Module 4 Artifact Narrative Overview

CS499 – Computer Science Capstone

Benjamin Leanna

SNHU

2024 – 03 – 30

**Brief Description of the Artifact:**

The Driver.java class, originating from the IT145 Application Development coursework, serves as the entry point for executing algorithms and data structures related to animal rescue management. Originally, the Driver class provided basic functionality for sorting animal data using the quicksort algorithm and performing binary search operations to locate specific animal records. However, the original implementation lacked comprehensive error handling mechanisms and optimization techniques. Through enhancements made in Milestone Two, the Driver class underwent significant improvements, including the optimization of sorting and searching algorithms, integration of error handling mechanisms, and enhancement of code readability. These enhancements transformed the Driver class into a robust and efficient component of the animal rescue management system.

**Justification for Inclusion:**

The selection of the Driver.java class for inclusion in the ePortfolio stems from its pivotal role in showcasing skills and abilities in software development, particularly in the domain of algorithmic problem-solving and data management. The enhancements made to the Driver class aimed to optimize its functionality, improve code readability, and ensure strong error handling, aligning with the objectives of the Computer Science program. By implementing efficient sorting and searching algorithms, the enhanced Driver class demonstrates proficiency in algorithmic principles and their practical application in computing solutions.

**Alignment with Course Objectives:**

The planned enhancements align closely with the course objectives outlined in Module One of this class. Specifically, the enhancements to the Driver class aimed to demonstrate proficiency in designing and evaluating computing solutions using algorithmic principles and computer science practices. The integration of the quicksort algorithm for sorting and binary search algorithm for searching directly addresses this objective, showcasing the ability to develop efficient solutions to complex problems. Additionally, the implementation of error handling mechanisms reflects a commitment to developing reliable and robust software systems, aligning with the course objective of employing strategies for building collaborative environments.

**Reflection on the Enhancement Process:**

The enhancement process provided valuable insights and learning experiences, allowing for the practical application of theoretical knowledge in software development. Throughout the process, challenges such as optimizing algorithm efficiency while maintaining code readability were encountered. However, these challenges served as opportunities for growth, fostering a deeper understanding of algorithmic principles and software design best practices. The enhancement process not only improved the functionality of the Driver class but also enhanced skills and knowledge in algorithmic problem-solving and software engineering practices, contributing to the continuous refinement of the artifact.