# Grazioso Salvare AAC README

## About the Project

This full-stack application allows users within Grazioso Salvare to gather information from the Austin Animal Center database about specific animals. This information includes but is not limited to animal type, breed, geographic location, and graphical breed statistics. Included within the application is the ability to select from 3 preset filters and 1 reset option, designated at the top of the application. Among the information displayed, you are also able to type in specific information into the filter section of the database table to show only relevant records. Above the filter section are column headers that allow you to sort in ascending or descending order for any necessary columns. The primary database system for this application is MongoDB due to its ease of use with the Python module Pymongo. The Dash framework was utilized to present the information pulled from the database on screen in a consistently formatted viewing window.

## Motivation

This application was implemented to allow users to easily access animal information on demand with a user interface that was clear and straightforward instead of requiring lengthy database queries. Using this application allows users to gather reporting information for their needs.

## Getting Started

To start using this application, you need to install the following software:

Python 3.6 or newer (<https://www.python.org/downloads/>)

MongoShell 1.8.0 or newer (<https://www.mongodb.com/try/download/shell>)

Any Python file editor, JupyterNotebook was used in this case (<https://jupyter.org/>)

Once these have been installed, you can start the use of this application.

## Installation

To install the Python module, you can use the following command within your Terminal after Python has been installed and the Python CRUD Module is saved on your computer.:

*pip install –user animal\_shelter\_operations*

Then, the application files are ready to be used as they now have access to the required modules, along with current open-source modules within the main file.

## Usage

Below are examples of the application’s interface once the file has been run.

### Screenshots

Initial screen when the application is opened:

A screenshot of a computer

Description automatically generated

In the top lefthand corner are the three preset filter options and the “Reset” option. The default filter when the application is opened is “Reset”, which has no filters. The database information is shown below the filters. Within this table, each column can be sorted in ascending or descending order by clicking the up/down arrows next to each column name. Below the column names is a field that you can type any text to filter the information within that column. On the left side of the table are buttons to select which record you would like to view the geographical location for. Below the table is my name, followed by the logo for Grazioso Salvare. Next to the logo is a graph that shows a list of the percentage of each breed within the displayed database. Due to the sheer number of different breeds, the chart is very cluttered. This is not a problem when any form of filtering is used to reduce the breed counts. To the right of the graph is a GPS location for the currently selected record. The map centers on the coordinates for the current record and starts with the map centered on Austin before any record is selected. As filters are added, the first record within the table is used for location information. If you click on the pin within the map, the name of the current animal is displayed.

Once the “Water Rescue” preset filter is used:

A screenshot of a computer

Description automatically generated

Shown below, the “Mountain Rescue” preset filter is used:

A screenshot of a computer

Description automatically generated

Shown below, the “Disaster Rescue” preset filter is used:

A screenshot of a map

Description automatically generated

Finally, the “Reset” preset filter is used to clear out any filtering:

A screenshot of a computer

Description automatically generated

## Steps Taken

Initially, the Austin Animal Center shelter had to be imported into MongoDB in a database named AAC with a collection named animals. Once the database was imported, an account was created with a specific username and password to limit access to the database features. A module was then created to manipulate the database using the created account. This module was the “animal\_shelter\_operations” module that was installed previously. With the module created, the Dash framework was utilized to write the application for the user to see within their web browser. The filtering, graph, map, and database views were created using the callback functions of Dash in JupyterNotebook.

## Challenges

There were many challenges within this project. I initially had trouble updating the table information based on filters within the table. I found that I forgot to pass the updated information back to the screen and was resending the original data. I also ran into many issues with the graph populating. I was accidentally pulling the full database information instead of the filtered data that was presented based on the buttons at the top. I continue to have one error on startup that I cannot figure out, which is an error when the map first populates. The error is with the row variable in which row 0 does not have an index 13 or 14. Indices 13 and 14 are the coordinates for the selected record. Everything after the initial startup works as intended.

## Contact

Chayne Pieri (Chayne.Pieri@snhu.edu)