4-2 Milestone Three: Enhancement Two: Algorithms and Data Structure

Jorgo Qendro

Southern New Hampshire University

CS-499: Computer Science Capstone

Professor Nembhard

June 13th, 2025

4-2 Milestone Three

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

For my Enhancement Two artifact, I selected a mobile Weight Tracker application I originally created in an earlier computer science course. The application allows users to input and monitor their weight over time, optionally set goals, and visualize their progress through a list of entries. The app was first developed as part of a course project that focused on building basic Android applications using SQLite for persistent storage. While the original version captured and displayed user data correctly, it lacked any advanced data analysis or efficient search functionality. I created this artifact during the fall semester of my academic program, and it represented one of my first opportunities to build a full-stack mobile solution.

**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 to include this artifact in my ePortfolio because it clearly illustrates my growth in implementing algorithms and using data structures in practical, user-facing applications. The artifact was enhanced to showcase these skills more deliberately through the addition of two significant features: a 7-day moving average calculation of user weight entries and a binary search algorithm to locate entries by date. The moving average feature demonstrates algorithmic thinking and list-based data manipulation, while the binary search algorithm applies efficient lookup logic to sorted data. These enhancements improved the artifact not only functionally but also in terms of software design and user value. The original application simply listed data entries, but now it provides meaningful feedback on trends and offers faster data access, aligning better with real-world user expectations.

**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?**

Yes, I met the course outcomes I had planned to address with this enhancement in Module One. Specifically, I demonstrated my ability to design and evaluate computing solutions that apply algorithmic principles and appropriate data structures to solve practical problems. The moving average algorithm calculates trends over time by using a sliding window technique over a list of historical weight entries, which required both accurate sorting and aggregation. The binary search enhancement involved iterating through a sorted list in logarithmic time, improving the efficiency of date lookups compared to a linear scan. These changes reflect competency in using both algorithmic design and thoughtful data structuring to improve application performance and capability. At this point, I do not need to update my outcome-coverage plans, as the enhancement fully aligns with my original goals.

**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?**

Reflecting on the process of enhancing and modifying the artifact, I found that the work reinforced my understanding of how foundational algorithmic concepts apply directly to end-user applications. One of the most valuable lessons I learned was how even relatively simple algorithmic additions, like a moving average or binary search, can significantly improve a product’s usefulness and responsiveness. I also deepened my appreciation for data preparation, particularly the need to ensure weight entries were correctly sorted before applying a search algorithm. Challenges during the enhancement included determining the best way to integrate the new logic into the existing MVVM (Model-View-ViewModel) architecture and ensuring LiveData updates triggered correctly in response to database changes. Ultimately, these difficulties led to a more modular and scalable solution, reinforcing my development skills and confidence in handling algorithm-focused features in production-level applications.