**CS-320 Project two**

1. **Describe your unit testing approach for each of the three features.**
   1. **To what extent was your approach aligned to the software requirements? Support your claims with specific evidence.**

My approach was directly linked to the requirements. This was mandatory as the two were hand and hand. For me to meet my requirements I would have to unit test my program. For example, we had requirements like the following string cannot be longer than 10 characters and cannot be null. This led to me using j unit testing to ensure that the string was not null and was not longer than 10 characters

* 1. **Defend the overall quality of your JUnit tests. In other words, how do you know your JUnit tests were effective based on the coverage percentage?**

I am very sure my j unit testing was effective. This is because when testing I would take different approaches when testing. I would ensure that when I input the correct information my test does not give me an exception thus the test fails. Then, I would do various testing of situations where my test would give me exceptions when it was supposed to. Then I would utilize this familiarity with the unit testing those situations and apply it to the same situation again. This made sure I used experience-based testing skills, and applied to making my unit test very effective to making sure it was up to par.

1. **Testing Techniques**
   1. **What were the software testing techniques that you employed in this project? Describe their characteristics using specific details.**

I mostly used white box, and experienced based testing. The two methods of testing were effective in helping me test during my j unit testing. This is because white box testing is where the person testing the program knows how this program is functioning inside and outside. This is the exact type of testing that would be effective because I do feel as if I knew the program inside and out because I was the one who programmed said program. I also used experienced based testing very often. This was super effective for me because the assignments were similar in many ways which makes my experience in testing those types of requirements effective.

* 1. **What are the other software testing techniques that you did not use for this project? Describe their characteristics using specific details.**

The main testing technique I did not use was Blackbox testing. This is because Blackbox testing would be a testing technique used if I was not familiar with the in-depth details of a program. Let's say I was testing a program that was supposed to ensure the user can play an AI in tic tac toe. While I do not know the exact details of how the program was created, I do know that in tic tac toe I should be able to win by getting three in a row. So, I would be testing based on the knowledge only since I do not know the program Indepth.

* 1. **For each of the techniques you discussed, explain the practical uses and implications for different software development projects and situations.**

The testing techniques that I used for this were error guessing, statement testing, and decision testing. I used error guessing because of my experience which is hand and hand with experience-based testing. Having prior experience with similar requirements made this a very effective method. Statement testing was by far the most used testing technique because of the fact of it going hand and hand with the unit testing that I was doing. Lastly, I used decision testing which flowed well with my statement testing. I ensured that when I executed a statement incorrectly that the program decided to throw an error which is why I believe those two are intertwined in my testing method.

1. **Mindset**
   1. **Assess the mindset that you adopted working on this project. In acting as a software tester, to what extent did you employ caution? Why was it important to appreciate the complexity and interrelationships of the code you were testing? Provide specific examples to illustrate your claims.**

When I was acting as a tester, I was more cautious the more I got familiar with all the requirements. I began thinking of more errors that could occur, and this made my testing more efficient. So, I think the relationship between the programs was very important as it allowed me to connect the dots and become more efficient with my testing.

* 1. **Assess the ways you tried to limit bias in your review of the code. On the software developer side, can you imagine that bias would be a concern if you were responsible for testing your own code? Provide specific examples to illustrate your claims.**

I believe bias was very influential during this, because in your head you believe your code is already efficient and sometimes, we need critiquing to see the errors and take off our blinders. The feedback I received helped me see I was being biased as I was missing key details that I was not aware of. I believe this is reflected in my grades, as my testing covered more, and met the requirements with the last project.

* 1. **Finally, evaluate the importance of being disciplined in your commitment to quality as a software engineering professional. Why is it important not to cut corners when it comes to writing or testing code? How do you plan to avoid technical debt as a practitioner in the field? Provide specific examples to illustrate your claims.**

I believe testing is by far the most important part of software, because it could have fatal consequences if not properly tested. Software is used for many things, such as medical equipment, banking systems, and many more things. Imagine having your bank accidentally having a software error where it sent customer information to the wrong people. Or your medical equipment functions improperly, not giving you enough medicine. This is why it is super important to test effectively and not cut ANY corners. It could be detrimental to the customer, and it's your job as a tester to ensure that does not occur.