Aakash Thapa

CS320 – Project Two

Dec 11, 2022

Summary

My approach was based on the requirements of the software. The goal was to satisfy each requirement by performing unit testing. As unit test helps to prevent bugs early in the process, we used the Junit testing tool for this project. My approach was to test only one code at one time, which makes things a lot easier. In the task object, the task id should not be null and cannot be longer than 10 characters, I used assertion to identify those testing requirements. The codes I used were

void tooLargeId() {

Task task = new Task(id, name, description);

*assertThrows*(IllegalArgumentException.class, () -> task.setId(null));

*assertThrows*(IllegalArgumentException.class,() -> task.setId(tooLargeId));

*assertEquals*(id, task.getTaskID());

}

We can see that when the taskid is null and has more characters, it will throw illegal arguments. The writing of the Junit test was slow at the beginning, with a lot of errors. It improved every week, slowly. The first test in the contact task was full of errors because I was not able to write perfect code and methods. Reading a lot and breaking things down into simple tests one at a time helped the code improve. The teacher also encouraged us by giving us plenty of time to correct codes when we were not able to do it that week. As pressure was applied to me, it provided me with additional motivation to work harder. I knew the Junit testing was working well by looking at the coverage percentage. I tried my best to keep the percentage higher than 80%. I ended up with my project at 87%. Also, in Eclipse, if the code was not tested or covered, it was shown in red colors, so this visual highlight was able to make me see clearly what was not covered during testing.

A lot of work had to be done to make the code effective. I made extensive use of the import function. For example, I had to import java.util.ArrayList in a few of the packages to use arrays for the list as it makes things easier. Similarly, I have to import java.util.date in the Appointment task to get dates. The best way to make code effective was through trial and error. I learned a lot by making mistakes. I had to use a lot of help from the internet and the books provided as well.

Reflection

The testing methods used in this project were dynamic to test both positive and negative requirements. A structure-based technique was also used in some cases to evaluate all possible outcomes. In these specific cases, multiple-condition coverage was used. The obvious methods that were not put to the test were based on experience. Because of my lack of knowledge, I find it difficult to use error guessing and exploratory testing procedures.

The mindset was poor at the beginning of the project, but with each success, it was put on the right track. It was very important to appreciate the complexity of the code because if one of the branches or any line of code was not tested, it would lower the coverage. It simply means the test is not good enough. I don’t think I was biased toward this project because most of the time I was struggling with writing code. Because of this struggle, I was biased toward only achieving the lower coverage percentage. However, this improved a lot when I was able to write better code. I think the bias depends on the moral values of the software developer. I don’t think just because the developer writes their own codes and tests themselves, they will be fair during testing. This is where discipline and dedication come into play.

The importance of discipline is immense and can’t be explained in words. We all know that the result of bad testing can result into loss of lives, finances, and create a mess into project which will create technical debt. All of this can be avoided if one can stay disciplined and committed. There was a time where I was about to give testing in many cases because I was not able to write correct functions, but my discipline and commitment helped me go through it. I believe one should not be hesitant to ask for help or feedbacks if they are struggling or not confident. It's better to ask for help than to cut corners.