Suvam Kharel

CS 499 Capstone

31 March 2025

**Module Four Narrative (Milestone 3)**

The artifact I chose for this assignment was the Event Tracking App that was a project I worked on CS 360: Mobile Architecture and Programming course, which was a course I took last year. This app allows users to manage and track events, with features such as adding events, storing descriptions, viewing them in a list, and deleting events. I chose the project as I wanted to improve the overall design of the app and modernize the look of the app with better UX/UI and functionality requirements to meet the current standards.

I selected this artifact for inclusion in my ePortfolio because its improvement in functionality demonstrates my skills required in category two. This enhancement shows my ability to:

• Implement Sorting Functionality; I added a sorting feature using a Spinner and SQLite’s ORDER BY clause, allowing users to organize events either alphabetically by name or chronologically by event date, improving accessibility and clarity in event browsing.

• Integrate Flexible Search Bar; I implemented a search bar that filters events in real time based on event name or date using a linear search. The search bar resets the list automatically when cleared, enhancing usability and maintaining easy and quick navigation to main dashboard.

• Optimize Event List Management; I introduced a master event list (allEvents) alongside the displayed list (events) to ensure efficient and accurate filtering, sorting, and resetting operations without data loss or display errors.

• Apply Practical Algorithm Design; While I initially considered binary search for performance, I intentionally used linear search to support partial match functionality and real time user input which is a better fit for mobile app UX with moderate data volume and for much better UX.

Overall, the artifact was enhanced by addressing the app’s functionality by using efficient data handling techniques for sorting and implementing search mechanisms.

For this artifact, I had planned to make the following changes:

1. Implement sorting using SQLite’s ORDER BY to prioritize events based on the preference of the user (For example: sorting my date, sorting by name, importance etc.)

2. Implement Binary Search to implement faster event lookup based on user input.

3. Implement the search bar in the UI to support the search feature.

All the enhancements were made except for the Binary search as it was more practical to implement the linear search based on the type of artifact we are working with and considering user behavior. It is a better option considering the artifact is a mobile app and users might not always be accurate when searching up certain name and date of an event and it might frustrate the user if the app does not meet the required functionality.

A significant challenge I faced was ensuring that the filtered and sorted event data remained in sync across different operations. Without careful management of the underlying lists, the search bar and sorting spinner could conflict with each other or show inconsistent results. To solve this, I implemented a separate allEvents list to serve as the master data source, while the displayed events list was used for UI updates. This structure allowed for clean and reliable updates when users searched, sorted, or cleared input.

As per feedback, I have included some before-after screenshots representing the Event Tracking app’s improved functionality from the progress made in milestone 1 and 2:

Login Screen:

*Before:*  
A screenshot of a cell phone

AI-generated content may be incorrect.

*After:*

A screenshot of a computer

AI-generated content may be incorrect.

Event Grid & Search Functionality:

*Before:*

A screen shot of a cell phone

AI-generated content may be incorrect.

*After:*

A screenshot of a computer

AI-generated content may be incorrect.

Adding Event:

*Before:*  
A screen shot of a cell phone

AI-generated content may be incorrect.

*After:*

A screenshot of a computer

AI-generated content may be incorrect.

Send SMS Screen:

*Before:*

A screenshot of a cell phone

AI-generated content may be incorrect.

*After:*

A screen shot of a computer

AI-generated content may be incorrect.

Overall, this process strengthened my skills in Android development, database integration, UI responsiveness, and debugging, while reinforcing the importance of thoughtful design choices when implementing algorithms in user-facing applications.