# CS 340 README Template

## About the Project/Project Title

MongoDB python CRUD operations. This project provides methods/functions to create, read, update, and delete documents within a Mongo database collection. This project is meant to act as a module to allow another application to call the methods within the AnimalShelter class to perform the CRUD operations.

## Motivation

This program was created with the intention of being a simple module that can be used by other programs or applications for CRUD implementation for the MongoDB. The module will allow for creation of a document, reading of queries to retrieve information, updating of documents in the collection, and deletion of document(s) in the collection. The module also has built in logging for easy debugging and testing. This allows for other programs to have an easily accessible, and easily modifiable, adaptable method for implementing CRUD operations. The result is a modular system that can be adapted and implemented within a wide range of applications and systems.

## Getting Started:

## Installation

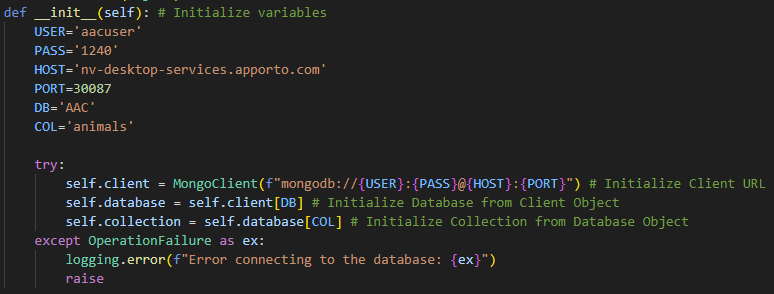
This project requires Python (preferably v. 3.11) and the installation of the PyMongo driver. The module itself is only the implementation of the class (AnimalShelter) and the methods associated with the CRUD commands. You will also need a MongoDB instance to connect to. The credentials for the MongoDB database (Username, Password, Host, Port number) are initialized in the \_\_init\_\_ method along with the database and collection to access.

## Usage

### Code Example

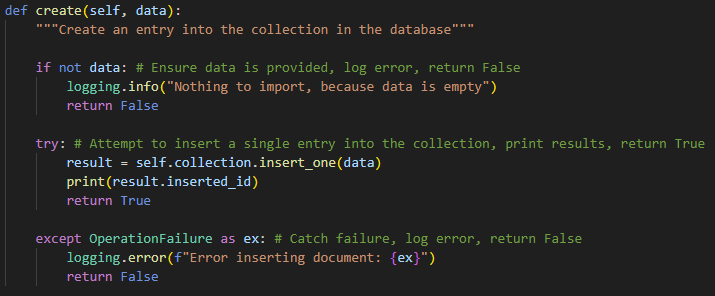
#### Connecting to a MongoDB:

To connect to a MongoDB instance, the user credentials, host, port, database, and collection must be provided in the \_\_init\_\_ method. Example:



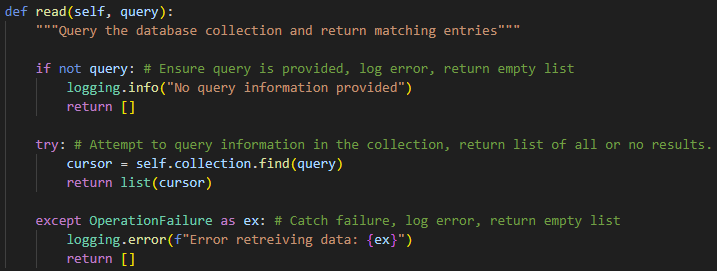
#### Insert\_One function into the collection:

To insert a single object / document into the specified collection, the create() method can be used. This method uses the “self” parameter for the collection specified in the \_\_init\_\_ method and a “data” parameter. The “data” parameter is instantiated locally upon passing; however, the parameter is not initialized. This data parameter initialization should be done in the application calling the method.



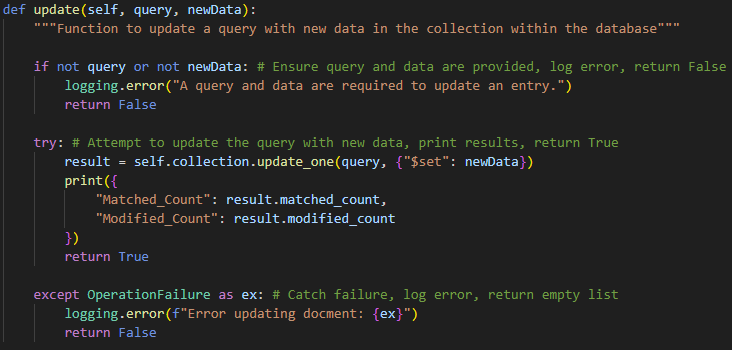
#### Query the database collection and return matching objects / documents

To query the collection, the read() method can be used. This method uses the “self” parameter for the collection specified in the \_\_init\_\_ method and a “query” parameter. The “query” parameter is instantiated locally, but not initialized upon passing. This query parameter initialization should be done in the application calling the \_\_init\_\_ method. If no “query” parameter is provided, the method will return an empty list. If a “query” parameter is provided, the method will return a list of all object/ documents that match the query (if any.)



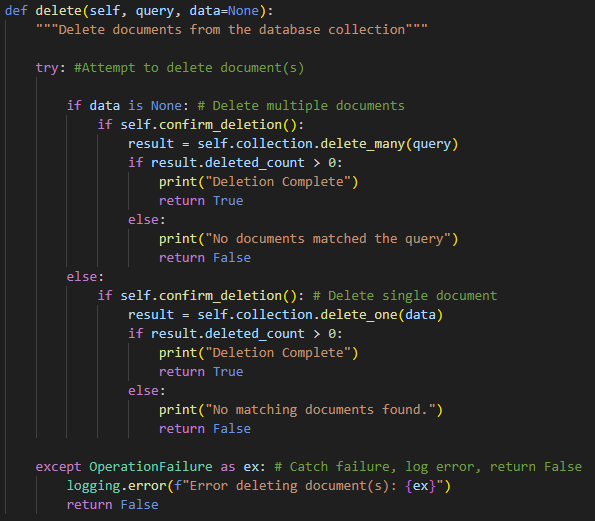
#### Update an object / document in the collection:

To update a single object / document in the specified collection, the update() method can be used. This method uses the “self” parameter for the collection specified in the \_\_init\_\_ method, a “query” parameter which will specify which object / document to update, and a “newData” parameter which will specify the data to update. The “query” and “newData” parameters are instantiated upon passing but are not initialized. These parameter initializations should be done in the application calling the method.



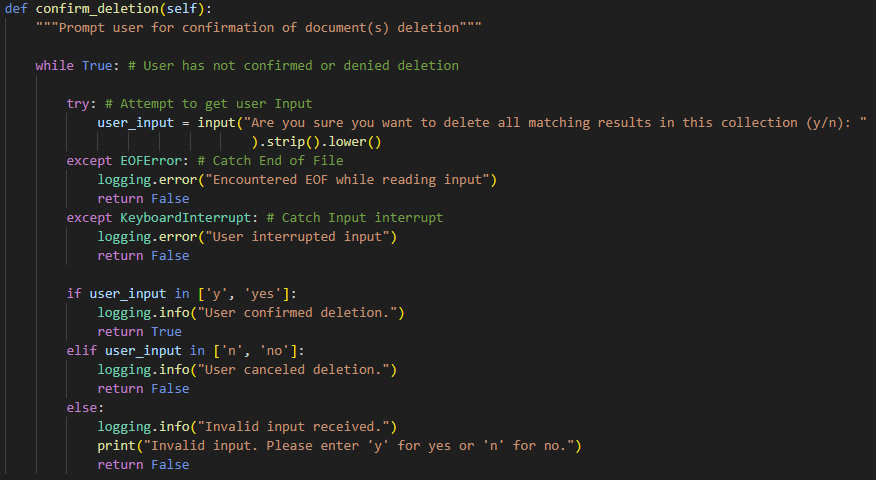
#### Delete an object / Delete many objects / documents in the collection:

To delete an object or multiple objects / documents in the specified collection, the delete() method can be used. This method uses the “self” parameter for the collection specified in the \_\_init\_\_ method, a “query” parameter which will specify the objects / document(s) to delete and an optional “data” parameter. The “query” parameter is instantiated but not initialized upon passing. The “data” parameter is instantiated and initialized upon passing to a default value of: “None.” If the data parameter value is left as default, the delete() method will perform the delete\_many function in the MongoDB, deleting all values specified by the “query” parameter *(USE WITH CAUTION.)* If the “data” parameter is not the default (None), the delete() method will perform the delete\_one function in MongoDB. The delete() method will request user confirmation before deleting any information.



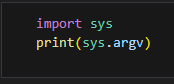
#### User Confirmation Method:

This method is used by the delete() method to confirm the user intends to delete the data. This method does not need to be called by another application, rather, it should be implemented within the other methods used in the module. The method will request the user provide basic y / n (yes or no) input to confirm the operation.



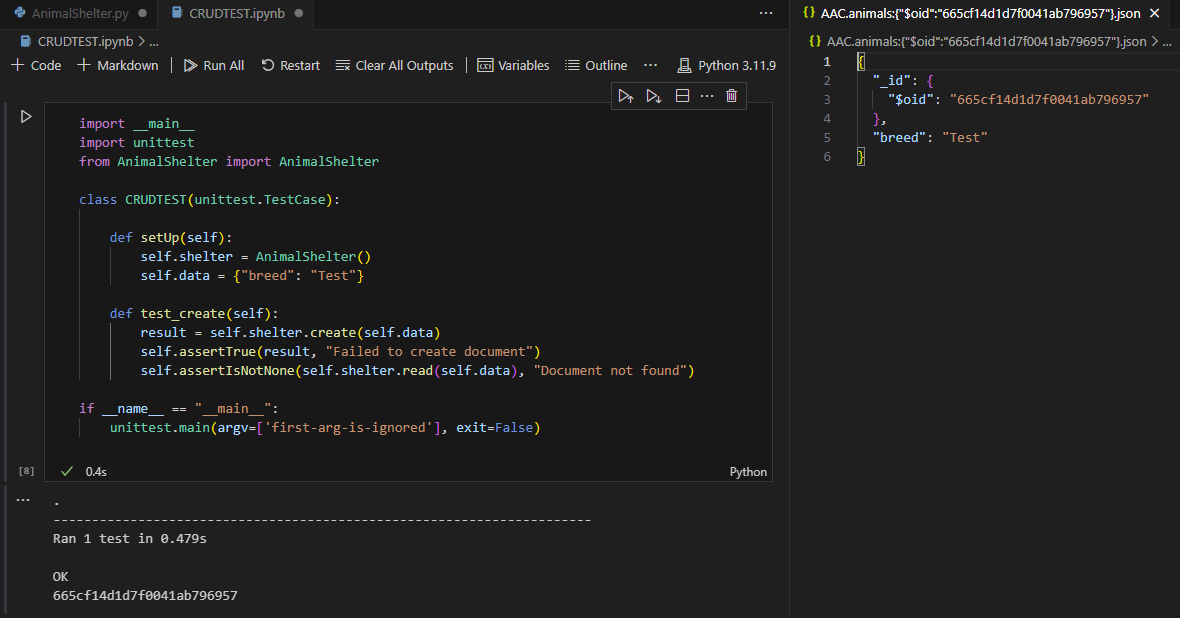
### Tests

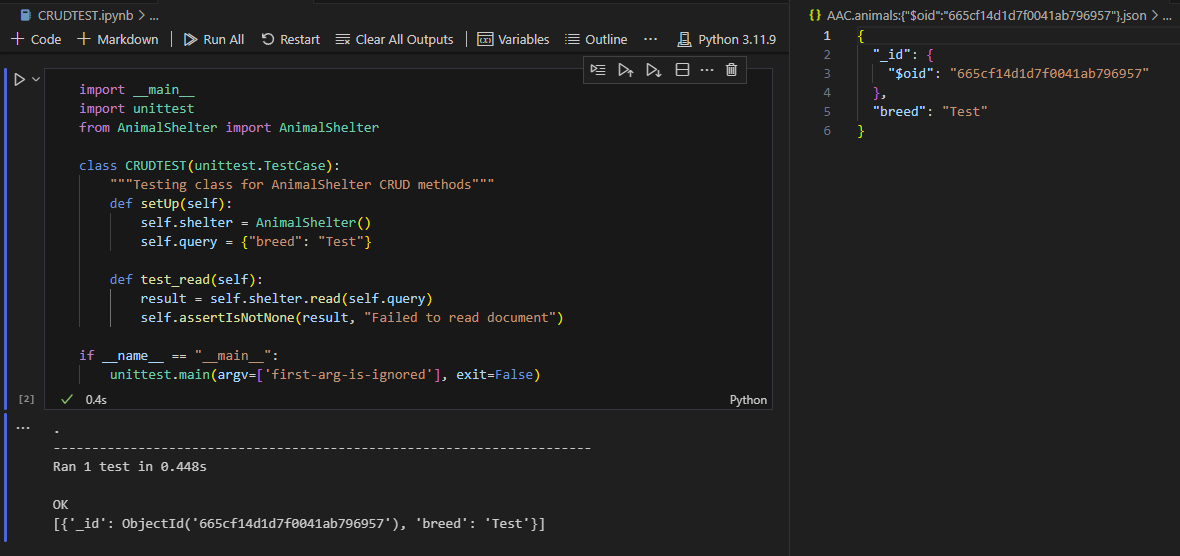
Testing can be achieved using Jupyter Notebook. \**Note:\** Jupyter Notebook does include arguments by default upon using the main() method. If you are receiving argument errors, you can check the arguments that Jupyter is passing by using:

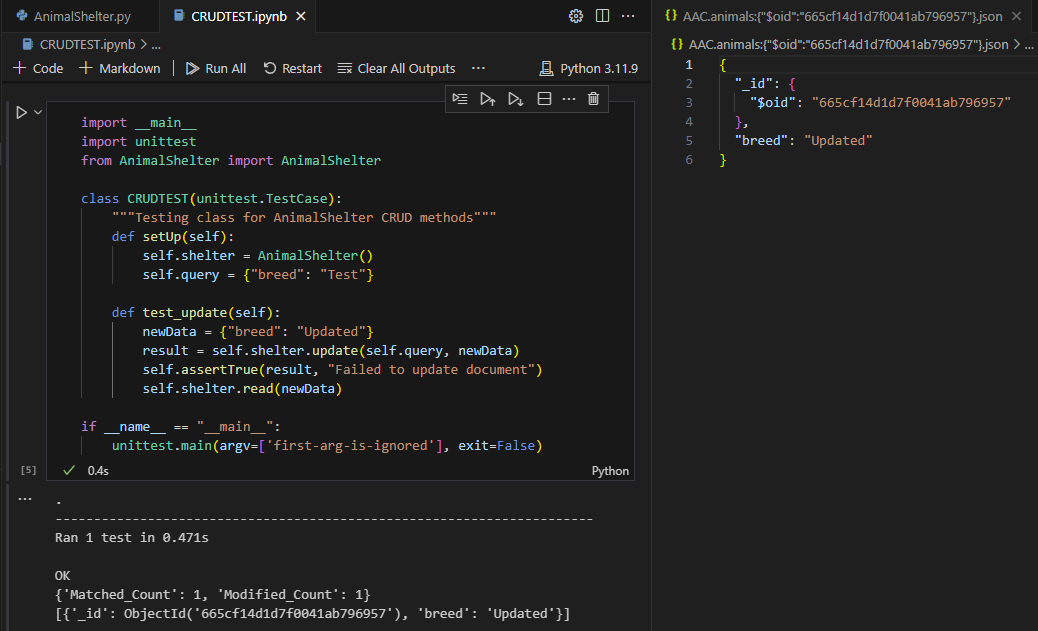
  
  
You can ignore the argument(s) passed by Jupyter using:  

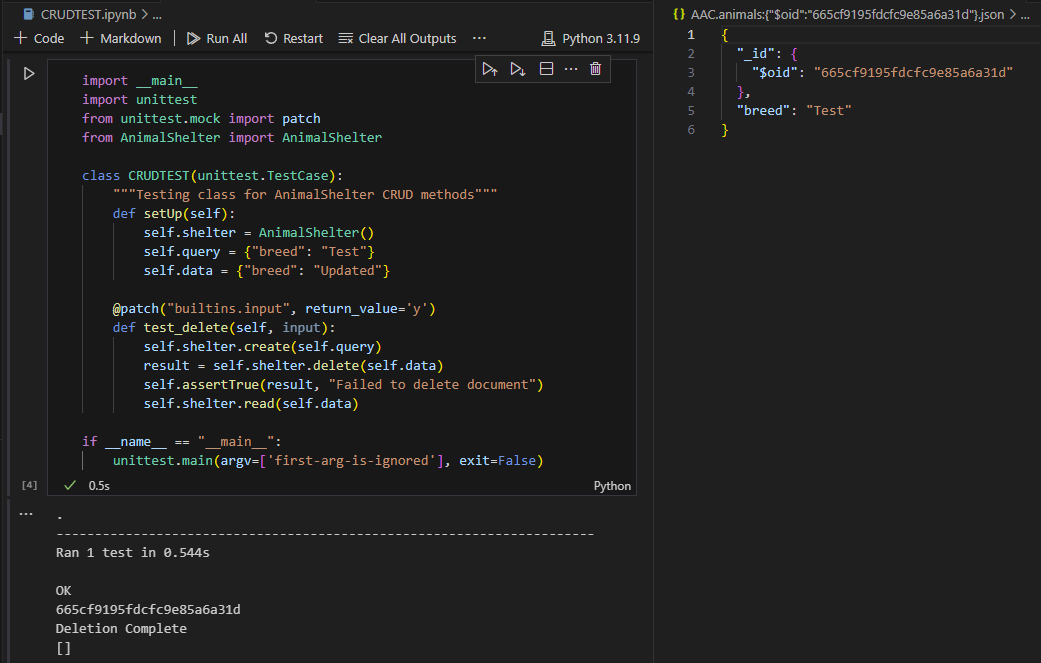

#### Create Method Testing to insert a document into the collection:

The following code can be used to test the usage of the Create method. An instance of the AnimalShelter class will need to be initialized, the “data” parameter used within the create method must also be initialized. Example with successful output:



1. Read Method Testing to retrieve document(s) from the collection:  
   The following code can be used to test the usage of the read() method. An instance of the AnimalShelter class will need to be initialized, the “query” parameter used within the read method must also be initialized. Example with successful output:  
   
2. Update Method Testing to update document in the collection:  
   The following code can be used to test the usage of the read() method. An instance of the AnimalShelter class will need to be initialized, the “query” parameter used within the update method must also be initialized. Example with successful output:



1. Delete Method Testing to delete document(s) in the collection:  
   The following code can be used to test the usage of the delete() method. An instance of the AnimalShelter class will need to be initialized, the “query” parameter and optional “data” parameter used within the delete method must also be initialized. The mock.patch module can be used to mock the user input for the confirmation of the deletion. In this example, the create() method is used to create a new document, and I am deleting the document specified in the data parameter, after the document has been deleted, the read() method is used for a secondary visual confirmation that the deletion was successful. Example with successful output:  
   

## Roadmap/Features

1. update\_Many to be added to the update() method
2. find\_One to be added to the read() method

## Contact

Joshua Lewis