**EX NO: 3**

**DISTRIBUTED COMPONENTS LABORATORY**

**JAVA RMI**

**DATE: 18-10-2020**

**Aim:**

To implement client sever programs using JAVA RMI.

**Question 1:**

**Develop a simple client/server banking service application using Java RMI. Whenever a user-logins, the user is allowed to perform the following operations:**

**Deposit operation: one can deposit some amount to an existing account**

**Withdraw operation: one can withdraw some amount from an existing account. While withdrawing, it should check for minimum balance. The minimum balance for an account is 500.**

**Balance Enquiry: one can check the balance of an account**

**Transfer operation: one can transfer some amount to another existing account. While Transferring check for Minimum balance.**

**If there isn't sufficient money in the account, then an error message should be returned. Check the existence of the account(s) in all the operations and return an error message if it doesn’t exist.**

**Code:**

**RMI\_Interface.java**

public interface RMI\_Interface extends java.rmi.Remote

{

    public String create(String accno,String password) throws java.rmi.RemoteException;

    public String login(String accno, String password) throws java.rmi.RemoteException;

    public String deposit(String accno, int amount) throws java.rmi.RemoteException;

    public String withdraw(String accno, int amount) throws java.rmi.RemoteException;

    public String balance(String accno) throws java.rmi.RemoteException;

    public String transfer(String sender, String reciever, int amount) throws java.rmi.RemoteException;

}

**RMI\_Impl.java:**

import java.rmi.\*;

import java.rmi.server.UnicastRemoteObject;

import java.util.\*;

class Account

{

    public String name;

    public String password;

    public int balance;

    Account(String name,String password)

    {

        this.name = name;

        this.password = password;

        this.balance = 0;

    }

}

public class RMI\_Impl extends UnicastRemoteObject implements RMI\_Interface

{

    private static final long serialVersionUID = 1L;

    ArrayList<Account> accounts = new ArrayList<>();

    public RMI\_Impl() throws RemoteException

    {

    }

    public String create(String accno,String password)

    {

        int flag = -1;

        for(int i=0;i<accounts.size();i++)

        {

            if(accounts.get(i).name.equals(accno))

            {

                flag = 361;

                break;

            }

        }

        if(flag==361)

        {

            return "Account already exits!";

        }

        else

        {

            accounts.add(new Account(accno,password));

            return "Account created successfully!";

        }

    }

    public String login(String accno, String password)

    {

        int flag = -1;

        for(int i=0;i<accounts.size();i++)

        {

            if(accounts.get(i).name.equals(accno) && accounts.get(i).password.equals(password))

            {

                flag = 361;

                break;

            }

        }

        if(flag==361)

        {

            return "SUCCESS";

        }

        else

        {

            return "FAIL";

        }

    }

    public String deposit(String accno, int amount)

    {

        int flag = -1;

        for(int i=0;i<accounts.size();i++)

        {

            if(accounts.get(i).name.equals(accno))

            {

                flag = i;

                break;

            }

        }

        if(flag!=-1)

        {

            accounts.get(flag).balance += amount;

            return "Deposit successful! Current balance: " + Integer.toString(accounts.get(flag).balance);

        }

        else

        {

            return "No account exists!";

        }

    }

    public String withdraw(String accno, int amount)

    {

        int flag = -1;

        for(int i=0;i<accounts.size();i++)

        {

            if(accounts.get(i).name.equals(accno))

            {

                flag = i;

                break;

            }

        }

        if(flag!=-1)

        {

            int curBalance = accounts.get(flag).balance - amount;

            if(curBalance < 500)

            {

                return "Transaction not allowed!";

            }

            else

            {

                accounts.get(flag).balance -= amount;

                return "Withdraw successful! Current balance: " + Integer.toString(accounts.get(flag).balance);

            }

        }

        else

        {

            return "No account exists!";

        }

    }

    public String balance(String accno)

    {

        int flag = -1;

        for(int i=0;i<accounts.size();i++)

        {

            if(accounts.get(i).name.equals(accno))

            {

                flag = i;

                break;

            }

        }

        if(flag!=-1)

        {

            return "Your balance is: " + Integer.toString(accounts.get(flag).balance);

        }

        else

        {

            return "No account exists!";

        }

    }

    public String transfer(String sender, String reciever, int amount)

    {

        int flag1 = -1, flag2 = -1;

        for(int i=0;i<accounts.size();i++)

        {

            if(accounts.get(i).name.equals(sender))

            {

                flag1 = i;

            }

            if(accounts.get(i).name.equals(reciever))

            {

                flag2 = i;

            }

        }

        if(flag1==-1 || flag2==-1)

        {

            return "Sender/Reciever account do no exists!";

        }

        else

        {

            int curBalance = accounts.get(flag1).balance - amount;

            if(curBalance < 500)

            {

                return "Transaction not allowed!";

            }

            else

            {

                accounts.get(flag1).balance -= amount;

                accounts.get(flag2).balance += amount;

                return "Transfer successful! Current balance: " + Integer.toString(accounts.get(flag1).balance);

            }

        }

    }

}

**RMI\_Server.java:**

import java.rmi.\*;

public class RMI\_Server {

    public static void main(String args[])

    {

        try

        {

            RMI\_Impl implclass=new RMI\_Impl();

            Naming.rebind("RMI\_INSTANCE",implclass);

        }

        catch (Exception e)

        {

            System.out.println("Exception occured:" + e);

        }

    }

}

**RMI\_Client.java:**

import java.rmi.\*;

import java.util.\*;

public class RMI\_Client

{

    public static void main(String[] argv)

    {

        if (argv.length !=1)

        {

            System.out.println("Usage: java RMI\_Client &lt;IP addr of host running RMI Server");

            System.exit(0);

        }

        String serverName = argv[0];

        try

        {

            RMI\_Interface obj = (RMI\_Interface)Naming.lookup("rmi://"+serverName+"/RMI\_INSTANCE");

            Scanner sc = new Scanner(System.in);

            boolean loginFlag = false;

            while(true)

            {

                System.out.println("1. Create account\n2. Login\n3. Deposit\n4. Withdraw\n5. Balance\n6. Transfer\n7.Exit\n");

                int choice = sc.nextInt();

                if(choice==1)

                {

                    System.out.println("Enter your account number: ");

                    String accno = sc.next();

                    System.out.println("Enter your password: ");

                    String password = sc.next();

                    String output = obj.create(accno, password);

                    System.out.println(output);

                }

                else if(choice==2)

                {

                    System.out.println("Enter your account number: ");

                    String accno = sc.next();

                    System.out.println("Enter your password: ");

                    String password = sc.next();

                    String output = obj.login(accno, password);

                    if(output.equals("SUCCESS"))

                    {

                        loginFlag = true;

                    }

                }

                else if(choice==3)

                {

                    if(loginFlag==true)

                    {

                        System.out.println("Enter your account number: ");

                        String accno = sc.next();

                        System.out.println("Enter amount to deposit: ");

                        int amount = sc.nextInt();

                        String output = obj.deposit(accno, amount);

                        System.out.println(output);

                    }

                    else

                    {

                        System.out.println("Login first!");

                    }

                }

                else if(choice==4)

                {

                    if(loginFlag==true)

                    {

                        System.out.println("Enter your account number: ");

                        String accno = sc.next();

                        System.out.println("Enter amount to withdraw: ");

                        int amount = sc.nextInt();

                        String output = obj.withdraw(accno, amount);

                        System.out.println(output);

                    }

                    else

                    {

                        System.out.println("Login first!");

                    }

                }

                else if(choice==5)

                {

                    if(loginFlag==true)

                    {

                        System.out.println("Enter your account number: ");

                        String accno = sc.next();

                        String output = obj.balance(accno);

                        System.out.println(output);

                    }

                    else

                    {

                        System.out.println("Login first!");

                    }

                }

                else if(choice==6)

                {

                    if(loginFlag==true)

                    {

                        System.out.println("Enter your account number: ");

                        String accno1 = sc.next();

                        System.out.println("Enter your friend's number: ");

                        String accno2 = sc.next();

                        System.out.println("Enter amount to transfer: ");

                        int amount = sc.nextInt();

                        String output = obj.transfer(accno1,accno2,amount);

                        System.out.println(output);

                    }

                    else

                    {

                        System.out.println("Login first!");

                    }

                }

                else if(choice==7)

                {

                    break;

                }

            }

            sc.close();

        }

        catch(Exception e)

        {

            System.out.println("Exception Occured " + e);

            System.exit(0);

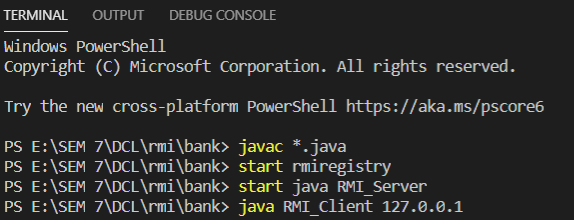
        }

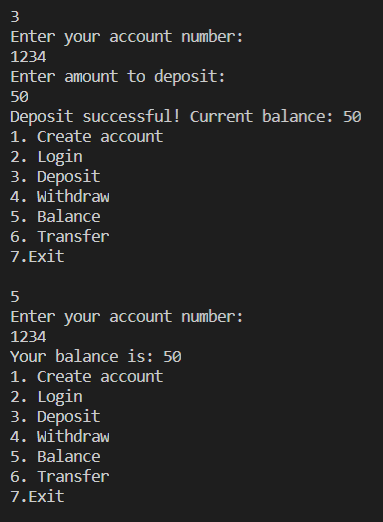
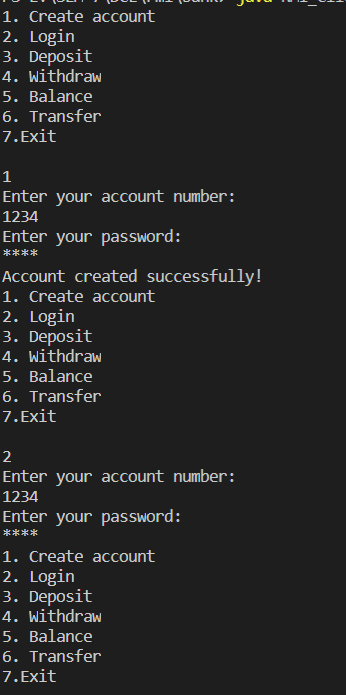
        System.out.println("Connection successful!");

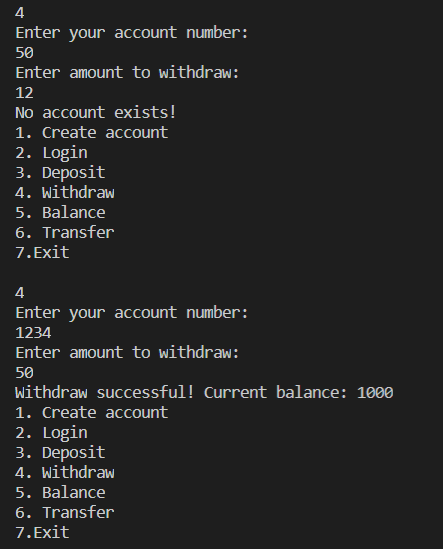
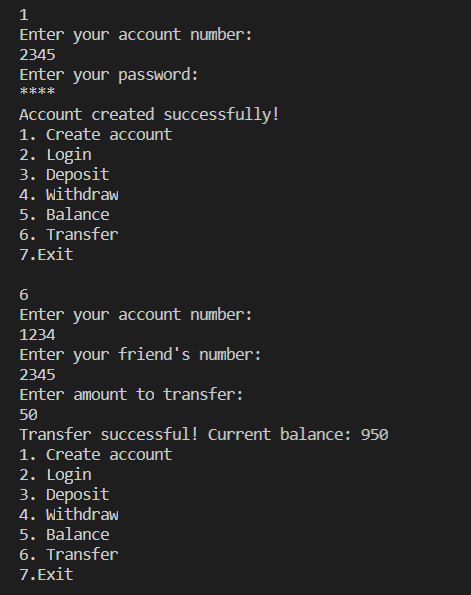
    }

}

**Output:**

****

****

****

**Question 2:**

**Develop a Patient Information system using Java RMI. This information system keeps track of the patient’s information and treatment details in a hospital or clinic. The Patient Information system offers following operations:**

**a.     Register patient details**

**b.    Search for Patient details by name**

**Use java swing or frames for the user interface.**

**Code:**

**RMI\_Interface.java:**

public interface RMI\_Interface extends java.rmi.Remote

{

    public String register(String patitenid, String name,String phoneNumber,int age,String entryDate, String dept, String disease, String doctor) throws java.rmi.RemoteException;

    public String search(String name) throws java.rmi.RemoteException;

}

**RMI\_Impl.java:**

import java.rmi.\*;

import java.rmi.server.UnicastRemoteObject;

import java.util.\*;

class Patient

{

    String patientid;

    String name;

    String phoneNumber;

    int age;

    String entryDate;

    String dept;

    String disease;

    String doctor;

    Patient(String patitenid, String name,String phoneNumber,int age,String entryDate,String dept,String disease, String doctor)

    {

        this.patientid = patitenid;

        this.name = name;

        this.phoneNumber = phoneNumber;

        this.age = age;

        this.entryDate = entryDate;

        this.dept = dept;

        this.disease = disease;

        this.doctor = doctor;

    }

    public String toString()

    {

        return this.patientid + "," + this.name + "," + this.phoneNumber + "," + this.age + "," + this.entryDate + "," + this.dept + "," + this.disease + "," + this.doctor;

    }

}

public class RMI\_Impl extends UnicastRemoteObject implements RMI\_Interface

{

    private static final long serialVersionUID = 1L;

    ArrayList<Patient> patients = new ArrayList<>();

    public RMI\_Impl() throws RemoteException

    {

    }

    public String register(String patitenid, String name,String phoneNumber,int age,String entryDate,String dept,String disease, String doctor)

    {

        Patient p = new Patient(patitenid,name,phoneNumber,age,entryDate,dept,disease,doctor);

        patients.add(p);

        return "SUCCESS";

    }

    public String search(String name)

    {

        for(int i=0;i<patients.size();i++)

        {

            if(patients.get(i).name.equals(name))

            {

                return patients.get(i).toString();

            }

        }

        return "NOT FOUND";

    }

}

**RMI\_Server.java:**

import java.rmi.\*;

public class RMI\_Server {

    public static void main(String args[])

    {

        try

        {

            RMI\_Impl implclass=new RMI\_Impl();

            Naming.rebind("RMI\_INSTANCE",implclass);

        }

        catch (Exception e)

        {

            System.out.println("Exception occured:" + e);

        }

    }

}

**RMI\_Client.java:**

import java.rmi.\*;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import javax.swing.JFrame;

import javax.swing.border.Border;

public class RMI\_Client

{

    public static void main(String[] argv)

    {

        if (argv.length !=1)

        {

            System.out.println("Usage: java RMI\_Client &lt;IP addr of host running RMI Server");

            System.exit(0);

        }

        String serverName = argv[0];

        try

        {

            RMI\_Interface myobj = (RMI\_Interface)Naming.lookup("rmi://"+serverName+"/RMI\_INSTANCE");

            JFrame jf = new JFrame();

            jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

            jf.setExtendedState(JFrame.MAXIMIZED\_BOTH);

            JButton register = new JButton("REGISTER");

            JButton search = new JButton("SEARCH");

            Label l1 = new Label("HOSPITAL MANAGEMENT SYSTEM");

            Font myFont = new Font("Times New Roman",Font.BOLD,20);

            Border border = BorderFactory.createLineBorder(new Color(255, 77, 77), 2);

            l1.setFont(myFont);

            l1.setBounds(650,120,350,40);

            register.setBounds(710,200,220,40);

            search.setBounds(710,280,220,40);

            register.addActionListener(new ActionListener()

            {

                public void actionPerformed(ActionEvent e)

                {

                    try

                    {

                        JFrame jf\_r = new JFrame();

                        jf\_r.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                        jf\_r.setExtendedState(JFrame.MAXIMIZED\_BOTH);

                        Label center = new Label("HOSPITAL MANAGEMENT SYSTEM");

                        Label l1 = new Label("Patient ID");

                        Label l2 = new Label("Name");

                        Label l3 = new Label("Phone Number");

                        Label l4 = new Label("Age");

                        Label l5 = new Label("Entry Date");

                        Label l6 = new Label("Department");

                        Label l7 = new Label("Status");

                        Label l8 = new Label("Doctor");

                        JTextField jt1 = new JTextField("");

                        JTextField jt2 = new JTextField("");

                        JTextField jt3 = new JTextField("");

                        JTextField jt4 = new JTextField("");

                        JTextField jt5 = new JTextField("");

                        JTextField jt6 = new JTextField("");

                        JTextField jt7 = new JTextField("");

                        JTextField jt8 = new JTextField("");

                        JButton register\_button = new JButton("REGISTER");

                        center.setFont(myFont);

                        l1.setFont(myFont);

                        l2.setFont(myFont);

                        l3.setFont(myFont);

                        l4.setFont(myFont);

                        l5.setFont(myFont);

                        l6.setFont(myFont);

                        l7.setFont(myFont);

                        l8.setFont(myFont);

                        jt1.setFont(myFont);

                        jt2.setFont(myFont);

                        jt3.setFont(myFont);

                        jt4.setFont(myFont);

                        jt5.setFont(myFont);

                        jt6.setFont(myFont);

                        jt7.setFont(myFont);

                        jt8.setFont(myFont);

                        jt1.setBorder(border);

                        jt2.setBorder(border);

                        jt3.setBorder(border);

                        jt4.setBorder(border);

                        jt5.setBorder(border);

                        jt6.setBorder(border);

                        jt7.setBorder(border);

                        jt8.setBorder(border);

                        center.setBounds(650,120,350,40);

                        l1.setBounds(650,200,180,40);

                        l2.setBounds(650,250,180,40);

                        l3.setBounds(650,300,180,40);

                        l4.setBounds(650,350,180,40);

                        l5.setBounds(650,400,180,40);

                        l6.setBounds(650,450,180,40);

                        l7.setBounds(650,500,180,40);

                        l8.setBounds(650,550,180,40);

                        jt1.setBounds(850,200,220,40);

                        jt2.setBounds(850,250,220,40);

                        jt3.setBounds(850,300,220,40);

                        jt4.setBounds(850,350,220,40);

                        jt5.setBounds(850,400,220,40);

                        jt6.setBounds(850,450,220,40);

                        jt7.setBounds(850,500,220,40);

                        jt8.setBounds(850,550,220,40);

                        register\_button.setBounds(700,650,220,40);

                        register\_button.addActionListener(new ActionListener()

                        {

                            public void actionPerformed(ActionEvent e)

                            {

                                try

                                {

                                    String result = myobj.register(jt1.getText(), jt2.getText(), jt3.getText(),Integer.parseInt(jt4.getText()),jt5.getText(),jt6.getText(),jt7.getText(),jt8.getText());

                                    JOptionPane.showMessageDialog(null,result,"Message",JOptionPane.PLAIN\_MESSAGE);

                                }

                                catch(Exception d1)

                                {

                                    System.out.println(d1);

                                }

                            }

                        });

                        jf\_r.add(center);

                        jf\_r.add(l1);

                        jf\_r.add(l2);

                        jf\_r.add(l3);

                        jf\_r.add(l4);

                        jf\_r.add(l5);

                        jf\_r.add(l6);

                        jf\_r.add(l7);

                        jf\_r.add(l8);

                        jf\_r.add(jt1);

                        jf\_r.add(jt2);

                        jf\_r.add(jt3);

                        jf\_r.add(jt4);

                        jf\_r.add(jt5);

                        jf\_r.add(jt6);

                        jf\_r.add(jt7);

                        jf\_r.add(jt8);

                        jf\_r.add(register\_button);

                        jf\_r.setLayout(null);

                        jf\_r.setVisible(true);

                    }

                    catch(Exception e1)

                    {

                        System.out.println(e1);

                    }

                }

            });

            search.addActionListener(new ActionListener()

            {

                public void actionPerformed(ActionEvent e)

                {

                    try

                    {

                        JFrame jf\_s = new JFrame();

                        jf\_s.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                        jf\_s.setExtendedState(JFrame.MAXIMIZED\_BOTH);

                        Label center = new Label("HOSPITAL MANAGEMENT SYSTEM");

                        Label l1 = new Label("Patient name");

                        JTextField jt1 = new JTextField("");

                        JButton search\_button = new JButton("SEARCH");

                        center.setFont(myFont);

                        l1.setFont(myFont);

                        jt1.setFont(myFont);

                        search\_button.setFont(myFont);

                        center.setBounds(650,120,350,40);

                        l1.setBounds(650,200,180,40);

                        jt1.setBounds(850,200,220,40);

                        search\_button.setBounds(700,280,220,40);

                        search\_button.addActionListener(new ActionListener()

                        {

                            public void actionPerformed(ActionEvent e)

                            {

                                try

                                {

                                    String result = myobj.search(jt1.getText());

                                    String unpack[] = result.split(",");

                                    if(unpack.length<8)

                                    {

                                        JOptionPane.showMessageDialog(null,"Not enough values!","Error",JOptionPane.ERROR\_MESSAGE);

                                    }

                                    Label l1 = new Label("Patient ID: " + unpack[0]);

                                    Label l2 = new Label("Name: " + unpack[1]);

                                    Label l3 = new Label("Phone Number: " + unpack[2]);

                                    Label l4 = new Label("Age: " + unpack[3]);

                                    Label l5 = new Label("Entry date: " + unpack[4]);

                                    Label l6 = new Label("Department: " + unpack[5]);

                                    Label l7 = new Label("Disease: " + unpack[6]);

                                    Label l8 = new Label("Doctor: " + unpack[7]);

                                    l1.setFont(myFont);

                                    l2.setFont(myFont);

                                    l3.setFont(myFont);

                                    l4.setFont(myFont);

                                    l5.setFont(myFont);

                                    l6.setFont(myFont);

                                    l7.setFont(myFont);

                                    l8.setFont(myFont);

                                    l1.setBounds(650,360,220,40);

                                    l2.setBounds(650,400,220,40);

                                    l3.setBounds(650,440,220,40);

                                    l4.setBounds(650,480,220,40);

                                    l5.setBounds(650,520,220,40);

                                    l6.setBounds(650,560,220,40);

                                    l7.setBounds(650,600,220,40);

                                    l8.setBounds(650,640,220,40);

                                    jf\_s.add(l1);

                                    jf\_s.add(l2);

                                    jf\_s.add(l3);

                                    jf\_s.add(l4);

                                    jf\_s.add(l5);

                                    jf\_s.add(l6);

                                    jf\_s.add(l7);

                                    jf\_s.add(l8);

                                }

                                catch(Exception d2)

                                {

                                    System.out.println(d2);

                                }

                            }

                        });

                        jf\_s.add(center);

                        jf\_s.add(l1);

                        jf\_s.add(jt1);

                        jf\_s.add(search\_button);

                        jf\_s.setLayout(null);

                        jf\_s.setVisible(true);

                    }

                    catch(Exception d1)

                    {

                        System.out.println(d1);

                    }

                }

            });

            jf.add(l1);

            jf.add(register);

            jf.add(search);

            jf.setLayout(null);

            jf.setVisible(true);

        }

        catch(Exception e)

        {

            System.out.println("Exception Occured " + e);

            System.exit(0);

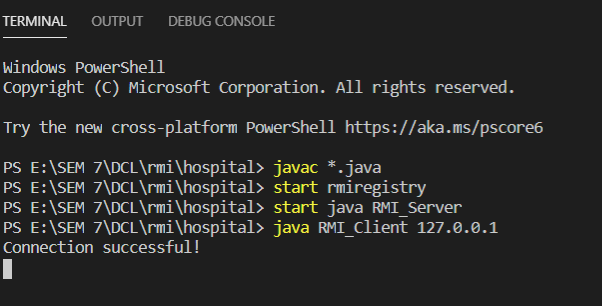
        }

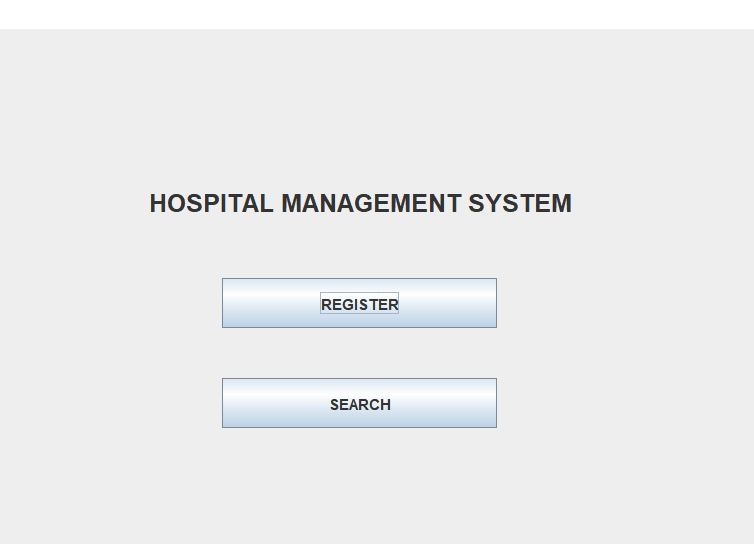
        System.out.println("Connection successful!");

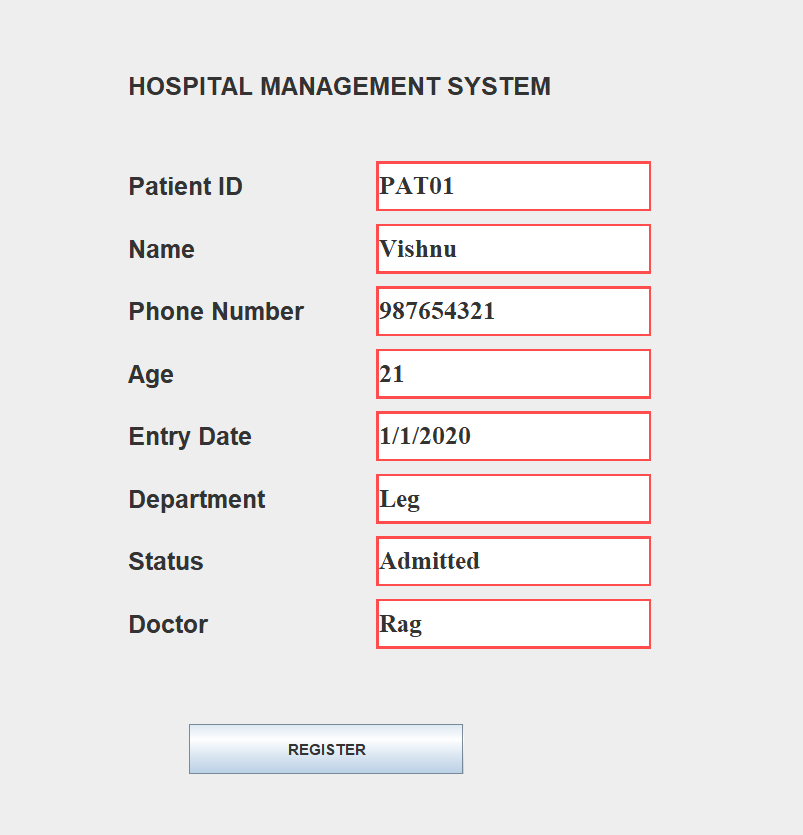
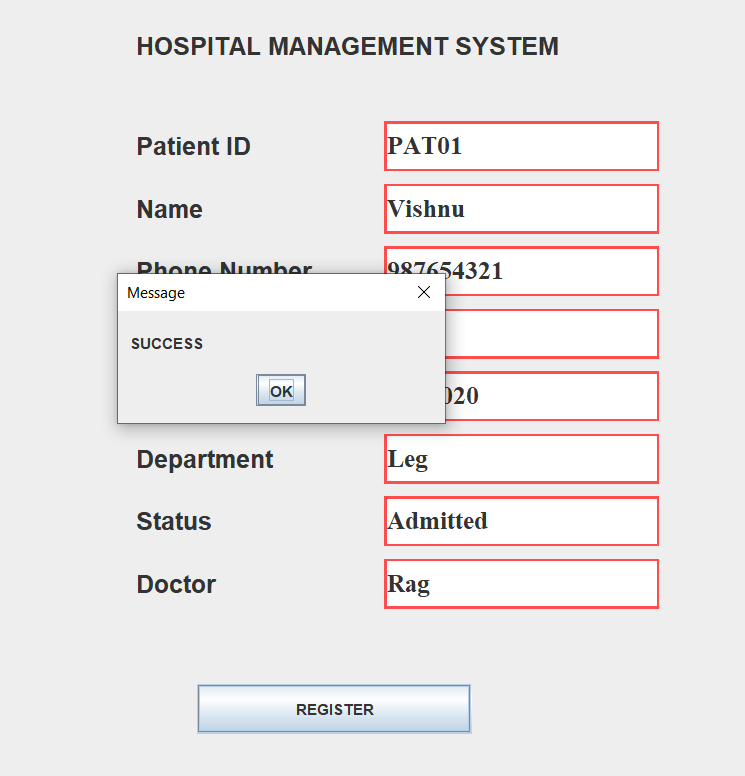
    }

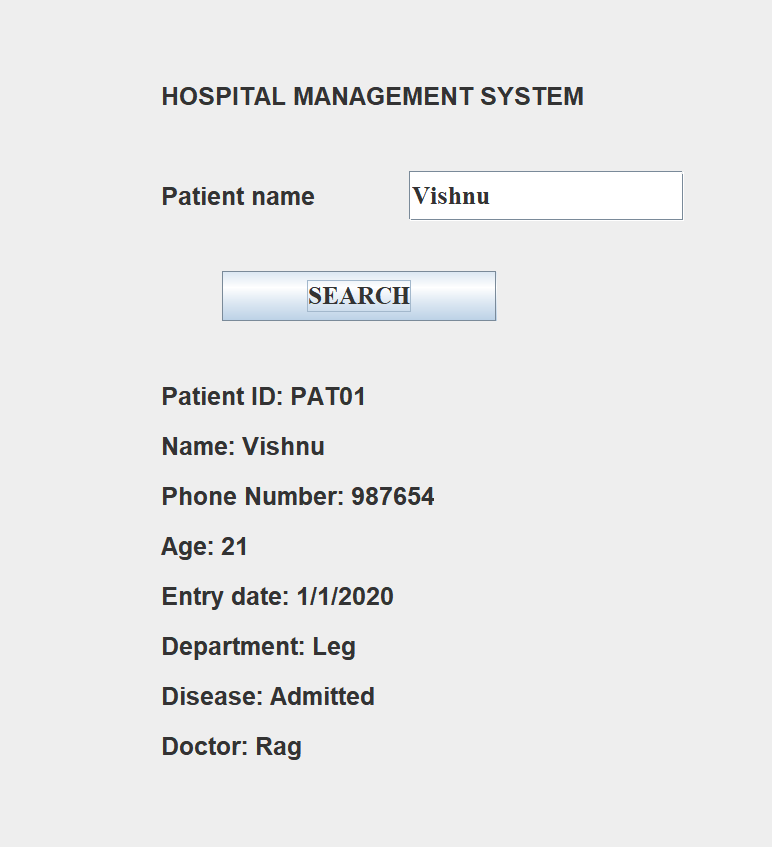
}

**Output:**

****





**Result:**

The given applications have been successfully implemented using JAVA RMI.