# CS 340 README Project 1

## About the Project

*The AACDatabaseLayer class is a Python module that provides functionality to interact with a MongoDB database for a web-based app. It facilitates Create (C), Read (R), Update (U), and Delete (D) operations using the PyMongo library for MongoDB operations. This module aims to simplify the process of database interaction and provide developers with an easy-to-use class for storing and retrieving data in a MongoDB database.*

## Motivation

*The motivation behind creating the AACDatabaseLayer Python module was to streamline the process of database interaction for our web-based applications. MongoDB is a popular NoSQL database, and by providing a simple and efficient class, developers can focus more on building the application's features rather than handling database operations.*

## Getting Started

*To use the ‘AACDatabaseLayer’ python module for interacting with a MongoDB database, please review the following steps.*

1. *Prerequisites:*
   1. *Python* 
      1. *If you don’t have Python installed go to* [*https://www.python.org/*](https://www.python.org/)*.*
   2. *PyMongo library installed.* 
      1. *If you don’t have PyMongo Library use the following code*
         1. *Pip install pymongo*
   3. *Jupyter Notebook or any other Python environment of your choice.*
2. *MongoDB Setup:* 
   1. *You need to have MongoDB server running.* 
      1. *If you don’t have MongoDB installed, you can also use* [*https://mongodb.com*](https://mongodb.com) *to install it on your device.*
   2. *Create a database in MongoDB and name it any preference you choose, my code will have it as “AAC”.*
   3. *Set up authentication for the MongoDB server and create a user with read/write privileges for the “AAC” database.*
3. *Importing the Module:*
   1. *Download or copy the ‘AACDatabaseLayer’ class into your Python environment.*
4. *Types of Imports you’ll need*
   1. *Import Os*
   2. *Import sys*
   3. *Import pprint*
   4. *From pymongo import MongoClient*
   5. *From bson.objectid import Object ID*

***Attributes and Working Functionality of CRUD Operations:***

*create: This method creates a new document in the specified collection with the provided data. The collection parameter is a string representing the name of the collection, and the data parameter is a dictionary containing the data to be inserted.*

*read: This method retrieves documents from the specified collection based on the provided filter. The collection parameter is a string representing the name of the collection, and the filter parameter is a dictionary containing the filter criteria.*

*update: This method updates documents in the specified collection based on the provided query and data.*

*delete: This method removes documents from the specified collection based on the provided query. The collection parameter is a string representing the name of the collection, and the query parameter is a dictionary containing the filter criteria for the documents to be deleted.*

*count\_all\_documents(collection): This method returns the count of all documents present in the specified collection. The collection parameter is a string representing the name of the collection.*

## Installation

*This Module requires the following tools:*

* *Python (version 3.9)*
* *PyMongo library*
* *Jupyter Notebook or any python environment of your choosing.*

## Usage

*Below you will see examples of screenshots of sample code, tests, and how they ran to demonstrate how to use the AAC database module for Create (C) and Read (R) operations.*

### Code Example

1. *Create your class, contructor and connect to MongoDB and set up current query options.*
   1. *Create Class definetiion accessing & connecting to MongoDB:*

*Class AACDatabaseLayer(object):*

* 1. *Constructors:*

*def \_\_init\_\_(self, HOST, PORT, username, password) :*

*self.host = HOST*

*self.port = PORT*

*self.username = username*

*self.password = password*

*# Store all credentials in a single URI*

* 1. *Connect to MongoDB:*

*def connect(self, logging=False) :*

*self.connection = MongoClient(self.uri)*

*create if statement and print the databases in MongoDB*

* 1. *Set a current MONGO DB for query options:*

*def setDatabase(self, database) :*

*self.db = self.connection[database]*

1. *Performing CRUD Operations:*
   1. *You can now use the ‘AACDatabaseLayer’ instance to perform CRUD operations on your MongoDB database:*
      1. *Create (C): ‘myConnection.create(collection, data)’*
      2. *Read (R): ‘myConnection.read(collection, filter)’*
      3. *Update (U): ‘myConnection.update(collection, query, data)’*
      4. *Delete (D): ‘myConnection.delete(collection, query)’*
      5. *Count All Documents: ‘myConnection.count\_all\_documents(collection)’*
2. *Setting up Environment Variables:*
   1. *Before running the module, set the required environment variables for MongoDB connection parameters(HOST, PORT, username, and password). Example*

*os.environ["MONGO\_HOST"] = "your\_mongo\_host"*

*os.environ["MONGO\_PORT"] = "your\_mongo\_port"*

*os.environ["MONGO\_USER"] = "your\_mongo\_username"*

*os.environ["MONGO\_PASS"] = "your\_mongo\_password"*

*You’ll want to make sure that you have authentication set up for your MongoDB server.*

1. *Verify that the environment variables worked:*
   1. *Print(Port, User, PASS, HOST)*
2. *Creating an Instance:*
   1. *Create an instance of the ‘AACDatabaseLayer’ class, providing the MongoDB connection parameters:*

*myConnection = AACDatabaseLayer(HOST, PORT, username, password)*

1. *Connecting to MongoDB:*
   1. *Use the ‘connect()’ method to establish a connection to the MongoDB server:*

*myConnection.connect(logging=True) # Set logging to True for debug output*

1. *Setting the Database:*
   1. *Use the ‘setDatabase()’ method to specify the desired database(example: “AAC”)*

*myConnection.setDatabase('AAC')*

1. *Example of reading documents from ‘animals’ collection:*

*filter\_query = {"animal\_id": "A728994"}*

*cursor = myConnection.read('animals', filter\_query)*

*for doc in cursor:*

*print(doc)*

1. *Example of a new document in the animals collection:*

*createData = {*

*"rec\_num": 1,*

*"age\_upon\_outcome": "3 years",*

*"animal\_id": "A746896",*

*"animal\_type": "Dog",*

*"breed":"Labrador",*

*"color":"Cream",*

*"date\_of\_birth": "10-11-2015" ,*

*"datetime": "2017-04-11 09:00:02" ,*

*"monthyear": "017-04-11T09:00:50" ,*

*"outcome\_subtype": "SCRP" ,*

*"outcome\_type": "Transfer" ,*

*"sex\_upon\_outcome": "Neutered Male" ,*

*"location\_lat": "30.5066578739495" ,*

*"location\_long": "30.5066578739698" ,*

*"age\_upon\_outcome\_in\_weeks": "156.767857142857" ,*

*}*

*myConnection.create('animals', new\_data)*

1. *Example of updating a document in the “animals” collection:*

*update\_query = {"animal\_id": "A728994"}*

*update\_data = {"age\_upon\_outcome ": “4 years”, "outcome\_Type": “Return to Owner”}*

*result = myConnection.update('animals', query, data)*

*print("Count of documents updated:", result)*

1. *Example of deleting document in the ‘animals’ collection:*

*query = {"animal\_id" : "A746896"}*

*deleted\_count = myConnection.delete('animals', {'animal\_id' : 'A746896'} )*

*print(str(deleted\_count) + " Document(s) successfully deleted.")*

1. *Verify your Results:*
   1. *To ensure that CRUD operations are successful, you can use the ‘read()’ and count\_all\_documents() methods or MongoDB shell commands to verify the data in the database.*
      1. *Count all documents example:*

*total\_document\_count = myConnection.count\_all\_documents('animals')*

* + 1. *Read documents example:*

*cursor = myConnection.read('animals', {})*

*print(cursor)*

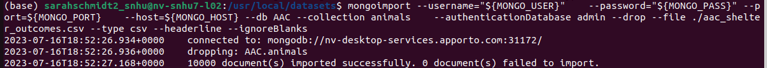
*for index, doc in enumerate(cursor):*

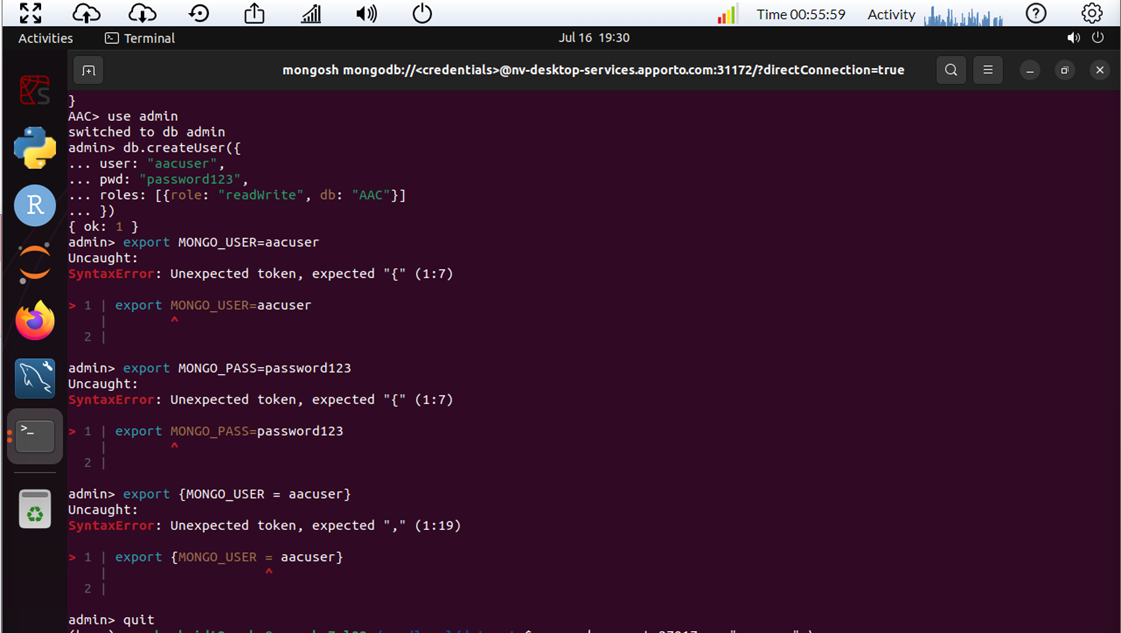
*print('\nDoc ' + str(index) + ':')*

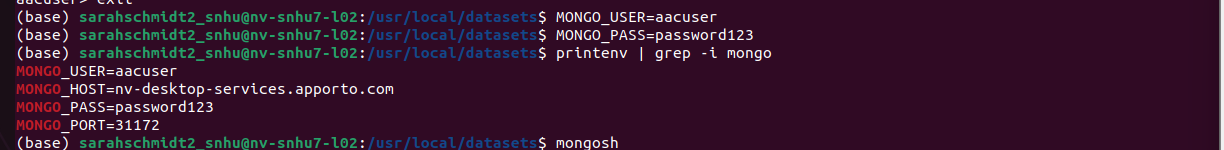
*for prop in doc:*

*pprint.pprint(str(prop) + ": " + str(doc[prop]))*

### Screenshots

MongoImport of AAC Animals:

MongoDB Authentication



*2 screenshots of the MyClassFile.py*

*A screenshot of a computer

Description automatically generated*

*A computer screen shot of a code

Description automatically generated*

*Screenshots of setting up DB*

*A screenshot of a computer

Description automatically generated*

*Screenshots of Connecting to DB, counting all documents and then reading the files*

*A screenshot of a computer program

Description automatically generated*

*Screenshots of testing the Create method, counting new docs and reading them out*

*A computer screen shot of a computer code

Description automatically generated*

*Screenshots of the test execution from above*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer

Description automatically generated*

*Screenshots of testing the update method with the count amount, and documents read*

*A screenshot of a computer program

Description automatically generated*

*Screenshots of the test execution of testing the update method with the count amount, and documents read*

*A white background with black text

Description automatically generated*

*A screenshot of a computer code

Description automatically generated*

*Screenshots of the testing of the delete method along with the new count, reading out the documents. With the test execution*

*A screenshot of a computer program

Description automatically generated*

## Contact

Sarah Schmidt