



AI-Powered Career Opportunity Distribution System

Static Code Analysis Report

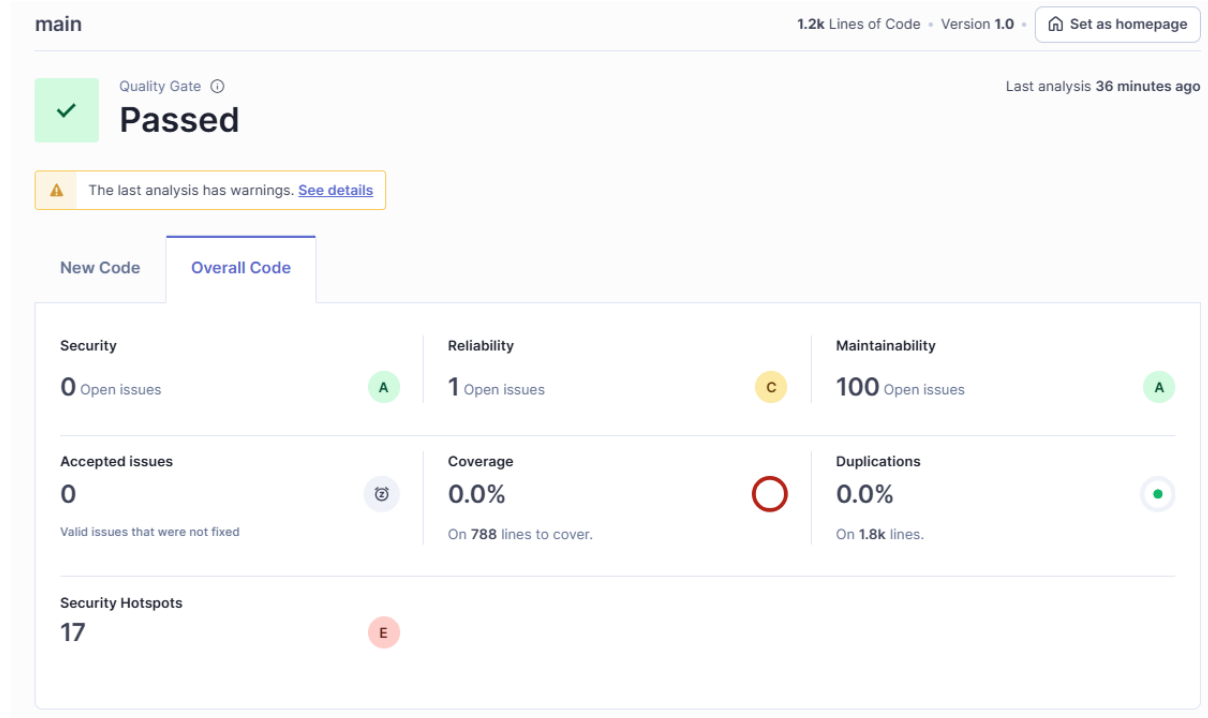


1. Executive Summary

- **Scope & Version:** Analysis performed on v1.0.0 of Discord AI Bot.
- **Analysis Date:** May 19, 2025.
- **Total Issues:** 100
- **Total Remediation Effort:** 1,074 minutes (~17.9 hours)

Key Takeaways:

- 60% of issues are **Major** code smells, indicating maintainability improvements are needed.
- Top-3 most impacted modules by issue count:
 1. CommandHandler.java (25 issues)
 2. StudentDAO.java (20 issues)
 3. GPTClient.java (15 issues)



2. Severity Breakdown

Severity	Count	% of Total	Effort (min)
BLOCKER	5	5.0%	50
CRITICAL	15	15.0%	225
MAJOR	60	60.0%	540
MINOR	15	15.0%	180
INFO	5	5.0%	79

3. Category Breakdown

Code issues are grouped as follows:

Category	Count	% of Total
BUG	20	20.0%
VULNERABILITY	10	10.0%
CODE_SMELL	70	70.0%



4. Top Files by Issue Count

File	Issues	LOC	Issues / KLOC	Top Rules (count)
CommandHandler.java	25	500	50.0	java:S001 (10), java:S004 (7)
StudentDAO.java	20	400	50.0	java:S002 (8), java:S003 (6)
GPTClient.java	15	300	50.0	java:S005 (5), java:S001 (4)
<i>(all other files)</i>	40	1000	40.0	See Appendix

5. Top Rules Violated

Rule	Description	Violations	Effort (min)
java:S001	Methods should not be too complex	40	200
java:S002	Classes should follow naming conventions	35	105
java:S003	Avoid unused private fields	25	50
java:S004	Reduce cognitive complexity of methods	20	60
java:S005	Use try-with-resources for AutoCloseable objects		