Project Two

Kyle Dale
SNHU: CS-340-Client/Server Dev
Professor Sanford
August 13, 2023

Project Explanation

The goal of this project is to create a portable python script that enables CRUD capability in a database. Specifically, for our project the CRUD capability will allow us to alter and filter a database of animals to more easily find, recruit, and train rescue dogs. Additionally, the created python module was used to connect the UI and database components to meet Grazioso Salvare specifications. As it stands now, proper CRUD functionality has been implemented, allowing the creation, reading, updating, and deletion of entries in a database. Additionally, a dashboard has been designed and implemented to the specifications of our client Grazioso Salvare, which allows users to filter the animals in the collection based on the required criteria for each rescue type.

Motivation/Use

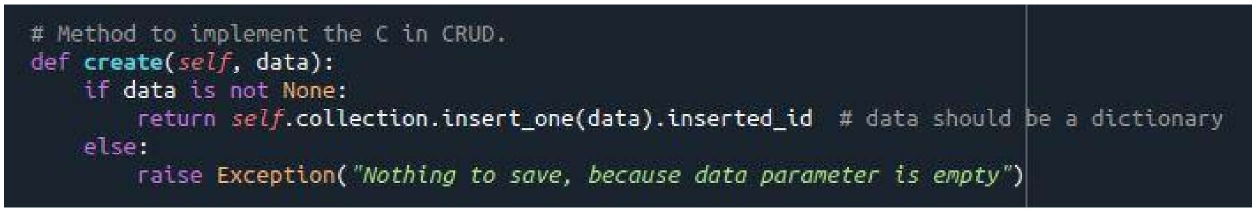
The motivation behind the creation of the python script is to possess a portable module that enables CRUD functionality in a database. For the purposes of this project, PyMongo library was selected as our Python driver for MongoDB. PyMongo was selected for a variety of compelling reasons:

1. PyMongo is one of the oldest, and most widely used drivers for Python when utilizing MongoDB. Additionally, PyMongo is the official native driver for Python with MongoDB, as such it is a natural choice (MongoDB, 2023).
2. As PyMongo is the official native driver for Python with MongoDB, it receives official support from MongoDB and is assured to adhere to best practices and standards (MongoDB, 2023).
3. An additional benefit of official support by MongoDB is that PyMongo will always be rich with tools inherently compatible with MongoDB and the MongoDB server.
4. PyMongo provides excellent documentation and support for those unfamiliar with the program. Extensive tutorials and reference material can be found at the official MongoDB site, which have proven incredibly useful (MongoDB, 2023).

Ultimately, as a beginner previously unfamiliar with MongoDB, PyMongo was the logical choice to familiarize myself with DB management and manipulation.

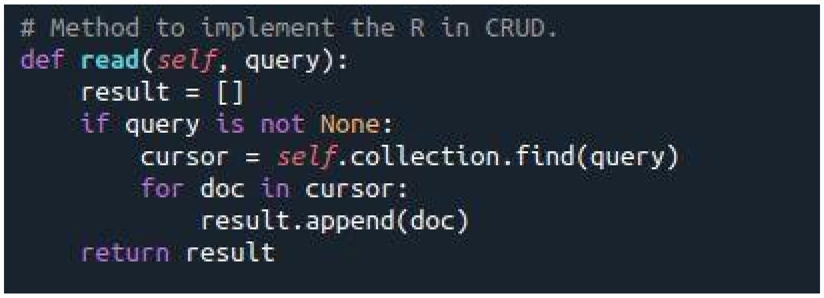
When concerning the attributes and working functionality of the CRUD operations, we can analyze each letter of CRUD individually to break down proper functionality:

1. For the *C* or create, we can see in the CRUD_Functionality_Module_KD that the associated method is create(data). This method is used to insert a new entry into the MongoDB collection, or animals in this case. The insert_one(data) method of PyMongo is used to perform insertion of the entry. Create takes a single argument, ‘data’, which is expected to be a dictionary containing the data for the new entry.



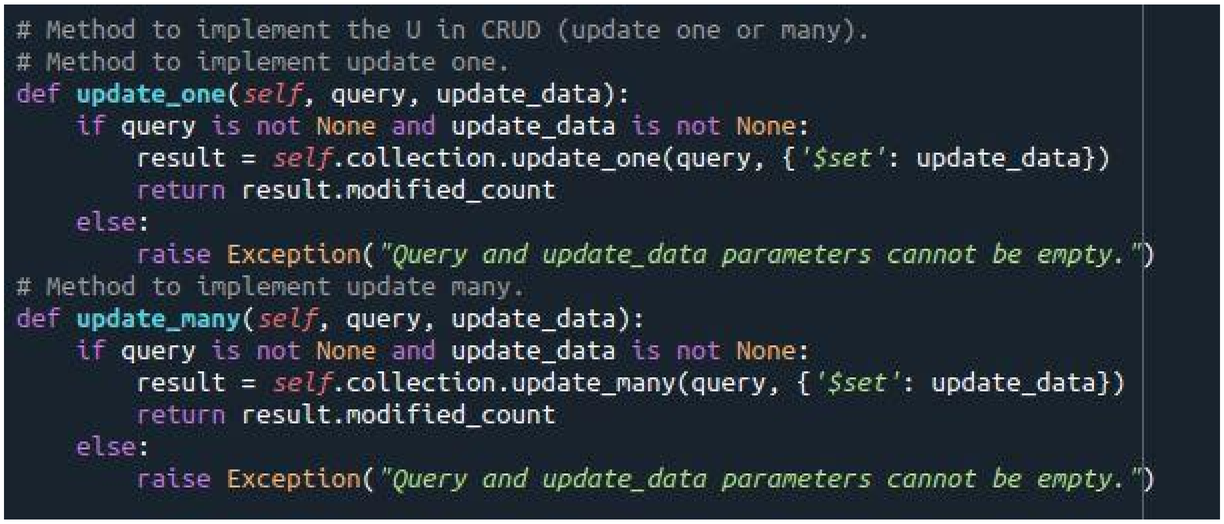
```python
# Method to implement the C in CRUD.
def create(self, data):
    if data is not None:
        return self.collection.insert_one(data).inserted_id  # data should be a dictionary
    else:
        raise Exception("Nothing to save, because data parameter is empty")
```

2. When concerning the next method, *R* or read, we can see in the CRUD_Functionality_Module_KD that the associated method is read(query). This method is used to retrieve entries in the DB based upon a specific example, in the example provided in the tester script, that would be the species 'Cats'. The find() method of PyMongo is used to perform the query and return the results.



```
# Method to implement the R in CRUD.
def read(self, query):
    result = []
    if query is not None:
        cursor = self.collection.find(query)
        for doc in cursor:
            result.append(doc)
    return result
```

3. When concerning the *U*, or update functionality, the implementation was slightly more complicated. We can see in the CRUD_Functionality_Module_KD.py that update functionality is actually associated with two methods, update_one(query, update_data) and update_many((query, update_data). These methods are utilized to update one, or multiple entries within the MongoDB collection matching the specified query. The update_one() and update_many() methods of PyMongo are utilized to perform the updates, as well as the '$set' operator to set the values.



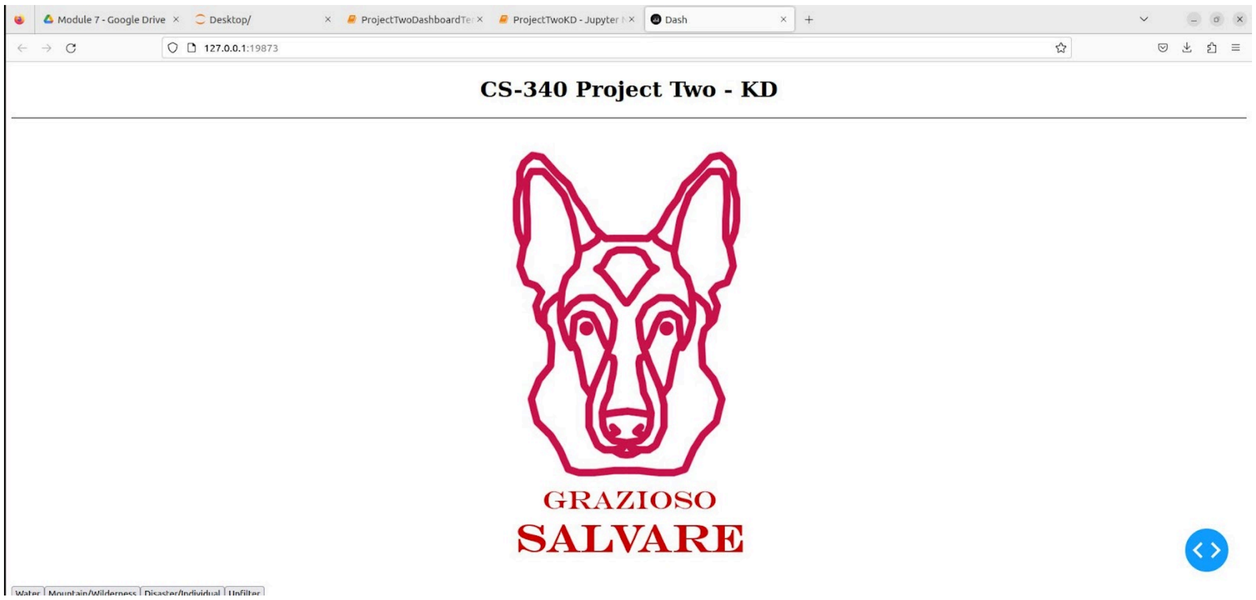
```
# Method to implement the U in CRUD (update one or many).
# Method to implement update one.
def update_one(self, query, update_data):
    if query is not None and update_data is not None:
        result = self.collection.update_one(query, {'$set': update_data})
        return result.modified_count
    else:
        raise Exception("Query and update_data parameters cannot be empty.")
# Method to implement update many.
def update_many(self, query, update_data):
    if query is not None and update_data is not None:
        result = self.collection.update_many(query, {'$set': update_data})
        return result.modified_count
    else:
        raise Exception("Query and update_data parameters cannot be empty.")
```

4. Finally, when concerning the *D* or deletion functionality, the implementation was similar to updating. We can see in the CRUD_Functionality_Module_KD.py that the methods used by deletion are delete_one(query) and delete_many(query). These methods are utilized to remove one or multiple entries inside the collection. The delete_one() and delete_many() methods of PyMongo are utilized to perform the deletions.

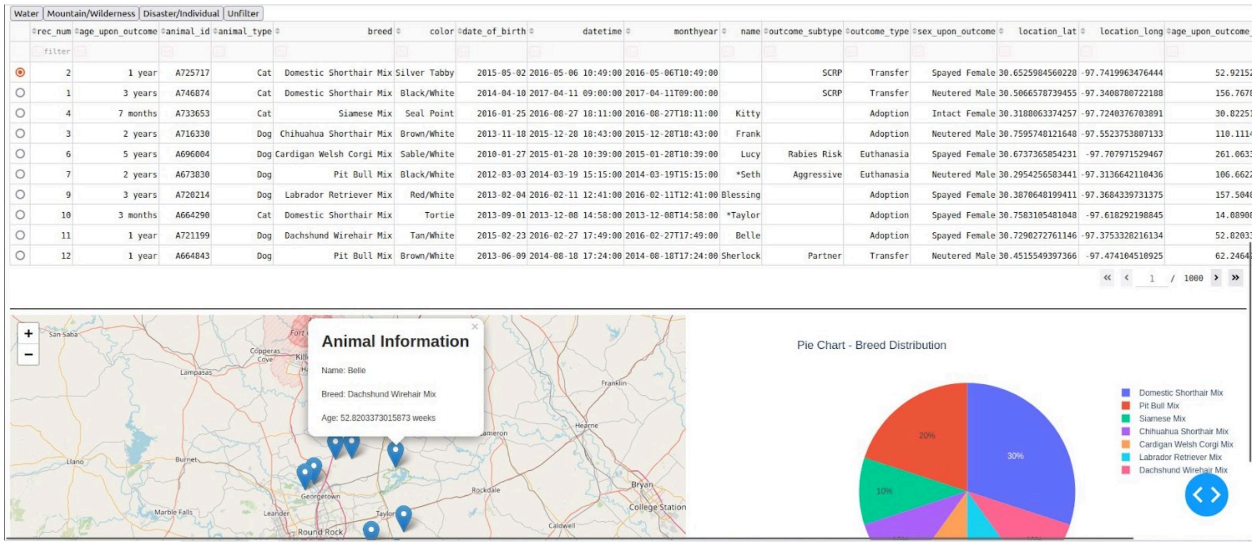
When concerning the use of the application, Grazioso Salvare will be able to utilize the created dashboard to easily filter an included database, in this case a database of animals to find the preferred species, gender, age, and breed of animal. This will allow them to efficiently locate potential recruits for their rescue animal training program. The page begins with a unique heading specific to my project, and follows with the logo for Grazioso Salvare. Following the logo, the page contains four buttons in the following order:

1. **Water**: Filters for animals matching water rescue criteria.
2. **Mountain/Wilderness**: Filters for animals matching M/W rescue criteria.
3. **Disaster/Individual**: Filters for animals matching the D/I rescue criteria.
4. **Unfilter**: Returns the displayed data to an unfiltered result. This can also be achieved using any of the other three buttons. I.E. if ‘Water’ is selected, clicking D/I once will unfilter the result, clicking again will filter for D/I.

After the various filter buttons is the data itself, displaying multiple relevant columns and is limited to 10 rows currently. The row limit and columns displayed can be changed with minimal alteration within the program. Underneath the data is a geo-map and pie chart, utilized to visualize the data more completely. Currently the geo-map will display map markers for each of the 10 rows of data displayed on the current page. This will update when changing to pages 2 or 3 etc. When clicking on a map market the user is provided with the animal's name, breed, and age. The pie chart displays a breakdown of the present breeds in the current data page. These widgets serve to provide Grazioso Salvare an easily digested UI for selected potential rescue recruits. Finally, the bottom of the page contains my unique signature and identifier.



CS-340 Project Two - KD



	rec_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_
	filter															
●	2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SCRP	Transfer	Spayed Female	30.6525984560228	-97.7419963476444	52.92152
○	1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722188	156.7678
○	4	7 months	A733653	Cat	Siamese Mix	Seal Point	2016-01-25	2016-08-27 18:11:00	2016-08-27T18:11:00	Kitty		Adoption	Intact Female	30.3188063374257	-97.7240376703891	30.82251
○	3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.5523753807133	110.1114
○	6	5 years	A696004	Dog	Cardigan Welsh Corgi Mix	Sable/White	2010-01-27	2015-01-28 10:39:00	2015-01-28T10:39:00	Lucy	Rabies Risk	Euthanasia	Spayed Female	30.6737365854231	-97.707971529467	261.0633
○	7	2 years	A673830	Dog	Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15:00	*Seth	Aggressive	Euthanasia	Neutered Male	30.2954256583441	-97.3136642110436	106.6622
○	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3870648199411	-97.3684339731375	157.5046
○	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor		Adoption	Spayed Female	30.7583105481048	-97.618292198845	14.08908
○	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7290272761146	-97.3753328216134	52.82033
○	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.4515549397366	-97.474104510925	62.24642

« ‹ 1 / 1000 › »

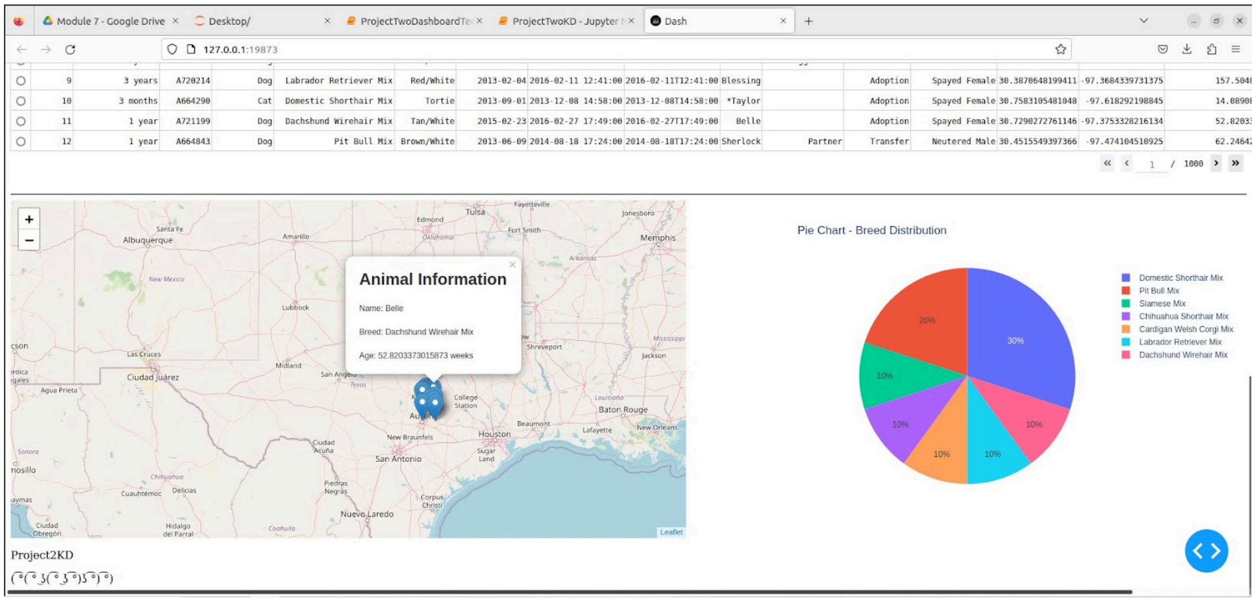
Animal Information

Name: Belle

Breed: Dachshund Wirehair Mix

Age: 52.8203373015873 weeks

Pie Chart - Breed Distribution



○	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3870648199411	-97.3684339731375	157.5046
○	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor		Adoption	Spayed Female	30.7583105481048	-97.618292198845	14.08908
○	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7290272761146	-97.3753328216134	52.82033
○	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.4515549397366	-97.474104510925	62.24642

« ‹ 1 / 1000 › »

Animal Information

Name: Belle

Breed: Dachshund Wirehair Mix

Age: 52.8203373015873 weeks

Pie Chart - Breed Distribution

Project2KD

(͡°(͡° ͜ʖ(͡° ͜ʖ ͡°)ʖ ͡°) ͡°)

(These images display the Grazioso Salvare dashboard before any input has been given to the page. It demonstrates the baselines functionality of the application, before filtering.)