Ryan Spadt

CS 499 Narrative for Milestone Two

11/15/25

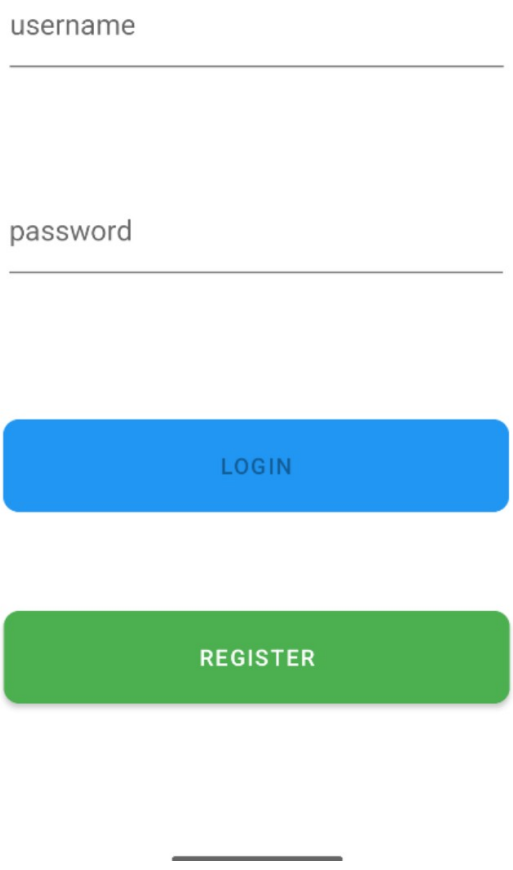
The artifact selected for this enhancement was a mobile application created in CS 360 last term. We were given an option to create one of three application concepts. I elected to create a weight tracking mobile application. This application was created using Java to run an Android mobile application. 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.

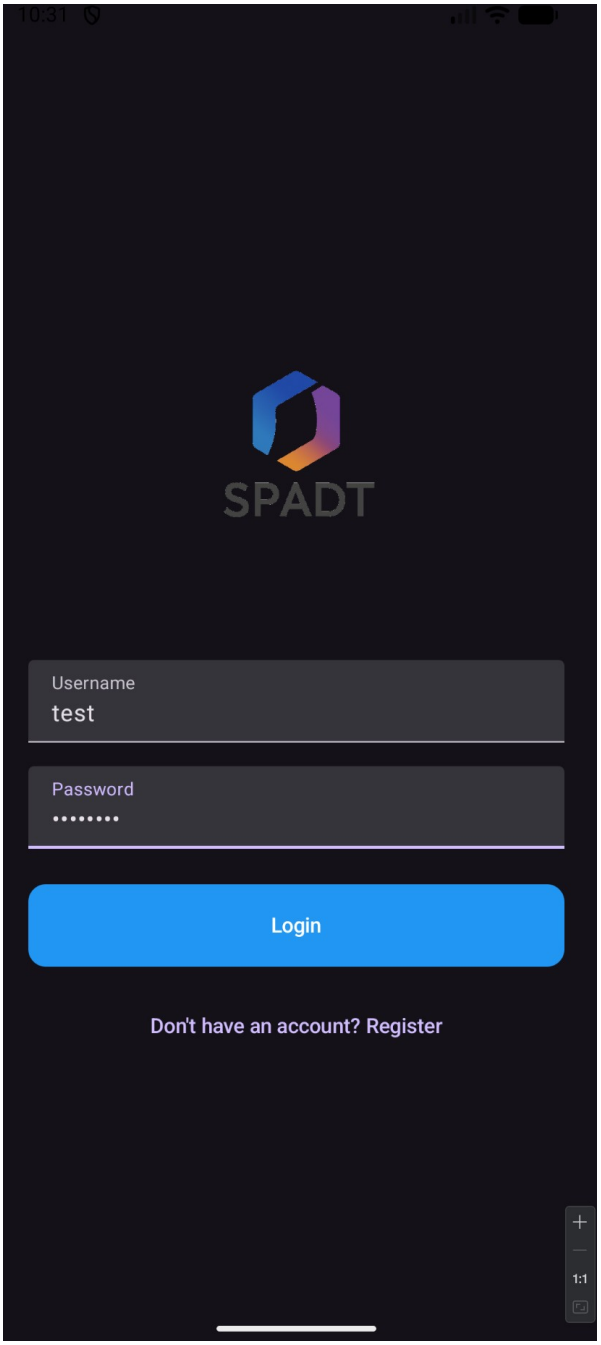
I selected this artifact because I found making mobile applications challenging, exciting, and rewarding. It's something I'd actually like to do in my free time after I have graduated from SNHU. I also have been tracking my weight in a similar fashion inside of a Google Sheets spreadsheet and thought 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.

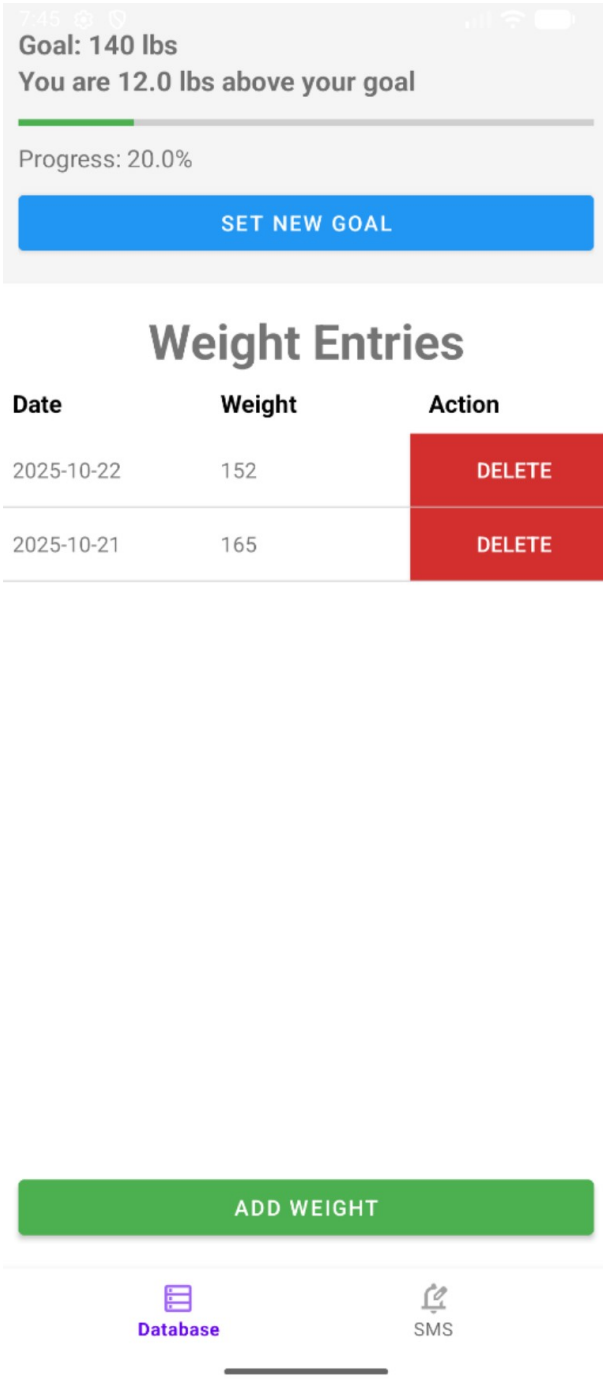
This enhancement was a greater level of work than I had originally anticipated and I'm glad I began this process very early. I did not know anything about KMM when I began this project, so I think this showcases my ability to learn new technologies/frameworks and apply

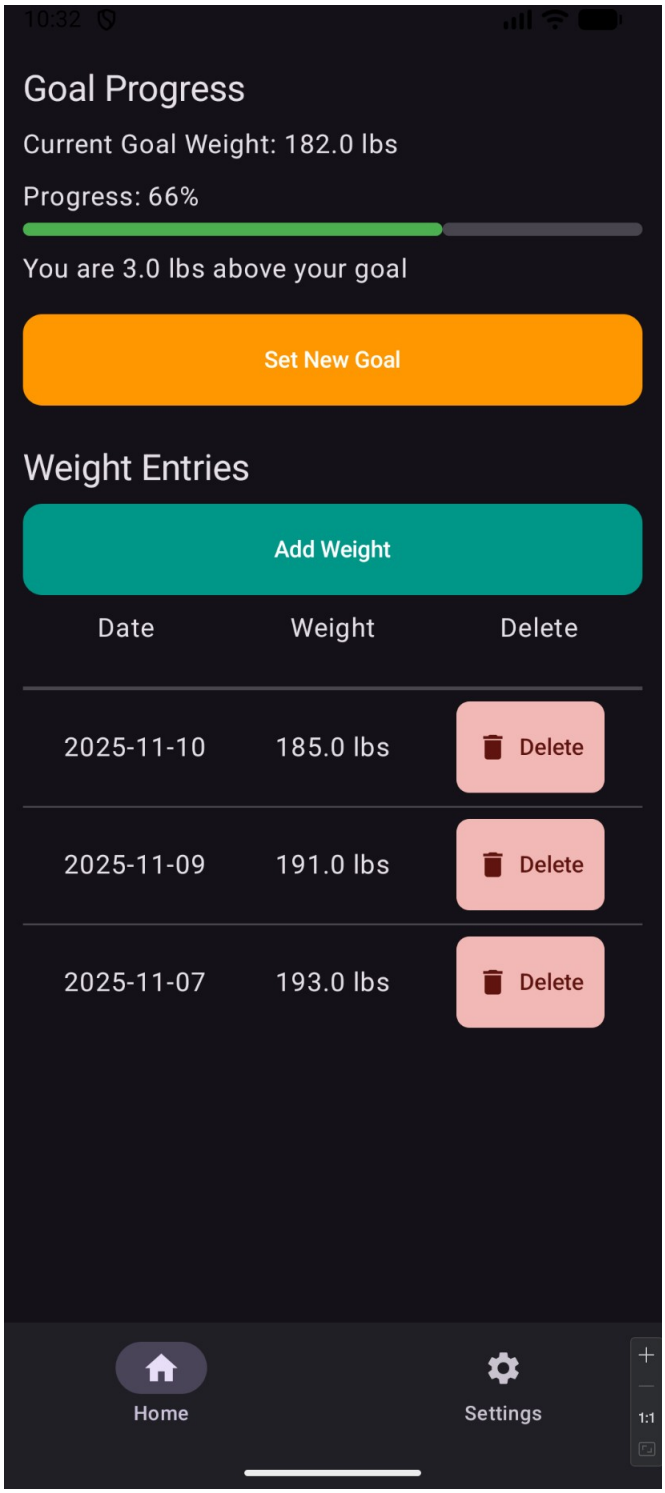
them to real projects. It also showcases my ability to take legacy applications and migrate them to a new framework/technology. Because doing a 1-1 on this solution wasn't an option. I needed to re-think from the bottom up how this needed to be designed. Designing the solution in a way that utilized commonMain and Jetpack Compose in KMM as much as possible, so that it minimized redundancy in my solution and the amount of platform-specific implementation. Also, the implementation of the date picker in iOS was especially challenging. I think it showcases my knowledge and skills of the KMM framework.

Overall, the artifact is improved because it now runs on iOS and Android in essentially the same unified code-base. I don't need to create an entirely new code-base in order to accommodate iOS users. And have expanded the number of users my application can support. I would also argue that the cleanliness of the code has improved because I have done a better job of creating modularization in my solution. Thanks to composables in Material 3, it allows me to create them almost like widgets. I have also improved the local database solution to SQLDelight, that it followed the same structure as Jetpack Compose. In that, I was able to build it in commonMain and use the same unified code-base for iOS or Android. Without this, I would have had to design two different local storage solutions. Which would have made it significantly more complex when moving towards remote storage solutions in enhancement three. I also improved the UI for all of the components to be more user friendly on mobile and implemented a dark mode theme toggle in the settings screen. Additionally, I added confirmation modals for actions like overwriting weight or goal entries, as well, as deleting weight entries. See the screen screen-shots below with original on the left and enhanced on the right.

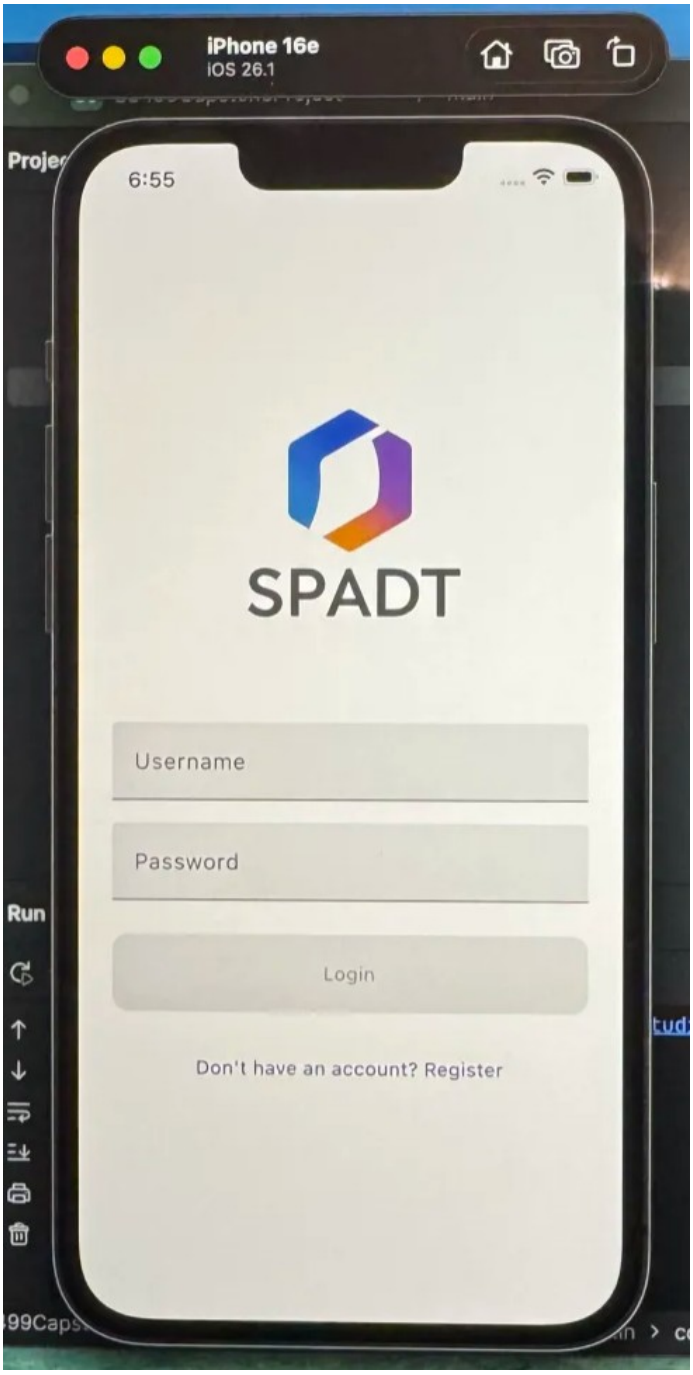

username
password
LOGIN
REGISTER

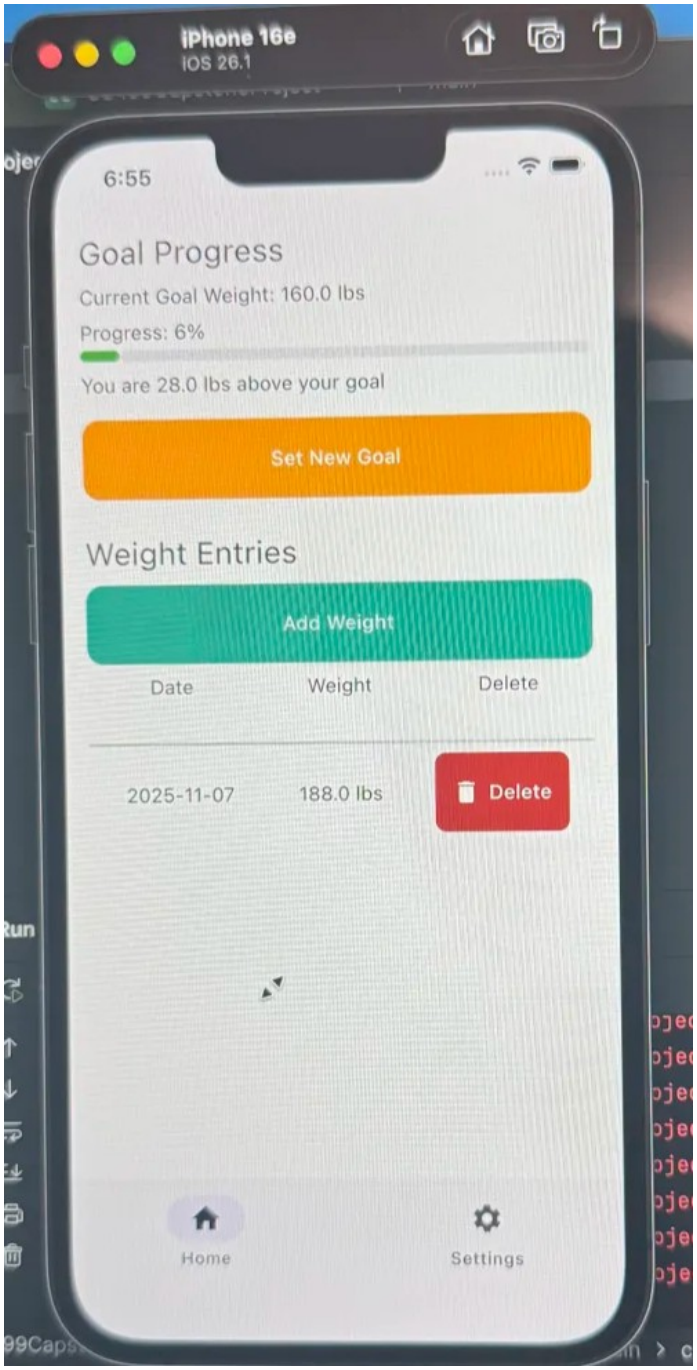

10:31
SPADT
Username
test
Password
Login
Don't have an account? Register
1:1


Goal: 140 lbs
You are 12.0 lbs above your goal
Progress: 20.0%
SET NEW GOAL
Weight Entries
Date
Weight
Action
2025-10-22
152
DELETE
2025-10-21
165
DELETE
ADD WEIGHT
Database
SMS


Goal Progress
Current Goal Weight: 182.0 lbs
Progress: 66%
You are 3.0 lbs above your goal
Set New Goal
Weight Entries
Add Weight
Date
Weight
Delete
2025-11-10
185.0 lbs
Delete
2025-11-09
191.0 lbs
Delete
2025-11-07
193.0 lbs
Delete
Home
Settings

Here is the solution running in light mode on iOS:



The outcomes I outlined that aligned with my enhancement were outcomes 3, 4, and 5. I think because of the nature of this enhancement 3 and 4 were automatically met by just completing the enhancement. Using modern tool-sets like KMM, Jetpack Compose, and SQLDelight are well-founded and delivering a great deal of value for accomplishing this enhancement. And while migrating I needed to really consider how the new architecture was going to interact with my existing legacy solution. Instead of doing a 1-1 conversion of the code-base, I thought deeply about what was the best way to approach it instead. As far as outcome 5 goes, outside of ensuring that the solution dependencies were updated there wasn't much opportunity to think about security. I mean, I performed testing to ensure that the user was only seeing their own data and that the queries to the DB were only returning their data. But all of the data is currently stored locally on their device. This outcome is going to be thought about more in enhancement 3.

I learned a great deal during this enhancement. I learned about Kotlin Multiplatform mobile, Jetpack Compose, and SQLDelight. It was an awkward thought process at first because I had to shift my thinking and understanding of how KMM and Compose worked. I wasn't just designing the solution for own device platform I was designing it for two. So I wanted to create as many modules as I could that would work on both device platforms. Outside of that difficulty, I also had a great deal of challenge when setting the project up. I bought a Macbook to be able to test the solution in an iOS simulator (I also needed to if I wanted to use it on my phone). Setting up a github repo and pulling it down to Macbook led me down this huge rabbit hole trying to fix the issues. I ultimately re-created the project on the Macbook and copied the code over, then used the built-in version control solution in Android Studio. I spent hours diagnosing issues in terminal though and it was frustrating to say the least.

In conclusion, I think this was a great first step. In addition to my planned enhancement of migration, I also included a dark mode theme toggle in the settings screen. The migration went well, and I was able to test all of the features on both iOS and Android. I'm excited to continue my work on this artifact for enhancement 2. And I hope I have time to implement some sort of calorie tracking and BMI calculation inside of the application as well. I would be able to totally replacing my Google Sheets spreadsheet then.