PROMINEO TECH

Relational Databases with MySQL Week 5 Coding 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, with your Java project code, to the 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:

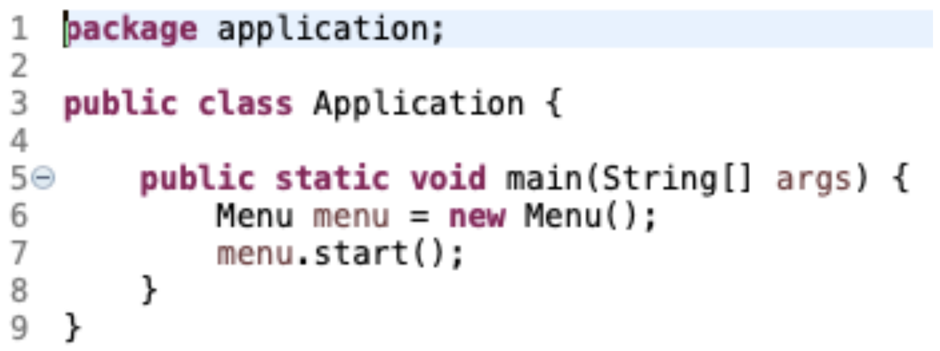
The application does not need to be as complex as the example in the video curriculum.

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

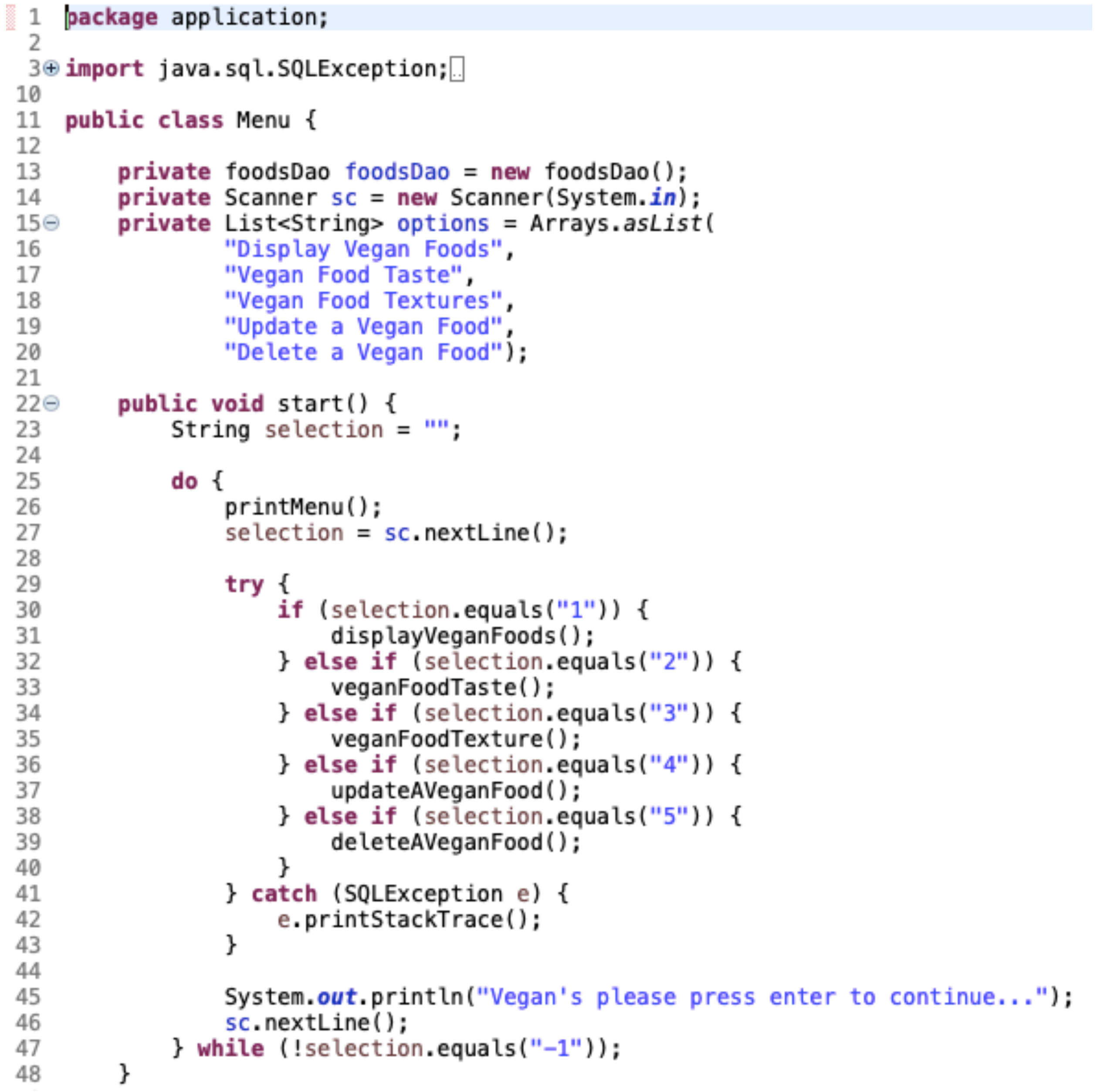
Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

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

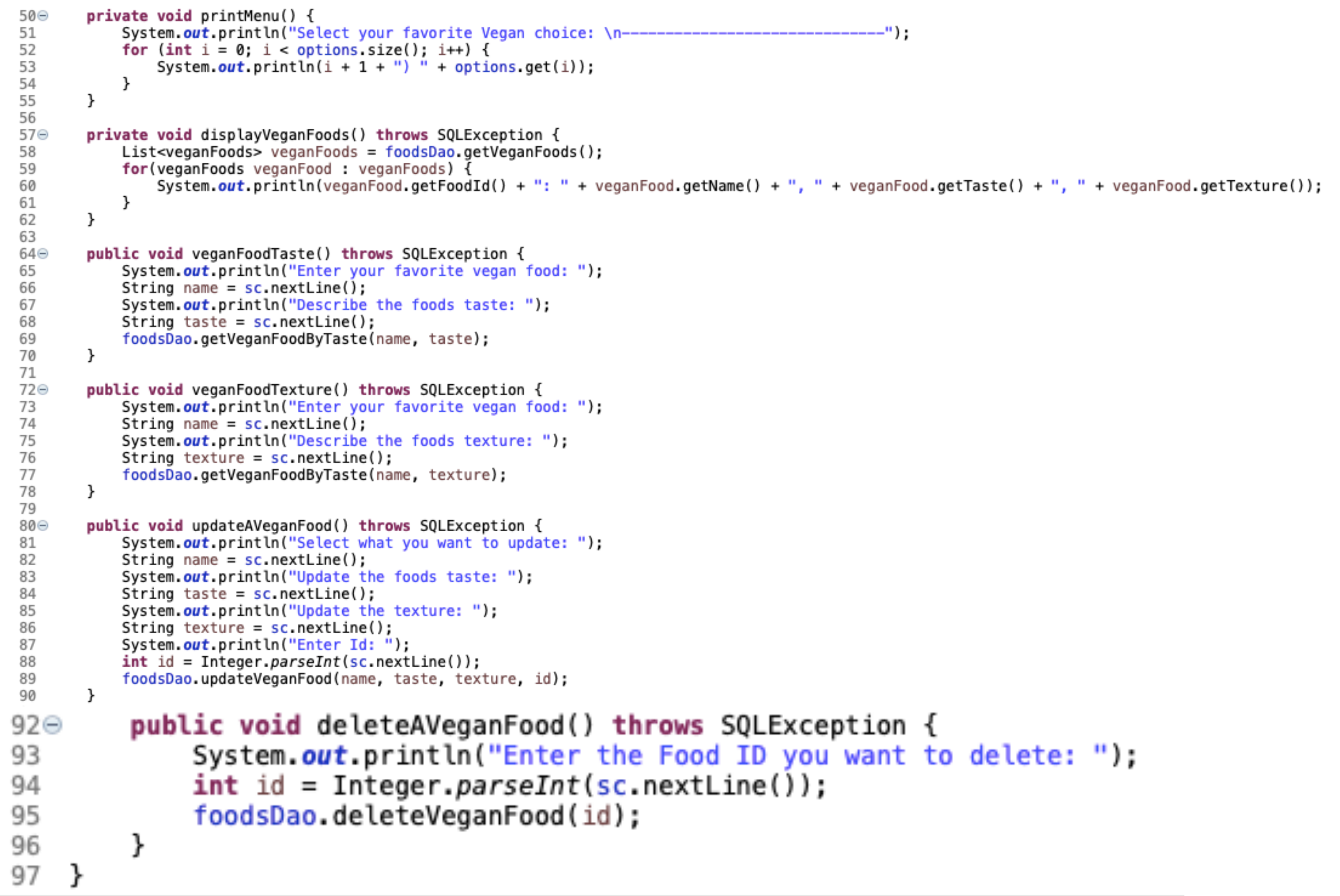
Screenshots of Code:



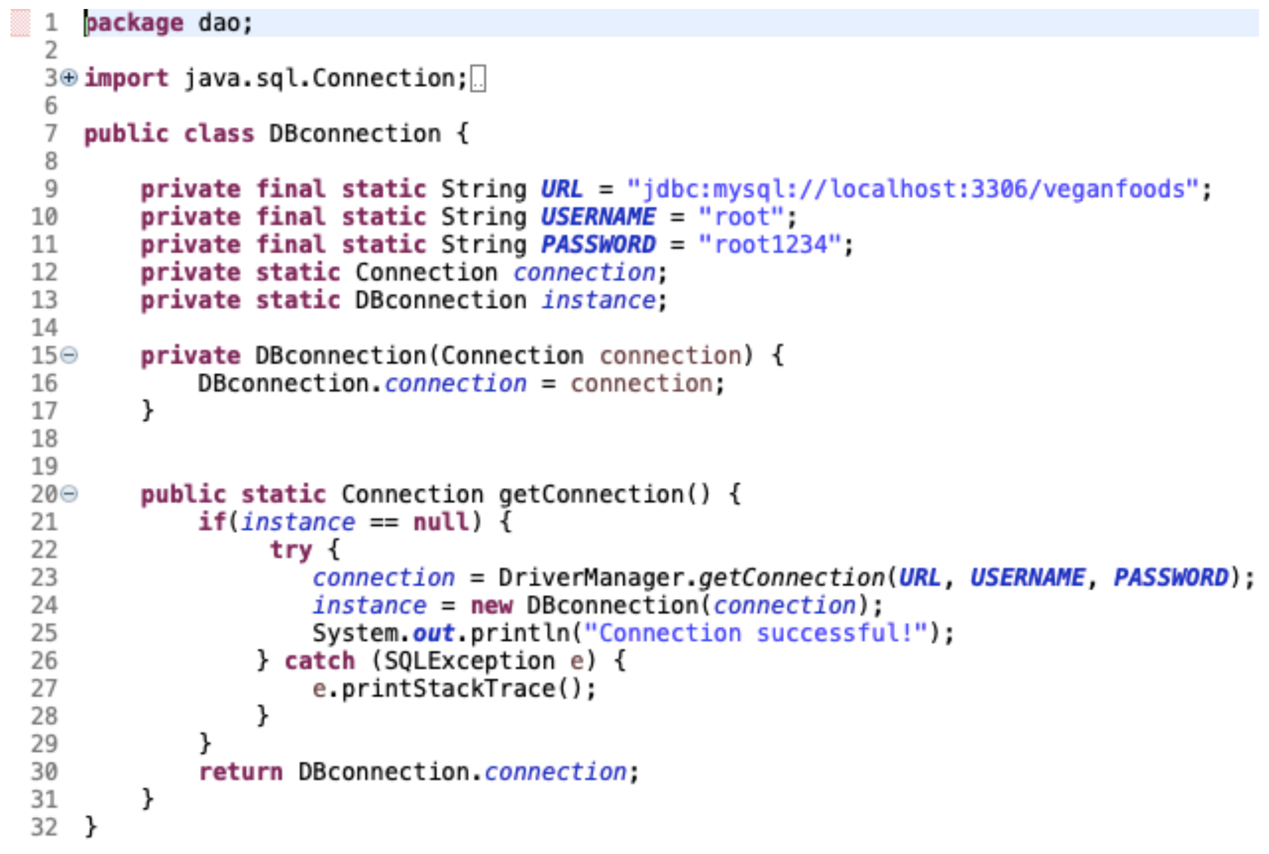
```java
package application;

public class Application {

	public static void main(String[] args) {
		Menu menu = new Menu();
		menu.start();
	}
}
```



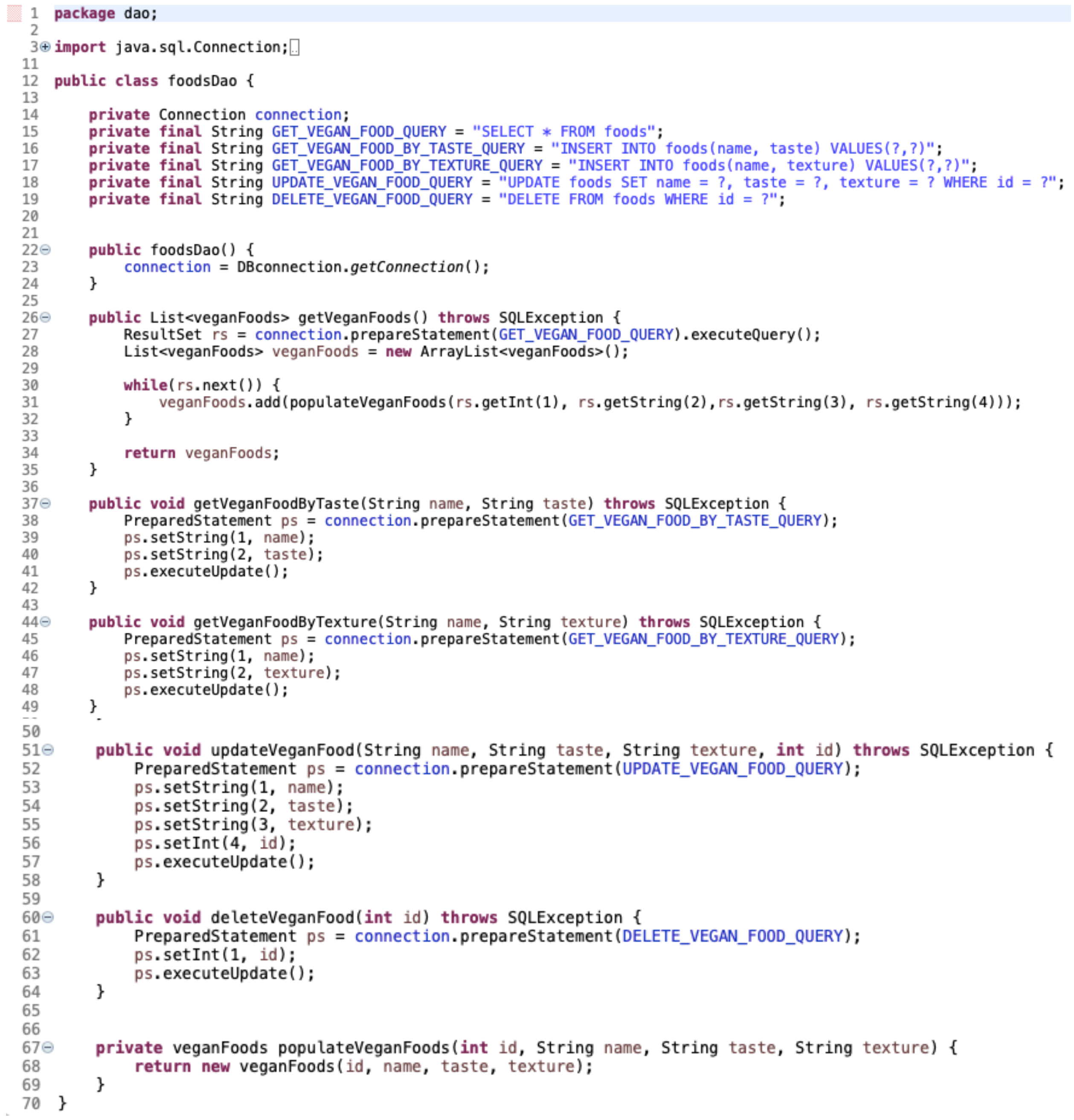
```java
package application;

import java.sql.SQLException;

public class Menu {

    private foodsDao foodsDao = new foodsDao();
    private Scanner sc = new Scanner(System.in);
    private List<String> options = Arrays.asList(
            "Display Vegan Foods",
            "Vegan Food Taste",
            "Vegan Food Textures",
            "Update a Vegan Food",
            "Delete a Vegan Food");

    public void start() {
        String selection = "";

        do {
            printMenu();
            selection = sc.nextLine();

            try {
                if (selection.equals("1")) {
                    displayVeganFoods();
                } else if (selection.equals("2")) {
                    veganFoodTaste();
                } else if (selection.equals("3")) {
                    veganFoodTexture();
                } else if (selection.equals("4")) {
                    updateAVeganFood();
                } else if (selection.equals("5")) {
                    deleteAVeganFood();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }

            System.out.println("Vegan's please press enter to continue...");
            sc.nextLine();
        } while (!selection.equals("-1"));
    }
```



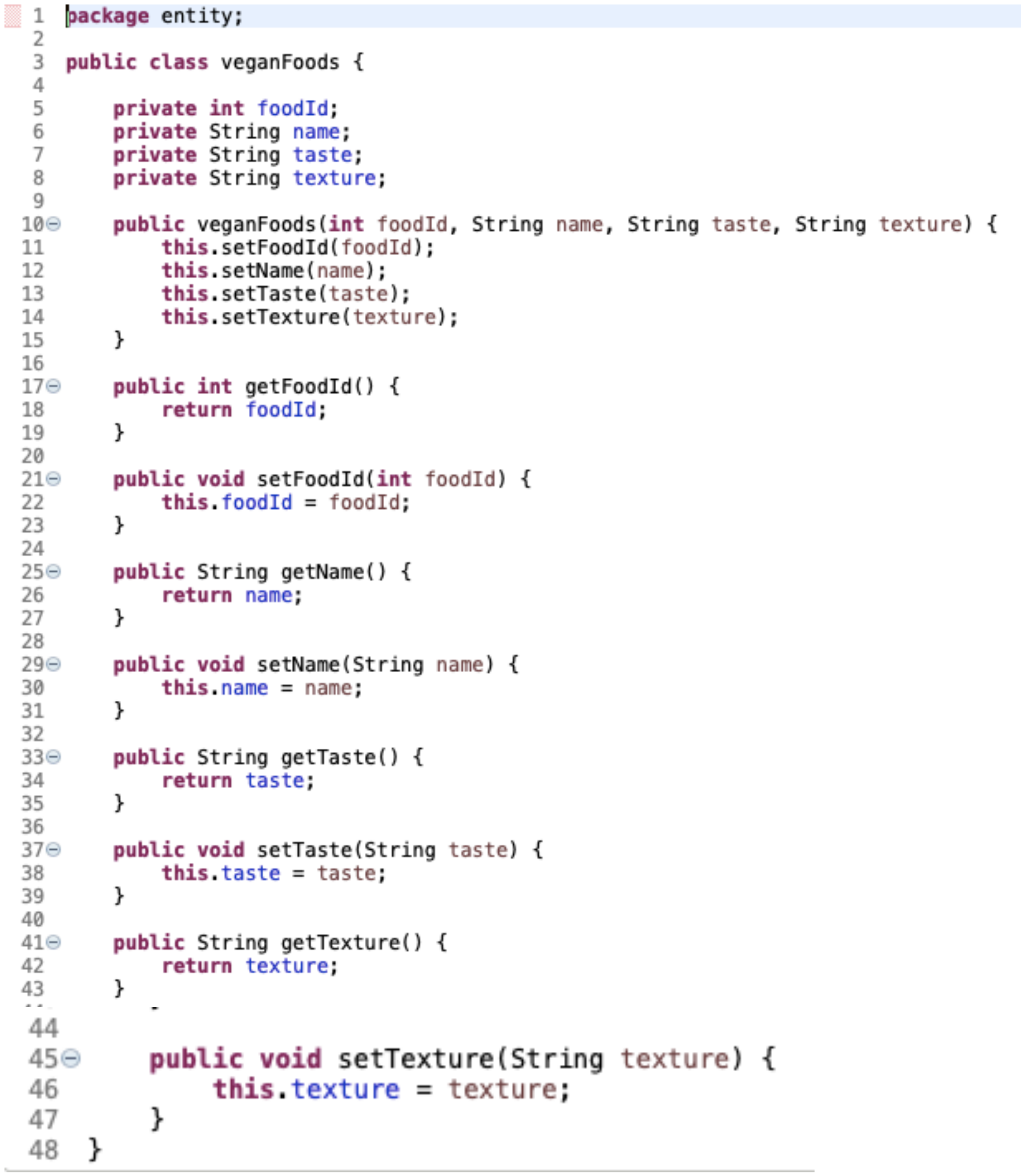
```java
    private void printMenu() {
        System.out.println("Select your favorite Vegan choice: \n-----------------------------");
        for (int i = 0; i < options.size(); i++) {
            System.out.println(i + 1 + ") " + options.get(i));
        }
    }

    private void displayVeganFoods() throws SQLException {
        List<veganFoods> veganFoods = foodsDao.getVeganFoods();
        for(veganFoods veganFood : veganFoods) {
            System.out.println(veganFood.getFoodId() + ": " + veganFood.getName() + ", " + veganFood.getTaste() + ", " + veganFood.getTexture());
        }
    }

    public void veganFoodTaste() throws SQLException {
        System.out.println("Enter your favorite vegan food: ");
        String name = sc.nextLine();
        System.out.println("Describe the foods taste: ");
        String taste = sc.nextLine();
        foodsDao.getVeganFoodByTaste(name, taste);
    }

    public void veganFoodTexture() throws SQLException {
        System.out.println("Enter your favorite vegan food: ");
        String name = sc.nextLine();
        System.out.println("Describe the foods texture: ");
        String texture = sc.nextLine();
        foodsDao.getVeganFoodByTaste(name, texture);
    }

    public void updateAVeganFood() throws SQLException {
        System.out.println("Select what you want to update: ");
        String name = sc.nextLine();
        System.out.println("Update the foods taste: ");
        String taste = sc.nextLine();
        System.out.println("Update the texture: ");
        String texture = sc.nextLine();
        System.out.println("Enter Id: ");
        int id = Integer.parseInt(sc.nextLine());
        foodsDao.updateVeganFood(name, taste, texture, id);
    }
    public void deleteAVeganFood() throws SQLException {
        System.out.println("Enter the Food ID you want to delete: ");
        int id = Integer.parseInt(sc.nextLine());
        foodsDao.deleteVeganFood(id);
    }
}
```



```java
package dao;

import java.sql.Connection;

public class DBconnection {

    private final static String URL = "jdbc:mysql://localhost:3306/veganfoods";
    private final static String USERNAME = "root";
    private final static String PASSWORD = "root1234";
    private static Connection connection;
    private static DBconnection instance;

    private DBconnection(Connection connection) {
        DBconnection.connection = connection;
    }


    public static Connection getConnection() {
        if(instance == null) {
            try {
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                instance = new DBconnection(connection);
                System.out.println("Connection successful!");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return DBconnection.connection;
    }
}
```

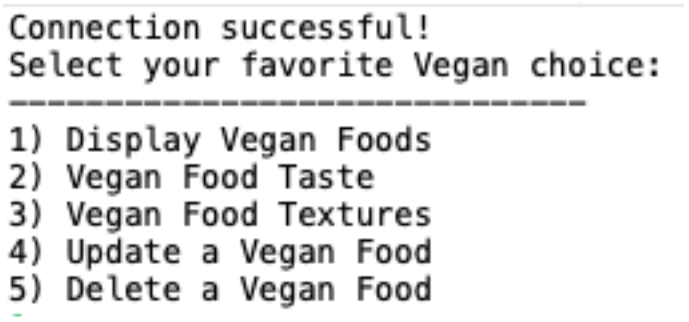


```java
package dao;

import java.sql.Connection;

public class foodsDao {

    private Connection connection;
    private final String GET_VEGAN_FOOD_QUERY = "SELECT * FROM foods";
    private final String GET_VEGAN_FOOD_BY_TASTE_QUERY = "INSERT INTO foods(name, taste) VALUES(?,?)";
    private final String GET_VEGAN_FOOD_BY_TEXTURE_QUERY = "INSERT INTO foods(name, texture) VALUES(?,?)";
    private final String UPDATE_VEGAN_FOOD_QUERY = "UPDATE foods SET name = ?, taste = ?, texture = ? WHERE id = ?";
    private final String DELETE_VEGAN_FOOD_QUERY = "DELETE FROM foods WHERE id = ?";


    public foodsDao() {
        connection = DBconnection.getConnection();
    }

    public List<veganFoods> getVeganFoods() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_VEGAN_FOOD_QUERY).executeQuery();
        List<veganFoods> veganFoods = new ArrayList<veganFoods>();

        while(rs.next()) {
            veganFoods.add(populateVeganFoods(rs.getInt(1), rs.getString(2),rs.getString(3), rs.getString(4)));
        }

        return veganFoods;
    }

    public void getVeganFoodByTaste(String name, String taste) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_VEGAN_FOOD_BY_TASTE_QUERY);
        ps.setString(1, name);
        ps.setString(2, taste);
        ps.executeUpdate();
    }

    public void getVeganFoodByTexture(String name, String texture) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_VEGAN_FOOD_BY_TEXTURE_QUERY);
        ps.setString(1, name);
        ps.setString(2, texture);
        ps.executeUpdate();
    }

    public void updateVeganFood(String name, String taste, String texture, int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_VEGAN_FOOD_QUERY);
        ps.setString(1, name);
        ps.setString(2, taste);
        ps.setString(3, texture);
        ps.setInt(4, id);
        ps.executeUpdate();
    }

    public void deleteVeganFood(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_VEGAN_FOOD_QUERY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }


    private veganFoods populateVeganFoods(int id, String name, String taste, String texture) {
        return new veganFoods(id, name, taste, texture);
    }
}
```



```java
package entity;

public class veganFoods {

    private int foodId;
    private String name;
    private String taste;
    private String texture;

    public veganFoods(int foodId, String name, String taste, String texture) {
        this.setFoodId(foodId);
        this.setName(name);
        this.setTaste(taste);
        this.setTexture(texture);
    }

    public int getFoodId() {
        return foodId;
    }

    public void setFoodId(int foodId) {
        this.foodId = foodId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getTaste() {
        return taste;
    }

    public void setTaste(String taste) {
        this.taste = taste;
    }

    public String getTexture() {
        return texture;
    }

    public void setTexture(String texture) {
        this.texture = texture;
    }
}
```

Screenshots of Running Application:



```
Connection successful!
Select your favorite Vegan choice:
--------------------------------
1) Display Vegan Foods
2) Vegan Food Taste
3) Vegan Food Textures
4) Update a Vegan Food
5) Delete a Vegan Food
```

URL to GitHub Repository: https://github.com/DesmondYo/VeganFood