COMSATS UNIVERSITY ISLAMABAD
(Lahore Campus)

Group Members:
1) SAFEER ALI MEHDI(FA19-BSE-051)
3) ALI ZAR(FA19-BSE-005)
4) M. BILAL ZAMEER KHAN(FA19-BSE-019)
5) M. ALI LIAQAT(FA19-BSE-077)

Section : A

Assignment # 01

Course : Topics in Software Engineering

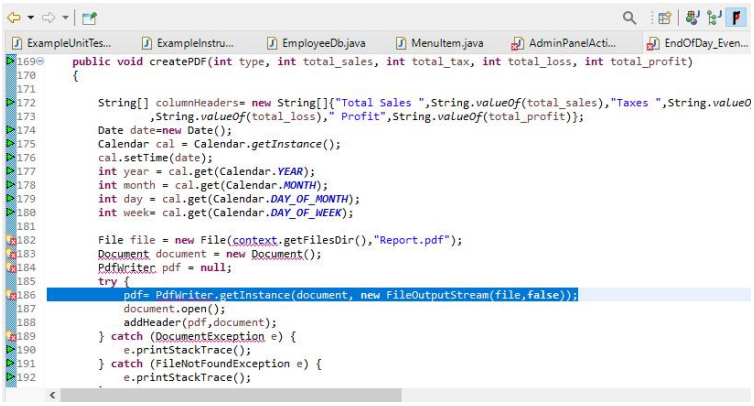
Teacher : DR. FATIMA SABIR

Date To Submit : 3rd OCT 2022

Enable the project using any IDE for code review. It would be better to use Eclipse or Netbeans. You may use VS Code too.

a) **Explore your project and find all options discussed in lecture 2 from Chapter 2. You need to prepare a google doc that highlights the potential issue that your code has. You may identify these issues with the help of**

i) Identify deprecated technology or APIs. Just report the diagram.



ii) Potential issues missing in the documentation but available in the code

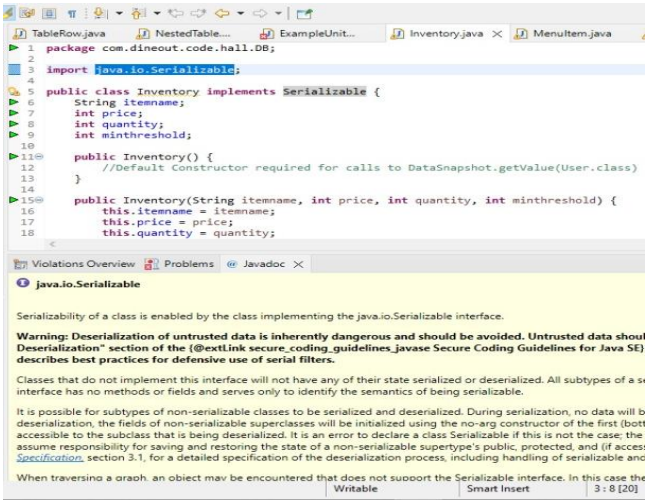
- Using inheritance between classes what was not mention in documentation.

- Using the different data types for the variables

- Different parameters are declare in code

iii) missing of technical documentation (if applicable)

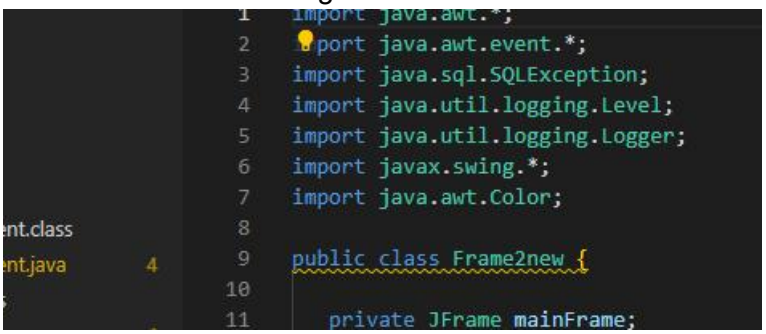
- Technical documentation is available for every builtin method



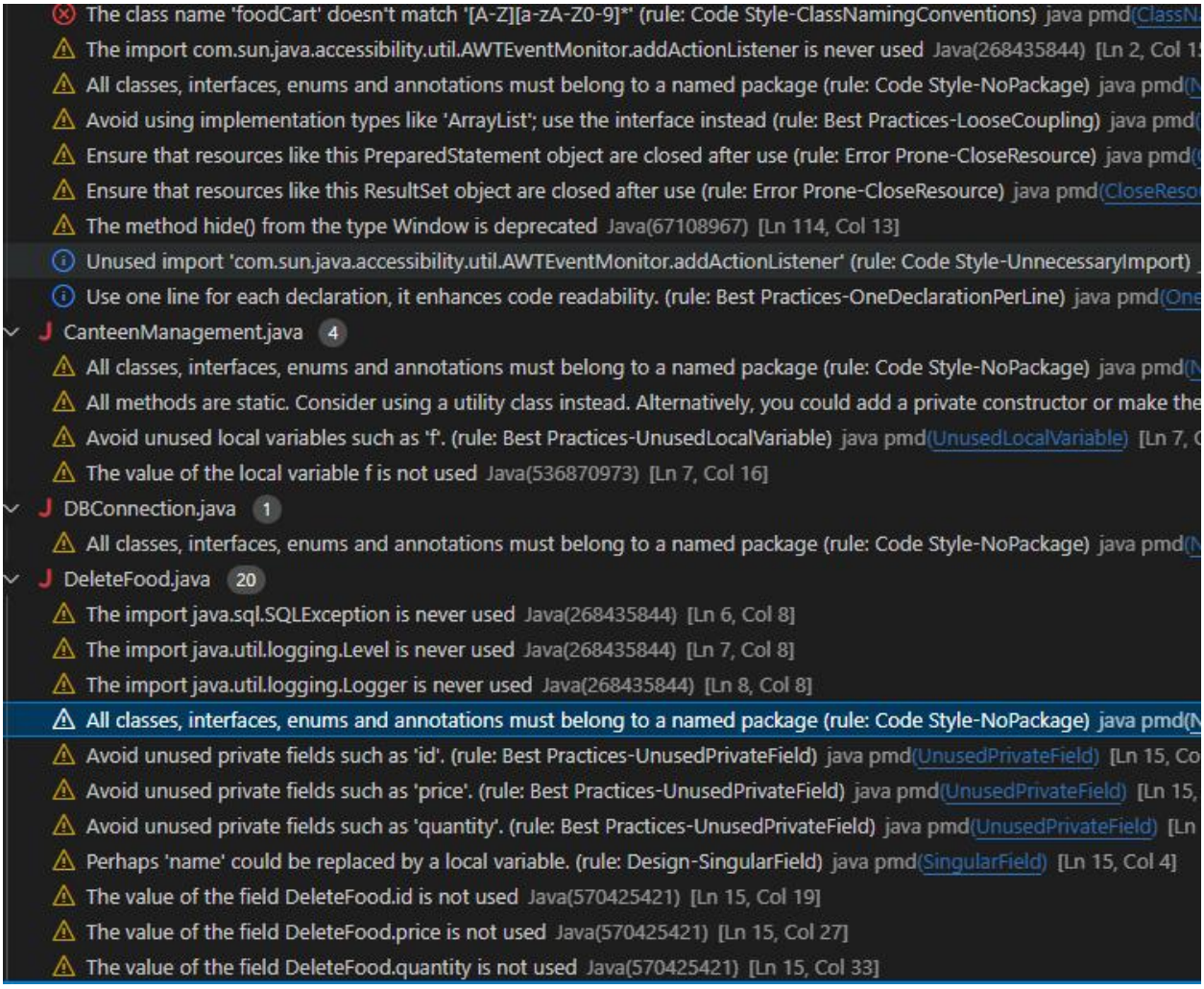
b) Use PMD to help identify potential coding errors and customize the rules you use to make sure only pertinent rules are applied to your source code.

i) Create & view code issues directly from your editor

- Not follow class naming conventions :



- Unused libraries and variables in the code



ii) Track & prioritize code improvements like technical debt

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

- One declaration per line
- Avoid too many methods used
- Avoid unused private methods
- Avoid dollar signs
- Class naming conventions to be followed
- Field declarations should be at the start of class
- Linguistic naming

iii) Check your code quality

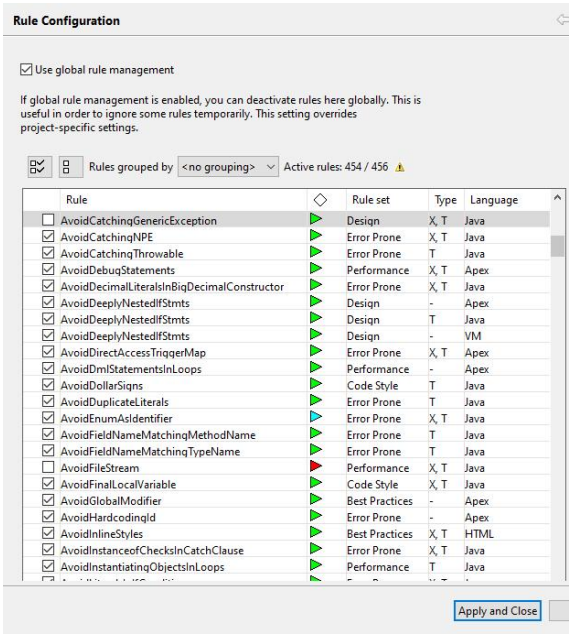
Due to the following reasons, the code has misinterpreted info, which leads to the unsatisfactory code quality:

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

iv) Apply at least 3 PMD rules with the help of tool

PMD rules are given below:

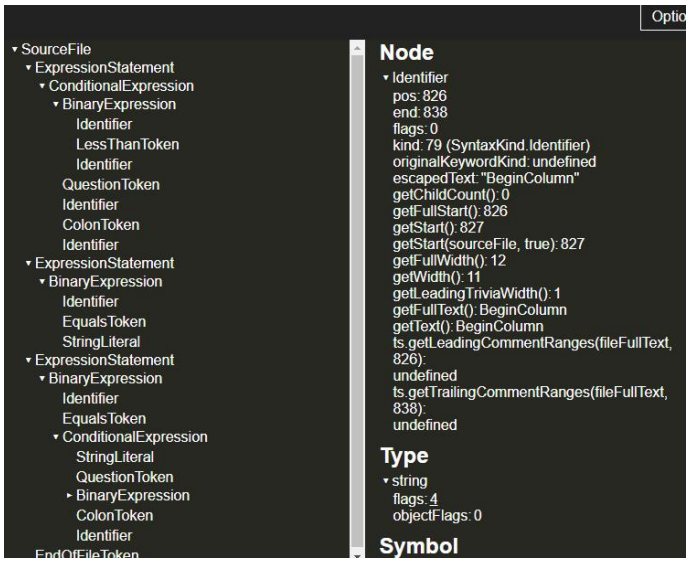
- Class naming conventions
- Class has at least one constructor
- Field declaration should be at start of the class
- Short variables name
- Avoid generic exceptions
- Avoid extra parenthesis



v) Generate the Abstract Syntax Tree of your source code suing 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



c) Use the Check Style tool to review your code. At least apply 2 rules on your code using CHeckStyle

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.