5-2 Milestone 4 Narrative

Michael P Clisbee

CS 499

Southern New Hampshire University

**Briefly describe the Artifact. What is it? When was it created?**

For Artifact three in Category Three: Databases, I selected work done in CS340 – Advanced Programming Concepts in Client/Server Development. For this course created a dynamic dashboard linking several databases in the Jupyter Notebook using .csv, .py, and .ipynb Python and Javascript language files in MongoDB. The key databases used were ProjectTwoDashboard.ipynb and Animal\_App.py, with a aac\_shelter\_outcomes.csv file. The objective was to apply database systems concepts and principles to create a client/server database application that interfaces with client-side code. This code pulled data to display geo-mapping, statistics, and other user data from a database powered by MongoDB. My enhancement plan included expanding on the current MongoDB API (application programming interface) via JavaScript to make it more user friendly, as well as providing more selection options through improving on the current code. This included enhancing CRUD (create, read, update, delete) elements of my code.

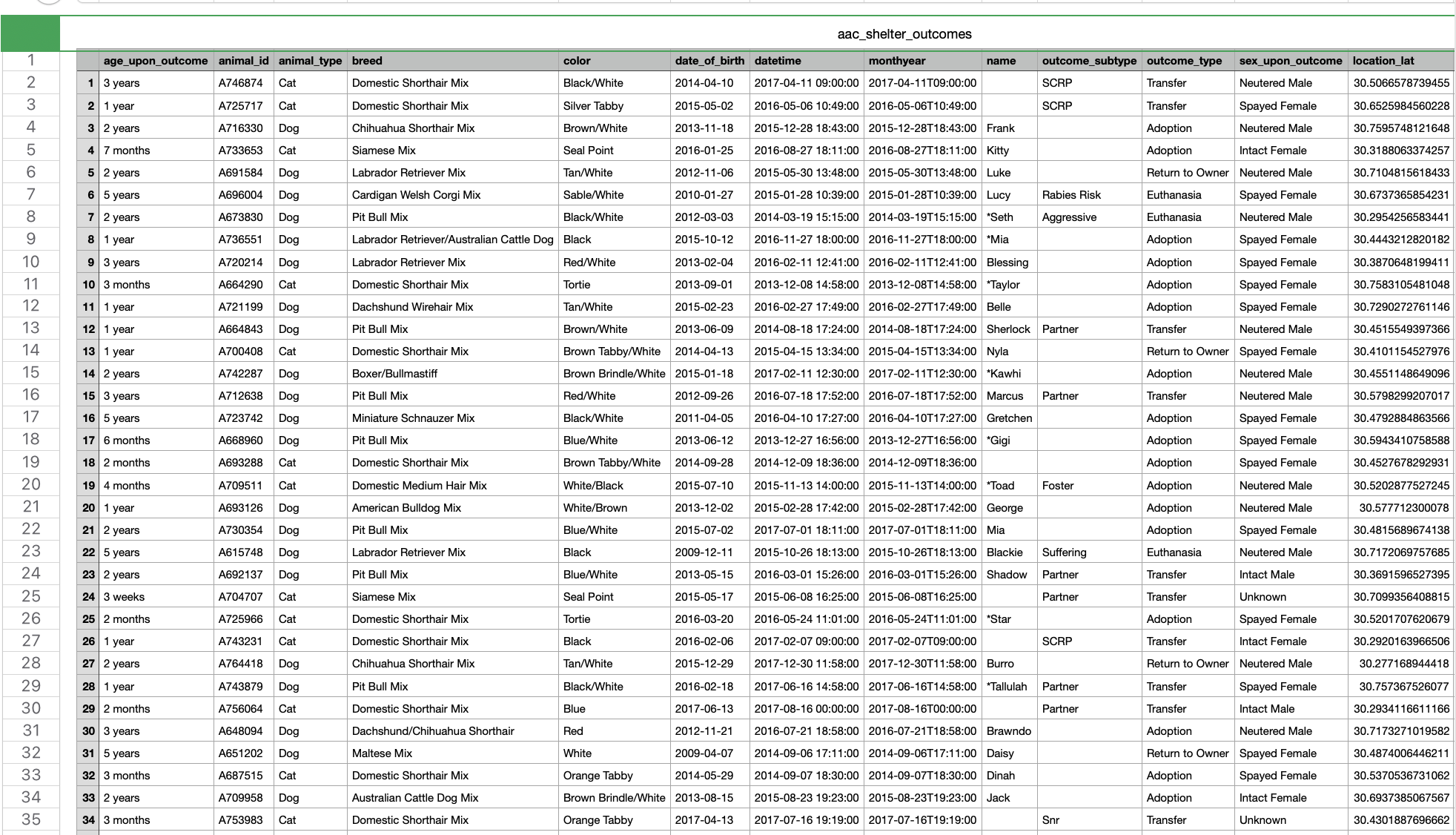
This artifact is an application the purpose of which is to provide a user-friendly interface to pull information from a database written in Python and powered by MongoDB, to retrieve information from animals, namely dogs, and update or alter same database. The objective of this project was to create a database, an API and a dynamic dashboard designed for the client to identify good dog candidates for search and rescue training. This artifact was created in the CS 340 SNHU for Advanced Programming Concepts, initially with minimal functionality and no security.

This project can be used to query MongoDB by creating and reading data from MongoDB*.* The code was altered and run via Jupyter Notebook, as well as through the XCode application. Jupyter Notebook was accessed for Mac via the Anaconda.Navigator application.

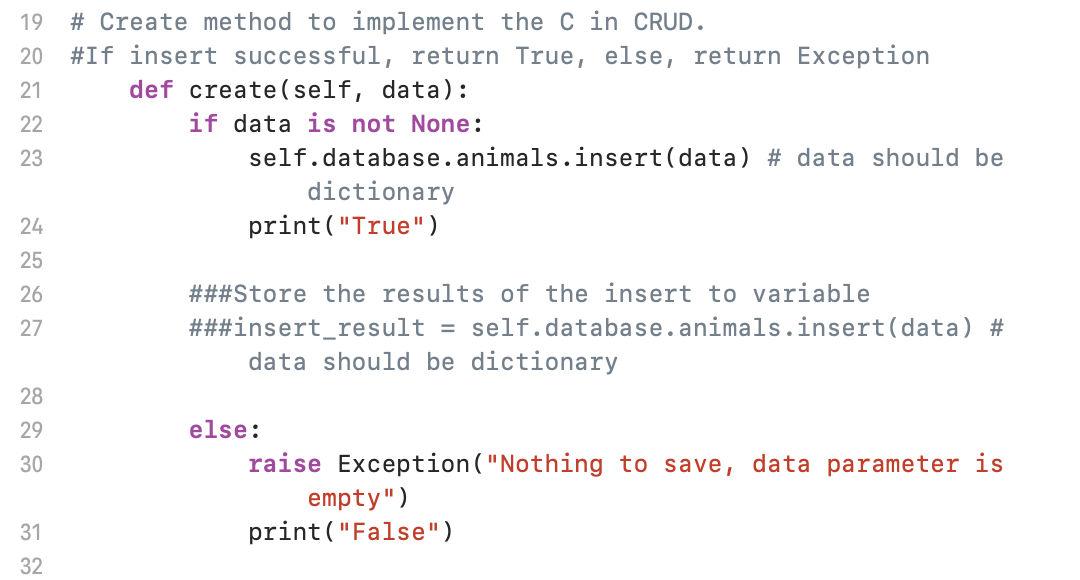
**Justify the inclusion of the Artifact in your ePorfolio. Why did you select this item? What specific components of the Artifact showcase your skills and abilities in software development? How was the artifact improved?**

This artifact represents what a fully structured database and its code can do, from an effective API and a base data file to CRUD functionality that enables the creation, readability, update capability and delete functionality, all of which are essential in creating a functional database. Some specific enhancements included additions to the password interface for added security, expanded coding for data creation and insertion into existing .csv file, expanded search functions in the ReadData operation to effectively search for existing data and display it, expanded update operations with easy-to-understand input keys to include error notifications and if/else loops, and delete functionality to include error notifications and if/else loops to eliminate existing data.

The enhancements increased functionality in several different areas as determined in the Code Review. When modifying the artifact, I did code referencing online to get ideas on how best to create a polished code. I learned several ways of writing the CRUD functionality to try and cover all possible errors, as well as how detailed the dashboard code needed to be to provide any sort of functionality.



The aac\_shelter\_outcomes.csv file is central to the mission of the database program. Here is existing data that will be either adjusted, deleted, added to, or searched through as long as the base code works as needed. Below is an example initial and final code enhancements for the C operation for C in CRUD, to create new lines of data in the .csv file:

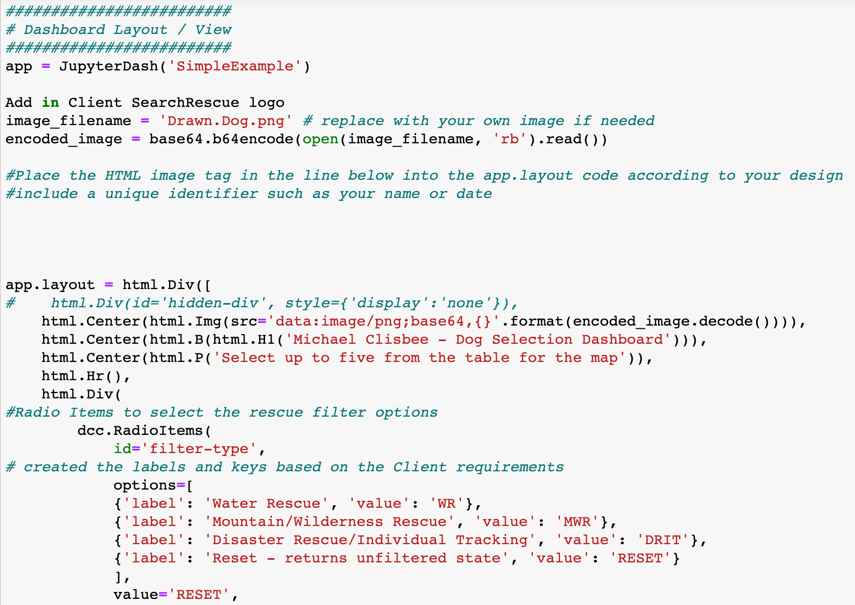
Initial code before enhancement for Create

Create code after enhancements that involved adding code to ensure user enters correct data fields. The first part of the Create code collects the data, while the second part validates and inserts results, or issues Exceptions via if/else statements.

Graphical user interface, text, application

Description automatically generated

Code adds security to application by adding username and password requirements. Still need to work out so unique passwords can be saved per username.



This is part of the final dashboard code that includes unique client logo, dashboard title and radio items to select rescue filters based on the client requirements. This code can be altered to meet client needs.

The Dashboard code is set up to collect all data and depending on data selected, create a specific pie chart and geolocation chart. This code is still being perfected, so cannot be displayed currently.

**Did you meet the course objectives you planned to meet with this enhancement in Module One?**

By fulfilling my enhancement plans, I demonstrated skills learned in the languages of Java and Python that I can apply toward improving existing code. Namely the ability to create effective test files for their respective java files to ensure they function correctly. As well, creating detailed CRUD functionality, adding Create, Read, Update and Delete functionality with safeguards included.

**Reflect on the process of enhancing and/or modifying the Artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

When modifying the artifact, I did code referencing online to get ideas on how best to create a polished code. I learned several ways of writing the CRUD functionality to try and cover all possible errors, as well as how detailed the dashboard code needed to be to provide any sort of functionality.