1. **Summary**
   1. **Describe your unit testing approach for each of the three features.**
      1. **To what extent was your approach aligned to the software requirements? Support your claims with specific evidence.**

My testing approach was directly aligned to the software requirements. The reason for this is that as a developer, the requirement of the software I am planning on writing are one of my primary concerns. Throughout the development process I refer back to what exactly the software needs to do and how it is expected to behave. Once I move forward with writing my unit tests, I am generally at a comfortable spot in my development and believe that most of the code that I have written is doing what it should be doing. The tests that I write support the code that I have written in that, I am testing all the code for both positive and negative scenarios. In other words, I don’t want to see just how the chunk of code I am testing does in the best possible situation but also in the worst possible situation. If we take my project one as an example for my test on Appointment.java, my first test is a positive scenario based on the code that I had written to satisfy the parameters of the exercise. The first test is the positive scenario, essentially everything went fine. In my subsequent test I test failure states and the behavior of the system. Does it crash? Does it return and error, while these weren’t specifically laid out in the requirements, you don’t want your application to terminate unexpectedly.

* + 1. **Defend the overall quality of your JUnit tests. In other words, how do you know your JUnit tests were effective based on the coverage percentage?**

I know my Junit tests were effective not only based on the coverage percentage but by the tests themselves. My coverage is 100% on the different classes that I have written tests for in project 1. However, I have seen tests with high percentages of coverage that didn’t really test anything. A prefect example is in Jest you have the ability to check is a function has been called. I have seen tests that call that function and then test to see if it has been called and that is it. All the lines of the function are hit in tests like that so coverage looks good but it is not testing anything.

My tests were created to test the success scenarios as well as the failure scenarios for each test I wrote. With testing it’s not good to assume that everyone will use the code as it was written. It needs to be tested for multiple failure states to confirm it is resilient enough for production.

* 1. **Describe your experience writing the JUnit tests.**
     1. **How did you ensure that your code was technically sound? Cite specific lines of code from your tests to illustrate.**

I was able to confirm that the code that I have written is technically sound by testing it. If we take contact service for instance. In the contact service I had written the code blindly, meaning I didn’t test at all while I was writing it initially. Generally, I stay away from doing this because inevitably it catches up with you, and you have to reword a lot of things. With this project I had reasoned out that it was very similar to the previous service I had written and besides that it wasn’t a running application so I had no easy way to test other than to do unit tests. I was right and wrong. It was an relatively straight forward components however typos can still happen and after I had written my tests on this component I caught an issue. I was setting the address to the phone number variable. Luckily, because this wasn’t a complex component it was an easy fix but it also goes to show the value of having some tests run early in development to catch these things before, they can become more of an issue.

* + 1. **How did you ensure that your code was efficient? Cite specific lines of code from your tests to illustrate.**

Efficiency with programming doesn’t lend itself to testing for me. I envision efficiency as not duplicating code, or using classed appropriately. With Project 1, the way I confirmed this is manually at first. I went through the code base and checked for any code that was duplicated. If we look at any of the services, I submitted for program 1 they are not duplicated anywhere. I then checked for code smells. The only real code smell I wasn’t able to address is the set Month method in appointmentTest and appointmentServiceTest, which was depreciated. I didn’t address those as in the instructions it was indicated to use these methods.

Another note on efficiency, I could have implemented a helper sort of class to do some common sort of functionality that we are using. For instance, iterating and removing an entry from a list. However I feel it makes the code cleaner to have these sorts of methods internal for the class rather than in a helper class for a project this size.

1. **Reflection**
   1. **Testing Techniques**
      1. **What were the software testing techniques that you employed in this project? Describe their characteristics using specific details.**

I used unit testing and white box testing for project one. White box testing is essentially testing the internal structure of the application but there is a large overlap with unit testing here which tests individual components in the code. The only reason I am including white box testing is there is some communications happening from the model to the service. For instance, in Appointment service test, we are creating appointments pulled from the appointment model. This would be an example of white box testing as we are testing not only the components but also the overlap.

* + 1. **What are the other software testing techniques that you did not use for this project? Describe their characteristics using specific details.**
    2. **For each of the techniques you discussed, explain the practical uses and implications for different software development projects and situations.**

We did not implement any automation testing in the project. Automation testing is essentially automating a series of events that show normal use of a program. An example would be a scenarios where the user logs in, creates and appointment, changes the appointment name and then deleted the appointment. The benefit of automation is it doesn’t test just one aspect of the code base, its tests how everything is working as a whole. This can also be frustrating, particularly when the service you are testing pulls data from another service that is not working correctly. That scenario would cause all your tests to fail. Generally though, automation gives us a good idea of where problems in the regular flow of an application are.

We also didn’t utilize any black-box testing. Black box testing is testing without knowing the ins and out of the code, essentially end user testing. The benefit here is that, as developers, we tend to code for the scenarios we would take. However, end users don’t always follow the same path and it will often lead to unexpected results that we can the program for. We also call these edge cases.

* 1. **Mindset**
     1. **Assess the mindset that you adopted working on this project. In acting as a software tester, to what extent did you employ caution? Why was it important to appreciate the complexity and interrelationships of the code you were testing? Provide specific examples to illustrate your claims.**

I don’t think of caution and testing going hand in hand. In my experience you are really not looking to be cautious when you are testing, you are looking to break the code somewhere. The tests that you write as a tester are very important and can be used for a long time after you are not working on a particular feature anymore. With that having been said you want to test the capabilities of everything that you have written and if something breaks? Well that is an opportunity to make it more resilient. (In practice, sometimes its rough to work so long on something and have it broken with a few test cases, but it is an important step.) Going back to the interrelationship of the code base though, its is always important in testing to understand where different components are getting their data. Lets take a look at appointment and appointment service. If I were running the appointmentServiceTest and get an error on the addAppointmentSuccess() test that doesn’t necessarily mean I have an issue in the appointment service. The reason for this is the service uses the model class appointment.java. If there is an error in creating an appointment for some reason, the appointmentSuccess test would always fail and if you didn’t know where the data was coming from you would have no way of fixing it..

* + 1. **Assess the ways you tried to limit bias in your review of the code. On the software developer side, can you imagine that bias would be a concern if you were responsible for testing your own code? Provide specific examples to illustrate your claims.**

I tried to limit bias by testing for all the failure options. This is important as your code should be resilient and the test should seek to test this resiliency. An example would be in the contact service test. For add contact we have two tests one is a success test and the other is a failure test. This is an attempt to limit bias as I know at some point that function will fail, by testing a failure scenario I am trying to anticipate what could happen if my “happy path scenario” isn’t followed.

* + 1. **Finally, evaluate the importance of being disciplined in your commitment to quality as a software engineering professional. Why is it important not to cut corners when it comes to writing or testing code? How do you plan to avoid technical debt as a practitioner in the field? Provide specific examples to illustrate your claims.**

It is of paramount importance to be disciplined when it comes to quality. This isn’t just for you as the developer now, but also all of the other developers who will touch this code in the future or the ones who will have to rebuild because of an oversite. To start with code quality, one of the primary reasons it is important is functionality, if you have a monolithic function that you have created and have to cut and paste that monstrosity across different classes there is a problem there (I have run into this). This is bad practice for several reasons. First, giant blocks of code like this generally do multiple things in one function that can be easily understandable and sustained if they are broken int smaller parts. And second if you have to change a piece of a function like that you have to search around to find where all instance of that occur. Its just bad practice, its sloppy and lazy. As a developer you want your code to be easily understandable so it can be adjusted when it inevitably will need to be and you want it to be reusable. Rather than have a copy pasted function just created a helper class, write the function once and use it however many times you need to. That way if a change needs to happen, you have only one area to change. Tech debt happens, that is the nature of our industry. A good example of this would be a little over a year ago we transitioned from selenium only e2e testing to protractor, a few months after we finished we got news the protractor would not longer be supported so we started looking for other options. So tech debt will happen regardless, but if you have a well maintained ordered code base you can mitigate a large portion of the downtime that tech debt carries.