# CS 320 Project Two

**Victor Landi**

**Southern New Hampshire University**

[**CS-320-T4512 Software Test Automation& QA 22EW4**](https://learn.snhu.edu/d2l/home/1009005)

**Instructor Alex Pettit**

**April 2022**

The unit test done for the three modules were the same methodical approach as

outlines in *Software Testing* (Hambling, 2015). For the file ContactService.java, to be able to delete contacts this was addressed in lines 23-33 of my unit test. The code was first run, then it ran the deleteContact method, which used the ID number per the requirements. For the Task class, two tests were run one to ensure the String field was not longer than twenty characters. The second was to ensure the name field was not null. Lines 34-39 and 41-46 for the task files. For the appointment module, tests were done to make sure the appointment date was not in the past.

Based on the coverage percentages , I was satisfied that these Junit tests ran on all files and situations satisfied the original requirements of the assignments. I received percentages of 90 and 100 percent on the tests and I can feel confident that I satisfied the client requirements and was able to prove that my code worked, and the tests worked.

The use of Junit testing provided an excellent method to test the code requirements for Contact and Task services. If for example I wanted to see if the parameters were met and would pass or fail for say the contact object, I would change the entered values. If I wanted to test if for example, the contact object could not be null nor have more than ten characters long, I would make the value null and run the Junit test. Then I would change the input to more than ten characters and run the test again. Then I would make the values fall within the accepted limits and run the Junit tests again. I saw that the coverage percentage of 100% after running the required test showed me that I could be confident that the requirements were met, and my code worked and was what the requirements called for. Accuracy, as in all things is vital in coding. If for instance, I misunderstood the proposed requirements, I could run the Junit tests, get 100% coverage but would have failed to meet the intended requirements of the client. It is so important to ensure a clear understanding of the requirements before starting any project or task. I see why Junit testing has become the best testing framework and the standard for testing the Java programming language. Most IDEs support it. As stated in blog from Parasoft, “JUnit is a Java unit testing framework that is one of the best test methods for regression testing. An open-source framework, it is used to write and run repeatable automated tests” Parasoft (2022).

Some other testing methods learned are Black Box Testing based on the requirements and functionality. The internal system design is not considered in these tests. In White Box testing, all the internal system design is known and tested. Functional testing tests the behavior and output, rather than testing small units of code. Regression testing can be performed on systems already deployed in real time environments. Junit testing is considered white box and regression testing.

Upon reflection of the experience of writing code and testing in this course it has become clear that Unit testing must always be done end to end to cover the smallest part of the code. Testing needs to be done for all input possibilities and for all exceptions when the code is executed. Test cases need to be thoroughly reviewed and examined by the team. A test environment needs to be created and reused every time for all units or sprints, and whenever change is introduced into the system. One only must look at all the costly, deadly, and wasteful disasters that have occurred thorough the history of software in industry to see the importance of clean, and well tested code. This is a practice which is as important as writing the code itself.

The implementation of Unit testing at every level and at each step of the software development lifecycle will lead to better product development and compliance with timelines. It will also lead to better client and customer relationships. It will reduce the about of bugs in the product and can increase profit and return on investment for the client but taking the time and putting in the effort to test the work. You would not paint a work like the “Mona Lisa” if you were not sure the paints and canvas were compatible and the painting would dry properly and not run, now, would you?

**References:**

García, B. (2017). *Mastering Software Testing with JUnit 5*. Packt Publishing.

Hambling, Brian Morgan, Peter Samaroo, Angelina Thompson, Geoff Williams, Peter. (2015).

*Software Testing - An ISTQB-BCS Certified Tester Foundation Guide (3rd Edition).* BCS

The Chartered Institute for IT. Retrieved from  
 <https://app.knovel.com/hotlink/toc/id:kpSTAIST01/software-testing-an-istqb/software->

testing-an-istqb

https://www.parasoft.com/blog/junit-tutorial-setting-up-writing-and-running-java-unit-tests/