The following is a summary and reflection for the mobile application that we are working on with Grand Strand Systems. In this it was our job to code the application and provide unit tests to ensure that the code met the specific client requirements. It was my job to provide contact, task, and appointment services. The purpose of the application was to allow services to add, update and delete contact information, tasks, and appointment objects. We broke this down originally into steps. We worked each section and then at the end came through and put everything together. So, one week we worked the contact class, the next we did tasks, and then we finally worked out the appointment services. In this briefing we will discuss a summary of the unit testing and experiences while writing it. Then we will reflect on our approach to programming and coding, as well as our commitment to the quality of the code as a software engineering professional.

Let’s talk about the testing approach from a technical standpoint. We will take a look at each feature individually and explain what we did for testing. Now for each aspect or service, we had both the class that set everything up and then we had a service class for all of them, which is where we worked the functionality. First, we have the Contact, which in the coding we labeled Contact Class. Here we had to ensure that contact information for each patient. We needed to have a contact Id, first name, last name, phone number and home address. Then there were requirements for length. The contact Id, first name, last name, and phone number could not be longer than 10 characters or there would be errors. Then the address could not be more than 30 characters. On top of this, we have to ensure that none of these needed items could be null. For the contact class itself for the testing, we went through and essentially made fake information, and then had the system plug in different variables for each to ensure that proper guidelines were met. We tested things that met the requirements and then also tested things that did not or that were too long for the parameters. In the contact class and the contact service we used the Before Each, Test and Assertions under the java unit to test the code and ensure it functioned properly. We first set up some generated some initial information for each piece of data we needed to make sure was in the system. Then we would use assert equals and assert not null to make sure that things were set properly and in fact not empty. For example, in the Contact Service we had a test for updating the first name, and we walked through what would be allowed and put in some throws to ensure things that were not allowed were thrown out. And only items or names that meet the parameters would be allowed. This was the same in the contact class we used the same testing methods and one by one added features or information into the contact information to get everything needed. We built constructors as we went along until we had everything we needed in there, and we used asset equals to make sure that it would set up what we want it to, so that it can move to the contact service, and not have any issues. Next we went over tasks, this is where we got descriptions of what was needing to be done in the appointment. Ow just like contact this had to have the unique Id, and then have a name which would be first and last, and then we would have the actual description. Again, none of these could be empty and they each had their own requirement for how long it was allowed to be. The id was 10 characters, the name was 20 characters, and the description was 50 characters. While going through with the task itself we used that class to set up parameters and how things could be set, and then we used task service, to go through the steps of finding, adding deleting and so forth. Assertions, Before Each, and test were used for this one as well. This way I could test step by step by simply adding one piece to the next until I had the full information. It also allowed me to put in some options that were too long and didn’t meet parameters so I could ensure the program would throw them out or give the user an error if they went outside the set parameters. Finally, we approached the appointment and appointment service in a similar fashion. We used the Before Each, Assertion, and test features through java unit testing, so we could build up what we needed to and ensure the program wouldn’t accept any empty information or any information that didn’t meet the length requirements that the customer provided to us. You can see this in the appointment service test for example where we test for ne appointment and go through everything line by line with the assert not null and assert not equals, and the assert throws. These three test to ensure we have proper information for the appointments and the descriptions. Then if not it would simply give the user an error or throw it out completely. All together the three components the contact class, the tasks, and the appointment services, would all work together for the user to be able to do what they need to add, delete, modify things in the system.

This project was no small feat, which is why looking back at it, I am appreciative that it was broken up into smaller pieces. This allowed me to work through the specifics for each product and get feedback before putting it all together. The feedback was very helpful, and I was able to implement it into the final product that was submitted previously. When I went through initially, I wasn’t able to get everything in the requirements that the customer was needed, and that was pointed out to me, so I then took it and reworked everything to ensure that it was there the way it needed to be. More specifically I hadn’t put in the requirements for the lengths of things and was simply testing without them, when I went back from the client feedback, I implemented the lengths and got them to where they needed to be. One reason I was appreciative of breaking this up into smaller pieces is getting the second or third pair of eyes on the code, and being able to see what I might have missed going through it the first time. This also helps eliminate any bias I might have on the code itself. For example, I thought I had everything I needed when I was looking at the contact class the first time through, and then when I presented it to the client, I was provided with the feedback of the things I was missing. I think in general when you are doing something you will take one of two approaches with it. Either you will think everything is good and it is going to work, or you will think that everything is horrible and will need a lot of reworks. There really isn’t a middle ground in it. Then when you test it, you get told usually the opposite, so I think it is always nice to have someone else take a look at the code and help you through it, either before or after testing just to make sure everything is where it needs to be and is functioning the way it needs to. Now with this and any project you take on, you have to be sure you are giving it your full commitment. I say this because just like in life it is easy to start doing lots of projects all at once and work on each a little here or a little there, but some of these projects will hve deadlines that need to be met, as well as security concerns that you will need to ensure are where they need to be. This usually works best when you are focused on one project and that has your full attention. This saves you from cutting corners on the code in order to just get it to the client. This may get them what they want faster but can lead to some consequences down the line. There could be a potential defect or bug that you didn’t catch because you were just working on getting it out. Now in life these things do happen for instance the Bangladesh Bank heist that happened through the system in 2016, there was something in the system that allowed things to be changed and overridden which allowed the robbers to get in. (Online, 2019) I feel that if things had been thoroughly tested, they could have saved millions of dollars from the start. The same is with any project, with this one we are dealing with personal information. So, making sure we are 100% focused on it will make everyone happy in the end. It will help alleviate those potential headaches of security breaches, by setting up for things to be edited in the program but not affect the code itself.

This project was a big project, but I am glad that I was able to work on it. It really did help me with my skills as a programmer, especially when it came to testing. Testing is not something I have had a lot of experience with, and this really opened my eyes to the importance of it in all stages of the SDLC. I look forward to putting these skills into play on future projects.

References

Online, T. S. (2019, July 29). *Software, printer problems delayed discovery of Bangladesh heist*. The Star. Retrieved December 5, 2022, from https://www.thestar.com.my/tech/tech-news/2016/03/17/software-printer-problems-delayed-discovery-of-bangladesh-heist/