

CS 499

Professor Dr. Fitzroy Nembhard

Saugat Niroula

9<sup>th</sup> June 2024

### **Narrative for Database Enhancement**

#### **Artifact Description and Origin:**

The artifact selected for the Database category is the Inventory Management Application. This application was initially developed for the CS360 Mobile Architecture and Programming course in September 2023 using Android Studio. The primary purpose of the app is to streamline inventory management in warehouse environments by offering features such as user authentication, inventory display, item addition and removal, quantity adjustment, and low inventory notifications. The motivation for this project stemmed from personal experiences in a warehouse setting, highlighting the need for a more efficient inventory management system.

#### **Justification for Inclusion:**

This artifact was chosen for inclusion in my ePortfolio because it demonstrates key skills and abilities in database management, particularly in the areas of integrating a cloud-based database, implementing complex SQL queries, and ensuring data security. The original application showcased foundational skills in managing a local SQLite database for inventory management. However, the enhancements made to the artifact further highlight my proficiency in integrating Firebase Firestore, optimizing data retrieval, and securing sensitive user information.

Including this artifact in my ePortfolio not only showcases my technical capabilities but also provides evidence of my ability to adapt to new technologies and implement advanced database solutions. The transition from a local SQLite database to Firebase Firestore illustrates my commitment to leveraging modern, scalable, and efficient data storage solutions. Moreover, the implementation of complex queries and real-time data synchronization through Firestore demonstrates my understanding of advanced database concepts and their practical applications. This enhanced artifact serves as a testament to my ability to manage and secure data effectively, making it a valuable addition to my professional portfolio and a strong indicator of my readiness for complex database management roles.

### **Enhancements and Course Objectives:**

In alignment with the enhancement plan, the following improvements were made:

#### **1. Integrate Cloud-Based Database:**

- **Firebase Firestore Integration:** Replaced the local SQLite database with Firebase Firestore for real-time data synchronization. This involved creating a **DataMigration** class to transition data from SQLite to Firestore without data loss. The DataMigration class contains methods such as **migrateItems()** and **migrateUsers()** that read from SQLite and write to Firestore.
- **Firestore Queries:** Updated data retrieval methods to use Firestore queries instead of SQLite queries, ensuring real-time data synchronization and improved scalability. For instance, methods like **getAllItems**, **searchItems**, and **sortItems** in the **DataController** class now utilize Firestore for data operations.

#### **2. Implement Complex SQL Queries and Triggers:**

- **Complex Queries:** Enhanced data retrieval and manipulation by implementing complex queries using Firestore. For example, the **filterItemsByQuantity** method in the **DataController** class allows filtering items with quantities greater than or equal to a specified value, ensuring efficient data handling.
- **Triggers for Inventory Updates:** Created Firestore functions to automatically update inventory levels and notify users when quantities are low. For instance, the **updateItem** method in the **DataController** class updates item quantities and triggers notifications when inventory levels fall below a predefined threshold.

### 3. Data Migration:

- **Data Migration Scripts:** Developed data migration scripts to transition data from SQLite to Firebase Firestore without data loss. The **DataMigration** class handles this process, ensuring that all existing data is accurately migrated to the new database. Methods like **migrateItems()** and **migrateUsers()** read data from SQLite and write it to Firestore, ensuring a seamless transition.
- **Seamless Transition:** The migration process was designed to be seamless, with minimal disruption to users. This ensures that the app continues to function smoothly during and after the transition. The **DataMigration** class facilitates this by managing the data transfer efficiently.

### 4. Improve Security:

- **Encryption of User Credentials:** Implemented encryption for user credentials to enhance security. The **DataMigration** class includes methods like **encrypt()** to

securely encrypt user passwords before storing them in Firestore, ensuring data protection.

- **Secure Data Storage:** Ensured that all sensitive data, such as user passwords, is securely stored in Firestore using encryption techniques. This is achieved through the use of the **encrypt()** method in the **DataMigration** class, which utilizes AES encryption to protect user credentials.

## 5. Testing Database Interactions:

- **Unit and Instrumentation Tests:** Added tests for database interactions to ensure reliability. For example, the **ExampleInstrumentedTest** class includes tests for Firestore integration and encryption functionality, verifying that data is correctly stored and retrieved. The **ExampleUnitTest** class was also updated to include tests for the new methods in the **DataController** class, ensuring the accuracy and reliability of data operations.

These enhancements align with the course outcomes by demonstrating advanced skills in designing and evaluating database solutions using best practices in data management and security. They also support the course objectives of implementing well-founded techniques and tools for efficient data management and retrieval.

## Learning and Challenges:

The process of enhancing this artifact provided valuable learning experiences. Integrating Firebase Firestore required a deeper understanding of cloud-based databases and real-time data synchronization. Implementing encryption techniques expanded my knowledge of data security and best practices for protecting sensitive information. Developing data migration scripts

involved understanding the intricacies of data transition and ensuring data integrity throughout the process.

Throughout the enhancement process, several challenges were encountered. Ensuring seamless data migration while maintaining data consistency required careful planning and execution. Implementing real-time data synchronization with Firestore involved updating existing data retrieval and manipulation methods. Additionally, ensuring the security of user credentials through encryption added a layer of complexity to the project.

Overall, the enhancements made to this artifact demonstrate significant progress toward achieving the course outcomes and showcase my capabilities in database management and security. The improved application not only offers enhanced functionality and better performance but also reflects a high standard of data security and reliability.

This narrative, along with the enhanced artifact, will be included in my ePortfolio to showcase my skills and abilities in database management and security, supporting my career advancement and demonstrating my proficiency in this critical area of computer science.