This test plan outlines the approach taken to ensure the correctness and reliability of our final project. The system comprises three primary modules: Account, Transactions, and Math. Each module has been thoroughly tested using a combination of automated OUnit tests and manual testing. The test cases have been designed using both black-box and glass-box testing techniques, ensuring comprehensive coverage of functionality.

1. Testing Approach

The Account module was rigorously tested for various functions such as account creation, data retrieval, updating, and deletion. Black-box testing was used to validate the overall functionality based on input and output, while glass-box testing ensured all internal paths and edge cases were covered. Specifically, we verified the creation of accounts with correct attributes, accurate data retrieval, proper account updates, and correct account deletions. These tests guarantee that the Account module performs all essential operations reliably.

The Transactions module was tested to ensure accurate transaction creation, addition, summation, and filtering. These tests focused on both the expected outcomes and the correctness of internal logic. We validated that transactions were created with correct attributes, added to files accurately, summed correctly, and filtered by category and date appropriately. This comprehensive approach ensures that the Transactions module handles all necessary operations effectively and accurately.

The Math module was tested for its utility functions, including average calculations, spending optimizations, and random selections. A combination of black-box and glass-box testing techniques was employed, along with randomized testing for functions involving random outcomes. We ensured that the average calculations, spending optimizations, and days until balance depletion were accurate. Additionally, we validated that random selections from predefined lists were correct, ensuring the module's utility functions are reliable.

1. Justification for Test Suite

The test suite demonstrates the correctness of the system by thoroughly covering key functionalities across all modules. Black-box testing validates the system's behavior from an end-user perspective, ensuring that it meets the specified requirements. Glass-box testing provides deeper insights into the internal workings, verifying that all paths and edge cases are handled correctly. The inclusion of randomized tests adds another layer of robustness, ensuring that non-deterministic functions behave as expected. This combination of techniques provides strong evidence that the system is reliable and functions as intended.

By combining these techniques, the test suite offers a comprehensive validation of the system's correctness. It ensures that the system's core functionalities are robust, edge cases are handled appropriately, and random behaviors are tested for reliability. This rigorous testing approach demonstrates confidence in the system's ability to perform accurately and reliably in real-world scenarios.