```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Ticket {

    String Movie;
    int seatNumber;
    String CustomerName;

    public Ticket(String Movie, int seatNumber, String CustomerName) {
        this.Movie = Movie;
        this.seatNumber = seatNumber;
        this.CustomerName = CustomerName;
    }

    @Override
    public String toString() {
        return Movie + "," + seatNumber + "," + CustomerName;
    }
}

class MovieTicketCRUD {

    static File ticketFile = new File("tickets.txt");
    static String[] Movie = {"Chello Divas", "Love Ni Bhavai", "3 Ekka", "Su Thayu", "GujjuBhai Most Wanted"};
    static int TotalSeat = 10;

    public static void WriteToFile(List<String> records) {
        try (FileWriter FIWR = new FileWriter(ticketFile, false)) {
            for (String r : records) {
                FIWR.write(r + "\n");
            }
        } catch (IOException IOErr) {
            System.out.println("File Writing ERROR: " + ticketFile.getName());
            System.out.println("File Writing ERROR: " + IOErr);
        }
    }

    public static List<String> readFile() {
        List<String> lines = new ArrayList<>();
        if (!ticketFile.exists()) {
            return lines;
        }
        try (Scanner scan = new Scanner(ticketFile)) {
            while (scan.hasNextLine()) {
                lines.add(scan.nextLine());
            }
        } catch (IOException IOErr) {
```

```
            System.out.println("Error Reading File");
        }
        return lines;
    }

    public static void bookTicket(Ticket ticket) {
        List<String> records = readFile();
        for (String lines : records) {
            String[] data = lines.split(",");
            if (data[0].equalsIgnoreCase(ticket.Movie) && Integer.parseInt(data[1]) == ticket.seatNumber) {
                System.out.println("Ticket Already Booked");
            }
        }

        records.add(ticket.toString());
        WriteToFile(records);
        System.out.println("Ticket Booked For: " + ticket.CustomerName + "\nMovie: " + ticket.Movie + "\nSeat
Number: " + ticket.seatNumber);
    }

    public static void viewBooking() {
        List<String> records = readFile();
        if (records.isEmpty()) {
            System.out.println("No Booking Found");
            return;
        }
        System.out.println("\n------ All Bookings -------");
        for (String line : records) {
            String d[] = line.split(",");
            System.out.println("Movie: " + d[0] + "   |   Seat: " + d[1] + "   |   Customer: " + d[2]);
        }
    }

    public static void UpdateBooking(String customerName, int newSeat) {
        List<String> records = readFile();
        boolean updated = false;
        for (int i = 0; i < records.size(); i++) {
            String d[] = records.get(i).split(",");
            if (d[2].equalsIgnoreCase(customerName)) {
                for (String r : records) {
                    String dd[] = r.split(",");
                    if (dd[0].equalsIgnoreCase(d[0]) && Integer.parseInt(dd[1]) == newSeat) {
                        System.err.println("Seat Already Booked");
                        return;
                    }
                }
                records.set(i, d[0] + "," + newSeat + "," + d[2]);
                updated = true;
                break;
            }
        }

        if (updated) {
```

```
            WriteToFile(records);
            System.out.println("Booking Updated Successfully");
        } else {
            System.out.println("Customer Not Found");
        }
    }

    public static void cancelBooking(String customerName) {
        List<String> records = readFile();
        boolean deleted = records.removeIf(line -> line.split(",")[2].equalsIgnoreCase(customerName));
        if (deleted) {
            WriteToFile(records);
            System.out.println("Booking Cancel Successfully");
        } else {
            System.out.println("Booking Not found");
        }
    }

    public static void checkAvailableSeat(String movieName) {
        boolean seats[] = new boolean[TotalSeat + 1];
        List<String> records = readFile();
        for (String line : records) {
            String d[] = line.split(",");
            if (d[0].equalsIgnoreCase(movieName)) {
                int seat = Integer.parseInt(d[1]);
                seats[seat] = true;
            }
        }

        System.out.println("Available Seats for Movie: " + movieName);
        for (int i = 0; i < TotalSeat; i++) {
            if (!seats[i]) {
                System.out.println(i + " ");
            }
        }
        System.out.println();
    }
}

public class MovieTicket extends MovieTicketCRUD {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int Choice;
        do {
            System.out.println("\n=============== Movie Ticket Booking System ==================");
            System.out.println("1. View Movies");
            System.out.println("2. Check Available Seats");
            System.out.println("3. Book Ticket");
            System.out.println("4. View All Bookings");
            System.out.println("5. Update Booking (Change Seat)");
            System.out.println("6. Cancel Booking");
            System.out.println("7. Exit");
```

```
System.out.println("Enter Choice:");

Choice = scan.nextInt();
scan.nextLine();

switch (Choice) {
   case 1 -> {
      for (String m : Movie) {
         System.out.println(m);
      }
   }

   case 2 -> {
      System.out.println("Enter Movie Name: ");
      String movieName = scan.next();
      checkAvailableSeat(movieName);
   }

   case 3 -> {
      System.out.println("Enter Movie Name: ");
      String bookMovie = scan.nextLine();
      checkAvailableSeat(bookMovie);
      System.out.println("Enter Seat Number (1-10): ");
      int seat = scan.nextInt();
      scan.nextLine();

      System.out.println("Enter Customer Name");
      String custName = scan.nextLine();
      bookTicket(new Ticket(bookMovie, seat, custName));
   }

   case 4 -> viewBooking();

   case 5 -> {
      System.out.println("Enter Customer Name:");
      String updateName = scan.nextLine();
      System.out.println("Enter New Seat Number");
      int newSeat = scan.nextInt();
      UpdateBooking(updateName, newSeat);
      scan.nextLine();
   }

   case 6 -> {
      System.out.println("Enter Customer Name to cancel Booking: ");
      String DeleteName = scan.nextLine();
      cancelBooking(DeleteName);
   }

   case 7 -> System.out.println("Exting.....");
   default -> {
      System.out.println("Invalid Choice");
      scan.close();
   }
```

```
            }
        } while (Choice != 7);
    }
}
```

Hospital Management System

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Patient {

   String P_Id, P_Name, P_Diseases;
   int P_Age;

   // Constructor: Patient
   public Patient(String P_Id, String P_Name, String P_Diseases, int P_Age) {
      this.P_Id = P_Id;
      this.P_Name = P_Name;
      this.P_Age = P_Age;
      this.P_Diseases = P_Diseases;
   }

   public String PatientData() {
      return P_Id + "," + P_Name + "," + P_Age + "," + P_Diseases;
   }
}

class Doctor {

   String D_Id, D_Name, D_Specialization;

   // Constructor: Doctor
   public Doctor(String D_Id, String D_Name, String D_Specialization) {
      this.D_Id = D_Id;
      this.D_Name = D_Name;
      this.D_Specialization = D_Specialization;
   }

   public String DoctorData() {
      return D_Id + "," + D_Name + "," + D_Specialization;
   }
}

class Appointment {

   String P_Id, D_Id, A_Date;

   // Constructor: Appointment
   public Appointment(String P_Id, String D_Id, String A_Date) {
      this.P_Id = P_Id;
      this.D_Id = D_Id;
```

```
        this.A_Date = A_Date;
    }

    public String AppointmentData() {
        return P_Id + "," + D_Id + "," + A_Date;
    }
}

public class HospitalManagementSystem {

    static File PatientFile = new File("Patient.txt");
    static File DoctorFile = new File("Doctor.txt");
    static File AppointmentFile = new File("Appointment.txt");

    public static void WriteToFile(File file, String Data, boolean append) {
        try {
            try (FileWriter FIWR = new FileWriter(file, append)) {
                FIWR.write(Data + "\n");
            }
        } catch (IOException IOErr) {
            System.out.println("File Writing ERROR: " + file.getName());
        }
    }

    public static List<String> readFile(File file) {
        List<String> lines = new ArrayList<>();
        if (!file.exists()) {
            return lines;
        }

        try {
            try (Scanner scan = new Scanner(file)) {
                while (scan.hasNextLine()) {
                    lines.add(scan.nextLine());
                }
            }
        } catch (IOException IOErr) {
            System.out.println("File Reading ERROR: " + file.getName());
        }

        return lines;
    }

    public static void AddPatient(Patient patient) {
        WriteToFile(PatientFile, patient.PatientData(), true);
        System.out.println("Patient Added Successfully");
    }

    public static void AddDoctor(Doctor doctor) {
        WriteToFile(DoctorFile, doctor.DoctorData(), true);
        System.out.println("Doctor Added Successfully");
    }
```

```
public static void AddAppointment(Appointment appointment) {
    boolean patientExist = false, doctorExist = false;

    for (String P : readFile(PatientFile)) {
        if (P.split(",")[0].equals(appointment.P_Id)) {
            patientExist = true;
        }
    }

    for (String D : readFile(DoctorFile)) {
        if (D.split(",")[0].equals(appointment.D_Id)) {
            doctorExist = true;
        }
    }

    if (!patientExist) {
        System.out.println("Invalid Patient ID or Patient not Found");
        return;
    }

    if (!doctorExist) {
        System.out.println("Invalid Doctor ID or Doctor not Found");
        return;
    }

    WriteToFile(AppointmentFile, appointment.AppointmentData(), true);
    System.out.println("Appointment Added Successfully");
}

public static void viewPatient() {
    List<String> P_Records = readFile(PatientFile);

    if (P_Records.isEmpty()) {
        System.out.println("No Patient Records Found.");
        return;
    }

    System.out.println("\n------------------------------ Patient Records ------------------------------");

    // Define fixed widths for columns
    int colWidth1 = 15; // ID
    int colWidth2 = 25; // Name
    int colWidth3 = 10; // Age
    int colWidth4 = 30; // Diseases

    // Print table header
    System.out.printf("%-" + colWidth1 + "s%-" + colWidth2 + "s%-" + colWidth3 + "s%-" + colWidth4 + "s%n",
            "ID", "Name", "Age", "Diseases");

    // Java 8-compatible replacement for "repeat()"
    int totalWidth = colWidth1 + colWidth2 + colWidth3 + colWidth4;
    System.out.println(new String(new char[totalWidth]).replace('\0', '='));
```

```
        // Print each patient record
        for (String line : P_Records) {
            String[] data = line.split(",");

            // Defensive programming for missing data
            String id = data.length > 0 ? data[0] : "";
            String name = data.length > 1 ? data[1] : "";
            String age = data.length > 2 ? data[2] : "";
            String diseases = data.length > 3 ? data[3] : "";

            System.out.printf("%-" + colWidth1 + "s%-" + colWidth2 + "s%-" + colWidth3 + "s%-" + colWidth4 +
"s%n",
                    id, name, age, diseases);
        }
    }

    public static void viewDoctor() {
        List<String> D_Records = readFile(DoctorFile);

        if (D_Records == null || D_Records.isEmpty()) {
            System.out.println("No Doctor Records Found.");
            return;
        }

        System.out.println("\n---------------------------- Doctor Records ----------------------------");

        int colWidth1 = 15; // ID
        int colWidth2 = 25; // Name
        int colWidth3 = 25; // Specialization

        System.out.printf("%-" + colWidth1 + "s%-" + colWidth2 + "s%-" + colWidth3 + "s%n",
                "ID", "Name", "Specialization");

        // Java 8-compatible string repeat
        System.out.println(new String(new char[colWidth1 + colWidth2 + colWidth3]).replace('\0', '='));

        for (String line : D_Records) {
            if (line == null || line.trim().isEmpty()) {
                continue;
            }

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

            String id = data.length > 0 ? data[0].trim() : "";
            String name = data.length > 1 ? data[1].trim() : "";
            String specialization = data.length > 2 ? data[2].trim() : "";

            System.out.printf("%-" + colWidth1 + "s%-" + colWidth2 + "s%-" + colWidth3 + "s%n",
                    id, name, specialization);
        }
    }

    public static void viewAppointment() {
```

```
    List<String> A_Records = readFile(AppointmentFile);

    if (A_Records == null || A_Records.isEmpty()) {
        System.out.println("No Appointment Records Found.");
        return;
    }

    System.out.println("\n---------------------------- Appointment Records ----------------------------");

    // Define fixed widths for columns
    int colWidth1 = 20; // Patient ID
    int colWidth2 = 20; // Doctor ID
    int colWidth3 = 25; // Appointment Date

    // Print table header
    System.out.printf("%-" + colWidth1 + "s%-" + colWidth2 + "s%-" + colWidth3 + "s%n",
            "Patient ID", "Doctor ID", "Appointment Date");

    // Java 8-compatible way to repeat "="
    System.out.println(new String(new char[colWidth1 + colWidth2 + colWidth3]).replace('\0', '='));

    // Print each appointment record
    for (String line : A_Records) {
        if (line == null || line.trim().isEmpty()) {
            continue; // Skip empty or blank lines
        }

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

        // Defensive programming for missing fields
        String patientId = data.length > 0 ? data[0].trim() : "";
        String doctorId = data.length > 1 ? data[1].trim() : "";
        String date = data.length > 2 ? data[2].trim() : "";

        System.out.printf("%-" + colWidth1 + "s%-" + colWidth2 + "s%-" + colWidth3 + "s%n",
                patientId, doctorId, date);
    }
}

public static void main(String[] args) {
    try (Scanner scan = new Scanner(System.in)) {
        int choice;

        do {
            System.out.println("\nHospital Management System\n");

            System.out.println("Select Appropriate Option");
            System.out.println("Press 1: Add Patient");
            System.out.println("Press 2: View Patient");
            System.out.println("Press 3: Add Doctor");
            System.out.println("Press 4: View Doctor");
            System.out.println("Press 5: Add Appointment");
            System.out.println("Press 6: View Appointment");
```

```
    System.out.println("Press 7: Exit");

    System.out.print("Enter Choice: ");
    choice = scan.nextInt();
  } while (choice >= 7);

  switch (choice) {
    // Add New Patient
    case 1:
      System.out.println("Enter Patient ID:");
      String P_id = scan.next();

      System.out.println("Enter Patient Name:");
      String P_name = scan.next();

      System.out.println("Enter Patient Age:");
      int P_age = scan.nextInt();

      System.out.println("Enter Patient Diseases:");
      String P_diseases = scan.next();

      AddPatient(new Patient(P_id, P_name, P_diseases, P_age));

    // view Patient List
    case 2:
      viewPatient();

    // Add New Doctor
    case 3:
      System.out.println("Enter Doctor ID: ");
      String D_id = scan.next();

      System.out.println("Enter Doctor Name: ");
      String D_name = scan.next();
      System.out.println("Enter Doctor Specification: ");
      String D_specification = scan.next();

      AddDoctor(new Doctor(D_id, D_name, D_specification));

    // view Doctor List
    case 4:
      viewDoctor();

    // Add new Appointment
    case 5:
      System.out.println("Enter Patient ID: ");
      String A_P_id = scan.next();

      System.out.println("Enter Doctor ID: ");
      String A_D_id = scan.next();

      System.out.println("Enter Date: ");
      String A_Date = scan.next();
```

```
                AddAppointment(new Appointment(A_P_id, A_D_id, A_Date));

            // view Appointment
            case 6:
                viewAppointment();

            default:
                System.out.println(" ! Incorrect Choice ! ");
        }
      }
    }
}
```