CS 499 Module 5 Narrative

James Nikolaou

CS 499

10/04/2024

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

The artifact is an android application that is called the ‘weight tracker app’. It’s function is to provide an application that allows the user to set a target weight that they hope to achieve and track it daily via each day of the week. It provides a login page with an option to sign up (via a sign-up page) for unique access and security. Once logged in, it allows the user to input a target weight, add weights per day of the week, delete weights for each day, add/modify weights per day, and adjust the target weight. The data within this application is stored in one central SQL lite database, amongst 3 tables: Login, WeightGrid, and TargetWeight tables. This project was created earlier this year in my CS360 course around May 2024.

* **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 software development? How was the artifact improved?**

I selected this artifact because it’s my favorite demonstration of a database system that’s doing work behind the scenes. Most people think of managing tables/databases with managing a large base of numbers and values. This project I feel shows how the everyday person may interact with databases without even realizing it. The original piece utilized a database named ‘appDatabase’ to store 3 different tables, ‘login/weightGrid/targetWeight’ to store user information to provide the usage of a login page, sign up page, track their daily weight, and set/modify a target weight.

The artifact was improved by improving the user experience by having a unique personal experience instead of sharing all the data between users by expanding the utilization of the database by adding new more complex queries and utilizing code to further manipulate the data within the database. The new stronger queries were focused on the second concept of manipulating the data more effectively and diversely. I added count queries to count the number of values within a certain parameter to in turn create code that allowed the addition/modification of data to the database to be more personalized.

For the target weight table and data, a select count of all values was executed to check the number of values in the table. The goal is to only have 1 target weight per unique user, whereas before there was only one target weight available at a time, with every single entry being added to create massive bloat in the database. The uniqueID comes from the creation of a user in the login page section, and with this uniqueID each user only gets one uniqueID. Ensuring the number of values in the target weight table is less than or equal to the uniqueID provides two routes: If it’s less than, then it adds the target weight to the table with the uniqueID to reference it for that user. If it’s already in there, then it modifies the current entry for the unique user.

The weightGrid table benefited from a similar concept. The new query checks for the count of days of a specified day (ie. Mondays, Tuesdays, etc) based on a given day and a given uniqueID. This data is then manipulated in the ‘add\_modify\_weight\_entry’ java file. It creates a checker to count the number of days with the query mentioned and checking for if a value exists at this spot or not. If it does not, the program will add the day and weight for that day for the unique user ID. If it does exist, it will update the weight for that day instead for the unique user ID. This again keeps the code from getting bloated, provides unique data usage for each unique user, and stops overlapping of data so the data is manipulated properly.

These two queries with addition of supporting code accomplished the main goal of the enhancement, being to provide a unique data set to each unique user where as before each user shared a data set.

* **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 wanted to focus on 3, 4, and 5.

3 is to design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.

For this outcome, my computing solutions were focused on manipulating the data within the built in database in the program. Besides navigating between the screen functions, the entire program was reliant on the usage of the data within the database. From computing the solution to either provide access to the app or not, to sign up a user or not, to add or modify weights to the target weight, to adding weights or modifying weights for the daily weights. Each user interaction input/output relationship went through the database in some regard. The predominant usage of CRUD operations within the program were the create, read, and update portions. The delete operation was limited to removing the value for a specific day. The create operation was used by inputting data for each of the three tables. The read operation was used to get any of the data within the tables, and the update operation updated the target weights or the daily weights.

4 is to demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value an accomplish industry-specific goals.

For this outcome, my organization of code and the overall design was to modulate the code by separating the files with providing enough of a relationship between the files to allow access from all applicable items between them. Each ‘code’ file that did the main work of each section had a data file and DAO file to ensure that the items were organized by like items. The code did allow for overlap between the files by constructors, objects, and functions being allowed to navigate through the various files. These allowed for a ‘plug and play’ concept with ensuring a proper level of encapsulation and inheritance was present.

For innovative techniques and skills, the usage of a singular database with three tables throughout the entire application while still being able to provide a unique user experience for each unique user kept the code simple, secure, and most importantly improving the user experience.

5 is to develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

The original artifact did have some decent security for access the data within the app itself, but it severely lacked in keeping all user’s data separate from each other. Once a user gained access to the application, they could see whatever data was manipulated last and could change it however they wanted. It kept no integrity in the data once a user gained access.

The enhanced artifact focused on ensuring each user had a separate set of values they could manipulate and only see/access. This keeps all the user’s data secure and confidential. This database setup does not allow for any access to the database itself, ensuring that there is no method to inhibit the functionality of the application. Input validation and unique user access/experiences are the driving forces in the updated security portion of the program.

Outcomes 1 and 2 were not part of my original goal but they were incorporated well within this project.

1 is to employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science.

For this outcome, well labeled documentation and modular code allows for multiple users to work on the project at a time and work on different things without impacting the other developers. The documentation allows any developer to step in at any given point and understand what each file of the code does, with inline comments to explain each part of the file. These two techniques allow teams to plan their development easier and more efficiently.

2 is to design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.

For this outcome, I do have simple yet informative text and instructions to help navigate the user. The login page provides a standardized username/password input with a button to login that should be comfortable for most users. The sign-up function prompts the user to enter the desired information for the application. Inside the app, there are clear labels to indicate what does what and what goes where in the app, with the fill in text portions providing hints/examples to help ensure the user puts in the proper input.

The comments within the code provided who wrote the code, what version, and a detailed explanation of each file within the program to help other developers understand the layout and functionality of the code. There are many inline comments within the code in each file to explain each item and its relevance to the program and the demonstration 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?**

In terms of adding code there honestly wasn’t a ton to do. A lot of the foundational work was done already. The bulk of my work came from learning and studying more about not only how SQL lite works specifically in an android environment, but how to manipulate it via Java and make sure it is reflected in XML appropriately. I really enjoy working on Android projects but having so many moving components at once can really become difficult quick as it seems every time one thing is changed/added it impacts three other items. The incorporation of a few strong and direct lines of code for SQL queries and java manipulating these queries changed a non-confidential database to a confidential database between users. Then the usage of the creation of the SQL database, the DAO’s, and the data sets for these tables and database were a tough concept to get set up as it feels different than other databases I’ve worked on so far in my coding career. The multiple layers that weave in and out of the mobile development world made this a challenging but rewarding project to work on.