**Module Two Milestone: Software Design/Engineering Narrative**

* **Briefly describe the artifact. What is it? When was it created?**

The artifact I selected for this module is an Android inventory management app called Stock Shark, designed for small businesses, warehouses, or individual users who need a simple and effective way to track stock levels. The app was originally developed during my CS 360: Mobile Architecture and Programming course, which I completed earlier this year.

* **Justify the inclusion of the artifact in your ePortfolio. Why did you select this item?** **What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

I chose this artifact to include in my ePortfolio for several reasons. First, I think it's the strongest showcase of my work thus far in the software design/engineering category. In its original version, I believe it included well-structured Java code, thoughtful use of Android’s Material Design principles, and a clear understanding of basic mobile UI/UX development. Indeed, in my opinion, it was the most complete demonstration of my skills – skills that I hope to continue developing and refining as I enhance it

Additionally, as someone who has a strong interest in UI/UX design, I think this artifact will be an important part of my portfolio as I pursue career opportunities. While I am still keeping my options open and would be happy to explore any realm of software development, the creativity and problem-solving involved in UI/UX design and app development is something that I find deeply engaging and rewarding, and this artifact is something that I hope demonstrates that I can deliver professional-quality mobile applications in the future.

While I haven’t had the chance to implement additional functionality and screens like intended, the original codebase has improved dramatically from its original state. In my enhancement outline, I planned a full overhaul of the application to modernize the code structure, improve usability, strengthen security, and extend the overall feature set, and I believe I’ve successfully achieved that goal. The codebase has been completely restructured and modularized to enhance readability, maintainability, and ensure reliable user input validation. While the list of improvements is extensive, some of the most notable updates include:

* Modularized Responsibilities: Separated concerns by organizing logic into focused classes and utility methods for cleaner, reusable code.
* Secured Database Operations: Improved SQLite queries to properly scope user data and prevent unauthorized access.
* Secure Coding Practices: Implemented password hashing, input validation, and basic SQL injection prevention to reinforce application security.
* Material Design Compliance: Adopted up-to-date Material Design components for a modern and intuitive user interface.
* Expanded XML Layouts: Added new pages and layouts to streamline navigation and support new features.
* Custom Drawable Resources: Created and added new icons and visual assets to enhance the app’s branding and usability.
* Robust Error Handling: Introduced consistent error messages, input checks, and fail-safes across user actions.
* Accessibility Improvements: Added content descriptions, real-time validation and better input focus handling.
* Enhanced User Experience: Delivered real-time input guidance, toasts, and visual feedback for smoother interactions.
* Comprehensive Documentation: Added JavaDoc comments to all core methods to clarify business logic, expected parameters, and error behavior.
* Android Best Practices: Externalized all string and dimension values to XML for localization and easier theme management.
* Debugging and Testing Workflow: Adopted a systematic approach to identifying, reproducing, and resolving bugs through emulator testing and logging.
* **Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

In my Module One plan, I ambitiously aimed to touch all five course outcomes. In retrospect, that was optimistic; however, while I don’t believe this single enhancement demonstrates complete competency in every category, I do think it made strong progress toward developing a security mindset and improving professional communication.

Wanting to ensure the comprehensive security of the app's user, I prioritized implementing proper security measures to ensure that user data remains safe and secure. This included things like password hashing, input validation, SQL injection prevention, error handling, user session management, and various defensive programming practices. While still not perfect, I'm confident that the security measures I implemented will be an important demonstration of my security mindset.

As for professional communication, while in-person communication skills are something I continue to develop, I feel like the comprehensive JavaDoc comments and clear code documentation I added are indicative of my ability to communicate technical concepts effectively through code. Admittedly, my previous app contained very little documentation or explanatory comments.; certain methods and complex operations lacked proper documentation, and there were no comments explaining the business logic, parameter expectations, or error conditions. Accordingly, this was something I wanted to significantly improve for my enhancement, and the entire code was completely refactored and documented to ensure that it would be much more readable and easier for other developers to understand and modify.

* **Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

One of the biggest lessons I learned from this enhancement process is how many moving parts go into creating a high-quality, production-ready application. As noted, the original codebase was minimal and straightforward—great for learning, but limited in complexity. Once I started layering in new features, accessibility standards, and proper security practices, I realized just how challenging real-world app development can be.

I faced challenges with complex debugging scenarios and Android-specific technical issues, such as resolving Material Design theming conflicts that caused mysterious runtime crashes, troubleshooting database operations that led to app freezing, and deciphering cryptic Android error messages like the "1.0dip" NumberFormatException that appeared in seemingly unrelated code sections.

Despite the challenges, however, I found the process extremely rewarding. Refactoring Stock Shark pushed me to research Android best practices more deeply, improved my debugging workflow, and forced me to think about app design from both the user’s and developer’s perspectives. It strengthened my confidence in mobile development and reinforced that this is an area of software engineering I’d like to keep exploring—especially in roles that combine creativity, problem-solving, and user empathy.