**DBMS Lab – Mini Project**

**Hospital Management System**

**Source Code for Table Creation and Population:**

--==========================================================================--

--                            DBMS Lab                              --

--            Mini Project - Hospital Management System             --

--==========================================================================--

--                          Team Members                            --

--              Abirathi M               -  185001006               --

--              Agnus Immaculate Silvia  -  185001011               --

--              Amith Kumar N            -  185001018               --

--              Imran Maajitha S M       -  185001061               --

--==========================================================================--

--  This is the SQL script file for the creation and population of  --

--        data required for the Hospital Management System.         --

--==========================================================================--

--                   \* Do not modify this file \*                    --

--==========================================================================--

REM Dropping the already created tables

drop table Appointments;

drop table Doctor;

drop table Receptionist;

drop table Patient;

drop table Admin;

REM Creating Admin table

create table Admin

(

    username varchar2(20) constraint ADMIN\_PK primary key,

    password varchar2(20)

);

REM Setting the deafult username and passowrd for admin

insert into Admin values('admin', 'admin');

commit;

REM Creating Patient table

create table Patient

(

    patientID varchar2(4) constraint PATIENT\_PK primary key,

    admitDate date,

    pName varchar2(30),

    age number(3),

    gender varchar2(7) constraint PATIENT\_GENDER\_CHECK check (gender in ('Male', 'Female')),

    address varchar2(75),

    city varchar2(20),

    phNo varchar2(10),

    maritalStatus varchar2(10) constraint PATIENT\_MARITAL\_CHECK check (maritalStatus in ('Single', 'Married')),

    ptType varchar2(15) constraint PATIENT\_TYPE\_CHECK check (ptType in ('In patient', 'Out patient')),

    wardNo varchar2(3),

    bedNo varchar2(3)

);

REM Creating Receptionist table

create table Receptionist

(

    receptionistID varchar2(4) constraint RECEPTIONIST\_PK primary key,

    rName varchar2(30),

    age number(3),

    gender varchar2(7) constraint RECEP\_GENDER\_CHECK check (gender in ('Male', 'Female')),

    address varchar2(75),

    city varchar2(20),

    phNo varchar2(10),

    maritalStatus varchar2(10) constraint RECEP\_MARITAL\_CHECK check (maritalStatus in ('Single', 'Married')),

    qualification varchar2(20),

    joinDate date,

    username varchar2(20) constraint RECEPTIONIST\_USERNAME\_UNIQ unique,

    password varchar2(20)

);

REM Creating Doctor table

create table Doctor

(

    doctorID varchar2(4) constraint DOCTOR\_PK primary key,

    dName varchar2(30),

    age number(3),

    gender varchar2(7) constraint DOCTOR\_GENDER\_CHECK check (gender in ('Male', 'Female')),

    address varchar2(75),

    city varchar2(20),

    phNo varchar2(10),

    maritalStatus varchar2(10) constraint DOCTOR\_MARITAL\_CHECK check (maritalStatus in ('Single', 'Married')),

    qualification varchar2(20),

    specialization varchar2(30),

    joinDate date,

    username varchar2(20) constraint DOCTOR\_USERNAME\_UNIQ unique,

    password varchar2(20)

);

REM Creating Appointments table

create table Appointments

(

    appDateTime date,

    doctorID varchar2(4),

    patientID varchar2(4),

    constraint APPOINTMENTS\_PK primary key(appDateTime, doctorID),

    constraint APPOINTMENTS\_DRID\_FK foreign key(doctorID) references Doctor(doctorID),

    constraint APPOINTMENTS\_PTID\_FK foreign key(patientID) references Patient(patientID)

);

REM Populating the relations

--patient(pid,admitdate,pname,age,gen,add,city,pno,martial,ptype,ward,bed)

insert into Patient values('p001','3-MAY-2020','sudha',20,'Female','ashoknagar','chennai','9423456721','Single','Out patient','w07','b03');

insert into Patient values('p002','16-JAN-2020','gobi',27,'Male','palavaram','kanchipuram','805643219','Single','Out patient','w03','b06');

insert into Patient values('p003','7-MAR-2020','govindraj',54,'Male','annanagar','Madurai','9876543218','Married','In patient','w01','b01');

--receptionist(rid,rname,age,gen,add,city,pno,martial,qual,joindate,uname,pwd)

insert into Receptionist values('r001','manju',25,'Female','uthiramerur','kanchipuram','9087066542','Single','BCA','2-FEB-2019','manj1','12manj');

insert into Receptionist values('r002','ram',34,'Male','tambaram','chennai','9329294557','Married','PGDHM','12-JAN-2019','ram1','23ram');

insert into Receptionist values('r003','sahul',40,'Male','chrompet','chennai','8056473821','Married','diploma in HM','22-MAR-2020','sa1','sahul');

--doctor(did,dname,age,gen,add,city,pno,martial,qual,spl,joindate,uname,pwd)

insert into Doctor values('d001','madhavan',47,'Male','ashoknagar','chennai','9867504321','Married','MBBS MD','Cardiologist','02-JAN-2015','mad1','maddy');

insert into Doctor values('d002','swathika',27,'Female','annanagar','kachipuram','8056789012','Single','BDS','Dentist','13-MAR-2017','swa12','swathi');

insert into Doctor values('d003','sariq',35,'Male','ettimadai','coimbatore','9345678282','Married','MBBS MD','Neurologist','12-FEB-2016','sar1','sari');

--appointment(appdatetime,did,pid)

insert into Appointments values(to\_date('12-MAY-2020 9:30:00','dd-mon-yyyy hh24:mi:ss'),'d001','p001');

insert into Appointments values(to\_date('2-FEB-2020 11:00:00','dd-mon-yyyy hh24:mi:ss'),'d003','p002');

insert into Appointments values(to\_date('22-MAR-2020 10:30:00','dd-mon-yyyy hh24:mi:ss'),'d002','p003');

insert into Appointments values(to\_date('7-APR-2020 9:30:00','dd-mon-yyyy hh24:mi:ss'),'d003','p002');

insert into Appointments values(to\_date('11-MAR-2020 12:30:00','dd-mon-yyyy hh24:mi:ss'),'d002','p003');

insert into Appointments values(to\_date('21-JAN-2020 9:30:00','dd-mon-yyyy hh24:mi:ss'),'d001','p001');

**Screenshots and Code:**

**Connectivity Code:**

The code for database connection is written as a static function named ‘execute()’ in a public class named ‘Oracle’ so that it can be called simply as ‘Oracle.execute()’ wherever necessary without rewriting the whole code again.

public class Oracle {

    static ResultSet execute(String query) throws SQLException

    {

        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "HR", "1234");

        PreparedStatement ps = con.prepareStatement(query);

        ResultSet rs = ps.executeQuery();

        return rs;

    }

}

**Home Page:**

****

**Code:**

private void adminActionPerformed(java.awt.event.ActionEvent evt) {

    new adminlogin().setVisible(true);

    dispose();

}

private void receptionistActionPerformed(java.awt.event.ActionEvent evt) {

    new receptionistLoginForm().setVisible(true);

    dispose();

}

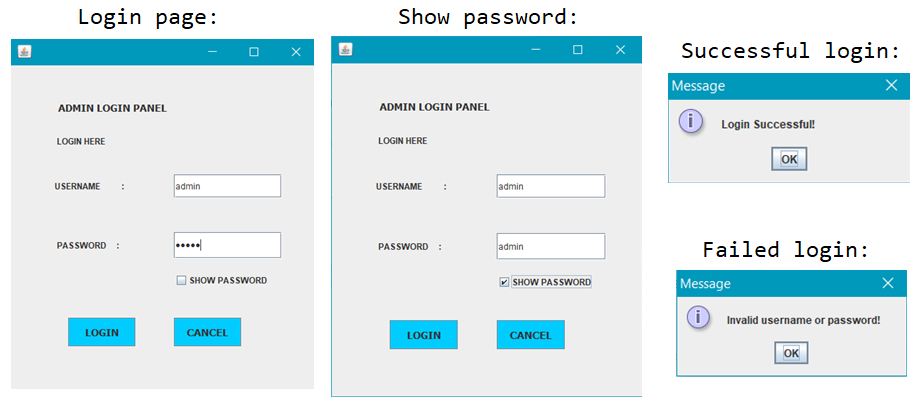
private void doctorActionPerformed(java.awt.event.ActionEvent evt) {

    new doctorlogin().setVisible(true);

    dispose();

}

**Admin Login Page:**

****

**Code:**

**Login Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    String uname = username.getText();

    String pwd = password.getText();

    if(username.getText().trim().isEmpty() || password.getText().trim().isEmpty())

        JOptionPane.showMessageDialog(this, "Please fill all the credentials.");

    else

{

        try

{

            String query = "select \* from admin";

            ResultSet rs;

            rs = Oracle.execute(query);

            if(rs.next())

            {

                String user = rs.getString("username");

                String pass = rs.getString("password");

                if(user.equals(uname) && pass.equals(pwd))

                {

                    JOptionPane.showMessageDialog(this, "Login Successful!");

                    new adminportal().setVisible(true);

                    this.dispose();

                }

                else

                    JOptionPane.showMessageDialog(this, "Invalid username or password!");

            }

            else

                JOptionPane.showMessageDialog(this, "Empty ResultSet!");

        } catch (SQLException ex) {

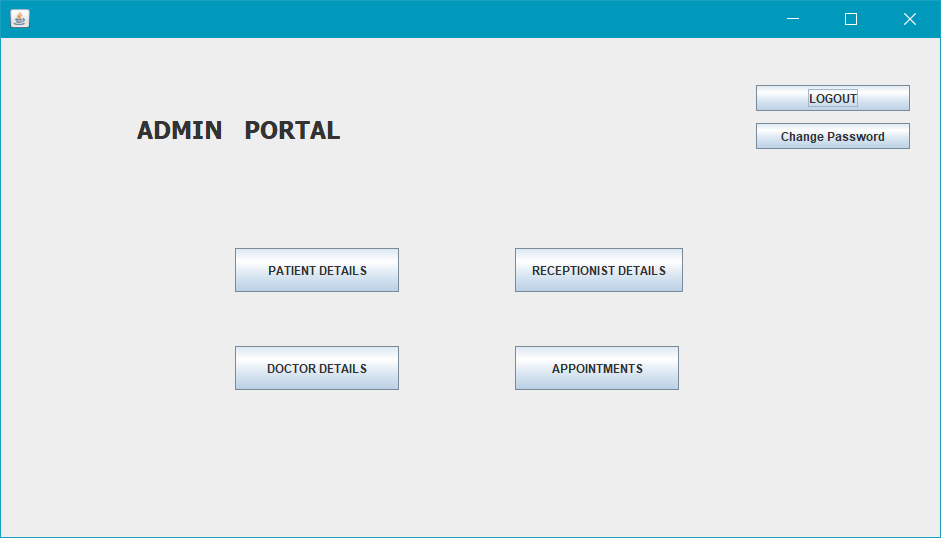
            Logger.getLogger(adminlogin.class.getName()).log(Level.SEVERE, null, ex);

        }

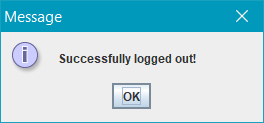
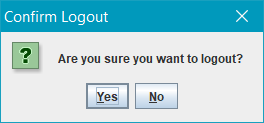
    }

}

**Admin Portal:**



Logout Button: After clicking yes:



**Code:**

private void appointmentsButtonActionPerformed(java.awt.event.ActionEvent evt) {

    new adminAppointDetails().setVisible(true);

    this.dispose();

}

private void logoutButtonActionPerformed(java.awt.event.ActionEvent evt) {

    int option = JOptionPane.showConfirmDialog(this, "Are you sure you want to logout?", "Confirm Logout", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

    if(option == JOptionPane.YES\_OPTION)

    {

        JOptionPane.showMessageDialog(this, "Successfully logged out!");

        this.dispose();

        new mainframe().setVisible(true);

    }

}

private void patDetButtonActionPerformed(java.awt.event.ActionEvent evt) {

    new adminPatientDetails().setVisible(true);

    this.dispose();

}

private void recButtonActionPerformed(java.awt.event.ActionEvent evt) {

    new adminRecepDetails().setVisible(true);

    this.dispose();

}

private void docDetButtonActionPerformed(java.awt.event.ActionEvent evt) {

    new adminDoctorDetails().setVisible(true);

    this.dispose();

}

private void changePasswordButtonActionPerformed(java.awt.event.ActionEvent evt) {

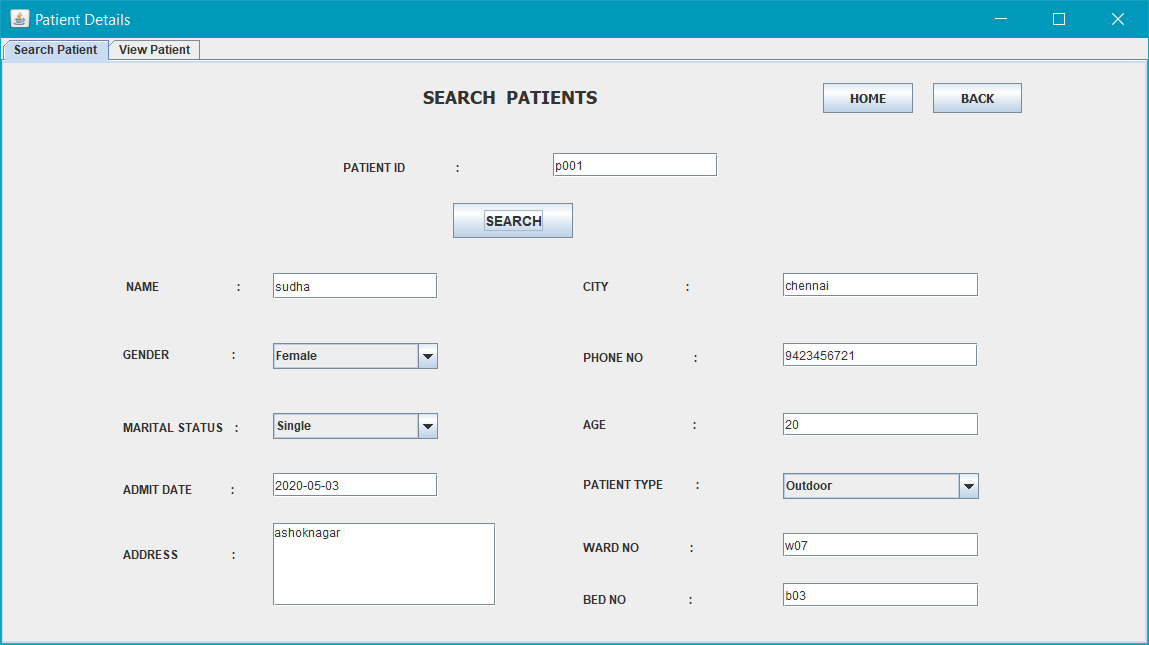
    new changePasswordAdmin().setVisible(true);

}

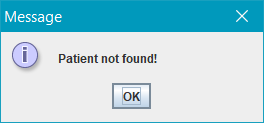
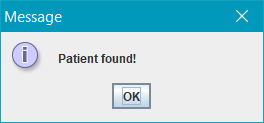
**Patient Details Window:**

**Search Patients:**

After entering the patient id and clicking search, the details of the patient are filled in the respective fields.



On successful search: On failed search:



**Code:**

**Search Button:**

private void searchButtonActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        ResultSet rs = Oracle.execute("select \* from patient where patientid = '"+pIDText.getText()+"'");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Patient found!");

            nameText.setText(rs.getString("pName"));

            int index = rs.getString("gender").equals("Male")? 0: 1;

            genderCombo.setSelectedIndex(index);

            index = rs.getString("maritalstatus").equals("Single")? 0: 1;

            maritalCombo.setSelectedIndex(index);

            dateText.setText(rs.getString("admitdate").substring(0,10));

            addText.setText(rs.getString("address"));

            cityText.setText(rs.getString("city"));

            phNoText.setText(rs.getString("phno"));

            ageText.setText(rs.getString("age"));

            index = rs.getString("pttype").equals("in patient")? 0: 1;

            patTypeCombo.setSelectedIndex(index);

            wardNoText.setText(rs.getString("wardno"));

            bedNoText.setText(rs.getString("bedno"));

        }

        else

            JOptionPane.showMessageDialog(this, "Patient not found!");

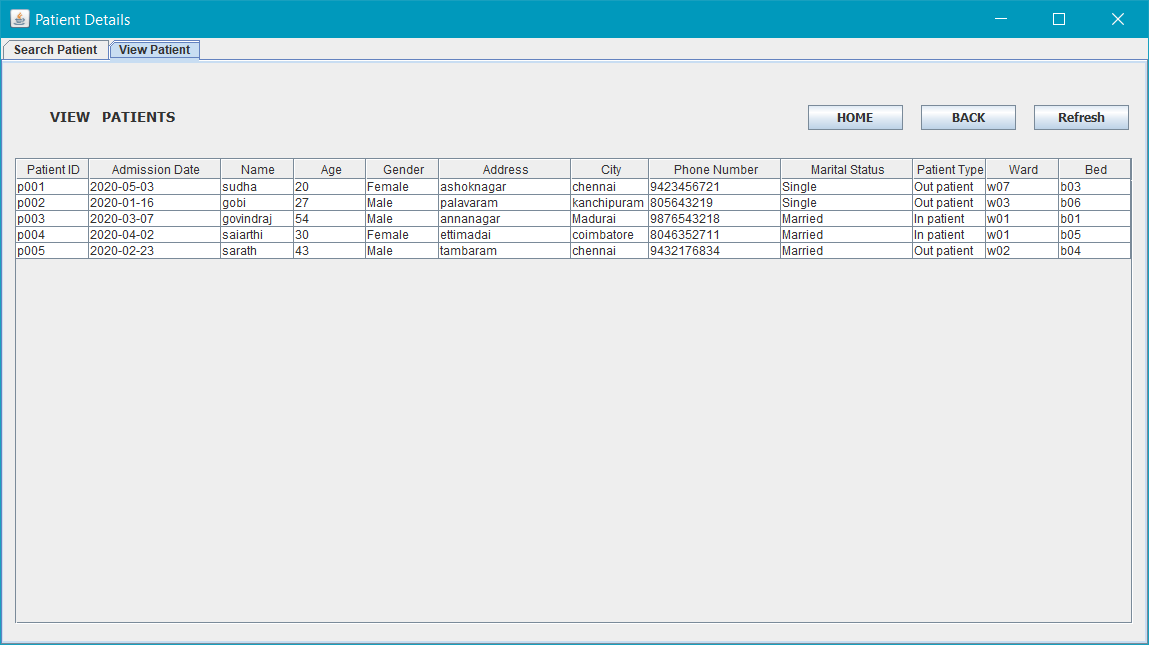
    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Patients:**



**Code:**

The code for Viewing the Database table is written in a separate function so that it can be called whenever the user wants to refresh its contents.

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select \* from patient");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String patientid, admitdate, pname, age, gender, address, city, phno, maritalstatus, pttype, wardno, bedno;

            patientid = rs.getString(1);

            admitdate = rs.getString(2).substring(0, 10);

            pname = rs.getString(3);

            age = rs.getString(4);

            gender = rs.getString(5);

            address = rs.getString(6);

            city = rs.getString(7);

            phno = rs.getString(8);

            maritalstatus = rs.getString(9);

            pttype = rs.getString(10);

            wardno = rs.getString(11);

            bedno = rs.getString(12);

            model.addRow(new Object[]{patientid, admitdate, pname, age, gender, address, city, phno, maritalstatus, pttype, wardno, bedno});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

public adminPatientDetails() {

    initComponents();

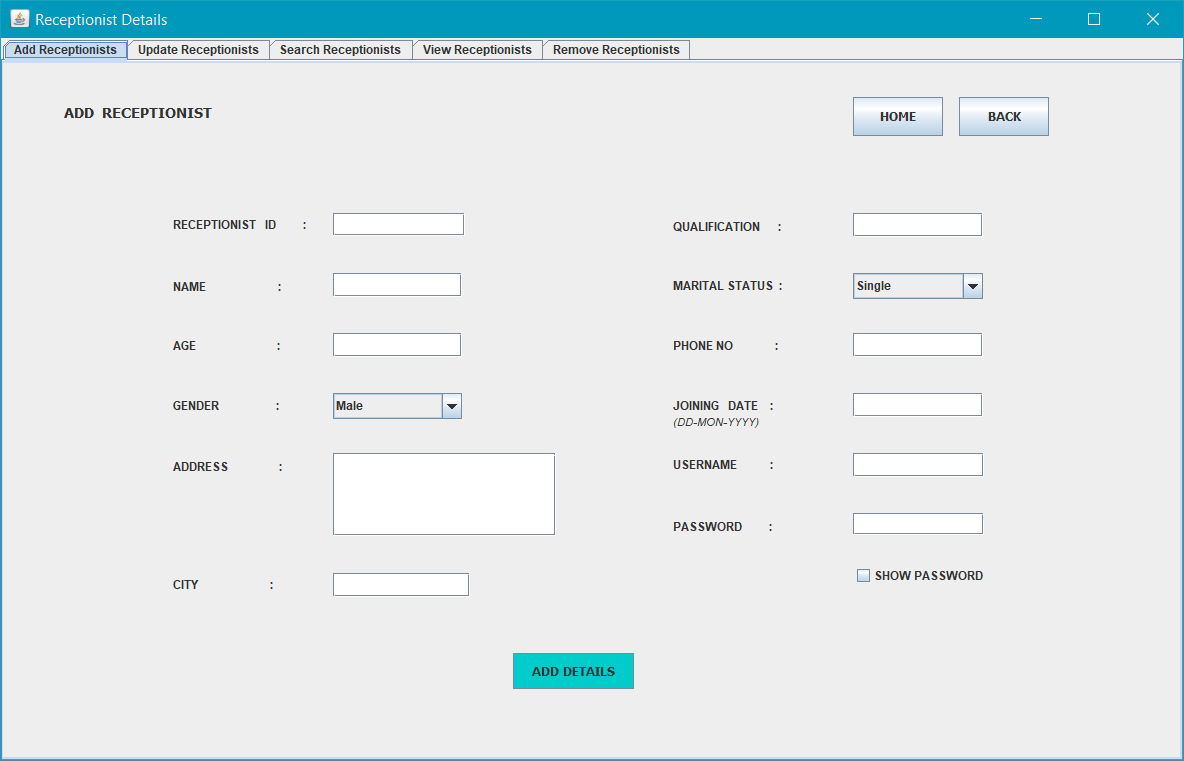
    viewRefresh();

}

ViewRefresh() function is called in the constructor of the frame so that the table is filled with the database values upon opening the frame.

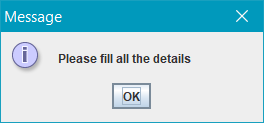
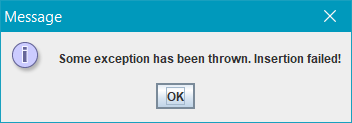
**Receptionist Details Window:**

**Add Receptionists:**



When some fields are empty: When some error occurs in the

query execution:

**Code:**

private void addRecepButtonActionPerformed(java.awt.event.ActionEvent evt) {

    id = recepIDText.getText().trim(); name = recepNameText.getText().trim();

    gender = recepGenderCombo.getSelectedItem().toString();

    address = recepAddressText.getText().trim();

    city = recepCityText.getText().trim(); phno = recepPhNoText.getText().trim();

    maritalstatus = recepMaritalCombo.getSelectedItem().toString();

    qualification = recepQualText.getText().trim();

    joindate = recepJoinDateText.getText().trim();

    username = recepUserNameText.getText().trim();

    password = recepPassText.getText().trim();

    try{

        age = Integer.parseInt(recepAgeText.getText().trim());

        if(id.isEmpty()||name.isEmpty()||address.isEmpty()||city.isEmpty()||qualification.isEmpty()||joindate.isEmpty()||username.isEmpty()||password.isEmpty())

            JOptionPane.showMessageDialog(this, "Please fill all the details");

        else{

            try {

                ResultSet rs = Oracle.execute("insert into receptionist values ('"+id+"','"+name+"','"+age+"','"+gender+"','"+address+"','"+city+"','"+phno+"','"+maritalstatus+"','"+qualification+"','"+joindate+"','"+username+"','"+password+"')");

                JOptionPane.showMessageDialog(this, "Added Sucessfully");

            } catch (SQLException ex) {

                JOptionPane.showMessageDialog(this, "Some exception has been thrown. Insertion failed!");

                Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    } catch (NumberFormatException e) {

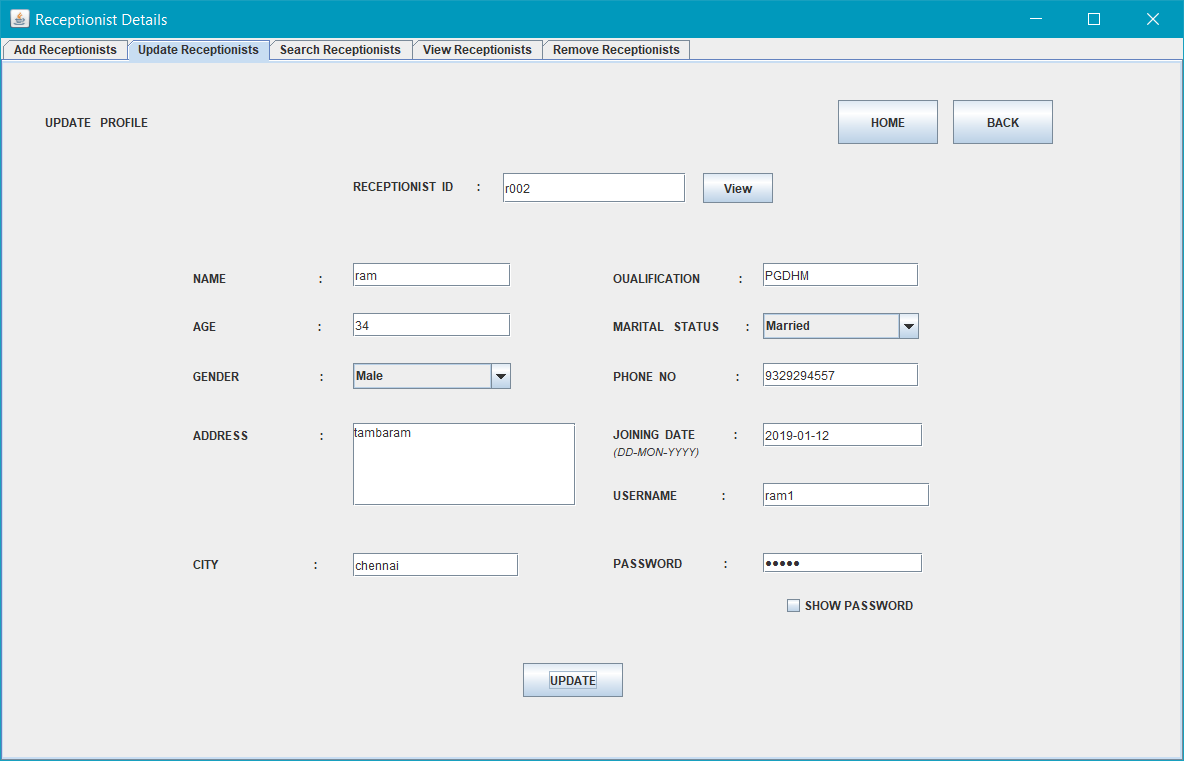
        JOptionPane.showMessageDialog(this, "Please fill all the details");

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, e);

    }

}

**Update Receptionists:**



**Code:**

private void recepUpdateButtonActionPerformed(java.awt.event.ActionEvent evt) {

    int response=JOptionPane.showConfirmDialog(this, "Sure You Want To Update?", "Confirm Update", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

    if(response==JOptionPane.YES\_OPTION){

        id = updateID.getText().trim();

        name = updateName.getText().trim();

        gender = updateGender.getSelectedItem().toString();

        address = updateAddress.getText().trim();

        city = updateCity.getText().trim();

        phno = updatePhNo.getText().trim();

        maritalstatus = updateMarital.getSelectedItem().toString();

        qualification = updateQual.getText().trim();

        joindate = updateJoinDate.getText().trim();

        username = updateUsername.getText().trim();

        password = updatePassword.getText().trim();

        try{

            age = Integer.parseInt(updateAge.getText().trim());

            if(id.isEmpty()||name.isEmpty()||address.isEmpty()||city.isEmpty()||qualification.isEmpty()||joindate.isEmpty()||username.isEmpty()||password.isEmpty())

               JOptionPane.showMessageDialog(this, "Please fill all the details");

            else{

                try {

                    ResultSet rs = Oracle.execute("update receptionist set rname='"+name+"',age='"+age+"',gender='"+gender+"',address='"+address+"',city='"+city+"',phno='"+phno+"',maritalstatus='"+maritalstatus+"',qualification='"+qualification+"',joindate='"+joindate+"',username='"+username+"',password='"+password+"' where receptionistid='"+id+"'");

                    JOptionPane.showMessageDialog(this, "Receptionist details updated sucessfully");

                } catch (SQLException ex) {

                    JOptionPane.showMessageDialog(this, "Some exception has been thrown. Updation failed!");

                    Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

        } catch (NumberFormatException e) {

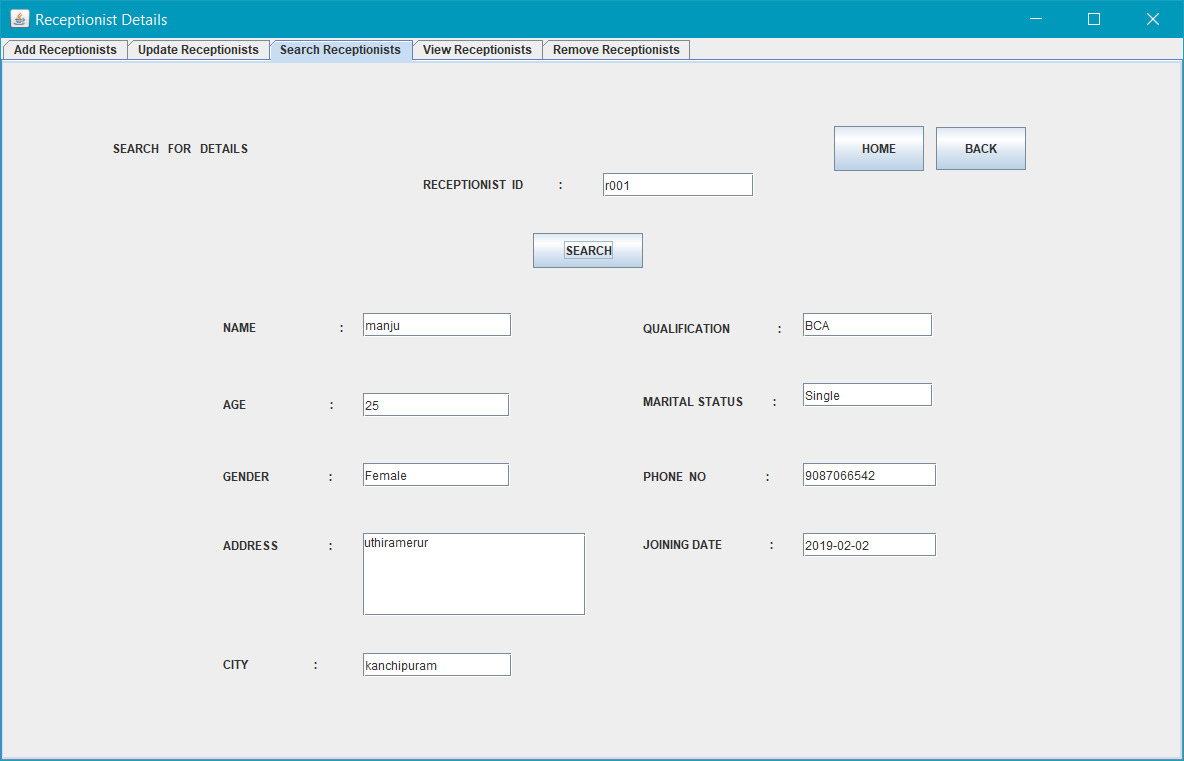
            JOptionPane.showMessageDialog(this, "Please fill all the details");

        }

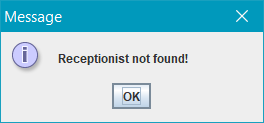
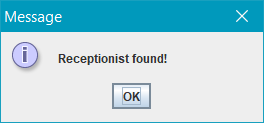
    }

}

**Search Receptionists:**



On successful search: On failed search:



**Code:**private void recepSearchButtonActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        ResultSet rs = Oracle.execute("select \* from receptionist where receptionistid = '"+searchRecepIDText.getText().trim()+"'");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Receptionist found!");

            searchRecepNameText.setText(rs.getString(2));

            searchRecepAgeText.setText(rs.getString(3));

            searchRecepGenderText.setText(rs.getString(4));

            searchRecepAddressText.setText(rs.getString(5));

            searchRecepCityText.setText(rs.getString(6));

            searchRecepPhNoText.setText(rs.getString(7));

            searchRecepMaritalText.setText(rs.getString(8));

            searchRecepQualText.setText(rs.getString(9));

            searchRecepJoinDateText.setText(rs.getString(10).substring(0,10));

        }else{

            JOptionPane.showMessageDialog(this, "Receptionist not found!");

        }

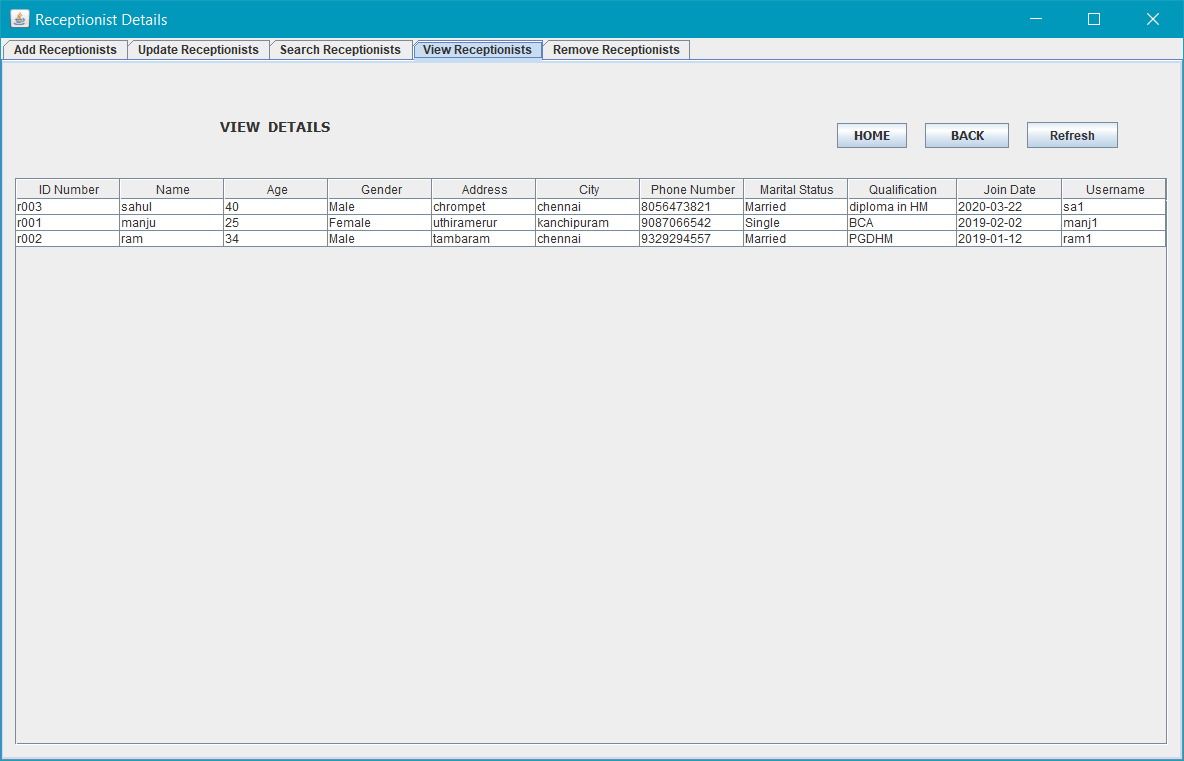
    } catch (SQLException ex) {

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Receptionists:**



**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select \* from receptionist");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next()) {

            String receptionistid, rname, age, gender, address, city, phno, maritalstatus, qualification, joindate, username;

            receptionistid = rs.getString(1);

            rname = rs.getString(2);

            age = rs.getString(3);

            gender = rs.getString(4);

            address = rs.getString(5);

            city = rs.getString(6);

            phno = rs.getString(7);

            maritalstatus = rs.getString(8);

            qualification = rs.getString(9);

            joindate = rs.getString(10).substring(0, 10);

            username = rs.getString(11);

            model.addRow(new Object[]{receptionistid, rname, age, gender, address, city, phno, maritalstatus, qualification, joindate, username});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

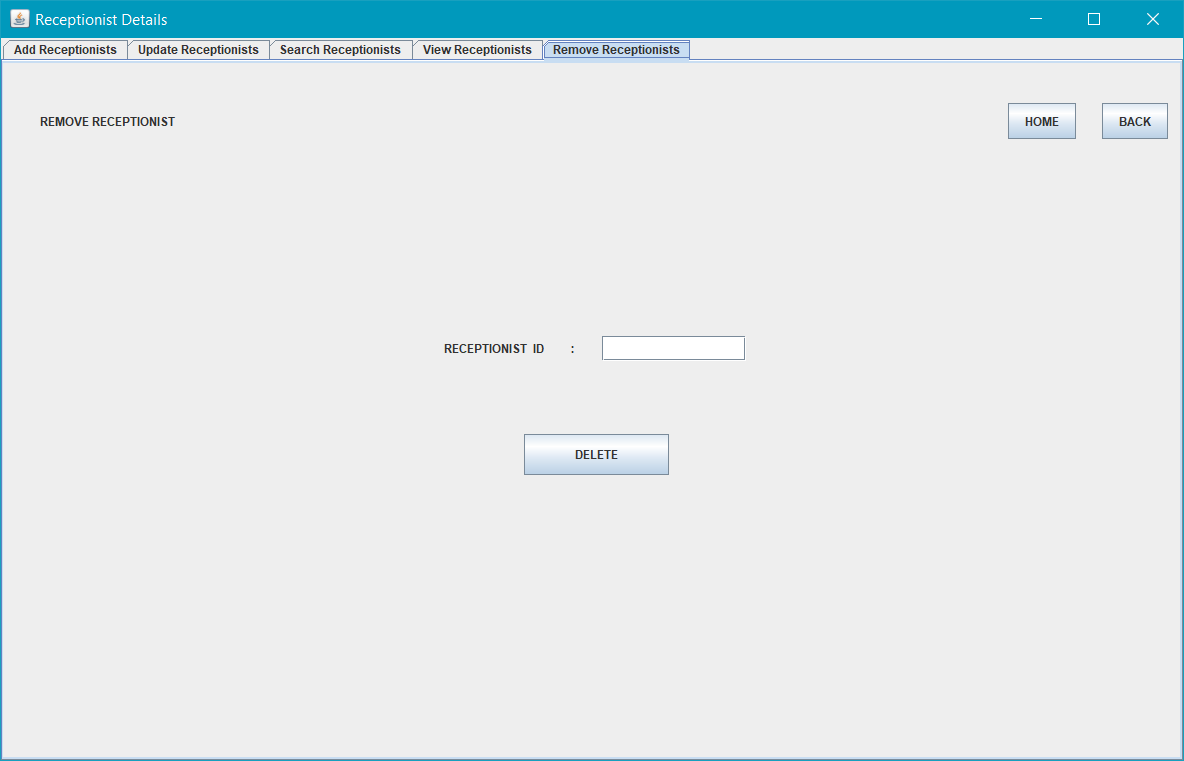
public adminRecepDetails() {

    initComponents();

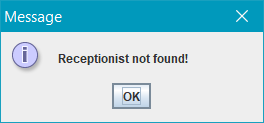
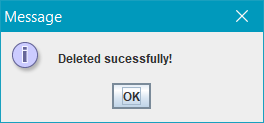
    viewRefresh();

}

**Remove Receptionists:**

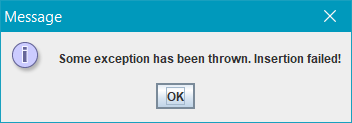


On successful deletion: Invalid Receptionist ID:



When some error occurs in the

query execution:



**Code:**

private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {

    int response=JOptionPane.showConfirmDialog(this, "Are you sure you want to delete?", "Confirm Delete", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

    if(response==JOptionPane.YES\_OPTION){

        try {

            ResultSet rs = Oracle.execute("select \* from receptionist where receptionistid = '"+deleteRecepIDText.getText().trim()+"'");

            if(rs.next())

            {

                Oracle.execute("delete from receptionist where receptionistid = '"+deleteRecepIDText.getText().trim()+"'");

                JOptionPane.showMessageDialog(null, "Deleted sucessfully!");

            } else

                JOptionPane.showMessageDialog(null, "Receptionist not found!");

        } catch (SQLException ex) {

            Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

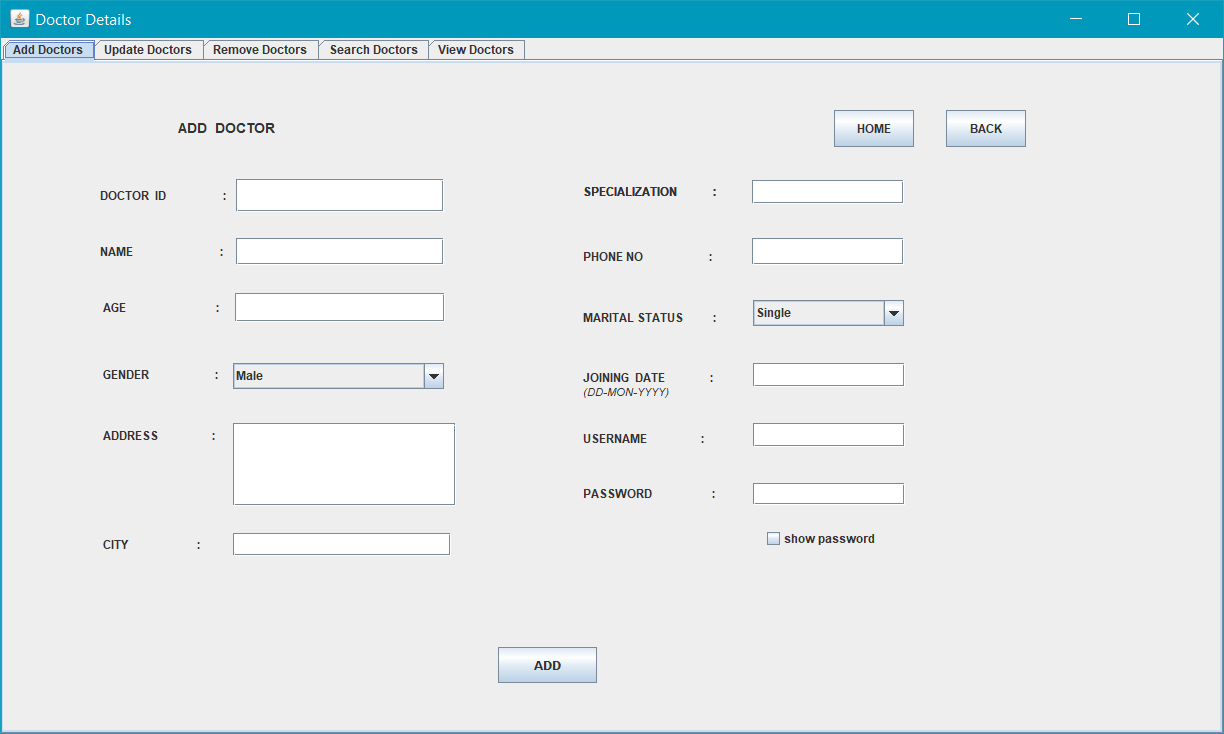
        }

    }

}

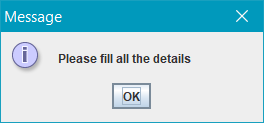
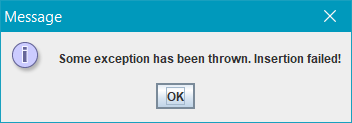
**Doctor Details Window:**

**Add Doctors:**



When some fields are empty: When some error occurs in the

query execution:

**Code:**

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

    id = doctorid.getText().trim();

    name = doctorname.getText().trim();

    gender = doctorgender.getSelectedItem().toString();

    address = doctoraddress.getText().trim();

    city = doctorcity.getText().trim();

    phno = doctorphone.getText().trim();

    maritalstatus = doctorms.getSelectedItem().toString();

    qualification = doctorspl1.getText().trim();

    specialization = doctorspl.getText().trim();

    joindate = doctorjd.getText().trim();

    username = doctorusername.getText().trim();

    password = doctorpass.getText().trim();

    try{

        age = Integer.parseInt(doctorage.getText().trim());

        if(id.isEmpty()||name.isEmpty()||address.isEmpty()||city.isEmpty()||qualification.isEmpty()||specialization.isEmpty()||joindate.isEmpty()||username.isEmpty()||password.isEmpty())

            JOptionPane.showMessageDialog(this, "Please fill all the details");

        else{

            try {

                ResultSet rs = Oracle.execute("insert into doctor values ('"+id+"','"+name+"','"+age+"','"+gender+"','"+address+"','"+city+"','"+phno+"','"+maritalstatus+"','"+qualification+"','"+specialization+"','"+joindate+"','"+username+"','"+password+"')");

                JOptionPane.showMessageDialog(this, "Added Sucessfully");

            } catch (SQLException ex) {

                JOptionPane.showMessageDialog(this, "Some exception has been thrown. Insertion failed!");

                Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    } catch (NumberFormatException e) {

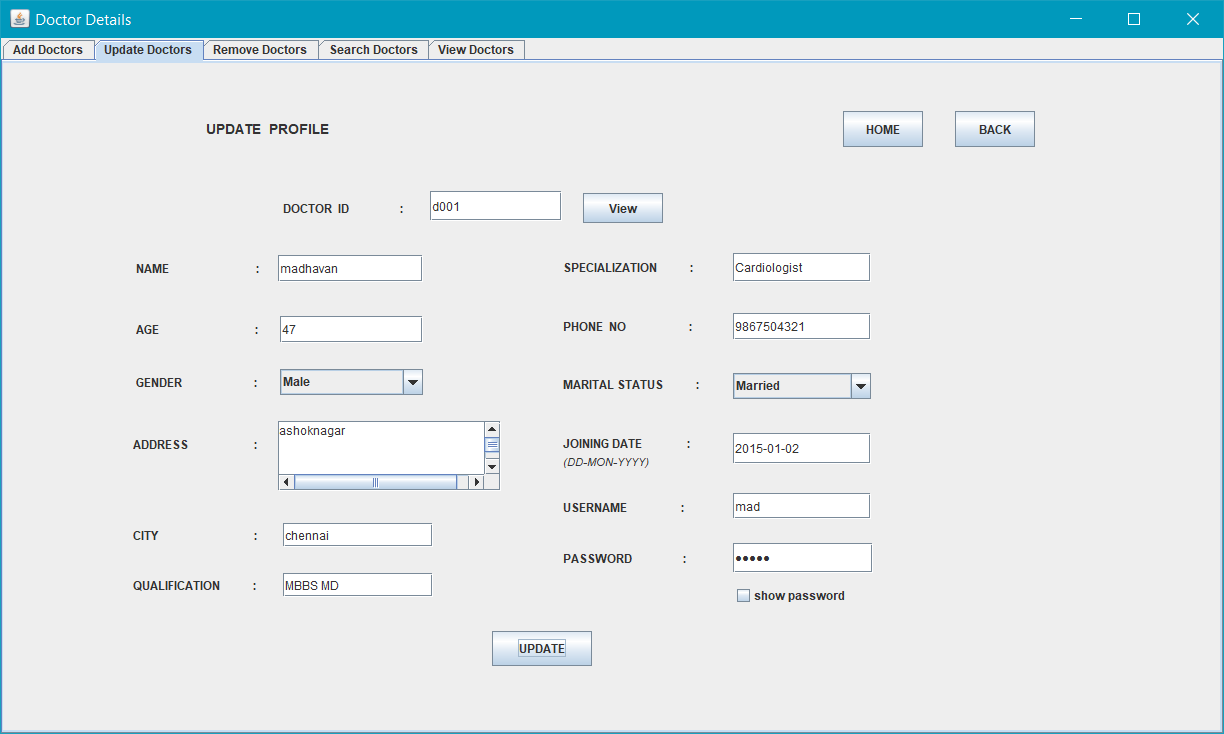
        JOptionPane.showMessageDialog(this, "Please fill all the details");

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, e);

    }

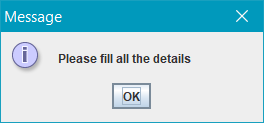
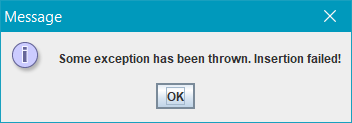
}

**Update Doctors:**



When some fields are empty: When some error occurs in the

query execution:

**Code:**

private void updateButtonActionPerformed(java.awt.event.ActionEvent evt) {

    int response=JOptionPane.showConfirmDialog(this, "Are you sure you want to update?", "Confirm Update", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

    if(response==JOptionPane.YES\_OPTION){

        id = updateID.getText().trim();

        name = updateName.getText().trim();

        gender = updateGender.getSelectedItem().toString();

        address = updateAddress.getText().trim();

        city = updateCity.getText().trim();

        phno = updatePhNo.getText().trim();

        maritalstatus = updateMarital.getSelectedItem().toString();

        qualification = updateQual.getText().trim();

        specialization = updateSpl.getText().trim();

        joindate = updateJoinDate.getText().trim();

        username = updateUsername.getText().trim();

        password = updatePassword.getText().trim();

        try{

            age = Integer.parseInt(updateAge.getText().trim());

            if(id.isEmpty()||name.isEmpty()||address.isEmpty()||city.isEmpty()||qualification.isEmpty()||specialization.isEmpty()||joindate.isEmpty()||username.isEmpty()||password.isEmpty()){

                JOptionPane.showMessageDialog(this, "Please fill all the details");

            }

            else{

                try {

                    ResultSet rs = Oracle.execute("update doctor set dname='"+name+"',age='"+age+"',gender='"+gender+"',address='"+address+"',city='"+city+"',phno='"+phno+"',maritalstatus='"+maritalstatus+"',qualification='"+qualification+"',specialization='"+specialization+"',joindate='"+joindate+"',username='"+username+"',password='"+password+"' where doctorid = '"+id+"'");

                    JOptionPane.showMessageDialog(this, "Doctor details updated sucessfully");

                } catch (SQLException ex) {

                    JOptionPane.showMessageDialog(this, "Some exception has been thrown. Updation failed!");

                    Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

        } catch (NumberFormatException e) {

            JOptionPane.showMessageDialog(this, "Please fill all the details");

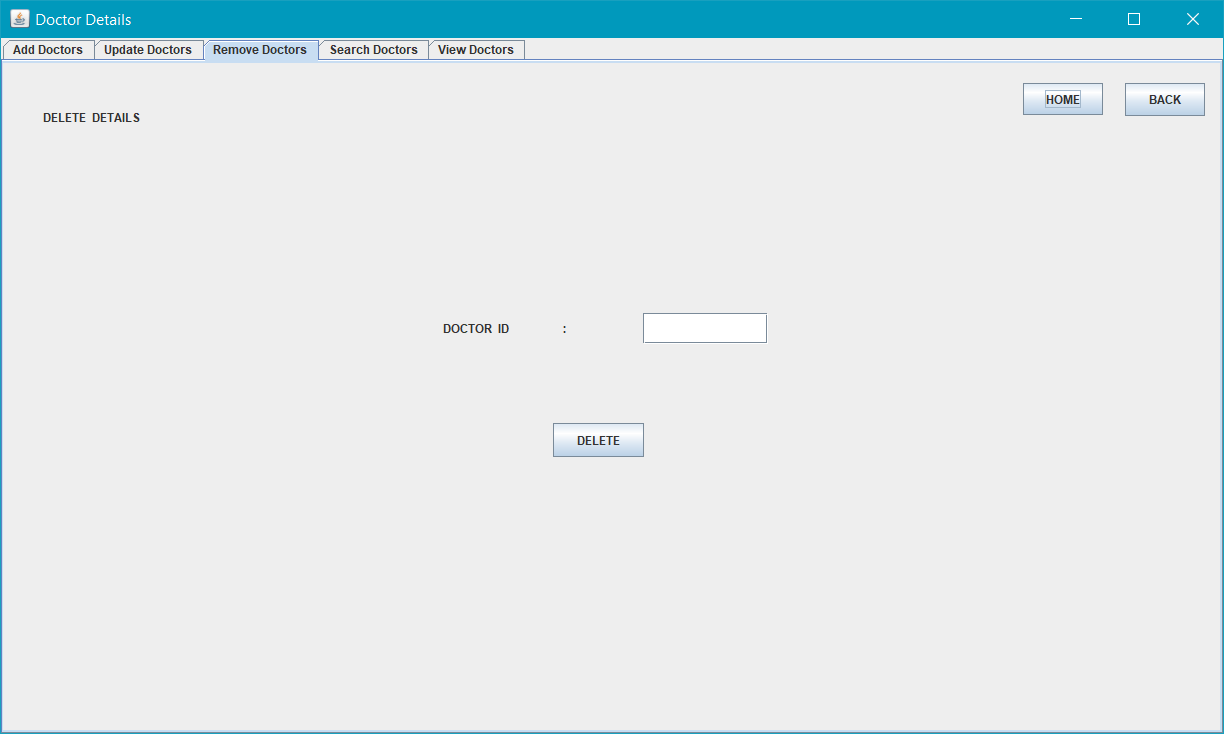
            Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, e);

        }

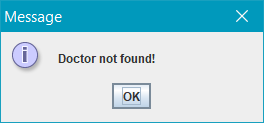
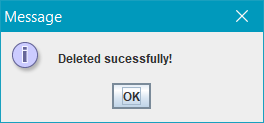
    }

}

**Remove Doctors:**



On successful deletion: Invalid Receptionist ID:



**Code:**

private void removeDoctorButtonActionPerformed(java.awt.event.ActionEvent evt) {

    int response=JOptionPane.showConfirmDialog(this, "Are you sure you want to delete?", "Confirm Delete", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

    if(response==JOptionPane.YES\_OPTION){

        try {

            ResultSet rs = Oracle.execute("select \* from doctor where doctorid = '"+deleteDoctorIDText.getText().trim()+"'");

            if(rs.next())

            {

                Oracle.execute("delete from doctor where doctorid = '"+deleteDoctorIDText.getText().trim()+"'");

                JOptionPane.showMessageDialog(null, "Deleted sucessfully!");

            } else {

                JOptionPane.showMessageDialog(null, "Doctor not found!");

            }

        } catch (SQLException ex) {

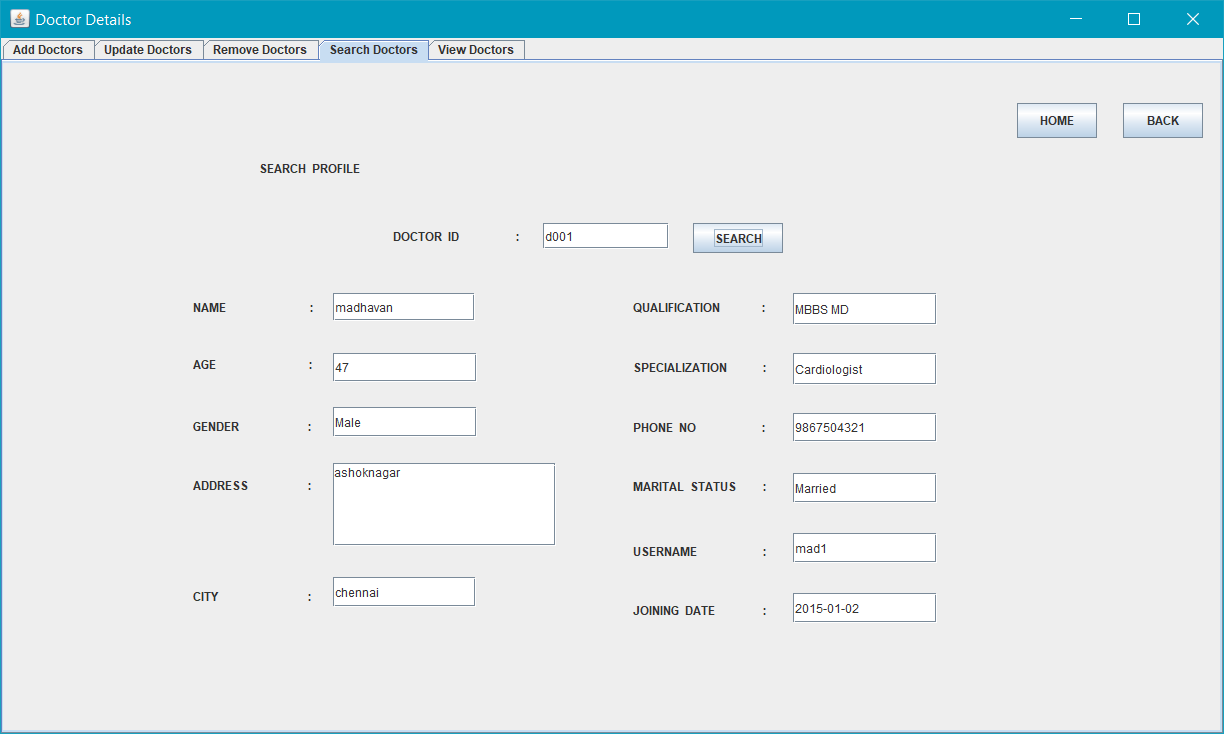
            Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

        }

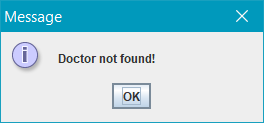
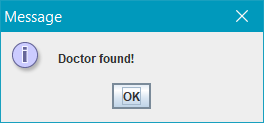
    }

}

**Search Doctors:**



On successful Search: On failed search:



**Code:**

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        ResultSet rs = Oracle.execute("select \* from doctor where doctorid = '"+searchID.getText().trim()+"'");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Doctor found!");

            searchName.setText(rs.getString(2));

            searchAge.setText(rs.getString(3));

            searchGender.setText(rs.getString(4));

            searchAddress.setText(rs.getString(5));

            searchCity.setText(rs.getString(6));

            searchPhNo.setText(rs.getString(7));

            searchMarital.setText(rs.getString(8));;

            searchQual.setText(rs.getString(9));

            searchSpl.setText(rs.getString(10));

            searchJoinDate.setText(rs.getString(11).substring(0,10));

            searchUsername.setText(rs.getString(12));

        }else

            JOptionPane.showMessageDialog(this, "Doctor not found!");

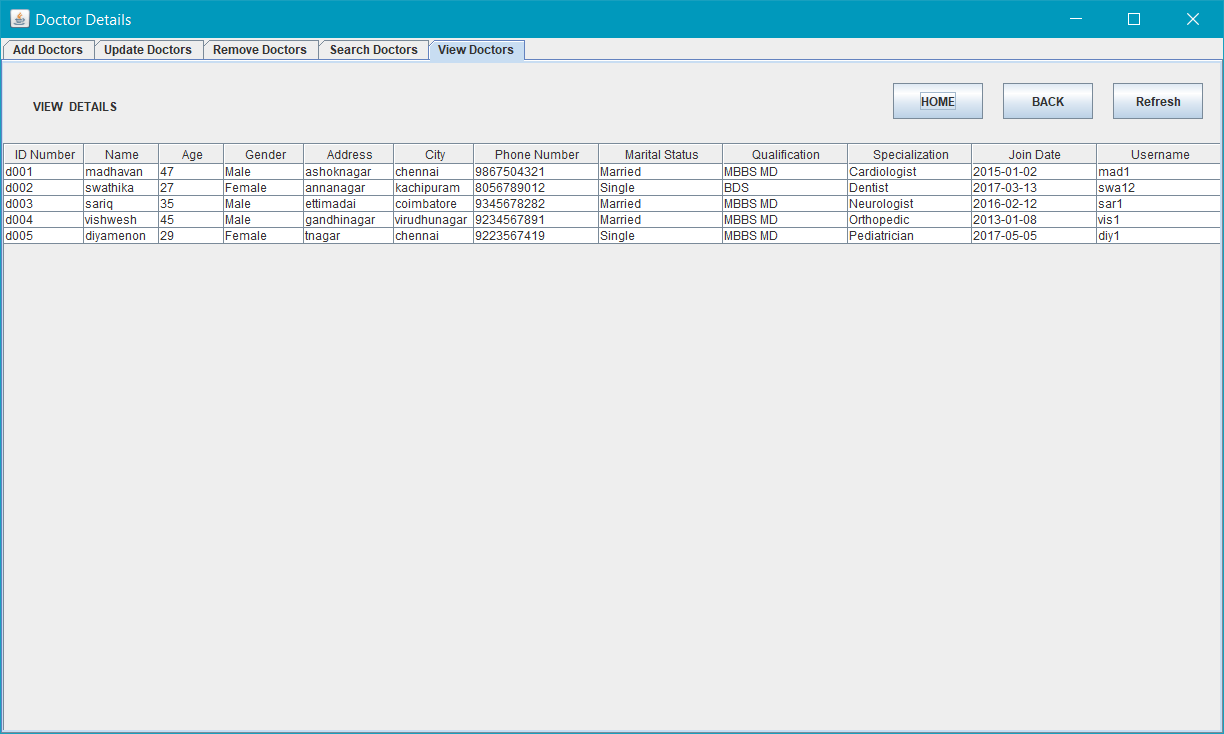
    } catch (SQLException ex) {

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Doctors:**



**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select \* from doctor");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String doctorid, dname, age, gender, address, city, phno, maritalstatus, qualification, specialization, joindate, username;

            doctorid = rs.getString(1);

            dname = rs.getString(2);

            age = rs.getString(3);

            gender = rs.getString(4);

            address = rs.getString(5);

            city = rs.getString(6);

            phno = rs.getString(7);

            maritalstatus = rs.getString(8);

            qualification = rs.getString(9);

            specialization = rs.getString(10);

            joindate = rs.getString(11).substring(0, 10);

            username = rs.getString(12);

            model.addRow(new Object[]{doctorid, dname, age, gender, address, city, phno, maritalstatus, qualification, specialization, joindate, username});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

public adminDoctorDetails() {

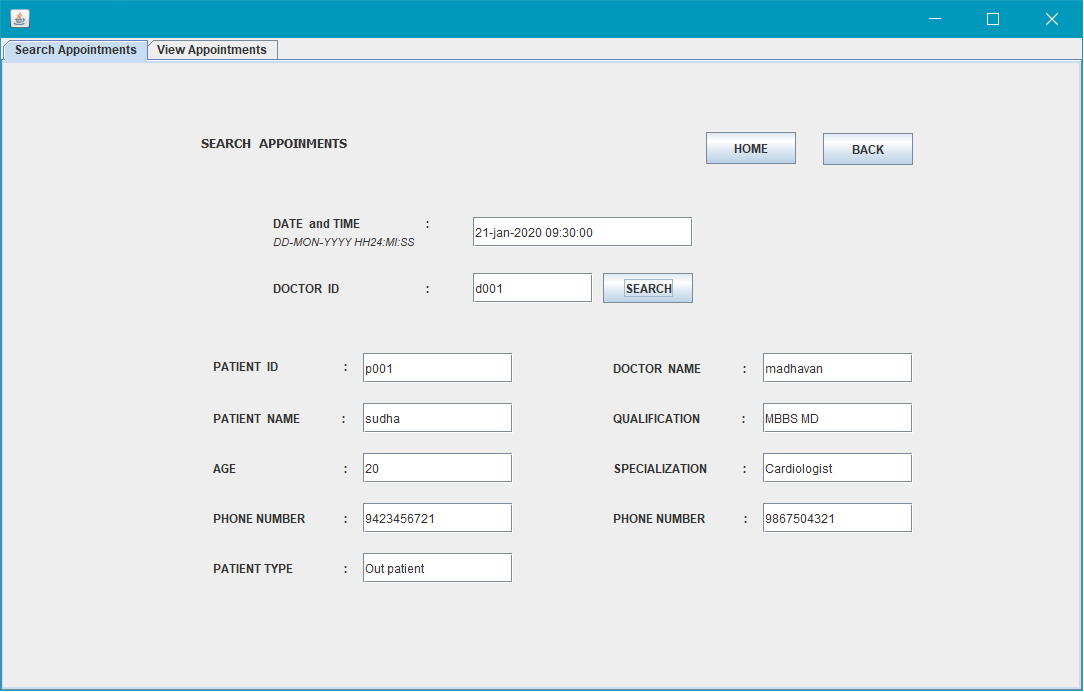
    initComponents();

    viewRefresh();

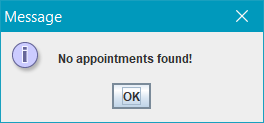
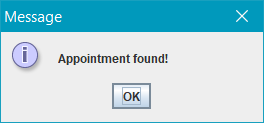
}

**Appointment Details Window:**

**Search Appointments:**

****

On successful Search: On failed search:



**Code:**

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        ResultSet rs = Oracle.execute("select \* from doctor where doctorid = '"+searchID.getText().trim()+"'");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Doctor found!");

            searchName.setText(rs.getString(2));

            searchAge.setText(rs.getString(3));

            searchGender.setText(rs.getString(4));

            searchAddress.setText(rs.getString(5));

            searchCity.setText(rs.getString(6));

            searchPhNo.setText(rs.getString(7));

            searchMarital.setText(rs.getString(8));;

            searchQual.setText(rs.getString(9));

            searchSpl.setText(rs.getString(10));

            searchJoinDate.setText(rs.getString(11).substring(0,10));

            searchUsername.setText(rs.getString(12));

        }else

            JOptionPane.showMessageDialog(this, "Doctor not found!");

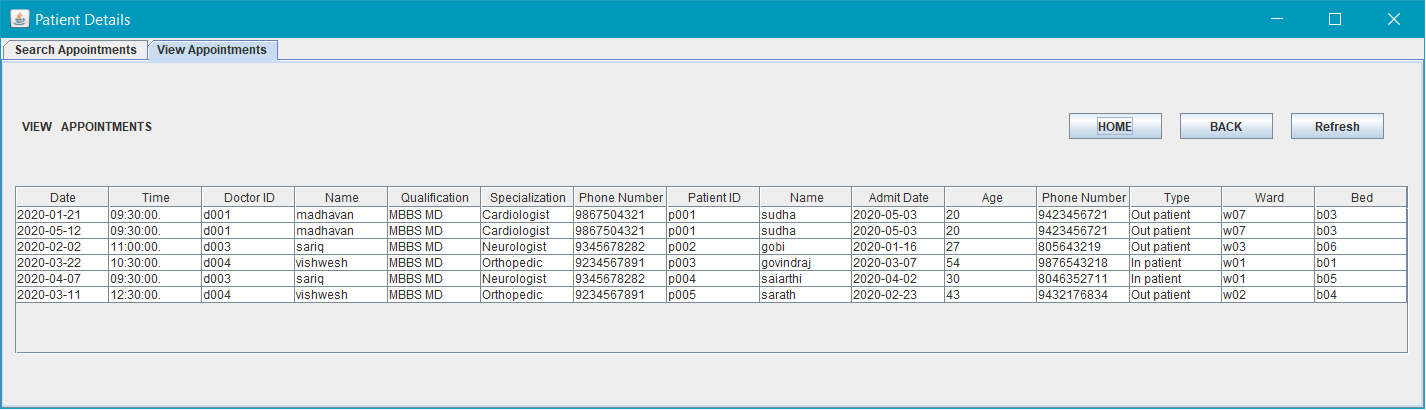
    } catch (SQLException ex) {

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Appointments:**

****

**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select \* from doctor");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String doctorid, dname, age, gender, address, city, phno, maritalstatus, qualification, specialization, joindate, username;

            doctorid = rs.getString(1);

            dname = rs.getString(2);

            age = rs.getString(3);

            gender = rs.getString(4);

            address = rs.getString(5);

            city = rs.getString(6);

            phno = rs.getString(7);

            maritalstatus = rs.getString(8);

            qualification = rs.getString(9);

            specialization = rs.getString(10);

            joindate = rs.getString(11).substring(0, 10);

            username = rs.getString(12);

            model.addRow(new Object[]{doctorid, dname, age, gender, address, city, phno, maritalstatus, qualification, specialization, joindate, username});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

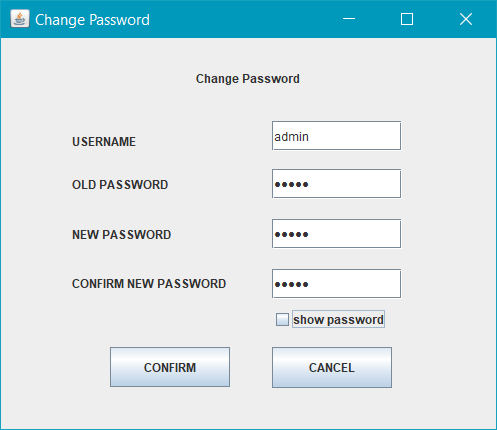
public adminDoctorDetails() {

    initComponents();

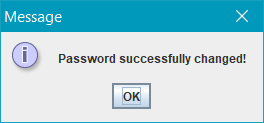
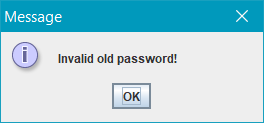
    viewRefresh();

}

**Change Password Window:**



Incorrect old password: On successful password change:



**Code:**

private void confirmButtonActionPerformed(java.awt.event.ActionEvent evt) {

    if(usernameText.getText().trim().isEmpty() || oldPass.getText().trim().isEmpty() || newPass.getText().trim().isEmpty() || confNewPass.getText().trim().isEmpty())

        JOptionPane.showMessageDialog(null,"Please fill all the credentials.");

    else {

        int option=JOptionPane.showConfirmDialog(this,"Are you sure you want to change your password?","Confirm",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

        if(option==JOptionPane.YES\_OPTION) {

            if(newPass.getText().equals(confNewPass.getText())) {

                try {

                    String query = "select \* from admin where username = '"+usernameText.getText()+"'";

                    ResultSet rs = Oracle.execute(query);

                    if(rs.next()) {

                        if(oldPass.getText().equals(rs.getString("password"))) {

                            query = "update admin set password = '"+newPass.getText()+"' where username = '"+usernameText.getText()+"'";

                            Oracle.execute(query);

                            JOptionPane.showMessageDialog(null,"Password successfully changed!");

                            this.dispose();

                        }

                        else

                            JOptionPane.showMessageDialog(this, "Invalid old password!");

                    }

                    else

                        JOptionPane.showMessageDialog(this, "Invalid Username");

                } catch (SQLException ex) {

                    Logger.getLogger(changePasswordAdmin.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

            else

                JOptionPane.showMessageDialog(this, "New password and Confirm new password do not match");

        }

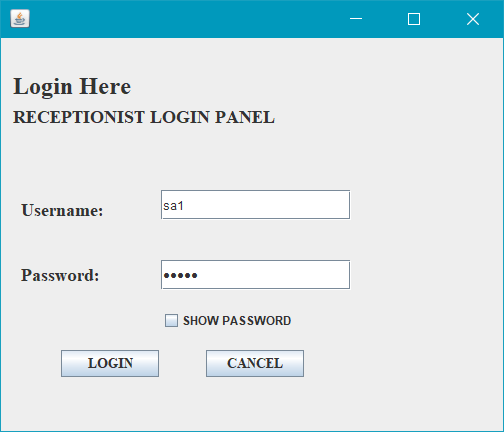
        else

            this.dispose();

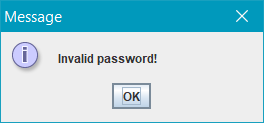
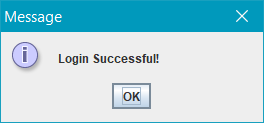
    }

}

**Receptionist Login Page:**



Incorrect password: On successful login:

**Code:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    String password=jpassword.getText();

    String username=jusername.getText();

    if(jusername.getText().trim().isEmpty() || jpassword.getText().trim().isEmpty())

        JOptionPane.showMessageDialog(this,"Please fill all the credentials.");

    else{

        try

        {

            String query = "select \* from receptionist where username = '"+username+"'";

            ResultSet rs;

            rs = Oracle.execute(query);

            if(rs.next())

            {

                String user = rs.getString("username");

                String pass = rs.getString("password");

                if(user.equals(username) && pass.equals(password))

                {

                    JOptionPane.showMessageDialog(this, "Login Successful!");

                    /\*\*

                     \* The username is also passed to the receptionistPortal to

                     \* know which receptionist is currently logged into the system

                     \*/

                    new receptionistPortal(username).setVisible(true);

                    this.dispose();

                }

                else

                    JOptionPane.showMessageDialog(this, "Invalid password!");

            }

            else

                JOptionPane.showMessageDialog(this, "Invalid Username!");

        } catch (SQLException ex) {

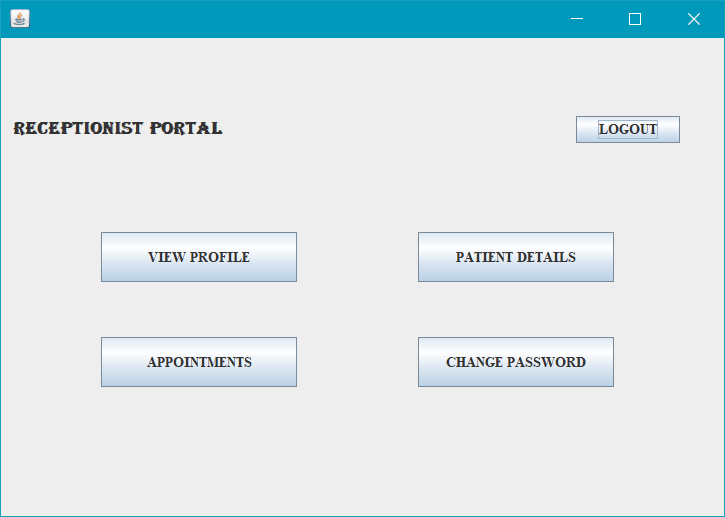
            Logger.getLogger(adminlogin.class.getName()).log(Level.SEVERE, null, ex);

        }

    }

}

**Receptionist Portal:**



**Code:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    new receprionistAppointments().setVisible(true);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    new receptionistProfile(username).setVisible(true);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    new recepPatientDetails().setVisible(true);

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    new changePasswordReceptionist().setVisible(true);

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    int response = JOptionPane.showConfirmDialog(this, "Are you sure you want to log out?", "Confirm Logout", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

    if(response == JOptionPane.YES\_OPTION) {

        this.dispose();

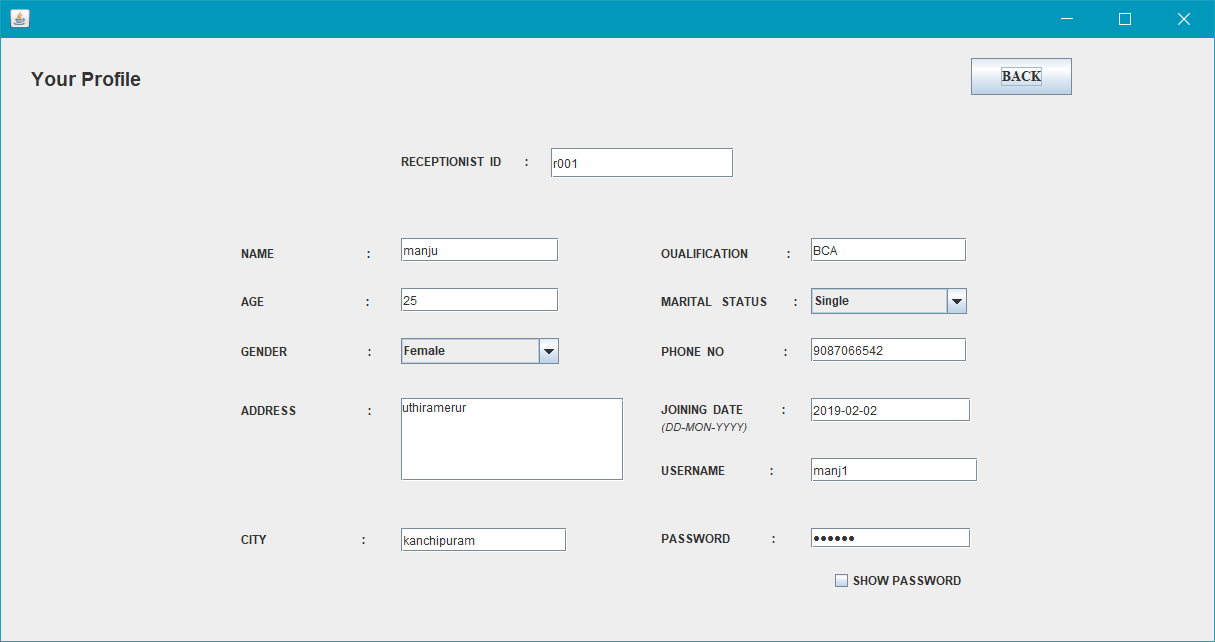
        new mainframe().setVisible(true);

    }

}

**View Profile Window:**

Since we have passed the receptionist’s username to the receptionistPortal, we can restrict the receptionist to view only their own profile.



**Code:**

public receptionistProfile(String username) {

    initComponents();

c = updatePassword.getEchoChar();

    this.username = username;

    try {

        ResultSet rs;

        rs = Oracle.execute("select receptionistid from receptionist where username = '"+username+"'");

        if(rs.next())

            updateID.setText(rs.getString(1));

        rs = Oracle.execute("select \* from receptionist where receptionistid = '"+updateID.getText().trim()+"'");

        if(rs.next())

        {

            updateName.setText(rs.getString(2));

            updateAge.setText(rs.getString(3));

            int index = rs.getString(4).equals("Male")?0:1;

            updateGender.setSelectedIndex(index);

            updateAddress.setText(rs.getString(5));

            updateCity.setText(rs.getString(6));

            updatePhNo.setText(rs.getString(7));

            index = rs.getString(8).equals("Single")?0:1;

            updateMarital.setSelectedIndex(index);

            updateQual.setText(rs.getString(9));

            updateJoinDate.setText(rs.getString(10).substring(0,10));

            updateUsername.setText(rs.getString(11));

            updatePassword.setText(rs.getString(12));

        }else

            JOptionPane.showMessageDialog(this, "Receptionist not found!");

    } catch (SQLException ex) {

        Logger.getLogger(receptionistProfile.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**Show Password Checkbox code:**

private void recepcheckbox1MouseClicked(java.awt.event.MouseEvent evt) {

    if(recepcheckbox1.isSelected())

        updatePassword.setEchoChar((char)0);

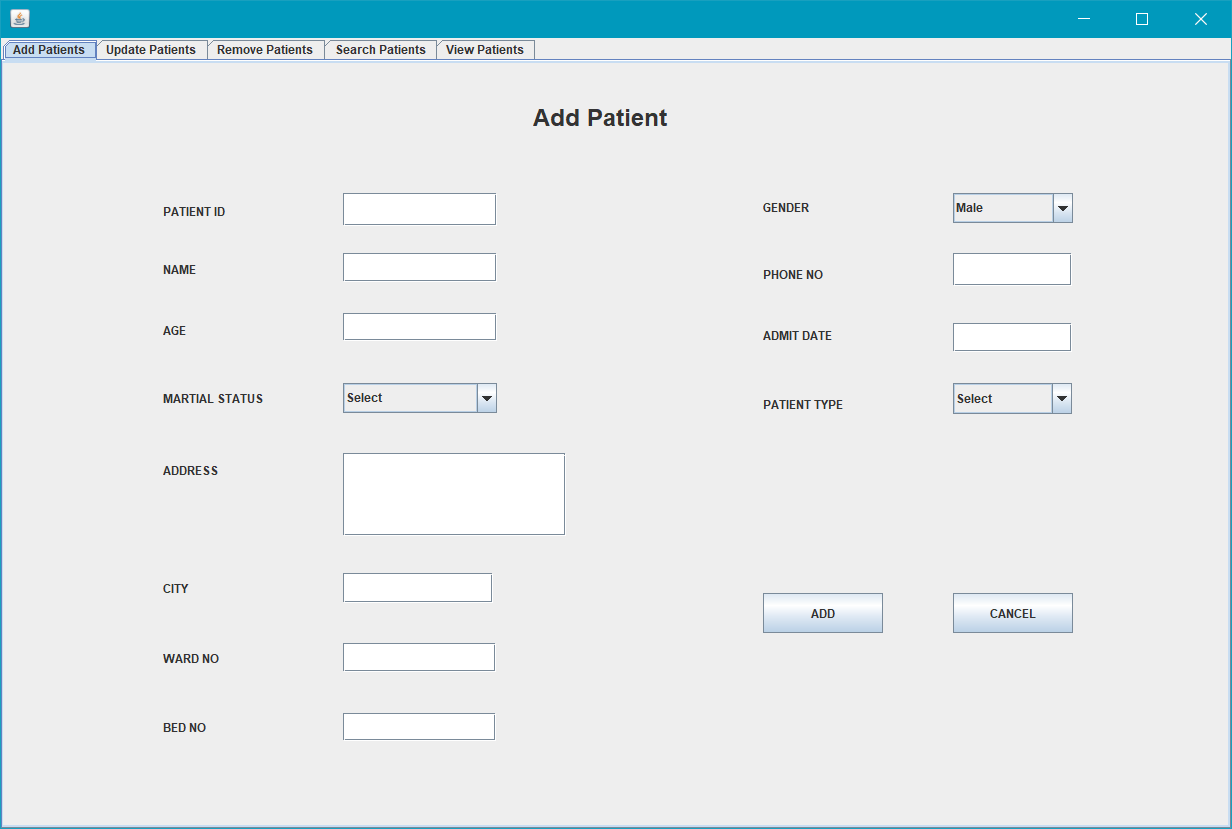
    else

        updatePassword.setEchoChar(c);

}

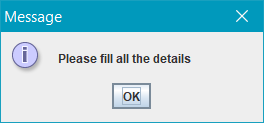
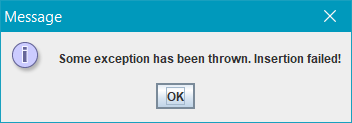
**Patient Details Window:**

**Add Patients:**



When some fields are empty: When some error occurs in the

query execution:

**Code:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    patientid = pid.getText().trim();

    admitdate = date.getText().trim();

    pname = name.getText().trim();

    gender = addGenderCombo.getSelectedItem().toString();

    address = add.getText().trim();

    city = addCityText.getText().trim();

    phno = pno.getText().trim();

    maritalstatus = marital.getSelectedItem().toString();

    pttype = ptype.getSelectedItem().toString();

    wardno = wno.getText().trim();

    bedno = bno.getText().trim();

    try{

        age = Integer.parseInt(addAgeText.getText().trim());

        if(patientid.isEmpty()||pname.isEmpty()||address.isEmpty()||city.isEmpty()||admitdate.isEmpty()||gender.isEmpty()||phno.isEmpty()||maritalstatus.isEmpty()||pttype.isEmpty()||wardno.isEmpty()||bedno.isEmpty())

            JOptionPane.showMessageDialog(this, "Please fill all the details");

        else{

            try {

                ResultSet rs = Oracle.execute("insert into patient values('"+patientid+"','"+admitdate+"','"+pname+"','"+age+"','"+gender+"','"+address+"','"+city+"','"+phno+"','"+maritalstatus+"','"+pttype+"','"+wardno+"','"+bedno+"')");

                JOptionPane.showMessageDialog(this, "Added Sucessfully");

            } catch (SQLException ex) {

                Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    } catch (NumberFormatException e) {

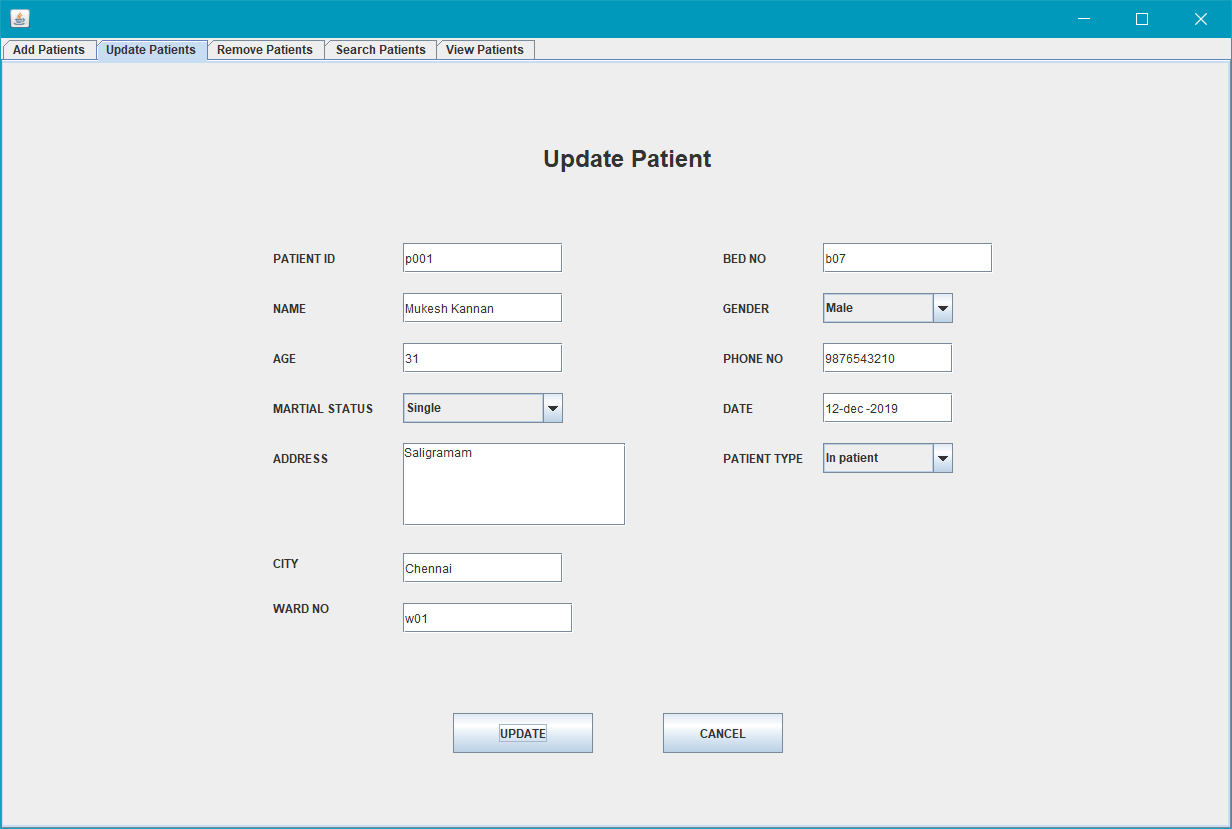
        JOptionPane.showMessageDialog(this, "Please fill all the details");

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, e);

    }

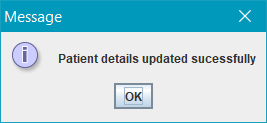
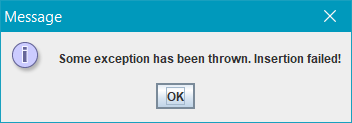
}

**Update Patients:**



On successful updation: When some error occurs in the

query execution:

**Code:**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    patientid = jpid.getText().trim();

    admitdate = jdate.getText().trim();

    pname = jname.getText().trim();

    gender = jgender.getSelectedItem().toString().trim();

    address = jadd.getText().trim();

    city = jcity.getText().trim();

    phno = jpno.getText().trim();

    maritalstatus = jmarital.getSelectedItem().toString().trim();

    pttype = jpttype.getSelectedItem().toString().trim();

    wardno = jwno.getText().trim();

    bedno = jbno.getText().trim();

    if(patientid.isEmpty()||admitdate.isEmpty()||pname.isEmpty()||gender.isEmpty()||address.isEmpty()||city.isEmpty()||phno.isEmpty()||maritalstatus.isEmpty()||pttype.isEmpty()||wardno.isEmpty()||bedno.isEmpty())

        JOptionPane.showMessageDialog(this,"Please fill all the details.");

    else{

        try{

            age = Integer.parseInt(jage.getText().trim());

            int option=JOptionPane.showConfirmDialog(this,"Are you sure you want to update?","Confirm Update",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

            if(option==JOptionPane.YES\_OPTION){

                try {

                    ResultSet rs = Oracle.execute("update patient set admitdate = '"+admitdate+"',pname = '"+pname+"',age = '"+age+"',gender = '"+gender+"',address  = '"+address+"',city = '"+city+"',phno = '"+phno+"',maritalstatus = '"+maritalstatus+"',pttype = '"+pttype+"',wardno = '"+wardno+"',bedno = '"+bedno+"')");

                    JOptionPane.showMessageDialog(this, "Patient details updated sucessfully");

                } catch (SQLException ex) {

                    Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

        } catch(NumberFormatException e) {

            JOptionPane.showMessageDialog(this, "Please fill all the details");

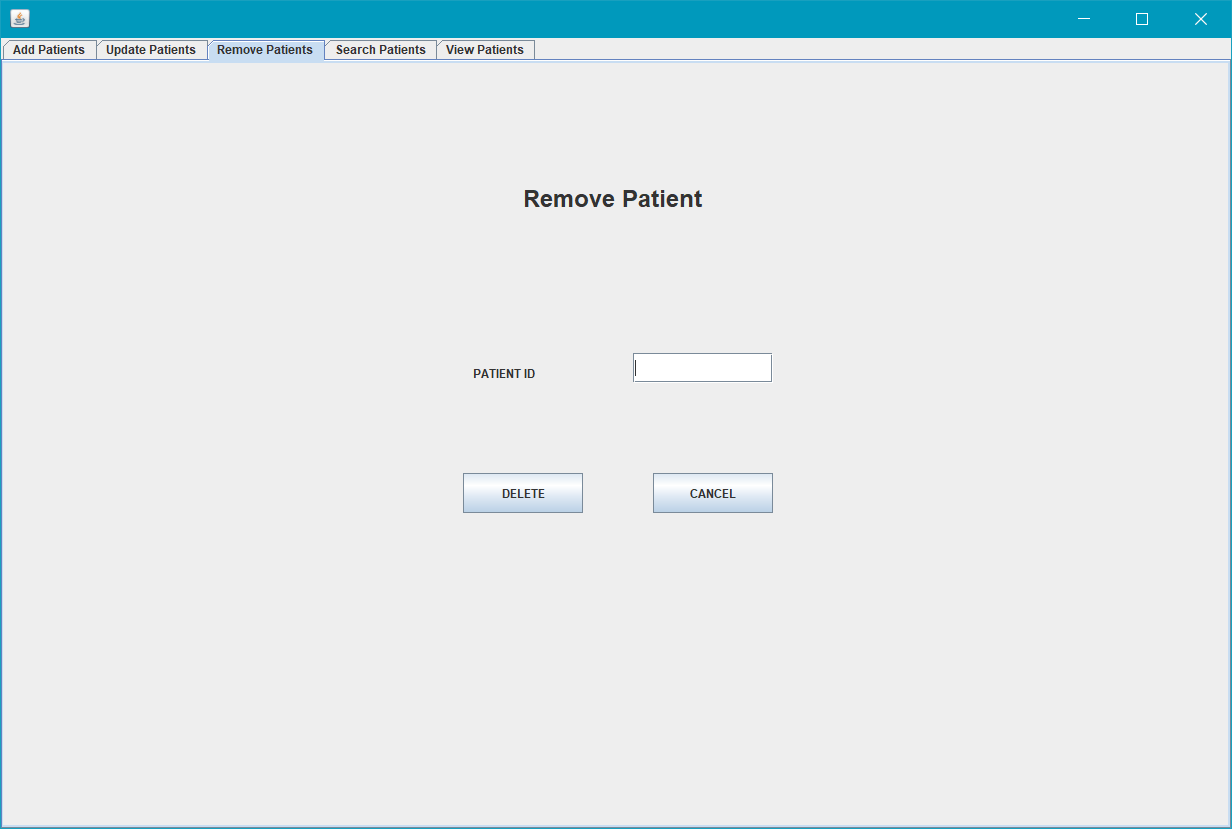
            Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, e);

        }

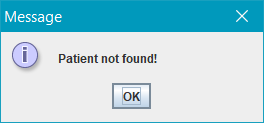
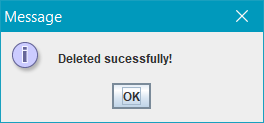
    }

}

**Remove Patients:**



On successful deletion: Invalid patient ID:



**Code:**

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    int response=JOptionPane.showConfirmDialog(this, "Are you sure you want to delete?", "Confirm Delete", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

    if(response==JOptionPane.YES\_OPTION){

        try {

            ResultSet rs = Oracle.execute("select \* from patient where patientid = '"+deletePatientIDText.getText().trim()+"'");

            if(rs.next())

            {

                Oracle.execute("delete from patient where patientid = '"+deletePatientIDText.getText().trim()+"'");

                JOptionPane.showMessageDialog(null, "Deleted sucessfully!");

            } else {

                JOptionPane.showMessageDialog(null, "Patient not found!");

            }

        } catch (SQLException ex) {

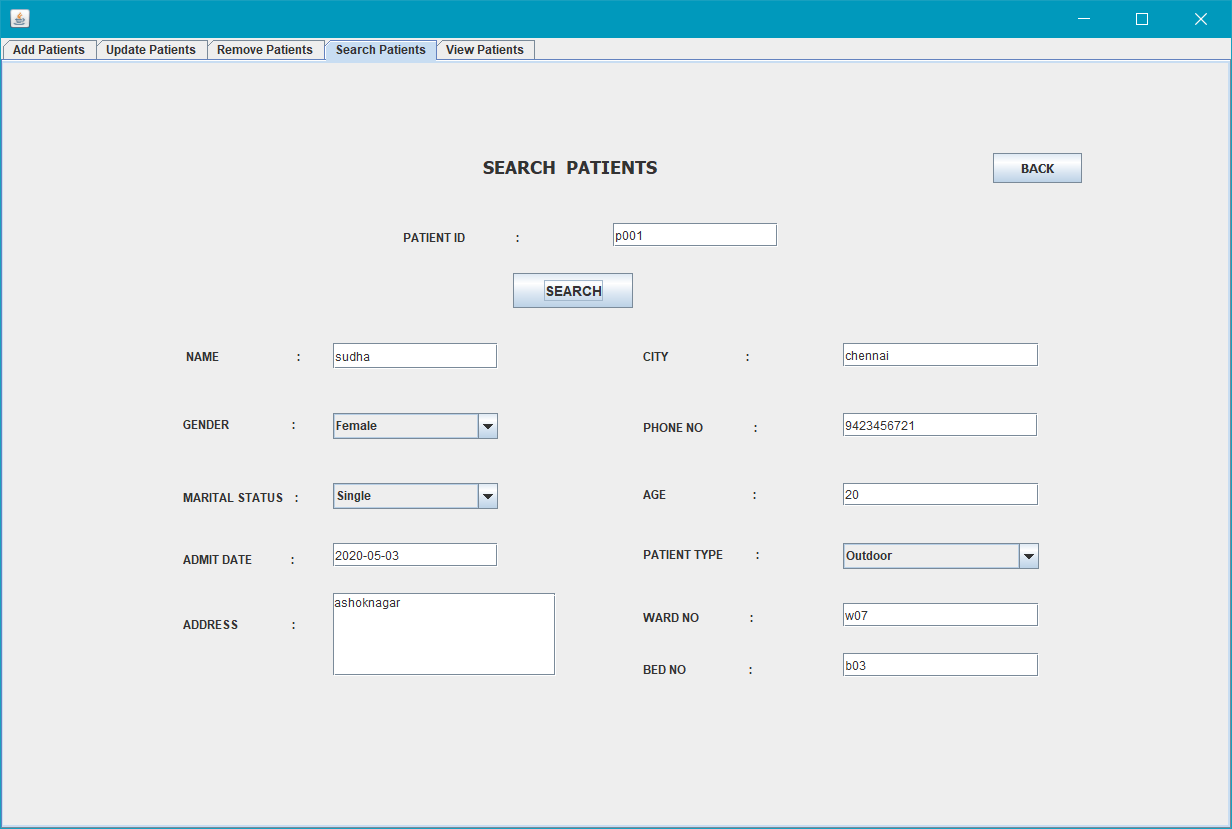
            Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

        }

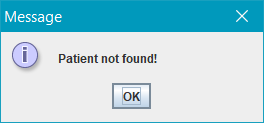
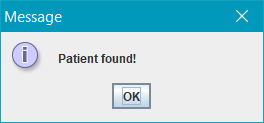
    }

}

**Search Patients:**



On successful search: On failed search:



**Code:**

private void searchButtonActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        ResultSet rs = Oracle.execute("select \* from patient where patientid = '"+pIDText.getText()+"'");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Patient found!");

            nameText.setText(rs.getString("pName"));

            int index = rs.getString("gender").equals("Male")? 0: 1;

            genderCombo.setSelectedIndex(index);

            index = rs.getString("maritalstatus").equals("Single")? 0: 1;

            maritalCombo.setSelectedIndex(index);

            dateText.setText(rs.getString("admitdate").substring(0,10));

            addText.setText(rs.getString("address"));

            cityText.setText(rs.getString("city"));

            phNoText.setText(rs.getString("phno"));

            ageText.setText(rs.getString("age"));

            index = rs.getString("pttype").equals("in patient")? 0: 1;

            patTypeCombo.setSelectedIndex(index);

            wardNoText.setText(rs.getString("wardno"));

            bedNoText.setText(rs.getString("bedno"));

        }

        else

        JOptionPane.showMessageDialog(this, "Patient not found!");

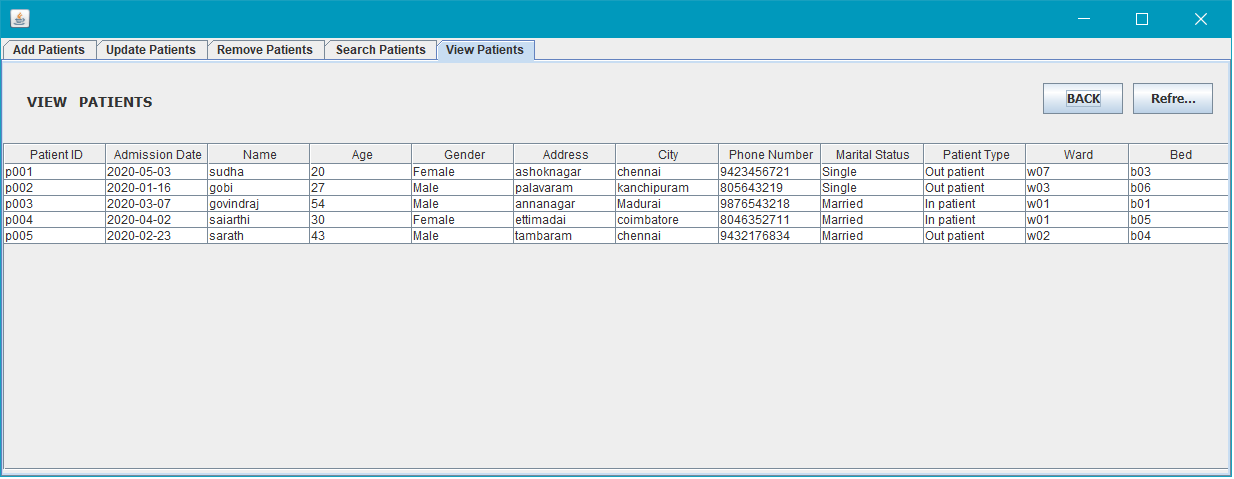
    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Patients:**

****

**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select \* from patient");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String patientid, admitdate, pname, age, gender, address, city, phno, maritalstatus, pttype, wardno, bedno;

            patientid = rs.getString(1);

            admitdate = rs.getString(2).substring(0, 10);

            pname = rs.getString(3);

            age = rs.getString(4);

            gender = rs.getString(5);

            address = rs.getString(6);

            city = rs.getString(7);

            phno = rs.getString(8);

            maritalstatus = rs.getString(9);

            pttype = rs.getString(10);

            wardno = rs.getString(11);

            bedno = rs.getString(12);

            model.addRow(new Object[]{patientid, admitdate, pname, age, gender, address, city, phno, maritalstatus, pttype, wardno, bedno});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

public recepPatientDetails() {

    initComponents();

    viewRefresh();

}

**Refresh Button Code:**

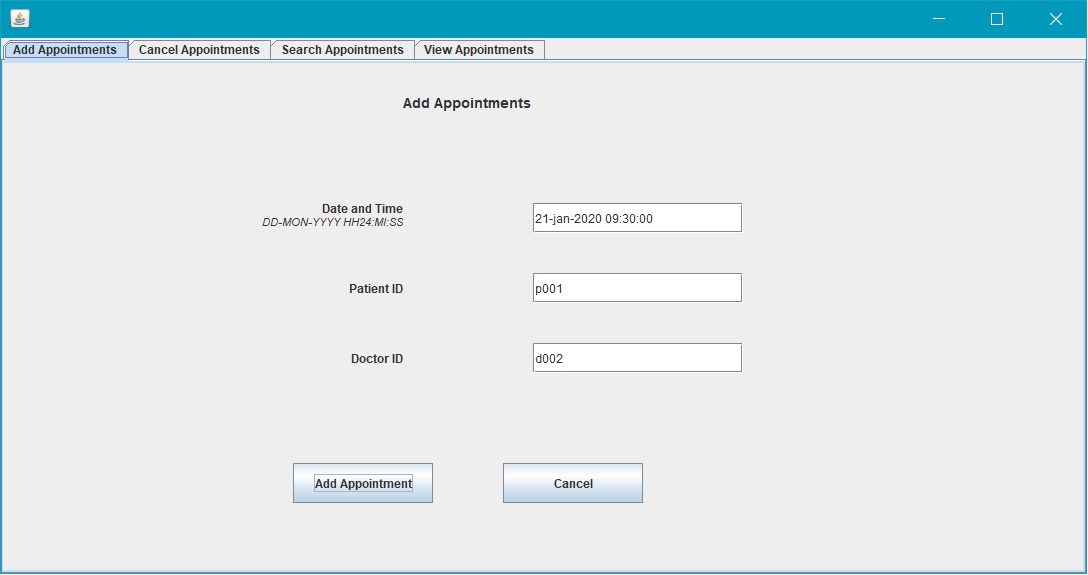
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    viewRefresh();

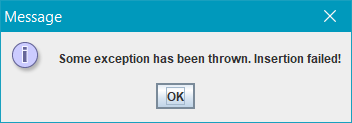
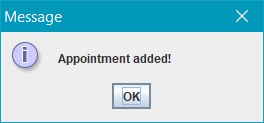
}

**Appointment Details:**

****

On successful appointment: When some error occurs in the

Query execution:



**Code:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        if(addAppDate.getText().isEmpty()||addAppDocID.getText().isEmpty()||addAppDocID.getText().isEmpty())

            JOptionPane.showMessageDialog(this, "Please fill all the details");

        else {

            ResultSet rs = Oracle.execute("insert into Appointments values(to\_date('"+addAppDate.getText().trim()+"','dd-mon-yyyy hh24:mi:ss'),'"+addAppDocID.getText().trim()+"','"+addAppPatID.getText().trim()+"')");

            JOptionPane.showMessageDialog(this, "Appointment added!"); }

    } catch (SQLException ex) {

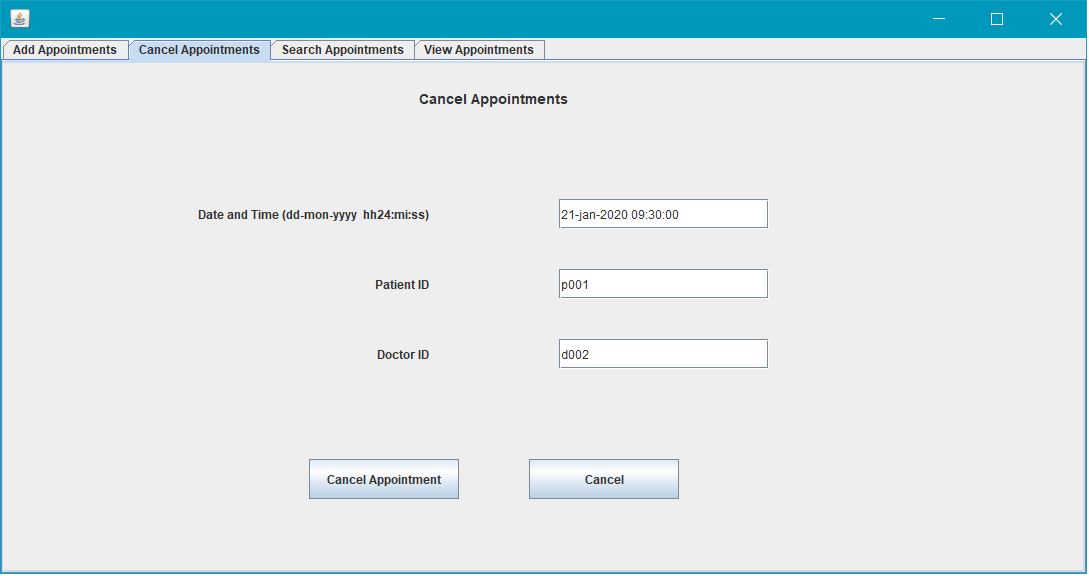
        Logger.getLogger(receprionistAppointments.class.getName()).log(Level.SEVERE, null, ex);

        JOptionPane.showMessageDialog(this, "Some exception has been thrown. Insertion failed!");

    }

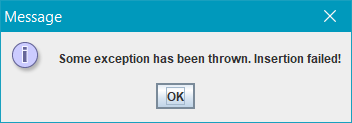
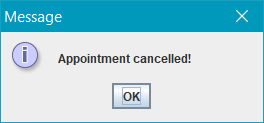
}

**Cancel Appointments:**

****

On successful cancellation: When some error occurs in the

Query execution:



**Code:**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        if(cancelAppDate.getText().isEmpty()||cancelDocID.getText().isEmpty()||cancelPatID.getText().isEmpty())

            JOptionPane.showMessageDialog(this, "Please fill all the details");

        else {

            ResultSet rs = Oracle.execute("delete from Appointments where appdatetime = (to\_date('"+cancelAppDate.getText().trim()+"','dd-mon-yyyy hh24:mi:ss')) and doctorID = '"+cancelDocID.getText().trim()+"'");

            JOptionPane.showMessageDialog(this, "Appointment cancelled!");

        }

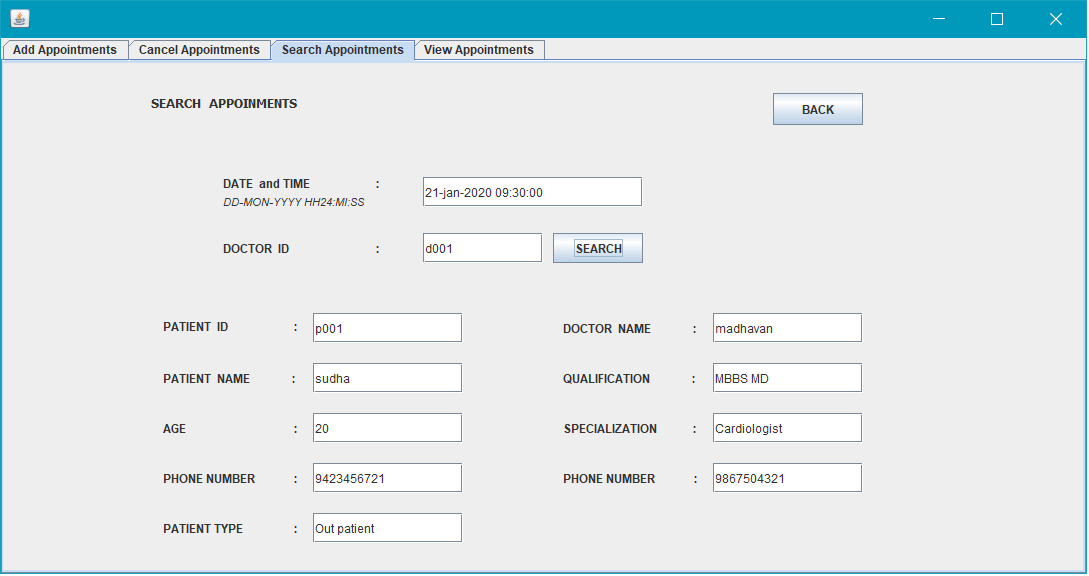
    } catch (SQLException ex) {

        Logger.getLogger(receprionistAppointments.class.getName()).log(Level.SEVERE, null, ex);

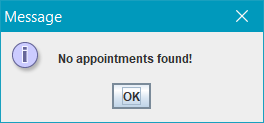
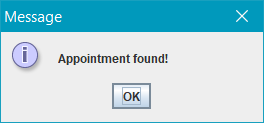
        JOptionPane.showMessageDialog(this, "Some exception has been thrown. Cancellation failed!"); }

}

**Search Appointments:**

****

On successful search: On failed search:



**Code:**

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        String appdatetime, doctorID;

        appdatetime = dateTime.getText();

        doctorID = searchDoctorID.getText();

        ResultSet rs = Oracle.execute("select p.patientID, p.pName, p.age, p.phNo, p.ptType, d.dName, d.qualification, d.specialization, d.phNo from patient p, doctor d, Appointments a where a.patientID = p.patientID and a.doctorID = d.doctorID and d.doctorID = '"+doctorID+"' and appDateTime = to\_date('"+appdatetime+"','dd-mon-yyyy hh24:mi:ss')");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Appointment found!");

            searchPatientID.setText(rs.getString(1));

            searchPatientName.setText(rs.getString(2));

            searchPatientAge.setText(rs.getString(3));

            searchPatientPhNo.setText(rs.getString(4));

            searchPatientType.setText(rs.getString(5));

            searchDoctorName.setText(rs.getString(6));

            searchDoctorQual.setText(rs.getString(7));

            searchDoctorSpl.setText(rs.getString(8));

            searchDoctorPhNo.setText(rs.getString(9));

        }

        else

        JOptionPane.showMessageDialog(this, "No appointments found!");

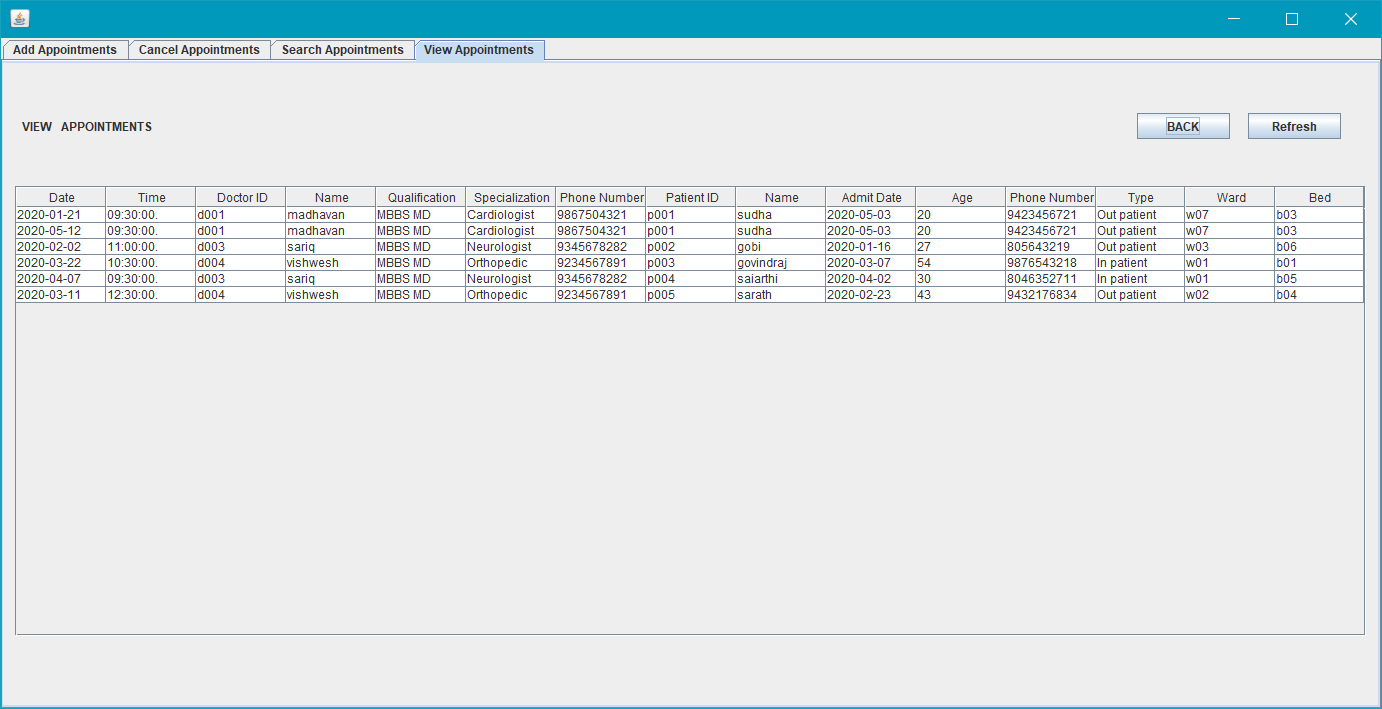
    } catch (SQLException ex) {

        Logger.getLogger(adminAppointDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Appointments:**



**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select a.appdatetime, d.doctorID, d.dname, d.qualification, d.specialization, d.phNo, p.patientID, p.pname, p.admitDate, p.age, p.phNo, p.pttype, p.wardNo, p.bedNo from Appointments a, Doctor d, Patient p where a.doctorID = d.doctorID and a.patientID = p.patientID");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String date, time, did, dname, qual, spl, dphno, pid, pname, admitdate, age, pphno, pttype, ward, bed;

            date = rs.getString(1).substring(0,10);

            time = rs.getString(1).substring(11,20);

            did = rs.getString(2); dname = rs.getString(3);

            qual = rs.getString(4); spl = rs.getString(5);

            dphno = rs.getString(6); pid = rs.getString(7);

            pname = rs.getString(8);

            admitdate = rs.getString(9).substring(0,10);

            age = rs.getString(10);

            pphno = rs.getString(11);

            pttype = rs.getString(12);

            ward = rs.getString(13);

            bed = rs.getString(14);

            model.addRow(new Object[]{date, time, did, dname, qual, spl, dphno, pid, pname, admitdate, age, pphno, pttype, ward, bed});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

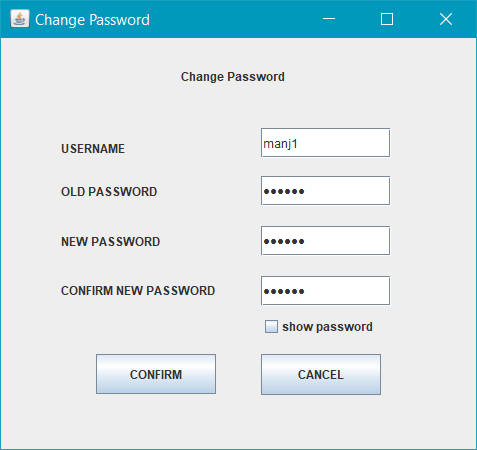
public receprionistAppointments() {

    initComponents();

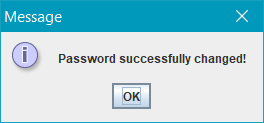
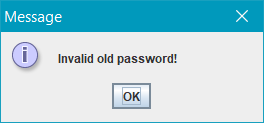
    viewRefresh();

}

**Change Password Window:**



Incorrect old password: On successful password change:



**Code:**

private void confirmButtonActionPerformed(java.awt.event.ActionEvent evt) {

    if(usernameText.getText().trim().isEmpty() || oldPass.getText().trim().isEmpty() || newPass.getText().trim().isEmpty() || confNewPass.getText().trim().isEmpty())

        JOptionPane.showMessageDialog(null,"Please fill all the credentials.");

    else{

        int option=JOptionPane.showConfirmDialog(this,"Are you sure you want to change your password?","Confirm",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

        if(option==JOptionPane.YES\_OPTION){

            if(newPass.getText().equals(confNewPass.getText()))

            {

                try {

                    String query = "select \* from receptionist where username = '"+usernameText.getText()+"'";

                    ResultSet rs = Oracle.execute(query);

                    if(rs.next()){

                        if(oldPass.getText().equals(rs.getString("password")))

                        {

                            query = "update receptionist set password = '"+newPass.getText()+"' where username = '"+usernameText.getText()+"'";

                            Oracle.execute(query);

                            JOptionPane.showMessageDialog(null,"Password successfully changed!");

                            this.dispose();

                        }

                        else

                            JOptionPane.showMessageDialog(null,"Invalid old Password!");

                    }

                    else

                        JOptionPane.showMessageDialog(this, "Invalid Username");

                } catch (SQLException ex) {

                    Logger.getLogger(changePasswordAdmin.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

            else

                JOptionPane.showMessageDialog(this, "New password and Confirm new password do not match");

        }

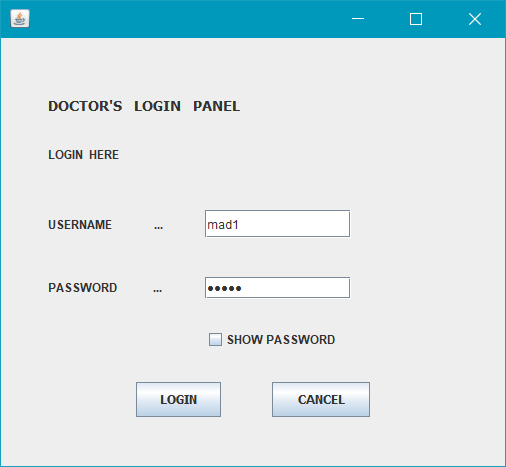
        else

            this.dispose();

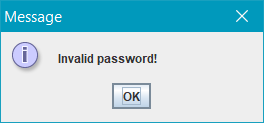
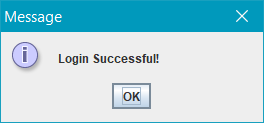
    }

}

**Doctor Login Page:**



Incorrect password: On successful login:

**Code:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    String password=docpassword.getText();

    String username=docusername.getText();

    if(docusername.getText().trim().isEmpty() || docpassword.getText().trim().isEmpty())

        JOptionPane.showMessageDialog(this,"Please fill all the credentials.");

    else{

        try

        {

            String query = "select \* from doctor where username = '"+username+"'";

            ResultSet rs;

            rs = Oracle.execute(query);

            if(rs.next())

            {

                String user = rs.getString("username");

                String pass = rs.getString("password");

                if(user.equals(username) && pass.equals(password))

                {

                    JOptionPane.showMessageDialog(this, "Login Successful!");

                    new doctorportal(username).setVisible(true);

                    this.dispose();

                }

                else

                    JOptionPane.showMessageDialog(this, "Invalid password!");

            }

            else

                JOptionPane.showMessageDialog(this, "Invalid Username!");

        } catch (SQLException ex) {

            Logger.getLogger(adminlogin.class.getName()).log(Level.SEVERE, null, ex);

        }

    }

}

**Show password button code:**

private void checkboxadminMouseClicked(java.awt.event.MouseEvent evt) {

    if(checkboxadmin.isSelected())

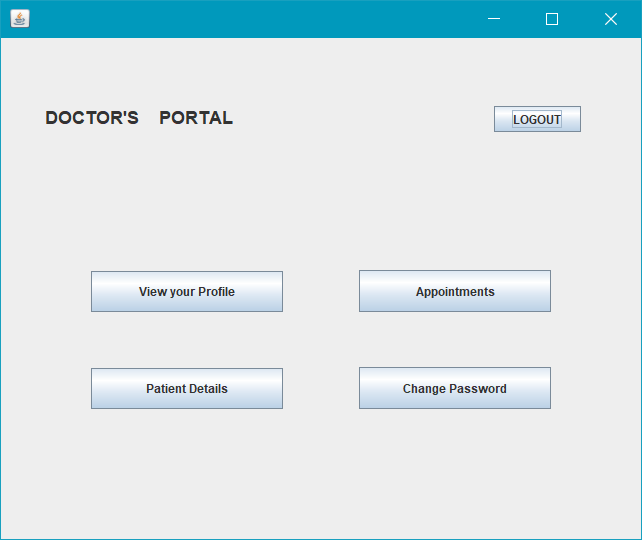
        docpassword.setEchoChar((char)0);

    else

        docpassword.setEchoChar(c);

}

**Doctor Portal:**



**Code:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    int option = JOptionPane.showConfirmDialog(this, "Are you sure you want to log out?", "Confirm Logout", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

    if(option == JOptionPane.YES\_OPTION)

    {

        this.dispose();

        new mainframe().setVisible(true);

    }

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    new doctorAppointments(username).setVisible(true);

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    new changePasswordDoctor().setVisible(true);

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    new doctorProfile(username).setVisible(true);

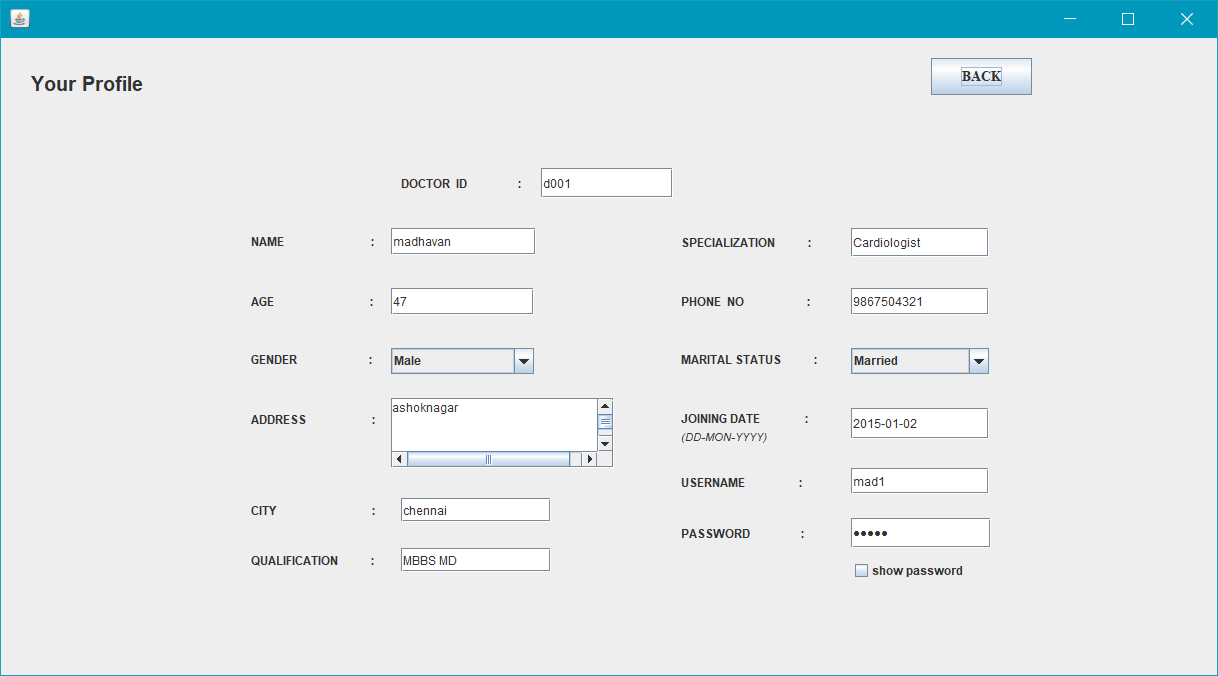
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    new doctorPatientDetails().setVisible(true);

}

**Profile Window:**



**Code:**

public doctorProfile(String username) {

    initComponents();

    c = updatePassword.getEchoChar();

    this.username = username;

    try {

        ResultSet rs;

        rs = Oracle.execute("select doctorid from doctor where username = '"+username+"'");

        if(rs.next())

            updateID.setText(rs.getString(1));

        rs = Oracle.execute("select \* from doctor where doctorid = '"+updateID.getText().trim()+"'");

        if(rs.next())

        {

            updateName.setText(rs.getString(2));

            updateAge.setText(rs.getString(3));

            int index = rs.getString(4).equals("Male")?0:1;

            updateGender.setSelectedIndex(index);

            updateAddress.setText(rs.getString(5));

            updateCity.setText(rs.getString(6));

            updatePhNo.setText(rs.getString(7));

            index = rs.getString(8).equals("Single")?0:1;

            updateMarital.setSelectedIndex(index);

            updateQual.setText(rs.getString(9));

            updateSpl.setText(rs.getString(10));

            updateJoinDate.setText(rs.getString(11).substring(0,10));

            updateUsername.setText(rs.getString(12));

            updatePassword.setText(rs.getString(13));

        }else

            JOptionPane.showMessageDialog(this, "Doctor not found!");

    } catch (SQLException ex) {

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, ex);

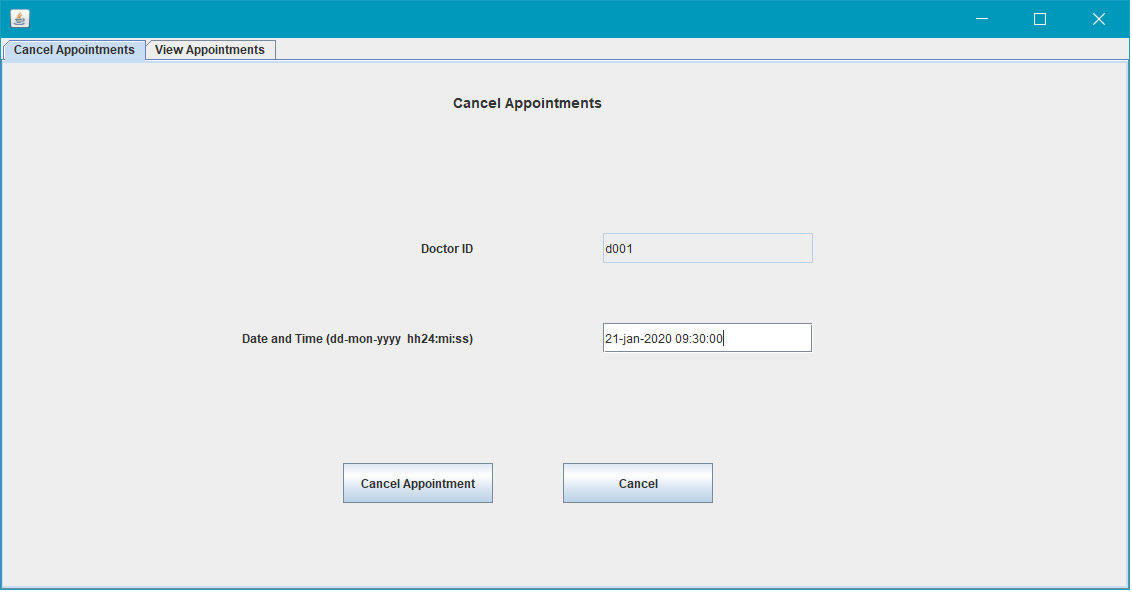
    }

}

**Appointment Details:**

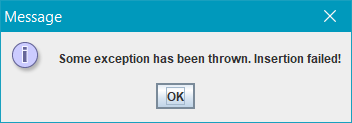
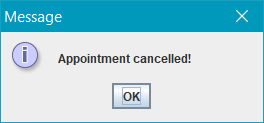
Since we know the ID of the doctor currently loggind into the system, we use it to restrict the doctor to view and cancel only his appointments.

**Cancel Appointments:**

****

On successful cancellation: When some error occurs in the

Query execution:



**Code:**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        if(cancelAppDate.getText().isEmpty()||cancelDocID.getText().isEmpty())

        JOptionPane.showMessageDialog(this, "Please fill all the details");

        else

        {

            ResultSet rs = Oracle.execute("delete from Appointments where appdatetime = to\_date('"+cancelAppDate.getText().trim()+"','dd-mon-yyyy hh24:mi:ss') and doctorID = '"+cancelDocID.getText().trim()+"'");

            JOptionPane.showMessageDialog(this, "Appointment cancelled!");

        }

    } catch (SQLException ex) {

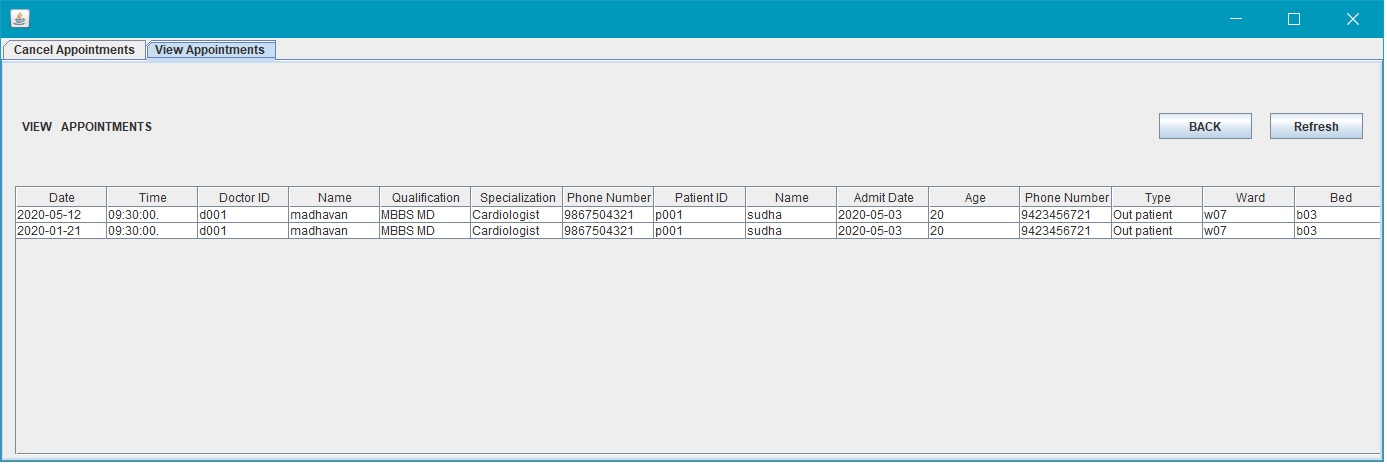
        Logger.getLogger(receprionistAppointments.class.getName()).log(Level.SEVERE, null, ex);

        JOptionPane.showMessageDialog(this, "Some exception has been thrown. Cancellation failed!");

    }

}

**View Appointments:**

****

**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select a.appdatetime, d.doctorID, d.dname, d.qualification, d.specialization, d.phNo, p.patientID, p.pname, p.admitDate, p.age, p.phNo, p.pttype, p.wardNo, p.bedNo from Appointments a, Doctor d, Patient p where a.doctorID = d.doctorID and a.patientID = p.patientID and d.doctorID = '"+doctorID+"'");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String date, time, did, dname, qual, spl, dphno, pid, pname, admitdate, age, pphno, pttype, ward, bed;

            date = rs.getString(1).substring(0,10);

            time = rs.getString(1).substring(11,20);

            did = rs.getString(2);

            dname = rs.getString(3);

            qual = rs.getString(4);

            spl = rs.getString(5);

            dphno = rs.getString(6);

            pid = rs.getString(7);

            pname = rs.getString(8);

            admitdate = rs.getString(9).substring(0,10);

            age = rs.getString(10);

            pphno = rs.getString(11);

            pttype = rs.getString(12);

            ward = rs.getString(13);

            bed = rs.getString(14);

            model.addRow(new Object[]{date, time, did, dname, qual, spl, dphno, pid, pname, admitdate, age, pphno, pttype, ward, bed});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

public doctorAppointments(String username) {

    initComponents();

    this.username = username;

    try {

        ResultSet rs = Oracle.execute("select doctorID from doctor where username = '"+username+"'");

        if(rs.next())

            doctorID = rs.getString(1);

    } catch (SQLException ex) {

        Logger.getLogger(doctorAppointments.class.getName()).log(Level.SEVERE, null, ex);

    }

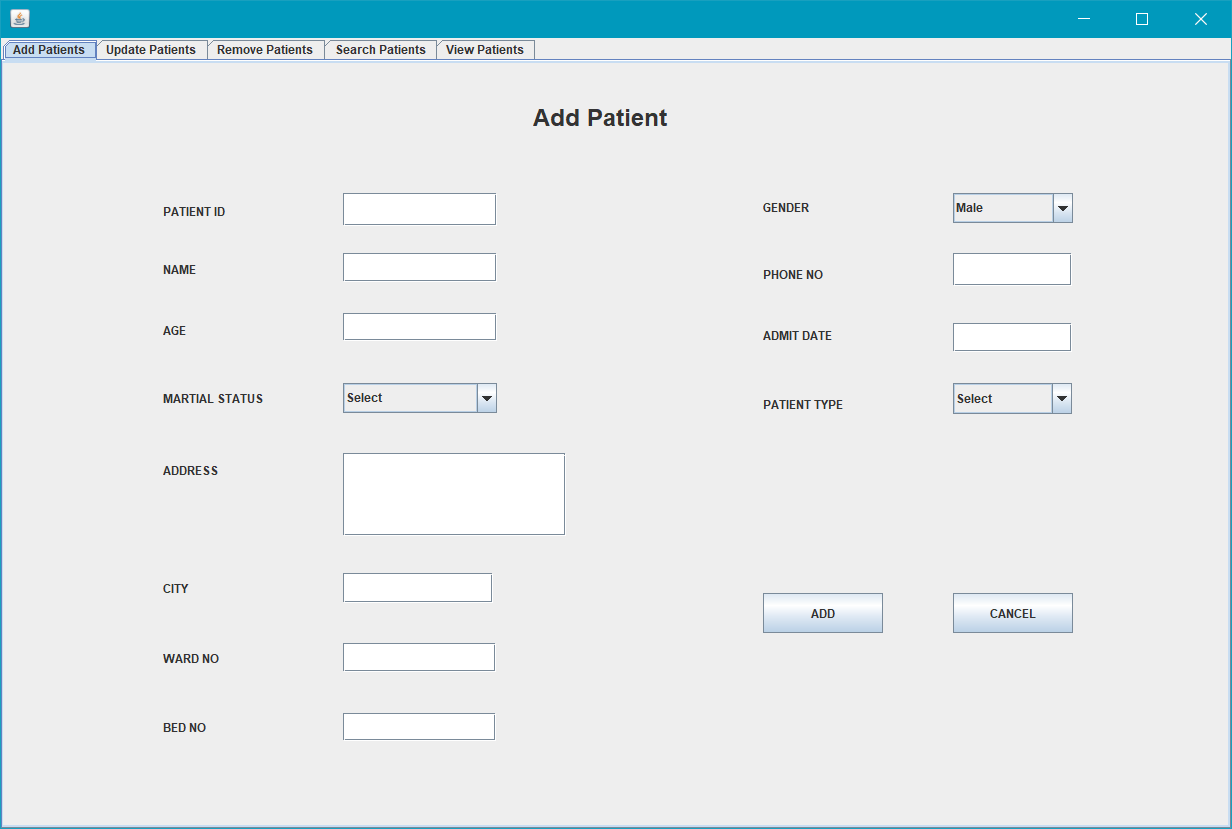
    cancelDocID.setText(doctorID);

    viewRefresh();

}

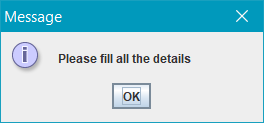
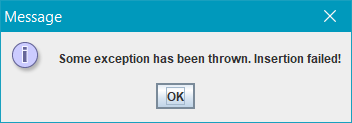
**Patient Details Window:**

**Add Patients:**



When some fields are empty: When some error occurs in the

query execution:

**Code:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    patientid = pid.getText().trim();

    admitdate = date.getText().trim();

    pname = name.getText().trim();

    gender = addGenderCombo.getSelectedItem().toString();

    address = add.getText().trim();

    city = addCityText.getText().trim();

    phno = pno.getText().trim();

    maritalstatus = marital.getSelectedItem().toString();

    pttype = ptype.getSelectedItem().toString();

    wardno = wno.getText().trim();

    bedno = bno.getText().trim();

    try{

        age = Integer.parseInt(addAgeText.getText().trim());

        if(patientid.isEmpty()||pname.isEmpty()||address.isEmpty()||city.isEmpty()||admitdate.isEmpty()||gender.isEmpty()||phno.isEmpty()||maritalstatus.isEmpty()||pttype.isEmpty()||wardno.isEmpty()||bedno.isEmpty())

            JOptionPane.showMessageDialog(this, "Please fill all the details");

        else{

            try {

                ResultSet rs = Oracle.execute("insert into patient values('"+patientid+"','"+admitdate+"','"+pname+"','"+age+"','"+gender+"','"+address+"','"+city+"','"+phno+"','"+maritalstatus+"','"+pttype+"','"+wardno+"','"+bedno+"')");

                JOptionPane.showMessageDialog(this, "Added Sucessfully");

            } catch (SQLException ex) {

                Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    } catch (NumberFormatException e) {

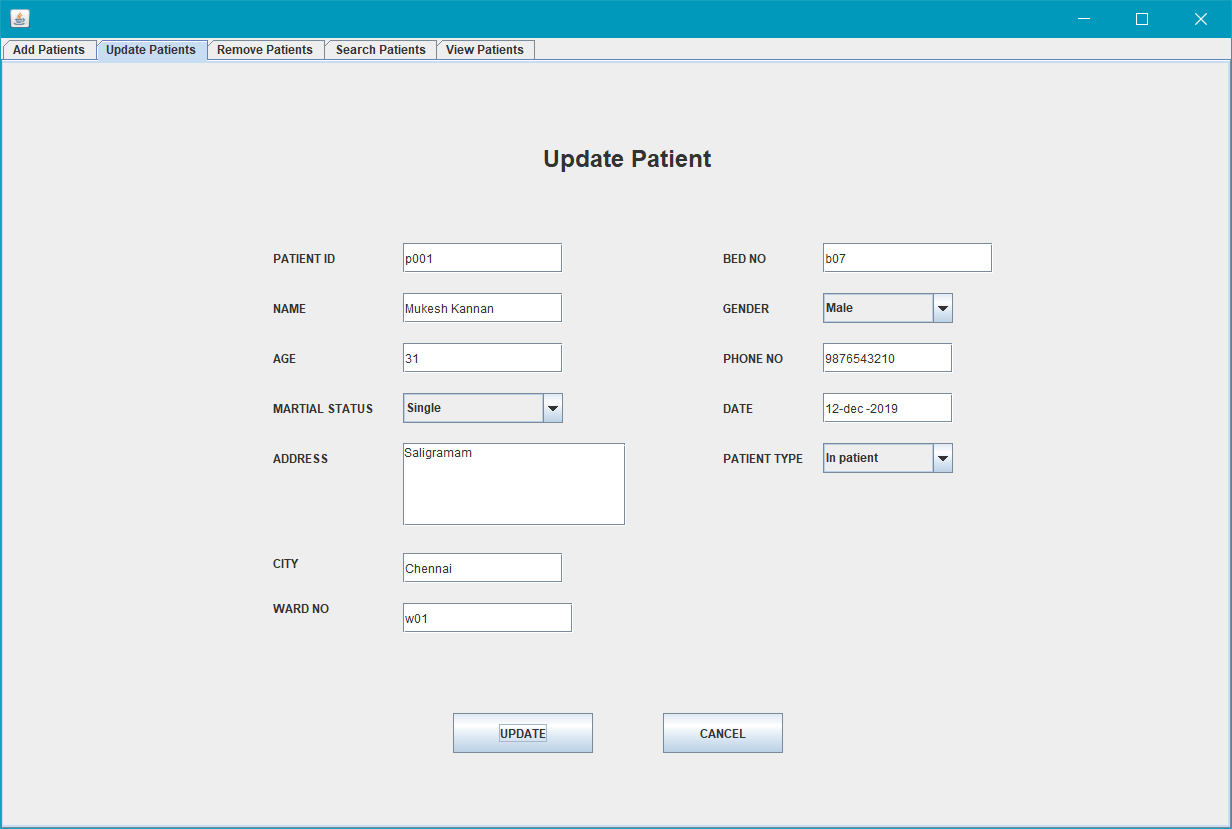
        JOptionPane.showMessageDialog(this, "Please fill all the details");

        Logger.getLogger(adminRecepDetails.class.getName()).log(Level.SEVERE, null, e);

    }

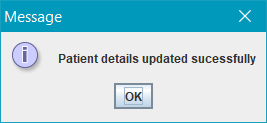
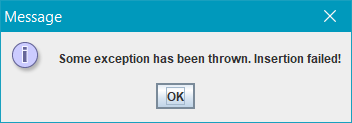
}

**Update Patients:**



On successful updation: When some error occurs in the

query execution:

**Code:**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    patientid = jpid.getText().trim();

    admitdate = jdate.getText().trim();

    pname = jname.getText().trim();

    gender = jgender.getSelectedItem().toString().trim();

    address = jadd.getText().trim();

    city = jcity.getText().trim();

    phno = jpno.getText().trim();

    maritalstatus = jmarital.getSelectedItem().toString().trim();

    pttype = jpttype.getSelectedItem().toString().trim();

    wardno = jwno.getText().trim();

    bedno = jbno.getText().trim();

    if(patientid.isEmpty()||admitdate.isEmpty()||pname.isEmpty()||gender.isEmpty()||address.isEmpty()||city.isEmpty()||phno.isEmpty()||maritalstatus.isEmpty()||pttype.isEmpty()||wardno.isEmpty()||bedno.isEmpty())

        JOptionPane.showMessageDialog(this,"Please fill all the details.");

    else{

        try{

            age = Integer.parseInt(jage.getText().trim());

            int option=JOptionPane.showConfirmDialog(this,"Are you sure you want to update?","Confirm Update",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

            if(option==JOptionPane.YES\_OPTION){

                try {

                    ResultSet rs = Oracle.execute("update patient set admitdate = '"+admitdate+"',pname = '"+pname+"',age = '"+age+"',gender = '"+gender+"',address  = '"+address+"',city = '"+city+"',phno = '"+phno+"',maritalstatus = '"+maritalstatus+"',pttype = '"+pttype+"',wardno = '"+wardno+"',bedno = '"+bedno+"')");

                    JOptionPane.showMessageDialog(this, "Patient details updated sucessfully");

                } catch (SQLException ex) {

                    Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

        } catch(NumberFormatException e) {

            JOptionPane.showMessageDialog(this, "Please fill all the details");

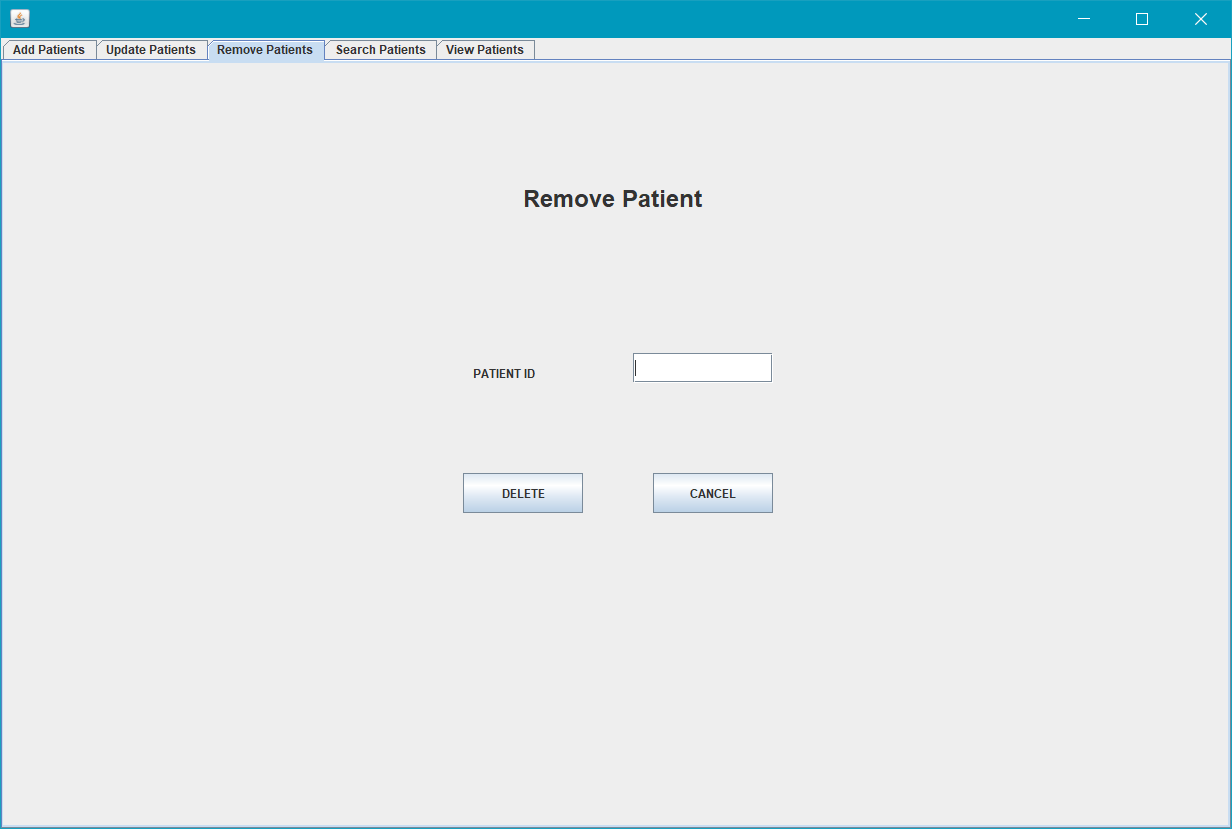
            Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, e);

        }

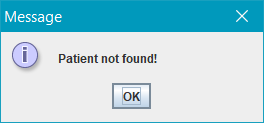
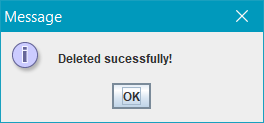
    }

}

**Remove Patients:**



On successful deletion: Invalid patient ID:



**Code:**

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    int response=JOptionPane.showConfirmDialog(this, "Are you sure you want to delete?", "Confirm Delete", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

    if(response==JOptionPane.YES\_OPTION){

        try {

            ResultSet rs = Oracle.execute("select \* from patient where patientid = '"+deletePatientIDText.getText().trim()+"'");

            if(rs.next())

            {

                Oracle.execute("delete from patient where patientid = '"+deletePatientIDText.getText().trim()+"'");

                JOptionPane.showMessageDialog(null, "Deleted sucessfully!");

            } else {

                JOptionPane.showMessageDialog(null, "Patient not found!");

            }

        } catch (SQLException ex) {

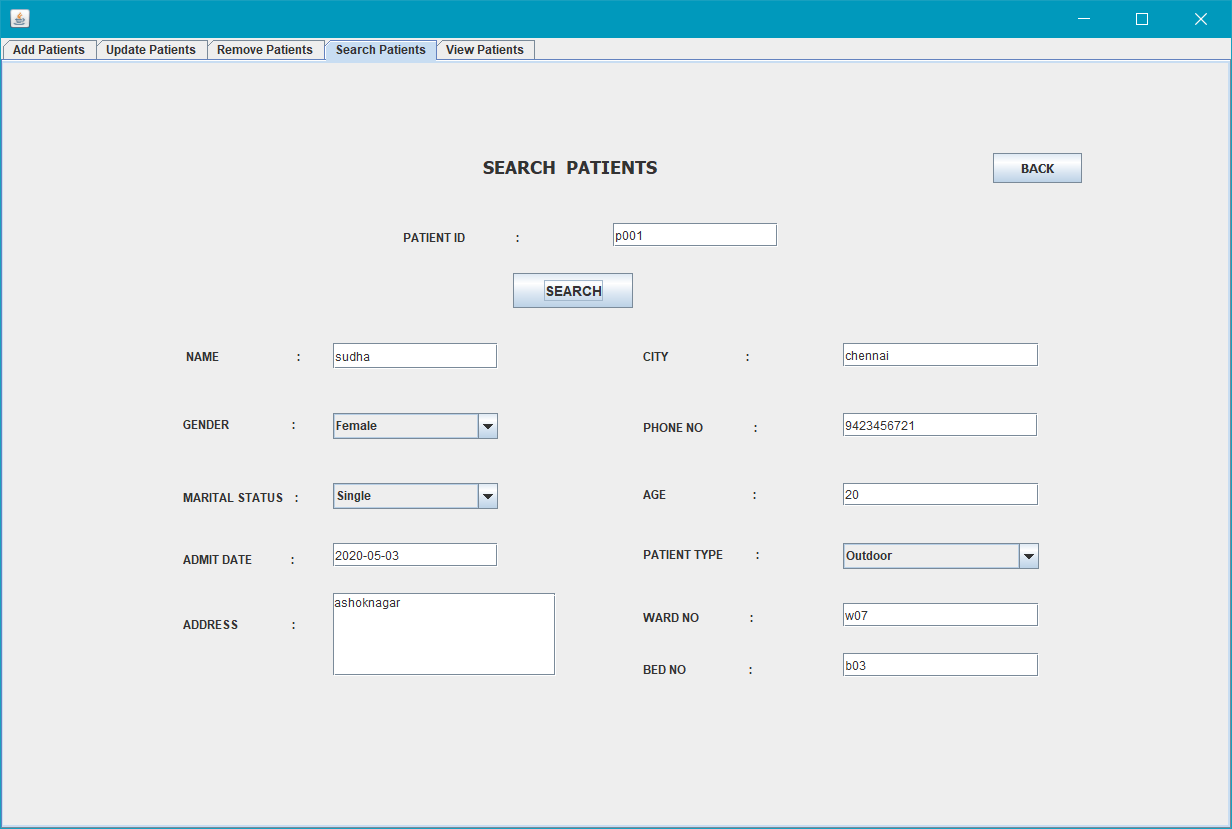
            Logger.getLogger(recepPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

        }

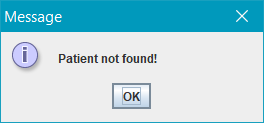
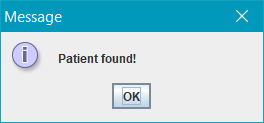
    }

}

**Search Patients:**



On successful search: On failed search:



**Code:**

private void searchButtonActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        ResultSet rs = Oracle.execute("select \* from patient where patientid = '"+pIDText.getText()+"'");

        if(rs.next())

        {

            JOptionPane.showMessageDialog(this, "Patient found!");

            nameText.setText(rs.getString("pName"));

            int index = rs.getString("gender").equals("Male")? 0: 1;

            genderCombo.setSelectedIndex(index);

            index = rs.getString("maritalstatus").equals("Single")? 0: 1;

            maritalCombo.setSelectedIndex(index);

            dateText.setText(rs.getString("admitdate").substring(0,10));

            addText.setText(rs.getString("address"));

            cityText.setText(rs.getString("city"));

            phNoText.setText(rs.getString("phno"));

            ageText.setText(rs.getString("age"));

            index = rs.getString("pttype").equals("in patient")? 0: 1;

            patTypeCombo.setSelectedIndex(index);

            wardNoText.setText(rs.getString("wardno"));

            bedNoText.setText(rs.getString("bedno"));

        }

        else

        JOptionPane.showMessageDialog(this, "Patient not found!");

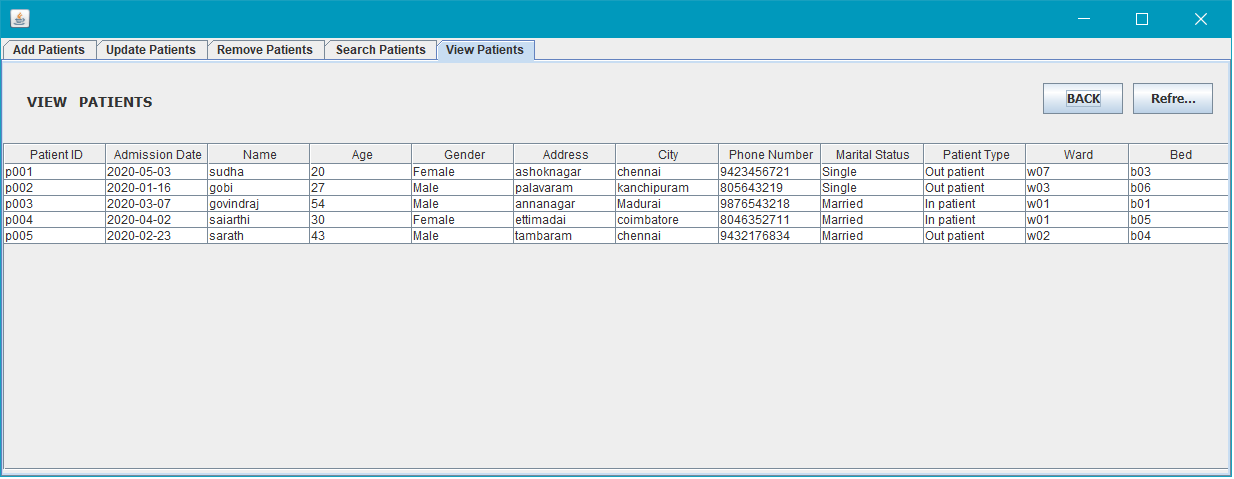
    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

**View Patients:**

****

**Code:**

private void viewRefresh()

{

    try {

        ResultSet rs = Oracle.execute("select \* from patient");

        DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

        model.setRowCount(0);

        while(rs.next())

        {

            String patientid, admitdate, pname, age, gender, address, city, phno, maritalstatus, pttype, wardno, bedno;

            patientid = rs.getString(1);

            admitdate = rs.getString(2).substring(0, 10);

            pname = rs.getString(3);

            age = rs.getString(4);

            gender = rs.getString(5);

            address = rs.getString(6);

            city = rs.getString(7);

            phno = rs.getString(8);

            maritalstatus = rs.getString(9);

            pttype = rs.getString(10);

            wardno = rs.getString(11);

            bedno = rs.getString(12);

            model.addRow(new Object[]{patientid, admitdate, pname, age, gender, address, city, phno, maritalstatus, pttype, wardno, bedno});

        }

    } catch (SQLException ex) {

        Logger.getLogger(adminPatientDetails.class.getName()).log(Level.SEVERE, null, ex);

    }

}

public recepPatientDetails() {

    initComponents();

    viewRefresh();

}

**Refresh Button Code:**

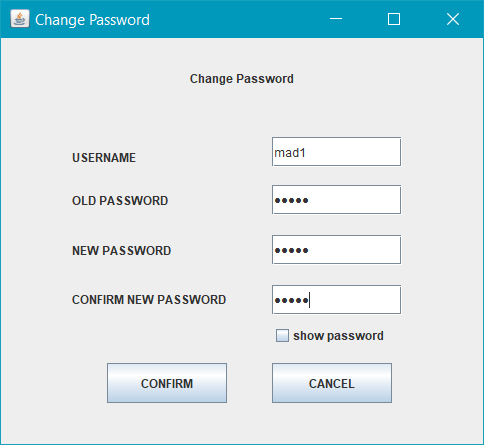
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

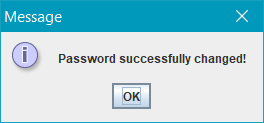
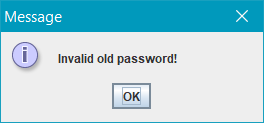
    viewRefresh();

}

**Change Password Window:**



Incorrect old password: On successful password change:



**Code:**

private void confirmButtonActionPerformed(java.awt.event.ActionEvent evt) {

    if(usernameText.getText().trim().isEmpty() || oldPass.getText().trim().isEmpty() || newPass.getText().trim().isEmpty() || confNewPass.getText().trim().isEmpty())

        JOptionPane.showMessageDialog(null,"Please fill all the credentials.");

    else{

        int option=JOptionPane.showConfirmDialog(this,"Are you sure you want to change your password?","Confirm",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

        if(option==JOptionPane.YES\_OPTION){

            if(newPass.getText().equals(confNewPass.getText()))

            {

                try {

                    String query = "select \* from doctor where username = '"+usernameText.getText()+"'";

                    ResultSet rs = Oracle.execute(query);

                    if(rs.next())

                    {

                        if(oldPass.getText().equals(rs.getString("password")))

                        {

                            query = "update doctor set password = '"+newPass.getText()+"' where username = '"+usernameText.getText()+"'";

                            Oracle.execute(query);

                            JOptionPane.showMessageDialog(null,"Password successfully changed!");

                            this.dispose();

                        }

                        else

                            JOptionPane.showMessageDialog(null,"Invalid old passsword!");

                    }

                    else

                        JOptionPane.showMessageDialog(this, "Invalid Username");

                } catch (SQLException ex) {

                    Logger.getLogger(changePasswordAdmin.class.getName()).log(Level.SEVERE, null, ex);

                }

            }

            else

                JOptionPane.showMessageDialog(this, "New password and Confirm new password do not match");

        }

        else

            this.dispose();

    }

}