# CS 340 README Template

## About the Project/Project Title

*The application will allow a user to access a database of animals. This will allow the user to create, read, update, and delete (CRUD) animals from this database.*

## Motivation

The use of CRUD operations is common when using any database. The formatting of these queries and handling of any errors can be cumbersome. In order to facilitate easier operations, they have been set into a simple *get* and *set* method.

## Getting Started

*The access this program, you must:*

1. *Open and enter MongoDB*
2. *Import the appropriate CSV files (‘acc\_shelter\_outcomes.csc’)*
3. *Create an index to parse the data stored in this database*
4. *Create an Admin account, as well as a accuser account*
5. *Install python and run programs from notebook*

## Installation/Pre-Conditions

* *The current version of Python is needed for both the .py and the .ipynd files to run*

*MongoDB is needed to access the database*

* *You will also need the database collection, available in MongoDB, example shown in CSV file below.*

A screenshot of a computer

Description automatically generated

## *User account along with permissions. See below for example of creating an account*

## Usage

In order to use this abstract, the following methods are supplied.

* These properties are available in the class to help understand interactions.
  + records\_update
  + record\_matched
  + records\_deleted
* Constructor (username, password)
  + This will facilitate initializing the MongoDB server.
* createRecord(data)
  + createRecord allows for creation of the record within the argument. This must be formatted according to Pymongo API.
    - True will return if creation was successful.
    - False will return if the action was unsuccessful.
* deletRecord(query)
  + deleteRecord searches the database for items matching the query. If matching items exist, they are found and deleted. There is no warning prompt so be sure this is the action you wish to use.
    - True will be returned if any records exist
    - False will be returned if 0 records exist.
    - Records\_deleted will track the number of records deleted
* getRecordCriteria(criteria=None)
  + If records are to be located using criteria, this method should be used. Criteria can be one or many.
  + By default the criteria is *None,* this will return all records.
* getRecordId(data)
  + When a record is located by the server assigned GUID, this method is used.
* updateRecord(newValue, query)
  + updateRecord searches the database for documents matching the query. The document values matching the *key:value* pairs in *newValue* will replace the existing data.
    - True will return if the records updated is greater than 0.
    - False is return if no records are updsated.
    - Records\_modified shows the number of records modified.
    - Records\_matched stores the number of records matched

### Code Example

Example to create the following record:  
  
*animals.createRecord({*

*'age\_upon\_outcome': "1 year",*

*'animal\_id': 'test\_id',*

*'animal\_type': 'test',*

*'breed': 'test breed',*

*'color': 'color',*

*'date\_of\_birth': '1900-01-01',*

*'datetime': '1900-01-01 12:00:00',*

*'monthyear': '1900-01-24T12:00:00',*

*'name': 'name',*

*'outcome\_subtype': '',*

*'outcome\_type': 'test',*

*'sex\_upon\_outcome': 'test',*

*'location\_lat': 10.10,*

*'location\_long': -10.10,*

*'age\_upon\_outcome\_in\_weeks': 123.123*

*})*

*)***True** was returned as the record was successfully written to the database.  
Example to query for a dog named Rex with object *animal*:

* Use the Constructor to create the object animal and initialize the MongoDB server.
  + *animal = AnimalShelter(‘password’, ‘username’)*
* Create the query object to store the returned documents.
  + *query = animals.getRecordCriteria({"name": "Rex"})*
* If the age of *Lucy* should be *8 months* then use this query
  + *query = animals.getRecordCriteria({"name": "Lucy", 'age\_upon\_outcome': '8 months'})*

In order to add animals to the database, you would use:

*Print(animals.create(“STRING\_TYPE”))*

This function will add the animals then return a Boolean to indicate either success or error.

The retrieval of information can be completed with:

*query = animals.getRecord({“name”: “NAME”)}*

The parameters can be adjusted based on your particular need.

### Tests

Code was tested with the use of invalid statement:

*Print(animals.create(0.0))*

This created an invalid argument while it tries to create an invalid data type.

### Screenshots

A screenshot of a computer program

Description automatically generated

Deleting a record.

A close up of a text

Description automatically generated

Number of documents deleted determined with available property.



Output:



Updating records with current date/time string

A computer screen shot of a computer code

Description automatically generated

## Contact

Your name: Jason Kremhelmer CS-340 August 3, 2024