# CS 340 README Template

## About the Project/Project Title

This module is a CRUD for a MongoDB database. It imports a database, then allows you to create an object, read an array of objects, update an object, and delete an object. It uses a Python API from MongoDB to do this.

## Motivation

This module was designed for Grazioso Salvare to search through a database to find acceptable dogs for search-and-rescue from AAC.

## Getting Started

To use the module, you need to have a database uploaded into MongoDb. This is how you would do it from a csv file. The first input in the screenshot below is the command, and you would just fill in the command with your specific information.

A screenshot of a computer

Description automatically generated

After importing a database, you need to set up a user account that you want to connect. You need to be on a user that has admin permissions, and then input the following, changing user and password to whatever you want for them.

A screenshot of a computer

Description automatically generated

Once you do this you can run db.getUsers() from the admin to ensure that you have properly created the user.

A screenshot of a computer

Description automatically generated

To use this module, copy the file to the same folder as your project.

Then just include it in your project like this:



In CRUD\_AnimalShelter.py you also must enter your information for the database in the following section:

A screen shot of a computer

Description automatically generated

## Installation

This module was designed to be used in a Jupyter notebook, but it can be used with any IDE that supports Python. If you would like to use Jupyter notebooks and do not already have it installed, you can get it [here](https://jupyter.org/install) and follow the instructions on the page.

This module also uses a couple of libraries: pymongo and bson.objectid

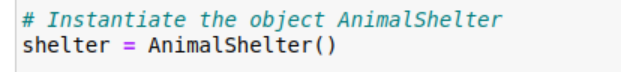
PyMongo allows us to connect to a database and be able to perform the CRUD operations.

Bson.objectid allows us to create a unique id for our objects.

## Usage

### Code Example

To use this class, first instantiate an object.

**To use the create function the data must be in a dictionary structure.

A white background with blue text

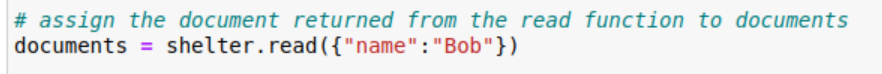
Description automatically generated

When creating a entry to the database, the function will return true if it is successful, and false otherwise, so a if else statement can be used:

A screen shot of a computer code

Description automatically generated

The read function returns an array of the found documents, so they can be assigned to a variable for later use or printing. The argument must be in a dictionary structure.



The update function returns the number of objects that were updated.

This function takes 2 arguments: the first is the dictionary to search for in the database, and the second is a dictionary that you want to change in the search.

A screenshot of a computer code

Description automatically generated

The delete function returns the number of objects that were deleted.

It takes one argument: a dictionary to search for in the database, and deletes everything with that dictionary.

A screenshot of a computer code

Description automatically generated

### Tests

To test the read function, you can set up a simple if / else statement to check if the function returned anything and print it out.

A white background with text

Description automatically generated

The testing of the create function is above as well.

The update function was tested with object that I know would return 0, 1, and 2 to test all the branches in the update function. The instance of returning 1 is above, and here are 0 and 2.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

I tested the functionality of delete above, but wanted to make sure it worked when nothing was found with the argument provided.

A screenshot of a computer code

Description automatically generated

## Roadmap/Features (Optional)

Need to figure out how to fix the hard coded animals collection in the create method to use the self.collections variable.

Probably should change the name of the class to something more generic since it will be able to work for any MongoDB database.

## Contact

Your name: Brian Engel