Criterion C: Development

This program is written in JAVA and SQLite. I have programmed and designed it in NetBeans also using several imported jar files. Most of the graphics is automatically programmed by the software itself so I only took care of programming the components.

Most of the following code is taken and adapted from:

https://www.youtube.com/user/ProgrammingKnowledge

http://www.codebind.com

The following libraries are taken from:

jcalendar-1.4: https://toedter.com/jcalendar/

rs2xml: https://jar-download.com/?search_box=Rs2%20xml

sqlitejdbc-v056: https://osdn.net/projects/sfnet_numerik/downloads/Jar-Files/sqlitejdbc-v056.jar/

Log In algorithm

The code above is used to create a connection with the database through checking if the username and password entered match or are existent. If not, a message saying "Username or password incorrect!" appears.

```
220     private void logInActionPerformed(java.awt.event.ActionEvent evt) {
221   String a = (String)statusSelect.getSelectedItem();
222         if (username.getText().trim().isEmpty() && password.getText().trim().isEmpty() && a.equals("Select")){
223       jLabel6.setText("No status selected.");
224       jLabel7.setText("Username is empty.");
225       jLabel8.setText("Password is empty.");
226   }//this shows an error message if any of the fields are left empty
227         else if (username.getText().trim().isEmpty() && password.getText().trim().isEmpty() ){
228       jLabel7.setText("Username is empty.");
229       jLabel8.setText("Password is empty.");
230         }
231         else if (password.getText().trim().isEmpty() && a.equals("Select")){
232       jLabel6.setText("No status selected.");
233       jLabel8.setText("Password is empty.");
234         }
235         else if (username.getText().trim().isEmpty() && a.equals("Select")){
236       jLabel6.setText("No status selected.");
237       jLabel7.setText("Username is empty.");
238         }
239         else if(username.getText().trim().isEmpty()){
240             jLabel7.setText("Username is empty.");
241         }
242         else if(password.getText().trim().isEmpty()){
243             jLabel8.setText("Password is empty.");
244         }
245         else if(a.equals("Select")){
246             jLabel6.setText("No status selected.");
247         }
```

This part of the code is used for the following reason: if any of the fields are left empty, the program displays a message in the empty fields. It does not proceed to the next function until all the fields are filled.

```
250     String un, pass;
251           un = username.getText();
252           pass = password.getText();
253           try {
254               Class.forName("org.sqlite.JDBC");
255           } catch (ClassNotFoundException ex) {
256               Logger.getLogger(CreateOpportunity.class.getName()).log(Level.SEVERE, null, ex);
257           }//connecting to the server
              Connection conn = null;
259           try{ //this part of the method connects to the database through a sql statemenent and executes it
260      String url = "jdbc:sqlite:C:\\Users\\Mario\\Documents\\NetBeansProjects\\IA\\mydatabase.sqlite";
261                conn = DriverManager.getConnection(url,"","");
262               Statement st = conn.createStatement();
263              String sql = ("SELECT*FROM Info WHERE Username='"+un+"' AND Password='"+pass+"'AND Status='"+a+"'");
264              PreparedStatement stat = conn.prepareStatement(sql);
265                ResultSet rs = stat.executeQuery();
266               conn.close();
267   if (rs.next()) { //if this statement is executed, the user logins
268           Browse br = new Browse();
269   br.setVisible(true);
270   this.setVisible(false);}
271   else if ((!username.getText().trim().isEmpty() && !password.getText().trim().isEmpty() && !a.equals("Select"))||
272           (!username.getText().trim().isEmpty() && !password.getText().trim().isEmpty())){
273       JOptionPane.showMessageDialog(null, "Username or Password are incorrect!");
274       //if the user imputs non-matchig password and username an error message appears
275   }
276          }
              catch(Exception e){
278           JOptionPane.showMessageDialog(null, e);
```

The code above is used to execute the sql query to check if the username and password match. If not, an error message appears and the user has to input data again until the information is correct.

Connection to database

```
1  import java.sql.*;
2  import javax.swing.*;
3  import java.lang.*;
4
5  public class database {
6
7
8      Connection conn = null;
9      public static Connection ConnecrDb() throws SQLException {
10         try{
11             Class.forName("org.sqlite.JDBC");
12
13         Connection conn = DriverManager.getConnection("jdbc:sqlite:mydatabase.sqlite");
14
15         return conn;
16     }
17     catch (ClassNotFoundException | SQLException e){
18     JOptionPane.showMessageDialog(null, e);
19 }
20
21         return null;
22     }
23 }
```

In the code above, I made a connection between the sql database file and my java program so I will be able to use the driver later in my code.

Create account

```
257     private void submitActionPerformed(java.awt.event.ActionEvent evt) {
258 String fname, lname, email, username, password, st, bd, qst, ans;
259             fname = FName.getText();
260             lname = setLName.getText();
261             email = setEmail.getText();
262             username = setUsername.getText();
263             password = setPass.getText();
264             st = (String)setStatus.getSelectedItem();
265             bd = ((JTextField)jDateChooser1.getDateEditor().getUiComponent()).getText();
266             qst = (String)question.getSelectedItem();
267             ans = answer.getText();
268 
269 
270 
271 if (fname.isEmpty()||lname.isEmpty()||email.isEmpty()||username.isEmpty()||
            password.equals("Select")||st.isEmpty()||bd.isEmpty()
273         ||qst.equals("Select")||ans.isEmpty()){
274     JOptionPane.showMessageDialog(null, "All fields are required!");
275 }
276 else if(!password.equals(confPass.getText())){
277     jLabell1.setText("Passwords don't match");
278 }
```

The if loop checks if any of the fields are empty and shows a message if they are not fully filled or if the confirming passwords don't match.

```
280 else{
281 
282             try {
283             Class.forName("org.sqlite.JDBC");
284         } catch (ClassNotFoundException ex) {
285             Logger.getLogger(CreateOpportunity.class.getName()).log(Level.SEVERE, null, ex);
286         }
287             Connection conn = null;
288 try{
289 
290             String url = "jdbc:sqlite:C:\\Users\\Mario\\Documents\\NetBeansProjects\\IA\\mydatabase.sqlite";
291              conn = DriverManager.getConnection(url);
292 
293             String sql = ("INSERT INTO Info (FirstName, LastName, Email, Username, Password, "
294                     + "Status, Birthday, Question, Answer) "
295                     + "VALUES ('"+fname+"', '"+lname+"', '"+email+"', '"+username+"', '"+password+"', '"
296                     +st+"', '"+bd+"', '"+qst+"', '"+ans+"')");
297 
298                   PreparedStatement preparedStmt = conn.prepareStatement(sql);
299                     preparedStmt.execute();
300 
301         JOptionPane.showMessageDialog(null, "New Account created!");
302         new LogIn().setVisible(true);
303         this.setVisible(false);
304         preparedStmt.close();
305         }
306         catch (HeadlessException | SQLException e){
307             JOptionPane.showMessageDialog(null, e);
308         }
```

If all the above conditions are met, the algorithm executes the sql query that adds all the information to a new entry in the table of the database being used.

Displaying the database table in a jTable

```
45      private void updateTable(){
46          try{
47          String sql = "SELECT * FROM Opportunities";
48          pst = conn.prepareStatement(sql);
49          rs = pst.executeQuery();
50          oppdatabase.setModel(DbUtils.resultSetToTableModel(rs));
51          }
            catch (Exception e){
53              JOptionPane.showMessageDialog(null, e);
54          }
55            finally
56      {
57        try
58        {
59            rs.close();
60            pst.close();
61          if(conn != null)
62            conn.close();
63        }
64        catch(SQLException e)
65        {
66
67          System.err.println(e);
68        }
69      }
70
71      }
```

The method above is used to display a selected table from the database in a jTable in the jFrame being used. It executes a query that gets all the data in the table and inputs it to the jTable.

Adding selected row in another table

```
328     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
329         int index = oppdatabase.getSelectedRow();
330         TableModel model = oppdatabase.getModel();
331 
332         String title = model.getValueAt(index, 0).toString();
333         String type = model.getValueAt(index, 1).toString();
334         String location = model.getValueAt(index, 3).toString();
335         String datel = model.getValueAt(index, 4).toString();
336         String date2 = model.getValueAt(index, 5).toString();
337         String description = model.getValueAt(index, 2).toString();
338         String requirements = model.getValueAt(index, 6).toString();
            String website = model.getValueAt(index, 7).toString();
340         String contact = model.getValueAt(index, 8).toString();
341         String applink = model.getValueAt(index, 9).toString();
342         String appdeadline = model.getValueAt(index, 10).toString();
343 
344             try {
345             Class.forName("org.sqlite.JDBC");
346         } catch (ClassNotFoundException ex) {
347             Logger.getLogger(CreateOpportunity.class.getName()).log(Level.SEVERE, null, ex);
348         }
              Connection conn = null;
```

```
350 try{
351 
352             String url = "jdbc:sqlite:C:\\Users\\Mario\\Documents\\NetBeansProjects\\IA\\dist\\mydatabase.sqlite";
353             //C:\\Users\\Mario\\Documents\\NetBeansProjects\\IA\\
354              conn = DriverManager.getConnection(url);
355 
356             String sql = ("INSERT INTO PersonalOpp (Title, Type, Description, Location, "
357                     + "Datel, Date2, Requirements, Website, Contact, ApplicationLink, ApplicationDeadline) "
358                     + "VALUES ('"+title+"', '"+type+"', '"+description+"', '"+location+"', '"+datel+"', '"
359                     +date2+"', '"+requirements+"', "
360                     + "'"+website+"', '"+contact+"', '"+applink+"', '"+appdeadline+"')");
361 
              PreparedStatement preparedStmt = conn.prepareStatement(sql);
363             preparedStmt.execute();
364 
365 
366         preparedStmt.close();
367         }
368         catch (HeadlessException | SQLException e){
369             JOptionPane.showMessageDialog(null, e);
370         }
```

The code above gets all the data in a selected row and adds it in another table through an sql query.

Displaying opportunities

```
407     private void oppdatabaseKeyPressed(java.awt.event.KeyEvent evt) {
408         int index = oppdatabase.getSelectedRow();
409         TableModel model = oppdatabase.getModel();
410 
411         String title = model.getValueAt(index, 0).toString();
412         String type = model.getValueAt(index, 1).toString();
413         String location = model.getValueAt(index, 3).toString();
414         String date1 = model.getValueAt(index, 4).toString();
415         String date2 = model.getValueAt(index, 5).toString();
416         String description = model.getValueAt(index, 2).toString();
417         String requirements = model.getValueAt(index, 6).toString();
418         String website = model.getValueAt(index, 7).toString();
419         String contact = model.getValueAt(index, 8).toString();
420         String applink = model.getValueAt(index, 9).toString();
421         String appdeadline = model.getValueAt(index, 10).toString();
422 
423         a.setVisible(true);
424         a.setLocationRelativeTo(null);
425 
426         a.title1.setText(title);
427         a.type1.setText(type);
428         a.location1.setText(location);
429         a.stdate1.setText(date1);
430         a.enddate1.setText(date2);
431         a.description1.setText(description);
432         a.requirements1.setText(requirements);
433         a.website1.setText(website);
434         a.contact1.setText(contact);
435         a.applink1.setText(applink);
436         a.appdl1.setText(appdeadline);
```

This code above shows how the data in a selected row can be displayed in another jFrame. This is possible through using getters to retrieve data from each cell in the row ad setters to display this data in jTextBox from another class.

Word count: 370