CS499

Rikki Xaysanasy

Professor Bhandari

9/27/2025

**Milestone Three – Algorithms & Data Structures Artifact**

Artifact Description  
The artifact I selected is *PetCheckIn.java*, which I first created in IT-145: Foundations in Application Development. Originally, it was a simple console program that allowed a user to check in a pet by entering details such as name, type, and age. The program tracked available spaces for dogs and cats and calculated the bill for a pet’s stay. I chose this artifact because it was one of the earliest programs I wrote and it represents the starting point of my computer science journey.

Justification for Inclusion  
I included this artifact in my ePortfolio because it highlights my growth in algorithms and data structures. The original version used basic conditional logic and counters. In my enhancement, I applied more advanced structures:

* HashMaps to store customers and pets for quick lookup.
* Queues to represent the order of check-ins.
* Stacks to allow undoing the most recent check-in.
* Binary search for finding pets by name.

These enhancements move the program beyond a simple input/output console app and into a demonstration of real algorithmic thinking. The artifact now showcases my ability to choose and implement data structures that fit business requirements.

Course Outcomes  
I planned to demonstrate skills in designing and evaluating computing solutions that apply algorithmic principles. With this enhancement, I believe I met that outcome. I can now show how a simple console application was refactored to use industry-relevant structures, paving the way for a scalable web application. No major updates are needed to my outcome-coverage plan because the enhancement stayed aligned with what I outlined.

Reflection on the Process  
Enhancing this artifact taught me the value of planning and structure. At first, it was just about adding new data structures. But I learned how important it is to integrate them cleanly, so the program still feels natural to use. For example, implementing a queue for check-ins forced me to think about fairness and order in a way I had not considered in the original code. I also ran into challenges when converting from the original Java version to a JavaScript/Node.js environment, such as handling asynchronous data flows and preparing the program to connect with a MongoDB database later on.