

Team 5 - Code review report (Sergei Vilka, Mahnoor Fatima, Trung Vu)

Objective

As a team, we conducted a code review using CheckStyle and SonarQube to assess code quality, identify issues, and implement improvements. This report summarizes our findings and offers actionable recommendations to further enhance the project's maintainability, readability, and performance.

Summary

Tool Used	Findings
CheckStyle	1,622 issues found – 1597 resolved (~98%)
SonarQube	Rating: A (A in Security, Reliability, Maintainability)
Code Coverage	20.8% (Needs improvement)
Code Duplication	14.3% (Actionable area)

Common & Priority Issues

1. Missing Javadoc Comments

Severity: Low

Details: Classes, methods, and fields lack Javadoc comments, making the codebase less self-explanatory.

2. Naming Convention Violations

Severity: Low

Details: Variables such as name must match the camelCase pattern `^[a-z][a-zA-Z0-9]*$`.

3. Line Exceeds 80 Characters

Severity: Low

Details: Long lines affect readability. Consider breaking them down.

4. Class Design Concern

Severity: High

Details: Class Book seems designed for extension, but key methods like getBookId() lack documentation or proper safeguards.

Solution: Mark class final if not intended to be extended.

5. Final Parameters & Hidden Fields

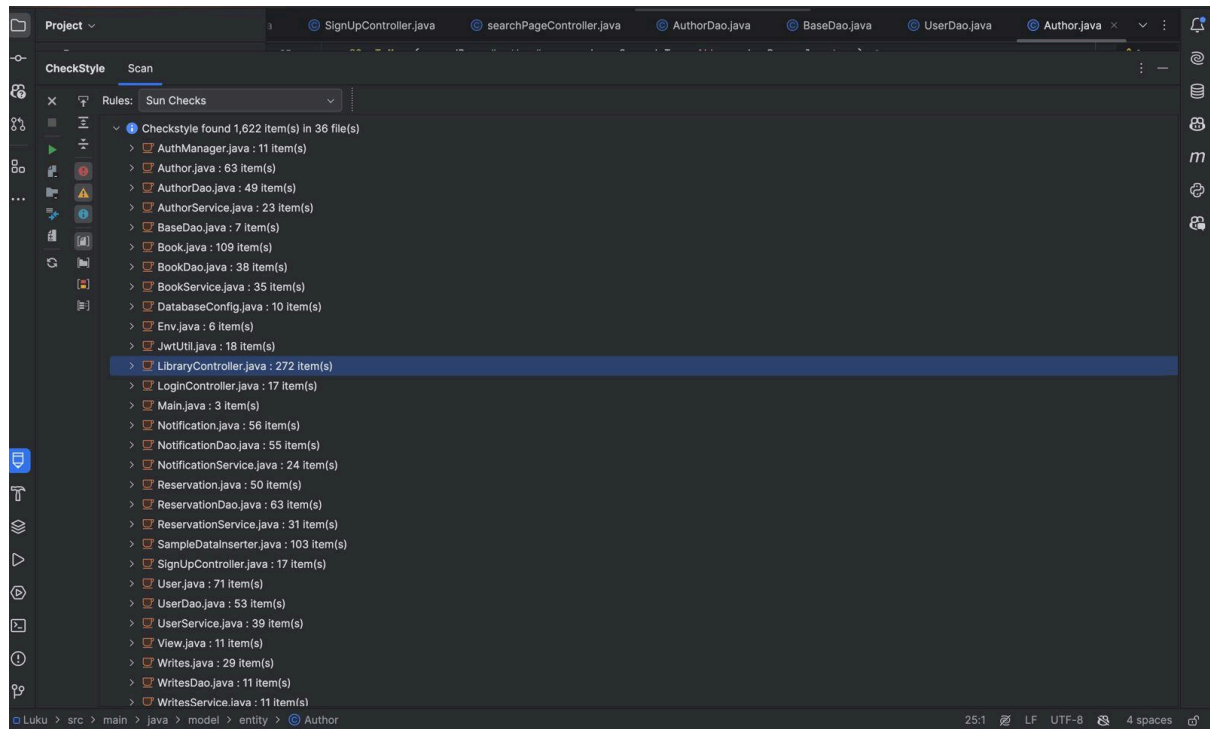
Severity: Medium

Details: Parameters such as bookId should be declared final. Also, local variables should not shadow fields.

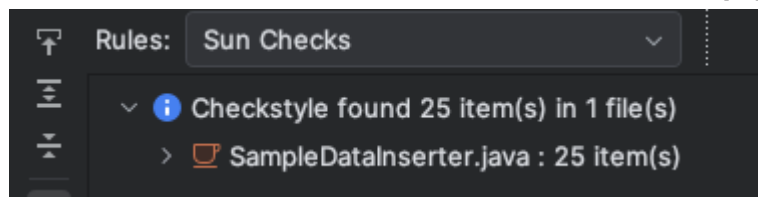
Examples:

```
! Using the '*.*' form of import should be avoided - model.entity.*. (7:20) [AvoidStarImport] <Sun Checks>
! Missing a Javadoc comment. (30:5) [JavadocVariable] <Sun Checks>
! Name 'View' must match pattern '^([a-z][a-zA-Z0-9]*$)'. (30:25) [StaticVariableName] <Sun Checks>
! Missing a Javadoc comment. (31:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (32:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (33:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (34:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (35:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (36:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (37:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (38:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (39:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (40:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (41:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (42:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (43:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (44:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (45:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (46:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (47:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (48:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (49:5) [JavadocVariable] <Sun Checks>
! Missing a Javadoc comment. (50:5) [JavadocVariable] <Sun Checks>
! Line is longer than 80 characters (found 84). (53:0) [LineLength] <Sun Checks>
! Each variable declaration must be in its own statement. (53:5) [MultipleVariableDeclarations] <Sun Checks>
! Missing a Javadoc comment. (53:5) [JavadocVariable] <Sun Checks>
! Each variable declaration must be in its own statement. (54:5) [MultipleVariableDeclarations] <Sun Checks>
! Missing a Javadoc comment. (54:5) [JavadocVariable] <Sun Checks>
! Line is longer than 80 characters (found 113). (55:0) [LineLength] <Sun Checks>
```

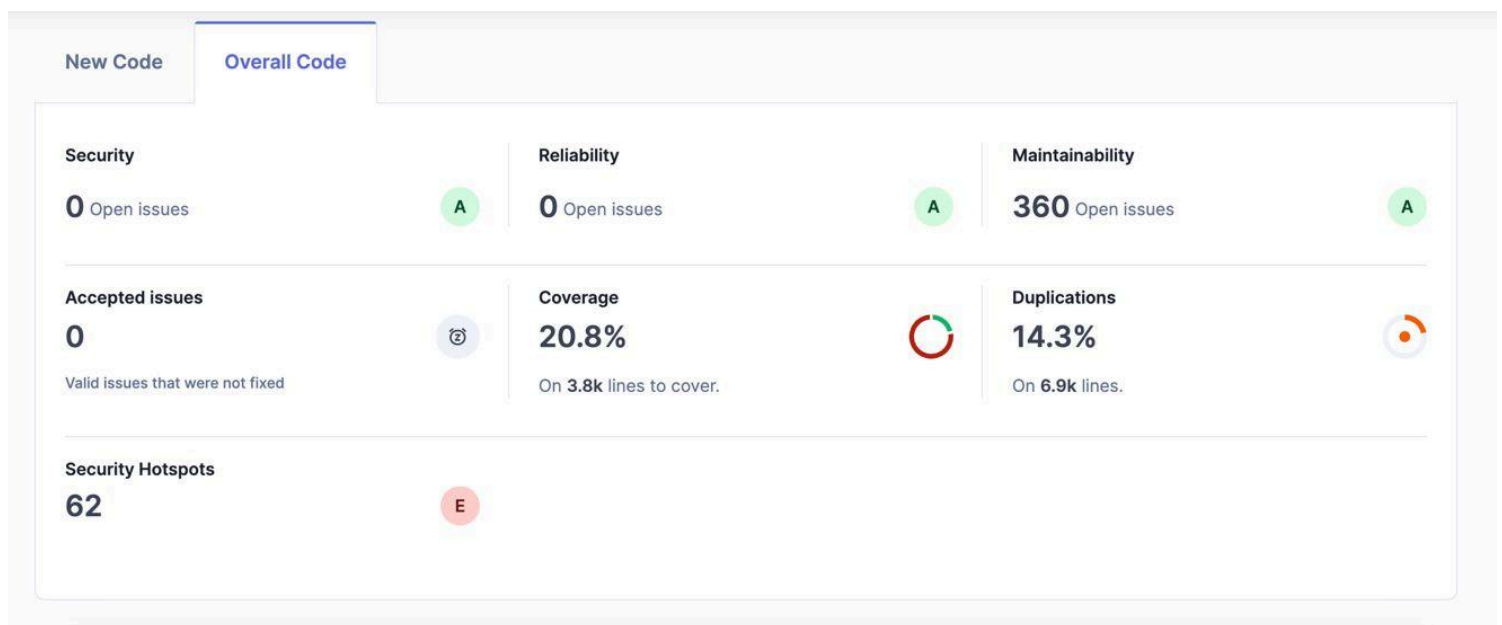
Initial amount of issues found by CheckStyle (1622):



Amount of issues after first round of code cleanup (25):



SonarQube report:



Short-Term Actions(completed):

- Add Javadoc comments for all public classes and methods.
- Refactor long lines for readability.
- Resolve hidden fields and enforce parameter immutability with final.
- Align all variable and method names with naming conventions.

Long-Term Improvements:

- Reduce code duplication by abstracting repeated logic.
- Set up CheckStyle and SonarQube as part of the CI/CD pipeline to enforce quality gates automatically.

Conclusion:

The codebase has undergone significant cleanup with 98% of static analysis issues resolved. The system is currently stable and functioning well, with a solid foundation for further improvement.