**Module 4: Unit Testing**

**Justin Hancock**

**CS-405 Secure Coding**

**09.25.2024**

In this assignment, I developed and tested various collection operations using the Google Test framework. The focus was on verifying that the collection behaves correctly in both positive and negative cases.

Throughout the testing process, I followed a systematic approach to identify and resolve issues. One of the key issues encountered during the tests was in Test 4 (CanAddToEmptyVector). The test was failing because the condition after adding an entry was incorrectly written as ASSERT\_TRUE(collection->empty()), which was checking if the collection was still empty after an addition. This was an error in logic (and developer error) as the collection should have no longer been empty after adding an element.

To resolve this issue, I first reviewed the failing test and verified that the logic contradicted the expected behavior. I changed the incorrect assertion to ASSERT\_FALSE(collection->empty()) to ensure that the test correctly checks that the collection is no longer empty after adding an entry. After making this correction, the test passed as expected.

Each test was designed to validate the expected behavior of the collection. For example, tests for adding values, resizing, and clearing the collection promised that the size and state of the collection were properly updated. Negative tests such as Test 14 and 16, verified that the collection handled invalid operations by throwing appropriate exceptions.

The debugging process was methodical, addressing errors in logic and verifying that each test correctly validated the behavior of the collection. Both positive and negative tests were used to confirm that the collection behaves as expected throughout various scenarios. Completing the unit tests this way proves that the system handles both valid and invalid operations correctly.A screenshot of a computer program

Description automatically generated