**5-2 Milestone Four: Narrative**

Kristie O’Brien

Science, Technology, Engineering, and Mathematics Department, Southern New Hampshire University

CS-499 Computer Science Capstone

Dr. Troy Hawk

August 3, 2025

**5-2 Milestone Four: Narrative**

The artifact that I have selected to enhance is my final project from my CS 360 class. It was originally completed at the end of the CS 360 course. The purpose of this artifact is to function as an inventory app that will allow a user to have a login credential set and their own inventory. The user can then update their personal inventory by adding items, deleting items, and changing the number of items in their inventory. If the user wants to, they may also opt in to receiving text messages should an item in their inventory reach zero. As of the time of the original completion of this project, it was functioning according to the specifications of the CS 360 final project criteria.

There were many reasons I included this project as one of the artifacts in my ePortfolio. One reason was that this project specifically showcases my coding skills towards a practical and fully functional app, which has the potential to benefit a user’s life. As such, it has a functional user interface. Therefore, this artifact showcases my skills to code for both backend code development with a database and logical functionality to manage a database, while also showing my ability to code a forward-facing user interface. So, this project shows a more holistic project compared to some of my earlier projects. Furthermore, I knew that this was a project I had the ability to enhance further and in a more robust way. Shortly after I took CS 360, I took a class on secure coding. In this class, I learned a great deal about the importance of secure coding against attacks and also against potential loss of data. As such, I decided to apply the security concepts I learned in this class to my fully functional app to make it more secure for the users and the code. So, I could show my security skills as well as my coding skills through this artifact, which supports the course outcomes of maintaining a security mindset for both offensive and defensive coding. By adding hashing with SHA components, I showed my skills in securing data for user security and code security. The SHA hashing component was specifically applied to user passwords. When a user enters a new password, that password is hashed. Additionally, by using SQL Injection protection, I also show how I am able to protect code defensively and offensively to protect against harmful queries. This shows my thinking with a security mindset towards code vulnerabilities that may be inherent in the code based on its reliance on SQL queries.

In considering all the above, I do believe that I have achieved the course outcome I set out to achieve with this artifact. Specifically, I set out to achieve course outcome five, which states: “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.”. This course outcome is entirely related to the importance of security. By implementing password hashing and SQL Injection protection, I have shown I can mitigate design flaws, ensure privacy for user data with hashed passwords, and anticipate adversarial exploits with SQL Injection protection. At this time, I do not currently have plans to change the outcome coverage.

The process of enhancing and modifying the artifact was actually rather smooth sailing compared to some of the earlier artifacts. Since it was relatively recently when I took my secure coding class, it was already fresh in my mind how to implement a regex pattern for SQL Injection protection. Additionally, from a cryptography class that I took for my math degree, I was familiar with the SHA process of hashing from a theoretical perspective. In earlier coding classes, I also used SHA for hashing data. As such, implementing these changes was not overly challenging. The challenges I faced were deciding how best to implement these changes in the queries themselves. This was mostly solved with a trial-and-error approach. After trying a few different configurations, the final status of this artifact was decided upon as it made the most sense intuitively.

In the process of working on this artifact, I learned several things. First and foremost, I learned how to implement a hashed password within SQL queries. Additionally, while I knew how to implement a protection against SQL Injection with a regex pattern recognition system, I learned how to utilize such protection throughout the code.