Jeffrey Flores

September 21, 2025

CS-499 – Milestone Two – Narrative

Weight Trackit is a mobile application designed to make weight logging and goal tracking fast, private, and motivating. Core features include daily weight entry, goal setting, progress visualization, real-time sync via Firebase (so entries persist across devices), and milestone notifications when targets are hit. The product features a clean, low-friction UI designed for users who prioritize functionality over clutter. The initial Android version launched during my Mobile Architect and Programming course in April 2025, and I created the iOS version in September 2025, bringing the project to a multi-platform state.

I selected this app because it represents the kind of work I want to do professionally, mobile engineering tied to cloud data and user-centric design. It showcases my skills with databases. It shows my ability to use Firebase Functions and write backend functions that communicate with the front end to send notifications. I was able to set up a listener in the front end of the app to work with this Firebase function. Every time a weight is created, it is compared in the backend, and then sends a notification if the user reaches their weight in the app.

The improvement I made with this enhancement was to automate the weight tracking feature. I enhanced the app by displaying notifications to users when they have reached their weight goals. This specific enhancement was more for the user experience and to give them an incentive to invest in using the app. Instead of just logging their weight every day, the user can look forward to being congratulated every time they hit their weight goals.

One of the course outcomes that I met with this enhancement and with this artifact was the algorithm design and evaluation. This allowed me to model a notification alert into an event listener function. I was also able to hit the Engineering tools and practice outcome by using Firebase SDKs and using the Firebase CLI to deploy backend functions. I was able to organize functions using the TypeScript language to write the files needed for the backend to work. Another outcome I achieved with this enhancement is solving a given problem using algorithmic principles. I was able to manage tradeoffs with using email notifications or just staying with in-app notifications.

The first issue I ran into when implementing the backend functions for notifications was a permission issue with using npm on the command line. I was forced to reconfigure the npm settings in my command line tool and set the owner of the project to myself. Another issue I ran into was deploying the functions to my backend Firebase database. To fix this issue, I had to go into my Google Console settings and grant access to myself in the console’s IAM settings to push the functions to the database. I learned about the tradeoffs of having in-app notifications versus email notifications. I saw that there were more possibilities with in-app notifications that can draw the user back to the app, rather than an email. There are also a lot more integrations needed for email notifications than there are for in-app notifications, which speeds up the time for delivery for a feature like this.