CS-499 Capstone Module 5 Databases

Professor Conlan

Nick Burnette

**Briefly describe the artifact:**

--A weight loss app using Android Studio from CS 360 that stores weights over time.

**Justify the inclusion of the artifact in my ePortfolio, why did I select it, which components showcase my skills, how was it improved?**

--I chose migrating the current DB setup to Room for this Databases artifact because migrating the current SQLite setup shows an understanding of modernization and maintainability. I can formally move the current DB to a new Room schema while keeping the UI nearly the same, and Room will manage future inserts and updates.

**Did you meet the course outcomes you planned to meet in Module One?**

--Yes. Throughout Enhancement 3, I met the following course outcomes:

* Build Collaborative Environments: by migrating to Room I have opened accessibility for others to contribute. Accurate and well-placed comments also are utilized.
* Create Computing Solutions: by migrating to Room I am keeping future versions up to date with a modernized and maintained database.
* Apply Standard Tools and Techniques: by building according to documentation I am creating a familiar codebase that other’s can easily pickup and understand. The codebase is tested and more secure.
* Adopt the Security Mindset: building with a well-established database like Room is a way of managing vulnerabilities and protecting data.

**Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating and improving it? What challenges did you face?**

--While enhancing this artifact I learned how integrated the tools that call on the database are, creating an important need to plan and be organized. I faced several challenges while integrating SQLite to Room, mostly in the number of calls made to the old database. Anytime a project can be built from the beginning with the key tools in mind will save the developer time, but part of this exercise was to migrate a database. Much of the challenges faced are listed below, but one challenge in particular was the migration of data. I did migrate the data but the old and the new had a WEIGHT/date swapped with date/WEIGHT. I wasn’t able to remigrate the old data without creating another migration and a new key version 5, which I wanted to avoid for cleanliness of code.

===== Journal Portion Below =====

(The rest of this document was performed informally during development of the third artifact.)

Objectives: Migrate the current SQLiteOpenHelper setup to Room, replace WeightDatabaseHelper calls with a RoomDatabase + @Dao.

Notes: Adding Room dependencies involves Gradle

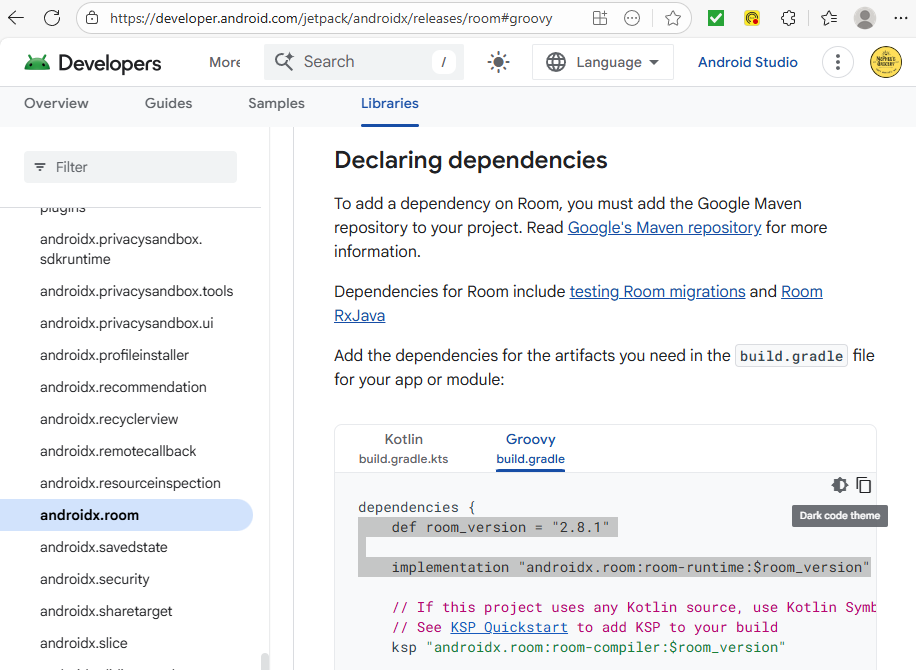
First step is to ensure my files are up to date with Git 🡪 pull from origin main, and then create a new branch to work safely on this Artifact 3 Enhancement. I named it db-migration-room:

A screenshot of a computer

AI-generated content may be incorrect.



I can’t connect things until I have something to connect them to, so I will start by creating Room. As a part of the Android Library, there are some Gradle dependencies to add so I start there. I visit the Developers Library for official documentation and version numbers:



I work this into the existence build.gradle dependencies:

A screen shot of a computer program

AI-generated content may be incorrect.

I am going to create WeightEntryEntity.java to start building the table, matching the current DB table layout:

A screen shot of a computer

AI-generated content may be incorrect.

Next create the DAO which will mirror the queries already used in WeightDatabaseHelper. Again I refer to the Android Developer’s page for guidance:

A screenshot of a computer

AI-generated content may be incorrect.

I used the other database format to construct the query, insert, and delete in my new WeightDao.java:

A screenshot of a computer program

AI-generated content may be incorrect.

I’m about to create the Room database, and I look at my old SQLiteOpenHelper version (under WeightDatabaseHelper.java) I see some unused imports to delete and I also notice DATABASE\_VERSION 3 is used, so I will increment the new AppDatabase.java with a DATABASE\_VERSION 4:

A screenshot of a computer program

AI-generated content may be incorrect.

I didn’t use ‘abstract’ very often but the IDE warned of errors until I added it, which resolved them. On research, I am reminded that ‘abstract’ is for defining methods that should be implemented by subclasses, but most importantly it made the IDE errors disappear.

To get the AppDatabase class to work I responsibly used ChatGPT to outline a migration, I consider it appropriate since I know what I want to do but it’s a complicated matter of correct syntax and layout. While researching the matter outside of ChatGPT, even in-browser AI will suggest code, making it difficult to avoid at times. (I do not enter school code into chat). For this section I am using it as a tool after applying my own critical thinking and exhausting the online references. I described my setup and described my needs. Obviously it has to be adapted by me to this project anyway.

A screenshot of a chat

AI-generated content may be incorrect.

I also commented the section in question for clarity:

A screen shot of a computer program

AI-generated content may be incorrect.

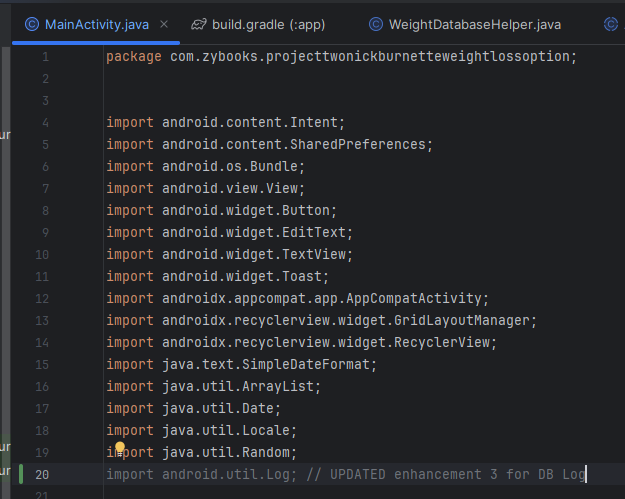
I have ensured the @Entity from WeightEntryEntity.java matches schema name ‘weights’ from the old db. I am ready to run my first test and see what I’m missing. I am going to keep the old app because I’m specifically testing the migration upgrade. I pull up the unchanged version of the app and then I refresh to add the changes, keeping all the old db info in place. This first shot is before the merge :

A screenshot of a cell phone

AI-generated content may be incorrect.A screenshot of a cell phone

AI-generated content may be incorrect.

The second photo shows a new db entrie of 67 lbs, but I haven’t verified the new database is what pulled it. I’m going to run a check through LogCat, starting with a new import in MainActivity.java:



Before getting far enough to review, I realize I am not calling the new Room db in MainActivity.java. I look at private void loadWeightData and I comment out the old deprecated calls and carefully insert the new listForUi to access Room using Dao:

A screen shot of a computer program

AI-generated content may be incorrect.

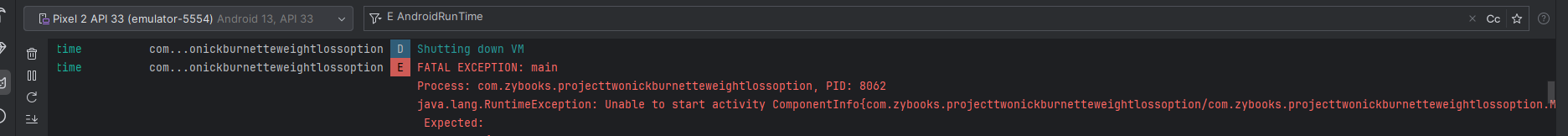
Getting errors in build.gradle so I found a typo with “” in the wrong place, got a rebuild to go through, but with these warnings:

A screen shot of a computer program

AI-generated content may be incorrect.

On research I discovered these warnings are letting me know that Room can export my db schema to JSON files on each build but if I don’t tell it where to export it warns me, this seems safe to ignore. I perform a sync when the IDE reminds me and am testing now:

Although the gradle build is successful, the app is crashing on startup



The error offscreen mentions a migration issue:



I have a Room schema mismatch during migration. I look at the table rebuild and work on creating a migration that builds the table with NOT NULL to preserve existing data. I realize I had been in progress already with a FIXME, so first photo is before:

A screen shot of a computer

AI-generated content may be incorrect.

And after:

A screenshot of a computer program

AI-generated content may be incorrect.

This should address the recent error, let’s hunt for the next error. Testing now and discovered another typo in ‘weiights’:





New error shows a migration issue where Room @Entity expects ‘weight’ but my DB in SQLite has ‘WEIGHTS’:

A screen shot of a computer code

AI-generated content may be incorrect.



I go to WeightEntryEntity.java and add the @ColumnInfo line to specify the name:

A computer screen with text and images

AI-generated content may be incorrect.

Going to build 🡪 rebuild project since I’m playing with db column names and will retest now (works at first, crashes on db entry, continues to crash on start again). Logcat error shows:



Which means the code path is still using the old SQLiteOpenHelper (version 3). The device has a Room upgraded DB at version 4, so I need to fix using the old helper anywhere, and make the on-device version consistent. I found some in TrendAnalyzer.java, here’s an example. For the CAPSTONE I’m leaving the old code commented out to identify changes made but for production I would remove the old code after successfully deprecated:

A computer screen shot of a program

AI-generated content may be incorrect.

Many of the db calls to fix are from TrendAnalyzer.java which was built for the second artifact. It of course is simpler not to have to rewire a db setup, but at least it shows the ability to do so.

After all the proper reference fixes, the next crash comes from dbHelper is now null, as I stopped using WeightDatabaseHelper, but something is still calling it, turns out to be addNewWeight() in MainActivity.java. I write a new code that calls db.weightDao().insert instead and we have a stable build that is creating new entries.

I have old entries from previous db with new entries from Room db, and I see them populating in the homepage recycler view at the bottom of the screen (New DB is dated 2025-10-1):

A screenshot of a cell phone

AI-generated content may be incorrect.

My goal is to have the user’s old data migrate seamlessly into the new db schema. In Android Studio I go to View 🡪 Tool Windows 🡪 Database Inspector and confirm ‘weights’ has the old db data, which it does (anything dated Sept is the old db):

A screenshot of a computer

AI-generated content may be incorrect.

Looking at the columns and entries, the old DB entries are mixed up. Date column has weight entries from old db, and WEIGHT column has dates. That would certainly boggle TrendAnalyzer.java.

To fix this I go to WeightDatabaseHelper.java and swap the table entry

A screen shot of a computer

AI-generated content may be incorrect.

This would have worked during initial migration, and I cannot replicate to redo. For a future migration I should have confirmed the table order. To preserve old data I would have to create another migration table and use a version 5 key, I don’t think that’s needed for this enhancement.

Now I’m getting a crash when attempting to ‘delete last entry’ the ‘d’ in CRUD. This sounds like the final holdout calling the old db, so I go to WeightDatabaseHelper.java and I find :

A screen shot of a computer

AI-generated content may be incorrect.

Which is clearly calling SQLiteDatabase. I built this in Dao and Room but need to update that in MainActivity.java. Here is the old call:

A screenshot of a computer

AI-generated content may be incorrect.

Here is the new call:

A computer screen shot of a computer code

AI-generated content may be incorrect.

On testing, everything appears to be in order, I checked every button and nothing is crashing. This should conclude enhancement 3!

A screenshot of a phone

AI-generated content may be incorrect.