

Laboratory 6

Title of the Laboratory Exercise: interface to the system

1. Introduction and Purpose of Experiment

A database connection is the means by which a database server and its client software communicate with each other. The client and the server can be on the same machine or on different machines. The client uses a database connection to send commands to and receive replies from the server. A database is stored as a file or a set of files on magnetic disk or tape, optical disk, or some other secondary storage device. By doing this lab, students will be able to connect the developed application with the database.

2. Aim and Objectives

Aim

- To design an interface and connect to the database

Objectives

At the end of this lab, the student will be able to

- Design and implement an interface for the application
- Connect the developed application with the database

3. Experimental Procedure

- Analyse the problem statement
- Design an interface for the given problem statement
- Connect the application with the database
- Test the implemented program
- Document the Results
- Analyse and discuss the outcomes of your experiment

4. Questions

- Consider the problem statement that you selected in Laboratory 2. Design a GUI with provision for insertion, deletion and display of a particular record in the database. Use appropriate components to display the page.

5. Calculations/Computations/Algorithms

SQL QUERIES

```
drop table if exists employee_details;
```

```
create table employee_details (  
    emp_id int,  
    emp_name varchar(255),
```

```
emp_type varchar(255),
emp_contact_no varchar(255),
emp_last_paid_on varchar(255),
emp_address varchar(255)
);

insert into employee_details values
(1, "satyajit ghana", "manager", "7892137665", "21/04/20", "bangalore"),
(2, "anusha sp", "manager", "9878786567", "19/04/20", "bangalore"),
(3, "shikhar singh", "developer", "9878783612", "21/04/20", "new delhi"),
(4, "samhitha r", "ceo", "9877873273", "04/03/20", "kolkata"),
(5, "vivek badani", "employee", "8878776743", "01/01/20", "mumbai"),
(6, "shubham a", "mail man", "7411856216", "02/02/20", "gandhinagar")
;

select * from employee_details;
```

EmployeeDetails.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab06;

import java.util.ArrayList;
import java.util.Map;
import java.sql.*;
import java.util.HashMap;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;

/**
 *
 * @author shadowleaf
 */
public class EmployeeDetails extends javax.swing.JFrame {

    /**
     * Creates new form EmployeeDetails
     */
    public EmployeeDetails() {
        initComponents();
    }

    public ArrayList<Map<String, String>> employeesList() {
        ArrayList<Map<String, String>> employees = new ArrayList<>();
        try {
            Class.forName("com.mysql.jdbc.Driver");

            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/employee", "root", "mysqlserv
eroxide");

            Statement st = conn.createStatement();
```

```
ResultSet rs = st.executeQuery("SELECT * FROM employee_details");

while(rs.next()) {
    Map<String, String> emp = new HashMap<>();
    emp.put("name", rs.getString("emp_name"));
    emp.put("id", String.valueOf(rs.getInt("emp_id")));
    emp.put("type", rs.getString("emp_type"));
    emp.put("contact_no", rs.getString("emp_contact_no"));
    emp.put("last_paid_on", rs.getString("emp_last_paid_on"));
    emp.put("address", rs.getString("emp_address"));

    System.out.println(emp.get("name"));

    employees.add(emp);
}

} catch(ClassNotFoundException | SQLException e) {
    System.out.println(e);
}

return employees;
}

public void showEmployees() {
    ArrayList<Map<String, String>> employees = employeesList();
    DefaultTableModel model = (DefaultTableModel)EmployeeTable.getModel();
    Object[] row = new Object[6];

    for (int i = 0 ; i < employees.size() ; i++) {
        row[0] = employees.get(i).get("id");
        row[1] = employees.get(i).get("name");
        row[2] = employees.get(i).get("type");
        row[3] = employees.get(i).get("contact_no");
        row[4] = employees.get(i).get("last_paid_on");
        row[5] = employees.get(i).get("address");

        model.addRow(row);
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    EmployeeTable = new javax.swing.JTable();
    jPanel1 = new javax.swing.JPanel();
    FetchEmployees = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```

EmployeeTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "ID", "Name", "Type", "Contact No", "Last Paid On", "Address"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, true, true
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane1.setViewportView(EmployeeTable);

FetchEmployees.setText("Fetch");
FetchEmployees.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        FetchEmployeesActionPerformed(evt);
    }
});

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Employee Details");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(10, 10, 10)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(FetchEmployees)
                .addGap(43, 43, 43))
            .addContainerGap(18, Short.MAX_VALUE))
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 38, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(FetchEmployees)
            .addContainerGap(18, Short.MAX_VALUE))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 600, Short.MAX_VALUE)
            .addContainerGap(18, Short.MAX_VALUE))
);

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFA
ULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 792, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 393, Short.MAX_VALUE)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SI
ZE, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void FetchEmployeesActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_FetchEmployeesActionPerformed
    showEmployees();
} // GEN-LAST:event_FetchEmployeesActionPerformed

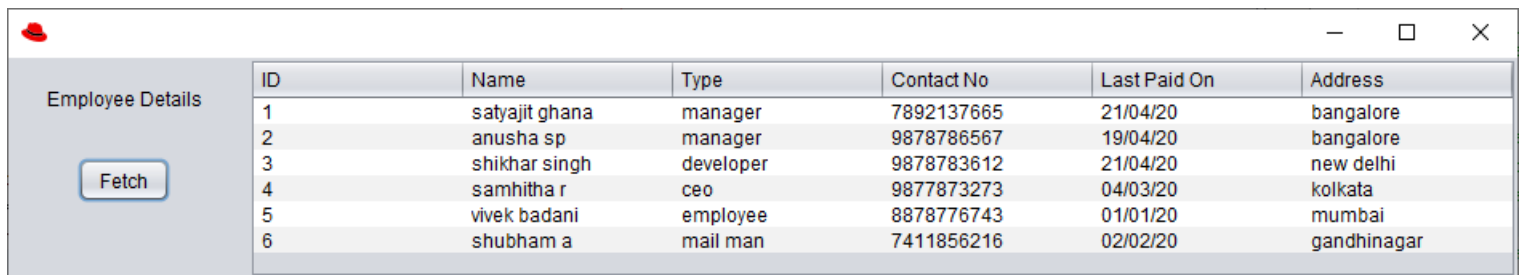
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeel
s()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(EmployeeDetails.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(EmployeeDetails.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(EmployeeDetails.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(EmployeeDetails.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    }
} //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new EmployeeDetails().setVisible(true);
        }
    });
}

```

```
}  
  
// Variables declaration - do not modify//GEN-BEGIN:variables  
private javax.swing.JTable EmployeeTable;  
private javax.swing.JButton FetchEmployees;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JScrollPane jScrollPane1;  
// End of variables declaration//GEN-END:variables  
}
```

6. Presentation of Results



ID	Name	Type	Contact No	Last Paid On	Address
1	satyajit ghana	manager	7892137665	21/04/20	bangalore
2	anusha sp	manager	9878786567	19/04/20	bangalore
3	shikhar singh	developer	9878783612	21/04/20	new delhi
4	samhitha r	ceo	9877873273	04/03/20	kolkata
5	vivek badani	employee	8878776743	01/01/20	mumbai
6	shubham a	mail man	7411856216	02/02/20	gandhinagar

Figure 1 Employee Details

When the Fetch button is clicked, the data is read from the server one-by-one and put into a ArrayList of Maps, where each index of the array stores one row of data from the database, and in each index of array we store a Map of Key, Value pair, i.e. of String, String, where the Key is the column name and the value is the value at that column in the database.

After the data is fetched, the table is populated by inserting these rows and columns into the JTable Object.

7. Conclusions

Java Swing is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckBox, JMenu, JColorChooser etc.

JDBC stands for Java Database Connectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

The java.sql and javax.sql are the primary packages for JDBC 4.0. This is the latest JDBC version at the time of writing this tutorial. It offers the main classes for interacting with your data sources.

8. Comments

1. Limitations of Experiments

- The database designed needs additional features like hashing for safely storing passwords
- The JDBC connector is not secured, and the transmission of data happened unsecurely

2. Limitations of Results

- Swing components that look like native components might not act exactly like native components
- To develop the application in swing, the individual has to be very careful with programming
- It can be slower than AWT

3. Learning happened

- We learnt how to use Java Swing to create GUI's in Java and connect it to the database using JDBC Driver.

4. Recommendations

- Be careful when using Swing Components as they behave differently in the editor and when run.

EmployeeDetails.form

```
<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9" type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <Properties>
    <Property name="defaultCloseOperation" type="int" value="3"/>
  </Properties>
  <SyntheticProperties>
    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
    <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
  </SyntheticProperties>
  <AuxValues>
    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_autoSetComponentName" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean" value="true"/>
    <AuxValue name="FormSettings_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer" value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle" type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer" value="2"/>
  </AuxValues>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="1" attributes="0">
          <Component id="jPanel1" min="-2" max="-2" attributes="0"/>
          <EmptySpace max="-2" attributes="0"/>
          <Component id="jScrollPane1" pref="792" max="32767" attributes="0"/>
        </Group>
      </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
      <Group type="103" groupAlignment="0" attributes="0">
        <Component id="jScrollPane1" pref="393" max="32767" attributes="0"/>
        <Component id="jPanel1" alignment="0" max="32767" attributes="0"/>
      </Group>
    </DimensionLayout>
  </Layout>
  <SubComponents>
    <Container class="javax.swing.JScrollPane" name="jScrollPane1">
      <AuxValues>
        <AuxValue name="autoScrollPane" type="java.lang.Boolean" value="true"/>
      </AuxValues>

      <Layout class="org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout"/>
      <SubComponents>
        <Component class="javax.swing.JTable" name="EmployeeTable">
          <Properties>
            <Property name="model" type="javax.swing.table.TableModel" editor="org.netbeans.modules.form.editors2.TableModelEditor">
              <Table columnCount="6" rowCount="0">
                <Column editable="false" title="ID" type="java.lang.Object"/>
              </Table>
            </Property>
          </Properties>
        </Component>
      </SubComponents>
    </Container>
  </SubComponents>
</Form>
```



```

        <Column editable="false" title="Name" type="java.lang.Object"/>
        <Column editable="false" title="Type" type="java.lang.Object"/>
        <Column editable="false" title="Contact No" type="java.lang.Object"/>
        <Column editable="true" title="Last Paid On" type="java.lang.Object"/>
        <Column editable="true" title="Address" type="java.lang.Object"/>
    </Table>
</Property>
<Property name="columnModel" type="javax.swing.table.TableColumnModel" editor="org.netbeans.mod
ules.form.editors2.TableColumnModelEditor">
    <TableColumnModel selectionModel="0">
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
        <Column maxWidth="-1" minWidth="-1" prefWidth="-1" resizable="true">
            <Title/>
            <Editor/>
            <Renderer/>
        </Column>
    </TableColumnModel>
</Property>
<Property name="tableHeader" type="javax.swing.table.JTableHeader" editor="org.netbeans.modules
.form.editors2.JTableHeaderEditor">
    <TableHeader reorderingAllowed="true" resizingAllowed="true"/>
</Property>
</Properties>
</Component>
</SubComponents>
</Container>
<Container class="javax.swing.JPanel" name="jPanel1">

<Layout>
    <DimensionLayout dim="0">
        <Group type="103" groupAlignment="0" attributes="0">
            <Group type="102" alignment="0" attributes="0">
                <EmptySpace max="-2" attributes="0"/>
                <Component id="jLabel1" max="32767" attributes="0"/>
            </Group>
        </Group>
    </DimensionLayout>

```

```
<EmptySpace max="-2" attributes="0"/>
</Group>
<Group type="102" alignment="1" attributes="0">
  <EmptySpace pref="43" max="32767" attributes="0"/>
  <Component id="FetchEmployees" min="-2" max="-2" attributes="0"/>
  <EmptySpace min="-2" pref="43" max="-2" attributes="0"/>
</Group>
</Group>
</DimensionLayout>
<DimensionLayout dim="1">
  <Group type="103" groupAlignment="0" attributes="0">
    <Group type="102" alignment="0" attributes="0">
      <EmptySpace max="-2" attributes="0"/>
      <Component id="jLabel1" min="-2" pref="38" max="-2" attributes="0"/>
      <EmptySpace type="separate" max="-2" attributes="0"/>
      <Component id="FetchEmployees" min="-2" max="-2" attributes="0"/>
      <EmptySpace max="32767" attributes="0"/>
    </Group>
  </Group>
</DimensionLayout>
</Layout>
<SubComponents>
  <Component class="javax.swing.JButton" name="FetchEmployees">
    <Properties>
      <Property name="text" type="java.lang.String" value="Fetch"/>
    </Properties>
    <Events>
      <EventHandler event="actionPerformed" listener="java.awt.event.ActionListener" parameters="java
      .awt.event.ActionEvent" handler="FetchEmployeesActionPerformed"/>
    </Events>
  </Component>
  <Component class="javax.swing.JLabel" name="jLabel1">
    <Properties>
      <Property name="horizontalAlignment" type="int" value="0"/>
      <Property name="text" type="java.lang.String" value="Employee Details"/>
    </Properties>
  </Component>
</SubComponents>
</Container>
</SubComponents>
</Form>
```