Christopher Anderson

Prof. Toledo Lopez

Software Test, Automation & Quality Assurance

12/6/2023

Project Two

Summary And Reflection Report

Throughout the development of this mobile application, I have focused on maximizing JUnit test coverage and being disciplined to software testing techniques. This software package has been developed to tailor specifically to the software’s requirements and is tested to ensure each system constraint is enforced. The most vital testing technique utilized for this project has proven to be boundary analysis, although assumptions could’ve been a major part of testing procedures. When functioning as a software tester rather than a software developer, it is paramount to pay close attention to your mindset to ensure that you have an unbiased commitment to deliver the highest quality testing possible. The success of a software product requires that developers strictly employ software testing techniques, and that regular testing is conducted with the maximum coverage for JUnit tests.

The client’s description of the software’s requirements was utilized as the guidelines for all JUnit tests and class development in this mobile application. While implementing JUnit tests, I focused on the restraints imposed upon classes by the system’s requirements and I’ve designed tests to confirm that these restraints are enforced, e.g. all classes in this system have variables that cannot be null and have a maximum length therefore I’ve developed tests specific to the variables’ maximum lengths and the possibility of encountering a null value. Based on JUnit coverage of lines of code, on average greater than 75% for each class and service that I’ve developed, I know that at least 75% of code for each class is being activated by the JUnit tests that I’ve implemented hence I am confident that my tests are checking these classes for all possible weaknesses. I have confirmed the validity of my code by avoiding the application of redundant error flags, tests and database iterations while also safeguarding critical functions where new data is introduced to class instances. This mobile application is a union between JUnit tests and classes developed through the guidance of the system requirements as described by the client.

For this mobile application’s development there were several highly potent software testing techniques that I utilized to ensure that testing was of an exceptional quality. Boundary Analysis is the most valuable testing technique that I’ve encountered, since this technique involves testing values that are along the boundaries of the software package’s constraints, e.g. when implementing the Task Class’s description variable I performed Boundary Analysis to create a test case that attempts to assign a 51 character String, due to the maximum length of a Task description being 50 characters, knowing that this String should trigger an illegal argument flag. Another interesting testing technique that I did not utilize for this project is Assumptions; this technique allows a developer to test initialization functions for the assumption that there is standard execution before performing the desired test. It is essential that these software testing techniques are properly implemented since Boundary Analysis is the practice of testing values along the edges of a system’s constraints to confirm that system boundaries are respected and if Assumptions are utilized there must be genuine understanding of how to incorporate this technique to verify that initialization operations for a test have been properly accomplished. Software testing techniques are highly potent tools that have guaranteed the quality of this mobile application.

When acting as a software tester rather than a software developer it is vital to manage your mindset to ensure that high quality testing is achievable. To allow for the highest quality testing it is paramount to consider how each service class applies to each object class and how the service classes will interact with external interface systems, this comprehension requires a great deal of caution since the designs for the UI and system interface have yet to be introduced so we must anticipate where bad data could be introduced and where illegal class access may be performed. I have worked to limit my bias toward my code by humbling myself with the assumption that my software is innately flawed, but should I not develop tests with this assumption I would have weak test cases since I would believe that my software cannot fail, e.g. during development I implemented many code statements to end operations when faulty data is introduced but I tested for these situations to verify that these statements were activated when necessary. The most important characteristic of a software tester is being disciplined since the development of test cases and the implementation of test functions can be time-consuming, but these operations are possibly more valuable than the development of the software itself since testing will confirm that the software achieves the desired goals while recognizing the system’s constraints. As a software tester, I will carefully develop high quality tests to ensure that my software reaches fruition with minimal flaws by focusing on the cultivation of a disciplined unbiased mindset.

Software testing techniques and JUnit test coverage have been paramount features in the successful development of this mobile application. System requirements as outlined by the client have been analyzed to produce system constraints and have been the mold to which the software has been developed to fit. The most significant software testing technique that I’ve employed for this application is Boundary Analysis, though there are opportunities for Assumptions to be introduced that would allow for greater confidence in test function actuation. When performing software testing it is vital to emphasize the importance of maintaining a disciplined and unbiased mindset to allow for the greatest quality of testing that will allow you to find any, if not all, flaws within your software product. Throughout the development of this mobile application I have utilized JUnit test coverage and software testing techniques to validate that this software package has fully matured.

Reference List

Garcia, B. (n.d.). *Mastering Software Testing with JUnit 5: A Comprehensive, Hands-on Guide on Unit Testing Framework for Java Programming Language* (S. Thakkar, N. Fernandes, M. Yusuf Imaratwale, R. Rosario, & C. Carneiro, Eds.) [Review of *Mastering Software Testing with JUnit 5*]. Packt Publishing Ltd. Retrieved December 6, 2023, from https://eds-s-ebscohost-com.ezproxy.snhu.edu/eds/detail/detail?vid=0&sid=e0c900cb-5e7b-4c34-b5f3-0c1ec0d6c62b%40redis&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=1626950&db=nlebk (Original work published 2017)

Hambling, B., Morgan, P., Samaroo, A., Thompson, G., & Williams, P. (2019). *Software Testing : An ISTQB-BCS Certified Tester Foundation guide* (B. Hambling, R. Youé, & D. Bannerman, Eds.; 4th ed.) [Review of *Software Testing : An ISTQB-BCS Certified Tester Foundation guide*]. BCS Learning & Development Ltd. https://ebookcentral-proquest-com.ezproxy.snhu.edu/lib/snhu-ebooks/reader.action?docID=5837074&ppg=6

‌