Holly Renfrew
Due Date: 11/16/25

Milestone Two Narrative
Software Design and Engineering Enhancement

The artifact I selected for this milestone is my Android Weight Tracker Application, originally created in CS-360: Mobile Architecture and Programming. The project was first developed in 2024 as a mobile app that allows users to track their weight, view progress charts, set goals, and receive SMS notifications when they meet those goals. The original version included multiple screens, an SQLite database, password hashing, and basic user authentication. However, the early version lacked several important software design elements such as secure data storage, modular structure, and defensive coding practices.

Justification for Inclusion & Improvements Made

I selected this artifact for the Software Design and Engineering category because it is the most complete example of my ability to build a functioning application from the ground up. The project includes UI design, database interactions, authentication logic, chart generation, and structured Android components, making it a strong foundation for demonstrating improved design, security, and maintainability.

For this enhancement, I implemented several major upgrades that significantly improved both the design and engineering quality of the app:

1. Migration to EncryptedSharedPreferences

   The original version stored user IDs and “remember me” flags in plain SharedPreferences. I replaced this with EncryptedSharedPreferences backed by Android’s MasterKey, ensuring that all saved session data is encrypted at rest.

   Skill demonstrated: secure data handling and proper use of modern Android cryptography APIs.

2. Login Lockout System (Brute-Force Protection)

   I added new database columns, failed_attempts and locked_until, and implemented logic to lock user accounts after repeated failed login attempts.

Skill demonstrated: identifying and mitigating authentication vulnerabilities.

3. Stronger Password Validation

   The original app only required one uppercase letter and one digit. I expanded this to require uppercase, lowercase, numeric, and symbolic characters.

   Skill demonstrated: defensive programming and secure input validation.

4. Threading Improvements and Performance Enhancements

   Database interactions were originally performed on the main UI thread. I added reviewer notes, restructured code to make threading updates possible, and prepared the application for future migration to ExecutorService or coroutines.

   Skill demonstrated: performance-aware software design.

5. Modular Code and Architectural Cleanup

   I refactored logic that was previously scattered across UI files back into DatabaseHelper, cleaned up naming conventions, normalized user input, and improved internal consistency across files.

   Skill demonstrated: modularity, maintainability, and clean software architecture.

These enhancements significantly improved the quality of the application, transforming it into a more production-ready example of mobile software engineering.

Course Outcomes Met

In my Module One plan, I intended to demonstrate strong progress toward the following outcome:

> **Outcome 4: Software Design and Engineering**
> "Use well-founded and innovative techniques, skills, and tools in computing practices to implement solutions that deliver value and accomplish industry-specific goals."

I fully met this planned outcome. The enhancements I implemented required restructuring the architecture of the application, securing sensitive data, mitigating vulnerabilities, and improving the robustness of user input handling. These are all core aspects of professional software engineering.

Additionally, I made progress toward:

> **Outcome 5: Security Mindset**
> “Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.”

I strengthened authentication, protected stored data, improved validation, and added protection against brute-force attacks. These improvements reflect an understanding of how adversarial behavior can exploit weak designs.

> **Outcome 2: Professional Communication**
> “Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.”

Throughout the enhancement process, I added in-line documentation, clarified misleading code segments, and improved naming consistency. Writing this narrative and documenting the enhancements also reinforces this outcome.

I do not need to update my original outcome-coverage plan, because the completed enhancements align exactly with what I set out to achieve.

Reflection on the Enhancement Process

Working through the enhancements for this milestone gave me the chance to examine the original project with a much more experienced perspective. When I created this app in CS-360, I was focused on getting things working. Returning to it now, I could clearly see issues related to maintainability, input validation, thread management, and secure storage that I would not have fully understood earlier in my program.

Some of the key lessons I learned through the enhancement process include:

- How critical secure storage is in mobile development:

  Transitioning to encrypted preferences wasn’t just about adding a library. It required understanding Android’s key management and modern security practices.

- The importance of anticipating misuse and attacks:

  Implementing the lockout system showed me how even simple applications can become vulnerable without proper defensive design.

- Modular design makes long-term maintenance possible:

  Pulling logic out of UI activities and consolidating it in helper classes gave the project a cleaner structure and made enhancements much easier.

- Small inconsistencies lead to bigger problems later:

  Fixing naming, validation, and data normalization issues helped prevent duplicate records and unpredictable behavior.

Challenges I faced

- Updating the database schema safely while preserving data.
- Adjusting UI code to work with encrypted preferences.
- Ensuring all dependent files reflected renamed variables or new resources.
- Keeping the app functional while refactoring significant logic into helper classes.

Despite the challenges, the process improved my confidence in modifying existing codebases, identifying weaknesses, and implementing higher-quality solutions.

Conclusion

Enhancing this Weight Tracker application allowed me to demonstrate meaningful progress in software design, secure coding practices, and professional application architecture. The improvements I implemented (encrypted data storage, authentication protections, stronger validation, and refactored structure) elevated the project to a standard that better reflects my skills as an emerging software engineer.

This artifact now showcases my ability to revisit, analyze, and meaningfully improve an existing software system. It represents not just where I began in the program, but the level of engineering thought and attention to security that I can apply now. Because of this, it is a strong addition to my ePortfolio and a clear demonstration of my growth in the Computer Science program.