

PORTFOLIO



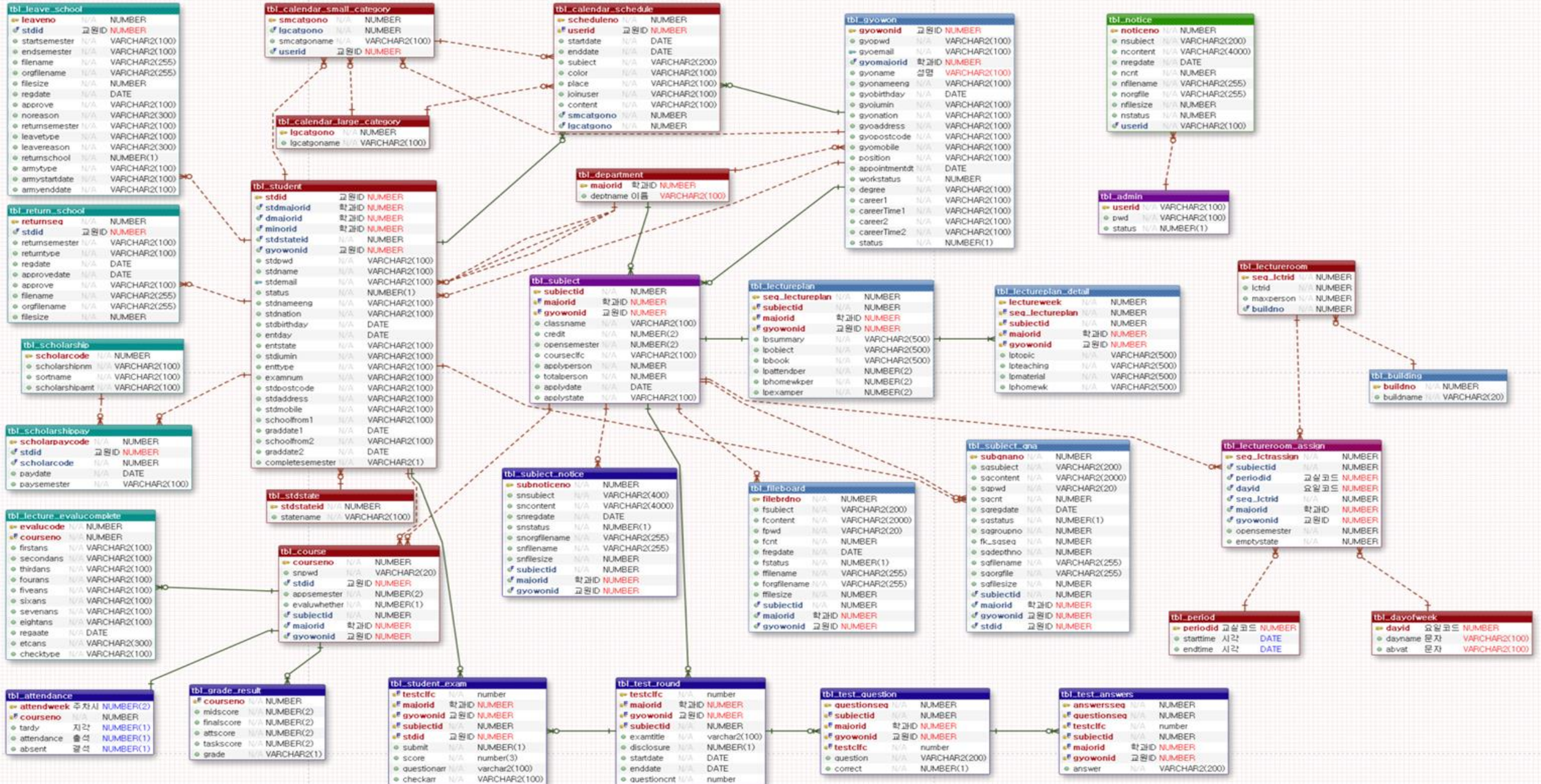
PROJECT

Theme : 학사 정보시스템 (LMS)

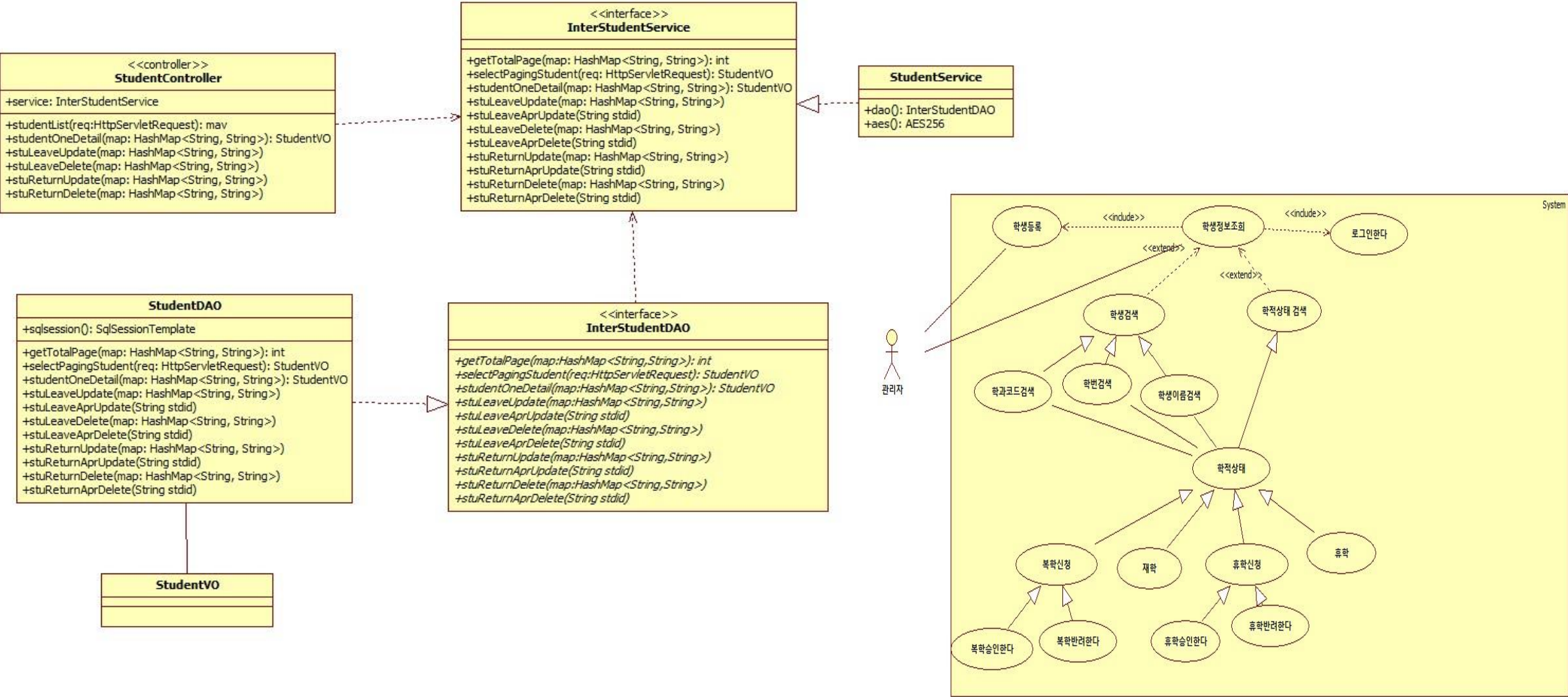
1. 개요

PERIOD	2022.04.23 ~ 2022.05.23
OBJECT	학생: 개인정보를 조회/수정, 성적확인이나 수강신청/ 조회 , 휴/복학 신청 교수: 자신의 개인정보, 강의에 대한 학생정보, 성적관련 정보 수정, 강의 개설 신청, 시험 출제 관리자: 학생/교원 등록, 학생/교원 조회, 학생 휴/복학 관리, 교수 강의 개설 관리, 게시판 관리
ROLE	- 학생/교원 등록 - 학생/교원 조회 - 휴/복학 신청 관리 - 교수강의 개설 조회 및 관리
CONTRIBUTION	25% 팀(6명)
GitHub	https://github.com/leejengjun/lmsfinal.git
Impression	Spring을 통하여 학사관리 시스템을 구현함으로써 관리자의 승인 처리, 회원 등록, 회원 조회 시 페이징 처리 등을 구현할 수 있었고 저번 프로젝트 때보다 익숙해지고 능력이 발전하여 이번 프로젝트에는 수월하게 진행이 될 수 있었고 에러의 발생 횟수도 많이 줄어나갔습니다. 또한 팀원들과 서로 도우며 맡은 파트의 연결 부분 등을 토론하면서 협업 의식을 더욱 느끼게 되었고 한 번 더 팀워크의 중요성을 깨달았습니다. 함께 열심히 해준 팀원분들 모두 감사합니다.

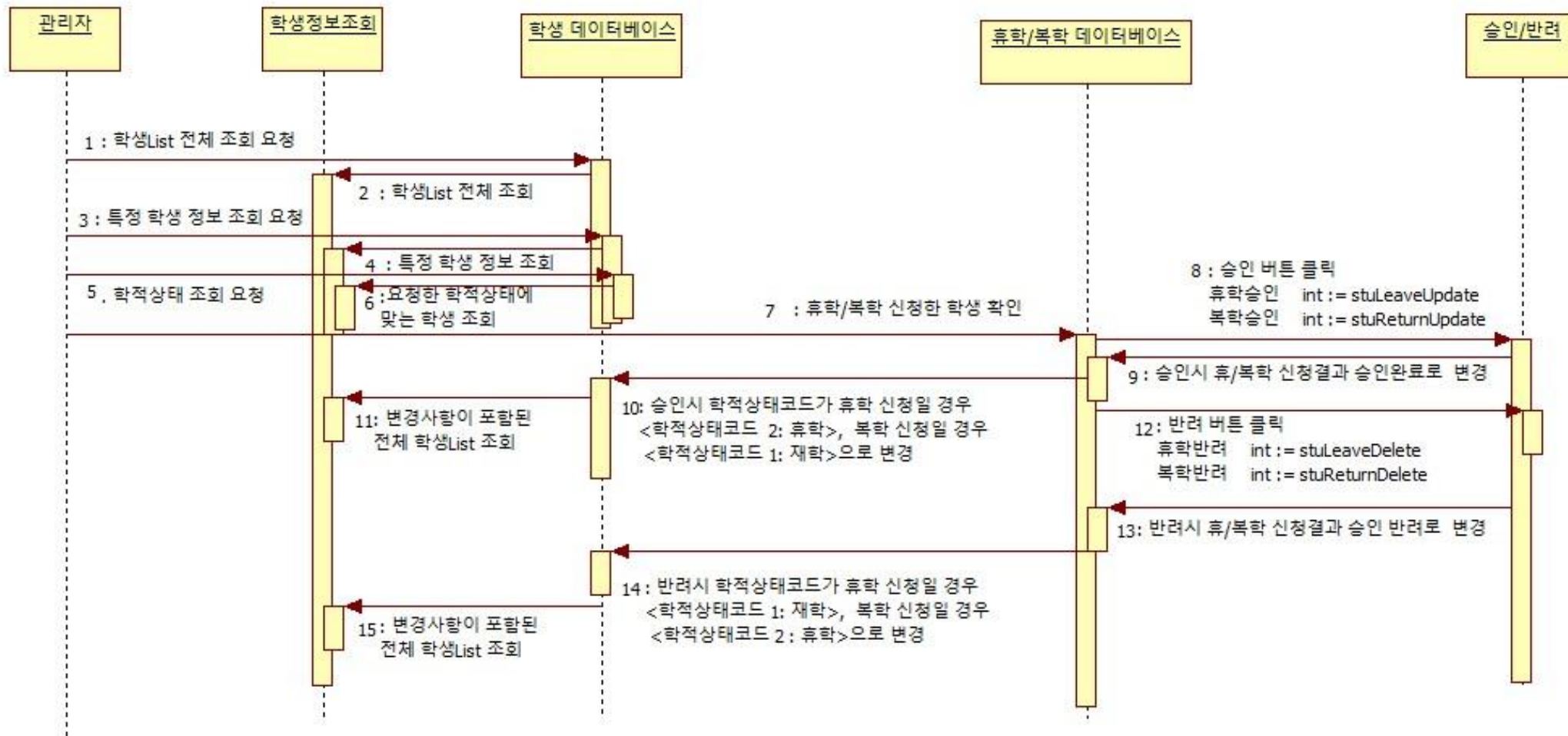
2. 담당 기능-eXERD



2. 담당 기능- Class Diagram, Usecase Diagram



2. 담당 기능- Sequence Diagram



2. 담당 기능- 회원관리_학생등록



학생 등록

학번 **학번 조회**

비밀번호

이름

주민등록번호 '-'를 제외하고 기입하시오.

생년월일 **이메일 중복 확인**

이메일

연락처 예) 01012345678

학과코드 **학과 찾기**

교원번호 **교원 찾기**

입학전형 입학전형 선택

입학구분 입학구분 선택

입학일자

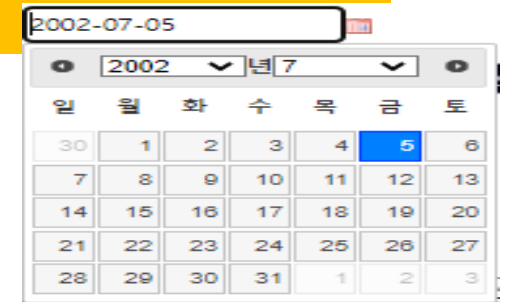
학적상태 ☐ 재학

가입하기

학번 조회 버튼 클릭 시 학생 테이블에 중복된 학번이 존재하는지 확인.

이메일 중복 확인 버튼 클릭 시 학생 테이블에 중복된 이메일이 존재하는지 확인.
이메일은 암호화.

datepicker를 사용하여 생년월일 또는 입학일자 선택.



```

$("input#datepicker").datepicker({
    dateFormat: 'yy-mm-dd'
    ,showOtherMonths: true
    ,showMonthAfterYear:true
    ,changeYear: true
    ,changeMonth: true
    ,showOn: "both"
    ,buttonImage: "http://jqueryui.com/resources/demos/datepicker/images/calendar.gif"
    ,buttonImageOnly: true
    ,buttonText: "선택"
    ,yearSuffix: "년"
    ,monthNamesShort: ['1','2','3','4','5','6','7','8','9','10','11','12']
    ,monthNames: ['1월','2월','3월','4월','5월','6월','7월','8월','9월','10월','11월','12월']
    ,dayNamesMin: ['일','월','화','수','목','금','토']
    ,dayNames: ['일요일','월요일','화요일','수요일','목요일','금요일','토요일']
    ,minDate: '-100y'
    ,yearRange: 'c-100:c+10'
});
$("input#datepicker").datepicker('setDate', 'today');
    
```

정규표현식 사용
비밀번호는 암호화

입학전형: 수시 또는 정시 또는 재외국민 선택.
입학구분: 신입학 또는 편입학 선택.

교원 찾기 버튼 클릭 시 Modal을 사용하여 Ajax를 통해 입력한 교원번호 또는 학과코드 또는 학과명에 해당하는 교원번호 조회

학과 찾기 버튼 클릭 시 Modal을 사용하여 Ajax를 통해 입력한 학과코드 또는 학과명에 해당하는 학과코드 조회

2. 담당 기능- 회원관리_학생등록

학과 찾기 버튼 클릭시 모달창이 뜬다.

13 정보통신공학과

```
Map<String, String> paraMap = new HashMap<>();
paraMap.put("searchType", searchType);
paraMap.put("searchWord", searchWord);

stdDeptlist = service.stdDeptlistSearch(paraMap);

JSONArray jsonArray = new JSONArray();

if( stdDeptlist != null ) {
    for(StudentVO studentvo : stdDeptlist) {
        JSONObject jsonObj = new JSONObject();
        jsonObj.addProperty("majorid", studentvo.getMajorid());
        jsonObj.addProperty("deptname", studentvo.getDeptname());
        jsonArray.add(jsonObj);
    } // end of for-----
}

return new Gson().toJson(jsonArray);
```

찾고자 하는 것을 입력 후 Enter 또는 조회 버튼
클릭시 이에 맞는 결과를 보여준다.

학과 찾기 버튼 클릭 시 Modal을 사용하여
Ajax를 통해 입력한 학과코드 또는 학과명에
해당하는 학과코드 조회

```
function goDpSearch(){
    $.ajax({
        url:"<%= ctxPath%>/stdDeptlist.lmsfinal",
        data:{"searchType":$("select#searchType").val(),
            "searchWord":$("input#searchWord").val()},
        dataType:"JSON",
        success:function(json){
            let html = "";
            if(json.length > 0) {
                $.each(json, function(index, item){
                    html += "<div class='result' id='index' style='cursor:pointer; display:block;";
                    html += "<span id='stdmajorid'>"+item.majorid+"</span>&nbsp;&nbsp;&nbsp;";
                    html += "<span id='deptname'>"+item.deptname+"</span>";
                    html += "</div>";
                });
            }
            else {
                alert("해당하는 학과가 없습니다. 다시 입력하세요.");
            }
            $("div#div_searchResult").html(html);
        },
        error: function(request, status, error){
            alert("code: "+request.status+"\n"+"message: "+request.responseText+"\n"+"error: "+error);
        }
    });
}

// end of function goDpSearch(){}-----
```

학과코드 칸 클릭시 모달창이 뜬다.

```
// === 학과코드에 포커스가 가는 경우 학과찾기 모달창 활성화
$("input#majorid").click(function(){
    $('#deptSearch').modal('show');
});

$("input#deptname").click(function(){
    $('#deptSearch').modal('show');
});

// === 학과 찾아 값 넣어주기
$("input#searchWord").keyup(function(event){
    if(event.keyCode == 13){
        // 엔터를 했을 경우
        goDpSearch();
    }
});

=== 검색 결과에 따라 학과 코드와 학과명 넣어주기
```

```
$(document).on("click", "div.result", function(){
    const stdmajorid = $(this).find("span#stdmajorid").text();
    $("input#stdmajorid").val(stdmajorid);
    //텍스트박스에 검색된 결과의 문자열을 입력해준다.
    $('#deptSearch').modal('hide');
});
```

2. 담당 기능- 회원관리_학생등록

교원 찾기

학과명 ▼ 컴퓨터 조회

- 11컴퓨터공학과의10022 사교스
- 11컴퓨터공학과의10001 김영희
- 11컴퓨터공학과의10004 윤교수
- 11컴퓨터공학과의10076 장미교
- 11컴퓨터공학과의10078 고을빈

Close

교원 찾기 버튼 클릭시 모달창이 뜬다.

교원 찾기

학과코드 ▼ 11 조회

- 11컴퓨터공학과의10022 사교스
- 11컴퓨터공학과의10001 김영희
- 11컴퓨터공학과의10004 윤교수
- 11컴퓨터공학과의10076 장미교
- 11컴퓨터공학과의10078 고을빈

Close

교원 찾기 버튼 클릭시 모달창이 뜬다.

```
if( stdGyowonlist != null ) {
    for(StudentVO studentvo : stdGyowonlist) {
        JSONObject jsonObj = new JSONObject();
        jsonObj.addProperty("gyowonid", studentvo.getGyowonid());
        jsonObj.addProperty("majorid", studentvo.getMajorid());
        jsonObj.addProperty("gyoname", studentvo.getGyoname());
        jsonObj.addProperty("deptname", studentvo.getDeptname());
        jsonArr.add(jsonObj);
    } // end of for
}
return new Gson().toJson(jsonArr);
```

찾고자 하는 것을 입력 후 Enter 또는 조회 버튼
클릭시 이에 맞는 결과를 보여준다.

교원번호 칸 클릭시 모달창이 뜬다.

```
// === 교원번호에 포커스가 가는 경우 교원찾기 모달창 활성화
$("input#majorid").click(function(){
    $('#gyowonSearch').modal('show');
});

$("input#gyowonid").click(function(){
    $('#gyowonSearch').modal('show');
});
```

교원 찾기 버튼 클릭 시 Modal을 사용하여 Ajax를 통한 입력한 교원번호 또는 학과코드 또는 학과명에 해당하는 교원번호 조회

```
$.ajax({
    url: "<%= ctxPath%>/stdGyowonlist.lmsfinal",
    data: { "searchType_1": $("<select#searchType_1>").val(),
            "searchWord_1": $("<input#searchWord_1>").val() },
    dataType: "JSON",
    success: function(json){
        let html = "";
        if(json.length > 0) {
            $.each(json, function(index, item){
                html += "<div class='gResult' id='index' s";
                html += "<span id='majorid'>" + item.maj
                html += "<span id='deptname'>" + item.de
                html += "<span id='gyowonid'>" + item.gy
                html += "<span id='gyoname'>" + item.gyo
                html += "</div>";
            });
        }
        else {
            alert("<해당하는 교원이 없습니다. 다시 입력하세요.>");
        }
    } // end of else -----
    $("<div#div_GySearchResult>").html(html);
});
```

// === 검색 결과에 따라 넣어주기

```
$(document).on("click", "div.gResult", function(){
    const gyowonid = $(this).find("span#gyowonid").text();
    $("#input#gyowonid").val(gyowonid);
    //텍스트박스에 검색된 결과의 문자열을 입력해준다.
    $('#gyowonSearch').modal('hide');
});
```

2. 담당 기능- 회원관리_학생등록

학번	<input type="text"/>	학번 조회
비밀번호	<input type="password"/>	
이름	<input type="text"/>	
주민등록번호	'-'을 제외하고 기입하시오.	
이메일	<input type="text" value="abc@def.com"/>	이메일 중복확인

가입하기

항목이 전부 입력이 되어야 가입이 완료된다.

```
//## 이메일 중복여부 검사하기 ##//
```

```
function isExistStdEmailCheck() {
```

```
b_flagStdEmailDuplicatClick = true;
```

```
$.ajax({
  url:"<%= ctxPath%>/stdemailDuplicateCheck.lmsfinal",
  data:{"stdemail":$("input#stdemail").val()},
  type:"post",
  dataType:"json",
  success : function (data) {
```

```
    if(data == 1) {
      alert("중복된 이메일입니다.");
    } else if (data == 0) {
      alert("사용 가능한 이메일입니다.")
    }
  }
});
```

학번 조회 또는 이메일 중복확인 버튼 클릭 시 학생 테이블에 중복된 학번 또는 중복된 이메일이 존재하는지 확인.

```
// ## 학번중복검사 하기 ## ///
```

```
$("#button#stdidcheck").click(function(){
```

```
b_flagStdIdDuplicatClick = true;
```

```
$.ajax({
  url:"<%= ctxPath%>/stdidDuplicateCheck.lmsfinal",
  data:{"stdid":$("input#stdid").val()},
  type:"post",
  success:function(data){
    if(data == 1) {
      alert("중복된 학번입니다.");
    } else if (data == 0) {
      alert("사용 가능한 학번입니다.")
    }
  }
});
```

Ajax를 통해 이메일 중복확인,
학번이 중복 되었는지 확인

```
// 이메일 정규표현식
```

```
const regExp = new RegExp(/^([0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*\@[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*\.[a-zA-Z]{2,3})$/i);
```

```
//주민번호정규표현식
```

```
const regExp = new RegExp(/[0-9]{2}([0-9]|1[0-2])([0-9]|1[1,2][0-9]|3[0,1])/);
```

이메일과 주민번호는 정규표현식을 사용한다.

2. 담당 기능- 회원관리_회원조회_학생조회

학생정보, 교원정보 각 클릭 시
학생 전체 목록, 교원 전체 목록 확인 가능.

회원전체 목록

검색 대상에 학생 이름, 학번, 학과코드,
학과명으로 조회 가능.

학적 상태에 재학, 휴학, 휴학신청,
복학신청 조회 가능.

김진아 님의 회원 상세정보

학번	202256565
학과코드	12
학과이름	정보보안학과
복수전공코드	
학적상태코드	재학
이름	김진아
이메일	jina@naver.com
생년월일	1996-07-15 00:00:00.0
입학일자	2022-03-02 00:00:00.0
입학전형	정시
입학구분	신입학
우편번호	
주소	
연락처	01056565656

이름 클릭 시 학생
상세 정보 확인 가능.

Paging 처리

1 2 [다음] [마지막]

학적 상태가 휴학 신청 또는 복학 신청인
경우 Ajax 를 통해 승인 또는 반려 가능.

학생정보

교원정보

학생이름

이정준

복학신청

조회

이름	학번	학과코드	학과명	학적상태
이정준	202277777	11	컴퓨터공학과	<div>복학신청</div> <div>승인</div> <div>반려</div>

휴학신청

승인

반려

2. 담당 기능- 회원관리_회원조회_학생조회

// == 학생리스트 확인 하기 == //

```
@RequestMapping(value="/studentList.lmsfinal", method={RequestMethod.GET})
public ModelAndView studentList(ModelAndView mav, HttpServletRequest request) {

    String searchType_1 = request.getParameter("searchType_1");
    String searchWord_1 = request.getParameter("searchWord_1");
    String str_currentShowPageNo = request.getParameter("currentShowPageNo");
    String deptname = request.getParameter("deptname");
    String stdstateid = request.getParameter("stdstateid");

    if(searchType_1 == null || (!"stdname".equals(searchType_1) && !"stdid".equals(searchType_1)
        && !"stdmajorid".equals(searchType_1) && !"deptname".equals(searchType_1)) ) {
        searchType_1 = "";
    }

    if(searchWord_1 == null || "".equals(searchWord_1) || searchWord_1.trim().isEmpty() ) {
        searchWord_1 = "";
    }

    if(stdstateid == null || "".equals(stdstateid) || stdstateid.trim().isEmpty() ) {
        stdstateid = "";
    }

    Map<String, String> paramMap = new HashMap<>();
    paramMap.put("searchType_1", searchType_1);
    paramMap.put("searchWord_1", searchWord_1);
    paramMap.put("deptname", deptname);
    paramMap.put("stdstateid", stdstateid);
```

페이징 처리 후 DB에 보낸 학생 목록을 가져온다.

페이징 처리를 통해 전체 학생목록을 보여준다.
Map에서 가져온 값이 ""이면 검색이 없는 것이다.

학생정보 | 교원정보
검색대상 | 재학 | 조회

이름	학번	학과코드	학과명	학적상태
권현빈	202200005	13	정보통신공학과	재학
권현우	202200006	11	컴퓨터공학과	재학
윤빈선	202200008	13	정보통신공학과	재학
김지우	202200010	12	정보보안학과	재학
이지은	202200016	11	컴퓨터공학과	재학
김주원	202298989	13	정보통신공학과	재학
원장미	202216166	11	컴퓨터공학과	재학

[맨처음] [이전] 3

```
int totalCount = 0; // 총게시물 건수
int sizePerPage = 7; //한 페이지당 보여줄 게시물 건수
int currentShowPageNo = 1;
int totalPages = 0; // 총 페이지 수

int startRno = 0; // 시작 행번호
int endRno = 0; // 끝 행번호

totalCount = service.getTotalPage(paramMap);
totalPages = (int) Math.ceil((double)totalCount/sizePerPage);

if(str_currentShowPageNo == null) {
    currentShowPageNo = 1;
}
else {
    try {
        currentShowPageNo = Integer.parseInt(str_currentShowPageNo);
        if( currentShowPageNo < 1 || currentShowPageNo > totalPages ) {
            currentShowPageNo = 1;
        }
    } catch(NumberFormatException e) {
        currentShowPageNo = 1;
    }
}
```

```
startRno = ((currentShowPageNo - 1) * sizePerPage) + 1;
endRno = startRno + sizePerPage - 1;
```

```
paramMap.put("startRno", String.valueOf(startRno));
paramMap.put("endRno", String.valueOf(endRno));
```

```
List<StudentVO> studentList = service.selectPagingStudent(paramMap);
```

```
if( !"".equals(searchType_1) && !"".equals(searchWord_1)) {
    mav.addObject("paramMap", paramMap);
}
```

```
if( !"".equals(stdstateid)) {
    mav.addObject("paramMap", paramMap);
}
```

Paging 처리

2. 담당 기능- 회원관리_회원조회_학생조회

Paging 처리

```
// [맨처음][이전] //
```

```
if(pageNo != 1) {
```

```
    if(searchType_1 != "") {
```

```
        pageBar += "<li style='display:inline-block; width:70px; font-size:12pt;'><a href='"+url+"?searchType_1="+searchType_1+"&searchWord_1="+searchWord_1+"&stdstateid="+stdstateid+"&currentShowPageNo=1'>[맨처음]</a></li>";
```

```
        pageBar += "<li style='display:inline-block; width:50px; font-size:12pt;'><a href='"+url+"?searchType_1="+searchType_1+"&searchWord_1="+searchWord_1+"&stdstateid="+stdstateid+"&currentShowPageNo="+pageNo-1+"'>[이전]</a></li>";
```

```
    }
```

```
    else if(stdstateid != null) {
```

```
        pageBar += "<li style='display:inline-block; width:70px; font-size:12pt;'><a href='"+url+"?stdstateid="+stdstateid+"&currentShowPageNo=1'>[맨처음]</a></li>";
```

```
        pageBar += "<li style='display:inline-block; width:50px; font-size:12pt;'><a href='"+url+"?stdstateid="+stdstateid+"&currentShowPageNo="+pageNo-1+"'>[이전]</a></li>";
```

```
    }
```

```
}
```

```
while( !(loop > blockSize || pageNo > totalPage) ) {
```

```
    if(pageNo == currentShowPageNo) {
```

```
        if(searchType_1 != "") {
```

```
            pageBar += "<li style='display:inline-block; width:30px; font-size:12pt; border:solid 0px gray; color:red; padding:2px 4px;'>"+pageNo+"</li>";
```

```
        }
```

```
        else if(stdstateid != null) {
```

```
            pageBar += "<li style='display:inline-block; width:30px; font-size:12pt; border:solid 0px gray; color:red; padding:2px 4px;'>"+pageNo+"</li>";
```

```
        }
```

```
    }
```

```
    else {
```

```
        if(searchType_1 != "") {
```

```
            pageBar += "<li style='display:inline-block; width:30px; font-size:12pt;'><a href='"+url+"?searchType_1="+searchType_1+"&searchWord_1="+searchWord_1+"&stdstateid="+stdstateid+"&currentShowPageNo="+pageNo+"'>"+pageNo+"</a></li>";
```

```
        }
```

```
        else if(stdstateid != null) {
```

```
            pageBar += "<li style='display:inline-block; width:30px; font-size:12pt;'><a href='"+url+"?stdstateid="+stdstateid+"&currentShowPageNo="+pageNo+"'>"+pageNo+"</a></li>";
```

```
        }
```

```
    }
```

```
    // [다음][마지막] //
```

```
    if( pageNo <= totalPage ) {
```

```
        if(searchType_1 != "") {
```

```
            pageBar += "<li style='display:inline-block; width:50px; font-size:12pt;'><a href='"+url+"?searchType_1="+searchType_1+"&searchWord_1="+searchWord_1+"&stdstateid="+stdstateid+"&currentShowPageNo="+pageNo+"'>[다음]</a></li>";
```

```
            pageBar += "<li style='display:inline-block; width:70px; font-size:12pt;'><a href='"+url+"?searchType_1="+searchType_1+"&searchWord_1="+searchWord_1+"&stdstateid="+stdstateid+"&currentShowPageNo="+totalPage+"'>[마지막]</a></li>";
```

```
        }
```

```
        else if(stdstateid != null) {
```

```
            pageBar += "<li style='display:inline-block; width:50px; font-size:12pt;'><a href='"+url+"?stdstateid="+stdstateid+"&currentShowPageNo="+pageNo+"'>[다음]</a></li>";
```

```
            pageBar += "<li style='display:inline-block; width:70px; font-size:12pt;'><a href='"+url+"?stdstateid="+stdstateid+"&currentShowPageNo="+totalPage+"'>[마지막]</a></li>";
```

```
        }
```

```
    }
```

```
}
```

pageNo가 매번 증가하는데 totalPage보다 커버리면 빠져 나온다.

2. 담당 기능- 회원관리_회원조회_학생조회

```
<!-- 학생회원전체조회 총개수를 건수(totalCount)-->
<select id = "getTotalPage" parameterType="HashMap" resultType="int">
    select count(*)
    from
    ( select stdname, stdid, stdmajorid, stdstateid, deptname
    from
    ( select A.stdname, A.stdid, A.stdmajorid, A.stdstateid, B.deptname
    from
    ( select stdid, stdmajorid, dmajorid, stdstateid ,scholarcode,
        gyowonid, stdpwd, stdname, stdemail, status, stdnameeng, stdnation, stdbirthday, entday
        ,entstate, stdjumin, enttype, examnum, stdpostcode
        ,stdaddress, stdmobile, schoolfrom1, graddate1, schoolfrom2, graddate2
        from tbl_student) A
    JOIN
    (select majorid, deptname
    from tbl_department) B
    ON A.stdmajorid = B.majorid
    <if test='searchType_1 != ""'>
    and lower(${searchType_1}) like '%' || lower("#{searchWord_1}") || '%' and lower(stdstateid) like '%' || lower("#{stdstateid}") || '%'
    </if>
    <if test='stdstateid != null'>
    and lower(stdstateid) like '%' || lower("#{stdstateid}") || '%'
    </if>
    ) V
    )
</select>

<!-- 학생회원전체조회 -->
<select id = "selectPagingStudent" parameterType="String" resultType="com.spring.lmsfinal.wonhyejin.model.StudentVO">

    select rno, stdname, stdid, stdmajorid, stdstateid, deptname
    from
    (
        select rownum AS rno, A.stdname, A.stdid, A.stdmajorid, A.stdstateid, B.deptname
        from
        (
            select stdid, stdmajorid, dmajorid, stdstateid ,scholarcode,
                gyowonid, stdpwd, stdname, stdemail, status, stdnameeng, stdnation, stdbirthday, entday
                ,entstate, stdjumin, enttype, examnum, stdpostcode
                ,stdaddress, stdmobile, schoolfrom1, graddate1, schoolfrom2, graddate2
            from tbl_student) A
        JOIN
        (select majorid, deptname
        from tbl_department) B
        ON A.stdmajorid = B.majorid
        <if test='searchType_1 != "" and searchWord_1 != ""'>
        where lower(${searchType_1}) like '%' || lower("#{searchWord_1}") || '%' and lower(stdstateid) like '%' || lower("#{stdstateid}") || '%'
        </if>
        <if test='stdstateid != null'>
        and lower(stdstateid) like '%' || lower("#{stdstateid}") || '%'
        </if>
        )Z
        where rno between #{startRno} and #{endRno}
    </select>
```

학생 테이블과 학과테이블을 Join 하여 학생 리스트를 보여준다.
If문을 사용하여 입력 또는 선택한 값을 보여준다.

2. 담당 기능- 회원관리_회원조회_학생 1명 조회

김진아 님의 회원 상세정보

학번	202256565
학과코드	12
학과이름	정보보안학과
복수전공코드	
학적상태코드	재학
이름	김진아
이메일	jina@naver.com
생년월일	1996-07-15 00:00:00.0
입학일자	2022-03-02 00:00:00.0
입학전형	정시
입학구분	신입학
우편번호	
주소	
연락처	01056565656

studentvo에 담아온 것을 불러온다.

회원목록

```
<c:if test="${not empty requestScope.studentvo}">
  <table style="width: 800px; margin: auto;" class="table table-bordered table-white" id="mvoInfo">
    <tr>
      <th style="width: 20%">학번</th>
      <td>${requestScope.studentvo.stdid}</td>
    </tr>
    <tr>
      <th>학과코드</th>
      <td>${requestScope.studentvo.stdmajorid}</td>
    </tr>
    <tr>
      <th>학과이름</th>
      <td>${requestScope.studentvo.deptname}</td>
    </tr>
    <tr>
      <th>복수전공코드</th>
      <td>${requestScope.studentvo.dmajorid}</td>
    </tr>
    <tr>
      <th>학적상태코드</th>
      <td>
        <c:if test="${requestScope.studentvo.stdstateid eq '1'}">재학</c:if>
        <c:if test="${requestScope.studentvo.stdstateid eq '2'}">휴학</c:if>
        <c:if test="${requestScope.studentvo.stdstateid eq '6'}">휴학신청</c:if>
        <c:if test="${requestScope.studentvo.stdstateid eq '7'}">복학신청</c:if>
      </td>
    </tr>
  </table>
</c:if>
```

```
// 학생 1명을 보여주는 페이지 요청 ==//
```

```
@RequestMapping(value="/studentOneDetail.lmsfinal")
public ModelAndView studentOneDetail(ModelAndView mav, HttpServletRequest request) {

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

    StudentVO studentvo = service.studentOneDetail(stdname);

    request.setAttribute("studentvo", studentvo);

    String stdemail = "";
    try {
        stdemail = aes.decrypt(studentvo.getStdemail());
    } catch (NoSuchAlgorithmException e) {

        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {

        e.printStackTrace();
    } catch (GeneralSecurityException e) {

        e.printStackTrace();
    }

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

    request.setAttribute("goBackURL", goBackURL);

    mav.addObject("stdemail", stdemail);
    mav.setViewName("wonhyejin/studentOneDetail.tiles2");
    return mav;
}
```

이메일을 복호화 해준다.

2. 담당 기능- 회원관리_회원조회_학생 휴학 처리

localhost:9090 내용:
휴학승인되었습니다

휴학신청

승인

반려

확인

승인을 누르면 휴학승인이 되어 학적상태가 '2' 인 휴학으로 바뀌고 tbl_leave_school 휴학테이블 approve(신청결과)가 승인완료로 바뀐다.

반려버튼을 누르면 '1' 재학으로 바뀌고 '승인반려' 로 바뀐다.

```
<!-- 학생상태변경_휴학승인-->
<update id="stuLeaveUpdate" parameterType="String">
    update tbl_student set stdstateid = '2'
    where stdid = #{stdid}
</update>

<!-- 학생상태변경_휴학승인_apr-->
<update id="stuLeaveAprUpdate" parameterType="String">
    update tbl_leave_school set approve = '승인완료'
    where stdid = #{stdid} and approve='승인전'
</update>

<!-- 학생상태변경_휴학거절-->
<update id="stuLeaveDelete" parameterType="String">
    update tbl_student set stdstateid = '1'
    where stdid = #{stdid}
</update>

<!-- 학생상태변경_휴학거절_apr-->
<update id="stuLeaveAprDelete" parameterType="String">
    update tbl_leave_school set approve = '승인반려'
    where stdid = #{stdid} and approve='승인전'
</update>
```

Index를 이용하여 특정 값의 위치를 찾는다

```
// 학적상태 변경_휴학신청 반려//

// 학적상태 변경_휴학신청 승인//
$("input.btn_s_approve").click(function(){
    // function updateApprove() {
    const idx = $("input.btn_s_approve").index($(this));
    alert(idx);
    const stdid = $("td#stdid").eq(idx).text();
    alert("확인용 stdid=>" + stdid);
    const stdstateid = $("p#stdstateid").eq(idx).text();
    alert("확인용 stdstateid=>" + stdstateid);
    $.ajax({
        url:"<%= ctxPath%>/stuLeaveUpdate.lmsfinal",
        data:{ "stdid":stdid
            , "stdstateid":$("select#stdstateid").val()},
        type:"post",
        dataType:"JSON",
        success:function(json){
            if(json.n > 0) {
                alert("휴학승인되었습니다");
            }
        }
    });
});
```

```
$("input.btn_s_reject").click(function(){
    const idx = $("input.btn_s_reject").index($(this));
    alert(idx);
    const stdid = $("td#stdid").eq(idx).text();
    alert("확인용 stdid=>" + stdid);
    const stdstateid = $("p#stdstateid").eq(idx).text();
    alert("확인용 stdstateid=>" + stdstateid);
    $.ajax({
        url:"<%= ctxPath%>/stuLeaveDelete.lmsfinal",
        data:{ "stdid":stdid
            , "stdstateid":$("select#stdstateid").val()},
        type:"post",
        dataType:"JSON",
        success:function(json){
            if(json.n > 0) {
                alert("휴학신청이 반려되었습니다");
            }
        }
    });
});
```

```
<c:when test="${studentvo.stdstateid eq '2'}">휴학
    <input type="hidden" id="s_approve" name="s_approve" class="btn btn-outline-dark btn-sm mr-3 btn_s_approve" value="승인"/>
    <input type="hidden" id="s_reject" name="s_reject" class="btn btn-outline-dark btn-sm mr-3 btn_s_reject" value="반려"/>
    <input type="hidden" id="sR_approve" name="sR_approve" class="btn btn-outline-dark btn-sm mr-3 btn_sR_approve" value="승인"/>
    <input type="hidden" id="sR_reject" name="sR_reject" class="btn btn-outline-dark btn-sm mr-3 btn_sR_reject" value="반려"/>
</c:when>

<c:when test="${studentvo.stdstateid eq '6'}">휴학신청<br>
    <input type="button" id="s_approve" name="s_approve" class="btn btn-outline-dark btn-sm mr-3 btn_s_approve" value="승인"/>
    <input type="button" id="s_reject" name="s_reject" class="btn btn-outline-dark btn-sm mr-3 btn_s_reject" value="반려"/>
    <input type="hidden" id="sR_approve" name="sR_approve" class="btn btn-outline-dark btn-sm mr-3 btn_sR_approve" value="승인"/>
    <input type="hidden" id="sR_reject" name="sR_reject" class="btn btn-outline-dark btn-sm mr-3 btn_sR_reject" value="반려"/>
</c:when>
```

2. 담당 기능- 회원관리_회원조회_교원 조회

- 학생정보, 교원정보 각 클릭 시
학생 전체 목록, 교원 전체 목록 확인 가능.

- 검색 대상에 교원 이름, 교원 번호, 학과
코드, 학과명으로 조회 가능.

- 이름 클릭 시 교원 상세 정보 확인 가능.

회원전체 목록

학생정보 교원정보

검색대상 ▼ 조회

이름	교원번호	학과코드	학과명	근로상태
장미교	10076	11	컴퓨터공학과	재직
고용빈	10078	11	컴퓨터공학과	재직
장미노	10077	12	정보보안학과	재직
김지혜	10012	13	정보통신공학과	재직
준교수	10015	13	정보통신공학과	재직

1 2 3 [다음] [마지막]

고용빈 님의 회원 상세정보

교원번호	10078
학과코드	11
학과이름	컴퓨터공학과
근로상태코드	재직
이름	고용빈
이메일	gogo@naver.com
생년월일	1986-07-08 00:00:00.0
임용일자	2022-03-02 00:00:00.0
우편번호	
주소	
연락처	010-4547-7897

회원목록

Paging 처리

2. 담당 기능- 강의관리_교수강의개설

교수강의 개설 조회

검색대상 ▼

신청상태 ▼

조회

번호	학과	강의코드	강의명	개설학기	이수학점	정원	강의실	강의시간	신청상태
17	정보보안학과	12013	화이트해킹	8	3	40	자연과학관 101	수 2,3,4	등록완료
18	정보보안학과	12012	test12	5	3	29	자연과학관 201	금 4,5,6	등록완료
19	정보보안학과	12011	test11	4	3	30	자연과학관 201	금 1,2,3	등록완료
20	정보보안학과	12010	test10	8	3	50	자연과학관 201	목 4,5,6	승인대기중

[맨처음] [이전] 5 6 7 8 [다음] [마지막]

페이징 처리를 통해 전체 학생목록을 보여준다.

applylectureList 를 불러온 뒤 gyoOpenedList에 담는다.

```
<c:if test="${not empty requestScope.applylectureList}">
  <c:forEach var="gyoOpenedList" items="${requestScope.applylectureList}" varStatus="status">
    <tr>
      <td align="center">${gyoOpenedList.rno}</td>
      <td align="center">${gyoOpenedList.deptname}</td>
      <td id="subjectid" align="center">${gyoOpenedList.subjectid}</td>
      <td align="center">${gyoOpenedList.classname}</td>

      <td align="center">${gyoOpenedList.opensemester}</td>
      <td align="center">${gyoOpenedList.credit}</td>

      <td align="center">${gyoOpenedList.totalperson}</td>
      <td align="center">${gyoOpenedList.lctrld}</td>
      <td align="center">${gyoOpenedList.dayid}</td>
```

```
<td id="applystate" align="center"><p id="applystate">${gyoOpenedList.applystate}</p>
```

```
<c:choose>
  <c:when test="${gyoOpenedList.applystate == '승인대기중'}">
    <input type="button" id="approve" name="approve" class="btn btn-outline-dark btn-sm mr-3 btn_approve" value="승인"/>
    <input type="button" id="reject" name="reject" class="btn btn-outline-dark btn-sm mr-3 btn_reject" value="반려"/>
  </c:when>

  <c:when test="${gyoOpenedList.applystate == '신청서미비'}">
    <input type="hidden" id="approve" name="approve" class="btn btn-outline-dark btn-sm mr-3 btn_approve" value="승인"/>
    <input type="button" id="reject" name="reject" class="btn btn-outline-dark btn-sm mr-3 btn_reject" value="반려"/>
  </c:when>

  <c:when test="${gyoOpenedList.applystate == '0'}">[신청서 누락]
    <input type="hidden" id="approve" name="approve" class="btn btn-outline-dark btn-sm mr-3 btn_approve" value="승인"/>
    <input type="button" id="reject" name="reject" class="btn btn-outline-dark btn-sm mr-3 btn_reject" value="반려"/>
  </c:when>

  <c:otherwise>
    <input type="hidden" id="approve" name="approve" class="btn btn-outline-dark btn-sm mr-3 btn_approve" value="승인"/>
    <input type="hidden" id="reject" name="reject" class="btn btn-outline-dark btn-sm mr-3 btn_reject" value="반려"/>
  </c:otherwise>
</c:choose>
```

```
</td>
```

```
</tr>
```

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

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

```
<c:if test="${empty requestScope.applylectureList}">
```

```
<tr>
```

```
<td colspan="11" style="text-align: center;">강의가 존재하지 않습니다</td>
```

2. 담당 기능- 강의관리_교수강의개설

교수강의 개설 조회

검색대상 신청상태 조회

번호	학과	강의코드	강의명	개설학기	이수학점	정원	강의실	강의시간	신청상태
17	정보보안학과	12013	화이트해킹	8	3	40	자연과학관 101	수 2,3,4	등록완료
18	정보보안학과	12012	test12	5	3	29	자연과학관 201	금 4,5,6	등록완료
19	정보보안학과	12011	test11	4	3	30	자연과학관 201	금 1,2,3	등록완료
20	정보보안학과	12010	test10	8	3	50	자연과학관 201	목 4,5,6	승인대기중 <div>승인</div> <div>반려</div>

[맨처음] [이전] 5 6 7 8 [다음] [마지막]

JOIN을 통해 번호, 학과, 강의코드, 강의명, 개설학기, 이수학점, 정원, 강의실, 강의시간, 신청 상태를 보여준다.

또한 검색대상 또는 신청상태 선택 시 이에 맞는 결과를 보여준다.

```
select rno, deptname, subjectid, classname, credit
, opensemester, applystate, applydate, totalperson
, dayid, lctrld, seq_lectureplan
from
(
select row_number() over(order by applydate desc) AS rno, deptname, subjectid, classname, credit
, opensemester, applystate, applydate, totalperson
, dayid, lctrld, seq_lectureplan
from
(
select distinct A.deptname, B.subjectid, B.classname, B.credit
, B.opensemester, B.applystate, B.applydate, nvl(B.totalperson, NULL) as totalperson
, nvl(F.abvat, NULL) || ' ' || nvl(H.periodlist, NULL) AS dayid
, nvl(E.buildname, NULL) || ' ' || nvl(D.lctrld, NULL) AS lctrld
, nvl(I.seq_lectureplan, NULL) AS seq_lectureplan
from
(select majorid, deptname
from tbl_department) A
JOIN
(select majorid, subjectid, gyowonid, classname, credit,
opensemester, applyperson, totalperson,
applystate, applydate
from tbl_subject
) B
ON A.majorid = B.majorid
LEFT JOIN
(select subjectid, periodid, dayid, seq_lctrld
from tbl_lecturerroom_assign
) C
ON B.subjectid = C.subjectid
LEFT JOIN
(
select buildno, lctrld, seq_lctrld
from tbl_lecturerroom
) D
ON C.seq_lctrld = D.seq_lctrld
LEFT JOIN
(select buildno, buildname
from tbl_building
) E
on D.buildno = E.buildno
LEFT JOIN
(select dayid, abvat
from tbl_dayofweek
) F
on C.dayid = F.dayid
LEFT JOIN
(select seq_lectureplan, subjectid
from tbl_lectureplan) G
on B.subjectid = G.subjectid
LEFT JOIN
(select subjectid, LISTAGG(periodid, ',') WITHIN GROUP (ORDER BY periodid) AS periodlist
from (select subjectid, dayid, periodid, seq_lctrld
from tbl_lecturerroom_assign)
group by subjectid
order by subjectid) H
on B.subjectid = H.subjectid
LEFT JOIN
(select seq_lectureplan, subjectid
from tbl_lectureplan) I
on B.subjectid = I.subjectid
)
where rno between #{startRno} and #{endRno}
<if test='searchType_1 != "" and searchWord_1 != "">
where lower({searchType_1}) like '%'|| lower({searchWord_1}) ||'%' and lower(applystate) like '%'||lower({applystate})||'%'
</if>
)
<if test="applystate != ''">
where lower(applystate) like '%'||lower({applystate})||'%'
</if>
)
```


2. 담당 기능- 강의관리_교수강의개설

25	컴퓨터공학과	11034	컴퓨터비전	8	3	40	공학관 201	화 7,8,8	승인대기중
									<input type="button" value="승인"/>
									<input type="button" value="반려"/>
26	컴퓨터공학과	11034	컴퓨터비전	8	3	40	공학관 201	수 7,8,8	승인대기중
									<input type="button" value="승인"/>
									<input type="button" value="반려"/>
27	컴퓨터공학과	11015	디지털신호처리	6	3	50	공학관 201	화 4,5,6	신청반려
28	컴퓨터공학과	11006	인터넷 보안과 프라이버시	2	3	60	공학관 101	목 4,5,6	등록완료

localhost:9090 내용:
등록되었습니다

확인

36	컴퓨터공학과	11034	컴퓨터비전	8	3	40	공학관 201	수 7,8,8	등록완료
----	--------	-------	-------	---	---	----	---------	---------	------

Index를 이용하여 특정 값의 위치를 찾는다.

30	컴퓨터공학과	11017	데이터마이닝 개론	6	3	50	공학관 201	금 5,6,7	신청반려
----	--------	-------	-----------	---	---	----	---------	---------	------

```
<c:choose>
  <c:when test="${gyoOpenedList.applystate == '승인대기중'}">
    <input type="button" id="approve" name="approve" class="btn btn-outline-dark btn-sm mr-3 btn_approve" value="승인"/>
    <input type="button" id="reject" name="reject" class="btn btn-outline-dark btn-sm mr-3 btn_reject" value="반려"/>
  </c:when>
  <c:otherwise>
    <input type="hidden" id="approve" name="approve" class="btn btn-outline-dark btn-sm mr-3 btn_approve" value="승인"/>
    <input type="hidden" id="reject" name="reject" class="btn btn-outline-dark btn-sm mr-3 btn_reject" value="반려"/>
  </c:otherwise>
</c:choose>
```

<c:when>문, <c:otherwise>을 이용하여 applystate에 따라 버튼을 활성화/비활성화한다.

/// === 교수 강의 개설 (버튼 클릭시 업데이트) 신청상태변경 등록완료 === ///

```
@ResponseBody
@RequestMapping(value="/gyoOpenedUpdate.lmsfinal", method = {RequestMethod.POST })
public String gyoOpenedUpdate(HttpServletRequest request) {
```

```
String subjectid = request.getParameter("subjectid");
String applystate = request.getParameter("applystate");

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

paraMap.put("subjectid", subjectid);
paraMap.put("applystate", applystate);

int n = service.gyoOpenedUpdate(paraMap);

JSONObject jsonobj = new JSONObject();

jsonobj.put("n", n);

return jsonobj.toString();
}
```

```
const idx = $("input.btn_approve").index($(this));
//alert(idx);
const subject = $("td#subjectid").eq(idx).text();
//alert("확인용 subject=>" + subject);
const applystate = $("p#applystate").eq(idx).text();
//alert("확인용 applystate=>" + applystate);
```

<!-- 교원 수강신청 승인 -->

```
<update id="gyoOpenedUpdate" parameterType="String">
  update tbl_subject set applystate = '등록완료'
  where subjectid = #{subjectid}
</update>
```

승인버튼을 누를 경우 tbl_subject(강의테이블)에 있는 신청상태가 '승인대기중' 에서 '등록완료'로 바뀌고 반려버튼을 누를 경우 '승인 대기중' 에서 '신청반려'로 넘어간다.

```
$.ajax({
  url:"<%= ctxPath%>/gyoOpenedUpdate.lmsfinal",
  data:{"subjectid":subject, "applystate":$("select#applystate").val()},
  type:"post",
  dataType:"JSON",
  success:function(json){
    if(json.n > 0) {
      alert("등록되었습니다");
    }
  }
});
```

THANK YOU