Jasleen Kaur

Southern New Hampshire University

12/10/2023

**Report**

My testing was closely aligned with the software requirements because they were addressing both function and nonfunctional aspects. This means in the contact service the requirement was that the system should be able to handle a range of inputs for contact details. My tests covered scenarios involving different input formats. Phone numbers, names and complied.

The tests demonstrated high quality coverage across the codebase. Using a coverage analysis tool I found that the tests covered 90% of the functionalities in both the contact and task service.

For instance, in the task services a critical requirement was handling user access to tasks. I did that by simulating the excess in the test and validated that the system maintain data integrity and consistency under the requirements for example I used public void testcontact() [

In the contact sercvies there was a need for good search functionality. To do this I designed tests that are able to help with algorithms accuracy while maintain effieicty For example I used public void testContact String contactID,String firstName,String lastName, String phoneNumber, String contactAddress)

The techniques I used was unit testing because it uses Characteristics which test individual units for accuracy and practical uses which test early bug detection, and it is very important for code integrity. I also used integration testing which ensures seamless component interactions and verifies. Another one is regression testing which reruns tests and prevents post change issues.

I did not use performance testing which evaluates system performance and security testing which identifies vulnerabilities in system security.

While working on this project the mindset I adopted was being very careful and paying attention to detail. I was making sure each line of code is correct and done properly with little to no mistakes. For example, after writing each line of code I would debug and make sure I do not debug all of the code in the end together. I was being very precise with each line of code. I was not biased with reviewing my code, I was very careful, and I used all my sources provided to make sure my code was done the correct way. It is very important to be disciplined with my work as a software engineer since this should not be taken lightly. Being careless can affect my work and potentially the company I work for. A lot of destruction can take place without carefully reviewing my work. Some ways to avoid technical debt is to take things one at a time and if you have a team have them look over your work and never cut corners.