CC-499 Capstone Project

4-2 Milestone Four: Enhancement Two: Algorithms and Data Structure

Name: Tyler Strickland

Professor: Fitzroy Nembhard

1. **Briefly describe the artifact. When was it created?**

   The artifact is an Android studio application I made in winter of 2023 for my mobile development class.

   What is it?

   It's a weight tracking application that records your weight according to a selected date.

2. **Justify the inclusion of the artifact in your ePortfolio.**
   As a up and coming developer I believe this artifact helps show my knowledge of Android development which I believe is a great starting point for my career in software engineering.
   Why did you select this item?
   I selected this object because it fits well in showing off my capability of using my learned practices in coding to improve a mobile project. A majority of developers will show their abilities in more popular languages such as Python and Java, although I want to present my abilities in Kotlin + Java to stand out in the mobile development market.
   What specific components of the artifact showcase your skills and abilities in software development?
   My priorities with this artifact lie in it's ability to show off major programming skills that a majority of projects will use such as its use of databases, the flow and processing of data between multiple classes and functions, and it's modularity that allows parts of the code to be reused and repurposed for other projects.
   How was the artifact improved?
   I implemented user ID numbers using an incremental counting system, utilizing a recursive algorithm as I utilized a while loop to keep adding to an integer and calling it to check if the ID is taken. This improvement makes sure the user doesn't lose their data or access someone elses if they are to change their username. The previous system used usernames to specify the information to pull in the database, however if someone were to change their username this could cause problems in the long run. This new algorithm checks if the user exists, if they don't at the point of creating an

account the system counts from 0 until a user ID that isn't in use is found. It then assigns this number to the user's data table since that's currently the only information associated with the account. Because the data table is made and checked in another class this would be similar to a divide and conquer algorithm, assigning the userID in one class and using it to find the user's data in another. This method could result in longer and longer wait times as more people create accounts, thus a randomized ID system would be more practical for larger scale projects.

The current data structure of the system is a Hash Table, seperating the tables by user ID then accessing weights and dates from that table. Because I implemented a userID, and the username was originally only used to log into the account then used to change the search parameter of the table and deprecated, I had to refactor the code to keep the username in temporary storage for another process that would grab the userID from the database before then deprecating it. Replacing the searched table name from the default table name plus the username to the default table name plus the userID required a lot of passing of the temporary data: in the login class I implemented a get method to pull it from the permanent data, passed that to the graph helper class to help be implemented into the permanent table names.

3. **Did you meet the course outcomes you planned to meet with this enhancement in Module One?**

   I actually didn't meet my course outcome for this enhancement as it does improve the user experience in terms of time to search but doesn't ultimately make the application much more convenient, only saving likely a few seconds on average.

   Do you have any updates to your outcome-coverage plans?

   To make the app more convenient as I planned for this enhancement I would only really be able to adjust the user interface, which doesn't fall within the focus of this class. Thus, I'll be dropping the course outcome one in favor of improving efficiency, as the efficiency of the code is inarguably an enhancement.

4. **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?**

   The implementation of the user ID was harder than I anticipated, mostly just implementing the functions to call the data between classes as I had to refactor some variables and values to be more maleable in how changeable they are and their accessibility (private vs public) without breaking the current functionality.