

Project : Patient Management System

The role of the student	Id	Student Name
1. Build PatientManagementSystem class and define variables and constructor function 2. Create the addPatient function for Add a new patient	444003238	شيماء لافي الزبيدي
1. Remove Patient with removePatient function 2. Add a new clinic using addClinic(String clinic) function	44410110	ريناد عوض العمري
1. Updating the clinics list with updateClinics(String[] updatedClinics) function 2. Remove Clinic with removeClinic(String clinic) function	444010345	ونام حسن الزهراني
1. Add a new blood type with addBloodType(String bloodType) function 2. Remove blood type with removeBloodType(String bloodType) function	44410353	ريم يوسف الفقيه
1. display records with displayRecords() function 2. building the main function	44410597	شهد علي البارقي

This code is a patient management system that allows the user to perform a range of different operations. These are the program functions available in this code and a simple explanation of each:

1. Add a new patient: Allows the user to add a new patient to the patient list.
2. Remove Patient: Allows the user to remove a patient from the patient list.
3. Add a new clinic: Allows the user to add a new clinic to the clinics list.
4. Updating the clinics list: The user is allowed to update the clinics list by entering a new list of clinics.
5. Remove Clinic: Allows the user to remove a clinic from the clinics list.
6. Add a new blood type: Allows the user to add a new blood type to the list of blood types.

7. Remove blood type: Allows the user to remove a blood type from the blood type list.
8. View records: It displays records of patients, clinics, and registered blood types.
9. End the program: Allows the user to terminate the program.

The program shows the user options for the functions to be performed. Allows the user to choose the action to be performed by entering the desired option number and entering the required part of the function

The program repeats continuously inside the while loop until the option (End the program) is chosen to end the program.

The code:

```
import java.util.Scanner;

public class PatientManagementSystem {

    private int MAX_PATIENTS = 100;
    private int MAX_CLINICS = 50;
    private int MAX_BLOOD_TYPES = 10;
    private String[] patients;
    private String[] clinics;
    private String[] bloodTypes;
    private int patientCount;
    private int clinicCount;
    private int bloodTypeCount;

    public PatientManagementSystem() {
        patients = new String[MAX_PATIENTS];
        clinics = new String[MAX_CLINICS];
        bloodTypes = new String[MAX_BLOOD_TYPES];
        patientCount = 0;
```

```
        clinicCount = 0;
        bloodTypeCount = 0;
    }
    public void addPatient(String patient) {
        if (patientCount < MAX_PATIENTS) {
            patients[patientCount] = patient;
            patientCount++;
            System.out.println("Patient added: " + patient);
        } else {
            System.out.println("Cannot add patient, maximum number of patients
exceeded.");
        }
    }
}
```

```
public void removePatient() {
    if (patientCount > 0) {
        String patient = patients[0];
        for (int i = 0; i < patientCount - 1; i++) {
            patients[i] = patients[i + 1];
        }
        patientCount--;
        System.out.println("Patient removed: " + patient);
    } else {
        System.out.println("No patients in the patient list.");
    }
}
```

```
public void addClinic(String clinic) {
    if (clinicCount < MAX_CLINICS) {
```

```
        clinics[clinicCount] = clinic;
        clinicCount++;
        System.out.println("Clinic added: " + clinic);
    } else {
        System.out.println("Cannot add clinic, maximum number of clinics exceeded.");
    }
}
```

```
public void updateClinics(String[] updatedClinics) {
    clinics = updatedClinics;
    clinicCount = updatedClinics.length;
    System.out.println("Clinic list updated.");
}
```

```
public void removeClinic(String clinic) {
    boolean found = false;
    for (int i = 0; i < clinicCount; i++) {
        if (clinics[i].equals(clinic)) {
            found = true;
            for (int j = i; j < clinicCount - 1; j++) {
                clinics[j] = clinics[j + 1];
            }
            clinicCount--;
            break;
        }
    }

    if (found) {
```

```
        System.out.println("Clinic removed: " + clinic);
    } else {
        System.out.println("Clinic not found in the clinic list.");
    }
}
```

```
public void addBloodType(String bloodType) {
    if (bloodTypeCount < MAX_BLOOD_TYPES) {
        bloodTypes[bloodTypeCount] = bloodType;
        bloodTypeCount++;
        System.out.println("Blood type added: " + bloodType);
    } else {
        System.out.println("Cannot add blood type, maximum number of blood types
exceeded.");
    }
}
```

```
public void removeBloodType(String bloodType) {
    boolean found = false;
    for (int i = 0; i < bloodTypeCount; i++) {
        if (bloodTypes[i].equals(bloodType)) {
            found = true;
            for (int j = i; j < bloodTypeCount - 1; j++) {
                bloodTypes[j] = bloodTypes[j + 1];
            }
            bloodTypeCount--;
            break;
        }
    }
}
```

```
if (found) {  
    System.out.println("Blood type removed: " + bloodType);  
} else {  
    System.out.println("Blood type not found in the blood type list.");  
}  
}
```

```
public void displayRecords() {  
    System.out.println(" Patient Records : ");  
    for (int i = 0; i < patientCount; i++) {  
        System.out.println(patients[i]);  
    }  
    System.out.println("- Clinic Records : ");  
    for (int i = 0; i < clinicCount; i++) {  
        System.out.println(clinics[i]);  
    }  
    System.out.println("| Blood Type Records : ");  
    for (int i = 0; i < bloodTypeCount; i++) {  
        System.out.println(bloodTypes[i]);  
    }  
}
```

```
public static void main(String[] args) {  
    PatientManagementSystem system = new PatientManagementSystem();  
    Scanner ob = new Scanner(System.in);  
    System.out.println("select options");  
    System.out.println("1. add a new patient");  
    System.out.println("2. Remove patient");  
}
```

```
System.out.println("3. Add a new clinic");
System.out.println("4. Update clinic list");
System.out.println("5. Remove clinic");
System.out.println("6. Add a new blood type");
System.out.println("7. Remove blood type");
System.out.println("8. Display records");
System.out.println("9. End the program");
```

```
while (true) {
    System.out.print("Choose action: ");
    int choice =ob. nextInt();
    ob.nextLine();

    switch (choice) {
        case 1:
            System.out.print("Enter patient name: ");
            String patientName =ob. nextLine();
            system.addPatient(patientName);
            break;
        case 2:
            system. removePatient();
            break;
        case 3:
            System.out.print("Enter the clinic name: ");
            String clinicName = ob. nextLine();
            system.addClinic(clinicName);
            break;
        case 4:
```

```
System.out.print("Enter the number of clinics: ");
int clinicCount =ob. nextInt();
ob.nextLine();
String[] updatedClinics = new String[clinicCount];
for (int i = 0; i < clinicCount; i++) {
    System.out.print("Enter the clinic name " + (i + 1) + ": ");
    updatedClinics[i] =ob.nextLine();
}
system.updateClinics(updatedClinics);
break;
case 5:
    System.out.print("Enter the name of the clinic you want to remove: ");
    String clinicToRemove =ob. nextLine();
    system.removeClinic(clinicToRemove);
    break;
case 6:
    System.out.print("Enter the new blood type: ");
    String bloodType =ob. nextLine();
    system.addBloodType(bloodType);
    break;
case 7:
    System.out.print("Enter the type of blood you want to remove: ");
    String bloodTypeToRemove =ob. nextLine();
    system.removeBloodType(bloodTypeToRemove);
    break;
case 8:
    system.displayRecords();
    break;
```



```
        case 9:
            System.out.println("The program has terminated.");
            System. exit(0);
            break;
        default:
            System.out.println("Invalid selection. Please try again.");
    }
}
}
```