

```

1  package application;
2
3  import javafx.application.Application;
4  import javafx.stage.Screen;
5  import javafx.stage.Stage;
6  import javafx.scene.layout.BorderPane;
7  import javafx.scene.Parent;
8  import javafx.scene.Scene;
9  import javafx.scene.image.Image;
10 import javafx.fxml.FXMLLoader;
11 import javafx.geometry.Rectangle2D;
12
13
14 public class Main extends Application {
15
16     private static Stage stage;
17
18     //method to open Stage
19     @Override
20     public void start(Stage primaryStage) {
21
22         try {
23
24             stage = primaryStage;
25             primaryStage.setResizable(false);
26             BorderPane root =
27                 (BorderPane)FXMLLoader.load(getClass().getResource("LoginPage.fxml"));
28             primaryStage.setTitle("Infirmary Management System");
29             Scene scene = new Scene(root,800,600);
30             primaryStage.getIcons().add( new Image("LogoIcon.png"));
31             primaryStage.setScene(scene);
32             primaryStage.show();
33
34         }
35         catch(Exception e) {
36             e.printStackTrace();
37         }
38     }
39
40     //method to change between different fxml files
41     public void changeScene(String fxml) {
42
43         try {
44             //making other fxml files non resizable and displaying it in the centre of the screen
45             stage.setResizable(false);
46             stage.setHeight(637.5999755859375);
47             stage.setWidth(814.4000244140625);
48             Rectangle2D primScreenBounds = Screen.getPrimary().getVisualBounds();
49             stage.setX((primScreenBounds.getWidth() - stage.getWidth()) / 2);
50             stage.setY((primScreenBounds.getHeight() - stage.getHeight()) / 2);
51             Parent pane = FXMLLoader.load(getClass().getResource(fxml));
52             stage.setScene().setRoot(pane);
53
54         }
55         catch (Exception e) {
56             e.printStackTrace();
57         }
58     }
59 }
60
61 //method to open fxml files that contains tables to show records
62 public void showPatientRecords(String fxml) {
63

```

```
64     try {
65         Parent pane = FXMLLoader.load(getClass().getResource(fxml));
66         stage.getScene().setRoot(pane);
67         stage.setResizable(true);
68     }
69
70     catch (Exception e) {
71         e.printStackTrace();
72     }
73
74 }
75
76 //main method
77 public static void main(String[] args) {
78     launch(args);
79 }
80 }
81
```

```

1  package application;
2
3  import javafx.fxml.FXML;
4  import javafx.scene.control.*;
5  import javafx.event.ActionEvent;
6  import java.io.IOException;
7  import java.sql.*;
8
9  public class LoginPageController {
10
11      @FXML
12      private Button loginButton;
13      @FXML
14      private TextField usernameTextField;
15      @FXML
16      private PasswordField passwordField;
17
18      public void userLogin (ActionEvent event) throws IOException {
19          checkLogin();
20      }
21
22      //gets Username and Password from database and matches with the credentials entered
23      private void checkLogin() throws IOException {
24
25          Main m = new Main();
26
27          try {
28
29              Connection connection =
30                  DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarary_management_system"
31                      , "root", "root");
32              Statement statement = connection.createStatement();
33
34              String sql = "SELECT * from login_details where
35                  username='"+usernameTextField.getText().toString()+"' and
36                  password='"+passwordField.getText().toString()+"'";
37              ResultSet resultset = statement.executeQuery(sql);
38
39              if (resultset.next()) {
40
41                  Alert alert = new Alert(Alert.AlertType.INFORMATION);
42                  alert.setTitle("Login Message");
43                  alert.setHeaderText(null);
44                  alert.setContentText("Welcome, You have successfully logged in!");
45                  alert.showAndWait();
46
47                  m.changeScene("HomePage.fxml");
48              }
49
50              else if (usernameTextField.getText().isEmpty() &&
51                  passwordField.getText().isEmpty()) {
52
53                  Alert alert = new Alert(Alert.AlertType.ERROR);
54                  alert.setTitle("Login Message");
55                  alert.setHeaderText(null);
56                  alert.setContentText("Please Enter Credentials!!");
57                  alert.showAndWait();
58
59              }
60
61              else {
62
63                  Alert alert = new Alert(Alert.AlertType.WARNING);
64                  alert.setTitle("Login Message");

```

```
60         alert.setHeaderText(null);
61         alert.setContentText("Incorrect Credentials!!");
62         alert.showAndWait();
63
64     }
65
66     resultSet.close();
67     connection.close();
68
69 }
70
71 catch (SQLException ae) {
72
73     System.out.println("Error!!! Connection Failed");
74     ae.printStackTrace();
75
76     Alert alert = new Alert(Alert.AlertType.ERROR);
77     alert.setTitle("Error Message");
78     alert.setHeaderText(null);
79     alert.setContentText("ERROR Connecting to MySQL Database!!");
80     alert.showAndWait();
81
82 }
83
84
85
86 //login without using database
87 /* if (usernameTextField.getText().toString().equals("admin") &&
88 passwordField.getText().toString().equals("admin")) {
89     messageLabel.setText("Login Successful!!");
90     m.changeScene("HomePage.fxml");
91 }
92
93 else if (usernameTextField.getText().isEmpty() && passwordField.getText().isEmpty()) {
94     messageLabel.setText("Please Enter Credentials!!");
95 }
96
97 else {
98     messageLabel.setText("Incorrect Credentials!!");
99 } */
100 }
101
102 }
103
```

```

1  package application;
2
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXML;
5  import javafx.scene.control.Alert;
6  import javafx.scene.control.Button;
7  import javafx.scene.control.ButtonType;
8
9  import java.io.IOException;
10
11 public class HomePageController {
12
13     @FXML
14     private Button logoutButton;
15
16     @FXML
17     private Button newPatientButton;
18
19     @FXML
20     private Button patientRecordsButton;
21
22     @FXML
23     private Button medicineStockButton;
24
25     //go back to login page with alert popup
26     public void userLogout(ActionEvent event) throws IOException {
27
28         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
29         alert.setTitle("Logout");
30         alert.setHeaderText(null);
31         alert.setContentText("Are you sure you want to Logout?");
32         if (alert.showAndWait().get() == ButtonType.OK){
33             Main m = new Main();
34             m.changeScene("LoginPage.fxml");
35         }
36
37     }
38
39     //changing different scenes
40     public void patientDetailsEntry(ActionEvent event) throws IOException {
41
42         Main m = new Main();
43         m.changeScene("SelectPatientEntryPage.fxml");
44
45     }
46
47     public void patientRecords(ActionEvent event) throws IOException {
48
49         Main m = new Main();
50         m.changeScene("SelectPatientRecordsPage.fxml");
51
52     }
53
54     public void medicineStock(ActionEvent event) throws IOException {
55
56         Main m = new Main();
57         m.changeScene("MedicineRecords.fxml");
58
59     }
60
61     //MenuBar
62
63     //Menu 1 - File
64

```

```
65 //Menu Item 1.1
66 public void HomePageMenuItem(ActionEvent event) {
67     MenuBar obj = new MenuBar();
68     obj.HomePage();
69 }
70
71 //Menu Item 1.2
72 public void LogOutMenuItem(ActionEvent event) {
73     MenuBar obj = new MenuBar();
74     obj.LogOut();
75 }
76
77 //Menu Item 1.3
78 public void ExitMenuItem(ActionEvent event) {
79     MenuBar obj = new MenuBar();
80     obj.Exit();
81 }
82
83 //Menu 2 - View
84
85 //Menu Item 2.1
86 public void NewPatientEntryMenuItem(ActionEvent event) {
87     MenuBar obj = new MenuBar();
88     obj.NewPatientEntry();
89 }
90
91 //Menu Item 2.2
92 public void ShowPatientRecordMenuItem(ActionEvent event) {
93     MenuBar obj = new MenuBar();
94     obj.ShowPatientRecord();
95 }
96
97 //Menu Item 2.3
98 public void MedicineRecordsMenuItem(ActionEvent event) {
99     MenuBar obj = new MenuBar();
100     obj.MedicineRecords();
101 }
102
103 //Menu 3 - Help
104
105 //Menu Item 3.1
106 public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
107     MenuBar obj = new MenuBar();
108     obj.CollegeWebsiteLinks();
109 }
110
111 //Menu Item 3.2
112 public void AboutMenuItem(ActionEvent event) {
113     MenuBar obj = new MenuBar();
114     obj.About();
115 }
116
117 }
118
```

```
1  package application;
2
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXML;
5  import javafx.scene.control.Button;
6
7  import java.io.IOException;
8
9  public class SelectPatientEntryPageController {
10
11      @FXML
12      private Button StaffButton;
13
14      @FXML
15      private Button StudentsButton;
16
17      @FXML
18      private Button OthersButton;
19
20      @FXML
21      private Button BackButton;
22
23
24      public void userBack(ActionEvent event) throws IOException {
25
26          Main m = new Main();
27          m.changeScene("HomePage.fxml");
28
29      }
30
31      public void staffDetailsEntry(ActionEvent event) throws IOException {
32
33          Main m = new Main();
34          m.changeScene("StaffPatientDetailsEntry.fxml");
35
36      }
37
38      public void studentDetailsEntry(ActionEvent event) throws IOException {
39
40          Main m = new Main();
41          m.showPatientRecords("PatientDetailsEntry.fxml");
42
43      }
44
45      public void OthersDetailsEntry(ActionEvent event) throws IOException {
46
47          Main m = new Main();
48          m.changeScene("OtherPatientDetailsEntry.fxml");
49
50      }
51
52      //MenuBar
53
54      //Menu 1 - File
55
56      //Menu Item 1.1
57      public void HomePageMenuItem(ActionEvent event) {
58          MenuBar obj = new MenuBar();
59          obj.HomePage();
60      }
61
62      //Menu Item 1.2
63      public void LogOutMenuItem(ActionEvent event) {
64          MenuBar obj = new MenuBar();
```

```
65         obj.LogOut();
66     }
67
68     //Menu Item 1.3
69     public void ExitMenuItem(ActionEvent event) {
70         MenuBar obj = new MenuBar();
71         obj.Exit();
72     }
73
74     //Menu 2 - View
75
76     //Menu Item 2.1
77     public void NewPatientEntryMenuItem(ActionEvent event) {
78         MenuBar obj = new MenuBar();
79         obj.NewPatientEntry();
80     }
81
82     //Menu Item 2.2
83     public void ShowPatientRecordMenuItem(ActionEvent event) {
84         MenuBar obj = new MenuBar();
85         obj.ShowPatientRecord();
86     }
87
88     //Menu Item 2.3
89     public void MedicineRecordsMenuItem(ActionEvent event) {
90         MenuBar obj = new MenuBar();
91         obj.MedicineRecords();
92     }
93
94     //Menu 3 - Help
95
96     //Menu Item 3.1
97     public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
98         MenuBar obj = new MenuBar();
99         obj.CollegeWebsiteLinks();
100    }
101
102    //Menu Item 3.2
103    public void AboutMenuItem(ActionEvent event) {
104        MenuBar obj = new MenuBar();
105        obj.About();
106    }
107
108 }
109
```



```
1  package application;
2
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXML;
5  import javafx.scene.control.Button;
6
7  import java.io.IOException;
8
9  public class SelectPatientRecordsPageController {
10
11      @FXML
12      private Button StaffButton;
13
14      @FXML
15      private Button StudentsButton;
16
17      @FXML
18      private Button OthersButton;
19
20      @FXML
21      private Button BackButton;
22
23
24      public void userBack(ActionEvent event) throws IOException {
25
26          Main m = new Main();
27          m.changeScene("HomePage.fxml");
28
29      }
30
31      public void staffRecords(ActionEvent event) throws IOException {
32
33          Main m = new Main();
34          m.showPatientRecords("StaffPatientRecords.fxml");
35
36      }
37
38      public void studentRecords(ActionEvent event) throws IOException {
39
40          Main m = new Main();
41          m.showPatientRecords("PatientRecords.fxml");
42
43      }
44
45      public void OthersRecords(ActionEvent event) throws IOException {
46
47          Main m = new Main();
48          m.showPatientRecords("OtherPatientRecords.fxml");
49
50      }
51
52      //MenuBar
53
54      //Menu 1 - File
55
56      //Menu Item 1.1
57      public void HomePageMenuItem(ActionEvent event) {
58          MenuBar obj = new MenuBar();
59          obj.HomePage();
60      }
61
62      //Menu Item 1.2
63      public void LogOutMenuItem(ActionEvent event) {
64          MenuBar obj = new MenuBar();
```

```
65         obj.LogOut();
66     }
67
68     //Menu Item 1.3
69     public void ExitMenuItem(ActionEvent event) {
70         MenuBar obj = new MenuBar();
71         obj.Exit();
72     }
73
74     //Menu 2 - View
75
76     //Menu Item 2.1
77     public void NewPatientEntryMenuItem(ActionEvent event) {
78         MenuBar obj = new MenuBar();
79         obj.NewPatientEntry();
80     }
81
82     //Menu Item 2.2
83     public void ShowPatientRecordMenuItem(ActionEvent event) {
84         MenuBar obj = new MenuBar();
85         obj.ShowPatientRecord();
86     }
87
88     //Menu Item 2.3
89     public void MedicineRecordsMenuItem(ActionEvent event) {
90         MenuBar obj = new MenuBar();
91         obj.MedicineRecords();
92     }
93
94     //Menu 3 - Help
95
96     //Menu Item 3.1
97     public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
98         MenuBar obj = new MenuBar();
99         obj.CollegeWebsiteLinks();
100    }
101
102    //Menu Item 3.2
103    public void AboutMenuItem(ActionEvent event) {
104        MenuBar obj = new MenuBar();
105        obj.About();
106    }
107
108 }
109
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.sql.Connection;
5  import java.sql.DriverManager;
6  import java.sql.PreparedStatement;
7  import java.sql.ResultSet;
8  import java.sql.SQLException;
9  import java.sql.Statement;
10 import java.time.LocalDate;
11
12 import javafx.collections.FXCollections;
13 import javafx.collections.ObservableList;
14 import javafx.event.ActionEvent;
15 import javafx.fxml.FXML;
16 import javafx.scene.control.Alert;
17 import javafx.scene.control.Button;
18 import javafx.scene.control.DatePicker;
19 import javafx.scene.control.TableColumn;
20 import javafx.scene.control.TableView;
21 import javafx.scene.control.TextField;
22
23 public class MedicineRecordsController {
24
25
26     @FXML
27     private TextField medicineNameTextField;
28
29     @FXML
30     private DatePicker medicineExpiryDateSelector;
31
32     @FXML
33     private TextField medicineQuantityTextField;
34
35     @FXML
36     private Button medicineSaveButton;
37
38     @FXML
39     private TextField injectionNameTextField;
40
41     @FXML
42     private DatePicker injectionExpiryDateSelector;
43
44     @FXML
45     private TextField injectionQuantityTextField;
46
47     @FXML
48     private Button injectionSaveButton;
49
50     //Importing table and columns
51     @FXML
52     private TableView<Medicine> medicineTable;
53
54     @FXML
55     private TableColumn<Medicine, String> col_medicineName;
56
57     @FXML
58     private TableColumn<Medicine, Double> col_medicineQuantity;
59
60     @FXML
61     private TableColumn<Medicine, String> col_medicineExpiryDate;
62
63     @FXML
64     private TableView<Injection> injectionTable;
```

```

65
66     @FXML
67     private TableColumn<Injection, String> col_injectionName;
68
69     @FXML
70     private TableColumn<Injection, Double> col_injectionQuantity;
71
72     @FXML
73     private TableColumn<Injection, String> col_injectionExpiryDate;
74
75     @FXML
76     private Button backButton;
77
78     @FXML
79     private Button medicineUpdateStockButton;
80
81     @FXML
82     private Button deleteMedicine;
83
84     @FXML
85     private Button injectionUpdateStockButton;
86
87     @FXML
88     private Button deleteInjection;
89
90     LocalDate medicine_expiryDate, injection_expiryDate;
91
92     public void backToHomePage(ActionEvent event) throws IOException{
93
94         Main m = new Main();
95         m.changeScene("HomePage.fxml");
96
97     }
98
99     public void showMedicineDate(ActionEvent event) throws IOException {
100
101         medicine_expiryDate = medicineExpiryDateSelector.getValue();
102
103     }
104
105     public void showInjectionDate(ActionEvent event) throws IOException {
106
107         injection_expiryDate = injectionExpiryDateSelector.getValue();
108
109     }
110
111     //save new medicine to the database
112     public void saveMedicine(ActionEvent event) throws IOException{
113
114         try {
115
116             Connection connection =
117                 DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarary_management_system"
118                     , "root", "root");
119
120             String sql = "INSERT INTO medicine_list (medicine_name, medicine_quantity,
121                 medicine_expiry_date) VALUES (?, ?, ?)";
122
123             PreparedStatement statement = connection.prepareStatement(sql);
124
125             statement.setString(1, medicineNameTextField.getText());
126             statement.setDouble(2, Double.parseDouble(medicineQuantityTextField.getText()));
127             statement.setString(3, medicine_expiryDate.toString());

```

```

126         int rowsInserted = statement.executeUpdate();
127
128         if (rowsInserted > 0) {
129             Alert alert = new Alert(Alert.AlertType.INFORMATION);
130             alert.setTitle("Message");
131             alert.setHeaderText(null);
132             alert.setContentText("Saved Successfully!");
133             alert.showAndWait();
134         }
135
136         Main m = new Main();
137         m.changeScene("MedicineRecords.fxml");
138
139         statement.close();
140         connection.close();
141
142     }
143
144     catch (SQLException ae) {
145
146         System.out.println("Error!!! Connection Failed");
147         ae.printStackTrace();
148
149         Alert alert = new Alert(Alert.AlertType.ERROR);
150         alert.setTitle("Error Message");
151         alert.setHeaderText(null);
152         alert.setContentText("ERROR Saving Medicine!!");
153         alert.showAndWait();
154
155     }
156
157 }
158
159 //update medicine stock
160 public void updateMedicine(ActionEvent event) throws IOException {
161
162     try {
163
164         Connection connection =
165             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
166                 , "root", "root");
167
168         String sql = "UPDATE medicine_list SET medicine_quantity=medicine_quantity+?,
169             medicine_expiry_date=? WHERE medicine_name=?";
170
171         PreparedStatement statement = connection.prepareStatement(sql);
172
173         statement.setDouble(1, Double.parseDouble(medicineQuantityTextField.getText()));
174         statement.setString(2, medicine_expiryDate.toString());
175         statement.setString(3, medicineNameTextField.getText());
176
177         int rowsUpdated = statement.executeUpdate();
178
179         if (rowsUpdated > 0) {
180             Alert alert = new Alert(Alert.AlertType.INFORMATION);
181             alert.setTitle("Message");
182             alert.setHeaderText(null);
183             alert.setContentText("Saved Successfully!");
184             alert.showAndWait();
185         }
186
187         Main m = new Main();
188         m.changeScene("MedicineRecords.fxml");

```

```

187
188         statement.close();
189         connection.close();
190
191     }
192     catch (SQLException ae) {
193
194         System.out.println("Error!!! Connection Failed");
195         ae.printStackTrace();
196
197         Alert alert = new Alert(Alert.AlertType.ERROR);
198         alert.setTitle("Error Message");
199         alert.setHeaderText(null);
200         alert.setContentText("ERROR Updating Stock!!");
201         alert.showAndWait();
202
203     }
204
205 }
206
207 //delete medicine records
208 public void deleteMedicine(ActionEvent event) throws IOException{
209
210     try {
211
212         Medicine selectedRow = medicineTable.getSelectionModel().getSelectedItem();
213         String medicineName = selectedRow.getMedicineName();
214
215         Connection connection =
216             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
217                 , "root", "root");
218
219         String sql = "DELETE FROM medicine_list WHERE medicine_name = ?";
220
221         PreparedStatement statement = connection.prepareStatement(sql);
222
223         statement.setString(1, medicineName);
224         statement.execute();
225
226         new Main().changeScene("MedicineRecords.fxml");
227
228     }
229     catch (SQLException ae) {
230
231         System.out.println("Error!!! Connection Failed");
232         ae.printStackTrace();
233
234         Alert alert = new Alert(Alert.AlertType.ERROR);
235         alert.setTitle("Error Message");
236         alert.setHeaderText(null);
237         alert.setContentText("ERROR Deleting Selected Row!!");
238         alert.showAndWait();
239
240     }
241
242 }
243
244 //save new injection
245 public void saveInjection(ActionEvent event) throws IOException{
246
247     try {
248
249         Connection connection =
250             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"

```

```

248         , "root", "root");
249
250     String sql = "INSERT INTO injection_list (injection_name, injection_quantity,
251     injection_expiry_date) VALUES (?, ?, ?)";
252
253     PreparedStatement statement = connection.prepareStatement(sql);
254
255     statement.setString(1, injectionNameTextField.getText());
256     statement.setDouble(2, Double.parseDouble(injectionQuantityTextField.getText()));
257     statement.setString(3, injection_expiryDate.toString());
258
259     int rowsInserted = statement.executeUpdate();
260
261     if (rowsInserted > 0) {
262         Alert alert = new Alert(Alert.AlertType.INFORMATION);
263         alert.setTitle("Message");
264         alert.setHeaderText(null);
265         alert.setContentText("Saved Successfully!");
266         alert.showAndWait();
267     }
268
269     Main m = new Main();
270     m.changeScene("MedicineRecords.fxml");
271
272     statement.close();
273     connection.close();
274
275 }
276
277 catch (SQLException ae) {
278
279     System.out.println("Error!!! Connection Failed");
280     ae.printStackTrace();
281
282     Alert alert = new Alert(Alert.AlertType.ERROR);
283     alert.setTitle("Error Message");
284     alert.setHeaderText(null);
285     alert.setContentText("ERROR Saving Injection!!");
286     alert.showAndWait();
287
288 }
289
290 }
291
292 //update injection info
293 public void updateInjection(ActionEvent event) throws IOException {
294
295     try {
296
297         Connection connection =
298         DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
299         , "root", "root");
300
301         String sql = "UPDATE injection_list SET injection_quantity=injection_quantity+?,
302         injection_expiry_date=? WHERE injection_name=?";
303
304         PreparedStatement statement = connection.prepareStatement(sql);
305
306         statement.setDouble(1, Double.parseDouble(injectionQuantityTextField.getText()));
307         statement.setString(2, injection_expiryDate.toString());
308         statement.setString(3, injectionNameTextField.getText());
309
310         int rowsUpdated = statement.executeUpdate();
311
312         if (rowsUpdated > 0) {

```

```

307         Alert alert = new Alert(Alert.AlertType.INFORMATION);
308         alert.setTitle("Message");
309         alert.setHeaderText(null);
310         alert.setContentText("Saved Successfully!");
311         alert.showAndWait();
312     }
313
314     Main m = new Main();
315     m.changeScene("MedicineRecords.fxml");
316
317     statement.close();
318     connection.close();
319
320 }
321 catch (SQLException ae) {
322
323     System.out.println("Error!!! Connection Failed");
324     ae.printStackTrace();
325
326     Alert alert = new Alert(Alert.AlertType.ERROR);
327     alert.setTitle("Error Message");
328     alert.setHeaderText(null);
329     alert.setContentText("ERROR Updating Stock!!");
330     alert.showAndWait();
331
332 }
333 }
334
335 //delete Injection reocrds in database
336 public void deleteInjection(ActionEvent event) throws IOException{
337
338     try {
339
340         Injection selectedRow = injectionTable.getSelectionModel().getSelectedItem();
341         String injectionName = selectedRow.getInjectionName();
342
343         Connection connection =
344             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarary_management_system"
345                 , "root", "root");
346
347         String sql = "DELETE FROM injection_list WHERE injection_name = ?";
348
349         PreparedStatement statement = connection.prepareStatement(sql);
350
351         statement.setString(1, injectionName);
352         statement.execute();
353
354         new Main().changeScene("MedicineRecords.fxml");
355
356     }
357
358     catch (SQLException ae) {
359
360         System.out.println("Error!!! Connection Failed");
361         ae.printStackTrace();
362
363         Alert alert = new Alert(Alert.AlertType.ERROR);
364         alert.setTitle("Error Message");
365         alert.setHeaderText(null);
366         alert.setContentText("ERROR Deleting Selected Row!!");
367         alert.showAndWait();
368
369     }
370 }

```



```

369
370
371 @FXML
372 public void initialize() throws Exception {
373
374     //get column info
375
376     col_medicineName.setCellValueFactory(cellData -> cellData.getValue().get_MedicineName());
377     col_medicineQuantity.setCellValueFactory(cellData ->
378     cellData.getValue().get_MedicineQuantity().asObject());
379     col_medicineExpiryDate.setCellValueFactory(cellData ->
380     cellData.getValue().get_MedicineExpiryDate());
381
382     col_injectionName.setCellValueFactory(cellData ->
383     cellData.getValue().get_InjectionName());
384     col_injectionQuantity.setCellValueFactory(cellData ->
385     cellData.getValue().get_InjectionQuantity().asObject());
386     col_injectionExpiryDate.setCellValueFactory(cellData ->
387     cellData.getValue().get_InjectionExpiryDate());
388
389     ObservableList<Medicine> medicineList = getMedicineRecords();
390     populateMedicineTable(medicineList);
391
392     ObservableList<Injection> injectionList = getInjectionRecords();
393     populateInjectionTable(injectionList);
394
395 }
396
397 //set items in medicine table
398 public void populateMedicineTable (ObservableList<Medicine> medicineList) {
399     medicineTable.setItems(medicineList);
400 }
401
402 //get medicine records from database
403 public static ObservableList<Medicine> getMedicineRecords() {
404
405     ObservableList<Medicine> medicineList = FXCollections.observableArrayList();
406
407     try {
408
409         Connection connection =
410         DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
411         , "root", "root");
412         Statement statement = connection.createStatement();
413
414         String sql = "SELECT * FROM medicine_list;";
415         ResultSet resultset1 = statement.executeQuery(sql);
416
417         medicineList = getMedicineObject(resultset1);
418
419     }
420
421     catch (SQLException ae) {
422
423         System.out.println("Error!!! Connection Failed");
424         ae.printStackTrace();
425
426         Alert alert = new Alert(Alert.AlertType.ERROR);
427         alert.setTitle("Error Message");
428         alert.setHeaderText(null);
429         alert.setContentText("ERROR Connecting to MySQL Database!!");
430         alert.showAndWait();
431
432     }
433
434 }

```

```

426         return medicineList;
427     }
428 }
429
430 public static ObservableList<Medicine> getMedicineObject(ResultSet resultset1) {
431     ObservableList<Medicine> medicineRecordList = FXCollections.observableArrayList();
432
433     try {
434         while (resultset1.next()) {
435             Medicine obj = new Medicine();
436             obj.setMedicineName(resultset1.getString("medicine_name"));
437             obj.setMedicineQuantity(resultset1.getDouble("medicine_quantity"));
438             obj.setMedicineExpiryDate(resultset1.getString("medicine_expiry_date"));
439
440             medicineRecordList.add(obj);
441         }
442     }
443
444     catch (SQLException ae) {
445         System.out.println("Error!!! Connection Failed");
446         ae.printStackTrace();
447
448         Alert alert = new Alert(Alert.AlertType.ERROR);
449         alert.setTitle("Error Message");
450         alert.setHeaderText(null);
451         alert.setContentText("ERROR Connecting to MySQL Database!!");
452         alert.showAndWait();
453     }
454
455     return medicineRecordList;
456 }
457
458 //add items to injection table
459 public void populateInjectionTable (ObservableList<Injection> injectionList) {
460     injectionTable.setItems(injectionList);
461 }
462
463 //get injection records from database
464 public static ObservableList<Injection> getInjectionRecords() {
465     ObservableList<Injection> injectionList = FXCollections.observableArrayList();
466
467     try {
468         Connection connection =
469             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmamy_management_system"
470                 , "root", "root");
471         Statement statement = connection.createStatement();
472
473         String sql = "SELECT * FROM injection_list;";
474         ResultSet resultset2 = statement.executeQuery(sql);
475
476         injectionList = getInjectionObject(resultset2);
477     }
478 }

```

```

488         catch (SQLException ae) {
489
490             System.out.println("Error!!! Connection Failed");
491             ae.printStackTrace();
492
493             Alert alert = new Alert(Alert.AlertType.ERROR);
494             alert.setTitle("Error Message");
495             alert.setHeaderText(null);
496             alert.setContentText("ERROR Connecting to MySQL Database!!");
497             alert.showAndWait();
498
499         }
500
501         return injectionList;
502
503     }
504
505     public static ObservableList<Injection> getInjectionObject(ResultSet resultset2) {
506
507         ObservableList<Injection> injectionRecordList = FXCollections.observableArrayList();
508
509         try {
510
511             while (resultset2.next()) {
512
513                 Injection obj = new Injection();
514                 obj.setInjectionName(resultset2.getString("injection_name"));
515                 obj.setInjectionQuantity(resultset2.getDouble("injection_quantity"));
516                 obj.setInjectionExpiryDate(resultset2.getString("injection_expiry_date"));
517
518                 injectionRecordList.add(obj);
519
520             }
521
522         }
523
524         catch (SQLException ae) {
525
526             System.out.println("Error!!! Connection Failed");
527             ae.printStackTrace();
528
529             Alert alert = new Alert(Alert.AlertType.ERROR);
530             alert.setTitle("Error Message");
531             alert.setHeaderText(null);
532             alert.setContentText("ERROR Connecting to MySQL Database!!");
533             alert.showAndWait();
534
535         }
536
537         return injectionRecordList;
538
539     }
540
541     //MenuBar
542
543     //Menu 1 - File
544
545     //Menu Item 1.1
546     public void HomePageMenuItem(ActionEvent event) {
547         MenuBar obj = new MenuBar();
548         obj.HomePage();
549     }
550
551     //Menu Item 1.2
552     public void LogOutMenuItem(ActionEvent event) {

```

```
552     MenuBar obj = new MenuBar();
553     obj.LogOut();
554 }
555
556 //Menu Item 1.3
557 public void ExitMenuItem(ActionEvent event) {
558     MenuBar obj = new MenuBar();
559     obj.Exit();
560 }
561
562 //Menu 2 - View
563
564 //Menu Item 2.1
565 public void NewPatientEntryMenuItem(ActionEvent event) {
566     MenuBar obj = new MenuBar();
567     obj.NewPatientEntry();
568 }
569
570 //Menu Item 2.2
571 public void ShowPatientRecordMenuItem(ActionEvent event) {
572     MenuBar obj = new MenuBar();
573     obj.ShowPatientRecord();
574 }
575
576 //Menu Item 2.3
577 public void MedicineRecordsMenuItem(ActionEvent event) {
578     MenuBar obj = new MenuBar();
579     obj.MedicineRecords();
580 }
581
582 //Menu 3 - Help
583
584 //Menu Item 3.1
585 public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
586     MenuBar obj = new MenuBar();
587     obj.CollegeWebsiteLinks();
588 }
589
590 //Menu Item 3.2
591 public void AboutMenuItem(ActionEvent event) {
592     MenuBar obj = new MenuBar();
593     obj.About();
594 }
595
596 }
```

```
1  package application;
2
3  import javafx.beans.property.IntegerProperty;
4  import javafx.beans.property.SimpleIntegerProperty;
5  import javafx.beans.property.SimpleStringProperty;
6  import javafx.beans.property.StringProperty;
7
8
9  public class Patient {
10
11     private IntegerProperty regNo, stdNo, year, age, phoneNo, leaveDays;
12     private StringProperty name, course, dept, sex, date, time, leaveOptions, complaint,
        remarks;
13
14     //constructor
15     public Patient() {
16
17         this.regNo = new SimpleIntegerProperty();
18         this.name = new SimpleStringProperty();
19         this.stdNo = new SimpleIntegerProperty();
20         this.year = new SimpleIntegerProperty();
21         this.course = new SimpleStringProperty();
22         this.dept = new SimpleStringProperty();
23         this.sex = new SimpleStringProperty();
24         this.age = new SimpleIntegerProperty();
25         this.phoneNo = new SimpleIntegerProperty();
26         this.date = new SimpleStringProperty();
27         this.time = new SimpleStringProperty();
28         this.leaveOptions = new SimpleStringProperty();
29         this.leaveDays = new SimpleIntegerProperty();
30         this.complaint = new SimpleStringProperty();
31         this.remarks = new SimpleStringProperty();
32
33     }
34
35     //getters and setters
36     public Integer getStdNo() {
37         return stdNo.get();
38     }
39
40     public void setStdNo(Integer stdNo) {
41         this.stdNo.set(stdNo);
42     }
43
44     public IntegerProperty get_StdNo() {
45         return stdNo;
46     }
47
48     public Integer getYear() {
49         return year.get();
50     }
51
52     public void setYear(Integer year) {
53         this.year.set(year);
54     }
55
56     public IntegerProperty get_Year() {
57         return year;
58     }
59
60     public Integer getAge() {
61         return age.get();
62     }
63 }
```

```
64     public void setAge(Integer age) {
65         this.age.set(age);
66     }
67
68     public IntegerProperty get_Age() {
69         return age;
70     }
71
72     public Integer getPhoneNo() {
73         return phoneNo.get();
74     }
75
76     public void setPhoneNo(Integer phoneNo) {
77         this.phoneNo.set(phoneNo);
78     }
79
80     public IntegerProperty get_PhoneNo() {
81         return phoneNo;
82     }
83
84     public Integer getLeaveDays() {
85         return leaveDays.get();
86     }
87
88     public void setLeaveDays(Integer leaveDays) {
89         this.leaveDays.set(leaveDays);
90     }
91
92     public IntegerProperty get_LeaveDays() {
93         return leaveDays;
94     }
95
96     public Integer getRegNo() {
97         return regNo.get();
98     }
99
100    public void setRegNo(Integer regNo) {
101        this.regNo.set(regNo);
102    }
103
104    public IntegerProperty get_RegNo() {
105        return regNo;
106    }
107
108    public String getName() {
109        return name.get();
110    }
111
112    public void setName(String name) {
113        this.name.set(name);
114    }
115
116    public StringProperty get_Name() {
117        return name;
118    }
119
120    public String getCourse() {
121        return course.get();
122    }
123
124    public void setCourse(String course) {
125        this.course .set(course);
126    }
127
```

```
128     public StringProperty get_Course() {
129         return course;
130     }
131
132     public String getDept() {
133         return dept.get();
134     }
135
136     public void setDept(String dept) {
137         this.dept.set(dept);
138     }
139
140     public StringProperty get_Dept() {
141         return dept;
142     }
143
144     public String getSex() {
145         return sex.get();
146     }
147
148     public void setSex(String sex) {
149         this.sex.set(sex);
150     }
151
152     public StringProperty get_Sex() {
153         return sex;
154     }
155
156     public String getDate() {
157         return date.get();
158     }
159
160     public void setDate(String date) {
161         this.date.set(date);
162     }
163
164     public StringProperty get_Date() {
165         return date;
166     }
167
168     public String getTime() {
169         return time.get();
170     }
171
172     public void setTime(String time) {
173         this.time.set(time);
174     }
175
176     public StringProperty get_Time() {
177         return time;
178     }
179
180     public String getLeaveOptions() {
181         return leaveOptions.get();
182     }
183
184     public void setLeaveOptions(String leaveOptions) {
185         this.leaveOptions.set(leaveOptions);
186     }
187
188     public StringProperty get_leaveOptions() {
189         return leaveOptions;
190     }
191
```

```

192         public String getComplaint() {
193             return complaint.get();
194         }
195
196         public void setComplaint(String complaint) {
197             this.complaint.set(complaint);
198         }
199
200         public StringProperty get_Complaint() {
201             return complaint;
202         }
203
204         public String getRemarks() {
205             return remarks.get();
206         }
207
208         public void setRemarks(String remarks) {
209             this.remarks.set(remarks);
210         }
211
212         public StringProperty get_Remarks() {
213             return remarks;
214         }
215
216     }
217
218
219     //private Integer regNo, stdNo, year, age, phoneNo, leaveDays;
220     //private String name, course, dept, sex, date, time, leaveOptions, complaint, remarks;
221     //
222     //
223     //public Patient(int regNo, String name, Integer stdNo, Integer year, String course, String
224     dept, String sex, Integer age, Integer phoneNo, String date, String time, String leaveOptions,
225     Integer leaveDays, String complaint, String remarks) {
226         //
227         // this.regNo = regNo;
228         // this.stdNo = stdNo;
229         // this.year = year;
230         // this.age = age;
231         // this.phoneNo = phoneNo;
232         // this.leaveDays = leaveDays;
233         // this.name = name;
234         // this.course = course;
235         // this.dept = dept;
236         // this.sex = sex;
237         // this.date = date;
238         // this.time = time;
239         // this.leaveOptions = leaveOptions;
240         // this.complaint = complaint;
241         // this.remarks = remarks;
242         //}
243         //
244         //
245         //public Integer getRegNo() {
246         // return regNo;
247         //}
248         //
249         //
250         //public void setRegNo(Integer regNo) {
251         // this.regNo = regNo;
252         //}
253         //
254         //
255         //public Integer getStdNo() {

```



```
254 // return stdNo;
255 //}
256 //
257 //
258 //public void setStdNo(Integer stdNo) {
259 //    this.stdNo = stdNo;
260 //}
261 //
262 //
263 //public Integer getYear() {
264 //    return year;
265 //}
266 //
267 //
268 //public void setYear(Integer year) {
269 //    this.year = year;
270 //}
271 //
272 //
273 //public Integer getAge() {
274 //    return age;
275 //}
276 //
277 //
278 //public void setAge(Integer age) {
279 //    this.age = age;
280 //}
281 //
282 //
283 //public Integer getPhoneNo() {
284 //    return phoneNo;
285 //}
286 //
287 //
288 //public void setPhoneNo(Integer phoneNo) {
289 //    this.phoneNo = phoneNo;
290 //}
291 //
292 //
293 //public Integer getLeaveDays() {
294 //    return leaveDays;
295 //}
296 //
297 //
298 //public void setLeaveDays(Integer leaveDays) {
299 //    this.leaveDays = leaveDays;
300 //}
301 //
302 //
303 //public String getName() {
304 //    return name;
305 //}
306 //
307 //
308 //public void setName(String name) {
309 //    this.name = name;
310 //}
311 //
312 //
313 //public String getCourse() {
314 //    return course;
315 //}
316 //
317 //
```

```
318 //public void setCourse(String course) {
319 //    this.course = course;
320 //}
321 //
322 //
323 //public String getDept() {
324 //    return dept;
325 //}
326 //
327 //
328 //public void setDept(String dept) {
329 //    this.dept = dept;
330 //}
331 //
332 //
333 //public String getSex() {
334 //    return sex;
335 //}
336 //
337 //
338 //public void setSex(String sex) {
339 //    this.sex = sex;
340 //}
341 //
342 //
343 //public String getDate() {
344 //    return date;
345 //}
346 //
347 //
348 //public void setDate(String date) {
349 //    this.date = date;
350 //}
351 //
352 //
353 //public String getTime() {
354 //    return time;
355 //}
356 //
357 //
358 //public void setTime(String time) {
359 //    this.time = time;
360 //}
361 //
362 //
363 //public String getLeaveOptions() {
364 //    return leaveOptions;
365 //}
366 //
367 //
368 //public void setLeaveOptions(String leaveOptions) {
369 //    this.leaveOptions = leaveOptions;
370 //}
371 //
372 //
373 //public String getComplaint() {
374 //    return complaint;
375 //}
376 //
377 //
378 //public void setComplaint(String complaint) {
379 //    this.complaint = complaint;
380 //}
381 //
```

```
382  //
383  //public String getRemarks() {
384  //    return remarks;
385  //}
386  //
387  //
388  //public void setRemarks(String remarks) {
389  //    this.remarks = remarks;
390  //}
391
```

```
1  package application;
2
3  import javafx.beans.property.IntegerProperty;
4  import javafx.beans.property.SimpleIntegerProperty;
5  import javafx.beans.property.SimpleStringProperty;
6  import javafx.beans.property.StringProperty;
7
8
9  public class StaffPatient {
10
11     private IntegerProperty regNo, age, phoneNo, leaveDays;
12     private StringProperty name, dept, sex, date, time, leaveOptions, complaint, remarks;
13
14     //default constructor
15     public StaffPatient() {
16
17         this.regNo = new SimpleIntegerProperty();
18         this.name = new SimpleStringProperty();
19         this.dept = new SimpleStringProperty();
20         this.sex = new SimpleStringProperty();
21         this.age = new SimpleIntegerProperty();
22         this.phoneNo = new SimpleIntegerProperty();
23         this.date = new SimpleStringProperty();
24         this.time = new SimpleStringProperty();
25         this.leaveOptions = new SimpleStringProperty();
26         this.leaveDays = new SimpleIntegerProperty();
27         this.complaint = new SimpleStringProperty();
28         this.remarks = new SimpleStringProperty();
29
30     }
31
32     //getters and setters
33     public Integer getAge() {
34         return age.get();
35     }
36
37     public void setAge(Integer age) {
38         this.age.set(age);
39     }
40
41     public IntegerProperty get_Age() {
42         return age;
43     }
44
45     public Integer getPhoneNo() {
46         return phoneNo.get();
47     }
48
49     public void setPhoneNo(Integer phoneNo) {
50         this.phoneNo.set(phoneNo);
51     }
52
53     public IntegerProperty get_PhoneNo() {
54         return phoneNo;
55     }
56
57     public Integer getLeaveDays() {
58         return leaveDays.get();
59     }
60
61     public void setLeaveDays(Integer leaveDays) {
62         this.leaveDays.set(leaveDays);
63     }
64 }
```

```
65     public IntegerProperty get_LeaveDays() {
66         return leaveDays;
67     }
68
69     public Integer getRegNo() {
70         return regNo.get();
71     }
72
73     public void setRegNo(Integer regNo) {
74         this.regNo.set(regNo);
75     }
76
77     public IntegerProperty get_RegNo() {
78         return regNo;
79     }
80
81     public String getName() {
82         return name.get();
83     }
84
85     public void setName(String name) {
86         this.name.set(name);
87     }
88
89     public StringProperty get_Name() {
90         return name;
91     }
92
93     public String getDept() {
94         return dept.get();
95     }
96
97     public void setDept(String dept) {
98         this.dept.set(dept);
99     }
100
101     public StringProperty get_Dept() {
102         return dept;
103     }
104
105     public String getSex() {
106         return sex.get();
107     }
108
109     public void setSex(String sex) {
110         this.sex.set(sex);
111     }
112
113     public StringProperty get_Sex() {
114         return sex;
115     }
116
117     public String getDate() {
118         return date.get();
119     }
120
121     public void setDate(String date) {
122         this.date.set(date);
123     }
124
125     public StringProperty get_Date() {
126         return date;
127     }
128
```

```
129     public String getTime() {
130         return time.get();
131     }
132
133     public void setTime(String time) {
134         this.time.set(time);
135     }
136
137     public StringProperty get_Time() {
138         return time;
139     }
140
141     public String getLeaveOptions() {
142         return leaveOptions.get();
143     }
144
145     public void setLeaveOptions(String leaveOptions) {
146         this.leaveOptions.set(leaveOptions);
147     }
148
149     public StringProperty get_leaveOptions() {
150         return leaveOptions;
151     }
152
153     public String getComplaint() {
154         return complaint.get();
155     }
156
157     public void setComplaint(String complaint) {
158         this.complaint.set(complaint);
159     }
160
161     public StringProperty get_Complaint() {
162         return complaint;
163     }
164
165     public String getRemarks() {
166         return remarks.get();
167     }
168
169     public void setRemarks(String remarks) {
170         this.remarks.set(remarks);
171     }
172
173     public StringProperty get_Remarks() {
174         return remarks;
175     }
176
177 }
```

```
1  package application;
2
3  import javafx.beans.property.IntegerProperty;
4  import javafx.beans.property.SimpleIntegerProperty;
5  import javafx.beans.property.SimpleStringProperty;
6  import javafx.beans.property.StringProperty;
7
8
9  public class OtherPatient {
10
11     private IntegerProperty regNo, age, phoneNo;
12     private StringProperty name, sex, date, time, complaint, remarks;
13
14     //constructor
15     public OtherPatient() {
16
17         this.regNo = new SimpleIntegerProperty();
18         this.name = new SimpleStringProperty();
19         this.sex = new SimpleStringProperty();
20         this.age = new SimpleIntegerProperty();
21         this.phoneNo = new SimpleIntegerProperty();
22         this.date = new SimpleStringProperty();
23         this.time = new SimpleStringProperty();
24         this.complaint = new SimpleStringProperty();
25         this.remarks = new SimpleStringProperty();
26
27     }
28
29     //Getters and Setters for Table View
30     public Integer getAge() {
31         return age.get();
32     }
33
34     public void setAge(Integer age) {
35         this.age.set(age);
36     }
37
38     public IntegerProperty get_Age() {
39         return age;
40     }
41
42     public Integer getPhoneNo() {
43         return phoneNo.get();
44     }
45
46     public void setPhoneNo(Integer phoneNo) {
47         this.phoneNo.set(phoneNo);
48     }
49
50     public IntegerProperty get_PhoneNo() {
51         return phoneNo;
52     }
53
54     public Integer getRegNo() {
55         return regNo.get();
56     }
57
58     public void setRegNo(Integer regNo) {
59         this.regNo.set(regNo);
60     }
61
62     public IntegerProperty get_RegNo() {
63         return regNo;
64     }
```

```
65
66     public String getName() {
67         return name.get();
68     }
69
70     public void setName(String name) {
71         this.name.set(name);
72     }
73
74     public StringProperty get_Name() {
75         return name;
76     }
77
78     public String getSex() {
79         return sex.get();
80     }
81
82     public void setSex(String sex) {
83         this.sex.set(sex);
84     }
85
86     public StringProperty get_Sex() {
87         return sex;
88     }
89
90     public String getDate() {
91         return date.get();
92     }
93
94     public void setDate(String date) {
95         this.date.set(date);
96     }
97
98     public StringProperty get_Date() {
99         return date;
100    }
101
102    public String getTime() {
103        return time.get();
104    }
105
106    public void setTime(String time) {
107        this.time.set(time);
108    }
109
110    public StringProperty get_Time() {
111        return time;
112    }
113
114    public String getComplaint() {
115        return complaint.get();
116    }
117
118    public void setComplaint(String complaint) {
119        this.complaint.set(complaint);
120    }
121
122    public StringProperty get_Complaint() {
123        return complaint;
124    }
125
126    public String getRemarks() {
127        return remarks.get();
128    }
```



```
129
130     public void setRemarks(String remarks) {
131         this.remarks.set(remarks);
132     }
133
134     public StringProperty get_Remarks() {
135         return remarks;
136     }
137
138 }
139
```

```
1  package application;
2
3  import javafx.beans.property.DoubleProperty;
4  import javafx.beans.property.SimpleDoubleProperty;
5  import javafx.beans.property.SimpleStringProperty;
6  import javafx.beans.property.StringProperty;
7
8  public class Medicine {
9
10     private DoubleProperty medicineQuantity;
11     private StringProperty medicineName, medicineExpiryDate;
12
13     //Constructor
14     public Medicine() {
15
16         this.medicineName = new SimpleStringProperty();
17         this.medicineQuantity = new SimpleDoubleProperty();
18         this.medicineExpiryDate = new SimpleStringProperty();
19     }
20
21     //Getters and Setters for Table View
22     public String getMedicineName() {
23         return medicineName.get();
24     }
25
26     public void setMedicineName(String medicineName) {
27         this.medicineName.set(medicineName);
28     }
29
30     public StringProperty get_MedicineName() {
31         return medicineName;
32     }
33
34     public Double getMedicineQuantity() {
35         return medicineQuantity.get();
36     }
37
38     public void setMedicineQuantity(Double medicineQuantity) {
39         this.medicineQuantity.set(medicineQuantity);
40     }
41
42     public DoubleProperty get_MedicineQuantity() {
43         return medicineQuantity;
44     }
45
46     public String getMedicineExpiryDate() {
47         return medicineExpiryDate.get();
48     }
49
50     public void setMedicineExpiryDate(String medicineExpiryDate) {
51         this.medicineExpiryDate.set(medicineExpiryDate);
52     }
53
54     public StringProperty get_MedicineExpiryDate() {
55         return medicineExpiryDate;
56     }
57
58 }
59
60
```

```
1  package application;
2
3  import javafx.beans.property.DoubleProperty;
4  import javafx.beans.property.SimpleDoubleProperty;
5  import javafx.beans.property.SimpleStringProperty;
6  import javafx.beans.property.StringProperty;
7
8  public class Injection {
9
10     private DoubleProperty injectionQuantity;
11     private StringProperty injectionName, injectionExpiryDate;
12
13     //Default Constructor
14     public Injection() {
15
16         this.injectionName = new SimpleStringProperty();
17         this.injectionQuantity = new SimpleDoubleProperty();
18         this.injectionExpiryDate = new SimpleStringProperty();
19     }
20
21     //Getters and Setters
22     public String getInjectionName() {
23         return injectionName.get();
24     }
25
26     public void setInjectionName(String injectionName) {
27         this.injectionName.set(injectionName);
28     }
29
30     public StringProperty get_InjectionName() {
31         return injectionName;
32     }
33
34     public Double getInjectionQuantity() {
35         return injectionQuantity.get();
36     }
37
38     public void setInjectionQuantity(Double injectionQuantity) {
39         this.injectionQuantity.set(injectionQuantity);
40     }
41
42     public DoubleProperty get_InjectionQuantity() {
43         return injectionQuantity;
44     }
45
46     public String getInjectionExpiryDate() {
47         return injectionExpiryDate.get();
48     }
49
50     public void setInjectionExpiryDate(String injectionExpiryDate) {
51         this.injectionExpiryDate.set(injectionExpiryDate);
52     }
53
54     public StringProperty get_InjectionExpiryDate() {
55         return injectionExpiryDate;
56     }
57
58 }
59
60
```

```
1  package application;
2
3  import javafx.scene.control.Alert;
4  import javafx.scene.control.ButtonType;
5
6  public class MenuBar {
7
8      Main m = new Main();
9
10     //Menu 1 - File
11
12     //Menu Item 1.1
13     public void HomePage() {
14
15         m.changeScene("HomePage.fxml");
16
17     }
18
19     //Menu Item 1.2
20     public void LogOut() {
21
22         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
23         alert.setTitle("Logout");
24         alert.setHeaderText(null);
25         alert.setContentText("Are you sure you want to Logout?");
26         if (alert.showAndWait().get() == ButtonType.OK){
27             m.changeScene("LoginPage.fxml");
28         }
29
30     }
31
32     //Menu Item 1.3
33     public void Exit() {
34
35         System.exit(0);
36
37     }
38
39     //Menu 2 - View
40
41     //Menu Item 2.1
42     public void NewPatientEntry() {
43
44         m.changeScene("SelectPatientEntryPage.fxml");
45
46     }
47
48     //Menu Item 2.2
49     public void ShowPatientRecord() {
50
51         m.showPatientRecords("SelectPatientRecordsPage.fxml");
52
53     }
54
55     //Menu Item 2.3
56     public void MedicineRecords() {
57
58         m.changeScene("MedicineRecords.fxml");
59
60     }
61
62     //Menu 3 - Help
63
64     //Menu Item 3.1
```

```
65     public void CollegeWebsiteLinks() {
66
67         m.changeScene("CollegeWebsiteLinks.fxml");
68
69     }
70
71     //Menu Item 3.2
72     public void About() {
73
74         m.changeScene("About.fxml");
75     }
76
77
78
79 }
80
```

```

1  package application;
2
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXML;
5  import javafx.scene.control.Alert;
6  import javafx.scene.control.Button;
7  import javafx.scene.control.ButtonType;
8
9  import java.io.IOException;
10
11 public class About {
12
13     @FXML
14     private Button HomePage;
15
16     @FXML
17     private Button Logout;
18
19     //method called when logout button is pressed
20     public void userLogout(ActionEvent event) throws IOException {
21
22         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
23         alert.setTitle("Logout");
24         alert.setHeaderText(null);
25         alert.setContentText("Are you sure you want to Logout?");
26         if (alert.showAndWait().get() == ButtonType.OK){
27             Main m = new Main();
28             m.changeScene("LoginPage.fxml");
29         }
30
31     }
32
33     public void HomePage(ActionEvent event) throws IOException {
34
35         Main m = new Main();
36         m.changeScene("HomePage.fxml");
37
38     }
39
40     //MenuBar
41
42     //Menu 1 - File
43
44     //Menu Item 1.1
45     public void HomePageMenuItem(ActionEvent event) {
46         MenuBar obj = new MenuBar();
47         obj.HomePage();
48     }
49
50     //Menu Item 1.2
51     public void LogOutMenuItem(ActionEvent event) {
52         MenuBar obj = new MenuBar();
53         obj.LogOut();
54     }
55
56     //Menu Item 1.3
57     public void ExitMenuItem(ActionEvent event) {
58         MenuBar obj = new MenuBar();
59         obj.Exit();
60     }
61
62     //Menu 2 - View
63
64     //Menu Item 2.1

```

```
65     public void NewPatientEntryMenuItem(ActionEvent event) {
66         MenuBar obj = new MenuBar();
67         obj.NewPatientEntry();
68     }
69
70     //Menu Item 2.2
71     public void ShowPatientRecordMenuItem(ActionEvent event) {
72         MenuBar obj = new MenuBar();
73         obj.ShowPatientRecord();
74     }
75
76     //Menu Item 2.3
77     public void MedicineRecordsMenuItem(ActionEvent event) {
78         MenuBar obj = new MenuBar();
79         obj.MedicineRecords();
80     }
81
82     //Menu 3 - Help
83
84     //Menu Item 3.1
85     public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
86         MenuBar obj = new MenuBar();
87         obj.CollegeWebsiteLinks();
88     }
89
90     //Menu Item 3.2
91     public void AboutMenuItem(ActionEvent event) {
92         MenuBar obj = new MenuBar();
93         obj.About();
94     }
95
96 }
97
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.net.URL;
5  import java.sql.*;
6  import java.time.LocalDate;
7  import java.time.LocalDateTime;
8  import java.time.format.DateTimeFormatter;
9  import java.util.ResourceBundle;
10
11  import javafx.collections.FXCollections;
12  import javafx.collections.ObservableList;
13  import javafx.event.ActionEvent;
14  import javafx.fxml.*;
15  import javafx.fxml.FXML;
16  import javafx.scene.control.*;
17
18  public class PatientDetailsEntryController implements Initializable {
19
20      // Buttons
21
22      @FXML
23      private Button backButton;
24
25      @FXML
26      private Button saveButton;
27
28      // Patient Details Entry Components
29
30      @FXML
31      public TextField nameTextField;
32
33      @FXML
34      private TextField studentNoTextField;
35
36      @FXML
37      private TextField regNoTextField;
38
39      @FXML
40      private RadioButton maleRadioButton;
41
42      @FXML
43      private RadioButton femaleRadioButton;
44
45      @FXML
46      private ToggleGroup gender;
47
48      @FXML
49      private TextField ageTextField;
50
51      @FXML
52      private TextField phoneNoTextField;
53
54      @FXML
55      private DatePicker dateSelector;
56
57      @FXML
58      private TextField timeTextField;
59
60      @FXML
61      private RadioButton amRadioButton;
62
63      @FXML
64      private RadioButton pmRadioButton;
```



```
65
66     @FXML
67     private ToggleGroup timeSelect;
68
69     @FXML
70     private TextArea complaintTextArea;
71
72     @FXML
73     private TextArea remarksTextArea;
74
75     @FXML
76     private TextField leaveDaysTextField;
77
78     // ComboBox
79
80     @FXML
81     public ComboBox<String> yearComboBox;
82
83     @FXML
84     public ComboBox<String> courseComboBox;
85
86     @FXML
87     public ComboBox<String> departmentComboBox;
88
89     @FXML
90     public ComboBox<String> leaveOptionsComboBox;
91
92     //Medicine
93     @FXML
94     public CheckBox medicineChecBox;
95
96     @FXML
97     public ComboBox<String> medicineComboBox;
98
99     @FXML
100    public TextField medicineQuantityTextField;
101
102    @FXML
103    public CheckBox injectionChecBox;
104
105    @FXML
106    public ComboBox<String> injectionComboBox;
107
108    @FXML
109    public TextField injectionQuantityTextField;
110
111    // Error Labels
112    @FXML
113    private Label nameErrorMessage;
114
115    @FXML
116    private Label stdNoErrorMessage;
117
118    @FXML
119    private Label ageErrorMessage;
120
121    @FXML
122    private Label genderErrorMessage;
123
124    @FXML
125    private Label deptErrorMessage;
126
127    @FXML
128    private Label yearErrorMessage;
```

```

129
130 @FXML
131 private Label courseErrorMessage;
132
133 @FXML
134 private Label phoneNoErrorMessage;
135
136 @FXML
137 private Label complaintErrorMessage;
138
139 @FXML
140 private Label remarksErrorMessage;
141
142 @FXML
143 private Label leaveDaysErrorMessage;
144
145 @FXML
146 private Label leaveOptionsErrorMessage;
147
148 public String sex, time, year, course, department, leaveOptions, ampm;
149 public LocalDate date;
150 public Integer lastRegNo;
151
152 //combo box items in a array list
153 ObservableList<String> yearList = FXCollections.observableArrayList("1", "2", "3", "4",
154 "5");
155 ObservableList<String> departmentList = FXCollections.observableArrayList("Civil
156 Department", "Architecture Department", "Electrical Department", "IT Department", "ECE
157 Department");
158 ObservableList<String> leaveOptionList = FXCollections.observableArrayList("No Leave",
159 "Bed Rest", "Full Day Leave", "Half Day Leave", "Referred to PGH");
160 ObservableList<String> medicineList = FXCollections.observableArrayList();
161 ObservableList<String> injectionList = FXCollections.observableArrayList();
162
163 @Override
164 public void initialize(URL arg0, ResourceBundle arg1) {
165     departmentComboBox.setItems(departmentList);
166     leaveOptionsComboBox.setItems(leaveOptionList);
167     yearComboBox.setItems(yearList);
168     medicineComboBox.setItems(medicineList);
169     injectionComboBox.setItems(injectionList);
170
171     //sets current data in date selector
172     DateSelector.setValue(LocalDate.now());
173     date = DateSelector.getValue();
174
175     //sets current time in text field
176     String pattern = "hh:mm";
177     timeTextField.setText((LocalTime.now()).format(DateTimeFormatter.ofPattern(pattern)));
178
179     //selects radio button depending on the time
180     String ampmPattern = "hh: mm a";
181     ampm = (LocalTime.now()).format(DateTimeFormatter.ofPattern(ampmPattern));
182     ampm = ampm.substring(6, 7);
183
184     if (ampm.equals("AM")) {
185         amRadioButton.setSelected(true);
186     } else {
187         pmRadioButton.setSelected(true);
188     }
189
190     time = timeTextField.getText() + ampm;
191
192     //calls a method go get last RegNo stored in the database

```

```

189         getLastRegNo();
190
191         //gets medicine names from database and saves it in the combo box
192         try {
193
194             Connection connection1 =
195                 DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmmary_management_system"
196                     , "root", "root");
197             Statement statement1 = connection1.createStatement();
198
199             String sql = "SELECT * FROM medicine_list;";
200             ResultSet resultset1 = statement1.executeQuery(sql);
201
202             while (resultset1.next()) {
203                 String medicineName = resultset1.getString("medicine_name");
204                 medicineComboBox.getItems().addAll(medicineName);
205             }
206
207             statement1.close();
208             connection1.close();
209             resultset1.close();
210
211         }
212
213         catch (SQLException ae) {
214
215             System.out.println("Error!!! Connection Failed");
216             ae.printStackTrace();
217
218             Alert alert = new Alert(Alert.AlertType.ERROR);
219             alert.setTitle("Error Message");
220             alert.setHeaderText(null);
221             alert.setContentText("ERROR fetching Medicine Name from Database!!");
222             alert.showAndWait();
223
224         }
225
226         //gets injection names from database and saves it in the combo box
227         try {
228
229             Connection connection2 =
230                 DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmmary_management_system"
231                     , "root", "root");
232             Statement statement2 = connection2.createStatement();
233
234             String sql = "SELECT * FROM injection_list;";
235             ResultSet resultset2 = statement2.executeQuery(sql);
236
237             while (resultset2.next()) {
238                 String injectionName = resultset2.getString("injection_name");
239                 injectionComboBox.getItems().addAll(injectionName);
240             }
241
242             statement2.close();
243             connection2.close();
244             resultset2.close();
245
246         }
247
248         catch (SQLException ae) {
249
250             System.out.println("Error!!! Connection Failed");
251             ae.printStackTrace();
252
253         }
254     }
255 }

```

```

249         Alert alert = new Alert(Alert.AlertType.ERROR);
250         alert.setTitle("Error Message");
251         alert.setHeaderText(null);
252         alert.setContentText("ERROR fetching Injection Name from Database!!");
253         alert.showAndWait();
254     }
255 }
256
257 //gets selected department and displays courses according to it
258 public void deptComboBoxSelected (ActionEvent event) throws IOException {
259     department = departmentComboBox.getValue();
260
261     if (department == "Civil Department") {
262         ObservableList<String> courseList = FXCollections.observableArrayList("ME in CM",
263             "BE CE", "BE EG");
264         courseComboBox.setItems(courseList);
265         // courseComboBox.getItems().addAll("ME in CM", "BE CE", "BE EG");
266     }
267
268     else if (department == "Architecture Department") {
269         ObservableList<String> courseList = FXCollections.observableArrayList("B Arch.");
270         courseComboBox.setItems(courseList);
271         // courseComboBox.getItems().addAll("B Arch.");
272     }
273
274     else if (department == "Electrical Department") {
275         ObservableList<String> courseList = FXCollections.observableArrayList("ME in RE",
276             "BE EE");
277         courseComboBox.setItems(courseList);
278         // courseComboBox.getItems().addAll("ME in RE", "BE EE");
279     }
280
281     else if (department == "IT Department") {
282         ObservableList<String> courseList = FXCollections.observableArrayList("BE IT");
283         courseComboBox.setItems(courseList);
284         // courseComboBox.getItems().addAll("BE IT");
285     }
286
287     else if (department == "ECE Department") {
288         ObservableList<String> courseList = FXCollections.observableArrayList("BE ECE",
289             "BE ICE");
290         courseComboBox.setItems(courseList);
291         // courseComboBox.getItems().addAll("BE ECE", "BE ICE");
292     }
293
294 }
295
296 //gets selected combo box item
297 public void comboChanged (ActionEvent event) throws IOException{
298     year = yearComboBox.getValue();
299     course = courseComboBox.getValue();
300     leaveOptions = leaveOptionsComboBox.getValue();
301
302     if (leaveOptions == "No Leave") {
303         leaveDaysTextField.setText("0");
304     }
305 }
306
307 }
308
309

```

```

310
311 public void homePage(ActionEvent event) throws IOException {
312
313     Main m = new Main();
314     m.changeScene("SelectPatientEntryPage.fxml");
315
316 }
317
318 public void sexRadioSelect (ActionEvent event) throws IOException {
319
320     if (maleRadioButton.isSelected()) {
321         sex = "Male";
322     }
323
324     else if (femaleRadioButton.isSelected()) {
325         sex = "Female";
326     }
327
328 }
329
330 public void timeRadioSelect (ActionEvent event) throws IOException {
331
332     if (amRadioButton.isSelected()) {
333         time = timeTextField.getText() + " AM";
334     }
335
336     else if (pmRadioButton.isSelected()) {
337         time = timeTextField.getText() + " PM";
338     }
339
340 }
341
342 //updates medicine quantity in database after prescribing it
343 public void medicinePrescribed(ActionEvent event) {
344
345     String medicineName = medicineComboBox.getValue();
346
347     try {
348
349         Connection connection =
350             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
351                 , "root", "root");
352
353         String sql = "UPDATE medicine_list SET medicine_quantity = (medicine_quantity-?)
354             WHERE medicine_name = ?";
355
356         PreparedStatement statement = connection.prepareStatement(sql);
357
358         statement.setDouble(1, Double.parseDouble(medicineQuantityTextField.getText()));
359         statement.setString(2, medicineName);
360
361         statement.executeUpdate();
362
363         statement.close();
364         connection.close();
365
366     }
367
368     catch (SQLException ae) {
369
370         System.out.println("Error!!! Connection Failed");
371         ae.printStackTrace();
372
373         Alert alert = new Alert(Alert.AlertType.ERROR);

```

```

371         alert.setTitle("Error Message");
372         alert.setHeaderText(null);
373         alert.setContentText("ERROR Prescribing Medicine!!");
374         alert.showAndWait();
375
376     }
377
378 }
379
380 //updates injection quantity in database after prescribing it
381 public void prescribeInjection(ActionEvent event) {
382
383     String injectionName = injectionComboBox.getValue();
384
385     try {
386
387         Connection connection =
388             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
389                 , "root", "root");
390
391         String sql = "UPDATE injection_list SET injection_quantity = (injection_quantity-?)
392             WHERE injection_name = ?;";
393
394         PreparedStatement statement = connection.prepareStatement(sql);
395
396         statement.setDouble(1, Double.parseDouble(injectionQuantityTextField.getText()));
397         statement.setString(2, injectionName);
398
399         statement.executeUpdate();
400
401         statement.close();
402         connection.close();
403
404     }
405
406     catch (SQLException ae) {
407
408         System.out.println("Error!!! Connection Failed");
409         ae.printStackTrace();
410
411         Alert alert = new Alert(Alert.AlertType.ERROR);
412         alert.setTitle("Error Message");
413         alert.setHeaderText(null);
414         alert.setContentText("ERROR Prescribing Injection!!");
415         alert.showAndWait();
416
417     }
418
419 }
420
421 public void saveDetails(ActionEvent event) throws IOException {
422
423     //checks if feilds are empty
424     try {
425         if (year.isEmpty() || course.isEmpty() || leaveOptions.isEmpty() || sex.isEmpty() ||
426             course.isEmpty() || department.isEmpty()) {
427             throw new NullPointerException();
428         }
429     }
430
431     catch (NullPointerException ae) {
432
433         System.out.println("Error!!! Connection Failed");
434
435     }
436
437 }

```

```

431         ae.printStackTrace();
432
433         Alert alert = new Alert(Alert.AlertType.ERROR);
434         alert.setTitle("Error Message");
435         alert.setHeaderText(null);
436         alert.setContentText("Please Fill-up all fields!!");
437         alert.showAndWait();
438
439     }
440
441     boolean allCorrect = checkAllValues();
442
443     //saves data in database if all fields are filled
444     if (allCorrect) {
445
446         try {
447
448             date = DateSelector.getValue();
449
450             Connection connection =
451             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarary_management_sys
452             tem", "root", "root");
453
454             String sql = "INSERT INTO patient_records (reg_no, name, student_no, sex, age,
455             year, course, department, phone_no, date, time, leave_options, no_of_days,
456             complaint, remarks) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
457
458             PreparedStatement statement = connection.prepareStatement(sql);
459
460             statement.setInt(1, Integer.parseInt(regNoTextField.getText()));
461             statement.setString(2, nameTextField.getText());
462             statement.setInt(3, Integer.parseInt(studentNoTextField.getText()));
463             statement.setString(4, sex);
464             statement.setInt(5, Integer.parseInt(ageTextField.getText()));
465             statement.setInt(6, Integer.parseInt(year));
466             statement.setString(7, course);
467             statement.setString(8, department);
468             statement.setInt(9, Integer.parseInt(phoneNoTextField.getText()));
469             statement.setString(10, date.toString());
470             statement.setString(11, time);
471             statement.setString(12, leaveOptions);
472             statement.setInt(13, Integer.parseInt(leaveDaysTextField.getText()));
473             statement.setString(14, complaintTextArea.getText());
474             statement.setString(15, remarksTextArea.getText());
475
476             int rowsInserted = statement.executeUpdate();
477
478             if (rowsInserted > 0) {
479                 Alert alert = new Alert(Alert.AlertType.INFORMATION);
480                 alert.setTitle("Message");
481                 alert.setHeaderText(null);
482                 alert.setContentText("Saved Successfully!");
483                 alert.showAndWait();
484             }
485
486             Main m = new Main();
487             m.changeScene("PatientDetailsEntry.fxml");
488
489             statement.close();
490             connection.close();
491
492         }
493
494     } catch (SQLException ae) {

```

```

491         System.out.println("Error!!! Connection Failed");
492         ae.printStackTrace();
493
494         Alert alert = new Alert(Alert.AlertType.ERROR);
495         alert.setTitle("Error Message");
496         alert.setHeaderText(null);
497         alert.setContentText("ERROR Connecting to MySQL Database!!");
498         alert.showAndWait();
499
500     }
501 }
502
503 //checks if all fields are filled correctly
504 public boolean checkAllValues() {
505     boolean checked = false;
506
507     int correct = 0;
508
509     if (nameTextField.getText().isBlank()) {
510         nameErrorMessage.setVisible(true);
511     } else {
512         nameErrorMessage.setVisible(false);
513         correct += 1;
514     }
515
516     if (studentNoTextField.getText().length() == 8) {
517         stdNoErrorMessage.setVisible(false);
518         correct += 1;
519     } else {
520         stdNoErrorMessage.setVisible(true);
521     }
522
523     if (maleRadioButton.isSelected() || femaleRadioButton.isSelected()) {
524         genderErrorMessage.setVisible(false);
525         correct += 1;
526     } else {
527         genderErrorMessage.setVisible(true);
528     }
529
530     if (ageTextField.getText().isBlank()) {
531         ageErrorMessage.setVisible(true);
532     } else {
533         ageErrorMessage.setVisible(false);
534         correct += 1;
535     }
536
537     if (phoneNoTextField.getText().length() == 8) {
538         phoneNoErrorMessage.setVisible(false);
539         correct += 1;
540     } else {
541         phoneNoErrorMessage.setVisible(true);
542     }
543
544     if (complaintTextArea.getText().isBlank()) {
545         complaintErrorMessage.setVisible(true);
546     } else {
547         complaintErrorMessage.setVisible(false);
548         correct += 1;
549     }
550
551     if (remarksTextArea.getText().isBlank()) {

```



```

555         remarksErrorMessage.setVisible(true);
556     } else {
557         remarksErrorMessage.setVisible(false);
558         correct += 1;
559     }
560
561     if (leaveDaysTextField.getText().isBlank()) {
562         leaveDaysErrorMessage.setVisible(true);
563     } else {
564         leaveDaysErrorMessage.setVisible(false);
565         correct += 1;
566     }
567
568     if (year.isEmpty()) {
569         yearErrorMessage.setVisible(true);
570     } else {
571         yearErrorMessage.setVisible(false);
572         correct += 1;
573     }
574
575     if (course.isEmpty()) {
576         courseErrorMessage.setVisible(true);
577     } else {
578         courseErrorMessage.setVisible(false);
579         correct += 1;
580     }
581
582     if (department.isEmpty()) {
583         deptErrorMessage.setVisible(true);
584     } else {
585         deptErrorMessage.setVisible(false);
586         correct += 1;
587     }
588
589     if (leaveOptions.isEmpty()) {
590         leaveOptionsErrorMessage.setVisible(true);
591     } else {
592         leaveOptionsErrorMessage.setVisible(false);
593         correct += 1;
594     }
595
596     if (correct >= 14) {
597         checked = true;
598     }
599
600     return checked;
601 }
602
603 //gets last regNo from database and increments it
604 public void getLastRegNo() {
605     try {
606
607
608         Connection connection =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
        , "root", "root");
609
610         String sql = "SELECT max(reg_no) from patient_records;";
611         Statement statement = connection.createStatement();
612         ResultSet resultset = statement.executeQuery(sql);
613
614         if (resultset.next()) {
615             lastRegNo = resultset.getInt(1);
616             lastRegNo++;

```

```

617         regNoTextField.setText(lastRegNo.toString());
618     }
619
620     else {
621         regNoTextField.setText("100");
622     }
623
624     resultSet.close();
625     statement.close();
626     connection.close();
627
628 }
629
630 catch (SQLException ae) {
631
632     System.out.println("Error!!! Connection Failed");
633     ae.printStackTrace();
634
635     Alert alert = new Alert(Alert.AlertType.ERROR);
636     alert.setTitle("Error Message");
637     alert.setHeaderText(null);
638     alert.setContentText("ERROR Generating Reg Nos!!");
639     alert.showAndWait();
640
641 }
642 }
643
644 //MenuBar
645
646 //Menu 1 - File
647
648 //Menu Item 1.1
649 public void HomePageMenuItem(ActionEvent event) {
650     MenuBar obj = new MenuBar();
651     obj.HomePage();
652 }
653
654 //Menu Item 1.2
655 public void LogoutMenuItem(ActionEvent event) {
656     MenuBar obj = new MenuBar();
657     obj.LogOut();
658 }
659
660 //Menu Item 1.3
661 public void ExitMenuItem(ActionEvent event) {
662     MenuBar obj = new MenuBar();
663     obj.Exit();
664 }
665
666 //Menu 2 - View
667
668 //Menu Item 2.1
669 public void NewPatientEntryMenuItem(ActionEvent event) {
670     MenuBar obj = new MenuBar();
671     obj.NewPatientEntry();
672 }
673
674 //Menu Item 2.2
675 public void ShowPatientRecordMenuItem(ActionEvent event) {
676     MenuBar obj = new MenuBar();
677     obj.ShowPatientRecord();
678 }
679
680 //Menu Item 2.3

```

```
681     public void MedicineRecordsMenuItem(ActionEvent event) {
682         MenuBar obj = new MenuBar();
683         obj.MedicineRecords();
684     }
685
686     //Menu 3 - Help
687
688     //Menu Item 3.1
689     public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
690         MenuBar obj = new MenuBar();
691         obj.CollegeWebsiteLinks();
692     }
693
694     //Menu Item 3.2
695     public void AboutMenuItem(ActionEvent event) {
696         MenuBar obj = new MenuBar();
697         obj.About();
698     }
699
700 }
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.net.URL;
5  import java.sql.*;
6  import java.time.LocalDate;
7  import java.time.LocalTime;
8  import java.time.format.DateTimeFormatter;
9  import java.util.ResourceBundle;
10
11  import javafx.collections.FXCollections;
12  import javafx.collections.ObservableList;
13  import javafx.event.ActionEvent;
14  import javafx.fxml.*;
15  import javafx.fxml.FXML;
16  import javafx.scene.control.*;
17
18  public class StaffPatientDetailsEntryController implements Initializable {
19
20      // Buttons
21
22      @FXML
23      private Button backButton;
24
25      @FXML
26      private Button saveButton;
27
28      // Patient Details Entry Components
29
30      @FXML
31      public TextField nameTextField;
32
33      @FXML
34      private TextField regNoTextField;
35
36      @FXML
37      private RadioButton maleRadioButton;
38
39      @FXML
40      private RadioButton femaleRadioButton;
41
42      @FXML
43      private ToggleGroup gender;
44
45      @FXML
46      private TextField ageTextField;
47
48      @FXML
49      private TextField phoneNoTextField;
50
51      @FXML
52      private DatePicker dateSelector;
53
54      @FXML
55      private TextField timeTextField;
56
57      @FXML
58      private RadioButton amRadioButton;
59
60      @FXML
61      private RadioButton pmRadioButton;
62
63      @FXML
64      private ToggleGroup timeSelect;
```

```
65
66     @FXML
67     private TextArea complaintTextArea;
68
69     @FXML
70     private TextArea remarksTextArea;
71
72     @FXML
73     private TextField leaveDaysTextField;
74
75     // ComboBox
76
77     @FXML
78     public ComboBox<String> departmentComboBox;
79
80     @FXML
81     public ComboBox<String> leaveOptionsComboBox;
82
83     //Medicine
84     @FXML
85     public CheckBox medicineChecBox;
86
87     @FXML
88     public ComboBox<String> medicineComboBox;
89
90     @FXML
91     public TextField medicineQuantityTextField;
92
93     @FXML
94     public CheckBox injectionChecBox;
95
96     @FXML
97     public ComboBox<String> injectionComboBox;
98
99     @FXML
100    public TextField injectionQuantityTextField;
101
102    // Error Labels
103    @FXML
104    private Label nameErrorMessage;
105
106    @FXML
107    private Label ageErrorMessage;
108
109    @FXML
110    private Label genderErrorMessage;
111
112    @FXML
113    private Label deptErrorMessage;
114
115    @FXML
116    private Label phoneNoErrorMessage;
117
118    @FXML
119    private Label complaintErrorMessage;
120
121    @FXML
122    private Label remarksErrorMessage;
123
124    @FXML
125    private Label leaveDaysErrorMessage;
126
127    @FXML
128    private Label leaveOptionsErrorMessage;
```

```

129
130     public String sex, time, department, leaveOptions, ampm;
131     public LocalDate date;
132     public Integer lastRegNo;
133
134     //creating array list for ComboBox
135     ObservableList<String> departmentList = FXCollections.observableArrayList("Management",
"Administration", "Civil Department", "Architecture Department", "Electrical Department",
"IT Department", "ECE Department", "S&H Department", "ICT", "Others");
136     ObservableList<String> leaveOptionList = FXCollections.observableArrayList("No Leave",
"Bed Rest", "Full Day Leave", "Half Day Leave", "Referred to PGH");
137     ObservableList<String> medicineList = FXCollections.observableArrayList();
138     ObservableList<String> injectionList = FXCollections.observableArrayList();
139
140     //method called while class is loaded
141     @Override
142     public void initialize(URL arg0, ResourceBundle arg1) {
143         departmentComboBox.setItems(departmentList);
144         leaveOptionsComboBox.setItems(leaveOptionList);
145         medicineComboBox.setItems(medicineList);
146         injectionComboBox.setItems(injectionList);
147
148         //gets current date and fills up the text field
149         DateSelector.setValue(LocalDate.now());
150         date = DateSelector.getValue();
151
152         //gets current time and fills up the text field
153         String pattern = "hh:mm";
154         timeTextField.setText((LocalTime.now()).format(DateTimeFormatter.ofPattern(pattern)));
155
156         //gets am and pm and selects the radio button
157         String ampmPattern = "hh: mm a";
158         ampm = (LocalTime.now()).format(DateTimeFormatter.ofPattern(ampmPattern));
159         ampm = ampm.substring(6, 7);
160
161         if (ampm.equals("AM")) {
162             amRadioButton.setSelected(true);
163         } else {
164             pmRadioButton.setSelected(true);
165         }
166
167         time = timeTextField.getText() + ampm;
168
169         //calls method to automatically fill up regNo Text field by incrementing the last regNo
        in the database
170         getLastRegNo();
171
172         //fills up medicine Combo Box by selecting available medicines in the database
173         try {
174
175             Connection connection1 =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmary_management_system"
                    , "root", "root");
176             Statement statement1 = connection1.createStatement();
177
178             String sql = "SELECT * FROM medicine_list;";
179             ResultSet resultset1 = statement1.executeQuery(sql);
180
181             while (resultset1.next()) {
182                 String medicineName = resultset1.getString("medicine_name");
183                 medicineComboBox.getItems().addAll(medicineName);
184             }
185
186             statement1.close();

```

```

187         connection1.close();
188         resultset1.close();
189
190     }
191
192     catch (SQLException ae) {
193
194         System.out.println("Error!!! Connection Failed");
195         ae.printStackTrace();
196
197         Alert alert = new Alert(Alert.AlertType.ERROR);
198         alert.setTitle("Error Message");
199         alert.setHeaderText(null);
200         alert.setContentText("ERROR fetching Medicine Name from Database!!");
201         alert.showAndWait();
202
203     }
204
205     //fills up injection Combo Box by selecting available injections in the database
206     try {
207
208         Connection connection2 =
209             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
210                 , "root", "root");
211         Statement statement2 = connection2.createStatement();
212
213         String sql = "SELECT * FROM injection_list;";
214         ResultSet resultset2 = statement2.executeQuery(sql);
215
216         while (resultset2.next()) {
217             String injectionName = resultset2.getString("injection_name");
218             injectionComboBox.getItems().addAll(injectionName);
219         }
220
221         statement2.close();
222         connection2.close();
223         resultset2.close();
224
225     }
226
227     catch (SQLException ae) {
228
229         System.out.println("Error!!! Connection Failed");
230         ae.printStackTrace();
231
232         Alert alert = new Alert(Alert.AlertType.ERROR);
233         alert.setTitle("Error Message");
234         alert.setHeaderText(null);
235         alert.setContentText("ERROR fetching Injection Name from Database!!");
236         alert.showAndWait();
237
238     }
239
240     // stores the item selected in the combo box in a string
241     public void comboChanged (ActionEvent event) throws IOException{
242
243         department = departmentComboBox.getValue();
244         leaveOptions = leaveOptionsComboBox.getValue();
245
246         if (leaveOptions == "No Leave") {
247             leaveDaysTextField.setText("0");
248         }

```

```

249
250     }
251
252     public void homePage(ActionEvent event) throws IOException {
253
254         Main m = new Main();
255         m.changeScene("SelectPatientEntryPage.fxml");
256
257     }
258
259     //gets selected radio button
260     public void sexRadioSelect (ActionEvent event) throws IOException {
261
262         if (maleRadioButton.isSelected()) {
263             sex = "Male";
264         }
265
266         else if (femaleRadioButton.isSelected()) {
267             sex = "Female";
268         }
269
270     }
271
272     //gets selected radio button
273     public void timeRadioSelect (ActionEvent event) throws IOException {
274
275         if (amRadioButton.isSelected()) {
276             time = timeTextField.getText() + " AM";
277         }
278
279         else if (pmRadioButton.isSelected()) {
280             time = timeTextField.getText() + " PM";
281         }
282
283     }
284
285     //updates medicine quantity column in database by subtracting the quantity prescribed
286     public void medicinePrescribed(ActionEvent event) {
287
288         String medicineName = medicineComboBox.getValue();
289
290         try {
291
292             Connection connection =
293                 DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
294                     , "root", "root");
295
296             String sql = "UPDATE medicine_list SET medicine_quantity = (medicine_quantity-?)
297                 WHERE medicine_name = ?;";
298
299             PreparedStatement statement = connection.prepareStatement(sql);
300
301             statement.setDouble(1, Double.parseDouble(medicineQuantityTextField.getText()));
302             statement.setString(2, medicineName);
303
304             statement.executeUpdate();
305
306             statement.close();
307             connection.close();
308
309         }
310
311         catch (SQLException ae) {

```



```

310         System.out.println("Error!!! Connection Failed");
311         ae.printStackTrace();
312
313         Alert alert = new Alert(Alert.AlertType.ERROR);
314         alert.setTitle("Error Message");
315         alert.setHeaderText(null);
316         alert.setContentText("ERROR Prescribing Medicine!!");
317         alert.showAndWait();
318
319     }
320
321 }
322
323 //updates injection quantity column in database by subtracting the quantity prescribed
324 public void prescribeInjection(ActionEvent event) {
325
326     String injectionName = injectionComboBox.getValue();
327
328     try {
329
330         Connection connection =
331             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmary_management_system"
332                 , "root", "root");
333
334         String sql = "UPDATE injection_list SET injection_quantity = (injection_quantity-?)
335             WHERE injection_name = ?";
336
337         PreparedStatement statement = connection.prepareStatement(sql);
338
339         statement.setDouble(1, Double.parseDouble(injectionQuantityTextField.getText()));
340         statement.setString(2, injectionName);
341
342         statement.executeUpdate();
343
344         statement.close();
345         connection.close();
346
347     }
348
349     catch (SQLException ae) {
350
351         System.out.println("Error!!! Connection Failed");
352         ae.printStackTrace();
353
354         Alert alert = new Alert(Alert.AlertType.ERROR);
355         alert.setTitle("Error Message");
356         alert.setHeaderText(null);
357         alert.setContentText("ERROR Prescribing Injection!!");
358         alert.showAndWait();
359
360     }
361
362 }
363
364 //saves details entered in the database
365 public void saveDetails(ActionEvent event) throws IOException {
366
367     boolean allCorrect = checkAllStaffValues();
368
369     if (allCorrect) {
370
371         try {
372
373             date = DateSelector.getValue();
374

```

```

371
372         Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmmary_management_sys
tem", "root", "root");

373
374         String sql = "INSERT INTO patient_records_staffs (reg_no, name, sex, age,
department, phone_no, date, time, leave_options, no_of_days, complaint,
remarks) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

375
376         PreparedStatement statement = connection.prepareStatement(sql);
377
378         statement.setInt(1, Integer.parseInt(regNoTextField.getText()));
379         statement.setString(2, nameTextField.getText());
380         statement.setString(3, sex);
381         statement.setInt(4, Integer.parseInt(ageTextField.getText()));
382         statement.setString(5, department);
383         statement.setInt(6, Integer.parseInt(phoneNoTextField.getText()));
384         statement.setString(7, date.toString());
385         statement.setString(8, time);
386         statement.setString(9, leaveOptions);
387         statement.setInt(10, Integer.parseInt(leaveDaysTextField.getText()));
388         statement.setString(11, complaintTextArea.getText());
389         statement.setString(12, remarksTextArea.getText());
390
391         int rowsInserted = statement.executeUpdate();
392
393         if (rowsInserted > 0) {
394             Alert alert = new Alert(Alert.AlertType.INFORMATION);
395             alert.setTitle("Message");
396             alert.setHeaderText(null);
397             alert.setContentText("Saved Successfully!");
398             alert.showAndWait();
399         }
400
401         Main m = new Main();
402         m.changeScene("StaffPatientDetailsEntry.fxml");
403
404         statement.close();
405         connection.close();
406
407     }
408
409     catch (SQLException ae) {
410
411         System.out.println("Error!!! Connection Failed");
412         ae.printStackTrace();
413
414         Alert alert = new Alert(Alert.AlertType.ERROR);
415         alert.setTitle("Error Message");
416         alert.setHeaderText(null);
417         alert.setContentText("ERROR Connecting to MySQL Database!!");
418         alert.showAndWait();
419
420     }
421 }
422
423
424 // checks if all feilds are filled up
425 public boolean checkAllStaffValues() {
426
427     boolean checked = false;
428
429     int correct = 0;
430

```

```
431     if (nameTextField.getText().isBlank()) {
432         nameErrorMessage.setVisible(true);
433     } else {
434         nameErrorMessage.setVisible(false);
435         correct += 1;
436     }
437
438     if (maleRadioButton.isSelected() || femaleRadioButton.isSelected()) {
439         genderErrorMessage.setVisible(false);
440         correct += 1;
441     } else {
442         genderErrorMessage.setVisible(true);
443     }
444
445     if (ageTextField.getText().isBlank()) {
446         ageErrorMessage.setVisible(true);
447     } else {
448         ageErrorMessage.setVisible(false);
449         correct += 1;
450     }
451
452     if (phoneNoTextField.getText().length() == 8) {
453         phoneNoErrorMessage.setVisible(false);
454         correct += 1;
455     } else {
456         phoneNoErrorMessage.setVisible(true);
457     }
458
459     if (complaintTextArea.getText().isBlank()) {
460         complaintErrorMessage.setVisible(true);
461     } else {
462         complaintErrorMessage.setVisible(false);
463         correct += 1;
464     }
465
466     if (remarksTextArea.getText().isBlank()) {
467         remarksErrorMessage.setVisible(true);
468     } else {
469         remarksErrorMessage.setVisible(false);
470         correct += 1;
471     }
472
473     if (leaveDaysTextField.getText().isBlank()) {
474         leaveDaysErrorMessage.setVisible(true);
475     } else {
476         leaveDaysErrorMessage.setVisible(false);
477         correct += 1;
478     }
479
480     if (department.isEmpty()) {
481         deptErrorMessage.setVisible(true);
482     } else {
483         deptErrorMessage.setVisible(false);
484         correct += 1;
485     }
486
487     if (leaveOptions.isEmpty()) {
488         leaveOptionsErrorMessage.setVisible(true);
489     } else {
490         leaveOptionsErrorMessage.setVisible(false);
491         correct += 1;
492     }
493
494     if (correct >= 9) {
```

```

495         checked = true;
496     }
497
498     return checked;
499 }
500
501 //gets last regNo from database and increments it to be filled up in the text field
502 public void getLastRegNo() {
503
504     try {
505
506         Connection connection =
507             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
508                 , "root", "root");
509
510         String sql = "SELECT max(reg_no) from patient_records_staffs;";
511         Statement statement = connection.createStatement();
512         ResultSet resultset = statement.executeQuery(sql);
513
514         if (resultset.next()) {
515             lastRegNo = resultset.getInt(1);
516             lastRegNo++;
517             regNoTextField.setText(lastRegNo.toString());
518         }
519
520         else {
521             regNoTextField.setText("200");
522         }
523
524         resultset.close();
525         statement.close();
526         connection.close();
527
528     }
529
530     catch (SQLException ae) {
531
532         System.out.println("Error!!! Connection Failed");
533         ae.printStackTrace();
534
535         Alert alert = new Alert(Alert.AlertType.ERROR);
536         alert.setTitle("Error Message");
537         alert.setHeaderText(null);
538         alert.setContentText("ERROR Generating Reg Nos!!");
539         alert.showAndWait();
540     }
541
542     //MenuBar
543
544     //Menu 1 - File
545
546     //Menu Item 1.1
547     public void HomePageMenuItem(ActionEvent event) {
548         MenuBar obj = new MenuBar();
549         obj.HomePage();
550     }
551
552     //Menu Item 1.2
553     public void LogoutMenuItem(ActionEvent event) {
554         MenuBar obj = new MenuBar();
555         obj.LogOut();
556     }

```

```
557
558 //Menu Item 1.3
559 public void ExitMenuItem(ActionEvent event) {
560     MenuBar obj = new MenuBar();
561     obj.Exit();
562 }
563
564 //Menu 2 - View
565
566 //Menu Item 2.1
567 public void NewPatientEntryMenuItem(ActionEvent event) {
568     MenuBar obj = new MenuBar();
569     obj.NewPatientEntry();
570 }
571
572 //Menu Item 2.2
573 public void ShowPatientRecordMenuItem(ActionEvent event) {
574     MenuBar obj = new MenuBar();
575     obj.ShowPatientRecord();
576 }
577
578 //Menu Item 2.3
579 public void MedicineRecordsMenuItem(ActionEvent event) {
580     MenuBar obj = new MenuBar();
581     obj.MedicineRecords();
582 }
583
584 //Menu 3 - Help
585
586 //Menu Item 3.1
587 public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
588     MenuBar obj = new MenuBar();
589     obj.CollegeWebsiteLinks();
590 }
591
592 //Menu Item 3.2
593 public void AboutMenuItem(ActionEvent event) {
594     MenuBar obj = new MenuBar();
595     obj.About();
596 }
597
598 }
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.net.URL;
5  import java.sql.*;
6  import java.time.LocalDate;
7  import java.time.LocalDateTime;
8  import java.time.format.DateTimeFormatter;
9  import java.util.ResourceBundle;
10
11  import javafx.collections.FXCollections;
12  import javafx.collections.ObservableList;
13  import javafx.event.ActionEvent;
14  import javafx.fxml.*;
15  import javafx.fxml.FXML;
16  import javafx.scene.control.*;
17
18  public class OtherPatientDetailsEntryController implements Initializable {
19
20      // Buttons
21
22      @FXML
23      private Button backButton;
24
25      @FXML
26      private Button saveButton;
27
28      // Patient Details Entry Components
29
30      @FXML
31      public TextField nameTextField;
32
33      @FXML
34      private TextField regNoTextField;
35
36      @FXML
37      private RadioButton maleRadioButton;
38
39      @FXML
40      private RadioButton femaleRadioButton;
41
42      @FXML
43      private ToggleGroup gender;
44
45      @FXML
46      private TextField ageTextField;
47
48      @FXML
49      private TextField phoneNoTextField;
50
51      @FXML
52      private DatePicker dateSelector;
53
54      @FXML
55      private TextField timeTextField;
56
57      @FXML
58      private RadioButton amRadioButton;
59
60      @FXML
61      private RadioButton pmRadioButton;
62
63      @FXML
64      private ToggleGroup timeSelect;
```

```

65
66     @FXML
67     private TextArea complaintTextArea;
68
69     @FXML
70     private TextArea remarksTextArea;
71
72     //Medicine
73     @FXML
74     public CheckBox medicineChecBox;
75
76     @FXML
77     public ComboBox<String> medicineComboBox;
78
79     @FXML
80     public TextField medicineQuantityTextField;
81
82     @FXML
83     public CheckBox injectionChecBox;
84
85     @FXML
86     public ComboBox<String> injectionComboBox;
87
88     @FXML
89     public TextField injectionQuantityTextField;
90
91     // Error Labels
92     @FXML
93     private Label nameErrorMessage;
94
95     @FXML
96     private Label ageErrorMessage;
97
98     @FXML
99     private Label genderErrorMessage;
100
101     @FXML
102     private Label phoneNoErrorMessage;
103
104     @FXML
105     private Label complaintErrorMessage;
106
107     @FXML
108     private Label remarksErrorMessage;
109
110     public String sex, time, ampm;
111     public LocalDate date;
112     public Integer lastRegNo;
113
114     ObservableList<String> medicineList = FXCollections.observableArrayList();
115     ObservableList<String> injectionList = FXCollections.observableArrayList();
116
117     @Override
118     public void initialize(URL arg0, ResourceBundle arg1) {
119         medicineComboBox.setItems(medicineList);
120         injectionComboBox.setItems(injectionList);
121
122         //automatic date select in date selector
123         DateSelector.setValue(LocalDate.now());
124         date = DateSelector.getValue();
125
126         //get time and fill in textfield
127         String pattern = "hh:mm";
128

```

```

timeTextField.setText((LocalTime.now()).format(DateTimeFormatter.ofPattern(pattern)))
;
129
130 //select radio button according to current time
131 String ampmPattern = "hh: mm a";
132 ampm = (LocalTime.now()).format(DateTimeFormatter.ofPattern(ampmPattern));
133 ampm = ampm.substring(6, 7);
134
135 if (ampm.equals("AM")) {
136     amRadioButton.setSelected(true);
137 } else {
138     pmRadioButton.setSelected(true);
139 }
140
141 time = timeTextField.getText() + ampm;
142
143 //get last reg no from database
144 getLastRegNo();
145
146 //get medicine combo box items from database
147 try {
148
149     Connection connection1 =
150     DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarium_management_sys
151     tem", "root", "root");
152     Statement statement1 = connection1.createStatement();
153
154     String sql = "SELECT * FROM medicine_list;";
155     ResultSet resultset1 = statement1.executeQuery(sql);
156
157     while (resultset1.next()) {
158         String medicineName = resultset1.getString("medicine_name");
159         medicineComboBox.getItems().addAll(medicineName);
160     }
161
162     statement1.close();
163     connection1.close();
164     resultset1.close();
165
166 }
167
168 catch (SQLException ae) {
169
170     System.out.println("Error!!! Connection Failed");
171     ae.printStackTrace();
172
173     Alert alert = new Alert(Alert.AlertType.ERROR);
174     alert.setTitle("Error Message");
175     alert.setHeaderText(null);
176     alert.setContentText("ERROR fetching Medicine Name from Database!!");
177     alert.showAndWait();
178
179 }
180
181 // injection combo box items from database
182 try {
183
184     Connection connection2 =
185     DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarium_management_sys
186     tem", "root", "root");
187     Statement statement2 = connection2.createStatement();
188
189     String sql = "SELECT * FROM injection_list;";
190     ResultSet resultset2 = statement2.executeQuery(sql);

```



```

187
188         while (resultset2.next()) {
189             String injectionName = resultset2.getString("injection_name");
190             injectionComboBox.getItems().addAll(injectionName);
191         }
192
193         statement2.close();
194         connection2.close();
195         resultset2.close();
196
197     }
198
199     catch (SQLException ae) {
200
201         System.out.println("Error!!! Connection Failed");
202         ae.printStackTrace();
203
204         Alert alert = new Alert(Alert.AlertType.ERROR);
205         alert.setTitle("Error Message");
206         alert.setHeaderText(null);
207         alert.setContentText("ERROR fetching Injection Name from Database!!");
208         alert.showAndWait();
209
210     }
211
212 }
213
214 public void homePage(ActionEvent event) throws IOException {
215
216     Main m = new Main();
217     m.changeScene("SelectPatientEntryPage.fxml");
218
219 }
220
221 //get selected radio button
222 public void sexRadioSelect (ActionEvent event) throws IOException {
223
224     if (maleRadioButton.isSelected()) {
225         sex = "Male";
226     }
227
228     else if (femaleRadioButton.isSelected()) {
229         sex = "Female";
230     }
231
232 }
233
234 public void timeRadioSelect (ActionEvent event) throws IOException {
235
236     if (amRadioButton.isSelected()) {
237         time = timeTextField.getText() + " AM";
238     }
239
240     else if (pmRadioButton.isSelected()) {
241         time = timeTextField.getText() + " PM";
242     }
243
244 }
245
246 //update medicine reocrds table after medicine is prescribed
247 public void medicinePrescribed(ActionEvent event) {
248
249     String medicineName = medicineComboBox.getValue();
250

```

```

251     try {
252
253         Connection connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarium_management_sys
            tem", "root", "root");

254
255         String sql = "UPDATE medicine_list SET medicine_quantity =
            (medicine_quantity-?) WHERE medicine_name = ?;";

256
257         PreparedStatement statement = connection.prepareStatement(sql);
258
259         statement.setDouble(1, Double.parseDouble(medicineQuantityTextField.getText()));
260         statement.setString(2, medicineName);
261
262         statement.executeUpdate();
263
264         statement.close();
265         connection.close();
266
267     }
268
269     catch (SQLException ae) {
270
271         System.out.println("Error!!! Connection Failed");
272         ae.printStackTrace();
273
274         Alert alert = new Alert(Alert.AlertType.ERROR);
275         alert.setTitle("Error Message");
276         alert.setHeaderText(null);
277         alert.setContentText("ERROR Prescribing Medicine!!");
278         alert.showAndWait();
279
280     }
281
282 }
283
284 //update injection records table when injection is prescribed
285 public void prescribeInjection(ActionEvent event) {
286
287     String injectionName = injectionComboBox.getValue();
288
289     try {
290
291         Connection connection =
292             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarium_management_sys
293             tem", "root", "root");

294
295         String sql = "UPDATE injection_list SET injection_quantity =
            (injection_quantity-?) WHERE injection_name = ?;";

296
297         PreparedStatement statement = connection.prepareStatement(sql);
298
299         statement.setDouble(1, Double.parseDouble(injectionQuantityTextField.getText()));
300         statement.setString(2, injectionName);
301
302         statement.executeUpdate();
303
304         statement.close();
305         connection.close();
306
307     }
308
309     catch (SQLException ae) {

```

```

309         System.out.println("Error!!! Connection Failed");
310         ae.printStackTrace();
311
312         Alert alert = new Alert(Alert.AlertType.ERROR);
313         alert.setTitle("Error Message");
314         alert.setHeaderText(null);
315         alert.setContentText("ERROR Prescribing Injection!!");
316         alert.showAndWait();
317
318     }
319
320 }
321
322 //save details in database
323 public void saveDetails(ActionEvent event) throws IOException {
324
325     boolean allCorrect = checkAllStaffValues();
326
327     if (allCorrect) {
328
329         try {
330
331             date = DateSelector.getValue();
332
333             Connection connection =
334                 DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarary_management
335                 _system", "root", "root");
336
337             String sql = "INSERT INTO patient_records_others (reg_no, name, sex, age,
338             phone_no, date, time, complaint, remarks) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
339
340             PreparedStatement statement = connection.prepareStatement(sql);
341
342             statement.setInt(1, Integer.parseInt(regNoTextField.getText()));
343             statement.setString(2, nameTextField.getText());
344             statement.setString(3, sex);
345             statement.setInt(4, Integer.parseInt(ageTextField.getText()));
346             statement.setInt(5, Integer.parseInt(phoneNoTextField.getText()));
347             statement.setString(6, date.toString());
348             statement.setString(7, time);
349             statement.setString(8, complaintTextArea.getText());
350             statement.setString(9, remarksTextArea.getText());
351
352             int rowsInserted = statement.executeUpdate();
353
354             if (rowsInserted > 0) {
355                 Alert alert = new Alert(Alert.AlertType.INFORMATION);
356                 alert.setTitle("Message");
357                 alert.setHeaderText(null);
358                 alert.setContentText("Saved Successfully!");
359                 alert.showAndWait();
360             }
361
362             Main m = new Main();
363             m.changeScene("OtherPatientDetailsEntry.fxml");
364
365             statement.close();
366             connection.close();
367
368         }
369
370     } catch (SQLException ae) {

```

```

369         System.out.println("Error!!! Connection Failed");
370         ae.printStackTrace();
371
372         Alert alert = new Alert(Alert.AlertType.ERROR);
373         alert.setTitle("Error Message");
374         alert.setHeaderText(null);
375         alert.setContentText("ERROR Connecting to MySQL Database!!");
376         alert.showAndWait();
377
378     }
379 }
380
381
382 //check if all fields are filled
383 public boolean checkAllStaffValues() {
384
385     boolean checked = false;
386
387     int correct = 0;
388
389     if (nameTextField.getText().isBlank()) {
390         nameErrorMessage.setVisible(true);
391     } else {
392         nameErrorMessage.setVisible(false);
393         correct += 1;
394     }
395
396     if (maleRadioButton.isSelected() || femaleRadioButton.isSelected()) {
397         genderErrorMessage.setVisible(false);
398         correct += 1;
399     } else {
400         genderErrorMessage.setVisible(true);
401     }
402
403     if (ageTextField.getText().isBlank()) {
404         ageErrorMessage.setVisible(true);
405     } else {
406         ageErrorMessage.setVisible(false);
407         correct += 1;
408     }
409
410     if (phoneNoTextField.getText().length() == 8) {
411         phoneNoErrorMessage.setVisible(false);
412         correct += 1;
413     } else {
414         phoneNoErrorMessage.setVisible(true);
415     }
416
417     if (complaintTextArea.getText().isBlank()) {
418         complaintErrorMessage.setVisible(true);
419     } else {
420         complaintErrorMessage.setVisible(false);
421         correct += 1;
422     }
423
424     if (remarksTextArea.getText().isBlank()) {
425         remarksErrorMessage.setVisible(true);
426     } else {
427         remarksErrorMessage.setVisible(false);
428         correct += 1;
429     }
430
431     if (correct >= 6) {
432         checked = true;

```

```

433     }
434
435     return checked;
436 }
437
438 //get last reg no from database, increment it and update the reg no text field
439 public void getLastRegNo() {
440
441     try {
442
443         Connection connection =
444             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarary_management_sys
445 tem", "root", "root");
446
447         String sql = "SELECT max(reg_no) from patient_records_others;";
448         Statement statement = connection.createStatement();
449         ResultSet resultset = statement.executeQuery(sql);
450
451         if (resultset.next()) {
452             lastRegNo = resultset.getInt(1);
453             lastRegNo++;
454             regNoTextField.setText(lastRegNo.toString());
455         }
456
457         else {
458             regNoTextField.setText("300");
459         }
460
461         resultset.close();
462         statement.close();
463         connection.close();
464
465     }
466
467     catch (SQLException ae) {
468
469         System.out.println("Error!!! Connection Failed");
470         ae.printStackTrace();
471
472         Alert alert = new Alert(Alert.AlertType.ERROR);
473         alert.setTitle("Error Message");
474         alert.setHeaderText(null);
475         alert.setContentText("ERROR Generating Reg Nos!!");
476         alert.showAndWait();
477     }
478
479 //MenuBar
480
481 //Menu 1 - File
482
483 //Menu Item 1.1
484 public void HomePageMenuItem(ActionEvent event) {
485     MenuBar obj = new MenuBar();
486     obj.HomePage();
487 }
488
489 //Menu Item 1.2
490 public void LogOutMenuItem(ActionEvent event) {
491     MenuBar obj = new MenuBar();
492     obj.LogOut();
493 }
494

```

```
495 //Menu Item 1.3
496 public void ExitMenuItem(ActionEvent event) {
497     MenuBar obj = new MenuBar();
498     obj.Exit();
499 }
500
501 //Menu 2 - View
502
503 //Menu Item 2.1
504 public void NewPatientEntryMenuItem(ActionEvent event) {
505     MenuBar obj = new MenuBar();
506     obj.NewPatientEntry();
507 }
508
509 //Menu Item 2.2
510 public void ShowPatientRecordMenuItem(ActionEvent event) {
511     MenuBar obj = new MenuBar();
512     obj.ShowPatientRecord();
513 }
514
515 //Menu Item 2.3
516 public void MedicineRecordsMenuItem(ActionEvent event) {
517     MenuBar obj = new MenuBar();
518     obj.MedicineRecords();
519 }
520
521 //Menu 3 - Help
522
523 //Menu Item 3.1
524 public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
525     MenuBar obj = new MenuBar();
526     obj.CollegeWebsiteLinks();
527 }
528
529 //Menu Item 3.2
530 public void AboutMenuItem(ActionEvent event) {
531     MenuBar obj = new MenuBar();
532     obj.About();
533 }
534
535 }
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.sql.Connection;
5  import java.sql.DriverManager;
6  import java.sql.PreparedStatement;
7  import java.sql.ResultSet;
8  import java.sql.SQLException;
9  import java.sql.Statement;
10
11  import javafx.collections.FXCollections;
12  import javafx.collections.ObservableList;
13  import javafx.event.ActionEvent;
14  import javafx.fxml.FXML;
15  import javafx.scene.control.Alert;
16  import javafx.scene.control.Button;
17  import javafx.scene.control.ButtonType;
18  import javafx.scene.control.TableColumn;
19  import javafx.scene.control.TableView;
20  import javafx.scene.control.TextField;
21
22  public class PatientRecordsController {
23
24
25      //table columns and table
26      @FXML
27      public TableView<Patient> recordsTable;
28      @FXML
29      public TableColumn<Patient, Integer> col_regNo;
30      @FXML
31      public TableColumn<Patient, String> col_name;
32      @FXML
33      public TableColumn<Patient, Integer> col_stdNo;
34      @FXML
35      public TableColumn<Patient, Integer> col_year;
36      @FXML
37      public TableColumn<Patient, String> col_course;
38      @FXML
39      public TableColumn<Patient, String> col_dept;
40      @FXML
41      public TableColumn<Patient, String> col_sex;
42      @FXML
43      public TableColumn<Patient, Integer> col_age;
44      @FXML
45      public TableColumn<Patient, Integer> col_phoneNo;
46      @FXML
47      public TableColumn<Patient, String> col_date;
48      @FXML
49      public TableColumn<Patient, String> col_time;
50      @FXML
51      public TableColumn<Patient, String> col_leaveOptions;
52      @FXML
53      public TableColumn<Patient, Integer> col_leaveDays;
54      @FXML
55      public TableColumn<Patient, String> col_complaint;
56      @FXML
57      public TableColumn<Patient, String> col_remarks;
58
59      @FXML
60      private Button backButton;
61
62      @FXML
63      private Button logoutButton;
64
```

```

65     @FXML
66     private TextField filterByTextField;
67
68     @FXML
69     private Button searchButton;
70
71     @FXML
72     private Button refreshButton;
73
74     @FXML
75     private Button deleteButton;
76
77     public void backToHomePage(ActionEvent event) throws IOException {
78
79         Main m = new Main();
80         m.changeScene("SelectPatientRecordsPage.fxml");
81
82     }
83
84     public void logout(ActionEvent event) throws IOException {
85
86         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
87         alert.setTitle("Logout");
88         alert.setHeaderText(null);
89         alert.setContentText("Are you sure you want to Logout?");
90         if (alert.showAndWait().get() == ButtonType.OK){
91             Main m = new Main();
92             m.changeScene("LoginPage.fxml");
93         }
94     }
95
96     //filters table data
97     public void filterTable(ActionEvent event) throws IOException {
98
99         col_regNo.setCellValueFactory(cellData -> cellData.getValue().get_RegNo().asObject());
100        col_name.setCellValueFactory(cellData -> cellData.getValue().get_Name());
101        col_stdNo.setCellValueFactory(cellData -> cellData.getValue().get_StdNo().asObject());
102        col_sex.setCellValueFactory(cellData -> cellData.getValue().get_Sex());
103        col_age.setCellValueFactory(cellData -> cellData.getValue().get_Age().asObject());
104        col_year.setCellValueFactory(cellData -> cellData.getValue().get_Year().asObject());
105        col_course.setCellValueFactory(cellData -> cellData.getValue().get_Course());
106        col_dept.setCellValueFactory(cellData -> cellData.getValue().get_Dept());
107        col_phoneNo.setCellValueFactory(cellData ->
108            cellData.getValue().get_PhoneNo().asObject());
109        col_date.setCellValueFactory(cellData -> cellData.getValue().get_Date());
110        col_time.setCellValueFactory(cellData -> cellData.getValue().get_Time());
111        col_leaveOptions.setCellValueFactory(cellData -> cellData.getValue().get_leaveOptions());
112        col_leaveDays.setCellValueFactory(cellData ->
113            cellData.getValue().get_LeaveDays().asObject());
114        col_complaint.setCellValueFactory(cellData -> cellData.getValue().get_Complaint());
115        col_remarks.setCellValueFactory(cellData -> cellData.getValue().get_Remarks());
116
117        ObservableList<Patient> patientList = searchPatient(filterByTextField.getText());
118        populateTable(patientList);
119
120    }
121
122    //search patient details from database and passes it as a list to display it
123    public static ObservableList<Patient> searchPatient(String str) {
124
125        ObservableList<Patient> patientList = FXCollections.observableArrayList();
126
127        try {

```



```

127         Connection connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmarium_management_system"
            , "root", "root");
128         Statement statement = connection.createStatement();
129
130         String sql = "SELECT * FROM patient_records WHERE '" + str + "' IN (reg_no, name,
            student_no, sex, age, year, course, department, phone_no, date, time,
            leave_options, no_of_days, complaint, remarks)";
131         ResultSet resultset = statement.executeQuery(sql);
132
133         patientList = getPatientObject(resultset);
134
135     }
136
137     catch (SQLException ae) {
138
139         System.out.println("Error!!! Connection Failed");
140         ae.printStackTrace();
141
142         Alert alert = new Alert(Alert.AlertType.ERROR);
143         alert.setTitle("Error Message");
144         alert.setHeaderText(null);
145         alert.setContentText("ERROR Searching for '" + str + "' in the Database!!!");
146         alert.showAndWait();
147
148     }
149
150     return patientList;
151
152 }
153
154 @FXML
155 public void initialize() throws Exception {
156
157     filterByTextField.setText(null);
158
159     //get column data
160     col_regNo.setCellValueFactory(cellData -> cellData.getValue().get_RegNo().asObject());
161     col_name.setCellValueFactory(cellData -> cellData.getValue().get_Name());
162     col_stdNo.setCellValueFactory(cellData -> cellData.getValue().get_StdNo().asObject());
163     col_sex.setCellValueFactory(cellData -> cellData.getValue().get_Sex());
164     col_age.setCellValueFactory(cellData -> cellData.getValue().get_Age().asObject());
165     col_year.setCellValueFactory(cellData -> cellData.getValue().get_Year().asObject());
166     col_course.setCellValueFactory(cellData -> cellData.getValue().get_Course());
167     col_dept.setCellValueFactory(cellData -> cellData.getValue().get_Dept());
168     col_phoneNo.setCellValueFactory(cellData ->
        cellData.getValue().get_PhoneNo().asObject());
169     col_date.setCellValueFactory(cellData -> cellData.getValue().get_Date());
170     col_time.setCellValueFactory(cellData -> cellData.getValue().get_Time());
171     col_leaveOptions.setCellValueFactory(cellData -> cellData.getValue().get_leaveOptions());
172     col_leaveDays.setCellValueFactory(cellData ->
        cellData.getValue().get_LeaveDays().asObject());
173     col_complaint.setCellValueFactory(cellData -> cellData.getValue().get_Complaint());
174     col_remarks.setCellValueFactory(cellData -> cellData.getValue().get_Remarks());
175
176     ObservableList<Patient> patientList = getPatientRecords();
177     populateTable(patientList);
178
179 }
180
181 //add lists to table
182 public void populateTable (ObservableList<Patient> patientList) {
183     recordsTable.setItems(patientList);
184 }

```

```

185
186 //selects all data from database
187 public static ObservableList<Patient> getPatientRecords() {
188
189     ObservableList<Patient> patientList = FXCollections.observableArrayList();
190
191     try {
192
193         Connection connection =
194             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmary_management_system"
195                 , "root", "root");
196         Statement statement = connection.createStatement();
197
198         String sql = "SELECT * FROM patient_records;";
199         ResultSet resultset = statement.executeQuery(sql);
200
201         patientList = getPatientObject(resultset);
202
203     }
204
205     catch (SQLException ae) {
206
207         System.out.println("Error!!! Connection Failed");
208         ae.printStackTrace();
209
210         Alert alert = new Alert(Alert.AlertType.ERROR);
211         alert.setTitle("Error Message");
212         alert.setHeaderText(null);
213         alert.setContentText("ERROR Connecting to MySQL Database!!");
214         alert.showAndWait();
215
216     }
217
218     return patientList;
219 }
220
221 //gets each column data
222 public static ObservableList<Patient> getPatientObject(ResultSet resultset) {
223
224     ObservableList<Patient> patientRecordList = FXCollections.observableArrayList();
225
226     try {
227
228         int count = 0;
229
230         while (resultset.next()) {
231
232             count++;
233
234             Patient obj = new Patient();
235
236             obj.setRegNo(resultset.getInt("reg_no"));
237             obj.setName(resultset.getString("name"));
238             obj.setStdNo(resultset.getInt("student_no"));
239             obj.setSex(resultset.getString("sex"));
240             obj.setAge(resultset.getInt("age"));
241             obj.setYear(resultset.getInt("year"));
242             obj.setCourse(resultset.getString("course"));
243             obj.setDept(resultset.getString("department"));
244             obj.setPhoneNo(resultset.getInt("phone_no"));
245             obj.setDate(resultset.getString("date"));
246             obj.setTime(resultset.getString("time"));

```

```

247         obj.setLeaveOptions(resultset.getString("leave_options"));
248         obj.setLeaveDays(resultset.getInt("no_of_days"));
249         obj.setComplaint(resultset.getString("complaint"));
250         obj.setRemarks(resultset.getString("remarks"));
251
252         patientRecordList.add(obj);
253
254     }
255
256     if (count==0) {
257
258         new Main().changeScene("PatientRecords.fxml");
259
260         Alert alert = new Alert(Alert.AlertType.ERROR);
261         alert.setTitle("Error Message");
262         alert.setHeaderText(null);
263         alert.setContentText("No Data in MySQL Database!!");
264         alert.showAndWait();
265     }
266
267 }
268
269 catch (SQLException ae) {
270
271     System.out.println("Error!!! Connection Failed");
272     ae.printStackTrace();
273
274     Alert alert = new Alert(Alert.AlertType.ERROR);
275     alert.setTitle("Error Message");
276     alert.setHeaderText(null);
277     alert.setContentText("ERROR Retriving Data from MySQL Database!!");
278     alert.showAndWait();
279
280 }
281 return patientRecordList;
282
283 }
284
285 //deletes selected row from database by passing regNo
286 public void deleteRow(ActionEvent event) throws IOException, NullPointerException{
287
288     Patient selectedRow = null;
289     int regNoValue;
290
291     try {
292
293         selectedRow = recordsTable.getSelectionModel().getSelectedItem();
294         regNoValue = selectedRow.getRegNo();
295
296         Connection connection =
297             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
298                 , "root", "root");
299
300         String sql = "DELETE FROM patient_records WHERE reg_no = ?";
301
302         PreparedStatement statement = connection.prepareStatement(sql);
303
304         statement.setInt(1, regNoValue);
305         statement.execute();
306
307         new Main().changeScene("PatientRecords.fxml");
308     }

```

```

309         catch (SQLException ae) {
310
311             System.out.println("Error!!! Connection Failed");
312             ae.printStackTrace();
313
314             Alert alert = new Alert(Alert.AlertType.ERROR);
315             alert.setTitle("Error Message");
316             alert.setHeaderText(null);
317             alert.setContentText("ERROR Deleting Selected Row!!");
318             alert.showAndWait();
319
320         }
321
322         catch (NullPointerException ae) {
323
324             System.out.println("Error!!! Retriving Reg_No");
325             ae.printStackTrace();
326
327             Alert alert = new Alert(Alert.AlertType.ERROR);
328             alert.setTitle("Error Message");
329             alert.setHeaderText(null);
330             alert.setContentText("ERROR!!! Please Select a Row!!");
331             alert.showAndWait();
332
333         }
334     }
335 }
336
337 //MenuBar
338
339 //Menu 1 - File
340
341 //Menu Item 1.1
342 public void HomePageMenuItem(ActionEvent event) {
343     MenuBar obj = new MenuBar();
344     obj.HomePage();
345 }
346
347 //Menu Item 1.2
348 public void LogoutMenuItem(ActionEvent event) {
349     MenuBar obj = new MenuBar();
350     obj.LogOut();
351 }
352
353 //Menu Item 1.3
354 public void ExitMenuItem(ActionEvent event) {
355     MenuBar obj = new MenuBar();
356     obj.Exit();
357 }
358
359 //Menu 2 - View
360
361 //Menu Item 2.1
362 public void NewPatientEntryMenuItem(ActionEvent event) {
363     MenuBar obj = new MenuBar();
364     obj.NewPatientEntry();
365 }
366
367 //Menu Item 2.2
368 public void ShowPatientRecordMenuItem(ActionEvent event) {
369     MenuBar obj = new MenuBar();
370     obj.ShowPatientRecord();
371 }
372

```

```
373 //Menu Item 2.3
374 public void MedicineRecordsMenuItem(ActionEvent event) {
375     MenuBar obj = new MenuBar();
376     obj.MedicineRecords();
377 }
378
379 //Menu 3 - Help
380
381 //Menu Item 3.1
382 public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
383     MenuBar obj = new MenuBar();
384     obj.CollegeWebsiteLinks();
385 }
386
387 //Menu Item 3.2
388 public void AboutMenuItem(ActionEvent event) {
389     MenuBar obj = new MenuBar();
390     obj.About();
391 }
392
393 }
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.sql.Connection;
5  import java.sql.DriverManager;
6  import java.sql.PreparedStatement;
7  import java.sql.ResultSet;
8  import java.sql.SQLException;
9  import java.sql.Statement;
10
11  import javafx.collections.FXCollections;
12  import javafx.collections.ObservableList;
13  import javafx.event.ActionEvent;
14  import javafx.fxml.FXML;
15  import javafx.scene.control.Alert;
16  import javafx.scene.control.Button;
17  import javafx.scene.control.ButtonType;
18  import javafx.scene.control.TableColumn;
19  import javafx.scene.control.TableView;
20  import javafx.scene.control.TextField;
21
22  public class StaffPatientRecordsController {
23
24      //Importing Table and Table Columns
25
26      @FXML
27      public TableView<StaffPatient> recordsTable;
28      @FXML
29      public TableColumn<StaffPatient, Integer> col_regNo;
30      @FXML
31      public TableColumn<StaffPatient, String> col_name;
32      @FXML
33      public TableColumn<StaffPatient, String> col_dept;
34      @FXML
35      public TableColumn<StaffPatient, String> col_sex;
36      @FXML
37      public TableColumn<StaffPatient, Integer> col_age;
38      @FXML
39      public TableColumn<StaffPatient, Integer> col_phoneNo;
40      @FXML
41      public TableColumn<StaffPatient, String> col_date;
42      @FXML
43      public TableColumn<StaffPatient, String> col_time;
44      @FXML
45      public TableColumn<StaffPatient, String> col_leaveOptions;
46      @FXML
47      public TableColumn<StaffPatient, Integer> col_leaveDays;
48      @FXML
49      public TableColumn<StaffPatient, String> col_complaint;
50      @FXML
51      public TableColumn<StaffPatient, String> col_remarks;
52
53      @FXML
54      private Button backButton;
55
56      @FXML
57      private Button logoutButton;
58
59      @FXML
60      private TextField filterByTextField;
61
62      @FXML
63      private Button searchButton;
64  }
```

```

65     @FXML
66     private Button refreshButton;
67
68     @FXML
69     private Button deleteButton;
70
71     public void backToHomePage(ActionEvent event) throws IOException {
72
73         Main m = new Main();
74         m.changeScene("SelectPatientRecordsPage.fxml");
75
76     }
77
78     public void logout(ActionEvent event) throws IOException {
79
80         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
81         alert.setTitle("Logout");
82         alert.setHeaderText(null);
83         alert.setContentText("Are you sure you want to Logout?");
84         if (alert.showAndWait().get() == ButtonType.OK){
85             Main m = new Main();
86             m.changeScene("LoginPage.fxml");
87         }
88     }
89
90     //method to display filtered data
91     public void filterTable(ActionEvent event) throws IOException {
92
93         col_regNo.setCellValueFactory(cellData -> cellData.getValue().get_RegNo().asObject());
94         col_name.setCellValueFactory(cellData -> cellData.getValue().get_Name());
95         col_sex.setCellValueFactory(cellData -> cellData.getValue().get_Sex());
96         col_age.setCellValueFactory(cellData -> cellData.getValue().get_Age().asObject());
97         col_dept.setCellValueFactory(cellData -> cellData.getValue().get_Dept());
98         col_phoneNo.setCellValueFactory(cellData ->
99         cellData.getValue().get_PhoneNo().asObject());
100        col_date.setCellValueFactory(cellData -> cellData.getValue().get_Date());
101        col_time.setCellValueFactory(cellData -> cellData.getValue().get_Time());
102        col_leaveOptions.setCellValueFactory(cellData -> cellData.getValue().get_leaveOptions());
103        col_leaveDays.setCellValueFactory(cellData ->
104        cellData.getValue().get_LeaveDays().asObject());
105        col_complaint.setCellValueFactory(cellData -> cellData.getValue().get_Complaint());
106        col_remarks.setCellValueFactory(cellData -> cellData.getValue().get_Remarks());
107
108        //calling search method by passing the string entered in the text field
109        ObservableList<StaffPatient> patientList = searchPatient(filterByTextField.getText());
110        populateTable(patientList);
111    }
112
113    //search passed string in any column of database
114    public static ObservableList<StaffPatient> searchPatient(String str) {
115
116        ObservableList<StaffPatient> patientList = FXCollections.observableArrayList();
117
118        try {
119
120            Connection connection =
121            DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmiry_management_system"
122            , "root", "root");
123            Statement statement = connection.createStatement();
124
125            String sql = "SELECT * FROM patient_records_staffs WHERE '" + str + "' IN (reg_no,
126            name, sex, age, department, phone_no, date, time, leave_options, no_of_days,
127            complaint, remarks);";

```

```

123         ResultSet resultset = statement.executeQuery(sql);
124
125         patientList = getPatientObject(resultset);
126
127     }
128
129     catch (SQLException ae) {
130
131         System.out.println("Error!!! Connection Failed");
132         ae.printStackTrace();
133
134         Alert alert = new Alert(Alert.AlertType.ERROR);
135         alert.setTitle("Error Message");
136         alert.setHeaderText(null);
137         alert.setContentText("ERROR Searching for '" + str + "' in the Database!!!");
138         alert.showAndWait();
139
140     }
141
142     return patientList;
143
144 }
145
146 //method called to add data in columns while class is loaded
147 @FXML
148 public void initialize() throws Exception {
149
150     filterByTextField.setText(null);
151
152     col_regNo.setCellValueFactory(cellData -> cellData.getValue().get_RegNo().asObject());
153     col_name.setCellValueFactory(cellData -> cellData.getValue().get_Name());
154     col_sex.setCellValueFactory(cellData -> cellData.getValue().get_Sex());
155     col_age.setCellValueFactory(cellData -> cellData.getValue().get_Age().asObject());
156     col_dept.setCellValueFactory(cellData -> cellData.getValue().get_Dept());
157     col_phoneNo.setCellValueFactory(cellData ->
158         cellData.getValue().get_PhoneNo().asObject());
159     col_date.setCellValueFactory(cellData -> cellData.getValue().get_Date());
160     col_time.setCellValueFactory(cellData -> cellData.getValue().get_Time());
161     col_leaveOptions.setCellValueFactory(cellData -> cellData.getValue().get_leaveOptions());
162     col_leaveDays.setCellValueFactory(cellData ->
163         cellData.getValue().get_LeaveDays().asObject());
164     col_complaint.setCellValueFactory(cellData -> cellData.getValue().get_Complaint());
165     col_remarks.setCellValueFactory(cellData -> cellData.getValue().get_Remarks());
166
167     ObservableList<StaffPatient> patientList = getPatientRecords();
168     populateTable(patientList);
169
170 }
171
172 //method to add all items to table
173 public void populateTable (ObservableList<StaffPatient> patientList) {
174     recordsTable.setItems(patientList);
175 }
176
177 //get table data from database
178 public static ObservableList<StaffPatient> getPatientRecords() {
179
180     ObservableList<StaffPatient> patientList = FXCollections.observableArrayList();
181
182     try {
183
184         Connection connection =
185             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmiry_management_system"
186                 , "root", "root");

```



```

183         Statement statement = connection.createStatement();
184
185         String sql = "SELECT * FROM patient_records_staffs;";
186         ResultSet resultset = statement.executeQuery(sql);
187
188         patientList = getPatientObject(resultset);
189
190     }
191
192     catch (SQLException ae) {
193
194         System.out.println("Error!!! Connection Failed");
195         ae.printStackTrace();
196
197         Alert alert = new Alert(Alert.AlertType.ERROR);
198         alert.setTitle("Error Message");
199         alert.setHeaderText(null);
200         alert.setContentText("ERROR Connecting to MySQL Database!!");
201         alert.showAndWait();
202
203     }
204
205     return patientList;
206
207 }
208
209 //assign data to columns
210 public static ObservableList<StaffPatient> getPatientObject(ResultSet resultset) {
211
212     ObservableList<StaffPatient> patientRecordList = FXCollections.observableArrayList();
213
214     try {
215
216         int count = 0;
217
218         while (resultset.next()) {
219
220             count++;
221
222             StaffPatient obj = new StaffPatient();
223
224             obj.setRegNo(resultset.getInt("reg_no"));
225             obj.setName(resultset.getString("name"));
226             obj.setSex(resultset.getString("sex"));
227             obj.setAge(resultset.getInt("age"));
228             obj.setDept(resultset.getString("department"));
229             obj.setPhoneNo(resultset.getInt("phone_no"));
230             obj.setDate(resultset.getString("date"));
231             obj.setTime(resultset.getString("time"));
232             obj.setLeaveOptions(resultset.getString("leave_options"));
233             obj.setLeaveDays(resultset.getInt("no_of_days"));
234             obj.setComplaint(resultset.getString("complaint"));
235             obj.setRemarks(resultset.getString("remarks"));
236
237             patientRecordList.add(obj);
238
239         }
240
241         if (count==0) {
242
243             new Main().changeScene("StaffPatientRecords.fxml");
244
245             Alert alert = new Alert(Alert.AlertType.ERROR);
246             alert.setTitle("Error Message");

```

```

247         alert.setHeaderText(null);
248         alert.setContentText("No Data in MySQL Database!!");
249         alert.showAndWait();
250     }
251
252 }
253
254 catch (SQLException ae) {
255
256     System.out.println("Error!!! Connection Failed");
257     ae.printStackTrace();
258
259     Alert alert = new Alert(Alert.AlertType.ERROR);
260     alert.setTitle("Error Message");
261     alert.setHeaderText(null);
262     alert.setContentText("ERROR Retriving Data from MySQL Database!!");
263     alert.showAndWait();
264
265 }
266 return patientRecordList;
267
268 }
269
270 //method to delete selected row from database and table view
271 public void deleteRow(ActionEvent event) throws IOException, NullPointerException{
272
273     StaffPatient selectedRow = null;
274     int regNoValue;
275
276     try {
277
278         selectedRow = recordsTable.getSelectionModel().getSelectedItem();
279         regNoValue = selectedRow.getRegNo();
280
281         Connection connection =
282             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
283                 , "root", "root");
284
285         String sql = "DELETE FROM patient_records_staffs WHERE reg_no = ?";
286
287         PreparedStatement statement = connection.prepareStatement(sql);
288
289         statement.setInt(1, regNoValue);
290         statement.execute();
291
292         new Main().changeScene("StaffPatientRecords.fxml");
293
294     }
295
296     catch (SQLException ae) {
297
298         System.out.println("Error!!! Connection Failed");
299         ae.printStackTrace();
300
301         Alert alert = new Alert(Alert.AlertType.ERROR);
302         alert.setTitle("Error Message");
303         alert.setHeaderText(null);
304         alert.setContentText("ERROR Deleting Selected Row!!");
305         alert.showAndWait();
306
307     }
308
309     catch (NullPointerException ae) {

```

```

309         System.out.println("Error!!! Retriving Reg_No");
310         ae.printStackTrace();
311
312         Alert alert = new Alert(Alert.AlertType.ERROR);
313         alert.setTitle("Error Message");
314         alert.setHeaderText(null);
315         alert.setContentText("ERROR!!! Please Select a Row!!!");
316         alert.showAndWait();
317
318     }
319
320 }
321
322 //MenuBar
323
324 //Menu 1 - File
325
326 //Menu Item 1.1
327 public void HomePageMenuItem(ActionEvent event) {
328     MenuBar obj = new MenuBar();
329     obj.HomePage();
330 }
331
332 //Menu Item 1.2
333 public void LogOutMenuItem(ActionEvent event) {
334     MenuBar obj = new MenuBar();
335     obj.LogOut();
336 }
337
338 //Menu Item 1.3
339 public void ExitMenuItem(ActionEvent event) {
340     MenuBar obj = new MenuBar();
341     obj.Exit();
342 }
343
344 //Menu 2 - View
345
346 //Menu Item 2.1
347 public void NewPatientEntryMenuItem(ActionEvent event) {
348     MenuBar obj = new MenuBar();
349     obj.NewPatientEntry();
350 }
351
352 //Menu Item 2.2
353 public void ShowPatientRecordMenuItem(ActionEvent event) {
354     MenuBar obj = new MenuBar();
355     obj.ShowPatientRecord();
356 }
357
358 //Menu Item 2.3
359 public void MedicineRecordsMenuItem(ActionEvent event) {
360     MenuBar obj = new MenuBar();
361     obj.MedicineRecords();
362 }
363
364 //Menu 3 - Help
365
366 //Menu Item 3.1
367 public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
368     MenuBar obj = new MenuBar();
369     obj.CollegeWebsiteLinks();
370 }
371
372 //Menu Item 3.2

```

```
373     public void AboutMenuItem(ActionEvent event) {
374         MenuBar obj = new MenuBar();
375         obj.About();
376     }
377
378 }
```

```
1  package application;
2
3  import java.io.IOException;
4  import java.sql.Connection;
5  import java.sql.DriverManager;
6  import java.sql.PreparedStatement;
7  import java.sql.ResultSet;
8  import java.sql.SQLException;
9  import java.sql.Statement;
10
11  import javafx.collections.FXCollections;
12  import javafx.collections.ObservableList;
13  import javafx.event.ActionEvent;
14  import javafx.fxml.FXML;
15  import javafx.scene.control.Alert;
16  import javafx.scene.control.Button;
17  import javafx.scene.control.ButtonType;
18  import javafx.scene.control.TableColumn;
19  import javafx.scene.control.TableView;
20  import javafx.scene.control.TextField;
21
22  public class OtherPatientRecordsController {
23
24      //table columns and table
25      @FXML
26      public TableView<OtherPatient> recordsTable;
27      @FXML
28      public TableColumn<OtherPatient, Integer> col_regNo;
29      @FXML
30      public TableColumn<OtherPatient, String> col_name;
31      @FXML
32      public TableColumn<OtherPatient, String> col_sex;
33      @FXML
34      public TableColumn<OtherPatient, Integer> col_age;
35      @FXML
36      public TableColumn<OtherPatient, Integer> col_phoneNo;
37      @FXML
38      public TableColumn<OtherPatient, String> col_date;
39      @FXML
40      public TableColumn<OtherPatient, String> col_time;
41      @FXML
42      public TableColumn<OtherPatient, String> col_complaint;
43      @FXML
44      public TableColumn<OtherPatient, String> col_remarks;
45
46      @FXML
47      private Button backButton;
48
49      @FXML
50      private Button logoutButton;
51
52      @FXML
53      private TextField filterByTextField;
54
55      @FXML
56      private Button searchButton;
57
58      @FXML
59      private Button refreshButton;
60
61      @FXML
62      private Button deleteButton;
63
64      public void backToHomePage(ActionEvent event) throws IOException {
```

```

65
66     Main m = new Main();
67     m.changeScene("SelectPatientRecordsPage.fxml");
68
69 }
70
71 public void logout(ActionEvent event) throws IOException {
72
73     Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
74     alert.setTitle("Logout");
75     alert.setHeaderText(null);
76     alert.setContentText("Are you sure you want to Logout?");
77     if (alert.showAndWait().get() == ButtonType.OK){
78         Main m = new Main();
79         m.changeScene("LoginPage.fxml");
80     }
81 }
82
83 //show filtered data
84 public void filterTable(ActionEvent event) throws IOException {
85
86     col_regNo.setCellValueFactory(cellData -> cellData.getValue().get_RegNo().asObject());
87     col_name.setCellValueFactory(cellData -> cellData.getValue().get_Name());
88     col_sex.setCellValueFactory(cellData -> cellData.getValue().get_Sex());
89     col_age.setCellValueFactory(cellData -> cellData.getValue().get_Age().asObject());
90     col_phoneNo.setCellValueFactory(cellData ->
91     cellData.getValue().get_PhoneNo().asObject());
92     col_date.setCellValueFactory(cellData -> cellData.getValue().get_Date());
93     col_time.setCellValueFactory(cellData -> cellData.getValue().get_Time());
94     col_complaint.setCellValueFactory(cellData -> cellData.getValue().get_Complaint());
95     col_remarks.setCellValueFactory(cellData -> cellData.getValue().get_Remarks());
96
97     ObservableList<OtherPatient> patientList = searchPatient(filterByTextField.getText());
98     populateTable(patientList);
99 }
100
101 //search patient form database
102 public static ObservableList<OtherPatient> searchPatient(String str) {
103
104     ObservableList<OtherPatient> patientList = FXCollections.observableArrayList();
105
106     try {
107
108         Connection connection =
109         DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmary_management_system"
110         , "root", "root");
111         Statement statement = connection.createStatement();
112
113         String sql = "SELECT * FROM patient_records_others WHERE '" + str + "' IN (reg_no,
114         name, sex, age, phone_no, date, time, complaint, remarks);";
115         ResultSet resultset = statement.executeQuery(sql);
116
117         patientList = getPatientObject(resultset);
118     }
119
120     catch (SQLException ae) {
121
122         System.out.println("Error!!! Connection Failed");
123         ae.printStackTrace();
124
125         Alert alert = new Alert(Alert.AlertType.ERROR);
126         alert.setTitle("Error Message");

```

```

125         alert.setHeaderText(null);
126         alert.setContentText("ERROR Searching for '" + str + "' in the Database!!!");
127         alert.showAndWait();
128
129     }
130
131     return patientList;
132
133 }
134
135 @FXML
136 public void initialize() throws Exception {
137
138     filterByTextField.setText(null);
139
140     //get column data
141     col_regNo.setCellValueFactory(cellData -> cellData.getValue().get_RegNo().asObject());
142     col_name.setCellValueFactory(cellData -> cellData.getValue().get_Name());
143     col_sex.setCellValueFactory(cellData -> cellData.getValue().get_Sex());
144     col_age.setCellValueFactory(cellData -> cellData.getValue().get_Age().asObject());
145     col_phoneNo.setCellValueFactory(cellData ->
146         cellData.getValue().get_PhoneNo().asObject());
147     col_date.setCellValueFactory(cellData -> cellData.getValue().get_Date());
148     col_time.setCellValueFactory(cellData -> cellData.getValue().get_Time());
149     col_complaint.setCellValueFactory(cellData -> cellData.getValue().get_Complaint());
150     col_remarks.setCellValueFactory(cellData -> cellData.getValue().get_Remarks());
151
152     ObservableList<OtherPatient> patientList = getPatientRecords();
153     populateTable(patientList);
154
155 }
156
157 //add items to table
158 public void populateTable (ObservableList<OtherPatient> patientList) {
159     recordsTable.setItems(patientList);
160 }
161
162 //get patient records from database
163 public static ObservableList<OtherPatient> getPatientRecords() {
164
165     ObservableList<OtherPatient> patientList = FXCollections.observableArrayList();
166
167     try {
168
169         Connection connection =
170             DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
171                 , "root", "root");
172         Statement statement = connection.createStatement();
173
174         String sql = "SELECT * FROM patient_records_others;";
175         ResultSet resultset = statement.executeQuery(sql);
176
177         patientList = getPatientObject(resultset);
178
179     }
180
181     catch (SQLException ae) {
182
183         System.out.println("Error!!! Connection Failed");
184         ae.printStackTrace();
185
186         Alert alert = new Alert(Alert.AlertType.ERROR);
187         alert.setTitle("Error Message");
188         alert.setHeaderText(null);

```

```

186         alert.setContentText("ERROR Connecting to MySQL Database!!");
187         alert.showAndWait();
188
189     }
190
191     return patientList;
192
193 }
194
195 public static ObservableList<OtherPatient> getPatientObject(ResultSet resultset) {
196
197     ObservableList<OtherPatient> patientRecordList = FXCollections.observableArrayList();
198
199     try {
200
201         int count = 0;
202
203         while (resultset.next()) {
204
205             count++;
206
207             OtherPatient obj = new OtherPatient();
208
209             obj.setRegNo(resultset.getInt("reg_no"));
210             obj.setName(resultset.getString("name"));
211             obj.setSex(resultset.getString("sex"));
212             obj.setAge(resultset.getInt("age"));
213             obj.setPhoneNo(resultset.getInt("phone_no"));
214             obj.setDate(resultset.getString("date"));
215             obj.setTime(resultset.getString("time"));
216             obj.setComplaint(resultset.getString("complaint"));
217             obj.setRemarks(resultset.getString("remarks"));
218
219             patientRecordList.add(obj);
220
221         }
222
223         if (count==0) {
224
225             new Main().changeScene("OtherPatientRecords.fxml");
226
227             Alert alert = new Alert(Alert.AlertType.ERROR);
228             alert.setTitle("Error Message");
229             alert.setHeaderText(null);
230             alert.setContentText("No Data in MySQL Database!!");
231             alert.showAndWait();
232         }
233
234     }
235
236     catch (SQLException ae) {
237
238         System.out.println("Error!!! Connection Failed");
239         ae.printStackTrace();
240
241         Alert alert = new Alert(Alert.AlertType.ERROR);
242         alert.setTitle("Error Message");
243         alert.setHeaderText(null);
244         alert.setContentText("ERROR Retriving Data from MySQL Database!!");
245         alert.showAndWait();
246
247     }
248     return patientRecordList;
249

```



```

250     }
251
252     //delete selected row from database
253     public void deleteRow(ActionEvent event) throws IOException, NullPointerException{
254
255         OtherPatient selectedRow = null;
256         int regNoValue;
257
258         try {
259
260             selectedRow = recordsTable.getSelectionModel().getSelectedItem();
261             regNoValue = selectedRow.getRegNo();
262
263             Connection connection =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/infirmaty_management_system"
                    , "root", "root");
264
265             String sql = "DELETE FROM patient_records_others WHERE reg_no = ?;";
266
267             PreparedStatement statement = connection.prepareStatement(sql);
268
269             statement.setInt(1, regNoValue);
270             statement.execute();
271
272             new Main().changeScene("OtherPatientRecords.fxml");
273
274         }
275
276         catch (SQLException ae) {
277
278             System.out.println("Error!!! Connection Failed");
279             ae.printStackTrace();
280
281             Alert alert = new Alert(Alert.AlertType.ERROR);
282             alert.setTitle("Error Message");
283             alert.setHeaderText(null);
284             alert.setContentText("ERROR Deleting Selected Row!!");
285             alert.showAndWait();
286
287         }
288
289         catch (NullPointerException ae) {
290
291             System.out.println("Error!!! Retriving Reg_No");
292             ae.printStackTrace();
293
294             Alert alert = new Alert(Alert.AlertType.ERROR);
295             alert.setTitle("Error Message");
296             alert.setHeaderText(null);
297             alert.setContentText("ERROR!!! Please Select a Row!!");
298             alert.showAndWait();
299
300         }
301
302     }
303
304     //MenuBar
305
306     //Menu 1 - File
307
308     //Menu Item 1.1
309     public void HomePageMenuItem(ActionEvent event) {
310         MenuBar obj = new MenuBar();
311         obj.HomePage();

```

```
312     }
313
314     //Menu Item 1.2
315     public void LogoutMenuItem(ActionEvent event) {
316         MenuBar obj = new MenuBar();
317         obj.LogOut();
318     }
319
320     //Menu Item 1.3
321     public void ExitMenuItem(ActionEvent event) {
322         MenuBar obj = new MenuBar();
323         obj.Exit();
324     }
325
326     //Menu 2 - View
327
328     //Menu Item 2.1
329     public void NewPatientEntryMenuItem(ActionEvent event) {
330         MenuBar obj = new MenuBar();
331         obj.NewPatientEntry();
332     }
333
334     //Menu Item 2.2
335     public void ShowPatientRecordMenuItem(ActionEvent event) {
336         MenuBar obj = new MenuBar();
337         obj.ShowPatientRecord();
338     }
339
340     //Menu Item 2.3
341     public void MedicineRecordsMenuItem(ActionEvent event) {
342         MenuBar obj = new MenuBar();
343         obj.MedicineRecords();
344     }
345
346     //Menu 3 - Help
347
348     //Menu Item 3.1
349     public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
350         MenuBar obj = new MenuBar();
351         obj.CollegeWebsiteLinks();
352     }
353
354     //Menu Item 3.2
355     public void AboutMenuItem(ActionEvent event) {
356         MenuBar obj = new MenuBar();
357         obj.About();
358     }
359
360 }
```

```

1  <?xml version "1.0" encoding "UTF-8"?>
2
3  <?import javafx.scene.control.Button?>
4  <?import javafx.scene.control.Label?>
5  <?import javafx.scene.control.PasswordField?>
6  <?import javafx.scene.control.TextField?>
7  <?import javafx.scene.image.Image?>
8  <?import javafx.scene.image.ImageView?>
9  <?import javafx.scene.layout.AnchorPane?>
10 <?import javafx.scene.layout.BorderPane?>
11 <?import javafx.scene.text.Font?>
12
13 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity"
14     minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0"
15     prefWidth="800.0" style="-fx-background-color: white;"
16     xmlns="http://javafx.com/javafx/15.0.1"
17     xmlns:fx="http://javafx.com/fxml/1"
18     fx:controller="application.LoginPageController">
19     <left>
20         <AnchorPane prefHeight="600.0" prefWidth="400.0"
21             BorderPane.alignment="CENTER">
22             <children>
23                 <ImageView fitHeight="200.0" fitWidth="200.0"
24                     layoutX="100.0" layoutY="122.0">
25                     <image>
26                         <Image url="@../Images/CollegeLogo200px.png" />
27                     </image>
28                 </ImageView>
29                 <Label layoutX="37.0" layoutY="347.0"
30                     text="College of Science and Technology">
31                     <font>
32                         <Font name="Roboto Condensed Bold" size="24.0" />
33                     </font>
34                 </Label>
35                 <Label layoutX="92.0" layoutY="396.0"
36                     text="Infirmary Management System">
37                     <font>
38                         <Font name="Roboto Condensed Bold" size="18.0" />
39                     </font>
40                 </Label>
41             </children>
42         </AnchorPane>
43     </left>
44     <right>
45         <AnchorPane prefHeight="600.0" prefWidth="400.0"
46             BorderPane.alignment="CENTER">
47             <children>
48                 <ImageView fitHeight="571.0" fitWidth="415.0"
49                     layoutX="32.0" layoutY="14.0" pickOnBounds="true"
50                     preserveRatio="true">
51                     <image>
52                         <Image url="@../Images/LoginRightPanel.png" />
53                     </image>
54                 </ImageView>
55                 <Label layoutX="139.0" layoutY="188.0" prefHeight="34.0"
56                     prefWidth="127.0" text="Admin Login">
57                     <font>
58                         <Font name="Roboto Condensed Bold" size="25.0" />
59                     </font>
60                 </Label>
61                 <Label layoutX="94.0" layoutY="289.0" text="Username">
62                     <font>
63                         <Font name="Roboto Condensed Bold" size="16.0" />
64                     </font>

```

```

65         </Label>
66         <TextField fx:id="usernameTextField" layoutX="90.0"
67             layoutY="325.0" prefHeight="25.0" prefWidth="220.0"
68             promptText="Enter Username">
69             <font>
70                 <Font size="15.0" />
71             </font>
72         </TextField>
73         <Label layoutX="95.0" layoutY="382.0" text="Password">
74             <font>
75                 <Font name="Roboto Condensed Bold" size="16.0" />
76             </font>
77         </Label>
78         <PasswordField fx:id="passwordField" layoutX="90.0"
79             layoutY="418.0" prefHeight="25.0" prefWidth="220.0"
80             promptText="Enter Password">
81             <font>
82                 <Font size="15.0" />
83             </font>
84         </PasswordField>
85         <Button fx:id="loginButton" layoutX="239.0" layoutY="489.0"
86             mnemonicParsing="false" onAction="#userLogin" prefHeight="32.0"
87             prefWidth="70.0" text="Login">
88             <font>
89                 <Font size="15.0" />
90             </font>
91         </Button>
92     </children>
93 </AnchorPane>
94 </right>
95 </BorderPane>
96

```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.scene.control.Button?>
6  <?import javafx.scene.control.Label?>
7  <?import javafx.scene.control.Menu?>
8  <?import javafx.scene.control.MenuBar?>
9  <?import javafx.scene.control.MenuItem?>
10 <?import javafx.scene.control.Separator?>
11 <?import javafx.scene.effect.ImageInput?>
12 <?import javafx.scene.image.Image?>
13 <?import javafx.scene.image.ImageView?>
14 <?import javafx.scene.layout.AnchorPane?>
15 <?import javafx.scene.layout.BorderPane?>
16 <?import javafx.scene.text.Font?>
17
18 <BorderPane focusTraversable="true" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0"
style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.HomePageController">
19     <top>
20         <AnchorPane prefHeight="183.0" prefWidth="800.0" BorderPane.alignment="CENTER">
21             <children>
22                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
23                     <menus>
24                         <Menu mnemonicParsing="false" text="File">
25                             <items>
26                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
27                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
28                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
29                             </items>
30                         </Menu>
31                         <Menu mnemonicParsing="false" text="View">
32                             <items>
33                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
34                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
36                             </items>
37                         </Menu>
38                         <Menu mnemonicParsing="false" text="Help">
39                             <items>
40                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
41                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
42                             </items>
43                         </Menu>
44                     </menus>
45                 </MenuBar>
46                 <ImageView fitHeight="250.0" fitWidth="800.0" layoutY="26.0"
pickOnBounds="true" preserveRatio="true">
47                     <image>
48                         <Image url="@../Images/HomePageHeader.png" />
49                     </image>

```

```

50         </ImageView>
51     <Separator layoutX="158.0" layoutY="188.0" prefWidth "485.0" />
52 </children>
53 </AnchorPane>
54 </top>
55 <center>
56     <AnchorPane prefHeight="350.0" prefWidth="800.0" BorderPane.alignment="CENTER">
57         <children>
58             <Button fx:id="newPatientButton" layoutX="65.0" layoutY="37.0"
59                 mnemonicParsing="false" onAction="#patientDetailsEntry" prefHeight="180.0"
60                 prefWidth="180.0" text="New Patient">
61                 <effect>
62                     <ImageInput>
63                         <source>
64                             <Image
65                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
66                                 tem/Images/NewPatient.png" />
67                         </source>
68                     </ImageInput>
69                 </effect>
70             </Button>
71             <Label layoutX="107.0" layoutY="245.0" text="New Patient">
72                 <font>
73                     <Font name="Roboto Condensed Bold" size "20.0" />
74                 </font>
75             </Label>
76             <Button fx:id="patientRecordsButton" layoutX="310.0" layoutY "37.0"
77                 mnemonicParsing="false" onAction="#patientRecords" prefHeight="180.0"
78                 prefWidth "180.0" text "Patient Records">
79                 <effect>
80                     <ImageInput>
81                         <source>
82                             <Image
83                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
84                                 tem/Images/PatientDetails.png" />
85                         </source>
86                     </ImageInput>
87                 </effect>
88             </Button>
89             <Label layoutX "337.0" layoutY="245.0" text="Patient Records">
90                 <font>
91                     <Font name="Roboto Condensed Bold" size "20.0" />
92                 </font>
93             </Label>
94             <Button fx:id="medicineStockButton" layoutX="555.0" layoutY "37.0"
95                 mnemonicParsing="false" onAction="#medicineStock" prefHeight "180.0"
96                 prefWidth "180.0" text "Export Prescription">
97                 <effect>
98                     <ImageInput>
99                         <source>
100                             <Image
101                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
102                                 tem/Images/Medicine.png" />
103                         </source>
104                     </ImageInput>
105                 </effect>
106             </Button>
107             <Label layoutX="585.0" layoutY="245.0" text="Medicine Stock">
108                 <font>
109                     <Font name="Roboto Condensed Bold" size "20.0" />
110                 </font>
111             </Label>
112             <Button fx:id="logoutButton" layoutX "684.0" layoutY="317.0"
113                 mnemonicParsing="false" onAction="#userLogout" prefHeight "26.0"

```

```
101         prefWidth="74.0" text="Logout">
102             <font>
103                 <Font name="Roboto Condensed Bold" size="18.0" />
104             </font>
105         </Button>
106     </children>
107 </AnchorPane>
108 </center>
109 </BorderPane>
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.scene.control.Button?>
6  <?import javafx.scene.control.Label?>
7  <?import javafx.scene.control.Menu?>
8  <?import javafx.scene.control.MenuBar?>
9  <?import javafx.scene.control.MenuItem?>
10 <?import javafx.scene.control.Separator?>
11 <?import javafx.scene.effect.ImageInput?>
12 <?import javafx.scene.image.Image?>
13 <?import javafx.scene.image.ImageView?>
14 <?import javafx.scene.layout.AnchorPane?>
15 <?import javafx.scene.layout.BorderPane?>
16 <?import javafx.scene.text.Font?>
17
18 <BorderPane focusTraversable="true" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0"
style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.SelectPatientEntryPageController">
19     <top>
20         <AnchorPane prefHeight="183.0" prefWidth="800.0" BorderPane.alignment="CENTER">
21             <children>
22                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
23                     <menus>
24                         <Menu mnemonicParsing="false" text="File">
25                             <items>
26                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
27                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
28                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
29                             </items>
30                         </Menu>
31                         <Menu mnemonicParsing="false" text="View">
32                             <items>
33                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
34                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
36                             </items>
37                         </Menu>
38                         <Menu mnemonicParsing="false" text="Help">
39                             <items>
40                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
41                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
42                             </items>
43                         </Menu>
44                     </menus>
45                 </MenuBar>
46                 <ImageView fitHeight="250.0" fitWidth="800.0" layoutY="26.0"
pickOnBounds="true" preserveRatio="true">
47                     <image>
48                         <Image url="@../Images/HomePageHeader.png" />
49                     </image>

```



```

50         </ImageView>
51     <Separator layoutX="158.0" layoutY="188.0" prefHeight="8.0" prefWidth="485.0" />
52     </children>
53 </AnchorPane>
54 </top>
55 <center>
56     <AnchorPane prefHeight="350.0" prefWidth="800.0" BorderPane.alignment="CENTER">
57         <children>
58             <Button fx:id="StudentsButton" layoutX="65.0" layoutY="78.0"
59                 mnemonicParsing="false" onAction="#studentDetailsEntry" prefHeight="180.0"
60                 prefWidth="180.0" text="New Patient">
61                 <effect>
62                     <ImageInput>
63                         <source>
64                             <Image
65                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
66                                 tem/Images/StudentsIcon.png" />
67                         </source>
68                     </ImageInput>
69                 </effect>
70             </Button>
71             <Label layoutX="119.0" layoutY="274.0" text="Students">
72                 <font>
73                     <Font name="Roboto Condensed Bold" size="20.0" />
74                 </font>
75             </Label>
76             <Button fx:id="StaffButton" layoutX="310.0" layoutY="78.0"
77                 mnemonicParsing="false" onAction="#staffDetailsEntry" prefHeight="180.0"
78                 prefWidth="180.0" text="Patient Records">
79                 <effect>
80                     <ImageInput>
81                         <source>
82                             <Image
83                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
84                                 tem/Images/Staff%20Icon.png" />
85                         </source>
86                     </ImageInput>
87                 </effect>
88             </Button>
89             <Label layoutX="380.0" layoutY="272.0" text="Staff">
90                 <font>
91                     <Font name="Roboto Condensed Bold" size="20.0" />
92                 </font>
93             </Label>
94             <Button fx:id="OthersButton" layoutX="555.0" layoutY="78.0"
95                 mnemonicParsing="false" onAction="#OthersDetailsEntry" prefHeight="180.0"
96                 prefWidth="180.0" text="Export Prescription">
97                 <effect>
98                     <ImageInput>
99                         <source>
100                             <Image
                                url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
                                tem/Images/OthersIcon.png" />
                                </source>
                                </ImageInput>
                                </effect>
                                </Button>
                                <Label layoutX="619.0" layoutY="272.0" text="Others">
                                    <font>
                                        <Font name="Roboto Condensed Bold" size="20.0" />
                                    </font>
                                </Label>
                                <Button fx:id="BackButton" layoutX="687.0" layoutY="334.0"
                                    mnemonicParsing="false" onAction="#userBack" prefHeight="26.0" prefWidth="74.0"

```

```
101         text="Back">
102             <font>
103                 <Font name="Roboto Condensed Bold" size="18.0" />
104             </font>
105         </Button>
106     <Label layoutX="198.0" layoutY="-11.0" prefHeight="49.0" prefWidth="364.0"
107     text="PATIENT DETAILS ENTRY">
108         <font>
109             <Font name="Calibri Bold" size="34.0" />
110         </font>
111     </Label>
112 </children>
113 </AnchorPane>
114 </center>
115 </BorderPane>
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.scene.control.Button?>
6  <?import javafx.scene.control.Label?>
7  <?import javafx.scene.control.Menu?>
8  <?import javafx.scene.control.MenuBar?>
9  <?import javafx.scene.control.MenuItem?>
10 <?import javafx.scene.control.Separator?>
11 <?import javafx.scene.effect.ImageInput?>
12 <?import javafx.scene.image.Image?>
13 <?import javafx.scene.image.ImageView?>
14 <?import javafx.scene.layout.AnchorPane?>
15 <?import javafx.scene.layout.BorderPane?>
16 <?import javafx.scene.text.Font?>
17
18 <BorderPane focusTraversable="true" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0"
style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.SelectPatientRecordsPageController">
19     <top>
20         <AnchorPane prefHeight="183.0" prefWidth="800.0" BorderPane.alignment="CENTER">
21             <children>
22                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
23                     <menus>
24                         <Menu mnemonicParsing="false" text="File">
25                             <items>
26                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
27                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
28                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
29                             </items>
30                         </Menu>
31                         <Menu mnemonicParsing="false" text="View">
32                             <items>
33                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
34                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
36                             </items>
37                         </Menu>
38                         <Menu mnemonicParsing="false" text="Help">
39                             <items>
40                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
41                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
42                             </items>
43                         </Menu>
44                     </menus>
45                 </MenuBar>
46                 <ImageView fitHeight="250.0" fitWidth="800.0" layoutY="26.0"
pickOnBounds="true" preserveRatio="true">
47                     <image>
48                         <Image url="@../Images/HomePageHeader.png" />

```

```

49         </image>
50     </ImageView>
51     <Separator layoutX="158.0" layoutY="188.0" prefWidth="485.0" />
52 </children>
53 </AnchorPane>
54 </top>
55 <center>
56     <AnchorPane prefHeight="350.0" prefWidth="800.0" BorderPane.alignment="CENTER">
57         <children>
58             <Button fx:id="StudentsButton" layoutX="65.0" layoutY="78.0"
59                 mnemonicParsing="false" onAction="#studentRecords" prefHeight="180.0"
60                 prefWidth="180.0" text="New Patient">
61                 <effect>
62                     <ImageInput>
63                         <source>
64                             <Image
65                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
66                                 tem/Images/StudentsIcon.png" />
67                         </source>
68                     </ImageInput>
69                 </effect>
70             </Button>
71             <Label layoutX="119.0" layoutY="274.0" text="Students">
72                 <font>
73                     <Font name="Roboto Condensed Bold" size="20.0" />
74                 </font>
75             </Label>
76             <Button fx:id="StaffButton" layoutX="310.0" layoutY="78.0"
77                 mnemonicParsing="false" onAction="#staffRecords" prefHeight="180.0"
78                 prefWidth="180.0" text="Patient Records">
79                 <effect>
80                     <ImageInput>
81                         <source>
82                             <Image
83                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
84                                 tem/Images/Staff%20Icon.png" />
85                         </source>
86                     </ImageInput>
87                 </effect>
88             </Button>
89             <Label layoutX="380.0" layoutY="272.0" text="Staff">
90                 <font>
91                     <Font name="Roboto Condensed Bold" size="20.0" />
92                 </font>
93             </Label>
94             <Button fx:id="OthersButton" layoutX="555.0" layoutY="78.0"
95                 mnemonicParsing="false" onAction="#OthersRecords" prefHeight="180.0"
96                 prefWidth="180.0" text="Export Prescription">
97                 <effect>
98                     <ImageInput>
99                         <source>
100                             <Image
101                                 url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagementSys
102                                 tem/Images/OthersIcon.png" />
103                         </source>
104                     </ImageInput>
105                 </effect>
106             </Button>
107             <Label layoutX="619.0" layoutY="272.0" text="Others">
108                 <font>
109                     <Font name="Roboto Condensed Bold" size="20.0" />
110                 </font>
111             </Label>
112             <Button fx:id="BackButton" layoutX="687.0" layoutY="334.0"

```

```
mnemonicParsing="false" onAction="#userBack" prefHeight="26.0" prefWidth="74.0"
text="Back">
101     <font>
102         <Font name="Roboto Condensed Bold" size="18.0" />
103     </font>
104 </Button>
105 <Label layoutX="269.0" layoutY="-6.0" text="PATIENT RECORDS">
106     <font>
107         <Font name="Calibri Bold" size="34.0" />
108     </font>
109 </Label>
110 </children>
111 </AnchorPane>
112 </center>
113 </BorderPane>
114
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.DatePicker?>
6  <?import javafx.scene.control.Label?>
7  <?import javafx.scene.control.Menu?>
8  <?import javafx.scene.control.MenuBar?>
9  <?import javafx.scene.control.MenuItem?>
10 <?import javafx.scene.control.ScrollPane?>
11 <?import javafx.scene.control.TableColumn?>
12 <?import javafx.scene.control.TableView?>
13 <?import javafx.scene.control.TextField?>
14 <?import javafx.scene.layout.AnchorPane?>
15 <?import javafx.scene.layout.BorderPane?>
16 <?import javafx.scene.text.Font?>
17
18 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.MedicineRecordsController">
19     <top>
20         <AnchorPane prefHeight="82.0" prefWidth="800.0" BorderPane.alignment="CENTER">
21             <children>
22                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
23                     <menus>
24                         <Menu mnemonicParsing="false" text="File">
25                             <items>
26                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
27                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
28                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
29                             </items>
30                         </Menu>
31                         <Menu mnemonicParsing="false" text="View">
32                             <items>
33                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
34                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
36                             </items>
37                         </Menu>
38                         <Menu mnemonicParsing="false" text="Help">
39                             <items>
40                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
41                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
42                             </items>
43                         </Menu>
44                     </menus>
45                 </MenuBar>
46                 <Label alignment="CENTER" layoutY="23.0" prefHeight="63.0" prefWidth="800.0"
style="-fx-background-color: #C5C6C7;" text="Medicine Records"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="23.0">
47                     <font>

```

```

48         <Font name="Roboto Condensed Bold" size="52.0" />
49     </font>
50     <padding>
51         <Insets left="40.0" />
52     </padding>
53 </Label>
54 </children>
55 </AnchorPane>
56 </top>
57 <center>
58     <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">
59         <children>
60             <ScrollPane prefHeight="950.0" prefWidth="800.0" AnchorPane.bottomAnchor="0.0"
61                 AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
62                 AnchorPane.topAnchor="0.0">
63                 <content>
64                     <AnchorPane minHeight="950.0" minWidth="800.0" prefHeight="450.0"
65                         prefWidth="800.0">
66                         <children>
67                             <Label layoutX="75.0" layoutY="85.0" prefHeight="20.0"
68                                 prefWidth="97.0" text="Medicine Name">
69                                     <font>
70                                         <Font name="Roboto Condensed Regular" size="15.0" />
71                                     </font>
72                                 </Label>
73                                 <TextField fx:id="medicineNameTextField" layoutX="173.0"
74                                     layoutY="83.0" prefHeight="25.0" prefWidth="175.0" />
75                                 <Label layoutX="75.0" layoutY="164.0" prefHeight="20.0"
76                                     prefWidth="97.0" text="Expiry Date">
77                                     <font>
78                                         <Font name="Roboto Condensed Regular" size="15.0" />
79                                     </font>
80                                 </Label>
81                                 <DatePicker fx:id="medicineExpiryDateSelector" layoutX="173.0"
82                                     layoutY="162.0" onAction="#showMedicineDate" />
83                                 <Label layoutX="75.0" layoutY="123.0" prefHeight="20.0"
84                                     prefWidth="97.0" text="Quantity">
85                                     <font>
86                                         <Font name="Roboto Condensed Regular" size="15.0" />
87                                     </font>
88                                 </Label>
89                                 <TextField fx:id="medicineQuantityTextField" layoutX="173.0"
90                                     layoutY="121.0" prefHeight="25.0" prefWidth="175.0"
91                                     promptText="Eg. 50 Tablets" />
92                                 <Button fx:id="medicineSaveButton" layoutX="292.0"
93                                     layoutY="209.0" mnemonicParsing="false"
94                                     onAction="#saveMedicine" prefHeight="31.0" prefWidth="55.0"
95                                     text="Save">
96                                     <font>
97                                         <Font name="Roboto Condensed Regular" size="15.0" />
98                                     </font>
99                                 </Button>
100                                <Label layoutX="451.0" layoutY="85.0" prefHeight="20.0"
101                                    prefWidth="97.0" text="Injection Name">
102                                    <font>
103                                        <Font name="Roboto Condensed Regular" size="15.0" />
104                                    </font>
105                                </Label>
106                                <TextField fx:id="injectionNameTextField" layoutX="549.0"
107                                    layoutY="83.0" prefHeight="25.0" prefWidth="175.0" />
108                                <Label layoutX="451.0" layoutY="164.0" prefHeight="20.0"
109                                    prefWidth="97.0" text="Expiry Date">
110                                    <font>
111                                        <Font name="Roboto Condensed Regular" size="15.0" />
112                                    </font>
113                                </Label>
114                                <TextField fx:id="injectionExpiryDateTextField" layoutX="549.0"
115                                    layoutY="162.0" prefHeight="25.0" prefWidth="175.0" />
116                                <Button fx:id="injectionSaveButton" layoutX="717.0"
117                                    layoutY="209.0" mnemonicParsing="false"
118                                    onAction="#saveInjection" prefHeight="31.0" prefWidth="55.0"
119                                    text="Save">
120                                    <font>
121                                        <Font name="Roboto Condensed Regular" size="15.0" />
122                                    </font>
123                                </Button>
124                            </children>
125                        </AnchorPane>
126                    </content>
127                </ScrollPane>
128            </children>
129        </AnchorPane>
130    </center>
131 </top>
132 </ScrollPane>
133 </content>
134 </AnchorPane>
135 </children>
136 </BorderPane>
137 </Window>
138 </fx:root>

```

```

96         </font>
97     </Label>
98     <DatePicker fx:id="injectionExpiryDateSelector" layoutX="549.0"
99     layoutY="162.0" onAction="#showInjectionDate" />
100     <Label layoutX="451.0" layoutY="123.0" prefHeight="20.0"
101     prefWidth="97.0" text="Quantity">
102         <font>
103             <Font name="Roboto Condensed Regular" size="15.0" />
104         </font>
105     </Label>
106     <TextField fx:id="injectionQuantityTextField" layoutX="549.0"
107     layoutY="121.0" prefHeight="25.0" prefWidth="175.0"
108     promptText="Eg. 20ml" />
109     <Button fx:id="injectionSaveButton" layoutX="667.0"
110     layoutY="209.0" mnemonicParsing="false"
111     onAction="#saveInjection" prefHeight="31.0" prefWidth="55.0"
112     text="Save">
113         <font>
114             <Font name="Roboto Condensed Regular" size="15.0" />
115         </font>
116     </Button>
117     <TableView fx:id="medicineTable" layoutX="56.0" layoutY="324.0"
118     prefHeight="500.0" prefWidth="290.0">
119         <columns>
120             <TableColumn fx:id="col_medicineName" prefWidth="120.0"
121             text="Medicine Name" />
122             <TableColumn fx:id="col_medicineQuantity"
123             prefWidth="60.0" text="Quantity" />
124             <TableColumn fx:id="col_medicineExpiryDate"
125             prefWidth="110.0" text="Expiry Date" />
126         </columns>
127     </TableView>
128     <TableView fx:id="injectionTable" layoutX="439.0"
129     layoutY="324.0" prefHeight="500.0" prefWidth="290.0">
130         <columns>
131             <TableColumn fx:id="col_injectionName"
132             prefWidth="120.0" text="Injection Name" />
133             <TableColumn fx:id="col_injectionQuantity"
134             prefWidth="60.0" text="Quantity" />
135             <TableColumn fx:id="col_injectionExpiryDate"
136             prefWidth="110.0" text="Expiry Date" />
137         </columns>
138     </TableView>
139     <Button fx:id="backButton" layoutX="75.0" layoutY="14.0"
140     mnemonicParsing="false" onAction="#backToHomePage"
141     prefHeight="33.0" prefWidth="68.0" text="Back">
142         <font>
143             <Font name="Roboto Condensed Regular" size="15.0" />
144         </font>
145     </Button>
146     <Button fx:id="medicineUpdateStockButton" layoutX="173.0"
147     layoutY="209.0" mnemonicParsing="false"
148     onAction="#updateMedicine" prefHeight="31.0" prefWidth="109.0"
149     text="Update Stock">
150         <font>
151             <Font name="Roboto Condensed Regular" size="15.0" />
152         </font>
153     </Button>
154     <Button fx:id="injectionUpdateStockButton" layoutX="549.0"
155     layoutY="209.0" mnemonicParsing="false"
156     onAction="#updateInjection" prefHeight="31.0" prefWidth="109.0"
157     text="Update Stock">
158         <font>
159             <Font name="Roboto Condensed Regular" size="15.0" />
160         </font>

```



```
137         </font>
138     </Button>
139     <Button fx:id="deleteMedicine" layoutX="234.0" layoutY="287.0"
mnemonicParsing="false" onAction="#deleteMedicine" prefHeight="31.0"
prefWidth="115.0" text="Delete Medicine">
140         <font>
141             <Font name="Roboto Condensed Regular" size="15.0" />
142         </font>
143     </Button>
144     <Button fx:id="deleteInjection" layoutX="613.0" layoutY="287.0"
mnemonicParsing="false" onAction="#deleteInjection" prefHeight="31.0"
prefWidth="115.0" text="Delete Injection">
145         <font>
146             <Font name="Roboto Condensed Regular" size="15.0" />
147         </font>
148     </Button>
149     </children>
150 </AnchorPane>
151 </content>
152 </ScrollPane>
153 </children>
154 </AnchorPane>
155 </center>
156 </BorderPane>
157
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.geometry.Insets?>
6  <?import javafx.scene.control.Button?>
7  <?import javafx.scene.control.CheckBox?>
8  <?import javafx.scene.control.ComboBox?>
9  <?import javafx.scene.control.DatePicker?>
10 <?import javafx.scene.control.Label?>
11 <?import javafx.scene.control.Menu?>
12 <?import javafx.scene.control.MenuBar?>
13 <?import javafx.scene.control.MenuItem?>
14 <?import javafx.scene.control.RadioButton?>
15 <?import javafx.scene.control.ScrollPane?>
16 <?import javafx.scene.control.TextArea?>
17 <?import javafx.scene.control.TextField?>
18 <?import javafx.scene.control.ToggleGroup?>
19 <?import javafx.scene.layout.AnchorPane?>
20 <?import javafx.scene.layout.BorderPane?>
21 <?import javafx.scene.text.Font?>
22
23 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.PatientDetailsEntryController">
24     <top>
25         <AnchorPane prefHeight="84.0" prefWidth="800.0" BorderPane.alignment="CENTER">
26             <children>
27                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
28                     <menus>
29                         <Menu mnemonicParsing="false" text="File">
30                             <items>
31                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
32                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
33                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
34                             </items>
35                         </Menu>
36                         <Menu mnemonicParsing="false" text="View">
37                             <items>
38                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
39                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
40                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
41                             </items>
42                         </Menu>
43                         <Menu mnemonicParsing="false" text="Help">
44                             <items>
45                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
46                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
47                             </items>
48                         </Menu>
49                     </menus>
50                 </MenuBar>

```

```

51         <Label alignment="CENTER" layoutY "26.0" prefHeight "58.0" prefWidth="800.0"
           style="-fx-background-color: #C5C6C7;" text="New Patient Entry - Student"
           textAlignment="CENTER" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
           AnchorPane.topAnchor="26.0">
52             <font>
53                 <Font name="Roboto Condensed Bold" size="52.0" />
54             </font>
55             <padding>
56                 <Insets left="40.0" />
57             </padding>
58         </Label>
59     </children>
60 </AnchorPane>
61 </top>
62 <center>
63     <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment "CENTER">
64         <children>
65             <ScrollPane prefHeight "717.0" prefWidth="800.0" AnchorPane.bottomAnchor "-1.0"
              AnchorPane.leftAnchor "0.0" AnchorPane.rightAnchor "0.0"
              AnchorPane.topAnchor "0.0">
66                 <content>
67                     <AnchorPane prefHeight="715.0" prefWidth "799.0">
68                         <children>
69                             <TextArea fx:id="complaintTextArea" layoutX "31.0"
                               layoutY "266.0" prefHeight="120.0" prefWidth "450.0" />
70                             <Label layoutX="32.0" layoutY "25.0" prefHeight "16.0"
                               prefWidth "60.0" text "Name">
71                                 <font>
72                                     <Font name "Cambria" size "15.0" />
73                                 </font>
74                             </Label>
75                             <TextField fx:id="nameTextField" layoutX="32.0" layoutY "50.0"
                               prefHeight="26.0" prefWidth="226.0" />
76                             <Label layoutX="306.0" layoutY "24.0" prefHeight "20.0"
                               prefWidth "118.0" text "Student Number ">
77                                 <font>
78                                     <Font name "Cambria" size "15.0" />
79                                 </font>
80                             </Label>
81                             <TextField fx:id="studentNoTextField" layoutX "306.0"
                               layoutY "49.0" prefHeight="26.0" prefWidth "172.0"
                               promptText "Eg. 02200174" />
82                             <Label layoutX="532.0" layoutY "26.0" prefHeight "16.0"
                               prefWidth "60.0" text "Reg No.">
83                                 <font>
84                                     <Font name "Cambria" size "15.0" />
85                                 </font>
86                             </Label>
87                             <TextField fx:id="regNoTextField" layoutX="532.0"
                               layoutY "51.0" prefHeight="26.0" prefWidth "176.0" />
88                             <Label layoutX="533.0" layoutY "95.0" prefHeight "16.0"
                               prefWidth "60.0" text "Year">
89                                 <font>
90                                     <Font name "Cambria" size "15.0" />
91                                 </font>
92                             </Label>
93                             <ComboBox fx:id="yearComboBox" layoutX="533.0" layoutY "120.0"
                               onAction "#comboChanged" prefHeight="26.0" prefWidth="63.0"
                               promptText "- " />
94                             <!-- <items> -->
95                             <!-- <FXCollections fx:factory="observableArrayList"> -->
96                             <!-- <String fx:value="1" /> -->
97                             <!-- <String fx:value="2" /> -->
98                             <!-- <String fx:value="3" /> -->

```

```

99      <!-- <String fx:value="4" /> -->
100     <!-- <String fx:value="5" /> -->
101     <!-- </FXCollections> -->
102     <!-- </items> -->
103     <!-- </ComboBox> -->
104     <Label layoutX="610.0" layoutY="95.0" prefHeight="16.0"
prefWidth="60.0" text="Course">
105         <font>
106             <Font name="Cambria" size="15.0" />
107         </font>
108     </Label>
109     <ComboBox fx:id="courseComboBox" layoutX="610.0"
layoutY="120.0" onAction="#comboChanged" prefHeight="26.0"
prefWidth="123.0" promptText="-" />
110     <!-- <items> -->
111     <!-- <FXCollections fx:factory="observableArrayList"> -->
112     <!-- <String fx:value="ME in RE" /> -->
113     <!-- <String fx:value="ME in CM" /> -->
114     <!-- <String fx:value="BE CE" /> -->
115     <!-- <String fx:value="BE EE" /> -->
116     <!-- <String fx:value="B Arch." /> -->
117     <!-- <String fx:value="BE ECE" /> -->
118     <!-- <String fx:value="BE IT" /> -->
119     <!-- <String fx:value="BE ICE" /> -->
120     <!-- <String fx:value="BE EG" /> -->
121     <!-- </FXCollections> -->
122     <!-- </items> -->
123     <!-- </ComboBox> -->
124     <Label layoutX="302.0" layoutY="93.0" prefHeight="20.0"
prefWidth="82.0" text="Department">
125         <font>
126             <Font name="Cambria" size="15.0" />
127         </font>
128     </Label>
129     <ComboBox fx:id="departmentComboBox" layoutX="302.0"
layoutY="118.0" onAction="#deptComboBoxSelected"
prefHeight="26.0" prefWidth="176.0" promptText="-" />
130     <!-- <items> -->
131     <!-- <FXCollections fx:factory="observableArrayList"> -->
132     <!-- <String fx:value="Civil Department" /> -->
133     <!-- <String fx:value="Architecture Department" /> -->
134     <!-- <String fx:value="Electrical Department" /> -->
135     <!-- <String fx:value="IT Department" /> -->
136     <!-- <String fx:value="ECE Department" /> -->
137     <!-- <String fx:value="Electrical Department" /> -->
138     <!-- </FXCollections> -->
139     <!-- </items> -->
140     <!-- </ComboBox> -->
141     <Label layoutX="31.0" layoutY="95.0" prefHeight="16.0"
prefWidth="60.0" text="Sex">
142         <font>
143             <Font name="Cambria" size="15.0" />
144         </font>
145     </Label>
146     <RadioButton fx:id="maleRadioButton" layoutX="31.0"
layoutY="124.0" mnemonicParsing="false"
onAction="#sexRadioSelect" text="Male">
147         <toggleGroup>
148             <ToggleGroup fx:id="gender" />
149         </toggleGroup>
150     </RadioButton>
151     <RadioButton fx:id="femaleRadioButton" layoutX="90.0"
layoutY="124.0" mnemonicParsing="false"
onAction="#sexRadioSelect" text="Female" toggleGroup="$gender" />

```

```

152         <Label layoutX="181.0" layoutY="95.0" prefHeight="16.0"
153             prefWidth="60.0" text="Age">
154             <font>
155                 <Font name="Cambria" size="15.0" />
156             </font>
157         </Label>
158         <TextField fx:id="ageTextField" layoutX="181.0" layoutY="120.0"
159             prefHeight="26.0" prefWidth="75.0" />
160         <Label layoutX="303.0" layoutY="169.0" prefHeight="16.0"
161             prefWidth="60.0" text="Date">
162             <font>
163                 <Font name="Cambria" size="15.0" />
164             </font>
165         </Label>
166         <DatePicker fx:id="DateSelector" layoutX="303.0"
167             layoutY="194.0" prefHeight="26.0" prefWidth="175.0" />
168         <Label layoutX="535.0" layoutY="169.0" prefHeight="16.0"
169             prefWidth="60.0" text="Time">
170             <font>
171                 <Font name="Cambria" size="15.0" />
172             </font>
173         </Label>
174         <TextField fx:id="timeTextField" layoutX="535.0"
175             layoutY="194.0" prefHeight="26.0" prefWidth="81.0"
176             promptText="Eg 12:00" />
177         <RadioButton fx:id="amRadioButton" layoutX="622.0"
178             layoutY="198.0" mnemonicParsing="false"
179             onAction="#timeRadioSelect" text="AM">
180             <toggleGroup>
181                 <ToggleGroup fx:id="timeSelect" />
182             </toggleGroup>
183         </RadioButton>
184         <RadioButton fx:id="pmRadioButton" layoutX="670.0"
185             layoutY="198.0" mnemonicParsing="false"
186             onAction="#timeRadioSelect" text="PM" toggleGroup="$timeSelect"
187             />
188         <Label layoutX="31.0" layoutY="239.0" prefHeight="20.0"
189             prefWidth="159.0" text="Complaint / Treatment ">
190             <font>
191                 <Font name="Cambria" size="15.0" />
192             </font>
193         </Label>
194         <Label layoutX="518.0" layoutY="239.0" prefHeight="20.0"
195             prefWidth="159.0" text="Remarks">
196             <font>
197                 <Font name="Cambria" size="15.0" />
198             </font>
199         </Label>
200         <TextArea fx:id="remarksTextArea" layoutX="519.0"
201             layoutY="266.0" prefHeight="120.0" prefWidth="232.0" />
202         <Label layoutX="31.0" layoutY="403.0" prefHeight="20.0"
203             prefWidth="97.0" text="Leave Options ">
204             <font>
205                 <Font name="Cambria" size="15.0" />
206             </font>
207         </Label>
208         <ComboBox fx:id="leaveOptionsComboBox" layoutX="31.0"
209             layoutY="429.0" onAction="#comboChanged" prefHeight="26.0"
210             prefWidth="190.0" promptText="- - " />
211         <Label layoutX="251.0" layoutY="403.0" prefHeight="20.0"
212             prefWidth="82.0" text="No. of Days ">
213             <font>
214                 <Font name="Cambria" size="15.0" />
215             </font>

```

```

197         </Label>
198         <TextField fx:id="leaveDaysTextField" layoutX "251.0"
199         layoutY "429.0" prefHeight="26.0" prefWidth "81.0" />
200         <Label layoutX="32.0" layoutY="169.0" prefHeight="20.0"
201         prefWidth "73.0" text="Phone No.">
202             <font>
203                 <Font name="Cambria" size="15.0" />
204             </font>
205         </Label>
206         <TextField fx:id="phoneNoTextField" layoutX="32.0"
207         layoutY="194.0" prefHeight="26.0" prefWidth="176.0"
208         promptText="Eg 17358766" />
209         <Button fx:id="saveButton" layoutX="667.0" layoutY="644.0"
210         mnemonicParsing="false" onAction="#saveDetails"
211         prefHeight="14.0" prefWidth="66.0" text="Save">
212             <font>
213                 <Font name="Cambria" size="15.0" />
214             </font>
215         </Button>
216         <Button fx:id="backButton" layoutX="576.0" layoutY="645.0"
217         mnemonicParsing="false" onAction="#homePage" prefHeight "14.0"
218         prefWidth "66.0" text="Back">
219             <font>
220                 <Font name="Cambria" size="15.0" />
221             </font>
222         </Button>
223         <Label layoutX="31.0" layoutY="492.0" prefHeight="20.0"
224         prefWidth "75.0" text="Medicine ">
225             <font>
226                 <Font name="Cambria" size="15.0" />
227             </font>
228         </Label>
229         <ComboBox fx:id="medicineComboBox" layoutX="115.0" layoutY="489.0"
230         prefHeight="26.0" prefWidth "186.0" promptText="- - " />
231         <Label layoutX="31.0" layoutY="532.0" prefHeight="20.0"
232         prefWidth "80.0" text="Quantity">
233             <font>
234                 <Font name="Cambria" size="15.0" />
235             </font>
236         </Label>
237         <TextField fx:id="medicineQuantityTextField" layoutX="117.0"
238         layoutY "529.0" prefHeight="26.0" prefWidth "185.0" text "0" />
239         <Label layoutX="417.0" layoutY "491.0" prefHeight="20.0"
240         prefWidth "77.0" text="Injection ">
241             <font>
242                 <Font name="Cambria" size="15.0" />
243             </font>
244         </Label>
245         <ComboBox fx:id="injectionComboBox" layoutX="536.0" layoutY="489.0"
246         prefHeight="26.0" prefWidth "186.0" promptText="- - " />
247         <Label layoutX="416.0" layoutY "531.0" prefHeight="20.0"
248         prefWidth "80.0" text="Quantity">
249             <font>
250                 <Font name="Cambria" size="15.0" />
251             </font>
252         </Label>
253         <TextField fx:id="injectionQuantityTextField" layoutX="538.0"
254         layoutY "528.0" prefHeight="26.0" prefWidth "184.0" text "0" />
255         <CheckBox fx:id="medicineChecBox" layoutX="115.0"
256         layoutY "576.0" mnemonicParsing="false"
257         onAction="#medicinePrescribed" prefHeight="26.0"
258         prefWidth "202.0" text="Prescribe Medicine">
259             <font>
260                 <Font name="Cambria" size="15.0" />

```

```

242         </font>
243     </CheckBox>
244     <CheckBox fx:id="injectionChecBox" layoutX="538.0"
        layoutY="576.0" mnemonicParsing="false"
        onAction="#prescribeInjection" prefHeight="26.0"
        prefWidth="202.0" text="Prescribe Injection">
245         <font>
246             <Font name="Cambria" size="15.0" />
247         </font>
248     </CheckBox>
249     <Label fx:id="nameErrorMessage" layoutX="144.0" layoutY="73.0"
        prefHeight="20.0" prefWidth="115.0" text="*Please Enter your name"
        textFill="RED" visible="false">
250         <font>
251             <Font name="Arial" size="10.0" />
252         </font>
253     </Label>
254     <Label fx:id="stdNoErrorMessage" layoutX="331.0" layoutY="72.0"
        prefHeight="20.0" prefWidth="150.0" text="*Please Enter a Valid Student
        No" textFill="RED" visible="false">
255         <font>
256             <Font name="Arial" size="10.0" />
257         </font>
258     </Label>
259     <Label fx:id="ageErrorMessage" layoutX="172.0" layoutY="146.0"
        prefHeight="20.0" prefWidth="87.0" text="*Please Enter Age"
        textFill="RED" visible="false">
260         <font>
261             <Font name="Arial" size="10.0" />
262         </font>
263     </Label>
264     <Label fx:id="genderErrorMessage" layoutX="35.0" layoutY="146.0"
        prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
        textFill="RED" visible="false">
265         <font>
266             <Font name="Arial" size="10.0" />
267         </font>
268     </Label>
269     <Label fx:id="deptErrorMessage" layoutX="391.0" layoutY="146.0"
        prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
        textFill="RED" visible="false">
270         <font>
271             <Font name="Arial" size="10.0" />
272         </font>
273     </Label>
274     <Label fx:id="yearErrorMessage" layoutX="532.0" layoutY="146.0"
        prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
        textFill="RED" visible="false">
275         <font>
276             <Font name="Arial" size="10.0" />
277         </font>
278     </Label>
279     <Label fx:id="courseErrorMessage" layoutX="646.0" layoutY="146.0"
        prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
        textFill="RED" visible="false">
280         <font>
281             <Font name="Arial" size="10.0" />
282         </font>
283     </Label>
284     <Label fx:id="phoneNoErrorMessage" layoutX="63.0" layoutY="220.0"
        prefHeight="20.0" prefWidth="150.0" text="*Please Enter a Valid Phone
        No" textFill="RED" visible="false">
285         <font>
286             <Font name="Arial" size="10.0" />

```

```

287         </font>
288     </Label>
289     <Label fx:id="complaintErrorMessage" layoutX="409.0" layoutY="386.0"
prefHeight="20.0" prefWidth="75.0" text="*Please Fill Up"
textFill="RED" visible="false">
290         <font>
291             <Font name="Arial" size="10.0" />
292         </font>
293     </Label>
294     <Label fx:id="remarksErrorMessage" layoutX="679.0" layoutY="386.0"
prefHeight="20.0" prefWidth="73.0" text="*Please Filll Up"
textFill="RED" visible="false">
295         <font>
296             <Font name="Arial" size="10.0" />
297         </font>
298     </Label>
299     <Label fx:id="leaveDaysErrorMessage" layoutX="261.0" layoutY="455.0"
prefHeight="20.0" prefWidth="75.0" text="*Please Fill Up"
textFill="RED" visible="false">
300         <font>
301             <Font name="Arial" size="10.0" />
302         </font>
303     </Label>
304     <Label fx:id="leaveOptionsErrorMessage" layoutX="134.0" layoutY "455.0"
prefHeight="20.0" prefWidth "87.0" text "*Please Select One"
textFill="RED" visible="false">
305         <font>
306             <Font name="Arial" size="10.0" />
307         </font>
308     </Label>
309     </children>
310 </AnchorPane>
311 </content>
312 </ScrollPane>
313 </children>
314 </AnchorPane>
315 </center>
316 </BorderPane>
317

```



```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.geometry.Insets?>
6  <?import javafx.scene.control.Button?>
7  <?import javafx.scene.control.CheckBox?>
8  <?import javafx.scene.control.ComboBox?>
9  <?import javafx.scene.control.DatePicker?>
10 <?import javafx.scene.control.Label?>
11 <?import javafx.scene.control.Menu?>
12 <?import javafx.scene.control.MenuBar?>
13 <?import javafx.scene.control.MenuItem?>
14 <?import javafx.scene.control.RadioButton?>
15 <?import javafx.scene.control.ScrollPane?>
16 <?import javafx.scene.control.TextArea?>
17 <?import javafx.scene.control.TextField?>
18 <?import javafx.scene.control.ToggleGroup?>
19 <?import javafx.scene.layout.AnchorPane?>
20 <?import javafx.scene.layout.BorderPane?>
21 <?import javafx.scene.text.Font?>
22
23 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.StaffPatientDetailsEntryController">
24     <top>
25         <AnchorPane prefHeight="84.0" prefWidth="800.0" BorderPane.alignment="CENTER">
26             <children>
27                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
28                     <menus>
29                         <Menu mnemonicParsing="false" text="File">
30                             <items>
31                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
32                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
33                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
34                             </items>
35                         </Menu>
36                         <Menu mnemonicParsing="false" text="View">
37                             <items>
38                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
39                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
40                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
41                             </items>
42                         </Menu>
43                         <Menu mnemonicParsing="false" text="Help">
44                             <items>
45                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
46                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
47                             </items>
48                         </Menu>
49                     </menus>
50                 </MenuBar>

```

```

51         <Label alignment="CENTER" layoutY "26.0" prefHeight "58.0" prefWidth "800.0"
           style="-fx-background-color: #C5C6C7;" text="New Patient Entry - Staff"
           textAlignment="CENTER" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
           AnchorPane.topAnchor="26.0">
52             <font>
53                 <Font name="Roboto Condensed Bold" size="52.0" />
54             </font>
55             <padding>
56                 <Insets left="40.0" />
57             </padding>
58         </Label>
59     </children>
60 </AnchorPane>
61 </top>
62 <center>
63     <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment "CENTER">
64         <children>
65             <ScrollPane prefHeight "717.0" prefWidth="800.0" AnchorPane.bottomAnchor "-1.0"
               AnchorPane.leftAnchor "0.0" AnchorPane.rightAnchor "0.0"
               AnchorPane.topAnchor "0.0">
66                 <content>
67                     <AnchorPane prefHeight="641.0" prefWidth "799.0">
68                         <children>
69                             <TextArea fx:id="complaintTextArea" layoutX "31.0"
                               layoutY "266.0" prefHeight="120.0" prefWidth "450.0" />
70                             <Label layoutX="32.0" layoutY "25.0" prefHeight "16.0"
                               prefWidth "60.0" text "Name">
71                                 <font>
72                                     <Font name="Cambria" size="15.0" />
73                                 </font>
74                             </Label>
75                             <TextField fx:id="nameTextField" layoutX="32.0" layoutY "50.0"
                               prefHeight="26.0" prefWidth="215.0" />
76                             <Label layoutX="532.0" layoutY "26.0" prefHeight "16.0"
                               prefWidth "60.0" text "Reg No.">
77                                 <font>
78                                     <Font name="Cambria" size="15.0" />
79                                 </font>
80                             </Label>
81                             <TextField fx:id="regNoTextField" layoutX="532.0"
                               layoutY "51.0" prefHeight="26.0" prefWidth "176.0" />
82                             <!-- <items> -->
83                             <!-- <FXCollections fx:factory="observableArrayList"> -->
84                             <!-- <String fx:value="ME in RE" /> -->
85                             <!-- <String fx:value="ME in CM" /> -->
86                             <!-- <String fx:value="BE CE" /> -->
87                             <!-- <String fx:value="BE EE" /> -->
88                             <!-- <String fx:value="B Arch." /> -->
89                             <!-- <String fx:value="BE ECE" /> -->
90                             <!-- <String fx:value="BE IT" /> -->
91                             <!-- <String fx:value="BE ICE" /> -->
92                             <!-- <String fx:value="BE EG" /> -->
93                             <!-- </FXCollections> -->
94                             <!-- </items> -->
95                             <!-- </ComboBox> -->
96                             <Label layoutX="303.0" layoutY "25.0" prefHeight "20.0"
                               prefWidth "82.0" text "Department">
97                                 <font>
98                                     <Font name="Cambria" size="15.0" />
99                                 </font>
100                            </Label>
101                            <ComboBox fx:id="departmentComboBox" layoutX "303.0"
                               layoutY "50.0" onAction="#comboChanged" prefHeight="26.0"
                               prefWidth "176.0" promptText="-" />

```

```

102         <!-- <items> -->
103         <!-- <FXCollections fx:factory="observableArrayList"> -->
104         <!-- <String fx:value="Civil Department" /> -->
105         <!-- <String fx:value="Architecture Department" /> -->
106         <!-- <String fx:value="Electrical Department" /> -->
107         <!-- <String fx:value="IT Department" /> -->
108         <!-- <String fx:value="ECE Department" /> -->
109         <!-- <String fx:value="Electrical Department" /> -->
110         <!-- </FXCollections> -->
111         <!-- </items> -->
112         <!-- </ComboBox> -->
113         <Label layoutX="31.0" layoutY="95.0" prefHeight="16.0"
prefWidth="60.0" text="Sex">
114             <font>
115                 <Font name="Cambria" size="15.0" />
116             </font>
117         </Label>
118         <RadioButton fx:id="maleRadioButton" layoutX="31.0"
layoutY="124.0" mnemonicParsing="false"
onAction="#sexRadioSelect" text="Male">
119             <toggleGroup>
120                 <ToggleGroup fx:id="gender" />
121             </toggleGroup>
122         </RadioButton>
123         <RadioButton fx:id="femaleRadioButton" layoutX="90.0"
layoutY="124.0" mnemonicParsing="false"
onAction="#sexRadioSelect" text="Female" toggleGroup="$gender" />
124         <Label layoutX="170.0" layoutY="93.0" prefHeight="16.0"
prefWidth="60.0" text="Age">
125             <font>
126                 <Font name="Cambria" size="15.0" />
127             </font>
128         </Label>
129         <TextField fx:id="ageTextField" layoutX="170.0" layoutY="118.0"
prefHeight="26.0" prefWidth="75.0" />
130         <Label layoutX="534.0" layoutY="93.0" prefHeight="16.0"
prefWidth="60.0" text="Date">
131             <font>
132                 <Font name="Cambria" size="15.0" />
133             </font>
134         </Label>
135         <DatePicker fx:id="DateSelector" layoutX="534.0"
layoutY="118.0" prefHeight="26.0" prefWidth="175.0" />
136         <Label layoutX="28.0" layoutY="166.0" prefHeight="16.0"
prefWidth="60.0" text="Time">
137             <font>
138                 <Font name="Cambria" size="15.0" />
139             </font>
140         </Label>
141         <TextField fx:id="timeTextField" layoutX="28.0" layoutY="191.0"
prefHeight="26.0" prefWidth="81.0" promptText "Eg 12:00" />
142         <RadioButton fx:id="amRadioButton" layoutX="124.0"
layoutY="195.0" mnemonicParsing="false"
onAction="#timeRadioSelect" text="AM">
143             <toggleGroup>
144                 <ToggleGroup fx:id="timeSelect" />
145             </toggleGroup>
146         </RadioButton>
147         <RadioButton fx:id="pmRadioButton" layoutX="172.0"
layoutY="195.0" mnemonicParsing="false"
onAction="#timeRadioSelect" text="PM" toggleGroup="$timeSelect"
/>
148         <Label layoutX="31.0" layoutY="239.0" prefHeight="20.0"
prefWidth="159.0" text="Complaint / Treatment ">

```

```

149         <font>
150             <Font name "Cambria" size "15.0" />
151         </font>
152     </Label>
153     <Label layoutX="518.0" layoutY="239.0" prefHeight="20.0"
prefWidth "159.0" text="Remarks">
154         <font>
155             <Font name "Cambria" size "15.0" />
156         </font>
157     </Label>
158     <TextArea fx:id="remarksTextArea" layoutX="519.0"
layoutY "266.0" prefHeight="120.0" prefWidth "232.0" />
159     <Label layoutX="305.0" layoutY "165.0" prefHeight="20.0"
prefWidth "97.0" text="Leave Options ">
160         <font>
161             <Font name "Cambria" size "15.0" />
162         </font>
163     </Label>
164     <ComboBox fx:id="leaveOptionsComboBox" layoutX "305.0"
layoutY "191.0" onAction="#comboBoxChanged" prefHeight="26.0"
prefWidth "190.0" promptText="- - " />
165     <Label layoutX="538.0" layoutY "165.0" prefHeight="20.0"
prefWidth "82.0" text="No. of Days ">
166         <font>
167             <Font name "Cambria" size "15.0" />
168         </font>
169     </Label>
170     <TextField fx:id="leaveDaysTextField" layoutX "538.0"
layoutY "191.0" prefHeight="26.0" prefWidth "81.0" />
171     <Label layoutX="303.0" layoutY "94.0" prefHeight "20.0"
prefWidth "73.0" text="Phone No.">
172         <font>
173             <Font name "Cambria" size "15.0" />
174         </font>
175     </Label>
176     <TextField fx:id="phoneNoTextField" layoutX "303.0"
layoutY "119.0" prefHeight="26.0" prefWidth "176.0"
promptText="Eg 17358766" />
177     <Button fx:id="saveButton" layoutX "662.0" layoutY="576.0"
mnemonicParsing="false" onAction "#saveDetails"
prefHeight "14.0" prefWidth="66.0" text="Save">
178         <font>
179             <Font name "Cambria" size "15.0" />
180         </font>
181     </Button>
182     <Button fx:id="backButton" layoutX "571.0" layoutY="577.0"
mnemonicParsing="false" onAction "#homePage" prefHeight "14.0"
prefWidth "66.0" text="Back">
183         <font>
184             <Font name "Cambria" size "15.0" />
185         </font>
186     </Button>
187     <Label layoutX="30.0" layoutY="418.0" prefHeight="20.0"
prefWidth "75.0" text="Medicine ">
188         <font>
189             <Font name "Cambria" size "15.0" />
190         </font>
191     </Label>
192     <ComboBox fx:id="medicineComboBox" layoutX "114.0" layoutY="415.0"
prefHeight="26.0" prefWidth "186.0" promptText="- - " />
193     <Label layoutX="30.0" layoutY="458.0" prefHeight="20.0"
prefWidth "80.0" text="Quantity">
194         <font>
195             <Font name "Cambria" size "15.0" />

```

```

196         </font>
197     </Label>
198     <TextField fx:id="medicineQuantityTextField" layoutX="116.0"
199         layoutY="455.0" prefHeight="26.0" prefWidth="185.0" text="0" />
200     <Label layoutX="416.0" layoutY="417.0" prefHeight="20.0"
201         prefWidth="77.0" text="Injection ">
202         <font>
203             <Font name="Cambria" size="15.0" />
204         </font>
205     </Label>
206     <ComboBox fx:id="injectionComboBox" layoutX="535.0" layoutY="415.0"
207         prefHeight="26.0" prefWidth="186.0" promptText="- - " />
208     <Label layoutX="415.0" layoutY="457.0" prefHeight="20.0"
209         prefWidth="80.0" text="Quantity">
210         <font>
211             <Font name="Cambria" size="15.0" />
212         </font>
213     </Label>
214     <TextField fx:id="injectionQuantityTextField" layoutX="537.0"
215         layoutY="454.0" prefHeight="26.0" prefWidth="184.0" text="0" />
216     <CheckBox fx:id="medicineChecBox" layoutX="114.0"
217         layoutY="502.0" mnemonicParsing="false"
218         onAction="#medicinePrescribed" prefHeight="26.0"
219         prefWidth="202.0" text="Prescribe Medicine">
220         <font>
221             <Font name="Cambria" size="15.0" />
222         </font>
223     </CheckBox>
224     <CheckBox fx:id="injectionChecBox" layoutX="537.0"
225         layoutY="502.0" mnemonicParsing="false"
226         onAction="#prescribeInjection" prefHeight="26.0"
227         prefWidth="202.0" text="Prescribe Injection">
228         <font>
229             <Font name="Cambria" size="15.0" />
230         </font>
231     </CheckBox>
232     <Label fx:id="nameErrorMessage" layoutX="134.0" layoutY="76.0"
233         prefHeight="20.0" prefWidth="115.0" text="*Please Enter your name"
234         textFill="RED" visible="false">
235         <font>
236             <Font name="Arial" size="10.0" />
237         </font>
238     </Label>
239     <Label fx:id="ageErrorMessage" layoutX="172.0" layoutY="146.0"
240         prefHeight="20.0" prefWidth="87.0" text="*Please Enter Age"
241         textFill="RED" visible="false">
242         <font>
243             <Font name="Arial" size="10.0" />
244         </font>
245     </Label>
246     <Label fx:id="genderErrorMessage" layoutX="35.0" layoutY="146.0"
247         prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
248         textFill="RED" visible="false">
249         <font>
250             <Font name="Arial" size="10.0" />
251         </font>
252     </Label>
253     <Label fx:id="deptErrorMessage" layoutX="391.0" layoutY="76.0"
254         prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
255         textFill="RED" visible="false">
256         <font>
257             <Font name="Arial" size="10.0" />
258         </font>
259     </Label>

```

```

241      <Label fx:id="phoneNoErrorMessage" layoutX="333.0" layoutY="146.0"
      prefHeight="20.0" prefWidth="150.0" text="*Please Enter a Valid Phone
      No" textFill="RED" visible="false">
242          <font>
243              <Font name="Arial" size="10.0" />
244          </font>
245      </Label>
246      <Label fx:id="complaintErrorMessage" layoutX="409.0" layoutY="386.0"
      prefHeight="20.0" prefWidth="75.0" text="*Please Fill Up"
      textFill="RED" visible="false">
247          <font>
248              <Font name="Arial" size="10.0" />
249          </font>
250      </Label>
251      <Label fx:id="remarksErrorMessage" layoutX="678.0" layoutY="386.0"
      prefHeight="20.0" prefWidth="73.0" text="*Please Filll Up"
      textFill="RED" visible="false">
252          <font>
253              <Font name="Arial" size="10.0" />
254          </font>
255      </Label>
256      <Label fx:id="leaveDaysErrorMessage" layoutX="554.0" layoutY="217.0"
      prefHeight="20.0" prefWidth="75.0" text="*Please Fill Up"
      textFill="RED" visible="false">
257          <font>
258              <Font name="Arial" size="10.0" />
259          </font>
260      </Label>
261      <Label fx:id="leaveOptionsErrorMessage" layoutX="411.0" layoutY="217.0"
      prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
      textFill="RED" visible="false">
262          <font>
263              <Font name="Arial" size="10.0" />
264          </font>
265      </Label>
266      </children>
267  </AnchorPane>
268 </content>
269 </ScrollPane>
270 </children>
271 </AnchorPane>
272 </center>
273 </BorderPane>
274

```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.geometry.Insets?>
6  <?import javafx.scene.control.Button?>
7  <?import javafx.scene.control.CheckBox?>
8  <?import javafx.scene.control.ComboBox?>
9  <?import javafx.scene.control.DatePicker?>
10 <?import javafx.scene.control.Label?>
11 <?import javafx.scene.control.Menu?>
12 <?import javafx.scene.control.MenuBar?>
13 <?import javafx.scene.control.MenuItem?>
14 <?import javafx.scene.control.RadioButton?>
15 <?import javafx.scene.control.ScrollPane?>
16 <?import javafx.scene.control.TextArea?>
17 <?import javafx.scene.control.TextField?>
18 <?import javafx.scene.control.ToggleGroup?>
19 <?import javafx.scene.layout.AnchorPane?>
20 <?import javafx.scene.layout.BorderPane?>
21 <?import javafx.scene.text.Font?>
22
23 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.OtherPatientDetailsEntryController">
24     <top>
25         <AnchorPane prefHeight="84.0" prefWidth="800.0" BorderPane.alignment="CENTER">
26             <children>
27                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
28                     <menus>
29                         <Menu mnemonicParsing="false" text="File">
30                             <items>
31                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
32                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
33                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
34                             </items>
35                         </Menu>
36                         <Menu mnemonicParsing="false" text="View">
37                             <items>
38                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
39                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
40                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
41                             </items>
42                         </Menu>
43                         <Menu mnemonicParsing="false" text="Help">
44                             <items>
45                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
46                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
47                             </items>
48                         </Menu>
49                     </menus>
50                 </MenuBar>

```

```

1    <Label alignment="CENTER" layoutY="26.0" prefHeight="58.0" prefWidth="800.0"
2    style="-fx-background-color: #C5C6C7;" text="New Patient Entry - Others"
3    textAlignment="CENTER" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
4    AnchorPane.topAnchor="26.0">
5        <font>
6            <Font name="Roboto Condensed Bold" size="52.0" />
7        </font>
8        <padding>
9            <Insets left="40.0" />
10        </padding>
11    </Label>
12</children>
13</AnchorPane>
14</top>
15<center>
16    <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">
17        <children>
18            <ScrollPane prefHeight="717.0" prefWidth="800.0" AnchorPane.bottomAnchor="-1.0"
19            AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
20            AnchorPane.topAnchor="0.0">
21                <content>
22                    <AnchorPane prefHeight="641.0" prefWidth="799.0">
23                        <children>
24                            <TextArea fx:id="complaintTextArea" layoutX="35.0"
25                            layoutY="220.0" prefHeight="120.0" prefWidth="450.0" />
26                            <Label layoutX="32.0" layoutY="25.0" prefHeight="16.0"
27                            prefWidth="60.0" text="Name">
28                                <font>
29                                    <Font name="Cambria" size="15.0" />
30                                </font>
31                            </Label>
32                            <TextField fx:id="nameTextField" layoutX="32.0" layoutY="50.0"
33                            prefHeight="26.0" prefWidth="215.0" />
34                            <Label layoutX="532.0" layoutY="25.0" prefHeight="16.0"
35                            prefWidth="60.0" text="Reg No.">
36                                <font>
37                                    <Font name="Cambria" size="15.0" />
38                                </font>
39                            </Label>
40                            <TextField fx:id="regNoTextField" layoutX="532.0"
41                            layoutY="50.0" prefHeight="26.0" prefWidth="176.0" />
42                            <!-- <items> -->
43                            <!-- <FXCollections fx:factory="observableArrayList"> -->
44                            <!-- <String fx:value="Civil Department" /> -->
45                            <!-- <String fx:value="Architecture Department" /> -->
46                            <!-- <String fx:value="Electrical Department" /> -->
47                            <!-- <String fx:value="IT Department" /> -->
48                            <!-- <String fx:value="ECE Department" /> -->
49                            <!-- <String fx:value="Electrical Department" /> -->
50                            <!-- </FXCollections> -->
51                            <!-- </items> -->
52                            <!-- </ComboBox> -->
53                            <Label layoutX="34.0" layoutY="102.0" prefHeight="16.0"
54                            prefWidth="60.0" text="Sex">
55                                <font>
56                                    <Font name="Cambria" size="15.0" />
57                                </font>
58                            </Label>
59                            <RadioButton fx:id="maleRadioButton" layoutX="35.0"
60                            layoutY="131.0" mnemonicParsing="false"
61                            onAction="#sexRadioSelect" text="Male">
62                                <toggleGroup>
63                                    <ToggleGroup fx:id="gender" />
64                                </toggleGroup>

```



```

102         </RadioButton>
103         <RadioButton fx:id="femaleRadioButton" layoutX="94.0"
layoutY="131.0" mnemonicParsing="false"
onAction="#sexRadioSelect" text="Female" toggleGroup="$gender" />
104     <Label layoutX="173.0" layoutY="102.0" prefHeight="16.0"
prefWidth="60.0" text="Age">
105         <font>
106             <Font name="Cambria" size="15.0" />
107         </font>
108     </Label>
109     <TextField fx:id="ageTextField" layoutX="173.0" layoutY="127.0"
prefHeight="26.0" prefWidth="75.0" />
110     <Label layoutX="537.0" layoutY="102.0" prefHeight="16.0"
prefWidth="60.0" text="Date">
111         <font>
112             <Font name="Cambria" size="15.0" />
113         </font>
114     </Label>
115     <DatePicker fx:id="DateSelector" layoutX="537.0"
layoutY="127.0" prefHeight="26.0" prefWidth="175.0" />
116     <Label layoutX="299.0" layoutY="102.0" prefHeight="16.0"
prefWidth="60.0" text="Time">
117         <font>
118             <Font name="Cambria" size="15.0" />
119         </font>
120     </Label>
121     <TextField fx:id="timeTextField" layoutX="299.0"
layoutY="127.0" prefHeight="26.0" prefWidth="81.0"
promptText="Eg 12:00" />
122     <RadioButton fx:id="amRadioButton" layoutX="395.0"
layoutY="131.0" mnemonicParsing="false"
onAction="#timeRadioSelect" text="AM">
123         <toggleGroup>
124             <ToggleGroup fx:id="timeSelect" />
125         </toggleGroup>
126     </RadioButton>
127     <RadioButton fx:id="pmRadioButton" layoutX="443.0"
layoutY="131.0" mnemonicParsing="false"
onAction="#timeRadioSelect" text="PM" toggleGroup="$timeSelect"
/>
128     <Label layoutX="35.0" layoutY="193.0" prefHeight="20.0"
prefWidth="159.0" text="Complaint / Treatment ">
129         <font>
130             <Font name="Cambria" size="15.0" />
131         </font>
132     </Label>
133     <Label layoutX="522.0" layoutY="193.0" prefHeight="20.0"
prefWidth="159.0" text="Remarks">
134         <font>
135             <Font name="Cambria" size="15.0" />
136         </font>
137     </Label>
138     <TextArea fx:id="remarksTextArea" layoutX="523.0"
layoutY="220.0" prefHeight="120.0" prefWidth="232.0" />
139     <Label layoutX="300.0" layoutY="25.0" prefHeight="20.0"
prefWidth="73.0" text="Phone No.">
140         <font>
141             <Font name="Cambria" size="15.0" />
142         </font>
143     </Label>
144     <TextField fx:id="phoneNoTextField" layoutX="300.0"
layoutY="50.0" prefHeight="26.0" prefWidth="176.0"
promptText="Eg 17358766" />
145     <Button fx:id="saveButton" layoutX="666.0" layoutY="530.0"

```

```

146 mnemonicParsing="false" onAction "#saveDetails"
147 prefHeight="14.0" prefWidth="66.0" text="Save">
148     <font>
149         <Font name="Cambria" size="15.0" />
150     </font>
151 </Button>
152 <Button fx:id="backButton" layoutX="575.0" layoutY="531.0"
153 mnemonicParsing="false" onAction "#homePage" prefHeight="14.0"
154 prefWidth="66.0" text="Back">
155     <font>
156         <Font name="Cambria" size="15.0" />
157     </font>
158 </Button>
159 <Label layoutX="34.0" layoutY="372.0" prefHeight="20.0"
160 prefWidth="75.0" text="Medicine ">
161     <font>
162         <Font name="Cambria" size="15.0" />
163     </font>
164 </Label>
165 <ComboBox fx:id="medicineComboBox" layoutX="118.0" layoutY="369.0"
166 prefHeight="26.0" prefWidth="186.0" promptText="- - " />
167 <Label layoutX="34.0" layoutY="412.0" prefHeight="20.0"
168 prefWidth="80.0" text="Quantity">
169     <font>
170         <Font name="Cambria" size="15.0" />
171     </font>
172 </Label>
173 <TextField fx:id="medicineQuantityTextField" layoutX="120.0"
174 layoutY="409.0" prefHeight="26.0" prefWidth="185.0" text="0" />
175 <Label layoutX="420.0" layoutY="371.0" prefHeight="20.0"
176 prefWidth="77.0" text="Injection ">
177     <font>
178         <Font name="Cambria" size="15.0" />
179     </font>
180 </Label>
181 <ComboBox fx:id="injectionComboBox" layoutX="539.0" layoutY="369.0"
182 prefHeight="26.0" prefWidth="186.0" promptText="- - " />
183 <Label layoutX="419.0" layoutY="411.0" prefHeight="20.0"
184 prefWidth="80.0" text="Quantity">
185     <font>
186         <Font name="Cambria" size="15.0" />
187     </font>
188 </Label>
189 <TextField fx:id="injectionQuantityTextField" layoutX="541.0"
190 layoutY="408.0" prefHeight="26.0" prefWidth="184.0" text="0" />
191 <CheckBox fx:id="medicineChecBox" layoutX="118.0"
192 layoutY="456.0" mnemonicParsing="false"
193 onAction "#medicinePrescribed" prefHeight="26.0"
194 prefWidth="202.0" text="Prescribe Medicine">
195     <font>
196         <Font name="Cambria" size="15.0" />
197     </font>
198 </CheckBox>
199 <CheckBox fx:id="injectionChecBox" layoutX="541.0"
200 layoutY="456.0" mnemonicParsing="false"
201 onAction "#prescribeInjection" prefHeight="26.0"
202 prefWidth="202.0" text="Prescribe Injection">
203     <font>
204         <Font name="Cambria" size="15.0" />
205     </font>
206 </CheckBox>
207 <Label fx:id="nameErrorMessage" layoutX="134.0" layoutY="76.0"
208 prefHeight="20.0" prefWidth="115.0" text="*Please Enter your name"
209 textFill="RED" visible="false">

```

```

190         <font>
191             <Font name "Arial" size "10.0" />
192         </font>
193     </Label>
194     <Label fx:id="ageErrorMessage" layoutX="175.0" layoutY="152.0"
prefHeight="20.0" prefWidth="87.0" text="*Please Enter Age"
textFill="RED" visible="false">
195         <font>
196             <Font name "Arial" size "10.0" />
197         </font>
198     </Label>
199     <Label fx:id="genderErrorMessage" layoutX="39.0" layoutY="153.0"
prefHeight="20.0" prefWidth="87.0" text="*Please Select One"
textFill="RED" visible="false">
200         <font>
201             <Font name "Arial" size "10.0" />
202         </font>
203     </Label>
204     <Label fx:id="phoneNoErrorMessage" layoutX="330.0" layoutY="77.0"
prefHeight="20.0" prefWidth="150.0" text="*Please Enter a Valid Phone
No" textFill="RED" visible="false">
205         <font>
206             <Font name "Arial" size "10.0" />
207         </font>
208     </Label>
209     <Label fx:id="complaintErrorMessage" layoutX="413.0" layoutY="340.0"
prefHeight="20.0" prefWidth="75.0" text="*Please Fill Up"
textFill="RED" visible="false">
210         <font>
211             <Font name "Arial" size "10.0" />
212         </font>
213     </Label>
214     <Label fx:id="remarksErrorMessage" layoutX="682.0" layoutY="340.0"
prefHeight="20.0" prefWidth="73.0" text="*Please Filll Up"
textFill="RED" visible="false">
215         <font>
216             <Font name "Arial" size "10.0" />
217         </font>
218     </Label>
219     </children>
220 </AnchorPane>
221 </content>
222 </ScrollPane>
223 </children>
224 </AnchorPane>
225 </center>
226 </BorderPane>
227

```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.Label?>
6  <?import javafx.scene.control.Menu?>
7  <?import javafx.scene.control.MenuBar?>
8  <?import javafx.scene.control.MenuItem?>
9  <?import javafx.scene.control.ScrollPane?>
10 <?import javafx.scene.control.TableColumn?>
11 <?import javafx.scene.control.TableView?>
12 <?import javafx.scene.control.TextField?>
13 <?import javafx.scene.effect.ImageInput?>
14 <?import javafx.scene.image.Image?>
15 <?import javafx.scene.layout.AnchorPane?>
16 <?import javafx.scene.layout.BorderPane?>
17 <?import javafx.scene.text.Font?>
18
19 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.PatientRecordsController">
20     <top>
21         <AnchorPane prefHeight="82.0" prefWidth="800.0" BorderPane.alignment="CENTER">
22             <children>
23                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
24                     <menus>
25                         <Menu mnemonicParsing="false" text="File">
26                             <items>
27                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
28                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
29                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
30                             </items>
31                         </Menu>
32                         <Menu mnemonicParsing="false" text="View">
33                             <items>
34                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
36                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
37                             </items>
38                         </Menu>
39                         <Menu mnemonicParsing="false" text="Help">
40                             <items>
41                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
42                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
43                             </items>
44                         </Menu>
45                     </menus>
46                 </MenuBar>
47                 <Label alignment="CENTER" layoutY="24.0" prefHeight="62.0" prefWidth="800.0"
style="-fx-background-color: #C5C6C7;" text="Patient Records - Student"
textAlignment="CENTER" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="24.0">

```

```

48         <font>
49             <Font name="Roboto Condensed Bold" size="52.0" />
50         </font>
51         <padding>
52             <Insets left="40.0" />
53         </padding>
54     </Label>
55 </children>
56 </AnchorPane>
57 </top>
58 <center>
59     <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">
60         <children>
61             <ScrollPane prefHeight="516.0" prefWidth="802.0" AnchorPane.bottomAnchor="0.0"
62                 AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
63                 AnchorPane.topAnchor="0.0">
64                 <content>
65                     <AnchorPane minHeight="1000.0" minWidth="3125.0" prefHeight="515.0"
66                         prefWidth="800.0">
67                         <children>
68                             <TableView id="recordsTable" fx:id="recordsTable"
69                                 editable="true" layoutX="60.0" layoutY="69.0"
70                                 prefHeight="900.0" prefWidth="3015.0">
71                                 <columns>
72                                     <TableColumn fx:id="col_regNo"
73                                         prefWidth="123.19998168945312" text="Reg No" />
74                                     <TableColumn fx:id="col_name"
75                                         prefWidth="275.1999969482422" text="Name" />
76                                     <TableColumn fx:id="col_stdNo"
77                                         prefWidth="157.5999755859375" text="Student No" />
78                                     <TableColumn fx:id="col_year"
79                                         minWidth="8.79998779296875"
80                                         prefWidth="77.5999755859375" text="Year" />
81                                     <TableColumn fx:id="col_course"
82                                         minWidth="1.5999755859375"
83                                         prefWidth="159.20001220703125" text="Course" />
84                                     <TableColumn fx:id="col_dept"
85                                         prefWidth="202.39996337890625" text="Department" />
86                                     <TableColumn fx:id="col_sex"
87                                         prefWidth="39.2000732421875" text="Sex" />
88                                     <TableColumn fx:id="col_age" minWidth="0.0"
89                                         prefWidth="38.39990234375" text="Age" />
90                                     <TableColumn fx:id="col_phoneNo"
91                                         prefWidth="117.5999755859375" text="Phone No." />
92                                     <TableColumn fx:id="col_date" minWidth="0.0"
93                                         prefWidth="148.8001708984375" text="Date" />
94                                     <TableColumn fx:id="col_time" prefWidth="108.0"
95                                         text="Time" />
96                                     <TableColumn fx:id="col_leaveOptions"
97                                         prefWidth="283.2000732421875" text="Leave Options" />
98                                     <TableColumn fx:id="col_leaveDays"
99                                         minWidth="3.199951171875" prefWidth="95.2000732421875"
100                                         text="No. of Days" />
101                                     <TableColumn fx:id="col_complaint"
102                                         prefWidth="703.9998779296875"
103                                         text="Complaint/Treatment" />
104                                     <TableColumn fx:id="col_remarks" minWidth="0.0"
105                                         prefWidth="480.800048828125" text="Remarks" />
106                                 </columns>
107                             </TableView>
108                             <Button fx:id="logoutButton" layoutX="650.0" layoutY="22.0"
109                                 mnemonicParsing="false" onAction="#logout" prefHeight="27.0"
110                                 prefWidth="62.0" text="Logout">
111                                 <font>

```

```

86         <Font name="Roboto Condensed Bold" size="15.0" />
87     </font>
88 </Button>
89 <Button fx:id="backButton" layoutX="571.0" layoutY="21.0"
mnemonicParsing="false" onAction="#backToHomePage"
prefHeight="27.0" prefWidth="62.0" text="Back">
90     <font>
91         <Font name="Roboto Condensed Bold" size="15.0" />
92     </font>
93 </Button>
94 <TextField fx:id="filterByTextField" layoutX="114.0"
layoutY="26.0" prefHeight="26.0" prefWidth="157.0"
promptText="Filter by... " />
95 <Label layoutX="60.0" layoutY="27.0" text="Search: ">
96     <font>
97         <Font name="Roboto Condensed Regular" size="15.0" />
98     </font>
99 </Label>
100 <Button fx:id="searchButton" layoutX="281.0" layoutY="23.0"
mnemonicParsing="false" onAction="#filterTable" prefHeight="30.0"
prefWidth="30.0" text="Search">
101     <font>
102         <Font name="Roboto Condensed Bold" size="15.0" />
103     </font>
104     <effect>
105         <ImageInput>
106             <source>
107                 <Image
url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
tSystem/Images/Search.png" />
108             </source>
109         </ImageInput>
110     </effect>
111 </Button>
112 <Button fx:id="refreshButton" layoutX="321.0" layoutY="23.0"
mnemonicParsing="false" onAction="#initialize" prefHeight="30.0"
prefWidth="30.0" text="Refresh">
113     <font>
114         <Font name="Roboto Condensed Bold" size="15.0" />
115     </font>
116     <effect>
117         <ImageInput>
118             <source>
119                 <Image
url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
tSystem/Images/Refresh.png" />
120             </source>
121         </ImageInput>
122     </effect>
123 </Button>
124 <Button fx:id="deleteButton" layoutX="361.0" layoutY="23.0"
mnemonicParsing="false" onAction="#deleteRow" prefHeight="30.0"
prefWidth="30.0" text="Delete">
125     <font>
126         <Font name="Roboto Condensed Bold" size="15.0" />
127     </font>
128     <effect>
129         <ImageInput>
130             <source>
131                 <Image
url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
tSystem/Images/Delete.png" />
132             </source>
133         </ImageInput>

```

```
134         </effect>
135     </Button>
136     </children>
137 </AnchorPane>
138 </content>
139 </ScrollPane>
140 </children>
141 </AnchorPane>
142 </center>
143 </BorderPane>
144
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.Label?>
6  <?import javafx.scene.control.Menu?>
7  <?import javafx.scene.control.MenuBar?>
8  <?import javafx.scene.control.MenuItem?>
9  <?import javafx.scene.control.ScrollPane?>
10 <?import javafx.scene.control.TableColumn?>
11 <?import javafx.scene.control.TableView?>
12 <?import javafx.scene.control.TextField?>
13 <?import javafx.scene.effect.ImageInput?>
14 <?import javafx.scene.image.Image?>
15 <?import javafx.scene.layout.AnchorPane?>
16 <?import javafx.scene.layout.BorderPane?>
17 <?import javafx.scene.text.Font?>
18
19 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.StaffPatientRecordsController">
20     <top>
21         <AnchorPane prefHeight="82.0" prefWidth="800.0" BorderPane.alignment="CENTER">
22             <children>
23                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
24                     <menus>
25                         <Menu mnemonicParsing="false" text="File">
26                             <items>
27                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
28                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
29                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
30                             </items>
31                         </Menu>
32                         <Menu mnemonicParsing="false" text="View">
33                             <items>
34                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
36                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
37                             </items>
38                         </Menu>
39                         <Menu mnemonicParsing="false" text="Help">
40                             <items>
41                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
42                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
43                             </items>
44                         </Menu>
45                     </menus>
46                 </MenuBar>
47                 <Label alignment="CENTER" layoutY="24.0" prefHeight="62.0" prefWidth="800.0"
style="-fx-background-color: #C5C6C7;" text="Patient Records - Staffs"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="24.0">

```



```

48         <font>
49             <Font name="Roboto Condensed Bold" size="52.0" />
50         </font>
51         <padding>
52             <Insets left="40.0" />
53         </padding>
54     </Label>
55 </children>
56 </AnchorPane>
57 </top>
58 <center>
59     <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">
60         <children>
61             <ScrollPane prefHeight="516.0" prefWidth="802.0" AnchorPane.bottomAnchor="0.0"
62             AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
63             AnchorPane.topAnchor="0.0">
64                 <content>
65                     <AnchorPane minHeight="1000.0" minWidth="2755.0" prefHeight="515.0"
66                     prefWidth="800.0">
67                         <children>
68                             <TableView id="recordsTable" fx:id="recordsTable"
69                             editable="true" layoutX="60.0" layoutY="69.0"
70                             prefHeight="900.0" prefWidth="2459.0">
71                                 <columns>
72                                     <TableColumn fx:id="col_regNo"
73                                     prefWidth="123.19998168945312" text="Reg No" />
74                                     <TableColumn fx:id="col_name"
75                                     prefWidth="275.1999969482422" text="Name" />
76                                     <TableColumn fx:id="col_dept"
77                                     prefWidth="202.39996337890625" text="Department" />
78                                     <TableColumn fx:id="col_sex"
79                                     prefWidth="39.2000732421875" text="Sex" />
80                                     <TableColumn fx:id="col_age" minWidth="0.0"
81                                     prefWidth="38.39990234375" text="Age" />
82                                     <TableColumn fx:id="col_phoneNo"
83                                     prefWidth="117.5999755859375" text="Phone No." />
84                                     <TableColumn fx:id="col_date" minWidth="0.0"
85                                     prefWidth="148.8001708984375" text="Date" />
86                                     <TableColumn fx:id="col_time" prefWidth="108.0"
87                                     text="Time" />
88                                     <TableColumn fx:id="col_leaveOptions"
89                                     prefWidth="283.2000732421875" text="Leave Options" />
90                                     <TableColumn fx:id="col_leaveDays"
91                                     minWidth="3.199951171875" prefWidth="95.2000732421875"
92                                     text="No. of Days" />
93                                     <TableColumn fx:id="col_complaint"
94                                     prefWidth="703.9998779296875"
95                                     text="Complaint/Treatment" />
96                                     <TableColumn fx:id="col_remarks" minWidth="0.0"
97                                     prefWidth="318.400146484375" text="Remarks" />
98                                 </columns>
99                             </TableView>
100                             <Button fx:id="logoutButton" layoutX="650.0" layoutY="22.0"
101                             mnemonicParsing="false" onAction="#logout" prefHeight="27.0"
102                             prefWidth="62.0" text="Logout">
103                                 <font>
104                                     <Font name="Roboto Condensed Bold" size="15.0" />
105                                 </font>
106                             </Button>
107                             <Button fx:id="backButton" layoutX="571.0" layoutY="21.0"
108                             mnemonicParsing="false" onAction="#backToHomePage"
109                             prefHeight="27.0" prefWidth="62.0" text="Back">
110                                 <font>
111                                     <Font name="Roboto Condensed Bold" size="15.0" />

```

```

89         </font>
90     </Button>
91     <TextField fx:id="filterByTextField" layoutX="114.0"
          layoutY="26.0" prefHeight="26.0" prefWidth="157.0"
          promptText="Filter by... " />
92     <Label layoutX="60.0" layoutY="27.0" text="Search: ">
93         <font>
94             <Font name="Roboto Condensed Regular" size="15.0" />
95         </font>
96     </Label>
97     <Button fx:id="searchButton" layoutX="281.0" layoutY="23.0"
mnemonicParsing="false" onAction="#filterTable" prefHeight="30.0"
prefWidth="30.0" text="Search">
98         <font>
99             <Font name="Roboto Condensed Bold" size="15.0" />
100         </font>
101         <effect>
102             <ImageInput>
103                 <source>
104                     <Image
          url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
          tSystem/Images/Search.png" />
105                 </source>
106             </ImageInput>
107         </effect>
108     </Button>
109     <Button fx:id="refreshButton" layoutX="321.0" layoutY="23.0"
mnemonicParsing="false" onAction="#initialize" prefHeight="30.0"
prefWidth="30.0" text="Refresh">
110         <font>
111             <Font name="Roboto Condensed Bold" size="15.0" />
112         </font>
113         <effect>
114             <ImageInput>
115                 <source>
116                     <Image
          url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
          tSystem/Images/Refresh.png" />
117                 </source>
118             </ImageInput>
119         </effect>
120     </Button>
121     <Button fx:id="deleteButton" layoutX="361.0" layoutY="23.0"
mnemonicParsing="false" onAction="#deleteRow" prefHeight="30.0"
prefWidth="30.0" text="Delete">
122         <font>
123             <Font name="Roboto Condensed Bold" size="15.0" />
124         </font>
125         <effect>
126             <ImageInput>
127                 <source>
128                     <Image
          url="file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
          tSystem/Images/Delete.png" />
129                 </source>
130             </ImageInput>
131         </effect>
132     </Button>
133     </children>
134 </AnchorPane>
135 </content>
136 </ScrollPane>
137 </children>
138 </AnchorPane>

```

```
139         </center>
140     </BorderPane>
141
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.Label?>
6  <?import javafx.scene.control.Menu?>
7  <?import javafx.scene.control.MenuBar?>
8  <?import javafx.scene.control.MenuItem?>
9  <?import javafx.scene.control.ScrollPane?>
10 <?import javafx.scene.control.TableColumn?>
11 <?import javafx.scene.control.TableView?>
12 <?import javafx.scene.control.TextField?>
13 <?import javafx.scene.effect.ImageInput?>
14 <?import javafx.scene.image.Image?>
15 <?import javafx.scene.layout.AnchorPane?>
16 <?import javafx.scene.layout.BorderPane?>
17 <?import javafx.scene.text.Font?>
18
19 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.OtherPatientRecordsController">
20     <top>
21         <AnchorPane prefHeight="82.0" prefWidth="800.0" BorderPane.alignment "CENTER">
22             <children>
23                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
24                     <menus>
25                         <Menu mnemonicParsing="false" text="File">
26                             <items>
27                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
28                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
29                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
30                             </items>
31                         </Menu>
32                         <Menu mnemonicParsing="false" text="View">
33                             <items>
34                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
36                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
37                             </items>
38                         </Menu>
39                         <Menu mnemonicParsing="false" text="Help">
40                             <items>
41                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
42                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
43                             </items>
44                         </Menu>
45                     </menus>
46                 </MenuBar>
47                 <Label alignment="CENTER" layoutY="24.0" prefHeight="62.0" prefWidth="800.0"
style="-fx-background-color: #C5C6C7;" text="Patient Records - Others"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="24.0">

```

```

48         <font>
49             <Font name="Roboto Condensed Bold" size="52.0" />
50         </font>
51         <padding>
52             <Insets left="40.0" />
53         </padding>
54     </Label>
55 </children>
56 </AnchorPane>
57 </top>
58 <center>
59     <AnchorPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">
60         <children>
61             <ScrollPane prefHeight="516.0" prefWidth="802.0" AnchorPane.bottomAnchor="0.0"
62                 AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
63                 AnchorPane.topAnchor="0.0">
64                 <content>
65                     <AnchorPane minHeight="1000.0" minWidth="2050.0" prefHeight="515.0"
66                         prefWidth="800.0">
67                         <children>
68                             <TableView id="recordsTable" fx:id="recordsTable"
69                                 editable="true" layoutX="60.0" layoutY="69.0"
70                                 prefHeight="900.0" prefWidth="1879.0">
71                                 <columns>
72                                     <TableColumn fx:id="col_regNo"
73                                         prefWidth="123.19998168945312" text="Reg No" />
74                                     <TableColumn fx:id="col_name"
75                                         prefWidth="275.1999969482422" text="Name" />
76                                     <TableColumn fx:id="col_sex"
77                                         prefWidth="39.2000732421875" text="Sex" />
78                                     <TableColumn fx:id="col_age" minWidth="0.0"
79                                         prefWidth="38.39990234375" text="Age" />
80                                     <TableColumn fx:id="col_phoneNo"
81                                         prefWidth="117.5999755859375" text="Phone No." />
82                                     <TableColumn fx:id="col_date" minWidth="0.0"
83                                         prefWidth="148.8001708984375" text="Date" />
84                                     <TableColumn fx:id="col_time" prefWidth="108.0"
85                                         text="Time" />
86                                     <TableColumn fx:id="col_complaint"
87                                         prefWidth="703.9998779296875"
88                                         text="Complaint/Treatment" />
89                                     <TableColumn fx:id="col_remarks" minWidth="0.0"
90                                         prefWidth="318.400146484375" text="Remarks" />
91                                 </columns>
92                             </TableView>
93                             <Button fx:id="logoutButton" layoutX="650.0" layoutY="22.0"
94                                 mnemonicParsing="false" onAction="#logout" prefHeight="27.0"
95                                 prefWidth="62.0" text="Logout">
96                                 <font>
97                                     <Font name="Roboto Condensed Bold" size="15.0" />
98                                 </font>
99                             </Button>
100                             <Button fx:id="backButton" layoutX="571.0" layoutY="21.0"
101                                 mnemonicParsing="false" onAction="#backToHomePage"
102                                 prefHeight="27.0" prefWidth="62.0" text="Back">
103                                 <font>
104                                     <Font name="Roboto Condensed Bold" size="15.0" />
105                                 </font>
106                             </Button>
107                             <TextField fx:id="filterByTextField" layoutX="114.0"
108                                 layoutY="26.0" prefHeight="26.0" prefWidth="157.0"
109                                 promptText="Filter by... " />
110                             <Label layoutX="60.0" layoutY="27.0" text="Search: ">
111                                 <font>

```

```

91         <Font name="Roboto Condensed Regular" size="15.0" />
92     </font>
93 </Label>
94 <Button fx:id="searchButton" layoutX="281.0" layoutY="23.0"
mnemonicParsing="false" onAction="#filterTable" prefHeight="30.0"
prefWidth="30.0" text="Search">
95     <font>
96         <Font name="Roboto Condensed Bold" size="15.0" />
97     </font>
98     <effect>
99         <ImageInput>
100             <source>
101                 <Image
url "file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
tSystem/Images/Search.png" />
102             </source>
103         </ImageInput>
104     </effect>
105 </Button>
106 <Button fx:id="refreshButton" layoutX="321.0" layoutY="23.0"
mnemonicParsing="false" onAction="#initialize" prefHeight="30.0"
prefWidth="30.0" text="Refresh">
107     <font>
108         <Font name="Roboto Condensed Bold" size="15.0" />
109     </font>
110     <effect>
111         <ImageInput>
112             <source>
113                 <Image
url "file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
tSystem/Images/Refresh.png" />
114             </source>
115         </ImageInput>
116     </effect>
117 </Button>
118 <Button fx:id="deleteButton" layoutX="361.0" layoutY="23.0"
mnemonicParsing="false" onAction="#deleteRow" prefHeight="30.0"
prefWidth="30.0" text="Delete">
119     <font>
120         <Font name="Roboto Condensed Bold" size="15.0" />
121     </font>
122     <effect>
123         <ImageInput>
124             <source>
125                 <Image
url "file:/C:/Users/USER/eclipse-workspace/InfirmaryManagemen
tSystem/Images/Delete.png" />
126             </source>
127         </ImageInput>
128     </effect>
129 </Button>
130 </children>
131 </AnchorPane>
132 </content>
133 </ScrollPane>
134 </children>
135 </AnchorPane>
136 </center>
137 </BorderPane>
138

```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.scene.control.Button?>
6  <?import javafx.scene.control.Hyperlink?>
7  <?import javafx.scene.control.Label?>
8  <?import javafx.scene.control.Menu?>
9  <?import javafx.scene.control.MenuBar?>
10 <?import javafx.scene.control.MenuItem?>
11 <?import javafx.scene.control.Separator?>
12 <?import javafx.scene.image.Image?>
13 <?import javafx.scene.image.ImageView?>
14 <?import javafx.scene.layout.AnchorPane?>
15 <?import javafx.scene.layout.BorderPane?>
16 <?import javafx.scene.text.Font?>
17
18 <BorderPane focusTraversable="true" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0"
style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.CollegeWebsiteLinksController">
19     <top>
20         <AnchorPane prefHeight="183.0" prefWidth="800.0" BorderPane.alignment="CENTER">
21             <children>
22                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
23                     <menus>
24                         <Menu mnemonicParsing="false" text="File">
25                             <items>
26                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
27                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
28                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
29                             </items>
30                         </Menu>
31                         <Menu mnemonicParsing="false" text="View">
32                             <items>
33                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
34                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
35                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
36                             </items>
37                         </Menu>
38                         <Menu mnemonicParsing="false" text="Help">
39                             <items>
40                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
41                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
42                             </items>
43                         </Menu>
44                     </menus>
45                 </MenuBar>
46                 <ImageView fitHeight="250.0" fitWidth="800.0" layoutY="26.0"
pickOnBounds="true" preserveRatio="true">
47                     <image>
48                         <Image url="@../Images/HomePageHeader.png" />
49                     </image>

```

```

50         </ImageView>
51     </children>
52 </AnchorPane>
53 </top>
54 <center>
55     <AnchorPane prefHeight="350.0" prefWidth="800.0" BorderPane.alignment="CENTER">
56         <children>
57             <Button fx:id="HomePage" layoutX="599.0" layoutY="337.0"
58                 mnemonicParsing="false" onAction="#HomePage" prefHeight="26.0" prefWidth="74.0"
59                 text="Home">
60                 <font>
61                     <Font name="Roboto Condensed Bold" size="18.0" />
62                 </font>
63             </Button>
64             <Button fx:id="Logout" layoutX="687.0" layoutY="337.0" mnemonicParsing="false"
65                 onAction="#userLogout" prefHeight="26.0" prefWidth="74.0" text="Logout">
66                 <font>
67                     <Font name="Roboto Condensed Bold" size="18.0" />
68                 </font>
69             </Button>
70             <Label layoutX="250.0" layoutY="52.0" prefHeight="28.0" prefWidth="104.0"
71                 text="CST Website">
72                 <font>
73                     <Font name="Roboto Condensed Bold" size="20.0" />
74                 </font>
75             </Label>
76             <Hyperlink fx:id="cstWebsite" layoutX="397.0" layoutY="55.0" onAction="#cstWebsite"
77                 prefHeight="23.0" prefWidth="183.0" text="https://cst.edu.bt/index.php/en/" />
78             <Label layoutX="250.0" layoutY="92.0" prefHeight="28.0" prefWidth="114.0" text="RUB
79                 IMS">
80                 <font>
81                     <Font name="Roboto Condensed Bold" size="20.0" />
82                 </font>
83             </Label>
84             <Hyperlink fx:id="rubIMS" layoutX="397.0" layoutY="94.0" onAction="#rubIMS"
85                 prefHeight="23.0" prefWidth="183.0" text="https://ims.rub.edu.bt/" />
86             <Label layoutX="250.0" layoutY="132.0" prefHeight="28.0" prefWidth="114.0"
87                 text="Mail">
88                 <font>
89                     <Font name="Roboto Condensed Bold" size="20.0" />
90                 </font>
91             </Label>
92             <Hyperlink fx:id="mail" layoutX="397.0" layoutY="134.0" onAction="#mail"
93                 prefHeight="23.0" prefWidth="183.0" text="http://mail.rub.edu.bt/" />
94             <Label layoutX="250.0" layoutY="172.0" prefHeight="28.0" prefWidth="114.0"
95                 text="VLE (Moodle)">
96                 <font>
97                     <Font name="Roboto Condensed Bold" size="20.0" />
98                 </font>
99             </Label>
100             <Hyperlink fx:id="vle" layoutX="397.0" layoutY="174.0" onAction="#vle"
101                 prefHeight="23.0" prefWidth="183.0" text="https://vle.cst.edu.bt/" />
102             <Label layoutX="250.0" layoutY="212.0" prefHeight="28.0" prefWidth="114.0"
103                 text="Library">
104                 <font>
105                     <Font name="Roboto Condensed Bold" size="20.0" />
106                 </font>
107             </Label>
108             <Hyperlink fx:id="library" layoutX="397.0" layoutY="214.0" onAction="#library"
109                 prefHeight="23.0" prefWidth="183.0" text="https://lib.cst.edu.bt/" />
110             <Label layoutX="250.0" layoutY="252.0" prefHeight="28.0" prefWidth="114.0"
111                 text="MRBS">
112                 <font>
113                     <Font name="Roboto Condensed Bold" size="20.0" />

```



```
100         </font>
101     </Label>
102     <Hyperlink fx:id="mrbs" layoutX="397.0" layoutY="254.0" onAction="#mrbs"
103     prefHeight="23.0" prefWidth="223.0" text="https://vle.cst.edu.bt/mrbs/web/day.php" />
104     <Separator layoutX="378.0" layoutY="48.0" orientation="VERTICAL" prefHeight="234.0"
105     prefWidth="4.0" />
106     <Label layoutX="319.0" layoutY="-4.0" prefHeight="31.0" prefWidth="124.0" text="Web
107     Links">
108         <font>
109             <Font name="Cambria Bold" size="25.0" />
110         </font>
111     </Label>
112 </children>
113 </AnchorPane>
114 </center>
115 </BorderPane>
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- <!DOCTYPE xml> -->
4
5  <?import javafx.scene.control.Button?>
6  <?import javafx.scene.control.Label?>
7  <?import javafx.scene.control.Menu?>
8  <?import javafx.scene.control.MenuBar?>
9  <?import javafx.scene.control.MenuItem?>
10 <?import javafx.scene.control.Separator?>
11 <?import javafx.scene.image.Image?>
12 <?import javafx.scene.image.ImageView?>
13 <?import javafx.scene.layout.AnchorPane?>
14 <?import javafx.scene.layout.BorderPane?>
15 <?import javafx.scene.text.Font?>
16
17 <BorderPane focusTraversable="true" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0"
style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.CollegeWebsiteLinksController">
18     <top>
19         <AnchorPane prefHeight="183.0" prefWidth="800.0" BorderPane.alignment="CENTER">
20             <children>
21                 <MenuBar prefHeight="19.0" prefWidth="800.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
22                     <menus>
23                         <Menu mnemonicParsing="false" text="File">
24                             <items>
25                                 <MenuItem mnemonicParsing="false" onAction="#HomePageMenuItem"
text="Home Page" />
26                                 <MenuItem mnemonicParsing="false" onAction="#LogOutMenuItem"
text="Log Out" />
27                                 <MenuItem mnemonicParsing="false" onAction="#ExitMenuItem"
text="Exit" />
28                             </items>
29                         </Menu>
30                         <Menu mnemonicParsing="false" text="View">
31                             <items>
32                                 <MenuItem mnemonicParsing="false"
onAction="#NewPatientEntryMenuItem" text="New Patient" />
33                                 <MenuItem mnemonicParsing="false"
onAction="#ShowPatientRecordMenuItem" text="Show Patient
Record" />
34                                 <MenuItem mnemonicParsing="false"
onAction="#MedicineRecordsMenuItem" text="Medicine Records" />
35                             </items>
36                         </Menu>
37                         <Menu mnemonicParsing="false" text="Help">
38                             <items>
39                                 <MenuItem mnemonicParsing="false"
onAction="#CollegeWebsiteLinksMenuItem" text="College Website
Links" />
40                                 <MenuItem mnemonicParsing="false" onAction="#AboutMenuItem"
text="About" />
41                             </items>
42                         </Menu>
43                     </menus>
44                 </MenuBar>
45                 <ImageView fitHeight="250.0" fitWidth="800.0" layoutY="26.0"
pickOnBounds="true" preserveRatio="true">
46                     <image>
47                         <Image url="@../Images/HomePageHeader.png" />
48                     </image>
49                 </ImageView>

```

```

50         </children>
51     </AnchorPane>
52 </top>
53 <left>
54     <AnchorPane prefHeight="350.0" prefWidth="800.0" BorderPane.alignment="CENTER">
55         <children>
56             <Button fx:id="HomePage" layoutX="599.0" layoutY="337.0"
57                 mnemonicParsing="false" onAction="#HomePage" prefHeight="26.0" prefWidth="74.0"
58                 text="Home">
59                 <font>
60                     <Font name="Roboto Condensed Bold" size="18.0" />
61                 </font>
62             </Button>
63             <Button fx:id="Logout" layoutX="687.0" layoutY="337.0" mnemonicParsing="false"
64                 onAction="#userLogout" prefHeight="26.0" prefWidth="74.0" text="Logout">
65                 <font>
66                     <Font name="Roboto Condensed Bold" size="18.0" />
67                 </font>
68             </Button>
69             <Label alignment="TOP_LEFT" layoutX="112.0" layoutY="53.0" prefHeight="88.0"
70                 prefWidth="604.0" text="Infirmary Management System is a Desktop Application
71                 developed using JavaFx, Java, MySQL and FXML for UI Design. This application is
72                 developed to run the College Infirmary smoothly with the use of Technology in
73                 this Modern Era. The Application helps to digitalize the current system of
74                 maintaining Patient and Medicine records in Register." wrapText="true">
75                 <font>
76                     <Font name="Roboto Condensed Regular" size="15.0" />
77                 </font>
78             </Label>
79             <Label layoutX="363.0" prefHeight="31.0" prefWidth="73.0" text="About">
80                 <font>
81                     <Font name="Cambria Bold" size="25.0" />
82                 </font>
83             </Label>
84             <Separator layoutX="195.0" layoutY="38.0" prefHeight="0.0" prefWidth="409.0" />
85             <Label alignment="TOP_LEFT" layoutX="112.0" layoutY="149.0" prefHeight="30.0"
86                 prefWidth="604.0" text="The Main Features of the Application are:" wrapText="true">
87                 <font>
88                     <Font name="Roboto Condensed Regular" size="15.0" />
89                 </font>
90             </Label>
91             <Label alignment="TOP_LEFT" layoutX="130.0" layoutY="173.0" prefHeight="22.0"
92                 prefWidth="346.0" text="1. Enter Patent Details through a User Interactive Form"
93                 wrapText="true">
94                 <font>
95                     <Font name="Roboto Condensed Regular" size="15.0" />
96                 </font>
97             </Label>
98             <Label alignment="TOP_LEFT" layoutX="130.0" layoutY="195.0" prefHeight="22.0"
99                 prefWidth="605.0" text="2. View, Sort, Search By (Filter), Resize Columns, Edit
100                 Column Positions in Data Display in Table " wrapText="true">
101                 <font>
102                     <Font name="Roboto Condensed Regular" size="15.0" />
103                 </font>
104             </Label>
105             <Label alignment="TOP_LEFT" layoutX="130.0" layoutY="218.0" prefHeight="22.0"
106                 prefWidth="282.0" text="3. Insert and Update New Medicine & Injections "
107                 wrapText="true">
108                 <font>
109                     <Font name="Roboto Condensed Regular" size="15.0" />
110                 </font>
111             </Label>
112             <Label alignment="TOP_LEFT" layoutX="130.0" layoutY="240.0" prefHeight="22.0"
113                 prefWidth="410.0" text="4. Subtract (Update) Medicines & Injections From Stock

```

```

98         if Prescribed" wrapText="true">
99             <font>
100                 <Font name="Roboto Condensed Regular" size="15.0" />
101             </font>
102         </Label>
103         <Label alignment="TOP_LEFT" layoutX="130.0" layoutY="262.0" prefHeight="22.0"
104         prefWidth="437.0" text="5. Show only Available Medicines in Prescribe Combo Box"
105         wrapText="true">
106             <font>
107                 <Font name="Roboto Condensed Regular" size="15.0" />
108             </font>
109         </Label>
110         <Label alignment="TOP_LEFT" layoutX="130.0" layoutY="284.0" prefHeight="22.0"
111         prefWidth="437.0" text="6. Get Automatic Date & Time while Filling up New
112         Patient Entry Form " wrapText="true">
113             <font>
114                 <Font name="Roboto Condensed Regular" size="15.0" />
115             </font>
116         </Label>
117         </children>
118     </AnchorPane>
119 </left>
120 </BorderPane>
121

```

```
1  package application;
2
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXML;
5  import javafx.scene.control.Alert;
6  import javafx.scene.control.Button;
7  import javafx.scene.control.ButtonType;
8  import javafx.scene.control.Hyperlink;
9
10 import java.awt.Desktop;
11 import java.io.IOException;
12 import java.net.URISyntaxException;
13 import java.net.URL;
14
15 public class CollegeWebsiteLinksController {
16
17     @FXML
18     private Button HomePage;
19
20     @FXML
21     private Button Logout;
22
23     @FXML
24     private Hyperlink cstWebsite;
25
26     @FXML
27     private Hyperlink rubIMS;
28
29     @FXML
30     private Hyperlink mail;
31
32     @FXML
33     private Hyperlink vle;
34
35     @FXML
36     private Hyperlink library;
37
38     @FXML
39     private Hyperlink mrbs;
40
41     //back to login page
42     public void userLogout(ActionEvent event) throws IOException {
43
44         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
45         alert.setTitle("Logout");
46         alert.setHeaderText(null);
47         alert.setContentText("Are you sure you want to Logout?");
48         if (alert.showAndWait().get() == ButtonType.OK){
49             Main m = new Main();
50             m.changeScene("LoginPage.fxml");
51         }
52     }
53
54     public void HomePage(ActionEvent event) throws IOException {
55
56         Main m = new Main();
57         m.changeScene("HomePage.fxml");
58     }
59
60 }
61
62 //calling openLinkInBrowser method and passing links as String
63 public void cstWebsite(ActionEvent event) throws IOException {
64     openLinkInBrowser("https://cst.edu.bt/index.php/en/");
```

```

65     }
66
67     public void rubIMS(ActionEvent event) throws IOException {
68         openLinkInBrowser("https://ims.rub.edu.bt/");
69     }
70
71     public void mail(ActionEvent event) throws IOException {
72         openLinkInBrowser("http://mail.rub.edu.bt/");
73     }
74
75     public void vle(ActionEvent event) throws IOException {
76         openLinkInBrowser("https://vle.cst.edu.bt/");
77     }
78
79     public void library(ActionEvent event) throws IOException {
80         openLinkInBrowser("https://lib.cst.edu.bt/");
81     }
82
83     public void mrbs(ActionEvent event) throws IOException {
84         openLinkInBrowser("https://vle.cst.edu.bt/mrbs/web/day.php");
85     }
86
87     //method to open link in browser
88     public void openLinkInBrowser(String str) throws IOException{
89         try {
90             Desktop.getDesktop().browse(new URL(str).toURI());
91         }
92         catch (URISyntaxException ae) {
93
94             ae.printStackTrace();
95
96             Alert alert = new Alert(Alert.AlertType.ERROR);
97             alert.setTitle("Error Message");
98             alert.setHeaderText(null);
99             alert.setContentText("ERROR Opening URL");
100            alert.showAndWait();
101        }
102    }
103
104    //MenuBar
105
106    //Menu 1 - File
107
108    //Menu Item 1.1
109    public void HomePageMenuItem(ActionEvent event) {
110        MenuBar obj = new MenuBar();
111        obj.HomePage();
112    }
113
114    //Menu Item 1.2
115    public void LogoutMenuItem(ActionEvent event) {
116        MenuBar obj = new MenuBar();
117        obj.LogOut();
118    }
119
120    //Menu Item 1.3
121    public void ExitMenuItem(ActionEvent event) {
122        MenuBar obj = new MenuBar();
123        obj.Exit();
124    }
125
126    //Menu 2 - View
127
128    //Menu Item 2.1

```

```
129     public void NewPatientEntryMenuItem(ActionEvent event) {
130         MenuBar obj = new MenuBar();
131         obj.NewPatientEntry();
132     }
133
134     //Menu Item 2.2
135     public void ShowPatientRecordMenuItem(ActionEvent event) {
136         MenuBar obj = new MenuBar();
137         obj.ShowPatientRecord();
138     }
139
140     //Menu Item 2.3
141     public void MedicineRecordsMenuItem(ActionEvent event) {
142         MenuBar obj = new MenuBar();
143         obj.MedicineRecords();
144     }
145
146     //Menu 3 - Help
147
148     //Menu Item 3.1
149     public void CollegeWebsiteLinksMenuItem(ActionEvent event) {
150         MenuBar obj = new MenuBar();
151         obj.CollegeWebsiteLinks();
152     }
153
154     //Menu Item 3.2
155     public void AboutMenuItem(ActionEvent event) {
156         MenuBar obj = new MenuBar();
157         obj.About();
158     }
159
160 }
161
```