from pymongo import MongoClient

from bson.objectid import ObjectId

class AnimalShelter(object):

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

search = none

def \_\_init\_\_(self):

# Initializing the MongoClient. This helps to

# access the MongoDB databases and collections.

self.client = MongoClient('mongodb://%s:%s@localhost:41644' % ("aacuser", 'abc123'))

# where xxxx is your unique port number

self.database = self.client['aac']

# Complete this create method to implement the C in CRUD.

def create(self, data):

if data is not None:

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

if insert != 0 :

return True

else:

return False

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

# Create method to implement the R in CRUD.

def read(self, data):

#Check to see if data is empty

if search is not None:

if data:

searchResult = self.database.animals.find(data)

return searchResult

else:

return "error"

#Exception("There is nothing to search for")

return error

# Complete this Update method to implement the U in CRUD.

def update(self, lookup, table):

#First Find the record to update

if table is not None:

# record to be updated found

update\_result = self.database.animals.update\_many(lookup, table)

result = "The Document Updated: " + json.dumps(update\_result.modified\_count)

return result

else: # record not found

return "error"

# Complete this delete method to implement the D in CRUD.

def delete(self, data):

#First Find the record to update

if data is not None:

delete\_result = self.database.animals.delete\_many(lookup, table)

result = "The Document Deleted: " + json.dumps(delete\_result.delete\_count)

return result

else: # record not found

return "error"