

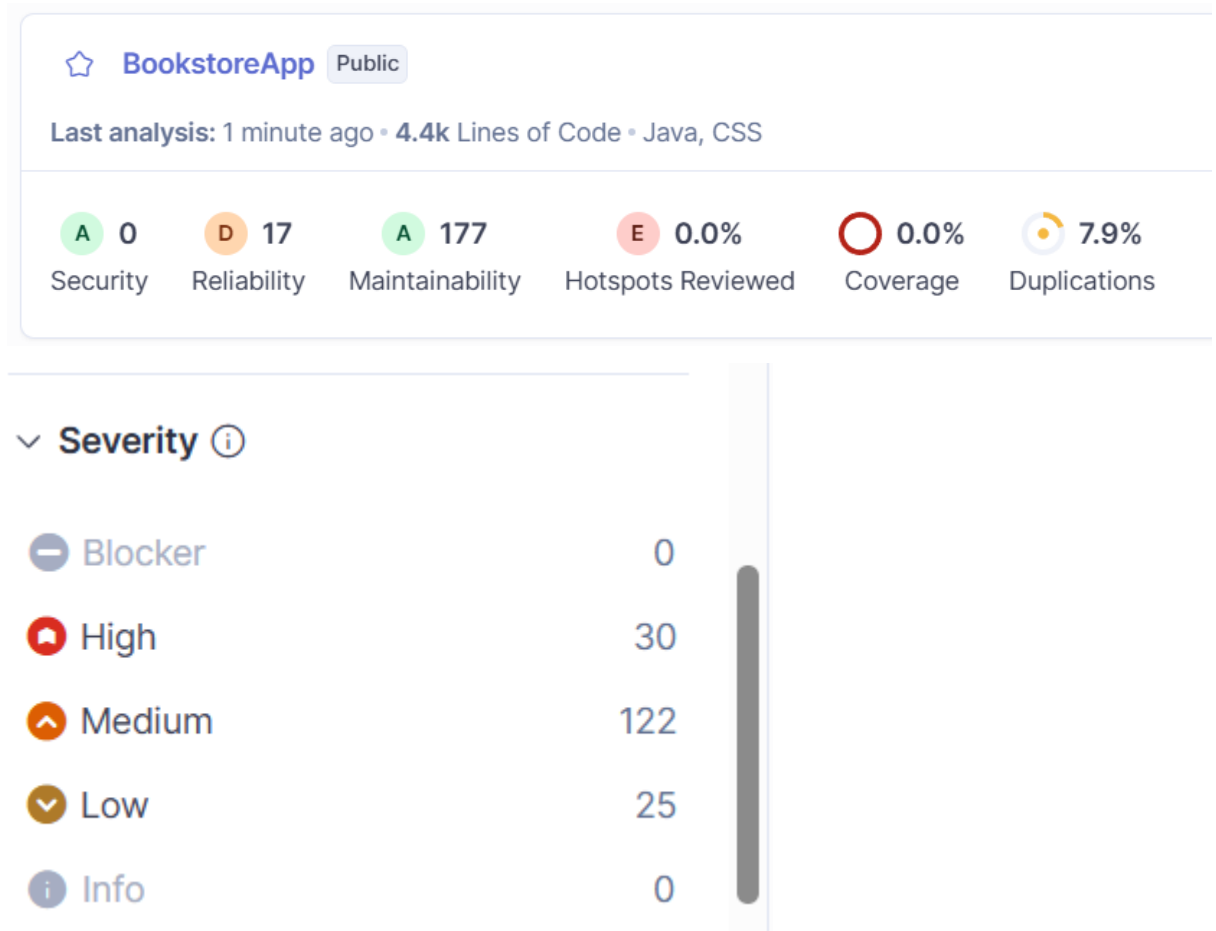
Code Review Findings

Introduction

In this code review and code clean-up/refactoring, SonarCube + SonarScanner and SpotBugs was used. Concentrating mainly around the code clean-up, after that SpotBugs was used to review the refactored code. The method worked well, since after using SonarCube, most of the time SpotBugs couldn't find any issues. We found that SpotBugs and SonarCube detected the same issues.

Code Clean-Up analysis

In the beginning of the clean-ups, we had over 200 **maintainability issues**. We managed to lower the results under 160, but it ended up increasing every time new code was written into the project.



Most of the alerts were **medium severity**, regarding the need to use “logger” instead of “System.out.print” or “System.err.println” commands. This is marked as code smell because console prints are hard to manage and they are not suitable for real applications. It is suggested to replace them with a logger (for example Java Logger), because it offers better control, filtering, formatting, and it is a standard way to handle outputs.

Another one that repeated, was the use of generic exceptions. Using too many generic exceptions makes it hard to understand what really went wrong. It is recommended to use more specific or custom exceptions.

src/main/java/com/group9/service/BookService.java

☐ [Replace this use of System.err by a logger.](#) Adaptability

Maintainability Medium bad-practice cert +

Open Not assigned L25 • 10min effort • 34 minutes ago

☐ [Replace generic exceptions with specific library exceptions or a custom exception.](#) Intentionality

Maintainability Medium cwe error-handling ... +

Open Not assigned L101 • 20min effort • 1 month ago

Where is the issue? Why is this an issue? How can I fix it? Activity More info

57 gerli...
58

veeti...

```
} else {  
    throw new RuntimeException("Book with ID " + rs.getInt(  
        "book_id") + " not found for order item ID " + rs.getInt("id"));
```

Replace generic exceptions with specific library exceptions or a custom exception.

High severity often indicated that there were unnecessary duplications or cognitive complexities, when one method contained too many if or try catch commands.

☐ [Define a constant instead of duplicating this literal "description" 3 times.](#) Adaptability

Maintainability High design +

Open Not assigned L43 • 8min effort • 1 month ago

Refactor this method to reduce its Cognitive Complexity from 16 to the 15 allowed. [🔗](#)

Cognitive Complexity of methods should not be too high [java:S3776](#) [🔗](#)

Line affected: L130 • Effort: 6min • Introduced: 3 days ago

Open ▾ Not assigned ▾ brain-overload ... +

Where is the issue? Why is this an issue? How can I fix it? Activity More info

```
140 // Gather translations
141 List<BookAttributeTranslation> translationsToSave = new ArrayList<>();
142 • for (String langCode : nameFields.keySet()) {
143   • if (langCode.equals("en")) continue; // Skip default language
144   String name = nameFields.get(langCode).getText().trim();
145   String desc = descFields.get(langCode).getText().trim();
146   • if (!name.isEmpty() • || !desc.isEmpty()) {
147     translationsToSave.add(new BookAttributeTranslation(langCode, name, desc));
148   }
```

Uncovered code

Software qualities impacted

Maintainability 🚫 High ▾

Code attribute

Adaptability | Not focused

Suggestion for code duplications, was to use **constants** instead.

Define a constant instead of duplicating this literal "unknownAttributeTypeError" 3 times. [🔗](#)

String literals should not be duplicated [java:S1192](#) [🔗](#)

Line affected: L110 • Effort: 8min • Introduced: 3 days ago

Open ▾ Not assigned ▾ design +

Where is the issue? Why is this an issue? How can I fix it? Activity

```
private static final String ACTION_1 = "action1"; // Compliant

public void run() {
  prepare(ACTION_1); // Compliant
  execute(ACTION_1);
  release(ACTION_1);
}
```

Software qualities impacted

Maintainability 🚫 High ▾

Code attribute

Adaptability | Not distinct

Low severity issues were mainly about used method-, field names, or constants not being static final.

☐ **Rename this field "nameTextField_ja" to match the regular expression `^[a-z][a-zA-Z0-9]*$`.** [🔗](#) Consistency

Maintainability 🟡 Low ▾

convention +

Open ▾ Not assigned ▾ L38 • 2min effort • 4 days ago

☐ **Rename this field "nameTextField_ar" to match the regular expression `^[a-z][a-zA-Z0-9]*$`.** [🔗](#) Consistency

Maintainability 🟡 Low ▾

convention +

Open ▾ Not assigned ▾ L39 • 2min effort • 4 days ago

☐ **Rename this field "descTextField_en" to match the regular expression `^[a-z][a-zA-Z0-9]*$`.** [🔗](#) Consistency

Maintainability 🟡 Low ▾

convention +

Open ▾ Not assigned ▾ L41 • 2min effort • 4 days ago

<input type="checkbox"/>	Make languageCode a static final constant or non-public and provide accessors if needed.	Adaptability
Maintainability	Low	cwe +
Open	Not assigned	L4 • 10min effort • 3 days ago
<input type="checkbox"/>	Make translatedName a static final constant or non-public and provide accessors if needed.	Adaptability
Maintainability	Low	cwe +
Open	Not assigned	L5 • 10min effort • 3 days ago
<input type="checkbox"/>	Make translatedDescription a static final constant or non-public and provide accessors if needed.	Adaptability
Maintainability	Low	cwe +
Open	Not assigned	L6 • 10min effort • 3 days ago