Jared Bickler

CS 320

2/23/2025

**Summary and Reflections Report**

**Unit Testing Approach**

Requirements were outlined before development began. An example of a requirement was from contactObject, where the object was directed to have a required firstName string field that cannot exceed ten characters. Each test ran through JUnit, giving a completion, an error, or a failure. Each test covers the requirements by adjusting and modifying information as necessary.

**Experience Writing JUnit Tests**

Simple and clean syntax, with clear formatting and supporting comments are critical to creating technically sound code. Comments are particularly important, allowing the developer to explain in plain language the intentions and functionality of particular classes, constructors, variables, and logical operations. As an example, the following code is meant to update a task name using the taskID. They are kept in an array and auto-assigned an ID to prevent creating duplicates. This code gets the requested task with its ID, and sets the name and updates the string.

public void updateTaskName(String updatedString, String taskID) {

for (int counter = 0; counter < taskList.size(); counter++) {

if (taskList.get(counter).getTaskID().equals(taskID)) {

taskList.get(counter).setTaskName(updatedString);

break;

}

if (counter == taskList.size() - 1) {

System.\*\*out\*\*.println("Task ID: " + taskID + " not found.");

}

}

}

**Reflection**

**Testing Techniques**

Individual objects are tested to ensure that the code functions as intended. Results are displayed in the console, and shown in JUnit as a pass, error, or fail. Some testing techniques were not used in milestones. One is Decision Table testing, which is used to define functions and the conditions that they will operate in. Another example is State Transition testing, which has a tester test the behavior of an Application Under Test (AUT).

Many of the milestones used Equivalence Partition techniques to test maximum character limits within IDs, names, and strings. These are used in situations where no character limit could be an issue in the software, such as a username or social media post. Boundary Value analysis tests values between two extrema. This, combined with Equivalence Partition is used in password creation to set parameters for minimum and maximum characters, and constraints on character types.

**Mindset**

Most time spent developing software is used in validating and debugging. A mindset of problem-solving is important to the developer. Most testing is about finding and solving problems, and working to mitigate them. Testing should be approached with efficiency in mind, much like coding itself.

Testing one’s own code presents the potential for bias towards its effectiveness. It is important to approach code with the same level of constructive criticism as one would approach someone else’s code with. This can prevent cutting corners or being too lenient on the effectiveness of the final product.