Experiment 3

**Design GUI for Registration and Log in using Swing**

Name: Sai Harsha Vardhan AVN

Roll-no: BCSE1823

Batch: ‘A’

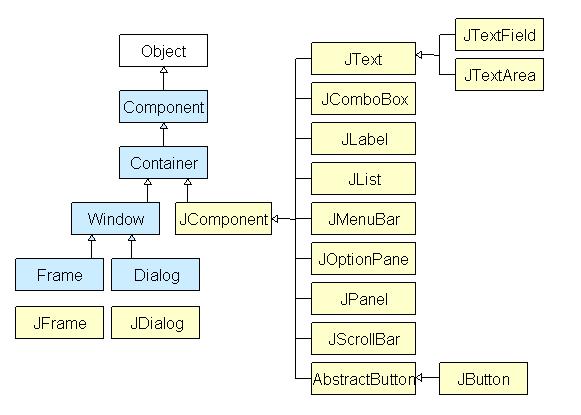
**Aim**: Design GUI for Registration and Log in using Swing

**Theory:**

**Swing in Java** is a Graphical User Interface (GUI) toolkit that includes the GUI components. Swing provides a rich set of widgets and packages to make sophisticated GUI components for Java applications. Swing is a part of Java Foundation Classes (JFC), which is an API for Java programs that provide GUI.

The Java Swing library is built on top of the Java Abstract Widget Toolkit (**AWT**), an older, platform dependent GUI toolkit. You can use the Java GUI programming components like button, textbox, etc. from the library and do not have to create the components from scratch.

### **Java Swing class Hierarchy Diagram**



**Code:**

1. Registration form Design -

import java.awt.\*;

import javax.swing.\*;

public class Form {

  public static void main(String arrgs[]) {

    JFrame f = new JFrame("employee details");

    JLabel l1 = new JLabel("name:");

    f.add(l1);

    JTextField t1 = new JTextField(30);

    f.add(t1);

    JLabel l2 = new JLabel("id:");

    f.add(l2);

    JTextField t2 = new JTextField(30);

    f.add(t2);

    JLabel l3 = new JLabel("password");

    f.add(l3);

    JPasswordField pwd = new JPasswordField(30);

    f.add(pwd);

    JLabel l4 = new JLabel("gender");

    f.add(l4);

    JRadioButton r1 = new JRadioButton("male");

    JRadioButton r2 = new JRadioButton("female");

    ButtonGroup bg = new ButtonGroup();

    bg.add(r1);

    bg.add(r2);

    f.add(r1);

    f.add(r2);

    f.setVisible(true);

    f.setSize(500, 500);

    f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

    f.setLayout(new GridLayout(10, 2));

  }

}

1. Login form Design –

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Login extends JFrame implements ActionListener{

    JPanel panel;

   JLabel user\_label, password\_label, message;

   JTextField userName\_text;

   JPasswordField password\_text;

   JButton submit, cancel;

   Login() {

      // Username Label

      user\_label = new JLabel();

      user\_label.setText("User Name :");

      userName\_text = new JTextField(30);

      // Password Label

      password\_label = new JLabel();

      password\_label.setText("Password :");

      password\_text = new JPasswordField(30);

      // Submit

      submit = new JButton("SUBMIT");

      panel = new JPanel(new GridLayout(3, 1));

      panel.add(user\_label);

      panel.add(userName\_text);

      panel.add(password\_label);

      panel.add(password\_text);

      message = new JLabel();

      panel.add(message);

      panel.add(submit);

      setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

      // Adding the listeners to components..

      submit.addActionListener(this);

      add(panel, BorderLayout.CENTER);

      setTitle("Please Login Here !");

      setSize(400,400);

      setVisible(true);

   }

   public static void main(String[] args) {

      new Login();

   }

   @Override

   public void actionPerformed(ActionEvent ae) {

      String userName = userName\_text.getText();

      String password = password\_text.getText();

      if (userName.trim().equals("admin") && password.trim().equals("admin")) {

         message.setText(" Hello " + userName + "");

      } else {

         message.setText(" Invalid user.. ");

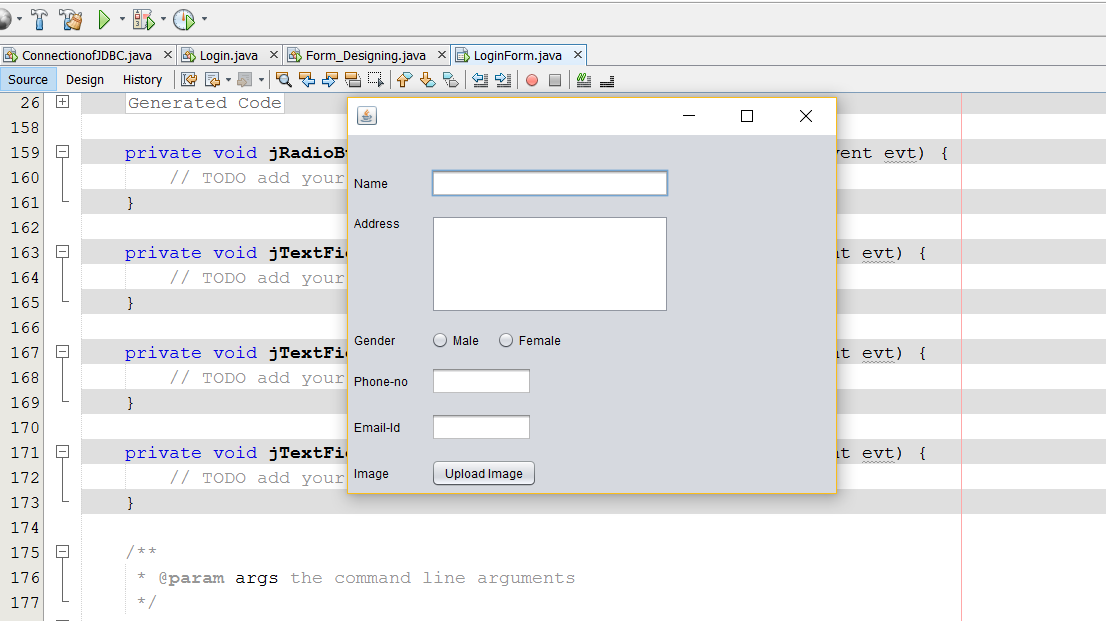
      }

   }

}

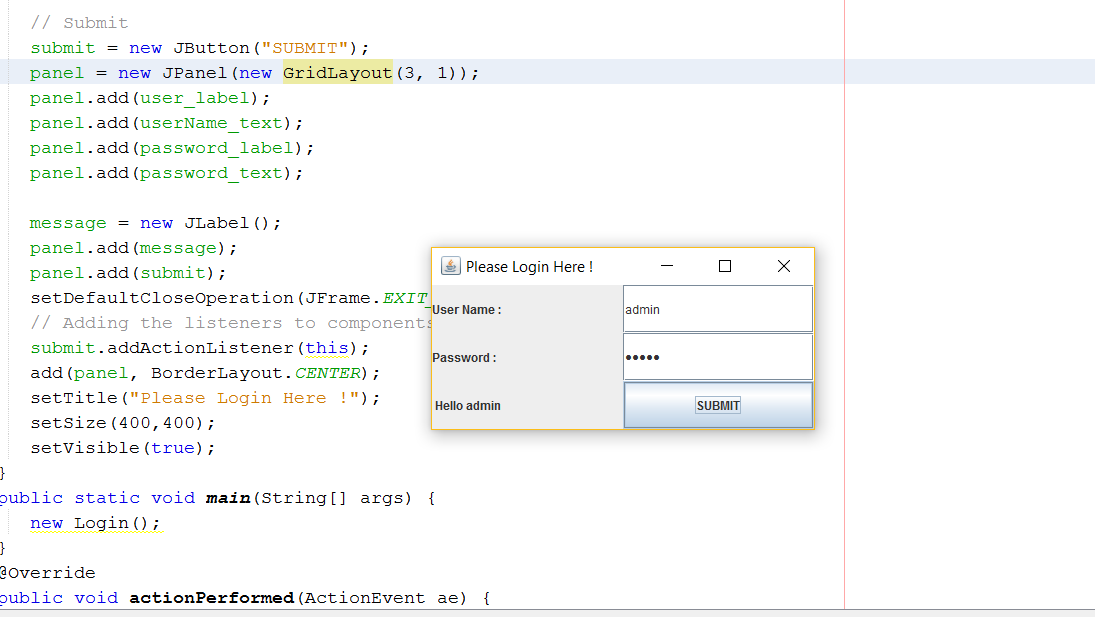
**Output:**

1. Registration form, design output.

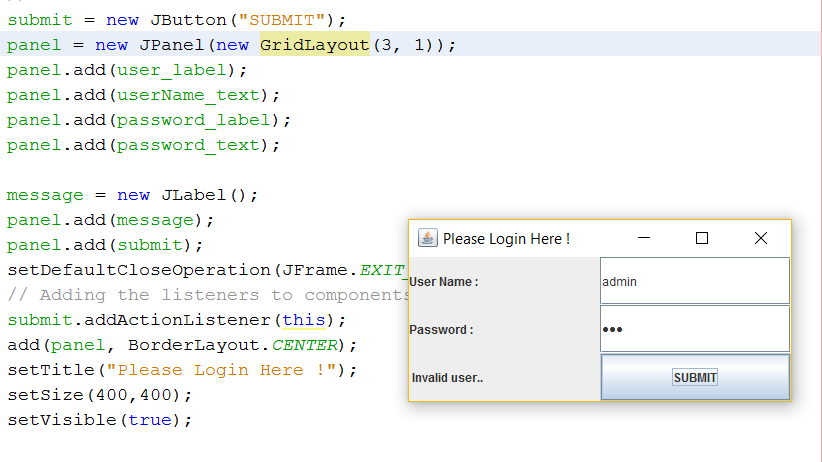


1. Login Form Design –

* If the entered details are correct



* If the entered details are incorrect

Display’s the invalid user as an message after clicking on the submit button.

**Conclusion:** Log in and Registration, designed successfully using Swing GUI.