**Tyler Barnes**

**7-2 Project 2**

**Southern New Hampshire University**

**CS 320 Software Test Automation**

**16 April 2024**

The approach I applied for all three features is an example of white box testing. I had to rewrite most of the assignments so it would work. I did some research for help in writing the service classes and tests. From the software requirements, I made sure I did what was asked of me. When writing the Appointment class, I had to make sure the ID was no longer than 10 characters, and that it wasn’t null. I created private variables and created getters and setters, and finally, a constructor that tied everything together.

The overall quality of the tests would be around 50%. I didn’t realize I had to look out for failed tests, I mostly tested the project when it worked. The tests were a bit more sound, I thought out how to write better tests that ran to the requirements. I thought the only thing I could improve on was testing the scenario that fails. I made sure the code was efficiently written. I went through a lot to get this to work, and I couldn’t figure out how to. I looked up more examples and once I had my version of code, it was easier to write.

As I said before, I wrote the tests with the mindset of wanting to implement the white-box testing for the service. I knew the project requirements and kept it in mind while writing the software.

Working on the project, I had to put myself in a different mindset. This was a new way of thinking for me. Running tests, and applying it to the code was a new way of thinking. This took me a while, but I got the hang of it. I realized while working on this, I could apply almost no caution. I did take into consideration whether the objects I created passed, but I didn’t check if they failed. I also created unnecessary objects that didn’t need to be within some of the services.

I noticed as I start to understand programming more and more, I tend to do things the longer (not long) way. I know friends in the field who told me the degree gets you started and when you start your first job, it’s like starting over again.

It’s important not to cut corners when writing or testing code. You are learning how to write it for future employers. Lack of care shows the personality and work ethic of the developer. This reflects negatively on them, whether it be intentional or not.