Maridelle Anne Gonzales

CS499

The artifact is a Pet Check-in/Checkout system created as part of my coursework in IT145 and refined during subsequent projects. It is a C++ console application that manages the boarding of pets, specifically dogs and cats, using object-oriented programming principles. The system was originally developed to demonstrate class design, inheritance, and polymorphism, and later enhanced to include file-based storage for persistent data management, enabling CRUD-like operations on pet records.

I included this artifact in my ePortfolio because it effectively showcases my software development skills, particularly in object-oriented design and data management. The system demonstrates the use of encapsulation, inheritance, and polymorphism: the base class Pet abstracts common properties, while derived classes Dog and Cat extend functionality and override methods such as displayInfo() for custom behavior. The enhancements I implemented, including file-based storage, search functionality, reporting of boarded pets, and updating pet data, illustrate my ability to connect OOP concepts to practical data persistence, effectively simulating database operations.

This artifact helped me meet the course outcomes for Databases, Software Design and Engineering, and Algorithms and Data Structures. Through the enhancements, I reinforced my understanding of data structures, file I/O as a form of lightweight database management, and modular design. It also allowed me to practice writing reusable, maintainable code and linking conceptual design to functional implementation.

During the enhancement process, I learned the importance of systematic design and planning, especially when integrating new features like search and update functionalities. One challenge I faced was managing file synchronization with in-memory data while ensuring memory was handled correctly when pets were checked out. I resolved this by implementing dedicated functions to update the file whenever the vector of pets changed, reinforcing my understanding of maintaining data consistency. Additionally, designing the interface to be intuitive for users while still demonstrating advanced OOP concepts was a valuable exercise in user-centered programming.