# Project Two

Jacob Burchett

CS 320

Professor Karl Lewis

21 April 2024

# Project Two

As developers, we should place heavy emphasis on the requirements as laid out by the customer. By placing this emphasis on the requirements and taking the extra steps as necessary on an individual basis, we can eliminate risks and reduce the total workload by removing many potential errors from the equation. For me, in order to ensure that my work aligns with the requirements set, I find it best to make a bulleted list, readily available on my second monitor. By doing so, I can work for 20-30 minutes and then look over my list. This provides constant reminders of what it is that I am doing and helps me avoid missing any of the essential components of my work. When it comes to coverage percentage, there is no absolute answer. Depending on a case-by-case basis, coverage can vary, and ultimately, it is up to the developer and their experience to decide what is correct. However, as a new developer, my goal was the widely recognized 80% in order to feel satisfied with my work.

With the assignment, I believe that I could have done a better job in testing for failing scenarios. In doing tests of this nature, you can identify issues that you may not have initially thought of. In turn, this can lead to improved code quality. Regardless, I still believe that my code is both sound and efficient.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

The above screenshots show the testing of new contact creation as well as the deletion of the contact within ContactServiceTest.

When it comes to testing, we can separate them into manual and automatic means (javatpoint, n.d.). With the assignment given, we can also say it is reasonable to place focus on white box testing. White box testing focuses on the internal structure and design of the code rather than black box testing, which focuses on the output without practical knowledge of focus on the code itself (GeeksforGeeks, 2023). More specifically, some of the techniques utilized for the assignment were unit testing and integration testing. Unit testing refers to the test of a specific part of code, such as a specific class (Schmitt, 2021). Integration testing considers how different piece on a system interact and/or work together. Other testing methods include usability testing and performance testing. Usability testing focuses on the functionality of the system in regards to user interaction (hotjar, 2023). Performance testing is a means of testing how a system handles a workload (opentext, n.d.).

The mindset that I had to adopt when working on the assignment was a new one for me. As someone with no experience in testing prior to this class, the entire process was new to me. I struggled with this at first. Turning to the internet, specifically stackoverflow, I eventually fell into the grove of being able to write code and tests without personal bias as the one writing both. A personal bias like that could lead to a decrease in quality as you may not want to change of adapt to work you have done when testing doesn’t go as planned. As previously stated, we always need to place emphasis on what the customer wants. When we go in with a bias for our own work, it may be hard to deliver on a system that is precisely what the customer wants. Therefore, it is important to move away from that mindset that your work is too good or more important so that the best possible work that aligns with the customer’s requests can be delivered.

References

GeeksforGeeks. (2023, December 27). *White box testing - software engineering*. https://www.geeksforgeeks.org/software-engineering-white-box-testing/

hotjar. (2023, October 9). *Usability testing: What it is, benefits, and what it isn’t*. Usability Testing: What It Is, Benefits, and What It Isn’t. https://www.hotjar.com/usability-testing/

javatpoint. (n.d.). *JUnit test case example in Java - Javatpoint*. www.javatpoint.com. https://www.javatpoint.com/junit-test-case-example-in-java#:~:text=The%20JUnit%20test%20case%20is,executed%20without%20using%20any%20tool.

opentext. (n.d.). *What is performance testing?*. OpenText. https://www.opentext.com/what-is/performance-testing#:~:text=Performance%20testing%20is%20a%20non,up%20under%20a%20given%20workload.

Schmitt, J. (2021, December 3). *Unit testing VS integration testing*. CircleCI. https://circleci.com/blog/unit-testing-vs-integration-testing/