CS-499

Professor Ogoh

Michael Turco

27 November 2025

Milestone Four Narrative

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

The artifact I chose was my full-stack mobile application from CS-360 Mobile Architecture and Programming. I called the app “Track My Weight”, and it is a weight tracker app that lets the user create an account and login, add new weight values, and check their weight history. I created this app back in February of this year, so just around 9 or 10 months ago. The current artifact can be found here, but you can check the commit history to go back to the original: [https://github.com/michaelTurco/CS-360-Mobile-Architecture-And-Programming]

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 picked this artifact to include in my portfolio, since it is a full-stack application and had a lot of room for improvement. Through my learning at SNHU and personal projects, I’ve learned a lot about both front-end design for UI and user experience, as well as back-end database management and coding standard best practices, and this artifact will be a good example of how I’ve grown over the years. For this artifact in the category of databases, I switch from using a local MySQL database, to a properly authenticated and functioning Firebase database. I implemented user accounts with register and sign in functionality, and each user has a private

document that only they can read, so that their data is secure. I also finished up an old unfinished page which was the ‘Account’ page. I added a way to change your account’s nickname, and a button to sign out of your account and return to the login screen. It was a lot of effort to refactor the code to get the user authentication, and Firebase Firestore database reading, and I think it is a great example I can use to show my skillset.

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?

Yes, I did indeed meet the outcomes I had planned back in Module One. I planned to reach course outcome #5 mainly, since it was related to adding security, but also outcome #4 too. By adding Firebase authentication and Firestore database, user accounts are much more secure, and the data is now hidden from users who don’t have access. This is drastically different from having stored the user passwords and data in plaintext on the local MySQL database, and it’s a big leap towards a more realistic app.

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?

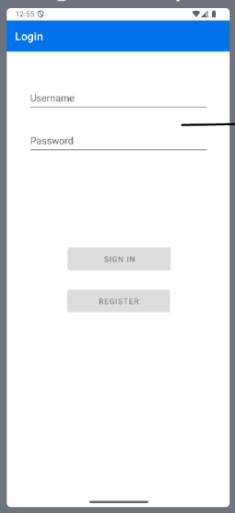
This one was a little bit more challenging and in-depth, since I was constantly referencing Firebase documentation for how to access and interact with the database and authentication system. The documentation was incredibly helpful, and I learned a lot about how to set up Firebase in an Android Studio application along the way. I started this enhancement by working on the account page, with the old MySQL system, and made the sign out button and nickname changing areas. Then I began the long process of slowly integrating Firebase authentication over the existing authentication, and it took a while to figure out what could be deleted and what couldn’t, without the project becoming un-runnable. After I got the authentication working, I

kept getting crashes relating to not being able to access the old outdated database variables since all of my page fragment code was still referencing the old database format. I had to disable basically every feature of my app, and slowly rebuild it using the new Firebase data system. It was a little tricky since in some spots I had to add new functionality to ‘wait’ for the response properly, since it’s possible that Firebase times out or returns a bad value. Normally the local MySQL database had an instant lookup time, but since Firebase is hosted, it now has to send a web request which can take time or fail. I had to add spinning loading icons and failure messages in the case where the Firebase data never returned. In the end, I’m very happy with how this implementation turned out. I love being able to see the data populate in Firebase as I add new weight values on my app, and disappear when I click the x, almost instantaneously too haha. I ended up refactoring my Firebase code to split up the authentication aspect from the login fragment, and it turned out very clean and readable. I'm very satisfied with the result!

I’ve attached an image below of my updated LucidSpark diagram showing my actual implementation in Android Studio:

Databases

Original Implementation



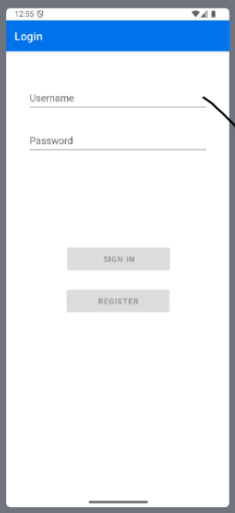
username and password is only stored locally, no authentication is actually happening

switch from local MySQL to Firebase database, add authentication

add a section to configure your 'nickname' that the app greets you with

add a button to 'log out' of the account

Rough example after enhancement (made using photo editor)



attempt Firebase login / register using these credentials, read the stored account data

limit the name to up to 16 characters

un-authenticate them, and bring them back to the login menu

Actual Implementation after Milestone Four