UNIVERSITÉ ABDELHAMIDHRI CONSTANTINE 2
FACULTÉ DES NOUVELLES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION NTIC
LE DÉPARTEMENT TECHNOLOGIES DES LOGICIELS ET DES SYSTÈMES D'INFORMATION TLSI

TP MEL
rapport 3

SPECIALTY: SOFTWARE ENGINEERING

Maintenance of Hotel Project (overView)

Supervised by:
• Dr. Manel DJENOUHAT

Presented by:
• Rouissa Rabah G2
• Belmokhi Dibadj G2
• Brahmia Mohamed Haythem Abderrahmen G2

DATE: 14/04/2025

Contents

List of Tables

List of Figures

1 Introduction

1.1 Project Context

The Hotel Management Desktop Application is designed to provide an efficient and reliable solution for managing hotel operations. It offers features for handling bookings, managing customer data, tracking reservations, and processing check-ins and check-outs. The application prioritizes user experience with a clean, intuitive interface.Built with multi-threading capabilities, it ensures responsive performance even during peak usage. Comprehensive error handling is integrated throughout to maintain stability and prevent data loss or system crashes.

1.2 Maintenance Objectives

Ongoing maintenance of the Hotel Management Application focuses on the following key areas:

1.2.1 Improving Code Readability

The codebase will be reviewed to ensure that logic is clear and variable names are meaningful. Consistent naming practices and well-structured code will help future developers understand and work with the project more effectively.

1.2.2 Removing Code Redundancy

All duplicate code blocks will be identified and refactored to reduce repetition. This will simplify maintenance, lower the risk of bugs, and improve overall code organization.

1.2.3 Enhancing Code Efficiency

Unnecessary or overly complex lines of code will be optimized or removed. Where possible, functions and components will be simplified into more modular forms, making the system easier to update, extend, and debug.

1.3 Tools and Methodology

To ensure the ongoing quality and maintainability of the Hotel Management Application, the following tools and practices will be adopted:

1.3.1 SonarCloud

SonarCloud will be used to perform regular static code analysis. It will help in detecting code smells, potential bugs, and overly complex code structures. By addressing these issues promptly, the codebase will remain clean, maintainable, and easier for future developers to work with.

1.3.2 GitHub

Version control will be handled through GitHub, providing a centralized platform for tracking changes, managing branches, and collaborating on updates. This ensures that all development activity is traceable and well-documented, reducing the risk of conflicts or data loss.By continuously improving code quality and keeping the system free from unnecessary complexity, the Hotel Management Application will remain robust, easy to maintain, and efficient, without compromising its core features.

2 Analysis of Current State

2.1 SonarQube Metrics Overview

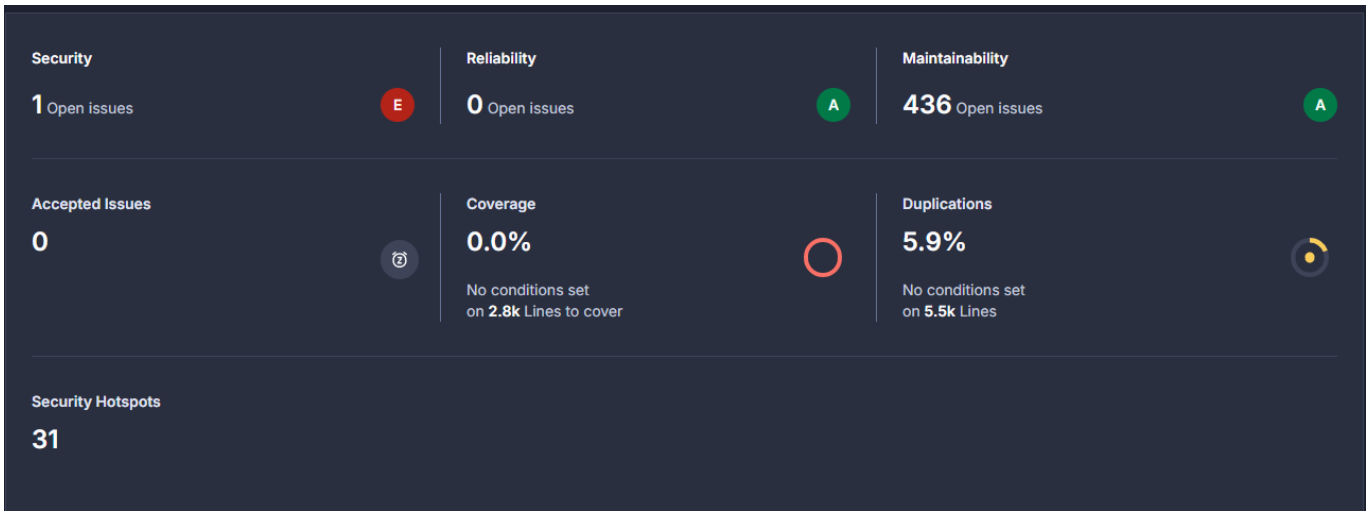

Figure 1: SonarQube Metrics Overviewe

3 Identified Issues

3.1 Security Issues

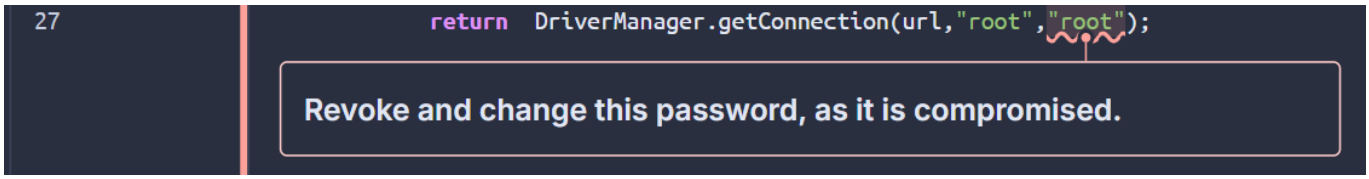

Figure 2: Security Issues

Figure 3: Security Issue - Issue Solution

3.2 Maintainability Issues

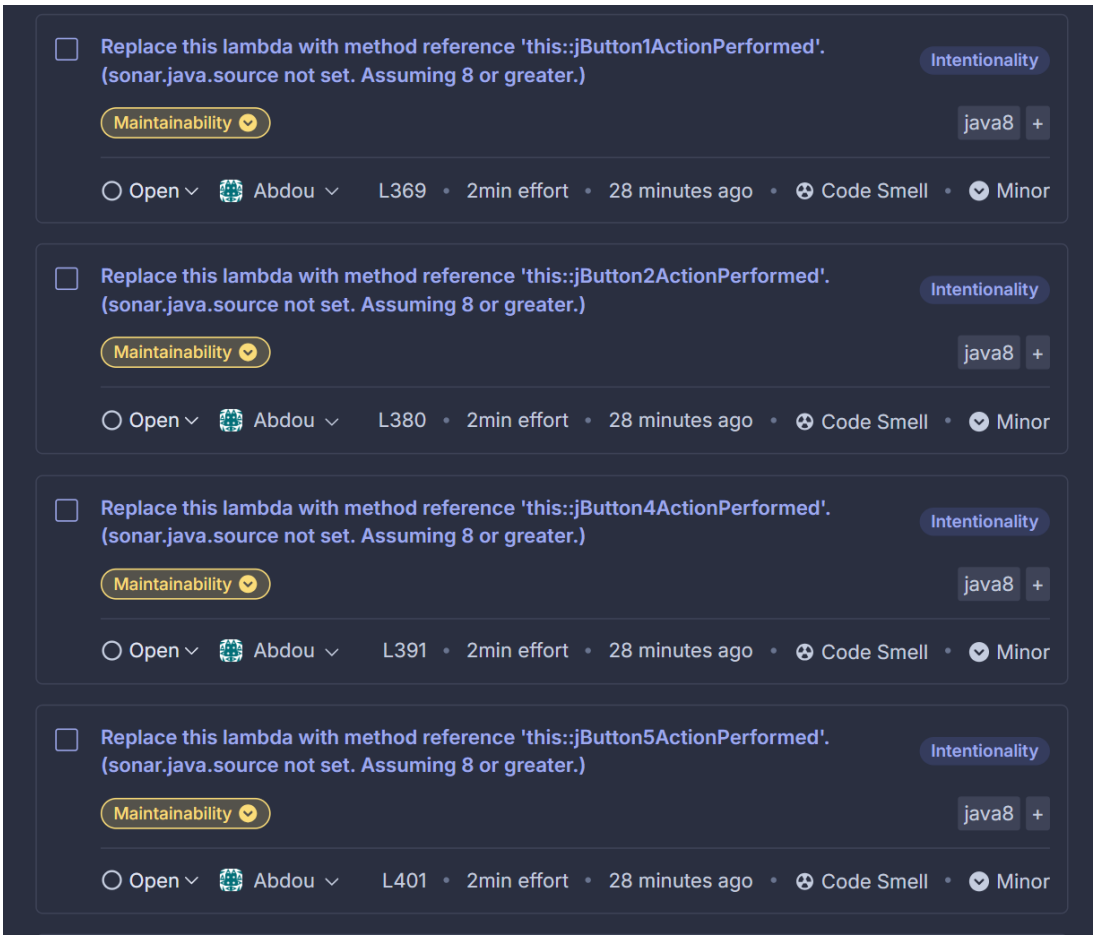

Figure 4: Maintainability Issues Overview

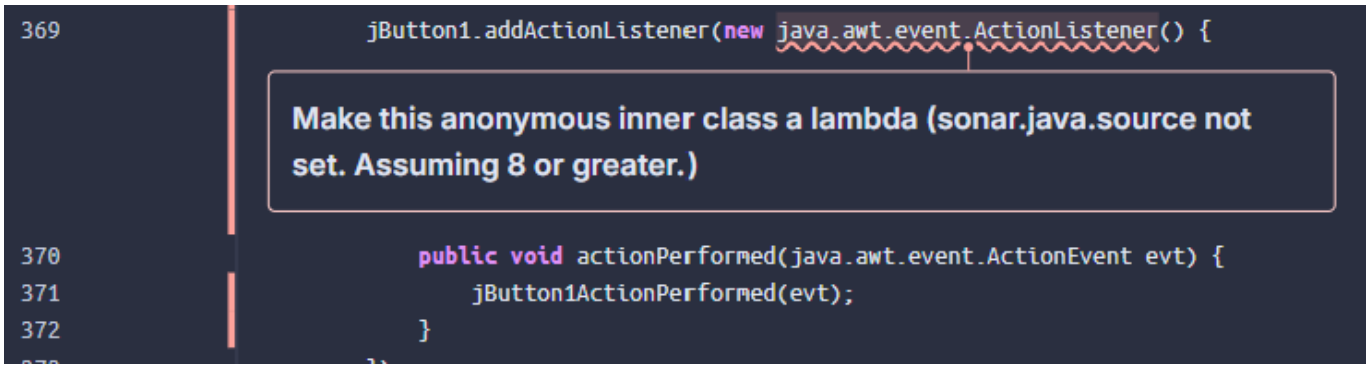

Figure 5: Maintainability Issues - First Issue

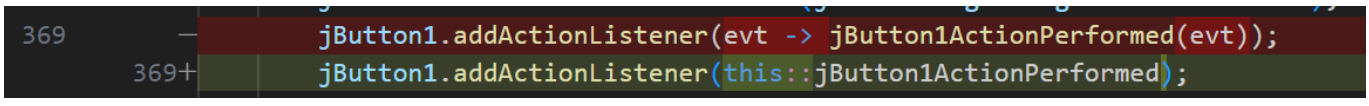

Figure 6: Maintainability Issues - First Issue Solution

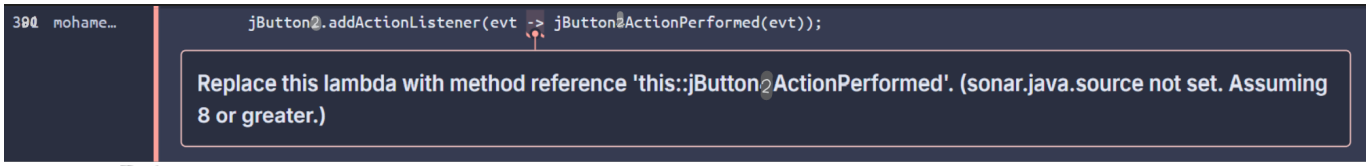

Figure 7: Maintainability Issues - Second Issue

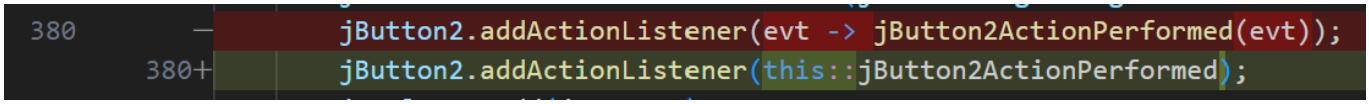

Figure 8: Maintainability Issues - Second Issue Solution

Figure 9: Maintainability Issues - Fourth Issue

Figure 10: Maintainability Issues - Fourth Issue Solution

Figure 11: Maintainability Issues - Fifth Issue

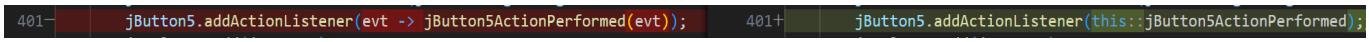

Figure 12: Maintainability Issues - Fifth Issue Solution

Figure 13: Maintainability Issues - Sixth Issue

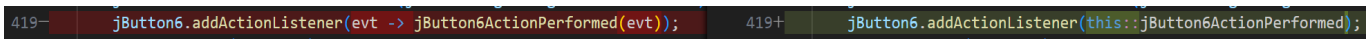

Figure 14: Maintainability Issues - Sixth Issue Solution

Figure 15: Maintainability Issues - Seventh Issue

Figure 16: Maintainability Issues - Seventh Issue Solution

4 Results and Verification

4.1 SonarQube Results After Maintenance

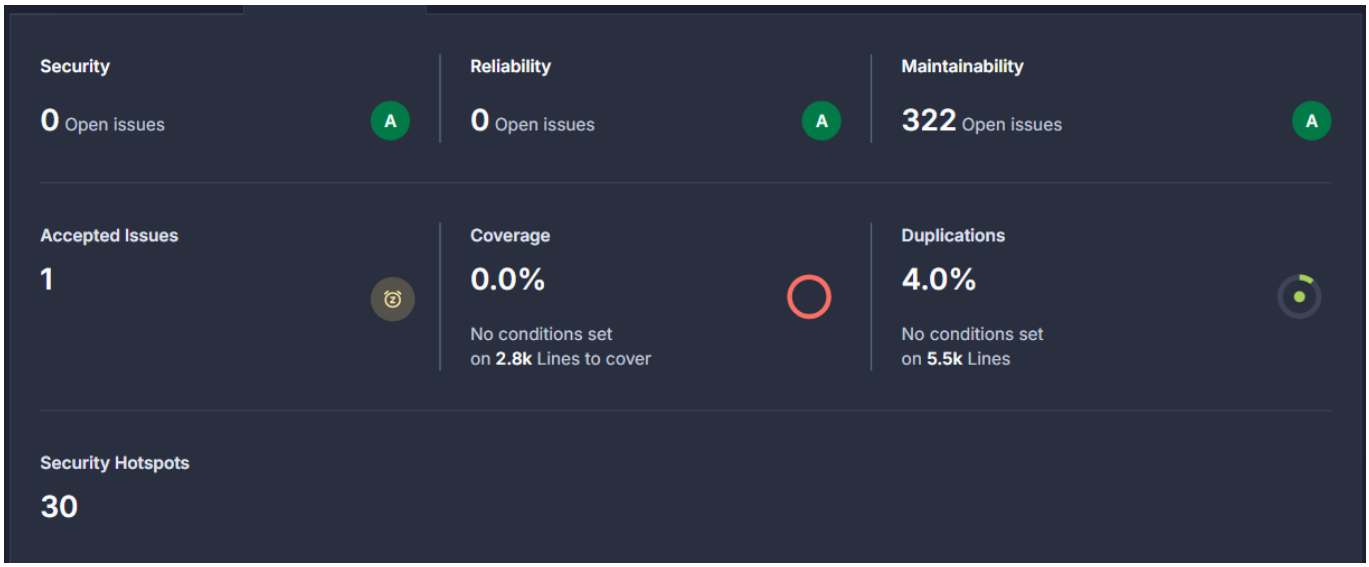

Figure 17: SonarQube Metrics Overview After Maintenance

Metric	Before	After
Security Issues	1	0
Reliability Issues	0	0
Maintainability Issues	436	322
Accepted Issues	0	1
Coverage	Not configured	Not configured
Duplications	5.9%	4.0%
Security Hotspots	31	30

Table 1: SonarQube Analysis Results - Before vs. After