Age  : we

Invalid input. Please enter a valid Input.

Enter your Age  : 20

Enter your Gender {Male - 1/ Female - 2}: w

Invalid input. Please enter a valid integer.
Enter your Gender {Male - 1/ Female - 2}        : 1
```

- Error Handling in Schedule Appointments function

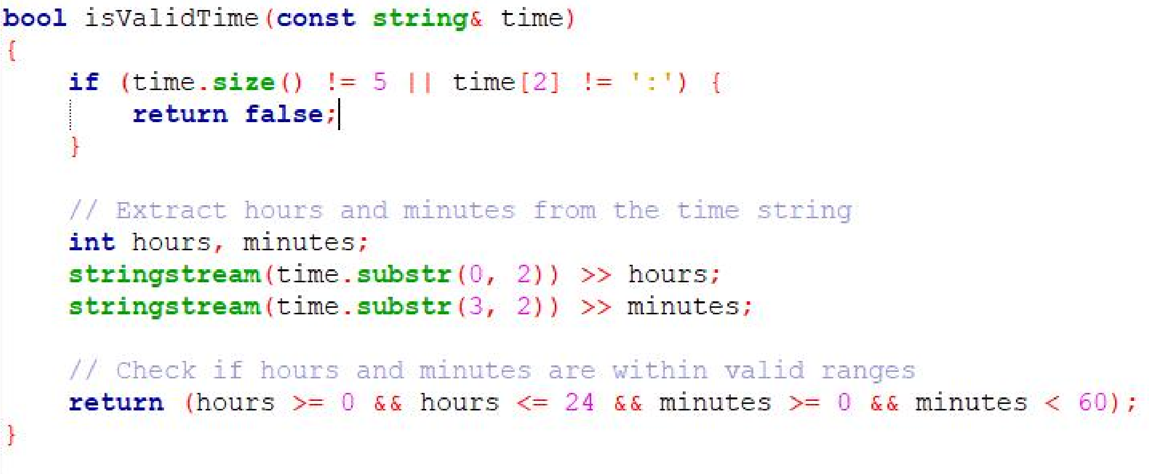
Checking for a valid appointment date



```cpp
bool isValidDate(const string& date)
{
    if (date.size() != 10 || date[4] != '-' || date[7] != '-') {
        return false;
    }

    // Extract year, month, and day from the date string
    int year, month, day;
    stringstream(date.substr(0, 4)) >> year;
    stringstream(date.substr(5, 2)) >> month;
    stringstream(date.substr(8, 2)) >> day;

    // Check if month and day are within valid ranges
    return (month >= 1 && month <= 12 && day >= 1 && day <= 31);
}
```

In this function we’re checking whether the user is inputting a valid date or not. The year should only contain 4 digits, the month should be greater than or equal to 1 and less than and equal to 12. The day should be greater than or equal to 1 and less than and equal to 31.

Checking for a valid appointment time



```cpp
bool isValidTime(const string& time)
{
    if (time.size() != 5 || time[2] != ':') {
        return false;
    }

    // Extract hours and minutes from the time string
    int hours, minutes;
    stringstream(time.substr(0, 2)) >> hours;
    stringstream(time.substr(3, 2)) >> minutes;

    // Check if hours and minutes are within valid ranges
    return (hours >= 0 && hours <= 24 && minutes >= 0 && minutes < 60);
}
```

In this function we’re checking whether the user is inputting a valid time or not. The hours should be greater than or equal 0 and less than or equal to 24. Minutes should be greater than or equal 0 and less than 60.

TESTING

TEST PLAN

Clinical Management System	
Test Plan ID	*01*
Brief Introduction about the system.	*The Clinical Management System is designed to streamline the management of patient records, appointments, and medical reports in a health care facility. This is designed to both doctors and patients. The main services that can be done are schedule appointments and manage patient reports*
Test Objectives	*The objective of the test plan is to ensure that this Clinical Management System functions including doctor functions and patient functions to meet the requirements and expectations.*
Features to be tested	*Registration of Doctor* *Registration of Patient* *Login of Doctor* *Login of Patient* *Scheduling Appointments* *Canceling Appointments* *Updating Personal information* *Accepting Appointments* *Update Patient Records* *Marking Appointment as complete*
Test Environment	*Personal Computer (Laptop – Acer Aspire 3), Code Blocks*

Test Approach	*Black box testing*
Testing Tasks	*Test planning, Test Design, Test development, Test execution, Test evaluation*

Test deliverables	*Test plan, Test Environment, Test summary, Test result, Test Evaluation report*
Schedule	

TEST CASES

Test Case - Registration of Doctor	
Test Unit: *Registration of Doctor*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *01*	***Test Type:*** *Black Box*
Test Description: *Registering a doctor to the System for the first time.*	***Test Execution Date:*** *10/02/2024*
Test Title: *Register Doctor*	***Test Execution Time:*** *7:13 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Entering the username, password, name, age, gender and phone number*	*01*	*Doctor Username:anura123* *Password:123* *Full name:Dr.anura* *Age:40* *Gender: 1 (Male)* *Phone number:0777106268*	*Doctor account created successfully!*	*Doctor account created successfully!*	*Pass*

Test Case - Registration of Patient	
Test Unit: *Registration of Patient*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *02*	***Test Type:*** *Black Box*
Test Description: *Registering a patient to the System for the first time.*	***Test Execution Date:*** *10/02/2024*
Test Title: *Register patient*	***Test Execution Time:*** *7:23 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Entering the username, password, name, age, gender, blood group and phone number*	*02*	*Patient Username:janodh123* *Password:123* *Full name:Janodh dissanayaka* *Age:20* *Gender: 1 (Male)* *Blood Group:B+* *Phone number:0762008458*	*Patient account created successfully!*	*Patient account created successfully!*	*Pass*

Test Case - Login of Doctor	
Test Unit: *Login of Doctor*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *03*	***Test Type:*** *Black Box*
Test Description: *logging in using the registered Doctor username and password*	***Test Execution Date:*** *10/02/2024*
Test Title: *Login doctor*	***Test Execution Time:*** *7:35 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Entering the correct username and correct password*	*03*	*Username:anura123* *Password:123*	*Welcome anura123*	*Welcome anura123*	*Pass*
02	*Entering a incorrect username and a incorrect password*	*03*	*Username:anura1* *Password:124*	*Login failed. Please check the username and password*	*Login failed. Please check the username and password*	*Pass*
03	*Entering the correct username and a incorrect password*	*03*	*Username:anura123* *Password:124*	*Login failed. Please check the username and password*	*Login failed. Please check the username and password*	*Pass*
04	*Entering a incorrect username and the correct password*	*03*	*Username:anura1* *Password:123*	*Login failed. Please check the username and password*	*Login failed. Please check the username and password*	*Pass*

Test Case - Login of Patient	
Test Unit: *Login of Patient*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *04*	***Test Type:*** *Black Box*
Test Description: *logging in using the registered patient username and password*	***Test Execution Date:*** *10/02/2024*
Test Title: *Login patient*	***Test Execution Time:*** *7:47 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Entering the correct username and correct password*	*04*	*Username:janodh123* *Password:123*	*Welcome janodh123*	*Welcome janodh123*	*Pass*
02	*Entering a incorrect username and a incorrect password*	*04*	*Username:janodh1* *Password:124*	*Login failed. Please check the username and password*	*Login failed. Please check the username and password*	*Pass*
03	*Entering the correct username and a incorrect password*	*04*	*Username:janodh123* *Password:124*	*Login failed. Please check the username and password*	*Login failed. Please check the username and password*	*Pass*
04	*Entering a incorrect username and the correct password*	*04*	*Username:janodh1* *Password:123*	*Login failed. Please check the username and password*	*Login failed. Please check the username and password*	*Pass*

Test Case - Schedule Appointments	
Test Unit: *Scheduling appointments*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *05*	***Test Type:*** *Black Box*
Test Description: *patient scheduling an Appointment to a selected doctor*	***Test Execution Date:*** *10/02/2024*
Test Title: *Schedule Appointment*	***Test Execution Time:*** *7:51 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Selecting an available Doctor by giving the Correct index*	*05*	*Select a doctor* *By entering the corresponding number:1 (Dr.Anura)*	*Display of "Enter the appointment date"*	*Display of "Enter the appointment date"*	*Pass*
02	*Selecting an available Doctor by giving the incorrect index*	*05*	*Select a doctor* *By entering the corresponding number:2*	*Display of "Invalid input. Please enter a valid number"*	*Display of "Invalid input. Please enter a valid number"*	*Pass*
03	*Enter a valid date as the appointment date*	*05*	*Enter the appointment Date:2024-02-10*	*Display of "Enter the appointment time: "*	*Display of "Enter the appointment time: "*	*Pass*
04	*Enter a invalid date as the appointment date*	*05*	*Enter the appointment Date:2024-13-10*	*Display of "Invalid date format"*	*Display of "Invalid date format"*	*Pass*

Test Case - Cancel Appointments	
Test Unit: *Canceling a pending appointment*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *06*	***Test Type:*** *Black Box*
Test Description: *patient should be able to cancel a pending appointment*	***Test Execution Date:*** *10/02/2024*
Test Title: *Cancel Appointment*	***Test Execution Time:*** *8:33 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Entering the number of the appointment that should be canceled.And giving ‘y’ to confirm it*	*06*	*Enter the number of the appointment you want to cancel (or 0 to cancel nothing):1* *Are You Sure You Want To Delete This Appointment (y/n)? y*	*Displaying* *“The Appointment Has Been Successfully Canceled.”*	*Displaying* *“The Appointment Has Been Successfully Canceled.”*	*Pass*
02	*Entering 0 to cancel nothing*	*06*	*Enter the number of the appointment you want to cancel (or 0 to cancel nothing):0*	*Displaying* *“No appointments were canceled.”*	*Displaying* *“No appointments were canceled.”*	*Pass*
03	*Giving ‘n’ to cancel nothing*	*06*	*Are You Sure You Want To Delete This Appointment (y/n)? n*	*Displaying* *“No appointments were canceled.”*	*Displaying* *“No appointments were canceled.”*	*Pass*

Test Case - Update Personal Information	
Test Unit: *Updating personal information of a patient*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *07*	***Test Type:*** *Black Box*
Test Description: *letting the patient to update the personal information that was given in the registration*	***Test Execution Date:*** *10/02/2024*
Test Title: *Update personal Information*	***Test Execution Time:*** *9:06 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Entering the new name, new age, gender, blood group and the new phone number*	07	*Enter new name: Chandupa* *Enter new age:21* *Enter Gender:male* *Enter Blood Group:B+* *Enter new phone:0777706267*	*Displaying “Personal Information updated successfully.”*	*Displaying “Personal Information updated successfully.”*	*Pass*

Test Case - Accept Appointments(Additional feature)	
Test Unit: *Accepting Appointments*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *08*	***Test Type:*** *Black Box*
Test Description: *Doctor should be able to Accept the pending appointments scheduled by Patients.*	***Test Execution Date:*** *10/02/2024*
Test Title: *Accept Appointments*	***Test Execution Time:*** *9:45 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Accepting the displayed pending appointment.*	*08*	*Accept this appointment? (yes/no): yes*	*Appointment accepted successfully.*	*Appointment accepted successfully.*	*Pass*
02	*Giving "no" to accept this appointment*	*08*	*Accept this appointment? (yes/no): no*	*No appointment accepted.*	*No appointment accepted.*	*Pass*

Test Case - Mark Appointments Complete	
Test Unit: *Mark Appointments complete*	***Tester:*** *Janodh dissanayaka*
Test Case ID: *09*	***Test Type:*** *Black Box*
Test Description: *When an appointment is completed , the doctor must be able to mark it as completed*	***Test Execution Date:*** *10/02/2024*
Test Title: *Mark appointment complete*	***Test Execution Time:*** *7:51 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Accepting an appointment by giving ‘y’ to confirm it.*	09	*Do you want to mark this appointment as complete? (y/n): y*	*Appointment marked as complete*	*Appointment marked as complete*	*Pass*

Test Case - Update Patient Records	
Test Unit:	***Tester:*** *Janodh dissanayaka*
Test Case ID: *10*	***Test Type:*** *Black Box*
Test Description: *patient scheduling an Appointment to a selected doctor*	***Test Execution Date:*** *10/02/2024*
Test Title: *Schedule Appointments*	***Test Execution Time:*** *7:51 p.m.*

Step No.	**Test Step**	**Test Case ID**	**Test Input**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**
01	*Selecting a registered patient from the ID and entering data to the reports*	10		*Patient records for “Patient name” updated successfully.*	*Patient records for Chandupa updated successfully.*	*Pass*

Results and Discussion

Summary of Project Outcomes: This Clinical Management System project has resulted in the successful development of a comprehensive software solution aimed at optimizing the management of clinical information, patient records, appointments, and administrative tasks within health care facilities. The system provides a user-friendly interface accessible to both doctors and patients, offering efficient appointment scheduling, comprehensive patient record management, secure user authentication, and effective schedule management for doctors.

Comparison with Initial Objectives: The outcomes of the project closely align with the initial objectives set forth during the beginning of this project. Efficient appointment scheduling functionality has been implemented, allowing patients to easily schedule, cancel, and view their appointments. Comprehensive patient record management features enable doctors to securely update and access patient records, including diagnosis reports, medication reports, and recommendation reports. Secure user authentication with hashed passwords ensures the confidentiality of patient data. Additionally, effective schedule management tools have been provided for doctors to keep track of their schedules and appointments.

Challenges Faced and Lessons Learned: Throughout the development process, several challenges were encountered, primarily related to file handling operations. Working with file I/O in C++ required careful attention to error handling and data integrity to ensure the reliable storage and retrieval of patient and doctor information. Overcoming these challenges involved thorough testing and debugging of file handling functionalities to identify and resolve potential issues.

Moreover, learning and implementing new tools and frameworks, such as vectors and unordered_map from the C++ Standard Template Library (STL), presented a learning curve. However, these data structures proved invaluable for efficiently managing collections of objects and storing user accounts, respectively. The challenges faced during the project underscored the importance of thorough research, testing, and iterative development practices.

Overall, the project provided valuable insights into software development methodologies, tools, and frameworks, enhancing proficiency in C++ programming, file handling, data structures, and user interface design. Through overcoming challenges and applying lessons learned, the project successfully delivered a robust and user-friendly Clinical Management System tailored to meet the needs of health care facilities.

References

Websites and Learning Platforms

- https://www.geeksforgeeks.org
- https://www.w3schools.com