Jamie Javis

08/13/2023

CS 320

7-2 Project: Project Two

Summary

My process was totally as per the project requirements since I saw that following the assignment requirements as a very key importance to this course. We should look at the contact class. The first and last name can't be invalid, and in this manner cannot be more than 10 characters in the string value, as per the given requirements. In our ContactTest class, the JUnit test had a declaration to see whether the information tested if the password used exceeded 10 characters. As per the requirements of the task class, the task ID/interesting ID couldn't surpass 10 characters. Since the task ID might be in excess of 10 characters, we involved a capability in our TaskTest JUnit test that would show a failure, allowing myself to know that something was wrong in the code or through the test values.

The assets made accessible in every module enormously added to the coverage percentage, which brought about a general improvement in the nature of my JUnit tests step by step. In contrast with the coverage percentage in the contact test, the task test had a somewhat low coverage rate, and the service test had the most advanced coverage rates. Any time a positive coverage % was shown, I realized that the JUnit tests I was running had effectively covered a large portion of the code's capabilities.

I used multiple strings and other resources to ensure my code was actually strong. As a visual, consider the contact class. To monitor the contacts, I made a rundown for the strings. An illustration of this is:

A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

Thus, this would simplify it to execute my JUnit tests and add, updated, or delete contacts.

I found a way different way to ensure my code was viable. I have some essential knowledge on coding from prior courses that were taken prior to this one at SNHU, however at some points I feel like I need a deeper understanding, I go to Discord, Reddit or YouTube for the most part. YouTube has a wide assortment of videos and would be there to help whenever, while Reddit and Discord require people to interact with me to help me understand topics and subjects. I also like running my code every now and again to ensure there aren't such a large number of issues as I go, which will assist with guaranteeing that my code is compelling. Typically, I find testing the code after each section and adding print lines will help ensure functions are working properly before going onwards. Moreover, I made consideration to declare every variable prior to utilizing it. As an example, consider the Contact class. I declared every one of the variables that connected prior to contrasting it with the required information.

A screenshot of a computer screen

Description automatically generated

Reflection

Each project now and in the future should have testing to help find success as well as failures. I utilized an extremely useful tactic while engaging software testing techniques for this project. I felt like I had a test for everything as I went over the requirements and my code, for example, the number of characters a first and last name that can have and that neither one of the names can be invalid. I took care as to name my tests in the JUnit tests in a way that was very like the names of the main classes they were testing for. Everything was held unbelievably coordinated thanks to this strategy. The step-by-step technique was another software testing strategy I utilized. Where I ensured I didn't miss anything by going through every model each in turn. I then made my test cases after I was sure that every requirement had been recognized by JUnit. To complete any logical and structural testing, I also utilized White Box Testing as well as JUnit testing.

Static testing is a technique for software testing that I did not utilize for this project. Static testing includes investigating code without running it to track down any issues. I never let my code be tested externally; I generally run it too. Nevertheless, I like to run the debug or build modes so I am able to look for my errors faster. I realize that static testing is centered around finding issues in time as a defense through walkthroughs, code examinations, and peer reviews.

My viewpoint has truly changed from when I began chipping away at this project. Honestly, I have no genuinely past information on any sort of software testing from before attending this school. I really trusted that "testing" alluded to simply running the program to check for issues at the base. Going about as a software tester is something I would characterize as devoted as differed to being just cautious. We can't take any easy routes to finish a program quicker and we normally need to ensure that the last result fulfills all client requests, we should continually guarantee that the eventual outcomes quality and usefulness are not compromised.

In the event that you composed and tested your own code, I understand the reason why somebody could think there may be a personal bias. The least complex technique to restrict predisposition, as I would like to think, is generally to keep up with modesty and to be available to change and allow learning opportunities. There is no opportunity to get better or for tracking down errors if we test our own code, naturally expecting that we are the best software engineer on the planet. Our eventual outcome could suffer accordingly. As an expert in software engineering, discipline is essential to my commitment to greatness. We do not just stick to a strict set of software engineering morals, however holding back on the creation, and testing of code is never adequate.

Work Cited

Introduction to code based testing and its importance. (2023a). Retrieved from http://www.browserstack.com/guide/code-based-testing#:~:text=Code%2Dbased%20testing%20is%20a,vulnerabilities%20before%20deploying%20the%20software