

Animal Village Application Design Document

Developer: Denzil Gideon M. Daulo

Date: 02/16/2020

Presented to: Paf

Objectives

Create an application that will execute the following behaviour:

- Scheduled behaviour of the animals during the day
- By start of the day until before noon the animals will attempt to unfriend their animal friends with the exception to its best friend forever
- By noontime the animals are grouped based on their favorite food and they will eat together during lunch time
- By afternoon the animals will attempt to befriend other animals
- By end of the day the application will display a table that will show the summary of the friendship status of each animals to other animals
- The mentioned behaviour of the animals will be done in the course of 10 days

Things Need to consider on the Animal Village

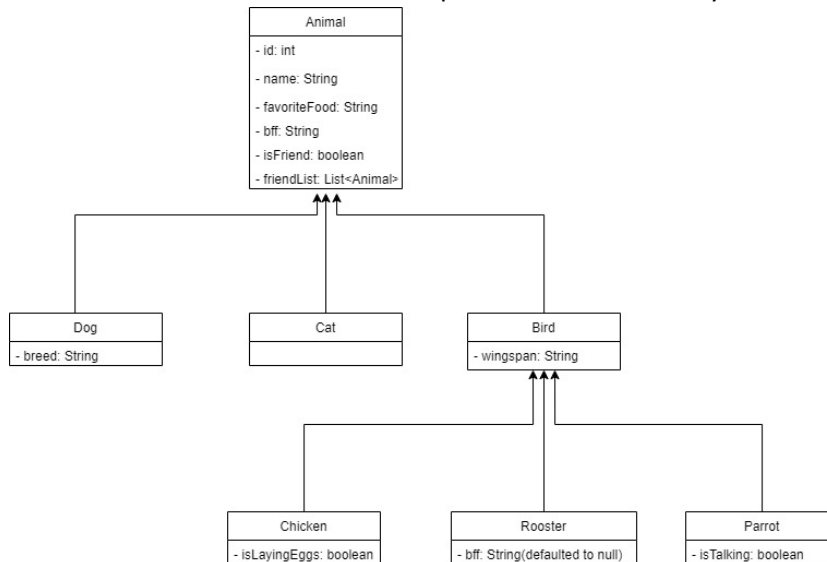
There are 5 types of Classes that needed to be considered to make the application work as expected:

- Models
- Helpers
- Utilities
- Scheduler
- Main Client

Models

- Responsible for containing the information that will be used to display in the tables and other information (i.e. favorite food and groupings for Lunch Time)
- Responsible for containing the information about the relationship of one Animal to another Animals
- There are about 5 classes of animals that are involved in the application
- Each class of animals has a unique trait that needed to be considered, thus they are needed to be specialized based on their type
- 3 of these classes are birds, which have a common trait. A parent class will be needed to organize the specialized bird class animals
- Each class of the animals have a very common trait. The common trait needed to be placed to a common parent class that will be inherited by each specialized class of Animals
- Common fields are also introduced to help establishing the relationship between animals:
 - o id:int – A unique Id for each of animal model. This will help on checking on how the relationship will be established

- isFriend:boolean – A friendship flag to check and set depending on the status of the relationship
- friendList:List<Animal> - A list that contains all of the involved animals. This goes hand in hand with isFriend flag which will be critical on establishing the relationships among animals.
- Below are the Model Relationship which is the summary of the above details

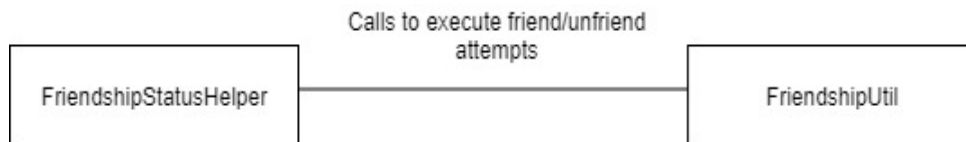


Helpers

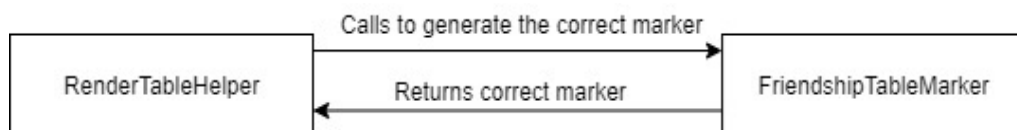
- Responsible for executing logic and algorithms that are essential to the application
- Divided into 4 classes which is specialized on very specific behaviour that is needed based on the expected activity of the animals in the application:
 - InitializationHelper
 - Responsible for initializing and displaying the information of the Animal Models
 - Responsible for establishing the initial relationship of each of the Animals
 - This is done by setting the friendship flag to true between Animal and its Best Friend Forever and setting the friendship between Animal and itself
 - Responsible on sorting and grouping the animals during their Lunchtime
 - FriendshipStatusHelper
 - Responsible for establishing attempts of each animal to befriend or unfriend another animals
 - RenderLunchTimeHelper
 - Responsible for displaying the animals that are grouped for lunch
 - Each grouped animals will be displayed eating their common food
 - RenderTableHelper
 - Responsible for printing and displaying the table properly
 - Responsible for displaying the animal relationships properly and correctly in the table
 - Usually executed during end of the day

Utilities

- Utility Class takes care algorithm heavy behaviours such as probability, which animal to befriend or unfriend and providing correct friendship-markers on the table.
 - FriendshipUtil
 - Responsible for executing algorithms that will check if the friendship can be established if a friendship/unfriendship attempt can be done by an animal to another animal
 - This utility class goes hand and hand with the FriendshipStatusHandler class

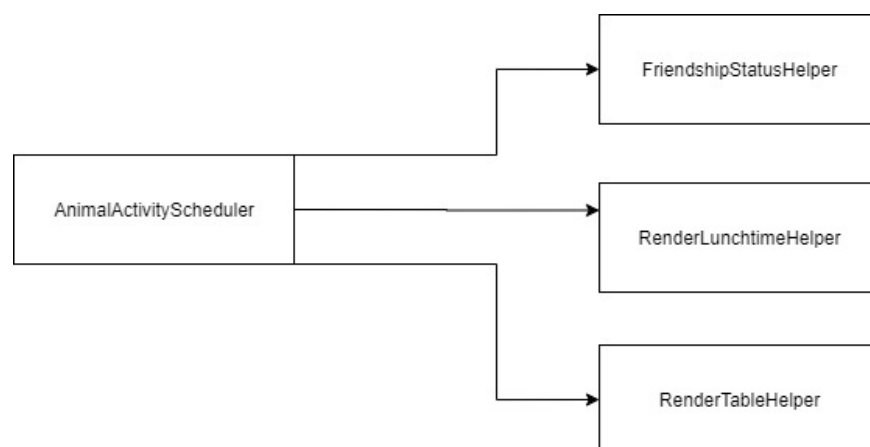


- FriendshipTableMarkerUtil
 - Responsible for providing the correct marker on the friendship table which is displayed by end of the day
 - This utility class goes hand and hand with the RenderTableHelper



Scheduler

- Scheduler's are classes that are responsible for organizing the activity of each animals based on the time of the day
- They are responsible for calling the Helpers on executing the behaviors of the application
 - AnimalActivityScheduler
 - Responsible for setting the day and time and executing specific helper class based on the time of the day



Main Client

- Main client of the application(MainAppClient) which is responsible for calling the initialization class and the scheduler class that will be shown to the user.
- This will call the InitializationHelper class to initialize the needed objects that will be used by the next behaviour
- Executes the AnimalActivityScheduler which is the main behaviour of the application

