

Foster Hare

CS-499

Professor Martinez

11/1/2025

Capstone Enhancement Plan

For my capstone enhancement, I will focus on improving my existing Android application, **EventBuddy**, and use it as the single artifact for my ePortfolio. This project provides a practical foundation to demonstrate growth in the three required areas, software design and engineering, algorithms and data structures, and databases, without needing to build a new application from scratch. By refactoring the codebase to use a clearer architecture, implementing efficient sorting and search algorithms, and improving the underlying SQLite database with indexes and safer queries, this approach allows me to highlight real development skills that align with my career goals.

1.) Software Design and Engineering

- Repository Layer
 - Add EventRepository that wraps SQLiteOpenHelper (or existing DAO).
 - Centralize input validation and error handling (e.g., null/empty titles, invalid dates).
 - Single place to log DB operations (duration, returned row count).
- Light MVVM wiring
 - Keep Activities/Fragments; add a minimal EventViewModel that calls EventRepository.
 - No UI overhaul—just redirect existing UI actions to the ViewModel/Repository.

- Unit/Instrumented Tests
 - 6–8 tests (JUnit + AndroidX Test or Robolectric): create, read, list, invalid input, and sort requests via the Repository.

2.) Algorithms and Data Structures

- Efficient sorting and Stable Multi-Filter
 - Implement `EventAlgorithms.sortEvents(List<Event>, SortKey)` using comparator (date/title).
 - Add a filter utility that narrows by optional fields (e.g., type/category) without extra DB hits when list is in memory.
- Binary Search on Sorted List

After initial load (sorted by title or date), implement `binarySearchByTitle` or `binarySearchByDate` to jump to matching entries quickly.
- Overlap Detection

Detect scheduling conflicts with a classic interval overlap check (sort by start, then scan).

3.) Databases

- Indexes
 - Add schema v2 migration to create indexes on `event_date` and `title`.
 - Ensure `PRAGMA foreign_keys=ON` if you have related tables.
- Parameterized Queries

Convert any raw string concatenation to SQLite parameter bindings to prevent injection and handle special characters safely.

- Transactional Safety

Wrap multi-row operations (if any) in a transaction for atomicity.

These enhancements allow me to meet all three CS-499 course outcomes while demonstrating real-world software engineering practices in a project I can continue expanding beyond graduation.