Milestone Two Narrative – Software Design and Engineering Enhancement.

Artifact Description

The artifact selected for this enhancement is the Weight Tracker Android application, which was originally created during CS 360: Mobile Architecture and Programming. The application is designed to help users track their weight over time, set a personal goal weight, and manage their progress using a local SQLite database. The app includes multiple activities, such as user login, goal weight input, and weight monitoring, and was developed using Android Studio, Java, XML layouts, and SQLite.

Justification for Inclusion in ePortfolio

I selected the Weight Tracker application for my ePortfolio because it represents a realistic, user-facing software system that integrates multiple aspects of software design and engineering, including user input handling, persistent data storage, and activity-based application flow. This artifact demonstrates my ability to design and enhance a functional mobile application while applying professional software engineering practices.

For this milestone, I enhanced the application by focusing on robust input validation and error handling, particularly within MainActivity, GoalWeightActivity, and DBHelper. The enhancements improved the quality and reliability of the software by ensuring that invalid, unsafe, or unrealistic user inputs are handled gracefully at both the user interface and database layers. Specifically, I implemented checks for empty inputs, invalid number formats, unrealistic goal values, and null user states. These improvements strengthen the application's reliability, data integrity, and user experience while demonstrating defensive programming principles.

Course Outcomes Alignment

The enhancements completed in this milestone allowed me to meet the course outcomes I planned to address in Module One. In particular, this work demonstrates progress toward:

- **Designing and evaluating computing solutions** by applying algorithmic logic and validation rules to solve real-world input handling problems while managing design trade-offs (Outcome 3).
- **Using well-founded tools and techniques** by implementing structured validation, layered error handling, and clean separation of concerns across activities and the database helper class (Outcome 4).
- **Developing a security mindset** by anticipating invalid or malicious input and preventing unsafe data from being stored or processed, even if UI-level validation is bypassed (Outcome 5).

At this stage, I do not need to revise my outcome coverage plans, as the enhancements align well with my original goals. Future milestones will allow me to build upon this foundation by further addressing algorithms, data structures, and database optimizations.

Reflection on the Enhancement Process

Through the process of enhancing this artifact, I learned the importance of implementing validation and error handling at multiple layers of an application, rather than relying solely on user interface checks. This experience reinforced the idea that robust software design requires anticipating failure cases and protecting the system from invalid data, even when the application appears to function correctly on the surface.

One challenge I faced was ensuring that enhancements improved the software without overcomplicating the code or introducing new bugs. By making small, intentional changes and testing frequently, I was able to enhance the application’s reliability while keeping the design clean and maintainable. Overall, this process helped me think more like a software engineer by emphasizing quality, security, and long-term maintainability rather than simply making the application “work.”