Software Design and Engineering - Narrative

Artifact: Event Tracking Android App

Diana Galvez Mendez

The artifact I selected for enhancement is my Event Tracking Android App, originally developed for my CS360 Mobile Architecture and Programming course. The app lets users log in, view upcoming events, and manage event details. It was built in Java using Android Studio, and the original version functioned, but it had a basic layout, minimal validation, and lacked a professional, maintainable structure.

I chose this artifact for my ePortfolio because it represents my growth as a developer, taking something that “works” and elevating it into something that meets modern software design standards.

I focused on this project because the original code had everything jammed into a few files, making it harder to read, maintain, or extend. My goal was to restructure the codebase, improve UI/UX, and implement better validation so the app feels more professional and user-friendly. This enhancement shows I can take an existing codebase and refactor it to meet real-world industry expectations.

To improve the app, I implemented the following changes:

1. Refactored MainActivity.java
   * Broke down the login logic into smaller, clearly named methods (initializeUI() and setupLoginButton()), improving readability and maintainability.
   * Created a helper class (LoginHelper.java) to handle credential validation, separating concerns and reducing code duplication.
2. Upgraded the Login UI
   * Replaced basic EditText elements with TextInputLayout and TextInputEditText from Material Design for a modern look and better user experience.
3. Added Real-Time Validation
   * Implemented a custom TextWatcherAdapter to give instant feedback if a user leaves fields empty, improving error prevention.
4. Improved Event Display Structure
   * Created a dedicated EventAdapter for the RecyclerView on the dashboard, keeping UI logic separate from the data layer.
5. Code Documentation & Readability
   * Added comments, consistent formatting, and logical grouping to make the code more approachable for future developers.

These enhancements show my ability to apply design principles like separation of concerns, modularity, and maintainability in a real project. They also demonstrate my awareness of UI/UX best practices by adopting Material Design standards and adding user-friendly validation.

Before these changes, the code worked but it wasn't maintainable.

Good software design isn’t just about getting the app to run, it’s about making it easy to maintain, intuitive to use, and adaptable for the future.