Robert Lowrey

CS – 499

August 3rd, 2024

**Milestone Four Assessment**

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

The artifact that I chose to meet this enhancement category is the same Python application that I have been working on. It was originally created in March 2023, during my enrollment in the CS–250 Software Development Lifecycle course. For the Software Engineering and Design enhancement, I ported the Java application to Python. Additionally, I added more destinations, expanded the descriptions, and implemented a feature where clicking on a destination’s image or description changes the background color to highlight the selected row. For the Algorithms and Data Structures enhancement, I implemented a feature that hides the travel destinations window until the user logs into the application with the correct username and password.

For this enhancement, I implemented a local NoSQL database using MongoDB Compass to store the usernames and passwords of users who create an account within the application. Therefore, to view the travel destinations, the user must have an account that is stored in the database. If the user is new to the application, they can click the “Create Account” button which will generate a window. Within the window, the individual can create a username, password, and role to be stored in the database. The chosen password must be entered twice to verify that the spelling and capitalization is correct. When the account creation is successful, the password is stored with a salted hash in the database, and the user can sign in to the application to view the travel destinations. Additionally, I have added role-based access control (RBAC), where the ‘Admin’ role can delete selected users from the MongoDB database.

**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?**

I chose this artifact because working on the enhancement of the database implementation would reintroduce me to the process of integrating a database within a Python project. It was also a new learning experience when it came to the Bcrypt library implementation, which helped familiarize me with creating a salted hash password before storing it in the database for an additional layer of security.

The enhancements that I have implemented showcase my skills and abilities in software development by maintaining a connection to a NoSQL database that stores usernames and passwords for different users. Another skill demonstrated by this enhancement is the ability to salt hashed passwords for an extra layer of security. Lastly, these enhancements highlight my technical proficiency in data modeling using tools applied during both the database and RBAC implementation.

**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?**

I have met all of the planned course outcomes from my Module One Enhancement Plan. The implementation of the database aligns with course outcome number four by using well-founded and innovative techniques that deliver value and achieve industry-specific goals of storing usernames and passwords in the NoSQL database. Additionally, the RBAC allows the ‘Admin’ role to delete users from the database by selecting their username in the ‘Admin Window’.

I also align with course outcome number one, which focuses on employing strategies for building collaborative environments that enable diverse audiences to support organizational decision making. This is achieved through the utilization of MongoDB. Since MongoDB is scalable and flexible, it enables future developers to contribute to new implementations. It can also enable multiple users to work on the same project to analyze, collectively improve and update the database.

The account creation feature aligns with course outcome number five by developing a security mindset that anticipates exploits in software architecture and design. This is achieved because passwords are stored as a salted hash in the database. As previously mentioned, when creating an account, the individual must enter their password twice using the correct capitalization and spelling when entering each one; otherwise, an error message will populate indicating that the passwords do not match. As of now, I do not have any updates to my outcome-coverage plans.

**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?**

I have learned a lot from implementing the NoSQL database in the artifact. I received a refresher on how to integrate MongoDB into a Python program. Additionally, I learned how to incorporate an account creation window that allows users to create usernames and passwords in the database. I also learned how to share a local MongoDB Compass database for others to view and utilize. Another concept I learned from this enhancement was how to use the Bcrypt library within Python to salt hashed passwords before storing them in the database. Furthermore, I learned how to remove unnecessary white space from user entries to ensure that only meaningful data is stored in the database. Lastly, I learned how to incorporate RBAC in the application to enable different types of access based on the roles that the user can choose from.

I faced some challenges along the way, one of which was having to change both my original plan and artifact for this enhancement. I originally chose to implement a database within an Android Studio project; however, my Android Studio development tool was no longer able to run the artifact that I wanted to enhance. After troubleshooting the issue for several days, I decided to change my artifact for this enhancement. The last challenge I faced was sharing the local database for submission. I eventually found, through extensive research, that exporting the collection to a JSON or CSV file to submit would be sufficient for viewing and accessing the database.