**Akila Jones**

**Southern New Hampshire University**

CS-320-11639-M01 Software Test, Automation QA 2024 C-4 (Jul - Aug)

**Summary:**

For the customer's mobile application, I created unit tests for the contact, task, and appointment services. To ensure the effectiveness of these tests, I used both black box and white box testing methods. Black box testing focused on inputting data to verify correct outputs, while white box testing involved analyzing the code to ensure logical accuracy. Additionally, I implemented JUnit test cases to address edge cases and identify any code errors. This comprehensive approach resulted in 99.7% code coverage, validating the code’s effectiveness and efficiency.

My experience writing Junit tests was positive. I successfully verified the technical soundness of my code by checking its logic and ensuring it produced the expected outputs. To enhance code efficiency, I employed methods such as refactoring and eliminating redundant code. Additionally, I utilized Junit assertions to confirm that the results matched expectations.

**Reflection Report:**

For this project, I employed black box and white box testing techniques. Black box testing involved inputting data and verifying that the outputs were correct, while white box testing focused on examining the code to ensure logical accuracy. I also utilized JUnit test cases to address edge cases and identify potential errors.

I did not use performance testing and security testing for this project. Performance testing measures system speed, scalability, and responsiveness, while security testing identifies and mitigates potential security risks.

The testing techniques I used were effective for verifying code correctness and efficiency, uncovering errors, and identifying edge cases. Not employing performance and security testing may result in overlooked issues and security vulnerabilities.

During the project, I adopted a cautious mindset, recognizing the complexity of the code. I minimized bias by conducting code reviews and using unit tests to thoroughly assess functionality. As a software developer, being aware of bias and diligent in code testing is crucial.

Finally, maintaining discipline in commitment to quality is essential. Avoiding shortcuts in code writing and testing helps prevent technical debt. I ensured this by using code reviews and unit tests to thoroughly check for errors and uphold code quality.