

Milestone Four: Enhancement Three: Databases

Joshua M James

Southern New Hampshire University

CS 499: Computer Science Capstone

Prof. Brooke

April 7, 2024

Milestone Four: Enhancement Three: Databases

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

The Artifact I'm going to look at is my inventory application. The goal of this application is to track inventory items. This application was developed within android studio using java featuring an SQLite database. This application was from my CS 360 Mobile Architecture and Programming course and was created last term.

2. 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?

Selecting my inventory application for the database section of the enhancements justifies my inclusion for my ePortfolio. I have selected this artifact because it has allowed me to showcase various software development skills and techniques. This inventory application has allowed me to demonstrate database design, UI/UX design, and data management. I demonstrated these skills by transitioning to Firebase from SQLite, creating user-friendly UI design for account creation and password resetting. Allowing me to design and develop a practical application. For my database enhancement, I initially proposed implementing passwords hashing with a SHA-256 algorithm to store user passwords and implementing a password reset feature. In my original artifact user passwords were being stored as plain text within the SQLite database. This was a vulnerability within my application as it does not follow secure practices. As well as there was no current way to for a user to reset their password if forgotten. To address this, I implemented password hashing using the SHA-256 algorithm. The hashed passwords were then stored into my SQLite

database. To enhance security, I implemented password hashing mechanism into the authentication process. First, I started by creating a new java class named PasswordHash. The hashPassword() function takes a plain text password as the input and returns a hashed password. The class converts the plain text into a byte array using the getBytes() method, then the byte array is converted into a hexadecimal string. While this worked as intended after tests, I ran into another limitation in my application. Including a password reset function that truly worked and remained secure was a problem. The inventory application stored and verified the user information locally using SQLite.

To enhance security, user experience, and scalability I decided to make my enhancement three to integrate Firebase to hash passwords and have password reset functionality. Firebase offers cloud-based development tools, and I specifically used Firebase Authentication. This transition required a few extra steps to get Firebase to work. I had to change the initial create account and login logic to get the password reset to work. After making this change, it allowed a user to create and login using Firebase Authentication and Firebase then hashes the password using the SCRYPT algorithm, then I was able to work on the password reset.

3. Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

With the addition of password hashing and password reset using Firebase Authentication, I was able to meet the course objectives planned in module one. The enhancement aligns with:

- Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.

Throughout the development of enhancement three of my inventory application, there was a challenge to ensure the security of user data stored in the SQLite database. To address this, I initially implemented password hashing with the SHA-256 algorithm. However, after some careful consideration I decided to implement Firebase to handle the user accounts. Implementing firebase addresses the problem in my inventory application of password hashing and password reset functionality. Demonstrating algorithm principles by involving user input, validating email addresses, and using Firebase Authentication services to send password reset emails. Also Using the SCRYPT Algorithm to hash users plain text passwords that is built into the Firebase Authentication Dashboard. The SCRYPT algorithm works by generating a pseudorandom sequence of bytes from an input, then this sequence goes through different iterations of hashing operations. According to Firebase, “Firebase Authentication uses an internally modified version of SCRYPT to hash account passwords. Even when an account is uploaded with a password using a different algorithm, Firebase Auth will rehash the password the first time that account successfully logs in.”

- Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.

Integrating Firebase Authentication provided a more efficient solution for my artifact. Allowing the user to reset create an account using Firebase within my application, allowed for an

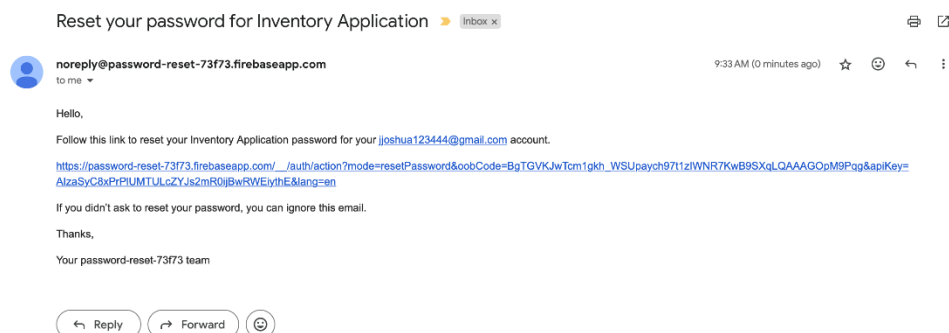
innovative approach by providing seamless password resets through email verification, and hashing of user passwords uses well-founded techniques that align with industry specific goals. Additionally, utilizing Firebase's libraries and frameworks, such as FirebaseAuth, aligned with industry standards and best practices, further enhancing the value and functionality of the application.

- 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.

By implementing password hashing and password reset functionality using Firebase Authentication, I have demonstrated a proactive approach to potential security vulnerabilities in my inventory application. Applying these changes ensures the privacy and security of user data. I also realized I design flaw with my current application that used SQLite to store user information locally. Using the `createUserWithEmailAndPassword` method to save new users information to the Firebase Authentication dashboard with a hashed password.

Identifier	Providers	Created ↓	Signed In	User UID
joshua123444@gmail...	📧	Apr 3, 2024	Apr 3, 2024	eIvb2y1lhWNVFs9kR3j2pZxHu...


As well using the `sendPasswordResetEmail` method to securely send a password reset link to the user's email.



Reset your password

for **jjoshua123444@gmail.com**

New password



SAVE

- Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science.

Integrating Firebase Authentication I was able to meet this course outcome. Firebase, being a service that is provided by Google, offers a platform for collaboration among diverse audiences, which will enable more organizational decision-making in the field of computer science. By using Firebase, this allows for an environment where others familiar with this platform can contribute and improve my artifact.

4. 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?

Reflecting on the process of enhancing and modifying, there were many valuable skills I have learned when improving my artifact. Implementing Firebase Authentication

gave me a better understanding of utilizing different technology, API integration, libraries, and frameworks it uses. In addition to this, it allowed me to better understand the process of transitioning from an SQLite database to a cloud database. This has given me more confidence in my ability to enhance security and usability. While learning a lot in this enhancement there were some challenges I have faced. Previously stated earlier, creating the initial hashing of the passwords using the SHA-256 algorithm, gave me the realization that there was a need to implement a working password reset using an email address. Transitioning from SQLite to Firebase Authentication posed a challenge to me initially due to the fact I had to make sure my previous classes were set up correctly for this integration.

References

Firebase authentication password hashing. Firebase Open Source. (n.d.).
<https://firebaseopensource.com/projects/firebase/scrypt/>