**Jeanelyn Roman**

**CS-499 Computer Science Capstone**

**Dr. Troy Hawk**

**Milestone Three Enhancement Two: Algorithms and Data Structure**

The artifact I chose for this milestone is my AppointmentService.java program from CS-320: Software Test Automation and QA. It was originally created during that course to build and test a small system that stores, retrieves, and deletes appointments. The code uses a HashMap to manage appointment data, and I focused on creating reliable logic for checking ID uniqueness, future appointment dates, and other basic input rules.

I included this artifact in my ePortfolio because it shows the logic and structure I’ve learned when working with data in Java. This assignment helped me understand how algorithms are used in real-world tasks like validating data and avoiding duplication. It also shows that I can work with basic data structures like HashMaps and apply conditions to make sure the system runs smoothly. I picked this one because it’s a simple project that really demonstrates how I think through logic and handling common problems.

The enhanced version improved the program by organizing the code better and making it easier to read. I renamed the method from addAppointment to createAppointment to make it clearer. I also moved the validation checks into their own method to avoid repeating code. I added JavaDoc comments to explain each method and what it does. These changes made the code cleaner, easier to understand, and more professional.

Yes, I believe I met the course outcome I planned for in Module One, which was to show my ability to solve problems using algorithms and clear structure. This enhancement also helped me build better habits for naming, commenting, and thinking through data handling. I don't have any major changes to my outcome plan right now.

During the enhancement, I learned how much a small change can improve readability and structure. I realized that separating validation into its own method made logic easier to follow. One challenge I had was figuring out how to improve the code without changing too much of its original purpose. I also had to carefully update the test file so it would point to the enhanced version and make sure the new method names matched. Overall, I feel more confident about breaking problems into smaller parts and making my code easier for others to read and maintain.