7-2 Project Two Submission

Dhiren Gurung

dhiren.gurung@snhu.edu

Southern New Hampshire University

**Summary**

There are the Contact Service, Task service, and Appointment Service. In the contact service object should have unique ID, firstName, lastName and phone number not exceeding 10 character and the address should not be more than 30 characters. In the Task service, the unique ID required and should not be more than 10 characters. Name should be less than 20 characters, and description should be less than 50 characters. For Appointment service, there is required unique appointment ID which needs less that 10 characters and required appointment date field and cannot be the past date, the description needs no longer than 50 characters.

Based on the software requirements, my approach meets the requirements because the testing is successful based on the method that are created to ensure only the require input such as unique id, name, address, description, addressing the limited characters. Based on the coverage and testing all the class and method there are 100% therefore I assume that the effectiveness of the JUnit test and quality meets the requirements. Testing in JUnit tests, this was my first experience to test the code in JUnit and it is interesting and had chance to learn many ideas. As my first experience in JUnit test, I did not have any knowledge before, but this testing provided me information and finally to check my code that actually meets the requirements or not. To ensure the code that is technically sound when do the require task such as add, delete, update etc. for example the following code update the information:

@Test

void taskUpdate() {

Task three = new Task ("14444","Amber Bdr Gurung","this is the new description for task thre");

assertEquals(false,taskService.taskUpdate("155555","Ram Bdr Lungeli","Description updated"));

}

How did you ensure that your code was efficient? Cite specific lines of code from your tests to illustrate.

To ensure my code is efficient is based on the user requirements meets such as ensuring the unique contact id with not exceedingly more than 10 charters. Following code ensure the requirements:

public Contact (String contactID, String firstName, String lastName, String phone, String address){  
 if (contactID.length() <=10 && contactID != null){  
 this.contactID = contactID;

**Reflection**

In this software testing the techniques that I used is white box testing and it examine the internal code to check individual method to ensure that the test is successful or not using JUnit testing. The feature we can use as whole class test and individual method do the test. I can directly create the testing doing right click on class that I needed to do test and create test inserting required members into the test terminal. Other testing techniques that are not using in this project is black box testing. For the black box testing only covers the comprehensive testing that the application functions or not.

For the white box testing we test the application’s coding, design, structure etc. There is input requirements and process the input by analyzing and output the reports. In this testing the techniques would be statement coverage, branch coverage, condition coverage etc. For the black box testing there are functional testing, nonfunctional testing, and regression testing.

While working on this project the mindset that I had to adopt as a software tester to test the code and see the output report which meets the requirements and employ the caution. To employ the caution is important to check every testing and see the that the code actually meets the requirements or not. An example that should be caution in the requirement such as the unique ID that every entry should be different ID characters and that should not be no longer than 10 characters. The description that should be no more than 50 characters. If there the responsibilities for the developer to test their own code, then there is less time consuming because I can find out the where is the error based on the testing report and resolve the quickly. Now a days the testing automation gives developers to test the code themselves. It is very important to be disciplined to commitment to the quality of the software. It is important not to cut corners in testing because one error in code while building software the consequences would be disaster and may loss the huge damage in every institution. For example, software error can stop all the system in hospital which cause many patients in danger for their health to be recover.