Create a Design Document for your project.

This document should discuss both high and low level design of your proposed application.  The final product may differ from this design, but the design is a place to start.

Requirements:

Must include all of these sections:

* Introduction

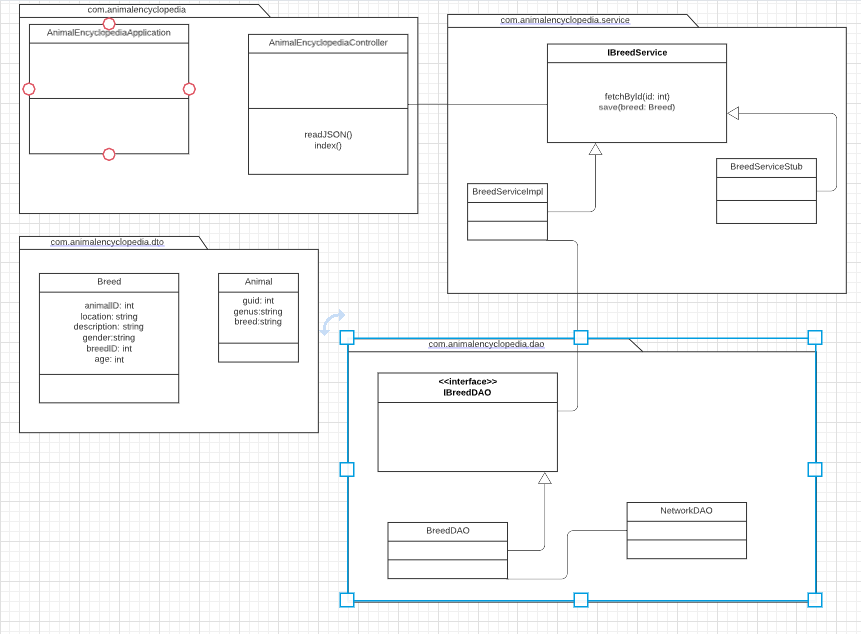
In this project, we will create an animal encyclopedia to serve people. First of all, this will be presented in the form of a website. And the main function is to query the information related to raising animals. The specific content is to set up information query index for each department based on different kinds of animals. People can inquire about vaccine, diet, hygiene, daily care and other information. It will also provide a search bar to facilitate people to quickly search the information interface. The website is intended to help people better raise their pets, so as to obtain more happiness.

* Storyboard (screen mockups): FluidUI, Powerpoint, paint, etc... will be fine.





* Functional Requirements in the format (fill in the square brackets with your own words):
  + - As a dog keeper
    - I want to know how to feed my dog
    - So that I can keep it healthy
    - Elaborate each of these with several examples in this format:
      * Given pet kind
      * When clicking index
      * Then get the view of food
    - As a cat keeper
    - I want to know how to wash my cat
    - So that I can keep it clean
    - Elaborate each of these with several examples in this format:
      * Given pet kind
      * When searching in search tube
      * Then get the direction to hygiene view
* **Class Diagram**



* **Class Diagram Description**

Controller class has methods/operations: readJSON, index() etc.

DTO has a bunch of attributes and getters/setters. Getters/setters are assumed. DTO has classes:Breed and Animal. Breed and Animal have their attributes listed above and their data type. Animal is broad and Breed is a more specific type of the animals.

For the classes: first box is the name, second box is the attributes, third box is any operations.

IBreedService has interfaces with operations such as: fetchByID, save, etc.

Controller has realtionship with IBreedService interfaces, will use/call the fetchById when called, etc.

AnimalServiceStub is implemented by IBreedService.

AnimalServiceImpl is the implementation that has the user relationship in the DAO.

**One or two lines for each class to describe use of interfaces**

Breed- specific type of animal

Animal- name of animal

Application- where the information is stored

Controller- where the information is called, and actions will be implemented

BreedServiceImpl- where the actual implementation will happen, will connect DAO and service

* A Scrumy.com link that contains:
  + Product Backlog: list stories (features) under the Stories column on Scrumy.
    - Product backlog items should be features, like "Take a picture and post on Facebook", not technical, like "Make the database"
  + Sprint Backlog for Sprint #1
    - These should be technical tasks that are required to implement one of the features.
* Scrum Roles, and who will fill those roles
* https://github.com/YifeiYang20/animal\_encyclopedia  project link