

```

1 package team.FixIt;
2
3 import java.sql.SQLException;
4 import java.util.logging.Level;
5 import java.util.logging.Logger;
6 import java.util.LinkedList;
7 import java.util.List;
8
9 import javafx.geometry.Insets;
10 import javafx.geometry.Pos;
11 import javafx.scene.*;
12 import javafx.scene.control.*;
13 import javafx.scene.layout.*;
14 import javafx.scene.text.Text;
15 import javafx.stage.*;
16 import team.FixIt.GUI.set;
17
18
19 public class MainDisplay
20 {
21
22     —>static boolean answer;
23     —>static String Account = "";
24     —>private static mysql MySQL;
25
26     —>private final static Logger LOGGER =
27         Logger.getLogger(MainDisplay.class.getName());
28
29     —>public static boolean choice(set set) throws SQLException
30     —>{
31         —>—>LOGGER.setLevel(Level.INFO);
32         —>—>Stage mainWindow = new Stage();
33         —>—>mainWindow.initModality(Modality.APPLICATION_MODAL);
34         —>—>mainWindow.setTitle("Main Menu");
35         —>—>GridPane grid = new GridPane();
36         —>—>////////////////////////////////////
37         —>—>Label MainMenuLabel = new Label("Main Menu");
38         —>—>MainMenuLabel.setId("Title");
39         —>—>grid.add(MainMenuLabel, 0, 2);
40         —>—>////////////////////////////////////
41         —>—>Button CreateBtn = new Button("Create Passwords");
42         —>—>HBox CreateHB_Box = new HBox(10);
43         —>—>CreateHB_Box.setAlignment(Pos.CENTER);
44         —>—>CreateHB_Box.getChildren().add(CreateBtn);
45         —>—>grid.add(CreateHB_Box, 1, 3);
46         —>—>CreateBtn.setId("MainMenuButton");
47         —>—>////////////////////////////////////
48         —>—>Button ListBtn = new Button("List passwords");
49         —>—>HBox ListHB_Box = new HBox(10);
50         —>—>ListHB_Box.setAlignment(Pos.CENTER);
51         —>—>ListHB_Box.getChildren().add(ListBtn);
52         —>—>grid.add(ListHB_Box, 1, 4);
53         —>—>ListBtn.setId("MainMenuButton");
54         —>—>////////////////////////////////////
55         —>—>Button DeleteBtn = new Button("Delete Passwords");
56         —>—>HBox DeleteHB_Box = new HBox(20);
57         —>—>DeleteHB_Box.setAlignment(Pos.CENTER);
58         —>—>DeleteHB_Box.getChildren().add>DeleteBtn);
59         —>—>grid.add>DeleteHB_Box, 1, 5);
60         —>—>DeleteBtn.setId("MainMenuButton");
61         —>—>////////////////////////////////////
62         —>—>Button LogOutBtn = new Button("Exit");
63         —>—>HBox LogOutHB_Box = new HBox(20);
64         —>—>LogOutHB_Box.setAlignment(Pos.CENTER);
65         —>—>LogOutHB_Box.getChildren().add(LogOutBtn);
66         —>—>grid.add(LogOutHB_Box, 1, 6);
67         —>—>LogOutBtn.setId("MainMenuButton");
68         —>—>////////////////////////////////////
69         —>—>CreateBtn.setOnAction(e ->
70         —>—>{
71         —>—>—>answer= false;
72         —>—>—>try

```

```

73  —————>answer = createLogin(set);
74  —————>if (answer)
75  —————>{
76  —————>mainWindow.close();
77  —————>}
78  —————>}
79  —————>catch (SQLException e1)
80  —————>{
81  —————>    >// TODO Auto-generated catch block
82  —————>    >e1.printStackTrace();
83  —————>}
84  —————>});
85  —————>ListBtn.setOnAction(e ->
86  —————>{
87  —————>    >answer= false;
88  —————>    >try
89  —————>    >{
90  —————>        >answer = listPasswords(set);
91  —————>        >if (answer)
92  —————>        >{
93  —————>            >mainWindow.close();
94  —————>        >}
95  —————>    >}
96  —————>    >catch (SQLException e1)
97  —————>    >{
98  —————>        >// TODO Auto-generated catch block
99  —————>        >e1.printStackTrace();
100 —————>    >}
101 —————>    >});
102 —————>DeleteBtn.setOnAction(e ->
103 —————>{
104 —————>    >answer= false;
105 —————>    >try
106 —————>    >{
107 —————>        >answer = deletePasswords(set);
108 —————>        >if (answer)
109 —————>        >{
110 —————>            >mainWindow.close();
111 —————>        >}
112 —————>    >}
113 —————>    >catch (SQLException e1)
114 —————>    >{
115 —————>        >// TODO Auto-generated catch block
116 —————>        >e1.printStackTrace();
117 —————>    >}
118 —————>    >});
119 —————>LogOutBtn.setOnAction(e ->
120 —————>{
121 —————>    >mainWindow.close();
122 —————>    >});
123 —————>VBox layout= new VBox(20);
124 —————>layout.getChildren().addAll(MainMenuLabel, CreateBtn, ListBtn, DeleteBtn,
    >LogOutBtn);
125 —————>layout.setAlignment(Pos.CENTER);
126 —————>Scene scene = new Scene(layout, common.HEIGHT, common.WIDTH);
127 —————>
    >scene.getStylesheets().add(GUI.class.getResource("CSS/Login.css").toExternalFo
    >rm());
128 —————>mainWindow.getIcons().add(GUI.iconImage);
129 —————>mainWindow.setScene(scene);
130 —————>mainWindow.showAndWait();
131 —————>return answer;
132 —————>}
133
134 —————>private static TilePane createTile(Button[] Btn, String[] arrayNames)
135 —————>{
136 —————>    >LOGGER.info("Running TilePane Creation");
137 —————>    >int size = arrayNames.length;
138 —————>    >TilePane tileButtons = new TilePane();
139 —————>    >tileButtons.setPadding(new Insets(10, 10, 10, 10));
140 —————>    >tileButtons.setPrefColumns(size);
141 —————>    >HBox hori = new HBox(20);
142 —————>    >for(int i = 0; i < size; i++)

```

```

143     }
144     }
145     }
146     }
147     }
148     }
149     }
150     }
151     }
152     }
153     }
154     }
155     }
156     }
157
158
159     private static boolean createLogin(set set) throws SQLException
160     {
161         }
162         }
163         }
164         }
165         }
166         }
167         }
168         }
169         }
170         }
171         }
172         }
173         }
174         }
175         }
176         }
177         }
178         }
179         }
180         }
181         }
182         }
183         }
184         }
185         }
186         }
187         }
188         }
189         }
190         }
191         }
192         }
193         }
194         }
195         }
196         }
197         }
198         }
199         }
200         }
201         }
202         }
203         }
204         }
205         }
206         }
207         }
208         }
209         }
210         }
211         }
212         }
213         }
214         }

```

```

215  —>Labels[1],
216  —>usernameField,
217  —>Labels[2],
218  —>passwordField,
219  —>Labels[3],
220  —>repasswordField,
221  —>text);
222  —>////////////////////////////////////
223  —>Label SceneTitle = new Label("Create Password");
224  —>SceneTitle.setId("Title");
225  —>TilePane tileTitle = new TilePane();
226  —>tileTitle.setPadding(new Insets(10, 10, 10, 10));
227  —>tileTitle.setPrefColumns(1);
228  —>HBox hbox = new HBox(20);
229  —>hbox.setMargin(SceneTitle, new Insets(10, 10, 10, 10));
230  —>hbox.getChildren().addAll(SceneTitle);
231  —>tileTitle.getChildren().add(hbox);
232  —>tileTitle.setAlignment(Pos.CENTER);
233  —>////////////////////////////////////
234  —>String[] arrayNames = {"Create", "Return to\nMain Menu", "Logout"};
235  —>Button[] Btn = new Button[3];
236  —>TilePane tileButtons = createTile(Btn, arrayNames);
237  —>////////////////////////////////////
238  —>BorderPane border = new BorderPane();
239  —>border.setTop(tileTitle);
240  —>border.setLeft(anchorpane);
241  —>border.setBottom(tileButtons);
242  —>////////////////////////////////////
243  —>MySQL = set.MySQL;
244  —>////////////////////////////////////
245  —>Btn[0].setOnAction(e->
246  —>{
247  —>    —>String accountString = accountField.getText();
248  —>    —>String usernameString = usernameField.getText();
249  —>    —>String passwordString = passwordField.getText();
250  —>    —>String repasswordString = repasswordField.getText();
251  —>    —>try
252  —>    —>{
253  —>        —>boolean exists = false;
254  —>        —>exists = MySQL.checkIfAccountExists("accounts", accountString);
255  —>        —>text.setText("");
256  —>        —>if(!exists)
257  —>        —>{
258  —>            —>if(usernameString.length() < 2)
259  —>            —>{
260  —>                —>text.setText("Enter a correct username");
261  —>            —>}
262  —>            —>else if(passwordString.length() < 2)
263  —>            —>{
264  —>                —>text.setText("Enter a correct password");
265  —>            —>}
266  —>            —>else if(!passwordString.equals(repasswordString))
267  —>            —>{
268  —>                —>text.setText("Passwords do not match");
269  —>            —>}
270  —>            —>else
271  —>            —>{
272  —>                —>MySQL.CreateNewLogin("accounts", accountString,
273  —>                —>                usernameString, passwordString);
274  —>                —>text.setText("Created New Login");
275  —>                —>accountField.setText("");
276  —>                —>usernameField.setText("");
277  —>                —>passwordField.setText("");
278  —>                —>repasswordField.setText("");
279  —>            —>}
280  —>            —>else
281  —>            —>{
282  —>                —>if(accountString.equals(""))
283  —>                —>{
284  —>                    —>text.setText("Please enter a account");
285  —>                —>}
286  —>            —>else

```

```

287     }
288     }>text.setText("Account Already Exists");>>>>>
289     }
290     }
291     }
292     }catch (SQLException e1)
293     {
294         // TODO Auto-generated catch block
295         e1.printStackTrace();
296     }
297     });
298     //////////////////////////////////////
299     >Btn[1].setOnAction(e->
300     {
301         >answer = false;
302         >mainWindow.close();
303     });
304     >Btn[2].setOnAction(e->
305     {
306         >answer = true;
307         >mainWindow.close();
308     });
309     >Scene mainScene = new Scene(border, common.HEIGHT, common.WIDTH);
310     >>
311         >>mainScene.getStylesheets().add(GUI.class.getResource("CSS/Login.css").toExtern
312         alForm());
313     >>mainWindow.setScene(mainScene);
314     >>mainWindow.getIcons().add(GUI.iconImage);
315     >>mainWindow.showAndWait();
316     >>return answer;
317     }
318 }
319
320 >private static boolean listPasswords(set set) throws SQLException
321 {
322     >Stage mainWindow = new Stage();
323     >mainWindow.initModality(Modality.APPLICATION_MODAL);
324     >mainWindow.setTitle("List Passwords");
325     >>////////////////////////////////////
326     >>Label SceneTitle = new Label("List Passwords");
327     >>SceneTitle.setId("Title");
328     >>TilePane tileTitle = new TilePane();
329     >>tileTitle.setPadding(new Insets(10, 10, 10, 10));
330     >>tileTitle.setPrefColumns(1);
331     >>HBox hbox = new HBox(20);
332     >>hbox.setMargin(SceneTitle, new Insets(10, 10, 10, 10));
333     >>hbox.getChildren().addAll(SceneTitle);
334     >>tileTitle.getChildren().add(hbox);
335     >>tileTitle.setAlignment(Pos.CENTER);
336     >>////////////////////////////////////
337     >>Label account = new Label("Account:");
338     >>TextField accountField = new TextField();
339     >>accountField.setId("Field");
340     >>AnchorPane anchorpane = new AnchorPane();
341     >>anchorpane.setTopAnchor(account, 50.0);
342     >>anchorpane.setLeftAnchor(account, 60.0);
343     >>anchorpane.setRightAnchor(account, 200.0);
344     >>////////////////////////////////////
345     >>MySQL = set.MySQL;
346     >>////////////////////////////////////
347     >>final ComboBox<String> accountComboBox = new ComboBox<String>();
348     >>LinkedList<String> mysqlAccounts = MySQL.PrintDetails("accounts");
349     >>for(int i = 0; i < mysqlAccounts.size(); i++)
350     {
351         >>>String value = mysqlAccounts.get(i);
352         >>>accountComboBox.getItems().add(value);
353     }
354     >>anchorpane.setTopAnchor(accountComboBox, 50.0);
355     >>anchorpane.setLeftAnchor(accountComboBox, 200.0);
356     >>anchorpane.setRightAnchor(accountComboBox, 0.0);>>>>
357     >>////////////////////////////////////
358     >>Text username = new Text();
359     >>username.setId("Output");
360     >>Text password = new Text();

```

```

358     --> password.setId("Output");
359     --> List<TextField> anchorList = new LinkedList<TextField>();
360     --> anchorpane.setTopAnchor(username, 125.0);
361     --> anchorpane.setLeftAnchor(password, 60.0);
362     --> anchorpane.setRightAnchor(username, 200.0);
363     --> anchorpane.setTopAnchor(password, 200.0);
364     --> anchorpane.setLeftAnchor(password, 60.0);
365     --> anchorpane.setRightAnchor(password, 200.0);>
366     --> username.setId("actiontarget");
367     --> password.setId("actiontarget");
368     --> anchorpane.getChildren().addAll(account, accountComboBox, username, password);
369     --> //////////////////////////////////////
370     --> String[] arrayNames = {"List", "Return to\nMain Menu", "Logout"};
371     --> Button[] Btn = new Button[3];
372     --> TilePane tileButtons = createTile(Btn, arrayNames);
373     --> //////////////////////////////////////
374     --> BorderPane border = new BorderPane();
375     --> border.setTop(tileTitle);
376     --> border.setLeft(anchorpane);
377     --> border.setBottom(tileButtons);
378     --> //////////////////////////////////////
379     --> Btn[0].setOnAction(e -> {
380     --> {
381     -->     --> System.out.print("Button List was pressed");
382     -->     --> String thisAccount = accountComboBox.getValue();
383     -->     --> try
384     -->     --> {
385     -->         --> String usernameText = MySQL.getUsername("accounts", thisAccount);
386     -->         --> username.setText(usernameText);
387     -->         --> String passwordText = MySQL.getPassword("accounts", thisAccount);
388     -->         --> password.setText(passwordText);
389     -->     --> }
390     -->     --> catch (SQLException e1)
391     -->     --> {
392     -->         --> e1.printStackTrace();
393     -->     --> }
394     --> });
395     --> //////////////////////////////////////
396     --> Btn[1].setOnAction(e -> {
397     --> {
398     -->     --> answer = false;
399     -->     --> mainWindow.close();
400     --> });
401     --> Btn[2].setOnAction(e -> {
402     --> {
403     -->     --> answer = true;
404     -->     --> mainWindow.close();
405     --> });
406     --> Scene mainScene = new Scene(border, common.HEIGHT, common.WIDTH);
407     --> mainScene.getStylesheets().add(GUI.class.getResource("CSS/Login.css").toExtern
        alForm());
408     --> mainWindow.setScene(mainScene);
409     --> mainWindow.getIcons().add(GUI.iconImage);
410     --> mainWindow.showAndWait();
411     --> return answer;
412     --> }
413
414     --> private static boolean deletePasswords(set set) throws SQLException
415     --> {
416     --> Stage mainWindow = new Stage();
417     --> mainWindow.initModality(Modality.APPLICATION_MODAL);
418     --> mainWindow.setTitle("Delete Passwords");
419     --> //////////////////////////////////////
420     --> Label SceneTitle = new Label("Delete Passwords");
421     --> SceneTitle.setId("Title");
422     --> TilePane tileTitle = new TilePane();
423     --> tileTitle.setPadding(new Insets(10, 10, 10, 10));
424     --> tileTitle.setPrefColumns(1);
425     --> HBox hbox = new HBox(20);
426     --> hbox.setMargin(SceneTitle, new Insets(10, 10, 10, 10));
427     --> hbox.getChildren().addAll(SceneTitle);
428     --> tileTitle.getChildren().add(hbox);

```

```

429  —> —> tileTitle.setAlignment(Pos.CENTER);
430  —> —> //////////////////////////////////////
431  —> —> Label account = new Label("Account:");
432  —> —> TextField accountField = new TextField();
433  —> —> accountField.setId("Field");
434  —> —> AnchorPane anchorpane = new AnchorPane();
435  —> —> anchorpane.setTopAnchor(account, 50.0);
436  —> —> anchorpane.setLeftAnchor(account, 60.0);
437  —> —> anchorpane.setRightAnchor(account, 200.0);
438  —> —> //////////////////////////////////////
439  —> —> MySQL = set.MySQL;
440  —> —> //////////////////////////////////////
441  —> —> final ComboBox<String> accountComboBox = new ComboBox<String>();
442  —> —> LinkedList<String> mysqlAccounts = MySQL.PrintDetails("accounts");
443  —> —> for(int i = 0; i < mysqlAccounts.size(); i++)
444  —> —> {
445  —> —> —> String value = mysqlAccounts.get(i);
446  —> —> —> accountComboBox.getItems().add(value);
447  —> —> —> }
448  —> —> —> anchorpane.setTopAnchor(accountComboBox, 50.0);
449  —> —> —> anchorpane.setLeftAnchor(accountComboBox, 200.0);
450  —> —> —> anchorpane.setRightAnchor(accountComboBox, 0.0); —> —>
451  —> —> —> //////////////////////////////////////
452  —> —> —> Text text = new Text();
453  —> —> —> text.setId("actiontarget");
454  —> —> —> anchorpane.setTopAnchor(text, 150.0);
455  —> —> —> anchorpane.setLeftAnchor(text, 200.0);
456  —> —> —> anchorpane.setRightAnchor(text, 0.0);
457  —> —> —> anchorpane.getChildren().addAll(account, accountComboBox, text);
458  —> —> —> //////////////////////////////////////
459  —> —> —> String[] arrayNames = {"Delete", "Return to\nMain Menu", "Logout"};
460  —> —> —> Button[] Btn = new Button[3];
461  —> —> —> TilePane tileButtons = createTile(Btn, arrayNames);
462  —> —> —> //////////////////////////////////////
463  —> —> —> BorderPane border = new BorderPane();
464  —> —> —> border.setTop(tileTitle);
465  —> —> —> border.setLeft(anchorpane);
466  —> —> —> border.setBottom(tileButtons);
467  —> —> —> //////////////////////////////////////
468  —> —> —> Btn[0].setOnAction(e ->
469  —> —> —> {
470  —> —> —> —> String thisAccount = accountComboBox.getValue();
471  —> —> —> —> if(!thisAccount.equals(null))
472  —> —> —> —> {
473  —> —> —> —> —> try
474  —> —> —> —> —> {
475  —> —> —> —> —> —> boolean deleted = MySQL.DeleteLogin(common.TABLE, thisAccount);
476  —> —> —> —> —> —> if (deleted)
477  —> —> —> —> —> —> {
478  —> —> —> —> —> —> —> text.setText("The login has been removed");
479  —> —> —> —> —> —> }
480  —> —> —> —> —> —> else
481  —> —> —> —> —> —> {
482  —> —> —> —> —> —> —> text.setText("The login was not removed");
483  —> —> —> —> —> —> }
484  —> —> —> —> —> —> }
485  —> —> —> —> —> —> catch (SQLException e1)
486  —> —> —> —> —> —> {
487  —> —> —> —> —> —> —> // TODO Auto-generated catch block
488  —> —> —> —> —> —> —> e1.printStackTrace();
489  —> —> —> —> —> —> }
490  —> —> —> —> —> —> —> //DeleteActionTarget.setText("Password Removed");
491  —> —> —> —> —> —> }
492  —> —> —> —> —> —> });
493  —> —> —> —> —> —> //////////////////////////////////////
494  —> —> —> —> —> —> Btn[1].setOnAction(e ->
495  —> —> —> —> —> —> {
496  —> —> —> —> —> —> —> answer = false;
497  —> —> —> —> —> —> —> mainWindow.close();
498  —> —> —> —> —> —> });
499  —> —> —> —> —> —> Btn[2].setOnAction(e ->
500  —> —> —> —> —> —> {
501  —> —> —> —> —> —> —> answer = true;

```

```
502  —————>mainWindow.close();
503  —————>});
504  —————>Scene mainScreen = new Scene(border, common.HEIGHT, common.WIDTH);
505  —————>
        mainScreen.getStylesheets().add(GUI.class.getResource("CSS/Login.css").toExternalForm());
506  —————>mainWindow.setScene(mainScene);
507  —————>mainWindow.getIcons().add(GUI.iconImage);
508  —————>mainWindow.showAndWait();
509  —————>return answer;
510  —————>}
511
512 }
513
```