James Ehrle

CS320

Project 2

Oct 14, 2025

My approach was well aligned with the software requirements in project one application, I used the requirements as a guide to complete each step and add each new item. There were many requirements for each item that was needed. Each class had its own requirements, Contact Class had the most items, and it included the contact ID, firstName, lastName, phone and address. Task Class had taskId, name, and description and the appointments Class had appointmentID, name, and description. Each of these items had specific requirements. Each item had specific character limits and couldn’t be null. Below is an example of how I achieved this with each item, I made condition that if it was equal to null or greater than the specified amount it would throw the IllegalArgumentException.A computer screen shot of a code

Description automatically generated

Another requirement that I ensured I met was that the ID couldn’t be changed so I made each ID a private string that once created it wouldn’t be able to be changed. The way that I tested for each condition was by creating a new instance of the item and testing it against each requirement. An example of this is below. A computer screen shot of a program

Description automatically generated

My overall quality of my Junit testing was very effective for all the classes, I know that it was effective because each of my coverage percentages are all close to 100% coverage. This indicates that each method of my Contact, Task, and Appointment classes are all function the way that they should and tested for the items that need to be tested. 







Even with the coverage percentage being higher I still went back and made sure that there wasn’t anything that I was missing after running the coverage on the specific class. This let me know that each method was tested correctly.

The way that I ensured my code was technically sound was by making functions that will be used though out such as the get method in the services to be able to loop though the list and get the specific ID. This also helps with repetitive code, that way I’m just calling the method instead of coding the loop though each time. I also used the Junit test to make sure that each item would pass based on the criteria’s set. This is particularly helpful when there isn’t an input that can be tested early on. That way I don’t have to go back and fix anything when the input starts to be used. The example of the get method is below. A screenshot of a computer program

Description automatically generated

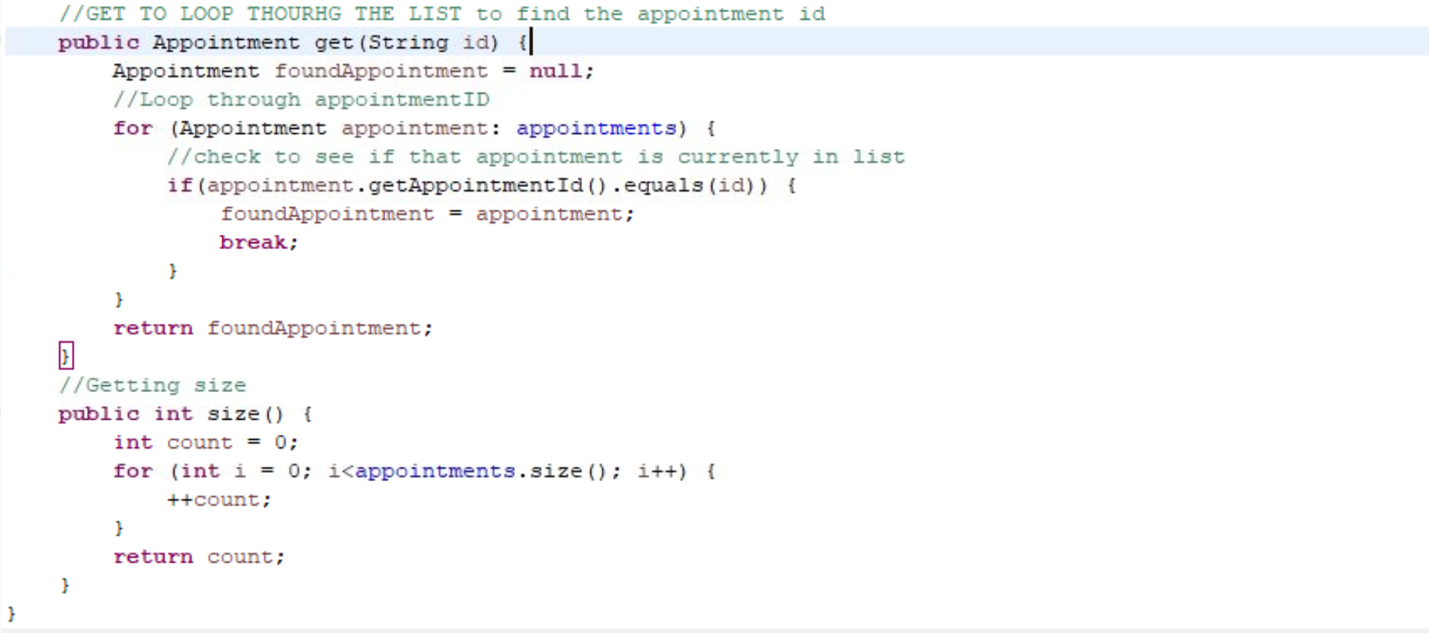
The way that I ensured my code was efficient was by adding comments to indicate what was going on with each method, that way if anyone had to look at my code it wouldn’t take them more time to try to figure it out. Another way that I made my code efficient was by separating the different conditions in each item. I did this by using IF and ELSE if instead of using and IF with and || (or) to separate each item. This helps because if the user inputs an item and the description is NULL they know exactly what is going on instead of it just saying that it is invalid it will tell them that the description cannot be NULL. A screenshot of a computer code

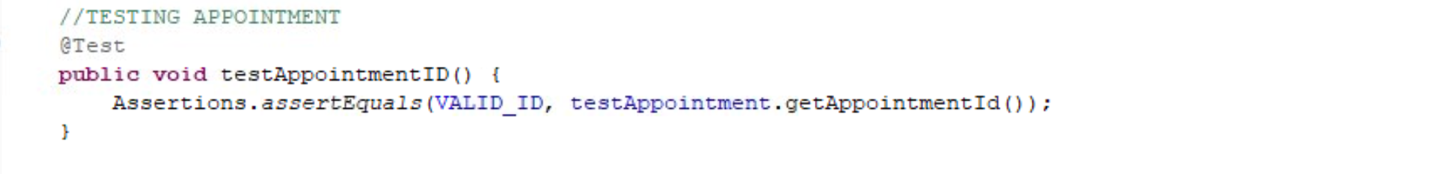
Description automatically generated

One of the techniques I used was static testing, which is testing the software without executing any code. I did this before the project and each milestone to make sure that I have everything I need before I start my testing. The main technique that I use for static testing is inspection, by doing this I inspect the requirements and inspect the structure that I am using to make sure that it will be the best possible design. Dynamic testing is another technique that I use to test the behavior of the mobile application by using Junit tests. Within dynamic testing I used black box testing which is testing cases directly from the specifications. As I implemented more code it, I tested it each new feature. This was really used as I developed the add, delete, and edit functions. Within dynamic testing I used unit testing this is the process of testing the individual software components such as each feature in the different milestones. This would include the inputs, the add, delete, edit. Another one that I used was integration testing, this was when I implemented the ContactService part of the mod 3 module. I made sure that it worked well with the contact class to be able to create the new case for the contact.

Some of the other software testing techniques that I didn’t use with in Dynamic testing were system testing, Acceptance testing, performance testing, and security testing. System testing is testing the whole system and making sure the entire software system meets the specified requirements and works as intended. Acceptance testing, which is the final stage of dynamic testing, and it makes sure the software meets the needs of the end-user. Performance testing is a type of dynamic testing that is focused on evaluating the performance of the software under workloads. Security testing is testing security risks within the security system.

System testing is great to use when the system units are all finished developing, this tests the whole system to ensure that it meets the specific requirements, this tests the software function, performance, and usability. Acceptance testing is the final stages and is good after the system has been tested, during this test it makes sure the needs are met by the end-user. This is testing the software function and usability from the end user’s perspective. The performance testing is great to use after the system has been tested for the usability, this is going to test the system under different workloads to ensure that it behaves how it is supposed to with things like large number of users and different inputs. Security testing is good to use to evaluate the vulnerabilities within the system to ensure that the software is secure. This step will ensure that the system won’t be vulnerable to attacks.

The mindset that I had to adapt while working on this project was patients. During this project I employed patients because testing the software though the Junit test can be more coding and more time, but it helped the outcome of the overall project tremendously. There was caution employed with every new method that I created, I didn’t want to get to far along and not test because I would have to go back and change multiple things. It is important to appreciate the complexity and interrelationships with the code because the more that you appreciate them the more attention that will be placed on them this will help ensure that both the methods and the tests are all coded correctly. The complexity of the date method gave me some issues at first but after I learned it, it made sense. The interrelationships including the different methods such as the get and size function help to keep the repetitiveness down and easier to read the code. 

The way that I tried to limit bias from the code review was by testing everything I could and not just assuming that I coded it correctly. An example of this is testing the actual item with each method, such as the appointmentID, I tested for the requirements and made sure it couldn’t be updated but I also tested to make sure that it was functioning correctly overall. 

By testing it with the VAILID\_ID and making sure that it was passing along the correct item to getAppointmentID. It is difficult to eliminate all bias during the testing side since I created the code but being able to plays a big role in the success of the project. The importance of being disciplined in the commitment to quality as a software engineer is incredibly high. Depending on the project it could cost a lot of money and possible lives if there isn’t a commitment to quality. Cutting corners in code could cause failures in systems, that is important not to have in any systems but some systems it could be more important. I plan on avoiding technical debt as a practitioner in the field by testing as much as my code as I can though out development. As I was developing this application, I would code a function then go to my test and code and test it to make sure it was working correctly. But within a system you must make sure the individual compensates work by their self as well as making sure all those individual items work together in the system as a whole. I plan on testing every aspect of the method including the function of it as well as the requirements. I think this is demonstrated well in the add function, I tested with one, multiple items and with duplicates.



References:

(2025, July 11). Dynamic Testing - Software Testing. Geeksforgeeks. https://www.geeksforgeeks.org/software-engineering/software-testing-dynamic-testing/