The Grand Strand System Application sets up an API infrastructure for a back-end application. The application would allow admins to create appointments, tasks and users within a system. This app was developed in the CS – 320 Software Testing course at SNHU and its main purpose was to develop good unit tests for codes.

This app was selected for the portfolio because it is a great indicator of my understanding of data structures and algorithms. The original implementation used ArrayLists in each service to store the instance of objects being created by the system. And during any CRUD operations within the code, loops were being used to iterate through the Lists to get to the desired object based on its id, which was passed as parameters in methods. This approach caused the time complexity of the algorithm to be linear, O(n), as the runtime of the code would increase with the size of the list. This would cause a system wide lag in case of a huge data set, as every operation would require the loop to iterate over a huge data set to get to the desired object. The enhancement replaced the List with Maps, which stored the object and its ID as key, value pairs. This lets me refactor the code to remove the dependency of loops and use map specific methods to get to the desired object based on the id parameter in the method. The time complexity of the new algorithm is constant, O(1), which means the operation will take the same amount of time regardless of the size of data set.

Yes, all the desired outcomes of this enhancement were met due to the improved algorithm and industry standard test coverage:

* Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.
* Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.

My approach to enhancing the artifact was commenting out the existing code and using that as reference when implementing the new data structure. Updating the unit tests was a bit challenging as the new implementation broke a lot of tests. I had to introduce a few more unit tests as the new codes made the coverage fall below 80%. Working on this enhancement, I realized the importance of keeping time complexity in mind when developing code, and how vital it is to implement the right data structure for the task developers aim to accomplish via the code.

Directions to run the app locally:

Please ensure the project is built using Java version 1.8, there is no driver file, but the functionality can be tested by running the entire test suite. Please ensure the jar files within the project folders are being used within the project setup.