**Artifact Two Narrative**

Brandon Stallons

Professor Bryant

CS-499-13403-M01 Computer Science Capstone 2024 C-5 (Sep-Oct)

September 28, 2024

The second artifact is an Android application I developed in my CS 360 Mobile Architecture and Programming course that tracks a user's weight. The app allows users to register, log in, set a goal weight, record their daily weights, and receive an SMS notification when they reach their goal weight. The app uses an SQLite database to store all user details within three separate tables. I selected this artifact for my ePortfolio because it provided an excellent opportunity to demonstrate my ability in software design patterns and data management techniques in a real-world context. Additionally, the enhancements made to this artifact accurately show my abilities in algorithms and data structures.

First, I refactored the DatabaseActivity class to implement a singleton pattern which increases efficient resource usage and dramatically improves data access consistency throughout the application. This enhancement also reduces the risk of data inconsistencies and avoids multiple database connections, which improves performance. The refactoring was followed by including a cache inside the singleton to store frequently accessed data such as user IDs, usernames, and goal weight. This was the original enhancement that I planned on making but ran into challenges that led to additional enhancements. This enhancement demonstrates my understanding of data structures such as hash maps, hash sets, and algorithms that aid in data retrieval optimization. The final enhancement I completed in this artifact was ensuring that all the other activities used the shared singleton instance. This allows the application to maintain a consistent database state and cached data.

The enhancement performed on this artifact improved it in many ways. The enhancements optimize the application's performance by reducing the numerous redundant database queries by caching the frequently accessed user data, which made the app more responsive and efficient. The improvements also increase code maintainability and scalability. This was accomplished by applying well-founded design patterns and data structures and by refactoring the code with industry best practices in mind. By allowing the code to be more flexible and better suited to scale with potential new features, the improvements enhanced the current capability of the application and laid a solid foundation for future development.

Through the enhancements made in artifact two, I have achieved course outcomes three and four. These are the two course outcomes that I had initially planned to achieve in the outcome-coverage plan. By refactoring the DatabaseActivity class to implement a singleton pattern and a caching mechanism, I applied algorithmic principles to solve the problem that was inefficient and inconsistent database access, which covers outcome three. Outcome four was achieved by utilizing well-founded design patterns and data structures that enhance the artifact's efficiency and functionality while simultaneously improving user experience by enhancing the apps performance. The outcome-coverage plan remains unchanged. I have achieved the first four course outcomes through the enhancements to the first two artifacts and the additional code review. The final course outcome will be achieved through the enhancements made to the final artifact.

Throughout the process of researching potential enhancements and implementing them, I learned many different things. I learned a lot about the importance of cache and how it can optimize data retrieval and reduce the number of unnecessary database interactions. I was also able to refresh the database knowledge that I had lost from lack of use. I learned how a singleton pattern could improve my application by increasing performance and data consistency. I learned new debugging skills that allowed me to confirm that the implemented enhancements were working correctly. The main issue that I ran into regarding this artifact was the multiple instances of the database helper, which was causing inconsistencies in the cache. After researching the issue, I realized that I needed to implement a singleton pattern so that all the different activities used the single singleton instance. The first few implementations broke some of the existing features, which took some tweaking to figure out, but it was minor compared to the first challenge. In summary, I gained a lot knowledge, and I was able to reclaim information that I had not utilized for a long time. I faced numerous challenges, nevertheless it was an enjoyable and rewarding experience.