

```

1 package assignment8;
2
3 import java.sql.*;
4 import javafx.application.Application;
5 import static javafx.application.Application.launch;
6 import javafx.geometry.*;
7 import javafx.scene.Scene;
8 import javafx.scene.control.*;
9 import javafx.scene.chart.*;
10 import javafx.scene.layout.*;
11 import javafx.scene.image.*;
12 import javafx.stage.*;
13 import oracle.jdbc.pool.OracleDataSource;
14
15 /**
16  * Assignment 8
17  * Yassine Elmellouki 1110964137
18  * This assignment abides by the Honor Code
19  */
20
21 //This class creates a GUI that displays DB data
22 public class Assignment8 extends Application {
23     Connection con;
24     Statement stmt;
25     Statement sSalary;
26     Statement sUpdate;
27     ResultSet rsS;
28     ResultSet rsE;
29
30     ComboBox<String> Combo = new ComboBox<>();
31     ComboBox<String> Combo1 = new ComboBox<>();
32     TextField eTxt = new TextField();
33     TextField fTxt = new TextField();
34     TextField lTxt = new TextField();
35     TextField sTxt = new TextField();
36     Button update = new Button("Update Employee");
37     Button previous = new Button("Previous");
38     Button next = new Button("Next");
39     Button chart = new Button("Show Chart");
40     Label eLabel = new Label("Emp ID:");
41     Label nLabel = new Label("First Name:");
42     Label lnLabel = new Label("Last Name:");
43     Label sLabel = new Label("Salary:");
44     Label tLabel = new Label("Job Title:");
45     Label dLabel = new Label("Department:");
46
47     Image img = new Image("save.gif");
48     Image img1 = new Image("left.gif");
49     Image img2 = new Image("right.gif");
50     Image img3 = new Image("chart.jpg");
51     ImageView v1 = new ImageView(img);
52     ImageView v2 = new ImageView(img1);
53     ImageView v3 = new ImageView(img2);
54     ImageView v4 = new ImageView(img3);
55
56     @Override
57     public void start(Stage primaryStage) {
58         try {
59             //create pane with buttons, labels and images
60             GridPane pane = new GridPane();
61             pane.setAlignment(Pos.CENTER);
62             pane.setPadding(new Insets(11.5, 12.5, 13.5, 14.5));
63             pane.setHgap(5.5);
64             pane.setVgap(5.5);
65             pane.add(eLabel, 0, 0);
66             pane.add(eTxt, 1, 0);
67             pane.add(nLabel, 0, 1);
68             pane.add(fTxt, 1, 1);
69             pane.add(lnLabel, 0, 2);
70             pane.add(lTxt, 1, 2);
71             pane.add(sLabel, 0, 3);
72             pane.add(sTxt, 1, 3);
73             pane.add(tLabel, 0, 4);
74             pane.add(Combo, 1, 4);
75             pane.add(dLabel, 0, 5);
76             pane.add(Combo1, 1, 5);
77             pane.add(update, 0, 6);
78             pane.add(previous, 0, 7);
79             pane.add(next, 1, 7);
80             pane.add(chart, 3, 7);
81             update.setGraphic(v1);
82             previous.setGraphic(v2);
83             next.setGraphic(v3);
84             chart.setGraphic(v4);
85             GridPane.setHalignment(update, HPos.LEFT);
86             //create a scene in a stage
87             Scene sc = new Scene(pane);
88             primaryStage.setTitle("Employee Database GUI");
89             primaryStage.setScene(sc);
90             primaryStage.show();
91             //connecting to the DB
92             String connectionString = "jdbc:oracle:thin:@localhost:1521:XE";
93             OracleDataSource ds = new OracleDataSource();
94             ds.setURL(connectionString);
95             con = ds.getConnection("emp", "emp");
96             stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
97             ResultSet rsT = stmt.executeQuery("select distinct title from employees");
98             while (rsT.next()) {
99                 Combo.getItems().add(rsT.getString(1));
100             }
101             ResultSet rsD = stmt.executeQuery("select departmentid, departmentname from departments");

```

```

101 Resultset rsD = stmt.executeQuery("select departmentid, departmentname from departments");
102 while (rsD.next()) {
103     Combo1.getItems().add(rsD.getString(2));
104 }
105 String query = "select employeeid, firstname, lastname, salary, title, "
106 + "departments.departmentname from employees left join "
107 + "departments on employees.departmentid=departments.departmentid "
108 + "order by lastname, firstname";
109 rsE = stmt.executeQuery(query);
110 rsE.next();
111 eTxt.setText(rsE.getString(1));
112 fTxt.setText(rsE.getString(2));
113 lTxt.setText(rsE.getString(3));
114 sTxt.setText(rsE.getString(4));
115 Combo.setValue(rsE.getString(5));
116 Combo1.setValue(rsE.getString(6));
117 update.setOnAction(e -> empUpdate());
118 next.setOnAction(e -> empName());
119 previous.setOnAction(e -> prev());
120 chart.setOnAction(e -> empSalary());
121 } catch (SQLException e) {
122     System.out.println("Error" + e);
123 }
124 }
125 void empName() {
126     try {
127         rsE.next();
128         eTxt.setText(rsE.getString(1));
129         fTxt.setText(rsE.getString(2));
130         lTxt.setText(rsE.getString(3));
131         sTxt.setText(rsE.getString(4));
132         Combo.setValue(rsE.getString(5));
133         Combo1.setValue(rsE.getString(6));
134     }
135     catch (SQLException e) {
136         System.out.println("Error " + e);
137     }
138 }
139 void empUpdate() {
140     try {
141         sUpdate = con.createStatement();
142         String jUpdate = Combo.getValue();
143         String dpUpdate = Combo1.getValue();
144         int a = Combo1.getItems().indexOf(dpUpdate) + 1;
145         String lnUpdate = lTxt.getText();
146         String slUpdate = sTxt.getText();
147         String eUpdate = eTxt.getText();
148         String fnUpdate = fTxt.getText();
149         String query = "Update employees set title = '" + jUpdate + "', departmentid = '"
150 + a + "', firstname = '" + fnUpdate + "', lastname = '" + lnUpdate
151 + "', salary = '" + slUpdate + "' where employeeid = '" + eUpdate + "'";
152         sUpdate.executeUpdate(query);
153         String q2 = "select employeeid, firstname, lastname, salary, title, "
154 + "departments.departmentname from employees left join "
155 + "departments on employees.departmentid=departments.departmentid "
156 + "order by lastname, firstname";
157         rsE = stmt.executeQuery(q2);
158     }
159     catch (SQLException e) {
160         System.out.println("Error " + e);
161     }
162 }
163 void empSalary() {
164     Stage stage = new Stage();
165     stage.setTitle("Bar Chart");
166     final CategoryAxis xAxis = new CategoryAxis();
167     final NumberAxis yAxis = new NumberAxis();
168     final BarChart<String, Number> bc
169         = new BarChart<String, Number>(xAxis, yAxis);
170     XYChart.Series max = new XYChart.Series();
171     XYChart.Series avg = new XYChart.Series();
172     XYChart.Series min = new XYChart.Series();
173     bc.setTitle("Salaries by Department");
174     xAxis.setLabel("Department");
175     yAxis.setLabel("Salary");
176     max.setName("Maximum");
177     avg.setName("Average");
178     min.setName("Minimum");
179
180     try {
181         sSalary = con.createStatement();
182         String query = "select max(salary) as Maximum, "
183 + "avg(salary) as Average, min(salary) as Minimum, "
184 + " departments.departmentname from employees "
185 + "left join departments on employees.departmentid=departments.departmentid group by"
186 + " departments.departmentname order by departments.departmentname";
187         rsS = sSalary.executeQuery(query);
188
189         while (rsS.next()) {
190             String dpt = rsS.getString(4);
191             if (dpt == null) {
192                 dpt = "-";
193             }
194             max.getData().add(new XYChart.Data(dpt, rsS.getInt(1)));
195             avg.getData().add(new XYChart.Data(dpt, rsS.getInt(2)));
196             min.getData().add(new XYChart.Data(dpt, rsS.getInt(3)));
197         }
198     } catch (SQLException e) {
199         System.out.println("Error " + e);
200     }
201     Scene scene = new Scene(bc, 800, 600);
202     bc.getData().addAll(max, avg, min);
203     stage.setScene(scene);
204     stage.show();
205 }
206 void prev() {
207     try {
208         rsE.previous();
209         eTxt.setText(rsE.getString(1));
210         fTxt.setText(rsE.getString(2));
211         lTxt.setText(rsE.getString(3));
212         sTxt.setText(rsE.getString(4));
213         Combo.setValue(rsE.getString(5));
214         Combo1.setValue(rsE.getString(6));
215     }
216     catch (SQLException e) {
217         System.out.println("Error " + e);
218     }
219 }

```

```
200     }
201     Scene scene = new Scene(bc, 800, 600);
202     bc.getData().addAll(max, avg, min);
203     stage.setScene(scene);
204     stage.show();
205 }
206 void prev() {
207     try {
208         rsE.previous();
209         eTxt.setText(rsE.getString(1));
210         fTxt.setText(rsE.getString(2));
211         lTxt.setText(rsE.getString(3));
212         sTxt.setText(rsE.getString(4));
213         Combo.setValue(rsE.getString(5));
214         Combo1.setValue(rsE.getString(6));
215     }
216     catch (SQLException e) {
217         System.out.println("Error " + e);
218     }
219 }
220 public static void main(String[] args) {
221     launch(args);
222 }
223 }
```