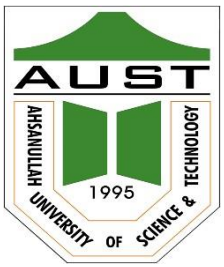

Ahsanullah University of Science and Technology (AUST)
Department of Electrical and Electronic Engineering

Course No.: EEE 1110

Course Title: Programming Language Laboratory

Doctor Appointment System

ID: **20230205181**

Name: **Tanjila Eity**

Section: **C2**

For the students of the

Department of Electrical and Electronic Engineering

1st Year, 1st Semester

Overview

The Doctor Appointment System is designed to efficiently manage patient appointments in a medical facility. It provides functionalities for adding, viewing, searching, and deleting patient appointments, ensuring a smooth and organized experience for both patients and healthcare providers.

Objective

The primary objective of this system is to maintain a record of patient appointments, including personal details such as name, gender, age, blood group, phone number, doctor assigned, appointment date, and serial number. The system aims to streamline the appointment management process, reduce errors, and enhance patient service.

Design and Implementation

Structures and Functions

Patient Structure: the details of a patient appointment.

- char name[50]: Stores the patient's name.
- char gender[10]: Stores the patient's gender.
- char blood[10]: Stores the patient's blood group.
- char phone[15]: Stores the patient's phone number.
- char doctorName[50]: Stores the name of the assigned doctor.
- char date[50]: Stores the appointment date.
- int id: Stores a unique identifier for the appointment.
- int age: Stores the patient's age.
- int serial: Stores the appointment serial number.

displayMenu () Function

Purpose: To display the main menu options to the user.

Functionality: Lists available actions such as adding an appointment, showing appointments, searching by ID, and deleting an appointment.

Validation Functions

- **isValidGender ()**: Validates gender input to ensure it is either "Male", "Female", or "Other".
- **isValidAge ()**: Validates age input to be between 1 and 120.
- **isValidBloodGroup ()**: Validates blood group input against a predefined list.
- **isValidPhoneNumber ()**: Validates that the phone number is exactly 11 digits and numeric.
- **invalidate ()**: Checks the date format (YYYY-MM-DD) and ensures the values are realistic.

AddPatient () Function

Purpose: To append a new patient appointment record to a text file.

Functionality: Opens a file in append mode and writes the patient details.

ShowPatient () Function

Purpose: To display all patient appointments stored in the text file.

Functionality: Reads from the file, populates a vector of Patient structs, and formats the output for readability.

searchAppointment () Function

Purpose: To find and display a specific appointment by ID.

Functionality: Reads through the file line by line and matches the input ID with stored records.

DeleteAppointment () Function

Purpose: To delete a specific appointment by ID.

Functionality: Reads the file, writes non-matching records to a temporary file, and replaces the original file with the updated one.

Problems Encountered

During the development of the system, a few challenges were encountered:

- **Input Validation**: Ensuring user input was valid required careful attention to detail, particularly for the phone number and date fields.
- **File Handling**: Managing file read/write operations without losing data proved tricky, especially during deletion and updates.
- **Data Formatting**: Ensuring proper alignment of output data required additional formatting considerations.

Future Findings

- **User Interface Enhancements**: Implementing a graphical user interface (GUI) could improve user experience.
- **Database Integration**: Transitioning from file-based storage to a database system would enhance data integrity and scalability.
- **Additional Features**: Implementing features like appointment reminders and a search filter by date could provide additional functionality.

Key Features

- User-friendly text-based menu for navigation.
- Robust input validation to minimize errors.
- Comprehensive appointment management capabilities.
- Clear and formatted output display for easy readability.

Conclusion

The Doctor Appointment Management System successfully meets its objective of efficiently managing patient appointments. Through careful design and implementation, it provides essential features for users while maintaining a focus on data integrity and user experience. Future enhancements could further improve the system's functionality and usability.