**Software Engineering Narrative 1**

Jordan Taylor

The CS-360 Mobil Architecture and Programming project provides a perfect starting point for application enhancement. This application provides proof of concept so users can see what an inventory app can do. The application also uses a database. For the application to be useful, it needs some enhancements such as a publicly accessible database and search functionality. The enhancements that are needed also add security concerns that were not originally present in the application.

The categories that need to be addressed for the project include software engineering, algorithms and data structures, and databases. My plan to move the database to a publicly accessible location touches on the software engineering and database categories. This week I was able to set up the resources needed to move the database including AWS API Gateway, Lambda, and DynamoDB. I have also developed some of the business logic within the Lambda function in addition to creating the tables needed in the database, while setting up resources and HTTP methods in the gateway to make it all accessible.

To achieve this, a considerable amount of troubleshooting was done. I came across issues with getting some of the python libraries that can easily be downloaded locally to work correctly inside of AWS Lambda. I ended up having to use my skills in software security to come up with an alternative method to salt and hash passwords. I also used a fair amount of time troubleshooting the permissions for the Lambda function to be able to talk to the database. It was also rewarding using my skills in software engineering to make the business logic modular so that when I came across an issue, I knew the exact module that was causing me a problem thanks to AWS CloudWatch.

I have not yet brought the software engineering category to its conclusion because of how intertwined the categories are within my project, but I do believe that I have the most difficult part out of the way. After getting the resources set up, I was able to test my API with Postman to add a user to the database. The user gets added to the database with a unique username in addition to a password salt and hash field. This milestone marks the completion of one complete route through the API – the POST user route. API gateway receives the user data in an HTTPS request, the main lambda function looks at the resource requested and invokes the create method in the CRUD module that I created, the CRUD module calls the password module to salt and hash the password, uses the AWS boto3 library to put the user data into DynamoDB, then calls the Response module to generate a response.

A screenshot of a computer

AI-generated content may be incorrect.

A white background with black and white clouds

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.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.