**Milestone Four: Enhancements and Reflection**  
Leah Launiuvao  
Southern New Hampshire University  
CS 499: Computer Science Capstone  
Instructor: Gene Bryant  
October 20, 2024

**Introduction:**

This milestone was my opportunity to apply my prospective enhancements to my **Inventory Management app**. This artifact is a mobile application which was designed to mange inventory, which I initially created during my *Mobile Architecture and Programming* course. I selected this item for my ePortfolio because it demonstrates the wide range of skills in software development that I have gained, including UI design, database management, security implementation, and user experience. Each of these enhancements were designed to improve different aspects of the app, from reliability, security to user experience. Throughout this process, I encountered and solved a variety of issues and faced many challenges, which has helped to strengthen my programming skills and enhance my understanding of the Android Studio environment.

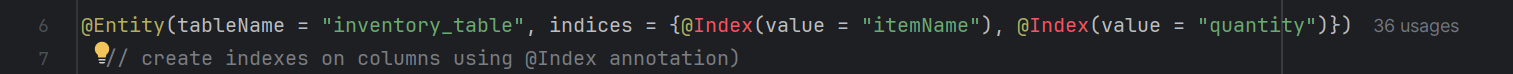
A screenshot of a login screen

Description automatically generatedA screenshot of a phone

Description automatically generated

**Enhancements:**

One of the key enhancements I implemented was database indexing to improve performance. My goal was to optimize search and retrieval times for larger data sets. To achieve this, I added indexes to the *InventoryItem* table for fields like *itemName* and quantity using the *@Index* annotation in the *@Entity* definition as seen below:



This helps optimize search times for commonly queried fields, making the app more efficient. No adjustments were needed in *AppDatabase.java*, as *Room* automatically incorporates the indexing changes. To test the impact, I populated the inventory table with more data by adding a loop in *InventoryActivity* to insert test items.

A computer screen shot of a code

Description automatically generated

I then created a test method, *testQueryPerformance()*, in *InventoryActivity* to measure the execution time for the *getAllItems()* query.

A screen shot of a computer code

Description automatically generated

This improvement demonstrates my understanding of database optimization techniques. By indexing the columns, I was able to significantly improve query performance, especially when dealing with large datasets. Including this enhancement in my ePortfolio showcases my ability to apply best practices in database management and optimization, which is crucial for scalable software.

Another important enhancement was integrating password hashing for enhanced security. The goal here was to secure user credentials by encrypting passwords. To accomplish this, I integrated password hashing using *bcrypt* to secure user credentials before storing them in the database. I updated *LoginActivity.java* to hash passwords during both account creation and verification. This enhancement took much longer than expected due to errors during implementation, but it was a valuable learning experience. It required modifying both *CreateAccountActivity* and *LoginActivity* to handle hashed passwords appropriately.

A computer screen shot of a computer code

Description automatically generated

This change highlights my focus on security and my ability to handle sensitive data securely. It demonstrates my ability to implement best practices in user data protection, which is essential in software development. Including this in my ePortfolio emphasizes my commitment to developing secure applications, which is fundamental to modern software development.

To further enhance the app, I also strengthened error handling by adding retry logic. The goal was to improve app reliability and user experience by adding robust error-handling mechanisms. I added try-catch blocks around CRUD operations in both*InventoryActivity*and *LoginActivity* to handle potential errors.

A screen shot of a computer program

Description automatically generated

Additionally, I implemented retry logic for CRUD operations with up to three attempts before giving up, making the app more resilient against temporary issues.

In the *loadInventoryData()* and *testQueryPerformance()* methods, I added retry-mechanisms to enhance reliability during data loading and testing. I also added Toast messages to inform users if operations failed after multiple tries to ensure that the user is informed. This demonstrates defensive programming, which is core skill in reliable app development. By including retry logic and error-handling, I made the app more robust and user-friendly. This enhancement shows my ability to anticipate and handle potential errors, ensuring a smooth user experience, even with complications.

I also made improvements specifically in *LoginActivity*, adding input validation to ensure that username and password fields are not empty before querying the database, which prevents unnecessary operations and improves user experience. Additionally, I introduced retry logic to improve resilience during login attempts and provided user feedback using Toast messages to inform users of retries or issues, making the app more transparent for users.

**Additional Enhancements:**

Along with these core enhancements, I added a home button with a home icon to the inventory management page. This allows users to easily navigate back to the home screen, improving the overall usability of the app.

A screenshot of a phone

Description automatically generated

Another enhancement I addressed was an issue with the Create Account feature. I created a *CreateAccountActivity* java class and the corresponding XML file to move the account creation feature to a separate page from the login page. This change was made to make the app more intuitive for users, providing a clearer distinction between account creation and logging in, ultimately improving the user experience.

A screenshot of a cell phone

Description automatically generated

**Challenges and Take Aways:**

Throughout this process, I encountered various challenges, from integrating *bcrypt* for password hashing to resolving layout issues with the home button icon. One particularly difficult problem was a build error that prevented the project from running. To troubleshoot this, I tried several approaches including but not limited to: deleting the *transform-4* file and making adjustments to various project configurations, but none of these worked. Ultimately, I discovered that removing the ids.xml file allowed the project to build and run successfully. This taught me the importance of systematic troubleshooting, trying one solution at a time and verifying its impact.

Another issue I faced was related to sourcing the custom fonts for my app. Initially, the fonts were not being applied correctly, which led to inconsistencies in the UI. To resolve this, I had to adjust the font-related *XML files* (lato.xml and nunito.xml) and ensure the resources were being referenced properly. This experience improved my understanding of Android resource management and taught me to pay close attention to details when managing UI components.

Although resolving these errors felt overwhelmingly tedious at times and seemingly without resolution, being able to ultimately find a solution was a very gratifying experience. Every challenge throughout this process that I encountered from layout issues to syntax was an essential lesson. Working through these challenges helped me become more familiar with Android Studio's diagnostic tools, coding suggestions and debugging, which proved invaluable in understanding how the different components interact. I also learned the importance of defensive programming and how retry logic can improve user experience by contributing to reliability. Working with database indexing taught me how to optimize data retrieval, which is crucial for scaling applications with larger datasets. By pushing through these challenges, I gained a deeper understanding of Android development as a whole and strengthened my skills in problem-solving and debugging. This experience has been a significant learning opportunity, helping me grow as a developer.

With these enhancements, I believe I have met the course outcomes I planned to address in Module One. Specifically, I have demonstrated my ability to design and implement secure, optimized, and user-friendly computing solutions. I also showcased my skills in defensive programming and database management. There were more additions that I wanted to implement and will likely do on my own time, as I underestimated the time required and roadblocks encountered. This experience has allowed me to grow in multiple areas, including error handling, database optimization, and security, all of which align with the course's expected outcomes. These enhancements make my artifact a strong representation of my abilities and are why I have included it in my ePortfolio to showcase what I have learned.