



seok830621



# DAEGU PLATE

Deagu plate is a Web project created by 남지윤, 이재진, 장원석, 김선철

# CONTENTS



## 01 PROFILE

- 팀원 소개



## 03 PERIOD

- 제작 기간



## 05 INTRODUCE

- 프로젝트 소개



## 02 PLAN

- 기획 의도



## 04 SERVICE

- 서비스 개요



## 06 USER SERVICE

- 유저 서비스



seok830621



# 01 PROFILE

## 팀원 소개

# 01 PROFILE 팀원 소개



seok830621



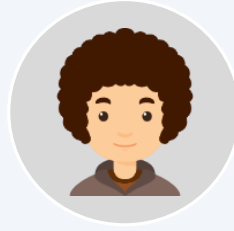
**이재진**

Planning

Data-Base

Back-End

API 연동



**김선철**

Data-Base

Back-End

API 연동



**남지윤**

Chatting Service

Front-End



**장원석**

Planning

Front-End



seok830621



## 02 PLAN

기획 의도

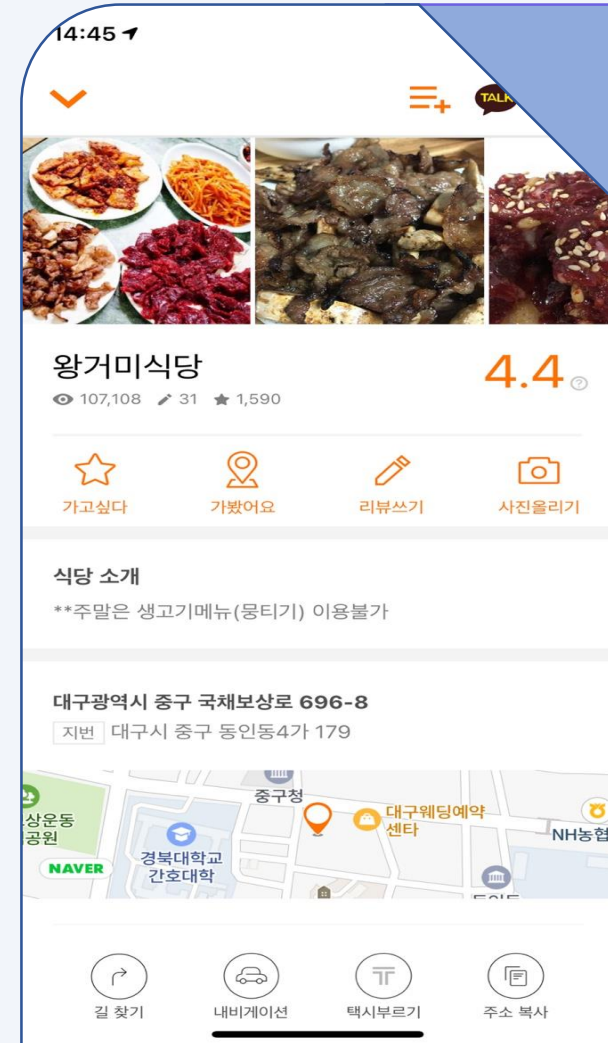
## 02 PLAN 기획 의도



seok830621



1. 망고 플레이트 대구 인기 맛집 TOP 200 추천
2. 맛집 가게 정보 열람이 가능하도록 디테일 페이지 구현
3. 유저간에 실시간 채팅으로 맛집 정보 공유





seok830621



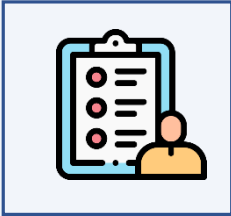
## 03 PERIOD

제작 기간

## 03 PERIOD 제작기간

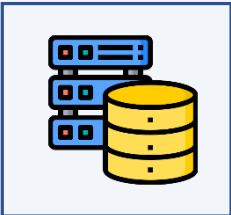


seok830621



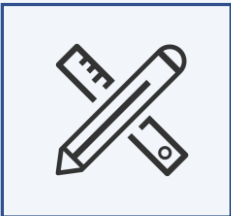
Planning

1 weeks



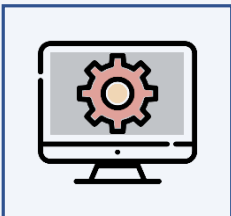
Data-Base

2 weeks



Front-end

3 weeks



Back-end

3weeks





seok830621



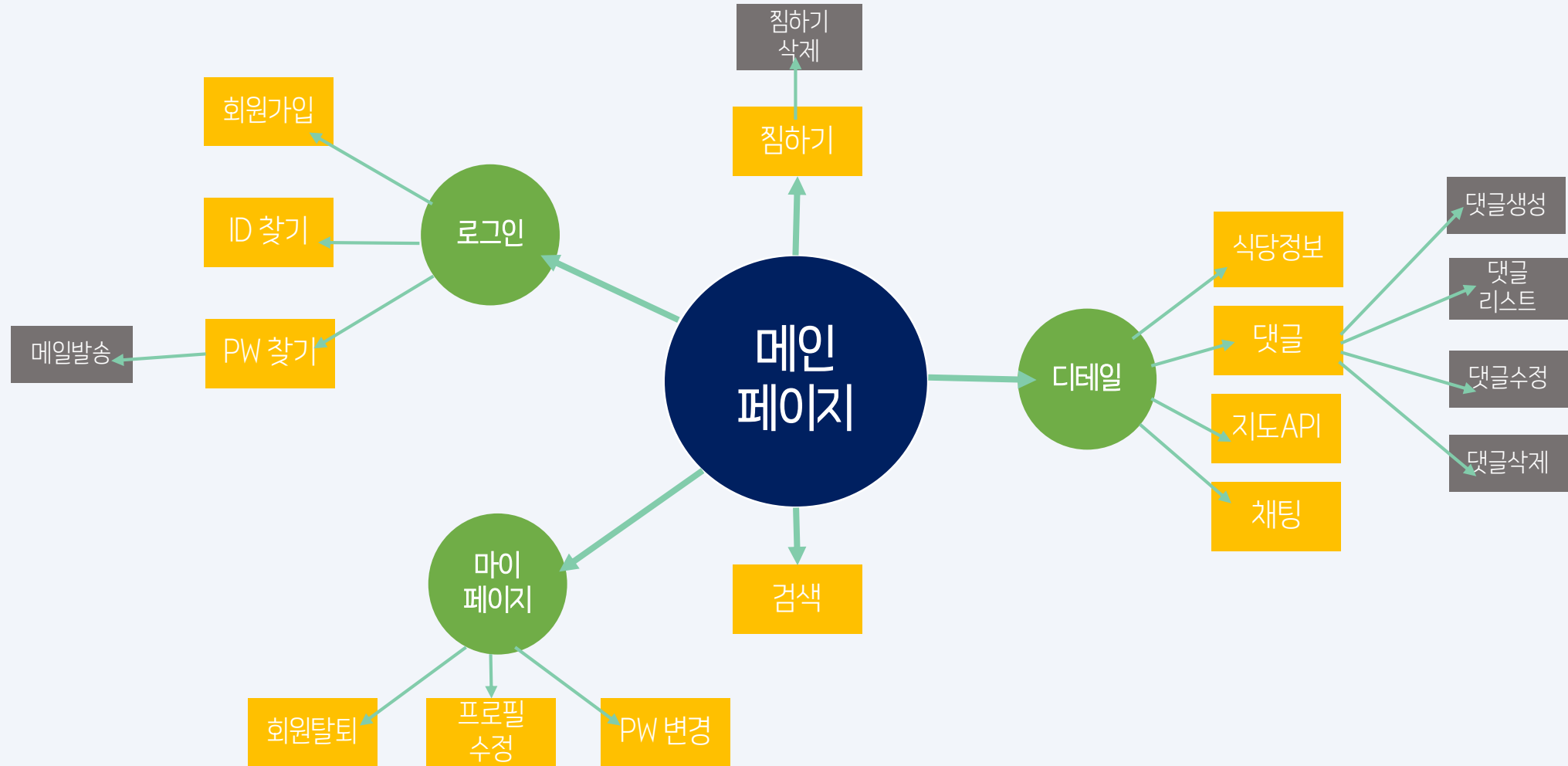
## 04 SERVICE

### 서비스 개요

## 04 SERVICE 서비스 개요



seok830621





seok830621



# 05 INTRODUCE

## 프로젝트 소개

## 05 INTRODUCE 프로젝트 소개

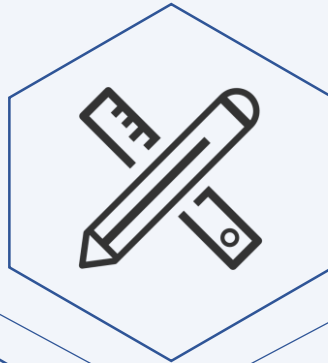


seok830621



개발 환경

01



테이블 명세표

03



02



Data-Base 관계도

## 05 INTRODUCE 프로젝트 소개 - 01. 개발 환경



FRONT-END  
LANGUAGE  
FrameWork

html, css, javascript

BACK-END  
LANGUAGE  
FrameWork

java, javascript, ajax, jsoup,  
json, spring, Tiles,  
jBcrypt, Servlet&JSP, OAuth



운영체제

Window 10  
Mac os



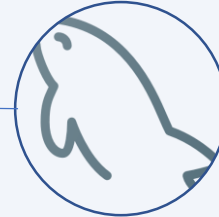
소스코드 작성도구

vs code  
Eclipse



웹 서버

Apache Tomcat  
9.0



데이터 베이스

MySQL  
8.0.23



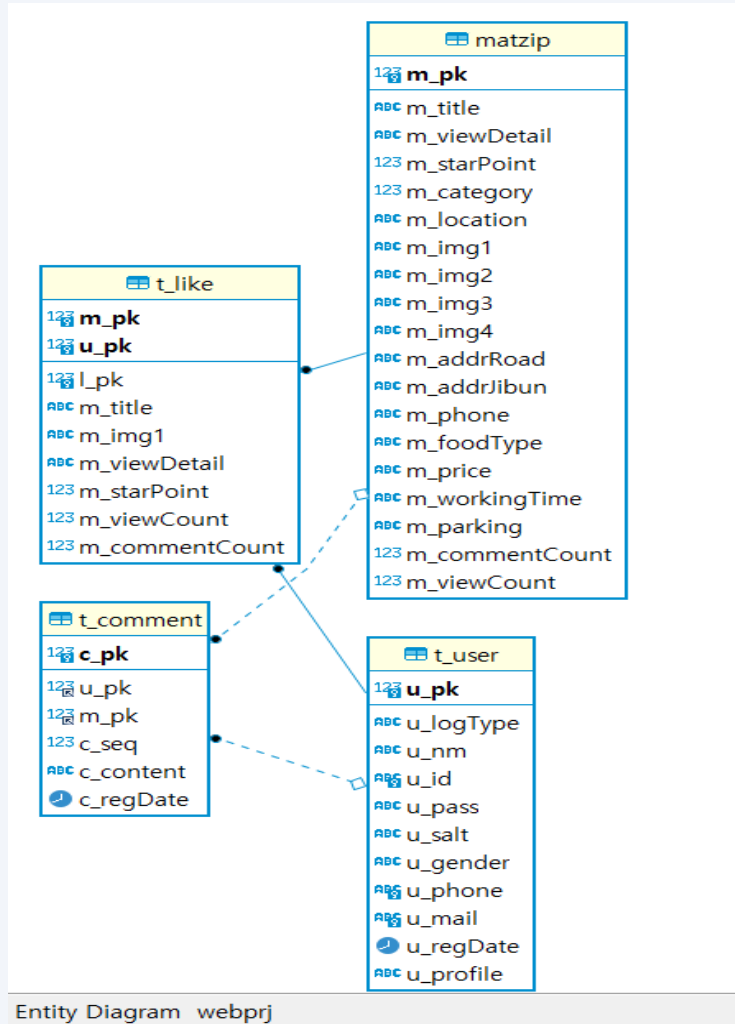
형상관리도구

GitHub

## 05 INTRODUCE 프로젝트 소개 - 02. Data-Base 관계도



seok830621



t\_user : 유저 정보 테이블

matzip : 가게 정보 테이블

t\_like : 좋아요 테이블

t\_comment : 댓글 테이블

## 05 INTRODUCE 프로젝트 소개 - 03. 테이블 명세표



seok830621



### 유저 정보 Table

Table	t_user					
No	FieldName	DataType	Null	Key	Extra	Comment
1	u_pk	int	NotNull	Primary Key	Auto_Increment	회원번호
2	u_logType	varchar(30)	NotNull		Default 'user'	로그인타입
3	u_nm	varchar(20)	NotNull			이름
4	u_id	varchar(20)	NotNull	Unique		아이디
5	u_pass	varchar(300)	Null			패스워드
6	u_salt	varchar(200)	Null			암호화키
7	u_gender	varchar(10)	Null			성별
8	u_phone	varchar(30)	Null	Unique		전화번호
9	u_mail	varchar(30)	Null	Unique		메일주소
10	u_regDate	datetime	Null		DEFAULT CURRENT_TIMESTAMP	회원등록일자
11	u_profile	varchar(400)	Null			프로필사진

## 05 INTRODUCE 프로젝트 소개 - 03. 테이블 명세표



### 가게정보 Table

Table	matzip					
No	FieldName	DataType	Null	Key	Extra	Comment
1	m_pk	int	NotNull	Primary Key	Auto_Increment	번호
2	m_title	varchar(20)	NotNull			이름
3	m_viewDetail	varchar(50)	NotNull			상세정보
4	m_starPoint	float	Null			별점
5	m_category	int	Null		Default 0	카테고리
6	m_location	varchar(10)	NotNull			위치
7	m_img1	varchar(500)	NotNull			이미지1
8	m_img2	varchar(500)	NotNull			이미지2
9	m_img3	varchar(500)	NotNull			이미지3
10	m_img4	varchar(500)	NotNull			이미지4
11	m_addrRoad	varchar(100)	Null			도로명주소
12	m_addrJibun	varchar(400)	Null			지번주소
13	m_phone	varchar(300)	Null			전화번호
14	m_foodType	varchar(300)	Null			음식종류
15	m_price	varchar(300)	Null			가격대
16	m_workingTime	varchar(100)	Null			영업시간
17	m_parking	varchar(300)	Null			주차여부
18	m_commentCount	int	Null		Default 0	댓글수
19	m_viewCount	int	Null		Default 0	조회수





seok830621



### 종아요 Table

Table	t_like					
No	FieldName	DataType	Null	Key	Extra	Comment
1	l_pk	int	NotNull	Unique	Auto_Increment	번호
2	m_pk	int	NotNull	Primary Key	REFERENCES matzip(m_pk) ON delete cascade	가게번호
3	u_pk	int	NotNull	Primary Key	REFERENCES t_user(u_pk) ON delete cascade	회원번호
4	m_title	varchar(50)	Null			가게이름
5	m_img1	varchar(500)	NotNull			썸네일
6	m_viewDetail	varchar(50)	NotNull			가게상세정보
7	m_starPoint	float	Null			가게별점
8	m_viewCount	int	Null			조회수
9	m_commentCount	int	Null			댓글수



seok830621



### 댓글 Table

Table	t_comment					
No	FieldName	DataType	Null	Key	Extra	Comment
1	c_pk	int	NotNull	Primary Key	Auto_Increment	댓글고유번호
2	u_pk	int	Null	Foreign Key	REFERENCES t_user(u_pk) ON delete cascade	회원번호
3	m_pk	int	Null	Foreign Key	REFERENCES matzip(m_pk) ON delete cascade	가게번호
4	c_seq	int	Null			가게별댓글번호
5	c_content	varchar(200)	NotNull			댓글내용
6	c_regDate	datetime	Null			댓글등록일자



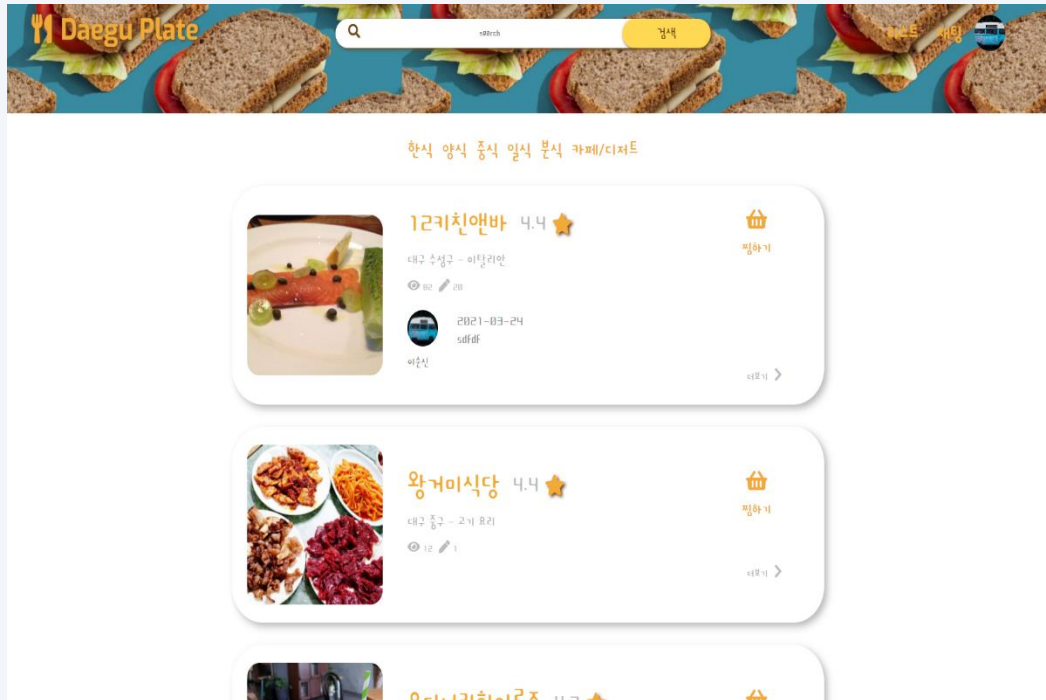
seok830621



# 06 USER SERVICE

유저 서비스

### < 주요 기능 >



💡 SNS (카카오,네이버) 로그인 서비스

💡 이메일을 이용한 PW 찾기

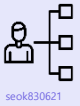
💡 jsoup 라이브러리를 이용한 망고플레이트

대구 인기 검색 맛집 top 200 데이터 크롤링

💡 kakao 지도 api 사용

💡 Rest Api를 이용한 댓글 CRUD구현

💡 자바 WebSocket을 활용한 실시간 채팅



### 로그인 화면

**Daegu Plate**

*id*

*password*

로그인    회원가입    ID/PW 찾기

**N**    TALK

Oauth2.0을 이용한 SNS 로그인

- 로그인 : 유저의 아이디와 비밀번호를 받아 로그인



seok830621



### 아이디 / 비밀번호 찾기 화면

## Find Id Pw

Find ID

Find PW

\* 인증된 메일만 찾기 가능 \*

아이디

이메일

확인

취소

Login

localhost:8181 내용:  
고객님의 이메일로 임시비밀번호를 발송 하였습니다.

확인

## Find Id Pw

Find ID

Find PW

\* 인증된 메일만 찾기 가능 \*

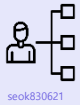
shzhem764

shzhem764@naver.com

확인

취소

Login



seok830621



### 임시 비밀번호 발송 화면 & 메일 관련 소스코드

☆ 임시비밀번호입니다 ✉

보낸사람 이재진 <shzhem764@naver.com>

받는사람 utf-8 <shzhem764@naver.com>

### 안녕하세요

귀하의 임시 비밀번호는 7341074737입니다.

로그인후 수정 부탁드립니다

```
@Component
public class MailUtils {

    public void sendMail(UserEntity param) {

        // Mail Server 설정
        String charset = "utf-8";
        String hostSMTP = "smtp.naver.com"; //네이버 이용시 smtp.naver.com/gmail이용시 smtp.gmail.com
        String hostSMTPid = "shzhem764@naver.com"; //서버이메일(보내는사람이메일주소)
        String hostSMTPpwd = "shzhem764@naver.com"; //서버이메일 비밀번호(보내는 사람 비밀번호)

        // 보내는 사람 Email, 제목, 내용
        String fromEmail = "shzhem764@naver.com"; //보내는 사람 이메일주소(받는 사람 이메일에 표시됨)
        String fromName = "이재진";
        String subject = "";
        String msg = "";

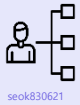
        subject = "임시비밀번호입니다";

        msg += "<h1>안녕하세요</h1>"
            + " 귀하의 임시 비밀번호는 " + param.getU_pass() + "입니다.<br>"
            + "로그인후 수정 부탁드립니다";

        String mail = param.getU_mail();
        try {
            HtmlEmail email = new HtmlEmail();
            email.setDebug(true);
            email.setCharset(charset);
            email.setSSL(true);
            email.setHostName(hostSMTP);
            email.setSmtpport(587); //네이버 이용시 587

            email.setAuthentication(hostSMTPid, hostSMTPpwd);
            email.setTLS(true);
            email.addTo(mail, charset);
            email.setFrom(fromEmail, fromName, charset);
            email.setSubject(subject);
            email.setHtmlMsg(msg);
            email.send();
            System.out.println("메일 전송 완료");
        } catch (Exception e) {
            System.out.println("메일 전송 실패");
            e.printStackTrace();
        }
    }
}
```

## 데이터 크롤링 과정



```
@Component
public class CrawlingUtils {

    //디렉터리로 가는 쿼리스트링 String이중배열로 넘어감
    public static String[][] getDetailString() throws Exception {

        //망고플레이트 페이지 수(1의값 20페이지)
        final String URL = "https://www.mangoplate.com/search/"
            + "%EB%8C%80%EA%B5%AC%EB%A7%9B%EC%A7%91?keyword=%EB%8C%80%EA%B5%AC%EB%A7%9B%EC%A7%91&page=";
        String[][] matZipString = new String[10][20];

        for (int i = 1; i <= 10; i++) {

            Document doc = Jsoup.connect(URL + i).get();

            Elements list = doc.select("div.info a");
            int j = 0;
            for (Element e : list) {
                matZipString[i - 1][j] = e.attr("href").substring(13);
                j++;
            }
        }
        for (int i = 0; i < 10; i++) {

            for (int j = 0; j < 20; j++) {

            }
        }
        return matZipString;
    }
}
```

```
public static List<MatZipEntity> insMatziList() throws Exception{

    //페이지 URL
    String PageUrl = "https://www.mangoplate.com/search/%EB%8C%80%EA%B5%AC%EB%A7%9B%EC%A7%91?keyword=%EB%8C%80%EA%B5%AC%EB%A7%9B%EC%A7%91&page=";
    String detailUrl = "https://www.mangoplate.com/restaurants/";
    //리턴할 맛집리스트(top200)
    List<MatZipEntity> matzipList = new ArrayList<>();

    String[][] matZipString = getDetailString();

    for (int i = 1; i <= 10; i++) {

        //페이지에 1값을 더해줌(10페이지)
        Document PageDoc = Jsoup.connect(PageUrl+i).get();
        System.out.println("에러페이지= "+i);

        for (int j = 0; j < 20; j++) {
            //객체 생성
            MatZipEntity matzipEntity = new MatZipEntity();

            Elements parseTitle = PageDoc.select("h2.title");
            Elements parseCategory = PageDoc.select("p.etc");
            Elements parseStarPoint = PageDoc.select("strong");

            String title = parseTitle.get(j).text();
            String viewDetail = parseCategory.get(j).text();
            String starPoint = parseStarPoint.get(j).text();
            String location_ = viewDetail.substring(3);

            String location = "";
        }
    }
}
```



### 데이터 크롤링 과정

```
int index_num = location_.indexOf("구");

if(index_num == 1 ) {
    location = location_.substring(0,index_num);
}
location = location_.substring(0,index_num+1);

matzipEntity.setM_title(title);
matzipEntity.setM_viewDetail(viewDetail);

if(starPoint != null && starPoint.length() > 0) {
    matzipEntity.setM_starPoint(Float.parseFloat(starPoint));
}else {
    matzipEntity.setM_starPoint(0);
}

matzipEntity.setM_location(location);

//디테일 url 추출
Document detailDoc = Jsoup.connect(detailUrl+matZipString[i-1][j]).get();

//디테일 이미지 추출
Elements parseDetailImg = detailDoc.select("img.center-cropping[src]");
matzipEntity.setM_img1(parseDetailImg.get(0).attr("src"));
matzipEntity.setM_img2(parseDetailImg.get(1).attr("src"));
matzipEntity.setM_img3(parseDetailImg.get(2).attr("src"));
matzipEntity.setM_img4(parseDetailImg.get(3).attr("src"));

//디테일 정보 추출
Elements parseDetailInfo = detailDoc.select("tbody td");
String addrTotal;
```

```
addrTotal = parseDetailInfo.get(0).text();
int int1 = addrTotal.indexOf("지번");
addrRoad = addrTotal.substring(0,int1-1);
addrJibun = addrTotal.substring(int1+3, addrTotal.length());

phone = parseDetailInfo.get(1).text();
foodType = parseDetailInfo.get(2).text();

price = parseDetailInfo.get(3).text();
parking = parseDetailInfo.get(4).text();
workingTime = parseDetailInfo.get(5).text();

matzipEntity.setM_addrJibun(addrJibun);
matzipEntity.setM_addrRoad(addrRoad);
matzipEntity.setM_phone(phone);
matzipEntity.setM_foodType(foodType);
matzipEntity.setM_price(price);
matzipEntity.setM_parking(parking);
matzipEntity.setM_workingTime(workingTime);

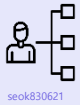
matzipList.add(matzipEntity);

}

}

return matzipList;
}
```

## 카카오 지도 API 사용



### 왕거미식당 4.4

리뷰쓰기 평하기

13

주소	대구광역시 중구 국제보상로 696-8
지번	대구시 중구 동인동4가 179
전화번호	053-427-6380
온라인주문가능	고게 요리
가격대	4만 원 이상
주차	주차 공간 없음
영업시간	17:00 - 23:00

재탕시작

```
// 지도 관련
// 지도를 표시할 div
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

// 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);

// 주소-좌표 변환 객체를 생성합니다
var geocoder = new kakao.maps.services.Geocoder();

// 주소로 좌표를 검색합니다
geocoder.addressSearch(mapAddr, function(result, status) {

    // 정상적으로 검색이 완료됐으면
    if (status === kakao.maps.services.Status.OK) {

        var coords = new kakao.maps.LatLng(result[0].y, result[0].x);

        // 결과값으로 받은 위치를 마커로 표시합니다
        var marker = new kakao.maps.Marker({
            map: map,
            position: coords
        });

        // 인포윈도우로 장소에 대한 설명을 표시합니다
        var infowindow = new kakao.maps.InfoWindow({
            content: `<div style="width:150px;text-align:center;padding:6px 0;">${mapTitle}</div>`
        });
        infowindow.open(map, marker);

        // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
        map.setCenter(coords);
    }

});
```



seok830621



### Rest API를 이용한 댓글 CRUD 구현

리뷰 (total)

 남지은	2021-03-25 남자친구랑 다시 오고 싶어요	 
 이재진	2021-03-25 다시 오고 싶은 마지막이에요	
 이재진	2021-03-25 너무 맛있어요	

더보기

```
@RestController
@RequestMapping("/cmt")
public class CmtController {

    @Autowired
    private CmtService service;

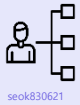
    @Autowired
    private SecurityUtils sUtils;

    // 댓글 리스트
    @GetMapping
    public List<CommentDomain> list(CommentDomain p, @RequestParam("listCount") int listCount){
        p.setListCount(listCount);
        List<CommentDomain> data = service.selCmtList(p);
        return data;
    }

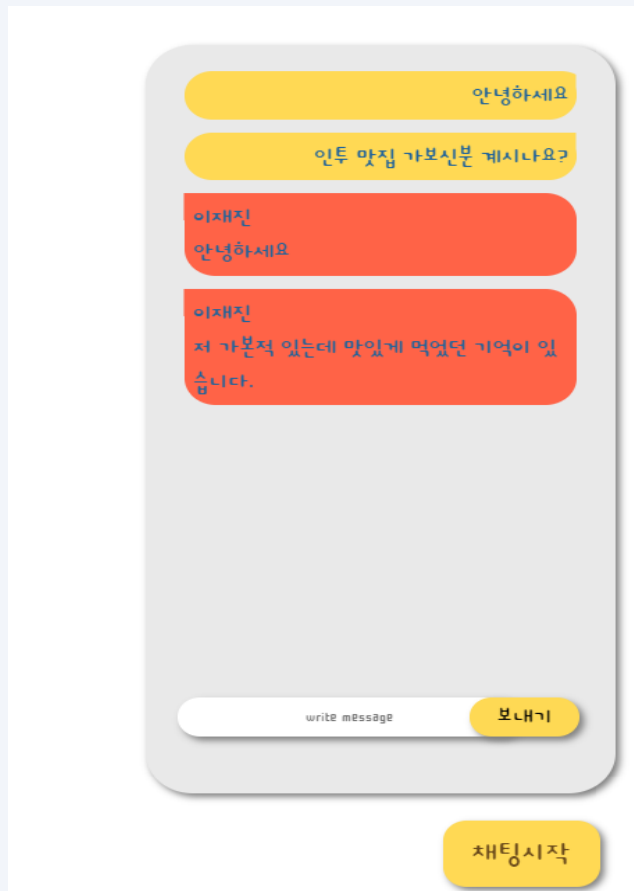
    // 댓글 작성
    @PostMapping
    public int ins(@RequestBody CommentEntity p, HttpSession hs) {
        p.setU_pk(sUtils.getLoginUserPk(hs));
        return service.insCmt(p);
    }

    // 댓글 수정
    @PutMapping
    public int upd(@RequestBody CommentEntity p, HttpSession hs) {
        p.setU_pk(sUtils.getLoginUserPk(hs));
        return service.updCmt(p);
    }

    // 댓글 삭제
    @DeleteMapping
    public int del(CommentEntity p, HttpSession hs) {
        p.setU_pk(sUtils.getLoginUserPk(hs));
        return service.delCmt(p);
    }
}
```



### 웹소켓을 활용한 실시간 채팅



```
@Component
public class SocketHandler extends TextWebSocketHandler {

    HashMap<String, WebSocketSession> sessionMap = new HashMap<>(); // 웹소켓 세션을 담아둘 맵

    @Override
    public void handleTextMessage(WebSocketSession session, TextMessage message) {
        // 메시지 발송
        String msg = message.getPayload();
        JSONObject obj = jsonObjectParser(msg);
        for (String key : sessionMap.keySet()) {
            WebSocketSession wss = sessionMap.get(key);
            try {
                wss.sendMessage(new TextMessage(obj.toJSONString()));
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    @SuppressWarnings("unchecked")
    @Override
    public void afterConnectionEstablished(WebSocketSession session) throws Exception {
        // 소켓 연결
        super.afterConnectionEstablished(session);
        sessionMap.put(session.getId(), session);
        JSONObject obj = new JSONObject();
        obj.put("type", "getId");
        obj.put("sessionId", session.getId());
        session.sendMessage(new TextMessage(obj.toJSONString()));
    }
}
```

```
@Override
public void afterConnectionClosed(WebSocketSession session, CloseStatus status) throws Exception {
    // 소켓 종료
    sessionMap.remove(session.getId());
    super.afterConnectionClosed(session, status);
}

//메시지 전송시 JSON파싱을 위해 message.getPayload()를 통해
//받은 문자열을 만든 함수 jsonObjectParser에 넣어서 JSONObject값으로 받아서 강제 문자열 형태로 보내주는부분
private static JSONObject jsonObjectParser(String jsonStr) {
    JSONParser parser = new JSONParser();
    JSONObject obj = null;
    try {
        obj = (JSONObject) parser.parse(jsonStr);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    return obj;
}
```

*The End*