# CS 340 README

## About the Project/Project Title

## This class provides a straightforward approach to executing Create-Read-Update-Delete (CRUD) tasks within MongoDB using Pymongo. It consolidates critical accessors and mutators (get and set methods) necessary for CRUD operations, along with facilitating MongoDB initialization.

## Motivation

Handling CRUD operations is a common task in database usage, but it can be tedious to format queries and manage errors. To ease this burden of formatting each new query or rewriting code for different error types, these operations have been abstracted into simple getter and setter methods.

## Getting Started

To begin using this class, refer to the following guide that makes the process easy to understand.

## Installation

1. Install Python 3.6:
   * Visit the Python website: https://www.python.org/downloads/
   * Download the Python 3.6 installer suitable for your operating system.
   * Follow the installation instructions provided by the installer.
2. Install Pymongo 4.2:
   * Open a terminal or command prompt.
   * Use pip, Python's package installer, to install Pymongo:

pip install pymongo==4.2

1. Install MongoDB 4.2:
   * Visit the MongoDB website: https://www.mongodb.com/try/download/community
   * Download the MongoDB Community Server installer suitable for your operating system.
   * Follow the installation instructions provided by the installer.

## Usage

*Constructor:*

* *Initializes the MongoDB server and establishes a connection to the database.*

*create(data):*

* *Creates a record using the provided data argument. The data must adhere to the format specified by the Pymongo API. An example is provided below. Returns True if the create action is successful; otherwise, returns False. If no argument is passed, an exception will be raised, and an error message will be provided to the user.*

*read(criteria=None):*

* *Locates one or more records based on the specified criteria. The criteria can be singular or multiple and must be formatted according to the Pymongo API. An example is provided below. If no criteria are provided, all records will be returned by default.*

### Tests

To search the dataset for an animal with specific criteria

1. Create the object animal and initialize the MondoDB server via the Constructor

A screenshot of a computer

Description automatically generated

1. Create the object that will store the returned documents

*A screenshot of a computer

Description automatically generated*

## Contact

Tonya Sanders