Ryan Spadt

CS 499 Milestone Four Narrative

11/28/25

“The artifact selected for this enhancement was a mobile application created in CS 360 last term. We were given an option to create one of three application concepts. I elected to create a weight tracking mobile application. This application was created using Java to run an Android mobile application. That would allow a user to sign in or register an account. Once they are logged in they would be able to set a weight goal and input weight entries on a daily basis to track their progress towards that goal. The application also supported the ability to send SMS messages to the user when they accomplished their goal based on the setting inside of the application and phone permissions.

I selected this artifact because I found making mobile applications challenging, exciting, and rewarding. It’s something I’d actually like to do in my free time after I have graduated from SNHU. I also have been tracking my weight in a similar fashion inside of a Google Sheets spreadsheet and thought this would be a perfect opportunity to create something I could use. If I wanted to use this application I would need to write it for iOS and found Kotlin Multiplatform Mobile. So I wanted to learn Kotlin Multiplatform Mobile anyway. And with the enhancement requirements I thought it aligned perfectly to migrate it from Java (Android-only) to KMM and have support for Android and iOS.” (Spadt, Narrative Milestone Two).

In all of my milestones, I have selected to complete enhancements on the same artifact. In this case, I improved upon the work that was completed in milestone 3. And the artifact has been improved significantly in this enhancement. The following improvements were included in this enhancement:

1. The Authentication process was moved from a Client-sided transaction to being orchestrated by the server-side.
2. Passwords are no longer stored on the client-side they are stored on the server-side and are hashed using Argon2id.
3. A Server sided application was created and included in the build.
4. Database was configured using PostgreSQL as the server sided database solution.
5. Authentication routes were created to facilitate the login and registration processes
6. Protected routes were created that require a user to be logged in.

- GET: /health – returns a text response to confirm API is working as expected
- GET: /calories/userId – returns all calorie intake records for a user from the server
- POST: /calorie – stores a new calorie intake record into the server database
- GET: /exercises/userId – returns all exercise records for a user from the server
- POST: /exercise – stores a new exercise record into the server database
- DELETE: /exercise/id – deleted an exercise record with id from the server database
- GET: /weights/userId – returns all weight records for a user from the server
- POST: /weight – upserts a weight record into the server database. Unique on userId and date
- DELETE: /weight/id – deletes a weight record with id from the server database
- GET: /goals/userId – returns the user's goal if it exists or null
- POST: /goal – upserts a goal record for the user into the server database unique on userId

7. Protected routes via JWT-Based authentication.

8. Setup Serialization/Deserialization for Ktor to send data over the network

9. Used Exposed for type-safe DSL and lightweight ORM for accessing the PostgreSQL database.

10. Setup a sync manager that when the app opens it syncs and non-synced records from the local database to the server database

11. Local database records on actions get synced to the server.

12. When the user logs in, it retrieves all the user's data from the server so they can use the app across multiple devices and have their data. Or if they wipe app data they can still get their data.

The expected course outcomes for this milestone were outcomes 3, 4, and 5. I believe I have accomplished all three aligned course outcomes in this enhancement. My ability to implement multiple database structures and the transformation of that data; demonstrates my ability to design and evaluate computing solutions that solve given problems using best practices and standards. I also needed to design these solutions in a way that navigated the trade-offs of the solution. Outcome four was accomplished by implementing a solution that utilizes best practice tools and solutions for driving solutions that deliver value. With my use of PostgreSQL, Argon2id, Exposed, and Ktor to hook up my client application up to a server application. And outcome five was accomplished by my focus on securing the resources of the application. Requiring users to authenticate over the network to the server. The implementation of access tokens that provide the user access to the application upon their session instantiating. The JWT-Based authentication I implemented to protect routes that require a user to be authenticated. As well as, the use of Argon2id to hash passwords. Designing the solution in a way that over-network communication is protected as much as possible. Returning UserDto to not transport protected fields like password back from the server. I designed it in a way that minimizes design flaws and security risks to ensure user privacy. I also ensured that any technology or dependencies that were added were the latest version to ensure that any vulnerabilities from old versions were not introduced.

This enhancement required a great level of effort. I had to lower the scope from a hybrid MongoDB and MySQL backend to simply a PostgreSQL backend. I experienced issues with timing, especially with data coming over the network instead of being locally based. For example, I was having an issue with the passive calories burned calculation not properly calculating. It was pulling the most recent weight and not necessarily the most recently authenticated user. To solve this, I needed to add a status and createdAt attributes and filter on them in the Session table. I also wrestled with configuration issues and setting up the server-side to run locally, I needed to set up environment variables and they wouldn't work so I hard-coded them in the dbConfig. Additionally, I had some challenges with the platform specific httpClient implementations and getting the access token to be asserted into the header.

I learned a lot about networking with Kotlin and Ktor during this enhancement and overall I like the way they are able to be configured and tested. It felt good to be back working with PostgreSQL as it was my go to database solution when I used to set up Django web applications. I think the solution is getting closer and closer to be production ready and I would like to continue working on it and improving it. With the holiday season I was too busy to get more done on this enhancement. You can see I started implementing the calorie intake module to set up tracking calories ingested. With a little bit of work it will be a full-scope calorie tracking web application.

Spadt, Ryan. “CS 499 Narrative for Milestone Two.” *CS 499 Narrative for Milestone Two*, 15 Nov. 2025. Accessed 22 Nov. 2025.