<u>Basic Concept:</u>

So, for Hospital management system we decided to make 4 appropriate class. The four classes are Hospital Information that will contain all the information of the hospital, Patients class for patient information, Doctors class for doctor information and Employees class for employee information.

We needed to appropriately design each of the class. We will use ArrayList and File for saving the information. We are going to use the concept of Abstract class, Inheritance, Association, Generic class and Interface.

<u>Classes:</u>

1-Hospital Information:

In this class it contains all the information of a Hospital. It is associated with all the other class. It inherits the abstract class

Information and overrides all of its abstract class. We have used Arraylist for containing the objects of all the other classes.

Code Representation:

```
package project;

import java.util.ArrayList;

/**
 *
 * @author Sifat
 */
abstract class Information
{
abstract void hospitalInformation();
abstract void hospitalDoctors();
abstract void hospitalPatients();
abstract void hospitalEmployees();
}

public class HospitalInformation extends Information {

    private final int NumberOfBeds=500;
    private int NumberOfIndoorPatients;
    private final int AvailableBeds=NumberOfBeds-NumberOfIndoorPatients;
    private final String Adress="91/5....";
    private final int NumberOfAmbulance=5;
    private final String ImergencyNumber="01......";
    private final ArrayList<Patients> PatientsList=new ArrayList<>();
    private final ArrayList<Doctors> DoctorsList=new ArrayList<>();
    private final ArrayList<Employees> EmployeesList=new ArrayList<>();


    public int getNumberofBeds()
    {
    return this.NumberOfBeds;
    }
    public void setNumberOfIndoorPatients(int a)
    {
     this.NumberOfIndoorPatients=a;
    }
    public int getNumberOfIndoorPatients()
    {
    return this.NumberOfIndoorPatients;
    }
    public int getAvailableBeds()
    {
    return this.AvailableBeds;
    }

    public String getAdress()
    {
    return this.Adress;
    }
    public int getNumberOfAmbulance()
    {
    return this.NumberOfAmbulance;
    }
    public String getImergencyNumber()
    {
    return this.ImergencyNumber;
    }




    @Override
    public void hospitalInformation()
    {
        System.out.println ("Adress: "+this.getAdress()+"Imergency Number: 
    "+this.ImergencyNumber+" Number of Ambulance: "+this.NumberOfAmbulance+"Number of 
    Beds: "+this.NumberOfBeds);
    }


    @Override
    public void hospitalDoctors()
    {
        System.out.println("Doctors: "+this.DoctorsList);
    }
    @Override
    public void hospitalPatients()
    {
    System.out.println("Patients: "+this.PatientsList);
    }
    @Override
    public void hospitalEmployees()
    {
    System.out.println("Employees: "+this.EmployeesList);  } }
```

2-Patients:

In this class we have created two child classes. They also use the instance variables of the parent class through super keyword. Both the child class have their own variables and method as well. This class designed to store information of patients.

Code Representation:

```
package project;

/**
 *
 * @author Sifat
 */
public class Patients

{

private int id;
private String Name;
private String Gender;
private int Age;
```

```
Patients (int id,String Name,String Gender,int Age)
{
this.id=id;
this.Name=Name;
this.Gender=Gender;
this.Age=Age;
}

public void setPatientID(int id)
{
this.id=id;
}
public void setPatientName(String Name)
{
this.Name=Name;
}
public void setPatientGender(String Gender)
{
this.Gender=Gender;
}
public void setPatientAge(int Age)
{
this.Age=Age;
}
public int getPatientID()
{
return this.id;
}
```

```
public String getPatientName()
{
return this.Name;
}
public String getPatientGender()
{
return this.Gender;
}
public int getPatientAge()
{
return this.Age;
}


@Override
public String toString ()
{
return id+Name+Gender+Age;
}

public void bill()
{}
}
class OutdoorPatient extends Patients
{
private final int visitingFee=500;

OutdoorPatient(int id,String Name,String Gender,int Age)
{
super(id,Name,Gender,Age);
```

```
}




public void Display ()

{

System.out.println ("Patirnt Name: "+this.getPatientName()+"\nPatient id: "+this.getPatientID()+"\nPatient Age: "+this.getPatientAge()+"\nPatient Gender"+this.getPatientGender()+"\nVisiting Fee: "+visitingFee);

}

@Override

public void bill ()


{

System.out.println("Bill: "+ visitingFee );

}

}

class IndoorPatient extends Patients


{

   private int bedFee,medicineFee,labTestFee,roomnumber;




   IndoorPatient (int id, String Name,String Gender,int Age, int bedFee,int medicineFee,int labTestFee,int roomnumber)

   {

   super(id,Name,Gender,Age);

   this.bedFee=bedFee;

   this.medicineFee=medicineFee;
   this.labTestFee=labTestFee;
   this.roomnumber=roomnumber;
   }

   public void setBedFee(int a)
   {
   this.bedFee=a;
   }
   public void setMedicineFee(int b)
   {
   this.medicineFee=b;
   }
   public void setLabTestFee(int c)
   {
   this.labTestFee=c;
   }
   public void setRoomnumber(int d)
   {
   this.roomnumber=d;
   }
   public int getBedFee()
   {
   return this.bedFee;
   }
   public int getMedicineFee()
   {
   return this.medicineFee;
   }
   public int getLabTestFee()
   {
    return this.labTestFee;
    }
    public int getRoomnumber()
    {
    return this.roomnumber;
    }



    public void Display  ()
    {
    System.out.println("Patirnt Name: "+this.getPatientName()+"\n Patient id: "+
this.getPatientID()+"\nPatient Age:"+this.getPatientAge()+"\nPatient
Gender:"+this.getPatientGender()+"\nRoom Number"+roomnumber+"\nBed Fee:
"+bedFee+"\nMedicine Fee: "+medicineFee+" Lab Test Fee: "+labTestFee);


    }


    @Override
    public void bill ()
    {
        System.out.println("Bill: "+ (bedFee+medicineFee+labTestFee));
    }



}
```

3-Doctors:

We have used a generic class for this. Few of the instance variables are generic type. This class is designed to contain all the information of doctors.

Code Representation:

```
package project;

/**
 *
 * @author Sifat
 * @param <T>
 */
public class Doctors <T>{




private String doctorName;
private T  doctorId;
private T  phoneNumber;
private String address ;
private String department;
private String  doctorSpecialization;
private T Salary;
```

```
Doctors(){}
  Doctors( T a, String b,T c,String d, String e,String f,T g)
  {
  this.doctorName=b;
  this.doctorId=a;
  this.phoneNumber=c;
  this.address=d;
  this.department=e;
  this.doctorSpecialization=f;
  this.Salary=g;
  }
   public T getdoctorId(){
       return doctorId;
   }
   public void setdoctorId(T doctorId){
       this.doctorId = doctorId;
   }

   public String getdoctorName(){
       return doctorName;
   }
   public void setdoctorName(String doctorName){
       this.doctorName = doctorName;
   }

   public T getphoneNumber(){
       return phoneNumber;
   }
   public void setphoneNumber(T phoneNumber){
       this.phoneNumber = phoneNumber;
```

```
}
public String getaddress(){
    return address;
}
public void setaddress(String address){
    this.address = address;
}
public String getdepartment (){
    return department;
}
public void setdepartment(String department){
    this.department = department;
}
public String getdoctorSpecialization (){
    return doctorSpecialization;
}
public void setdoctorSpecialization(String doctorSpecialization){
    this.doctorSpecialization = doctorSpecialization;
}
public void setDoctorSalary(T a)
{
this.Salary=a;
}
public T getDoctorSalary()
{
return this.Salary;
}

public void display()
{
    System.out.println("Doctor Name: "+doctorName+"\nDoctor ID: "+doctorId+"\nPhonenumber: "+phoneNumber+"\nAdress: "+address+"\nDepartment: "+department+"\nSpecialization"+doctorSpecialization+"\nSalary"+Salary);
    }
}
```

4-Employees:

This class has implemented the comparable interface. Which is a default interface in java. It has some built in methods which can be used to compare between objects. This class contains the employee information.

Code Representation:

```
package project;

/**
 *
 * @author Sifat
 */
public class Employees implements Comparable

{
 private String  employeeName;
private int  employeeId,Salary;

private String  phoneNumber;

private String address ;

private String shift;

private String  designation;



    Employees(){}

    Employees( int employeeId , String employeeName ,String phoneNumber,String address,
String shift, String designation,int Salary )

    {

        this.employeeId=employeeId;

        this.employeeName=employeeName;

        this.shift=shift;

        this.designation=designation;

        this.phoneNumber=phoneNumber;

        this.address=address;

        this.Salary=Salary;



    }



    public int getemployeeId(){

        return employeeId;

    }

    public void setemployeeId(int employeeId){

        this.employeeId = employeeId;

    }
```

```
public String getemployeeName(){
    return employeeName;
}
public void setemployeeName(String employeeName){
    this.employeeName = employeeName;
}

public String getphoneNumber(){
    return phoneNumber;
}
public void setphoneNumber(String phoneNumber){
    this.phoneNumber = phoneNumber;
}
public String getaddress(){
    return address;
}
public void setaddress(String address){
    this.address = address;
}
public String getshift (){
    return shift;
}
public void setshift(String shift){
    this.shift = shift;
}
public String getdesignation (){
    return designation;
}
public void setdesignation(String designation){
    this.designation = designation;
    }


    public int getSalary(){
        return Salary;
    }
    public void setSalary(int Salary){
        this.Salary= Salary;
    }


    public void Display(){
        System.out.println( "Employee ID: "+employeeId+"\nEmployee Name: 
"+employeeName+"\nWorking Shift: "+shift+"\nDesignation: "+designation+"\nPhone 
Number: "+phoneNumber+"\nAdress:  "+address+"\nSalary"+Salary);


 }


    @Override
    public int compareTo(Object t) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of 
generated methods, choose Tools | Templates.
    }
}
```

5-Main Class:

In this class we have created a main menu where user have the options of choosing one of the five auctions. 1.add 2.delete 3.Print 4.Search 5.Exit.

By add option user can add information of a patient or a doctor or employee. The information is also stored in the "Hospital Management System" text file.

By delete option user can delete information of a patient or a doctor or an employee.

By print option user can print the basic information of the hospital. User can also print the list of Patients, Doctors and Employees.

By search option user can search for information of a patient or a doctor or an employee using their id.

Code Representation:

```
package project;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Scanner;

/**
 *
 * @author Sifat
 */
public class Project {


    public static void main(String[] args) throws FileNotFoundException {

        Scanner i = new Scanner(System.in);
         File file=new File("Hospital Management.txt");
         PrintWriter write = new PrintWriter(file);
         ArrayList<Patients> PatientsList=new ArrayList<>();
         ArrayList<Doctors> DoctorsList=new ArrayList<>();
         ArrayList<Employees> EmployeesList=new ArrayList<>();

         x:
while (true) {
System.out.println("1.Add \n 2.Delete \n 3.Print \n 4.Search \n 5.Exit");
```

```
String select1 = i.next();
switch (select1.charAt(0)) {
case '1': {
while (true){
System.out.println("a. Add a Patient");
System.out.println("b. Add a Doctor");
System.out.println("c. Add an Emplyoee");
System.out.println("d. Go To Main Menu");

String select2 = i.next();
switch (select2.charAt(0)) {
case 'a': {
  System.out.println("1.Outdoor Patient\n 2.Indoor Patient");
  int select3=i.nextInt();

  switch (select3) {
    case 1:
      int a,d;
      String b;
      String c;
      System.out.println("Patient ID :");
      a=i.nextInt();
      System.out.println("Patient Name:");
      b=i.next();
      System.out.println("Patient Age:");
      d=i.nextInt();
      System.out.println("Patient Gender:");
      c=i.next();
      Patients p1 = new OutdoorPatient(a,b,c,d);
      PatientsList.add(p1);
      write.print(p1);
      System.out.println("Patient added \n");
      break;
    case 2:
      int e,h,j,k,l,m;
      String f;
      String g;
      System.out.println("Patient ID :");
      e=i.nextInt();
      System.out.println("Patient Name:");
      f=i.next();
      System.out.println("Patient Age:");
      h=i.nextInt();
      System.out.println("Patient Gender:");
      g=i.next();
      System.out.println("Bed Fee:");
      j=i.nextInt();
      System.out.println("Medicine Fee: ");
      k=i.nextInt();
      System.out.println("Labtest Fee: ");
      l=i.nextInt();
      System.out.println("Room number: ");
      m=i.nextInt();
      Patients p2 = new IndoorPatient(e,f,g,h,j,k,l,m);
      PatientsList.add(p2);
      write.print(p2);
      System.out.println("Patient added \n");

      break;
    default:
          System.out.println("Invalid choice");

   }}

   break;
   case 'b':
   { Doctors doc=new Doctors();
   System.out.println("Enter Doctor Name: ");
   doc.setdoctorId(i.next());

   System.out.println("Enter Doctor Id: ");
   doc.setdoctorId(i.nextInt());

   System.out.println("Enter Department: ");
   doc.setdepartment(i.next());

   System.out.println("Enter Doctor Specialization: ");
   doc.setdoctorSpecialization(i.next());

   System.out.println("Enter Salary: ");
   doc.setDoctorSalary(i.nextInt());

   System.out.println("Enter Phonenumber");
     doc.setphoneNumber(i.next());

    System.out.println("Enter Adress: ");
    doc.setaddress(i.next());
    DoctorsList.add(doc);
    write.print(doc);
    System.out.println("Doctor added \n");
```

```
}
    break;
    case 'c':
    {
        Employees emp=new Employees();
        System.out.println("Enter Employee Name: ");
    emp.setemployeeName(i.next());

    System.out.println("Enter Employee Id: ");
    emp.setemployeeId(i.nextInt());

    System.out.println("Enter Working Shift: ");
    emp.setshift(i.next());

    System.out.println("Enter Designation: ");
    emp.setdesignation(i.next());

    System.out.println("Enter Salary: ");
    emp.setSalary(i.nextInt());

    System.out.println("Enter Phonenumber");
       emp.setphoneNumber(i.next());

   System.out.println("Enter Adress: ");
   emp.setaddress(i.next());
   EmployeesList.add(emp);
   write.print(emp);
   System.out.println("Doctor added \n");
    }
    break;
    case 'd': continue x;
    default: {
System.out.println("Not a valid choice " +select2.charAt(0));}

}}}
case'2':{
    while (true){
System.out.println("a. Delete a Patient");
System.out.println("b. Deelete a Doctor");
System.out.println("c. Delete an Emplyoee");
System.out.println("d. Go To Main Menu");

String select4= i.next();
switch (select4.charAt(0)) {
case 'a': {
System.out.println("Patient Id:");
int pID =i.nextInt();
i.nextLine();
for (int x = 0; x < PatientsList.size(); x++) {
if (pID == PatientsList.get(x).getPatientID()) {
PatientsList.remove(x);
System.out.println("Patient removed \n");
}
else{
if( x == PatientsList.size()-1 )
System.out.println("No Patient data found for " + pID);
}
}
}
break;
```

```
case 'b': {
System.out.println("Doctor Id:");
String dID = i.next();
for (int y = 0; y < DoctorsList.size(); y++) {
if (dID ==DoctorsList.get(y).getdoctorId()) {
DoctorsList.remove(y);
System.out.println("Doctor removed \n");
}
else{
if( y == DoctorsList.size() -1)
System.out.println("No Doctor data found for " + dID);
}
}
}
break;
case 'c': {
System.out.println ("Employee Id:");
int eID = i.nextInt();
i.nextLine();
for (int z = 0; z < EmployeesList.size(); z++) {
if ( eID == EmployeesList.get(z).getemployeeId() ) {
EmployeesList.remove(z);
System.out.println("Employee removed \n");
}
else{
if( z == EmployeesList.size()-1)
System.out.println("No Employee data found for " + eID);
}
}
}
```

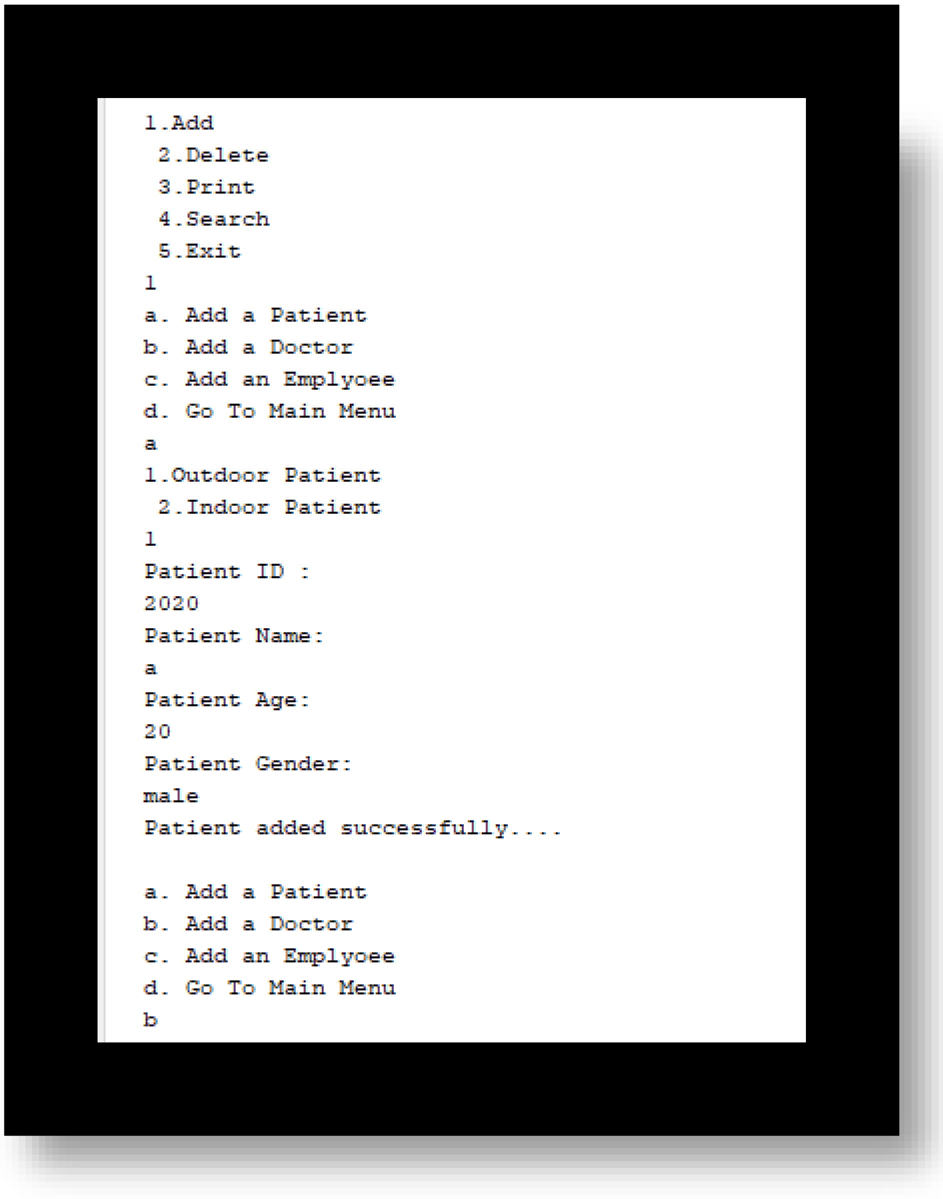
```
break;
case 'd': {
continue x;
}
default: {
System.out.println("Not a valid choice " +select4.charAt(0));
}

    }
}
}
case '3':{

while (true){
System.out.println("a. Print Hospital Information");
System.out.println("b. Print Patients List");
System.out.println("c. Print Doctors List");
System.out.println("d. Print Employees List");
System.out.println("e. Go To Main Menu");
  String select5 = i.next();
switch (select5.charAt(0)) {

case 'a':{  HospitalInformation hos=new HospitalInformation();

hos.hospitalInformation();
}
break;
case 'b': {
System.out.println(PatientsList);
}
```

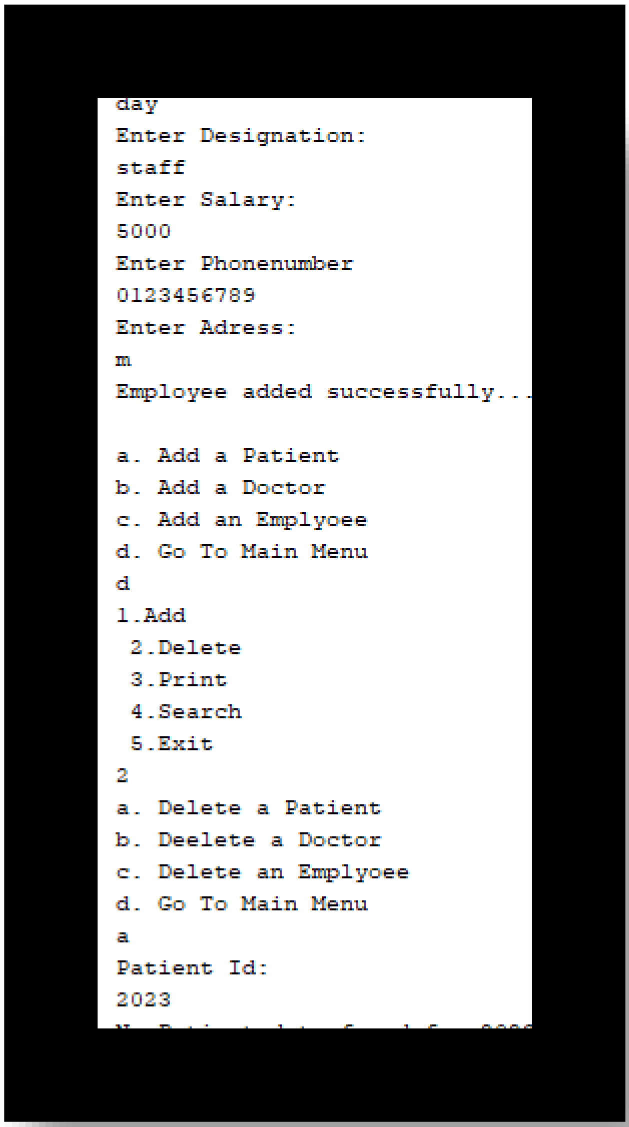
```
break;
case 'c': {
System.out.println(DoctorsList);
}

break;
case 'd': {
System.out.println(EmployeesList);
}

break;
case 'e': {
continue x;
}

default: {
System.out.println("No option in character " +select5.charAt(0));
}
}}}
case '4':{
   while (true) {
System.out.println("a. Search a Patient");
System.out.println("b. Search a Doctor");
System.out.println("c. Search a Employee");
System.out.println("d. Go To Main Menu");
String select6 = i.next();
switch (select6.charAt(0)) {
case 'a': {
System.out.println("Enter Patient ID:");
```

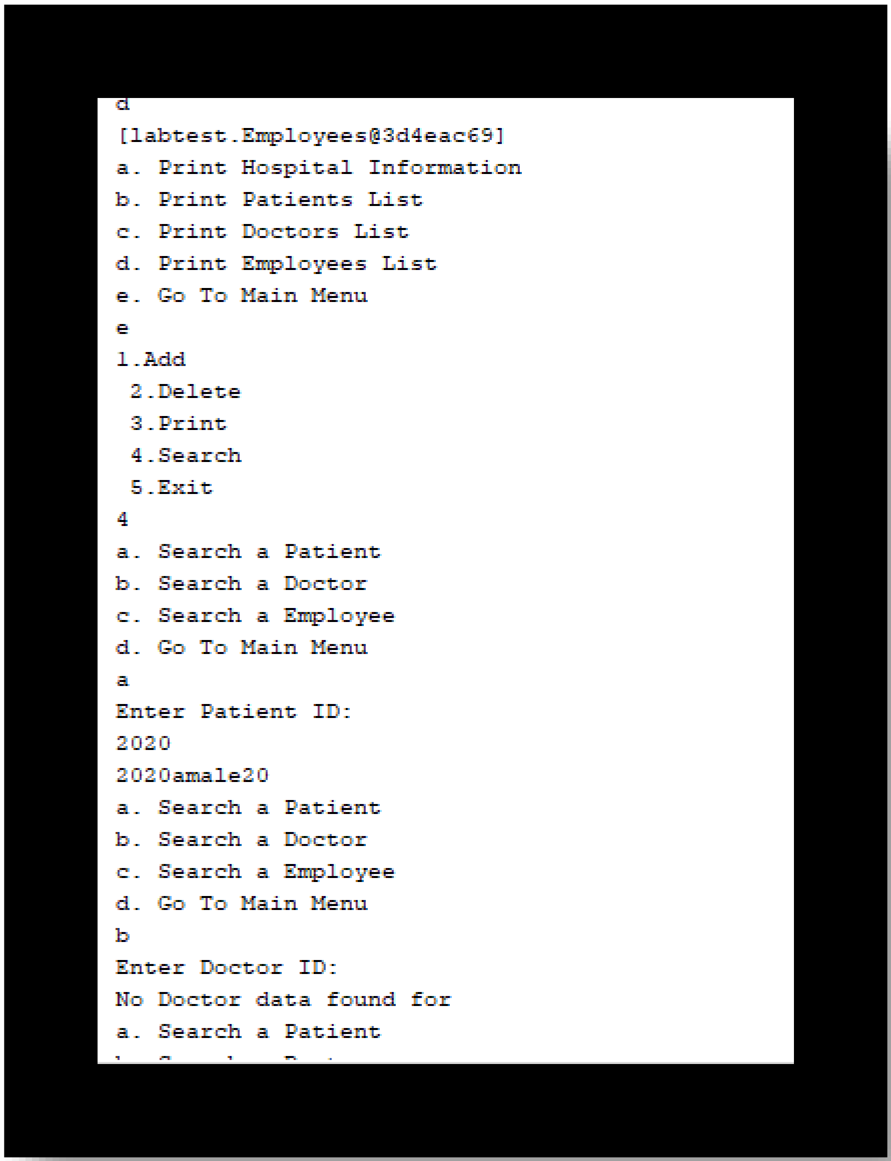
```
int pID = i.nextInt();

for (int x = 0; x < PatientsList.size(); x++) {
if (pID == PatientsList.get(x).getPatientID()) {
System.out.println(PatientsList.get(x).toString());
}
else{
if( x == PatientsList.size() - 1)
System.out.println("No Patient data found for " + pID);
}
}
}
break;
case 'b': {
System.out.println("Enter Doctor ID:");
String dID = i.next();
for (int y = 0; y< DoctorsList.size(); y++) {
if (dID == DoctorsList.get(y).getdoctorId()) {
System.out.println(DoctorsList.get(y).toString());
}
else{
if( y == DoctorsList.size() - 1)
System.out.println("No Doctor data found for " + dID);
}
}
}
break;
case 'c': {
System.out.println("Entter Employee ID:");
int eID = i.nextInt();
```

```
i.nextLine();
for (int z = 0; z < EmployeesList.size(); z++) {
if (eID == EmployeesList.get(z).getemployeeId()) {
System.out.println(EmployeesList.get(z).toString());
}
else{
if( z == EmployeesList.size() - 1)
System.out.println("No Employee data found for " + eID);
}
}}
break;
case'd': continue x;

default: {
System.out.println("No option in character " +select6.charAt(0));
}}}}



case '5': {break x;}
default: {
System.out.println("No option in character " +select1.charAt(0));

}
}}}}
```

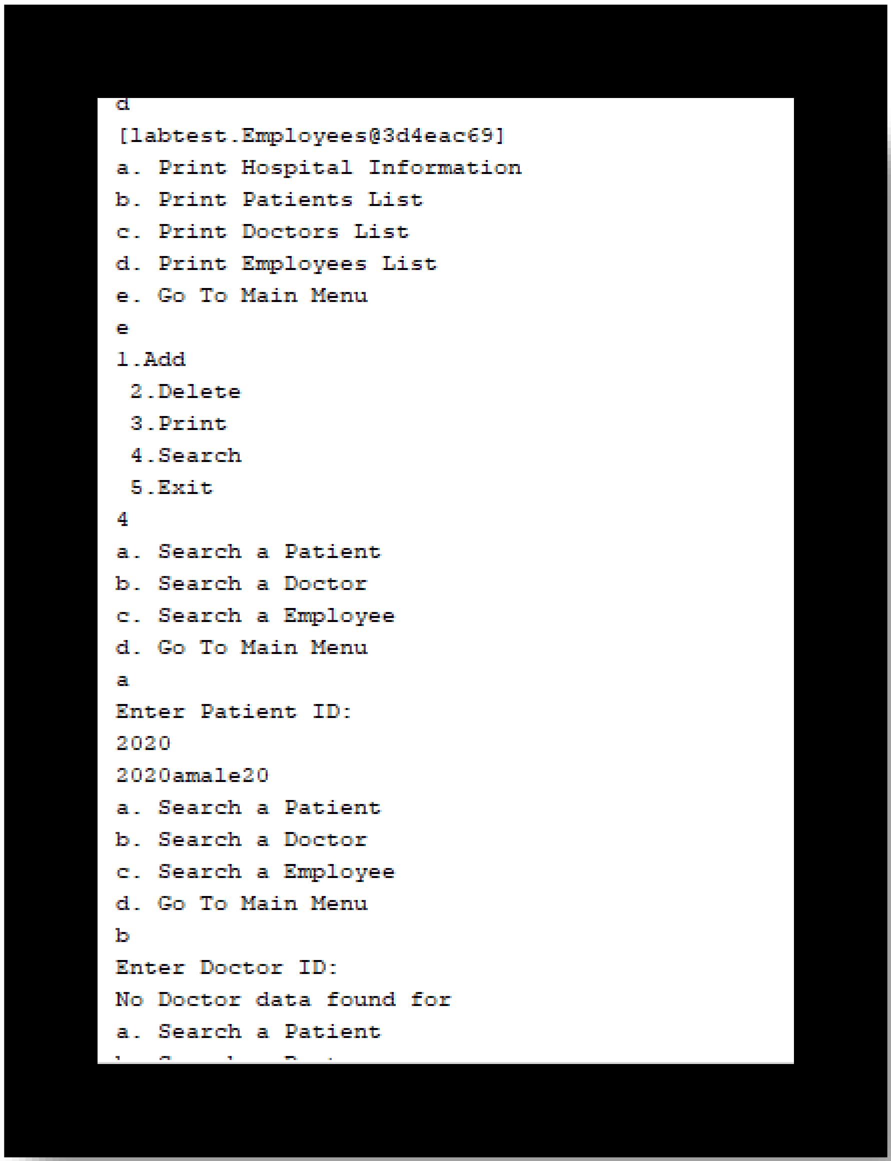
Output:


```
1.Add
 2.Delete
 3.Print
 4.Search
 5.Exit
1
a. Add a Patient
b. Add a Doctor
c. Add an Emplyoee
d. Go To Main Menu
a
1.Outdoor Patient
 2.Indoor Patient
1
Patient ID :
2020
Patient Name:
a
Patient Age:
20
Patient Gender:
male
Patient added successfully....

a. Add a Patient
b. Add a Doctor
c. Add an Emplyoee
d. Go To Main Menu
b
```

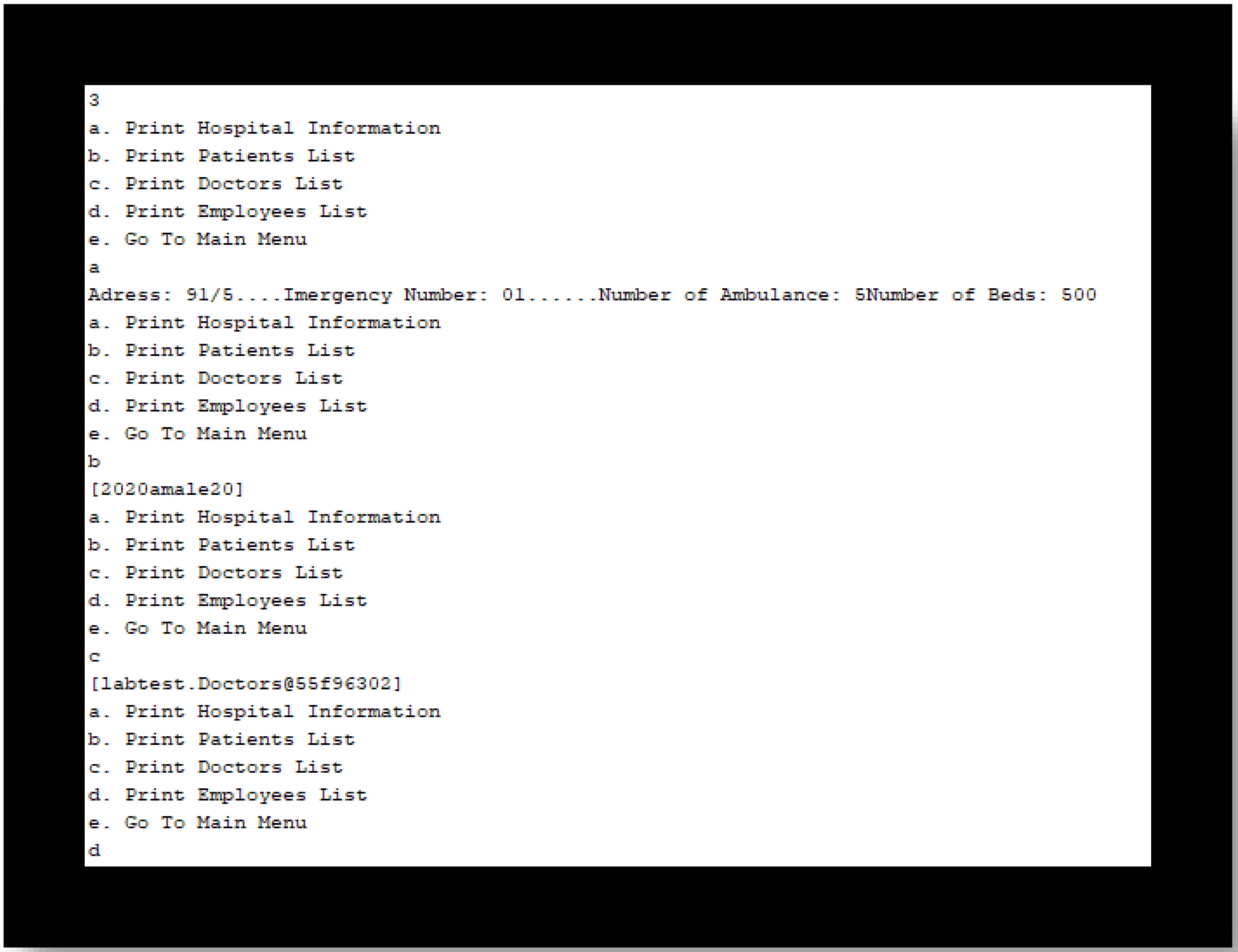

```
day
Enter Designation:
staff
Enter Salary:
5000
Enter Phonenumber
0123456789
Enter Adress:
m
Employee added successfully...

a. Add a Patient
b. Add a Doctor
c. Add an Emplyoee
d. Go To Main Menu
d
1.Add
 2.Delete
 3.Print
 4.Search
 5.Exit
2
a. Delete a Patient
b. Deelete a Doctor
c. Delete an Emplyoee
d. Go To Main Menu
a
Patient Id:
2023
```



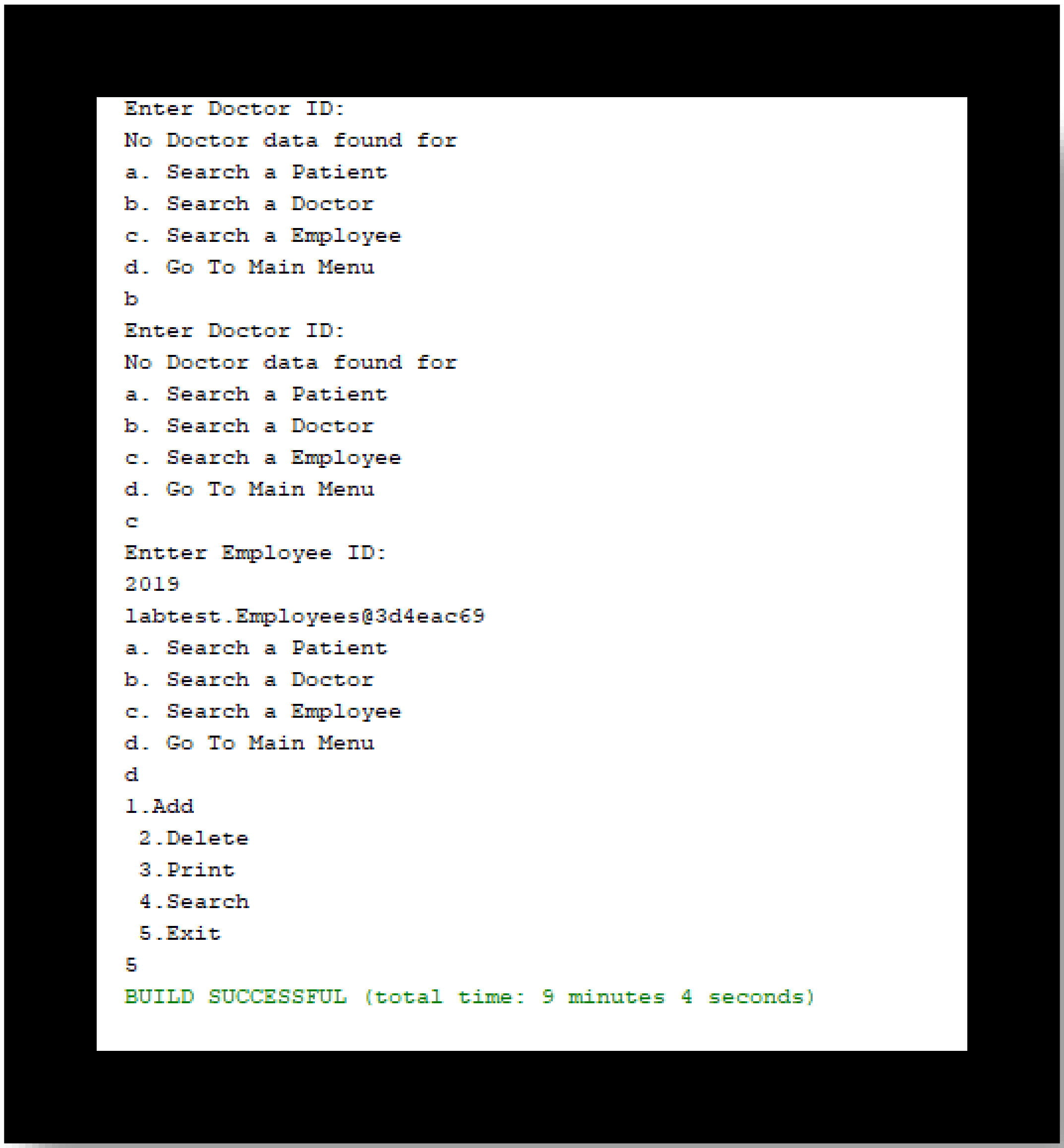
```
d
[labtest.Employees@3d4eac69]
a. Print Hospital Information
b. Print Patients List
c. Print Doctors List
d. Print Employees List
e. Go To Main Menu
e
1.Add
 2.Delete
 3.Print
 4.Search
 5.Exit
4
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
a
Enter Patient ID:
2020
2020amale20
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
b
Enter Doctor ID:
No Doctor data found for
a. Search a Patient
```



```
d
[labtest.Employees@3d4eac69]
a. Print Hospital Information
b. Print Patients List
c. Print Doctors List
d. Print Employees List
e. Go To Main Menu
e
1.Add
 2.Delete
 3.Print
 4.Search
 5.Exit
4
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
a
Enter Patient ID:
2020
2020amale20
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
b
Enter Doctor ID:
No Doctor data found for
a. Search a Patient
```



```
3
a. Print Hospital Information
b. Print Patients List
c. Print Doctors List
d. Print Employees List
e. Go To Main Menu
a
Adress: 91/5....Imergency Number: 01......Number of Ambulance: 5Number of Beds: 500
a. Print Hospital Information
b. Print Patients List
c. Print Doctors List
d. Print Employees List
e. Go To Main Menu
b
[2020amale20]
a. Print Hospital Information
b. Print Patients List
c. Print Doctors List
d. Print Employees List
e. Go To Main Menu
c
[labtest.Doctors@55f96302]
a. Print Hospital Information
b. Print Patients List
c. Print Doctors List
d. Print Employees List
e. Go To Main Menu
d
```



```
Enter Doctor ID:
No Doctor data found for
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
b
Enter Doctor ID:
No Doctor data found for
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
c
Entter Employee ID:
2019
labtest.Employees@3d4eac69
a. Search a Patient
b. Search a Doctor
c. Search a Employee
d. Go To Main Menu
d
1.Add
 2.Delete
 3.Print
 4.Search
 5.Exit
5
BUILD SUCCESSFUL (total time: 9 minutes 4 seconds)
```