

기쁨 다정영실하운지원

WGU e-campus

대학교 통합 정보 홈페이지

여성시대

목차 CONTENTS

01

팀 소개

02

개발배경

03

구현 목표

04

프로젝트
산출물

- 1) 요구사항 정의서
- 2) 단위업무 정의서
- 3) 유스케이스 다이어그램
- 4) ERD (논리/물리)
- 5) 테이블정의서

05

개발환경

06

스토리보드

07

프로젝트
일정

08

코드 리뷰

09

소감

01

팀 소개

팀 소개

01

식당 등록/수정/삭제
요일메뉴 등록/수정
식당 조회
식권 구매/조회



명다정

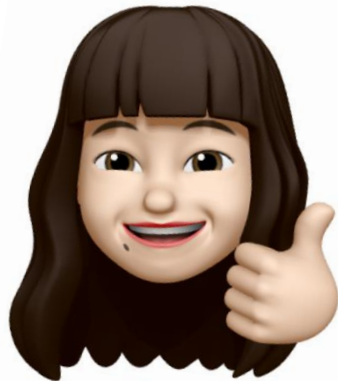
조장
과목 개설/조회
수강 신청/취소
성적 입력/수정/조회
강의평가 입력/조회
수강생 조회



최하윤

익명게시판 등록/수정/삭제
공지사항 등록/수정/삭제
학사일정 등록/수정/삭제

회원가입/로그인
정보수정
아이디/비밀번호 찾기
회원탈퇴



이영실



김기쁨



한지원

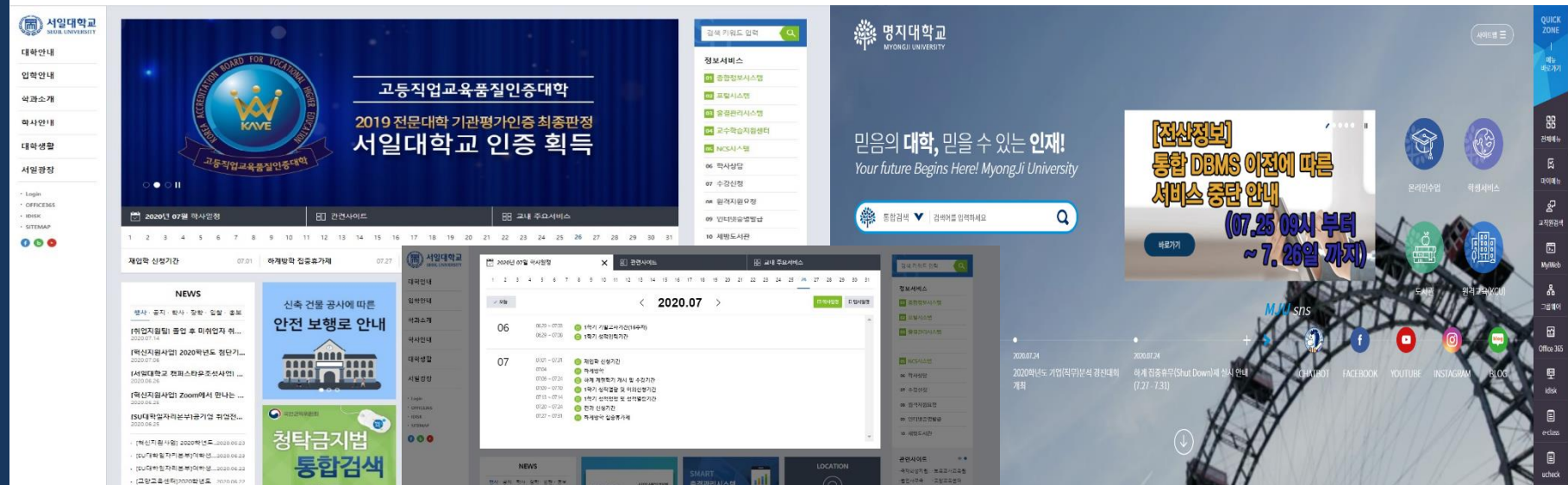
스쿨버스 등록/삭제
스쿨버스 예약/결제
스쿨버스 예약 현황확인

02

개발 배경

기존에 만들어진 대학교 홈페이지에서 불편한 기능들을 보완하여 학생들의 수강신청, 성적조회 서비스는 물론 더 나아가 학생들의 편리한 대학생활을 위한 학식 예약 서비스와 스쿨버스 좌석 예약 서비스도 함께 운영, 관리할 수 있는 시스템을 구현한다.

벤치마킹 사이트



03

구현 목표

저희 홈페이지는요?

편리한 UI

- 사용자 중심 인터페이스
- 한눈에 들어오는 학사정보

수강신청

- 어디에서나 간편하게 수강신청

성적조회

- 수강한 과목의 성적조회
- 강의평가와 건의사항 작성

예약 서비스

- 학식 예약과 스쿨버스 예약

04

프로젝트 산출물

1) 요구사항 정의서

번호	서비스(메뉴)	구분	요구사항	구분
1	회원가입	아이디 입력	5~10자의 영문 소문자, 숫자, 특수문자(-, _,)만 입력 가능.	교수, 학생
2		비밀번호	8~16자의 소문자, 숫자, 특수문자(!, @, #, \$, %, ^, &, *)가 모두 포함.	
3		필수 사항	회원가입 시 아이디, 비밀번호, 이름, 성별, 교수코드or학번, 학과(선택), 휴대폰번호, 이메일, 생년월일, 직책(교수, 학생)체크 필수로 입력	
4		선택 사항	주소	
5	로그인	아이디 찾기	교수코드or 학번, 이름, 등록된 전화번호 확인	
6	마이페이지	회원 수정 불가능	아이디, 학번, 학과, 직책(교수, 학생)체크	
7		회원 수정 가능	비밀번호, 이름, 성별, 휴대폰 번호, 이메일, 생년월일	
8		회원 탈퇴	회원 탈퇴 기능 제공 (개인 정보 보관 기간 : 1년, 관리자에게 문의 시 계정복구 가능하도록)	
9	학사 일정	등록, 수정, 삭제	관리자만 학사 일정을 등록, 수정, 삭제 가능	관리자
10		화면	월별로 메인페이지에서 확인 가능	교수, 학생
11	메인 배너	화면	배너 : 슬라이드 형식	
12	학식	메뉴 등록, 수정, 삭제	1. 오늘의 식단(매일 달라지는 메뉴) 2. 한식, 중식, 분식(매일 같은 메뉴)	관리자
13		메뉴 화면	오늘의 식단 : 캘린더 형식으로 확인 가능	학생
14		결제	메뉴사진, 가격, 체크박스 버튼 - 원하는 메뉴 선택 후 전자 결제 - 결제 완료 시 모바일 티켓 발급	
15		나의 학식 정보	모바일 티켓 확인 가능	
16	스쿨 버스	등록, 삭제	버스 정보 등록 (날짜, 노선, 좌석)	관리자
17		예약	날짜, 노선, 좌석 선택 후 예약 -예약 완료 시 모바일 티켓 발급	학생
18		나의 버스 정보	나의 예약 현황에서 예약한 버스의 정보 확인 가능	학생

19	수강 정보 페이지 (강의 페이지)	강의등록, 강의목 록 확인, 수강생 목록 확인	교수만 강의 등록가능. 본인의 강의 목록 확인 가능 신청한 학생의 목록 확인 가능	교수
20		수강 신청, 취소	학생은 등록된 강의 신청, 취소 가능	학생
21	성적 정보 페이지	성적 등록, 수정, 확인	교수만 성적을 등록, 수정 가능. 모든 학생들의 성적 확인 가능	교수
22		성적 열람	학생은 자신의 성적 열람 가능	학생
23	강의평가 페이지	강의 평가	학생은 수강한 강의에 대해 평가할 수 있다	학생
24		강의 평가 열람	학생들이 입력한 강의 평가 열람 가능	교수
25	게시판	공지사항	관리자만 입력, 수정, 삭제 가능	관리자
26		게시글	1. 게시글 입력 가능 (본인 글만 수정, 삭제 가능) -다른 사람에게는 게시글 작성자의 정보가 익명으로 출력 2. 모든 게시글 확인 가능.	관리자, 학생, 교수
27		게시글 관리	학생이 올린 게시글 삭제 가능	관리자
28		댓글	1. 댓글 입력 가능 (본인 댓글만 수정, 삭제 가능) -댓글 작성자의 정보를 선택 가능(익명 또는 사용자 이름) 2. 모든 댓글 확인 가능.	관리자, 학생, 교수

2) 단위업무 정의서

요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
회원가입	학생 교수	* 개인정보 동의 체크 * 아이디, 비밀번호, 이름, 성별, 학번, 학과, 휴대폰번호, 이메일, 생년월일, 직책(교수, 학생), 주소	1. 회원 가입 메뉴를 클릭한다. 2. [서비스 이용 약관 및 개인정보보호정책]에 동의한다. 3. [아이디, 비밀번호, 이름, 성별, 교수코드, 학번, 학과(체크), 휴대폰번호, 이메일, 생년월일, 직책(교수, 학생)체크, 주소를 입력한다. 4. 회원가입 버튼을 클릭한다. 5. 가입이 완료되면 "회원가입이 완료되었습니다" 라는 메시지를 띄운다.	2-1 동의를 거부할 시 "개인정보 수집 및 이용에 대한 안내 모두 동의해주세요" 라는 메시지를 띄운다. 3-1 필수사항 항목 중 하나라도 비어있을 경우 "필수 정보를 입력해주세요" 라는 메시지를 띄운다. 3-2 아이디가 중복되었을 경우 "중복된 아이디입니다." 라는 메시지를 띄운다. 3-3 아이디 또는 비밀번호가 형식에 맞지 않을 경우 "형식에 맞게 입력해 주세요" 라는 메시지를 띄운다. 3-4 존재하지 않는 코드와 학번을 입력한 경우 "존재하지 않는 번호입니다" 라는 메시지를 띄운다. 4 입력한 직책, 교수코드, 학번과 학과가 DB정보와 일치하면 자동으로 가입을 성공한다. 일치하지 않으면 "일치하는 정보를 찾을 수 없습니다" 라는 메시지를 띄운다.
요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
로그인	학생 교수 관리자	* 아이디, 비밀번호 * 아이디 찾기 버튼 * 비밀번호 찾기 버튼	1. 로그인 메뉴를 클릭한다. 2. [아이디, 비밀번호]를 입력한다. 3. 로그인 버튼을 클릭한다. 4. 사용자(학생, 교수) 로그인 성공 시 메인페이지로 연결된다. 5. 아이디 찾기 버튼을 누르면 교수코드or 학번, 이름, 전화번호를 입력할 수 있는 화면이 출력된다.	2-1 아이디와 비밀번호가 틀렸을 경우 "등록되지 않은 아이디입니다" 또는 "비밀번호가 일치하지 않습니다"라는 메시지를 띄운다. * 관리자가 로그인 했을 경우 - 관리자 메인페이지로 넘어가게 된다. 5-1 일치하는 정보가 없을 경우 "일치하는 정보가 없습니다" 라는 메시지를 띄운다. 5-2 일치하는 정보가 있으면 아이디를 보여주는 창이 출력된다.

요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
마이페이지	학생 교수	* 수정하기 버튼 클릭 * 비밀번호, 이름, 성별, 휴대폰번호, 이메일, 생년월일, 주소	1. 로그인 후 수정하기 버튼을 클릭한다. 2. 회원정보 수정 화면으로 연결된다. 3. [비밀번호, 비밀번호 확인, 이름, 성별, 휴대폰번호, 이메일, 생년월일, 주소] 중 수정하고 싶은 값을 입력한다. 4. 수정완료 버튼을 클릭한다. 5. 첫 마이페이지 화면으로 연결된다.	3-1 비밀번호가 형식에 맞지 않을 경우 "형식에 맞게 입력해주세요"라는 메시지를 띄운다 3-2 비밀번호와 비밀번호 확인이 일치하지 않을 경우 "비밀번호를 다시 확인해주세요"라는 메시지를 띄운다. 3-3 필수항목 중 하나라도 비어있을 경우 "모든 항목을 채워주세요" 라는 메시지를 띄운다.
		* 탈퇴하기 버튼 클릭 * 아이디, 비밀번호	1. 로그인 후 탈퇴하기 버튼을 클릭한다. 2. 아이디와 비밀번호를 입력한다. 3. 정말 탈퇴하시겠습니까?라는 팝업 메시지에 "예"를 누른다. 4. 탈퇴가 완료된 후 첫 메인페이지 화면으로 연결된다.	2-1 아이디와 비밀번호가 일치하지 않을 경우 "일치하지 않습니다. 다시 입력해주세요" 라는 메시지를 띄운다. 3-1 "아니오" 버튼을 눌렀을 경우 마이페이지 화면으로 연결되고 삭제가 되지 않는다.
요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
학사일정	관리자	* 등록 버튼, 수정 버튼 * 날짜, 시간, 내용 * 확인 버튼	<학사 일정 등록> 1. 학사 일정 메뉴를 클릭한다. 2. 등록 버튼을 클릭한다. 3. [날짜, 시간, 내용]을 입력한다. 4. 확인 버튼 클릭한다.	3-1 날짜, 시간, 내용 중 하나라도 입력되지 않은 경우 "모든 항목을 입력해주세요"라는 메시지를 띄운다
			<학사 일정 수정> 1. 학사 일정 메뉴를 클릭한다. 2. 수정 버튼을 클릭한다. 3. 기존의 [날짜, 시간, 내용]을 수정한다. 4. 확인 버튼 클릭한다.	3-1 날짜, 시간, 내용 중 하나라도 입력되지 않은 경우 "모든 항목을 입력해주세요"라는 메시지를 띄운다
		* 삭제 버튼	<학사 일정 삭제> 1. 학사 일정 메뉴를 클릭한다. 2. 삭제 할 학사 일정을 선택한다. 3. 삭제 버튼을 클릭한다.	2-1 항목이 선택되지 않았을 경우 "삭제 할 항목을 선택해주세요"라는 메시지를 띄운다.

요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
학식	학생 교수	* 메뉴 보기 버튼 * 체크 박스(메뉴) * 결제 버튼	1. 학식 메뉴 보기 버튼을 클릭한다. 2. 원하는 날짜의 메뉴를 확인한다. 3. 원하는 메뉴의 체크박스에 체크한다. 4. 결제하기 버튼을 클릭한다. 5. 결제에 성공한다. 6. 모바일 티켓이 발급된다.	3-1 어떤 메뉴에도 체크가 되지 않았을 경우 "메뉴를 선택해주세요"라는 메시지를 띄운다 5-1 결제에 실패했을 경우 "결제에 실패했습니다"라는 메시지를 띄운다.
		*모바일 티켓 확인 버튼	1. 모바일 티켓 확인 버튼을 누른다. 2. 결제한 모바일 티켓에 대한 목록을 출력한다.	
	관리자	* 등록버튼, 수정 버튼 * 메뉴, 메뉴사진, 가격 * 확인버튼	1. 메뉴 등록하기 버튼을 클릭한다. 2. [메뉴, 메뉴사진, 가격]을 입력한다. 3. 확인 버튼을 누른다.	2-1 메뉴,메뉴사진,가격 중 하나라도 공백이 있을 경우 각 항목에 맞춰 '항목을 입력해주세요' 라는 메시지를 띄운다.
			1. 메뉴 수정하기 버튼을 클릭한다. 2. [메뉴, 메뉴사진, 가격]을 입력한다. 3. 확인 버튼을 누른다.	2-1 메뉴,메뉴사진,가격 중 하나라도 공백이 있을 경우 각 항목에 맞춰 '항목을 입력해주세요' 라는 메시지를 띄운다.
		* 삭제 버튼	1. 메뉴 삭제하기 버튼을 클릭한다. 2. 삭제할 메뉴를 선택한다. 3. 메뉴가 삭제된다.	2-1 삭제 할 메뉴를 선택하지 않았을 경우 "삭제 할 메뉴를 선택해주세요" 라는 메시지를 띄운다.

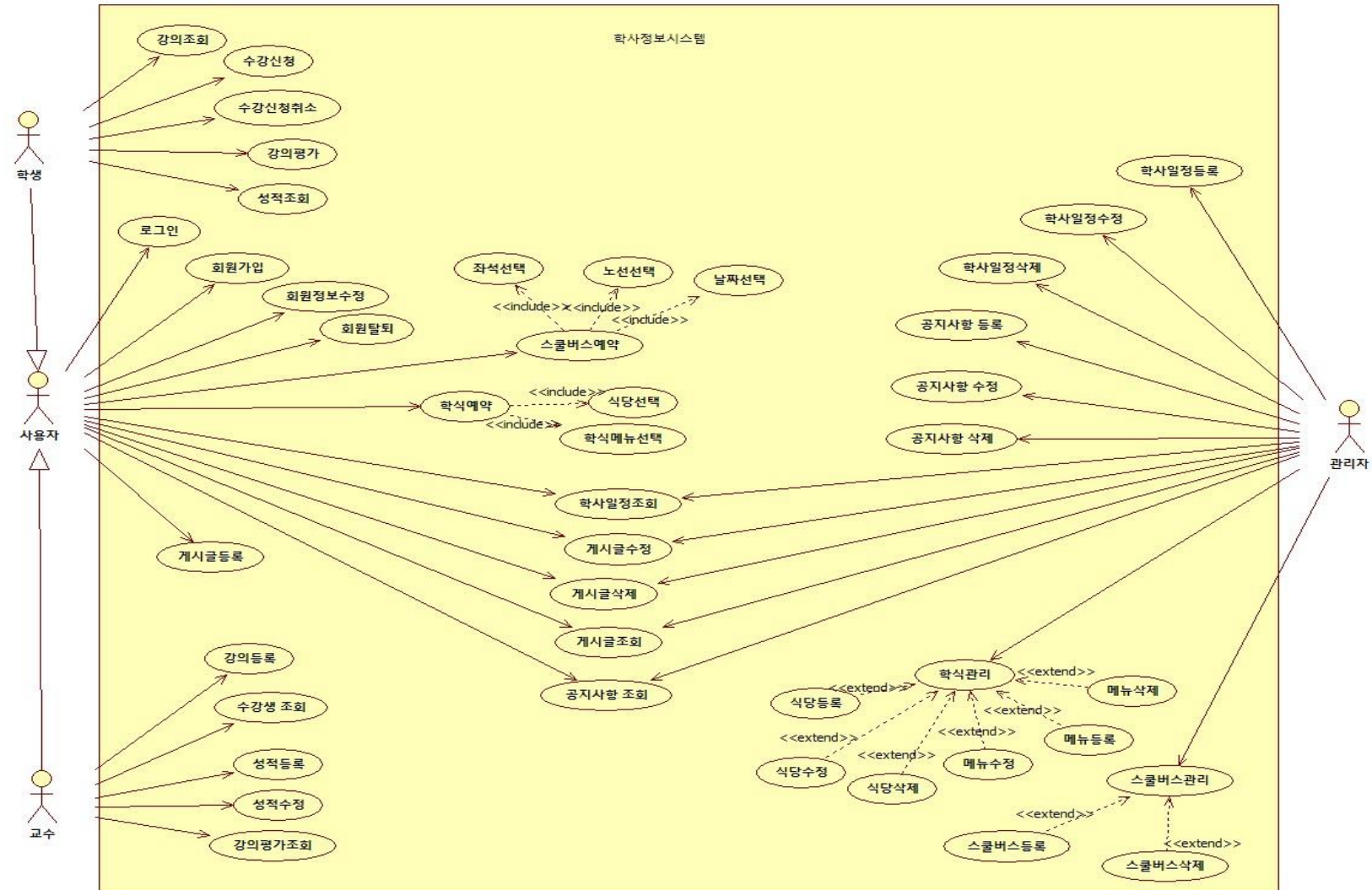
요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
스쿨버스	관리자	* 스쿨버스 등록 버튼 * 날짜, 노선	1. 스쿨버스 등록하기 버튼을 클릭한다. 2. [날짜, 노선, 메모]를 입력한다. 3. 스쿨버스 노선이 추가된다.	2-1 날짜, 노선, 메모 중 하나라도 입력되지 않은 경우 "입력란을 작성해주세요"라는 안내문구가 뜬다.
		* 스쿨버스 삭제 버튼	1. 스쿨버스 삭제하기 버튼을 클릭한다. 2. 삭제 할 버스의 정보를 입력한다. 3. 스쿨버스 노선이 삭제된다.	2-1 DB에 없는 스쿨버스가 입력되었다면 "입력란을 작성해주세요"라는 안내문구가 뜬다.
	학생 교수	* 스쿨버스 예약 버튼 * 시간, 노선, 좌석	1. 스쿨버스 예약 버튼을 클릭한다. 2. 날짜와 노선, 좌석을 선택한다. 3. 모바일 티켓이 발급된다.	2-1 선택된 좌석을 확인하기 위해 "~번 좌석이 선택되었습니다"라는 메시지를 띄운다. 3-1 예약에 실패했을 경우 예약 실패 오류 페이지 창으로 이동한다.
		* 마이페이지 예약 현황	1. 마이페이지에서 나의 버스 예약 현황 버튼을 클릭한다. 2. 예약한 모바일 티켓에 대한 목록을 출력한다.	2-1 예약한 버스가 없을 경우 "존재하는 버스 예약이 없습니다"라는 문구를 띄운다.

요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
강의 페이지	교수	* 과목 개설 버튼	<과목 개설> 1. 과목 개설 메뉴를 클릭한다. 2. [전공/교양, 학과, 시간, 학년, 과목명, 학점, 정원]을 입력한다 3. 등록 버튼 클릭한다.	3-1 과목 개설 시, 같은 시간에 이미 개설된 과목이 있는 경우 "같은 시간에 개설된 과목이 존재합니다"라는 메시지를 띄운다
		* 수강생 조회 버튼	<수강생 조회> 1. 과목 개설 메뉴를 클릭한다. 2. 본인의 강의 목록이 출력된다. 3. 조회하고 싶은 강의의 수강생 조회 버튼을 클릭한다. 4. 해당 강의의 수강생 목록이 출력된다.	
	학생	* 조회 버튼 * [전공/교양] 탭 * 학과 * 과목명 * 수강하기 버튼	<수강 신청> 1. 수강 신청 버튼을 클릭한다 2. 모든 과목이 출력된다. 3. 수강하고 싶은 과목의 신청 버튼을 누른다.	2-1 2. [전공/교양] 탭, 학과 또는 과목명을 입력한다. 3. 조건에 맞는 과목이 출력된다. 4. 수강하고 싶은 과목의 신청 버튼을 누른다.
		* 수강 취소 버튼	<수강 확인> 1. 사용자가 신청한 과목 목록이 출력된다. 2. 취소하고 싶은 과목에 취소 버튼을 누른다.	

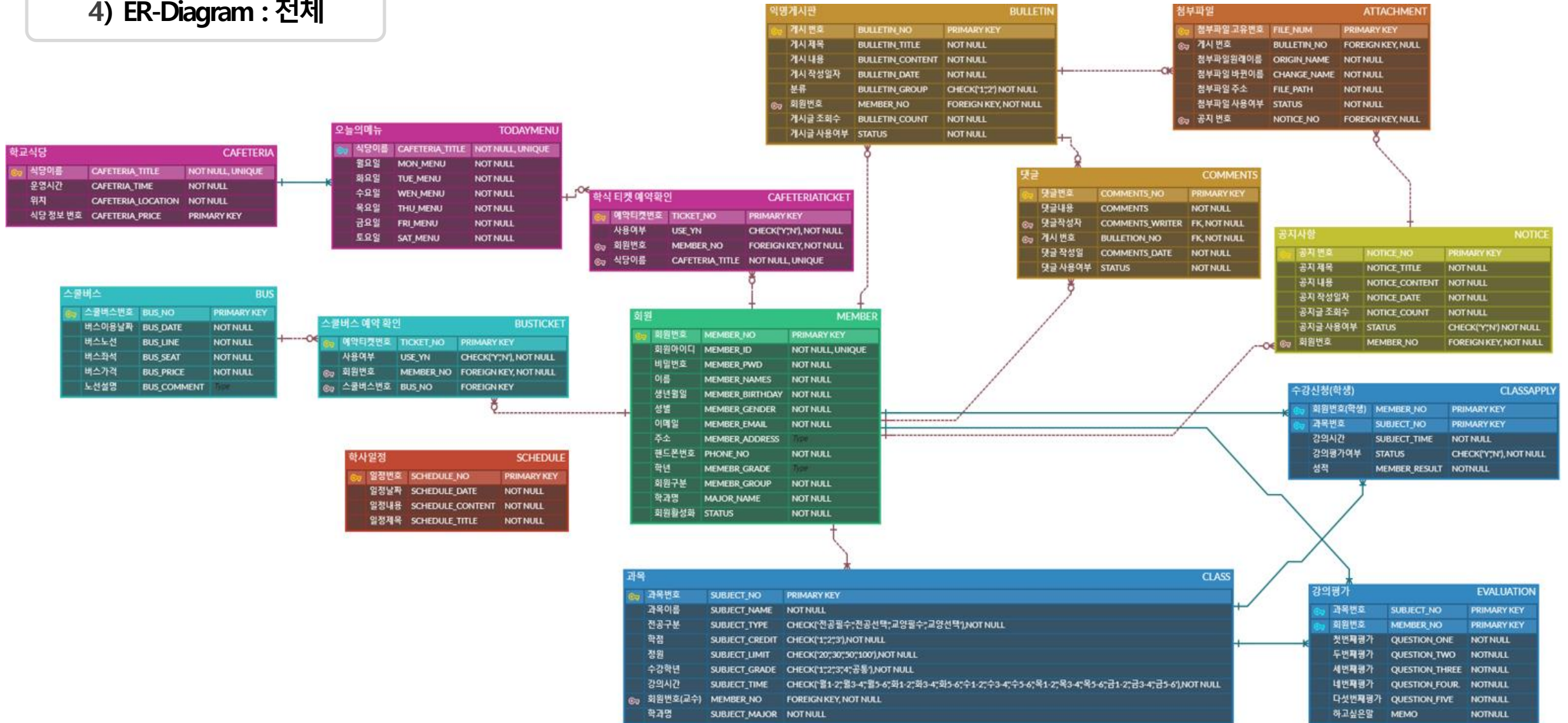
요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
성적 페이지	교수	<ul style="list-style-type: none"> * 수강생 목록 버튼 * 성적 * 성적 입력 버튼 	<성적 등록&수정> 1. 성적 입력 버튼을 클릭한다. 2. 본인의 강의 목록을 출력한다. 3. 성적을 등록할 강의를 선택한다. 4. 수강생 목록을 출력한다. 5. 성적을 입력한다. 6. 성적 입력 버튼을 누른다.	
	학생		<성적 확인> 1. 이번학기 성적조회 버튼을 클릭한다. 2. 수강하고 있는 과목의 성적이 출력된다.	
요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
강의평가 페이지	학생	<ul style="list-style-type: none"> * 강의 평가 버튼 * 강의 만족도, 건의사항 * 제출하기 버튼 	<강의 평가 등록> 1. 강의 평가 버튼을 누른다 2. 해당 학생이 수강하고 있는 강의 목록을 출력한다. 3. 강의 평가를 등록할 강의를 선택한다. 4. [강의 만족도, 건의사항]을 작성한다. 5. 제출하기 버튼을 누른다.	4-1 강의의 만족도, 건의사항 중 하나라도 공백이 있을 경우 "모든 항목을 입력해주세요"라는 메시지를 띄운다.
	교수	<ul style="list-style-type: none"> * 평가 확인 버튼 	<강의 평가 열람> 1. 평가 확인 버튼을 누른다. 2. 해당 교수의 강의 목록을 출력한다. 3. 열람할 강의를 선택한다. 4. 학생이 작성한 강의 평가를 출력한다.	

요구사항명	사용자	기능입력정보	예상 흐름도	대안 흐름도
게시판	관리자	* 작성 버튼 * 제목, 내용 * 확인 버튼	<공지사항 작성> 1. 공지사항 작성 버튼을 누른다. 2. 제목, 내용을 입력한다. 3. 확인 버튼을 누른다.	2-1 제목, 내용 중 하나라도 비어있을 경우 "항목을 모두 채워주세요"라는 메시지를 띄운다.
		* 수정 버튼 * 제목, 내용 * 확인 버튼	<공지사항 수정> 1. 수정 버튼을 누른다. 2. 제목, 내용을 입력한다. 3. 확인 버튼을 누른다.	
		* 삭제 버튼	<공지사항 삭제> 1. 삭제 버튼을 누른다	
	학생 교수	* 등록 버튼 * 카테고리, 제목, 내용 * 확인 버튼	<게시글 등록> 1. 등록 버튼을 누른다. 2. 카테고리 선택, 제목, 내용을 입력한다. 3. 확인 버튼을 누른다.	2-1 카테고리 선택, 제목, 내용 중 하나라도 공백이 있을 경우 "모든 항목을 채워주세요"라는 메시지를 띄운다,
		* 수정 버튼 * 카테고리, 제목, 내용 * 확인 버튼	<게시글 수정 - 본인 글만 수정 가능> 1. 수정 버튼을 누른다. 2. 카테고리 선택, 제목, 내용을 입력한다. 3. 확인 버튼을 누른다.	2-1 카테고리 선택, 제목, 내용 중 하나라도 공백이 있을 경우 "모든 항목을 채워주세요"라는 메시지를 띄운다,
		* 삭제 버튼	<게시글 삭제 - 본인 글만 삭제 가능> 1. 삭제 버튼을 누른다	
	관리자	* 삭제 버튼	<게시글 삭제 - 모든 게시글 삭제 가능> 1. 삭제 버튼을 누른다	
	관리자 학생 교수	* 내용 * 댓글 작성자의 정보 * 확인 버튼	<댓글 입력> 1. 내용을 입력한다. 2. 댓글 작성자의 정보(익명 또는 사용자 이름)를 체크한다. 3. 확인 버튼을 누른다.	1-1 내용이 비어있을 경우 "내용을 입력해주세요"라는 메시지를 띄운다. 2-1 댓글 작성자의 정보가 체크되지 않은 경우 "익명 또는 사용자 이름에 체크해주세요"라는 메시지를 띄운다.
		* 수정 버튼 * 내용 * 댓글 작성자의 정보 * 확인 버튼	<댓글 수정 - 본인 댓글만 수정 가능> 1. 수정 버튼을 누른다. 2. 내용을 입력한다. 3. 댓글 작성자의 정보(익명 또는 사용자 이름)를 체크한다. 4. 확인 버튼을 누른다.	2-1 내용이 비어있을 경우 "내용을 입력해주세요"라는 메시지를 띄운다. 3-1 댓글 작성자의 정보가 체크되지 않은 경우 "익명 또는 사용자 이름에 체크해주세요"라는 메시지를 띄운다.
		* 삭제 버튼	<댓글 삭제 - 본인 댓글만 삭제 가능> 1. 삭제 버튼을 누른다.	

3) 유스케이스 다이어그램



4) ER-Diagram : 전체



5) 테이블 정의서 : 회원(학생/교수)

회원(학생/교수) MEMBER						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
회원번호	MEMBER_NO	VARCHAR2(50)	NOTNULL	YES	NO	
회원아이디	MEMBER_ID	VARCHAR2(50)	NOTNULL	NO	NO	
비밀번호	MEMBER_PWD	VARCHAR2(300)	NOTNULL	NO	NO	
이름	MEMBER_NAMES	VARCHAR2(50)	NOTNULL	NO	NO	
생년월일	MEMBER_BIRTHDAY	VARCHAR2(50)	NOTNULL	NO	NO	
성별	MEMBER_GENDER	VARCHAR2(50)	NOTNULL	NO	NO	
이메일	MEMBER_EMAIL	VARCHAR2(50)	NOTNULL	NO	NO	
주소	MEMBER_ADDRESS	VARCHAR2(50)	NULL	NO	NO	
핸드폰번호	PHONE_NO	VARCHAR2(50)	NOTNULL	NO	NO	
학년	MEMBER_GRADE	VARCHAR2(10)	NULL	NO	NO	
구분	MEMBER_GROUP	VARCHAR2(10)	NOTNULL	NO	NO	
학과명	MAJOR_NAME	VARCHAR2(50)	NOTNULL	NO	NO	
회원활성화	STATUS	VARCHAR2(1)	NOTNULL	NO	NO	

과목 SUBJECT						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
과목번호	SUBJECT_NO	VARCHAR2(50)	NOTNULL	YES	NO	
회원번호(교수)	MEMBER_NO	VARCHAR2(50)	NOTNULL	NO	YES	MEMBER
과목이름	SUBJECT_NAME	VARCHAR2(50)	NOTNULL	NO	NO	
전공구분	SUBJECT_TYPE	VARCHAR2(50)	NOTNULL	NO	NO	
학점	SUBJECT_CREDIT	NUMBER(10)	NOTNULL	NO	NO	
정원	SUBJECT_LIMIT	NUMBER(10)	NOTNULL	NO	NO	
수강학년	SUBJECT_GRADE	VARCHAR2(10)	NOTNULL	NO	NO	
강의시간	SUBJECT_TIME	VARCHAR2(50)	NOTNULL	NO	NO	
학과명	SUBJECT_MAJOR	VARCHAR2(50)	Null	NO	NO	

수강신청(학생) SUBJECTAPPLY						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
회원번호	MEMBER_NO	VARCHAR2(50)	NOTNULL	YES	YES	MEMBER
과목번호	SUBJECT_NO	VARCHAR2(50)	NOTNULL	YES	YES	SUBJECT
강의시간	SUBJECT_TIME	VARCHAR2(50)	NOTNULL	NO	NO	
강의평가여부	STATUS	VARCHAR2(10)	NOTNULL	NO	NO	
성적	MEMBER_RESULT	VARCHAR2(10)	NOTNULL	NO	NO	

강의평가 EVALUATION						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
회원번호	MEMBER_NO	VARCHAR2(50)	NOTNULL	YES	YES	MEMBER
과목번호	SUBJECT_NO	VARCHAR2(50)	NOTNULL	YES	YES	SUBJECT
첫번째평가	QUESTION_ONE	VARCHAR2(50)	NOTNULL	NO	NO	
두번째평가	QUESTION_TWO	VARCHAR2(50)	NOTNULL	NO	NO	
세번째평가	QUESTION_THREE	VARCHAR2(50)	NOTNULL	NO	NO	
네번째평가	QUESTION_FOUR	VARCHAR2(50)	NOTNULL	NO	NO	
다섯번째평가	QUESTION_FIVE	VARCHAR2(50)	NOTNULL	NO	NO	
하고싶은말	MEMO	VARCHAR2(1000)	NOTNULL	NO	NO	

공지사항 NOTICE						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
공지 번호	NOTICE_NO	VARCHAR2(50)	NOTNULL	YES	NO	
공지 제목	NOTICE_TITLE	VARCHAR2(300)	NOTNULL	NO	NO	
공지 내용	NOTICE_CONTENT	VARCHAR2(3000)	NOTNULL	NO	NO	
공지 작성일자	NOTICE_DATE	DATE	NOTNULL	NO	NO	
공지글 조회수	NOTICE_COUNT	NUMBER	NOTNULL	NO	NO	
공지글 사용여부	STATUS	VARCHAR2(1)	NOTNULL	NO	NO	
회원번호(관리자번호)	MEMBER_NO	VARCHAR2(50)	NOTNULL	YES	NO	MEMBER

댓글 COMMENTS						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
댓글번호	COMMENTS_NO	VARCHAR2(50)	NOTNULL	YES	NO	
댓글내용	COMMENTS	VARCHAR2(1000)	NOTNULL	NO	NO	
댓글작성일	COMMENTS_DATE	DATE	NOTNULL	NO	NO	
댓글 사용여부	STATUS	VARCHAR2(1)	NOTNULL	NO	NO	
게시 번호	BULLETIN_NO	VARCHAR2(50)	NOTNULL	NO	YES	BULLETIN
댓글 작성자	MEMBER_NO	VARCHAR2(50)	NULL	NO	YES	MEMBER

익명게시판 BULLETIN						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
게시 번호	BULLETIN_NO	VARCHAR2(50)	NOTNULL	YES	NO	
게시 제목	BULLETIN_TITLE	VARCHAR2(300)	NOTNULL	NO	NO	
게시 내용	BULLETIN_CONTENT	VARCHAR2(3000)	NOTNULL	NO	NO	
게시 작성일자	BULLETIN_DATE	DATE	NOTNULL	NO	NO	
게시글 조회수	BULLETIN_COUNT	NUMBER	NOTNULL	NO	NO	
분류	BULLETIN_GROUP	NUMBER	NOTNULL	NO	NO	
게시글 사용여부	STATUS	VARCHAR2(1)	NOTNULL	NO	NO	
회원번호	MEMBER_NO	VARCHAR2(50)	NOTNULL	YES	NO	MEMBER

5) 테이블 정의서 : 관리자

스쿨버스 BUS						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
스쿨버스 번호	BUS_NO	NUMBER	NOTNULL	YES	NO	
버스이용날짜	BUS_DATE	DATE	NOTNULL	NO	NO	
버스노선	BUS_LINE	VARCHAR2(50)	NOTNULL	NO	NO	
버스좌석	BUS_SEAT	VARCHAR2(50)	NOTNULL	NO	NO	
버스가격	BUS_PRICE	NUMBER(10)	NOTNULL	NO	NO	
노선설명	BUS_COMMENT	VARCHAR2(1000)	NULL	NO	NO	

스쿨버스 티켓 예약확인 BUSTICKET						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
예약티켓번호	TICKET_NO	NUMBER(10)	NOTNULL	YES	NO	
사용여부	USE_YN	VARCHAR2(10)	NOTNULL	NO	NO	
회원번호	MEMBER_NO	VARCHAR2(50)	NOTNULL	NO	YES	MEMBER
스쿨버스 번호	BUS_NO	NUMBER(10)	NOTNULL	NO	YES	BUS

첨부파일 ATTACHMENT						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
첨부파일 고유번호	FILE_ID	VARCHAR2(50)	NOTNULL	YES	NO	
게시번호(익명)	BULLETIN_NO	VARCHAR2(50)	NULL	NO	YES	BULLETIN
공지번호(공지)	NOTICE_NO	VARCHAR2(50)	NULL	NO	YES	NOTICE
첨부파일 원래이름	ORIGIN_NAME	VARCHAR2(300)	NOTNULL	NO	NO	
첨부파일 바뀐이름	CHANG_NAME	VARCHAR2(300)	NOTNULL	NO	NO	
첨부파일 주소	FILE_PATH	VARCHAR2(1000)	NOTNULL	NO	NO	
첨부파일 사용여부	STATUS	VARCHAR2(1)	NOTNULL	NO	NO	

학교식당 CAFETERIA						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
식당이름	CAFETERIA_TITLE	VARCHAR2(50)	NOTNULL	YES	NO	
운영시간	CAFETERIA_TIME	VARCHAR2(50)	NOTNULL	NO	NO	
위치	CAFETERIA_LOCATION	VARCHAR2(50)	NOTNULL	NO	NO	
가격	CAFETERIA_PRICE	NUMBER	NOTNULL	NO	NO	

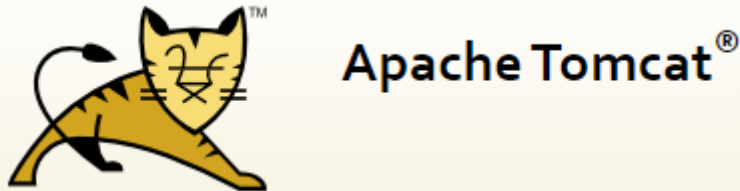
오늘의 메뉴 TODAYMENU						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
식당이름	CAFETERIA_TITLE	VARCHAR2(50)	NOTNULL	YES	YES	
월요일	MON_MENU	VARCHAR2(300)	NOTNULL	NO	NO	
화요일	TUE_MENU	VARCHAR2(300)	NOTNULL	NO	NO	
수요일	WEN_MENU	VARCHAR2(300)	NOTNULL	NO	NO	
목요일	THU_MENU	VARCHAR2(300)	NOTNULL	NO	NO	
금요일	FRI_MENU	VARCHAR2(300)	NOTNULL	NO	NO	
토요일	SAT_MENU	VARCHAR2(300)	NOTNULL	NO	NO	

학식 티켓 예약확인 CAFETERIATICKET						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
예약티켓번호	TICKET_NO	NUMBER	NOTNULL	YES	NO	
사용여부	USE_YN	VARCHAR2(10)	NOTNULL	NO	NO	
회원번호	MEMBER_NO	VARCHAR2(50)	NOTNULL	NO	YES	MEMBER
식당이름	CAFETERIA_TITLE	VARCHAR2(50)	NULL	NO	YES	TODAYMENU

학사일정 SCHEDULE						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
일정번호	SCHEDULE_NO	VARCHAR2(50)	NOTNULL	YES	NO	
일정날짜	SCHEDULE_DATE	DATE	NOTNULL	NO	NO	
일정제목	SCHEDULE_TITLE	VARCHAR2(300)	NOTNULL	NO	NO	
일정내용	SCHEDULE_CONTENT	VARCHAR2(1000)	NOTNULL	NO	NO	

05

개발 환경



통합개발환경 : Eclipse 4.15.0

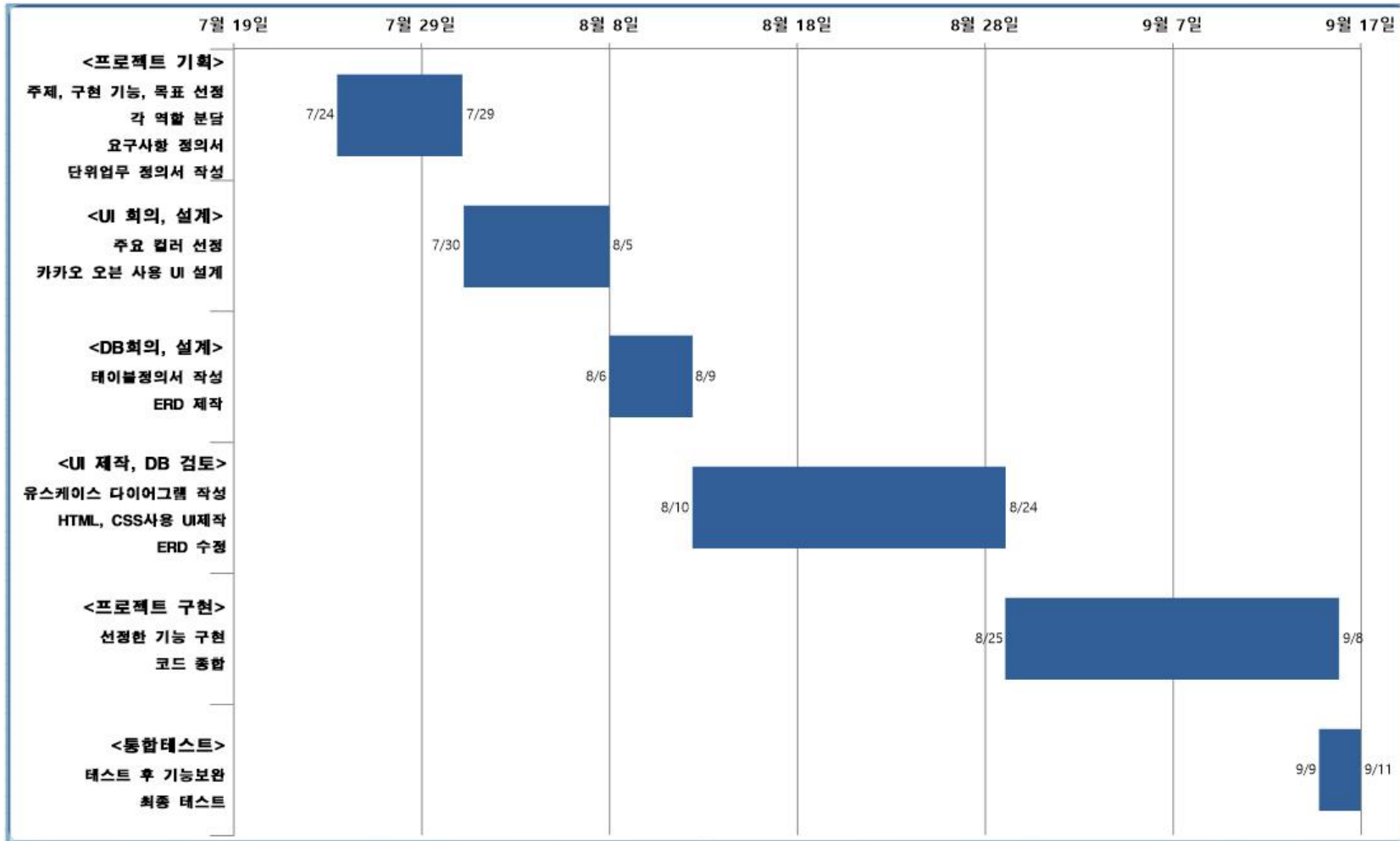
DB : Oracle 11g

Server : Apache Tomcat 9.0

07

프로젝트 일정

간트 차트



08

코드 리뷰

코드 리뷰

08

코드리뷰 1

1

```
$(function() {
    $('.cafeTabs').children().click(function() {
        $('.cafeTabs').children().css({ "background" : "rgb(236, 236, 236)", "color" : "black"});
        $(this).css({ "background" : "rgb(56, 103, 166)", "color" : "white"});

        $('#listName').val($(this).text());
        var listName= $('#listName').val(); // listName에 현재 내가 누른 버튼의 입력 값을 저장

        $.ajax({
            url : 'insertList.cafe',
            type : 'get',
            data : {listName:listName, insertCafeteria:1},
            success : function(data) {
                console.log('업데이트 서버 성공 시 호출되는 함수')
                console.log("listName : " + data)

                var locationC = "";
                var timeC = "";
                var priceC = "";
                if(data != null){
                    locationC = data.cafeteriaLocation;
                    timeC = data.cafeteriaTime;
                    priceC = data.cafeteriaPrice;
                    console.log("locationC : " + locationC)
                    console.log("timeC : " + timeC)
                    console.log("priceC : " + priceC)

                    $('#cafePlace').text(locationC);
                    $('#cafeTime').text(timeC);
                    $('#cafePrice').text(priceC);
                }
            }
        })
    })
})
```

2

```
if(insertCafeteria.equals("5")) { // 5가 들어오면 새로운 식당리스트 생성
    String cafeteriaTitle = request.getParameter("ca");
    String cafeteriaTime = request.getParameter("cafeTime");
    String cafeteriaLocation = request.getParameter("cafePlace");
    int cafeteriaPrice = Integer.parseInt(request.getParameter("cafePrice"));

    Cafeteria cafeteria = new Cafeteria(cafeteriaTitle, cafeteriaTime, cafeteriaLocation, cafeteriaPrice);
    int result = new CafeteriaService().insertCafeteria(cafeteria);

    if(result > 0) {
        // result 추가 후 새로운 식당 리스트 받아오는 작업
        ArrayList<Cafeteria> cList = new CafeteriaService().selectList();
        request.setAttribute("cList", cList);
        request.getRequestDispatcher("WEB-INF/views/cafeteria/insertCafeteria.jsp").forward(request, response);
    } else {
        request.setAttribute("msg", "식당등록 실패");
        request.getRequestDispatcher("WEB-INF/views/common/errorPage.jsp").forward(request, response);
    }
} else if(insertCafeteria.equals("3")) { // 3이 들어오면 식당정보 해당 내용으로 수정
    String cafeteriaTitle = request.getParameter("listName");
    String cafeteriaTime = request.getParameter("cafeTime");
    String cafeteriaLocation = request.getParameter("cafePlace");
    int cafeteriaPrice = Integer.parseInt(request.getParameter("cafePrice"));

    Cafeteria cafeteria = new Cafeteria(cafeteriaTitle, cafeteriaTime, cafeteriaLocation, cafeteriaPrice);
    int result = new CafeteriaService().updateCafeteria(cafeteria);

    if(result > 0) {
        ArrayList<Cafeteria> cList = new CafeteriaService().selectList();
        request.setAttribute("cList", cList);
        request.getRequestDispatcher("WEB-INF/views/cafeteria/insertCafeteria.jsp").forward(request, response);
    }
}
```

3

```
$('#btnAdd').click(function(){
    var insertCafeteria = $('#insertCafeteria').val(5)
    $('#cafePlace').val("");
    $('#cafeTime').val("");
    $('#cafePrice').val("");

    $.ajax({
        url : 'insertList.cafe',
        type : 'get',
        data : {insertCafeteria:insertCafeteria},
        success : function(insertCafeteria) {
            console.log('서버 성공 시 호출되는 함수')
        },
        error : function(data) {
            console.log('서버 실패 시 호출되는 함수')
        }
    })
})
```

한 버튼에 listName이
어떤 값이
들어오냐에 따라
조회,추가,수정,삭제를
구현

코드리뷰 1

```
if(tMenu == null) { // 요일메뉴 정보가 없으면 insert

    int result = new TodayMenuService().insertTodayMenu(todayMenu);

    if(result > 0) {
        // 변경 후 식당리스트 다시 불러오기
        ArrayList<Cafeteria> cList = new CafeteriaService().selectList();
        request.setAttribute("cList", cList);
        request.getRequestDispatcher("WEB-INF/views/cafeteria/insertCafeteria.jsp").forward(request, response);
    } else {
        request.setAttribute("msg", "오늘의 식단 등록 실패");
        request.getRequestDispatcher("WEB-INF/views/common/errorPage.jsp").forward(request, response);
    }
} else { // 요일 메뉴 정보가 있으면 update
    int result = new TodayMenuService().updateTodayMenu(todayMenu);

    if(result > 0) {
        ArrayList<Cafeteria> cList = new CafeteriaService().selectList();
        request.setAttribute("cList", cList);
        request.getRequestDispatcher("WEB-INF/views/cafeteria/insertCafeteria.jsp").forward(request, response);
    } else {
        request.setAttribute("msg", "오늘의 식단 수정 실패");
        request.getRequestDispatcher("WEB-INF/views/common/errorPage.jsp").forward(request, response);
    }
}
```

요일 메뉴 정보가
없으면 추가 구현
있으면 수정 구현

코드리뷰 2

회원이입시 정규식을 이용한
유효성검사 script코드를 구현하였다.

```
<!-- 유효성 체크 -->
<!-- 회원구분에서 교수를 선택했을때는 학년 선택비활성화 -->
<!--교수를 선택했을때는 다시 학년이 없어지도록 --> */
$('#choice').change(function(){
    var state = $('#choice option:selected').val();
    if(state == '학생'){
        $('#userGradeChoice').show();
    } else {
        $('#userGradeChoice').hide();
    }
});

//onsubmit : 공백확인
//입력한 유효성검사 1. 정규식 변수 만들기
//모든 공백 체크 정규식
var empJ = /\s/g;
//아이디 5~10자이내의 영소문자, 숫자, 특수문자(-_)
var idJ = /^[a-z0-9_]{5,10}$/;
//비밀번호 8~16자이내의 영소문자, 숫자, 특수문자(!@#%&* )허용"
var pwdJ= /^[a-z0-9!@#%&*]{8,16}$/;
//이름정규식 가~힣, 한글로 이뤄진 문자만으로 2~6자리이름
var nameJ= /^[가-힣]{2,6}$/;
//이메일 검사
var emailJ= /^[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*.[a-z]{2,3}$/i;
//전화번호 -를 뺀 010으로 시작하는 12자리
var phoneJ= /(010)([1-9]{1}[0-9]{2,3})([0-9]{4})$/;

//교수 코드, 학번 S나P로 시작해야하고 숫자8자리여야한다.
var userNoJ= /([SP])([0-9]{8})$/;

//아이디유효성검사
$('#userId').blur(function(){
    if(idJ.test($(this).val())){
        //콘솔창에 확인하기
        console.log(idJ.test($(this).val()));
        $('#userId').text('');
        $('#idFont').text('사용가능한 아이디입니다!');
        $('#idFont').css({'color': 'mediumblue', 'font-size': 'small'});
    } else {
        $('#idFont').text('5~10자의 영문 소문자, 숫자와 특수문자(-,_)만 사용 가능합니다. ');
        $('#idFont').css({'color': 'red', 'font-size': 'small'});
    }
});
```

```
//비밀번호 확인
$('#userPwd').blur(function(){
    if(pwdJ.test($(this).val())){
        //콘솔창에 확인하기
        console.log('true');
        $('#pwdFont').text('사용가능한 비밀번호입니다. ');
        $('#pwdFont').css({'color': 'mediumblue', 'font-size': 'small'});
    } else {
        $('#pwdFont').text('8~16자 영문 소문자, 숫자, 특수문자(!@#%&* )를 허용합니다. ');
        $('#pwdFont').css({'color': 'red', 'font-size': 'small'});
    }
});

//비밀번호 일치 확인
$('#userPwdConfirm').blur(function(){
    if($('#userPwd').val() == $(this).val()){
        $('#pwdConfirmFont').text('비밀번호가 일치합니다. ');
        $('#pwdConfirmFont').css({'color': 'mediumblue', 'font-size': 'small'});
    } else {
        $('#pwdConfirmFont').text('비밀번호가 일치하지 않습니다. ');
        $('#pwdConfirmFont').css({'color': 'red', 'font-size': 'small'});
    }
});

//이름확인
$('#userName').blur(function(){
    if(nameJ.test($(this).val())){
        //콘솔창에 확인하기
        console.log(nameJ.test($(this).val()));
        $('#userName').text('');
        $('#nameFont').text('');
    } else {
        $('#nameFont').text('올바른 이름이 아닙니다. ');
        $('#nameFont').css({'color': 'red', 'font-size': 'small'});
    }
});

$('#userEmail').blur(function(){
    if(emailJ.test($(this).val())){
        //콘솔창에 확인하기
        console.log(emailJ.test($(this).val()));
        $('#userEmail').text('');
        $('#emailFont').text('');
    } else {
        $('#emailFont').text('올바른 이메일 주소가 아닙니다. ');
        $('#emailFont').css({'color': 'red', 'font-size': 'small'});
    }
});
```


코드리뷰 2

회원가입시 정규식을 이용한
유효성검사 script코드를 구현하였다.

```
//휴대폰번호유효성검사
$("#userPhone").blur(function(){
    if(phoneJ.test($(this).val())){
        //콘솔창에 확인하기
        console.log(phoneJ.test($(this).val()));
        $("#userPhone").text('');
        $('#phoneFont').text('');
    } else {
        $('#phoneFont').text('올바른 휴대폰번호가 아닙니다. ');
        $('#phoneFont').css({'color':'red','font-size':'small'});
    }
});

//교수코드 학번 유효성검사
$("#userNo").blur(function(){
    if(userNoJ.test($(this).val())){
        //콘솔창에 확인하기
        console.log(userNoJ.test($(this).val()));
        $("#userNo").text('');
        $('#userNoFont').text('');
    } else {
        $('#userNoFont').text('올바른 교수코드나 학번이 아닙니다. ');
        $('#userNoFont').css({'color':'red','font-size':'small'});
    }
});
```

```
action checkSubmit(){
    if($('#userId').val()==''){
        alert("아이디를 입력하지 않았습니다.");
        $('#userId').focus();
        return false;
    }

    if($('#userPwd').val()==''){
        alert("비밀번호를 입력하지 않았습니다.");
        $('#userPwd').focus();
        return false;
    }

    if($('#userPwdConfirm').val()==''){
        alert("비밀번호를 재확인을 입력하지 않았습니다.");
        $('#userPwdConfirm').focus();
        return false;
    }

    if($('#userName').val()==''){
        alert("이름을 입력하지 않았습니다.");
        $('#userName').focus();
        return false;
    }

    if($('#years').val()==''|| $('#month').val()==''|| $('#days').val() == ''){
        alert("생일을 선택하지 않았습니다.");
        $('#years').focus();
        return false;
    }

    if($('#gender').val()==''){
        alert("성별을 선택하지 않았습니다.");
        $('#gender').focus();
        return false;
    }

    if($('#userEmail').val()==''){
        alert("이메일을 입력하지 않았습니다.");
        $('#userEmail').focus();
        return false;
    }
}
```

```
if($('#userPhone').val()==''){
    alert("휴대폰번호를 입력하지 않았습니다.");
    $('#userPhone').focus();
    return false;
}

if($('#department').val()==''){
    alert("학과를 선택하지 않았습니다.");
    $('#department').focus();
    return false;
}

if($('#choice').val()==''){
    alert("회원 종류를 선택하지 않았습니다.");
    $('#choice').focus();
    return false;
}

//학생은 학년을 선택해야함
if($('#choice').val()=='학생' && $('#grade').val()==''){
    alert("학년을 선택하지 않았습니다.");
    $('#grade').focus();
    return false;
}

if($('#userNo').val()==''){
    if($('#choice').val()=='학생'){
        alert("학번을 입력하지 않았습니다.");
        $('#userNo').focus();
    } else{
        alert("교수코드를 입력하지 않았습니다.");
        $('#userNo').focus();
    }
    return false;
}
```


코드리뷰 3

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    int busNo = Integer.parseInt(request.getParameter("busno"));  
    String busDate1 = request.getParameter("date");  
    String busLine = request.getParameter("nosun");  
    String busSeat = request.getParameter("choice");  
    int busPrice = Integer.parseInt(request.getParameter("priceShow"));  
  
    String memberNo = request.getParameter("memberNo");  
  
    int ticketNo = (int)(Math.random()*100000 + 1);  
    Date busDate = Date.valueOf(busDate1);  
  
    Bus bus = new Bus(busNo, busDate, busLine, busSeat, busPrice);  
    BusTicket busTicket = new BusTicket(ticketNo, memberNo, busNo);  
  
    System.out.println("InsertMemberServlet : " + bus);  
    System.out.println("InsertBusTicketServlet : " + busTicket);  
  
    int result = new BusService().insertBus(bus);  
    int result2 = new BusService().insertBusTicket(busTicket);  
  
    String page = "";  
    if(result > 0 && result2 > 0) {  
        page = "WEB-INF/views/bus/ticket.jsp";  
        request.setAttribute("bus", bus);  
        response.sendRedirect(request.getContextPath());  
        request.getSession().setAttribute("msg", "회원가입에 성공하였습니다.");  
    } else {  
        page = "WEB-INF/views/common/errorPage.jsp";  
        request.setAttribute("msg", "버스 예약에 실패하였습니다.");  
    }  
}
```

버스 예약을 받기 위해
bus와 busTicket의 객체 정보를
reservationForm.jsp에서 전달받아
DB에 저장을 해주는 서블릿이다.

코드리뷰 4

JSP

```
// 과목 조회 버튼
$(document).on("click", "#div1Btn", function(){
    var subjectType = $('#seperate1 option:selected').val();
    var subjectMajor = $('#seperate2 option:selected').val();
    var subjectName = $('#subName').val();

    var memNo = $('#memberNoIn').val();

    $.ajax({
        url : 'apply.su',
        type : 'POST',
        dataType : 'json',
        data : {subjectType:subjectType,subjectMajor:subjectMajor,subjectName:subjectName},
        success : function(response){
            console.log("서버 성공");
            console.log(response);

            $('#openDiv2Table1 tbody').empty();
            var str = '<tr>';
            $.each(response, function(i){
                str += '<td>' + response[i].subjectType + '</td><td>' + response[i].subjectName +
                    '</td><td>' + response[i].subjectMajor + '</td><td>' + response[i].subjectCredit +
                    '</td><td>' + response[i].subjectTime + '</td><td>교수</td><td>' +
                    response[i].subjectLimit + '</td><td>' + response[i].subjectGrade + '학년</td><td>' +
                    response[i].subjectNo + '</td><td><button id="btnAp" class="btnAp" value=' + memNo +
                    '>수강하기</button></td>';
                str += '</tr>';
            });
            $('#openDiv2Table1 tbody').append(str);
            alert("조회되었습니다.");
        },
        error : function(data){
            alert("항목을 하나 이상 입력해주세요");
            console.log("서버 실패");
        }
    });
});
```

서블릿 – 사용자가 입력한 값에 따른 조회 조건 나누기

```
} else if(!subjectType.equals("없음") && subjectMajor.equals("없음") && subjectName.equals("")) { // 전공탐만 입력
    ArrayList<Subject> searchList = new SubjectService().selectType(subjectType);
    response.setContentType("application.json; charset=UTF-8");

    Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
    gson.toJson(searchList, response.getWriter());

} else if(subjectType.equals("없음") && !subjectMajor.equals("없음") && subjectName.equals("")) { // 학과만 입력
    ArrayList<Subject> searchList = new SubjectService().selectMajor(subjectMajor);
    response.setContentType("application.json; charset=UTF-8");

    Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
    gson.toJson(searchList, response.getWriter());
} else if(subjectType.equals("없음") && subjectMajor.equals("없음") && !subjectName.equals("")) { // 과목명만 입력
    ArrayList<Subject> searchList = new SubjectService().selectSubjectName(subjectName);
    response.setContentType("application.json; charset=UTF-8");

    Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
    gson.toJson(searchList, response.getWriter());
} else if(!subjectType.equals("없음") && !subjectMajor.equals("없음") && subjectName.equals("")) { // 전공탐, 학과
    ArrayList<Subject> searchList = new SubjectService().selectTypeMajor(subjectType,subjectMajor);
    response.setContentType("application.json; charset=UTF-8");

    Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
    gson.toJson(searchList, response.getWriter());
} else if(subjectType.equals("없음") && !subjectMajor.equals("없음") && !subjectName.equals("")) { // 학과, 과목명
    ArrayList<Subject> searchList = new SubjectService().selectMajorSub(subjectMajor,subjectName);
    response.setContentType("application.json; charset=UTF-8");

    Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
    gson.toJson(searchList, response.getWriter());
} else if(!subjectType.equals("없음") && subjectMajor.equals("없음") && !subjectName.equals("")) { // 전공탐, 과목명
    ArrayList<Subject> searchList = new SubjectService().selectTypeSub(subjectType,subjectName);
    response.setContentType("application.json; charset=UTF-8");

    Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
    gson.toJson(searchList, response.getWriter());
} else if(!subjectType.equals("없음") && !subjectMajor.equals("없음") && !subjectName.equals("")) { // 모두 입력
```

사용자(학생)가 전공탐, 학과, 과목명을 입력한 후 조회 버튼을 누르면 조건에 맞는 과목 데이터가 조회된다.

서블릿 : 이때 전공탐, 학과, 과목 명 중 사용자가 입력한 값으로 조회한다.
 예를 들면, 학과만 적으면 해당 학과에 맞는 과목이 조회되고,
 학과, 과목명을 적으면 두가지 조건을 만족시키는 과목이 조회되고,
 전공탐, 학과, 과목명 모두 적으면 세가지 조건을 모두 만족시키는 과목이 조회된다.

JSP : 사용자가 입력한 값을 서블릿으로 보낸다.
 서블릿에서 조회된 과목 리스트는 비동기 방식을 통하여 테이블에 값이 들어가고,
 사용자는 본인이 조회한 과목 리스트를 새롭게 볼 수 있다.

코드리뷰 4

z-index 속성을 이용하여, 메인사진(image/1.png)위에 div와 아이콘(왼쪽버튼.png,오른쪽버튼.png)을 적용하였다.

```
<div class="main_1_1" style = "position : relative; z-index : 3;"></div>
<div class="main_1_2" style = "position : relative; z-index : 3;"></div>
<img src = "image/왼쪽버튼.png" class="main_1_4" width = "80px" height = "80px" style = "position : relative; z-index : 3; cursor: pointer;">
<img src = "image/오른쪽버튼.png" class="main_1_5" width = "80px" height = "80px" style = "position : relative; z-index : 3; cursor: pointer;">
<img src = "image/1.png" class="main_1_0" id="main_1_0" style = "position : relative; z-index : 1;">
```

```
<script>
    var mainImage = document.getElementById('main_1_0');
    var imgArray = ["image/4.png", "image/3.png", "image/1.png"];
    var imageIndex = 0;

    function changeImage(){
        mainImage.setAttribute("src",imgArray[imageIndex]);
        imageIndex++;
        if(imageIndex >= 3){
            imageIndex = 0;
        }
    }
    setInterval(changeImage,4000);
```

메인 배너가 약 4초에 한번씩 자동으로 변경된다.
< 3개의 사진을 배열에 넣어서, 지정한 4초의 시간이 지나면 index값을 증가시켜 변경된 사진을 보여준다.
배열의 마지막 index인 2를 초과하면, 처음 index인 0이 되어, 가장 처음의 사진부터 다시 출력한다.

Image/1.png



image/2.png



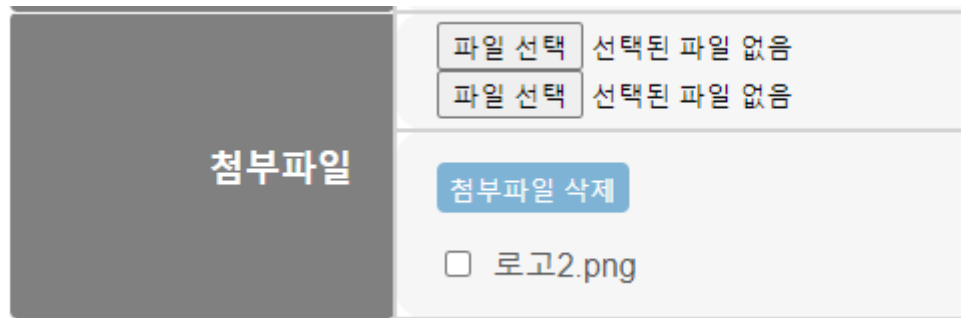
코드리뷰 5

```
<script>
// 첨부파일삭제
$("#Xbutton").click(function(){

    var checkbox = $("input[class=check_b]:checked").length;
    var fList = new Array(checkbox);
    var fileId = $("input[class=check_b]:checked").next().val();
    var changeName = $("input[class=check_b]:checked").next().next().val();
    var originName = $("input[class=check_b]:checked").next().next().next().val();
    var bNo = $("#bNo").val();

    $.ajax({
        url: "BulletinAttachmentDelete.at",
        dataType : "json",
        data: {fileId:fileId, changeName:changeName, originName:originName, bNo:bNo},
        success: function(date){
            alert("파일삭제 성공");
            $("input[class=check_b]:checked").parent().remove();
            if($(".filename").length == 0){
                $("#b_file2").html("첨부된 파일이 없습니다");
            }
        },
        error: function(data){
            alert("파일삭제에 실패 했습니다.");
        }
    });
});

</script>
```



본인이 쓴 게시물 수정할 때 등록된 첨부파일을 삭제한다면 첨부파일 삭제버튼을 눌러서 새로그침 없이 비동기 방식으로 삭제하게 만들었다. 첨부파일이 아예 없을 때는 첨부된 파일이 없습니다 라는 문구가 출력되게 하였다

코드리뷰 5

```
// 수정하기
// 체크박스 체크하고 수정하기 버튼누르면 일정 등록부분이 수정부분으로 바뀌게하기(하나에 한개밖에 못함)
$("#modifyB").click(function(){
    var checklength = $("input[class=sCheck]:checked").length;

    if(checklength > 1){
        alert("수정은 하나씩 가능합니다");
        $("input[class=sCheck]").prop("checked", false);
    } else {
        var sNo = $("input[class=sCheck]:checked").prev("input[class=sNo]").eq(0).val();

        // 수정할 학사일정 가져오기
        $.ajax({
            url: "AdminSelectOne.sc",
            dataType: "json",
            data: {sNo:sNo},
            success: function(data){
                var scheduleInsert = $("div[id=scheduleInsert]"); // 일정등록하기부분 수정하기로 바꾸기위해서 가져온다

                var divtitle2 = $("label[id=divtitle2]").html("일정 수정하기"); // 등록하기부분을 수정으로 바꿔주기
                var sModifyB = $("<input type=button id=sModifyB value=수정하기>"); // 수정하기 버튼만들어서 바꿔주기
                scheduleInsert.append(sModifyB);

                var insertT = $("input[id=insertT]").val(data[0].scheduleTitle);
                var insertD = $("input[id=insertD]").val(data[0].scheduleDate);
                var insertC = $("textarea[id=insertC]").val(data[0].scheduleContent);

                $(sModifyB).click(function(){
                    var modifyT = $(insertT).val();
                    var modifyD = $(insertD).val();
                    var modifyC = $(insertC).val();

                    $.ajax({
                        url: "AdminUpdate.sc",
                        type: "post",
                        data: {modifyT:modifyT, modifyD:modifyD, modifyC:modifyC, sNo:sNo},
                        success: function(data){
                            alert("수정되었습니다");
                            location.reload();
                        },
                        error: function(data){
                            alert("수정에 실패했습니다");
                        }
                    });
                });
            }
        });
    }
}); // 수정할 학사일정가져오기 + 수정하기 끝
```

등록하기

일정 등록하기

학사일정 제목

학사일정 날짜

학사일정 내용

➔

수정하기

일정 수정하기

학사일정 제목

학사일정 날짜

학사일정 내용

학사일정 페이지를 페이지 내에서만 모든 처리를 할 수 있게 모든 작업을 ajax를 이용했다. 특히 처음에는 등록하기 부분이었던 곳을 수정하기 버튼을 누르면 수정하기로 바뀌게 작업하였다

09

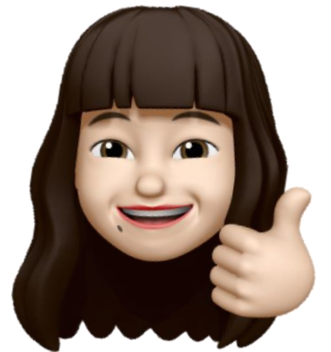
소감



명다정

프로젝트를 진행 할수록 실력이 조금씩 느는것을 느꼈고, 미니프로젝트 때와는 다르게 준비 기간도 길고 회의 할 시간도 많아서 그런지 협업이 더욱 잘 됐던 것 같다! 또한.. 준비 기간이 길더라도 여유를 가지지않고 꼼꼼히 해야함을 느꼈다!

수업시간에 따라가기 힘들었던 부분을 프로젝트 기간동안 스스로 기능을 구현해 보면서 그동안 배웠던 것을 복습해보고 머리속에 숙성시킬 수 있었다. 원격으로 하다보니 팀원들과 많이 소통을 못해서 아쉽기도 하고, 충분한 프로젝트 시간이 있었지만 처음에 설계한 기능들을 완벽하게 구현 하는것이 어려워서 미흡한 부분이 많음을 느꼈다. 기초와 개념이 중요하다는걸 느끼게 되어서 프로젝트 끝난 후에도 부족한 부분을 채우기 위해 계속 노력해야겠다. 각자 맡은 부분을 열심히 완성하느라 고생한 팀원들에게 고맙습니다.



이영실



한지원

코로나로 인해 원격수업으로 진행하여 소통하고 혼자서 코드를 작성하는데 어려움을 겪었다. 하지만 그러한 환경에도 불구하고 100% 완벽하게는 아니지만 최선을 다해 집중하여 내가 맡은 일을 수행하여 뿌듯했다



김기쁨

세미 프로젝트 동안 우리 팀원들과
함께 이야기하고, 프로그램을
구현하면서 너무 즐거웠다.
물론 힘들었던 부분도 많았지만,
이로 인해 더욱 성장할 수 있었던 것 같다.
그리고 프로젝트를 진행하면서 소통의
중요함을 확실히 느꼈다.
소통을 더욱 잘 할 수 있도록
노력해야겠다는 생각이 들었다.
함께 고생해 준 우리 팀원들 너무 감사하고,
모두 성장한 모습을 바탕으로 각자의
자리에서 꼭 승리했으면 좋겠다.



최하윤

세미프로젝트 하기전에는 막연한
두려움부터 앞섰다. 내가 과연
잘 할 수 있을까부터 시작해서
완성을 못하면 어찌지까지 너무 걱정을
많이 했지만 짐이 되지 않기 위해서 최선을
다해 하다보니 어느샌가 완성이 되어있어서
뿌듯하다.
우리 팀원분들 다들 고생하셨습니다!

**THANK
YOU**

여 성 시 대