

SonarQube Integration Project Submission

(Address Book Module)

1) Project Shared

I have integrated SonarQube Community Build with my Java project:

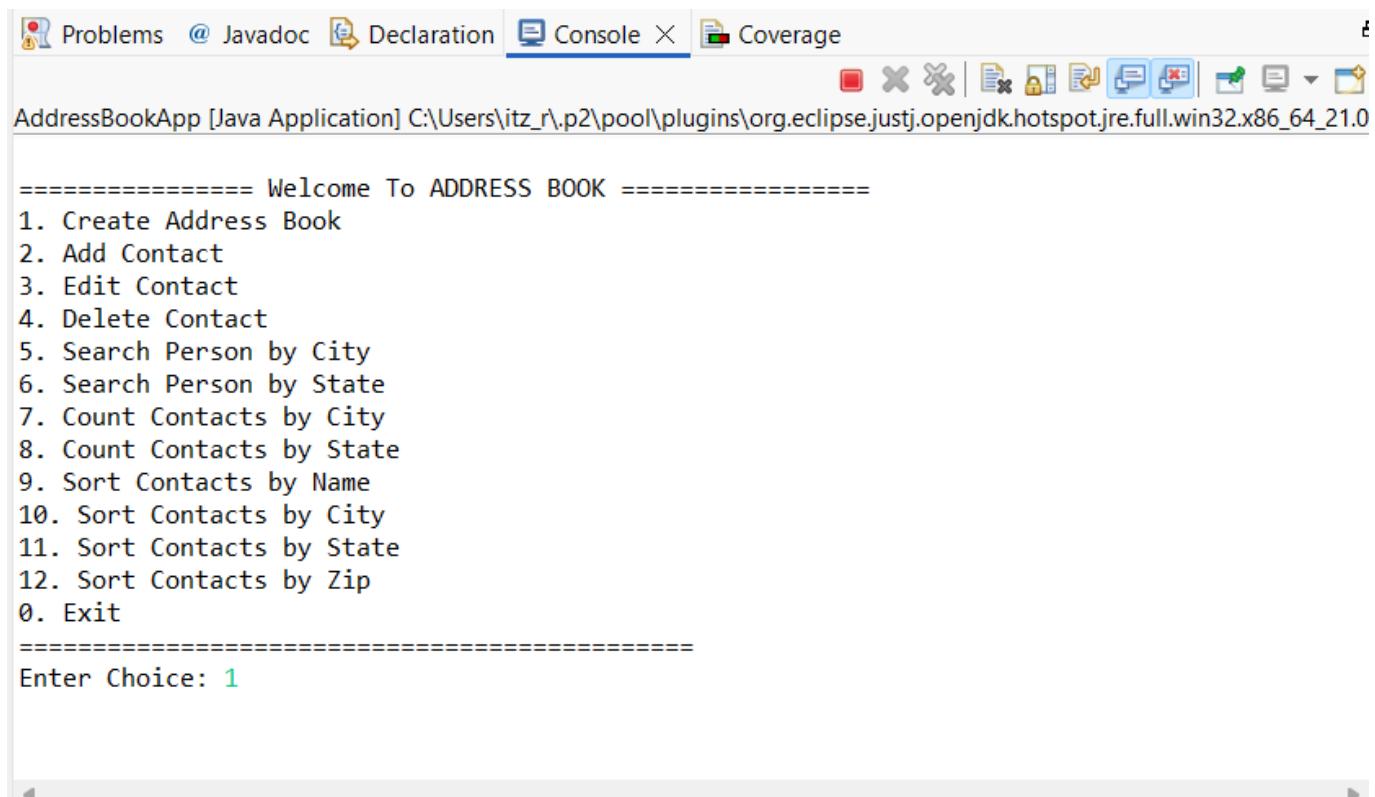
- ProjectName:** AddressBookSystem
- Module Included:** Address Book Module
- Type:** Core Java + Eclipse Project
- Source Folder:** src
- Compiled Output Folder:** bin

2) Screenshots Provided

I have attached screenshots of:

A) Project Execution

Output of AddressBookSystem running successfully in Eclipse / Console

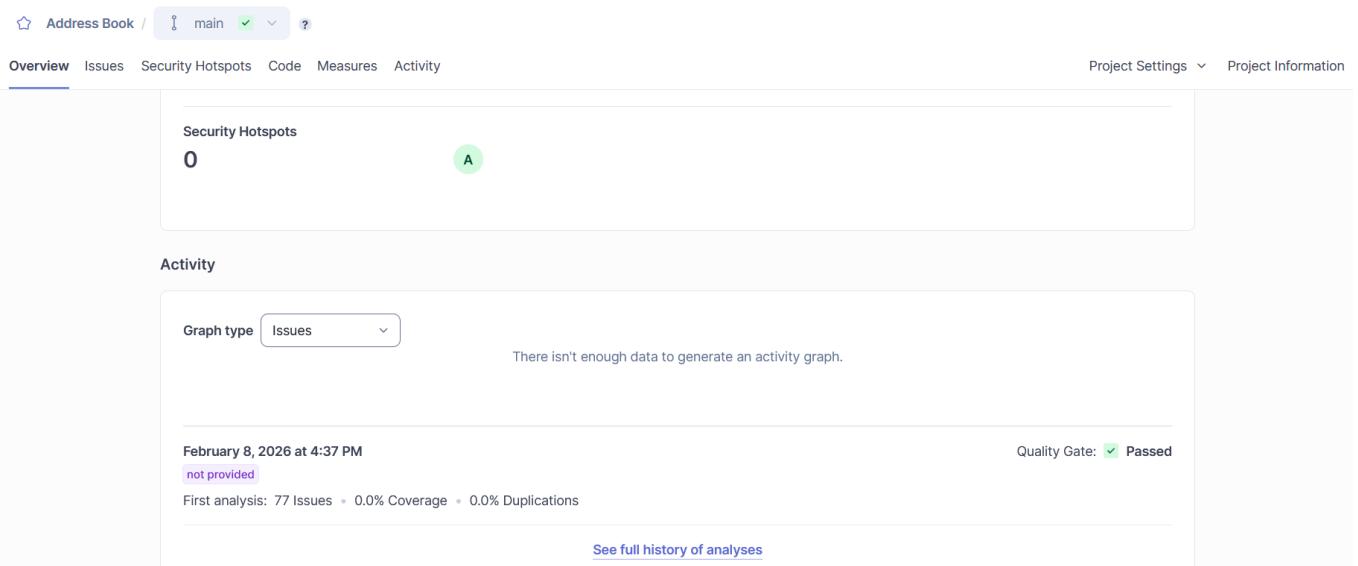
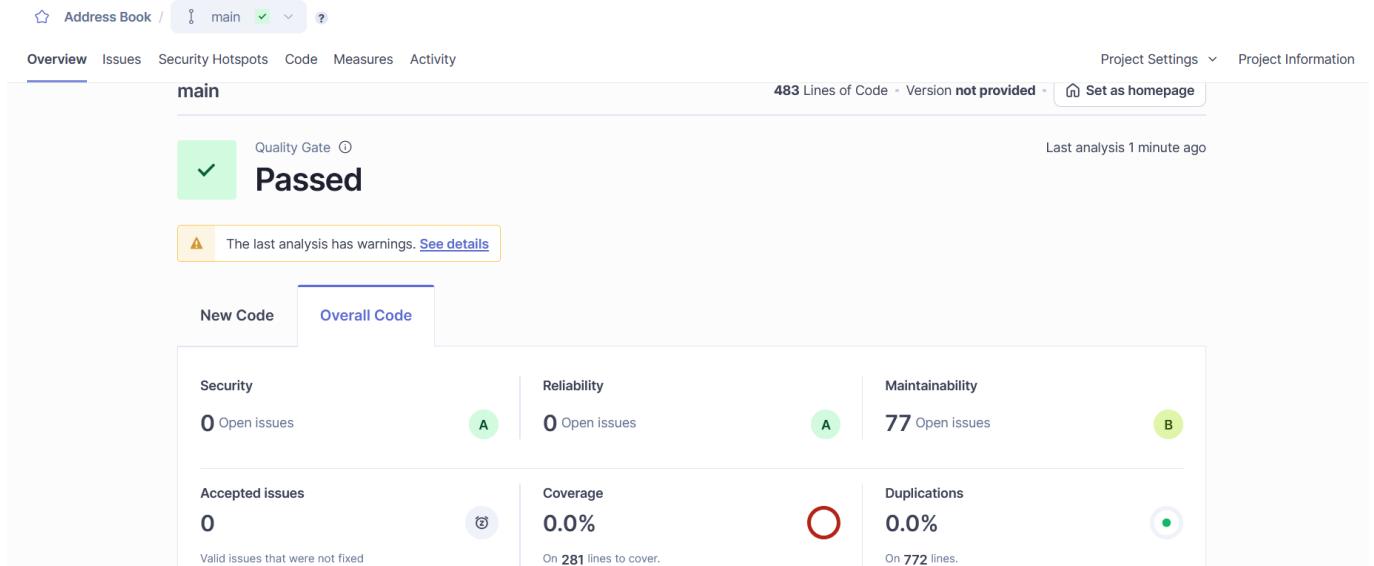


The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The title bar indicates the project is 'AddressBookApp [Java Application]' and the workspace path is 'C:\Users\itz_r\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0'. The console window displays the following text:

```
===== Welcome To ADDRESS BOOK =====
1. Create Address Book
2. Add Contact
3. Edit Contact
4. Delete Contact
5. Search Person by City
6. Search Person by State
7. Count Contacts by City
8. Count Contacts by State
9. Sort Contacts by Name
10. Sort Contacts by City
11. Sort Contacts by State
12. Sort Contacts by Zip
0. Exit
=====
Enter Choice: 1
```

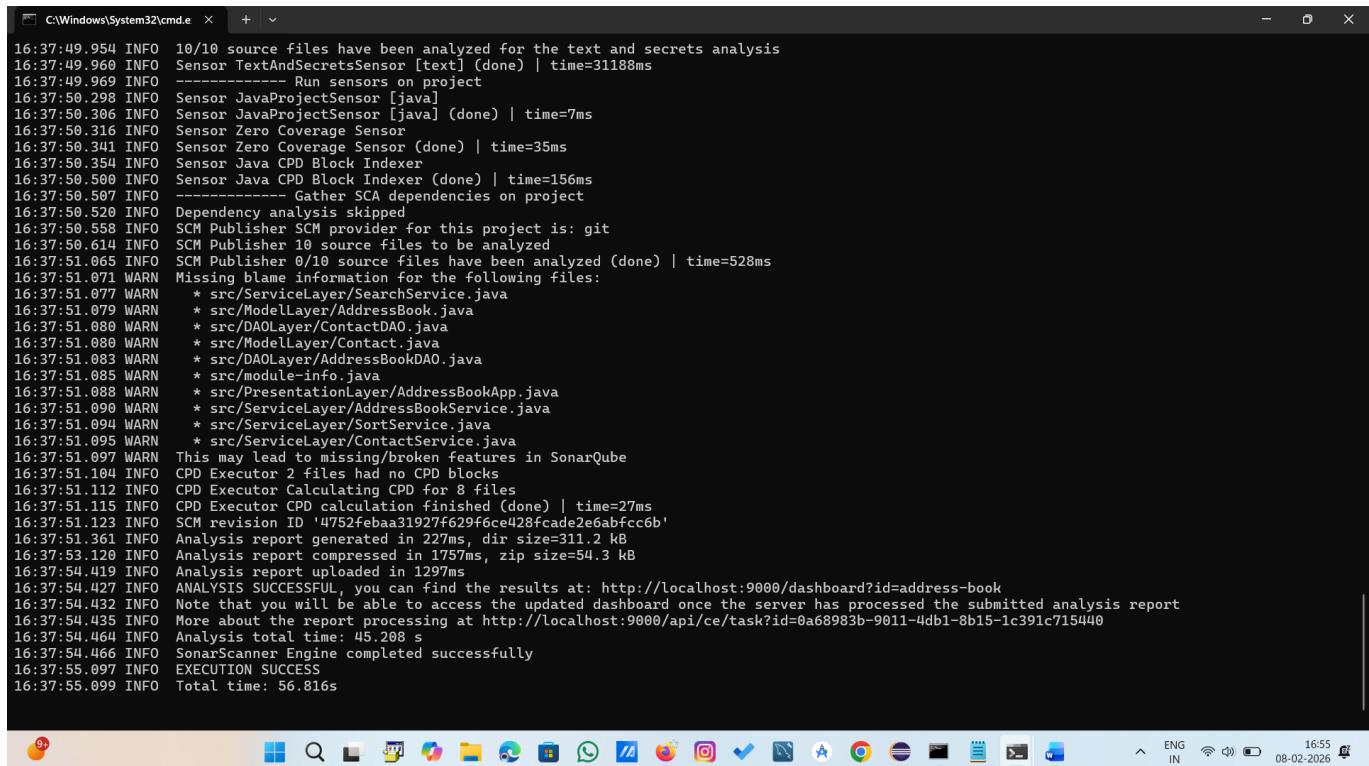
B) SonarQube Report

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



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.exe  +  -  x
16:37:49.954 INFO 10/10 source files have been analyzed for the text and secrets analysis
16:37:49.960 INFO Sensor TextAndSecretsSensor [text] (done) | time=31188ms
16:37:49.969 INFO ----- Run sensors on project
16:37:50.298 INFO Sensor JavaProjectSensor [java]
16:37:50.306 INFO Sensor JavaProjectSensor [java] (done) | time=7ms
16:37:50.316 INFO Sensor Zero Coverage Sensor
16:37:50.341 INFO Sensor Zero Coverage Sensor (done) | time=35ms
16:37:50.354 INFO Sensor Java CPD Block Indexer
16:37:50.500 INFO Sensor Java CPD Block Indexer (done) | time=156ms
16:37:50.507 INFO ----- Gather SCA dependencies on project
16:37:50.520 INFO Dependency analysis skipped
16:37:50.558 INFO SCM Publisher SCM provider for this project is: git
16:37:50.614 INFO SCM Publisher 10 source files to be analyzed
16:37:51.065 INFO SCM Publisher 0/10 source files have been analyzed (done) | time=528ms
16:37:51.071 WARN Missing blame information for the following files:
16:37:51.077 WARN * src/ServiceLayer/SearchService.java
16:37:51.079 WARN * src/ModelLayer/AddressBook.java
16:37:51.080 WARN * src/DAOLayer/ContactDAO.java
16:37:51.088 WARN * src/ModelLayer/Contact.java
16:37:51.083 WARN * src/DAOLayer/AddressBookDAO.java
16:37:51.085 WARN * src/module-info.java
16:37:51.088 WARN * src/PresentationLayer/AddressBookApp.java
16:37:51.090 WARN * src/ServiceLayer/AddressBookService.java
16:37:51.094 WARN * src/ServiceLayer/SortService.java
16:37:51.095 WARN * src/ServiceLayer/ContactService.java
16:37:51.097 WARN This may lead to missing/broken features in SonarQube
16:37:51.104 INFO CPD Executor 2 files had no CPD blocks
16:37:51.112 INFO CPD Executor Calculating CPD for 8 files
16:37:51.115 INFO CPD Executor CPD calculation finished (done) | time=27ms
16:37:51.123 INFO SCM revision ID '4752febaa31927f629f6ce428fcade2e6abfcc6b'
16:37:51.361 INFO Analysis report generated in 227ms, dir size=311.2 kB
16:37:53.120 INFO Analysis report compressed in 1757ms, zip size=54.3 kB
16:37:54.419 INFO Analysis report uploaded in 1297ms
16:37:54.427 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=address-book
16:37:54.432 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
16:37:54.435 INFO More about the report processing at http://localhost:9000/api/ce/task?id=0a68983b-9011-4db1-8b15-1c391c715440
16:37:54.464 INFO Analysis total time: 45.208 s
16:37:54.466 INFO SonarScanner Engine completed successfully
16:37:55.097 INFO EXECUTION SUCCESS
16:37:55.099 INFO Total time: 56.816s
```

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: **77 Code Smells**

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

 Duplications: **0%**