**Enhancement Two: Algorithms and Data Structure**

Cydnie Fisher

Southern New Hampshire University

CS-499 Computer Science Capstone

Professor Mike Alesso

November 21, 2024

**Artifact Description**

For the second enhancement category, algorithms and data structures, I will add to what was completed in the first enhancement category. The original version of the artifact was the final project for IT145, which was created using Java in February of 2023. This artifact is a program created for an animal search and rescue company, allowing them to intake new animals, reserve rescue animals, and print lists of animals based on animal type and reserved status. Once an animal is reserved, it can be sent out on rescue missions to save humans from dangerous, and potentially life-threatening, situations. For the first enhancement category, the artifact was converted into Python, and the overall functionality remained the same.

**Why This Artifact Was Selected**

IT 145 was one of the first coding classes I had taken through SNHU. At that time, I did not have a strong understanding of algorithms or data structures. The project requirements wanted the dogs and monkeys to be stored in an array list, but I now understand that there is a much more efficient way to store and search for the animals. Overall, this will be an excellent way for me to prove that my understanding of algorithms and data structures has improved while obtaining my degree.

**How This Artifact Was Improved and Skills This Showcases**

During the enhancement process, I began by switching the list of dogs and monkeys to dictionaries. I also implemented unique IDs that are automatically assigned to the animal once they are entered into the system. These unique IDs can then be used when iterating through the dictionaries. This decreases the runtime complexity from O(n) to O(1), which increases the efficiency of the program. This showcases my understanding of algorithmic principles. I also implemented a helper function that ensures that duplicate animals cannot be entered into the system. This works by comparing all animal attributes, except the unique ID, since the ID will never be the same. Lastly, I implemented a helper function that ensures that if two animals of the same animal type share the same name, when reserving an animal, the user will be able to specify which animal they want to be reserved. These last two enhancements showcase my ability to thoroughly examine the functionality of a program. When initially creating the program, I overlooked that two animals of the same animal type might share the same name, and both need to be added to the system. This would be a huge oversight if it was created for an actual company since some animal names are incredibly common.

**Course Outcomes Met**

By completing this enhancement, I have met the following course outcomes:

* Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.
  + Replacing the array lists for dogs and monkeys with dictionaries, and incorporating unique IDs to search through the dictionary, decreased the runtime complexity from O(n) to O(1), and therefore increased the efficiency of the program using algorithmic principles. Incorporating a way for the program to ensure that the same animal can’t be added more than once increases data handling, following computer science practices. Also, ensuring that the appropriate animal is reserved when multiple animals of the same animal type share the same name also increases data handling.

**Reflecting on the Enhancement Process**

While completing the code for my second enhancement category, I ran into a few different issues. When implementing the unique IDs, I had to research how to implement them in a way that they would be created automatically, without any help from the user. Also, when creating the helper function for checking if the animal is unique, I initially overlooked that the unique IDs would make it so animals would never have the same exact attributes, and could therefore be entered more than once. To mitigate this, I rewrote the function so that it would ignore the unique IDs. I also realized that when reserving an animal, if two animals of the same animal type had the same name, the first animal would automatically be reserved. Therefore, I altered my original enhancement plan and implemented a helper function to mitigate this. Now, if this situation occurs, the program will output all of those animals to the user, and prompt the user to select the animal they want to reserve and then reserve that animal.