7-2 Project Two

Marisa Kuyava

CS 320: Software Test Automation & QA

Southern New Hampshire University

February 14, 2023

**To what extent was your approach aligned to the software requirements?**

While working on this project I made sure my testing approach was 100% aligned with the software requirements provided by the client. When creating the test, one test was created for each of the requirement scenarios in each class to ensure all requirements in all classes were satisfactory. Tests were created for each class to ensure objects could successfully be created and testing was done to ensure all error handling for invalid input works correctly and efficiently. Below are some examples of these tests from the AppointmentTest, ContactTest and TaskTest classes.

Text

Description automatically generatedText

Description automatically generated Text

Description automatically generated

**Defend the overall quality of your JUnit tests.**

The JUnit tests in all three of the classes (Contact, Task and Appointment) were written to provide 100% coverage for all outlined class requirements. Below are examples of test classes for Contact Service, Task Service and Appointment Service, all showing 100% coverage.

* Text

  Description automatically generated
* Text

  Description automatically generated
* Text

  Description automatically generated

**How did you ensure that your code was technically sound?**

To ensure my code was technically sound I followed best practices for commenting, formatting, and code annotation, including keeping variables names simple and meaningful.

* Simple and meaningful variables names

Text

Description automatically generated

* Clear, easy to understand commenting.

Text

Description automatically generated

**How did you ensure that your code was efficient?**

Utilizing DRY code and keeping my code as simple as possible. To ensure my code met the requirements and functionality appropriately JUnit tests were written for each class to test the requirements and to ensure error handling is don’t correctly and efficiently. Below are examples of tests that I created to check the requirement functionality of the Contact Service, Task Service and Appointment Service classes.

* 
* 
* 
* 
* 
* 
* 
* 

**What were the software testing techniques that you employed in this project?**

White Box Testing, specifically Unit Testing is the software testing technique that I utilized while working on this project. I used Statement Coverage to ensure all executable statements were run, tested, and functioned correctly. I also used Branch Coverage to ensure that for each of the conditional statements ever branch was tested and functioned appropriately. These techniques allowed me to confirm that the software function as expected and verify that the client requirements were all met in the software that I built.

**What are the other software testing techniques that you did not use for this project?**

Experience based software testing techniques were not used for this project. Experience based software testing techniques include Error guessing, Checklist-based testing, and Exploratory testing. Because I had adequate specifications to build my specification-based test and I have no experience with these testing techniques none of these were utilized.

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

White box testing techniques can be used when the functionality and requirements of the software are fully known. White box testing is beneficial because tests can be utilized in early stages of development and testing does not require a functioning user interface. Experience based testing techniques can be used whenspecifications and/or requirements are lacking, or time is limited. In Experienced based testing the focus is place on areas that are most likely to have errors or failures.

**Assess the mindset that you adopted working on this project.**

My mindset while working on this project was that of a student, or learner. Adapting a learning mindset is important when I am learning something new as I know there are times when I will get frustrated and want to give up. Things don’t always work right on the first attempt, or second or tenth, there will be times when the concepts don’t make 100% sense right away, but if I keep trying, learn from any errors along the way, and find more learning recourses I can find success. Prior to this course I really did not have any experience with JUnit or really any type of testing, so looking at things from a Software Tester mindset was new to me. As such I utilized a lot of caution to ensure that all aspects of the project are coded correctly and that I fully understood the requirements of what I was being asked to code and test. To ensure all classes worked correctly with each other and the JUnit tests were testing appropriately it was important to understand the interrelationships of the code and complexity of the project that was being built and tested.

**Assess the ways you tried to limit bias in your review of the code.**

By creating tests for the functionality of each requirement and testing for both valid and in valid input I was able to help limit bias. In testing the code and running the test multiple times I was able to confirm though the tests that the software was functioning according to the client requirements.

**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?**

It is extremely important to be disciplined in commitment to quality as a software developer to produce code that is effective and efficient. Providing quality code will also be providing code with less bugs and code errors. Cutting corners when writing or testing code can result in errors and bugs being found later in development, or even worse after deployment, which can result in increased and unforeseen costs and consequences to repair the issues. To avoid technical debt, I intend to follow best practices while coding and testing, use Agile practices and strive for quality in everything I code.

**Resources**

Hambling, Brian, et al. *Software Testing : An ISTQB-BCS Certified Tester Foundation guide - 4th edition*, edited by Brian Hambling, BCS Learning & Development Limited, 2019. *ProQuest Ebook Central*, <https://ebookcentral-proquest-com.ezproxy.snhu.edu/lib/snhu-ebooks/detail.action?docID=5837074>.