

A JOURNAL OF THE “RESUME ADDER” APPLICATION

Honors Thesis

**Presented in Partial Fulfillment of the Requirements
For the Degree of Bachelor of Science**

In the School of Arts and Sciences
at Salem State University

By

Jackson Schultz

Dr. Joseph Kasprzyk
Faculty Advisor
Department of Computer Science

Commonwealth Honors Program
Salem State University
2014

Abstract

The computer science programming process has many different elements. From the software development life cycle, to the hands on implementation, to providing the user with the results they are looking for, the procedure has to not on complex, but precise. There have been many different writings related to the SDLC, but this document details the process that I used to design my application for user operation.

Journal of my CS Project

Jackson Schultz

3/2/14

From Day 1, I had a feeling I would be in for something new, in terms of developing a program. Writing a computer science program from start to finish, and going through the steps of the whole software design lifecycle was definitely a trial of what I have learned in my four years as a Computer Science major. My goals for this project were to gain a more substantive knowledge of the programming process, server communication, database access, experience working with the Apache POI, and increase my knowledge of java programming.

The first mountain I had to climb was to present my project idea to a group of faculty. I wasn't sure how well received my idea was going to be, but I hoped they would enjoy the concept. My idea was to create a program that lets a user add information to his or her resume. Really what I wanted to do for this project was to create an application that would help a user keep track of their accomplishments, be able to write them into different text fields, and store them somewhere they can view them. This application can be used by a user who just wants to convenience themselves when writing a resume or can be used by a company or employer to see what their employees are accomplishing or presenting on their off time/company time.

For my design of the project, I planned on having a user create an account in the database. In their account, the user would enter their preferred username and password, confirm their password, and enter an email address in which they'd like to have files sent.

After the account is created, the user will be able to log in with their credentials. Once logged in, a window opens with four differently labeled text areas titled Achievements, Skills, Work History, and Organizations. The user should then begin to enter information to the text fields of their choice. Once satisfied with their input, they are able to push the “save” button, which essentially adds the user’s inputted information into a formatted word document titled *username*/Resume.docx. That file will then be emailed to the user based on the email address that they registered in the database when their account was created. They may now close out the program, log back in, and click the refresh button, which will pull in all of the code that they inputted the last time they used the application.

Development was the third stage of my project. I began by creating my ResumeEditor.java class, whose primary function was to create the window that allows users to input their text. Working with the Apache POI was something I had never done before, but I was fortunate there were many different resources and tutorials for how to use these libraries. I had to import libraries, specifically dom4j-1.6.1.jar, xmlbeans-2.3.0.jar, poi-3.10-FINAL-20140208.jar, poi-ooxml-3.10-FINAL-20140208.jar, and poi-scratchpad-3.10-FINAL-20140208.jar. These libraries give a coder the ability to read and write to, in full functionality, the elements of the Microsoft Suite, i.e.: Microsoft Word and Microsoft Excel. The classes that I imported for this code are:

```
import org.apache.poi.xwpf.usermodel.ParagraphAlignment;
import org.apache.poi.xwpf.usermodel.XWPFDocument;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFRun;
```

These imports give me the ability to read and write, assuming they are coded correctly. Each line that the user enters into the text fields is stored as a paragraph, and that was how I would know which lines of text to pull.

After I had a preliminary idea of how the class would work, I began to move into my login class. For initial testing purposes, I used a test username and password that I hard coded into my login class, rather than dealing with linking it to a database right away. I was able to create the login panel, and it did log in with the hard coded username and password. Once the correct credentials were entered into the login class, ResumeEditor.java would open.

My next job was to log into my database, and set up a few test elements that I would later try and pull. I began by creating a table called users, and filling it with login credentials. My table consisted of four categories, ID, username, password, and information. I started with just one value, username: Jackson, password: password. Once I got that all set up, I had to find a way to pull the information into the java program. I knew that I would need to import a library so that I could import methods that I would use. I imported the mysql-connector-java-5.1.30-bin.jar library so that I could use java to work with mysql. I was planning on adding the connection within my loginPanel.java class, but my supervisor advised me to create a new class that solely deals with connection. Now, I am glad that it was implemented successfully.

The class that I made to create a connection was called ConnectionCreator.java. This class imports java.sql.Connection and java.sql.DriverManager. These classes allow me to create a connection by setting strings equal paths and login information.

```
private static String url = "jdbc:mysql://173.244.1.36:3306/";
```

```
private static String dbName = "w_schultz_db";  
private static String driver = "com.mysql.jdbc.Driver";  
private static String userName = "w_schultz";  
private static String password = "New2014";
```

I then used instances to create the connection.

I decided it would be a good idea to provide the user with the ability receive an email with the resume file that they designed attached. When save is clicked after a user updates their achievement information, an email from me will be sent to that person's email address. The file is formatted with titles before each inputted value from the user.

From there I added my CreateUserGUI.java class, which essentially adds a user to a database based on the connection. The user should press the button "New User" and enter his or her information, username, password, confirm password, and their email address in which they would like files to be sent. Lastly, my sendPassword.java class checks the database for the entered username, matches that data line, reads the information, and send the password to that user's email.

For my testing portion of the process, I used a few different values to test that I could create a user, create a file, and send it to the user's email address. I had to test that the database was functional, and could pull and read passwords if the user forgot that as well. Everything was functional by presentation time.

I learned a lot about the Software Development Life Cycle, Object Oriented Programming, working with different packages, and combining a programming language with mysql. I also learned a lot from my failures and from the problems that occurred, specifically trying to get the program to write to a remote server and store the file. I

worked with one of the faculty members to try and get my server path created and set up, and unfortunately it was much more involved than we had previously thought. We were able to read the file that was already on the server, but not write to it. I plan to expand this project and get it to write to the files to LinkedIn further down the road. I enjoyed this project and feel like I have learned a lot about the computer science disciplines.

Works Cited

<http://www.mysql.com>

<http://docs.oracle.com/javase/5/api/javax/mail/package-summary.html>

<http://developer.linkedin.com/apis>

<http://poi.apache.org>

<https://netbeans.org>

<http://docs.oracle.com/javase/tutorial/>

<http://stackoverflow.com/questions/5085105/sending-files-from-server-to-client-in-java>

Appendix
Sample Code:
Class name is contained in “public class x...”

```
/*  
 * To change this license header, choose License Headers in Project  
Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
*/  
package resumeaddertry;  
  
import java.awt.BorderLayout;  
import java.awt.Container;  
import java.awt.Dimension;  
import java.awt.FlowLayout;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JPasswordField;  
import javax.swing.JTextField;  
import resumeaddertry.LoginPanel;  
  
public class changeAccountInfo extends JFrame implements ActionListener  
{  
    private static final long serialVersionUID = 2852777524066323484L;  
    private JPanel mainPanel;  
    private JPanel buttonPanel;  
    //private JLabel oldUserName = new JLabel("Old user name",  
JLabel.RIGHT);
```

```

private JLabel userName = new JLabel("New User name",
JLabel.RIGHT);
private JLabel password = new JLabel("New Password", JLabel.RIGHT);
private JLabel confirmPassword = new JLabel("repeat New Password",
JLabel.RIGHT);
private JTextField usernameArea = new JTextField(20);
//private JTextField oldUsernameArea = new JTextField(20);
private JPasswordField passwordField = new JPasswordField(20);
private JPasswordField confirmPasswordField = new JPasswordField(20);
private JLabel emailLabel = new JLabel("New Email Address",
JLabel.RIGHT);
private JTextField emailArea = new JTextField();
private JButton editUserButton;
private JButton cancelButton;
private Connection theConnection;
private JButton loginButton = new JButton("Login");
private String user = "";
private String pass = "";
private int id;
private String curious;

public changeAccountInfo( Connection aConnection, String s, String p )
{
    super("Edit your account information");
    user = s;
    pass = p;
    //System.out.println(user);
    //System.out.println(pass);
    theConnection = aConnection;
    initialize();
}

@Override
public void actionPerformed((ActionEvent ae) )
{
    String newUserUsername = usernameArea.getText();
    String newPassword = String.valueOf(passwordField.getPassword());
    String confirmNewPassword =
String.valueOf(confirmPasswordField.getPassword());
    String newUserEmailAddress = String.valueOf(emailArea.getText());

```

```

if ( newUserUserName.isEmpty() )
{
    JOptionPane.showMessageDialog(null, "Failed to enter a user
name"); //make sure user enters username
}
else if ( newUserEmailAddress.isEmpty() )
{
    JOptionPane.showMessageDialog(null, "Failed to enter an email
address"); //make sure that user enters email
}
else if ( newPassword.equals(confirmNewPassword) )
{
    try
    {
        String query1 = "select id from users where username = \"\" +
//selection information
        user + "\"";
        String query2 = "select username from users where username = \"\"
+
        user + "\"";
        String query3 = "select password from users where username = \"\"
+
        user + "\"";
        String query4 = "select information from users where username =
\"\" +
        user + "\"";
        ResultSet res1 =
theConnection.createStatement().executeQuery(query1);
        ResultSet res2 =
theConnection.createStatement().executeQuery(query2);
        ResultSet res3 =
theConnection.createStatement().executeQuery(query3);
        ResultSet res4 =
theConnection.createStatement().executeQuery(query4); //set to result set

        res1.next(); //next command
        res2.next();
        res3.next();
        res4.next();
    }
}
}

```

is

```
!!! before you can interact with a ResultSet in order to access its
!!! contents, you must first issue a .next(), otherwise the ResultSet
```

```
!!! internally pointing to BEFORE the initial row of the results
```

```
!!! and for your attempt at the next line - AW, COME ON!
```

```
!!! curious = res1.getString(query1);
```

```
curious = res1.getString("id");
```

```
String us = res2.getString("username");
```

```
String ps = res3.getString("password");
```

```
String in = res4.getString("information"); //pull fields
```

```
System.out.println(curious); // print values, all are being pulled
```

```
System.out.println(us);
```

```
System.out.println(ps);
```

```
System.out.println(in);
```

```
//      String query4 = "select information from users where username =
//      \"\" +
```

```
//      user + "\";";
```

```
String insertNewUserQuery = ("UPDATE users SET username "
+ "= "+ newUserName + ",password = "+ newPassword + ",
information = "
+ newUserEmailAddress + " WHERE id = " + curious + ";");
```

```
//query
```

```
System.out.println(insertNewUserQuery);
```

```
+ //String insertNewUserQuery = "update users set username = \"\"
```

```
+ //newUserName + "\" = where id = \"\" + curious + "\";";
```

```
//      password, information) " +
```

```
//      "values (\"\" + newUserName + "\", \"\" + newPassword + "\",
```

```
//      \"\" +
```

```
//      newUserEmailAddress + "\") where id = \"\" +
```

```
//      curious + "\";";
```

```
theConnection.createStatement().executeUpdate(insertNewUserQuery);
```

```

        JOptionPane.showMessageDialog(null, "New user inserted into
database");
        this.dispose();

    }
    catch ( java.sql.SQLException anSQLException )
    {
        System.out.println("SQLException caught: " +
anSQLException.getMessage());
        System.out.println(" reported database state: " +
anSQLException.getSQLState());
        anSQLException.printStackTrace();
    }
    System.out.println("leaving changeAccount");
}
}

public changeAccountInfo( Connection aConnection )
{
    super("Edit your account information");
    theConnection = aConnection;
    initialize();
}

private void initialize()
{
    this.setLocation(500, 280);
    this.setMinimumSize(new Dimension(300, 200));

    mainPanel = new JPanel(new BorderLayout());

    JPanel gridPanel = new JPanel(new GridLayout(5, 2, 5, 5));
    //gridPanel.add(oldUserName);
    //gridPanel.add(oldUsrnameArea);
    gridPanel.add(userName);
    gridPanel.add(usrnameArea);
    gridPanel.add(password);
    gridPanel.add(passwordField);
    gridPanel.add(confirmPassword);

```

```
gridPanel.add(confirmPasswordField);
gridPanel.add(emailLabel);
gridPanel.add(emailArea);
mainPanel.add(gridPanel, BorderLayout.NORTH);
```

```
buttonPanel = new JPanel(new FlowLayout());
```

```
editUserButton = new JButton("Edit User");
editUserButton.addActionListener(changeAccountInfo.this);
//cancelButton.addActionListener(cancelAction.this);
buttonPanel.add(editUserButton);
//buttonPanel.add(cancelButton);
```

```
mainPanel.add(buttonPanel, BorderLayout.SOUTH);
```

```
Container aContainer = this.getContentPane();
aContainer.add(mainPanel);
```

```
this.pack();
```

```
}
```

```
public class cancelActionPerformed implements ActionListener
```

```
{
```

```
    @Override
```

```
    public void actionPerformed( ActionEvent ae )
```

```
    {
```

```
        dispose();
```

```
    }
```

```
}
```

```
}
```

```
package resumeaddertry;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
public class ConnectionCreator
```

```
{
```

```
    private static Connection theConnection;
```

```
    private static String url = "jdbc:mysql://173.244.1.36:3306/";
```

```

private static String dbName = "w_schultz_db";
private static String driver = "com.mysql.jdbc.Driver";
private static String userName = "w_schultz";
private static String password = "New2014";

private ConnectionCreator()
{
    // prevent instantiation of a class containing nothing but static elements
}

public static Connection createConnector()
{
    if ( theConnection != null )
    {
        //System.out.println("Already connected - using existing
connection");
        return theConnection;
    }

    //System.out.println("Connecting...");
    try
    {
        Class.forName(driver).newInstance();
        Connection newConnection = DriverManager.getConnection(url +
dbName, userName,
password); //this shows we've connected to the database

        /*
        st = newConnection.createStatement();
        res = st.executeQuery("SELECT * FROM users");

        ResultSetMetaData resultSetInfo = res.getMetaData();
        int fieldCount = resultSetInfo.getColumnCount();
        for ( int index = 1; index <= fieldCount; index++ )
        {
            // System.out.print("(" + index + ")" +
resultSetInfo.getColumnName(index) +
            " ");
        }

```

```

//System.out.println();

while ( res.next() )
{
    int id = res.getInt("id");
    String msg = res.getString("username");
//    System.out.println(id + "\t" + msg);
    String username = res.getString("username");
    String password = res.getString("password");
    String information = res.getString("information");
//    System.out.println("Username: " + username +
        " password: " + password +
        " information: " + information);
}
*/

theConnection = newConnection;
return theConnection;
}
catch ( java.lang.ClassNotFoundException classNotFoundException )
{
    classNotFoundException.printStackTrace(); //Thrown when an
application tries to load in a class through its string name using: //The
forName method in class Class. //The findSystemClass method in class
ClassLoader . //The loadClass method in class ClassLoader.
    return null;
}
catch ( java.lang.InstantiationException instantiationException )
{
    instantiationException.printStackTrace(); //Thrown when an
application tries to create an instance of a class using the newInstance
method in class Class, but the specified class object cannot be instantiated
    return null;
}
catch ( java.lang.IllegalAccessException illegalAccessException )
{
    illegalAccessException.printStackTrace(); //An
IllegalAccessException is thrown when an application tries to reflectively
create an instance (other than an array), set or get a field, or invoke a

```


method, but the currently executing method does not have access to the definition of the specified class, field, method or constructor.

```
        return null;
    }
    catch ( java.sql.SQLException sqlException ) //An exception that
provides information on a database access error or other errors.
    {
        sqlException.printStackTrace();
        return null;
    }
}
}
```

```
/*
```

```
* To change this license header, choose License Headers in Project
Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package resumeaddertry;
```

```
import javax.swing.JLabel;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import java.awt.Container;
import java.awt.Dimension;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
```

```

public class CreateUserGUI extends JFrame implements ActionListener
{
    private static final long serialVersionUID = 2852777524066323484L;

    private JPanel mainPanel;
    private JPanel buttonPanel;
    private JLabel userName = new JLabel("User name", JLabel.RIGHT);
    private JLabel password = new JLabel("Password", JLabel.RIGHT);
    private JLabel confirmPassword = new JLabel("repeat Password",
JLabel.RIGHT);
    private JTextField usrnameArea = new JTextField(20);
    private JPasswordField passwordField = new JPasswordField(20);
    private JPasswordField confirmPasswordField = new JPasswordField(20);
    private JLabel emailLabel = new JLabel("Email Address",
JLabel.RIGHT);
    private JTextField emailArea = new JTextField();

    private JButton createUserButton;
    private static Connection theConnection;

    @java.lang.Override
    public void actionPerformed( ActionEvent ae )
    {
        String newUserName = usrnameArea.getText(); //set value = value in
text area
        String newPassword = String.valueOf(passwordField.getPassword());
//set value = value in password area
        String confirmNewPassword =
String.valueOf(confirmPasswordField.getPassword()); //set value = value in
confirmpassword area

        //@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        String newUserEmailAddress = String.valueOf(emailArea.getText());
//same thing for email

        if ( newUserName.isEmpty() )
        {
            JOptionPane.showMessageDialog(null, "Failed to enter a user
name"); //if the user leaves username area blank, show

```



```

        String insertNewUserQuery =
            "insert into users(id, username, password, information) " +
            "values (" + newID + ", \"" + newUserName + "\", \"" +
newPassword + "\", \"" +
            newUserEmailAddress + "\");";
//        System.out.println("in CreateUserGUI, inserting new user into
database, " +
//        "query is " + insertNewUserQuery);
        try
        {

theConnection.createStatement().executeUpdate(insertNewUserQuery);
            JOptionPane.showMessageDialog(null, "New user inserted
into database");
            this.dispose();
        }
        catch ( java.sql.SQLException anSQLException )
        {
            //System.out.println("Oops, something went wrong while
inserting new user into database");
            //System.out.println(anSQLException.getMessage());
            //anSQLException.printStackTrace();
        }
    }
    catch ( SQLException anException )
    {
//        System.out.println("Exception caught while trying to create \n" +
//        "a new user in CreateUserGUI: " + anException.getMessage());
        anException.printStackTrace();
        throw new RuntimeException(anException);
    }
}
else
{
    JOptionPane.showMessageDialog(null, "Passwords do not match");
}
}

public CreateUserGUI( Connection aConnection )

```

```

    {
        super("Create a new user");
        theConnection = aConnection;
        initialize(); //run initialize()
    }

    private void initialize() //initialize puts all the elements in the GUI and
    waits to be ran
    {
        this.setLocation(500, 280);
        this.setMinimumSize(new Dimension(300, 200));

        mainPanel = new JPanel(new BorderLayout());

        JPanel gridPanel = new JPanel(new GridLayout(4, 2, 5, 5));
        gridPanel.add(userName);
        gridPanel.add(usrnameArea);
        gridPanel.add(password);
        gridPanel.add(passwordField);
        gridPanel.add(confirmPassword);
        gridPanel.add(confirmPasswordField);
        gridPanel.add(emailLabel);
        gridPanel.add(emailArea);
        mainPanel.add(gridPanel, BorderLayout.NORTH);

        buttonPanel = new JPanel(new FlowLayout());

        createUserButton = new JButton("Create User");
        createUserButton.addActionListener(CreateUserGUI.this);
        buttonPanel.add(createUserButton);

        mainPanel.add(buttonPanel, BorderLayout.SOUTH);

        Container aContainer = this.getContentPane();
        aContainer.add(mainPanel);

        this.pack();
    }
}

```

```

package resumeaddertry;

```

```

import java.sql.Connection;
import java.sql.ResultSet;

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

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import javax.swing.JLabel;

@SuppressWarnings("serial")
public class LoginPanel extends JFrame
{
    private JButton loginButton = new JButton("Login");
    private JButton newUserButton = new JButton("New User");
    private JButton editAccount = new JButton("Edit Account");
    private JPanel mainPanel = new JPanel();
    private JTextField userNameTextField = new JTextField();
    private JPasswordField passwordPasswordField = new
JPasswordField(15);
    private JLabel usnameLabel = new JLabel("Username");
    private JLabel pswordLabel = new JLabel("Password");
    private Connection theConnection;
    private JButton forgotPassword = new JButton("I forgot my password");
    private String suppliedUserName;

    public LoginPanel( Connection aConnection ) //creating the gui and
importing the variable
    {
        super("Login Authentication");

        theConnection = aConnection;

        setSize(330, 250);
        setLocation(500, 280);

```

```

mainPanel.setLayout(null);

userNameTextField.setBounds(115, 30, 150, 20);
passwordPasswordField.setBounds(115, 65, 150, 20);
usernameLabel.setBounds(35, 27, 80, 25);
pswordLabel.setBounds(38, 62, 80, 25);

newUserButton.setBounds(10, 100, 80, 20);
loginButton.setBounds(110, 100, 80, 20);
editAccount.setBounds(210, 100, 100, 20);
forgotPassword.setBounds(90, 150, 160, 20);

loginButton.addActionListener(new LoginButtonActionProcessor());
newUserButton.addActionListener(new NewUserButtonAction());
editAccount.addActionListener(new EditButtonAction());
forgotPassword.addActionListener(new forgottenPasswordAction());

mainPanel.add(newUserButton);
mainPanel.add(editAccount);
mainPanel.add(usernameLabel);
mainPanel.add(pswordLabel);
mainPanel.add(loginButton);
mainPanel.add(userNameTextField);
mainPanel.add(passwordPasswordField);
mainPanel.add(forgotPassword);

getContentPane().add(mainPanel);
}
public class EditButtonAction implements ActionListener //edit button
action. when clicke, implement
{
    @Override
    public void actionPerformed( ActionEvent ae ){

        String suppliedUserName = userNameTextField.getText();
        char[] temp = passwordPasswordField.getPassword();
        String suppliedPassword = String.valueOf(temp);
        String text = String.copyValueOf(temp);
        // System.out.println(suppliedUserName);
        //System.out.println(text);
    }
}

```

```

        boolean isValidUser = validateUserPassword(suppliedUserName,
suppliedPassword);
        if ( isValidUser )
        {
            //ResumeEditor newApp = new
ResumeEditor(suppliedUserName);
            // changeAccountInfo created = new
changeAccountInfo(theConnection);
            changeAccountInfo created = new
changeAccountInfo(theConnection, suppliedUserName, text);

            created.setVisible(true);
            dispose();
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Wrong Password /
Username");
            userNameTextField.setText("");
            passwordPasswordField.setText("");
            userNameTextField.requestFocus();
        }
    }
}

```

```

private class LoginButtonActionProcessor implements ActionListener
//login button action, when clicked, implement
{
    @Override
    public void actionPerformed( ActionEvent ae )
    {
        String suppliedUserName = userNameTextField.getText();
        char[] temp = passwordPasswordField.getPassword();
        String suppliedPassword = String.valueOf(temp);

        boolean isValidUser = validateUserPassword(suppliedUserName,
suppliedPassword);

```



```

        if ( isValidUser )
        {
            //ResumeEditor newApp = new
ResumeEditor(suppliedUserName);
            ResumeEditor newApp = new ResumeEditor(suppliedUserName,
theConnection);
            newApp.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            newApp.setVisible(true);
            /* make the login panel dispose of itself after it has performed its
intended
            * functionality */
            dispose();
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Wrong Password /
Username");
            userNameTextField.setText("");
            passwordPasswordField.setText("");
            userNameTextField.requestFocus();
        }
    }
}

```

```

private class NewUserButtonAction implements ActionListener
{
    @java.lang.Override
    public void actionPerformed((ActionEvent ae)
    {
        CreateUserGUI created = new CreateUserGUI(theConnection);
        created.setVisible(true);
    }
}

```

```

private class forgottenPasswordAction implements ActionListener //user
forgot password?
{
    @java.lang.Override
    public void actionPerformed( ActionEvent ae )
    {

```

```

        try {
            sendPassword emailer = new sendPassword(theConnection,
suppliedUserName);
        } catch (Exception anException) {
//            System.out.println("    Attempting to"
//                + " send password");
//            System.out.println("    " + anException.getMessage());
            anException.printStackTrace();
        }
//sendPassword passwd = new sendPassword(theConnection);
//created.setVisible(true);
    }
}

```

```

public boolean validateUserPassword( String suppliedUserName, String
suppliedPassword ) //checking the password for the user

```

```

{
    try
    {
        String query = "select password from users where username = \"\" +
            suppliedUserName + \"\";";
//System.out.println("in LoginPanel.getData, query is " + query);

        ResultSet res =
theConnection.createStatement().executeQuery(query);

        if ( res.next() )
        {
            String actualPassword = res.getString("password");
//System.out.println("Password for " + suppliedUserName + ": " +
actualPassword);
            if ( actualPassword.equalsIgnoreCase(suppliedPassword) )
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    }
}

```

```

        else
        {
            //System.out.println("User " + suppliedUserName + " cannot be
located");
            return false;
        }
    }
    catch ( Exception ex )
    {
        //System.out.println(ex);
        return false;
    }
}
}
}

```

```

package resumeaddertry;

```

```

import java.sql.Connection;
import javax.swing.JFrame;

```

```

public class Main
{

```

```

    public static void main( String[] args ) throws Exception
    {
        Connection aConnection = ConnectionCreator.createConnector();
//create and import connection

        if ( aConnection == null )
        {
            throw new Exception("Unable to create a connection to the user
database - contact" +
                " Willis J. Schultz at wjacksonschultz at gmail.com"); //exception
that will show to user
        }
        LoginPanel aLoginPanel = new LoginPanel(aConnection); //import
loginpanel
        aLoginPanel.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        aLoginPanel.setVisible(true); //run loginpanel
    }
}

```

```
}  
}
```

```
package resumeaddertry;
```

```
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;
```

```
import java.sql.Connection;  
import java.awt.Dimension;  
import java.awt.FlowLayout;  
import java.awt.BorderLayout;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

```
import java.util.List;
```

```
import java.awt.Container;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTextArea;
```

```
import org.apache.poi.xwpf.usermodel.ParagraphAlignment;  
import org.apache.poi.xwpf.usermodel.XWPFDocument;  
import org.apache.poi.xwpf.usermodel.XWPFParagraph;  
import org.apache.poi.xwpf.usermodel.XWPFRun;
```

```
@SuppressWarnings("serial")  
public class ResumeEditor extends JFrame {
```

```
    private JTextArea achievementsTextArea;  
    private JTextArea skillsTextArea;  
    private JTextArea workHistoryTextArea;  
    private JTextArea organizationsTextArea;
```

```
    private JLabel infoLabel;
```

```

private JLabel achievementsLabel;
private JLabel skillsLabel;
private JLabel workHistoryLabel;
private JLabel organizationsLabel;

private JButton saveButton;
private JButton refreshButton;

private JPanel mainPanel;

private JPanel headerPanel;
private JPanel dataPanel;
private JPanel buttonPanel;

private String fileName;
private String location = ""; //cs.salemstate.edu/~w_schultz/Resumes/";
private String suppliedUserName;

private Connection theConnection;

public ResumeEditor(String theSuppliedUserName, Connection
aConnection) { //import args
    super("Resume editor GUI");

    suppliedUserName = theSuppliedUserName; //set value(s)
    theConnection = aConnection;
    fileName = suppliedUserName + "Resume.docx";

    //System.out.println("Inside ResumeAdder() constructor");
    initialize();
}

private void initialize() {
    Container aContainer = this.getContentPane(); //import
    aContainer.setLayout(new FlowLayout()); // import flowlayout

    mainPanel = initializeMainPanel(); //initalize the panel

    aContainer.add(mainPanel); // add the panel to the frame

```

```

        this.setMinimumSize(new Dimension(100, 100)); //demensions
        this.setMaximumSize(new Dimension(100, 100));

        this.pack(); //put everything in together
    }

    private JPanel initializeMainPanel() {
        mainPanel = new JPanel(new BorderLayout(10, 10));

        headerPanel = initializeHeaderPanel();
        mainPanel.add(headerPanel, BorderLayout.NORTH);

        dataPanel = initializeDataPanel();
        mainPanel.add(dataPanel, BorderLayout.CENTER);

        buttonPanel = initializeButtonPanel();
        mainPanel.add(buttonPanel, BorderLayout.SOUTH);

        return mainPanel;
    }

    private JPanel initializeHeaderPanel() {
        headerPanel = new JPanel(new FlowLayout());

        infoLabel = new JLabel("Please Enter your information in a text area,"
            + " based on Category\n");
        headerPanel.add(infoLabel);

        return headerPanel;
    }

    private JPanel initializeDataPanel() { //the information from the datapanel
        //System.out.println("Hello from initializeDataPanel");
        dataPanel = new JPanel(new GridLayout(3, 1, 5, 10));

        infoLabel = new JLabel("Please Enter your information in the text areas
below"
            + " based on Category\n");
        achievementsLabel = new JLabel("Achievements", JLabel.RIGHT);

```

```

skillsLabel = new JLabel("Skills", JLabel.RIGHT);
workHistoryLabel = new JLabel("Work History", JLabel.RIGHT);
organizationsLabel = new JLabel("Organizations", JLabel.RIGHT);

achievementsTextArea = new JTextArea(5, 40);
achievementsTextArea.setLineWrap(true);
achievementsTextArea.setWrapStyleWord(true);
skillsTextArea = new JTextArea(5, 40);
skillsTextArea.setLineWrap(true);
skillsTextArea.setWrapStyleWord(true);
workHistoryTextArea = new JTextArea(5, 40);
workHistoryTextArea.setLineWrap(true);
workHistoryTextArea.setWrapStyleWord(true);
organizationsTextArea = new JTextArea(5, 40);
organizationsTextArea.setLineWrap(true);
organizationsTextArea.setWrapStyleWord(true);

JPanel achievementPanel = new JPanel(new FlowLayout());
achievementPanel.add(achievementsLabel);
achievementPanel.add(achievementsTextArea);
JPanel skillsPanel = new JPanel(new FlowLayout());
skillsPanel.add(skillsLabel);
skillsPanel.add(skillsTextArea);
JPanel workHistoryPanel = new JPanel(new FlowLayout());
workHistoryPanel.add(workHistoryLabel);
workHistoryPanel.add(workHistoryTextArea);
JPanel organizationPanel = new JPanel(new FlowLayout());
organizationPanel.add(organizationsLabel);
organizationPanel.add(organizationsTextArea);

//System.out.println(dataPanel.getLayout().toString());
dataPanel.add(achievementPanel);
dataPanel.add(skillsPanel);
dataPanel.add(workHistoryPanel);
dataPanel.add(organizationPanel);

//System.out.println("Leaving initializeDataPanel");
return dataPanel;
}

```

```

private JPanel initializeButtonPanel() {
    buttonPanel = new JPanel(new FlowLayout());

    saveButton = new JButton("Save");
    refreshButton = new JButton("Refresh");

    refreshButton.addActionListener(new RefreshActionofCreate());
    saveButton.addActionListener(new SaveActionofCreate());

    buttonPanel.add(saveButton);
    buttonPanel.add(refreshButton);

    return buttonPanel;
}

//-----
-----
public class SaveActionofCreate implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent anActionEvent) {
        try {
            //System.out.println("Inside
SaveActionOfCreate.actionPerformed()");

            XWPFDocument createdDocument = new XWPFDocument();

            XWPFParagraph Label1 = createdDocument.createParagraph();
            XWPFRun runLabel1 = Label1.createRun();
            runLabel1.setFontSize(20);
            runLabel1.setText("Achievements: \n");

            XWPFParagraph achievements =
createdDocument.createParagraph();
            XWPFRun run = achievements.createRun();
            achievements.setAlignment(ParagraphAlignment.CENTER);
            run.setText(achievementsTextArea.getText());
            run.addBreak();

            XWPFParagraph Label2 = createdDocument.createParagraph();

```



```
XWPFRun runLabel2 = Label2.createRun();
runLabel2.setFontSize(20);
runLabel2.setText("Skills: \n");
```

```
XWPFParagraph skills = createdDocument.createParagraph();
XWPFRun run2 = skills.createRun();
skills.setAlignment(ParagraphAlignment.CENTER);
run2.setText(skillsTextArea.getText()); //activity
run2.addBreak();
```

```
XWPFParagraph Label3 = createdDocument.createParagraph();
XWPFRun runLabel3 = Label3.createRun();
runLabel3.setFontSize(20);
runLabel3.setText("Work History: \n");
```

```
XWPFParagraph workHistory =
createdDocument.createParagraph();
XWPFRun run3 = workHistory.createRun();
workHistory.setAlignment(ParagraphAlignment.CENTER);
run3.setText(workHistoryTextArea.getText()); //activity
run3.addBreak();
```

```
XWPFParagraph Label4 = createdDocument.createParagraph();
XWPFRun runLabel4 = Label4.createRun();
runLabel4.setFontSize(20);
runLabel4.setText("Organizations: \n");
```

```
XWPFParagraph organizations =
createdDocument.createParagraph();
XWPFRun run4 = organizations.createRun();
organizations.setAlignment(ParagraphAlignment.CENTER);
run4.setText(organizationsTextArea.getText());
run4.addBreak();
```

```
try {
    //"http://dev1:8080/data/xml/myTestFile123.xml
    ///!! DO NOT UNCOMMENT THIS LINE OR CHANGE THE
ONE BELOW IT
    //FileOutputStream output = new FileOutputStream(location +
fileName);
```

```

        FileOutputStream output = new FileOutputStream(location +
fileName);
        createdDocument.write(output);
        output.close();
    } catch (Exception anException) {
//      System.out.println("  Attempting to write document to
disk,\n"
//          + "  exception caught");
//      System.out.println("  " + anException.getMessage());
//      anException.printStackTrace();
    }

    //@@@@
    try {
        SendFile mailer = new SendFile(theConnection,
suppliedUserName);
    } catch (Exception anException) {
//      System.out.println("  Attempting to mail document,"
//          + "  exception caught");
//      System.out.println("  " + anException.getMessage());
//      anException.printStackTrace();
    }
    } catch (Exception umbrellaException) {
//      System.out.println("  near end of
SaveActionOfCreate.actionPerformed(),\n"
//          + "  unexpected exception caught");
//      System.out.println("  " + umbrellaException.getMessage());
//      umbrellaException.printStackTrace();
    }
}
}
}

```

```

public class RefreshActionofCreate implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        //System.out.println("Inside
RefreshActionofCreate.actionPerformed()");

        File aFile = new File(location + fileName);

```



```

import javax.mail.MessagingException;
import javax.mail.Multipart;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;

public class SendFile {

    private Connection theConnection;
    private String email;

    //public static void main (String[] args) {
    //    SendFile send = new SendFile(aConnection);
    // }
    public SendFile(Connection aConnection, String suppliedUserName) {
    //email the file

        theConnection = aConnection;
        //@@@ observation = "wjacksonschultz@gmail.com" is NOT a user
name...
        final String username = "wjacksonschultz@gmail.com"; //username we
will send emails from
        email = username;
        final String password = "hampden14"; //password of the sender account

        Properties emailProperties = new Properties();
        emailProperties.put("mail.smtp.auth", true);
        emailProperties.put("mail.smtp.starttls.enable", true);
        emailProperties.put("mail.smtp.host", "smtp.gmail.com");
        emailProperties.put("mail.smtp.port", "587");

        Session session = Session.getInstance(emailProperties,
            new javax.mail.Authenticator() {
                protected PasswordAuthentication getPasswordAuthentication()
            }
        )
        {
            return new PasswordAuthentication(username, password);
        }
    }
}

```

```

        }
    });
    try {
        //String query = "select information from users where information =
\"\" +
        // email + "\"";

        //@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        String query = "select information from users where username = \"\"
        + suppliedUserName + "\""; //db query

        //System.out.println("in CreateUserGUI, checking is user name
exists, query is " + query);
        ResultSet res =
theConnection.createStatement().executeQuery(query);

        //@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        if (res.next()) {
            email = res.getString("information"); //pull information field value
and set to variable email
        } else {
            email = null;
        }

    } catch (java.sql.SQLException anSQLException) {
        anSQLException.printStackTrace();
    }
    try {

        Message message = new MimeMessage(session);
        message.setFrom(new InternetAddress(username));

        //@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        String targetEmail = email; //this is the email address we will send
the information
        //System.out.println("Target email address is " + targetEmail);
        message.setRecipients(Message.RecipientType.TO,
            InternetAddress.parse(targetEmail));
        message.setSubject("Testing Subject"); //subject line
        message.setText("PFA");
    }
}

```

```

MimeBodyPart messageBodyPart = new MimeBodyPart();

Multipart multipart = new MimeMultipart();

messageBodyPart = new MimeBodyPart();
String file;
//file = "/Users/jacksonschultz/NetBeansProjects/Resume.docx";
//cs.salemstate.edu/~w_schultz/Resumes/
//@@@@@@@@@@@@@@@@
//file = "cs.salemstate.edu/~w_schultz/Resumes/index.html";
file = "" + suppliedUserName + "Resume.docx"; //file to be attached
String fileName = suppliedUserName + "Resume.docx"; //same as
above
DataSource source = new FileDataSource(file);
messageBodyPart.setDataHandler(new DataHandler(source));
messageBodyPart.setFileName(fileName); //attach
multipart.addBodyPart(messageBodyPart);

message.setContent(multipart);

// System.out.println("Sending");

Transport.send(message);

// System.out.println("Done sending mail");

} catch (MessagingException anException) {
    //@@@@@@@@
// System.out.println(" In SendFile, exception caught: ");
// System.out.println(" " + anException.getMessage());
    anException.printStackTrace();
}
}
}
}

```

```

/*
* To change this license header, choose License Headers in Project
Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.

```

```
*/
package resumeaddertry;

import java.awt.BorderLayout;
import java.sql.Connection;
import java.sql.ResultSet;

import java.util.Properties;

import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;

import javax.mail.Message;
import javax.mail.MessagingException;
import java.lang.NullPointerException;
import javax.mail.Multipart;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;
import javax.swing.JFormattedTextField;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class sendPassword extends JFrame{

    private Connection theConnection;
    private String email;
    private String forgottenPassword;
    JFrame frame = new JFrame();

    public sendPassword(Connection aConnection, String suppliedUserName)
    {
```

```

theConnection = aConnection;

final String senderEmail = "wjacksonschultz@gmail.com"; //sender
email

final String password = "hampden14"; //sender password

Properties emailProperties = new Properties();
emailProperties.put("mail.smtp.auth", true);
emailProperties.put("mail.smtp.starttls.enable", true);
emailProperties.put("mail.smtp.host", "smtp.gmail.com");
emailProperties.put("mail.smtp.port", "587");

JPanel p = new JPanel(new BorderLayout());
JFormattedTextField f = new JFormattedTextField();
p.add(new JLabel("Please enter your Username:"),
BorderLayout.NORTH);
f.setColumns(20);
f.getText();
p.add(f);
JOptionPane.showConfirmDialog(null, p, "Email finder",
JOptionPane.OK_CANCEL_OPTION);

String testUsername = f.getText();
//System.out.println(testUsername);

Session session = Session.getInstance(emailProperties,
    new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication()
    {
        return new PasswordAuthentication(senderEmail, password);
    }
    });
try {

String query = "select information from users where username = \"\"
+ testUsername + \"\"";

String query1 = "select password from users where username = \"\"

```



```

        + testUsername + "\"";

        //System.out.println("in CreateUserGUI, checking is user name
exists, query is " + query);
        ResultSet res =
theConnection.createStatement().executeQuery(query);
        ResultSet res1 =
theConnection.createStatement().executeQuery(query1);

        //@@@@@@@@@@@@@@@@
        if (res.next()) {
            email = res.getString("information");
            // System.out.println(email);
            res1.next();
            forgottenPassword = res1.getString("password");
            // System.out.println(forgottenPassword);

        } else {

            testUsername = null;
            JOptionPane.showMessageDialog(null, "invalid username");
            throw new java.lang.NullPointerException();
        }

    } catch (java.sql.SQLException anSQLException) {
        anSQLException.printStackTrace();
    } catch (java.lang.NullPointerException ee) {
        ee.printStackTrace();
    }
}
try {

    Message message = new MimeMessage(session);
    //message.setFrom(new InternetAddress(username));

    String newTarget = email;
    //@@@@@@@@@@@@@@@@
    //String targetEmail = email;
    String passwd = forgottenPassword;
    //System.out.println(passwd);
// System.out.println("Target email address is " + newTarget);

```

```
message.setRecipients(Message.RecipientType.TO,
    InternetAddress.parse(newTarget));
message.setSubject("Testing Subject"); //subject line
message.setText("PFA");

// MimeBodyPart messageBodyPart = new MimeBodyPart();

message.setText("Your forgotten Password is: " + passwd);
//attached password

System.out.println("Sending");

Transport.send(message);

System.out.println("Done sending mail");

} catch (MessagingException anException) {
    //@@@@@@@@
    System.out.println("    In sendPassword, exception caught: ");
    System.out.println("    " + anException.getMessage());
    anException.printStackTrace();
} catch (java.lang.NullPointerException ec) {
    ec.getMessage();
}
}
}
```
