Test assignment

The following animals live on the village yard:

1. Dog: “Rex”
   Breed: “Shepherd”
   Favorite food: “Royal Canin”
   Best friend forever : “Tom”

2. Dog “Max”
   Breed: “Shepherd”
   Favorite food: “Purina ONE”
   Best friend forever : “Jay”

3. Dog “Tom”
   Breed: “Husky”
   Favorite food: “Royal Canin”
   Best friend forever : “Rex”

4. Dog “Jay”
   Breed: “Husky”
   Favorite food: “Purina ONE”
   Best friend forever : “Max”

5. Cat “Zoe”
   Favorite food: “9Lives”
   Best friend forever : “Ada”

6. Cat “Ada”
   Favorite food: “Purina Friskies”
   Best friend forever : “Zoe”

7. Chicken “Meg”
   Favorite food: “Purina Layena”
   Lays eggs: yes
   Wingspan: 0.4m
   Best friend forever : “Lis”

8. Chicken “Lis”
   Favorite food: “Manna Pro”
   Lays eggs: yes
   Wingspan: 0.35m
   Best friend forever : “Meg”

9. Chicken “Emi”
   Favorite food: “Purina Layena”
   Lays eggs: no
   Wingspan: 0.25m
   Best friend forever : “Lua”

10. Chicken “Lua”
    Favorite food: “Manna Pro”
    Lays eggs: no
    Wingspan: 0.3m
    Best friend forever : “Emi”

11. Rooster “Bob”
    Favorite food: “Manna Pro”
    Wingspan: 0.5m

12. Parrot “Mac”
    Favorite food: “Lafeber Original”
    Wingspan: 0.33m
    Can speak: yes
    Best friend forever : “Alf”

13. Parrot “Alf”
    Favorite food: “Kaytee Fiesta”
    Wingspan: 0.25m
    Can speak: no
    Best friend forever : “Mac”

The task is to write a "10 days animals live simulation" console application.

An animal has a set of attributes and can establish or terminate friendship with few other animals from the village yard.

On start-up app prints out all animals with their attributes.

The day starts with a message: “Day-N” (N from 1 to 10).
Every day each animal can unfriend one animal before lunchtime. If animal “A” unfriends animal “B”, “B” unfriends “A” automatically. App prints out all “unfriend” events.
For example: “Alf and Bob are not friends anymore”.

During lunchtime, the app prints out each animal’s favorite food grouped by the brand.
For example: “Rex and Tom are eating Royal Canin”.

After lunch, each animal tries to establish a friendship with a random animal. Animal “A”, asks animal “B” to become friends. If “B” accepts friendship, “A” and “B” become friends. App prints out the result, for example:
“Alf is asking Bob to be friends. Alf and Bob are friends now. (Or Bob doesn’t want to be friends)”

At the end of the day the app prints out friendship table for all animals.
For example, “A” has a friendship with “B” and “C”, “B” and “C” are not friends:

```
---------------------
|   | A | B | C |...|
| A | \ | X | X |   |
| B | X | \ |   |   |
| C | X |   | \ |   |
|...|   |   |   | \ |
---------------------
```

Friendship rules:

1. “New friend” is a random animal from the village yard.
2. “Best friend forever” never unfriended.
3. If an animal has 3 or more friends, the probability to lose friend 90%, the probability to get a new friend 10%
4. If an animal has 2 or fewer friends, the probability to lose friend 10%, the probability to get a new friend 90%
5. Rules 2-4 applied for initiating friend requests and for answering as well.

Technical requirements:
Java 8 or newer, No third-party libraries except JUnit, Maven project config (pom.xml file)
Unit tests are not required, but welcome.
Write a short design document describing the most important aspects (PDF or TXT format).

Notes:
If you think some part of the test assignment is unclear or there is a mistake, don't worry. Decide for yourself what would be a logical or fun thing to do, and shortly explain it in the design document.
Object-oriented design and high-quality coding will be deciding factors.