```
package version2;
import java.util.ArrayList;
import java.util.Scanner;
class Doctor {
  private String doctorId;
  private String name;
  private String specialization;
  private ArrayList<String> availableSlots;
  private double consultationFee;
  private ArrayList<Appointment> scheduledAppointments;
  public Doctor(String doctorId, String name, String specialization, double consultationFee) {
     this.doctorId = doctorId;
     this.name = name;
     this.specialization = specialization;
     this.availableSlots = new ArrayList<>();
     this.consultationFee = consultationFee;
     this.scheduledAppointments = new ArrayList<>();
  }
  public void addAvailableSlot(String timeSlot) {
     this.availableSlots.add(timeSlot);
  }
  public boolean isAvailable(String timeSlot) {
     for (Appointment appointment : scheduledAppointments) {
        if (appointment.getAppointmentTime().equals(timeSlot)) {
           return false;
        }
     }
     return availableSlots.contains(timeSlot);
  }
  public void addAppointment(Appointment appointment) {
     scheduledAppointments.add(appointment);
  }
  public double getConsultationFee() {
     return consultationFee;
  }
  public String getDoctorId() {
     return doctorId;
  }
  public String getName() {
     return name;
  }
  public String getSpecialization() {
     return specialization;
  }
  public ArrayList<String> getAvailableSlots() {
     return availableSlots;
  }
  @Override
```

```
    public String toString() {
        return "Dr. " + name + " (" + specialization + ") - Fee: \u20B9" + consultationFee;
    }
}
class Patient {
    private String patientId;
    private String name;
    private int age;
    private String medicalHistory;
    public ArrayList<String> previousAppointments;
    public Patient(String patientId, String name, int age, String medicalHistory) {
        this.patientId = patientId;
        this.name = name;
        this.age = age;
        this.medicalHistory = medicalHistory;
        this.previousAppointments = new ArrayList<>();
    }
    public String getPatientId() {
        return patientId;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public String getMedicalHistory() {
        return medicalHistory;
    }
    public void addAppointment(String appointmentDetails) {
        this.previousAppointments.add(appointmentDetails);
    }
    @Override
    public String toString() {
        return name + " (Age: " + age + ")";
    }
}
class Appointment {
    private Patient patient;
    private Doctor doctor;
    private String appointmentDate;
    private String appointmentTime;
    private String status;
    private double billAmount;
    public Appointment(Patient patient, Doctor doctor, String appointmentDate, String appointmentTime) {
        this.patient = patient;
        this.doctor = doctor;
        this.appointmentDate = appointmentDate;
        this.appointmentTime = appointmentTime;
```

```
        this.status = "Scheduled";
        this.billAmount = doctor.getConsultationFee();
    }
    public String getAppointmentTime() {
        return appointmentTime;
    }
    public void cancelAppointment() {
        this.status = "Cancelled";
    }
    public void rescheduleAppointment(String newDate, String newTime) {
        this.appointmentDate = newDate;
        this.appointmentTime = newTime;
        this.status = "Rescheduled";
    }
    @Override
    public String toString() {
        return patient.getName() + " has an appointment with Dr. " + doctor.getName() + " on " +
appointmentDate + " at " + appointmentTime + ". Status: " + status + " | Bill: \u20B9" + billAmount;
    }
}
public class HMS {
    static Scanner scanner = new Scanner(System.in);
    public static void menu(ArrayList<Doctor> doctorList, ArrayList<Patient> patientList) {
        boolean running = true;
        while (running) {
            System.out.println("\n----- Hospital Management System -----");
            System.out.println("1. Add New Patient");
            System.out.println("2. Book Appointment");
            System.out.println("3. Add New Doctor");
            System.out.println("4. View All Doctors");
            System.out.println("5. View All Patients");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            try {
                int choice = Integer.parseInt(scanner.nextLine());
                switch (choice) {
                    case 1:
                        addNewPatient(patientList);
                        break;
                    case 2:
                        bookAppointment(doctorList, patientList);
                        break;
                    case 3:
                        addNewDoctor(doctorList);
                        break;
                    case 4:
                        viewDoctors(doctorList);
                        break;
                    case 5:
```

```
                viewPatients(patientList);
                break;
            case 6:
                System.out.println("Exiting HMS. Thank you!");
                running = false;
                break;
            default:
                System.out.println("Invalid choice.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Please enter a valid number.");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
public static void addNewPatient(ArrayList<Patient> patientList) {
    try {
        System.out.print("Enter Patient ID: ");
        String pid = scanner.nextLine();
        boolean exists = patientList.stream().anyMatch(p -> p.getPatientId().equals(pid));
        if (exists) {
            System.out.println("Patient already exists.");
            return;
        }
        System.out.print("Enter Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Age: ");
        int age = Integer.parseInt(scanner.nextLine());
        System.out.print("Enter Medical History: ");
        String history = scanner.nextLine();
        Patient patient = new Patient(pid, name, age, history);
        patientList.add(patient);
        System.out.println("Patient added successfully.");
    } catch (NumberFormatException e) {
        System.out.println("Invalid age input.");
    } catch (Exception e) {
        System.out.println("Error adding patient: " + e.getMessage());
    }
}
public static void bookAppointment(ArrayList<Doctor> doctorList, ArrayList<Patient> patientList) {
    try {
        System.out.print("Enter Patient ID: ");
        String pid = scanner.nextLine();
        Patient patient = null;
        for (Patient p : patientList) {
            if (p.getPatientId().equals(pid)) {
                patient = p;
                break;
```

```
            }
        }
        if (patient == null) {
            System.out.println("Patient not found. Please add the patient first.");
            return;
        }
        System.out.print("Enter Appointment Date: ");
        String date = scanner.nextLine();
        Doctor selectedDoctor = suggestDoctor(patient.getMedicalHistory(), doctorList);
        if (selectedDoctor == null) return;
        System.out.println("Available Slots: " + selectedDoctor.getAvailableSlots());
        System.out.print("Choose Slot: ");
        String timeSlot = scanner.nextLine();
        if (!selectedDoctor.isAvailable(timeSlot)) {
            System.out.println("Slot not available.");
            return;
        }
        Appointment appointment = new Appointment(patient, selectedDoctor, date, timeSlot);
        selectedDoctor.addAppointment(appointment);
        patient.addAppointment(appointment.toString());
        System.out.println("Appointment Created: " + appointment);
    } catch (Exception e) {
        System.out.println("Error booking appointment: " + e.getMessage());
    }
}
public static Doctor suggestDoctor(String history, ArrayList<Doctor> doctorList) {
    if (history.toLowerCase().contains("heart") || history.toLowerCase().contains("cardio")) {
        for (Doctor doc : doctorList) {
            if (doc.getSpecialization().equalsIgnoreCase("Cardiologist")) return doc;
        }
    } else if (history.toLowerCase().contains("brain") || history.toLowerCase().contains("neuro")) {
        for (Doctor doc : doctorList) {
            if (doc.getSpecialization().equalsIgnoreCase("Neurologist")) return doc;
        }
    }
    System.out.println("Doctors Available:");
    for (int i = 0; i < doctorList.size(); i++) {
        System.out.println((i + 1) + ". " + doctorList.get(i));
    }
    System.out.print("Choose doctor (enter number): ");
    int choice = Integer.parseInt(scanner.nextLine());
    if (choice < 1 || choice > doctorList.size()) {
        System.out.println("Invalid doctor.");
        return null;
    }
    return doctorList.get(choice - 1);
}
public static void addNewDoctor(ArrayList<Doctor> doctorList) {
    try {
```

```
            System.out.print("Enter Doctor ID: ");
            String id = scanner.nextLine();
            System.out.print("Enter Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Specialization: ");
            String spec = scanner.nextLine();
            System.out.print("Enter Fee: ");
            double fee = Double.parseDouble(scanner.nextLine());
            Doctor doc = new Doctor(id, name, spec, fee);
            System.out.print("No. of slots: ");
            int n = Integer.parseInt(scanner.nextLine());
            for (int i = 0; i < n; i++) {
                System.out.print("Enter Slot " + (i + 1) + ": ");
                doc.addAvailableSlot(scanner.nextLine());
            }
            doctorList.add(doc);
            System.out.println("Doctor added.");
        } catch (NumberFormatException e) {
            System.out.println("Please enter valid numbers for fee and slots.");
        } catch (Exception e) {
            System.out.println("Error adding doctor: " + e.getMessage());
        }
    }
    public static void viewDoctors(ArrayList<Doctor> doctorList) {
        System.out.println("---- Doctors ----");
        for (Doctor d : doctorList) {
            System.out.println("ID: " + d.getDoctorId() + ", " + d);
        }
    }
    public static void viewPatients(ArrayList<Patient> patientList) {
        System.out.println("---- Patients ----");
        for (Patient p : patientList) {
            System.out.println(p);
            for (String app : p.previousAppointments) {
                System.out.println("  -> " + app);
            }
        }
    }
    public static void main(String[] args) {
        ArrayList<Doctor> doctorList = new ArrayList<>();
        ArrayList<Patient> patientList = new ArrayList<>();
        // Add default doctors
        Doctor doc1 = new Doctor("D001", "John Smith", "Cardiologist", 1500);
        doc1.addAvailableSlot("10:00 AM");
        doctorList.add(doc1);
        Doctor doc2 = new Doctor("D002", "Emily Clark", "Neurologist", 1200);
        doc2.addAvailableSlot("09:00 AM");
        doctorList.add(doc2);
        menu(doctorList, patientList);
```

```
  }
}
```

```
----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 1
Enter Patient ID: 1
Enter Name: P.M.Mathujaa
Enter Age: 19
Enter Medical History: Fever
Patient added successfully.

----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 1
Enter Patient ID: 1
Patient already exists.
```

```
----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 1
Enter Patient ID: 2
Enter Name: Majaa
Enter Age: 20
Enter Medical History: nil
Patient added successfully.

----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 1
Enter Patient ID: 3
Enter Name: Mathu
Enter Age: 21
Enter Medical History: heart pain
Patient added successfully.
```

```
----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 2
Enter Patient ID: 3
Enter Appointment Date: 2025-05-10
Available Slots: [10:00 AM]
Choose Slot: 10:00 AM
Appointment Created: Mathu has an appointment with Dr. John Smith on 2025-05-10 at 10:00 AM. Status: Scheduled | Bill: ₹1500.0

----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 2
Enter Patient ID: 2
Enter Appointment Date: 2025-05-15
Doctors Available:
1. Dr. John Smith (Cardiologist) - Fee: ₹1500.0
2. Dr. Emily Clark (Neurologist) - Fee: ₹1200.0
Choose doctor (enter number): 2
Available Slots: [09:00 AM]
Choose Slot: 09:00 AM
Appointment Created: Majaa has an appointment with Dr. Emily Clark on 2025-05-15 at 09:00 AM. Status: Scheduled | Bill: ₹1200.0
```

```
----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 3
Enter Doctor ID: 3
Enter Name: david
Enter Specialization: dentist
Enter Fee: 1000.0
No. of slots: 2
Enter Slot 1: 11:00 AM
Enter Slot 2: 7:00 AM
Doctor added.

----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 4
---- Doctors ----
ID: D001, Dr. John Smith (Cardiologist) - Fee: ₹1500.0
ID: D002, Dr. Emily Clark (Neurologist) - Fee: ₹1200.0
ID: 3, Dr. david (dentist) - Fee: ₹1000.0
```

```
----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 5
---- Patients ----
P.M.Mathujaa (Age: 19)
Majaa (Age: 20)
  -> Majaa has an appointment with Dr. Emily Clark on 2025-05-15 at 09:00 AM. Status: Scheduled | Bill: ₹1200.0
Mathu (Age: 21)
  -> Mathu has an appointment with Dr. John Smith on 2025-05-10 at 10:00 AM. Status: Scheduled | Bill: ₹1500.0

----- Hospital Management System -----
1. Add New Patient
2. Book Appointment
3. Add New Doctor
4. View All Doctors
5. View All Patients
6. Exit
Enter your choice: 6
Exiting HMS. Thank you!
```