Milestone III

CPTS 422 SOFTWARE ENGINEERING PRINCIPLES II

Jackson Schuur

Sophia Schuur

Katherine Freund

Austin Ryf

MILESTONE REPORT III

Team Member Roles

Sophie and Jack led the improved unit tests area of the project. They corrected and improved upon old test cases from Milestone 2, found and corrected bugs in the program, and added new test cases to reflect the bugs found as well as new edge test cases. Kathy and Austin contributed to the implementation of integration tests. They created and tested integration tests with respect to the bugs found from unit testing. Each team member contributed to their respective documentation. Each team member tested their own test cases, and when we felt finished with our cases, we met as a team and verified all cases together. All members contributed to the creation of this document.

Testing Framework & Justification

We used the Jest framework for developing tests for our project. Our project is simple and small in scope, so Jest proved a solid choice of framework because it is simple, works out of the box, and requires little configuration.

We used the top-down testing technique for several reasons. Our calculator is more functional/procedural than object-oriented, our program is simple rather than complex, and we used stubs/mocks for the integration testing. These attributes led to top-down integration testing making the most sense.

Project Activities

Our team felt confident about each of our strengths and decided early on how we wanted to split the testing. As a team we have developed a schedule system we have stuck to since the beginning of the project. We felt confident about the requirements set during the first two milestones and when needed, we conversed over instant messaging about expectations for what needed to be done and who was going to accomplish it. We use a dedicated group chat for this purpose. We met as a team a couple weeks ago to get started on the testing code and clarify expectations. We met earlier this week to begin writing this document and finished earlier today.

Date	Notes
Nov 17th	Clarify expectations. Discuss task assignments.
Dec 3rd	Begin document. Discussed testing code. Verified testing code.
Dec 5th	Completed document. Finalized git pushes. Verified all tests working and all areas of document are complete.

Test Outcomes

Bug #1: Digit Cap

Originally, the user could input more digits than could be displayed on screen. This number would be reflected in the debugger, but not on the UI. To overcome this, we put a length cap at 13 digits. Now, when the user tries to enter a number or a decimal after 13 digits are already displayed on the UI, the program ignores the 14th.

Bug #2: Answer Overflow

Closely linked to the first bug, this issue also applies if the length of the answer after applying an operator to two operands exceeds 13. To fix this, we decided to display an alert to the user after the second operand if the answer length exceeds 13. Then the program starts again from scratch (at 0).

Unit Tests Changes

We added some unit tests to demonstrate the fixes described in the previous section. We also added a handful to test multiple operands and dividing by zero. We corrected some old test cases from Milestone 2 to use strings, as used in the calc program itself, instead of integers.

Integration Tests

Jest was also used to create the integration tests. Building the integration tests was easy and followed similar patterns to the unit tests in terms of process. For the integration tests, we tested the connections between the user interface and the interior calculations. We also tested the relationship between behavior of the solve functions with the input handlers.

Software Documentation Improvements

Our team did not find improving upon older software documentation from previous milestones necessary as leaving the old documentation as-is helped us better understand the progress and changes we made, as well as how to improve it. We found that looking back on old documentation improved our understanding.

Experience Summary

As in the previous milestone, we found that moving the tests to a different scope and using Jest allowed us to swiftly and seamlessly develop tests. Since we gained prior experience using designated testing frameworks and writing unit tests in the previous milestone, we found that writing new unit and integration tests came to us quicker and easier because of it.