Secure Database Application in Eclipse Using JDBC

Complete all the instructions in this document and submit it as Assignment 2.

Submission Modes (choose 1 or 2):

1. Demonstration of the complete secure database application to the Professor in in class ...showing all the requirements of the prescribed graphical user design with the operations of insert, update, retrieve, search and delete fully functional as outlined in the document. This is individual presentation.
2. Fully documented and narrated video of the complete application ...showing all the requirements of the prescribed graphical user design with the operations of insert, update, retrieve, search and delete fully functional as outlined in the document. This is individual video presentation.

Assignment 2 Instructions

Using MySQL Workbench, create a database named **"mydb"** and a table named **"book"** inside it with columns

- ID
- Name
- Edition
- Price

```
create database myDB;
use myDB;
CREATE TABLE `mydb`.`book` (
  `ID` INT NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR (250) NULL,
  `Edition` INT NULL,
  `Price` INT NULL,
  PRIMARY KEY (`ID`));
```

Creating a Java Project in Eclipse

- Open Eclipse IDE and click on the Java Project under the new section of File Menu (***File>>New>>Java Project***).

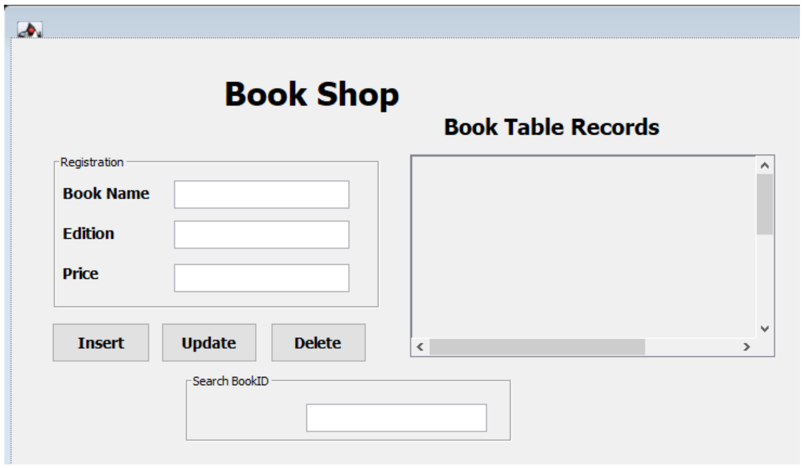
- Now give a name to your project (**aJavaCrud** as in this example) and click on "Finish".

- **Note**: WindowBuilder is required but not pre-installed in Eclipse. To install WindowBuilder for graphical design capabilities follow the below instructions:
    i) In Eclipse main Menu bar, click on help,
    ii) Click Install New Software
    iii) Click Add
    iv) Under Name, type WindowBuilder,
    v) Under Location, copy: https://download.eclipse.org/windowbuilder/1.11.0/
    vi) Click Add
    vii) Check on the checkboxes
    viii) Click next and allow it to install the available options

- After installing WindowBuilder in Eclipse, right click on your Project “src” and create a new Java class (***New>>Class>>Other>>WindowBuilder>>Swing Designer>>Application Window***) as shown below:

- Now give a name to your class (e.g., **JavaCrud**)

Change to Design view and create the following design using the WindowBuilder Palette:



Adding MySQL connector jar file in Eclipse

- In order to connect your java program with MySQL database, you need to include **MySQL JDBC driver** which is a JAR file, namely **mysql-connector-java-8.0.31.jar**. The version number in the Jar file can be different. You can download the latest version of MySQL connector from this link (**MySQL Connector Java download**)
- After visiting the link, select platform independent from the drop-down menu and then download the **Platform Independent (Architecture Independent), ZIP Archive.** Extract the zip archive and you will get the jar file.
- Next, you have to include this jar file into your program so that you will be able to connect your java program with MySQL database.
- Right-click on the project, go to the properties section then select Java Build Path and click on the Add External JARs button.
- Click on Add External Jars … and navigate to where your unzipped Jar file is and select and open the jar file. Click Apply and Close.

Adding rs2xml.jar file in Eclipse

Download or use the **rs2xml.jar** file provided for in class. Add it as an External JARs following the above steps for configuring MySQL connector jar file. Note: This Jar file is required for proper retrieval and display of database results into your Java table.

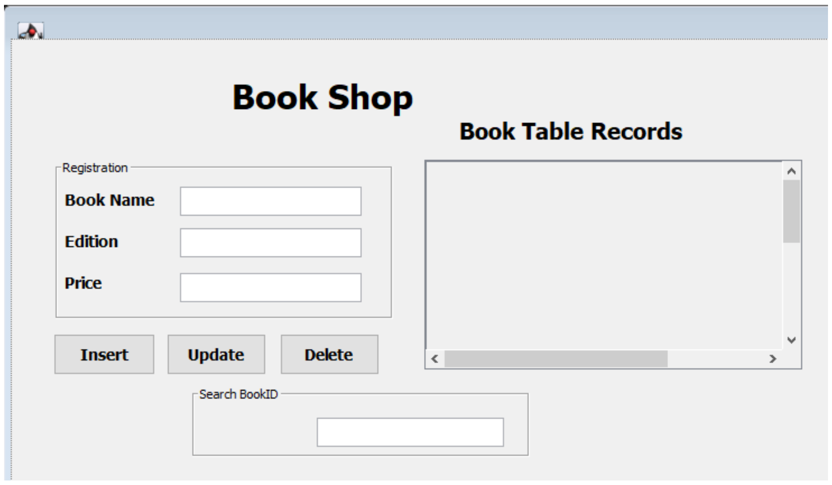
Connecting Java Program with MySQL Database

After adding the required jar files, now you are ready to connect your Java program to MySQL Database.

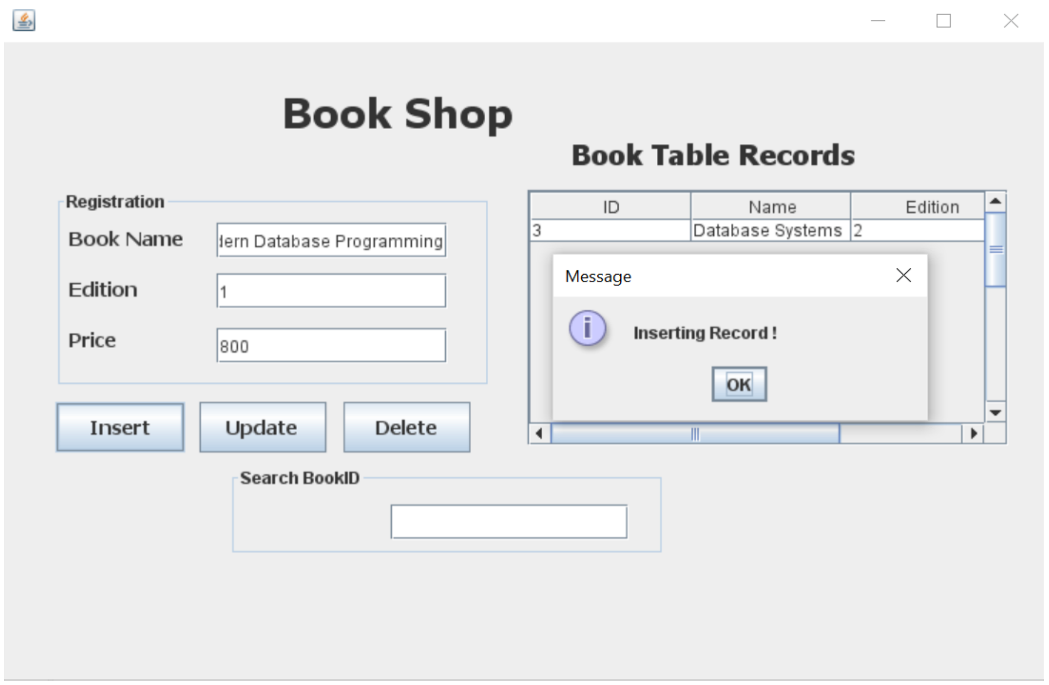
- Establish a connection using **DriverManager.getConnection(String URL)** which returns a **Connection** reference.
- In **String URL** parameter, you have to write like this:
  - **jdbc:mysql://cloudSQLhost:3306/myDB", "root", "root"** where:
  - **jdbc** is the API.
  - **mysql** is the database.
  - **cloudSQLhost/localhost** is the name of the MySQL server.
  - **3306** is the port number.
  - **myDB** is the database name. If your database name is different, then you have to replace this name with your database name.
  - The first **root** is the username of the MySQL database. It is the default username for the MySQL database.
  - The second **root** is the password that you gave while installing the MySQL database.
- **Note**:
  - The current MySQL user name and password MUST be used
  - SQL Exception might occur while connecting to the database, so you need to surround it with the **try-catch** block.

Your final application should follow the below instructions

1. The main User/Client Presentation layer design should be exactly as below:



2. Inserting a record into the book table should emulate the following graphic:



Book Shop

Book Table Records

Registration

Book Name: ɨern Database Programming

Edition: 1

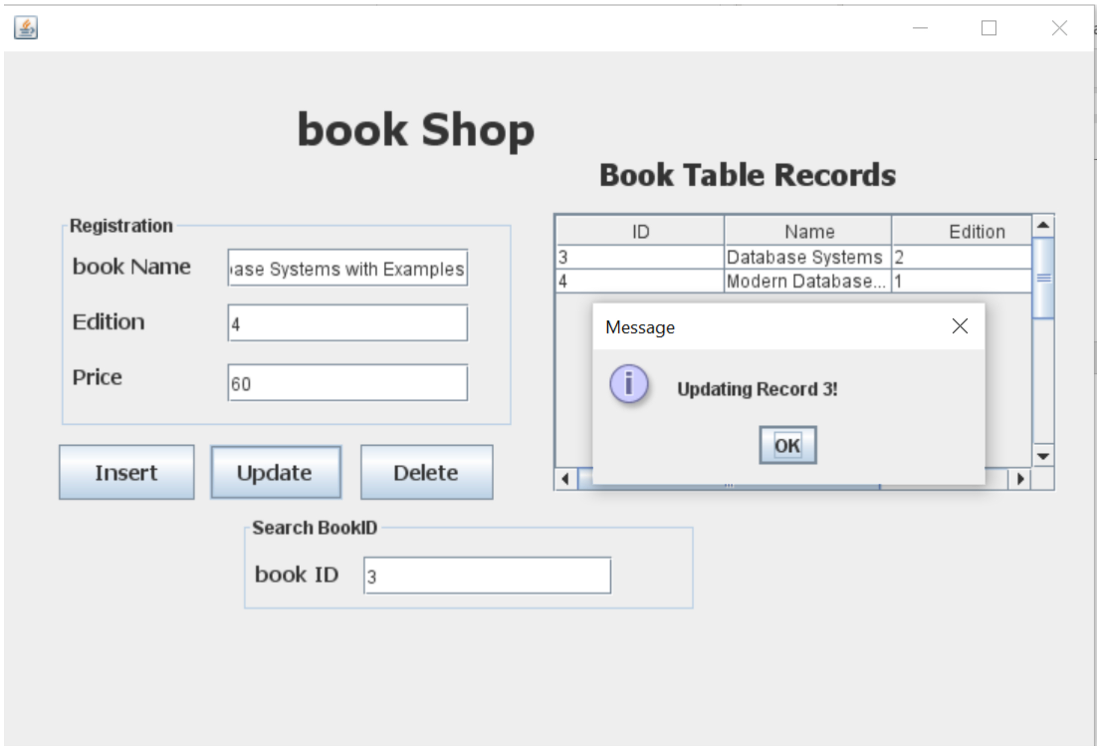
Price: 800

ID	Name	Edition
3	Database Systems	2

Message

Inserting Record !

3. Updating book table record should emulate the following graphic



book Shop

Book Table Records

Registration

book Name: ɪase Systems with Examples

Edition: 4

Price: 60

ID	Name	Edition
3	Database Systems	2
4	Modern Database...	1

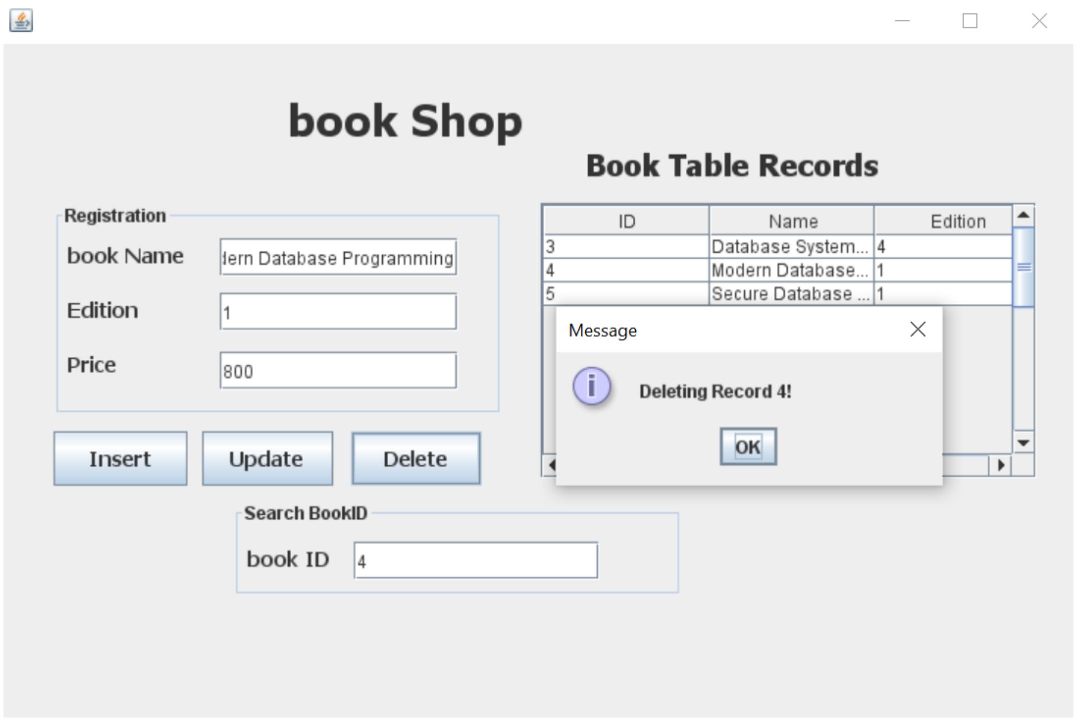
Message

Updating Record 3!

Search BookID

book ID: 3

4. Deleting a book record should emulate the following graphic



book Name: lern Database Programming
Edition: 1
Price: 800
book ID: 4

ID	Name	Edition
3	Database System...	4
4	Modern Database...	1
5	Secure Database ...	1

Deleting Record 4!

The code for above Java Application is given below:

```
package eJavaCRUD;

import java.sql.*; //MUST first be added manually

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import javax.swing.JPanel;
import javax.swing.border.TitledBorder;

import net.proteanit.sql.DbUtils;//uses rs2xml.jar file

import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JTable;
import javax.swing.JScrollPane;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

public class eJavaCrud {
	private JFrame frame;
	private JTextField txtEdition;
	private JTextField txtBName;
	private JTextField txtPrice;
	private JTable table;
	private JTextField txtBookID;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					eJavaCrud window = new eJavaCrud();
					window.frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the application.
	 */
	public eJavaCrud() {
		initialize();
		Connect();
		table_load();//should be commented at the beginning and later opened
	}

	// the main methods
	Connection con;
	PreparedStatement pst;
	ResultSet rs;

	public void Connect()
	    {
	        try {
	            Class.forName("com.mysql.cj.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost/mydb","Geo","$routa_3");
	        }
	        catch (ClassNotFoundException ex)
	        {
	          ex.printStackTrace();
	        }
	        catch (SQLException ex)
	        {
	            ex.printStackTrace();
	        }

	    }
```

```
        //for automatically populating records in the display table
          public void table_load()
            {
              try
              {
                  pst = con.prepareStatement("select * from book");
                  rs = pst.executeQuery();
                  table.setModel(DbUtils.resultSetToTableModel(rs));
              }
              catch (SQLException e)
               {
                     e.printStackTrace();
                }
            }


        /**
         * Initialize the contents of the frame.
         */
        private void initialize() {
                frame = new JFrame();
                frame.setBounds(100, 100, 747, 505);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.getContentPane().setLayout(null);

                JLabel lblNewLabel = new JLabel("book Shop");
                lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 30));
                lblNewLabel.setBounds(197, 22, 171, 59);
                frame.getContentPane().add(lblNewLabel);

                JPanel panel = new JPanel();
panel.setBorder(new TitledBorder(null, "Registration", TitledBorder.LEADING,
TitledBorder.TOP, null, null));
                panel.setBounds(38, 108, 305, 144);
                frame.getContentPane().add(panel);
                panel.setLayout(null);

                JLabel lblBName = new JLabel("book Name");
                lblBName.setFont(new Font("Tahoma", Font.BOLD, 14));
                lblBName.setBounds(10, 22, 93, 26);
                panel.add(lblBName);

                JLabel lblPrice = new JLabel("Price");
                lblPrice.setFont(new Font("Tahoma", Font.BOLD, 14));
                lblPrice.setBounds(10, 96, 93, 26);
                panel.add(lblPrice);

                JLabel lblEdition = new JLabel("Edition");
                lblEdition.setFont(new Font("Tahoma", Font.BOLD, 14));
                lblEdition.setBounds(10, 59, 93, 26);
                panel.add(lblEdition);

                txtEdition = new JTextField();
                txtEdition.setBounds(113, 61, 162, 26);
                panel.add(txtEdition);
```

```
		txtEdition.setColumns(10);

		txtBName = new JTextField();
		txtBName.setColumns(10);
		txtBName.setBounds(113, 24, 162, 26);
		panel.add(txtBName);

		txtPrice = new JTextField();
		txtPrice.setColumns(10);
		txtPrice.setBounds(113, 101, 162, 26);
		panel.add(txtPrice);

		//Inserting records ---start here
		JButton btnInsert = new JButton("Insert");
		btnInsert.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				String bname,edition,price;
				bname = txtBName.getText();
				edition = txtEdition.getText();
				price = txtPrice.getText();
				try {
pst = con.prepareStatement("insert into book(Name,Edition,Price)values(?,?,?)");
				pst.setString(1,bname);
				pst.setString(2,edition);
				pst.setString(3,price);
				pst.executeUpdate();
JOptionPane.showMessageDialog(null, "Inserting Record "+txtBookID.getText()+"!");

				table_load(); //ONLY activate after completing the table
scroll to automatically display added records

				txtBName.setText("");
				txtEdition.setText("");
				txtPrice.setText("");
				txtBName.requestFocus();
				   }

				catch (SQLException e1)
					{
				e1.printStackTrace();

			}
			}
		});
		btnInsert.setFont(new Font("Tahoma", Font.BOLD, 14));
		btnInsert.setBounds(38, 263, 91, 37);
		frame.getContentPane().add(btnInsert);

		JScrollPane scrollPane_1 = new JScrollPane();
		scrollPane_1.setBounds(369, 108, 337, 187);
		frame.getContentPane().add(scrollPane_1);

		JScrollPane scrollPane = new JScrollPane();
		scrollPane_1.setViewportView(scrollPane);
```

```java
             table = new JTable();
             scrollPane.setViewportView(table);

             JPanel panel_1 = new JPanel();
panel_1.setBorder(new TitledBorder(null, "Search BookID", TitledBorder.LEADING,
TitledBorder.TOP, null, null));
             panel_1.setBounds(160, 310, 305, 65);
             frame.getContentPane().add(panel_1);
             panel_1.setLayout(null);

//BookID to automatically refresh the registration fields after inserting Book ID in
the Search box
             txtBookID = new JTextField();
             txtBookID.addKeyListener(new KeyAdapter() {

                    public void keyReleased(KeyEvent e) {

                                  try {

                                       String id = txtBookID.getText();

pst = con.prepareStatement("select Name,Edition,Price from book where ID = ?");
                                           pst.setString(1, id);
                                           ResultSet rs = pst.executeQuery();

                                       if(rs.next()==true)
                                       {
                                           String name = rs.getString(1);
                                           String edition = rs.getString(2);
                                           String price = rs.getString(3);

                                           txtBName.setText(name);
                                           txtEdition.setText(edition);
                                           txtPrice.setText(price);

                                       }
                                       else
                                       {
                                        txtBName.setText("");
                                        txtEdition.setText("");
                                        txtPrice.setText("");
                                        txtBookID.requestFocus();
                                       }

                                   }

                                  catch (SQLException ex) {

                                   }
                    }
             });
             txtBookID.setColumns(10);
             txtBookID.setBounds(82, 28, 167, 26);
             panel_1.add(txtBookID);
```

```
                JLabel lblBookId = new JLabel("book ID");
                lblBookId.setFont(new Font("Tahoma", Font.BOLD, 14));
                lblBookId.setBounds(10, 26, 93, 26);
                panel_1.add(lblBookId);

                //Update method
                JButton btnUpdate = new JButton("Update");
                btnUpdate.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                String bname,edition,price,bid;
                                bname = txtBName.getText();
                                edition = txtEdition.getText();
                                price = txtPrice.getText();
                                bid  = txtBookID.getText();
                                try {
pst = con.prepareStatement("update book set Name=?,Edition=?,Price=? where ID =?");
                                                    pst.setString(1, bname);
                                            pst.setString(2, edition);
                                            pst.setString(3, price);
                                            pst.setString(4, bid);
                                            pst.executeUpdate();
JOptionPane.showMessageDialog(null, "Updating Record "+txtBookID.getText()+"!");
                                            table_load();

                                            txtBName.setText("");
                                            txtEdition.setText("");
                                            txtPrice.setText("");
                                            txtBookID.setText("");
                                            txtBookID.requestFocus();
                                }

                                catch (SQLException e1) {
                                e1.printStackTrace();
                                }
                        }
                });
                btnUpdate.setFont(new Font("Tahoma", Font.BOLD, 14));
                btnUpdate.setBounds(139, 263, 89, 37);
                frame.getContentPane().add(btnUpdate);

                //Delete Method
                JButton btnDelete = new JButton("Delete");
                btnDelete.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                   String bid;
                   bid = txtBookID.getText();

                    try {
                        pst = con.prepareStatement("delete from book where ID =?");

                               pst.setString(1, bid);
                               pst.executeUpdate();
JOptionPane.showMessageDialog(null, "Deleting Record "+txtBookID.getText()+"!");
                               table_load();
                        txtBName.setText("");
                        txtEdition.setText("");
                        txtPrice.setText("");
                        txtBookID.setText("");
                        txtBName.requestFocus();
                                }

                    catch (SQLException e1) {

                                e1.printStackTrace();
                        }
                    }
                });
                btnDelete.setFont(new Font("Tahoma", Font.BOLD, 14));
                btnDelete.setBounds(240, 263, 89, 37);
                frame.getContentPane().add(btnDelete);

                JLabel lblBookTableRecords = new JLabel("Book Table Records");
                lblBookTableRecords.setFont(new Font("Tahoma", Font.BOLD, 20));
                lblBookTableRecords.setBounds(400, 58, 214, 48);
                frame.getContentPane().add(lblBookTableRecords);
        }
}
```