**CS – 320: Module Seven Reflection**

**Ryan Blough**

1. **Summary**
2. Describe your unit testing approach for each of the three features.
3. To what extent was your approach **aligned to the software requirements?**

My approach aligned with the software requirements because I was able to separate all three of the different classes into separate packages. Which will make it easier to make adjustments in the future if needed, as they are all separated and not crammed together making it difficult to decipher between the three. The tests however, are in there own “test” package, which allows the classes to work in conjunction with each other so that the JUnit tests function properly together.

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 know that my JUnit tests were effective because I could make them intentionally pass and fail, by adjusting the specific class in the different packages. Which allowed to me to determine that I had full coverage while running the JUnit tests.

1. Describe your experience writing the JUnit tests.
2. How did you ensure that your code was **technically sound?**

I know that my code was technically sound because I could make the JUnit test pass and fail like I signified above. An example of this is in Lines 37 – 40 in the Appointment Test class, which shows that if the Address is longer than 50 characters then that specific test will fail.

1. How did you ensure that your code was **efficient?**

I know that my code was efficient because I could easily update, add or delete things when I needed to. An example of this is in Lines 28 – 33 of the Task Service Test class, which signifies the need to Update a Task Description, and accomplishes it successfully when called upon.

1. **Reflection**
   1. Testing Techniques
      1. What were the **software testing techniques** that you employed in this project?

I applied a few different testing techniques for this project including Manual Testing, with the manual tester being myself. I would visually inspect the code to make sure that it free of errors and then test it. Another method that I used is the Unit Testing, using the JUnit tests, I was able to determine what to change and what to leave alone to make things either pass or fail, whenever I wanted to adjust something within the code. And also, Integration Testing, as I would test the entire Java Project as a whole to ensure that all of the different packages and classes were working together to produce the desired output.

* + 1. What are the **other software testing techniques** that you did not use for this project?

I did not apply Regression Testing, as a I was careful to not effect other components of the code when I did make any necessary changes to the code. I did not apply Security Testing, as this was not a part of this assignment, but if this was the real world, you would absolutely want to ensure that there is no security vulnerabilities involved with your code that would put your company or its customers at unnecessary risk.

* 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?

The mindset that I adopted when working on this project was that of a customer, I wanted to make sure that all of the components worked together to make it as easy possible for the people who would end up using the application. I used an extreme amount of caution to make sure that each of the components would not affect one another when running the JUnit tests. Scrupulously monitoring what each test in the test package would effect and what changed when you made a specific adjustment to each one of the tests. It is important to appreciate the complexity and interrelationships of the code because it is of vital importance that they work congruently together and not work against you when you are applying the test classes.

* + 1. Asses 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 you own code?

I tried to limit bias by trying my best to view the code from the point of view of the customer, it is important to take this view as it would be easy to look at your code and say that it is perfect when there are obvious errors to it that you do not catch. By taking this viewpoint it is easy to justify the mistakes that were made and to go back to correct them accordingly. When developing code it is important to take this stance because while you very well may use the code as a customer, when things are “buggy” it could turn people away from using the app which will take away money from the company and require a fix to the issue.

* + 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?

Quality is one of most important things to developing code, if you are not disciplined while you are writing the code, then mistakes will happen and the application will have errors and it will make using it annoying to the customer base. Cutting corners is definitely not ideal as it will almost guarantee that a mistake in the code will be made, therefore you must be vigilant and anticipate all the possible mistakes that can happen, and have an appropriate response to them. I plan to be vigilant and very detailed with my code, so that when mistakes due arise a catch them before they go to the next stage of development.