ENHANCEMENT ONE: SOFTWARE DESIGN AND ENGINEERING

Abigail Sibazeu

CS-499 Computer Science Capstone

Southern New Hampshire University

01/26/2025

This document presents a narrative that complements the artifact enhancements within the realm of software design and engineering. It articulates the justification for selecting this artifact for inclusion in this portfolio category and offers insights into the methodologies employed during its creation. The emphasis is placed on the educational growth and understanding gained throughout the development process (Southern New Hampshire University, 2022).

The artifact chosen for the software design and engineering category is the mobile application named Inventory App. This application was developed to facilitate efficient tracking of item inventories using Android-based mobile platforms. Its conception, design, and development were integral components of the CS360 Mobile Architecture and Programming course within the computer science program. The app was implemented using Java as the programming language and leveraged SQLite as the relational database system. Development was carried out in the Android Studio integrated development environment (IDE), with the application tested and executed using a Nexus emulator.

This artifact encapsulates the comprehensive journey of software design and engineering. It embodies thoughtful considerations for enhancing the user interface and overall user experience, ensuring accessibility and ease of use through recognizable icons and intuitive navigation. The engineering approach prioritized robust practices such as data validation, adherence to security-oriented principles like "default deny," and crafting an architecture that supports both reliability and scalability.

Key aspects of the design process include establishing the logical relationships and interactions among algorithms, data structures, classes, methods, and the database. This ensured harmonious functionality across the application. The artifact also integrates a relational database to manage operations for adding, retrieving, updating, and deleting data, empowering users with a seamless and practical tool for inventory management.

Graphical user interface

Description automatically generated

*Figure 1 Design of Add Item Activity Layout*

The design and development of the mobile application demonstrate expertise in crafting software solutions that address and incorporate user requirements. This process involves interpreting user needs and translating them into a cohesive programmatic framework with well-structured relationships and activities. The development approach integrates the visual and interactive elements of the user interface with the functional and logical components, ensuring a seamless balance between design and implementation.

Significant enhancements introduced to the artifact improved its functionality and usability. The application now supports complete CRUD (Create, Read, Update, Delete) operations within the item's activity, enabling robust user interaction with the inventory. Additionally, improvements to the SQL Helper class enhanced the organization, functionality, and efficiency of its methods, strengthening the application’s backend architecture. These enhancements align with the objective of designing and evaluating computing solutions that effectively address specific problems while leveraging algorithmic principles and adhering to computer science standards. Design trade-offs were carefully managed to achieve an optimal solution.

Mobile development best practices were followed throughout the process, with the application of user-centered design principles and industry standards. These principles ensured that the application delivers value, meets user expectations, and aligns with professional goals. Innovative techniques, advanced tools, and well-established practices were employed to implement computing solutions that are both practical and impactful. Additionally, quality assurance practices were integrated to identify and mitigate potential vulnerabilities, ensuring the delivery of reliable and secure software.

A screenshot of a phone

Description automatically generated

*Figure 2 Login Activity and Register Activity Screenshot*

User authentication was implemented using the individual's name, phone number, and email as credentials, along with a password recovery mechanism to reinforce the zero-trust security model. This approach highlights a proactive security mindset throughout the development process, with measures designed to anticipate adversarial exploits in the software's architecture and design. The process focused on identifying potential vulnerabilities, mitigating design flaws, and ensuring the privacy and security of data and resources.

Security considerations were embedded at every stage of the software development lifecycle, from initiation and product assurance to testing and compatibility checks. Secure coding standards tailored to the target development language and platform were rigorously applied to enhance the application's resilience against threats. This commitment to security exemplifies an emphasis on safeguarding user information and maintaining robust protection of system resources while adhering to industry best practices for secure software development.

A screenshot of a computer

Description automatically generated with medium confidence Text

Description automatically generated

*Figure 3 JAVA Files Structure and Sample of Coding Formatting*

Industry-standard Java coding best practices were meticulously followed to ensure clarity, maintainability, and organization of the application code. These practices included the use of in-line comments, meaningful naming conventions, consistent formatting, and proper indentation, all in adherence to established coding standards. These measures contribute to improved code readability and streamline application organization.

Global variables were identified using camelCase CapWords naming conventions, while method-scoped variables adhered to camelCase lowercase or lowercase dash\_names for clear differentiation. Java class files were named using camelCase CapWords, whereas layout files followed a lowercase naming format for consistency and ease of identification.

Class method names were designed to reflect their purpose and functionality, aligning with their intended actions. Comment statements were incorporated to provide concise summaries of class and method functionalities, aiding in understanding and maintenance. Methods overriding superclass methods were named in camelCase starting with a lowercase word, while other methods followed camelCase with CapWords naming. These conventions collectively ensured a structured, professional approach to code development, aligning with industry standards for high-quality software engineering.

A screen shot of a calculator

Description automatically generated

*Figure 4 Items Activity List and Item Edit Feature Screenshot*

As a designer and engineer, the initial step involves carefully planning and programming the purpose of the idea while identifying the target audience. Each audience has unique needs and preferences, which must be interpreted and integrated into the development process. Prioritizing user requirements ensures that the most critical or desirable features are addressed first, akin to ranking tasks for incorporation during the app's initial launch phase.

An essential aspect of the process is analyzing competitive applications to gain insights into how similar challenges in UI/UX design have been addressed. Understanding how other apps present features comparable to the envisioned idea can guide design choices and enhance the overall user experience. Conducting research through app stores or online resources to identify and evaluate similar applications is a crucial strategy. This comparative analysis informs the development process, ensuring that the application aligns with user expectations and industry standards while distinguishing itself in a competitive market.

A screenshot of a smartphone

Description automatically generated

*Figure 5 App Info Screenshot*

The design and development of a mobile application is a complex process that requires a thorough understanding of best practices to create a high-quality app that resonates with the target audience. It is unrealistic to expect the creation of an exceptional application as an individual effort. Success in this endeavor hinges on recognizing the value of collaboration throughout the entire app lifecycle, from initial planning to final implementation.

Building a strong, collaborative team is essential for identifying strengths and addressing weaknesses. Such a team fosters the ability to digest and apply effective mobile app development practices, ensuring a well-rounded approach to the project. The process should focus on integrating design components seamlessly with programmatic code, continually seeking opportunities to innovate and refine the app. By embracing a team-oriented mindset and maintaining a focus on improvement, challenges can be effectively addressed, paving the way for a successful, user-focused application.