

졸업작품 최종보고서

컴퓨터소프트웨어공학과

2023. 00. 00.

프로젝트 명	데뷔탕트 하우스
팀명	세르비아

구분	학번	이름
팀장		조은서
팀원		김가연
팀원		배채연
팀원		이무현
지도교수		

목 차

I. 프로젝트 개요	2
1. 개발 배경 및 필요성	2
2. 개발 내용	3
3. 팀 구성 및 역할	5
II. 관련 작품 분석	5
1. 관련 작품 설명	5
2. 차별성 및 우수성	7
III. 관련 기술 연구(개인별)	8
1. 라즈베리파이를 사용한 하드웨어 기술	8
2. DB 연동과 서버	10
3. 지도 API	13
4. 회원 가입 및 로그인 API	15
IV. 시스템 설계	16
1. 기능 설명	16
2. 개발 환경, 도구 및 언어	16
3. 기능 설계	17
3.1 세부 기능(Use Case 시나리오)	17
3.2 UI/UX 설계	20
3.3 데이터베이스 설계	24
3.4 통신 설계	26
3.5 디바이스 제어	27
V. 구현	28
1. 구현 내용	28
2. 실제 구현 UI/UX	30
VI. 결론	33
1. 문제점 및 개선 방안	33
2. 개인별 소감	34

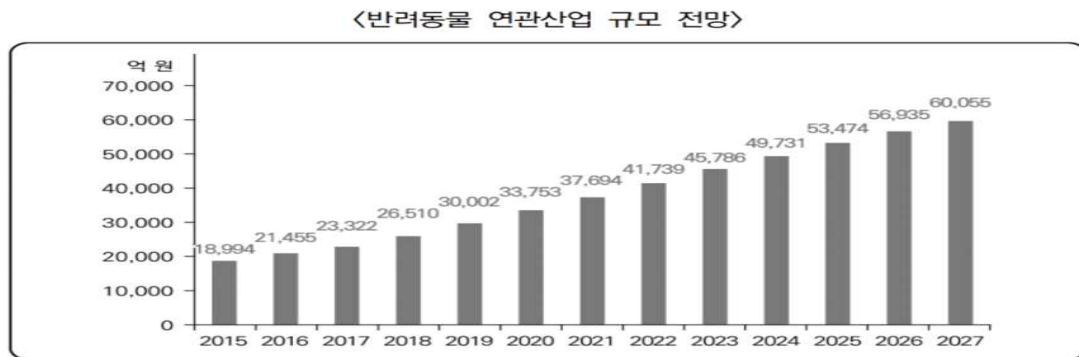
VII. 참고자료	35
VIII. 부록	36
1. 개인별 자기평가(수행내용 및 문제해결 내용)	36
1.1 김가연 20202871	36
1.2 김민지 20190659	37
2. 프로그램 소스	38

I. 프로젝트 개요

1. 개발 배경 및 필요성

국내 반려동물 인구 1500만 시대가 도래하고 1인 가구와 반려동물을 가족처럼 생각하는 펫팸족(Pet+Family)이 빠르게 늘면서 관련 사업이 점차 커지고 있다. KB 경영연구소가 발표한 '2021 한국 반려동물 보고서'에 따르면 반려동물을 기르는 '반려 가구'는 604만 가구로 한국 전체 가구의 약 30%를 차지하고 있다. 그리고 향후 반려동물을 양육하기를 희망하는 가구가 늘어나는 추세이다. 따라서 반려 가구는 더욱 늘어날 전망이다.

또한 한국농촌경제연구원이 발표한 반려동물 시장 규모는 2009년 9000억 원에서 2015년 1조 9000억, 작년에는 3조 4000억 원으로 증가했다. 2027년까지는 6조 원대 성장을 이룰 것이라고 전망하며 연평균 10% 이상 성장 중이다.



출처: 한국농촌경제연구원

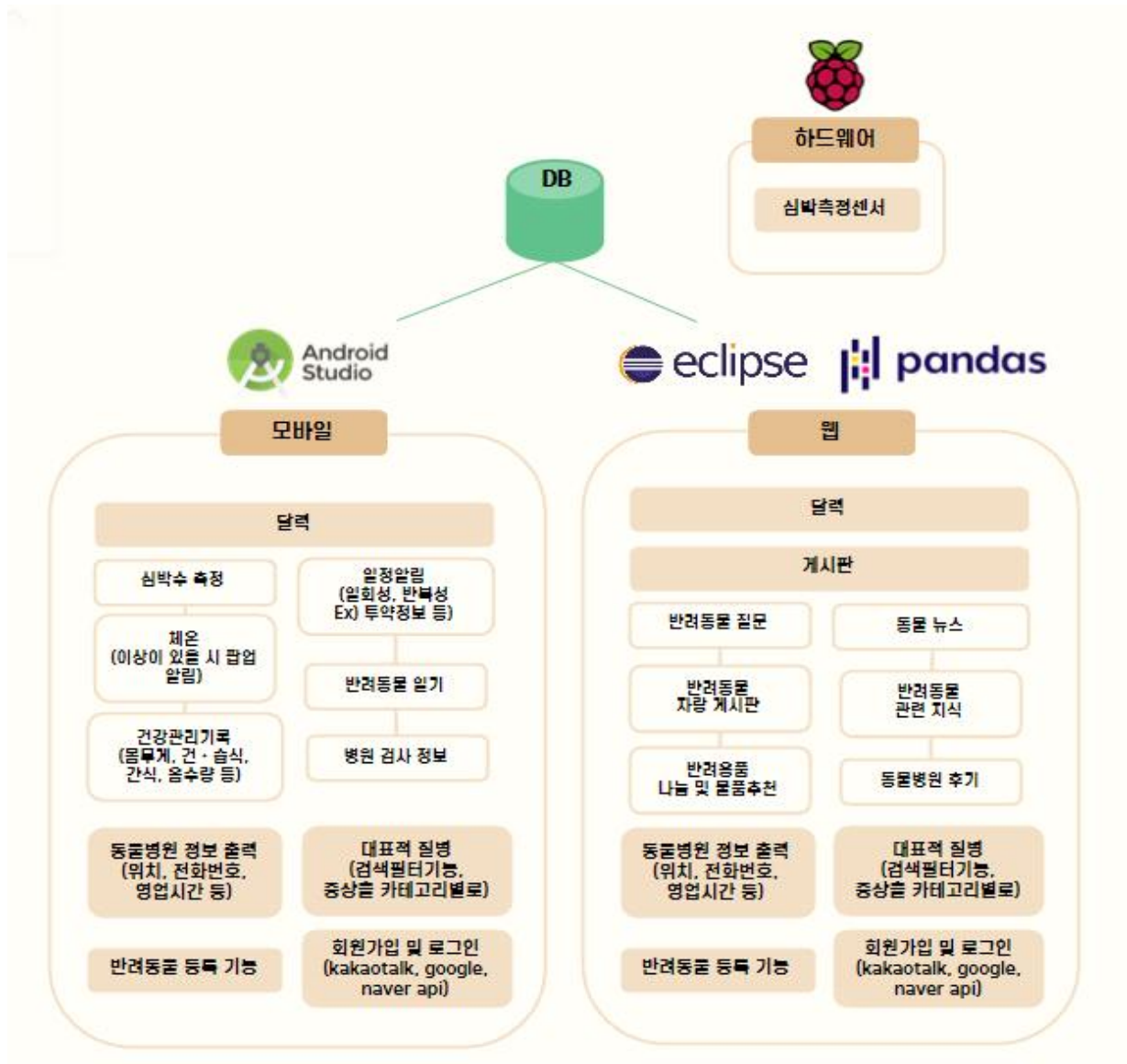
[그림 2] 반려동물 연관산업 규모 전망

펫팸족이라는 말이 만들어질 만큼 우리 사회에서 반려동물은 이미 가족 내지는 삶의 동반자로까지 여겨지고 있고 이 같은 추세는 더욱 강화될 것이다.

시장 규모가 커지는 만큼 반려동물 건강에 대한 염려 또한 커지고 있다. 반려동물이 아플 때 증상을 검색하여 흠어진 정보를 모으고 증상과 대조하는 시간이 많이 소요되고, 병원 방문 후 정해진 투약 시간을 잊지 않아야 하는 등 반려동물 건강 관리는 번거로운 요소가 많다. 기존의 반려동물 헬스케어 앱은 보호자가 직접 손으로 측정하고 기록하여 정확도가 떨어질 수 있다. 또한 질병의 증상이 발견될 시 대처방안 등이 명확하게 나타나지 않거나 아예 없는 경우가 많으며 주변 동물 병원의 위치 또한 알려주지 않는다. 이러한 점들을 개선하고 보완하고자 반려동물 헬스케어 서비스를 개발하고자 한다.

2. 개발 내용

2-1. 시스템 구성도



[그림 3] 시스템 구성도



[그림 4] 구현 센서 예시

2-2. 서비스 시나리오

연지수 씨는 12살 고양이 한 마리와 3살 강아지 한 마리와 함께 살고 있다. 올해로 12살이 된 노령의 고양이가 걱정되어 연지수 씨는 며칠 전 수시로 심장박동을 체크할 수 있는 pit-a-pet 심장박동 측정기를 구매하게 되었다. 측정기를 구매 후 측정기와 연동되는 앱을 설치하여 기록할 반려동물로 고양이 한 마리와, 강아지 한 마리를 등록했다. 반려동물을 등록하니 매일 달력에 반려동물 상태를 기록할 수 있는 기능이 있었다.

등록한 반려동물 중 고양이를 골라 달력에 기록을 시작했다. 하루 동안 먹은 사료, 간식의 양, 음수량을 기록하고 고양이가 자고 있을 때 심장박동 측정기를 고양이의 발바닥에 접촉시켜 심장박동을 측정했다. 측정기로 측정을 하니 자동으로 데이터가 연동되어 기록되어 있었다. 그 외에도 고양이 화장실을 치워 주며 대변과 소변의 상태와 횟수 또한 기록하고, 오늘 헤어볼을 토해서 구토 카테고리에 기록했다. 그리고 오늘 하루 동안 고양이와 있었던 재밌는 일들을 일기 카테고리에 적었다. 다음으로 등록한 반려동물 중 강아지를 골라 강아지 달력에 기록을 했다. 고양이와 마찬가지로 하루 동안 먹은 양과 심박수, 대소변도 기록했다.

그런데 오늘 강아지가 기침을 하고 콧물이 나서 이상 증상에 기침과 콧물 흘림을 기록하고 강아지가 어디가 아픈 것인지 검색하기 위해 질병 카테고리에 가서 질병 증상 검색 필터를 이용하여 기침과 콧물 증상을 필터에 적용 후 질병을 검색하니 감기 등 기침, 콧물 증상이 포함되어 있는 질병들이 여러 개 나왔다. 그중 감기가 가장 유력한 질병인 것 같아 다음 날 연지수 씨는 동물 병원에 가야겠다고 생각했다.

그래서 병원 검색 카테고리에 집 근처에 있는 동물 병원을 찾아보고 전화로 진료를 예약 후 앱에 일정 카테고리에 병원 방문 일정을 시간과 함께 기록하여 알람이 울리게 설정했다. 다음 날 병원 예약 시간에 맞춰 울린 알람을 보고 강아지를 데리고 동물 병원에서 진료를 본 후 감기라는 진단을 받아 약을 처방받았다. 하루에 두 번 3일 동안 투약하는 약이라 투약 기록 카테고리에 주기를 반복성으로 설정하고 오전 10시와 오후 10시 두 번 투약 알람을 설정하여 투약하는 것을 잊지 않고 먹일 수 있게 되었다.

강아지를 투약하는 과정에서 약을 먹지 않아 잘 먹일 수 있는 방법이 있을까 고민하던 중 여러 사람의 의견을 들어 보면 좋을 것 같아 pit-a-pet 웹 사이트에 있는 게시판 기능을 사용하여 질문 글을 올렸다. 몇 분 뒤, 여러 명의 댓글이 달려 그중 연지수 씨의 강아지가 제일 거부감이 없을 방법으로 투약할 수 있게 되었다.

연지수 씨는 매일 반려동물의 상태를 pit-a-pet 앱과 웹에 기록하며 반려동물의 사소한 변화도 한눈에 알아볼 수 있어 걱정이 덜어졌다. 특히 고양이의 심장박동 수를 매일 측정하여 노령의 고양이에게 자주 생기는 심근비대증을 초기에 알 수 있어 다행이라고 생각했다.

3. 팀 구성 및 역할

구분	학번	이름	담당 역할 및 개발 분야
팀장	20202871	김가연	<ul style="list-style-type: none"> • Raspberry Pi에서 심박센서 프로그램 제작 • 아두이노와 라즈베리파이 플라스크 서버를 이용한 연동 • Android환경에서의 UI,UX 구성과 기능 제작 • DB(서버)연동 및 기능 구현 • React 언어를 사용한 웹 프론트엔드, 백엔드 • React-Native 언어를 사용한 앱 프론트엔드, 백엔드 • Firebase 서버 개발환경 구축 및 연동
팀원	20190659	김민지	<ul style="list-style-type: none"> • Android환경에서의 UI,UX 구성과 기능 제작 • DB(서버)연동 및 기능 구현 • React 언어를 사용한 웹 프론트엔드, 반응형 • React-Native 언어를 사용한 앱 프론트엔드, 백엔드 • Firebase 서버 개발환경 구축 및 연동

II. 관련 작품 분석

1. 관련 작품 설명

1) 갤럭시 핏(Galaxy Fit)



[그림 5] 갤럭시 핏

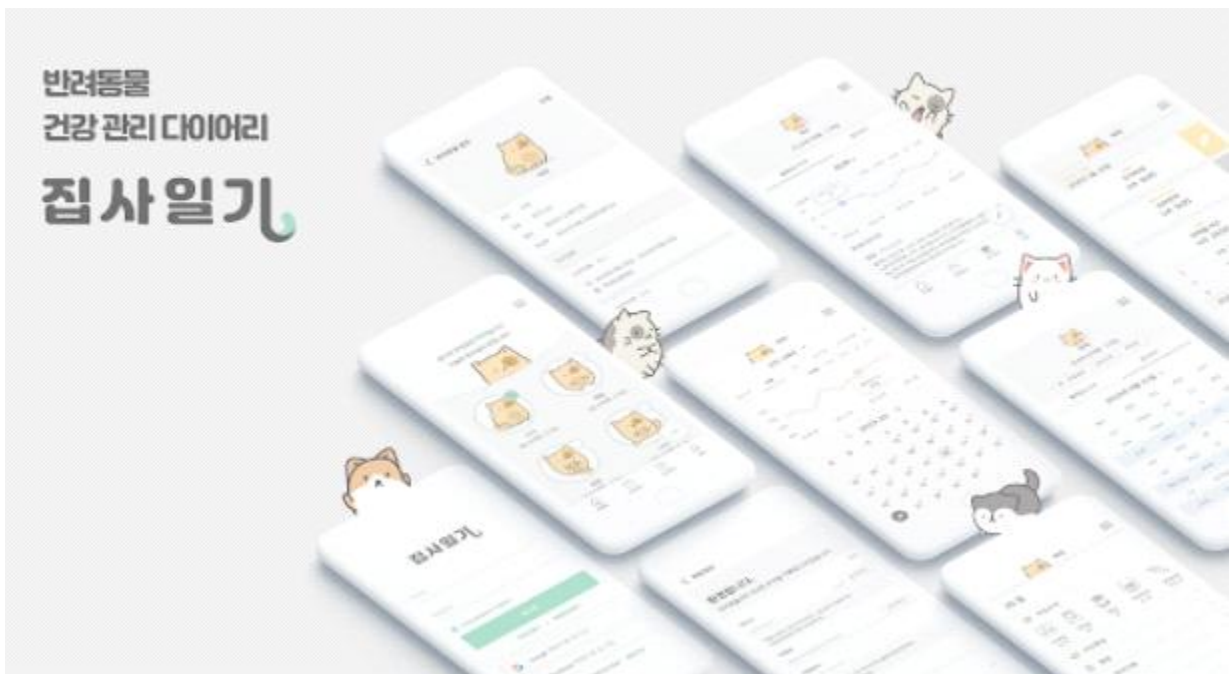
삼성전자에서 만든 웨어러블 스마트 밴드 시리즈로써 갤럭시 핏과 갤럭시 핏²가 있다. 2019년 2월 20일 삼성 언팩에서 공개된 스마트 밴드로, 삼성 기어 핏, 기어 핏2, 기어 핏2 프로의 후속작이고, 삼성 헬스 앱과 연동된다.

메모리는 2MB RAM과 32MB 내장 메모리가 탑재되어 있으며, 블루투스 5.0을 지원한다. 갤럭시 웨어러블을 설치하여 사용할 수 있으며, 삼성전자 자체 빌드 웨어러블 전용 RTOS를 운영체제로 사용한다.

갤럭시 핏의 특징

- 심박 수 센서와 자이로센서가 탑재되어 삼성 헬스와의 연동이 가능하다,
- 연동된 스마트폰으로 수신된 카카오톡은 상대의 이름을 포함하여 간단한 문자 확인 정도는 가능하다.
- 무선 충전만을 지원하며 배터리 교체가 불가능하다.

2) 집사일기



[그림 6] 집사일기

반려동물 건강 관리 일기로써 반려동물의 만성질환 진단을 받고 돌보는 방법을 모르는 주인들을 위해 건강 정보를 매일 기록하여 체계적인 관리를 할 수 있도록 만든 앱이다. 깜빡하기 쉬운 투약 알림부터, 동물 병원 방문 일정, 반려동물과 관련된 일정을 등록하면 잊지 않도록 알람을 보내주고, 진단명을 등록하면 해당 질병에 대한 간략한 설명과 관리에 대한 가이드가 제공된다. 달력에는 몸무게, 호흡수, 심박수, 체온 등을 직접 채서 기록할 수 있고, 일기 기록이 가능하다.

집사 일기의 특징

(1) 건강일기

- 식사량, 음수량, 대소변, 이상 증상, 호흡수, 심박수, 체온 등 매일의 건강을 기록한다.

- 약 투약률, 피하수액 등 매일의 관리를 기록한다.
- 권장 음수량, 사료량 등 건강 관리에 대한 정보를 알려준다.
- 사진과 함께 매일의 관리를 기록한다. (그림일기 + 나중에 회의하기)
- 기록한 내역들을 카테고리/기간 별로 모아보거나 키워드로 검색할 수 있다.

(2) 일정 일기

- 투약 알람을 등록하면 투약 시간마다 알람을 보내준다.
- 예방접종, 구충제 복용 등 반복적인 일정을 등록해 알림을 받을 수 있다.
- 병원 방문, 영양제, 산책 목욕 등 반려동물의 주요 일정을 알림으로 알려준다.

2. 차별성 및 우수성

정확성 - 심박수 측정 센서를 사용한 정확한 진단

기존의 직접 손으로 측정하는 방법, 측정 대상이 흥분하고 긴장한 상태로 병원에서 측정하는 방법의 부정확성 문제가 발생한다.

이 문제를 대응하여, 심박수 측정 센서를 사용한 하드웨어를 제작하여 정확한 진단이 가능하다. 시중에 사람의 심박수 측정 장치만 출시되어 있으며 반려동물을 위한 심박수 장치는 없다. 또한 사람의 경우 심박수를 측정할 때 손목에 밴드 혹은 시계 형식으로 착용해야 했는데, 반려동물은 착용 시 불편함을 겪을 수 있고 이를 우려하는 보호자가 있을 수 있어 직접 착용하지 않고 발바닥에 하드웨어를 맞대는 방식으로 차별성을 두었다.

편리성 - 동물 병원 정보 제공과 데이터 분석을 통한 질병 검색 필터 및 정보

기존의 반려동물 건강을 기록하는 앱에서는 동물 병원 정보를 제공하지 않는 경우가 대다수이다. ‘pit-a-pet’ 애플리케이션과 웹은 동물 병원의 위치, 전화번호, 영업시간 등의 기본 정보와 위급 상황 시 찾아갈 수 있는 24시 동물 병원의 정보를 제공하여 한 플랫폼에서 정보를 파악할 수 있는 편리성이 있다.

반려동물의 이상 증상 발견 시, 포털사이트에 증상을 검색하고 대조하는 시간을 줄일 수 있다. 데이터 파일을 사용하여 데이터를 분석한 검색 필터 기능으로 질병에 대한 정보를 제공한다. 사용자가 반려동물의 질병 지식의 폭이 넓어져 관련 상황 발생 시 신속하고 정확하게 대처할 수 있는 교육 효과가 있다.

심장 박동 센서를 사용하여 가정 내에서도 정확한 진단이 가능하고, 기존의 반려동물 건강 앱에서의 기능을 개선하고 없는 기능을 추가하였기 때문에 우수성이 있다.

III. 관련 기술 연구

1. 라즈베리파이를 사용한 하드웨어 기술

1.1 라즈베리파이 심박센서 기술

학번	작성자	관련 기술
20202871	김가연	라즈베리파이 심박센서 기술

[그림 7]은 심장박동 센서이다. 해당 센서는 LED에서 빛을 쏘고 조도센서에서 반사되어 나오는 빛의 양을 측정하여 심박을 측정하는 센서이다. 심박을 측정하는 원리는 빛을 쏘고 조도센서에서 반사되는 값을 받아 LED에서는 지속적으로 빛을 쏘아주고, 조도센서에서는 모세혈관에 반사되어 돌아오는 빛의 양을 측정하게 된다. 이 값이 혈류량이 많아지면 빛의 반사량이 줄고 혈류량이 적으면 빛의 반사량이 늘어나게 된다. 따라서 혈류량이



[그림 7] 심박 센서

심장박동에 따라 변하는 원리를 이용하여 심박수를 측정하는 것이다. 해당 모듈을 사용하기 위해서는 라이브러리를 받는다. 분당 심장박동 수, 심전도 그래프를 실행하는 코드는 아래와 같다.

```
import spidev
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import animation
import time
spi=spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz=500000
def read_spi_adc adcChannel):
    adcValue=0
    buff = spi.xfer2([1,(8 + adcChannel) << 4, 0])
    adcValue = ((buff[1]&3)<<8)+buff[2]
    return adcValue
x_num=100
x=np.arange(x_num)
y=np.empty(x_num)
y[:]=np.NaN
fig=plt.figure(figsize=[6.5,3])
```

```

ax = plt.axes()
ax = plt.axes(xlim=(0,x_num),ylim=(750,820))
line,=ax.plot(x,y,lw=2)

def init() :
    line.set_data([],[])
    return line,

def animate(i) :
    global y
    y[i]= read_spi_adc(0)
    line.set_data(x,y)
    if i==x_num-1 :
        y[:]=np.NaN
        line.set_data(x,y)
    return line,

ani=animation.FuncAnimation(fig,animate,init_func=init,frames=x_num, interval=1)
plt.show()

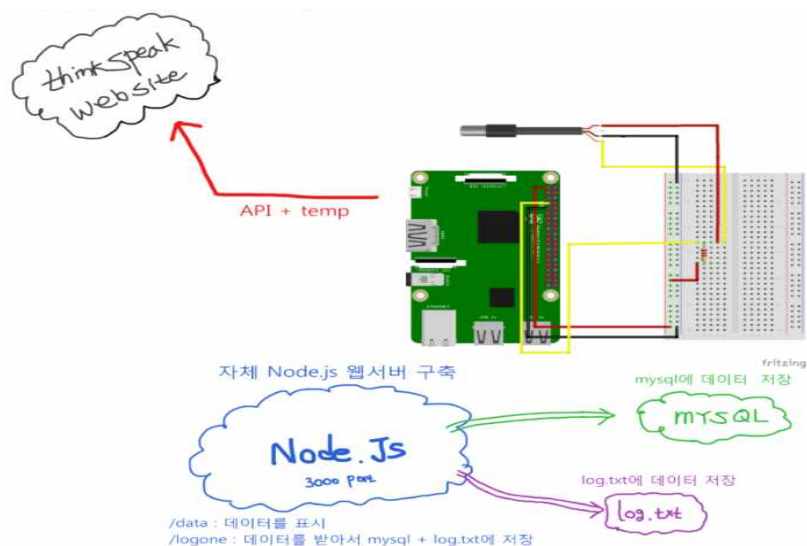
```

[코드 1] 심박 센서

1.2 라즈베리파이 데이터 연동

학번	작성자	관련 기술
20202871	김가연	라즈베리파이 데이터 연동

심장박동 센서의 값을 읽은 다음 Thingspeak 사이트를 사용해 데이터를 웹으로 전송하면 실시간으로 그래프가 웹사이트에 그려지도록 작성하고 추가로 Node.js 웹서버를 구동한 다음 이 웹서버에도 데이터 값을 전송하는 코드를 작성할 것이다. 또한 Node.js는 MySQL에 데이터를 입력하고 log.txt 파일을 자체적으로 생성하도록 작성할 것이다.



[그림 8] 라즈베리파이 데이터 연동

2. DB 연동과 서버

2.1 React와 Firebase 연동

학번	작성자	관련 기술
201900659	김민지	React와 Firebase 연동 기술 React, node.js, JavaScript, Firebase

React와 Firebase를 연동하기 위해선 먼저 cmd 창에서 React 프로젝트를 생성한다. 그리고 Firebase 사이트에 들어가 마찬가지로 프로젝트를 생성한다. Firebase에서 요구하는 값을 입력하면 개인 인증키인 firebase SDK 파일을 받을 수 있다. 인증키 파일을 자바스크립트 파일을 생성하여 내용을 추가한다. 위의 인증키는 보안상의 문제가 발생할 수 있으므로, 프로젝트 루트 폴더의 .env 파일에 보관하여 보안 문제를 처리한다. .env 파일에서 환경변수의 맨 앞의 REACT_APP_을 입력해 주어야 리액트가 인식해 줄 수 있다. 이 과정을 실행하지 않으면 에러가 날 수 있으니 주의해야 한다. 리액트와 파이어베이스의 연동을 확인해 보기 위해 Visual Studio Code의 터미널창에 npm run start를 입력한다.

아래의 [코드 2]는 React와 Firebase가 성공적으로 연동하고 정상적으로 작동하는지 확인하는 예제 코드이다.

```
import firebase from 'firebase/compat/app';
import 'firebase/compat/analytics'
import 'firebase/compat/auth';
import 'firebase/compat/firestore';

const firebaseConfig = {
  apiKey: process.env.REACT_APP_FIREBASE_APIKEY,
  authDomain: process.env.REACT_APP_FIREBASE_AUTH_DOMAIN,
  databaseURL: process.env.REACT_APP_FIREBASE_DATA_BASEURL,
  projectId: process.env.REACT_APP_FIREBASE_PROJECTID,
  storageBucket: process.env.REACT_APP_FIREBASE_STORAGEBUCKET,
  messagingSenderId: process.env.REACT_APP_FIREBASE_MESSAGINGSENDERID,
  appId: process.env.REACT_APP_FIREBASE_APPID,
  measurementId: process.env.REACT_APP_FIREBASE_MEASUREMENTID
};

export default firebase.initializeApp(firebaseConfig);
```

[코드 2] Firebase

2.1 React와 Firestore 연동

학번	작성자	관련 기술
201900659	김민지	React와 Firestore 연동 기술 React, node.js, JavaScript, Firebase, Firestore

Firebase를 연동할 프로젝트를 열고 터미널창에 `npm install firebase` 명령어를 입력한다. 앱에서 Firebase를 초기화하고 Firebase 앱 객체를 생성한다. config를 가져오기 위한 자바스크립트 파일을 생성한 후 Firebase SDK를 추가한다. 사용할 파일에 데이터베이스인 firestore를 import 한 후에 firestore가 연동되었는지 확인하기 위해 콘솔창으로 확인한다.

다음은 firebase 서버가 구축되었다고 가정할 때, React와 데이터베이스인 Firestore가 성공적으로 연동 후 데이터 입력, 데이터 출력 그리고 삭제할 수 있는 기능을 설명하는 예제 코드이다.

```
//데이터 입력 코드
import {
  collection,
  deleteField,
  doc,
  getDoc,
  setDoc,
  Timestamp,
  updateDoc,
} from "@firebase/firestore";
//import는 필수이다.

//현재 로그인된 user의 기본적인 데이터는 authService.currentUser에서 참조 가능하다.
const ref = doc(dbService, "User", authService.currentUser.uid);
//최상위 User > UID폴더내에 값을 넣고자 하는 경우이다.

setDoc(
  ref,
  { [nickname]: { imageUrl: imageUrl, fullName: fullName } },
  { merge: true }
);
//setDoc을 사용해서 앞서 선언한 doc와 원하는 데이터를 입력해주면된다.
//Firestore에 object 형식의 데이터를 삽입할 수 있으며, 위와 같이 입력하면
//nickname이 키 값, 이하 부분이 value 값이 된다.

//Firestore의 setDoc은 해당 doc의 모든 요소를 입력하는 변수로 대체해버리는데,
//merge : true 옵션 혹은 updateDoc를 사용시에 기존의 데이터에 합치게 된다.
```

//데이터 출력 코드

//깊이가 있는 곳에 데이터를 삽입하고싶은 경우는 아래와 같이 path를 /를 사용해 만들어주면된다.
//결과적으로 아래 코드는 Notice/1128/1/title doc의 모든 필드를 불러오는 동작을 한다.

```
const ref = doc(dbService, "Notice/1128/1", title);
let res = await getDoc(ref);
//Firestore의 모든 함수는 비동기 함수이므로 await&async 혹은 then을 적절히 사용하자.
//get은 데이터가 없을 경우 null을 반환하며, 가져온 데이터는 아래와 같이 res.data()의 형식으로 뽑아 낼 수 있다.
return res.data() ?? {};
```

//데이터 삭제 코드

//데이터 삭제시는 아래와 같이 update함수와 함께 (merge : true를 사용해도 동일함)
//해당 Field에 deleteField() 함수를 입력해주면 된다.

```
const ref = doc(dbService, "User", authService.currentUser.uid);
updateDoc(ref, { [nickname]: deleteField() });
```

[코드 3] Firestore

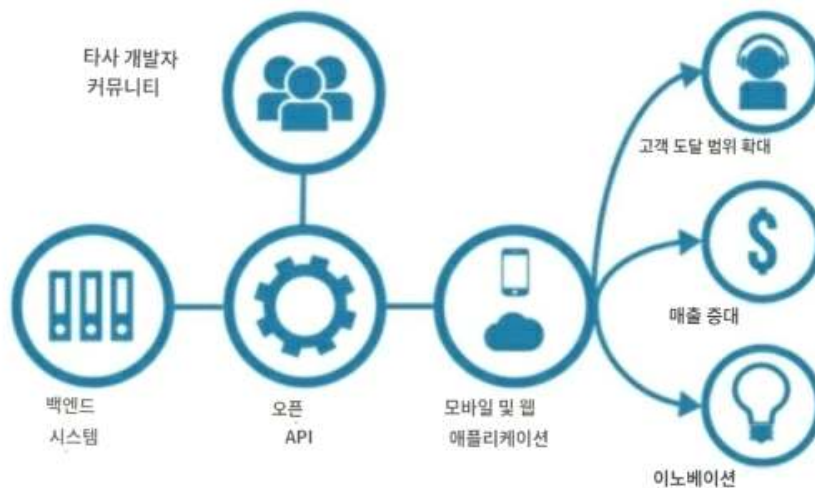
[코드 3]은 데이터베이스 테이블을 개발한 후 기능에 필요한 요소들을 import 한다. Firebase에서 발급받은 SDK의 정보와 생성한 config 파일의 정보가 일치하는지 확인하고 일치하지 않을 때는 연동이 되지 않을 수 있으니 주의해야 한다. Firebase의 데이터베이스인 Firestore의 모든 함수는 비동기 함수이므로 동기와 비동기를 적절히 활용하여 데이터를 출력할 수 있다.

웹의 기능에 따라 React를 사용하여 함수를 통해 필요한 기능을 구현하고, 데이터베이스 테이블을 개발하여 이를 바탕으로 Firestore를 사용할 때 필요한 정보를 입력할 것이다.

3. 지도 API(카카오)

학번	작성자	관련 기술
20202871	김가연	Kakao Maps API, kakao developers, Google Developers Console

공개 API는 개발자라면 누구나 사용할 수 있도록 OS나 시스템, 앱, 라이브러리 등을 활용하여 응용 프로그램을 작성할 수 있게 하는 다양한 인터페이스를 의미한다. 사유 응용 소프트웨어나 웹 서비스의 프로그래밍적인 권한을 제공하는 공개 API의 대표적인 예시로는 네이버 지도, 구글맵 등에서 시도되고 있으며 누구나 접근하여 사용할 수 있다는 장점이 있다.



[그림 10] 공개 API

그중에서도 이번 프로젝트에 사용할 API는 지도 API(Maps API)와 장소 검색 API이다. 구글은 단계적인 유료화를 통해 2018년 구글 맵스 API를 무료 사용권 제도를 사용한 전면 유료화 방침을 사용하였다. 이처럼 공개된 API 일지라도 데이터 사용량에 따라 비용을 지불해야 하는 경우가 있거나 완전히 무료일지라도 사용자가 회원가입을 통한 신원 확인 후 서비스 제공자로부터 공개키(또는 사용 권한 토큰)를 별도로 발급받아 오픈 API를 사용하도록 장려함으로써 무분별한 데이터 남용을 막는 사례가 늘고 있다.



[그림 11] 키 발급 과정(kakao Maps API)

3.1 지도 API

지도 API 활용 예시	동물병원 검색 API 예시
	

키워드로 장소를 검색하고 목록과 마커로 결과를 표출해 주는 API는 Kakao Maps API를 사용할 것이다. 목록과 마커에 마우스 오버를 하게 되면 해당 장소를 표시하고 있는 마커에 인포윈도우로 장소명을 표시한다.

예제 코드

```

1 <div class="map_wrap">
2   <div id="map" style="width:100%;height:100%;position:relative;overflow:hidden;"></div>
3
4   <div id="menu_wrap" class="bg_white">
5     <div class="option">
6       <div>
7         <form onsubmit="searchPlaces(); return false;">
8           키워드 : <input type="text" value="동물병원" id="keyword" size="15">
9           <button type="submit">검색하기</button>
10        </form>
11      </div>
12    </div>
13    <hr>
14    <ul id="placeslist"></ul>
15    <div id="pagination"></div>
16  </div>
17</div>
18<script>
19// 마커를 얻을 배열입니다
20var markers = [];
21
22var mapContainer = document.getElementById("map"), // 지도를 표시할 div
23    mapOption = {
24      center: new kakao.maps.LatLng(37.566826, 126.9786567), // 지도의 중심좌표
25      level: 3 // 지도의 확대 레벨
26    };
27
28// 지도를 생성합니다
29var map = new kakao.maps.Map(mapContainer, mapOption);
30
31// 장소 검색 객체를 생성합니다
32var ps = new kakao.maps.services.Places();
33
34// 검색 결과 목록이나 마커를 클릭했을 때 장소명을 표출할 인포윈도우를 생성합니다
35var infowindow = new kakao.maps.InfoWindow({zIndex:1});
36
37// 키워드로 장소를 검색합니다
38searchPlaces();
39
40// 키워드 검색을 요청하는 함수입니다
41function searchPlaces() {
42
43  var keyword = document.getElementById("keyword").value;
44
45  if (!keyword.replace(/^\s+|\s+$/g, '')) {
46    alert('키워드를 입력해주세요!');
47    return false;
48  }
49
50  // 장소검색 객체를 통해 키워드로 장소검색을 요청합니다
51  ps.keywordSearch( keyword, placesSearchCB);
52
53  // 장소검색이 완료됐을 때 호출되는 콜백함수 입니다
54  function placesSearchCB(data, status, pagination) {
55    if (status === kakao.maps.services.Status.OK) {

```

```

61      displayPlaces(data);
62
63      // 페이지 번호를 표출합니다
64      displayPagination(pagination);
65    }
66    else if (status === kakao.maps.services.Status.ZERO_RESULT) {
67
68      alert('검색 결과가 존재하지 않습니다.');

```

<https://apis.map.kakao.com/web/sample/keywordList/>

<https://apis.map.kakao.com/web/sample/keywordList/>

[코드 5] Kakao Maps API

4. 회원가입 및 로그인 API

4.1 구글 소셜 로그인 API

JavaScript에서 API를 연동하기 위해선 사용자 인증 정보를 만들고 OAuth 클라이언트 ID 버튼을 누른다. 앱의 이름과 로고 그리고 지원 이메일과 승인된 도메인을 입력한다. 하단의 창이 뜨면 웹 애플리케이션이라고 선택을 한 후, 하단의 URL을 입력한다. 이후, 발급된 ID와 PWD를 통해 웹에 적용한다.

(1) 구글 로그인 연동을 위한 사용자 링크

```
<script src="https://apis.google.com/js/platform.js" async defer></script>
```

(2) 하단의 meta 태그를 <head>안에 추가시켜준다.

```
<meta name="google-signin-client_id" content="YOUR_CLIENT_ID.apps.googleusercontent.com">
```

(3) 원하는 위치에 버튼을 생성해준다.

```
<div class="g-signin2" data-onsuccess="onSignIn"></div>
```

(4) 다음과 같은 자바스크립트 코드를 추가한다.

```
function onSignIn(googleUser) {  
    var profile = googleUser.getBasicProfile();  
    console.log('ID: ' + profile.getId()); // Do not send to your  
    console.log('Name: ' + profile.getName());  
    console.log('Image URL: ' + profile.getImageUrl());  
    console.log('Email: ' + profile.getEmail()); // This is null  
}
```


3. 기능 설계

3.1 세부기능(Use Case 시나리오)

세부기능명	반려동물 등록
액터명	주 액터: 사용자
개요	사용자가 반려동물을 등록하기 위해 앱과 웹을 이용한다.
사전조건	회원가입 및 로그인
사후조건	반려동물 등록 정보를 받는다.
기본흐름	<ol style="list-style-type: none"> 1. 사용자가 시스템에 [반려동물 등록]을 요구한다. 2. 시스템은 반려동물 등록 정보를 사용자에게 전달한다. 반려동물 정보는 아래의 정보들로 구성된다. 강아지, 고양이 3. 시스템은 사용자가 등록한 반려동물 정보를 DB에 저장한다. 4. 사용자는 시스템에 [반려동물 등록] 종료를 요구한다. 5. 시스템은 [반려동물 등록]을 종료 한다.
대체흐름	<p>기본흐름 2에서 반려동물 등록 정보가 없다면 사용자에게 반려동물 등록을 요구한다.</p> <p>기본흐름 3에서 DB 연결과 저장에 실패하면 [반려동물 등록]을 다시 실행한다.</p>

세부기능명	건강 기록 작성
액터명	주 액터: 사용자
개요	사용자가 건강 기록을 작성하기 위해 앱과 웹을 이용한다.
사전조건	회원가입 및 로그인, 반려동물이 등록되어 있어야 한다.
사후조건	건강 기록 작성 정보를 받는다.
기본흐름	<ol style="list-style-type: none"> 1. 사용자가 시스템에 [건강 기록 작성]을 요구한다. 2. 시스템은 건강 기록 작성 정보를 사용자에게 전달하고 요청한다. 건강 기록 작성 정보는 아래의 정보들로 구성된다. 달력 기록, 알림 달력 기록의 세부 기능들은 아래의 기능들로 구성된다. 심박수, 주요 수치(사료, 간식, 음수량 등), 이상 증상, 대/소변, 구토 알림의 세부 기능들은 아래의 기능들로 구성된다. 일정 등록, 투약 알림 등록 3. 시스템은 사용자가 등록한 [건강 기록 작성]을 DB에 저장한다. 4. 사용자는 시스템에 [건강 기록 작성] 종료를 요구 한다. 5. 시스템은 [건강 기록 작성]을 종료 한다.
대체흐름	<p>기본흐름 2에서 입력이 없다면 사용자에게 건강 기록 작성 정보를 전달한다.</p> <p>기본흐름 3에서 DB 연결과 저장에 실패하면 [건강 기록 작성]을 다시 실행한다.</p>

세부기능명	알림
액터명	주 액터: 사용자
개요	사용자가 일정관리와 반려동물의 정해진 투약시간을 위해 [알림]을 이용한다.
사전조건	회원가입 및 로그인, 반려동물 등록, 일정 혹은 투약 정보가 등록되어 있어야 한다.
사후조건	일정과 투약 알림을 받는다.
기본흐름	<ol style="list-style-type: none"> 1. 사용자의 일정 혹은 투약 정보가 등록되어 있다. 2. 시스템이 [알림]을 실행한다. 3. 시스템은 사용자에게 알림을 발송한다. 알림의 내용은 등록된 일정의 내용 혹은 정해진 투약 시간으로 구성되어 있다. 4. 시스템은 [알림]을 종료한다.
대체흐름	없음

세부기능명	동물 병원 정보
액터명	주 액터: 사용자
개요	사용자가 동물 병원 위치 및 영업 시간등을 알기 위해 [동물 병원 정보]를 이용한다.
사전조건	회원가입 및 로그인, 반려동물 등록
사후조건	동물 병원 정보를 받는다.
기본흐름	<ol style="list-style-type: none"> 1. 사용자가 시스템에 [동물 병원 정보]를 요구한다. 2. 시스템은 지도 API를 이용한 동물 병원 정보를 반환한다. 3. 사용자가 시스템에 [동물 병원 정보] 종료를 요구한다. 4. 시스템은 [동물 병원 정보]를 종료한다.
대체흐름	없음

세부기능명	반려동물 커뮤니티
액터명	주 액터: 사용자
개요	사용자가 다른 사용자와 소통하기 위해 [반려동물 커뮤니티]를 이용한다
사전조건	회원가입 및 로그인
사후조건	반려동물 커뮤니티 정보를 받는다.
기본흐름	<ol style="list-style-type: none"> 1. 사용자가 시스템에 [반려동물 커뮤니티]를 요구한다. 2. 시스템은 [반려동물 커뮤니티]로 이동한다. <p>반려동물 커뮤니티는 아래의 정보들로 구성된다.</p> <p>반려동물에 관한 질문, 반려동물 자랑 게시판, 반려동물 물품 나눔 및 추천</p> <ol style="list-style-type: none"> 3. 사용자는 글쓰기, 수정, 삭제를 할 수 있다. 4. 사용자의 커뮤니티 활동 내역을 DB에 저장한다. 5. 사용자가 시스템에 [반려동물 커뮤니티] 종료를 요구한다. 6. 시스템은 [반려동물 커뮤니티]를 종료한다.
대체흐름	기본흐름 4에서 DB 연결과 저장에 실패하면 [반려동물 커뮤니티]를 다시 실행한다.

세부기능명	심박수 측정
액터명	주 액터: 사용자
개요	사용자가 사용자가 반려동물의 심박수를 측정하기 위해 [심박수 측정]을 이용한다.
사전조건	회원가입 및 로그인, 반려동물 등록
사후조건	심박수 측정 정보를 받는다.
기본흐름	<ol style="list-style-type: none"> 1. 사용자가 시스템에 [심박수 측정]을 요구한다. 2. 시스템은 심박 센서로 정보를 받아 사용자에게 전달한다. 3. 사용자는 시스템에 [심박수 측정] 종료를 요구한다. 4. 시스템은 [심박수 측정]을 종료 한다.
대체흐름	기본흐름 2에서 심박 센서 정보를 받지 않았다면 심박수 측정을 요구한다.

세부기능명	사용자 등록
액터명	주 액터: 사용자
개요	사용자가 정보 등록을 위해 [사용자 등록]을 이용한다.
사전조건	없음
사후조건	DB에 사용자 정보를 등록한다
기본흐름	<ol style="list-style-type: none"> 1. 사용자가 시스템에 [사용자 등록]을 요구한다. 2. 시스템은 사용자에게 정보 입력을 요청한다. <p>정보는 사용자의 이름, 나이, 성별, 키 등으로 구성된다.</p> <ol style="list-style-type: none"> 3. 시스템은 사용자의 입력 정보를 DB에 저장한다. 4. 시스템은 결과를 반환하고 [사용자 등록]을 종료한다.
대체흐름	<p>기본흐름 2에서 사용자 정보가 없다면 사용자에게 사용자 정보 입력을 요구한다.</p> <p>기본흐름 3에서 DB 연결과 저장에 실패하면 [사용자 등록]을 다시 실행한다.</p>

세부기능명	관리자 등록
액터명	주 액터: 관리자
개요	관리자가 시스템 관리를 위해 [관리자 등록]을 이용한다.
사전조건	없음
사후조건	DB에 관리자 정보를 등록한다.
기본흐름	<ol style="list-style-type: none"> 1. 관리자가 시스템에 [관리자 등록]을 요구한다. 2. 시스템은 관리자에게 정보 입력을 요청한다. 정보는 관리자의 이름, 키로 구성된다. 3. 시스템은 관리자의 입력 정보를 DB에 저장한다. 4. 시스템은 결과를 반환하고 [관리자 등록]을 종료한다.
대체흐름	<p>기본흐름 2에서 관리자 정보가 없다면 관리자에게 관리자 정보 입력을 요구한다.</p> <p>기본흐름 3에서 DB 연결과 저장에 실패하면 [관리자 등록]을 다시 실행한다.</p>

3.2 UI/UX 설계

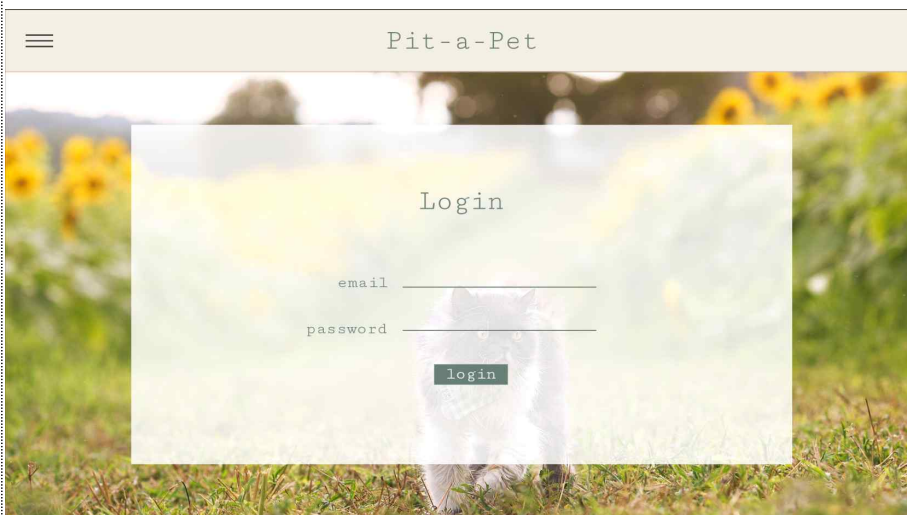
*웹 UI/UX - 개발과정에서 UI는 변동이 있을 수 있습니다

1) 메인 화면



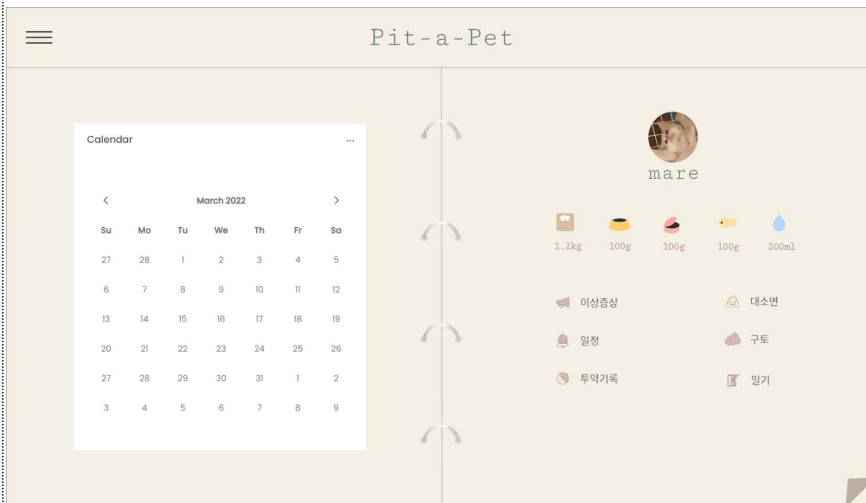
[그림 17 웹 메인 페이지]

2) 로그인 화면



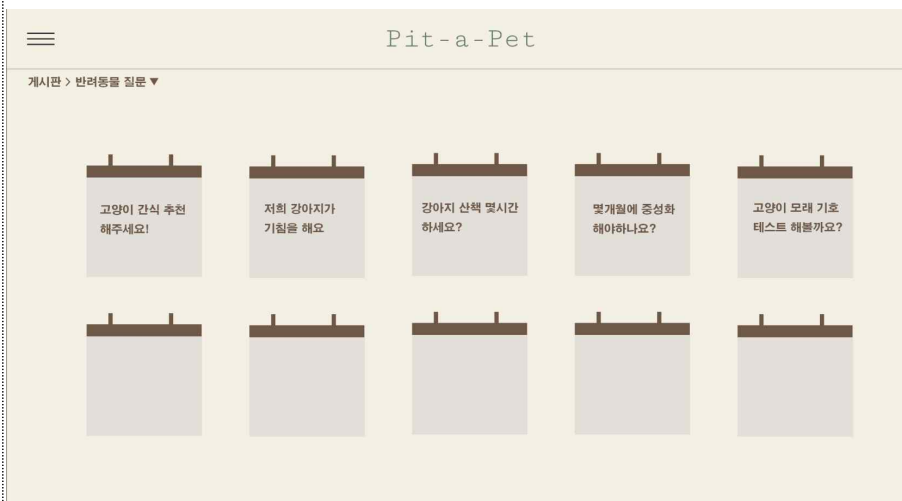
[그림 18 웹 로그인 페이지]

3) 캘린더 화면



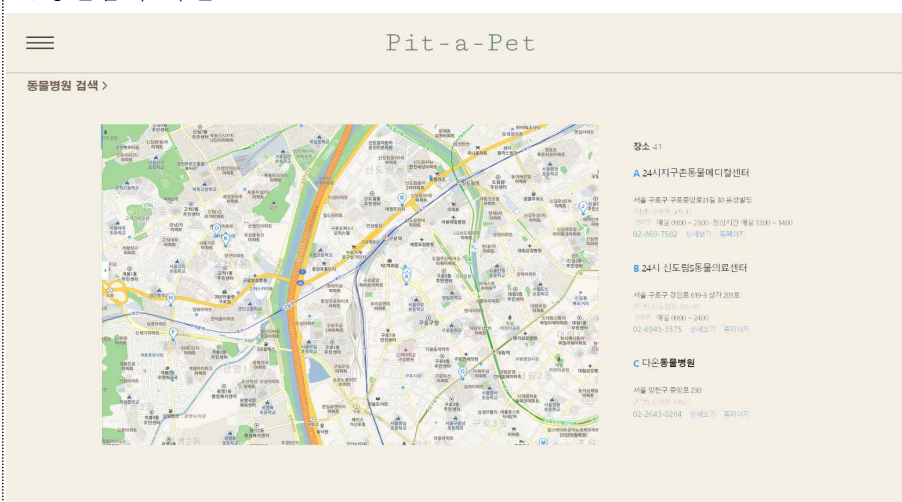
[그림 19] 웹 달력 페이지

4) 게시판 화면



[그림 20] 웹 게시판 페이지

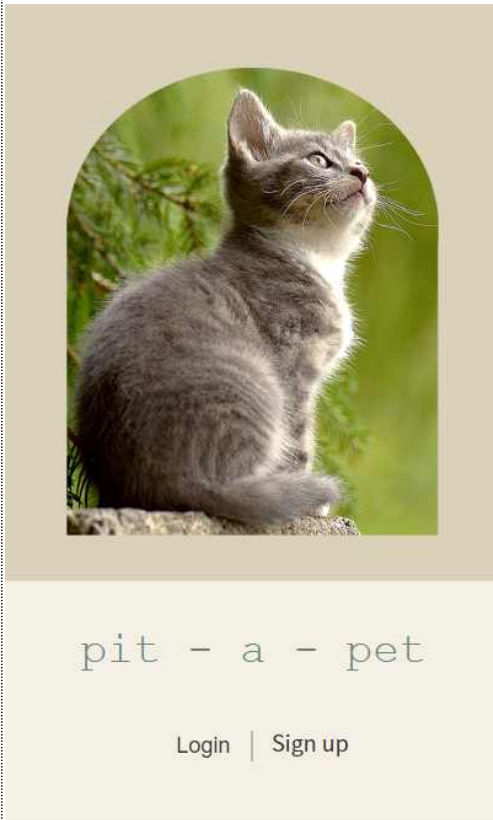
5) 병원검색 화면



[그림 21] 병원 검색 페이지

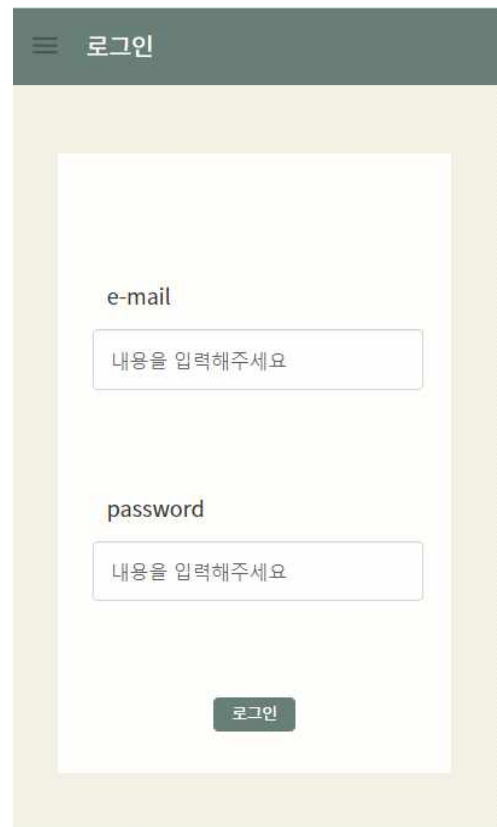
*앱 UI/UX 개발과정에서 UI는 변동이 있을수 있습니다

1) 앱 시작 화면



[그림 23] 앱 메인 화면

2) 앱 로그인 화면



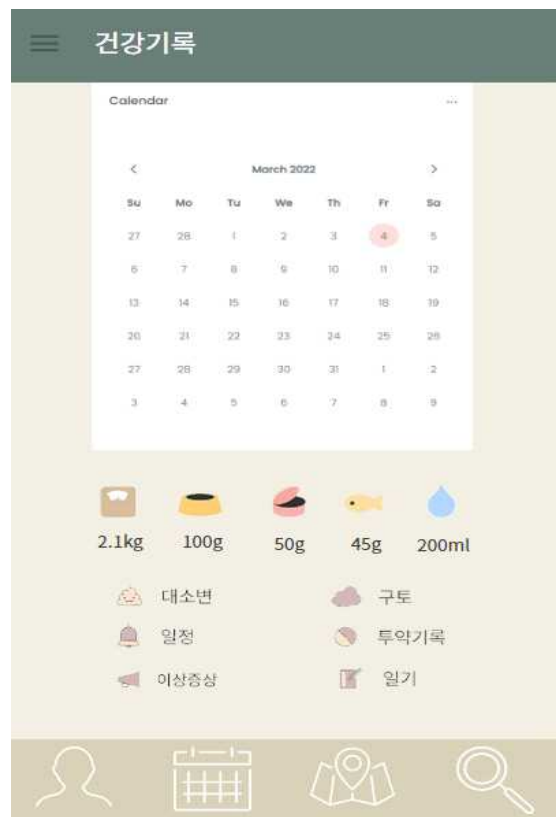
[그림 24] 앱 로그인 화면

3) 반려동물 등록 화면



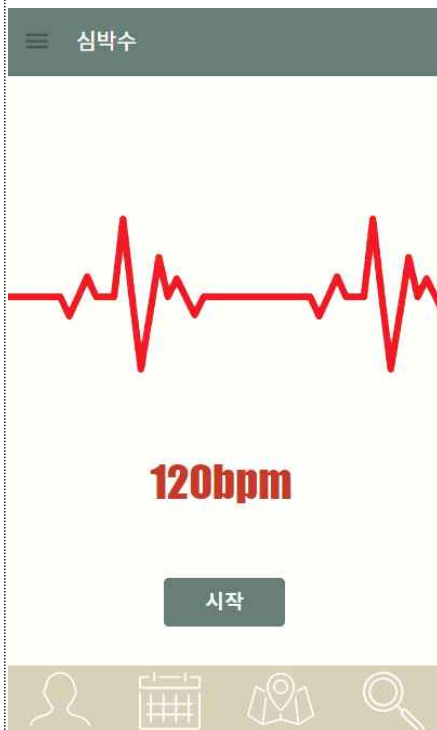
[그림 25] 앱 반려동물 등록 화면

4) 달력 건강 기록 화면



[그림 26] 앱 달력 건강 기록 화면

5) 심박수 및 심전도 화면



[그림 27] 앱 심박관련 화면

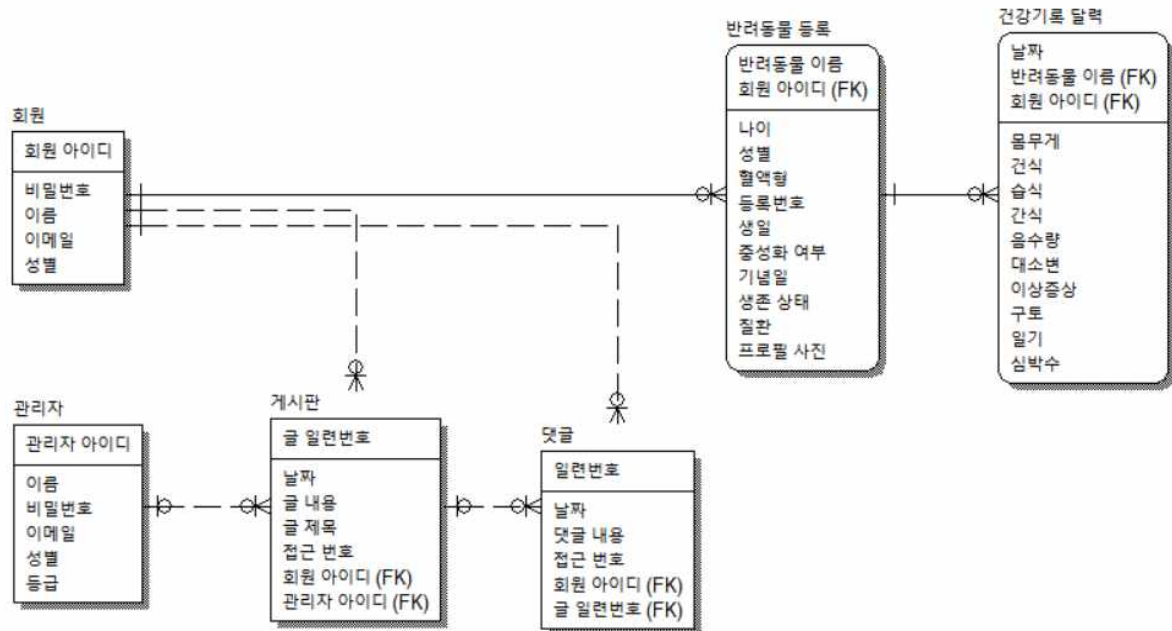
6) 병원 검색 화면



[그림 28] 앱 병원 검색 화면

3.3 데이터베이스 설계

3.3.1 DB 설계



[그림 30] DB 설계 및 관계 설정

데이터베이스명	pitapetDB	테이블명	회원(member)		
테이블 설명	회원 정보를 저장하는 테이블				
컬럼명	타입	NOT NULL	PK	FK	비고
회원 아이디(mem_id)	VARCHAR(20)	Y	Y	Y	반려동물 등록, 건강기록 달력, 게시판, 댓글 테이블
비밀번호(mem_pw)	VARCHAR(20)	Y			암호화
이름(mem_name)	VARCHAR(20)	Y			
이메일(mem_email)	VARCHAR(20)	Y			
성별(mem_gen)	VARCHAR(20)	Y			남/여

데이터베이스명	pitapetDB	테이블명	관리자(admin)		
테이블 설명	관리자 정보를 저장하는 테이블				
컬럼명	타입	NOT NULL	PK	FK	비고
관리자 아이디(ad_id)	VARCHAR(20)	Y	Y		
비밀번호(ad_pw)	VARCHAR(20)	Y			암호화
이름(ad_name)	VARCHAR(20)	Y			
이메일(ad_email)	VARCHAR(20)	Y			
성별(ad_gen)	VARCHAR(20)	Y			남/여
등급(rank)	INT	Y			

데이터베이스명	pitapetDB	테이블명	반려동물 등록(pet)		
테이블 설명	반려동물 등록 정보를 저장하는 테이블				
컬럼명	타입	NOT NULL	PK	FK	비고
반려동물 이름(pet_name)	VARCHAR(20)	Y	Y		건강기록 달력 테이블
반려동물 나이(p_age)	INT				
반려동물 성별(p_gen)	VARCHAR(20)				
혈액형(blood)	VARCHAR(20)				
등록번호(p_num)	INT				
생일(birth)	VARCHAR(20)				
중성화 여부(neutering)	VARCHAR(20)				
기념일(anniversary)	VARCHAR(20)				
생존 상태(state)	VARCHAR(20)				
질환(p_disease)	VARCHAR(20)				
프로필 캐릭터(profile)	img	Y			
회원 아이디(mem_id)	VARCHAR(20)	Y	Y	Y	회원 테이블

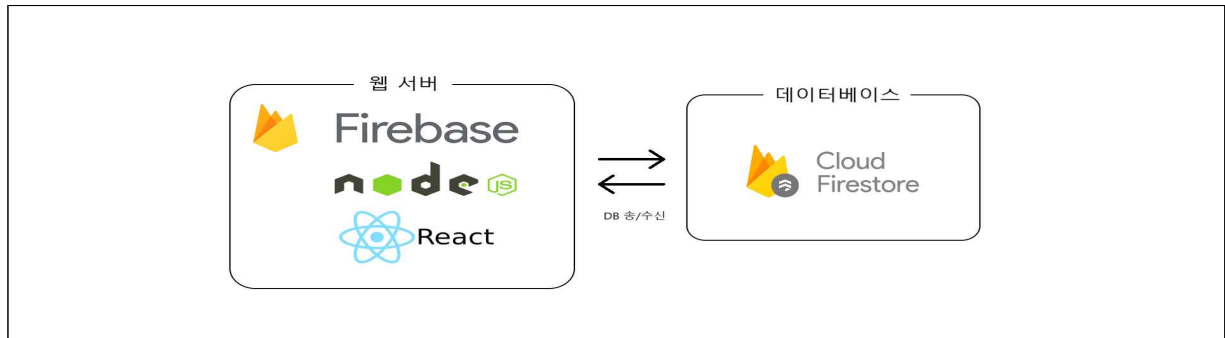
데이터베이스명	pitapetDB	테이블명	건강기록 달력(calender)		
테이블 설명	반려동물의 건강을 달력에 기록하는 테이블				
컬럼명	타입	NOT NULL	PK	FK	비고
날짜(date)	datetime	Y	Y		
몸무게(weight)	VARCHAR(20)				
건식(dry)	VARCHAR(20)				
습식(wet)	VARCHAR(20)				
음수량(water)	VARCHAR(20)				
대소변(poo)	VARCHAR(20)				
이상증상(symptom)	VARCHAR(45)				
구토(vomit)	VARCHAR(20)				
심박수(heart)	INT				
일기(diary)	VARCHAR(200)				
반려동물 이름(pet_name)	VARCHAR(20)	Y	Y	Y	반려동물 등록 테이블
회원 아이디(mem_id)	VARCHAR(20)	Y	Y	Y	회원 테이블

데이터베이스명	pitapetDB	테이블명	게시판(board)		
테이블 설명	게시판의 요소를 저장하는 테이블				
컬럼명	타입	NOT NULL	PK	FK	비고
날짜(b_date)	datetime	Y			
글 제목(b_title)	VARCHAR(20)	Y			
글 내용(b_content)	VARCHAR(20)	Y			
접근 번호(b_access)	INT	Y			
글 일련번호(b_id)	INT	Y	Y		
회원 아이디(mem_id)	VARCHAR(20)	Y		Y	회원 테이블

3.4 통신 설계

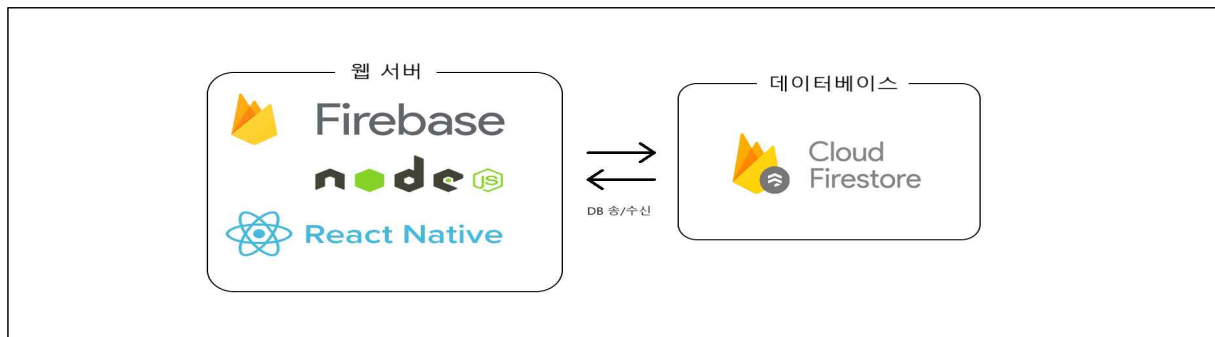
3.4.1 웹서버 ↔ Firebase Cloud Firestore 통신

사용자의 정보 및 반려동물의 정보를 저장하는 모든 웹서버는 Firebase Cloud Firestore를 이용한다.

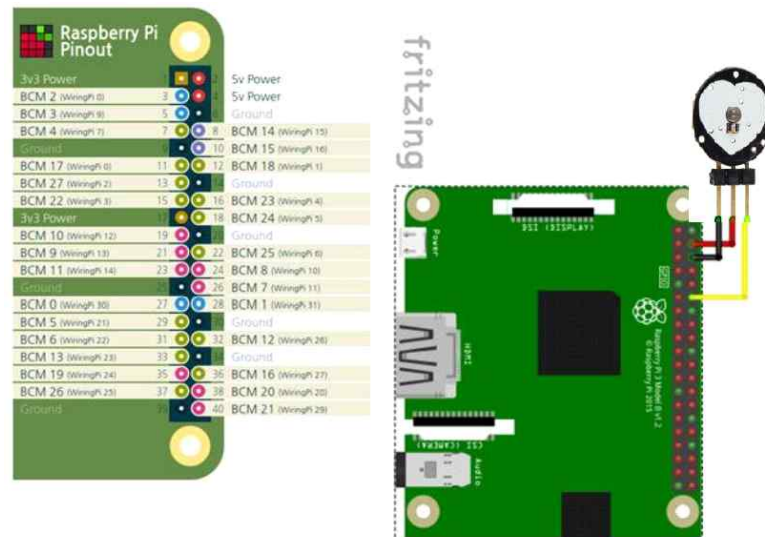


3.4.2 앱서버 ↔ Firebase Cloud Firestore 통신

사용자의 정보 및 반려동물의 정보를 저장하는 모든 앱서버는 Firebase Cloud Firestore를 이용한다.



3.5 디바이스 제어



[그림 31] 디바이스 제어 설계도

1) 구성

라즈베리파이	심박센서(Heart-Pulse-Sensor)
4Pin, 5v Power	2Pin, VCC
6Pin, Ground (GND)	3Pin, GND
12Pin, BCM 18 (WiringPi1)	1Pin, DAT

Gpio Pin을 사용하였다. 선 연결은 4번째 Pin인 5V Power에 VCC를 꽂았고, 6번째 Pin인 Ground에 심박센서 Ground를 꽂았다. 그리고 심박센서를 좀 더 정교하게 동작하기 위해서 PWM이 지원되는 12번째 Pin인 BCM18핀에 DAT를 꽂아서 사용했다.

2) 작동

1. 안드로이드에서 라즈베리파이로 구현된 웹서버에 접속한다.
2. 접속된 웹서버에서 버튼을 클릭한다.
3. 라즈베리파이를 통해 측정된 값을 웹서버에 전송하여 보여준다.

V. 구현

1. 구현 내용

1.1 웹 구현

1.1.1 회원가입 / 로그인 페이지

ID와 Password를 입력 후 DB에 있는 값과 일치하는지 비교 후 로그인을 할 수 있다. 회원가입과 구글 계정 로그인이 가능한 버튼을 만들고 각각의 페이지를 연결했다.

1.1.2 구글 소셜 로그인 API

Firebase Authentication을 활용하여 발급한 구글 로그인 API의 ID와 Password를 통해 웹에 적용했다.

1.1.3 카카오 Maps API

카카오 개발자 사이트에 접속 후 개발자를 등록 및 앱 생성을 한다. 다음으로 웹 플랫폼을 추가해 사이트 도메인을 등록하고 JavaScript 키를 발급받는다. 이 키를 지도 API의 appkey로 사용하여 카카오 Maps API를 웹에 적용할 수 있게 된다. 장소를 검색하면 목록으로 출력하고, 현재 위치를 받아와 위치 기반 가까운 동물병원 목록을 출력한다.

1.1.4 게시판 페이지

반려동물 관련 글을 작성하고 삭제할 수 있으며 글의 제목과 내용, 그리고 파일을 첨부할 수 있다. 게시판과 데이터베이스를 연동하여 글 작성, 삭제 등을 관리할 수 있다.

1.2 앱 구현

1.2.1 반려동물 등록

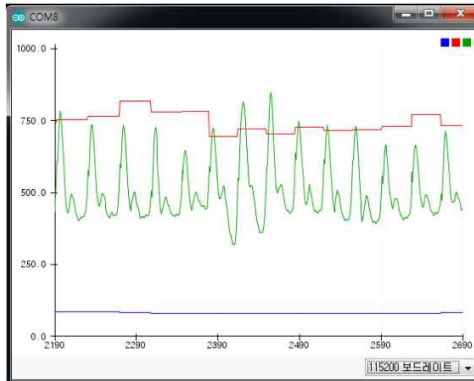
반려동물 등록에 이름, 성별, 나이 등 필요한 요소들을 선언하고 그 값을 저장하는 함수와 변경 내용을 데이터베이스와 실시간 연동하는 함수를 구현했다.

1.2.2 심박수 캘린더 실시간 등록

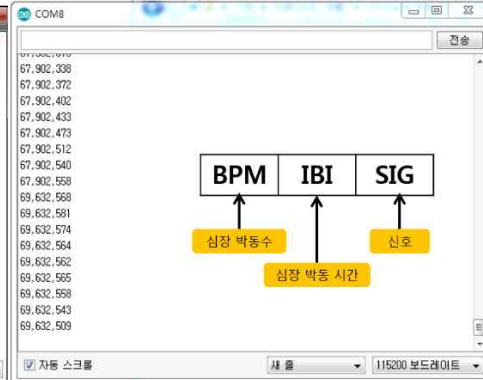
앱의 핵심적인 기능인 캘린더 기능은 심박센서로 측정한 심박수를 캘린더에 실시간으로 등록된다. 심박수를 포함한 음수량, 식사량과 같은 각 요소에 정보를 입력하면 데이터베이스에 저장되고, 캘린더에 저장한 날짜를 클릭하면 데이터베이스에 저장되었던 값이 호출되어 해당 날짜의 기록을 볼 수 있다.

1.3 심박수 측정 기능 구현

아두이노와 심박센서를 연결하고 반려동물의 심박수를 측정하면 [그림 32]와 같은 시리얼 화면에 그래프가 표시된다. 아두이노의 심박센서 라이브러리를 이용하여 구동하면 [그림 33]과 같은 시리얼 창이 나타난다.

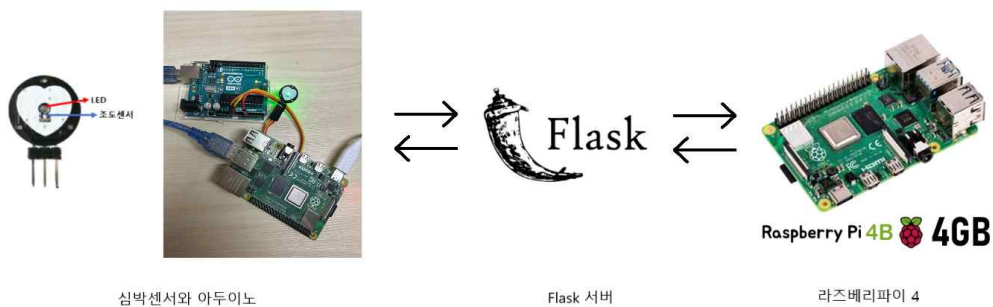


[그림 32] 심박수 시리얼 화면



[그림 33] 심박센서 라이브러리 시리얼

라즈베리파이와 아두이노의 연동은 아두이노 환경에서 더 적합한 심박센서를 작업 후 라즈베리파이로 Flask 서버를 사용해 값을 받아온다. 라즈베리파이로 넘어온 값을 html 캘린더에 값을 넘겨주고 심박수를 받아오는 함수를 구현하여 이 함수를 호출하면 실시간으로 캘린더와 심박수 측정 데이터를 연동할 수 있다.



[그림 34] 심박센서 기능

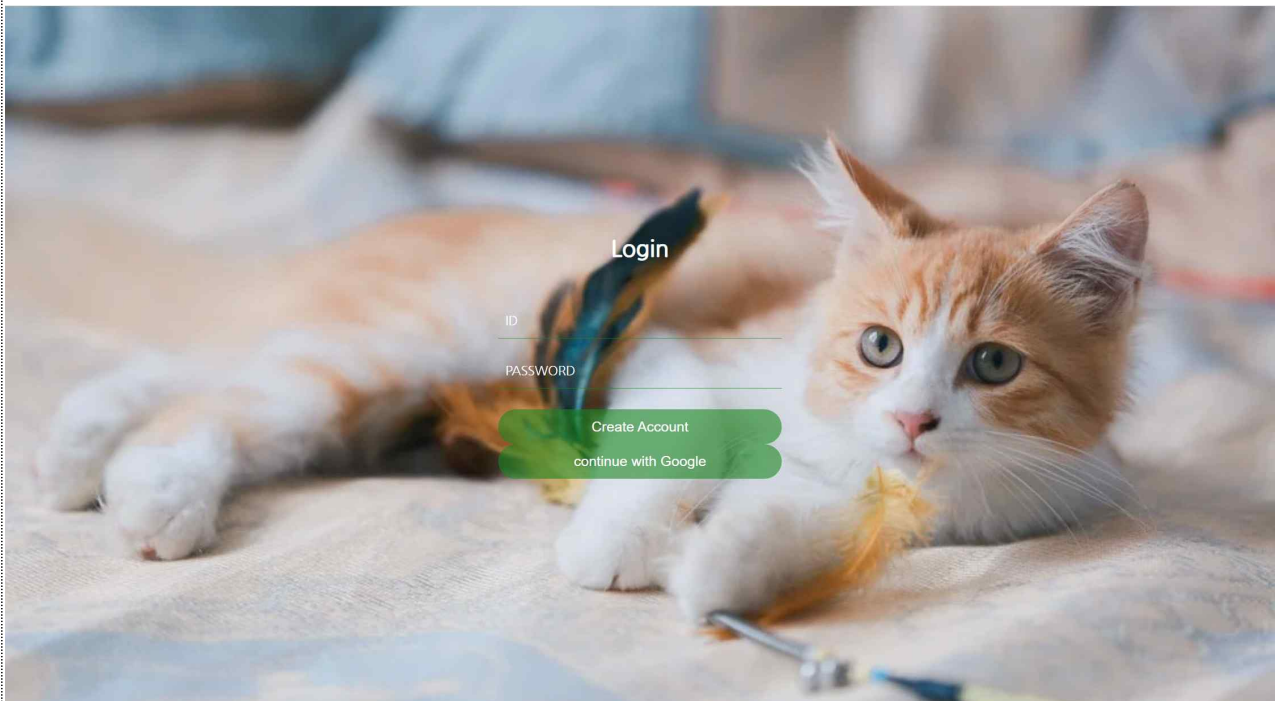
2. 실제 구현 UI/UX

웹 UI/UX

1) 메인 페이지

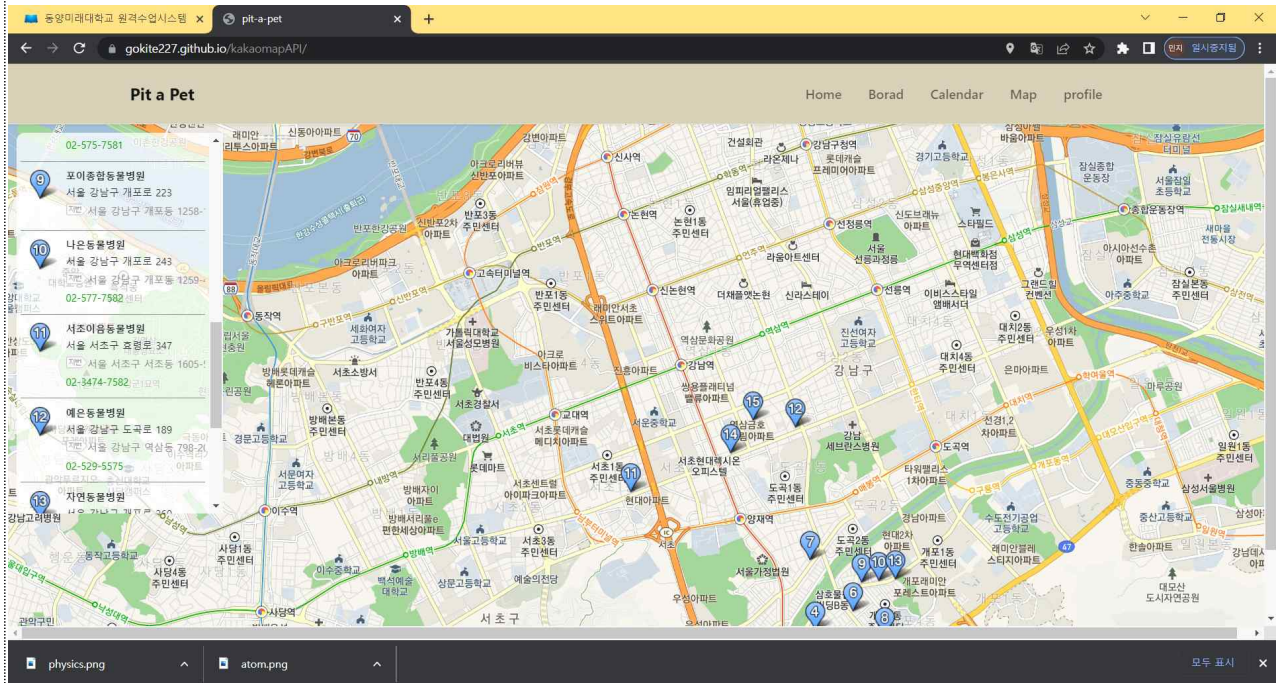


2) 로그인 페이지

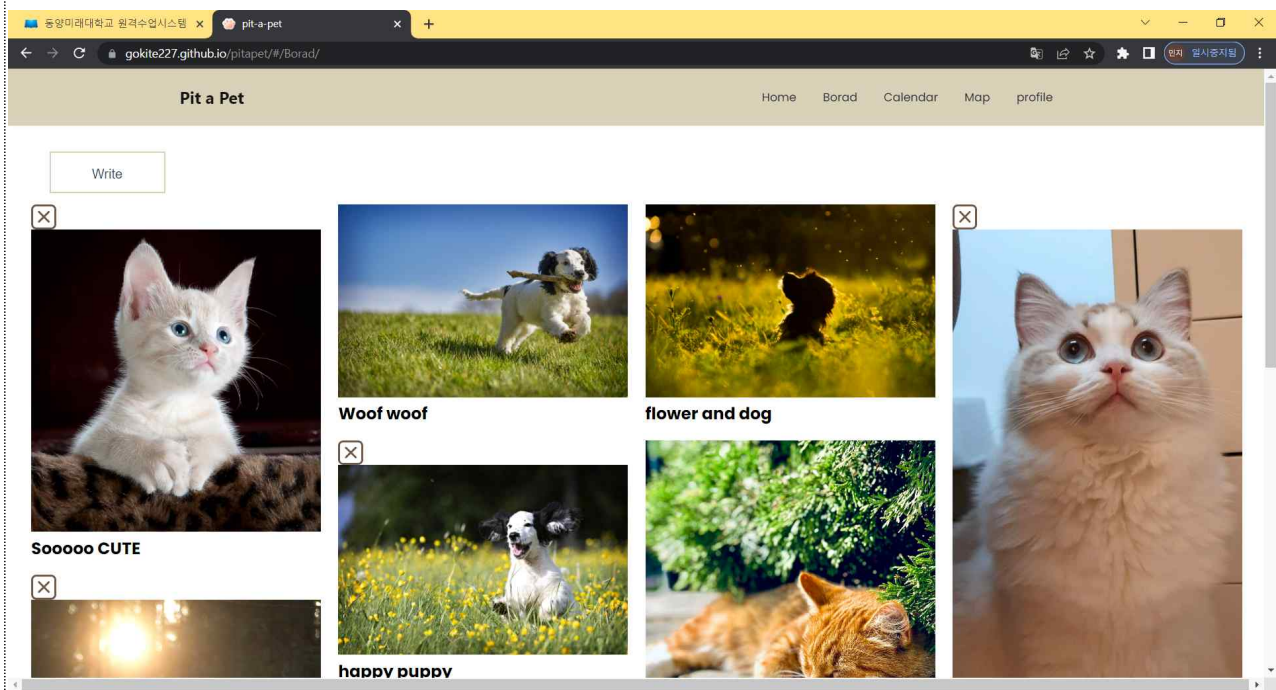


웹 UI/UX

3) 지도 페이지



4) 게시판 페이지

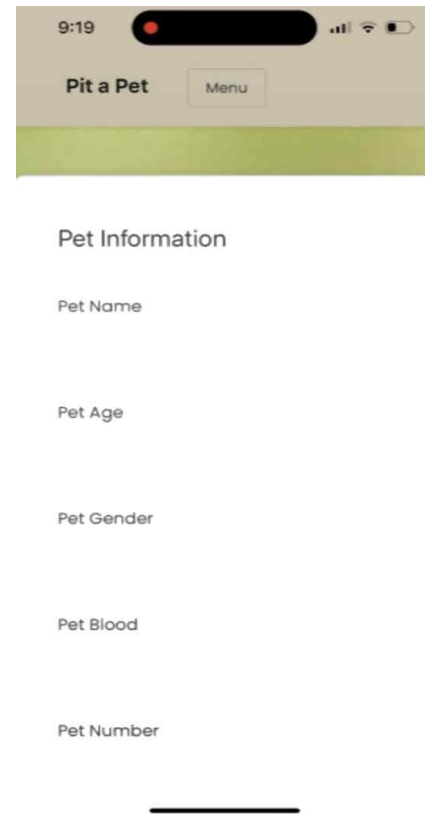


앱 UI/UX

1) 앱 시작 화면

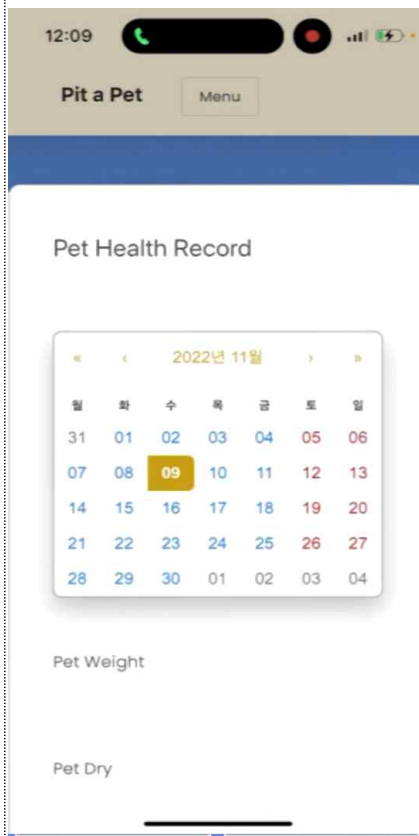


2) 앱 반려동물 등록 화면



앱 UI/UX

3) 캘린더 등록 화면



4) 앱 심박수 측정 결과 화면



VI. 결론

1. 문제점 및 개선 방안

1.1 구현하지 못한 기능

질병 검색 필터와 알림 기능은 시간이 부족한 문제와 기술 구현의 난이도 문제로 구현하지 못한 기능이다.

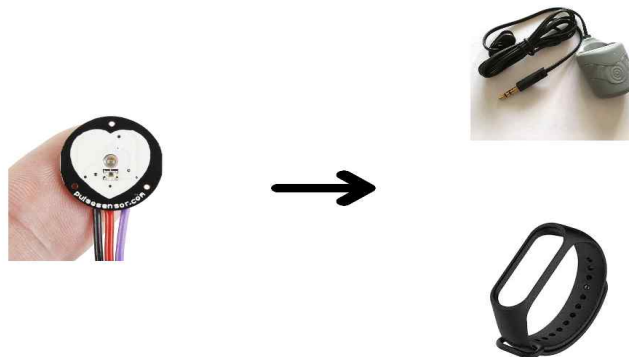
질병 검색 필터 기능은 기술 구현의 난이도가 높았는데, 데이터 분석 자료에 필요한 csv 파일은 구했지만, 필요한 데이터를 어떤 기준을 잡고 분석해야 할지 그리고 어떤 방법으로 데이터를 필터링해야 하는지 등의 문제점이 있었다. 그리고 위의 문제를 해결했다고 가정했을 때, 데이터 분석은 python 언어와 pandas 라이브러리를 사용하는 데 이를 웹, 앱, 데이터베이스 그리고 서버와의 연동 기술 구현에 어려움이 있을 것이라 예상된다.

질병 검색 필터의 기술적 어려움의 개선 방안은 예제와 자료들을 찾아보고 팀과 소통하여 데이터 분석을 위한 명확한 기준을 세워야 할 것이다. 그리고 연동 기술 구현의 어려움은 React와 연동하기 쉬운 다른 언어와 라이브러리로 대체할 수 있다. 또한 데이터를 분석하고 필터링 된 데이터를 데이터베이스에 입력하고 키워드를 검색하면 키워드가 데이터베이스 안에 있는 정보들 중 일치하는 데이터를 데이터베이스에 입력하고 키워드를 검색하면 키워드가 데이터베이스 안에 있는 정보들 중 일치하는 정보를 찾아 출력하는 함수를 구현하여 해결할 수 있다.

알림 기능은 시간이 부족한 문제로 구현을 하지 못한 기능이다. 따라서 체계적으로 프로젝트 개발 시간을 관리하고 매일 하나씩 혹은 미리미리 기능을 구현하는 노력이 필요하다.

1.2 심박센서의 보완

현재의 반려동물의 심박수를 측정하는 심박센서는 PCB 기판이다. 반려동물의 육구(발바닥)에 닿았을 때 반려동물이 불편감을 느낄 수 있는 문제가 있어 측정을 거부할 수도 있다. 따라서 심박센서에 실리콘처럼 부드러운 재질을 위에 입히거나, 갤럭시 핏처럼 센서 주변을 시곗줄 모양의 밴드를 만들어 개선할 수 있다.



[그림 35] 심박센서의 보완

1.3 완성도

시간상의 문제로 구현하지 못한 기능도 생기고, 구현한 기능도 완성도가 부족한 문제점이 있다. 자료와 예제를 찾는데 시간을 많이 소모한 점도 있고, 프로젝트를 체계적으로 관리하지 못해 발생한 문제이기 때문에, 프로젝트를 매일 조금씩 개발하는 등의 체계적으로 관리하고, 기능 구현에 어려움이 있거나 문제가 발생하면 팀원과 바로 소통하는 능력이 부족했다. 문제에 직면했을 때, 일정 시간 이상 넘어가면 같이 소통하고 회의하는 과정이 필요하다.

2. 개인별 소감

학번	작성자	개인별 소감
20202871	김가연	3년동안 배운 지식을 총합해서 큰 프로젝트를 진행할 수 있어서 좋았다. 웹, 앱, 하드웨어 그리고 데이터베이스 등 3년동안 배웠던 것들을 모두 사용하고 과제가 아닌 직접 문제를 해결하기 위해 아이디어를 내고 그것을 위한 작품을 만들어 낸 것이 졸업작품의 의의인 것 같다.

학번	작성자	개인별 소감
20190659	김민지	졸업작품 이전의 프로젝트들은 웹이면 웹, 앱이면 앱 처럼 하나의 분야만 집중적으로 개발했었는데, 졸업작품은 컴퓨터소프트웨어공학과와 커리큘럼에 들어간 대부분의 기술들을 융합하고 연계하며 개발을 한 것이라 난이도가 어려웠지만 그만큼 스스로 발전할 수 있는 기회가 되었다. 또한 직접 프로젝트의 설계부터 구현까지 세세하게 작업한 것이 결과물로 보여졌을 때, 매우 뿌듯하였고 졸업작품 진행과정에서 팀원과 소통하고 조율하는 방법을 배울 수 있었다.

VII. 참고자료

번호	종류	제목	저자	발행연도
1	웹사이트	[JSP] GET방식으로 서버DB에 데이터 저장하기 (https://beombool2.tistory.com/5)		
2	웹사이트	[MySQL/JAVA]MySQL과 이클립스 연동하기 (https://sso-feeling.tistory.com/105)		
3	웹사이트	카카오 로그인 (https://developers.kakao.com/docs/latest/ko/kakaologin/common)		
4	웹사이트	카카오 Maps API (https://apis.map.kakao.com/)		
5	웹사이트	[Android] 안드로이드<->PHP<->Mysql 연동 완벽정리 (https://yoo-hyeok.tistory.com/16)		
6	웹사이트	라즈베리파이에서 아날로그 입력값 가져오기-SPI 통신 (https://blog.naver.com/simjk98/221220087802)		
7	웹사이트	라즈베리파이 심박센서 사용하기 실시간 그래프 그리기 (https://blog.naver.com/simjk98/221873207244)		
8	웹사이트	라즈베리파이 아두이노 연동하기 (https://robotai.tistory.com/m/64)		
9	웹사이트	리액트 파이어베이스 연동하기 (https://velog.io/@hyeonseop/React%EC%97%90%EC%84%9C-Firebase-%EC%93%B0%EA%B8%B0)		
10	웹사이트	리액트 파이어베이스 데이터베이스 연동하기 (https://duckgugong.tistory.com/m/207)		
11	기사	‘팻코노미’ 시대... 반려동물 산업 ‘폭발 성장’에 전문대 인재 양성 총력 (http://news.unn.net/news/articleView.html?idxno=516864)	이중삼 기자	2021.10.12
12	기사	[2021 반려가구 현황과 양육실태] KB금융지주 경영연구소① (http://www.dailygaewon.com/news/articleView.html?idxno=11299)	개원	2021.04.22
13	웹사이트	심박센서 실리콘 보완 사진 (https://ko.aliexpress.com/item/4000250651653.html)		
14	웹사이트	심박센서 밴드 보완 사진 (https://korean.alibaba.com/product-detail/Factory-Wholesale-M3-M4-M4-Pro-1600071018059.html)		
15	웹사이트	React, Firebase 연동 (https://kth990303.tistory.com/143)		
16	웹사이트	React, Firestore 연동 (https://muhly.tistory.com/119)		

VIII. 부록

1. 개인별 자기평가(수행내용 및 보완해야 할 점)

1.1 김가연-20202871

학번	20202871	이름	김가연
기여도	(10 / 10점)	기술수준	(9 / 10점)
구분	내용		
수행 내용	<ul style="list-style-type: none"> •웹 게시글 페이지, 게시글 작성 페이지 파이어 베이스 데이터 연동 및 프론트엔드 •웹 캘린더 페이지 파이어베이스 데이터 연동 및 프론트엔드 •웹 지도 페이지 카카오 maps API를 사용한 근처 동물병원 검색 리스트 •웹 반려동물 정보 작성 페이지 파이어 베이스 데이터 연동 및 프론트엔드 •라즈베리 파이, 아두이노, 심박센서 연동 및 센서값 받아오기 •라즈베리 파이 flask 이용, 웹서버 연동 및 센서값 배포 •리액트 네이티브 이용 웹 뷰로 앱에 화면 띄우기 		
과제수행 시 문제점 및 해결방안	<p>•스프링 프레임 워크로 데이터 베이스에 연동하는 것에 어려움을 겪어 웹을 구현하는 것에 차질이 있었지만 개발언어를 리액트로 바꾸고 파이어베이스를 사용하여 데이터 베이스 연동을 용이하게 하여 어려움을 극복하였다. 또한 파이어 베이스를 이용함으로 써 구글 연동 로그인 또한 구현할 수 있었다.</p> <p>•카카오 maps API를 사용하면서 사용 방법 및 토큰 발급에 대해 처음 시도해 보아서 어려움을 겪었지만 카카오 maps API 공식 페이지를 참고하여 여러 가지 예제들과 사용방식에 대해 참고하여 원하는 기능을 구현할 수 있었다.</p> <p>•라즈베리 파이 센서 값을 웹페이지에 연동하는 것이 쉬운 일이라고 생각했는데 라즈베리에서 직접 웹서버를 열고 라즈베리로 열은 웹서버에 웹페이지와 연동해야 하는 과정이 필요하여 생각보다 진행에 어려움을 겪었다. 각종 라즈베리파이 관련 자료, 유튜브, 강의 등을 찾아보며 지금 직면한 오류, 구현해야 하는 기능에 대해 해결을 하였다.</p> <p>•라즈베리 파이가 아날로그 센서값을 받아오지 못하여 따로 ADC라는 아날로그 신호를 디지털 신호로 변환해주는 것이 필요하였는데 이를 아두이노로 대체하여 심박센서를 아두이노에 연결하고 센서가 연결된 아두이노를 라즈베리파이에 연결하여 아날로그 값을 받아 올 수 있도록 해결하였다.</p>		

1.2 김민지-20190659

학번	20190659	이름	김민지
기여도	(9 / 10점)	기술수준	(8 / 10점)
구분	내용		
수행 내용	<ul style="list-style-type: none"> • 웹 메인페이지 부트스트랩과 JavaScript를 이용한 프론트엔드 및 반응형 구현 • 웹 캘린더 페이지 React를 이용한 프론트엔드 및 반응형 구현 • 웹 반려동물 프로필 페이지 React를 이용한 프론트엔드 및 반응형 구현 • 웹 파이어베이스 서버 연동 • 앱 React-Native, Node.js를 이용한 초기 메인, 로그인, 지도 앱 화면 프론트엔드 구현 • 앱 React-Native, Firebase, Firebase Authentication을 이용한 초기 로그인 백엔드 구현 • 앱 React-Native, Firebase, Firebase Cloudstore 이용한 초기 데이터베이스 구현 		
과제수행 시 문제점 및 해결방안	<ul style="list-style-type: none"> • 스프링 프레임 워크로 서버인 톰캣과 MySql을 연동하는 것에 어려움을 겪어 웹에 구현하는 것에 시간이 오래 걸리는 문제가 있었다. 그러나 개발 툴을 VisualStudio를 사용하고, 언어는 배우기가 간단하고 예제가 많아 사용하기 쉬운 React를 사용했다. 또한 서버를 AWS와 비슷한 Firebase를 이용하여 쉽고 빠른 개발을 할 수 있게 되어 시간이 부족한 문제를 해결할 수 있었다. • 앱 개발에서 처음 써보는 언어인 React-Native를 사용하여 개발 시간이 느려지고 초기의 앱은 완성도가 높지 않았다. 그러나 관련 언어의 강의를 보고 학교 도서관에서 기술서를 찾아 차근차근 공부하며 낯선 언어 문법과 구조를 파악하며 개발을 진행하였다. 또한 해당 언어의 공식 문서와 예제들을 적용하며 점차 실력이 나아지게 되었다. • 웹과 앱을 모니터, 태블릿, 모바일 등 어떤 기기에서도 구현한 화면이 깨지지 않고 크기가 화면의 비율에 맞추어지는 반응형으로 프론트엔드를 수정하는 과정에 어려움이 있었다. flex 안에 있는 다양한 속성들의 기능을 알고 있어야 하며, 같은 반응형이여도 rem, vh 등 단위가 달라서 이 단위들의 차이점을 숙지하는데 시간이 오래 걸렸다. 그러나 인터넷의 다양한 예제들을 보고 크롬의 개발자도구에서 직접 수정해보며 따로 공부하지 않아도 자연스럽게 체득하게 되어 이후부터는 수월하게 개발할 수 있었다. 		

2. 프로그램 소스

웹 - 로그인 페이지 소스

1) 기본 로그인 함수

```
const onSubmit = async (event) => { //로그인 동기 / 비동기 함수
  event.preventDefault();
  try {
    let data;
    if (newAccount) { //새로운 계정이면 이메일과 비밀번호 입력
      data = await authService.createUserWithEmailAndPassword(
        email,
        password
      );
    } else { // 그게 아니면 이메일과 비밀번호로 로그인
      data = await authService.signInWithEmailAndPassword(email, password);
    }
    console.log(data); // 값을 콘솔창에 찍어 확인
  } catch (error) {
    seterror(error.message); //에러 메세지 확인
  }
};
```

2) 구글 소셜 로그인 API

```
const onSocialClick = async(event) => { //구글 소셜 로그인 API 함수
  const {
    target: {name},
  } = event;
  let provider;
  if (name === "google") {
    provider = new firebaseInstance.auth.GoogleAuthProvider(); //Firebase 서버를 사용하여 구글 api 발급 후
    // 보유한 구글 계정으로 로그인 가능
  }
  const data = await authService.signInWithPopup(provider);
  console.log(data);
};
```


웹 - 지도 페이지 소스

1) 현재 위치 받아오기

```
1  useEffect(() => {
2    if (navigator.geolocation) {
3      // Geolocation을 이용해서 접속 위치를 얻어옵니다
4      navigator.geolocation.getCurrentPosition(
5        (position) => {
6          setState((prev) => ({
7            ...prev,
8            center: {
9              lat: position.coords.latitude, // 위도
10             lng: position.coords.longitude, // 경도
11            },
12            isLoading: false,
13          )))
14        },
15        (err) => {
16          setState((prev) => ({
17            ...prev,
18            errMsg: err.message,
19            isLoading: false,
20          )))
21        }
22      )
23    } else {
24      // HTML5의 Geolocation을 사용할 수 없을때 마커 표시 위치와 인포윈도우 내용을 설정합니다
25      setState((prev) => ({
26        ...prev,
27        errMsg: "geolocation을 사용할수 없어요..",
28        isLoading: false,
29      )))
30    }
31  }, [])
```

2) 키워드로 장소 검색

```
1  useEffect(() => {
2    if (!map) return
3    const ps = new kakao.maps.services.Places()
4
5    ps.keywordSearch(b, (data, status, _pagination) => {
6
7      if (status === kakao.maps.services.Status.OK) {
8        // 검색된 장소 위치를 기준으로 지도 범위를 재설정하기위해
9        // LatLngBounds 객체에 좌표를 추가합니다
10       const bounds = new kakao.maps.LatLngBounds()
11       let markers = []
12
13       for (var i = 0; i < data.length; i++) {
14         // @ts-ignore
15         markers.push({
16           position: {
17             lat: data[i].y,
18             lng: data[i].x,
19           },
20           content: data[i].place_name,
21         })
22         // @ts-ignore
23         bounds.extend(new kakao.maps.LatLng(data[i].y, data[i].x))
24       }
25       setMarkers(markers)
26
27       // 검색된 장소 위치를 기준으로 지도 범위를 재설정합니다
28       map.setBounds(bounds)
29     }
30   })
31 }, [map])
32
```

웹 - 지도 페이지 소스

3) 내 위치에서 가까운 동물병원을 리스트에 담아 마커로 표시

```
1 function displayPlaces(places) {
2
3     var listEl = document.getElementById('placesList'),
4     menuEl = document.getElementById('menu_wrap'),
5     fragment = document.createDocumentFragment(),
6     bounds = new kakao.maps.LatLngBounds(),
7     listStr = '';
8
9     // 검색 결과 목록에 추가된 항목들을 제거합니다
10    removeAllChildNods(listEl);
11
12    // 지도에 표시되고 있는 마커를 제거합니다
13    removeMarker();
14
15    for ( var i=0; i<places.length; i++ ) {
16
17        // 마커를 생성하고 지도에 표시합니다
18        var placePosition = new kakao.maps.LatLng(places[i].y, places[i].x),
19        marker = addMarker(placePosition, i),
20        itemEl = getListitem(i, places[i]); // 검색 결과 항목 Element를 생성합니다
21
22        // 검색된 장소 위치를 기준으로 지도 범위를 재설정하기위해
23        // LatLngBounds 객체에 좌표를 추가합니다
24        bounds.extend(placePosition);
25
26        // 마커와 검색결과 항목에 mouseover 했을때
27        // 해당 장소에 인포윈도우에 장소명을 표시합니다
28        // mouseout 했을 때는 인포윈도우를 닫습니다
29        (function(marker, title) {
30            kakao.maps.event.addListener(marker, 'mouseover', function() {
31                displayInfowindow(marker, title);
32            });
33
34            kakao.maps.event.addListener(marker, 'mouseout', function() {
35                infowindow.close();
36            });
37
38            itemEl.onmouseover = function () {
39                displayInfowindow(marker, title);
40            };
41
42            itemEl.onmouseout = function () {
43                infowindow.close();
44            };
45        })(marker, places[i].place_name);
46
47        fragment.appendChild(itemEl);
48    }
49
50    // 검색결과 항목들을 검색결과 목록 Element에 추가합니다
51    listEl.appendChild(fragment);
52    menuEl.scrollTop = 0;
53
54    // 검색된 장소 위치를 기준으로 지도 범위를 재설정합니다
55    map.setBounds(bounds);
56 }
```

웹 - 게시판 페이지 소스

1) 게시판 데이터 베이스 연동

```
1 const BoradWrite = ({ userObj }) => {
2   const [nweet, setNweet] = useState("");
3   const [attachment, setAttachment] = useState();
4
5   const onSubmit = async (event) => { //동기&비동기 함수
6     event.preventDefault();
7     //뉴윗 추가
8     let attachmentUrl = "";
9     if (attachment != "") {
10      const attachmentRef = storageService
11        .ref()
12        .child(`${userObj.uid}/${uuidv4()}`);
13      const response = await attachmentRef.putString(attachment, "data_url");
14      attachmentUrl = await response.ref.getDownloadURL();
15    }
16    const nweetObj = { //글의 내용, 날짜, 유저의 아이디
17      text: nweet,
18      createdAt: Date.now(),
19      creatorId: userObj.uid,
20      attachmentUrl,
21    };
22    await dbService.collection("mweet").add(nweetObj); //비동기, dbService에 nweetbj 추가
23    setNweet("");
24    setAttachment("");
25  };
26  const onChange = (event) => { //값이 바뀌면 반영
27    const {
28      target: { value },
29    } = event;
30    setNweet(value);
31  };
32  const onFilechange = (event) => { //파일 읽기
33    const {
34      target: { files },
35    } = event;
36    const theFile = files[0];
37    const reader = new FileReader();
38    reader.onloadend = (finishedEvent) => {
39      const {
40        currentTarget: { result },
41      } = finishedEvent;
42      setAttachment(result);
43    };
44    reader.readAsDataURL(theFile);
45  };
46  const onClearAttachment = () => setAttachment(null);
```

앱 - 반려동물 데이터베이스 연동

1) 반려동물 캘린더 데이터베이스 연동



