Tobias Burgess

CS-320

Project Two

08/17/2025

**Summary and Reflections Report**

For the first milestone, Contact Service, being new to JUnit testing, I leaned heavily on the prompt provided in the announcements as well as chapter 3 of *Mastering Software Testing with JUnit 5 : Comprehensive Guide to Develop High Quality Java Applications*. I used these resources to try to meet the requirements. Looking back at it now, I think my test classes are a bit of a mess with inconsistent testing and missing “assertThrows” code. This is most likely a result of me not really knowing what I was doing, not giving myself enough time, and piecing together what I thought was needed in a rushed and unorganized manner. Despite what the code is lacking, when running coverage as Junit test, I still have 88.5% for Contact.java, and 95.9% for ContactService.java, which is not bad.

For milestone two, Task Service, I felt like my understanding of Junit testing improved a lot over the previous milestone. I was able to concentrate much more easily on making sure I had all of the proper pieces in the proper places instead of struggling to understand the material. I do think my TaskTest class could be cleaned up a little with a helper method and @BeforeEach to initialize a valid task for some of the tests. Otherwise, I think my Junit testing for Task Service is pretty well done with no major mistakes and 100% coverage for both classes.

Milestone 3, Appointment Service, was much easier given the repetitive nature of the requirements across milestones. This and my growing familiarity and comfort with writing Junit tests allowed me to focus even more on writing clean and easy to read code. My coverage was 100% for both classes in Appointment Service and overall seems to meet all requirements with no major corrections to be made.

The main testing technique for these unit tests was black box testing, where input was fed in and the output was checked. Also, some negative testing was used to check that code failed properly. My mindset was that I probably messed something up so I should really try to test it as thoroughly as possible, this is one positive aspect of my lack of experience. The same benefit also applies to bias, as I don’t yet feel confident enough in my work to think it doesn’t have any problems.

Discipline is something that I think needs to come with practice. If I’m not solid with my knowledge and use of certain testing techniques, I think it’s very difficult to be disciplined in an effective manner. It’s easy to overlook things, especially as projects get bigger and more complex, being disciplined is the only way to ensure we don’t miss the mistakes we’re trying to catch. I definitely feel a little more disciplined than when I started this class, as far as unit testing goes.