

오늘의 요리®

SAUCE

팀장 : 양지훈

조원 : 한재원, 이희진, 유승모, 박형진, 유슬아

발표일 : 2015년 10월 02일 금요일

목 차

1. 개 요

1 - 1 프로젝트 개요

1 - 2 프로젝트 개발 환경

2. 팀 소개

2 - 1 팀 소개

2 - 2 팀원 소개

3. 개발 일정

3 - 1 팀 개발 일정

4. 프로그램 분석

4 - 1 페이지 분석표

4 - 2 페이지 상세 내역

4 - 1 프로그램 UML

5. DataBase 구조

5 - 1 테이블 기술서

5 - 2 물리 ERD

5 - 3 주요 쿼리

6. 주요 소스 설명

6 - 1 Controller Type 별 소스 설명

6 - 2 주요 소스 설명(사용자)

6 - 3 주요 소스 설명(관리자)

7. 프로젝트 후기

□ 프로젝트 개요

레시피를 보고, 자기만의 레시피를 올릴 수 있으며 보고싶은 쉘프 동영상을 볼 수 있다

□ 프로젝트 개발 환경

Java : mvc모델 중 컨트롤러 구현

Oracle : 프로그램과 연동되어 관리하는 데이터베이스 구축

JavaScript : 웹페이지 제공과 DB 연동 (Java와 Servlet연결)

Servlet : mvc모델 구현

Apache : 웹페이지 제공 및 사진 링크

그 외 기타 등등 (jQuery , Ajax , CSS , HTML5)

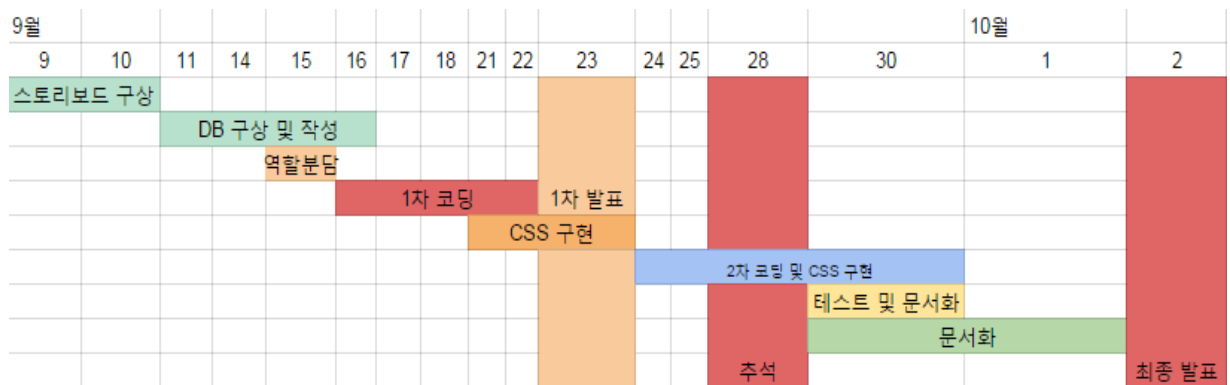


□ Team Name : SAUCE

□ 팀원 소개

팀 원	참여부분	기 타
양지훈	•총 괄 •메인, 로그인, 회원가입, CSS 및 디자인	조 장
유승모	•상세보기 •비디오	조 원
박형진	•관리자 페이지	조 원
류슬아	•공지게시판 •도움말 페이지	조 원
이희진	•DDL작성 •마이 페이지	조 원
한재원	•레시피 쓰기, 수정	조 원

❑ 프로젝트 개발 일정(팀)



❑ 프로젝트 개발 일정(개인)

구 분	양 지 훈	박 형 진	류 슬 아	유 승 모	이 희 진	한 재 원
1주차	아이디어 회의 및 주제선정, 시나리오 작성, DB / UI 설계					
2주차	메인, 로그인, 회원가입	관리자 페이지	공지게시판 도움말 페이지	상세보기 비디오	마이 페이지	레시피 쓰기, 수정
3주차	프로그램 기능 구현					

□ 페이지 분석표

구 분		상 세 설 명
회원가입	Java, Jsp, ajax	웹 서버에서 제공하는 회원가입 페이지가 요구하는 정보를 작성 시 Servlet, JSP와 연동된 웹 페이지와 DB의 회원 테이블에 저장된다.
로그인	Java, Jsp	ID와 Password를 입력 후 로그인 버튼 클릭시 DB의 정보와 일치 여부가 확인 되면 로그인을 승인하고 사이트 메인 화면으로 넘겨준다.
레시피 작성	Java, Jsp	로그인한 회원만 작성이 가능하다. 요리에 대한 메인 사진과 이름, 재료를 적는다. 레시피 순서를 적을 때 1개에서 20개까지 적을 수 있다. 작성하기 버튼을 누르면 DB의 레시피 테이블과 요리순서 테이블에 저장된다.
레시피 수정, 삭제	Java, Jsp	본인이 작성한 요리만 수정, 삭제가 가능하다.
레시피 보기	Java, Jsp, ajax	레시피의 내용은 누구나 볼 수 있다. 댓글 작성, 신고는 로그인을 해야 가능하고, 각각 댓글, 신고 DB에 삽입된다.
검색화면	Java, Jsp, ajax	종류별, 상황별로 레시피 검색이 가능하고, 원하는 키워드로 검색이 가능하다. 조회수, 최근날짜순 추천순으로 글을 볼 수 있다. 레시피 찜이 가능하고, 별도로 관리자의 이메일로 의견 제출을 할 수 있다.
쉐프 비디오	Java, Jsp	원하는 쉐프의 요리 영상을 볼 수 있다

구 분		상 세 설 명
마이페이지 이용	Java, Jsp, ajax	자신이 쓴 레시피와 댓글 짚한 레시피를 볼 수 있고 취소가 가능하다. 회원 정보수정과 회원탈퇴를 할 수 있다.
회원탈퇴	Java, Jsp	회원탈퇴시 작성한 레시피, 댓글이 삭제된다
관리자 기능	Java, Jsp, ajax	웹 페이지에서의 별도의 로그인에 필요하며 모든 회원에 대해서 회원 이메일 검색의 권한을 가지며 신고를 많이 받은 경우 해당 회원 을 탈퇴 시킬 수 있고 부적합한 레시피나 댓글의 경우도 삭제가 가능 하다. 공지사항을 작성 할 수 있다.

□ 인트로 페이지



: 로그인(회원가입, 비밀번호 찾기), 비회원으로 시작

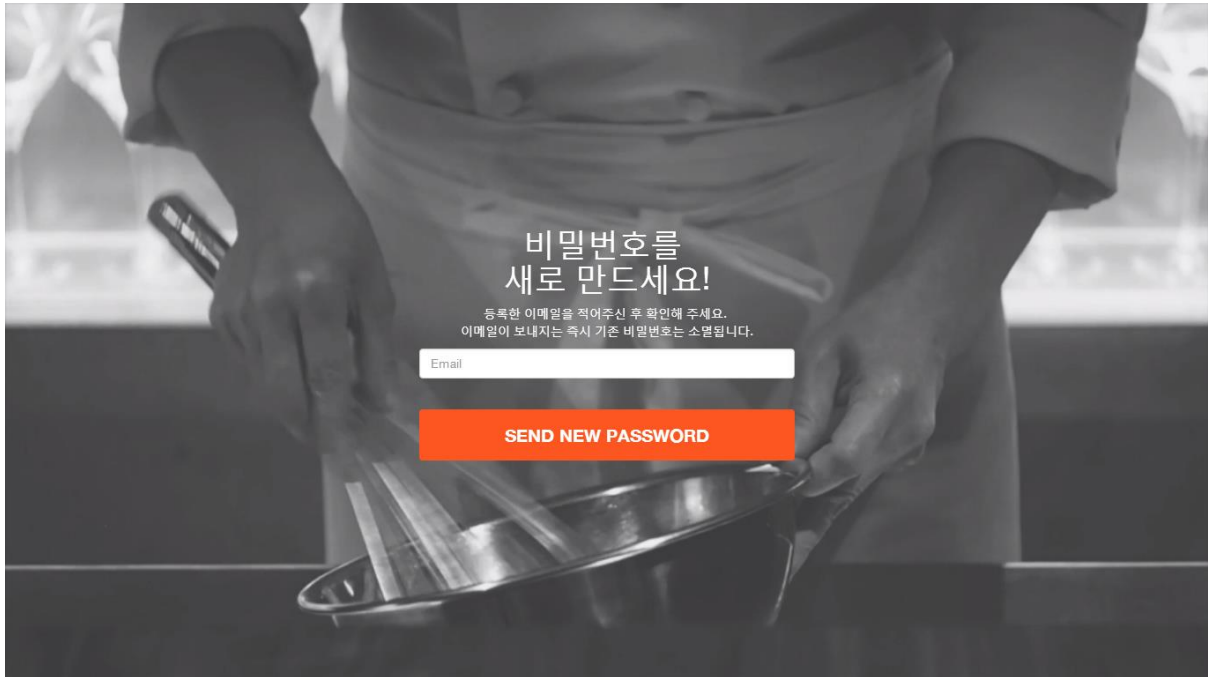
□ 회원가입 페이지

: 이메일 중복 확인 가능, 비밀번호 확인 가능, 프로필 사진 미리보기 가능

□ 로그인 페이지

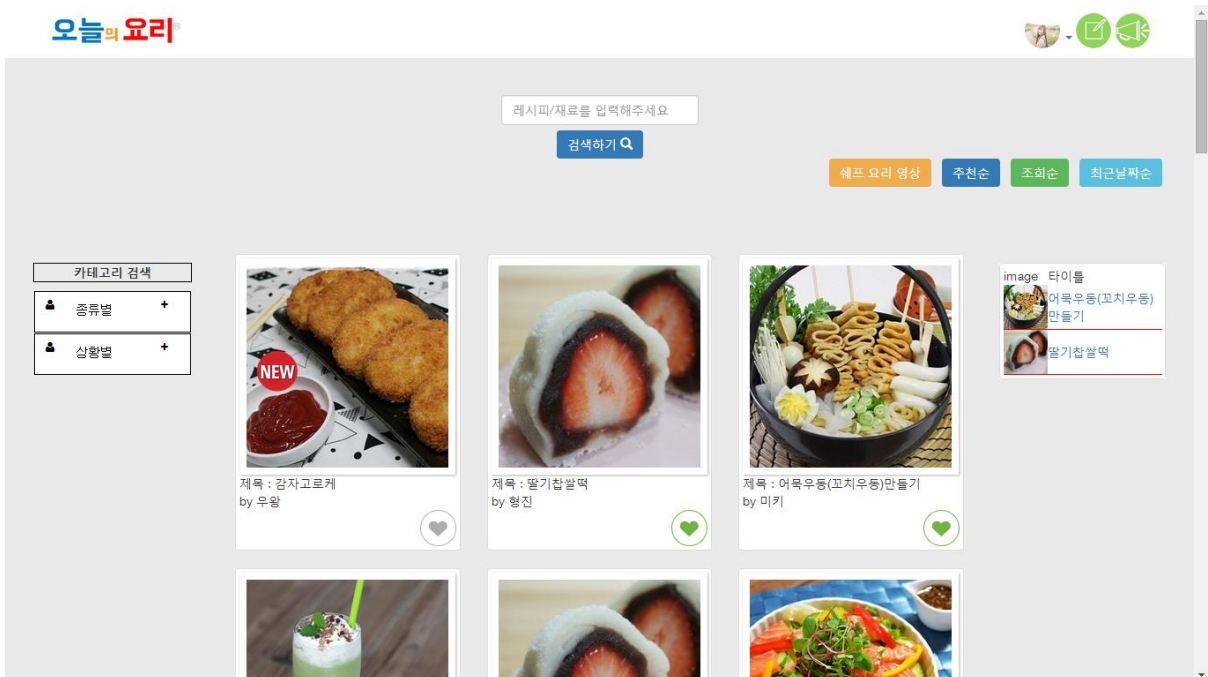
: 로그인 모달 창, 회원가입 가능, 비밀번호 찾기 가능

□ 비밀번호 찾기



: 비밀번호 새로 발급해서 이메일로 전송

□ 메인페이지 (레시피 뷰 페이지)

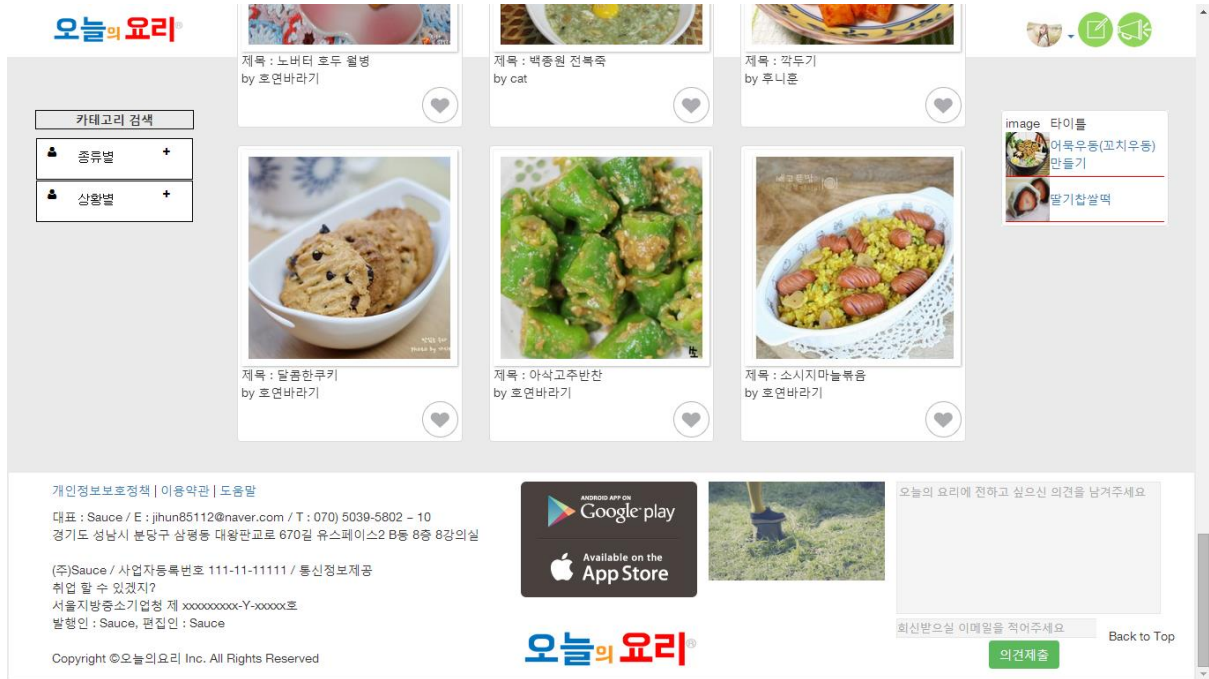


: 레시피 검색(종류별, 상황별, 제목 및 재료 별 검색, 찜, 마이페이지, 레시피

Chapter 4 프로그램 분석

쓰기, 공지사항 확인), 최근순, 조회순, 추천순으로 정렬 가능

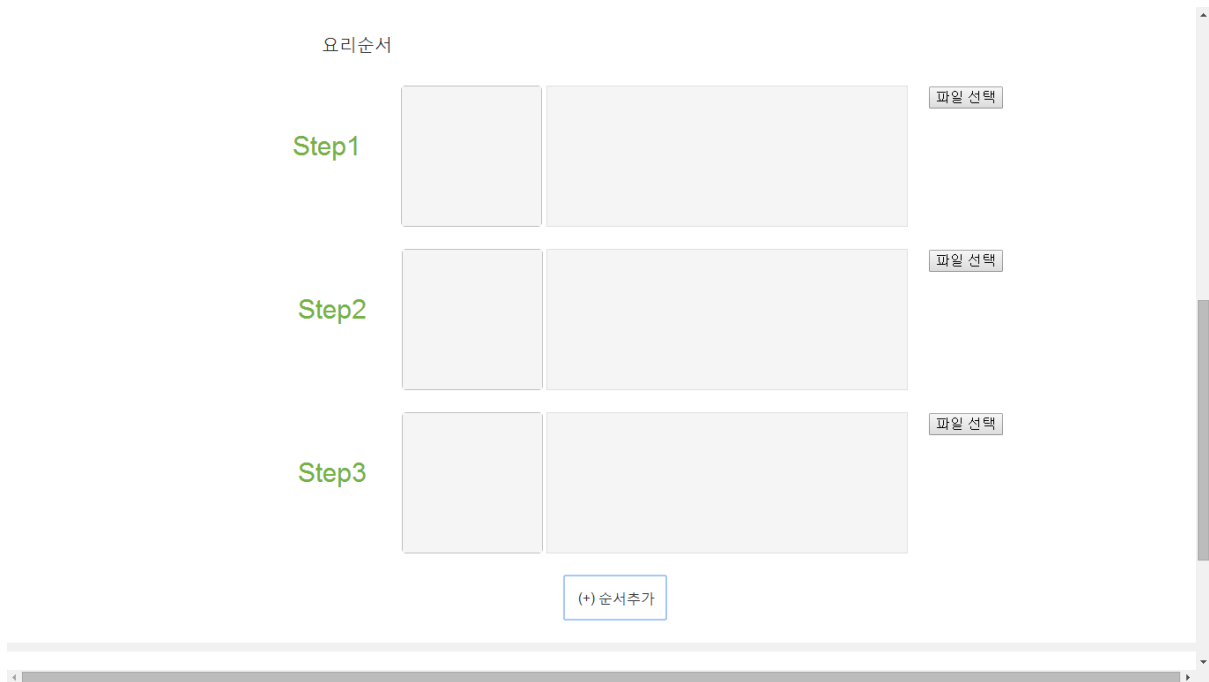
*Footer 고객 의견사항 전달



: 하단부의 footer에서 고객 의견 사항 전달 가능, 관리자에게 이메일 발송

□ 레시피 작성 페이지





: 요리순서 추가 가능, 요리 이미지 미리보기 가능

□ 레시피 상세 페이지

오늘의 요리



일상 · 밥/죽/떡

백종원표 마파두부

👤 인원 : 2인

🕒 조리시간 : 1시간이상

🌶️ 난이도 : 초급

💰 비용 : 9000

📅 등록일 : 2015-10-01 / 👤 조회수 : 1 / ❤️ 좋아요 : 0



레시피 : 3



[주재료]

양파 1개, 두부 한모, 돼지고기 갈은것 1컵, 전분가루 0.5T, 물 1/3컵, 간장 1/4컵, 고춧가루 3T, 된장 1T, 고추장 0.5T, 다진마늘 1T

- 1 마파두부에 꼭 빠져서는 안되는 양파! 한개를 다져주세요. 종이컵으로는 한컵정도 나오는 정도예요.



: 레시피 상세 페이지, 레시피 상세 내용 확인, 레시피 좋아요, 신고 가능, 댓글 작성 및 댓글 좋아요, 신고 가능

□ 마이페이지



: 회원정보 수정 가능, 찜한 레시피 확인 및 삭제, 작성한 레시피 수정 및 삭제, 레시피 댓글 확인 및 삭제, 최근순, 조회순, 좋아요 순으로 정렬 가능

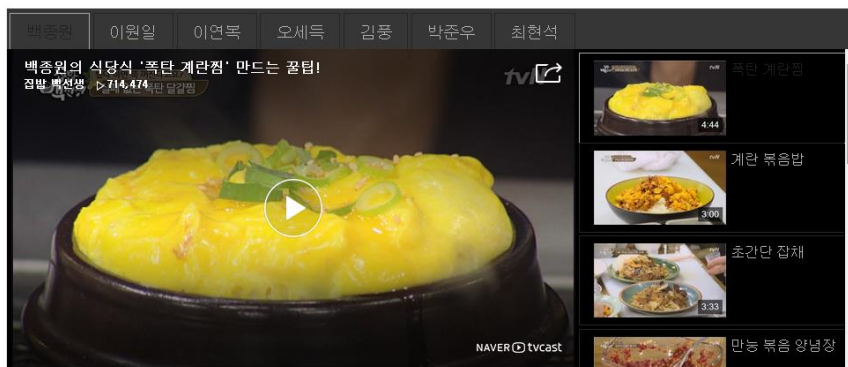
□ 공지사항 페이지



: 관리자가 작성한 공지사항 확인 가능

☐ 셰프 영상 페이지

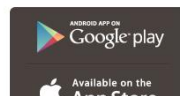
오늘의 요리



개인정보보호정책 | 이용약관 | 도움말

대표 : Sauce / E : jhun85112@naver.com / T : 070) 5039-5802 ~ 10
경기도 성남시 분당구 삼평동 대왕판교로 670길 유스페이스2 8동 8층 8강의실

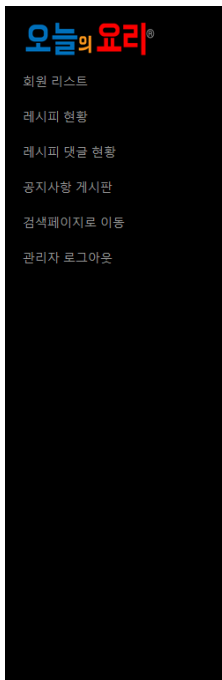
/즈(Sauce / 사업자등록번호 111-11-11111 / 투시정보제공



오늘의 요리에 전하고 싶으신 의견을 남겨주세요

: 셰프(백종원, 이원일, 이연복, 오세득, 김풍, 박준우, 최현석)의 요리영상 플레이어

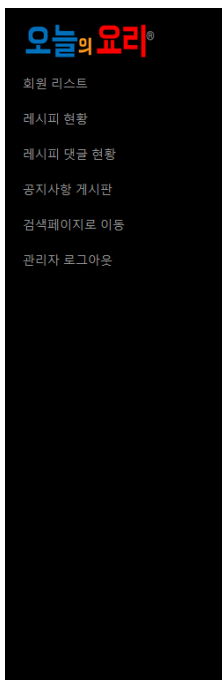
□ 관리자 페이지 - 회원리스트



MNUM	Email	password	닉네임	회원 삭제
23	rsa@aaa.com	1111	유슬아	회원강제탈퇴
1	jihun85112@empal.com	1234	후니훈	회원강제탈퇴
41	aa@aa.com	123	lala	회원강제탈퇴
4	admin@admin.com	1234	admin	
42	jihun85112@naver.com	1234	평권	회원강제탈퇴
61	test2@test.com	1234	등대	회원강제탈퇴
101	wodnjs1996@naver.com	1234	우왕	회원강제탈퇴
104	heejin@email.com	123	호연바라기	회원강제탈퇴
103	adf@aaa.com	1	gg	회원강제탈퇴
121	dmfl0dmfl@naver.com	123123	cat	회원강제탈퇴
122	tocook@aaa.com	1000	요리잉	회원강제탈퇴
123	babo@email.com	123	형진	회원강제탈퇴
124	dkdkdk@dkdkdk.com	555	미키	회원강제탈퇴

: 회원 목록 조회 및 회원 강제 탈퇴 가능

□ 관리자 페이지 - 레시피 현황



번호	제목	작성일자	작성자	작성자가이메일	추천수	댓글수	신고횟수	신고날짜
126	노버터 호두 월병	2015. 10. 1	호연바라기	heejin@email.com	0	0	0	
155	치즈스틱	2015. 10. 1	우왕	wodnjs1996@naver.com	0	0	0	
147	생연어샐러드	2015. 10. 1	미키	dkdkdk@dkdkdk.com	0	0	0	
128	LA참쌀파이 만들기	2015. 10. 1	요리잉	tocook@aaa.com	1	1	0	
129	백종원 - 양파 캐러멜 카레	2015. 10. 1	cat	dmfl0dmfl@naver.com	0	0	0	
130	국민반찬 엄마표 두부조림	2015. 10. 1	요리잉	tocook@aaa.com	1	1	0	
131	청포떡	2015. 10. 1	호연바라기	heejin@email.com	0	0	0	
124	깍두기	2015. 10. 1	후니훈	jihun85112@empal.com	0	0	0	
132	백선생 - 김치전	2015. 10. 1	cat	dmfl0dmfl@naver.com	0	0	0	
103	소시지마늘볶음	2015. 9. 25	호연바라기	heejin@email.com	0	0	0	
104	아삭고추반찬	2015. 9. 25	호연바라기	heejin@email.com	0	0	0	
133	백종원 김치볶음밥 레시피	2015. 10. 1	cat	dmfl0dmfl@naver.com	0	0	0	
134	갈비우거지 된장찌개	2015. 10. 1	요리잉	tocook@aaa.com	0	0	0	
125	백종원 전복죽	2015. 10. 1	cat	dmfl0dmfl@naver.com	0	0	0	
137	당면김말이	2015. 10. 1	형진	babo@email.com	0	0	0	
114	달콤한쿠키	2015. 9. 25	호연바라기	heejin@email.com	0	0	0	
138	비빔만두/찹만두	2015. 10. 1	요리잉	tocook@aaa.com	0	0	0	

: 레시피 현황 리스트 및 작성자로 검색하여 강제 삭제 가능

□ 관리자 페이지 - 댓글 현황

오늘의 요리® 회원 리스트 레시피 현황 레시피 댓글 현황 공지사항 게시판 검색페이지로 이동 관리자 로그인	댓글번호	댓글내용	댓글작성자	작성자번호	작성자닉네임	추천횟수	추천자번호	신고횟수	신고자번호	
	82	양파는 시른데!!	heejin@email.com	104	호연바라기	0	0	0	0	댓글삭제
	83	버섯 귀여웁	babo@email.com	123	형진	0	0	0	0	댓글삭제
	81	진짜 맛있겠다...	heejin@email.com	104	호연바라기	0	0	0	0	댓글삭제

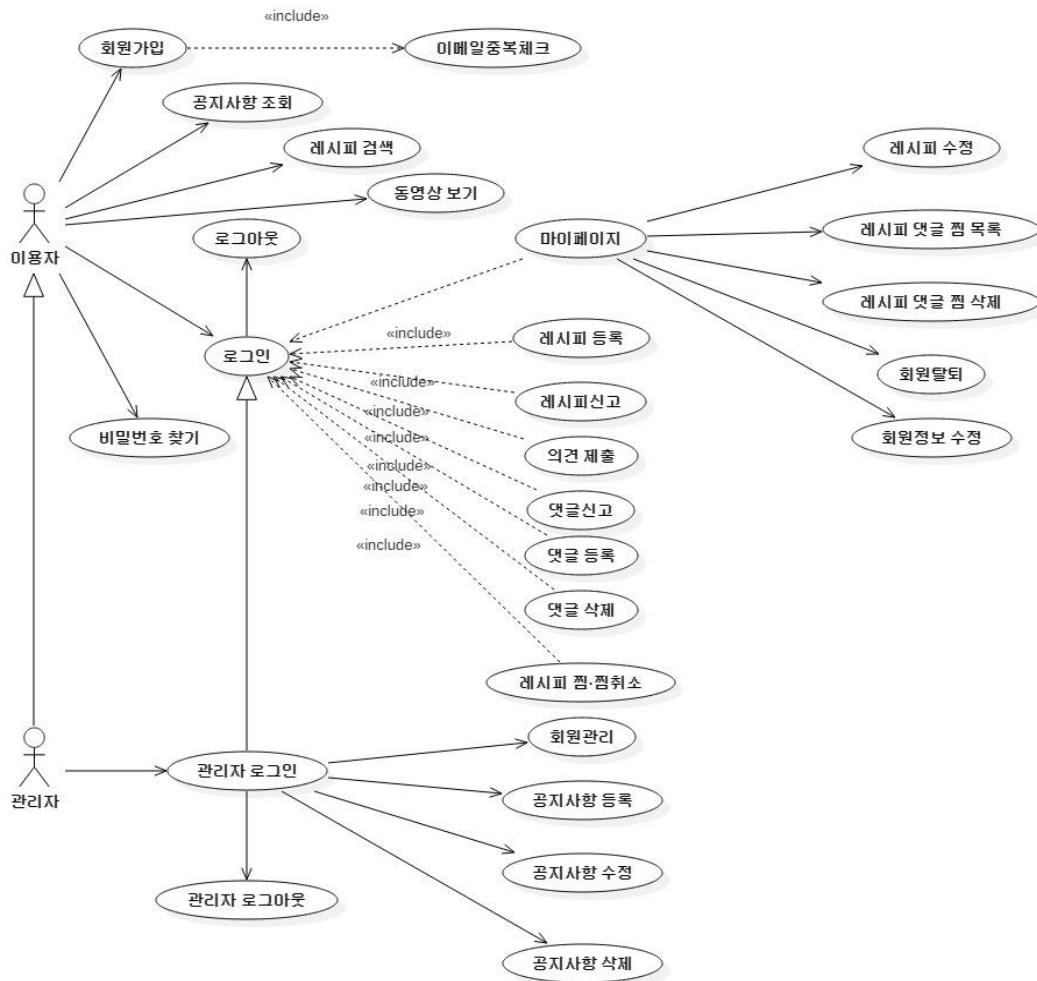
: 댓글 내용 및 리스트 확인, 댓글 강제 삭제 가능

□ 관리자전용 공지사항 페이지

오늘의 요리® 회원 리스트 레시피 현황 레시피 댓글 현황 공지사항 게시판 검색페이지로 이동 관리자 로그인	공지사항				게시글 등록
	번호	제목	작성자	작성일	
	26	공지]재원재원	admin	2015. 9. 30	
	25	공지공지	admin	2015. 9. 17	
	24	공지사항 테스트4	admin	2015. 9. 17	
	23	공지사항 테스트3	admin	2015. 9. 17	
	22	공지사항 테스트2	admin	2015. 9. 17	
	21	공지사항 테스트1	admin	2015. 9. 17	

: 관리자가 공지사항을 확인하고 작성 및 수정 가능

□ 프로그램 UML



□ 테이블 기술서

○ Member: 회원의 정보를 저장하는 테이블

No	속 성	타 입	N N	Ky	설 명
1	Mnum	Number	✓	PK	회원번호
2	Email	varchar(50)	✓		회원 이메일
3	Pwd	varchar(20)	✓		회원 비밀번호
4	Nick	varchar(20)	✓		회원 닉네임
5	Last	Date	✓		마지막접속시간
6	Profilpic	varchar(2000)	n u l l		프로필 사진

○ Recommend_recipe : 회원들이 할 수 있는 레시피 추천

No	속 성	타 입	N N	Ky	설 명
1	Mnum	Number	✓	FK	회원번호
2	Num	Number	✓	FK	레시피 번호
3	Rec_time	Date	✓		추천시간

○ Warning_recipe : 회원들이 할 수 있는 레시피 신고

No	속 성	타 입	N N	Ky	설 명
1	Mnum	Number	✓	FK	회원번호
2	Num	Number	✓	FK	레시피 번호
3	Warn_time	Date	✓		신고시간

Chapter 5 Database 분석

○ Recipe : 회원이 레시피 정보를 추가 및 저장 / 수정 / 삭제하는

테이블

No	속 성	타 입	N N	Ky	설 명
1	Num	Number	✓	PK	레시피 번호
2	Title	varchar2(60)	✓		제목
3	Cooktype	varchar2(50)	✓		요리종류
4	Situation	varchar2(50)	✓		요리상황
5	Material	varchar2(1000)	✓		재료
6	Pay	varchar2(8)			비용
7	Cooktime	varchar2(30)	✓		조리 시간
8	Hardly	varchar2(30)	✓		난이도
9	Person	varchar2(30)	✓		인원
10	Mnum	Number		FK	회원번호
11	Writetime	Date	✓		작성시간
12	Count	Number			조회수
13	Recommend	Number			추천수(좋아요)
14	Tip	varchar2(1000)			요리팁
15	Image	varchar2(2000)	✓		대표 이미지

○ CookStep : 회원들이 사용할 수 있는 요리순서

No	속 성	타 입	N N	Ky	설 명
1	Num	Number	✓	FK	레시피 번호
2	Step	Number(2)	✓	PK	요리순서
3	Image	varchar2(2000)	✓		이미지
4	Contents	varchar2(2000)	✓		설명

Chapter 5 Database 분석

○ QnA : 관리자가 사용할 수 있는 공지사항

No	속 성	타 입	N N	Ky	설 명
1	Num	Number	✓	PK	게시글 번호
2	Mnum	Number	✓	FK	회원번호
3	Subject	varchar2(50)	✓		제목
4	Reg_date	Date	✓		등록일자
5	Content	varchar2(2000)	✓		글내용

○ Recipe_comment : 회원들과 관리자가 사용할 수 있는 레시피 댓글

No	속 성	타 입	N N	Ky	설 명
1	Cnum	Number	✓	PK	댓글번호
2	Ccomment	varchar2(1000)	✓		내용
3	Cdate	Date	✓		날짜
4	Mnum	Number	✓	FK	회원번호
5	Crecommend	Number			추천수
6	Warning	varchar2(50)			신고
7	Num	Number	✓	FK	레시피 번호

○ Recommend_comment : 회원들이 사용할 수 있는 댓글추천

No	속 성	타 입	N N	Ky	설 명
1	Mnum	Number	✓	FK	회원 번호
2	Cnum	Number	✓	FK	댓글번호
3	Rec_time	Date			추천시간

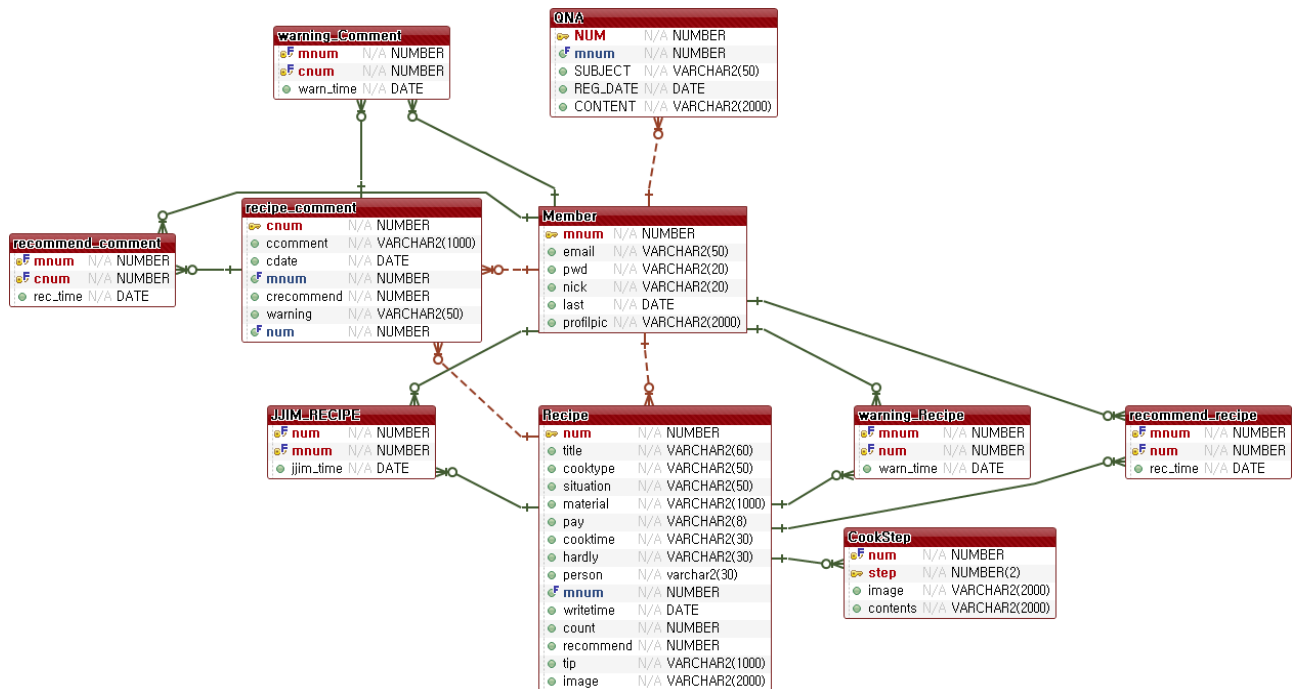
Chapter 5 Database 분석

○ Warning_comment : 회원들이 사용할 수 있는 댓글 신고

No	속 성	타 입	N N	Ky	설 명
1	Mnum	Number	✓	FK	회원 번호
2	Cnum	Number	✓	FK	댓글번호
3	Warn_time	Date	✓		신고시간

○ JJIM_Recipe : 회원들이 사용할 수 있는 찜

No	속 성	타 입	N N	Ky	설 명
1	Num	Number	✓	FK	레시피 번호
2	Mnum	Number	✓	FK	회원 번호
3	Jjim_time	Date	✓		찜한시간



□ 주요 쿼리

○ 회원 가입(중복 입력 방지)

```
INSERT INTO MEMBER VALUES(SEQ_MEMBER.NEXTVAL,?,?,?,SYSDATE,?)
```

○ 회원 아이디 중복 체크(중복 입력 방지)

```
SELECT * FROM MEMBER WHERE EMAIL=ID
```

○ 회원 닉네임 중복 체크(중복 입력 방지)

```
SELECT * FROM MEMBER WHERE EMAIL=?
```

○ 회원 정보 검색

```
SELECT MNUM, EMAIL,PWD,NICK FROM MEMBER
```

○ 회원 정보 수정

```
UPDATE MEMBER SET NICK = ? , EMAIL=? WHERE MNUM=?
```

○ 레시피 등록

```
INSERT INTO RECIPE VALUES(SEQ_RECIPE.NEXTVAL,?,?,?,?,?,?,?,SYSDATE,?,?,?,?,?)
```

```
SELECT MAX(NUM) FROM RECIPE
```

```
INSERT INTO COOKSTEP VALUES(?,?,?,?,?)
```

○ 레시피 수정

```
DELETE FROM COOKSTEP WHERE NUM=?
```

```
INSERT INTO COOKSTEP VALUES(?, ?, ?, ?)
```

○ 레시피 삭제

```
DELETE FROM RECIPE WHERE MNUM=? AND NUM=?
```

○ 찜 등록

```
INSERT INTO JJIM_RECIPE(NUM, MNUM, JJIM_TIME) VALUES(NUM,MNUM, SYSDATE)
```

○ 찜 삭제

```
DELETE FROM JJIM_RECIPE WHERE MNUM=MNUM AND NUM=NUM
```

○ 회원정보 검색

```
SELECT MNUM, EMAIL,PWD,NICK FROM MEMBER
```

○ 찜 리스트

```
SELECT R.TITLE TITLE, R.IMAGE IMAGE, R.MNUM MNUM, R.NUM NUM FROM RECIPE
```

```
R WHERE R.NUM IN (SELECT NUM FROM JJIM_RECIPE WHERE MNUM="+MNUM+")
```

○ 레시피 목록

```
SELECT NUM, NICK, MNUM, TITLE, WRITETIME, IMAGE, COOKTYPE, SITUATION,
```

```
COUNT, RECOMMEND FROM VIEW_MEMBER_RECIPE ORDER BY WRITETIME DESC
```

○ 레시피 검색(종류별)

```
"SELECT NUM, NICK, MNUM, TITLE, WRITETIME, IMAGE, COOKTYPE, SITUATION,  
COUNT, RECOMMEND FROM VIEW_MEMBER_RECIPE WHERE COOKTYPE LIKE  
'%"+COOKTYPE+"%' ORDER BY WRITETIME"
```

○ 레시피 검색(상황별)

```
"SELECT NUM, NICK, MNUM, TITLE, WRITETIME, IMAGE, COOKTYPE, SITUATION,  
COUNT, RECOMMEND FROM VIEW_MEMBER_RECIPE WHERE SITUATION LIKE  
'%"+SITUATION+"%' ORDER BY WRITETIME"
```

○ 레시피 검색(키워드별)

```
"SELECT NUM, NICK, MNUM, TITLE, WRITETIME, IMAGE, COOKTYPE, SITUATION,  
COUNT, RECOMMEND FROM VIEW_MEMBER_RECIPE WHERE MATERIAL LIKE  
'%"+SEARCH+"%' OR TITLE LIKE '%"+SEARCH+"%' ORDER BY WRITETIME"
```

○ 레시피 목록(최근 글)

```
SELECT NUM, NICK, MNUM, TITLE, WRITETIME, IMAGE, COOKTYPE, SITUATION,  
COUNT, RECOMMEND FROM VIEW_MEMBER_RECIPE ORDER BY WRITETIME DESC
```

○ 레시피 목록(조회수)

```
"SELECT NUM, NICK, MNUM, TITLE, WRITETIME, IMAGE, COOKTYPE, SITUATION,  
COUNT, RECOMMEND FROM VIEW_MEMBER_RECIPE ORDER BY COUNT DESC";
```

○ 찜 목록

```
"SELECT NUM FROM JJIM_RECIPE WHERE MNUM="+MNUM
```

○ 신고 목록(레시피)

```
"SELECT R.NUM NUM, R.TITLE TITLE, R.WRITETIME WRITETIME, M.NICK NICK
,M.EMAIL EMAIL, R.RECOMMEND, (SELECT COUNT(WR.NUM) FROM WARNING_RECIPE
WR WHERE R.NUM=WR.NUM) WCOUNT, (SELECT WARN_TIME FROM WARNING_RECIPE
WR WHERE R.NUM = WR.NUM) WARNINGTIME, (SELECT COUNT(RC.NUM) FROM
RECIPE_COMMENT RC WHERE R.NUM=RC.NUM) RECOUNT FROM RECIPE R, MEMBER M
WHERE R.MNUM = M.MNUM ORDER BY WCOUNT DESC"
```

○ 신고 목록(사용자)

```
"SELECT R.NUM NUM, R.TITLE TITLE, R.WRITETIME WRITETIME, M.NICK NICK ,M.EMAIL
EMAIL, (SELECT COUNT(WR.NUM) FROM WARNING_RECIPE WR WHERE R.NUM=WR.NUM)
WCOUNT, (SELECT WARN_TIME FROM WARNING_RECIPE WR WHERE R.NUM = WR.NUM)
WARNINGTIME FROM RECIPE R, MEMBER M WHERE R.MNUM = M.MNUM AND M.NICK LIKE
'%" + NICK + "%' ORDER BY WCOUNT DESC"
```

○ 신고 목록(사용자)

```
"DELETE FROM WARNING_RECIPE WHERE NUM="+NUM
"DELETE FROM RECOMMEND_RECIPE WHERE NUM="+NUM
"DELETE FROM RECIPE WHERE NUM=?"
```

○ 레시피 내용 보기

```
"SELECT * FROM RECIPE WHERE NUM = "+NUM
```

○ 요리순서 보기

```
"SELECT * FROM COOKSTEP WHERE NUM = "+NUM
```


□ 오늘의 요리에서 사용한 Controller Type

1. Controller - Command Type : 회원가입, 레시피(등록, 수정, 검색, 신고), 레시피 댓글(등록, 신고)

```
if(com.equals("datail.recipe")){//레시피 상세 페이지 진입

    recipecommand = new RecipeCountCommand();
    recipecommand.recipe(request, response);
    //조회수 1 증가

    recipecommand = new RecipeViewCommand();
    recipecommand.recipe(request, response);
    //상세 페이지 내용
    view = "Recipe/RecipeView.jsp";

    System.out.println(view);

}
```

2. Action Type : 공지사항

```
String command = request.getParameter("command");
System.out.println("BoardServlet 진입 : " + command);
//웹브라우저에서는 아무것도 나타나지 않지만 콘솔창에서는 확인
//할수있음
//ActionFactory 객체 생성한 후 BoardServlet 을 요청하면서
//전달된 command 요청 파라미터 값을 ActionFactory 객체의
//getAction()에 전달
//getAction()메소드가 리턴해준 action 객체로 execute()메소드를
//호출 함
ActionFactory af=ActionFactory.getInstance();
Action action=af.getAction(command);
//서블릿이 받은 커맨드 값을 콘솔창에 출력하여 커맨드 값을 제대로
//받았는지 확인 함
if(action != null){
    //getAction()은 커맨드에 맞는 Action 객체를 반환하지만
    //null 값을 갖는 액션을 반환하면 예외가 발생할 수 있기때문에 null 이
    //아닐경우에만 execute()호출 함
    action.execute(request, response);
}
```

```
//getAction()가 BoardServlet 의 doGet()에서 호출됨
//getAction()에서는 command 를 받아서 거기에 맞는 액션을 리턴
public Action getAction(String command) {
    Action action = null;
    System.out.println("ActionFactory : " + command);
    if (command.equals("board_list")) {
        action = new BoardListAction();
    } else if (command.equals("board_write_form")) {
        //board_list 에서 board_write_form 으로 이동
        action = new BoardWriteFormAction();
    } else if (command.equals("board_write")) {
        action = new BoardWriteAction();
    } else if (command.equals("board_view")) {
        action = new BoardViewAction();
    } else if (command.equals("board_update_form")) {
        action = new BoardUpdateFormAction();
    } else if (command.equals("board_update")) {
        action = new BoardUpdateAction();
    } else if (command.equals("board_delete")) {
        action = new BoardDeleteAction();
    } else if (command.equals("reply_write")) {
        action = new ReplyWriteAction();
    }
    return action;
}
```

3. Properties Type : 마이페이지(레시피 수정, 삭제, 댓글 삭제), 찜 삭제, 회원 정보 수정

```
#writeForm.do=edu.kosta.boardAction.writeFormAction
/mypage_index.do=com.kosta99.recipe.action.MyPageIndexAction
/mypage_jjim.do=com.kosta99.recipe.action.MyPageJjimAction
/mypage_recipe.do=com.kosta99.recipe.action.MyPageRecipeAction
/mypage_recipe_comment.do=com.kosta99.recipe.action.MyPageRecipeCommentAction
/mypage_updatePwd.do=com.kosta99.recipe.action.MyPageUpdatePwdAction
/mypage_updateInfo.do=com.kosta99.recipe.action.MyPageUpdateInfoAction
/mypage_updateForm.do=com.kosta99.recipe.action.MyPageUpdateFormAction
/test.do=com.kosta99.recipe.action.MyPageIndexAction
/member_out.do=com.kosta99.recipe.action.MemberOutAction
/del_my_recipe.do=com.kosta99.recipe.action.RecipeDeleteAction
/cancle_jjim.do=com.kosta99.recipe.action.JjimCancleAction
```

Chapter 6 주요 소스 분석

```
/del_my_comment.do=com.kosta99.recipe.action.CommentDeleteAction
```

```
public void init(ServletConfig config) throws ServletException {
    String props = config.getInitParameter("dirPath");
    System.out.println("dirPath : " + config.getInitParameter("dirPath"));
    Properties pr = new Properties();
    FileInputStream f = null;

    try {
        // command.properties 파일을 연다
        f = new FileInputStream(new File(props));
        // command.properties 를 로드하여 적재한다
        pr.load(f);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if(f != null) try {f.close();} catch (Exception e) {e.printStackTrace();}
    } // try-catch-finally

    Iterator keyIter = pr.keySet().iterator();
    //hashmap 에 저장된 key 값을 불러온다

    while(keyIter.hasNext()) {
        String command = (String)keyIter.next();
        String commandName = pr.getProperty(command);

        try {
            Class commandClass =
                Class.forName(commandName);
            //value 값으로 클래스화
            Object commandInstance =
                commandClass.newInstance();
            //그 클래스의 인스턴스 생성
            commandMap.put(command, commandInstance);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
} // init

try {
    String command = request.getRequestURI();
    System.out.println("command : " + command);
}
```

```
//command :/*.do

if(command.indexOf(request.getContextPath()) ==
0){
    //들어온 url 이 contextPath 경로로 시작한다면
    command =
command.substring(request.getContextPath().length());
    System.out.println("command : " +
command); // command : /*.do
} // if end

cp = (CommandAction)commandMap.get(command);

System.out.println("cp : " + cp);
// cp : command 패턴이 적용된 Service 이름의
클래스

if(cp == null) {
    view = "/index2.jsp"; //command Action이
RequestDispatcher dp =
request.getRequestDispatcher(view);
dp.forward(request, response);
    return;
} else {
    view = cp.process(request, response);
}
```

□ 주요 소스 안내

- 페이지의 실제 경로를 알아내는 소스

```
String requestURI = request.getRequestURI(); //서블릿으로
요청된 url
String contextPath = request.getContextPath()+"/"; //실제 전체
경로
String com = requestURI.substring(contextPath.length());

//요청된 url에서 root까지의 경로를 잘라낸다
```

- 절대경로를 표시하는 EL 문

```
${pageContext.request.contextPath} //root까지의 절대 경로
```

- Properties

```
@WebServlet(name="TestController", urlPatterns={"*.do"},
initParams={
@WebInitParam(name="dirPath",
Value = "E:/PROGRAMING/workspace/TodayCook_Type1/WebContent/WEB-INF/command.properties")) //Properties 실제 경로
```

//initParams : key와 value값으로 이뤄져 있어서, 값을 두 번 써준다??

□ 찜

```
function likeClick(i, n){
//좋아요 버튼 클릭 시, 해당 레시피의 버튼을 변수도 불러온다.
    var likeck = i;
// i : 해당 누른 버튼(반복된 for 문에서 특정위치의 해당버튼에 해당한다.
    일종의 this)
    var listnum = n; // n : 해당 버튼이 눌린 레시피의 번호
    if(likeck.src.match("btn_like.gif")){
//만약 좋아요의 버튼 링크가 btn_like.git 와 일치한다면

        $.ajax({ //ajax 실행
            type: "POST", //타입은 post
            url: "main/search_jjim.jsp", //찜 DB 접속 페이지
            data : {"num" : listnum},
            //parameter name: num, value: listnum(레시피 번호)
            success: function(data) {
                //해당 레시피번호를 찜리스트에 담는다.
                if(jQuery.trim(data)=='YES'){
                    //해당 레시피번호가 중복되지 않고 잘 담겼다면
                    alert("찜 했습니다.");

                    likeck.src="..${pageContext.request.contextPath}/img/number/b
tn_like_on.gif";
                    //src 를 btn_like_on 으로 변경하여 좋아요했다는 표시로 변경한다.

                }
            }
        });
    }
}
```

```
        else{
            //만약 해당 레시피번호가 중복되어 담기지않았다면
            alert("경고! 이미 등록된
                찜입니다!");

            likeck.src="..${pageContext.request.contextPath}/img/number/b
            tn_like.gif";

            //해당 버튼을 btn_like 로 변경한다.
        }//else

        $.ajax({
            //위의 ajax 가 실행되고 나면 이 ajax 를
                실행한다.
            type: "POST", //타입은 post
            url: "jjim.jjim",
            //servlet jjim.jjim 을 실행, 해당 페이지로 jjim 리스트를 발송한다.
            dataType: 'html',
            //데이터 타입은 html
            success : function(html) {
                //해당페이지로 jjim 리스트가 발송되면

                jQuery('.jjimlist').html(jQuery.trim(html));
                //jjimlist div 에 html 을 불러온다.
            }
        })//ajax

    }//success

    });//ajax
} else if(likeck.src.match("btn_like_on.gif")){
    //만약 눌러진 버튼이 btn_like_on 이라면,
    $.ajax({ //ajax 를 실행한다.
        type: "POST", //타입은 post
        url: "main/search_jjim_cancle.jsp",
        //찜 취소페이지
        data : {"num" : listnum},
        // parameter name: num, value: listnum(해당 레시피 번호)
        success: function(data) {
            //해당 레시피 번호에 해당하는 찜 취소
            if(jQuery.trim(data)=='YES'){
                //찜이 취소되면

                alert("찜취소 했습니다.");

                likeck.src="..${pageContext.request.contextPath}/img/number/b
                tn_like.gif";
```

```

        //해당 버튼을 btn_like 로 변경한다.
    }//if

    $.ajax({
        //위의 ajax 가 실행되고 나면 이 ajax 를 실행한다.
        type: "POST", //타입은 post
        url: "jjim.jjim",
        //servlet jjim.jjim 을 실행, 해당 페이지로 jjim 리스트를 발송한다.
        dataType: 'html',
        //데이터 타입은 html
        success : function(html) {
            //해당페이지로 jjim 리스트가 발송되면

            jQuery('.jjimlist').html(jQuery.trim(html));
            //jjimlist div 에 html 을 불러온다.

        }
    })//ajax

}

};//success

});//ajax

};//else if

```

}//function

□ 레시피 리스트, 사용자에게 따른 찜 현황, 베스트, NEW 보여주기

```

<c:forEach items="${Recipelist}" var="vo" >
<!-- jstl 중 core 를 이용하여, recipelist 를 vo 라는 변수에 담아 사용한다.
-->

    <!-- forEach : 자바의 for 문과 동일하다 -->
    <div class="col-md-4">
    <div class="thumbnail">
    <!-- 부트스트랩 thumbnail -->

    <div class="ih-item square effect13 top_to_bottom"
        style="width:245px;height:245px">
    <!-- 마우스 오버시, 글자가 나타나는 이펙트 -->
    <a href="detail.recipe?num=${vo.getNum()}">
    <div class="new">
    <!-- ${pageContext.request.contextPath} : e1 문에서
    제공하는 절대경로. 현 페이지의 절대위치를 제공한다. -->
    
    </div>
    <c:if test="${vo.getRecommend() > 1 }">

```

```

<!-- 1: 베스트 게시물 조건, 추천수가 1 보다 컸을때 -->
<div class="best">
    
</div>
</c:if>

<c:forEach items="${NewRecipe }" var="n">
    <c:if test="${vo.getNum() == n.getNum()}">
        <div class="new">
            
<!-- NEW 이미지는 기존 Recipe 이미지보다 위에 있어야 하므로, z-index 를
이용하여 위로 올린다. -->
        </div>
    </c:if>
</c:forEach>

<!-- NewRecipe 를 변수 n 에 담아 사용하되, NewRecipe 가 null 이
아닐때까지 반복한다. -->

<br>
        <div class="info">
            <h3>Recipe View</h3>
        </div></a></div>

제목 : ${vo.getTitle()}<br>
<!-- 레시피 제목을 가져와서 표기한다. -->
<font color:grey>by</font> ${vo.getNick()}<br>
<!-- 레시피 작성자의 닉네임을 가져와서 표기한다. -->

<div class="text-right">
<!-- 부트스트랩에서 제공하는 위치 지정 div 로, align 을 오른쪽으로
지정한다. -->
    <c:set var="on" value="btn_like.gif"/>
<!-- btn_like.gif 값을 변수 on 으로 지정한다. -->
    <c:forEach items="${Rjjim }" var="j">
<!-- Rjjim 을 변수 j 에 담아 Rjjim 이 null 이 아닐때 까지 반복한다. -->
        <c:if test="${vo.getNum() == j}">
            <!-- 만약 RecipeList 의 getNum()이 j 와 같다면, 즉, 찜리스트에 담긴
레시피번호와 동일하다면 -->

```



```

                                <c:set var="on"
value="btn_like_on.gif"/>
    <!-- on 의 변수값을 btn_like_on.gif, 이미 좋아요가 되어있다는 의미의
이미지로 바꾼다. -->
                                </c:if>
                                </c:forEach>
                                <c:if test="${login ne null }"> <!-- login 이 null 이
아니라면 -->
                                
                                <!-- 찜리스트와 비교되어진 좋아요버튼을 표시한다. -->
                                </c:if>
                                </div>
                                </div>
                                </div>
</c:forEach>

```

□ 레시피 좋아요

```

/* 레시피의 좋아요 버튼을 눌렀을때 작동하는 function */

function RlikeClick(i,num,mnum){
    var likeck = i;
/* i 에는 버튼이 눌린 객체가 들어가 있다. */
/* if 에서 버튼의 이미지가 활성화 비활성화를 체크해서 번가라 실행 되게끔
되어있다. */
    if(likeck.src.match("btn_like.gif")){

        likeck.src="..${pageContext.request.contextPath}/img/number/b
tn_like_on.gif";
        /* 이미지를 활성화 이미지로 교체해 준다. */
        CommentLike(num,mnum,"Rlike.recipe");
        /* CommentLike 라는 Ajax 에 레시피번호 유저 번호와
URL 을 넘겨준다.
(Rlike.recipe 은 레시피의 좋아요 체크)*/

    }else if(likeck.src.match("btn_like_on.gif")){

        likeck.src="..${pageContext.request.contextPath}/img/number/b
tn_like.gif";
        /* 이미지를 활성화 이미지로 교체해 준다. */
        CommentLike(num,mnum,"Rdel_like.recipe");
        /* CommentLike 라는 Ajax 에 레시피번호 유저 번호와

```

URL 을 넘겨준다.
(Rdel_like.recipe 은 레시피의 좋아요 해제)*/

```
    }
}

□레시피 신고
```

```
/* 레시피 신고 */
function error_Recipe_Click(i,num,mnum){
/* 클릭된 버튼의 객체(i), 레시피번호(num), 회원번호(mnum) */
    var warning = i;
/* i 에는 버튼이 눌린 객체가 들어가 있다. */
/* if 에서 버튼의 이미지가 활성화 비활성화를 체크해서 번가라 실행 되게끔 되어있다. 한번 신고하면 신고는 풀리지 않는다.*/
    if(warning.src.match("btn_error.gif")){

        warning.src="..${pageContext.request.contextPath}/img/number/
btn_error_on.gif";
/* 레시피의 신고 이미지가 활성화 이미지로 교체된다. */
        Recipe_warning(num,mnum,"Rwarning.recipe");
/* Recipe_warning 라는 레시피 신고 Ajax 로
레시피번호(num), 유저번호(mnum), URL 을 넘겨준다.
(Rwarning.recipe 은 레시피 신고 )*/
    }else if(warning.src.match("btn_error_on.gif")){
        alert("이미 신고한 댓글 입니다.");
    }
}

}
```

```
/* 레시피 신고 Ajax */
function Recipe_warning(num,mnum,url)
{
    var xmlhttp;
    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            /* xmlhttp.readyState==4 이고 xmlhttp.status==200
```

```

        이면 Ajax 가 성공적으로 실행 된것이다.*/
        alert("해당 게시물을 신고했습니다.");
    }
}
xmlhttp.open("GET",url+"?num="+num+"&mnum="+mnum,true);
/* GET 방식으로 넘겨받은 URL 에 레시피번호(num) 회원번호(mnum)를
파라미터로 넘겨준다. */
xmlhttp.send();
}

```

□ 댓글 좋아요

```

/* 댓글에 좋아요 버튼을 눌렀을때 작동하는 function */
function likeClick(i,cnum,mnum){
    var likeck = i;
/* i 에는 버튼이 눌린 객체가 들어가 있다. */
/* if 에서 버튼의 이미지가 활성화 비활성화를 체크해서 번가라 실행 되게끔
되어있다. */
    if(likeck.src.match("btn_like.gif")){

        likeck.src="..${pageContext.request.contextPath}/img/number/b
tn_like_on.gif";

        /* 이미지를 활성화 이미지로 교체해 준다. */
        CommentLike(cnum,mnum,"like.recipe");
        /* CommentLike 라는 Ajax 에 댓글번호 유저 번호와
URL 을 넘겨준다.
(like.recipe 은 댓글의 좋아요 체크)*/

    }else if(likeck.src.match("btn_like_on.gif")){

        likeck.src="..${pageContext.request.contextPath}/img/number/b
tn_like.gif";

        /* 이미지를 비활성화 이미지로 교체해 준다. */
        CommentLike(cnum,mnum,"del_like.recipe");
        /* CommentLike 라는 Ajax 에 댓글번호 유저 번호와
URL 을 넘겨준다.
(del_like.recipe 은 댓글의 좋아요 해제)*/

    }
}
}

```

□ 댓글 신고

```

/* 댓글 신고 */
function errorClick(i,cnum,mnum) {

```

```
/* 클릭된 버튼의 객체(i), 댓글번호(cnum),
                                     회원번호(mnum) */
    var warning = i;
/* i 에는 버튼이 눌린 객체가 들어가 있다. */
/* if 에서 버튼의 이미지가 활성화 비활성화를 체크해서 번가라 실행 되게끔
   되어있다. 한번 신고하면 신고는 풀리지 않는다.*/
    if(warning.src.match("btn_error.gif")){

        warning.src="..${pageContext.request.contextPath}/img/number/
btn_error_on.gif";
/* 댓글의 신고 이미지가 활성화 이미지로 교체된다. */
        Comment_warning(cnum,mnum,"Cwarning.recipe");
/* Comment_warning 라는 댓글 신고 Ajax 로 댓글번호(cnum),
   유저번호(mnum), URL 을 넘겨준다. (Rwarning.recipe 은 레시피 신고 )*/
    }else if(warning.src.match("btn_error_on.gif")){
        alert("이미 신고한 게시물 입니다.");
    }
}

/* 댓글 신고 ajax */
function Comment_warning(cnum,mnum,url)
{
    var xmlhttp;
    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            /* xmlhttp.readyState==4 이고 xmlhttp.status==200 이면 Ajax 가
               성공적으로 실행 된것이다.*/
            alert("해당 댓글을 신고했습니다.");
        }
    }
}

    xmlhttp.open("GET",url+"?cnum="+cnum+"&mnum="+mnum,true);
/* GET 방식으로 넘겨받은 URL 에 댓글번호(cnum) 회원번호(mnum)를 파라미터로
   넘겨준다. */
    xmlhttp.send();
```

```
}

```

□ 레시피, 댓글 좋아요 ajax

```
/* 댓글 좋아요 Ajax */
function CommentLike(cnum,mnum,url)
{
var xmlhttp;
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
else
    { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        /* xmlhttp.readyState==4 이고 xmlhttp.status==200 이면 Ajax 가
        성공적으로 실행 된것이다.*/
        /* URL 에 del 이 있으면 싫어요 없으면 좋아요라는 경고창이 뜬다. */
        if(url.match("del")){
            alert("싫어요");
        }else{
            alert("좋아요");
        }
    }
}

xmlhttp.open("GET",url+"?cnum="+cnum+"&mnum="+mnum,true);
/* GET 방식을 넘겨받은 URL 에 파라미터로 댓글번호(cnum)와 유저번호(mnum)를
넘겨준다. */
xmlhttp.send();
}
```

□ 레시피 상세 페이지 소스

```
<div></div>
```

[illegible]

```

<br>

<c:set var="CK" value="btn_like.gif"/>
<!-- jstl 로 CK 라는 변수에 btn_like.gif(비활성화
이미지)라는 값을 넣어놓는다. -->
<!-- jstl 반복문으로 로그인한 유저가 해당 레시피를
좋아요 했으면 CK 변수에 값으로
btn_like_on.gif(활성화 이미지)을 넣어준다. -->
<c:forEach items="${RLike }" var="r">
    <c:if test="${sessionScope.mnum == r}">
        <c:set var="CK"
            value="btn_like_on.gif"/>
    </c:if>
</c:forEach>

<c:set var="Rerror" value="btn_error.gif" />
<!-- jstl 로 CK 라는 변수에 btn_error.gif(비활성화
이미지)라는 값을 넣어놓는다. -->
<!-- jstl 반복문으로 로그인한 유저가 해당 레시피를
신고 했으면
CK 변수에 값으로 btn_error_on.gif(활성화 이미지)을
넣어준다. -->
<c:forEach items="${Rwarning }" var="RW">
    <c:if test="${sessionScope.mnum == RW}">
        <c:set var="Rerror"
            value="btn_error_on.gif"/>
    </c:if>
</c:forEach>
<!-- 해당 이미지가 클릭되면 javascript 에
RlikeClick 함수에 인자로 클릭된 객체(this)와
레시피번호(Recipe.getNum())와 로그인한
유저번호(mnum)를 넘겨준다. -->

    <!-- 해당 이미지가 클릭되면 javascript 에
    error_Recipe_Click 함수에 인자로 클릭된
    객체(this)와 레시피번호(Recipe.getNum())와
    로그인한 유저번호(mnum)를 넘겨준다. -->
    
    </div>
</c:if>
</div>

```

```

<br>
    <div style="clear:both; padding-left: 100px; background-color:
    #F1F1F2; border-radius:10px" >
        <dl>
            <dt>[주재료]</dt>
            <!-- 주제료(Recipe.getMaterial()) -->
            <dd>${Recipe.getMaterial() }</dd>
        </dl>
    </div>

    <!-- jstl 로 i 라는 변수를 선언 -->
    <c:set var="i" value="0"/>
    <!-- Step 은 요리의 설명과 이미지를 담고있다. forEach(반복문)에 변수로
    s 에 Step 을 담는다. ArrayList 로 CookStepVO 을 담고 있다. -->
    <c:forEach items="${Step}" var="s">
        <!-- i 는 반복문을 돌면서 값이 1 씩 증가 한다. -->
        <c:set var="i" value="${i+1 }"/>
        <div style="margin-bottom: 30px;">
            <div style="clear:both;">
                <!-- icon_step_${i }.gif(해당 gif 는 숫자 이미지이다.)에
                i 가 증가하면 해당 이미지가 바뀌어서 보여주게된다. -->
                
            </div>
            <div>
                <!-- 레시피 설명(s.getContents()) -->
                <font size="6pt">${s.getContents()}</font>
            </div>
            <div style="margin-left: 60px">
                <!-- 레시피이미지(s.getImage()) --><br>
                
            </div>
        </div>
    </c:forEach>

    <div style="margin-bottom: 30px; margin-top: 30px">
    <div>
        
        <!-- 레시피 팁(Recipe.getTip()) -->
        <br><div style="margin-left: 30px; font-size:
24px">${Recipe.getTip() }</div>
    </div>
    </div>

```



```

<div>
<!-- Comment 를 반복문의 변수로 i 에 담는다. Comment 는 ArrayList 로
Recipe_CommentVO 를 담고 있다. -->
    <c:forEach items="${Comment }" var="i">
    <div style="clear:both; margin-bottom:30px;">
    <!-- 댓글을 작성한 유저의 프로필사진(i.getProfilpic()) --
>
        
    <!-- 댓글을 작성한 유저의 닉네임(i.getNick())과 작성일(i.getCdate()) --
>
        <font color="#44b6b5"
size="4px">아이디:${i.getNick() }</font> <font color="#999"
size="2px">작성일:${i.getCdate() }</font>
    <!-- 로그인 했으면(sessionScope.mnum != null) 댓글에 좋아요와 신고를
볼수 있게끔 한다. -->
        <c:if test="${sessionScope.mnum != null }">
        <div style="display: inline;">
        <!-- jstl 로 on 변수에 btn_like.gif 라는 값을 넣는다. -->
        <c:set var="on" value="btn_like.gif"/>
    <!-- 반복문 변수에 Clike(은 로그인한 유저가 좋아요한 댓글번호를 담고
있다.)를 j 에 담는다 -->
    <!-- j 와 i 의 댓글번호가 같으면 on 변수에 btn_like_on.gif 을 담는다. -->
        <c:forEach items="${CLike }" var="j">
            <c:if test="${i.getCnum() == j}">
                <c:set var="on"
value="btn_like_on.gif"/>
            </c:if>
        </c:forEach>

        <!-- jstl 변수 Cerror 에 btn_error.gif 라는 값을 넣는다. -->
        <c:set var="Cerror" value="btn_error.gif"/>
    <!-- 반복문 변수 Cwar 에 Cwarning(은 로그인한 유저가 신고한 댓글의 번호를
담고있다)을 넣는다. -->
    Cwar 화 i.getCnum()(i.getCnum 은 댓글 번호이다.)이 같으면 Cerror 변수에
btn_error_on.gif 라는 값을 넣는다. -->
        <c:forEach items="${Cwarning }" var="CWar">
            <c:if test="${i.getCnum() == CWar}">
            <c:set var="Cerror" value="btn_error_on.gif"/>
            </c:if>
        </c:forEach>

```

<!-- 댓글의 좋아요 버튼을 누르면 javascript 의 likeClick 로 클릭된 객체와 댓글번호와 유저번호를 인자로 전달해준다. -->

```

```

<!-- 댓글의 신고 버튼을 누르면 javascript 의 errorClick 로 클릭된 객체와 댓글번호와 유저번호를 인자로 전달해준다. -->

```

```

if 문으로 해당 댓글을 작성한 사람이 자신의 댓글을 삭제 할 수 있다.(자신이 작성한 댓글에는 삭제글자가 보인다.) -->

```
<c:if test="${sessionScope.mnum == i.getMnum() }">
|
<a href="commentDelete.recipe?cnum=${i.getCnum()}&num=${Recipe.getNum
() }&mnum=${mnum }">삭제</a>
```

```
</c:if>
```

```
</div>
```

```
</c:if>
```

```
<!-- 댓글 내용(i.getCcomment()) -->
```

```
<br>${i.getCcomment() }
```

```
</div>
```

```
</c:forEach>
```

<!-- 로그인은 해서 세션에 회원번호가 있으면 댓글을 작성 할 수 있다. -->

```
<c:if test="${sessionScope.mnum != null }">
```

```
<div style="clear:both;">
```

```
<!-- 로그인한 회원의 프로필 이미지 -->
```

```

```

```
</div>
```

```
<div>
```

```
<form action="comment.recipe"
```

```
method="GET">
```

<!-- 댓글을 작성하면 hidden 타입으로 레시피 번호를 보내준다. -->

```
<input type="hidden" name="num"
```

```
value="${Recipe.getNum() }">
```

<!-- 댓글을 작성하면 hidden 타입으로 회원번호를 보내준다. -->

```
<input type="hidden" name="mnum"
```

```
value="${mnum }">
```

```
<textarea rows="6" cols="50"
```

```
name="content" style="height:100px; float: left;
```

```
resize:none;"></textarea>
```

```
<input type="submit" value="등록"
```

```
style="width:100px; height:100px;">
```

```

        </form>
    </div>
</c:if>
</div>

```

□ 레시피 작성 페이지 중 요리순서

```

<p class="cont_tit4" style="width:670px;">요리순서</p>
    <div id="mystep">
        <p class="cont_tit2_1 ui-sortable-handle" style="vertical-align: middle;">Step1</p>
        <img id="step_img[0]" name="step_img[0]" class="imgs0" style="width: 160px; background-color: #f5f5f5; height: 160px; border-radius: 5px;"/>
        <textarea rows="5" cols="40" name="step[]" id="step[]" class="textareas" style="height: 160px; width: 410px; resize: none;"></textarea>
        <input type="file" name="fileupload[0]" id="fileupload[0]" class="files0" onclick="JavaScript:preview(0)" style="background-color:white; border-color: white; width: 100px; float: right; display: inline-block;"/><br><br>
    </div>
    <input name="btnButton" id="btnButton" class="btn btn-default" type="button" value="(+) 순서추가" onclick="JavaScript:fncCreateElement();" style="background-color: white; border-color: white; cursor: pointer;">
    <!-- 요리순서를 추가하는 버튼 -->

<script>
var i=1;
function fncCreateElement(){
if(i==20){alert('더 이상 추가할 수 없습니다.'); return;}
/* 20 개를 넘어가면 더이상 추가할 수 없다 */
var mySpan = document.getElementById('mystep');

var myStepnum = document.createElement('p');/* 태그를 생성 */
var Stepnum = document.createTextNode("Step" + (i+1));
/* p 태그에 출력할 문자 생성 */
myStepnum.setAttribute('class', "cont_tit2_1 ui-sortable-handle");//속성설정
myStepnum.setAttribute('style', "vertical-align: middle;");//속성설정
myStepnum.appendChild(Stepnum);//문자 삽입

```

Chapter 6 주요 소스 분석

mystep.appendChild(myStepnum);//mySpan 영역에 p 태그를 추가한다

```
var myElements = document.createElement('img');
myElements.setAttribute('name',"step_img["+i+"]");
myElements.setAttribute('id',"step_img["+i+"]");
myElements.setAttribute('class',"imgs"+i);
myElements.setAttribute('style',"width: 160px; background-
color: #f5f5f5; height:
160px; border-radius: 5px;");
mystep.appendChild(myElements);

var myElement4 = document.createElement('textarea');
myElement4.setAttribute('rows',"5");
myElement4.setAttribute('cols',"40");

myElement4.setAttribute('name',"step[]");
myElement4.setAttribute('id',"step[]");
myElement4.setAttribute('class',"textareas");
myElement4.setAttribute('style',"height: 160px; width: 410px;
resize: none;
margin-left: 4px;");
mystep.appendChild(myElement4);

var myElement1 = document.createElement('input');
myElement1.setAttribute('type',"file");

myElement1.setAttribute('name',"fileupload["+i+"]");

myElement1.setAttribute('id',"fileupload["+i+"]");
myElement1.setAttribute('class',"files"+i);
myElement1.setAttribute('onclick',
"JavaScript:preview["+i+"]");
myElement1.setAttribute('style',"background-color:white;
width: 100px; float: right; border-color: white; display: inline-
block;");
mystep.appendChild(myElement1);
/** Remove Element */

var myElement2 = document.createElement('br');
mystep.appendChild(myElement2);
var myElement3 = document.createElement('br');
mystep.appendChild(myElement3);
i++;/* i 증가 */
}
```

```
</script>
```

□ 의견전달

```
$('#customerbtn').click(function(){ //footer 의 의견제출 버튼 클릭 시,
    if($('#customerEmail').val()!=""){
        //고객 email 이 빈칸이 아니라면
        $.ajax({ //ajax 를 실행한다.
            Type: 'POST', // 타입은 post
            url :
                "${pageContext.request.contextPath}/main/CustomerInsert.jsp",
            //고객의견발송페이지
            data: $('#customer').serialize(),
            //폼 전송(폼의 모든 내용을 parameter 로
                전송한다.)
            success: function(data){

                alert('의견이 제출되었습니다. 감사합니다.');
```

```
                $('#customerSend').each(function(){
                    $(this).val("");
                }); //내용 삭제 기능
                $('#customerEmail').each(function(){
                    $(this).val("");
                }); //이메일칸 내용 삭제 기능
            }
        }); //ajax
    }else{
        alert('이메일을 적어주세요!');
```

```
    }
}); //click
```

□ 메일 발송 페이지

```
<% //고객의견 전달 페이지
request.setCharacterEncoding("utf-8");

StringBuffer sb = new StringBuffer();
String board = request.getParameter("board_txt"); //컨텐츠 내용
String customerEmail = request.getParameter("customerEmail");
sb.append(board);
sb.append("<br>["+customerEmail+"]");
String content = sb.toString();
```

// 입력값 받음

```
Properties p = new Properties(); // 정보를 담은 객체
```

```
p.put("mail.smtp.user", "jihun85112");
```

```
p.put("mail.smtp.host", "smtp.naver.com"); // 네이버 SMTP
```

//네이버 SMTP 접속 정보

```
p.put("mail.smtp.port", "465");
```

```
p.put("mail.smtp.starttls.enable", "true");
```

```
p.put("mail.smtp.auth", "true");
```

```
p.put("mail.smtp.debug", "true");
```

```
p.put("mail.smtp.socketFactory.port", "465");
```

```
p.put("mail.smtp.socketFactory.class",
```

```
"javax.net.ssl.SSLSocketFactory");
```

```
p.put("mail.smtp.socketFactory.fallback", "false");
```

// SMTP 서버에 접속하기 위한 정보들

```
try{
```

```
    Authenticator auth = new SMTPAuthenticator();
```

```
    Session ses = Session.getInstance(p, auth);
```

```
    ses.setDebug(true);
```

```
    MimeMessage msg = new MimeMessage(ses); // 메일의 내용을 담은 객체
```

```
    msg.setSubject("고객 의견사항 전달내용"); // 제목
```

```
    Address fromAddr = new InternetAddress("jihun85112@naver.com");
```

```
    msg.setFrom(fromAddr); // 보내는 사람
```

```
    Address toAddr = new
```

```
InternetAddress("jihun85112@naver.com");//받는 사람 이메일
```

```
    msg.addRecipient(Message.RecipientType.TO, toAddr); // 받는 사람
```

```
    msg.setContent(content, "text/html; charset=UTF-8"); // 내용과  
인코딩
```

```
    Transport.send(msg); // 전송
```

```
} catch(Exception e){
```

```
    e.printStackTrace();
```

```
    //out.println("<script>alert('Send Mail  
Failed..');history.back();</script>");
```

```
    // 오류 발생시 뒤로 돌아가도록
```

```
    return;
```

```
}
```

```
//out.println("<script>alert('Send Mail
Success!!');location.href='successMail.jsp';</script>");
// 성공 시
%>
```

□ 레시피 작성자 검색 ajax

```
$('#findnickrecipe').click(function(){
    var nick = $('#input#findnick').val(); //작성자검색란의 값
    $.ajax({          //ajax 실행
        type: "POST", //타입은 post
        url:
            "..${pageContext.request.contextPath}/findnickrecipe.recipe",
            //닉네임 확인페이지
        data: {"nick" : nick}, //parameter 로 name: nick, value:
            nick(작성자검색란의 값)을 전달한다
        dataType: 'html', //데이터타입은 html
        success : function(html) {
            jQuery('.find').html(jQuery.trim(html));
            //작성자 검색이 되면, 해당 결과물을 가져온다

            $('#findnick').each(function(){
                $(this).val("");
            }); //input 칸 삭제

            $('.list').hide(); //밑의 list 를 숨긴다.
        } //success
    }) //ajax
}); //click
```

□ 레시피 삭제 ajax

```
function RecipeDelete(i, j){ //레시피 삭제 버튼 클릭시
    //i : this 를 담기위한 임시 변수
    //j : 레시피 번호가 담긴 변수
    //alert('레시피 삭제 버튼이 눌렸습니다. Recipelist 페이지');
    var deletenum = j;
    $.ajax({ //ajax 실행
        type: "POST", //타입 post
        url:
            "..${pageContext.request.contextPath}/deleterecipe.recipe",
```

```

//레시피 삭제 페이지
data : {"num" : deletenum},
//parameter 전달, name: num, value: deletenum
success: function(data) { //해당 레시피를 삭제한다
    alert('해당 레시피가 삭제되었습니다.
    정리하기를 눌러주세요.')
    $.ajax({ //ajax 를 실행한다
        type: "POST", //타입은 post
        url:
"..${pageContext.request.contextPath}/AdminRecipeResultList.recipe",
        //servlet 을 실행한다.
        dataType: 'html',
        //데이터타입은 html
        success : function(html) {

            jQuery('.list').html(jQuery.trim(html));
            //레시피결과물이 담긴 페이지를 불러온다.
        }
    })//ajax

    }//success

});
};

```

□ 레시피 정리하기 버튼 클릭 시 실행되는 function

```

$('#findlistdelete').click(function(){//정리하기 버튼

    jQuery('.find').empty();//검색 결과물을 비운다.
    $('.list').show();//리스트를 띄운다.

});

```

□ 찜 등록 ajax 실행 페이지

```

<% //ajax 실행 페이지. 찜리스트에 레시피번호와 유저번호를 담는다.
    int num =
    Integer.parseInt(request.getParameter("num"));
    int mnum =
    ((Integer)(session.getAttribute("mnum"))).intValue();

```



```

JJimDAO dao = JJimDAO.getInstance(); //DB 연결 실행
System.out.println("ajax 실행");
System.out.println("num : "+num);
System.out.println("mnum : "+mnum);
String str="";
try{
    System.out.println("찜 등록 실행");
    boolean isjjim = dao.jjim(num, mnum);
//jjim 함수 실행, 매개변수로 레시피번호와 유저번호 전달
    System.out.println(isjjim);
    if(isjjim){ //DB 에 입력 성공 시
        str="YES";
        //찜 등록 성공
        out.print(str);
    }else{ //DB 입력 실패 시
        str="NO";
        //찜 등록 실패
        out.print(str);
    }
}catch(Exception e){
    System.out.println(e+"=> 중복체크 실패");
    e.printStackTrace();
}
%>

```

❑ 찜 취소 ajax 실행 페이지

```

<% //ajax 실행 페이지. 찜리스트에 담긴 레시피번호와 유저번호를 삭제한다.
int num=Integer.parseInt(request.getParameter("num"));
int mnum =
((Integer)(session.getAttribute("mnum"))).intValue();
JJimDAO dao = JJimDAO.getInstance(); //DB 연결 실행
System.out.println("ajax 실행");
System.out.println("num : "+num);
System.out.println("mnum : "+mnum);
String str="";
try{
    System.out.println("찜 취소 실행");
    dao.jjimcancle(num, mnum);
//매개변수로 레시피번호와 유저번호 전달, jjimcancle 함수 실행
    str="YES";
    //찜 취소 성공
    out.print(str);
}

```

```

    }catch(Exception e){
        System.out.println(e+"=> 취소 실패");
        e.printStackTrace();
    }
%>

```

□ 레시피 수정 페이지

```

<c:forEach items="${Step}" var="s" varStatus="status">
    <p class="cont_tit2_1 ui-sortable-handle" style="vertical-align: middle;">Step${i+1}</p>
    
    <textarea rows="5" cols="40" name="step[]" id="step[]"
class="textareas"
style="height: 160px; width: 410px; resize:
none;">${s.getContents()}</textarea>

    <input type="file" name="fileupload[${i}]"
id="fileupload[${i}]"
class="files[${i}]" onclick="JavaScript:preview(${i})"
style="background-color:white;
border-color: white; width: 100px; float: right; display:
inline-block;">
    <input type="hidden" name="h_img[${i }]" id="h_img[${i }]"
value="${s.getImage()}" /></input><br><br>
    <c:set var="i" value="${i+1 }" />
    <!-- 원래 있던 순서까지 for 문이 돈다 -->
</c:forEach>

<script>
    var j= ${i }; //j 는 i+1 이다

    console.log(j);
    function fncCreateElement(){
//요리순서에 관련된 element 들 추가하는 function.
    console.log(j);

    if(j>20){alert('더 이상 추가할 수 없습니다. '); return;}
    var mySpan = document.getElementById('mystep');

```

```
var myStepnum = document.createElement('p');
var Stepnum = document.createTextNode("Step"+(j+1));
myStepnum.setAttribute('class', "cont_tit2_1 ui-sortable-handle");
myStepnum.setAttribute('style', "vertical-align: middle;");
myStepnum.appendChild(Stepnum);
mystep.appendChild(myStepnum);
```

```
var myElements = document.createElement('img');
myElements.setAttribute('name', "step_img["+j+"]");
myElements.setAttribute('id', "step_img["+j+"]");
myElements.setAttribute('class', "imgs"+j);
myElements.setAttribute('style', "width: 160px; background-color: #f5f5f5; height: 160px; border-radius: 5px;");
mystep.appendChild(myElements);
```

```
var myElement4 = document.createElement('textarea');
myElement4.setAttribute('rows', "5");
myElement4.setAttribute('cols', "40");
/* myElement4.setAttribute('name', "step"+i);
myElement4.setAttribute('id', "step"+i); */
myElement4.setAttribute('name', "step["+j+"]");
myElement4.setAttribute('id', "step["+j+"]");
myElement4.setAttribute('class', "textareas");
myElement4.setAttribute('style', "height: 160px; width: 410px; resize: none; margin-left: 4px;");
mystep.appendChild(myElement4);
```

```
var myElement1 = document.createElement('input');
myElement1.setAttribute('type', "file");
myElement1.setAttribute('name', "fileupload["+j+"]");
myElement1.setAttribute('id', "fileupload["+j+"]");
myElement1.setAttribute('class', "files"+j);
myElement1.setAttribute('onclick',
    "JavaScript:preview["+j+"]");
myElement1.setAttribute('style', "background-color:white; width: 100px; float: right; border-color: white; display: inline-block;");
mystep.appendChild(myElement1);
/** Remove Element */
/*
var deleteEle = document.getElementById('txt1');
mySpan.removeChild(deleteEle);
```

```
        */
        var myElement2 = document.createElement('br');
        mystep.appendChild(myElement2);
        var myElement3 = document.createElement('br');
        mystep.appendChild(myElement3);
        j++;
    }

</script>
```

□ 프로필 사진 변경

```
/** part 의 헤더에서 filename 값만 분리해서 리턴*/
private String getFileName(Part part) {
    String contentDisp = part.getHeader("content-disposition");
    System.out.println("content-disposition header= " + contentDisp);
    String[] tokens = contentDisp.split(";");
    for (String token : tokens) {
        if (token.trim().startsWith("filename")) {
            return token.substring(token.indexOf("=") + 2,
                token.length() - 1);
        }
    }
    return "";
}
```

작성자 : 박형진

정말 어려웠습니다.

너무 생소하고 힘들었다 그러나 최대한 해내려고 욕심을 가지고 했습니다.

일단 이번 프로젝트에서 맡은 역할은 관리자 페이지를 만드는 것이었습니다.

처음에 맡았을 때 정말 막막했습니다. 에러도 많이 발생하고 뜻대로 잘되지도 않았습니다.

하지만 조장 덕분에 물어보고 많은 것을 배우고 잘 알려주셔서 정말 재미있게 만들었습니다. 또 지훈이형이 맡았던 것도 보면서 물어보고 배우고 뜻깊은 시간이었던 것 같습니다. 곧 또 2차프로젝트를 시작하겠지만 거기서도 열심히 배우고 맡은일을 해내야겠다는 생각을 다짐했습니다.

마지막으로 정말 아쉬웠습니다.

코드를 조금만 더 분석하고 수정하면 뭔가 벌어질 것만 같은 느낌에 열심히 했지만 소득이 없어 억울하기도 했습니다만 중요한 건 진짜로 코드분석을 조금만 더 하고 MVC 구조를 파악하고 수정만 하면 뭔가 된다는 점을 깨닫게 되었습니다.

나름대로 이번에 관리자 페이지를 완성 해냈지만 도움을 많이 받아서 만족스럽지는 않습니다. 다음 프로젝트 때에는 꼭 스스로 완성토록 노력할 것입니다.

작성자 : 양지훈

조원들과의 회의를 통해 결정된 프로젝트는, 어느 웹페이지에서나 제공하는 내용이라 그나마 어렵지 않을 것으로 생각되었습니다. 그러나, 벤치마킹하려는 사이트의 내용을 보면서, 제 생각이 얼마나 안일한 생각이었는지를 실감했습니다. 상용화 되어있는 사이트들은 제 생각보다 더 뛰어난 기술들과 아이디어로 이뤄져 있었고, 그것을 그대로 사용하기에도 벅찬 정도였습니다. 때문에, 벤치마킹하기로 한 사이트의 구성과 형태는 최대한 비슷하게 구현하되, 소스는 제가 가진 실력과 인터넷을 토대로 가능하면 쉽게, 누구나 알아볼 수 있되, 기능 구현은 난이도 있도록 구현코자 하였습니다.

프로젝트 진행 과정에서, git 을 사용하는데 있어 어려움을 많이 겪어서 진행에 있어 난항도 있었습시다만, 그래도 불평불만없이 잘 따라와주는 팀원들 덕에 프로젝트가 잘 마무리될 수 있어서 다행이라고 생각합니다.

더불어 서로의 부족했던 실력을 키울 수 있던 좋은 기회였을 뿐만 아니라, 팀원들간에 서로 의지할 수 있었던 좋은 기간이었습니다.

웹사이트의 제작이 쉽고도 어렵다는 것을 다시 한번 느낍니다. 공부에 끝이 없다는 것을 배웠고, 알고 있는 지식은 실무에 비하면 초보자 수준이며, 실무에 적응하기 위해선 끝없이 정진해야함을 다시 한번 크게 느끼게 되었습니다.

팀장으로서, 조원들이 만든 소스를 Mapping 하면서 조원들의 코드를 보고 많은 걸 느꼈고, 많이 배웠습니다. 팀이 프로젝트를 진행할 수 있도록 이끌어 가는 법, 팀 내에 불협화음이 생기지 않도록 조절하는 법, 기분을 맞춰주고 프로젝트에 집중할 수 있도록 만들어주는 법 등 팀장으로서 팀과 조화롭게 진행하고 이끌어 가는 법을 배울 수 있었던 행복한 기간이었습니다.

앞으로, 좋은 결과물을 많이 만들어 낼 수 있도록 공부에 정진할 것입니다.

작성자 : 유슬아

Chapter 7 프로젝트 후기

이번 팀 프로젝트를 하면서 느낀점은 말 그대로 조원들과 함께 만들어 가는게 진정한 프로젝트라는걸 느꼈다.

그동안 했던 프로젝트는 의사소통이 제대로 되지 않아 조원들 서로에게 힘든 프로젝트를 많이 해봤는데, 이번 프로젝트는 조원 모두 서로 의견에 귀기울여주고, 열심히 참여하는 모습에 나 또한 더 열심히 할려고 노력했다. 주제 선정부터 문서 작업까지 조원들과 함께 해서 힘든 작업도 문제없이 원만히 해결 할 수 있었다.

짧은 기간동안 해야하는 부담감도 있었지만, 모두 다 같이 노력해서 기간 내에 마무리 할 수 있었던것 같다.

다음 프로젝트를 할때에는 지금보다 더 노력해서 조원들에게 많은 도움이 될 수 있도록 실력을 향상시켜야 겠다.

작성자 : 이희진

일단 프로젝트가 끝나서 매우 기쁩니다.

배웠던 지식을 적용해보고, 배우지 않았던 것들도 새로 배우면서 프로젝트를 진행하였습니다. 6 명이나 되는 많은 인원과 3 주라는 기간 동안 프로젝트를 한 것은 처음이라 좋은 경험이었던 것 같습니다. 웹 사이트를 만드는 일이 엄청 어려운 일이라는 것을 깨달았습니다.

작성자 : 한재원

드디어 1 차 프로젝트가 끝났습니다. 만들면서 에러 때문에 고민을 하는 경우가 많았는데 에러를 해결하고 다 완성하니 정말 이걸 3 주만에 다 만들었구나 싶습니다. 3 주동안 만든 게 이것저것 있지만 가장 기억에 남는 것 요리순서 쓰기부분이 생각납니다. 제가 맡은 부분 중 가장 에러를 많이 발견한 부분이고 주요기능을 완성하기 위해서 꼭 필요한 부분이라서 정말 완성했을 때는 드디어 성공했구나 싶었습니다. 다음에 프로젝트를 할 때는 지금보다 더 잘 할 수 있을 거라는 생각과 자신감이 생깁니다.

작성자 : 유승모

처음 프로젝트 회의를 하면서 무슨 웹사이트를 만들까 고민을 하면서 인터넷으로 연령대별로 관심사가 무엇인지를 조사해 보았는데 20 대는 아르바이트, 맛집, 여행 이었는데 그 중에 요즘 뜨고 있는 TV 에서 셰프들이 만든 요리를 집에서 해먹는 사람들이 늘어나고 관심이 높아졌는데 프로젝트때 나온 레시피웹 사이트를 만들자는 의견이 모여면서 프로젝트로 요리 레시피 웹사이트로 결정되었습니다.

프로젝트 회의를 계속하면서 DB 를 설계하고 역할을 나누면서 담당파트가 레시피에서 레시피 상세 페이지와 셰프 비디오를 맡게 되었습니다. 레시피 상세 페이지를 만들면서 레시피 추천과 댓글 추천등을 만들면서 처음으로 Oracle 의 트리거를 처음 써봤는데 추천을 증가시키고 감소시키는 것은 쉬워 졌는데 레시피 삭제와 댓글 삭제가 생각 보다 복잡해져서 트리거 때문에 프로젝트가 너무 힘들어 지는 것이 아닌가 고민을 많이 했었습니다. 프로젝트를 진행 중 팀원들과 주기적으로 회의를 하면서 프로젝트의 필요한 부분이 추가되고 수정 되어가면서 완성되는 것에 크나큰 만족을 느꼈습니다.

아쉬웠던 부분은 셰프 비디오를 하면서 비디오를 다운 받아서 연결 하는 것이 좋을까 링크를 걸어서 하는 것이 좋을까 고민을 하면서 비디오를 다운 받으면 프로젝트의 용량이 너무 커지는 것 같고 링크를 걸면 페이지가 로딩이 오래 걸려서 해당 페이지가 너무 무거워지는 것이 아닌가 고민을 많이 했었습니다. 또한 레시피 상세 페이지가 진행이 늦어지면서 셰프 비디오에 시간을 많이 투자 못한 것이 아쉬운 부분으로 남았지만 팀원들과 프로젝트를 진행하면 예전 프로젝트보다 전체적으로 완성도가 높아서 개발을 하면서 큰 보람을 느꼈고 개인적으로 큰 만족을 한 프로젝트였습니다.

-향후 개인 계획-

프로젝트를 하면서 부트스트랩을 사용하였는데 부트스트랩을 잘 몰라서 팀원을 못 도와줘서 다음 번 프로젝트 때는 이를 보완 하고 다음번 프로젝트를 Spring 으로 진행을 할 예정 이기 때문에 Spring, jQuery 공부를 하여 보다 나은 동적이고 디자인적인 웹사이트를 만들 것 입니다.

또한 이번 프로젝트 때는 API 는 사용하지 않았지만 다음번 프로젝트 때는 API 를 적용시켜 보기 위해 유용한 API 가 무엇이 있는지 조사해 볼 예정입니다.