Brooke Slampak

CS-320

**Module Four Journal**

For both the ContactService and TaskService milestones, I made sure my unit tests matched the project requirements as closely as possible. Every feature, such as adding, updating, and deleting contacts or tasks, had its own test to make sure it worked the way it was supposed to. For example, in the ContactService tests, I updated each field one at a time (like first name, last name, etc.) and checked to see if the change happened. I followed the same approach for the TaskService, testing task creation, updates, and deletion using the task ID.

I believe the quality of my tests was solid because they covered several test cases. In the contact and task tests, I didn’t just check the normal inputs, I also tested what would happen with invalid or missing values. For example, I tested names that were too long or fields set to null and made sure the program either fixed the issue or threw the right error. Using assertThrows helped confirm that the code was catching problems like it should.

To make sure my code was technically sound, I used clear and specific assertions in each test. For example, after updating a contact’s first name, I used assertEquals("Jane", service.getContact(id).getFirstName()) to make sure the change happened. I did the same thing in the TaskService tests. After updating the description, I checked that the new text was stored correctly. I also made sure to test situations where the input should fail, so I could confirm that the code didn’t let bad input slip through.

As far as efficiency goes, I kept my tests simple and to-the-point. I reused service objects within each test method so I didn’t waste time creating new ones. In the TaskService tests, I added a few tasks and deleted one to check that the deletion didn’t affect the others. I also used @AfterEach to clear out any saved tasks after each test, so the tests wouldn’t interfere with each other. Overall, I tried to write tests that were easy to understand, implement, and run while still being effective.