

CS-499: Computer Science Capstone  
5-2 Milestone Four: Enhancement Three: Databases Narrative  
Ariana Mikhak  
July 28, 2025

### **Artifact Description**

The artifact I selected for this enhancement is EventTrackerApp, an Android mobile application originally developed in CS-360. EventTrackerApp allows users to create, edit, and manage personal events with optional SMS notifications and offline functionality through a local SQLite database.

The original version of the app stored all events in a single table and performed only basic CRUD operations. This structure worked for small datasets but was limited in maintainability and efficiency. It lacked relational organization, optimized queries for summaries and filtering, and any way to export user data.

For this database-focused enhancement, I redesigned the data model and query logic to create a cleaner, more scalable, and more professional-grade database layer.

### **Justification for Inclusion**

EventTrackerApp is a strong artifact to demonstrate database enhancement because event-driven mobile apps rely on fast, reliable, and consistent data storage.

The original single-table design was simple but inflexible. Enhancing the database allowed me to:

- Normalize the schema into multiple tables to improve maintainability

- Use JOIN queries and a view to efficiently retrieve and filter events
- Provide CSV export functionality for reporting or external use

This enhancement shows my ability to implement a professional-quality database design and integrate it into a working mobile application.

### **Course Outcomes and Coverage**

This enhancement supports the following CS-499 outcomes:

1. Design and evaluate computing solutions using computer science practices and standards while managing trade-offs
  - I refactored the database schema to a normalized three-table design:
    - Users (stores credentials and contact info)
    - Categories (groups events with color-coding)
    - Events (linked to users and categories via foreign keys)
  - I implemented SQLite views and JOIN queries for efficient event retrieval and filtering.
  - The trade-off was between keeping the schema extremely simple versus introducing relational structure for maintainability and future scalability. I chose a straightforward normalization approach that avoids unnecessary complexity.
2. Demonstrate the use of well-founded techniques, skills, and tools in computing practices
  - Implemented CSV export to allow users to back up and share event data.

- Applied foreign keys to maintain data integrity between users, events, and categories.
- Leveraged SQLite's built-in indexing on primary and foreign keys for reasonable performance without manually creating additional indexes.

These changes demonstrate an understanding of practical mobile database design that balances simplicity, reliability, and extensibility.

### **Reflection on the Enhancement Process**

Enhancing the database required revisiting how EventTrackerApp stored and accessed data. I started by designing a normalized relational schema and implementing foreign keys to protect data integrity. This restructuring made it easier to query events by user and category and laid the groundwork for more advanced features in the future.

I then created JOIN-based queries and a view for streamlined event retrieval and summaries.

Finally, I implemented CSV export functionality, giving users the ability to back up and analyze their data outside the app.

This enhancement reinforced my understanding of database design, query formulation, and maintainability in mobile applications. While the implementation remained relatively straightforward, it brought EventTrackerApp much closer to a production-ready standard and demonstrates skills that will translate directly to professional development and my upcoming MCS studies.