

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		



날씨 정보 조회 프로젝트 보고서

1			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
목차			

1. 작품의 개요

시스템의 정의 및 개발배경
 활용방안 및 개발목표
 개발환경
 작품의 기능 및 제약사항

2. 시스템 구성도

전체 구성도(구조설계)

3. 사용자 매뉴얼

4. 작품 상세 설명서

상세 설계
 데모시스템 구현

5. 결론 및 향후 개선방향

2			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
1. 작품의 개요			

개발 동기 및 필요성

날씨과 관련된 정보를 조회할 수 있는 홈페이지를 만들어보고 싶다는 생각이 들어 만들게 되었습니다. 날씨마다 입고나갈 옷차림을 알 수 있으면 좋을 것 같아 개발하게 되었습니다.

개발시스템의 목표

웹을 이용하여 날씨정보를 보여주는 서비스를 개발하는 것이 최종 목표입니다.

제약사항

전국을 보여주기 위해서 해당 API 호출에 제한이 있으므로 어느정도 호출을 하게 되면 호출을 할 수 없음 그러므로 많은 곳을 서비스 할 수 없음

3			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
1. 개발환경			

하드웨어 자원

프로세서 : AMD RYZEN R5 1600 @3.6 GHz
 메모리 : 16GB
 그래픽카드: GeForce GTX 1060
 모니터 : 크로스오버 32ss QHD DP FREEDOM

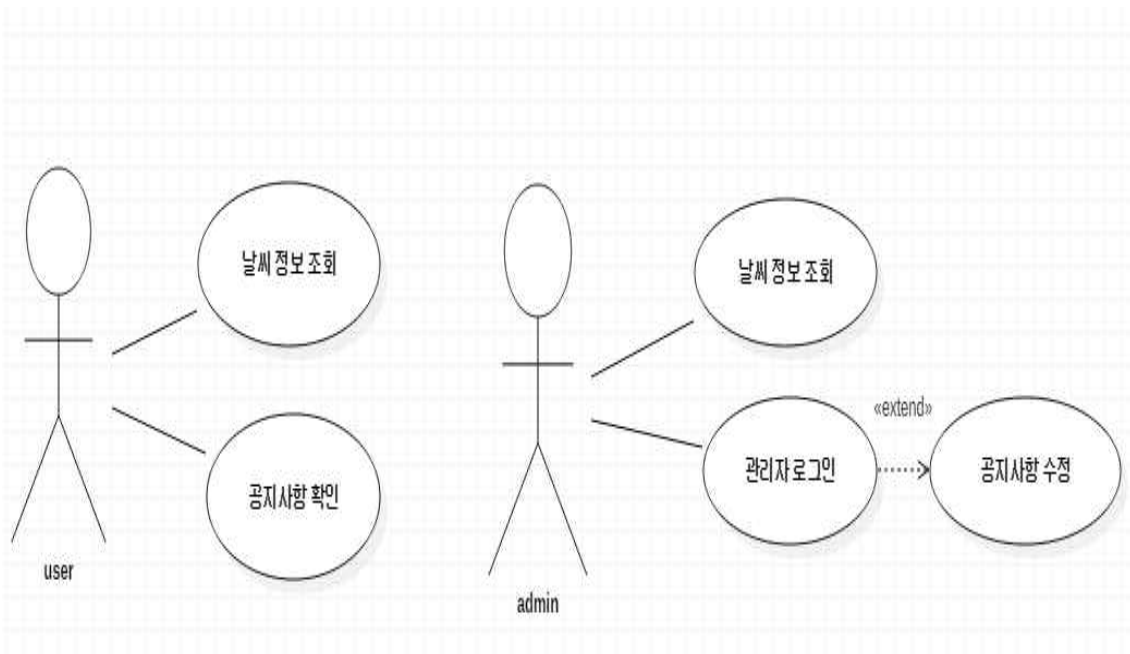
소프트웨어 자원

사용 툴 : IntelliJ IDEA, 그림판, Mysql workbanch

Client : Chrome

4			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
2. 시스템 구성도 (기능)			



속성	설명
날씨정보조회	메인 페이지에서 날씨정보를 조회할 수 있음
공지사항 확인	공지사항을 확인 할 수 있음
관리자 로그인	관리자는 로그인 할 수 있음
공지사항 수정	관리자는 공지사항을 수정 할 수 있음

5			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
3. 사용자 메뉴얼			

날씨 정보	Home	공지사항	관리자로그인
-------	------	------	--------

실시간 날씨 정보 사이트!

측정날짜	측정시간	위치	기온	날씨	낙뢰	강수형태	습도	풍향	풍속
20171219	1600	서울특별시	-3.1℃		없음	없음	39	W-NW (298)	2.1m/s
20171219	1600	경기도	-2.8℃		없음	없음	43	W-NW (309)	1.4m/s
20171219	1600	강원도	-2.6℃		없음	없음	35	S-SW (204)	2.0m/s
20171219	1600	충청북도	-3.2℃		없음	없음	37	W-NW (297)	1.3m/s
20171219	1600	전라남도	0.3℃		호림	없음	64	N-NE (16)	1.9m/s
20171219	1600	대구광역시	-0.5℃		없음	없음	29	W-NW (308)	2.9m/s
20171219	1600	부산광역시	4.4℃		없음	없음	32	N-NE (17)	1.4m/s
20171219	1600	제주특별자치도	4.5℃		호림	없음	75	NW (331)	3.8m/s

- ① 홈페이지 접속 시 보여주는 페이지
- ② 날씨정보를 확인 할 수 있음
- ③ 관리자로그인 버튼 클릭 시 로그인 모달 등장

6			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
3. 사용자 매뉴얼 (로그인)			

The screenshot shows a web application interface. At the top, there are navigation links: '날씨정보', 'Home', and '공지사항'. A green button labeled '관리자로그인' is in the top right. A white login modal is centered, titled '로그인'. It contains two input fields: '아이디' (ID) and '암호' (Password), with placeholder text '아이디' and '비밀번호' respectively. Below the password field is a '로그인' button. The background is a greyed-out table with weather data.

측정날짜	측정시간	종류	풍속
20171219	1600	W-NW (298)	2.1m/s
20171219	1600	W-NW (309)	1.4m/s
20171219	1600	S-SW (204)	2.0m/s
20171219	1600	W-NW (297)	1.3m/s
20171219	1600	N-NE (16)	1.9m/s
20171219	1600	W-NW (308)	2.9m/s
20171219	1600	N-NE (17)	1.4m/s
20171219	1600	NW (331)	3.8m/s

① 아이디 란에 등록된 이메일 입력

② 암호 란에 등록된 패스워드 입력

③ 로그인 버튼 클릭 시 본인 이메일과 패스워드가 일치 할 경우
로그인이 성공함 실패 시 실패알람 등장

7			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
3. 사용자 매뉴얼 (공지사항)			

날씨정보	Home	공지사항	admin : 로그아웃	공지사항
------	------	------	--------------	------

공지사항

제목 : 공지사항

작성자 : admin || 작성날짜 : 2017-12-19 17:41:02.0

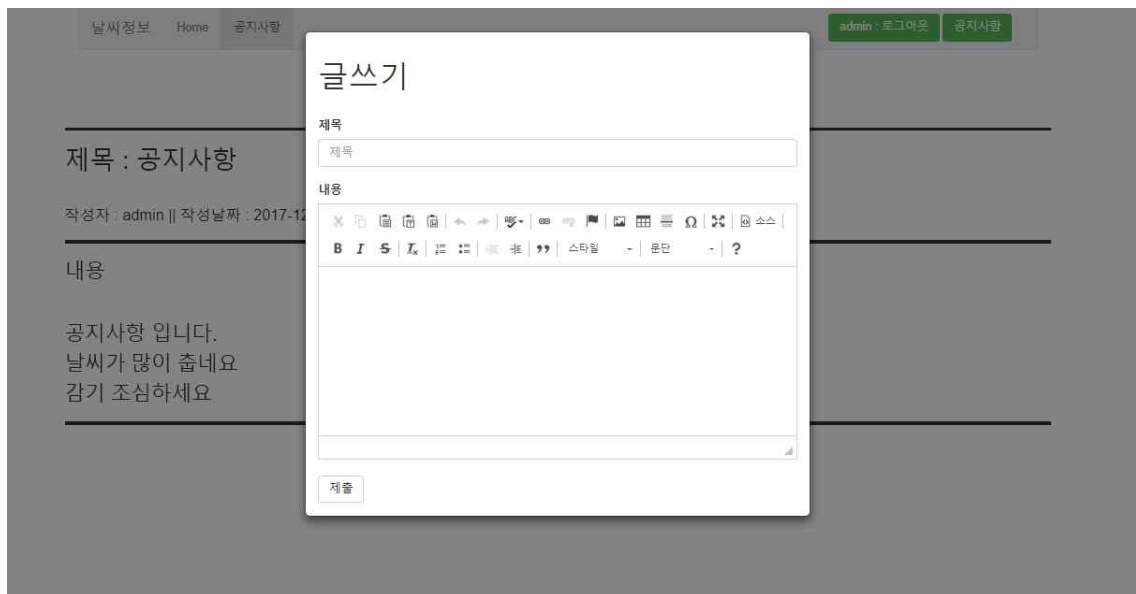
내용

공지사항 입니다.
날씨가 많이 춥네요
감기 조심하세요

- ① 공지사항 버튼으로 해당 화면으로 이동할 수 있음
- ② 공지사항을 확인 할 수 있음
- ③ 관리자는 공지사항 버튼을 통해서 공지사항을 변경 할 수 있음

8			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
3. 사용자 매뉴얼 (게시물 작성)			

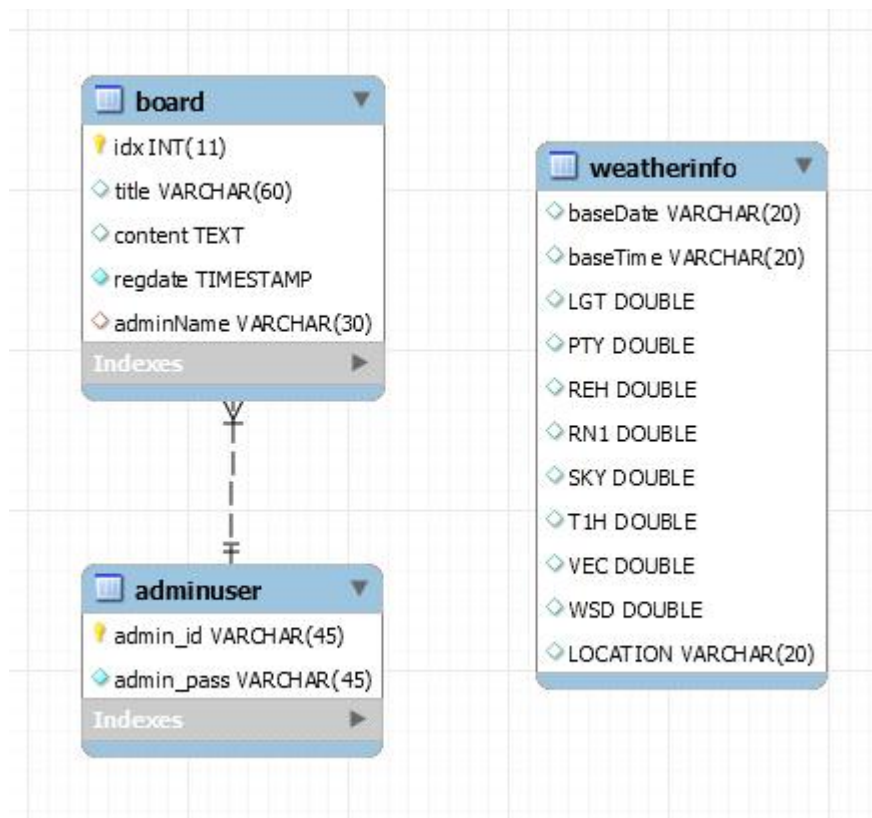


- ① 공지사항 버튼을 선택 시 해당 모달 출력
- ② 각 입력란에 값들을 작성 할 수 있음
- ③ 모든 입력 값들을 입력 후 제출 버튼을 선택할 경우 해당 공지사항으로 변경함

9			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 작품 상세 설명서 (Mysql)			

Mysql



10			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 작품 상세 설명서 (Mysql)			

Mysql (weatherinfo)

속성	설명	타입
baseDate	실측날짜	STRING
baseTime	실측시간	STRING
LGT	낙뢰 없음(0), 있음(1)	INT
PTY	강수형태 없음(0), 비(1), 진눈개비(2), 눈(3)	INT
REH	습도	INT
RN1	1시간 강수량	DOUBLE
SKY	하늘상태 맑음(1), 구름조금(2), 구름많음(3), 흐림(4)	INT
T1H	1시간 기온	DOUBLE
VEC	풍향 0 ~ 360	INT
WSD	풍속	DOUBLE
LOCATION	관측위치	STRING

Mysql (board)

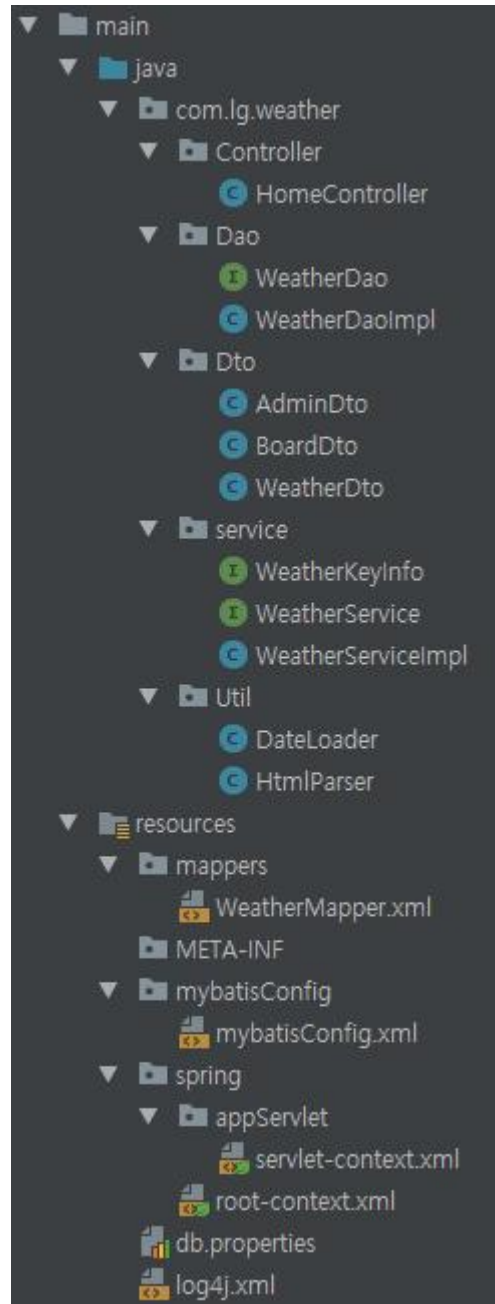
속성	설명	타입
idx	게시물 고유 번호	INT
title	제목	VARCHAR
content	내용	TEXT
regdate	등록일자	TIMESTAMP
adminuser	작성자 관리	VARCHAR

Mysql (adminuser)

속성	설명	타입
admin_id	관리자 아이디 저장	VARCHAR
admin_pass	패스워드 정보 저장	VARCHAR

11			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 작품 상세 설명서 (Spring 구조)			



12			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

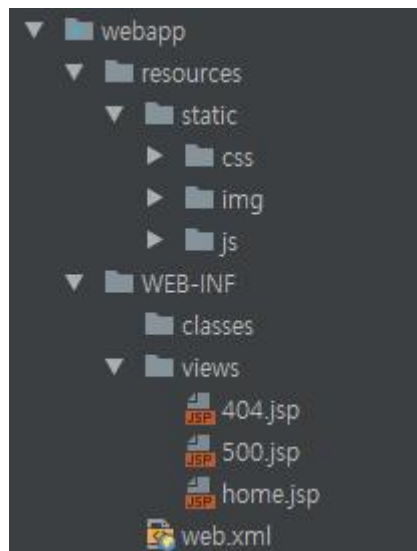
날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 작품 상세 설명서 (Spring 구조)			

Spring 구조 설명

속성	설명
HomeController	메인페이지 컨트롤러
WeatherDao	관련 SQL 관리 인터페이스
WeatherDaoImpl	관련 SQL 관리
AdminDto	관리자 DTO
BoardDto	공지사항 DTO
WeatherDto	날씨 정보 DTO
WeatherKeyinfo	동네예보조회서비스 서비스키 및 관리지역
WeatherService	날씨 정보 제공 인터페이스
WeatherServiceImpl	날씨 정보 제공 서비스 구현
DateLoader	날짜 와 시간 출력
HtmlParser	html 파서
WeatherMapper	Weather 관련 SQL문 정보 관리
mybatisConfig	mybatis 설정 정보
servlet-context	servlet 설정 정보
root-context	root 설정 정보
db.properties	DB 접속정보 관리

13			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 작품 상세 설명서 (Spring 구조)			



설명

속성	설명
css	bootstrap css 관리
img	이미지 관리 (옷차림, 날씨아이콘)
js	js 관리 (bootstrap, jquery)
404.jsp	404 에러 페이지
500.jsp	500 에러 페이지
home.jsp	메인화면

14			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 주요 코드			

```

@Override
public void insertinfo(WeatherDto weatherDto) throws Exception {
    String now = new DateLoader().DateLoader();

    for (int j=0; j<REGION.length; j++) {
        weatherStrUrl.setLength(0);

        weatherStrUrl.append("http://newsky2.kma.go.kr/service/SecndSrtpdFrstInfoService2/ForecastGrib?serviceKey=" +
            SERVICEKEY + "&base_date=" + now.substring(0, 8) +
            "&base_time=" + now.substring(9, 13));
        weatherStrUrl.append("&nx=" + REGION[j][0] + "&ny=" + REGION[j][1]
            + "&numOfRows=10&pageSize=10&pageNo=1&startPage=1&_type=json");
        String jsonSouce = new
            HtmlParser().HtmlParser(weatherStrUrl.toString());

        try {
            JSONParser jsonParser = new JSONParser();
            JSONObject jsonObject = (JSONObject)
            jsonParser.parse(jsonSouce);
            JSONObject response = (JSONObject) jsonObject.get("response");
            JSONObject body = (JSONObject) response.get("body");
            JSONObject items = (JSONObject) body.get("items");
            JSONArray item = (JSONArray) items.get("item");

            JSONObject weatherInfo = (JSONObject) item.get(0);
            weatherDto.setBaseDate(weatherInfo.get("baseDate").toString());
            weatherDto.setBaseTime(weatherInfo.get("baseTime").toString());
            weatherDto.setLOCATION(LOCATION[j]);

            for (int i = 0; i < item.size(); i++) {
                weatherInfo = (JSONObject) item.get(i);
                if (weatherInfo.get("category").equals("T1H")) {

```

```

weatherDto.setT1H(Double.parseDouble(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("RN1")) {

weatherDto.setRN1(Double.parseDouble(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("SKY")) {

weatherDto.setSKY(Integer.parseInt(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("PTY")) {

weatherDto.setPTY(Integer.parseInt(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("LGT")) {

weatherDto.setLGT(Integer.parseInt(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("VEC")) {

weatherDto.setVEC(Integer.parseInt(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("WSD")) {

weatherDto.setWSD(Double.parseDouble(weatherInfo.get("obsrValue").toString()));
        } else if (weatherInfo.get("category").equals("REH")) {

weatherDto.setREH(Integer.parseInt(weatherInfo.get("obsrValue").toString()));
        }
    }
} catch (Exception e) {
    e.getMessage();
}
weatherDao.insertinfo(weatherDto);
}
}

```

이름	설명
weatherStrUrl	해당 URL을 통해서 API 호출
JsonParser	JSON 객체에 쉽게 접근하기 위한 파서
weatherDao.insertinfo	호출한 정보를 DB에 저장

15			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
4. 주요 코드			

```

$('#login-btn').click(function() {
    $.ajax({
        type: "POST",
        url: "/loginProcessing",
        data: {
            "username" : $('#login-id').val(),
            "password" : $('#login-pw').val(),
        },
        success: function() {
            alert('로그인 성공');
            location.reload();
        }, error: function() {
            alert('로그인 정보가 올바르지 않습니다. ');
        }
    });
});

$('#create').click(function() {
    CKupdate();
    $.ajax({
        type: "POST",
        url: "/createBoard",
        data: {
            "title" : $('#title').val(),
            "text" : $('#editor').val(),
            "id" : $('#adminid').val(),
        },
        success: function() {
            alert('게시글 등록 성공');
            location.reload();
        }, error: function() {
            alert('게시글 등록 실패');
        }
    });
});

```

이름	설명
ajax	ajax를 이용하여 로그인 처리, 게시글 등록
CKupdate()	Ckeditor를 사용할 경우 인스턴스 업데이트 후 실행 하여함

16			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근

날씨 정보 조회 프로젝트 보고서			
작성일자	2017.12.19.	작성자	임 근
프로젝트명	날씨 정보 조회		
5. 결론 및 향후 개선방향			

결론
<p>1. 날씨정보 조회서비스 API를 이용할 수 있는 좋은 계기가 되었습니다.</p> <p>2. 해당 API를 사용하기 좋은 방법을 더 고민해야 한다고 생각했습니다.</p>
느낀점과 향후 개선방향
<p>일정 기간 동안 해당 프로젝트를 진행하며 날씨 관련 API 호출이 어떤 방식인지 알 수 있었습니다. 앞으로 추가할 것이 있다면 모든 지역을 검색하여 날씨 정보를 조회하는 것입니다.</p>

17			
프로젝트기간	2017.12.06.~2017.12.19	작성자	임 근