Advanced Java Semester 2
JD522
Formative 1

Jonathan Van Eyssen 20231438

Main Window Code

```
import java.util.ArrayList;



public class mainFrame extends javax.swing.JFrame {
   ArrayList Users = new ArrayList();
   /**
    * Creates new form mainFrame
    */
   public mainFrame() {
      initComponents();
   }

   /**
    * 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">
   private void initComponents() {

      jLabel1 = new javax.swing.JLabel();
```

```
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
regUsername = new javax.swing.JTextField();
regPassword = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
loginUsername = new javax.swing.JTextField();
regFirstName = new javax.swing.JTextField();
regLastName = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
loginBtn = new javax.swing.JButton();
regBtn = new javax.swing.JButton();
resultLab = new javax.swing.JLabel();
loginPassword = new javax.swing.JPasswordField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Register/Login");
setMaximumSize(new java.awt.Dimension(840, 700));
setMinimumSize(new java.awt.Dimension(840, 700));
setResizable(false);
addWindowListener(new java.awt.event.WindowAdapter() {
   public void windowOpened(java.awt.event.WindowEvent evt) {
      formWindowOpened(evt);
   }
});

jLabel1.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
```

```
jLabel1.setText("Registration");

jLabel2.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
jLabel2.setText("Login");

jLabel3.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jLabel3.setText("Username");

jLabel4.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jLabel4.setText("Password");

jLabel5.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jLabel5.setText("First Name");

jLabel6.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jLabel6.setText("Last Name");

regUsername.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        regUsernameMouseEntered(evt);
    }
});

regPassword.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        regPasswordMouseEntered(evt);
    }
});
regPassword.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        regPasswordActionPerformed(evt);
    }
});

jLabel7.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jLabel7.setText("Username");

jLabel8.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jLabel8.setText("Password");

jSeparator1.setOrientation(javax.swing.SwingConstants.VERTICAL);

loginBtn.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
loginBtn.setText("Login");
loginBtn.setOpaque(true);
loginBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginBtnActionPerformed(evt);
    }
});

regBtn.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
regBtn.setText("Register");
regBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        regBtnActionPerformed(evt);
    }
});

resultLab.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
```

```
    resultLab.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    resultLab.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

    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(67, 67, 67)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel1)
          .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(layout.createSequentialGroup()
                .addGap(79, 79, 79)
                .addComponent(regBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addComponent(regPassword, javax.swing.GroupLayout.PREFERRED_SIZE, 288,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(regUsername, javax.swing.GroupLayout.PREFERRED_SIZE, 288,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel5)
              .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(regLastName, javax.swing.GroupLayout.PREFERRED_SIZE, 288,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(regFirstName, javax.swing.GroupLayout.PREFERRED_SIZE, 288,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

              .addComponent(loginPassword, javax.swing.GroupLayout.PREFERRED_SIZE, 288,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addComponent(loginUsername, javax.swing.GroupLayout.PREFERRED_SIZE, 288,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addComponent(jLabel2)

              .addGroup(layout.createSequentialGroup()

                .addGap(86, 86, 86)

                .addComponent(loginBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)))))

          .addGap(62, 62, 62))

        .addGroup(layout.createSequentialGroup()

          .addGap(28, 28, 28)

          .addComponent(resultLab, javax.swing.GroupLayout.PREFERRED_SIZE, 786,
javax.swing.GroupLayout.PREFERRED_SIZE)

          .addContainerGap(26, Short.MAX_VALUE))

    );

    layout.setVerticalGroup(

      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

      .addGroup(layout.createSequentialGroup()

        .addGap(26, 26, 26)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

          .addComponent(jLabel2)
```

```
            .addComponent(jLabel1))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel3)
                        .addGap(18, 18, 18)
                        .addComponent(regUsername, javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel4)
                        .addGap(18, 18, 18)
                        .addComponent(regPassword, javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel5)
                        .addGap(18, 18, 18)
                        .addComponent(regFirstName, javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel6)
                        .addGap(18, 18, 18)
                        .addComponent(regLastName, javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel7)
                        .addGap(18, 18, 18)
                        .addComponent(loginUsername, javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
```

```
                .addComponent(jLabel8)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(loginPassword, javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(52, 52, 52)
                .addComponent(loginBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(30, 30, 30)
            .addComponent(regBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE))
          .addComponent(jSeparator1))
        .addGap(18, 18, 18)
        .addComponent(resultLab, javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(25, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void formWindowOpened(java.awt.event.WindowEvent evt) {
//This method adds the text to the label at the bottom of the window as direction to the user.
    resultLab.setText("Please Register or Login above.");

  }

  private void regBtnActionPerformed(java.awt.event.ActionEvent evt) {
    //This is to register the user. It validates the input through the methods in the class
    String regUser = regUsername.getText();
    boolean checkedUser = formativeLogin.correctUserName(regUser);
    String regPass = regPassword.getText();
```

```
    boolean Passcheck = formativeLogin.meetPasswordComplexity(regPass);
    String regUfirstName = regFirstName.getText();
    boolean checkFirst = formativeLogin.firstNotempty(regUfirstName);
    String regUlastName = regLastName.getText();
    boolean checkLast = formativeLogin.lastNotempty(regUlastName);
    //this checks if the user registration is correct and returns the relevant message.
    ArrayList list = formativeLogin.regUser(checkedUser, Passcheck, checkFirst, checkLast);
    String message = (String)list.get(1);
    boolean passed = (boolean)list.get(0);
    if (passed == true){
      resultLab.setText(message);
    formativeLogin UserNew = new formativeLogin(regUser,regPass,regUfirstName,regUlastName);
    Users.add(UserNew);
    regUsername.setText("");
    regPassword.setText("");
    regFirstName.setText("");
    regLastName.setText("");
    } else {
      resultLab.setText(message);
    }



}

private void loginBtnActionPerformed(java.awt.event.ActionEvent evt) {
    //this is used to check with the login details against the saved data.
    String LogUserName = loginUsername.getText();
    String LogPassword = new String(loginPassword.getPassword());
    String message = formativeLogin.login(Users, LogUserName, LogPassword);
```

```
        resultLab.setText(message);
        loginUsername.setText("");
        loginPassword.setText("");
    }

    private void regPasswordActionPerformed(java.awt.event.ActionEvent evt) {
        //This is for future use.
    }

    private void regPasswordMouseEntered(java.awt.event.MouseEvent evt) {
        //This is the text that appears to tell the user the password requirements in a tooltip
        regPassword.setToolTipText("<HTML>Password Requirements: <BR>•At least 8 characters long <BR>•Contain at least one capital letter <BR>•Contain a number <BR>•Contain a special character</HTML>");
    }

    private void regUsernameMouseEntered(java.awt.event.MouseEvent evt) {
        regUsername.setToolTipText("<HTML>Username Requirements: <BR>Must contain a #. <BR>Maximum 8 characters.");
        //this is a tooltip for the username requirements.
    }

    /**
     * @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.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(mainFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(mainFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(mainFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(mainFrame.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 mainFrame().setVisible(true);
```

```
        }
      });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JButton loginBtn;
    private javax.swing.JPasswordField loginPassword;
    private javax.swing.JTextField loginUsername;
    private javax.swing.JButton regBtn;
    private javax.swing.JTextField regFirstName;
    private javax.swing.JTextField regLastName;
    private javax.swing.JTextField regPassword;
    private javax.swing.JTextField regUsername;
    private javax.swing.JLabel resultLab;
    // End of variables declaration
}

//VH/FH
```

Formative Login Class Code

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.util.ArrayList;

public class formativeLogin {

  private String Username;
  private String Password;
  private String FirstName;
  private String LastName;

  public formativeLogin(String Username, String Password, String FirstName, String LastName) {
    this.Username = Username;
    this.Password = Password;
    this.FirstName = FirstName;
    this.LastName = LastName;
  }
  //these are getters and setters for the formativeLogin object
  public String getUsername() {
    return Username;
  }

  public void setUsername(String Username) {
    this.Username = Username;
  }

  public String getPassword() {
```

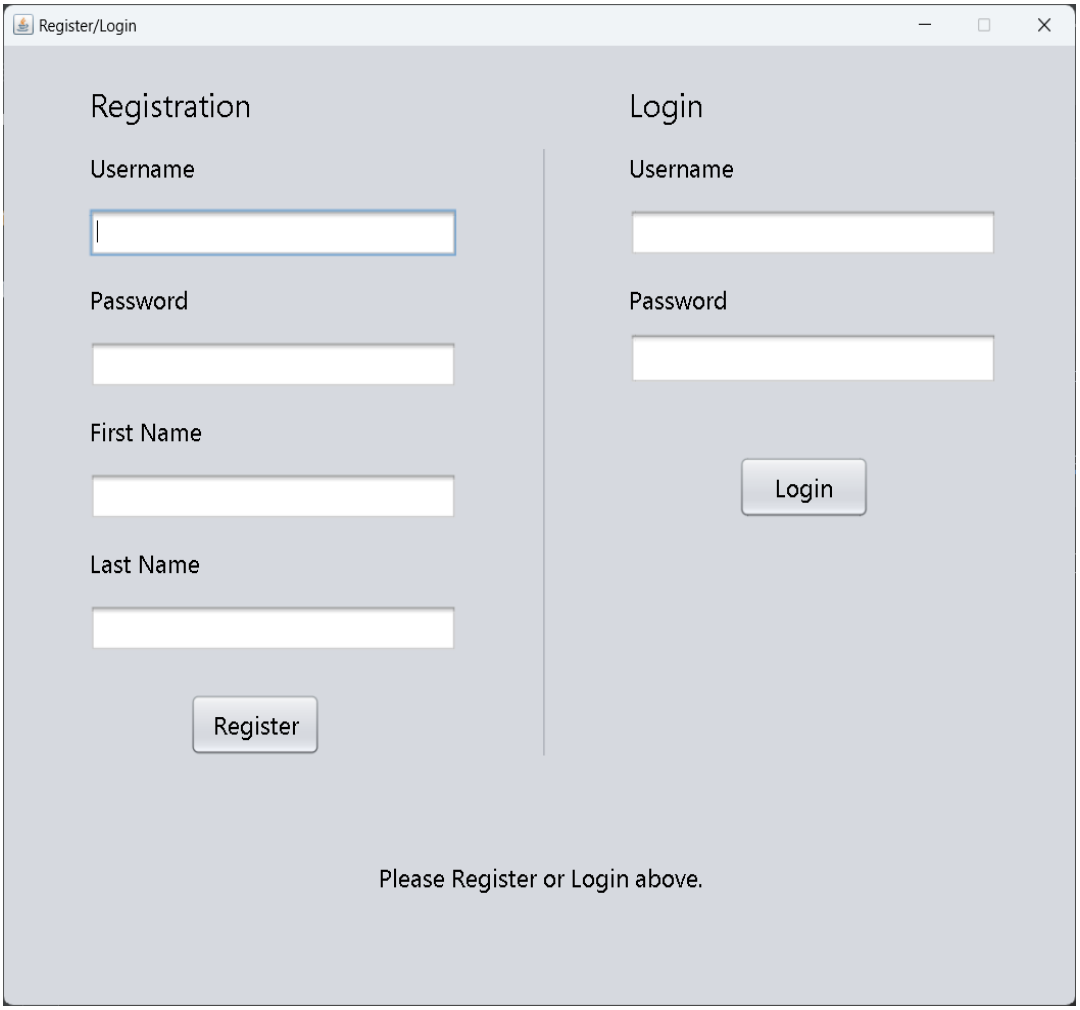
```
    return Password;
  }

  public void setPassword(String Password) {
    this.Password = Password;
  }

  public String getFirstname() {
    return FirstName;
  }

  public void setFirstname(String Firstname) {
    this.FirstName = Firstname;
  }

  public String getLastname() {
    return LastName;
  }

  public void setLasrname(String Lastname) {
    this.LastName = Lastname;
  }
  //method used to check if the username meets the requirements
  public static boolean correctUserName(String Username) {

    return Username.contains("#") && Username.length() < 8;
  }
  //this is to check if the user can be registered or not.
  public static ArrayList regUser(boolean checkedUser, boolean Passcheck, boolean checkFirst,
boolean checkLast){
```

```
    ArrayList list = new ArrayList();
    if (checkedUser == true && Passcheck == true && checkFirst == true && checkLast == true){
        boolean validUser = true;
        list.add(validUser);
        String message = "<HTML>Username Accepted, Proceed<BR>Password
Accepted<BR>Registration successful!";
        list.add(message);
        return list;
    } else if (checkedUser == false){
        boolean validUser = false;
        list.add(validUser);
        String message = "<HTML>Username does not meet the criteria, <BR>please ensure that your
username contains an pound sign <BR>and is no more than 8 characters in length.";
        list.add(message);
        return list;
    } else if (Passcheck == false){
        boolean validUser = false;
        list.add(validUser);
        String message = "Password not Accepted, please check that you have met all the criteria
required.";
        list.add(message);
        return list;
    } else if (checkFirst == false){
        boolean validUser = false;
        list.add(validUser);
        String message = "First Name cannot be empty!";
        list.add(message);
        return list;
    } else if (checkLast == false){
        boolean validUser = false;
        list.add(validUser);
```

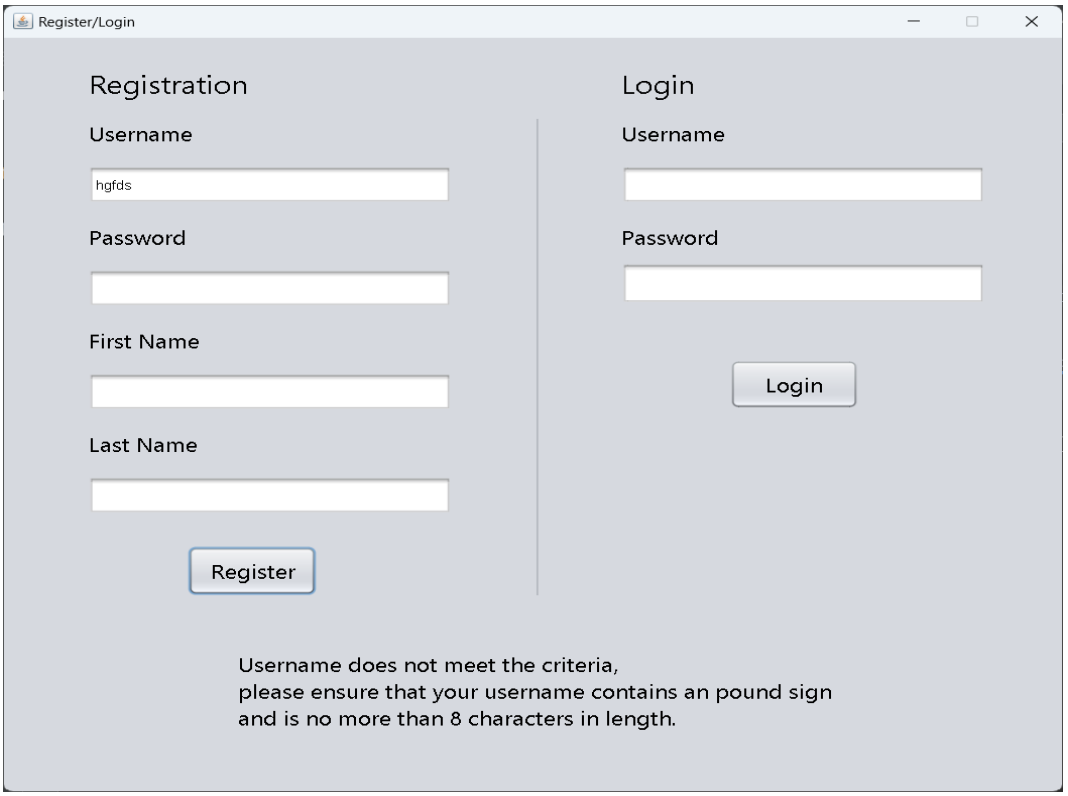
```
        String message = "Last Name cannot be empty!";
        list.add(message);
    } else {
        boolean validUser = false;
        list.add(validUser);
        String message = "Registration failed";
        list.add(message);
        return list;
    }
        boolean validUser = false;
        list.add(validUser);
        String message = "Registration failed";
        list.add(message);
        return list;
}
//this is used to check if the password requirements are met.
public static boolean meetPasswordComplexity(String regPass) {
    Pattern pat = Pattern.compile("^(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-z\\d@$!%*?&]{8,}$");
    Matcher mat = pat.matcher(regPass);
    boolean passCorrect = mat.find();
    return passCorrect == true;
}
//checks if the firstName is not empty
public static boolean firstNotempty(String FirstName) {
    return !FirstName.isEmpty();
}
//checks if the last Name is not empty
public static boolean lastNotempty(String LastName) {
    return !LastName.isEmpty();
}
```

```
   //Here is where the checking if the login details are checked if they are accurate.
   public static String login(ArrayList<formativeLogin> Users, String logUsername, String logPassword)
{


      for (formativeLogin user : Users) {
         if (logUsername.equals(user.getUsername())) {
            if (logPassword.equals(user.getPassword())) {
               String message = returnLogin(user);
               return message;
            } else {
               return "Incorrect password, try again";
            }


         } else {


            continue;
         }
      }
      return "Incorrect credentials have been supplied, try again";
   }
   //this method returns the successful message when the user has successfully logged in.
   public static String returnLogin(formativeLogin user) {
      return "<HTML>Congratulations " + user.FirstName + " " + user.LastName + ", <BR>You have made
it to the second year. <BR>Wishing you all the best!!</HTML>";
   }
}
//VH/FH
```

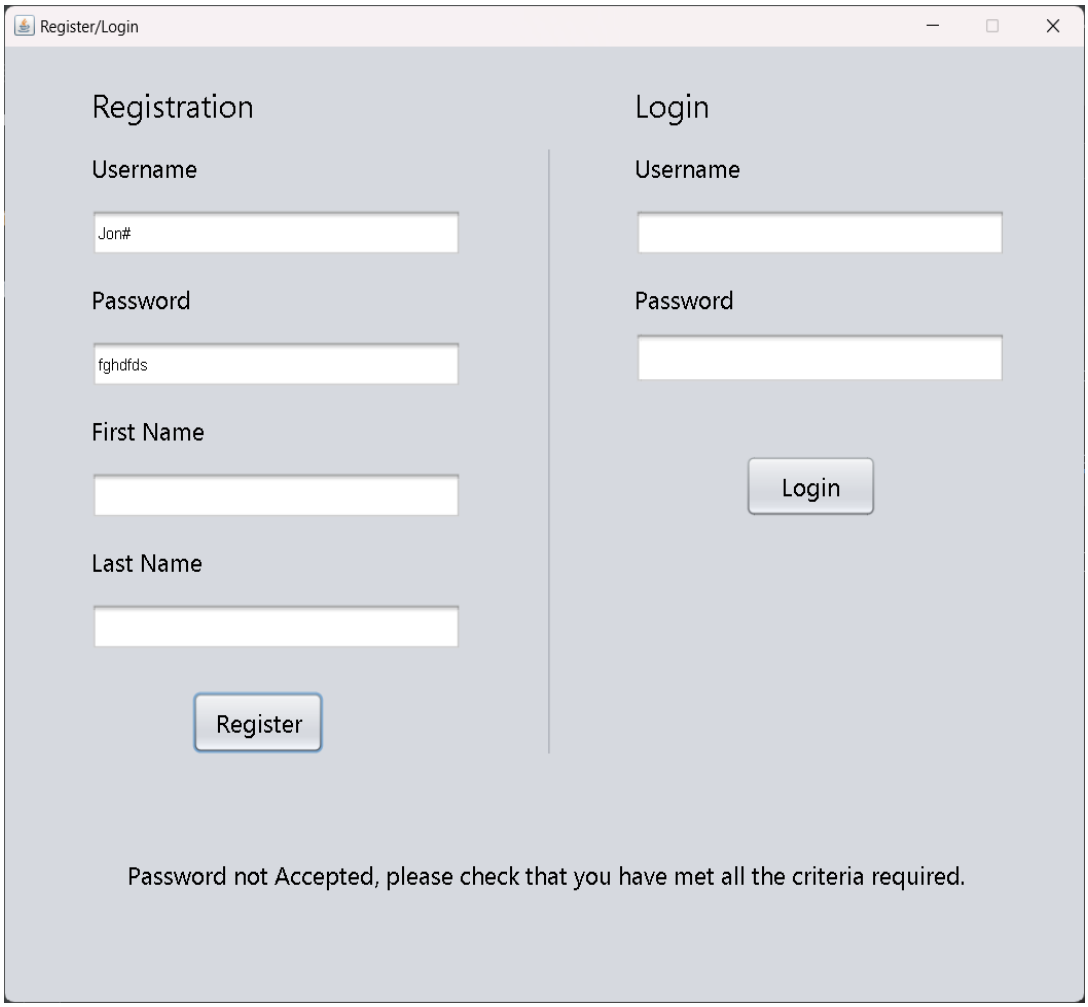
Screenshots of Output



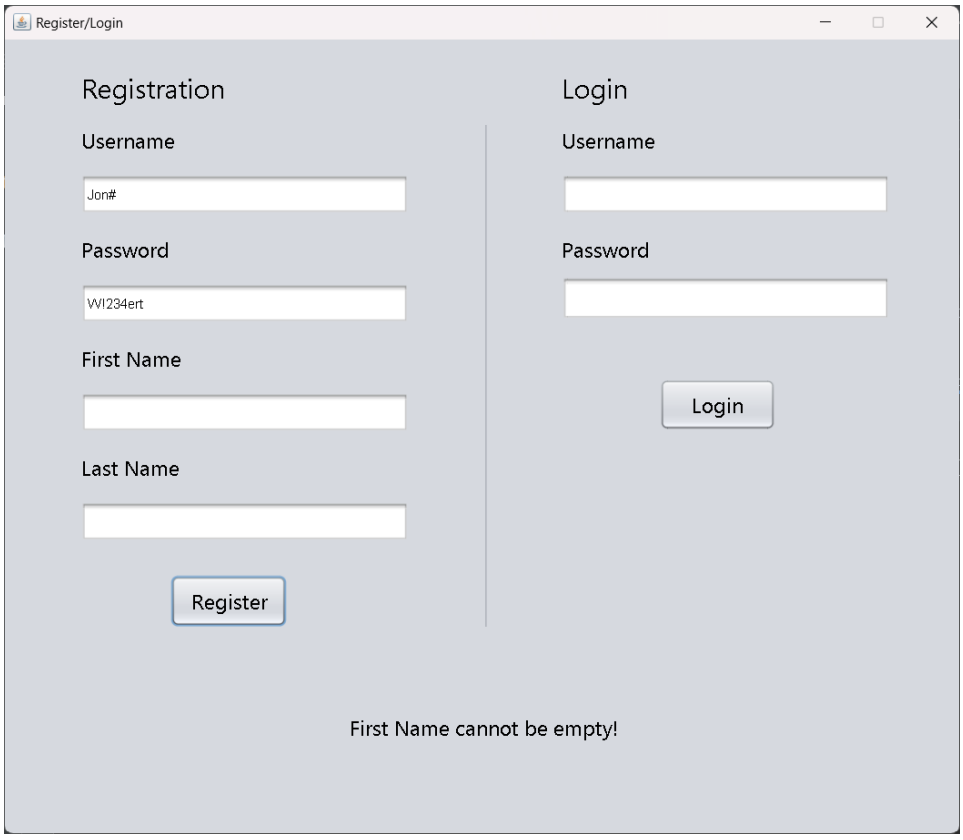
Startup with context for user on what to do



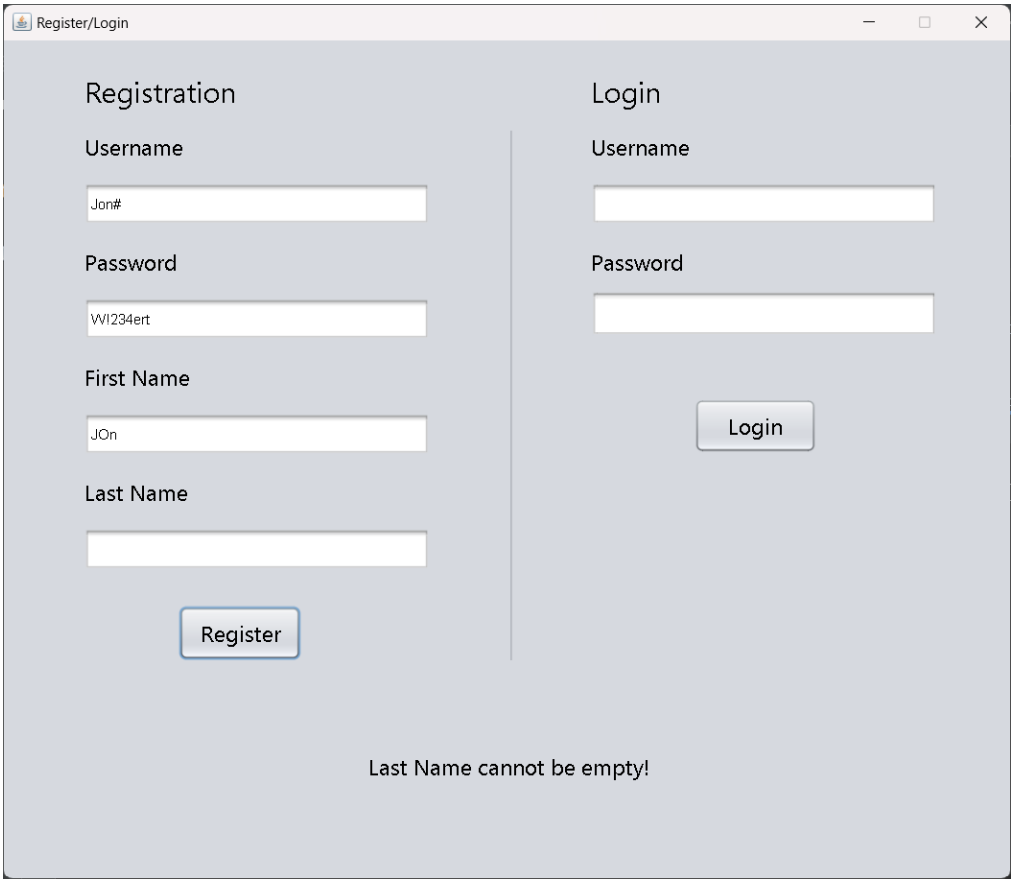
Error message for Registration Username



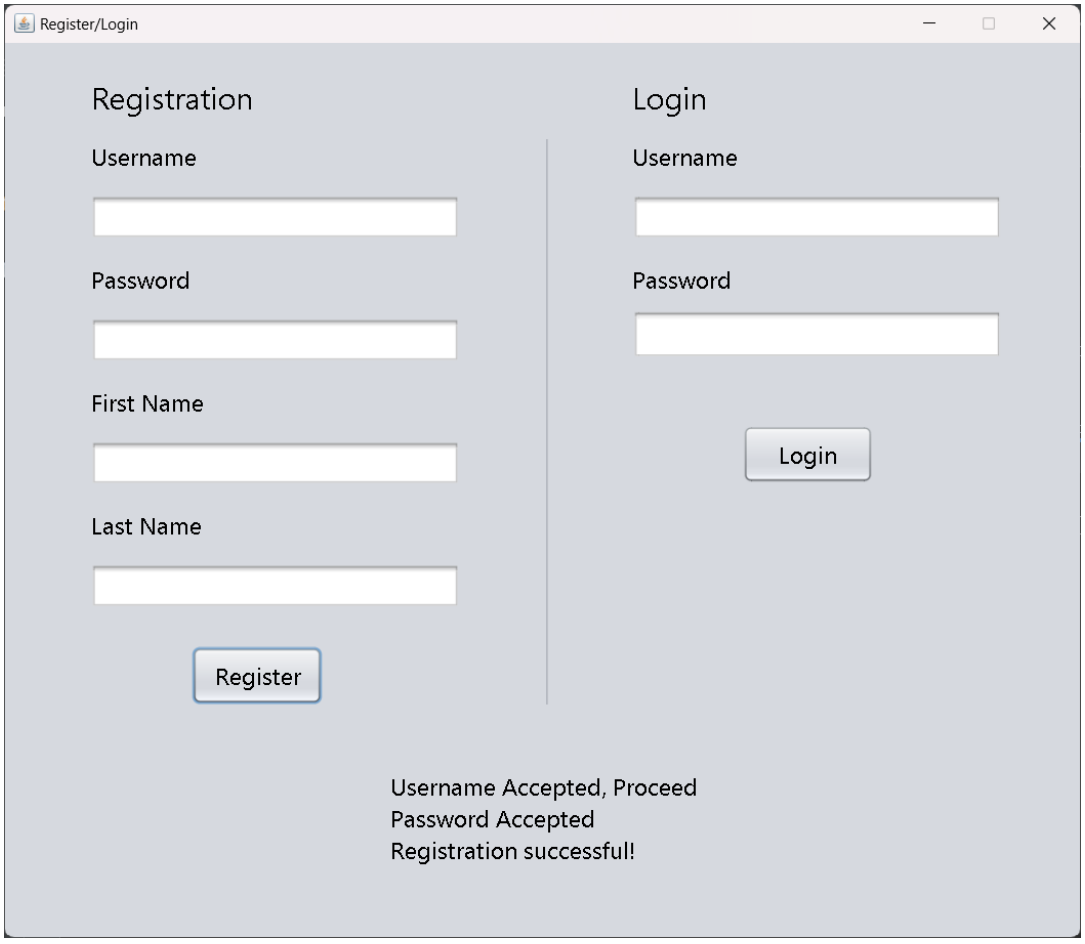
Error message for Registration Password



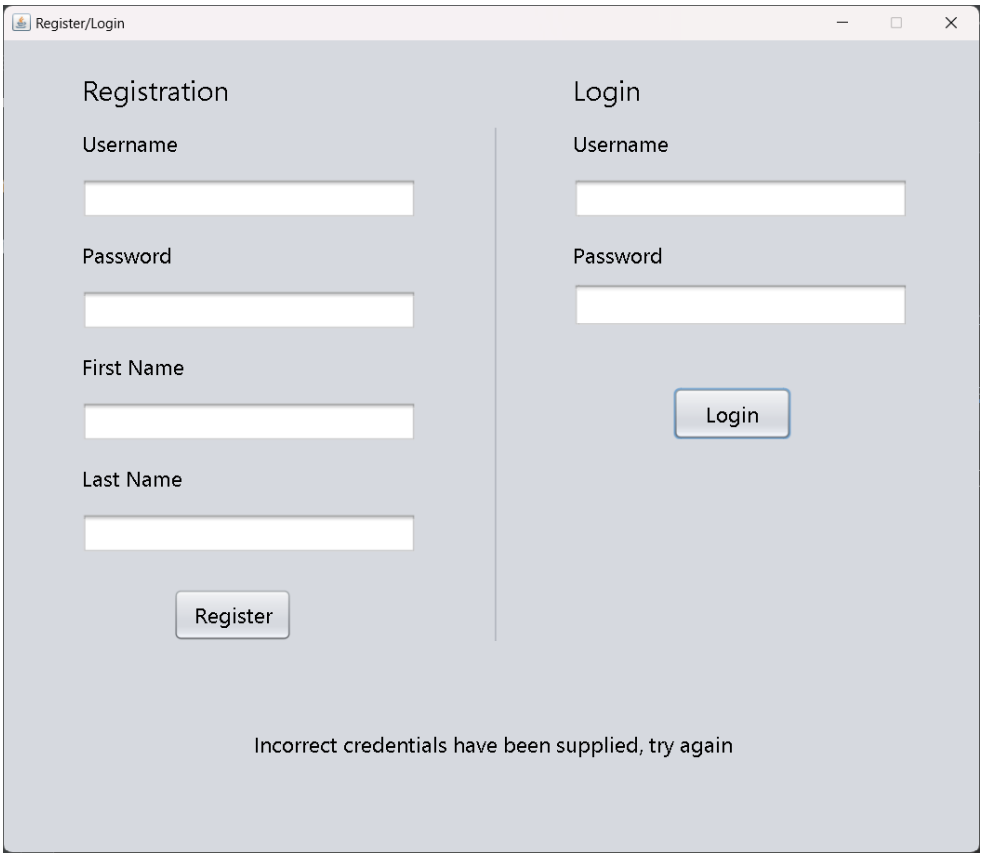
Error message for Empty First Name



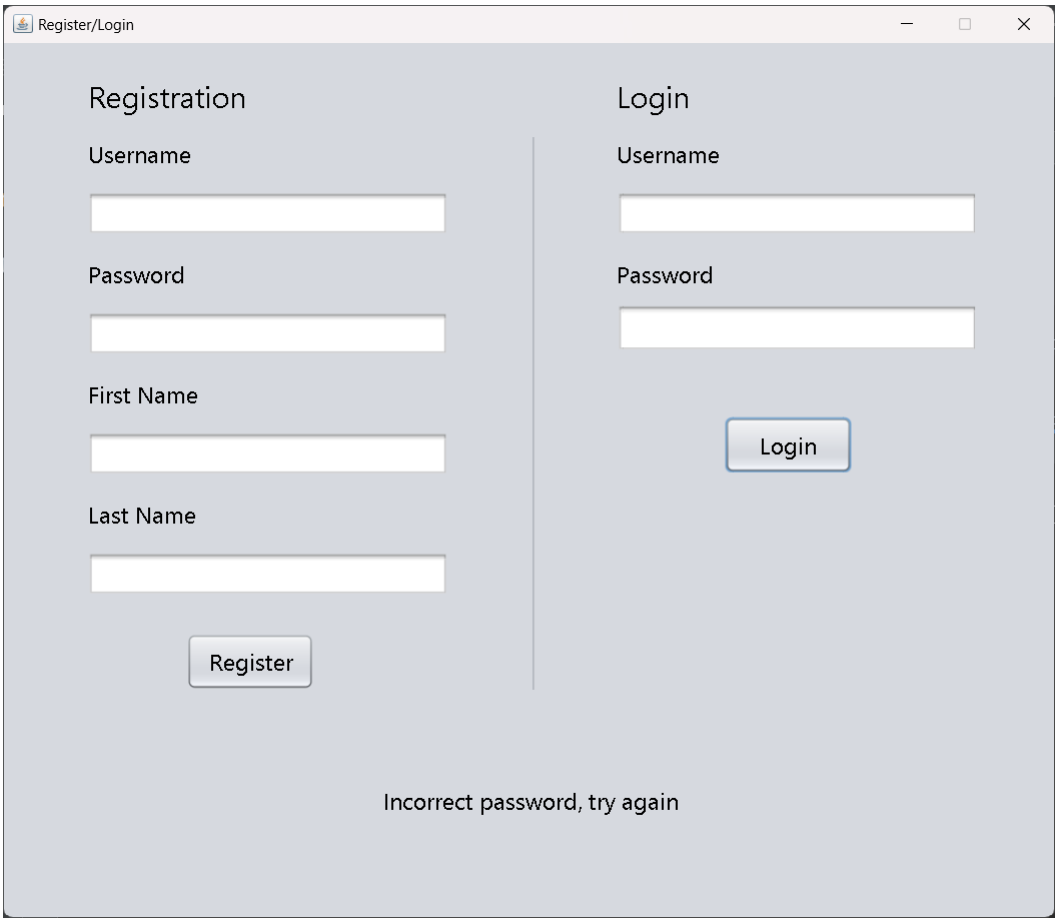
Error message for Empty Last Name



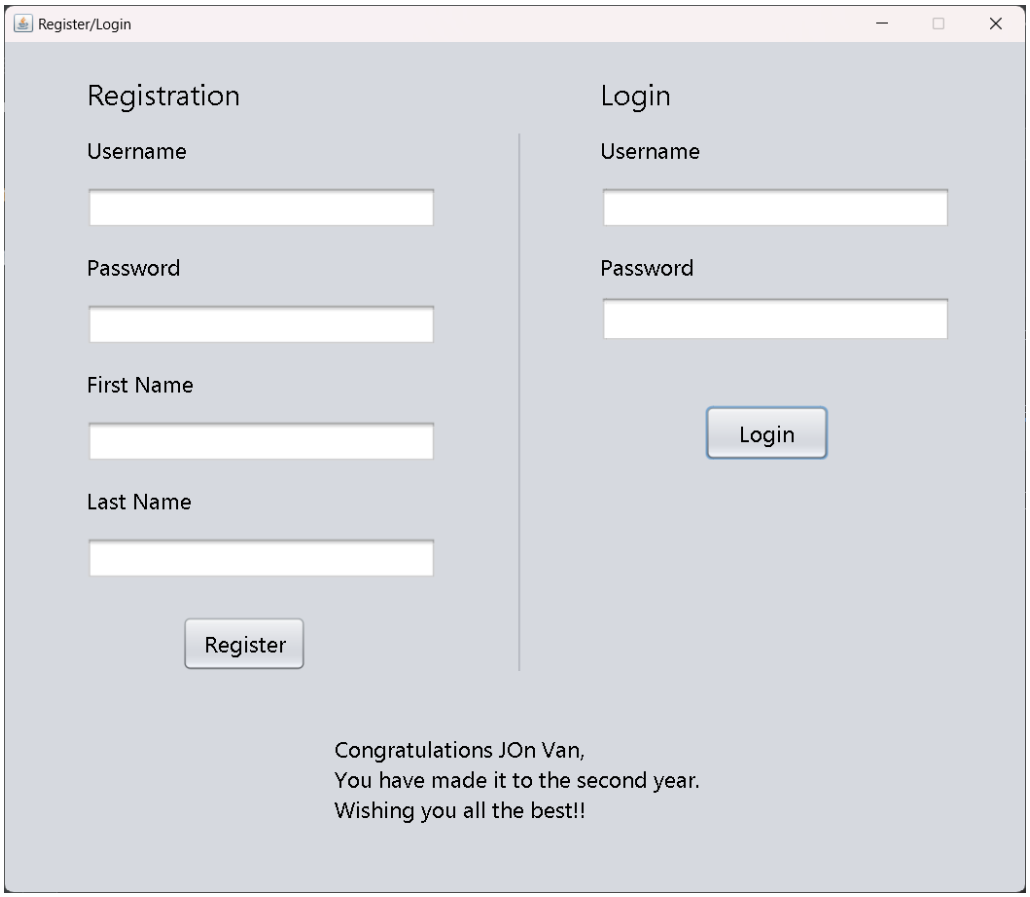
Successful Registration Attempt



Error message for Wrong Details



Error message for Incorrect password



Successful Login

Declaration of Authenticity

A critical aspect of any assignment is *authenticity*. Because you are completing much of the work for the assignments *unsupervised*, the examiner must be convinced that it is all your own work. For this reason, you must complete the *Declaration of Authenticity* provided in the study guide and have it counter-signed by your manager, mentor, or lecturer.

NB	The declaration of authenticity is a legal document, and if found that you have made a false declaration, then not only will your results be declared null and void, but you could also have criminal charges brought against you. It is not worth taking the risk!

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I Jonathan Van Eyssen hereby
(FULL NAME)

declare that the contents of this assignment is entirely my own work with the exception of the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date

Signature: _____ **Date:** 20 August 2024