Module 4

In module 3, I was using the JUNIT method to verify the requirement for Contact, Contact Service, and it would be the same for Task and Task Service in module 4. To adhere with the software requirements, every conditions from Contact and Contact Service must be covered by JUNIT testing method including the limit characters and null prerequisite for every String variable in OOP from contactID, firstName, lastName, phoneNumber to the address and every functions surround the Contact Service such as addContact, deleteContact, and update String for every String variable, except for contactID since it needs to be uniqued.

Furthermore, the method for to verify the requirement would be the same for Task and Task Service. I believe the test percentage of coverage for all of the requirement is over 90% since it checks for errors if the condition is not met. For example, String taskName of Task object cannot be null and character should not be longer than 20 chars. After we set up the constructor, and do the getter, setter for the String taskName, we can set the condition for taskName like below:

public void setTaskName(String taskName) {

if (taskName == null) {

throw new IllegalArgumentException("Task name cannot be null");

} else if (taskName.length() > 20) {

throw new IllegalArgumentException("Task name cannot be longer than 20 chars");

}

this.taskName = taskName;

With this method, we can do the JUNIT testing to verify the requirement, JUNIT will check if the taskName is longer than 20 chars or null, then throw the illegal argument exception.

*@Test*

*@DisplayName*("Task Name is longer than 20 chars will throw errors")

void testTaskName\_case1() {

Assertions.*assertThrows*(IllegalArgumentException.class, () -> {

new Task("TaskID1234", "Task Name More Than 20 Chars","Task Description");

});

}

*@Test*

*@DisplayName*("Task Name is null will throw errors")

void testTaskName\_case2() {

Assertions.*assertThrows*(IllegalArgumentException.class, () -> {

new Task("TaskID1234", null,"Task Description");

});

}