Caleb Allen

Southern New Hampshire University

Computer Science Capstone

10/13/23

Enhancement Narrative

Category I

Files identified for Category I: Software Design and Engineering

SumoWeightTracker application (all files)

activity\_login.xml

For this category I decided to update the software for the SumoWeightTracker Application that I created in CS360: Mobile App Development.

The “activity\_login.xml” is the layout for the applications login, or “splash” page. This file included the methods and instructions for the application to have someone log in as a user or an administrator and assign roles to their respective session(s). Once logged in as a user, there is only one activity, and that is to input a weight for the aspiring sumo wrestler.

I have updated the application with more things to do and wanted to have all options available to the user after logging in. The main menu has been updated to now navigate to multiple options and have a return method, so the user is never unable to exit the app without having to close it completely and restart. I wanted to include this artifact in my ePortfolio to demonstrate my ability to add more functionality to an existing application. The files I selected as an example of these improvements are ones that I feel reflect the most amount of change. I have added buttons to the main page and followed best practices by commenting on the code for other developers (in a team environment) to review or understand why I did what I did, or what the intention of my code is. Adding comments to the code also increases security by adding a layer of transparency. The rest of the application maintains the old functionality. You will have to log in to modify your weight inputs or deleting your entries (ensuring integrity and authorization, best practices – security), but now you have access to statistics, recipes and providing suggestions to improve the application or experience.

When updating or creating code for this project, I followed best practices by testing often and commenting on code with brevity. Using the automated testing function in Android Studio is also considered best practice, not only for code security, but productivity. Validating all data that is input (making all numerical inputs only accept numbers, all alphabetical inputs to take letters, etc.) is a best practice I also followed as a habit learned from coding.

Category II

Files identified for Category II: Algorithm and Data Structures

NotificationSMS.java

Activity\_notificationSMS.xml

AndroidManifest.xml

For this category I have expanded the complexity of the SumoWeightTracker application by including a web “scraping” function. I updated the “NotificationSMS.java” class to be able to reach out to the applications website and pull text information down to display in a notification window. Users can now scroll through notification texts that can include weight gaining recipes and tips on how to gain weight for the Sumo hobbyist. I had to include permissions for this update, which was completed in the AndroidManifest.xml

This upgrade was quite the undertaking. Reaching out to a website includes lot of includes (no pun intended) to work and permissions to be set. A note for me to keep in mind is to start planning and configuring all the permissions within the pseudocode. Permissions for apps are common, but important to know so for limitations from the beginning.

Once we make the connection to the website, we need to then pull data (if there is an update to pull) so for this we are using a tool called textview. Textview is a text version of a website, a bit rudimentary but works for the functionality I am trying to incorporate into the application for now. The app now connects to a website (sumoweighttracker.com) and once their searches for textview fragments to bring update the notification activity. This took so long figuring out from stack overflow and youtube the steps needed to reach out to a website and pull data, as this is a bit of a very proprietary function it seems. Not a lot of open-source material covers this topic.

Category III

Files identified for Category III: Database

Activity\_database.xml

AndroidManifest.xml

Database.java

DatabaseHelper.java

For this enhancement I had to create a litesql instance on the app in java form. This is the simplest form I could think of that would include all enhancements I wanted to incorporate at this time. By adding the SQL lite database, I have also, by design, increased the scalability of the application. Adding this method now to the application should have been included in the initial code but was outside the scope of the original coursework. To simplify the task at hand I opted to create a DatabaseHelper Class that would serve as the source code for all current and future database updates for developers. All in all, I added the functionality of sorting data, and changed the way entries are removed from the database. I have also included a timestamp (not editable by users at this point) for all entries. I also cleaned out and repopulated the database with updated timestamps so that functionality would not be lost when the application goes to look for information that is not there (when I expanded the database to include and look for timestamps.)

Most of the updates to the Database class were for the inclusion of the new DatabaseHelper class and needed parameters for the timestamp methods. More permissions were needed (read and write to storage) since the database will be stored on the phone or tablet’s internal storage and will be edited by the application (modifying the data structure of the phone). This portion was missing until now and like all other permission requirements should be thoroughly reviewed, at least in my workflow.

Throughout this course I have employed strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science by providing contextual, in-code comments that result in easily readable and understandable.

I have designed, developed, and delivered professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts by discussing experiences and best practices in communication and clearly conveying my ideas and explain my thought process through written communication.

I have designed and evaluated 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 by programming solutions that solve logic problems and implement them in software. Reaching out to a website with an application may not seem that in depth, but it is a very involved evolution. I had to create a few methods to cover all the functionality needed to complete this step and it is still, in my opinion, an amateur attempt. I have also discussed experiences and best practices in designing and evaluating computing solutions within this narrative for the project.

I have demonstrated the ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals by employing iterative testing techniques in the code by using the software development life cycle to create realistic production schedules for software projects, creating industry-standard software designs, discussing experiences and best practices in using well-founded and innovative techniques, skills, and tools in computing practices.

I have developed 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 by address potential design flaws in software architecture during the requirements phase (user permissions), eradicating security vulnerabilities or bugs by correcting all errors and addressing all warnings in the IDE, defining an approach that ensures all data are explicitly validated. When creating the “DatabaseHelper” class I demonstrated the ability to consider future changes to objects and classes by making all methods reside in it. Now when adding more activities that want to interface with the database there is a common structure that is designed to manage these methods and keep them all in one place, instead of creating new classes that must be updated whenever a new class is added.