**Project Two**

Eric Singleterry

Southern New Hampshire University

CS – 320 Software Test Automation & QA

April 14, 2022

My approach for testing these three features was primarily making sure the project accepted the three source files with their own individual packages. The first source file contained the packaged named “contactService” which held the first portion of the java files which are “Contact” and “ContactService”. The Contact.java file contained the variables which pertained to the unique Personal Identifiable Information (PII) of the customer. These variables include the customer’s first name, last name, phone number, and address. These variables also contained unique conditions such as string length; for example, the names and phone numbers cannot exceed 10 characters, but the address had a limit of 30. The second package in the first source file also contained the JUnit test files called “ContactTest” and ContactServiceTest. These files were meant to test the conditions set to the variable’s character length such as, creating a new contact with their Contact ID and PII and having their character length limit met.

The second source file contained the package named ‘taskService” which held the variables for what the task was going to be about. These variables were “taskID’, “description”, and “name”. These variables also contained their own conditions such as the ID not being able to exceed 10 characters, the name cannot exceed 20 characters and the description cannot exceed 50 characters. This source file also contained a second test package which had “TaskTest” and “TaskServiceTest” as the JUnit test files to ensure the variables met their conditions.

The third and final source file contanted the “AppointService” package which held the java files named “Appointment” and “AppointmentService”. This package was to ensure the customer could have a appointment scheduled at a specific date. The variables for this packaged were “appointmentID”, “date”, and “description”. These variables also had conditions such as: appointment ID cannot exceed 10 characters, date cannot be a past date from the current date and the description cannot exceed 50 characters. The date was a little tricky to configure, but I was able to do a quick google search and found an example from “javapoint.com” which I was able to implement into the source code. This third source file also contained a test package with the JUnit test cases of “AppointmentTest” and AppointmentServiceTest”. These tests were put in place to ensure the appointments were made with a date and description.

Overall this was a challenging task to figure out my low level of code creation with the addition of adding JUnit test code to coincide with the original code base. I double and triple checked that all variables were included in each java file and that each variable would have an effect @Test when it came down to the JUnit tests. I went with a “don’t fix something that isn’t broken approach” when it came to adding all the files into one project. I ensured that all source files contained every java file and JUnit file. What I would’ve done differently is change the names of the source files to its corresponding package file, i.e., src = contactService src, src2 = taskService src, and src3 = appointmentService src. My final thought on this project is that it could still use some work and enhancements on procedures. I do not agree with the low string conditions for the names. 10 characters for a first name might seem sufficient for most names, but last names are more complicated. Some people tend to have two last names combined with a hyphen, which can cause an issue when identifying a customer solely based off name.