



Assignment Cover Letter

(Individual Work)

Student Information: **Surname** **Given Names** **Student ID Number**
 1. Chan Elizabeth Wirawan **2201797001**

Course Code	: COMP6510	Course Name	: Programming Language
Class	: L2AC	Name of Lecturer(s)	: 1. Minaldi Loewis 2. Jude Joseph Lamug Martinez

Major : CS

Title of Assignment : Matchmaking Database
(if any)

Type of Assignment : Final Project

Submission Pattern

Due Date : 2 July 2019 **Submission Date** : 2 July 2019

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
 2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
 3. The above information is complete and legible.
 4. Compiled pages are firmly stapled.
 5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

Plagiarism, Cheating
BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:
Chan Elizabeth Wirawan

(Name of Student)
Chan Elizabeth Wirawan

Table Of Contents

I.	Cover	1
II.	Table Of Contents	2
III.	Introduction	3
IV.	Solution Design	3
V.	Discussion	
	i. Implementation	5
	ii. How it works	5
	iii. Code Explanation	6
VI.	Result Evidence	12
VII.	References	14

Introduction

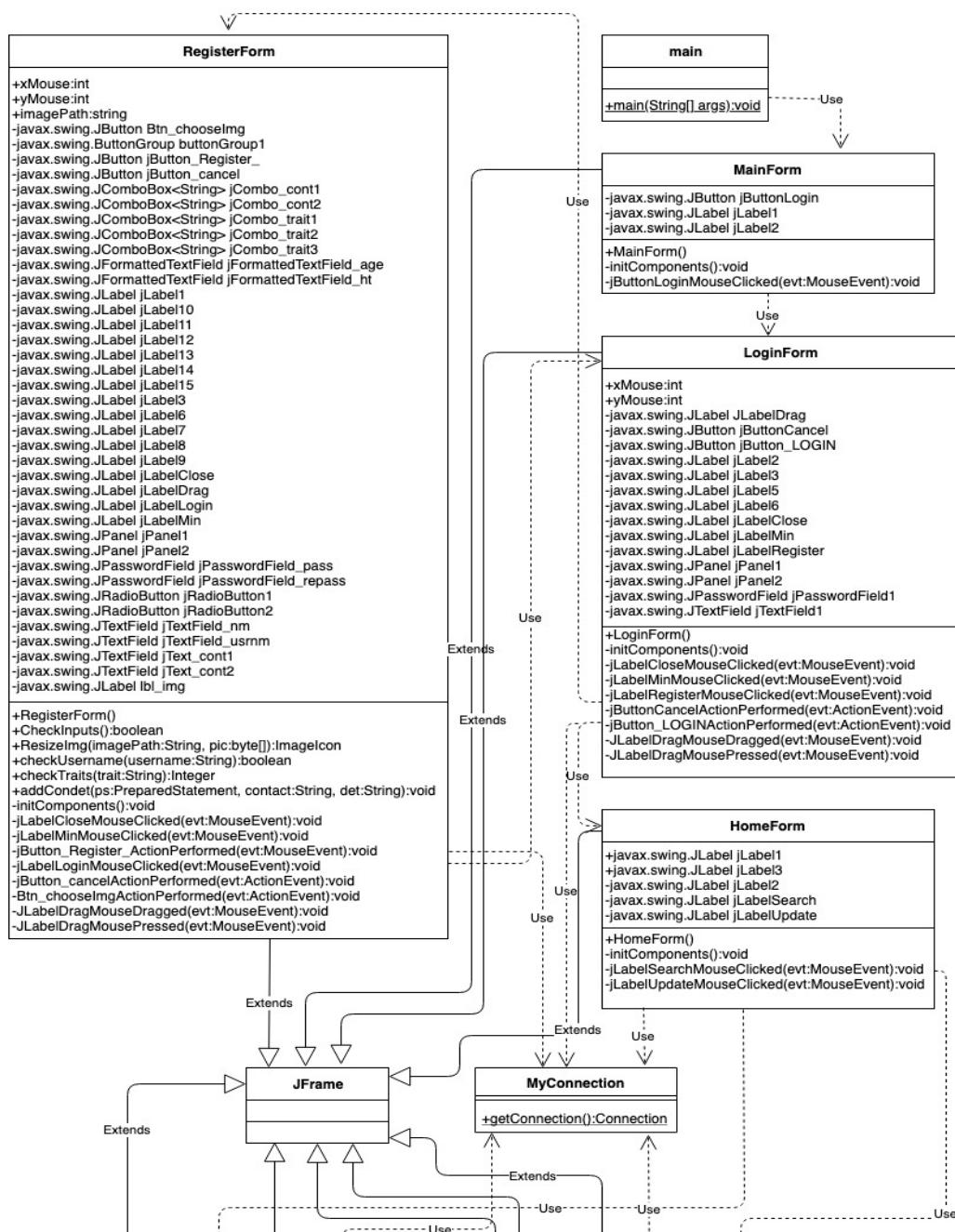
Concept:

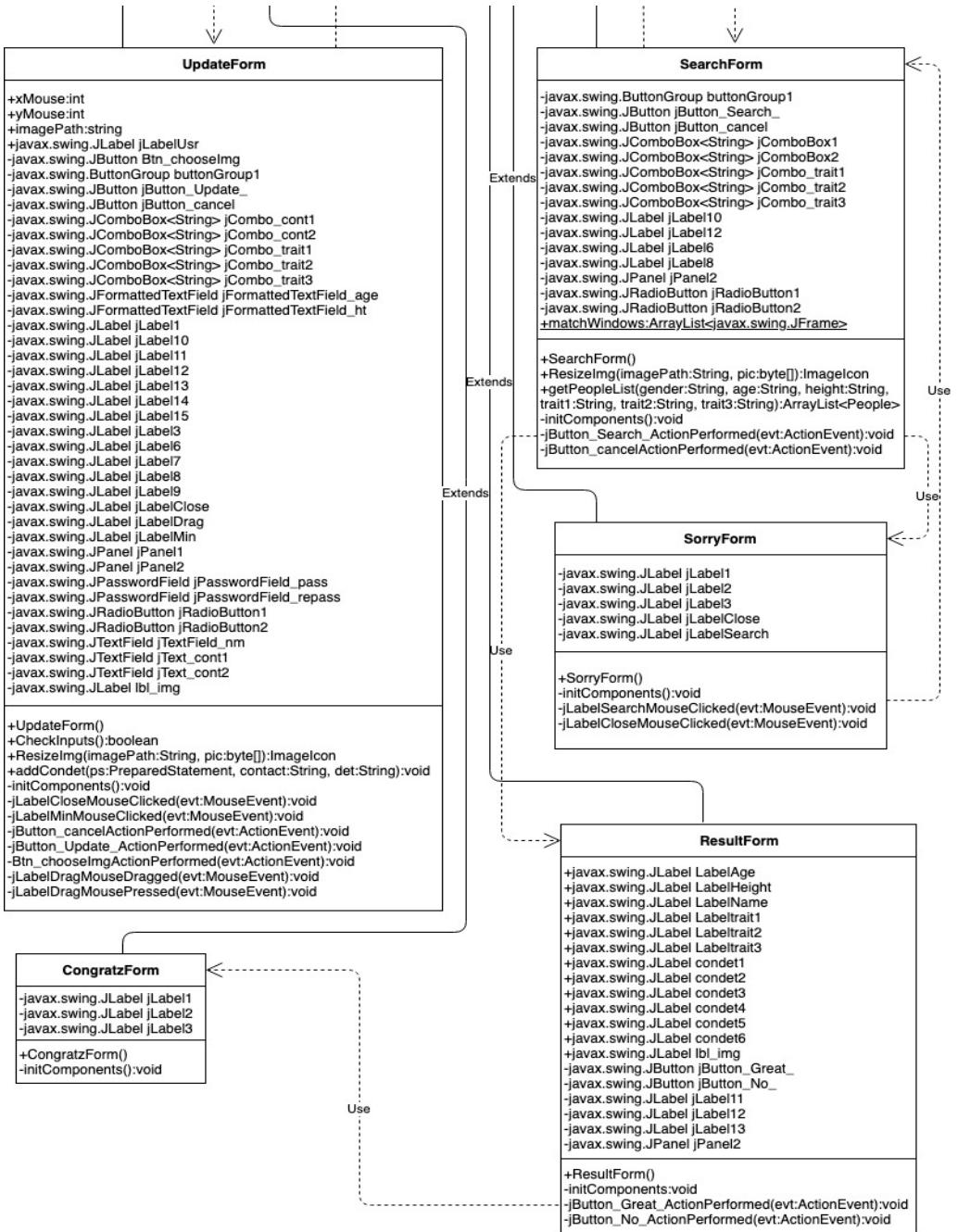
This program is meant to design matchmaking database using SQL and java programming language. The matchmaking database is a SQL-based database which store certain data of people who are looking for mate and find their ideal mates through the database with the applied criteria.

Problem (Why this app):

I want to help people to find mate that match the criteria of who they want to spend the rest of the life with. And if they are not the right one, they can still be friends.

Solution Design





Discussion

Implementation:

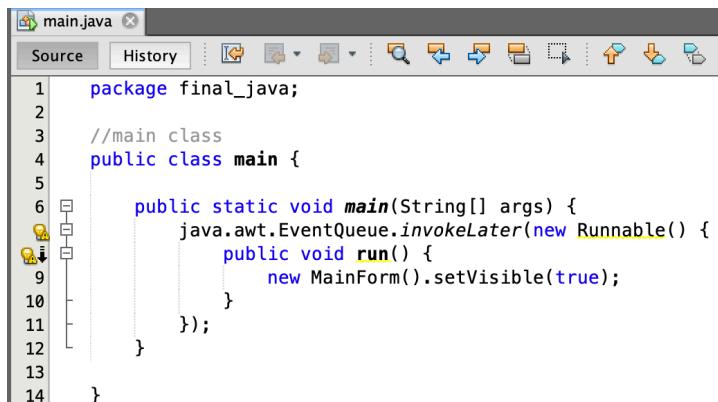
The API that is implemented in the program include: Java.sql, Java.util.logging, Javax.swing, Java.io.File, Java.text.NumberFormat and Java.awt

- Java.awt was used to make the layout and event happening on the GUI
- Java.io.File was used to list the contents of a directory and represent the pathname of the files chosen from the directory
- Java.text.NumberFormat was used for formatting and parsing numbers
- Java.util.logging was used to log messages for a specific system or application component and defines a set of standard logging levels that can be used to control logging output
- Java.sql was used to access and process data stored in the MySQL database
- Javax.swing was used to develop the GUI

How the program works:

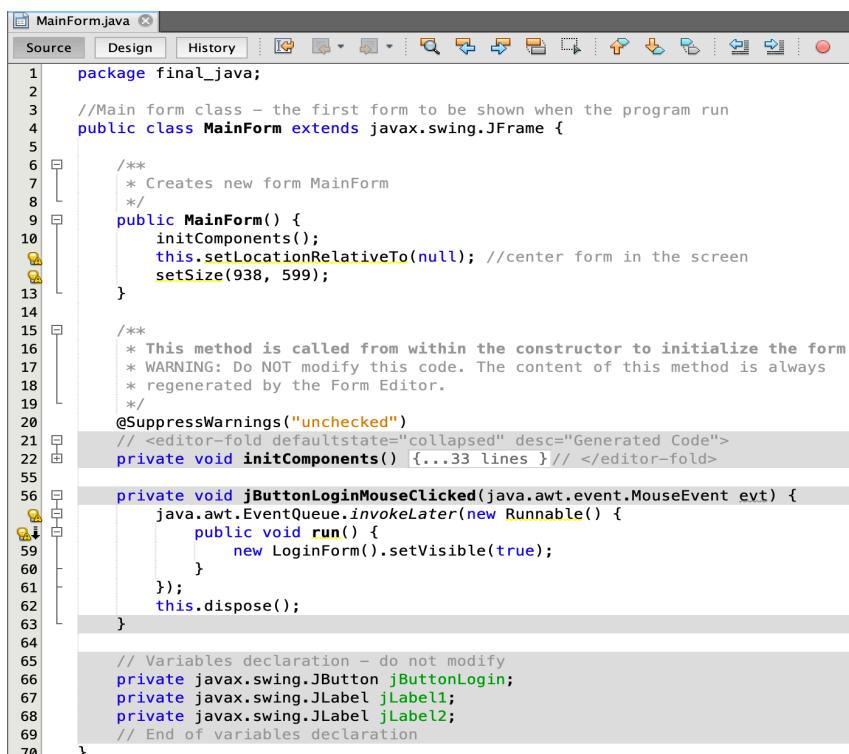
- In the main form, there will be a login button. You must click the button to move forward to the login form.
- In the login form, you are asked to input your username and password. Then, there are a cancel button and login button. The cancel button is clicked to terminate the program and Login button is click to move forward to home form after the program verify the username and password. If you are a new user, you need to register. There will also be a labelled text that can be clicked to proceed to the register form.
- In the register form, you will need to fill certain information about you and also set your username and password. Then, there are a cancel button and register button. The cancel button is clicked to terminate the program and Register button is click to register you by saving your data to the MySQL database. There also a labelled text that if clicked, the program will proceed to the registration form.
- In the home form, there are two labels where you can choose. One label to search for a partner and the other one to update your information. If you choose the search label, you will be direct to search form. If you choose the update label, you will be direct to update form.
- In the search form, you will need to fill certain criteria about the partner you want of. Then, there are a cancel button and a search button. The cancel button is clicked to terminate the program and Search button is click to search the database for a match.
- If there are a match or more with the criteria you inputted, a result form will show the information about the people that match. Then, there are a not interested button and great button. If the not interested button is clicked, it will dispose of the particular window. If the great button is clicked, congratulation form will come out.
- If there is no match, a sorry form will show. Then, there is a close button and search again button. Cancel button is clicked to terminate the program and Search again button is clicked to go back to the search form.

Code explanation:



```
main.java
Source History ...
1 package final_java;
2
3 //main class
4 public class main {
5
6     public static void main(String[] args) {
7         java.awt.EventQueue.invokeLater(new Runnable() {
8             public void run() {
9                 new MainForm().setVisible(true);
10            }
11        });
12    }
13}
14}
```

In main.java, I call the main form.



```
MainForm.java
Source Design History ...
1 package final_java;
2
3 //Main form class - the first form to be shown when the program run
4 public class MainForm extends javax.swing.JFrame {
5
6     /**
7      * Creates new form MainForm
8      */
9     public MainForm() {
10        initComponents();
11        this.setLocationRelativeTo(null); //center form in the screen
12        setSize(938, 599);
13    }
14
15    /**
16     * This method is called from within the constructor to initialize the form.
17     * WARNING: Do NOT modify this code. The content of this method is always
18     * regenerated by the Form Editor.
19     */
20    @SuppressWarnings("unchecked")
21    // <editor-fold defaultstate="collapsed" desc="Generated Code">
22    private void initComponents() {...33 lines ...} // </editor-fold>
23
24    private void jButtonLoginMouseClicked(java.awt.event.MouseEvent evt) {
25        java.awt.EventQueue.invokeLater(new Runnable() {
26            public void run() {
27                new LoginForm().setVisible(true);
28            }
29        });
30        this.dispose();
31    }
32
33    // Variables declaration - do not modify
34    private javax.swing.JButton jButtonLogin;
35    private javax.swing.JLabel jLabel1;
36    private javax.swing.JLabel jLabel2;
37    // End of variables declaration
38}
```

In MainForm.java, the class inherits javax.swing.JFrame to make the GUI form. I set the size of the form to (939, 599). I make the login button in the form and the action carry on when the button is click. The action is to proceed to the login form after all previous GUI events are processed.

The screenshot shows the Java code editor with the file `LoginForm.java` open. The code defines a class `LoginForm` that extends `javax.swing.JFrame`. It includes imports for various Java.sql and Java.util.logging packages. The constructor initializes components and sets the window size to 400x315. Several mouse event handlers are present, including `jLabelCloseMouseClicked`, `jLabelMinMouseClicked`, `jLabelRegisterMouseClicked`, `jButtonCancelActionPerformed`, `jButton_LOGINActionPerformed`, `JLabelDragMouseDragged`, and `JLabelDragMousePressed`. A generated code block is also visible.

```
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8 import javax.swing.JFrame;
9 import javax.swing.JOptionPane;
10
11 public class LoginForm extends javax.swing.JFrame {
12
13     int xMouse;
14     int yMouse;
15
16     /** Creates new form LoginForm ...3 lines */
17     public LoginForm() {
18         initComponents();
19         setSize(400, 315);
20     }
21
22
23     /** This method is called from within the constructor to initialize the form ...5 lines */
24     @SuppressWarnings("unchecked")
25     Generated Code
26
27     private void jLabelCloseMouseClicked(java.awt.event.MouseEvent evt) {
28         System.exit(0);
29     }
30
31     private void jLabelMinMouseClicked(java.awt.event.MouseEvent evt) {...3 lines}
32
33     private void jLabelRegisterMouseClicked(java.awt.event.MouseEvent evt) {...9 lines}
34
35     private void jButtonCancelActionPerformed(java.awt.event.ActionEvent evt) {...3 lines}
36
37     private void jButton_LOGINActionPerformed(java.awt.event.ActionEvent evt) {...36 lines}
38
39     private void JLabelDragMouseDragged(java.awt.event.MouseEvent evt) {...6 lines}
40
41     private void JLabelDragMousePressed(java.awt.event.MouseEvent evt) {...4 lines}
42 }
```

In `LoginForm.java`, the class inherits `javax.swing.JFrame` to make the GUI form. I set the size of the form to (400, 315). The form has its own close and minimizes button along with its function. There are also two other buttons, cancel button and login button. The cancel button is clicked to terminate the program. The `jButton_LOGINActionPerformed` function is to verify the username and password inputted with the data stored in the MySQL database when the login button is clicked. If there is a match, then proceed to the home form. If there is no match, then a warning message is shown.

The screenshot shows the Java code editor with the file `RegisterForm.java` open. The code defines a class `RegisterForm` that extends `javax.swing.JFrame`. It includes imports for various Java.awt, Java.io, and Java.sql packages. The constructor initializes components and sets the window size to 400x735. It contains methods for checking inputs, resizing images, and handling mouse events. A generated code block is also present.

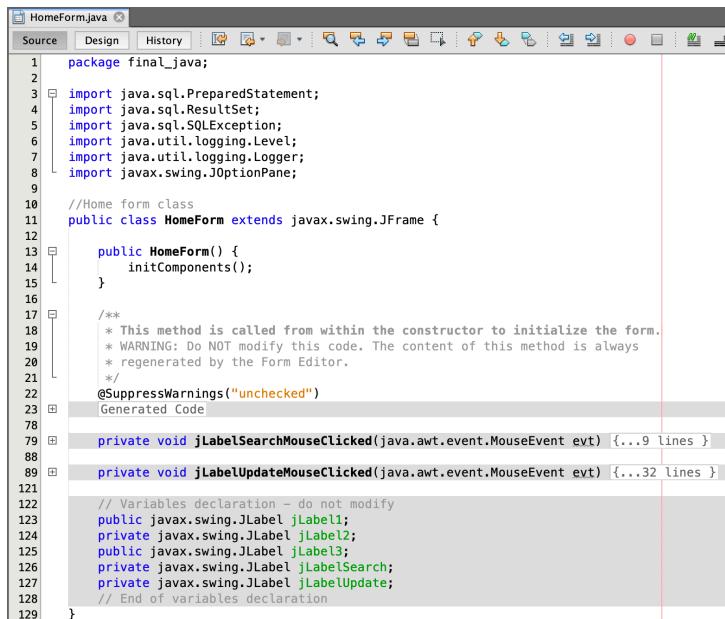
```
3 import java.awt.Image;
4 import java.io.File;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.text.NumberFormat;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javax.swing.ImageIcon;
12 import javax.swing.JFileChooser;
13 import javax.swing.JFrame;
14 import javax.swing.JOptionPane;
15 import javax.swing.filechooser.FileNameExtensionFilter;
16 import javax.swing.text.NumberFormatter;
17
18 public class RegisterForm extends javax.swing.JFrame {
19
20     /**
21      * Creates new form RegisterForm
22      */
23     public RegisterForm() {
24         initComponents();
25         setSize(400, 735);
26     }
27
28     int xMouse;
29     int yMouse;
30     String imagePath = null;
31
32     // Check Input Fields
33     public boolean CheckInputs(){...13 lines}
34
35     // Resize image
36     public ImageIcon ResizeImg(String imagePath, byte[] pic){...14 lines}
37
38     /** This method is called from within the constructor to initialize the form ...5 lines */
39     @SuppressWarnings("unchecked")
40     Generated Code
41
42     private void jLabelCloseMouseClicked(java.awt.event.MouseEvent evt) {...3 lines}
43
44     private void jLabelMinMouseClicked(java.awt.event.MouseEvent evt) {...3 lines}
45 }
```

```

402 // function to check if the username already exist
403     public boolean checkUsername(String username)
404     {...21 lines}
405
406     public void addCondet(PreparedStatement ps, String contact, String det){...26 lines}
407
408     private void jButton_cancelActionPerformed(java.awt.event.ActionEvent evt) {...3 lines}
409
410     private void Btn_chooseImgActionPerformed(java.awt.event.ActionEvent evt) {...17 lines}
411
412     private void jLabelDragMouseDragged(java.awt.event.MouseEvent evt) {...6 lines}
413
414     private void jLabelDragMousePressed(java.awt.event.MouseEvent evt) {...4 lines}

```

In RegisterForm.java, the class inherits java.swing.JFrame to make the GUI form. I set the size of the form to (400, 735). The form has its own close and minimizes button. There are also two other buttons, cancel button and register button. The cancel button is clicked to terminate the program. And when the register button is clicked, the data inputted are stored in the MySQL database. But before the data stored, there are certain functions to be verified. The CheckInputs function is to make sure the fields to be filled is not empty, else cannot proceed. The addCondet function is to add contact detail to the database. The Btn_chooseImgActionPerformed function is to choose an image from the directory and store the image path to the database. The checkUsername function checks if the username register already exists before or not. If the username already exists, there is a warning message and ask the user to register a different username. If the username is new, then the registration will be successful and moved on to the home form. There also a labelled text that if clicked, the program will proceed to the registration form.



```

1 package final_java;
2
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8 import javax.swing.JOptionPane;
9
10 //Home form class
11 public class HomeForm extends javax.swing.JFrame {
12
13     public HomeForm() {
14         initComponents();
15     }
16
17     /**
18      * This method is called from within the constructor to initialize the form.
19      * WARNING: Do NOT modify this code. The content of this method is always
20      * regenerated by the Form Editor.
21      */
22     @SuppressWarnings("unchecked")
23     // Generated Code
24
25     private void jLabelSearchMouseClicked(java.awt.event.MouseEvent evt) {...9 lines}
26
27     private void jLabelUpdateMouseClicked(java.awt.event.MouseEvent evt) {...32 lines}
28
29     // Variables declaration - do not modify
30     public javax.swing.JLabel jLabel1;
31     private javax.swing.JLabel jLabel2;
32     public javax.swing.JLabel jLabel3;
33     private javax.swing.JLabel jLabelSearch;
34     private javax.swing.JLabel jLabelUpdate;
35     // End of variables declaration

```

In HomeForm.java, the class inherits java.swing.JFrame to make the GUI form. I set the size of the form to (656,398). The form has two clickable labels, search label, and update label. If the search label is chosen, the user will be direct to the search form. If the update label is chosen, the user will be direct to update form.

```

3 import java.awt.Image;
4 import java.io.File;
5 import java.sql.PreparedStatement;
6 import java.sql.SQLException;
7 import java.text.NumberFormat;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10 import javax.swing.ImageIcon;
11 import javax.swing.JFileChooser;
12 import javax.swing.JFrame;
13 import javax.swing.JOptionPane;
14 import javax.swing.filechooser.FileNameExtensionFilter;
15 import javax.swing.text.NumberFormatter;
16
17 // Update Form class
18 public class UpdateForm extends javax.swing.JFrame {
19
20     /**
21      * Creates new form UpdateForm
22     */
23     public UpdateForm() {...3 lines}
24
25     String imagePath = null;
26
27     // Check Input Fields
28     public boolean CheckInputs(){...13 lines}
29
30     // Resize image
31     public ImageIcon ResizeImg(String imagePath, byte[] pic){...14 lines}
32
33     /**
34      * This method is called from within the constructor to initialize the form.
35      * WARNING: Do NOT modify this code. The content of this method is always
36      * regenerated by the Form Editor.
37     */
38     @SuppressWarnings("unchecked")
39     Generated Code
40
41     private void jLabelCloseMouseClicked(java.awt.event.MouseEvent evt) {...3 lines}
42
43     private void jLabelMinMouseClicked(java.awt.event.MouseEvent evt) {...3 lines}
44
45     private void jButton_cancelActionPerformed(java.awt.event.ActionEvent evt) {...3 lines}
46
47     private void jButton_Update_ActionPerformed(java.awt.event.ActionEvent evt) {...80 lines}
48
49     private void Btn_chooseImgActionPerformed(java.awt.event.ActionEvent evt) {...17 lines}
50
51     public void addCondet(PreparedStatement ps, String contact, String det){...26 lines}

```

In `UpdateForm.java`, the class inherits `java.swing.JFrame` to make the GUI form. I set the size of the form to (400, 740). The form has its own close and minimizes button. There are also two other buttons, cancel button and update button. The cancel button is clicked to cancel the search and go back to the login form. And when the update button is clicked, the updated data inputted by the user are updated in the MySQL database. The `CheckInputs` function is to make sure the fields to be filled is not empty, else cannot update. The `addCondet` function is to update contact detail to the database. The `Btn_chooseImgActionPerformed` function is to choose another image from the directory and store the image path to the database. If these certain functions are not fulfilled, then a warning message is given else the update is successful.

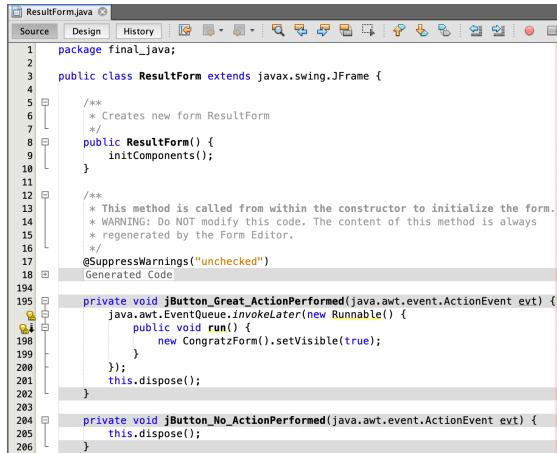
```

3 import static final java.MyConnection.getConnection;
4 import java.awt.Image;
5 import java.sql.Connection;
6 import java.sql.PreparedStatement;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.sql.Statement;
10 import java.util.ArrayList;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13 import javax.swing.ImageIcon;
14
15 //Search form class
16 public class SearchForm extends javax.swing.JFrame {
17
18     /**
19      * Creates new form SearchForm
20     */
21     public SearchForm() {
22         initComponents();
23         setSize(399, 492);
24     }
25
26     // Resize image
27     public ImageIcon ResizeImg(ResultForm rf, String imagePath, byte[] pic){...14 lines}
28
29     public ArrayList<People> getPeopleList(String gender, String age, String height, String trait1, String trait2, String trait3){...105 lines}
30
31     /**
32      * This method is called from within the constructor to initialize the form ...5 lines
33     */
34     @SuppressWarnings("unchecked")
35     Generated Code
36
37     private void jButton_SearchActionPerformed(java.awt.event.ActionEvent evt) {...72 lines}
38
39     private void jButton_cancelActionPerformed(java.awt.event.ActionEvent evt) {
40         System.exit(0);
41     }
42
43     private void radioButton2ActionPerformed(java.awt.event.ActionEvent evt) {...3 lines}
44
45     private void radioButtonActionPerformed(java.awt.event.ActionEvent evt) {...3 lines}
46
47     // Variables declaration - do not modify
48     private javax.swing.ButtonGroup buttonGroup;
49     private javax.swing.JButton jButton_Search;
50     private javax.swing.JButton jButton_Cancel;
51     private javax.swing.JComboBox<String> jComboBox1;
52     private javax.swing.JComboBox<String> jComboBox2;
53     private javax.swing.JComboBox<String> jComboBox3;
54     private javax.swing.JComboBox<String> jComboBox_trait1;
55     private javax.swing.JComboBox<String> jComboBox_trait2;
56     private javax.swing.JComboBox<String> jComboBox_trait3;
57     private javax.swing.JLabel jLabel1;
58     private javax.swing.JLabel jLabel12;
59     private javax.swing.JLabel jLabel16;
60
61 
```

In `SearchForm.java`, the class inherits `java.swing.JFrame` to make the GUI form. I set the size of the form to (399, 492). The user is asked to fill certain criteria about the partner you want of. The form has two buttons, cancel button and search button. The cancel button is clicked

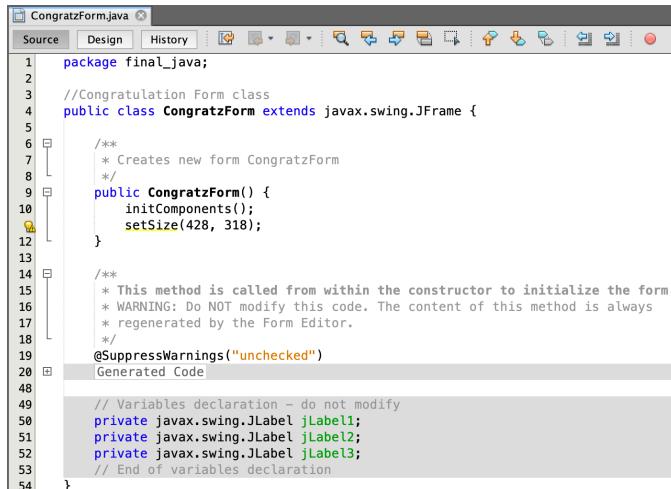
to cancel the search and go back to the login form. And when the search button is clicked, the criteria of the mate chosen by the user is compared with the data inside MySQL database.

If there are matches, then the result form will be shown. If there is no match, then the sorry form will be shown.



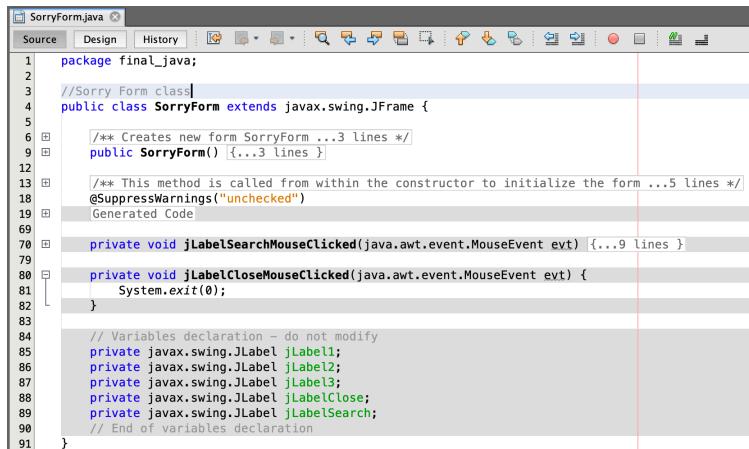
```
1 package final_java;
2
3 public class ResultForm extends javax.swing.JFrame {
4
5     /**
6      * Creates new form ResultForm
7      */
8     public ResultForm() {
9         initComponents();
10    }
11
12    /**
13     * This method is called from within the constructor to initialize the form.
14     * WARNING: Do NOT modify this code. The content of this method is always
15     * regenerated by the Form Editor.
16     */
17    @SuppressWarnings("unchecked")
18    // Generated Code
19
20    private void jButton_GreatActionPerformed(java.awt.event.ActionEvent evt) {
21        java.awt.EventQueue.invokeLater(new Runnable() {
22            public void run() {
23                new CongratzForm().setVisible(true);
24            }
25        });
26        this.dispose();
27    }
28
29    private void jButton_NoActionPerformed(java.awt.event.ActionEvent evt) {
30        this.dispose();
31    }
32}
```

In the ResultForm.java, the class inherits java.swing.JFrame to make the GUI form like the MainForm. I set the size of the form to (400, 699). This form has two buttons, “NOT INTERESTED” button and “GREAT” button. The “NOT INTERESTED” button is clicked to terminate the program. And when the “GREAT” button is clicked, the congratulation form will be shown.



```
1 package final_java;
2
3 //Congratulation Form class
4 public class CongratzForm extends javax.swing.JFrame {
5
6     /**
7      * Creates new form CongratzForm
8      */
9     public CongratzForm() {
10         initComponents();
11         setSize(428, 318);
12     }
13
14     /**
15      * This method is called from within the constructor to initialize the form.
16      * WARNING: Do NOT modify this code. The content of this method is always
17      * regenerated by the Form Editor.
18      */
19     @SuppressWarnings("unchecked")
20     // Generated Code
21
22     // Variables declaration - do not modify
23     private javax.swing.JLabel jLabel1;
24     private javax.swing.JLabel jLabel2;
25     private javax.swing.JLabel jLabel3;
26     // End of variables declaration
27 }
```

In the CongratzForm.java, the class inherits java.swing.JFrame to make the GUI form like the MainForm. I set the size of the form to (428, 318).

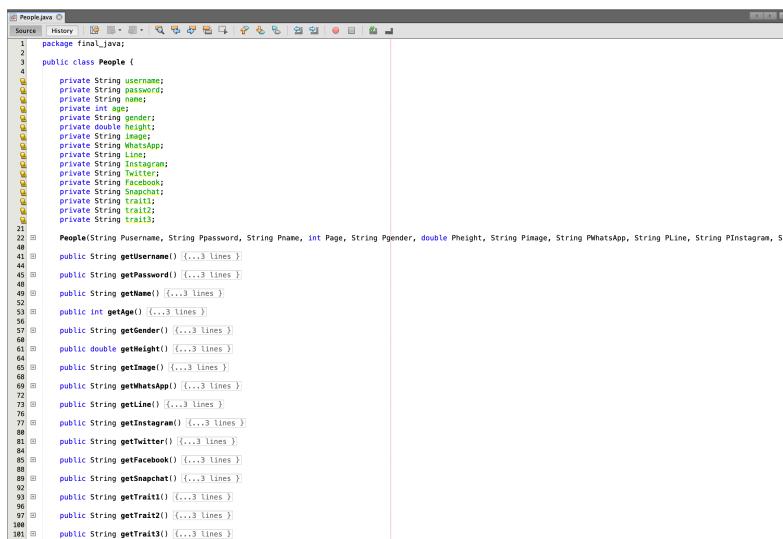


```

1 package final_java;
2
3 //Sorry Form class
4 public class SorryForm extends javax.swing.JFrame {
5
6     /** Creates new form SorryForm ...3 lines */
7     public SorryForm() {...3 lines}
8
9     /** This method is called from within the constructor to initialize the form ...5 lines */
10    @SuppressWarnings("unchecked")
11    Generated Code
12
13    private void jLabelSearchMouseClicked(java.awt.event.MouseEvent evt) {...9 lines}
14
15    private void jLabelCloseMouseClicked(java.awt.event.MouseEvent evt) {
16        System.exit(0);
17    }
18
19    // Variables declaration - do not modify
20    private javax.swing.JLabel jLabel1;
21    private javax.swing.JLabel jLabel2;
22    private javax.swing.JLabel jLabel3;
23    private javax.swing.JLabel jLabelClose;
24    private javax.swing.JLabel jLabelSearch;
25    // End of variables declaration
26
27 }

```

In the SorryForm.java, the class inherits java.swing.JFrame to make the GUI form like the MainForm. I set the size of the form to (400, 300). This form has two buttons, “close” button and “search again” button. The “close” button is clicked to terminate the program. If the search again button is chosen, the user will be direct to the search form again.

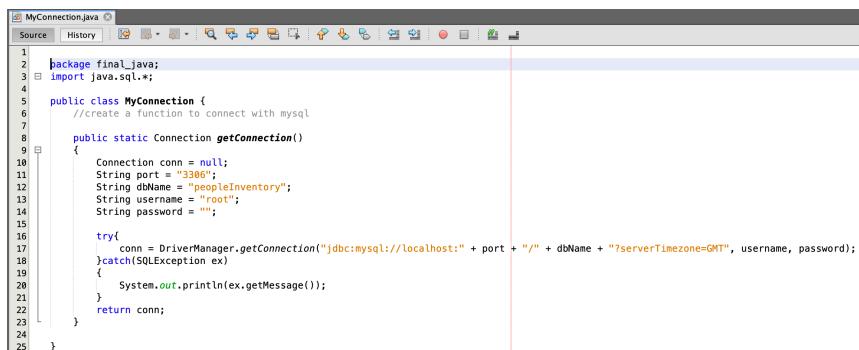


```

1 package final_java;
2
3 public class People {
4
5     private String username;
6     private String password;
7     private String name;
8     private int age;
9     private double gender;
10    private double height;
11    private String image;
12    private String WhatsApp;
13    private String Line;
14    private String Instagram;
15    private String Twitter;
16    private String Facebook;
17    private String Snapchat;
18    private String trait1;
19    private String trait2;
20    private String trait3;
21
22    People(String Pusername, String Password, String Name, int Age, double Height, String Image, String WhatsApp, String Line, String Instagram, String Twitter, String Facebook, String Snapchat, String Trait1, String Trait2, String Trait3)
23
24    public String getUsername() {...3 lines}
25    public String getPassword() {...3 lines}
26    public String getName() {...3 lines}
27    public int getAge() {...3 lines}
28    public String getGender() {...3 lines}
29    public double getHeight() {...3 lines}
30    public String getImage() {...3 lines}
31    public String getWhatsApp() {...3 lines}
32    public String getLine() {...3 lines}
33    public String getInstagram() {...3 lines}
34    public String getTwitter() {...3 lines}
35    public String getFacebook() {...3 lines}
36    public String getSnapchat() {...3 lines}
37    public String getTrait1() {...3 lines}
38    public String getTrait2() {...3 lines}
39    public String getTrait3() {...3 lines}
40
41 }

```

In the People.java, the people class contain the elements that describe the user



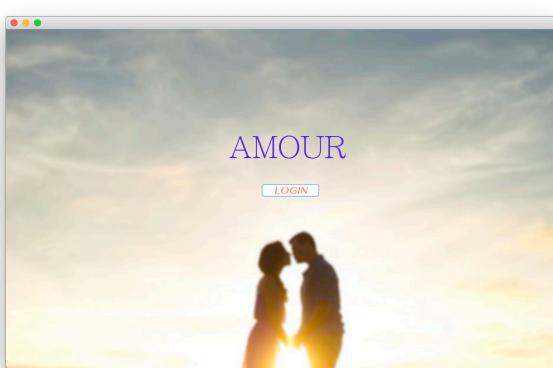
```

1 package final_java;
2
3 import java.sql.*;
4
5 public class MyConnection {
6     //create a function to connect with mysql
7
8     public static Connection getConnection()
9     {
10         Connection conn = null;
11         String port = "3306";
12         String dbName = "peopleInventory";
13         String username = "root";
14         String password = "";
15
16         try{
17             conn = DriverManager.getConnection("jdbc:mysql://localhost:" + port + "/" + dbName + "?serverTimezone=GMT", username, password);
18         }catch(SQLException ex)
19         {
20             System.out.println(ex.getMessage());
21         }
22         return conn;
23     }
24 }

```

In the MyConnection.java, the class has a function to connect java with the database and to allow easier access to the database in other classes.

Result Evidence



Login - X

Username:

Password:

[Click here to create a new account](#)

Register - X

Name:

Age:

Height: cm

Gender: female male

Personal traits:

Adve... Adve... Adve...

Contact Detail:

What...
What...

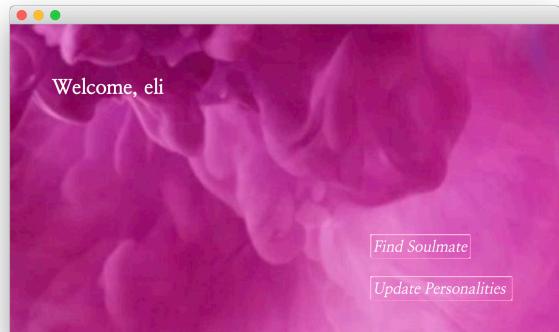
Image:

Username:

Password:

Retype Pass:

[Click here to login](#)



Update

- X

Username: eli

Name:

Age:

Height: cm

Gender: female male

Personal traits:

Adve... Adve... Adve...

Contact Detail:

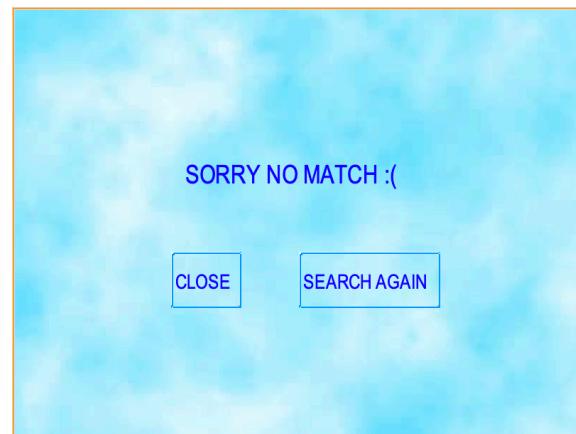
What...
 What...

Image:

Choose Image

Password:

Retype Pass:



Gender: female male

Age: 15-19

Height: 145-155

Personal traits:

Adventurous
 Adventurous
 Adventurous



References

- MySQL Commands:
<http://g2pc1.bu.edu/~qzpeng/manual/MySQL%20Commands.htm>
- Design the form: <https://www.youtube.com/watch?v=XAowXcmQ-kA>
- Create the login and register form: <https://www.youtube.com/watch?v=i5UG6ACtnEg>
- Get Selected RadioButton and CheckBox:
<https://www.youtube.com/watch?v=eyWXZ3gEfGQ>
- Set background of the form: <https://www.youtube.com/watch?v=0EZQLyfCVWQ>
- Retrieve data from MySQL to java: <https://www.youtube.com/watch?v=obR6l8fKS4c>
- Search values from MySQL database and set it into JTextField:
<https://www.youtube.com/watch?v=uuhEb0k3vVE>
- Background Image:
 1. <https://backgroundcheckall.com/light-blue-and-white-background-3/>
 2. <https://i.pinimg.com/originals/a3/76/9b/a3769b98fa6c9dee6ede74deec85de52.jpg>
 3. <https://cdn.lonerwolf.com/wp-content/uploads/2016/03/soulmate-twin-flame-kindred-spirit-signs-difference-min.jpg>
- Moving undecorated JFrame:
https://www.youtube.com/watch?time_continue=293&v=lGCkMyxriqk
- Color for JFrame : <http://www.flatuicolorpicker.com/#>