COMSATS University Islamabad

**Topics in Software Engineering**

**Group Members:** ALI HASSAN (FA19-BSE-075)

Ayesha Shahzad (FA19-BSE-065)

Mehak Butt (FA19-BSE-047)

Babar Jawad Anjum (FA19-BSE-025)

**Course Title:** Topics in Software Engineering

**Section:** A

**Assessment:** Assignment 1

**Submitted To:** Mam Fatima Sabir

**Date:** October 3rd, 2022

**Campus:** Lahore Campus

**PART A**

1. **Deprecated technology or APIs:**

* Java.util.Date;

The above-mentioned java method is deprecated now that is used in the given Restaurant Management System. The new method is replaced by *Calendar.get(Calendar.DAY\_OF\_MONTH)*

* Firebase:

The given project is using firebase database since 2018. As this project is a Legacy Project, so due to this, Firebase has been updated many time during these four years. So many of the Firebase things is outdated now used in the project.

1. **Potential issues missing in docs but available in code:**

Attendance Activity is present in the code but missing in the documentation.

In code, they have used adapters in many places but in the documentation, I have not seen anything related to adapter so this is missing in the documentation but available in the code.

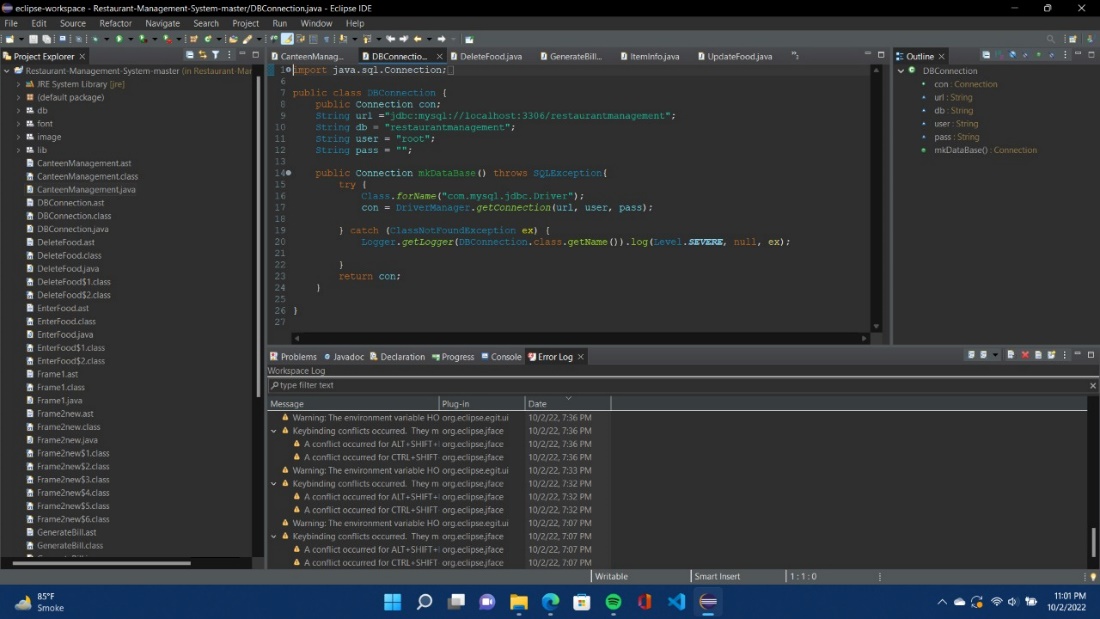
1. **Missing of technical documentation:**

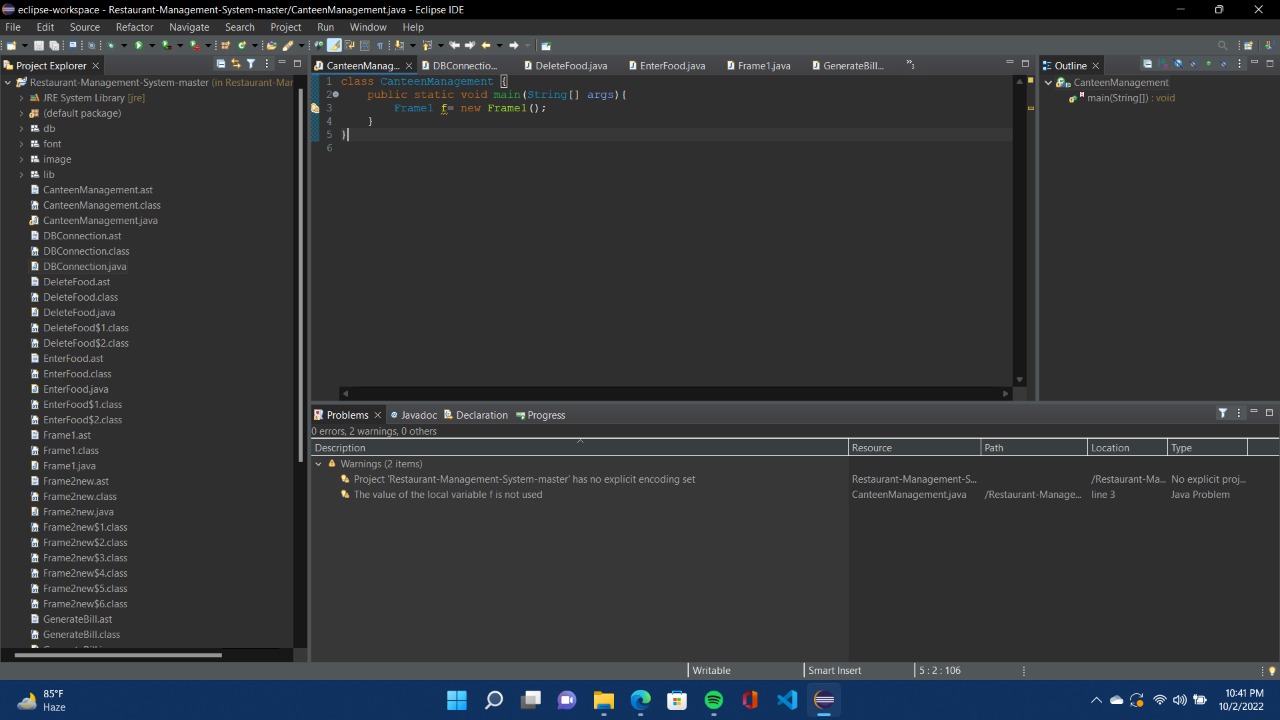
After reading the documentation, everything is included in the document like Functional requirements, Non-Functional requirements etc. etc. Therefore, this is not applicable for the given project because according to me, everything is available in the documents. Nothing is missing.

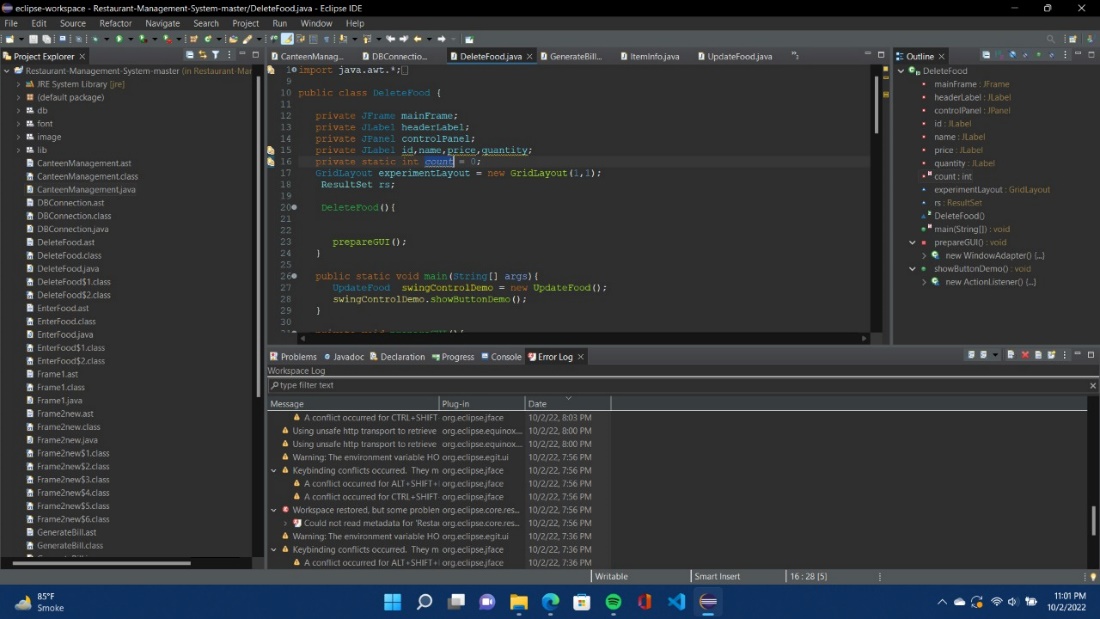
**PART B**

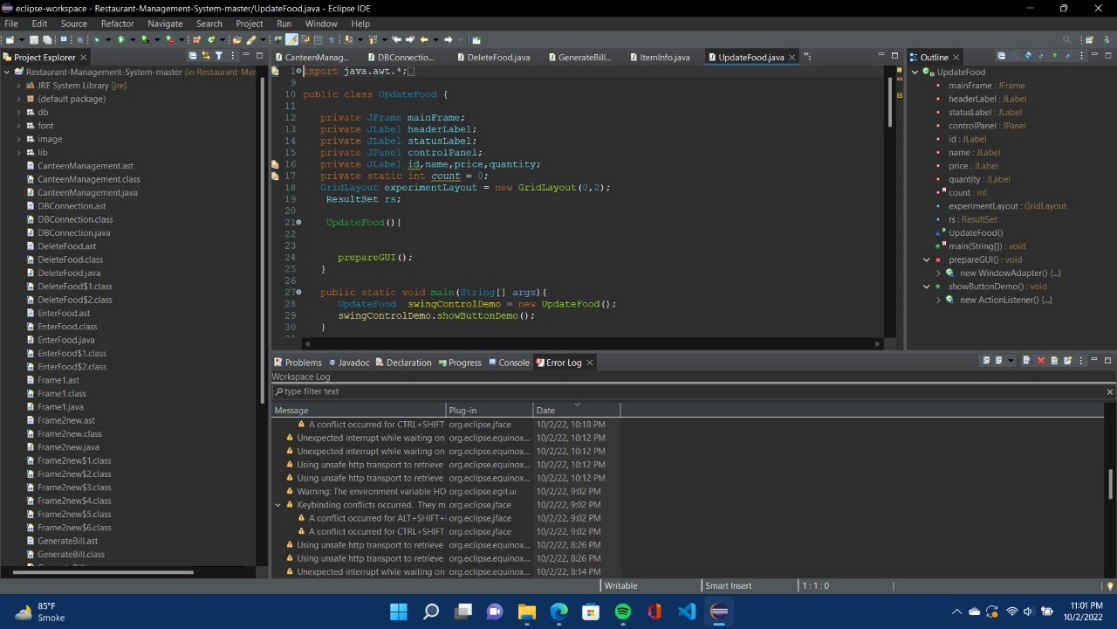
1. **Create and view code issues directory from your editor:**

* Keybinding conflicts
* Unused value of local variable
* Unsafe HTTP transport

****

****

****

****

1. **Track and prioritize code improvements like technical debt:**

For the tracking and prioritizing of rules, we would suggest:

* One declaration per line
* At least one constructor
* Avoid dollar signs
* Class naming conventions to be followed
* Field declarations should be at the start of class
* Linguistic naming

1. **Check your code quality:**

Due to the following reasons, the code has misinterpreted info, which leads to the unsatisfactory code quality:

* Improper use of spaces and brackets is taking extra memory
* Declared but unused variables in all over the code
* Poor indentation

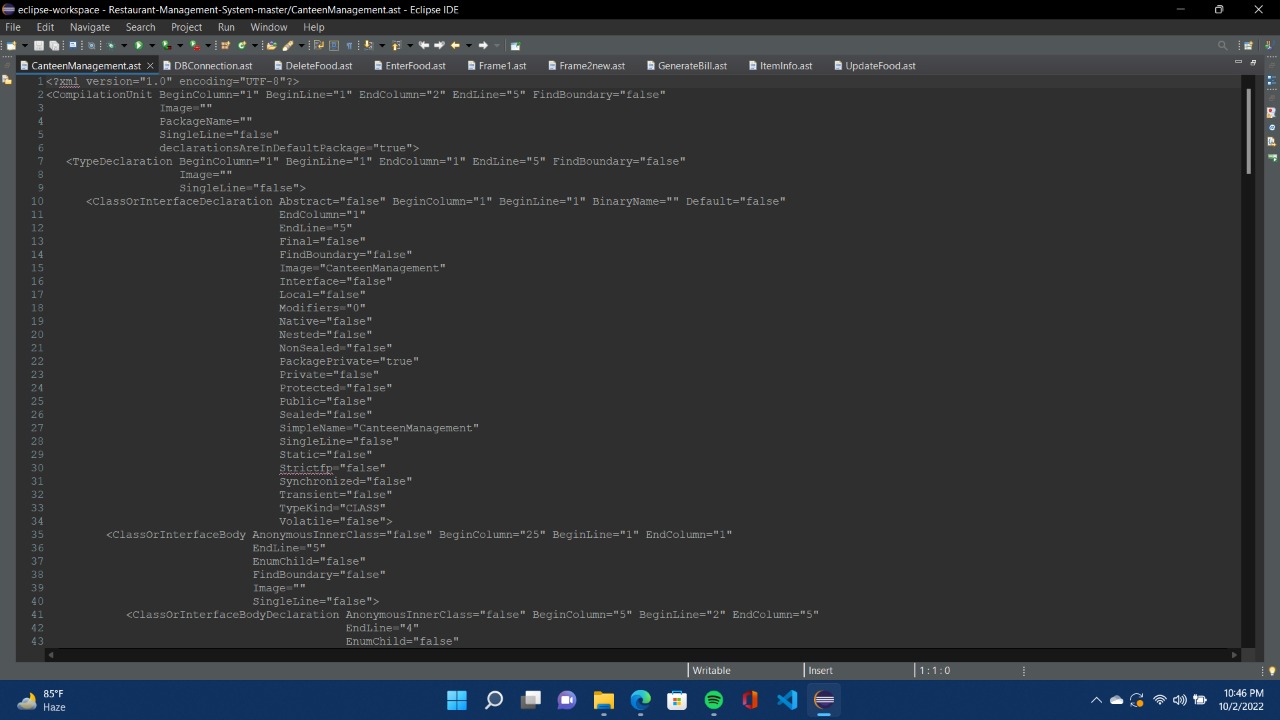
1. **Apply at least 3 PMD rules with the help of tool:**

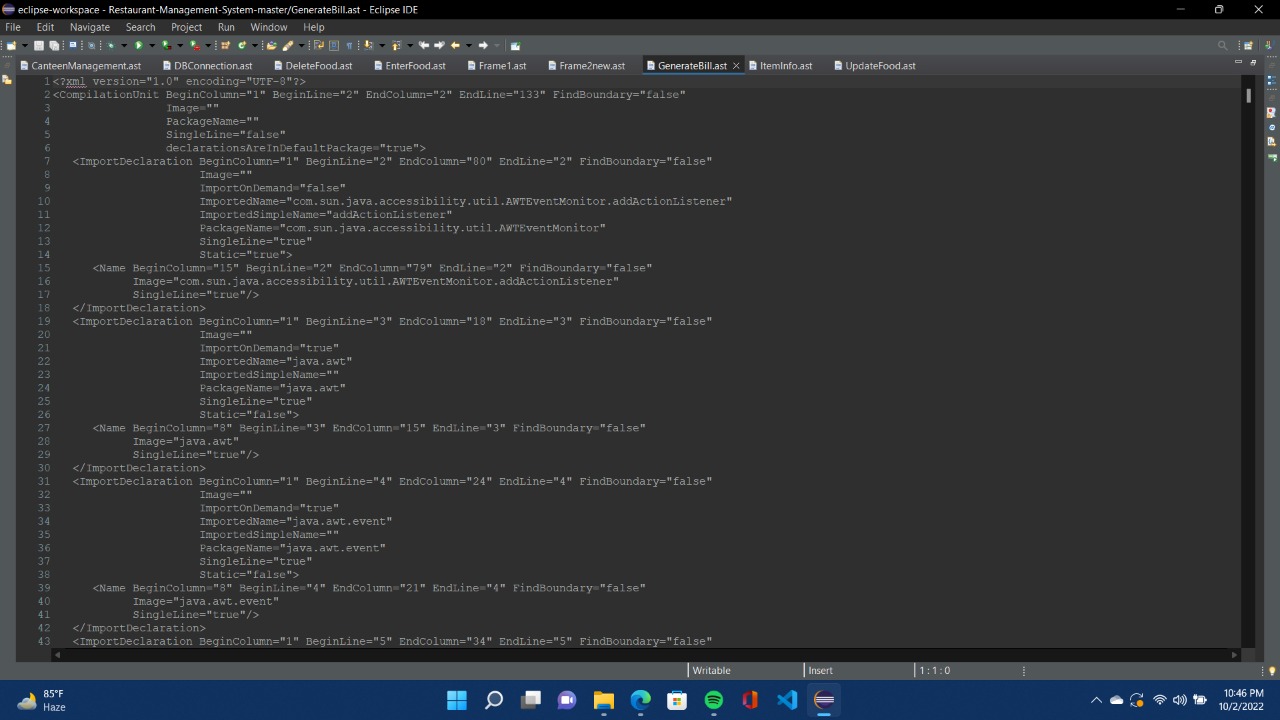
Change is recommended due to unused variables in the code

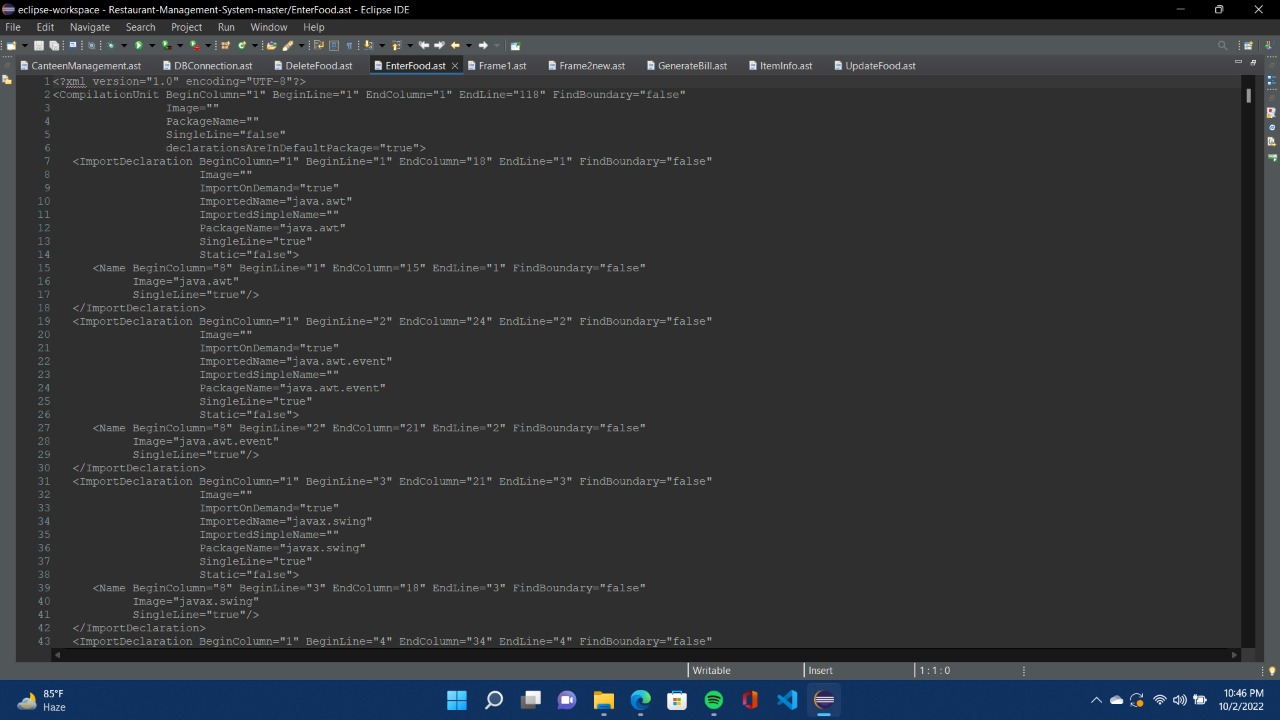
1. **Generate the Abstract Syntax Tree of your source code using PMD:**

Following screenshots of the abstract tree files of some java codes depict the information about the:

* Starting and ending of each method
* Declaration of variables
* Starting and ending point of the code
* Placement of the images in the overall code

****

****

****

**PART C**

**Rule # 1: Whitespaces & Indentation**

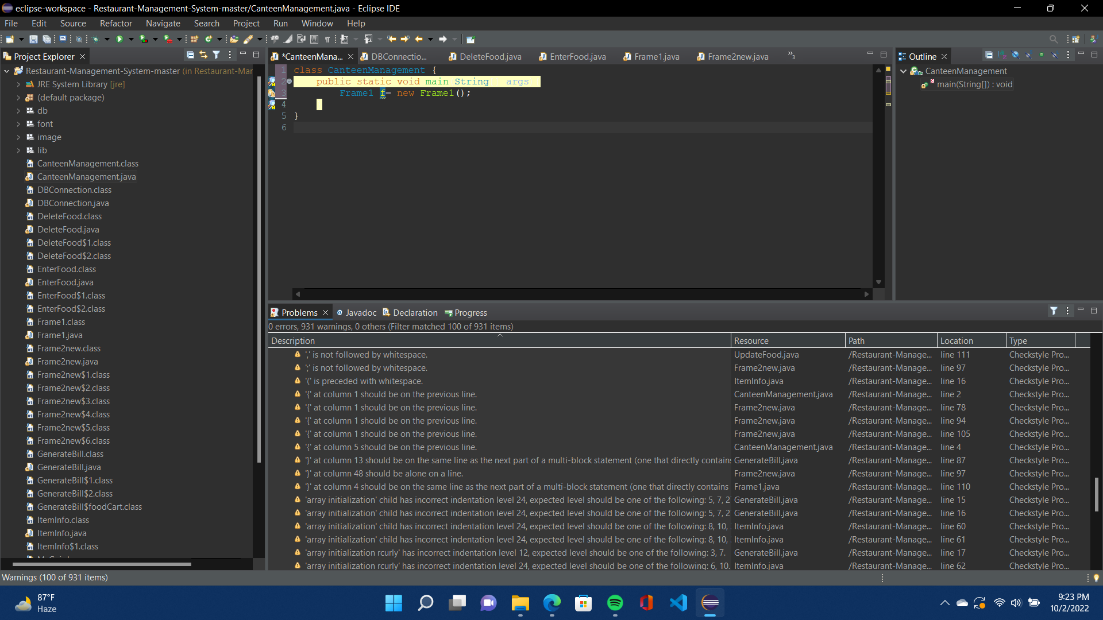
Checkstyle strictly enforces the whitespaces and indentation in the JAVA code. From the provided project we have interpreted that:

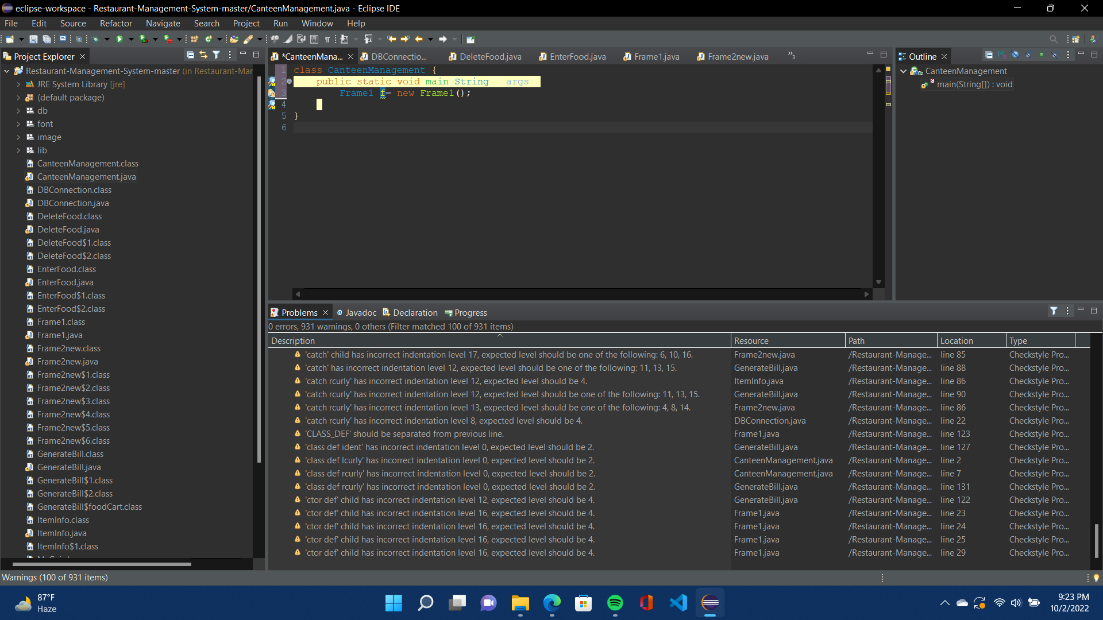
* There are several extra whitespaces left in the java code files.
* Before ending and starting of the brackets, extra or no space can be seen.
* There is no proper sequence followed in the java code files, which leads to the poor indentation.
* Every class should be indented properly so that the objects.

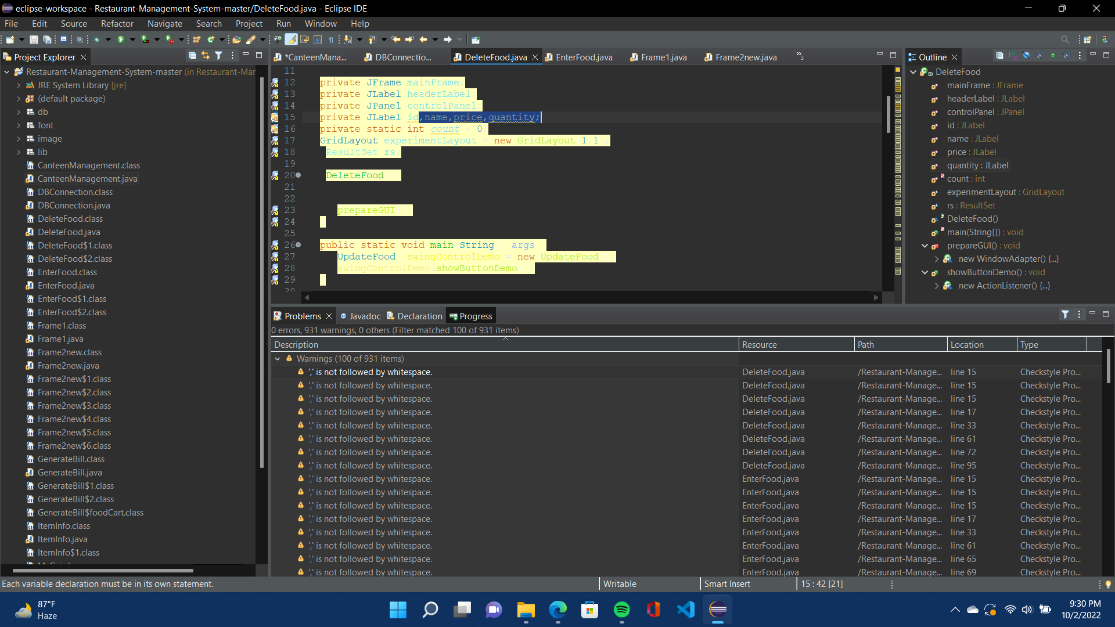
**Rule # 2: Commenting**

By applying the Checkstyle’s commenting rule, we have observed the following:

* No Single line block comment is written within an empty code block.
* Checkstyle limits to clearly detect user intention of explanation target - above or below, if comment is placed at the end of the empty code block.
* In the provided java files, the group of methods are not separated by single line comment border.

****

****

****