

웹기반 클라우드 기술을 이용한 반려견 헬스케어 서비스

# 설 계 서

2020.05.07

가천대학교 컴퓨터공학과

팀장 ■ 이다은(201632230)

팀원 ■ 전현우(201533970)

팀원 ■ 황승환(201433931)

## 문서 정보

본 문서는 웹기반 클라우드 기술을 이용한 반려견 헬스케어 서비스 개발을 위한 요구규격 및 설계서입니다.

버 전	1.2
작성일	2020-05-07
상 태	<input checked="" type="checkbox"/> 완료 <input type="checkbox"/> 진행 중 <input type="checkbox"/> 초안

버전	변경한 사람	변경한 날짜	버전업 변경(또는 추가)내용
0.1	이병문 교수님	2020-04-13	요구규격 및 설계서 양식(템플릿) 정의
0.2	이다은	2020-04-16	요구기능 정의 작성
0.3	전현우	2020-04-18	프로젝트 개요 작성
0.4	황승환	2020-04-20	스토리보드 설계 작성
0.4	전현우	2020-04-21	프로세스(기능) 설계 작성
0.5	황승환	2020-04-22	사용자UI 설계 작성
0.6	이다은	2020-04-24	REST API 설계 작성
0.7	전현우	2020-04-26	유스케이스 모델 및 환경구성 작성
0.8	이다은	2020-04-29	데이터베이스 설계 작성
0.9	황승환	2020-05-01	관리자UI 설계 작성
1.0	이다은	2020-05-03	설계서 작성 완료
1.1	황승환	2020-05-05	화면(UI) 설계 수정
1.2	이다은	2020-05-07	설계서 최종 작성 완료

## 목 차

<b>1. 프로젝트 개요</b>	
1.1 개발목표와 범위	4
1.2 개발일정/산출물	6
1.3 개발조직/역할	6
<b>2. 서비스기능 정의</b>	
2.1 서비스 개요	7
2.2 유스케이스 모델	10
2.3 요구기능 정의(사용자)	15
2.4 요구기능 정의(관리자)	20
<b>3. 프로세스(기능) 설계</b>	
3.1 시스템 아키텍처	23
3.2 사용자기능 설계(Sequence Diagram)	24
3.3 관리자기능 설계(Sequence Diagram)	49
<b>4. 화면(UI)설계</b>	
4.1 스토리보드 설계	72
4.2 초기화면(로그인 전화면)설계	74
4.3 사용자UI 설계	75
4.4 관리자UI 설계	121
<b>5. REST API 인터페이스 설계</b>	
5.1 서버/클라이언트 구조	148
5.2 REST API 정의	149
5.3 상세 REST API 설계	152
<b>6. 데이터베이스 설계</b>	
6.1 ERD	204
6.2 논리적설계(테이블명세서)	205
6.3 물리적설계(SQL스크립트)	208
<b>7. 환경구성</b>	
7.1 개발환경	213
7.2 운영환경	213
7.3 소스디렉터리 구조	214

## 1. 프로젝트 개요

### 1.1. 개발목표와 범위

#### ■ 개발목표

본 설계서는 PC 환경에서 반려견의 의료 서비스(질병 진단, 병원 추천, 병원 예약)를 위한 반려견 헬스케어 웹 서비스 개발을 위한 설계서이다. 주 사용자는 반려견을 키우는 보호자들이며, 도메인을 통해 접속하고 사용하기 편리하도록 고안되는 것을 전제로 한다. 헬스케어 웹 서비스를 효과적으로 관리하기 위해 관리자 기능도 구성하였으며, 최대한 편리성을 추구할 수 있도록 개발한다. 관리자는 웹기반 시각화된 UI를 통해 사용자 관리, 질병 관리, 병원 관리를 할 수 있도록 한다.

아래 그림 1.1은 본 서비스의 전체적인 구성을 나타내며, 반려견의 헬스케어를 희망하는 사용자가 본 서비스를 이용한다.

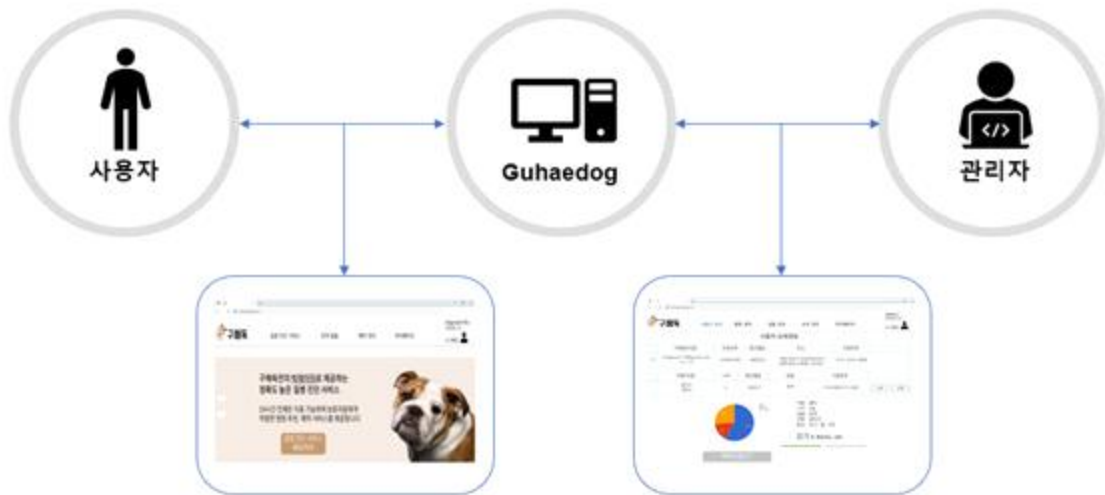


그림 1.1 구해독 웹서비스의 설계범위

## ■ 개발필요성

지금 대한민국은 반려동물 르네상스 시대라 해도 과언이 아니다. 실제 반려동물 양육인구는 1000만 명에 이르는 것으로 추정된다. 국민 5명당 1명이 반려동물을 키우고 있다는 뜻이다. 특히, 서울시 반려동물 보유 현황에 따르면 반려동물의 약 85%는 반려견이 차지한다. 이처럼 반려견을 키우는 인구가 늘어나면서 반려견의 헬스케어에 대한 관심도 자연스레 증가하고 있다. 하지만 반려견의 헬스케어에 부담을 가지는 보호자들도 많다. 무엇보다 평균 33.2만 원의 질병 치료 비용에 큰 부담감을 느낀다. 단순히 질병 진단을 받기 위해서 동물병원에 방문하는 경우에도 보호자가 지불해야 하는 비용은 만만치 않다.

위와 같은 문제점을 해결하기 위해 다양한 헬스케어 웹/앱 서비스가 출시되었다. 그러나, 기존의 서비스는 의사와의 실시간 소통을 통해 반려견의 질병을 진단해 주기에 24시간 이용할 수 없어 매우 제한적이다. 만약 서비스 이용 불가능 시간에 보호자의 반려견에게 위급 상황이 발생하는 경우, 보호자는 반려견의 질병 진단과 올바른 대처를 하기 어렵게 된다.

앞에서 언급한 문제점과 기존 서비스의 단점을 보완하고자 24시간 이용이 가능한 반려견 헬스케어 웹 서비스 ‘구해독’을 기획하였다. 사용자는 구해독의 반려견 질병 진단 서비스를 통해 반려견의 질병 진단에 들어가는 비용을 줄일 수 있다. 더 나아가 사용자는 사용자 위치 기반 동물병원 조회 및 예약 서비스로 쉽게 동물병원 예약이 가능하다. 한편, 관리자는 질병 관리 서비스로 등록된 질병 정보와 사용자들이 진단한 결과를 시각화된 자료로 확인할 수 있다. 더불어, 관리자는 병원 관리 서비스로 사용자들의 동물병원 예약 현황을 쉽게 조회하고 수정할 수 있다. 이러한 방법으로 더 많은 클라이언트가 구해독을 이용하게 만든다.

본 서비스는 앞의 과정을 통해 헬스케어 웹 서비스를 제공하여 클라이언트의 편리성을 증대시키고, 일부 사용자들에게는 질병 진단 비용 절약의 기회도 제공한다. 이 서비스 구조는 기존 서비스의 한계를 극복하며 사용자, 관리자 모두가 만족할 수 있는 결과를 만들어 낸다. 따라서 이와 같은 반려견 헬스케어 서비스가 필요하게 된다.

## 1.2. 개발일정/산출물

본 설계서는 아래 그림 1.2와 같이 개발일정에 따라 진행하며, 각 단계별로 산출물을 개발한다.

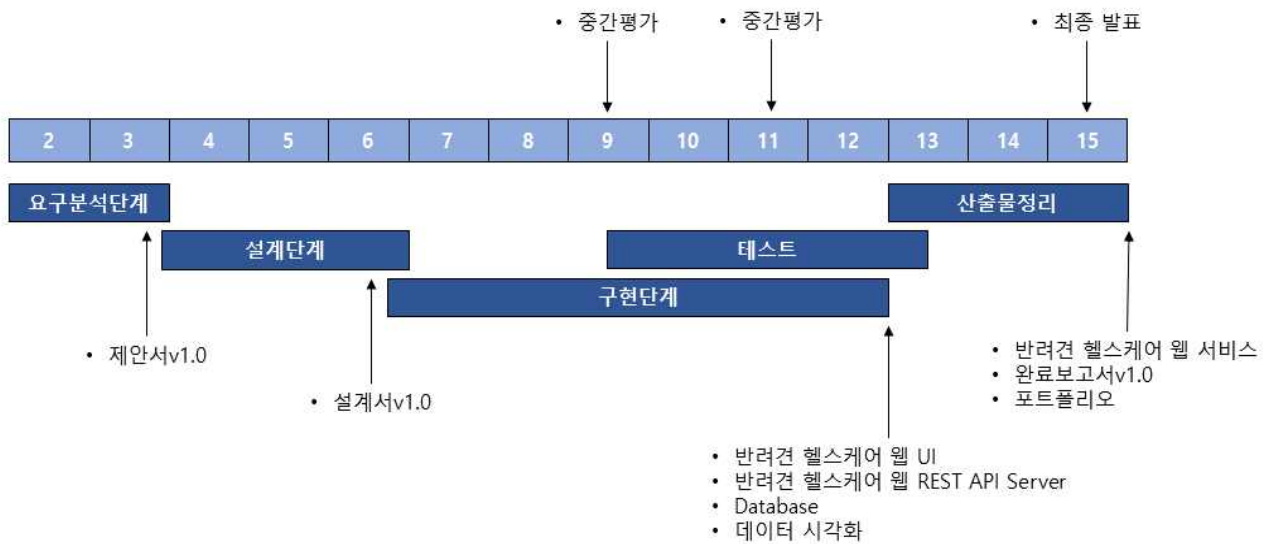


그림 1.2 개발일정 및 산출물

## 1.3. 개발조직/역할

역할	이름	역할(개발기능)	비고
팀장	이다은	질병 조회 기능, 병원 예약 기능, 병원 관리 기능, 시각화 기능	
팀원	전현우	사용자 관리 기능, 반려견 관리 기능, 질병 관리 기능, 병원 추천 기능	
팀원	황승환	사용자 인증 기능, 관리자 인증 기능, 반려견 등록 기능	

## 2. 서비스기능 정의

### 2.1. 서비스 개요

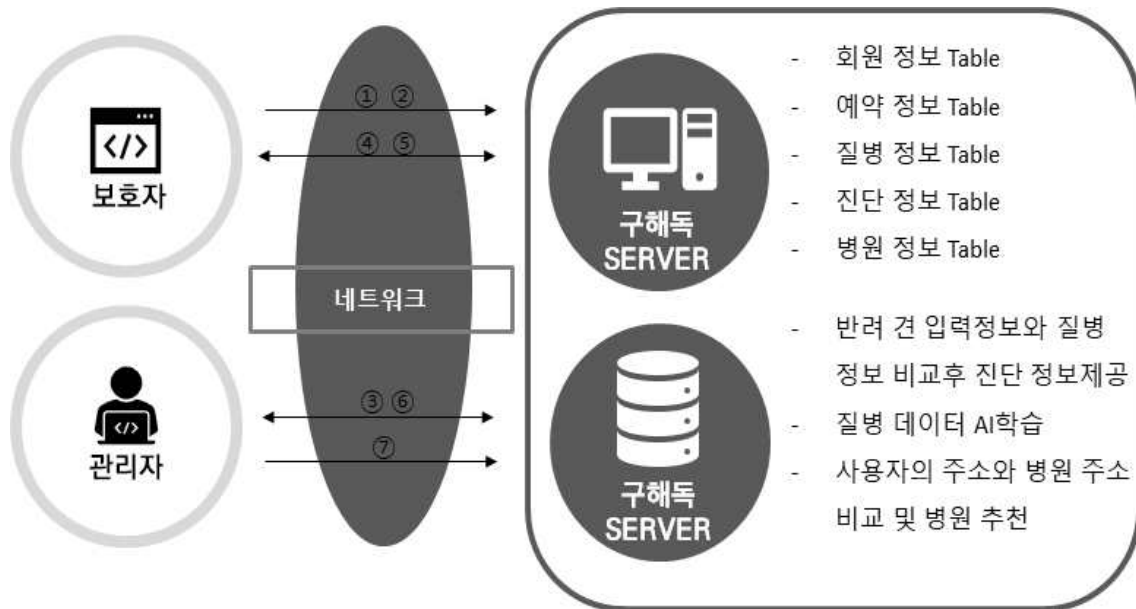


그림 2.1.1 구해독 서비스 시나리오 구성도

- ① 사용자는 웹을 통해 서버에 정해진 사용자 가입 양식(이름/이메일/주소/비밀번호/비밀번호 확인)에 따라 사용자 가입을 요청한다.
- ② 사용자는 웹을 통해 서버에 정해진 입력 양식(나이/몸무게/진단 차트)에 따라 진단 정보를 요청하고 서버(AI)에서 추론된 반려견의 진단 정보를 시각화 자료와 함께 요청할 수 있다.
- ③ 관리자는 웹을 통해 사용자 가입이 된 사용자들의 정보를 관리(사용자 정보 조회, 사용자 정지)할 수 있고, 사용자의 반려견 정보를 조회, 수정, 삭제할 수 있다.
- ④ 사용자는 웹을 통해 지도 API로 데이터베이스에 등록된 병원 정보를 추천받아 확인하고 예약 등록, 조회, 수정, 취소할 수 있다.
- ⑤ 관리자는 웹을 통해 서버에 등록된 사용자의 예약 정보를 조회, 등록, 삭제할 수 있다.
- ⑥ 관리자는 웹을 통해 데이터베이스에 등록된 반려견의 질병 정보와 반려견의 질병 진단 정보에 대한 시각화 자료를 요청할 수 있다.

## 2.2. 유스케이스 모델

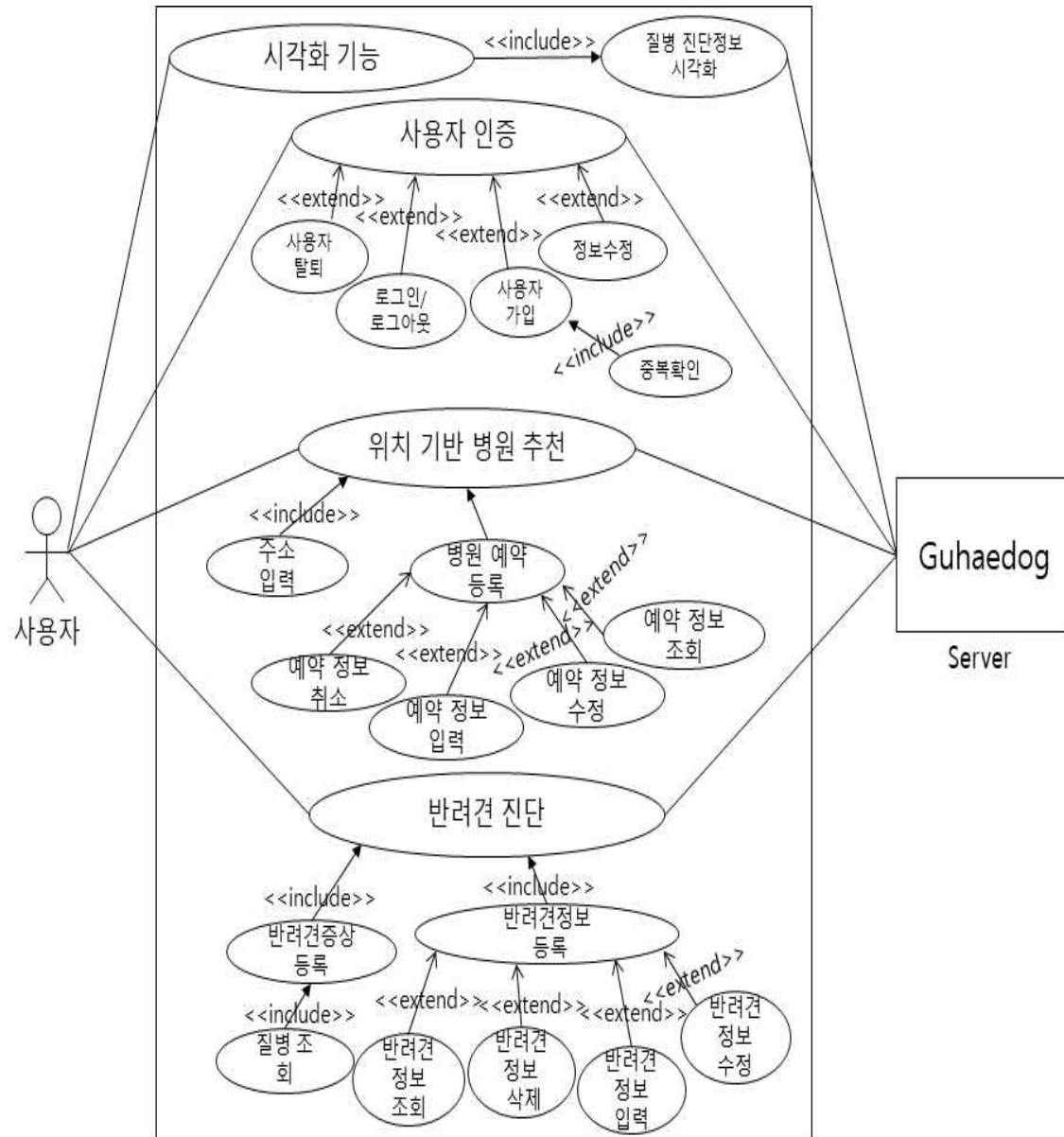


그림 2.2.1 사용자용 유스케이스 다이어그램

사용자는 내 정보 관리를 위해 사용자 가입, 로그인, 로그아웃, 정보수정, 사용자 탈퇴 관련 페이지에 접근이 가능하다. 반려견의 질병 진단을 위해 반려견의 정보와 증상 상태를 등록하는 페이지에 접근하여 질병 진단 결과를 조회할 수 있고, 반려견 정보를 조회, 수정, 삭제할 수 있다. 질병 진단 결과 조회 후, 사용자 위치 기반 병원 추천 페이지에 접근하여 동물병원 예약 조회, 등록, 수정, 삭제를 할 수 있다.



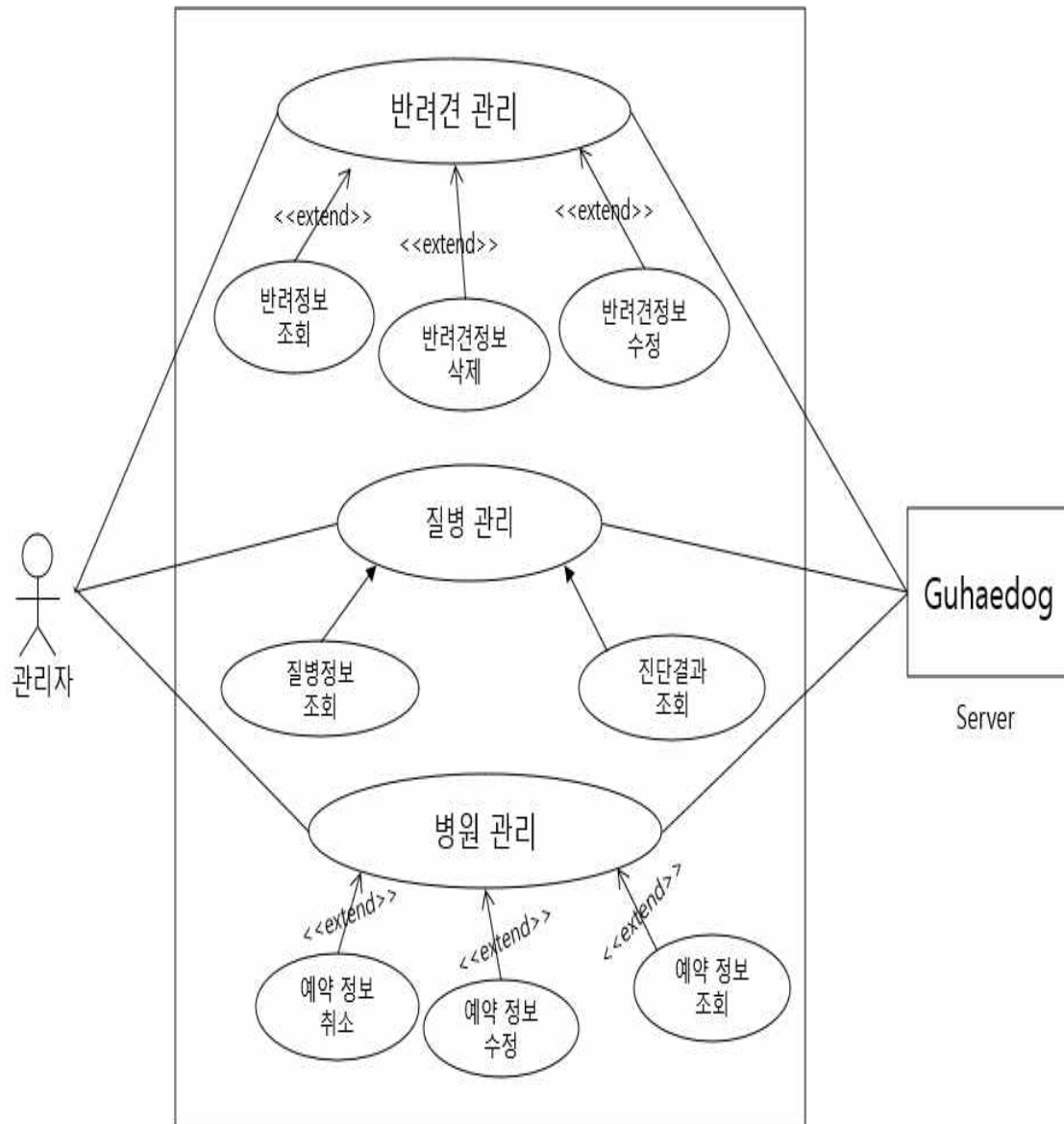


그림 2.2.4 관리자용 유스케이스 다이어그램 - 1

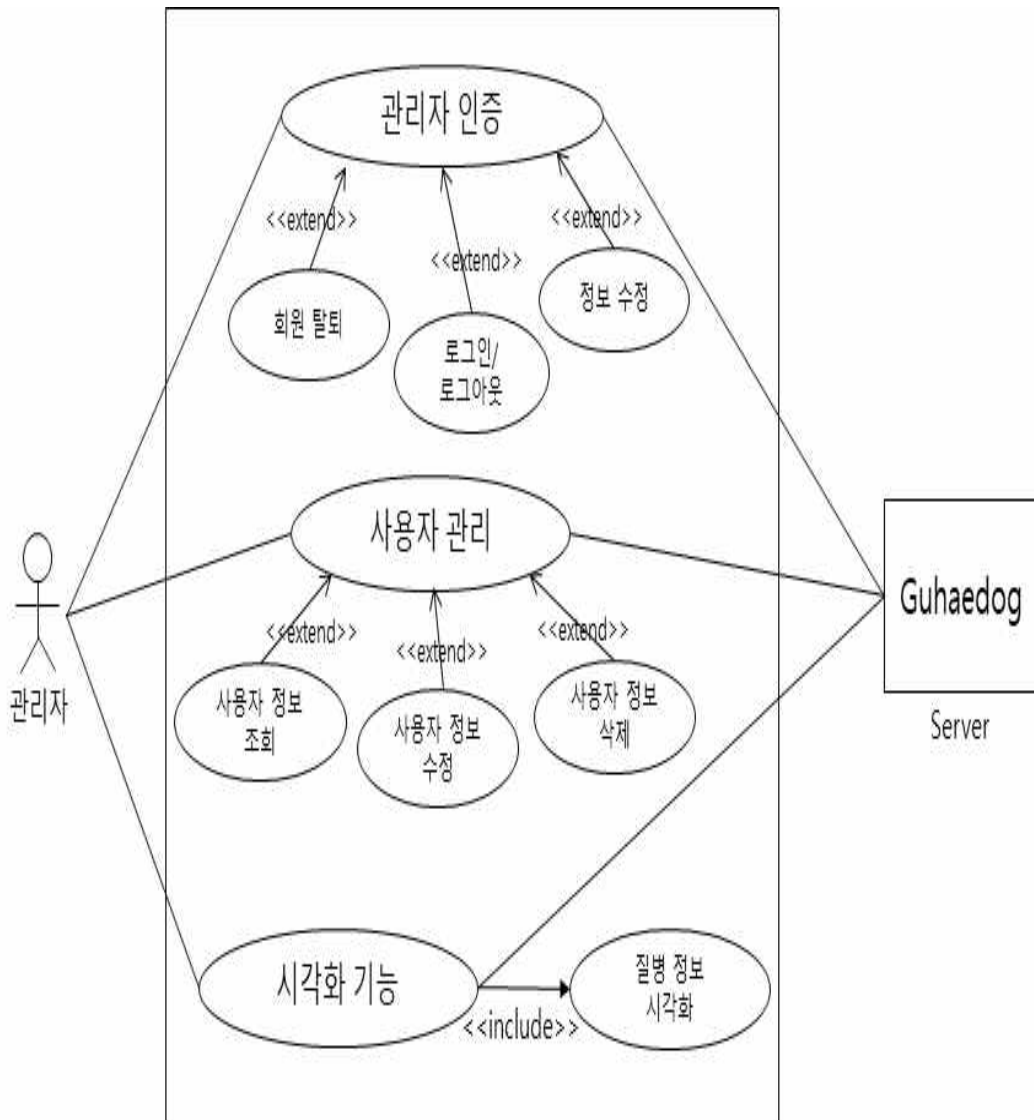


그림 2.2.5 관리자용 유스케이스 다이어그램 - 2

관리자는 내 정보 관리를 위해 로그인, 로그아웃, 정보수정, 관리자 탈퇴 페이지에 접근이 가능하다. 사용자 관리 기능으로 사용자 정보를 조회, 수정, 삭제할 수 있고, 질병 관리 기능으로 등록된 질병 정보를 조회하고 반려견의 질병 진단 결과를 조회할 수 있다. 사용자의 동물병원 예약 관리를 위해 예약 정보를 조회, 수정, 삭제할 수 있다. 시각화 기능을 이용하여 질병 진단 정보를 시각화 자료로 확인할 수 있다.

### 2.3. 요구기능 정의 (사용자)

번호	주요기능명	상세기능명	개발담당
1)	사용자 인증	사용자 가입	황승환
2)		로그인	황승환
3)		로그아웃	황승환
4)		정보수정	황승환
5)		사용자 탈퇴	황승환
6)	반려견 등록	반려견 정보 등록	황승환
7)		반려견 정보 조회	황승환
8)		반려견 정보 수정	전현우
9)		반려견 정보 삭제	전현우
10)	질병 조회	질병 증상 등록	이다은
11)		진단 결과 조회	이다은
12)	병원 추천	위치 기반 추천	전현우
13)	병원 예약	예약 등록	이다은
14)		예약 조회	이다은
15)		예약 수정	이다은
16)		예약 취소	이다은
17)	시각화 기능	질병 진단 결과 시각화	이다은

#### 1) 사용자 가입

- 본 서비스를 이용하기 위해 사용자 정보를 등록한다.
- 웹 페이지에서 아이디, 비밀번호, 이름, 주소 등의 개인정보를 입력받는다.
- 사용자로부터 받은 정보는 데이터베이스에 저장된다.

#### 2) 로그인

- 사용자로부터 아이디, 비밀번호를 로그인 폼을 통해 받아 데이터베이스에 저장된 값과 비교한다.
- 비교 결과에 따라 로그인 성공 혹은 실패 화면을 보여준다.
- 사용자는 로그인으로 사용자 인증을 마친 후에 서비스 기능을 사용할 수 있다.
- 사용자의 계정이 정지되어 있으면 로그인을 할 수 없다.

#### 3) 로그아웃

- 사용자의 로그인 정보를 세션에서 종료시킨다.
- 사용자가 로그아웃했다면 다시 로그인을 해야 서비스 기능을 이용할 수 있다.

#### 4) 정보수정

- 사용자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 사용자는 이메일, 이름, 종류를 제외한 정보를 수정할 수 있다.

#### 5) 사용자 탈퇴

- 사용자는 자신의 계정을 삭제할 수 있다.
- 사용자가 탈퇴했다면 해당 사용자의 데이터는 데이터베이스에서 삭제된다.

#### 6) 반려견 정보 등록

- 사용자는 본인의 반려견에 대한 정보를 등록하여 주요 서비스를 이용할 수 있다.
- 웹 페이지에서 반려견의 이름, 나이, 종류 등의 정보를 받는다.
- 사용자로부터 받은 정보는 데이터베이스에 저장된다.

#### 7) 반려견 정보 조회

- 사용자는 본인이 등록한 반려견의 정보를 데이터베이스에서 조회한다.

#### 8) 반려견 정보 수정

- 사용자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 사용자는 반려견의 식별번호와 성별을 제외한 정보를 수정할 수 있다.

#### 9) 반려견 정보 삭제

- 사용자는 반려견의 정보를 삭제할 수 있다.
- 반려견 정보가 삭제됐다면 해당 반려견의 데이터는 데이터베이스에서 삭제된다.

#### 10) 질병 증상 등록

- 사용자는 본인의 반려견의 질병 증상을 등록하여 질병 진단 서비스를 이용할 수 있다.
- 웹 페이지에서 문진표를 통해 질병 증상 정보를 받는다.
- 사용자로부터 받은 정보는 데이터베이스에 저장된다.

#### 11) 진단 결과 조회

- 사용자는 본인의 반려견의 질병 진단 결과를 데이터베이스에서 조회한다.

#### 12) 위치 기반 추천

- 사용자는 질병 진단 결과와 함께 동물병원 추천 정보도 제공받는다.
- 사용자의 주소 정보를 토대로 근처 동물병원을 추천해준다.

#### 13) 예약 등록

- 사용자는 본인의 정보와 반려견의 정보를 등록하여 동물병원에 예약할 수 있다.
- 웹 페이지에서 병원예약 폼을 통해 사용자 및 반려견의 정보를 입력받는다.
- 사용자로부터 받은 정보는 데이터베이스에 저장된다.

#### 14) 예약 조회

- 사용자는 동물병원 예약 정보를 데이터베이스에서 조회한다.

15) 예약 수정

- 사용자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 사용자는 예약 날짜, 예약 시간 등을 수정할 수 있다.

16) 예약 취소

- 사용자는 동물병원 예약을 취소할 수 있다.
- 병원예약이 취소됐다면 해당 병원예약 데이터는 데이터베이스에서 삭제된다.

17) 질병 진단 결과 시각화

- 반려견의 질병 진단 결과를 막대 그래프와 같은 시각자료로 시각화한다.
- 사용자는 반려견의 질병 진단 결과를 시각자료로 한눈에 조회한다.

## 2.4. 요구기능 정의 (관리자)

번호	주요기능명	상세기능명	비고
1)	관리자 인증	로그인	황승환
2)		로그아웃	황승환
3)		정보수정	황승환
4)		관리자 탈퇴	황승환
5)	사용자 관리	사용자 정보 조회	전현우
6)		사용자 정보 수정	전현우
7)		사용자 정보 삭제	전현우
8)	반려견 관리	반려견 정보 조회	전현우
9)		반려견 정보 수정	전현우
10)		반려견 정보 삭제	전현우
11)	질병 관리	질병 정보 조회	전현우
12)		진단 결과 조회	전현우
13)	병원 관리	예약 조회	이다운
14)		예약 수정	이다운
15)		예약 취소	이다운
16)	시각화 기능	질병 진단 정보 시각화	이다운

### 1) 로그인

- 관리자로부터 아이디, 비밀번호를 로그인 폼을 통해 받아 데이터베이스에 저장된 값과 비교한다.
- 비교 결과에 따라 로그인 성공 혹은 실패 화면을 보여준다.
- 관리자는 로그인으로 관리자 인증을 마친 후에 서비스 기능을 사용할 수 있다.
- 관리자의 계정이 정지되어 있으면 로그인을 할 수 없다.

### 2) 로그아웃

- 관리자의 로그인 정보를 세션에서 종료시킨다.
- 관리자가 로그아웃했다면 다시 로그인을 해야 서비스 기능을 이용할 수 있다.

### 3) 정보수정

- 관리자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 관리자는 이메일, 이름, 종류를 제외한 정보를 수정할 수 있다.

### 4) 관리자 탈퇴

- 관리자는 자신의 계정을 삭제할 수 있다.
- 관리자가 탈퇴했다면 해당 관리자의 데이터는 데이터베이스에서 삭제된다.

5) 사용자 정보 조회

- 관리자는 사용자의 모든 정보를 데이터베이스에서 조회한다.
- 관리자 페이지에 사용자의 정보가 테이블 형태로 출력된다.

6) 사용자 정보 수정

- 관리자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 관리자는 사용자의 주소, 전화번호 등을 수정할 수 있다.

7) 사용자 정보 삭제

- 관리자는 사용자의 정보를 삭제할 수 있다.
- 사용자 정보가 삭제됐다면 해당 사용자의 데이터는 데이터베이스에서 삭제된다.

8) 반려견 정보 조회

- 관리자는 반려견의 모든 정보를 데이터베이스에서 조회한다.
- 관리자 페이지에 반려견의 정보가 테이블 형태로 출력된다.

9) 반려견 정보 수정

- 관리자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 관리자는 반려견의 이름, 나이, 종류 등을 수정할 수 있다.

10) 반려견 정보 삭제

- 관리자는 반려견의 정보를 삭제할 수 있다.
- 반려견 정보가 삭제됐다면 해당 반려견의 데이터는 데이터베이스에서 삭제된다.

11) 질병 정보 조회

- 관리자는 질병의 모든 정보를 데이터베이스에서 조회한다.
- 사용자 상세정보 페이지에 반려견의 질병 정보가 출력된다.

12) 진단 결과 조회

- 관리자는 반려견의 모든 질병 진단 결과를 데이터베이스에서 조회한다.
- 관리자 페이지에 반려견의 질병 진단 결과가 테이블의 값으로 출력된다.

13) 예약 조회

- 관리자는 동물병원 예약의 모든 정보를 데이터베이스에서 조회한다.
- 관리자 페이지에 동물병원 예약의 정보가 테이블 형태로 출력된다.

14) 예약 수정

- 관리자로부터 수정할 정보를 정보수정 폼을 통해 입력받는다.
- 데이터베이스에 저장된 값을 수정한다.
- 관리자는 동물병원 예약의 예약 시간, 예약 일정 등을 수정할 수 있다.

15) 예약 취소

- 관리자는 동물병원 예약을 취소할 수 있다.
- 병원예약이 취소됐다면 해당 병원예약 데이터는 데이터베이스에서 삭제된다.

## 16) 질병 진단 정보 시각화

- 모든 반려견의 질병 진단 정보를 파이 그래프와 같은 시각자료로 시각화한다.
- 관리자는 반려견의 질병 진단 정보를 시각자료로 한눈에 조회한다.



### 3. 프로세스(기능) 설계

#### 3.1. 시스템 아키텍처

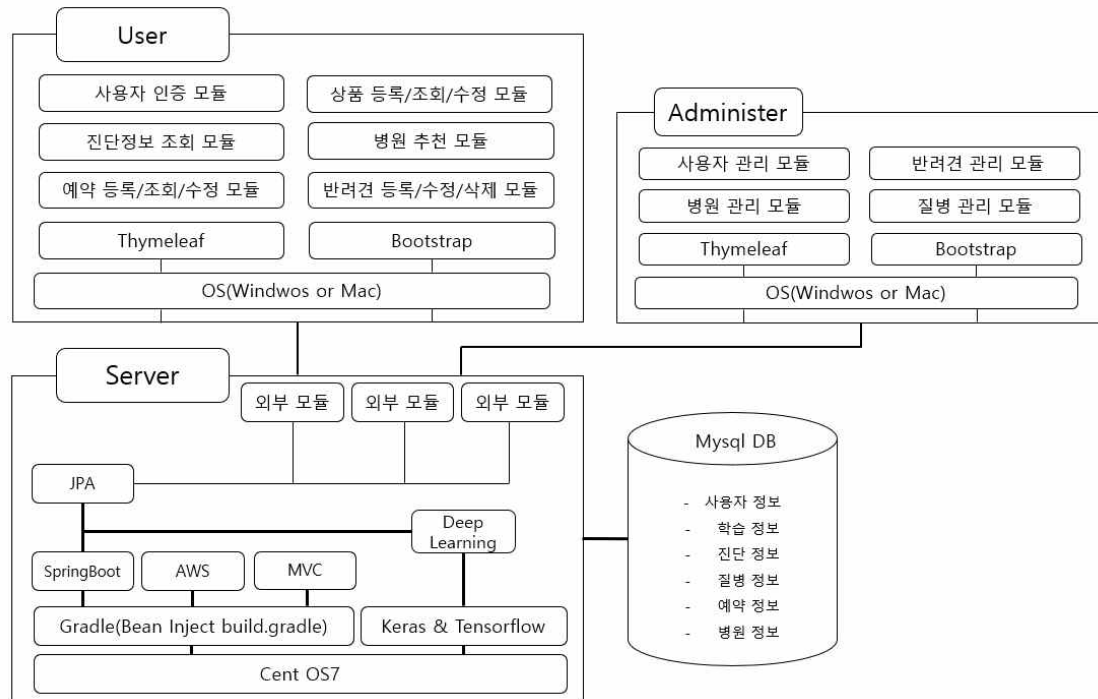


그림 2.1 시스템 아키텍처

웹기반 반려견 헬스케어 서비스 구해독은 웹과 서버로 구성된다. 웹서버는 AWS에서 제공하는 CentOS7 운영체제 기반으로 Gradle, Spring, Thymeleaf, Bootstrap을 사용하며, Gradle 프로젝트는 홈 디렉터리 구성 후 구현에 필요한 외부 모듈을 다운로드하여 설치한다. 또한, 웹서버는 AWS에서 RDS를 제공받고, 외부 API(JPA, JDBC)를 통해 받아온 데이터를 Deep Learning 기술을 이용하여 학습한 후 추론된 결과를 클라이언트(사용자, 관리자)에게 제공한다.

클라이언트는 Web UI를 이용하여 구해독 서비스를 이용할 수 있다. 웹 UI는 Bootstrap을 사용하며, 모든 리소스를 HTTP 인터페이스인 CRUD(Create, Read, Update, Delete) 메소드로 접근하기 위해 Restful하게 설계하였다. 사용자 기능으로 사용자 인증, 반려견 등록, 질병 조회, 병원 추천, 병원 예약, 시각화 기능을 지원한다. 관리자 기능으로 관리자 인증, 사용자 관리, 반려견 관리, 질병 관리, 병원 관리, 시각화 기능을 제공한다. 각 기능은 모듈화되어 독립적으로 운영된다.

### 3.2. 사용자기능 설계 (Sequence Diagram)

#### 1) 사용자 인증 - 사용자 가입 기능

사용자가 질병 진단, 상품 구매 및 판매, 진료 예약, 수익 관리 서비스를 이용하기 위해서는 사용자 가입을 통해 사용자 인증을 받아야 한다. 사용자 가입 기능을 구현하기 위해 아래와 같이 설계한다.

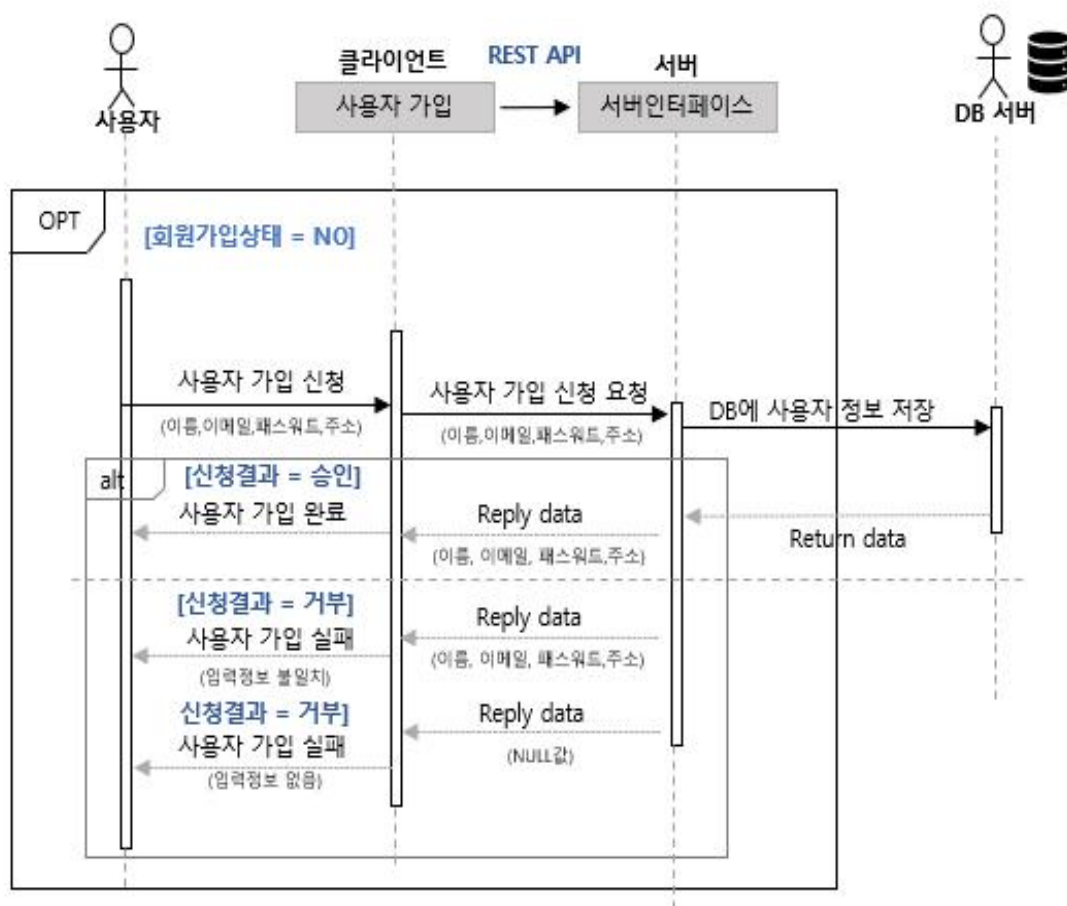


그림 3.2.1 사용자 가입의 Sequence Diagram

그림 3.2.1을 보면 처음 사용자는 사용자 가입이 되어있지 않은 상태이다. 웹 사이트에서 사용자 가입 신청을 통해 서버에 사용자 가입 신청 요청을 하고, 올바르게 정보를 입력하였다면 데이터베이스에 사용자 정보를 저장하고 사용자 가입을 승인받는다. 만약 사용자가 입력한 정보 중 패스워드와 패스워드 확인 값이 서로 다르다면 승인을 거부하며, 입력정보가 없는 경우에도 승인을 거부한다.

## 2) 사용자 인증 - 로그인 기능

사용자는 서비스 가입 후 서비스를 제공받기 위해 본인의 이메일/비밀번호를 이용한 로그인이 필요하다. 로그인 기능을 구현하기 위해 아래와 같이 설계한다.

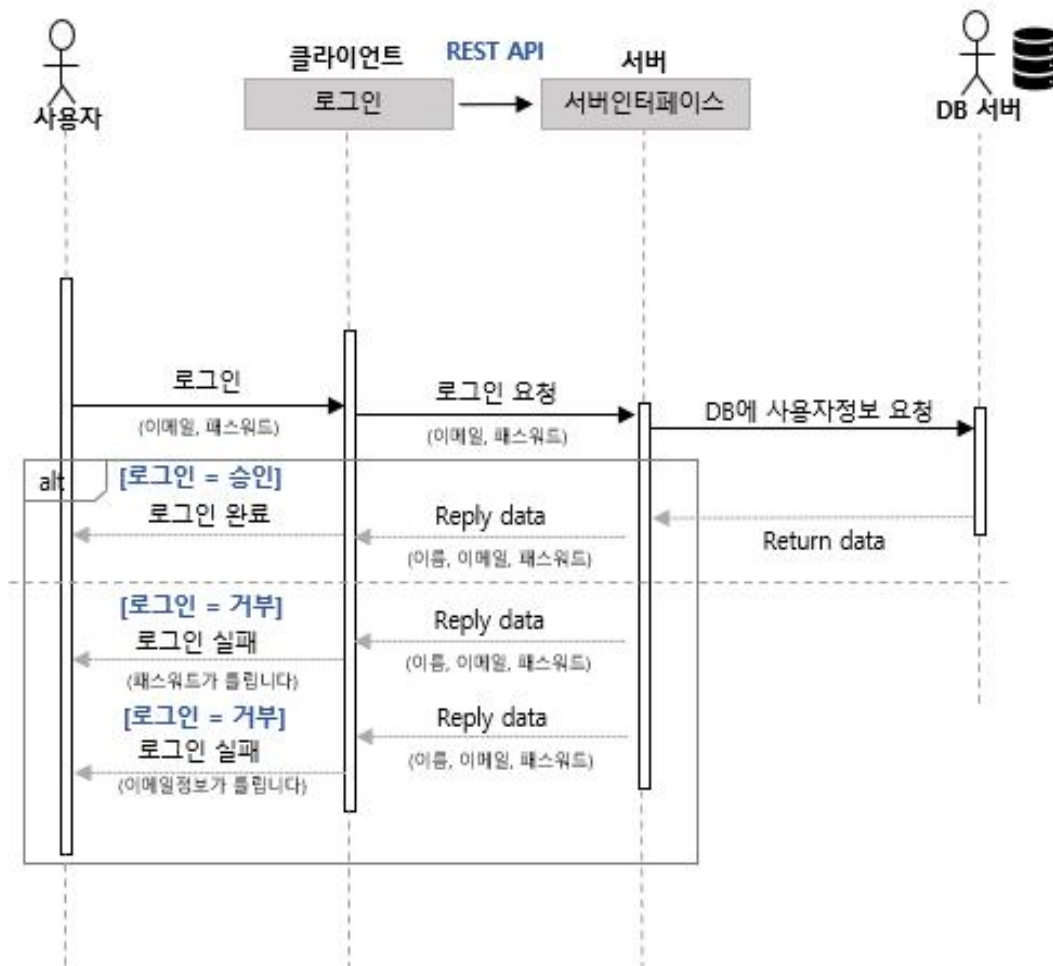


그림 3.2.2 로그인의 Sequence Diagram

그림 3.2.2를 보면 서비스 가입상태가 되어있는 경우, 사용자는 이메일 주소/패스워드를 입력 후 로그인 요청을 한다. 웹에서는 사용자가 로그인 버튼을 클릭하면 서버 인터페이스에 로그인 요청을 한다. 입력받은 정보를 바탕으로 서버는 데이터베이스에 사용자 정보가 올바른지 요청하고, 요청받은 정보를 서버로 다시 되돌려준다. 이메일 주소와 패스워드가 일치한다면 로그인을 승인하고 사용자가 요청하는 서비스를 제공한다. 만약 로그인 요청결과가 거부 되었을 경우, 사용자에게 로그인 실패 메시지와 실패 사유를 함께 나타낸다.

### 3) 사용자 인증 - 로그아웃 기능

사용자는 로그인 후 서비스를 종료하기 위해서는 로그아웃 기능이 필요하다. 로그아웃 기능을 구현하기 위해 아래와 같이 설계한다.

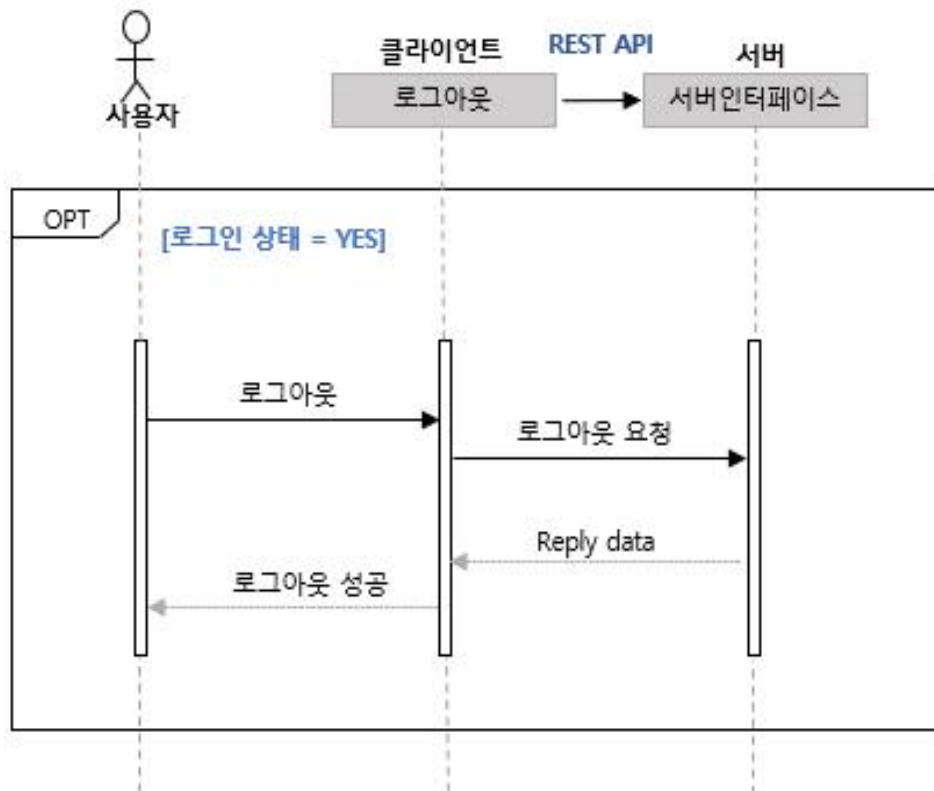


그림 3.2.3 로그아웃의 Sequence Diagram

그림 3.2.3을 보면 로그인이 되어있는 상태의 사용자는 서버에게 로그아웃 요청을 할 수 있다. 로그아웃 요청을 받으면 서버는 사용자와의 연결을 끊으면서 로그아웃이 성공적으로 완료되었다는 내용과 함께 메시지를 제공한다.

#### 4) 사용자 인증 - 정보수정 기능

사용자는 로그인 후 본인의 정보를 수정할 수 있다. 정보수정 기능을 구현하기 위해 아래와 같이 설계한다.

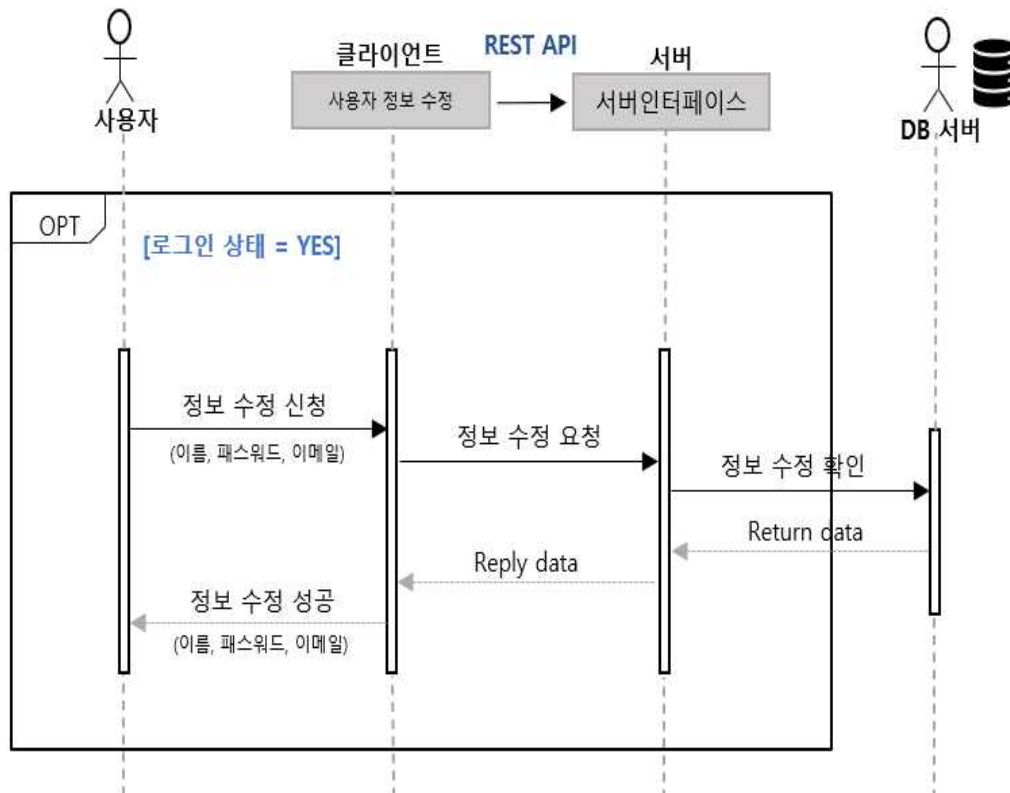


그림 3.2.4 정보수정의 Sequence Diagram

그림 3.2.4을 보면 로그인이 되어있는 상태의 사용자는 서버에게 정보 수정을 요청하고 있다. 서버는 요청정보를 확인한 후 데이터베이스의 정보를 수정하여 정보수정이 완료되었다는 내용과 함께 메시지를 제공한다.

## 5) 사용자 인증 - 사용자 탈퇴 기능

사용자는 로그인 후 사용자 탈퇴를 할 수 있다. 사용자 탈퇴 기능을 구현하기 위해 아래와 같이 설계한다.

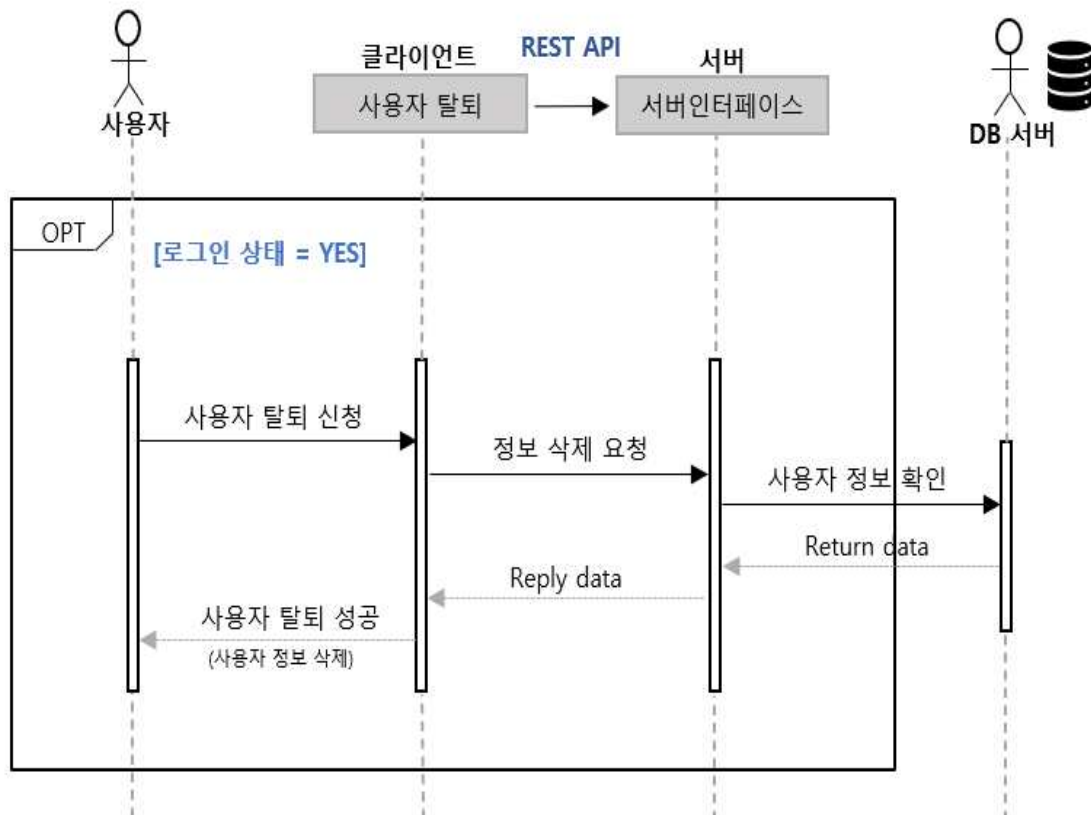


그림 3.2.5 사용자 탈퇴의 Sequence Diagram

그림 3.2.5를 보면 로그인이 되어있는 상태의 사용자는 서버에게 사용자 탈퇴 요청을 할 수 있다. 사용자 탈퇴 요청을 받으면 서버는 사용자와의 연결을 끊으면서 사용자 탈퇴가 성공적으로 완료되었다는 내용과 함께 메시지를 제공한다.

## 6) 반려견 등록 - 반려견 정보 등록 기능

사용자는 로그인 후 반려견 정보를 입력할 수 있다. 반려견 정보입력 서비스를 구현하기 위해 아래와 같이 설계한다.

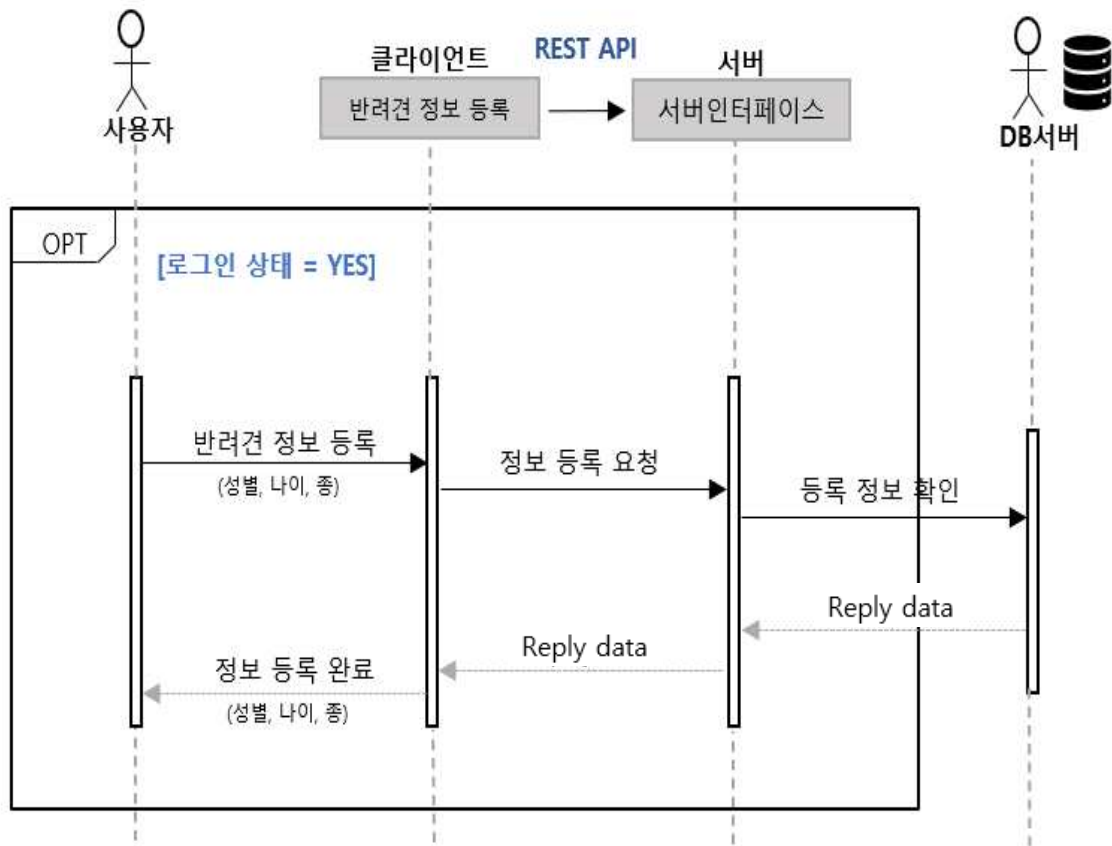


그림 3.2.6 반려견 정보 등록의 Sequence Diagram

그림 3.2.6을 보면 웹은 사용자에게 반려견의 정보를 입력할 수 있는 양식을 제공하고 사용자는 입력한 정보를 저장하기 위해 서버에 요청한다. 서버는 데이터베이스의 반려견 테이블에 사용자가 입력한 반려견 정보를 저장하고 사용자에게 메시지를 전달해 준다.

## 7) 반려견 등록 - 반려견 정보 조회 기능

사용자는 반려견을 등록한 후 자신의 반려견 정보를 조회할 수 있다. 반려견 정보 조회 서비스 기능을 구현하기 위해 아래와 같이 설계한다.

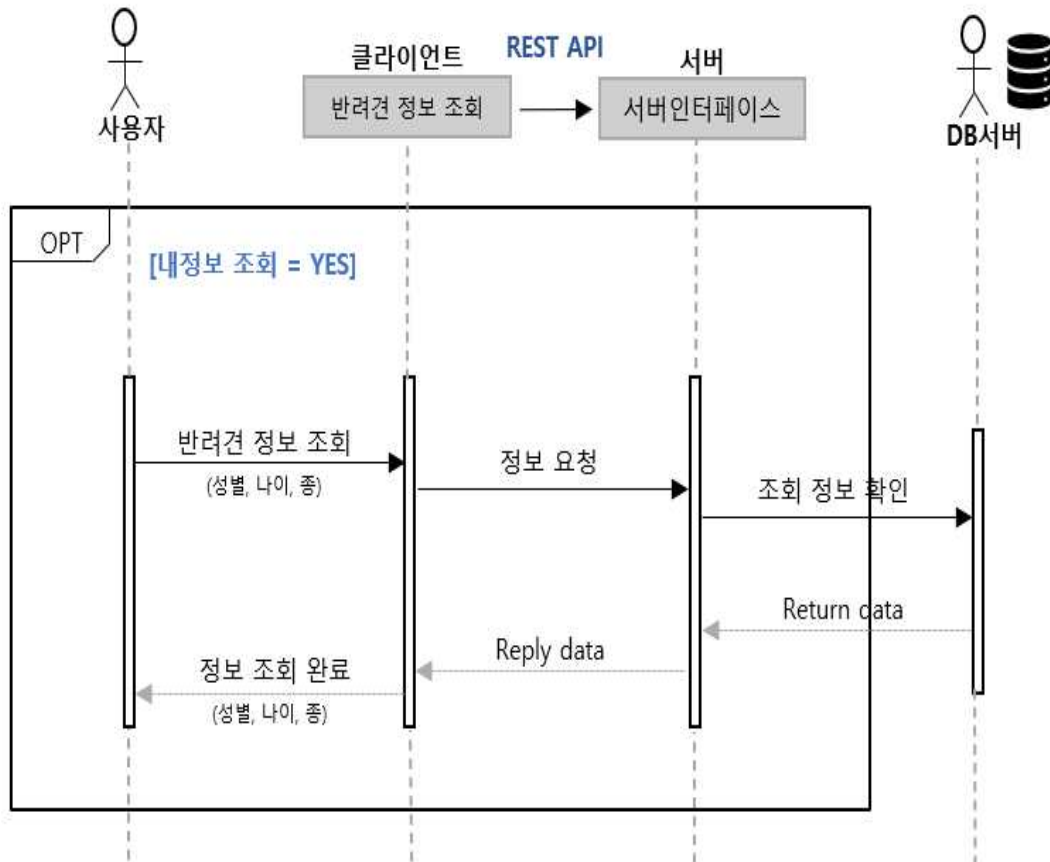


그림 3.2.7 반려견 정보 조회의 Sequence Diagram

그림 3.2.7을 보면 사용자는 등록한 반려견 정보 조회를 서버에게 요청한다. 서버는 데이터베이스의 반려견 테이블에 저장된 반려견 정보를 확인하여 사용자가 요청한 조회정보의 데이터를 전달해 준다.



## 8) 반려견 등록 - 반려견 정보 수정 기능

사용자는 로그인 후 자신의 반려견 정보를 수정할 수 있다. 반려견 정보수정 서비스 기능을 구현하기 위해 아래와 같이 설계한다.

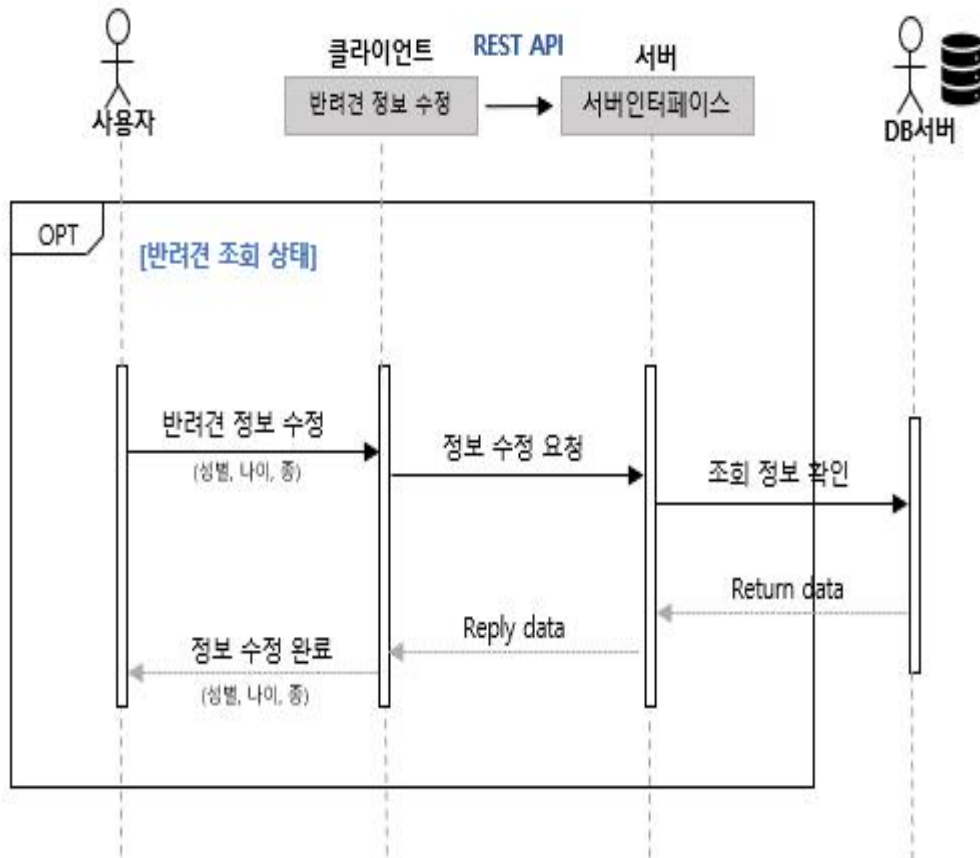


그림 3.2.8 반려견 정보 수정의 Sequence Diagram

그림 3.2.8을 보면 사용자는 수정한 반려견 정보의 등록을 서버에게 요청한다. 서버는 데이터베이스의 반려견 테이블에 저장된 반려견 정보를 확인하여 사용자가 요청한 등록 정보로 변경한 후 사용자에게 메시지를 전달해준다.

## 9) 반려견 등록 - 반려견 정보 삭제 기능

사용자는 로그인 후 자신의 반려견 정보를 삭제할 수 있다. 반려견 정보삭제 서비스 기능을 구현하기 위해 아래와 같이 설계한다.

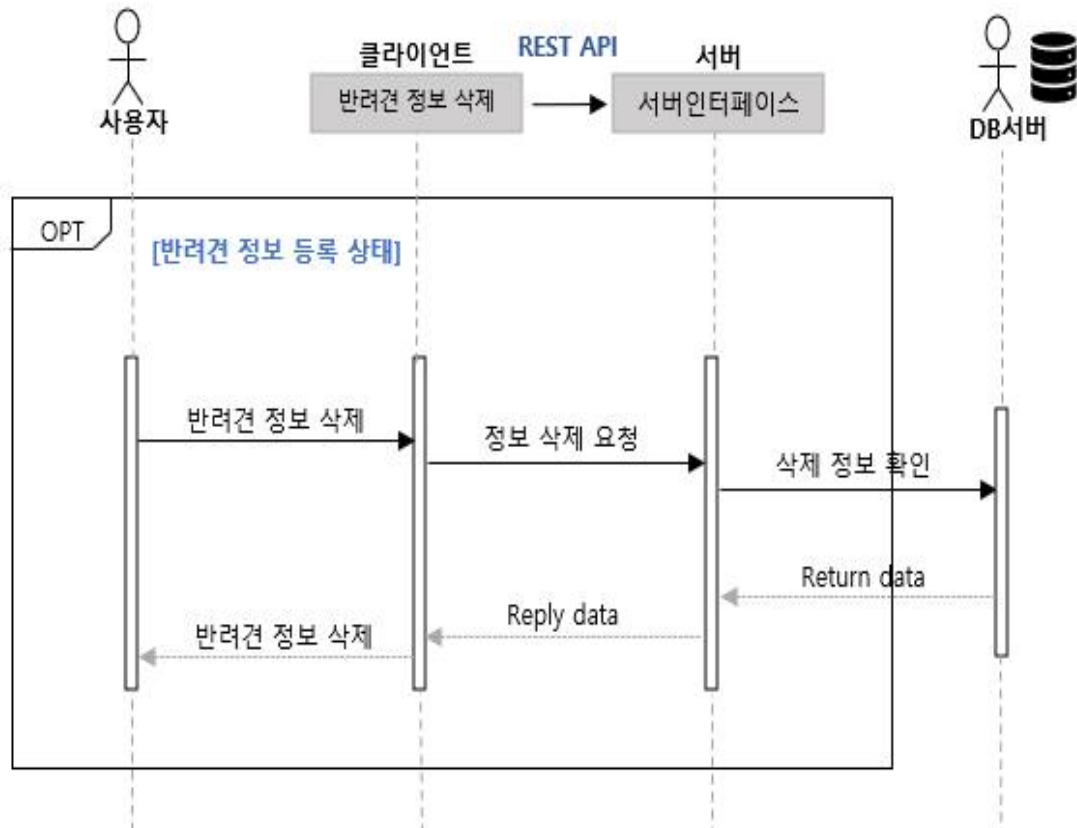


그림 3.2.9 반려견 정보 삭제의 Sequence Diagram

그림 3.2.9를 보면 사용자는 서버에게 등록된 반려견 정보 삭제를 요청한다. 서버는 요청 정보를 확인한 후 데이터베이스 반려견 테이블에 저장된 정보를 삭제한다. 서버는 사용자에게 메시지로 정보를 전달해준다.

## 10) 질병 조회 - 질병 증상 등록 기능

사용자는 서버에서 제공하는 진단 차트에 반려견의 질병 정보를 입력하여 등록할 수 있다. 증상 등록 서비스 기능을 구현하기 위해 아래와 같이 설계한다.

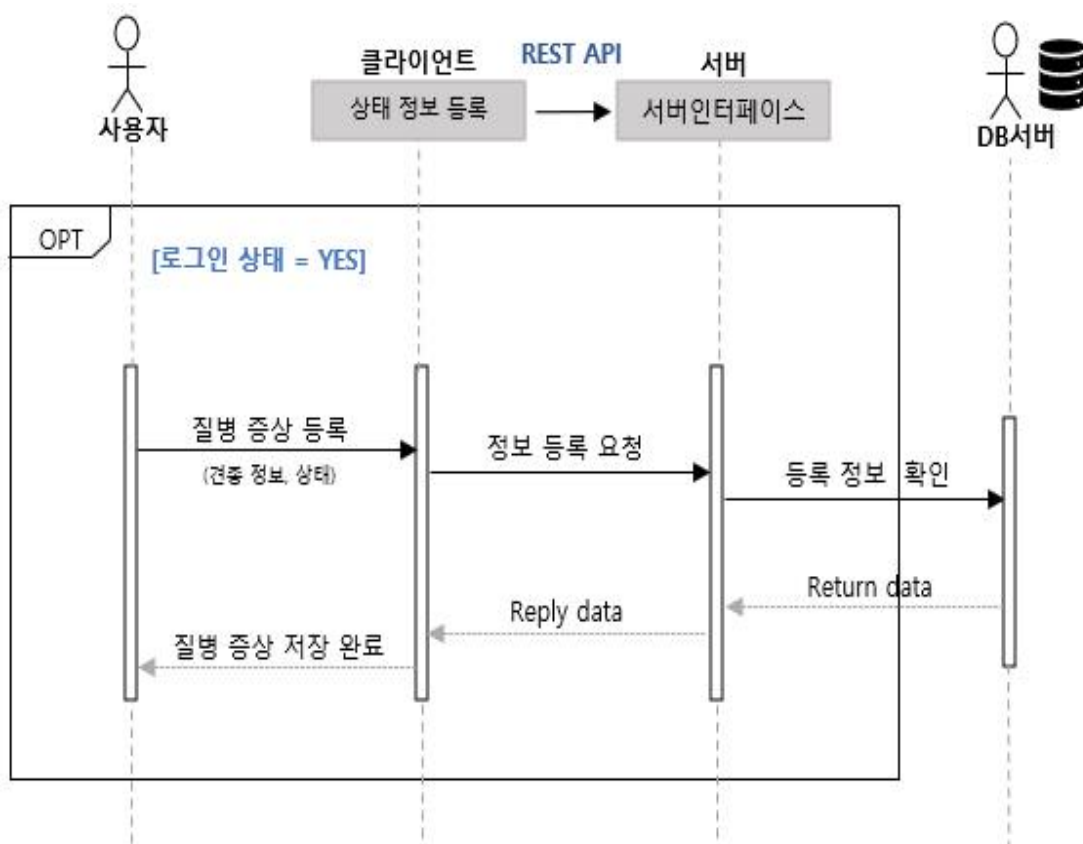


그림 3.2.10 질병 증상 등록의 Sequence Diagram

그림 3.2.10을 보면 사용자는 웹에서 제공하는 양식에 따라 반려견 질병 증상을 작성하고 서버에 등록을 요청하고 있다. 서버는 데이터베이스의 반려견 테이블에 요청 정보를 저장하고 사용자에게 메시지를 통해 정보를 전달해준다.

## 11) 질병 조회, 시각화 기능 - 진단 결과 조회 기능 및 질병 진단 결과 시각화 기능

사용자는 등록된 증상으로 진단 서비스를 제공받을 수 있다. 서버에서 제공하는 진단 차트에 자신의 반려견 정보를 입력할 경우 질병 진단조회 서비스를 받을 수 있다. 질병 진단조회 서비스 기능을 구현하기 위해 아래와 같이 설계한다.

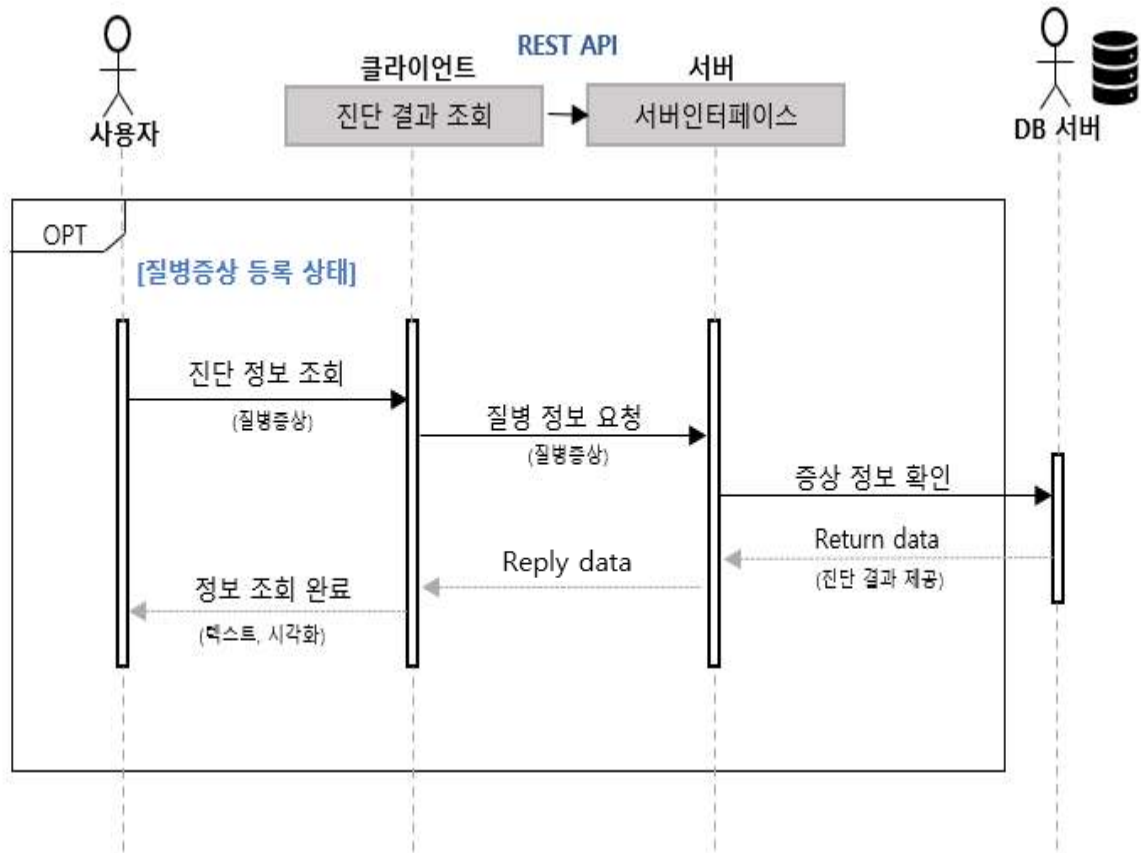


그림 3.2.11 진단 결과 조회 및 질병 진단 결과 시각화의 Sequence Diagram

그림 3.2.11을 보면 사용자는 웹에서 반려견의 정보를 서버에게 요청한다. 서버는 질병 증상 정보를 확인하고 데이터베이스의 질병 정보 테이블에서 증상에 맞는 데이터 값을 사용자에게 텍스트 형식과 시각화된 자료로 전달해준다.

## 21) 병원 추천 - 위치 기반 추천 기능

사용자는 질병 진단 서비스에서 위치 기반으로 병원을 추천받을 수 있다. 병원 추천 기능을 구현하기 위해 아래와 같이 설계한다.

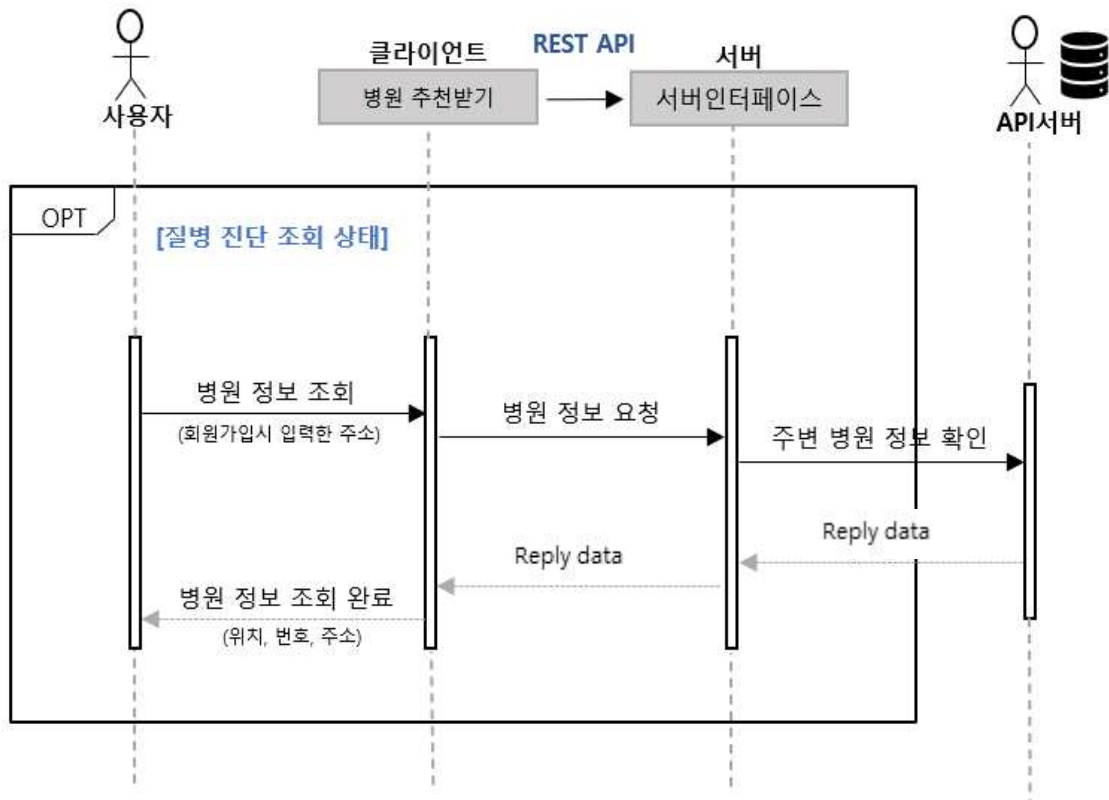


그림 3.2.21 위치 기반 추천의 Sequence Diagram

그림 3.2.21을 보면 사용자는 주소기반으로 병원 정보 조회를 서버에게 요청한다. 서버는 사용자의 주소를 확인하고 API 데이터베이스에서 병원의 정보를 사용자에게 전달해 준다.

## 22) 병원 예약 - 예약 등록 기능

사용자는 이용하고자 하는 동물병원 예약이 가능하다. 예약 등록 기능을 구현하기 위해 아래와 같이 설계한다.

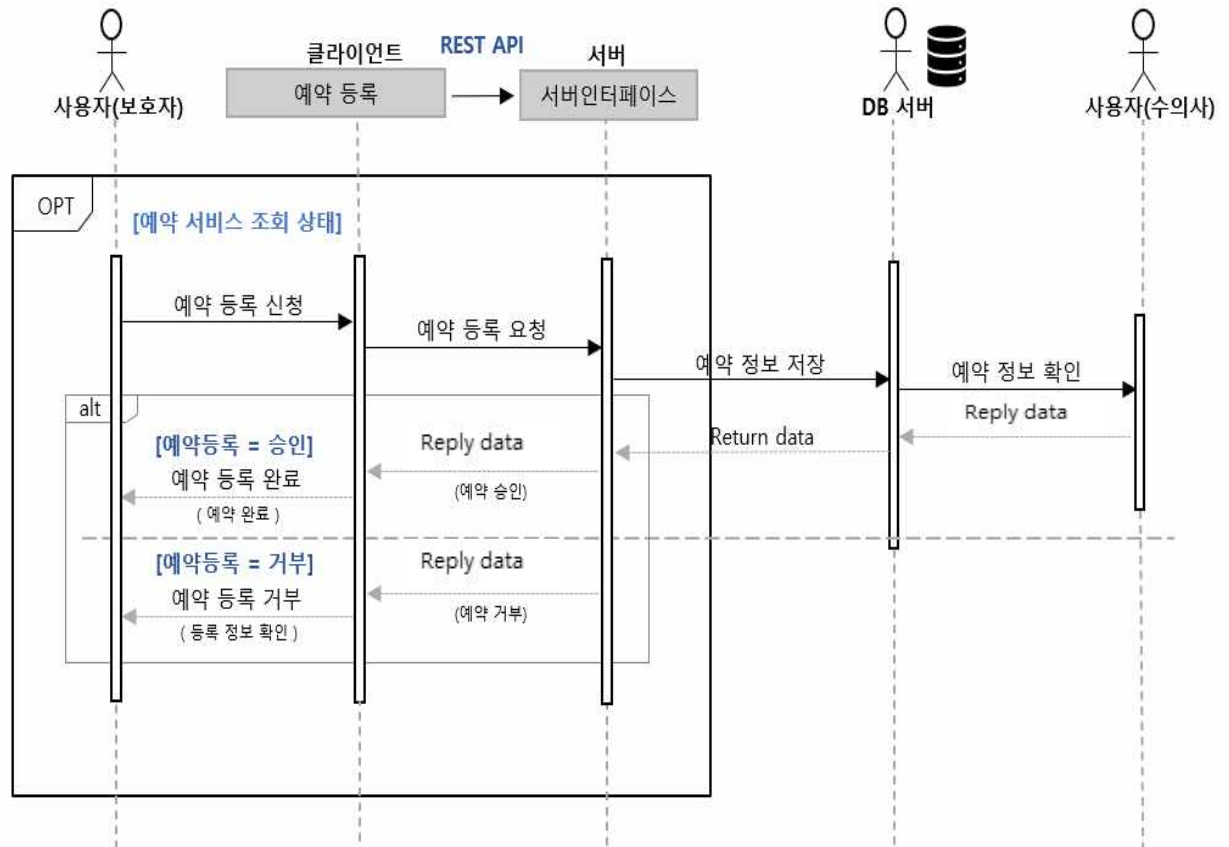


그림 3.2.22 예약 등록의 Sequence Diagram

그림 3.2.22를 보면 사용자가 웹에서 예약 등록을 신청하면 서버는 사용자가 등록한 예약 정보를 데이터베이스에 저장한다. 사용자는 웹에서 데이터베이스에 저장된 예약 정보를 요청하여 확인하고 예약 승인 또는 거부를 할 수 있다.

## 23) 병원 예약 - 예약 조회 기능

사용자는 본인이 등록한 예약 정보를 조회할 수 있다. 예약 조회 기능을 구현하기 위해 아래와 같이 설계한다.

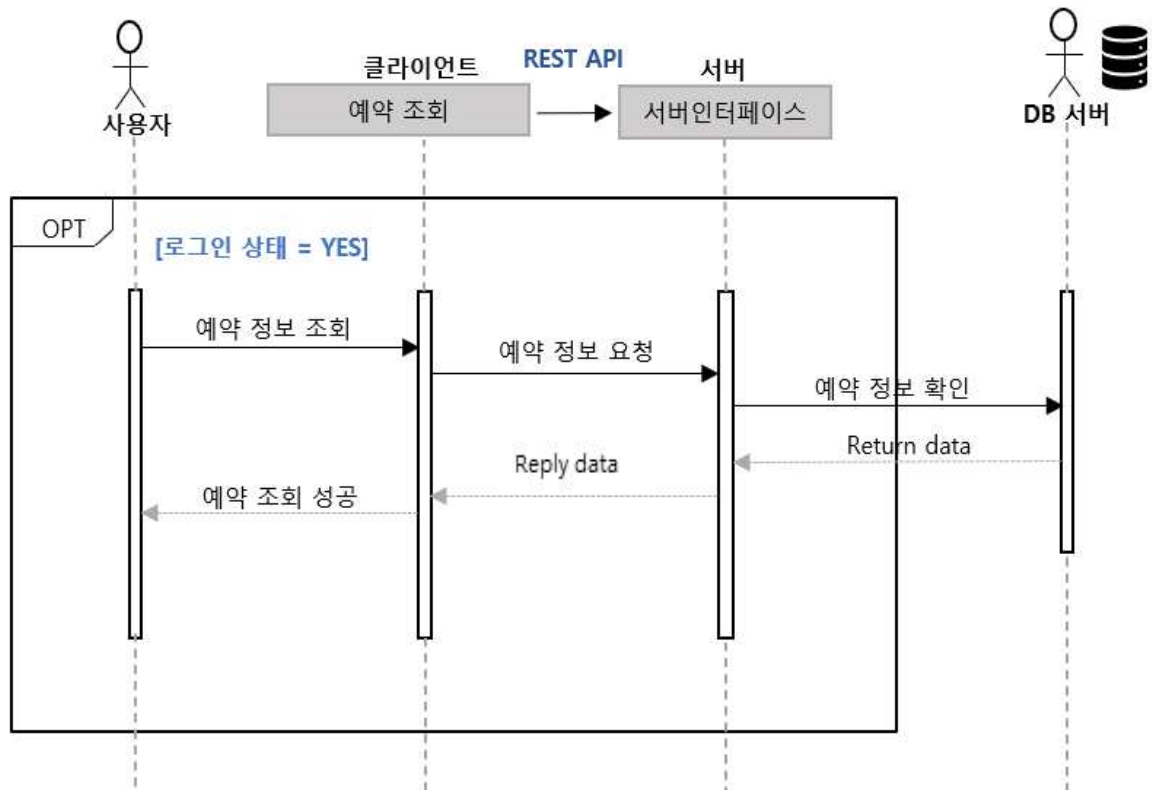


그림 3.2.23 예약 조회의 Sequence Diagram

그림 3.2.23을 보면 사용자는 웹에서 본인이 등록한 예약 정보를 서버에게 요청하면 서버는 예약 정보를 데이터베이스에서 확인한다. 서버는 요청 데이터를 전송하여 예약 정보 데이터를 사용자에게 보여준다.

## 24) 병원 예약 - 예약 수정 기능

사용자는 본인이 등록한 예약 정보를 수정할 수 있다. 예약 수정기능을 구현하기 위해 아래와 같이 설계한다.

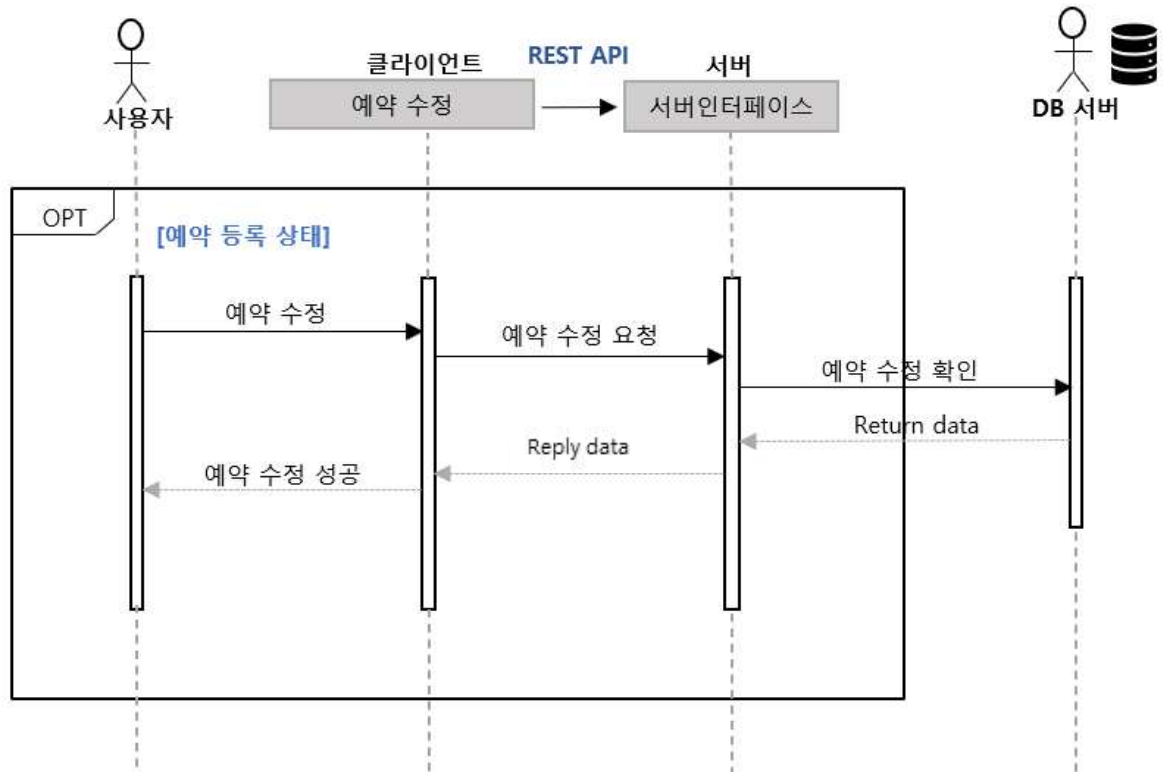


그림 3.2.24 예약 수정의 Sequence Diagram

그림 3.2.24를 보면 사용자는 웹에서 본인이 등록한 예약 정보를 수정하여 수정한 정보를 서버에 등록 요청을 한다. 서버는 수정된 예약 정보를 확인하고 데이터베이스에 저장된 등록 정보를 수정된 정보로 수정시킨다.



## 25) 병원 예약 - 예약 취소 기능

사용자는 본인이 등록한 예약을 취소할 수 있다. 예약 취소기능을 구현하기 위해 아래와 같이 설계한다.

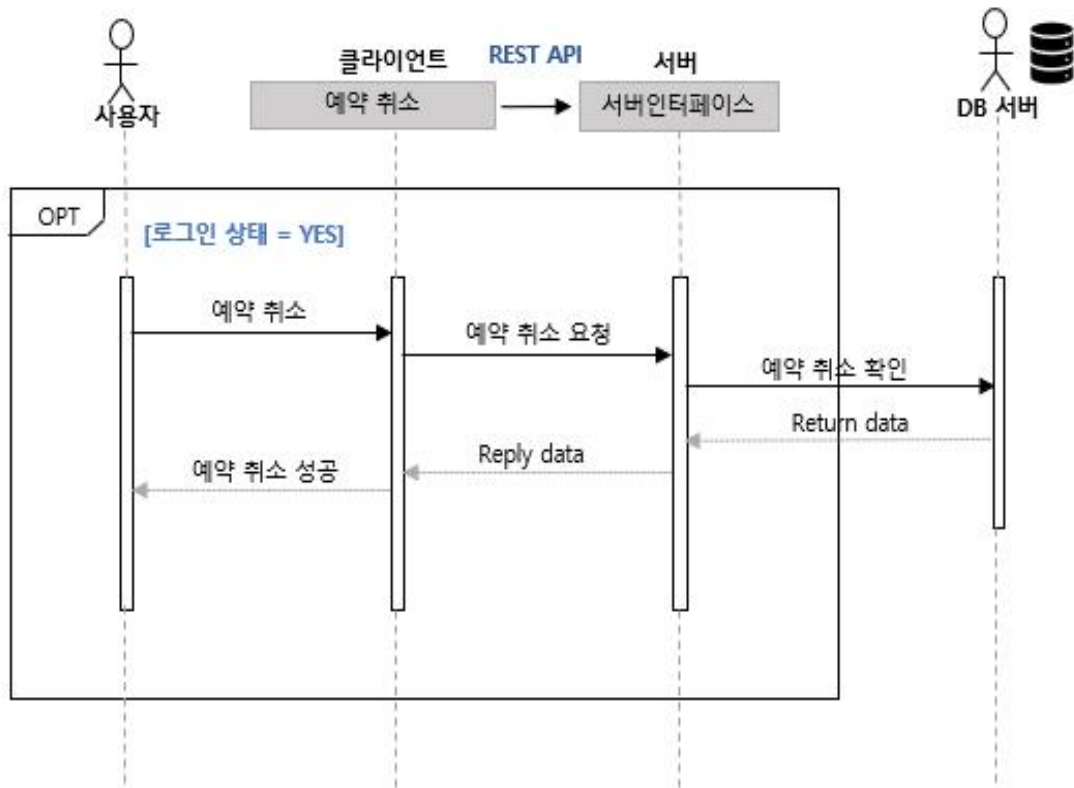


그림 3.2.25 예약 취소의 Sequence Diagram

그림 3.2.25를 보면 사용자는 본인이 등록한 예약 정보를 서버에게 요청하여 조회한 후, 예약 취소를 서버에게 요청한다. 서버는 데이터베이스에서 예약 취소 정보를 확인하고 등록된 예약 정보를 삭제한다.

### 3.3. 관리자 기능 설계 (Sequence Diagram)

#### 1) 관리자 인증 - 로그인 기능

관리자는 서비스를 관리하기 위해 등록된 본인의 이메일과 비밀번호를 이용한 로그인 필요하다. 로그인 기능을 구현하기 위해 아래와 같이 설계한다.

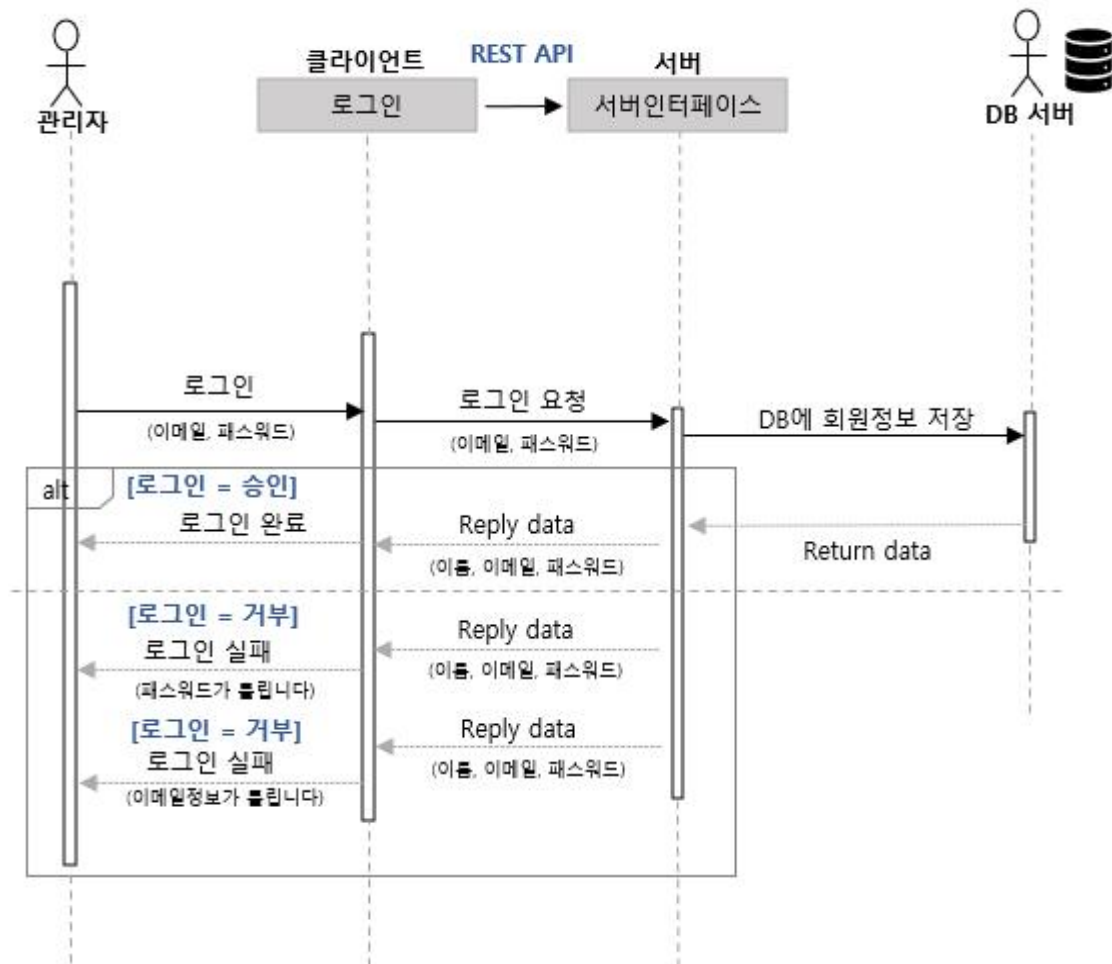


그림 3.3.1 로그인의 Sequence Diagram

그림 3.3.1을 보면 관리자는 관리자 이메일과 비밀번호 값을 입력하여 관리자로서의 로그인을 서버에게 요청한다. 웹에서는 사용자가 로그인 버튼을 클릭하면 서버 인터페이스에 로그인 요청을 한다. 입력받은 정보를 바탕으로 서버는 데이터베이스 관리자 정보에 관리자가 맞는지 확인하고 관리자 인증이 완료되면 서버는 관리자에게 관리자 홈을 제공한다. 관리자는 관리자 홈과 로그인 완료 메시지를 받는다. 만약 관리자 인증이 거부되면 거부 된 이유를 알려준다.

## 2) 관리자 인증 - 로그아웃 기능

관리자는 로그아웃 기능이 필요하다. 로그아웃 기능을 구현하기 위해서 아래와 같이 설계한다.

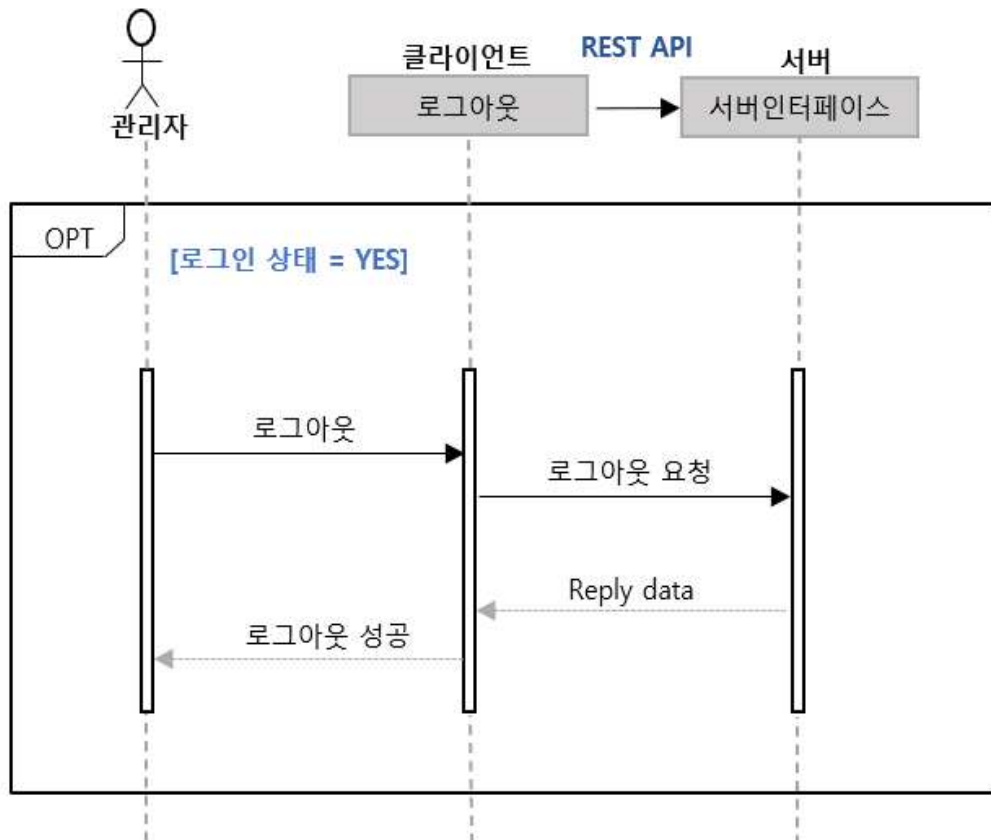


그림 3.3.2 로그아웃의 Sequence Diagram

그림 3.3.2를 보면 관리자는 서버에게 로그아웃 요청을 하면 로그아웃에 성공한다.

## 3) 관리자 인증 - 정보수정 기능

관리자는 본인의 정보를 수정할 수 있다. 정보수정 기능을 구현하기 위해서 아래와 같이 설계한다.

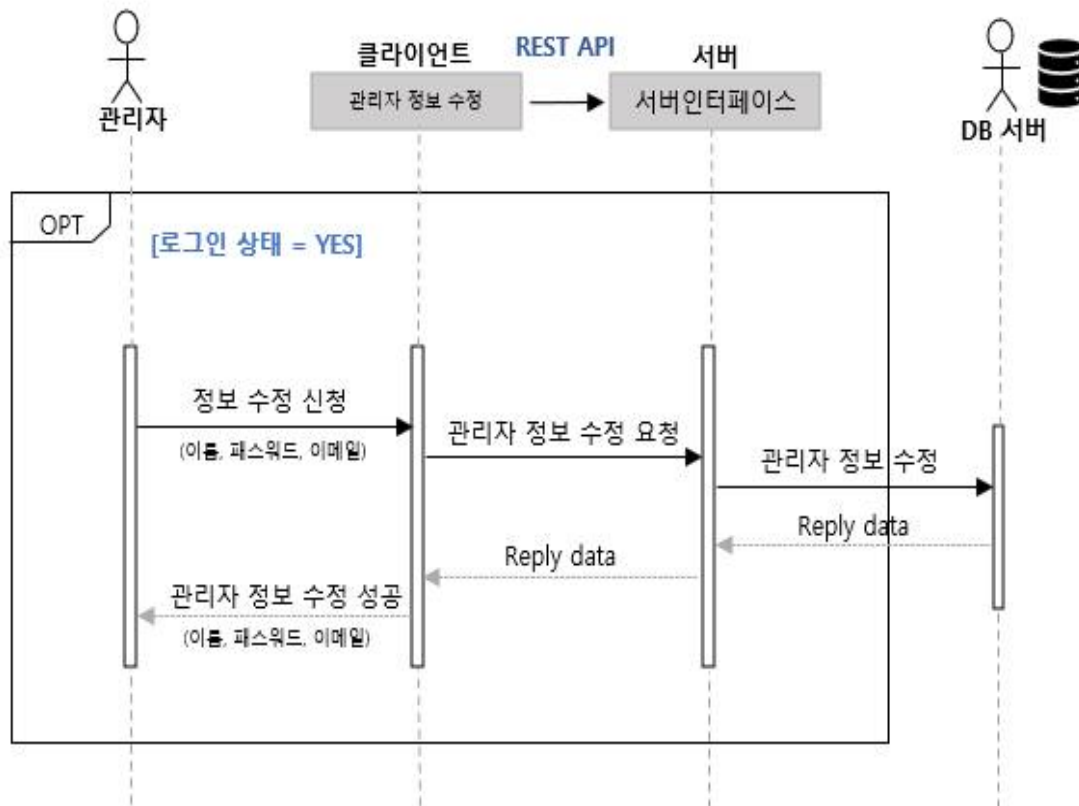


그림 3.3.3 정보수정의 Sequence Diagram

그림 3.3.3을 보면 관리자는 서버에게 수정한 정보의 등록을 요청하고 있다. 서버는 데이터베이스의 관리자 테이블에서 관리자 정보를 확인한 후 수정된 정보를 저장한다. 서버는 메시지를 사용하여 관리자에게 정보를 전달해준다.

#### 4) 관리자 인증 - 관리자 탈퇴 기능

관리자는 탈퇴 기능이 필요하다. 사용자 탈퇴 기능을 구현하기 위해서 아래와 같이 설계한다.

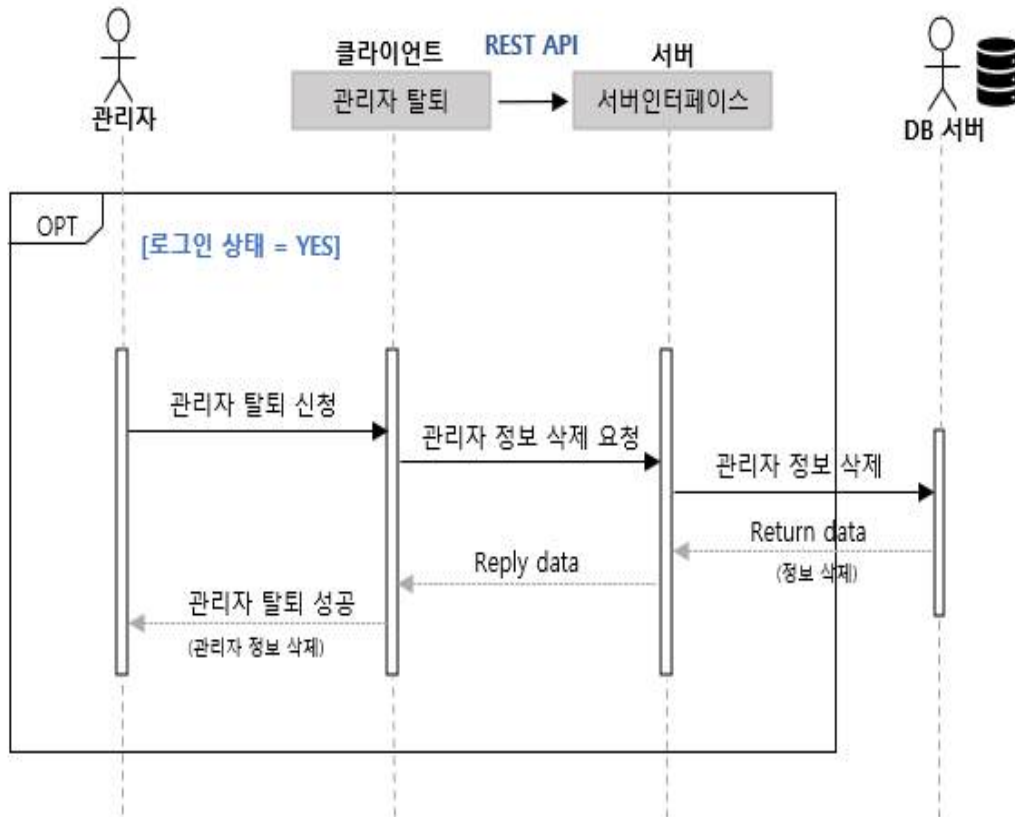


그림 3.3.4 관리자 탈퇴의 Sequence Diagram

그림 3.3.4를 보면 관리자는 서버에게 탈퇴를 요청하였다. 서버는 데이터베이스의 관리자 테이블에 요청한 관리자 정보를 확인한 후 정보를 삭제한다. 메시지를 통해 관리자에게 정보를 전달해 준다.

## 5) 사용자 관리 - 사용자 정보 조회 기능

관리자는 사용자 관리를 위하여 사용자를 조회할 수 있는 권한을 가지고 있어야 한다. 사용자 정보 조회 기능을 구현하기 위해서 아래와 같이 설계한다.

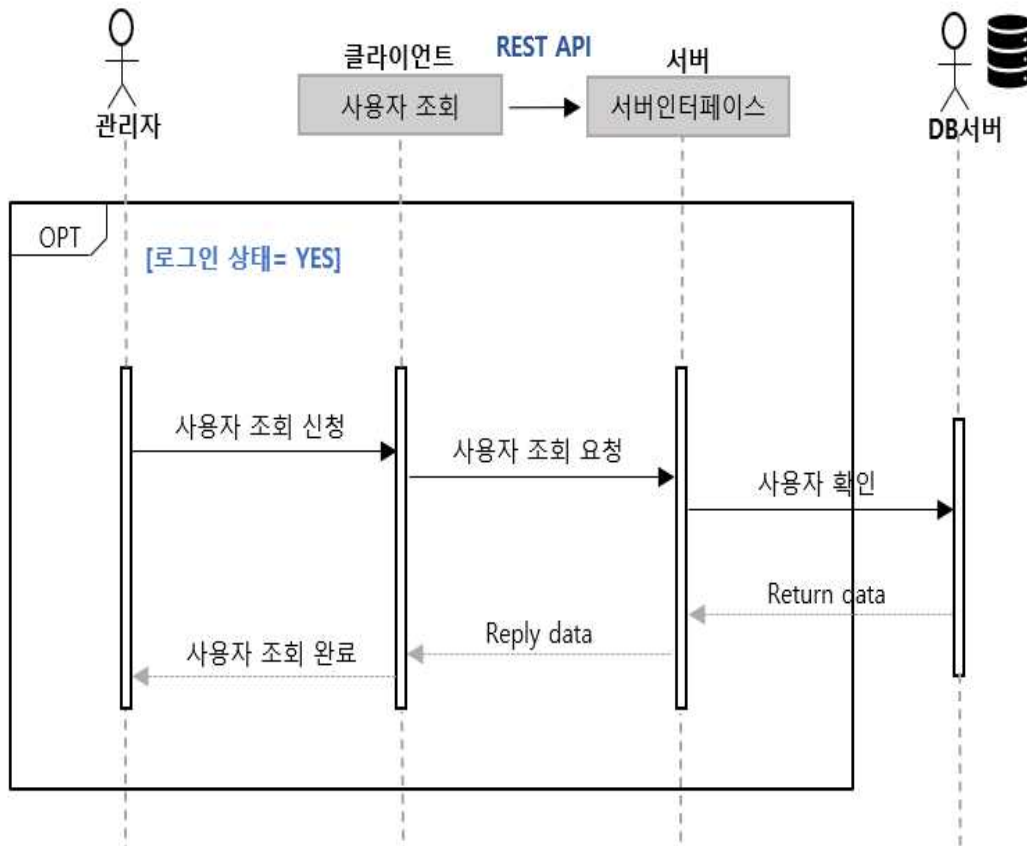


그림 3.4.5 사용자 정보 조회의 Sequence Diagram

그림 3.4.5를 보면 관리자는 서버로 사용자 조회 요청을 한다. 서버 인터페이스는 데이터베이스에 사용자 정보를 요청하고 사용자 정보 리스트를 받아온다. 사용자 정보 리스트는 관리자에게 반환된다.

## 6) 사용자 관리 - 사용자 정보 수정 기능

관리자는 사용자의 정보를 수정할 권한을 가지고 있다. 사용자 정보수정 기능을 구현하기 위해 아래와 같이 설계한다.

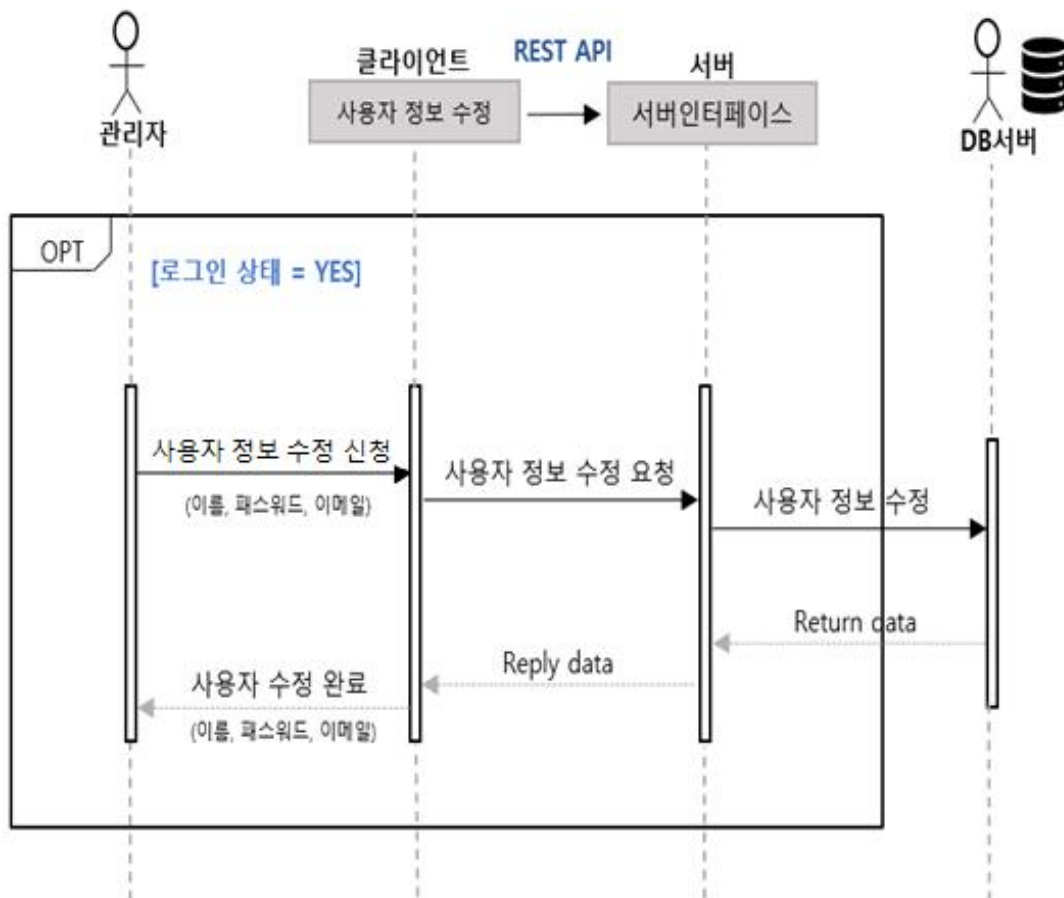


그림 3.4.6 사용자 정보 수정의 Sequence Diagram

그림 3.4.6을 보면 관리자는 서버로 특정 사용자의 정보수정을 요청한다. 서버는 데이터베이스에서 사용자의 정보를 확인하고 관리자의 요청에 맞게 사용자의 정보를 수정한다. 수정 정보는 메시지를 통해 관리자에게 전달한다.

## 7) 사용자 관리 - 사용자 정보 삭제 기능

관리자는 부적절한 특정 사용자를 삭제할 수 있는 권한을 가지고 있다. 특정 부적절한 행동을 하는 사용자들을 삭제할 수 있다. 사용자 삭제 기능을 구현하기 위해 아래와 같이 설계한다.

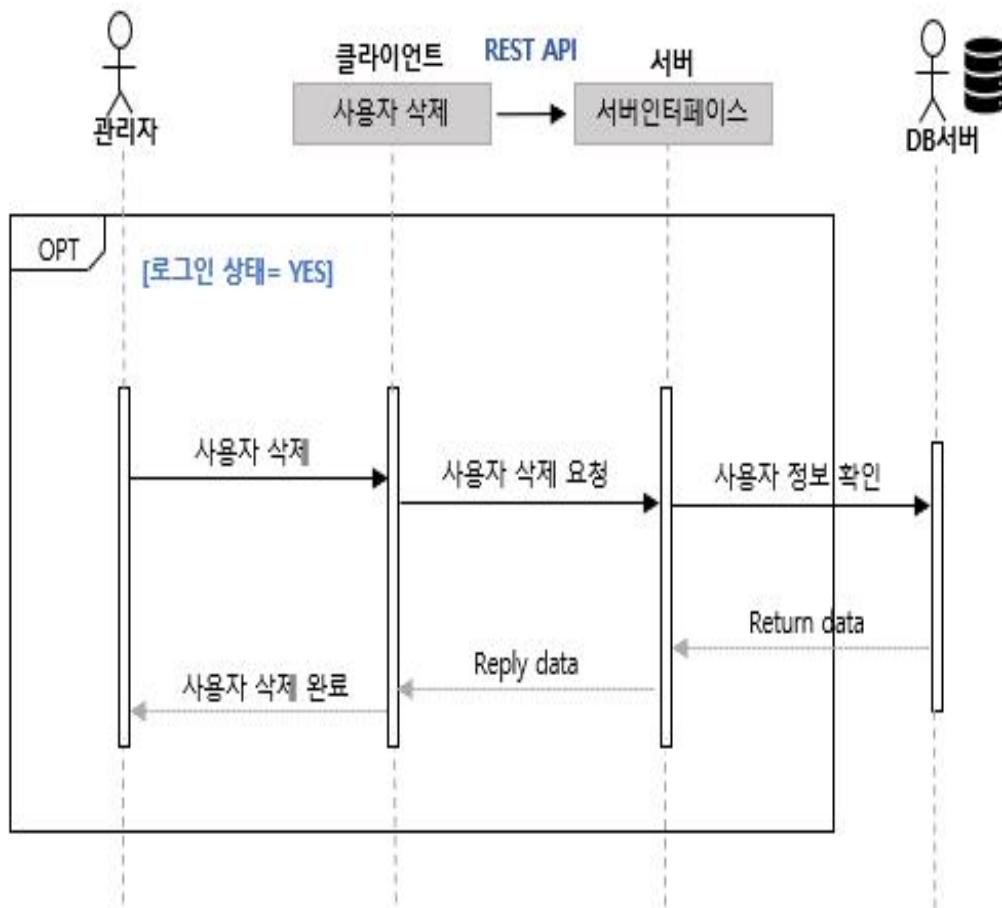


그림 3.4.7 사용자 정보 삭제의 Sequence Diagram

그림 3.4.7을 보면 관리자는 서버에게 사용자의 삭제를 요청하고 있다. 서버는 데이터베이스의 사용자 테이블에 접근하여, 특정 사용자의 정보를 삭제한다. 사용자 삭제가 완료되면 관리자에게 삭제가 완료되었다고 메시지를 통해 알려준다.



## 8) 반려견 관리 - 반려견 정보 조회 기능

관리자는 사용자가 등록한 반려견 정보를 조회할 수 있다. 반려견 정보 조회기능을 구현하기 위해 아래와 같이 설계한다.

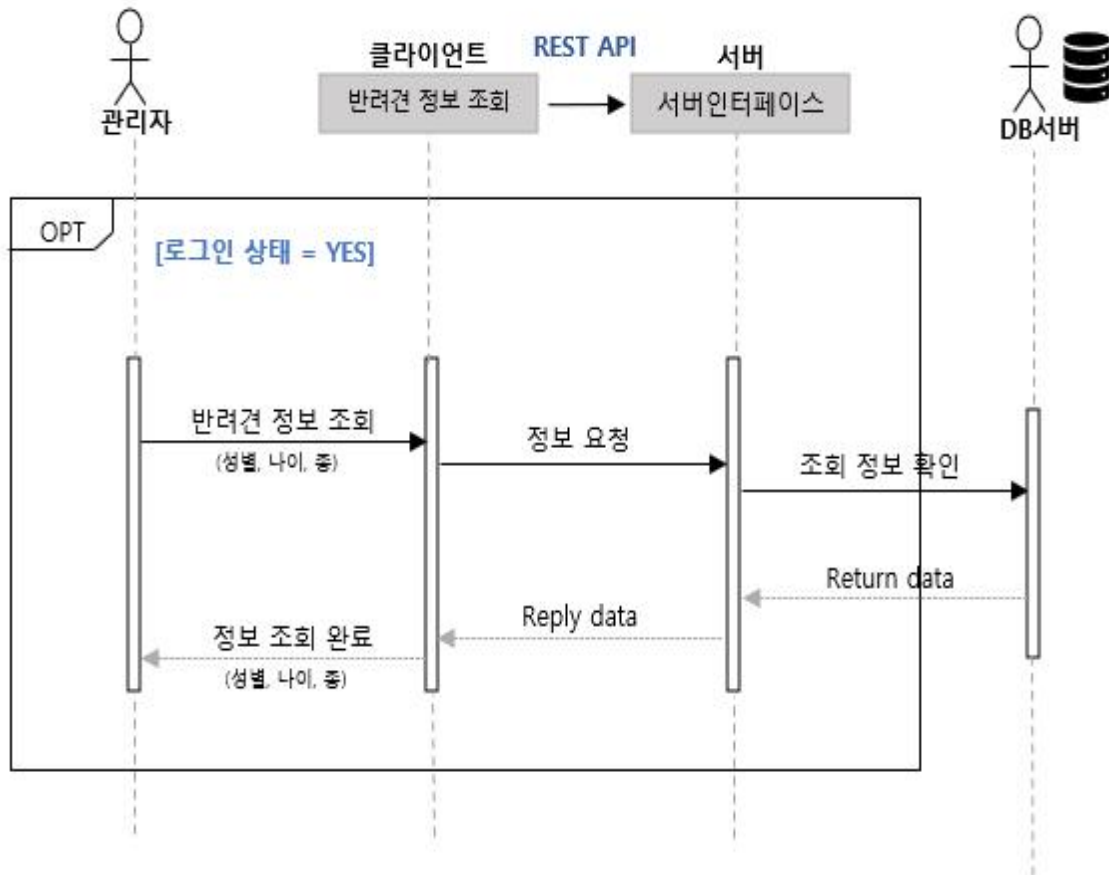


그림 3.3.8 반려견 정보 조회의 Sequence Diagram

그림 3.3.8을 보면 관리자는 서버에게 반려견의 정보 조회를 요청한다. 서버는 데이터베이스의 반려견 테이블에서 요청한 정보를 조회하여 메시지로 사용자에게 정보를 전달한다.

## 9) 반려견 관리 - 반려견 정보 수정 기능

관리자는 사용자가 등록한 반려견 정보를 수정할 수 있다. 반려견 정보수정 기능을 구현하기 위해 아래와 같이 설계한다.

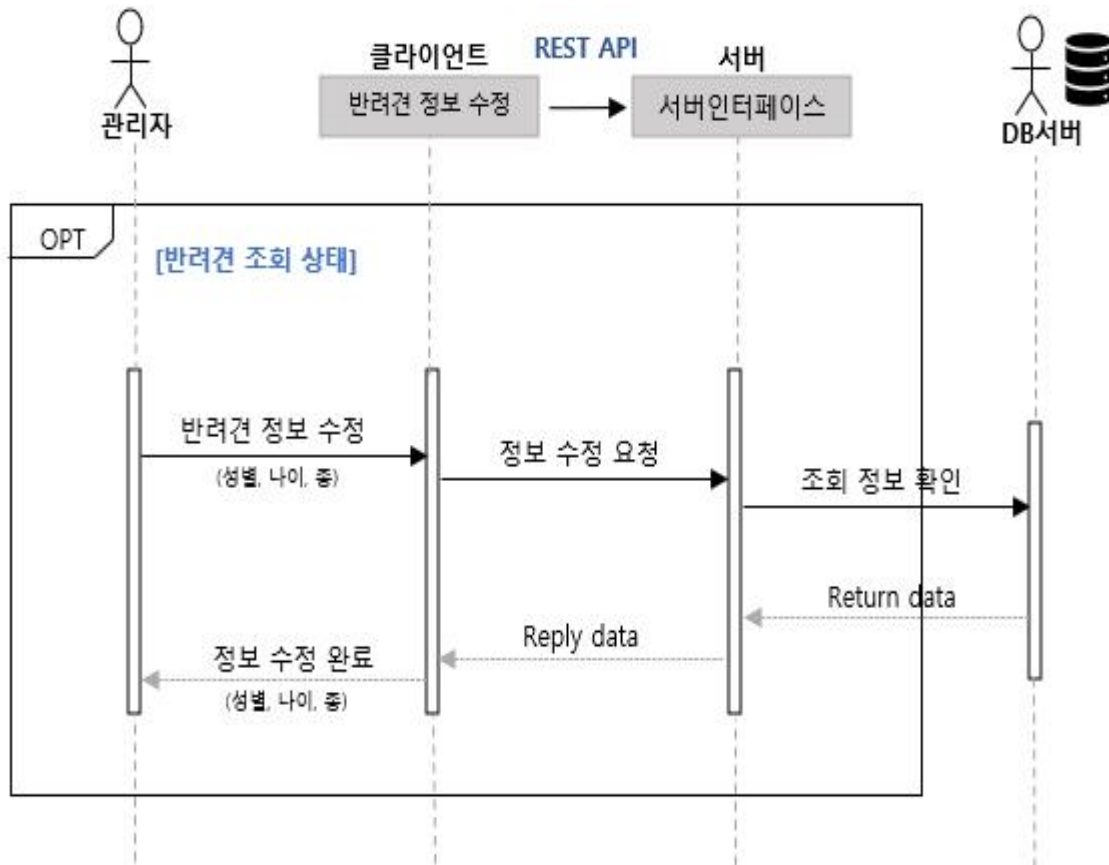


그림 3.3.9 반려견 정보 수정의 Sequence Diagram

그림 3.3.9를 보면 관리자는 서버에게 수정된 반려견 정보 등록을 요청한다. 서버는 데이터베이스의 반려견 테이블에서 반려견 정보를 확인하고 관리자가 요청한 정보로 수정하여 저장한다. 관리자에게 수정된 정보를 메시지로 전달한다.

## 10) 반려견 관리 - 반려견 정보 삭제 기능

관리자는 사용자가 등록한 반려견 정보를 삭제할 수 있다. 반려견 정보 삭제 기능을 구현하기 위해 아래와 같이 설계한다.

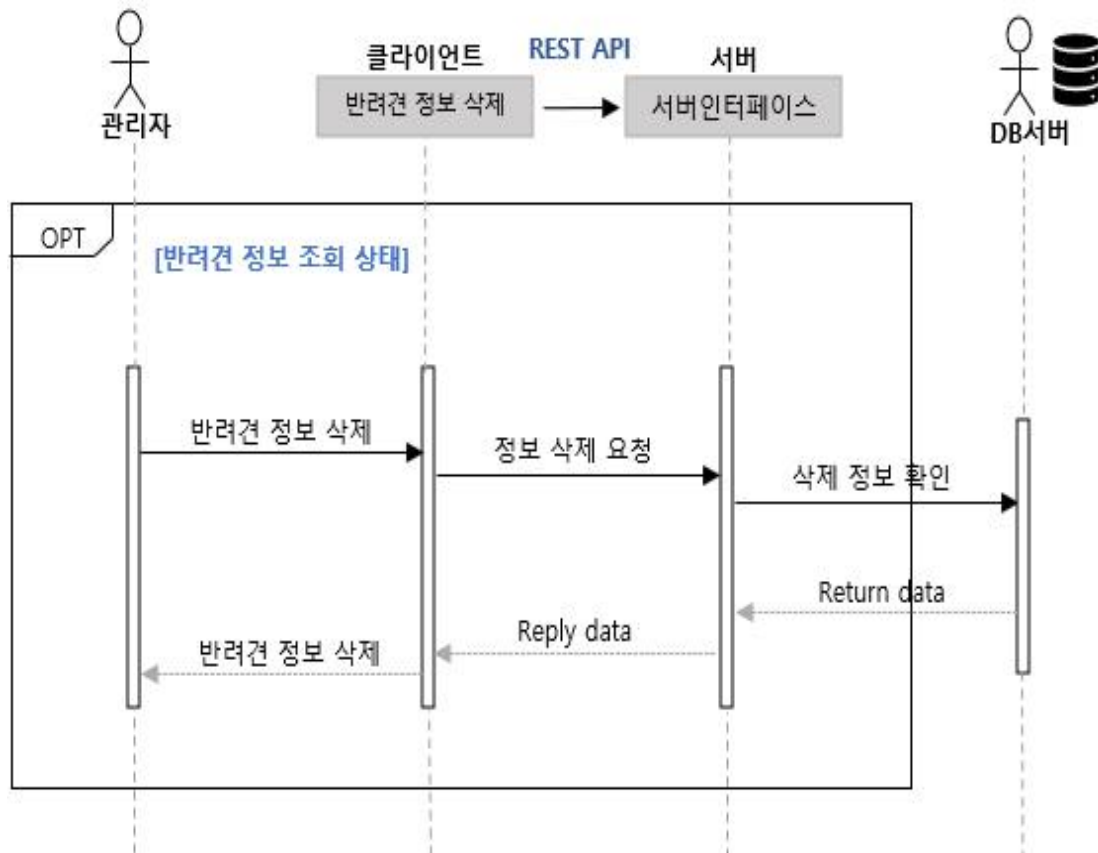


그림 3.3.10 반려견 정보 삭제의 Sequence Diagram

그림 3.3.10을 보면 관리자는 서버에게 조회한 반려견 정보 삭제를 요청한다. 서버는 데이터베이스의 반려견 테이블에서 요청 정보를 확인한 후 데이터를 삭제한다. 서버는 사용자에게 삭제 정보를 메시지로 전달한다.

## 11) 질병 관리 - 질병 정보 조회 기능

관리자는 전체적인 질병 정보를 조회할 수 있다. 질병 정보 조회기능을 구현하기 위해 아래와 같이 설계한다.

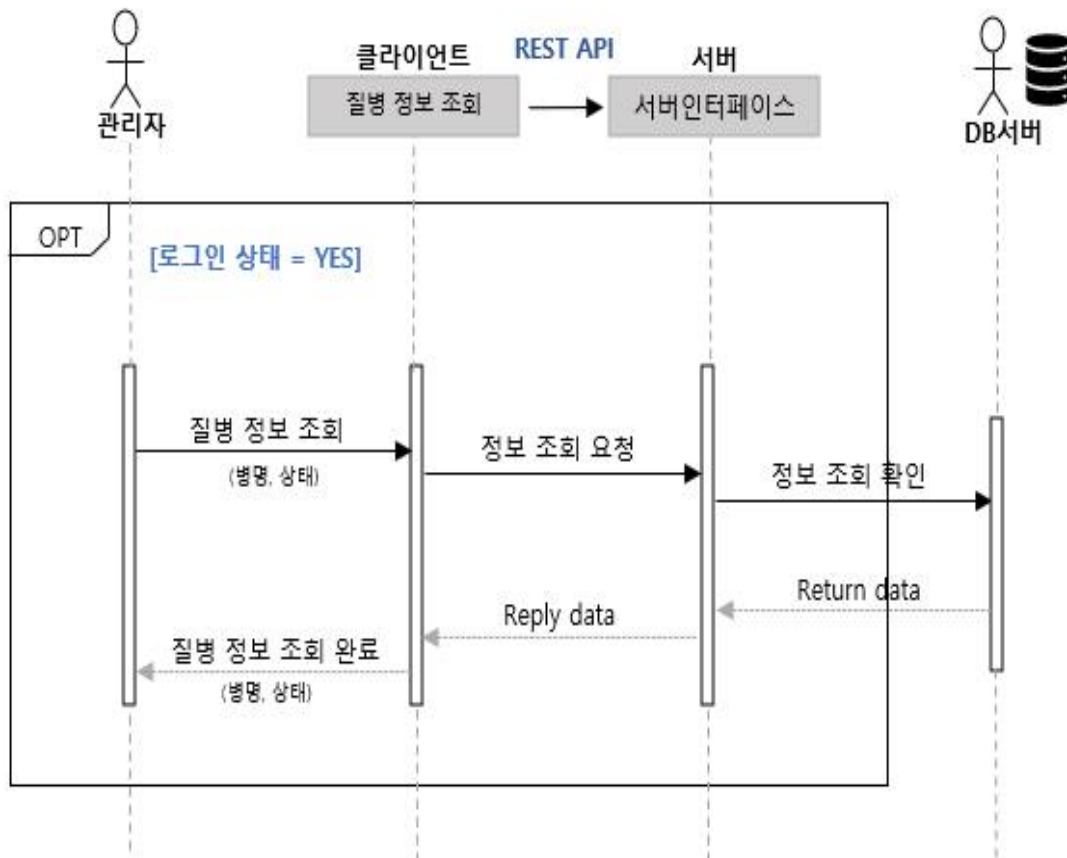


그림 3.3.11 질병 정보 조회의 Sequence Diagram

그림 3.3.11을 보면 관리자는 서버에게 질병 정보를 요청한다. 서버는 데이터베이스의 질병 테이블에서 사용자가 요청한 정보를 확인한 후 데이터를 전달해 준다.

## 12) 질병 관리, 시각화 기능 - 진단 결과 조회 및 질병 진단 정보 시각화

관리자는 각 사용자의 반려견의 진단 결과를 시각화된 자료와 함께 조회할 수 있다. 질병 진단 결과 조회 기능을 구현하기 위해 아래와 같이 설계한다.

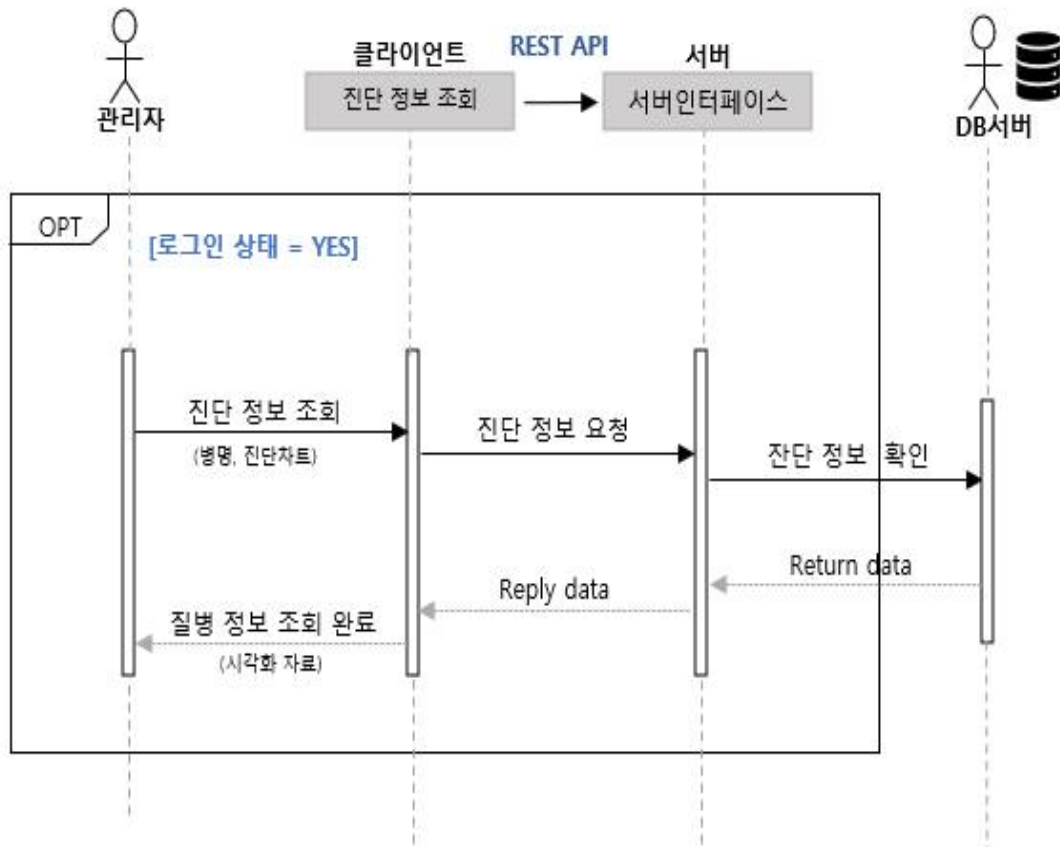


그림 3.3.12 진단 결과 조회 및 질병 진단 정보 시각화의 Sequence Diagram

그림 3.3.12를 보면 관리자는 반려견의 진단 정보를 서버에게 요청하고 있다. 서버는 데이터베이스의 진단 테이블에서 관리자가 요청한 정보를 확인하여 텍스트 자료와 시각화 자료 데이터를 사용자에게 전달해 준다.

## 20) 병원 관리 - 예약 조회 기능

관리자는 사용자가 등록한 예약 정보를 조회할 수 있다. 예약 조회 기능을 구현하기 위해 아래와 같이 설계한다.

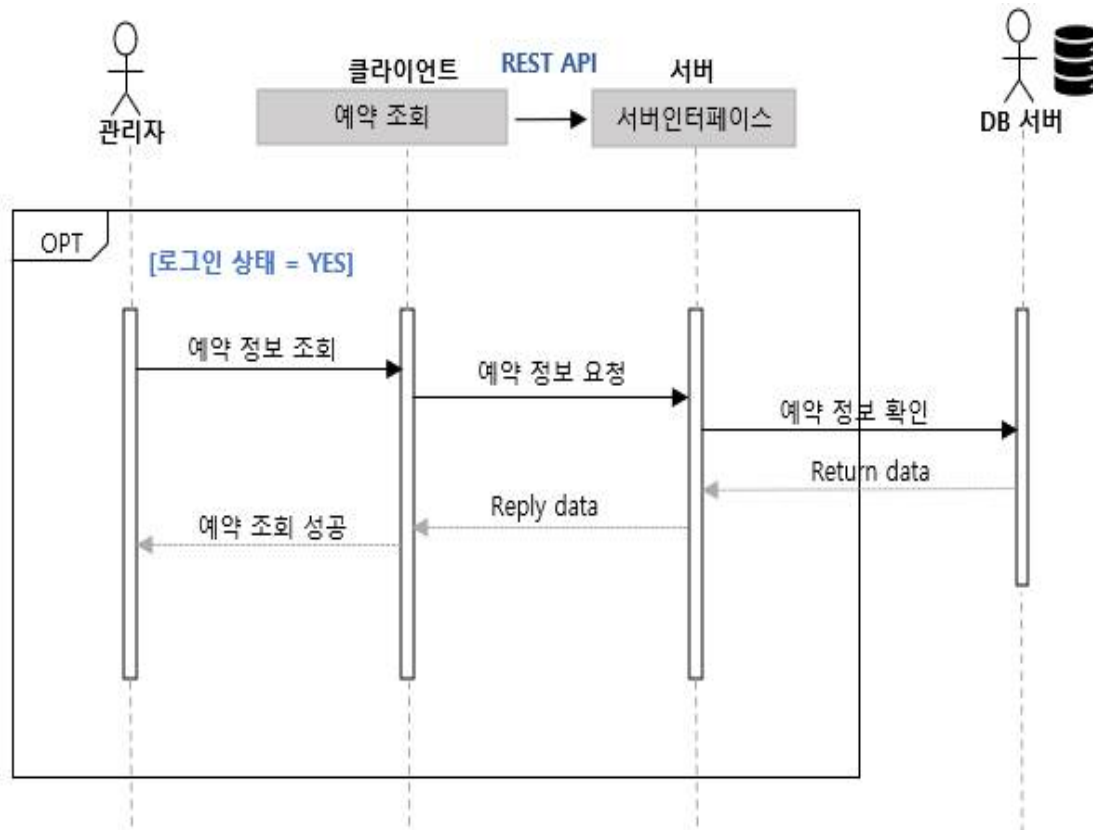


그림 3.3.20 예약 조회의 Sequence Diagram

그림 3.3.20을 보면 관리자는 사용자가 등록한 예약 정보를 서버에 요청한다. 서버는 데이터베이스의 예약 정보 테이블에서 요청한 정보를 확인한 후 관리자에게 데이터를 전송하여 정보를 제공한다. 서버는 예약 정보를 사용자에게 메시지로 전송한다.

## 21) 병원 관리 - 예약 수정 기능

관리자는 사용자가 등록한 예약 정보를 수정할 수 있다. 예약 수정 기능을 구현하기 위해 아래와 같이 설계한다.

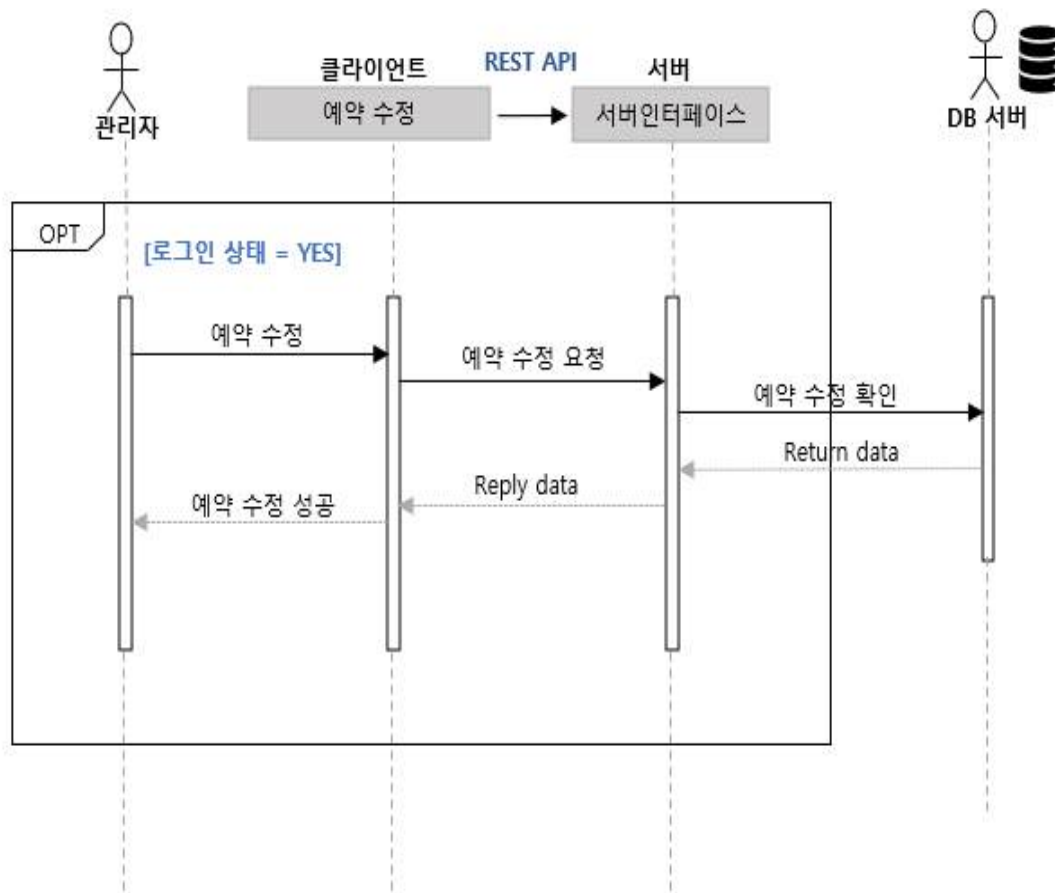


그림 3.3.21 예약 수정의 Sequence Diagram

그림 3.3.21을 보면 관리자는 서버에게 수정한 예약 정보 등록을 요청한다. 서버는 데이터베이스의 예약 테이블을 확인한 후 관리자가 수정한 예약 정보로 수정하여 저장한 후 관리자에게 메시지를 통해 정보를 전달한다.

## 22) 병원 관리 - 예약 취소 기능

관리자는 사용자가 등록한 예약 정보를 삭제할 수 있다. 예약 취소 기능을 구현하기 위해 아래와 같이 설계한다.

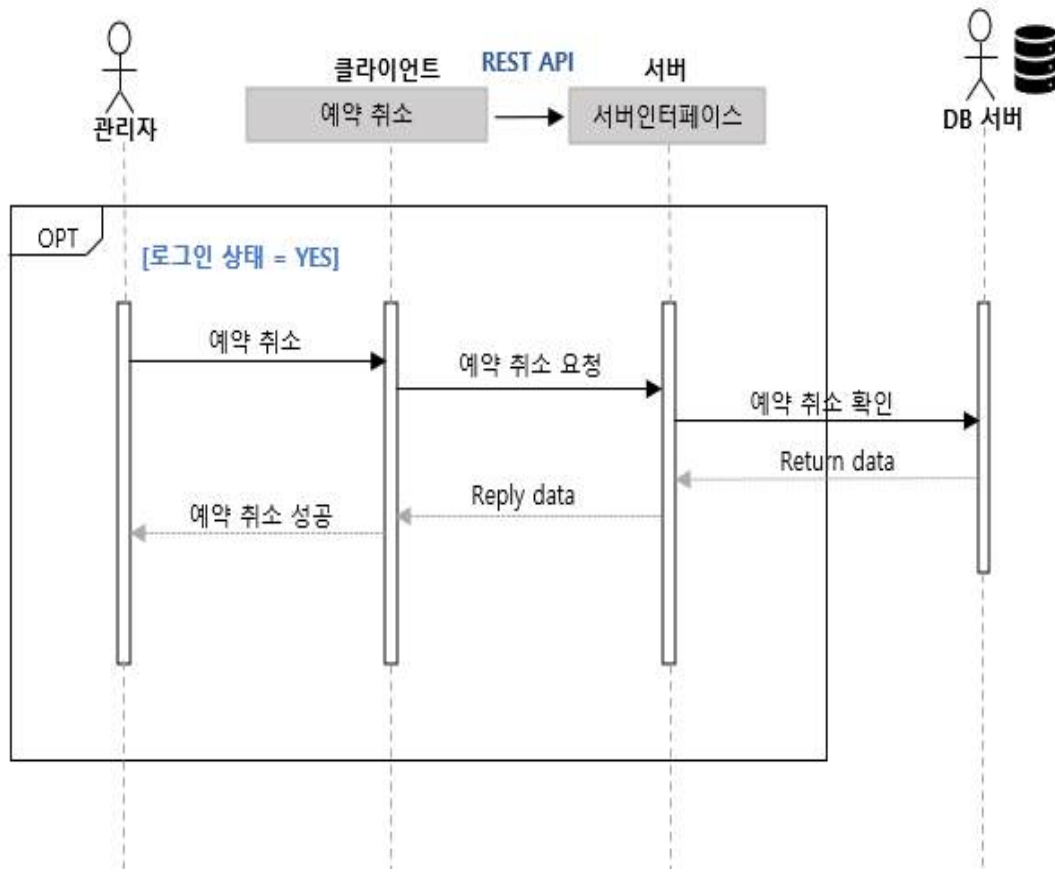


그림 3.3.22 예약 취소의 Sequence Diagram

그림 3.3.22를 보면 관리자는 서버에 등록된 예약 정보의 삭제를 요청한다. 서버는 데이터베이스의 예약 테이블에서 관리자가 요청한 정보를 삭제한다. 관리자에게 메시지를 통해 삭제 정보를 전달한다.



## 4. 화면(UI) 설계

### 4.1. 스토리보드(메뉴) 구성

#### ■ 사용자

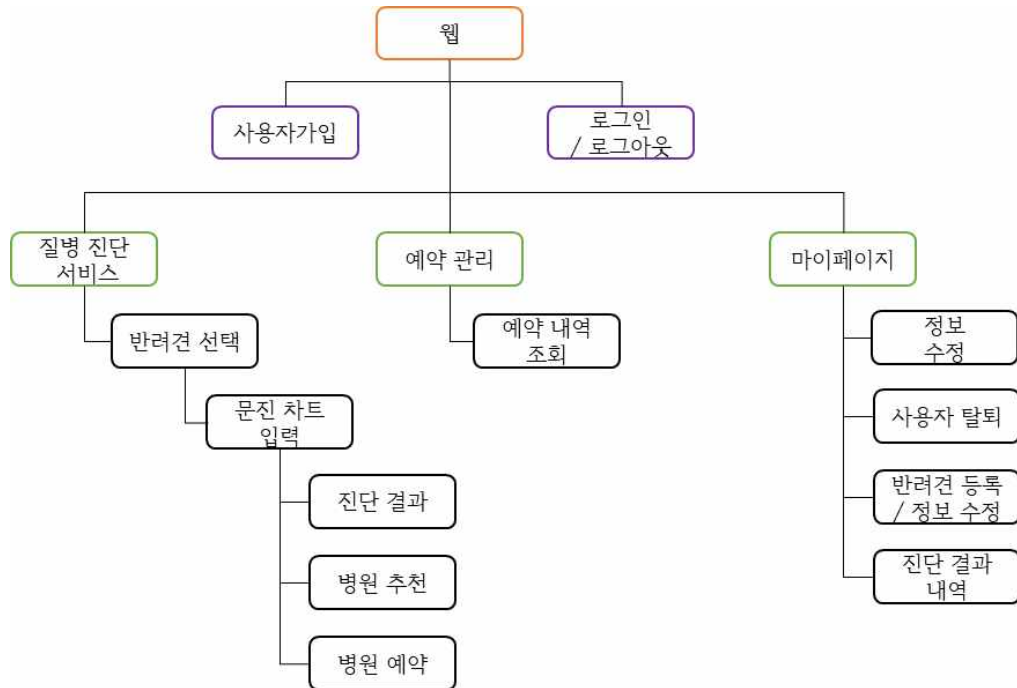


그림 4.1 사용자용 메뉴구조도

#### ■ 관리자

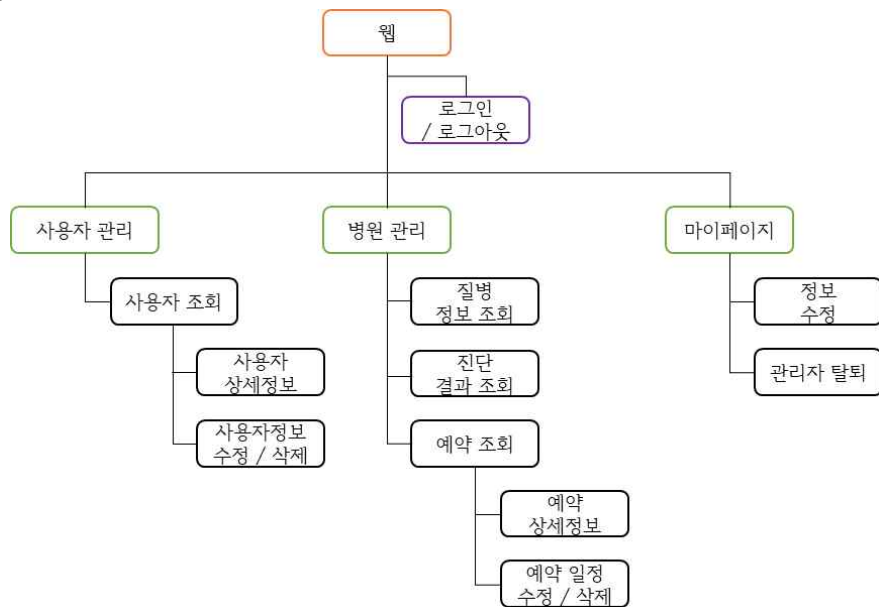
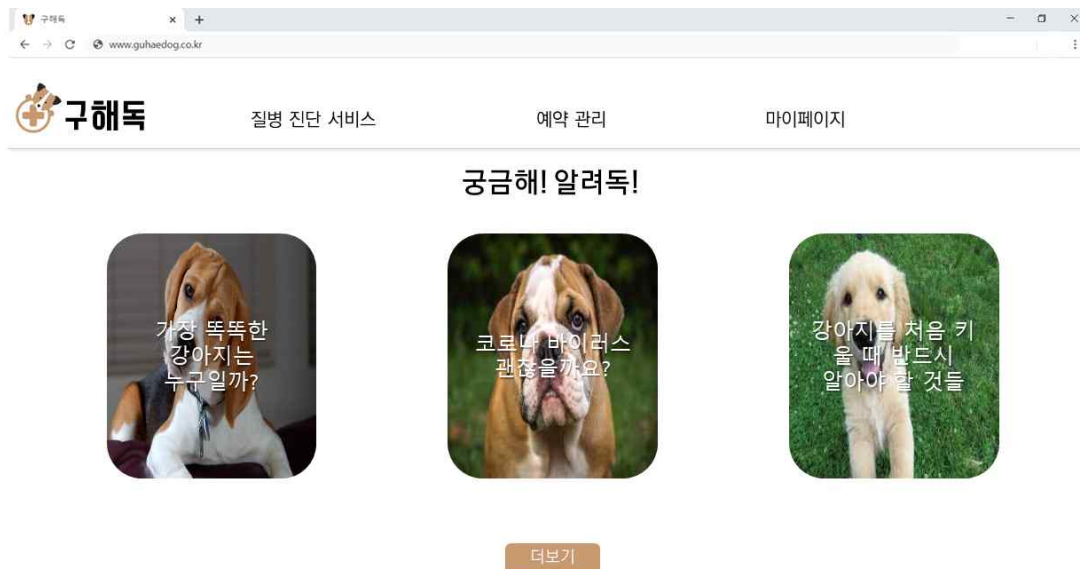
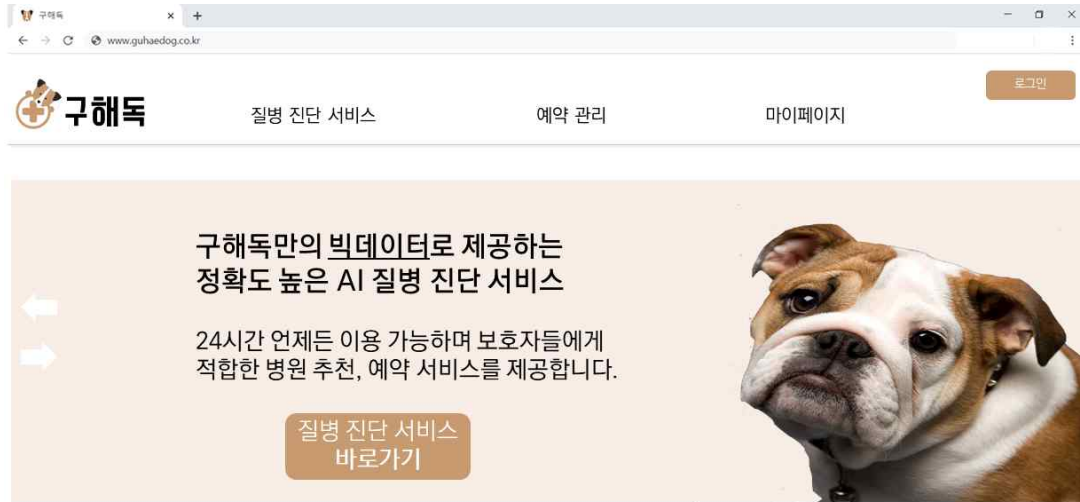


그림 4.2 관리자용 메뉴 구조도

## 4.2. 초기화면(로그인 전화면) 설계



- 웹을 접속하면 가장 먼저 보이는 메인 페이지이다.
- 네비게이션 바에서 '질병 진단 서비스', '예약 관리', '마이페이지' 기능을 선택할 수 있다.
- 오른쪽 상단의 '로그인' 버튼을 통해 사용자 인증 및 사용자 가입이 가능하다.
- 네비게이션 바 밑의 메인 페이지의 캐러셀 콘텐츠를 통해 '구해독 서비스'에 대한 정보를 제공한다.
- 메인 페이지의 하단 콘텐츠를 통해 사용자들에게 필요한 정보들을 제공하고 '더보기' 버튼을 누르면 더 많은 정보를 볼 수 있다.

### 4.3. 사용자UI 설계

#### ■ 서비스 로그인 화면 - 화면번호 A01

구해독

질병 진단 서비스      예약 관리      마이페이지

### 사용자 로그인

이메일 주소

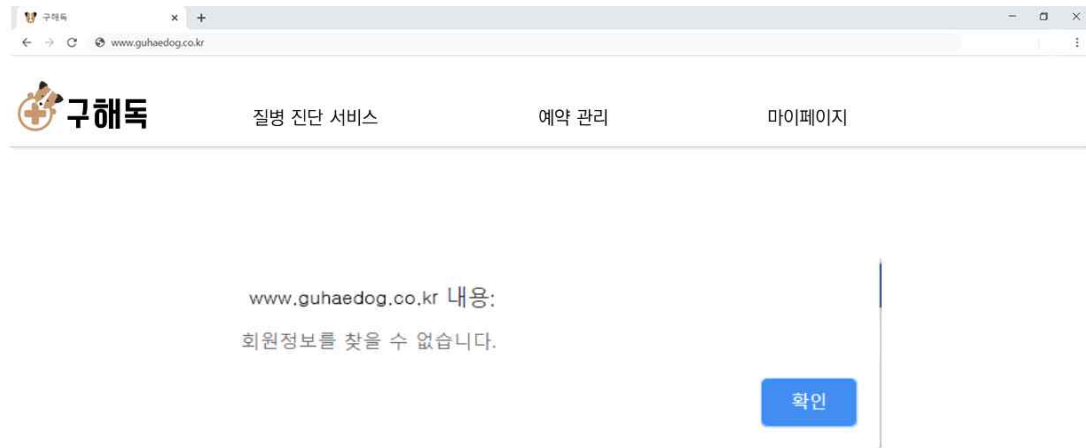
비밀번호

로그인

사용자가입

- '로그인' 버튼을 누르면 나타나는 화면이다.
- '사용자 가입' 버튼을 누르면 서비스 가입 화면으로 이동한다.
- '이메일'과 '비밀번호'를 입력하고 '로그인' 버튼을 누르면 메인 화면으로 이동한다.

■ 로그인 실패 화면 - 화면번호 A02



- 로그인이 실패했을 때 나타나는 화면이다.
- '확인' 버튼을 누르면 다시 로그인 화면(화면번호 A01)으로 이동한다.

## ■ 서비스 사용자 가입 화면 - 화면번호 A03

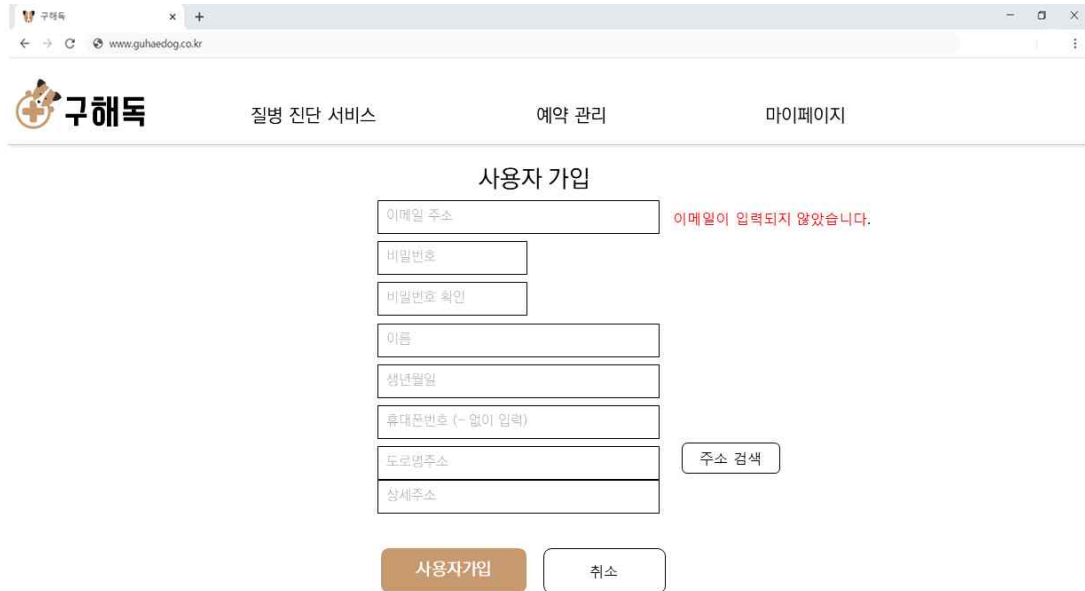
The screenshot shows a web browser window with the URL www.guhaedog.co.kr. The page title is '구해독' (Guhaedog). The navigation bar includes '질병 진단 서비스' (Disease Diagnosis Service), '예약 관리' (Reservation Management), and '마이페이지' (My Page). The main content area is titled '사용자 가입' (User Registration). The form contains the following fields:

- 이메일 주소 (Email Address)
- 비밀번호 (Password)
- 비밀번호 확인 (Confirm Password)
- 이름 (Name)
- 생년월일 (Date of Birth)
- 휴대폰번호 (-없이 입력) (Mobile Phone Number)
- 도로명주소 (Road Name Address)
- 상세주소 (Detailed Address)

There is a '주소 검색' (Address Search) button next to the address fields. At the bottom of the form, there are two buttons: '사용자가입' (Join User) and '취소' (Cancel).

- 신규 사용자를 위한 서비스 사용자 가입 화면이다.
- 사용자 가입을 위해 텍스트 박스에 '이메일 주소', '비밀번호', '이름', '생년월일', '휴대폰 번호', '주소'를 입력받는다.
- '주소 검색'을 누르면 주소를 찾아 입력할 수 있으며, 카카오 주소 API를 이용한다.
- '사용자 가입' 버튼을 누르면 사용자 가입이 완료되고, '취소' 버튼을 누르면 로그인 화면으로 이동한다.

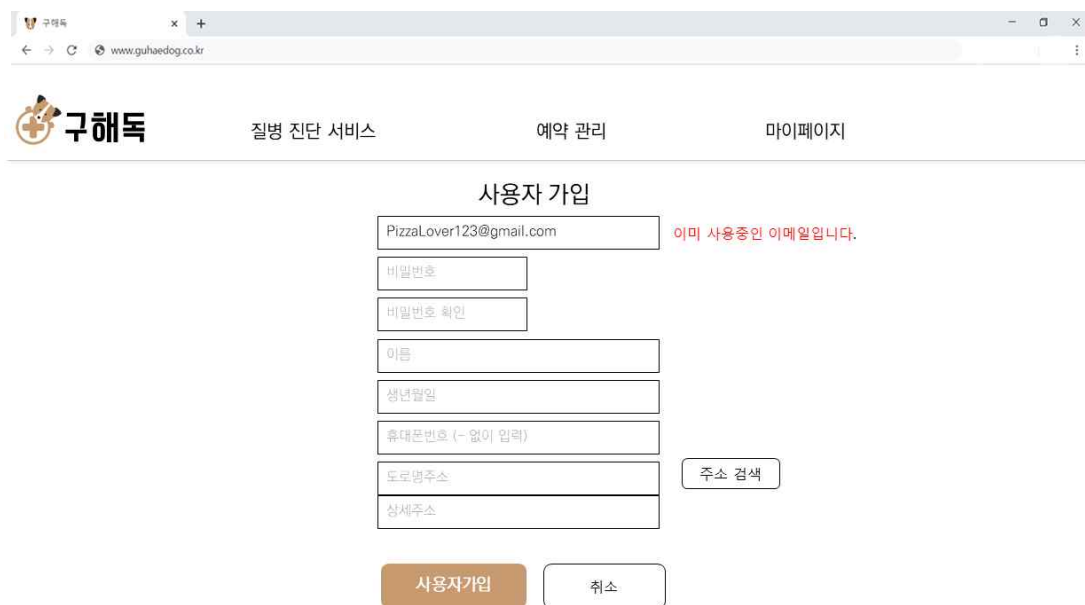
■ 서비스 사용자 가입 오류 화면(이메일 텍스트 창이 공란일 때) - 화면번호 A04



The screenshot shows a web browser window with the URL www.guhaedog.co.kr. The page has a header with the logo '구해독' and navigation links: '질병 진단 서비스', '예약 관리', and '마이페이지'. The main content area is titled '사용자 가입' (User Registration). It contains several input fields: '이메일 주소' (Email Address), '비밀번호' (Password), '비밀번호 확인' (Confirm Password), '이름' (Name), '생년월일' (Date of Birth), '휴대폰번호 (-없이 입력)' (Mobile Number), '도로명주소' (Road Name Address), and '상세주소' (Detailed Address). A red error message '이메일이 입력되지 않았습니다.' is displayed next to the empty email field. There is a '주소 검색' (Address Search) button next to the address fields. At the bottom, there are two buttons: '사용자가입' (Register User) and '취소' (Cancel).

- '이메일 주소' 텍스트 창이 공란이면 다음과 같이 경고 문구가 나타난다.

■ 서비스 사용자 가입 오류 화면(아이디 중복 확인) - 화면번호 A05



The screenshot shows the same web browser window as the previous one. The '이메일 주소' field now contains the text 'PizzaLover123@gmail.com'. A red error message '이미 사용중인 이메일입니다.' is displayed next to the field. The rest of the form and buttons are the same as in the previous screenshot.

- 이미 사용 중인 이메일이라면 다음과 같이 경고 문구가 나타난다.

■ 서비스 사용자 가입 오류 화면(아이디 중복 확인) - 화면번호 A06

The screenshot shows a web browser window with the URL www.guhaedog.co.kr. The page has a header with the logo '구해독' and navigation links: '질병 진단 서비스', '예약 관리', and '마이페이지'. The main content area is titled '사용자 가입' (User Registration). It contains a form with the following fields: '이메일' (Email) with the value 'DogLover123@gmail.com', '비밀번호' (Password), '비밀번호 확인' (Confirm Password), '이름' (Name), '생년월일' (Date of Birth), '휴대폰번호 (-없이 입력)' (Mobile Number), '도로명주소' (Road Name Address), and '상세주소' (Detailed Address). A '주소 검색' (Address Search) button is next to the address fields. Below the form are two buttons: '사용자가입' (Register User) and '취소' (Cancel). A red error message '사용 가능한 이메일입니다.' (This is an email that can be used) is displayed next to the email field.

- 사용하지 않은 아이디라면 다음과 같이 문구가 나타난다.

■ 서비스 사용자 가입 오류 화면(비밀번호 불일치) - 화면번호 A07

The screenshot shows the same web browser window as the previous one, but with a different error message. The '비밀번호' (Password) and '비밀번호 확인' (Confirm Password) fields are now filled with asterisks. A red error message '비밀번호가 일치하지 않습니다.' (Passwords do not match) is displayed next to the confirm password field. The rest of the form and buttons are the same as in the previous screenshot.

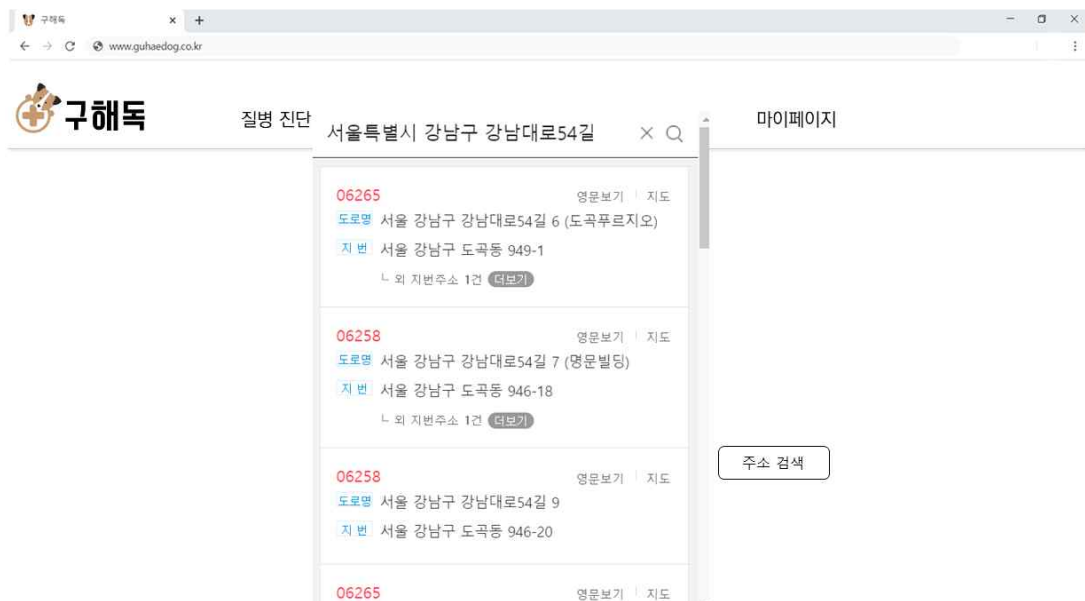
- 비밀번호와 비밀번호 확인이 불일치하면 다음과 같이 경고 문구가 나타난다.

## ■ 서비스 사용자 가입 화면(주소 검색 API) - 화면번호 A08



- '주소 검색' 버튼을 누르면 카카오 주소 검색 API를 통해 주소를 찾을 수 있다.

## ■ 서비스 사용자 가입 화면(주소 검색 API) - 화면번호 A09



- 사용자의 주소를 입력한 후 해당하는 주소를 클릭하면 주소가 입력된다.



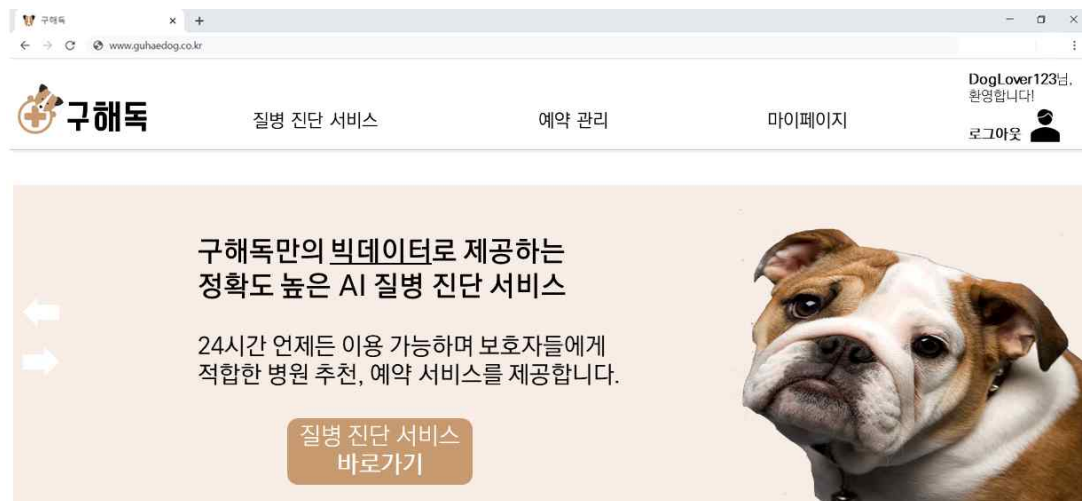
■ 서비스 사용자 가입 오류 화면(텍스트 창이 하나라도 공란일 때) - 화면번호 A10

The screenshot shows a web browser window with the URL www.guhaedog.co.kr. The page has a header with the logo '구해독' and navigation links: '질병 진단 서비스', '예약 관리', and '마이페이지'. The main content area is titled '사용자 가입' (User Registration). It contains a form with the following fields and elements:

- Email field: DogLover123@gmail.com
- Warning message: 'www.guhaedog.co.kr 내용: 입력되지 않은 정보가 존재합니다.' (Content of www.guhaedog.co.kr: There is information that has not been entered.)
- Confirm button: '확인' (Confirm)
- Phone number field: '휴대폰번호 (-없이 입력)' (Mobile number (-no input))
- Address fields: '도로명주소' (Road name address) and '상세주소' (Detailed address)
- Search button: '주소 검색' (Search address)
- Registration button: '사용자가입' (User registration)
- Cancel button: '취소' (Cancel)

- 입력되지 않은 텍스트 창이 하나라도 존재할 때 '사용자 가입' 버튼을 누르면 다음과 같이 경고 문구가 나타난다.

■ 서비스 로그인 완료 화면 - 화면번호 A11



- 로그인 후 메인 화면으로, '로그인' 버튼이 사라지고 해당하는 사용자에게 대한 환영 메시지와 '로그아웃'이 생긴다.
- '로그아웃'을 누르면 로그아웃 된다.

## ■ 마이페이지 화면(정보수정) - 화면번호 A12

The screenshot shows a web browser window with the URL 'www.guhaedog.co.kr'. The page header includes the '구해독' logo and navigation links: '질병 진단 서비스', '예약 관리', and '마이페이지'. The '마이페이지' link is highlighted in orange. Below the header, there is a sub-header with '정보 수정' (Edit Information), '사용자 탈퇴' (Logout), '반려견 등록 / 정보 수정', and '진단 결과 내역'. The '정보 수정' (Edit Information) button is highlighted in orange. The main content area shows a form for editing user information. The form includes fields for email (DogLover123@gmail.com), password (masked with asterisks), name (홍길동), phone number (960523), and address (서울 강남구 강남대로54길 7 (영문빌딩) 506동 1202호). There is a '주소 검색' (Search Address) button next to the address field. At the bottom, there are two buttons: '수정완료' (Save Changes) and '취소' (Cancel).

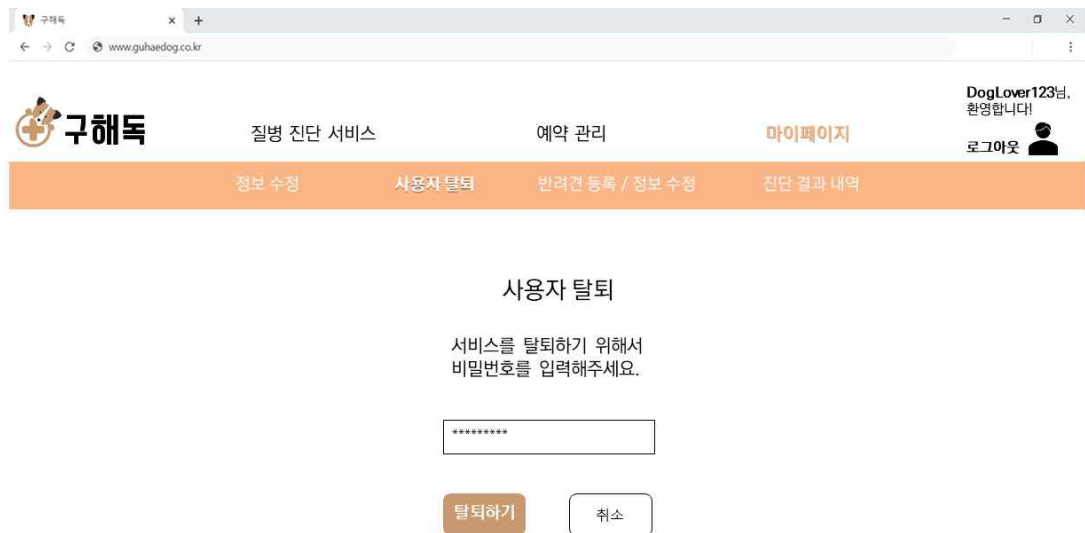
- ‘마이페이지’ - ‘정보 수정’을 눌렀을 때의 페이지 화면이다.
- ‘마이페이지’의 텍스트의 색이 수정되고 위와 같이 ‘마이페이지’에 해당하는 메뉴들이 펼쳐지고 ‘정보 수정’의 텍스트 색이 수정된다.
- 사용자는 해당 페이지에서 자신의 정보를 수정할 수 있다.
- 이메일, 이름은 수정할 수 없다.
- ‘수정 완료’ 버튼을 누르면 정보가 수정되고, 취소를 누르면 원상복귀 된다.

■ 마이페이지 화면(정보수정 : 수정 완료) - 화면번호 A13



- 수정이 완료되면 다음과 같은 문구가 나타난다.
- '확인' 버튼을 누르면 화면번호 A11로 이동한다.

■ 마이페이지 화면(사용자 탈퇴) - 화면번호 A14



- '마이페이지' - '사용자 탈퇴'를 눌렀을 때의 페이지 화면이다.
- 해당 페이지에서 사용자는 서비스를 탈퇴할 수 있다.
- 가운데에 박스에 사용자 계정 비밀번호를 입력한다.
- '탈퇴하기' 버튼을 누르면 탈퇴가 완료되고, '취소' 버튼을 누르면 탈퇴가 취소한다.

■ 마이페이지 화면(사용자 탈퇴 오류) - 화면번호 A15



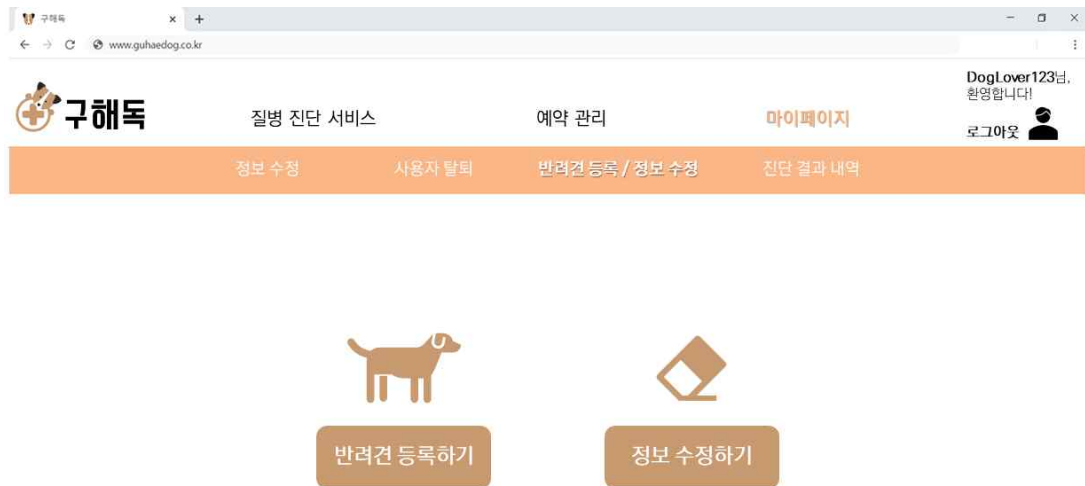
- 입력한 비밀번호가 사용자 계정의 비밀번호와 일치하지 않으면 다음과 같은 경고 문구가 나타난다.
- '확인' 버튼을 누르면 다시 화면번호 A14로 돌아간다.

■ 마이페이지 화면(사용자 탈퇴 완료) - 화면번호 A16



- 입력한 비밀번호가 사용자 계정의 비밀번호와 일치하면 다음과 같은 문구가 나타난다.
- '확인' 버튼을 누르면 로그인 전 초기화면으로 돌아간다.

■ 마이페이지 화면(등록 / 정보수정 선택화면) - 화면번호 A17



- 마이페이지' - '반려견 등록 / 정보수정'을 눌렀을 때의 페이지 화면이다.
- '반려견 등록하기' 버튼을 누르면 반려견 등록을 위한 페이지로 이동한다.
- '정보 수정하기' 버튼을 누르면 등록된 반려견의 정보를 수정하기 전 반려견 정보를 간단히 확인할 수 있는 페이지로 이동한다.

■ 마이페이지 화면(반려견 등록) - 화면번호 A18

- 화면번호 A17에서 ‘반려견 등록하기’ 버튼을 눌렀을 때의 페이지 화면이다.
- 반려견 정보를 등록하기 위해서 보호자로부터 ‘식별번호’, ‘이름’, ‘나이’, ‘생일’, ‘건종’, ‘성별’ 정보를 입력받는다.
- ‘등록 완료’ 버튼을 누르면 보호자의 반려견 정보가 등록되고, ‘취소’ 버튼을 누르면 이전 페이지로 이동한다.

■ 마이페이지 화면(반려견 등록 완료) - 화면번호 A19

- 화면번호 A18에서 ‘반려견 등록하기’ 버튼을 눌렀을 때의 페이지 화면이다.
- 등록이 완료되면 다음과 같은 문구가 나오며 ‘확인’ 버튼을 누르면 화면번호 A19로 이동한다.

■ 마이페이지 화면(반려견 정보 조회) - 화면번호 A20



반려견 정보 조회

견종(이름)	나이	생년월일	성별	식별번호	
말티즈 (별이)	4	160421	암컷	410200013112907	<button>수정</button>

- 화면번호 A17에서 '정보 수정하기' 버튼을 눌렀을 때의 페이지 화면이다.
- 수정 페이지로 이동하기 전 등록된 반려견의 견종, 이름, 나이, 생년월일, 성별, 식별번호와 같은 정보를 보여준다.
- '수정' 버튼을 누르면 반려견 정보를 수정할 수 있는 페이지로 이동한다.

■ 마이페이지 화면(반려견 정보 수정) - 화면번호 A21

- 화면번호 A20에서 '수정' 버튼을 눌렀을 때의 페이지 화면이다.
- '식별번호'와 '성별'은 수정할 수 없으며 나머지는 수정할 수 있다.
- '수정 완료' 버튼을 누르면 수정 사항들이 수정되고, '등록 삭제' 버튼을 누르면 등록됐던 반려견 정보가 모두 삭제된다.

■ 마이페이지 화면(반려견 정보 수정 완료) - 화면번호 A22

- 화면번호 A21에서 '수정 완료' 버튼을 눌렀을 때의 페이지 화면이다.
- 수정이 완료되면 다음과 같은 문구가 나오며 '확인' 버튼을 누르면 화면번호 A17로 이동한다.



## ■ 마이페이지 화면(반려견 등록 삭제 완료) - 화면번호 A23



- 화면번호 A21에서 '등록 삭제' 버튼을 눌렀을 때의 페이지 화면이다.
- 수정이 완료되면 다음과 같은 문구가 나오며 '확인' 버튼을 누르면 화면번호 A17로 이동한다.

## ■ 마이페이지 화면(진단 결과 내역) - 화면번호 A24



## 진단 결과 내역

	반려견 이름	진단 날짜	견종	나이	증상	진단 결과
3	별이	2020.04.22	말티즈	4	기침, 재채기, 눈물과다	감기
2	별이	2020.04.16	말티즈	4	기침, 재채기, 콧물	감기
1	별이	2020.03.22	말티즈	4	소화불량, 구토, 식욕저하	위장염

- '마이페이지' - '진단 결과 내역'을 눌렀을 때의 페이지 화면이다.
- 보호자의 질병 진단 서비스를 이용하고 난 후 결과 내역을 확인할 수 있다.
- '반려견 이름'을 선택하면 상세정보 페이지로 이동한다.

■ 마이페이지 화면(진단 결과 내역 : 상세정보) - 화면번호 A25



- 화면번호 A24에서 '반려견 이름'을 클릭했을 때의 페이지 화면이다.
- 기존의 정보와 더불어 질병 진단 서비스를 통해 얻게 된 결과를 시각화해서 보여준다.

■ 질병 진단 서비스 화면(반려견 선택) - 화면번호 A26

The screenshot shows the '질병 진단 서비스' (Disease Diagnosis Service) interface. At the top, there's a navigation bar with '구해독' logo, '질병 진단 서비스', '예약 관리', and '마이페이지'. The main content area has a message: '반갑습니다! 홍길동님, 질병 진단 서비스가 필요한 아이가 누구인가요?' (Hello! Mr. Honggil, which child needs the disease diagnosis service?). Below this is a dropdown menu with the text '옵션 : 반려견' (Option: Pet) and a list of pet names. The name '별이' (Belli) is selected and highlighted in blue. At the bottom, there is a '다음' (Next) button.

- '질병 진단 서비스' 카테고리를 눌렀을 때 나오는 페이지 화면이다.
- 반려견 선택 박스를 누르면 사용자가 등록한 반려견의 이름이 나온다.
- '다음' 버튼을 누르면 질병 진단 서비스를 위한 다음 화면으로 넘어간다.

■ 질병 진단 서비스 화면(문진표 작성) - 화면번호 A27

**구해독** 질병 진단 서비스 예약 관리 마이페이지

<input type="checkbox"/> 감기	<input type="checkbox"/> 설사
<input type="checkbox"/> 열	<input type="checkbox"/> 소화불량
<input type="checkbox"/> 구토	<input type="checkbox"/> 탈수
<input type="checkbox"/> 소변	<input type="checkbox"/> 무기력
<input type="checkbox"/> 안구확장	<input type="checkbox"/> 식욕저하
<input type="checkbox"/> 복부팽만	<input type="checkbox"/> 황달
<input type="checkbox"/> 눈물과다	<input type="checkbox"/> 털 빠짐
<input type="checkbox"/> 눈물자국	<input type="checkbox"/> 고열
<input type="checkbox"/> 구취	<input type="checkbox"/> 발작
<input type="checkbox"/> 체중감소	<input type="checkbox"/> 기절
<input type="checkbox"/> 이상행동	<input type="checkbox"/> 절뚝거림
<input type="checkbox"/> 호흡이상	<input type="checkbox"/> 과한 침

별이는 나이 4살(성년)의 귀여운 말티즈군요!

우리 별이가 어디가 아픈지 알기 위해서 어떤 증상을 가지고 있는지 알아야 만해요!

별이의 증상을 잘 생각해보고 왼쪽의 문진표를 작성해 제출해주세요!

다음

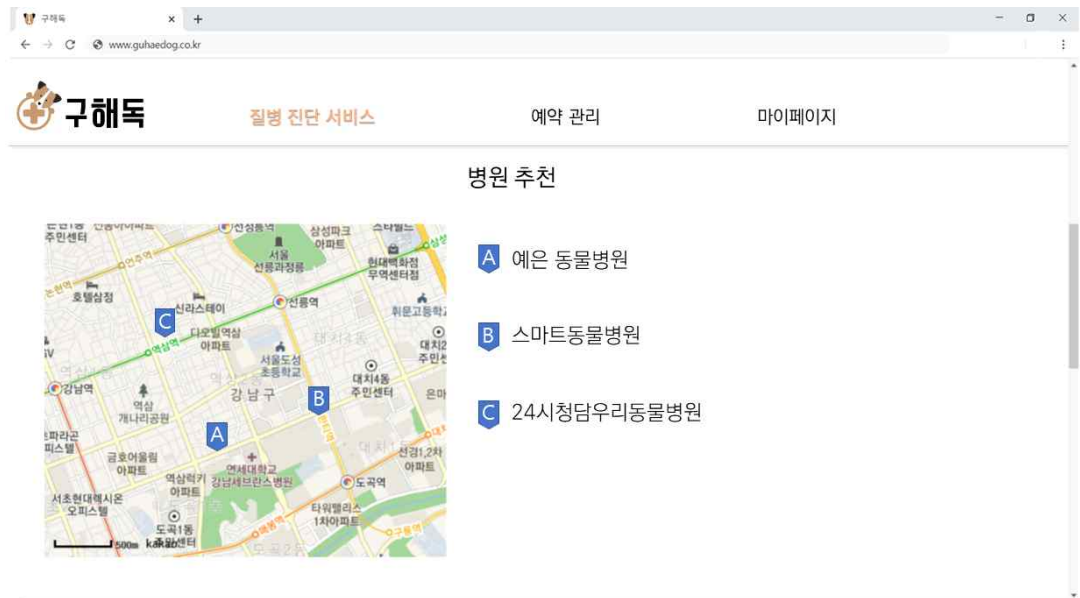
- 반려견 등록을 통해 저장된 반려견의 정보를 통해 나이와 견종 정보를 가져온다.
- 보호자는 왼쪽의 문진표를 통해서 반려견의 증상을 체크할 수 있다.
- '다음' 버튼을 누르면 진단 결과 페이지로 이동한다.

■ 질병 진단 서비스 화면(진단 결과) - 화면번호 A28



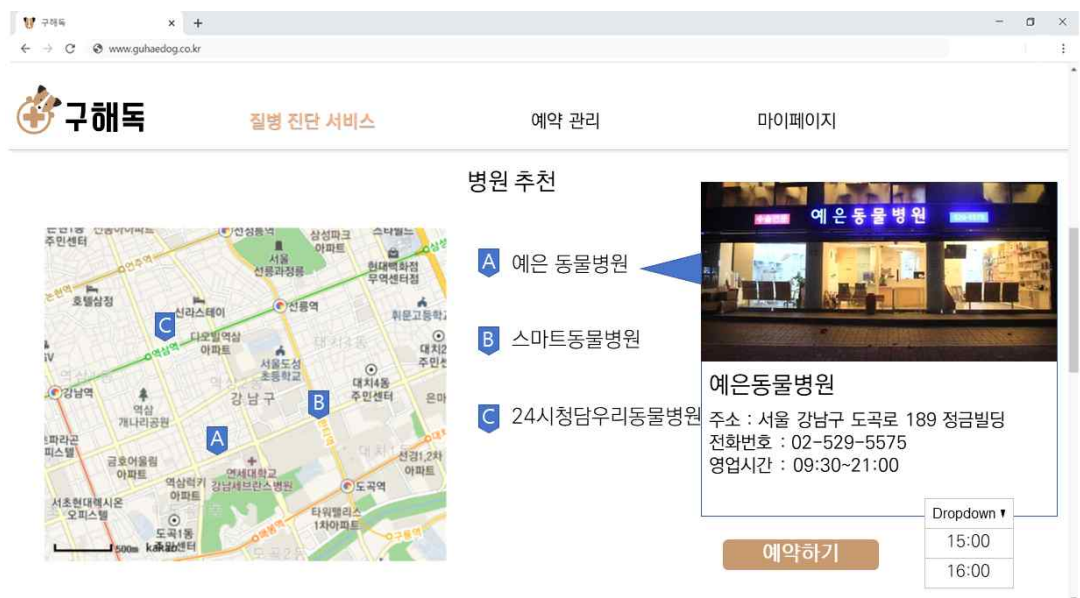
- 문진을 통하여 예상되는 질병('피부병', '감기', '구강종양')들을 보호자들이 한눈에 인식할 수 있도록 차트로 시각화한다.
- 차트 아래에는 가장 높은 질병인 '감기'에 대한 유용한 정보를 가진 기사를 제공한다.
- '기사 더보기'를 클릭하면 해당하는 기사로 이동한다.

■ 질병 진단 서비스 화면(병원 추천) - 화면번호 A29



- 사용자 가입을 통해 받아온 보호자의 주소 정보를 토대로 가까운 동물병원들을 추천한다.

■ 질병 진단 서비스 화면(병원 상세 정보 및 예약) - 화면번호 A30



- 화면번호 A29에서 추천받은 동물병원 중 하나를 선택하면 동물병원에 대한 상세정보를 보여준다.
- '예약하기' 버튼을 누르면 병원 예약 페이지로 이동한다

## ■ 질병 진단 서비스 화면(병원 예약) - 화면번호 A31

The screenshot shows a web browser window with the URL www.guhaedog.co.kr. The page has a header with the logo '구해독' and navigation links: '질병 진단 서비스', '예약 관리', and '마이페이지'. Below the header is a table with hospital information:

병원이름	영업시간	전화번호	주소
예은동물병원	09:30~21:00	02-529-5575	서울 강남구 도곡로 189 청금빌딩

Below the table is a section titled '병원 예약' (Hospital Reservation) with a form:

견종(이름)	나이	생년월일	성별	식별번호	예약 시간
말티즈 (별이)	4	160421	암컷	410200013112907	Dropdown ▼ 15:00 16:00 17:00

At the bottom of the form is an orange button labeled '예약완료' (Reservation Complete).

- 화면번호 A30에서 병원 상세정보를 보여주는 페이지에서 '예약하기' 버튼을 눌렀을 때의 페이지 화면이다.
- 병원에 대한 정보와 사용자의 반려견에 대한 정보를 확인할 수 있고 예약 시간을 정할 수 있다.
- '예약 완료' 버튼을 누르면 병원 예약이 완료된다.

## ■ 예약 관리 화면(예약 내역 조회) - 화면번호 A32

	동물병원명	예약일자(방문시간)	주소	전화번호	예약 상태
2	예은 동물병원	2020.04.22 (15:00)	서울 강남구 도곡로 189 정금빌딩	02-529-5575	예약완료 <button>수정</button> <button>취소</button>
1	스마트 동물병원	2020.04.16 (16:00)	서울 강남구 선릉로 418	02-554-7582	방문완료

- ‘예약 관리’를 눌렀을 때의 페이지 화면이다.
- 보호자가 신청한 동물 병원 예약 내역을 조회할 수 있다.
- 예약 후 진료를 위해 방문을 완료하면 예약 상태는 ‘방문 완료’ 상태가 되고, 방문 전 이고 예약을 완료하면 ‘예약 완료’ 상태이다.
- ‘수정’ 버튼을 누르면 예약을 수정할 수 있는 페이지로 이동하고 ‘취소’ 버튼을 누르면 예약이 취소된다.

## ■ 예약 관리 화면(예약 수정) - 화면번호 A33

병원이름	영업시간	전화번호	주소
예은동물병원	09:30~21:00	02-529-5575	서울 강남구 도곡로 189 정금빌딩

병원 예약 수정

견종(이름)	나이	생년월일	성별	식별번호	예약 시간
말티즈 (별이)	4	160421	암컷	410200013112907	Dropdown ▼ 15:00 16:00 17:00

예약수정

- 화면번호 A32에서 수정 버튼을 눌렀을 때의 페이지 화면이다.
- 드롭다운을 통해 예약 시간을 수정할 수 있고 ‘예약 수정’ 버튼을 누르면 완료된다.

■ 예약 관리 화면(예약 내역 조회 : 예약 수정 결과) - 화면번호 A34



	동물병원명	예약일자(방문시간)	주소	전화번호	예약 상태
2	예은 동물병원	2020.04.22 (16:00)	서울 강남구 도곡로 189 정금빌딩	02-529-5575	예약수정 수정 취소
1	스마트 동물병원	2020.04.16 (16:00)	서울 강남구 선릉로 418	02-554-7582	방문완료

- 화면번호 A33에서 수정 완료 버튼을 눌렀을 때의 결과이다.
- 예약 상태가 예약 수정으로 바뀌고 방문시간이 15:00에서 16:00로 변경된 것을 확인할 수 있다.

■ 예약 관리 화면(예약 내역 조회 : 예약 취소 시) - 화면번호 A35

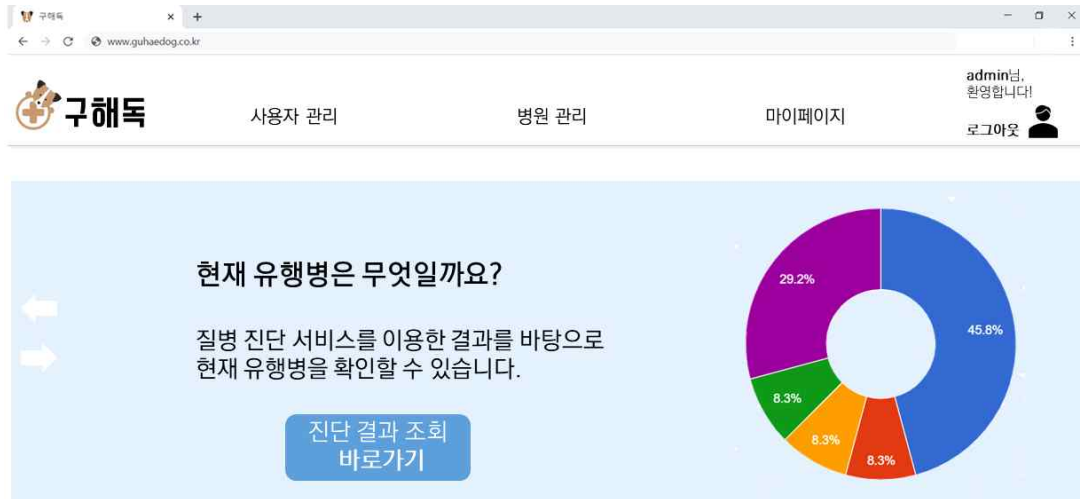


	동물병원명	예약일자(방문시간)	주소	전화번호	예약 상태
2	예은 동물병원	2020.04.22 (16:00)	서울 강남구 도곡로 189 정금빌딩	02-529-5575	예약취소
1	스마트 동물병원	2020.04.16 (16:00)	서울 강남구 선릉로 418	02-554-7582	방문완료

- 화면번호 A34에서 '취소' 버튼을 눌렀을 때의 결과이다.
- 예약 상태가 예약 수정에서 예약 취소로 바뀐 것을 확인할 수 있다.

## 4.4. 관리자 UI 설계

### ■ 서비스 로그인 완료 화면 - 화면번호 C01



- 관리자 계정으로 로그인 했을 때의 페이지 화면이다.
- 관리자와 관련된 카테고리들로 바뀌고 디자인이 수정된다.
- 캐러셀 콘텐츠도 관리자와 관련된 내용으로 바뀐다.



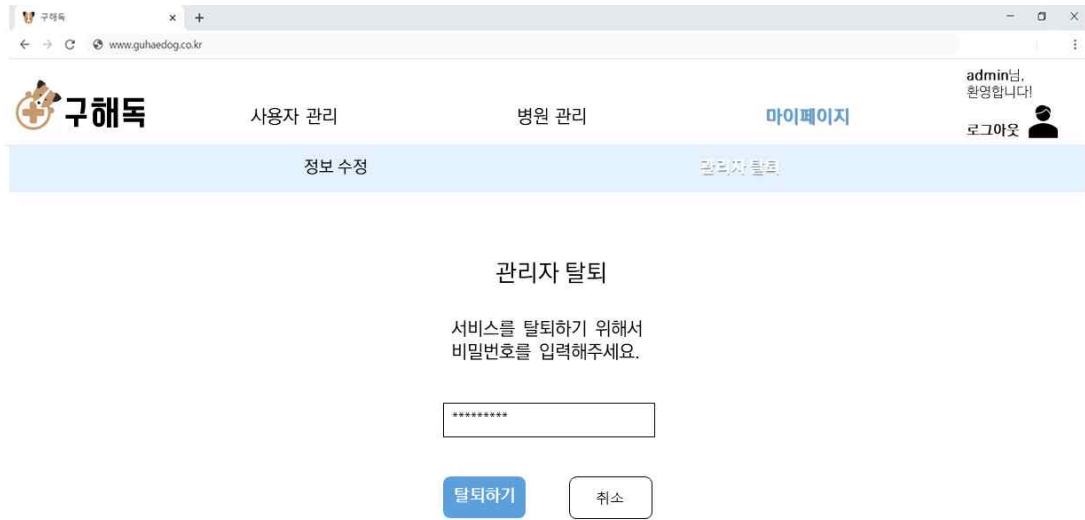
## ■ 마이페이지 화면(정보 수정) - 화면번호 C02

- ‘마이페이지’ - ‘개인정보 수정’을 눌렀을 때의 페이지 화면이다.
- ‘마이페이지’의 텍스트의 색이 수정되고 위와 같이 ‘마이페이지’에 해당하는 메뉴들이 펼쳐지고 ‘개인정보 수정’의 텍스트 색이 수정된다.
- 사용자는 해당 페이지에서 자신의 모든 정보를 수정할 수 있다.
- ‘수정 완료’ 버튼을 누르면 개인정보가 수정되고, ‘취소’를 누르면 원상복귀 된다.

## ■ 마이페이지 화면(개인정보 수정 완료) - 화면번호 C03

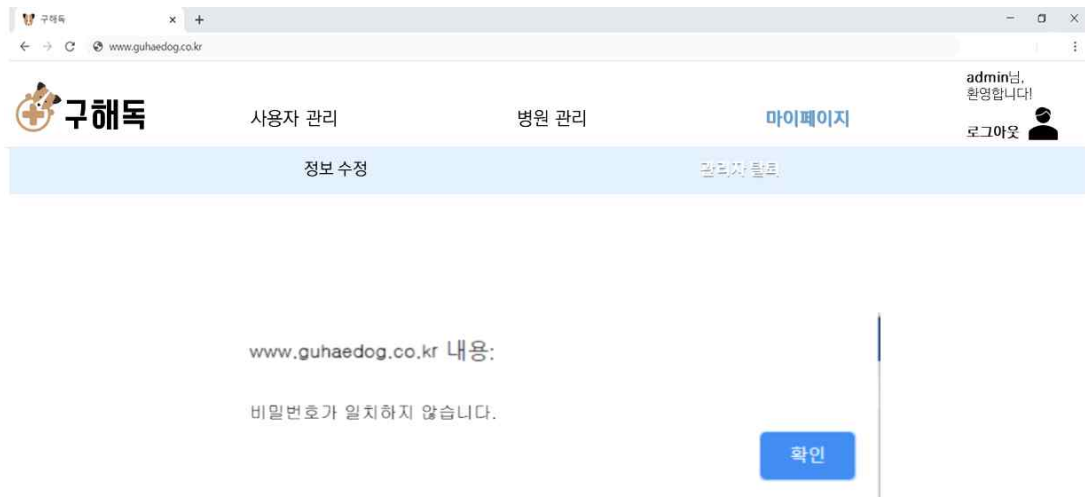
- 수정이 완료되면 다음과 같은 문구가 나타난다.
- ‘확인’ 버튼을 누르면 화면번호 C01로 이동한다.

■ 마이페이지 화면(관리자 탈퇴) - 화면번호 C04



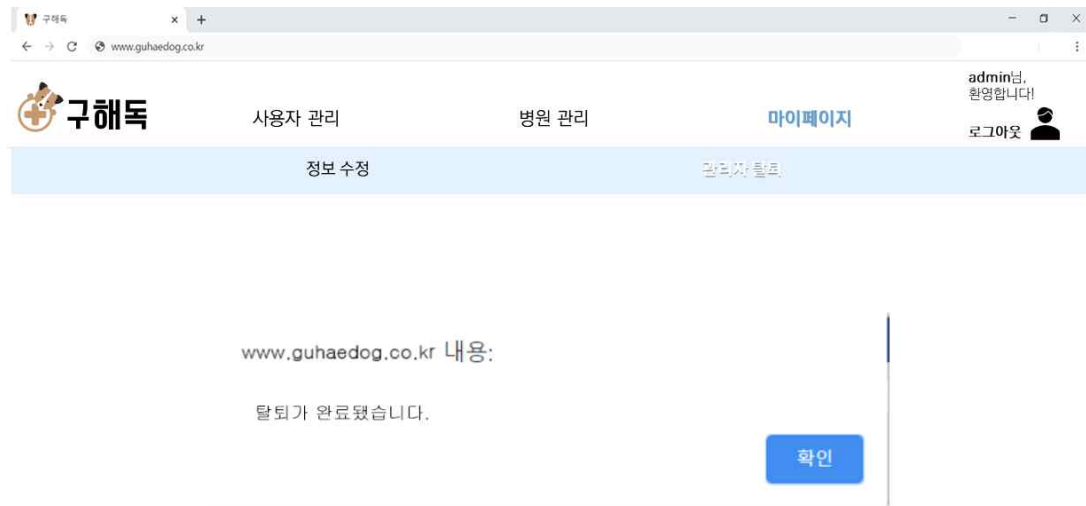
- ‘마이페이지’ - ‘사용자 탈퇴’를 눌렀을 때의 페이지 화면이다.
- 탈퇴하기’ 버튼을 누르면 탈퇴가 완료되고, ‘취소’ 버튼을 누르면 탈퇴가 취소한다.

■ 마이페이지 화면(관리자 탈퇴 오류) - 화면번호 C05



- 입력한 비밀번호가 관리자 계정의 비밀번호와 일치하지 않으면 다음과 같은 경고 문구가 나타난다.
- ‘확인’ 버튼을 누르면 다시 화면번호 C04로 돌아간다.

■ 마이페이지 화면(관리자 탈퇴 성공) - 화면번호 C06



- 입력한 비밀번호가 관리자 계정의 비밀번호와 일치하면 다음과 같은 문구가 나타난다.
- '확인' 버튼을 누르면 로그인 전 초기화면으로 돌아간다.

■ 사용자 관리 화면(사용자 조회) - 화면번호 C07



사용자 조회

	이메일(이름)	비밀번호	생년월일	주소	전화번호	
4	Doglover123@gmail.com (홍길동)	123456789	960523	서울 강남구 강남대로54길 7 (영문빌딩) 506동 1202호	010-3232-5656	<button>수정</button> <button>삭제</button>
3	kimcs456@gmail.com (김철수)	456789123	940122	서울 마포구 마포대로32길 6 (용문빌딩) 303동 902호	010-1414-2312	<button>수정</button> <button>삭제</button>
2	pengsukim@naver.com (김행수)	131313133	991010	수원시 영통구 원천대로18길 (래미안) 106동 1001호	010-3232-5136	<button>수정</button> <button>삭제</button>
1	dongja11@nate.com (옥동자)	aaa111222	861123	서울 강남구 강남대로54길 7 (영문빌딩) 504동 101호	010-2432-5656	<button>수정</button> <button>삭제</button>

- '사용자 관리'를 눌렀을 때의 사용자 조회 페이지 화면이다.
- '사용자'의 '이름'을 클릭하면 보호자가 키우고 있는 반려견과 질병 진단 서비스 결과에 대한 정보를 확인할 수 있는 페이지로 이동한다.
- '수정' 버튼을 누르면 사용자 정보를 수정할 수 있는 페이지로 이동한다.
- '삭제' 버튼을 누르면 사용자 정보가 삭제된다.

■ 사용자 관리 화면(정보 수정) - 화면번호 C08

구해독 사용자 관리 병원 관리 마이페이지 admin님, 환영합니다! 로그아웃

### 정보 수정

DogLover123@gmail.com

\*\*\*\*\*

\*\*\*\*\*

홍길동

960523

01032325656

서울 강남구 강남대로54길 7 (영문빌딩)

주소 검색

505동 1505호

수정완료

- 화면번호 C07에서 '수정' 버튼을 눌렀을 때의 페이지 화면이다.
- 사용자에게 관련된 모든 정보를 수정할 수 있다.
- '수정 완료' 버튼을 누르면 개인정보 수정이 완료된다.

■ 사용자 관리 화면(수정 완료) - 화면번호 C09

구해독 사용자 관리 병원 관리 마이페이지 admin님, 환영합니다! 로그아웃

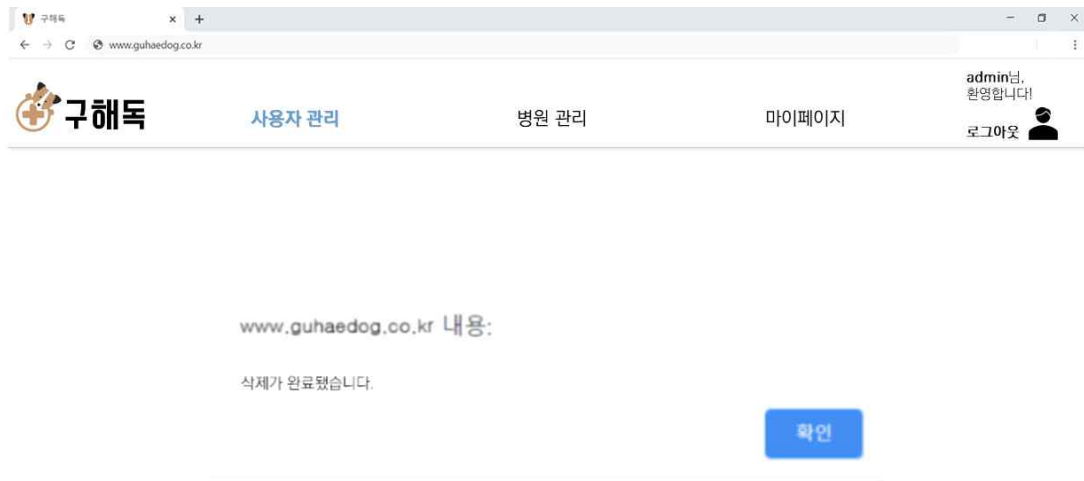
www.guhaedog.co.kr 내용:

수정이 완료되었습니다.

확인

- 화면번호 C08에서 '수정 완료' 버튼을 누르고 수정이 완료되면 다음과 같은 문구가 나타난다.

■ 사용자 관리 화면(삭제 완료) - 화면번호 C10



- 화면번호 C07에서 '삭제' 버튼을 누르고 삭제가 완료되면 다음과 같은 문구가 나타난다.

■ 사용자 관리 화면(사용자 상세정보 : 반려견 미등록 - 화면번호 C11)



- 화면번호 C07에서 '사용자'의 '이름'을 눌렀을 때의 페이지 화면이다.
- 사용자가 반려견을 등록하지 않았으므로 사용자 정보만을 알 수 있다.

■ 사용자 관리 화면(사용자 상세정보 : 반려견 등록 - 화면번호 C12)

사용자 관리

병원 관리

마이페이지

admin님,  
환영합니다!

로그아웃

사용자 상세정보

이메일(이름)	비밀번호	생년월일	주소	전화번호
4 Doglover123@gmail.com (홍길동)	123456789	960523	서울 강남구 강남대로54길 7 (영문빌딩) 505동 1505호	010-3232-5656

견종(이름)	나이	생년월일	성별	식별번호
말티즈 (별이)	4	160421	암컷	410200013112907

● 감기

● 피부병

● 구강질환

이름 : 별이  
나이 : 4살  
성별 : 암컷  
견종 : 말티즈  
증상 : 감기, 열, 구토

감기 로 예상되는 상태

수정

삭제

- 화면번호 C07에서 '보호자'의 '이름'을 눌렀을 때의 페이지 화면이다.
- 반려견을 등록했으므로 반려견의 견종, 나이, 생년월일, 성별, 식별번호와 같은 정보를 알 수 있다.
- 질병 진단 서비스를 이용했을 때의 결과를 확인할 수 있다.
- '수정' 버튼을 누르면 반려견 정보를 수정할 수 있는 페이지로 이동한다.
- '삭제' 버튼을 누르면 반려견 정보가 삭제된다.

## ■ 사용자 관리 화면(반려견 정보 수정) - 화면번호 C13

반려견 정보 수정

410200013112907

별이

4

160521

말티즈

☐ 수컷 ☐ 암컷

수정완료

- 화면번호 C12에서 '수정' 버튼을 눌렀을 때의 페이지 화면이다.
- 반려견에 관련된 모든 정보를 수정할 수 있다.
- '수정 완료' 버튼을 반려견 정보 수정이 완료된다.

## ■ 사용자 관리 화면(반려견 수정 완료) - 화면번호 C14

www.guhaedog.co.kr 내용:

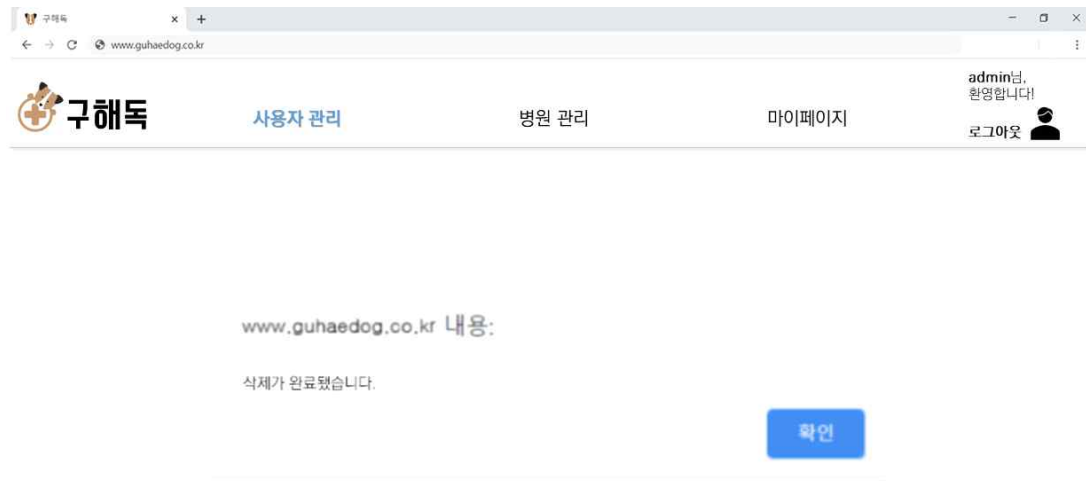
수정이 완료되었습니다.

확인

- 화면번호 C13에서 '수정 완료' 버튼을 누르고 수정이 완료되면 다음과 같은 문구가 나타난다.



■ 사용자 관리 화면(반려견 삭제 완료) - 화면번호 C15



- 화면번호 C12에서 '삭제' 버튼을 누르고 삭제가 완료되면 다음과 같은 문구가 나타난다.

## ■ 병원 관리 화면(예약 조회) - 화면번호 C16

	사용자 이름	예약일자(방문시간)	병원 주소(병원 이름)	전화번호	예약 상태
2	홍길동	2020.04.22 (16:00)	서울 강남구 강남대로54길 7 (명문빌딩) 506동 1202호 (예은동물병원)	010-3232-5656	예약수정 수정 취소 방문완료
1	이영희	2020.04.16 (16:00)	서울 강남구 강남대로54길 18-6 201호 (24시정담우리동물병원)	010-1234-5678	방문완료

- '병원 관리' - '예약 조회'를 눌렀을 때의 페이지 화면이다.
- 사용자들의 예약 내역을 조회할 수 있다.
- '사용자 이름', '예약일자(방문시간)', '병원 주소', '병원 이름', '전화번호', '예약 상태' 정보를 확인할 수 있다.
- '수정' 버튼을 누르면 예약을 수정할 수 있는 페이지로 이동하고, '취소' 버튼을 누르면 예약이 취소된다.
- '방문 완료' 버튼을 누르면 예약 상태가 방문 완료가 된다.

■ 병원 관리 화면(예약 조회 : 예약 일정 수정) - 화면번호 C17

	사용자 이름	예약일자(방문시간)	병원 주소(병원 이름)	전화번호	예약 상태
2	홍길동	2020.04.22 Dropdown ▼ 16:00 17:00 18:00	서울 강남구 강남대로54길 7 (명문빌딩) 506동 1202호 (예은동물병원)	010-3232-5656	예약수정 수정 취소

- 화면번호 C16에서 '수정' 버튼을 눌렀을 때의 페이지 화면이다.
- 관리자는 이 페이지에서 예약 일정을 수정할 수 있다.
- 텍스트 박스와 드롭다운을 통해 예약 날짜, 시간을 수정할 수 있다.
- '수정' 버튼을 누르면 수정이 완료되고 '취소' 버튼을 누르면 화면번호 B16으로 이동한다.

■ 병원 관리 화면(예약 조회 : 예약 일정 수정 완료) - 화면번호 C18

www.guhaedog.co.kr 내용:  
수정이 완료되었습니다.

확인

- 화면번호 C17에서 '수정' 버튼을 누르고 수정이 완료되면 다음과 같은 문구가 나타난다.

■ 병원 관리 화면(예약 조회 : 예약 일정 수정 결과) - 화면번호 C19

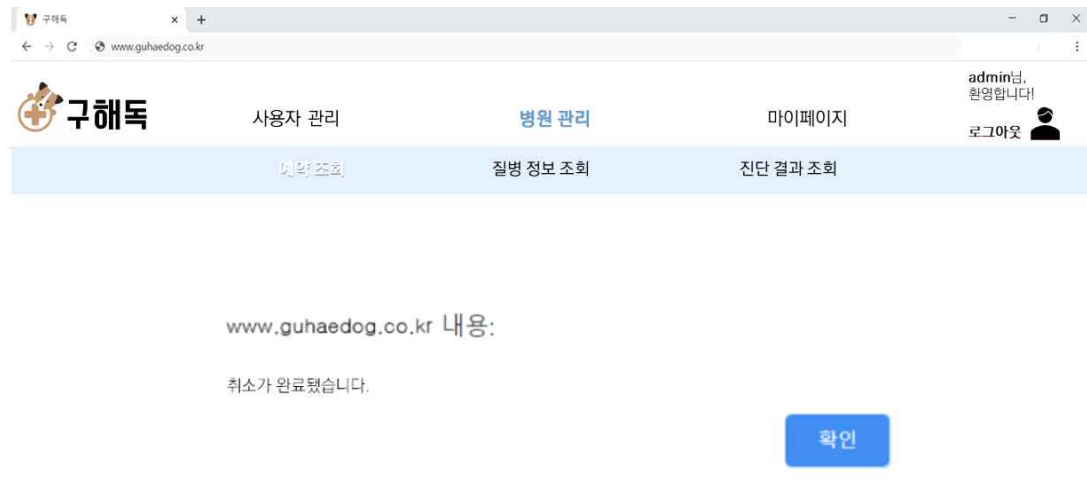


The screenshot shows a web browser window with the URL www.guhaedog.co.kr. The page has a header with the logo '구해독' and navigation links: '사용자 관리', '병원 관리' (active), and '마이페이지'. Below the header is a sub-header with '예약 조회' (active), '질병 정보 조회', and '진단 결과 조회'. The main content area is titled '예약 내역 조회' and contains a table with the following data:

	사용자 이름	예약일자(방문시간)	병원 주소(병원 이름)	전화번호	예약 상태
2	홍길동	2020.04.22 (18:00)	서울 강남구 강남대로54길 7 (명문빌딩) 506동 1202호 (예은동물병원)	010-3232-5656	예약수정 [수정] [취소] [방문완료]
1	이영희	2020.04.16 (16:00)	서울 강남구 강남대로54길 18-6 201호 (24시정담우리동물병원)	010-1234-5678	방문완료

- 화면번호 C18에서 '확인' 버튼을 눌렀을 때의 페이지 화면이다.
- 방문시간이 17시에서 18시로 수정된 것을 확인할 수 있다.

■ 병원 관리 화면(예약 조회 : 예약 일정 취소 완료) - 화면번호 C20



The screenshot shows the same web browser window as the previous one, but the main content area is now a confirmation message. It says 'www.guhaedog.co.kr 내용:' followed by '취소가 완료됐습니다.' (Cancellation is complete). There is a blue button labeled '확인' (Confirm) at the bottom right.

- 화면번호 C19에서 '취소' 버튼을 누르고 취소가 완료되면 다음과 같은 문구가 나타난다.

■ 병원 관리 화면(예약 조회 : 예약 일정 취소 결과) - 화면번호 C21

구해독

www.guhaedog.co.kr

구해독

사용자 관리

병원 관리

마이페이지

admin님,  
환영합니다!

로그아웃 

예약 조회

질병 정보 조회

진단 결과 조회

예약 내역 조회

	사용자 이름	예약일자(방문시간)	병원 주소	전화번호	예약 상태
2	홍길동	2020.04.22 (18:00)	서울 강남구 강남대로54길 7 (명문빌딩) 506동 1202호 (예은동물병원)	010-3232-5656	예약취소
1	이영희	2020.04.16 (16:00)	서울 강남구 강남대로54길 18-6 201호 (24시청담우리동물병원)	010-1234-5678	방문완료

- 화면번호 C20에서 '확인' 버튼을 눌렀을 때의 페이지 화면이다.
- 예약 상태가 '예약 수정'에서 '예약 취소'가 된 것을 확인할 수 있다.

■ 병원 관리 화면(예약 조회 : 방문 완료 결과) - 화면번호 C22

구해독

www.guhaedog.co.kr

구해독

사용자 관리

병원 관리

마이페이지

admin님,  
환영합니다!

로그아웃 

예약 조회

질병 정보 조회

진단 결과 조회

예약 내역 조회

	사용자 이름	예약일자(방문시간)	병원 주소(병원 이름)	전화번호	예약 상태
2	홍길동	2020.04.22 (18:00)	서울 강남구 강남대로54길 7 (명문빌딩) 506동 1202호 (예은동물병원)	010-3232-5656	방문완료
1	이영희	2020.04.16 (16:00)	서울 강남구 강남대로54길 18-6 201호 (24시청담우리동물병원)	010-1234-5678	방문완료

- 화면번호 C19에서 '방문 완료' 버튼을 눌렀을 때의 페이지 화면이다.
- 예약 상태가 '예약 수정'에서 '방문 완료'로 바뀐 것을 확인할 수 있다.

## ■ 병원 관리 화면(질병 정보 조회) - 화면번호 C23



### 질병 정보 조회

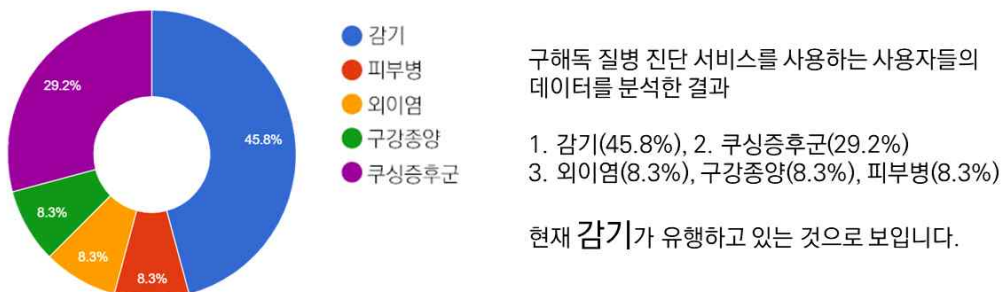
	질병 이름	질병 증상
4	감기	기침, 재채기, 코흘림, 미열
3	쿠싱 증후군	발작, 미열, 재채기
2	피부병	두드러기, 가려움, 코피
1	관절염	절뚝거림, 균형감각 상실,

- '병원 관리' - '질병 정보 조회'를 눌렀을 때의 페이지 화면이다.
- 질병 정보 조회 페이지에서는 등록된 질병의 이름과 그에 대한 증상에 대한 정보를 확인할 수 있다.

## ■ 병원 관리 화면(진단 결과 조회) - 화면번호 C24



### 진단 결과 조회



- '병원 관리' - '진단 결과 조회'를 눌렀을 때의 페이지 화면이다.
- 질병 진단 서비스들의 결과들을 종합하여 차트로 시각화한다.
- 차트를 통해 현재 어떠한 질병이 유행성을 띄우고 있는지 확인할 수 있다.

## 5. REST API 인터페이스 설계

### 5.1. 서버/클라이언트 구조

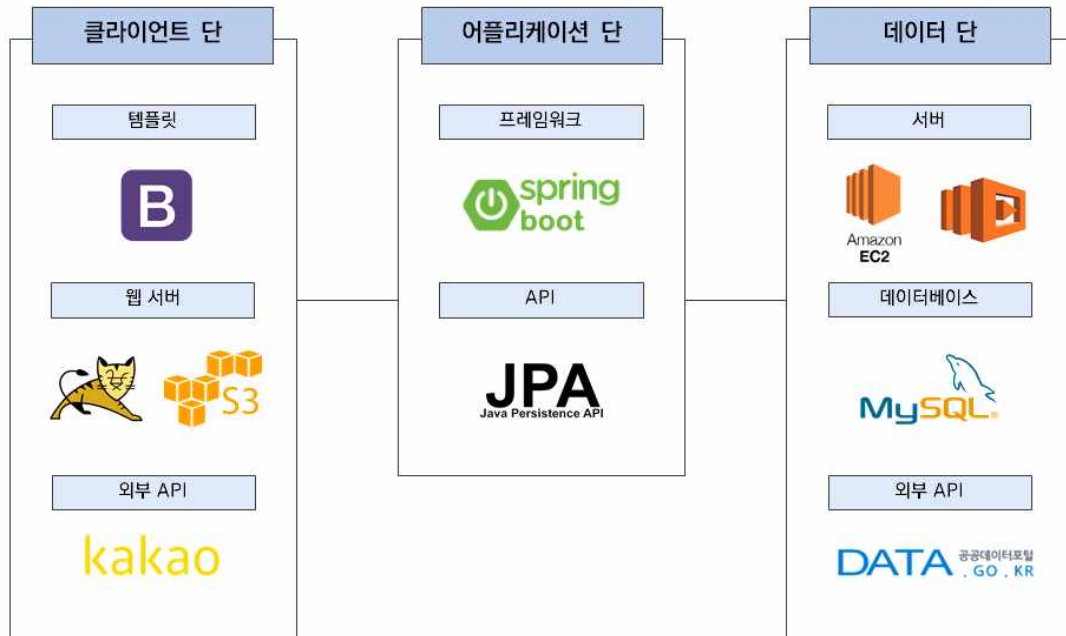


그림 5.1 서비스 인터페이스 구조

본 설계는 그림 5.1과 같이 클라이언트(사용자, 관리자)와 서버(웹 서버)가 AWS 클라우드를 통해 필요한 정보를 송수신하는 인터페이스 전송 구조이다. 즉, 클라이언트가 서버에게 정보를 요청하거나 응답하는 방식으로 REST API를 이용하여 설계한다.

구해독의 메인 서버는 Spring Boot, JPA, Bootstrap 등 내부에 여러 모듈로 구성되어 있다. 데이터베이스는 MySQL을 사용하고, 외부 API인 공공데이터 오픈 API를 이용하여 필요한 데이터를 수집 및 가공하고 저장한다. 그리고 AWS 클라우드는 S3, EC2, Lambda를 사용한다. 구해독의 클라이언트의 일부 기능은 외부 API인 카카오 API를 이용하여 설계한다.

클라이언트가 서버에게 요청하는 메시지 타입은 크게 4가지인 조회(GET), 생성(POST), 수정(PUT), 삭제(DELETE)로 구분하고, 각 요청 별로 알맞은 REST API 방식으로 메시지를 설계한다. 클라이언트와 서버의 통신을 원활히 설계하기 위해 다음과 같이 각 연동 별로 구분하여 REST API를 정의한다.

- 웹 서비스 - 서버와의 연동
- 데이터베이스 - 서버와의 연동

이어서 각 연동 별로 REST API를 정의한 후 상세설계를 한다.

## 5.2. REST API 정의

Method	URI	Description
GET	/user/<user_id>/signin	사용자 로그인
	/user/<user_id>/signout	사용자 로그아웃
	/user/dog/<dog_num>	반려견 정보 조회(사용자)
	/user/dog_disease/<dog_num>	반려견 진단 결과 조회(사용자)
	/user/hospital/<location>/recommendation	병원 위치 기반 추천
	/user/hospital/registration/<regist_num>	병원예약 조회(사용자)
	/admin/<admin_id>/signin	관리자 로그인
	/admin/<admin_id>/signout	관리자 로그아웃
	/admin/<user_id>	사용자 정보 조회
	/admin/dog/<dog_num>	반려견 정보 조회(관리자)
	/admin/dog_disease/<dog_num>	반려견 진단 결과 조회(관리자)
	/admin/hospital/registration/<regist_num>	병원예약 조회(관리자)

Method	URI	Description
POST	/user/signup	사용자 가입
	/user/dog	반려견 정보 등록
	/user/diagnosis/<dog_num>	반려견 질병 증상 등록
	/user/hospital/registration	병원예약 등록

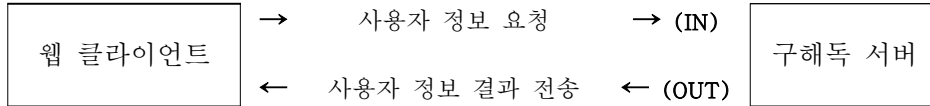


Method	URI	Description
PUT	/user/<user_id>/userinfo	정보수정(사용자)
	/user/dog/<dog_num>/doginfo	반려견 정보 수정(사용자)
	/user/hospital/<regist_num>/registration	병원예약 수정(사용자)
	/admin/<admin_id>/admininfo	정보수정(관리자)
	/admin/<user_id>/userinfo	사용자 정보 수정
	/admin/dog/<dog_num>/doginfo	반려견 정보 수정(관리자)
	/admin/hospital/<regist_num>/registration	병원예약 수정(관리자)

Method	URI	Description
DELETE	/user/<user_id>/delete	사용자 탈퇴
	/user/dog/<dog_num>/delete	반려견 정보 삭제
	/user/hospital/<regist_num>/delete	병원예약 취소(사용자)
	/admin/<admin_id>/delete	관리자탈퇴
	/admin/<user_id>/delete	사용자 정보 삭제
	/admin/dog/<dog_num>/delete	반려견 정보 삭제
	/admin/hospital/<regist_num>/delete	병원예약 취소(관리자)

### 5.3. 상세 REST API 설계

GET /user/<user\_id>/signin



#### Parameter

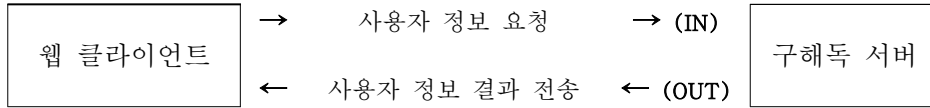
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
user_info		O	Array	사용자 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user123"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "user_info" :
  [
    [
      "user_id" : "user123" ,
      "user_pwd" : "pwd123" ,
      "user_name" : "김회원" ,
      "user_addr" : "서울특별시 강남구 강남대로 78" ,
      "user_birth" : "950423" ,
      "user_phone" : "01012345678" ,
      "regist_day" : "2020-04-20" ,
      "user_flag" : 0 ,
      "status" : "active"
    ]
  ]
}
    
```

GET /user/<user\_id>/signin



#### Parameter

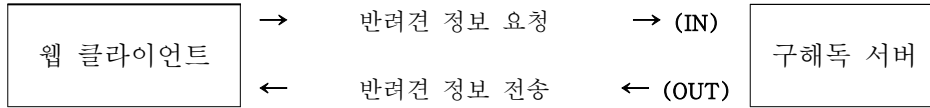
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
user_info		O	Array	사용자 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user123"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "user_info" :
  [
    [
      "user_id" : "user123" ,
      "user_pwd" : "pwd123" ,
      "user_name" : "김회원" ,
      "user_addr" : "서울특별시 강남구 강남대로 78" ,
      "user_birth" : "950423" ,
      "user_phone" : "01012345678" ,
      "regist_day" : "2020-04-20" ,
      "user_flag" : 0 ,
      "status" : "active"
    ]
  ]
}
    
```

GET /user/dog/<dog\_num>



#### Parameter

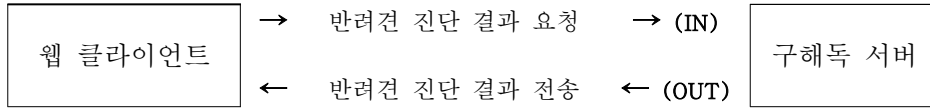
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
dog_info		O	Array	반려견 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 1
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "dog_info" :
  [
    [
      "dog_num" : 1 ,
      "dog_name" : "초코" ,
      "dog_age" : 5 ,
      "dog_sex" : "여자" ,
      "dog_birth" : "150607" ,
      "dog_type" : "토이푸들"
    ]
  ]
}
    
```

GET /user/dog\_disease/<dog\_num>



#### Parameter

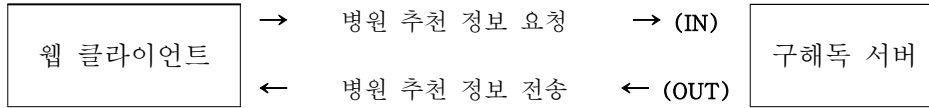
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
disease_info		O	Array	반려견 질병 진단 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 1
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "disease_info" :
  [
    [
      "ds_num" : 1 ,
      "ds_name" : "감기" ,
      "ds_type" : "바이러스" ,
      "ds_symptom" : "기침" ,
      "ds_level" : 1
    ]
  ]
}
    
```

GET /user/hospital/<location>/recommendation



#### Parameter

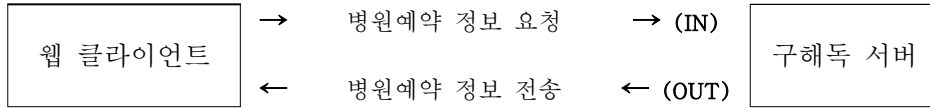
속성	전송방향		Type	Description
	IN	OUT		
location	O		String	요청한 사용자 주소
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
hospital_info		O	Array	병원 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "location" : "서울특별시 용산구"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "order_info" :
  [
    [
      "hosp_num" : 1 ,
      "hosp_name" : "튼튼 동물병원" ,
      "hosp_addr" : "서울특별시 용산구 후암동 3-12" ,
      "open_days" : "월화수목금" ,
      "open_hours" : "09:00~18:00" ,
      "regist_day" : "2020-03-29" ,
      "status" : "영업중"
    ]
  ]
}
    
```

GET /user/hospital/registration/<regist\_num>



#### Parameter

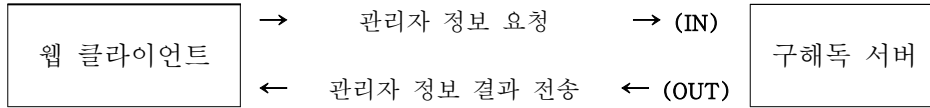
속성	전송방향		Type	Description
	IN	OUT		
regist_num	O		String	요청한 병원예약 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
regist_info		O	Array	병원예약 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "regist_num" : 1
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "order_info" :
  [
    [
      "regist_num" : 1 ,
      "visit_day" : "2020-04-29 16시" ,
      "regist_day" : "2020-04-23" ,
      "status" : "예약완료"
    ]
  ]
}
    
```

GET /admin/<admin\_id>/signin



#### Parameter

속성	전송방향		Type	Description
	IN	OUT		
admin_id	O		String	요청한 관리자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
admin_info		O	Array	관리자 정보

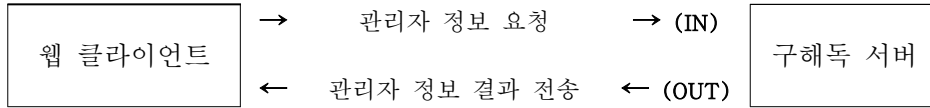
#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "admin_id" : "admin456"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "admin_info" :
  [
    [
      "admin_id" : "admin456" ,
      "admin_pwd" : "pwd456" ,
      "admin_name" : "박관리" ,
      "regist_day" : "2020-04-22" ,
      "status" : "active"
    ]
  ]
}
    
```



GET /admin/<admin\_id>/signout



#### Parameter

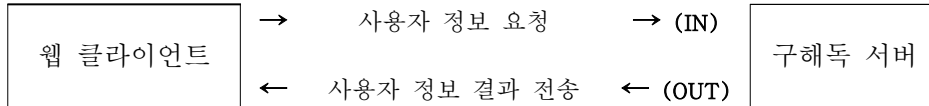
속성	전송방향		Type	Description
	IN	OUT		
admin_id	O		String	요청한 관리자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
admin_info		O	Array	관리자 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "admin_id" : "admin456"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "admin_info" :
  [
    [
      "admin_id" : "admin456" ,
      "admin_pwd" : "pwd456" ,
      "admin_name" : "박관리" ,
      "regist_day" : "2020-04-22" ,
      "status" : "active"
    ]
  ]
}
    
```

GET /admin/<user\_id>



#### Parameter

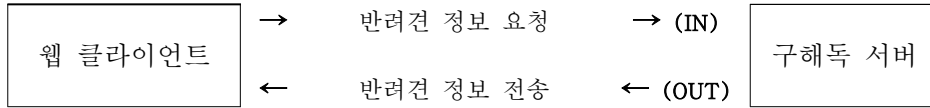
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
user_info		O	Array	사용자 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user789"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "order_info" :
  [
    [
      "user_id" : "user789" ,
      "user_pwd" : "pwd789" ,
      "user_name" : "이회원" ,
      "user_addr" : "서울특별시 논현구 논현대로 34" ,
      "user_birth" : "970809" ,
      "user_phone" : "01022223333" ,
      "regist_day" : "2020-04-23" ,
      "user_flag" : 0 ,
      "status" : "active"
    ]
  ]
}
    
```

GET /admin/dog/<dog\_num>



#### Parameter

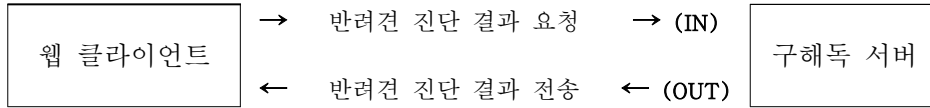
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
dog_info		O	Array	반려견 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 1
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "dog_info" :
  [
    [
      "dog_num" : 1 ,
      "dog_name" : "초코" ,
      "dog_age" : 5 ,
      "dog_sex" : "여자" ,
      "dog_birth" : "150607" ,
      "dog_type" : "토이푸들"
    ]
  ]
}
    
```

GET /admin/dog\_disease/<dog\_num>



#### Parameter

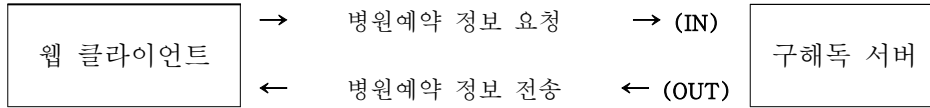
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
disease_info		O	Array	반려견 질병 진단 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 1
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "disease_info" :
  [
    [
      "ds_num" : 1 ,
      "ds_name" : "감기" ,
      "ds_type" : "바이러스" ,
      "ds_symptom" : "기침" ,
      "ds_level" : 1
    ]
  ]
}
    
```

GET /admin/hospital/registration/<regist\_num>



#### Parameter

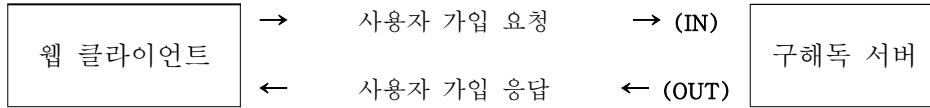
속성	전송방향		Type	Description
	IN	OUT		
regist_num	O		String	요청한 병원예약 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
regist_info		O	Array	병원예약 정보

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "regist_num" : 1
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "order_info" :
  [
    [
      "regist_num" : 1 ,
      "visit_day" : "2020-04-29 16시" ,
      "regist_day" : "2020-04-23" ,
      "status" : "예약완료"
    ]
  ]
}
    
```

POST /user/signup



Parameter

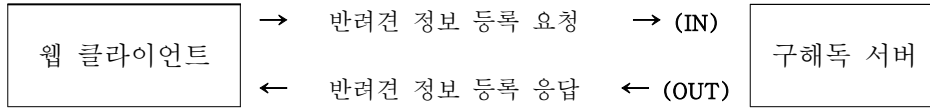
속성	전송방향		Type	Description
	IN	OUT		
user_info	O		Array	사용자 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_info" :
  [
    "user_id" : "user123",
    "user_pwd" : "pwd123" ,
    "user_name" : "김회원" ,
    "user_addr" : "서울특별시 강남구 강남대로 78" ,
    "user_birth" : "950423" ,
    "user_phone" : "01012345678" ,
    "regist_day" : "2020-04-20" ,
    "user_flag" : 0 ,
    "status" : "active"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

POST /user/dog



Parameter

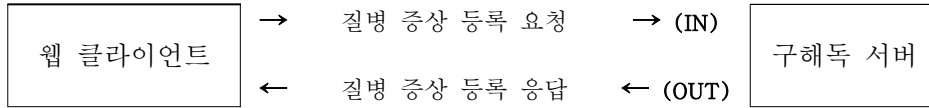
속성	전송방향		Type	Description
	IN	OUT		
dog_info	O		Array	반려견 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_info" :
  [
    "dog_num" : 1 ,
    "dog_name" : "초코" ,
    "dog_age" : 5 ,
    "dog_sex" : "여자" ,
    "dog_birth" : "150607" ,
    "dog_type" : "토이푸들"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

POST /user/diagnosis/<dog\_num>



#### Parameter

속성	전송방향		Type	Description
	IN	OUT		
dog_info	O		Array	반려견 정보
diagnosis_info	O		Array	반려견 질병 증상 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

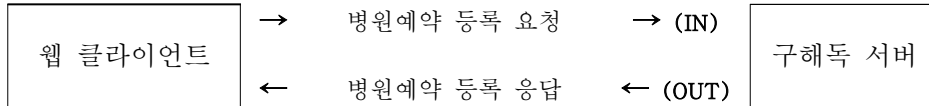
#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_info" :
  [
    "dog_num" : 1 ,
    "dog_name" : "초코" ,
    "dog_age" : 5 ,
    "dog_sex" : "여자" ,
    "dog_birth" : "150607" ,
    "dog_type" : "토이푸들"
  ] ,
  "diagnosis_info" :
  [
    "dg_num" : 1 ,
    "dg_symptom" : "기침" ,
    "dg_day" : "2020-04-24"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```



POST /user/parent/hospital/registration



Parameter

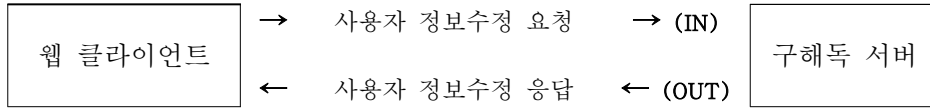
속성	전송방향		Type	Description
	IN	OUT		
dog_info	O		Array	반려견 정보
regist_info	O		Array	병원예약 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_info" :
  [
    "dog_num" : 1 ,
    "dog_name" : "초코" ,
    "dog_age" : 5 ,
    "dog_sex" : "여자" ,
    "dog_birth" : "150607" ,
    "dog_type" : "토이푸들"
  ] ,
  "regist_info" :
  [
    "regist_num" : 1 ,
    "visit_day" : "2020-04-29" ,
    "regist_day" : "2020-04-26" ,
    "status" : "예약완료" ,
    "hosp_num" : 5
    "user_id" : "user123"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /user/<user\_id>/userinfo



#### Parameter

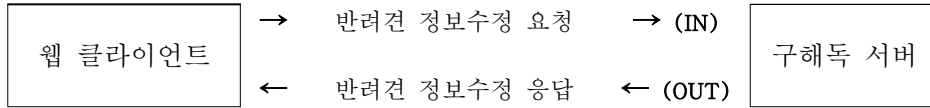
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
user_info	O		Array	요청한 사용자 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user123" ,
  "product_info" :
  [
    "user_pwd" : "pwd789" ,
    "user_addr" : "서울특별시 종로구 사직대로 89-2"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /user/dog/<dog\_num>/doginfo



#### Parameter

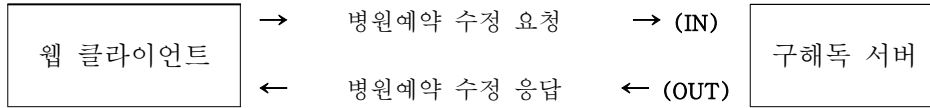
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
dog_info	O		Array	요청한 반려견 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 1 ,
  "dog_info" :
    [
      "dog_name" : "토리" ,
      "dog_age" : 4
    ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /user/hospital/<regist\_num>/registration



#### Parameter

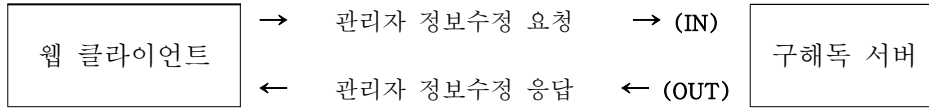
속성	전송방향		Type	Description
	IN	OUT		
regist_num	O		int	요청한 병원예약 식별번호
regist_info	O		Array	요청한 병원예약 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "regist_num" : 3 ,
  "regist_info" :
  [
    "visit_day" : "2020-05-02" ,
    "status" : "예약변경"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /admin/<admin\_id>/admininfo



#### Parameter

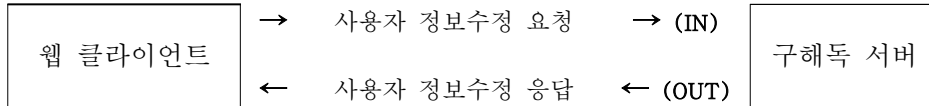
속성	전송방향		Type	Description
	IN	OUT		
admin_id	O		String	요청한 관리자 아이디
admin_info	O		Array	요청한 관리자 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "admin_id" : "admin456" ,
  "admin_info" :
    [
      "admin_pwd" : "pwd332" ,
      "status" : "stop"
    ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /admin/<user\_id>/userinfo



#### Parameter

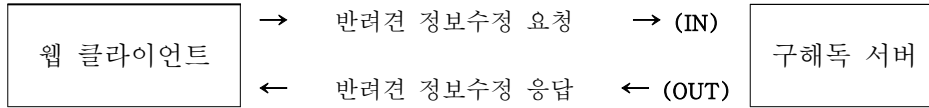
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
user_info	O		Array	요청한 사용자 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user123" ,
  "product_info" :
  [
    "user_pwd" : "pwd789" ,
    "user_addr" : "서울특별시 종로구 사직대로 89-2"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /admin/dog/<dog\_num>/doginfo



#### Parameter

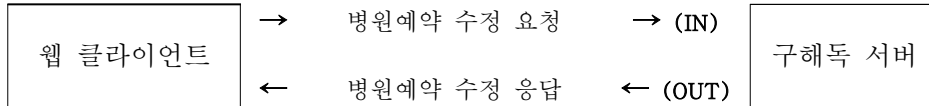
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
dog_info	O		Array	요청한 반려견 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 1 ,
  "dog_info" :
    [
      "dog_name" : "토리" ,
      "dog_age" : 4
    ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

PUT /admin/hospital/<regist\_num>/registration



#### Parameter

속성	전송방향		Type	Description
	IN	OUT		
regist_num	O		int	요청한 병원예약 식별번호
regist_info	O		Array	요청한 병원예약 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

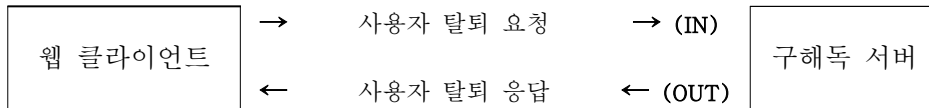
#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "regist_num" : 3 ,
  "regist_info" :
  [
    "visit_day" : "2020-05-02" ,
    "status" : "예약변경"
  ]
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```



**DELETE** /user/<user\_id>/delete



#### Parameter

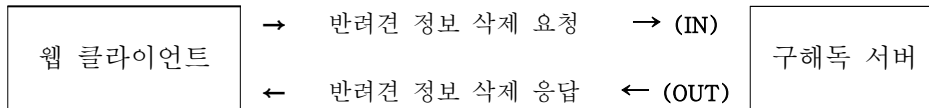
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user123"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

**DELETE** /user/dog/<dog\_num>/delete



**Parameter**

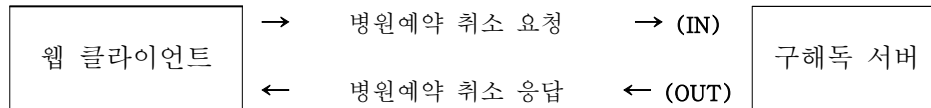
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

**실제 전송내용 (JSON 의 경우라면)**

```

전송방향 : IN
{
  "dog_num" : 8
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

**DELETE** /user/hospital/<regist\_num>/delete



#### Parameter

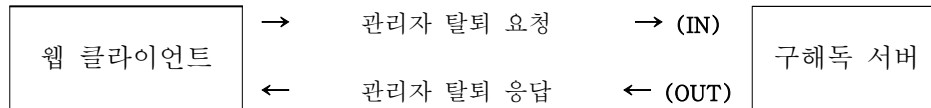
속성	전송방향		Type	Description
	IN	OUT		
regist_num	O		int	요청한 병원예약 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "regist_num" : 2
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

**DELETE** /admin/<admin\_id>/delete



#### Parameter

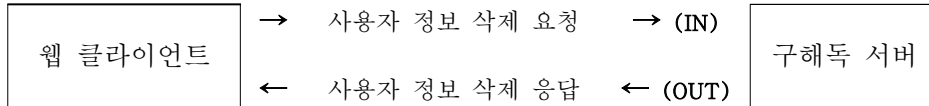
속성	전송방향		Type	Description
	IN	OUT		
admin_id	O		String	요청한 관리자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "admin_id" : "admin256"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

DELETE /admin/<user\_id>/delete



#### Parameter

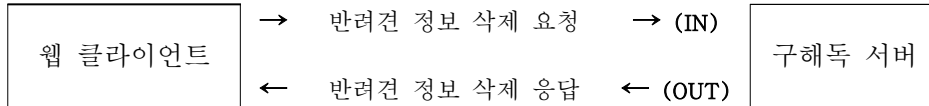
속성	전송방향		Type	Description
	IN	OUT		
user_id	O		String	요청한 사용자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "user_id" : "user123"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

**DELETE** /admin/dog/<dog\_num>/delete



#### Parameter

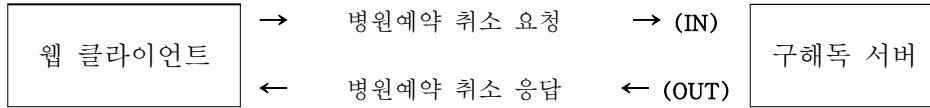
속성	전송방향		Type	Description
	IN	OUT		
dog_num	O		int	요청한 반려견 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "dog_num" : 8
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

**DELETE** /admin/hospital/<regist\_num>/delete



#### Parameter

속성	전송방향		Type	Description
	IN	OUT		
regist_num	O		int	요청한 병원예약 식별번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

#### 실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "regist_num" : 4
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : ""
}
    
```

## 6. 데이터베이스 설계

### 6.1. ERD





## 6.2. 논리적 DB설계 (테이블명세서)

테이블명(사용자) :			T1_USER			
NO	Attribute	Data Type	NN	PK	FK	Description
1	USER_ID	VARCHAR(50)	O	O		사용자아이디
2	USER_PWD	VARCHAR(50)	O			사용자비밀번호
3	USER_NAME	VARCHAR(20)	O			사용자이름
4	USER_ADDR	TEXT	O			사용자주소
5	USER_BIRTH	INTEGER	O			사용자생일
6	USER_PHONE	INTEGER	O			사용자전화번호
7	REGIST_DAY	DATETIME	O			가입일자
8	USER_FLAG	INTEGER	O			사용자종류
9	STATUS	VARCHAR(20)	O			사용자상태

테이블명(관리자) :			T2_ADMIN			
NO	Attribute	Data Type	NN	PK	FK	Description
1	ADMIN_ID	VARCHAR(50)	O	O		관리자아이디
2	ADMIN_PWD	VARCHAR(50)	O			관리자비밀번호
3	ADMIN_NAME	VARCHAR(20)	O			관리자이름
4	REGIST_DAY	DATETIME	O			가입일자
5	STATUS	VARCHAR(20)	O			관리자상태

테이블명(반려견) :			T2_DOG			
NO	Attribute	Data Type	NN	PK	FK	Description
1	DOG_NUM	INTEGER	O	O		반려견식별번호
2	DOG_NAME	VARCHAR(30)	O			반려견이름
3	DOG_AGE	INTEGER	O			반려견나이
4	DOG_SEX	VARCHAR(20)	O			반려견성별
5	DOG_BIRTH	INTEGER	O			반려견생일
6	DOG_TYPE	VARCHAR(50)	O			반려견종류
7	USER_ID	VARCHAR(50)	O		O	사용자아이디

테이블명(질병) :			T2_DOG_DISEASE			
NO	Attribute	Data Type	NN	PK	FK	Description
1	DS_NUM	INTEGER	O	O		질병식별번호
2	DS_NAME	VARCHAR(50)	O			질병명
3	DS_TYPE	VARCHAR(50)	O			질병종류
4	DS_SYMPTOM	TEXT	O			질병증상
5	DS_LEVEL	INTEGER	O			질병위험정도
6	USER_ID	VARCHAR(50)	O		O	사용자아이디
7	DOG_NUM	INTEGER	O		O	반려견식별번호

테이블명(진단) :			T2_DIAGNOSIS			
NO	Attribute	Data Type	NN	PK	FK	Description
1	DG_NUM	INTEGER	O	O		진단식별번호
2	DG_SYMPTOM	TEXT	O			진단증상
3	DG_DAY	DATETIME	O			등록일자
4	USER_ID	VARCHAR(50)	O		O	사용자아이디
6	DOG_NUM	INTEGER	O		O	반려견식별번호

테이블명(병원) :			T2_HOSPITAL			
NO	Attribute	Data Type	NN	PK	FK	Description
1	HOSP_NUM	INTEGER	O	O		동물병원식별번호
2	HOSP_NAME	VARCHAR(50)	O			동물병원명
3	HOSP_ADDR	TEXT	O			동물병원주소
4	OPEN_DAYS	TEXT	O			동물병원운영일자
5	OPEN_HOURS	TEXT	O			동물병원영업시간
6	REGIST_DAY	DATETIME	O			등록일자
7	STATUS	VARCHAR(20)	O			동물병원상태
8	DG_NUM	INTEGER	O		O	진단식별번호
9	DOG_NUM	INTEGER	O		O	반려견식별번호

테이블명(예약) :			T2_REGISTRATION			
NO	Attribute	Data Type	NN	PK	FK	Description
1	REGIST_NUM	INTEGER	O	O		예약식별번호
2	VISIT_DAY	TEXT	O			예약방문일자
3	REGIST_DAY	DATETIME	O			예약일자
4	STATUS	VARCHAR(20)	O			예약상태
5	DG_NUM	INTEGER	O		O	진단식별번호
6	DOG_NUM	INTEGER	O		O	반려견식별번호
7	HOSP_NUM	INTEGER	O		O	동물병원식별번호

### 6.3. 물리적 DB설계 (SQL스크립트)

테이블명	T2_USER
<pre> CREATE TABLE t2_user (   user_id varchar(50) NOT NULL UNIQUE,   user_pwd varchar(50) NOT NULL,   user_name varchar(20) NOT NULL,   user_addr text NOT NULL,   user_birth integer NOT NULL,   user_phone integer NOT NULL,   regist_day datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,   user_flag integer NOT NULL,   status varchar(20) NOT NULL,   PRIMARY KEY (user_id) ); </pre>	

테이블명	T2_ADMIN
<pre> CREATE TABLE t2_admin (   admin_id varchar(50) NOT NULL UNIQUE,   admin_pwd varchar(50) NOT NULL,   admin_name varchar(20) NOT NULL,   regist_day datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,   status varchar(20) NOT NULL,   PRIMARY KEY (admin_id) ); </pre>	

테이블명	T2_DOG
<pre> CREATE TABLE t2_dog (   dog_num integer NOT NULL AUTO_INCREMENT,   dog_name varchar(30) NOT NULL,   dog_age integer NOT NULL,   dog_sex varchar(20) NOT NULL,   dog_birth integer NOT NULL,   dog_type varchar(50) NOT NULL,   user_id varchar(50) NOT NULL,   PRIMARY KEY (dog_num),   FOREIGN KEY (user_id)   REFERENCES t2_user (user_id) ); </pre>	

테이블명	T2_DOG_DISEASE
<pre> CREATE TABLE t2_dog_disease (   ds_num integer NOT NULL AUTO_INCREMENT,   ds_name varchar(50) NOT NULL,   ds_type varchar(50) NOT NULL,   ds_symptom text NOT NULL,   ds_level integer NOT NULL,   user_id varchar(50) NOT NULL,   dog_num integer NOT NULL,   PRIMARY KEY (ds_num),   FOREIGN KEY (user_id)   REFERENCES t2_user (user_id),   FOREIGN KEY (dog_num)   REFERENCES t2_dog (dog_num), ); </pre>	

테이블명	T2_DIAGNOSIS
<pre> CREATE TABLE t2_diagnosis (   dg_num integer NOT NULL AUTO_INCREMENT,   dg_symptom text NOT NULL,   dg_day datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,   user_id varchar(50) NOT NULL,   dog_num integer NOT NULL,   PRIMARY KEY (dg_num),   FOREIGN KEY (user_id) REFERENCES t2_user (user_id),   FOREIGN KEY (dog_num) REFERENCES t2_dog (dog_num) ); </pre>	

테이블명	T2_HOSPITAL
<pre> CREATE TABLE t2_hospital (   hosp_num integer NOT NULL AUTO_INCREMENT,   hosp_name varchar(50) NOT NULL,   hosp_addr text NOT NULL,   open_days text NOT NULL,   open_hours text NOT NULL,   regist_day datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,   status varchar(20) NOT NULL,   dg_num integer NOT NULL,   dog_num integer NOT NULL,   PRIMARY KEY (hosp_num),   FOREIGN KEY (dg_num) REFERENCES t2_diagnosis (dg_num),   FOREIGN KEY (dog_num) REFERENCES t2_dog (dog_num) ); </pre>	

테이블명	T2_REGISTRATION
<pre> CREATE TABLE t2_registration (   regist_num integer NOT NULL AUTO_INCREMENT,   visit_day text NOT NULL,   regist_day datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,   status varchar(20) NOT NULL,   dg_num integer NOT NULL,   dog_num integer NOT NULL,   hosp_num integer NOT NULL,   PRIMARY KEY (regist_num),   FOREIGN KEY (dg_num) REFERENCES t2_diagnosis (dg_num),   FOREIGN KEY (dog_num) REFERENCES t2_dog (dog_num),   FOREIGN KEY (hosp_num) REFERENCES t2_hospital (hosp_num) ); </pre>	

## 7. 환경구성

### 7.1. 개발환경 및 운영환경

#### ■ 서버

구분	개발환경	규격	비고
개발툴	편집툴	VSCode	
	개발IDE	IntelliJ IDEA 2019.3.4.	
	프레임워크	SpringBoot, Bootstrap	
	사용할 미들웨어(모듈)	JDBC 4.3.24 JPA 2.2.5 Thymeleaf 2.1.4 Tomcat 9.0.3 Lombok 1.18.1 Webmvc 5.1.13 H2database 1.4 Mysql 8.0.13 junit4.12	
서버	TCP Port 주소	8080	Docker, EC2
	IP 주소	AWS	
	OS	CentOS	
	DB	Mysql	
서버 하드웨어	CPU	Intel Core2Duo	
	RAM	4G RAM	
	DISK	500G DISK (SATAII)	
	Network	100Mbps	

#### ■ 클라이언트

구분	개발환경	규격	비고
클라이언트	OS	Windows / Linux	
	Browser	Chrome	
하드웨어	CPU	Intel Core i7-7700HQ	
	RAM	16GB	
	DISK	1TB	
	Network	600Mbps	

## 7.2. 소스디렉터리 구조

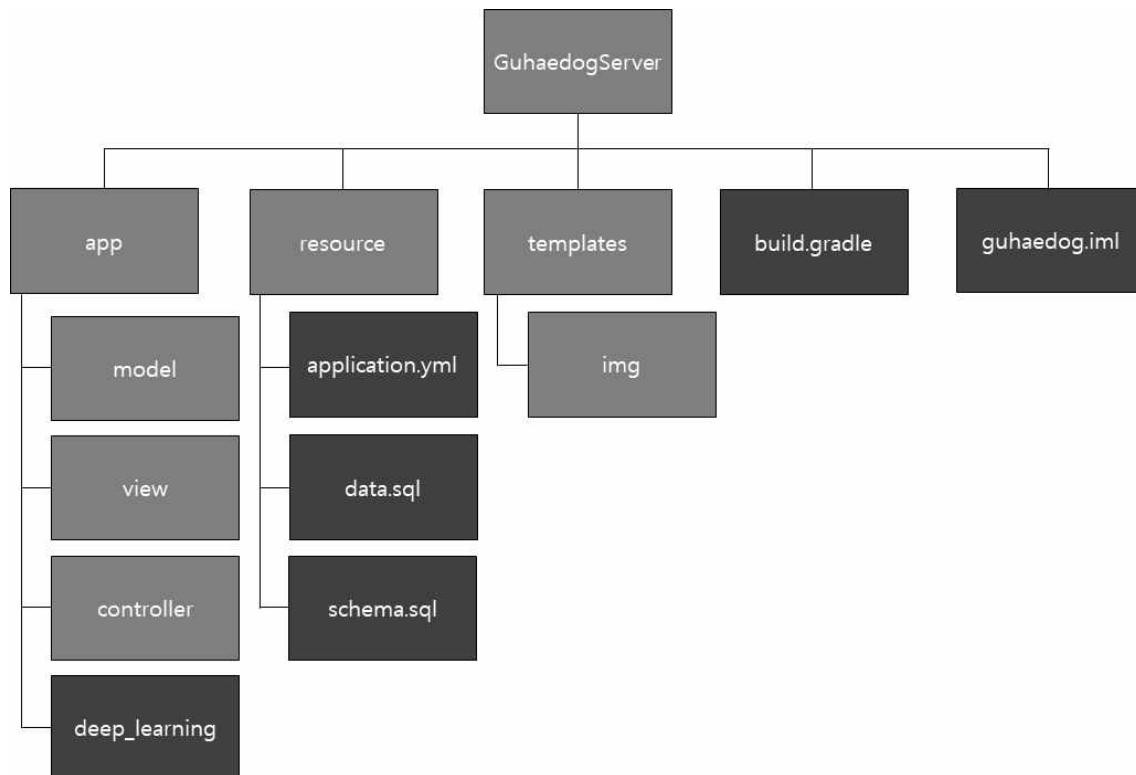


그림 7.2.1 디렉터리 구조

그림 7.2.1과 같은 개발 구조로 MVC 모델 구조를 정의한다. 루트 디렉토리는 GuhaedogServer로 설정한다. 하위 디렉토리는 /app, /resource, /templates와 /app/model, /app/view, /app/controller, /app/deep\_learning, /templates/img로 구성되어 있다.

/app/model : 웹의 비즈니스 로직과 사용되는 데이터를 다룬다.

/app/view : 최종 사용자에게 보여줄 로직을 담당한다. Thymeleaf, Ajax를 통해 구현된다.

/app/controller : 사용의 요청에 따라 비즈니스 로직을 선택하여 호출하고 로직 결과를 view를 통해 보여준다.

/app/deep\_learning : 사용자에게 입력받은 정보와 학습 알고리즘을 사용하여 추론한 결과를 view를 통해 보여준다.

/resource : DB의 환경 설정을 세팅하는 디렉터리이다. applicaion.yml은 사용 DB명 지정, url 지정, jpa와의 연동을 설정한다. data.sql은 DB의 sql문을 명세하며, schema.sql은 DB의 구조와 제약조건에 관해 전반적으로 명세한다.



/templates/img : 정적 파일(이미지 파일 등)을 위한 디렉터리로, 자바스크립트 파일, 스타일 시트 등이 들어가며 브라우저에 HTML 파일 로딩 후, 해당 파일을 호출하면 불러오는 역할을 한다.

/build.gradle : Java의 SpringBoot 프레임워크를 사용하면서 필요한 라이브러리, 로그, 서버 설정 내용이 담겨있는 핵심 파일이다.

/guhaedog.iml : IntelliJ, JAVA 응용 프로그램을 개발하는 데 사용되는 IDE에 의해 생성된 모듈 파일이며 자바, 플러그인, Gradle 성분의 정보, 의존성 및 기타 설정을 저장한다.

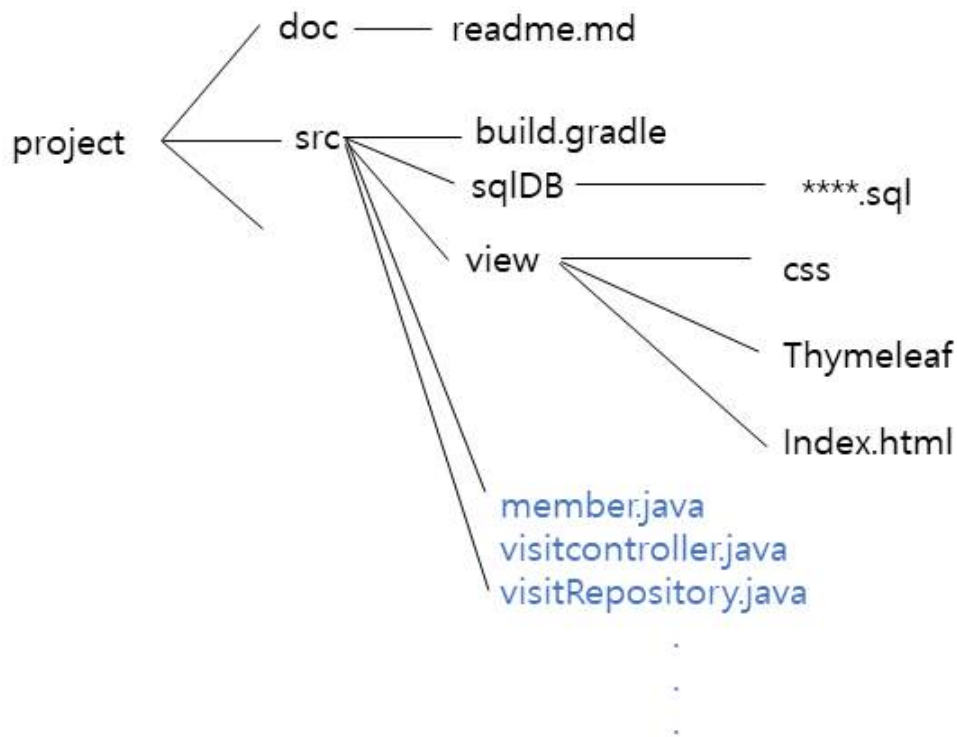


그림 7.2.2 디렉터리 및 파일환경

그림 7.2.2와 같이 세부 디렉터리와 파일명으로 모듈화를 정의한다.