

**Object Oriented Programming**

**Project Document**

**MediMeet (Doctor Appointment System)**

**SUBMITTED BY:**

Hasaan Ahmad SP22-BSE-017

Muhammad Mujtaba SP22-BSE-036

Muhammad Haider Sheikh SP22-BSE-033

**Supervisor: Sir Muzaffar Iqbal**

**Doctor Appointment Management System**

**Description:**

The Doctor Appointment Management System is a comprehensive and user-friendly Java app designed to streamline and automate the process of managing appointments between doctors and patients. It provides a convenient platform for doctors, patients, and administrators to interact, schedule appointments, and maintain essential medical records.

**Key Features:**

1. User Registration and Login: The system allows doctors, patients, and administrators to register their profiles and securely log in using their credentials.
2. Doctor Dashboard: Doctors can access their personalized dashboard to manage their appointments, update availability, view patient details, and prescribe medications. They can also maintain their professional profiles, including specialization, qualifications, and experience.
3. Patient Dashboard: Patients have their own dashboard to schedule appointments with preferred doctors, view upcoming and past appointments, access prescriptions, and update personal information.
4. Appointment Scheduling: Patients can search for doctors based on specialization, view their availability, and book appointments at suitable dates and times. The system also provides automated reminders to patients regarding their scheduled appointments.
5. Admin Panel: Administrators have administrative privileges to manage the system, including user registration, doctor profiles, appointment management, and generating reports. They can add or remove doctors from the system and oversee the overall functionality.
6. Prescription Management: Doctors can generate and store electronic prescriptions for patients, including details of prescribed medications, dosage instructions, and duration. Patients can access and download their prescriptions for reference and follow-up purposes.

**Student Contributions:**

**Hasaan Ahmad:**

Hasaan Ahmad has been primarily responsible for developing the Doctor class, which includes the sign-in and registration system of the application. He has implemented the login functionality, where doctors can enter their credentials and access their personalized dashboard. Additionally, Hasaan has worked extensively on the inheritance aspect of the project, ensuring that the Doctor class inherits properties and methods from the User class. This inheritance mechanism allows for the reuse of common attributes and behaviors across different user types in the system.

**Muhammad Mujtaba:**

Muhammad Mujtaba has played a key role in handling file handling operations within the project. He has focused on implementing file input/output operations for storing and retrieving data related to doctors, patients, appointments, and other relevant information. Mujtaba has ensured the proper organization and management of data in files, allowing for efficient data retrieval and storage. Moreover, he has also worked on the prescription handling functionality for doctors, ensuring that prescriptions are properly recorded and associated with the respective patients.

**Muhammad Haider Sheikh:**

Muhammad Haider Sheikh has taken the lead in developing the graphical user interface (GUI) for the project. He has designed and implemented the user interfaces that provide an intuitive and user-friendly experience for both doctors and patients. Haider has worked on creating interactive screens for viewing appointments, accessing patient information, and managing doctor profiles. Additionally, he has made contributions to the composition aspect of the prescription and appointment classes, ensuring that these classes collaborate effectively to store and retrieve necessary information

**Important Classes:**

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Arrays;

import java.util.Scanner;

public class Doctor extends User {

  private int doctorID;

  private String specialization;

  private String hospital;

  private String department;

  private String qualification;

  private String experience;

  private String consultationFee;

  private int[] currentAppointments; // Open appointments which this doctor is currently part of

  public Doctor(

      String name,

      String email,

      String password,

      String phone,

      String address,

      int doctorID,

      String specialization,

      String hospital,

      String department,

      String qualification,

      String experience,

      String consultationFee,

      int[] currentAppointments) {

    super(name, email, password, phone, address);

    this.doctorID = doctorID;

    this.specialization = specialization;

    this.hospital = hospital;

    this.department = department;

    this.qualification = qualification;

    this.experience = experience;

    this.consultationFee = consultationFee;

    this.currentAppointments = currentAppointments;

  }

  public static void menu() {

    // It should have Login, Signup, and Exit options

    System.out.println("1. Login");

    System.out.println("2. Signup");

    System.out.println("3. Exit");

    // Switch cases choice

    Scanner input = new Scanner(System.in);

    System.out.println("Enter your choice: ");

    int choice = Integer.parseInt(input.nextLine());

    switch (choice) {

      case 1:

        // Login

        login();

        break;

      case 2:

        // Signup

        signup();

        break;

      case 3:

        // Exit

        System.exit(0);

        break;

      default:

        System.out.println("Invalid choice!");

        break;

    }

  }

  public static void login() {

    // Login using doctorID and password. fetch Details from text file

    Scanner input = new Scanner(System.in);

    System.out.println("Enter your doctorID: ");

    int doctorID = Integer.parseInt(input.nextLine());

    System.out.println("Enter your password: ");

    String password = input.nextLine();

    // Read from file

    try {

      // Also make object of Doctor class whose details matches with the doctorID and

      // password

      File myObj = new File("doctors.txt");

      Scanner sc = new Scanner(myObj);

      while (sc.hasNextLine()) {

        String data = sc.nextLine();

        String[] doctorDetails = data.split(",");

        if (doctorDetails[5].equals(doctorID) && doctorDetails[2].equals(password)) {

          // Login Successful

          System.out.println("Login Successful");

          // Create a doctor object

          Doctor d = new Doctor(

              doctorDetails[0],

              doctorDetails[1],

              doctorDetails[2],

              doctorDetails[3],

              doctorDetails[4],

              Integer.parseInt(doctorDetails[5]),

              doctorDetails[6],

              doctorDetails[7],

              doctorDetails[8],

              doctorDetails[9],

              doctorDetails[10],

              doctorDetails[11],

              new int[0]);

          // Call doctorMenu

          doctorMenu(d);

          break;

        }

      }

    } catch (Exception e) {

      System.out.println("Error occured while saving");

    }

  }

  public static void signup() {

    // Register a new doctor

    Scanner input = new Scanner(System.in);

    System.out.println("Enter your name: ");

    String name = input.nextLine();

    System.out.println("Enter your email: ");

    String email = input.nextLine();

    System.out.println("Enter your password: ");

    String password = input.nextLine();

    System.out.println("Enter your phone: ");

    String phone = input.nextLine();

    System.out.println("Enter your address: ");

    String address = input.nextLine();

    System.out.println("Enter your doctorID: ");

    int doctorID = Integer.parseInt(input.nextLine());

    System.out.println("Enter your specialization: ");

    String specialization = input.nextLine();

    System.out.println("Enter your hospital: ");

    String hospital = input.nextLine();

    System.out.println("Enter your department: ");

    String department = input.nextLine();

    System.out.println("Enter your qualification: ");

    String qualification = input.nextLine();

    System.out.println("Enter your experience: ");

    String experience = input.nextLine();

    System.out.println("Enter your consultationFee: ");

    String consultationFee = input.nextLine();

    int[] currentAppointments = new int[0];

    Doctor doctor = new Doctor(

        name,

        email,

        password,

        phone,

        address,

        doctorID,

        specialization,

        hospital,

        department,

        qualification,

        experience,

        consultationFee,

        currentAppointments);

    // Write to object to the file

    try {

      FileWriter fw = new FileWriter("doctors.txt", true);

      fw.write(doctor.toString() + "\n");

      fw.close();

    } catch (IOException e) {

      System.out.println("Error writing to file doctors.txt");

      e.printStackTrace();

    }

  }

  public static void doctorMenu(Doctor d) {

    // It should have the following options:

    // 1. View Appointments

    // 2. Add availability

    // 3. Remove availability

    // 4. Logout

    System.out.println("Welcome " + d.getName() + " to the Doctor's Portal");

    System.out.println("1. View Appointments");

    System.out.println("2. Add availability");

    System.out.println("3. Remove availability");

    System.out.println("4. Logout");

    System.out.println("Please choose an option:");

    // Switch cases choice

  }

  public int getDoctorID() {

    return doctorID;

  }

  public void setDoctorID(int doctorID) {

    this.doctorID = doctorID;

  }

  public String getSpecialization() {

    return specialization;

  }

  public void setSpecialization(String specialization) {

    this.specialization = specialization;

  }

  public String getHospital() {

    return hospital;

  }

  public void setHospital(String hospital) {

    this.hospital = hospital;

  }

  public String getDepartment() {

    return department;

  }

  public void setDepartment(String department) {

    this.department = department;

  }

  public String getQualification() {

    return qualification;

  }

  public void setQualification(String qualification) {

    this.qualification = qualification;

  }

  public String getExperience() {

    return experience;

  }

  public void setExperience(String experience) {

    this.experience = experience;

  }

  public String getConsultationFee() {

    return consultationFee;

  }

  public void setConsultationFee(String consultationFee) {

    this.consultationFee = consultationFee;

  }

  public int[] getCurrentAppointments() {

    return currentAppointments;

  }

  public void setCurrentAppointments(int[] currentAppointments) {

    this.currentAppointments = currentAppointments;

  }

  @Override

  public String toString() {

    return (super.toString() + "Doctor [doctorID=" + doctorID + ", specialization=" + specialization + ", hospital="

        + hospital

        + ", department=" + department + ", qualification=" + qualification + ", experience=" + experience

        + ", consultationFee=" + consultationFee + ", currentAppointments=" + Arrays.toString(currentAppointments)

        + "]");

  }

}

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class Patient extends User {

  private String patientID;

  private String diagnosis;

  private String prescription;

  private int doctorID;

  private String appointmentDate;

  private String appointmentTime;

  private String appointmentStatus;

  private String appointmentID;

  private String appointmentFee;

  public Patient(

    String name,

    String email,

    String password,

    String phone,

    String address,

    String patientID,

    String diagnosis,

    String prescription,

    int doctorID,

    String appointmentDate,

    String appointmentTime,

    String appointmentStatus,

    String appointmentID,

    String appointmentFee

  ) {

    super(name, email, password, phone, address);

    this.patientID = patientID;

    this.diagnosis = diagnosis;

    this.prescription = prescription;

    this.doctorID = doctorID;

    this.appointmentDate = appointmentDate;

    this.appointmentTime = appointmentTime;

    this.appointmentStatus = appointmentStatus;

    this.appointmentID = appointmentID;

    this.appointmentFee = appointmentFee;

  }

  public String getPatientID() {

    return patientID;

  }

  public void setPatientID(String patientID) {

    this.patientID = patientID;

  }

  public String getDiagnosis() {

    return diagnosis;

  }

  public void setDiagnosis(String diagnosis) {

    this.diagnosis = diagnosis;

  }

  public String getPrescription() {

    return prescription;

  }

  public void setPrescription(String prescription) {

    this.prescription = prescription;

  }

  public int getDoctorID() {

    return doctorID;

  }

  public void setDoctorID(int doctorID) {

    this.doctorID = doctorID;

  }

  public String getAppointmentDate() {

    return appointmentDate;

  }

  public void setAppointmentDate(String appointmentDate) {

    this.appointmentDate = appointmentDate;

  }

  public String getAppointmentTime() {

    return appointmentTime;

  }

  public void setAppointmentTime(String appointmentTime) {

    this.appointmentTime = appointmentTime;

  }

  public String getAppointmentStatus() {

    return appointmentStatus;

  }

  public void setAppointmentStatus(String appointmentStatus) {

    this.appointmentStatus = appointmentStatus;

  }

  public String getAppointmentID() {

    return appointmentID;

  }

  public void setAppointmentID(String appointmentID) {

    this.appointmentID = appointmentID;

  }

  public String getAppointmentFee() {

    return appointmentFee;

  }

  public void setAppointmentFee(String appointmentFee) {

    this.appointmentFee = appointmentFee;

  }

  @Override

  public String toString() {

    return (

      super.toString() +

      "Patient [patientID=" +

      patientID +

      ", diagnosis=" +

      diagnosis +

      ", prescription=" +

      prescription +

      ", doctorID=" +

      doctorID +

      ", appointmentDate=" +

      appointmentDate +

      ", appointmentTime=" +

      appointmentTime +

      ", appointmentStatus=" +

      appointmentStatus +

      ", appointmentID=" +

      appointmentID +

      ", appointmentFee=" +

      appointmentFee +

      "]"

    );

  }

  public static void menu() {

    System.out.println("Welcome to the Patient Panel");

    System.out.println("1. Login");

    System.out.println("2. Register");

    System.out.println("3. Exit");

    System.out.println("Enter your choice: ");

    Scanner sc = new Scanner(System.in);

    int choice = sc.nextInt();

    switch (choice) {

      case 1:

        login();

        break;

      case 2:

        register();

        break;

      case 3:

        System.exit(0);

        break;

      default:

        System.out.println("Invalid choice");

        break;

    }

  }

  static void register() {

    System.out.println("Enter your name: ");

    Scanner sc = new Scanner(System.in);

    String name = sc.nextLine();

    System.out.println("Enter your email: ");

    String email = sc.nextLine();

    System.out.println("Enter your ID: ");

    String patientID = sc.nextLine();

    System.out.println("Enter your password: ");

    String password = sc.nextLine();

    System.out.println("Enter your phone: ");

    String phone = sc.nextLine();

    System.out.println("Enter your address: ");

    String address = sc.nextLine();

    Patient patient = new Patient(

      name,

      email,

      password,

      phone,

      address,

      patientID,

      "",

      "",

      0,

      "",

      "",

      "",

      "",

      ""

    );

    try (FileWriter fw = new FileWriter("patient.txt", true)) {

      fw.write(patient.toString() + "\n");

      fw.close();

      System.out.println("Registration successful");

    } catch (IOException e) {

      e.printStackTrace();

    }

  }

  public static void login() {

    System.out.println("Enter your ID: ");

    Scanner sc = new Scanner(System.in);

    String patientID = sc.nextLine();

    System.out.println("Enter your password: ");

    String password = sc.nextLine();

    try (Scanner scanner = new Scanner(new File("patient.txt"))) {

      boolean loginSuccessful = false;

      while (scanner.hasNextLine()) {

        String line = scanner.nextLine();

        if (

          line.contains("patientID=" + patientID) && line.contains(password)

        ) {

          loginSuccessful = true;

          break;

        }

      }

      if (loginSuccessful) {

        System.out.println("Login successful");

      } else {

        System.out.println("Invalid ID or password");

      }

    } catch (IOException e) {

      e.printStackTrace();

    }

  }

}

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class Admin extends User {

  private String adminID;

  private String adminPassword;

  public Admin(

      String name,

      String email,

      String password,

      String phone,

      String address,

      String adminID,

      String adminPassword) {

    super(name, email, password, phone, address);

    this.adminID = adminID;

    this.adminPassword = adminPassword;

  }

  //

  public String getAdminID() {

    return adminID;

  }

  public void setAdminID(String adminID) {

    this.adminID = adminID;

  }

  public String getAdminPassword() {

    return adminPassword;

  }

  public void setAdminPassword(String adminPassword) {

    this.adminPassword = adminPassword;

  }

  @Override

  public String toString() {

    return "super().toString() " + "Admin [adminID=" + adminID + ", adminPassword=" + adminPassword + "]";

  }

  // Admin Menu

  public static void menu() {

    System.out.println("Welcome to the Admin Panel");

    System.out.println("1. Login");

    System.out.println("2. Exit");

    System.out.println("Enter your choice: ");

    Scanner sc = new Scanner(System.in);

    int choice = sc.nextInt();

    switch (choice) {

      case 1:

        register();

        break;

      case 2:

        System.exit(0);

        break;

      default:

        break;

    }

  }

  public static void register() {

    Scanner sc = new Scanner(System.in);

    // Register method should take all the data and save it into file with one space

    // after each entry. And line break after whole data of one admin

    System.out.println("Enter your Name: ");

    String name = sc.nextLine();

    System.out.println("Enter your Email: ");

    String email = sc.nextLine();

    System.out.println("Enter your Password: ");

    String password = sc.nextLine();

    System.out.println("Enter your Phone: ");

    String phone = sc.nextLine();

    System.out.println("Enter your Address: ");

    String address = sc.nextLine();

    System.out.println("Enter your Admin ID: ");

    String adminID = sc.nextLine();

    System.out.println("Enter your Admin Password: ");

    String adminPassword = sc.nextLine();

    Admin admin = new Admin(name, email, password, phone, address, adminID, adminPassword);

    // Save admin object into file

    System.out.println("Admin Registered Successfully");

    try {

      FileWriter fw = new FileWriter("admin.txt", true);

      fw.write(admin.toString() + "\n");

      fw.close();

    } catch (IOException e) {

      e.printStackTrace();

    }

  }

}