

SSAFY 프로젝트 최종 보고서

주택 실거래가 확인 Web 페이지 설계 및 구현

[Where is My Home]



삼성 청년 아카데미 8기 서울 19반 9조

김종섭 | 이승현

목차

I 개요

- 프로젝트 개요
- 가상고객 설정

II 프로젝트 설계

- 요구사항 정의
 - 기능적 요구사항
 - 비기능적 요구사항
- 요구사항 및 기능 세부분석
 - 기능 세부분석 및 검토
 - 비기능 세부분석 및 검토
- 화면설계서 (mock up)

III 웹페이지 구현

- Todo
- R&R / Kanban
- 사용 기술스택
- 구현 중 발생 issue 및 고안점

IV 결과물 및 최종 결론

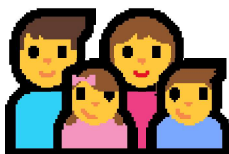
- 프로젝트 결과
 - Database ERROR 다이어그램
 - Backend 클래스 다이어그램
- 설계 ↔ 구현결과물 비교
- 프로젝트 피드백 및 결과물 분석

1. 개요

◎ 프로젝트 개요

구분	내용	비고
프로젝트명	SSAFY - Where is My Home	
프로젝트 내용	주택을 구하는 고객의 요구에 맞추어, 기간별 주택 실거래가 내역을 검색하고 부가기능을 제공할 수 있는 웹사이트를 제작	
프로젝트 일정	2022-11-17 ~ 2022-11-24	
프로젝트 목표	<p>프로젝트를 설계해보며 기능의 구현에 초점을 두는것이 아닌 구조 설계 (Architecture) 에 관한 심층적인 이해를 목표로 한다.</p> <p>발생하는 issue 들에 대하여 팀원과의 많은 고민과 토의를 거쳐 현 사용하는 기술 스택의 문제점, 한계점및 해결책을 논의한다.</p> <p>싸피 과정 중 중점적으로 배웠던 보안성을 주 고안점으로 두고, 프로젝트에 적용하여 프로젝트에 실제로 적용 시 발견될 수 있는 취약점 및 프레임워크의 보안 취약성에 대하여도 분석해본다.</p>	
담당 팀원	김종섭 - Sub Backend Engineer 이승헌 - Main Backend Engineer	

◎ 가상고객 설정



- 초. 중등학교 자녀를 둔 4인 가족이 이사를 앞두고 집을 알아보는 상황을 설정.
- 원하는 관심지역, 관심매물을 설정하고 주변 편의시설을 확인할 수 있게 요청
- 비슷한 상황의 유저들과도 게시판으로 소통하여 정보를 얻을 수 있게 함
- 개인정보 보호를 위해 기본적인 수준의 보안성을 요청

2. 프로젝트 설계

◎ 요구사항 정의

□ 기능적 요구사항

구분	내용	비고
회원관리	로그인, 로그아웃 회원 가입 회원 정보 조회, 수정 회원탈퇴	
주택 실거래가	시/도, 군/구, 동으로 아파트 목록 검색 아파트 상세 정보 조회 아파트 최근 실거래가 차트로 조회 검색한 거래 정보를 지도에 표시 관심 지역 목록 추가	
편의시설 검색	관심 지역 정보 수정 / 삭제 관심지역 주변 편의시설 검색	
게시판 이용	공지사항 : 조회 자유게시판 : 조회 / 글쓰기 / 수정 / 삭제	

□ 비기능적 요구사항

구분	내용	비고
정확성	API로 가져온 공공 데이터의 정확성 체크 각 데이터들의 유효성 보증	
사용자 편의성	사전 지식 없이도 쓰기 편해야 함	
가용성	언제나 서비스 가능	

보안성	유저 데이터는 안전하게 이용될 수 있어야 함 captcha 인증 : 회원가입 유효성 검사 : 데이터입력 Brute Force 방어 : 로그인 횟수제한 이메일 인증 : 회원가입, 비밀번호 찾기(재발급) XSS 스크립트 해킹 방지 : 게시판 글쓰기 & 회원가입	
-----	--	--

◎ 요구사항 세부분석

□ 기능 세부분석 및 검토

▶ 회원관리 기능

① 로그인은 기존의 server session 을 이용하는 것이 아닌, JWT 를 이용

※ Vue Cli 로 Backend ↔ Frontend 서버를 각각 운용하므로, 로그인시 생성된 쿠키를 front → Back 서버로 전송하기에 어려움이 있기 때문 // JWT 의 문제점 및 보안취약성은 하단에 기재

② 회원가입 시 민감한 개인정보를 입력받지 않음 (이름, 핸드폰번호 등)

※ 보안상 취약한 JWT를 사용 및 거래기능 같은 개인정보가 필요한 기능이 없기 때문

▶ 주택 실거래가

① 시, 군구, 동 데이터를 공공 데이터 포털에서 크롤링해 DB에 저장 후 데이터로 사용

② 원하는 지역의 시, 군구, 동을 입력하여 해당 내용을 kakao map api를 사용하여 지도에 출력

③ 원하는 지역 검색 후, 그지역 추가정보를 알고 싶다면 관심지역으로 저장하여 활용 가능

▶ 편의시설 검색

→ 선택했던 관심지역 주변의 상권이나 편의시설을 보여주어 고객이 원하는 지역의 정보를 쉽게 얻을 수 있게 함.

▶ 게시판 이용

① 게시판 기능을 활용하여 글을 자유롭게 쓰고 댓글을 달아 유저들 끼리 정보를 공유할 수 있게 함.

② 유저가 탈퇴하면 이용하였던 모든 글이 사라져 개인정보를 보호할 수 있도록 설계함.

□ 비기능 세부분석 및 검토

▶ 정확성

- 각각 데이터 테이블의 무결성을 위해 PK, FK 를 설정하여 사용
 - ↳ FK CASCADE 를 이용하여 데이터의 무결성 보증 및 테이블 간 구속성 강화

▶ 사용자 편의성

- 유저 편의성을 위해, 직관적인 UI를 제공
- 많은 사이트에서 채택중인 SPA(Single Page Application) 방식을 적용함

▶ 가용성

- 어디서든 사용할 수 있도록 모바일 환경 최적화

▶ ☆☆ 보안성 ☆☆

① 크로스 사이트 스크립트 (XSS) 공격 대비 로직

- 네이버 API 중 루시필터를 사용하여 크로사이트 스크립트 공격 방어

② JWT 토큰 재확인

- JWT 토큰을 사용하여 글쓰기가 허용된 사용자인지 확인
- JWT 토큰을 디코딩하여 글을 작성하는 유저와 토큰의 정보가 일치하는지 확인

③ brute force 공격 방어

- 일정시간 내 로그인 시도횟수가 5회가 초과할 시 이메일 인증을 통과해야 로그인 가능

④ 유저 Password 암호화

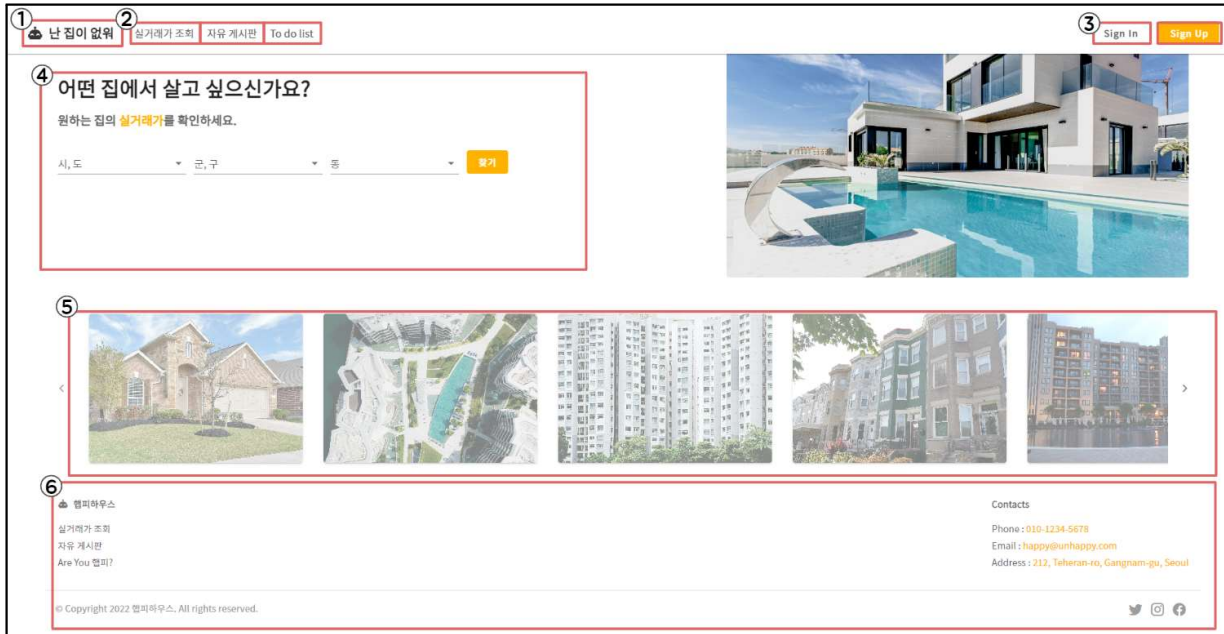
- 유저 Password를 랜덤한 salt key를 통해 암호화 하여 보안

⑤ 유저 정보 확인 재인증

- 유저정보 조회, 수정시 비밀번호 확인으로 재인증
- 회원정보 수정 시 특정문구 입력으로 재인증

◎ 화면설계 (Mock Up)

① 메인페이지



1) 홈버튼

2) 실거래가 페이지. 자유게시판. 관심지역 이동 버튼으로 구성

- 로그인 전에는 해당 버튼(기능)이 보이지 않음

3) 각각 로그인, 회원가입 버튼으로 구성

4) 시 / 군.구 / 동 으로 검색하면 해당 데이터로 검색하여 실거래가 조회 페이지로 이동

5) 슬라이드로, 주택들의 이미지를 미리보기 할 수 있도록 구성

- 디자인 구성에 따라 추후 변동 가능

6) Footer 에 기본적인 About 정보 게시

② 로그인 페이지 / 회원가입 페이지

로그인

아이디

비밀번호

로그인

카카오로 로그인

회원가입

아이디

as

최소 4 글자를 입력해주세요.

비밀번호

pwd is not valid.

비밀번호 확인

이름

a

최소 3 글자를 입력해주세요.

이메일

as

유효한 이메일 양식이 아닙니다.

회원가입

카카오 회원가입

1) 아이디 비밀번호 입력해야 로그인 할수 있게 구현

2) 아이디(이메일), 비밀번호, 닉네임, 이름, 비밀번호 확인 으로 구성

- **각각 입력값은 유효성 검사를 실시**
 - 아이디(이메일) : @ .com 이 포함
 - 비밀번호 : 대문자, 소문자, 8자이상, 특수문자 포함 & 재확인 - 두 비밀번호가 같은지
 - 이름 & 닉네임 - 개행문자 포함여부 확인 (포함안되게)

③ 게시판

자유 게시판

1

Q 제목으로 검색

클 작성하기

2

번호	제목	작성자	작성일	조회수
1	title	writer	date	hit
2	title2	writer2	date2	hit2
3	title3	writer3	date3	hit3
4	title4	writer4	date4	hit4
5	title5	writer5	date5	hit5

4

<

1

2

>

1) 제목, 사용자로 검색하여 원하는 게시판 글을 찾기

2) **글목록을 확인하며, 해당되는 글을 눌렀을 시 세부페이지로 이동**

- 세부 페이지에서는 글수정(삭제) 가능

④ 실거래가 조회 페이지

난 집이 없어

시,도 경기도 구,군 송파구 동 잠실동

서울시 송파구 잠실동

관심지역 등록

더글러스하우스
15층, 2021. 11. 4
서울특별시 강남구 삼성동

상세 보기

효성빌라101
15층, 2021. 11. 4
서울특별시 강남구 삼성동

상세 보기

삼일리브힐
15층, 2021. 11. 4

The map displays the area around Songpa Station (잠실역) in Songpa-gu, Seoul. It shows major roads like Hannam-daero (한남대로) and Jamsil-ro (잠실로), as well as landmarks such as Jamsil Sports Complex (잠실체육관), Jamsil High School (잠실고등학교), and various residential complexes. A blue circle highlights the specific area of interest near the station.

- 1) 시 / 군.구 / 동 으로 검색하여 해당 위치를 세부검색(2) 과 지도(3) 에 띄움
- 2) 세부검색 네비게이션 바에 간단한 정보 표시
- 3) 카카오 맵 api를 이용하여, (1) (2) 에서 클릭한 아파트 위치로 이동, 마커로 표시
- 4) 해당 동을 관심지역에 추가
 - 관심지역은 DB에 저장
 - 저장된 관심지역은 관심지역 관리 페이지에서 편의시설 검색에 사용

⑤ 관심지역 보기 페이지



- 1) 원하는 편의시설 및 지역정보를 드롭다운 식으로 하여 검색, 지도에 띄움
- 2) 실거래가 조회페이지에서 저장한 관심지역을 클릭하여 해당 위치로 지도이동
 - 관심지역은 해당 페이지에서 삭제 가능
 - 최대 5개로 제한
- 3) 카카오 맵 api를 이용하여, (2) 의 위치로 이동하여 (1) 정보를 마커로 표시

3. 웹페이지 구현

◎ Todo

기능	내용	비고
라우팅	<ul style="list-style-type: none"> ㉠ 라우터 가드 <ul style="list-style-type: none"> ↳ Vue api인 라우터 가드를 사용하여 허가되지 않은 사용자 (로그인되지 않은) 의 페이지간 이동을 막는다. 	
JWT 토큰	<ul style="list-style-type: none"> ㉠ JWT 기능을 사용하여 유저 권한을 확인한다. ㉠ JWT 토큰을 로그인한 유저에게 발급하여, refresh token 을 userDB 에 저장하여 토큰이 만료되면 재확인할 수 있도록 한다. ㉠ Access Token 의 지속시간을 5분, refresh token 의 지속시간을 1주일 정도로 차이를 두어 탈취 위험도를 낮춘다 ※ 근본적인 보안 해결책이 되지 못한다. 	
회원 관리	<ul style="list-style-type: none"> ㉠ 회원가입 <ul style="list-style-type: none"> ↳ 회원가입 시 유효성 검사를 실행한다. ↳ 비밀번호를 저장할 시, SHA256 암호화를 이용한다. SALT 값을 random UUID 로 만들어 DB에 저장하고 해당 SALT로 사용자가 입력한 비밀번호 암호화를 하여 암호화된 비밀번호 값을 DB에 최종적으로 저장한다. ↳ 회원가입이 완료되면 이메일 인증을 통하여 해당 이메일이 사용자의 이메일인지 확인하는 과정을 거친다. ↳ 카카오 회원가입 (OAuth) ㉠ 회원정보 수정 <ul style="list-style-type: none"> ↳ 회원정보 확인 시 비밀번호 재인증을 거친다. ↳ 회원정보에서 id 는 수정할수 없으며, 비밀번호 및 이름은 유효성검사 및 중복검사를 재진행하여 수정한다. ㉠ 회원 탈퇴 <ul style="list-style-type: none"> ↳ 회원 탈퇴 시 '네 탈퇴하겠습니다' 를 입력 후 탈퇴가능 ↳ 탈퇴시 본인이 작성했던 모든글, 관심지역이 삭제된다 	
로그인, 아웃 기능	<ul style="list-style-type: none"> ㉠ 세션대신 JWT 토큰으로 로그인기능을 구현 ㉠ 로그인 시 실사용자임을 확인하기 위해 Google Captcha 인증 사용 ㉠ 로그인 5회이상 틀릴 시 계정 임시잠금 및 메일인증으로 brute force 공격 예방 ㉠ 동일한 id 가 다른 ip 에서 접속이 되었을때, 히스토리에 사용자의 ip를 남겨 본인인증 요청 	

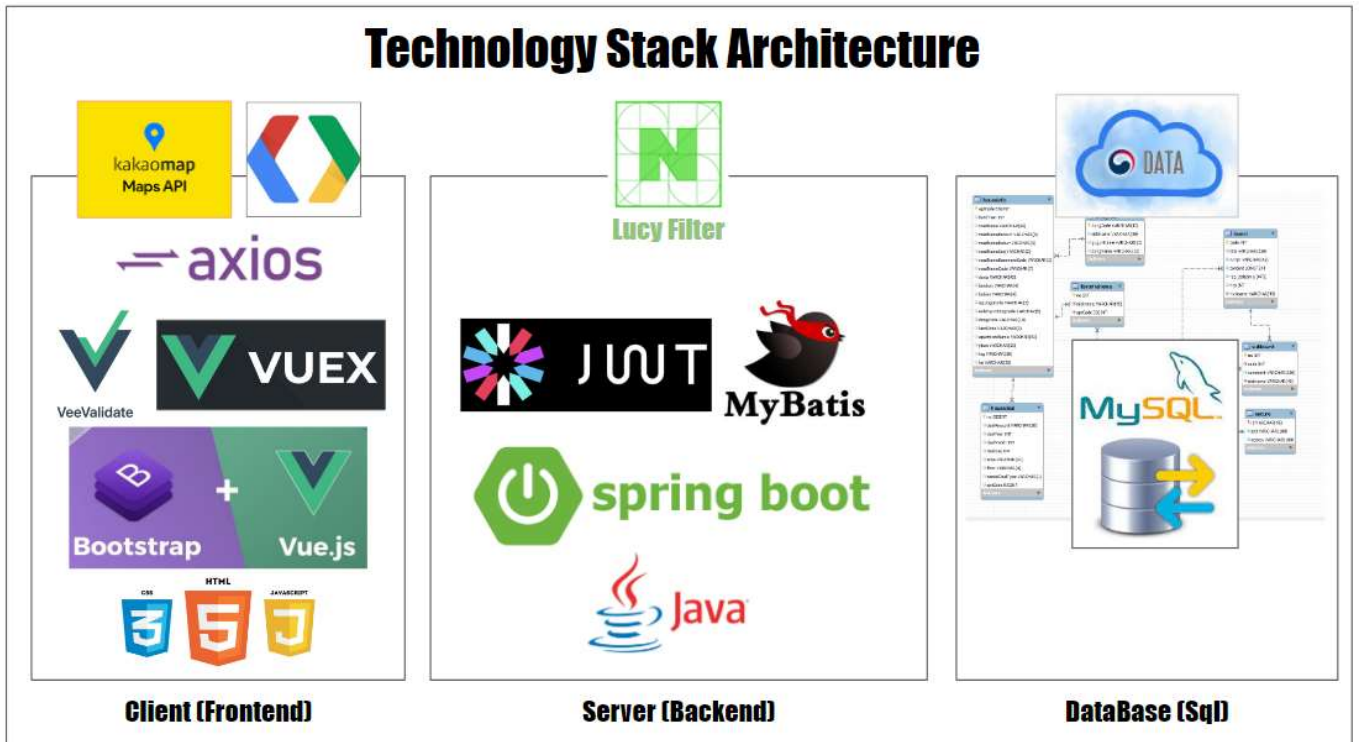
게시판 페이지	<ul style="list-style-type: none"> ㉠ 게시물 목록 <ul style="list-style-type: none"> ↳ 게시글을 1페이지에 10개씩 볼 수 있도록 페이징 ↳ 검색창에 제목으로 게시글을 검색할 수 있게 만든다 ↳ 로그인한 사용자라면 글 쓰기 버튼을 사용할 수 있다 ㉡ 글쓰기 기능 <ul style="list-style-type: none"> ↳ 로그인 한 사용자라면 글 쓰기 가능 ㉢ 글 상세보기 <ul style="list-style-type: none"> ↳ 로그인 한 사용자라면 글 상세보기 가능 ↳ 글 작성자와 로그인한 사용자가 같을 시 삭제 수정 가능 ↳ 좋아요 기능 ㉣ 댓글 기능 <ul style="list-style-type: none"> ↳ 로그인 한 사용자라면 댓글 작성 가능 ↳ 댓글 작성자와 로그인한 사용자가 같을 시 삭제 수정 가능 ↳ 댓글이 5개 이상 있을 시 더보기를 클릭해야 확인 가능 	<p>추가된 기능</p> <ul style="list-style-type: none"> ↳ 좋아요 기능 ↳ 댓글 페이징
실거래가 확인 페이지	<ul style="list-style-type: none"> ㉠ 아파트 매물 검색 <ul style="list-style-type: none"> ↳ 시, 군, 구 날짜 입력 시 해당 정보 아파트 매물 검색 ↳ 검색된 매물을 카카오맵을 통해 마킹 ↳ 검색된 매물 클릭 시 카카오맵을 통해 위치 보여줌 ↳ 검색된 매물 링크 클릭시 네이버 검색 페이지로 이동 ㉡ 관심 아파트 매물 추가 <ul style="list-style-type: none"> ↳ 매물 추가 버튼 클릭 시 관심 매물 리스트에 추가 	
관심지역 페이지	<ul style="list-style-type: none"> ㉠ 관심매물 정보 출력 <ul style="list-style-type: none"> ↳ 관심 매물 클릭 시 해당 위치로 카카오 맵 이동 ↳ 토글 가능한 버튼으로 근처 학교 위치 마커로 보여줌 ↳ 거래 내역 확인 버튼 클릭시 최근 8년간 거래내역을 그래프로 보여줌 	<p>추가된 기능</p> <ul style="list-style-type: none"> ↳ 거래내역 그래프 <p>삭제된 기능</p> <ul style="list-style-type: none"> ↳ 근처 동물병원 마킹

◎ R&R

□ R&R (업무분담)

	김종섭	이승헌
내용	<p>Front : 페이지 그리드 및 디자인 수정</p> <p>Vuex, 라우팅</p> <p>Back : User Security (SHA-256, AES-128), JWT Token, Kakao OAuth, email 인증, Captcha</p> <p>Full : 로그인, 아웃 / 회원관리</p>	<p>Front : 메인페이지 및 기본 Component 구조</p> <p>카카오, Google API 적용</p> <p>Back : Lucy Filter, CSRF Token</p> <p>DB : 전체 구조 설계, DATA Crawling</p> <p>Full : 게시판 / 실거래가확인 / 관심지역</p>

◎ 사용 기술스택



◎ 구현 중 발생 ISSUE 및 고안점

□ 툴 및 프레임워크 별 호환성 이슈

▶ ESAPI 를 사용하지 않은 이유

- ES Api는 사용자를 인증하고 확인하는 정보를 파일시스템에 저장하고 관리하기 때문에 DB 사용 불가능
- 현재 프로젝트는 DB를 통해 사용정보를 관리함. 때문에 적합하지 않음

▶ Vue.js 의 문제점

Vue

- 현재 프로젝트는 Vue.js 서버와 Spring boot 서버가 동시에 돌아가며 요청을 주고받는 식으로 구성되어있음
- 서로 다른 포트에서 돌아가기때문에 세션을 공유할 수 없어 세션을 통한 사용자 인증이 불가능
- Build 하여 Spring Project Static 파일에 넣으면 이부분을 해결 가능하나 디버깅이 힘들어짐

▶ JWT 를 사용한 이유 및 한계점

- Vue Cli 를 사용하여 View를 구현할 경우, 자체적인 서버를 통하여 view 페이지가 가동되므로 포트가 두개로 나누어지게 됨.
- 세션을 통해 사용자를 인증할 수 없기 때문에 JWT 를 사용함
- JWT는 사용자가 토큰을 보여주지만 하면 인증이 되는 식이기 때문에 보안이 취약함

▶ Spring Boot 의 장단점 및 기존 자바와 비교했을때? (수업시간에 배운내용)

- JAVA과 비교할 시 개발자의 편의성이 상당히 올라감 (annotation을 통한 설정 간소화)
- JAVA에 비해 프로젝트가 많이 무거워짐

□ 보안

▶ XSS 공격과 Vue

- 현재 프로젝트에서 게시글, 댓글 등으로 <script> 문을 넣어 작성하게 되면 XSS 공격이 일어날 위험이 있음
- 실제로 글 내용에 <script> alert(); </script> 를 넣고 동작해 보니 script 문이 실행되지 않음
- 이유를 알기 위해 build 한 script 파일을 열어 데이터의 리턴 형식을 보니 return ["" + data + ""] 식으로 리턴하고 있었다
- 즉 Vue는 기본적으로 데이터를 html 방식으로 보여주는 것이 아니라 text 형식으로 보여주기 때문에 <script> 문이 실행되지 않음

▶ Lucy 필터 적용 (response head, body 등등에 관하여)

- Lucy Filter는 Request 를 받을 때 적용하는 것과 Response를 받을 때 사용하는 것이 있다
- Request 를 받을 때 필터링 할 시 "< . >" 등의 문자가 & 이런식으로 변해서 DB에 저장되기 때문에 DB에 칼럼 이 담을 수 있는 크기를 초과해 버릴 수 있을 것 같다는 생각이 들었다
- 따라서 데이터를 response 할 때 필터링을 적용하는 방식을 사용했다

▶ CSRF 미적용 사유

- 기존에는 글쓰기 요청을 보낼 때 CSRF 토큰을 포함한 채로 글쓰기 폼 자체를 리턴해주어 인증하였다
- 현재 프로젝트는 싱글 페이지 어플리케이션이기 때문에 요청을 보낼 때 폼을 리턴하는게 아닌 url을 변경한다
- 따라서 중간과정을 하나 생략했기 때문에 CSRF 토큰을 발행하고 확인하는데에 문제가 생겨 사용하지 않음

▶ url 에서 ip 입력후 이동할 수 있는 issue

→ router guard 를 통해 로그인 하지 않은 사용자는 페이지에 접근 할 수 없게 만들었다

▶ 인증되지 않은 사용자가 글쓰기 요청을 보낸 경우

→ 비 정상적인 사용자가 글쓰기 요청을 계속 보내 문제를 일으킬 수 있다

→ 글작성시 JWT 토큰을 헤더에 포함하여 서버로 보내 JWT 토큰을 해독해 얻은 id 값과 요청을 보낸 사용자의 id값이 일치한다면 글쓰기 요청이 수행되게 만든다

→ 추가적으로 DB에 등록되어있는 사용자인지까지 확인 (id, 토큰 비교)

▶ REST API를 사용하지 않은 이유

→ REST API 는 기본적으로 서버와 서버간의 통신을 위해 만들어진 기능이다

→ 현재 프로젝트는 서버와 클라이언트의 통신으로 이루어져있기 때문에 적합하지 않다

→ 또한 현재 프로젝트는 톱캣이 내장되어 실행되는 서버라 톱캣에서 PUT, DELETE 같은 메소드등을 허용하지 않는 값이라고 막아준다 . 하지만 모종의 실수로 PUT, DELETE를 허용하게 될 시 치명적인 문제가 될 수 있다.

실험 → <https://jealous-capibara-3a7.notion.site/REST-API-6f7334d83645479091204e99425080c5>

□ API

▶ KAKAO API

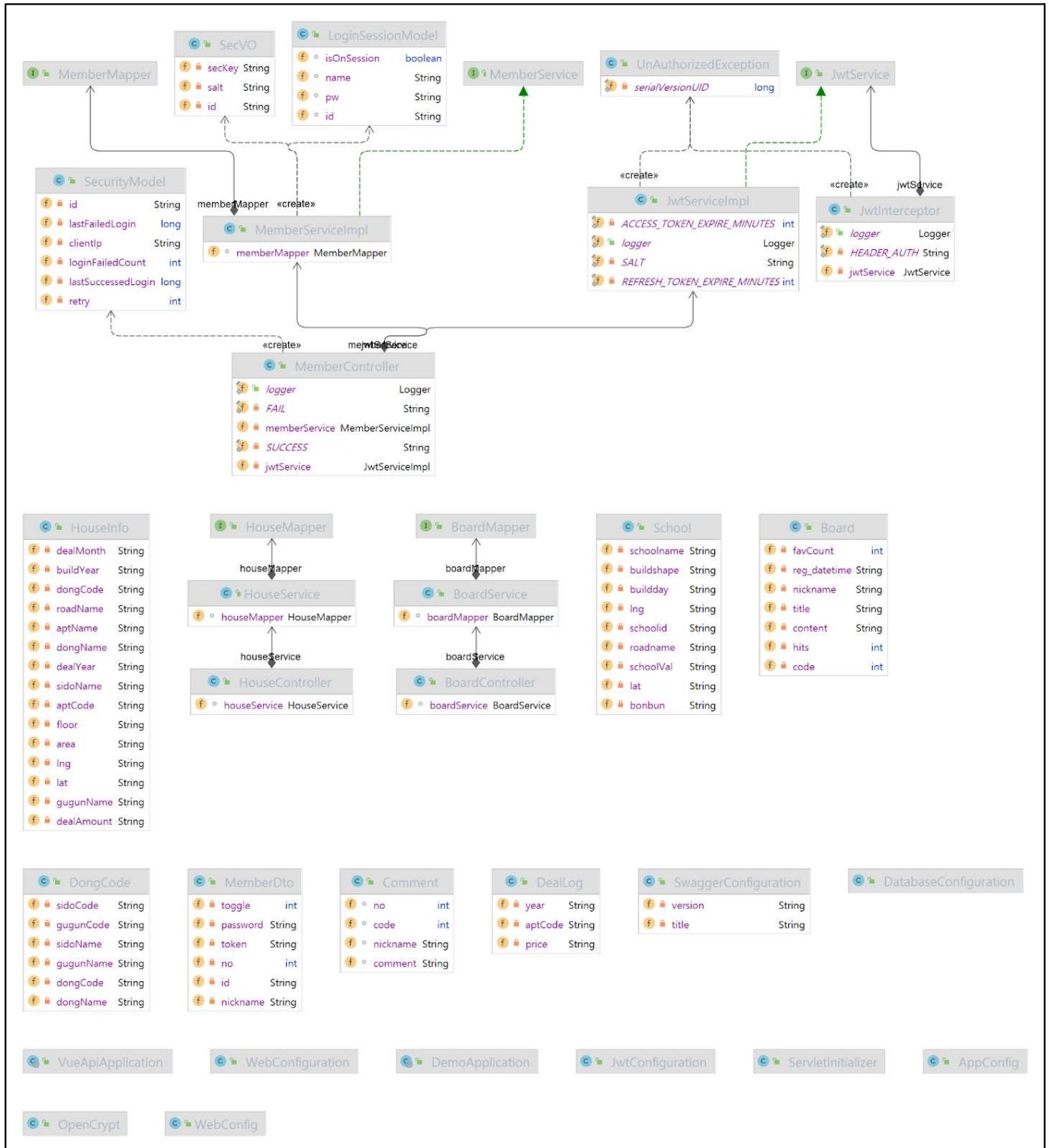
→ 카카오 맵 API 를 통해 검색 된 매물의 위치와 근처 학교의 위치를 표시하였다

▶ GOOGLE GRAPH API

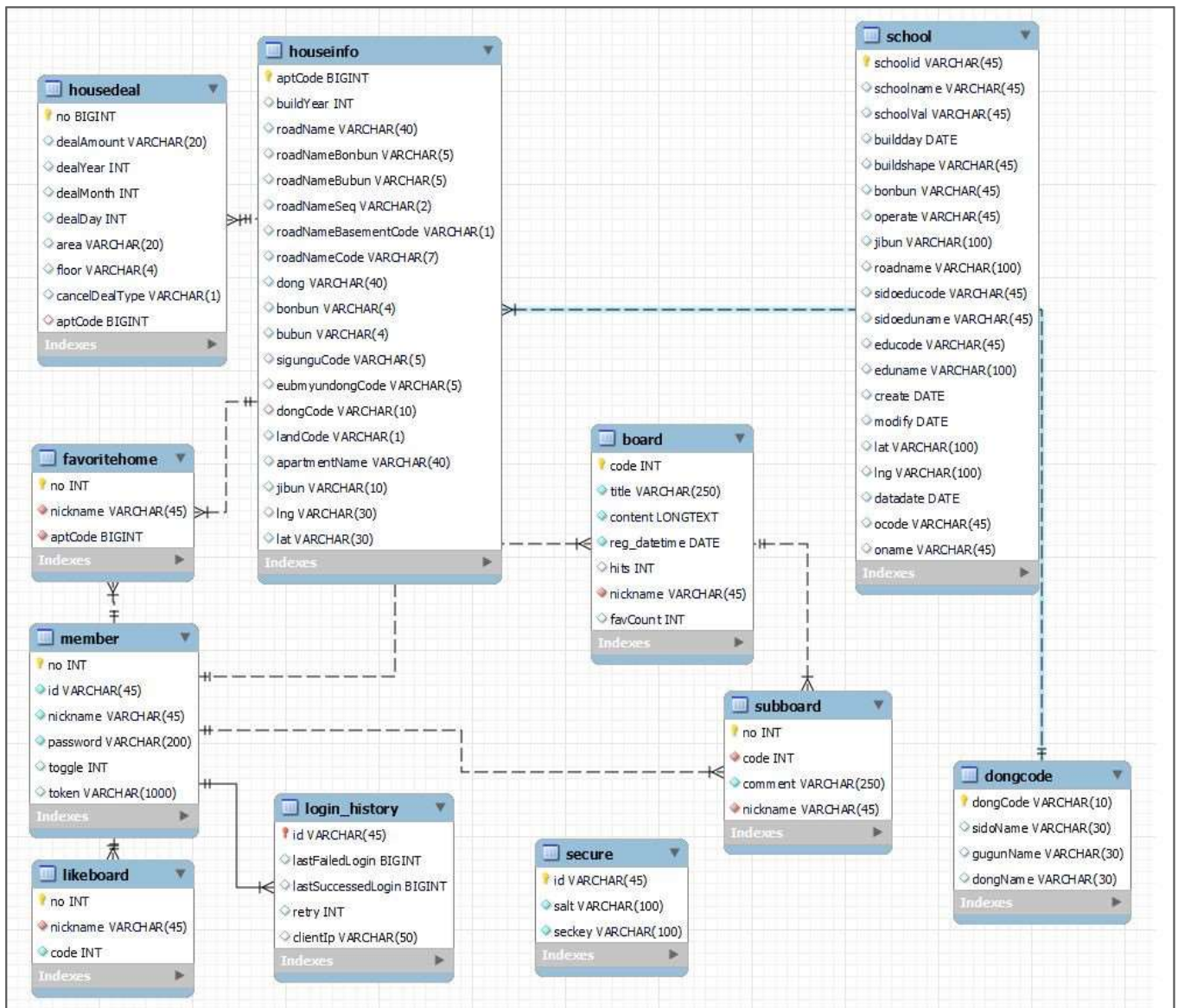
→ 구글 그래프 API 를 통해 관심 매물의 최근 8년간 거래가를 나타내었다

4. 결과물 및 최종 결론

◎ Backend class diagram



© Database ERR Diagram



◎ 설계 ~ 최종구현 결과물 비교

내용	구현	미구현	추가구현
기능	<p>[유저 관리] 로그인, 로그아웃, 회원가입, 회원정보 수정, 탈퇴</p> <p>[실거래가 검색] 시 / 군.구 / 동 으로 아파트목록 검색, 아파트 상세정보, 최근 실거래가 차트로 조회, 검색한 정보를 지도에 표시, 관심 지역 추가</p> <p>[관심지역] 관심 지역 정보 수정 / 삭제 관심지역 주변 편의시설 검색</p> <p>[게시판] 게시글 목록 불러오기 / 글쓰기 / 글 상세보기 / 수정 및 삭제</p>	<p>[유저관리]</p> <p>Email 인증 Kakao OAuth Google Recaptcha</p> <p>[관심지역]</p> <p>근처 동물병원 마킹</p>	<p>[관심지역]</p> <p>거래내역 그래프</p> <p>[게시판]</p> <p>댓글 기능 + 페이징 좋아요 기능</p>
비기능 (주로 보안)	<p>JWT - 토큰 발급 및 사용자 확인 / 라우터 가드</p> <p>사용자 password 암호화 / 유효성 검사</p> <p>brute force 방어 / login history 체크 - ip확인</p>		

◎ 프로젝트 피드백 및 결과물 분석

□ 미구현 사유

▶ Email 인증

JavaMailSender 기능을 이용하여 초기에 이메일 인증을 계획하였다. 하지만 단순 인증키를 받아 사이트에서 인증하는 것이 아닌, URL 링크를 전달받아 해당 URL 을 클릭 시 자동인증이 되는 방법을 찾아보았지만 프로젝트 기간이 끝나 정보수집 단계에서 그치고 말았다. 다음 프로젝트때는 1순위로 구현해볼 예정인 기능.

▶ KAKAO 회원가입

이번 프로젝트에서 JWT 토큰을 사용하여 로그인 유저를 체크하는데, 카카오 회원가입 시 토큰이 이와 정상적으로 상호작용 되는지 고민하던 중, 프로젝트 기간이 끝나 적용하지 못하였다.
다음 프로젝트 때는 충분하게 숙지하고 공부하여 적용해볼 예정이다.

▶ GOOGLE Recaptcha

API 사용 난이도가 어렵고 시간이 많이들어 구현을 하지 못하였다.
다음 프로젝트 때는 충분하게 숙지하고 공부하여 적용해볼 예정이다.

▶ 근처 동물병원 마킹

초기에 동물병원을 관심지역 지도에 표기하려 했으나, 지역별 데이터로 부분적으로 제공이 되어 크롤링하고 합치는데 많은 시간이 소요될 것으로 예상되어 이번 프로젝트에서 제외하였다.

□ 추가구현 사유

▶ 관심지역 실거래가 그래프

사용자가 관심지역의 거래금액 내역을 연도순으로 파악하여 내가 사려고 하는 집의 가격의 변화를 그래프로 알아볼 수 있게끔 만들었다.

▶ 게시판 좋아요 기능 및 댓글 페이징

사용자 간의 소통이 원활할 수 있도록 하는 댓글 기능 중, 페이징 기능으로 UI 를 개선하였으며 유저들이 게시판 글 중에서 유익한 글들을 확인할 수 있도록 좋아요 기능을 추가하였다.