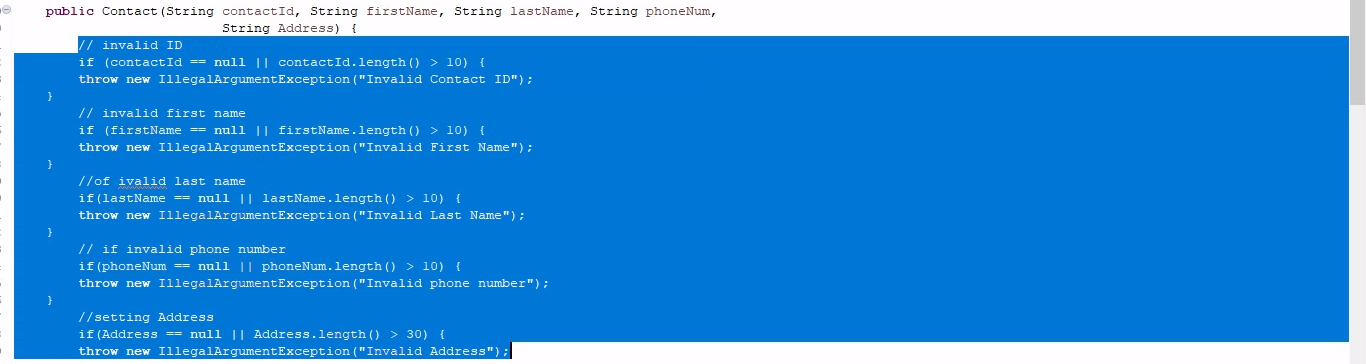
Ashlee Wood

Project Two

### **Summary and Reflections Report**

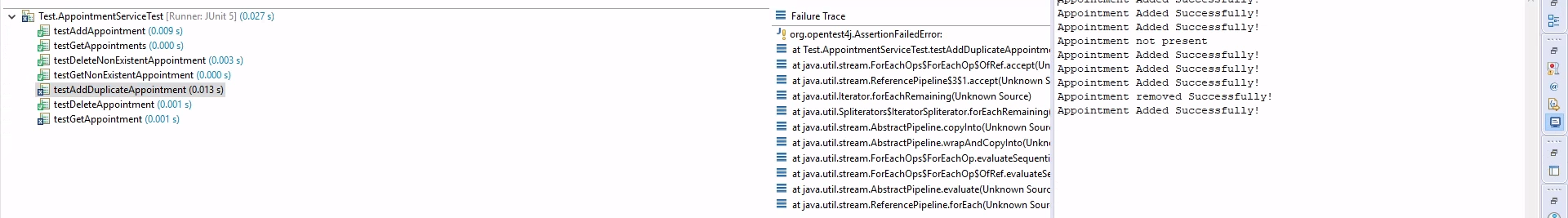
1. **Summary**
   1. Describe your unit testing approach for each of the three features.
      1. To what extent was your approach **aligned to the software requirements**? Support your claims with specific evidence.

**I used unit testing for all three features of the mobile application. When regarding the first assignment of this software project, which is the contact class and testing. I was able to implement small portions of the code one at a time. The contact class needed to work before the test could be implemented as the contact service class before its testing. This is similar testing for the task class and appointment classes and JUnit testing. All these codes had requirements I had to make sure I applied such as the length of the name, phone number, while not being null for any value. This is supported in my software program with the methods to implement this.**



* + 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?

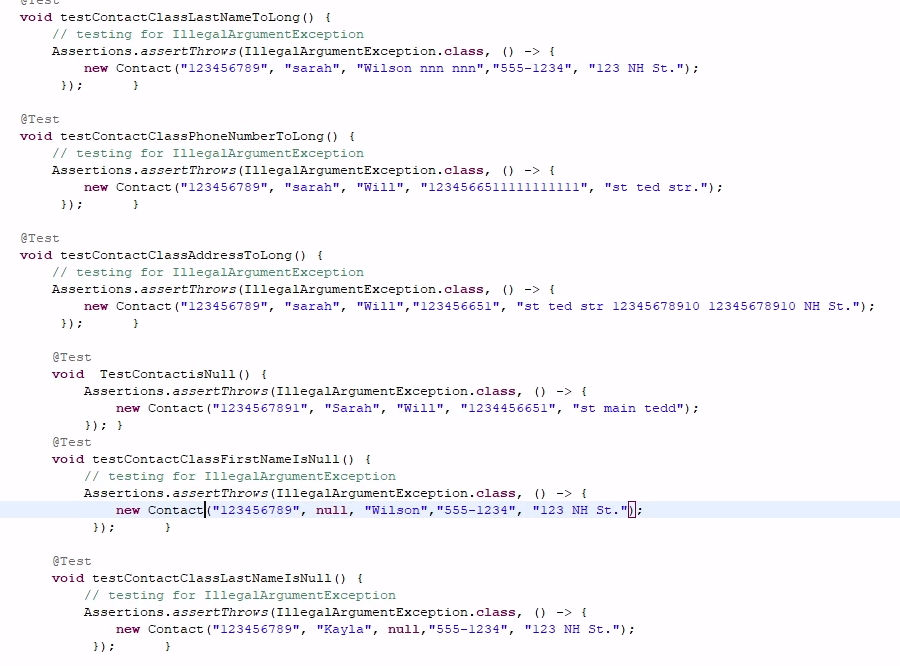
**My JUnit tests were effective based on the coverage percentages I had in my software program. This is shown in my application as such:**



**This shows the errors that will be caught in the software application. This will show if my tests were coded correctly and if they will output the errors in these scenarios. I was not able to get 100 percentage in my unit testing. This is okay as the time management decisions I made aligned with my clients' requirements.**

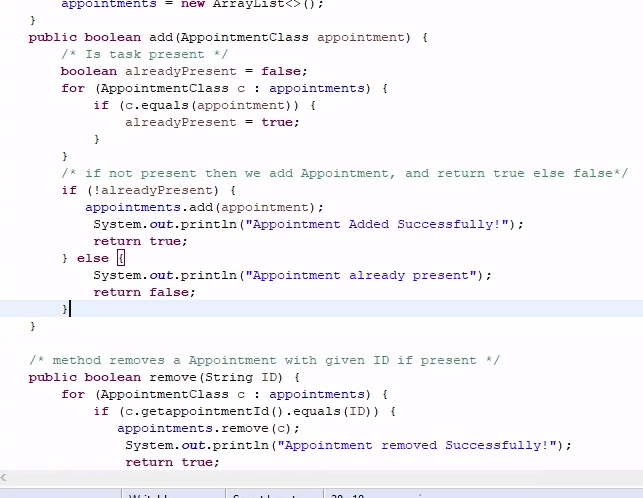
* 1. Describe your experience writing the JUnit tests.
     1. How did you ensure that your code was **technically sound**? Cite specific lines of code from your tests to illustrate.

**I tested with the JUnit tests to make sure my code was technically sound in each class of my project. The contact class required testing to make sure the Contact was not too long including the first and last name, address, and phone number. These variables can also not be null. This is set in the contact class and tested in the test of contact class.**



* + 1. How did you ensure that your code was **efficient**? Cite specific lines of code from your tests to illustrate.

**To ensure my code was efficient I made sure my naming variables for my functions, classes and variables were specific to goals for the program. This makes the code easier to understand and compare to the test classes that are the counter part of the set-up classes. I also tried not to over comment the code as some functions are simple and visible of capabilities with the code. Making sure the code comments are not about what the code is doing but why is it doing it.**



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

As mentioned in the grading Rubic, I was able to execute equivalence partitioning in my testing classes. This technique will define test cases that find errors in the classes to reduce the overall test cases that need to be developed. This was able to save me time on the project and reduce the lines and testing in the coding program. Boundary value analysis is a part of this process as well, testing the edge cases of the software. I was able to employ this with my program.

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

**Some software development techniques I did not use were performance testing and security testing. Performance testing is used to evaluate the level of quality in the program. Security testing is used to ensure there are no vulnerabilities in the program's security. I did not use these as I focused mainly on the JUnit testing.**

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

**Performance testing has practical uses and implications in software development by ensuring the program is running to its full potential with quality. Security testing is important in software projects to keep the program secure and protected from unauthorized access and other security-related issues. Equivalence partitioning is used to reduce test cases to keep the test coverage effective. Boundary value analysis is used to test the values of valid and invalid partition boundaries.**

* 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.

**The mindset** that I adopted working on this project was to be sufficient and patient as I implemented each component of the classes and test classes. I ran the test classes one by one to make sure before I continued, I was making progress. Using caution throughout the entire project was the most precise way to keep the software program flowing efficiently and correctly. It is important to appreciate the complexity and interrelationships of the code I was testing. I made sure to test all the Junit combinations of tests for all the outputs and inputs that there could be. A specific example is my coding of the appointment class.

* + 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 was able to limit bias in my review of the code by delving into the functionality of the code. I did not let my own ego get into the way of me understanding when I was wrong and incorporating the feedback into my program. Making sure to review your code and assess the mistakes and take the time to learn why you made the errors will help in future expectations. Some issues I had in my code specifically was forgetting small things like the setters and getters in my class. I was rushing through the code and did not double check.**

* + 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.

**It is important to stay disciplined in my commitment to quality as a software engineer professional to ensure reliability in the field. This is crucial to build clients' trust and connections. It helps me to build on my skills too and keeps me accountable. Cutting corners when writing or testing code does not benefit anyone. It only shows the client you do not respect the work and yourself. You will also form this as a habit when things get stressful and use it as a fall back; it is the same as cheating in a class, it doesn't teach you anything. I will avoid technical debt as a practitioner in the field by staying on top of my work, and all the details. I will practice my skills to stay relevant in the issues that may arise. Feedback is important to understand and implement. An example was in my code where I took mt feedback and reestablished my classes and tests.**