프로젝트 소개

프로젝트 명 : 홈 밥

사먹는 음식에 질린 홈밥러들에게 필요한 홈페이지

요리 실력이 없어도 맛있는 집밥 요리를 해먹는 레시피 맛집!

이곳저곳 음식사이트 돌아다니지 말고 홈밥에서 모든걸 확인하세요!

내가 자주해먹는 나만의 레시피 등록하고 레시피왕이 되어보세요!

사용자 별 검색 기록을 통한 메뉴 추천을 해드립니다!

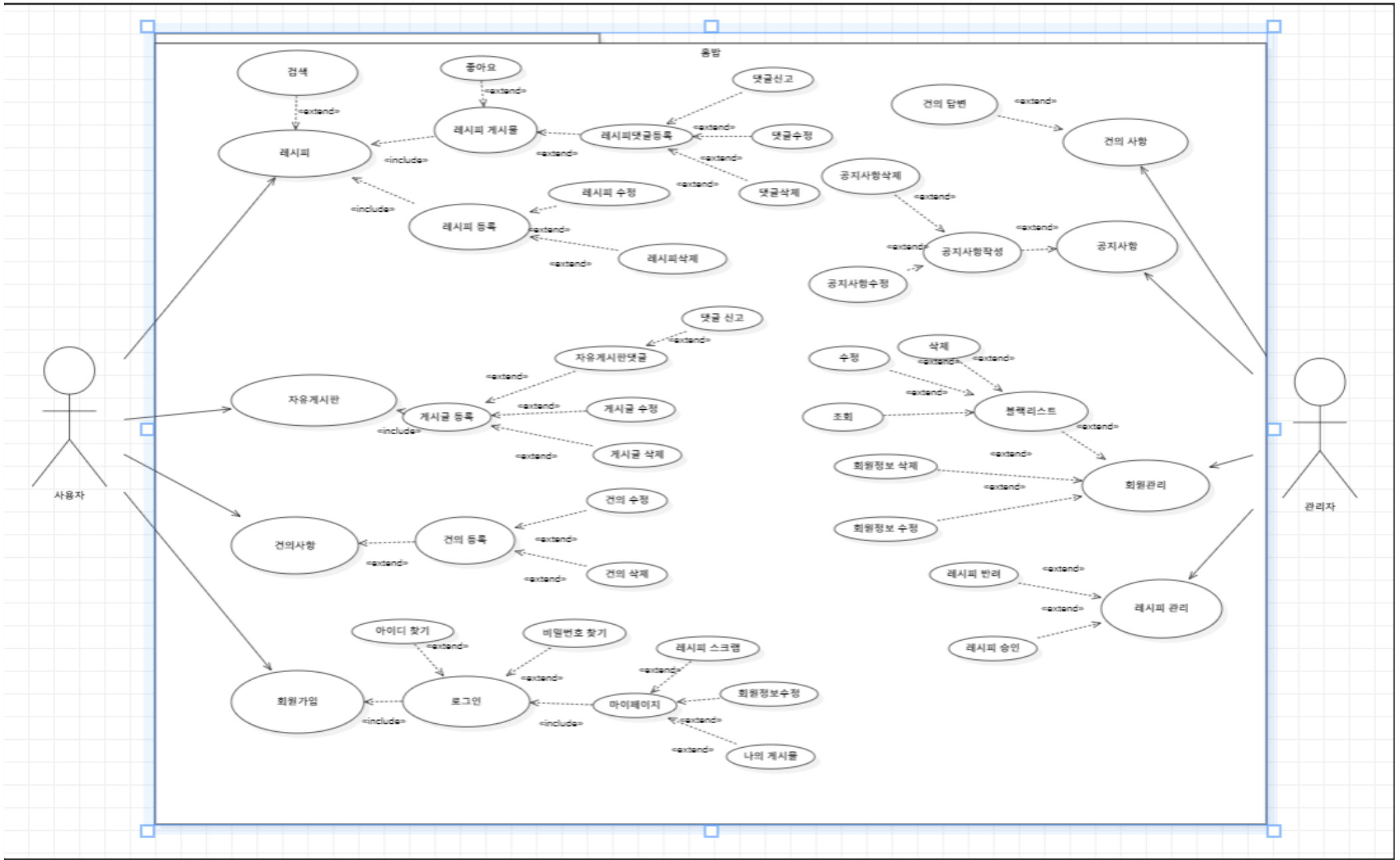
프로젝트 일정(WBS)

2022년 7월

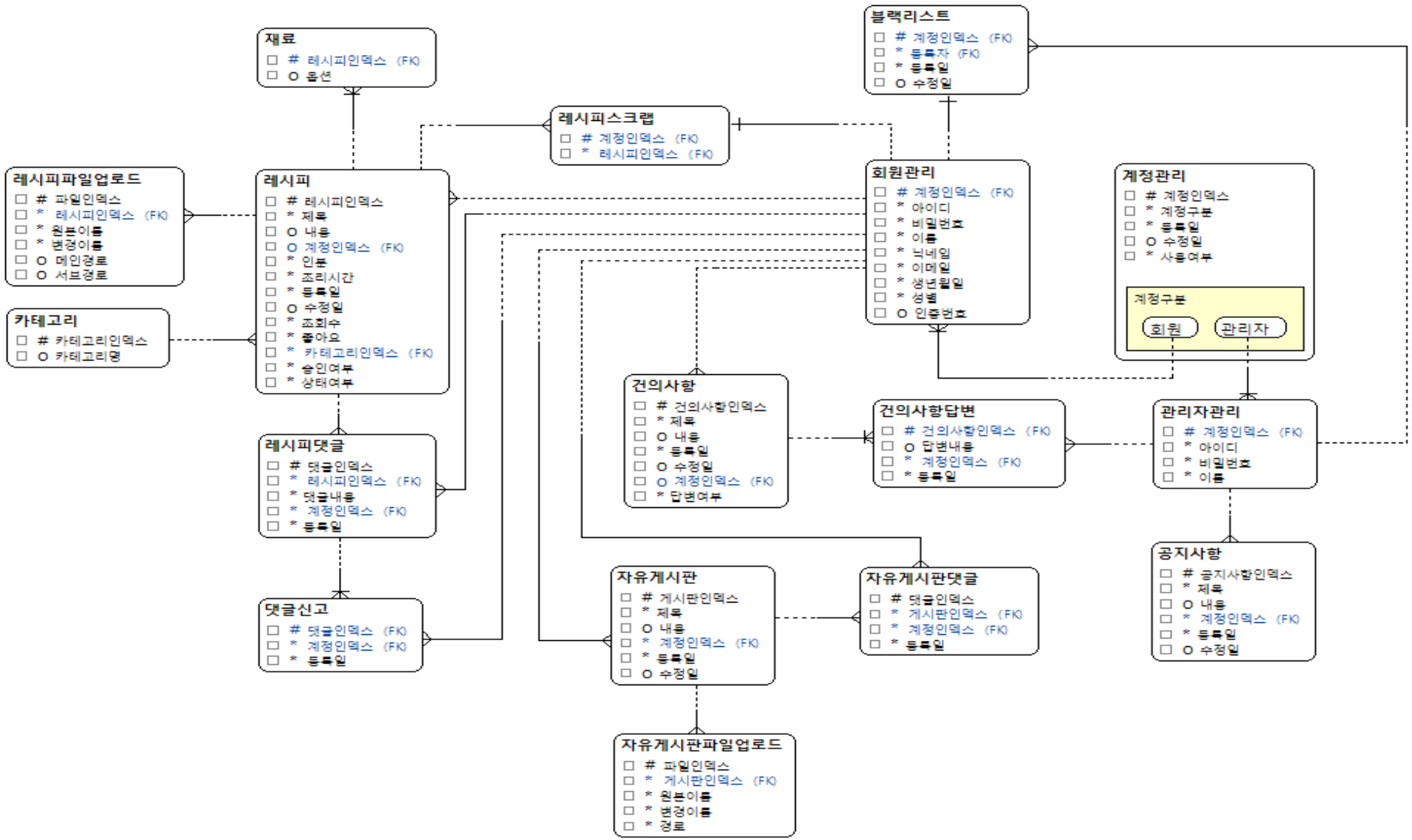
오늘

일	월	화	수	목	금	토
26	27	28	29	30	7월 1일	2 제목 없음
3	4	5	6 첫 정식모임	7 수업시간 회의	8	9
10	11	12	13 UML 완료 조 모임	14 휴강	15 데이터모델링 1차	16
17	18 휴강	19	20 조모임	21 코드작성시작 예정일	22 데이터모델링 완료 데이터 삽입 시작	23
24	25 이클립스 팀 프로젝트 시작	26	27 조모임	28	29	30
31	8월 1일	2	3	4	5	6 마지막 테스트

유스케이스 모델


홈밥
검색
좋아요
댓글신고
레시피 게시물
레시피댓글등록
댓글수정
댓글삭제
레시피
레시피 수정
레시피 등록
레시피삭제
건의 답변
건의 사항
공지사항삭제
공지사항작성
공지사항
공지사항수정
댓글 신고
자유게시판댓글
자유게시판
게시글 등록
게시글 수정
게시글 삭제
수정
삭제
조회
블랙리스트
회원정보 삭제
회원정보 수정
회원관리
건의 수정
건의사항
건의 등록
건의 삭제
레시피 반려
레시피 승인
레시피 관리
아이디 찾기
비밀번호 찾기
레시피 스크랩
회원가입
로그인
마이페이지
회원정보수정
나의 게시물
사용자
관리자
<<extend>>
<<include>>

ERD(논리 데이터 베이스 모델)



메인페이지 - 추천수 TOP5 게시글

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        RecipeService recipeService = new RecipeService();

        List<RecipeDTO> selectLikeRecipeList = recipeService.selectLikeRecipeList();

        for(RecipeDTO recipe : selectLikeRecipeList) {

            System.out.println("테스트 List : " + recipe);

        }

        String path = "";

        if(selectLikeRecipeList != null) {

            path = "/WEB-INF/views/main/main.jsp";

            request.setAttribute("selectLikeRecipeList", selectLikeRecipeList);

        }

        request.getRequestDispatcher(path).forward(request, response);

    }
```

```xml
<!-- 메인페이지 추천레시피 추천 수 높은 5개 게시글 -->
    <select id="seleteLikeRecipeList" resultMap="recipeResultMap">
        SELECT
                A.TITLE
              , A.LIKE_CNT
              , A.MEMBER_IDX
              , A.RCP_IDX
              , A.STATE_YN
              , B.MEMBER_IDX
              , C.SUB_PATH
              , C.RCP_IDX
          FROM RECIPE A
          JOIN USERS_MNG B ON (A.MEMBER_IDX = B.MEMBER_IDX)
          JOIN RCP_FILEUPLOAD C ON (A.RCP_IDX = C.RCP_IDX)
         WHERE A.STATE_YN = 'Y'
         <![CDATA[
           AND ROWNUM <= 5
           ]]>
         ORDER BY A.LIKE_CNT DESC
    </select>
```

- 메인페이지에 레시피 게시글 추천수가 높은 순서대로 5개를 조회하여 출력 하였습니다

레시피 – 게시글 조회

```java
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        String currentPage = request.getParameter("currentPage");

        int pageNo = 1;

        if(currentPage != null && !"".equals(currentPage)) {

            pageNo = Integer.parseInt(currentPage);
        }

        if(pageNo <= 0) {

            pageNo = 1;
        }

        RecipeService recipeService = new RecipeService();

        int totalCount = recipeService.selectTotalCount();

        System.out.println("totalBoardCount : " + totalCount);

        int limit = 8;
        int buttonAmount = 5;

        SelectCriteria selectCriteria = null;

        selectCriteria = Pagenation.getSelectCriteria(pageNo, totalCount, limit, buttonAmount);
        System.out.println("selectCriteria"+selectCriteria);

        List<RecipeDTO> recipeList = recipeService.selectRecipeList(selectCriteria);
        System.out.println("recipeList : " + recipeList);

        for (RecipeDTO recipe : recipeList) {

            System.out.println(recipe);
        }

        String path = "";

        if(recipeList != null) {

            path = "/WEB-INF/views/users/recipe/recipePage.jsp";

            request.setAttribute("currentPage", currentPage);
            request.setAttribute("recipeList", recipeList);
            request.setAttribute("selectCriteria", selectCriteria);
        } else {

            path = "/WEB-INF/views/common/failed.jsp";

            request.setAttribute("message", "게시판 조회 실패!");
        }

        request.getRequestDispatcher(path).forward(request, response);

    }
```

```xml
<!-- 레시피 페이지 전체 리스트 조회 -->
    <select id="selectRecipeList" resultMap="recipeResultMap">
     SELECT
            A.RNUM
          , A.RCP_IDX
          , A.TITLE
          , A.VIEW_CNT
          , A.LIKE_CNT
          , A.CTGR_IDX
          , E.NICKNAME
          , F.SUB_PATH
          , F.RCP_IDX
          , D.CTGR_TIT
       FROM (SELECT ROWNUM RNUM
                  , B.RCP_IDX
                  , B.TITLE
                  , B.VIEW_CNT
                  , B.LIKE_CNT
                  , B.CTGR_IDX
                  , B.MEMBER_IDX
        FROM (SELECT C.RCP_IDX
                   , C.TITLE
                   , C.VIEW_CNT
                   , C.LIKE_CNT
                   , C.CTGR_IDX
                   , C.MEMBER_IDX
              FROM RECIPE C
             WHERE C.STATE_YN = 'Y'
                ORDER BY C.RCP_IDX DESC
              )B
                <![CDATA[
               WHERE ROWNUM <= #{ endRow }
               ]]>
         )A
         JOIN CATEGORY D ON (A.CTGR_IDX = D.CTGR_IDX)
         JOIN USERS_MNG  ON(A.MEMBER_IDX = E.MEMBER_IDX)
         JOIN RCP_FILEUPLOAD F ON(A.RCP_IDX = F.RCP_IDX)
        WHERE A.RNUM >= #{ startRow }
        ORDER BY A.RCP_IDX DESC
    </select>
```

- 전체 레시피 게시글을 조회하여 한 페이지당 조회 개수를 8개로 설정하였고, 개수 초과 시 다음 페이지로 이동하게 만들었습니다.

레시피 – 키워드 및 카테고리별 게시물 검색

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        String keyword = request.getParameter("keyword");
        String type = request.getParameter("type");

        System.out.println("테스트 keyword : " + keyword + " 테스트 type : " + type);

        RecipeService recipeService = new RecipeService();
        List<RecipeDTO> recipe = recipeService.searchRecipe(keyword, type);

        for(RecipeDTO search : recipe) {

            System.out.println("값 확인 " + search);
        }

        request.setAttribute("recipe", recipe);
        request.setAttribute("keyword", keyword);

        request.getRequestDispatcher("/WEB-INF/views/users/recipe/searchRecipe.jsp").forward(request,
response);
    }
```

```xml
<!-- 레시피 검색  -->
    <select id="searchRecipe" resultMap="recipeResultMap">
        SELECT
                A.RCP_IDX
              , A.TITLE
              , A.VIEW_CNT
              , A.LIKE_CNT
              , A.CTGR_IDX
              , B.NICKNAME
              , C.SUB_PATH
              , C.RCP_IDX
              , D.CTGR_TIT
        FROM RECIPE A
        JOIN USERS_MNG B ON (A.MEMBER_IDX = B.MEMBER_IDX)
        JOIN RCP_FILEUPLOAD C ON (A.RCP_IDX = C.RCP_IDX)
        JOIN CATEGORY D ON (A.CTGR_IDX = D.CTGR_IDX)
       <where>
        <if test="type.equals('title')">
             A.TITLE LIKE '%'||#{keyword}||'%'
        </if>
        <if test="type.equals('writer')">
             B.NICKNAME LIKE '%'||#{keyword}||'%'
        </if>
        <if test="type.equals('category')">
             D.CTGR_TI LIKE '%'||#{keyword}||'%'
        </if>
         AND A.STATE_YN = 'Y'
       </where>
       ORDER BY A.RCP_IDX DESC
    </select>
```

- 원하는 키워드 및 카테고리별로 레시피 게시물을 검색할 수 있습니다.

레시피 – 레시피 게시물 상세페이지(1 / 3)

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        RecipeService service = new RecipeService();

        int rcpIdx = Integer.parseInt(request.getParameter("rcpIdx"));

        RecipeDTO recipe =  new RecipeService().selectOneRecipeWriting(rcpIdx);

        String currentPage = request.getParameter("currentPage");
        int pageNo = 1;

        if(currentPage != null && !"".equals(currentPage)) {
            pageNo = Integer.parseInt(currentPage);
        }

        if(pageNo <= 0) {
            pageNo = 1;
        }


        int totalCount = service.recipeCommentsCount(rcpIdx);

        System.out.println("totalCount : "+totalCount);

        int limit = 99;
        int buttonAmount = 5;

        SelectCriteria selectCriteria = null;

        selectCriteria = Pagenation.getSelectCriteria(pageNo, totalCount, limit, buttonAmount);

        System.out.println("selectCriteria :" + selectCriteria);

        String rcpIdx2 = request.getParameter("rcpIdx");

        Map<Object, Object> commentsMap = new HashMap<>();
        commentsMap.put("rcpIdx2", rcpIdx2);
        commentsMap.put("selectCriteria", selectCriteria);

        System.out.println("commentsMap : " + commentsMap);

        List<RecipeCommentsDTO> recipeCommentsList = service.recipeCommentsSelect(commentsMap);

        System.out.println("recipeCommentsList : " + recipeCommentsList);
        System.out.println("selectCriteria : " + selectCriteria);

        String path = "";

        System.out.println("currentPage2 : " + currentPage);

        Object session = request.getSession().getAttribute("loginUser");

        System.out.println("session : " + session);

        List<RecipeLikeDTO> recipeLike = new ArrayList<RecipeLikeDTO>();

        if(session != null) {

            int memberIdx = ((UsersMngDTO)
request.getSession().getAttribute("loginUser")).getMemberIdx();
            System.out.println("memberIdx : " + memberIdx);

            recipeLike = service.selectRecipeLike(rcpIdx, memberIdx);

        } else {

            recipeLike = service.selectRecipeLike2(rcpIdx);
        }

        if(recipeCommentsList != null) {

            path = "/WEB-INF/views/users/recipe/recipeComments.jsp";

            request.seAttribute("selectCriteria", selectCriteria);
            request.setAttribute("recipeCommentsList", recipeCommentsList);
            request.setAttribute("recipe", recipe);
            request.setAttribute("recipeLike", recipeLike);

        } else {

            path = "/WEB-INF/views/common/failed.jsp";

            request.setAttribute("message", "실패!");

        }

        request.getRequestDispatcher(path).forward(request, response);
    }
```

```xml
<!-- 조회수 증가 -->
    <update id="incrementRecipeCount">
      UPDATE
             RECIPE A
         SET A.VIEW_CNT = (SELECT B.VIEW_CNT
                             FROM RECIPE B
                            WHERE B.RCP_IDX = #{ rcpIdx }
                           ) + 1
       WHERE A.RCP_IDX = #{ rcpIdx }
    </update>

    <!-- 추천 클릭시 한 계정 중복추천을 방지하기 위하여 추천 테이블에 추가 -->
    <insert id="likeCountRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeLikeDTO">
      INSERT
        INTO RCP_LIKE
      (
        RCP_IDX, MEMBER_IDX, STATE_YN
      )
      VALUES
      (
        #{ rcpIdx }, #{ memberIdx }, 'Y'
      )
    </insert>

<!-- 레시피 게시물 상세조회 -->
    <select id="selectOneRecipeWriting" resultMap="recipeResultMap">
      SELECT
             A.RCP_IDX
           , A.TITLE
           , A.CONTENT
           , A.REG_DATE
           , A.MOD_DATE
           , A.VIEW_CNT
           , A.LIKE_CNT
           , A.CTGR_IDX
           , B.CTGR_TIT
           , A.APR_STATE
           , A.STATE_YN
           , A.SERVING
           , A.COOKING_TIME
           , C.SUB_PATH
           , D.NICKNAME
           , A.MEMBER_IDX
           , E.STATE_YN
        FROM RECIPE A
        JOIN CATEGORY B ON (A.CTGR_IDX = B.CTGR_IDX)
        JOIN RCP_FILEUPLOAD C ON (A.RCP_IDX = C.RCP_IDX)
        JOIN USERS_MNG D ON (A.MEMBER_IDX = D.MEMBER_IDX)
        LEFT JOIN RCP_LIKE E ON (A.MEMBER_IDX = E.MEMBER_IDX)
       WHERE A.RCP_IDX = #{ rcpIdx }
         AND A.STATE_YN = 'Y'
    </select>
```

- 레시피 게시글 상세페이지 조회 및 클릭 시 조회수가 +1 증가하도록 만들었습니다.

레시피 – 레시피 게시물 상세페이지(2 / 3)

```java
// 댓글 삭제
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        String parameter = request.getParameter("rcpIdx");

        System.out.println("parameter : " + parameter);

        String[] arrIdx = parameter.split("\\?");

        String rcpIdx = arrIdx[0];

        String paramMemberIdx = arrIdx[1];
        String[] arrMemberIdx = paramMemberIdx.split("\\=");
        String memberIdx = arrMemberIdx[1];

        String paramCmtIdx = arrIdx[2];
        String[] arrCmtIdx = paramCmtIdx.split("\\=");
        String cmtIdx = arrCmtIdx[1];

        System.out.println("rcpIdx " + rcpIdx);
        System.out.println("memberIdx " + memberIdx);
        System.out.println("cmtIdx " + cmtIdx);

        Map<String, String> delete = new HashMap<>();
        delete.put("rcpIdx", rcpIdx);
        delete.put("memberIdx", memberIdx);
        delete.put("cmtIdx", cmtIdx);

        System.out.println("delete : " + delete);

        RecipeCommentsService service = new RecipeCommentsService();

        int result = service.commentsDelete(delete);

        if(result > 0) {

            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        } else {

            JOptionPane.showMessageDialog(null, "실패하였습니다.", "알림", JOptionPane.INFORMATION_MESSAGE);
            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        }

    }

// 댓글 신고
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        String memberIdx = request.getParameter("memberIdx");
        String cmtIdx = request.getParameter("cmtIdx");
        String rcpIdx = request.getParameter("rcpIdx");

        System.out.println("cmtIdx : " + cmtIdx);
        System.out.println("memberIdx : " + memberIdx);
        System.out.println("rcpIdx : " + rcpIdx);

        Map<String, String> siren = new HashMap<>();
        siren.put("cmtIdx", cmtIdx);
        siren.put("memberIdx", memberIdx);

        RecipeCommentsService service = new RecipeCommentsService();

        int result = service.insertSiren(siren);

        if(result> 0) {

            JOptionPane.showMessageDialog(null, "신고 완료되었습니다.");
            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        } else {

            JOptionPane.showMessageDialog(null, "실패하였습니다.", "알림", JOptionPane.INFORMATION_MESSAGE);
            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        }

    }
```

```java
// 댓글 등록
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        String cmtContent = request.getParameter("cmtContent");
        String rcpIdx = request.getParameter("rcpIdx");

        HttpSession session = request.getSession();
        UsersMngDTO loginUser = (UsersMngDTO) session.getAttribute("loginUser");

        int memberIdx = loginUser.getMemberIdx();

        System.out.println("memberIdx : " + memberIdx);
        System.out.println("rcpIdx : " + rcpIdx);

        RecipeCommentsService service = new RecipeCommentsService();

        Map<String, Object> commentInsert = new HashMap<>();
        commentInsert.put("cmtContent", cmtContent);
        commentInsert.put("memberIdx", memberIdx);
        commentInsert.put("rcpIdx", rcpIdx);


        int result = service.commentsInsert(commentInsert);

        if(result > 0) {

            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        } else {

            JOptionPane.showMessageDialog(null, "실패하였습니다.", "알림", JOptionPane.INFORMATION_MESSAGE);
            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        }

    }
```

```xml
<!-- 레시피 게시물 댓글 등록 -->
    <insert id="commentsInsert">
        INSERT
          INTO RCP_COMMENTS A
        (
          A.CMT_IDX
        , A.CMT_CONTENT
        , A.REG_DATE
        , A.MEMBER_IDX
        , A.RCP_IDX
        )
        VALUES
        (
          SEQ_CMT_IDX.NEXTVAL
        , #{ cmtContent }
        , SYSDATE
        , #{ memberIdx }
        , #{ rcpIdx }
        )
    </insert>

    <!-- 레시피 게시물 댓글 삭제 -->
    <delete id="commentsDelete">
        DELETE
          FROM RCP_COMMENTS
         WHERE MEMBER_IDX = #{ memberIdx }
           AND RCP_IDX = #{ rcpIdx }
           AND CMT_IDX = #{ cmtIdx }
    </delete>

    <!-- 레시피 게시물 댓글 신고 -->
    <insert id="insertSiren">
        INSERT
          INTO CMT_REPORT A
        (
          A.CMT_IDX
        , A.MEMBER_IDX
        , A.REG_DATE
        , A.STATE_YN
        )
        VALUES
        (
          #{ cmtIdx }
        , #{ memberIdx }
        , SYSDATE
        ,'Y'
        )
    </insert>
```

- 레시피 게시글 상세페이지 내 에서 댓글 등록, 삭제, 신고를 할 수 있도록 만들었습니다.

레시피 – 레시피 게시물 상세페이지(3 / 3)

```java
// 추천 등록
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        int rcpIdx = Integer.parseInt(request.getParameter("rcpIdx"));
        int memberIdx = Integer.parseInt(request.getParameter("memberIdx"));

        RecipeLikeDTO recipeLike = new RecipeLikeDTO();

        recipeLike.setRcpIdx(rcpIdx);
        recipeLike.setMemberIdx(memberIdx);

        RecipeService recipeService = new RecipeService();
        int recipe = recipeService.likeCountRecipe(rcpIdx, memberIdx);

        if(recipe > 0) {

            request.setAttribute("recipeLike", recipeLike);
            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        }

    }

// 추천 취소
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        int rcpIdx = Integer.parseInt(request.getParameter("rcpIdx"));
        int memberIdx = Integer.parseInt(request.getParameter("memberIdx"));

        RecipeLikeDTO recipeLike = new RecipeLikeDTO();

        recipeLike.setRcpIdx(rcpIdx);
        recipeLike.setMemberIdx(memberIdx);

        System.out.println("추천 취소 눌렀을 때 테스트입니다 테스트 : " + recipeLike);

        RecipeService recipeService = new RecipeService();
        int recipe = recipeService.likeCancelRecipe(rcpIdx, memberIdx);

        if(recipe > 0) {

            response.sendRedirect(request.getContextPath()+"/recipeComments/view?rcpIdx="+rcpIdx);
        }

    }
```

```xml
<!-- 추천 클릭시 한 계정 중복추천을 방지하기 위하여 추천 테이블에 추가 -->
    <insert id="likeCountRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeLikeDTO">
      INSERT
        INTO RCP_LIKE
      (
        RCP_IDX, MEMBER_IDX, STATE_YN
      )
      VALUES
      (
        #{ rcpIdx }, #{ memberIdx }, 'Y'
      )
    </insert>

    <!-- 추천 클릭시 추천수 증가 -->
    <update id="likeCountRecipe2" parameterType="com.homebab.users.recipe.model.dto.RecipeDTO">
      UPDATE
            RECIPE A
         SET A.LIKE_CNT = (SELECT B.LIKE_CNT
                             FROM RECIPE B
                            WHERE B.RCP_IDX = #{ rcpIdx }
                           ) + 1
       WHERE A.RCP_IDX = #{ rcpIdx }
    </update>

    <!-- 추천 클릭시 추천 테이블 내 정보 삭제 -->
    <delete id="deleteTableRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeLikeDTO">
      DELETE
        FROM RCP_LIKE
       WHERE RCP_IDX = #{ rcpIdx }
    </delete>

    <!-- 추천 클릭시 추천수 감소 -->
    <update id="likeCencelRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeLikeDTO">
        UPDATE
              RECIPE A
           SET A.LIKE_CNT = (SELECT B.LIKE_CNT
                               FROM RECIPE B
                              WHERE B.RCP_IDX = #{ rcpIdx }
                             ) - 1
         WHERE A.RCP_IDX = #{ rcpIdx }
    </update>

    <!-- 추천 테이블 조회 -->
    <select id="selectRecipeLike" resultMap="recipeLikeResultMap">
      SELECT
            STATE_YN
        FROM RCP_LIKE
       WHERE MEMBER_IDX = #{ memberIdx }
         AND RCP_IDX = #{ rcpIdx }
    </select>
```

- 레시피 게시글 상세페이지 내 에서 추천 등록 및 취소를 할 수 있도록 만들었습니다.

레시피 – 게시글 등록

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        request.getRequestDispatcher("/WEB-INF/views/users/recipe/insertRecipe.jsp").forward(request,
response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        if(ServletFileUpload.isMultipartContent(request)) {

            String rootLocation = getServletContext().getRealPath("/");
            int maxFileSize = 1024 * 1024 * 10;

            System.out.println("파일 저장 ROOT 경로 : " + rootLocation);

            String fileUploadDirectory = rootLocation + "/upload/recipe/original/";
            String thumbnailDirectory = rootLocation + "/upload/recipe/sub/";

            File directory = new File(fileUploadDirectory);
            File directory2 = new File(thumbnailDirectory);

            directory.mkdirs();
            directory2.mkdirs();

            Map<String, String> parameter = new HashMap<>();
            List<Map<String, String>> fileList = new ArrayList<>();

            DiskFileItemFactory fileItemFactory = new DiskFileItemFactory();
            fileItemFactory.setRepository(new File(fileUploadDirectory));
            fileItemFactory.setSizeThreshold(maxFileSize);

            ServletFileUpload fileUpload = new ServletFileUpload(fileItemFactory);

            try {
                List<FileItem> fileItems = fileUpload.parseRequest(request);

                for(int i = 0; i < fileItems.size(); i++) {

                    FileItem item = fileItems.get(i);

                    if(!item.isFormField()) {

                        if(item.getSize() > 0) {

                            String filedName = item.getFieldName();
                            String originFileName = item.getName();

                            int dot = originFileName.lastIndexOf(".");
                            String ext = originFileName.substring(dot);

                            String randomFileName = UUID.randomUUID().toString().replace("-", "") +
ext;


                            File storeFile = new File(fileUploadDirectory + "/" + randomFileName);

                            item.write(storeFile);

                            Map<String, String> fileMap = new HashMap<>();

                            fileMap.put("filedName", filedName);
                            fileMap.put("originFileName", originFileName);
                            fileMap.put("savedFileName", randomFileName);
                            fileMap.put("savePath", fileUploadDirectory);

                            fileList.add(fileMap);

                            int width = 0;
                            int height = 0;
                            if("recipeKeyVisual".equals(filedName)) {

                                width = 800;
                                height = 400;
                            }

                            Thumbnails.of(fileUploadDirectory + randomFileName)
                                    .size(width, height)
                                    .toFile(thumbnailDirectory + "recipe_" + randomFileName);

                            fileMap.put("recipePath", "/upload/recipe/sub/recipe_" + randomFileName);

                            fileList.add(fileMap);

                        }

                    } else {

                        parameter.put(item.getFieldName(), new String(item.getString().getBytes("ISO-
8859-1"), "UTF-8"));
```

```
System.out.println("fileList : " + fileList);
                System.out.println("parameter : " + parameter);

                int memberIdx = ((UsersMngDTO)
request.getSession().getAttribute("loginUser")).getMemberIdx();

                RecipeDTO recipe = new RecipeDTO();

                recipe.setTitle(parameter.get("recipeTitle"));
                recipe.setContent(parameter.get("content"));
                recipe.setMemberIdx(memberIdx);
                recipe.setCtgrIdx(Integer.parseInt(parameter.get("category")));
                recipe.setServing(Integer.parseInt(parameter.get("recipeServing")));
                recipe.setCookingTime(Integer.parseInt(parameter.get("cookingTime")));

                recipe.setFileList(new ArrayList<RecipeFileUploadDTO>());
                List<RecipeFileUploadDTO> list = recipe.getFileList();
                for(int i = 0; i < fileList.size(); i++) {
                    Map<String, String> file = fileList.get(i);

                    RecipeFileUploadDTO FileInfo = new RecipeFileUploadDTO();
                    FileInfo.setOrgName(file.get("originFileName"));
                    FileInfo.setChangeFileName(file.get("savedFileName"));
                    FileInfo.setMainPaht(file.get("savePath"));
                    FileInfo.setSubPath(file.get("recipePath"));

                    list.add(FileInfo);
                }
                RecipeService recipeService = new RecipeService();

                int result = recipeService.insertRecipe(recipe);

                String path = "";

                if (result > 0) {

                    path = "/WEB-INF/views/common/success.jsp";
                    request.setAttribute("successCode", "insertRecipe");
                } else {

                    path = "/WEB-INF/views/common/failed.jsp";
                    request.setAttribute("message", "레시피 게시판 작성에 실패하셨습니다.");
                }

                 request.getRequestDispatcher(path).forward(request, response);

            } catch (Exception e) {

                int cnt = 0;
                for(int i = 0; i < fileList.size(); i++) {

                    Map<String, String> file = fileList.get(i);

                    File deleteFile = new File(fileUploadDirectory + "/" + file.get("savedFileName"));
                    boolean isDeleted = deleteFile.delete();

                    if(isDeleted) {
                        cnt++;
                    }
                }
                +
                if(cnt == fileList.size()) {

                    e.printStackTrace();
                    System.out.println("업로드에 실패한 모든 사진 삭제 완료!");
                } else {

                    e.printStackTrace();
                }
```

```
<!-- 레시피 게시글 등록 -->
    <insert id="insertRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeDTO">
        INSERT
          INTO RECIPE
        (
          RCP_IDX, TITLE, CONTENT, MEMBER_IDX
        , REG_DATE, VIEW_CNT, LIKE_CNT, CTGR_IDX
        , APR_STATE, STATE_YN, SERVING, COOKING_TIME
        )
        VALUES
        (
          SEQ_RECIPE.NEXTVAL, #{ title }, #{ content }, #{ memberIdx }
        , TO_CHAR(SYSDATE,'YYYY.MM.DD HH24:MI'), 1, 0, #{ ctgrIdx }
        , DEFAULT, 'Y', #{ serving }, #{ cookingTime }
        )
    </insert>

    <!-- 레시피 게시글 내 파일 업로드 시 저장 -->
    <insert id="insertFileUpload">
        INSERT
          INTO RCP_FILEUPLOAD
        (
          FILE_IDX, ORGNAME, CHANGE_FILENAME
        , MAIN_PATH, SUB_PATH, RCP_IDX
        )
        VALUES

          SEQ_RCP_FILEUPLOAD.NEXTVAL, #{ orgName }, #{ changeFileName }
        , #{ mainPaht }, #{ subPath }, SEQ_RECIPE.CURRVAL
        )
    </insert>
```

- 레시피 게시물을 작성할 수 있고 첨부파일 업로드가 가능합니다

레시피 – 게시글 수정

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        int rcpIdx = Integer.parseInt(request.getParameter("rcpIdx"));

        RecipeService recipeService = new RecipeService();
        RecipeDTO recipe = recipeService.updateSelectRecipe(rcpIdx);

        System.out.println(recipe);

        String path = "";
        if(recipe != null) {

            System.out.println(recipe);
            path = "/WEB-INF/views/users/recipe/updateRecipe.jsp";
            request.setAttribute("recipe", recipe);


        } else {
            System.out.println(recipe);
            path = "/WEB-INF/views/common/failed.jsp";
            request.setAttribute("message", "레시피 게시물 수정 실패!");


        request.getRequestDispatcher(path).forward(request, response);
    }
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        Map<String, String> fileMap = new HashMap<>();
        Map<String, String> parameter = new HashMap<>();
        List<Map<String, String>> fileList = new ArrayList<>();

        if(ServletFileUpload.isMultipartContent(request)) {

            String rootLocation = getServletContext().getRealPath("/");
            int maxFileSize = 1024 * 1024 * 10;

            System.out.println("현재 루트 ROOT 위치 : " + rootLocation);

            String fileUploadDirectory = rootLocation + "/upload/recipe/original/";
            String thumbnailDirectory = rootLocation + "/upload/recipe/sub/";

            File directory = new File(fileUploadDirectory);
            File directory2 = new File(thumbnailDirectory);

            directory.mkdirs();
            directory2.mkdirs();

            DiskFileItemFactory fileItemFactory = new DiskFileItemFactory();
            fileItemFactory.setRepository(new File(fileUploadDirectory));
            fileItemFactory.setSizeThreshold(maxFileSize);

            ServletFileUpload fileUpload = new ServletFileUpload(fileItemFactory);

            try {
                List<FileItem> fileItems = fileUpload.parseRequest(request);

                for(int i = 0; i < fileItems.size(); i++) {

                    FileItem item = fileItems.get(i);

                    if(item.isFormField()) {

                        if(item.getSize() > 0) {

                            String fieldName = item.getFieldName();
                            String originFileName = item.getName();

                            int dot = originFileName.lastIndexOf(".");
                            String ext = originFileName.substring(dot);

                            String randomFileName = UUID.randomUUID().toString().replace("-", "") +
ext;

                            File storedFile = new File(fileUploadDirectory + "/" + randomFileName);

                            item.write(storedFile);

                            fileMap.put("fieldName", fieldName);
                            fileMap.put("originFileName", originFileName);
                            fileMap.put("savedFileName", randomFileName);
                            fileMap.put("savePath", fileUploadDirectory);

                            fileList.add(fileMap);

                            int width = 0;
                            int height = 0;
                            if("recipeKeyVisual".equals(fieldName)) {

                                width = 900;
                                height = 400;
                            }

                            Thumbnails.of(fileUploadDirectory + randomFileName)
                                      .size(width, height)
                                      .toFile(thumbnailDirectory + "recipe_" + randomFileName);

                            fileMap.put("recipePath", "/upload/recipe/sub/recipe_" + randomFileName);

                            fileList.add(fileMap);

                        }

                        } else {

                            parameter.put(item.getFieldName(), new String(item.getString().getBytes("ISO-
8859-1"), "UTF-8"));
                        }
                    }

                    RecipeDTO recipe = new RecipeDTO();

                    System.out.println("fileList : " + fileList);
                    System.out.println("parameter : " + parameter);

                    int memberIdx = ((UsersMngDTO)
request.getSession().getAttribute("loginUser")).getMemberIdx();

                    recipe.setTitle(parameter.get("recipeTitle"));
                    recipe.setContent(parameter.get("content"));
                    recipe.setMemberIdx(memberIdx);
                    recipe.setCtgrIdx(Integer.parseInt(parameter.get("category")));
                    recipe.setServing(Integer.parseInt(parameter.get("recipeServing")));
                    recipe.setCookingTime(Integer.parseInt(parameter.get("cookingTime")));
                    recipe.setRcpIdx(Integer.parseInt(parameter.get("rcpIdx")));

                    int rcpIdx = Integer.parseInt(parameter.get("rcpIdx"));

                    recipe.setFileList(new ArrayList<RecipeFileUploadDTO>());
                    List<RecipeFileUploadDTO> list = recipe.getFileList();
                    for(int i = 0; i < fileList.size(); i++) {
                        Map<String, String> file = fileList.get(i);

                        RecipeFileUploadDTO fileInfo = new RecipeFileUploadDTO();
                        fileInfo.setOrgName(file.get("originFileName"));
                        fileInfo.setChangeFileName(file.get("savedFileName"));
                        fileInfo.setMainPath(file.get("savePath"));
                        fileInfo.setSubPath(file.get("recUpPath"));

                        list.add(fileInfo);
                    }

                    RecipeService recipeService = new RecipeService();

                    int result = recipeService.updateRecipe(recipe, rcpIdx);

                    String path = "";

                    if (result > 0) {

                        path = "/WEB-INF/views/common/success.jsp";
                        request.setAttribute("successCode", "updateRecipe");
                        request.setAttribute("rcpIdx", Integer.parseInt(parameter.get("rcpIdx")));
                    } else {

                        path = "/WEB-INF/views/common/failed.jsp";
                        request.setAttribute("message", "레시피 게시물 수정 실패!");
                    }

                    request.getRequestDispatcher(path).forward(request, response);

                } catch (Exception e) {

                    int cnt = 0;
                    for(int i = 0; i < fileList.size(); i++) {

                        Map<String, String> file = fileList.get(i);

                        File deleteFile = new File(fileUploadDirectory + "/" + file.get("savedFileName"));
                        boolean isDeleted = deleteFile.delete();

                        if(isDeleted) {
                            cnt++;
                        }
                    }

                    if(cnt == fileList.size()) {

                        e.printStackTrace();
                        System.out.println("업로드된 파일을 서버에서 정상적으로 삭제 했습니다.");
                    } else {

                        e.printStackTrace();
                    }
```

```
<!-- 수정 할 레시피 페이지 게시글 조회 -->
    <select id="updateSelectRecipe" resultMap="recipeResultMap">
        SELECT
                A.RCP_IDX
              , A.TITLE
              , A.CONTENT
              , A.REG_DATE
              , A.MOD_DATE
              , A.VIEW_CNT
              , A.LIKE_CNT
              , A.CTGR_IDX
              , B.CTGR_TIT
              , A.APR_STATE
              , A.STATE_YN
              , A.SERVING
              , A.COOKING_TIME
              , C.SUB_PATH
              , D.NICKNAME
              , A.MEMBER_IDX
          FROM RECIPE A
          JOIN CATEGORY B ON (A.CTGR_IDX = B.CTGR_IDX)
          JOIN RCP_FILEUPLOAD C ON (A.RCP_IDX = C.RCP_IDX)
          JOIN USERS_MNG D ON (A.MEMBER_IDX = D.MEMBER_IDX)
         WHERE A.RCP_IDX = #{ rcpIdx }
           AND A.STATE_YN = 'Y'
         ORDER BY A.RCP_IDX DESC
    </select>

<!-- 레시피 게시글 내용 수정 -->
    <update id="updateRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeDTO" >
        UPDATE
                RECIPE
           SET TITLE = #{ title }
             , CONTENT = #{ content }
             , CTGR_IDX = #{ ctgrIdx }
             , SERVING = #{ serving }
             , COOKING_TIME = #{ cookingTime }
             , MOD_DATE = SYSDATE
         WHERE RCP_IDX = #{ rcpIdx }

    </update>

<!-- 레시피 게시글 수정 후 파일 저장 -->
    <insert id="updateInsertFileUpload"
parameterType="com.homebab.users.recipe.model.dto.RecipeFileUploadDTO">
        INSERT
          INTO RCP_FILEUPLOAD
        (
          FILE_IDX, ORGNAME, CHANGE_FILENAME
        , MAIN_PATH, SUB_PATH, RCP_IDX
        )
        VALUES
        (
          SEQ_RCP_FILEUPLOAD.NEXTVAL, #{ orgName }, #{ changeFileName }
        , #{ mainPaht }, #{ subPath }, #{ rcpIdx }
        )
    </insert>
```

- 현재 로그인 세션을 조회하여 로그인한 계정이 등록한 게시물 수정 가능하게 만들었습니다.

레시피 – 게시글 삭제

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        // 게시물 삭제를 위해 해당 게시물 인덱스를 받아옴.
        int rcpIdx = Integer.parseInt(request.getParameter("rcpIdx"));
        RecipeService recipeService = new RecipeService();

        int deleteRecipe = recipeService.deleteRecipe(rcpIdx);

        String path = "";

        if(deleteRecipe > 0) {

            path = "/WEB-INF/views/common/success.jsp";
            request.setAttribute("successCode", "deleteRecipe");
        } else {

            path = "/WEB-INF/views/common/failed.jsp";
            request.setAttribute("message", "레시피 게시물 삭제 실패!");
        }

        request.getRequestDispatcher(path).forward(request, response);
    }
```

```xml
<!-- 레시피 파일 삭제 -->
    <delete id="deleteFile" parameterType="com.homebab.users.recipe.model.dto.RecipeFileUploadDTO">
        DELETE
          FROM RCP_FILEUPLOAD
         WHERE RCP_IDX = #{ rcpIdx }
    </delete>

    <!-- 레시피 게시글 삭제 -->
    <update id="deleteRecipe" parameterType="com.homebab.users.recipe.model.dto.RecipeDTO">
        UPDATE
               RECIPE
           SET STATE_YN = 'N'
         WHERE RCP_IDX = #{ rcpIdx }
    </update>
```

- 현재 로그인 세션을 조회하여 로그인한 계정이 등록한 게시물만 삭제 가능하게 만들었습니다.
- 게시물 내 파일이 존재하면 파일 데이터 먼저 삭제 후 게시글 삭제가 가능합니다.