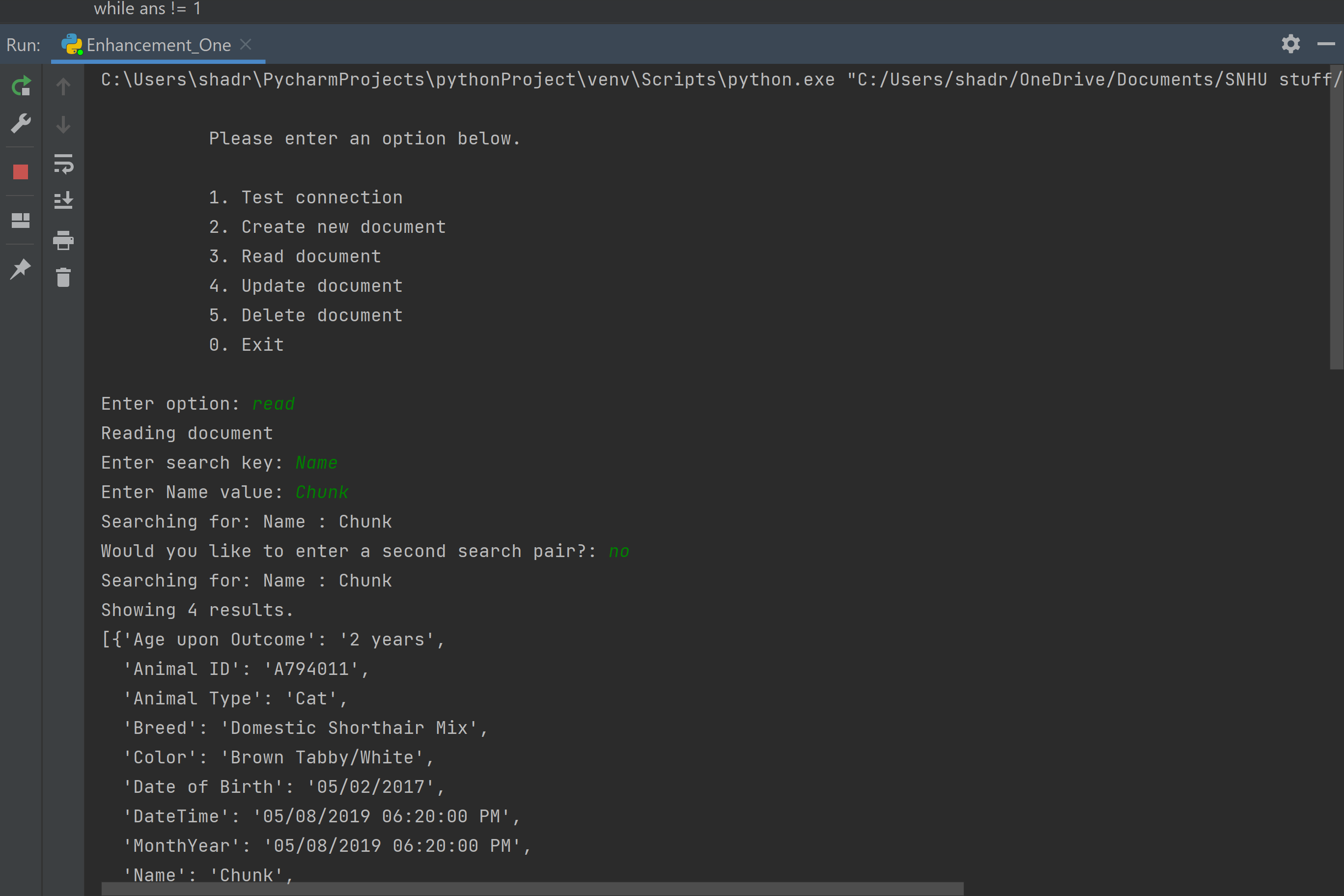
Shad Riley

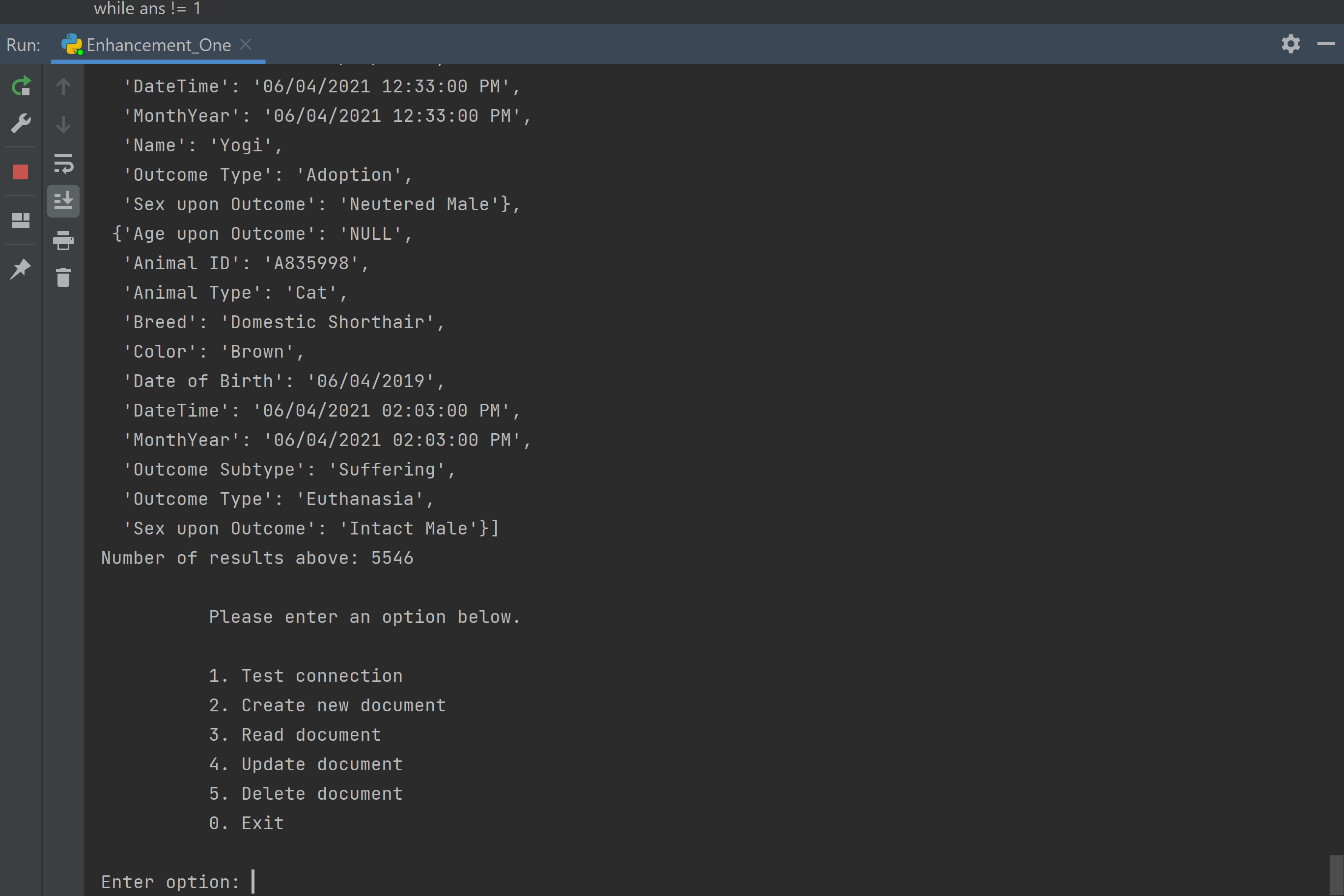
Enhancement One Narrative

CS 499

This project was created initially for the Client Server Development course CS 340. The project uses python functions to manipulate data in a MongoDB database. The modules were later incorporated into a python dash framework however I have not included the dash side of this project in with this enhancement. The enhancement I have produced creates a command line interface that the user can use to easily add, change, or remove data in the database. This is done in a python program that calls the functions. Below is an example of that command line interface.



The options on the main menu can be selected by a user who enters an option in the text box. This user input is not case sensitive and takes multiple values such as the corresponding number next to the option you want or simply the text of that option. For example, the input “read” as you can see above would access the read function. To complete this function, the user enters a key/value pair they wish to search for. This may yield a large number of documents depending on the parameters they entered. So I also added the option to search for a second key/value pair with the first one. As you can see above, the program asks if you would like to enter a second pair. If you answer yes, this will let you add a second pair. If not, it will search with the one key/value pair entered. This search could yield many results. So above and below the result output, the program displays the number of documents printed. Below is the output after searching the animals database for the color brown.

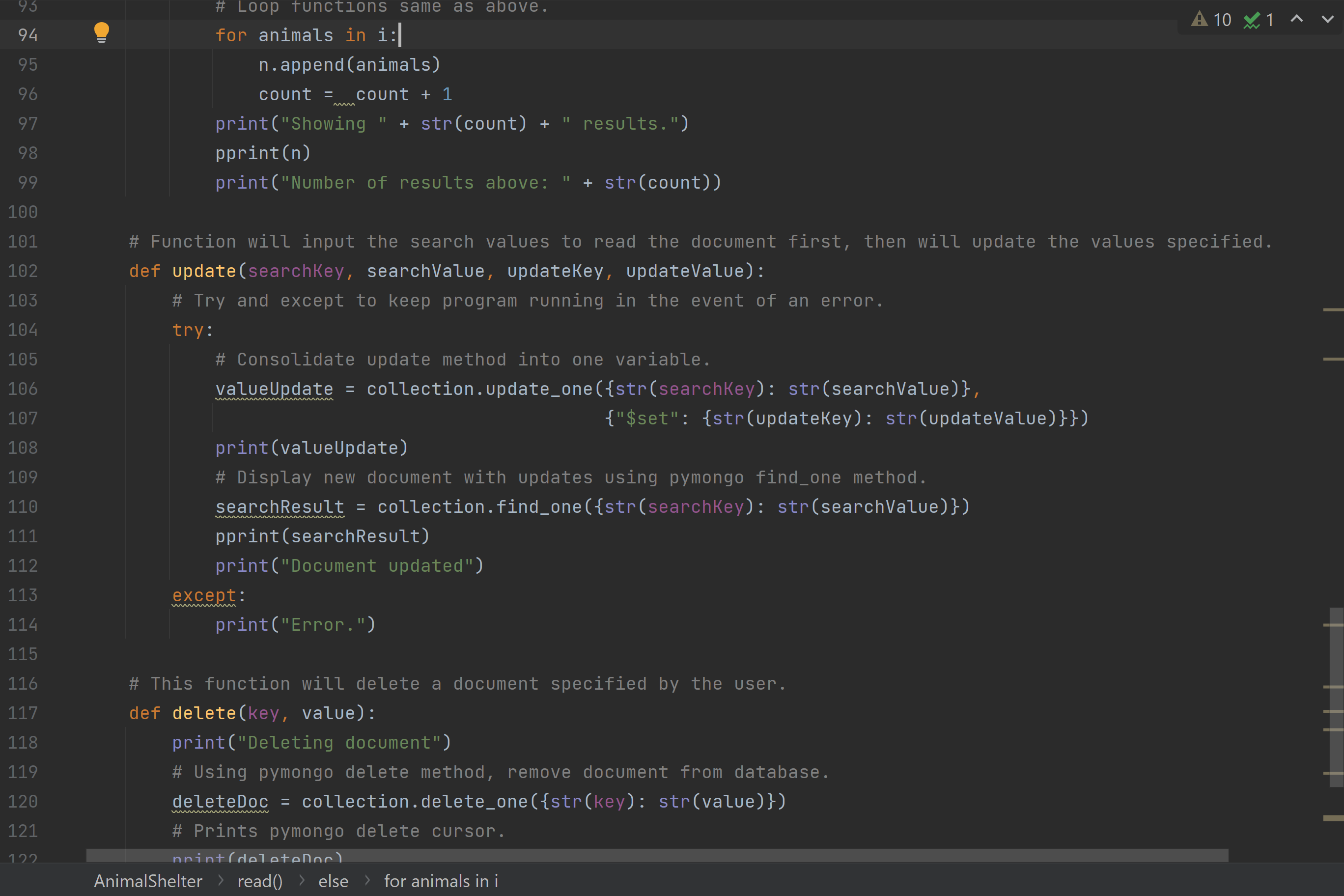


This project is an excellent example of my skills in software design and engineering. This program uses multiple levels of loops, if-else statements, error exceptions, and python functions. This project uses the pymongo distribution which allows me to manipulate the mongoDB database. Also, reviewing the code is simple as detailed annotations are made throughout the code in both the Enhancement\_One.py file and the MongoDB\_Modules.py file.

To improve the project, I built upon the python functions that already existed from the project and constructed a command line interface program that used these functions. To do this, many changes had to be made to these functions. The Enhancement\_One.py file was created from the ground up. You can see multiple layers of loops to allow the user to run this program as long as they need and verify entries they make are correct. The user controls the program through multiple prompts. The main menu is printed on the screen and accepts many entries such as “Read”, “delete”, “Create Document”, etc. This makes the program user friendly and easy to use.

Reflecting on the development process, there were absolutely some challenges and learning experiences. Let’s discuss the update function and the difficulties it created. My first iteration of this project (submitted in my CS 340 course) had a bug that I didn’t notice at first. After extensive testing, I realized I had an issue with my update function. It passed one key value pair and updated the first document it found that matched it, not necessarily the one you wanted. The difficulty was that the pymongo read and update methods were different. How did I ensure that the correct document was passed into the update function?

My second iteration required the user to use the MongoDB unique object ID, a 24 character string. This fixed the issue of updating the right document but created another problem. Entering a 24 character string was not user friendly and easily could be entered incorrectly. Now, we have an update function that uses the read function first to print all the documents that match a key value pair, or two key value pairs if the user entered them. This provides a short list of documents with Animal ID numbers (not necessarily unique for MongoDB but should be unique based on the requirements of the project). The update function then asks the user to indicate which document by entering the Animal ID number. Below is the update function from the MongoDB\_Modules.py file.



User interface from Enhancement\_One.py file.

Text

Description automatically generated