

CS-499: Computer Science Capstone

4-2 Milestone Three: Enhancement Two: Algorithms and Data Structure Narrative

Ariana Mikhak

July 18, 2025

### **Artifact Description**

The artifact I selected is the same Android mobile application I used for the first enhancement—EventTrackerApp, developed originally in CS-360. The app allows users to create, manage, and schedule events, with optional SMS notifications and offline functionality via a local SQLite database.

The original version used simple data structures like ArrayList for storing and iterating through events. While functional, that design wasn't scalable and resulted in inefficient linear-time lookups and scheduling logic. For this enhancement, I focused on optimizing that logic by introducing more advanced data structures to improve performance, usability, and responsiveness. I also continued to refine the generative AI title suggestion feature introduced in the first enhancement.

### **Justification for Inclusion**

This artifact remains a strong candidate for enhancement because its functionality centers around time-based event scheduling, a problem that benefits significantly from efficient data structure choices. The enhancements I made demonstrate my ability to analyze inefficiencies in a real application and apply algorithmic reasoning to improve performance.

I replaced the event storage logic with a PriorityQueue to manage time-sensitive scheduling and a HashMap to allow for constant-time lookups by event name. I also implemented conflict detection methods to alert users when events overlap. These additions reflect a deeper level of technical refinement and align with industry best practices for scalable, responsive mobile applications.

## **Course Outcomes and Coverage**

This enhancement directly supports the following CS-499 outcomes:

- **Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.**

The transition from linear to logarithmic and constant-time operations demonstrates deliberate performance trade-offs and technical analysis in selecting the appropriate data structure for each use case.

- **Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.**

By improving the app's performance and reliability under larger event loads, this enhancement delivers meaningful improvements that would matter in a production-grade mobile app.

## **Reflection on the Enhancement Process**

The process of replacing list-based logic with a PriorityQueue and HashMap required a deeper understanding of how data flow interacts with user interfaces in Android. I refactored the event

loading process to populate these structures and ensured synchronization between the UI's LiveData, the backend SQLite database, and these new internal caches.

Implementing conflict detection based on event dates introduced additional logic for filtering and comparing overlapping entries, which added both technical challenge and real-world usability improvements.

This enhancement helped reinforce the value of algorithmic thinking in day-to-day software development. I now better understand how to evaluate bottlenecks, select the right data structures, and apply them in a way that improves both performance and maintainability.

Looking ahead to graduate study in computer science at the University of Illinois Urbana-Champaign, this work serves as a bridge between my undergraduate development experience and the more advanced algorithmic analysis I'll be doing at the master's level.