```
import java.io.*;
import java.util.*;
abstract class Person {
  protected String id, name;
  protected int age;
  public Person(String id, String name, int age) {
    this.id = id;
    this.name = name;
    this.age = age;
  }
  public abstract void displayDetails();
}
class Doctor extends Person {
  String specialization;
  public Doctor(String id, String name, int age, String specialization) {
    super(id, name, age);
    this.specialization = specialization;
  }
  @Override
  public void displayDetails() {
    System.out.println("Doctor ID: " + id + ", Name: " + name + ", Age: " + age + ", Specialization: " +
specialization);
  }
}
class Patient extends Person {
  String disease;
  public Patient(String id, String name, int age, String disease) {
    super(id, name, age);
    this.disease = disease;
  }
  @Override
  public void displayDetails() {
    System.out.println("Patient ID: " + id + ", Name: " + name + ", Age: " + age + ", Disease: " + disease);
  }
}
public class HospitalManagementSystem {
  static final String DOCTOR_FILE = "doctors.txt";
  static final String PATIENT_FILE = "patients.txt";
  static Scanner sc = new Scanner(System.in);
  public static void main(String[] args) {
    while (true) {
      System.out.println("\n--- Hospital Management Menu ---");
      System.out.println("1. Add Doctor");
      System.out.println("2. Add Patient");
      System.out.println("3. Search Doctor");
      System.out.println("4. Search Patient");
      System.out.println("5. Update Doctor");
      System.out.println("6. Update Patient");
      System.out.println("7. Delete Doctor");
```

```
        System.out.println("8. Delete Patient");
        System.out.println("9. Exit");
        System.out.print("Choose an option: ");
        int choice = Integer.parseInt(sc.nextLine());
        try {
            switch (choice) {
                case 1 -> addDoctor();
                case 2 -> addPatient();
                case 3 -> searchById(DOCTOR_FILE);
                case 4 -> searchById(PATIENT_FILE);
                case 5 -> updateById(DOCTOR_FILE);
                case 6 -> updateById(PATIENT_FILE);
                case 7 -> deleteById(DOCTOR_FILE);
                case 8 -> deleteById(PATIENT_FILE);
                case 9 -> {
                    System.out.println("Exiting...");
                    return;
                }
                default -> System.out.println("Invalid option. Try again.");
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
static void addDoctor() throws IOException {
    System.out.print("Enter Doctor ID: ");
    String id = sc.nextLine();
    System.out.print("Enter Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Age: ");
    int age = Integer.parseInt(sc.nextLine());
    System.out.print("Enter Specialization: ");
    String spec = sc.nextLine();
    Doctor d = new Doctor(id, name, age, spec);
    writeToFile(DOCTOR_FILE, d.id + "," + d.name + "," + d.age + "," + spec);
    System.out.println("Doctor added.");
}
static void addPatient() throws IOException {
    System.out.print("Enter Patient ID: ");
    String id = sc.nextLine();
    System.out.print("Enter Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Age: ");
    int age = Integer.parseInt(sc.nextLine());
    System.out.print("Enter Disease: ");
    String dis = sc.nextLine();
    Patient p = new Patient(id, name, age, dis);
    writeToFile(PATIENT_FILE, p.id + "," + p.name + "," + p.age + "," + dis);
```

```
    System.out.println("Patient added.");
}
static void writeToFile(String filename, String content) throws IOException {
    BufferedWriter writer = new BufferedWriter(new FileWriter(filename, true));
    writer.write(content + "\n");
    writer.close();
}
static void searchById(String filename) throws IOException {
    System.out.print("Enter ID to search: ");
    String id = sc.nextLine();
    BufferedReader reader = new BufferedReader(new FileReader(filename));
    String line;
    boolean found = false;
    while ((line = reader.readLine()) != null) {
        String[] parts = line.split(",");
        if (parts[0].equals(id)) {
            System.out.println("Record found: " + line);
            found = true;
            break;
        }
    }
    reader.close();
    if (!found) System.out.println("Record not found.");
}
static void updateById(String filename) throws IOException {
    System.out.print("Enter ID to update: ");
    String id = sc.nextLine();
    File inputFile = new File(filename);
    File tempFile = new File("temp.txt");
    BufferedReader reader = new BufferedReader(new FileReader(inputFile));
    BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));
    String line;
    boolean updated = false;
    while ((line = reader.readLine()) != null) {
        String[] parts = line.split(",");
        if (parts[0].equals(id)) {
            System.out.print("Enter new Name: ");
            String name = sc.nextLine();
            System.out.print("Enter new Age: ");
            int age = Integer.parseInt(sc.nextLine());
            System.out.print("Enter new Field (Specialization/Disease): ");
            String field = sc.nextLine();
            writer.write(id + "," + name + "," + age + "," + field + "\n");
            updated = true;
        } else {
            writer.write(line + "\n");
        }
    }
    reader.close();
```

```
        writer.close();
        inputFile.delete();
        tempFile.renameTo(inputFile);
        if (updated) System.out.println("Record updated.");
        else System.out.println("ID not found.");
    }
    static void deleteById(String filename) throws IOException {
        System.out.print("Enter ID to delete: ");
        String id = sc.nextLine();
        File inputFile = new File(filename);
        File tempFile = new File("temp.txt");
        BufferedReader reader = new BufferedReader(new FileReader(inputFile));
        BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));
        String line;
        boolean deleted = false;
        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(",");
            if (parts[0].equals(id)) {
                deleted = true;
                continue;
            }
            writer.write(line + "\n");
        }
        reader.close();
        writer.close();
        inputFile.delete();
        tempFile.renameTo(inputFile);
        if (deleted) System.out.println("Record deleted.");
        else System.out.println("ID not found.");
    }
}
```

```
--- Hospital Management Menu ---
1. Add Doctor
2. Add Patient
3. Search Doctor
4. Search Patient
5. Update Doctor
6. Update Patient
7. Delete Doctor
8. Delete Patient
9. Exit
Choose an option: 1
Enter Doctor ID: 101
Enter Name: nataraj
Enter Age: 24
Enter Specialization: bonw
Doctor added.
```

```
--- Hospital Management Menu ---
1. Add Doctor
2. Add Patient
3. Search Doctor
4. Search Patient
5. Update Doctor
6. Update Patient
7. Delete Doctor
8. Delete Patient
9. Exit
Choose an option: 2
Enter Patient ID: 1001
Enter Name: rahul
Enter Age: 19
Enter Disease: headache
Patient added.
```

```
--- Hospital Management Menu ---
1. Add Doctor
2. Add Patient
3. Search Doctor
4. Search Patient
5. Update Doctor
6. Update Patient
7. Delete Doctor
8. Delete Patient
9. Exit
Choose an option: 5
Enter ID to update: 101
Enter new Name: nataraj
Enter new Age: 25
Enter new Field (Specialization/Disease): dentist
Record updated.
```

```
--- Hospital Management Menu ---
1. Add Doctor
2. Add Patient
3. Search Doctor
4. Search Patient
5. Update Doctor
6. Update Patient
7. Delete Doctor
8. Delete Patient
9. Exit
Choose an option: 8
Enter ID to delete: 1001
Record deleted.
```