Software Requirements and Design Document

for

Veterinary Hospital and Shelter Management System

Fiza Jameel (22i-0964)

Sahrish Mustafa (22i-0977)

Hadiya Tanveer (22i-1113)

TrigSync Devs

26 November 2024

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this system is to take the hassle out of managing veterinary hospitals and shelters. By offering an easy-to-use platform, it helps veterinarians, shelter staff, and even pet owners work together seamlessly. From tracking medical histories to scheduling appointments and overseeing shelter operations, this system is built to make things easier and more efficient.

1.2 Product Scope

This system is designed to handle all the essential tasks of veterinary hospitals and shelters. It helps veterinarians manage medical records, ensures appointments are scheduled without overlaps, and keeps track of animals in shelters. Additionally, the system provides features for generating travel certificates for pets, enabling online checkups, and allowing users to refer friends to the hospital or shelter. The goal is to provide a one-stop solution that everyone can rely on.

1.3 Title

Veterinary Hospital and Shelter Management System

1.4 Objectives

The system aims to simplify scheduling and record-keeping, making it easier for staff to manage daily operations efficiently. It helps shelters track animals and streamline adoption processes while enhancing communication between staff and pet owners. By providing tools to better organize resources and processes, the system ensures higher-quality care for animals and improves overall management.

1.5 Problem Statement

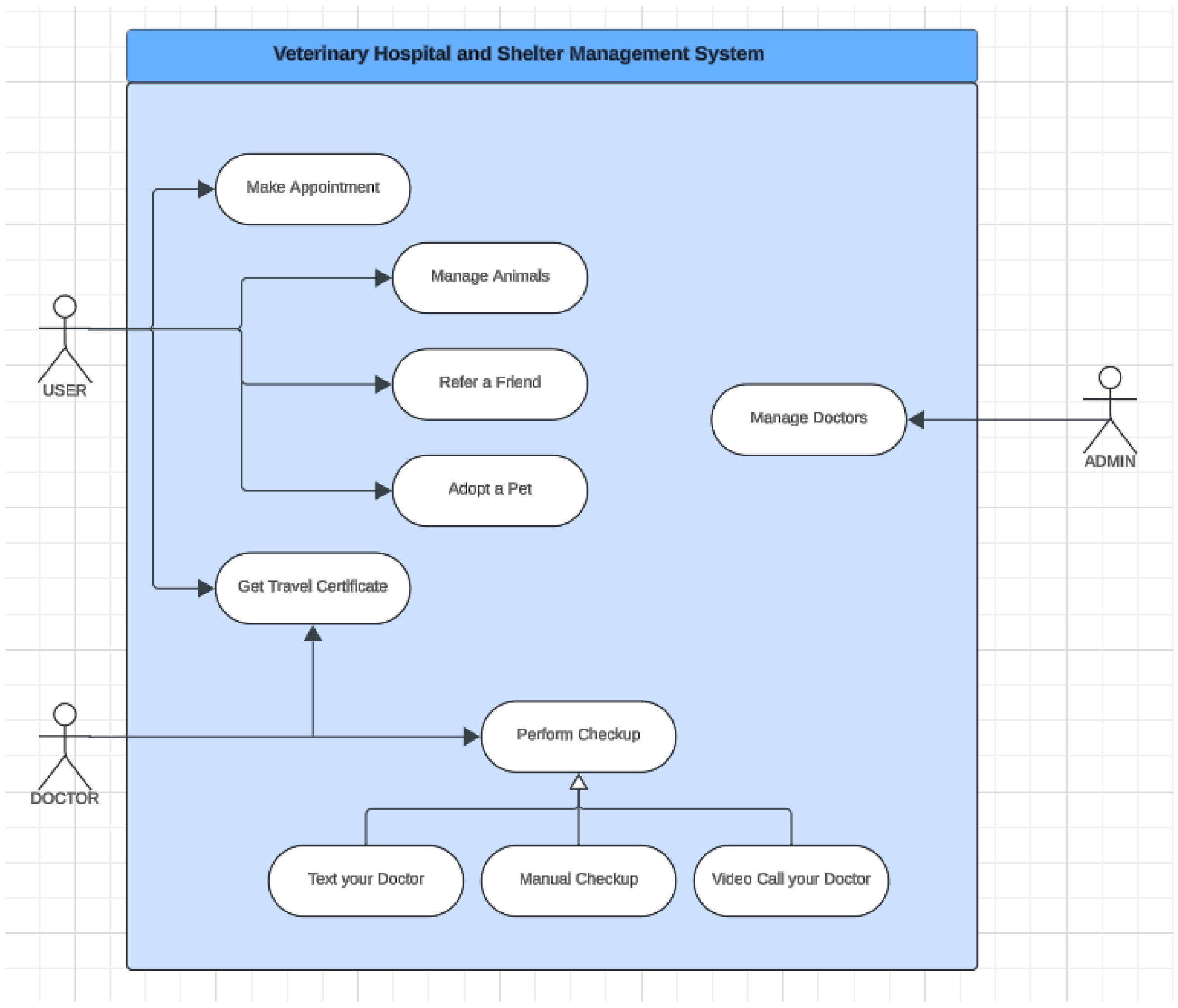
Running a veterinary hospital or a shelter can be chaotic when everything is done manually. It's easy to lose track of important details, whether it's a pet's medical history or a shelter animal's adoption status. This often results in unnecessary delays and poor service. By introducing a digital solution, we can simplify these operations, save time, and improve care for the animals who need it most.

2. Overall Description

2.1 List of Use Cases

- Manage Animals
- Adopt a Pet
- Book Appointment
- Refer a Friend
- Get a Travel Certificate
- Manage Animals
- Manage Doctors
- Get a Checkup

2.2 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

The system must handle multiple users simultaneously, ensuring quick access to features like records and appointments with a response time of under 3 seconds. It should process large datasets efficiently, even during peak usage.

3.2 Safety Requirements

Data integrity and reliability are crucial, with backup mechanisms in place to prevent loss due to hardware failures or accidental deletions. The system must operate consistently without crashing.

3.3 Software Quality Attributes

The system will be user-friendly, reliable, and scalable to accommodate growth. It will also support maintainability, ensuring easy updates and minimal disruptions.

3.4 Business Rules

Appointments require mutual confirmation, travel certificates can only be issued by authorized staff, and shelter animals must pass health checks before adoption. Referral bonuses will be validated before application.

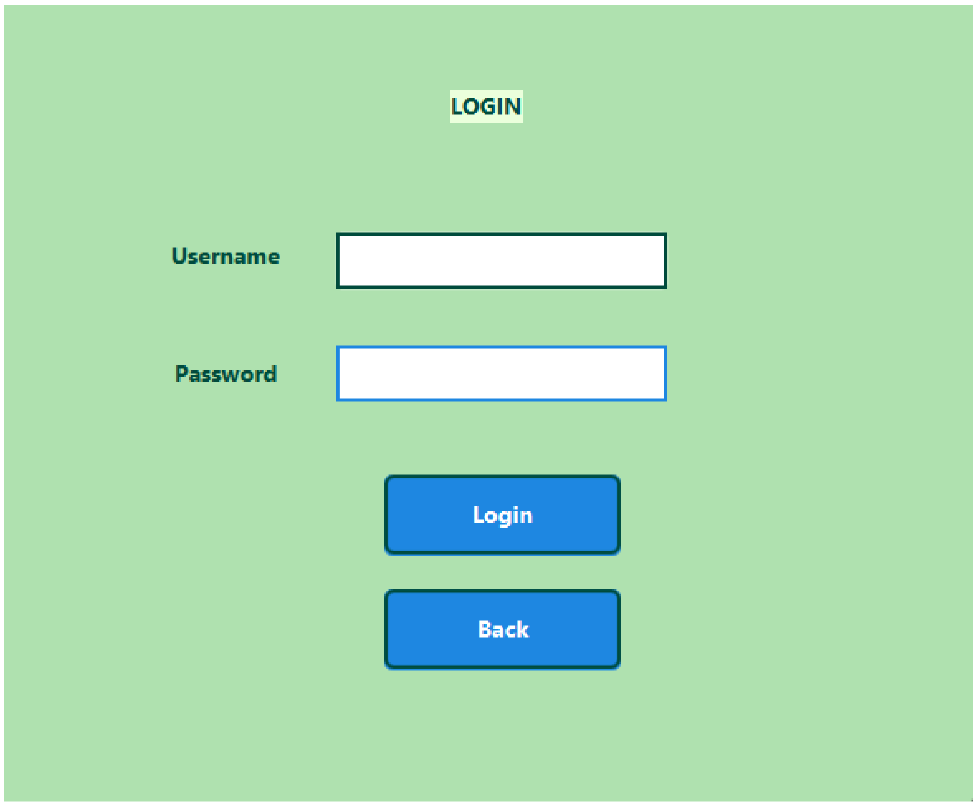
3.5 Operating Environment

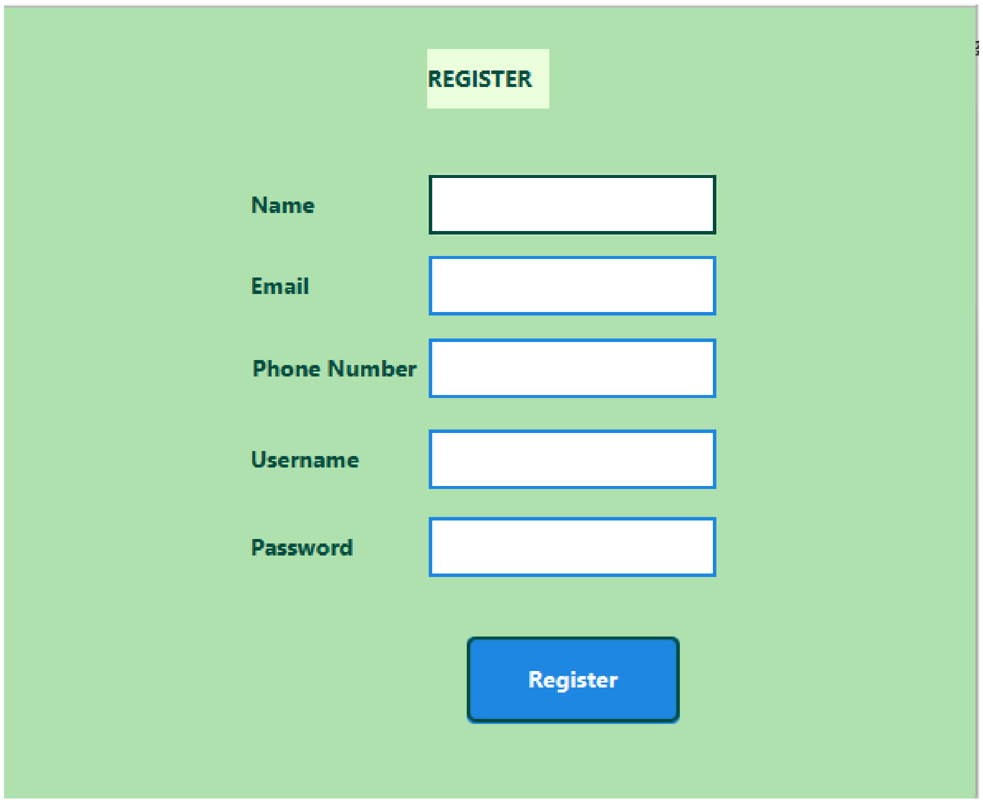
The Veterinary System application is designed to operate on modern hardware platforms with minimum requirements of a dual-core CPU, 4 GB RAM, and 500 MB of free disk space. It supports Windows 10 or later, macOS Mojave or later. The application is built using Java 17 and JavaFX for the user interface, with MySQL 8.0 for database management and JDBC for data connectivity.

3.6 User Interfaces

The interfaces include Login/Register interface, User interface, Doctor interface, Admin Interface.
User interface has book an appointment, request a travel certificate, adopt a pet and refer a friend interfaces.
Doctor interface has checking its appointments, performing/approving checkup and approving travel certificates.
Admin interface has added the doctor.

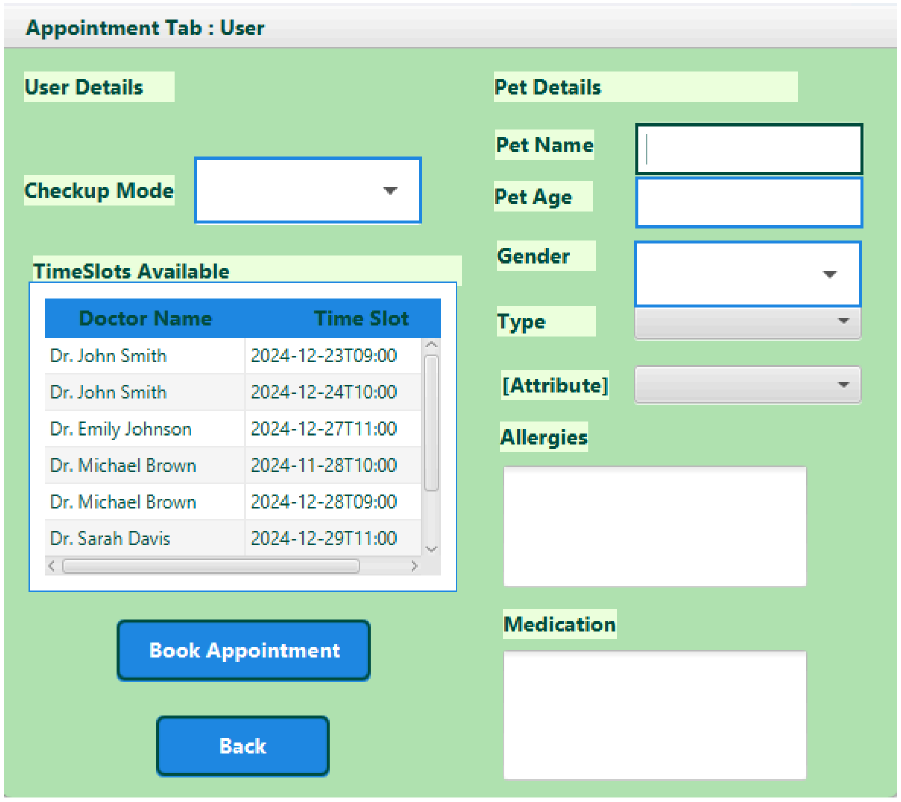
i. Login and Register Interfaces





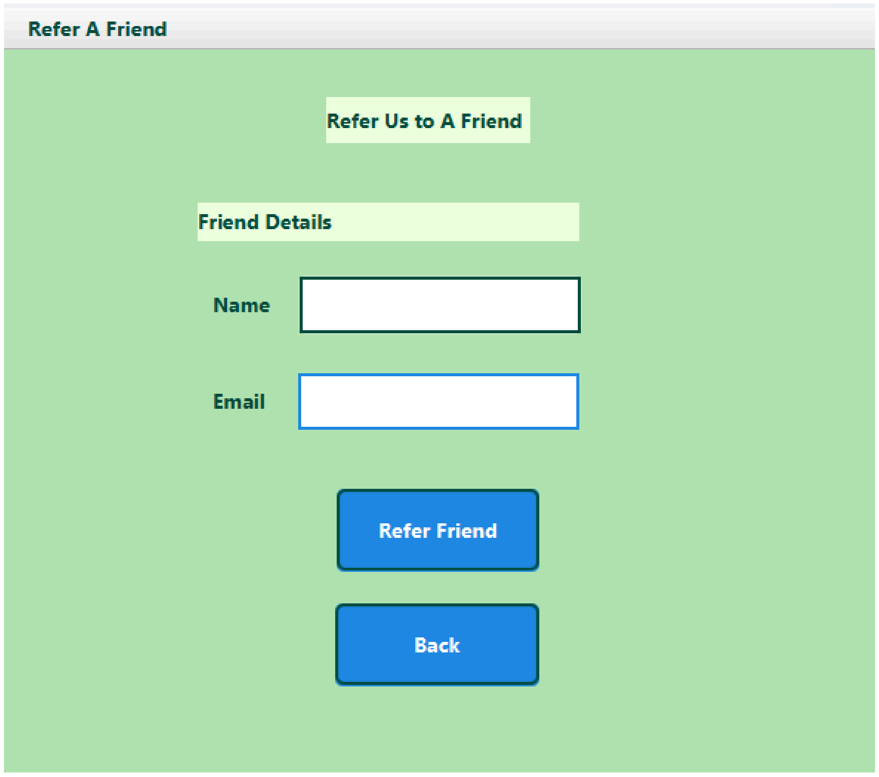
ii. User Interface

a. Book an appointment

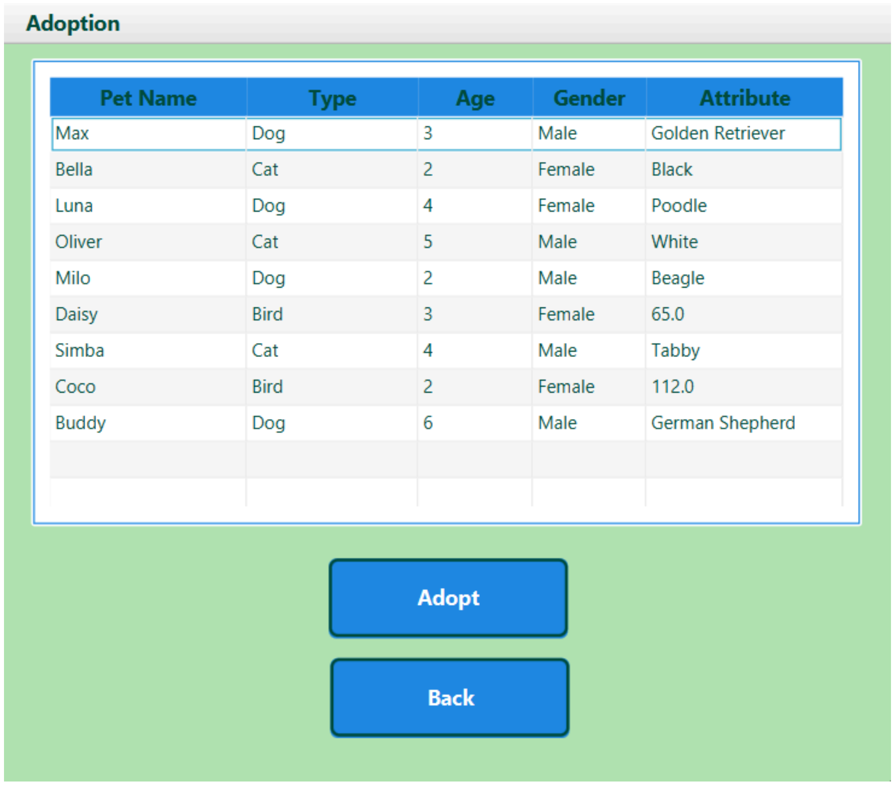


Doctor Name	Time Slot
Dr. John Smith	2024-12-23T09:00
Dr. John Smith	2024-12-24T10:00
Dr. Emily Johnson	2024-12-27T11:00
Dr. Michael Brown	2024-11-28T10:00
Dr. Michael Brown	2024-12-28T09:00
Dr. Sarah Davis	2024-12-29T11:00

b. Refer a Friend

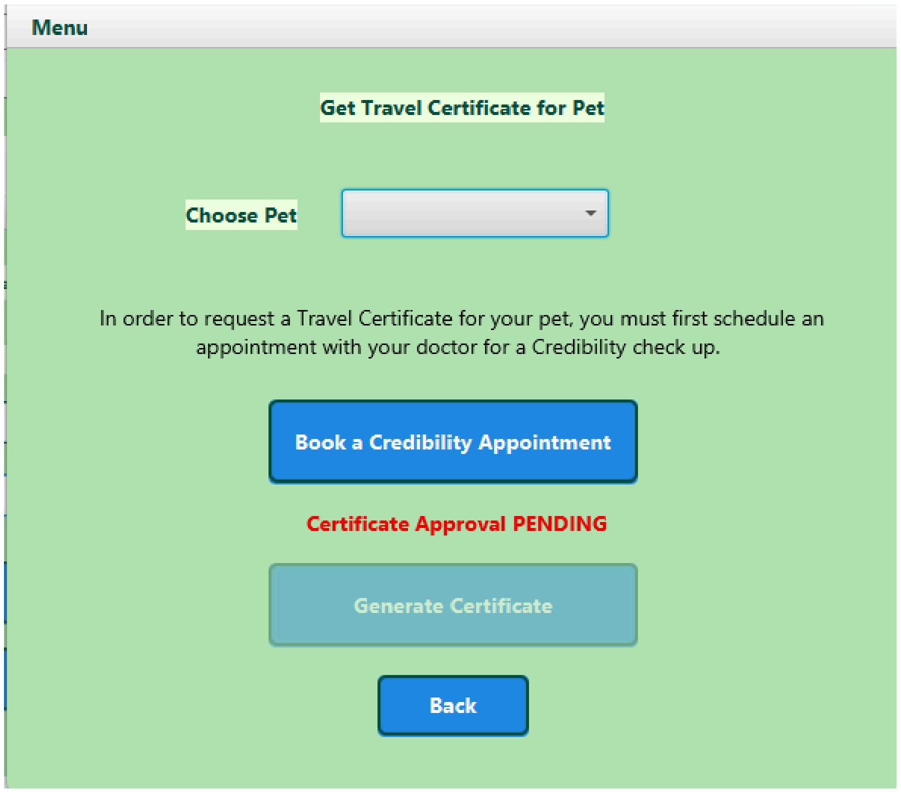


c. Adopt an animal



Pet Name	Type	Age	Gender	Attribute
Max	Dog	3	Male	Golden Retriever
Bella	Cat	2	Female	Black
Luna	Dog	4	Female	Poodle
Oliver	Cat	5	Male	White
Milo	Dog	2	Male	Beagle
Daisy	Bird	3	Female	65.0
Simba	Cat	4	Male	Tabby
Coco	Bird	2	Female	112.0
Buddy	Dog	6	Male	German Shepherd

d. Get travel certificate

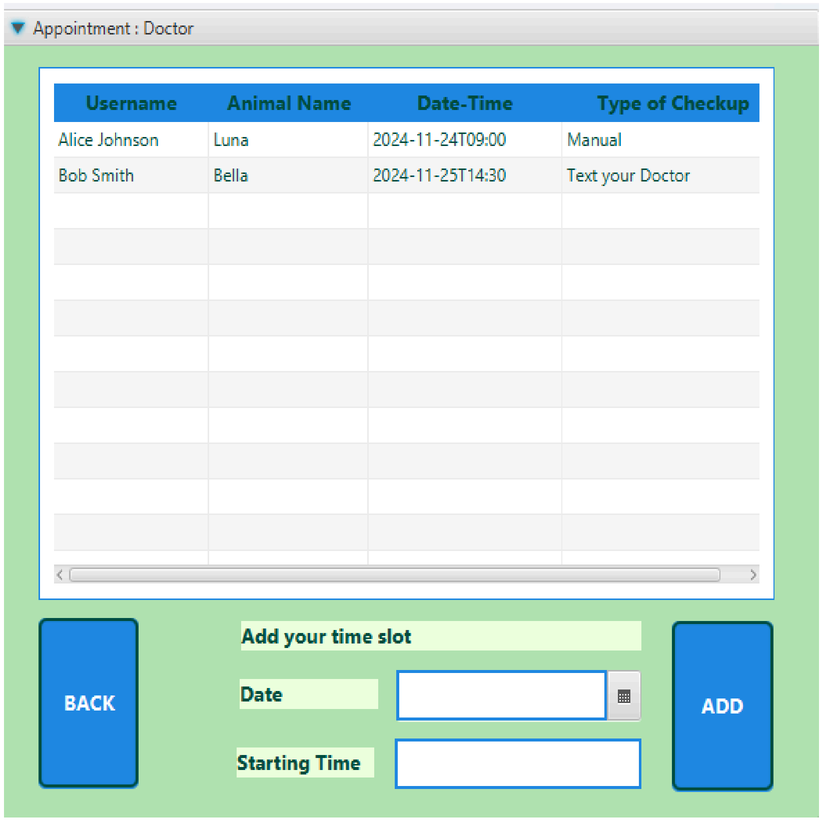

Menu
Get Travel Certificate for Pet
Choose Pet
Book a Credibility Appointment
Generate Certificate
Back

In order to request a Travel Certificate for your pet, you must first schedule an appointment with your doctor for a Credibility check up.

Certificate Approval PENDING

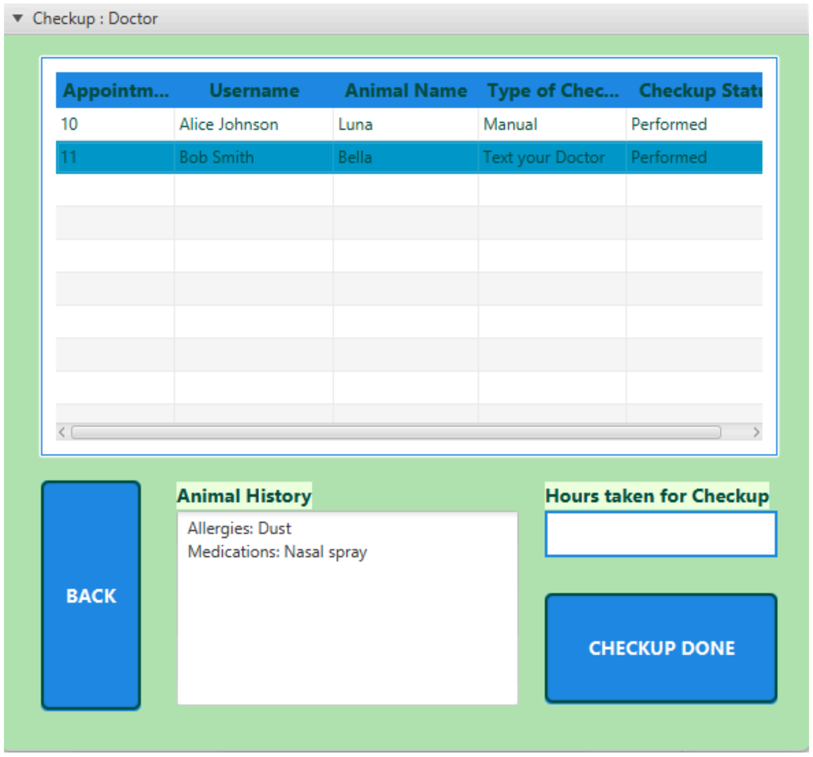
iii. Doctor Interface

a. Check appointments



Username	Animal Name	Date-Time	Type of Checkup
Alice Johnson	Luna	2024-11-24T09:00	Manual
Bob Smith	Bella	2024-11-25T14:30	Text your Doctor

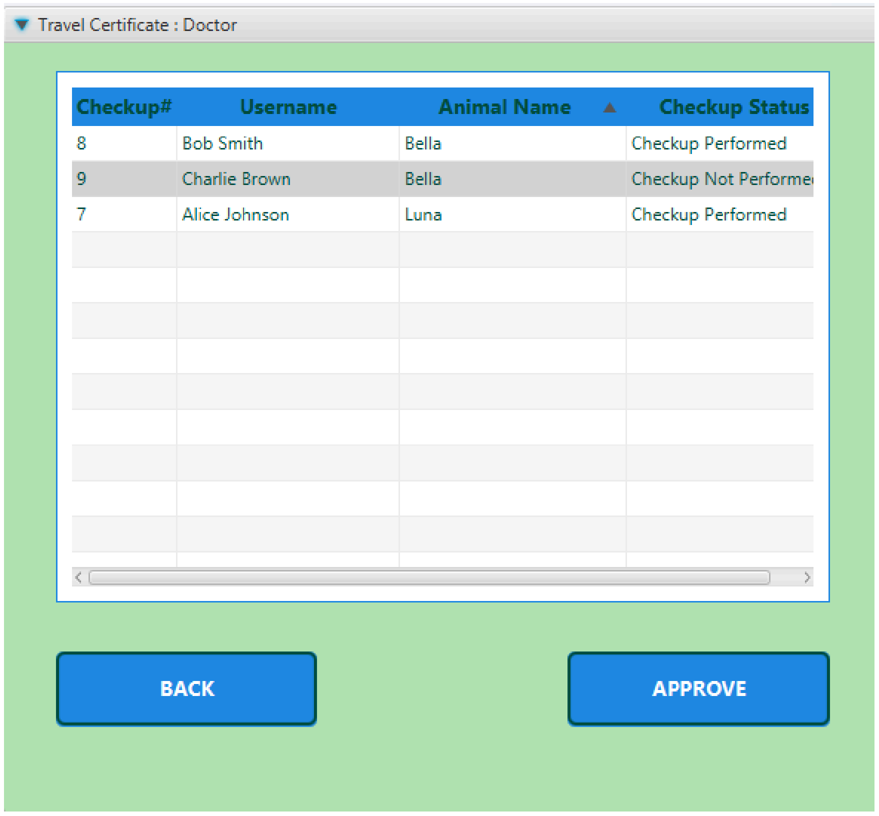
b. Perform Checkups



Appointm...	Username	Animal Name	Type of Chec...	Checkup Stat
10	Alice Johnson	Luna	Manual	Performed
11	Bob Smith	Bella	Text your Doctor	Performed

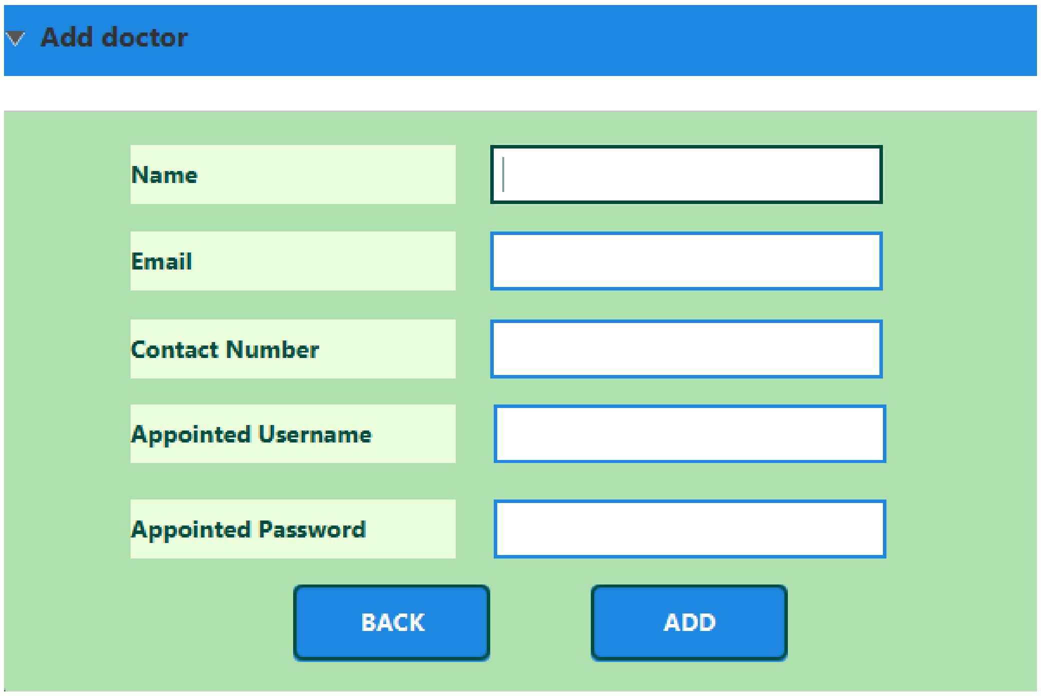
Allergies: Dust
Medications: Nasal spray

c. Approve Travel certificate

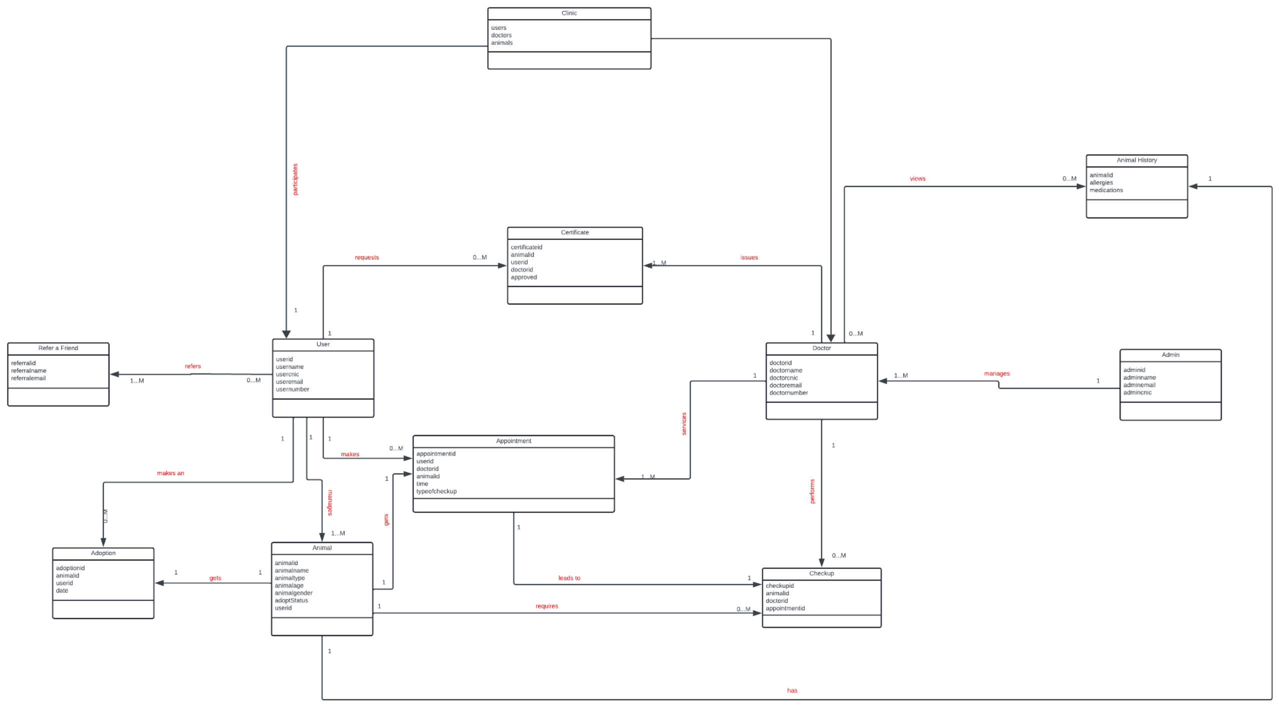


Checkup#	Username	Animal Name	Checkup Status
8	Bob Smith	Bella	Checkup Performed
9	Charlie Brown	Bella	Checkup Not Performe
7	Alice Johnson	Luna	Checkup Performed

iv. Admin

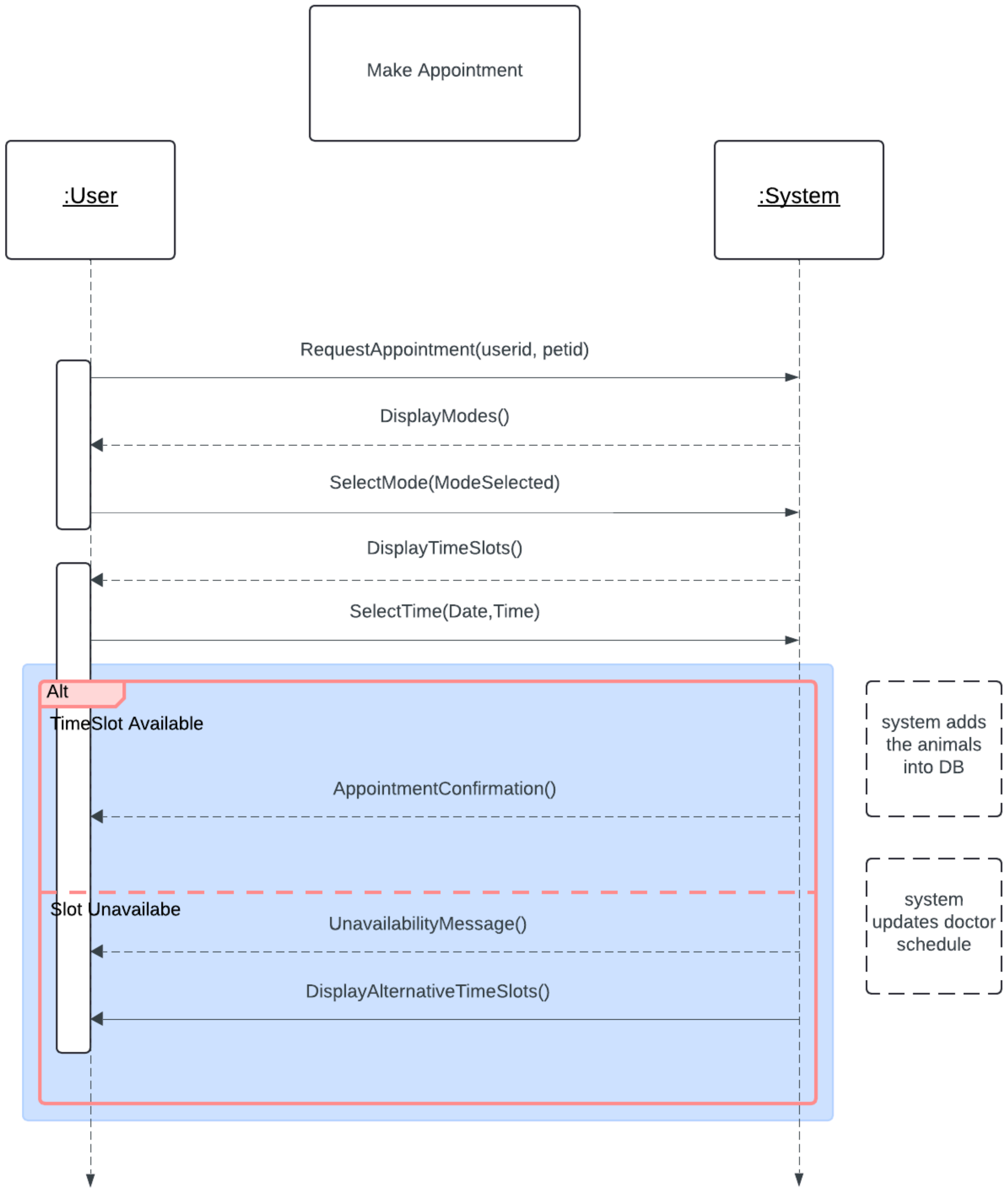


4. Domain Model

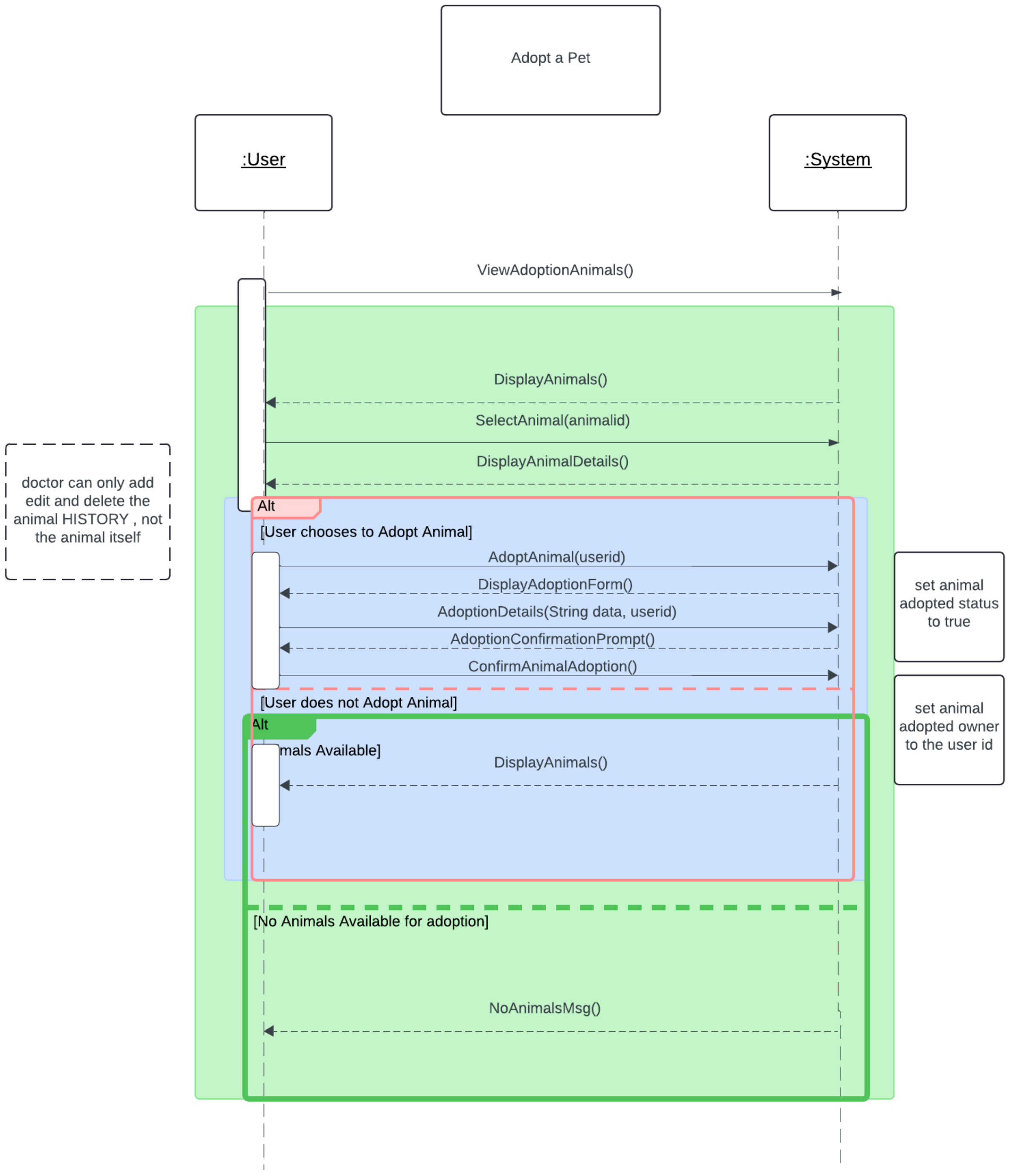


5. System Sequence Diagram

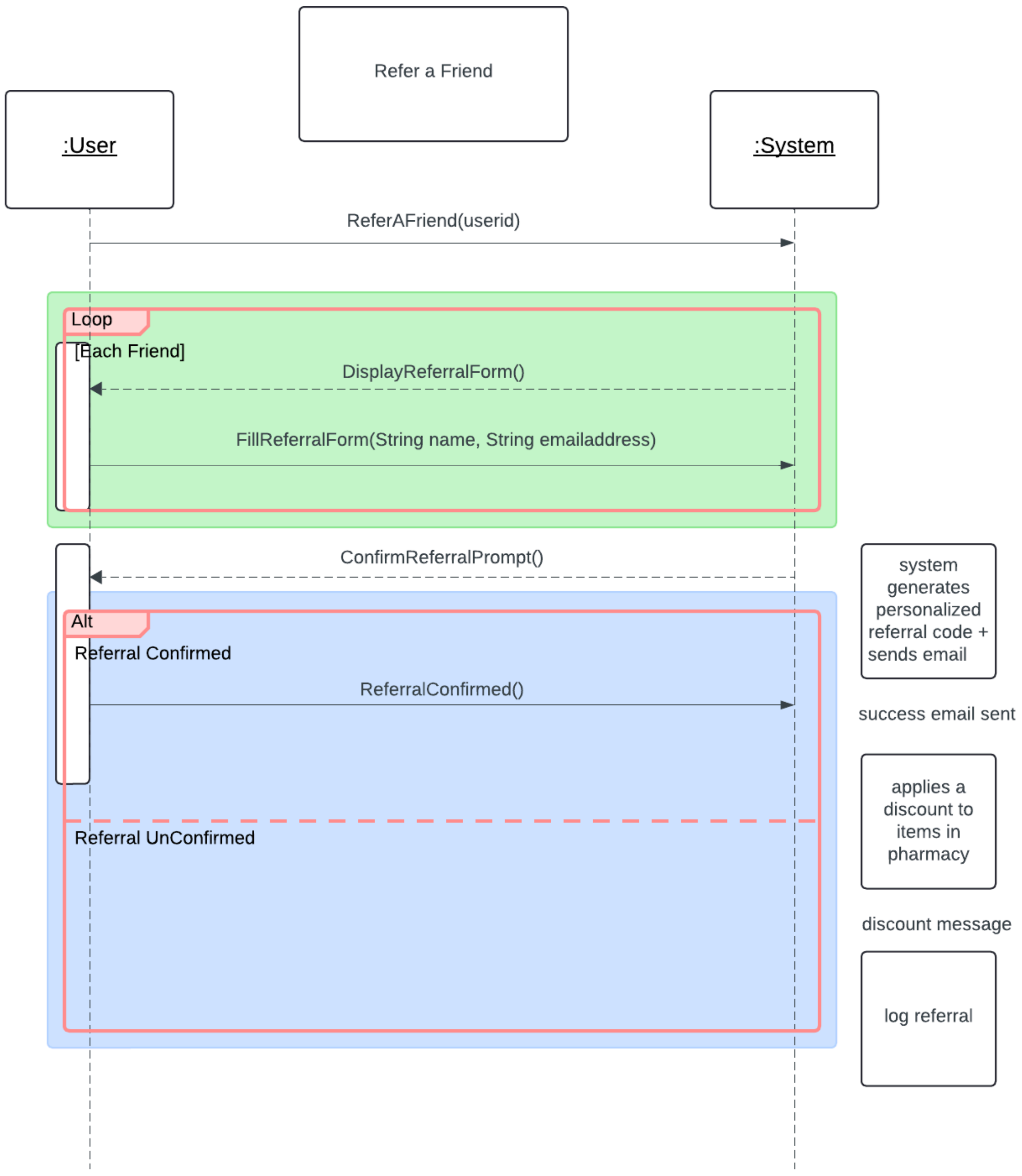
1. Make an appointment



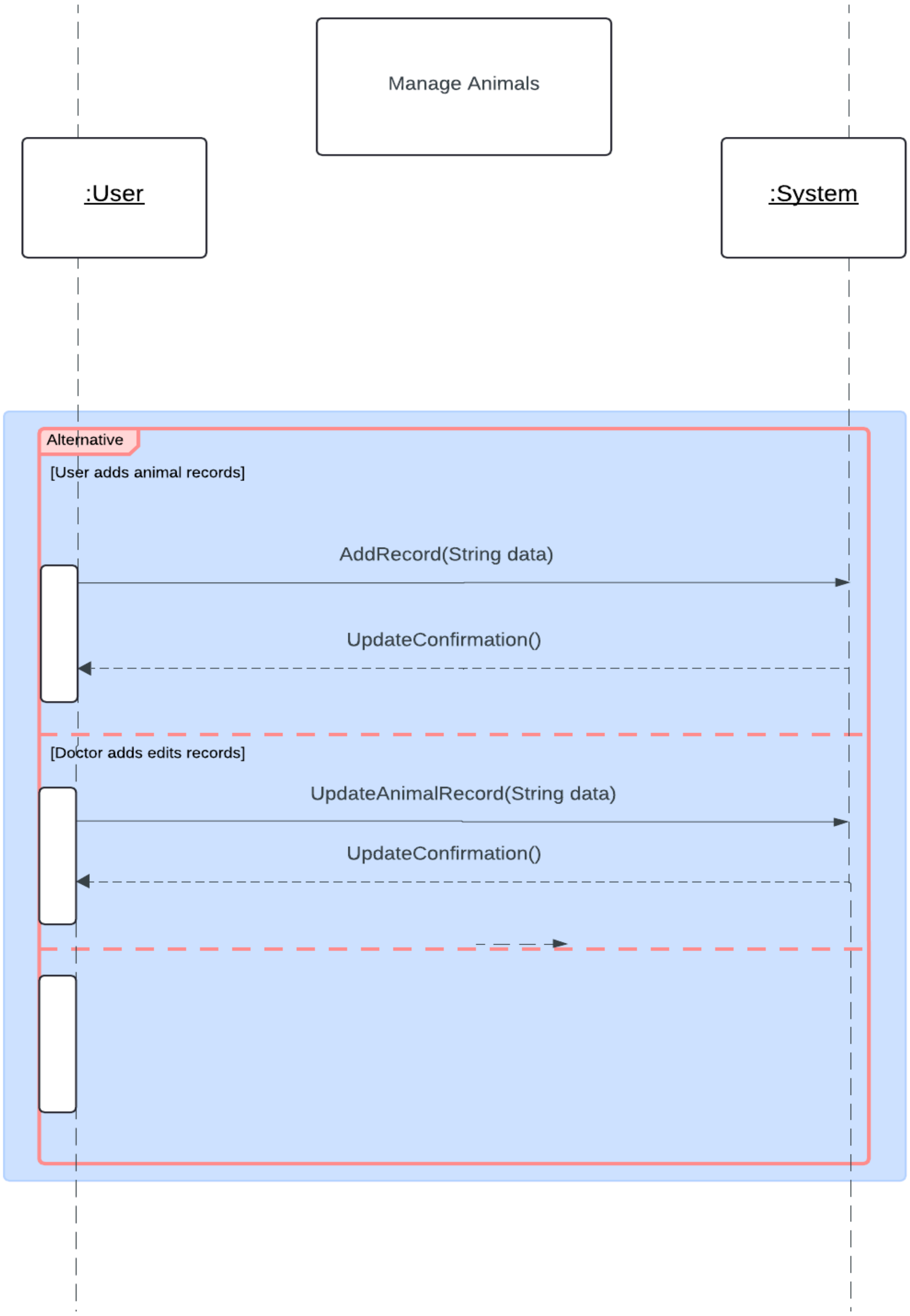
2. Adopt an animal


Adopt a Pet
:User
:System
ViewAdoptionAnimals()
DisplayAnimals()
SelectAnimal(animalid)
DisplayAnimalDetails()
doctor can only add edit and delete the animal HISTORY , not the animal itself
Alt
[User chooses to Adopt Animal]
AdoptAnimal(userid)
DisplayAdoptionForm()
AdoptionDetails(String data, userid)
AdoptionConfirmationPrompt()
ConfirmAnimalAdoption()
set animal adopted status to true
set animal adopted owner to the user id
[User does not Adopt Animal]
Alt
mals Available]
DisplayAnimals()
[No Animals Available for adoption]
NoAnimalsMsg()

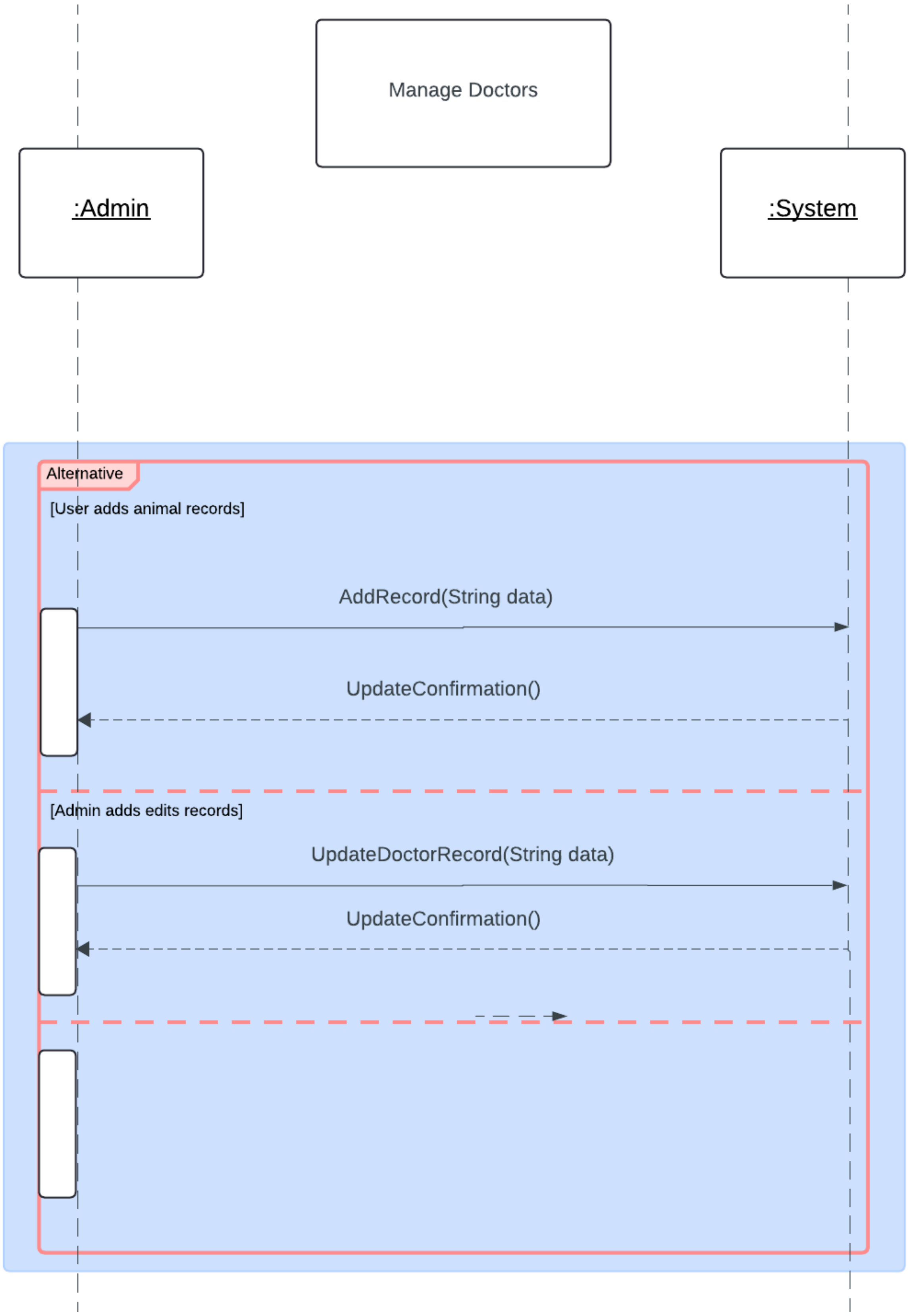
3. Refer a friend



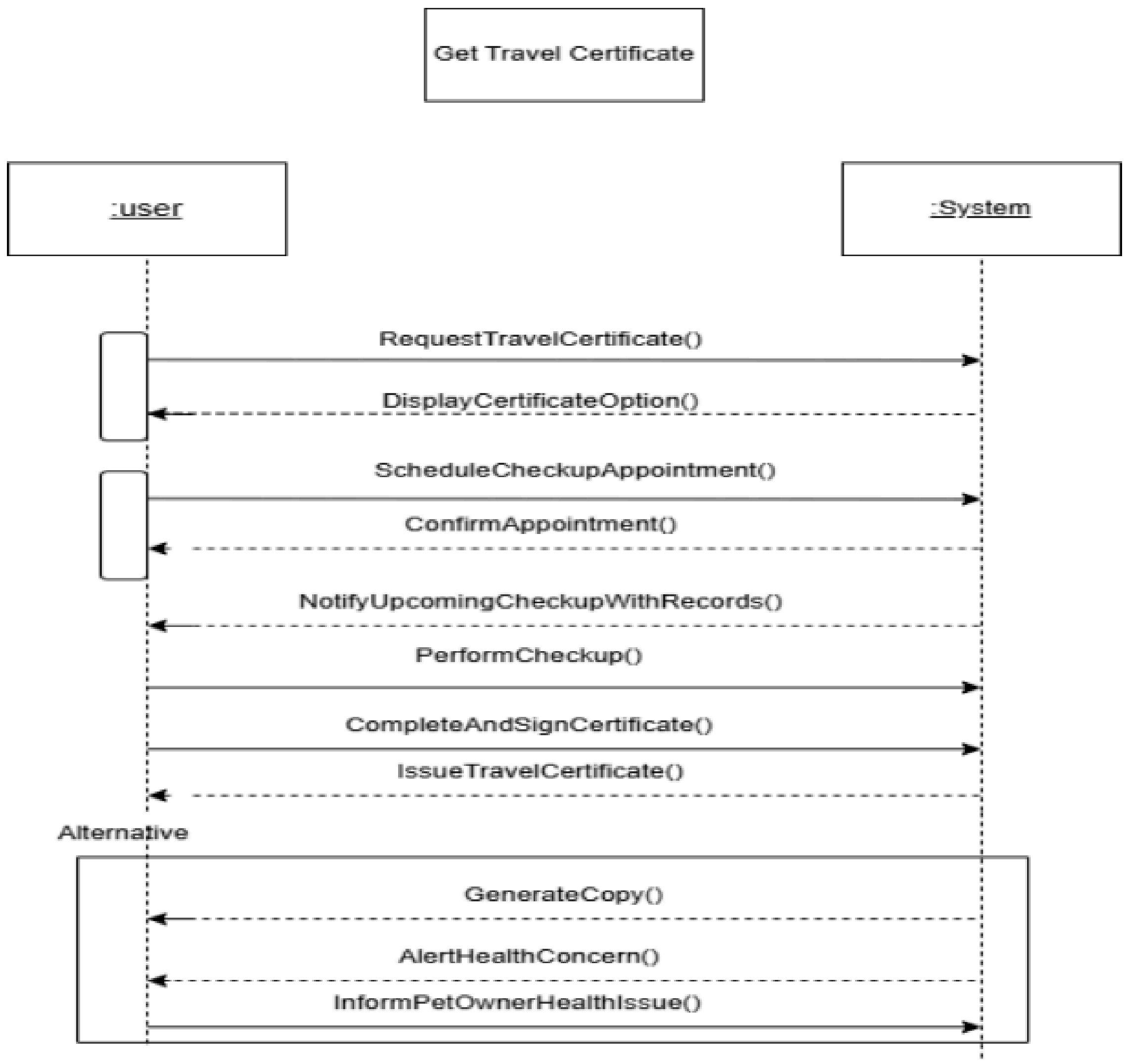
4. Manage Animals



5. Manage Doctor

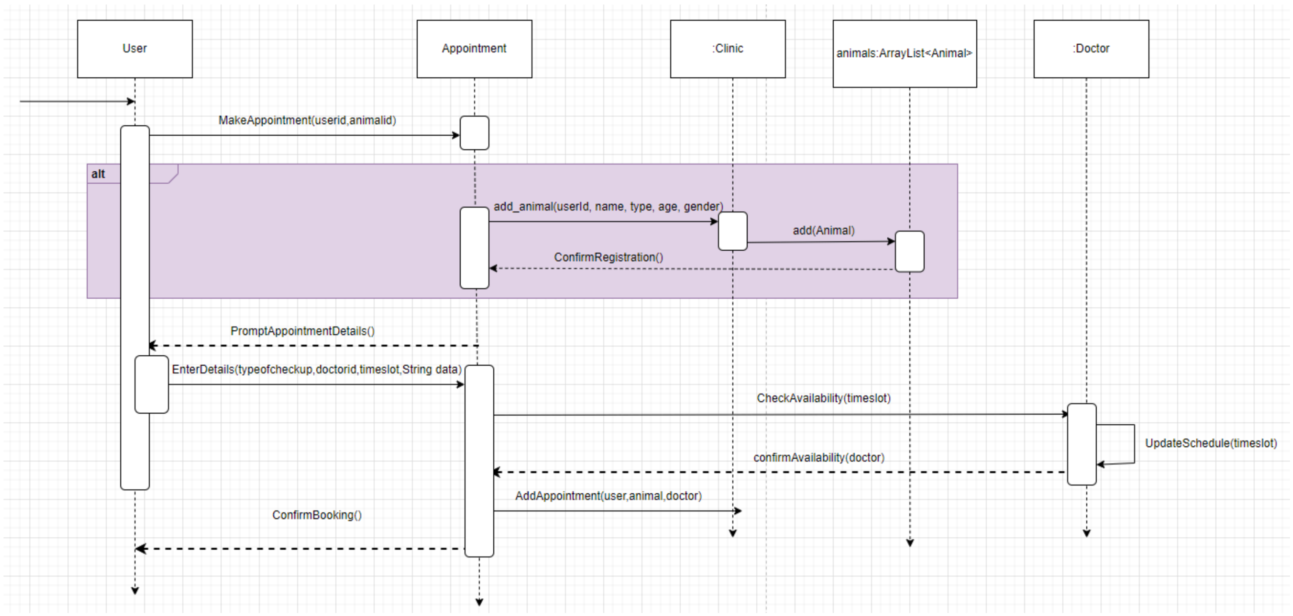


7. Travel Certificate

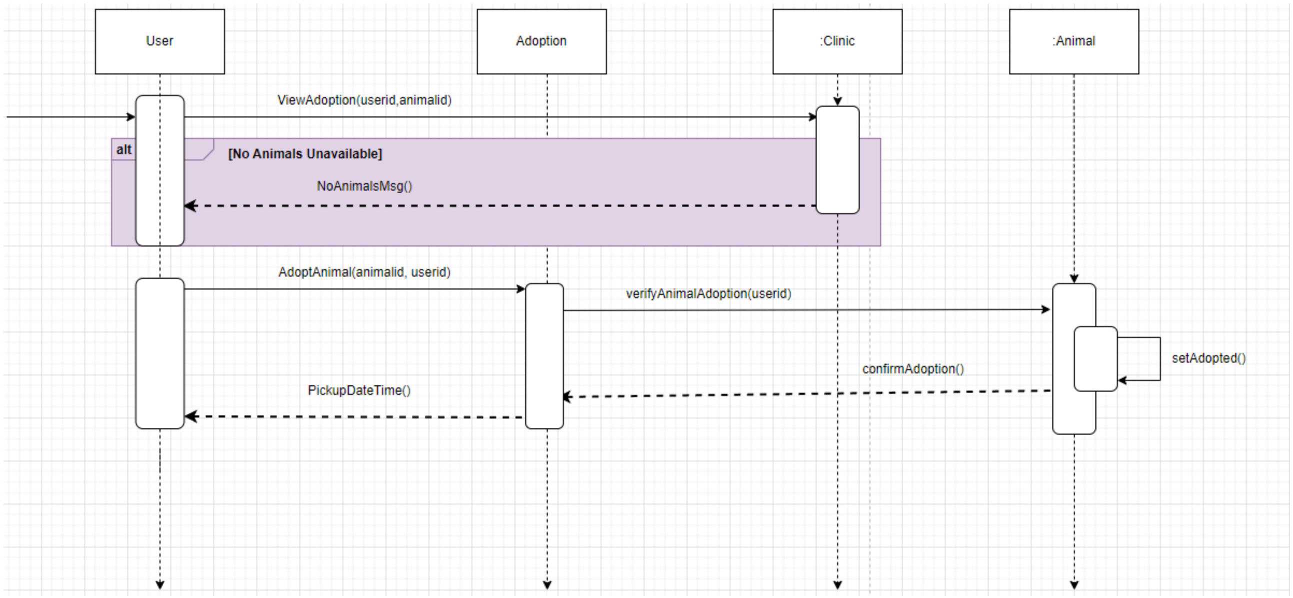

Get Travel Certificate
:user
:System
RequestTravelCertificate()
DisplayCertificateOption()
ScheduleCheckupAppointment()
ConfirmAppointment()
NotifyUpcomingCheckupWithRecords()
PerformCheckup()
CompleteAndSignCertificate()
IssueTravelCertificate()
Alternative
GenerateCopy()
AlertHealthConcern()
InformPetOwnerHealthIssue()

6. Sequence Diagram

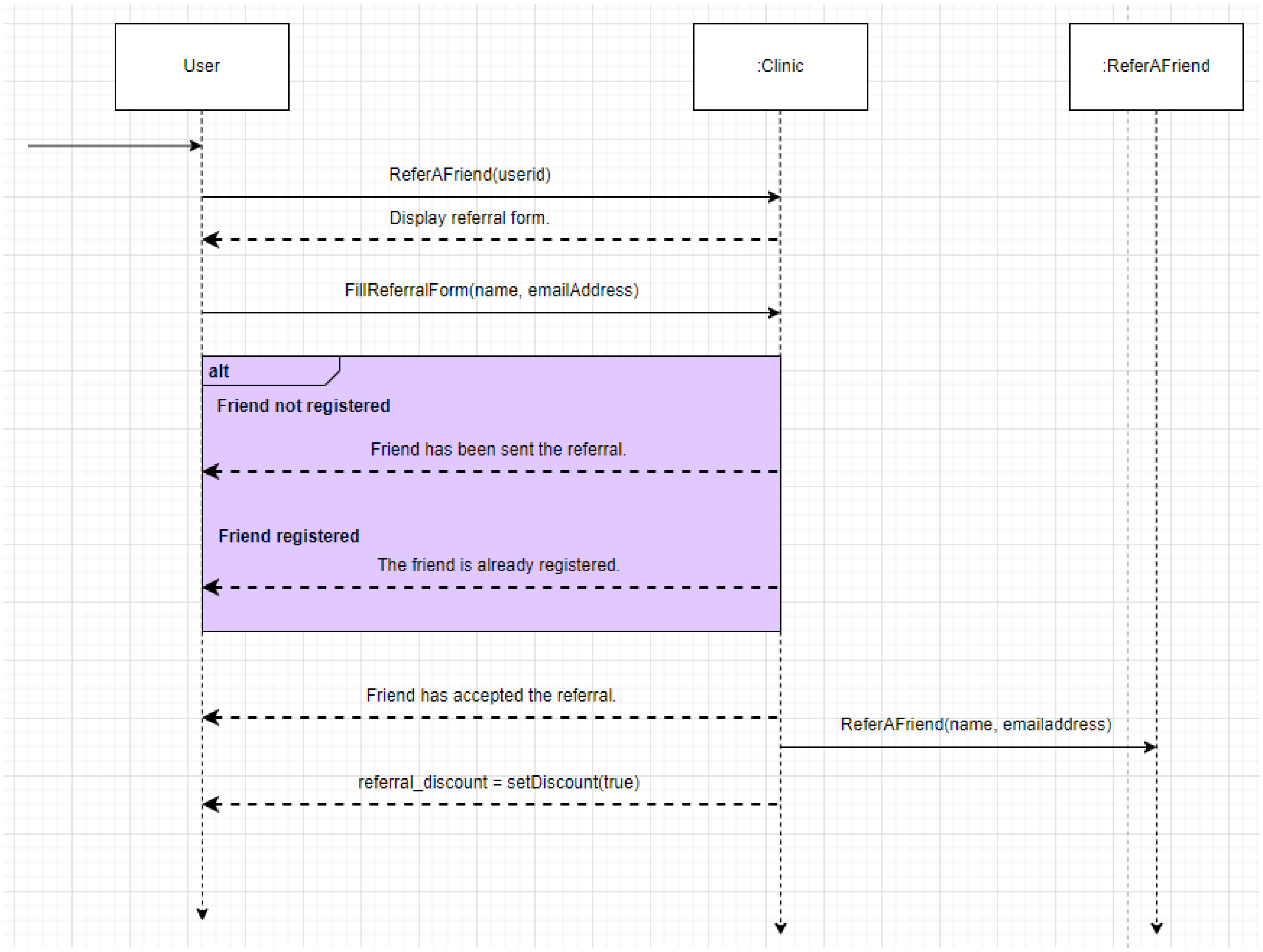
1. Book an Appointment


User
Appointment
:Clinic
animals:ArrayList<Animal>
:Doctor
MakeAppointment(userid,animalid)
alt
add_animal(userId, name, type, age, gender)
add(Animal)
ConfirmRegistration()
PromptAppointmentDetails()
EnterDetails(typeofcheckup,doctorid,timeslot,String data)
CheckAvailability(timeslot)
UpdateSchedule(timeslot)
confirmAvailability(doctor)
AddAppointment(user,animal,doctor)
ConfirmBooking()

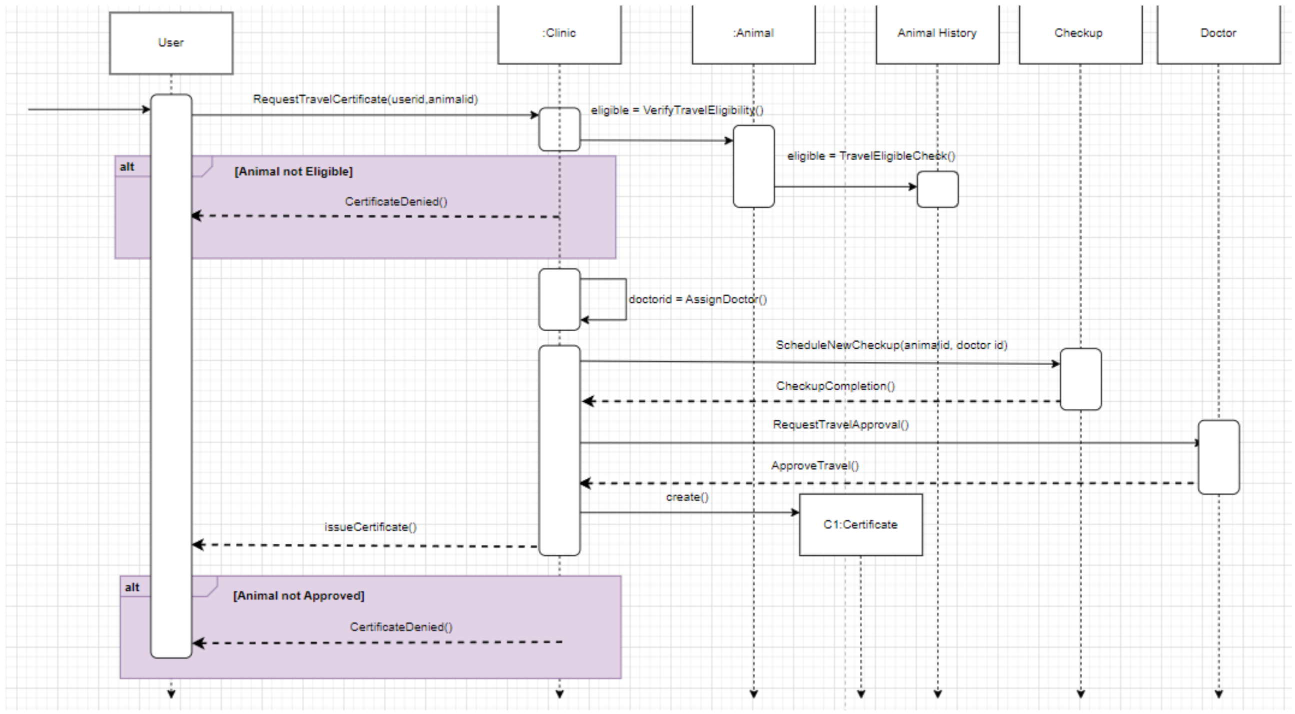
2. Adopt an Animal



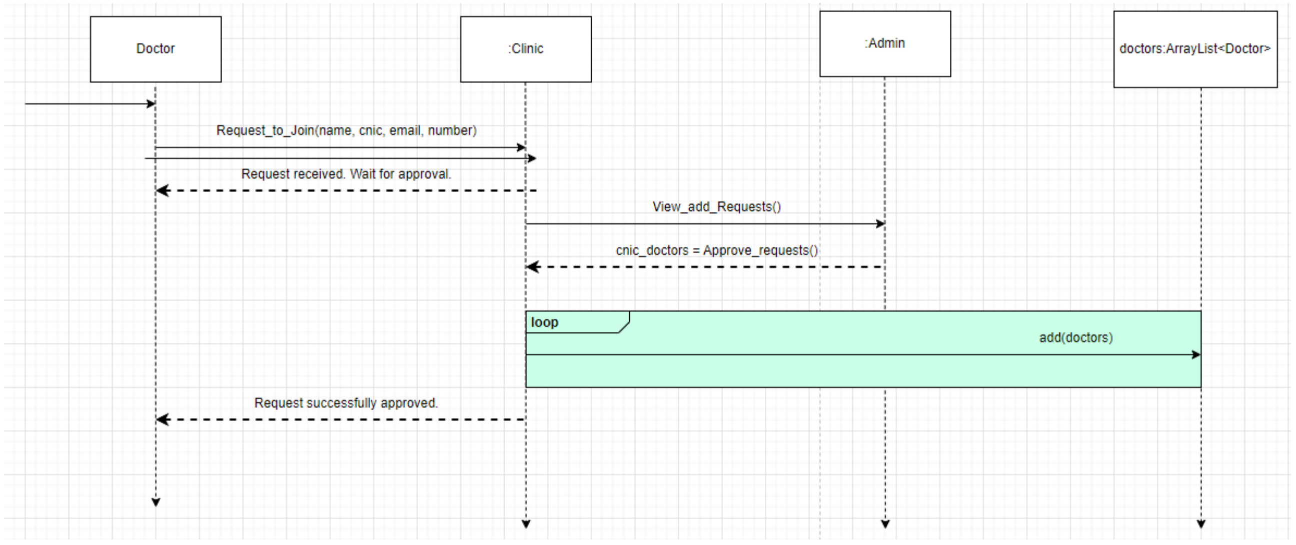
3. Refer a Friend


User
:Clinic
:ReferAFriend
ReferAFriend(userid)
Display referral form.
FillReferralForm(name, emailAddress)
alt
Friend not registered
Friend has been sent the referral.
Friend registered
The friend is already registered.
Friend has accepted the referral.
ReferAFriend(name, emailaddress)
referral_discount = setDiscount(true)

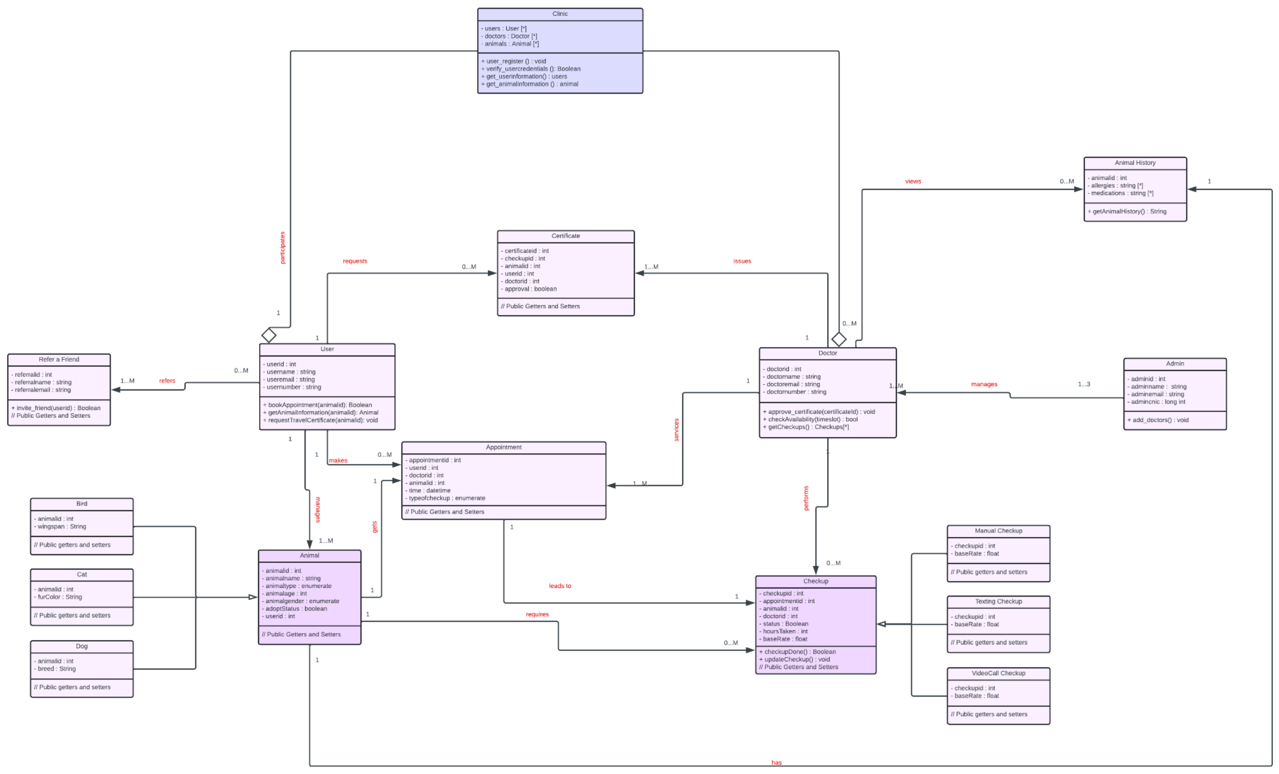
4. Travel Certificate



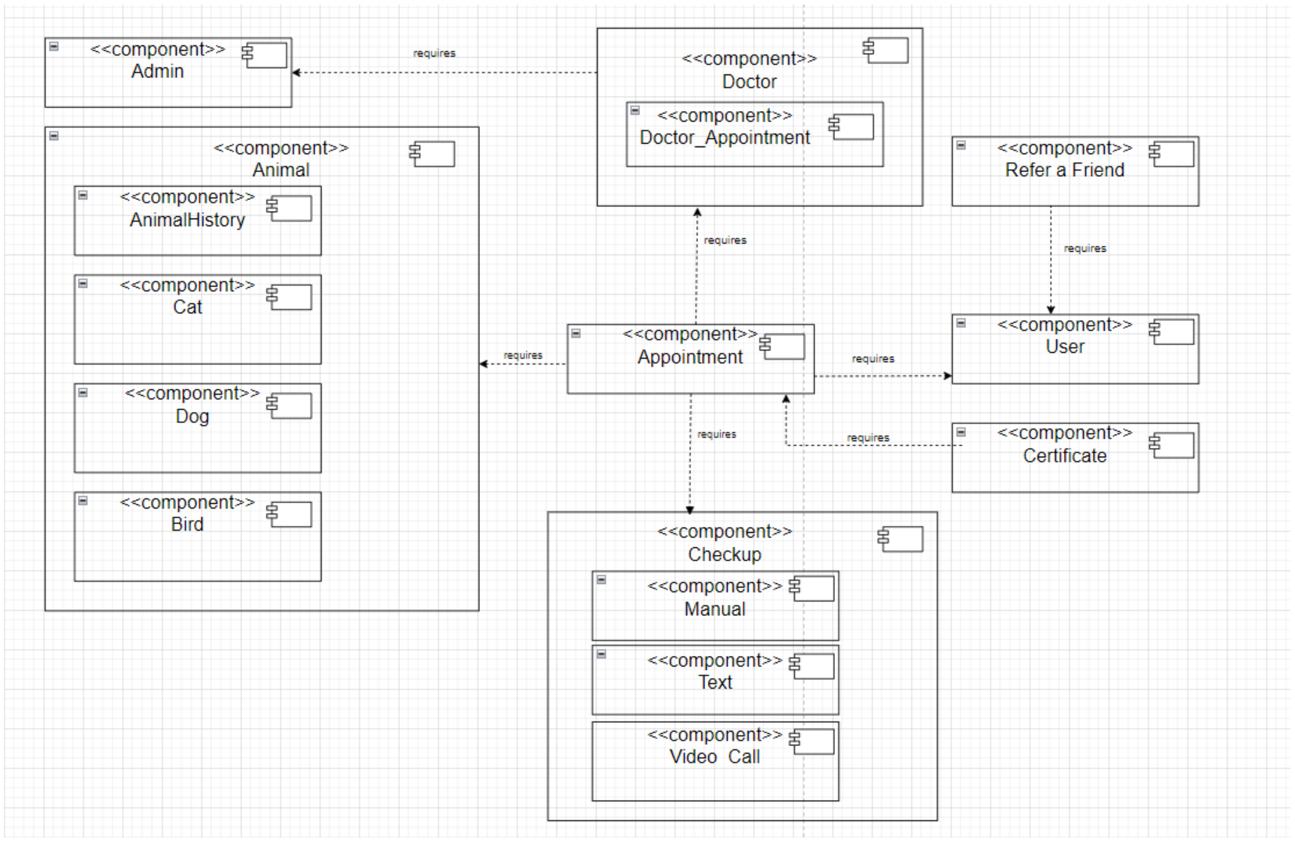
5. Add Doctors


Doctor
:Clinic
:Admin
doctors:ArrayList<Doctor>
Request_to_Join(name, cnic, email, number)
Request received. Wait for approval.
View_add_Requests()
cnic_doctors = Approve_requests()
loop
add(doctors)
Request successfully approved.

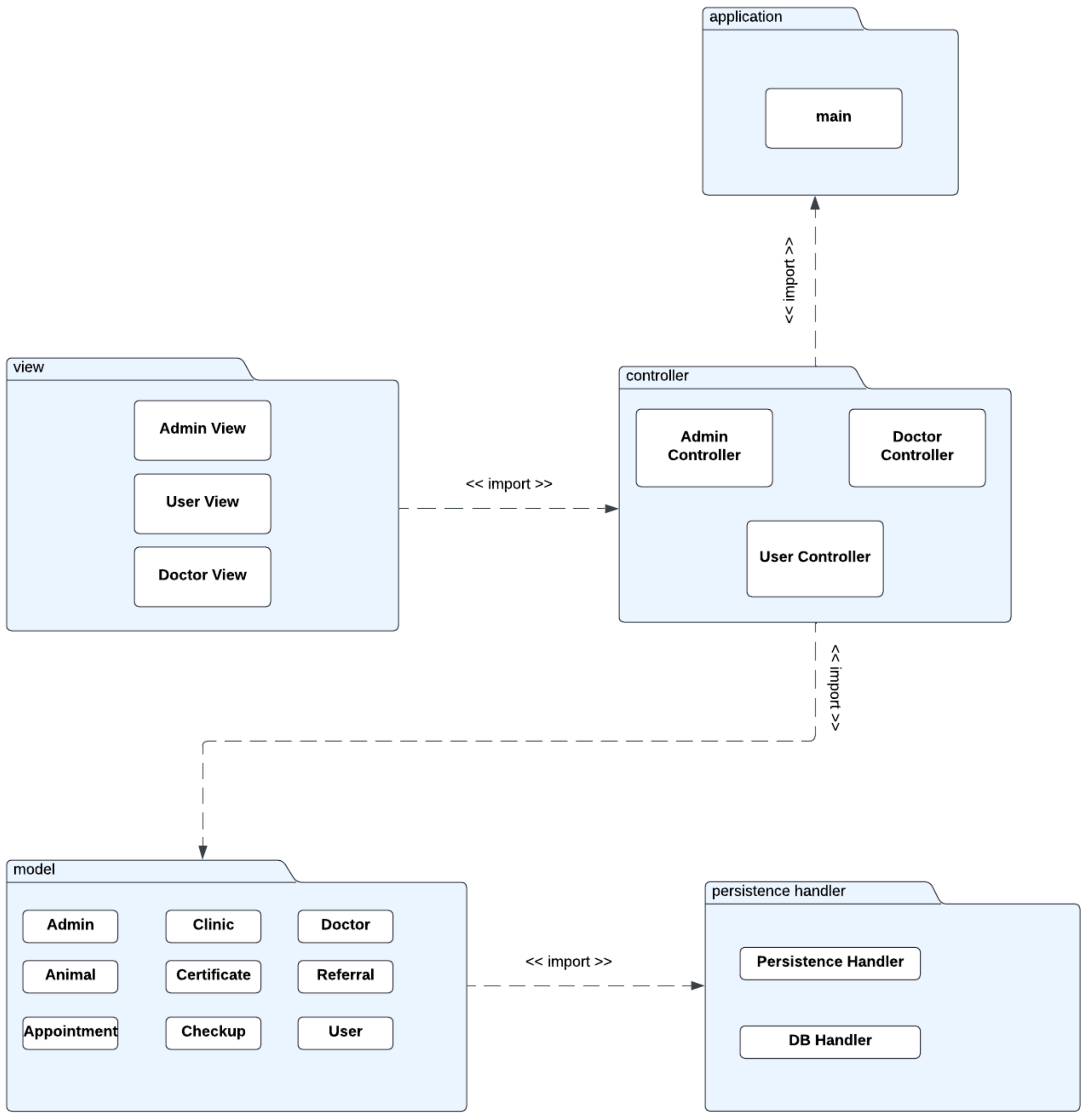
7. Class Diagram



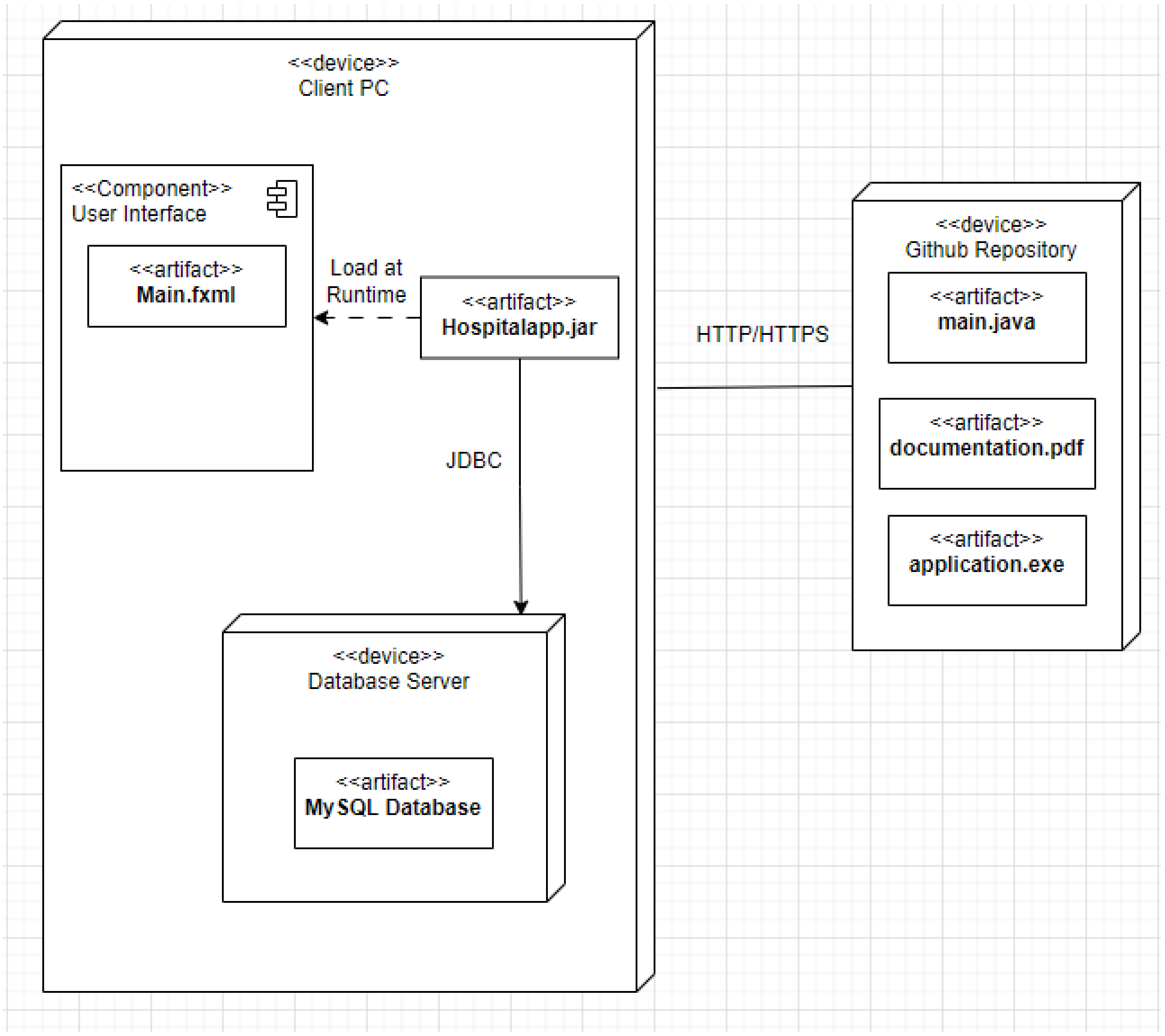
8. Component Diagram



9. Package Diagram



10. Deployment Diagram



LINKS FOR THE DIAGRAMS:

https://lucid.app/lucidchart/aad3a256-9972-48e4-a2d7-0fac9d19525c/edit?invitationId=inv_5bed9afa-8e1c-4f75-a6aa-0071278d5c06

https://drive.google.com/file/d/1E5j6QBydA95ZMGi5p-TfgJebSTpBZV1_/view?_usp=sharing