CS-320-T2824 Software Test Automation & QA

4-2 Journal: Unit Testing Approach and Writing Junit Tests

Eric Wallace

Nov. 21, 2022

During the milestone this week, I followed the testing requirements to 100% code coverage. Take for instance, each function that is used to update the name and description were tested for null values and length violations to which both passed the tests. I did write one method that I did not write last week and that was a general updateTask method which updated both fields. Each method used except for the generateTaskID method was tested, the reason why the generate method wasn’t tested was because it is a private method used by the service class for inserting a new task, it generates a random string that is 10 characters in length and does not need testing.

Both the contact and task services had 100% code coverage. I verified that all methods for each service had both a null and invalid length test to ensure all angles were covered during testing. I know the Junit tests were effective because first and foremost, the tests were run and returned no errors. Another thing I did in the module for this week is I streamlined the testing by including multiple tests for different methods. An example of this is when I tested the update methods, I included multiple test criteria within a test. This reduced the number tests needed to complete the requirements. It also reduced the amount of time I needed to write the code for the milestone.

The first piece of code that ensured each service was technically sound would have to be the ID generators. By setting those methods as private it increases the level of security and minimizes errors as it removes the human equation from the process. Another way I made sure all fields were accessible by the service classes and each method had error handling setup. This decreases the chance incorrect values.

How I ensure that my code is efficient is by following industry standards and guidelines for naming conventions. I also use plugins that help with making sure my code does not have any security vulnerabilities and adheres to industry standards.