Ty Wheeler

3-2 Milestone Two Narrative

CS 499

5/19/25

*Artifact description* I selected my Goal Weight Tracker Android app that I developed in my CS 360 mobile programming course for this artifact. This app allows users to log their weight entries, set a goal weight, and receive notifications when the goal is reached. It was created using Java and SQLite within Android Studio and included features such as local data storage, SMS permission handling, and a basic login screen. The original version functioned as intended but was lacking in security features, modularity and modern design principles.

*Justification for inclusion in ePortfolio* I selected this app for my ePortfolio because it demonstrates my skills in full-stack mobile development, secure application design, and user interface enhancement. The enhancements I proposed and implemented showcase my ability to improve an existing application through refactoring, secure authentication implementation, and modern UI/UX redesign. These are core competencies in software development and engineering. These key enhancements specifically were implementing Firebase Authentication with email verification and two-factor authentication, refactoring the project to follow the Model-View-ViewModel architecture pattern, and redesigning all screens using material design components for a more modern and user friendly appearance. I also fixed key issues such as validating user inputs for specific fields. The previous version lacked this so users could input invalid weight or date entries into the data logging screen. There is also improved navigation with consistent back buttons in the top left corner. These improvements transformed the app from a functional prototype into a more secure and polished product that better aligns with industry standards.This artifact directly addresses course outcomes such as the use of well-founded and innovative techniques, skills, and tools in computing practices to implement secure and scalable solutions. It also demonstrates evaluating computing solutions using algorithmic principles and software design standards. The last course outcome it demonstrates is developing a security mindset by anticipating potential flaws and integrating secure authentication and permission-based flows

*Reflection on the enhancement process*

Enhancing this app taught me about the importance of modularity and clean architecture in software development. Migrating to MVVM involved creating WeightRepository.java and WeightViewModel.java, and refactoring ActivityDataActivity.java to decouple logic from the UI. This separation improved the overall scalability and readability while also allowing the app to react in real time and update the RecyclerView when weight logs change. Implementing Firebase Authentication required changing MainActivity.java to replace the local login system with Firebase email and password authentication. I also introduced a simulated 2FA mechanism using SmsManager, where a 6-digit code was sent to the emulator and validated before going to the app’s main dashboard. I updated build.gradle to include Firebase and lifecycle dependencies and configured the google-services.json properly to integrate with the Firebase console properly. For input validation, I updated ActivityLogActivity.java to include regex checks and date parsing logic using SimpleDateFormat to validate weight input and MM/DD/YYYY formatted dates. This prevented invalid or unrealistic entries, like 20/00/205 or 145.56789 lbs, from being saved to the database. Regarding the visuals and UI I redesigned activity\_main.xml, activity\_data.xml, activity\_log.xml, and activity\_sms.xml to follow Material Design guidelines. I used TextInputLayout, MaterialButton, and proper constraint layouts to create more accessible and modern looking interfaces. Each screen now has more consistent styling, spacing, and back navigation with ImageButton elements.

One challenge I encountered was avoiding ClassCastException when updating the layout of activity\_main.xml. I originally changed the register button from a button to a TextView for styling purposes, but this broke the cast in MainActivity.java. I resolved this by styling the button to visually resemble a link while keeping the same type compatibility. Another challenge was keeping the local SQLite weight log functionality intact while introducing Firebase. I resolved this by maintaining a hybrid architecture where Firebase handles authentication and SQLite continues managing personal user data. Overall this project enhanced my understanding of Android architecture patterns, secure authentication workflows, UI design principles, and user input validation, all while evolving an existing prototype into a polished mobile application.