SCD-Lab

Lab#5

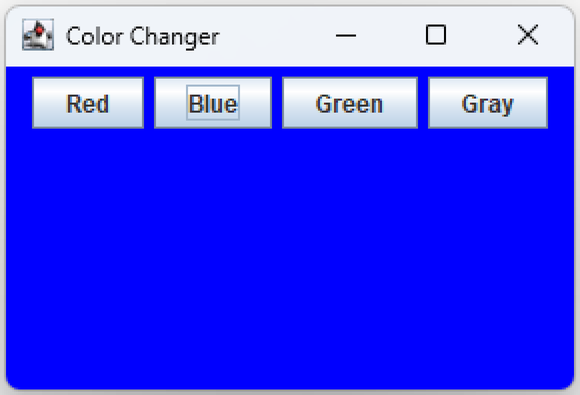
Roll.No: 22F-3639

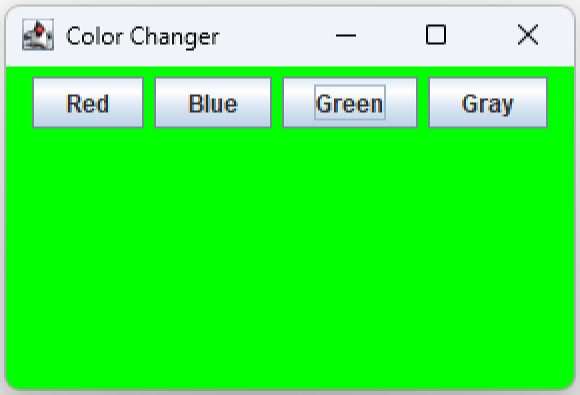
Name: Anas-Altaf

T-1

```
package t1;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
public class BackColorChanger extends JFrame implements ActionListener {
	private JButton redBtn, blueBtn, greenBtn, grayBtn;
	public BackColorChanger() {
		setTitle("Color Changer");
		setLayout(new FlowLayout());
		redBtn = new JButton("Red");
		blueBtn = new JButton("Blue");
		greenBtn = new JButton("Green");
		grayBtn = new JButton("Gray");
		redBtn.addActionListener(this);
		blueBtn.addActionListener(this);
		greenBtn.addActionListener(this);
		grayBtn.addActionListener(this);
		add(redBtn);
		add(blueBtn);
		add(greenBtn);
		add(grayBtn);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setSize(300, 200);
		setLocation(0, 0);
		setVisible(true);
	}
	@Override
	public void actionPerformed(ActionEvent e) {
		if (e.getSource() == redBtn) {
			getContentPane().setBackground(Color.RED);
		} else if (e.getSource() == blueBtn) {
			getContentPane().setBackground(Color.BLUE);
```

```
        } else if (e.getSource() == greenBtn) {
            getContentPane().setBackground(Color.GREEN);
        } else if (e.getSource() == grayBtn) {
            getContentPane().setBackground(Color.GRAY);
        }
    }
    public static void main(String[] args) {
        new BackColorChanger();
    }
}
```





T-2

```
package t2;
```

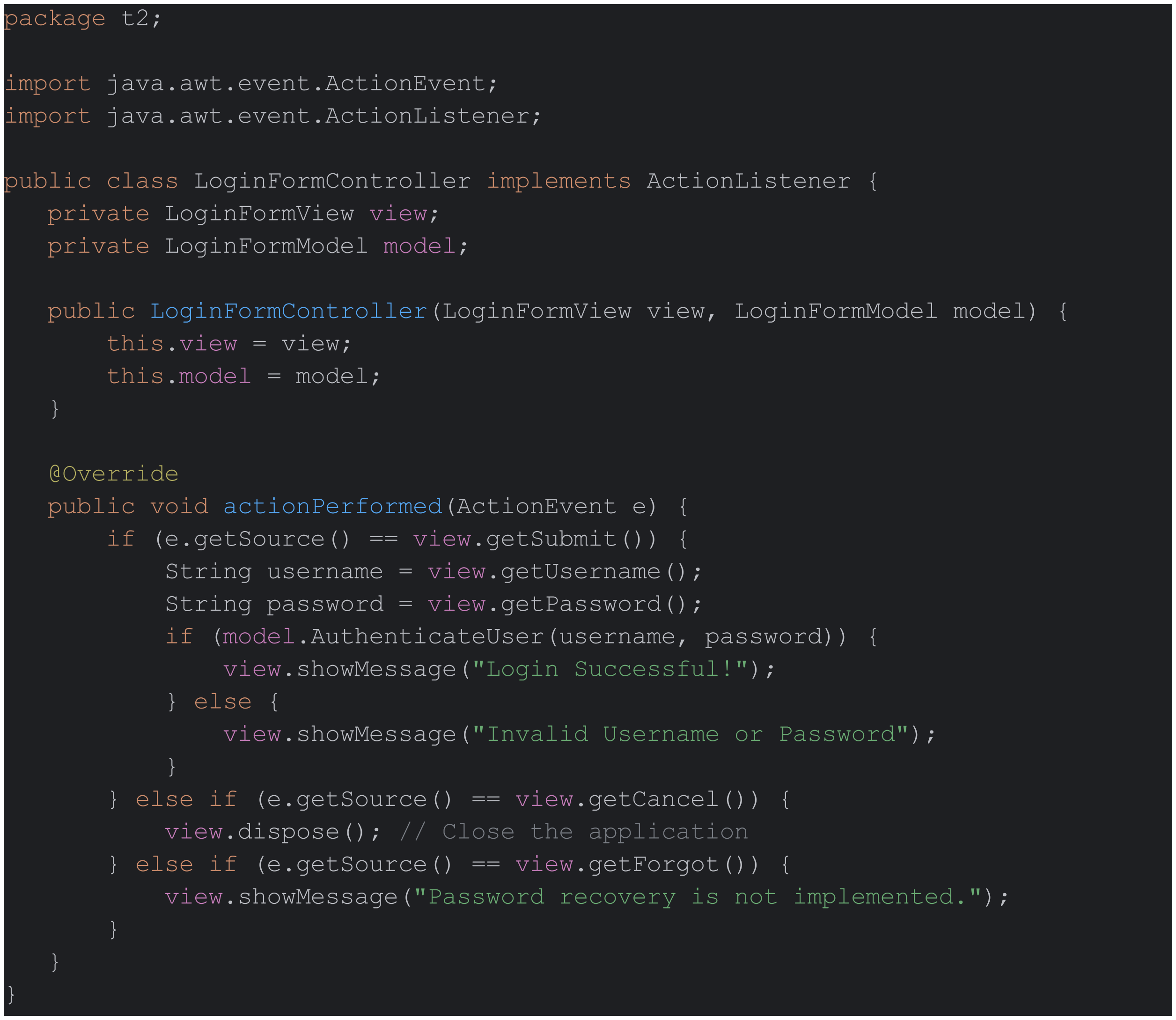
```java
import javax.swing.*;
import java.awt.*;

public class LoginFormView extends JFrame {

   private static final long serialVersionUID = 1L;
   private JButton submit, cancel, forgot;
   private JTextField username;
   private JPasswordField password;
   private JLabel userLabel, passWordLabel;

   public LoginFormView() {
       setTitle("Login Form");

       setLayout(new FlowLayout());
       submit = new JButton("Submit");
       cancel = new JButton("Cancel");
       forgot = new JButton("Forgot Password ?");

       username = new JTextField(25);
       password = new JPasswordField(25);

       userLabel = new JLabel("Username: ");
       passWordLabel = new JLabel("Password: ");

       this.add(userLabel);
       this.add(username);
       this.add(passWordLabel);
       this.add(password);
       this.add(submit);
       this.add(cancel);
       this.add(forgot);

       setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       setSize(400, 300);
       setLocationRelativeTo(null); // Center the window
       setVisible(true);
   }

   public void setController(LoginFormController controller) {
       submit.addActionListener(controller);
       cancel.addActionListener(controller);
       forgot.addActionListener(controller);
   }

   public String getUsername() {
       return username.getText();
   }
```

```
    public String getPassword() {
        return new String(password.getPassword()); // Convert to String
    }

    public void showMessage(String message) {
        JOptionPane.showMessageDialog(this, message);
    }

    public JButton getSubmit() {
        return submit;
    }

    public JButton getCancel() {
        return cancel;
    }

    public JButton getForgot() {
        return forgot;
    }
}
```

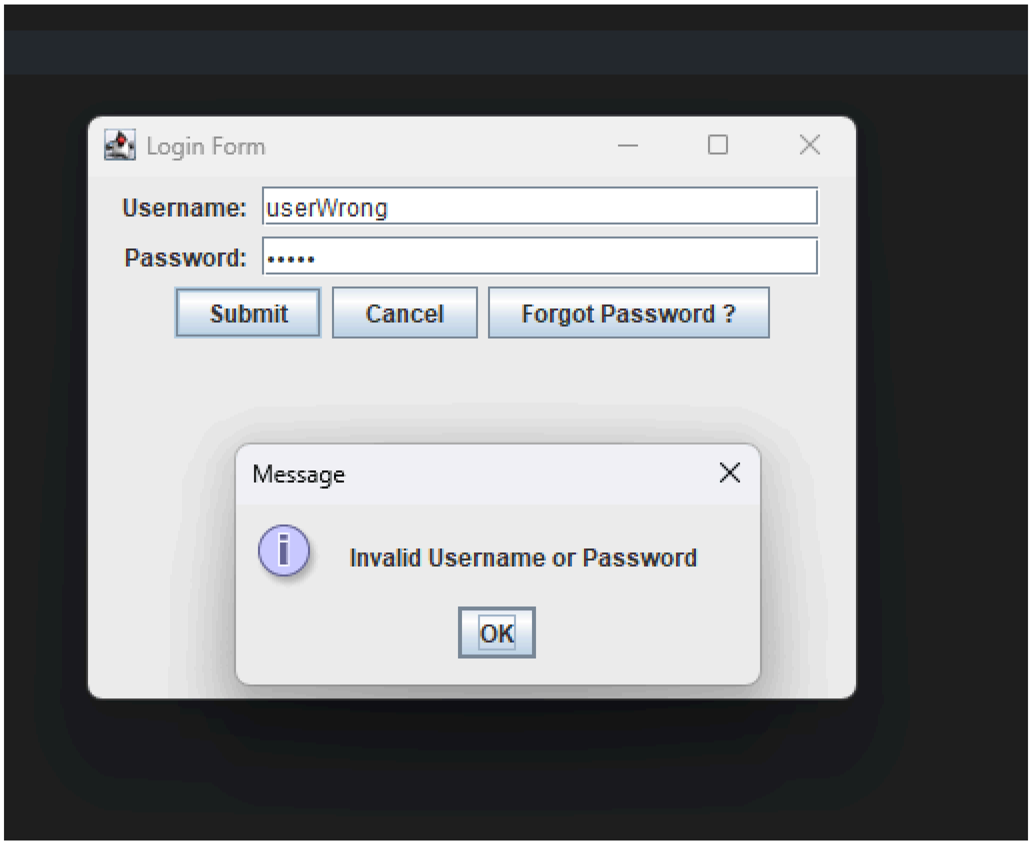
Model

```
package t2;

import java.util.HashMap;
import java.util.Objects;

public class LoginFormModel {
    HashMap<String, String> users = new HashMap<>();

    LoginFormModel() {
        users.put("user1", "pass1");
        users.put("user2", "pass2");
    }

    public boolean AuthenticateUser(String user, String inputPass) {
        if (users.containsKey(user)) {

            String passWord = users.get(user);
            return Objects.equals(passWord, inputPass);

        }
        return false;

    }
}
```

Controller



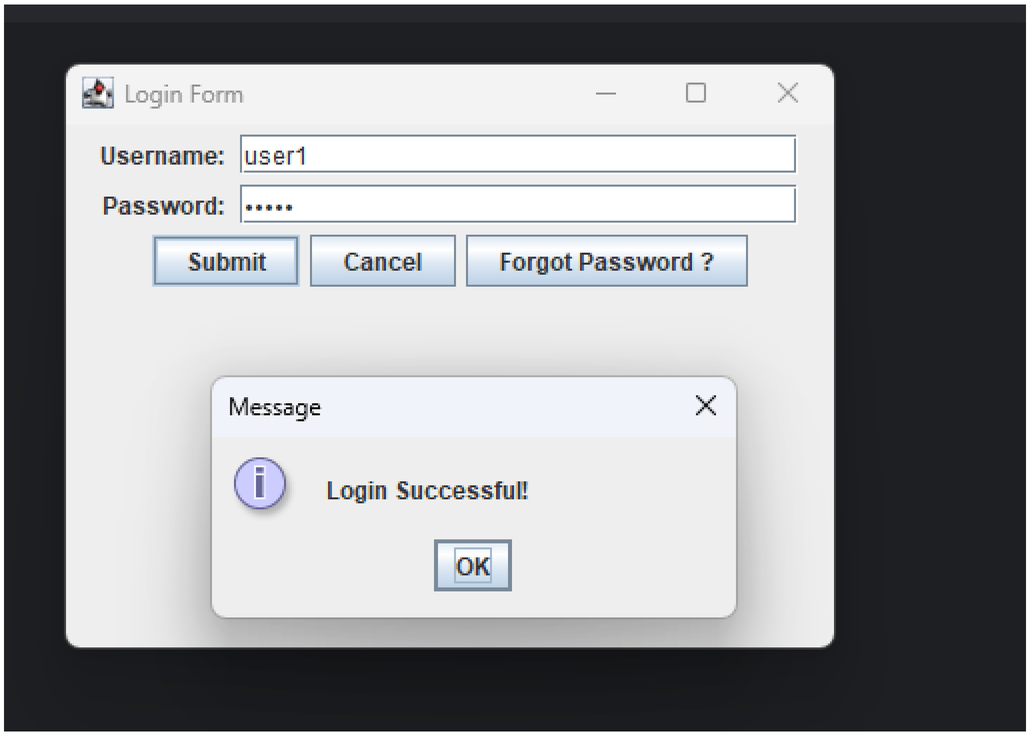
```java
package t2;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class LoginFormController implements ActionListener {
   private LoginFormView view;
   private LoginFormModel model;

   public LoginFormController(LoginFormView view, LoginFormModel model) {
       this.view = view;
       this.model = model;
   }

   @Override
   public void actionPerformed(ActionEvent e) {
       if (e.getSource() == view.getSubmit()) {
           String username = view.getUsername();
           String password = view.getPassword();
           if (model.AuthenticateUser(username, password)) {
               view.showMessage("Login Successful!");
           } else {
               view.showMessage("Invalid Username or Password");
           }
       } else if (e.getSource() == view.getCancel()) {
           view.dispose(); // Close the application
       } else if (e.getSource() == view.getForgot()) {
           view.showMessage("Password recovery is not implemented.");
       }
   }
}
```



Username: userWrong

Password: •••••

Invalid Username or Password



Username: user1

Password: •••••

Login Successful!

T-3

```
public class BMICalculator {
    private double weight;
    private double height;
    private boolean isMetric;

    public BMICalculator(boolean isMetric) {
        this.isMetric = isMetric;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

    public void setHeight(double height) {
        this.height = height;
    }

    public double calculateBMI() {
        if (isMetric) {
            return weight / Math.pow(height / 100, 2); // height in meters
        } else {
            return (weight / Math.pow(height, 2)) * 703; // height in inches
        }
    }

    public String getBMICategory(double bmi) {
        if (bmi < 18.5) return "Underweight";
        else if (bmi < 24.9) return "Normal weight";
        else if (bmi < 29.9) return "Overweight";
        else return "Obesity";
    }
}


import javax.swing.*;
import java.awt.*;

public class BMICalculatorView extends JFrame {
    private JTextField weightField = new JTextField(10);
    private JTextField heightField = new JTextField(10);
    private JTextField bmiField = new JTextField(10);
    private JRadioButton metricButton = new JRadioButton("Metric", true);
```

```java
private JRadioButton imperialButton = new JRadioButton("Imperial");
private JButton calculateButton = new JButton("Calculate BMI");

public BMICalculatorView() {
   setTitle("BMI Calculator");
   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   setSize(400, 200);
   setLayout(new GridLayout(5, 2));

   ButtonGroup group = new ButtonGroup();
   group.add(metricButton);
   group.add(imperialButton);

   add(new JLabel("Weight (kg/lbs):"));
   add(weightField);
   add(new JLabel("Height (cm/inches):"));
   add(heightField);
   add(metricButton);
   add(imperialButton);
   add(calculateButton);
   add(new JLabel("BMI:"));
   add(bmiField);

   bmiField.setEditable(false);
}

public String getWeight() {
   return weightField.getText();
}

public String getHeight() {
   return heightField.getText();
}

public boolean isMetric() {
   return metricButton.isSelected();
}

public void setBMI(String bmi) {
   bmiField.setText(bmi);
}

public void addCalculateListener(ActionListener listenForCalcButton) {
   calculateButton.addActionListener(listenForCalcButton);
```

```
    }
}

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class BMICalculatorController {
    private BMICalculator model;
    private BMICalculatorView view;

    public BMICalculatorController(BMICalculator model, BMICalculatorView view) {
        this.model = model;
        this.view = view;

        this.view.addCalculateListener(new CalculateListener());
    }

    class CalculateListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            double weight = Double.parseDouble(view.getWeight());
            double height = Double.parseDouble(view.getHeight());

            model.setWeight(weight);
            model.setHeight(height);

            double bmi = model.calculateBMI();
            String bmiCategory = model.getBMICategory(bmi);

            view.setBMI(String.format("%.2f (%s)", bmi, bmiCategory));
        }
    }
}

public class BMICalculatorApp {
    public static void main(String[] args) {
        BMICalculator model = new BMICalculator(true);
        BMICalculatorView view = new BMICalculatorView();
        BMICalculatorController controller = new BMICalculatorController(model, view);

        view.setVisible(true);
    }
}
```