# CS 340 README Template

## Project One

*The application facilitates Create-Read-Update-Delete (CRUD) action into MongoDB via Jupiter Notebook and allow access to the database of animals in Austin Animal Center (AAC). Project One provide knowledge of the interaction between MongoDB and Jupiter Notebook.*

## Motivation

*Project One is created to test developer skill by working with database and manipulation of the data. The developer will have firsthand experience on creating CRUD action and interact with database from MongoDB. Project One provide experience with Python language and compilation using Jupiter Notebook.*

## Getting Started

*To get started first the user will need to open the Client Server Development (Linux).*

* *The user will open the Open in Terminal and prompt the command /usr/local/database*
* *Import the database by typing mongoimport –port #### --db AAC –collection animals –type cvs –headerline – file aac\_shelter\_outcomes.csv to import the database.*
* *The user will need to open Jupyter Notebook*
  + *Note: To keep organization you can create a new folder for your CRUD*
* *Open Jupiter Notebook*
  + *Note: For better organization I will recommend creating a new folder*
* *Create a .py and .ipynb*

## Installation

* *MongoDB and access to the database*
* *Jupyter Notebook*

## Usage

### Code Example

* Create – Inser dpcuments tyo
  + Input: Argument to function to set key and values
  + Output: True if success, else False

def create(self, data):

if data is not None:

result = self.database.animals.insert(data) # data should be dictionary

if inster != 0:

return True

else:

return False

else:

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

* Read – Queries for documents from MongoDB
  + Input: Key/Value lookup pair use with the MongoDB
  + Result if successful, else Error message

def read(self, criteria=None):

if criteria is not None:

data = seld.database.animals.find(criteria,{"\_id": False})

for documents in data:

print(document)

else:

data = self.database.animals.find({},{"\_id":False})

return data

* Update – Change documents from MongoDB
  + Input: Lookup for a pair and set as acceptable to change for MongoDB
  + Output: Result JSON formant if successful, else Error message

def update(self, searchData, updateData):

if searchData is not None:

result = self.database.animals.update\_many(searchData, {'$set':{updateData}})

return result.raw\_result

else:

return "{}"

#Return the dataset else error

return result.raw\_result

* Delete – Remove documents from MongoDB
  + Input: Key/Value lookup to use with the MongoDB
  + Output: result in JSON format if successful, else Error Message

def delete(self, deleteData):

if deleteData is not None:

result = self.database.animals.delete\_many(deleteData)

else:

return "{}"

#Return the dataset else error

return result.raw\_result

* Test Create
  + Test the create function to verify the functionality

print(animals.create({

"age\_upon\_outcome": "2 year",

"animal\_id": "test",

"animal\_type": "Cat",

"breed": "Domestic Short Hair",

"color": "Black/White",

"date\_of\_birth": "2021-12-19",

"datetime": "2022-04-03 12:00:00",

"monthyear": "2022-04-03 12:00:00",

"name": "Binx",

"outcome\_subtype": "SCRP",

"outcome\_type": "Transfer",

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

"location\_lat": 15.15,

"location\_long": -80.300,

"age\_upon\_outcome\_in\_weeks": 15

})

)

* Test Read
  + Read function would be tested

query = animals.read({"name": "Binx"})

for animal in query:

print (animal)

* Test Update
  + Update function would be tested

updateAnimal = animals.update({"name": "Binx"}, {"outcome\_type": "Adopted"})

print(updateAnimal)

* Test Delete
  + Delete function would be tested

deleteAnimal = animals.delete({"name":"Binx"})

print(deleteAnimal)

### Tests

Graphical user interface, text, application, email, website

Description automatically generatedCreate Method

A picture containing text

Description automatically generatedRead Method

Update Metod

A picture containing graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidenceDelete Method

## Test of Methods

Create Method Test

Text, letter

Description automatically generated

## Read Method Test

Text, letter

Description automatically generated

Graphical user interface

Description automatically generatedUpdate Method Test

Graphical user interface, text, application, chat or text message

Description automatically generatedDelete Method Test

## Contact

Your name: Sergio Mateos