Project Two

Mardelyth Arteaga-Garay

CS-320-11465-M01

August 25, 2024

My unit testing approach for all three features, contact, task, and appointment, was thorough and aligned with the software requirements, ensuring that each specification was met accurately. For the contact feature, the requirement stated that the contact ID must not exceed 10 characters, and the phone number had to be exactly 10 digits long. The task feature required that the name be no longer than 20 characters long. In the appointment feature, I made sure the appointment date was not a date in the past. By ensuring I adhered to the requirements, I was also able to make sure that the user wouldn't encounter any errors without knowing how to fix them. With a high coverage percentage, we know the JUnit tests were effective. To ensure that the code was technically sound, I made sure I followed all the specified requirements and adhered to best practices in software development. To do so, I tried avoiding code repetition and eliminating unnecessary lines of code. This made the code easier to maintain and more efficient. To ensure that the code was technically sound, I made sure to follow all the requirements that I was asked to include. I think that by following this approach throughout the development and testing process, I was able to deliver what was required of me and meet the users' needs.

Describe their characteristics using specific details. One software testing technique that I employed in this project is unit testing. Unit testing involves testing individual units or components of the software in isolation to ensure that each one functions as intended. An example of this would be when I tested the individual methods within the contact, task, and appointment classes to verify that they met the requirements.

I would run parts at a time and fix any problems as I encountered them which I think helped improve the quality of my work. I also implemented system testing to ensure the software met the system requirements. System testing involves testing the complete and integrated software system to ensure that it meets the specified

system requirements. The software testing techniques that I didn't use are usability testing and performance testing. Usability testing evaluates the software's user-friendliness and ease of use and performance testing tests the software to determine its performance characteristics such as speed, scalability, and stability. I didn't find myself specifically testing for these characteristics All these different techniques serve different purposes and for me some were more important that other while working on this project.

While working on this project, I adopted a cautious mindset. I tried to read over what was required of me carefully to fully understand and deliver. Also, while I am coding, I always try to be meticulous with my work because I find it easier to be meticulous as I work as opposed to having to go back and fix more mistakes. It was important to appreciate the complexity and interrelationships of the code I was testing to ensure no repetitiveness or duplicates. I tried to limit bias in my review of the code by always allowing myself to challenge my work and myself. Being disciplined in my commitment to quality is very important given I only want to put out work that is a reflection of it.

It is important to not cut corners because I find that if I do so, I always end up spending more time trying to fix any mistakes that I had originally overlooked.

My approach to this project was to create something of quality while adhering to the software requirements. I was able to do this by ensuring I was testing as I worked to fix errors as they appeared. The testing methods that I used also helped create something that is maintainable and efficient.

References

GeeksforGeeks. (2023, December 6). *Software testing techniques*. https://www.geeksforgeeks.org/software-testing-techniques/