# CS499 Artifact One Narrative

For my Artifact in the Software Engineering and Design category, I chose to enhance one of the applications I developed in Java during my previous course, IT-145: Java Programming. I was tasked with developing a Java software application for the fictitious company Grazioso Salvare to help track search-and-rescue animals. These search-and-rescue animals are obtained and trained by the company to rescue humans from life-threatening situations. The required functionality of the application was to intake animals, reserve animals, and print available animals to the user's console. My enhancements for this artifact include porting it to Python, adding a SQLite database for persistent storage, and implementing login functionality with secure credential storage.

I selected this artifact for my ePortfolio to demonstrate a comprehensive understanding of programming languages and concepts, including object-oriented design, characterized by modularity, reusability, scalability, and maintainability. I also want to demonstrate my security mindset by creating software that is secure by design. Initially, I followed my enhancement plan by manually porting each class, including the database class and its associated services, to perform CRUD actions and other methods. However, I soon discovered SQLAlchemy, a Python SQL toolkit and Object-Relational Mapper, which already included all the functionality I needed. This was achieved through a declarative mapping that defines both the Python object model and its database metadata, describing the actual SQL tables in our database. I used single-table inheritance for the RescueAnimal parent class and its children, Dog and Monkey, making my queries more efficient because only one table is involved rather than complex joins across multiple tables. Additionally, a new class, User, was added, which was necessary for implementing login functionality and is mapped to the Users table. All database methods were decoupled and implemented in the DataBaseServices class, and the program’s main logic was implemented in a separate Driver class as well. To secure login credentials, I implemented PBKDF2 hashing to store users' passwords securely in our database, and added brute-force protection to the login functionality.

By implementing my enhancements, I achieved both of my planned course outcomes, demonstrating my programming skills and ability to use the latest tools in computer science, as well as my security mindset in developing applications that ensure the privacy and enhanced security of data.

During the process of enhancing and modifying the artifact, I learned new tools and libraries, such as SQLAlchemy, which gives application developers the full power and flexibility of SQL. I learned how to obtain and securely store credentials using PBKDF2 (Password-Based Key Derivation Function 2) with the help of another powerful library, hashlib. Porting was challenging at first due to the differences between the two languages, but once I found the right tools, it became clearer, and the final solution became achievable.

Resources:

*Sqlalchemy*. SQLAlchemy. (n.d.). <https://www.sqlalchemy.org/>

*Hashlib - secure hashes and message digests*. Python documentation. (n.d.). <https://docs.python.org/3/library/hashlib.html>