Project Two

Wilda Cheng

As a software engineer for Grand Strand Systems, I developed the tested the client’s mobile application. I created three features with unit tests for the contact, task, and appointment services. My approach when creating unit test for each of the three features by employing various testing techniques. My approach aligns with the software requirements by verifying the product and ensuring that it fulfills the requirements with the utilization of test cases. Each test case covers some aspect of the software requirements. Checking the quality of the Junit tests is important, so to ensure that they were effective I analyzed the coverage percentage. Each feature had the feature’s class and its service requirements, with their respective Junit tests. Each test should have a minimum of 80% coverage percentage, and regarding my tests, each test is above 95%, meeting well above the minimum coverage requirement. My experience writing Junit tests was a positive and learning experience. To ensure that my code for my tests were technically sound, I checked that the logic was correct and producing the expected outputs. To ensure that my code was efficient, I applied various methods such as refactoring my code, removing any redundancy, and applying JUnit Assertion methods. Utilizing JUnit Assertions methods was helpful in validating the expected output with the actual output of a test, letting us know if a test case passed or failed.

The testing techniques I employed in this project include a combination of black box and white box testing. Black box testing is a software testing methodology that I used to test specific functions or features of the software. For example, deleting a contact by its contact ID and if there is no match sending an error message. White box testing is another testing methodology that involves examining the code and checking that the logic is correct. A software testing technique that I did not apply for this project is security testing, which is used for identifying and mitigating potential security risks.

While working on this project as a software tester, I had to adopt my mindset of caution and awareness due to the complexity of the program. I employed caution and limited bias in my review by utilizing testing techniques such as code review and unit tests to carefully examine the code’s functionality. On the software developer side, I can imagine how bias would be a concern if they were responsible for testing their own code. As the developer writing the code, it is easy to have errors go unnoticed since you have presumptions of what the function or features should do. It is important to be disciplined in the quality of writing and testing code.