Bruce Gaudet

CS-499

7/24/25

4-2 Milestone Three Narritive

For my artifact in the Algorithms and Data Structure category, I selected the Inventory Management Android App that I originally developed in CS-360. This application allows users to add, view, and manage inventory items in a structured grid layout using a local SQLite database. The project includes a login and registration system, dynamic UI components for adding and removing items, low stock notifications, and basic user feedback through toasts and input validation. The app was built using Kotlin in Android Studio and showcases the core skills necessary for database backed mobile development.

I selected this artifact because it provided the best opportunity to demonstrate my skills in working with algorithms and data structures in a real-world context. The original app relied on structured data input, inventory tracking logic, and user driven interactions with database records, all of which required methodical data handling, condition checking, and efficient logic branching. The enhancements I performed focused on optimizing SQL query execution, improving the input validation logic using conditional structures, and restructuring the inventory search algorithm to better handle user queries. I also refactored portions of the code for clarity and scalability, including simplifying loops, tightening error checking logic, and modularizing repetitive actions into helper functions. These improvements not only enhanced the usability and performance of the app but also strengthened the underlying data structure integrity.

The enhancements I implemented fully aligned with my planned outcomes from Module One. My goal was to demonstrate mastery in designing and evaluating computing solutions that solve practical problems through algorithmic principles and clean data management. I met this goal by making the application more secure, efficient, and maintainable. The updated logic improves how inventory data is stored, retrieved, and displayed, while also ensuring the app remains stable with improper user input. No updates to my outcome coverage plan are needed at this time.

One key challenge I faced was ensuring that database operations, particularly those involving adding and removing items or updating quantities, worked reliably across different devices without data corruption or user interface freezing. Initially, I encountered race conditions where user actions outpaced SQLite write operations, leading to delays or crashes. To solve this, I restructured those database interactions using Kotlin coroutines for asynchronous execution and added progress indicators to inform the user of ongoing operations. I also added edge case handling for low stock warnings, ensuring that alerts triggered only once per item per threshold. These kinds of refinements required careful testing and reinforced my understanding of concurrent logic and memory integrity.

Overall, this enhancement taught me how to think critically about data driven application design, how to refactor and improve existing code using well grounded techniques, and how to ensure that my solutions are scalable and secure. This artifact now represents a strong demonstration of my ability to solve problems using data structures and algorithms, and it is a central piece of my professional ePortfolio.