Relational Databases with MySQL Week 10 Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push an .sql file with all your queries and your Java project code to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

In this week's coding activity, you will create a menu driven application backed by a MySQL database.

To start, choose one item that you like. It could be vehicles, sports, foods, etc....

Create a new Java project in Eclipse.

Create a SQL script in the project to create a database with one table. The table should be the item you picked.

Write a Java menu driven application that allows you to perform all four CRUD operations on your table.

Tips:

The application does not need to be as complex as the example in the video curriculum.

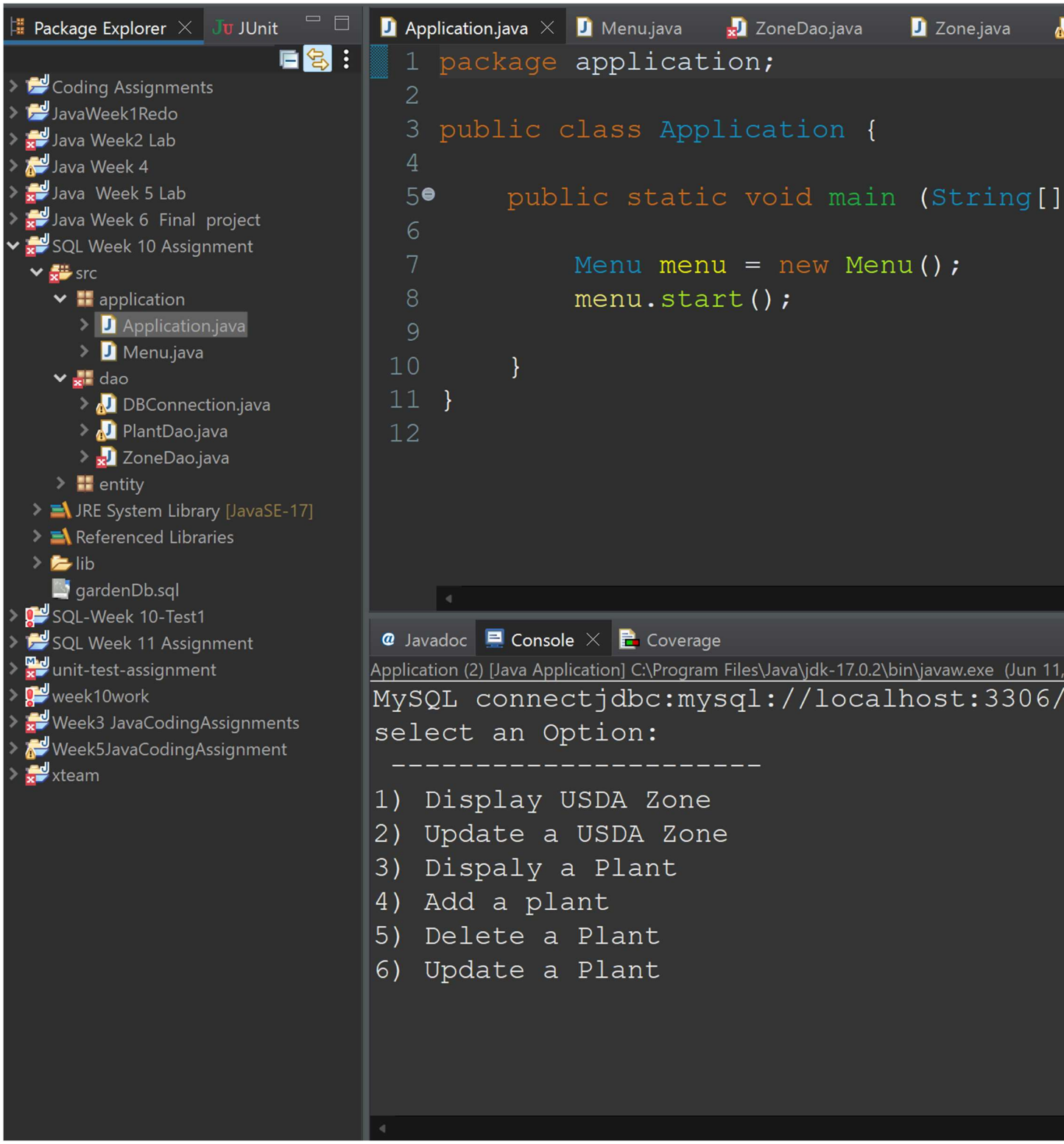
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

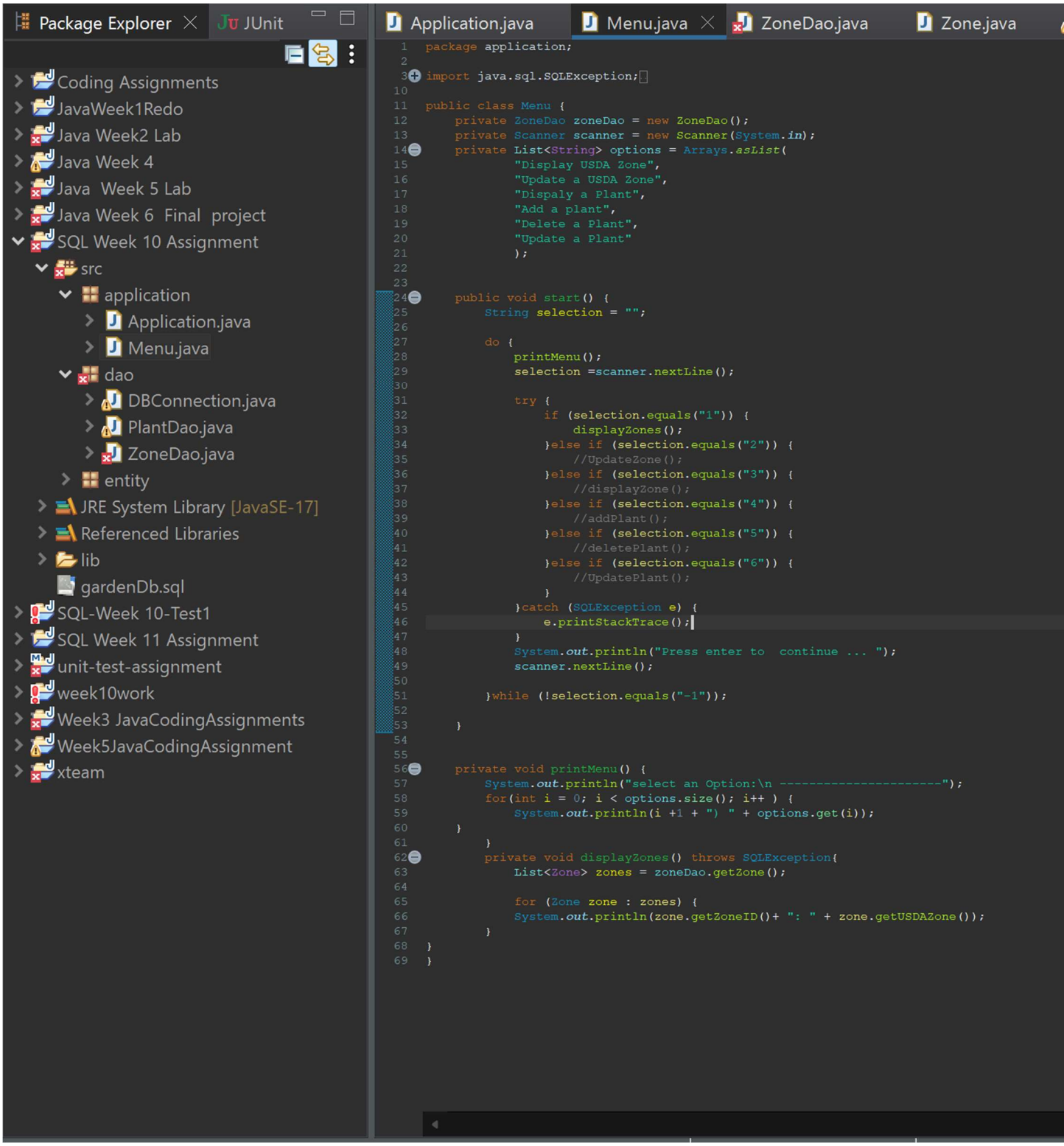
Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

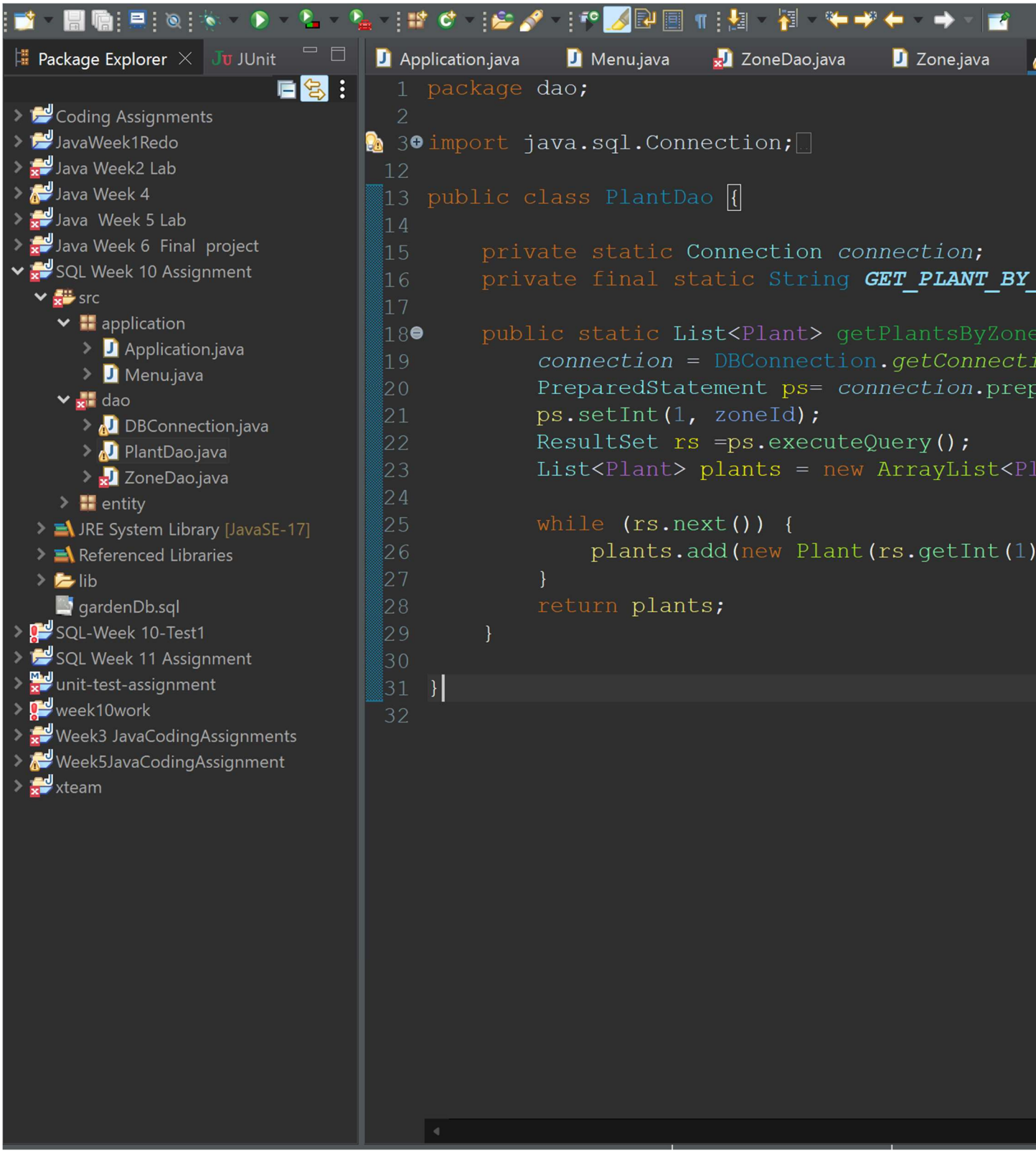
Remember that both parameters on PreparedStatements and the ResultSet columns are based on indexes that start with 1, not 0.

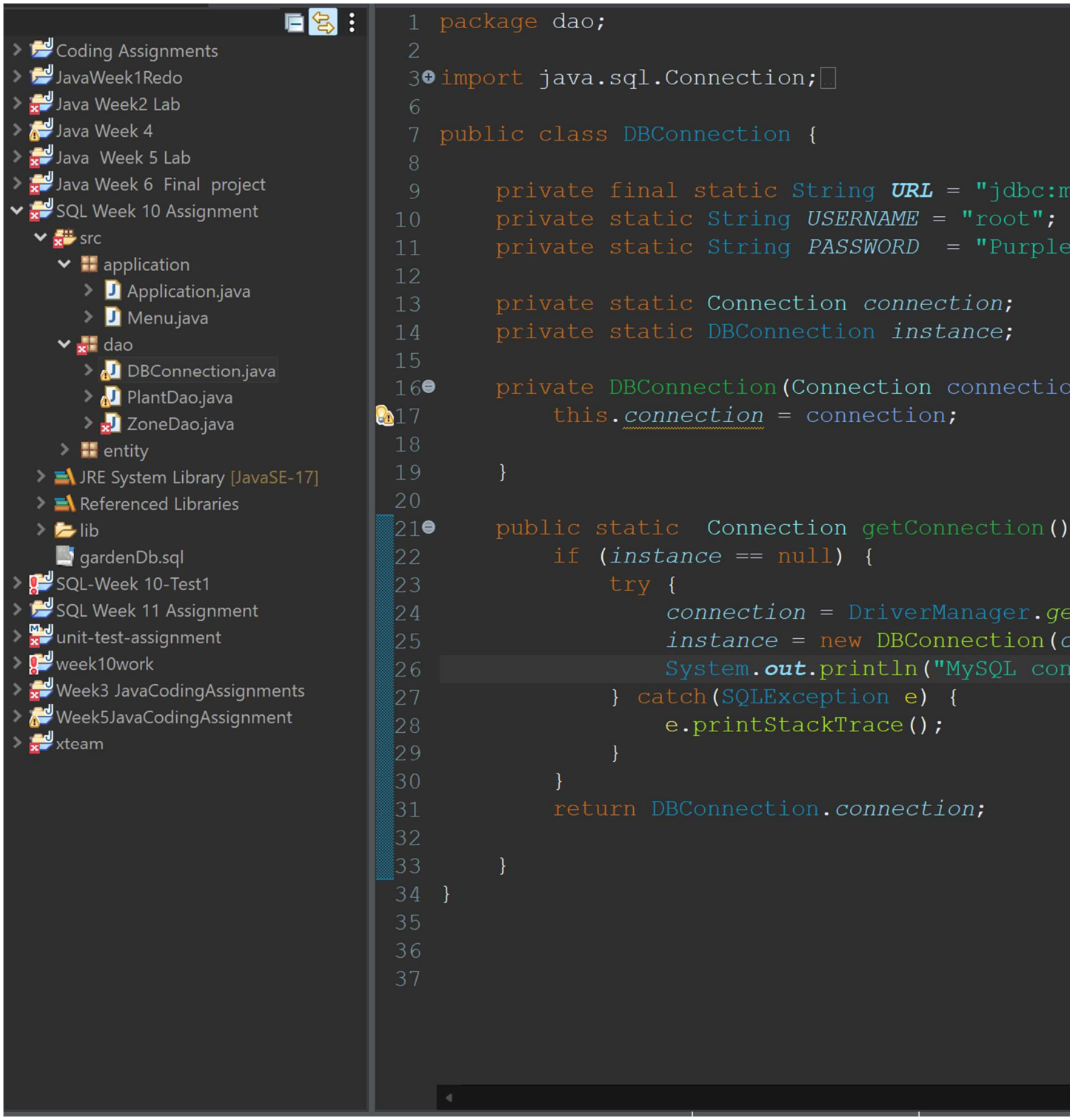
I Was not able to fix my code… and could not get past creating the menu.

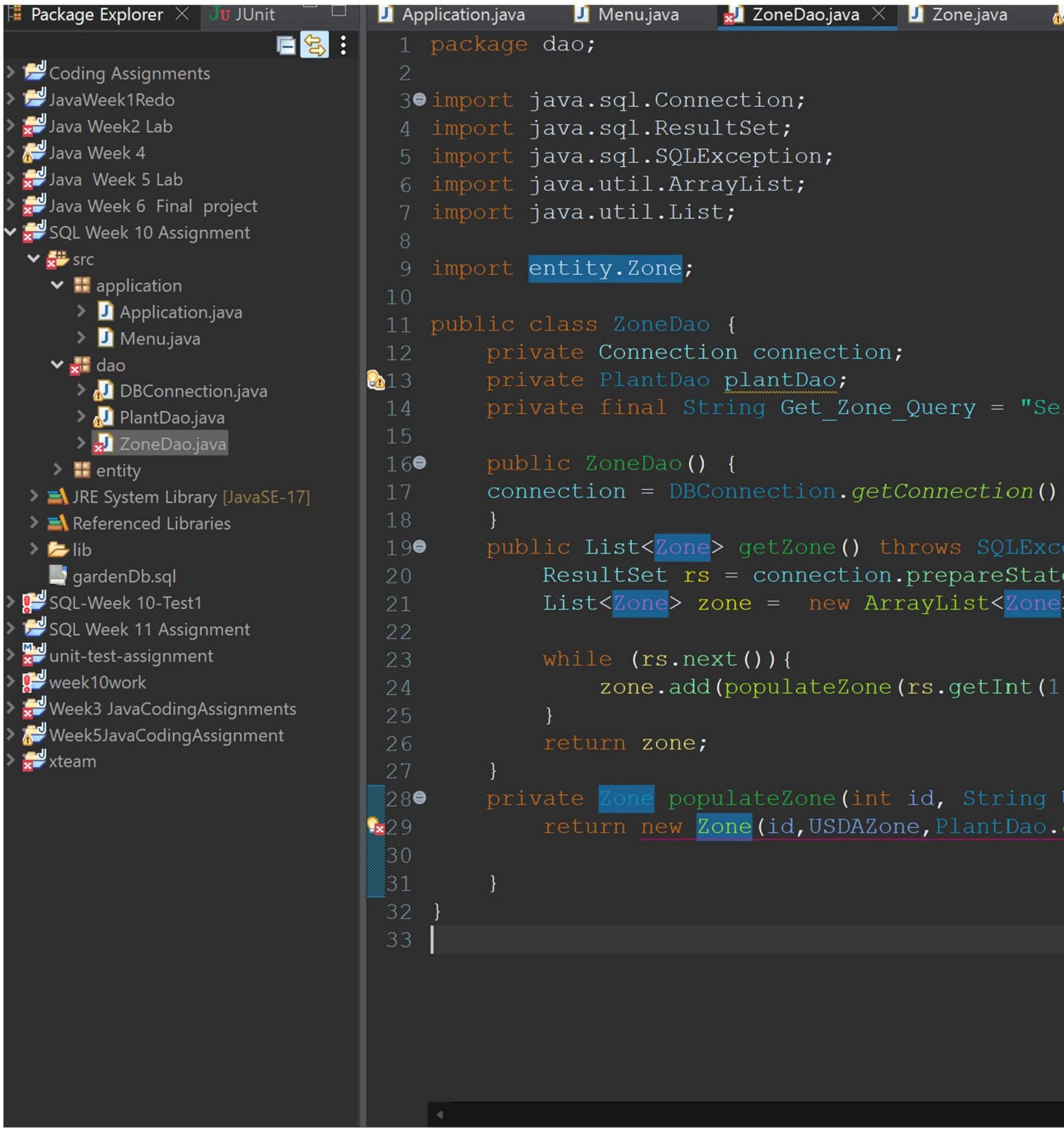
Screenshots of Code:


Package Explorer
JUnit
Coding Assignments
JavaWeek1Redo
Java Week2 Lab
Java Week 4
Java Week 5 Lab
Java Week 6 Final project
SQL Week 10 Assignment
src
application
Application.java
Menu.java
dao
DBConnection.java
PlantDao.java
ZoneDao.java
entity
JRE System Library [JavaSE-17]
Referenced Libraries
lib
gardenDb.sql
SQL-Week 10-Test1
SQL Week 11 Assignment
unit-test-assignment
week10work
Week3 JavaCodingAssignments
Week5JavaCodingAssignment
xteam
Application.java
Menu.java
ZoneDao.java
Zone.java
1 package application;
2
3 public class Application {
4
5 public static void main (String[]
6
7 Menu menu = new Menu();
8 menu.start();
9
10 }
11 }
12
Javadoc
Console
Coverage
Application (2) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Jun 11,
MySQL connectjdbc:mysql://localhost:3306/
select an Option:
1) Display USDA Zone
2) Update a USDA Zone
3) Dispaly a Plant
4) Add a plant
5) Delete a Plant
6) Update a Plant


Package Explorer
JUnit
Coding Assignments
JavaWeek1Redo
Java Week2 Lab
Java Week 4
Java Week 5 Lab
Java Week 6 Final project
SQL Week 10 Assignment
src
application
Application.java
Menu.java
dao
DBConnection.java
PlantDao.java
ZoneDao.java
entity
JRE System Library [JavaSE-17]
Referenced Libraries
lib
gardenDb.sql
SQL-Week 10-Test1
SQL Week 11 Assignment
unit-test-assignment
week10work
Week3 JavaCodingAssignments
Week5JavaCodingAssignment
xteam
Application.java
Menu.java
ZoneDao.java
Zone.java
1 package application;
2
3 import java.sql.SQLException;
10
11 public class Menu {
12 private ZoneDao zoneDao = new ZoneDao();
13 private Scanner scanner = new Scanner(System.in);
14 private List<String> options = Arrays.asList(
15 "Display USDA Zone",
16 "Update a USDA Zone",
17 "Dispaly a Plant",
18 "Add a plant",
19 "Delete a Plant",
20 "Update a Plant"
21);
22
23
24 public void start() {
25 String selection = "";
26
27 do {
28 printMenu();
29 selection =scanner.nextLine();
30
31 try {
32 if (selection.equals("1")) {
33 displayZones();
34 }else if (selection.equals("2")) {
35 //UpdateZone();
36 }else if (selection.equals("3")) {
37 //displayZone();
38 }else if (selection.equals("4")) {
39 //addPlant();
40 }else if (selection.equals("5")) {
41 //deletePlant();
42 }else if (selection.equals("6")) {
43 //UpdatePlant();
44 }
45 }catch (SQLException e) {
46 e.printStackTrace();
47 }
48 System.out.println("Press enter to continue ... ");
49 scanner.nextLine();
50
51 }while (!selection.equals("-1"));
52
53 }
54
55
56 private void printMenu() {
57 System.out.println("select an Option:\n ----------------------");
58 for(int i = 0; i < options.size(); i++) {
59 System.out.println(i +1 + ") " + options.get(i));
60 }
61 }
62 private void displayZones() throws SQLException{
63 List<Zone> zones = zoneDao.getZone();
64
65 for (Zone zone : zones) {
66 System.out.println(zone.getZoneID()+ ": " + zone.getUSDAZone());
67 }
68 }
69 }

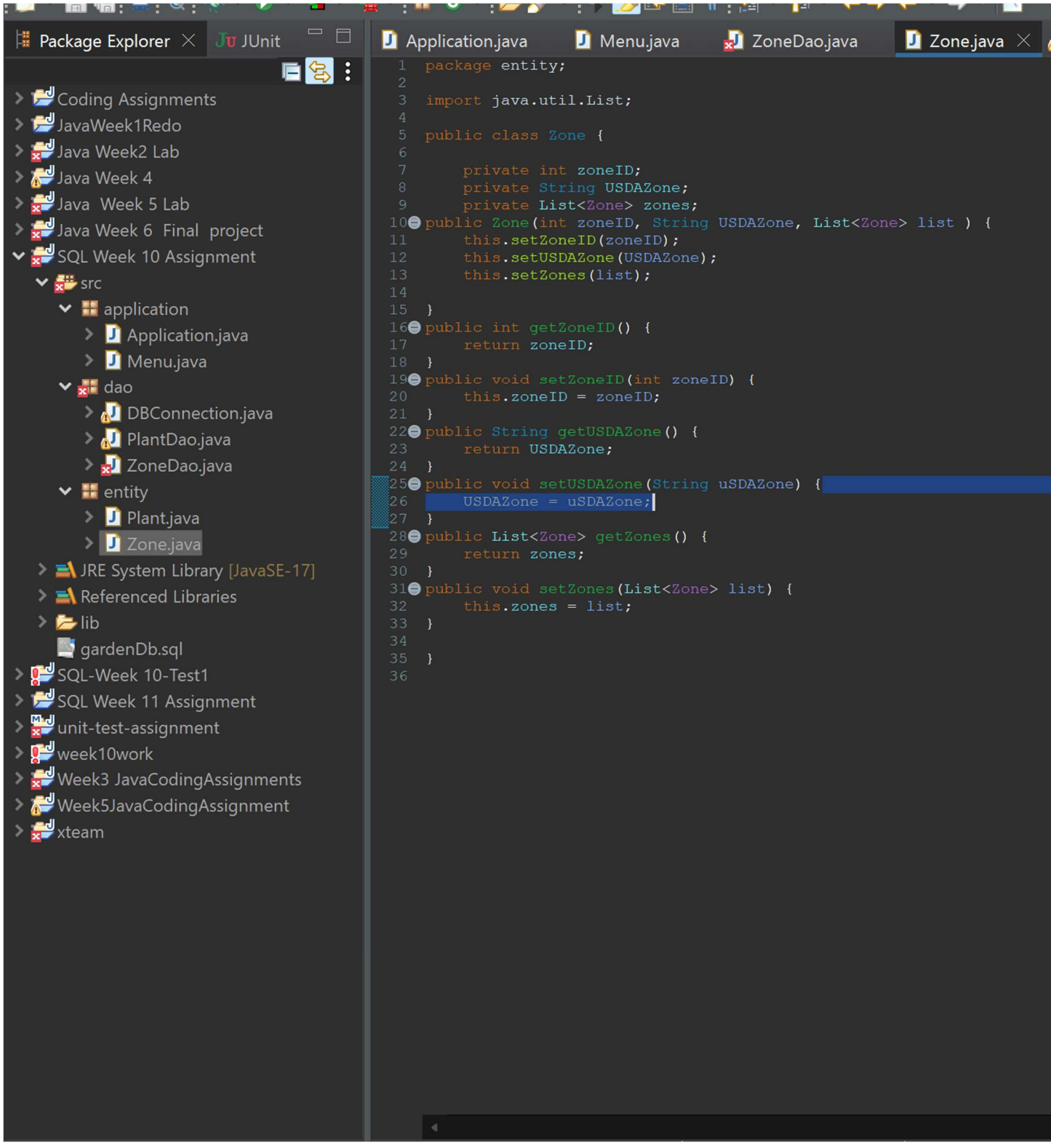

Package Explorer
JUnit
Coding Assignments
JavaWeek1Redo
Java Week2 Lab
Java Week 4
Java Week 5 Lab
Java Week 6 Final project
SQL Week 10 Assignment
src
application
Application.java
Menu.java
dao
DBConnection.java
PlantDao.java
ZoneDao.java
entity
JRE System Library [JavaSE-17]
Referenced Libraries
lib
gardenDb.sql
SQL-Week 10-Test1
SQL Week 11 Assignment
unit-test-assignment
week10work
Week3 JavaCodingAssignments
Week5JavaCodingAssignment
xteam
Application.java
Menu.java
ZoneDao.java
Zone.java
1 package dao;
2
3 import java.sql.Connection;
12
13 public class PlantDao {
14
15 private static Connection connection;
16 private final static String GET_PLANT_BY_
17
18 public static List<Plant> getPlantsByZone
19 connection = DBConnection.getConnecti
20 PreparedStatement ps= connection.prep
21 ps.setInt(1, zoneId);
22 ResultSet rs =ps.executeQuery();
23 List<Plant> plants = new ArrayList<Pl
24
25 while (rs.next()) {
26 plants.add(new Plant(rs.getInt(1)
27 }
28 return plants;
29 }
30
31 }
32


Coding Assignments
JavaWeek1Redo
Java Week2 Lab
Java Week 4
Java Week 5 Lab
Java Week 6 Final project
SQL Week 10 Assignment
src
application
Application.java
Menu.java
dao
DBConnection.java
PlantDao.java
ZoneDao.java
entity
JRE System Library [JavaSE-17]
Referenced Libraries
lib
gardenDb.sql
SQL-Week 10-Test1
SQL Week 11 Assignment
unit-test-assignment
week10work
Week3 JavaCodingAssignments
Week5JavaCodingAssignment
xteam
1 package dao;
2
3 import java.sql.Connection;
6
7 public class DBConnection {
8
9 private final static String URL = "jdbc:m
10 private static String USERNAME = "root";
11 private static String PASSWORD = "Purple
12
13 private static Connection connection;
14 private static DBConnection instance;
15
16 private DBConnection(Connection connectio
17 this.connection = connection;
18
19 }
20
21 public static Connection getConnection()
22 if (instance == null) {
23 try {
24 connection = DriverManager.ge
25 instance = new DBConnection(c
26 System.out.println("MySQL con
27 } catch(SQLException e) {
28 e.printStackTrace();
29 }
30 }
31 return DBConnection.connection;
32
33 }
34 }
35
36
37

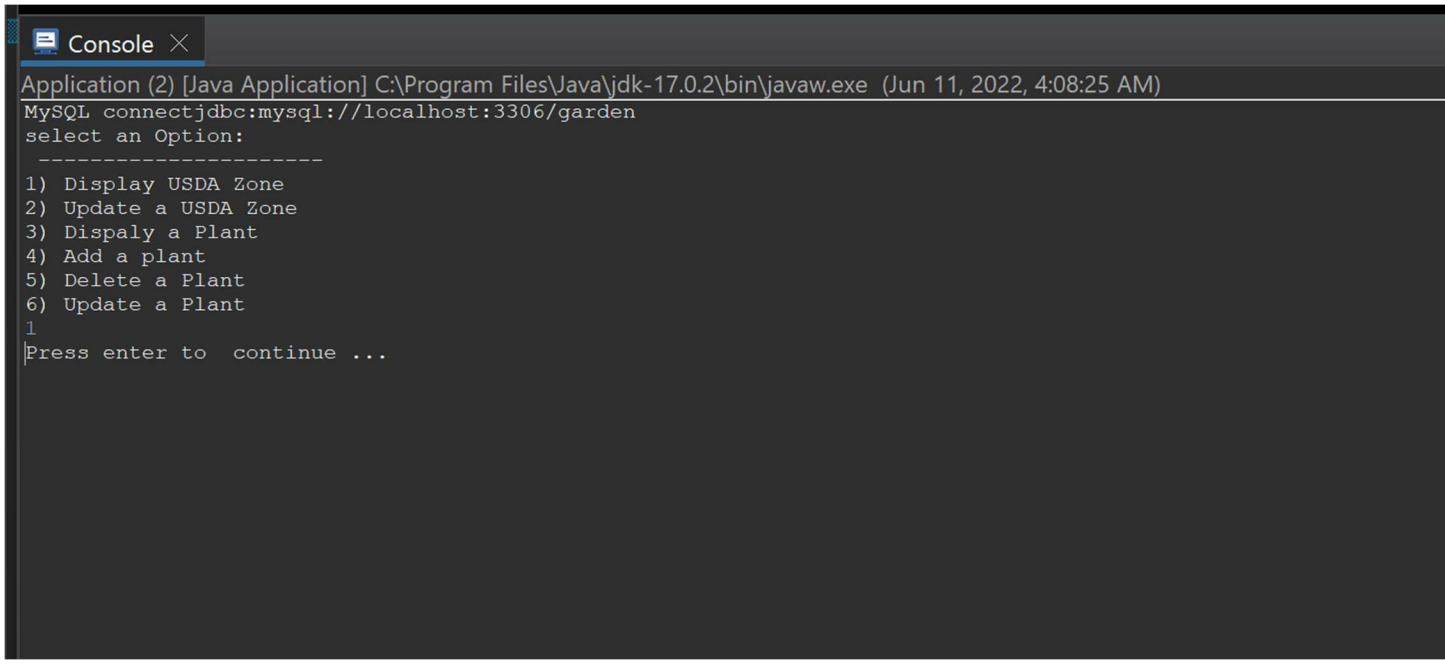

Package Explorer
JUnit
Coding Assignments
JavaWeek1Redo
Java Week2 Lab
Java Week 4
Java Week 5 Lab
Java Week 6 Final project
SQL Week 10 Assignment
src
application
Application.java
Menu.java
dao
DBConnection.java
PlantDao.java
ZoneDao.java
entity
JRE System Library [JavaSE-17]
Referenced Libraries
lib
gardenDb.sql
SQL-Week 10-Test1
SQL Week 11 Assignment
unit-test-assignment
week10work
Week3 JavaCodingAssignments
Week5JavaCodingAssignment
xteam
Application.java
Menu.java
ZoneDao.java
Zone.java
1 package dao;
2
3 import java.sql.Connection;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.util.ArrayList;
7 import java.util.List;
8
9 import entity.Zone;
10
11 public class ZoneDao {
12 private Connection connection;
13 private PlantDao plantDao;
14 private final String Get_Zone_Query = "Se
15
16 public ZoneDao() {
17 connection = DBConnection.getConnection()
18 }
19 public List<Zone> getZone() throws SQLExc
20 ResultSet rs = connection.prepareStat
21 List<Zone> zone = new ArrayList<Zone
22
23 while (rs.next()){
24 zone.add(populateZone(rs.getInt(1
25 }
26 return zone;
27 }
28 private Zone populateZone(int id, String
29 return new Zone(id,USDAZone,PlantDao.
30
31 }
32 }
33

Application.java | Menu.java | ZoneDao.java | Zone.java | PlantDao.java | DBConnection.java

```java
package entity;

public class Plant {

    private int plantId;
    private String botanicalName;
    private String commonName;
    private String plantFamily;


    public Plant( int id, String botanicalName, String commonName, int plantId, String plantFamily) {
        this.plantId = plantId;
        this.setBotanicalName(botanicalName);
        this.setCommonName(commonName);
        this.setPlantFamily(plantFamily);


        }


    public String getBotanicalName() {
        return botanicalName;
    }


    public void setBotanicalName(String botanicalName) {
        this.botanicalName = botanicalName;
    }


    public String getCommonName() {
        return commonName;
    }


    public void setCommonName(String commonName) {
        this.commonName = commonName;
    }

    public int  plantId() {
        return plantId;
}


    public void setplantId(int plantId) {
        this.plantId = plantId;
    }


    public String getPlantFamily() {
        return plantFamily;
    }


    public void setPlantFamily(String plantFamily) {
        this.plantFamily = plantFamily;
    }
}
```


Package Explorer
JUnit
Coding Assignments
JavaWeek1Redo
Java Week2 Lab
Java Week 4
Java Week 5 Lab
Java Week 6 Final project
SQL Week 10 Assignment
src
application
Application.java
Menu.java
dao
DBConnection.java
PlantDao.java
ZoneDao.java
entity
Plant.java
Zone.java
JRE System Library [JavaSE-17]
Referenced Libraries
lib
gardenDb.sql
SQL-Week 10-Test1
SQL Week 11 Assignment
unit-test-assignment
week10work
Week3 JavaCodingAssignments
Week5JavaCodingAssignment
xteam
Application.java
Menu.java
ZoneDao.java
Zone.java
1 package entity;
2
3 import java.util.List;
4
5 public class Zone {
6
7 private int zoneID;
8 private String USDAZone;
9 private List<Zone> zones;
10 public Zone(int zoneID, String USDAZone, List<Zone> list) {
11 this.setZoneID(zoneID);
12 this.setUSDAZone(USDAZone);
13 this.setZones(list);
14
15 }
16 public int getZoneID() {
17 return zoneID;
18 }
19 public void setZoneID(int zoneID) {
20 this.zoneID = zoneID;
21 }
22 public String getUSDAZone() {
23 return USDAZone;
24 }
25 public void setUSDAZone(String uSDAZone) {
26 USDAZone = uSDAZone;
27 }
28 public List<Zone> getZones() {
29 return zones;
30 }
31 public void setZones(List<Zone> list) {
32 this.zones = list;
33 }
34
35 }
36

Screenshots of Running Application:



URL to GitHub Repository:

[JessicaSmiejan/Week-10-Coding-assignment (github.com)]