CAANES SYSTEM
Alvaro Mancera
Celeto Ogrimen
Gloria Gallego

Project : Caanes System

15th April 2017

OVERVIEW

The Veterinary Clinic System called CAANES (Specialized animal care center) it is a veterinary care for domestic animals. Since its inception a paper-based system has been used by the owner. However with it's growing demands it is necessary for a computerized system to be used in order to keep tracks of the pets as well as their medical history.

GOALS

1. The system need to have the principal data **information for the Animal**.
2. The system need to be able to create different **type of species** of animals and **breed.**
3. The system need to have the principal data **information for the Owner**.
4. The system need to have the principal data **information for the Doctors**.
5. The system need to have **multiple contact location** for the animal's owner and Doctors.
6. The system need to have **multiple contact phone** for the animal's owner and Doctors.
7. The system need to have **multiple contact email** for the animal's owner and Doctors.
8. The system need to be able to associate a **microchip with the Animal.**
9. The system need to have the record of the **animal's vaccines**.
10. The system need to have the record of the **animal's deworming.**

SPECIFICATIONS

Using Java programming language with object oriented concepts a new web system will be built

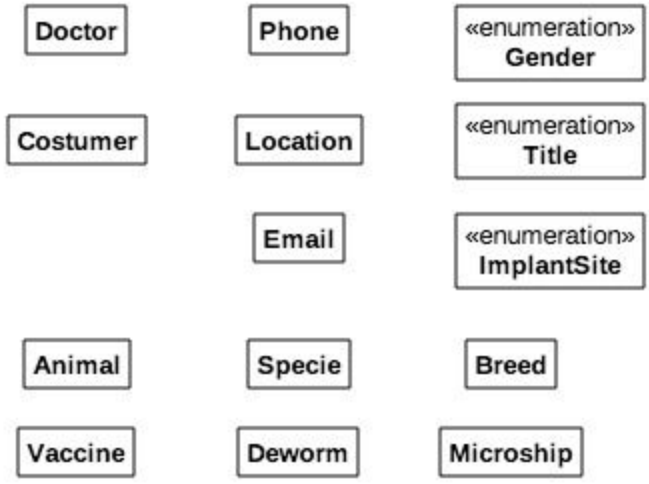
The tools that we use was:

- Java 8
- MySQL
- Spring MVC Framework
- Maven for managing Libraries and Components
- Tomcat 7 as Web Application Server

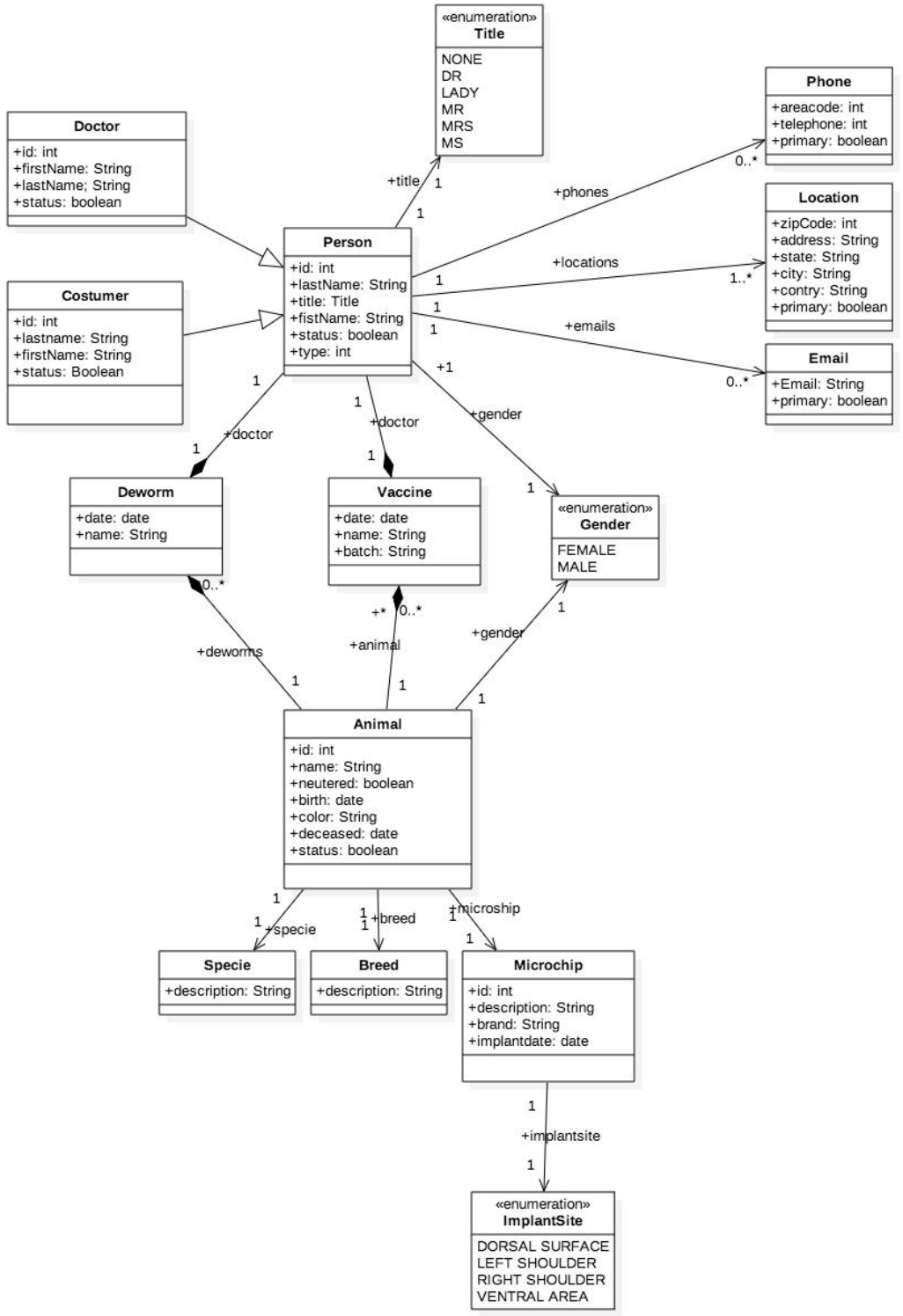
- github.

MILESTONES

Identify Candidate Classes(Concepts) and Attributes.



Create Class Diagram (Inheritance and Associations)

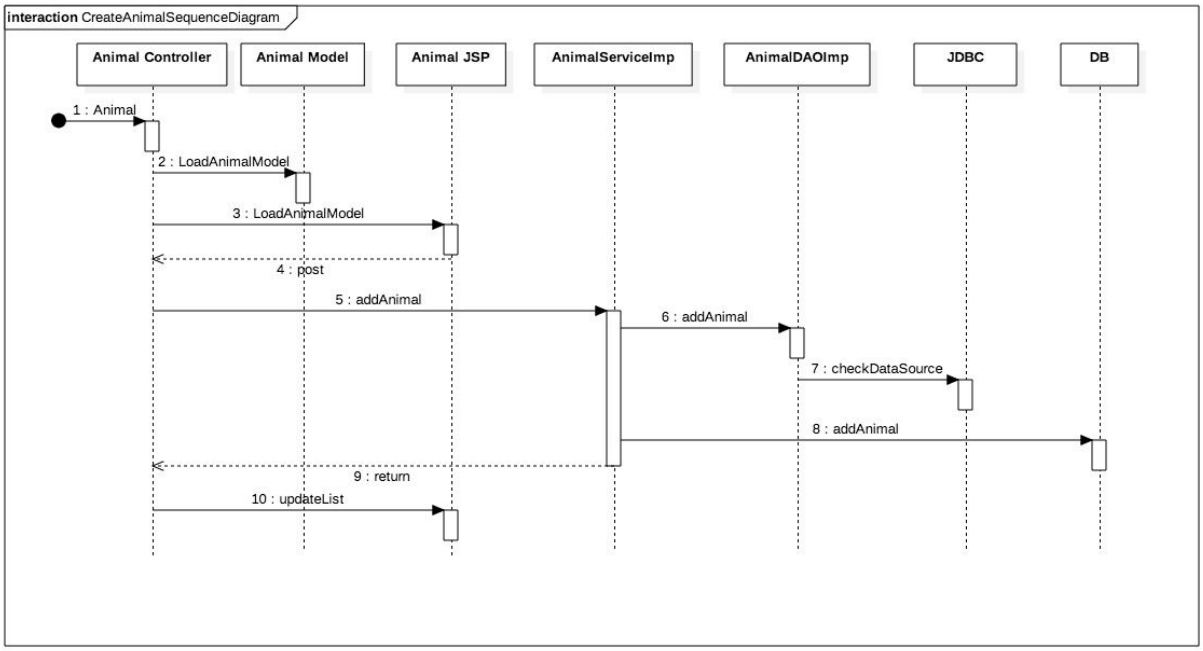


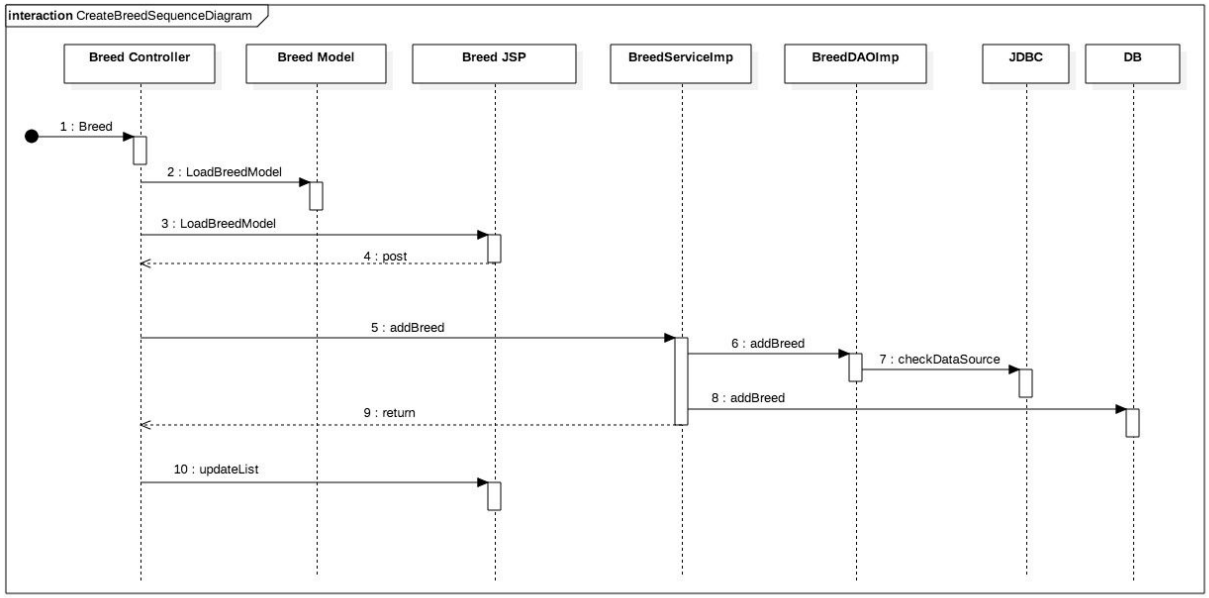
Data Dictionary

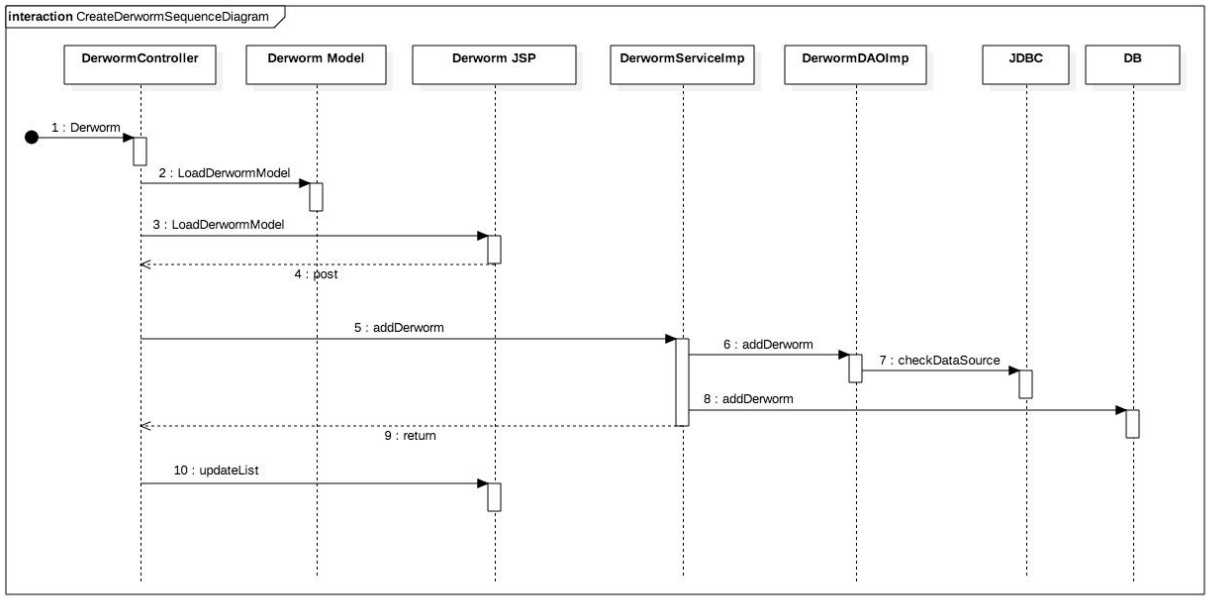
#	CLASS	DESCRIPTION
1	Doctor	A doctor is a person who works in the Clinic. This person need to be associate with the vaccine, microchips and Deworm.
2	Costumer	The owner of the animal.
3	Person	The basic information for the person (the information that share the Doctor or Costumer)
4	Phone	The basic information for the contact phone for a person.
5	Location	The basic information for the contact location for a person.
6	Email	The basic information for the email phone for a person.
7	Animal	A Animal is a the patient for the clinic.
8	Deworm	The clinic need to have the record of each deworm that give to each animal.
9	Vaccine	The clinic need to have the record of each vaccine that give to each animal.
10	Specie	All the list of possible Species that the clinic can treat.
11	Breed	All the list of possible Breed that the clinic can treat.
12	Microchip	The clinic need to have the record of the microchip that implant to each animal.

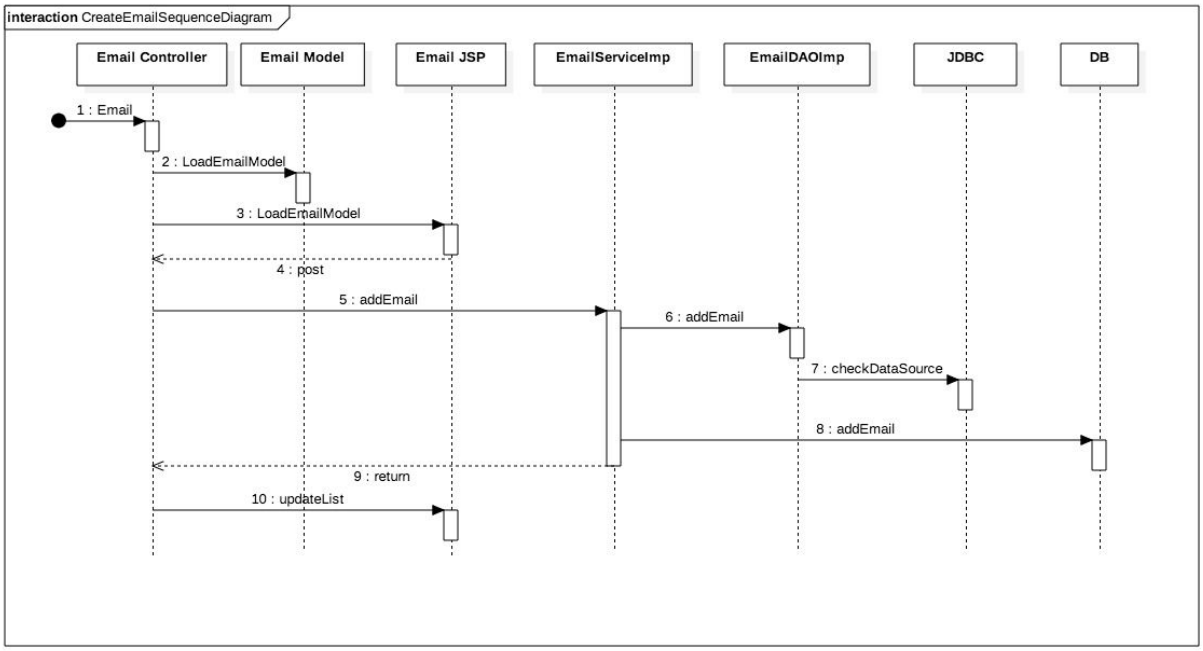
#	ENUM	DESCRIPTION
1	Tittle	The title for the person (Mr, Mrs...)
2	Gender	Female or Male
3	ImplantSite	The Site for the implant of the microchip in the body of the animal. (Dorsal, Shoulder...)

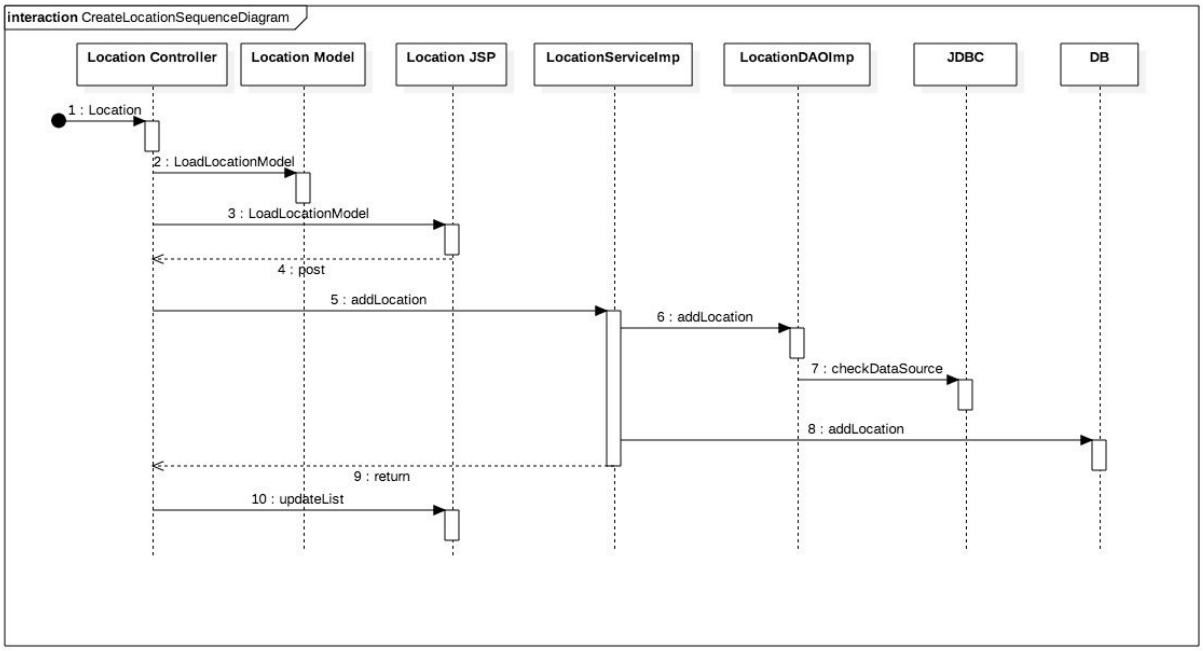
Create Sequence Diagrams

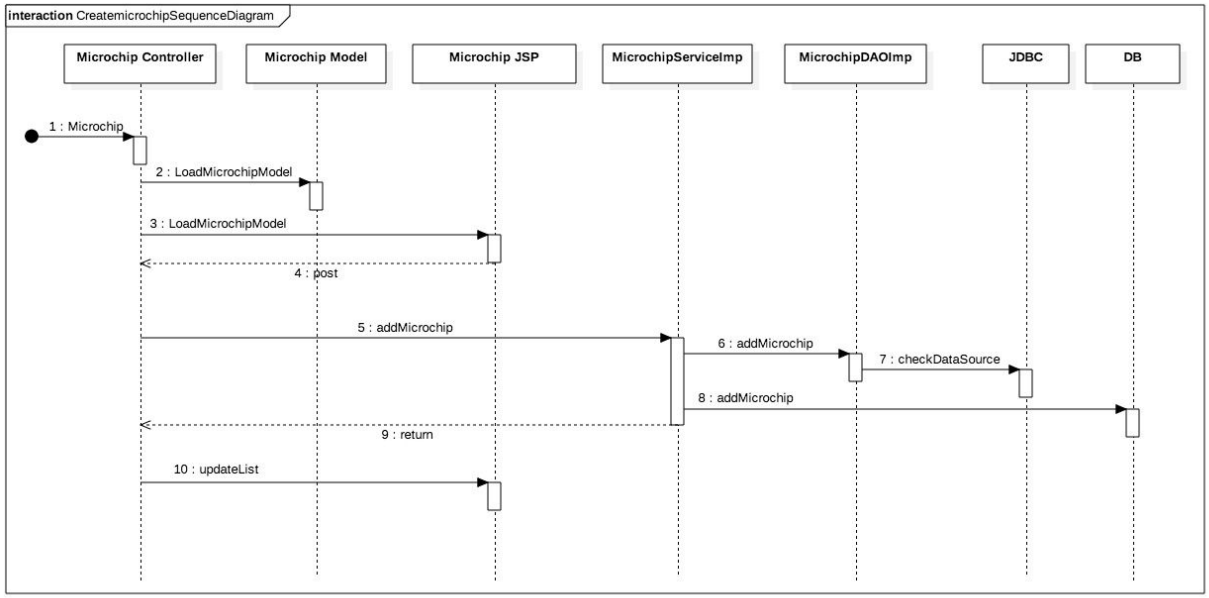


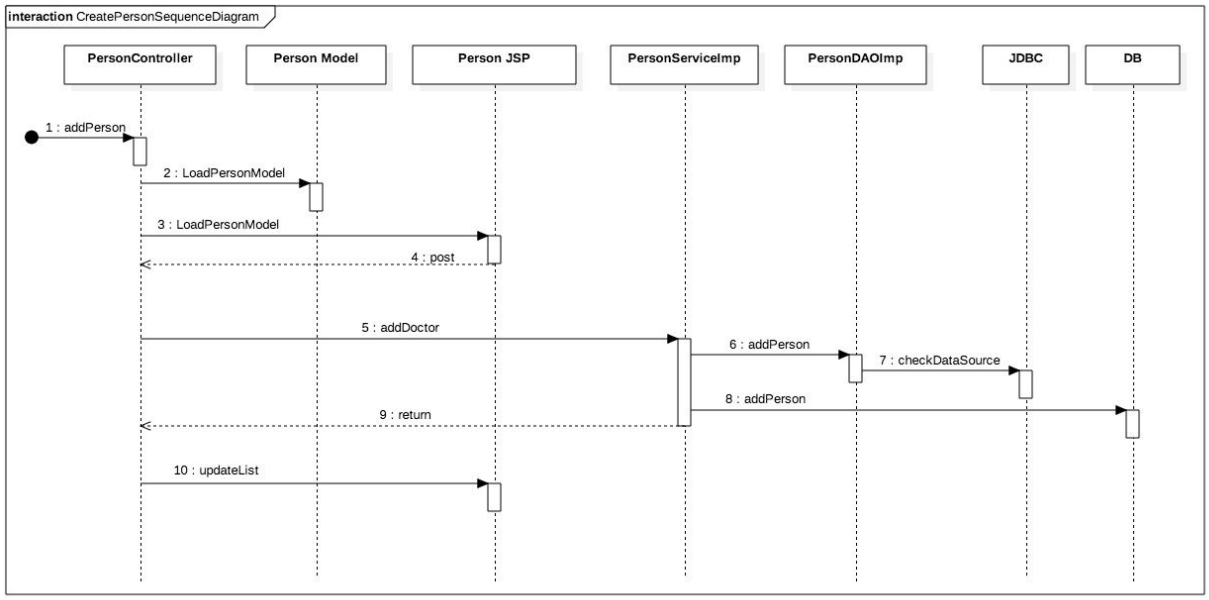

interaction CreateBreedSequenceDiagram
Breed Controller
Breed Model
Breed JSP
BreedServiceImp
BreedDAOImp
JDBC
DB
1 : Breed
2 : LoadBreedModel
3 : LoadBreedModel
4 : post
5 : addBreed
6 : addBreed
7 : checkDataSource
8 : addBreed
9 : return
10 : updateList

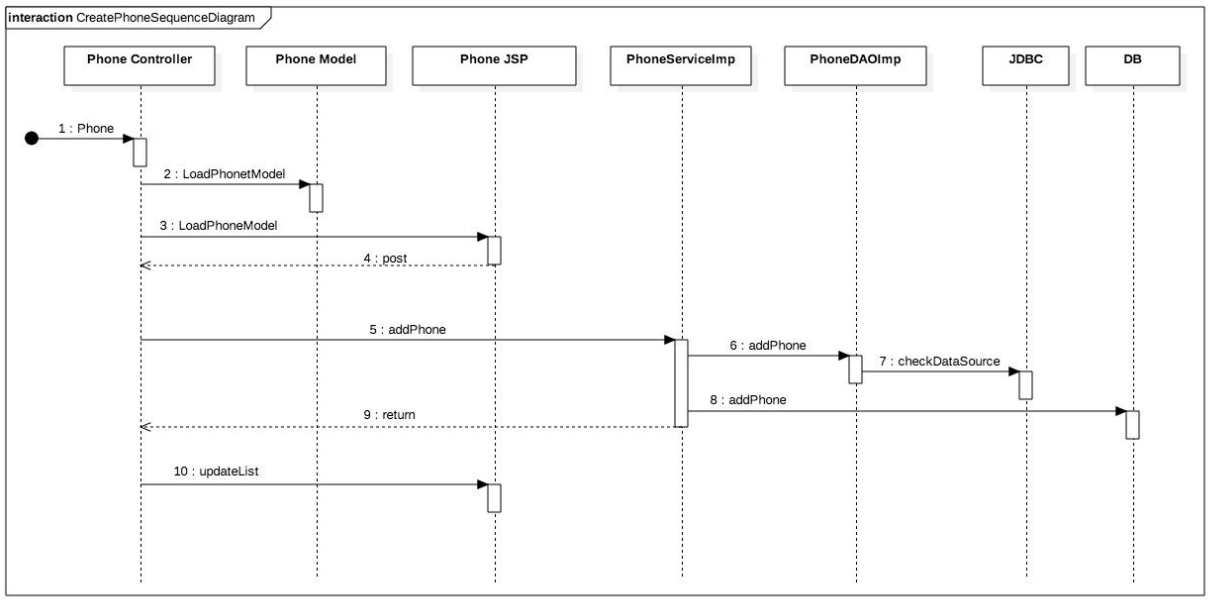

interaction CreateDerwormSequenceDiagram
DerwormController
Derworm Model
Derworm JSP
DerwormServiceImp
DerwormDAOImp
JDBC
DB
1 : Derworm
2 : LoadDerwormModel
3 : LoadDerwormModel
4 : post
5 : addDerworm
6 : addDerworm
7 : checkDataSource
8 : addDerworm
9 : return
10 : updateList

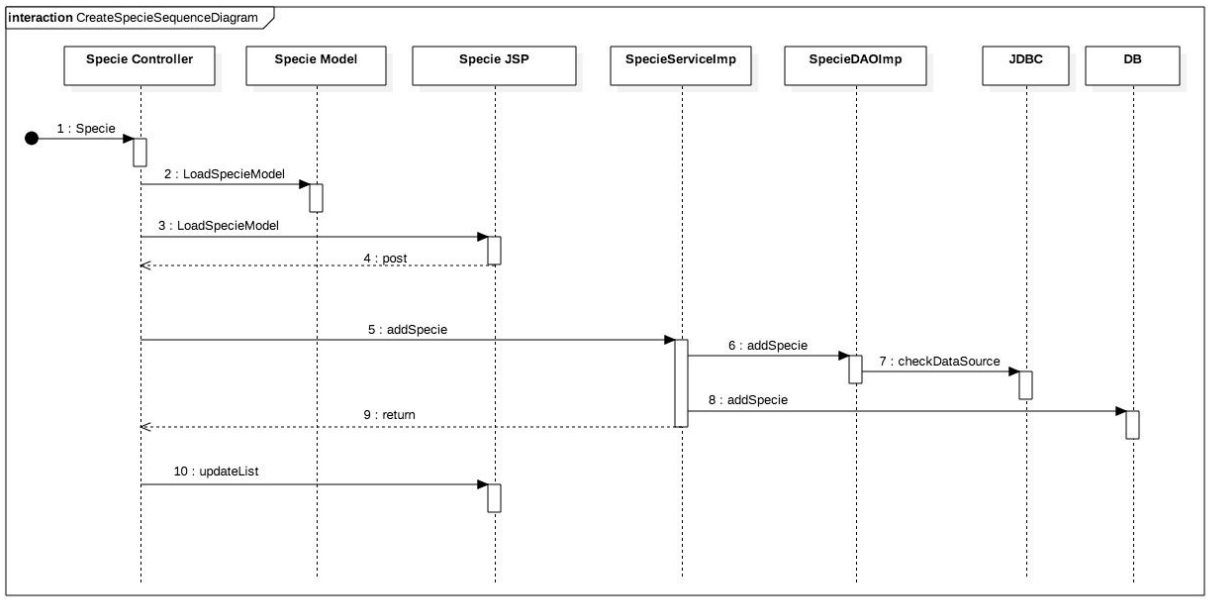

interaction CreateEmailSequenceDiagram
Email Controller
Email Model
Email JSP
EmailServiceImp
EmailDAOImp
JDBC
DB
1 : Email
2 : LoadEmailModel
3 : LoadEmailModel
4 : post
5 : addEmail
6 : addEmail
7 : checkDataSource
8 : addEmail
9 : return
10 : updateList

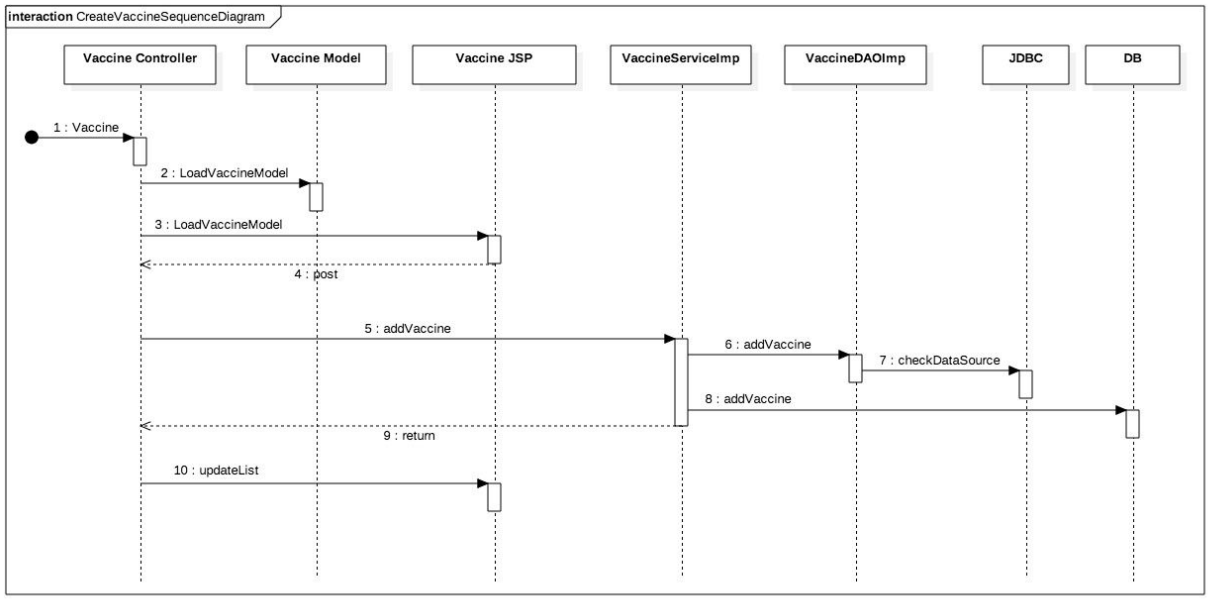

interaction CreateLocationSequenceDiagram
Location Controller
Location Model
Location JSP
LocationServiceImp
LocationDAOImp
JDBC
DB
1 : Location
2 : LoadLocationModel
3 : LoadLocationModel
4 : post
5 : addLocation
6 : addLocation
7 : checkDataSource
8 : addLocation
9 : return
10 : updateList


interaction CreatemicrochipSequenceDiagram
Microchip Controller
Microchip Model
Microchip JSP
MicrochipServiceImp
MicrochipDAOImp
JDBC
DB
1 : Microchip
2 : LoadMicrochipModel
3 : LoadMicrochipModel
4 : post
5 : addMicrochip
6 : addMicrochip
7 : checkDataSource
8 : addMicrochip
9 : return
10 : updateList

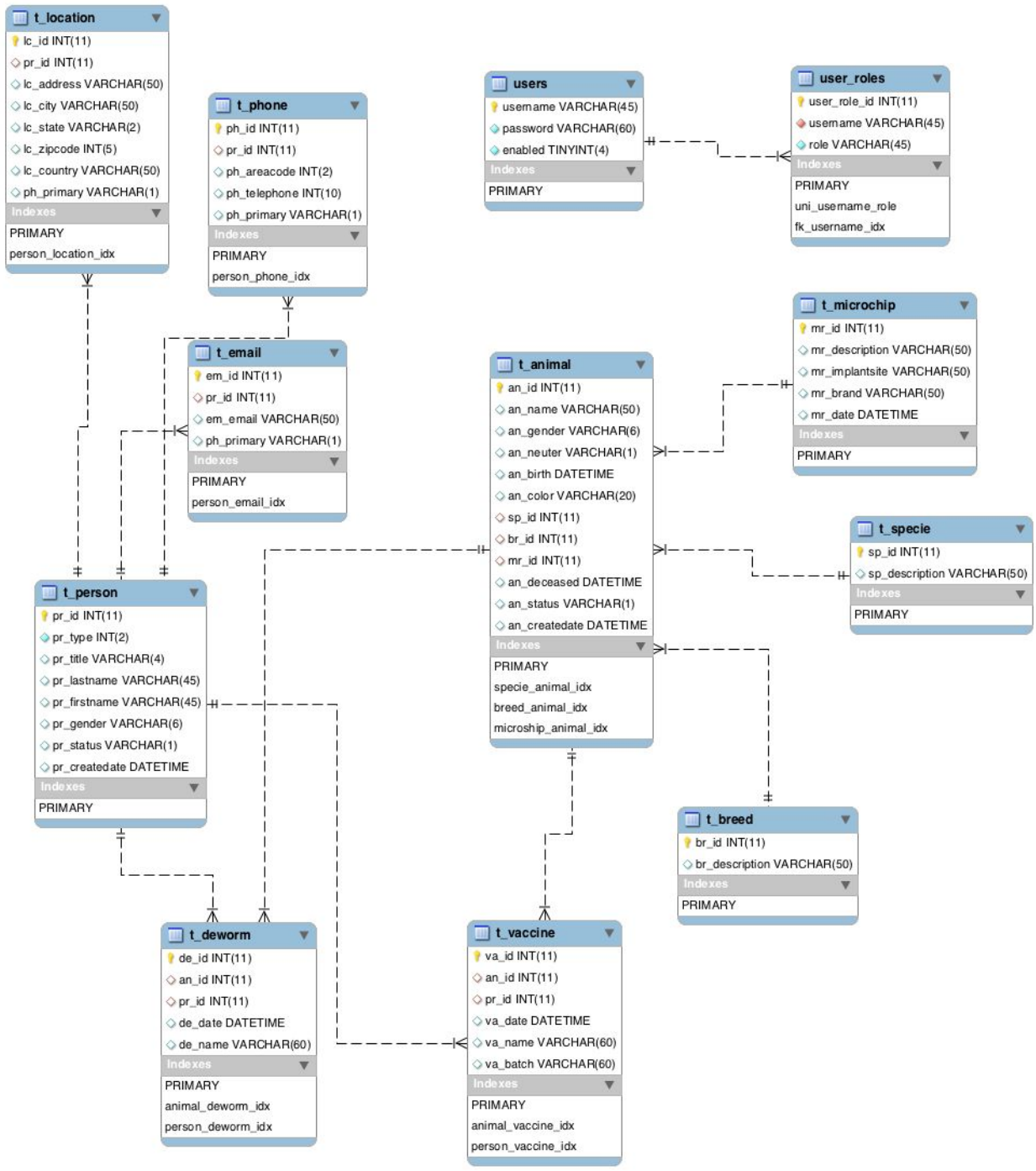

interaction CreatePersonSequenceDiagram
PersonController
Person Model
Person JSP
PersonServiceImp
PersonDAOImp
JDBC
DB
1 : addPerson
2 : LoadPersonModel
3 : LoadPersonModel
4 : post
5 : addDoctor
6 : addPerson
7 : checkDataSource
8 : addPerson
9 : return
10 : updateList


interaction CreatePhoneSequenceDiagram
Phone Controller
Phone Model
Phone JSP
PhoneServiceImp
PhoneDAOImp
JDBC
DB
1 : Phone
2 : LoadPhonetModel
3 : LoadPhoneModel
4 : post
5 : addPhone
6 : addPhone
7 : checkDataSource
8 : addPhone
9 : return
10 : updateList


interaction CreateSpecieSequenceDiagram
Specie Controller
Specie Model
Specie JSP
SpecieServiceImp
SpecieDAOImp
JDBC
DB
1 : Specie
2 : LoadSpecieModel
3 : LoadSpecieModel
4 : post
5 : addSpecie
6 : addSpecie
7 : checkDataSource
8 : addSpecie
9 : return
10 : updateList


interaction CreateVaccineSequenceDiagram
Vaccine Controller
Vaccine Model
Vaccine JSP
VaccineServiceImp
VaccineDAOImp
JDBC
DB
1 : Vaccine
2 : LoadVaccineModel
3 : LoadVaccineModel
4 : post
5 : addVaccine
6 : addVaccine
7 : checkDataSource
8 : addVaccine
9 : return
10 : updateList

DataBase Diagram



Implementation

See the folder attached.