DBMS PROJECT REPORT

B.Tech-ICT
Semester-IV
Course Name - DBMS
(Database Management System)

Healthcare Management System

Group No :- 34
Group Members:-
Yugamsinh Chavda (AU1841090)
Parth D Patel (AU1841123)
Kishan Patel (AU1841132)

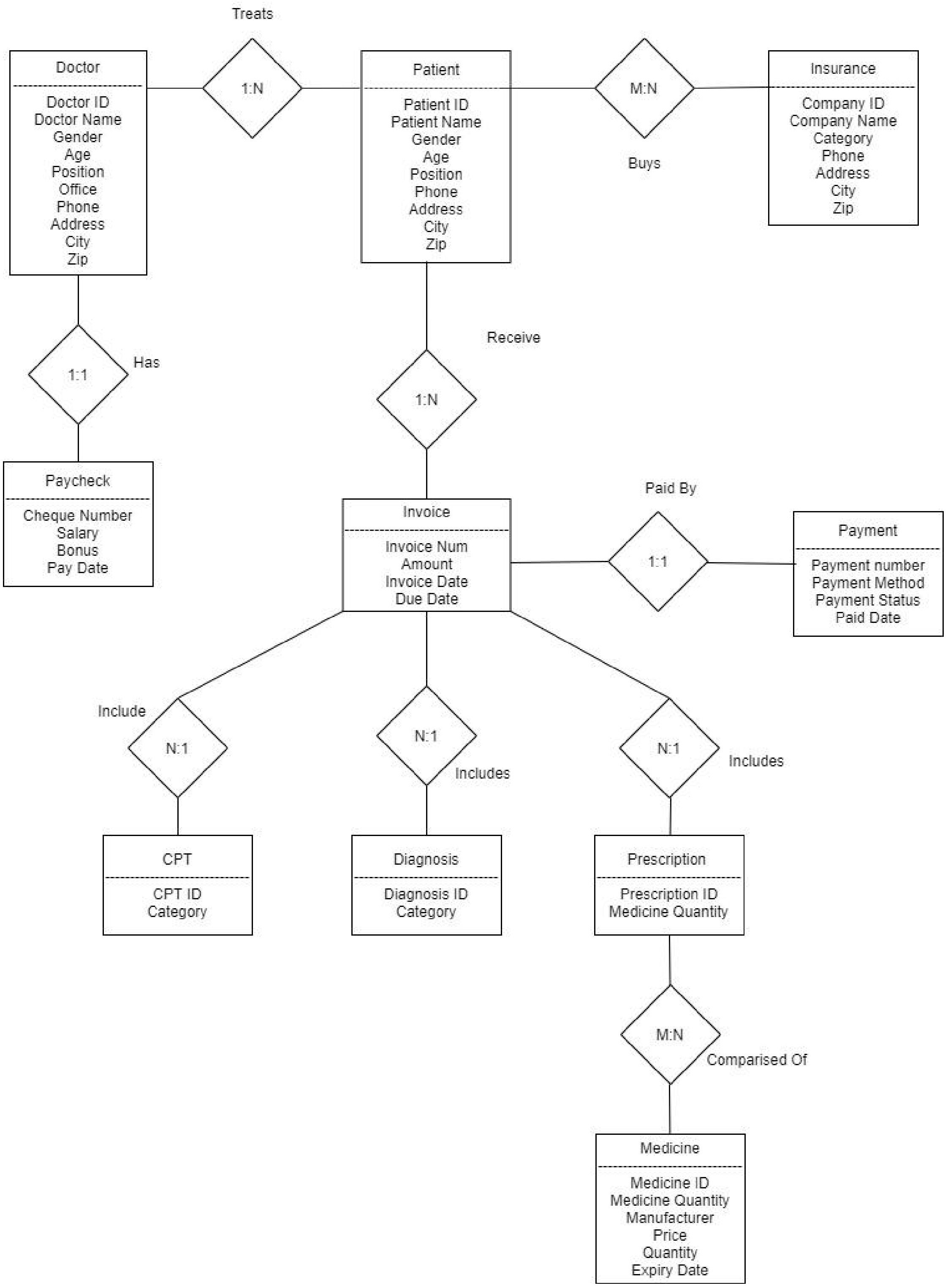
Description

In our healthcare management system we have made software for patient and doctor portfolio management systems and also monitor and management of medicines. Using this software the management will easily access and edit patient and doctor ‘s data according to their necessities.This system is to facilitate the center to retrieve, update, and report the patient information efficiently,in turn helping the doctors make timely,effective diagnoses.

Currently, different departments in the healthcare center have their own separated systems leading to the lack of communications and the inefficient data sharing. For example, the finance department uses simple EXCEL spreadsheets to record the paycheck information of the employees which is inconvenient to retrieve and update employee’s information; in the clinic department, the doctors have to write down the prescriptions for the patients and keep paper documents, and also do not have any information about the patients’ insurance plans; the medicine department has to keep the prescription and inventory records on their own computer system. While each system serves a distinctive purpose, there is no coordinating, assimilating and representing of data. The systems may have duplicate data which is a waste of space. The different systems also may have different application programs which cause incompatible files. Due to these kinds of disadvantages of the current system, we can propose our health care management system instead.Health care management is a database management system.

The DBMS can track and update all the information of recorded patients in the healthcare center during a particular time span. The major advantages of the DBMS are easy to retrieve and update information, efficient data sharing and communication, and reliable backup and security.

E-R Diagram


Treats
Doctor
Doctor ID
Doctor Name
Gender
Age
Position
Office
Phone
Address
City
Zip
1:N
Patient
Patient ID
Patient Name
Gender
Age
Position
Phone
Address
City
Zip
M:N
Buys
Insurance
Company ID
Company Name
Category
Phone
Address
City
Zip
1:1
Has
Paycheck
Cheque Number
Salary
Bonus
Pay Date
Receive
1:N
Invoice
Invoice Num
Amount
Invoice Date
Due Date
Paid By
1:1
Payment
Payment number
Payment Method
Payment Status
Paid Date
Include
N:1
N:1
Includes
N:1
Includes
CPT
CPT ID
Category
Diagnosis
Diagnosis ID
Category
Prescription
Prescription ID
Medicine Quantity
M:N
Comparised Of
Medicine
Medicine ID
Medicine Quantity
Manufacturer
Price
Quantity
Expiry Date

Data-Dictionary

- Table- Doctor

Column Name	Data Type	Constraints	Format	Description
D_ID	Varchar2(5)	Primary key	D1	Doctor id
D_NAME	Varchar2(50)	Not Null	Samarth	Doctor Name
GENDER	Varchar2(10)	Not Null	Male	Gender of Doctor
AGE	Number(22)	Not Null	32	Doctor’s Age
POSITION	Varchar2(30)	Not Null	Dentist	Doctor’s Medical Field Name
OFFICE	Varchar2(30)	Not Null	Navarangpura	Office or clinic Location
PHONE	Number(22)	Not Null	98567412	Doctor’s Contact Number
ADDRESS	Varchar2(100)	Not Null	12,2nd floor, Krishna Complex.	Doctor’s Clinic Address
CITY	Varchar2(50)	Not Null	Ahmedabad	Name of city where Clinic is Located
ZIP	Number(22)	Not Null	382006	Pincode

- Table - Patient

Column Name	**Data Type**	**Constraints**	**Format**	**Description**
P_ID	Varchar2(5)	Primary Key	P1	Patient Id
P_NAME	Varchar2(50)	Not Null	Jay	Patient Name
GENDER	Varchar2(10)	Not Null	Male	Gender of Patient
AGE	Number(22)	Not Null	32	Patient’s Age
POSITION	Varchar2(30)	Not Null	Dentist	Patient’s Working Field Name
PHONE	Number(22)	Not Null	98567412	Patient’s Contact Number
ADDRESS	Varchar2(100)	Not Null	12,2nd floor, Krishna Complex.	Patient’s Residential Address
CITY	Varchar2(50)	Not Null	Ahmedabad	Name of city where Resident is Located
ZIP	Number(22)	Not Null	382006	Pincode

- Table - Patient_Doc

Column Name	Data Type	Constraints	Format	Description
D_ID	Varchar2(5)	Primary Key(1)	D1	Doctor Id
P_ID	Varchar2(5)	Primary Key(2)	P1	Patient Id

- Table - Medicine

Column Name	Data Type	Constraints	Format	Description
M_ID	Varchar2(5)	Primary Key	M1	Medicine Id
M_NAME	Varchar2(25)	Not Null	Chlorophyll	Medicine Name
MANUFACTURER	Varchar2(25)	Not Null	Cipla	Medicine's Manufacturer Name
PRICE	Number(22)	Not Null	500	Medicine's Price
QTY	Number(22)	Not Null	10 Tablets	Medicine's Qty
EXP_DATE	Date	Not Null	19-04-2022	Expiry Date

- Table - Medicine_P

Column Name	Data Type	Constraints	Format	Description
PRESCRIPTION_ID	Varchar2(5)	Primary Key(1)	PR1	Prescription Id
M_ID	Varchar2(5)	Primary Key(2)	M1	Medicine Id

- Table - Insurance

Column Name	Data Type	Constraints	Format	Description
INSCO_ID	Varchar2(10)	Primary Key	IC1	Insurance Id
INSCO_NAME	Varchar2(25)	Not Null	LIC	Insurance Company Name
CATEGORY	Varchar2(20)	Not Null	Full	Term Period Of Insurance Policy
PHONE	Number(22)	Not Null	98567412	Policy Holder’s Contact Number
ADDRESS	Varchar2(50)	Not Null	12,2nd floor, Krishna Complex.	Policy Holder’s Residential Address
CITY	Varchar2(25)	Not Null	Ahmedabad	Name of city where Resident is Located
ZIP	Number(22)	Not Null	382006	Pincode

- Table - PatientInsurance

Column Name	Data Type	Constraints	Format	Description
P_ID	Varchar2(5)	Primary Key(1)	P1	Patient Id
INSCO_ID	Varchar2(10)	Primary Key(2)	IC1	Insurance Id

- Table - Diagnosis

Column Name	Data Type	Constraints	Format	Description
DIAGNOSIS_ID	Varchar2(5)	Primary Key	DG1	Diagnosis Id
CATEGORY	Varchar2(20)	Not null	Mental Pain	Disease Category

- Table - CPT

Column Name	Data Type	Constraints	Format	Description
CPT_ID	Varchar2(5)	Primary Key	C1	CPT Id
CATEGORY	Varchar2(20)	Not null	Anesthesia	Category of CPT

- Table - Prescription

Column Name	Data Type	Constraints	Format	Description
PRESCRIPTION_ID	Varchar2(5)	Primary Key	PR1	Prescription Id
MEDICINE_Q	Number(22)	Not null	6	Qty of Medicine

- Table - Invoice_P

Column Name	Data Type	Constraints	Format	Description
INVOICE_NUM	Varchar2(10)	Primary Key	I1	Invoice Number
P_ID	Varchar2(5)	Not Null	P1	Patient Id
CPT_ID	Varchar2(5)	Not Null	C1	CPT Id
DIAGNOSIS_ID	Varchar2(5)	Not Null	DG1	Diagnosis Id
PRESCRIPTIN_ID	Varchar2(5)	Not Null	PR1	Prescription Id
AMOUNT	Number(22)	Not Null	1500	Total Amount
INVOICE_DATE	Date	Not Null	19-04-2020	Date whenInvoice is generated
DUE_DATE	Date	Not Null	20-04-2020	Deadline Date to Pay amount of Invoice

- Table - Paycheck_Doc

Column Name	Data Type	Constraints	Format	Description
CHK_NUM	Varchar2(5)	Primary Key	A1	Cheque Number
D_ID	Varchar2(5)	Primary Key	D1	Doctor’s Id
SALARY	Number(22)	Not null	50000	Amount Of Salary
BONUS	Number(22)	Not null	5000	Bonus amount
PAY_DATE	Date	Not null	20-04-2020	Date of Salary Received

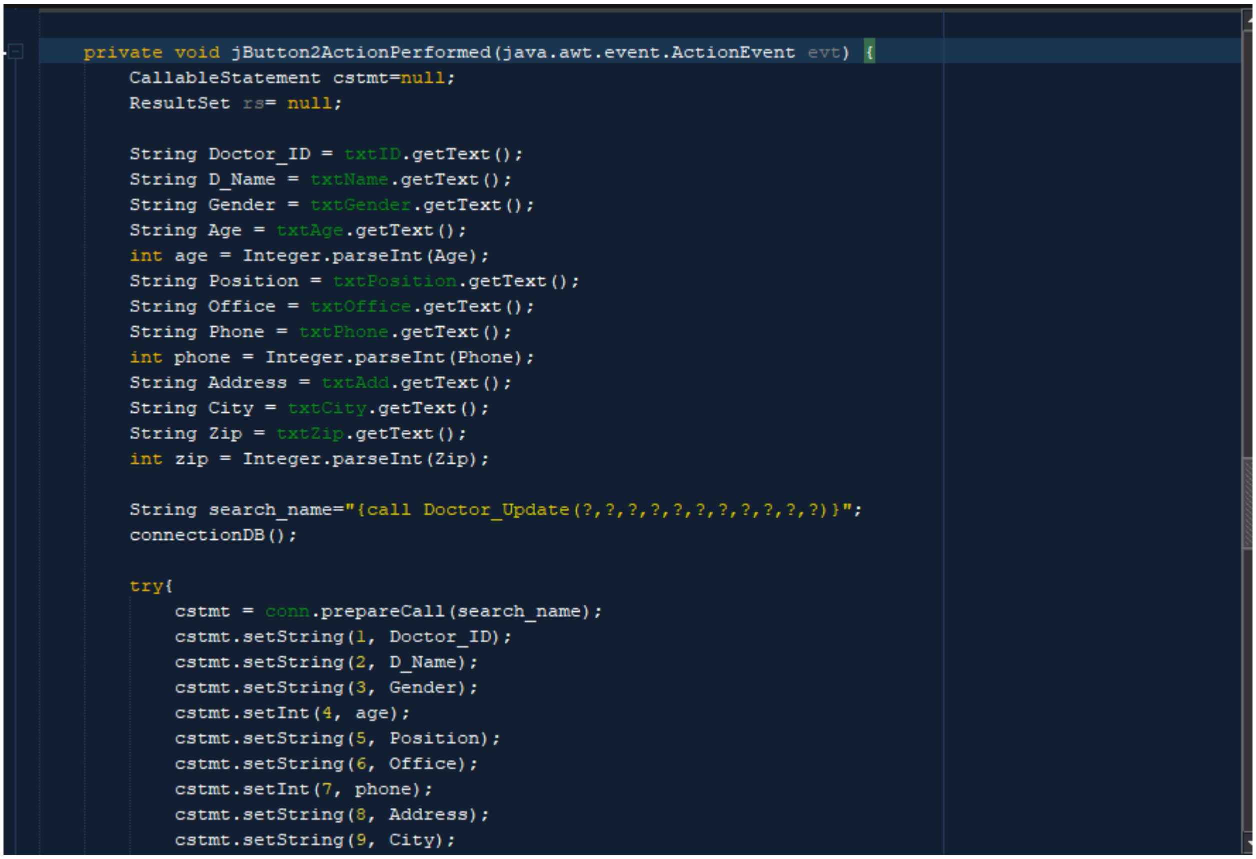
- Table - Payment_P

Column Name	Data Type	Constraints	Format	Description
P_NUM	Varchar2(10)	Primary Key	PAY1	Payment Id
INVOICE	Varchar2(10)	Not null	I1	Invoice Id
PAY_METHOD	Varchar2(20)	Not null	cash	PaymentMethod
PAY_STATUS	Varchar2(20)	Not null	paid	Payment Status
PAID_DATE	Date	Not null	20-04-2020	Payment Date

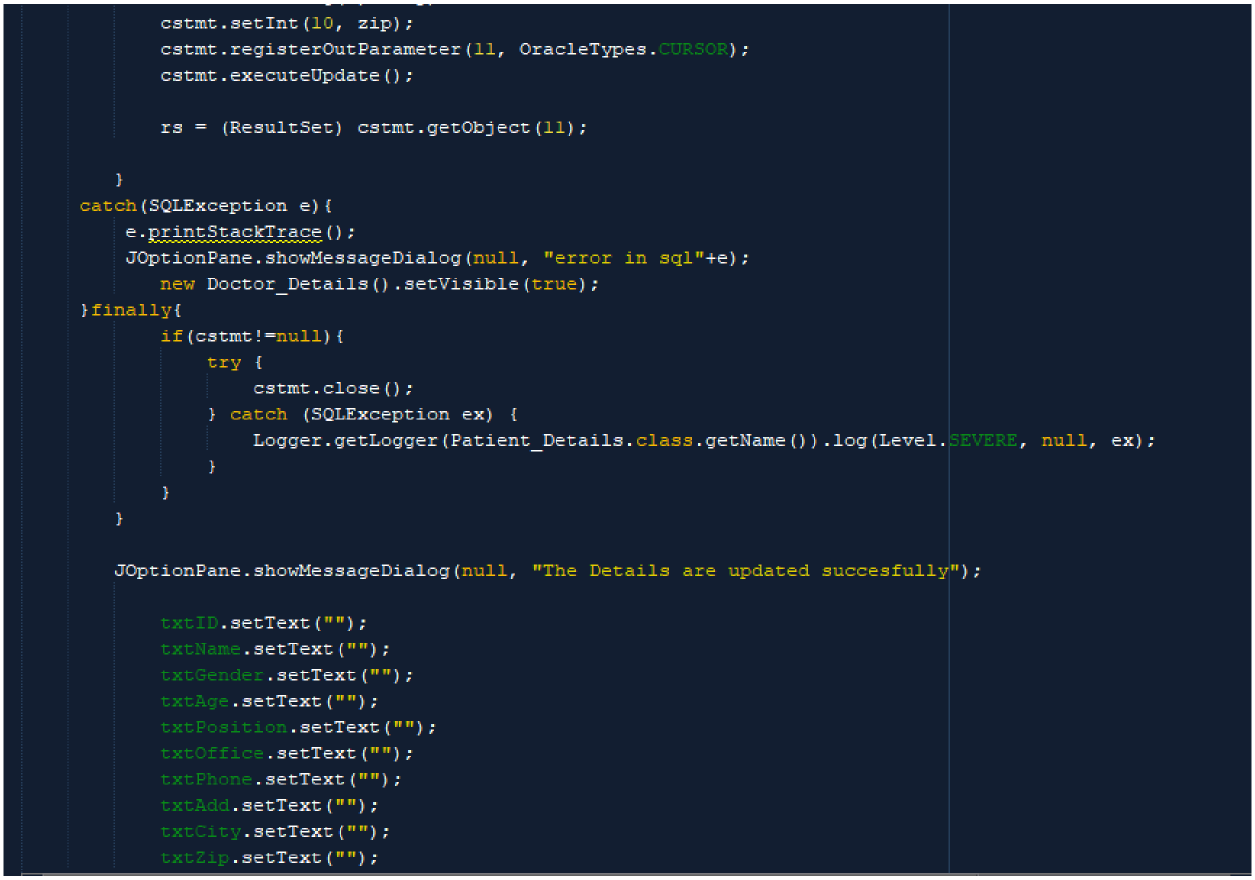
Stored Procedures

For Updating:

Calling a Stored Procedure in Java:(For Updating)



```java
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    CallableStatement cstmt=null;
    ResultSet rs= null;

    String Doctor_ID = txtID.getText();
    String D_Name = txtName.getText();
    String Gender = txtGender.getText();
    String Age = txtAge.getText();
    int age = Integer.parseInt(Age);
    String Position = txtPosition.getText();
    String Office = txtOffice.getText();
    String Phone = txtPhone.getText();
    int phone = Integer.parseInt(Phone);
    String Address = txtAdd.getText();
    String City = txtCity.getText();
    String Zip = txtZip.getText();
    int zip = Integer.parseInt(Zip);

    String search_name="{call Doctor_Update(?,?,?,?,?,?,?,?,?,?,?)}";
    connectionDB();

    try{
        cstmt = conn.prepareCall(search_name);
        cstmt.setString(1, Doctor_ID);
        cstmt.setString(2, D_Name);
        cstmt.setString(3, Gender);
        cstmt.setInt(4, age);
        cstmt.setString(5, Position);
        cstmt.setString(6, Office);
        cstmt.setInt(7, phone);
        cstmt.setString(8, Address);
        cstmt.setString(9, City);
```



```
       cstmt.setInt(10, zip);
       cstmt.registerOutParameter(11, OracleTypes.CURSOR);
       cstmt.executeUpdate();

       rs = (ResultSet) cstmt.getObject(11);

   }
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "error in sql"+e);
       new Doctor_Details().setVisible(true);
}finally{
       if(cstmt!=null){
           try {
               cstmt.close();
           } catch (SQLException ex) {
               Logger.getLogger(Patient_Details.class.getName()).log(Level.SEVERE, null, ex);
           }
       }
   }

   JOptionPane.showMessageDialog(null, "The Details are updated succesfully");

       txtID.setText("");
       txtName.setText("");
       txtGender.setText("");
       txtAge.setText("");
       txtPosition.setText("");
       txtOffice.setText("");
       txtPhone.setText("");
       txtAdd.setText("");
       txtCity.setText("");
       txtZip.setText("");
```

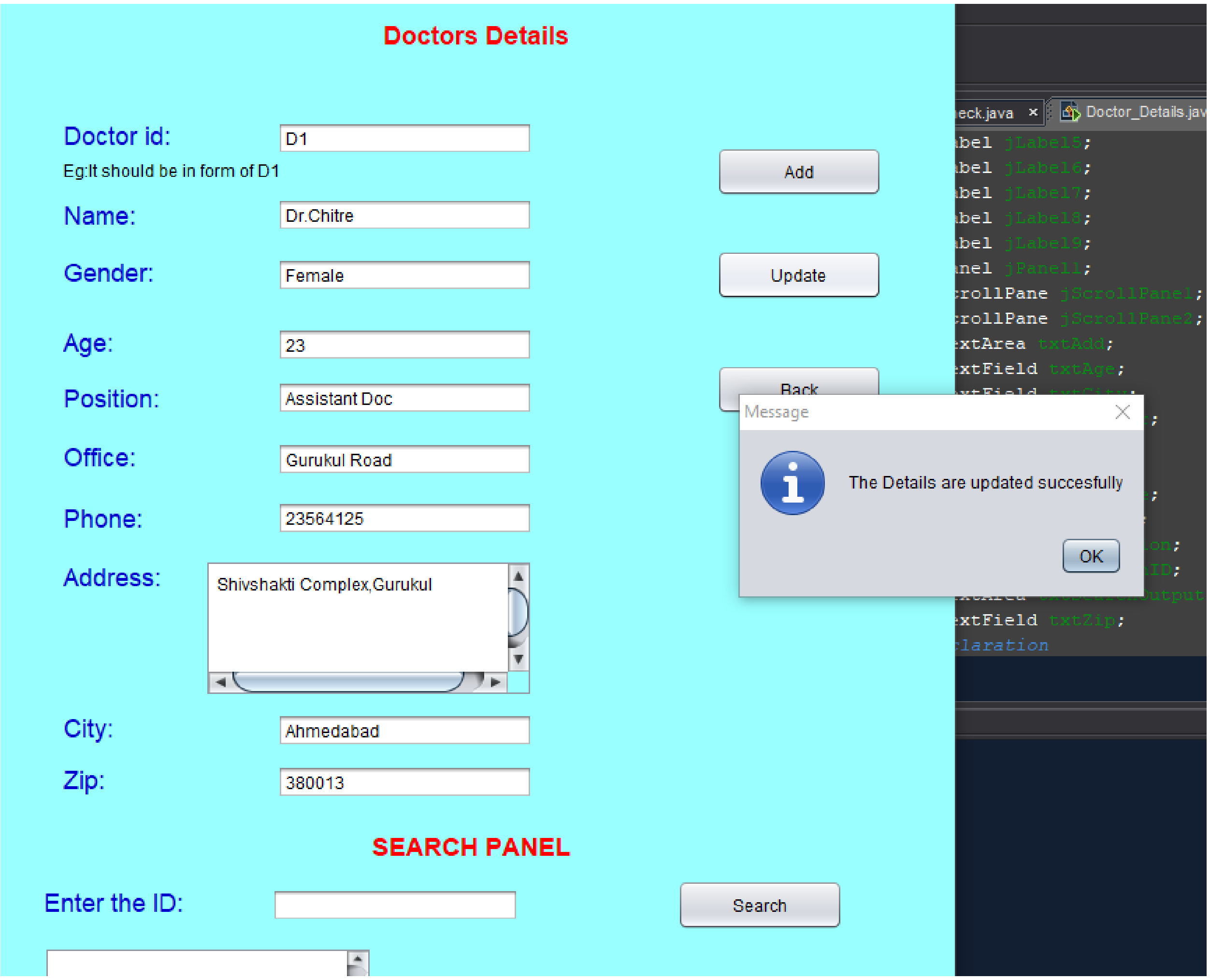
- Doctor Table:

Oracle

create or replace procedure Doctor_Update(x in varchar,y in varchar,z in varchar,a in int,b in varchar,c in varchar,d in int,e in varchar,f in varchar,g in int, c_p out sys_refcursor) as cursor c_d is select * from Doctor for update nowait;
r_d c_d%ROWTYPE;
begin
open c_d;
LOOP
FETCH c_d into r_d;

```
EXIT WHEN c_d%NOTFOUND;
if(r_d.D_id = x) then
update Doctor set D_name = y,Gender = z,Age = a,Position = b,Office =
c,Phone = d,Address = e,City = f,Zip = g where current of c_d;
end if;
end loop;
close c_d;
End;
```

Output:



Doctor id: D1

Eg:It should be in form of D1

Name: Dr.Chitre

Gender: Female

Age: 23

Position: Assistant Doc

Office: Gurukul Road

Phone: 23564125

Address: Shivshakti Complex,Gurukul

City: Ahmedabad

Zip: 380013

Enter the ID:

Message

The Details are updated succesfully

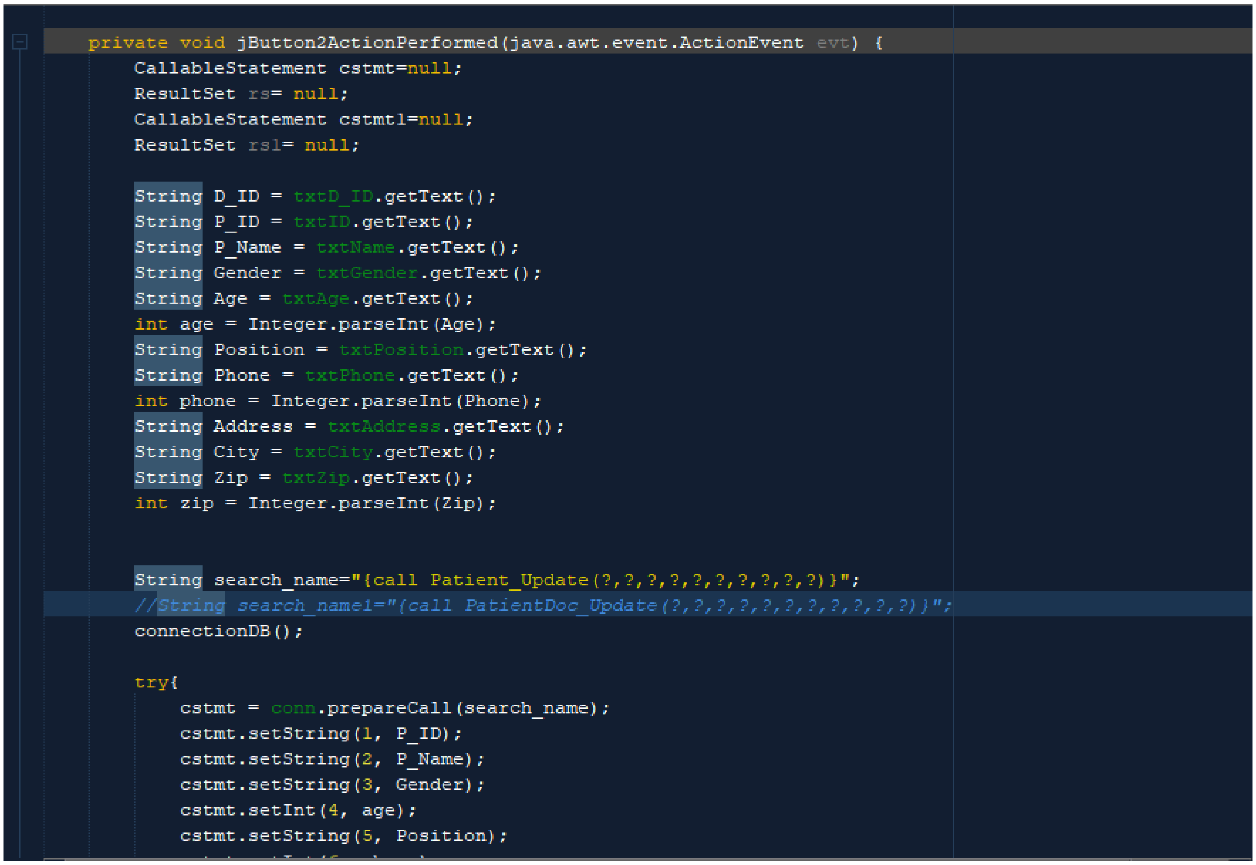
OK

```
abel jLabel5;
abel jLabel6;
abel jLabel7;
abel jLabel8;
abel jLabel9;
anel jPanel1;
crollPane jScrollPane1;
crollPane jScrollPane2;
extArea txtAdd;
extField txtAge;
extField txtZip;
claration
```

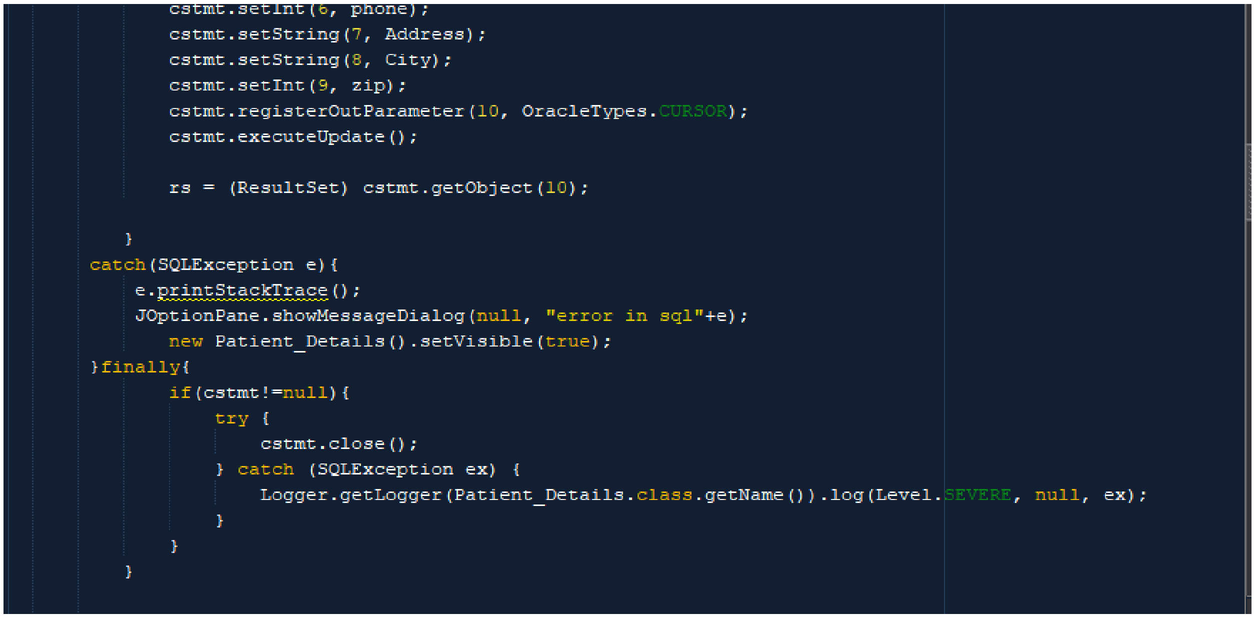
- Patient Table

```
create or replace procedure Patient_Update(x in varchar,y in
varchar,z in varchar,a in int,b in varchar,d in int,e in varchar,f in
varchar,g in int, c_p out sys_refcursor) as cursor c_pa is select * from
Patient for update nowait;
r_pa c_pa%ROWTYPE;
begin
open c_pa;
LOOP
FETCH c_pa into r_pa;
EXIT WHEN c_pa%NOTFOUND;
if(r_pa.P_id = x) then
update Patient set P_name = y,Gender = z,Age = a,Position =
b,Phone = d,Address = e,City = f,Zip = g where current of c_pa;
end if;
end loop;
close c_pa;
end;
```

Calling the Procedure in Java

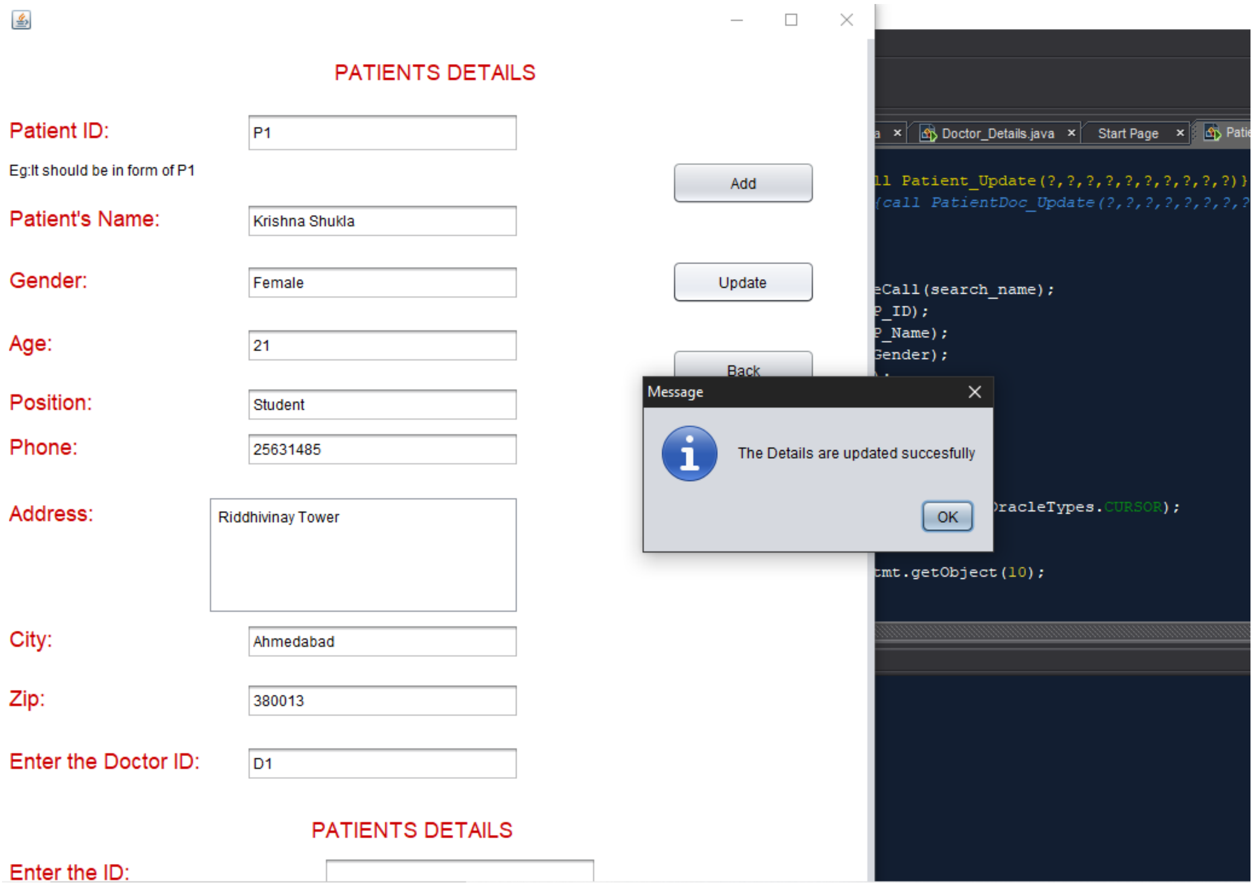


```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        CallableStatement cstmt=null;
        ResultSet rs= null;
        CallableStatement cstmt1=null;
        ResultSet rs1= null;

        String D_ID = txtD_ID.getText();
        String P_ID = txtID.getText();
        String P_Name = txtName.getText();
        String Gender = txtGender.getText();
        String Age = txtAge.getText();
        int age = Integer.parseInt(Age);
        String Position = txtPosition.getText();
        String Phone = txtPhone.getText();
        int phone = Integer.parseInt(Phone);
        String Address = txtAddress.getText();
        String City = txtCity.getText();
        String Zip = txtZip.getText();
        int zip = Integer.parseInt(Zip);


        String search_name="{call Patient_Update(?,?,?,?,?,?,?,?,?,?)}";
        //String search_name1="{call PatientDoc_Update(?,?,?,?,?,?,?,?,?,?,?)}";
        connectionDB();

        try{
            cstmt = conn.prepareCall(search_name);
            cstmt.setString(1, P_ID);
            cstmt.setString(2, P_Name);
            cstmt.setString(3, Gender);
            cstmt.setInt(4, age);
            cstmt.setString(5, Position);
```



```
            cstmt.setInt(6, phone);
            cstmt.setString(7, Address);
            cstmt.setString(8, City);
            cstmt.setInt(9, zip);
            cstmt.registerOutParameter(10, OracleTypes.CURSOR);
            cstmt.executeUpdate();

            rs = (ResultSet) cstmt.getObject(10);

        }
    catch(SQLException e){
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "error in sql"+e);
            new Patient_Details().setVisible(true);
    }finally{
            if(cstmt!=null){
                try {
                    cstmt.close();
                } catch (SQLException ex) {
                    Logger.getLogger(Patient_Details.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        }
```

Output:



PATIENTS DETAILS

Patient ID: P1

Eg:It should be in form of P1

Patient's Name: Krishna Shukla

Gender: Female

Age: 21

Position: Student

Phone: 25631485

Address: Riddhivinay Tower

City: Ahmedabad

Zip: 380013

Enter the Doctor ID: D1

PATIENTS DETAILS

Enter the ID:

Message

The Details are updated succesfully

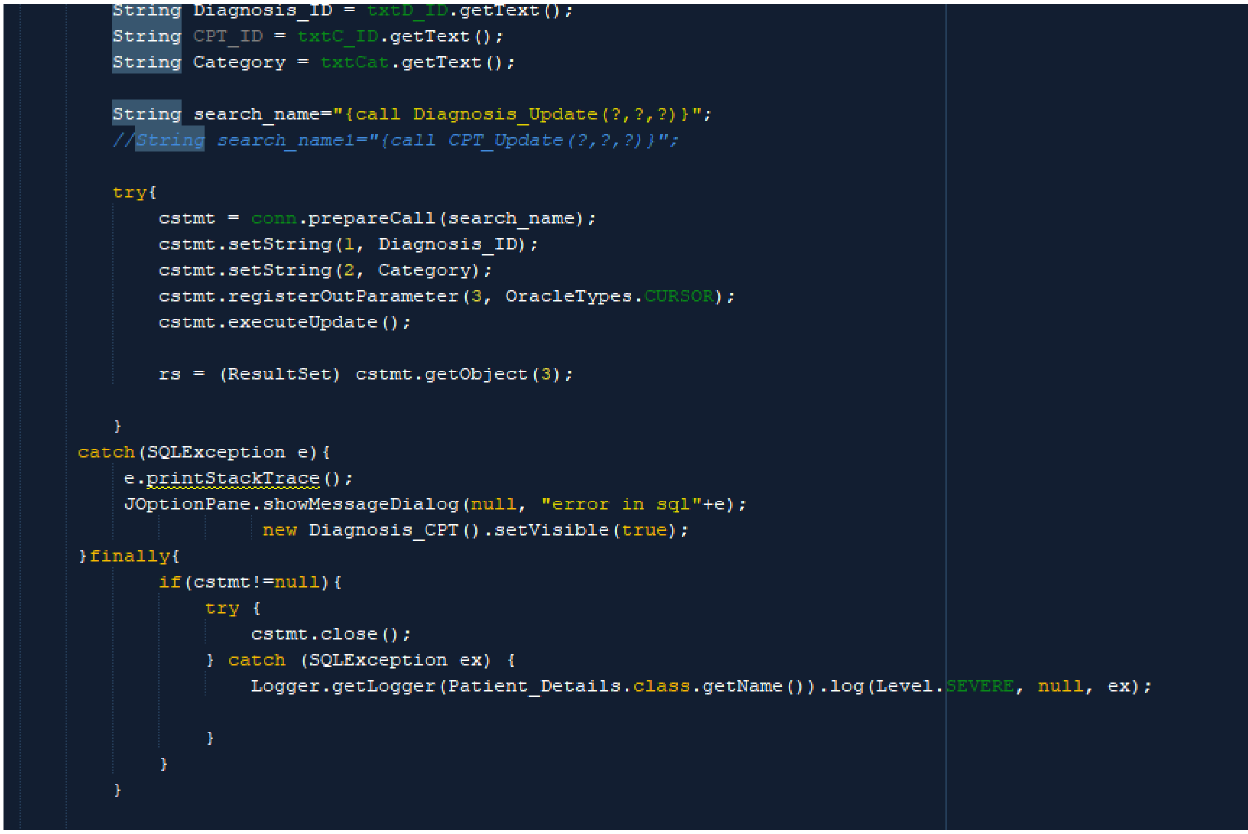
```
ll Patient_Update(?,?,?,?,?,?,?,?,?,?)}
{call PatientDoc_Update(?,?,?,?,?,?,?,?

eCall(search_name);
P_ID);
P_Name);
Gender);

)racleTypes.CURSOR);

tmt.getObject(10);
```

- **Diagnosis Table**

create or replace procedure Diagnosis_Update(x in varchar,y in varchar, c_p out sys_refcursor) as cursor c_d is select * from Diagnosis for update nowait;
r_d c_d%ROWTYPE;
begin
open c_d;
LOOP
FETCH c_d into r_d;
EXIT WHEN c_d%NOTFOUND;
if(r_d.Diagnosis_ID = x) then
update Diagnosis set Category = y where current of c_d;

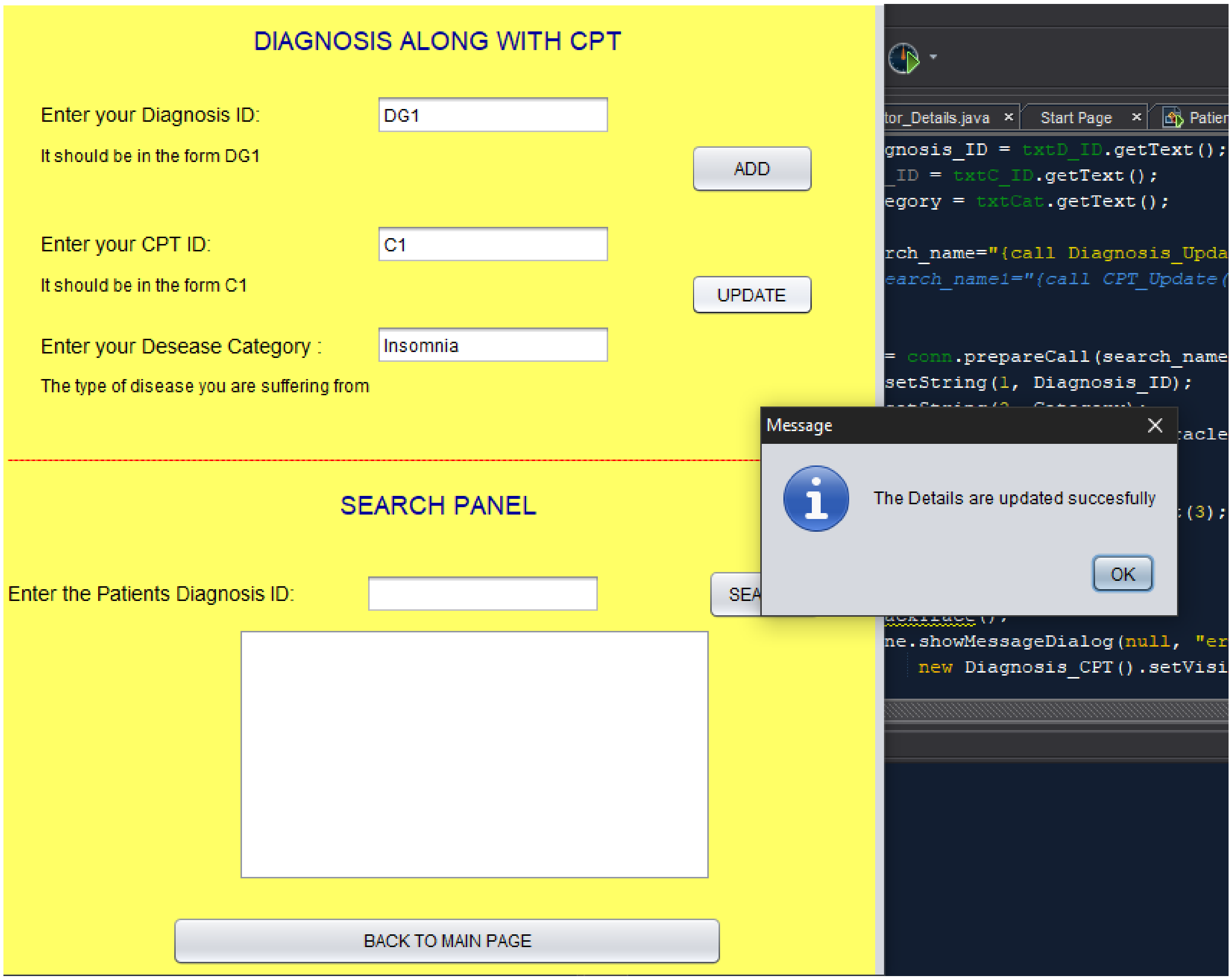
```
end if;
end loop;
close c_d;
end;
```

Calling the Procedure in Java



```
        String Diagnosis_ID = txtD_ID.getText();
        String CPT_ID = txtC_ID.getText();
        String Category = txtCat.getText();

        String search_name="{call Diagnosis_Update(?,?,?)}";
        //String search_name1="{call CPT_Update(?,?,?)}";

        try{
            cstmt = conn.prepareCall(search_name);
            cstmt.setString(1, Diagnosis_ID);
            cstmt.setString(2, Category);
            cstmt.registerOutParameter(3, OracleTypes.CURSOR);
            cstmt.executeUpdate();

            rs = (ResultSet) cstmt.getObject(3);

        }
     catch(SQLException e){
         e.printStackTrace();
         JOptionPane.showMessageDialog(null, "error in sql"+e);
                     new Diagnosis_CPT().setVisible(true);
     }finally{
            if(cstmt!=null){
                try {
                    cstmt.close();
                } catch (SQLException ex) {
                    Logger.getLogger(Patient_Details.class.getName()).log(Level.SEVERE, null, ex);

                }
            }
        }
```

Output:



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG1

It should be in the form DG1

Enter your CPT ID: C1

It should be in the form C1

Enter your Desease Category : Insomnia

The type of disease you are suffering from

SEARCH PANEL

Enter the Patients Diagnosis ID:

Message

The Details are updated succesfully

```
gnosis_ID = txtD_ID.getText();
_ID = txtC_ID.getText();
egory = txtCat.getText();

rch_name="{call Diagnosis_Upda
earch_name1="{call CPT_Update(


= conn.prepareCall(search_name
setString(1, Diagnosis_ID);
```

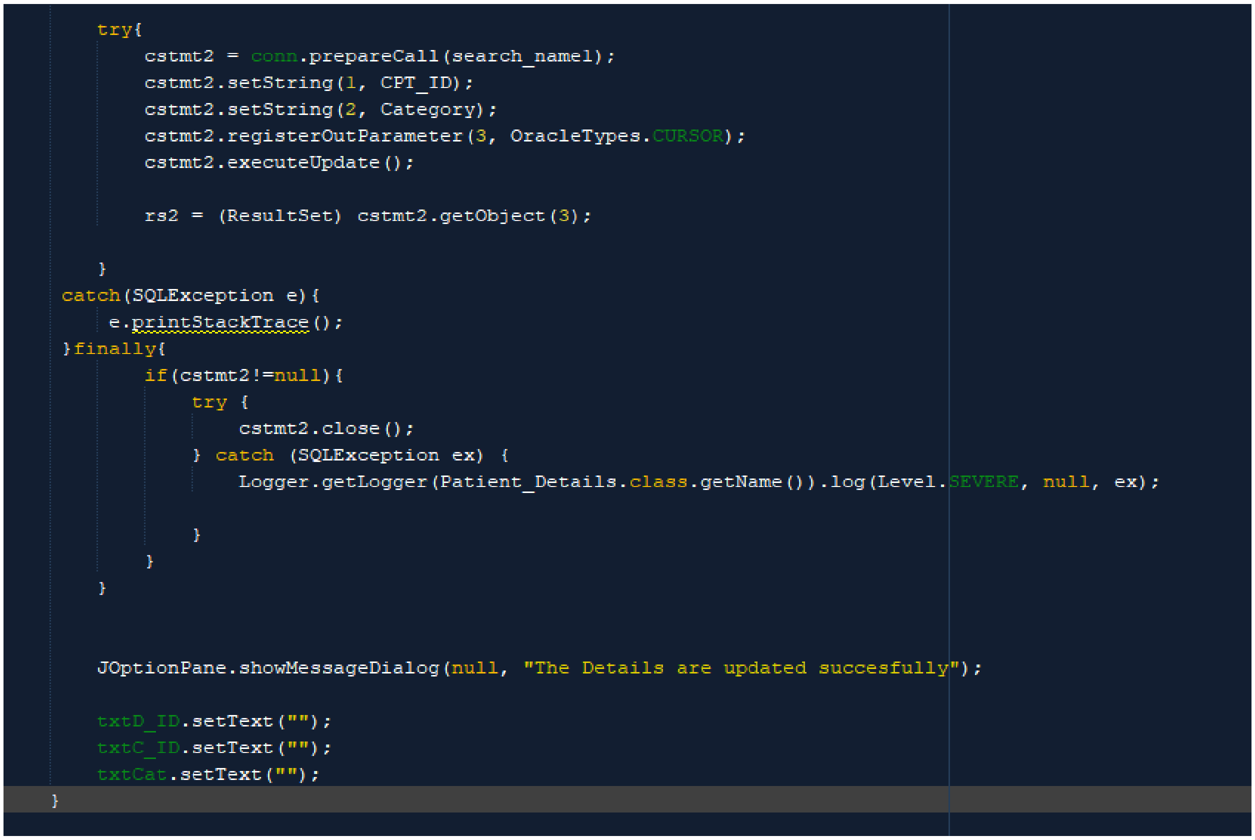
```
ne.showMessageDialog(null, "er
    new Diagnosis_CPT().setVisi
```

- **CPT Table**

create or replace procedure CPT_Update(x in varchar,y in varchar, c_p out sys_refcursor) as cursor c_cp is select * from CPT for update nowait;
r_cp c_cp%ROWTYPE;
begin
open c_cp;
LOOP
FETCH c_cp into r_cp;
EXIT WHEN c_cp%NOTFOUND;
if(r_cp.CPT_ID = x) then
update CPT set Category = y where current of c_cp;

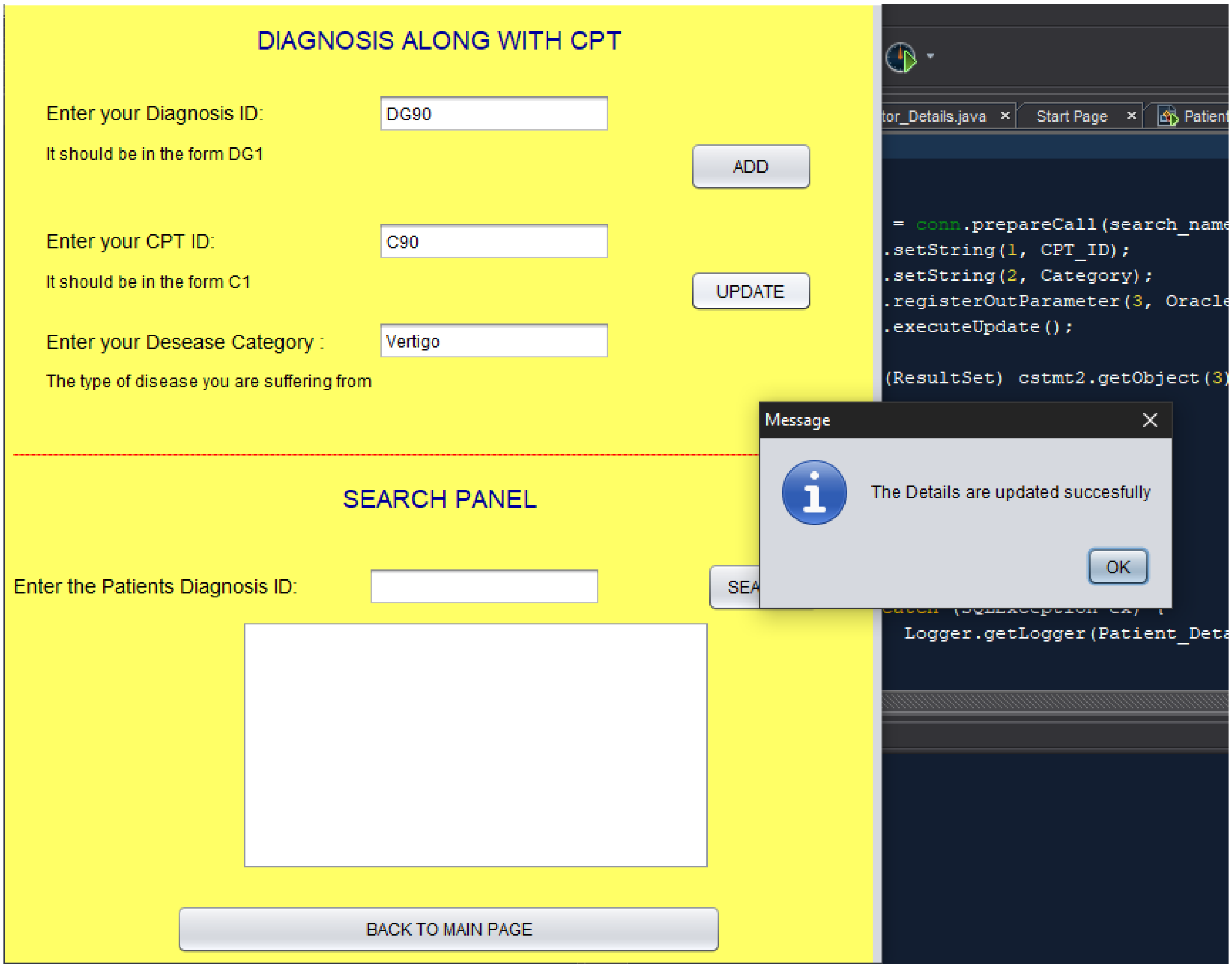
end if;
end loop;
close c_cp;
end;

Calling the Procedure in Java



```
    try{
        cstmt2 = conn.prepareCall(search_name1);
        cstmt2.setString(1, CPT_ID);
        cstmt2.setString(2, Category);
        cstmt2.registerOutParameter(3, OracleTypes.CURSOR);
        cstmt2.executeUpdate();

        rs2 = (ResultSet) cstmt2.getObject(3);

    }
 catch(SQLException e){
     e.printStackTrace();
 }finally{
        if(cstmt2!=null){
            try {
                cstmt2.close();
            } catch (SQLException ex) {
                Logger.getLogger(Patient_Details.class.getName()).log(Level.SEVERE, null, ex);

            }
        }
    }


    JOptionPane.showMessageDialog(null, "The Details are updated succesfully");

    txtD_ID.setText("");
    txtC_ID.setText("");
    txtCat.setText("");
}
```

Output:



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG90

It should be in the form DG1

Enter your CPT ID: C90

It should be in the form C1

Enter your Desease Category : Vertigo

The type of disease you are suffering from

SEARCH PANEL

Enter the Patients Diagnosis ID:

Message

The Details are updated succesfully

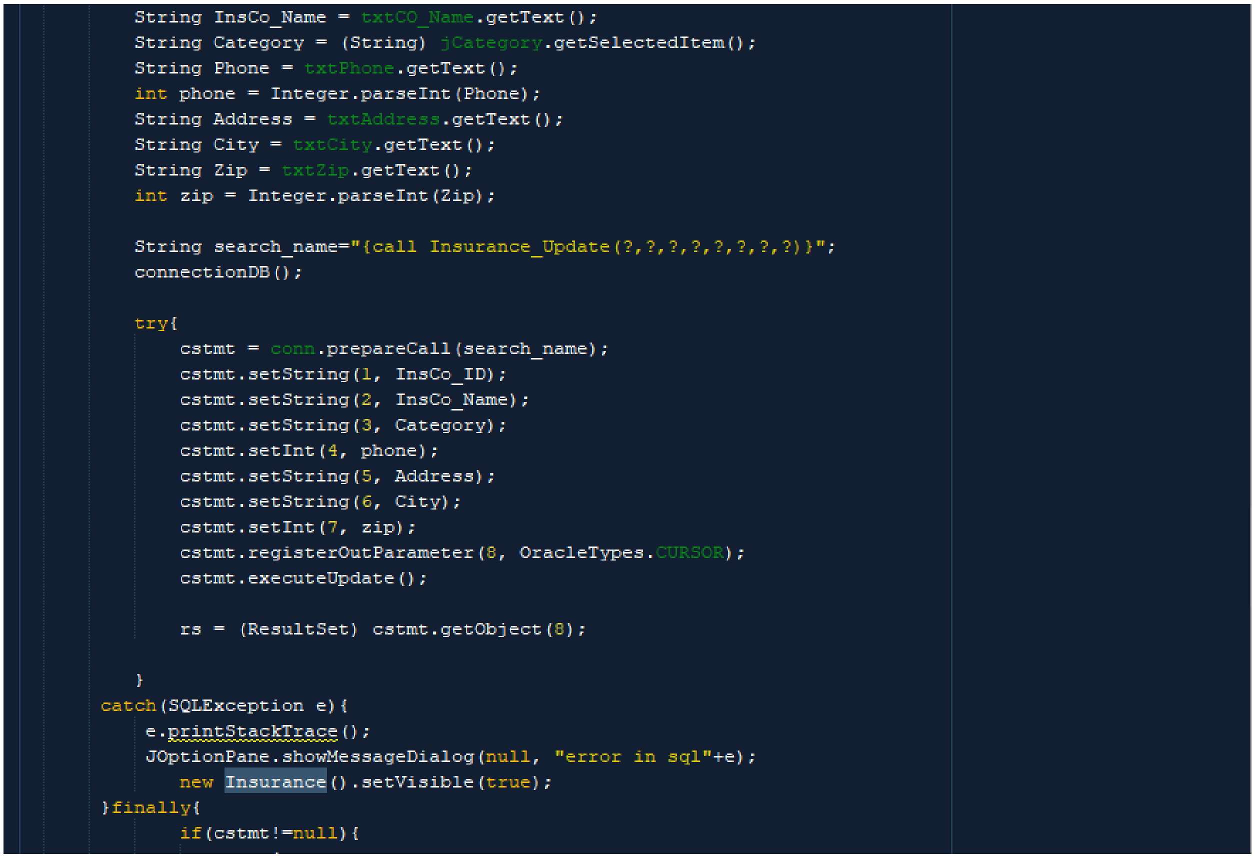
```
 = conn.prepareCall(search_name
.setString(1, CPT_ID);
.setString(2, Category);
.registerOutParameter(3, Oracle
.executeUpdate();

(ResultSet) cstmt2.getObject(3)

  Logger.getLogger(Patient_Deta
```

- **Insurance Table**

create or replace procedure Insurance_Update(x in varchar,y in varchar,z in varchar,a in int,b in varchar,c in varchar,d in int, c_p out sys_refcursor) as
cursor c_i is select * from Insurance for update nowait;
r_i c_i%ROWTYPE;
begin
open c_i;
LOOP

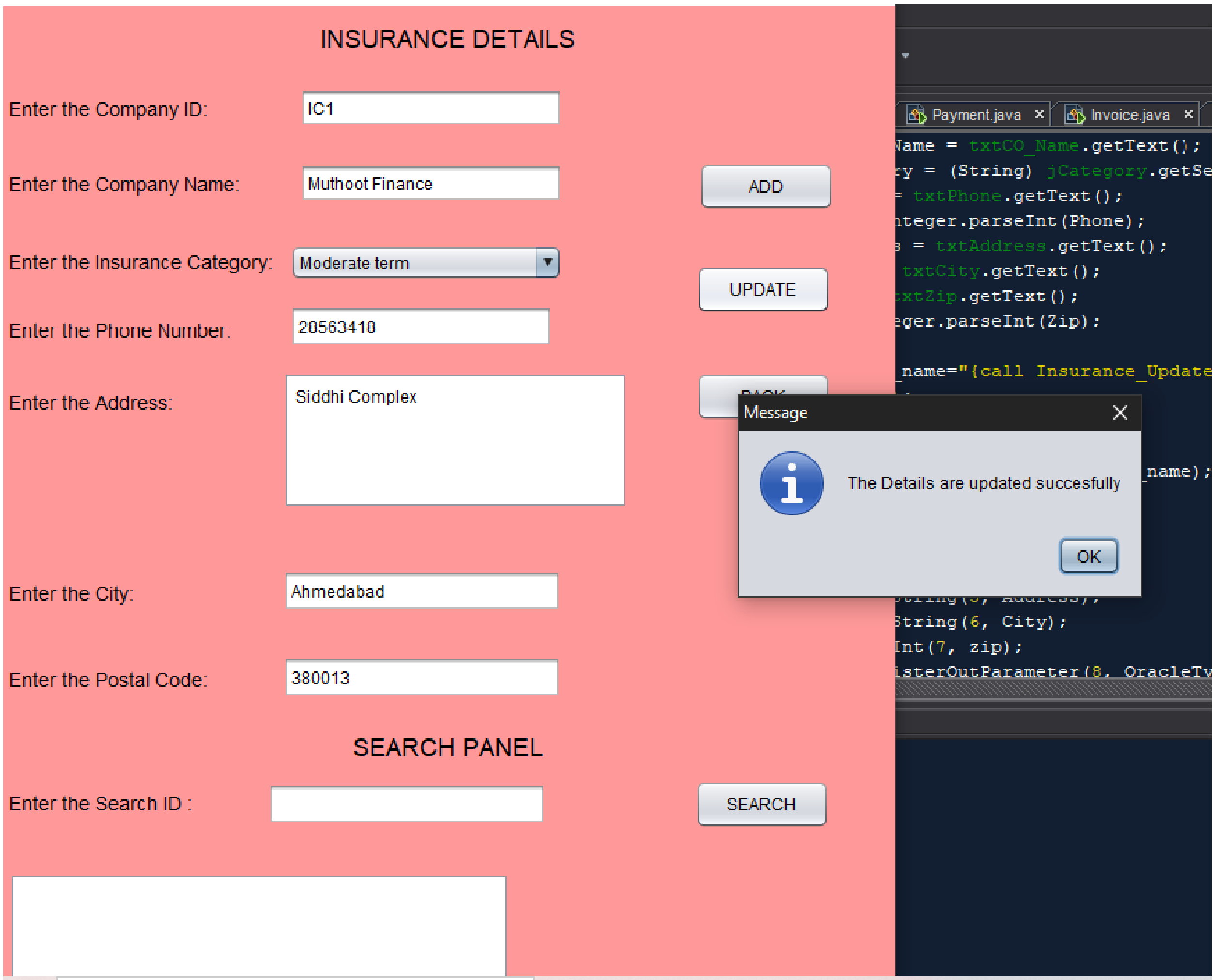
```
FETCH c_i into r_i;
EXIT WHEN c_i%NOTFOUND;
if(r_i.InsCo_ID = x) then
update Insurance set InsCo_Name = y,Category = z,Phone = a,Address =
b,City = c,Zip = d where current of c_i;
end if;
end loop;
close c_i;
End;
```

Calling the Procedure in Java



```
   String InsCo_Name = txtCO_Name.getText();
   String Category = (String) jCategory.getSelectedItem();
   String Phone = txtPhone.getText();
   int phone = Integer.parseInt(Phone);
   String Address = txtAddress.getText();
   String City = txtCity.getText();
   String Zip = txtZip.getText();
   int zip = Integer.parseInt(Zip);

   String search_name="{call Insurance_Update(?,?,?,?,?,?,?,?)}";
   connectionDB();

   try{
       cstmt = conn.prepareCall(search_name);
       cstmt.setString(1, InsCo_ID);
       cstmt.setString(2, InsCo_Name);
       cstmt.setString(3, Category);
       cstmt.setInt(4, phone);
       cstmt.setString(5, Address);
       cstmt.setString(6, City);
       cstmt.setInt(7, zip);
       cstmt.registerOutParameter(8, OracleTypes.CURSOR);
       cstmt.executeUpdate();

       rs = (ResultSet) cstmt.getObject(8);

   }
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "error in sql"+e);
       new Insurance().setVisible(true);
}finally{
       if(cstmt!=null){
```

Output



INSURANCE DETAILS

Enter the Company ID: IC1

Enter the Company Name: Muthoot Finance

Enter the Insurance Category: Moderate term

Enter the Phone Number: 28563418

Enter the Address: Siddhi Complex

Enter the City: Ahmedabad

Enter the Postal Code: 380013

SEARCH PANEL

Enter the Search ID :

Message

The Details are updated succesfully

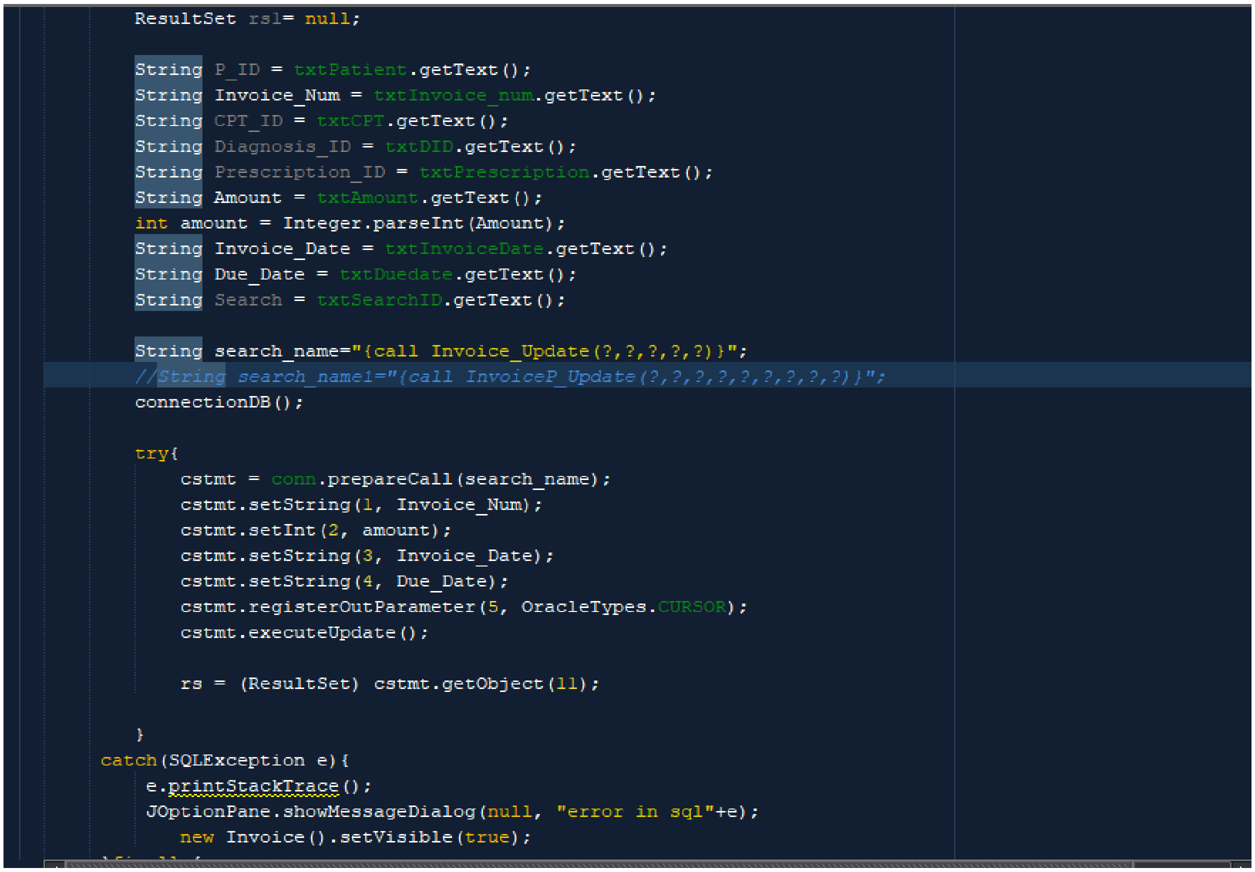
```
Name = txtCO_Name.getText();
ry = (String) jCategory.getSe
= txtPhone.getText();
nteger.parseInt(Phone);
s = txtAddress.getText();
 txtCity.getText();
xtZip.getText();
eger.parseInt(Zip);

_name="{call Insurance_Update

_name);

String(6, City);
Int(7, zip);
isterOutParameter(8, OracleTy
```

- Invoice Table

create or replace procedure Invoice_Update(x in varchar,y in int,z in varchar,a in varchar, c_p out sys_refcursor) as cursor c_iv is select * from Invoice for update nowait;
r_iv c_iv%ROWTYPE;
begin
open c_iv;
LOOP

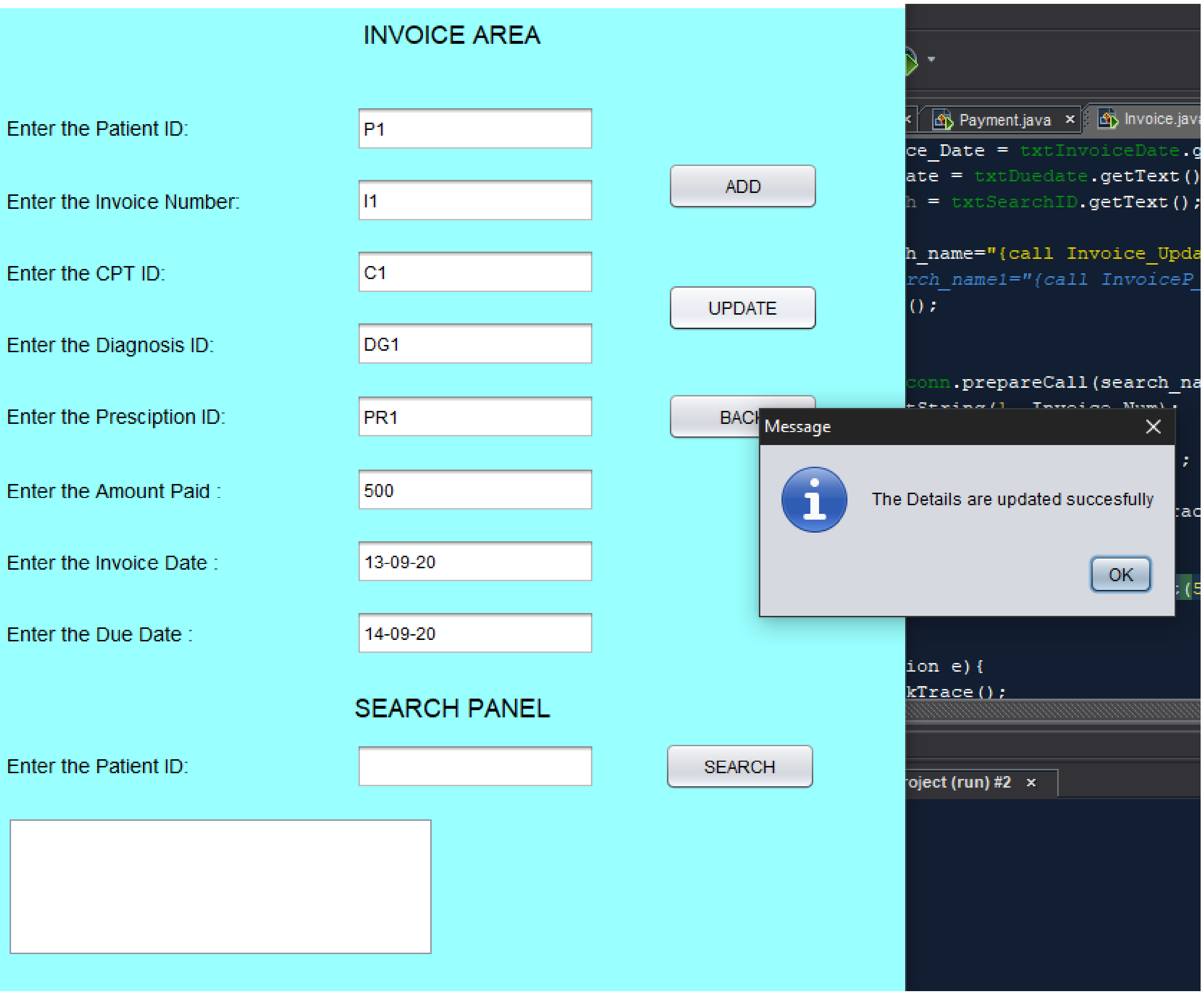
```
FETCH c_iv into r_iv;
EXIT WHEN c_iv%NOTFOUND;
if(r_iv.Invoice_Num = x) then
update Invoice set Amount = y,Invoice_Date = z,Due_Date = a where
current of c_iv;
end if;
end loop;
close c_iv;
end;
```

Calling the Procedure in Java



```
    ResultSet rs1= null;

    String P_ID = txtPatient.getText();
    String Invoice_Num = txtInvoice_num.getText();
    String CPT_ID = txtCPT.getText();
    String Diagnosis_ID = txtDID.getText();
    String Prescription_ID = txtPrescription.getText();
    String Amount = txtAmount.getText();
    int amount = Integer.parseInt(Amount);
    String Invoice_Date = txtInvoiceDate.getText();
    String Due_Date = txtDuedate.getText();
    String Search = txtSearchID.getText();

    String search_name="{call Invoice_Update(?,?,?,?,?)}";
    //String search_name1="{call InvoiceP_Update(?,?,?,?,?,?,?,?,?)}";
    connectionDB();

    try{
        cstmt = conn.prepareCall(search_name);
        cstmt.setString(1, Invoice_Num);
        cstmt.setInt(2, amount);
        cstmt.setString(3, Invoice_Date);
        cstmt.setString(4, Due_Date);
        cstmt.registerOutParameter(5, OracleTypes.CURSOR);
        cstmt.executeUpdate();

        rs = (ResultSet) cstmt.getObject(11);

    }
catch(SQLException e){
     e.printStackTrace();
     JOptionPane.showMessageDialog(null, "error in sql"+e);
        new Invoice().setVisible(true);
```

Output:



Enter the Patient ID: P1

Enter the Invoice Number: I1

Enter the CPT ID: C1

Enter the Diagnosis ID: DG1

Enter the Presciption ID: PR1

Enter the Amount Paid : 500

Enter the Invoice Date : 13-09-20

Enter the Due Date : 14-09-20

Enter the Patient ID:

The Details are updated succesfully

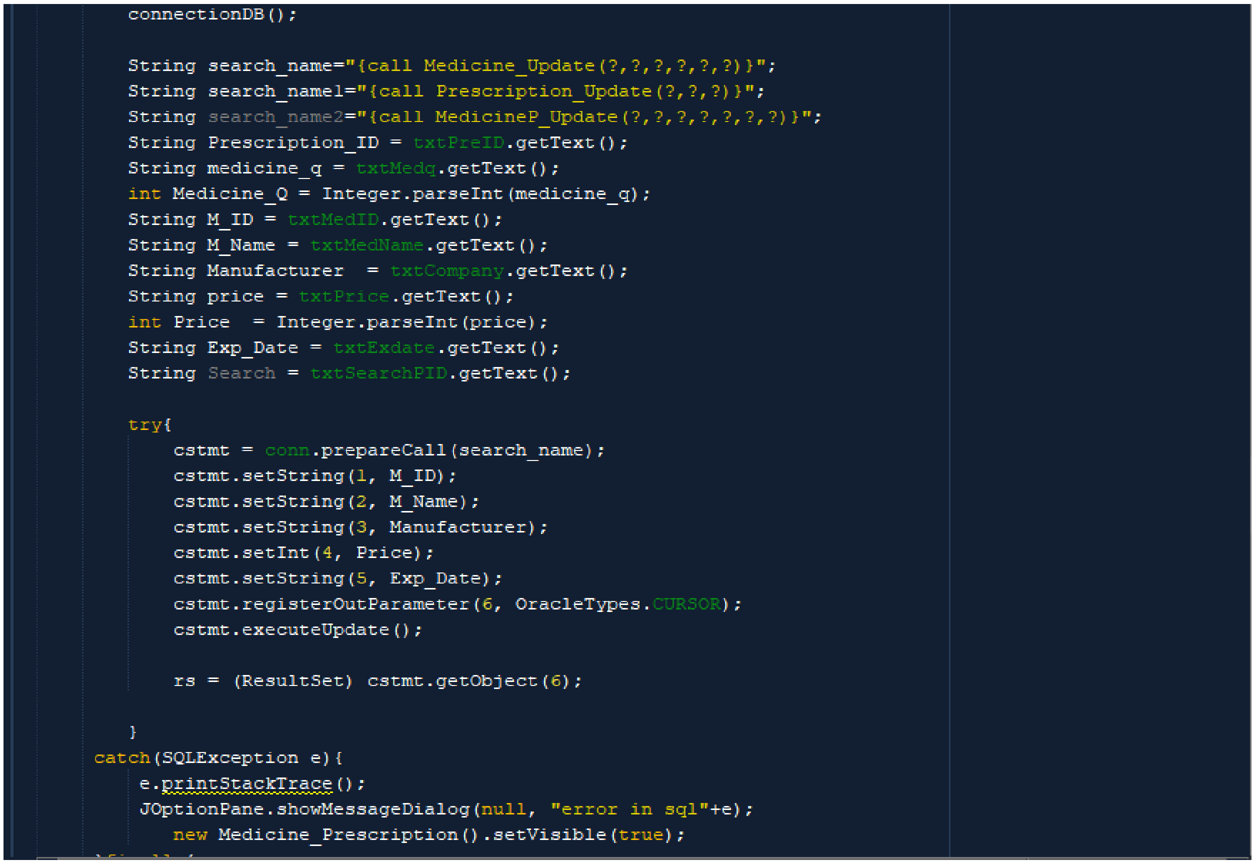
```
ce_Date = txtInvoiceDate.g
ate = txtDuedate.getText()
h = txtSearchID.getText();

h_name="{call Invoice_Upda
rch_name1="{call InvoiceP_
();


conn.prepareCall(search_na

ion e){
kTrace();
```

- **Medicine Table**

create or replace procedure Medicine_Update(x in varchar,y in varchar,z in varchar,a in int,b in varchar, c_p out sys_refcursor) as cursor c_m is select * from Medicine for update nowait;
r_m c_m%ROWTYPE;
begin
open c_m;
LOOP
FETCH c_m into r_m;

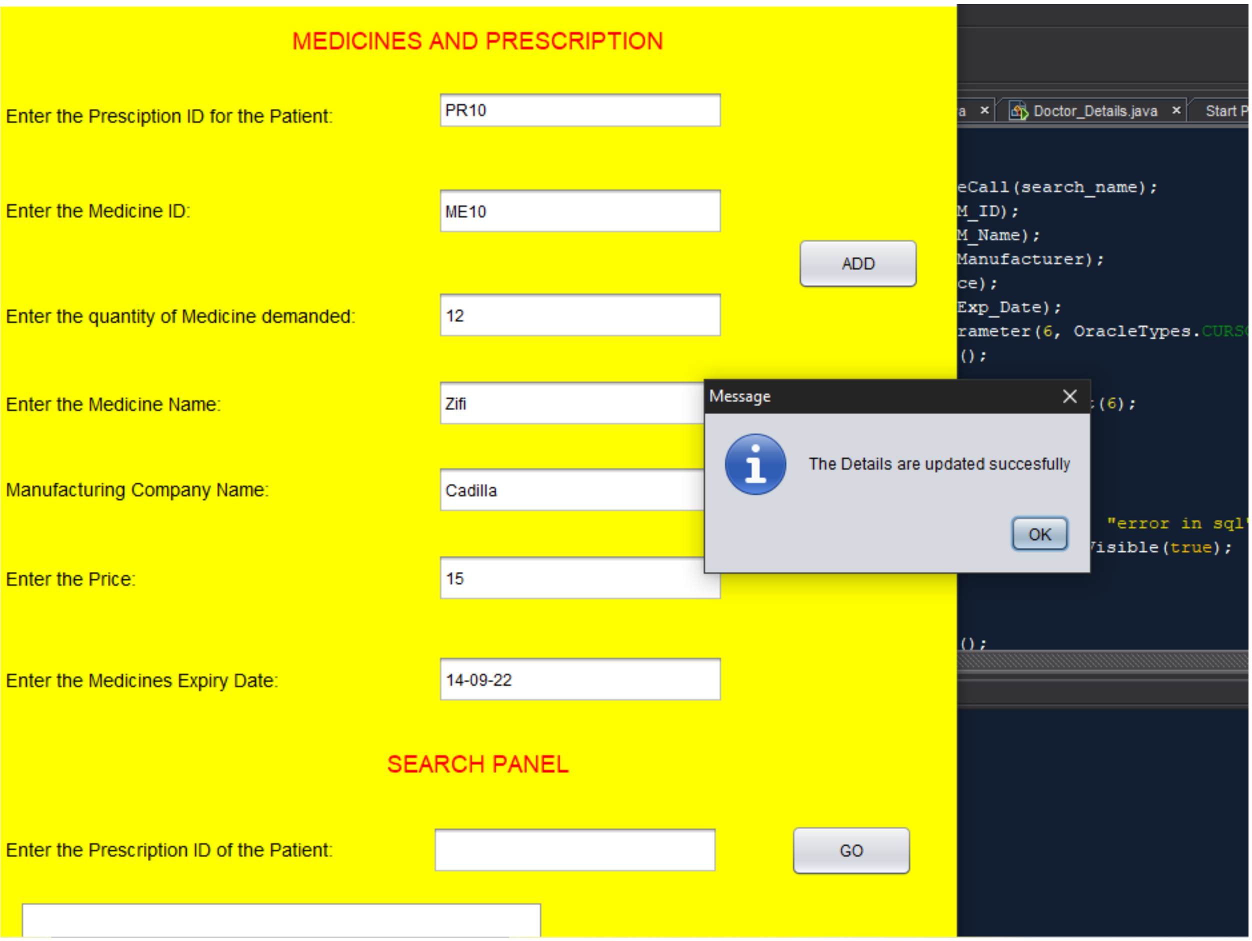
```
EXIT WHEN c_m%NOTFOUND;
if(r_m.M_ID = x) then
update Medicine set M_Name = y,Manufacturer = z,Price = a,Exp_Date = b
where current of c_m;
end if;
end loop;
close c_m;
end;
```

Calling the Procedure in Java



```
        connectionDB();

        String search_name="{call Medicine_Update(?,?,?,?,?,?)}";
        String search_name1="{call Prescription_Update(?,?,?)}";
        String search_name2="{call MedicineP_Update(?,?,?,?,?,?,?)}";
        String Prescription_ID = txtPreID.getText();
        String medicine_q = txtMedq.getText();
        int Medicine_Q = Integer.parseInt(medicine_q);
        String M_ID = txtMedID.getText();
        String M_Name = txtMedName.getText();
        String Manufacturer  = txtCompany.getText();
        String price = txtPrice.getText();
        int Price  = Integer.parseInt(price);
        String Exp_Date = txtExdate.getText();
        String Search = txtSearchPID.getText();

        try{
            cstmt = conn.prepareCall(search_name);
            cstmt.setString(1, M_ID);
            cstmt.setString(2, M_Name);
            cstmt.setString(3, Manufacturer);
            cstmt.setInt(4, Price);
            cstmt.setString(5, Exp_Date);
            cstmt.registerOutParameter(6, OracleTypes.CURSOR);
            cstmt.executeUpdate();

            rs = (ResultSet) cstmt.getObject(6);

        }
     catch(SQLException e){
         e.printStackTrace();
         JOptionPane.showMessageDialog(null, "error in sql"+e);
            new Medicine_Prescription().setVisible(true);
```

Output:



MEDICINES AND PRESCRIPTION

Enter the Presciption ID for the Patient: PR10

Enter the Medicine ID: ME10

Enter the quantity of Medicine demanded: 12

Enter the Medicine Name: Zifi

Manufacturing Company Name: Cadilla

Enter the Price: 15

Enter the Medicines Expiry Date: 14-09-22

SEARCH PANEL

Enter the Prescription ID of the Patient:

Message

The Details are updated succesfully

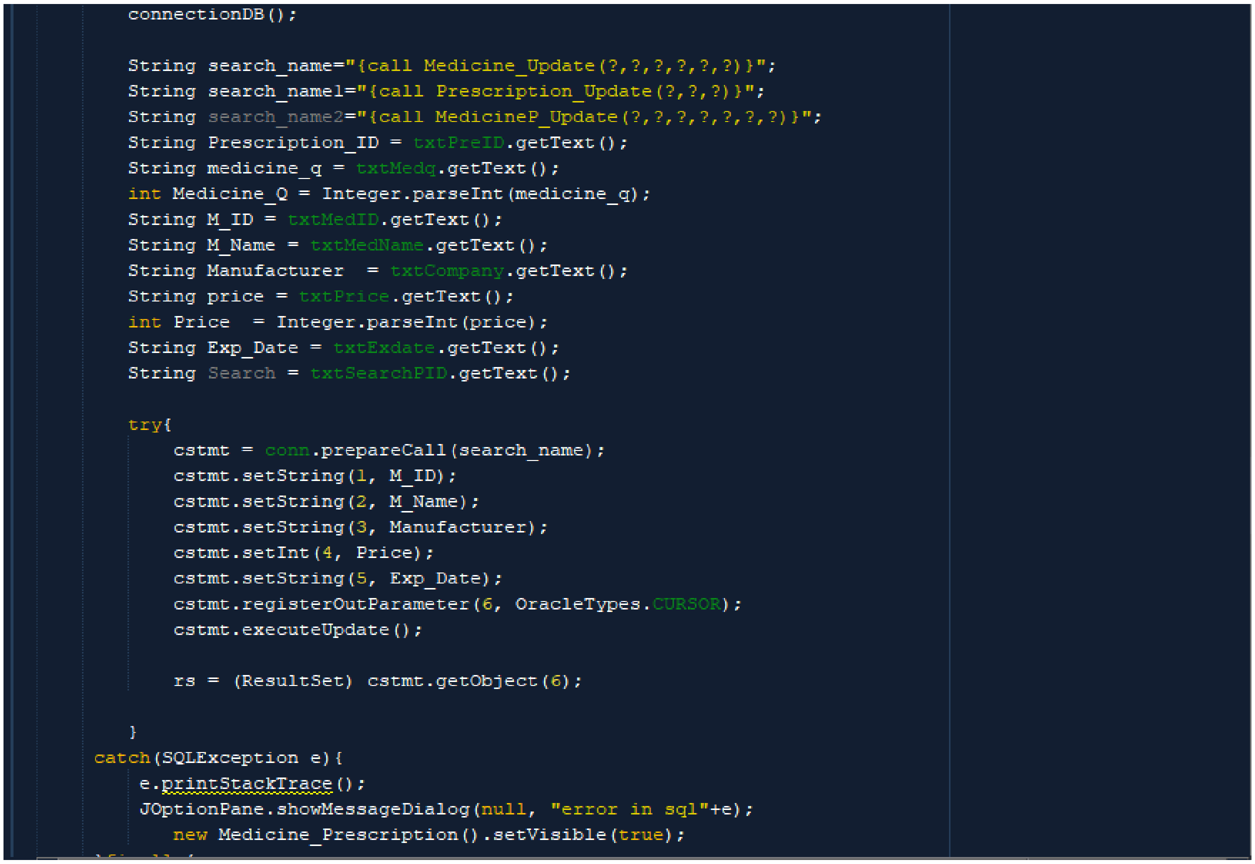
```
eCall(search_name);
M_ID);
M_Name);
Manufacturer);
ce);
Exp_Date);
rameter(6, OracleTypes.CURS
();
:(6);
 "error in sql
Visible(true);
();
```

- **Prescription Table**

create or replace procedure Prescription_Update(x in varchar,y in int, c_p out sys_refcursor) as cursor c_pre is select * from Prescription for update nowait;
r_pre c_pre%ROWTYPE;
begin
open c_pre;
LOOP
FETCH c_pre into r_pre;
EXIT WHEN c_pre%NOTFOUND;

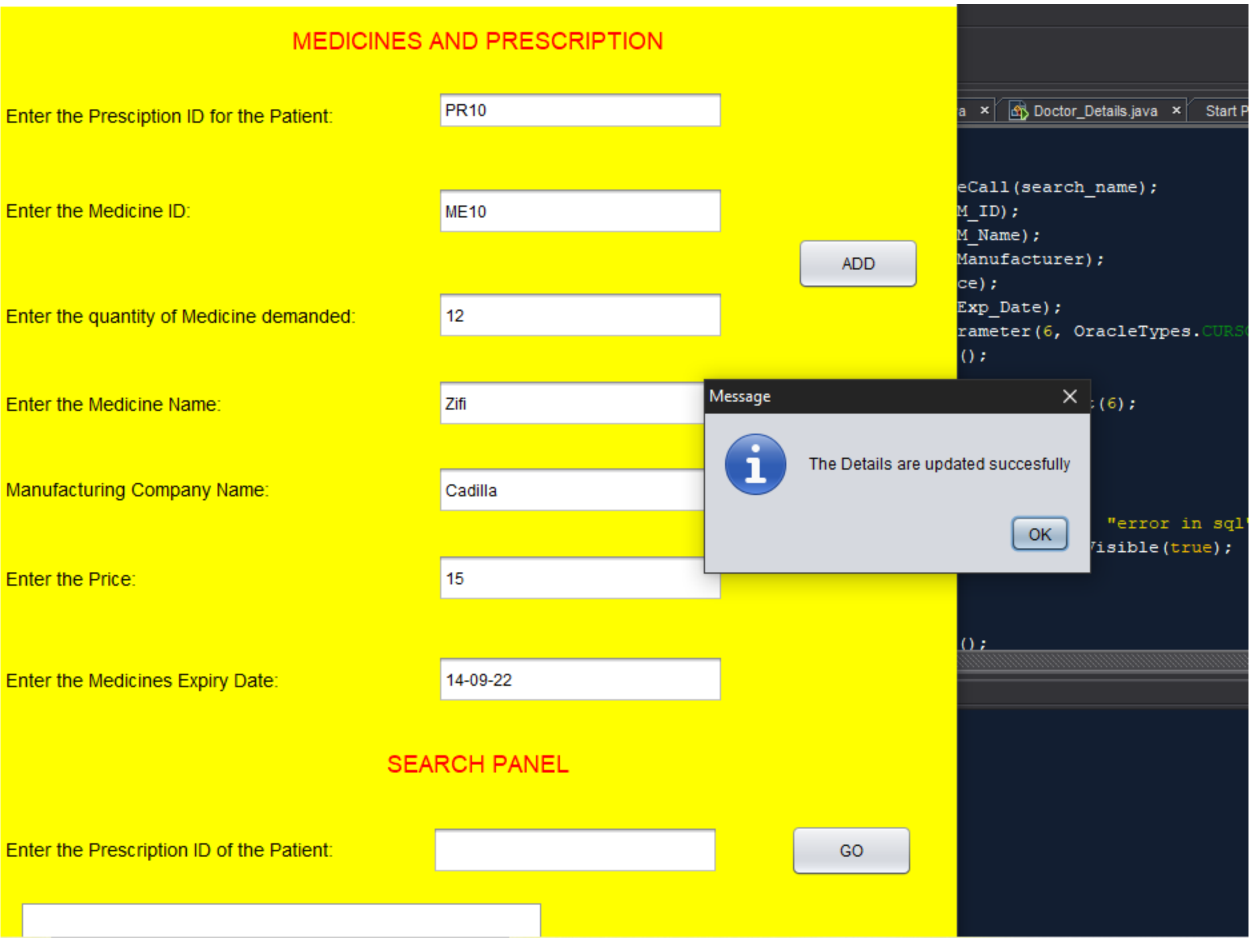
```
if(r_pre.Prescription_ID = x) then
update Prescription set Medicine_Q = y where current of c_pre;
end if;
end loop;
close c_pre;
end;
```

Calling the Procedure in Java



```
   connectionDB();

   String search_name="{call Medicine_Update(?,?,?,?,?,?)}";
   String search_name1="{call Prescription_Update(?,?,?)}";
   String search_name2="{call MedicineP_Update(?,?,?,?,?,?,?)}";
   String Prescription_ID = txtPreID.getText();
   String medicine_q = txtMedq.getText();
   int Medicine_Q = Integer.parseInt(medicine_q);
   String M_ID = txtMedID.getText();
   String M_Name = txtMedName.getText();
   String Manufacturer  = txtCompany.getText();
   String price = txtPrice.getText();
   int Price  = Integer.parseInt(price);
   String Exp_Date = txtExdate.getText();
   String Search = txtSearchPID.getText();

   try{
       cstmt = conn.prepareCall(search_name);
       cstmt.setString(1, M_ID);
       cstmt.setString(2, M_Name);
       cstmt.setString(3, Manufacturer);
       cstmt.setInt(4, Price);
       cstmt.setString(5, Exp_Date);
       cstmt.registerOutParameter(6, OracleTypes.CURSOR);
       cstmt.executeUpdate();

       rs = (ResultSet) cstmt.getObject(6);

   }
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "error in sql"+e);
       new Medicine_Prescription().setVisible(true);
}finally{
```

Output:



MEDICINES AND PRESCRIPTION

Enter the Presciption ID for the Patient: PR10

Enter the Medicine ID: ME10

Enter the quantity of Medicine demanded: 12

Enter the Medicine Name: Zifi

Manufacturing Company Name: Cadilla

Enter the Price: 15

Enter the Medicines Expiry Date: 14-09-22

SEARCH PANEL

Enter the Prescription ID of the Patient:

The Details are updated succesfully

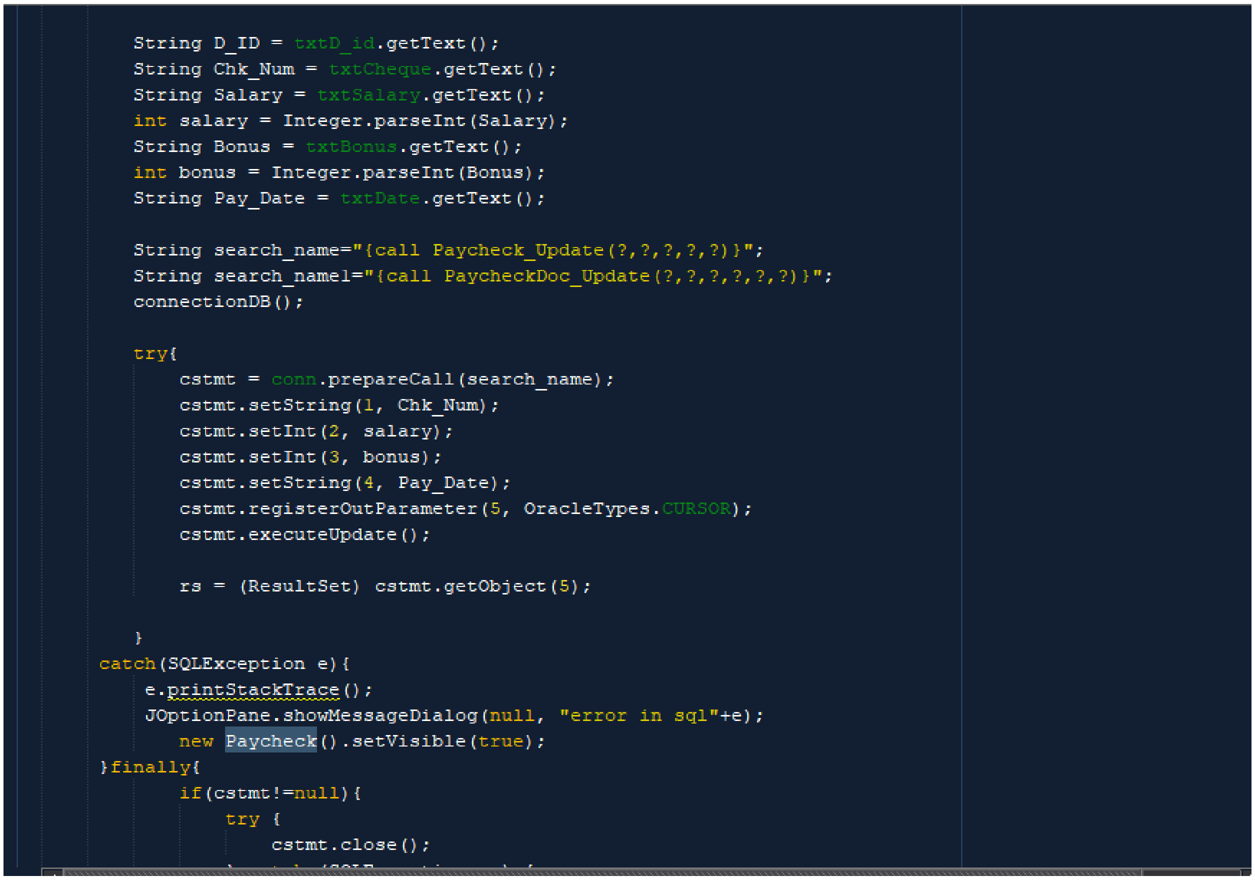
```
eCall(search_name);
M_ID);
M_Name);
Manufacturer);
ce);
Exp_Date);
rameter(6, OracleTypes.CURS
();
:(6);
 "error in sql
Visible(true);
();
```

- **Paycheck Table**

create or replace procedure Paycheck_Update(x in varchar,y in int,z in int,a in varchar, c_p out sys_refcursor) as cursor c_pay is select * from Paycheck for update nowait;
r_pay c_pay%ROWTYPE;
begin
open c_pay;
LOOP
FETCH c_pay into r_pay;
EXIT WHEN c_pay%NOTFOUND;

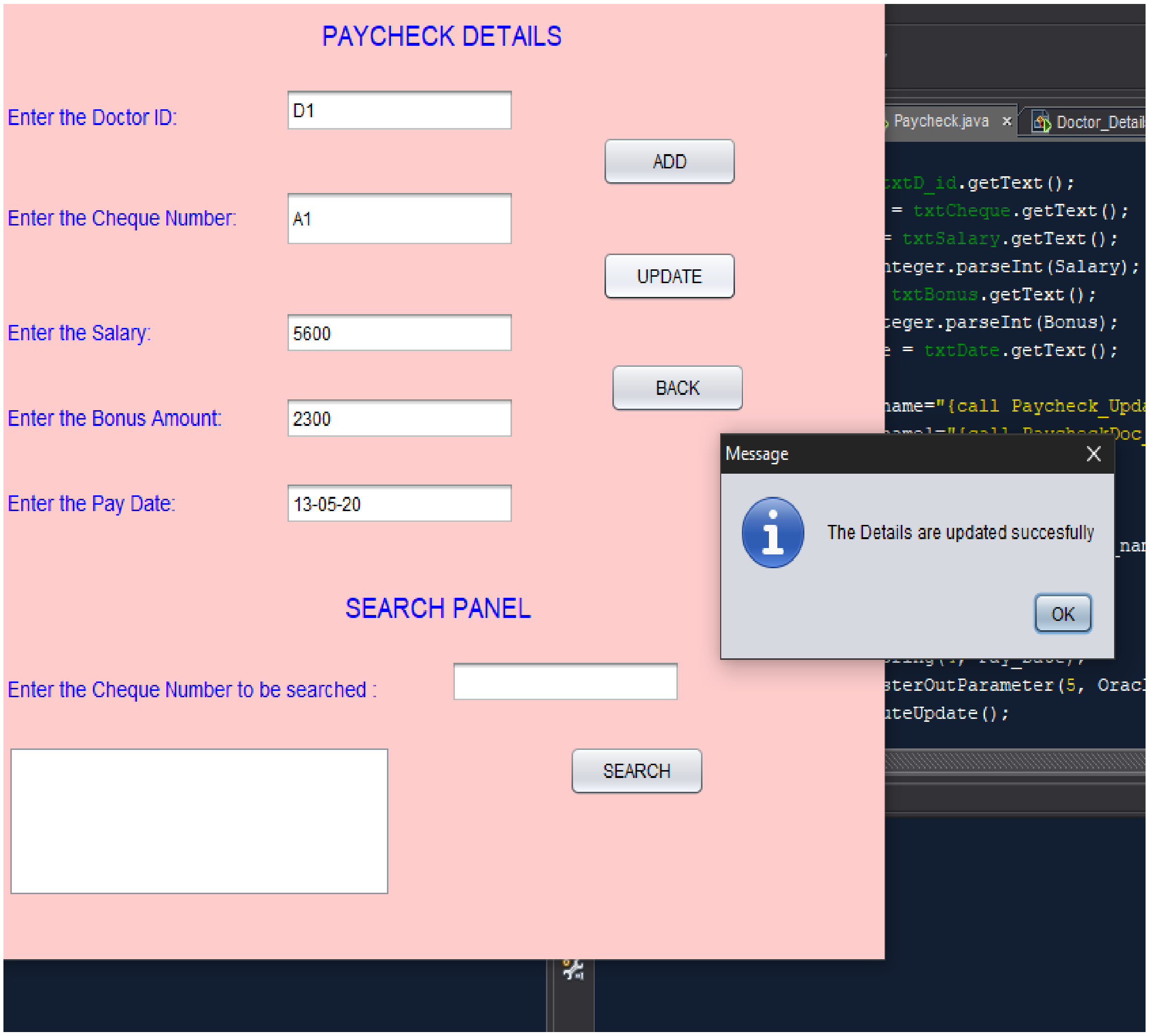
if(r_pay.Chk_Num = x) then
update Paycheck set Salary = y,Bonus = z,Pay_Date = a where current of c_pay;
end if;
end loop;
close c_pay;
end;

Calling the Procedure in Java



```
   String D_ID = txtD_id.getText();
   String Chk_Num = txtCheque.getText();
   String Salary = txtSalary.getText();
   int salary = Integer.parseInt(Salary);
   String Bonus = txtBonus.getText();
   int bonus = Integer.parseInt(Bonus);
   String Pay_Date = txtDate.getText();

   String search_name="{call Paycheck_Update(?,?,?,?,?)}";
   String search_name1="{call PaycheckDoc_Update(?,?,?,?,?,?)}";
   connectionDB();

   try{
       cstmt = conn.prepareCall(search_name);
       cstmt.setString(1, Chk_Num);
       cstmt.setInt(2, salary);
       cstmt.setInt(3, bonus);
       cstmt.setString(4, Pay_Date);
       cstmt.registerOutParameter(5, OracleTypes.CURSOR);
       cstmt.executeUpdate();

       rs = (ResultSet) cstmt.getObject(5);

   }
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "error in sql"+e);
       new Paycheck().setVisible(true);
}finally{
       if(cstmt!=null){
           try {
               cstmt.close();
```

Output



PAYCHECK DETAILS

Enter the Doctor ID: D1

Enter the Cheque Number: A1

Enter the Salary: 5600

Enter the Bonus Amount: 2300

Enter the Pay Date: 13-05-20

SEARCH PANEL

Enter the Cheque Number to be searched :

Message

The Details are updated succesfully

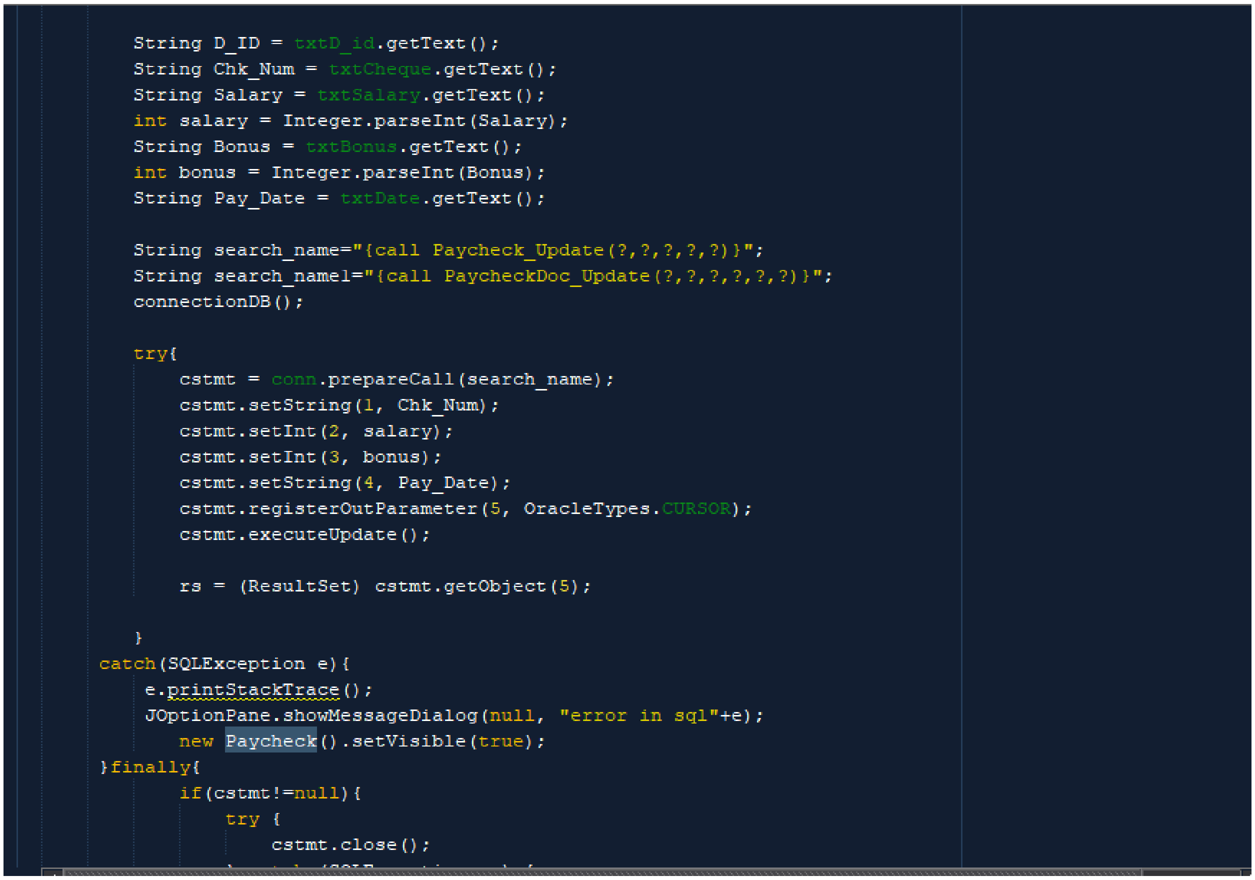
```
xtD_id.getText();
 = txtCheque.getText();
= txtSalary.getText();
nteger.parseInt(Salary);
 txtBonus.getText();
teger.parseInt(Bonus);
e = txtDate.getText();

name="{call Paycheck_Upda
                          _nar
sterOutParameter(5, Oracl
uteUpdate();
```

- **Paycheck_Doc** Table

create or replace procedure PaycheckDoc_Update(x in varchar,y in varchar,z in int,a in int,b in varchar, c_p out sys_refcursor) as cursor c_payd is select * from Paycheck_Doc for update nowait;
r_payd c_payd%ROWTYPE;
begin
open c_payd;
LOOP

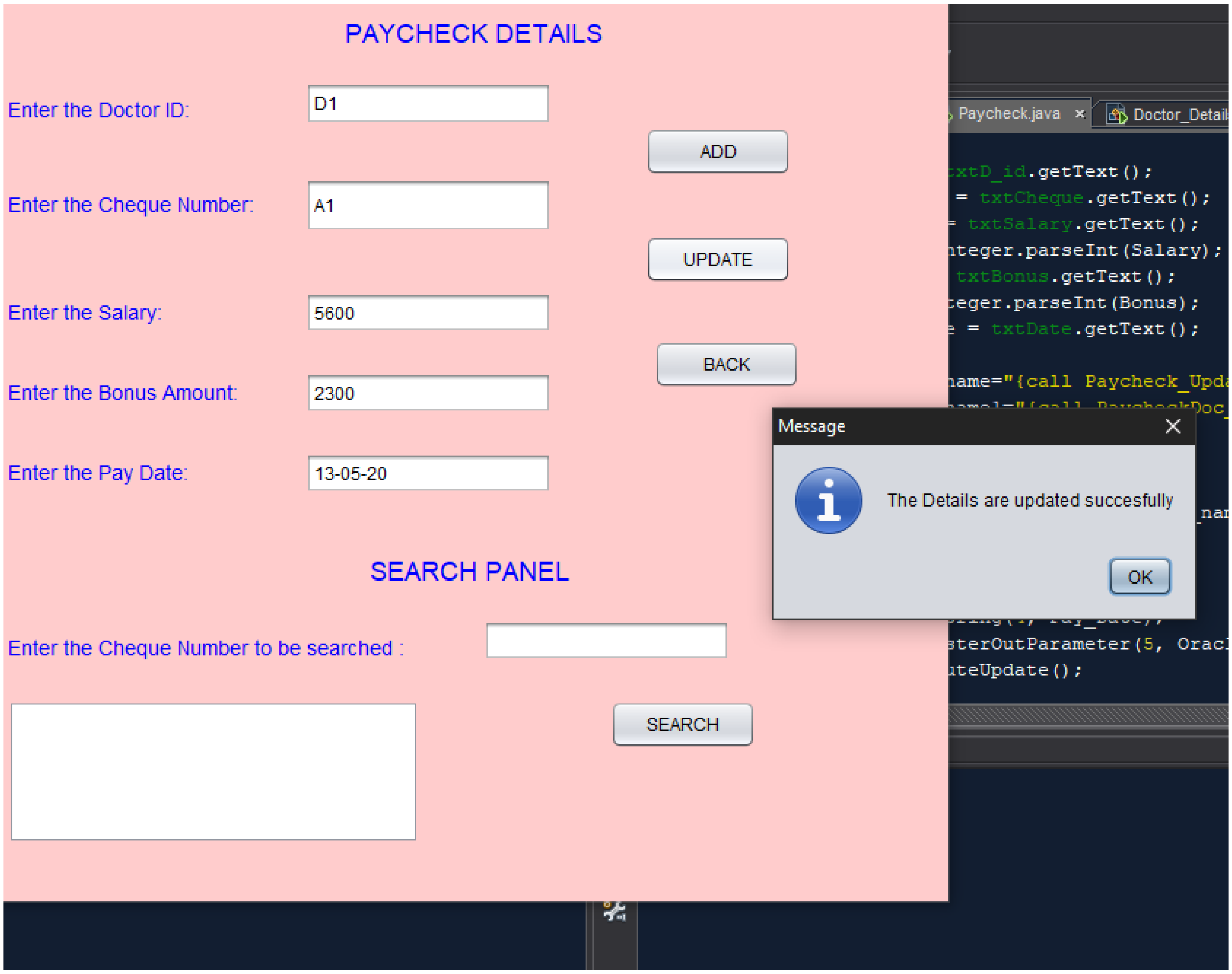
FETCH c_payd into r_payd;
EXIT WHEN c_payd%NOTFOUND;
if(r_payd.Chk_Num = x) then
update Paycheck_Doc set D_Id = y,Salary = z,Bonus = a,Pay_Date = b
where current of c_payd;
end if;
end loop;
close c_payd;
end;

Calling the Procedure in Java



```
       String D_ID = txtD_id.getText();
       String Chk_Num = txtCheque.getText();
       String Salary = txtSalary.getText();
       int salary = Integer.parseInt(Salary);
       String Bonus = txtBonus.getText();
       int bonus = Integer.parseInt(Bonus);
       String Pay_Date = txtDate.getText();

       String search_name="{call Paycheck_Update(?,?,?,?,?)}";
       String search_name1="{call PaycheckDoc_Update(?,?,?,?,?,?)}";
       connectionDB();

       try{
           cstmt = conn.prepareCall(search_name);
           cstmt.setString(1, Chk_Num);
           cstmt.setInt(2, salary);
           cstmt.setInt(3, bonus);
           cstmt.setString(4, Pay_Date);
           cstmt.registerOutParameter(5, OracleTypes.CURSOR);
           cstmt.executeUpdate();

           rs = (ResultSet) cstmt.getObject(5);

       }
    catch(SQLException e){
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "error in sql"+e);
           new Paycheck().setVisible(true);
    }finally{
           if(cstmt!=null){
               try {
                   cstmt.close();
```

Output



PAYCHECK DETAILS

Enter the Doctor ID: D1

Enter the Cheque Number: A1

Enter the Salary: 5600

Enter the Bonus Amount: 2300

Enter the Pay Date: 13-05-20

SEARCH PANEL

Enter the Cheque Number to be searched :

The Details are updated succesfully

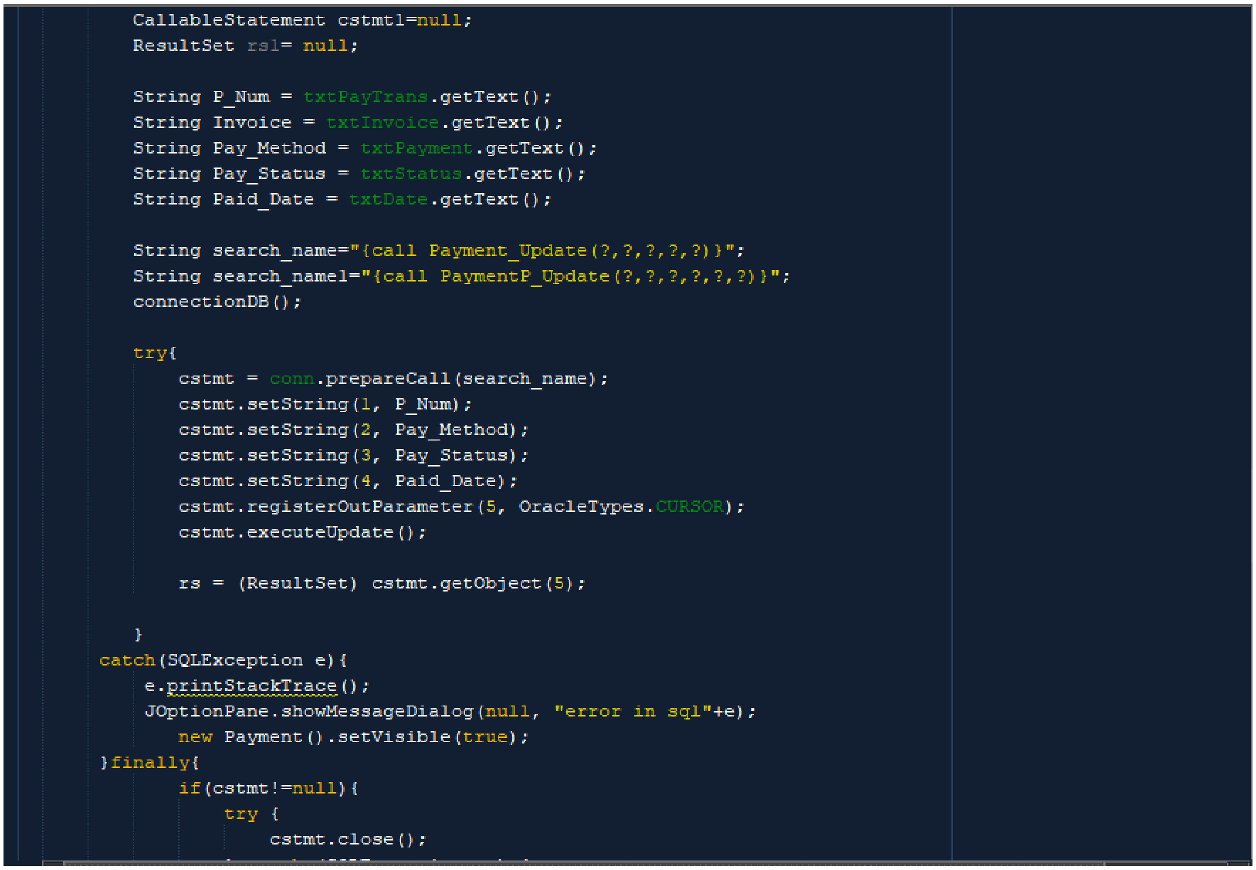
```
xtD_id.getText();
 = txtCheque.getText();
= txtSalary.getText();
nteger.parseInt(Salary);
 txtBonus.getText();
teger.parseInt(Bonus);
e = txtDate.getText();

name="{call Paycheck_Upda
                                 _nar
sterOutParameter(5, Oracl
uteUpdate();
```

- **Payment Table**

create or replace procedure Payment_Update(x in varchar,y in varchar,z in varchar,a in varchar, c_p out sys_refcursor) as cursor c_pt is select * from Payment for update nowait;
r_pt c_pt%ROWTYPE;
begin
open c_pt;
LOOP
FETCH c_pt into r_pt;
EXIT WHEN c_pt%NOTFOUND;

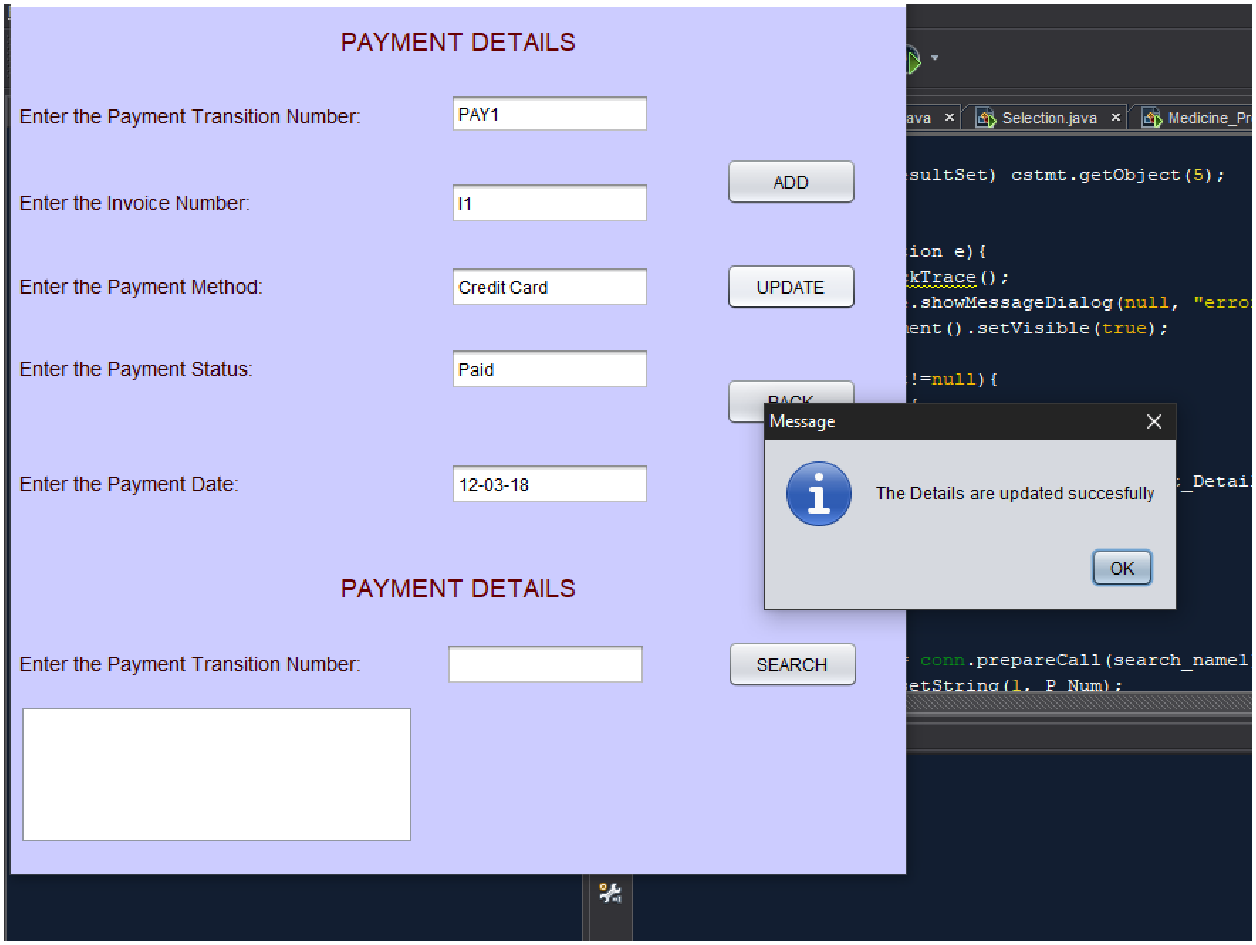
```
if(r_pt.P_Num = x) then
update Payment set Pay_Method = y,Pay_Status = z,Paid_Date = a where
current of c_pt;
end if;
end loop;
close c_pt;
end;
```

Calling the Procedure in Java



```
   CallableStatement cstmtl=null;
   ResultSet rsl= null;

   String P_Num = txtPayTrans.getText();
   String Invoice = txtInvoice.getText();
   String Pay_Method = txtPayment.getText();
   String Pay_Status = txtStatus.getText();
   String Paid_Date = txtDate.getText();

   String search_name="{call Payment_Update(?,?,?,?,?)}";
   String search_namel="{call PaymentP_Update(?,?,?,?,?,?)}";
   connectionDB();

   try{
       cstmt = conn.prepareCall(search_name);
       cstmt.setString(1, P_Num);
       cstmt.setString(2, Pay_Method);
       cstmt.setString(3, Pay_Status);
       cstmt.setString(4, Paid_Date);
       cstmt.registerOutParameter(5, OracleTypes.CURSOR);
       cstmt.executeUpdate();

       rs = (ResultSet) cstmt.getObject(5);

   }
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "error in sql"+e);
       new Payment().setVisible(true);
}finally{
       if(cstmt!=null){
           try {
               cstmt.close();
```

Output:



PAYMENT DETAILS

Enter the Payment Transition Number: PAY1

Enter the Invoice Number: I1

Enter the Payment Method: Credit Card

Enter the Payment Status: Paid

Enter the Payment Date: 12-03-18

Message

The Details are updated succesfully

PAYMENT DETAILS

Enter the Payment Transition Number:

```
sultSet) cstmt.getObject(5);

ion e){
kTrace();
.showMessageDialog(null, "erro
ent().setVisible(true);

!=null){

_Detai

 conn.prepareCall(search_namel
etString(1, P_Num);
```

- **For Searching:**

- **Doctor Table**

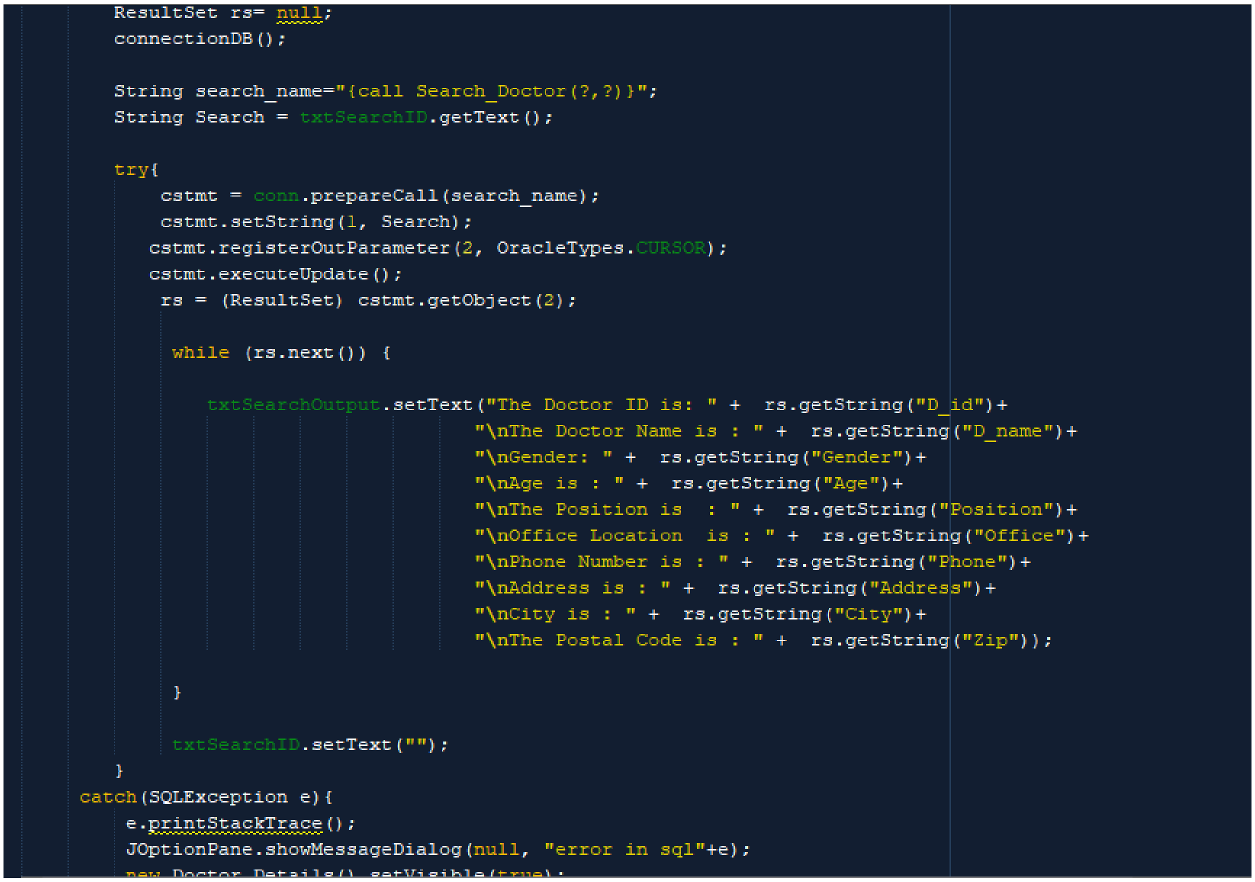
create or replace procedure Doc(id varchar) as
cursor c_doc is select
D_ID,D_name,Gender,Age,Position,Office,Phone,Address,City,Zip
from Doctor;
r_doc c_doc%ROWTYPE;

```
begin
OPEN c_doc;
LOOP
FETCH c_doc into r_doc;
EXIT WHEN c_doc%NOTFOUND;
if(r_doc.D_id=id) then
   dbms_output.put_line('Name of the Doctor is : '|| r_doc.D_name);
   dbms_output.put_line('Gender : '|| r_doc.Gender);
   dbms_output.put_line('Age : '|| r_doc.Age);
   dbms_output.put_line('Position is : '|| r_doc.Position);
   dbms_output.put_line('Office Location : '|| r_doc.Office);
   dbms_output.put_line('Phone : '|| r_doc.Phone);
   dbms_output.put_line('Address : '|| r_doc.Address);
   dbms_output.put_line('City : '|| r_doc.City);
   end if;
end loop;
close c_doc;
end;

declare
Doctor_Id varchar(5):=:id;
begin
Doc(Doctor_Id);
End;
```

For Calling in java a little change in the procedure

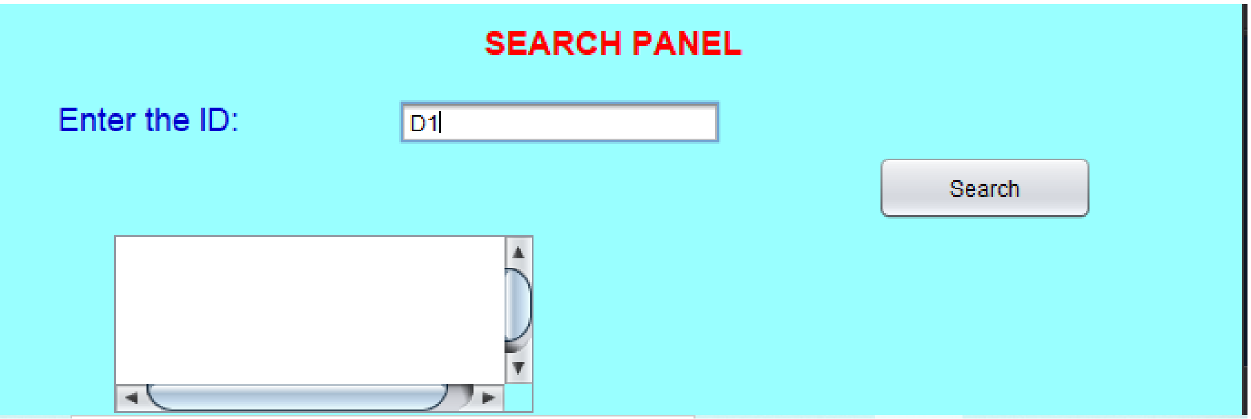
```
create or replace procedure Search_Doctor(x in varchar, c_doc out
sys_refcursor) as
begin
open c_doc for select * from Doctor where D_ID=x;
End;
```

Calling the Procedure in Java



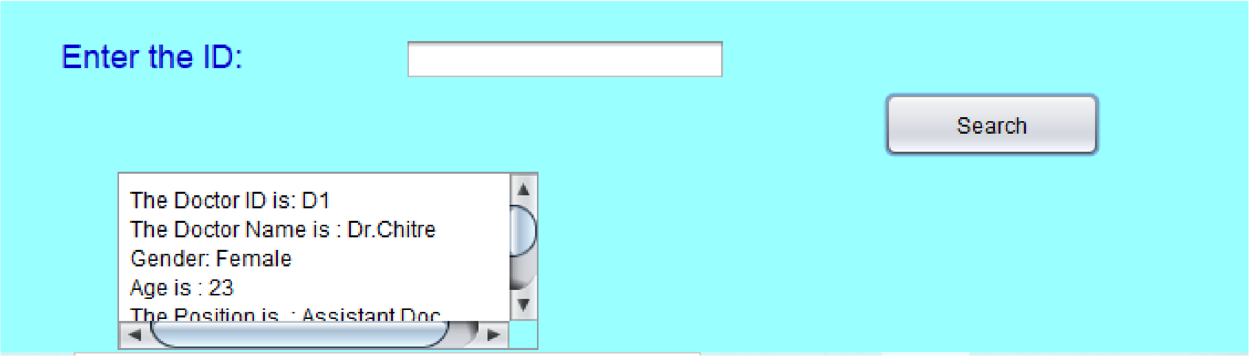
```
        ResultSet rs= null;
        connectionDB();

        String search_name="{call Search_Doctor(?,?)}";
        String Search = txtSearchID.getText();

        try{
            cstmt = conn.prepareCall(search_name);
            cstmt.setString(1, Search);
           cstmt.registerOutParameter(2, OracleTypes.CURSOR);
           cstmt.executeUpdate();
            rs = (ResultSet) cstmt.getObject(2);

             while (rs.next()) {

                txtSearchOutput.setText("The Doctor ID is: " +  rs.getString("D_id")+
                                       "\nThe Doctor Name is : " +  rs.getString("D_name")+
                                       "\nGender: " +  rs.getString("Gender")+
                                       "\nAge is : " +  rs.getString("Age")+
                                       "\nThe Position is  : " +  rs.getString("Position")+
                                       "\nOffice Location  is : " +  rs.getString("Office")+
                                       "\nPhone Number is : " +  rs.getString("Phone")+
                                       "\nAddress is : " +  rs.getString("Address")+
                                       "\nCity is : " +  rs.getString("City")+
                                       "\nThe Postal Code is : " +  rs.getString("Zip"));

             }

             txtSearchID.setText("");
        }
     catch(SQLException e){
         e.printStackTrace();
         JOptionPane.showMessageDialog(null, "error in sql"+e);
         new Doctor_Details().setVisible(true);
```

Output:



D1

After pressing the button:



```
The Doctor ID is: D1
The Doctor Name is : Dr.Chitre
Gender: Female
Age is : 23
The Position is  : Assistant Doc
```

- Patient Table

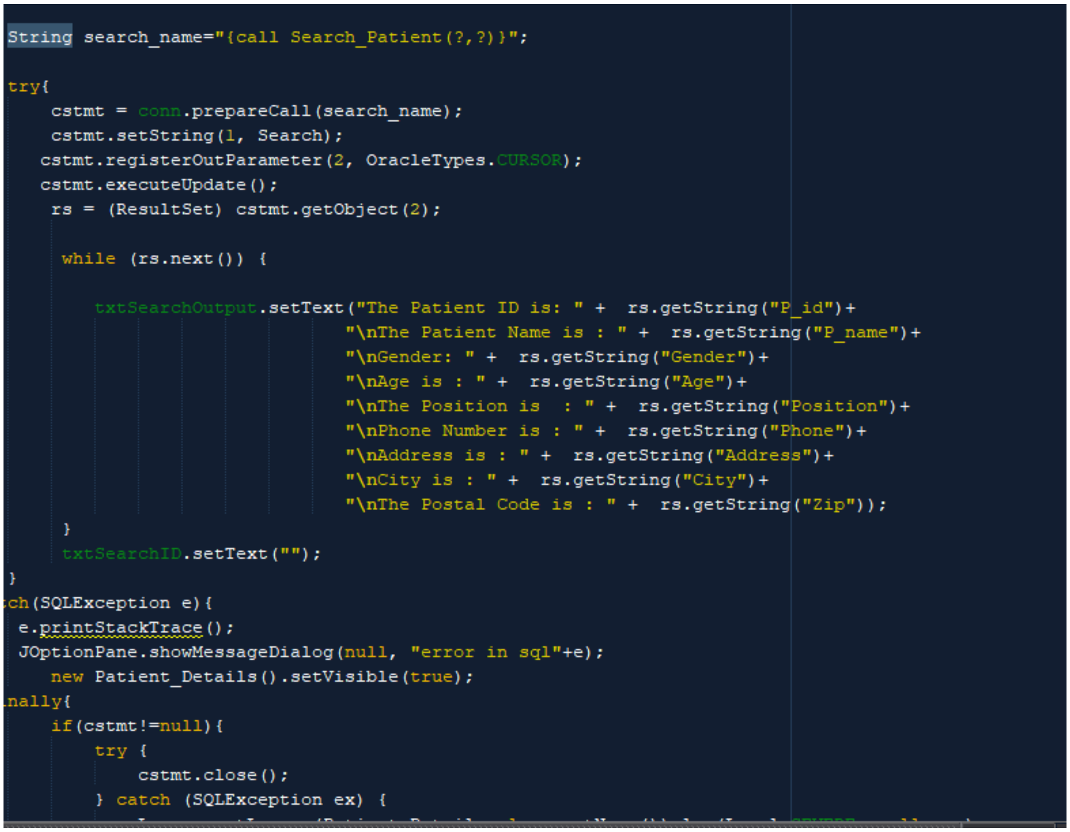
```
create or replace procedure Search_Patient(id varchar) as
cursor c_pat is select
P_id,P_name,Gender,Age,Position,Phone,Address,City,Zip
from Patient;
r_pat c_pat%ROWTYPE;
begin
OPEN c_pat;
LOOP
FETCH c_pat into r_pat;
EXIT WHEN c_pat%NOTFOUND;
if(r_pat.P_id=id) then
   dbms_output.put_line('Name of the Patient is : '|| r_pat.P_name);
   dbms_output.put_line('Gender : '|| r_pat.Gender);
   dbms_output.put_line('Age : '|| r_pat.Age);
   dbms_output.put_line('Position is : '|| r_pat.Position);
   dbms_output.put_line('Phone : '|| r_pat.Phone);
   dbms_output.put_line('Address : '|| r_pat.Address);
   dbms_output.put_line('City : '|| r_pat.City);
   dbms_output.put_line('Pincode : '|| r_pat.Zip);
end if;
end loop;
```

close c_pat;
end;
declare
Patient_Id varchar(5):=:id;
begin
Search_Patient(Patient_Id);
End;

For Calling in java a little change in the procedure

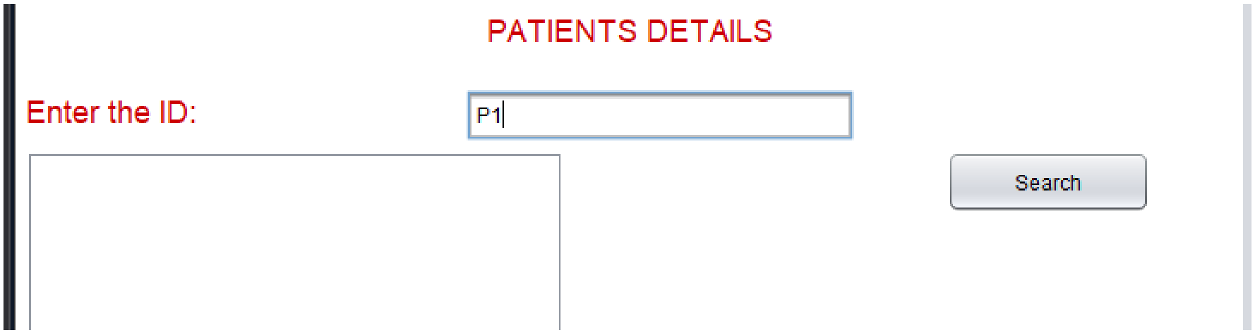
create or replace procedure Search_Patient(id in varchar, c_pat out sys_refcursor) as
begin
open c_pat for select * from Patient where id=P_ID;
End;

Calling the Procedure in Java

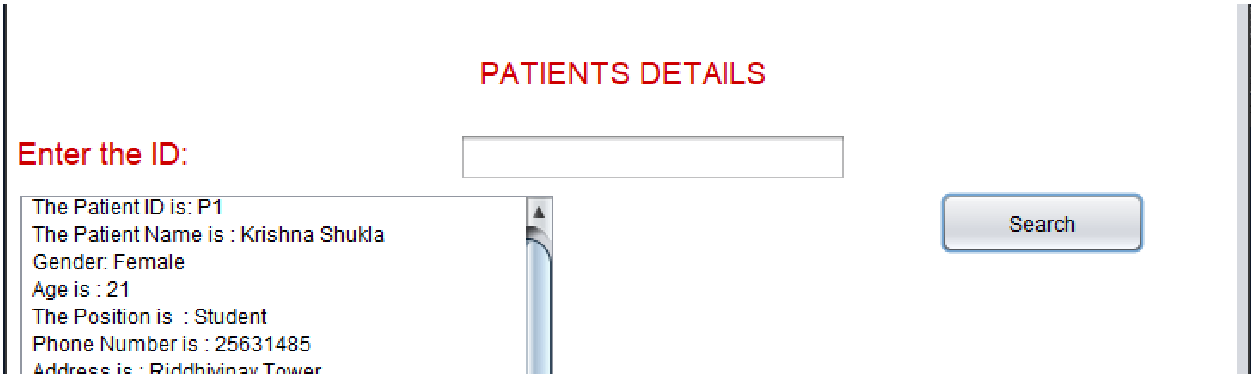


```
String search_name="{call Search_Patient(?,?)}";

try{
    cstmt = conn.prepareCall(search_name);
    cstmt.setString(1, Search);
   cstmt.registerOutParameter(2, OracleTypes.CURSOR);
   cstmt.executeUpdate();
    rs = (ResultSet) cstmt.getObject(2);

     while (rs.next()) {

        txtSearchOutput.setText("The Patient ID is: " +  rs.getString("P_id")+
                              "\nThe Patient Name is : " +  rs.getString("P_name")+
                              "\nGender: " +  rs.getString("Gender")+
                              "\nAge is : " +  rs.getString("Age")+
                              "\nThe Position is  : " +  rs.getString("Position")+
                              "\nPhone Number is : " +  rs.getString("Phone")+
                              "\nAddress is : " +  rs.getString("Address")+
                              "\nCity is : " +  rs.getString("City")+
                              "\nThe Postal Code is : " +  rs.getString("Zip"));
     }
     txtSearchID.setText("");
}
ch(SQLException e){
 e.printStackTrace();
 JOptionPane.showMessageDialog(null, "error in sql"+e);
    new Patient_Details().setVisible(true);
nally{
    if(cstmt!=null){
        try {
            cstmt.close();
        } catch (SQLException ex) {
```

Output:



After:



```
The Patient ID is: P1
The Patient Name is : Krishna Shukla
Gender: Female
Age is : 21
The Position is  : Student
Phone Number is : 25631485
Address is : Riddhivinay Tower
```

- **For Diagnosis and CPT**

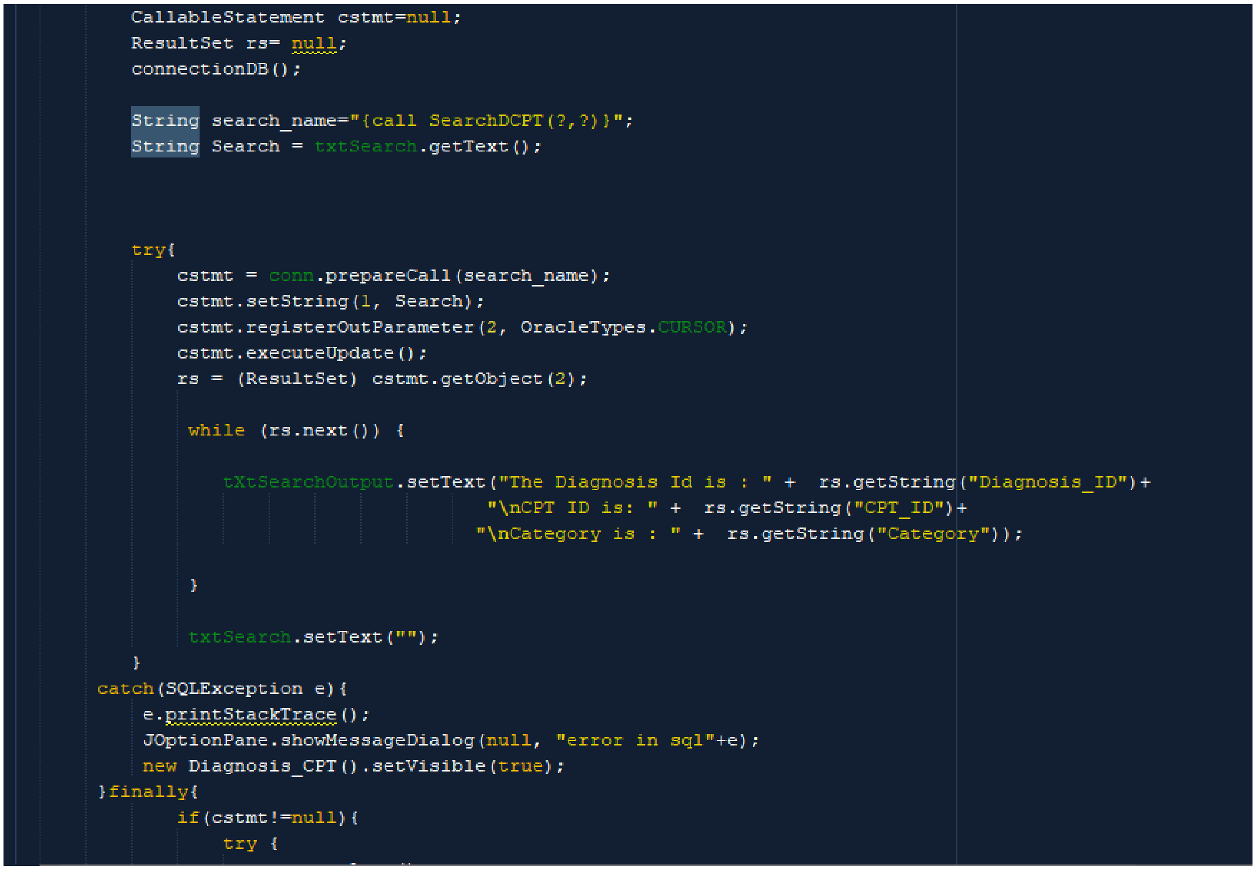
```
create or replace procedure SearchDCPT(x in varchar) as
cursor c_all is select
Diagnosis.Diagnosis_ID,CPT_ID,Diagnosis.Category from
Diagnosis,CPT where Diagnosis.Category = CPT.Category;
r_all c_all%ROWTYPE;
begin
OPEN c_all;
LOOP
FETCH c_all into r_all;
EXIT WHEN c_all%NOTFOUND;
if(r_all.Diagnosis_ID = x) then
```

```
    dbms_output.put_line('Diagnosis Id : '||r_all.Diagnosis_ID);
    dbms_output.put_line('CPT Id : '||r_all.CPT_ID);
    dbms_output.put_line('Category is : '||r_all.Category);
end if;
end loop;
close c_all;
end;

declare
DID varchar(5):=:id;
begin
SearchDCPT(DID);
end;
```

For Calling in java a little change in the procedure

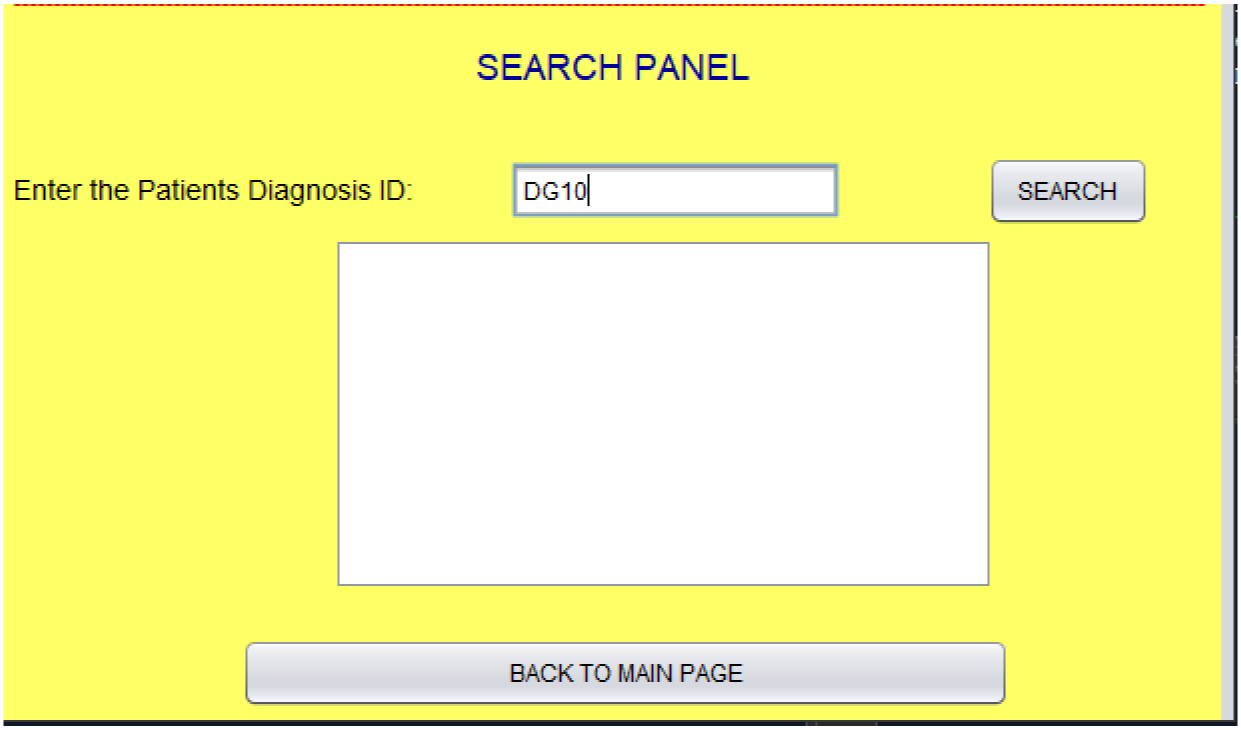
```
create or replace procedure SearchDCPT(id in varchar, c_all out
sys_refcursor) as
begin
open c_all for select
Diagnosis.Diagnosis_ID,CPT_ID,Diagnosis.Category from
Diagnosis,CPT where Diagnosis.Category = CPT.Category and
Diagnosis_ID=id;
End;
```

Calling the Procedure in Java



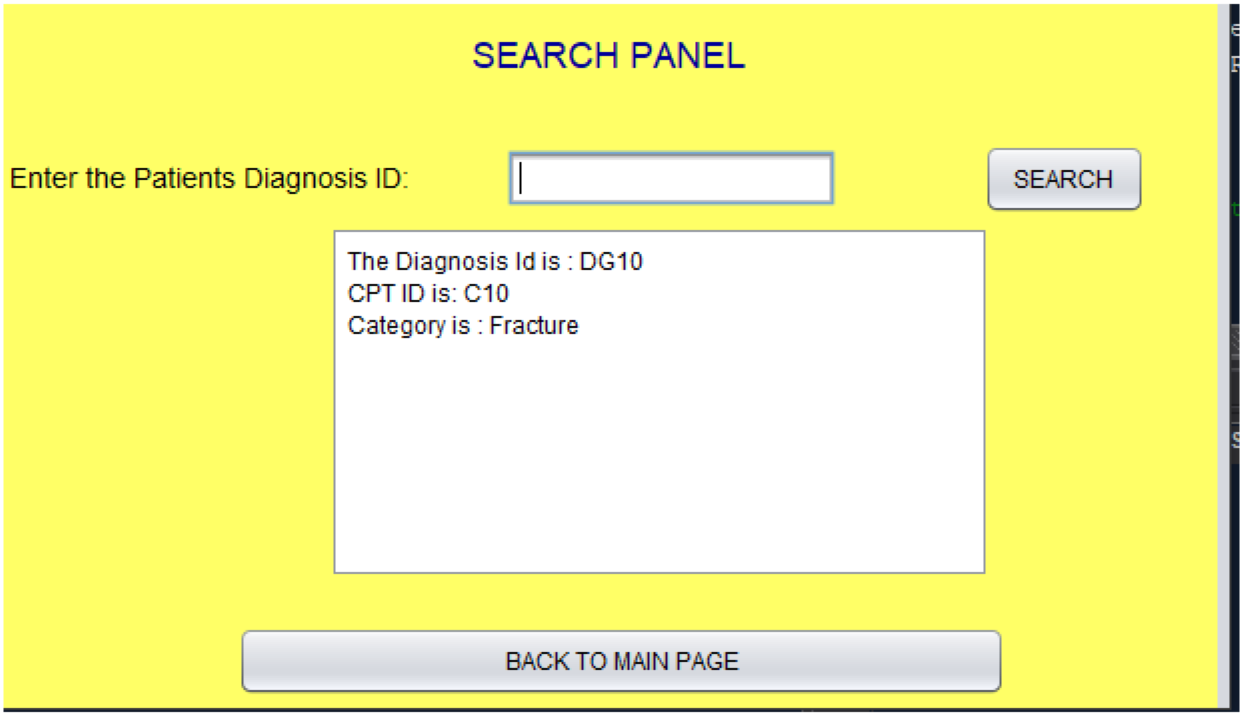
```
    CallableStatement cstmt=null;
    ResultSet rs= null;
    connectionDB();

    String search_name="{call SearchDCPT(?,?)}";
    String Search = txtSearch.getText();



    try{
        cstmt = conn.prepareCall(search_name);
        cstmt.setString(1, Search);
        cstmt.registerOutParameter(2, OracleTypes.CURSOR);
        cstmt.executeUpdate();
        rs = (ResultSet) cstmt.getObject(2);

         while (rs.next()) {

            tXtSearchOutput.setText("The Diagnosis Id is : " +  rs.getString("Diagnosis_ID")+
                                       "\nCPT ID is: " +  rs.getString("CPT_ID")+
                                      "\nCategory is : " +  rs.getString("Category"));

         }

         txtSearch.setText("");
    }
 catch(SQLException e){
     e.printStackTrace();
     JOptionPane.showMessageDialog(null, "error in sql"+e);
     new Diagnosis_CPT().setVisible(true);
 }finally{
        if(cstmt!=null){
            try {
```

Output:



Enter the Patients Diagnosis ID: DG10

After:



```
The Diagnosis Id is : DG10
CPT ID is: C10
Category is : Fracture
```

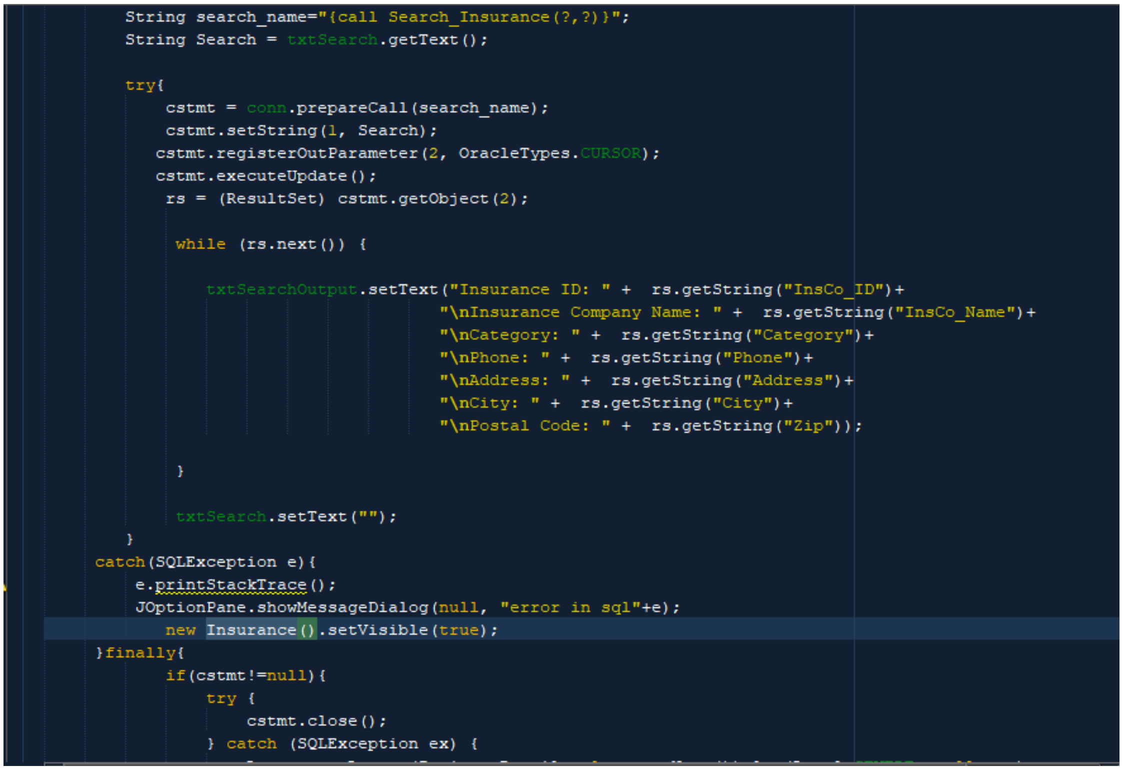
- **Insurance Table**

```
create or replace procedure Search_Insurance(x varchar) as
cursor c_ins is select * from Insurance;
r_ins c_ins%ROWTYPE;
begin
OPEN c_ins;
LOOP
FETCH c_ins into r_ins;
EXIT WHEN c_ins%NOTFOUND;
if(r_ins.InsCo_ID=x) then
    dbms_output.put_line('The Insurance Company ID is '||r_ins.InsCo_ID);
    dbms_output.put_line('The Insurance Company name is
'||r_ins.InsCo_Name);
    dbms_output.put_line('Category is '||r_ins.Category);
    dbms_output.put_line('Phone Number is '||r_ins.Phone);
    dbms_output.put_line('Address is '||r_ins.Address);
    dbms_output.put_line('City : '||r_ins.City);
    dbms_output.put_line('The Postal Code is '||r_ins.Zip);
end if;
end loop;
close c_ins;
end;

declare
InsCo_ID varchar(10):=:ID;
begin
Search_Insurance(InsCo_ID);
End;
```

For Calling in java a little change in the procedure

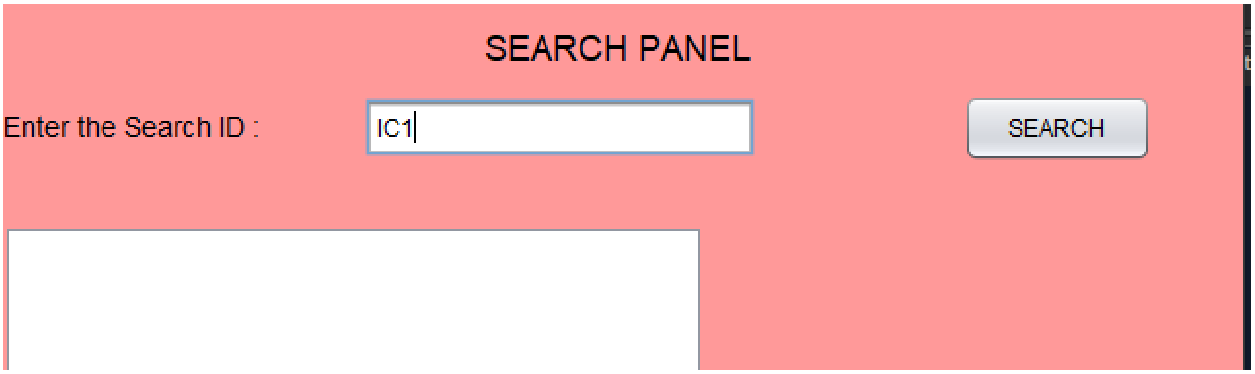
```
create or replace procedure Search_Insurance(x in varchar, c_ins out
sys_refcursor) as
begin
open c_ins for select * from Insurance where InsCo_ID=x;
End;
```

Calling the Procedure in Java



```java
            String search_name="{call Search_Insurance(?,?)}";
            String Search = txtSearch.getText();

            try{
                cstmt = conn.prepareCall(search_name);
                cstmt.setString(1, Search);
               cstmt.registerOutParameter(2, OracleTypes.CURSOR);
               cstmt.executeUpdate();
                rs = (ResultSet) cstmt.getObject(2);

                 while (rs.next()) {

                    txtSearchOutput.setText("Insurance ID: " +  rs.getString("InsCo_ID")+
                                           "\nInsurance Company Name: " +  rs.getString("InsCo_Name")+
                                           "\nCategory: " +  rs.getString("Category")+
                                           "\nPhone: " +  rs.getString("Phone")+
                                           "\nAddress: " +  rs.getString("Address")+
                                           "\nCity: " +  rs.getString("City")+
                                           "\nPostal Code: " +  rs.getString("Zip"));


                 }

                 txtSearch.setText("");
            }
         catch(SQLException e){
             e.printStackTrace();
             JOptionPane.showMessageDialog(null, "error in sql"+e);
                new Insurance().setVisible(true);
         }finally{
                if(cstmt!=null){
                    try {
                        cstmt.close();
                    } catch (SQLException ex) {
```

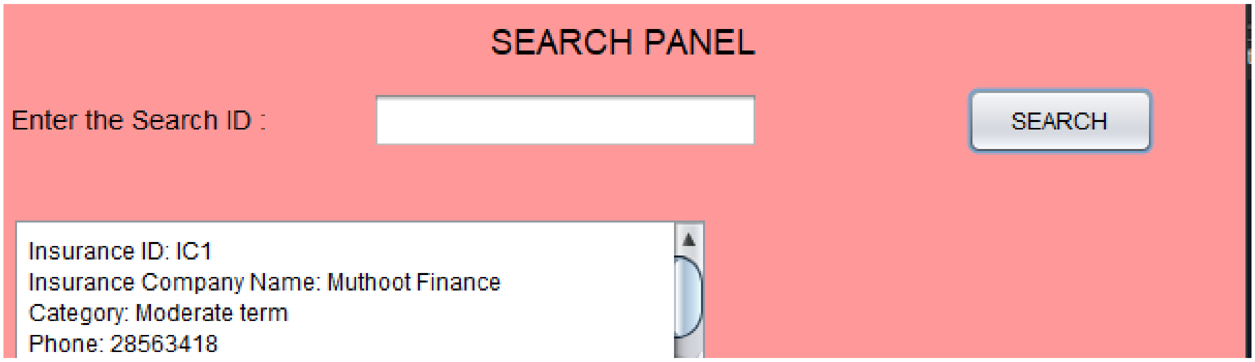
Output



SEARCH PANEL

Enter the Search ID : IC1

→ After:



Insurance ID: IC1
Insurance Company Name: Muthoot Finance
Category: Moderate term
Phone: 28563418

- **Invoice Table**

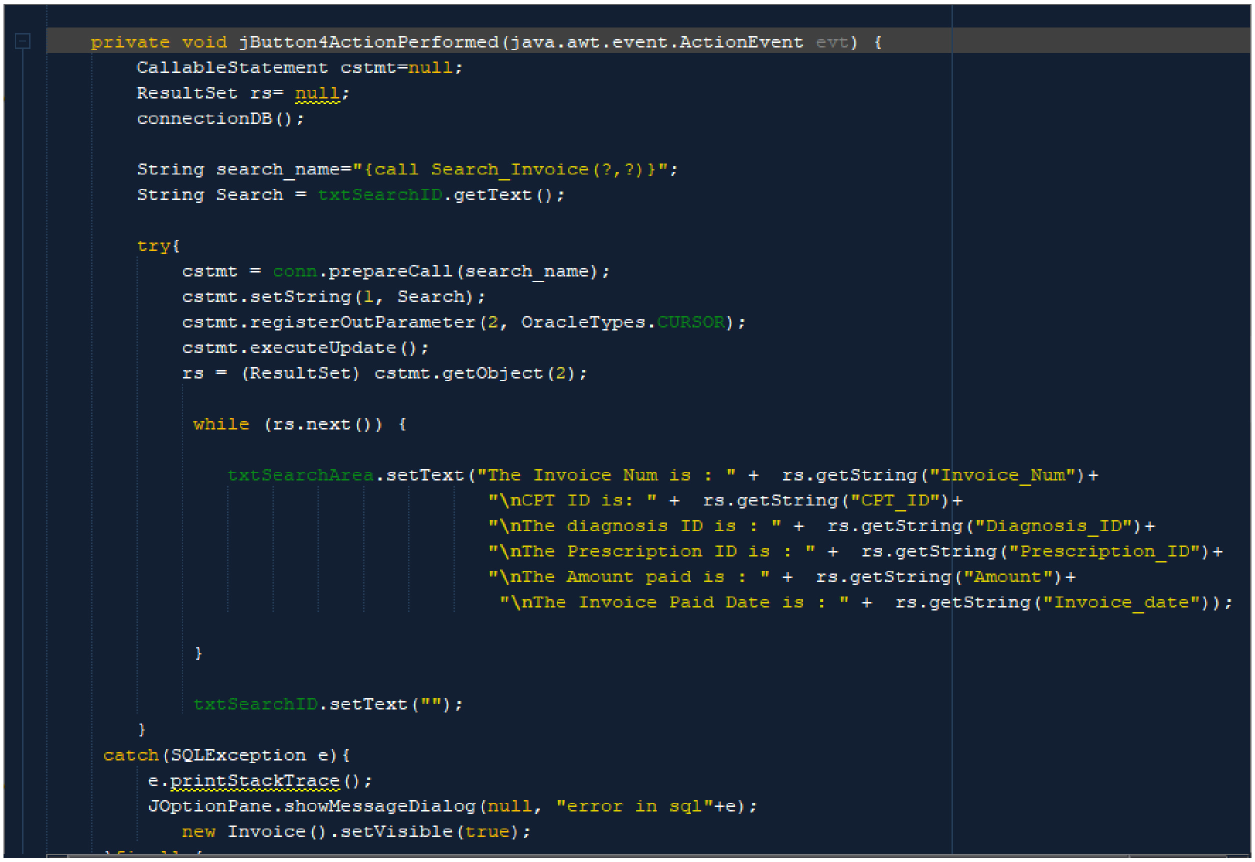
```
create or replace procedure SearchInvoice(x in varchar) as
cursor c_sel is select * from Invoice_P;
r_sel c_sel%ROWTYPE;
begin
OPEN c_sel;
LOOP
FETCH c_sel into r_sel;
EXIT WHEN c_sel%NOTFOUND;
if(r_sel.P_ID = x) then
    dbms_output.put_line('Invoice Num is : '||r_sel.Invoice_Num);
    dbms_output.put_line('CPT ID is : '||r_sel.CPT_ID);
    dbms_output.put_line('Diagnosis ID is : '||r_sel.Diagnosis_ID);
    dbms_output.put_line('Prescription ID  is : '||r_sel.Prescription_ID);
    dbms_output.put_line('Amount paid is : '||r_sel.Amount);
    dbms_output.put_line('Invoice Date  is : '||r_sel.Invoice_date);
    dbms_output.put_line('Due Date Num is : '||r_sel.Due_Date);
end if;
end loop;
close c_sel;
end;
```

declare
Patient varchar(5):=:id;
begin
SearchInvoice(Patient);
End;

For Calling in java a little change in the procedure

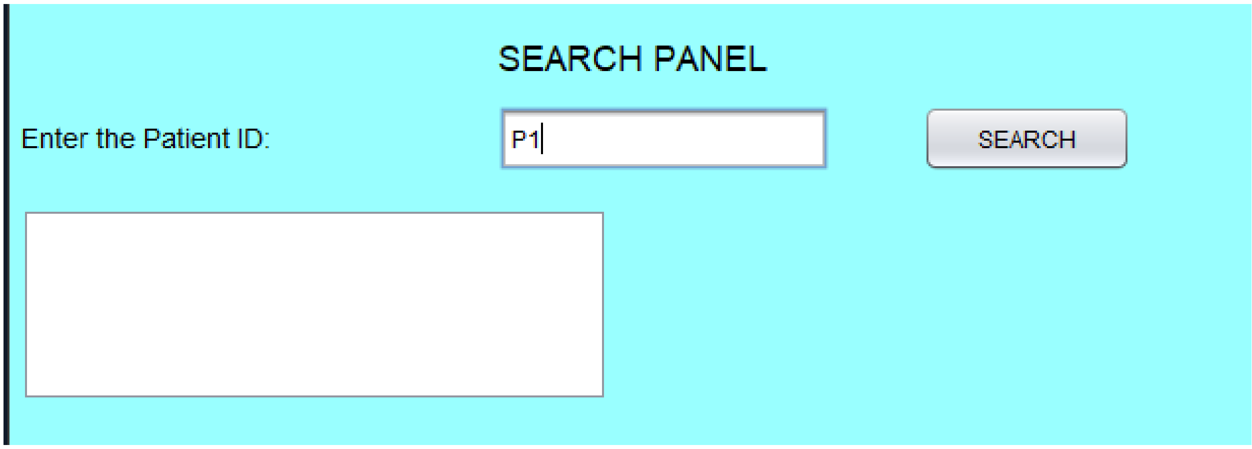
create or replace procedure Search_Invoice(id in varchar, c_in out sys_refcursor) as
begin
open c_in for select * from Invoice_P where id=P_ID;
End;

Calling the Procedure in Java



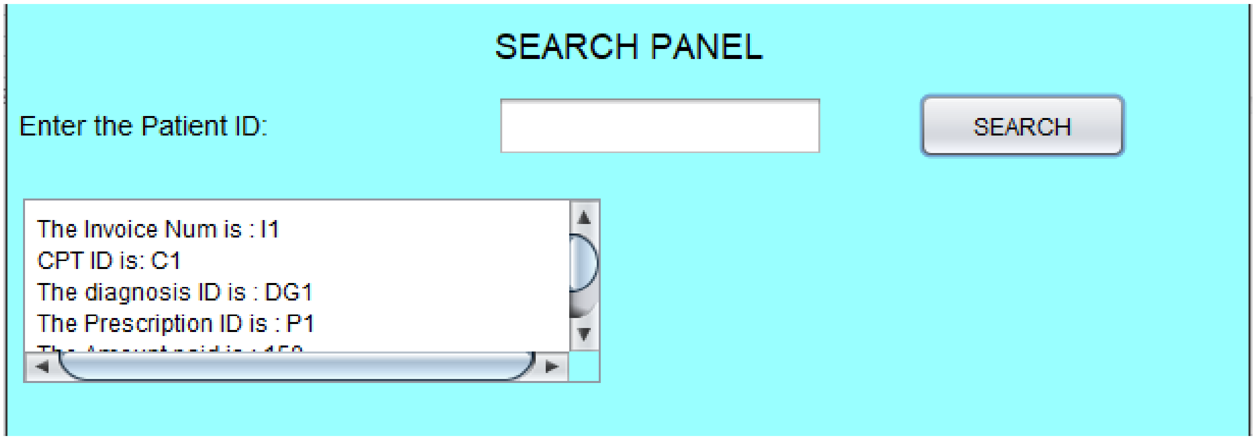
```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    CallableStatement cstmt=null;
    ResultSet rs= null;
    connectionDB();

    String search_name="{call Search_Invoice(?,?)}";
    String Search = txtSearchID.getText();

    try{
        cstmt = conn.prepareCall(search_name);
        cstmt.setString(1, Search);
        cstmt.registerOutParameter(2, OracleTypes.CURSOR);
        cstmt.executeUpdate();
        rs = (ResultSet) cstmt.getObject(2);

         while (rs.next()) {

            txtSearchArea.setText("The Invoice Num is : " +  rs.getString("Invoice_Num")+
                                   "\nCPT ID is: " +  rs.getString("CPT_ID")+
                                   "\nThe diagnosis ID is : " +  rs.getString("Diagnosis_ID")+
                                   "\nThe Prescription ID is : " +  rs.getString("Prescription_ID")+
                                   "\nThe Amount paid is : " +  rs.getString("Amount")+
                                    "\nThe Invoice Paid Date is : " +  rs.getString("Invoice_date"));

         }

         txtSearchID.setText("");
    }
 catch(SQLException e){
     e.printStackTrace();
     JOptionPane.showMessageDialog(null, "error in sql"+e);
        new Invoice().setVisible(true);
```

Output



Enter the Patient ID: P1

→ After:



Enter the Patient ID:

```
The Invoice Num is : I1
CPT ID is: C1
The diagnosis ID is : DG1
The Prescription ID is : P1
The Amount paid is : 150
```

- **Medicine and Prescription Table**

→ **Function:**

create or replace function CountBill(x varchar) return int as
temp int;
cursor c_med is select Prescription.Prescription_ID,Prescription.Medicine_Q,Medicine_P.M_ID,Medicine_P.M_Name,Medicine_P.Manufacturer,Medicine_P.Price,Medicine_P.

```
Exp_Date from Prescription,Medicine_P where
Prescription.Prescription_ID=Medicine_P.Prescription_ID;
r_med c_med%ROWTYPE;
begin
temp:=0;
OPEN c_med;
LOOP
FETCH c_med into r_med;
EXIT WHEN c_med%NOTFOUND;
if(r_med.Prescription_ID=x) then
    temp:= r_med.Medicine_Q*r_med.Price;
end if;
end loop;
close c_med;
return temp;
end;

declare
Prescription varchar(5):=:ID;
bill int;
begin
bill:= CountBill(Prescription);
dbms_output.put_line('The Total Bill is :'||bill);
end;

create or replace procedure Search_MedPre(x varchar) as
cursor c_med is select
Prescription.Prescription_ID,Prescription.Medicine_Q,Medicine_P.M_ID,Medicine_P.M_Name,Medicine_P.Manufacturer,Medicine_P.Price,Medicine_P.Exp_Date from Prescription,Medicine_P where
Prescription.Prescription_ID=Medicine_P.Prescription_ID;
r_med c_med%ROWTYPE;
begin
```

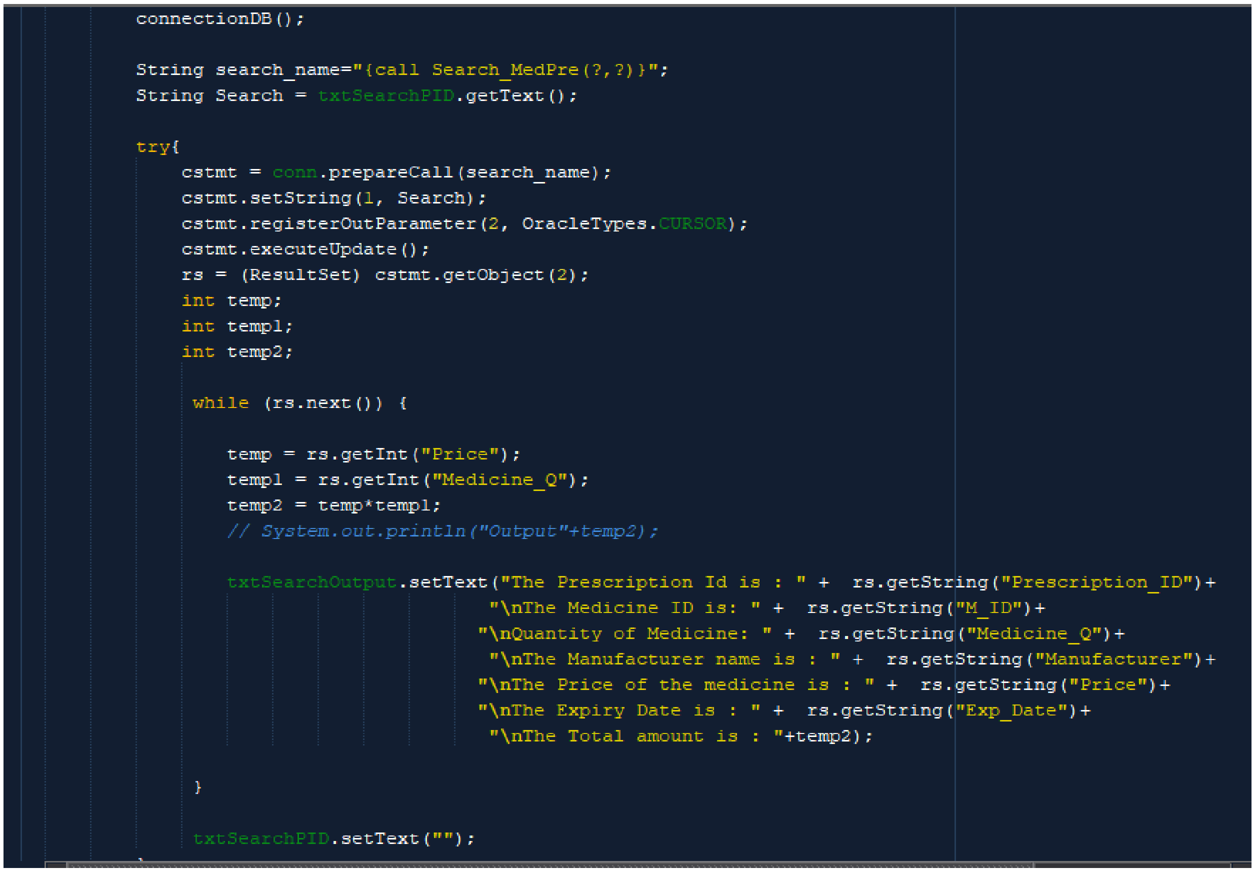
```
OPEN c_med;
LOOP
FETCH c_med into r_med;
EXIT WHEN c_med%NOTFOUND;
if(r_med.Prescription_ID=x) then
      dbms_output.put_line('The Prescription ID is: '||r_med.Prescription_ID);
      dbms_output.put_line('The Medicine ID is: '||r_med.M_ID);
      dbms_output.put_line('The Quantity of medicine is:
'||r_med.Medicine_Q);
      dbms_output.put_line('The Medicine name is: '||r_med.M_Name);
      dbms_output.put_line('The Manufacturer name is:
'||r_med.Manufacturer);
      dbms_output.put_line('The Price is: '||r_med.Price);
      dbms_output.put_line('The Expiry Date is: '||r_med.Exp_Date);
      dbms_output.put_line('------------------------------------');
end if;
end loop;
close c_med;
end;

declare
Prescription varchar(5):=:ID;
begin
Search_MedPre(Prescription);
end;
```

→ For Calling in java a little change in the procedure

```
create or replace procedure Search_MedPre(x in varchar, c_med out
sys_refcursor) as
begin
open c_med for select
Prescription.Prescription_ID,Prescription.Medicine_Q,Medicine_P.Quantity,
```

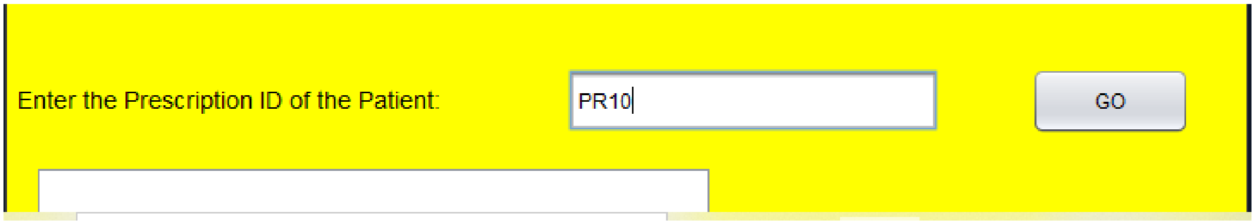
Medicine_P.M_ID,Medicine_P.M_Name,Medicine_P.Manufacturer,Medicine_P.Price,Medicine_P.Exp_Date from Prescription,Medicine_P where Prescription.Prescription_ID=Medicine_P.Prescription_ID and Prescription.Prescription_ID=x;
End;

→ Calling the Procedure in Java



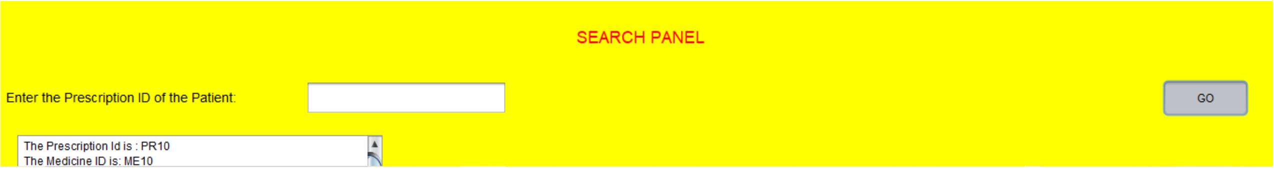
```
        connectionDB();

        String search_name="{call Search_MedPre(?,?)}";
        String Search = txtSearchPID.getText();

        try{
            cstmt = conn.prepareCall(search_name);
            cstmt.setString(1, Search);
            cstmt.registerOutParameter(2, OracleTypes.CURSOR);
            cstmt.executeUpdate();
            rs = (ResultSet) cstmt.getObject(2);
            int temp;
            int temp1;
            int temp2;

             while (rs.next()) {

                temp = rs.getInt("Price");
                temp1 = rs.getInt("Medicine_Q");
                temp2 = temp*temp1;
                // System.out.println("Output"+temp2);

                txtSearchOutput.setText("The Prescription Id is : " +  rs.getString("Prescription_ID")+
                                       "\nThe Medicine ID is: " +  rs.getString("M_ID")+
                                      "\nQuantity of Medicine: " +  rs.getString("Medicine_Q")+
                                       "\nThe Manufacturer name is : " +  rs.getString("Manufacturer")+
                                      "\nThe Price of the medicine is : " +  rs.getString("Price")+
                                      "\nThe Expiry Date is : " +  rs.getString("Exp_Date")+
                                       "\nThe Total amount is : "+temp2);

             }

             txtSearchPID.setText("");
```

→ Output:



Enter the Prescription ID of the Patient: PR10

→ After:



The Prescription Id is : PR10
The Medicine ID is: ME10

- **Paycheck Table**

→ **Function:**

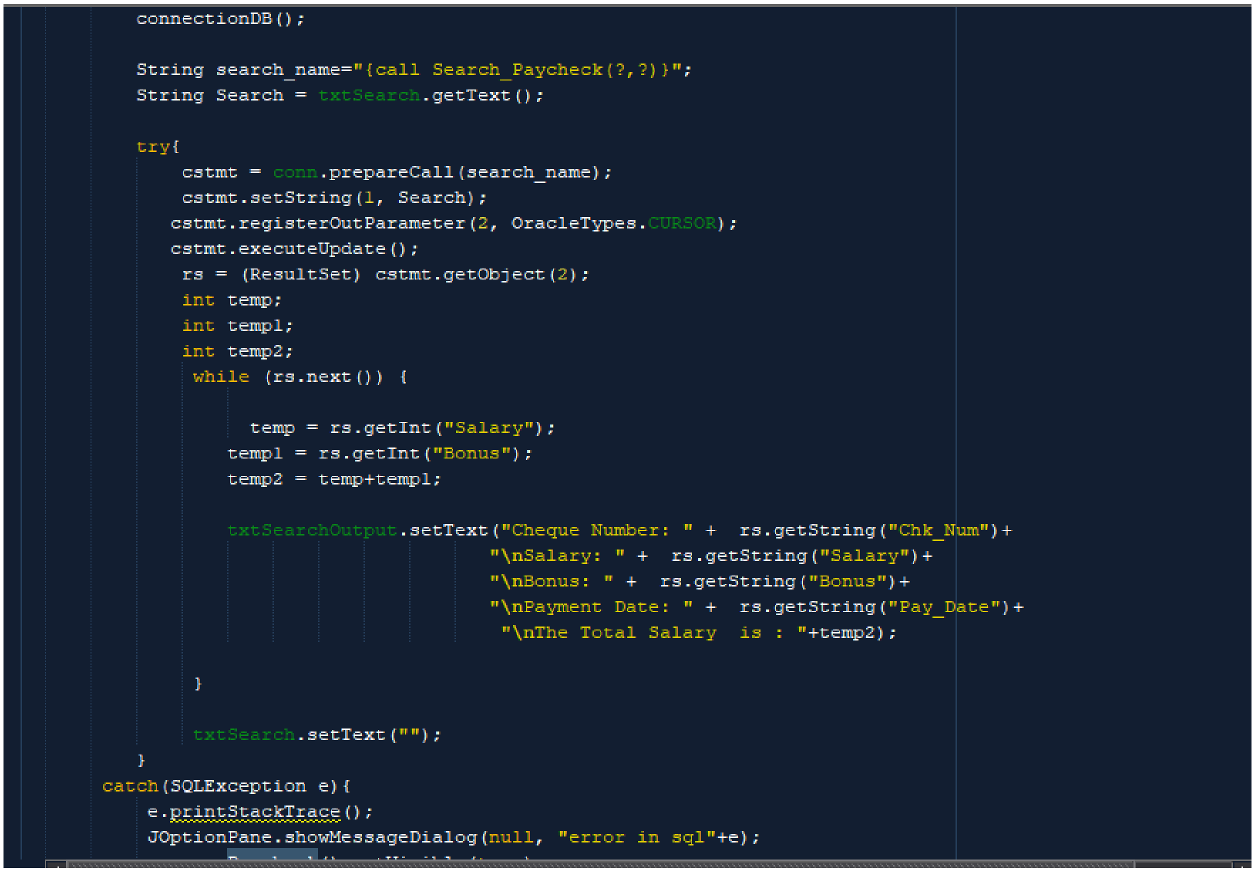
```
create or replace function CountSalary(x varchar) return int as
temp int;
cursor c_det is select
Paycheck_Doc.Chk_Num,Paycheck_Doc.D_Id,Paycheck.Salary,Paycheck.
Bonus,Paycheck.Pay_Date from Paycheck_Doc,Paycheck where
Paycheck_Doc.Chk_Num = Paycheck.Chk_Num;
r_det c_det%ROWTYPE;
begin
temp:=0;
OPEN c_det;
LOOP
FETCH c_det into r_det;
EXIT WHEN c_det%NOTFOUND;
if(r_det.Chk_Num = x) then
    temp := r_det.Salary+r_det.Bonus;
end if;
end loop;
close c_det;
return temp;
end;
```

```
declare
Chk_Num varchar(5):=:ID;
bill int;
begin
bill:= CountSalary(Chk_Num);
dbms_output.put_line('The Total Salary is :'||bill);
end;

create or replace procedure Search_Paycheck(x varchar) as
cursor c_pay is select * from Paycheck;
r_pay c_pay%ROWTYPE;
begin
OPEN c_pay;
LOOP
FETCH c_pay into r_pay;
EXIT WHEN c_pay%NOTFOUND;
if(r_pay.Chk_Num=x) then
   dbms_output.put_line('The Cheque Number is :'||r_pay.Chk_Num);
   dbms_output.put_line('Amount of salary is :'||r_pay.Salary);
   dbms_output.put_line('Bonus is :'||r_pay.Bonus);
   dbms_output.put_line('The Pay Date is :'||r_pay.Pay_Date);
end if;
end loop;
close c_pay;
end;

declare
Chk_Number varchar(5):=:Number;
begin
Search_Paycheck(Chk_Number);
end;
```

→ For Calling in java a little change in the procedure

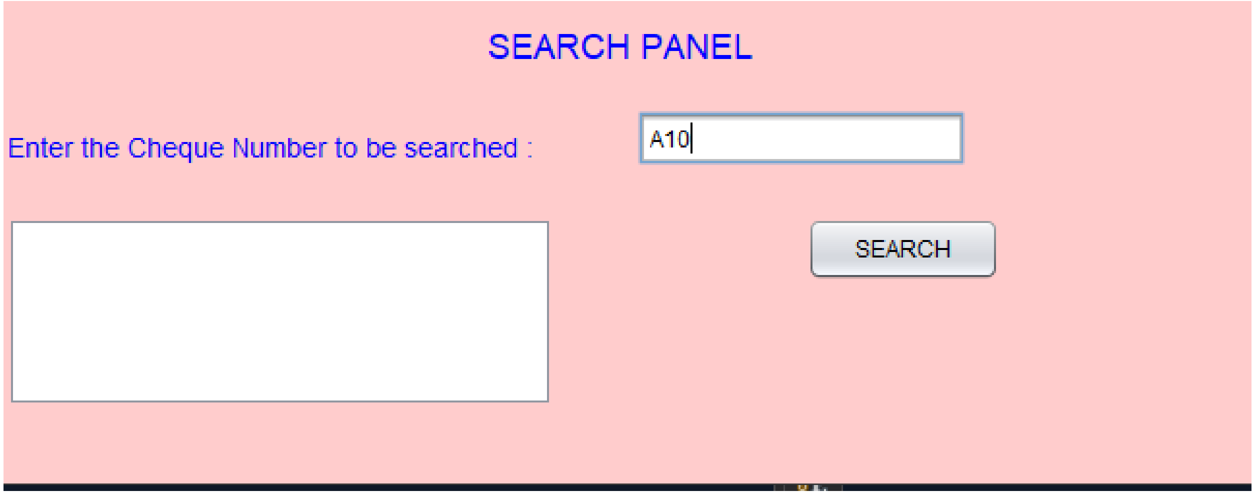
create or replace procedure Search_Paycheck(x in varchar, c_pay out sys_refcursor) as
begin
open c_pay for select * from Paycheck where Chk_Num=x;
end;

→ Calling the Procedure in Java



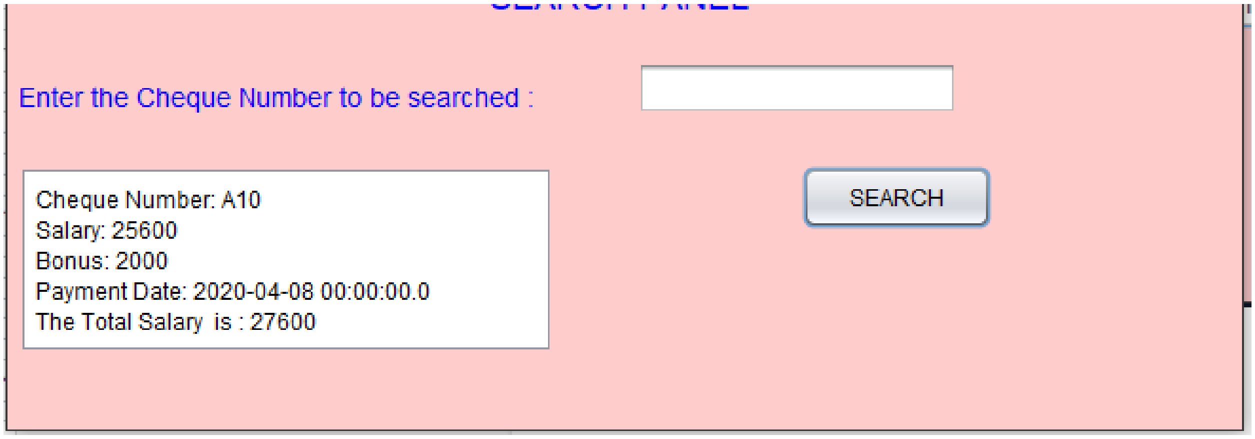
```
        connectionDB();

        String search_name="{call Search_Paycheck(?,?)}";
        String Search = txtSearch.getText();

        try{
            cstmt = conn.prepareCall(search_name);
            cstmt.setString(1, Search);
           cstmt.registerOutParameter(2, OracleTypes.CURSOR);
           cstmt.executeUpdate();
            rs = (ResultSet) cstmt.getObject(2);
            int temp;
            int templ;
            int temp2;
             while (rs.next()) {

                  temp = rs.getInt("Salary");
                templ = rs.getInt("Bonus");
                temp2 = temp+templ;

                txtSearchOutput.setText("Cheque Number: " +  rs.getString("Chk_Num")+
                                       "\nSalary: " +  rs.getString("Salary")+
                                       "\nBonus: " +  rs.getString("Bonus")+
                                       "\nPayment Date: " +  rs.getString("Pay_Date")+
                                        "\nThe Total Salary  is : "+temp2);

             }

             txtSearch.setText("");
        }
     catch(SQLException e){
         e.printStackTrace();
         JOptionPane.showMessageDialog(null, "error in sql"+e);
```

→ Output



Enter the Cheque Number to be searched : A10

→ After



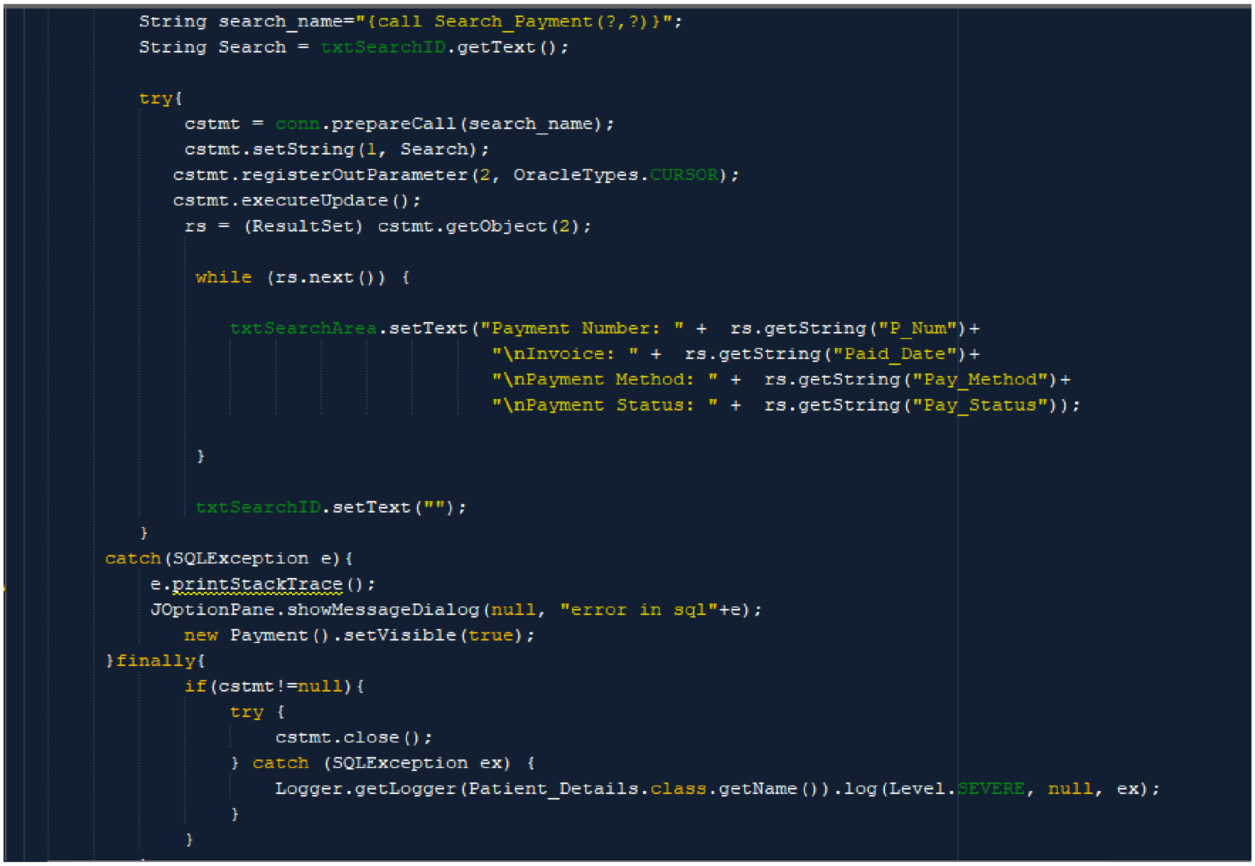
Enter the Cheque Number to be searched :

```
Cheque Number: A10
Salary: 25600
Bonus: 2000
Payment Date: 2020-04-08 00:00:00.0
The Total Salary  is : 27600
```

- **Payment Table**

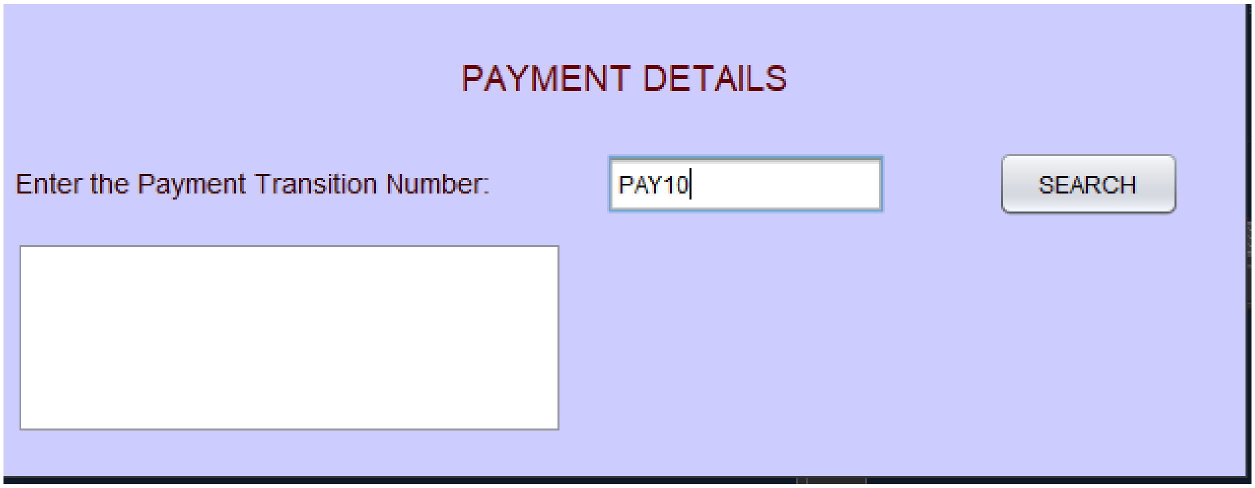
```
create or replace procedure Search_Payment(id in varchar, c_in out
sys_refcursor) as
begin
open c_in for select * from Payment where id=P_NUM;
end;
```

→ Calling the Procedure in Java



```
   String search_name="{call Search_Payment(?,?)}";
   String Search = txtSearchID.getText();

   try{
       cstmt = conn.prepareCall(search_name);
       cstmt.setString(1, Search);
      cstmt.registerOutParameter(2, OracleTypes.CURSOR);
      cstmt.executeUpdate();
       rs = (ResultSet) cstmt.getObject(2);

        while (rs.next()) {

           txtSearchArea.setText("Payment Number: " +  rs.getString("P_Num")+
                                  "\nInvoice: " +  rs.getString("Paid_Date")+
                                  "\nPayment Method: " +  rs.getString("Pay_Method")+
                                  "\nPayment Status: " +  rs.getString("Pay_Status"));

        }

        txtSearchID.setText("");
   }
catch(SQLException e){
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "error in sql"+e);
       new Payment().setVisible(true);
}finally{
       if(cstmt!=null){
           try {
               cstmt.close();
           } catch (SQLException ex) {
               Logger.getLogger(Patient_Details.class.getName()).log(Level.SEVERE, null, ex);
           }
       }
```

→ Output



PAYMENT DETAILS

Enter the Payment Transition Number: PAY10

→ After

- **For General Searching:**

1. **Complete Information about the patient**

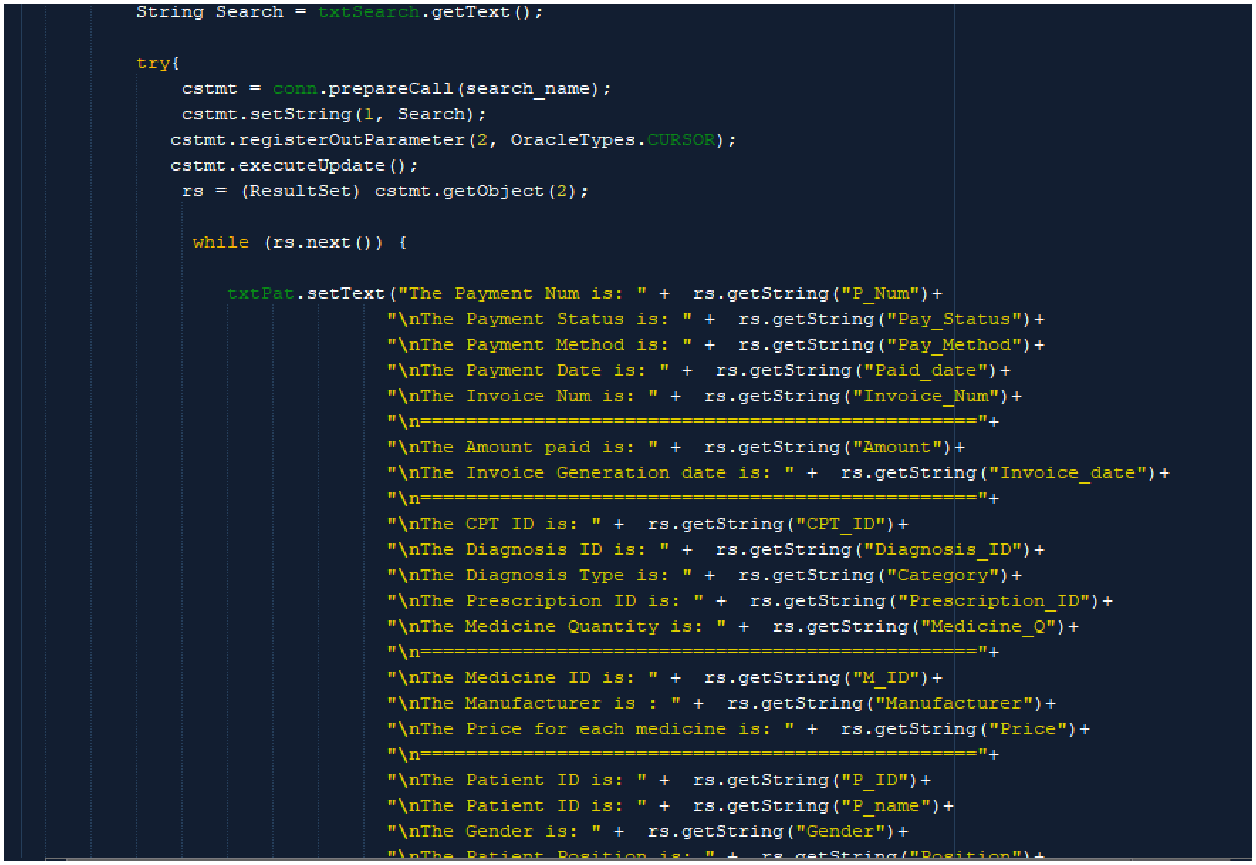
```
→ create or replace procedure PatientFullDetails(x in varchar) as
cursor c_all is select
Medicine_P.Price,Medicine_P.Manufacturer,Medicine_P.M_ID,Doctor.D_name,Diagnosis.Category,Patient.Zip,Patient.City,Patient.Address,Patient.Phone,Patient.Gender,Patient.Position,Patient_Doc.D_id,Patient.P_name,Prescription.Medicine_Q,Invoice_P.Prescription_ID,Invoice_P.Diagnosis_ID,Invoice_P.CPT_ID,Invoice_P.P_ID,Invoice.Invoice_Date,Invoice.Invoice_Num,
Invoice.Amount,Payment_P.P_Num,Payment_P.Invoice,Payment_P.Pay_Status,Payment_P.Paid_Date,Payment_P.Pay_Method from
Invoice,Payment_P,Invoice_P,Prescription,Patient,Patient_Doc,Diagnosis,Doctor,Medicine_P where Payment_P.Invoice = Invoice.Invoice_Num and
Invoice.Invoice_Num=Invoice_P.Invoice_Num and
Invoice_P.Prescription_ID = Prescription.Prescription_ID and Patient.P_id =
Invoice_P.P_ID and Patient.P_id = Patient_Doc.P_id and
Diagnosis.Diagnosis_ID = Invoice_P.Diagnosis_ID and Doctor.D_id =
Patient_Doc.D_id and Medicine_P.Prescription_ID =
Invoice_P.Prescription_ID;
r_all c_all%ROWTYPE;
temp int;
begin
OPEN c_all;
LOOP
FETCH c_all into r_all;
EXIT WHEN c_all%NOTFOUND;
if(r_all.P_id=x) then
   temp:=r_all.Medicine_Q*r_all.Price;
   dbms_output.put_line('Payment Num: '||r_all.P_Num);
   dbms_output.put_line('Payment Status : '||r_all.Pay_Status);
   dbms_output.put_line('Payment Method : '||r_all.Pay_Method);
```

```
    dbms_output.put_line('Payment Date : '||r_all.Paid_date);
    dbms_output.put_line('Invoice Number : '||r_all.Invoice_Num);
    dbms_output.put_line('---------------------');
    dbms_output.put_line('Amount paid : '||r_all.Amount);
    dbms_output.put_line('Invoice Generation Date : '||r_all.Invoice_date);
    dbms_output.put_line('---------------------');
    dbms_output.put_line('CPT ID : '||r_all.CPT_ID);
    dbms_output.put_line('Diagnosis ID : '||r_all.Diagnosis_ID);
    dbms_output.put_line('Diagnosis type : '||r_all.Category);
    dbms_output.put_line('Prescription ID : '||r_all.Prescription_ID);
    dbms_output.put_line('Medicine Quantity : '||r_all.Medicine_Q);
    dbms_output.put_line('---------------------');
    dbms_output.put_line('Medicine ID : '||r_all.M_ID);
    dbms_output.put_line('Manufacturing Company : '||r_all.Manufacturer);
    dbms_output.put_line('Price for Each medicine : '||r_all.Price);
    dbms_output.put_line('Total price for the medicines are : '||temp);
    dbms_output.put_line('---------------------');
    dbms_output.put_line('Patient ID : '||r_all.P_ID);
    dbms_output.put_line('Patient Name : '||r_all.P_name);
    dbms_output.put_line('Patient Gender : '||r_all.Gender);
    dbms_output.put_line('Patient Position : '||r_all.Position);
    dbms_output.put_line('Patient Phone number : '||r_all.Phone);
    dbms_output.put_line('Patient Address : '||r_all.Address);
    dbms_output.put_line('Patient City : '||r_all.City);
    dbms_output.put_line('Patient Postal Code : '||r_all.Zip);
    dbms_output.put_line('---------------------');
    dbms_output.put_line('Doctor ID : '||r_all.D_id);
    dbms_output.put_line('Doctor Name : '||r_all.D_name);
end if;
end loop;
close c_all;
end;
```

→ **For Calling in java a little change in the procedure**

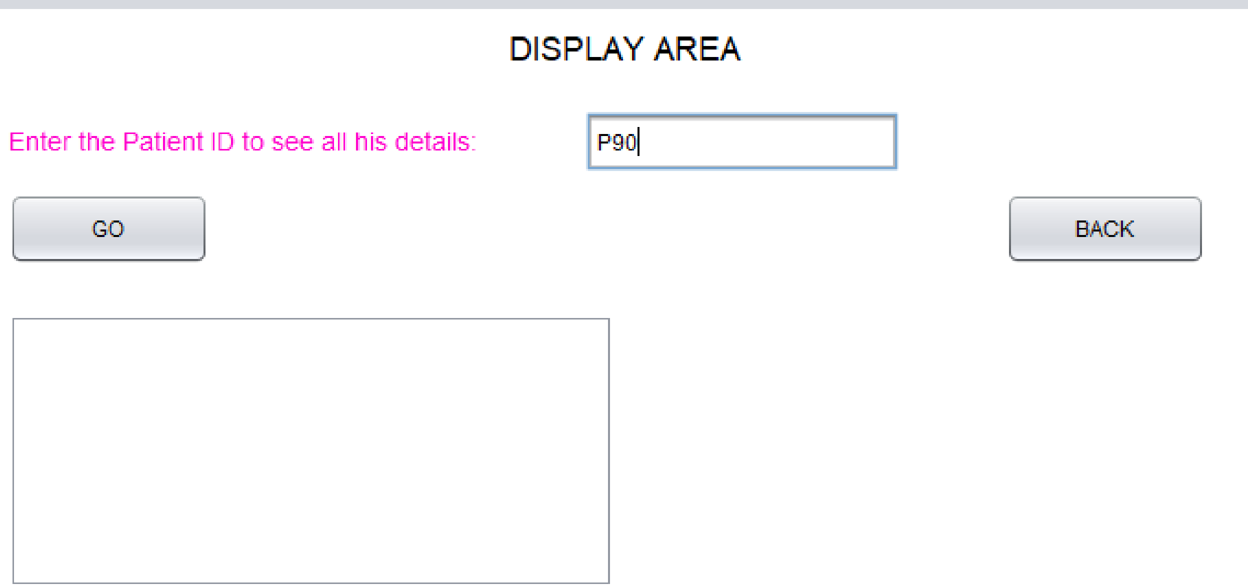
```
create or replace procedure PatientFull(x in varchar,c_dat out
sys_refcursor) as
begin
open c_dat for select
Medicine_P.Price,Medicine_P.Manufacturer,Medicine_P.M_ID,Doctor.D_name,Diagnosis.Category,Patient.Zip,Patient.City,Patient.Address,Patient.Phone,Patient.Gender,Patient.Position,Patient_Doc.D_id,Patient.P_name,Prescription.Medicine_Q,Invoice_P.Prescription_ID,Invoice_P.Diagnosis_ID,Invoice_P.CPT_ID,Invoice_P.P_ID,Invoice.Invoice_Date,Invoice.Invoice_Num,
Invoice.Amount,Payment_P.P_Num,Payment_P.Invoice,Payment_P.Pay_Status,Payment_P.Paid_Date,Payment_P.Pay_Method from
Invoice,Payment_P,Invoice_P,Prescription,Patient,Patient_Doc,Diagnosis,Doctor,Medicine_P where Payment_P.Invoice = Invoice.Invoice_Num and
Invoice.Invoice_Num=Invoice_P.Invoice_Num and
Invoice_P.Prescription_ID = Prescription.Prescription_ID and Patient.P_id =
Invoice_P.P_ID and Patient.P_id = Patient_Doc.P_id and
Diagnosis.Diagnosis_ID = Invoice_P.Diagnosis_ID and Doctor.D_id =
Patient_Doc.D_id and Medicine_P.Prescription_ID =
Invoice_P.Prescription_ID and Patient.P_id = x;
End;
```

→ Calling the Procedure in Java



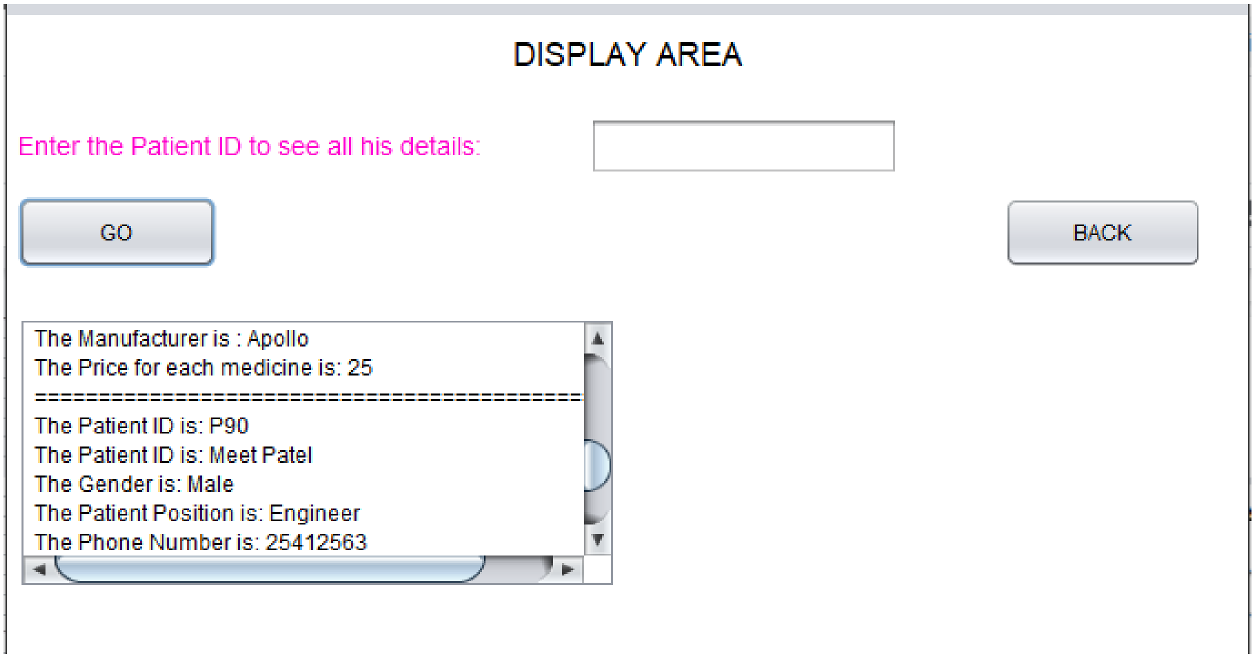
```
String Search = txtSearch.getText();

try{
    cstmt = conn.prepareCall(search_name);
    cstmt.setString(1, Search);
   cstmt.registerOutParameter(2, OracleTypes.CURSOR);
   cstmt.executeUpdate();
    rs = (ResultSet) cstmt.getObject(2);

     while (rs.next()) {

        txtPat.setText("The Payment Num is: " +  rs.getString("P_Num")+
                      "\nThe Payment Status is: " +  rs.getString("Pay_Status")+
                      "\nThe Payment Method is: " +  rs.getString("Pay_Method")+
                      "\nThe Payment Date is: " +  rs.getString("Paid_date")+
                      "\nThe Invoice Num is: " +  rs.getString("Invoice_Num")+
                      "\n=================================================="+
                      "\nThe Amount paid is: " +  rs.getString("Amount")+
                      "\nThe Invoice Generation date is: " +  rs.getString("Invoice_date")+
                      "\n=================================================="+
                      "\nThe CPT ID is: " +  rs.getString("CPT_ID")+
                      "\nThe Diagnosis ID is: " +  rs.getString("Diagnosis_ID")+
                      "\nThe Diagnosis Type is: " +  rs.getString("Category")+
                      "\nThe Prescription ID is: " +  rs.getString("Prescription_ID")+
                      "\nThe Medicine Quantity is: " +  rs.getString("Medicine_Q")+
                      "\n=================================================="+
                      "\nThe Medicine ID is: " +  rs.getString("M_ID")+
                      "\nThe Manufacturer is : " +  rs.getString("Manufacturer")+
                      "\nThe Price for each medicine is: " +  rs.getString("Price")+
                      "\n=================================================="+
                      "\nThe Patient ID is: " +  rs.getString("P_ID")+
                      "\nThe Patient ID is: " +  rs.getString("P_name")+
                      "\nThe Gender is: " +  rs.getString("Gender")+
                      "\nThe Patient Position is: " +  rs.getString("Position")+
```

→ Output



P90

→ After:



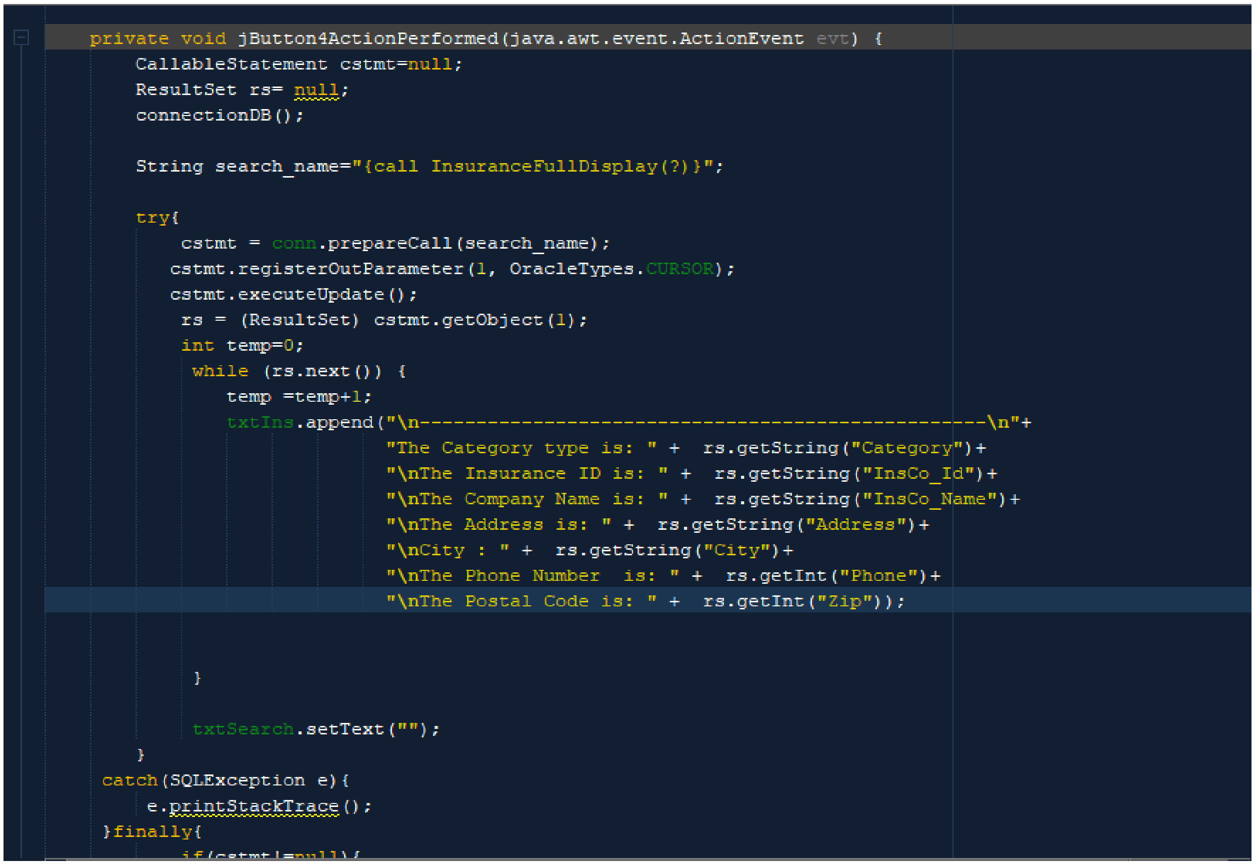
```
The Manufacturer is : Apollo
The Price for each medicine is: 25
============================================
The Patient ID is: P90
The Patient ID is: Meet Patel
The Gender is: Male
The Patient Position is: Engineer
The Phone Number is: 25412563
```

2.Displaying the List of Insurances Category wise:

```
create or replace procedure InsuranceDisplay as
cursor c_cat is select distinct(Category) from Insurance;
cursor c_dat is select
InsCo_ID,InsCo_Name,Phone,Address,City,Zip,Category from Insurance;
r_cat c_cat%ROWTYPE;
r_dat c_dat%ROWTYPE;
category varchar(20);
temp int;
begin
OPEN c_cat;
LOOP
FETCH c_cat into r_cat;
EXIT WHEN c_cat%NOTFOUND;
category:=r_cat.Category;
    dbms_output.put_line('Category: '||category);
    dbms_output.put_line('---------------------');
```

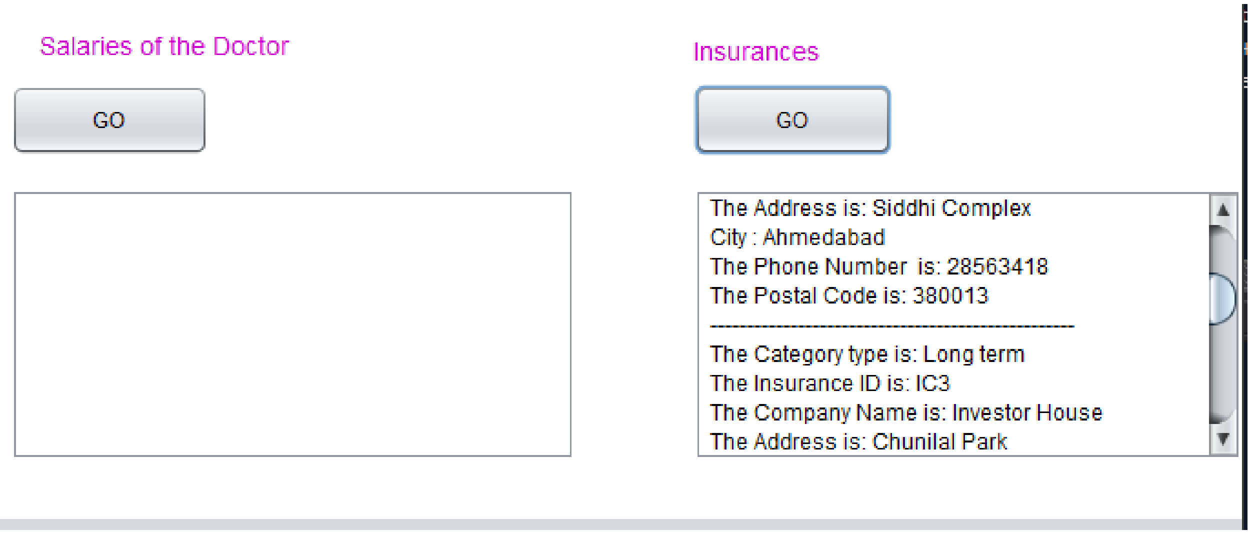
```
temp:=0;
OPEN c_dat;
LOOP
FETCH c_dat into r_dat;
EXIT WHEN c_dat%NOTFOUND;
if(r_dat.Category=category) then
    dbms_output.put_line('The Insurance Number is:'||r_dat.InsCo_ID);
    dbms_output.put_line('The Insurance Name is:'||r_dat.InsCo_Name);
    dbms_output.put_line('The Phone Number is:'||r_dat.Phone);
    dbms_output.put_line('The Address is:'||r_dat.Address);
    dbms_output.put_line('The City is:'||r_dat.City);
    dbms_output.put_line('The Postal Code is:'||r_dat.Zip);
dbms_output.put_line('===========================================
==');
temp:=temp+1;
end if;
end loop;
close c_dat;
    dbms_output.put_line('Total:'||temp);
    dbms_output.put_line('---------------------');
end loop;
close c_cat;
end;

declare
begin
InsuranceDisplay();
end;
```

→ Calling the Procedure in Java



```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    CallableStatement cstmt=null;
    ResultSet rs= null;
    connectionDB();

    String search_name="{call InsuranceFullDisplay(?)}";

    try{
        cstmt = conn.prepareCall(search_name);
       cstmt.registerOutParameter(1, OracleTypes.CURSOR);
       cstmt.executeUpdate();
        rs = (ResultSet) cstmt.getObject(1);
        int temp=0;
         while (rs.next()) {
            temp =temp+1;
            txtIns.append("\n--------------------------------------------------\n"+
                          "The Category type is: " +  rs.getString("Category")+
                          "\nThe Insurance ID is: " +  rs.getString("InsCo_Id")+
                          "\nThe Company Name is: " +  rs.getString("InsCo_Name")+
                          "\nThe Address is: " +  rs.getString("Address")+
                          "\nCity : " +  rs.getString("City")+
                          "\nThe Phone Number  is: " +  rs.getInt("Phone")+
                          "\nThe Postal Code is: " +  rs.getInt("Zip"));


         }

         txtSearch.setText("");
    }
 catch(SQLException e){
     e.printStackTrace();
 }finally{
        if(cstmt!=null){
```

→ Output



```
The Address is: Siddhi Complex
City : Ahmedabad
The Phone Number  is: 28563418
The Postal Code is: 380013
--------------------------------------------------
The Category type is: Long term
The Insurance ID is: IC3
The Company Name is: Investor House
The Address is: Chunilal Park
```

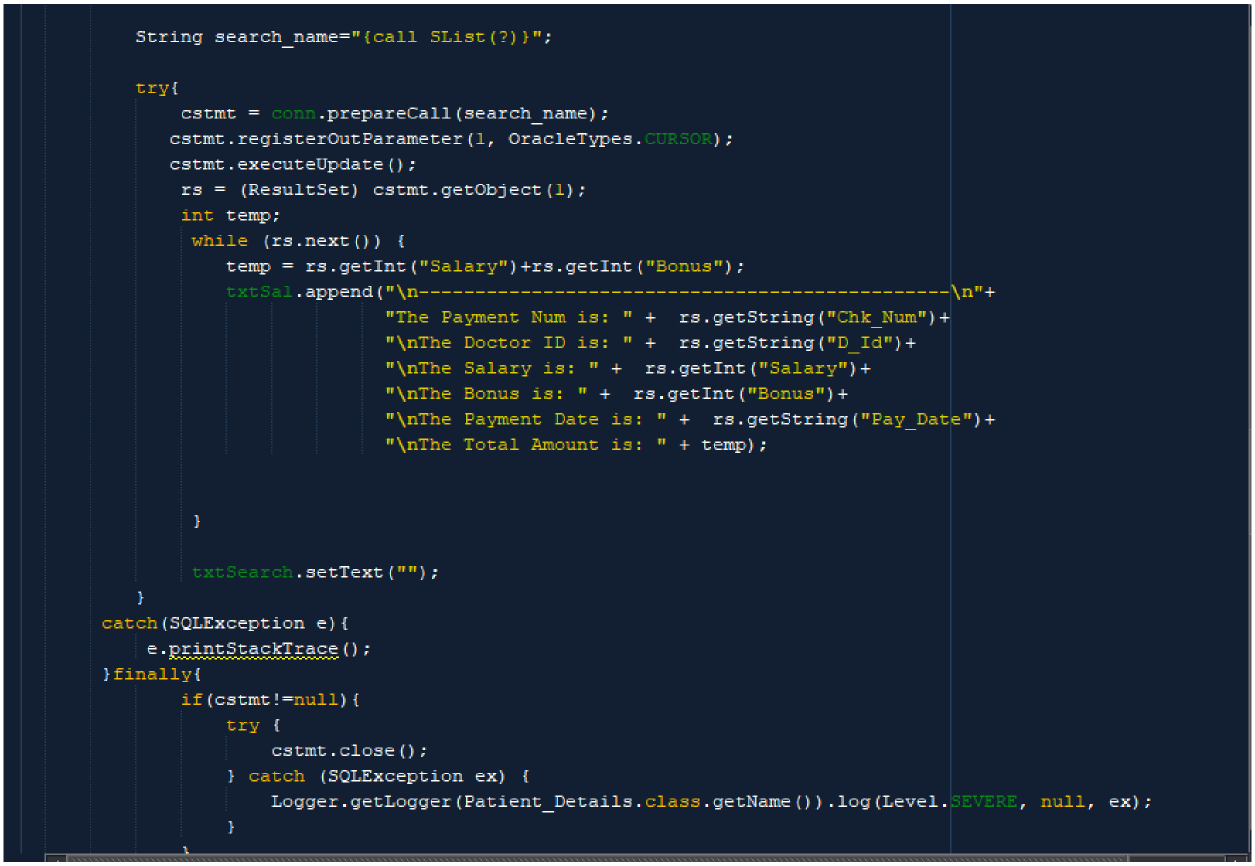
3. Displaying the total salaries of all doctors:

```
create or replace procedure SalaryList() as
cursor c_li is select * from Paycheck_Doc;
r_li c_li%ROWTYPE;
temp int;
begin
OPEN c_li;
LOOP
FETCH c_li into r_li;
EXIT WHEN c_li%NOTFOUND;
   temp:=r_li.Salary+r_li.Bonus;
   dbms_output.put_line('Cheque Number:'||r_li.Chk_Num);
   dbms_output.put_line('Doctor ID:'||r_li.D_Id);
   dbms_output.put_line('Salary:'||r_li.Salary);
   dbms_output.put_line('Bonus:'||r_li.Bonus);
   dbms_output.put_line('Paid Date:'||r_li.Pay_Date);
   dbms_output.put_line('Total Amount:'||temp);
   dbms_output.put_line('--------------------------');
end loop;
close c_li;
End;

declare
begin
SalaryList();
end;
```

→ For Calling in java a little change in the procedure

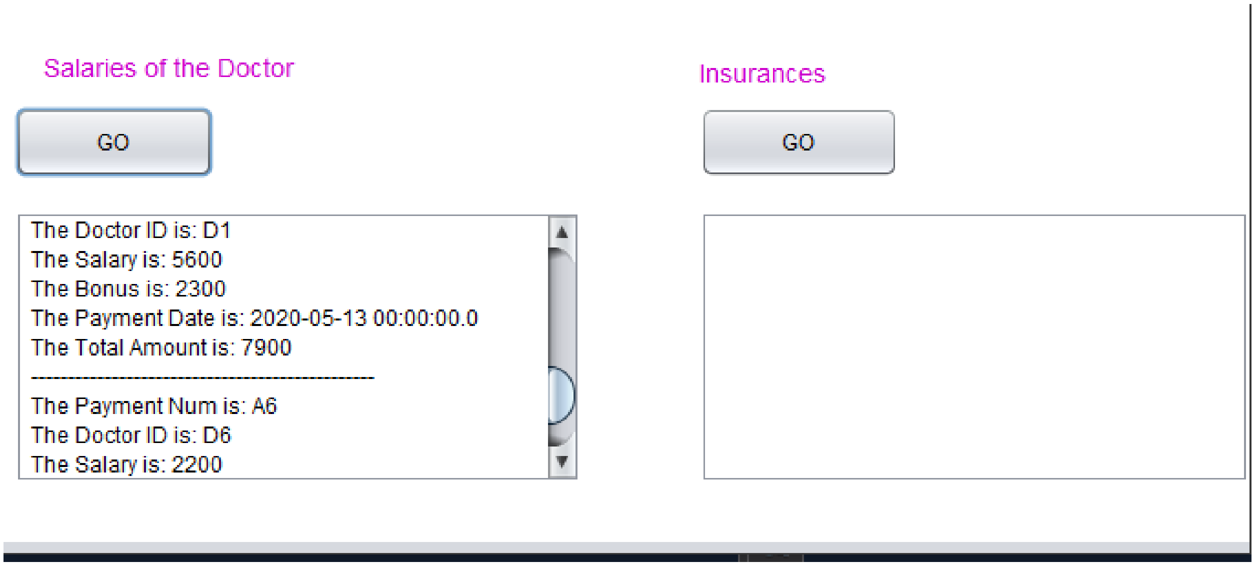
```
create or replace procedure SList(c_med out sys_refcursor) as
begin
open c_med for select * from Paycheck_Doc;
end;
```

→ Calling the Procedure in Java



```
        String search_name="{call SList(?)}";

        try{
            cstmt = conn.prepareCall(search_name);
           cstmt.registerOutParameter(1, OracleTypes.CURSOR);
           cstmt.executeUpdate();
            rs = (ResultSet) cstmt.getObject(1);
            int temp;
             while (rs.next()) {
                temp = rs.getInt("Salary")+rs.getInt("Bonus");
                txtSal.append("\n----------------------------------------------\n"+
                              "The Payment Num is: " +  rs.getString("Chk_Num")+
                              "\nThe Doctor ID is: " +  rs.getString("D_Id")+
                              "\nThe Salary is: " +  rs.getInt("Salary")+
                              "\nThe Bonus is: " +  rs.getInt("Bonus")+
                              "\nThe Payment Date is: " +  rs.getString("Pay_Date")+
                              "\nThe Total Amount is: " + temp);


             }

             txtSearch.setText("");
        }
     catch(SQLException e){
         e.printStackTrace();
     }finally{
            if(cstmt!=null){
                try {
                    cstmt.close();
                } catch (SQLException ex) {
                    Logger.getLogger(Patient_Details.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
```

→ Output:

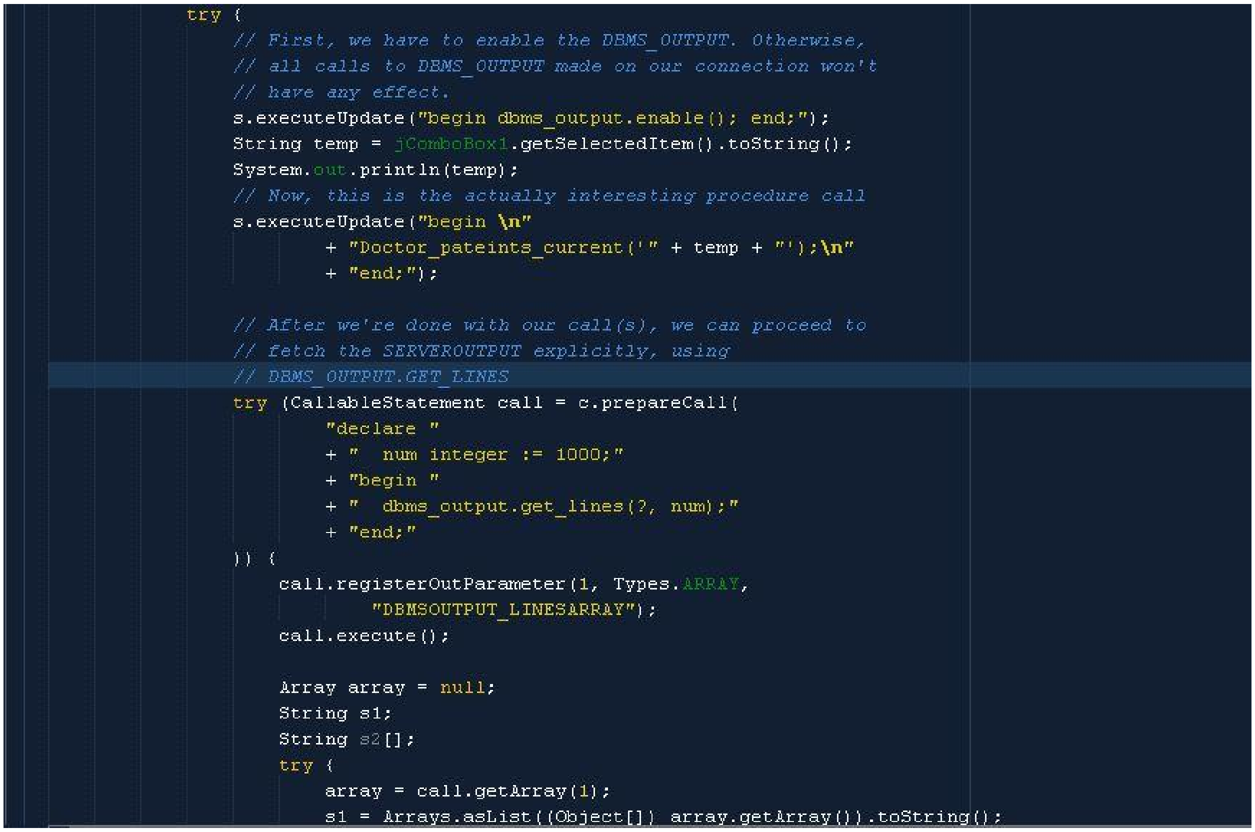


```
The Doctor ID is: D1
The Salary is: 5600
The Bonus is: 2300
The Payment Date is: 2020-05-13 00:00:00.0
The Total Amount is: 7900
----------------------------------------------
The Payment Num is: A6
The Doctor ID is: D6
The Salary is: 2200
```

4. For display number of patients and their patient a doctor is attending

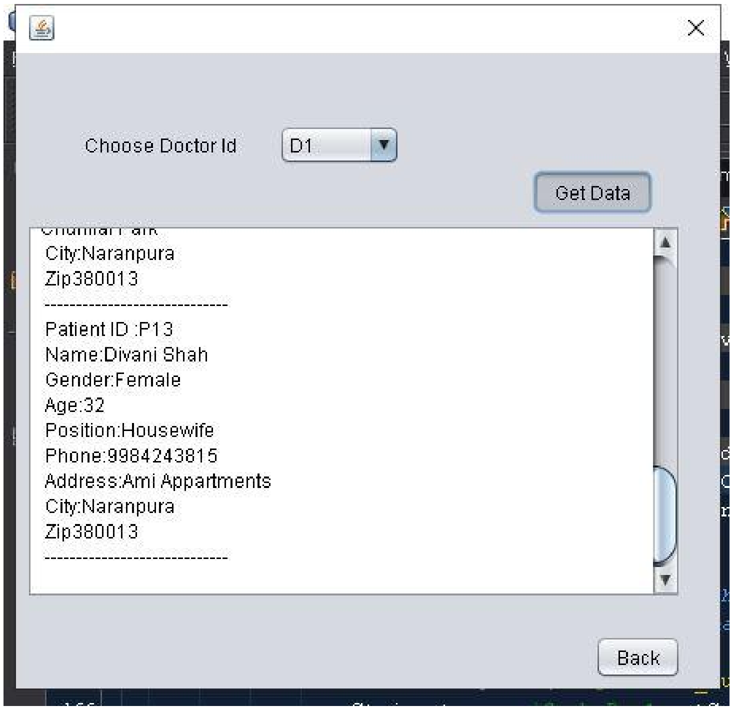
```
create or replace procedure Doctor_pateints_current(D varchar) as
cursor d_li is select
Patient.P_id,Patient.P_name,Patient.Gender,Patient.Age,Patient.Position,Patient.Phone,Patient.Address,Patient.City,Patient.Zip from
Patient,Patient_Doc where Patient_Doc.D_id=D AND
Patient.P_id=Patient_Doc.P_id;
p_li d_li%ROWTYPE;
temp int;
begin
select count(*) into temp from Patient_Doc where D_id=D;
dbms_output.put_line('total number of patient:'||temp);
  dbms_output.put_line('----------------------------');
OPEN d_li;
LOOP
FETCH d_li into p_li;
EXIT WHEN d_li%NOTFOUND;
  dbms_output.put_line('Patient ID :'||p_li.P_id);
  dbms_output.put_line('Name:'||p_li.P_name);
  dbms_output.put_line('Gender:'||p_li.Gender);
  dbms_output.put_line('Age:'||p_li.Age);
  dbms_output.put_line('Position:'||p_li.Position);
  dbms_output.put_line('Phone:'||p_li.Phone);
  dbms_output.put_line('Address:'||p_li.Address);
  dbms_output.put_line('City:'||p_li.City);
  dbms_output.put_line('Zip'||p_li.Zip);
  dbms_output.put_line('----------------------------');
end loop;
close d_li;
end;
```

→ Calling procedure in java :



```
try {
    // First, we have to enable the DBMS_OUTPUT. Otherwise,
    // all calls to DBMS_OUTPUT made on our connection won't
    // have any effect.
    s.executeUpdate("begin dbms_output.enable(); end;");
    String temp = jComboBox1.getSelectedItem().toString();
    System.out.println(temp);
    // Now, this is the actually interesting procedure call
    s.executeUpdate("begin \n"
            + "Doctor_pateints_current('" + temp + "');\n"
            + "end;");

    // After we're done with our call(s), we can proceed to
    // fetch the SERVEROUTPUT explicitly, using
    // DBMS_OUTPUT.GET_LINES
    try (CallableStatement call = c.prepareCall(
            "declare "
            + "  num integer := 1000;"
            + "begin "
            + "  dbms_output.get_lines(?, num);"
            + "end;"
    )) {
        call.registerOutParameter(1, Types.ARRAY,
                "DBMSOUTPUT_LINESARRAY");
        call.execute();

        Array array = null;
        String s1;
        String s2[];
        try {
            array = call.getArray(1);
            s1 = Arrays.asList((Object[]) array.getArray()).toString();
```

→ Output:



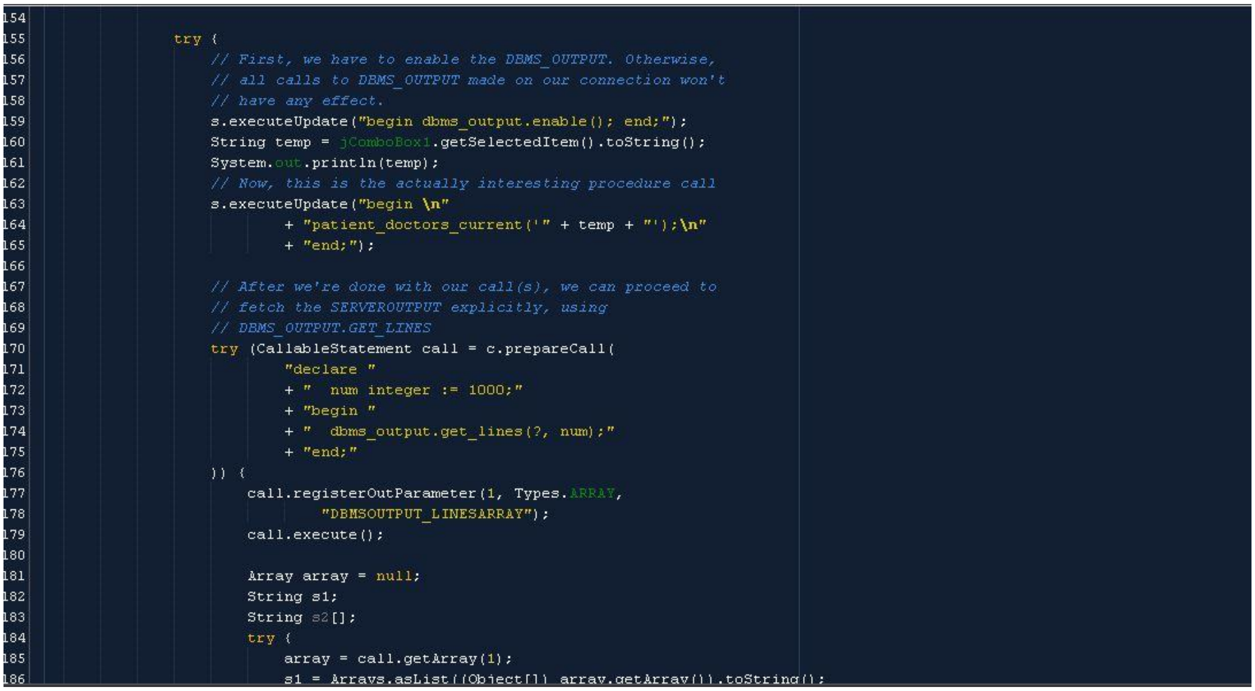
```
City:Naranpura
Zip380013
------------------------------
Patient ID :P13
Name:Divani Shah
Gender:Female
Age:32
Position:Housewife
Phone:9984243815
Address:Ami Appartments
City:Naranpura
Zip380013
------------------------------
```

5.For display number of doctor and their details a patient is meeting

```
create or replace procedure patient_doctors_current(P varchar) as
cursor p_li is select
Doctor.D_id,Doctor.D_name,Doctor.Gender,Doctor.Age,Doctor.Position,Doctor.Office,Doctor.Phone,Doctor.Address,Doctor.City,Doctor.Zip from
Doctor,Patient_Doc where Patient_Doc.P_id=P AND
Doctor.D_id=Patient_Doc.D_id;
d_li p_li%ROWTYPE;
temp int;
begin
select count(*) into temp from Patient_Doc where P_id=P;
dbms_output.put_line('total number of Doctors attending patient:'||temp);
   dbms_output.put_line('----------------------------');
OPEN p_li;
LOOP
FETCH p_li into d_li;
EXIT WHEN p_li%NOTFOUND;
   dbms_output.put_line('Doctor ID :'||d_li.D_id);
   dbms_output.put_line('Name:'||d_li.D_name);
   dbms_output.put_line('Gender:'||d_li.Gender);
   dbms_output.put_line('Age:'||d_li.Age);
   dbms_output.put_line('Position:'||d_li.Position);
   dbms_output.put_line('Phone:'||d_li.Phone);
   dbms_output.put_line('Office:'||d_li.Office);
   dbms_output.put_line('Address:'||d_li.Address);
   dbms_output.put_line('City:'||d_li.City);
   dbms_output.put_line('Zip'||d_li.Zip);
   dbms_output.put_line('----------------------------');
end loop;
close p_li;
end;
```

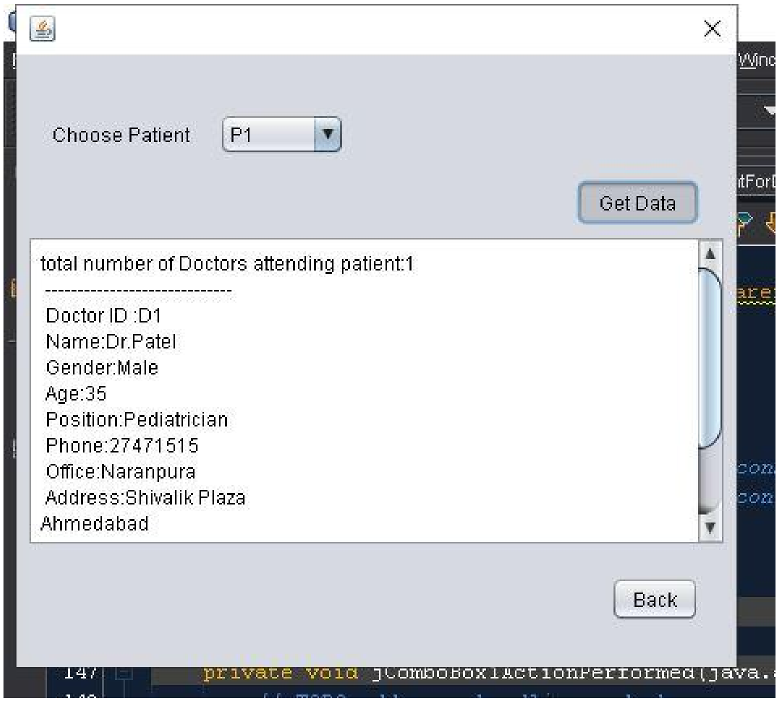
begin
patient_doctors_current('P5');
end;

→ Calling procedure in java:-



```
            try {
                // First, we have to enable the DBMS_OUTPUT. Otherwise,
                // all calls to DBMS_OUTPUT made on our connection won't
                // have any effect.
                s.executeUpdate("begin dbms_output.enable(); end;");
                String temp = jComboBox1.getSelectedItem().toString();
                System.out.println(temp);
                // Now, this is the actually interesting procedure call
                s.executeUpdate("begin \n"
                        + "patient_doctors_current('" + temp + "');\n"
                        + "end;");

                // After we're done with our call(s), we can proceed to
                // fetch the SERVEROUTPUT explicitly, using
                // DBMS_OUTPUT.GET_LINES
                try (CallableStatement call = c.prepareCall(
                        "declare "
                        + "  num integer := 1000;"
                        + "begin "
                        + "  dbms_output.get_lines(?, num);"
                        + "end;"
                )) {
                    call.registerOutParameter(1, Types.ARRAY,
                            "DBMSOUTPUT_LINESARRAY");
                    call.execute();

                    Array array = null;
                    String s1;
                    String s2[];
                    try {
                        array = call.getArray(1);
                        s1 = Arrays.asList((Object[]) array.getArray()).toString();
```

→ Output



Choose Patient P1

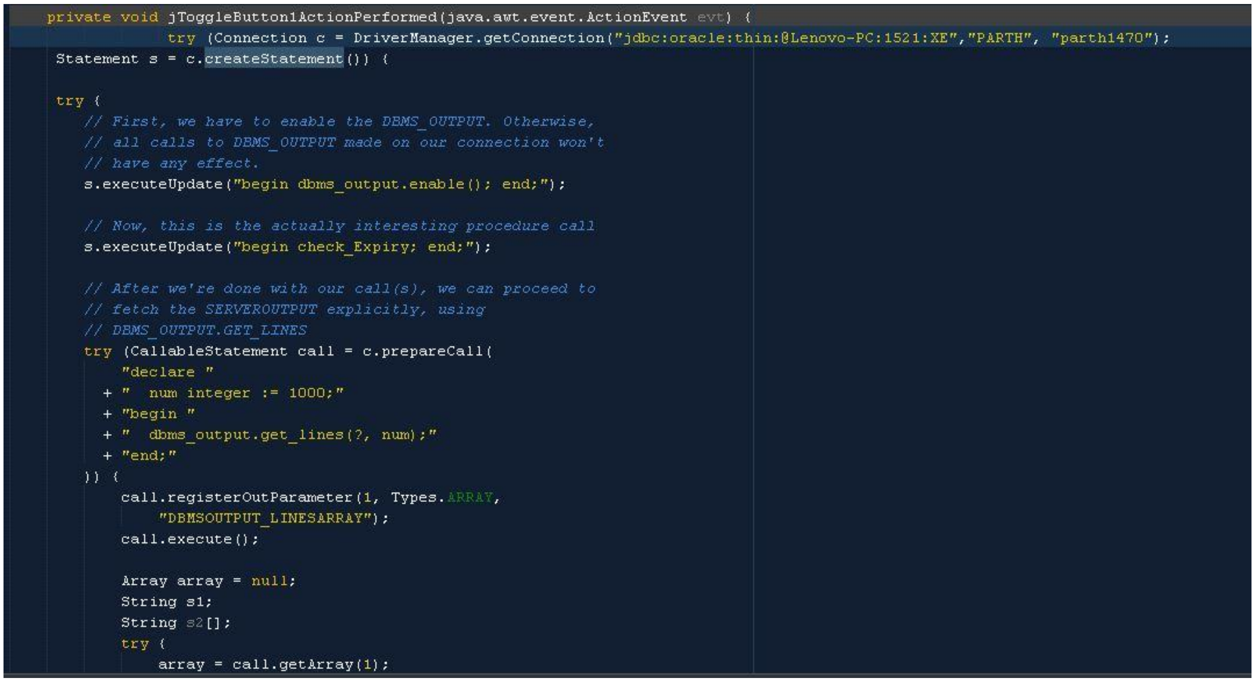
```
total number of Doctors attending patient:1
 -----------------------------
 Doctor ID :D1
 Name:Dr.Patel
 Gender:Male
 Age:35
 Position:Pediatrician
 Phone:27471515
 Office:Naranpura
 Address:Shivalik Plaza
Ahmedabad
```

6. For checking is there any medicine expired in catalogue and remove it as well

```
create or replace procedure check_Expiry as
cursor date_li is select * from Medicine;
med_li date_li%ROWTYPE;
current_Date date;
count_Current int:=0;
loss int :=0;
begin
   dbms_output.put_line('----------------------------');
select sysdate into current_Date from dual;
OPEN date_li;
LOOP
FETCH date_li into med_li;
EXIT WHEN date_li%NOTFOUND;
 if (med_li.Exp_Date <= current_Date ) then
   dbms_output.put_line('Medicine_id :'||med_li.M_id);
   dbms_output.put_line('Name :'||med_li.M_name);
   dbms_output.put_line('Manufacturer :'||med_li.Manufacturer);
   dbms_output.put_line('Price :'||med_li.Price);
   dbms_output.put_line('Quantity :'||med_li.Qty);
   dbms_output.put_line('Expiry date :'||med_li.Exp_Date);
   loss := loss +(med_li.Price*med_li.Qty);
   count_Current := count_Current +1;
   delete from Medicine_P where M_id=med_li.M_id;
   delete from Medicine where M_id=med_li.M_id;
 end if;
end loop;
close date_li;
 if (count_Current=0) then
   dbms_output.put_line('No medicine has been expired');
   dbms_output.put_line('----------------------------');
```

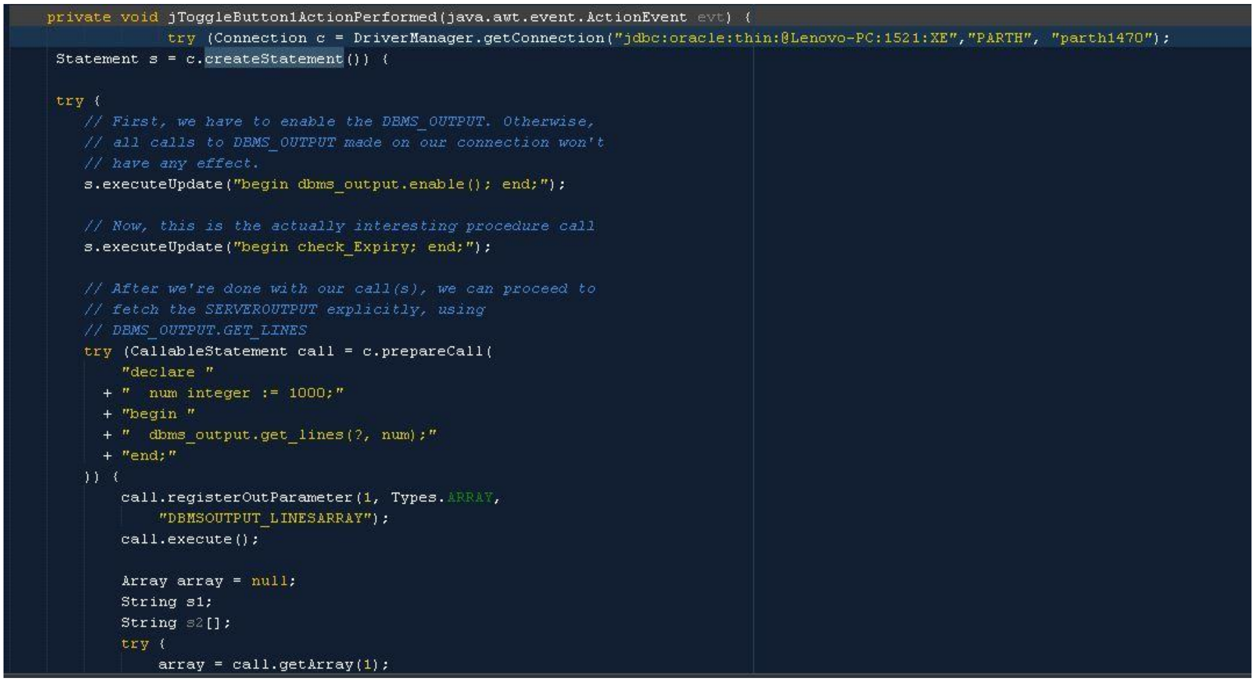
```
else
  dbms_output.put_line('-----------------------------');
    dbms_output.put_line('Total number of Medicines removed are
'||count_Current);
  dbms_output.put_line('-----------------------------');
  dbms_output.put_line('Total loss of rupees '||loss);
end if;
end;

begin
check_Expiry;
end;
```

→ Calling procedure in java :-



```
private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
            try (Connection c = DriverManager.getConnection("jdbc:oracle:thin:@Lenovo-PC:1521:XE","PARTH", "parth1470");
 Statement s = c.createStatement()) {

 try {
   // First, we have to enable the DBMS_OUTPUT. Otherwise,
   // all calls to DBMS_OUTPUT made on our connection won't
   // have any effect.
   s.executeUpdate("begin dbms_output.enable(); end;");

   // Now, this is the actually interesting procedure call
   s.executeUpdate("begin check_Expiry; end;");

   // After we're done with our call(s), we can proceed to
   // fetch the SERVEROUTPUT explicitly, using
   // DBMS_OUTPUT.GET_LINES
   try (CallableStatement call = c.prepareCall(
       "declare "
     + "  num integer := 1000;"
     + "begin "
     + "  dbms_output.get_lines(?, num);"
     + "end;"
   )) {
       call.registerOutParameter(1, Types.ARRAY,
           "DBMSOUTPUT_LINESARRAY");
       call.execute();

       Array array = null;
       String s1;
       String s2[];
       try {
           array = call.getArray(1);
```

Output :-



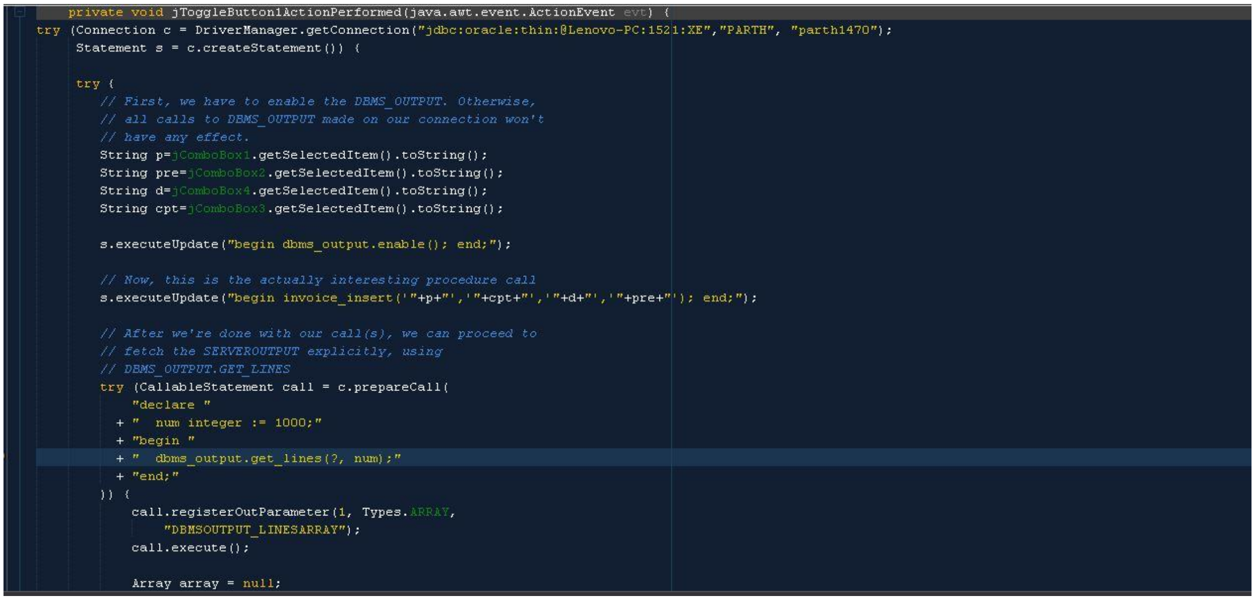
```
private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
                try (Connection c = DriverManager.getConnection("jdbc:oracle:thin:@Lenovo-PC:1521:XE","PARTH", "parth1470");
 Statement s = c.createStatement()) {

 try {
    // First, we have to enable the DBMS_OUTPUT. Otherwise,
    // all calls to DBMS_OUTPUT made on our connection won't
    // have any effect.
    s.executeUpdate("begin dbms_output.enable(); end;");

    // Now, this is the actually interesting procedure call
    s.executeUpdate("begin check_Expiry; end;");

    // After we're done with our call(s), we can proceed to
    // fetch the SERVEROUTPUT explicitly, using
    // DBMS_OUTPUT.GET_LINES
    try (CallableStatement call = c.prepareCall(
        "declare "
      + "  num integer := 1000;"
      + "begin "
      + "  dbms_output.get_lines(?, num);"
      + "end;"
    )) {
        call.registerOutParameter(1, Types.ARRAY,
            "DBMSOUTPUT_LINESARRAY");
        call.execute();

        Array array = null;
        String s1;
        String s2[];
        try {
            array = call.getArray(1);
```

7. To generate an invoice of a patient according to prescription.

create or replace procedure invoice_insert(Patient_id varchar,CPT_id varchar,Diagnosis_id varchar,Prescrip_id varchar)
as
Med_qua int;
amount int :=0;
Med_price int;
Invoice_Dte Date;
due_Dte Date;
temp2 int;
temp5 int;
temp3 varchar(4);
temp4 varchar(5);
temp6 varchar(1):='I';
Stock int;

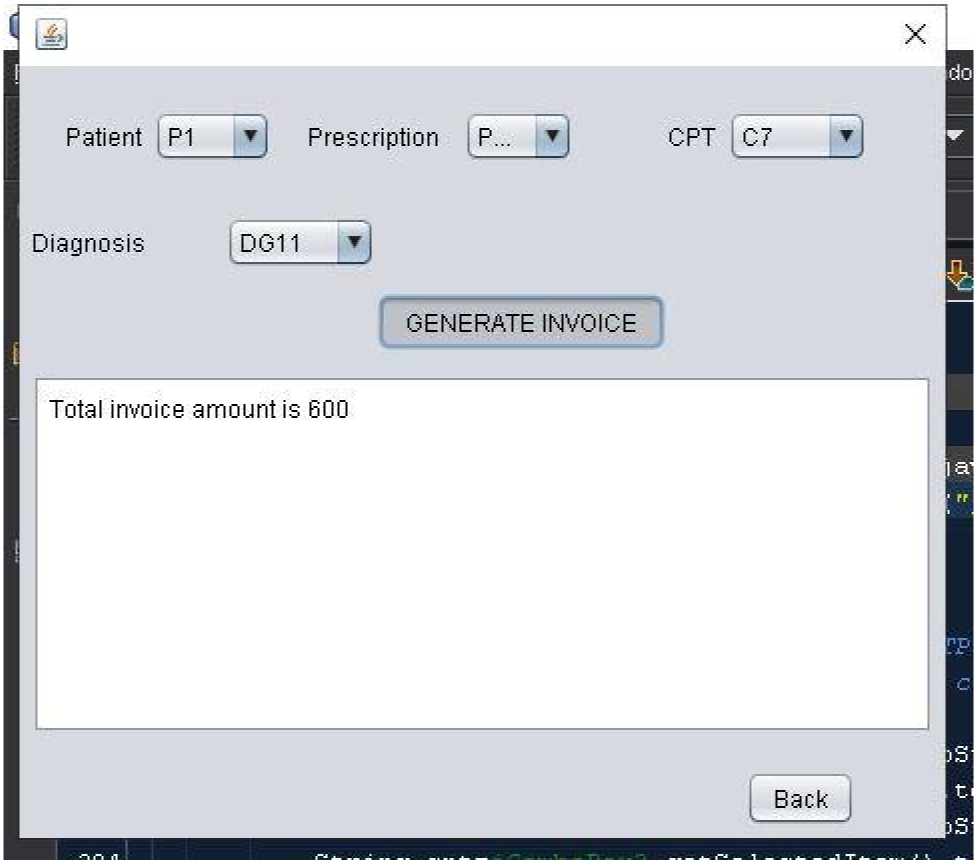
```
cursor med_prescription iS select M_id from Medicine_P where
Prescription_ID=Prescrip_id;
temp_id  med_prescription%ROWTYPE;
begin
  SELECT sysdate into Invoice_Dte from dual;
  select Medicine_Q into Med_qua from Prescription where
Prescription_ID=Prescrip_id;
  OPEN med_prescription;
  LOOP
FETCH med_prescription into temp_id;
    EXIT WHEN med_prescription%notfound;
    select Price,Qty into Med_price,Stock from Medicine where
M_ID=temp_id.M_id;
    amount:= amount + (Med_price*Med_qua);
    update Medicine_P set Qty= Stock- Med_qua where
M_id=temp_id.M_id;
    update Medicine set Qty= Stock- Med_qua where M_id=temp_id.M_id;
  END loop;

  CLOSE med_prescription;
select count(*) into temp2 from Invoice;

dbms_output.put_line('Total invoice amount is '||amount);
temp5:= temp2 + 1 ;
temp3:= TO_CHAR(temp5);
temp4:= CONCAT(temp6,temp3);
due_Dte:=ADD_MONTHS(Invoice_Dte,2);
insert into invoice values (temp4,amount,Invoice_dte,due_Dte);
insert into Invoice_P values
(temp4,Patient_id,CPT_id,Diagnosis_id,Prescrip_id,amount,Invoice_dte,du
e_Dte);
End;
```

Calling procedure in java :-



```
    private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try (Connection c = DriverManager.getConnection("jdbc:oracle:thin:@Lenovo-PC:1521:XE","PARTH", "parth1470");
     Statement s = c.createStatement()) {

     try {
        // First, we have to enable the DBMS_OUTPUT. Otherwise,
        // all calls to DBMS_OUTPUT made on our connection won't
        // have any effect.
        String p=jComboBox1.getSelectedItem().toString();
        String pre=jComboBox2.getSelectedItem().toString();
        String d=jComboBox4.getSelectedItem().toString();
        String cpt=jComboBox3.getSelectedItem().toString();

        s.executeUpdate("begin dbms_output.enable(); end;");

        // Now, this is the actually interesting procedure call
        s.executeUpdate("begin invoice_insert('"+p+"','"+cpt+"','"+d+"','"+pre+"'); end;");

        // After we're done with our call(s), we can proceed to
        // fetch the SERVEROUTPUT explicitly, using
        // DBMS_OUTPUT.GET_LINES
        try (CallableStatement call = c.prepareCall(
            "declare "
          + "  num integer := 1000;"
          + "begin "
          + "  dbms_output.get_lines(?, num);"
          + "end;"
        )) {
            call.registerOutParameter(1, Types.ARRAY,
                "DBMSOUTPUT_LINESARRAY");
            call.execute();

            Array array = null;
```

Output:-



Patient P1 Prescription P... CPT C7

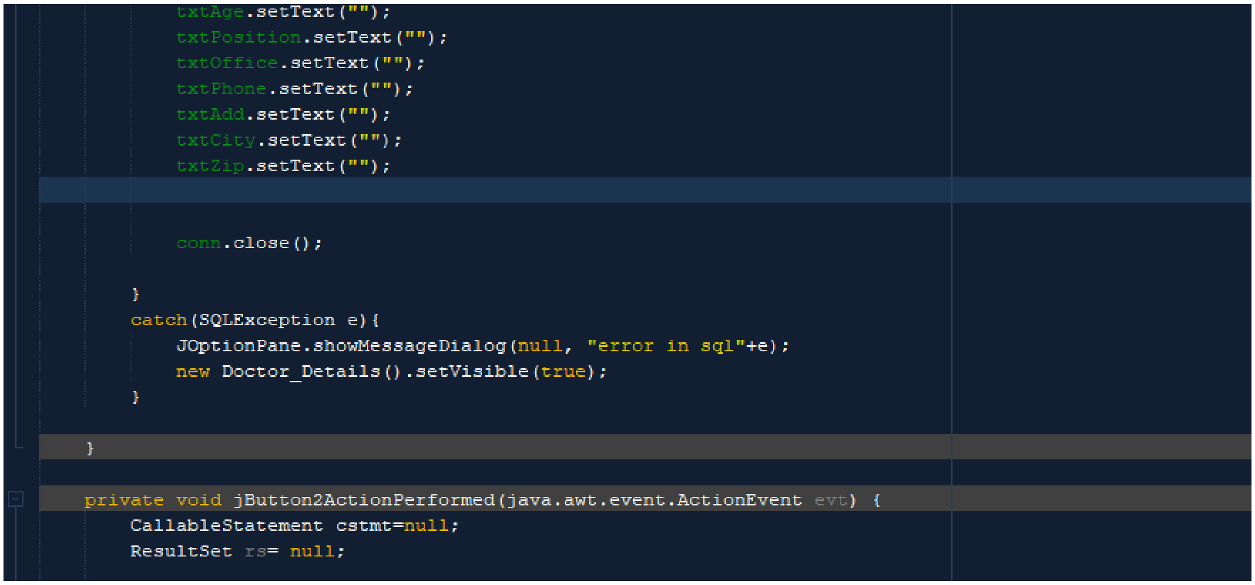
Diagnosis DG11

```
Total invoice amount is 600
```

- # Triggers :-

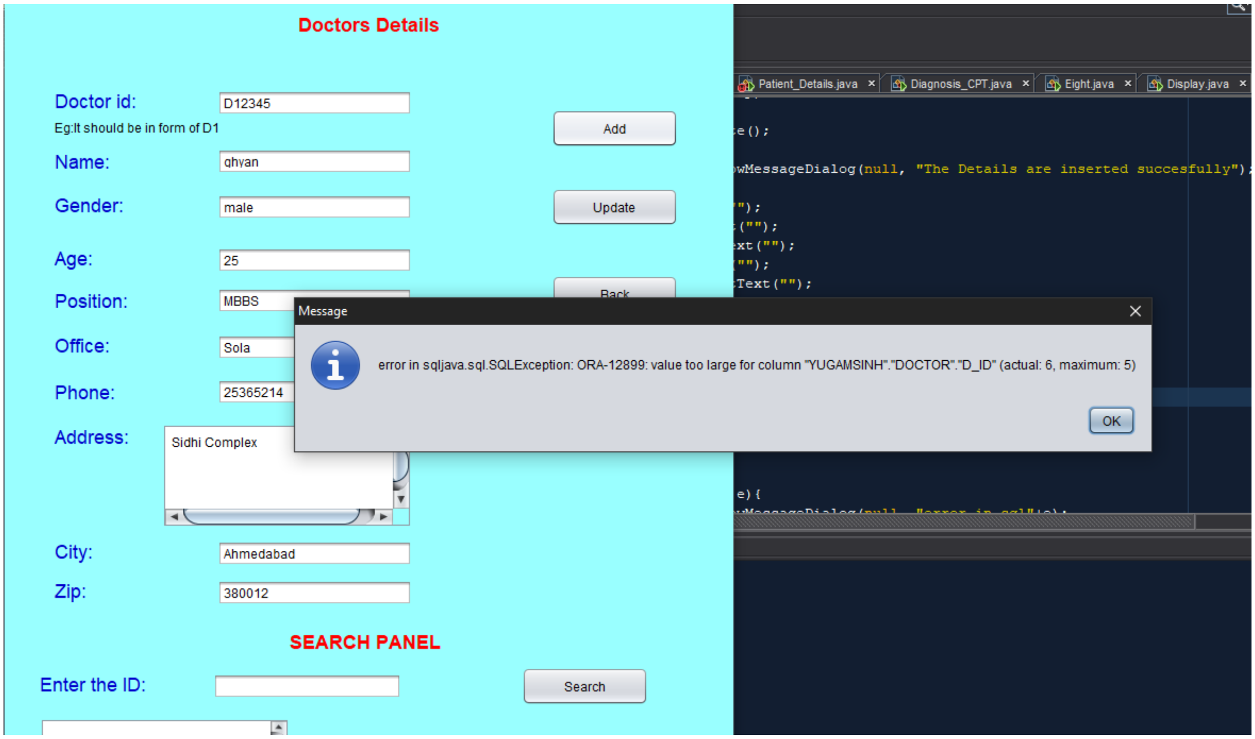
- ## For Doctor Table

→ Calling the Trigger in Java



```
                txtAge.setText("");
                txtPosition.setText("");
                txtOffice.setText("");
                txtPhone.setText("");
                txtAdd.setText("");
                txtCity.setText("");
                txtZip.setText("");


                conn.close();

            }
            catch(SQLException e){
                JOptionPane.showMessageDialog(null, "error in sql"+e);
                new Doctor_Details().setVisible(true);
            }

        }

        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
            CallableStatement cstmt=null;
            ResultSet rs= null;
```

```
create or replace trigger trD before insert or update on Doctor
for each row
begin
if(length(:new.D_id)>=5) then
    raise_application_error(-20001,'Please enter within range of 4 chars
preferably');
end if;
End;
```



Doctor id: D12345

Eg:It should be in form of D1

Name: ghyan

Gender: male

Age: 25

Position: MBBS

Office: Sola

Phone: 25365214

Address: Sidhi Complex

City: Ahmedabad

Zip: 380012

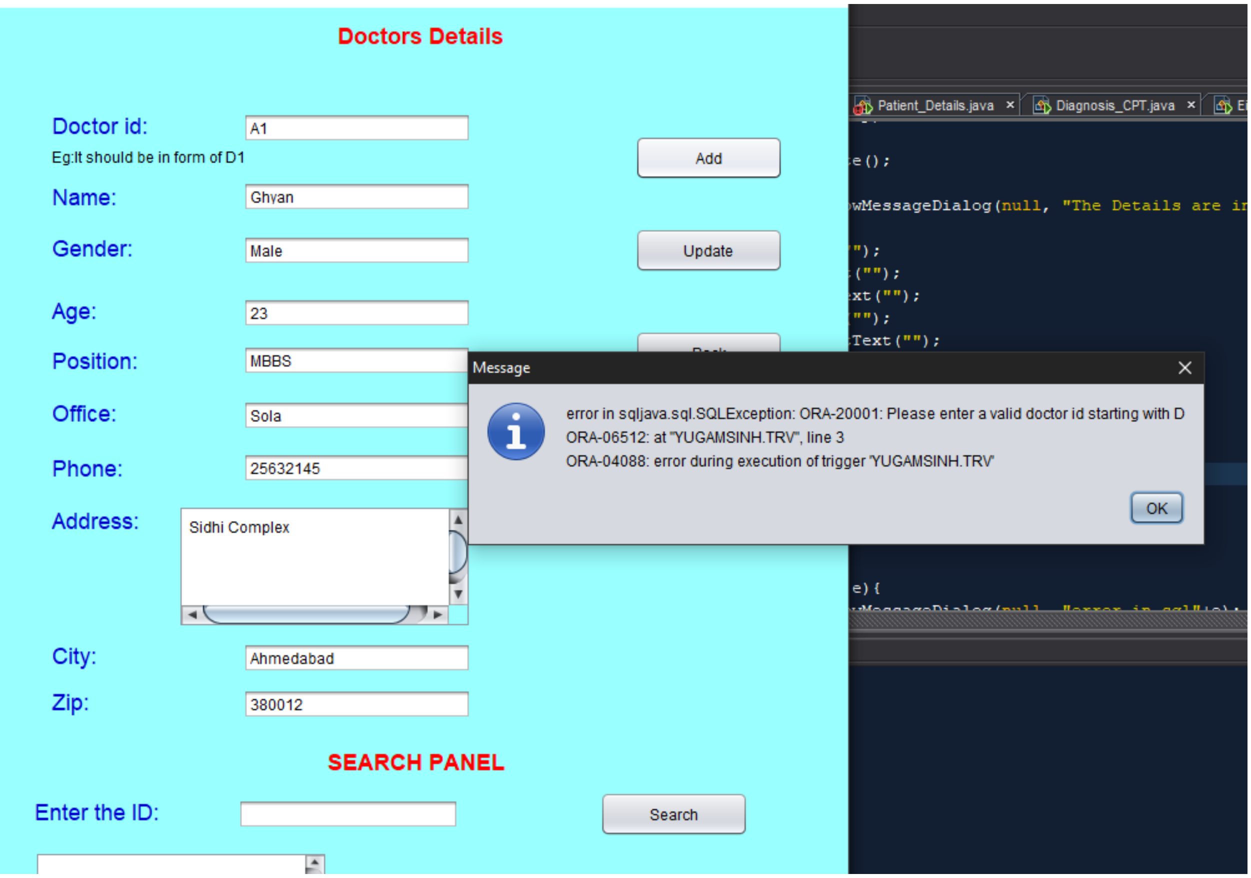
Enter the ID:

Message

error in sqljava.sql.SQLException: ORA-12899: value too large for column "YUGAMSINH"."DOCTOR"."D_ID" (actual: 6, maximum: 5)

```
e();

wMessageDialog(null, "The Details are inserted succesfully");

");
("");
xt("");
("");
Text("");

e){
```

→ create or replace trigger trV before insert or update on Doctor for each row
begin
if (substr(:new.D_id,1,1) <> 'D') then
raise_application_error(-20001,'Please enter a valid doctor id starting with D');
end if;
end;



Doctors Details

Doctor id: A1

Eg:It should be in form of D1

Name: Ghyan

Gender: Male

Age: 23

Position: MBBS

Office: Sola

Phone: 25632145

Address: Sidhi Complex

City: Ahmedabad

Zip: 380012

SEARCH PANEL

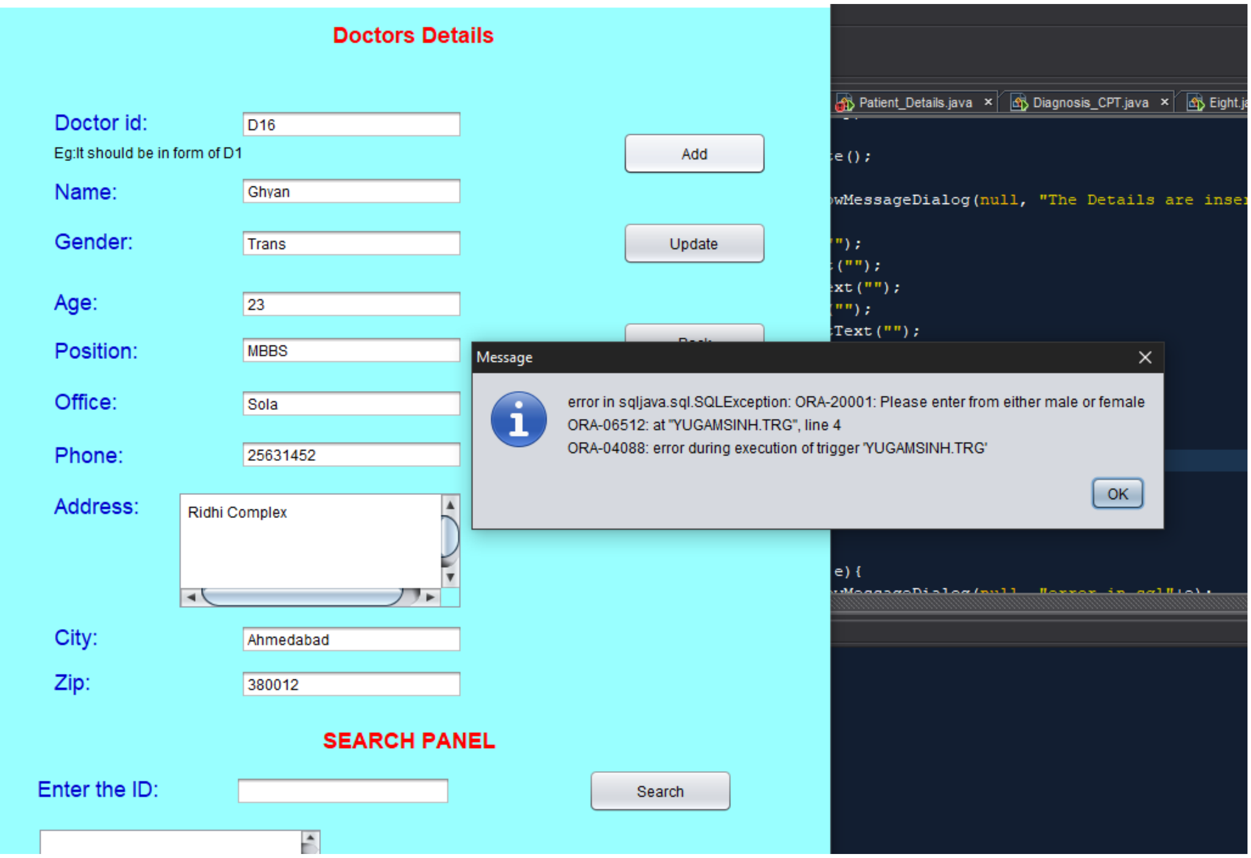
Enter the ID:

Message

```
error in sqljava.sql.SQLException: ORA-20001: Please enter a valid doctor id starting with D
ORA-06512: at "YUGAMSINH.TRV", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRV'
```

```
→ create or replace trigger trG before insert or update on Doctor
for each row
begin
if inserting then
      if(:new.Gender<>'Female' and :new.Gender<>'Male' and
:new.Gender<>'female' and :new.Gender<>'male') then
               raise_application_error(-20001,'Please enter from either male or
female');
end if;
end if;
if updating then
      if(:new.Gender<>'Female' and :new.Gender<>'Male' and
:new.Gender<>'female' and :new.Gender<>'male') then
   raise_application_error(-20001,'Please enter from either male or female');
```

```
end if;
end if;
end;
```



Doctors Details

Doctor id: D16

Eg:It should be in form of D1

Name: Ghyan

Gender: Trans

Age: 23

Position: MBBS

Office: Sola

Phone: 25631452

Address: Ridhi Complex

City: Ahmedabad

Zip: 380012

SEARCH PANEL

Enter the ID:

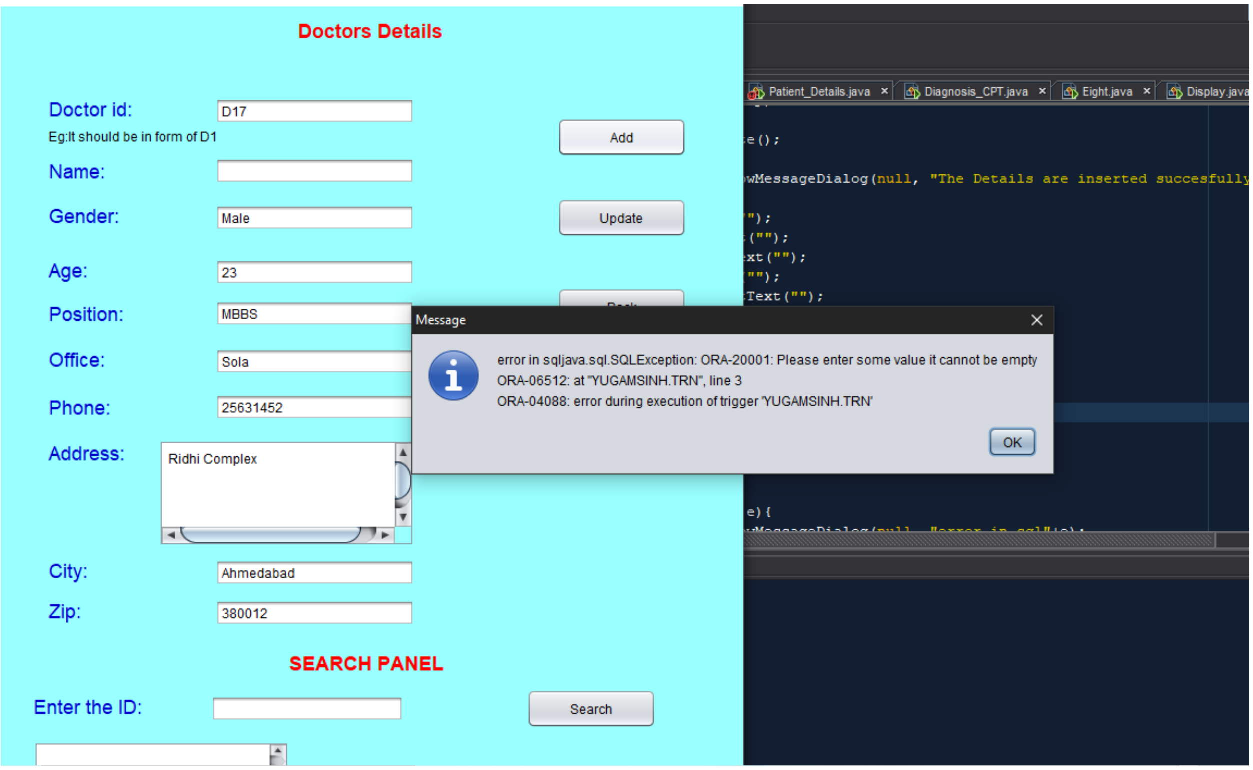
Message

error in sqljava.sql.SQLException: ORA-20001: Please enter from either male or female
ORA-06512: at "YUGAMSINH.TRG", line 4
ORA-04088: error during execution of trigger 'YUGAMSINH.TRG'

```
e();
wMessageDialog(null, "The Details are inser
");
("");
xt("");
("");
Text("");
e){
```

→ create or replace trigger trN before insert or update on Doctor for each row

```
begin
	if(:new.D_name is null) then
		raise_application_error(-20001,'Please enter some value it cannot be empty');
	end if;
end;
```



Doctors Details

Doctor id:	D17
Eg:It should be in form of D1	
Name:	
Gender:	Male
Age:	23
Position:	MBBS
Office:	Sola
Phone:	25631452
Address:	Ridhi Complex
City:	Ahmedabad
Zip:	380012

SEARCH PANEL

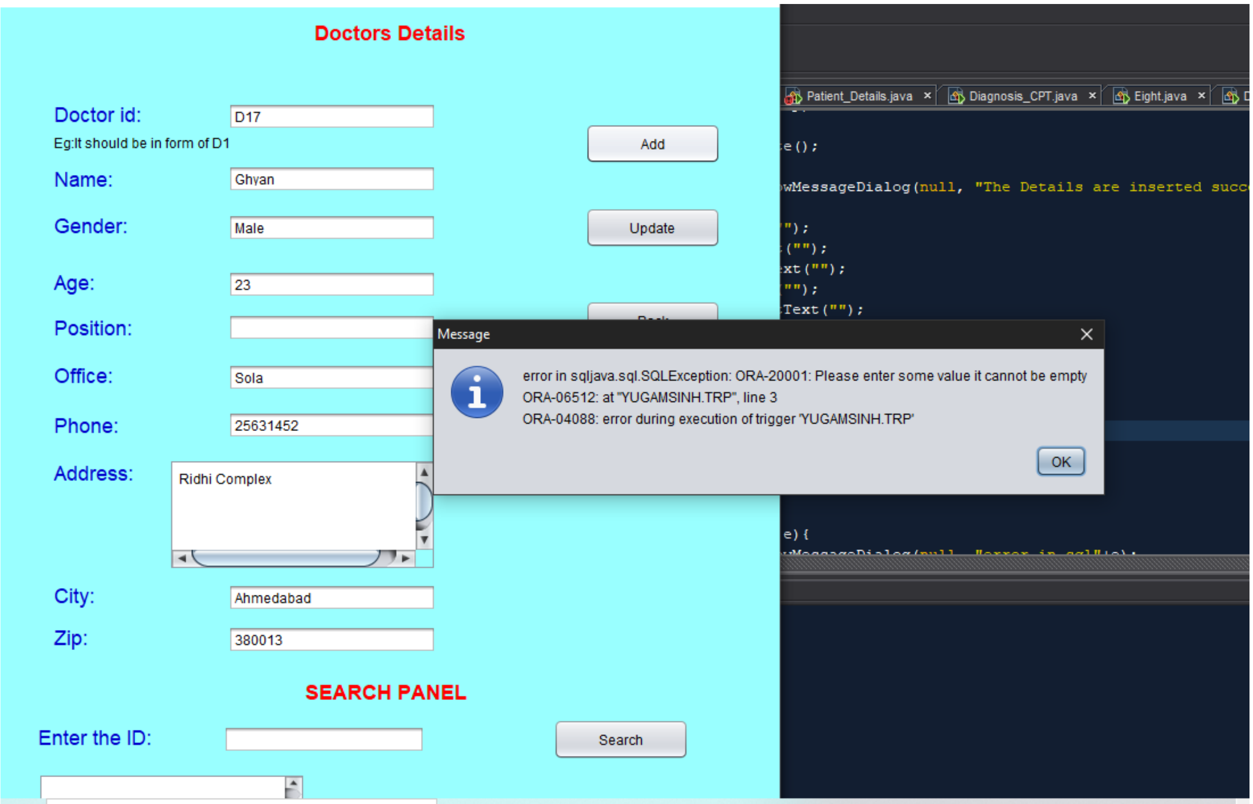
Enter the ID:

Message

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRN", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRN'
```

```
e();
wMessageDialog(null, "The Details are inserted succesfully
");
("");
xt("");
"");
Text("");
e){
```

→
```
create or replace trigger trP before insert or update on Doctor for each row
begin
        if(:new.Position is null) then
                raise_application_error(-20001,'Please enter some value it cannot be empty');
        end if;
end;
```



Doctors Details

Doctor id: D17
Eg:It should be in form of D1
Name: Ghyan
Gender: Male
Age: 23
Position:
Office: Sola
Phone: 25631452
Address: Ridhi Complex
City: Ahmedabad
Zip: 380013

SEARCH PANEL

Enter the ID:

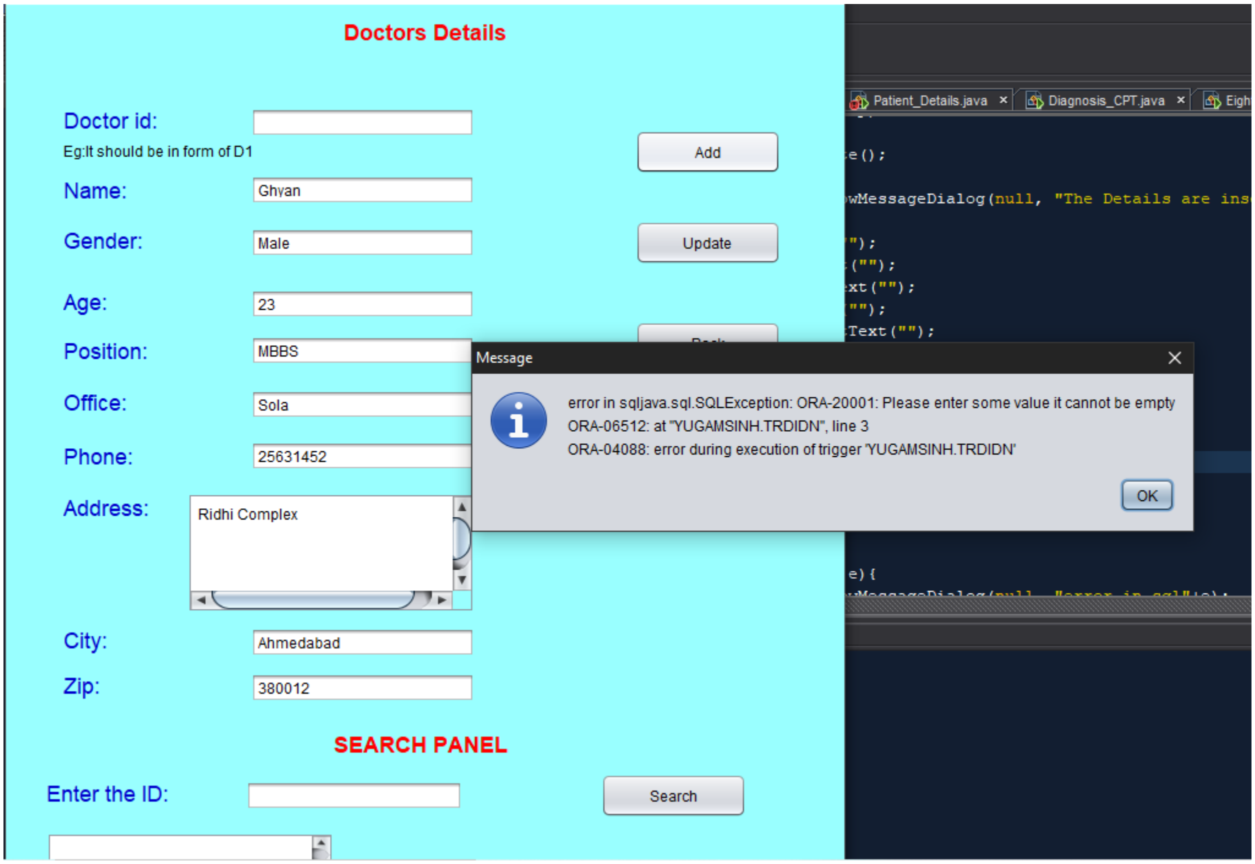
Message

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRP", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRP'
```

```
e();
wMessageDialog(null, "The Details are inserted succ
");
("");
xt("");
("");
Text("");
e){
```

→ create or replace trigger trDIDN before insert or update on Doctor for each row

```
begin
    if(:new.D_id is null) then
        raise_application_error(-20001,'Please enter some value it cannot be empty');
    end if;
end;
```



Doctors Details

Doctor id:	
Eg:It should be in form of D1	
Name:	Ghyan
Gender:	Male
Age:	23
Position:	MBBS
Office:	Sola
Phone:	25631452
Address:	Ridhi Complex
City:	Ahmedabad
Zip:	380012

SEARCH PANEL

Enter the ID:

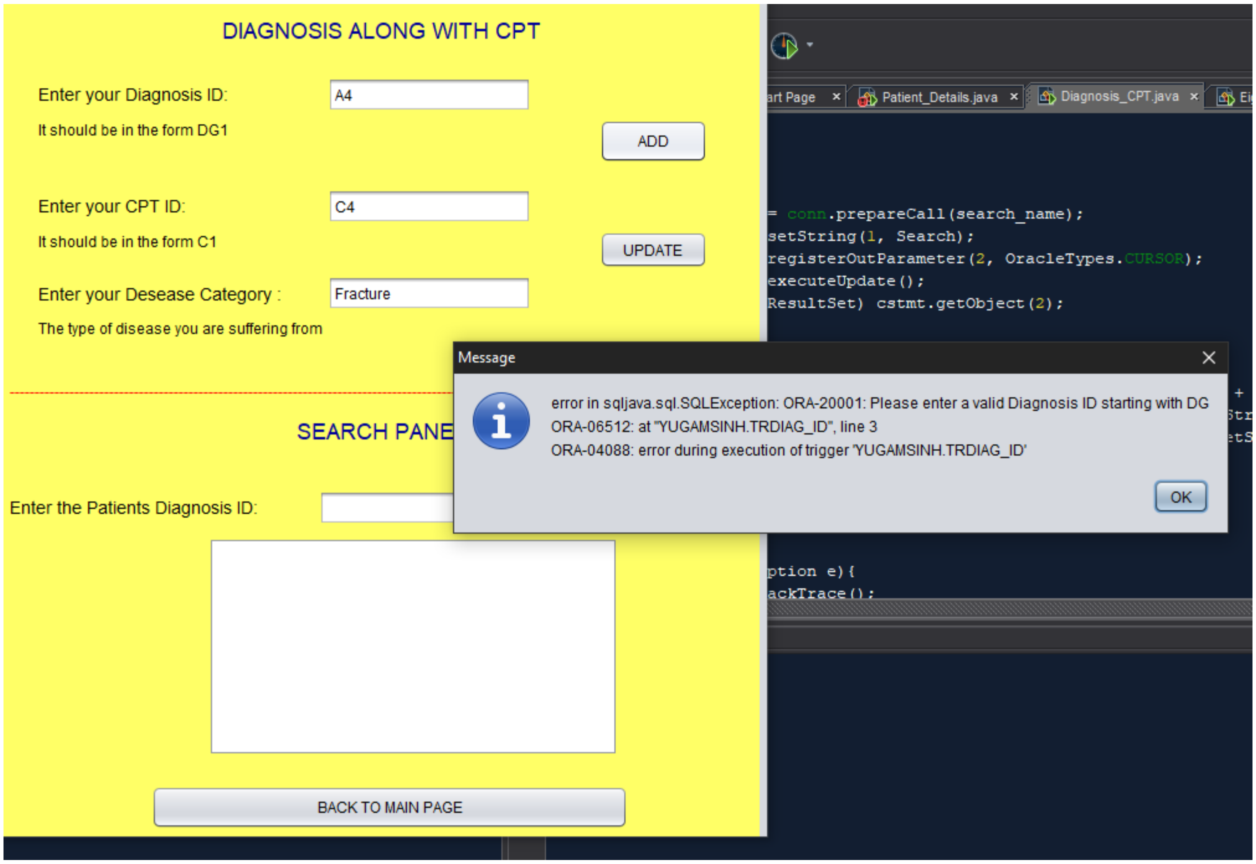
```
e();
wMessageDialog(null, "The Details are ins
");
("");
xt("");
("");
Text("");
e){
```

Message

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRDIDN", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRDIDN'
```

- **Diagnosis Table**

```
→ create or replace trigger trDiag_ID before insert or update on Diagnosis for each row
begin
if (substr(:new.Diagnosis_ID,1,2) <> 'DG') then
raise_application_error(-20001,'Please enter a valid Diagnosis ID starting with DG');
end if;
end;
```



Enter your Diagnosis ID: A4

It should be in the form DG1

Enter your CPT ID: C4

It should be in the form C1

Enter your Desease Category : Fracture

The type of disease you are suffering from

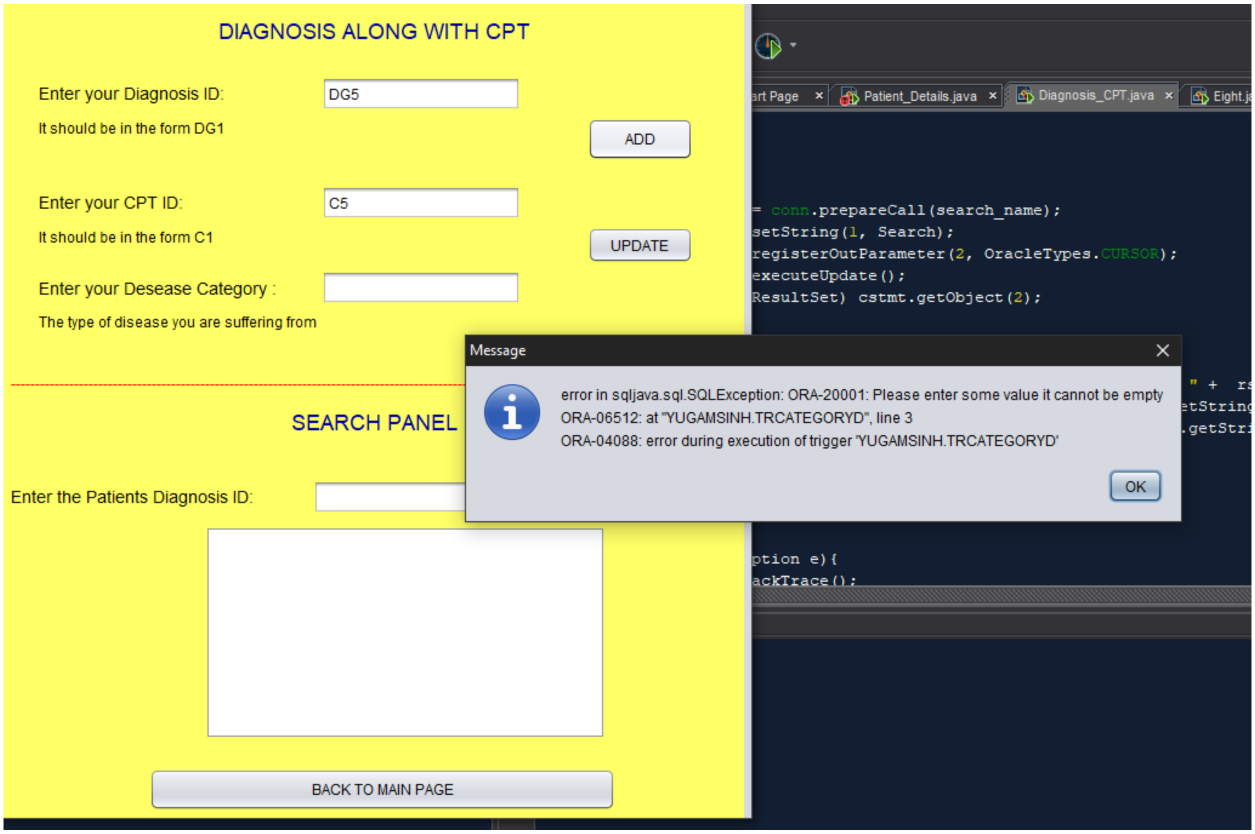
Enter the Patients Diagnosis ID:

```
= conn.prepareCall(search_name);
setString(1, Search);
registerOutParameter(2, OracleTypes.CURSOR);
executeUpdate();
ResultSet) cstmt.getObject(2);
```

```
error in sqljava.sql.SQLException: ORA-20001: Please enter a valid Diagnosis ID starting with DG
ORA-06512: at "YUGAMSINH.TRDIAG_ID", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRDIAG_ID'
```

```
ption e){
ackTrace();
```

```
→ create or replace trigger trDiagD before insert or update on Diagnosis for each row
begin
      if(:new.Diagnosis_ID is null) then
          raise_application_error(-20001,'Please enter some value it cannot be empty');
      end if;
end;
```



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG5

It should be in the form DG1

Enter your CPT ID: C5

It should be in the form C1

Enter your Desease Category :

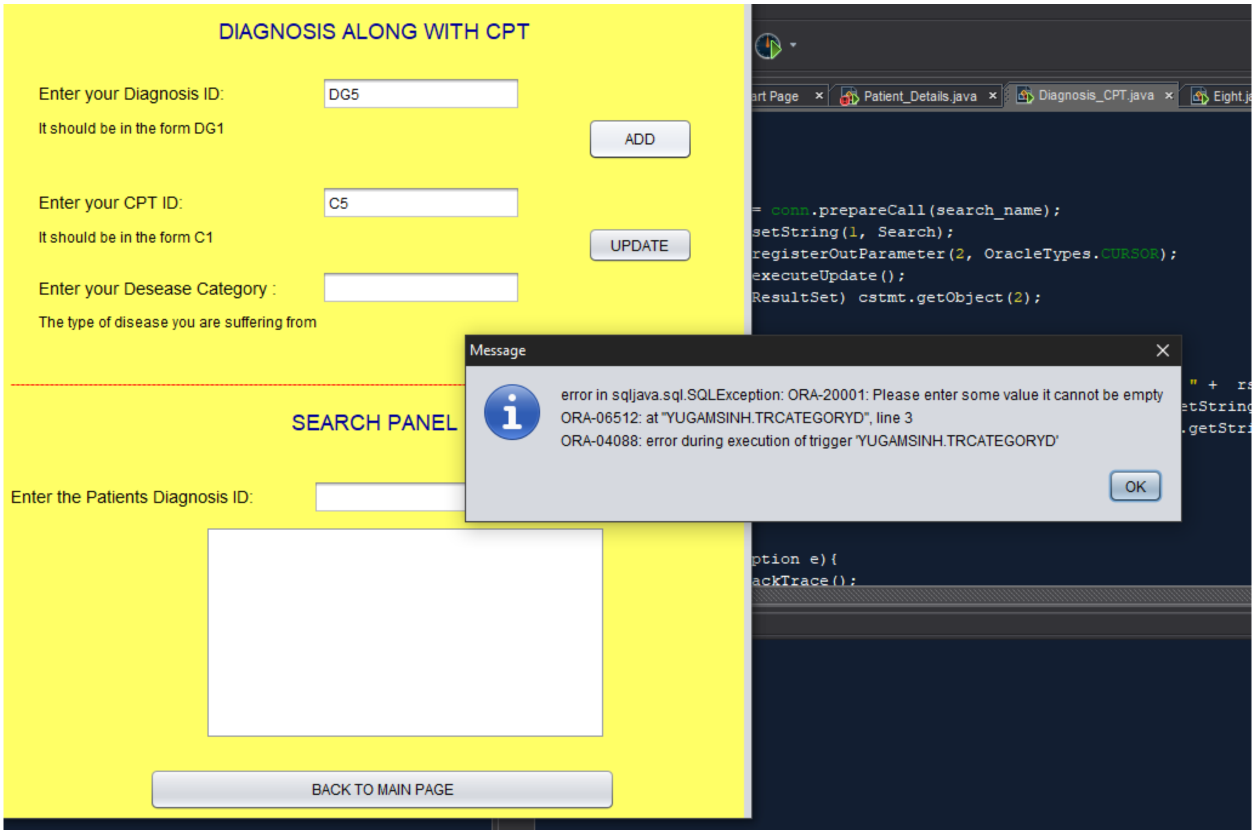
The type of disease you are suffering from

Enter the Patients Diagnosis ID:

```
= conn.prepareCall(search_name);
setString(1, Search);
registerOutParameter(2, OracleTypes.CURSOR);
executeUpdate();
ResultSet) cstmt.getObject(2);
" + rs
etString
.getStri
ption e){
ackTrace();
```

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRCATEGORYD", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRCATEGORYD'
```

```
→ create or replace trigger trCategoryD before insert or update on Diagnosis for each row
begin
      if(:new.Category is null) then
          raise_application_error(-20001,'Please enter some value it cannot be empty');
      end if;
end;
```



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG5

It should be in the form DG1

Enter your CPT ID: C5

It should be in the form C1

Enter your Desease Category :

The type of disease you are suffering from

SEARCH PANEL

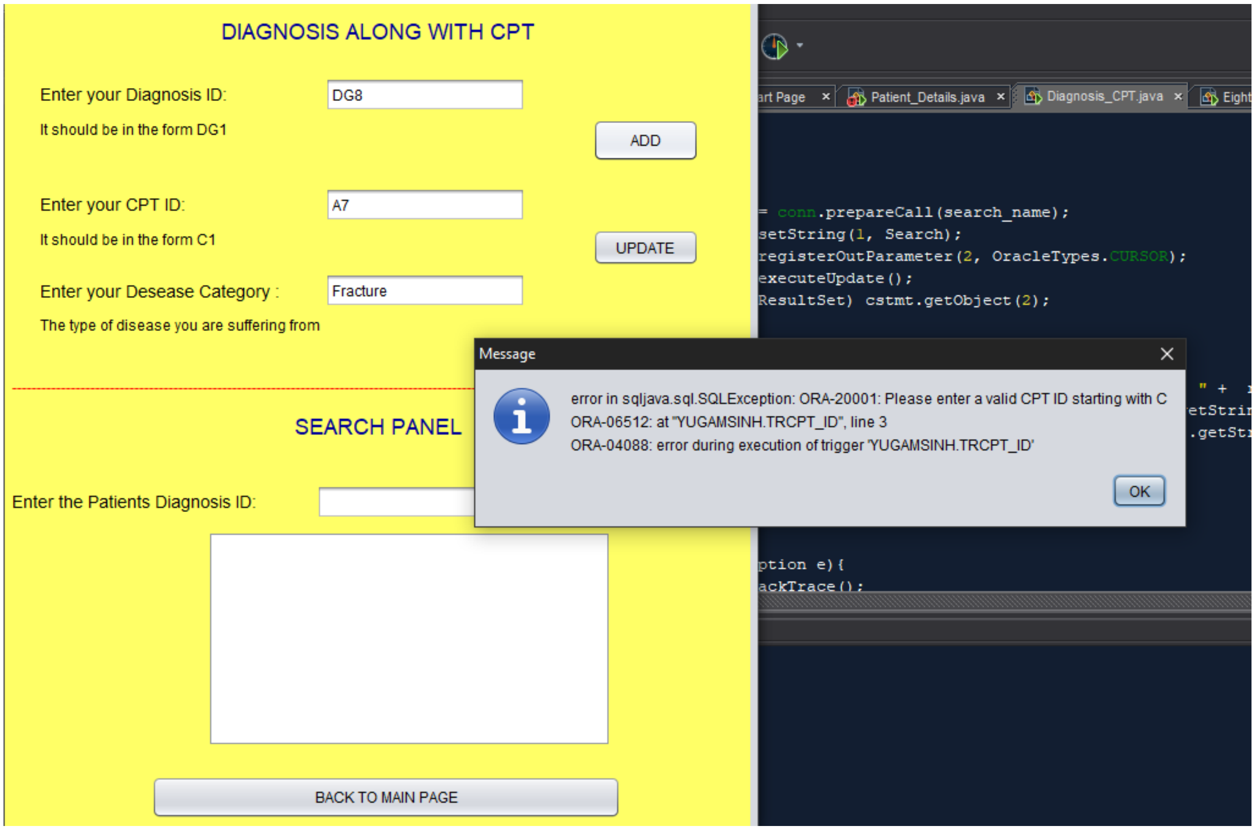
Enter the Patients Diagnosis ID:

```
= conn.prepareCall(search_name);
setString(1, Search);
registerOutParameter(2, OracleTypes.CURSOR);
executeUpdate();
ResultSet) cstmt.getObject(2);
```

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRCATEGORYD", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRCATEGORYD'
```

- **CPT Table**

→ create or replace trigger trCPT_ID before insert or update on CPT for each row
begin
if (substr(:new.CPT_ID,1,1) <> 'C') then
raise_application_error(-20001,'Please enter a valid CPT ID starting with C');
end if;
end;



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG8

It should be in the form DG1

Enter your CPT ID: A7

It should be in the form C1

Enter your Desease Category : Fracture

The type of disease you are suffering from

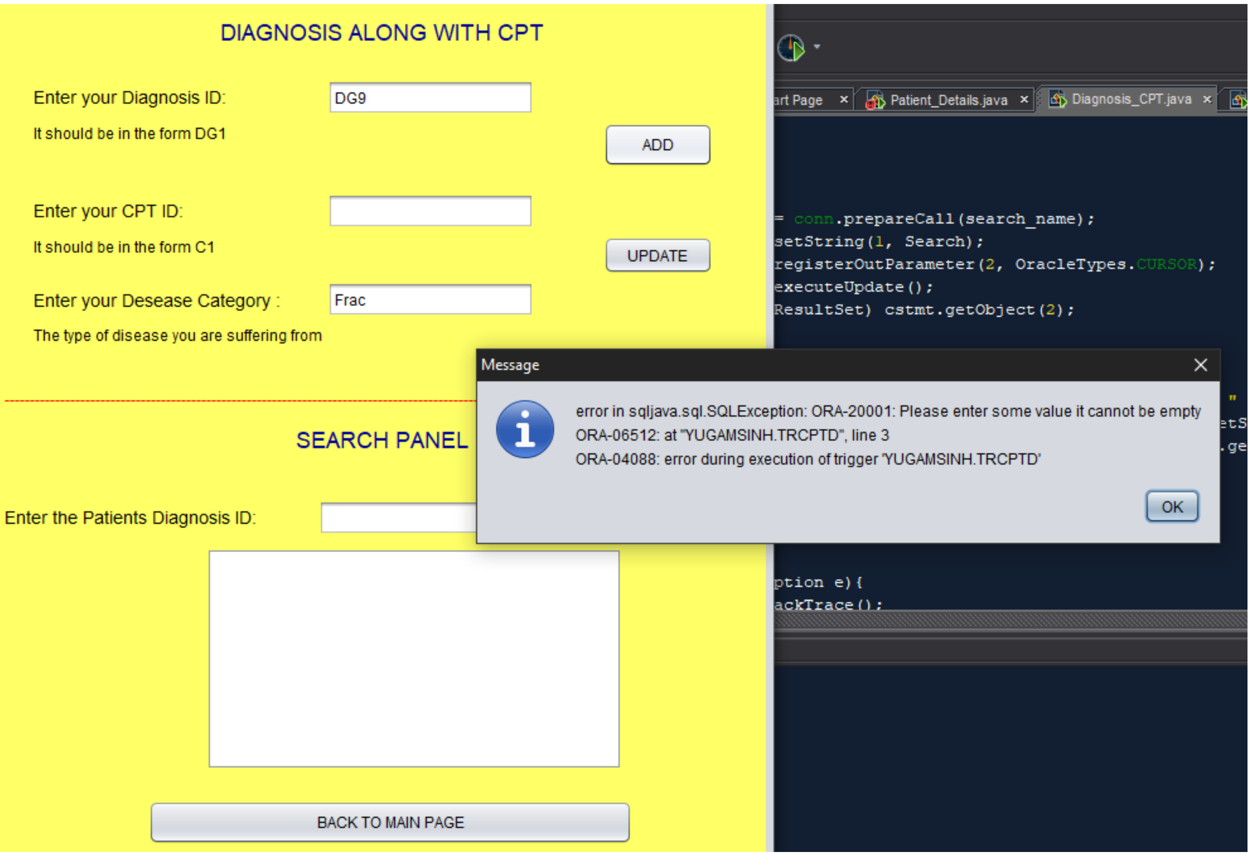
SEARCH PANEL

Enter the Patients Diagnosis ID:

```
error in sqljava.sql.SQLException: ORA-20001: Please enter a valid CPT ID starting with C
ORA-06512: at "YUGAMSINH.TRCPT_ID", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRCPT_ID'
```

```
= conn.prepareCall(search_name);
setString(1, Search);
registerOutParameter(2, OracleTypes.CURSOR);
executeUpdate();
ResultSet) cstmt.getObject(2);

                                            " + 
                                           etStri
                                           .getSt

ption e){
ackTrace();
```

```
→ create or replace trigger trCPTD before insert or update on CPT for each
row
begin
        if(:new.CPT_ID is null) then
              raise_application_error(-20001,'Please enter some value it cannot
be empty');
        end if;
end;
```



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG9

It should be in the form DG1

Enter your CPT ID:

It should be in the form C1

Enter your Desease Category : Frac

The type of disease you are suffering from

SEARCH PANEL

Enter the Patients Diagnosis ID:

Message

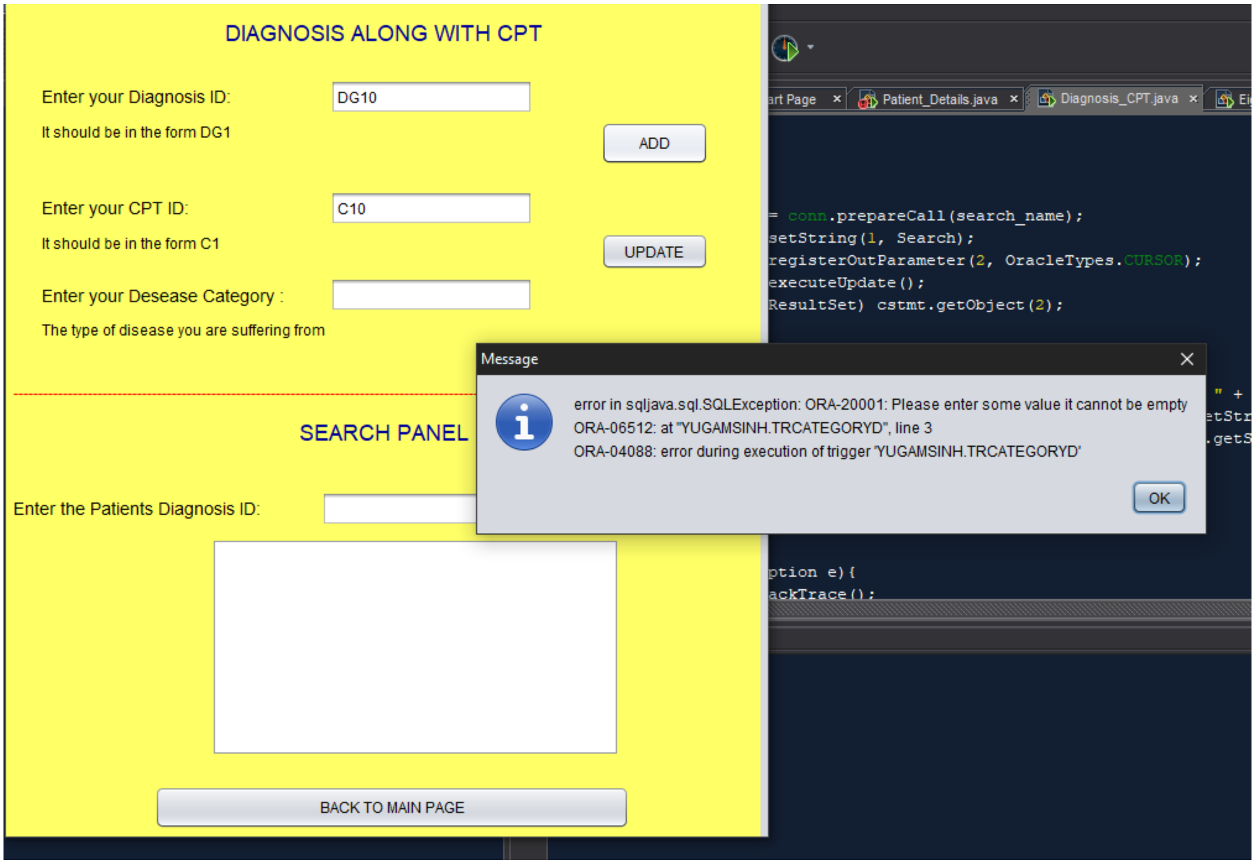
```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRCPTD", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRCPTD'
```

```
= conn.prepareCall(search_name);
setString(1, Search);
registerOutParameter(2, OracleTypes.CURSOR);
executeUpdate();
ResultSet) cstmt.getObject(2);
```

```
ption e){
ackTrace();
```

→ create or replace trigger trCategoryC before insert or update on CPT for each row

```
begin
    if(:new.Category is null) then
        raise_application_error(-20001,'Please enter some value it cannot be empty');
    end if;
end;
```



DIAGNOSIS ALONG WITH CPT

Enter your Diagnosis ID: DG10

It should be in the form DG1

Enter your CPT ID: C10

It should be in the form C1

Enter your Desease Category :

The type of disease you are suffering from

SEARCH PANEL

Enter the Patients Diagnosis ID:

```
= conn.prepareCall(search_name);
setString(1, Search);
registerOutParameter(2, OracleTypes.CURSOR);
executeUpdate();
ResultSet) cstmt.getObject(2);
```

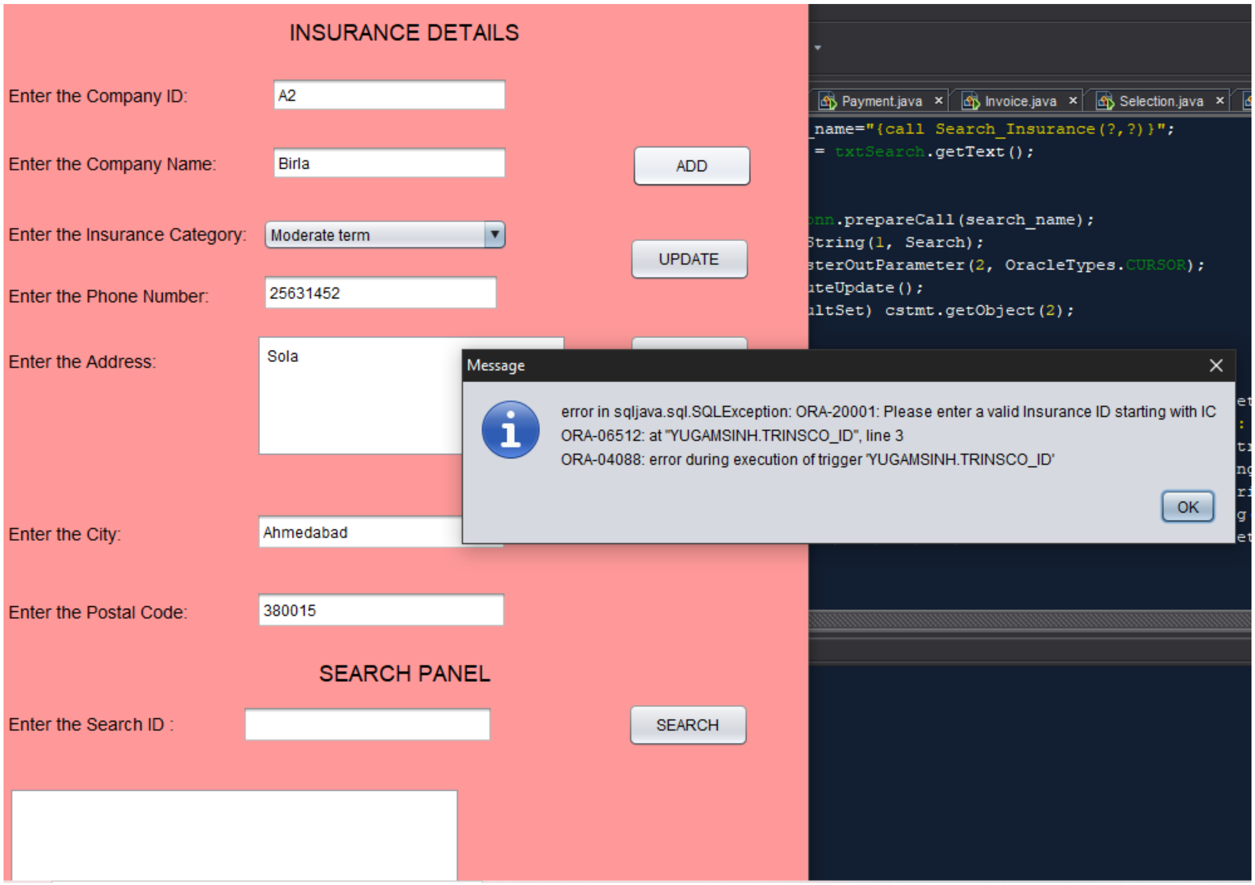
Message

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRCATEGORYD", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRCATEGORYD'
```

```
" +
etStr
.getS

ption e){
ackTrace();
```

- **Insurance Table**

→ create or replace trigger trInsCo_ID before insert or update on Insurance for each row
begin
if (substr(:new.InsCo_ID,1,2) <> 'IC') then
raise_application_error(-20001,'Please enter a valid Insurance ID starting with IC');
end if;
end;



INSURANCE DETAILS

Enter the Company ID: A2

Enter the Company Name: Birla

Enter the Insurance Category: Moderate term

Enter the Phone Number: 25631452

Enter the Address: Sola

Enter the City: Ahmedabad

Enter the Postal Code: 380015

SEARCH PANEL

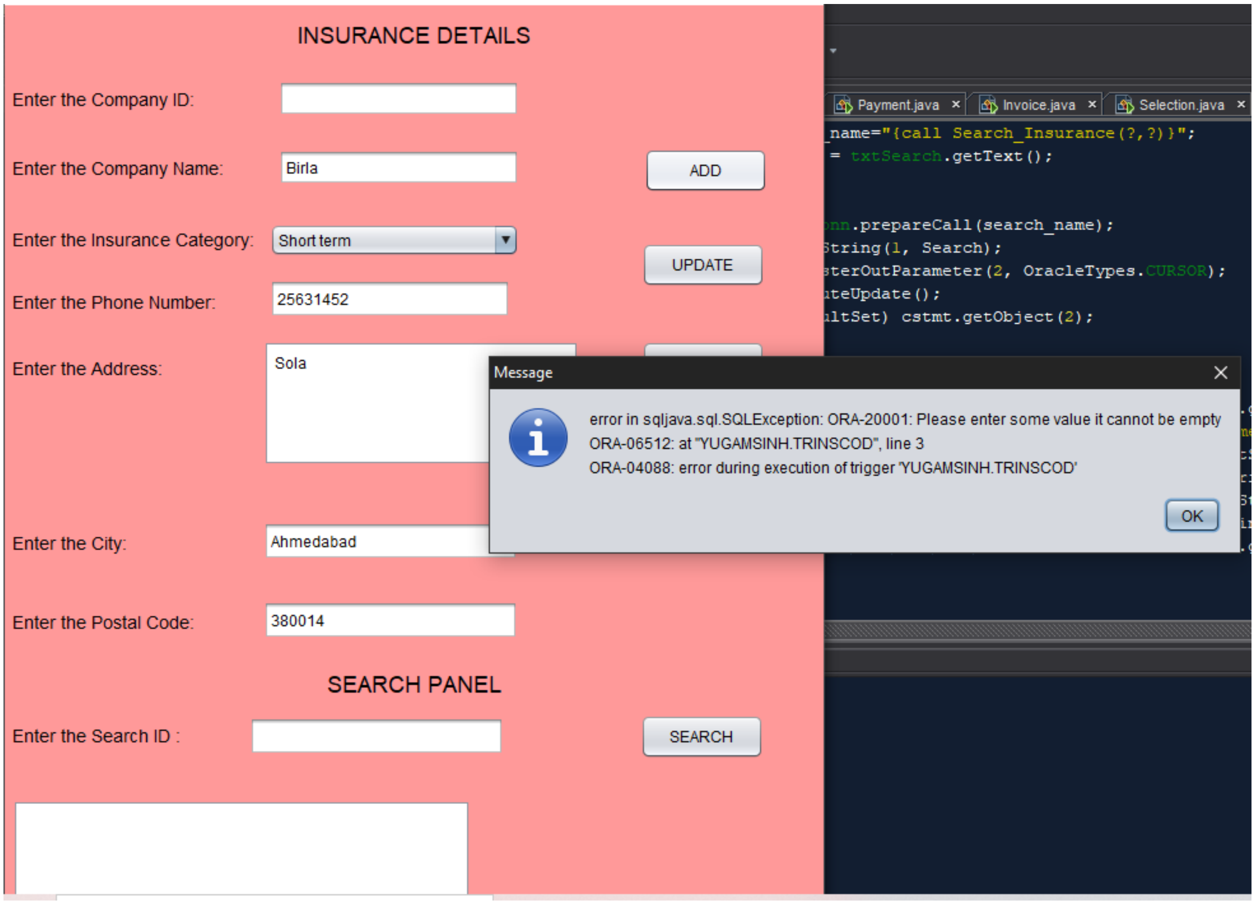
Enter the Search ID :

```
_name="{call Search_Insurance(?,?)}";
 = txtSearch.getText();


onn.prepareCall(search_name);
String(1, Search);
sterOutParameter(2, OracleTypes.CURSOR);
uteUpdate();
ultSet) cstmt.getObject(2);
```

```
error in sqljava.sql.SQLException: ORA-20001: Please enter a valid Insurance ID starting with IC
ORA-06512: at "YUGAMSINH.TRINSCO_ID", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRINSCO_ID'
```

→ create or replace trigger trInsCoD before insert or update on Insurance for each row

```
begin
        if(:new.InsCo_ID is null) then
            raise_application_error(-20001,'Please enter some value it cannot be empty');
        end if;
end;
```



INSURANCE DETAILS

Enter the Company ID:

Enter the Company Name: Birla

Enter the Insurance Category: Short term

Enter the Phone Number: 25631452

Enter the Address: Sola

Enter the City: Ahmedabad

Enter the Postal Code: 380014

SEARCH PANEL

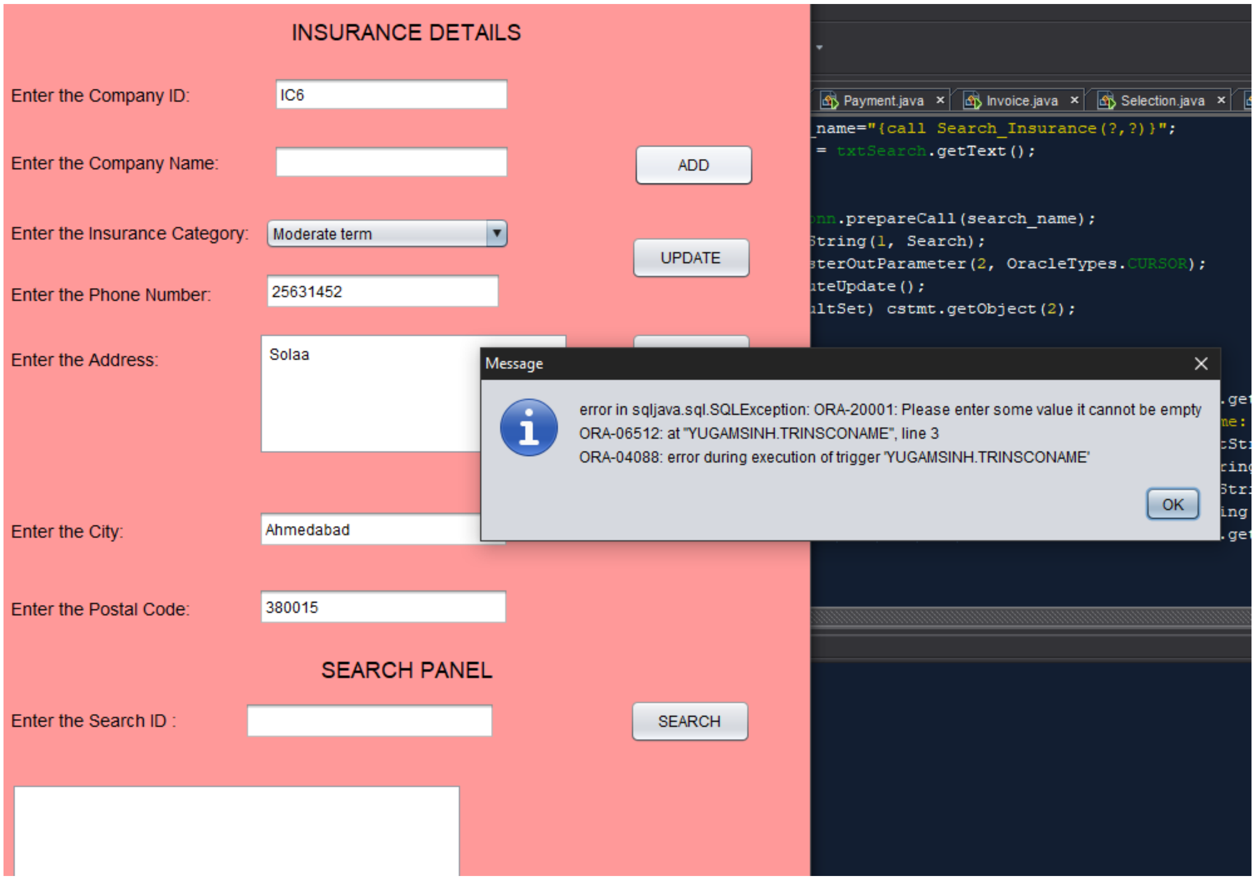
Enter the Search ID :

```
_name="{call Search_Insurance(?,?)}";
= txtSearch.getText();

onn.prepareCall(search_name);
String(1, Search);
sterOutParameter(2, OracleTypes.CURSOR);
uteUpdate();
ultSet) cstmt.getObject(2);
```

```
error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRINSCOD", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRINSCOD'
```

→ create or replace trigger trInsCoName before insert or update on Insurance for each row

```
begin
    if(:new.InsCo_Name is null) then
        raise_application_error(-20001,'Please enter some value it cannot be empty');
    end if;
end;
```



INSURANCE DETAILS

Enter the Company ID: IC6

Enter the Company Name:

Enter the Insurance Category: Moderate term

Enter the Phone Number: 25631452

Enter the Address: Solaa

Enter the City: Ahmedabad

Enter the Postal Code: 380015

SEARCH PANEL

Enter the Search ID :

error in sqljava.sql.SQLException: ORA-20001: Please enter some value it cannot be empty
ORA-06512: at "YUGAMSINH.TRINSCONAME", line 3
ORA-04088: error during execution of trigger 'YUGAMSINH.TRINSCONAME'

```
_name="{call Search_Insurance(?,?)}";
 = txtSearch.getText();

onn.prepareCall(search_name);
String(1, Search);
sterOutParameter(2, OracleTypes.CURSOR);
uteUpdate();
ultSet) cstmt.getObject(2);
```

```
→ create or replace trigger trCategory before insert or update on Insurance
for each row
begin
        if(:new.Category is null) then
               raise_application_error(-20001,'Please enter some value it cannot
be empty');
        end if;
end;
```

- **Invoice Table and Invoice_P**

```
→ create or replace trigger trInNuM before insert or update on Invoice for
each row
begin
if (substr(:new.Invoice_Num,1,1) <> 'I') then
```

```
raise_application_error(-20001,'Please enter a valid Invoice Num starting
with I');
end if;
end;
```

```
→ create or replace trigger trINum before insert or update on Invoice for
each row
begin
    if(:new.Invoice_Num is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

```
→ create or replace trigger trIN before insert or update on Invoice_P for
each row
begin
if (substr(:new.Invoice_Num,1,1) <> 'I') then
raise_application_error(-20001,'Please enter a valid Invoice Num starting
with I');
end if;
end;
```

```
→ create or replace trigger trPIDInvoice before insert or update on
Invoice_P for each row
begin
if (substr(:new.P_ID,1,1) <> 'P') then
raise_application_error(-20001,'Please enter a valid Patient ID starting with
P');
end if;
end;
```

```
→ create or replace trigger trCPTInvoice before insert or update on
Invoice_P for each row
begin
if (substr(:new.CPT_ID,1,1) <> 'C') then
raise_application_error(-20001,'Please enter a valid CPT ID starting with
C');
end if;
end;

→ create or replace trigger trDiagInvoice before insert or update on
Invoice_P for each row
begin
if (substr(:new.Diagnosis_ID,1,2) <> 'DG') then
raise_application_error(-20001,'Please enter a valid Diagnosis ID starting
with DG');
end if;
end;

→ create or replace trigger trreInvoice before insert or update on Invoice_P
for each row
begin
if (substr(:new.Prescription_ID,1,2) <> 'PR') then
raise_application_error(-20001,'Please enter a valid Prescription ID starting
with PR');
end if;
end;

→ create or replace trigger trInNULL before insert or update on Invoice_P
for each row
begin
    if(:new.Invoice_Num is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
```

```
    end if;
end;

→ create or replace trigger trPatNULL before insert or update on Invoice_P
for each row
begin
    if(:new.P_ID is null) then
       raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;

→ create or replace trigger trCPTNULL before insert or update on
Invoice_P for each row
begin
    if(:new.CPT_ID is null) then
       raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;

→ create or replace trigger trDiagNULL before insert or update on
Invoice_P for each row
begin
    if(:new.Diagnosis_ID is null) then
       raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;

→ create or replace trigger trIPreNULL before insert or update on
Invoice_P for each row
begin
```

```
    if(:new.Prescription_ID is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

- **Medicine Table**

```
→ create or replace trigger trMID before insert or update on Medicine for
each row
begin
if (substr(:new.M_ID,1,2) <> 'ME') then
raise_application_error(-20001,'Please enter a valid Medicine id starting
with ME');
end if;
end;
```

```
→ create or replace trigger trMedID before insert or update on Medicine for
each row
begin
    if(:new.M_ID is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
→ create or replace trigger trMedName before insert or update on Medicine
for each row
begin
    if(:new.M_Name is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

- **For Prescription**

```
→ create or replace trigger trPre before insert or update on Prescription for
each row
begin
if (substr(:new.Prescription_ID,1,2) <> 'PR') then
raise_application_error(-20001,'Please enter a valid prescription id starting
with PR');
end if;
end;
```

```
→ create or replace trigger trPreID before insert or update on Prescription
for each row
begin
    if(:new.Prescription_ID is null) then
       raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

```
→ create or replace trigger trMedicineQ before insert or update on
Prescription for each row
begin
    if(:new.Medicine_Q<1) then
       raise_application_error(-20001,'The Medicine Quantity is atleast 1');
    end if;
end;
```

- **For Medicine_P**

```
→ create or replace trigger trPreIDP before insert or update on Medicine_P
for each row
begin
```

```
if (substr(:new.Prescription_ID,1,2) <> 'PR') then
raise_application_error(-20001,'Please enter a valid Prescription id starting
with PR');
end if;
end;
```

```
→ create or replace trigger trMIDP before insert or update on Medicine_P
for each row
begin
if (substr(:new.M_ID,1,2) <> 'ME') then
raise_application_error(-20001,'Please enter a valid Medicine id starting
with ME');
end if;
end;
```

```
→ create or replace trigger trMedIDP before insert or update on
Medicine_P for each row
begin
    if(:new.M_ID is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

```
→ create or replace trigger trPreIDPnull before insert or update on
Medicine_P for each row
begin
    if(:new.Prescription_ID is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

```
→ create or replace trigger trMedNameP before insert or update on
Medicine_P for each row
begin
      if(:new.M_Name is null) then
          raise_application_error(-20001,'Please enter some value it cannot
be empty');
      end if;
end;
```

```
→ create or replace trigger trMedNamePrice before insert or update on
Medicine_P for each row
begin
      if(:new.Price=0) then
          raise_application_error(-20001,'Please enter the value properly.It
cannot be Zero');
      end if;
end;
```

- **Patient Table**

```
→ create or replace trigger trPa before insert or update on Patient
for each row
begin
if(length(:new.P_id)>=5) then
    raise_application_error(-20001,'Please enter within range of 4 chars
preferably');
end if;
end;
```

```
→ create or replace trigger trPID before insert or update on Patient for
each row
begin
if (substr(:new.P_id,1,1) <> 'P') then
raise_application_error(-20001,'Please enter a valid doctor id starting with
P');
end if;
end;

→ create or replace trigger trPIDN before insert or update on Patient for
each row
begin
    if(:new.P_id is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;

→ create or replace trigger trPName before insert or update on Patient for
each row
begin
    if(:new.P_name is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;

→ create or replace trigger trPG before insert or update on Patient
for each row
begin
if inserting then
   if(:new.Gender<>'Female' and :new.Gender<>'Male' and
:new.Gender<>'female' and :new.Gender<>'male') then
```

```
   raise_application_error(-20001,'Please enter from either male or female');
end if;
end if;
if updating then
      if(:new.Gender<>'Female' and :new.Gender<>'Male' and
:new.Gender<>'female' and :new.Gender<>'male') then
   raise_application_error(-20001,'Please enter from either male or female');
end if;
end if;
End;
```

- **Patient_Doc**

```
→ create or replace trigger trPDIDD before insert or update on Patient_Doc
for each row
begin
if(length(:new.D_id)>=5) then
     raise_application_error(-20001,'Please enter within range of 4 chars
preferably');
end if;
end;
```

```
→ create or replace trigger trPDIDP before insert or update on Patient_Doc
for each row
begin
if(length(:new.P_id)>=5) then
     raise_application_error(-20001,'Please enter within range of 4 chars
preferably');
end if;
end;
```

```
→ create or replace trigger trPStart before insert or update on Patient_Doc
for each row
```

```
begin
if (substr(:new.P_id,1,1) <> 'P') then
raise_application_error(-20001,'Please enter a valid patient id starting with
P');
end if;
end;
```

```
→ create or replace trigger trDStart before insert or update on Patient_Doc
for each row
begin
if (substr(:new.D_id,1,1) <> 'D') then
raise_application_error(-20001,'Please enter a valid Doctor id starting with
D');
end if;
end;
```

```
→ create or replace trigger trDoctorIDNull before insert or update on
Patient_Doc for each row
begin
    if(:new.D_id is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

```
→ create or replace trigger trPatientIDNull before insert or update on
Patient_Doc for each row
begin
    if(:new.P_id is null) then
        raise_application_error(-20001,'Please enter some value it cannot
be empty');
    end if;
end;
```

```
→ create or replace trigger trPatName before insert or update on
Patient_Doc for each row
begin
      if(:new.P_name is null) then
          raise_application_error(-20001,'Please enter some value it cannot
be empty');
      end if;
end;
```

```
→ create or replace trigger trPGender before insert or update on
Patient_Doc
for each row
begin
if inserting then
     if(:new.Gender<>'Female' and :new.Gender<>'Male' and
:new.Gender<>'female' and :new.Gender<>'male') then
            raise_application_error(-20001,'Please enter from either male or
female');
end if;
end if;
if updating then
     if(:new.Gender<>'Female' and :new.Gender<>'Male' and
:new.Gender<>'female' and :new.Gender<>'male') then
            raise_application_error(-20001,'Please enter from either male or
female');
end if;
end if;
end;
```

- **Paycheck Table**

```
→ create or replace trigger trChk_Num before insert or update on
Paycheck for each row
```

```
begin
if (substr(:new.Chk_Num,1,1) <> 'A') then
raise_application_error(-20001,'Please enter a valid patient id starting with
A');
end if;
end;
```

```
→ create or replace trigger trCHKNull before insert or update on Paycheck
for each row
begin
      if(:new.Chk_Num is null) then
          raise_application_error(-20001,'Please enter some value it cannot
be empty');
      end if;
end;
```

```
→ create or replace trigger trSalary before insert or update on Paycheck for
each row
begin
      if(:new.Salary<3000) then
          raise_application_error(-20001,'Please enter some value greater
than 3000');
      end if;
end;
```

```
→ create or replace trigger trBonus before insert or update on Paycheck for
each row
begin
      if(:new.Bonus<1000) then
          raise_application_error(-20001,'Please enter some value greater
than 1000');
      end if;
end;
```

- **Paycheck_Doc Table**

```
→ create or replace trigger trChk_NumD before insert or update on Paycheck_Doc for each row
begin
if (substr(:new.Chk_Num,1,1) <> 'A') then
raise_application_error(-20001,'Please enter a valid patient id starting with A');
end if;
end;
```

```
→ create or replace trigger trCHKNullD before insert or update on Paycheck_Doc for each row
begin
      if(:new.Chk_Num is null) then
          raise_application_error(-20001,'Please enter some value it cannot be empty');
      end if;
end;
```

```
→ create or replace trigger trChk_NumDID before insert or update on Paycheck_Doc for each row
begin
if (substr(:new.D_Id,1,1) <> 'D') then
raise_application_error(-20001,'Please enter a valid patient id starting with D');
end if;
end;
```

```
→ create or replace trigger trCHKNullDoc before insert or update on Paycheck_Doc for each row
begin
      if(:new.D_Id is null) then
```

```
        raise_application_error(-20001,'Please enter some value it cannot be empty');
    end if;
end;
```

```
→ create or replace trigger trSalaryD before insert or update on Paycheck_Doc for each row
begin
    if(:new.Salary<3000) then
        raise_application_error(-20001,'Please enter some value greater than 3000');
    end if;
end;
```

```
→ create or replace trigger trBonusD before insert or update on Paycheck_Doc for each row
begin
    if(:new.Bonus<1000) then
        raise_application_error(-20001,'Please enter some value greater than 1000');
    end if;
end;
```

- **Payment Table**

```
→ create or replace trigger trPay before insert or update on Payment for each row
begin
if (substr(:new.P_Num,1,3) <> 'PAY') then
raise_application_error(-20001,'Please enter a valid Payment Num starting with PAY');
end if;
end;
```

```
→ create or replace trigger trPaytype before insert or update on Payment
for each row
begin
if inserting then
      if(:new.Pay_Method<>'credit card' and :new.Pay_Method<>'cash' and
:new.Pay_Method<>'cheque' and :new.Pay_Method<>'Credit Card'
 and :new.Pay_Method<>'Cash' and :new.Pay_Method<>'credit Cash') then
             raise_application_error(-20001,'Please enter from either credit
card or Cheque or Cash');
end if;
end if;
if updating then
     if(:new.Pay_Method<>'credit card' and :new.Pay_Method<>'cash' and
:new.Pay_Method<>'cheque' and :new.Pay_Method<>'Credit Card'
 and :new.Pay_Method<>'Cash' and :new.Pay_Method<>'credit Cash') then
             raise_application_error(-20001,'Please enter from either credit
card or Cheque or Cash');
end if;
end if;
end;
```

```
→ create or replace trigger trPStatus before insert or update on Payment
for each row
begin
if inserting then
      if(:new.Pay_Status<>'Paid' and :new.Pay_Status<>'Pending' and
:new.Pay_Status<>'paid' and :new.Pay_Status<>'pending') then
             raise_application_error(-20001,'Please enter from either Paid or
Pending');
end if;
end if;
if updating then
```

```
    if(:new.Pay_Status<>'Paid' and :new.Pay_Status<>'Pending' and :new.Pay_Status<>'paid' and :new.Pay_Status<>'pending') then
            raise_application_error(-20001,'Please enter from either Paid or Pending');
end if;
end if;
end;
```

```
→ create or replace trigger trPaN before insert or update on Payment for each row
begin
      if(:new.P_Num is null) then
          raise_application_error(-20001,'Please enter some value it cannot be empty');
      end if;
end;
```

- **Payment_P Table:**

```
→ create or replace trigger trPayP before insert or update on Payment_P for each row
begin
if (substr(:new.P_Num,1,3) <> 'PAY') then
raise_application_error(-20001,'Please enter a valid Payment Num starting with PAY');
end if;
end;
```

```
→ create or replace trigger trPayInvoice before insert or update on Payment_P for each row
begin
if (substr(:new.Invoice,1,1) <> 'I') then
```

```
raise_application_error(-20001,'Please enter a valid Invoive Num starting
with I');
end if;
end;

→ create or replace trigger trPaNum before insert or update on Payment_P
for each row
begin
      if(:new.P_Num is null) then
          raise_application_error(-20001,'Please enter some value it cannot
be empty');
      end if;
end;

→ create or replace trigger trInvoiceNULL before insert or update on
Payment_P for each row
begin
      if(:new.Invoice is null) then
          raise_application_error(-20001,'Please enter some value it cannot
be empty');
      end if;
end;

→ create or replace trigger trPaymettype before insert or update on
Payment_P
for each row
begin
if inserting then
     if(:new.Pay_Method<>'credit card' and :new.Pay_Method<>'cash' and
:new.Pay_Method<>'cheque' and :new.Pay_Method<>'Credit Card'
 and :new.Pay_Method<>'Cash' and :new.Pay_Method<>'credit Cash') then
             raise_application_error(-20001,'Please enter from either credit
card or Cheque or Cash');
```

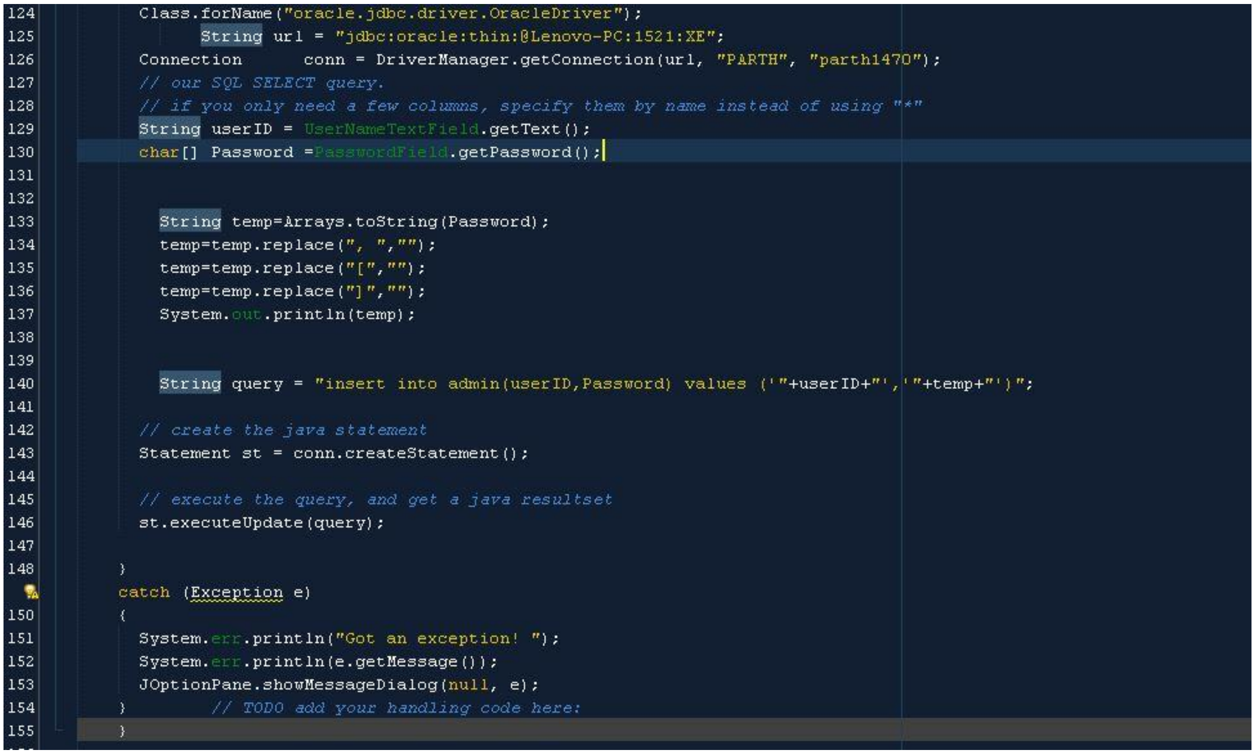
```
end if;
end if;
if updating then
     if(:new.Pay_Method<>'credit card' and :new.Pay_Method<>'cash' and
:new.Pay_Method<>'cheque' and :new.Pay_Method<>'Credit Card'
 and :new.Pay_Method<>'Cash' and :new.Pay_Method<>'credit Cash') then
               raise_application_error(-20001,'Please enter from either credit
card or Cheque or Cash');
end if;
end if;
end;

→ create or replace trigger trPayStatus before insert or update on
Payment_P
for each row
begin
if inserting then
      if(:new.Pay_Status<>'Paid' and :new.Pay_Status<>'Pending' and
:new.Pay_Status<>'paid' and :new.Pay_Status<>'pending') then
               raise_application_error(-20001,'Please enter from either Paid or
Pending');
end if;
end if;
if updating then
     if(:new.Pay_Status<>'Paid' and :new.Pay_Status<>'Pending' and
:new.Pay_Status<>'paid' and :new.Pay_Status<>'pending') then
               raise_application_error(-20001,'Please enter from either Paid or
Pending');
end if;
end if;
End;
```

- Trigger for checking password Strength

```
create or replace trigger checkPassword_admin
before insert or update on admin
for each row
declare
temp int:=TO_NUMBER(LENGTH(:new.Password));
begin
    if inserting then
       if ( temp < 7 ) then
                              raise_application_error(-20001,'Password length should
be greater than or equal to 8');
       end if;
       if (LENGTH(TRIM(TRANSLATE(:new.Password, ' +-.0123456789',' ')))
IS NULL ) then
                              raise_application_error(-20001,'Password should
contain character');
       end if;

    if ((LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.ABCDEFGHIJJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz','
'))) IS NULL)  OR  (LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.abcdefghijklmnopqrstuvwxyz',' '))) IS NULL)) then
                              raise_application_error(-20001,'Password should
contain a number');
       end if;
if ((LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.ABCDEFGHIJJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!
@#$%^&*()-_+=?/><.;:0123456789',' '))) IS NULL)  OR
(LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.abcdefghijklmnopqrstuvwxyz',' '))) IS NULL)) then
```

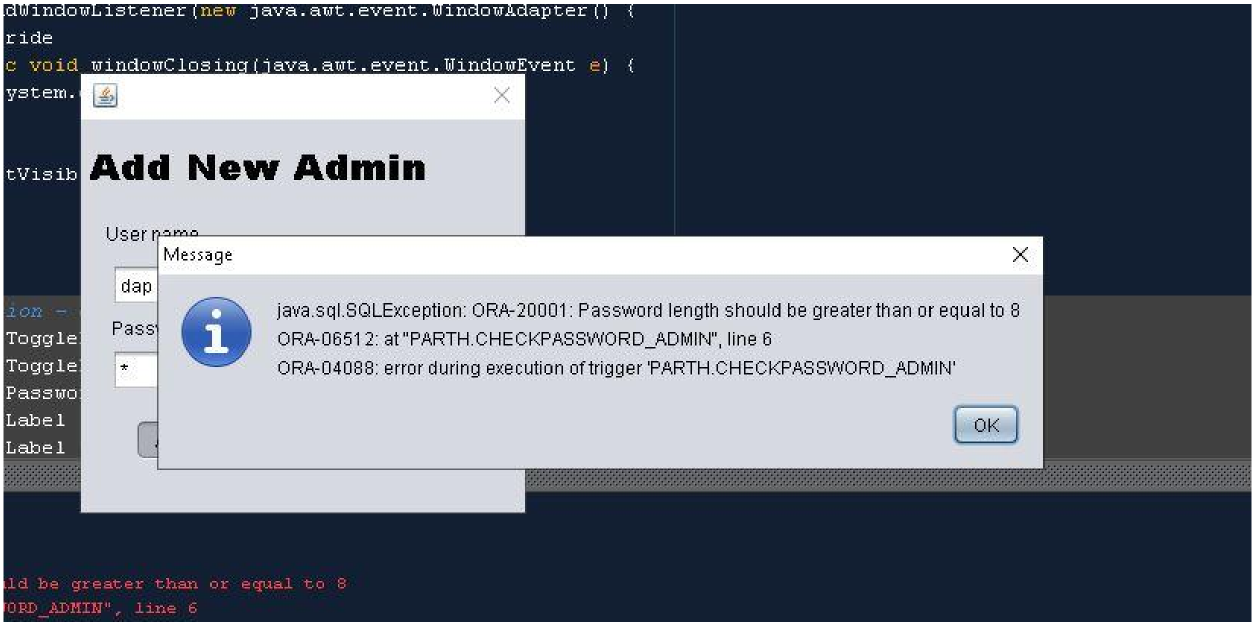
```
                        raise_application_error(-20001,'Password should
contain a special character');
      end if;

    end if;
    if updating then
      if (temp < 7 ) then
                        raise_application_error(-20000,'Password length should
be greater than or equal to 8');
      end if;
      if (LENGTH(TRIM(TRANSLATE(:new.Password, ' +-.0123456789',' ')))
IS NULL ) then
                        raise_application_error(-20001,'Password should
contain character');
      end if;
    end if;
   if ((LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.ABCDEFGHIJJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz','
'))) IS NULL)  OR  (LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.abcdefghijklmnopqrstuvwxyz',' '))) IS NULL)) then
                        raise_application_error(-20001,'Password should
contain a number');
      end if;
if ((LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.ABCDEFGHIJJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!
@#$%^&*()-_+=?/><.;:0123456789',' '))) IS NULL)  OR
(LENGTH(TRIM(TRANSLATE(:new.Password,
'+-.abcdefghijklmnopqrstuvwxyz',' '))) IS NULL)) then
                        raise_application_error(-20001,'Password should
contain a special character');
      end if;

end;
```

→ For java :-



```
        Class.forName("oracle.jdbc.driver.OracleDriver");
              String url = "jdbc:oracle:thin:@Lenovo-PC:1521:XE";
        Connection      conn = DriverManager.getConnection(url, "PARTH", "parth1470");
        // our SQL SELECT query.
        // if you only need a few columns, specify them by name instead of using "*"
        String userID = UserNameTextField.getText();
        char[] Password =PasswordField.getPassword();


          String temp=Arrays.toString(Password);
          temp=temp.replace(", ","");
          temp=temp.replace("[","");
          temp=temp.replace("]","");
          System.out.println(temp);


          String query = "insert into admin(userID,Password) values ('"+userID+"','"+temp+"')";

        // create the java statement
        Statement st = conn.createStatement();

        // execute the query, and get a java resultset
        st.executeUpdate(query);

      }
      catch (Exception e)
      {
        System.err.println("Got an exception! ");
        System.err.println(e.getMessage());
        JOptionPane.showMessageDialog(null, e);
      }         // TODO add your handling code here:
      }
```

→ Output:-



```
dWindowListener(new java.awt.event.WindowAdapter() {
ride
c void windowClosing(java.awt.event.WindowEvent e) {
ystem.
tVisib
```

Add New Admin

User name

dap

Pass

*

Message

java.sql.SQLException: ORA-20001: Password length should be greater than or equal to 8
ORA-06512: at "PARTH.CHECKPASSWORD_ADMIN", line 6
ORA-04088: error during execution of trigger 'PARTH.CHECKPASSWORD_ADMIN'

```
ion -
Toggle
Toggle
Passwo
Label
Label

ld be greater than or equal to 8
ORD_ADMIN", line 6
```