

CS499 Milestone 2: Software Engineering and Design Narrative

This artifact is an Android application I made last term in CS360. It was designed and developed to help users track their weight over time. It features user login, weight input, a dashboard display and other various settings allowing for SMS notifications and goal tracking. For CS499, I decided to use this app to enhance to align it with modern design principles and industry standards. I selected this artifact because it represents a comprehensive demonstration of software engineering principles, and my ability to recognize and implement them efficiently. To satisfy this, I restructured the whole app to follow a Model-View ViewModel(MVVM) architecture. Included in this, I added persistent data storage with Room, and responsive UI with LiveData and RecyclerView. Enhancements included redesigning the login flow, I ran into a few threading violations which I ended up fixing, and implementing a dashboard backed by the database queries.

The enhancement aligns with the course outcome: “Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for implementing computer solutions”. We achieved this through refactoring to MVVM and asynchronous DB access. I’ve improved maintainability, security in passwords and database access, and overall responsiveness of the app. Right now, it is a very early version of the MVVM architecture and room persistent storage, so there are still features to add. I need to add a way to delete weights from the table, as well as add some input validation for weights to ensure users cannot abuse the database at all. The current LiveData dates in the dashboard display the raw date value, so I can convert that to actual readable dates.

Like mentioned earlier, I faced a major challenge when trying to resolve threading issues in Room database access. I learned how to decouple UI and data logic using these ViewModels and Repository patterns. I also gained practical experience in the debugging RecyclerView binding issues. Lastly, I learned how much easier it is to dynamically update UI using LiveData. I tested the user login kind of lightly, I wasn't really sure if I should add input validation to limit what users could put in the user or password field. For example, should I allow users to use special characters like "-", "_", ";", ":" or should I just allow letters and numbers. I understand allowing extra special characters increases security, but it also raises the risk of users forgetting their password.