**Project Two**

Michael Foster

Southern New Hampshire University

CS320: Software Test, Automation QA

Eidan Rosado

8/24/2025

The development of the mobile application consisted of three features; the first was an appointment service, the second a contact service, and the last was a task service. Each one of these features requires an object class for instantiating the object for that service, and a service class to manage the service as well as each instantiated object. With the requirements given I quickly realized that using a Test-Driven Development (TDD) approach would best suit my development. This tied the software requirements directly to my approach and can be seen in the test cases I designed. An example of this can be seen with the requirement that the contact object was required to have a last name filed that was a String and could not be longer than 10 characters or null. To ensure this requirement was met I created a test case that verified that a String that was 10 characters long was instantiated, as well as when a String of 11 characters or a null value was passed the instantiation failed. This pattern of checking that a method failed when passed invalid options as well as testing for successful outcomes with variables within the specified parameters can be seen throughout my JUnit tests.

When creating unit tests the goal of every tester is to create technically sound and efficient tests. In software testing it is not feasible to test every possible condition and therefore depending on how critical a part of a system is to the overall functional will dictate the level of testing. Typically, a coverage rating of 80% or higher is considered reasonable. (García, 2017) Analyzing the coverage of my JUnit tests they meet this standard. My contact service and task service both had a coverage rating of 100% while my appointment service had a coverage rating of 96.2%. Part of the challenge is to achieve a high level of coverage efficiently. A good example of how I kept my unit tests efficient can be seen in my task object’s unit test. To improve the efficiency of the unit test I created test cases that tested similar specifications found in multiple requirements. An example of this can be seen below:

A computer screen with text on it

AI-generated content may be incorrect.

To ensure my code was technically sound I created tests cases that tested that a method would fail when expected but pass when expected. A good example of this was in my contact test object where a test case verifies that instantiation will not occur with invalid parameters passed to the constructor but will occur when valid parameters are passed. This example can be seen below:

A computer screen shot of a program

AI-generated content may be incorrect.

In the creation of this application, I primarily used white box testing using unit testing. White box testing is testing that is conducted with knowledge of the code base. To create the unit tests knowledge of both the test cases and code base are necessary so methods like constructors can be passed the correct types of variables. Test methods that were not utilized in this process included both functional and nonfunctional testing. Functional testing not conducted included integration testing and system testing. Integration testing is used to validate that different subsystems interact in the expected manor. In a project like this it would be seen if verifying that the calling system was able to properly utilize the microservices we created. System testing is done to test if the complete system meets the requirements. As we only created a small portion of the application and the overall system is not finished, we are unable to conduct system testing at this time. Nonfunctional tests not conducted included performance testing and regression testing. Performance testing is conducted to ensure that the system performs well in multiple ways including that the system can scale effectively and that the user does not experience high latency. This type of testing would be done when testing the application on actual mobile devices as nonfunctional testing is more focused on user experience, and therefore how the product runs in the development environment is not an accurate picture. Regression testing is conducted to make sure that new additions to an existing deployed codebase do not introduce new defects. This was not conducted as the code that was designed for the initial release of the product.

When in the role of software tester there are a few things to be conscious about when it comes to your mindset, the first is caution. When testing an application, it can be easy to fall into the trap of believing you understand exactly what the end goal is from a quick read of the requirements. To build the best test cases you truly need to understand what the end state that must be achieved is. The second and possibly most important is your bias, especially if you wrote any of the code being tested. If you wrote the code, it will be extremely difficult to not write test cases based on the code you wrote. It is important when writing test cases to not make assumptions about what the end state should be, and how the code should get to the end state. The focus should be on validating that all the specifications in the requirements are met. The last trait is discipline; testing software can be quite a tedious task, which is problematic as it is arguably the most important part of the development process. To ensure that tests are created correctly it is important to document work, and verify as each step is completed, from identifying the specifications in the requirements to framing those specifications in something like a user story. Then actually writing the test case and when writing the test case ensuring that both valid and invalid conditions are tested. In the end all three mindset traits are intertwined and if the tester is conscious of their mindset when working they will greatly reduce their technical debt.

**References**

García, B. (2017). Mastering software testing with junit 5: Comprehensive guide to develop high quality java applications. Packt Publishing.