UCS1412 – Database Lab
Assignment – 10
Title: Database Application Programming using JDBC

Name: Aviansh Gupta
Reg No: 185001028
CSE - A

Table, Procedures and Functions Definitions

```
--Table Creation
CREATE TABLE Emp_Payroll(
eid varchar2(4) CONSTRAINT Empid_pk PRIMARY KEY,
ename varchar2(15),
dob date,
sex varchar2(1),
designation varchar2(20),
basic float(8),
da float(8),
hra float(8),
pf float(8),
mc float(8),
gross float(8),
tot_deduc float(8),
net_pay float(8));

-- Procedure for calculating Net Pay
CREATE OR REPLACE PROCEDURE calNet_Pay
(id IN Emp_Payroll.eid%TYPE,BP IN Emp_Payroll.basic%TYPE) IS
      DAt Emp_Payroll.da%TYPE;
      HRAt Emp_Payroll.hra%TYPE;
      PFt Emp_Payroll.pf%TYPE;
      MCt Emp_Payroll.mc%TYPE;
      GROSSt Emp_Payroll.gross%TYPE;
      TOT_DED Emp_Payroll.tot_deduc%TYPE;
      NP Emp_Payroll.net_pay%TYPE;
BEGIN
      DAt:= 0.6*BP;
      HRAt:= 0.11*BP;
      PFt:= 0.04*BP;
      MCt:= 0.03*BP;
      GROSSt:= BP+DAt+HRAt;
      TOT_DED:= PFt+MCt;
```

```
      NP:= GROSSt-TOT_DED;
      update Emp_Payroll
      set da = DAt,
      hra = HRAt,
      pf = PFt,
      mc = MCt,
      gross = GROSSt,
      tot_deduc = TOT_DED,
      net_pay = NP
      where eid = id;
END;
```

-- Procedure to Insert Employee Data

```
CREATE OR REPLACE PROCEDURE insertEmp
(id IN Emp_Payroll.eid%TYPE,name IN Emp_Payroll.ename%TYPE,birthd IN
date,gender IN Emp_Payroll.sex%TYPE,desig IN
Emp_Payroll.designation%TYPE,bp IN Emp_Payroll.basic%TYPE) IS
BEGIN
      insert into Emp_Payroll
values(id,name,birthd,gender,desig,bp,NULL,NULL,NULL,NULL,NULL,NULL,N
ULL);
      calNet_Pay(id,bp);
END;
```

Note: SQL%FOUND condition not added in update and delete procedures because it's already checked in the app. Program before calling.

-- Procedure to update Employee data

```
CREATE OR REPLACE PROCEDURE updateEmp
(id IN Emp_Payroll.eid%TYPE,name IN Emp_Payroll.ename%TYPE,birthd IN
date,gender IN Emp_Payroll.sex%TYPE,desig IN
Emp_Payroll.designation%TYPE,bp IN Emp_Payroll.basic%TYPE) IS
BEGIN
      update Emp_Payroll
      set ename = name,dob = birthd,sex = gender,designation =
desig,basic = bp
      where eid = id;
      calNet_Pay(id,bp);
END;
```

-- Procedure to delete Employee data

```
CREATE OR REPLACE PROCEDURE deleteEmp
(id IN Emp_Payroll.eid%TYPE) IS
BEGIN
      delete from Emp_Payroll where eid = id;
```

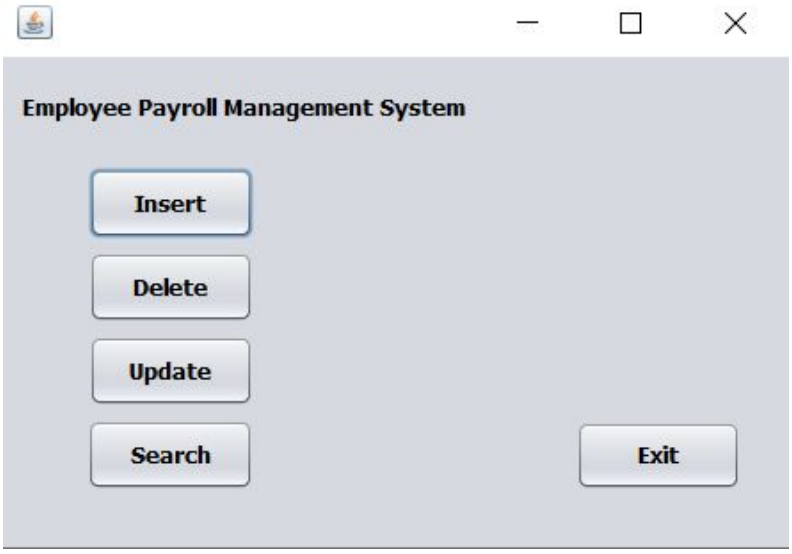
```
END;

-- Procedure to search Employee used in Sql Plus
CREATE OR REPLACE PROCEDURE searchEmp
(id IN Emp_Payroll.eid%TYPE) IS
     entry Emp_Payroll%ROWTYPE;
     CURSOR c1 IS select * from Emp_Payroll where eid = id;
BEGIN
     OPEN c1;
     FETCH c1 INTO entry;
     IF c1%FOUND THEN
          dbms_output.put_line('Eid: '||entry.eid);
          dbms_output.put_line('Name: '||entry.ename);
          dbms_output.put_line('DOB: '||entry.dob);
          dbms_output.put_line('Gender: '||entry.sex);
          dbms_output.put_line('Designation: '||entry.designation);
          dbms_output.put_line('Basic Pay: '||entry.basic);
          dbms_output.put_line('DA: '||entry.da);
          dbms_output.put_line('HRA: '||entry.hra);
          dbms_output.put_line('PF: '||entry.pf);
          dbms_output.put_line('MC: '||entry.mc);
          dbms_output.put_line('Gross Pay: '||entry.gross);
          dbms_output.put_line('Deductions: '||entry.tot_deduc);
          dbms_output.put_line('Net Pay: '||entry.net_pay);
     ELSE
          dbms_output.put_line('EID Not Found');
     END IF;
END;
-- Function to search Employee used in Netbeans Application
CREATE OR REPLACE FUNCTION search_Emp
(id IN Emp_Payroll.eid%TYPE) RETURN int IS
     entry Emp_Payroll%ROWTYPE;
     CURSOR c1 IS select * from Emp_Payroll where eid = id;
BEGIN
     OPEN c1;
     FETCH c1 INTO entry;
     IF c1%FOUND THEN
          RETURN 1;
     ELSE
          RETURN 0;
     END IF;
END;
```

Main Connection Code Used in all Operations:

```
import java.sql.*;                // Import java's SQL library

Connection conn = NULL;           // Declare connection variable

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

conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");                // Establishing connection

conn.close()    // Closing the connection after operation is complete

/* Note: This code should always be kept in a try{}catch(){} block */
```

Screenshots of the working and the GUI code:

Home Screen:



Home screen has five buttons:

1) Insert Button

Takes the user to the record insertion window.

Code:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        new insertEmp().setVisible(true);
        dispose();
}
```

2) Delete Button

Takes the user to the delete record window.

Code:

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
        new deleteEmp().setVisible(true);
        dispose();
}
```

3) Update Button:

Takes the user to the update record window.

Code:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
        new updateEmp().setVisible(true);
        dispose();
}
```

4) Search Button:

Takes the user to the search employee window.

Code:

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{
        new searchEmp().setVisible(true);
        dispose();
}
```

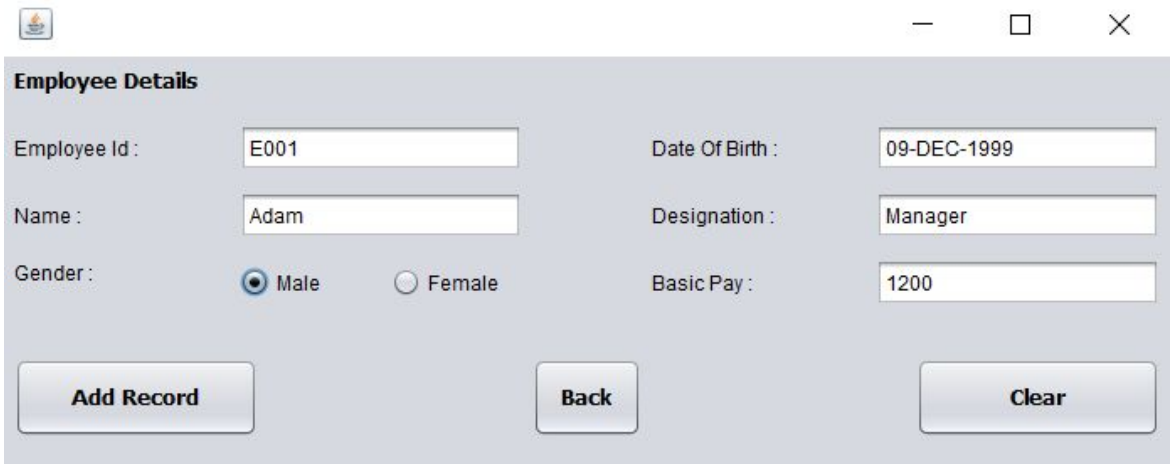
5) Exit Button

Quits the application.

Code:

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{
        System.exit(0);
}
```

Insert Employee Screen:



The insert screen has some text fields and three buttons and two radio buttons for gender selection. Users need to type respective information in each box to add the data.

1) Gender Radio Buttons:

Used to select the gender of an employee.

Code:

```
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // For Male Gender:
        gender ="M";
        jRadioButton1.setSelected(true);
        jRadioButton2.setSelected(false);
    }

    private void
jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // For Female Gender:
        gender ="F";
        jRadioButton1.setSelected(false);
        jRadioButton2.setSelected(true);
    }
//Gender is declared as a private variable in code.
```

2) Add Record Button:

Used to add the employee details to the database.

Code:

```
 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
            Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
// Above is the connection code.
//Below is the sql statement to be executed.
            String sql = "insert into Emp_Payroll"
                        +"(eid,ename,dob,sex,designation,basic)"
                        +"values (?,?,?,?,?,?)";
            pst=conn.prepareStatement(sql);
/* Adding data to the table from the text fields*/
            pst.setString(1,jTextField1.getText());
            pst.setString(2,jTextField2.getText());
            pst.setString(3,jTextField3.getText());
            pst.setString(4,gender);
            pst.setString(5,jTextField4.getText());
            pst.setFloat(6,Float.parseFloat(jTextField5.getText()));
            pst.execute(); /* Executing The SQL statement */

            cst=conn.prepareCall("{call calNet_Pay(?,?)}");
/* Execution the procedure calNet_Pay to calculate Net Pay*/
            cst.setString(1,jTextField1.getText());
            cst.setFloat(2,Float.parseFloat(jTextField5.getText()));
            cst.executeUpdate();
            JOptionPane.showMessageDialog(null,"Employee Data Added
successfully");
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
    }
```

3) Back Button

Takes the user back to the home screen.

Code:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
        new main().setVisible(true);
        dispose();
}
```

4) Clear Button

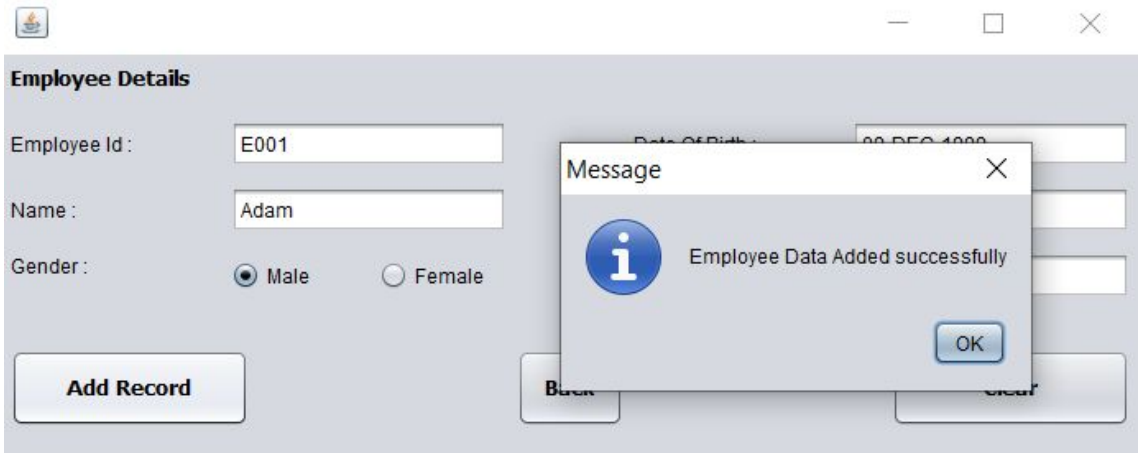
Clears the text fields on the screen.

Code:

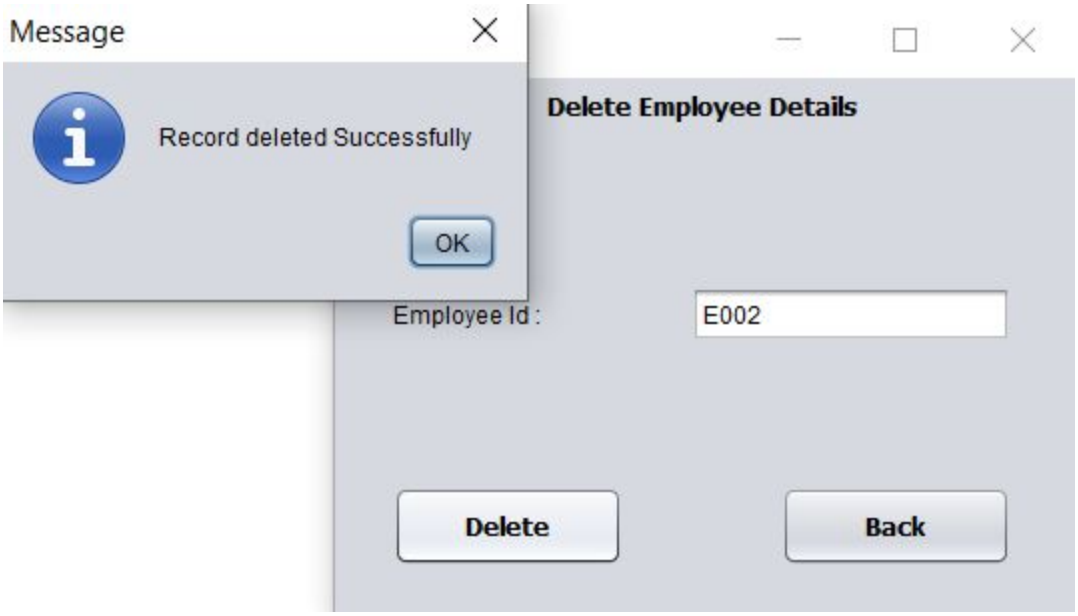
```
 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
```

```
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField5.setText("");
}
```

After writing all the details the add record button is pressed and the employee details are added to the database.



Delete Employee Details Window:



Delete screen has a text field for the employee id and two buttons.

1) Delete Button:

Used to delete the employee details corresponding to the eid supplied if found.

Code:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
            cst=conn.prepareCall("{? = call search_Emp(?)}");
            cst.setString(2,jTextField1.getText());
            cst.registerOutParameter(1, java.sql.Types.INTEGER);
            cst.execute();
            flag = cst.getInt(1);
            cst.close();
            conn.close();
/* Calling the searchEmp procedure and setting the flag to 1 if eid
found and 0 if not found */
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
        if (flag == 1){
            try{
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
                cst=conn.prepareCall("{call deleteEmp(?)}");
                cst.setString(1,jTextField1.getText());
                cst.executeUpdate();
                JOptionPane.showMessageDialog(null,"Record deleted
Successfully");
                cst.close();
                conn.close();
/* Calling the deleteEmp procedure to delete the employee details */
            }
            catch(Exception e){
                JOptionPane.showMessageDialog(null,e);
            }
        }
        else{
```

```
                JOptionPane.showMessageDialog(null,"Employee Id not
Found!!");
            }
        }
```

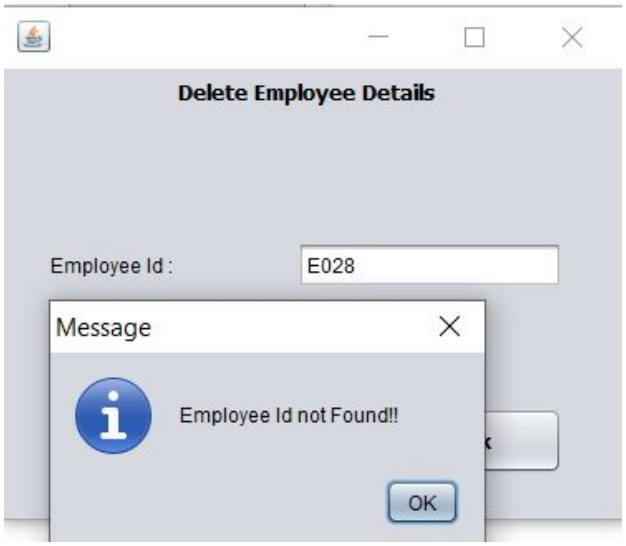
2) Back Button

Takes the user back to the home screen.

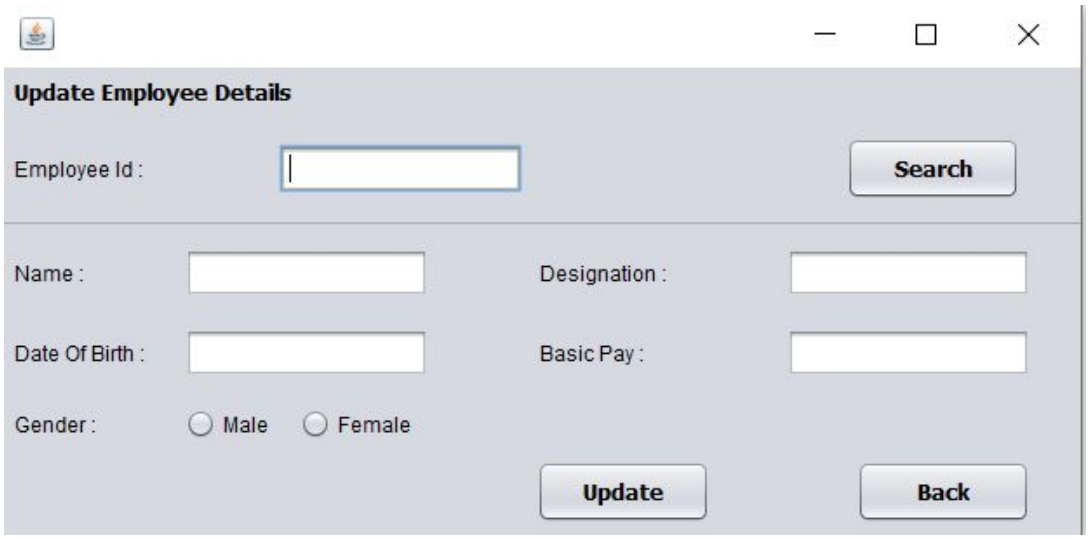
Code:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
            new main().setVisible(true);
            dispose();
}
```

If eid supplied is not found in the database an error message is shown to the user.



Update Employee Details Window:



The update window has some text fields and three buttons.

1) Search Button

Searches the eid in the database and **if found autofills the text fields** with the existing details so the user can change it and if not found an error message is displayed.

Code:

```
 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
            cst=conn.prepareCall("{? = call search_Emp(?)}");
            cst.setString(2,jTextField1.getText());
            cst.registerOutParameter(1, java.sql.Types.INTEGER);
            cst.execute();
            flag = cst.getInt(1);
            cst.close();
        }
/* Calling the searchEmp procedure and setting the flag to 1 if eid
found and 0 if not found */

        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }

        if (flag == 1){
            try{
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
                String sql ="select * from Emp_Payroll where eid=? ";
/* Auto filling the text fields with the existing details */
                pst=conn.prepareStatement(sql);
                pst.setString(1,jTextField1.getText());
                rs=pst.executeQuery();
                while(rs.next()){
                    String add1 =rs.getString(2);
                    jTextField2.setText(add1);

                    String add2 =rs.getString(3);
                    add2 = add2.substring(0,10);
```

```
                    jTextField3.setText(add2);

                    String add3 =rs.getString(5);
                    jTextField4.setText(add3);

                    String add4 =rs.getString(6);
                    jTextField5.setText(add4);
                }
                conn.close();
                pst.close();
                rs.close();
            }
            catch(Exception e){
                JOptionPane.showMessageDialog(null, e);
            }
        }
        else{
            JOptionPane.showMessageDialog(null,"Employee Id not
Found!!");
        }
    }
```

2) Update Button

Updates the employee details.

Code:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
            cst=conn.prepareCall("{call updateEmp(?,?,?,?,?,?)}");
/* Execution the procedure updateEmp */
            cst.setString(1,jTextField1.getText());
            cst.setString(2,jTextField2.getText());
            cst.setString(3,jTextField3.getText());
            cst.setString(4,gender);
            cst.setString(5,jTextField4.getText());
            cst.setFloat(6,Float.parseFloat(jTextField5.getText()));
            cst.executeUpdate();
            JOptionPane.showMessageDialog(null,"Data updated
successfully");
        }
```

```
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
    }
```

3) Back Button

Takes the user back to the home screen.

Code:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
        new main().setVisible(true);
        dispose();
}
```

4) Gender Radio Buttons

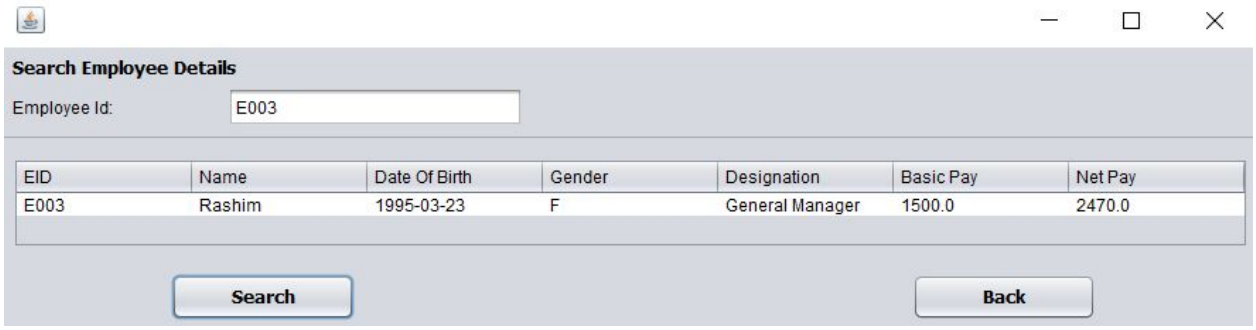
Use to change the gender.

Code:

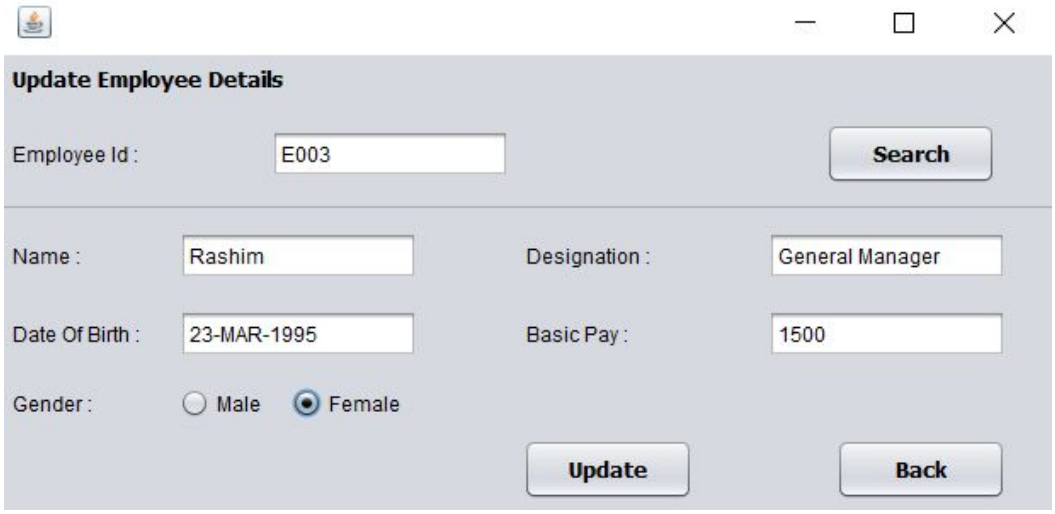
```
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        gender ="M";
        jRadioButton1.setSelected(true);
        jRadioButton2.setSelected(false);
    }

private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
        gender ="F";
        jRadioButton1.setSelected(false);
        jRadioButton2.setSelected(true);
    }
```

Working:

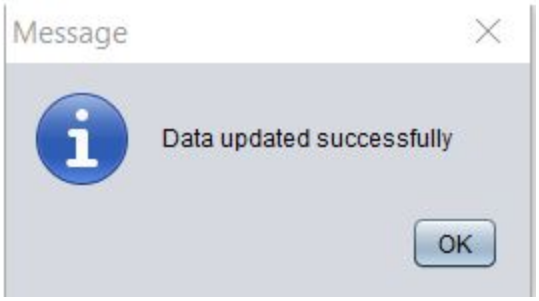
Old details of the employee whose details need to be updated.



Details found

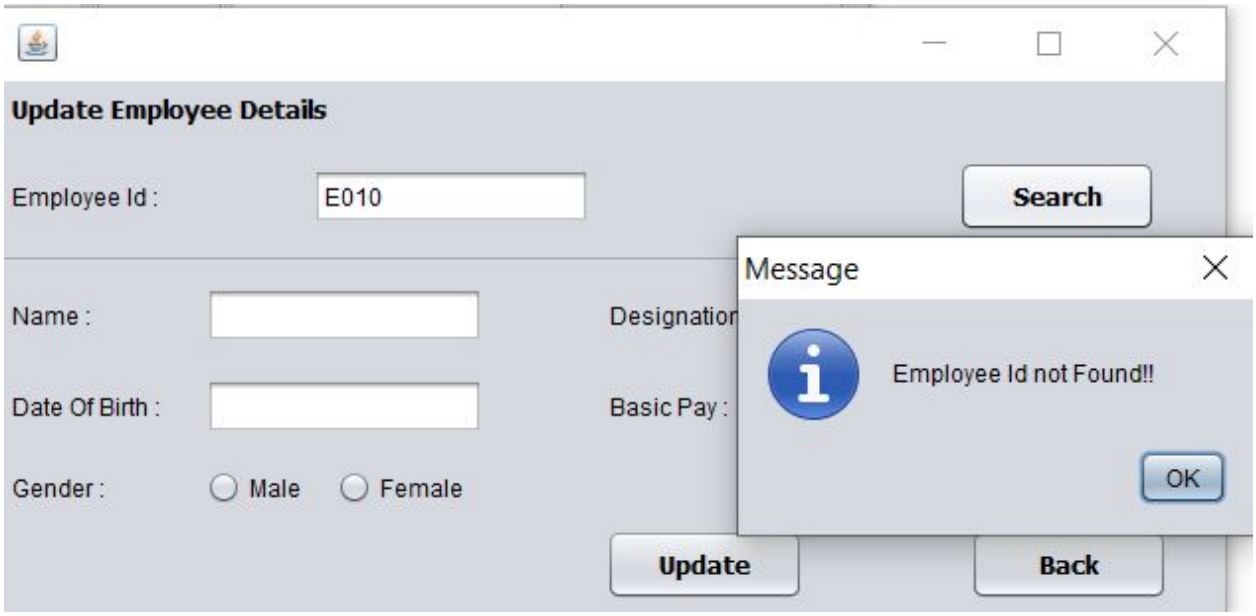


Data updated



Update shown in the search window.

If eid not found



Search Employee Details Window:

Search window has a text field and two buttons.

1) Search Button

Used to search the employee details in the database and displays if found if not found an error message is displayed.

Code:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
            cst=conn.prepareCall("{? = call search_Emp(?)}");
            cst.setString(2,jTextField1.getText());
            cst.registerOutParameter(1, java.sql.Types.INTEGER);
```

```
            cst.execute();
            flag = cst.getInt(1);
            cst.close();
            conn.close();
        }
/* Calling the searchEmp procedure and setting the flag to 1 if eid
found and 0 if not found */
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
        if (flag == 1){
            try{
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:or
cl1","scott","a1028");
                DefaultTableModel model=(DefaultTableModel)
jTable1.getModel();
                String sql ="select * from Emp_Payroll where eid=? ";
                pst=conn.prepareStatement(sql);
                pst.setString(1,jTextField1.getText());
                rs=pst.executeQuery();
                while(rs.next()){
                    String id=rs.getString(1);
                    String name=rs.getString(2);
                    String dob=rs.getString(3);
                    dob = dob.substring(0,10);
                    String gender=rs.getString(4);
                    String desig=rs.getString(5);
                    float bp = Float.parseFloat(rs.getString(6));
                    float np = Float.parseFloat(rs.getString(13));
                    model.addRow(new Object[]
                       {id,name,dob,gender,desig,bp,np});
                }

                conn.close();
                pst.close();
                rs.close();
/* Displaying the details */
            }
            catch(Exception e){
                JOptionPane.showMessageDialog(null,e);
            }
```

```
        }
        else{
             JOptionPane.showMessageDialog(null,"Employee Id not
Found!!");
        }
    }
```

2) Back Button

Takes the user back to the home screen.

Code:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
        new main().setVisible(true);
        dispose();
}
```

Working:

Search If Found

If not found.