Project Two

CS-320

Southern New Hampshire University

Alyssa Navarro

In this project, there were three features I worked towards developing. This includes developing the contact, task, and appointment services for the mobile application. In order to develop these features, I needed to incorporate unit tests. The first step was to write a test for the attributes in an object so if there was any input that didn’t meet requirements then errors would pop up in the application. There is an example of this in my code for the Contact object, where I made sure that any input that is “null” for a field would cause the program to fail. To make sure my bases were covered, and the program would run as it was supposed to I made tests for the good and the bad. There are tests for the bad inputs that result in errors and tests for good inputs to make sure it would not crash and would run properly. This makes sure all requirements were met and the program would run without errors. For the services, tests were made for each method that was listed in the requirements. By doing this it made sure that the service methods were going to produce the correct output and results.

A way that I ensured my code was technically sound was creating problems in the code on purpose to see how the program would react. If the application did not throw any errors, then there would be something wrong with my code. By adding the problems, it was effective to test and see what would happen if someone else were to enter something wrong. I was in the shoes of an application user in that respect because I was testing out the application. There is a good example of throwing in a mistake in the ContactTest.java file from Project One. It shows that the passing of “Null” in ContactID would cause an error to occur in the program. This would not allow the user to continue or proceed without putting in a valid ContactID that met the requirements.

The primary technique I used to test the objects was the technique of boundary value analysis. Using this technique, it shows errors at the boundary values, and it doesn’t just test only common or expected inputs. This is a more realistic way of testing because there is no saying what users might insert to fill in the field. The technique also works to check that the boundaries were set or if they were even set correctly in the objects that were being tested.

I believe that the testing mindset was important in this project. It was important to always test the code being worked on to make sure you weren’t building up on something that was full of errors. If each object was written without being tested, then the final product would not run at all with the different syntax or runtime errors that could’ve been unnoticed in the code. I personally like to test code every step of the way and make sure to run the debugger. Using the debugger goes line by line to make sure things are okay and if not then it highlights where exactly the code goes wrong. No matter how good someone is at coding there can be simple mistakes that are made. So I believe it is crucial to always check your work because if not then your code could be so corrupted that you won’t know where to start to fix it and most likely might start all over.