Angel M. Rivera Moreira

CS 499

3-2 Milestone

July 20, 2025

The artifact selected for enhancement is the *Minimal Calendar App*, originally developed for the CS 360: Mobile Architecture and Programming course during the Spring 2025 term. This Android-based application enables users to manage daily events by storing them locally using SQLite. The original implementation consisted of several core components, including LoginActivity, CalendarActivity, and a helper class for database operations. These components worked together to provide basic login functionality, event creation, and UI rendering. The application was initially designed to be functional yet simple, featuring straightforward navigation and a static user interface.

Justification for Inclusion

This artifact demonstrates core software engineering principles, including modularity and reusability. It was selected for the ePortfolio because it reflects backend integration through full CRUD operations, UI-driven logic, and structured Android activity workflows. Enhancements performed include:

- Refactoring shared logic for maintainability.
- Implementing error handling via try-catch blocks.
- Improving input validation and user feedback.
- Preparing for RecyclerView integration for scalable rendering.
- Replacing plaintext password storage with SHA-256 hashing (SecurityUtils.java).
- Centralizing schema constants in a config file (DbConfig.java).

These enhancements align with secure, maintainable design and reflect readiness for production-level engineering.

Alignment With Course Outcomes

Enhancements support the following outcomes:

- Outcome 3: Applied algorithmic design to support future event grouping logic via Map<String, List<Event>>.
- Outcome 4: Demonstrated use of effective design techniques, including centralized configuration and abstraction.
- Outcome 5: Developed a security mindset by eliminating plaintext password storage and introducing hashing.

No changes were made to the original outcome plan from Module One.

Reflection

This enhancement process revealed the long-term cost of technical debt and the importance of upfront modular design. Rather than overhauling the architecture, I chose incremental upgrades aligned with the assignment scope. Creating SecurityUtils.java and DbConfig.java improved code clarity and security, reinforcing best practices that I aim to apply professionally.