

2021년도

학사학위논문

React와 Node.js를 이용한 웹 기반  
동물병원 관리 시스템 구현

Implement of a Web-Based Animal  
Hospital Management System  
using React and Node.js.

2021년 11월 30일

순천향대학교 공과대학  
컴퓨터공학과

노서현 구민주

React와 Node.js를 이용한 웹 기반

동물병원 관리 시스템 구현

Implement of a Web-Based Animal

Hospital Management System

using React and Node.js.

지도교수 천 인 국

이 논문을 공학사학위 논문으로 제출함

2021년 11월 30일

순천향대학교 공과대학

컴퓨터공학과

노서현 구민주

노 서 현

구 민 주 의 공학사학위논문을 인준함

2021년 11월 30일

심 사 위 원 인

심 사 위 원 인

순천향대학교 공과대학

컴퓨터공학과

## 초 록

최근 한국 사회의 1인 세대화와 고령화가 급속하게 진행되면서 ‘펫팸족 (Pet+Family)’이라는 신조어가 등장할 정도로 반려동물에 관한 개인의 관심과 수요가 증가하고 있다. 그 결과로, 식품, 케어 용품, 케어 서비스 뿐만 아니라 반려동물의 건강과 관련된 반려동물 시장 규모가 눈에 띄게 증가하고 있다. 이에, 본 논문은 동물병원 서비스와 관련하여 반려인들이 겪는 불편함에 주목하였고, 반려동물의 건강에 관한 진료 정보를 얻기 어렵다는 소비자들의 불만을 해결하기 위해 반려동물의 건강관리를 위한 전용 관리 및 지역 동물병원과의 연동을 통해 케어 서비스망의 구축과 반려동물을 기르는 사람들의 관리 서비스 운영을 위하여 필요한 Web Site의 구축에 관한 연구이다.

동물병원 관리 시스템은 최근 많이 이용되고 있는 Javascript 라이브러리인 React와 Javascript 런타임으로써 확장성 있는 네트워크 애플리케이션 개발에 사용되는 소프트웨어 플랫폼인 Node.js를 이용하여 구현하도록 한다.

본 논문에서는 병원 사용자는 회원 관리 및 진료 내역 관리가 가능하고, 일반 사용자는 회원 가입 및 로그인, 동물병원 진료 예약, 진료 내역을 확인할 수 있는 사이트를 구현하였고, 해당 웹사이트의 제작과 설계 과정에 대해 기술하였으며, React와 Node.js를 사용해 구현함으로써 가지는 장단점에 대해 분석하였다.

병원 측 사용자는 회원 관리페이지에서 회원 조회 및 삭제, 진료 예약 내역 확인, 진료 내용을 입력하는 것을 가능하도록 하였으며, 일반 사용자는 회원 가입 및 로그인, 등록된 반려동물의 진료 예약 및 진료내역 확인이 가능하도록 구현하

였다. 이는 반려인들이 반려동물의 건강 관리 내역을 한눈에 보고 쉽게 관리할 수 있도록 돕는다. 병원 또한 예약을 온라인으로 관리함으로써 진료 및 수술 시간을 더 쉽고 빠르게 확인이 가능하다.

주요어 : 동물병원 관리 시스템, React, Node.js

## 차 례

제 1 장 서 론 .....	1
제 2 장 이론적 배경 .....	4
2.1 선행 연구 .....	4
2.1.1. Petdoc .....	4
2.1.2. 펫츰 .....	4
2.2. 필요 기술 .....	5
2.2.1. Javascript .....	5
2.2.2 React .....	5
2.2.3. Node.js .....	7
제 3 장 동물 병원 관리 시스템 설계 .....	9
3.1. 요구사항 분석 .....	9
3.2. 시스템 구성 .....	10
3.3. 서버 .....	12
3.4. 클라이언트 .....	13
3.5. 데이터베이스 .....	19
제 4 장 동물 병원 관리 시스템 구현 및 테스트 .....	24
4.1. 개발 환경 .....	24
4.2. 화면 구성 .....	24

제 5 장 결 론 .....	34
-----------------	----

## 그림 차례

그림 1-1	1
그림 1-2	2
그림 2-1	6
그림 2-2	8
그림 3-1	10
그림 3-2	11
그림 3-3	12
그림 3-4	14
그림 3-5	15
그림 3-6	17
그림 3-7	18
그림 4-1	25
그림 4-2	26
그림 4-3	27
그림 4-4	28
그림 4-5	29
그림 4-6	30
그림 4-7	31



그림 4-8	.....	31
그림 4-9	.....	32
그림 4-10	.....	33

## 표 차 례

표 3-1	.....	6
표 3-2	.....	10
표 3-3	.....	15
표 3-4	.....	16
표 3-5	.....	16
표 3-6	.....	17
표 3-7	.....	17
표 4-1	.....	19

## 제 1 장 서 론

최근 한국 사회의 1인 세대화와 고령화가 급속하게 진행되면서 ‘펫팸족 (Pet+Family)’이라는 신조어가 등장할 정도로 반려동물에 대한 개인의 관심과 수요가 크게 증가하고 있다. 이에 따라, 반려동물 시장의 규모 또한 점점 커지고 있는 추세이다. 식품, 케어 용품, 케어 서비스뿐만 아니라 반려동물의 건강과 관련되어 반려인들의 관심이 집중되고 있다. 특히, 인간보다 수명이 짧은 반려동물의 건강을 염려하여 보험, 장례, 동물병원 서비스 등의 시장이 커지고 있음에 따라 다양한 반려동물 플랫폼들도 등장하기 시작했다.



[그림 1-1] 펫팸족의 등장과 반려동물 시장 규모 증가

본 논문은 동물병원 서비스와 관련하여 반려인들이 겪는 불편함에 주목하였고, 반려동물 의료 서비스와 관련된 문제들이 두드러졌다. 그중 가장 큰 문제점은 반려동물의 건강에 관한 진료 정보를 소비자들이 얻기 어렵다는 것이다. 따라서 이러한 소비자들의 불만을 해결하기 위해 반려동물의 건강 정보를 제공하고 관리 및 지역 동물병원과의 연동을 통해 케어 서비스망의 구축과 반려동물의 관리 서비스 제공을 위하여 필요한 Web Site의 구축 방법에 관하여 연구한다.



[그림 1-2] 동물병원 관련 소비자 불만족이유

동물병원 관리 시스템은 React와 Node.js를 이용하여 서비스를 구현한다. React는 웹의 View를 개발할 수 있도록 하는 Javascript 라이브러리이며, 컴포넌트 기반으로 재사용성이 뛰어나다. 특히, Virtual Dom을 활용하여 업데이트해야 하는 DOM 요소를 찾아 해당 부분만 업데이트하므로 리렌더링이 잦은 웹에서 빠른 성능을 보이는 것이 특징이다. 따라서 최근 개발자들이 가장 선호하는 프론트엔드 프레임워크이다.

Node.js는 Javascript 런타임으로써 확장성이 있는 네트워크 애플리케이션 개발에 사용되는 소프트웨어 플랫폼이다. 이는 Javascript를 동일하게 사용하여 서

머를 빠르게 개발하고 응용할 수 있어 각광받고 있다.

## 제 2 장 이론적 배경

### 2.1 선행 연구

#### 2.1.1. Petdoc

Petdoc은 수의사의 전문 의료 상담을 실시간 채팅으로 제공하여 반려인들이 반려동물의 건강 상담을 빠르게 받아 볼 수 있는 앱이다. 특수 동물들의 의료상담 또한 가능하다는 것이 특징이며, 토막 건강 상식을 제공하는 것과 타 이용자의 상담 내역 조회도 가능하다. 반려인이 직접 반려동물의 케어를 기록할 수 있다는 것도 해당 앱의 특징이다.

#### 2.1.2. 펫츰

서울대학교 수의과대학의 연구팀이 개발에 참여한 반려동물의 건강관리에 초점을 맞춘 앱이다. 반려동물의 연령대에 맞춰 검진 스케줄과 체크리스트, daily To-Do List를 제공하여 반려인이 체계적이고 주기적으로 반려동물을 케어할 수 있도록 돕는 것이 특징이다.

제안한 동물병원 관리 시스템에서는 위의 두 서비스 이용자들의 리뷰를 참고하여 개인의 케어 기록이 아니라 지역 병원과의 연계를 통해 전문적인 케어 기록을 관리할 수 있는 서비스를 제안한다. 위의 서비스들은 반려인이 직접 일기처럼 기록하며, 수의사의 상담 연계 수준에 그치며, 그마저도 잘 이루어지고 있지 않다는 평과 병원 연계의 필요성을 말하는 의견이 대부분이었다. 따라서 반려동물의 건강관리를 보다 전문적으로 하기 위해서는 병원과의 연계가 중요하므로 구현 방향 또한 병원과 반려인의 상호작용에 초점을 맞추어 설계한다.

## 2.2. 필요 기술

### 2.2.1. Javascript

자바스크립트(Javascript)는 정적인 HTML을 동적으로 표현하기 위해 만들어진 프로그래밍 언어로 클라이언트 단에서 웹 페이지의 동작을 담당하며, 프로토타입 기반 객체 생성을 지원하는 동적 스크립트 언어이다. 부모 자식의 구분, 메인 메소드, 클래스-인스턴스의 관계가 없으며, 모든 함수가 클래스도, 인스턴스도 될 수 있어 모든 요소가 따로 작동한다. 이는 개발 속도가 빠르고 문법이 간단하다는 장점이 있지만, 복잡한 프로그램을 만들기는 어렵다는 특징이 있다.

브라우저에서 웹 페이지를 불러올 때, 실행 환경(브라우저) 안에서 HTML, CSS, Javascript 코드가 실행된다. 자바스크립트는 HTML과 CSS가 결합되어 웹 페이지 상에 올려진 후에 브라우저의 자바스크립트 엔진에 의해 실행된다. 이는 웹 브라우저에 설치된 Browser API 중 하나인 DOM(Document Object Model)<sup>1)</sup>을 통해 HTML과 CSS를 동적으로 수정하는 것을 가능케 한다.

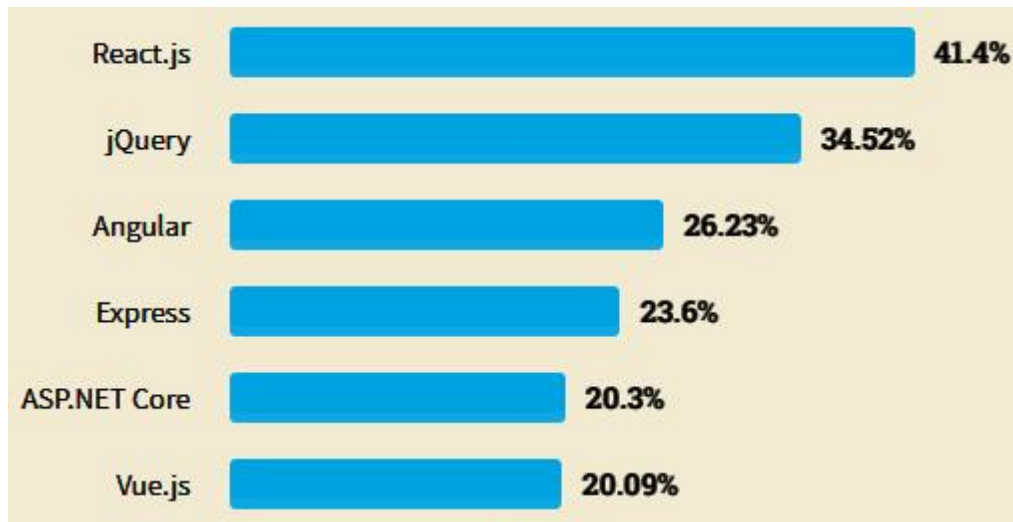
### 2.2.2 React

프레임워크(Framework)란, 개발자가 원하는 기능의 구현에 집중해 빠른 속도로 개발을 할 수 있도록 필요한 기능을 갖추고 있는 클래스나 라이브러리의 모음을 의미한다. 자바스크립트의 프레임워크에는 React, Vue.js, Angular 등이 있

---

1) DOM(Document Object Model) : 프로그래밍 언어가 DOM 구조에 접근할 수 있게 하여, 문서의 구조, 스타일, 내용 등을 변경할 수 있게 돕는다.

으며, 본 논문에서는 React를 이용한다. 아래 [그림 2-1]의 설문조사 결과를 보면 최근 가장 많이 사용하는 웹 프레임워크로 React가 가장 높은 순위를 차지했다.



[그림 2-1] Stack Overflow 2021 Developer Survey 웹 프레임워크 순위

React는 사용자 인터페이스를 구축하기 위해 사용되는 선언적이고 효율적인 오픈 소스 자바스크립트 라이브러리로, 컴포넌트를 이용하여 코드의 재사용성과 유지보수성을 증가시켜 복잡한 UI를 쉽게 구성하도록 돕는다. 또한 Dirty checking<sup>2)</sup>과 Virtual DOM<sup>3)</sup>을 활용해 업데이트 할 DOM 요소를 찾아 해당 부분만 업데이트 하므로, 리렌더링이 잦은 동적인 웹에서 빠른 퍼포먼스를 보여주는 것을 가능케 하며, 모듈형으로 개발한다. 이는 배우기가 간단하고, 복잡함이 적다는 장점이 있다. 그러나 View 외의 라우팅, API 통신 등의 기능들은 직접 구현하거나 추가 라이브러리를 사용하여 구현해야 하기 때문에 자바스크립트 배경지

2) Dirty Checking : 코드에 별도로 데이터베이스에 update하는 쿼리가 존재하지 않음에도 불구하고 update 쿼리가 실행되는 것을 말한다.

3) Virtual DOM : 실제 DOM에 접근하여 조작하는 대신, 추상화시킨 자바스크립트 객체를 구성하여 사용하는 것을 말한다.



식이 부족할 경우 사용이 힘들다는 단점이 있다.

### 2.2.3. Node.js

Node.js는 자바스크립트 엔진으로 빌드된 자바스크립트 런타임<sup>4)</sup>으로, 웹브라우저에 종속적인 자바스크립트를 외부에서도 실행할 수 있는 환경을 제공한다. 이는 확장성이 있는 네트워크 어플리케이션 개발에 사용되는 소프트웨어 플랫폼으로 특히 서버 사이드에서 많이 사용되고 있다. 내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 별도의 소프트웨어가 없이 동작하는 것이 가능하며, 플랫폼의 제약에서 벗어나 어디든 상관없이 실행이 가능하게 한다.

Node.js를 이용하면 자바스크립트를 동일하게 사용하여 서버 로직을 처리하는 것이 가능하고, npm(node package manager)을 통하여 다양한 모듈(패키지)을 제공한다는 장점이 있다. 그러나 단일 스레드(Single Thread)이기 때문에 하나의 작업에 많은 시간이 걸리는 경우, 전체 시스템의 성능이 떨어진다는 것과 코드가 수행되고 난 후에야 에러 발생 유·무의 확인이 가능하다는 단점이 존재하지만, Javascript를 구현하는 환경으로 Node.js가 가장 많이 사용되고 있음을 [그림 2-2]를 통해 알 수 있다.

---

4) 런타임(Runtime) : 프로그래밍 언어가 구동되는 환경



[그림 2-2] Other-Tools 설문조사, The State of javascript 2020

## 제 3 장 동물 병원 관리 시스템 설계

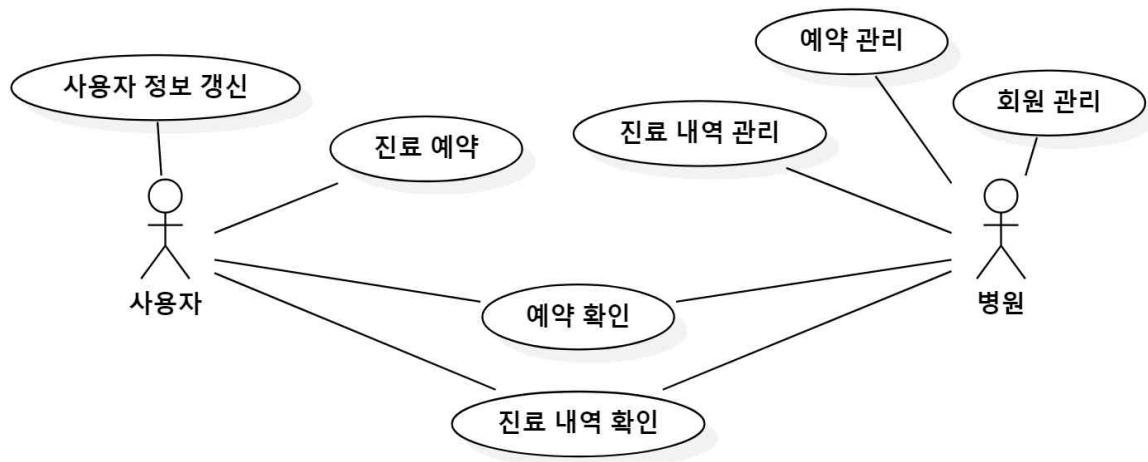
동물 병원 관리 시스템의 설계는 아래와 같다.

### 3.1. 요구사항 분석

동물 병원 관리 시스템의 요구 분석서는 아래의 [표 3-1]과 같다.

[표 3-1] 동물 병원 시스템의 요구 분석서

Index	화면명	요구사항명	요구사항 내용
1	회원 가입	비회원 가입	[프로필 이미지], [ID], [비밀번호], [이름], [생년월일]을 입력한다.
2	로그인	로그인	[ID], [비밀번호]를 입력한다.
3	진료 예약	예약	[예약 날짜], [예약할 동물 이름], [예약할 병원 이름]을 선택한다.
4	예약 조회	예약 내역 조회	예약한 내역을 조회한다. 현재 날짜보다 과거에 있는 예약 내역은 제외한다.
5	진료 내역 조회	진료 내역 조회	[진료를 받았던 동물 이름]을 선택하면 현재까지 받았던 진료 기록을 표시한다.

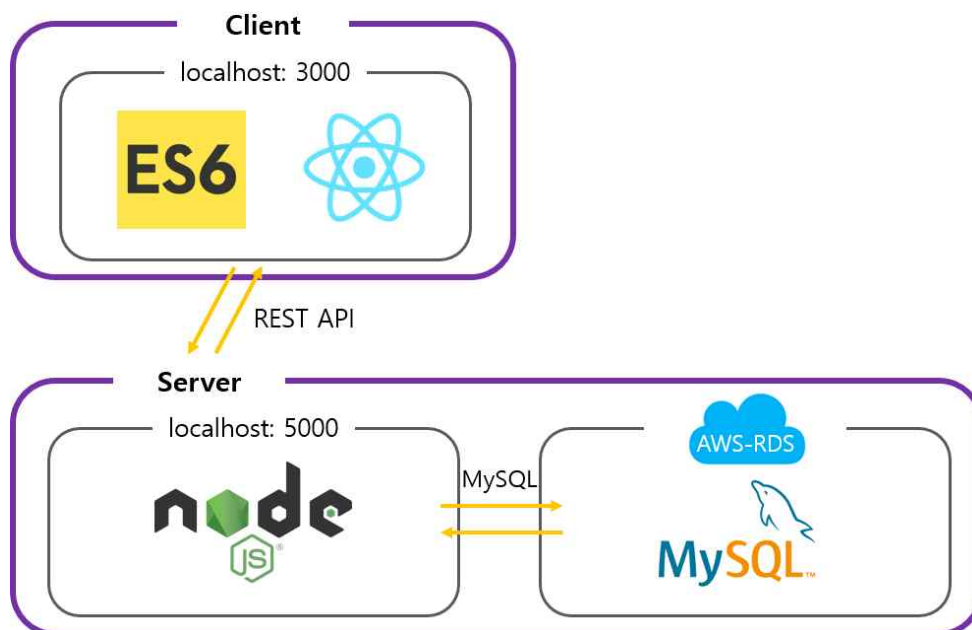


[그림 3-1] 유스케이스 다이어그램

동물 병원 관리 시스템의 유스케이스 다이어그램은 위의 [그림 3-1]과 같다. 동물 병원 관리 시스템의 사용자는 일반 사용자(회원)와 병원 사용자이다. 이들은 공통적으로 예약 확인과 진료 내역을 확인할 수 있으며, 일반 사용자는 사용자의 개인 정보 갱신, 진료 예약의 기능을 이용할 수 있다. 병원 사용자는 병원을 이용하는 회원의 관리, 예약 관리, 일반 사용자의 진료 내역 관리 기능을 이용할 수 있다.

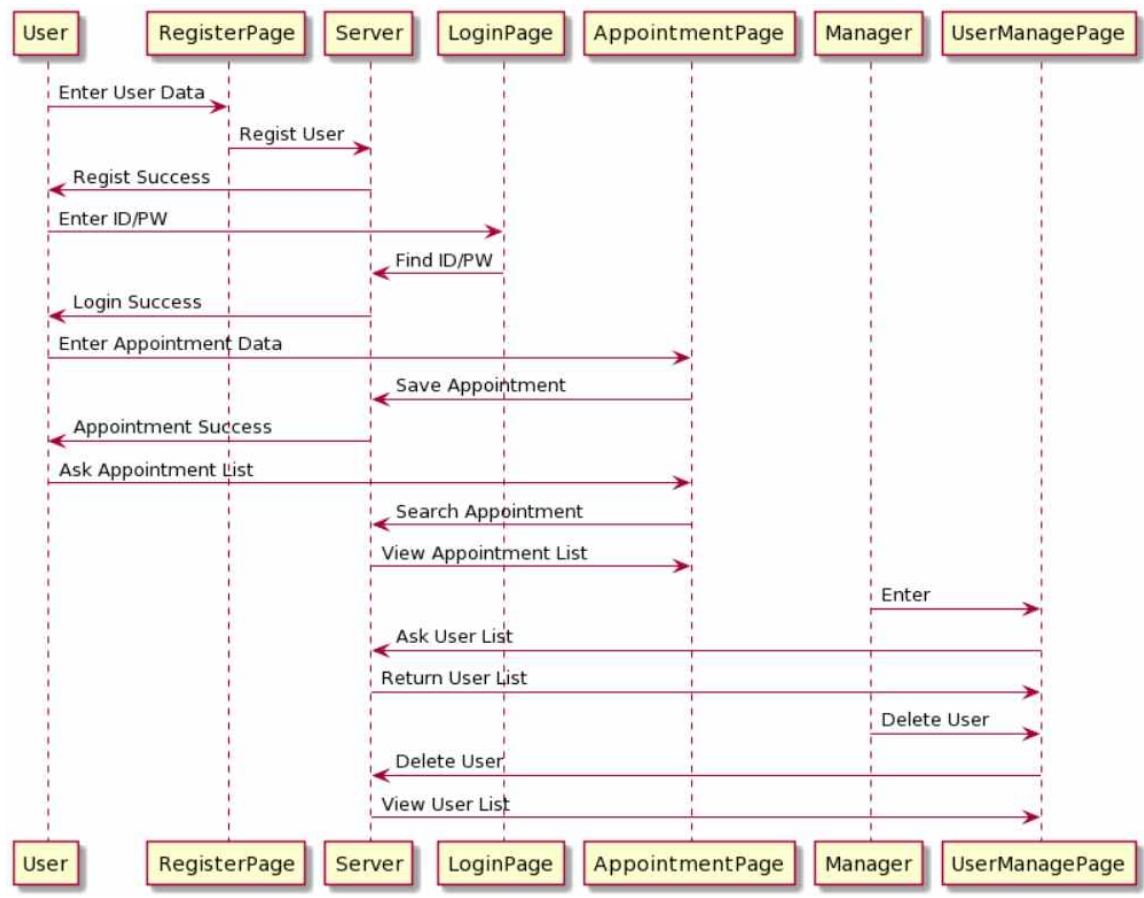
### 3.2. 시스템 구성

동물 병원 관리 시스템의 시스템 구성도는 아래의 [그림 3-2]와 같다.



[그림 3-2] 동물 병원 관리 시스템 구성도

동물 병원 관리 시스템은 클라이언트 개발에 자바스크립트 ES6와 React, 서버 개발에는 Node.js를 이용한다. 데이터베이스는 AWS에서 제공하는 분산 관계형 데이터베이스인 RDS의 MySQL 인스턴스를 이용한다. 클라이언트와 서버는 REST API를 통해 통신하고, 서버와 데이터베이스는 서버로부터 쿼리문을 수신 받으면 데이터베이스가 서버로 데이터를 송신하는 형식으로 통신한다. 클라이언트는 3000번, 서버는 5000번으로 서로 다른 포트 번호를 이용하여 통신하기 때문에 프록시 설정을 통해 원활한 통신이 이루어질 수 있도록 한다.



[그림 3-3] 시퀀스 다이어그램

위의 [그림 3-3]은 시나리오를 보여주기 위한 시퀀스 다이어그램이다. 클라이언트(User, Manager)의 요청에 따라 어떠한 순서로 어떤 객체들과 상호작용하였는지를 보여주며, 자세한 흐름은 3.4.에서 기술하도록 한다.

### 3.3. 서버

동물 병원 관리 시스템의 서버 개발에는 Node.js를 이용하여 구축하고, 기본

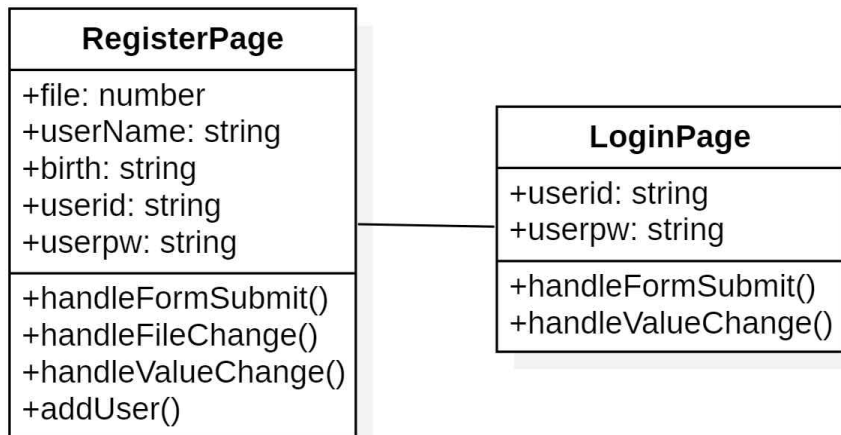
적으로 REST API를 이용하여 클라이언트와 통신한다. 아래의 [표 3-2]는 동물 병원 관리 시스템의 API 명세서이다.

[표 3-2] 동물 병원 관리 시스템 서버 API 명세서

Index	Method	URI	Description
1	POST	/api/users/register	회원 가입
2	POST	/api/users/login	로그인
3	GET	/api/users/info	회원 정보 조회
4	DELETE	/api/users/:id	회원 삭제
5	POST	/api/animals	동물 목록 조회
6	POST	/api/animals/animal_name	동물 이름 조회
7	POST	/api/animals/animal_num	동물 번호 조회
8	GET	/api/hospitals	병원 목록 조회
9	POST	/api/hospitals/h_name	병원 이름 조회
10	POST	/api/hospitals/h_num	병원 번호 조회
11	POST	/api/appointment	진료 예약
12	POST	/api/appointment/check	예약 내역 조회
13	POST	/api/med_history	진료 내역 조회

### 3.4. 클라이언트

동물 병원 관리 시스템의 클라이언트 개발에는 React를 사용하여 구현한다. 클라이언트의 클래스 다이어그램은 아래의 [그림 3-4]와 같다.

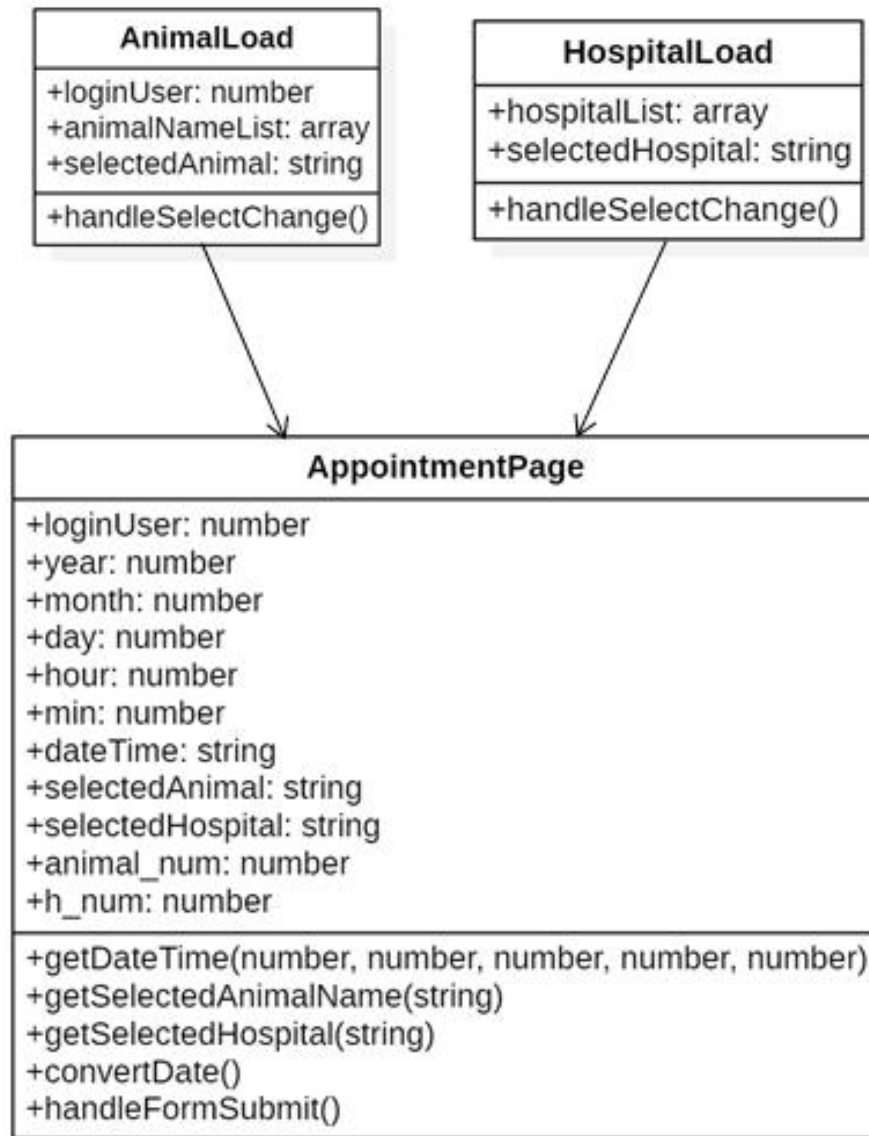


[그림 3-4] 회원 가입, 로그인 페이지 클래스 다이어그램

RegisterPage(회원 가입 페이지)는 파일 이름, 사용자 이름, 생년월일, 아이디, 비밀번호를 입력받는다. handleFileChange()와 handleValueChange()는 사용자가 파일이나 input의 입력 값을 변경하였을 때 상태를 변경해 주는 함수이다. 회원 가입 버튼을 클릭하게 되면 handleFormSubmit() 함수가 addUser()함수를 호출하여 서버에 회원의 정보들을 전송하여 데이터베이스에 저장하고, 회원 가입의 성공 유·무를 반환하여 성공이면 로그인 페이지로 전달한다.

LoginPage(로그인 페이지)는 아이디, 비밀번호를 입력받는다. 이때, handleValueChange()는 회원 가입 페이지에서와 같은 기능을 수행한다. 사용자가 제출 버튼을 누르게 되면 handleFormSubmit() 함수가 호출되어 실행된다. 이 함수는 입력받은 정보를 서버에 전달하여 데이터베이스에 존재하는 ID인지와 입력받은 비밀번호가 데이터베이스에 저장되어 있는 비밀번호와 동일한 지를 확인한다. ID와 ID의 비밀번호가 모두 일치하면 로그인 성공을, 하나라도 일치하지 않는다면 로그인 실패를 반환한다. 로그인에 성공하였을 때는 현재 로그인 한 사용자의 아이디 번호를 쿠키에 저장하여 메인 페이지로 전달한다.



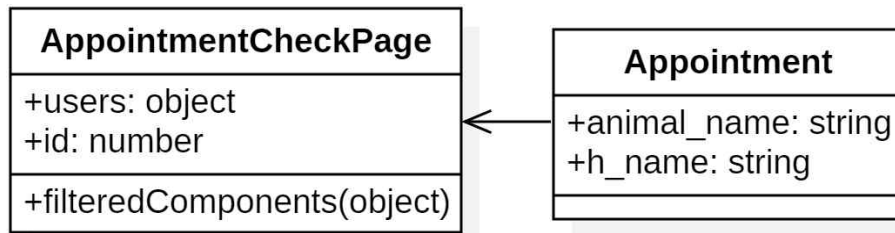


[그림 3-5] 예약 페이지(Appointment Page) 클래스 다이어그램

위의 [그림 3-5]는 예약 페이지의 클래스 다이어그램이다. Appointment

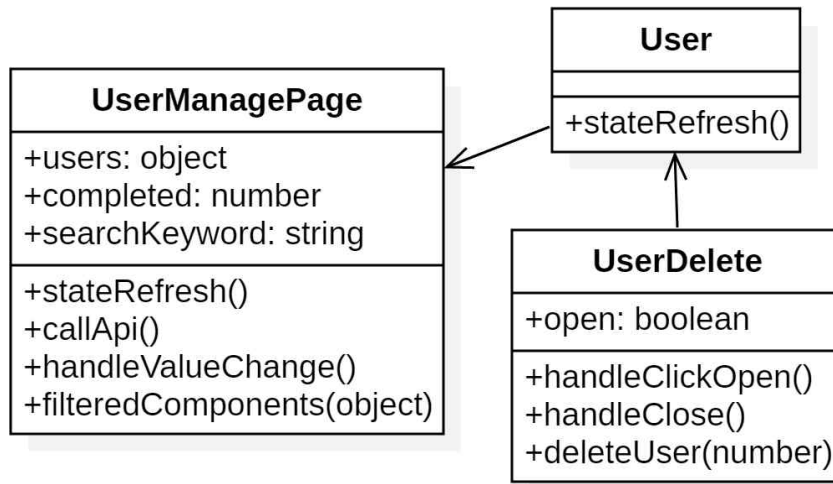
Page(예약 페이지)에서는 쿠키에 저장되어있는 loginUser(현재 로그인 중인 사용자)를 가져와 getDateTIme() 함수를 사용해 사용자가 선택한 예약 날짜를 받아 오고 year, month, day, hour, min에 각각 저장하고 convertDate() 함수를 호출한다. 호출된 convertDate() 함수는 값을 “YYYY-MM-DD hh:mm”의 형태로 변환하여 dateTime 변수에 저장한다. 그리고 나서 현재 로그인 중인 사용자에게 등록되어 있는 동물들의 이름을 드롭다운 메뉴로 보여주고, 선택된 동물의 이름(selectedAnimal)을 다시 반환하여 주는 AnimalLoad() 함수가 호출된다. 반환된 동물의 이름은 다시 예약 페이지의 selectedAnimal 변수에 저장한다. 이후의 HospitalLoad()에서 또한 마찬가지로 전체 동물병원 이름을 드롭다운 메뉴로 보여주고, 선택된 병원의 이름(selectedHospital)을 반환한다. 반환된 동물병원의 이름은 selectedHospital 변수에 저장한다.

모든 항목의 선택을 완료한 후에 예약 버튼을 누르면 handleFormSubmit() 함수가 호출된다. 해당 함수는 selectedAnimal과 selectedHospital을 서버에 전송하여 해당 동물과 동물병원의 고유 번호를 가져와 각각 animal\_num과 h\_num 변수에 저장한다. 그 후 dateTime, loginUser, animal\_num, h\_num을 다시 서버에 전송하여 APPOINTMENT 데이터베이스에 예약 정보들을 저장한다. 예약에 성공하면 예약 성공을 메인 화면으로 다시 전달한다.



[그림 3-6] 예약 확인 페이지 클래스 다이어그램

예약 확인 페이지(AppointmentCheckPage)는 사용자가 예약했던 내역들을 확인할 수 있는 페이지이다. 예약 확인 페이지에 접속하면 현재 쿠키에 저장되어있는 id를 서버로 전송하여 해당 아이디로 예약된 내역을 데이터베이스를 이용하여 검색하여 object의 형태로 받아오게 된다. 받아온 object 들은 users 변수에 저장한다. 이후에 filteredComponents() 함수에 이 users를 매개변수로 넘겨주면 각 요소에 대해 자바스크립트의 내장함수인 map() 함수를 이용하여 Appointment로 전달한다. Appointment는 전달받은 예약 내역들을 render()함수를 이용하여 화면에 그려준다.



[그림 3-7] 회원 관리페이지 클래스 다이어그램

회원 관리 페이지(**UserManagePage**)는 전체 회원들의 정보를 확인하고 관리할 수 있는 페이지이다. 먼저 `callApi()` 함수를 통해 서버에 `get` 요청을 보내 **USER** 데이터베이스에 있는 내용을 받아와 `users`에 저장한다. 특정 사용자를 찾을 때는 찾으려는 사용자의 이름을 검색창에 입력하면 입력된 값이 `handleValueChange()` 함수를 통해 `searchKeyword` 변수에 저장된다. 값이 변경되면 `callApi()` 함수를 다시 호출하여 사용자 목록을 페이지에 다시 렌더링(Rendering)<sup>5)</sup>하게 된다. `filteredComponents()` 함수에서는 매개변수로 받은 `users`를 `map` 함수를 사용하여 **User**로 전달하면서 동시에 `stateRefresh()` 함수도 같이 넘겨주게 된다. **User**는 받은 사용자 정보들을 화면에 그려준다. 또한, **UserDelete**에서 사용자를 삭제할 수 있도록 `stateRefresh()` 함수와 현재의 `id` 값을 **UserDelete**로 넘겨준다.

5) 렌더링(Rendering) : html 요소(element)나 React 요소 등의 코드가 시각적으로 그려지는 것을 의미한다.

UserDelete에서는 사용자의 삭제가 가능하다. 삭제 버튼을 누르게 되면 모달 창(Modal Window)<sup>6)</sup>에서 삭제 여부를 묻는다. 모달 창이 열려있는지 확인하기 위해 open 변수에는 false 값을 기본으로 저장한다. 삭제 버튼을 누르면 handleClickOpen() 함수가 호출되어 open 변수에 true 값을 저장하고 화면에 모달 창을 띄워준다. 모달 창을 닫게 되면 handleClose() 함수가 실행되어 모달 창이 닫히면서 open 변수에 false 값을 저장한다. 모달 창에서 삭제 버튼을 누르면 deleteUser() 함수에서 id를 서버로 전달해 해당 고객의 데이터를 삭제하고 stateRefresh() 함수를 통해 삭제 이후에 새롭게 바뀐 고객 데이터의 목록을 화면에 출력한다.

### 3.5. 데이터베이스

동물 병원 관리 시스템의 데이터베이스 테이블은 USER, ANIMAL, HOSPITAL, APPOINTMENT, CHART로 총 5개로 구성되어 있다. 각 테이블의 구성도는 아래에 작성된 표와 같다.

---

6) 모달 창(Modal Window) : UI에서 자식 윈도우에서 부모 윈도우로 돌아가기 전에 사용자의 상호동작을 요구하는 창을 말한다.

[표 3-3] DB 테이블 구성도 : USER

테이블 이름	열 이름	데이터 형식	null 유무	기본키	외래키	FK 테이블 이름	FK 열 이름	비고
USER	id	INT	NOT NULL	PK				AUTO_ INCREMEN T
	image	VARCHAR (1024)						
	name	VARCHAR (64)						
	birth	VARCHAR (64)						
	created Date	DATE TIME						
	isDeleted	INT						0: default, 1: 삭제된 id
	userid	VARCHAR (64)						
	userpw	VARCHAR (64)						

위의 [표 3-3]은 사용자의 정보를 저장하는 USER 데이터베이스의 테이블 구성도이다. id는 AUTO\_INCREMENT로 자동으로 부여된다. id는 Primary Key로 사용자들을 식별하기 위한 고유 번호로써 사용된다. image는 이미지 파일 이름을 비트 형식으로 받아와 저장한다. 사용자의 이름(name), 생년월일(birth)을 VARCHAR 형태로 저장하고, 가입 날짜(createDate)는 DATETIME 형식으로 저장한다. isDeleted는 아이디의 삭제 여부로 처음 가입할 때는 0이고 관리자가 해당 사용자를 삭제하는 경우에 1로 UPDATE 된다. 사용자가 로그인하였을 때 이용하는 아이디(userid)와 비밀번호(userpw)는 VARCHAR 형식으로 저장한다.

[표 3-4] DB 테이블 구성도 : ANIMAL

테이블 이름	열 이름	데이터 형식	null 유무	기본키	외래키	FK 테이블 이름	FK 열 이름	비고
ANIMAL	animal_num	INT	NOT NULL	PK				AUTO_INCREMENT
	id	INT			FK	USER	id	
	animal_name	VARCHAR (64)						

위의 [표 3-4]는 등록된 동물들의 정보를 나타내는 ANIMAL 데이터베이스 테이블이다. 동물 번호(animal\_num)은 AUTO\_INCREMENT로 자동으로 부여가 되며, Primary Key로 동물들을 식별하는 고유 번호로써 사용된다. id는 해당 동물 주인의 id 번호로, USER 테이블의 id를 참조하는 외래키(Foreign Key)이다. 동물 이름(animal\_name)은 VARCHAR 형태로 저장한다.

[표 3-5] DB 테이블 구성도 : HOSPITAL

테이블 이름	열 이름	데이터 형식	null 유무	기본키	외래키	FK 테이블 이름	FK 열 이름	비고
HOSPITAL	h_num	INT	NOT NULL	PK				AUTO_INCREMENT
	h_name	VARCHAR (64)						

HOSPITAL 데이터베이스 테이블은 현재 시스템에 등록된 동물병원들의 목록을 보여주는 테이블이다. 동물병원 번호(h\_num)는 AUTO\_INCREMENT로 자동

으로 부여되며, Primary Key로 해당 동물병원을 식별하는 고유 번호로써 사용된다. 동물병원 이름(h\_name)은 VARCHAR 형식으로 저장된다.

[표 3-6] DB 테이블 구성도 : APPOINTMENT

테이블 이름	열 이름	데이터 형식	null 유무	기본키	외래키	FK 테이블 이름	FK 열 이름	비고
APPOINT MENT	date_ time	VARCHAR (64)	NOT NULL	PK				
	id	INT			FK	USER	id	
	animal _num	INT			FK	ANIMAL	animal _num	
	h_num	INT			FK	HOSPITAL	h_num	

APPOINTMENT 데이터베이스 테이블은 예약 내역들을 저장하는 테이블이다. 예약 날짜(date\_time)를 VARCHAR 형식으로 저장하고, Primary Key로 예약을 식별하는 고유한 데이터로 사용된다. id는 예약한 사용자의 id로, USER 테이블의 id를 참조하는 외래키이다. 예약한 동물의 번호(anima\_num)는 ANIMAL 테이블의 animal\_num을 참조하는 외래키이다. 예약한 동물병원의 번호(h\_num) 또한 HOSPITAL 테이블의 h\_num을 참조하는 외래키이다.



[표 3-7] DB 테이블 구성도 : CHART

테이블 이름	열 이름	데이터 형식	null 유무	기본키	외래키	FK 테이블 이름	FK 열 이름	비고
CHART	chart_ num	INT	NOT NULL	PK				AUTO_ INCRE MENT
	animal_ num	INT			FK	ANIMAL	animal_ _num	
	chart_ date	VARCHAR (64)						
	chart_ details	VARCHAR (128)						
	h_num	INT			FK	HOSPITAL	h_num	
	classify	INT						0: 진료 1: 접종

CHART 데이터베이스 테이블은 진료 내역들을 저장하는 테이블이다. 차트 번호는 AUTO\_INCREMENT를 이용하여 자동으로 부여되는 번호로, 진료 내역들을 식별하기 위해 고유한 번호로 사용되는 Primary Key이다. 진료받은 동물의 번호(animal\_num)은 ANIMAL 테이블의 animal\_num을 참조하는 외래키이다. 진료받은 날짜(chart\_date)와 진료 내용(chart\_details)은 VARCHAR 형식으로 저장한다. 진료받은 동물병원의 번호(h\_num)는 HOSPITAL 테이블의 h\_num을 참조하는 외래키이다. classify는 진료 종류를 구분 짓는 항목으로 예방접종을 받은 경우에는 1을, 진료를 받은 경우에는 0을 저장한다.

## 제 4 장 동물 병원 관리 시스템 구현 및 테스트

### 4.1. 개발 환경

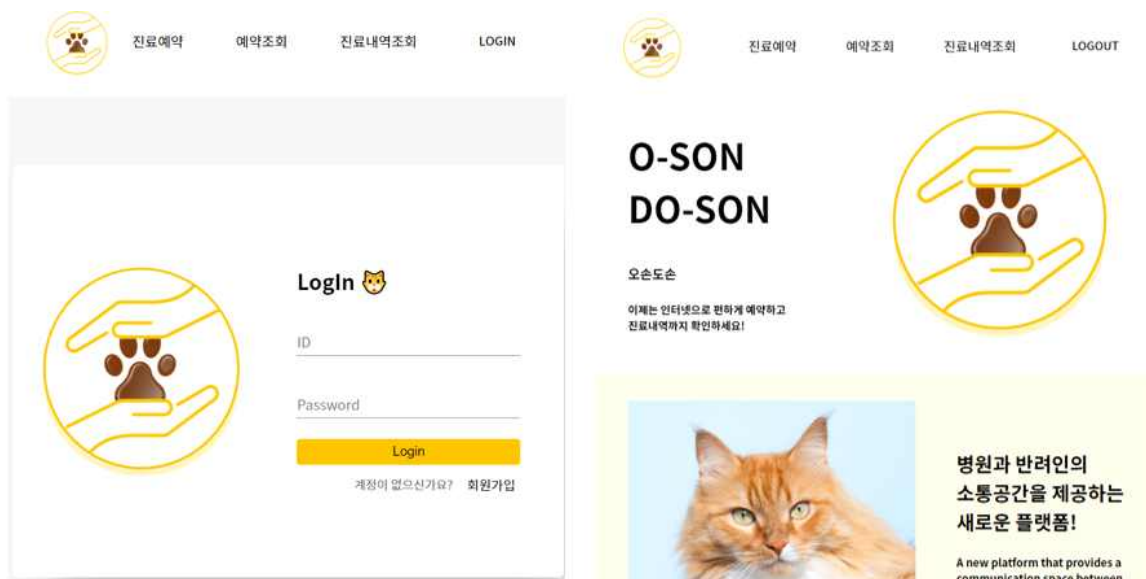
동물 병원 관리 시스템의 개발 환경은 아래의 [표 4-1]과 같다.

[표 4-1] 동물 병원 관리 시스템 개발 환경

구분		기능
H/W	CPU	Intel i5-8265U
	메모리	8GB
S/W	운영체제	Windows 10
	개발 언어	Javascript

### 4.2. 화면 구성

아래의 [그림 4-1]은 로그인 화면과 로그인 성공 시에 이동하는 메인 화면이며, [그림 4-2]는 로그인 화면의 서버 코드이다.



[그림 4-1] 로그인 페이지, 메인 화면

```

let sql = 'SELECT * FROM USER WHERE userid = ?';
let params = [userid, userpw];


connection.query(sql, params[0], (err, row) => {
  if(err) console.log(err)


  if(row.length>0){ //ID가 존재하면
    bcrypt.compare(params[1], row[0].userpw, (err, result) => {
      if(result){
        res.cookie("loginUser", row[0].id).status(200).json({
          success: true,
          id: row[0].id,
          userid: userid
        })
        console.log("성공")
      } else {
        res.json({
          success: false,
          message: '비밀번호가 틀렸습니다.'
        })
        console.log("비밀번호 틀림")
      }
    })
  } else{
    res.json({
      success: false,
      message: '존재하지 않는 id입니다.'
    })
    console.log("id없음")
  }
})
})

```

[그림 4-2] 로그인 기능 서버 코드

로그인 화면의 서버 코드는 MySQL 쿼리문과 파라미터를 데이터베이스로 보내고, 그 결과(id)가 존재한다면 비밀번호를 비교하여 로그인 성공, 실패 여부를 클라이언트로 전달한다.


[진료예약](#)
[예약조회](#)
[진료내역조회](#)
[LOGIN](#)



## REGISTER

프로필 이미지 선택

ID

Password

이름

생년월일

회원가입

[그림 4-3] 회원 가입 페이지 화면

동물 병원 관리 시스템의 회원 가입 페이지의 화면은 위의 [그림 4-3]과 같다. 프로필 이미지를 선택하고 ID, Password, 이름, 생년월일을 입력한 후 추가 버튼을 눌러 가입할 수 있다.

```

let sql1 = "SELECT * FROM USER WHERE userid = ?";
let sql2='INSERT INTO USER VALUES (null, ?, ?, ?, now(), 0, ?, ?)';
let params = [image, name, birth, userid, userpw];

connection.query(sql1, userid, (err, rows) => {
  if(rows.length>0){
    res.json({
      registerSuccess: false,
      message: "이미 존재하는 ID입니다."
    })
  } else{
    bcrypt.hash(params[4], saltRounds, (err, hash) => {
      params[4] = hash
      connection.query(sql2, params,
        (err, rows, fields) => {
          if(err) console.log(err)

          res.json({
            registerSuccess: true,
            message: "회원가입 성공!"
          });
        })
    })
  }
})
})

```


[그림 4-4] 회원 가입 서버 코드

위의 [그림 4-4]는 회원 가입 페이지의 서버 코드이다. 데이터베이스에서 쿼리문 성공·실패 여부를 통해 사용자가 입력한 id가 이미 존재하는 아이디인지 확인한다. 이미 존재하는 아이디가 아니라면 비밀번호를 해시<sup>7)</sup>형식으로 암호화하여 회원 가입 성공 메시지를 클라이언트로 보낸다.


---

7) 해시(hash) : 다양한 길이를 가진 데이터를 고정된 길이를 가진 데이터로 매핑(mapping)한 값

[진료예약](#)[예약조회](#)[진료내역조회](#)[LOGOUT](#)



### 진료 예약 🐶

예약 날짜, 시간  
2021/11/05 14:20 

동물 이름  
강아지1 ▾

병원 이름  
하늘 ▾

[예약하기](#)

[그림 4-5] 진료 예약 페이지 화면

위의 [그림 4-5]는 예약 페이지 화면이다. 원하는 예약 날짜와 시간을 선택할 수 있으며, 진료받기를 원하는 동물의 이름과 진료를 받을 동물병원의 이름을 선택하고 나서 예약 버튼을 눌러 예약할 수 있다.

```

let sql = `INSERT INTO APPOINTMENT(date_time, id, animal_num, h_num)
VALUES ( ?, ?, ?, ? )`;
let params = [date_time, id, animal_num, h_num];

connection.query(sql, params, (err, row) => {
  if(err) console.log(err)

  res.json({
    appointmentSuccess: true,
    message: "예약 성공!"
  });
})

```

[그림 4-6] 예약 서버 코드

위의 [그림 4-6]은 예약 과정의 서버 코드이다. 클라이언트로부터 받은 정보들을 데이터베이스에 저장하고 난 뒤에 다시 예약 성공 메시지를 클라이언트로 보낸다.



[진료예약](#)[예약조회](#)[진료내역조회](#)[LOGOUT](#)

## 예약 조회

예약 날짜	동물 이름	병원 이름
2021-10-20 10:34	그림	희망
2021-10-21 15:34	강아지1	도담
2021-11-01 10:00	그림	희망
2021-11-05 14:20	강아지1	하늘

[그림 4-7] 예약 내역 확인 페이지 화면

위의 [그림 4-7]은 현재까지 예약한 내역을 확인할 수 있는 예약 내역 확인 페이지이다. 현재 로그인한 아이디에 등록되어 있는 모든 동물의 예약 내역들이 출력된다.

```
app.post('/api/appointment/check', (req, res) => {
  let id = req.body.id;
  connection.query(
    "SELECT * FROM APPOINTMENT WHERE id = ?", id,
    (err, row) => {
      res.send(row);
    }
  )
})
```

[그림 4-8] 예약 내역 확인 서버 코드

위의 [그림 4-8]은 예약 내역 확인 서버의 코드이다. 현재 로그인 중인 아이디로 예약된 모든 내역을 반환한다.



차트 번호	진료 날짜	진료 내용	병원 이름	분류(접종, 진료)
3	2021-10-20 10:40	피부병	하늘	진료

[그림 4-9] 진료 내역 확인 페이지 화면

위의 [그림 4-9]는 재까지 받은 진료 내역을 모두 보여주는 화면이다. 진료 내역을 확인할 동물 이름을 선택하면 해당 동물의 진료 날짜, 진료 내용, 진료 병원, 접종/진료 내용을 보여준다.

```
app.post('/api/med_history', (req, res) => {
  let animal_name = req.body.animal_name;
  connection.query(
    `SELECT * from CHART
     WHERE animal_num = ( SELECT animal_num FROM ANIMAL WHERE animal_name = ? )`,
    animal_name,
    (err, row) => {
      res.send(row);
    }
  )
})
```

[그림 4-10] 진료 내역 확인 서버 코드

위의 [그림 4-10]은 진료 내역 확인 서버 코드이다. 사용자가 선택한 동물의 이름을 받으면 그 동물의 진료 내역을 데이터베이스로부터 받아와 클라이언트로 보낸다.

## 제 5 장 결 론

본 논문에서는 반려동물의 진료 정보를 소비자들이 얻기 어렵다는 문제점에 주목하였다. 따라서 반려동물의 건강 정보를 제공하고 소비자들이 보다 반려동물의 건강을 편하게 관리할 수 있도록 한다. 제안된 동물병원 관리 시스템은 병원 사용자와 일반 사용자로 나뉜다. 병원 사용자는 회원 관리페이지에서 회원을 조회 및 삭제할 수 있고, 진료 예약 내역 확인과 진료 내용을 입력하는 것을 가능하도록 한다. 일반 사용자는 회원 가입 및 로그인이 가능하고, 등록된 반려동물의 예약 날짜와 시간, 병원을 선택하여 예약 함으로써, 해당 반려동물의 진료내역을 확인할 수 있도록 한다.

동물병원 관리 시스템은 React와 Node.js로 설계한 시스템으로, 업데이트를 원하는 DOM 요소만을 찾아 업데이트하는 React의 특징으로 인해 빠른 퍼포먼스를 보이며, Node.js의 특징으로 확장성이 있으며, npm을 통해 다양한 모듈을 이용해 개발해 나갈 수 있다는 장점이 있다. 또한, Node.js는 싱글 스레드(Single Thread)로 내부의 Event Loop를 이용하여 비동기 메시지를 처리하여 멀티스레드(Multi Thread)에서 발생하는 문제들에서는 자유롭다. 하지만, 요청을 처리하는 스레드가 하나이므로 대용량 파일을 처리하거나 CPU를 많이 소모하게 되는 작업들을 추가하게 된다면 처리시간이 오래 걸릴 수 있다는 단점이 있다. 하지만 본 웹사이트 구현에 있어서 대용량 파일을 처리하는 작업은 없어 짧은 시간 안에 처리되므로 Node.js의 단점이 나타나는 일은 없었다.

동물병원 관리 시스템은 반려동물을 기르는 일반 사용자들이 반려동물의 진료 내역을 볼 수 있게 함으로써 사용자들이 보다 편리하게 반려동물의 건강을 관리

할 수 있도록 돕는다. 병원 사용자 측에서도 예약을 온라인으로 관리함으로써 진료 및 수술 시간을 더 쉽고 빠르게 확인할 수 있도록 한다. 또한 간단한 진료 내역을 제공하며 소비자들이 반려동물의 건강 정보를 혼동하는 일이 없도록 한다. 타 서비스들은 반려인들이 직접 반려동물의 건강상태를 기록하거나, 실시간으로 간단한 상담을 제공하는 데에 그치는 반면, 제안된 동물 병원 관리 시스템은 병원 방문과 진료 기록들을 볼 수 있어 더 정확한 건강상태가 기록된다는 점에서 더 유용하다.

본 논문의 동물병원 관리 시스템은 병원 자체에서 구축하는 것이 아니기 때문에 외부에서 접속할 수 있도록 배포하는 것이 중요하다. 따라서 구현 과정에서 AWS ec2의 클라우드 서비스를 이용하여 배포할 예정이었다. ec2 서버와의 연결은 완료하였으나, 실제로 접속했을 때 클라이언트와 서버, 데이터베이스의 REST API 통신에 어려움이 있어, 현재 단계에서는 구현하지 못하였다. 향후, 현재의 동물 병원 관리 시스템을 클라우드 서비스를 이용하여 배포할 계획이며, 현재 구현된 기능 외에 병원 관리자의 예약 내역 관리, 진료 내역 입력, 고객의 예약 내역 변경, 취소 등의 기능을 추가할 계획이다. 이 외에도 병원에 방문하지 않아도 궁금한 점을 해결할 수 있도록 고객과 병원의 실시간 상담 채팅 기능도 추가할 예정이다. 또한 서버와 클라이언트의 불필요하게 많은 API의 호출 횟수를 줄이고, 불필요한 코드의 반복 등을 정리하여 현재보다 빠르고 정확하게 동작하도록 수정할 계획이다.

## 참 고 문 헌

- [1] 장덕성, 조현욱, ‘자기 학습 계획을 갖는 웹 기반 학습 시스템의 설계 및 구현’, 정보처리학회논문지. 컴퓨터 및 통신시스템, 11권 4호, pp.297-302, 2004
- [2] 이복음, ‘웹 프론트엔드 프레임워크 및 라이브러리 장단점 연구 - Angular, React, Vue 중심으로 -’, 고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과 석사 학위논문, 2019
- [3] 황정연, ‘펫팸족의 등장과 진화하는 펫보험’, 굿초보 매거진, 2019  
<https://www.goodchobo.com/magazine/MZTY/259>
- [4] 김세형, ‘동물병원서 세모눈 되는 이유 1위..‘과잉진료할까봐’’, notePet, 2020  
[https://www.notepet.co.kr/news/article/article\\_view/?idx=21195](https://www.notepet.co.kr/news/article/article_view/?idx=21195)
- [5] 이효정, ‘반려동물 전용 앱 추천 8’, Noblesse, 2020  
<https://www.noblesse.com/home/news/magazine/detail.php?no=9509>
- [6] Node.js: 비동기 프로그래밍 이해  
<https://www.nextree.co.kr/p7292/>
- [7] JavaScript - MDN Web Docs - Mozilla  
<https://developer.mozilla.org/ko/docs/Web/JavaScript>
- [8] The state of javascript 2020 - Front-end Framework Survey  
<https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>
- [9] The state of javascript 2020 - Other-Tools Survey  
<https://2020.stateofjs.com/en-US/other-tools/#runtimes>
- [10] Stack Overflow Developer Survey 2021  
<https://insights.stackoverflow.com/survey/2021#most-popular-technologies-webf>

rame-prof

[11] 자습서: React 시작하기

<https://ko.reactjs.org/tutorial/tutorial.html>

[12] React와 Node.js로 만드는 고객 관리 시스템 개발 강좌

[https://ndb796.tistory.com/category/React와 Node.js로 만드는 고객 관리 시스템 개발 강좌](https://ndb796.tistory.com/category/React와%20Node.js로%20만드는%20고객%20관리%20시스템%20개발%20강좌)

[13] Building Data Science Web Application with React, NodeJS, and MySQL

<https://www.byperth.com/2018/04/19/guide-building-data-science-web-application-with-react-nodejs-and-mysql/>

## ABSTRACT

Recently, as the single-person generation and aging of Korean society progresses rapidly, individual interest and demand for companion animals are increasing to the extent that a new word called ‘PetFam(Pet+Family)’ has appeared. As a result, the size of the companion animal market related to the health of companion animals as well as food, care products, and care services is increasing remarkably.

In this paper, these animal hospital management sites are implemented using React, a Javascript library that has recently been widely used, and Node.js, a software platform used to develop scalable network applications as Javascript runtime.

In this paper, hospital users can manage members and treatment details, and general users can register and log in, make reservations for veterinary hospital treatment, and check treatment details. It was described, and the advantages and disadvantages of implementing it using React and Node.js were analyzed.

This thesis focuses on the inconvenience experienced by companion animals in relation to the veterinary hospital service. This is a study on the establishment of a web-based animal hospital management system necessary for the establishment of a care service network and the operation of



management services for pets through dedicated management of pets and linkage with local animal hospitals.

Hospital users can inquire and delete membership, check medical reservation details, and enter medical treatment details on the membership management page, and general users can sign up and log in, make medical reservations for registered pets, and check medical treatment details. This helps pets see and easily manage their health care details at a glance. Hospitals also manage reservations online, making it easier and faster to check the time of treatment and surgery.

Keywords: Animal hospital management system, React, Node.js

## 감사의 글

설렘과 꿈을 안고 학교 정문을 들어서던 게 엇그제 같은데, 벌써 4년이라는 시간이 흘렀습니다. 많은 분들의 아낌없는 도움으로 비로소 제가 그 꿈에 첫 발자국을 뒀을 수 있게 되었습니다.

컴퓨터공학과와 모든 교수님께 감사의 인사를 올립니다. 바쁘신 와중에도 학위논문의 지도 교수님을 맡아주시고 좋은 조언을 해주신 천인국 교수님께 감사드립니다. 또한 캡스톤 디자인 수업을 통해 논문 작성의 방향을 잡아주신 이상정 교수님 감사드립니다. 4년 동안 항상 성심성의껏 지도해주시고, 많은 조언 해주신 하상호 교수님, 홍인식 교수님, 이해각 교수님, 남윤영 교수님께 감사드립니다. 교수님의 지도를 통해 제게 부족한 점이 무엇이고 어떤 방향으로 나아가야 할지를 알 수 있었던 것 같습니다.

이 논문 작성을 함께한 구민주 언니에게도 감사를 표하고 싶습니다. 혼자였다면 버거웠을 논문 작성을 덕분에 잘 끝마칠 수 있었습니다. 또한 4년 동안 학과 생활을 함께해준 친구들(임주현, 이승은)에게도 감사드립니다.

자신감을 잃고 고민할 때마다 많은 조언을 해주신 사랑하는 부모님께 감사드립니다. 앞으로도 지금과 같이 열심히 해서 부모님께서 주신 은혜에 보답할 수 있는 딸이 되도록 하겠습니다.

이 외에도 여기에 미처 적지 못한 많은 분께 감사드립니다. 여러분이 있어 제가 있었습니다. 항상 감사하고 사랑합니다.

컴퓨터 공학도의 꿈을 가지고 입학하던 날이 아직 생생한데, 어느덧 4년이란 시간이 흘러 졸업을 목전에 두게 되었습니다. 많은 분들의 관심어린 조언과 도움으로 이제는 더 큰 꿈을 꾸며 세상으로 나갈 수 있게 되었습니다.

컴퓨터공학과와 모든 교수님께 감사의 인사를 올립니다. 바쁘신 와중에 제 학위 논문의 지도교수님을 맡아주시고 조언해 주신 천인국 교수님께 감사드립니다. 또한, 캡스톤 디자인 수업을 통하여 논문의 주제선정부터 작성 방향을 지도하여 주신 이상정 교수님 감사드립니다. 회로이론에 흥미를 갖도록 도와주신 홍인식 교수님 감사드립니다. 항상 열정을 가지고 수업해 주신 하상호 교수님, 이해각 교수님, 남윤영 교수님께 감사드립니다. 교수님의 지도를 통해 부족한 점은 보완하고 나아갈 방향을 정할 수 있었습니다.

이 논문 작성을 함께한 노서현 학생에게도 감사를 포함합니다. 혼자서는 쓰기 힘들었을 논문을 덕분에 잘 마무리할 수 있었습니다. 또한, 4년 동안 즐거운 학교 생활을 할 수 있도록 함께해준 동기들(임주현, 이승은)에게도 감사드립니다.

길을 잃고 방황해도 끝없는 사랑으로 보듬고 올바르게 이끌어주신 사랑하는 부모님께 감사드립니다. 이끌어 주신대로 올바르게 끝은 사람으로 살아가겠습니다.

이 외에도 미처 적지 못한 주변의 많은 분들께 감사드립니다. 여러분이 있어 제가 오늘날의 제가 있었습니다. 앞으로 모두 행복하길 바라며 항상 감사하고 사랑합니다.

2021년 11월 30일

노서현

구민주