

## Code Coverage Report

Missed Instructions: 29 out of 705 instructions were missed, resulting in a 95% coverage rate.

Missed Branches: 1 out of 20 branches was missed, also resulting in a 95% coverage rate.

Complexity : The complexity metric indicates a value of 40, suggesting moderate complexity within the codebase.

Missed Lines, Methods, and Classes: There were 9 missed lines, 2 missed methods, and 1 missed class identified.

### Detailed Analysis:

src.main.app: This portion of the codebase exhibited the lowest coverage, with only 0% instruction coverage. There were 2 missed branches and 2 missed lines, indicating potential gaps in testing for critical functionalities. Further attention is required to improve coverage in this area.

test: While the test code showed a high overall coverage rate of 98%, it's important to note that this is typical in test code due to the nature of comprehensive testing. However, ensuring that critical edge cases and error handling are adequately covered is essential for robust testing.

src.sort, src.model, src.view, src.search, src.controller: These sections of the codebase demonstrated relatively high coverage rates, ranging from 95% to 100%. This indicates strong testing practices and comprehensive test suites in these areas.

Conclusion: The code coverage analysis provided valuable insights into the effectiveness of testing efforts within the project. While certain areas demonstrated strong coverage, there is room for improvement, particularly in src.main.app. By addressing these areas and adopting best practices for testing, the project can enhance its overall code quality and reliability.