Deliverable D03-D04: SonarLint Report

-

Group E8.02

## 

Github Repository: <https://github.com/JStockwell/Acme-One-E8.02>

Members:

* Gregorio Ortega Soldado ([greortsol@alum.us.es](mailto:greortsol@alum.us.es))
* Alejandro Manuel Gestoso Torres ([alegestor@alum.us.es](mailto:alegestor@alum.us.es))
* Jaime Stockwell Mendoza ([jaistomen@alum.us.es](mailto:jaistomen@alum.us.es))
* Pablo Aurelio Sánchez Valenzuela ([pabsanval1@alum.us.es](mailto:pabsanval1@alum.us.es))
* Manuel Cabra Morón ([mancabmor1@alum.us.es](mailto:mancabmor1@alum.us.es))
* Fernando Claros ([ferclabar@alum.us.es](mailto:ferclabar@alum.us.es))

## Fecha: 01/03/2022

# 

# Table of Contents

| 1. Executive Summary 2. Revision Table 3. Introduction 4. Contents 5. Conclusions | 2  2  3  3  5 |
| --- | --- |

# 

# Executive Summary

We are a group of 6 java developers, one of us is also the project manager of the team. We all are studying 3rd year in a software engineering degree and we have experience in projects due to other subjects like AISS, DP1, PSG1, IISSI1 and IISSI2.

Our product is a web information system which uses java technology, an IDE like Eclipse and 2 programs to interact with databases like MariaDB and DBeaver.

Our procedure is using github to establish a common cloud and a control version using branches that will commit to the main code and kanban boards to organize the work. We also use Scrum methodology to organize ourselves as a team.

# Revision Table

| Revision | Description | Date |
| --- | --- | --- |
| v1.0 | Initial Creation | 17/04/2022 |
| v1.1 | D03 Report | 25/04/2022 |
| v1.2 | D04 Report | 2/06/2022 |

# 

# Introduction

Ensuring all the functionalities of the code of a project is a fundamental process of developing a software system. It is as important as the code itself that every single piece of code is thoroughly tested by the time of making an important delivery to the client. It is also important in a group project that all the members of the development team know where miswritten code is failing, even if it’s not made by him/her.

For these reasons we have come up with a plan to maintain the code tested at every moment, which we will detail in the following point.

# Contents

D03 SonarLint Report

The results of the SonarLint report were a total of 34 code smells and bugs:

* 2 major bugs
* 6 major code smells
* 21 minor bugs
* 1 minor code smell
* 4 Information code smells

### Major Bugs

The two major bugs are found in AnyItemShowService.java and InventorItemShowService.java. In both cases, at the end of the authorise function, there is a call to an item that can be nullable, without there being a possible NullPointerException try catch.

### Major Code Smells

Three of the major code smells can be summed up into one big code smell: our non-conventional enumeration for the Status.java enum. To fix these code smells, we should make sure that all text inside of an enum is all caps, following this regex: “*^[A-Z][A-Z0-9]\*(\_[A-Z0-9]+)\*$”.* The other 3 code smells can also be grouped up into one big code smell, which are duplicate characters in some of the regex in SystemConfiguration.java.

### Minor Bugs

All minor bugs come from every form.jsp in the application, as we do not add descriptions to the tables we create.

### Minor Code Smells

The only minor code smell is that the *“getCode”* function in the class PatronageReport.java should be replaced with a StringBuilder.

### Information Code Smells

Three information code smells refer to TODOs placed inside the class Patronage.java, to indicate future tasks that cannot be completed as of now, as well as a new TODO in PatronageReport, to remind the team to solve the only remaining minor code smell.

D04 SonarLint Report

The results of the SonarLint report were a total of 44 code smells and bugs:

* 3 major bugs
* 6 major code smells

### Major Bugs

The three major bugs are in SystemConfiguration.java, where the regex expressions used may lead to a stack overflow for large inputs.

### Major Code Smells

Four of the major code smells can be summed up into one big code smell: our non-conventional enumeration for the Status.java enum. To fix these code smells, we should make sure that all text inside of an enum is all caps, following this regex: “*^[A-Z][A-Z0-9]\*(\_[A-Z0-9]+)\*$”.* The other 2 code smells are boolean expressions that should always return true in InventorItemPublishService.java and InventorItemUpdateService.java

# Conclusion

SonarLint has been a very helpful tool that has aided the tracking of work that needed to be done, as it can highlight TODO comments. Furthermore, it is a great way to spot mistakes, correct simple redundant code and improve the overall quality of the final code. It will definitely become an indispensable part of all of our future projects.