

# 우산 알리미 최종보고서



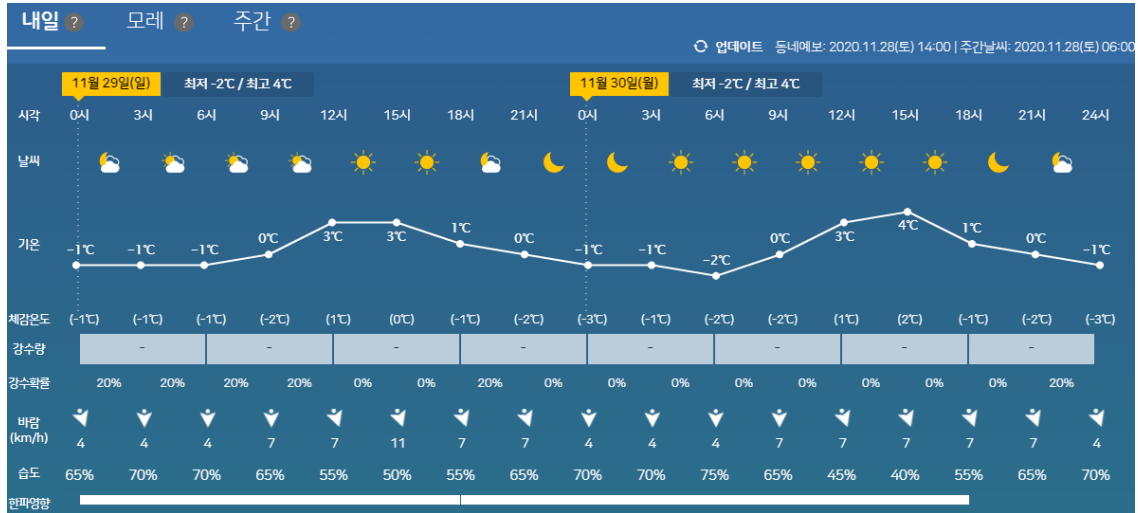
과 목 명	소프트웨어 공학
담 당 교 수	강동현 교수님
학 과	컴퓨터공학과
조 명	F 조
조 원	박민철 20163051 정영빈 20163083
제 출 일	2020.11.23

## 목차

<b>1. 프로젝트 개요 .....</b>	<b>2</b>
1) 프로젝트 소개 .....	2
2) 개발목적 .....	2
3) 기대효과 .....	2
4) 프로그램 기능 요약 .....	2
5) 개발환경 .....	2
<b>2. 구조적 명세 .....</b>	<b>3</b>
<b>3. 요구사항 .....</b>	<b>5</b>
1) 기능적 요구사항 .....	5
2) 비기능적 요구사항 .....	5
3)도메인 요구사항 .....	5
<b>4. Use-case .....</b>	<b>5</b>
4.1 Use-case diagram .....	5
4.2 Use-case description .....	6
<b>5. Database modeling .....</b>	<b>7</b>
5.1 Database model .....	7
5.2 Database Description .....	7
<b>6. Class Diagram .....</b>	<b>8</b>
6.1 Class Diagram .....	8
6.2 Class Diagram Description .....	8
<b>7. Sequence Diagram .....</b>	<b>11</b>
7.1 로그인 .....	11
7.2 회원가입 .....	12
7.3 로그아웃 .....	13
7.4 메일신청 .....	13
7.5 메일 전송 취소 .....	14
7.6 메일 전송 .....	14
<b>8 테스트 .....</b>	<b>15</b>
8.1 테스트 케이스 및 시나리오 .....	15
8.2 제약사항 테스트 케이스 .....	23
8.3 데이터베이스 연결 테스트 .....	24

# 1. 프로젝트 개요

## 1) 프로젝트 소개



[ 그림 1 기상청 날씨정보 ]

이 프로그램은 기상청 API 를 활용하여 강수 확률을 파악하고 우산 지참 여부를 메일을 통해 알려 주는 우산 알리미 프로그램이다. 예를 들어 출근하는 날 아침, 중요한 약속 1 시간 전 등 사용자가 원하는 시간을 설정하면 해당 시간에 강수량을 파악하여 우산의 지참 여부를 알려준다.

## 2) 개발목적

외출을 했을 때 갑자기 비가 쏟아져서 당황했던 경험이 있을 것이다. 이때 급하게 근처 편의점에서 우산을 구매하거나 비를 맞으며 이동할 수 밖에 없다. 외출하기 전 날씨를 확인하기 위해서는 기상 뉴스를 보거나 날씨 애플리케이션을 실행해야 하지만 바쁜 현대인의 삶 특성상 매일 확인하는 것은 매우 어려운 일이다. 누군가 미리 비가 온다고 알려 준다면 이러한 불편함을 해소할 수 있다고 생각하여 '우산 알리미'를 제작하고자 한다.

## 3) 기대효과

스마트폰의 기본 애플리케이션, 기상청 등 많은 날씨를 제공하는 애플리케이션들이 이미 시중에 존재한다. 하지만 기존의 애플리케이션들은 프로그램을 실행해야만 날씨 정보를 확인할 수 있는 번거로움이 존재한다. 하지만 우산 알리미는 한번 설정을 완료하면 강수확률이 존재하는 경우 별도의 사용자 조작 없이 우산을 지참하라는 메일을 전송한다. 메일을 수신함으로써 우산을 챙기지 않아 비를 맞아야 하는 상황을 사전에 방지할 수 있다. 또한 웹 애플리케이션으로 동작하여 스마트폰과 컴퓨터 환경이 인터넷에 연결되어 있다면 우산알리미에 접속할 수 있으며 언제 어디서나 알림을 설정하고 메일을 수신할 수 있다.

## 4) 프로그램 기능 요약

- 첫 화면에서 로그인할 수 있다.
- 사용자는 메일이 전송될 시간과 강수 확률 정보를 확인할 지역을 자유롭게 설정하고 변경할 수 있다.
- 강수 확률이 존재하는 경우 사용자의 이메일로 메일을 전송한다.
- 더 이상 메일을 받고 싶지 않은 경우 신청취소 버튼을 클릭하여 취소할 수 있다.

## 5) 개발환경

OS	WINDOWS 10 EDU
DB	mysql Ver 8.0.21 for Win64 on x86_64 (MySQL Community Server - GPL)
Front-end Language	javascript – react 라이브러리
Back-end Language	Node.js
Development IDE	Visual studio code

## 2. 구조적 명세

### 1) 회원가입

기능	사용자의 계정을 생성한다.
설명	사용자에게 아이디와 비밀번호, 이메일을 입력 받아 데이터베이스에 등록한다. 동일한 아이디가 데이터베이스에 이미 등록된 경우 경고를 출력한다. 모든 데이터의 입력과 확인을 완료한 후 회원가입 버튼을 클릭하여 데이터베이스에 저장할 수 있다. 이때 비밀번호는 hash 함수를 통해 암호화된다.
입력	사용자의 아이디, 비밀번호, 이메일
출처	사용자가 입력한 문자열
출력	회원가입에 성공할 경우 "회원가입 성공"을 출력하고, 로그인 화면으로 이동한다. 회원가입에 실패할 경우 "회원가입 실패"를 출력한다.
목적지	데이터베이스
동작	사용자가 입력한 비밀번호를 무작위로 생성한 솔트를 통해 hash 함수로 암호화하고, 사용자가 입력한 아이디, 이메일, 솔트, 암호화된 비밀번호를 데이터베이스에 저장한다.
필요사항	사용자가 아이디, 비밀번호, 이메일을 모두 입력해야 한다.
사전조건	사용자가 입력하는 아이디가 데이터베이스에 존재하지 않아야 한다.
사후조건	데이터베이스에 사용자가 입력한 데이터가 정확하게 저장되어야 한다.

### 2) 로그인

기능	아이디와 비밀번호를 입력하여 메인 화면으로 이동한다.
설명	사용자가 입력한 아이디와 비밀번호를 데이터베이스에서 검색하여 일치하는 데이터가 존재하지 않을 경우 경고를 출력한다. 데이터가 존재할 경우 메인 신청 화면으로 이동한다.
입력	사용자가 입력한 아이디, 비밀번호
출처	사용자가 입력한 문자열
출력	로그인에 성공할 경우 메인 신청 화면으로 이동한다. 로그인에 실패할 경우 로그인 실패를 출력한다.
목적지	데이터베이스
동작	사용자가 입력한 아이디를 데이터베이스에서 검색하여 솔트와 암호화된 비밀번호를 반환한다. 사용자가 입력한 비밀번호를 솔트를 이용하여 hash 함수로 암호화하고 데이터베이스에 저장된 비밀번호와 비교한다. 일치하는 경우 메인 화면으로 이동하고, 일치하지 않는 경우 경고를 출력한다.
필요사항	사용자가 아이디와 비밀번호를 모두 입력해야 한다.
사전조건	회원가입에 성공한 아이디와 비밀번호가 존재해야 한다.
사후조건	로컬 스토리지에 로그인 인증 정보를 저장하고 메인 화면으로 이동한다.

### 3) 로그아웃

기능	로그인 화면으로 이동하고 메인 화면의 접근을 차단한다.
설명	사용자가 로그아웃 버튼을 클릭할 경우 로그인 인증 정보를 삭제하고 로그인 화면으로 이동한다.
입력	사용자가 로그아웃 버튼을 클릭한다.
출처	사용자가 마우스가 클릭한 로그아웃 버튼
출력	로그인 화면으로 이동
목적지	로그인 화면
동작	로컬스토리지에 저장된 로그인 인증 정보를 삭제하고 로그인 화면으로 이동한다.
필요사항	로그인 버튼을 클릭해야 한다.
사전조건	로컬 스토리지에 로그인 인증 정보가 존재해야 한다.

사후조건	로그인 화면으로 이동한다.
------	----------------

#### 4) 알림 메일 신청

기능	알림 메일을 신청한다.
설명	사용자가 알림 메일이 전송될 시간과 날씨 정보를 확인할 지역을 메인 화면의 리스트에서 선택하고 메일신청 버튼을 클릭하여 데이터베이스에 저장을 요청한다.
입력	사용자가 선택한 시간과 지역 데이터
출처	사용자가 리스트에서 선택한 시간과 지역 데이터
출력	데이터베이스에 정상적으로 저장된 시간과 지역 데이터
목적지	데이터베이스
동작	시간과 지역을 모두 선택하고 메일신청을 클릭할 경우 데이터베이스에 저장하고 결과를 출력한다. 데이터가 모두 선택되지 않은 경우 경고를 출력한다.
필요사항	시간과 지역 데이터가 모두 선택되어야 한다.
사전조건	사용자가 선택할 지역과 시각이 리스트에 저장되어 있어야 한다.
사후조건	사용자가 선택한 정보가 정상적으로 데이터베이스에 저장되어야 한다.

#### 5) 알림 메일 전송

기능	알림 메일을 전송한다.
설명	현재 시각이 정각이 될 경우 해당 시각에 전송해야 하는 사용자를 데이터베이스에서 검색한다. 해당 사용자가 입력한 지역의 날씨 정보를 기상청 API 로 검색하여 메일로 전송한다.
입력	기상청 기상 데이터
출처	기상청
출력	사용자의 이메일로 날씨 정보 전송
목적지	사용자의 이메일
동작	매시 정각마다 알림 메일을 전송할 사용자를 검색한다. 해당 사용자가 입력한 지역을 기상청 API 로 검색하여 메일을 전송한다.
필요사항	요청한 기상 데이터에서 강수확률이 존재할 경우 우산을 지참하라는 메일을 전송한다.
사전조건	사용자의 이메일과 알림 메일 요청정보를 미리 알고 있어야 한다.
사후조건	사용자가 신청취소 버튼을 클릭하기 전까지 메일을 전송해야 한다.

#### 6)알림 메일 신청 취소

기능	알림 메일 신청을 취소한다.
설명	사용자가 신청취소 버튼을 클릭할 경우 알림 메일의 신청을 취소하고, 신청 화면으로 이동한다.
입력	사용자가 신청취소 버튼을 클릭한다.
출처	사용자가 입력한 마우스
출력	메일 신청 화면
목적지	데이터베이스
동작	사용자의 메일 신청 상태를 취소로 변경하고 메일 신청화면으로 이동한다.
필요사항	데이터베이스에 메일 신청 상태가 정상적으로 변경되어야 한다.
사전조건	사용자가 메일을 신청한 상태여야 한다.
사후조건	메일을 수신할 지역과 날씨를 확인할 지역을 다시 선택할 수 있어야 한다.

### 3. 요구사항

#### 1) 기능적 요구사항

메일이 전송될 시간대를 설정할 수 있다.
강수 확률을 확인할 지역을 선택할 수 있다.
신청취소 버튼을 클릭하여 메일 신청을 취소할 수 있다.
메일이 전송될 시각과 기상정보를 확인할 지역을 선택하고 신청버튼을 클릭하여 신청한다.
로그인, 로그아웃이 가능하다.
사용자가 설정한 시각에 날씨정보를 확인하여 강수확률이 존재하는 경우 강수확률 정보와 날씨정보를 전송한다.

#### 2) 비기능적 요구사항

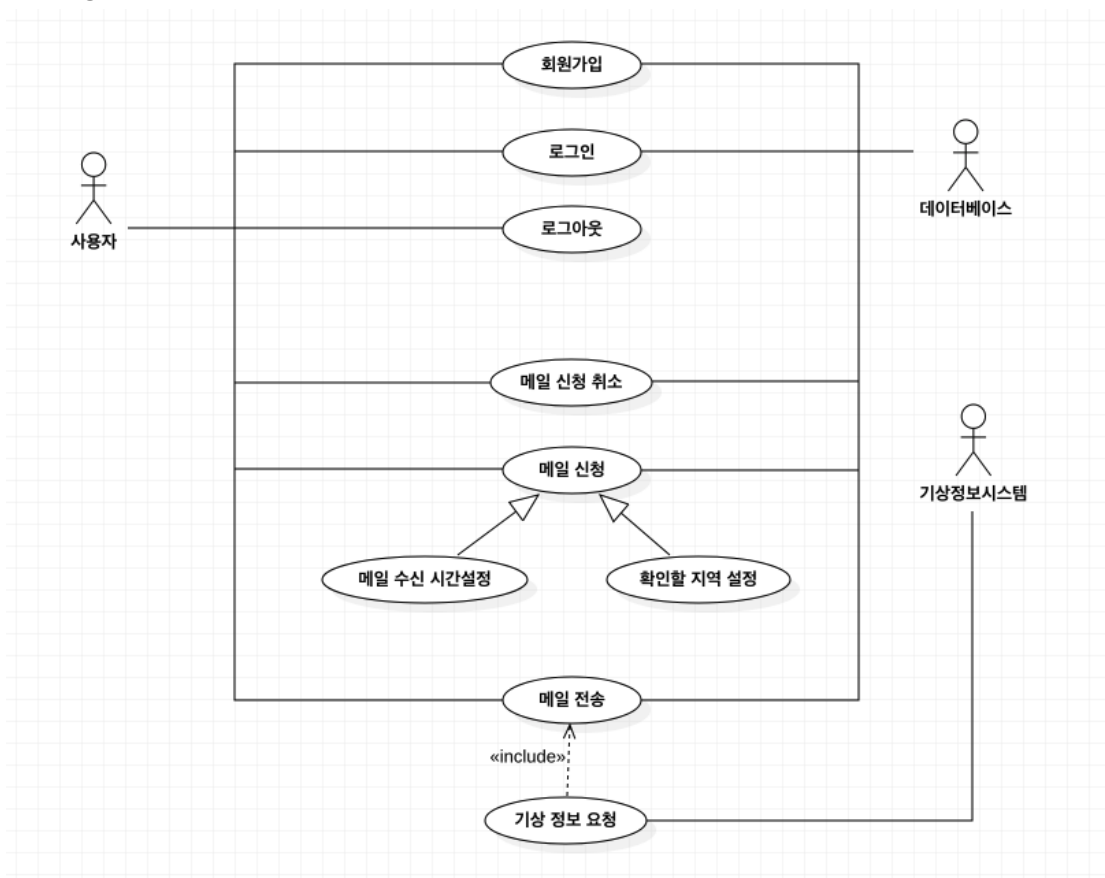
로그인된 이후 모든 기능을 사용할 수 있다.
메일 신청 버튼을 클릭한 이후 메일 수신이 가능하다.
메일 전송 시 사용자가 설정한 시간에서 5 분 이내에 전송이 완료되어야 한다.
메일 신청은 24 시간 단위로 매시 정각으로만 설정이 가능하다.
아이디는 4~12 자의 영문과 숫자만 입력 가능하며 중복은 불가능하다.

#### 3)도메인 요구사항

웹 애플리케이션 환경에서 동작해야 한다.
기상청에서 제공하는 날씨 API 를 이용해야 한다.
DBMS 는 MYSQL 을 이용해야 한다.

### 4. Use-case

#### 4.1 Use-case diagram



[그림 2. use-case diagram]

## 4.2 Use-case description

### 1) 회원가입

system	우산 알리미
Use case	회원가입
Actor	사용자, 데이터베이스
Description	사용자가 아이디, 비밀번호, 이메일을 입력하여 회원가입을 요청한다.
Stimulus	사용자가 입력한 데이터를 데이터베이스에 저장을 요청한다.
Response	입력한 아이디가 중복되지 않을 경우 전송된 데이터를 데이터베이스에 저장하고 결과를 전송한다.
Comments	이미 등록된 아이디를 중복해서 사용할 수 없으며 영문과 숫자만 입력할 수 있다. 비밀번호는 영문, 숫자만 사용할 수 있다. 이메일은 인증번호를 전송하여 이메일을 인증해야 한다.

### 2) 메일 신청

system	우산 알리미
Use case	메일 환경설정
Actor	사용자, 데이터베이스
Description	메일이 전송될 시간과 강수확률을 확인할 지역을 설정한다.
Stimulus	강수확률을 확인할 지역과 메일이 전송될 시간을 리스트에서 선택하고 신청하기 버튼을 클릭하여 데이터베이스에 저장을 요청한다.
Response	데이터베이스에 전송된 데이터를 저장하고 저장 결과를 출력한다.
Comments	알림 메일이 전송될 시간을 설정할 때 24 시간 단위로 매시 정각으로만 설정할 수 있다.

### 3) 메일 신청 취소

system	우산 알리미
Use case	신청 취소
Actor	사용자, 데이터베이스
Description	사용자가 메일을 신청한 상태에서 신청 취소 버튼을 클릭하여 메일 신청을 취소한다.
Stimulus	사용자가 신청취소 버튼을 클릭하여 메일 신청 취소를 요청한다.
Response	데이터베이스에 저장된 알림메일 신청 상태를 취소하고 메일 신청화면으로 돌아간다.
Comments	신청화면으로 돌아간 이후 메일을 수신할 지역과 날씨를 확인할 지역을 다시 선택할 수 있어야 한다.

### 4)기상 정보 요청

system	우산 알리미
Use case	기상 정보 요청
Actor	기상정보 시스템
Description	기상청 API 를 통해 날씨 정보를 확인한다.
Stimulus	매시 정각마다 해당 시각에 메일을 요청한 사용자를 데이터베이스에서 검색한다.
Response	기상청 API 를 통해 데이터베이스에서 응답된 사용자가 선택한 지역의 날씨정보를 확인한다.
Comments	신청화면으로 돌아간 이후 메일을 수신할 지역과 날씨를 확인할 지역을 다시 선택할 수 있어야 한다.

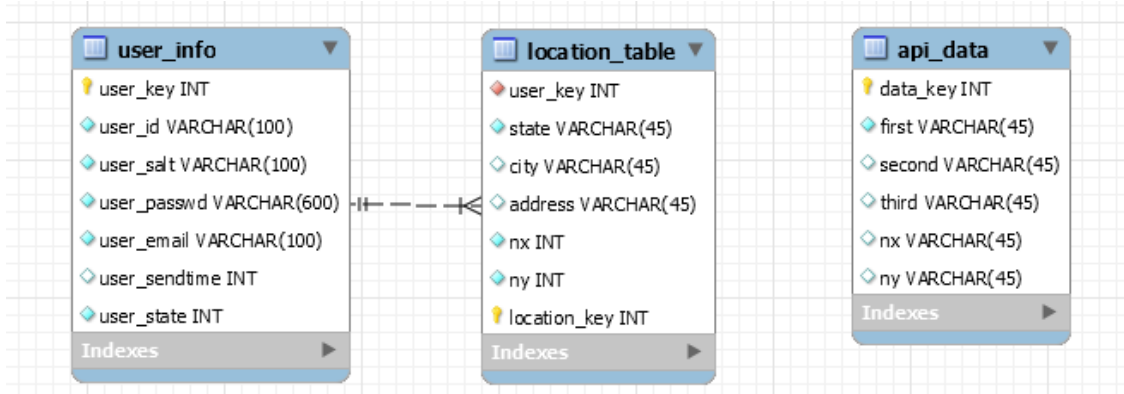
### 5) 메일 전송

System	우산 알리미
Use case	메일 전송
Actor	데이터베이스, 사용자

Description	기상청 API 에서 응답된 데이터를 분석하여 비가 예상되는 경우 우산지참 알림을 메일로 전송한다.
Stimulus	기상청 API 에서 응답된 데이터를 분석한다.
Response	강수확률이 존재하는 경우 사용자의 이메일로 우산 지참 알림을 전송한다.
Comments	사용자가 설정한 시간에서 5 분 이내에 메일의 전송이 완료되어야 한다.

## 5. Database modeling

### 5.1 Database model



[그림 3. 데이터베이스 모델]

### 5.2 Database Description

#### 1) user\_info: 사용자정보 테이블

Column	DataType	Description
user_key	INT	주 식별자로 사용되며 auto_increment 를 통해 1 씩 증가된다.
user_id	VARCHAR	사용자가 입력한 아이디이며 중복이 불가능한 unique 값을 가진다.
user_salt	VARCHAR	비밀번호의 hash 암호화에 사용될 key 이며 무작위로 생성한 값
user_passwd	VARCHAR	사용자가 입력한 비밀번호를 salt 를 이용하여 생성된 hash 암호
user_email	VARCHAR	사용자가 입력한 이메일
user_sendtime	INT	사용자가 입력한 메일을 수신할 시간
user_state	INT	이메일 수신 신청 여부

#### 2) location\_table: 사용자 위치 정보 테이블

Column	DataType	Description
location_key	INT	사용자 위치 정보 테이블의 주 식별자로 사용된다.
user_key	INT	사용자 정보 테이블에서 상속된 속성
state	VARCHAR	사용자가 입력한 도 단위의 주소
city	VARCHAR	사용자가 입력한 시 단위의 주소
address	VARCHAR	사용자가 입력한 동, 읍, 리 단위의 주소
nx	INT	사용자가 입력한 주소의 기상청 위도
ny	INT	사용자가 입력한 주소의 기상청 경도

#### 3) api\_data: 기상청 좌표 테이블

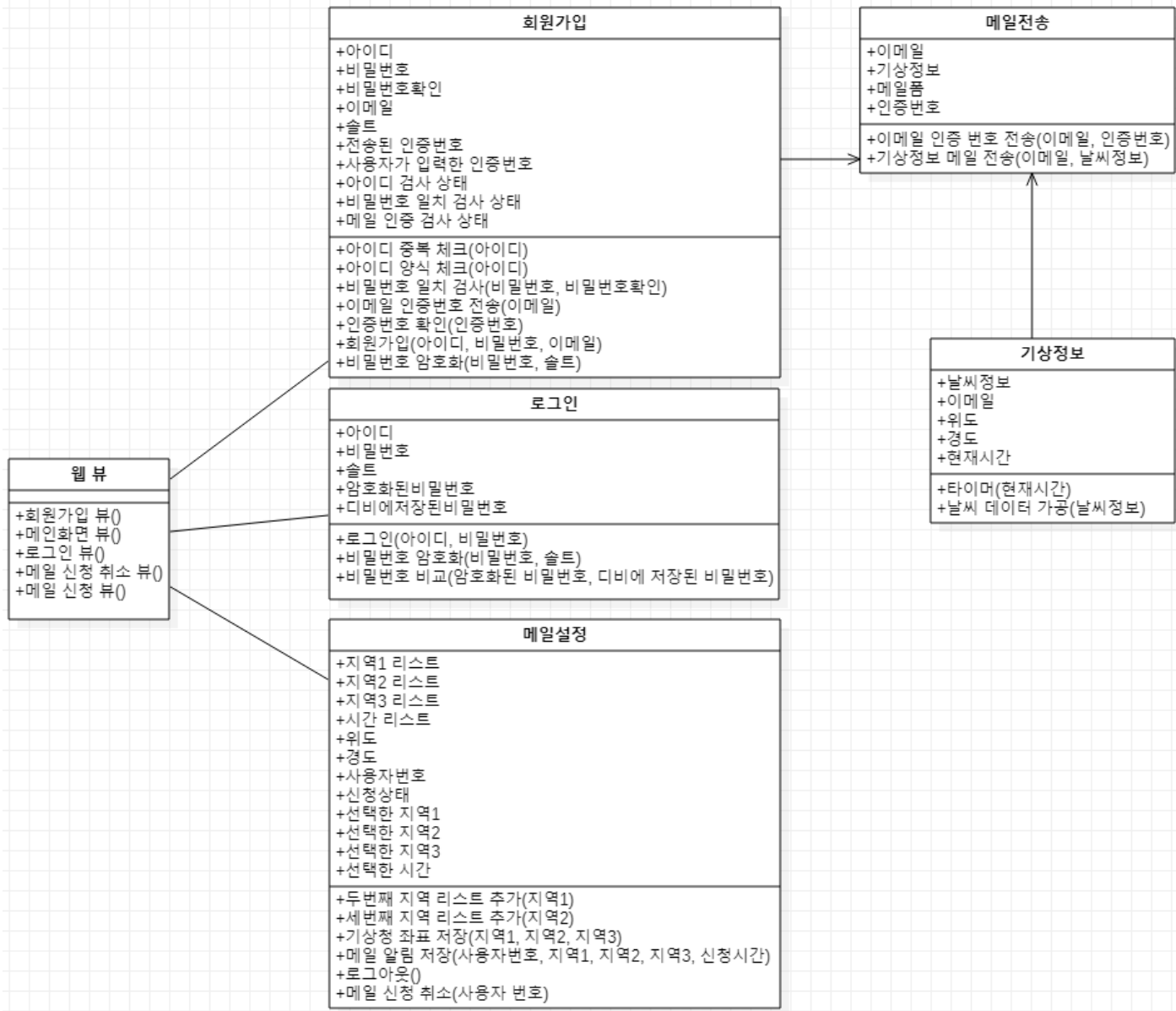
Column	DataType	Description
data_key	INT	좌표 테이블의 주 식별자
first	VARCHAR	도 단위의 주소
second	VARCHAR	시 단위의 주소
third	VARCHAR	동, 읍, 리 단위의 주소



nx	INT	기상청에서 사용하는 위도
ny	INT	기상청에서 사용하는 경도

## 6. Class Diagram

### 6.1 Class Diagram



[그림 4. 클래스 다이어그램]

### 6.2 Class Diagram Description

#### 1) 회원가입 클래스

- 회원가입과 관련된 기능을 수행한다.
- 웹 뷰 클래스와 개념상 연결된 연관 관계이며, 메일 전송 클래스와 단방향 연관 관계이다.

- 변수

변수명	선언	자료형	기본값	설명
아이디	property	String	null	아이디 입력란에 입력되는 문자열
비밀번호	property	String	null	비밀번호 입력란에 입력되는 문자열
비밀번호 확인	property	String	null	비밀번호 확인란에 입력되는 문자열
이메일	property	String	null	이메일 입력란에 입력되는

솔트	let	Number	-	hash 암호화에 사용되는 무작위로 생성한 키
전송된 인증번호	var	String	null	무작위로 생성한 이메일 인증번호
사용자가 입력한 인증번호	property	Number	-	인증번호 입력란에 입력되는 문자열
아이디 검사 상태	property	Boolean	false	아이디 중복 검사 완료 상태
비밀번호 일치 검사 상태	property	Boolean	false	비밀번호와 비밀번호 확인 일치 검사 상태
메일 인증 검사 상태	property	Boolean	false	메일 인증 완료 상태

#### - 함수

함수명	파라미터	함수 설명
아이디 중복 체크	아이디	이미 데이터베이스에 등록된 아이디인지 검사한다.
아이디 양식 체크	아이디	아이디의 양식을 검사한다.
비밀번호 일치 검사	비밀번호, 비밀번호 확인	두 비밀번호가 일치하는지 검사한다.
이메일 인증번호 전송	이메일	메일 전송 클래스에 메일 전송을 요청한다.
인증번호 확인	인증번호	인증번호가 일치하는지 검사한다.
회원가입	아이디, 비밀번호, 이메일	데이터베이스에 저장한다.
비밀번호 암호화	비밀번호, 솔트	비밀번호를 hash 암호화한다.

### 2) 로그인 클래스

- 로그인과 관련된 기능을 수행한다.

- 웹 뷰 클래스와 개념상 연결된 연관 관계이다.

#### - 변수

변수명	선언	자료형	기본값	설명
아이디	property	String	null	아이디 입력란에 입력되는 문자열
비밀번호	property	String	null	비밀번호 입력란에 입력되는 문자열
솔트	let	Number	-	hash 암호화에 사용되는 무작위로 생성한 키
암호화된 비밀번호	let	String	null	hash 로 암호화된 사용자가 입력한 비밀번호
디비에 저장된 비밀번호	let	String	null	데이터베이스에 저장된 사용자의 비밀번호

#### - 함수

함수명	파라미터	함수 설명
로그인	아이디, 비밀번호	로그인을 요청한다.
비밀번호 암호화	비밀번호, 솔트	사용자가 입력한 비밀번호를 hash 암호화한다.
비밀번호 비교	암호화된 비밀번호, 디비에 저장된 비밀번호	두 비밀번호가 일치하는지 검사한다.

### 3) 메일 설정 클래스

- 메인 화면의 기능을 수행하며 메일의 신청과 취소, 로그아웃의 기능이 있다.

- 웹 뷰 클래스와 개념상 연결된 연관 관계이다.

#### - 변수

변수명	선언	자료형	기본값	설명
지역 1 리스트	property	Object	-	도 단위의 지역을 출력하는 리스트
지역 2 리스트	property	Object	-	시 단위의 지역을 출력하는 리스트
지역 3 리스트	property	Object	-	동, 읍, 리 단위의 지역을 출력하는 리스트
시간 리스트	property	Object	-	시간을 출력하는 리스트
위도	property	Number	-	사용자가 선택한 지역의 기상청 위도 좌표
경도	property	Number	-	사용자가 선택한 지역의 기상청 경도 좌표

사용자 번호	JSON	Number	-	데이터베이스에 저장된 사용자의 user_key
신청상태	property	boolean	null	사용자의 메일 신청 상태
선택한 지역 1	property	String	null	사용자가 선택한 도 단위의 지역
선택한 지역 2	property	String	null	사용자가 선택한 시 단위의 지역
선택한 지역 3	property	String	null	사용자가 선택한 동, 읍, 리 단위의 지역
선택한 시간	property	String	null	사용자가 선택한 시간

#### - 함수

함수명	파라미터	함수 설명
두번째 지역 리스트 추가	지역 1	지역 2를 두번째 지역 리스트에 추가
세번째 지역 리스트 추가	지역 2	지역 3을 세번째 지역 리스트에 추가
기상청 좌표 저장	지역 1, 지역 2, 지역 3	사용자가 선택한 지역의 기상청 좌표 저장
메일 알림 저장	지역 1, 지역 2, 지역 3, 신청시간	데이터베이스에 저장
메일 신청 취소	사용자 번호	메일 신청 상태 취소
로그아웃	-	로그아웃을 실행한다.

#### 4) 웹 뷰

- 요청에 따라 화면과 링크를 이동시킨다.

- 회원가입 클래스, 회원 클래스, 메일설정 클래스와 개념상 서로 연결된 연관관계이다

#### - 함수

함수명	파라미터	함수 설명
회원가입 뷰	-	회원가입 화면을 출력하고 URL 을 이동시킨다.
메인 화면 뷰	-	메인 화면을 출력하고 URL 을 이동시킨다.
로그인 뷰	-	로그인을 출력하고 URL 을 이동시킨다.
메일 신청 취소 뷰	-	메일 신청을 완료한 경우 메일 신청 취소를 메인 화면으로 출력한다.
메일 신청 뷰	-	메일 신청을 취소한 경우 메일 신청 화면을 메인 화면으로 출력한다.

#### 5) 메일 전송

- 사용자의 이메일로 메일을 전송하는 기능을 수행한다.

- 기상정보 클래스, 회원가입 클래스와 개념상 서로 연결된 연관 관계이다.

#### - 변수

변수명	선언	자료형	기본값	설명
이메일	let	String	null	메일을 전송할 이메일
기상정보	var	Object	-	사용자가 선택한 지역의 기상정보
메일 품	let	Object	-	메일을 전송할 양식
인증번호	let	String	null	무작위로 생성한 이메일 인증번호

#### - 함수

함수명	파라미터	함수 설명
이메일 인증 번호 전송	이메일, 인증번호	사용자의 이메일로 인증번호를 전송한다.
기상정보 메일 전송	이메일, 날씨정보	사용자의 이메일로 우산 알림을 전송한다.

#### 6) 기상정보

- 매시 정각마다 기상청 API 에서 기상 정보를 요청하고 가공하는 기능을 수행한다.

- 메일 전송 클래스와 단방향 연관 관계이다.

## - 변수

변수명	선언	자료형	기본값	설명
날씨 정보	var	Object	-	사용자가 선택한 지역의 기상정보
이메일	let	String	null	메일을 전송할 이메일
위도	let	Number	-	사용자가 선택한 지역의 기상청 위도 좌표
경도	let	Number	-	사용자가 선택한 지역의 기상청 경도 좌표
현재시간	var	String	null	시스템의 현재시간

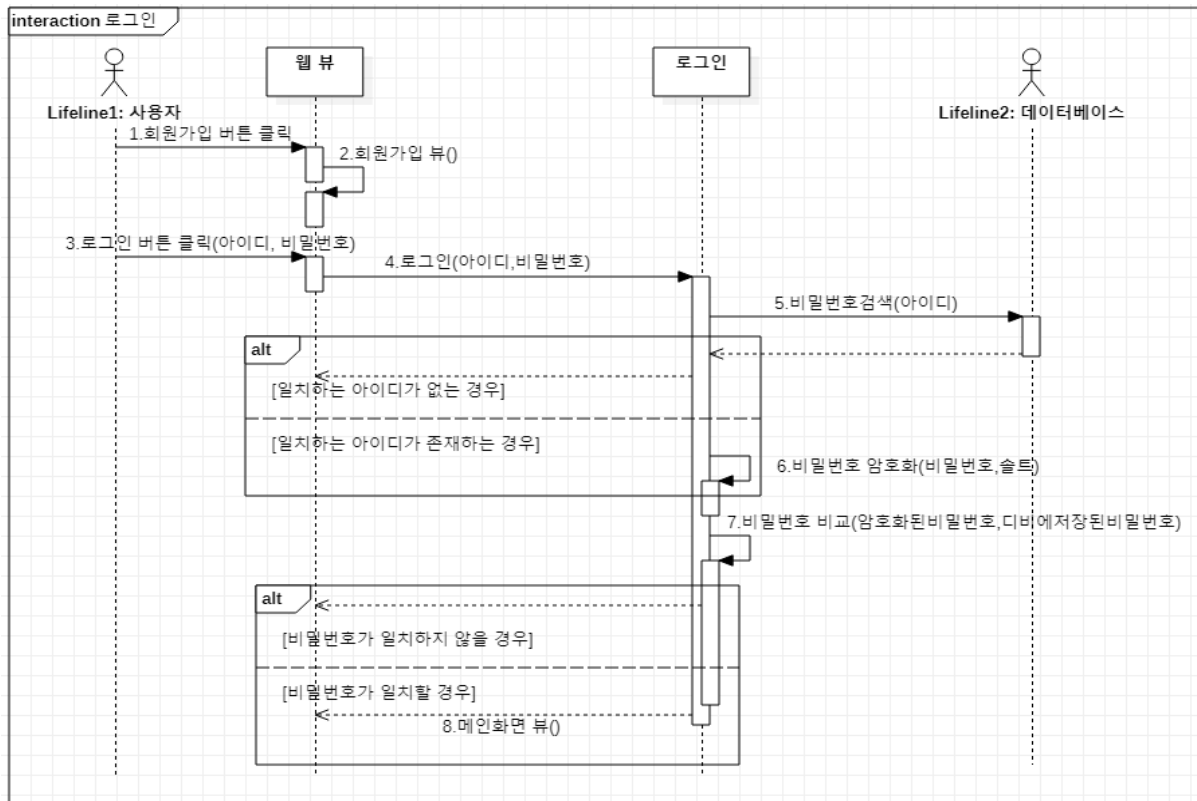
## - 함수

함수명	파라미터	함수 설명
타이머	현재시간	현재시간이 정각인지 확인한다.
날씨 데이터 가공	날씨 정보	기상청 API 에서 응답 받은 날씨정보를 가공한다.

## 7. Sequence Diagram

### 7.1 로그인

사용자의 로그인을 수행하는 함수의 호출 과정을 Sequence Diagram 으로 보여준다.

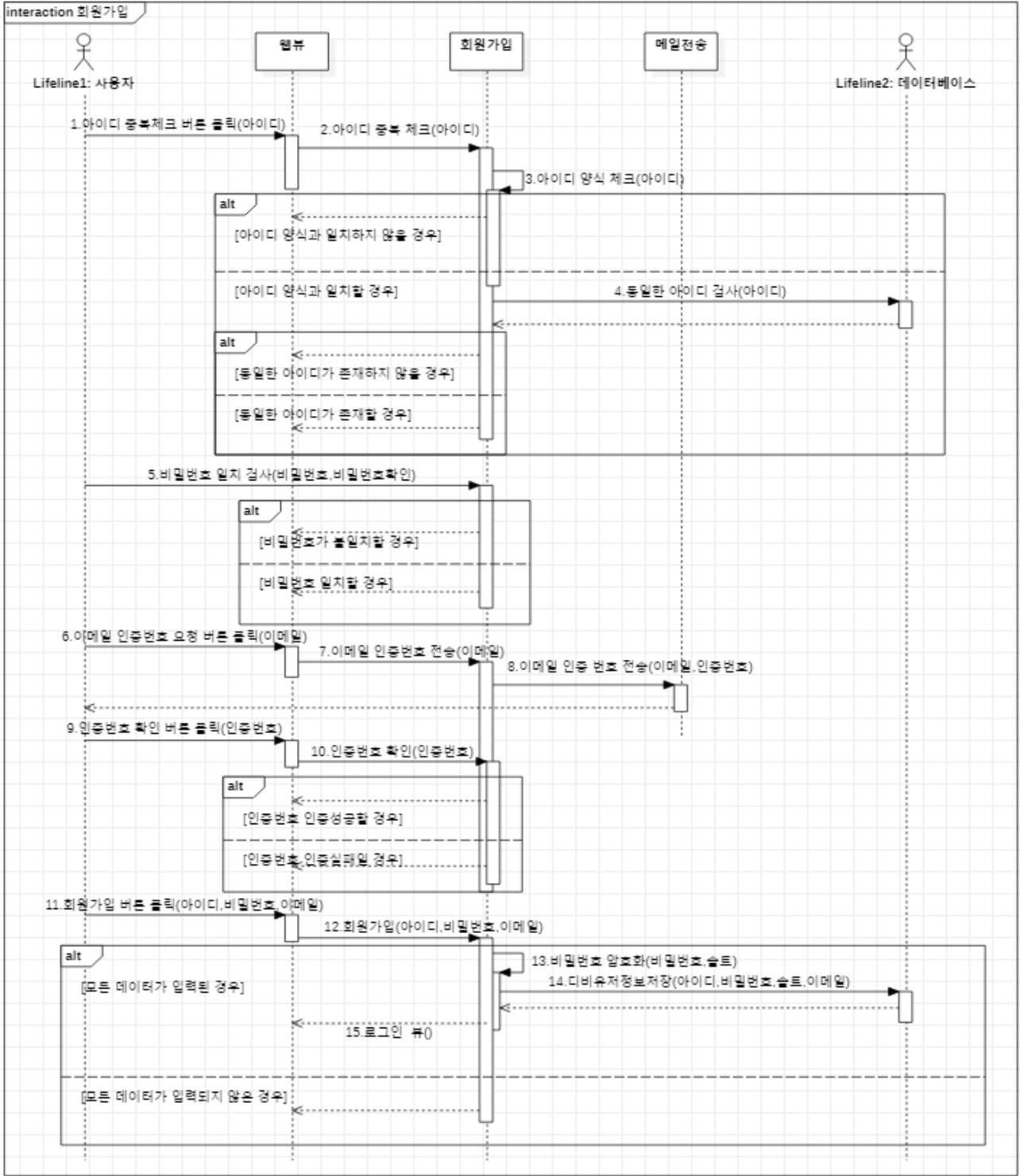


[그림 5 로그인 시퀀스 다이어그램]

1. 사용자가 회원가입 버튼을 클릭하여 회원가입 화면을 요청한다.
2. 회원가입 뷰 함수를 실행하여 회원가입 화면을 출력한다.
3. 사용자가 아이디와 비밀번호를 입력하고 로그인 버튼을 클릭한다.
4. 웹뷰 클래스가 회원 클래스에 로그인을 요청한다.
5. 데이터베이스에 사용자가 입력한 아이디의 비밀번호와 솔트를 검색한다.
6. 일치하는 아이디가 존재할 경우 사용자가 입력한 비밀번호를 암호화한다.
7. 암호화된 비밀번호와 데이터베이스에 저장된 비밀번호를 비교한다.
8. 두 비밀번호가 일치할 경우 메인화면 뷰 함수를 실행하여 메인화면으로 이동한다.

## 7.2 회원가입

사용자의 회원가입을 수행하는 함수의 호출 과정을 Sequence Diagram 으로 보여준다.



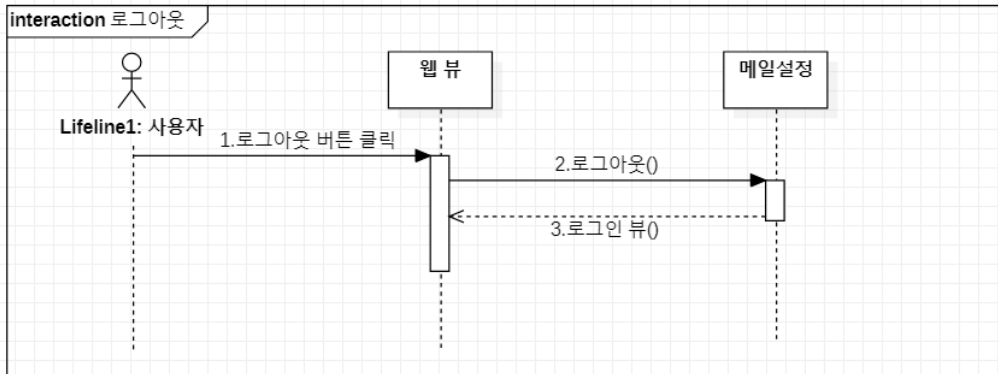
[그림 6 회원가입 시퀀스 다이어그램]

- |  |
|--|
| 1. 사용자가 중복체크 버튼을 클릭하여 아이디의 확인을 요청한다.                 |
| 2. 웹 뷰 클래스가 회원가입 클래스의 아이디 중복 체크 함수를 실행한다.            |
| 3. 아이디가 4~12 글자 사이, 숫자와 대소문자만 사용하고 있는지 아이디 양식을 점검한다. |
| 4. 아이디 양식과 일치할 경우 데이터베이스에 동일한 아이디가 존재하는지 확인한다.       |
| 5. 사용자가 비밀번호와 비밀번호 확인을 입력할 경우 비밀번호 일치 검사 함수를 실행한다.   |
| 6. 사용자가 이메일을 입력하고 이메일 인증번호 요청 버튼을 클릭한다.              |
| 7. 웹 뷰 클래스가 회원가입 클래스에 이메일 인증번호 전송을 요청한다.             |

8. 인증번호를 생성하고 이메일 인증번호 전송 함수를 실행하여 사용자에게 메일을 전송한다.
9. 사용자가 인증번호를 입력하고 인증번호 확인 버튼을 클릭한다.
10. 웹 뷰 클래스가 회원 클래스에 인증번호 확인을 요청한다.
11. 사용자가 모든 데이터를 입력하고 회원가입 버튼을 클릭한다.
12. 웹 뷰 클래스가 회원 클래스의 회원가입 함수를 실행한다.
13. 비밀번호를 랜덤으로 생성한 솔트값을 이용하여 hash 암호화한다.
14. 아이디, 암호화된 비밀번호, 솔트, 이메일을 데이터베이스에 저장한다.
15. 회원 클래스가 웹 뷰 클래스에 로그인 화면으로 이동을 요청한다.

### 7.3 로그아웃

사용자의 로그아웃을 수행하는 함수의 호출 과정을 Sequence Diagram 으로 보여준다.

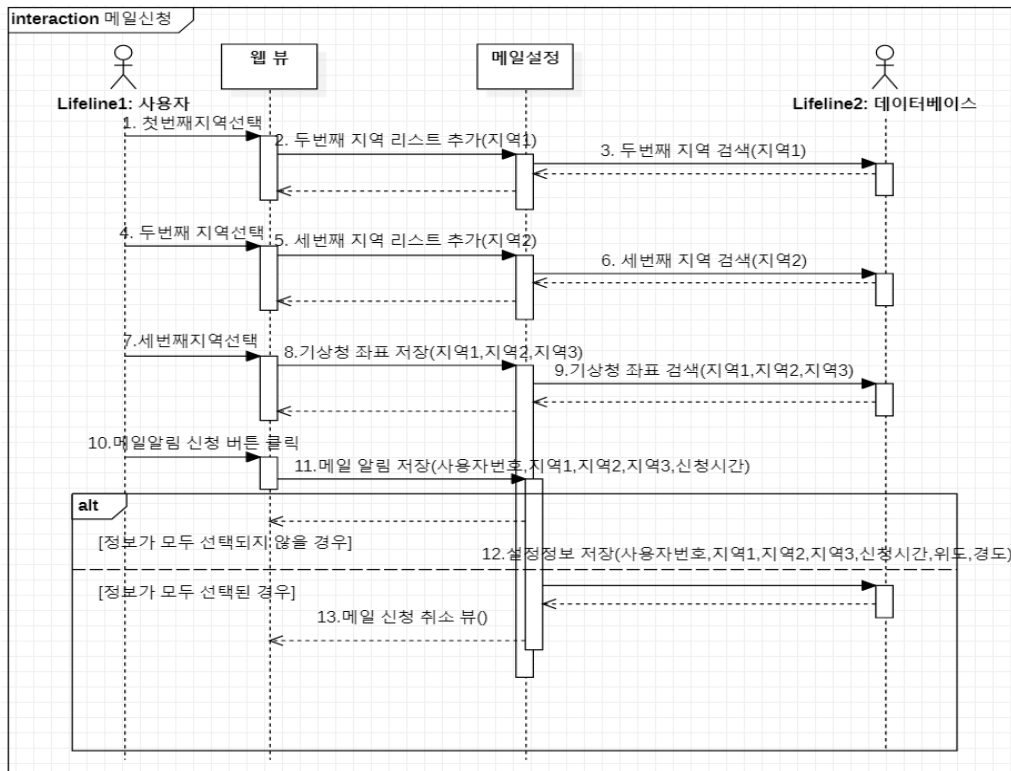


[그림 7 로그아웃 시퀀스 다이어그램]

1. 사용자가 로그아웃 버튼을 클릭하여 로그아웃을 요청한다.
2. 웹 뷰 클래스가 회원 클래스의 로그아웃 함수를 실행한다.
3. 회원 클래스가 로그인 뷰 함수 실행을 요청한다.

### 7.4 메일신청

사용자의 메일을 신청하기 위한 함수의 호출 과정을 Sequence Diagram 으로 보여준다.



[그림 8 메일신청 시퀀스 다이어그램]

- 용어 설명

사용자가 선택한 도 단위의 지역은 지역 1 라고 부른다.

사용자가 선택한 시 단위의 지역은 지역 2 라고 부른다.

사용자가 선택한 동, 읍, 리 단위의 지역은 지역 3 라고 부른다.

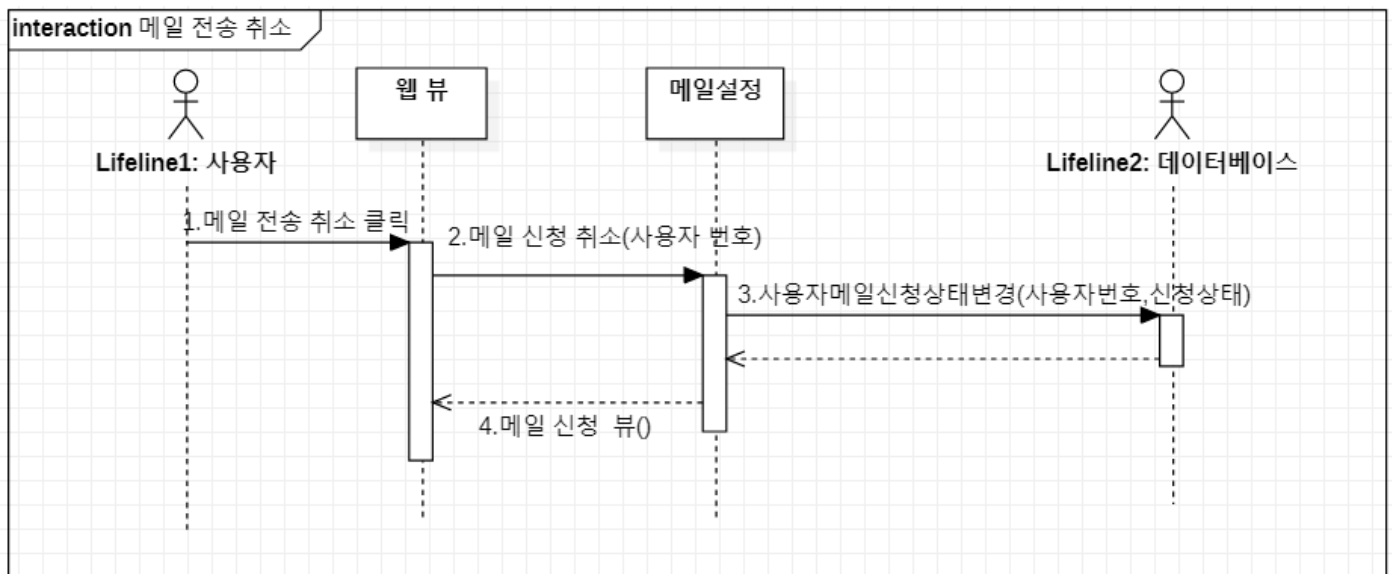
두번째 지역 리스트는 지역 2 를 선택할 수 있는 리스트이다.

두번째 지역 리스트는 지역 3 를 선택할 수 있는 리스트이다.

1. 사용자가 지역 1 을 리스트에서 선택한다.
2. 웹 뷰 클래스가 메일설정 클래스의 두번째 지역 출력함수를 실행한다.
3. 사용자가 선택한 지역 1 에 해당하는 지역 2 를 데이터베이스에서 검색한다.
4. 사용자가 지역 2 를 리스트에서 선택한다.
5. 웹 뷰 클래스가 메일설정 클래스의 세번째 지역 출력함수를 실행한다.
6. 사용자가 선택한 지역 2 에 해당하는 지역 3 을 데이터베이스에서 검색한다.
7. 사용자가 지역 3 를 리스트에서 선택한다.
8. 웹뷰 클래스가 메일설정 클래스의 기상청 좌표 저장 함수를 실행한다.
9. 사용자가 선택한 지역에 해당하는 기상청 좌표를 데이터베이스에서 검색한다.
10. 사용자가 모든 데이터를 선택하고 메일 알람 신청 버튼을 클릭한다.
11. 웹 뷰 클래스가 메일 설정 클래스의 메일 알람 저장 함수를 실행한다.
12. 모든 정보가 입력된 경우 데이터베이스에 저장한다.
13. 웹 뷰 클래스에 메일 신청화면으로 이동한다.

## 7.5 메일 전송 취소

사용자의 메일 전송 취소 기능을 수행하기 위한 함수의 호출 과정을 Sequence Diagram 으로 보여준다.

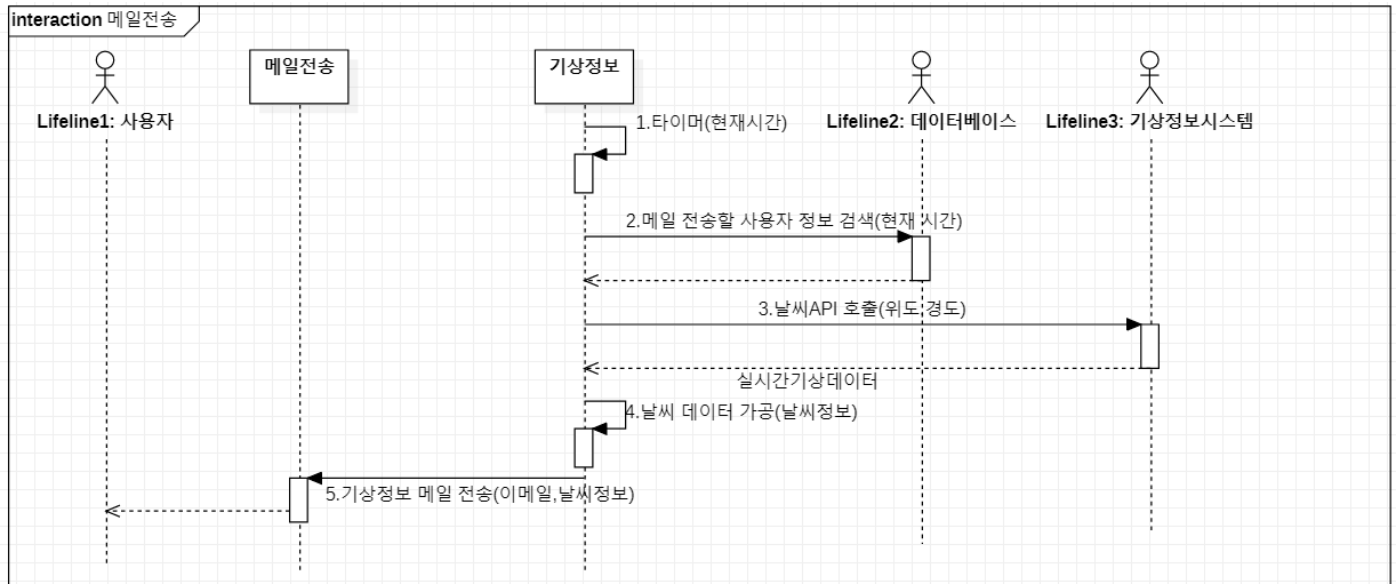


[그림 9. 메일전송 취소 시퀀스 다이어그램]

1. 사용자가 메일 전송 취소 버튼을 클릭한다.
2. 웹 뷰 클래스가 메일설정 클래스이 메일 전송 취소 함수를 실행한다.
3. 로컬 스토리지에 저장된 사용자 번호와 메일 신청 상태를 데이터베이스에 전송하여 메일 신청 상태를 수정한다.
4. 웹뷰 클래스에 메인화면으로 이동을 요청한다.

## 7.6 메일 전송

사용자에게 알람 메일을 전송하는 함수의 호출 과정을 Sequence Diagram 으로 보여준다.



[그림 10. 메일전송 시퀀스 다이어그램]

1. 타이머 함수를 통해 현재시각이 정각인지 확인한다.
2. 데이터베이스에 현재 시간을 전송하여 메일을 전송할 사용자를 검색한다.
3. 데이터베이스에서 검색된 사용자의 지역 정보를 이용하여 기상정보 시스템을 호출한다.
4. 기상 정보 시스템에서 전송된 데이터를 가공한다.
5. 기상정보 메일 전송 함수를 이용하여 사용자에게 기상 알림 메일을 전송한다.

## 8 테스트

### 8.1 테스트 케이스 및 시나리오

우산 알리미의 요구사항이 반영되었는지 검증하기 위한 기능테스트와 테스트 시나리오에 기반한 비기능 테스트를 수행한다.

기능	테스트 케이스 수
회원가입	36
로그인	11
메일 신청	22
메일 취소	6
메일 전송	12
합계	87

흰색	회색	청회색
비정상적인 입력	정상적인 입력	결과 실패

#### 1) 회원가입 기능 테스트 케이스

기능	테스트 조건	예상 결과	결과
아이디 중복체크	아이디를 4 자 이상 12 자 이하 영어와 숫자로만 입력한 경우	성공 출력	PASS
	아이디를 4 자 미만으로 입력한 경우	경고 출력	PASS
	아이디를 12 자 초과하여 입력한 경우	경고 출력	PASS
	아이디에 특수문자를 포함하여 입력한 경우	경고 출력	PASS
	아이디에 공백을 포함하여 입력한 경우	경고 출력	PASS
	아이디에 영어 이외의 언어를 입력한 경우	경고 출력	PASS



	아이디에 대문자를 포함하여 입력한 경우	성공 출력	PASS
	데이터베이스에 존재하는 아이디를 입력한 경우	경고 출력	PASS
	서버와 통신이 불량인 경우	경고 출력	FAIL
비밀번호 입력	비밀번호를 99 자 이하로 공백을 포함하지 않고 입력한 경우	정상 입력	PASS
	비밀번호를 100 자 이상으로 입력한 경우	경고 출력	PASS
	비밀번호에 99 자 이하로 공백을 포함하여 입력한 경우	경고 출력	PASS
	비밀번호와 비밀번호 확인에 아무것도 입력하지 않은 경우	동작 안함	PASS
	서버와 통신이 불량인 경우	정상 동작	PASS
비밀번호 확인 입력	공백이 아닌 비밀번호와 일치하는 경우	성공 출력	PASS
	비밀번호 확인 이후 비밀번호를 변경한 경우	실패 출력	PASS
	비밀번호 확인에 100 글자 이상 입력한 경우	실패 출력	PASS
	비밀번호 확인에 공백을 추가한 경우	실패 출력	PASS
	서버와 통신이 불량인 경우	정상 동작	PASS
이메일 인증번호 신청	100 자 미만으로 이메일을 입력한 경우	성공 출력	PASS
	100 자 이상으로 이메일을 입력한 경우	실패 출력	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
	사용자가 입력한 이메일로 인증번호 전송	전송 성공	PASS
	서버와 연결이 불량인 경우	에러 출력	FAIL
이메일 인증번호 확인	이메일 인증번호가 일치하는 경우	성공 출력	PASS
	이메일 인증번호가 일치하지 않는 경우	실패 출력	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
회원가입 신청	모든 입력과 확인을 완료한 경우	성공 출력	PASS
	모든 입력을 완료하지 않은 경우	실패 출력	PASS
	모든 입력을 완료하고 확인을 하지 않은 경우	실패 출력	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
암호화	무작위로 생성되는 salt 출력	정상 출력	PASS
	비밀번호를 salt 를 이용하여 암호화하여 로그 출력할 경우	정상 출력	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL

- 아이디 중복체크

4 자 미만으로 아이디를 입력할 경우	13 자 초과로 입력한 경우
<p>localhost:3000의 메시지</p> <p>아이디는 4~12자의 영문 대소문자와 숫자로만 입력가능합니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>111</p>	<p>localhost:3000의 메시지</p> <p>아이디는 4~12자의 영문 대소문자와 숫자로만 입력가능합니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>aaaaaaaaaaaaaaaaaaaaaaaaaaaa</p>

아이디에 특수문자 추가한 경우	아이디에 공백을 추가한 경우
<p>localhost:3000의 메시지</p> <p>아이디는 4~12자의 영문 대소문자와 숫자로만 입력가능합니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>aa@aa</p>	<p>localhost:3000의 메시지</p> <p>아이디는 4~12자의 영문 대소문자와 숫자로만 입력가능합니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>aa aa</p>
아이디에 영어 이외의 언어를 입력한 경우	이미 사용중인 아이디를 입력할 경우
<p>localhost:3000의 메시지</p> <p>아이디는 4~12자의 영문 대소문자와 숫자로만 입력가능합니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>aaa토리</p>	<p>localhost:3000의 메시지</p> <p>이미 사용중인 아이디 입니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>admin1</p>
4 ~12 자 사이 소문자, 숫자만 입력할 경우	4~12 자 사이 대문자, 숫자만 입력할 경우
<p>localhost:3000의 메시지</p> <p>사용가능한 아이디 입니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>admin2</p>	<p>localhost:3000의 메시지</p> <p>사용가능한 아이디 입니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>ADMIN2</p>
이미 사용 중인 아이디를 대문자로 변경해서 입력한 경우	
<p>localhost:3000의 메시지</p> <p>이미 사용중인 아이디 입니다.</p> <p>확인</p> <p>아이디 중복확인</p> <p>ADMIN1</p>	

#### - 비밀번호 입력

비밀번호가 일치하는 경우	비밀번호가 일치하지 않는 경우
<p>비밀번호</p> <p>....</p> <p>비밀번호확인</p> <p>.... </p> <p>비밀번호가 일치합니다.</p>	<p>비밀번호</p> <p>....</p> <p>비밀번호확인</p> <p>.....</p> <p>비밀번호가 일치하지 않습니다.</p>
비밀번호에 100 자 이상 입력한 경우	비밀번호에 아무것도 입력하지 않은 경우
<p>localhost:3000의 메시지</p> <p>아이디와 비밀번호의 길이가 너무 길니다!!</p> <p>확인</p> <p>아이디 중복확인</p> <p>아이디를 입력하세요.</p> <p>비밀번호</p> <p>.....</p>	<p>비밀번호</p> <p>비밀번호를 입력하세요.</p> <p>비밀번호확인</p> <p>비밀번호를 한번더 입력하세요.</p> <p>비밀번호가 일치하지 않습니다.</p>

## - 이메일 인증

100 자 이하의 이메일을 입력한 경우	100 자 이상의 이메일을 입력한 경우
<p>localhost:3000의 메시지</p> <p>인증 메일이 전송되었습니다.</p> <p>확인</p>	<p>localhost:3000의 메시지</p> <p>이메일은 100자까지만 입력이 가능합니다</p> <p>확인</p>
이미 등록된 이메일을 입력하는 경우	이메일 인증이 실패한 경우
<p>localhost:3000의 메시지</p> <p>이미 가입된 메일입니다.</p> <p>확인</p>	<p>localhost:3000의 메시지</p> <p>인증실패</p> <p>확인</p>
이메일 인증이 성공한 경우	
<p>localhost:3000의 메시지</p> <p>인증성공</p> <p>확인</p>	

## - 회원가입

아이디 중복검사를 하지 않은 경우	비밀번호가 일치하지 않은 경우
<p>localhost:3000의 메시지</p> <p>아이디 중복검사를 해주세요</p> <p>확인</p>	<p>localhost:3000의 메시지</p> <p>비밀번호가 일치하지 않습니다.</p> <p>확인</p>
이메일 중복체크를 하지 않은 경우	모든 확인이 완료된 경우
<p>localhost:3000의 메시지</p> <p>메일 인증을 해주세요</p> <p>확인</p>	<p>localhost:3000의 메시지</p> <p>회원가입 성공</p> <p>확인</p>

## - 암호화

랜덤으로 생성된 salt 와 hash 암호화의 결과
<p>랜덤으로 생성된 salt</p> <p>1371306980837</p> <p>암호화된 비밀번호</p> <p>6ada2331638e040aabcce6108f5c46f2ef42df87b1e8b2b6c933ec71fec07ec9e2364a9f23953e37672020369a27b92b7a01041af745576bf34df2f445356d520</p>

## 2) 로그인 기능 테스트 케이스

기능	테스트 조건	예상 결과	결과
로그인 신청	로컬 스토리지에 등록하지 않은 상태로 로그인 하지 않고 url 에 /main 을 추가하여 이동한 경우	실패 출력	PASS
	로컬 스토리지에 등록된 상태로 로그인 하지 않고 url 에 /main 을 추가하여 이동한 경우	메인 화면 출력	PASS
	데이터베이스에 등록된 아이디와 비밀번호를 입력한 경우	성공 출력	PASS
	데이터베이스에 등록되지 않은 아이디를 입력한 경우	실패 출력	PASS
	데이터베이스에 등록된 비밀번호와 일치하지 않은 경우	실패 출력	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
메인화면	로그인을 완료한 사용자가 메일을 신청한 경우	신청 취소 화면	PASS

출력	로그인을 완료한 사용자가 메일을 신청하지 않은 경우	메일 신청 화면	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
암호화	비밀번호를 salt 를 이용하여 암호화하여 로그 출력할 경우	로그 출력	PASS
	비밀번호를 잘 못 입력한 경우 데이터베이스의 값과 일치하는가	성공 출력	PASS
	비밀번호를 잘 입력한 경우 데이터베이스의 값과 일치하는가	성공 출력	PASS
	비밀번호를 틀리게 입력하는 경우 데이터베이스의 값과 일치하는가	실패 출력	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL

데이터베이스에 등록되지 않은 아이디를 입력할 경우	일치하지 않는 비밀번호를 입력할 경우
<p>localhost:3000의 메시지</p> <p>아이디와 비밀번호를 확인하세요</p> <p>확인</p>	<p>localhost:3000의 메시지</p> <p>아이디와 비밀번호를 확인하세요</p> <p>확인</p>
로그인 하지 않고 메인 화면으로 접속할 경우	
<p>localhost:3000의 메시지</p> <p>로그인하세요</p> <p>확인</p>	

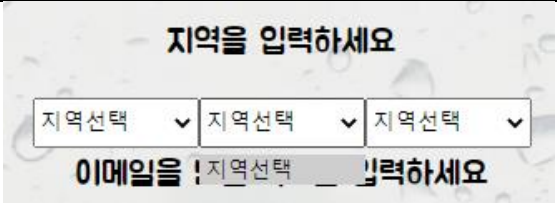
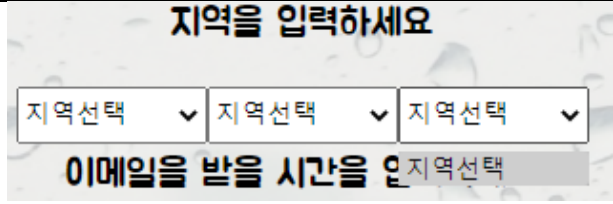
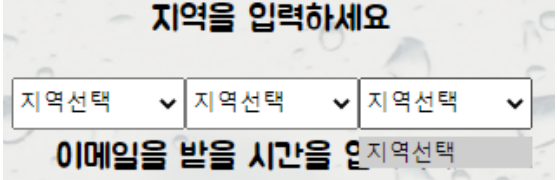
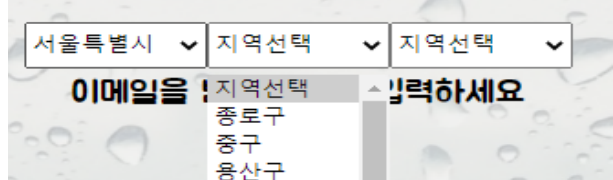
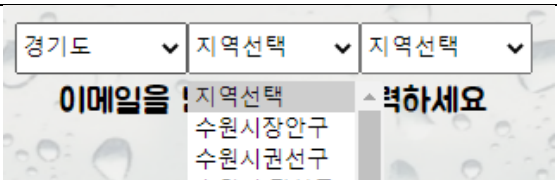
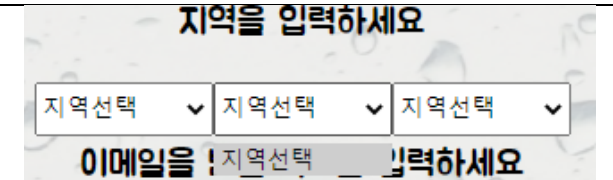
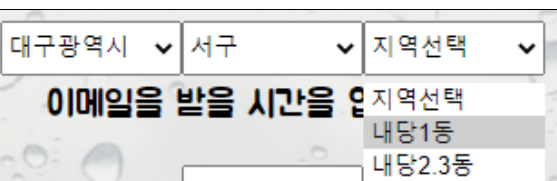
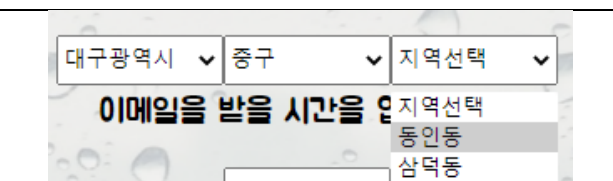
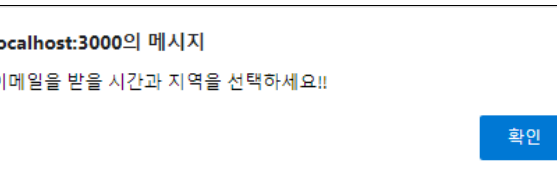
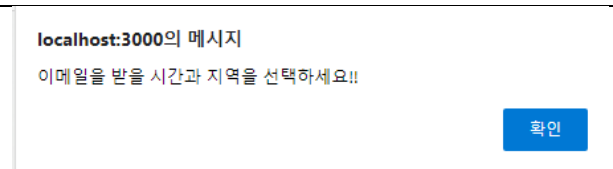
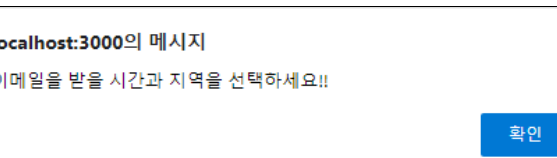
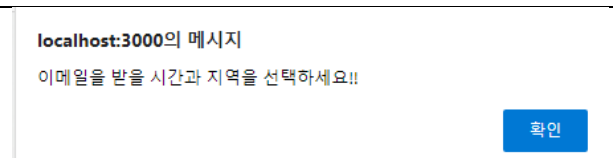
데이터베이스에 저장된 값과 일치하는 비밀번호를 입력한 경우
<p>데이터베이스에 저장된 비밀번호</p> <p>f6b72d5e50ac1b4e1a8122a61ecd33b78a0077ad331cb2c2d16760b225f2c591e1a483be2801c033d971ff9babaed67982807c80e87e697c54e573d:1bb89353</p> <p>사용자가 입력한 비밀번호</p> <p>f6b72d5e50ac1b4e1a8122a61ecd33b78a0077ad331cb2c2d16760b225f2c591e1a483be2801c033d971ff9babaed67982807c80e87e697c54e573d:1bb89353</p> <p>비밀번호가 일치합니다.</p>
데이터베이스에 저장된 값과 일치하지 않는 비밀번호를 입력한 경우
<p>데이터베이스에 저장된 비밀번호</p> <p>f6b72d5e50ac1b4e1a8122a61ecd33b78a0077ad331cb2c2d16760b225f2c591e1a483be2801c033d971ff9babaed67982807c80e87e697c54e573d:1bb89353</p> <p>사용자가 입력한 비밀번호</p> <p>50956ecbbc7cc93c6d02166304d395e9c0acedfba299df9537cc1167314a83e6b2a3b58face7c64329d3f2e63859dc2687bb1efcaa7d0c4cf21a0a9!aca80417</p> <p>비밀번호가 일치하지 않습니다.</p>

### 3) 메일 신청 기능 테스트 케이스

기능	테스트 조건	예상 결과	결과
리스트 1 지역선택	리스트 1 에서 지역을 선택할 경우	선택 값에 해당하는 지역 2 리스트에 추가	PASS
	리스트 1 에서 "지역선택"을 선택하는 경우	동작 없음	PASS
	리스트 1 에서 지역 선택 이후 리스트 1 의 지역을 다시 선택하는 경우	선택 값에 해당하는 지역 2 로 리스트 2 초기화	PASS
	리스트 2 까지 지역 선택 이후 리스트 1 을 다시 선택하는 경우	선택 값에 해당하는 지역 2 로 리스트 2 초기화, 리스트 3 초기화	PASS
	리스트 3 까지 지역 선택 이후 리스트 1 을 다시 선택하는 경우	선택 값에 해당하는 지역 2 로 리스트 2 초기화, 리스트 3 초기화	PASS
	리스트 1 까지 지역 선택 이후 리스트 1 의 "지역선택"을 선택할 경우	리스트 2, 리스트 3 초기화	PASS

	리스트 2 까지 지역 선택 이후 리스트 1 의 "지역선택"을 선택할 경우	리스트 2, 리스트 3 초기화	PASS
	리스트 3 까지 지역 선택 이후 리스트 1 의 "지역선택"을 선택할 경우	리스트 2, 리스트 3 초기화지역 선택 완료 초기화	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
리스트 2 지역선택	리스트 2 에서 지역을 선택할 경우	선택 값에 해당하는 지역 3 리스트에 추가	PASS
	리스트 2 에서 "지역선택"을 선택하는 경우	동작 없음	PASS
	리스트 2 에서 지역 선택 이후 리스트 2 의 지역을 다시 선택하는 경우	선택 값에 해당하는 지역 3 으로 리스트 3 초기화	PASS
	리스트 3 까지 지역 선택 이후 리스트 2 을 다시 선택하는 경우	선택 값에 해당하는 지역 3 으로 리스트 3 초기화	PASS
	리스트 2 까지 지역 선택 이후 리스트 2 의 "지역선택"을 선택할 경우	리스트 3 초기화	PASS
	리스트 3 까지 지역 선택 이후 리스트 2 의 "지역선택"을 선택할 경우	리스트 3 초기화지역 선택 완료 초기화	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
리스트 3 지역선택	리스트 3 에서 지역을 선택할 경우	지역 선택 완료,해당 지역의 기상청 좌표 저장	PASS
	리스트 3 에서 "지역선택"을 선택하는 경우	동작 없음	PASS
	리스트 3 에서 지역 선택 이후 리스트 3 의 지역을 다시 선택하는 경우	지역 선택 완료,해당 지역의 기상청 좌표 저장	PASS
	리스트 3 까지 선택 이후 리스트 3 의 "지역 선택"을 선택할 경우	지역 선택 완료 초기화	PASS
	서버와 통신이 불량인 경우	실패 출력	FAIL
시간신청	시간 리스트에서 메일 전송 요청 시각을 선택할 경우	시간 선택 완료	PASS
	시간 리스트에서 메일 전송 요청 시각을 선택한 이후 다시 선택할 경우	선택한 값으로 값 변경	PASS
	서버와 연결이 불량인 경우	정상 동작	PASS
신청하기	모든 지역과 시각을 선택한 경우	메일 신청 상태 변경이후 메일 신청 취소 화면으로 이동	PASS
	모든 지역과 시각이 선택되지 않은 경우	경고 출력	PASS
	모든 지역과 시각을 선택한 이후 리스트 1 에서 "지역선택"을 선택한 경우	경고 출력	PASS
	모든 지역과 시각을 선택한 이후 리스트 2 에서 "지역선택"을 선택한 경우	경고 출력	PASS
	모든 지역과 시각을 선택한 이후 리스트 3 에서 "지역선택"을 선택한 경우	경고 출력	PASS
	서버와 연결이 불량인 경우	실패 출력	FAIL
로그아웃	로그아웃 버튼 클릭할 경우	로컬 스토리지에서 유저정보삭제, 로그인 화면으로 이동	PASS
	서버와 연결이 불량인 경우	로컬 스토리지에서 유저정보삭제,	PASS

		로그인 화면으로 이동	
--	--	-------------	--

리스트 1의 "지역선택"을 클릭할 경우 리스트 2	리스트 1의 "지역선택"을 클릭할 경우 리스트 3
	
리스트 2의 "지역선택"을 클릭할 경우 리스트 3	리스트 1의 지역을 선택할 경우
	
리스트 1의 지역을 선택한 후 다시 선택할 경우	리스트 1의 지역선택 후 "지역선택"을 다시 선택할 경우
	
리스트 2의 지역을 선택할 경우	리스트 2의 지역을 다시 선택할 경우
	
리스트 3의 선택이후 "지역선택"을 다시 선택한 경우	지역과 시간을 모두 선택하지 않은 경우
	
리스트 1의 지역만 선택한 경우	리스트 2의 지역 까지만 선택한 경우
	

#### 4. 메일신청 취소 테스트 케이스

기능	테스트 조건	예상 결과	결과
저장출력	메일 신청 취소하기 화면 접속	데이터베이스에 저장된 지역과 메일 요청 시각 출력	PASS
	서버와 연결이 불량인 경우	실패 출력	FAIL
취소하기	취소하기 버튼 클릭 시	메일 신청 상태 변경 이후 메일 신청 화면으로 이동	PASS
	서버와 연결이 불량인 경우	실패 출력	FAIL
로그아웃	로그아웃 버튼 클릭할 경우	로컬 스토리지에서 사용자정보삭제,로그인 화면으로 이동	PASS
	서버와 연결이 불량인 경우	로컬 스토리지에서 사용자정보삭제,로그인 화면으로 이동	PASS

#### 5. 메일 전송 테스트 케이스

기능	테스트 조건	예상 결과	결과
타이머	매시 정각마다 시간 로그 출력	정상 로그 출력	PASS
	매시 정각마다 유저 검색하는 함수 호출	정상 호출	PASS
	서버와 연결이 불량인 경우	실패 로그 출력	PASS
메일을 전송할 유저 검색	메일을 전송할 사용자가 없는 경우	출력 없음	PASS
	메일을 전송할 사용자가 1 명인 경우	전송할 사용자 출력	PASS
	메일을 전송할 사용자가 여러 명인 경우	전송할 사용자 출력	PASS
	서버와 연결이 불량인 경우	실패 로그 출력	PASS
기상청 api 호출	기상청 x 축 좌표 68, 기상청 y 축 좌표 100 을 파라미터로 기상청 API 호출	응답 로그 출력	PASS
	사용자가 선택한 지역의 좌표로 기상청 API 호출	응답 로그 출력	PASS
	서버와 연결이 불량인 경우	실패 로그 출력	FAIL
메일 전송	사용자의 이메일로 전송될 경우	전송 로그 출력	PASS
	서버가 연결이 불량인 경우	실패 로그 출력	PASS

메일을 전송할 유저가 존재하는 경우

```
[ RowDataPacket { nx: 68, ny: 100, user_email: '2finavi@gmail.com' } ]
```

메일을 전송할 유저가 여러 명인 경우

```
[
  RowDataPacket { nx: 68, ny: 100, user_email: 'mingtorr@gmail.com' },
  RowDataPacket { nx: 68, ny: 100, user_email: '2finavi@gmail.com' }
]
```

1 분마다 로그를 출력한다.

```
Example app listening at http://localhost:3000
06:30
06:31
06:32
06:33
06:34
```

메일을 전송할 유저가 없는 경우

```
54:58
54:59
전송할 유저 없음
55:01
55:02
```

기상청 API 로 응답 받은 날씨 데이터

```
{
  '$': { seq: '2' },
  hour: [ '3' ],
  day: [ '1' ],
  temp: [ '-2.0' ],
  tmx: [ '5.0' ],
  tmn: [ '-3.0' ],
  sky: [ '1' ],
  pty: [ '0' ],
  wfKor: [ '맑음' ],
  wfEn: [ 'Clear' ],
  pop: [ '0' ],
  r12: [ '0.0' ],
  s12: [ '0.0' ],
  ws: [ '3.1' ],
  wd: [ '0' ],
  wdKor: [ '북' ],
  wdEn: [ 'N' ],
  reh: [ '65' ],
  r06: [ '0.0' ],
  s06: [ '0.0' ]
}
```

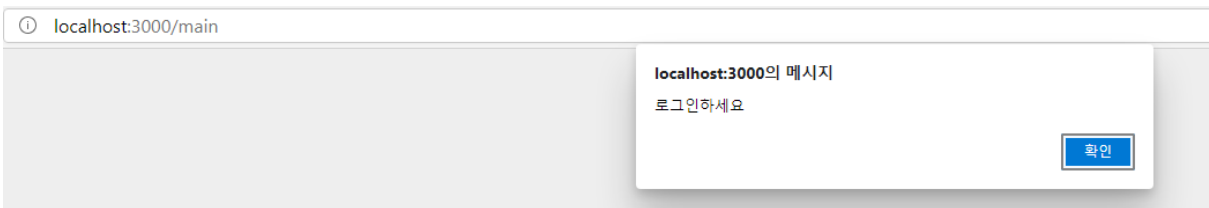
날씨 알리미가 사용하는 기상 목록

```
{
  pop : 0,
  pty : 0,
  reh : 65,
  sky : 1,
  tmn : -3.0,
  tmx : 5.0
}
```

## 8.2 제약사항 테스트 케이스

기능	제약사항	예상 결과	결과
로그인	로그인된 이후 모든 기능이 사용가능하다	경고 출력	PASS
메일신청	메일 신청을 완료한 이후 메일전송이 가능하다.	신청 이전 전송 안됨	PASS
메일전송	메일 전송 시 설정한 시간에서 5 분 이내에 전송이 완료되어야 한다.	5 분 이내에 전송완료	PASS
메일신청	시간을 설정할 때 24 시간 단위로 매시 정각으로만 설정이 가능하다.	매시 정각만 시간리스트에 출력	PASS
아이디	4~12 글자의 영문과 숫자로만 입력 가능하며 중복은 불가능하다.	조건에 맞을 경우 가입승인	PASS

- 로그인인 된 이후에 모든 기능이 사용 가능하다.



[로그인을 하지 않고 메인 화면 접근 시]

로그인을 하지 않고 url 을 수정하여 접근할 경우 "로그인 하세요" 라는 메시지를 출력하고 로그인 화면으로 이동시킨다.

- 메일 신청 버튼을 클릭한 이후 메일 전송이 가능하다.

user_email	user_sendtime	user_state	user_email	user_sendtime	user_state
mingtorr@gmail.com	8	1	mingtorr@gmail.com	8	1
20163051@changwon.ac.kr	8	0	20163051@changwon.ac.kr	8	1
2finavi@gmail.com	8	1	2finavi@gmail.com	8	1
			NULL	NULL	NULL

[그림 신청하기 버튼을 누르기 전]                      [그림 신청하기 버튼을 누른 후]

신청하기 버튼을 누르면 user\_info 테이블의 user\_state 가 1 로 바뀌어 메일을 전송받을 수 있다.

- 메일 전송 시 설정한 시간에서 5 분 이내에 전송이 완료되어야 한다.

1 차	2 차	3 차	4 차	5 차	6 차
user found! user found! 44:01 44:02 Email sent: 2 Email sent: 2 44:03	user found! user found! 45:01 45:02 45:03 Email sent: 2 45:04 Email sent: 2	user found! user found! 47:01 47:02 47:03 Email sent: 2 47:04 Email sent: 2	user found! 48:01 48:02 48:03 Email sent: 2	user found! user found! 47:01 47:02 47:03 Email sent: 2 47:04 Email sent: 2	user found! user found! 46:01 46:02 Email sent: 2 46:03 Email sent: 2
3sec	4sec	4sec	3sec	3sec	3sec

평균 3.3 초 안에 메일을 전송할 수 있다.

- 시간을 설정할 때 24 시간 단위로 매시 정각으로만 설정이 가능하다.

메인 화면의 시간 리스트	데이터 베이스에 저장된 유저 시각
---------------	--------------------



0:00 0:00 1:00 2:00 3:00 4:00 5:00		user_email	user_sendtime
		3a3e2f2b...	mingtorr@gmail.com
		1a483be2...	20163051@changwon.ac.kr
		186cded2...	2finavi@gmail.com

시간의 시 단위만 잘라서 데이터베이스에 저장한다.

- 아이디는 4~12 글자의 영문과 숫자만 입력 가능하며 중복은 불가능하다.

4 글자 이하 입력 시	특수문자 입력 시
	
12 글자 이상 입력 시	이미 사용 중인 아이디
	

### 8.3 데이터베이스 연결 테스트

- 회원가입 데이터베이스 통신

함수명	통신방식	테스트 조건	예상결과	결과
/CheckId		사용자가 입력한 아이디와 동일한 아이디 검색		
/CheckId	post	동일한 아이디가 존재하는 경우	true	pass
/CheckId	post	동일한 아이디가 존재하지 않는 경우	true	pass

동일한 아이디가 존재하는 경우	동일한 아이디가 존재하지 않는 경우
<pre>{ check_Id: 'admin' } res: false</pre>	<pre>{ check_Id: 'admin222' } res: true</pre>

- 로그인 데이터베이스 통신

함수명	통신방식	테스트 조건	예상결과	결과
/login		로그인 함수		
/login	post	입력한 정보가 데이터베이스와 일치할 경우	user_key	pass
/login	post	입력한 정보가 데이터베이스와 일치하지 않을 경우	false	pass

입력한 정보가 데이터베이스와 일치할 경우	입력한 정보가 데이터베이스와 일치하지 않을 경우
<pre>{ id: 'admin1', passwd: 'qq' } res: false</pre>	<pre>{ id: 'admin1', passwd: '1234' } { userkey: '5' }</pre>

- 메인 화면 데이터베이스 통신

함수명	설명
/getCity	리스트 1 의 지역 선택할 경우 해당하는 지역 2 를 반환한다.
/getCity2	리스트 2 의 지역 선택할 경우 해당하는 지역 3 를 반환한다.
/getCity3	리스트 3 의 지역 선택할 경우 해당하는 기상청 좌표를 반환한다.
/storeData	사용자가 입력한 지역과 시간을 저장하고 결과를 반환한다.
/cancelstate	사용자의 메일신청 상태를 취소하고 결과를 반환한다.
/getuserState	사용자의 알림 메일 신청 정보를 반환한다.

함수명	통신방식	입력	출력	결과
/getCity	post	첫번째 지역	첫번째 지역의 시단위의 지역	pass
/getCity2	post	첫번째 지역, 두번째 지역	두번째 지역의 동, 읍, 리 지역	pass
/getCity3	post	첫번째 지역, 두번째 지역, 세번째 지역	기상청 좌표	pass
/storeData	post	요청시각, 지역 1, 지역 2, 지역 3, 기상청 x 좌표, 기상청 y 좌표, 사용자 번호	true	pass
/cancelstate	post	사용자 번호	true	pass
/getuserState	post	사용자 번호	사용자가 선택한 지역, 메일 요청한 시간	pass

부산광역시를 선택했을 경우 /getCity	부산광역시 중구를 선택했을 경우 /getCity2
<pre>부산광역시 [   RowDataPacket { second: '중구' },   RowDataPacket { second: '서구' },   RowDataPacket { second: '동구' },   RowDataPacket { second: '영도구' },   RowDataPacket { second: '부산진구' },   RowDataPacket { second: '동래구' },   RowDataPacket { second: '남구' },   RowDataPacket { second: '북구' },   RowDataPacket { second: '해운대구' },   RowDataPacket { second: '사하구' },   RowDataPacket { second: '금정구' },   RowDataPacket { second: '강서구' },   RowDataPacket { second: '연제구' },   RowDataPacket { second: '수영구' },   RowDataPacket { second: '사상구' },   RowDataPacket { second: '기장군' } ]</pre>	<pre>{ city2: '중구', city1: '부산광역시' } [   RowDataPacket { third: '중앙동' },   RowDataPacket { third: '동광동' },   RowDataPacket { third: '대청동' },   RowDataPacket { third: '보수동' },   RowDataPacket { third: '부평동' },   RowDataPacket { third: '광복동' },   RowDataPacket { third: '남포동' },   RowDataPacket { third: '영주제1동' },   RowDataPacket { third: '영주제2동' } ]</pre>
부산 광역시 중구 중앙동을 선택했을 경우 /getCity3	
<pre>{ city3: '중앙동', city2: '중구', city1: '부산광역시' } [ RowDataPacket { nx: '97', ny: '74' } ]</pre>	

부산 광역시 중구 중앙동의 지역을 오전 6 시에 신청할 경우 /storeData	
<pre>[ '6', '부산광역시', '중구', '중앙동', '97', '74', '5' ] true</pre>	
부산 광역시 중구 중앙동의 지역을 오전 6 시에 신청할 경우 데이터베이스에 저장된 위치 정보	

	user_key	state	city	address	nx	ny	location_key	
▶	5	부산광역시	중구	중앙동	97	74	3	

부산 광역시 중구 중앙동의 지역을 오전 6 시에 신청할 경우 데이터베이스에 저장된 사용자 정보

	user_key	user_id	user_salt	user_passwd	user_email	user_sendtime	user_state
▶	5	admin1	152352954430	f6b72d5e50ac1b4e1a8122a61ecd33b78a0077ad331cb2c2d16760b225f2c591e1a483be2...	20163051@changwon.ac.kr	6	1

신청취소를 클릭할 경우 /cancelstate							
[ '5' ] res: true							
신청취소를 클릭할 경우 데이터베이스에 저장된 사용자 정보							
	user_key	user_id	user_salt	user_passwd	user_email	user_sendtime	user_state
▶	5	admin1	152352954430	f6b72d5e50ac1b4e1a8122a61ecd33b78a0077ad331cb2c2d16760b225f2c591e1a483be2...	20163051@changwon.ac.kr	6	0

부산 광역시 중구 중앙동의 지역을 오전 6 시에 신청할 경우 /getuserState	
<pre> { userkey: '5' } [   RowDataPacket {     state: '부산광역시',     city: '중구',     address: '중앙동',     user_sendtime: 6   } ] </pre>	