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

Bradley Wells

Southern New Hampshire University

CS499- Computer Science Capstone

Dr. Maciosek

January 27, 2025

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

I have chosen the same Android mobile application developed at SNHU in the CS360 class, which was centered around Mobile Architecture and Programming to make my enhancement in the area of Data Structures and Algorithms. The intent of the application was to track inventory items, with the app centered around having a main data table visible to the user and give the user the ability to perform CRUD (Create, Read, Update, Delete) functions within the application, altering the data in the database. An additional requirement of this application was to send an SMS notification to the user's phone number when any inventory item in the database reached a stock level of zero. This application was created in 2024 for the course and was progressively built with robust planning, User Interface design, and coding functionality into the application with Java.

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?

This artifact was selected because the project offered a complexity level suitable for showcasing in a portfolio but was still rudimentary enough to make worthwhile enhancements. It is also unique in the way that I will be able to perform all three areas of enhancement on a single artifact and continue to produce a better quality product throughout the duration of the course. This was also my first experience creating something full stack in nature, creating an application that connected the presentation layer to a backend database. While the artifact was fully functioning and met all the original requirements, there were some features lacking that would not only enhance user experience but also require the use of Data Structures and Algorithms.

The features I added were the ability to search the inventory items that appear in the user data table and the ability to sort the inventory items alphabetically in the data table. The search bar utilizes Hash Map because it can handle large data sets with fast performance with an O(1) average time complexity for searching, inserting, and deleting items. This means that regardless of the number of items stored, the time to find an item remains constant (Geeks for Geeks, 2024). This makes it an excellent choice for an inventory application, which could have large datasets. The sort function uses a Heap Sort algorithm to quickly alphabetize the item that exists in the database. Heap sort was selected due to its ability to scale to a large dataset with a time complexity of ((O)nlogn) in the best, worst, and average cases (Alake, 2024). These enhancements improve the application because it decreases the amount of time a user will spend finding an item in the inventory whether they use the search or sort function. These improvements would be critical features for an application like this that would be expected to deal with a large number of items in the inventory.

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 do believe I met the course outcomes I planned to meet in Module 1, which were outcomes 3, 4, and 5. By designing an efficient search solution using HashMap for optimized lookups and selecting Heap Sort, it demonstrates critical thinking in evaluating trade-offs between exact and partial match algorithms, addressing Outcome 3. Additionally, the implementation of HashMap, combined with caching and dynamic UI updates, showcases the ability to create an adaptable solution that meets real-world needs, fulfilling Outcome 4. Focusing on input validation and encapsulating the HashMap within a dedicated class reflects strong software engineering practices and a security mindset, aligning with Outcome 5. I will continue to meet the other course outcomes with enhancement 3, in the category of databases.

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?

While enhancing the artifact in the area of algorithms and data structure I had challenges that led to learning when implementing the search and sort features into the application. One of the biggest learning was understanding the trade-offs between different search and sort methods, particularly in terms of time complexity and memory usage. With hash map searching, I was challenged at implementing the logic in the code to handle searching more intuitively for the user. For example, I had to implement the logic to allow "Green Beans" to appear if the user searched for "Beans".

For heap sorting, the main challenge for me was the time that had elapsed since my time in a data structures and algorithms class. I was able to utilize references to build the heap sort function and call it as a utility in the code. Implementing heap sort also reinforced my understanding of the relationship between array representations and tree-like structures. Throughout the process, I encountered challenges in integrating these algorithms with the existing codebase and ensuring they worked seamlessly with the user interface. This experience highlighted the importance of writing clean, modular code and thoroughly testing each component. Overall, this implementation process not only improved my technical skills but also enhanced my problem-solving abilities, but it led to more in-depth learnings on the pros and cons of data structures and algorithms for specific use cases.

References

Alake, R. (2024, October 3). Heap sort explained. Built In. https://builtin.com/data-science/heap-sort

Geeks for Geeks. (2024, March 28). *Applications, advantages and disadvantages of hash data structure.* https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-hash-data-structure/

Southern New Hampshire University. (n.d). *Undergraduate course syllabus- CS 499: computer science capstone.*