Brooke Slampak

CS 320

**Module Five Journal**

For each milestone in this project, I used unit testing with JUnit to make sure each part of the program worked the way it was supposed to. Unit testing is done by focusing on testing small, individual pieces of code like methods or classes. I created tests that checked both valid and invalid inputs, like making sure names weren’t too long, IDs weren’t null, and dates weren’t in the past. I also tested to see if exceptions were thrown when they should be. This helped me ensure sure each class worked correctly individually.

There are other types of testing I didn’t use for these milestones, like integration testing, system testing, and acceptance testing. Integration testing tests how different parts of the program work together, such as making sure the appointment service and contact service can interact without issue. System testing involves looking at the whole program to see if every component works as a complete system, and acceptance testing is usually done by users or clients to see if the program meets their needs. These types of tests are more about how everything works together rather than just checking individual methods.

Each kind of testing has its place, depending on the project. Unit tests are great for catching bugs early and making sure each small piece of the code does what it’s supposed to. Integration tests are useful when you have multiple components that need to communicate with each other. System and acceptance tests are important later on in development, especially when you’re getting close to delivering a final product. For these milestones, unit testing was the most practical choice since we were building and testing small, individual services without a user interface or database.