

# 1인 가구를 위한 안전지수 및 안전지도 서비스



## 프로젝트 수행 결과서 요약본

Version	0.1
Modify	2023.09.03
Project	1인 가구를 위한 안전지수 및 안전지도 서비스
Author	최원

## Revision History

[illegible]

# Contents

## I. 프로젝트 개요

1. 기획 의도.....	5p
2. 개발 목표.....	7p
3. 기대 효과.....	8p

## II. 기술 및 경험

1. 프로젝트 기술 스택 .....	10p
2. 교육과정 기술 경험 .....	11p

## III. 프로젝트 설계

1. 개발 일정 .....	13p
2. 요구사항 및 화면 설계 .....	15p
3. UML .....	18p
4. ER Diagram .....	22p

## IV. 프로젝트 구현

1. 주요 기능 소스 코드 .....	24p
2. Software Architecture .....	31p
3. 프로젝트 시연 .....	32p
4. 향후 계획 및 프로젝트 소감 .....	33p

# I. 프로젝트 개요

1. 기획 의도
2. 개발 목표
3. 기대 효과

# 기획 의도

## 이번엔 등산로 범죄, 무너지는 안전 대한민국



### 일상 공간이 위험지대화

2023년 08월 17일 신림동 야산 등산로에서 흉악범죄가 발생했다.

범인은 손에 너클을 끼고 CCTV가 없는 장소를 골라 범행을 저질렀다.

이달 초에는 퇴근길 백화점에서 20대 남성이 무차별적으로 흥기를 휘둘렀고,

지난달에는 30대 남성이 전철역 부근을 지나던 행인들을 공격했다.

한가로운 동네 뒷산, 퇴근길 백화점, 전철역 출구, 인도 등 일상의 모든 곳이 위험지대가 될 수 있다는 공포가 확산되고 있다.

1인 가구 지역 안전성 분석 결과

- 1인 가구 80%가 범죄 위험에 불안
- 비아파트에 사는 1인 가구의 범죄 발생은 약 3배 높게 나타남

1인 가구 범죄피해 경험

[단위: 건, %, 명]

구분		폭력범죄				재산범죄			
		발생 건수	피해율	발생 지수	피해자율	발생 건수	피해율	발생 지수	피해자율
전체		47	1.5	24	0.8	43	1.4	38	1.2
성별	남성	13	0.9	9	0.6	19	1.3	19	1.3
	여성	34	2.1	14	0.9	23	1.4	20	1.2
세대별	청년	19	1.3	13	0.9	19	1.3	11	0.7
	중장년	15	1.5	8	0.8	16	1.6	13	1.3
	노년	14	2.5	4	0.7	16	2.8	15	2.6
주택 유형별	아파트	9	1.0	3	0.9	6	0.7	5	0.6
	비아파트	38	1.7	20	0.9	37	1.7	34	1.5

# 개발 목표



1인가구를위한안전지수및안전지도서비스

## 1인가구를 위한 주거안전성 강화

서울시 1인 가구를 위한 안전한 주거 환경을 선택할 수 있도록 자치구별 안전지수를 산출 후 시각화하고, 주변 치안시설 및 주요 기관, 범죄 및 사고 데이터를 안전지도로 제공하는 서비스 구현을 목표로 함

# 기대 효과

## OOGG 서비스의 주요 기대 효과

### 거주 희망지역 안전정보 파악

- 안전지수 서비스를 통해 거주 희망지역의 안전지수 및 안전 시설물 파악



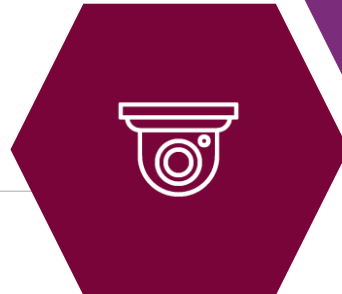
### 범죄 및 사고 발생지역 파악

- 안전지도 서비스를 통해 지역별 범죄 발생 건수 확인
- 주요 위험 요소에 대비하여 안전한 거주 선택 가능



### 안전한 1인가구 생활 환경 조성

- 치안시설 및 안전기관 정보를 종합적으로 파악하여 안심하고 생활할 수 있는 환경 조성





## II. 기술 및 경험

1. 프로젝트 기술 스택
2. 교육과정 기술 경험

# 기술 스택

## 언어



HTML5



CSS3



JavaScript



Java

1.8.0



JSP



R

## 백엔드



Servlet & JSP

4.0



Spring Framework

4.3.9



Spring Boot

4.7.0

## 데이터베이스



Oracle DataBase

## 서버



Tomcat

9.0

## 개발 환경



Visual Studio Code



Eclipse



Oracle SQL Developer



R Studio

# 기술 경험

<b>Front-end</b>	 <ul style="list-style-type: none"> <li>+ HTML5</li> <li>+ CSS3</li> <li>+ Clone Project</li> </ul>	 <ul style="list-style-type: none"> <li>+ ECMAScript 6</li> <li>+ Ajax Programming</li> </ul>	 <ul style="list-style-type: none"> <li>+ Bootstrap 구조 및 설정</li> <li>+ 반응형 layout 구현</li> <li>+ carousel 구현</li> </ul>
<b>Back-end</b>	 <ul style="list-style-type: none"> <li>+ 자바 기본 문법</li> <li>+ 자바 클래스</li> <li>+ 자바 인터페이스</li> <li>+ Mini Project</li> </ul>	 <ul style="list-style-type: none"> <li>+ DB 설치 및 환경설정</li> <li>+ 테이블 제약 조건</li> <li>+ SQL 작성 및 실습</li> <li>+ 데이터 모델링</li> </ul>	 <ul style="list-style-type: none"> <li>+ 웹 개발 환경 구축</li> <li>+ JSP Programming</li> <li>+ Servlet Programming</li> <li>+ MVC Pattern</li> </ul>
	 <ul style="list-style-type: none"> <li>+ Spring Framework 구조 이해</li> <li>+ DI, AOP Programming</li> <li>+ JDBC Programming</li> <li>+ Spring Web MVC</li> <li>+ MyBatis 연동</li> </ul>	 <ul style="list-style-type: none"> <li>+ SpringBoot 설치 및 환경설정</li> <li>+ SpringBoot DI</li> <li>+ Basic Web Programming</li> </ul>	 <ul style="list-style-type: none"> <li>+ 리눅스의 이해</li> <li>+ ubuntu 실습</li> </ul>
<b>Data Analysis</b>	 <ul style="list-style-type: none"> <li>+ R 기본 문법</li> <li>+ 데이터 전처리</li> <li>+ 회귀분석</li> </ul>	 <ul style="list-style-type: none"> <li>+ Python 기본 문법</li> <li>+ 데이터 검증 및 예측</li> <li>+ 머신러닝의 이해</li> </ul>	 <ul style="list-style-type: none"> <li>+ RESTful</li> <li>+ JSON &amp; XML</li> <li>+ 공공 DB Open API</li> </ul>
<b>etc.</b>	 <ul style="list-style-type: none"> <li>+ 협업 프로젝트 형상 관리</li> <li>+ 프로젝트, 포트폴리오 관리</li> </ul>	 <ul style="list-style-type: none"> <li>+ React 기초 및 실습</li> </ul>	 <ul style="list-style-type: none"> <li>+ MSA의 이해</li> </ul>

## Detail

### Java Mini Project

- Address programming (getter/setter)
- Address programming (Array[])
- Address programming (Inheritance)
- Car specification information programming (abstract)
- Phone specification information programming (interface)
- Exception handling basic programming
- Collection Framework programming (List)
- Address programming (File Class )

### Servlet & JSP Mini Project

- Member Management programming (Singleton Pattern)
- Address Book programming (Command Pattern)
- Board programming (Command Pattern)

### Spring Framework Web MVC Programming

- Employees MVC programming

### Spring Boot Basic Web Programming

- JDBC Template programming
- Simple Bulletin Board System programming

## III. 프로젝트 설계

1. 개발 일정
2. 요구사항 및 화면 설계
3. UML
4. ER Diagram

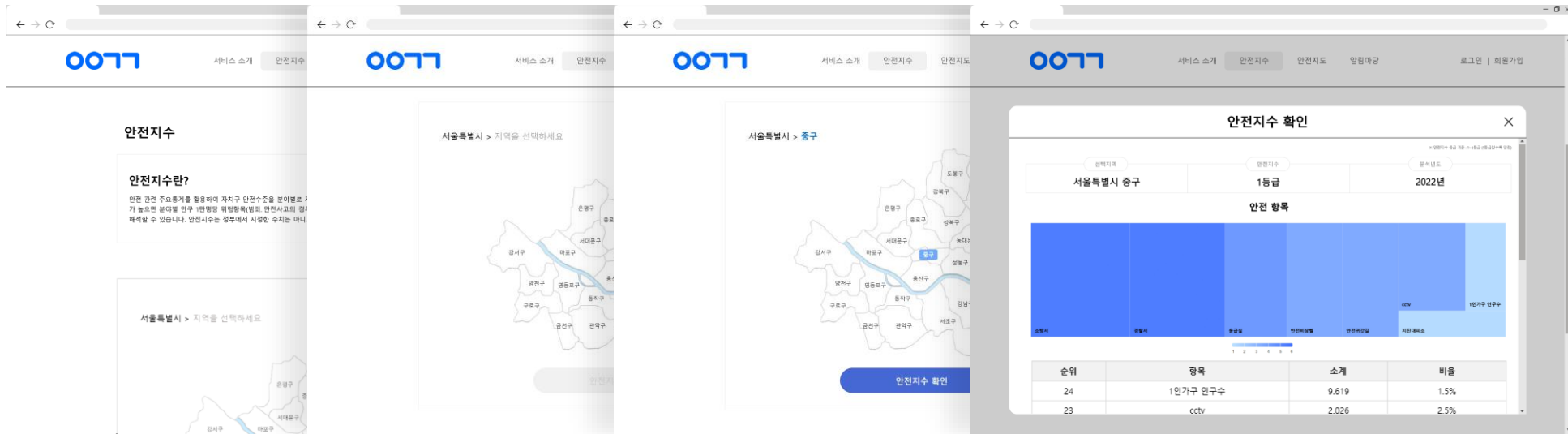
# 개발 일정

구분	작업이름	시작일	종료일	작업기간
기획	기획			18
	아이디어 구상	2023-04-27	2023-04-30	4
	프로젝트 기획서 작성	2023-05-01	2023-05-03	3
	프로젝트 기획서 중간 발표	2023-05-04	2023-05-04	1
	프로젝트 기획서 수정	2023-05-08	2023-05-12	5
	개발 스케줄표 작성	2023-05-15	2023-05-19	5
Project I	UI 개발			37
	요구사항 정의서/명세서 작성	2023-05-22	2023-05-25	4
	화면 설계서 작성	2023-05-26	2023-05-29	4
	메인 페이지 레이아웃 구현	2023-05-30	2023-06-01	3
	메인 페이지 주요 UI 구현	2023-06-01	2023-06-04	4
	안전지수 페이지 레이아웃 구현	2023-06-05	2023-06-07	3
	안전지수 페이지 주요 기능 UI 구현	2023-06-08	2023-06-11	4
	안전지도 페이지 레이아웃 구현	2023-06-12	2023-06-14	3
	안전지도 페이지 주요 기능 UI 구현	2023-06-15	2023-06-18	4
	게시판 페이지 레이아웃 및 주요 기능 UI 구현	2023-06-19	2023-06-22	4
	회원관리 페이지 레이아웃 및 주요 기능 UI 구현	2023-06-23	2023-06-26	4
	데이터 분석			11
	서울시 안전지수 (안전시설물, 주요 기관, 범죄, 사고) 데이터 수집	2023-06-27	2023-06-30	4
	서울시 안전지수 데이터 전처리 작업	2023-07-03	2023-07-05	3
	서울시 자치구별 안전지수 산출	2023-07-06	2023-07-09	4

# 개발 일정

구분	작업이름	시작일	종료일	작업기간
Project II	설계			6
	Usecase Diagram 작성	2023-07-10	2023-07-11	2
	Class Diagram 작성	2023-07-12	2023-07-13	2
	Sequence Diagram 작성	2023-07-14	2023-07-15	2
	WEB 개발			16
	안전지수 서비스 구현	2023-07-17	2023-07-21	5
	안전지도 서비스 구현	2023-07-24	2023-07-28	5
	회원관리 서비스 구현	2023-07-31	2023-08-02	3
	게시판 서비스 구현	2023-08-02	2023-08-04	3
Project III	Spring Framework 개발			13
	안전지수 서비스 구현	2023-08-09	2023-08-11	3
	안전지도 서비스 구현	2023-08-14	2023-08-16	3
	회원관리 서비스 구현	2023-08-17	2023-08-18	2
	게시판 서비스 구현	2023-08-21	2023-08-23	3
	MyBatis 변환	2023-08-24	2023-08-25	2
	프로젝트 테스트 및 디버깅	2023-08-26	2023-08-27	2
마무리	프로젝트 최종 보고서 작성	2023-08-28	2023-08-30	3
	프로젝트 배포 및 서버 업로드	2023-08-31	2023-09-01	2
	프로젝트 발표	2023-09-04	2023-09-04	1

# 요구사항

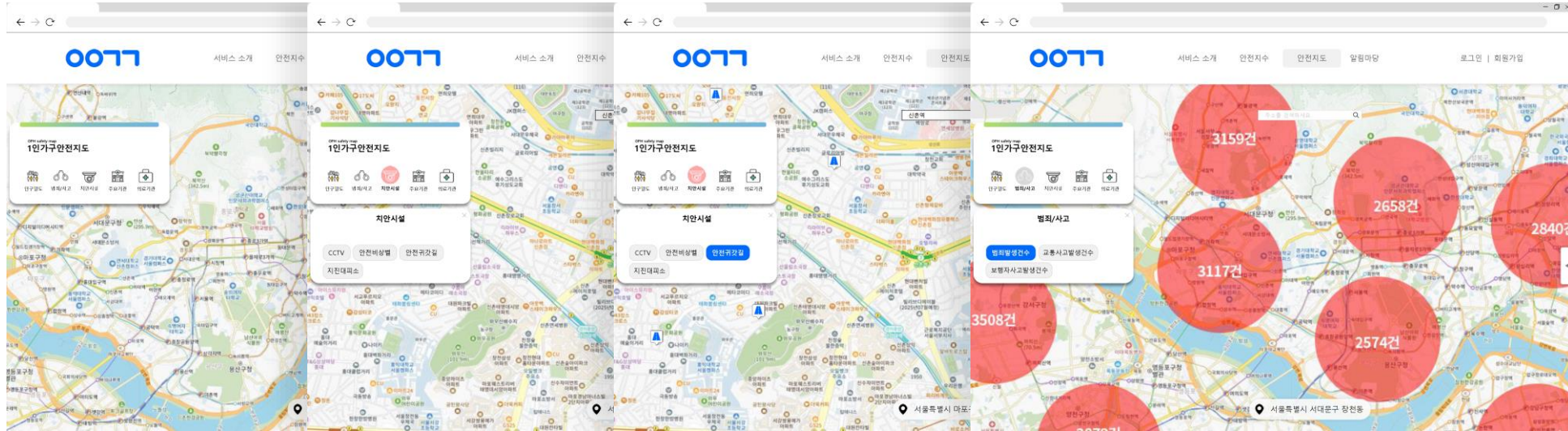


안전지수페이지 (안전지수차트데이터 기능) 스토리보드 설계 내용

## 안전지수 요구사항 정의서

- ① 서울시 지도 그림에서 25개 자치구를 구역별 글씨로 나타내야 하며, 사용자가 글씨를 클릭하여 선택이 가능하여야 한다.
- ② 초기 화면은 자치구가 선택되지 않아야 하며, 서울시 지도 하단에 안전지수 확인 버튼은 비활성화 되어야 한다.
- ③ 서울시 지도 좌측 상단에는 사용자가 선택한 자치구를 확인 가능하게 해야 되며, 초기 화면에는 “지역을 선택하세요.”라고 표시되어야 한다.
- ④ 사용자가 자치구를 선택하면 자치구 버튼은 활성화되며, 서울시 지도 좌측 상단에는 사용자가 선택한 자치구가 표시되어야 한다.
- ⑤ 사용자가 안전지수 확인 버튼을 클릭하면 안전지수와 안전지수 관련 항목을 표시하는 Modal 창이 나타나야 한다.
- ⑥ Modal 창 외부에는 투명도를 지정한 검은색 배경으로 되어야 한다.
- ⑦ Modal 창 상단에는 사용자가 선택한 자치구, 자치구의 안전지수 등급, 분석된 데이터의 기준연도 정보가 표시되어야 한다.
- ⑧ Modal 창 내용으로는 안전지수 등급을 산출하기 위해 사용된 영향 변수를 그래프와 표로 직관성 있게 나타나야 한다.
- ⑨ Modal 창 우측 상단에 닫기 버튼이 있어야 하며, 닫기 버튼을 클릭하거나 Modal 창 외부를 클릭하면 Modal 창이 종료되어야 한다.

# 요구사항



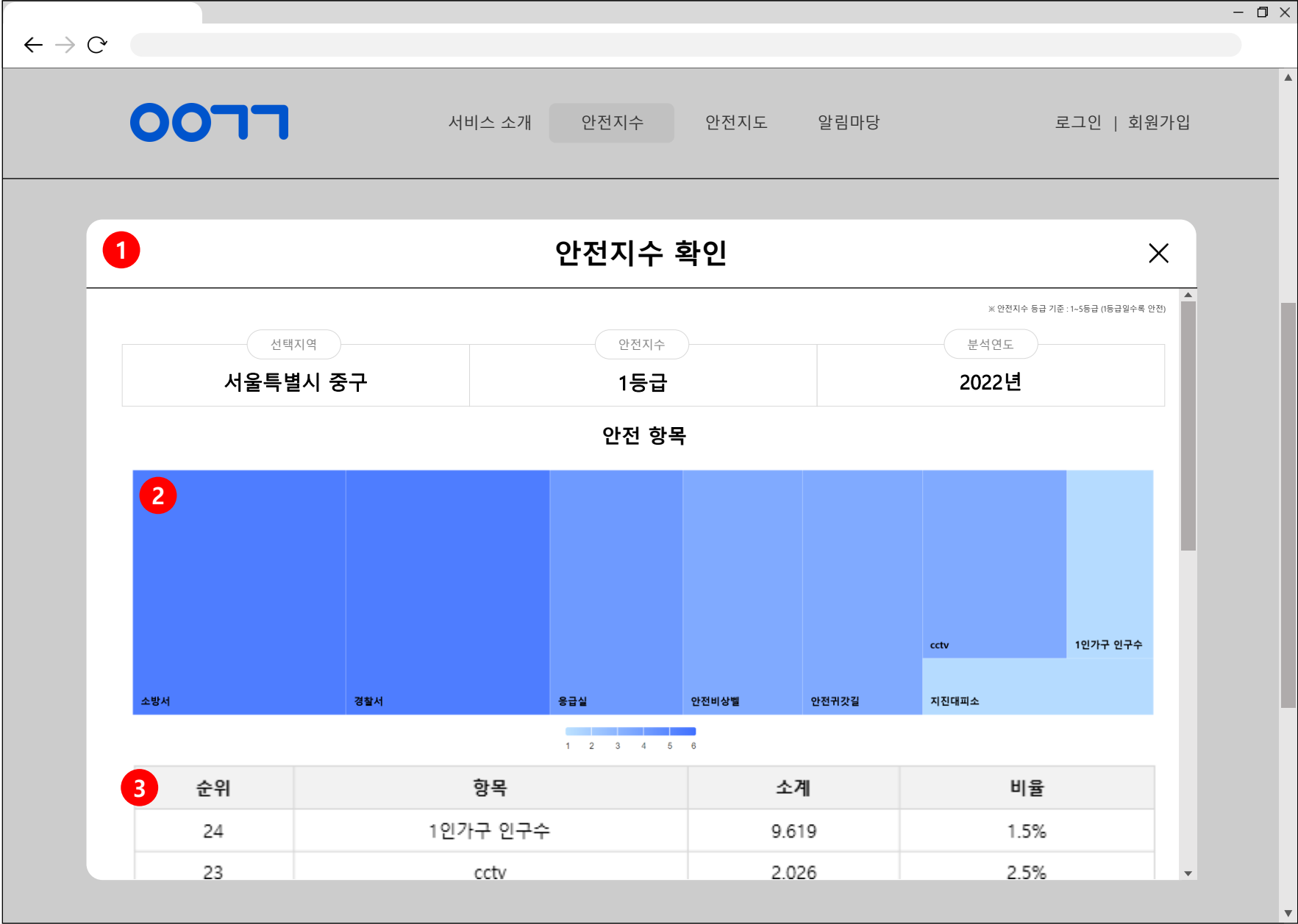
안전지도페이지 (안전항목데이터시각화기능) 스토리보드 설계 내용

## 안전지도 요구사항 정의서

- ① 안전지도 페이지 초기 지도 위치는 서울특별시 중구이어야 한다.
- ② 지도는 클릭한 상태에서 이동할 수 있어야 하며, 드래그를 통해 지도의 확대/축소가 가능해야 한다.
- ③ 지도 상단에는 주소를 검색할 수 있는 주소 검색창이 있어야 한다.
- ④ 지도 하단에는 현재 위치하고 있는 주소를 서울시 자치구 법정동 형식으로 나타내야 한다.
- ⑤ 지도 좌측에는 안전항목을 선택할 수 있는 카테고리가 있어야 한다.
- ⑥ 주소 검색창을 입력하지 않으면 “주소를 검색하세요.”라고 표시되어야 한다.
- ⑦ 주소 검색창에는 주소를 입력 후 해당 주소로 이동할 수 있는 버튼이 있어야 하며, 돋보기 아이콘으로 되어야 한다.
- ⑧ 안전항목 카테고리에는 인구밀도, 범죄/사고, 치안시설, 주요기관, 의료시설 항목으로 되어 있어야 하고 아이콘, 글자 형식으로 나열되어야 한다.
- ⑨ 사용자가 안전항목을 클릭하면 안전항목 카테고리 밑에 해당 항목의 필터가 나타나야 하며, 필터를 클릭하면 지도에 정보를 표시해야 한다.

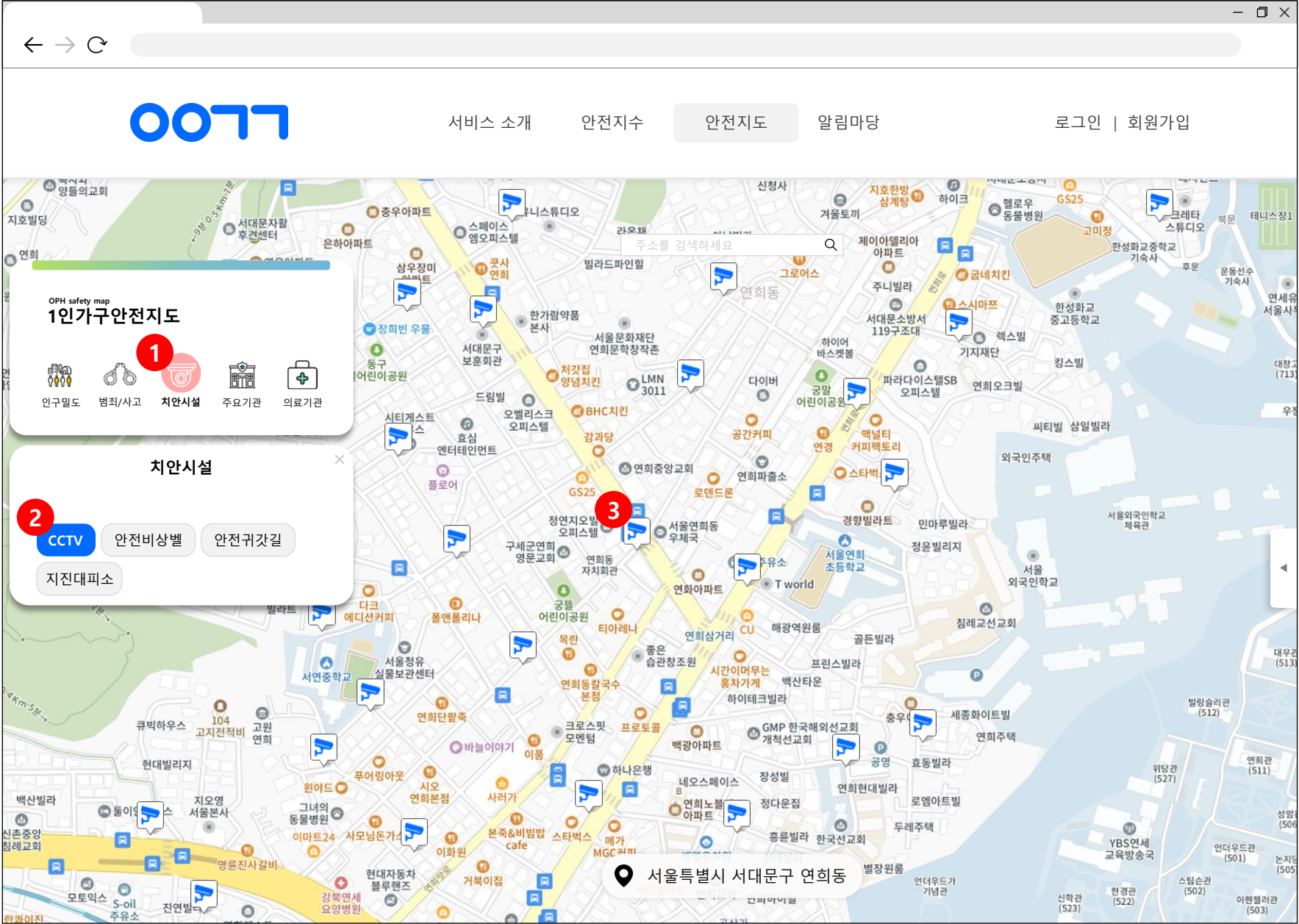


# 화면 설계



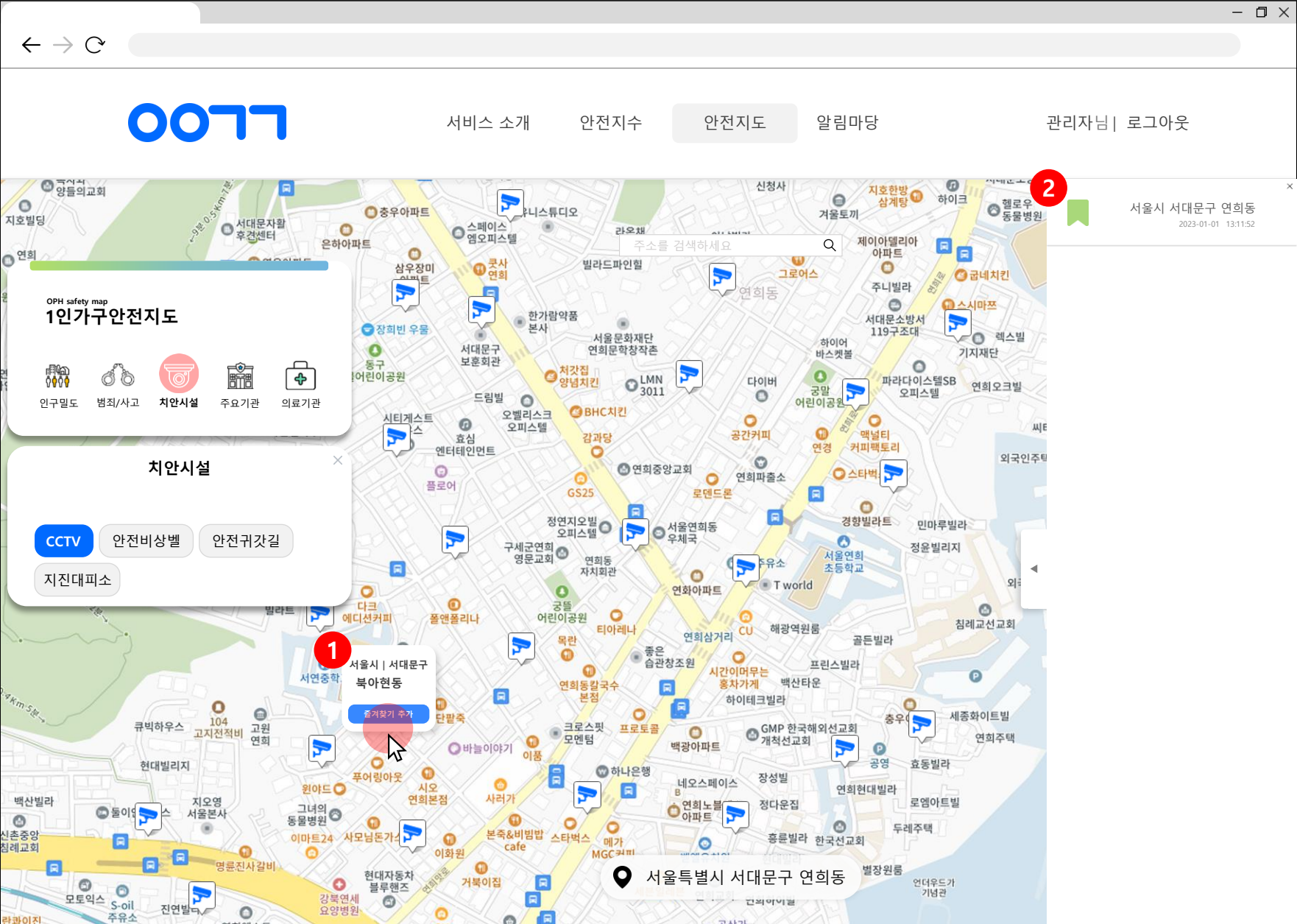
Description	
1	안전지수 확인 Modal 창 현재 선택한 지역, 안전지수 등급, 분석된 데이터 년도 표시
2	안전지수 항목 데이터를 Tree 차트로 표시
3	안전지수 항목 데이터를 테이블로 표시

# 화면 설계



Description	
1	정보를 얻을 안전지도 카테고리 선택 (ex. 치안시설)
2	정보를 얻을 항목 선택 (ex. CCTV)
3	CCTV 데이터 지도에 표시

# 화면 설계



Description	
1	지도를 한 번 클릭 시 즐겨찾기 마커 표시
2	즐거찾기 추가 클릭 시 즐겨찾기 리스트에 저장

# UML (Usecase Diagram)

## 시스템 상황 분석

메인 페이지

서비스에 대한 기능 소개 페이지

안전지수

사용자는 정보를 얻을 자치구를 선택함

사용자가 선택한 자치구의 안전지수는 그래프 형태로 나타남

안전지도

사용자는 안전지도 카테고리를 선택할 수 있음

안전지도 카테고리가 선택되면 안전지도에 항목 데이터 표시됨

사용자는 원하는 지역을 즐겨찾기로 저장 가능

저장된 즐겨찾기는 사이드바에서 리스트로 확인 가능

게시판

사용자는 공지사항, 1인 가구 소식 게시글을 확인할 수 있음

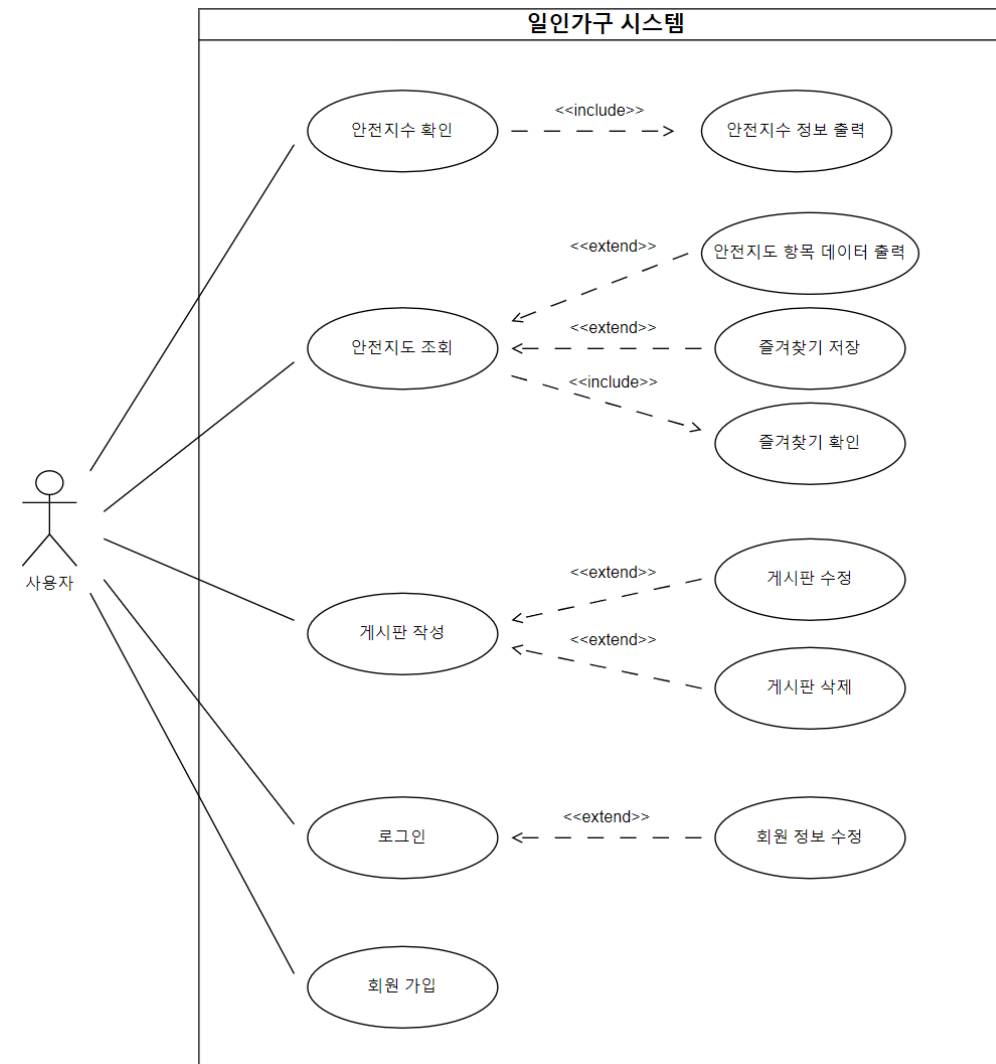
사용자는 QnA 게시글을 작성할 수 있음

사용자는 QnA 게시글을 수정 및 삭제할 수 있음

회원정보 수정

로그인했을 경우에만 회원 정보 수정 가능

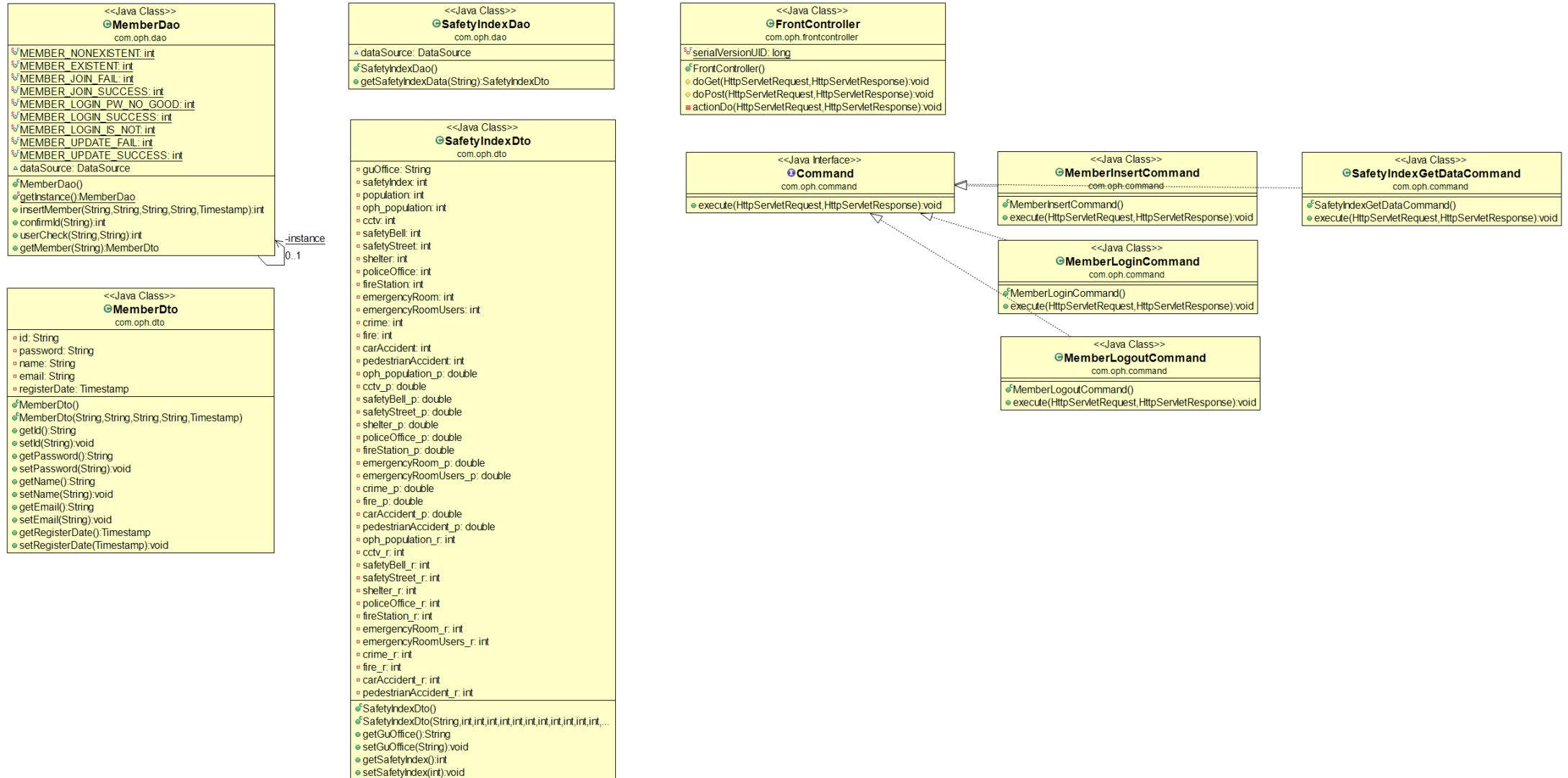
## Usecase Diagram





# UML (Class Diagram)

## Class Diagram



# ER Diagram

## ER Diagram

SCOTT.OPH_SAFETYINDEX	
GUOFFICE	VARCHAR2 (50 BYTE)
SAFETYINDEX	NUMBER
POPULATION	NUMBER
OPH_POPULATION	NUMBER
CCTV	NUMBER
SAFETYBELL	NUMBER
SAFETYSTREET	NUMBER
SHELTER	NUMBER
POLICEOFFICE	NUMBER
FIRESTATION	NUMBER
EMERGENCYROOM	NUMBER
EMERGENCYROOMUSERS	NUMBER
CRIME	NUMBER
FIRE	NUMBER
CARACCIDENT	NUMBER
PEDESTRIANACCIDENT	NUMBER
OPH_POPULATION_P	NUMBER
CCTV_P	NUMBER
SAFETYBELL_P	NUMBER
SAFETYSTREET_P	NUMBER
SHELTER_P	NUMBER
POLICEOFFICE_P	NUMBER
FIRESTATION_P	NUMBER
EMERGENCYROOM_P	NUMBER
EMERGENCYROOMUSERS_P	NUMBER
CRIME_P	NUMBER
FIRE_P	NUMBER
CARACCIDENT_P	NUMBER
PEDESTRIANACCIDENT_P	NUMBER
OPH_POPULATION_R	NUMBER
CCTV_R	NUMBER
SAFETYBELL_R	NUMBER
SAFETYSTREET_R	NUMBER
SHELTER_R	NUMBER
POLICEOFFICE_R	NUMBER
FIRESTATION_R	NUMBER
EMERGENCYROOM_R	NUMBER
EMERGENCYROOMUSERS_R	NUMBER
CRIME_R	NUMBER
FIRE_R	NUMBER
CARACCIDENT_R	NUMBER
PEDESTRIANACCIDENT_R	NUMBER

SCOTT.OPH_MEMBER	
ID	VARCHAR2 (20 BYTE)
PASSWORD	VARCHAR2 (20 BYTE)
NAME	VARCHAR2 (20 BYTE)
EMAIL	VARCHAR2 (20 BYTE)
REGISTERDATE	DATE

## IV. 프로젝트 구현

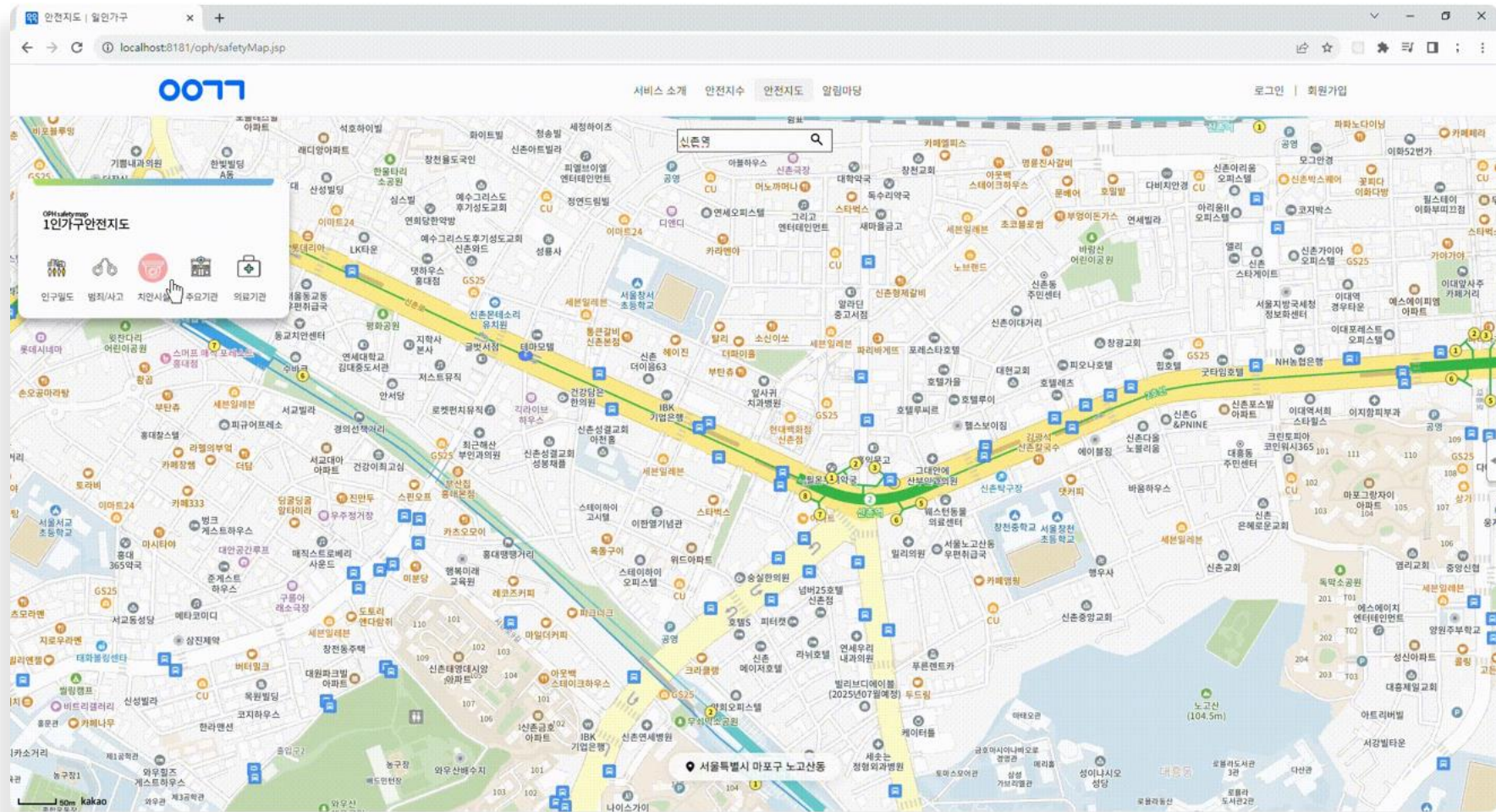
1. 주요 기능 및 소스 코드
2. Software Architecture
3. 프로젝트 시연
4. 프로젝트 수행 소감 및 향후 계획

# 주요 기능





# 주요 기능



# 소스 코드

## < SafetyIndexGetDataCommand.class >

```
package com.opf.command;

import java.io.PrintWriter;

public class SafetyIndexGetDataCommand implements Command {

    @Override
    public void execute(HttpServletRequest request, HttpServletResponse response) {

        String selectGuOffice = request.getParameter("guOffice");
        System.out.println("Select Gu Office : " +selectGuOffice);    // 확인용 로깅

        SafetyIndexDao dao = new SafetyIndexDao();
        SafetyIndexDto dto = dao.getSafetyIndexData(selectGuOffice);

        // JSON 형식으로 응답하기 위해 content type 설정
        response.setContentType("application/json");
        response.setCharacterEncoding("UTF-8");

        // 응답 데이터를 JSON 형식으로 변환하여 클라이언트에게 전송
        try (PrintWriter out = response.getWriter()) {
            Gson gson = new Gson();
            String json = gson.toJson(dto);
            out.write(json);
            System.out.println("Json Response : " + json);    // 확인용 로깅
        } catch (Exception e) {
            e.printStackTrace();
        }

    }

}
```

# 소스 코드

## <MemberInsertCommand.class>

```
package com.opf.command;

import java.sql.Timestamp;

public class MemberInsertCommand implements Command {

    @Override
    public void execute(HttpServletRequest request, HttpServletResponse response) {

        String id = request.getParameter("id");
        String password = request.getParameter("password");
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        Timestamp registerDate = new Timestamp(new Date().getTime()); // 현재 시간을 생성하여 rDate로 삽입

        MemberDao dao = MemberDao.getInstance();

        int result = dao.confirmId(id); // 아이디 중복 확인

        if (result == MemberDao.MEMBER_EXISTENT) {
            // 중복 아이디인 경우
            request.setAttribute("joinResult", MemberDao.MEMBER_JOIN_FAIL);
        } else {
            // 중복 아이디가 아닌 경우 회원 가입 수행
            result = dao.insertMember(id, password, name, email, registerDate);
            request.setAttribute("joinResult", result);
        }

    }

}
```

# 소스 코드

## < MemberLoginCommand.class >

```
package com.opf.command;

import javax.servlet.http.HttpServletRequest;

public class MemberLoginCommand implements Command {

    @Override
    public void execute(HttpServletRequest request, HttpServletResponse response) {

        String id = request.getParameter("id");
        String password = request.getParameter("password");

        MemberDao dao = MemberDao.getInstance();
        int result = dao.userCheck(id, password);

        if (result == dao.MEMBER_LOGIN_IS_NOT) { // 아이디가 존재하지 않는 경우
            request.setAttribute("loginResult", result);
        } else if (result == dao.MEMBER_LOGIN_PW_NO_GOOD) { // 비밀번호가 틀린 경우
            request.setAttribute("loginResult", result);
        } else if (result == dao.MEMBER_JOIN_SUCCESS) { // 로그인 성공
            HttpSession session = request.getSession();
            MemberDto dto = dao.getMember(id);

            session.setAttribute("id", id);
            session.setAttribute("name", dto.getName());
            session.setAttribute("loginStatus", true);
            request.setAttribute("loginResult", result);
        }
    }
}
```

# 소스 코드

## <MemberLogoutCommand.class>

```
package com.opf.command;

import javax.servlet.http.HttpServletRequest;

public class MemberLogoutCommand implements Command {

    @Override
    public void execute(HttpServletRequest request, HttpServletResponse response) {

        HttpSession session = request.getSession();
        session.invalidate(); // 세션 무효화 (로그아웃)

    }

}
```

# 소스 코드

## < MemberModifyCommand.class >

```
package com.opf.command;

import javax.servlet.http.HttpServletRequest;

public class MemberModifyCommand implements Command {

    @Override
    public void execute(HttpServletRequest request, HttpServletResponse response) {

        HttpSession session = request.getSession();

        String id = (String) session.getAttribute("id");
        String password = request.getParameter("password");
        String email = request.getParameter("email");

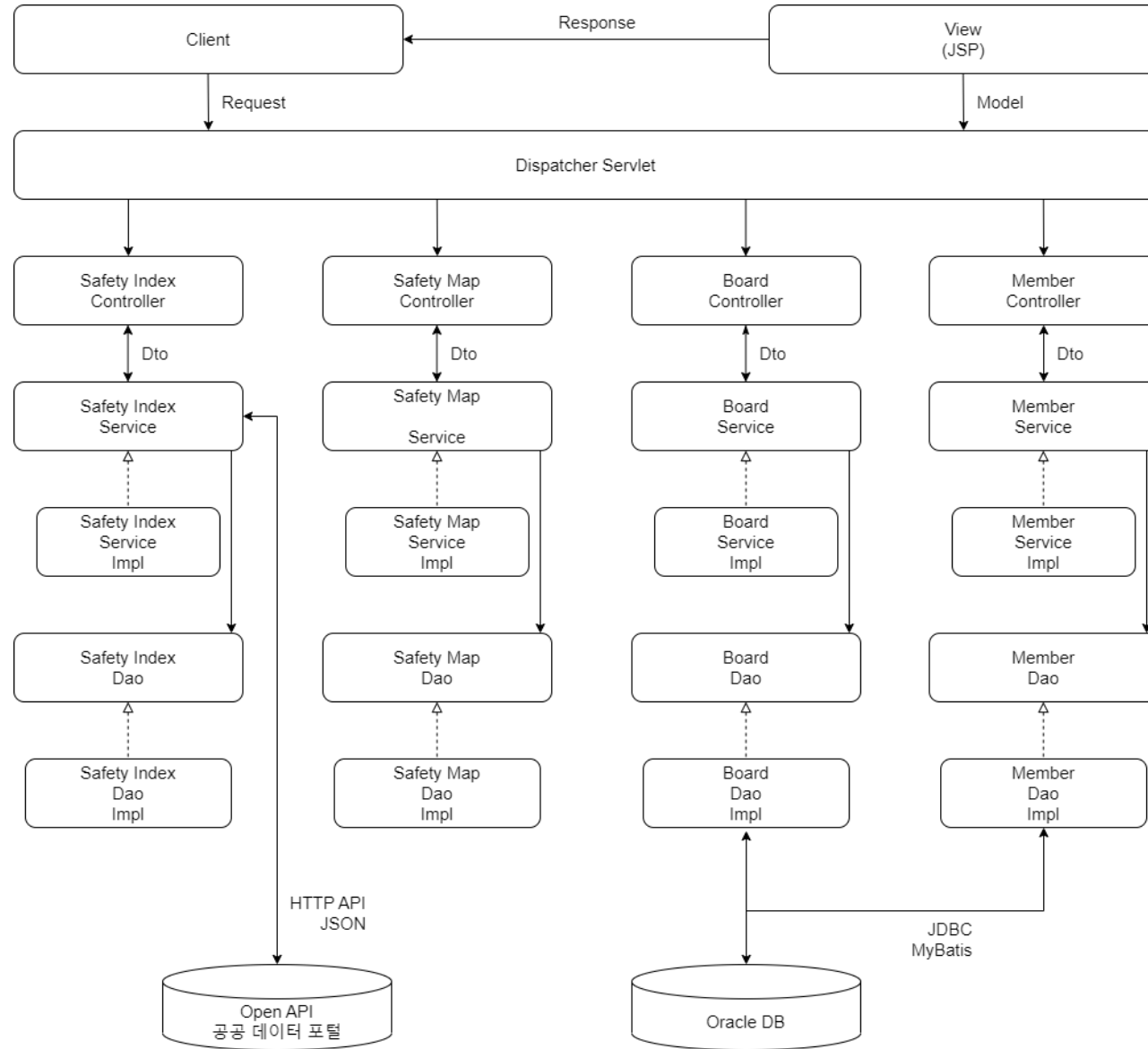
        MemberDao dao = new MemberDao();
        int result = dao.updateMember(id, password, email);

        request.setAttribute("updateResult", result);

    }

}
```

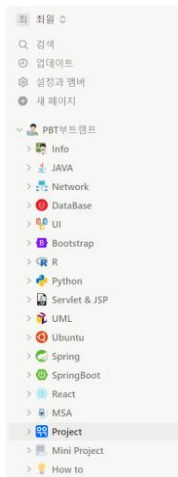
# Software Architecture



# 프로젝트 수행 소감

“

‘공부’ 이상의 ‘경험’을 얻을 수 있는 시간



## Project

### Notice

▶ 할 일 목록

### 회의록

- ▶ @2023년 4월 12일  
프로젝트 정할 때
- ▶ @2023년 4월 13일  
프로젝트 기획
- ▶ @2023년 4월 18일  
5/26 ~ 6/26 기간 내 해야 할 작업
- ▶ @2023년 4월 21일  
프로젝트 기획서 내용
- ▶ @2023년 4월 24일  
프로젝트 선정 시 중요한 점
- ▶ @2023년 4월 25일  
06/25까지 해야 할 일/4의 문제점
- ▶ @2023년 4월 27일  
프로젝트 기획 14 14 14 14

지난 6개월간의 모험을 떠나면서 느낀 점은 기술 이론 강의와 실습, 미니 프로젝트를 수행했지만 채워지지 않는 부분들이 많았던 것 같습니다. 단기간 많은 기술을 배우고 학습하면서 이런 기술들이 어떻게 연결되고 서로 상호작용되는지 혼란스럽기도 하고 이해가 되지 않는 부분이 많았습니다. 그리고 시스템적인 오류나 문제를 접할 기회도 적었습니다. 이런 부족했던 부분이 프로젝트를 진행하면서 기획부터 개발까지 단순 ‘공부’ 이상의 ‘경험’을 얻을 수 있는 시간이었습니다.



# 향후 계획



## 데이터분석서비스기능강화

평균 비교, 상관 관계 분석, 시각화 개선 등  
안전지수 서비스 기능 강화를 할 예정



## Spring 및 SpringBoot개발

Servlet & JSP 개발 완료된 서비스를  
Spring과 Spring Boot 개발을 최종 목표로 함

1인 가구를 위한

안전지수 및 안전지도 서비스

