Assignment 10

Student Name: Omar Sebri
Student ID: 3722350

Part A:

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DisplayMenu{
    public static void main(String[] args) {
        Connection connector = openConnection();
        if (connector == null)
        {   System.err.println("Unable to connect to the database");
            System.exit(1);
        }
        PreparedStatement menuLisStatement =
prepareMenuLisStatement(connector);
        displayMenuItems(menuLisStatement);
        closeConnection(connector);
    }
    /* a method that establishes connection */
    private static Connection openConnection(){
        final String url = "jdbc:mysql://localhost:3306/donut shop";
        final String user = "root";
        final String password = "zuizui007";
        Connection conn = null;
        try{
            conn = DriverManager.getConnection(url, user, password);
        }
        catch(Exception e){
            System.out.println(e.getMessage());
        }
        return conn;
    }
    /* closes connection with the database*/
    private static void closeConnection(Connection connect){
        try{
            connect.close();
        }
        catch(Exception e){
            System.out.println(e.getMessage());
        }
    }
```

```java
    /** prepares the sql statement used to retrieve the informations from the
menu  */
    public static PreparedStatement prepareMenuLisStatement(Connection
connect){
        PreparedStatement result = null;
        try{
            String query = "SELECT Menu_types, Menu_id, item_name, item_price
FROM "+
            "menu_items left OUTER JOIN menu_categories "+
            "ON menu_items.Item_category=menu_categories.Menu_ID "+
            "ORDER BY menu_id, item_id;";
            result=connect.prepareStatement(query);
        }
        catch(SQLException e){
            System.out.println(e.getMessage());
        }
        return result;
    }


    private static void displayMenuItems(PreparedStatement listMenuStatement)
{
        int last = 0; /**keeps track of the last menu_id retrieved */
        int temp =0;  /** keeps track of the current menu_id */
        try{
        ResultSet result = listMenuStatement.executeQuery();
        System.out.println("====================== DONUT SHOP MENU
======================");
        /** whenever the menu_id changes, the new menu gets menu type gets
printed */
        while(result.next()){
            temp=result.getInt(2);
            if(temp!=last){
                System.out.println(" == "+result.getString(1)+" ==");
                last=temp;
            }
                System.out.printf ("%-55s%.2f\n",
                                    result.getString(3),
                                    result.getDouble(4));

                    last=result.getInt(2);
        }
        System.out.println("==================================================
=============");

        }
        catch(SQLException e){
            System.out.println(e.getMessage());
        }
```

```
    }
}
```

Output:

```
======================= DONUT SHOP MENU =======================
 == Hot Beverage ==
Coffee (Original Blend, Dark Roast or Decaf) Small     2,01
Coffee (Original Blend, Dark Roast or Decaf) Medium    2,11
Coffee (Original Blend, Dark Roast or Decaf) Large     2,19
Tea                                                    1,68
 == Cold Beverage ==
Iced Coffee (Original or Dark Roast) Small             1,99
Iced Coffee (Original or Dark Roast) Medium            2,38
Iced Coffee (Original or Dark Roast) Large             3,02
 == Handcrafted Sandwich ==
California Turkey                                      7,15
Turkey Bacon Club                                      6,49
Avocado BLT                                            5,49
===============================================================
```

Part B:

Code:

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.*;
import javafx.geometry.*;
import javafx.event.ActionEvent;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

import java.sql.CallableStatement;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class AdjustPrice extends Application{
    private Label itemLabel;
    private TextField item;
    private Button adjust;
    private Label status;
    public void start(Stage primaryStage){
        primaryStage.setTitle("Adjust the price of a menu item");
        item = new TextField();
        item.setPrefWidth(50);
```

```java
        itemLabel = new Label("Menu Item Id:");
        adjust = new Button("Adjust price");
        status = new Label("Enter the menu item id");
        adjust.setOnAction(this::eventHandler);

        GridPane mainPane = new GridPane();
        mainPane.setAlignment(Pos.CENTER);
        mainPane.add(itemLabel,0,0,1,1);
        mainPane.add(item,1,0,1,1);
        mainPane.add(adjust,0,1,1,1);
        mainPane.add(status,0,2,1,1);
        mainPane.setHgap(12);
        mainPane.setVgap(12);
        Scene scene = new Scene(mainPane, 600, 250);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public void eventHandler(ActionEvent event){
        int item_id;
        String query = null;
        String message=null;
        int newp;
        try{
            item_id= Integer.parseInt(item.getText());
        }
        catch(Exception e){
            System.out.println(e.getMessage());
            return;
        }
        try{
            Connection connector = DriverManager.getConnection
                   ("jdbc:mysql://localhost:3306/donut shop",
                    "root",
                    "zuizui007");
                    /** callable statement that execute the price change */
            String call = "call adjustPrice(?);";
            CallableStatement procedureCall = connector.prepareCall(call);
            procedureCall.setInt(1, item_id);
            int affectedRows = procedureCall.executeUpdate();

            if (affectedRows == 0)
                status.setText("Adjustment failed: non existant item");

            else
                query = "select item_name, item_price from menu_items "+
                                        "where item_id=? ;";
            /** a prepared helps retrive the updated price and the item from
the table */
```

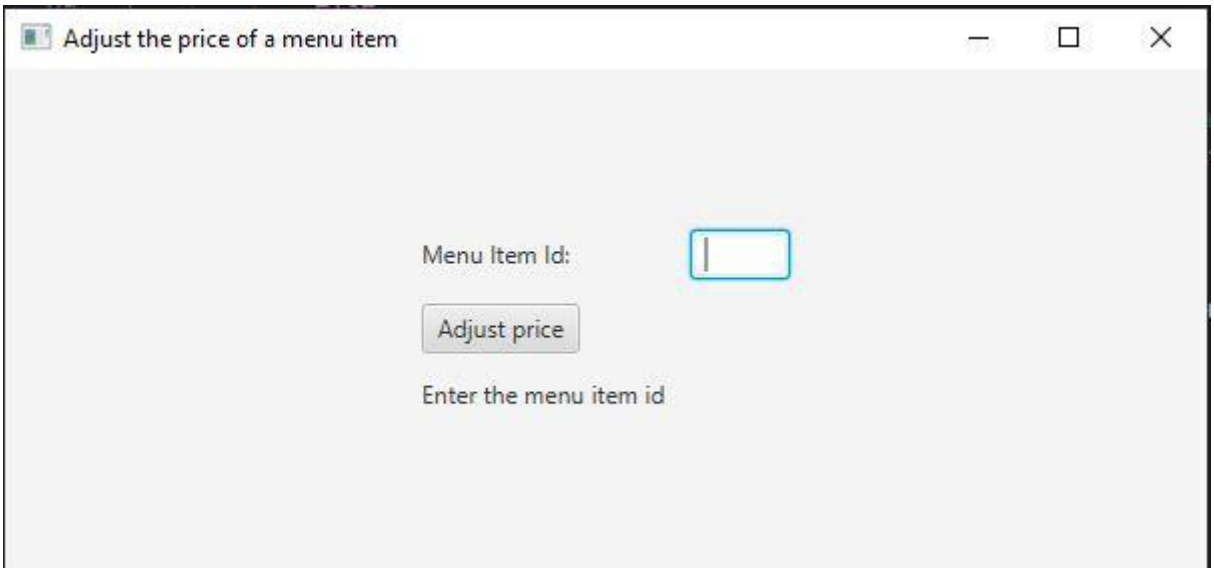
```
            PreparedStatement preparePrice =
connector.prepareStatement(query);
            preparePrice.setInt(1, item_id);
            ResultSet result = preparePrice.executeQuery();
            result.next();
            message = "The price for "+result.getString(1)+" is now "+
            result.getDouble(2)+"$" ;
            status.setText(message);

            connector.close();

        }
        catch(SQLException e){
            System.out.println(e.getMessage());
        }
    }
    public static void main(String[] args) {
        launch(args);
    }

}
```

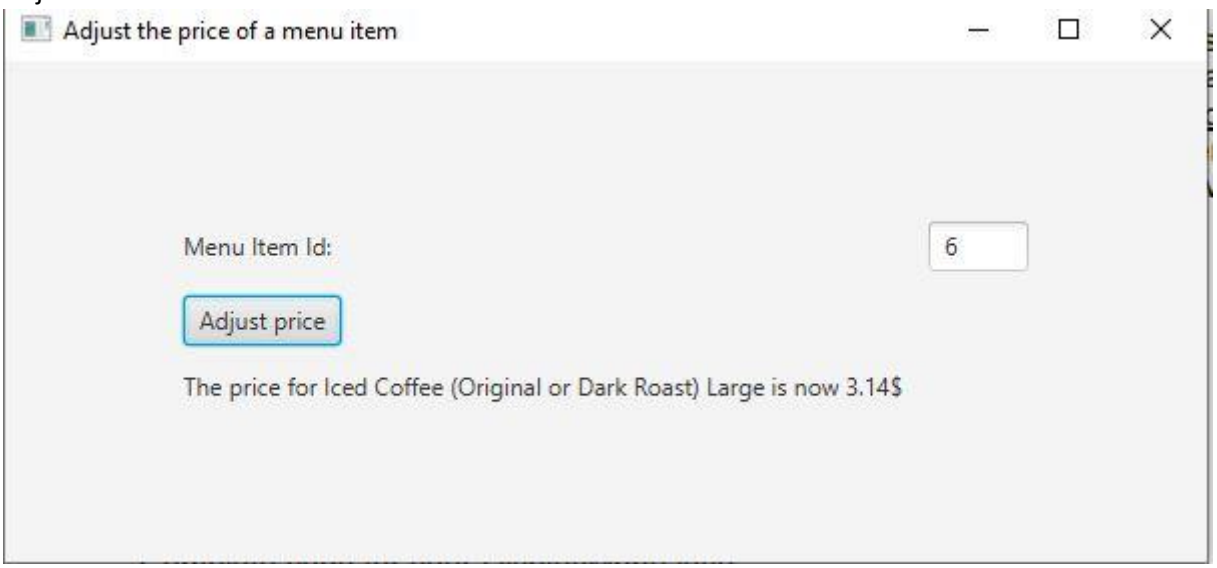
Output:




Menu item before adjusting:

| item_id | item_name | item_price |
|---|---|---|
| 6 | Iced Coffee (Original or Dark Roast) Large | 3,02 |

Adjustment:




Attempting: