# CS 340 README Template

## About the Project/Project Title

This project is a scenario for Global Rain, a software engineering company that specializes in custom software design and development. The team is tasked with a project for an international rescue-animal training company Grazioso Salvare. This project is aimed at creating an end user application database for Grazioso Salvare, in which they can keep track of animals they work with.

## Motivation

Grazioso Salvare identifies dogs that are good candidates for search and rescue training. These dogs will be able to find and help rescue humans or other animals in life threatening situations. Grazioso Salvare is seeking an application that can work with data that they already have from animals shelters to be able to identify and categorize available dogs.

## Getting Started

Currently, the project is in its beginning stages but to currently run the program you will need programs listed under “Installation” section.  
Within MongoDB, the database AAC needs to imported. Also, users need to be created for the program to run outside of the terminal or command prompt. The users created were an administrator with full read/write ability, and a user called “aacuser” that had also has a read/write ability.

The Python driver for Mongo that was used was PyMongo, and it was chosen for a few different reasons. First and foremost, PyMongo provides the methods for CRUD, which is the essence of the module created for this project. Secondly, PyMongo is made specifically for working with MongoDB, which is the database used. In the Create function, the user can add an entry, or document, into the database with as much information available, in which it will be stored for later retrieval. The Read function, will search for a certain item in the collection by a key/value pair (i.e. {“name” : “Meeko”} ) If found, it will return the entry (or entries) in JSON format. Next, the Update function will search for an item by a key/value pair, and edit it to change data within the entry. If successful, it will again return the entry in JSON format. Last, the Delete function will delete an item by key/value pair and return the key/value pair to reassure the user that it had in fact been deleted.

## Installation

*Jupyter Notebook @ https://jupyter.org*

*MongoDB @ https://www.mongodb.com/docs/manual/installation/*

## Usage

### Code Example

Graphical user interface, text, application

Description automatically generatedThe code starts with importing the MongoClient from PyMongo, and a declaration of the class CRUD\_shelter and \_\_init\_\_ function to connect to the database.

### Tests

A picture containing text

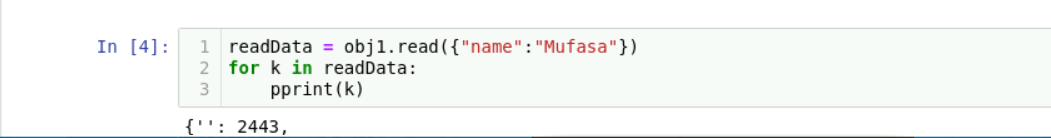
Description automatically generatedTesting the code start with the two first kernels pictured below. One being importing the module, and the other as some variable declarations.

Next, there we can test out the CREATE method by passing the ‘data’ variable

Graphical user interface, text, application, email

Description automatically generated

After ‘data’ has successfully been added, we can call the READ function by searching and outputting results based on a key/value pair



## Once we can see the output of the READ function, we can try the update function. This is done by inputting the collection document we would like to update(based on a key/value pair) with the data we will update it with. The function will then update each appearance of the “data” variable with the “updateData” variable.

Graphical user interface, text, application, email

Description automatically generated

Last but not least, we use the DELETE function, which will remove the entry from the database, and return the key/value pair to verify that it is missing and has been deleted.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Screenshots

Text

Description automatically generated

Text

Description automatically generated

## Contact

Matthew Trembley

matthew.trembley@snhu.edu