

CS 499

Professor Dr. Fitzroy Nembhard

Saugat Niroula

1st June 2024

Narrative for Algorithms and Data Structure Enhancement

Artifact Description and Origin:

The artifact selected for the Algorithms and Data Structures category is the Inventory Management Application. This application was initially developed for the CS360 Mobile Architecture and Programming course in September 2023 using Android Studio. The primary purpose of the app is to streamline inventory management in warehouse environments by offering features such as user authentication, inventory display, item addition and removal, quantity adjustment, and low inventory notifications. The motivation for this project stemmed from personal experiences in a warehouse setting, highlighting the need for a more efficient inventory management system.

Justification for Inclusion:

This artifact was chosen for inclusion in my ePortfolio because it demonstrates key skills and abilities in algorithms and data structures, particularly in the areas of data retrieval optimization, advanced data structures implementation, and enhanced sorting and filtering functionalities. The original application showcased foundational skills in creating a mobile application with basic inventory management capabilities. However, the enhancements made to

the artifact further highlight my proficiency in implementing efficient data retrieval techniques, utilizing advanced data structures, and adding comprehensive sorting and filtering features.

Enhancements and Course Objectives:

In alignment with the enhancement plan, the following improvements were made:

1. Optimize Data Retrieval:

- **Indexing on Frequently Queried Columns:** Indexes were added to frequently queried columns in the database to improve query performance. This enhancement ensures faster retrieval times for operations like searching and sorting inventory items. For example, the **searchItems** method was optimized by adding an index to the **item_name** column, significantly reducing query execution time.
- **In-Memory Caching:** Implemented in-memory caching for frequently accessed data to reduce database load and improve performance. The **DataController** class now uses a **HashMap** to cache inventory items, allowing for faster data access and manipulation.

2. Advanced Data Structures:

- **HashMap:** Utilized a **HashMap** for quick lookup and retrieval of inventory items by their names. This enhancement is particularly beneficial for operations requiring frequent access to specific items, such as updating quantities or deleting items.

3. Enhanced Sorting and Filtering:

- **Sorting:** Added functionalities for sorting inventory items by different criteria (e.g., name, quantity). The **sortItems** method in the **DataController** class allows users to sort items based on their preferred criteria, improving the usability and efficiency of the inventory management process.
- **Filtering:** Implemented filtering options to display only items meeting certain conditions. For example, the **filterItemsByQuantity** method enables users to filter inventory items based on their quantities, helping them quickly identify items that need restocking.

These enhancements align with the course outcomes by demonstrating advanced skills in designing and evaluating computing solutions using algorithmic principles and data structures. They also support the course objectives of implementing well-founded techniques and tools for efficient data management and retrieval.

Learning and Challenges:

The process of enhancing this artifact provided valuable learning experiences. Implementing indexing and in-memory caching required a deeper understanding of database optimization and efficient data retrieval techniques. Integrating advanced data structures such as HashMaps expanded my knowledge of algorithmic principles and their practical applications in real-world scenarios.

Throughout the enhancement process, several challenges were encountered. For instance, ensuring the efficient management of the in-memory cache while maintaining data consistency with the database required careful synchronization and handling of concurrent data access. Additionally, implementing the sorting and filtering functionalities involved understanding and

applying different sorting algorithms and filtering techniques to meet the application's requirements.

Overall, the enhancements made to this artifact demonstrate significant progress toward achieving the course outcomes and showcase my capabilities in algorithms and data structures. The improved application not only offers enhanced functionality and better performance but also reflects a high standard of code quality and maintainability.

This narrative, along with the enhanced artifact, will be included in my ePortfolio to showcase my skills and abilities in algorithms and data structures, supporting my career advancement and demonstrating my proficiency in this critical area of computer science.