Project 2

Denise Kendall

When testing my code, I would test to see that the application worked as intended and what a user should except if the requirements are not met. When a user is creating a task, it checks to make sure that is not longer than 10 characters and then checks the array list to make sure that it hasn’t been used. If the task meets all the requirements a task is created, but if it isn’t met then an error message will appear. This applies with names, phone number, address, and descriptions. The quality of my test is good because covers a high percentage. The coverage checks to make sure the requirements are met and that each test has passed. To know my code is technically sound, I made sure all the requirements are met and has a solution for each situation that the user may run into. If all the requirements are met, then the program will continue as normal and if it isn’t met then a message will inform the user. For example, if a contact ID is already in the system when trying to add a new one or if the name is too long then a message will appear. I was able to see that my code was efficient because it didn’t repeat lines that weren’t necessary, such as the set and get methods. The set methods check to see the requirements are met and if they are then it will application will set the value. Since we created these methods, we don’t have to repeat the same code repeatedly. We can call the set methods instead. We also created loops to search the array list to find the correct contact and task ID.

One of the software testing techniques I used was unit testing. I tested the code to make sure it worked as intended, such as creating, deleting, and updating a contact. Another test I used was manual testing. I tested to make sure the coding worked as it was supposed to when encountering the same task service ID. If the same contact, task service, or appointment service has the same ID then it will return a message to the user instead of creating a new contact or service. A test that I didn’t use was load testing. This tests the application’s response to increasing demand. I did not feel like it applied at this stage. Another test that I didn’t use is penetration testing. This focuses on the application’s security like verifying the robustness. This is beyond my expertise and our focus is to make sure the application functions as intended. However, this will be very useful for larger projects. The penetration test will make sure that the system can handle different types of attacks. Loading testing will be great for applications that will have multiple users or data that is being passed. Unit testing is beneficial for all types of projects because you check the functionality of the application and verify that it meets requirements. Manual testing can be useful when you want to test a specific situation such as the one, we are testing with this application.

I was cautious when it came to testing different outcomes. There are different messages that will appear if the user doesn’t meet the requirements to create a contact, appointment, and task. It is important to appreciate the complexity and interrelationships of the code because it lets you understand what you need to test. If you can understand how each part affects the other, you can know what kind of outcome you need to test. There can be a biased view when it comes to testing your own code. We hope that we don’t make mistakes and maybe we don’t test certain parts of the code. If there were fresh eyes that would test the code instead, they might see something that we don’t. It is harder to see things especially when you are constantly looking at something. Sometimes it is a good thing to take a step back and clear your head. When I was getting help from another person. They suggested testing what would happen if someone tried to remove a contact that didn’t exist. I didn’t think about that because I was focused on making sure the functions worked. It is always a good idea to be disciplined in everything you do. Quality is better than quantity. If you cut corners now, you will have to go back and fix it later. If you don’t have time to do it now, then you won’t have time to do it later. I plan to make sure that code is efficient and up to standards. I used the set and get methods to make my code efficient. If I was also the one to test the code, I would make sure that I tested all the possible outcomes. I would also have someone else look at my code and give me feedback. When I was creating the contact and contact service code, I had several people with more experience double check my work. They each were able to show me different and more efficient ways to write my code. This helped me write the appointment and task code.