CS 499 Milestone Three Narrative

The artifact I chose for this enhancement is the AppointmentService component from my CS-320 Software Testing and Quality Assurance course project. The original version was a Java-based backend service that handled basic CRUD operations for appointment records using in-memory storage. This artifact was first created in early 2025, and for this enhancement, I focused on improving its data handling efficiency and retrieval accuracy by redesigning its algorithms and data structures. Since I am using the same artifact to enhance for all three of the project categories, the enhancements made for this milestone partially build on the enhancements made in the last milestone.

I chose this artifact for my ePortfolio because it demonstrates my ability to design and implement efficient data storage and retrieval mechanisms, which are important skills for any software engineer. This enhancement introduced a secondary data structure, a TreeMap<LocalDate, List<Appointment>>, alongside the existing HashMap<String, Appointment>. This change allows for deterministic ordering of appointments by date, allowing for efficient separation of “upcoming” and “previous” appointments. By integrating multiple data structures and optimizing retrieval algorithms, the artifact showcases my understanding of both hash-based and tree-based structures, as well as how to select the appropriate structure for a given use case. In addition to the new data structure, I implemented logic to ensure deterministic ordering in endpoint responses, refined validation rules, and added functionality to export appointment data in CSV and JSON formats. These improvements make the service more robust, scalable, and adaptable for real-world use cases.

In Module One, I planned to meet Outcome 3 and Outcome 4. With these enhancements, I achieved those outcomes and also met Outcome 1 and Outcome 2. For Outcome 3, I designed and implemented a dual-index approach that balances constant-time ID access with logarithmic-time sorted queries, which supports both high performance and flexibility. For Outcome 4, I applied industry-relevant tools and techniques by using Spring Boot to provide well-structured REST endpoints for retrieving all, upcoming, previous, and ranged appointments, as well as for exporting the data. For Outcome 1, the CSV and JSON export feature supports collaborative decision-making by allowing administrators, staff, and clients to work with the same structured data in formats they can use. For Outcome 2, I improved the clarity and professionalism of the project through clear code comments, informative error messages, and an expanded README that explains how to set up, use, and test the application.

This enhancement reinforced the importance of selecting the right data structures to improve both the functionality and performance of a system. Keeping the hash map and tree map synchronized required careful attention to update and delete logic to ensure data consistency. Implementing deterministic ordering for upcoming and previous appointment queries improved both the usability of the API and the predictability of results for the frontend. The addition of export functionality encouraged me to think about how the service could support collaboration beyond the UI, making the data more accessible for different audiences. Improving validation, error messages, and documentation also strengthened my ability to communicate technical behavior clearly, which is a critical skill for working on software projects that will be maintained or extended by others.