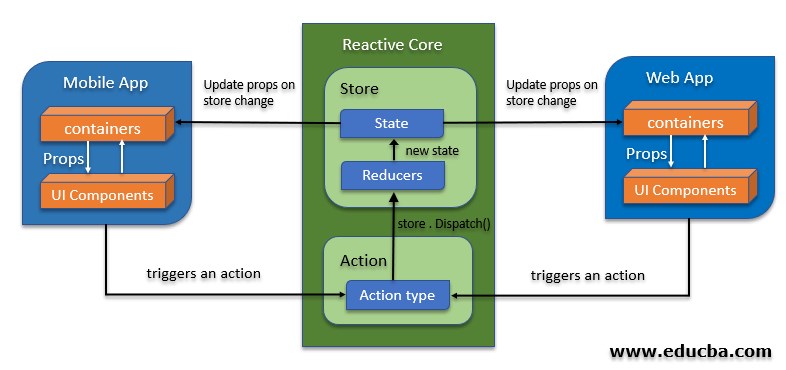
|  |  |
| --- | --- |
| A picture of a winding road and trees  aSSIGNEMNT #1  10/2/2022 | TOPICS IN SOFTWARE ENGINEERING-1  SUBMITTED TO FATIMA SABIR  FA19-BSE-058  HASNAIN ALLAH RAKHA  FA19-BSE-064  MOIZ RASHEED  FA19-BSE-098  MUHAMMAD SALMAN AHMED |

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

1. **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**
2. **Identify deprecated technology or APIs. Just report the diagram.**

**Solution: -**

* The technology used in project for mobile application development is java and xml however these days in market the demand is high for android applications made using ReactNative and flutter as compared to java and xml hence ReactNative and flutter is used for android applications.



* Older version of java and xml are used which may cause problem when new functionality will be added later on.

1. **Potential issues missing in the documentation but available in the code**

**Solution :**

This part cannot be figured out since the application in not compiled and run to find potential issues.

**iii) missing of technical documentation ( if applicable )**

**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.**

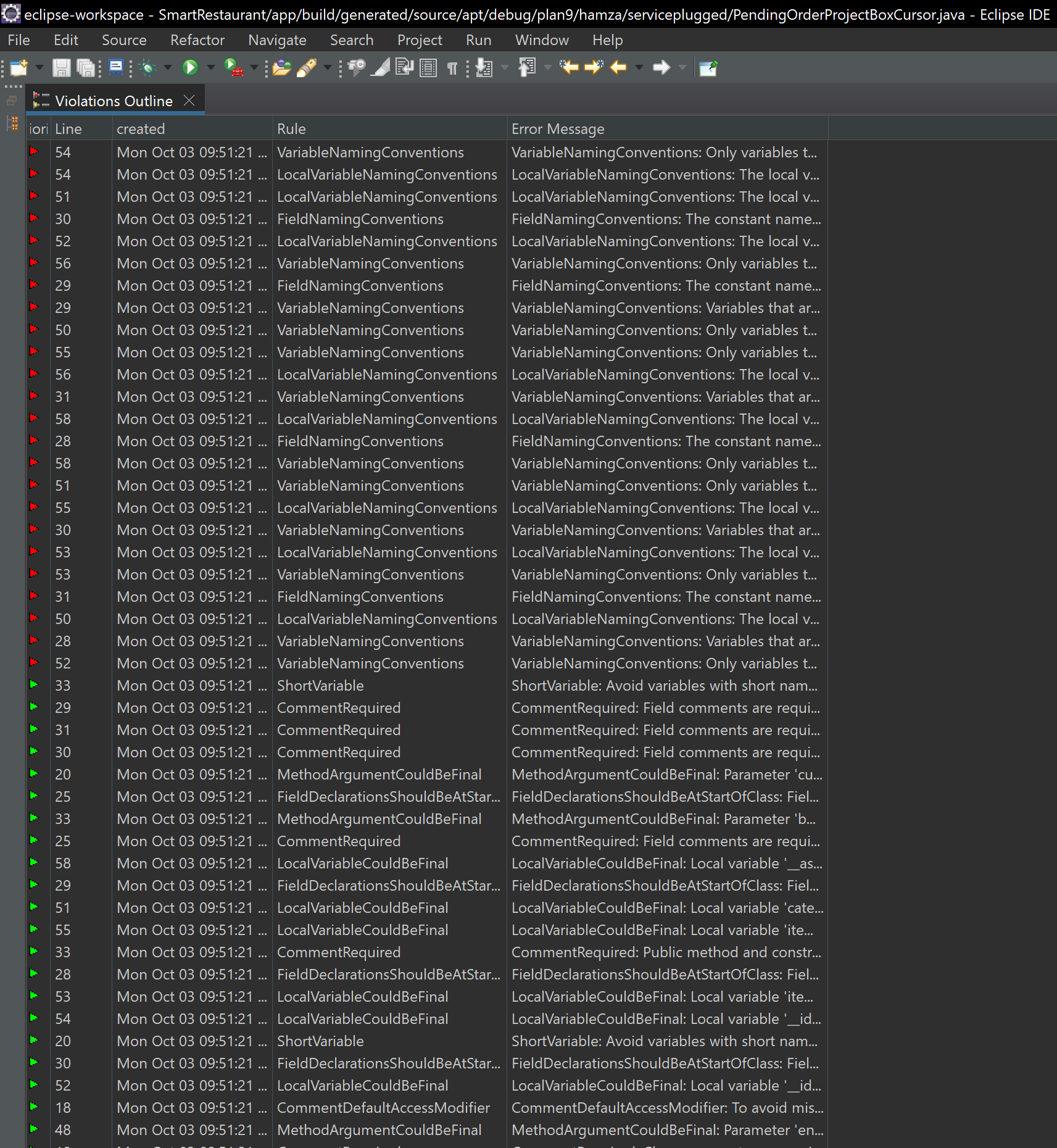
1. Create & view code issues directly from your editor
2. Track & prioritize code improvements like technical debt
3. Check your code quality
4. Apply at least 3 PMD rules with the help of tool
5. Generate the Abstract Syntax Tree of your source code using PMD.

**Solution :-**

**Following Java File is chosen**

SmartRestaurant > app/build/generated/source/apt/debug>plan9.hamza.serviceplugged > PendingOrderProjectBoxCursor.java

**Create & view code issues directly from your editor:**

****

**Code Issues Using PMD:**

* **Rule FieldNamingConventionIssue**

Field Naming Convention Issue: The Constant name ***\_\_ID\_item\_price //LINE 31***

* **Rule CommentRequired**

CommentRequired: Field comments are required private final static int ***\_\_ID\_category\_id*** = PendingOrderProjectBox\_.***category\_id***.id; //LINE 29

* **Rule LocalVariableCouldBeFinal**

LocalVariableCouldBeFinal: Local variable ‘name’ could be final String name = entity.getName();// LINE 49

**Fixing Code Issues Created By PMD:**

* **Rule FieldNamingConventionIssue**

Field Naming Convention issue can be solved by using following naming convention

private final static int ***ID\_ITEM\_PRICE***=PendingOrderProjectBox\_.***item\_price***.id;//LINE 31

* **Rule CommentRequired**

CommentRequired issue can be solved by adding comment who added this variable

/\*\*

\* **@author** Hasnain Allah Rakha

\*/

private final static int ***\_\_ID\_category\_id*** = PendingOrderProjectBox\_.***category\_id***.id; //LINE 29

* **Rule LocalVariableCouldBeFinal**

LocalVariableCouldBeFinal can be solved if ‘name’ will not be assigned again final String name = entity.getName(); //LINE 49

**Abstract Syntax Tree of Source Code**

* Abstract Syntax tree of source Code is attached as separate file.

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

**Code Issues By Using CheckStyle**

* ‘memeber def’ modifer has incorrect indentation level 4

private final static int ***\_\_ID\_category\_id*** = PendingOrderProjectBox\_.***category\_id***.id;//LINE 29

The Above can be solved by just giving two spaces before each line of code

* static modifier out of order with the jls suggestions

private final static int ***\_\_ID\_item\_price*** = PendingOrderProjectBox\_.***item\_price***.id; //LINE 31

The Above can be solved by adding static before private and final.

Static private final int ***\_\_ID\_item\_price*** = PendingOrderProjectBox\_.***item\_price***.id;