Shanine Efferson

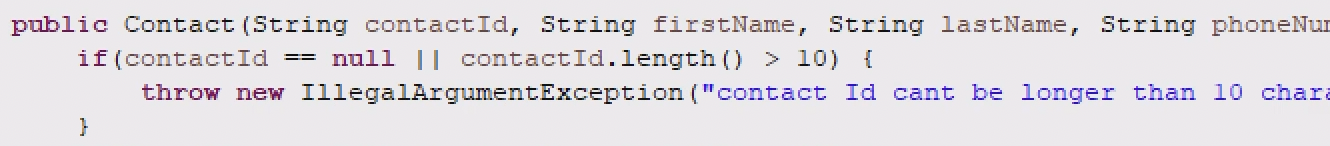
SNHU

CS 320

Project Two

While developing the mobile application for the customer, Grand Strand Systems, I was tasked with creating three features that would be used in the mobile application. The three features were contact, appointment and task services. These services allowed the mobile application to effectively manage contact information, appointment scheduling and organization. I will describe my approach when developing these three services and how I tested these services using Junit tests and other techniques to ensure efficiency in my code. I will also describe what I learned as a software developer ad how I adopted a mindset that helped me ensure accuracy, quality and efficiency of the code.

To make sure I aligned with software requirements I made sure to follow the requirements provided in each module. The contact management Service featured needs the ability to add, update and delete objects. The contact object required a contact ID, first and last name that is no longer than 10 characters. A phone string no longer than 10 digits and an address field no longer than 30 characters. To make sure I aligned with the requirements I used testing to check for valid and invalid inputs.



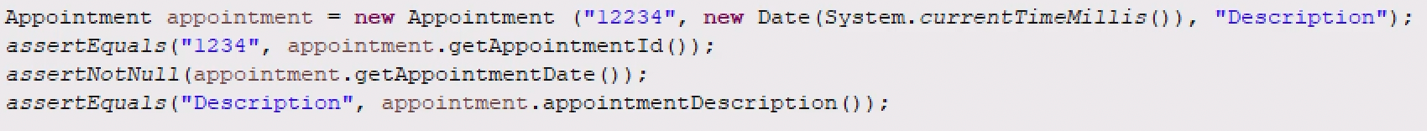
If the input is invalid, in this case longer than 10 digits, an exception will be thrown. To know that my Junit testing was valid the code coverage was above 90%.

To ensure the code was technically sound I used assertions to test the outputs. In the appointment service test class, I tested the expected inputs and tested if the input is null. This also allowed efficiency by making sure the code is well written and readable.

A close up of a text

Description automatically generated

The main software techniques that I employed to test this application was unit testing. Unit testing tests the methods and classes of the code to make sure it follows requirements. For the three services, contact, appointment and task services, I created classes as well as test classes to make sure each service followed the requirements outlined by the customer. In the appointment class I had to make sure the date for the appointment were not scheduled in the past. I did this by creating a unit test to validate the user’s input. Unit testing is used to identify errors early in the code, this saves time and money and allows changes to be made easily in the code. Below is an insert from the Appointment test class that uses unit testing to validate user inputs.



Some of the software technique I did not use for the application is system testing and integration testing. System testing involves testing the entire application to make sure all of the components are working as expected. Integrated testing is when the code is tested based on the interaction with a database. Since the project did not include any database to import information from, integration testing was not necessary for this assignment. System testing was also not needed because each service was tested and created individually and did not include testing all services together.

The mindset I adaptive throughout the testing phases and development of the services, was to be thorough in my testing and have attention to detail. As a software developer I had to account for various situations and make sure I tested properly or these different scenarios. I had to have a very focused mind and keep the users in mind.

To limit biases in my code I made sure to take the feedback provided by my professor and peers and implement that into the future modules. As developer I had to be open to feedback and understand that my code was not perfect and could perform better if something was changed or implemented. It is important to not cut corners as a software professional because it will create issues later in the code and make these issues harder to fix.