3-2 Milestone: Enhancement 1- Software Design and Engineering

Bradley Wells

Southern New Hampshire University

CS499- Computer Science Capstone

Dr. Maciosek

January 20, 2025

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

The artifact I have chosen to make an enhancement on in the category of Software Design and Engineering is an Android mobile application developed at SNHU in the CS360 class, which was centered around Mobile Architecture and Programming. The intent of the application was to track inventory items, with the app's main feature being an inventory data table visible to the user and give the user the ability to perform CRUD (Create, Read, Update, Delete) functions within the application, altering the data in the database. An additional requirement of this application was to send an SMS notification to the user's phone number when any inventory item in the database reached a stock level of zero. This application was created in 2024 for the course and was progressively built with robust planning, User Interface design, and coding functionality into the application with Java.

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?

This artifact was selected because the project offered a complexity level suitable for showcasing in a portfolio but was still rudimentary enough to make worthwhile enhancements. It is also unique in the way that I will be able to perform all three areas of enhancement on a single artifact and continue to produce a better quality product throughout the duration of the course. This was also my first experience creating something full stack in nature, creating an application that connected the presentation layer to a backend database. While the artifact was fully functioning and met all the original requirements, there were some improvements that could be made in the area of software design and engineering. Specifically, these improvements included rewriting the existing Java source code into Kotlin and refactoring the application to better organize the source code. Refactoring the code made this application more modular in design, offering more scalability from the existing project and breaking the project into distinct, data, middleware, and presentation layers, separating out utilities like sorting and SMS management. The decision to port from Java to Kotlin was made for two primary reasons. Firstly, the preferred language for Android applications by Google is Kotlin for its more modern features (Android Developers, 2024). Additionally, converting to Kotlin represented a challenge for me and the experience of better understanding another programming language.

This enhancement showcases several key computer science skills, demonstrating versatility and technical prowess. Full-stack development capabilities are evident in the creation of an application that connects the presentation layer to a backend database, highlighting proficiency in both frontend and backend development. Software design and engineering expertise is demonstrated through the refactoring process, which improved the application's architecture and modularity by breaking it down into distinct layers and separating utilities. This modular approach not only improves current functionality but also enhances the project's scalability and maintainability, demonstrating foresight in software architecture design. The transition from Java to Kotlin showcases the ability to work with multiple programming languages, indicating adaptability and a commitment to staying current with industry trends. Additionally, this enhancement demonstrates mobile application development skills, specifically for the Android platform.

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?

With this enhancement, I planned on meeting course outcomes three and four from the course syllabus. These outcomes involve designing and evaluating computing solutions that follow algorithmic principles while managing design trade-offs, and using well-founded techniques, skills, and tools to implement computer solutions that meet industry goals. The plan to meet these outcomes was twofold: refactoring the code base to improve modularity and optimization (at the cost of additional initial development time) and migrating the code language from Java to Kotlin.

I believe I successfully met these course outcomes with this enhancement, as the proposed changes in the area of software design and engineering were completed and successful. Moving forward with the other enhancements, the remainder of the course outcomes will be demonstrated by employing strategies to build collaborative environments, delivering professional communications, and developing a security mindset. These outcomes will be met through the following enhancements: implementing a search and sort function in the next module, migrating the database in the following module, and adding password hashing for storing sensitive data in the database, which is currently lacking in the artifact.

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?

Throughout the process of enhancing the artifact in the area of software design and engineering, I learned a lot about the application planning process and encountered challenges when converting Java code to Kotlin. Specifically, I learned that more effort needs to be invested in planning an application or project than in the actual coding itself. Starting with a more well-thought-out plan makes coding and following modular design principles much easier.

Due to this realization and the process of re-coding the application, it felt almost like creating a brand new application. However, it was a valuable learning experience that allowed me to follow better practices in my design.

I faced challenges with this enhancement when converting from Java to Kotlin. Kotlin is a newer language to me, and while similar to Java, it is more concise. Fortunately, there is substantial documentation on Android Developers and from JetBrains to assist in understanding some of the intricacies. Another helpful practice was focusing on individual methods in classes to move through the fairly sizable code base systematically, which helped prevent feeling overwhelmed.

I'm aware that many Integrated Development Environments include plugins or tools for converting programming languages. However, I chose not to use these as they likely would have resulted in more confusion or errors in the project enhancement, as well as taking away the learning opportunity from myself.

References

Android Developers. (2024, June 27). *Android's Kotlin-first approach*. https://developer.android.com/kotlin/first

Southern New Hampshire University. (n.d). *Undergraduate course syllabus- CS 499: computer science capstone.*