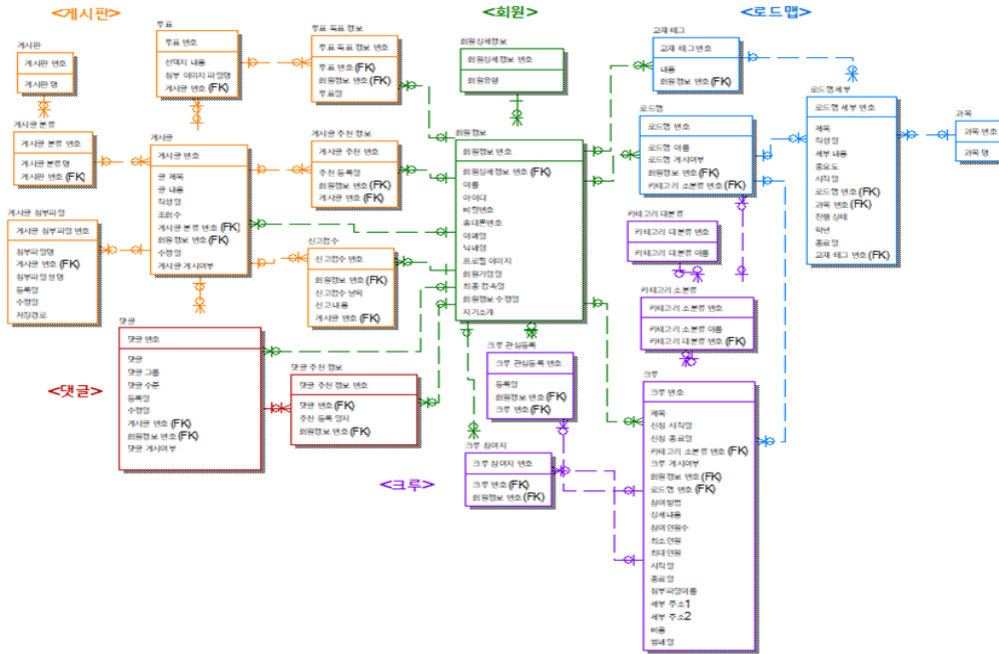


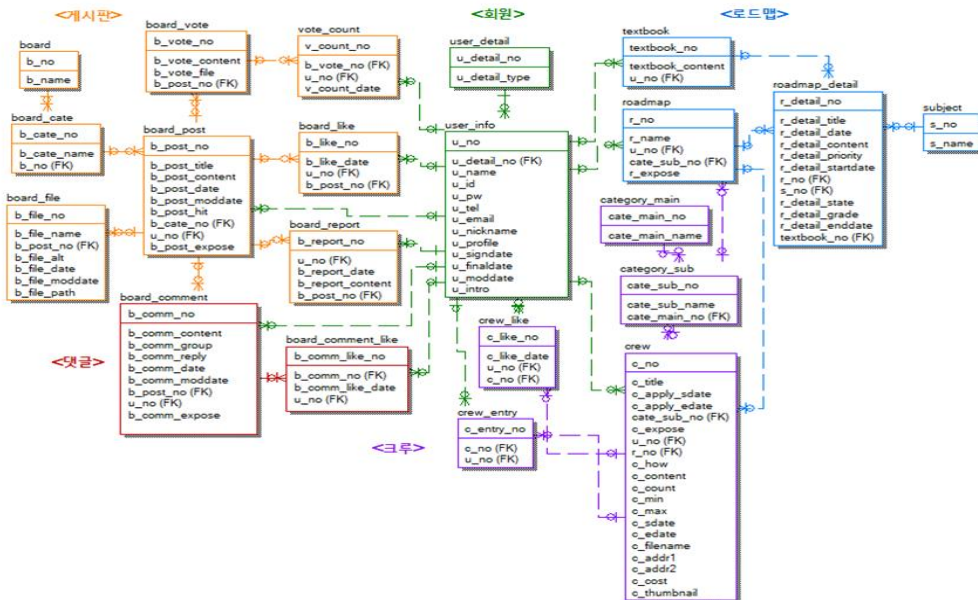
프로젝트 기술서

| | |
|-------------|---|
| 프로젝트 명 | Triple A |
| 포트폴리오 사이트주소 | https://github.com/hyeoni11i/Project.git |
| 개발 기간 | 2021.11.08 ~ 2021.12.06 |
| 투입인원 | 5명 |
| 주요 기술 | <ul style="list-style-type: none">● Spring Boot, Mybatis 사용● Java, MVC2 활용 개발● Ajax 활용 DB 접근● Oracle 11g 활용 기본 CRUD 구현 |
| 리뉴얼 목적 | <ul style="list-style-type: none">● 코로나로 인해 등교하지 못하면서 학생들의 자기주도 학습능력 차이, 또래와의 공감대 형성 부족 등 여러가지 문제가 있다는 것을 기사로 접하고 조금이나마 학습에 도움이 되는 점을 찾다가 기획하게 되었다. |
| 본인의 역할 | 로드맵 |

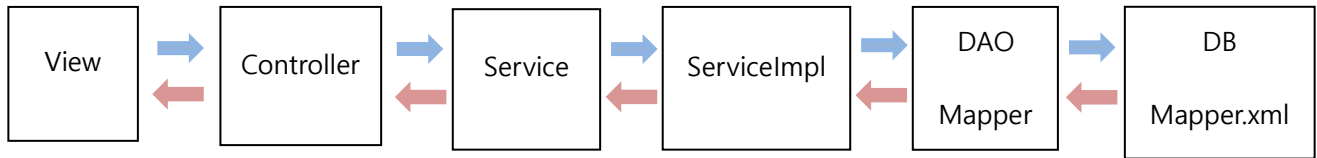
[논리적(Logical) DB 설계]



[물리적(Physical) DB 설계]



[구성 및 흐름]



➤ 나의 로드맵 리스트

1. 내가 생성한 로드맵 출력
2. 로드맵 생성하기
3. 로드맵 클릭 시 해당 세부 로드맵 페이지로 이동

나의 로드맵

| | |
|---------------------------------------|--------------------------------------|
| <div>나의 로드맵</div> <div>로드맵 생성하기</div> | <div>로드맵 생성하기</div> |
| <div>학습</div> <div>2학기 기말고사</div> | <div>학습</div> <div>2학기 중간고사</div> |
| <div>학습</div> <div>2022 수능 대비</div> | <div>공모전</div> <div>디자인 공모전 준비</div> |
| <div>학습</div> <div>1학기 기말고사</div> | <div>공모전</div> <div>동영상 공모전 준비</div> |
| <div>자격증</div> <div>한능검 자격증</div> | <div>학습</div> <div>1학기 중간고사</div> |
| <div>취미 및 기타</div> <div>베이킹 배우기</div> | <div>자격증</div> <div>컴활 자격증</div> |

로드맵 생성

유형

로드맵 이름

[View]

1. Ajax로 저장기능 구현

```
// 로드맵 저장
$(function(){
    $("#save_btn").click(function(){
        var cate_sub_no = $("#cate_sub_no").val();

        if($("#r_name").val() == ""){
            alert ("제목을 입력하세요!");
            return false;
        } else{
            var r_name = $("#r_name").val();
        }

        $.ajax({
            url: "/roadmapwrite",
            type: "post",
            data: {
                "u_no": u_no,
                "cate_sub_no": cate_sub_no,
                "r_name": r_name
            },
            success: function(data){
                var chtml = "";

                for(var i=0; i < data.length; i++){
                    chtml += "<a href='roadmap_detail?u_no="+ data[i].roadmapVo.u_no + "&cate_sub_no="+ data[i].categorySubVo.cate_sub_no + "&cate_sub_name="
                        + data[i].categorySubVo.cate_sub_name + "&r_no=" + data[i].roadmapVo.r_no + "&r_name=" + data[i].roadmapVo.r_name + "'><div class='roadmap'>";
                    if (data[i].categorySubVo.cate_sub_name == '취미 및 기타'){
                        chtml += "<div class='roadmap_cate' style=width:100px> " + data[i].categorySubVo.cate_sub_name + "</div>";
                    } else {
                        chtml += "<div class='roadmap_cate'>" + data[i].categorySubVo.cate_sub_name + "</div>";
                    }
                    chtml += "<div class='roadmap_name'>" + data[i].roadmapVo.r_name + "</div>";
                    chtml += "</div></a>";
                }
                $("#roadmap_list").html(chtml);
                $("#cate_sub_no").val("1");
                $("#r_name").val("");
                $('#myModal').hide();
                location.href = "/my_roadmap";
            },
            error: function(){
                alert("에러!");
            }
        }); //ajax
    }); // save btn
});

<div id="list_box">
    <div id="category">
        <div id="list_cate">
            <ul class="noticeTab">
                <li><a class="${ cate == 0 ? 'active' : '' }" href="/my_roadmap">전체</a></li>
                <li><a class="${ cate == 1 ? 'active' : '' }" href="/my_roadmap?cate_sub_no=1">학습</a></li>
                <li><a class="${ cate == 3 ? 'active' : '' }" href="/my_roadmap?cate_sub_no=3">자격증</a></li>
                <li><a class="${ cate == 5 ? 'active' : '' }" href="/my_roadmap?cate_sub_no=5">공모전</a></li>
                <li><a class="${ cate == 21 ? 'active' : '' }" href="/my_roadmap?cate_sub_no=21">취미 및 기타</a></li>
            </ul>
        </div>
        <div id="cate_button">
            <button id="add_roadmap" onclick="new_Roadmap()"><p>로드맵 생성하기</p></button>
        </div>
    </div>
</div>

<div class="roadmap_list">
    <c:forEach items="${list}" var="roadmapCateSubVo">
        <a href="roadmap_detail?u_no=${roadmapCateSubVo.roadmapVo.u_no}&cate_sub_no=${roadmapCateSubVo.categorySubVo.cate_sub_no}
            +&r_no=${roadmapCateSubVo.roadmapVo.r_no}&r_name=${roadmapCateSubVo.roadmapVo.r_name}">
            <div class="roadmap">
                <c:if test="${roadmapCateSubVo.categorySubVo.cate_sub_name != '취미 및 기타'}">
                    <div class="roadmap_cate">${roadmapCateSubVo.categorySubVo.cate_sub_name}</div>
                </c:if>
                <c:if test="${roadmapCateSubVo.categorySubVo.cate_sub_name == '취미 및 기타'}">
                    <div class="roadmap_cate" style="width: 100px">${roadmapCateSubVo.categorySubVo.cate_sub_name}</div>
                </c:if>
                <div class="roadmap_name">${roadmapCateSubVo.roadmapVo.r_name}</div>
            </div>
        </a>
    </c:forEach>
</div>
</div>
```

[Controller]

```
// 로드맵 리스트
@RequestMapping("/my_roadmap")
public String my_roadmap(RoadmapVo roadmapVo, Model model) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));

    List<RoadmapCateSubVo> list = roadmapListService.roadmapList(roadmapVo);
    model.addAttribute("list", list);
    model.addAttribute("cate", roadmapVo.getCate_sub_no());
    return "roadmap/my_roadmap";
}

// 로드맵 생성하기
@ResponseBody
@RequestMapping("/roadmapwrite")
public Map<String, Object> roadmapWrite(RoadmapVo roadmapVo) {

    int result = roadmapListService.roadmapWrite(roadmapVo);
    //System.out.println("result : " + result);
    //roadmapVo.setCate_sub_no(0);
    List<RoadmapCateSubVo> list = roadmapListService.roadmapList(roadmapVo);
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("list", list);
    map.put("roadmapVo", roadmapVo);

    return map;
}
```

[Service]

```
public interface RoadmapListService {

    // 로드맵 리스트
    List<RoadmapCateSubVo> roadmapList(RoadmapVo roadmapVo);

    // 로드맵 생성하기
    int roadmapWrite(RoadmapVo roadmapVo);
}
```

[ServiceImpl]

```
// 로드맵 리스트
@Override
public List<RoadmapCateSubVo> roadmapList(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));

    // 로드맵 리스트
    List<RoadmapCateSubVo> list = roadmapListMapper.selectRoadmapList(roadmapVo);

    return list;
}

// 로드맵 생성하기
@Override
public int roadmapWrite(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));
    int result = roadmapListMapper.insertRoadmap(roadmapVo);

    return result;
}
```

[Mapper]

```
@Mapper
public interface RoadmapListMapper {

    // 로드맵 리스트
    List<RoadmapCateSubVo> selectRoadmapList(RoadmapVo roadmapVo);

    // 로드맵 생성하기
    int insertRoadmap(RoadmapVo roadmapVo);
}
```

[Mapper.xml]

```
<!-- 로드맵 생성하기 -->
<insert id="insertRoadmap">
    insert into roadmap values (roadmap_seq.nextval, #{r_name}, #{u_no}, #{cate_sub_no}, 'Y')
    <selectKey keyProperty="r_no" resultType="int" order="AFTER">
        select roadmap_seq.currval from dual
    </selectKey>
</insert>

<resultMap id="categorySubVoMap" type="com.site.team1.vo.CategorySubVo">
    <result column="cate_sub_no" property="cate_sub_no" />
    <result column="cate_sub_name" property="cate_sub_name" />
</resultMap>

<resultMap id="roadmapVoMap" type="com.site.team1.vo.RoadmapVo">
    <result column="r_no" property="r_no" />
    <result column="r_name" property="r_name" />
    <result column="u_no" property="u_no" />
</resultMap>

<resultMap id="roadmapCateSubVoMap" type="com.site.team1.vo.RoadmapCateSubVo">
    <collection property="categorySubVo" resultMap="categorySubVoMap"></collection>
    <collection property="roadmapVo" resultMap="roadmapVoMap"></collection>
</resultMap>

<select id="selectRoadmapList" resultMap="roadmapCateSubVoMap">
    select c.cate_sub_no, c.cate_sub_name, r.r_no, r.r_name, r.u_no from category_sub c, roadmap r
    <choose>
        <when test="cate_sub_no == 0">where c.cate_sub_no=r.cate_sub_no and r.cate_sub_no not in('2', '4', '6', '22') and r.expose not in('N') and u_no=#{u_no}</when>
        <otherwise>where c.cate_sub_no=r.cate_sub_no and c.cate_sub_no=#{cate_sub_no} and r.expose not in('N') and u_no=#{u_no}</otherwise>
    </choose>
    order by r_no desc
</select>
```

[DAO]

```
@NoArgsConstructor
@AllArgsConstructor
@Data
public class RoadmapVo {

    private int r_no;
    private String r_name;
    private String r_expose;

    private int u_no;
    private int cate_sub_no;
}

@Data
@NoArgsConstructor
@AllArgsConstructor
public class CategorySubVo {

    private int cate_sub_no;
    private String cate_sub_name;
    private int cate_main_no;
}

@NoArgsConstructor
@AllArgsConstructor
@Data
public class RoadmapCateSubVo {

    private RoadmapVo roadmapVo;
    private CategorySubVo categorySubVo;
}
```

세부 로드맵 페이지

여덟글자 여덟글자 의
로드맵

- 2학기 기말고사
- 2학기 중간고사
- 2022 수능 대비
- 다자인 공모전 준비
- 1학기 기말고사
- 동영상 공모전 준비
- 한글검 자격증
- 1학기 중간고사
- 배어링 배우기
- 검정 자격증
- 1학기 공부계획

+ 로드맵 생성

2학기 중간고사

학습

제목 변경 삭제

사이드바 하단의 + 로드맵생성 버튼을 클릭하여 새 페이지를 추가합니다.
+ 생성 버튼을 클릭하여 세부 로드맵을 추가합니다.
세부 로드맵을 누르면 상세 입력페이지가 열립니다.
세부 로드맵을 클릭한 상태로 다른 곳으로 넘어 올 수 있습니다.

시작 전 2

2학기 중간고사

21-11-26 ~ 21-11-29

수학

김고리

★★★★

진행 중 1

2일차 국어

21-11-27 ~ 21-11-30

국어

자이스토리

★★★★

완료 1

3일차 사탐

21-12-01 ~ 21-12-03

윤여객 사탐

자이스토리

★★★★

상태 없음 1

한국사 만능

21-11-23 ~ 21-12-04

국어

국어개념완성

★★★★

1일차 수학

21-11-25 ~ 21-11-26

수학

자이스토리

★★★★

+ 생성

로드맵 작성

제목

상태 상태 없음

학년 선택 안함

과목 선택 안함

교재 선택 안함 교재 생성

내용

중요도 선택 안함

시작일

종료일

저장

취소

로드맵 수정

제목

상태 진행 중

학년 고등학교 3학년

과목 국어

교재 자이스토리 교재 생성

내용

중요도 ★★★★

시작일

종료일

수정

취소

제목 변경

변경

취소

교재

생성

취소

[View]

```
// 세부 로드맵 저장
$(function(){
    $(".detail_save").click(function(){
        var r_no = $("#param.r_no");
        var r_detail_state = $("#r_detail_state").val();
        var r_detail_grade = $("#r_detail_grade").val();
        var s_no = $("#s_no").val();
        var textbook_no = $("#textbook").val();
        var r_detail_content = $("#r_detail_content").val();
        var r_detail_priority = $("#r_detail_priority").val();
        var r_detail_startdate = convertDateFormat($("#r_detail_startdate").val());
        var r_detail_enddate = convertDateFormat($("#r_detail_enddate").val());

        if($("#r_detail_title").val() == ""){
            alert ("제목을 입력하세요!");
            return false;
        } else{
            var r_detail_title = $("#r_detail_title").val();
        }

        if(r_detail_startdate > r_detail_enddate){
            alert("시작일보다 종료일이 빠릅니다. 다시 선택하세요!");
            $(".date_border").css("outline-color", "red");
            $(".date_border").focus();
            return false;
        }

        $.ajax({
            url: "./detailWrite",
            type: "post",
            data: {
                "r_no": r_no,
                "r_detail_title": r_detail_title,
                "r_detail_state": r_detail_state,
                "r_detail_grade": r_detail_grade,
                "s_no": s_no,
                "textbook_no": textbook_no,
                "r_detail_content": r_detail_content,
                "r_detail_priority": r_detail_priority,
                "r_detail_startdate": r_detail_startdate,
                "r_detail_enddate": r_detail_enddate
            },
            success: function(data){
                var r_detail_startdate = convertDateFormat(data[0].r_detail_startdate).substring(2,10);
                var r_detail_enddate = convertDateFormat(data[0].r_detail_enddate).substring(2,10);
                var chtml="";
                chtml += "<div class='roadmap_content_box' id='"+ data[0].r_detail_no +'>";
                chtml += "<div class='roadmap_content on' id='to-do' draggable='true' ondragstart='onDragStart(event)'>";
                chtml += "<div>";
                chtml += "<div class='pad title'>";
                chtml += "<p><i class='fas fa-file-alt'></i> "+ data[0].r_detail_title + "</p>";
                chtml += "</div>";
                chtml += "<i class='fas fa-trash-alt temp_del' onclick='temp_del(\"+ data[0].r_detail_no +\")'></i>";
                chtml += "</div>";

                if(data[0].r_detail_state == '시작 전'){ $("#todopend").prepend(chtml); }
                if(data[0].r_detail_state == '진행 중'){ $("#doingpend").prepend(chtml); }
                if(data[0].r_detail_state == '완료'){ $("#donepend").prepend(chtml); }
                if(data[0].r_detail_state == '상태 없음'){ $("#trashpend").prepend(chtml); }
                $(".#myModal").hide();
                location.reload(true);
                $("#r_detail_title").val("");
                $("#r_detail_state").val("상태 없음");
                $("#r_detail_grade").val("선택 안함");
                $("#s_no").val("0");
                $("#r_detail_content").val("");
                $("#r_detail_priority").val("선택 안함");
                $("#r_detail_startdate").val("");
                $("#r_detail_enddate").val("");
            },
            error: function(){
                alert("날짜를 입력하세요!");
            }
        });
    });
});
}); // detail_save
}); // function
```


[Controller]

```
// 로드맵 제목 변경
@ResponseBody
@RequestMapping("/modiTitle")
public RoadmapVo modiTitle(RoadmapVo roadmapVo) {
    int result = roadmapDetailService.modiTitle(roadmapVo);

    // 수정된 로드맵 리스트
    RoadmapVo modiRoadmap = roadmapDetailService.modiRoadmapList(roadmapVo);
    return modiRoadmap;
}

// 세부 로드맵 리스트
@RequestMapping("/roadmap_detail")
public Map<String, Object> roadmapDetail(RoadmapVo roadmapVo, RoadmapDetailVo roadmapDetailVo, RoadmapDetailOutVo roadmapDetailOutVo){

    Map<String, Object> map = new HashMap<String, Object>();

    // 사이드바 로드맵 리스트
    List<RoadmapVo> list = roadmapDetailService.roadmapList(roadmapVo);

    // 세부 로드맵 리스트
    List<RoadmapDetailVo> rdList = roadmapDetailService.roadmapDetail(roadmapDetailVo);

    // 과목 리스트
    List<SubjectVo> sList = roadmapDetailService.selectSubject();

    // 교재 리스트
    List<TextbookVo> tList = roadmapDetailService.selectTextbook(roadmapVo);

    // 시작전 카운트
    int sCount = roadmapDetailService.startCount(roadmapDetailVo);

    // 진행중 카운트
    int iCount = roadmapDetailService.ingCount(roadmapDetailVo);

    // 완료 카운트
    int eCount = roadmapDetailService.endCount(roadmapDetailVo);

    // 상태없음 카운트
    int nCount = roadmapDetailService.noneCount(roadmapDetailVo);

    map.put("list", list);
    map.put("rdList", rdList);
    map.put("sList", sList);
    map.put("tList", tList);
    map.put("sCount", sCount);
    map.put("iCount", iCount);
    map.put("eCount", eCount);
    map.put("nCount", nCount);

    return map;
}

// 교재 저장
@ResponseBody
@RequestMapping("/subjectwrite")
public List<TextbookVo> subjectWrite(TextbookVo textbookVo, RoadmapVo roadmapVo) {

    int result = roadmapDetailService.subjectWrite(textbookVo);

    // 교재 리스트
    List<TextbookVo> tList = roadmapDetailService.selectTextbook(roadmapVo);
    return tList;
}

// 세부 로드맵 저장
@ResponseBody
@RequestMapping("/detailwrite")
public List<RoadmapDetailVo> detailRoadmapWrite(RoadmapDetailVo roadmapDetailVo){

    int result = roadmapDetailService.detailRoadmapWrite(roadmapDetailVo);

    // 세부 요약 로드맵 리스트
    List<RoadmapDetailVo> rdList = roadmapDetailService.roadmapDetail(roadmapDetailVo);
    //System.out.println("rdList : " + rdList);
    return rdList;
}
```

[Controller]

```
// 세부 로드맵 삭제
@ResponseBody
@RequestMapping("/detailroadmapDel")
public int detailroadmapDel(RoadmapDetailVo roadmapDetailVo) {

    int result = roadmapDetailService.detailroadmapDel(roadmapDetailVo);
    return result;
}

// 수정 모달창에 값 가져오기
@ResponseBody
@RequestMapping("/inModi")
public RoadmapDetailVo inModi(RoadmapDetailVo roadmapDetailVo) {

    // 세부 모달 리스트
    RoadmapDetailVo inList = roadmapDetailService.inModi(roadmapDetailVo);
    //System.out.println("inList : " + inList);
    return inList;
}

// 세부 로드맵 수정
@ResponseBody
@RequestMapping("/detailModi")
public RoadmapDetailVo detailModi(RoadmapDetailVo roadmapDetailVo) {

    int result = roadmapDetailService.detailModi(roadmapDetailVo);

    // 세부 모달 리스트
    RoadmapDetailVo inList = roadmapDetailService.inModi(roadmapDetailVo);
    return inList;
}

@ResponseBody
@RequestMapping("/count_state")
public Map<String, Object> count_state(RoadmapDetailVo roadmapDetailVo) {

    Map<String, Object> map = new HashMap<String, Object>();

    // 시작전 카운트
    int sCount = roadmapDetailService.startCount(roadmapDetailVo);

    // 진행중 카운트
    int iCount = roadmapDetailService.ingCount(roadmapDetailVo);

    // 완료 카운트
    int eCount = roadmapDetailService.endCount(roadmapDetailVo);

    // 상태없을 카운트
    int nCount = roadmapDetailService.noneCount(roadmapDetailVo);

    map.put("sCount", sCount);
    map.put("iCount", iCount);
    map.put("eCount", eCount);
    map.put("nCount", nCount);

    return map;
}

// 로드맵 삭제
@ResponseBody
@RequestMapping("/roadmapDel")
public RoadmapVo roadmapDel(RoadmapVo roadmapVo, RoadmapDetailVo roadmapDetailVo) {
    // 삭제시 이전 로드맵 가져오기
    RoadmapVo preRoadmap = roadmapDetailService.preRoadmapList(roadmapVo);

    int result = roadmapDetailService.roadmapDel(roadmapDetailVo);
    return preRoadmap;
}
```

[Controller]

```
//로드맵 드래그 앤 드롭
@ResponseBody
@RequestMapping("/detailroadmapDrag")
public int detailroadmapDrag(RoadmapDetailVo roadmapDetailVo) {
    //로드맵 세부 상태 변경
    int result = roadmapDetailService.detailroadmapDrag(roadmapDetailVo);
    return result;
}
```

[Service]

```
public interface RoadmapDetailService {

    // 로드맵 제목 변경
    int modItitle(RoadmapVo roadmapVo);

    // 수정된 로드맵 리스트
    RoadmapVo modIRoadmapList(RoadmapVo roadmapVo);

    // 사이드바 로드맵 리스트
    List<RoadmapVo> roadmapList(RoadmapVo roadmapVo);

    // 세부 로드맵 리스트
    List<RoadmapDetailVo> roadmapDetail(RoadmapDetailVo roadmapDetailVo);

    // 교재 저장
    int subjectWrite(TextbookVo textbookVo);

    // 교재 리스트
    List<TextbookVo> selectTextbook(RoadmapVo roadmapVo);

    // 과목 리스트
    List<SubjectVo> selectSubject();

    // 세부 로드맵 저장
    int detailRoadmapWrite(RoadmapDetailVo roadmapDetailVo);

    // 세부 로드맵 삭제
    int detailroadmapDel(RoadmapDetailVo roadmapDetailVo);

    // 시작전 카운트
    int startCount(RoadmapDetailVo roadmapDetailVo);

    // 진행중 카운트
    int ingCount(RoadmapDetailVo roadmapDetailVo);

    // 완료 카운트
    int endCount(RoadmapDetailVo roadmapDetailVo);

    // 상태없음 카운트
    int noneCount(RoadmapDetailVo roadmapDetailVo);

    // 수정 모달창에 값 가져오기
    RoadmapDetailVo inModi(RoadmapDetailVo roadmapDetailVo);

    // 세부 로드맵 수정
    int detailModi(RoadmapDetailVo roadmapDetailVo);

    // 로드맵 삭제
    int roadmapDel(RoadmapDetailVo roadmapDetailVo);

    // 삭제시 이전 로드맵 가져오기
    RoadmapVo preRoadmapList(RoadmapVo roadmapVo);

    // 로드맵 드래그 앤 드롭
    int detailroadmapDrag(RoadmapDetailVo roadmapDetailVo);
}
```

[ServiceImpl]

```
// 로드맵 제목 변경
@Override
public int modititle(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int result = roadmapDetailMapper.updateTitle(roadmapVo);
    return result;
}

// 수정된 로드맵 리스트
@Override
public RoadmapVo modiRoadmapList(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));
    RoadmapVo modiRoadmap = roadmapDetailMapper.modiRoadmapList(roadmapVo);
    return modiRoadmap;
}

// 사이드바 로드맵 리스트
@Override
public List<RoadmapVo> roadmapList(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));

    List<RoadmapVo> list = roadmapDetailMapper.roadmapList(roadmapVo);
    return list;
}

// 세부 로드맵 리스트
@Override
public List<RoadmapDetailVo> roadmapDetail(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    List<RoadmapDetailVo> rdList = roadmapDetailMapper.selectRoadmapDetail(roadmapDetailVo);
    return rdList;
}

// 교재 저장
@Override
public int subjectWrite(TextbookVo textbookVo) {
    textbookVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int result = roadmapDetailMapper.insertSubject(textbookVo);
    return result;
}

// 교재 리스트
@Override
public List<TextbookVo> selectTextbook(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));

    List<TextbookVo> tList = roadmapDetailMapper.selectTextbook(roadmapVo);
    return tList;
}

// 과목 리스트
@Override
public List<SubjectVo> selectSubject() {
    List<SubjectVo> sList = roadmapDetailMapper.selectSubject();
    return sList;
}

// 세부 로드맵 저장
@Override
public int detailRoadmapWrite(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int result = roadmapDetailMapper.insertDetailRoadmap(roadmapDetailVo);
    return result;
}

// 세부 로드맵 삭제
@Override
public int detailroadmapDel(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int result = roadmapDetailMapper.detailroadmapDelete(roadmapDetailVo);
    return result;
}

// 상태별 카운트
@Override
public int startCount(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int sCount = roadmapDetailMapper.startCount(roadmapDetailVo);
    return sCount;
}
```

[ServiceImpl]

```
// 진행중 카운트
@Override
public int ingCount(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int iCount = roadmapDetailMapper.ingCount(roadmapDetailVo);
    return iCount;
}

// 완료 카운트
@Override
public int endCount(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int eCount = roadmapDetailMapper.endCount(roadmapDetailVo);
    return eCount;
}

// 상태없음 카운트
@Override
public int noneCount(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int nCount = roadmapDetailMapper.noneCount(roadmapDetailVo);
    return nCount;
}

// 수정 모달창에 값 가져오기
@Override
public RoadmapDetailVo inModi(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    // 세부 축약 로드맵 리스트
    RoadmapDetailVo inList = roadmapDetailMapper.inModi(roadmapDetailVo);
    return inList;
}

// 세부 로드맵 수정
@Override
public int detailModi(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int result = roadmapDetailMapper.detailModi(roadmapDetailVo);
    return result;
}

// 로드맵 삭제
@Override
public int roadmapDel(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    int result = roadmapDetailMapper.roadmapDel(roadmapDetailVo);
    return result;
}

// 삭제시 이전 로드맵 가져오기
@Override
public RoadmapVo preRoadmapList(RoadmapVo roadmapVo) {
    roadmapVo.setU_no((Integer)session.getAttribute("session_u_no"));

    RoadmapVo preRoadmap = roadmapDetailMapper.preRoadmapList(roadmapVo);
    return preRoadmap;
}

//로드맵 드래그 앤 드롭
@Override
public int detailroadmapDrag(RoadmapDetailVo roadmapDetailVo) {
    roadmapDetailVo.setU_no((Integer)session.getAttribute("session_u_no"));

    //로드맵 세부 상태 변경
    int result = roadmapDetailMapper.updateRoadmapDetailState(roadmapDetailVo);
    return result;
}
```

[Mapper]

```
@Mapper
public interface RoadmapDetailMapper {

    // 로드맵 제목 변경
    int updateTitle(RoadmapVo roadmapVo);

    // 수정된 로드맵 리스트
    RoadmapVo modiRoadmapList(RoadmapVo roadmapVo);

    // 사이드 바 로드맵 리스트
    List<RoadmapVo> roadmapList(RoadmapVo roadmapVo);

    // 세부 로드맵 리스트
    List<RoadmapDetailVo> selectRoadmapDetail(RoadmapDetailVo roadmapDetailVo);

    // 교재 저장
    int insertSubject(TextbookVo textbookVo);

    // 교재 리스트
    List<TextbookVo> selectTextbook(RoadmapVo roadmapVo);

    // 과목 리스트
    List<SubjectVo> selectSubject();

    // 세부 로드맵 저장
    int insertDetailRoadmap(RoadmapDetailVo roadmapDetailVo);

    // 세부 로드맵 삭제
    int detailRoadmapDelete(RoadmapDetailVo roadmapDetailVo);

    // 시작전 카운트
    int startCount(RoadmapDetailVo roadmapDetailVo);

    // 진행중 카운트
    int ingCount(RoadmapDetailVo roadmapDetailVo);

    // 완료 카운트
    int endCount(RoadmapDetailVo roadmapDetailVo);

    // 상태없음 카운트
    int noneCount(RoadmapDetailVo roadmapDetailVo);

    // 수정 모달창에 값 가져오기
    RoadmapDetailVo inModi(RoadmapDetailVo roadmapDetailVo);

    // 세부 로드맵 수정
    int detailModi(RoadmapDetailVo roadmapDetailVo);

    // 로드맵 삭제
    int roadmapDel(RoadmapDetailVo roadmapDetailVo);

    // 삭제시 이전 로드맵 가져오기
    RoadmapVo preRoadmapList(RoadmapVo roadmapVo);

    // 로드맵 세부 상태 변경
    int updateRoadmapDetailState(RoadmapDetailVo roadmapDetailVo);
}
```

[Mapper.xml]

```
<!-- 로드맵 제목 변경 -->
<update id="updateTitle">
    update roadmap set r_name=#{r_name} where r_no=#{r_no}
</update>

<!-- 수정된 로드맵 리스트 -->
<select id="modiRoadmapList" resultType="com.site.team1.vo.RoadmapVo">
    select * from roadmap where r_no=#{r_no}
</select>

<!-- 사이드바 로드맵 리스트 -->
<select id="roadmapList" resultType="com.site.team1.vo.RoadmapVo">
    select * from roadmap where u_no=#{u_no} and cate_sub_no not in('2', '4', '6', '22') and r_expose not in('N') order by r_no desc
</select>

<!-- 세부 로드맵 리스트 -->
<select id="selectRoadmapDetail" resultType="com.site.team1.vo.RoadmapDetailVo">
    select d.r_detail_no, d.r_detail_title, d.r_detail_date, d.r_detail_content, d.r_detail_priority, d.r_detail_startdate, d.r_detail_enddate,
    d.r_detail_state, d.r_detail_grade, d.s_no, s.s_name, t.textbook_no, t.textbook_content
    from roadmap_detail d, subject s, textbook t
    where (d.s_no(+) = s.s_no and d.textbook_no = t.textbook_no(+)) and d.r_no=#{r_no}
    order by r_detail_no desc
</select>

<!-- 수정 modal 리스트 -->
<select id="inModi" resultType="com.site.team1.vo.RoadmapDetailVo">
    select d.r_detail_no, d.r_detail_title, d.r_detail_date, d.r_detail_content, d.r_detail_priority, d.r_detail_startdate, d.r_detail_enddate,
    d.r_detail_state, d.r_detail_grade, d.s_no, s.s_name, t.textbook_no, t.textbook_content
    from roadmap_detail d, subject s, textbook t
    where (d.s_no(+) = s.s_no and d.textbook_no = t.textbook_no(+)) and r_detail_no=#{r_detail_no}
</select>

<!-- 교재 저장 -->
<insert id="insertSubject">
    insert into textbook values (textbook_seq.nextval, #{textbook_content}, #{u_no})
</insert>

<!-- 교재 리스트 -->
<select id="selectTextbook" resultType="com.site.team1.vo.TextbookVo">
    select * from textbook where u_no=#{u_no} and textbook_no not in('0') order by textbook_no desc
</select>

<!-- 과목 리스트 -->
<select id="selectSubject" resultType="com.site.team1.vo.SubjectVo">
    select * from subject order by s_no asc
</select>

<!-- 세부 로드맵 저장 -->
<insert id="insertDetailRoadmap">
    insert into roadmap_detail values (roadmap_detail_seq.nextval, #{r_detail_title}, sysdate, #{r_detail_content}, #{r_detail_priority}, #{r_detail_startdate}, #{r_detail_enddate},
    #{r_no}, #{s_no}, #{r_detail_state}, #{r_detail_grade}, #{textbook_no})
</insert>

<!-- 세부 로드맵 수정 -->
<!-- , r_detail_date=#{r_detail_date}, -->
<update id="detailModi">
    update roadmap_detail set r_detail_title=#{r_detail_title}, r_detail_content=#{r_detail_content}, r_detail_priority=#{r_detail_priority},
    r_detail_startdate=#{r_detail_startdate}, r_detail_enddate=#{r_detail_enddate}, s_no=#{s_no}, r_detail_state=#{r_detail_state}, r_detail_grade=#{r_detail_grade}, textbook_no=#{textbook_no}
    where r_detail_no=#{r_detail_no}
</update>

<!-- 세부 로드맵 삭제 -->
<delete id="detailRoadmapDelete">
    delete from roadmap_detail where r_detail_no=#{r_detail_no}
</delete>

<!-- 시작전 카운트 -->
<select id="startCount" resultType="int">
    select count(*) from roadmap_detail where r_no=#{r_no} and r_detail_state='시작 전'
</select>

<!-- 진행중 카운트 -->
<select id="ingCount" resultType="int">
    select count(*) from roadmap_detail where r_no=#{r_no} and r_detail_state='진행 중'
</select>

<!-- 완료 카운트 -->
<select id="endCount" resultType="int">
    select count(*) from roadmap_detail where r_no=#{r_no} and r_detail_state='완료'
</select>

<!-- noneCount 카운트 -->
<select id="noneCount" resultType="int">
    select count(*) from roadmap_detail where r_no=#{r_no} and r_detail_state='상태 없음'
</select>

<!-- 로드맵 삭제 -->
<update id="roadmapDel">
    update roadmap set r_expose='N' where r_no=#{r_no}
</update>
```

[Mapper.xml]

```
<!-- 삭제시 이전 로드맵 가져오기 -->
<select id="preRoadmapList" resultType="com.site.team1.vo.RoadmapVo">
    select * from (select rownum as rnum, r.* from (select * from roadmap where cate_sub_no not in('2', '4', '6', '22') and r_expose not in('N') and u_no=#{u_no} order by r_no desc) r)
    where rnum=(select rnum from (select rownum as rnum, r.* from (select * from roadmap
    where cate_sub_no not in('2', '4', '6', '22') and r_expose not in('N') and u_no=#{u_no} order by r_no desc) r)
    where r_no=#{r_no})+1
</select>

<!-- 로드맵 세부 상태 변경 -->
<update id="updateRoadmapDetailState">
    update roadmap_detail
    set    r_detail_state = #{r_detail_state}
    where r_detail_no = #{r_detail_no}
</update>
```

[DAO]

```
@NoArgsConstructor
@AllArgsConstructor
@Data
public class RoadmapVo {

    private int r_no;
    private String r_name;
    private String r_expose;

    private int u_no;
    private int cate_sub_no;
}

@NoArgsConstructor
@AllArgsConstructor
@Data
public class TextbookVo {

    private int textbook_no;
    private String textbook_content;

    private int u_no;
}

@NoArgsConstructor
@AllArgsConstructor
@Data
public class SubjectVo {

    private int s_no;
    private String s_name;
}

@NoArgsConstructor
@AllArgsConstructor
@Data
public class RoadmapDetailVo {

    private int u_no;

    private int r_detail_no;
    private String r_detail_title;
    private Timestamp r_detail_date;
    private String r_detail_content;
    private String r_detail_priority;
    private Timestamp r_detail_startdate;
    private String r_detail_state;
    private String r_detail_grade;
    private Timestamp r_detail_enddate;

    private int r_no;

    private int s_no;
    private String s_name;

    private int textbook_no;
    private String textbook_content;
}
```