CS-499

Milestone Three Narrative

Logan Riedell

4/11/2025

**Briefly describe the artifact. What is it? When was it created?**

The artifact is an Appointment System application that I originally developed in Java for a previous semester. It was a straightforward console application that handled appointments, contacts, and tasks with storage that was in-memory and simple service classes. For this course, I rebuilt and improved the system in Python as part of my ePortfolio to better exhibit my knowledge of algorithms and data structures.

**Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?**

I chose this item since it addresses several facets of backend software development and let me utilize practical logic to my improvements. Appointment conflict detection and job prioritizing let me use algorithmic thinking and choose suitable data structures. Conflict detection was achieved by means of logic preventing overlapping appointments by means of checking for duplicate date/time inputs before permitting a new one. Using Python's heapq library, I built a priority queue for tasks such that the most pressing ones are always available first. These improvements show how I can use data structures and algorithmic thinking to tackle pragmatic issues.

**Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

I achieved the results I intended, particularly for Outcome 3, designing and assessing solutions employing algorithmic concepts, and Outcome 4, applying well-founded tools and methods in computing practices. By clearly recording and testing my code, I also advanced toward Outcome 2. Going forward, I intend to keep improving how I explain my thought process behind my ePortfolio decisions, thereby promoting Outcome 1 even more.

**Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

This improvement made me clearly see how to strike simplicity with performance. One of the key choices I had to make was whether to prioritize activities using a heap or utilize a dictionary for fast access to them. Ultimately, I employed both: the dictionary let me easily search for tasks by ID and the heap let me always get the most critical work first. Learning how to maintain those two structures in sync was beneficial. Though everything is presently saved in memory, I also had to consider how to create the appointment conflict logic such that it could handle more data later. Writing tests helped me find a few flaws early and gave me more confidence in the code.