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

Kylee Middleton

Southern New Hampshire University

CS-499 12001: Computer Science Capstone

Jeff Phillips

February 1, 2026

The artifact that I chose for this milestone is my Android-based Inventory Management application that I had originally created for my CS- 360 Mobile Architecture and Programming class. This application allows users to create an account, log in securely, and manage a list of inventory items stored in a local SQLite database. Its core functionality is to allow users to add new items, edit item quantities, delete items, and view inventory details in a grid-based user interface. For Milestone Three of CS-499, I enhanced the artifact focusing on algorithms and data structure concepts and improved the overall functionality of the application.

I selected this artifact for my ePortfolio because it is a realistic end-to-end software solution that demonstrates practical problem solving in algorithms and data structures. Inventory management is something used in many different fields and environments and something that even I use in my own workplace. This personal experience gives me insight into what features are most helpful when considering algorithms and data structures for an inventory application. For example, I implemented a case-insensitive search algorithm that allows users to filter inventory items by either their name or location. Implementing this required managing in-memory data structures and applying string normalization to help ensure accurate and consistent results. I also added multiple sorting options using comparator-based algorithms. This feature allows the user to sort inventory items alphabetically (A-Z) or by quantity (High to Low or Low to High). This feature is helpful when users don’t remember the exact name of an item or where they are located or simply want a more efficient way to view inventory items that meet their personal preferences. The last thing that I added is a multi-tier duplicate detection algorithm for when users are adding new items to the inventory list. The system will distinguish between exact duplicates (same name and location) and partial duplicates (same name but in different locations). If a duplicate is detected, the user will be shown a pop-up warning that the item may already exist in the database and is also shown the existing item’s name and location. The user must then provide confirmation to add the item or can hit cancel to end the process and not add the item.

Overall, the enhancements that I have completed for this milestone align with my original proposal and through the addition of the search, sorting, and duplicate detection logic, I was able to demonstrate fundamental data structure concepts. My artifact also demonstrates progress in software engineering and database-related outcomes. The application uses a layered design that separates all user interface logic from database access and application logic, which helps to improve code organization, maintainability, and readability. At this time, I do not believe that I need to make any updates to my proposal or planned enhancements for this milestone or the project as a whole.

Ultimately, enhancing this artifact has been a valuable learning experience that has reinforced my previous learning and helped me to learn more about the importance of thoughtful algorithm design in user-facing applications. One of the main lessons that I learned was understanding how even small design decisions, such as when to filter data in memory versus querying the database again, can affect both performance and code readability. One of the main challenges that I faced was ensuring that the sort and search functions interacted correctly without producing unexpected or incorrect results. Managing multiple data lists, maintaining consistent state, and ensuring that UI updates are reflecting the current algorithmic logic required careful testing and iteration.

Overall, this milestone presented me with a challenge that helped to build both my skills and my confidence, and highlighted the importance of writing clean, well-documented code.