Project Two

Alexis Scott

Southern New Hampshire University

CS-320 Software Test Automation

Project Two

I have faced many challenges while developing the mobile application for the customer. I feel that overall, my testing approach was pretty well aligned with the software requirements. By following the requirements, I was able to develop test that showed if my code worked properly or not. One example of this is that the requirements stated that the phone number had to be exactly 10 characters long and not empty. I was able to set up a test that passed a too long of a phone number through the code and see if it would properly work and catch the error. I think that the overall quality of my JUnit tests are pretty good. I know this by testing the coverage percentage multiple times and fixing things that seem to not work. I try to get my coverage percentage as perfect as I can but unfortunately, they are not 100%.

Even though I think things turned out fine in the end, I feel like I struggled a bit with my JUnit test. With this being my first time writing JUnit test I was quite unsure about what I was doing at first. By using the resources that I was given I was able to put together proper code following the requirements given. I ensured that it was technically sound and efficient by making sure it ran properly. To make sure that it ran properly I wrote multiple test that tested different scenarios where the input should not be accepted.

Throughout this project I mainly used boundary value analysis for most of my JUnit test. I would often input a string that would be accepted because it was right at the limit that was set. I would also input a string that went way beyond that limit to see if it would throw an IllegalArgumentException, this helped check multiple things.

While working on this project the mindset that I adopted was on that was very careful and aware of what was needed at the time. I think it could be very easy to get a little to carried away while being a tester. You could write test that are not needed causing extra work or you can even do the opposite and not write enough test causing the project to go through not fully knowing if it works properly. It is best if a tester follows the requirements needed and write the test that will suit the certain project best.

It can be hard to limit bias while reviewing code for either yourself or someone who is a close coworker. Personally, I think it is best to have someone else testing your code rather than testing yourself because someone else may find something that you may overlook in your own work. Also, when we make mistakes it can be very difficult to see that in our own work but if you did have to be responsible for your own testing, I think that it is best to stick to the requirements that are given to you in the beginning.

I think that being disciplined in your commitment to quality as a software engineering professional is very important. Writing good quality and well tested code can be very time consuming and very draining to some. Though this may be true it is important not to cut corners when it comes to writing or testing code. Sometimes the information that is taken in by the code we write can be very personal and quickly writing or testing code could cause the information to be wrong or even worst it could be leaked to untrustworthy people. By making sure that everything is the way it needs to be and making things better for the future users it is important to put in the extra to make sure things are correct.