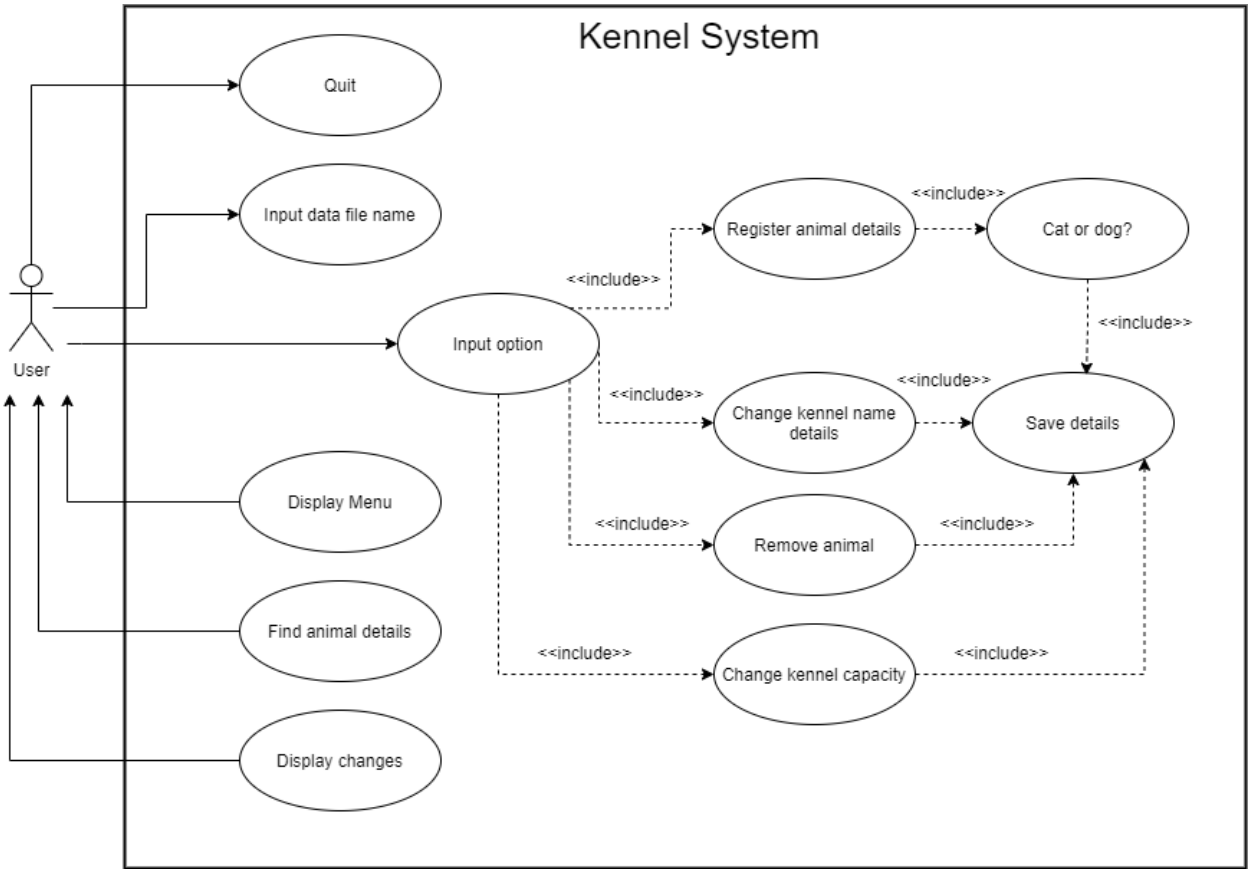
Jay Staddon

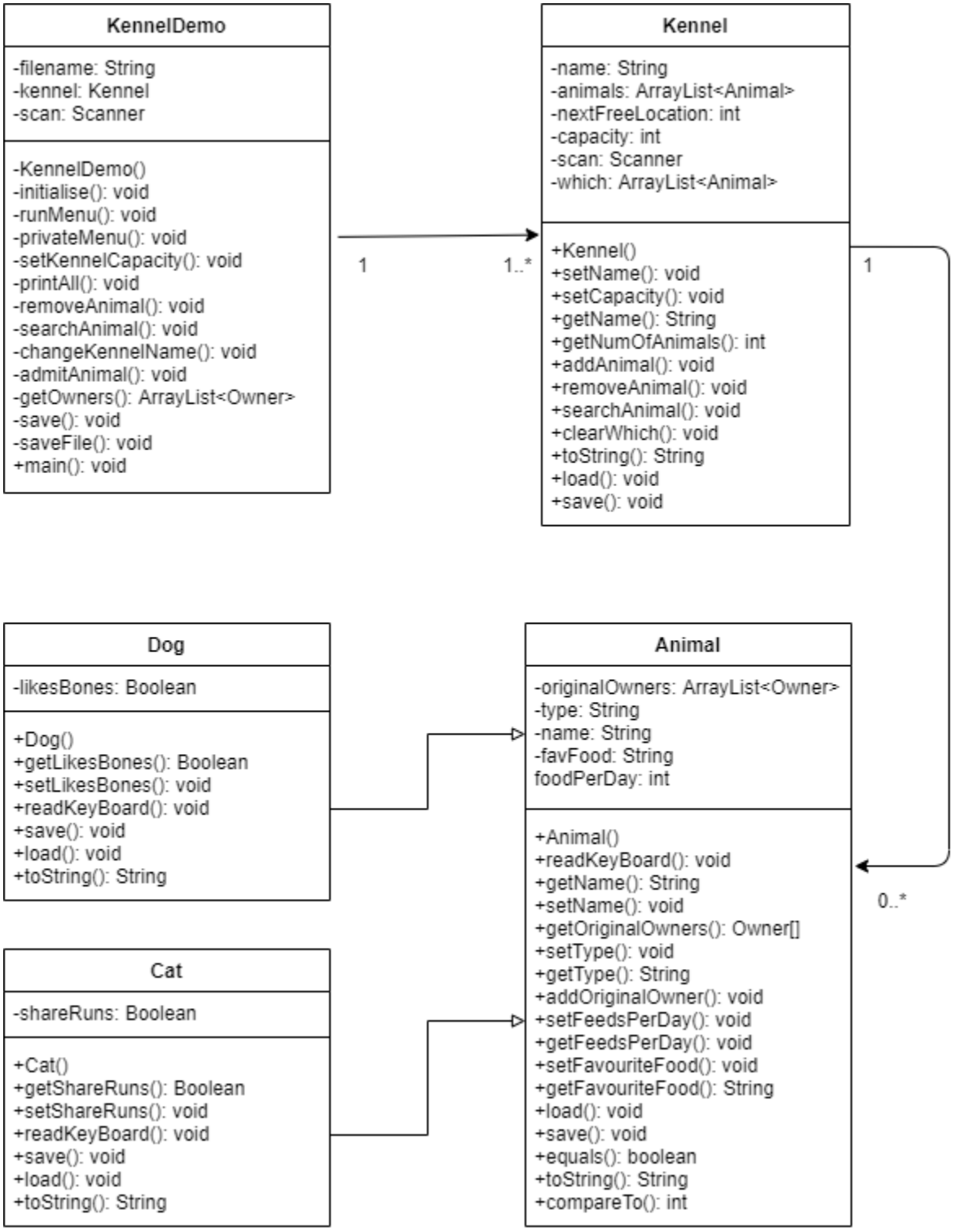
Mini-Assignment 3: Kennels

Email: jas130@aber.ac.uk
Date: 25/03/2021

UML use case diagram



UML class diagram



Report

When I started the assignment I first read through it a few times to make sure I had a proper understanding of what it was we had to do. I then went on to try out the provided kennels.zip to see how it functioned and how it could be improved upon.

The first thing I looked to change was the Dog class, turning it into an Animal class which could then be inherited by a Dog and a Cat class. This meant I had to change some constructors and edit how the other classes referenced it. The main challenge here was making sure I didn't miss any code which still only referenced Dog, including the toStrings, messing up the code.

After creating a Dog and Cat class with their variables I went onto looking at how I would now save the data the user adds to the kennel. I decided on a similar approach to the original code, but I would add the animals type separately, inside each subclass meaning there wouldn't be any errors or misassigned types once the code was loaded. The difficulty I found here was making sure all the data would save in the correct format and load back up again, although after some planning I managed to make it function to my liking.

I then went onto doing general improvements to my code, making it function the way the assignment said and to also the way I wanted it to. This includes allowing the user to select which animal to remove if their names are the same, cancel a decision, display cats, dogs or all animals, displays animals in alphabetical order and user interface improvements.

When it came to testing my solutions I would run them through the debugger to see the pathway my code was taking and to make sure I hadn't inadvertently used the wrong variable or method. Once I knew that the code would function properly there I would then attempt to break my code to see if it would still function correctly, even with the wrong input type.

Within my code, it has lots of features for the user. However, with more time there are a few additional elements I would wish to add. One such addition would be allowing the user to use any piece of data to look up an animal, eg. searching for dogs that like bones. Another feature I would like to add is the ability to edit the details of an animal, eg. a cat can no longer share runs. I would also like to go back and make sure that if the wrong input is entered it asks the user to correct their mistake.

After producing this code I believe that I should be awarded 80% of the marks, as I have produced a detailed use-case diagram and class diagram as well as given an overview of my code creation process. I have also implemented all of step 3 into my code and have used inheritance to improve the overall structure of the program. The flair in my code is shown by the additional features which were implemented to make the user experience better.

Code

```
**********HELLO**********

Please enter the filename of kennel information: data.txt
Using file data.txt

Kennel: Jay's Kennel

1 -  Add a new Animal
2 -  Set up Kennel name
3 -  Display all inmates
4 -  Search for an animal
5 -  Remove an animal
6 -  Set kennel capacity
q -  Quit
What would you like to do:
```

The main menu screen

```
What would you like to do:
1
What kind of animal do you want to add
1: Dog
2: Cat
0: Cancel
```

Option 1 lets you pick which animal you would like to make

```
1
enter:
Name of animal
Rover
What is his/her favourite food?
lamb
How many times is he/she fed a day? (as a number)
2
Do they likes bones (Y or N)
y
Enter owners name
John
Enter owners phone number
12345
Another owner (Y/N)?
n
```

It lets you enter the details of the animal

```
What would you like to do:
2
Enter new kennel name:
Jay's Kennel
```

Option 2 lets you enter a new kennel name

```
Do you want to search for:
1: Dogs
2: Cats
3: All
```

Option 3 lets you search for a certain type of animal

```
3
Data in Kennel: Jay's Kennel is:

Pet name: Rover
Type: Dog
Original Owner: [John 12345]
Favfood: lamb
FoodPerDay: 2
likes Bones: true
```

It then displays those animals

```
What would you like to do:
4
Which animal do you want to search for
rover
1
Pet name: Rover
Type: Dog
Original Owner: [John 12345]
Favfood: lamb
FoodPerDay: 2
likes Bones: true
```

Option 4 lets you look up animals by name

```
What would you like to do:
5
Which animal do you want to remove
rover

1
Pet name: Rover
Type: Dog
Original Owner: [John 12345]
Favfood: lamb
FoodPerDay: 2
likes Bones: true

Removed rover
```

Option 5 lets you remove an animal

```
What would you like to do:
6
Enter max number of animals: 10
```

Option 6 lets you change the kennel capacity

```
What would you like to do:
q
**********GOODBYE**********
```

Option q closes the program