7-2 Project Two Submission

Duane Wegner

Southern New Hampshire University

CS-320 Software Test, Automation

**Timothy Alexander, M.S., CIS**

December 10, 2024

**Summary and Reflections Report on Software Testing**

As a software engineer at Grand Strand Systems, I developed and tested key features for a mobile application, focusing on managing contacts, tasks, and appointments. This report outlines my approach to unit testing, the techniques I used, and the mindset I adopted during testing. Additionally, it discusses how I ensured the quality of the software, and the lessons learned through this process.

**Summary**

For this project, I utilized unit testing to validate the functionality of the contact, task, and appointment services within the mobile application. Each service had specific behaviors that needed to be tested. For the contact service, I wrote tests to verify that the application could correctly create, update, and delete contacts. I also ensured that all the contact fields, such as name, phone number, and email, were properly captured and stored. For the task service, I tested the functionality of adding, updating, and deleting tasks. A critical part of this was verifying that the description field adhered to a specified length limit. Lastly, for the appointment service, I created tests to ensure appointments could not be scheduled for past dates, as per the software requirements.

The unit testing approach was directly aligned with the software requirements, ensuring that the features functioned as intended. To verify this, I used code coverage tools to measure how much of the code was being tested, confirming that critical parts of the system were properly covered. The tests also included edge cases, such as invalid inputs, to ensure the software could handle real-world scenarios.

While writing the tests, I ensured that the code was technically sound by thoroughly examining the logic for each service. I checked for potential errors and tested edge cases to guarantee that the application handled all inputs correctly. To ensure the efficiency of the code, I focused on simplicity and reusability. For example, I used helper methods to reduce redundancy and make the testing process more efficient.

**Reflection**

The primary software testing technique I employed was unit testing, which allows for the verification of individual components of the software. Unit testing focuses on ensuring that each function or method behaves as expected in isolation, making it a good fit for this project, where testing individual features like contacts, tasks, and appointments was crucial.

There are other testing techniques I did not use, such as integration testing and system testing. Integration testing focuses on how multiple components interact with each other, which could have been beneficial for ensuring that the contact, task, and appointment services worked well together. System testing, on the other hand, would have tested the application as a whole to simulate real-world use, ensuring that the entire system functioned smoothly from the user’s perspective. These tests could have been beneficial, especially in larger, more complex systems where interactions between components are crucial.

Each testing technique has its own strengths. Unit testing is quick and efficient for verifying isolated components, making it ideal for small-scale applications or when specific parts of the system need validation. However, integration testing and system testing are crucial for larger projects where the interaction between different parts of the system can introduce unexpected issues. By understanding the strengths and weaknesses of each technique, I can make more informed decisions about which testing methods to apply depending on the scope and complexity of the software being developed.

**Mindset**

Throughout this project, I maintained a careful and methodical approach to testing. As a software tester, it’s important to consider all possible inputs, even those that might seem unusual or unlikely. For example, when testing the appointment service, I made sure to verify that the system wouldn’t allow users to schedule appointments in the past. This attention to detail helped ensure that the software would function as expected, even when users didn’t follow typical usage patterns.

It’s also important to avoid bias when reviewing code. It can be easy to assume that the software works as expected, especially when you’ve written it yourself. However, I made a conscious effort to test the application from the perspective of the end user, considering scenarios that I might not have initially thought of. For example, when testing the task service, I questioned how the system would behave if the task description exceeded the allowed character limit or if it contained unusual characters. This mindset helped me uncover potential issues that might not have been immediately obvious from a developer's perspective.

Finally, I recognized the importance of discipline and attention to quality in software development. Cutting corners or skipping tests might save time in the short term, but it can lead to technical debt in the future, which can result in costly fixes and unreliable software. For instance, if I had skipped writing unit tests for the task service, I could have missed a bug that might have caused task descriptions to be saved incorrectly. To avoid this, I made sure to follow best practices and write comprehensive tests, which helped me identify issues early and ensure the long-term reliability of the software.

**Conclusion**

In conclusion, my unit testing approach for this project was effective in validating the key features of the mobile application. By using unit tests to check the contact, task, and appointment services, I ensured that the individual components of the system functioned as required. Although I focused primarily on unit testing, I recognize the value of integration and system testing and plan to incorporate them into future projects for more comprehensive testing. Maintaining a careful, unbiased, and disciplined mindset throughout the testing process allowed me to identify and address potential issues early, ensuring the overall quality of the software.