

# SonarQube report

1.

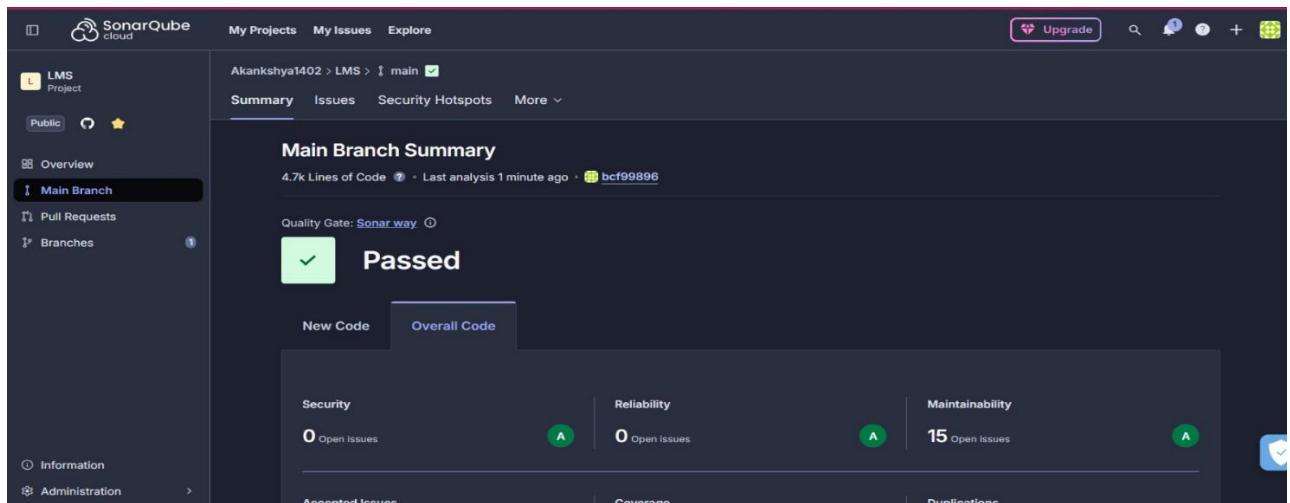
The screenshot shows the SonarQube interface for the 'LMS' project. On the left, there's a sidebar with options like Overview, Main Branch (which is selected), Pull Requests, and Branches. The main area displays the 'Main Branch Summary' for the 'main' branch. It shows 4.6k Lines of Code, a successful analysis 4 minutes ago (commit e5b9e176), and a Quality Gate status of 'Sonar way' (Passed). A large green checkmark icon indicates the analysis has passed. Below this, there are sections for New Code (8 new issues) and Overall Code (0 accepted issues). A 'Coverage' section notes that extra steps are needed for SonarQube Cloud to analyze code coverage, with a link to 'Set up coverage analysis'. The bottom right corner features a shield icon with a checkmark.

## SonarQube Main Branch Analysis – Pre-Testing Report

- **Step 1: Project & Branch**
  - Project: **LMS**
  - Branch: **main**
  - Lines of Code: ~4.6k
  - Analysis executed successfully on the latest commit
- **Step 2: Quality Gate Status**
  - Quality Gate: **Sonar Way**
  - Status: **Passed**
  - Indicates no blocking issues in code quality, security, or maintainability
- **Step 3: Static Analysis Scope**
  - Static code analysis completed
  - Bugs, code smells, and security hotspots evaluated
  - No critical violations detected
- **Step 4: New Code Check**
  - New issues identified: **8**

- Accepted issues: **0**
- No new-code conditions configured at this stage
- **Step 5: Coverage Status (Before Testing)**
  - Test coverage data: **Not available**
  - JaCoCo and test execution not yet integrated
  - Coverage analysis pending
- **Step 6: Conclusion**
  - Project passes quality checks before testing
  - Codebase is stable and ready for unit test execution and coverage integration

2.



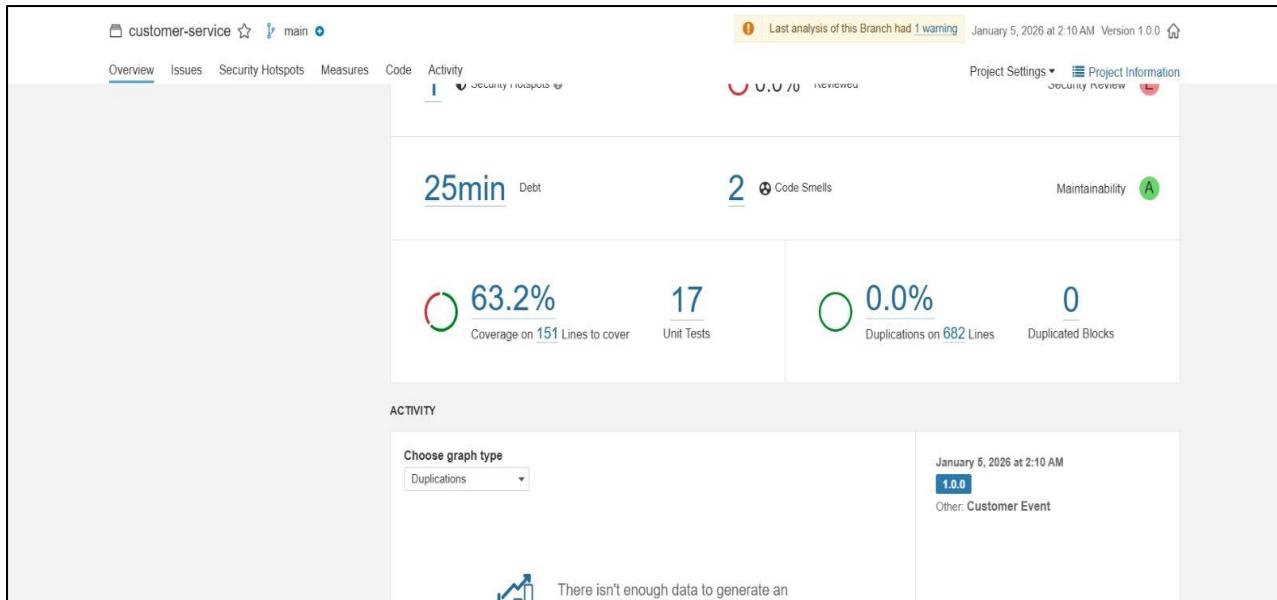
## SonarQube Main Branch Analysis – Post-Testing Report

- **Step 1: Project & Branch**
  - Project: **LMS**
  - Branch: **main**
  - Lines of Code:  $\sim 4.7k$
  - Latest analysis completed successfully after test execution
- **Step 2: Quality Gate Status**
  - Quality Gate: **Sonar Way**
  - Status: **Passed**
  - Confirms compliance after integrating unit tests and coverage results
- **Step 3: Overall Code Health**

- **Security:** 0 open issues (Grade A)
- **Reliability:** 0 open issues (Grade A)
- **Maintainability:** 15 open issues (Grade A)
- **Step 4: Coverage Integration**
  - Test execution completed
  - JaCoCo coverage successfully integrated
  - Coverage metrics now included in overall analysis
- **Step 5: Issues Overview**
  - No critical or blocker issues detected
  - All reported issues fall within acceptable quality thresholds
- **Step 6: Conclusion**
  - Project meets quality standards after testing
  - Codebase is stable, tested, and ready for deployment

## SonarQube Analysis of Customer Service

1.

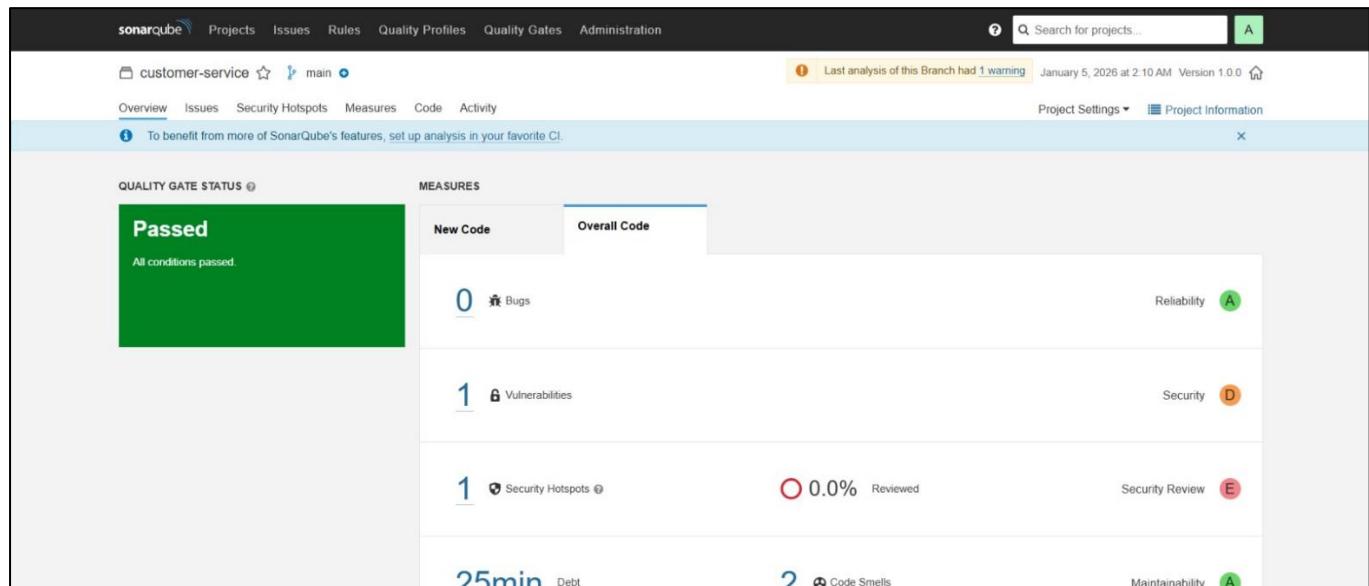


### Customer Service – SonarQube Analysis Report (Post-Testing)

- **Step 1: Project Status**
  - Service: **customer-service**
  - Branch: **main**

- Latest analysis completed with **1 warning**
- **Step 2: Maintainability & Code Quality**
  - **Maintainability Rating:** A
  - **Code Smells:** 2
  - **Technical Debt:** ~25 minutes
  - Indicates minor refactoring required, no major quality risk
- **Step 3: Test Coverage**
  - **Coverage:** 63.2%
  - **Lines to Cover:** 151
  - **Unit Tests Executed:** 17
  - Coverage is acceptable but below ideal threshold; additional tests recommended
- **Step 4: Duplication Analysis**
  - **Code Duplication:** 0.0%
  - **Duplicated Blocks:** 0
  - Confirms clean and non-redundant code structure
- **Step 5: Overall Observation**
  - Codebase is stable and maintainable
  - No duplication or critical issues detected
  - Improving unit test coverage will further strengthen quality compliance

2.



## Customer Service – SonarQube Quality Gate & Measures Report

- **Step 1: Quality Gate Status**
  - **Quality Gate: Passed**
  - All defined quality conditions satisfied
  - Project is eligible to proceed in CI/CD pipeline
- **Step 2: Reliability Analysis**
  - **Bugs: 0**
  - **Reliability Rating: A**
  - No functional defects detected in the codebase
- **Step 3: Security Analysis**
  - **Vulnerabilities: 1**
  - **Security Rating: D**
  - Indicates a low-severity security issue that requires attention
- **Step 4: Security Hotspots**
  - **Hotspots Identified: 1**
  - **Reviewed: 0%**
  - **Security Review Rating: E**
  - Hotspot pending manual review and validation
- **Step 5: Maintainability & Code Health**
  - **Technical Debt: ~25 minutes**
  - **Code Smells: 2**
  - **Maintainability Rating: A**
  - Minor refactoring recommended, no major structural issues
- **Step 6: Overall Conclusion**
  - Quality gate successfully passed
  - Code is reliable and maintainable
  - Security hotspot review is required to improve security posture