Nickolas Askey

CS-320

August 13, 2023

Project Two Written Submission

**Summary:**

For this project, I was tasked with creating a contact service, task service, and appointment service class, as well as utilizing J Unit testing to ensure that each of these three classes were functioning properly, did not have any glaring errors, and perform up to the standard of the client. First was the contact service. For this, I created a contact, contactservice, contacttest, and contactservicetest class. Each of these classes were designed to effectively and efficiently allow the user to create a contact with a first name, last name, address, phone number, and a unique ID that is specific to the individual contact. During the testing phases of the contact class, I was able to cover well over 80% of the program and verify that the program was working properly and had little to no issues with the code itself. Following the contact service, I created classes for the task and appointment services and utilized a similar test coverage on each at well over 80%. These included task and appointment classes, taskservice and appointment service, tasktest and appointmenttest, and taskservicetest and appointmentservicetest. I was able to verify that the logic was correct in the programs and that every input given provided an expected output. I was able to test each of the variables in each program and made sure that every length was correct, every type of variable was correct, and every method returned the correct information.

During the initial process of creating a functional code to meet the requirements of the client, I had made an error in what was asked of me, and neglected to utilize JUnit testing, and instead used print lines to display errors. After having changed to the aforementioned JUnit tests, I realized just how important it can be. Instead of manually checking the program by using the public main to print out information, a JUnit test will ensure that each method within a class is functioning properly and will display a message if any of them fail to run. This was incredibly useful in the Test and ServiceTest classes in each of the different features since it made testing not only faster, but also easier. One big issue with testing the way that I had initially is that if there IS an issue with one of the earlier tests, it will stop the program and display the error message; JUnit testing displays all the issues within the class without stopping the program early. This can be super useful in testing, since it gives us a chance to see where all the issues lie, rather than only seeing one single issue at a time.

**Reflection:**

One focus of testing for this project was manual testing in which the programmer tests manually without the use of automatic testing. This would entail reviewing lines of code and checking for errors in the formatting and removing any typos with the naked eye without using a system to review the code for us. Reviewing the code in this way can greatly help reduce the number of errors that pop up during development. Utilizing this form of testing with JUnit testing ensured that the program ran and functioned smoothly while meeting all the client’s needs. Another focus was on functional testing in which we tested the functional requirements of the software to make sure that they are all being met. JUnit testing is a specific form of unit testing, and we also utilize acceptance testing since we are trying to meet the needs of both the client and end-user. Some testing areas that I did not utilize in this project are non-functional testing, in which the programmer tests non-functional requirements such as performance and security, and usability testing, in which the programmer tests the ease of use and user-friendliness of the program. While both are incredibly important, without a more complicated program or more information to work with, it isn’t nearly as important to utilize them for this assignment.

While working on this project, it was very important to employ caution. While this program wasn’t as complicated as other programs that I may work with, there are still aspects of it that need more careful and considerate thought. For example, in the contact class, each contact had a phone number, and the only listed requirement was for it to be 10 digits long. An easy mistake to have made would be for the phone number to be brought in as a string that is 10 units long but forget to ensure that the information entered as a phone number were numbers and not letters. Allowing for small mistakes like this to exist in code can lead to both upset clients and end-users, as well as easier access for hacking and unwanted activity in a program. So many methods in the program relied on other methods, so if there was a mistake in one of these, there was a good chance that they could all have been messed up, so a cautious mindset can really help prevent these errors from occurring.

Bias is another issue that can occur while creating a program. When you create a program, it can be hard to pick out the issues since you are the one who created it. Limiting bias can be detrimental in creating strong code, which is why so many programmers will have other programmers test their code before they move on to another step, since it is easier to see the issues with someone else’s code than your own. One way that I tried to limit bias in my own program was by using unit tests to examine the functionality of the code.

Lastly, it is important to practice discipline in our commitment to quality as a software engineering professional. The last thing that you would want if you hired someone to do a job for you is for them to cut corners; this is why we, as software engineers, should ensure that we are doing everything we can to not only meet the needs set forth by the client, but to completely surpass these expectations. Cutting corners on programs can lead to big issues down the line, like maintenance, costly issues, or program hacks. Using unit testing and practicing good software testing techniques and greatly improve the amount of discipline we have in our work, and also improve the public’s opinion on our work. Once you have the image of consistently cutting corners and not putting out the very best work you can, it can be a lot harder to find work with clients since they won’t want to work with you again.

Citations

Bose, S. (2023, May 25). *Software testing techniques: Explained with examples*. BrowserStack.https://www.browserstack.com/guide/software-testing-techniques

Neeru360. (2023, February 6). *Software testing techniques*. GeeksforGeeks. https://www.geeksforgeeks.org/software-testing-techniques/