



What should we EAT?

Team. 뭐먹지 | 은명진, 김세은, 김민주, 최규, 천상환



CONTENTS

01 프로젝트 소개

- 기획의도
- 역할
- 개발환경

02 분석/설계

- GanttChart / UseCase / UI StoryBoard
- ERD / Class Diagram / Sequence Diagram

03 샘플 서비스 시연

04 기술소개

05 소감



SELECTED FOR THE BEST OF THE BEST

001

프로젝트 소개





오늘 뭐 먹지?

남녀노소 누구나 구분없이 매일 한번쯤은 하는 고민

#식단관리

내가 먹은 음식을 기록해요.
취향에 맞는 음식을 추천해드려요.

#음식점 추천

마음에 숨겨둔 나의 인생 맛집!
널리 알려서 많은 사람들과 공유해요.

#레시피 공유

나만이 만든 나만의 레시피
요즘 인기인 레시피를 추천해드려요.

01 프로젝트 소개

역할



은명진

레시피 기능
REST api 구성
Bootstrap 적용
Thymleaf 레이아웃
이미지 업로드



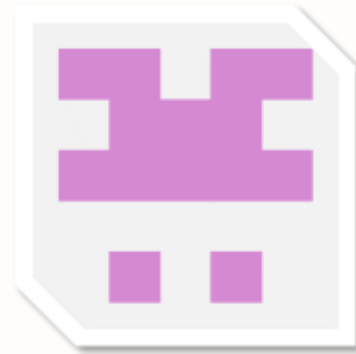
천상환

식단 관리 기능
회원 관리 기능
음식점 관리 기능
Bootstrap 적용



최규

식단관리 기능
식단 작성 기능
사다리타기 api 조정
Bootstrap 적용



김세은

회원관리 기능
Bootstrap 적용



김민주

음식점 기능
음식점 추천
Bootstrap 적용

01 프로젝트 소개

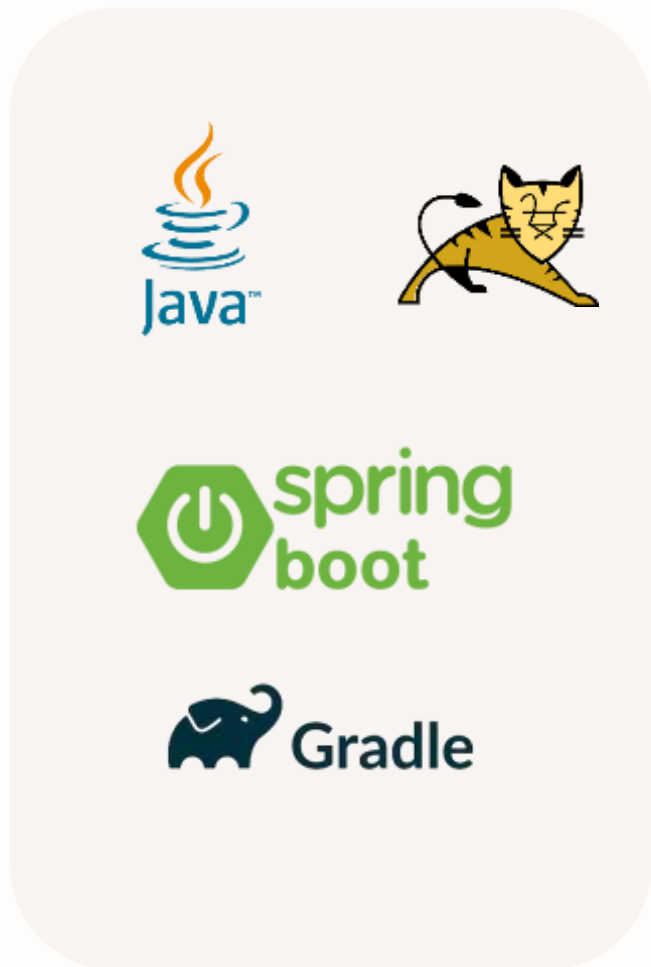
개발환경



VIEW



Controller



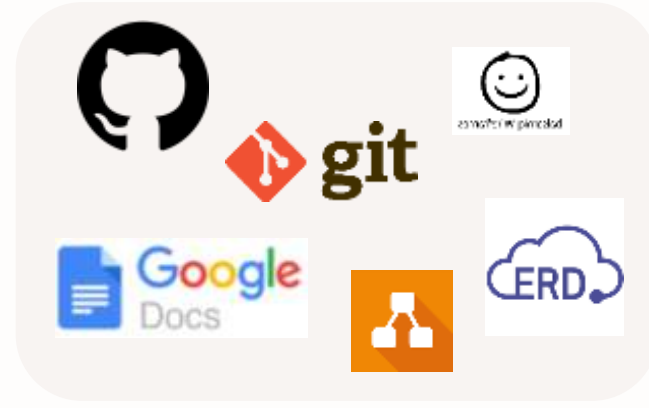
DB



IDE



Tool





002

002

분석/설계



02 분석/설계 참고사이트



Q
로그인
회원가입
만개스토어 바로가기

일반판
전
다이어트
김치
소고기
<
>

추천
분류
랭킹
매거진
클래스
더보기

우리 밥상을 색다르게 만들어줄
우리품종 경기버섯 요리공모전

제스프리 그린키위
레시피 공모전

2021 한국광어
첫 요리공모전

쇼핑 BEST

[늦.반.겐] 와이즈랩 FD1 무선 가정용진공포장기 음식 압축기 ...

39,000원 70개 구매

★★★★★ (1)

[늦.반.겐] NEW 노와 쿡웨어 7종 세트

68,900원 1770개 구매

★★★★★ (302)

[브랜드특가] 네오플램 샤베트 프라이팬/냄비 11종 모음(무...

28,700원 8501개 구매

★★★★★ (1,353)

[만개MADE] 19가지 자연원료로 만든 만개한알 (40g*3입) 1팩

9,900원 2545개 구매

★★★★★ (298)

트라발로 향수공병 퍼퓸팟 크리스탈

11,900원 73개 구매

★★★★★ (2)

02 분석/설계
참고사이트



메인가기 ▶

로그인 | 맛집찾기 | 쿠키 | 매거진 | 마이메뉴판 | 맛지도 | 이벤트 | 더보기 ▼

Menupan 전국맛집찾기

☐ 통합검색 ☒ 맛집

검색

지역별맛집 | 실시간인기맛집 | 테마별맛집 | 루프 맛집찾기 | 예약맛집

추천맛집하기 | 맛집신문고 | 맛집Q&A

강북

경기북부

인천

강남

강원

경기남부

충청

대전

대구

울산

전라

광주

경상

부산

제주

지역명을 클릭해 보세요

강북 지역 맛집찾기

찾고자 하는 상세 지역명을 클릭해주세요

광진/건대입구

광화문/시청

노원/도봉/미아

대학로

동대문

마포

명동

부암동/평창동

삼청동

상암

서촌/경복궁

성동/성수

성북

숙대/서출역

신촌/이대

안암/고대

연남동/연희동

연신내/불광

이촌동/용산

이태원/한남동

종로/인사동

충무로/신당동

홍대/상수/합정

회기

국내최초
실시간 전국맛집랭킹!

현지인들만 아는
진짜맛집

free TOP 1000

App.다운로드

오늘의 추천맛집 2021년 9월 17일, 깨다롭게 선정된 전국각지의 숨어있는 맛집을 추천합니다.

로이즈(ROYS) 제2롯데월드몰점

☎ 카페/주점 | 송파구 신원동

★★★★★ 3.6 | 22명 참여

로이즈는 건강한 아름다움을 추구하는 사람들을 위한 프리미엄 생과일 특화 빙수 전문점이다...

오늘의 추천맛집 자세히보기

신규 취재 맛집(전국)

new

이/미/지/문/비/중/임니다.

한정인추천 · 모범음식점

황토방

📍 한식 | 경기 | 여주군

저희 "황토방"을 찾아주셔서 감사합니다. 우리 "황토방"은 여...

new

췌스떡볶이강릉대학교로

📍 분식 | 강원 | 강릉시

귀게로 마는 췌기로 마오 췌기로오 기

맛집탐방 메뉴판도움과 회원들이 직접 찾아가 평가하고 분석한 리얼 맛집탐방

정통 중화요리

더보기

02 분석/설계
참고사이트

청정원 | 영양정보와 건강식단

균형과 건강 “푸드를 아는 사람들”

영양정보 조회

추천식단

나만의 식단

영양정보 게시판

식중독지수

청정원

상세보기

총 열량 0 kcal

나의 운동량 조회

선택

선택

닉네임

나만의 식단명

아침	점심	저녁	간식
+	+	+	+
+	+	+	+
+	+	+	+
+	+	+	+
+	+	+	+

+

행 추가

-

행 삭제

식단저장

청정원 브랜드관

classic 한국전통식품군

smart 편의형 제품

cuisine 서구식품군

fresh 냉장/냉동

essential 조미료/향신료

organic 유기농식품군

DAESANG

대성주식회사

청정원

건강한 커뮤니티

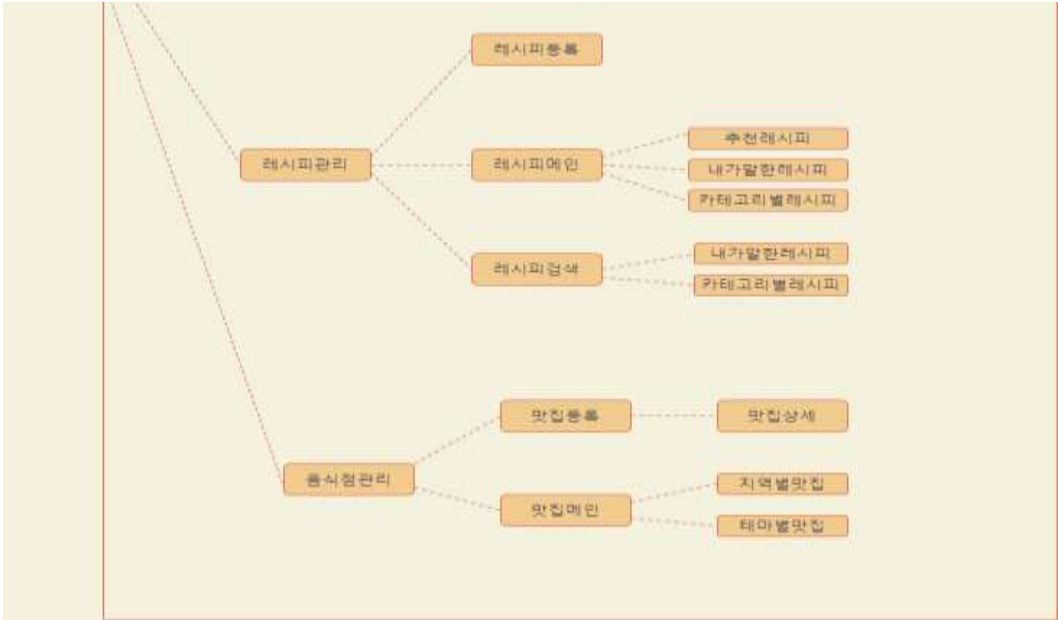
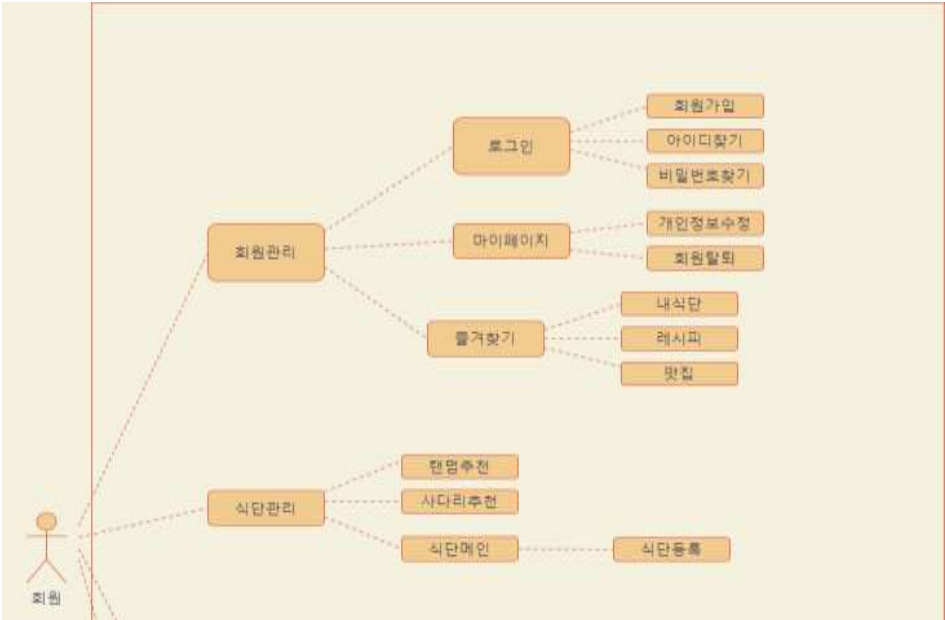
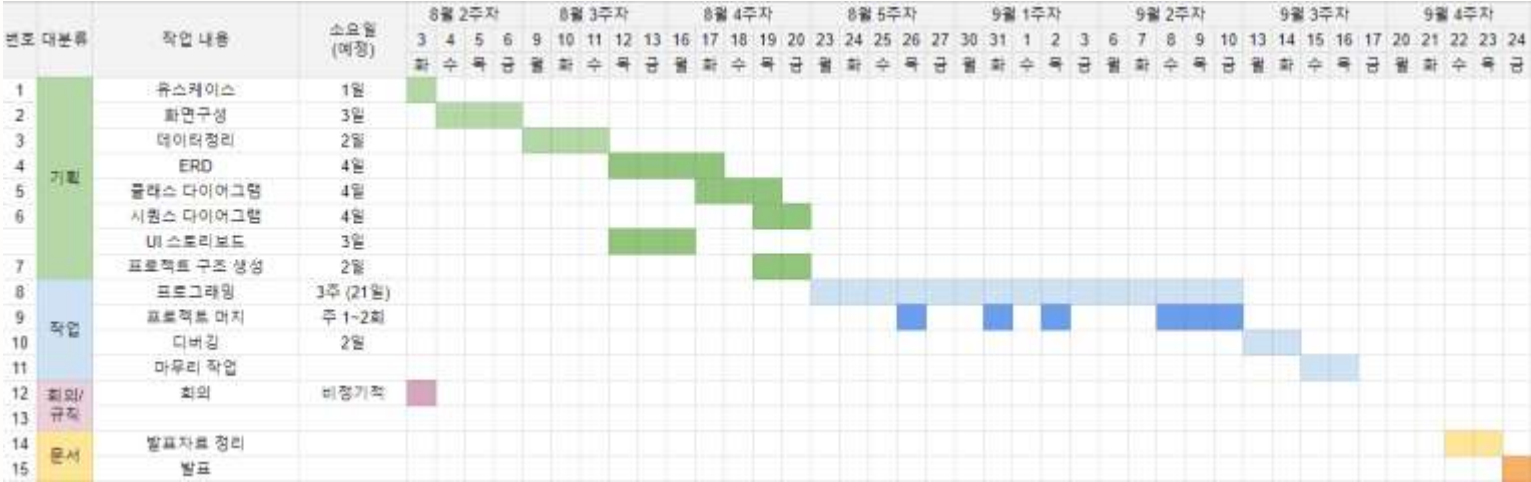
정원 e 샵

DAESANG

COPYRIGHT © DAESANG 2014 ALL RIGHTS RESERVED


02 분석/설계

GanttChart / UseCase



UI StoryBoard - Member

번호	회원 이름	파일 이름	담당자	작성일
U_002	회원가입	user_join	김세은	김세은



메뉴추천
레시피
맛집
마이페이지
로그인
회원가입

회원가입

A

B

번호	화면 이름	파일 이름	담당자	작성일
U_006	회원탈퇴	user_withdrawal	김세은	김세은



매뉴추천 레시피 맛집 마이페이지

식단등록 즐겨찾기 개인정보수정 회원탈퇴

회원탈퇴

A

B

번호	화면 이름	파일 이름	담당자	작성일
U_005	개인정보수정	user_favorite	김세은	김세은

메뉴추천 레시피 맛집 마이페이지
로그아웃

식단등록 즐겨찾기 개인정보수정 회원탈퇴

즐거찾기

레시피

맛집

<p style="font-size: small; text-align: center;">레시피이름 해로 애플 주원수</p>	<p style="font-size: small; text-align: center;">레시피이름 해로 애플 주원수</p>	<p style="font-size: small; text-align: center;">레시피이름 해로 애플 주원수</p>
<p style="font-size: small; text-align: center;">상호명 지역 주원수</p>	<p style="font-size: small; text-align: center;">상호명 지역 주원수</p>	<p style="font-size: small; text-align: center;">상호명 지역 주원수</p>



02 분석/설계

UI StoryBoard - Admin



번호	화면 이름	파일 이름	담당자	작성자	화면 설명
A_001	관리자 로그인	admin_login	김세은	김세은	뭐먹지 관리자 로그인 페이지

페이지 설명	
A	ID, PW 입력 창
B	로그인 버튼 - ID/PW 입력 시 - 회원관리 페이지로 이동

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
A_002	회원관리	admin_management	김세은	김세은	뭐먹지 관리자 회원관리 페이지

페이지 설명	
A	체크박스로 선택
B	체크박스로 선택 후 해당 부분 클릭 시 수정 가능
C	수정 버튼: 수정 후 클릭 삭제 버튼: 체크 후 클릭

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
A_003	게시글관리	admin_recipe	김세은	김세은	뭐먹지 게시글 관리 페이지

페이지 설명	
A	체크박스로 선택
B	등록 버튼 - 레시피 작성 페이지(R_003)로 이동 삭제 버튼 - A를 사용하여 1~n개 삭제 가능

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
A_004	게시글관리	admin_eatery	김세은	김세은	뭐먹지 게시글 관리 페이지

페이지 설명	
A	체크박스로 선택
B	등록 버튼 - 음식점 등록 페이지(E_002)로 이동 삭제 버튼 - A를 사용하여 1~n개 삭제 가능

02 분석/설계

UI StoryBoard - suggestion

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
P_001	랜덤 추천 페이지	suggestion_random	최규, 천상환	최 규	랜덤으로 레시피, 음식점 추천

페이지 설명	
A	추천 레시피 - 레시피 상세 페이지(R_004)로 이동 - 해당 레시피 이름, 재료, 태그 추천수 노출
B	추천 음식점 - 음식점 상세 페이지(R_002)로 이동 - 해당 음식점 상호명, 위치, 추천수 노출
C	한 번 더 버튼 - 새로운 추천 레시피, 추천 음식점 추천

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
P_002_1	사다리 타기 페이지	suggestion_ladder	최규, 천상환	최규, 천상환	사다리 타기 페이지

페이지 설명	
A	Input 박스 - 원하는 메뉴 입력
B	버튼 - 연달 선택 - 클릭 시 이어지는 선 출력 - 버튼 클릭 전까지 선 보이지 않음 - C 도착시 색상 변경
C	Input 박스 - 원하는 내용 입력
D	처음으로 버튼 - 사다리타기 시작 페이지(P_002)으로 전환

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
P_002	사다리 타기 시작 페이지	suggestion_ladder	최규, 천상환	최 규	사다리 타기 메인 페이지

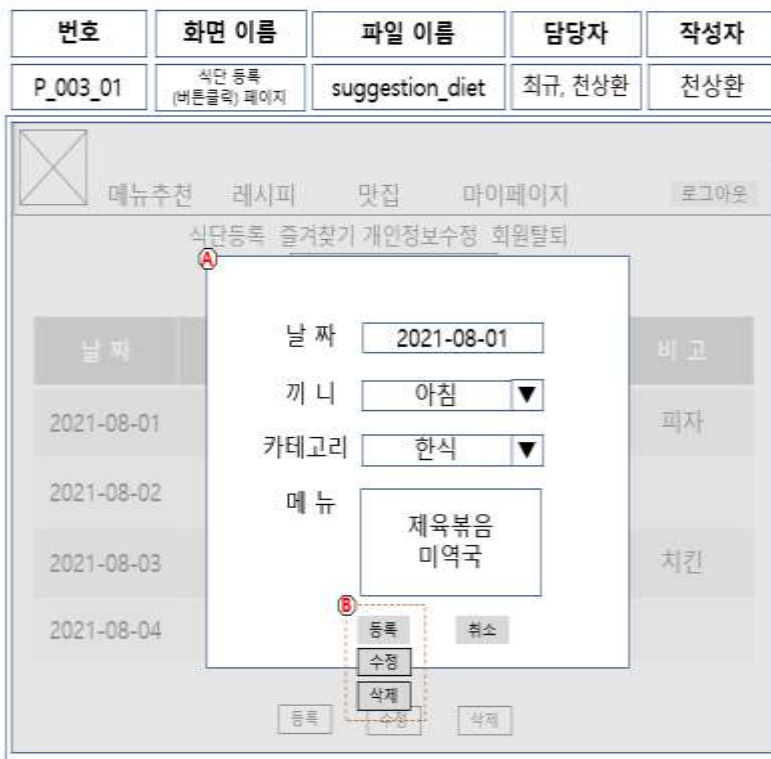
페이지 설명	
A	Input 박스 - 정해진 수에서 원하는 수를 입력
B	사다리 시작 버튼 - A에 입력된 내용 유효성 검사 - 유효성 검사 시 유효성 검사 내용 출력 - 사다리 타기 페이지(P_002_1)로 이동

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
P_003	식단 등록 페이지	suggestion_diet	최규, 천상환	최 규	사용자 메뉴 등록 페이지


페이지 설명	
A	식단 등록 - 사용자가 등록한 식단을 출력
B	등록, 수정, 삭제 버튼 - 등록: modal로 등록 창(P_003_1) 출력 - 수정: modal로 수정 창(P_003_1) 출력 - 삭제: modal로 삭제 창(P_003_1) 출력



UI StoryBoard - suggestion



번호	화면 이름	파일 이름	담당자	작성일	화면 설명
P_004	식단메뉴 추천 페이지	suggestion_advice	최규, 천상환	최 규	등록된 식단으로 메뉴 추천


메뉴추천 레시피 맛집 마이페이지 로그아웃

랜덤추천 사다리타기 식단메뉴추천

식단메뉴 추천

A

날 짜	아 침	점 심	저 녁	비 고
2021-08-01	한식	양식	말식	과자
2021-08-02		한식	중식	
2021-08-03			양식	치킨
2021-08-04	한식	일식	한식	

B

이번 주 식단 목록입니다.
 한식 n회, 양식 n회, 말식 n회, 중식 n회
 이번 주에 가장 적게 드신 ○○을 추천합니다.

C

화면 설명	
식단 등록 페이지 버튼 클릭	
페이지 설명	
A	<p>정보 입력 폼</p> <ul style="list-style-type: none"> - 날짜 : 선택 또는 입력 - 끼니 : 해당 끼니 선택 - 카테고리 : 해당 카테고리 선택 - 메뉴 : 식사한 메뉴 입력
B	<p>버튼</p> <ul style="list-style-type: none"> - 식단 등록 페이지에서 해당 버튼 클릭했을 경우 <ol style="list-style-type: none"> 1. 등록 : 등록 버튼 활성화, 입력 후 등록 버튼 클릭 2. 수정 : 수정 버튼 활성화, 원하는 날짜와 끼니에 등록된 메뉴 출력, 내용 변경 후 수정 버튼 클릭 3. 삭제 : 삭제 버튼 활성화, 원하는 날짜와 끼니에 등록된 메뉴 출력, 확인 후 삭제 버튼 클릭

02 분석/설계

UI StoryBoard - Eatery



번호	화면 이름	파일 이름	담당자	작성자	화면 설명
E_001	맛집메인페이지	eatery_main	김민주	김민주	원격지 맛집 메인 페이지

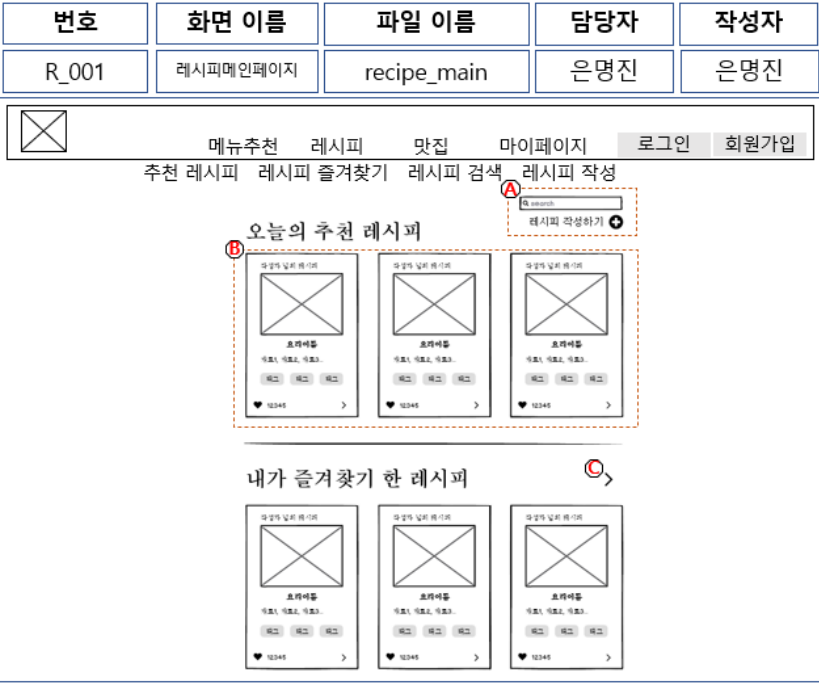
번호	화면 이름	파일 이름	담당자	작성자	화면 설명
E_003	맛집 상세 페이지	eatery_detail	김민주	김민주	원격지 맛집 상세 페이지

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
E_002	맛집 등록 페이지	eatery_create	김민주	김민주	원격지 맛집 등록 페이지

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
E_004	지역별 맛집 페이지	eatery_area	김민주	김민주	원격지 지역별 맛집 페이지

02 분석/설계

UI StoryBoard - Recipe

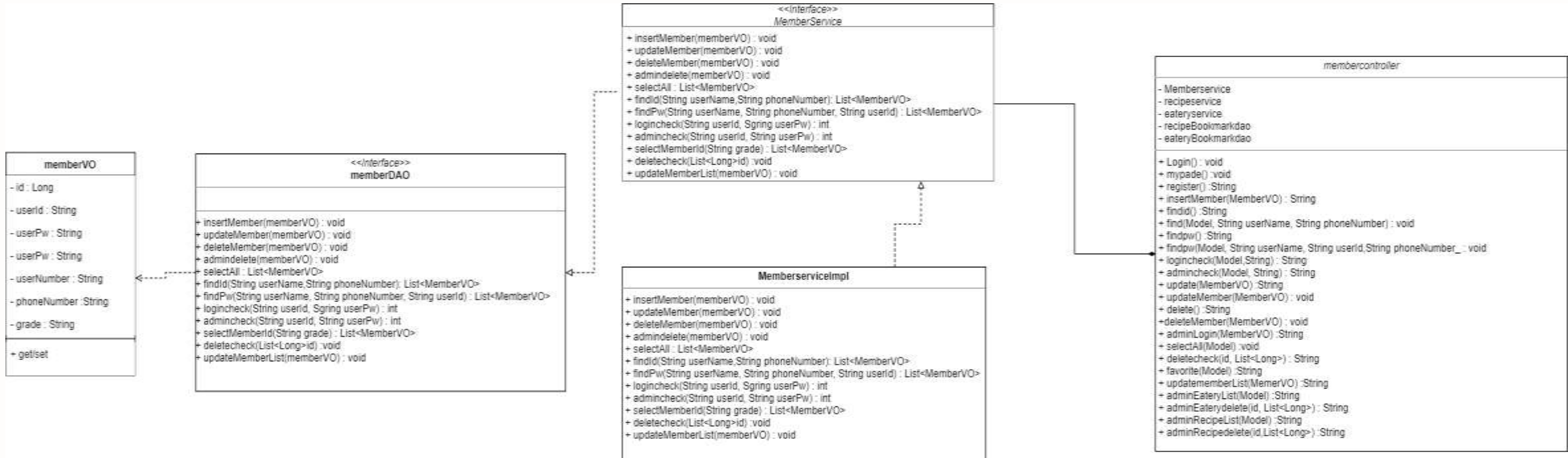


번호	화면 이름	파일 이름	담당자	작성자	화면 설명
R_001	레시피메인페이지	recipe_main	은명진	은명진	레시피 메인 페이지
페이지 설명					
검색바 - 입력한 단어를 포함한 검색결과를 출력 레시피 작성하기					

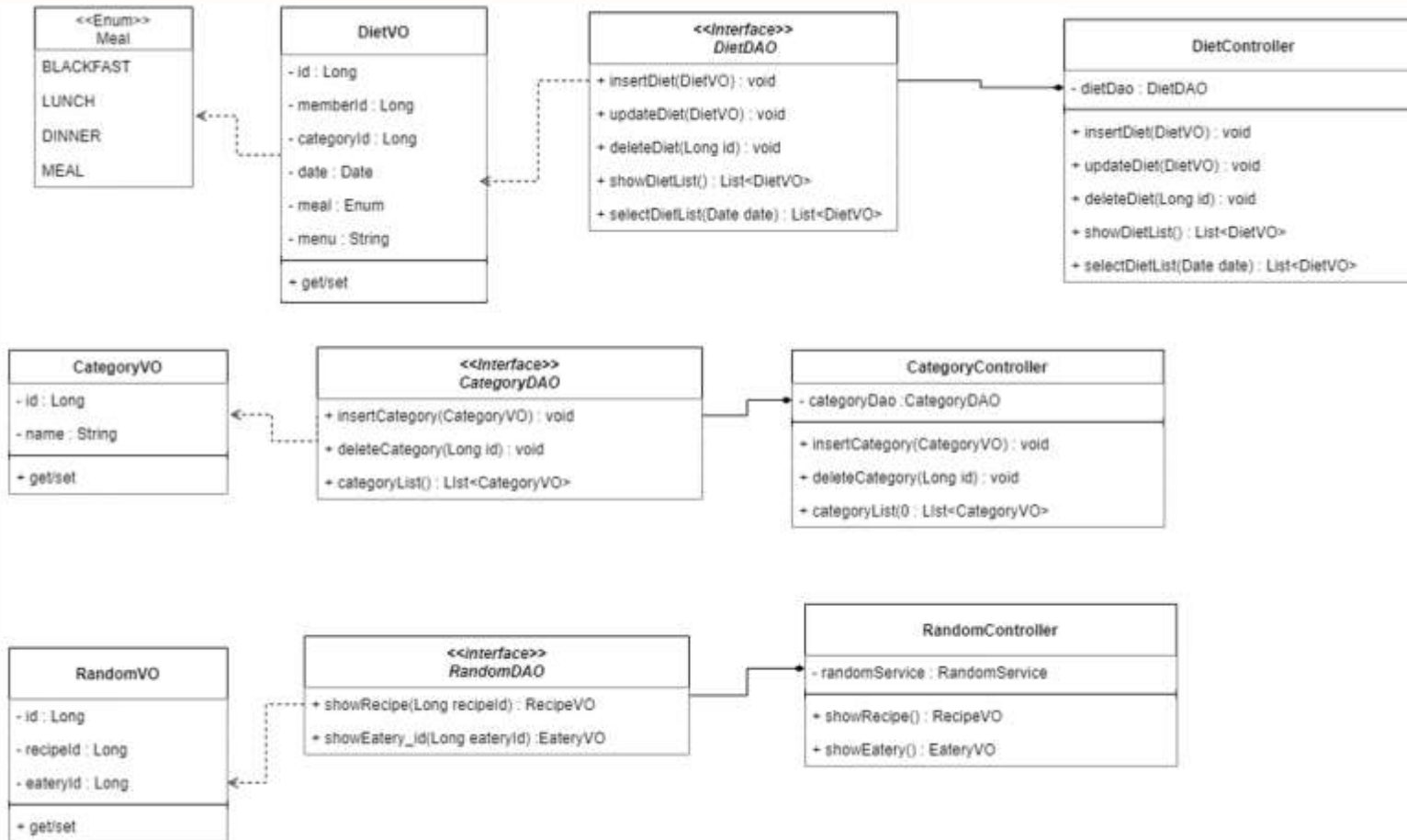
번호	화면 이름	파일 이름	담당자	작성자	화면 설명
R_003	레시피작성페이지	recipe_create	은명진	은명진	레시피 작성 페이지
페이지 설명					
메뉴추천 레시피 맛집 마이페이지 로그인 회원가입 추천 레시피 레시피 즐겨찾기 레시피 검색 레시피 작성 레시피 작성					

번호	화면 이름	파일 이름	담당자	작성자	화면 설명
R_003	레시피작성페이지	recipe_create	은명진	은명진	레시피 작성 페이지
페이지 설명					
이미지 삽입 - 작성할 레시피의 메인 이미지를 삽입. 클릭하여 파일 로드창을 노출 - 이미지는 똑같은 비율의 작은 사이즈의 이미지를 썸네일로 사용					
태그추가 - 해당 레시피의 태그를 추가 (최대 5개까지 추가 가능) - 클릭 시 태그를 입력할 modal을 노출					
요리 과정 - 이미지를 클릭하여 A와 같이 이미지 삽입 - 우측의 설명과 함께 하나의 요리 과정으로 데이터를 저장					
요리단계 추가 - 클릭하여 C의 칸을 한칸 추가 - 작성이 되지 않은 요리과정은 작성완료 버튼 클릭시 자동으로 삭제					

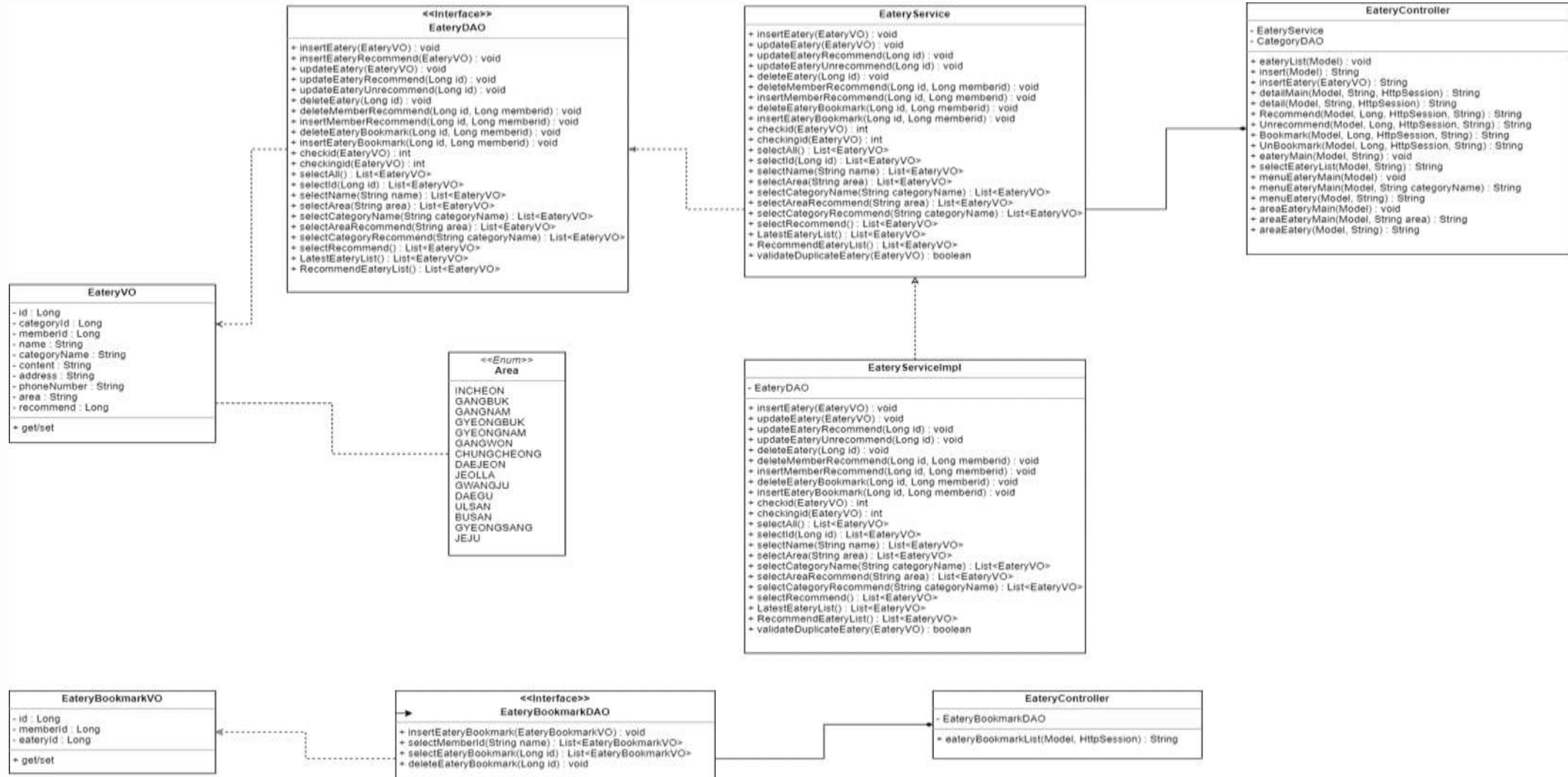




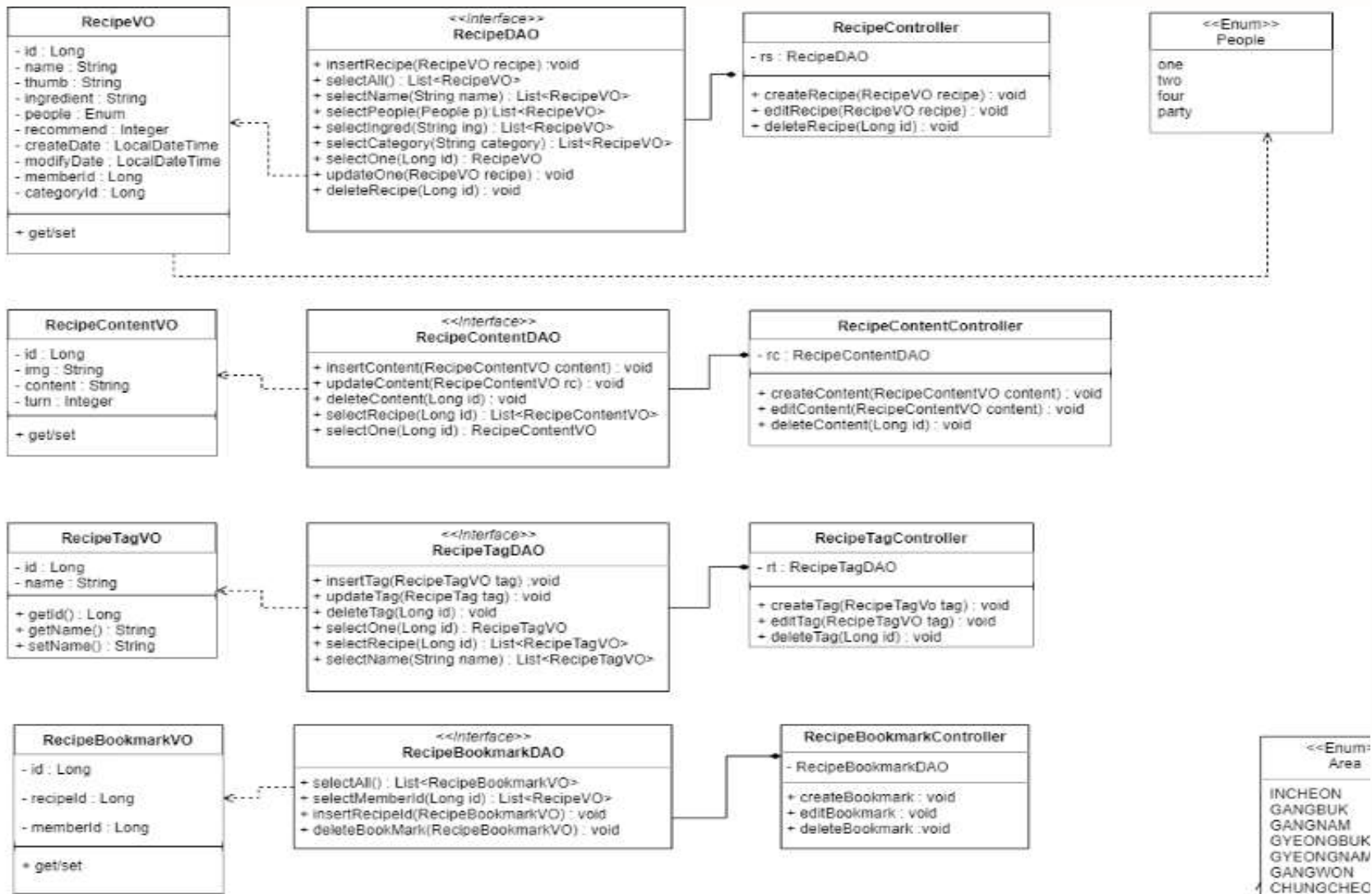
Class Diagram - Suggestion



Class Diagram - Eatery

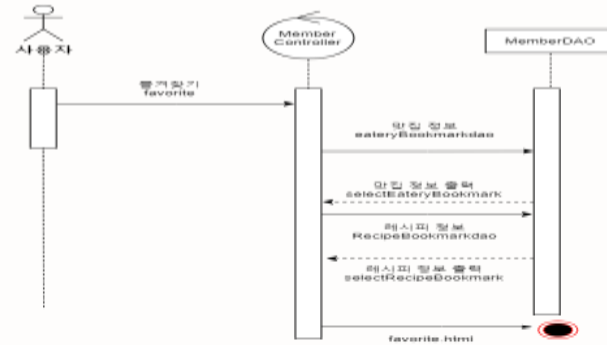
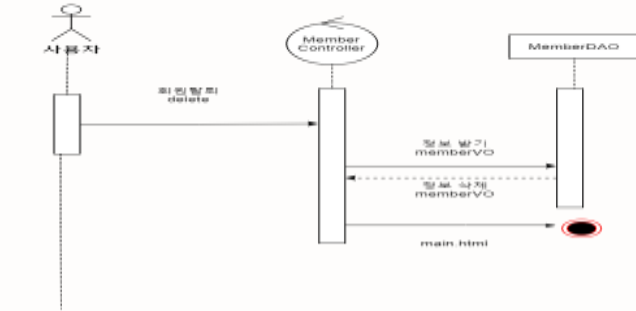
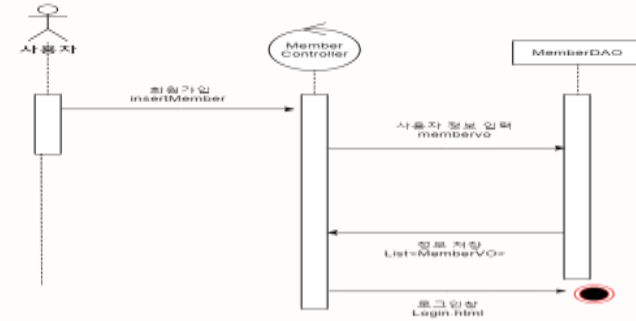
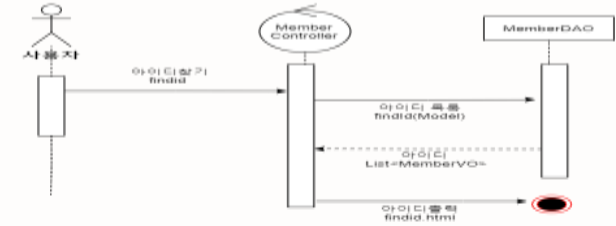
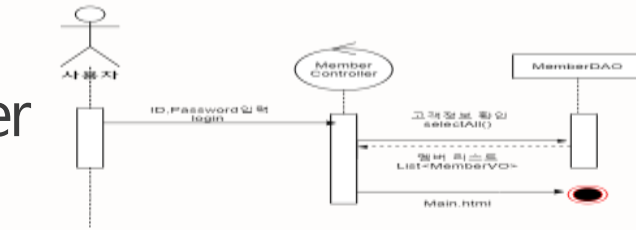


Class Diagram - Recipe

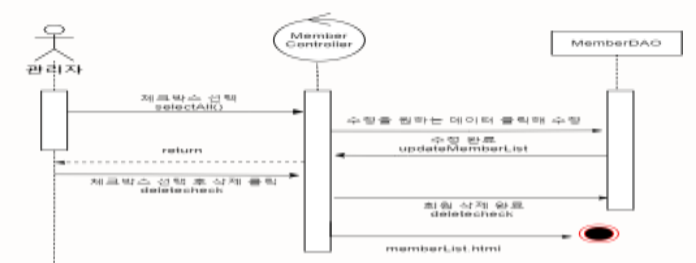
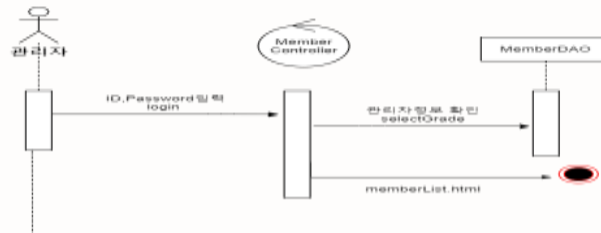


02 분석/설계

Sequence Diagram - Member

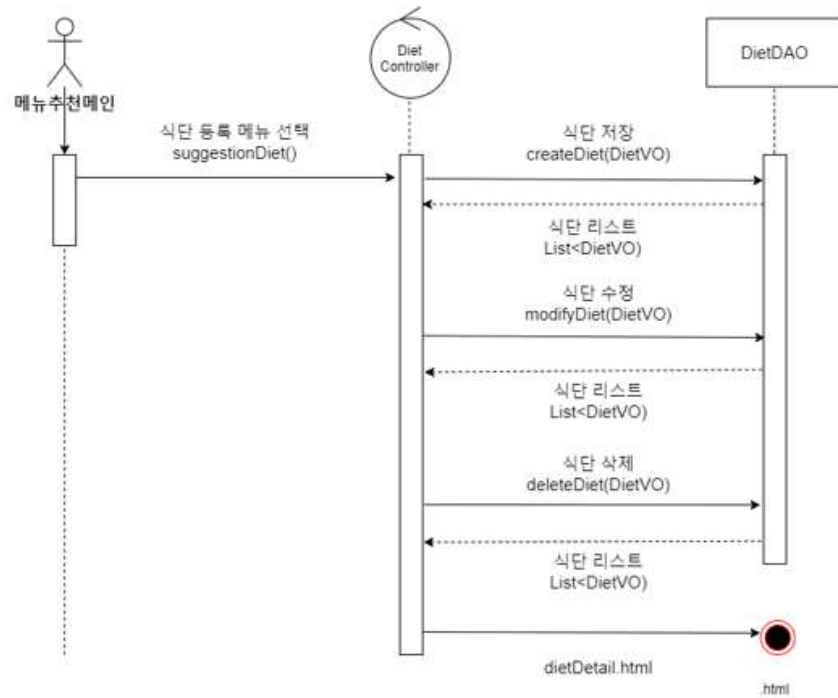


관리자 시퀀스 다이어그램

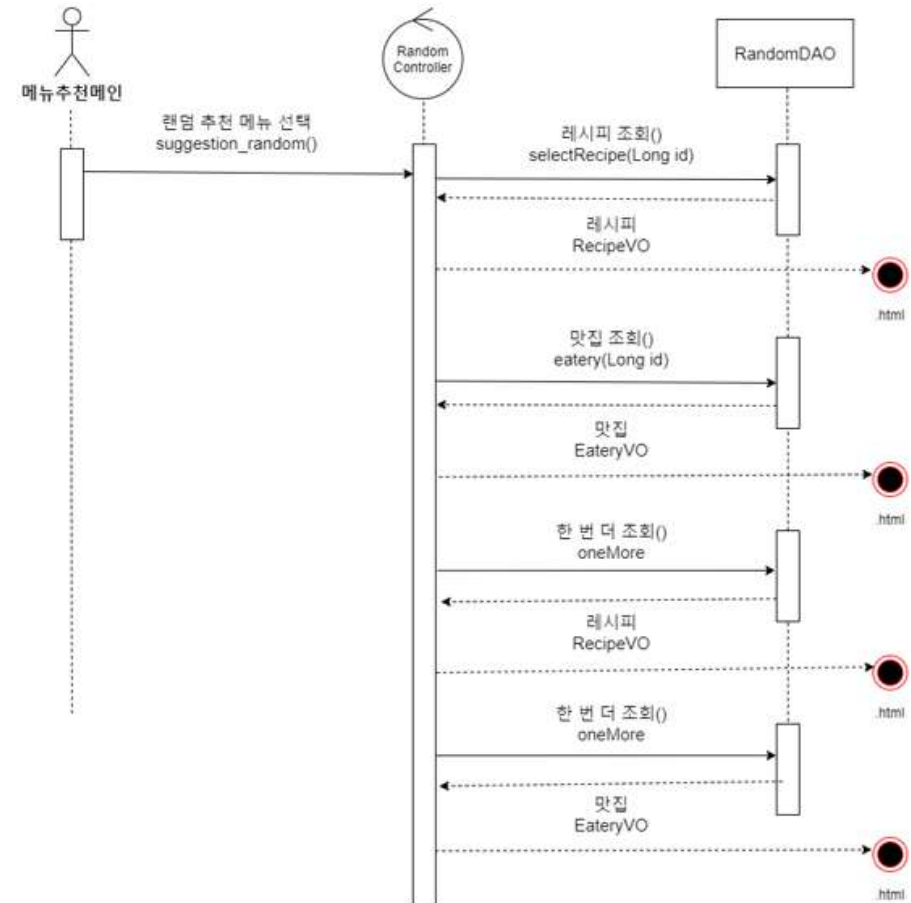




식단 등록



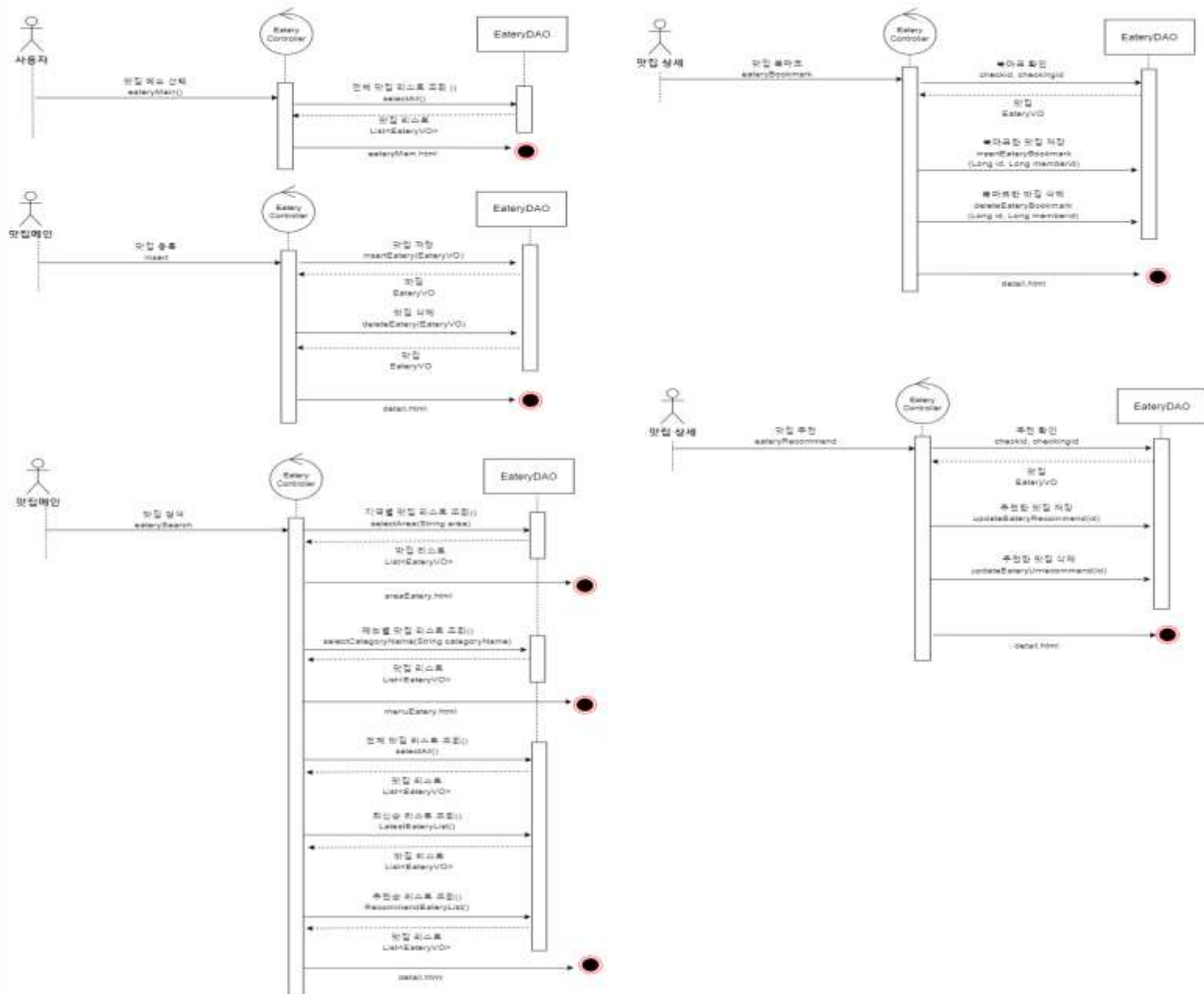
랜덤 추천



02 분석/설계

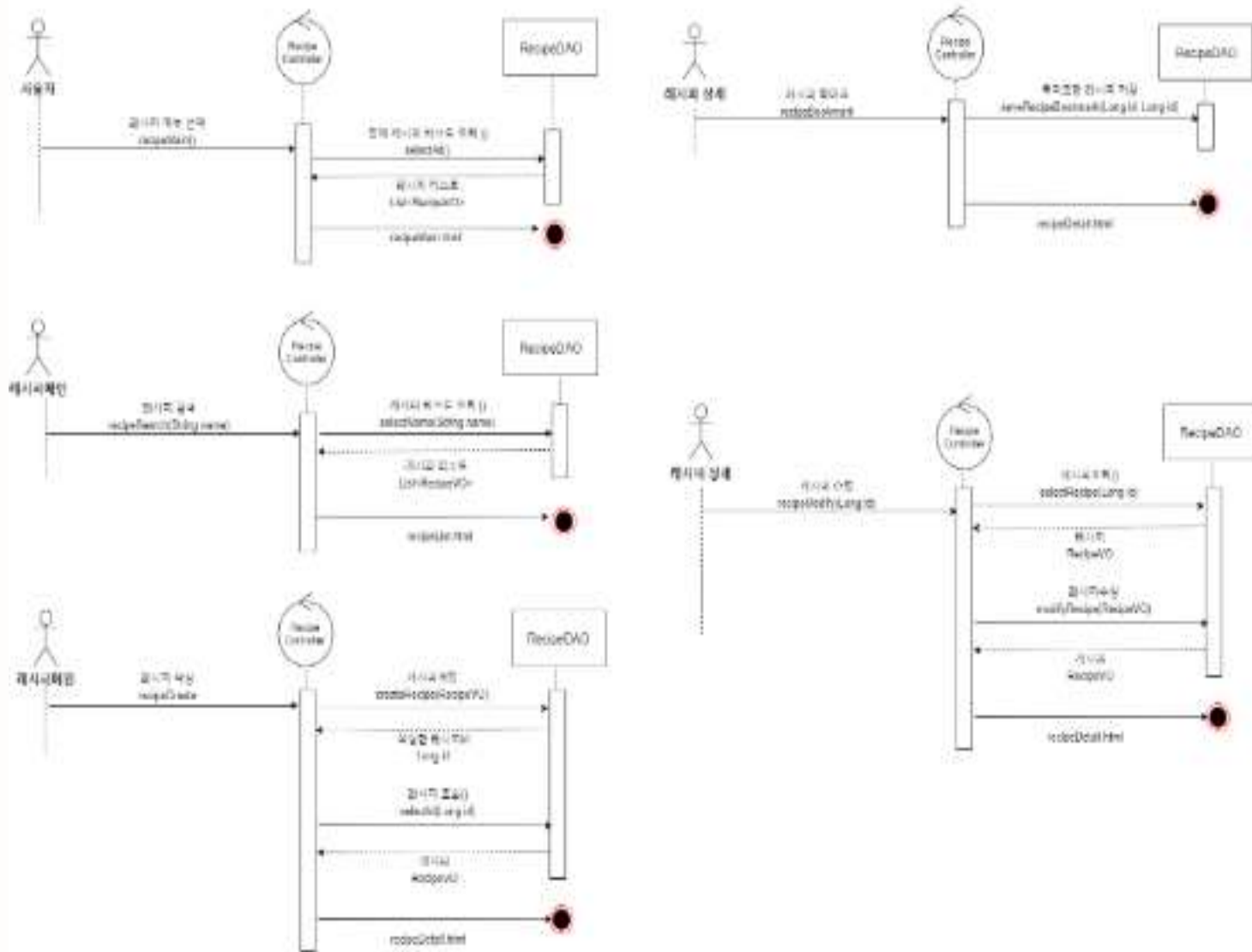
Sequence Diagram - Eatery

맛집 시퀀스 다이어그램





레시피 시퀀스 다이어그램





003

샘플 서비스 시연





004

기술소개





로그인

아이디

비밀번호

[아이디찾기](#) [비밀번호찾기](#) [회원가입](#)

© 뭐먹지?

로그인(login.html)에서 ID와 PW를 POST로 전송하여 Controller의 logincheck에서 확인
입력한 ID와 PW가 회원 테이블에 존재할 경우 HttpSession을 사용하여 회원 아이디와 로그인 체크 여부, 회원 정보를 저장
존재하지 않을 경우 로그인 페이지 출력

```
@GetMapping("/Login") // 로그인  
public void Login() {  
}  
  
@PostMapping("/Login") // 로그인  
public String Login(HttpSession session, String userId, String userPw, Long id) {  
    int check = memberservice.logincheck(userId, userPw);  
    if(check==1) {  
        session.setAttribute("loginCheck", true);  
        session.setAttribute("userId", userId);  
        session.setAttribute("member", memberservice.selectMemberId(userId));  
        return "redirect:/";  
    } else {  
        return "/Login";  
    }  
}
```

@Override

```
public int logincheck(String userId, String userPw) { // 로그인  
  
    return memberdao.logincheck(userId, userPw);  
}
```

```
int logincheck(@Param("userId") String userId, @Param("userPw") String userPw); // 로그인  
  
<!-- 로그인 -->  
<select id="logincheck" parameterType="String" resultType="int">  
    SELECT count(*) FROM member  
    WHERE userId = #{userId} and userPw =  
        #{userPw}  
</select>
```

04 기술소개

멤버



회원가입

이름 :

아이디 :

비밀번호 :

핸드폰 번호 :

© 뭐먹지?

회원가입(resister.html)에서 정보 입력
후 가입하기 버튼 클릭했을 경우
Controller의 insertMember에 전달

```
@GetMapping("/register")
public String register() { // 회원가입창띄우기
    return "register";
}

@PostMapping("insertMember")
public String insertMember(MemberVO membervo) { // 회원가입
    System.out.println(membervo);
    memberservice.insertMember(membervo);
    return "/Login";
}
```

`void insertMember(MemberVO membervo);` // .(생성)회원가입

```
@Override
public void insertMember(MemberVO membervo) {
    memberdao.insertMember(membervo);
}
```

```
<!-- 회원가입 -->
<insert id="insertMember" parameterType="com.eat.vo.MemberVO"
    useGeneratedKeys="true" keyProperty="id" keyColumn="id">
    insert into
    member(userId,userPw,userName,phoneNumber,grade)
    values(#{userId},#{userPw},#{userName},#{phoneNumber},0)
</insert>
```

```
<h3>회원가입</h3>

<form action = "insertMember" method="post">

    <div class=userName>
        <label for = "userName">이름 :</label>
        <input id = "userName" name = "userName" type="text">
    </div>
    <div class="userId">
        <label for = "userId">아이디 :</label>
        <input id = "userId" name = "userId" type="text">
    </div>
    <div class="userPw">
        <label for = "userPw">비밀번호 :</label>
        <input id = "userPw" name = "userPw" type="text">
    </div>
    <div class="phoneNumber">
        <label for = "phoneNumber">핸드폰 번호 :</label>
        <input id = "phoneNumber" name = "phoneNumber" type="text">
    </div>

    <div class="button">
        <input type = "submit" value="가입하기"><input type="reset" value="취소하기">
    </div>
</form>
```

04 기술소개

멤버

```
@GetMapping("adminEateryList") //관리자 음식점 게시판
public String adminEateryList(Model model){
    model.addAttribute("selectAll", eateryservice.selectAll());
    return "/adminEateryList";
}

@PostMapping("adminEateryDelete") //관리자 음식점 게시판 삭제
public String adminEateryDelete(@RequestParam("id") List<Long> id) {
    memberservice.adminEateryDelete(id);
    return "redirect:/adminEateryList";
}

void adminEateryDelete(@Param("id") List<Long> id); // 관리자 음식점 게시판 삭제

void adminEateryRecommendDelete(@Param("id") List<Long> id); // 관리자 음식점 게시판 삭제

void adminEateryBookmarkDelete(@Param("id") List<Long> id); // 관리자 음식점 게시판 삭제

void adminMemberRecommendDelete(@Param("id") List<Long> id); // 관리자 음식점 게시판 삭제

@Override
public void adminEateryDelete(List<Long> id) { // 관리자 음식점 게시판 삭제
    // TODO Auto-generated method stub
    memberdao.adminEateryBookmarkDelete(id);
    memberdao.adminEateryRecommendDelete(id);
    memberdao.adminMemberRecommendDelete(id);
    memberdao.adminEateryDelete(id);
}

<!-- 관리자 음식점 삭제 -->
<delete id="adminEateryRecommendDelete" parameterType="java.util.List">
    DELETE FROM eateryrecommend
    WHERE
    <foreach collection="id" item="id" open="" close=""
        separator="or">
        (eateryid = #{id})
    </foreach>
</delete>

<!-- 관리자 음식점 삭제 -->
<delete id="adminEateryBookmarkDelete" parameterType="java.util.List">
    DELETE FROM eaterybookmark
    WHERE
    <foreach collection="id" item="id" open="" close=""
        separator="or">
        (eateryid = #{id})
    </foreach>
</delete>

<!-- 관리자 음식점 삭제 -->
<delete id="adminMemberRecommendDelete" parameterType="java.util.List">
```

```
<h1>음식점 게시판</h1>

<div class="eatery">
    <form id=form action="adminEateryDelete" method="post">
        <table border="1">
            <thead>
                <tr>
                    <th scope="col"><input id="allCheck" type="checkbox" name="allCheck"></th>
                    <th>상호명</th>
                    <th>내용</th>
                    <th>주소</th>
                    <th>전화번호</th>
                    <th>카테고리</th>
                    <th>추천</th>
                </tr>
            </thead>

            <tr th:each="eatery, index : ${selectAll}">

                <!-- 체크박스 -->
                <td class="header" width="30">
                    <div class="chkclass">
                        <input name="id" type="checkbox" id="id" class="id" th:value="${eatery.id}">
                        .....
                    </div>
                </td>
                <td>
                    <input type="text" th:id="name+${index.count}" class="name" name="name" th:value="${eatery.name}" readonly>
                </td>
                <td>
                    <input type="text" th:id="content+${index.count}" class="content" name="content" th:value="${eatery.content}" readonly>
                </td>
                <td>
                    <input type="text" th:id="address+${index.count}" class="address" .....>
                </td>
                <td>
                    <input type="text" th:id="phoneNumber+${index.count}" class="phone" .....>
                </td>
                <td>
                    <input type="text" th:id="categoryName+${index.count}" class="cate" .....>
                </td>
                <td>
                    <input type="text" th:id="recommend+${index.count}" class="recomme" .....>
                </td>
            </tr>
        </table>
        <input type="button" value="삭제"/>
    </form>
</div>

<script>
    $(function() {
        $('#allCheck').click(function() {
            $('#id').prop('checked', this.checked);
        });
    });
</script>
```

음식점 게시판

번호	상호명	내용	주소	전화번호	카테고리	추천
1	음식점1	음식점1	주소1	00-123-1234	한식	4
2	음식점2	음식점2	주소2	00-123-1234	한식	3
3	음식점3	음식점3	주소3	00-123-1234	한식	10
4	음식점4	음식점4	주소4	00-123-1234	한식	10
5	음식점5	음식점5	주소5	00-123-1234	한식	8
6	음식점6	음식점6	주소6	00-123-1234	한식	3
7	음식점7	음식점7	주소7	00-123-1234	한식	27
8	음식점8	음식점8	주소8	00-123-1234	한식	4
9	음식점9	음식점9	주소9	00-123-1234	한식	10
10	음식점10	음식점10	주소10	00-123-1234	한식	10
11	음식점11	음식점11	주소11	00-123-1234	한식	10
12	음식점12	음식점12	주소12	00-123-1234	한식	10
13	음식점13	음식점13	주소13	00-123-1234	한식	10
14	음식점14	음식점14	주소14	00-123-1234	한식	10
15	음식점15	음식점15	주소15	00-123-1234	한식	10
16	음식점16	음식점16	주소16	00-123-1234	한식	10
17	음식점17	음식점17	주소17	00-123-1234	한식	10
18	음식점18	음식점18	주소18	00-123-1234	한식	10
19	음식점19	음식점19	주소19	00-123-1234	한식	10
20	음식점20	음식점20	주소20	00-123-1234	한식	10

음식점 테이블의 정보를 출력
체크박스 선택한 값(음식점 테이블의 ID)을 리스트로 전달 받아 쿼리를 보낼 때 foreach문으로 리스트를 순회하면서 ID와 일치하는 음식점 정보와 추천 정보를 삭제



마이페이지

[개인정보수정](#)

[회원탈퇴](#)

[즐거찾기](#)

© 뭐먹지?

마이페이지(개인정보수정, 탈퇴, 즐겨찾기)
비로그인인 경우 로그인 페이지로 이동
로그인한 유저만 사용 가능
즐거찾기 : 로그인한 사용자의 ID를 이용하여 정보를 출력

```
@GetMapping("update")
public void update(HttpSession session, Model model) {    // 수정(개인정보수정)
    String userId = (String) session.getAttribute("userId");
    model.addAttribute("selectOne", memberservice.selectOne(userId));
}
```

```
@PostMapping("update")
public void updateMember(MemberVO membervo) { // 수정(개인정보수정)
    System.out.println(membervo);
    memberservice.updateMember(membervo);
}
```

```
@Override
public void deleteMember(MemberVO membervo) { // 회원탈퇴
    memberdao.deleteMember(membervo);
}
```

```
@GetMapping("favorite") //즐거찾기 페이지
public String favorite(Model model, HttpSession session) {
    MemberVO member = (MemberVO) session.getAttribute("member");
    model.addAttribute("selectEateryBookmark", eateryBookmarkdao.selectEateryBookmark(member.getId()));

    model.addAttribute("selectRecipeBookmark", recipeBookmarkdao.selectRecipeBookmark(member.getId()));
    return "/favorite";
}
```


04 기술소개 식단

```

1<!DOCTYPE HTML>
2<html xmlns:th="http://www.thymeleaf.org"
3      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4      layout:decorator="~/fragments/layout">
5  <meta charset="UTF-8">
6  <title>SU</title>
7  <th:block layout:fragment="content">
8    <div class="container">
9      <script src="//code.jquery.com/jquery-3.3.1.min.js"></script>
10     <link rel="stylesheet" type="text/css" href="/css/dietlist.css">
11     <script type="text/javascript" src="/js/dietlist.js"></script>
12     <th:block layout:fragment="form">
13       <!-- 테이블 시작 -->
14       <table class="date">
15         <form action="/selectDietlist" method="Post">
16           <input form="date" type="date" id="date" name="date" val
17           <input form="date" type="date" id="date2" name="date2">
18           <button class="btn btn-outline-primary" type="submit">
19             </form>
20           <table id="list" border="1">
21             <thead>
22               <tr>
23                 <th>선택</th>
24                 <th>음식명</th>
25                 <th>날짜</th>
26                 <th>카테고리</th>
27                 <th>칼로리</th>
28                 <th>메뉴</th>
29               </tr>
30             </thead>
31             <tr>
32               <th>each<th>suggestion, index : ${showDietlist}">
33                 <td th:text="${index.count}"></td>
34                 <td th:text="${suggestion.date}" id="date"></td>
35                 <td th:text="${suggestion.meal : ${T(con.eat.vo.Meal).values}">
36                 <td th:text="${suggestion.categoryName}"></td>
37                 <td th:text="${suggestion.menu}"></td>
38               </tr>
39               <tr>
40                 <th>each<th>suggestion, index : ${selectDietlist}">
41                 <td th:text="${index.count}"></td>
42                 <td th:text="${suggestion.date}" id="date"></td>
43                 <td th:text="${suggestion.meal : ${T(con.eat.vo.Meal).values}">
44                 <td th:text="${suggestion.categoryName}"></td>
45                 <td th:text="${suggestion.menu}"></td>
46               </tr>
47             </tbody>
48             <!-- 테이블 종료 -->
49             <!-- 버튼들 -->
50             <button class="btn btn-outline-primary" type="button" id=
51             <button class="btn btn-outline-primary" type="button" id=

```

기본 구성을 최대한 UI 스토리보드에 맞추어 제작하였고 팝업이 아닌 모달창을 사용하여 페이지 내에서 식단 등록, 수정, 삭제 기능을 동작할 수 있도록 구현

식단 등록

날짜 연도-월-일

끼니 아침

카테고리 한식

메뉴

등록

초기화

닫기 창

등록

수정

삭제

```

1 $(document).ready(function() {
2     document.getElementById("insertDietOpen").onclick = function() {
3         document.getElementById("modalInsert").style.display = "block";
4     }
5     document.getElementById("insertModalClose").onclick = function() {
6         document.getElementById("modalInsert").style.display = "none";
7     }
8
9     document.getElementById("updateDietOpen").onclick = function() {
10        document.getElementById("modalUpdate").style.display = "block";
11    }
12    document.getElementById("updateModalClose").onclick = function() {
13        document.getElementById("modalUpdate").style.display = "none";
14    }
15
16    document.getElementById("deleteDietOpen").onclick = function() {
17        document.getElementById("modalDelete").style.display = "block";
18    }
19    document.getElementById("deleteModalClose").onclick = function() {
20        document.getElementById("modalDelete").style.display = "none";
21    }
22 });
23

```

04 기술소개

식단



```
<mapper namespace="com.eat.dao.DietDAO">
  <insert id="insertDiet" parameterType="com.eat.vo.DietVO">
    insert into suggestion(meal, menu, date, memberId, categoryId)
    select #{meal}, #{menu}, #{date}, #{memberId}, #{categoryId} from dual
    where 1 > (select count(*) from suggestion
              where date = #{date} and meal=#{meal})
  </insert>

  <select id="showDietList" resultType="com.eat.vo.DietVO" parameterType="Long">
    select s.*, c.name as "categoryName" from suggestion as s
    right join category as c on s.categoryId = c.id
    where c.id = s.categoryId and memberId = #{memberId} ORDER BY date
  </select>

  <select id="selectDietList" parameterType="com.eat.vo.DietVO" resultType="com.eat.vo.DietVO">
    select s.*, c.name as "categoryName" from suggestion as s
    right join category as c on s.categoryId = c.id
    where c.id = s.categoryId and date(date) between #{date} and #{date2} and memberId=#{memberId} ORDER BY date
  </select>

  <update id="updateDiet" parameterType="com.eat.vo.DietVO">
    UPDATE suggestion SET meal =#{meal}, menu =#{menu}, categoryId=#{categoryId}
    WHERE date = #{date} and meal=#{meal} and memberId = #{memberId}
  </update>

  <delete id="deleteDiet" parameterType="com.eat.vo.DietVO">
    DELETE FROM suggestion WHERE date =#{date} and meal=#{meal} and memberId = #{memberId}
  </delete>
</mapper>

@PostMapping("/suggestion/selectDietList")
public String selectDietList(@RequestParam("date") @DateTimeFormat(iso = DateTimeFormat.ISO.DATE, pattern="yyyy-MM-dd")LocalDate date,
                             @RequestParam("date2")@DateTimeFormat(iso = DateTimeFormat.ISO.DATE, pattern="yyyy-MM-dd")LocalDate date2,
                             Model model,DietVO dietVO,HttpSession session) {
    MemberVO member = (MemberVO) session.getAttribute("member");
    if(member!=null) {
        dietVO.setMemberId(member.getId());
    }
    model.addAttribute( "selectDietList" , dietDAO.selectDietList(date, date2,dietVO.getMemberId()));
    model.addAttribute( "categorylist" , categoryDAO.categoryList());
    return "/suggestion/dietList";
}
```

식단 리스트

연도-월-일 연도-월-일 조회

번호	날짜	끼니	카테고리	메뉴
1	2021-09-06	아침	한식	김치찌개
2	2021-09-06	점심	일식	초밥
3	2021-09-22	점심	일식	초밥
4	2021-09-22	아침	한식	고기
5	2021-09-23	점심	중식	짬뽕



식단 리스트

연도-월-일 연도-월-일 조회

번호	날짜	끼니	카테고리	메뉴
1	2021-09-22	점심	일식	초밥
2	2021-09-22	아침	한식	고기
3	2021-09-23	점심	중식	짬뽕

사용자가 선택한 날짜를 데이터 포맷 형식으로 받고 세션에서 로그인 한 ID를 사용하여 해당 사용자의 선택한 기간의 식단 리스트를 출력하고 등록, 수정, 삭제 기능을 구현

등록 수정 삭제

등록 수정 삭제

04 기술소개 랜덤추천

```
<!-- 음식점 테이블 시작 -->
<div class="randombox">
  <div class="randomEatery" onClick="location.href='#'">
    <h3>추천 음식점!</h3>
    <table border="2">
      <tbody th:each="eatery : ${showEatery}">
        <tr>
          <th>이미지</th>
          <td></td>
        </tr>
        <tr>
          <th>상호명</th>
          <td th:text="${eatery.name}"></td>
        </tr>
        <tr>
          <th>카테고리</th>
          <td th:text="${eatery.categoryName}"></td>
        </tr>
        <tr>
          <th>전화번호</th>
          <td th:text="${eatery.phoneNumber}"></td>
        </tr>
      </tbody>
    </table>
  </div>
<!-- 음식점 테이블 종료 -->
```

음식점과 레시피 테이블에 등록된 정보를
랜덤으로 불러와 출력
버튼을 클릭 시 등록된 다른 정보를
랜덤으로 재출력

```
27
28 @GetMapping("/suggestion/randomList")
29 public void showRandom(Model model, Model model2) {
30     int en = (int) (Math.random()*(eateryDAO.selectAll().toArray().length));
31     List<EateryVO> eateryVO = eateryDAO.selectAll();
32     model.addAttribute("showEatery", randomDAO.showEatery(eateryVO.get(en).getId()));
33
34     int rn = (int) (Math.random()*(recipeDAO.selectAll().toArray().length));
35     List<RecipeVO> recipeVO = recipeDAO.selectAll();
36     model.addAttribute("showRecipe", randomDAO.showRecipe(recipeVO.get(rn).getId()));
37 }

8 <select id="showEatery" resultType="com.eat.vo.EateryVO" parameterType="Long">
9     select e.name, e.phoneNumber, c.name as "categoryName" from eatery as e
10     right join category as c on e.categoryId = c.id
11     where c.id = e.categoryId and e.id = #{eateryId}
12 </select>

14 <select id="showRecipe" resultType="com.eat.vo.RecipeVO" parameterType="Long">
15     select r.name, r.ingredient, r.createDate, c.name as "categoryName" from recipe as r
16     inner join category as c on r.categoryId = c.id
17     where c.id = r.categoryId and r.id = #{recipeId}
18 </select>
```

랜덤 추천

추천 음식점!

이미지	
상호명	식당3
카테고리	일식
전화번호	3333333

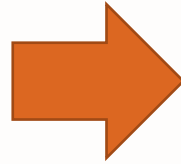
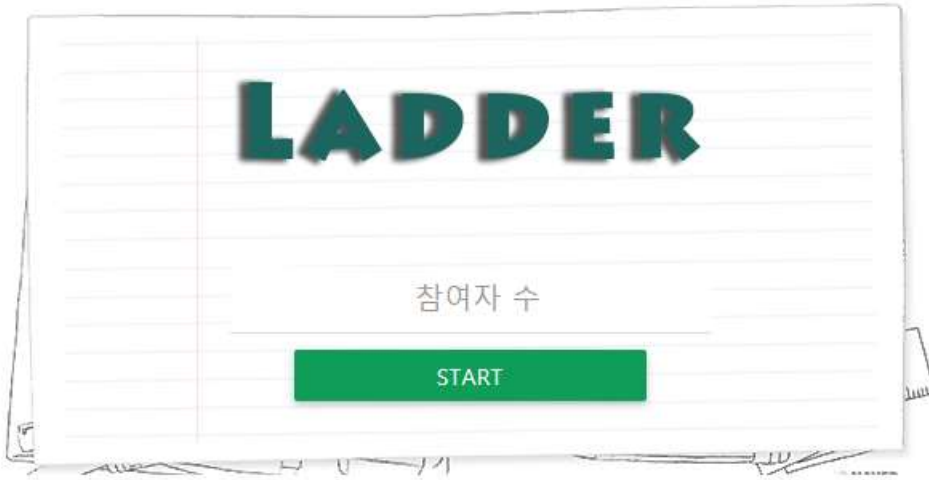
추천 레시피!

요리이름	4
카테고리	중식
재료	4

한 번 더!

04 기술소개

사다리 타기



사다리 기능은 API를 찾아 사용하였고
부트스트랩과 연동되면서 문제가 되는 부분을
CSS와 JS를 수정하여 상세 조정하였고,
사다리 가로줄을 숨겨서 재미와 랜덤 요소를
증가시키고 전반적인 디자인을 조의 컬러에 맞게 수정
사다리 페이지에서 버튼 클릭시 참여자 수 페이지로
돌아가 재시작

```
function canvasDraw(){
  ladder.css({
    'width' : ( widthNode-1 ) * 100 + 6,
    'height' : ( heightNode -1 ) * 25 + 6,
    'background-color' : '#fff'
  });
  ladder_canvas
  .attr('width' , ( widthNode-1 ) * 100 + 6)
  .attr('height' , ( heightNode-1 ) * 25 + 6);

  setDefaultFootPrint();
  reSetCheckFootPrint();
  setDefaultRowLine();
  setRandomNodeData();
  drawDefaultLine();
  //drawNodeLine(); //가로선
  userSetting();
  resultSetting();
}
```


04 기술소개

음식점-main

지역별 맛집

지역				
강북	강남	경북	경남	강원
인천	충청	대전	대구	울산
전라	광주	부산	경상	제주

메뉴별 맛집

메뉴		
한식	중식	일식
양식	분식	패스트푸드
베이커리	치킨	기타

신규 맛집

음식점 이름	카테고리	지역 추천수
유가네	한식	경남 6
올지다락	일식	경북 46
봉추집담	한식	인천 1
조스떡볶이	분식	대구 60
트끼정	한식	광주 3
고촌치킨	치킨	부산 30
피자스쿨	양식	강원 73
파리바게트	베이커리	울산 221
낙골새	한식	부산 23
맥도날드	패스트푸드	강원 66
장칼집	분식	강원 20
돼지국밥	한식	부산 11
바우글	한식	강남 2
로프	베이커리	강남 0

==선택== ▾

```
@GetMapping("/eateryMain")
public void eateryMain(Model model, String area) {
    model.addAttribute("selectAll", eateryservice.selectAll());
    model.addAttribute("selectArea", eateryservice.selectArea(area));
}

@PostMapping("/selectEateryList")
public String selectEateryList(Model model, String checking) {
    if (checking.equals("추천순")) {
        model.addAttribute("RecommendEateryList", eateryservice.RecommendEateryList());
        return "/eateryMain";
    } else if (checking.equals("최신순")) {
        model.addAttribute("LatestEateryList", eateryservice.LatestEateryList());
        return "/eateryMain";
    } else {
        return "/eateryMain";
    }
}
```

```
<select id="LatestEateryList" resultType="com.eat.vo.EateryVO">
    SELECT e.*, c.name as categoryName, r.recommend FROM eatery as e
    right join category as c
    on e.categoryId = c.id
    right join eateryrecommend as r
    on e.id = r.eateryid
    where c.id = e.categoryId ORDER BY id DESC
</select>
```

```
<select id="RecommendEateryList" resultType="com.eat.vo.EateryVO">
    SELECT e.*, c.name as categoryName, r.recommend FROM eatery as e
    right join category as c
    on e.categoryId = c.id
    right join eateryrecommend as r
    on e.id = r.eateryid
    where c.id = e.categoryId
    ORDER BY recommend DESC
</select>
```

```
$(document).ready(function() {
    $(td[name='name']).on("click", function() {
        var name = $(this).closest('td').attr('value') + "";
        $('#input[id=name]').val(name);
        $('#detail').attr("action", "detail");
        $('#detail').submit();
    });
});
```

```
<tr th:each="eatery : ${selectAll}">
    <td th:text="${eatery.name}" id="name" name="name"
        th:value="${eatery.name}"></td>
    <td th:text="${eatery.categoryName}"></td>
    <td th:each="area : ${T(com.eat.vo.Area).values()}"
        th:if="${#strings.toString(eatery.area)}==${#strings.toString(area.name())}"
        th:text="${area.getKrName}"></td>
    <td th:text="${eatery.recommend}"></td>
</tr>
<tr th:each="eatery : ${LatestEateryList}">
    <td th:text="${eatery.name}" name="name" th:value="${eatery.name}"></td>
    <td th:text="${eatery.categoryName}"></td>
    <td th:each="area : ${T(com.eat.vo.Area).values()}"
        th:if="${#strings.toString(eatery.area)}==${#strings.toString(area.name())}"
        th:text="${area.getKrName}"></td>
    <td th:text="${eatery.recommend}"></td>
</tr>
<tr th:each="eatery : ${RecommendEateryList}">
    <td th:text="${eatery.name}" name="name" th:value="${eatery.name}"></td>
    <td th:text="${eatery.categoryName}"></td>
    <td th:each="area : ${T(com.eat.vo.Area).values()}"
        th:if="${#strings.toString(eatery.area)}==${#strings.toString(area.name())}"
        th:text="${area.getKrName}"></td>
    <td th:text="${eatery.recommend}"></td>
</tr>
</table>

<form action="selectEateryList" method="Post" id="select">
    <select name="checking" form="select" onchange="this.form.submit()">
        <option>==선택==</option>
        <option value="추천순">추천순</option>
        <option value="최신순">최신순</option>
    </select>
```

지역이나 메뉴를 누르면 해당 페이지로 이동

신규맛집: 페이지 출력시 Controller에서 Eateryservice 호출하여 RecommendEateryList, LatestEateryList 정보를 리턴

음식점 이름을 누르면 음식점의 name을 받아 detail 화면으로 이동





04 기술소개

음식점-recommend, bookmark

상세 페이지

음식점 이름	카테고리	지역	주소	전화	음식점 설명
유가네	한식	경남	11212		

상세 페이지

음식점 이름	카테고리	지역	주소	전화	음식점 설명
올지다락	일식	경북		00	

추천

즐거찾기

detail 페이지는 로그인 상태일 경우
추천과 즐겨찾기 버튼 노출

HttpSession의 회원 ID를 전달 받아
EateryBookmark와
MemberRecommend 테이블의 정보
를 비교하여 버튼 상태 변경

```
@PostMapping("recommend")
public String Recommend(Model model, Long id, HttpSession session, String name) {
    MemberVO member = (MemberVO) session.getAttribute("member");
    EateryVO eateryvo = new EateryVO();
    eateryvo.setName(name);
    if(null!=member) {
        eateryservice.insertMemberRecommend(id, member.getId());
        eateryservice.updateEateryRecommend(id);
        eateryvo.setMemberId(member.getId());
        model.addAttribute("checkid", eateryservice.checkid(eateryvo));
        model.addAttribute("checkingid", eateryservice.checkingid(eateryvo));
    }
    model.addAttribute("selectId", eateryservice.selectId(id));
    return "/detail";
}

@PostMapping("unrecommend")
public String UnRecommend(Model model, Long id, HttpSession session, String name) {
    MemberVO member = (MemberVO) session.getAttribute("member");
    EateryVO eateryvo = new EateryVO();
    eateryvo.setName(name);
    if(null!=member) {
        eateryservice.deleteMemberRecommend(id, member.getId());
        eateryservice.updateEateryUnRecommend(id);
        eateryvo.setMemberId(member.getId());
        model.addAttribute("checkid", eateryservice.checkid(eateryvo));
        model.addAttribute("checkingid", eateryservice.checkingid(eateryvo));
    }
    model.addAttribute("selectName", eateryservice.selectName(name));
    return "/detail";
}

@PostMapping("bookmark")
public String Bookmark(Model model, Long id, HttpSession session, String name) {
    MemberVO member = (MemberVO) session.getAttribute("member");
    EateryVO eateryvo = new EateryVO();
    eateryvo.setName(name);
    if(!member.equals(null)) {
        eateryservice.insertEateryBookmark(id, member.getId());
        eateryvo.setMemberId(member.getId());
        model.addAttribute("checkid", eateryservice.checkid(eateryvo));
        model.addAttribute("checkingid", eateryservice.checkingid(eateryvo));
    }
    model.addAttribute("selectName", eateryservice.selectName(name));
    return "/detail";
}

@PostMapping("unbookmark")
public String UnBookmark(Model model, Long id, HttpSession session, String name) {
    MemberVO member = (MemberVO) session.getAttribute("member");
    EateryVO eateryvo = new EateryVO();
    eateryvo.setName(name);
    if(null!=member) {
        eateryservice.deleteEateryBookmark(id, member.getId());
        eateryvo.setMemberId(member.getId());
        model.addAttribute("checkid", eateryservice.checkid(eateryvo));
        model.addAttribute("checkingid", eateryservice.checkingid(eateryvo));
    }
    model.addAttribute("selectName", eateryservice.selectName(name));
    return "/detail";
}
```

```
<!-- 추천수 증가 -->
<update id="updateEateryRecommend" parameterType="Long">
    update eateryRecommend
    set recommend = recommend+1
    where eateryid = #{id}
</update>

<insert id="insertMemberRecommend" parameterType="Long">
    insert into memberRecommend(eateryid, memberid)
    select #{id}, #{memberId} from dual
    where 1>(select count(*) from memberRecommend
    where eateryid=#{id} and memberid=#{memberId})
</insert>

<!-- 추천 취소 -->
<update id="updateEateryUnRecommend" parameterType="Long">
    update eateryRecommend
    set recommend = recommend-1
    where eateryid = #{id}
</update>

<delete id="deleteMemberRecommend" parameterType="EateryVO">
    delete from memberRecommend
    where eateryid=#{id} and memberid=#{memberId}
</delete>

<!-- 즐겨찾기 추가 -->
<insert id="insertEateryBookmark" parameterType="Long">
    insert into eateryBookmark(eateryid, memberid)
    select #{id}, #{memberId} from dual
    where NOT EXISTS (select eateryid, memberid from eateryBookmark
    where memberid=#{memberId} and eateryId= #{id})
</insert>

<!-- 즐겨찾기 취소 -->
<delete id="deleteEateryBookmark" parameterType="EateryVO">
    delete from eateryBookmark
    where eateryid=#{id} and memberid=#{memberId}
</delete>

<select id="checkid" parameterType="EateryVO" resultType="int">
    select count(*) as count from memberRecommend as mr
    right join eatery as e
    on e.id = mr.eateryId
    where e.name=#{name} and mr.memberid=#{memberId}
</select>

<select id="checkingid" parameterType="EateryVO" resultType="int">
    select count(*) as count from eateryBookmark as eb
    right join eatery as e
    on e.id = eb.eateryId
    where e.name=#{name} and eb.memberid=#{memberId}
</select>
```

추천수 증가 : recommend+1
추천 취소 : 추천 상태인 경우 취소 버튼 노출
시켜주고 recommend-1

04 기술소개

음식점-recommend, bookmark

```
<form id="checkrecommend" action="recommend" method="post">
  <!-- (input) 평가 id 받는 부분 종료-->
  <input th:each="entry : ${selectNone}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectNone}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input name="memberid" th:value="${checkingid}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="memberid" th:value="${entry.memberid}" style="display: none" readonly>
  <input type="button" id="recommend" class="btn btn-secondary" value="추천">
</form>

<form id="checkunrecommend" action="unrecommend" method="post">
  <!-- (input) 평가 id 받는 부분 종료-->
  <input th:each="entry : ${selectNone}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectNone}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input name="memberid" th:value="${checkingid}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="memberid" th:value="${entry.memberid}" style="display: none" readonly>
  <input type="button" id="unrecommend" style="display: none" class="btn btn-secondary" value="추천">
</form>

<form id="checkbookmark" action="bookmark" method="post">
  <!-- (input) 평가 id 받는 부분 종료-->
  <input th:each="entry : ${selectNone}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectNone}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input name="memberid" th:value="${checkingid}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="memberid" th:value="${entry.memberid}" style="display: none" readonly>
  <input type="button" id="bookmark" class="btn btn-secondary" value="즐거찾기">
</form>

<form id="checkunbookmark" action="unbookmark" method="post">
  <!-- (input) 평가 id 받는 부분 종료-->
  <input th:each="entry : ${selectNone}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectNone}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input name="memberid" th:value="${checkingid}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="id" th:value="${entry.id}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="name" th:value="${entry.name}" style="display: none" readonly>
  <input th:each="entry : ${selectId}" name="memberid" th:value="${entry.memberid}" style="display: none" readonly>
  <input type="button" id="unbookmark" style="display: none" class="btn btn-secondary" value="즐거찾기">
</form>
```

```
$(document).ready(function() {
  var check = $("input[id='checkid']").val();
  var checking = $("input[id='checkingid']").val();
  //추천,즐거찾기
  if(check==3){
    $("input[id='recommend']").hide();
    $("input[id='unrecommend']").hide();
  } else if(check==1){
    $("input[id='recommend']").hide();
    $("input[id='unrecommend']").show();
  } else if(check==0){
    $("input[id='recommend']").show();
    $("input[id='unrecommend']").hide();
  }

  if(checking==3){
    $("input[id='bookmark']").hide();
    $("input[id='unbookmark']").hide();
  } else if(checking==1){
    $("input[id='bookmark']").hide();
    $("input[id='unbookmark']").show();
  } else if(checking==0){
    $("input[id='bookmark']").show();
    $("input[id='unbookmark']").hide();
  }
  //추천
  $("input[id='recommend']").on("click",function(){
    $("#checkrecommend").submit();
    $("input[id='recommend']").hide();
    $("input[id='unrecommend']").show();
  });

  $("input[id='unrecommend']").on("click",function(){
    $("#checkunrecommend").submit();
    $("input[id='recommend']").show();
    $("input[id='unrecommend']").hide();
  });
  //즐거찾기
  $("input[id='bookmark']").on("click",function(){
    $("#checkbookmark").submit();
    $("input[id='bookmark']").hide();
    $("input[id='unbookmark']").show();
  });

  $("input[id='unbookmark']").on("click",function(){
    $("#checkunbookmark").submit();
    $("input[id='bookmark']").show();
    $("input[id='unbookmark']").hide();
  });
});
```

js, jquery를 사용하여 버튼의 상태를 변경시키도록 작성

check==3: 아이디어가 없을 때 추천, 즐겨찾기가 hide

아이디어가 있을 경우
check==0: recommend show
bookmark show

check==1: unrecommend show
unbookmark show



04 기술소개

음식점-areaEatery

지역				
강북	강남	경북	경남	강원
연천	충청	대전	대구	울산
전라	광주	부산	경상	제주

지역별 인기 맛집

음식점 이름 카테고리 지역 추천수		
교촌치킨	치킨	부산 30
낙곱새	한식	부산 23
돼지국밥	한식	부산 11

지역별 맛집

음식점 이름 카테고리 지역 추천수		
돼지국밥	한식	부산 11
낙곱새	한식	부산 23
교촌치킨	치킨	부산 30

```

@GetMapping("/areaEatery")
public void areaEateryMain(Model model) {
    model.addAttribute("selectAll", eateryservice.selectAll());
    model.addAttribute("selectRecommend", eateryservice.selectRecommend());
}

@GetMapping("/selectAreaEatery")
public String areaEateryMain(Model model, @RequestParam("area") String area) {

    model.addAttribute("selectArea", eateryservice.selectArea(area));
    model.addAttribute("selectRecommendArea", eateryservice.selectAreaRecommend(area));
    return "/areaEatery";
}

@PostMapping("/areaEatery")
public String areaEatery(Model model, String area) {
    model.addAttribute("selectRecommendArea", eateryservice.selectAreaRecommend(area));
    model.addAttribute("selectArea", eateryservice.selectArea(area));
    return "/areaEatery";
}

```

```

<select id="selectArea" parameterType="String" resultType="com.eat.vo.EateryVO">
    SELECT e.*, c.name as categoryName, r.recommend FROM eatery as e
    right join category as c
    on e.categoryId = c.id
    right join eateryrecommend as r
    on e.id = r.eateryid
    WHERE area = #{area} order by id desc
</select>

```

```

<select id="selectRecommend" resultType="EateryVO">
    SELECT e.*, c.name as categoryName, r.recommend FROM eatery as e
    right join category as c
    on e.categoryId = c.id
    right join eateryrecommend as r
    on e.id = r.eateryid
    where c.id = e.categoryId order by recommend desc
</select>

```

```

<select id="selectAreaRecommend" parameterType="String" resultType="com.eat.vo.EateryVO">
    SELECT e.*, c.name as categoryName, r.recommend FROM eatery as e
    right join category as c
    on e.categoryId = c.id
    right join eateryrecommend as r
    on e.id = r.eateryid
    where area = #{area} order by recommend desc
</select>

```

Eateryservice의 selectAll, selectRecommend 메소드를 호출하고 areaEateryMain으로 전달되어 출력

다른 지역을 클릭할 때 경우 해당하는 인기맛집과 지역별 맛집이 출력



04 기술소개

레시피



```
<div class="img-container">
  <div id="download-img">
    
    <input style="..." name="file" type="file" id="input-image" th:field="*{thumb}"
      onchange="readImage(this)">
  </div>
</div>
```

Input으로 파일을 받아옴.
Input 값이 변경되면 readImage를 실행

```
function readImage(input) {
  if(input.files && input.files[0]) {
    // FileReader 인스턴스 생성
    const reader = new FileReader()
    // 이미지가 로드된 경우
    reader.onload = e => {
      const previewImage = document.getElementById("preview-image")
      previewImage.src = e.target.result
    }
    // reader가 이미지 읽도록 하기
    reader.readAsDataURL(input.files[0])
  }
}
```

This로 선택한 파일 정보를 받아온 뒤
Id="preview-image"의 값을
가져온 파일 정보로 바꿔준다.

생성한 파일정보를 reader에게 전달하여
화면에 뜨는 이미지를 변경.

04 기술소개

레시피



```
<form th:object="${recipeSearch}" class="form-inline">
  <div class="form-group mb-2">
    <input type="text" th:field="*${searchName}" class="form-control" placeholder="검색어" />
  </div>
  <div class="form-group mx-sm-1 mb-2">
    <select th:field="*${status}" class="form-control">
      <option value="">검색 종류</option>
      <option th:each="status : ${T(com.eat.vo.SearchStatus).values()}"
        th:value="${status}"
        th:text="${status}">option
      </option>
    </select>
  </div>
</form>
```

검색 종류를 구분하기 위한 RecipeSearch 클래스를 받아와 해당 값과 검색단어를 전달한다.

Select를 통해 검색 종류에 따른 쿼리문을 Service에서 보내 결과값을 리스트로 받아 온 뒤 화면에 띄운다.

```
@GetMapping(value="/recipeList")
public String recipeList(@ModelAttribute("recipeSearch")RecipeSearch recipeSearch, Model model, HttpSession httpSession){
    List<RecipeVO> recipeList = recipeService.selectStatus(recipeSearch.getStatus(), recipeSearch.getSearchName());
    model.addAttribute("recipeList", recipeList);
    Object member = httpSession.getAttribute("member");
    if(member == null){
        System.out.println("회원정보없음");
        return "/index";
    }

    return "/recipeList";
}
```

04 기술소개

레시피



```
@PostMapping(value="/recipe/new")
public String create(@Validated RecipeForm recipeForm, HttpSession httpSession){

    Object member = httpSession.getAttribute( name: "member");
    MemberVO memberVO = (MemberVO)member;

    RecipeVO recipe = new RecipeVO();
    recipe.setThumb("/img/" + recipeForm.getThumb());
    recipe.setName(recipeForm.getName());
    recipe.setIngredient(recipeForm.getIngredient());
    recipe.setPeople(recipeForm.getPeople());
    recipe.setCreateDate(LocalDateTime.now());
    recipe.setMemberId(memberVO.getId());

    Long recipeId = recipeService.saveRecipe(recipe);
    recipeService.manufactureTag(recipeId, recipeForm.getOriginTag());
    recipeService.tempSaveContent(recipeId, recipeForm.getContent());

    return "redirect:/recipe/" + recipeId + "/detail";
}
```

레시피 태그 입력칸에 있는 text를 String으로 받아
manufactureTag에서 #를 기준으로 분리 한 뒤
RecipeTag테이블에 insert 해준다.

```
@Override
public void manufactureTag(Long recipeId, String originTag) {
    String manufactTag = originTag.replace( target: " ", replacement: "");
    String tagList[] = manufactTag.split( regex: "#");

    int max = 11;

    if(!(tagList.length > max))
        max = tagList.length;

    for(int i=1; i<max; i++){
        RecipeTagVO tagVO = new RecipeTagVO();
        tagVO.setRecipeId(recipeId);
        tagVO.setName(tagList[i]);
        tagDAO.insertRecipeTag(tagVO);
    }
}
```

가져온 String의 공백을 모두 제거하고
#를 단위로 글자를 자른다.
(최대 10개이기때문에 max값을 10으로 고정)
List를 순회하며 받아온 RecipeId와 함께 등록한다.

04 기술소개

레시피



```
@GetMapping("/recipe/{recipeId}/detail")
public String detailRecipe(Model model, @PathVariable("recipeId") Long recipeId, HttpSession httpSession){
    Object member = httpSession.getAttribute(name: "member");
    MemberVO memberVO = (MemberVO)member;
    if(memberVO == null)
        return "redirect:/";

    RecipeVO recipeVO = recipeService.selectOne(recipeId);
    String tagList = recipeService.combineTag(recipeId);
    RecipeContentVO recipeContentVO = recipeService.selectContent(recipeId);
    String content = "";

    if(recipeContentVO != null)
        content = recipeContentVO.getContent();

    RecipeForm form = new RecipeForm();
    form.setThumb(recipeVO.getThumb());
    form.setName(recipeVO.getName());
    form.setIngredient(recipeVO.getIngredient());
    form.setPeople(recipeVO.getPeople());
    form.setRecommend(recipeVO.getRecommend());

    form.setOriginTag(tagList);
    form.setContent(content);

    model.addAttribute(attributeName: "form", form);

    if(recipeService.bookmarkCheck(recipeId, memberVO.getId())){
        //삭제상태
        model.addAttribute(attributeName: "bookmark", attributeValue: "북마크");
    }else {
        model.addAttribute(attributeName: "bookmark", attributeValue: "북마크취소");
    }

    return "recipeDetail";
}
```

```
@PostMapping("/recipe/{recipeId}/favorite")
public String bookMarkRecipe(@PathVariable("recipeId") Long recipeId, HttpSession httpSession){
    Object member = httpSession.getAttribute(name: "member");
    MemberVO memberVO = (MemberVO)member;
    if(recipeService.bookmarkCheck(recipeId, memberVO.getId())){
        recipeService.insertRecipeBookmark(recipeId, memberVO.getId());
    }else
        recipeService.deleteRecipeBookmark(recipeId, memberVO.getId());

    return "redirect:/recipe/" + recipeId + "/detail";
}
```

레시피 상세 – 기존에 등록되어 있는 레시피 정보를 로드
세션에서 멤버 정보를 받아
Bookmark테이블에 해당 레시피가 등록된 상태인지 체크 후
버튼에 들어갈 value값을 문자열로 전송

버튼을 누르면 등록 되어있는지 여부를 확인 후
Insert 혹은 delete를 실행



RECIPE NO. 005
RECIPE NO. 005

005

소감





은명진

- 저를 제외한 팀원 모두 매일 대면에 나오시고 다들 추석 연휴를 반납하고 코드작업을 하시는 등 열성적이게 작업 해 주셔서 저 또한 부족하게나마 프로젝트를 진행 할 수 있었던 것 같습니다. 모두 고생하셨습니다.



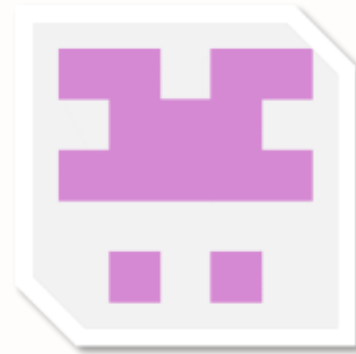
천상환

- 프로젝트를 진행함에 있어 많은 부분이 부족하였습니다. 조원들과 회의하고 검색을 통해 기능을 구현하고 맞추어 가면서 서로에게 많은 공부가 되었던 것 같습니다. 부족하지만 서로 의지하고 진행된 프로젝트 기간동안 즐거웠습니다.



최규

- 부족한 부분이 많은데 조원들의 도움을 통해 많은 걸 배울 수 있어서 좋았습니다.



김세은

- 이번 프로젝트를 통해 협동과 전반적인 구조를 이해하는데 큰 도움이 되었습니다.



김민주

- 서툴렀지만 조원분들이 열심히 도와주셔서 잘 끝낼 수 있었던 거 같습니다.



SQUARESPACE.COM/GOO
ICONS BY THE NOON PROJECT

감사합니다