

SonarQube Integration Project Submission

(Address Book Module)

1) Project Shared

I have integrated SonarQube Community Build with my Java project:

- ProjectName:** AddressBookSystem
- Module Included:** Address Book Module
- Type:** Core Java + Eclipse Project
- Source Folder:** src
- Compiled Output Folder:** bin

2) Screenshots Provided

I have attached screenshots of:

A) Project Execution

Output of AddressBookSystem running successfully in Eclipse / Console

```
AddressBookApp (1) [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-0741\

Welcome to Address Book

Address Book System
1. Create Address Book
2. Open Address Book
3. Show All Address Books
4. Search Person by City/State
5. Exit
Enter choice:
```

B) SonarQube Report

- SonarQube Dashboard showing:
 - Quality Gate Status: **Passed**
 - Maintainability Issues
 - Coverage and Duplication Summary

Address Book / main ✓

Overview Issues Security Hotspots Code Measures Activity Project Settings Project Information

main

358 Lines of Code - Version not provided - Set as homepage Last analysis 32 minutes ago

Passed

The last analysis has warnings. See details

New Code Overall Code

Security 0 Open issues	Reliability 0 Open issues	Maintainability 53 Open issues
Accepted issues 0	Coverage 0.0% On 225 lines to cover.	Duplications 6.7% On 504 lines.
Security Hotspots 0	A	

Activity

Graph type: Issues

— Issues □ New Code

February 8, 2026 at 9:26 PM
not provided

Quality Gate: Passed

New analysis: +0 Issues

February 8, 2026 at 8:49 PM
First analysis: 53 Issues • 0.0% Coverage • 6.7% Duplications

Quality Gate: Passed

[See full history of analyses](#)

C) Sonar Scanner Execution

- Command Prompt output showing:
 - **ANALYSIS SUCCESSFUL**
 - Dashboard URL
(<http://localhost:9000/dashboard?id=address-book&codeScope=overall>)

```
C:\Windows\System32\cmd.e + - X
21:26:44.172 INFO 6/6 source files have been analyzed for the text and secrets analysis
21:26:44.174 INFO Sensor TextAndSecretsSensor [text] (done) | time=484ms
21:26:44.179 INFO ----- Run sensors on project
21:26:44.318 INFO Sensor JavaProjectSensor [java]
21:26:44.322 INFO Sensor JavaProjectSensor [java] (done) | time=5ms
21:26:44.326 INFO Sensor Zero Coverage Sensor
21:26:44.340 INFO Sensor Zero Coverage Sensor (done) | time=17ms
21:26:44.341 INFO Sensor Java CPD Block Indexer
21:26:44.374 INFO Sensor Java CPD Block Indexer (done) | time=33ms
21:26:44.375 INFO ----- Gather SCA dependencies on project
21:26:44.382 INFO Dependency analysis skipped
21:26:44.385 INFO SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
21:26:44.391 INFO CPD Executor 1 file had no CPD blocks
21:26:44.392 INFO CPD Executor Calculating CPD for 5 files
21:26:44.422 INFO CPD Executor CPD calculation finished (done) | time=30ms
21:26:44.551 INFO Analysis report generated in 110ms, dir size=289.8 kB
21:26:45.019 INFO Analysis report compressed in 467ms, zip size=43.5 kB
21:26:45.154 INFO Analysis report uploaded in 134ms
21:26:45.156 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=address-book1
21:26:45.157 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:26:45.158 INFO More about the report processing at http://localhost:9000/api/ce/task?id=1c2097eb-983a-4d0c-a8a6-746ec70152e0
21:26:45.185 INFO Analysis total time: 6.460 s
21:26:45.186 INFO SonarScanner Engine completed successfully
21:26:45.290 INFO EXECUTION SUCCESS
21:26:45.291 INFO Total time: 12.678s
C:\Users\Lenovo\eclipse-workspace\Address-Book>
```

3) Step-by-Step Procedure Followed (SonarQube Integration)

Step 1: Install SonarQube

1. Download SonarQube Community Build for Windows
2. Extract it to a folder (example): **C:\sonarqube-26.2.0**
3. Open Command Prompt as Administrator
4. Go to: **sonarqube\bin\windows-x86-64**
5. Run: **StartSonar.bat**

Step 2: Verify SonarQube is Running

1. Open browser: **<http://localhost:9000>**
2. Login using default credentials:
 - Username: **admin**
 - Password: **admin**
3. Set new password after login

Step 3: Create a New Project in SonarQube

1. Click **Create Project**
2. Enter Project Key: **address-book**
3. Generate a token (used for scanner authentication)

Step 4: Download SonarScanner

1. Download SonarScanner CLI for Windows
2. Extract it (example): **C:\sonar-scanner-8.0.1**
3. Add the bin folder to system PATH: **C:\sonar-scanner-8.0.1\bin**

Step 5: Compile the Java Project

SonarQube needs .class files for Java analysis.

In Eclipse project folder, compile classes into bin.

(For Eclipse projects, bin is created automatically when project builds.)

Step 6: Run Sonar Scanner

Open Command Prompt inside the project folder:

Example: **C:\Users\...\eclipse-workspace\AddressBookSystem>**

Run:

```
sonar-scanner.bat -D"sonar.projectKey=address-book" -D"sonar.sources=src"  
-D"sonar.java.binaries=bin"      -D"sonar.host.url=http://localhost:9000"      -  
D"sonar.token=TOKEN"
```

Step 7: View the SonarQube Report

After execution, open: <http://localhost:9000/dashboard?id=address-book>

This displays:

- o Code Smells
- o Bugs
- o Vulnerabilities
- o Quality Gate Status

4) Methodology / Approach Used (Classroom Method)

Approach Followed

I followed the same approach used in the classroom:

Manual SonarQube Integration using SonarScanner CLI

This method includes:

1. Running SonarQube Server locally
2. Creating project in SonarQube UI
3. Generating token
4. Running SonarScanner command from project folder
5. Viewing analysis report in SonarQube dashboard

Note about Different Approaches

Different approaches can also be used to integrate SonarQube, such as:

- SonarQube integration using **Maven**
- SonarQube integration using **Gradle**
- SonarQube integration using **Jenkins CI/CD**
- SonarLint plugin in Eclipse for live analysis

But in this project, I used the **classroom method: SonarScanner CLI**.

5) SonarQube Analysis Result Summary

From the report:

 **Quality Gate: PASSED**

 **Security Issues: 0**

 **Reliability Issues: 0**

 **Maintainability Issues: 53 Code Smells**

 **Coverage: 0% (JUnit not added)**

 **Duplications: 6.7%**