**Unit Testing Approach and Writing JUnit Tests**

Jackson Peterson

CS-320-H7020

Professor Mike Prasad

While completing the assignment, I made a concerted effort to ensure that my work aligned closely with the specified software requirements. To illustrate, the task involved creating a Contact and ContactService class, which proved to be a straightforward process. However, when it came to testing the Task, I found it to be among the simplest tests to compose. The primary objective was to verify that the class fulfilled the stipulated requirements.

In retrospect, I recognize that the overall efficacy of my JUnit tests for the Contact and Task services was somewhat lacking. I believe there is room for improvement in my test-writing skills, particularly in areas such as testing updates, additions, and deletions. Moving forward, I aim to enhance my testing proficiency through further research on techniques that can contribute to the refinement of my assignments.

When conducting the tests, I ensured that they were executed in accordance with the specified requirements. This proved to be somewhat challenging, given my relative unfamiliarity with running unit tests. The testing process was aligned with the functions I had developed. The provided code snippet illustrates how I constructed the test for the "addTask" function.

For developing the code to make sure it was efficient, I tested the lines of code accordingly. The

code above is for the TaskClass I created. The overall goal is to ensure that the inputs should be

able to be valid to help with check to see if the unit test runs.

For developing the code to make sure it was efficient, I tested the lines of code accordingly. The

code above is for the TaskClass I created. The overall goal is to ensure that the inputs should be

able to be valid to help with check to see if the unit test runs.

For developing the code to make sure it was efficient, I tested the lines of code accordingly. The

code above is for the TaskClass I created. The overall goal is to ensure that the inputs should be

able to be valid to help with check to see if the unit test runs.

In ensuring the efficiency of the code development, I conducted tests on the lines of code in a systematic manner. The provided code snippet pertains to the TaskClass I formulated. The primary objective is to guarantee that the inputs are valid, contributing to the verification of successful unit test execution.

void testaddTask()

{

TaskService taskServiceClass = new TaskService();

TaskClass task = new TaskClass("123444","John Amble", "Description");

assertTrue(taskServiceClass.addTask(task));

task = new TaskClass("123444","John Amble", "Description");

assertFalse(taskServiceClass.addTask(task));

void testaddTask()

{

TaskService taskServiceClass = new TaskService();

TaskClass task = new TaskClass("123444","John Amble", "Description");

assertTrue(taskServiceClass.addTask(task));

task = new TaskClass("123444","John Amble", "Description");

assertFalse(taskServiceClass.addTask(task))