Simple Data driven Java application

1. Execute the registration.sql file on your local db connection.
2. Select new project from the file menu.

Graphical user interface, text, application

Description automatically generated

1. Select

Step 1. Java with maven and then Java application

Step 2: Select Java Application

Step 3: Click Next

Graphical user interface, text, application

Description automatically generated

1. Creating the project

Graphical user interface, text, application

Description automatically generated

Change the project name to mysqldemo

Change the project location to any location you want

Change the group id to your name

Finally press the finish button

At this point the IDE will create the project and the necessary folder structure for a Maven build.

Press the go button on the tool bar.

Graphical user interface, text, application

Description automatically generated

This will create build the project and run the application. At this point, the IDE will grab all the necessary resources for your application from the web. After the processing is done you should see the Build Success message in the output Screen.

Graphical user interface, text, application

Description automatically generated

1. Right click on the root of the project tree and Create a JFrame Form

Graphical user interface, application

Description automatically generated

Next, change the name to MainForm, the press Finish.

Graphical user interface, application

Description automatically generated

1. Create the controls on the form by dragging the swing controls onto the layout. All controls will have the following naming convention: <name>Control. For example if you have textbox that pertains to a first name, it will be “firstNameTextBox”. Following this convention, you should be able to intuitively get the name from code later on. You can organize the layout any way you want, but the required controls are as follows:

A screenshot of a computer

Description automatically generated with medium confidence

1. Change the title to your name then contact form as shown above.
2. Add a label and change the text to Users. Increase the font to make it bigger than the rest.
3. Add a button and name it refreshButton(under the code tab in the properties window) and change the text to Refresh under the properties tab in the properties window.

Graphical user interface, table

Description automatically generated Graphical user interface, application, table

Description automatically generated

1. Create a list and change the name to userList similar to how you changed the name for the refresh button.
2. Create all the controls as shown on the image above and rename them accordingly
3. Create a new button and rename it
4. Create a delete button and rename it
5. Create a save button and rename it
6. Press the clean and build and ensure your application builds and runs. It should prompt you to select the main class. Select the MainForm class and this should launch your application.Graphical user interface, text, application, Word

   Description automatically generated

You should see something similar to this, however you will not see the user list yet. It should be blank. You can close out of the application now.

Graphical user interface, application

Description automatically generated

1. Add the JDBC Connector
2. Add the jdbc connector to your project by opening the pom.xml file in the project files.
3. Add the following code to the file right after the </properties> end tag. Change the name to your name instead of kenji.

<name>Kenji's Contact Form</name>

<dependencies>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

</dependencies>

1. Press the clean and rebuild button.
2. You should see the dependencies updated with the mysql connector library now.

Graphical user interface, application

Description automatically generated

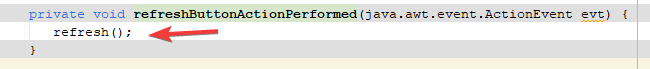
1. Create a connection to the database.

Doble click on the Refresh button

Graphical user interface, application

Description automatically generated

This should take you to the code screen. A function of refreshButtonActionPerformed() should be created for you. This is the action function for the event handler. Here we want to add a function called refresh(); as shown below (donot forget the semi-colons):



At this point the application won’t build because the refresh() function doesn’t exist yet. We have to create it. Add the refresh() function right after the action function with the following code:

public void refresh(){

try {

//Connect to the database

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/registration", "root","");

//initialize the query

String query = "select \* from reguser";

//Create a new statement

Statement sta = connection.createStatement();

//Execute the Query and assign the results to the rs object

ResultSet rs = sta.executeQuery(query);

//instantiate a new DefaultListModel

DefaultListModel listModel;

listModel = new DefaultListModel();

//Loop through the results and add the items to the lsit model

while (rs.next()) {

listModel.addElement(rs.getString("ID") + " - " + rs.getString("first\_name") + " " +

rs.getString("last\_name"));

}

//Set the model of the userList to the list Model

userList.setModel(listModel);

//Close the connection

connection.close();

//Catch any errors

} catch (Exception exception) {

exception.printStackTrace();

}

}

Clean and rebuild the code and you should see a list of users on your UI now.

Graphical user interface, application

Description automatically generated

1. The next step is to add the details of the user after you select a user. Double click the user list and this will automatically create a userListValueChanged() function.

Add the following code the function:

private void userListValueChanged(javax.swing.event.ListSelectionEvent evt) {

//Get the index of the selected item

int index = userList.getSelectedIndex();

//Log the output

System.out.println("Index Selected: " + index);

//Grab the selected value of the item.

String s = (String) userList.getSelectedValue();

//log the value

System.out.println("Value Selected: " + s);

//This function breaks apart the values by splitting the string based on "-"

String[] parts = s.split("-");

//Grab the first item - this is the id

String ID = parts[0].trim();

System.out.println("ID: " + ID);

//Call the update form funciton to update the screen

updateform(ID);

// TODO add your handling code here:

}

1. If you noticed, the updateform() function does not exist yet, so you have to create this function.

public void updateform(String id){

try {

//Establish mysql connection

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/registration", "root","");

//Initialize the query

String query = "select \* from reguser where ID="+id;

//create the statement

Statement sta = connection.createStatement();

//Set the results

ResultSet rs = sta.executeQuery(query);

//Because we are targeting a primary key, we simply set the values to the results taht are retrieved.

while (rs.next()) {

//Set the controls to the dtabase values

fname.setText(rs.getString("first\_name"));

lname.setText(rs.getString("last\_name"));

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

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

state.setText(rs.getString("state"));

zip.setText(rs.getString("zip"));

phone.setText(rs.getString("phone"));

occupation.setText(rs.getString("occupation"));

bdate.setText(rs.getString("birth\_date"));

email.setText(rs.getString("email"));

userId.setText(rs.getString("ID"));

}

//close the connection

connection.close();

} catch (Exception exception) { //catch any exceptions that may have occured

exception.printStackTrace(); //print any errors

}

}

1. Clean and Rebuild the project. You should see the details of each user every time you click a user in the user list now.

Graphical user interface, application

Description automatically generated

1. That’s it! You should be able to use this technique to do Assignment 5!