**Module 5 Narrative**

Jordan Taylor

The artifact that I chose to work on for the databases category is the CS-360 Inventory Mobile Application. I created this application at the end of the previous term. I felt this application was a unique opportunity to use all three categories, and all the code was written by me based on the assignment requirements. There was no fill in the blank sort of scenarios for the application like there are often times with assignments. The application as a whole really shows the skills that I have in creating all aspects of an application both front-end and back-end. With the addition of the new database, it also shows that I have cloud development skills. The databases outcome is now complete along with the Data Structures and Algorithms and Software Engineering categories.

When working on the database items specifically, I learned that in AWS, DynamoDB is very easy to set up and get working thanks to their cloud database management system. Documents can be added and removed with ease. The NoSQL key/value pair provides an extremely flexible static data storage solution to work with. Most of the difficulty was getting familiar with Boto3; the Python library used to interact with the database. AWS provides decent documentation that makes it fairly intuitive to use the methods provided.

Refactoring the front-end of the application to use the REST API that I created to interact with the new database was particularly challenging. Even though the front-end was written in a modular fashion, there was more work to be done than anticipated. I also needed to learn a new library called Retrofit to interact with the API. Once I understood how the library worked, things became easier though. Retrofit did not have very good documentation or examples so I had to rely on google to search for examples to fill in the gaps that the Retrofit documentation was missing. I also made some assumptions about how the database worked that ended up making me have to refactor some of my REST API business logic. I thought that if I was going to add a duplicate item then I would receive some sort of error. I wanted to rely on that for the front end of the application when creating users but found out that if an add\_item method is used to add a duplicate item, it simply rewrites the current one. I had to make an additional database read call to ensure that the item ID didn’t already exist. This read call is made from the API to avoid unnecessary network traffic to and from the device making the request.

**Database Tables, Items, and Users**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**Models Used in the Front-End**

**A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**