Michael Ladderbush

Module 5 Journal

In developing each milestone, I implemented a thorough testing approach. I heavily relied on JUnit Testing. Creating test cases for individual methods inside of my Java classes, guaranteeing that each segment of code operated as intended across various scenarios. Additionally, I used input validation to test how the program handled diverse user inputs, focusing on preventing errors or unexpected behavior caused by invalid inputs. My test cases deliberately provided invalid inputs, allowing me to assess the effectiveness of methods within my classes and make sure that they would be able to handle any user input.

While I primarily utilized JUnit Testing, Input Validation Testing, I did not incorporate other testing techniques. I didn’t use System Testing, which evaluates the entire integrated system. Regression Testing for ensuring new changes don't impact existing functionalities, Performance Testing for assessing speed and responsiveness, Usability Testing for evaluating user-friendliness, Security Testing for identifying vulnerabilities, and Acceptance Testing to validate if the software meets acceptance criteria.

I found JUnit Testing to be indispensable for individual Java methods and I can imagine that when creating modular code in a collaborate environment it would be extremely important to test each individual piece of code that is introduced. Input Validation Testing is important as well and in my experience in other classes I’ve learned that input validation plays a huge role in maintaining secure and robust systems that will be able to handle either invalid data or potential security breaches. While I focused on these forms of testing, the broader spectrum of testing techniques such as System Testing, Regression Testing, Performance Testing, Usability Testing and Acceptance Testing all play a vital role. Systems testing ensures that the system as a whole is compliant with a multitude of systems. Performance Testing similarly allows for developers to see how efficient their code is and how it will impact users. Regression testing will be vital once the code-base grows and changes and additions become frequent because the impact on older parts of the code could be extremely important. Usability and Acceptance testing will also help to meet client needs and ensure that the product is actually useable when all is said and done. Though I only used a few forms of testing there are many more and all are important for creating a robust product.