# Project Two: CS 340 README

## Grazioso Salvare

*This open-source program provides read and write CRUD functionality to Grazioso Salvare database.*

## Motivation

*This specific program was created for Grazioso Salvare but can be modified for any database connecting to Mongo DB. Grazioso Salvare is an international rescue-animal training company. This program helps provide data to a nonprofit near Austin, Texas who will partner with Grazioso Salvare to help categorize and identify available dogs to place successfully into training.*

## Getting Started

*To get a local copy up and running, follow these example steps below.*

## Installation

*The tools you will need include:*

* *VirtualLab (Apporto) \*Note: This was previously configured via SNHU*
* *Linux Terminal \*Note: This was previously configured via SNHU*
* *MongoDB \*Note: Previously configured via SNHU*
  + *To install:* [*https://www.mongodb.com/docs/manual/installation/*](https://www.mongodb.com/docs/manual/installation/)
    - *Once Mongo DB is installed, follow steps from the documentation to start the MongoDB shell*
    - *To see all databases type:*
      * *show dbs*
    - *Here you should see a database called “admin”. To use the admin database (this will ensure you have privileges to create a new user) type:*
      * *use admin*
    - *To create the user type:*

A close-up of a text

AI-generated content may be incorrect.

*NOTE: This above example uses example values. The role will enable read & write permissions, this value SHOULD NOT be changed. The user, pwd, and db can be any values. Also be sure to double check the command formatting.*

* *Jupyter Notebook*
  + *https://jupyter.org/install*
* *Spyder IDE*
  + *https://docs.spyder-ide.org/current/installation.html*

## Purpose of CRUD Python Module

*The purpose of the CRUD Python module is to provide an easy way to interact with MongoDB using Python code to perform basic Create, Read, Update and Delete (CRUD) operations on a collection in the database. This example uses the “animals” collection. These operations help to manage the data in an efficient way.*

***Create****: Adds a new document to the specified collection*

***Read****: Retrieves or “gets” a document or list of documents based upon filtering*

***Update****: Modifies a document based on specified filter*

***Delete****: Removes document(s) from the database collection*

**Using the CRUD Module**

*In order to utilize the CRUD Module, an instance of MongoDB should be run with the Python “pymongo” driver. Once the module is set up, the user can create an instance of the CRUD class (in this example, the class name is AnimalShelter.py). See section below for more details.*

## More Details On Usage for Create and Read

*To start, open Spyder IDE and create a new Python file. For example, the file name could be something like:*

**

*Note: The filename must end with .py*

*Insert the following text under* ***Code Example*** *into the file.*

*Note: Be sure to change the following variables to the variables you will use for your database:*

*USER = ‘yourUsername’*

*PASS = ‘yourPassword’*

*HOST = 'nv-desktop-services.apporto.com'*

*PORT = 12345 \*\*enter desired port \*\**

*DB = 'AAC'*

*COL = 'animals'*

*You can also check your individual mongo environmental variables using the following command:*

*printenv grep | -i mongo*

### Code Example (this will act as the CRUD module)

*# -\*- coding: utf-8 -\*-*

*"""*

*This script will help read and write documents from Mongo DB*

*"""*

*from pymongo import MongoClient*

*from bson.objectid import ObjectId*

*class AnimalShelter(object):*

*""" CRUD operations for Animal collection in MongoDB """*

*def \_\_init\_\_(self, USER, PASS):*

*# Initializing the MongoClient. This helps to access the MongoDB databases and collections.*

*# This is hard-wired to use the aac database, the animals collection, and the specified user.*

*# Note: USER and PASS are parameters above. These will likely be from your environment*

*# variables.*

*# Definitions of the connection string variables are*

*# unique to the individual Apporto environment.*

*#*

*# You must edit the connection variables below to reflect*

*# your own instance of MongoDB!*

*#*

*# Connection Variables*

*HOST = 'nv-desktop-services.apporto.com'*

*PORT = 12345*

*DB = 'AAC'*

*COL = 'animals'*

*#*

*# Initialize Connection*

*#*

*self.client = MongoClient('mongodb://%s:%s@%s:%d' % (USER,PASS,HOST,PORT))*

*self.database = self.client['%s' % (DB)]*

*self.collection = self.database['%s' % (COL)]*

*print("Connected successfully.")*

*# Create method to implement the C in CRUD.*

*def create(self, data):*

*if data is not None:*

*self.database.animals.insert\_one(data) # data should be dictionary*

*return True;*

*else:*

*raise Exception("Nothing to save, because data parameter is empty")*

*return False;*

*# Method to implement the R in CRUD.*

*def read(self, data):*

*if data is not None:*

*readResult = self.database.animals.find(data) # data should be dictionary*

*return readResult;*

*else:*

*raise Exception("Nothing to read, because data parameter is empty.")*

*\*\*\*\*\*\*\*\*\*\*\*STOP CODING HERE\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*Test your script for read and write. Ensure they are working correctly. See more details below.*

*Once you have confirmed the functionality of both, add the following code for update and delete*

*\*\*\*\*\*\*\*ADD THIS CODE FOR UPDATE & DELETE TO THE CRUD MODULE AFTER CONFIRMING FUNCTIONALITY FOR CREATE AND READ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*# Update method to implement the U in CRUD.*

*def update(self, data, update\_data):*

*if data is not None and update\_data is not None:*

*try:*

*# read before update*

*docs = self.read(data)*

*if len(docs) > 1 :*

*result = self.database.animals.udpdate\_many(data, {"$set": update\_data})*

*elif len(docs) == 1:*

*result = self.database.animals.update\_one(data, {"$set": update\_data})*

*else:*

*return {"message": "No matching documents found to update."}*

*if result.matched\_count > 0:*

*return {"Success": True, "modified\_count": result.modified\_count}*

*else:*

*return {"message": "No matching documents found to update."}*

*except Exception as e:*

*raise Exception(f"Error updating data: {str(e)}")*

*else:*

*raise Exception("Query or update\_data is empty.")*

*# Delete method to implement the D in CRUD*

*def delete(self, data):*

*if data is not None:*

*try:*

*# Delete the matching document(s)*

*result = self.database.animals.delete\_one(data)*

*if result.deleted\_count > 0:*

*return {"success": True, "deleted\_count": result.deleted\_count}*

*else:*

*return {"message": "No matching documents found to delete."}*

*except Exception as e:*

*raise Exception(f"Error deleting data: {str(e)}")*

*else:*

*raise Exception("Query is empty.")*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF CODE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*\*\*\*NOTE\*\*\*\*\*\**

*In the update function, read before writing (or updating is called to ensure if an object has been recently updated, we are first retrieving the updated object and the performing an update. If this code is utilized for large scale projects, consider adding threading. This aspect will not be discussed here as it is out of scope.*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

### Tests for Create and Read

*To create a python testing script in Jupyter Notebook:*

1. *Open Jupyter Notebook, this will start a kernel instance browser session.*
2. *Create a new Python3 (ipkernel) file in the same folder where you saved your .py file from above. If not, you will need to specify the filepath to the CRUD module you created with the .py file later on when specifying imports.*

A white text box with black lines

AI-generated content may be incorrect.

1. *Click File -> Save as… -> Documents/Assignment4/testScriptfile (Note: No need to put a file extension here, only use the name. By default this file will be saved as a .ipynb file)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*After clicking save, you will see in the browser the filename extension such as:*

**

*Tests for UPDATE & DELETE can be found in the Proof of Functionality Section*

*Note: If you update and object attribute, you may need to update your script to perform deletes or other updates.*

1. *TDLR (for testing script):*

*Copy and paste the following text:*

*from animalShelterModule import AnimalShelter*

*CRUD = AnimalShelter("userName", "password")*

*created = CRUD.create({"animal\_type": "Dog"})*

*print (created)*

*readResult = CRUD.read({"animal\_type": "Dog"})*

*#convert cursor to a list*

*readResultList = list(readResult)*

*print(readResultList)*

*#loop through documents and print results*

*for r in readResultList:*

*print(r)*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*Example test script with explanations*

*#importing the CRUD MODULE and the class name from the module.*

*# here the class name is Animal Shelter.*

**

*The import will follow the structure:*

*From <filename> import <className>*

*IN PYTHON SCRIPT:*

*from animalShelterModule import AnimalShelter*

*#Instantiate the CRUD object with the required arguments. Note self is not required as it is #part of the user session.*

*IN PYTHON SCRIPT:*

*CRUD = AnimalShelter("username", "password")*

*#write to the database, note this will return True if successful, false otherwise*

*IN PYTHON SCRIPT:*

*created = CRUD.create({"animal\_type": "Dog"})*

*#Prints the object that was written*

*IN PYTHON SCRIPT:*

*print (created)*

*# Read from the database*

*# Note this will return an empty list [] if no results are a match*

*# If this search term is broad, many results may be returned*

*IN PYTHON SCRIPT:*

*readResult = CRUD.read({"animal\_type": "Dog"})*

*# Note: Mongo DB will return results using Mongo Cursor*

*# This will help convert results to view an empty list easily*

*IN PYTHON SCRIPT:*

*# convert cursor to a list*

*readResultList = list(readResult)*

*# print the result*

*IN PYTHON SCRIPT:*

*print(readResultList)*

*IN PYTHON SCRIPT:*

*# loop through documents and print results*

*for r in readResultList:*

*print(r)*

1. *To run the script, click “Run”. If file changes are made to the .py file (from Code Example above), you will need to restart the kernel for the browser to pick up the changes. This can be done by clicking the button that looks like the fast forward symbol as shown below.*

*A screenshot of a computer

AI-generated content may be incorrect.*

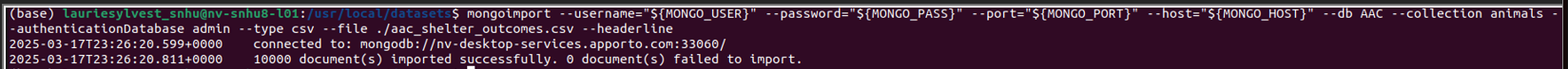
1. *If configured properly, the output should show:*

A screenshot of a computer

AI-generated content may be incorrect.

### Proof of Functionality

1. *MongoDB import execution*



1. *User authentication execution*

A screenshot of a computer program

AI-generated content may be incorrect.

1. *CRUD functionality test execution*
   1. *CREATE & READ*

A screenshot of a computer

AI-generated content may be incorrect.

* 1. *UPDATE*

A screenshot of a computer

AI-generated content may be incorrect.

*Note:*

*A screenshot of a computer code

AI-generated content may be incorrect.*

* 1. *DELETE*

*A screenshot of a computer program

AI-generated content may be incorrect.*

*Note:*

*A white background with blue text

AI-generated content may be incorrect.*

**Additional Screenshots showing functionality (Happy Path):**

**A screenshot of a computer

AI-generated content may be incorrect.**

### A screenshot of a computer AI-generated content may be incorrect.

**Additional Screenshots showing functionality (Unhappy Path):**

1. **Read (A.k.a. Search) For Document Not Existing, Connects to database, but output is empty list**

**A screenshot of a computer program

AI-generated content may be incorrect.**

1. **Create Document with invalid syntax**

**Note: Connected to Database, but create does not return true**

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Update document that doesn’t exist returns message for no matching document**

**A screenshot of a computer program

AI-generated content may be incorrect.**

1. **Update document with incorrect data field on read before writing**

**A screenshot of a computer program

AI-generated content may be incorrect.**

1. **Delete document doesn’t exist returns no matching document message**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Wrong username**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Wrong Password**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Frontend Usage**

1. **Grazioso Salvare logo links to SNHU homepage**

**A close-up of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Present unique identifier in header to credit dashboard creator**

**A black background with white text

AI-generated content may be incorrect.**

1. **Filter options to filter results for desired dog rescue type (include additional filter for reset)**

**A table with text on it

AI-generated content may be incorrect.**

* 1. **Water**

**A screen shot of a computer

AI-generated content may be incorrect.**

**(\*\*Note there a few dog types for water rescue training, but the only types present in the database which are ‘Intact Female’ and meet the desired age requirements are Labrador Retriever Mix)**

* 1. **Mountain**

**A screenshot of a computer

AI-generated content may be incorrect.**

* 1. **Disaster**

**A screen shot of a computer

AI-generated content may be incorrect.**

* 1. **Reset filters**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Data table responds to dynamic filters**

**(example -> filter for ‘black’)**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Geolocation chart**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

1. **Pie chart that responds to dynamic filters and filter rescue buttons**

**For example, all dog data (\*Note dog breed counts < 100 categorized as other for easy viewing)**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Filter For Water Type Rescue (Note: The breeds here are by count because the database values for the filters are less than 100 results)**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Filter For Mountain Type Rescue (Note: The breeds here are by count because the database values for the filters are less than 100 results)**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Filter For Disaster Type Rescue (Note: The breeds here are by count because the database values for the filters are less than 100 results)**

**A screenshot of a computer

AI-generated content may be incorrect.**

## Contact: Laurie Sylvester