Milestone Four

Russell Reckhow

Southern New Hampshire University

CS499: Computer Science Capstone

Professor Joseph Martinez

Narrative

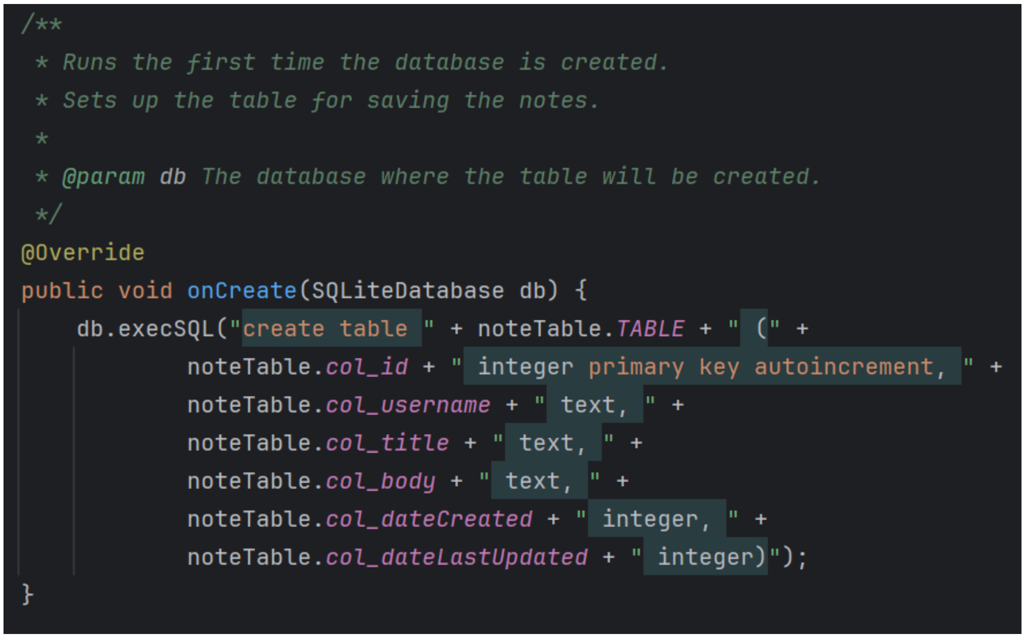
The artifact I used was my Weight Tracker app from CS360. It was created around June of this year. For this enhancement, I added a notes database so users can create, read, update and delete notes. This feature lets users save notes in the app, so they can keep track of extra details along with their weight.

I added this because it shows I can expand existing software by adding a database. It also shows both frontend and backend work. On the frontend, I updated the interface so that users can add a new note, see all their notes, and edit or delete them. If there are no notes saved, the screen will show a message, so the user knows the list is empty. On the backend, I built the database using SQLite and set up the tables and queries for notes.

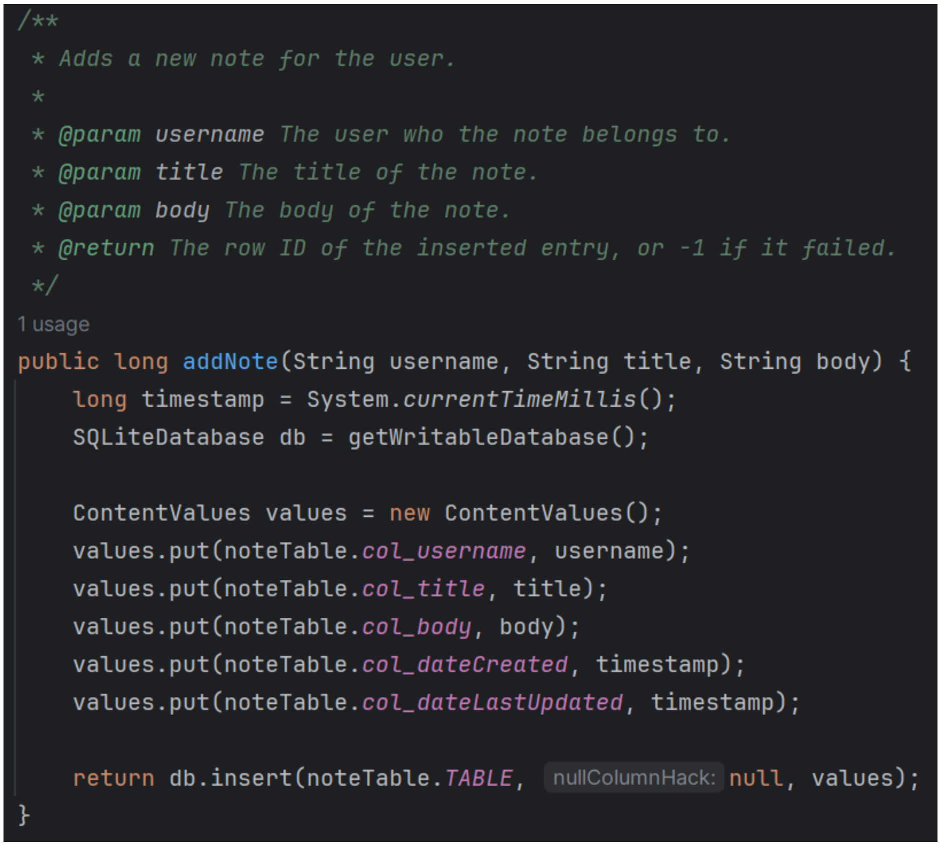
This enhancement met the outcomes I wanted so far. It shows I can design and use solutions by working with a database to store and manage user data. It also shows I can use tools and techniques like Android Studio, SQLite, and Java code for database operations. I showed communication by writing about it in my ePortfolio and GitHub and talking about it in my code review video. It also connects to security because notes are tied to usernames, the database safely handles user input, and the app asks for confirmation before deleting, which helps protect user data.

I learned how to connect my app to a database and handle CRUD operations in Android. I also learned how to manage cases where no notes were shown and how to update the UI when changes are made. There were a few challenges. The first was when pressing to add a note and saving it, nothing would show. I forgot to add LayoutManager and once added I was able to see the list of notes. The second issue I had was the top note being pushed slightly underneath the action bar. So, I added some padding to the top to fix it. Third issue I had was the bottom of the list wasn’t lined up with the nav bar, so I had to add padding at the bottom to fix it. Solving these problems helped me get better at troubleshooting and finding different ways to fix things.

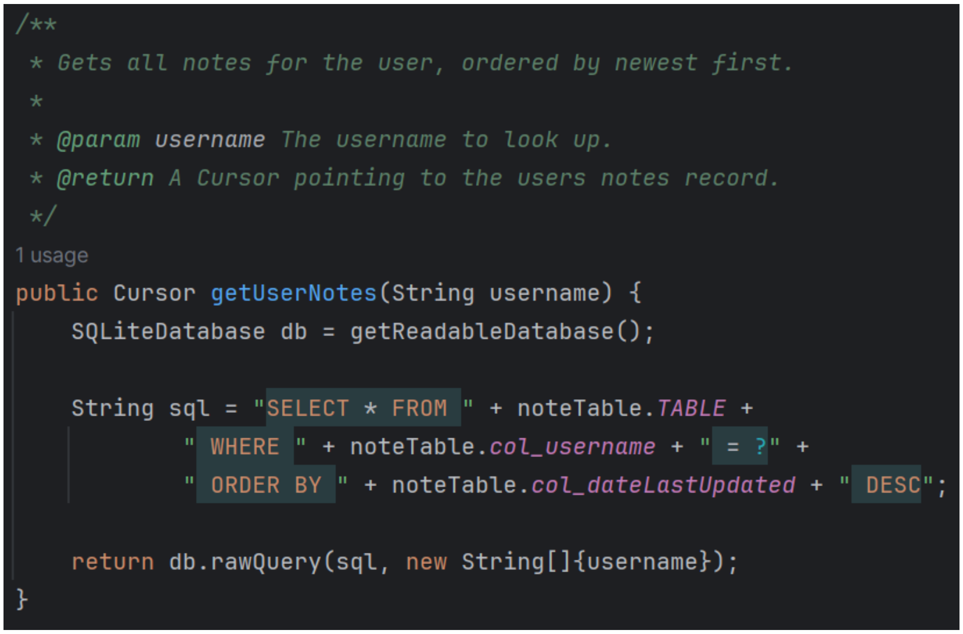
NoteDatabase Code:



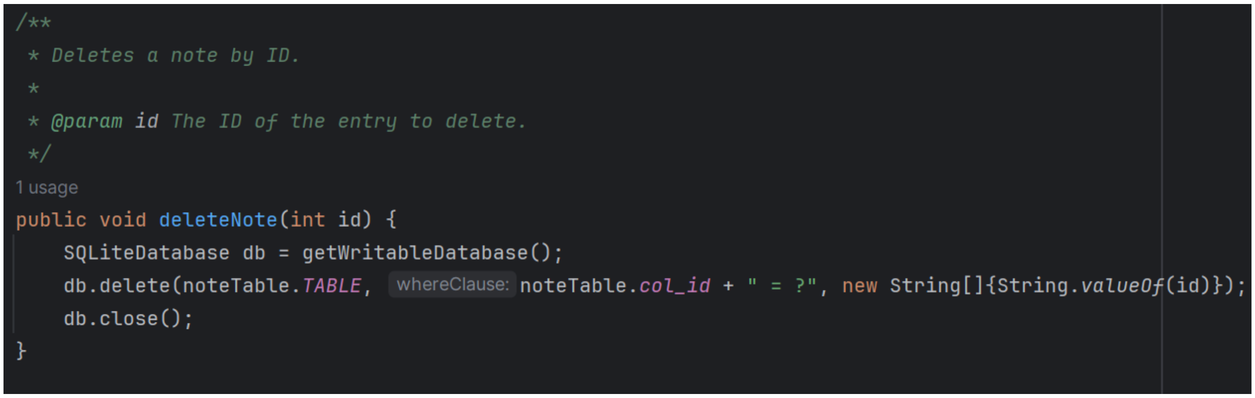
```java
/**
 * Runs the first time the database is created.
 * Sets up the table for saving the notes.
 *
 * @param db The database where the table will be created.
 */
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table " + noteTable.TABLE + " (" +
            noteTable.col_id + " integer primary key autoincrement, " +
            noteTable.col_username + " text, " +
            noteTable.col_title + " text, " +
            noteTable.col_body + " text, " +
            noteTable.col_dateCreated + " integer, " +
            noteTable.col_dateLastUpdated + " integer)");
}
```



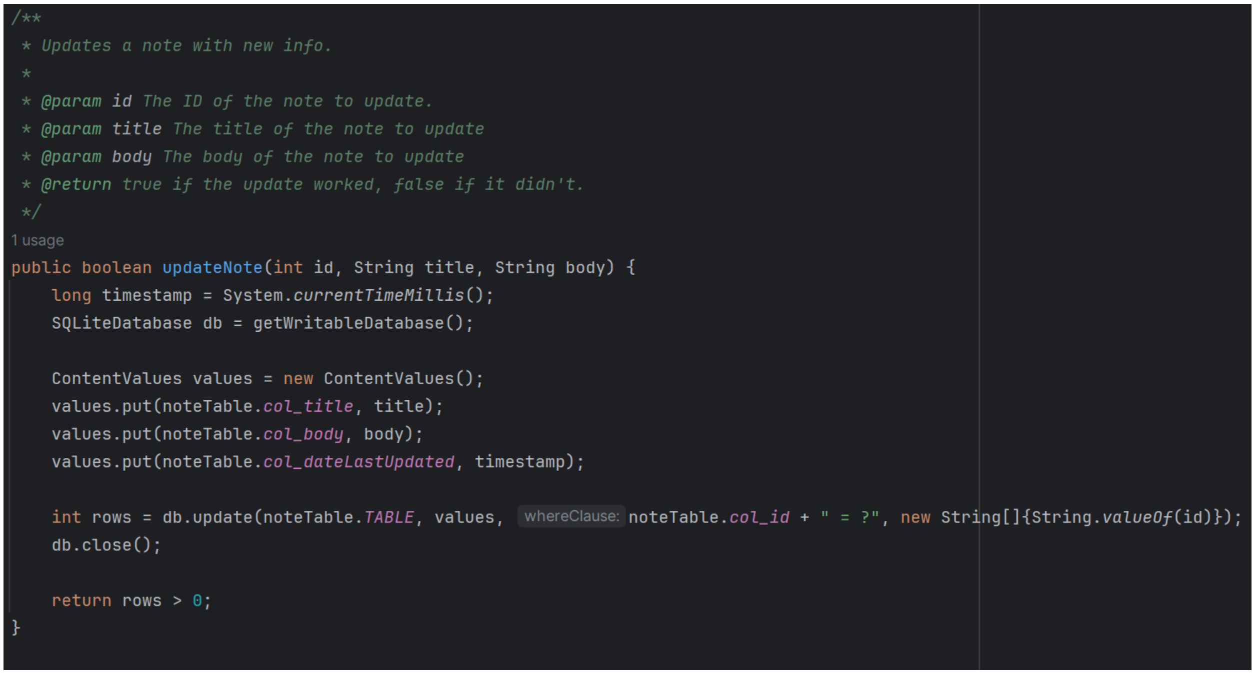
```java
/**
 * Adds a new note for the user.
 *
 * @param username The user who the note belongs to.
 * @param title The title of the note.
 * @param body The body of the note.
 * @return The row ID of the inserted entry, or -1 if it failed.
 */
1 usage
public long addNote(String username, String title, String body) {
    long timestamp = System.currentTimeMillis();
    SQLiteDatabase db = getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(noteTable.col_username, username);
    values.put(noteTable.col_title, title);
    values.put(noteTable.col_body, body);
    values.put(noteTable.col_dateCreated, timestamp);
    values.put(noteTable.col_dateLastUpdated, timestamp);

    return db.insert(noteTable.TABLE, nullColumnHack: null, values);
}
```



```
/**
 * Gets all notes for the user, ordered by newest first.
 *
 * @param username The username to look up.
 * @return A Cursor pointing to the users notes record.
 */
1 usage
public Cursor getUserNotes(String username) {
    SQLiteDatabase db = getReadableDatabase();

    String sql = "SELECT * FROM " + noteTable.TABLE +
            " WHERE " + noteTable.col_username + " = ?" +
            " ORDER BY " + noteTable.col_dateLastUpdated + " DESC";

    return db.rawQuery(sql, new String[]{username});
}
```

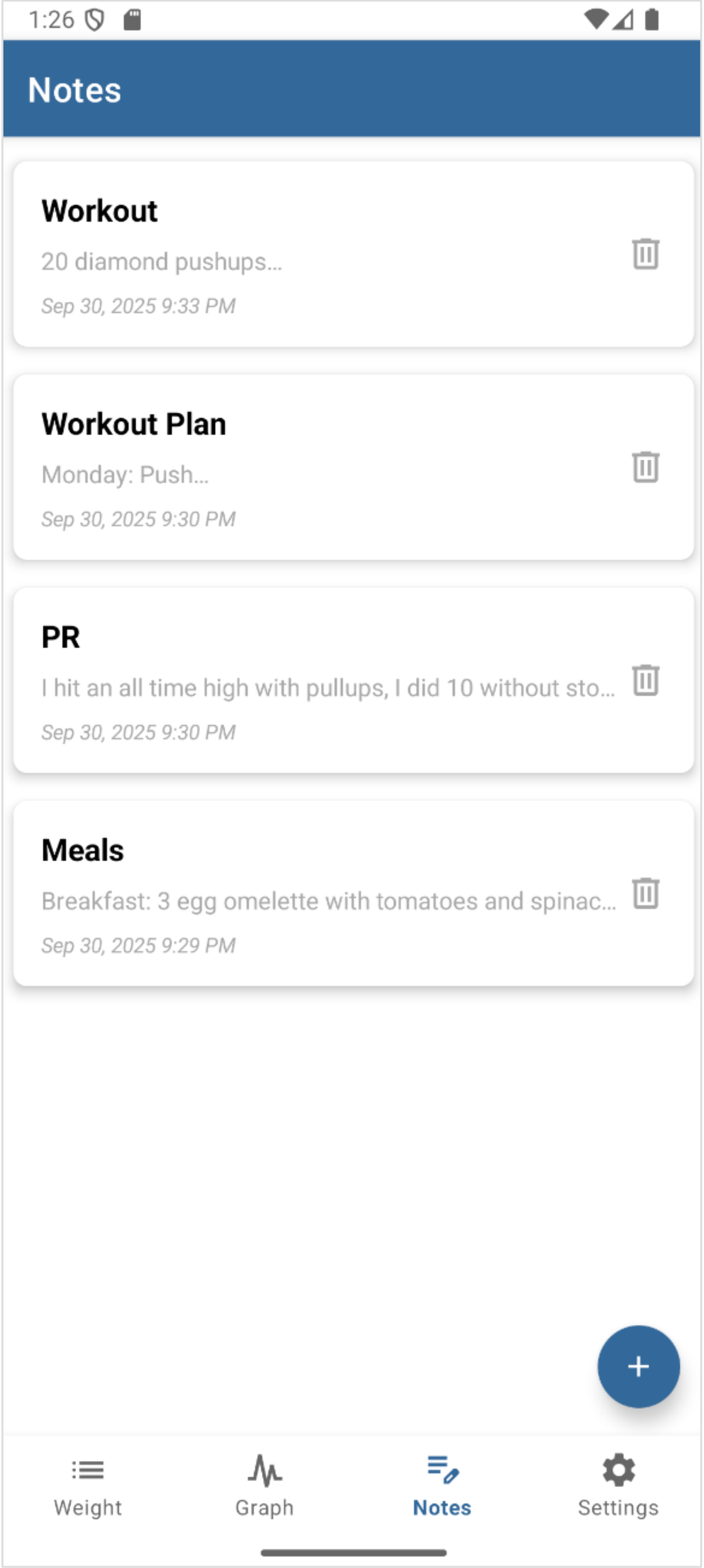


```
/**
 * Deletes a note by ID.
 *
 * @param id The ID of the entry to delete.
 */
1 usage
public void deleteNote(int id) {
    SQLiteDatabase db = getWritableDatabase();
    db.delete(noteTable.TABLE,  whereClause: noteTable.col_id + " = ?", new String[]{String.valueOf(id)});
    db.close();
}
```



```
/**
 * Updates a note with new info.
 *
 * @param id The ID of the note to update.
 * @param title The title of the note to update
 * @param body The body of the note to update
 * @return true if the update worked, false if it didn't.
 */
1 usage
public boolean updateNote(int id, String title, String body) {
    long timestamp = System.currentTimeMillis();
    SQLiteDatabase db = getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(noteTable.col_title, title);
    values.put(noteTable.col_body, body);
    values.put(noteTable.col_dateLastUpdated, timestamp);

    int rows = db.update(noteTable.TABLE, values,  whereClause: noteTable.col_id + " = ?", new String[]{String.valueOf(id)});
    db.close();

    return rows > 0;
}
```

Notes in App:



1:26

Notes

Workout

20 diamond pushups...

Sep 30, 2025 9:33 PM

Workout Plan

Monday: Push...

Sep 30, 2025 9:30 PM

PR

I hit an all time high with pullups, I did 10 without sto...

Sep 30, 2025 9:30 PM

Meals

Breakfast: 3 egg omelette with tomatoes and spinac...

Sep 30, 2025 9:29 PM