**CS 567 Final Project**

**Venkata Susai Haneesh Vallamkonda (vv474)**

**ToDoList Manager Testing Report**

**Introduction:**

This report presents a thorough analysis of the testing activities conducted on the ToDoList Manager. The manager consists of functionalities to add, remove, mark as done, view, search, set due dates, and save/load tasks. The primary focus of testing was on unit testing using Python's unittest framework.

**Test Environment**

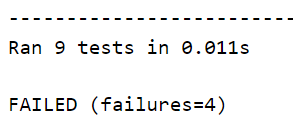
* **Operating System**: Windows
* **Programming Language**: Python
* **Testing Tools Used**: Unittest.py,Coverage.py, universalmutator.py

**Unit Testing:**

The ToDoList Manager underwent extensive unit testing to ensure the reliability and correctness of its functionalities. The unit tests cover various scenarios, including adding tasks, marking them as done, setting due dates, searching for tasks, and saving/loading tasks. Each unit test is designed to validate the expected behavior of the corresponding function.

**Unit Test Cases:**

1. **test\_add\_task:**
2. **test\_remove\_task:**
3. **test\_mark\_as\_done:**
4. **test\_view\_all\_tasks:**
5. **test\_view\_pending\_tasks:**
6. **test\_view\_completed\_tasks:**
7. **test\_set\_due\_date**
8. **test\_search\_tasks:**
9. **test\_save\_and\_load\_tasks:**



**Coverage Analysis**

A coverage analysis was performed using **coverage.py**. The results are as follows:

A close up of a number

Description automatically generated

* **Total Statements**: 124
* **Missed Statements**: 9
* **Overall Coverage**: 93%

The coverage report indicates that 10 statements within **main.py** and 0 statement within **main\_test.py** were not executed during testing. The missed statements in **main.py** correspond to lines18,25,47,48,50,67-70.

These uncovered lines suggest potential areas where additional tests may be necessary to ensure comprehensive evaluation of all code paths.

**Mutation Testing**

The mutation test results were as follows using universal mutator:

**Mutant Test for Main.py:**

**A list of black text

Description automatically generated**

The Score is 100%

**Discussion:**

The ToDoList Manager exhibits a commendable 93% overall coverage, demonstrating robust unit testing. However, 9 missed statements in key areas highlight potential gaps. In contrast, a flawless 100% mutation score emphasizes the suite's effectiveness in detecting and rectifying faults. This discrepancy underscores the importance of a balanced testing strategy, combining extensive coverage with mutation testing for comprehensive fault detection. The ToDoList Manager's approach ensures both reliability and robustness.

**Recommendations:**

**1. Enhance Test Cases:** Address the 9 missed statements identified in the coverage report by crafting new tests, ensuring comprehensive coverage**.**

**2. Evaluate Mutation Survivors:** Scrutinize each surviving mutant to understand test failures. Modify or create new tests to effectively detect and eliminate these mutants.

**3. Adopt Continuous Testing Practices**: Integrate coverage analysis and mutation testing into a routine cycle for early issue detection and continuous improvement.

**4. Conduct Code Review and Refactoring:** Review areas where mutants persisted, considering refactoring for enhanced code clarity and simplified fault detection.

**5. Optimize Test Suite:** Streamline or eliminate redundant tests without compromising fault detection, optimizing overall test suite efficiency**.**

**Conclusion:**

The ToDoList Manager testing reveals a commendable 93% coverage and a perfect 100% mutation score, emphasizing a robust testing approach. While the coverage identifies potential gaps, mutation testing showcases the suite's fault-detection prowess. Balancing both strategies ensures reliability and robustness. The recommendations aim to further elevate the system's quality and resilience, aligning with industry best practices.