David Anguiano

Professor Rangitsch

7-2 Project Two

6/7/2022

1. **Summary**
   1. **Describe your unit testing approach for each of the three features.** 
      1. **To what extent was your approach aligned to the software requirements?**   
         My approach ensured that all the requirements were met. There are validations in place to ensure that character limits are enforced. I also implemented Unit Tests with Junit to confirm that the validations work as expected. For example, within the Contact Service feature, I ensured that contactId length could not be longer than 10 characters. To confirm this I created a Test case called testFailContactAdd() with 11 characters for the contactId this test case’s assertion expects a false which means it did not add or create an entry, this test passes every time. For the Task Service, I ran similar test cases with character limits but I also set up testUpdate() and testDelete() test cases. The testDelete() test case uses the deleteTask() method which searches for a contactId that matches the input, if it finds an entry it returns true otherwise it returns false. My test cases returned the proper expectation when I ran assertions. Finally, for the Appointment Service feature, I ran test cases on the Date field to ensure that it did not allow past Dates to be entered I tested this using the testPassAppointmentAdd() Test Case which may need to be renamed.
      2. **Defend the overall quality of your JUnit tests. In other words, how do you know your JUnit tests were effective based on the coverage percentage?**  
         I believe that my Test cases were effective and guaranteed over 80% of test coverage. In all features, I made sure to test character limits, null entries in different fields, and any additional validations like past dates entry and update methods.
   2. **Describe your experience writing the JUnit tests.** 
      1. **How did you ensure that your code was technically sound?**   
         I ensured that my code was technically sound by providing comments on all my methods and the intent behind every portion of the code that was presented. In the AppointmentServiceTest Class test case starting at line 13 and ending at line 28 I provided comments that provide insight into the purpose of the code.
      2. **How did you ensure that your code was efficient?**   
         I ensured that my code was efficient by only including the necessary methods and variables within the code. In the testFailAppointmentAdd() test case, starting at line 29 and ending at line 39 I exhibit this efficiency by only using 10 lines including comments and spacing. When running this test case only takes 0.002s to complete one of the lowest times in my test cases.
2. **Reflection**
   1. Testing Techniques
      1. **What were the software testing techniques that you employed in this project?**   
         Testing techniques that I’ve been using for the milestones are Unit testing and Regression Testing. For Unit testing I’ve been testing the whole class by testing each individual method available, this allows me to ensure that any logic implemented within that class is working as expected.

Since our assignments are very similar, I’ve used the first assignment as a template for all current assignments. I change the package, class, and method names but the logic is very close so once I make the appropriate changes I regression test to assure that the program compiles and tests properly.

* + 1. **What are the other software testing techniques that you did not use for this project?**   
       Performance and Security Testing are additional testing techniques that I have not used for these milestones. I could set up a loop to test the Performance of getter or setter methods but there’s no need or demand for them. Depending on the needs of the client or system expectations if my test were to fail then I would have to adjust the code in the classes to ensure better performance. In my mind Security means access to different methods or information if I use this as the Definition for Security then there have been no tests or security matrix developed to test. If Security means validations such as character limits or null checks then some security testing has been done but not completely.
    2. **For each of the techniques you discussed, explain the practical uses and implications for different software development projects and situations.**  
       Unit Testing allows developers to ensure that different components are working as expected and allows them to have quick and easy preset tests to evaluate their system. It also allows Regression testing to receive attention when test cases fail or don’t initiate. Performance testing allows developers to inefficiencies in code and challenges them to develop better techniques. Security testing ensures that the correct users can access and manipulate information when they’re allowed to do so.
  1. Mindset
     1. **Assess the mindset that you adopted working on this project. In acting as a software tester, to what extent did you employ caution? Why was it important to appreciate the complexity and interrelationships of the code you were testing?**When acting as a software tester I employed caution when setting up test cases. I made sure to set up test cases that covered all the requirements listed. I implemented Happy and UnHappy Path scenarios to ensure everything worked as expected. It’s important to consider the complexity and interrelationship of the code because the system is a compilation of parts. We must test the parts individually and then again as a whole to ensure that the parts work well together to deliver a system that is bulletproof. That is why I set up test cases for the Class Methods and the Service Class Methods to ensure that all parts are working as expected.
     2. **Assess the ways you tried to limit bias in your review of the code. On the software developer side, can you imagine that bias would be a concern if you were responsible for testing your own code?**   
        I tried to limit bias in my review of the code by trying to imagine myself as any other tester. I tried to look at each requirement in a different way than a developer, I always asked myself the following questions. How can I break this? Does this make sense? What is the intent of this code? As a software developer, asking these questions allows me to push my development standards. I can see how easy it would be as a developer to set up “Happy Path” scenarios that align with the way I set my code up. With an external tester, a person who is unfamiliar with the code can run tests that push the boundaries and challenge the code implemented.
     3. **Finally, evaluate the importance of being disciplined in your commitment to quality as a software engineering professional. Why is it important not to cut corners when it comes to writing or testing code? How do you plan to avoid technical debt as a practitioner in the field?**   
        It’s incredibly important to be disciplined when setting up test cases. If any corners are cut there could be serious holes in requirements, system expectations will not be met. Thorough test coverage is mandatory as it can prevent technical debt which includes delayed product release and increase development costs and life expectancy. In some cases, depending on the software, sloppy testing could cost lives as well.