

TEST REPORT

GROUP 5 – SWT301

I. Introduce about static test tool: SonarLint for NetBeans

1. About Sonarlint

SonarLint for NetBeans is a plugin that integrates the SonarLint code analysis tool with the NetBeans Integrated Development Environment (IDE). SonarLint is a code quality tool that analyzes source code for bugs, vulnerabilities, and code smells. It provides developers with real-time feedback on their code as they write it, helping them to identify and fix potential issues before they become major problems.

The SonarLint for NetBeans plugin adds the SonarLint tool to the NetBeans IDE, allowing developers to easily run code analysis on their projects from within the IDE. The plugin highlights issues in the code editor and provides detailed information about each issue, including a description of the problem and suggestions for how to fix it. The plugin can be configured to use rules from different quality profiles, depending on the requirements of the project.

Overall, SonarLint for NetBeans is a powerful tool for developers who want to improve the quality of their code and ensure that it meets best practices and industry standards.

2. Advantages and Disadvantages

Advantages:

- Code quality improvement: SonarLint helps to identify and fix bugs, vulnerabilities, and code smells, leading to improved code quality.
- Early detection of issues: SonarLint provides real-time feedback on code as developers write it, helping to catch issues early in the development cycle.
- Integration with IDE: The SonarLint plugin integrates seamlessly with the NetBeans IDE, making it easy for developers to use the tool within their workflow.
- Customizable rules: The plugin allows developers to customize the quality profiles to suit their specific requirements, enabling them to focus on the issues that matter most.

Disadvantages:

- False positives/negatives: Like any code analysis tool, SonarLint may generate false positives or false negatives, requiring additional effort to identify and correct issues.

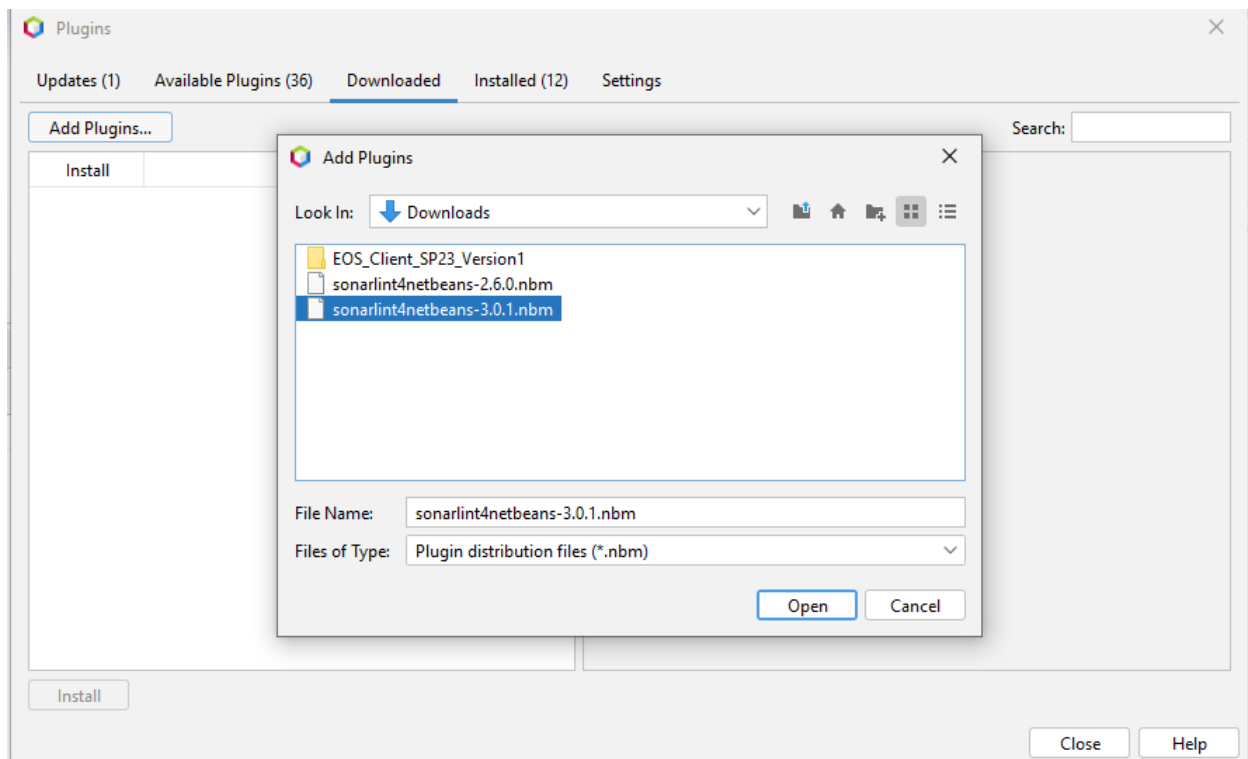
3. How to install and use

- Access to <https://plugins.netbeans.apache.org/catalogue/?id=21>

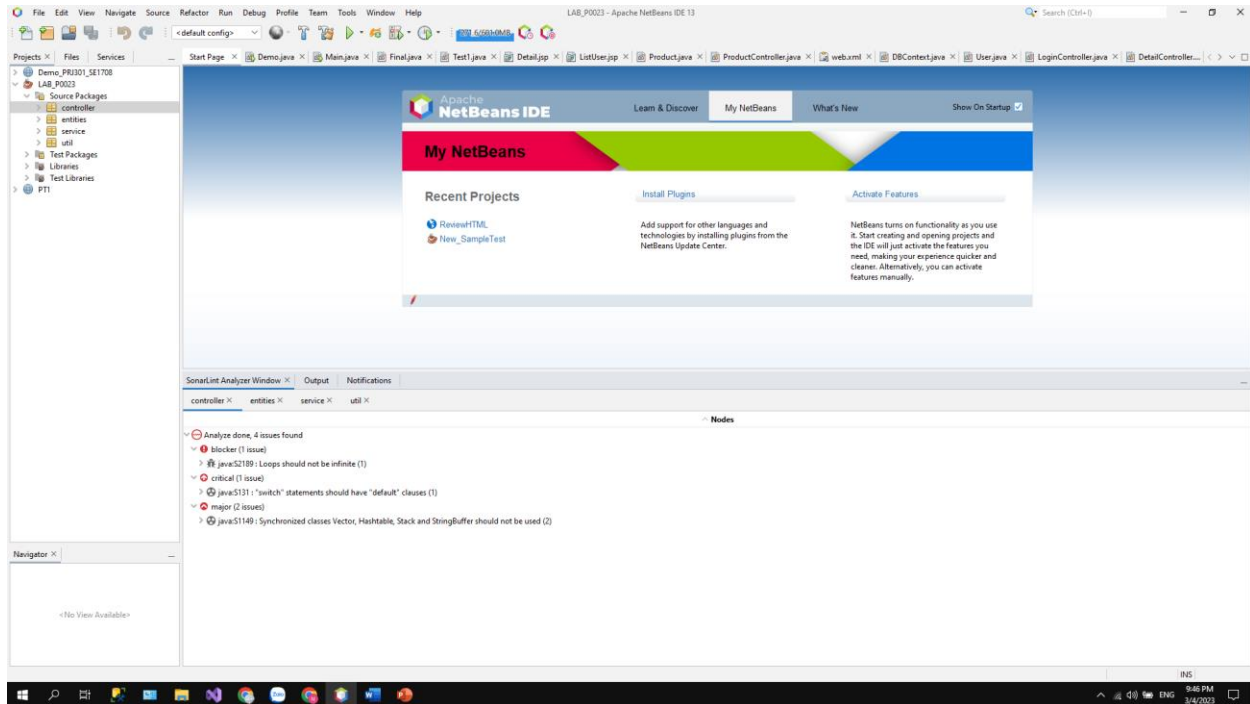
Download matrix

Plugin version	NetBeans compatible
2.0.0	NB 12.0 NB 11.0
2.1.0	NB 12.0 NB 11.0
2.2.0	NB 12.0 NB 11.0
2.3.0	NB 12.0 NB 11.0
2.4.0	NB 12.0 NB 11.0
2.5.0	NB 13 NB 12.0 NB 11.0
2.6.0	NB 15 NB 14 NB 13 NB 12.0 - Verified NB 11.0 - Verified
3.0.0	NB 16 NB 15 NB 14 NB 13
3.0.1	NB 17 - Verified NB 16 - Verified NB 15 - Verified NB 14 - Verified NB 13 - Verified

- Download the version corresponding to the corresponding netbean
- Open netbeans → Tool → Plugin → Downloaded → Choose file had been downloaded



- Click Install and restart Netbeans
- Right click on folder of project you want to analyze, choose Analyze with SonarLint
- You can double click on the bug report to view details



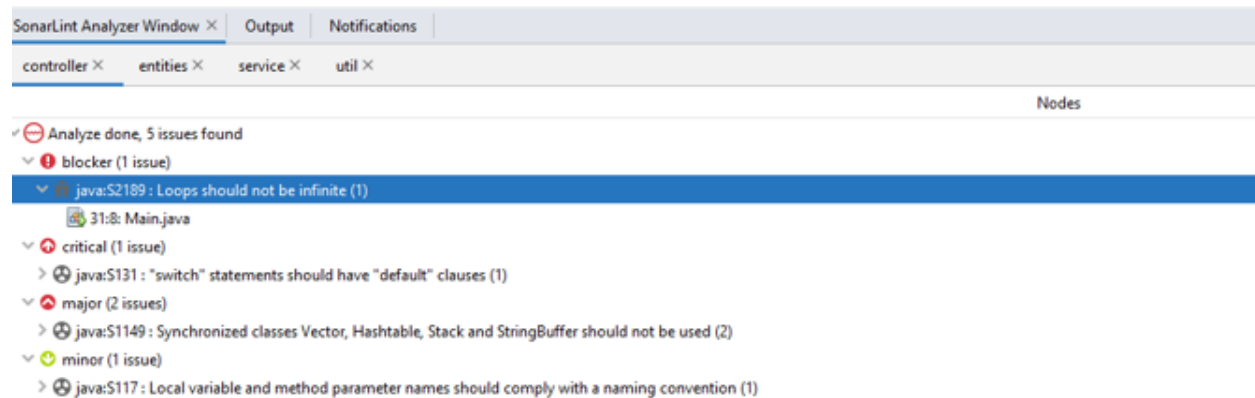
II. Report

Object for analysis: source code from LAB211 subject

Total LOCs: 620

Total number of types of bugs found: 14

A. Folder “controller”:



1. *java:S2189 : Loops should not be infinite (1);* Level: blocker

Solution: don't use "while(true)"

2. *java:S131 : "switch" statements should have "default" clauses (1);* Level: critical

Solution: Add the "default" clauses

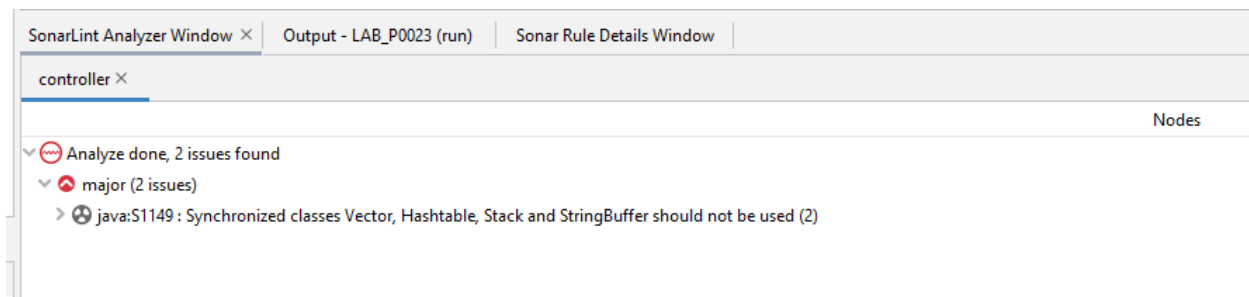
3. *java:S117 : Local variable and method parameter names should comply with a naming convention (1);* Level: minor

Solution: Rename to correct format

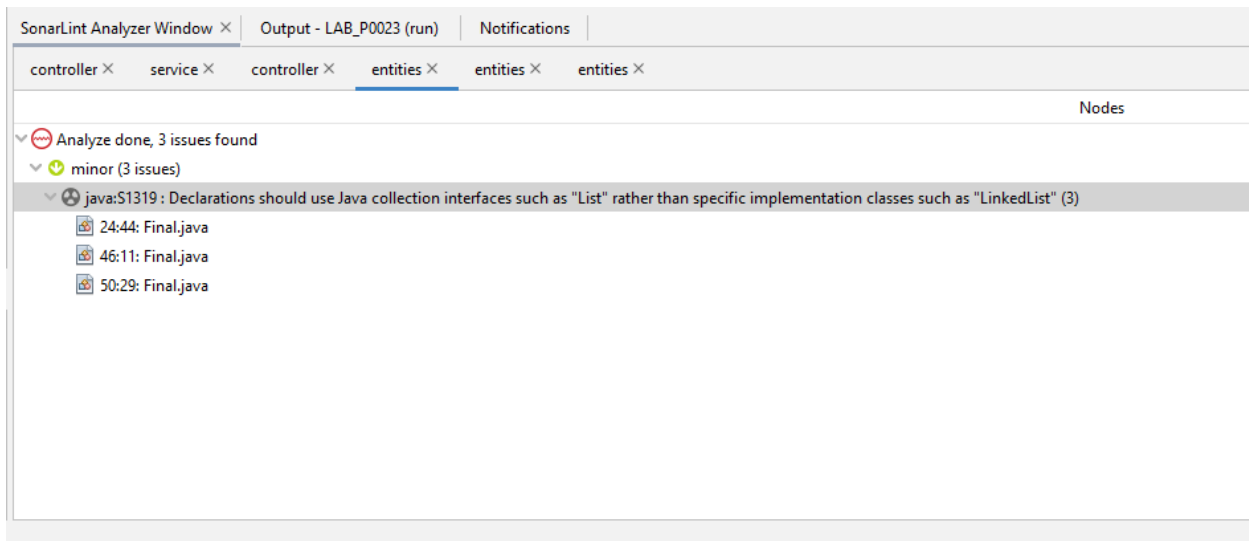
4. *java:S1149 : Synchronized classes Vector, Hashtable, Stack and StringBuffer should not be used (2)* Level: major

Haven't solution for this problem

After fixed:



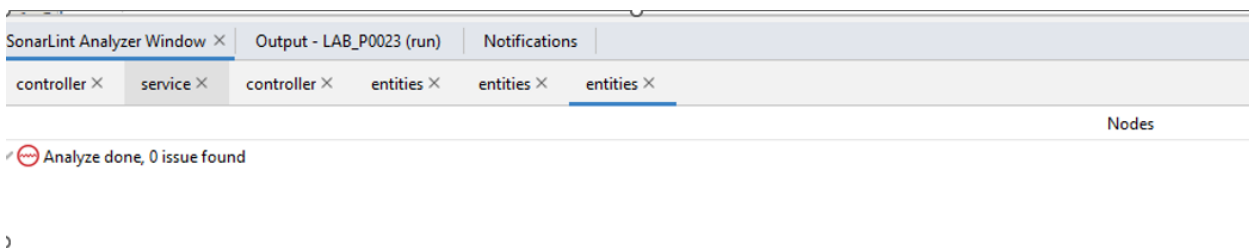
B. Folder “entities”:



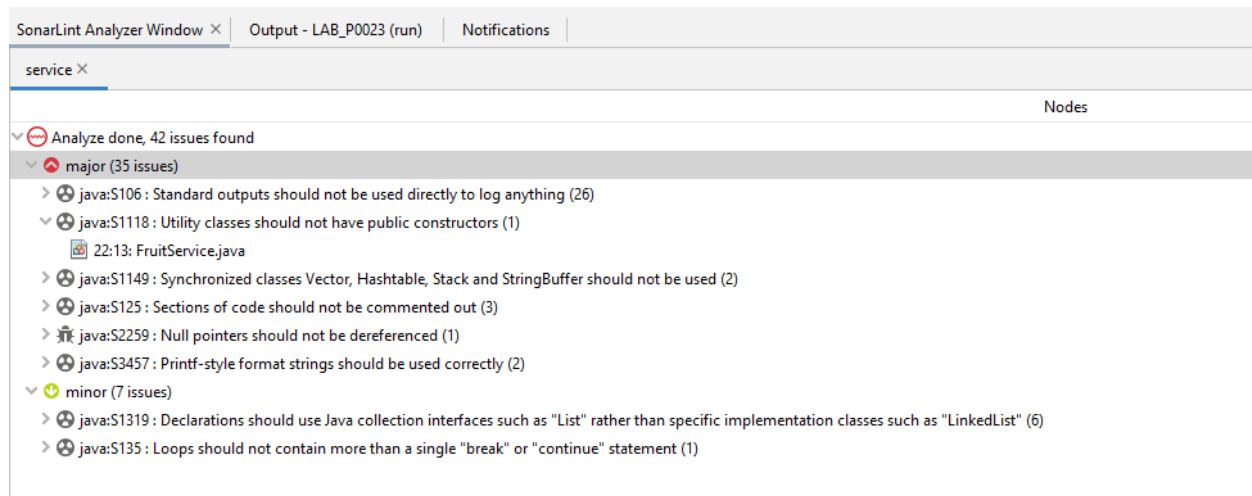
1. *java:S1319: Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList" (3)* Level: minor

Solution: Change “ArrayList” to “List”

After fixed:



C. Folder “service”:



SonarLint Analyzer Window × | Output - LAB_P0023 (run) | Notifications

service ×

Analyze done, 42 issues found

major (35 issues)

- java:S106 : Standard outputs should not be used directly to log anything (26)
- java:S1118 : Utility classes should not have public constructors (1)
 - 22:13: FruitService.java
- java:S1149 : Synchronized classes Vector, Hashtable, Stack and StringBuffer should not be used (2)
- java:S125 : Sections of code should not be commented out (3)
- java:S2259 : Null pointers should not be dereferenced (1)
- java:S3457 : Printf-style format strings should be used correctly (2)

minor (7 issues)

- java:S1319 : Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList" (6)
- java:S135 : Loops should not contain more than a single "break" or "continue" statement (1)

1. *java:S106 : Standard outputs should not be used directly to log anything (26)* Level: major

Don't need to fixed, because the requirement is display in the console

2. *java:S1118 : Utility classes should not have public constructors (1)* Level: major

Solution: set constructors to default

3. *java:S125 : Sections of code should not be commented out (3)* Level: major

Solution: Remove comment (don't have to)

4. *java:S3457 : Printf-style format strings should be used correctly (2)* Level: major

Solution: Fix it to correct format

5. *java:S135 : Loops should not contain more than a single "break" or "continue" statement (1)*

Haven't solution for this problem

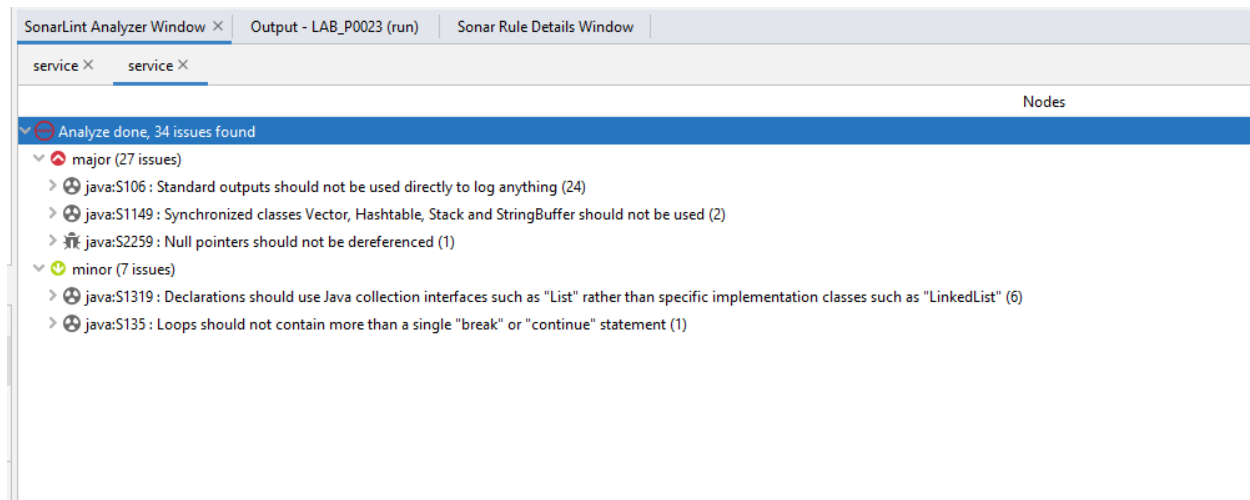
6. *java:S1319 : Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList" (6)* Level: minor

Haven't solution for this problem

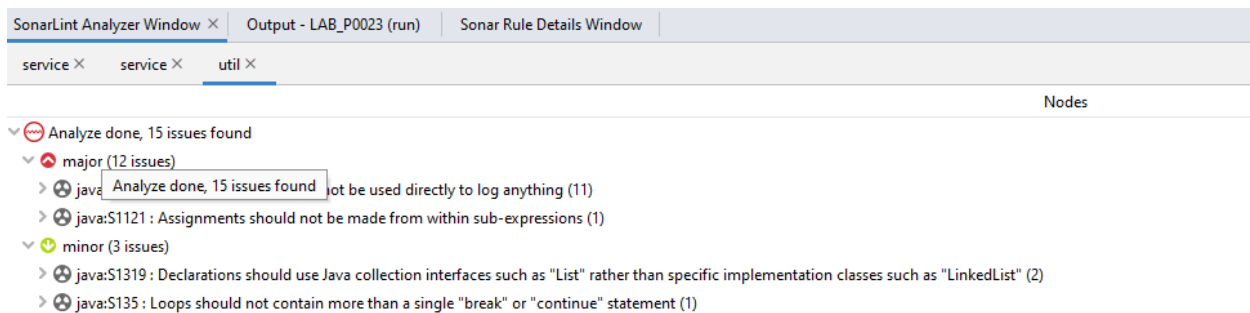
7. *java:S2259 : Null pointers should not be dereferenced (1)* Level: major

Haven't solution for this problem

After fixed:



D. Folder “util”:



1. *java:S135 : Loops should not contain more than a single "break" or "continue" statement (1)*

Level: minor

Solution: Remove "continue" statement

2. *java:S1121 : Assignments should not be made from within sub-expressions (1)* Level: major

Don't have to fix

3. *java:S1319 : Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList" (2)* Level: minor

Haven't solution for this problem

After fixed:

