JAVA AWT BASED- REGISTRATION FORM- SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

K.GRETA <1602-18-737-069>

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31,2019

BONAFIDE CERTIFICATE

This to Certify that the project report titled **“FEEDBACK RETRIEVAL SYSTEM OF HOTELS”** project work of **Ms K.GRETA** bearing **Roll.no:1602-18-737-069** who carried out this project under my supervision in the IV semester for the academic year 2019-2020.

Signature	**Signature**
External Examiner	**Internal Examiner**

AIM:

To create a **Java GUI based Feedback form for Hotels** which takes the details of all the hotels, feedbacks and persons who are giving feedback from the user. These values are to be updated in the database using **JDBC connectivity .**

ABSTRACT:

Feedback Retrieval System for Hotels

Feedback retrieval system for hotels basically helps all the hotels to collect feedback from people. This database stores all the details of users, hotels, feedbacks, services provided by the hotels.

The User enters his details first and checks the information of the hotel and then gives the feedback in form of suggestions and rating. The user will also post the complaints which will be further referred and improved by the management of the hotels. Users can also view the feedbacks given for other hotels by choosing the hotels. Users can also view the details of the hotels before the choose to stay in the hotel.

INTRODUCTION:

REQUIREMENT SPECIFICATION:

This database consists of 4 Entites namely Persons, Hotels, Services, Feedback and 3 relationships i.e.,

Person: Entity set contains the information of the person who gives feedback.

Attributes:

- Pid NUMBER(primary).
- Name VARCHAR
- Contact NUMBER(10)
- Mail VARCHAR2(20).

Hotels: It is an Entity set which contains the information of all the hotels.

Attributes:

- Name VARCHAR2(10)
- Hid NUMBER(2) (primary)
- Contact NUMBER(10)
- Address VARCHAR2(30).

Services: It is an entity which provides the information of services provided by the Hotel.

- Hid NUMBER(2)

- Sid NUMBER(5),
- Rooms VARCHAR2(100),
- Food VARCHAR2(100)
- Gamezone VARCHAR2(100),
- Ambience VARCHAR2(100).

Feedback: It helps in collecting the feedback from persons.

Attributes:

- Fid NUMBER(3) (primary)
- Complaints VARCHAR2(100)
- Suggestions VARCHAR2(100)
- Rating VARCHAR2(100)

Cardinalities:

1.Checks:Person checks the information of hotels.

2.Provides:Services provided by the hotel.

3.Gives:Feedback given by the user for a particular hotel.

The cardinalities are

a)Person(Many)-hotels(Many)

b) Person(one)-feedback(many).

This helps the organisation to improvise their hotels.

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 7, SQL*Plus.

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

JAVA SWING:

Swing is a GUI widget toolkit for Java. It is part of Oracle 's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT).

SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Java-SQL Connectivity using JDBC:

Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a

Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

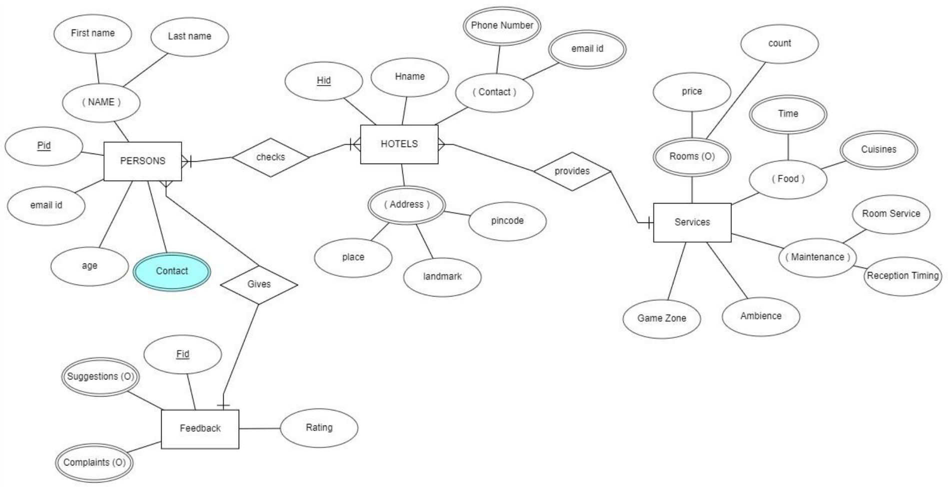
The connection to the database can be performed using Java programming (JDBC API) as:

```
private void connToDb(){
             try {
             Class.forName("oracle.jdbc.driver.OracleDriver");
connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:xe","greta","GRETA091
0");
             statement = connection.createStatement();

           } catch (SQLException connectException) {
             System.out.println(connectException.getMessage());
             System.out.println(connectException.getSQLState());
             System.out.println(connectException.getErrorCode());
             System.exit(1);
           }
           catch (Exception e)
          {
               System.err.println("Unable to find and load driver");
               System.exit(1);
          }
}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

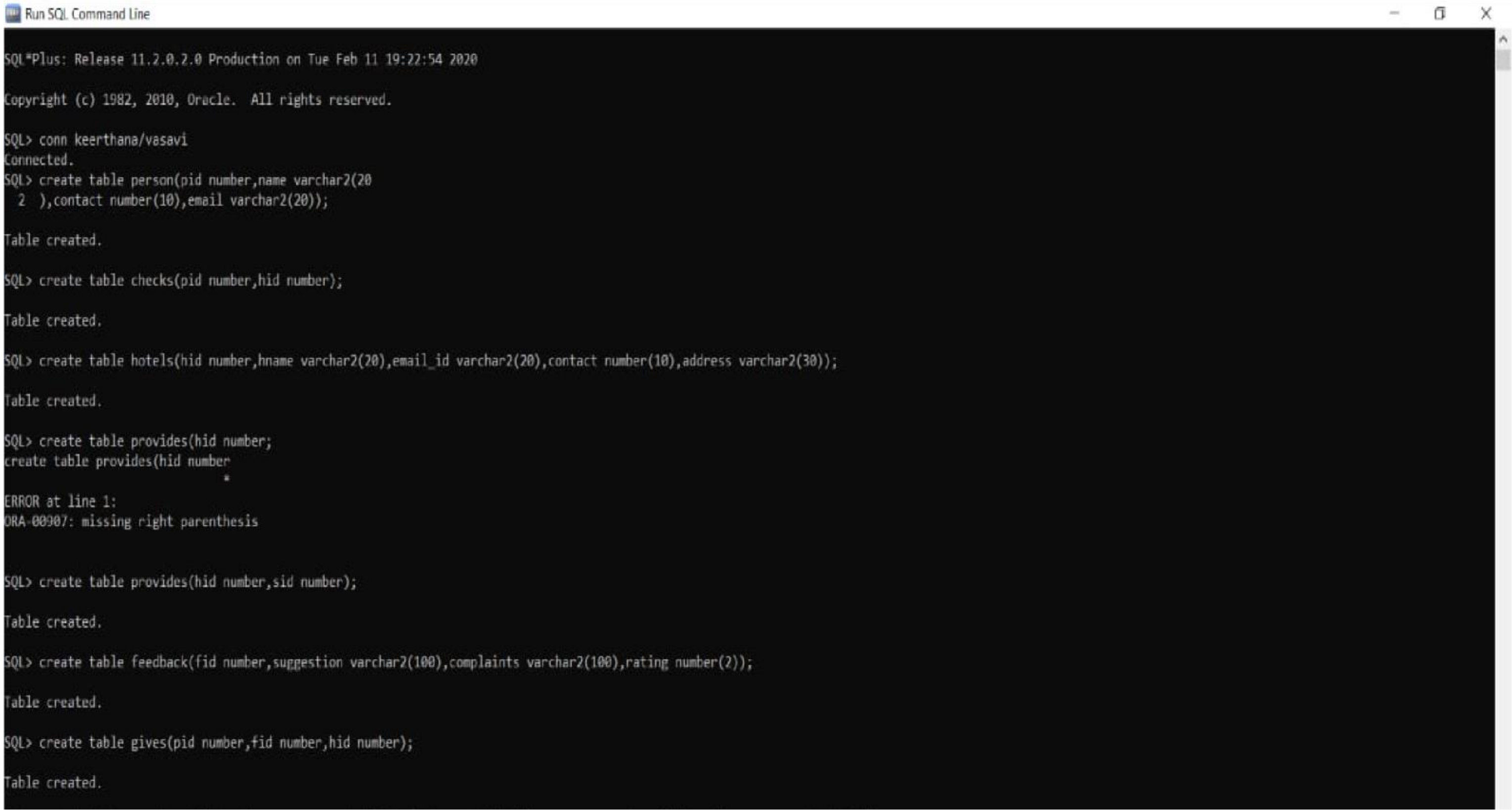
DESIGN:

1.ER DIAGRAM:



Tables

Created in SQL for above mentioned purpose is as:



```
SQL*Plus: Release 11.2.0.2.0 Production on Tue Feb 11 19:22:54 2020

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn keerthana/vasavi
Connected.
SQL> create table person(pid number,name varchar2(20
  2  ),contact number(10),email varchar2(20));

Table created.

SQL> create table checks(pid number,hid number);

Table created.

SQL> create table hotels(hid number,hname varchar2(20),email_id varchar2(20),contact number(10),address varchar2(30));

Table created.

SQL> create table provides(hid number;
create table provides(hid number
                               *
ERROR at line 1:
ORA-00907: missing right parenthesis


SQL> create table provides(hid number,sid number);

Table created.

SQL> create table feedback(fid number,suggestion varchar2(100),complaints varchar2(100),rating number(2));

Table created.

SQL> create table gives(pid number,fid number,hid number);

Table created.
```

Program

IMPLEMENTATION:

Main GUI:

```
import java.awt.EventQueue;

import java.awt.TextArea;

import javax.swing.JFrame;

import java.awt.CardLayout;

import java.awt.Color;

import java.awt.BorderLayout;

import javax.swing.SpringLayout;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JFormattedTextField;

import javax.swing.JToggleButton;

import java.awt.event.ActionListener;

 import java.sql.Connection;

 import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;
import java.awt.event.ActionEvent;
import java.awt.List; import
java.awt.event.ItemListener; import
java.awt.event.ItemEvent;


public class PGUI1 {
        TextArea ftf = new TextArea();
        Connection connection;
Statement statement;

private JFrame frame;
private JTextField tfname;
private JTextField tfid;
private JTextField tfage;
private JTextField tfcont;
private JTextField tfemail;     List
list = new List(10);     ResultSet
rs;      public PGUI1()

        {       try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connection();
initialize();
        }
        private void connection()
                try
                {
                  connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","greta","GRETA0910");
                  statement = connection.createStatement();
                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
   }
        public static void main(String[] args) {
```

```
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					PGUI1 window = new PGUI1();
					window.getFrame().setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}}});}
	private void loadList()
	{
		try
		{
		  rs = statement.executeQuery("SELECT PID FROM PERSONS");
	  while (rs.next())
		  {
			list.add(rs.getString("PID"));
		  }
		}
		catch (SQLException e)
		{
		  ftf.append(e.getMessage());
		}}
	private void initialize() {
```

```
                setFrame(new JFrame());
                getFrame().getContentPane().setBackground(Color.PINK);
SpringLayout springLayout = new SpringLayout();
getFrame().getContentPane().setLayout(springLayout);

                JLabel lblNewLabel = new JLabel("Person Name:");
getFrame().getContentPane().add(lblNewLabel);

                        JLabel lblNewLabel_1 = new JLabel("Person Id:");
                         getFrame().getContentPane().add(lblNewLabel_1);

                JLabel lblNewLabel_2 = new JLabel("Age:");
getFrame().getContentPane().add(lblNewLabel_2);
JLabel lblNewLabel_3 = new JLabel("Contact:");
getFrame().getContentPane().add(lblNewLabel_3);
JLabel lblEmailId = new JLabel("Email Id:");
getFrame().getContentPane().add(lblEmailId);

                tfname = new JTextField();
                getFrame().getContentPane().add(tfname);
                tfname.setColumns(10);
```

```
        tfid = new JTextField();
getFrame().getContentPane().add(tfid);
        tfid.setColumns(10);        tfage =
new JTextField();
getFrame().getContentPane().add(tfage);
        tfage.setColumns(10);


        tfcont = new JTextField();
tfcont.setColumns(10);          tfemail = new
JTextField();
getFrame().getContentPane().add(tfemail);
        tfemail.setColumns(10);
         JToggleButton tglbtnUpdateSailor = new JToggleButton("update");
        tglbtnUpdateSailor.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try
                {
                    Statement statement = connection.createStatement();
                    int i = statement.executeUpdate("UPDATE PERSONS "
                    + "SET NAME ='" + tfname.getText() + "', "
                    + "AGE = " + tfage.getText() + ","
                    + "CONTACT = '" + tfcont.getText()
                    + "',"+"EMAIL_ID='"+tfemail.getText()
                    + " 'WHERE PID= "
                    + list.getSelectedItem());
                    ftf.append("\nUpdated " + i + " rows successfully");
                    list.removeAll();
                    loadList();
                }
                catch (SQLException insertException)
                {
                    ftf.append(insertException.getMessage());
}}});
        getFrame().getContentPane().add(tglbtnUpdateSailor);
JToggleButton tglbtnBack = new JToggleButton("back");
tglbtnBack.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
                try {
                    MAINGUI window = new MAINGUI();
                    (window).getframe().setVisible(true);
                } catch (Exception es) {
                    es.printStackTrace();
                }}});
        frame.getContentPane().add(tglbtnBack);
        loadList();
        list.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent arg0) {
                try
                {
                    rs = statement.executeQuery("SELECT * FROM PERSONS where
PID ="+list.getSelectedItem());
                    rs.next();
                    tfid.setText(rs.getString("PID"));
tfname.setText(rs.getString("NAME"));
tfage.setText(rs.getString("AGE"));
tfcont.setText(rs.getString("CONTACT"));
tfemail.setText(rs.getString("email_id"));


            }
                catch (SQLException selectException)
                {
                    ftf.append(selectException.getMessage());
                }}});
        frame.getContentPane().add(list);
getFrame().setBounds(100, 100, 450, 300);
getFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public JFrame getFrame() {
        return frame;
    }

    public void setFrame(JFrame frame) {

        this.frame = frame;

    }

}
```

2.Insert

```
import java.awt.EventQueue;

import java.awt.TextArea;

import javax.swing.JFrame;

import java.awt.CardLayout;

import java.awt.Color;

import javax.swing.SpringLayout;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JFormattedTextField;
import javax.swing.JToggleButton;
import java.awt.event.ActionListener;
import java.sql.Connection;

import java.sql.DriverManager;
```

```
 import java.sql.SQLException;

import java.sql.Statement;

import java.awt.event.ActionEvent;


public class PGUI2 {
        TextArea ftf = new TextArea();
        Connection connection;
Statement statement;
private JFrame frame;
private JTextField tfname;
private JTextField tfid;
private JTextField tfage;
private JTextField tfcont;
private JTextField tfemail;
        public PGUI2()
        {try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);

                }

                connection();

                initialize();

        }

        private void connection() {

                try

                {

                  connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","greta","GRETA0910");

                  statement = connection.createStatement();

                }

                catch (SQLException connectException)

                {

                  System.out.println(connectException.getMessage());

                  System.out.println(connectException.getSQLState());

                  System.out.println(connectException.getErrorCode());

                  System.exit(1);

                }

        }

        public static void main(String[] args) {

                EventQueue.invokeLater(new Runnable() {

                        public void run() {
```

```
                try {
                    PGUI2 window = new PGUI2();
                    window.getFrame().setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }}});}
    private void initialize() {
        setFrame(new JFrame());
        getFrame().getContentPane().setBackground(Color.PINK);
SpringLayout springLayout = new SpringLayout();
getFrame().getContentPane().setLayout(springLayout);


        JLabel lblNewLabel = new JLabel("Person Name:");
getFrame().getContentPane().add(lblNewLabel);
        JLabel lblNewLabel_1 = new JLabel("Person Id:");
        getFrame().getContentPane().add(lblNewLabel_1);
JLabel lblNewLabel_2 = new JLabel("Age:")
getFrame().getContentPane().add(lblNewLabel_2);
JLabel lblNewLabel_3 = new JLabel("Contact:");
getFrame().getContentPane().add(lblNewLabel_3);
JLabel lblEmailId = new JLabel("Email Id:");
       getFrame().getContentPane().add(lblEmailId);
        tfname = new JTextField();
```

```
                getFrame().getContentPane().add(tfname);
                tfname.setColumns(10);
                tfid = new JTextField();
                getFrame().getContentPane().add(tfid);
                tfid.setColumns(10);
                tfage = new JTextField();
                getFrame().getContentPane().add(tfage);
                tfage.setColumns(10);


                tfcont = new JTextField();
getFrame().getContentPane().add(tfcont);
                tfcont.setColumns(10);
tfemail = new JTextField();
getFrame().getContentPane().add(tfemail);
                tfemail.setColumns(10);
                getFrame().getContentPane().add(ftf);
                JToggleButton tglbtnUpdateSailor = new JToggleButton("Insert Person");
tglbtnUpdateSailor.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                try
                                {
                                  Statement statement = connection.createStatement();
                                   String query= "INSERT INTO PERSONS VALUES(" + tfid.getText() +
", " + "'" + tfname.getText() + "'," +tfage.getText() + "," + "'" + tfcont.getText() +
"',"+"'"+tfemail.getText()+"'"+")";

                                          int i = statement.executeUpdate(query);

  ftf.append("\nInserted " + i + " rows successfully");

                                        }

                                        catch (SQLException insertException)

                                        {



ftf.append(insertException.getMessage());}}});

getFrame().getContentPane().add(tglbtnUpdateSailor);

JToggleButton tglbtnBack = new JToggleButton("back");

tglbtnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

                                        try {

                                                  MAINGUI window = new MAINGUI();

                                                  (window).getframe().setVisible(true);

                              } catch (Exception es) {

                                                  es.printStackTrace();

                                        }}});

                    frame.getContentPane().add(tglbtnBack);

getFrame().setBounds(100, 100, 450, 300);

getFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

          }
```

```
    public JFrame getFrame() {

        return frame;

    }


    public void setFrame(JFrame frame) {

        this.frame = frame;

}}
```

3.Delete

```
import java.awt.EventQueue;

import java.awt.TextArea;

import javax.swing.JFrame;

import java.awt.CardLayout;

import java.awt.Color;

import java.awt.BorderLayout;
import javax.swing.SpringLayout;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.JFormattedTextField;
import javax.swing.JToggleButton;
import java.awt.event.ActionListener;
import java.sql.Connection; import
java.sql.DriverManager;

import java.sql.ResultSet; import
java.sql.SQLException; import
java.sql.Statement; import
java.awt.event.ActionEvent; import
java.awt.List; import
java.awt.event.ItemListener; import
java.awt.event.ItemEvent;


public class PGUI3 {
        TextArea ftf = new TextArea();
        Connection connection;
Statement statement;
```

```
private JFrame frame;

private JTextField tfname;

private JTextField tfid;

private JTextField tfage;

private JTextField tfcont;

private JTextField tfemail;      List

list = new List(10);     ResultSet

rs;        public PGUI3()

{                    try

                     {          Class.forName("oracle.jdbc.driver.OracleDriver");

                     }

                     catch (Exception e)

                     {

                                System.err.println("Unable to find and load driver");

                                System.exit(1);

                     }

                     connectToDB();

                     initialize();

           }


           private void connectToDB() {

                     try

                     {
```

```
         connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","greta","GRETA0910");
         statement = connection.createStatement();


        }
        catch (SQLException connectException)
        {
         System.out.println(connectException.getMessage());
         System.out.println(connectException.getSQLState());
         System.out.println(connectException.getErrorCode());
         System.exit(1);
        }
  }
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    PGUI3 window = new PGUI3();
                    window.getFrame().setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }}});}
    private void loadList()
```

```
	{		try
			{
			  rs = statement.executeQuery("SELECT PID FROM PERSONS");
	  while (rs.next())
			  {
					list.add(rs.getString("PID"));
			  }
			}
			catch (SQLException e)
			{
			  ftf.append(e.getMessage());
			}}
	private void initialize() {
			setFrame(new JFrame());
			getFrame().getContentPane().setBackground(Color.PINK);
SpringLayout springLayout = new SpringLayout();
getFrame().getContentPane().setLayout(springLayout);

			JLabel lblNewLabel = new JLabel("Person Name:");
getFrame().getContentPane().add(lblNewLabel);

			JLabel lblNewLabel_1 = new JLabel("Person Id:");
	getFrame().getContentPane().add(lblNewLabel_1);
```

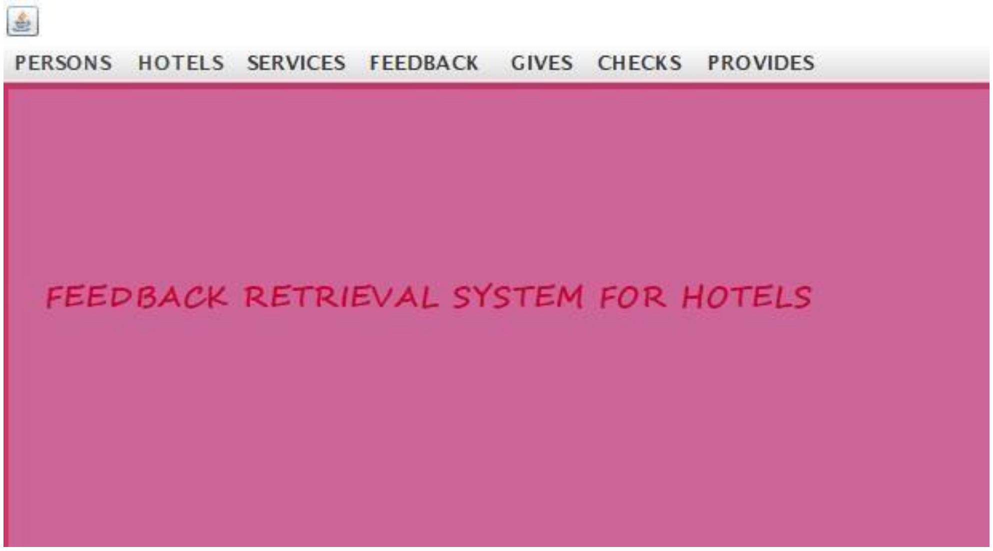
```
JLabel lblNewLabel_2 = new JLabel("Age:");
getFrame().getContentPane().add(lblNewLabel_2);
JLabel lblNewLabel_3 = new JLabel("Contact:");
getFrame().getContentPane().add(lblNewLabel_3);
            JLabel lblEmailId = new JLabel("Email Id:");

            getFrame().getContentPane().add(lblEmailId);
tfname = new JTextField();
getFrame().getContentPane().add(tfname);
            tfname.setColumns(10);
            tfid = new JTextField();
            getFrame().getContentPane().add(tfid);
            tfid.setColumns(10);
            tfage = new JTextField();
getFrame().getContentPane().add(tfage);
            tfage.setColumns(10);
tfcont = new JTextField();
getFrame().getContentPane().add(tfcont);
            tfcont.setColumns(10);
tfemail = new JTextField();
getFrame().getContentPane().add(tfemail);
            tfemail.setColumns(10);
            getFrame().getContentPane().add(ftf);
```

```
        JToggleButton tglbtnDeletePerson = new JToggleButton("delete");
tglbtnDeletePerson.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try
                {
                    Statement statement = connection.createStatement();
                    int i = statement.executeUpdate("DELETE FROM PERSONS WHERE PID = "
                            + list.getSelectedItem());
                    ftf.append("\nDeleted " + i + " rows successfully");
                    tfid.setText(null);
                    tfname.setText(null);
            tfcont.setText(null);
                    tfage.setText(null);
                    tfcont.setText(null);
    tfemail.setText(null);
list.removeAll();
                    loadList();
                }
                catch (SQLException insertException)
                {
                    ftf.append(insertException.getMessage());
                }
            }
        }
    );
        getFrame().getContentPane().add(tglbtnDeletePerson);
JToggleButton tglbtnBack = new JToggleButton("back");
tglbtnBack.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
                try {
                    MAINGUI window = new MAINGUI();
                    (window).getframe().setVisible(true);
                } catch (Exception es) {
                    es.printStackTrace();
                }}});
        frame.getContentPane().add(tglbtnBack);
        loadList();
        list.addItemListener(new ItemListener() {
public void itemStateChanged(ItemEvent arg0) {
                try
                {
                    rs = statement.executeQuery("SELECT * FROM PERSONS where PID ="+list.getSelectedItem());
                    rs.next();
                    tfid.setText(rs.getString("PID"));
tfname.setText(rs.getString("NAME"));
tfage.setText(rs.getString("AGE"));
tfcont.setText(rs.getString("CONTACT"));
tfemail.setText(rs.getString("email_id"));


            }
                catch (SQLException selectException)
                {
                    ftf.append(selectException.getMessage());
                }}});
        frame.getContentPane().add(list);
        getFrame().setBounds(100, 100, 450, 300);
        getFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
public JFrame getFrame() {
        return frame;
    }


    public void setFrame(JFrame frame) {
        this.frame = frame;
    }
}
```

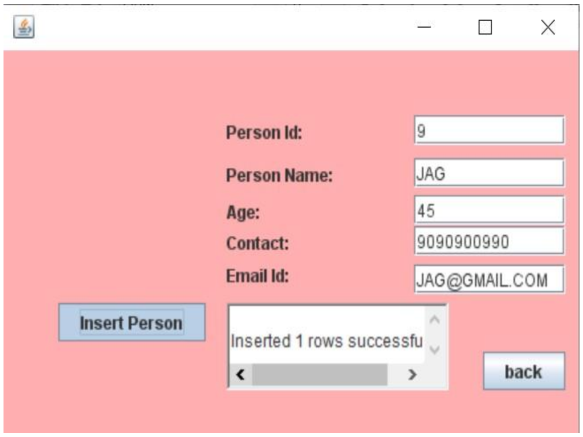
OUTPUT SCREENSHOTS:

Java GUI Screenshot:

GUI MAIN:



GUI OF TABLE : PERSONS



Person Id: 9

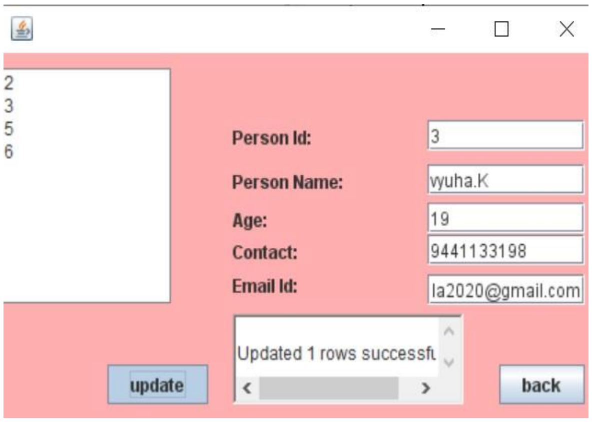
Person Name: JAG

Age: 45

Contact: 9090900990

Email Id: JAG@GMAIL.COM

Inserted 1 rows successfu



2
3
5
6

Person Id: 3

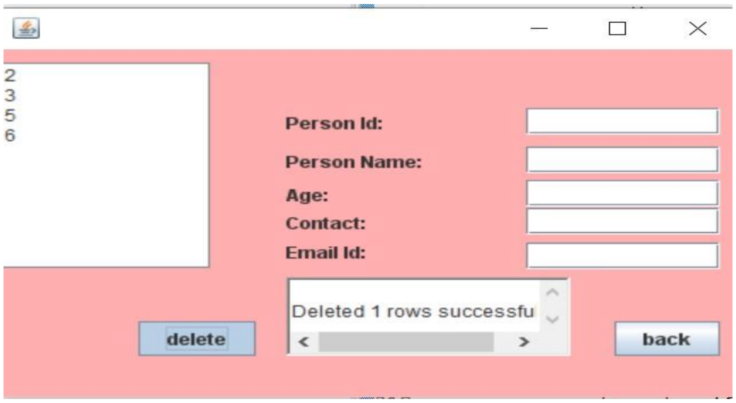
Person Name: vyuha.K

Age: 19

Contact: 9441133198

Email Id: la2020@gmail.com

Updated 1 rows successfu



2
3
5
6

Person Id:

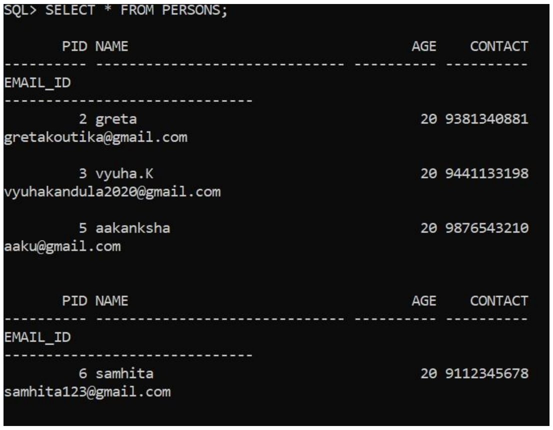
Person Name:

Age:

Contact:

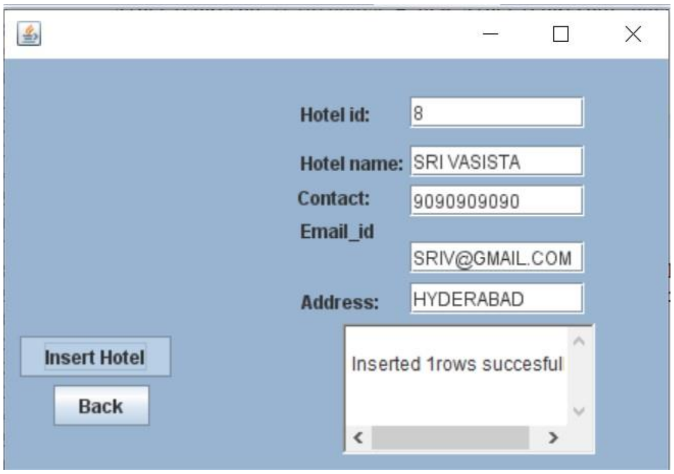
Email Id:

Deleted 1 rows successfu



```
SQL> SELECT * FROM PERSONS;

       PID NAME                                  AGE    CONTACT
---------- ------------------------------ ---------- ----------
EMAIL_ID
------------------------------
         2 greta                                  20 9381340881
gretakoutika@gmail.com

         3 vyuha.K                                20 9441133198
vyuhakandula2020@gmail.com

         5 aakanksha                              20 9876543210
aaku@gmail.com


       PID NAME                                  AGE    CONTACT
---------- ------------------------------ ---------- ----------
EMAIL_ID
------------------------------
         6 samhita                                20 9112345678
samhita123@gmail.com
```

GUI OF TABLE HOTELS:



Hotel id: 8

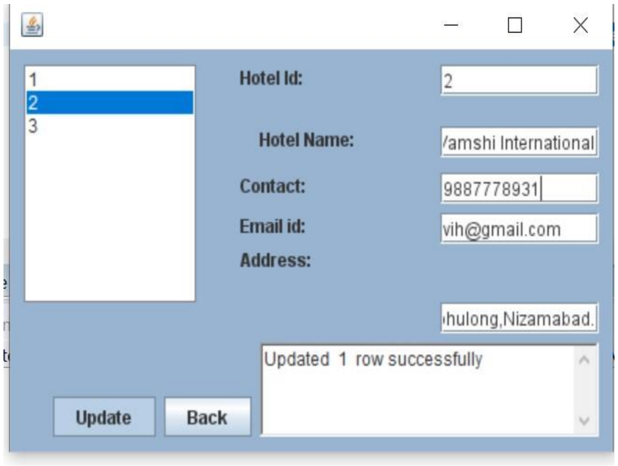
Hotel name: SRI VASISTA

Contact: 9090909090

Email_id SRIV@GMAIL.COM

Address: HYDERABAD

Inserted 1rows succesful



1
2
3

Hotel Id: 2

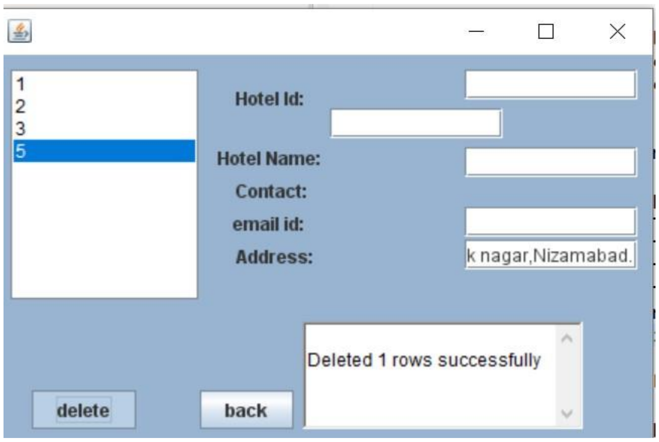
Hotel Name: /amshi International

Contact: 9887778931

Email id: vih@gmail.com

Address: ›hulong,Nizamabad.

Updated 1 row successfully



1
2
3
5

Hotel Id:

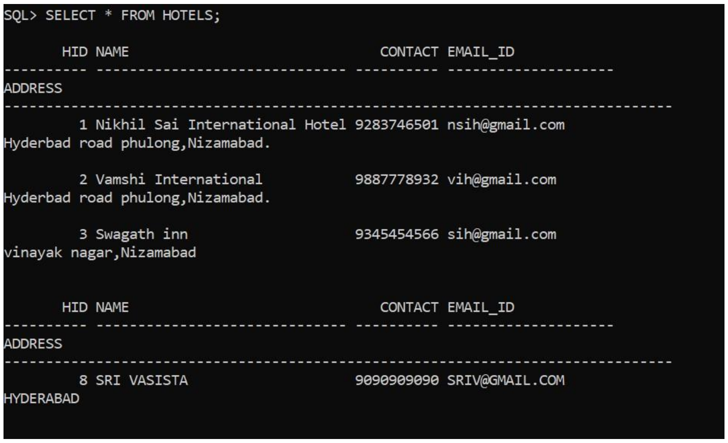
Hotel Name:

Contact:

email id:

Address: k nagar,Nizamabad.

Deleted 1 rows successfully

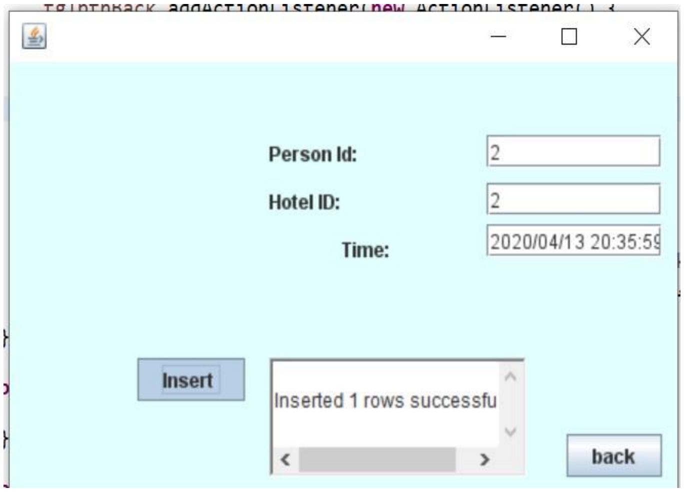


```
SQL> SELECT * FROM HOTELS;

       HID NAME                              CONTACT EMAIL_ID
---------- ------------------------------ ---------- --------------------
ADDRESS
--------------------------------------------------------------------------------
         1 Nikhil Sai International Hotel 9283746501 nsih@gmail.com
Hyderbad road phulong,Nizamabad.

         2 Vamshi International           9887778932 vih@gmail.com
Hyderbad road phulong,Nizamabad.

         3 Swagath inn                    9345454566 sih@gmail.com
vinayak nagar,Nizamabad


       HID NAME                              CONTACT EMAIL_ID
---------- ------------------------------ ---------- --------------------
ADDRESS
--------------------------------------------------------------------------------
         8 SRI VASISTA                    9090909090 SRIV@GMAIL.COM
HYDERABAD
```

GUI OF RELATION CHECKS:

ROLL NUMBER:1602-18-737-069

NAME:K.GRETA

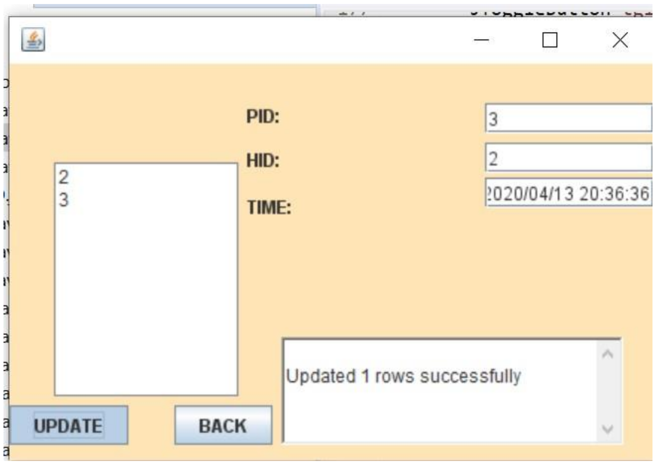


Person Id: 2

Hotel ID: 2

Time: 2020/04/13 20:35:59

Inserted 1 rows successfu



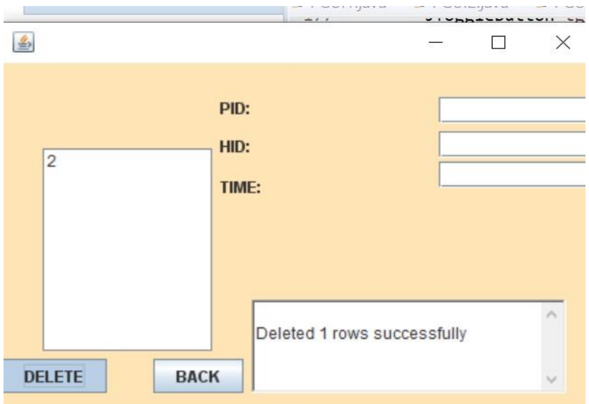
2
3

PID: 3

HID: 2

TIME: 2020/04/13 20:36:36

Updated 1 rows successfully



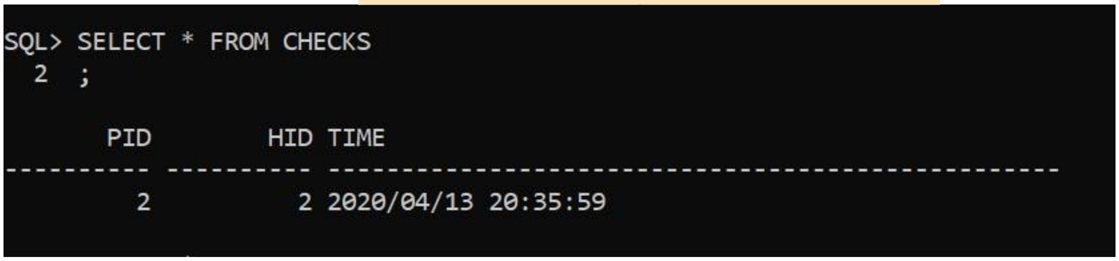
2

PID:

HID:

TIME:

Deleted 1 rows successfully

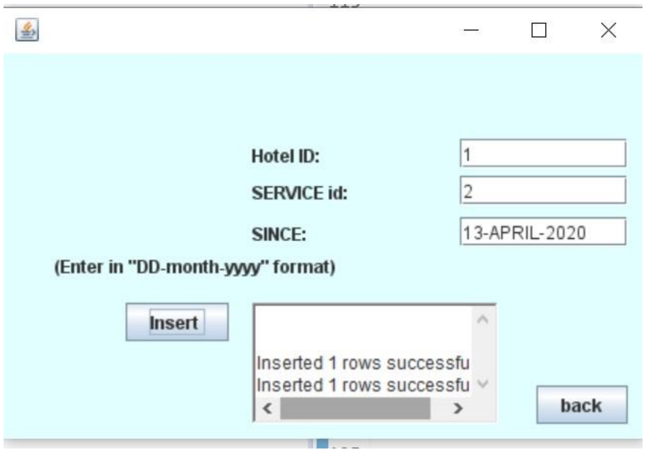


```
SQL> SELECT * FROM CHECKS
  2  ;

       PID        HID TIME
---------- ---------- --------------------------------------------------
         2          2 2020/04/13 20:35:59
```

GUI OF RELATION PROVIDES:

ROLL NUMBER:1602-18-737-069

NAME:K.GRETA



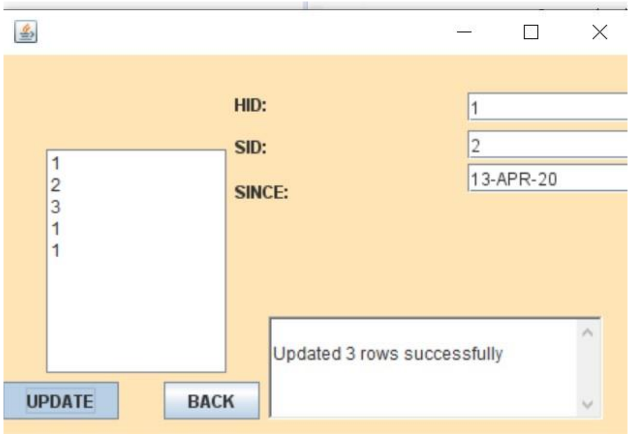
Hotel ID: 1

SERVICE id: 2

SINCE: 13-APRIL-2020

(Enter in "DD-month-yyyy" format)

```
Inserted 1 rows successfu
Inserted 1 rows successfu
```



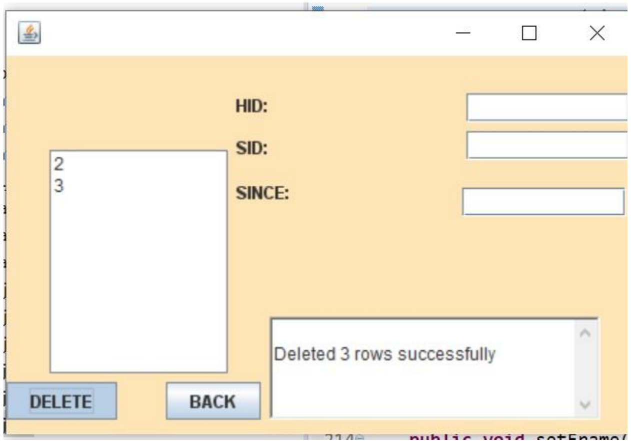
HID: 1

SID: 2

SINCE: 13-APR-20

```
1
2
3
1
1
```

Updated 3 rows successfully



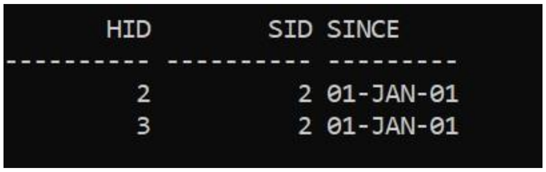
HID:

SID:

SINCE:

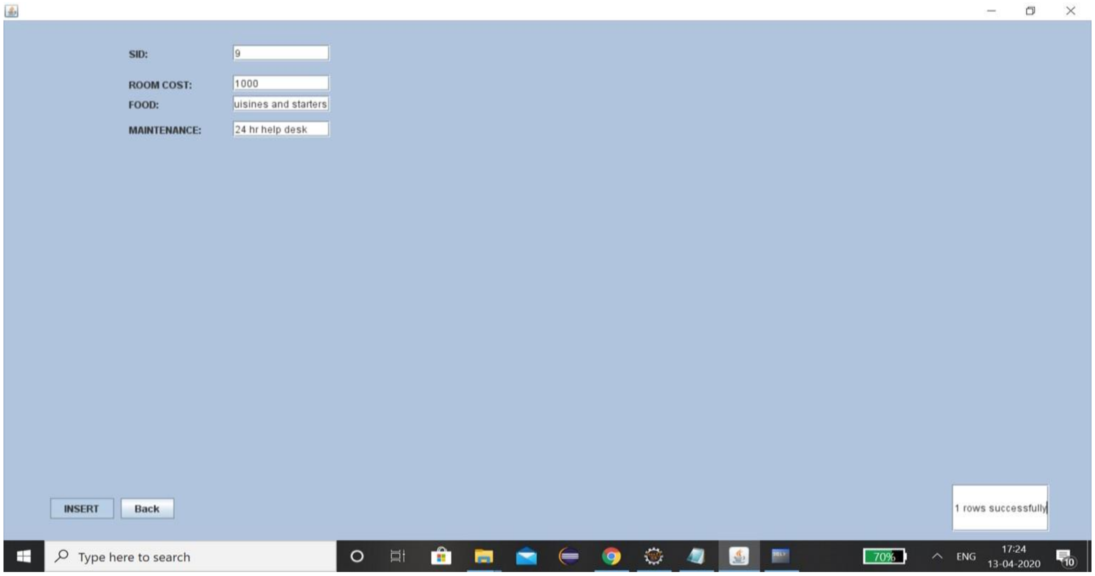
```
2
3
```

Deleted 3 rows successfully



```
       HID        SID SINCE
---------- ---------- ---------
         2          2 01-JAN-01
         3          2 01-JAN-01
```

GUI OF TABLE SERVICES:



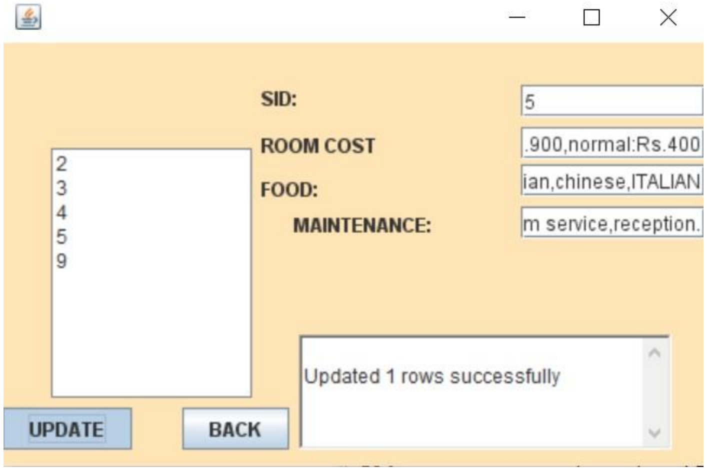
SID: 9

ROOM COST: 1000

FOOD: uisines and starters

MAINTENANCE: 24 hr help desk

1 rows successfully



SID: 5

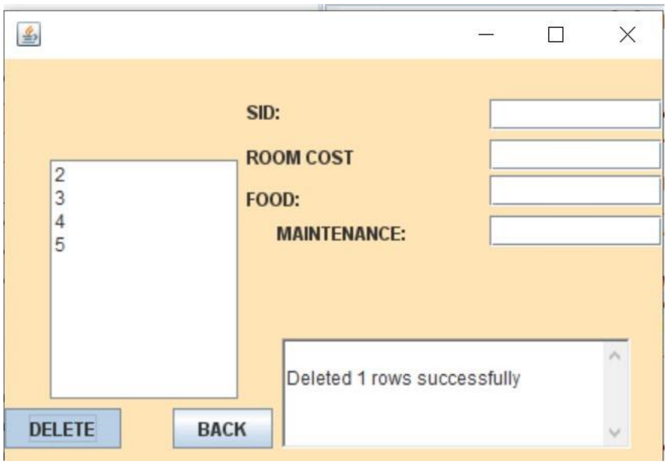
ROOM COST .900,normal:Rs.400

FOOD: ian,chinese,ITALIAN

MAINTENANCE: m service,reception.

2
3
4
5
9

Updated 1 rows successfully



SID:

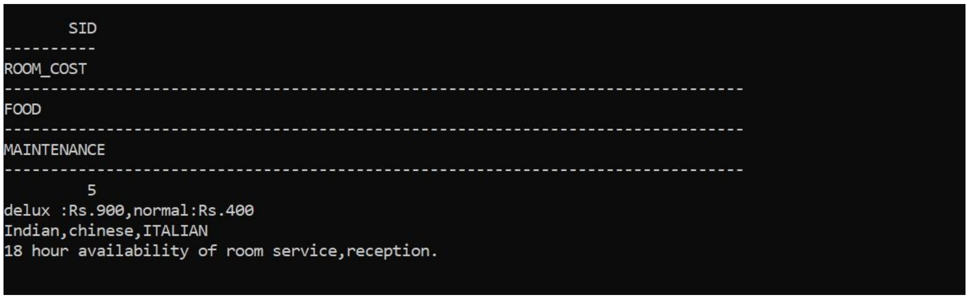
ROOM COST

FOOD:

MAINTENANCE:

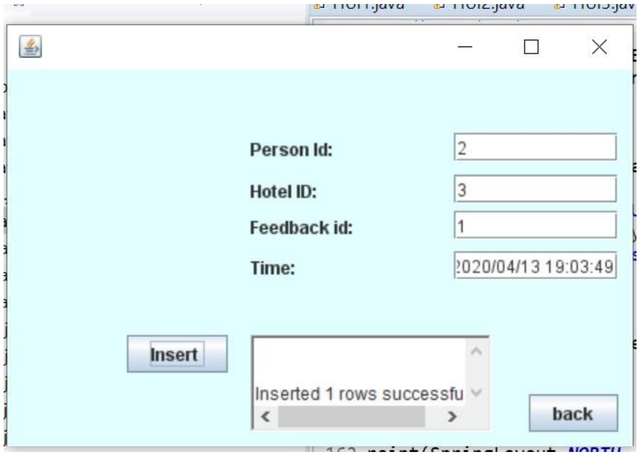
2
3
4
5

Deleted 1 rows successfully



```
       SID
----------
ROOM_COST
--------------------------------------------------------------------------------
FOOD
--------------------------------------------------------------------------------
MAINTENANCE
--------------------------------------------------------------------------------
         5
delux :Rs.900,normal:Rs.400
Indian,chinese,ITALIAN
18 hour availability of room service,reception.
```

GUI OF TABLE GIVES:



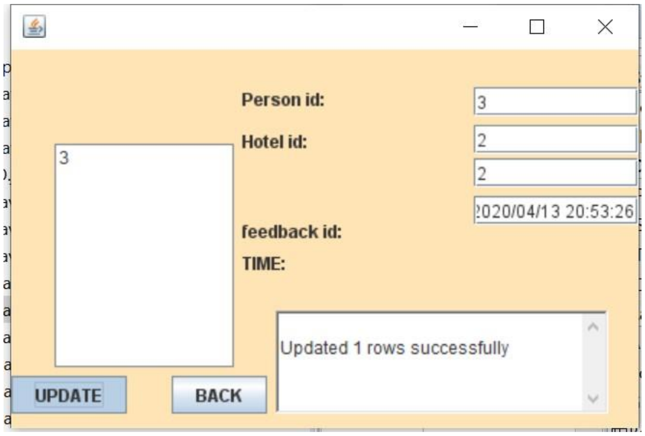
Person Id: 2

Hotel ID: 3

Feedback id: 1

Time: 2020/04/13 19:03:49

Inserted 1 rows successfu



Person id: 3

Hotel id: 2

2

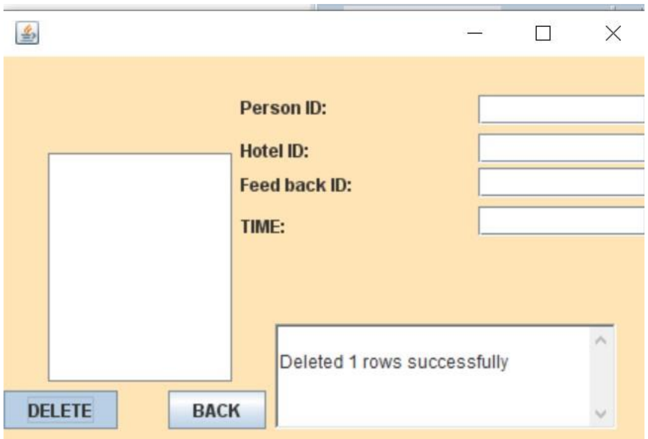
2020/04/13 20:53:26

feedback id:

TIME:

3

Updated 1 rows successfully



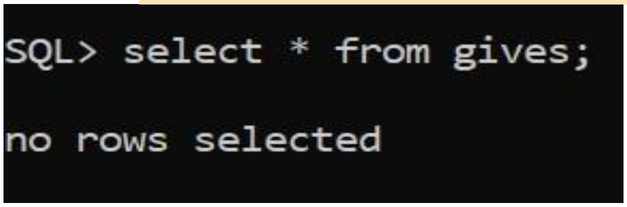
Person ID:

Hotel ID:

Feed back ID:

TIME:

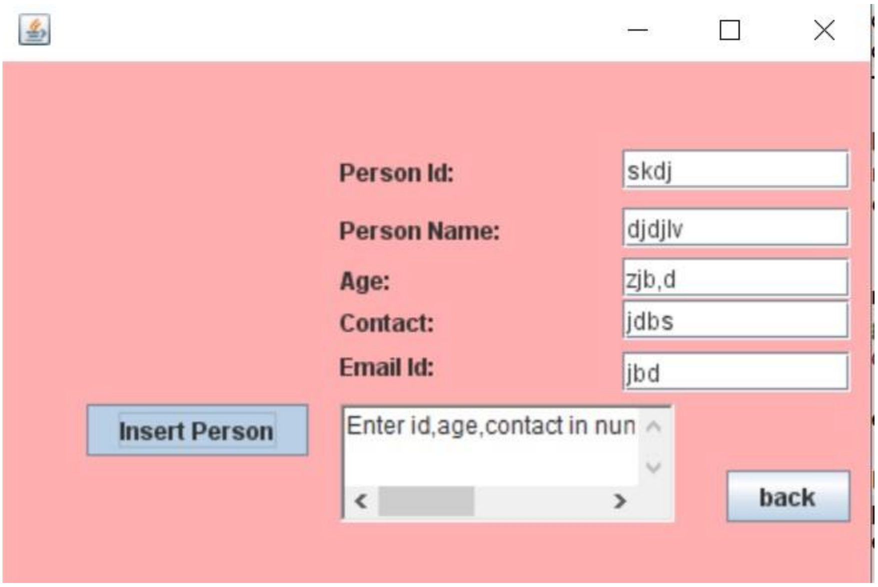
Deleted 1 rows successfully



```
SQL> select * from gives;

no rows selected
```

Testing:

1.If you enter the wrong data it prompts the user saying “Enter id, age, contact in numbers**”.**



Person Id: skdj

Person Name: djdjlv

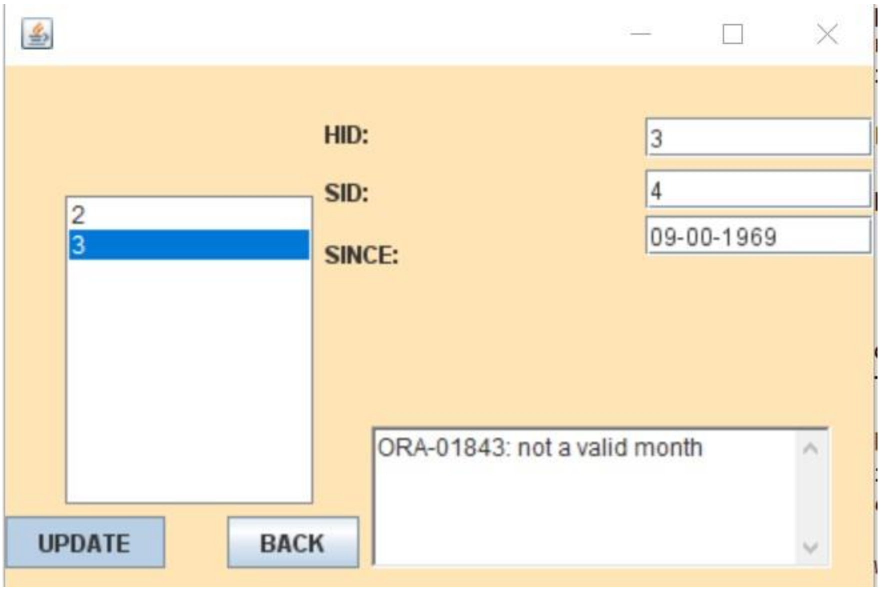
Age: zjb,d

Contact: jdbs

Email Id: jbd

```
Enter id,age,contact in nun
```

2.If you enter the wrong date of birth it will show an error.



HID: 3

SID: 4

SINCE: 09-00-1969

2
3

```
ORA-01843: not a valid month
```

So just click on the text field so that it automatically gives the time .

Github link:

https://github.com/Gretakoutika/database-management-system.git

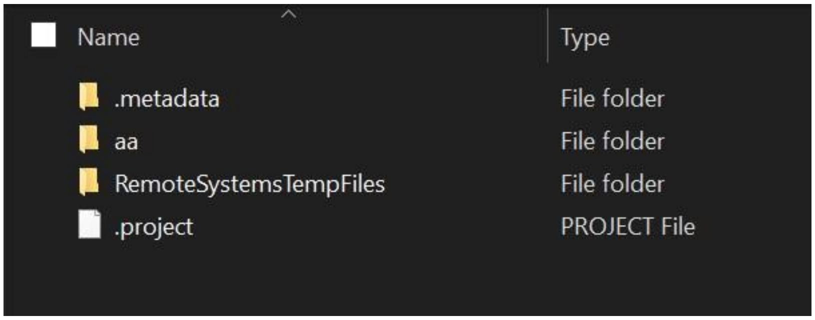
File folder structure:

1. You will find a folder named “aa”.
2. In src you will find all the files where in .java files contain the code
3.

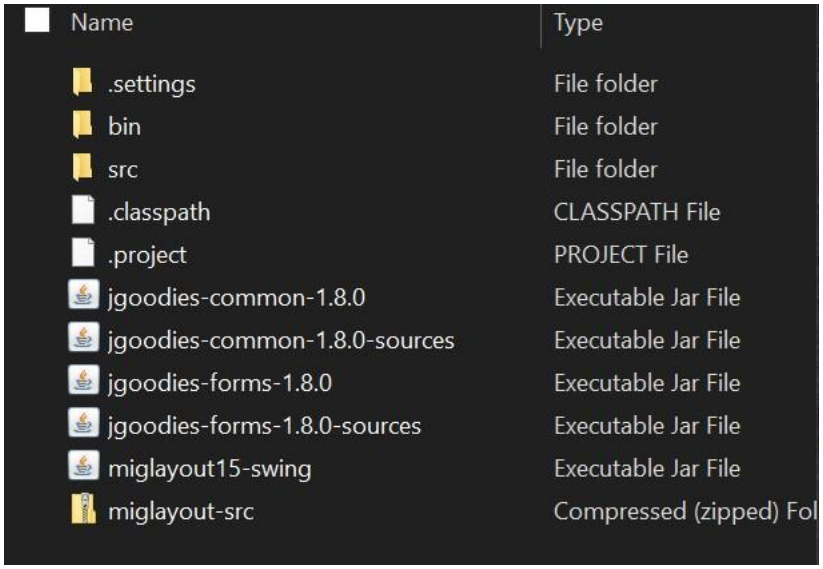
- PUI1-Persons UI(insert).
- PUI2- persons UI(modify).
- PUI3- persons UI(Delete).

4.Similarly,

- HUI-Hotels.
- CUI-Checks.
- PRUI-Provides.
- GUI-Gives.
- FUI-Feedback.
- SUI-Services.



Name	Type
.metadata	File folder
aa	File folder
RemoteSystemsTempFiles	File folder
.project	PROJECT File



Name	Type
.settings	File folder
bin	File folder
src	File folder
.classpath	CLASSPATH File
.project	PROJECT File
jgoodies-common-1.8.0	Executable Jar File
jgoodies-common-1.8.0-sources	Executable Jar File
jgoodies-forms-1.8.0	Executable Jar File
jgoodies-forms-1.8.0-sources	Executable Jar File
miglayout15-swing	Executable Jar File
miglayout-src	Compressed (zipped) Fol

DISCUSSION AND FUTURE WORK:

I want to create an app based on this mini project through which the data can be uploaded to the database so easily and also the backup will be maintained so that the data will be secured if there is any issue with the database.

CONCLUSION:

Thus, this Java AWT feedback retrieval system created which is connected to the Oracle 11g database. Therefore, all the entries in the form are directly updated on the tables created in the database.

ROLL NUMBER:1602-18-737-069

NAME:K.GRETA

References:

- **https://docs.oracle.com/javase/7/docs/api/**
- **DATA SYTSEMS CONCEPT BY :HENRY F KORTH.**
- **https://javatpoint.com/dbms-tutorial**