



2017. 2. 2 ~ 2017. 7. 6 | KOSTA 151th

모델링 기반 Front-End & Back-End 개발자 양성과정

Classic

강의 시간 강사와 수강생 사이의 원활한 소통과
자료 업로드를 공유할 수 있는 강의실단위 카페/밴드형
웹 사이트



Classic

[강의 관리 서비스]



담당강사 : 임경균

이현정, 최은혜, 최석환, 위진학, 김은영, 노지영, 박소현

목차

PART 1. 프로젝트 주제소개	5
프로젝트주제	
주제 목적	
프로젝트 선정 이유	
PART 2. 프로젝트 일정	7
PART 3. 마일스톤	9
PART 4. 프로젝트개발환경	13
시스템구성	
아키텍처 개요	
개발환경	
PART 5. 프로젝트 요구사항	16
스토리보드	
PART 6. UML 다이어그램	40
USECASE	
Class Diagram	
마일스톤	

PART 7. DB Modeling	42
DB Modeling	
테이블 정의서	
PART 8. Project Setting	52
Spring MVC 프로젝트의 설정	
POM.xml	
root-context.xml	
servlet-context.xml	
security-context.xml	
PART 9. 시스템구현	79
PART 10. 트러블 슈팅 보완사항	105
PART 11. 프로젝트 후기	108
PART 12. 참조문헌	111

PART 1

프로젝트 주제 소개

프로젝트 주제를 소개하고 주제를 선정하게 된 이유와

프로젝트 목표를 제시합니다.

PART 1. 프로젝트 주제소개

■ 프로젝트 주제

-프로젝트 제목: Classic (class Intelligent Commander)

-프로젝트 주제:

강의를 하고 수업을 듣는 교육기관 내에서 강의를 가르치는 강사와 강의를 수강하는 학생들의 편의성을 극대화 시켜 학생과 선생 간의 원활한 의사소통을 도모하고 자료의 효율적 관리를 도울 수 있도록 기능을 갖추었습니다.

■ 주제 목적

1). 수업 내에서 필요한 자료를 나누어 강사와 수강생 간의 자료의 나눔을 편리하게!

현재 존재하는 수업시간 선생님과 학생들 간의 자료 공유는 포털 사이트 내의 카페/밴드에 초점이 맞춰져 있습니다. 저희는 대상 고객을 수업이 이루어질 수 있는 공간, 자료의 나눔이 필요한 공간에서 사용하는 서비스를 제공하고자 합니다.

2). 강의실 내에서 이루어지는 의사소통 을 간편하게!

클래스 내에 필요한 출결 사항, 건의사항을 바로 확인할 수 있도록 예전부터 중요시되던 수업시간 내에 규칙들을 클래식 하게, 강의장 내에 답답한,좁좁한 환경을 쉽게 파악하고 조절할 수 있게 도와주는 서비스를 제공하고자 합니다.

■ 프로젝트 선정 이유

국가에서 지원하는 취업지원 과정이 전국에서 많이 늘어나고 있어 수강하는 학생들과 강사님들도 많이 늘어나고 있는 가운데 강사님께서 자료를 관리하시는 부분이나 학생들이 수업을 듣고 자료를 정리해 두는 부분에 있어서 갖춰진 툴이 부족해 학생들은 수업을 듣고 추후에 자료정리에 애를 먹거나 강사님 같은 경우에는 매번 강의를 하실 때 마다 같은 내용을 찾으셔야 하는 불편함과, 과거에 수업을 하실 때 사용하셨던 자료들을 관리하는 것에 대한 어려움을 가지고 강의를 하시게 됩니다.

저희는 평소에 위와 같은 어려움에 대한 개선방향을 제시하고 직접 느끼고 경험한 것을 바탕으로 맞춤화 된 서비스 툴을 개발 하고자 노력하였습니다. 학생들이나 강사님들 에게 평소 들었던 개선사항을 반영하였고 직접 강사님들께서 필요로 하시는 기능을 조사하여 기능을 기획하였습니다.

강사님과 학생 간의 원활한 의사소통을 위해서 여러 가지 API를 참조하여 사용하였으며 Spring에서 제공하는 고급기능들을 최대한 활용하여 프로젝트의 목적을 달성하고자 노력 하였습니다.

학생들의 입장에서 필요로 하거나 개선되었으면 하는 점들을 조사, 파악하여 학생들이 보다 편리한 강의를 수강할 수 있도록 하였습니다. 그 중에서 국비지원 교육에서 중요하게 다뤄지는 출결 사항의 개선을 위하여 학생들의 출결 관리를 가시적으로 확인할 수 있는 기능을 넣었고 안드로이드를 접목하며 자신의 출결 관리를 보다 더 정확하고 확실하게 할 수 있도록 노력하였습니다.

PART 2

프로젝트 일정

프로젝트 일정의 세부적인 내용을 보여줍니다.

PART 2. 프로젝트 일정

■ 프로젝트 일정

1주 현성웨이		6/1	6/2	6/3	6/4	6/5	6/6	6/7	6/8	6/9	6/10	6/11	6/12	6/13	6/14	6/15	6/16	6/17	6/18	6/19	6/20	6/21	6/22	6/23	6/24	6/25	6/26	6/27	6/28	6/29	6/30	7/1	7/2	
요구사항정의	린캔버스																																	
	요구사항 정의서																																	
	아키텍처 개요																																	
	일정관리 (간트차트)																																	
	업무분담																																	
	프로젝트 제안서																																	
설계/분석	데이터 정의서																																	
	적용기술서 (코드 정의서)																																	
	UI설계 (스토리보드)																																	
	ERD																																	
	Class Diagram																																	
	Sequence Diagram																																	
	UseCase Diagram																																	
구현	주요기능명세서																																	
	테스트 (요구사항 정의서 체크)																																	
이행	발표 PPT																																	
	시연 동영상																																	
스터디	IoT , 안드로이드 스터디																																	
	소켓 , 스케줄링 스터디																																	
프론트 엔드	UI 타일즈 작업																																	
	UI 구현																																	
백엔드(안드로이드)	로그인 UI 구현																																	
	로그인 DB 연동 + 구현																																	
	큐알코드 생성																																	
	큐알코드 스캔 및 입력기능.																																	
백엔드	로그인 구현																																	
	회원가입 구현																																	
	메인 페이지																																	
	소켓 쪽지 알람																																	
	스케줄링																																	
	캘린더																																	
	히스토리																																	
	과제 게시판																																	
	수업 게시판																																	
	통합 관리 게시판																																	
	오늘의 수업																																	
	링크 게시판																																	
	오류 검사																																	

PART 3

마일스톤

PART 3. 마일스톤

■ 마일스톤 - 정의

마일스톤 명	기간	마일스톤 정의	마일스톤 목표 이미지	비고
요구사항 정의	2D	<ul style="list-style-type: none"> 프로젝트를 수행하기 위한 요구사항을 정의 하는 단계. 	<ul style="list-style-type: none"> 본 프로젝트의 성공적 개발을 위해 중요 이해관계자를 식별하고 그들로부터 본 시스템에 가장 적합한 요구사항을 수집하여 프로젝트 성공의 기반을 마련한다. 	<ul style="list-style-type: none"> 6월 1일 ~ 6월2일
분석/ 설계	10D	<ul style="list-style-type: none"> 수집된 요구사항을 분석하고 정리 하는 단계 문서작업을 기반한 아키텍처 선정 단계 	<ul style="list-style-type: none"> 이해관계자들로부터 수집된 요구사항의 범위 및 제약사항을 설정하여 구체적으로 적용하기 위한 문서화 한다. 기술 위험과 적용 가능한 조직의 제약사항을 해결하고 팀 역할 분배 및 목적에 부합하는 아키텍처 확립한다. 	<ul style="list-style-type: none"> 6월 3일 ~ 6월 12일
개발	20D	<ul style="list-style-type: none"> 확립된 아키텍처 기반으로 개발 진행 단계 	<ul style="list-style-type: none"> 맡은 역할의 목표를 효과적으로 달성한다. 	<ul style="list-style-type: none"> 6월 13일 ~ 7월 2일
이행	1D	<ul style="list-style-type: none"> 이해관계자들이 소프트웨어 시스템의 결과를 수용하는 단계 	<ul style="list-style-type: none"> 이해관계자들 목적에 부합한다는 것을 수용하고 해당 시스템을 제공한다. 	<ul style="list-style-type: none"> 7월 3일

■ 마일스톤 - 단계별 알파 상태 카드 배열

마일스톤 명	이해관계자	기회	요구사항	S/W	팀	작업방식	작업
요구사항 정의	식별	기회식별	개념정의	아키텍처 선정	팀요건 정의	원칙수립	과업확정
	대표선출	솔루션 필요성 확인	요건정의		팀구성	작업방식 확정	사전준비
	작업참여	솔루션 가치 확인			작업수행	시범적용	
분석/설계	작업참여	솔루션 타당성 검증	범위정의	아키텍처 검증	팀빌딩	전체적용	작업시작
	배포기준 합의						작업진행
개발(구현)	작업참여	솔루션 개발	요건합의	사용가능	작업수행	전체적용	작업시작
			요건구현	배포준비		작업방식 내재화	작업진행
							목표달성
이행	배포합의		니즈충족	운영	팀해산	사용종료	작업종료

PART 3. 마일스톤

■ 마일스톤 - 단계별 수행계획 수립

마일스톤(단계)	알파	처음 알파상태	목표 알파상태	주요 활동 및 산출물
Setup 요구정의	이해관계자		식별, 대표선출, 작업참여	이해관계자를 식별하고 가상으로 이해관계자 역할을 분담, 대표를 선출하여 비정기적 회의를 진행한다.
	기회		기회식별, 솔루션필요성확인, 솔루션가치확인	아이템에 대한 공모, 솔루션의 필요성 및 가치와 프로젝트 목적에 대해 지속적으로 토론, 린캔버스 제출
	요구사항		개념정의	요구사항정의서
	소프트웨어 시스템		아키텍처 선정	프로젝트에 적용될 언어 및 기술을 정리, DB, WAS 등의 버전을 확정지어 기획서에 반영한다. 아키텍처 개요
	팀		팀요건정의, 팀구성, 팀빌딩	업무분담
	작업방식		원칙수립, 작업방식확정, 시범적용	칸반보드
	작업		과업확정, 사전준비	칸트차트(프로젝트 일정표)
분석 / 설계	이해관계자		작업참여 배포기준 합의	가상의 이해관계자 대표들이 역할에 의거해 비정기적인 회의를 통해 피드백을 제공한다
	기회		솔루션개발	요구분석에 기반해 솔루션의 개발 필요성에 대한 팀원 전체의 명확한 이해를 위한 토론을 한다
	요구사항		범위정의 요건합의	Use Case Diagram, Sequence Diagram, Class Diagram
	소프트웨어 시스템		아키텍처 검증	적용기술서 (코드 정의서), UI설계 (스토리보드), ERD(논리), 테이블 정의서 작성
	팀		작업수행	일별 일정관리(칸반보드) 및 시작회의와 마무리회의를 통해 변화 적응성 및 효과적인 업무를 위해 토론한다
	작업방식		전체적용	GIT을 이용하여, 작업방식의 원활한 피드백을 통해 팀 소통과 협업을 지원할수 있도록 한다
	작업		작업시작 작업진행	매일 시작회의, 마무리회의를 통해 작업진행을 상호체크하고 KPT를 통해 위험요소를 통제한다

마일스톤(단계)	알파	처음 알파상태 * 주의: 알파별로 작성	목표 알파상태	주요 활동 및 산출물
구현(개발)	이해관계자		배포합의	가상의 이해관계자 대표들이 배포기준에 의거 배포를 합의한다
	기회		솔루션개발	기회를 명확히 해결하는 시스템을 구현한다
	요구사항		요건합의 요건구현	요구사항에 대한 구현 체크를 통한 구현 피드백
	소프트웨어 시스템		사용가능 배포준비	ERD(물리), Class Diagram, 주요기능 명세서
	팀		작업수행	일별 일정관리(칸반보드) 및 시작회의와 마무리회의를 통해 변화 적응성 및 효과적인 업무를 위해 토론한다
	작업방식		전체적용 작업방식내재화	GIT을 이용하여, 작업방식의 원활한 피드백을 통해 팀 소통과 협업을 지원할수 있도록 한다
	작업		작업시작 작업진행 목표달성	매일 시작회의,마무리회의를 통해 작업진행을 상호체크 하고 KPT를 통해 위험요소를 통제한다
이행	이해관계자		배포합의	
	기회			
	요구사항		니즈충족	
	소프트웨어 시스템			
	팀		팀해산	
	작업방식		사용종료	
	작업		작업종료	발표 PPT, 시연 동영상

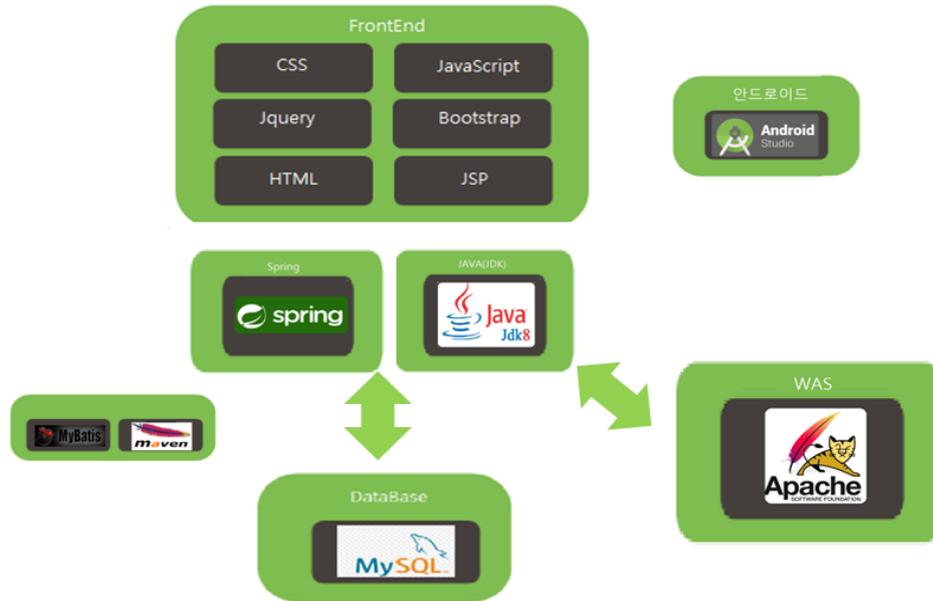
PART 4

프로젝트 개발환경

프로젝트시 사용했던 개발 틀을 보여줍니다.

PART 4. 프로젝트 개발 환경

■ 아키텍처 개요



■ 개발 환경

PROJECT : ADVENTURE TEAM ADIUTOR RISK LOG						
#	작업 구분	작업 상세	위험도	조치	상태	
	◆STS툴과 정	내외설				
		1. STS 버전 확인	높음			
		spring-tool-suite-3.8.4.RELEASE-e4.6.3-win32	중간			
		2. JDK 버전 확인	낮음			
		- jdk1.6	높음			
		jdk-8u121-windows-x64	중간			
		3. 프로젝트 시작 전 workspace 경로 확인	낮음			
		- C:\WKosta151\team1_project\Wlabs	높음			
		4. Open Perspective - Spring 으로 설정	중간			
		5. 톰캣 설치 및 Port번호 변경	낮음			
		- 톰캣 설치(apache-tomcat-7.0.75)	높음			
		- Apache Tomcat v7.0	높음			
		- 설치 후, sever.xml 문서의 65번 줄, 한글 처리(URIEncoding= "UTF-8") 미 port번호 8090으로 변경	높음			
		<Connector URIEncoding= "UTF-8" connectionTimeout="20000" port= "8060" protocol= "HTTP/1.1" redirectPort= "8443" />	높음			
		6. Preferences의 Web > CSS, HTML, JSP Files 의 Encoding ISO 10646/Unicode(UTF-8)로 변경.	높음			
		7. 새 프로젝트 생성	높음			
		- 프로젝트 명 : Class-ic	높음			
		- 프로젝트 생성시 web.xml 문서 체크	높음			
		8. 프로젝트 생성 후	높음			
		- Properties > Resource > Text file encoding 'UTF-8'로 변경	높음			
		9. 프로젝트 진행 시 필요한 jar파일 붙이기	높음			
		- WebContent > WEB-INF > lib 폴더	높음			
		- DB연동 : MySQL Installer 5.7.18	높음			

	· JSTL 사용 : javax.servlet.jsp.jstl-1.2.1.jar / javax.servlet.jsp.jstl-api-1.2.1.jar	높음
	· 파일업로드 : imgscalr-lib 4.2 commons-fileupload 1.3.2	높음
	10. DBCP를 위한 context.xml 문서 생성	높음
	- META-INF > context.xml	높음
	- context.xml 문서의 DB 계정 username =: "handSomeWare" password="1004"	높음
	11. 프로젝트 관련 정보를 담은 문서 파일(DB설정, sql, 프로젝트 설명 문서 등)은 META-INF 폴더에 저장.	높음
◆Database 설정	1. MySQL Tool 버전 확인	높음
	- 미정	높음
	2. Oracle 계정 생성 및 접속 확인	높음
	- 계정 생성 : Team1	높음
	- 관리자 > 다른 사용자 > 사용자 생성	높음
	> 사용자 이름 : handSomeWare, 비밀번호 : 1004	높음
	> 기본 테이블 스페이스 USERS, 임시 테이블 스페이스 TEMP	높음
	> 부여된 롤 : CONNECT, RESOURCE 항목 권한이 부여됨 체크	높음
	> 할당량 : USERS 무제한 체크	높음
	- 접속 확인	높음
	새 접속 > 접속 이름 : Team1, 사용자 이름 : handSomeWare, 비밀번호 : 1004	높음
	> 테스트 > 상태 : 성공 > 확인 후 접속	높음
	3. 프로젝트에 필요한 Table 정보 확인	높음
	- 설계한 Table의 제약 정보 확인	높음
	*DB실행이 안될 시,	높음
	제어판 > 관리도구 > 서비스 > mysql_Service의 상태가 시작됨인지 확인	높음
◆형상관리 설정 (STS 내부 툴 사용)	Workspace 확인 : C:\Kosta151\team1_project\Labs	높음
	작업 중 파일 변경 시, commit 시간은 PM13시00분, PM8시.	높음
	COMMIT시, 변경 사항에 대한 history 의무 작성	높음

PART 5

프로젝트 요구사항

프로젝트에서 필요로 하는 사항들을 정리하여 보여줍니다.

PART 5. 프로젝트 요구사항

■ 프로젝트 요구사항

1. 효율적 게시 글 관리 서비스

- 다양한 수업(기수)이 개설 될 때마다 중복된 게시 글을 수동적이고 반복적으로 작성해야 하는 불편함을 해소할 수 있는 게시판 통합 시스템을 제공한다.
- 같은 글을 다른 기수(수업)에서 사용할 때 낭비되는 데이터의 비효율을 해소한다.
- 게시 글을 작성할 때 링크를 첨부하는 경우가 있는데 이 때, 링크만 따로 찾고자 할 때 번거로움이 발생한다. 따라서 링크만 따로 관리, 검색 할 수 있는 게시판 서비스를 제공한다.
- 파일 업로드, 다운로드 기능을 제공한다.
- 학생이 수 많은 글 가운데 오늘 배운 내용을 확인하고 싶을 때 게시 글을 일일이 찾아 들어가야 하는 불편함을 해소하기 위해, 강사가 오늘의 수업에 쓰인 게시 글을 저장하면, 학생이 캘린더에서 확인할 수 있는 서비스를 제공한다.
- 강사는 공지를 등록할 수 있다.
- 학생은 게시 글이 마음에 들 경우 북마크에 추가해 헤더에서 빠르게 접근할 수 있다.

2. 효율적인 일정 관리

- 캘린더에서 커리큘럼 및 일정을 등록하면 가시적으로 시간 순으로 출력되어 일정을 한눈에 볼 수 있게 한다.
- 학생에게 TO-DO-LIST를 작성할 수 있고 한 내용은 수정, 삭제, 빗금 긋기 등을 할 수 있는 서비스를 제공한다
- 강사에게 포스트잇 붙이는 것 같은 메모 서비스를 제공하여, 강사가 수업 시 필요하거나 기억해야 할 내용을 등록, 수정, 삭제할 수 있다.

3. 편리한 학생 관리 서비스

- 토털사이트 카페 이용 시 학생 가입 인증을 위해 필요한 과정의 불편함을 QR 찍기를 통해 해소한다.
- 강사는 학생의 출결 사항과 출석률을 한눈에 확인할 수 있으며, 학생 별 출결 사항을 엑셀로 다운이 가능하다.
- 학생은 사이트에서 QR로 입실 퇴실을 찍을 수 있고, 자신의 출결사항을 확인할 수 있다.
- 강사가 학생의 프로필 사진 및 간단한 정보를 출석부에서 확인할 수 있게 하여 수업 초기 학생 파악에 용이하게 한다
- 과제 게시판을 조별로 관리할 때, 조 편성을 하고, 그에 해당하는 조별 게시판이 자동 추가되는 서비스를 제공한다.

4. 커뮤니티

- 다른 사람과 쪽지를 주고 받고 실시간으로 쪽지 수신 알림을 받을 수 있다.

5. IOT 서비스

- 교실의 쾌적한 환경 유지를 위해, 이산화탄소농도 측정을 하고, 이에 따라 공기청정기를 제어할 수 있는 서비스를 제공한다.

PART 5. 프로젝트 요구사항

■ 스토리 보드

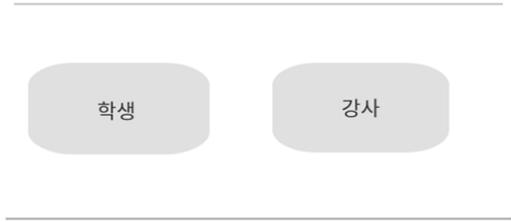
Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 - 로그인하지 않은 상태	Tool	오븐	Team	KOSTA 151 기 1조		



①로그인 하지 않았을 시

- 로그인 화면을 출력
- 로그인 성공 시 기수를 선택할 수 있는 화면 출력
- 회원가입 클릭 시 회원가입 입력 페이지로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 - 회원가입	Tool	오븐	Team	KOSTA 151 기 1조		



①회원가입 클릭 시

회원가입을 할 때는 학생과 강사로 나뉘어 가입을 할 수 있음

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 - 회원가입 페이지	Tool	오븐	Team	KOSTA 151 기 1조		

① 회원정보입력

- 아이디 는 중복된 아이디가 있을 시 체크함
- 비밀번호는 숫자 + 문자 + 특수문자로 구성
- 휴대폰번호는 숫자만 입력가능
- 이름은 문자만 입력가능

②취소버튼

- 취소 버튼 클릭 시 회원정보 입력 칸의 모든 내용이 리셋 됨

③완료버튼

- 완료 버튼 클릭 시 회원가입이 완료되고 로그인 페이지가 출력

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 (학생) - QR 코드	Tool	오븐	Team	KOSTA 151 기 1조		

① QR 버튼

- QR 버튼 클릭 시 QR 코드 출력

②QR 코드

- 닫기 버튼 클릭 시 QR코드 창 닫힘

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 (학생)-북마크	Tool	오븐	Team	KOSTA 151 기 1조		

① 북마크 버튼
북마크 버튼을 클릭 시 북마크 했던 게시글들이 출력

② 북마크 버튼 클릭
• 수업보드에서 북마크 했던 게시글들이 출력

• 게시글 클릭 시 게시글 상세보기로 이동

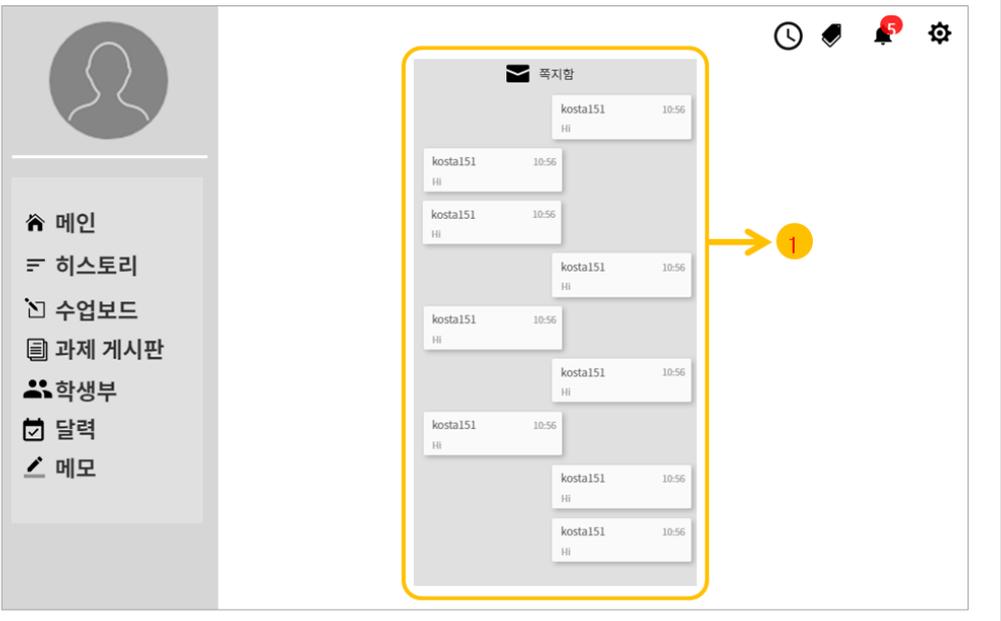
Project	Class_ic	Date	2017.06.05	Version	1.0	Page	2
Title	Headr 영역 (학생)-알림기능	Tool	오븐	Team	KOSTA 151 기 1조		

① 알림 버튼
쪽지가 오면 숫자로 알림을 나타냄

② 알림 버튼 클릭 시
강사에게만 쪽지를 보낼 수 있음

③ more 클릭 시
쪽지함으로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	3
Title	Headr 영역 (학생)-알림기능-쪽지함	Tool	오븐	Team	KOSTA 151 기 1조		



① 쪽지함
쪽지의 내용을 보여줌

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	4
Title	Headr 영역 -설정	Tool	오븐	Team	KOSTA 151 기 1조		



① 설정 버튼
설정 버튼 클릭 시
회원정보 수정, 로그아웃 버튼 출력

② 설정 버튼

- 내 프로필 사진을 출력
- 회원정보수정 버튼 클릭 시
회원정보 수정 페이지로 이동
- 로그아웃 버튼 클릭 시
로그인 페이지로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	4
Title	Headr 영역 -설정 - 메모	Tool	오븐	Team	KOSTA 151 기 1조		

① 메모 버튼
메모버튼 클릭 시
메모를 할 수 있는 모달 출력

② 메모 쓰기
메모를 할 수 있는 창을 팝오버
기능으로 나타내고
메모를 쓰기를 완료하고
메모완료 버튼 클릭 시
메모페이지로 내용을 저장함

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	4
Title	Headr 영역 -설정 - 링크	Tool	오븐	Team	KOSTA 151 기 1조		

① 링크 버튼
링크버튼 클릭 시
링크

② 링크 검색
검색을 하면
검색한 내용에 알맞은
글 또는 링크를 리스트로 출력
함
리스트를 클릭 시 해당
상세보기 페이지로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	3
Title	메인 페이지	Tool	오븐	Team	KOSTA 151 기 1조		

① 현재 상태

- 아두이노를 통하여 현재 이산화탄소 농도를 나타냄
- 남은 수업 일수를 나타냄
- 오늘의 출석한 수강생의 수를 나타냄

② 수업 보드

내가 듣고 있는 수업 카테고리가 나타내짐

③ To do list

오늘의 할 일을 적을 수 있고 할 일을 완료했을 시 체크박스를 누르면 완료 표시가 됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	3
Title	메인 페이지	Tool	오븐	Team	KOSTA 151 기 1조		

① 히스토리 확인

교사가 입력한 히스토리를 확인할 수 있음

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 🏠 메인
- ≡ 히스토리
- 📅 수업보드
- 📁 과제 게시판
- 👥 학생부
- 📄 달력
- 📝 메모
- 출결 관리
- 교실 현황

1차

더보기

2차

더보기

3차

1

2

더보기

① 카테고리 폴더

- 메인 카테고리

② '더보기' 클릭 시

- 상세 게시판으로 페이지 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 🏠 메인
- ≡ 히스토리
- 📅 수업보드
- 📁 과제 게시판
- 👥 학생부
- 📄 달력
- 📝 메모
- 출결 관리
- 교실 현황

< Back

3차

1

1조

2조

3조

+

글번호	내용	일자
5	제어문(For문)	17/06/05
4	자바 학습 외국 사이트	17/06/05
3	java 메모리 구조	17/06/05
2	이클립스 세팅	17/06/05
1	자바 기본 환경 세팅 하기	17/05/23

◀
◁
1
2
3
▷
▶

글쓰기

① 상세 게시판

- 각 조마다 탭이 있음

- '+' 버튼 클릭 시 새로운 게시판 생성

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 출결 관리
- 교실 현황

< Back
🔍 🔔 ⚙️

3차

1
1조
2조
3조
+

글번호	내용	일자
5	제어문(For문)	17/06/05
4	자바 학습 외국 사이트	17/06/05
3	java 메모리 구조	17/06/05
2	이클립스 세팅	17/06/05
1	자바 기본 환경 세팅 하기	17/05/23

<<
<
1
>
>>

2 글쓰기

① 상세 게시판

- 각 조마다 탭이있음
- '+' 버튼 클릭 시 새로운 게시판 생성

② '글쓰기' 버튼 클릭

- 글쓰기 폼으로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 출결 관리
- 교실 현황

< Back
🔍 🔔 ⚙️

제목

내용

파일 파일 추가

+

취소
등록

글쓰기

① 글쓰기 폼

- 제목,내용,파일 업로드 가능

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (학생부)	Tool	오븐	Team	KOSTA 151 기 1주		

이름

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 출결 관리
- 교실 현황

① 학생 리스트

- 학생들이 등록된 이미지가 리스트로 출력
- 등록된 이미지가 없으면 기본 이미지로 대체됨.

② 학생 로그인 시

- 로그인 된 상태이면 이미지에 불이 들어오고 아니면 회색처리 됨.

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역(달력)	Tool	오븐	Team	KOSTA 151 기 1주		

이름

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 출결 관리
- 교실 현황

① 일정

- 일정 기간 표시 됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역(달력-일정등록)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 🏠 메인
- ≡ 히스토리
- 📅 수업보드
- 📄 과제 게시판
- 👤 학생부
- 📅 달력
- 📝 메모
- 출결 관리
- 교실 현황

글번호	일자	내용
1	17/06/05 10:30AM	Spring

확인

① 일정

- 일정 내용 상세 출력
- 내용 링크를 클릭하면 해당 상세 페이지로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역(메모)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 🏠 메인
- ≡ 히스토리
- 📅 수업보드
- 📄 과제 게시판
- 👤 학생부
- 📅 달력
- 📝 메모
- 출결 관리
- 교실 현황

글번호	내용	일자
1	과제 하기!!	17/05/23
2	Spring 시작~	17/06/05

① 메모 전체 게시판

- 메모 전체 내용이 출력

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역(출결 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 🏠 메인
- ≡ 히스토리
- 📅 수업보드
- 📄 과제 게시판
- 👥 학생부
- 📄 달력
- 📝 메모
- 출결 관리
- 📍 교실 현황

출결일시
전송일시
구분

2017.06.02 18:57	2017.06.02 18:57	[퇴실]
2017.06.02 08:53	2017.06.02 08:53	[입실]

① 출결 게시판

-기한 내에 등록된 날짜에 맞는 출석현황 출력

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역(교실 상태게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

이름

- 🏠 메인
- ≡ 히스토리
- 📅 수업보드
- 📄 과제 게시판
- 👥 학생부
- 📄 달력
- 📝 메모
- 출결 관리
- 교실 현황

Temperature
Light
Air Conditioner ON/OFF

22

°C

temperature_0
HOME_STATION

오프 전

447

lux

light_0
HOME_STATION

오프 전

powerSwitch_0
HOME_STATION

오프 전

기기

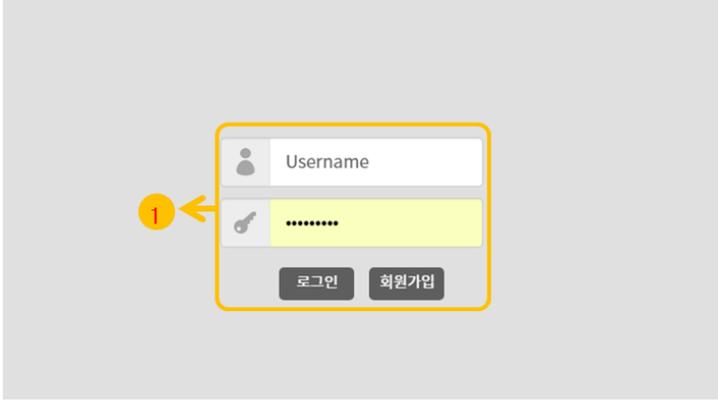
설정

Graph

① 교실 상태

-교실 상태에 관한 데이터를 그래프로 출력
(예: 온도, CO2 농도 등)

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 - 로그인하지 않은 상태	Tool	오븐	Team	KOSTA 151 기 1조		



①로그인 하지 않았을 시

- 로그인 화면을 출력
- 로그인 성공 시 기수를 선택할 수 있는 화면 출력
- 회원가입 클릭 시 회원가입 입력 페이지로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 - 회원가입	Tool	오븐	Team	KOSTA 151 기 1조		



①회원가입 클릭 시

회원가입을 할 때는 학생과 강사로 나뉘어 가입을 할 수 있음

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역 - 회원가입 페이지	Tool	오븐	Team	KOSTA 151 기 1조		

① 회원정보입력

- 아이디 는 중복된 아이디가 있을 시 체크함
- 비밀번호는 숫자 + 문자 + 특수문자로 구성
- 휴대폰번호는 숫자만 입력가능
- 이름은 문자만 입력가능

②취소버튼

- 취소 버튼 클릭 시 회원정보 입력 칸의 모든 내용이 리셋 됨

③완료버튼

- 완료 버튼 클릭 시 회원가입이 완료되고 로그인 페이지가 출력

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	로그인 후 첫 페이지 영역	Tool	오븐	Team	KOSTA 151 기 1조		

① 헤더

로그인 시 헤더가 생성되고 모든 페이지에 고정되어 따라다녀 접근이 용이함

- 🏠 통합 관리 시스템

통합 관리 시스템 에서 통합적인 글 등록과 이동 등 관리가 가능

②기수 선택/추가

자신이 가입된 기수를 보여줌.
(학생 ver : 초대 수락이 완료되면 자동 추가 생성)
(교사 ver: 교사는 클래스를 추가할 수 있음)
클릭하면 해당 클래스로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Headr 영역	Tool	오븐	Team	KOSTA 151 기 1조		

📄 📱 🔔 ⚙️

강의 등록

강의명 (기수):

강의 기간:

강의 기간:

수업 시작 시간:

수업 종료 시간:

취소
완료

① 강의 등록
강의 기간을 선택하면 datepicker로 지정할 수 있다

② 취소버튼
• 취소 버튼 클릭 시 메인 페이지로 이동

③ 완료버튼
• 완료 버튼을 클릭 시 성공이면 성공메시지 출력 및 첫 페이지로 이동
실패면 입력메시지 출력

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (교사 메인)	Tool	오븐	Team	KOSTA 151 기 1조		

임경균 강사

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 교실상태

151 기 학생수 : 20 명
교육 기간: 2017.2.2~2017.7.6

현재 이산화탄소 농도 1000ppm 남은 수업 일수 D-48 진도 62%

📄 📱 🔔 ⚙️

MEMO...

① 클래스명, 학생수, D-day 표시
남은 일 수와 진도를 가시적으로 보여줌
아두이노를 통하여 현재 이산화탄소 농도를 나타냄

② 캘린더 기능
• 이번 달의 일정을 캘린더로 보여주고 클릭하면 세부 내용을 확인할 수 있음

③ 메모 기능
• 기억해야 할 내용을 메모 할 수 있음

31 / 126

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (히스토리)	Tool	오븐	Team	KOSTA 151 기 1조		

① **히스토리 출력**
휴일, 커리큘럼 등 주요 일정을 히스토리 형식으로 보여줌

② **히스토리 추가**
클릭 시 히스토리 추가 모달창 이 뜸

③ **히스토리 수정**
수정 모달창이 뜸

④ **히스토리 삭제**
클릭 시 삭제 됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (히스토리)	Tool	오븐	Team	KOSTA 151 기 1조		

히스토리 추가 클릭 시

① **히스토리 등록/수정**
datepicker 로 기간 입력/수정 가능

② **확인**
히스토리에 날짜 순으로 추가 됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (수업보드)	Tool	오븐	Team	KOSTA 151 기 1조		

The screenshot shows a dashboard with a left sidebar menu. The '수업보드' (Lesson Board) menu item is highlighted. The main area contains three content cards: 'JAVA', 'DB', and 'WEB'. A '+ button is also visible. Numbered callouts point to the 'DB' card (1), the bottom right area (2), and the '+' button (3).

① 수업 보드
수업 관련 게시글을 강사가 올릴 수 있는 게시판
최근 글 5개를 출력 ,접근가능

② 더보기
해당 게시판으로 이동

③카테고리 추가
클릭 시 카테고리 추가 모달창 이 뜬

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (수업보드)	Tool	오븐	Team	KOSTA 151 기 1조		

The screenshot shows a modal window for adding a category. It has a text input field labeled '대 분류 카테고리 명' (Main category name) and buttons for '취소' (Cancel) and '추가' (Add). Below it, there are three tabs labeled '1차', '2차', and '3차', with the '3차' tab selected. A second modal window is shown below the tabs, with a text input field labeled '소 분류 카테고리 명' (Sub-category name) and '취소' and '추가' buttons. Numbered callouts point to the main modal (1) and the sub-modal (2).

① 카테고리 추가
-+ 버튼 클릭 시 추가 폼이 뜬

②카테고리 추가
-추가 버튼 클릭 시 탭이 추가 됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (수업보드)	Tool	오븐	Team	KOSTA 151 기 1조		

① 상세 게시판
 '+' 버튼 클릭 시 새로운 게시판 생성

② '글쓰기' 버튼 클릭
 - 글쓰기 폼으로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (수업보드)	Tool	오븐	Team	KOSTA 151 기 1조		

① 글쓰기 폼
 - 제목, 내용, 파일 업로드 가능

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

① 상세 게시판
 - 각 조마다 탭이 있음
 - '+' 버튼 클릭 시 새로운 게시판 생성 해주는 모달 창 옴

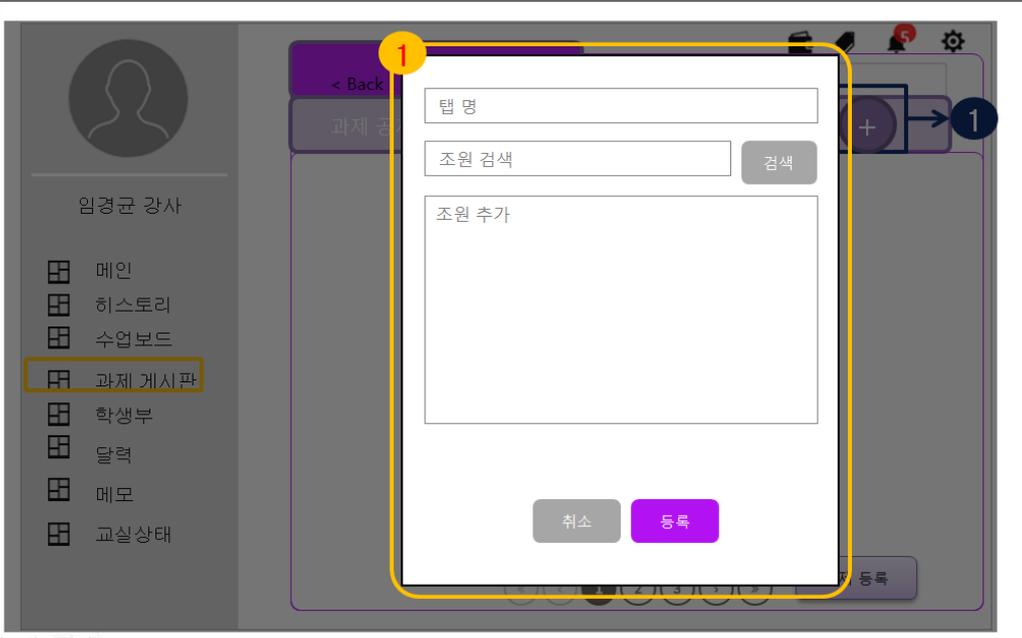
② 알림 기능
 - 새 과제 공지는 알림으로 위
 예 옴, 클릭 시 상세페이지로 이동

③ 과제 등록
 - 과제 등록 버튼 클릭 시 글쓰기 창이 뜬다

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		

① 과제 등록 폼
 - 제목, 내용, 파일, 링크 업로드 가능

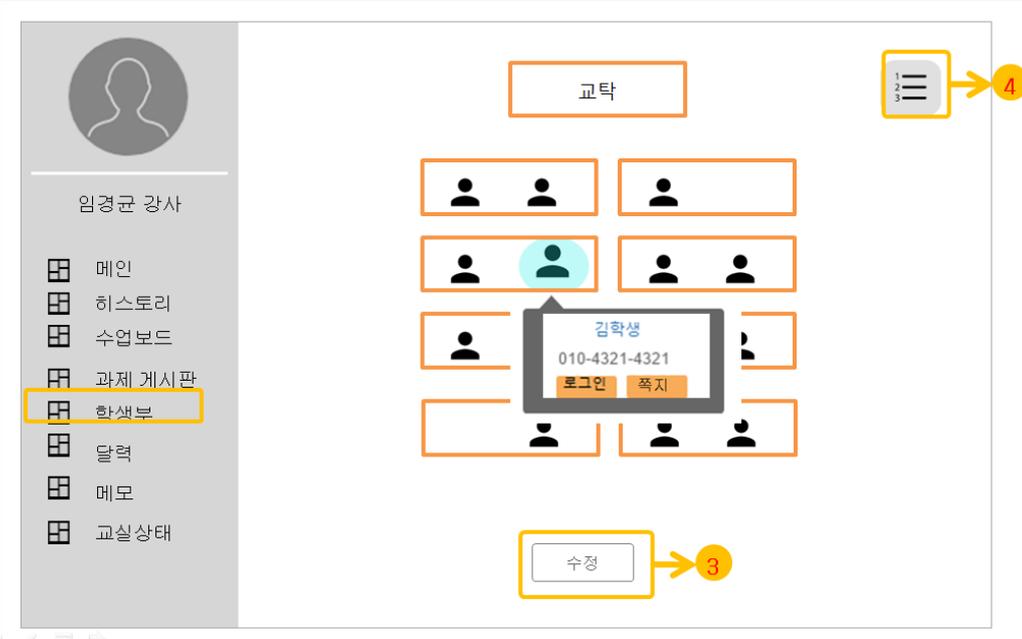
Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (과제 게시판)	Tool	오븐	Team	KOSTA 151 기 1조		



① 탭 추가 클릭
- 탭명(조명) 과 조원을 검색하여 추가할 수 있음

② 등록
- 버튼 클릭 시 탭이 추가 됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (학생부)	Tool	오븐	Team	KOSTA 151 기 1조		



① 학생부
좌석 배치도에 해당 학생 사진이 보여지는 형식의 학생부 출력

② 마우스 오버
마우스 오버시 툴팁으로 로그인 한 학생의 간략한 정보, 로그인 여부를 보여줌
쪽지 버튼

③ 수정
학생부 수정 기능 폼 이동

④ 전체 학생부
전체 학생부 표로 출력

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (학생부)	Tool	오븐	Team	KOSTA 151 기 1조		

수강생 목록
Home / student / 수강생 목록

수강생 목록 It shows the list of all students that quit.

Showing 1 to 10 of 39 entries

No	사진	성명(PIN)	전화번호	접속 여부	전화번호
39		김우진 (010101)	010-3461-4589	사용중	010-3461-4589
38		김학생 (154724)	010-4321-4321	사용중	010-4321-4321
37		장태희 (102938)	010-3461-4589	사용중	010-3461-4589
36		백수지 (123450)	010-3461-4589	사용중	010-3461-4589

① 전체 학생부
전체 학생부를 표로 보여주고 검색 할 수 있음

② 아이콘 클릭
아이콘 클릭 시 다시 좌석배치 화면으로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (학생부)	Tool	오븐	Team	KOSTA 151 기 1조		

교탁

학생

김학생
010-4321-4321
로그인

취소 저장

① 책상
드래그 앤 드롭으로 책상을 배치할 수 있음
책상 아이콘은 무한으로 생성됨

② 학생
드래그 앤 드롭으로 학생을 배치할 수 있음(학생 아이콘은 해당 학생의 가입 시 설정한 사진으로 띄워지며 한번 드래그하면 사라짐(중복방지))
마우스 오버시 툴팁으로 학생의 간략한 정보를 보여줌

③취소/저장
자리배치가 저장되고 학생부 카테고리 클릭 시 저장된 정보가 보여짐

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (달력)	Tool	오븐	Team	KOSTA 151 기 1조		

임경균 강사

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 교실상태

① 캘린더
일정을 추가 삭제 할 수 있다

② 그날 일정 클릭
세부내용 모달창 옴

③ 일정 추가
일정추가 모달창 옴

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (달력)	Tool	오븐	Team	KOSTA 151 기 1조		

임경균 강사

- 메인
- 히스토리
- 수업보드
- 과제 게시판
- 학생부
- 달력
- 메모
- 교실상태

① 일정

- 일정 내용 상세 출력
- 내용 링크를 클릭하면 해당 상세 페이지로 이동

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (달력)	Tool	오븐	Team	KOSTA 151 기 1조		

① 일정

- 일정 내용 추가
- Datepicker로 기간 설정
- +히스토리하면 히스토리 게시판에도 연동되어 데이터 추가됨

Project	Class_ic	Date	2017.06.05	Version	1.0	Page	1
Title	Content 영역 (메모)	Tool	오븐	Team	KOSTA 151 기 1조		

① 메모

- 메모 기능
- 메인 페이지에서 메모 등록도 가능하나 최근 3개의 메모만 출력이 페이지에서는 누적 메모 수정, 삭제, 관리 기능

PART 6

UML 다이어그램

**프로젝트에서 사용된 Class 를
Class Diagram 을 이용하여 보여줍니다.**

PART 7

DB Modeling

프로젝트에 사용된 Data Base 의 모델링을 보여줍니다.

PART 7. DB Modeling

■ 테이블 정의서

티 타입명		값한		작성일	2017.07.04		
테이블명		ClassicAuthority		작성자	한섭웨어		
테이블 설명							
번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	email	이메일	N/A	VARCHAR(100)	Not Null		PFK
2	authority	값한명	N/A	VARCHAR(50)	Not Null		
엔티티 타입명		멤버		작성일	2017.07.04		
테이블명		Member		작성자	한섭웨어		
테이블 설명							
번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	email	이메일	N/A	VARCHAR(100)	Not Null		PK
2	name	이름	N/A	VARCHAR(20)	Not Null		
3	pwd	패스워드	N/A	VARCHAR(500)	Not Null		
4	phone	핸드폰번호	N/A	VARCHAR(50)			
5	photoSrc	사진경로	N/A	VARCHAR(200)			
6	code	인증코드	N/A	VARCHAR(500)			
엔티티 타입명		출결관리		작성일	2017.07.04		
테이블명		Attendance		작성자	한섭웨어		
테이블 설명							
번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	attendNo	출결번호	N/A	INT	Not Null		PK
2	email	이메일	N/A	VARCHAR(100)	Not Null		FK
3	attendDate	출결일	N/A	DATE	Not Null		

4	inClass	입실시간	N/A	TIME		
5	outClass	퇴실시간	N/A	TIME		
6	attendState	출결상태	N/A	VARCHAR(20)	Not Null	
7	classCode	기수명(기수코드)	N/A	VARCHAR(100)	Not Null	

엔티티 타입명		쪽지	작성일		2017.07.04	
테이블명		Message	작성자		한섬웨어	
테이블 설명						

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	messageNo	쪽지 글번호	N/A	INT	Not Null		PK
2	msContent	내용	N/A	TEXT	Not Null		
3	msSendTime	보낸시간	N/A	DATETIME	Not Null		
4	emailS	이메일	N/A	VARCHAR(100)	Not Null		FK

엔티티 타입명		TO DO LIST	작성일		2017.07.04	
테이블명		ToDo	작성자		한섬웨어	
테이블 설명						

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	todoNo	TODO 게시번호	N/A	INT	Not Null		PK
2	email	이메일	N/A	VARCHAR(100)	Not Null		FK
3	todoList	내용	N/A	VARCHAR(300)			

엔티티 타입명		카테고리	작성일		2017.07.04	
테이블명		Category	작성자		한섬웨어	
테이블 설명						

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	cateCode	메인카테고리코드	N/A	VARCHAR(100)	Not Null		PK
2	cateTitle	카테고리명	N/A	VARCHAR(400)	Not Null		

3	email	이메일	N/A	VARCHAR(100)	Not Null		FK
---	-------	-----	-----	--------------	----------	--	----

엔티티 타입명		서브카테고리		작성일	2017.07.04		
----------------	--	--------	--	------------	------------	--	--

테이블명		SubCategory		작성자	한섬웨어		
-------------	--	-------------	--	------------	------	--	--

테이블 설명							
---------------	--	--	--	--	--	--	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	subcateCode	서브카테고리코드	N/A	VARCHAR(100)	Not Null		PK
2	cateCode	메인카테고리코드	N/A	VARCHAR(100)	Not Null		FK
3	subcateTitle	서브카테고리명	N/A	VARCHAR(400)	Not Null		
4	email	이메일	N/A	VARCHAR(100)	Not Null		FK

엔티티 타입명		통합 수업 게시판		작성일	2017.07.04		
----------------	--	-----------	--	------------	------------	--	--

테이블명		LectureBoard		작성자	한섬웨어		
-------------	--	--------------	--	------------	------	--	--

테이블 설명							
---------------	--	--	--	--	--	--	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	lectureNo	수업글 번호	N/A	INT	Not Null		PK
2	classCode	기수명(기수코드)	N/A	VARCHAR(100)	Not Null		FK
3	cateCode	메인카테고리코드	N/A	VARCHAR(100)	Not Null		FK
4	subcateCode	서브카테고리코드	N/A	VARCHAR(100)	Not Null		FK
5	lectureTitle	제목	N/A	VARCHAR(500)	Not Null		
6	lectureContent	내용	N/A	LONGTEXT	Not Null		
7	lectureDate	등록일	N/A	DATE			

엔티티 타입명		학생관리(조편성 UI)		작성일	2017.07.04		
----------------	--	--------------	--	------------	------------	--	--

테이블명		StudentGroup		작성자	한섬웨어		
-------------	--	--------------	--	------------	------	--	--

테이블 설명							
---------------	--	--	--	--	--	--	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	groupNo	조편성번호	N/A	INT	Not Null		PK

2	email	이메일	N/A	VARCHAR(100)	Not Null		FK
3	partyName	조명(서브)	N/A	VARCHAR(200)			
4	cateCode	새 컬럼	N/A	VARCHAR(100)			
5	groupLeaderState	조장상태	N/A	VARCHAR(20)			
6	classCode	기수번호	N/A	VARCHAR(100)			

엔티티 타입명	과제 카테고리(조편성)	작성일	2017.07.04
----------------	--------------	------------	------------

테이블명	GroupCategory	작성자	한심웨어
-------------	---------------	------------	------

테이블 설명	
---------------	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	GroupCateNo	과제카테고리번호	N/A	INT	Not Null		PK
2	cateCode	메인카테고리코드	N/A	VARCHAR(100)	Not Null		FK
3	email	이메일	N/A	VARCHAR(100)	Not Null		
4	classCode	기수코드(classCode)	N/A	VARCHAR(100)			
5	partyName	조명	N/A	VARCHAR(100)			

엔티티 타입명	달력(플캘린더)	작성일	2017.07.04
----------------	----------	------------	------------

테이블명	Calendar	작성자	한심웨어
-------------	----------	------------	------

테이블 설명	
---------------	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	calNo	일정번호	N/A	INT	Not Null		PK
2	classCode	기수명(기수코드)	N/A	VARCHAR(100)	Not Null		FK
3	calStart	시작일자	N/A	DATE			
4	calEnd	종료일자	N/A	DATE			
5	calTitle	일정제목	N/A	VARCHAR(400)			
6	calContent	일정내용	N/A	TEXT			
7	color	일정색깔	N/A	VARCHAR(50)	Not Null		

엔티티 타입명	강의(교실)관리	작성일	2017.07.04
----------------	----------	------------	------------

테이블명		Class	작성자	핵심웨어			
테이블 설명							
번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	classCode	기수명(기수코드)	N/A	VARCHAR(100)	Not Null		PK
2	classTitle	강좌명	N/A	VARCHAR(500)			
3	classStart	강의시작일	N/A	DATE			
4	classOpenTime	강의 시작시간	N/A	TIME			
5	classEnd	강의종료일	N/A	DATE			
6	classCloseTime	강의 종료시간	N/A	TIME			
7	email	이메일(강사)	N/A	VARCHAR(100)	Not Null		FK

엔티티 타입명	메모	작성일	2017.07.04				
---------	----	-----	------------	--	--	--	--

테이블명	Memo	작성자	핵심웨어				
------	------	-----	------	--	--	--	--

테이블 설명							
--------	--	--	--	--	--	--	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	memoNo	메모 글번호	N/A	INT	Not Null		PK
2	email	이메일	N/A	VARCHAR(100)	Not Null		FK
3	memoText	내용	N/A	TEXT	Not Null		
4	memoDate	등록일	N/A	DATETIME			

엔티티 타입명	과제 게시판(기수별)	작성일	2017.07.04				
---------	-------------	-----	------------	--	--	--	--

테이블명	AssignBoard	작성자	핵심웨어				
------	-------------	-----	------	--	--	--	--

테이블 설명							
--------	--	--	--	--	--	--	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	assignNo	과제글 번호	N/A	INT	Not Null		PK
2	classCode	기수명(기수코드)	N/A	VARCHAR(100)	Not Null		FK
3	cateCode	메인카테고리코드	N/A	VARCHAR(100)	Not Null		FK
4	GroupCateNo	과제카테고리번호	N/A	INT			FK

5	email	이메일	N/A	VARCHAR(100)	Not Null		FK
6	assignTitle	제목	N/A	VARCHAR(150)	Not Null		
7	assignContent	내용	N/A	TEXT	Not Null		
8	assignDate	등록일	N/A	DATE	Not Null		
9	assignNotice	과제(공지)여부	N/A	VARCHAR(20)			
10	partyName	조 이름	N/A	VARCHAR(200)			

엔티티 타입명		기수관리		작성일	2017.07.04		
테이블명		ClassMember		작성자	한섬웨어		
테이블 설명							

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	email	이메일	N/A	VARCHAR(100)			FK
2	classCode	기수명(기수코드)	N/A	VARCHAR(100)			FK

엔티티 타입명		파일		작성일	2017.07.04		
테이블명		FileLecture		작성자	한섬웨어		
테이블 설명							

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	fileNo	파일 글번호	N/A	INT	Not Null		PK
2	lectureNo	수업글 번호	N/A	INT	Not Null		FK
3	fileSrc	파일명 1	N/A	VARCHAR(500)			
4	fileSrc2	파일명 2	N/A	VARCHAR(500)			

엔티티 타입명		링크		작성일	2017.07.04		
테이블명		Link		작성자	한섬웨어		
테이블 설명							

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	linkNo	링크번호	N/A	INT	Not Null		PK

2	linkTitle	링크제목	N/A	VARCHAR(200)			
3	linkSrc	링크주소	N/A	VARCHAR(200)	Not Null		
4	lectureNo	수업글 번호	N/A	INT			
5	assignNo	과제글 번호	N/A	INT			
6	classCode	기수명(기수코드)	N/A	VARCHAR(100)			

엔티티 타입명	과제 게시판 파일	작성일	2017.07.04
----------------	-----------	------------	------------

테이블명	AssignBoardFile	작성자	한심웨어
-------------	-----------------	------------	------

테이블 설명	
---------------	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	assignFileNo	과제 게시판 파일번호	N/A	INT	Not Null		PK
2	assignNo	과제글 번호	N/A	INT	Not Null		FK
3	fileSrc	파일명 1	N/A	VARCHAR(500)			
4	fileSrc2	파일명 2	N/A	VARCHAR(500)			

엔티티 타입명	그날의 강의	작성일	2017.07.04
----------------	--------	------------	------------

테이블명	Today'sLecture	작성자	한심웨어
-------------	----------------	------------	------

테이블 설명	
---------------	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	lectureBM	강의 날짜(북마크 등록일)	N/A	INT	Not Null		PK
2	lectureNo	수업글 번호	N/A	INT	Not Null		FK
3	todayNo	오늘의 강의 넘버	N/A	INT			

엔티티 타입명	쪽지 발송인	작성일	2017.07.04
----------------	--------	------------	------------

테이블명	MessageSend	작성자	한심웨어
-------------	-------------	------------	------

테이블 설명	
---------------	--

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	sendNo	발송이력번호	N/A	INT	Not Null		PK

2	emailS	이메일(보내는 사람)	N/A	VARCHAR(100)	Not Null		FK
3	messageNo	쪽지 글번호	N/A	INT	Not Null		FK
4	deleteStateS	삭제여부	N/A	VARCHAR(20)			

엔티티 타입명	쪽지 수신인	작성일	2017.07.04
테이블명	MessageReceive	작성자	한섬웨어
테이블 설명			

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	receiveNo	수신이력번호	N/A	INT	Not Null		PK
2	messageNo	쪽지 글번호	N/A	INT	Not Null		FK
3	emailR	이메일(수신자)	N/A	VARCHAR(100)	Not Null		FK
4	readTime	읽은 시간	N/A	DATETIME			
5	readState	확인 상태(읽기전, 읽은후)	N/A	VARCHAR(20)			
6	deleteStateR	삭제여부	N/A	VARCHAR(20)			

엔티티 타입명	통합수업 강의별 관리	작성일	2017.07.04
테이블명	ClassByLecture	작성자	한섬웨어
테이블 설명			

번호	컬럼명	속성명	도메인	데이터타입	NULL 여부	기본값	KEY
1	lectureNo	수업글 번호	N/A	INT			FK
2	classCode	기수명(기수코드)	N/A	VARCHAR(100)			FK

PART 8

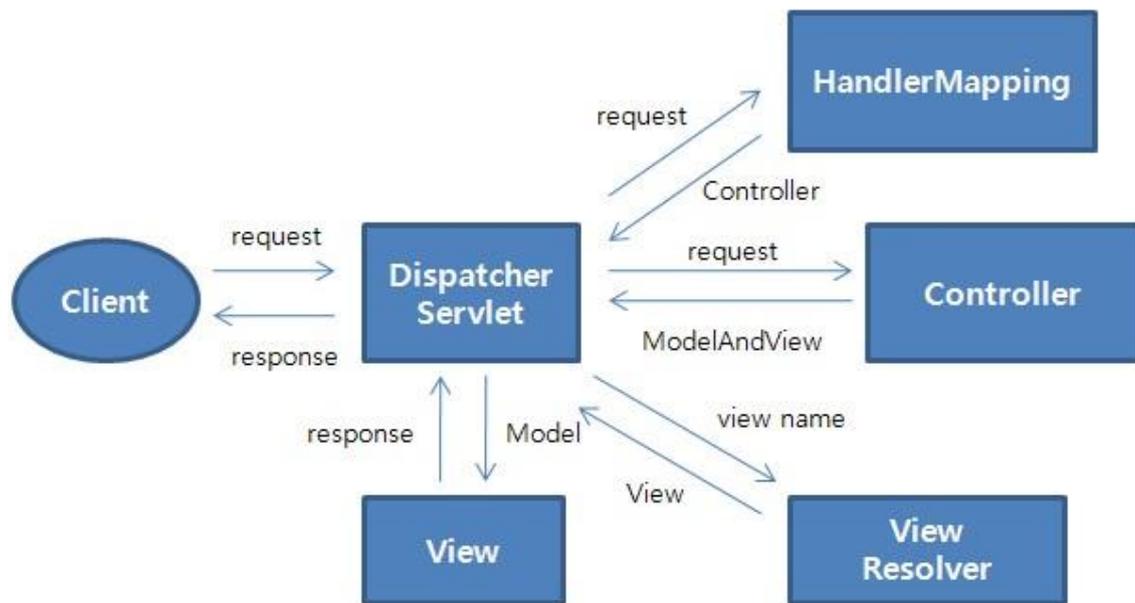
Project Setting

**프로젝트를 하기위한 모든 개발자들의 세팅을 맞추고
개발환경을 보여 줍니다.**

PART 8. Project Setting

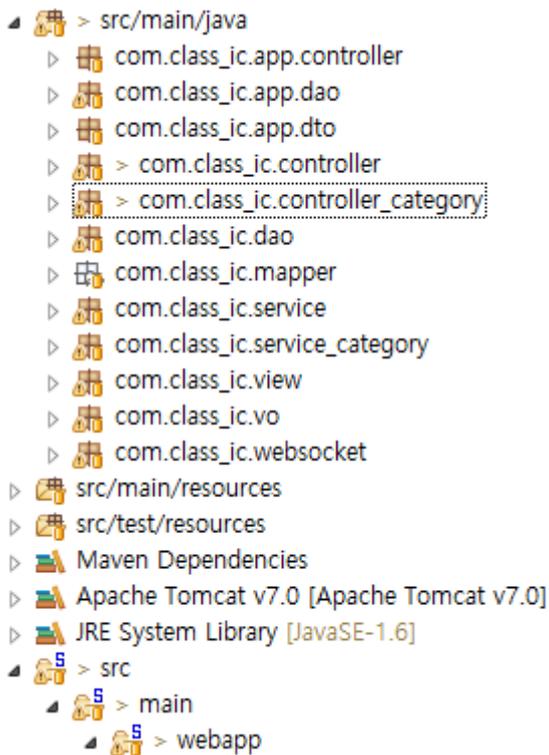
■ Spring MVC 프로젝트의 설정

MVC 패턴 기반 Spring Framework 프레임워크를 활용한 웹 개발 프로젝트



사용된 프레임워크는 Spring Framework 외에도 Mybatis, Maven을 함께 활용.

개발툴인 STS (Spring Tool Suite)의 Spring Legacy 템플릿을 활용하여, Spring MVC project로 패키지를 구성



전체적인 패키지 구성

Spring Legacy 템플릿 구성으로

src/main/java: Back-End 구현 java 파일

src/main/webapp: Front-End 구현 jsp 파일

MVC 패턴이므로, 패키지 구성

Controller / Service / DAO / VO

Mybatis 파일은 mapper 패키지로 구성

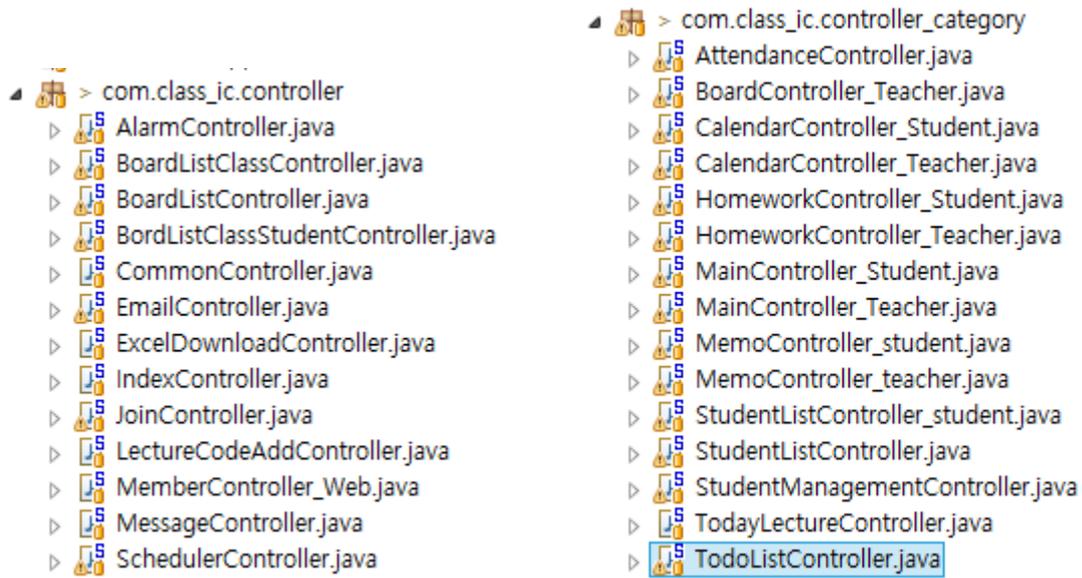
안드로이드 어플관련 파일또한 MVC 패턴을 활용하므로, 역할에 따른 패키지명은 동하되,

app을 추가해서 구성.

Controller

controller는 전체 사이트에 공통으로 사용하는 것은 controller

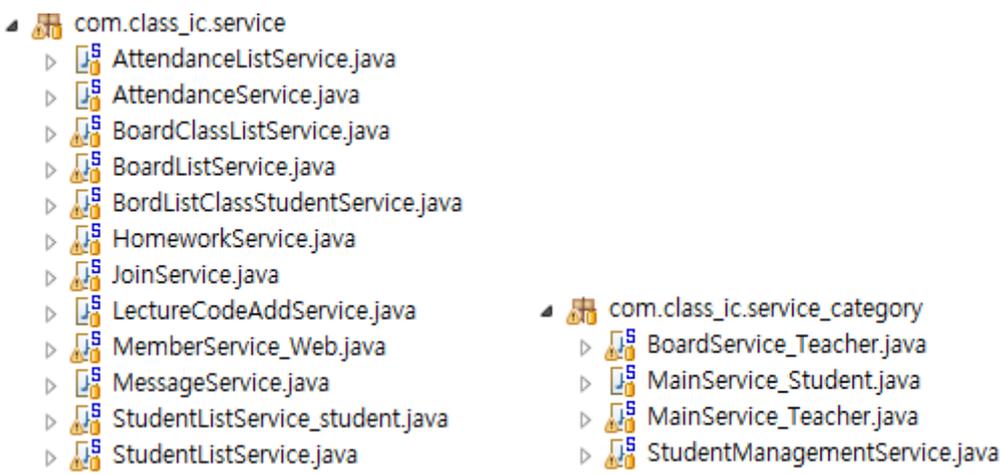
카테고리에서만 사용하는 것은 controller_category 패키지로 분리



- Service

service 또한 controller 와 마찬가지로 공통으로 사용하는 것은 service

카테고리에서만 사용하는 것은 service_category



- DAO

-  com.class_ic.dao
 - ▶  AlarmDAO.java
 - ▶  AttendanceDAO.java
 - ▶  AttendanceListDAO.java
 - ▶  BoardClassDAO.java
 - ▶  BoardDAO.java
 - ▶  CalendarDAO.java
 - ▶  HomeworkDAO.java
 - ▶  LectureAddDAO.java
 - ▶  MailDAO.java
 - ▶  MemberDAO.java
 - ▶  MemoDAO.java
 - ▶  MessageDAO.java
 - ▶  StudentListDAO.java
 - ▶  StudentManagementDAO.java
 - ▶  TeacherMainDAO.java
 - ▶  TestDAO.java
 - ▶  TodayLectureDAO.java
 - ▶  TodoListDAO.java

- VO

-  com.class_ic.vo
 - ▶  All_Alarm_DTO.java
 - ▶  AttandanceDTO.java
 - ▶  AttandanceListDTO.java
 - ▶  BoardVO.java
 - ▶  BookmarkDTO.java
 - ▶  CalendarDTO.java
 - ▶  CategoryDTO.java
 - ▶  ClassByLectureDTO.java
 - ▶  ClassDTO.java
 - ▶  Criteria.java
 - ▶  HomeworkDTO.java
 - ▶  LectureBoardDTO.java
 - ▶  LectureDTO.java
 - ▶  LinkDTO.java
 - ▶  MemberDTO.java
 - ▶  MemoVO.java
 - ▶  MessageDTO.java
 - ▶  PageMaker.java
 - ▶  SearchCriteria.java
 - ▶  StudentGroupDTO.java
 - ▶  StudentTableDTO.java
 - ▶  SubCategoryDTO.java
 - ▶  Test.java
 - ▶  TodayClassDTO.java
 - ▶  TodayLectureVO.java
 - ▶  TodaysLectureDTO.java
 - ▶  TodoListDTO.java

- mapper

- com.class_ic.mapper
 - Alarm_DAO.xml
 - AttendanceDAO.xml
 - AttendanceListDAO.xml
 - BoardClassDAO.xml
 - BoardDAO.xml
 - CalendarDAO.xml
 - HomeworkDAO.xml
 - LectureAddDAO.xml
 - MailDAO.xml
 - member.xml
 - MemberDAO.xml
 - memoDAO.xml
 - MessageDAO.xml
 - StudentListDAO.xml
 - StudentManagementDAO.xml
 - TeacherMainDAO.xml
 - TodayLectureDAO.xml
 - TodoListDAO.xml

- view

Bean Name Resolver를 통해 구현되는 코드 패키지

- com.class_ic.view
 - ExcelBuilder.java
 - ExcelBuilder2.java
 - QRCodeView.java

- websocket

웹소켓에 활용되는 코드 패키지

- com.class_ic.websocket
 - WebSocketHandler.java
 - WebSocketHandshakeInterceptor.java

■ POM.xml

```
Spring
pom.xml

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>
```

```
<artifactId>spring-expression</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>
```

```
<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- DB mysql -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.39</version>

</dependency>

<!-- File 처리 -->

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.3.1</version>

</dependency>

<!-- img scalr -->

<dependency>
```

```
<groupId>org.imgscalr</groupId>

<artifactId>imgscalr-lib</artifactId>

<version>4.2</version>

</dependency>

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>1.4</version>

</dependency>

<!-- Tiles (UI) -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-core</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-jsp</artifactId>

<version>2.2.2</version>

</dependency>

<!-- JSP UI JSTL -->
```

```
<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- log 작업 -->

<!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-api -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.6.6</version>

</dependency>

<!-- Mybatis 설정하기 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.1.1</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>
```

```
<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<!-- 인증과 권한 (Spring security) 설정 -->

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-core</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>
```

```
<artifactId>spring-security-taglibs</artifactId>
```

```
<version>4.0.4.RELEASE</version>
```

```
</dependency>
```

```
DB -mysql-
```

```
<!-- DB mysql -->
```

```
<dependency>
```

```
<groupId>mysql</groupId>
```

```
<artifactId>mysql-connector-java</artifactId>
```

```
<version>5.1.39</version>
```

```
</dependency>
```

```
파일 처리
```

```
<!-- File 처리 -->
```

```
<dependency>
```

```
<groupId>commons-fileupload</groupId>
```

```
<artifactId>commons-fileupload</artifactId>
```

```
<version>1.3.1</version>
```

```
</dependency>
```

```
<!-- img scalr -->
```

```
<dependency>
```

```
<groupId>org.imgscalr</groupId>
```

```
<artifactId>imgscalr-lib</artifactId>
```

```
<version>4.2</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>commons-io</groupId>
```

```
<artifactId>commons-io</artifactId>
```

```
<version>1.4</version>
```

```
</dependency>
```

ExcelFile 다운로드

```
<!-- ExcelFile 다운로드를 위한 dependency -->
```

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
```

```
<dependency>
```

```
<groupId>org.apache.poi</groupId>
```

```
<artifactId>poi</artifactId>
```

```
<version>3.9</version>
```

```
</dependency>
```

Tiles

```
<!-- Tiles (UI) -->
```

```
<dependency>
```

```
<groupId>org.apache.tiles</groupId>
```

```
<artifactId>tiles-core</artifactId>
```

```
<version>2.2.2</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.apache.tiles</groupId>
```

```
<artifactId>tiles-jsp</artifactId>
```

```
<version>2.2.2</version>
```

```
</dependency>
```

```
JSP
```

```
<!-- JSP UI JSTL -->
```

```
<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
```

```
<dependency>
```

```
<groupId>javax.servlet</groupId>
```

```
<artifactId>jstl</artifactId>
```

```
<version>1.2</version>
```

```
</dependency>
```

```
log
```

```
<!-- log 작업 -->
```

```
<dependency>
```

```
<groupId>org.slf4j</groupId>
```

```
<artifactId>slf4j-api</artifactId>
```

```
<version>1.6.6</version>
```

```
</dependency>
```

```
Mybatis
```

```
<!-- Mybatis 설정하기 -->
```

```
<dependency>
```

```
<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.1.1</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>
```

Security

```
<!-- 인증과 권한 (Spring security) 설정 -->

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-core</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>

<dependency>
```

```
<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-taglibs</artifactId>

<version>4.0.4.RELEASE</version>

</dependency>
```

Validation

```
<!-- validation -->

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.1.0.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>5.1.1.Final</version>

</dependency>
```

Email
<pre><!-- 이메일 --> <dependency> <groupId>org.springframework</groupId> <artifactId>spring-context-support</artifactId> <version>4.2.5.RELEASE</version> </dependency></pre>
Schedule
<pre><!-- schedule --> <dependency> <groupId>org.springframework</groupId> <artifactId>spring-context-support</artifactId> <version>4.0.4.RELEASE</version> </dependency> <dependency> <groupId>javax.mail</groupId> <artifactId>mail</artifactId> <version>1.4.7</version> </dependency></pre>
Ajax/JSON
<pre><!-- jackson --> <dependency></pre>

```
<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-core</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-core-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.9.13</version>
```

```
Ajax/JSON
```

```
<!-- jackson -->
```

```
<dependency>
```

```
<groupId>com.fasterxml.jackson.core</groupId>
```

```
<artifactId>jackson-core</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-core-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>net.sf.ezmorph</groupId>

<artifactId>ezmorph</artifactId>
```

```
<version>1.0.6</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>net.sf.json-lib</groupId>
```

```
<artifactId>json-lib</artifactId>
```

```
<version>2.4</version>
```

```
<classifier>jdk15</classifier>
```

```
</dependency>
```

```
QR
```

```
<!--QR zxing -->
```

```
<dependency>
```

```
<groupId>com.google.zxing</groupId>
```

```
<artifactId>core</artifactId>
```

```
<version>3.1.0</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>com.google.zxing</groupId>
```

```
<artifactId>javase</artifactId>
```

```
<version>3.1.0</version>
```

```
</dependency>
```

```
WebSocket
```

```
<!-- websocket -->
```

```
<dependency>

<groupId>javax.websocket</groupId>

<artifactId>javax.websocket-api</artifactId>

<scope>provided</scope>

<version>1.0</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-websocket</artifactId>

<version>4.0.9.RELEASE</version>

</dependency>
```

■ web.xml

Filter
<pre><!-- filter --> <filter> <filter-name>EncodingFilter</filter-name> <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class> <init-param> <param-name>encoding</param-name> <param-value>UTF-8</param-value> </init-param> <init-param> <param-name>forceEncoding</param-name> <param-value>>true</param-value> </init-param> </filter> <filter-mapping> <filter-name>EncodingFilter</filter-name> <url-pattern>/*</url-pattern> </filter-mapping></pre>
Security (web.xml)
<pre><!-- Security --> <filter></pre>

```
<filter-name>springSecurityFilterChain</filter-name>

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

</filter>

<filter-mapping>

<filter-name>springSecurityFilterChain</filter-name>

<url-pattern>/*</url-pattern>

</filter-mapping>
```

Error

```
<!--error -->

<error-page>

<error-code>404</error-code>

<location>/WEB-INF/views/error/404.jsp</location>

</error-page>

<error-page>

<error-code>500</error-code>

<location>/WEB-INF/views/error/500.jsp</location>

</error-page>
```

■ root-context.xml

DB 설정

```
<bean name="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="com.mysql.jdbc.Driver"/>
<property name="url"
value="jdbc:mysql://192.168.0.49:3306/classic?useSSL=false"/>
<property name="username" value="kosta"/>
<property name="password" value="1004"/>
</bean>
```

Email

```
<!-- gmail -->
<bean id="mailSender"
class="org.springframework.mail.javamail.JavaMailSenderImpl" >
<property name="host" value="smtp.gmail.com" />
<property name="port" value="587" />
<property name="defaultEncoding" value="utf-8"/>
<property name="username" value="kolsun93@gmail.com" />
<property name="password" value="kolsun930717." />
<property name="javaMailProperties">
<props>
<prop key="mail.smtp.starttls.enable">true</prop>
```

```
<prop key="mail.smtp.auth">true</prop>

<prop key="mail.debug">true</prop>

<prop key="mail.transport.protocol">smtp</prop>

</props>

</property>

</bean>
```

■ **spring-servlet.xml**

Mybatis

```
<!-- mybatis -->
```

```
<beans:bean id="sqlSessionFactoryBean"
```

```
class="org.mybatis.spring.SqlSessionFactoryBean">
```

```
<beans:property name="dataSource"
```

```
ref="dataSource">
```

```
</beans:property>
```

```
<beans:property name="mapperLocations"
```

```
value="classpath*:com/class_ic/mapper/*.xml" />
```

```
</beans:bean>
```

```
<beans:bean id="sqlSession"
```

```
class="org.mybatis.spring.SqlSessionTemplate">
```

```
<beans:constructor-arg index="0"
```

```
ref="sqlSessionFactoryBean" />
```

```
</beans:bean>
```

Resolver

```
<!-- Resolver -->
```

```
<beans:bean
```

```
class="org.springframework.web.servlet.view.BeanNameViewResolver">
```

```
<beans:property name="order" value="0" />
```

```
</beans:bean>
```

```

<beans:bean id="urlBasedViewResolver"
class="org.springframework.web.servlet.view.UrlBasedViewResolver">
<beans:property name="viewClass"
value="org.springframework.web.servlet.view.tiles2.TilesView" />
<beans:property name="order" value="1" />
</beans:bean>
<beans:bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<beans:property name="order" value="2" />
<beans:property name="prefix" value="/WEB-INF/views/" />
<beans:property name="suffix" value=".jsp" />
</beans:bean>

```

스케줄링

```

<!-- scheduler -->
<task:annotation-driven scheduler="scheduler" />
<task:scheduler id="scheduler" pool-size="10" />

```

쪽지, 알림 (웹소켓)

```

<!-- websocket -->
<websocket:handlers>
<websocket:mapping handler="chatHandler" path="/chat-ws.htm" />
<websocket:handshake-interceptors>
<beans:bean

```

```

class="com.class_ic.websocket.WebSocketHandshakeInterceptor" />

</websocket:handshake-interceptors>

</websocket:handlers>

<beans:bean id="chatHandler"

class="com.class_ic.websocket.WebSocketHandler" />

<mvc:default-servlet-handler />

```

파일업로드

```

<!-- 파일 업로드 -->

<beans:bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<beans:property name="maxUploadSize" value="10485760" />

</beans:bean>

```

Tiles

```

<!-- Tiles 설정하기 -->

<beans:bean id="tilesConfigurer"

class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">

<beans:property name="definitions">

<beans:list>

<beans:value>/WEB-INF/Tiles.xml</beans:value>

</beans:list>

</beans:property>

</beans:bean>

```

■ security-context.xml

```
Security 설정
<security:http use-expressions="true">
<security:csrf disabled="true"/>
<security:form-login
login-processing-url="/login.htm"
username-parameter="email"
password-parameter="pwd"
authentication-failure-url="/login.htm"/>
<security:logout logout-url="/logout.htm" l
logout-success-url="/index.htm" invalidate-session="true" />
<security:intercept-url p
attern="/teacher*" access="hasAnyRole('ROLE_TEACHER')"/>
<security:intercept-url pattern="/student*" access="hasAnyRole('ROLE_TEACHER',
'ROLE_STUDENT')"/>
</security:http>
<bean id="bcryptEncoder"
class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder" />
```

PART 9

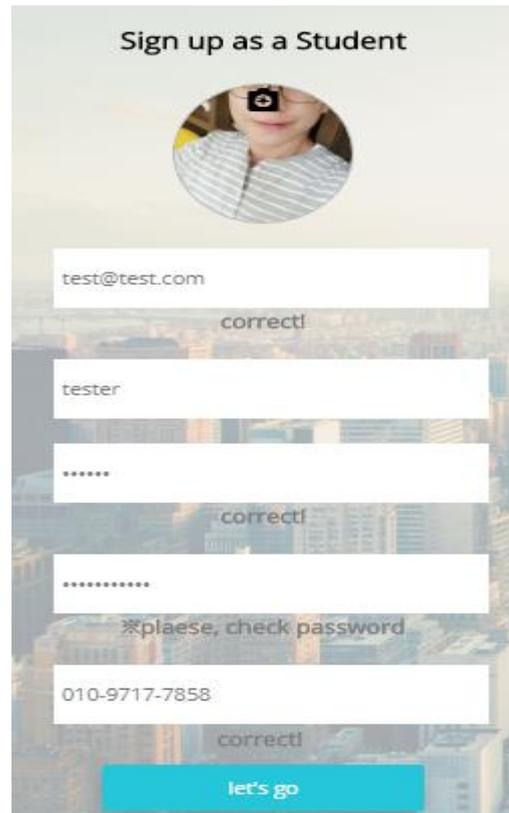
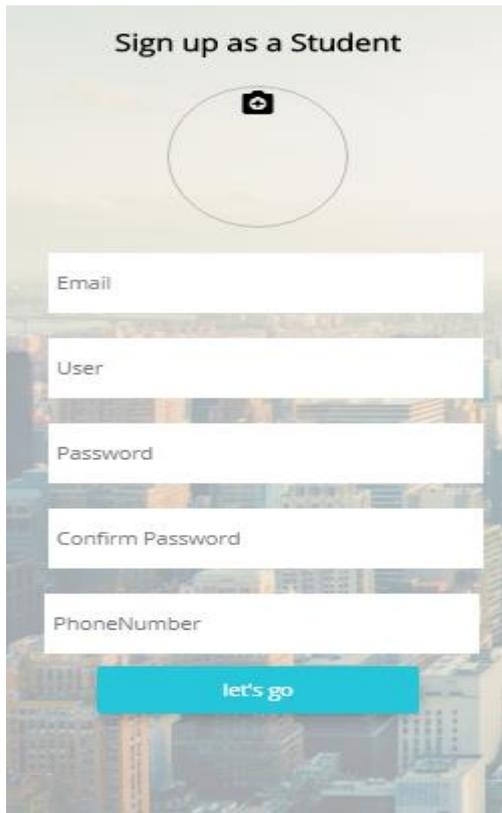
시스템 구현

프로젝트에서 쓰였던 기능들에 대한 코드를 설명해 줍니다.

PART 9. 시스템 구현

■ 회원가입 validation

Contoller	JoinController.java		
View Page	joinus.jsp	Main Function	로그인 validation
Technology	Jquery keyup, Spring validation		
코드 설명	회원가입 validation 을 통해, keyup 하는 순간, 해당 회원가입 양식이 맞는지 확인해준다.		



```

$('#uid, #pwd_st ,#pwdconfirm_st, #email_st,
#phone_st').after('<strong style="color:gray;FONT-SIZE:15px"></strong>');

email_st.keyup(function(){
    var s = $(this).next('strong');
    if (email_st.val().length == 0) { // 입력 값이 없을 때
        s.text('');
    }// strong 요소에 포함된 문자 지움
    else if( re_mail.test(email_st.val()) != true ){ // 유효하지
않은 문자 입력 시
        s.text('*Please, confirm email form');
    }else if( re_mail.test(email_st.val()) == true ){

```

```

        s.text('correct!');
    }
});

// #upw 인풋에서 onkeyup 이벤트가 발생하면
pwd_st.keyup( function() {
    var s = $(this).next('strong'); // strong 요소를 변수에 할당
    if (pwd_st.val().length == 0) { // 입력 값이 없을 때
        s.text(''); // strong 요소에 포함된 문자 지움
    } else if (pwd_st.val().length < 6) { // 입력 값이 6보다 작을 때
        s.text('※Please, write more longer'); // strong문자출력
    } else if (pwd_st.val().length > 18) { // 입력 값이 18보다 클 때
        s.text('※Please, Write more shorter'); // strong문자출력
    } else { // 입력 값이 6 이상 18 이하일 때
        s.text('correct!'); // strong 요소에 문자 출력
    }
});

pwdconfirm_st.keyup( function() {
    var s = $(this).next('strong'); // strong 요소를 변수에 할당
    if (pwdconfirm_st.val().length == 0) { // 입력 값이 없을 때
        s.text(''); // strong 요소에 포함된 문자 지움
    } else if (pwdconfirm_st.val() == pwd_st.val()) {
        s.text('password equal!');
    } else {
        s.text('※plaease, check password');
    }
});

// #tel 인풋에 onkeydown 이벤트가 발생하면
// 하이픈(-) 키가 눌렸는지 확인
// 하이픈(-) 키가 눌렀다면 입력 중단
phone_st.keyup( function() {

    var s = $(this).next('strong');
    if (phone_st.val().length == 0) { // 입력 값이 없을 때
        s.text('');
    } // strong 요소에 포함된 문자 지움
    else if( re_tel.test(phone_st.val()) != true ){
    // 유효하지 않은 문자 입력 시
        s.text('※Please, Confirm your tel number');

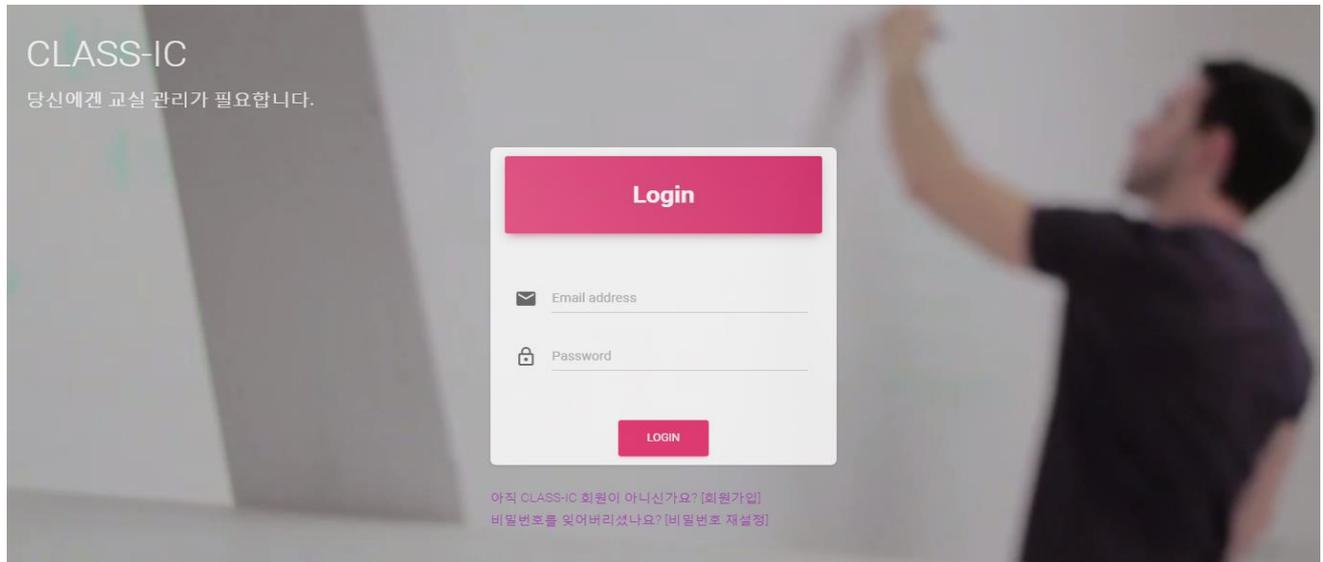
    } else if( re_tel.test(phone_st.val()) == true ){
        s.text('correct!');
    }
});

```

```
}  
});
```

■ 로그인

Table	Member/ClassicAuthority	Mapper	MemberDAO
Contoller	MemberController_Web	Service	MemberService_Web
View Page	main	Main Function	로그인
Technology	Spring Security		
코드 설명	암호화된 비밀번호로 Security 권한에 맞게 로그인		



```
public ModelAndView loginService(HttpSession session,String email, @RequestParam("pwd") String
rawPassword, ModelAndView mv){

    MemberDAO member_dao = sqlsession.getMapper(MemberDAO.class);
    AlarmDAO alarm_DAO = sqlsession.getMapper(AlarmDAO.class);
    MemberDTO member = member_dao.login(email);
    String encodedPassword = member.getPwd();
    String memberAuthority = member_dao.confirmAuthority(email);
    boolean result = bCryptPasswordEncoder.matches(rawPassword, encodedPassword);
    mv.addObject("member", member);
    session.setAttribute("email", member.getEmail());

    if(result){
        if(memberAuthority.equals("ROLE_TEACHER")){
            mv.setViewName("common/thselect_teacher");
            System.out.println("강사 로그인 성공");
        }else{
            mv.setViewName("common/thselect_student");
            System.out.println("학생 로그인 성공");}
        System.out.println("비밀번호 일치");
    }else{
        mv.setViewName("common/main");
        System.out.println("비밀번호 확인 실패");}
    return mv;
}
```

■ PASSWORD 찾기

Table	Member	Mapper	MailDAO.xml
Contoller	EmailController.java	Service	mailSender.createMimeMessage()
View Page	searchPW.jsp	Main Function	패스워드 찾기
Technology	Mailsender 객체를 이용하여 서버의 내용을 해당 이메일로 보내준다.		
코드 설명	Mailsender 객체에 구글 아이디,비밀번호를 저장해준뒤, 이메일 보내기 요청이 오면, 난수를생성하여 해당 이메일로 보내줌과 동시에 난수를 DB 에 저장한다. 사용자는 그 난수를 입력하여 비밀번호를 바꿀수 있다. mailsende 클래스에서 title,content 를 입력한후 createMimeMessage(); 를 해주면, 이메일이 전송된다.		



☰ 이메일검색 🔍 상세 ▾ 받은메일함 | 11 / 75 📧 인원은메일 삭제

답장 전체답장 전달 숨 삭제 스팸신고 안읽음 이동 ▾ ... 번역

☆ 5tulips@naver.com 님의 비밀번호 안내입니다. 📧

보낸사람 ☆ <kolsun93@gmail.com>
받는사람 <5tulips@naver.com>

인증코드 : 89505

인증코드를 입력하여 비밀번호를 변경하세요!



aaa@naver.com	최석환	\$2a\$10\$Y8rc1BLzO55evs5xoot3iuv1b/4lO4cOw...	01097177858	asd
a@a.com	hwan	\$2a\$10\$4BGBoJt1vMOvSirwo16cPuT9Erk83Itaz...	010-9717-7858	11
5tulips@naver.com	최석환2	1004	1212	1212

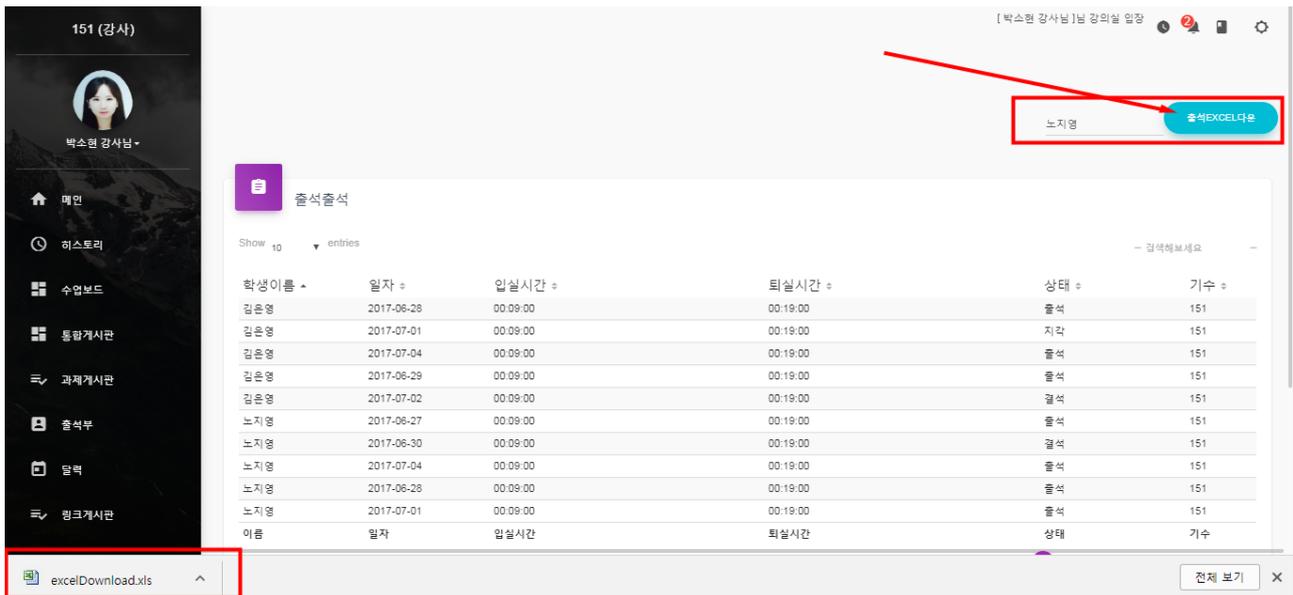
```
String rand = Integer.toString((int)(Math.random()*100000));
MailDAO maildao = sqlsession.getMapper(MailDAO.class);
    maildao.update_Rand(email,rand);

MimeMessage message = mailSender.createMimeMessage();
    MimeMessageHelper messageHelper
        = new MimeMessageHelper(message, true, "UTF-8");
String title = email+" 님의 비밀번호 안내입니다.";
String content = "인증코드 : "rand";
    messageHelper.setTo(email); // 받는사람 이메일
    messageHelper.setSubject(title); // 메일제목은 생략가능
    messageHelper.setText(content); // 메일 내용

    mailSender.send(message);
```

■ 엑셀다운로드

Table	Attendance	Mapper	AttendanceListDAO.xml
Contoller	ExcelDownloadController.java	Service	AttendanceListService
View Page	attendance.jsp	Main Function	buildExcelDocument
Technology	POI jar 파일을 이용하고, buildExcel 클래스를 통해 엑셀을 다운로드 한다.		
코드 설명	학생이름을 입력한 후 엑셀다운로드 버튼을 누르면 컨트롤러와 서비스를 거쳐 Mapper AttendanceListDAO.xml 에서 select 하여 DB 값들을 가져와, buildExcelDocument 에서 엑셀 출력 파일을 만들어준다. buildExcelDocument 에서는 출력하고싶은 컬럼값들과 변수들을 미리 설정해 주어야 한다.		



```
// get data model which is passed by the Spring container
List<AttendanceDTO> attendanceList
= (List<AttendanceDTO>) model.get("memberattendacnelist");

// create a new Excel sheet
HSSFWorkbook sheet
= (HSSFWorkbook) workbook.createSheet("Java Books");

sheet.setDefaultColumnWidth(10);

// create style for header cells
CellStyle style = workbook.createCellStyle();
Font font = workbook.createFont();
font.setFontName("Arial");
style.setFillForegroundColor(HSSFColor.BLUE.index);
style.setFillPattern(CellStyle.SOLID_FOREGROUND);
font.setBoldweight(HSSFFont.BOLDWEIGHT_BOLD);
font.setColor(HSSFColor.WHITE.index);
style.setFont(font);

for (AttendanceDTO attendance : attendanceList) {
```

```
        HSSFRow aRow = sheet.createRow(rowCount++);
        aRow.createCell(0).setCellValue(attendance.getEmail());
        SimpleDateFormat transFormat = new
SimpleDateFormat("yyyy-MM-dd");
        String attenddate =
transFormat.format(attendance.getAttendDate());
        aRow.createCell(1).setCellValue(attenddate);

        aRow.createCell(2).setCellValue(attendance.getInClass());

        aRow.createCell(3).setCellValue(attendance.getOutClass());

        aRow.createCell(4).setCellValue(attendance.getAttendState());

        aRow.createCell(5).setCellValue(attendance.getClassCode());

    }
```

■ 과제 게시판

Table	ASSIGNBOARD	Mapper	HomeworkDAO
Contoller	HomeworkController_Teacher, HomeworkController_Student	Service	HomeworkService
View Page	homework_board	Main Function	카테고리 별 조 추가, 카테고리,조 별 글 검색
Technology	Javascript,ajax,jquery,java,bootstrap		
코드 설명	(강사) 통합게시판에서 추가된 과정(메인 카테고리)선택 하고 조 추가 Ajax 로 데이터를 서버로 넘기고 넘긴 데이터는 HomeworkController_Teacher 에서 HomeworkService 를 통해서 DB 에 저장된다.		

```
//메인 카테고리 출력
function showMainCate() {
var email = "<%= (String)session.getAttribute("email")%>";

$.ajax({
    type : "post",
    url:"selectCate.htm",
    data : {"email" : email},
    dataType : 'Json',
    success : function(data) {
        $.each(data, function(){
            $(".selectCateList").append("<option value='"+this.cateCode+"'>" +
this.cateCode + "</option> ");
        });
    },
    error:function(request, status,
```

```

error){
                                //console.log(error);
                                alert("code:" + request.status + "\n" +
"message:" + request.responseText + "\n"+ "error: " +error )
                                }

                                });
    } //end showMainCate

    //조 : 팀명 저장
    function addHomework() {

        var email = "<%= (String)session.getAttribute("email")%>";
        var classCode =
"<%= (String)session.getAttribute("classCode")%>";

        console.log(email)

        var cateCode = $("#selectCateList01 option:selected").val();
        var partyName = $("#partyName").val();

        $.ajax({

            type : "post",
            url:"addHomework.htm",
            data : {"cateCode": cateCode,
"partyName":partyName, "email":email, "classCode":classCode},
            dataType: 'text',
            success : function(data) {

                                    $("#partyName").val("");
                                },

            error:function(request, status, error){
                alert("code:" + request.status + "\n" + "message:" +
request.responseText + "\n"+ "error: " +error )
            }

        });

    } //end addHomework

    //카테고리에 따른 조 출력
    function showTeamList() {

        var email = "<%= (String)session.getAttribute("email")%>";
        var classCode =
"<%= (String)session.getAttribute("classCode")%>";
        var cateCode = $("#selectCateList02").val();

        console.log("showTeamList cateCode : "+cateCode)
        console.log(classCode)
        console.log(email)

        $.ajax({

            type : "post",

```

```

classCode , "cateCode":cateCode},
    url:"selectTeam.htm",
    data : {"email" : email, "classCode":
    dataType : 'Json',
    success : function(data) {
        $("#selectTeamList").empty();
        $.each(data, function(){
            $("#selectTeamList").append("<option value='"+this.partyName+"'> " +
            this.partyName + "</option> ");
            console.log("partyName: "+this.partyName)
        });
        selectCateCodeList();
    },
    error:function(request, status, error){
        alert("code:" + request.status + "\n" +
        "message:"+ request.responseText + "\n"+ "error: " +error )
    }
    });
} //end showTeamList

//과제게시판 전체 정렬
function selectAllList() {
    var email = "<%= (String)session.getAttribute("email")%>";
    var classCode =
    "<%= (String)session.getAttribute("classCode")%>";

    $.ajax({
        type : "post",
        url:"selectAllList.htm",
        data: {"email" : email, "classCode": classCode },
        dataType : 'html',
        success : function(data) {
            $('#selectdatatable').html(data);
        },
        error:function(request, status, error){
            //console.log(error);
            alert("code:" + request.status + "\n" + "message:" +
            request.responseText + "\n"+ "error: " +error )
        }
    });
}
}

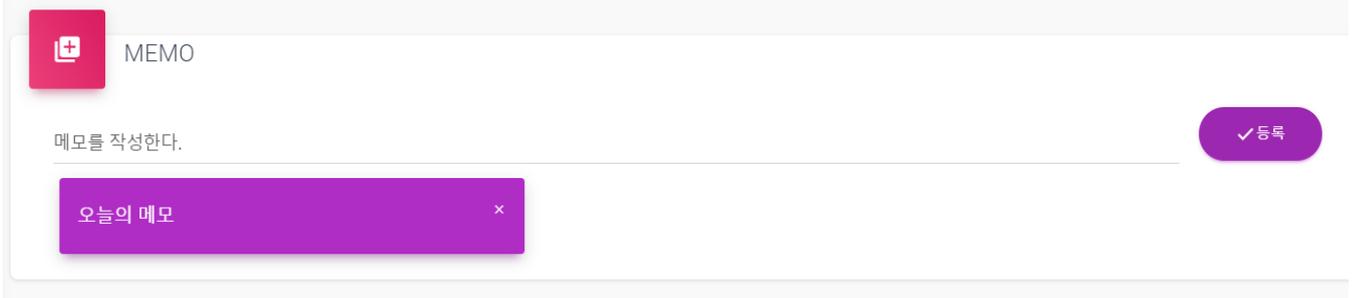
```

```
//partyName별 출력
//서브카테고리가 변경이 되면 Ajax를 태움
function selectCateCodeList() {
    var classCode =
"<%= (String)session.getAttribute("classCode")%>";
    var cateCode = $("#selectCateList02").val();
    var partyName = $('#selectTeamList').val();

    $.ajax({
        type : "post",
        url : 'homeworkSelectList.htm',
        data : {
            cateCode : cateCode,
            classCode : classCode,
            partyName : partyName
        },
        dataType : 'text',
        success : function(data) {
            $('#selectdatatable').empty();
            $('#selectdatatable').html(data);
        }
    });
}
});
```

■ 메모 게시판

Table	Memo	Mapper	memoDAO
Contoller	MemoController_teacher, MemoController_student-		
View Page	Teacher_main	Main Function	비동기로 글 등록,삭제 (ajax)
Technology	Javascript,ajax,jquery,java,bootstrap		
코드 설명	글을 작성시 ajax 를 이용해서 비동기로 데이터를 넘기고 MemoController 에서 Mybatis 를 이용해 DB 에 저장한다.		



```
$(function() {

    //글 출력 : 2017.06.21 최은혜

    var email = sessionId;

    $.ajaxSetup({
        cache : false
    });
    $.ajax({
        cache : false,
        type : "post",
        url : "selectMemo.htm",
        data : {
            "email" : email
        },
        dataType : 'json',
        async : false,
        success : function(data) {

            $.each(

data,

function(index, item) {

$('.memo-content')

.append(

<div class="alert alert-primary item col-xs-4"
```

```

id="'+item.memono+'"> <input type="hidden" class="memoNo" value="'+item.memono+ '"
name="memoNo" />'

        + item.memotext

        + '<button class="deletebutton close"
aria-hidden="true" value="'+item.memono+'"><i class="material-
icons">close</i></button></div>')

    console.log("memoNO: "
        + item.memono)

    });

    }

    });

});
</script>
<!-- end 메모 -->

<!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.0.0/moment.min.js"></
script>
<script type="text/javascript">
$(document).ready(function() {

    $.ajax({
        type : "get",
        url : "boardmain.htm",
        dataType : 'html',
        success : function(data) {

            $('#boardlist').html(data);

        }

    });
});

```

■ 데이터 테이블

Table	LectureBoard	Mapper	BoardDAO
Contoller	BoardListController	Service	BoardListService
View Page	totalLectureBoard.jsp	Main Function	데이터 테이블
Technology	Java, jsp, css, javascript, jquery, html, spring framework, ajax, bootstrap		
코드 설명	데이터 테이블을 이용한 게시판 구현.		

The screenshot shows a web application interface for a 'Teacher Integrated Management Page'. On the left is a dark sidebar with a user profile for '151 (강사) 박소원 강사님' and navigation links: '메인', '이력', '수업보드', '통합게시판', '과제게시판', '중석부', '달력', '링크게시판'. The main content area is titled '교사 통합 관리 페이지' and contains an 'integrated management system' section. It features a search bar and a table with the following data:

check	글번호	category	sub category	제목	작성날짜	Actions
<input type="checkbox"/>	34	JAVA	파일 업로드	IO는 쉽다	2017-07-04	✎
<input type="checkbox"/>	33	JAVA	파일 업로드	IO는 쉽다	2017-07-04	✎
<input type="checkbox"/>	32	JAVA	파일 업로드	파일 업로드이러니	2017-07-04	✎
<input type="checkbox"/>	31	DATABASE	oracle	오라클 사용법	2017-07-04	✎
<input type="checkbox"/>	30	C언어	C언어 할계미트 포인터	포인터라니 포인터라니	2017-07-04	✎
<input type="checkbox"/>	29	C언어	C언어 할계미트 포인터	포인터 밖의 무한애록에 빠져봅시다 용감!	2017-07-04	✎
<input type="checkbox"/>	28	C언어	C언어 할계미트 포인터	포인터가 어렵다구요?????	2017-07-04	✎
<input type="checkbox"/>	27	C언어	C언어 기초	연산자	2017-07-04	✎

```
$(document).ready(
    function() {

        $('#datatables').DataTable(
            {
                "pagingType" : "full_numbers",
                "lengthMenu" : [ [ 10, 25, 50, -1 ],
                                [ 10, 25, 50, "All" ] ],
                responsive : true,
                language : {
                    search : "_INPUT_",
                    searchPlaceholder : "Search records",
                }
            }
        );

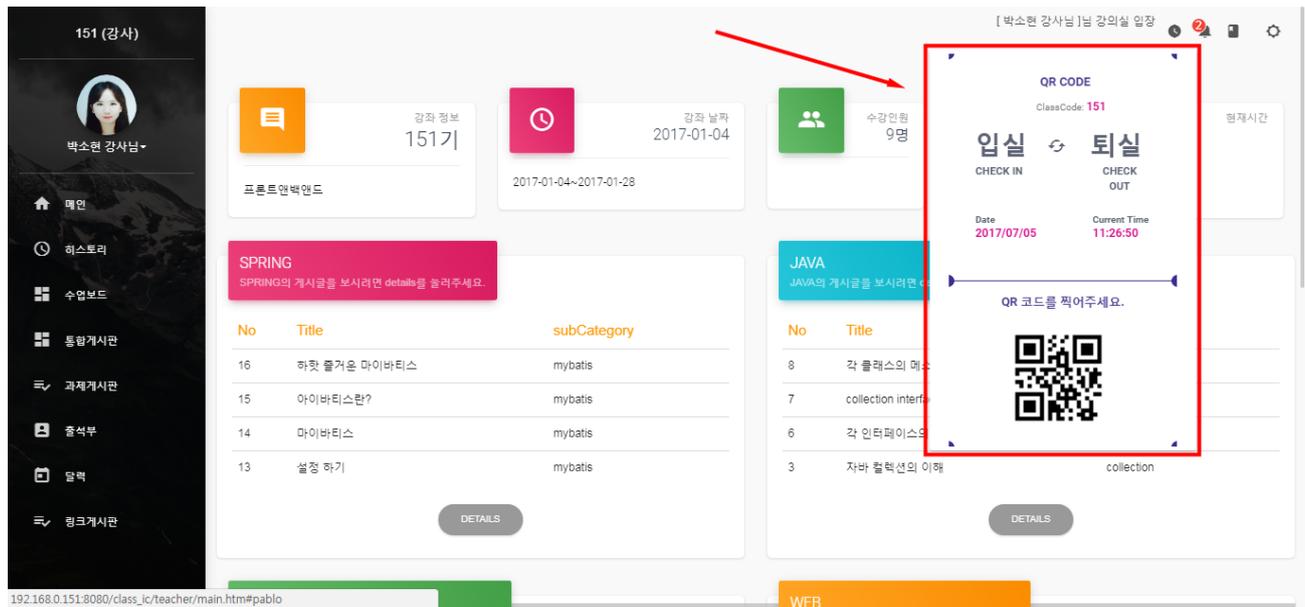
        var table = $('#datatables').DataTable();

        // Delete a record
        table.on('click', '.remove', function(e) {
            $tr = $(this).closest('tr');
            table.row($tr).remove().draw();
            e.preventDefault();
        });
    });
```

```
    $('.card .material-datatables label').addClass('form-group');  
});
```

■ QR Code(Zxing)

Table	MEMBER,ATTENDANCE		
Contoller	LectureCodeAddController		
View Page	헤더를 갖고 있는 모든 페이지	Main Function	renderMergedOutputModel
Technology	Zxing		
코드 설명	BeanNameViewResolver 이용해 Controller 에서 Return 되는 ModelAndView 이름을 가져가 Resolver 에서 Java Class 함수를 이용해 이미지 생성		



QRCodeView.java

```
public class QRCodeView extends AbstractView {
    public QRCodeView() {
        // 뷰의 콘텐츠 타입 설정
        // 이미지로 뿌려야 함.
        setContentType("image/png; charset=UTF-8");
    }
    /*
    @description : QR 생성에서 중요한 부분 가져온 Parameter 를 QR 로 변경되는 부분
    */
    @Override
    protected void renderMergedOutputModel(Map<String, Object> model, HttpServletRequest req,
        HttpServletResponse res)
        throws Exception {
        res.setContentType(getContentType());
        String userAgent = req.getHeader("User-Agent");
        boolean ie = userAgent.indexOf("MSIE") > -1;
        String fileName = "QRCode.png";
        if (ie) {
            fileName = URLEncoder.encode(fileName, "UTF-8");
        } else {
            fileName = new String(fileName.getBytes("UTF-8"), "ISO-8859-1");
        }
        res.setHeader("Content-Transfer-Encoding", "binary");
        OutputStream out = res.getOutputStream();
        QRCodeWriter qrCodeWriter = new QRCodeWriter();
        String text = (String) model.get("content");
        text = new String(text.getBytes("UTF-8"), "ISO-8859-1");
        BitMatrix bitMatrix = qrCodeWriter.encode(text, BarcodeFormat.QR_CODE, 100, 100);
        MatrixToImageWriter.writeToStream(bitMatrix, "png", out);
    }
}
```

```

        out.flush();
    }
}

```

QRCodeView.java

```

/*
 * @description : 페이지 로드시 QR 코드를 불러내기 위한 URL
 */
@RequestMapping("/createCode.htm")
public ModelAndView createCode(@RequestParam String content){
    //ModelAndView 바로 리턴
    return new ModelAndView("qrcodeview", "content", content);
}

```

Script

```

<script>
    /* 페이지 로드시 QR 코드 가져옴. */
    $(document).ready(function() {
        $('#qrclick').click(function() {
            var url = "/class_ic/common/createCode.htm";
            $("#img").attr("src", url + "?content=" + sessionClassCode);
        });
    });
</script>

```

■ 수업보드

✓ Table	LectureBoard	Mapper	BoardClassDAO
Contoller	BoardListClassController	Service	BoardClassListService
View Page	Board.jsp	Main Function	데이터 테이블, Spring
Technology	Java, jsp, css, javascript, jquery, html, spring framework, ajax, bootstrap		
코드 설명	통합게시판에서 기수로 보내진 글들을 볼 수 있게 해 준다.		

ORACLE ADD

DATABASE/oracle

Show 10 entries Search records

check	글번호	category	sub category	제목	작성날짜	Actions
<input type="checkbox"/>	21	DATABASE	oracle	ORACLE 11G	2017-07-04	
<input type="checkbox"/>	20	DATABASE	oracle	디렉토리	2017-07-04	
<input type="checkbox"/>	19	DATABASE	oracle	검사 후 이런 오류	2017-07-04	
<input type="checkbox"/>	12	DATABASE	oracle	로그인 하는 창	2017-07-04	

Showing 1 to 4 of 4 entries FIRST PREVIOUS 1 NEXT LAST

✓ 글쓰기 ✕ 체크 삭제

DATABASE
DATABASE의 게시글을 보시려면 details를 눌러주세요.

No	Title	subCategory
21	Oracle 11g	oracle
20	디렉토리	oracle
19	검사 후 이런 오류	oracle
12	로그인 하는 창	oracle

DETAILS

WEB
WEB의 게시글을 보시려면 details를 눌러주세요.

No	Title	subCategory
25	잡껴하는 자바 스크립트	jquery
24	제이쿼리는 쉽다.	jquery
23	제이쿼리	jquery
22	자바스크립트를 읽으니 제이쿼리로 오라	jquery

DETAILS

C언어
C언어의 게시글을 보시려면 details를 눌러주세요.

No	Title	subCategory
30	포인터라니 포인터라니	C언어 할계이로 포인터
29	포인터 별의 무한매력에 빠져봅시다 풍덩!	C언어 할계이로 포인터
28	포인터가 어렵다구요?????	C언어 할계이로 포인터
27	연산자	C언어 기초

DETAILS

PLUS CATEGORY

board count : 5
새로운 게시판을 생성하려면 이곳을 눌러 생성 하십시오!

ADD BOARD

```

package com.class_ic.dao;

import java.util.ArrayList;

import com.class_ic.vo.ClassByLectureDTO;
import com.class_ic.vo.LectureBoardDTO;
import com.class_ic.vo.SubCategoryDTO;

public interface BoardClassDAO {

    //기수에 담겨져있는 알맞는 카테고리 select
    ArrayList<SubCategoryDTO > selectCate(String email);

    //카테고리에 맞는 게시글을 select 해옴
    ArrayList<LectureBoardDTO> selectCateboard(SubCategoryDTO dto);

    //카테고리와 서브카테고리에 알맞는 게시글을 select 해옴.
    ArrayList<LectureBoardDTO> selectCateSubcateboard(SubCategoryDTO dto);

    //각 카테고리에 알맞는 서브카테고리를 중복 없이 select 해옴.
    ArrayList<SubCategoryDTO> selectSubcate(SubCategoryDTO dto);

    // 출력된 게시글 중 x 버튼을 누를 경우에 수업담기에 담겨 있던 것을 지워줌.
    int deletx(ClassByLectureDTO dto);

    // 기수 담기테이블에 insert
    public void boardMultiSend(ClassByLectureDTO dto);

    // 수정화면에 출력
    public ArrayList<LectureBoardDTO> totalboardEdit(int lectureNo);

    //학생에 맞는 강사님 이메일 찾기

    public String selectTeacher(String classCode);

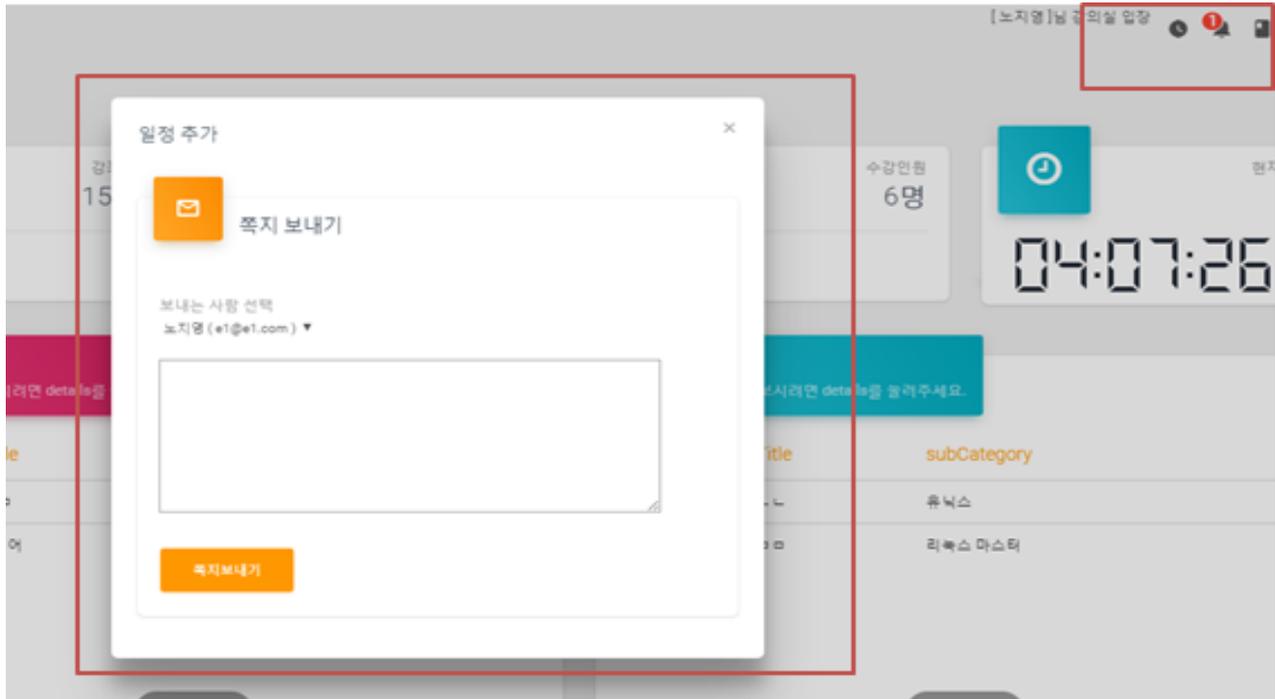
}

```

:위와 같은 주요 기능들을 가지고 있다.

■ 웹소켓

Table	Message, MessageReceive, MessageSend	Mapper	Alarm_DAO
Contoller	AlarmController		
View Page	Header.jsp	Main Function	소켓
Technology	Java, jsp, css, javascript, jquery, html, spring framework, ajax, bootstrap		
코드 설명	Spring 에서 제공하는 웹소켓을 이용한 알람 기능과 쪽지 기능.		



```
public class WebSocketHandler extends TextWebSocketHandler {

    private Map<Object, WebSocketSession> users = new ConcurrentHashMap();
    private Map<String, String> ids = new ConcurrentHashMap();

    @Override
    public void afterConnectionEstablished(WebSocketSession session) throws Exception {

        log("접속 성공" + session.getId() + "웹소켓 세션 아이디");
        String userid = (String) session.getAttributes().get("email");

        System.out.println("userID!!! : " + userid);

        users.put(userid, session);
        ids.put(session.getId(), userid);
    }
}
```

```

@Override
public void afterConnectionClosed(WebSocketSession session, CloseStatus status) throws Exception
{
    log((String) session.getAttributes().get("userId"));

    users.remove("parammsg");
}

@Override
protected void handleTextMessage(WebSocketSession session, TextMessage message) throws
Exception {

    System.out.println("-----");

    System.out.println("받은 ID 값 : " + message.getPayload());

    String msg = message.getPayload();

    for (WebSocketSession s : users.values()) {
        System.out.println("-----선택받은사람-----");
        System.out.println("아이디>" + msg + "<");
        System.out.println("-----");
        System.out.println(msg + "에게 메세지 전송!");
        if (ids.get(s.getId()).equals(msg)) {

            s.sendMessage(message);

        } else {

        }

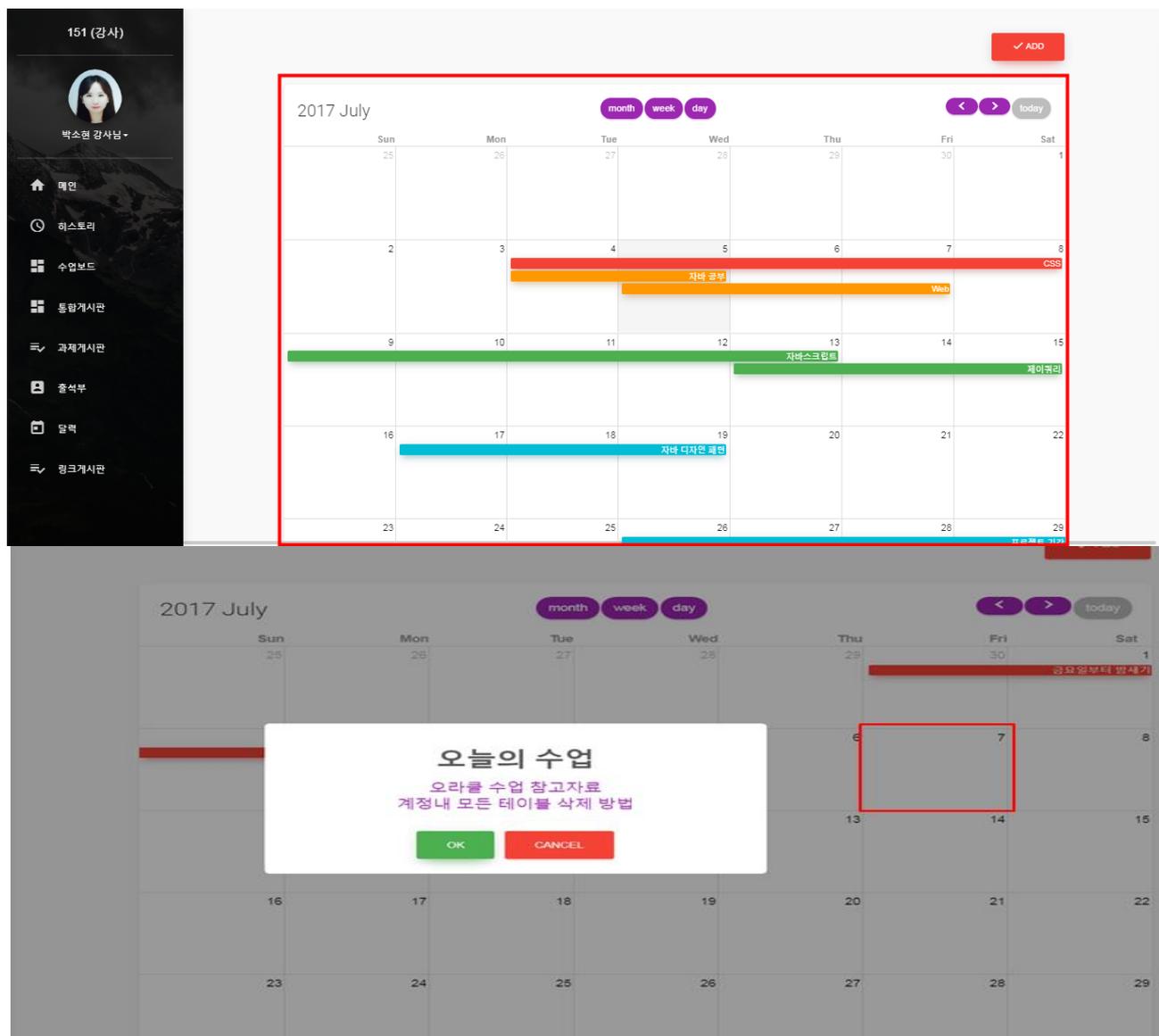
        log(s.getId() + "에게 메시지 전송");
    }
}
}

```

✓

■ 풀 캘린더

Table	TodayLecture, LectureBoard	Mapper	CalendarDAO.xml
Contoller	CalendarController_Student, CalendarController_Teacher		
View Page	calendar.jsp	Main Function	fullcalendar
Technology	Javascript, ajax, jquery, java, bootstrap		
코드 설명	Fullcalendar 내부함수 select 에서 ajax 를 이용하여 비동기로 DB 와 연동하여 일정을 등록하고 삭제할 수 있음. 추가된 일정은 히스토리와 연동.		



주요 코드 넣으세요

```
// 캘린더 시작
initFullCalendar: function(){

    $calendar = $('#fullCalendar');

    today = new Date();
    y = today.getFullYear();
    m = today.getMonth()+1;
    d = today.getDate();

    $calendar.fullCalendar({
        viewRender: function(view, element) {
            // We make sure that we activate the perfect scrollbar when the view isn't on
            Month
            if (view.name != 'month'){
                $(element).find('.fc-scroller').perfectScrollbar();
            }
        },
        header: {
            left: 'title',
            center: 'month,agendaWeek,agendaDay',
            right: 'prev,next,today'
        },
        defaultDate: today,
        selectable: true,
        selectHelper: true,
        views: {
            month: { // name of view
                titleFormat: 'YYYY MMMM'
                // other view-specific options here
            },
            week: {
                titleFormat: " YYYY MMM D "
            },
            day: {
                titleFormat: 'YYYY MMMM D'
            }
        },
        select: function(start, end, event, jsEvent, view, calEvent,today ) {

            var todaylist=new Array();
            var html="";
            var clickdate=start._d;
            console.log(clickdate)

            $.ajax({
                url:"todayclass.htm",
                type:"POST",
                data:{clickdate: clickdate},
                dataType:'json',
                success:function(data){

                    var html='';
                    $.each(data, function() {

                        console.log(this.todayTitle);
                        console.log(this.lectureNum);

                        html+= '<a
href="totalclassBoard_contentview.htm?lectureNo='+this.lectureNum+' ">'+this.todayTitle+'</a>
```

```

<br>'

        });

        // on select we show the Sweet Alert modal with an input
        swal({
            title: '오늘의 수업',
            html: html,
            showCancelButton: true,
            confirmButtonClass: 'btn btn-success',
            cancelButtonClass: 'btn btn-danger',
            buttonsStyling: false
        }) //swal 끝

    }
}); //ajax 끝

},

    editable: true,
    eventLimit: true, // allow "more" link when too many events
    /* color classes: [ event-blue | event-azure | event-green | event-orange | event-
red ],*/
    events: {url : 'CalendarList.htm' },
    eventClick: function(calEvent, jsEvent, view) {
        var test="테스트";
        var id=calEvent.id;
        console.log(calEvent); //아이디 값
        console.log(jsEvent);
        console.log(view);
        swal({
            title: 'Are you sure?',
            text: 'You will not be able to recover this calendar!',
            type: 'warning',
            showCancelButton: true,
            confirmButtonText: '일정 삭제하기',
            cancelButtonText: '일정 유지하기',
            confirmButtonClass: "btn btn-success",
            cancelButtonClass: "btn btn-danger",
            buttonsStyling: false
        }).then(function() {

            $.ajax({

                type : "GET",
                url : "CalendarEditDelete.htm",
                data: {id: id},
                dataType : "text",
                error : function(){
                    swal({
                        title: 'Cancelled ',
                        text: '일정 삭제를 실패 하셨습니다. 관리자에게 문의하세요.',
                        type: 'error',
                    });
                }
            });
        });
    }
}

```

```
        confirmButtonClass: "btn btn-info",
        buttonsStyling: false
    })

    },
    success : function(data){

        swal({
            title: 'Deleted!',
            text: '삭제가 완료 되었습니다.',
            type: 'success',
            confirmButtonClass: "btn btn-success",
            buttonsStyling: false
        }).then(function() {

            location.href="calendar.htm";
        })

    }

    });

}, function(dismiss) {
    // dismiss can be 'overlay', 'cancel', 'close', 'esc', 'timer'
    if (dismiss === 'cancel') {
        swal({
            title: 'Cancelled',
            text: '삭제를 취소 하셨습니다.',
            type: 'error',
            confirmButtonClass: "btn btn-info",
            buttonsStyling: false
        })
    }
})

}

});

},
}
}
```

■ 출석 여부 판단

Table	Attendance, member	Mapper	AttendanceDAO
Contoller	SchedulerController	Service	AttendanceService
View Page	data_table.jsp	Main Function	스케줄링, JAVA
Technology	Java, jsp, css, javascript, jquery, html, spring framework, ajax, bootstrap		
코드 설명	Spring 스케줄링을 이용한 출석판단.		

출석출석

학생이름	일자	입실시간	퇴실시간	상태	기수
이현정	2017-07-02	00:09:00	00:19:00	출석	151
이현정	2017-06-27	00:09:00	00:19:00	출석	151
이현정	2017-06-30	00:09:00	00:19:00	출석	151
이현정	2017-06-28	00:09:00	00:19:00	결석	151
이현정	2017-07-01	00:09:00	00:19:00	지각	151
이현정	2017-06-26	00:09:00	00:19:00	출석	151
이현정	2017-06-29	00:09:00	00:19:00	출석	151
이름	일자	입실시간	퇴실시간	상태	기수

우리반 출석률

LEGEND

- 출석
- 결석
- 지각
- 휴학

@Controller

```
public class SchedulerController {
    @Autowired
    private SqlSession sqlSession;

    @Autowired
    private AttendanceService attendanceservice;

    // 0 정상 출근 , 1 지각
    private int compareLate=1;
    // 0 정상 퇴근 , 2 조퇴
    private int compareEarlyLeave=2;
    // 3 결석

    @Scheduled(cron="0 42 17 * * *")
    public void insert() {
        System.out.println("아침입니다 출석된 테이블을 insert 하겠습니다.");
        attendanceservice.insertAttendance();
    }
}
```

```

        System.out.println("attendance 테이블 insert 완료 되었습니다.");
    }
    @Scheduled(cron="0 42 17 * * *")
    public void updateState() {
        AttendanceDAO attendanceDao=sqlsession.getMapper(AttendanceDAO.class);
        System.out.println("저녁입니다 출석된 테이블의 결석여부를 판단하겠습니다.");

        ArrayList<AttendanceDTO> todaylist=attendanceDao.todayAttendanceData();
        System.out.println(todaylist.size());
        LectureDTO classdto= new LectureDTO();

        //내가 찍은 입실 시간 인덱스 1 이 시간 2 가 분이다.
        String[] inclassArr;
        //내가 찍은 퇴실 시간 인덱스 1 이 시간 2 가 분이다.
        String[] outclassArr;
        //지정된 입실 시간 인덱스 0 이 시간 1 이 분이다.
        String[] opentimeArr;
        //지정된 퇴실 시간 인덱스 0 이 시간 1 이 분이다.
        String[] closetimeArr;

        for(AttendanceDTO list:todaylist){
            String attendState="";
            classdto=attendanceDao.selectStartEndTime(list.getClassCode());
            System.out.println(classdto.getClassOpenTime());
            System.out.println(classdto.getClassCloseTime());
            System.out.println(list.getInClass());
            System.out.println(list.getOutClass());

            //둘다 널이 아니면(입퇴실 둘다 찍은 사람만 출석 판단함.
            if((list.getInClass()!=null)&&(list.getOutClass()!=null)){

                inclassArr=list.getInClass().split(":");
                outclassArr=list.getOutClass().split(":");
                opentimeArr=classdto.getClassOpenTime().split(":");
                closetimeArr=classdto.getClassCloseTime().split(":");

                compareLate(inclassArr[1],inclassArr[2],opentimeArr[0],opentimeArr[1]);
                compareEarlyLeave(outclassArr[1],outclassArr[2],closetimeArr[0],closetimeArr[1]);
                System.out.println("구분:"+compareLate+","+compareEarlyLeave);
                System.out.println("구분 2:"+(compareLate+compareEarlyLeave));
                System.out.println();
                System.out.println(list.getInClass()+","+list.getOutClass());
            }
        }
    }

```

```

switch ((compareLate+compareEarlyLeave)) {
//정상 출결
case 0:
    attendState="출석";
    break;//지각
case 1:
    attendState="지각";
    break;//조퇴
case 2:
    attendState="조퇴";
    break;//결석
case 3:
    attendState="결석";
    break;
default:
    break;
}
System.out.println(list.getEmail());
System.out.println(attendState);
String email=list.getEmail().trim();
System.out.println(attendanceDao.updateAttendaceState(attendState,email)); }
}

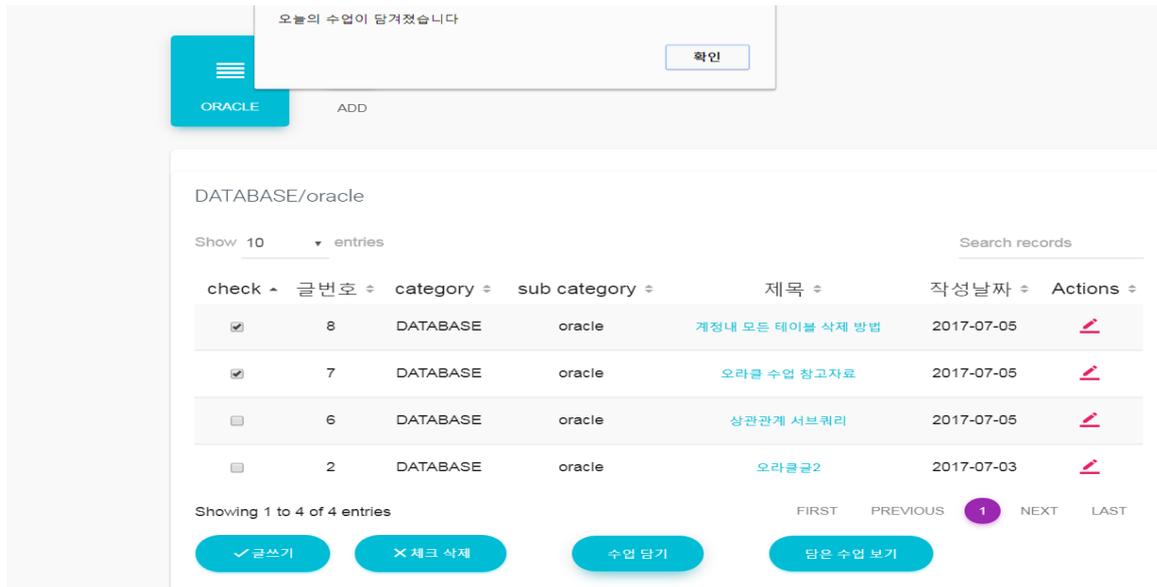
```

-insert() 메소드에 매일 아침 스케줄링이 가동되어 member 에 존재하는 모든 학생에게 오늘일자의 Attendance 컬럼에 default 값을 insert 시켜준다.

-updateState() 메소드에 매일 저녁 스케줄링이 가동되어 아침에 insert 된 자신의 오늘 날짜의 컬럼을 기수에 시간에 맞게 update 를 시켜준다.

■ 오늘의 강의 저장

Table	TodayLecture, LectureBoard	Mapper	TodayLectureDAO, CalendarDAO
Contoller	TodayLectureController		
View Page	classboardlist_data_table.jsp, todayLecture_storage.jsp	Main Function	fullcalendar
Technology	Javascript, ajax, jquery, java, bootstrap		
코드 설명	Checkbox 로 선택한 값을 비동기로 DB 와 연동하여 오늘의 수업 테이블에 오늘 날짜로 등록한다.		



```
// 멀티컨텐츠 선택받기
function multi_storage()
{
    var cnt2 = $("input[name='multy[]']:checked").length;

    if(cnt2 < 1){
        alert(" 게시물을 선택하여 주세요");
        return;
    }

    var chk = document.getElementsByName("multy[]");
    var len = chk.length;    //체크박스의 전체 개수
// var checkRow = '';    //체크된 체크박스의 value를 담기위한 변수
    var checkCnt = 0;    //체크된 체크박스의 개수
    var checkLast = '';    //체크된 체크박스 중 마지막 체크박스의 인덱스를 담기위한 변수
    var cnt = 0;
    var rowid = new Array();    //체크된 체크박스의 모든 value 값을 담는다

    for(var i=0; i<len; i++){

        if(chk[i].checked == true){
            checkCnt++;    //체크된 체크박스의 개수
        }
    }
}
```

```

        checkLast = i;    //체크된 체크박스의 인덱스
    }
}

var count=0;
for(var i=0; i<len; i++){
    if(chk[i].checked == true){ //체크가 되어있는 값 구분
rowid.push(chk[i].value);
        count++;
    }

}

var data="";
for(var i=0;i<count;i++){
    if(i==count-1){
        data+= rowid[i];

        }else{

            data+= rowid[i]+",";

        }

}
alert(data);
//
    alert(rowid);    //'value1', 'value2', 'value3' 의 형태로 출력된다.
//ajax 로 보낼데이터를 배열형태로 허용해준당
jQuery.ajaxSettings.traditional = true;

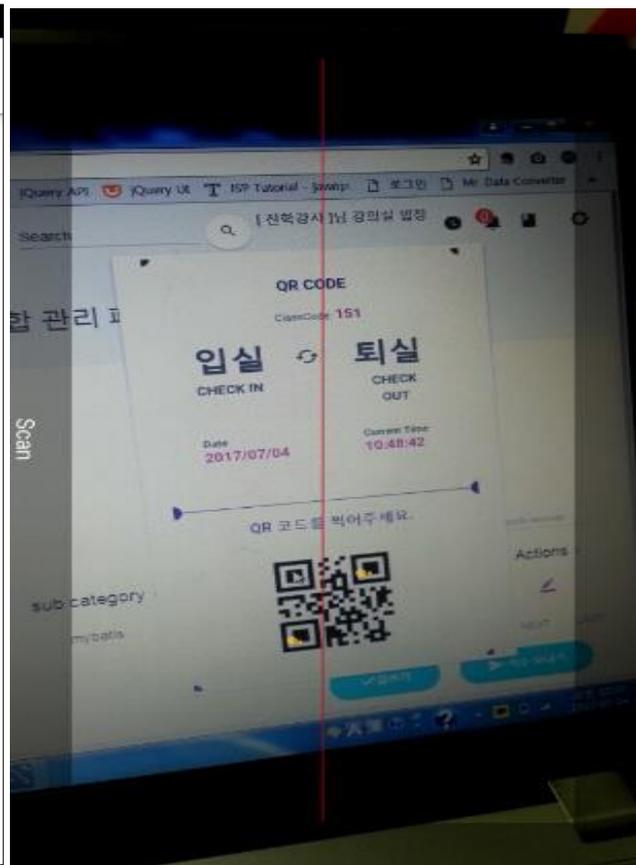
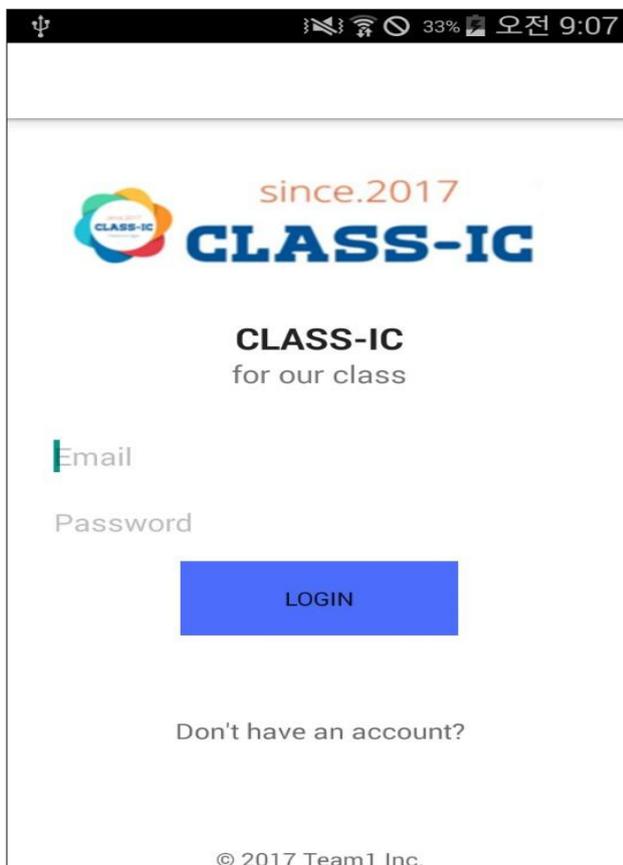
$.ajax({
    type: 'POST',
    url: 'multi_storage.htm',
    data: { data: data } ,
    dataType: 'text',

    success: function() {
        alert('오늘의 수업이 담겨졌습니다 ');
    },
    error: function() {
        alert('오늘의 수업이 안 담겨졌네요 다시 담아주세요');
    }
});
}
</script>

```

■ Android

Table	MEMBER,ATTENDANCE	Mapper	LectureaddDao,attendanceDAO,Member
Contoller	MemberController		
View Page	Activity_main.xml activity_reader.xml ,login.xml	Main Function	ReaderActivity,MainActivity
Technology	Android,Zxing		
코드 설명	같은 Ip 를 쓰는 웹 서버를 통해 스프링 방식으로 시큐리티 처리된 ID 와 패스워드를 가져와 일치 여부 확인 후 안드로이드 코드에서는 일치되면 JSON 1 값을 가져와 로그인을 완료시킨 후 QR 스캐너 코드를 동작시킨다 QR 스캐너에서 안드로이드 카메라로 인식을 시키면 또 다시 웹 IP 주소를 통해 값을 전달 하며 DB 에 입력시킨다.		



MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    //SharedPreferences -> id 만 넣어주기
    private EditText etEmail;
    private EditText etPassword;
    private Button btnRegist;
    SharedPreferences pref;
    SharedPreferences.Editor editor;
    private String Email;
    private String ps;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login);
        pref=getSharedPreferences("Login", Activity.MODE_PRIVATE);
```

```

editor = pref.edit();
etEmail = (EditText) findViewById(R.id.etEmail);
etPassword = (EditText) findViewById(R.id.etPassword);
btnRegist = (Button) findViewById(R.id.l_login);
btnRegist.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new Post().execute();
    }
});
btnRegist.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        new Post().execute();
    }
});
}
private class Post extends AsyncTask<Void,Void,String> {
    //R.string.base_uri => http://192.168.1.102:8080
    final String url = "http://192.168.0.151:8080"+"class_ic/aplogin";
    @Override
    protected void onPreExecute() {
        Email=etEmail.getText().toString();
        ps=etPassword.getText().toString();
    }
    @Override
    protected String doInBackground(Void... params) {
        MultiValueMap<String, String> parameters = new LinkedMultiValueMap<>();
        parameters.add("id",Email);
        parameters.add("pw", ps);
        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);
        HttpEntity<MultiValueMap<String, String>> request = new
        HttpEntity<>(parameters, headers);
        RestTemplate restTemplate = new RestTemplate();
        List<HttpMessageConverter<?>> messageConverters = new
        ArrayList<HttpMessageConverter<?>>();
        messageConverters.add(new FormHttpMessageConverter());
        messageConverters.add(new StringHttpMessageConverter());
        restTemplate.setMessageConverters(messageConverters);
        String result = restTemplate.postForObject(url, parameters, String.class);
        return result;
    }
    @Override
    protected void onPostExecute(String s) {
        JSONObject json = null;
        try {
            json = new JSONObject(s);
            if(json.getString("count").equals("1")){
                editor.putString("id",Email);
                editor.commit();
                Toast.makeText(getApplicationContext(), "로그인이 완료되었습니다.",
                Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(MainActivity.this,ReaderActivity.class);
                intent.putExtra("id",Email);
                startActivity(intent);
                finish();
            }else{
                Toast.makeText(getApplicationContext(),"로그인이
                안됨",Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
  }
}

```

ReaderActivity.java

```

public class ReaderActivity extends AppCompatActivity{
    private Button scan_btn;
    private Button scan_btn2;
    private SharedPreferences pref;
    private String state;
    String qr_content;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reader);
        pref=getSharedPreferences("Login", Activity.MODE_PRIVATE);
        scan_btn = (Button) findViewById(R.id.scan_btn);
        scan_btn2=(Button) findViewById(R.id.scan_btn2);
        final Activity activity = this;
        scan_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                state="inClass";
                Intent intent = getIntent();
                IntentIntegrator integrator = new IntentIntegrator(activity);
                integrator.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE_TYPES);
                integrator.setPrompt("Scan");
                integrator.setCameraId(0);
                integrator.setBeepEnabled(false);
                integrator.setBarcodeImageEnabled(false);
                integrator.initiateScan();
            }
        });
        scan_btn2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                state="outClass";
                Intent intent = getIntent();
                IntentIntegrator integrator = new IntentIntegrator(activity);
                integrator.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE_TYPES);
                integrator.setPrompt("Scan");
                integrator.setCameraId(0);
                integrator.setBeepEnabled(false);
                integrator.setBarcodeImageEnabled(false);
                integrator.initiateScan();
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        IntentResult result = IntentIntegrator.parseActivityResult(requestCode,
resultCode, data);
        if (result != null) {
            if (result.getContents() == null) {
                Toast.makeText(this,"You cancelled the scanning",
Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(this, result.getContents(), Toast.LENGTH_LONG).show();
                qr_content=result.getContents();
            }
        }
    }
}

```

```

        Log.d("qr",qr_content);
        new Post().execute();
        // 자신을 호출한 Activity 로 데이터를 보낸다.
        finish();
    }
} else {
    super.onActivityResult(requestCode, resultCode, data);
}
}
private class Post extends AsyncTask<Void,Void,String> {
    //R.string.base_uri => http://192.168.1.102:8080
    final String url = "http://192.168.0.151:8080"+"class_ic/attendance";
    @Override
    protected String doInBackground(Void... params) {
        MultiValueMap<String, String> parameters = new LinkedMultiValueMap<>();
        Calendar cal = Calendar.getInstance();
        String hour=String.valueOf(cal.get(Calendar.HOUR_OF_DAY));
        String minute=String.valueOf(cal.get(Calendar.MINUTE));
        parameters.add("qr",qr_content);
        parameters.add("email",pref.getString("id",null));
        parameters.add("state",state);
        if(Integer.parseInt(minute)<10){
            parameters.add("time",hour+"0"+minute);
        }else if(Integer.parseInt(minute)==0){
            parameters.add("time",hour+"00");
        }else{
            parameters.add("time",hour+minute);
        }
        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);
        HttpEntity<MultiValueMap<String, String>> request = new
        HttpEntity<>(parameters, headers);
        RestTemplate restTemplate = new RestTemplate();
        List<HttpMessageConverter<?>> messageConverters = new
        ArrayList<HttpMessageConverter<?>>();
        messageConverters.add(new FormHttpMessageConverter());
        messageConverters.add(new StringHttpMessageConverter());
        restTemplate.setMessageConverters(messageConverters);
        String result = restTemplate.postForObject(url, parameters, String.class);
        return result;
    }
    @Override
    protected void onPostExecute(String s) {
        Toast.makeText(getApplicationContext(), "출결처리가 완료되었습니다.",
        Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(ReaderActivity.this,ReaderActivity.class);
        startActivity(intent);
        finish();
    }
}
}
}
}

```

PART 10

트러블 슈팅 보완사항

프로젝트에 있었던 문제점과 해결방안을 기록하였다.

■ 트러블 슈팅 보안사항

Date	Division	Issue	Trouble shooting
6 월 1 일	트러블	프로젝트 주제 설정	주제를 설정하는데 많은 시간을 낭비
6 월 2 일	이슈	부트스트랩 선정	프로젝트 주제와 알맞은 부트스트랩을 찾게 되면서 UI에 시간을 많이 투자하지 않게 되었다.
6 월 3 일	트러블	게시판 DB 설계	테이블을 빼 주는 등 최대한 역정규화가 되지 않게 해결
6 월 4 일	트러블	Git 사용의 어려움	써보지 않은 git으로 형상관리를 하려해서 어려움을 겪었지만 검색을 통하여 사용법을 익혀 해결
6 월 5 일	트러블	Spring에 대한 이해부족	spring에 대한 이해부족으로 프로젝트를 진행하기에 속도가 더뎠지만 배웠던 내용의 복습을 통해 해결
6 월 6 일	트러블	세팅에 시간 허비	프로젝트를 하기에 앞서 STS, DB 설정들이 필요했는데 서로 컴퓨터의 문제점이 있어 같은 버전의 프로그램과 틀로 설정하여 해결
6 월 7 일	트러블	부정확성	스토리보드와 다른 이해를 가지고 있어서 프로젝트를 진행하는데 속도가 늦어져 팀원들 간의 회의를 통해 해결
6 월 8 일	트러블	에뮬레이터에서만 실행되는 안드로이드	에뮬레이터에서만 실행되는 안드로이드로 인해 시간을 낭비, 새로운 안드로이드 기계를 구함으로 인해 해결
6 월 9 일	트러블	풀 캘린더 API	부트에 있던 풀 캘린더를 사용하는데 이미 설정 되어있는 js를 수정해서 사용하려 하니 동작되지 않아 어려움을 겪었지만
6 월 10 일	트러블	CSS 충돌	CSS가 충돌 나 버튼 등 UI가 깨져 div태그를 씌워 css를 나눔으로 인해 해결
6 월 11 일	트러블	데이터테이블 충돌	select된 데이터가 데이터테이블로 적용되지 않아 데이터테이블 js 경로를 직접 설정해주어 해결
6 월 12 일	트러블	소켓 데이터 출력	소켓에서 상대방의 아이디를 가져와 비교해서 메시지를 어떻게 보낼지 어려웠지만 세션 값으로 하나의 Map 안에 담아 비교하여 해결
6 월 13 일	트러블	QR 코드 zxing	API를 가져다 쓰는데 스프링 방식으로 어떻게 바꿀지, API 안에서 QR 코드의 사이즈와 값을 어떻게 담아 옮기는지 분석하는데 오래 걸렸지만 필요 없는 코드를 빼고 축소시켜 이해하며 해결
6 월 14 일	트러블	안드로이드	DB까지 저장하는데 있어서 개인서버를 쓰다 보니 많은 오류들이 발생, IP 주소와 스프링 방식을 통해서 해결
6 월 15 일	트러블	데이터 출력	mapper를 잘못설정해 에러가 나 에러를 고치는데 많은 시간을 낭비했지만 이후 비슷한 에러가 나면 바로바로 처리할 수 있었음
6 월 16 일	트러블	to do list	to do list css를 따와서 우리 부트에 맞게 적용 하려는데 충돌이 일어 남, class 이름을 바꿔주며 적용
6 월 17 일	트러블	리스트 출력이 되지 않음	parameter 이름을 잘 못 써줘서 parameter 값이 넘어가지 않아 제대로 이름을 설정하여 해결

6 월 18 일	트러블	시큐리티	시큐리티가 복잡해 설정하는데 많은 시간을 소유
6 월 19 일	트러블	모달 충돌	2 개의 모달에 같은 ID 값을 주어 모달이 충돌이 났지만 ID 값을 다르게 하여 해결
6 월 20 일	트러블	사진 출력	회원 사진이 원하는 모양으로 출력되지 않았지만 크기를 지정해 주어 해결
6 월 22 일	트러블	히스토리 출력	히스토리들이 한쪽으로는 출력되어 if 문을 써 양쪽으로 나뉘어 출력되게 하여 해결
6 월 23 일	트러블	ajax	ajax 를 사용하여 비동기로 데이터 값을 받아오는데 어려움을 겪었지만 검색을 통해 익혀 해결
6 월 24 일	트러블	select box	select 된 값들이 select box 로 나타나지 않았는데 충돌이 났던 부분을 수정함으로써 해결
6 월 25 일	트러블	게시판 로직	게시판이 서로 연동되어 있는 것들이 많아 로직을 짜는데 어려움을 겪었지만 팀원들 모두 의견을 나누며 효율적인 로직 생성
6 월 26 일	트러블	Git commit 실수	Commit 을 잘못하여 이전에 했던 내용들이 날라갔지만 백업해 뒀던 자료들이 있어 해결
6 월 27 일	트러블	쪽지 알람	쪽지를 보냈을 때 알람이 뜨지 않았는데 설정해 둔 IP 와 노트북 IP 가 맞지 않아 생긴 문제여서 IP 를 다시 설정해주며 해결
6 월 28 일	트러블	파일 업로드	파일 업로드 파일 업로드가 되지 않아 다른 사람들의 코드를 통해 수정해 나가며 해결
6 월 29 일	트러블	게시판 DB	cascade 를 잘못 설정하여 다시 설정하며 해결
6 월 30 일	트러블	타일즈	타일즈를 설정하는데 어려움을 겪었으나 경로 설정을 다시 해주며 해결
6 월 31 일	트러블	서비스	서비스로 따로 빼 주지 않아 후에 한번에 서비스단을 빼 줌
7 월 1 일	트러블	한글 깨짐	UTF-8 을 설정해주지 않아 한글이 깨졌지만 설정해줌으로써 해결
7 월 2 일	트러블	json	json 형태의 데이터를 ajax 로 뺄 때 데이터가 넘어가지 않았지만 ' ; ' 를 제거해주며 해결
7 월 3 일	트러블	버튼	버튼을 누를 시 리셋이 되지 않아 js 로 기능을 빼주며 해결
7 월 4 일	트러블	차트	js 파일로 차트가 빠져있어 js 를 가져와 코드를 가공하는데 어려움을 겪었지만 코드를 분석, 이해하며 해결
7 월 5 일	트러블		새로고침 했을 시 메인에 데이터 값이 출력되지 않았지만 경로를 설정해 주며 해결
7 월 6 일	트러블	Git 합침	git 을 합치는 와중에 서로 충돌이 일어나 다시 코드를 작성 해야 했음

PART 11

프로젝트 후기

프로젝트를 통한 느낀 점을 제시하였다.

PART 10. 프로젝트 후기

■ 프로젝트 후기

- 이현정

파이널 프로젝트의 주제선정부터 구현단계까지 정말 쉴 새 없이 달려오다 보니 어느새 약 한 달이라는 시간이 순식간에 지나간 거 같다. 이렇게까지 시간이 빨리 흘러가는 거 였나 하고 새삼 놀랐다. 이번 주제를 ERP로 선정하고, 그에 들어갈 여러가지 기능과 기술을 생각하며 팀원들과 설렘을 갖고 프로젝트를 진행했다.

생각했던 만큼 진도를 못 빼는 괴로움 이라든가 진행하면서 에러를 잡느라 많은 시간을 허비 했다 거나 하는 이슈가 있었지만, 그래도 스케줄에 맞춰 착착 진행 할 수 있었다. 하지만, 도중에 크게 계획이 틀어질 만한 트러블이 있었고, 그것은 프로젝트 마감 1주일을 남긴 시점이었다. 마지막 1주일은 거의 먹고 자는 시간만 빼면 계속 코딩만 한 것 같다. 특히 마지막 3일밤은 하나의 보탬도 없이 팀원들을 1.2시간만 재우고 합숙을

- 최은혜

이번 파이널 프로젝트를 하면서 정말 많은 것을 배우고 많은 것을 경험 할 수 있었던 것 같고 다들 너무 수고하는 모습을 보여주어서 너무 고맙고 함께 밤을 샜던 마지막 3박4일은 나중에 추억으로 남아 두고두고 기억 날 것 같다. 6개월 동안 쉴 없이 달리면서 힘든 점도 있었고 어려운 점도 있었지만 같은 조원 분들과 해결 해 나아가는 방법을 함께 배워가면서 잘 이겨낼 수 있었던 것 같다. 짧지 않은 시간 동안에 많은 것들을 배우는 일이 힘들다고 느껴질 수도 있겠지만 저에게는 어떠한 순간보다도 가치 있는 경험이 된 것 같아서 뿌듯하게 생각 된다.

- 노지영

프로젝트를 시작한지 어느덧 6개월이 지나 벌써 수료할 시간이 왔는데 저에게는 정말 뜻 깊은 일들이 참 많이 있었던 6개월이 된 것 같다. 많은 사람들과 협업하는 것에 대한 이해도가 높아졌고 프로젝트를 할 때 부딪히게 되는 문제들에 대한 해결 능력을 키울 수 있는 기회가 된 것 같다. 마지막에 파이널을 했던 조원들을 너무 좋은 분들을 만나게 되어서 몸이 힘들어도 즐겁게 임할 수 있었다. 코스타에서 교육받기 전에는 자바나 웹프로그래밍은 전혀 몰랐는데 정말 많은 것들을 배우고 가서 나중에 어떤 기술을 배워도 잘 배워나갈 수 있는 밑바탕을 잘 그려놓은 것 같은 생각이 든다. 함께 했던 모든 분들 너무 고생 하시고 묵묵히 열심히 해 주었던 분들 모두 잘 돼서 현업에서 만날 수 있었으면 좋겠다. 감사합니다.

- 박소현

처음에 코스타에 왔을 때에는 설레기도 하고 걱정도 많이 되었는데 언니 오빠들 친구들 강사님 과 함께 지내면서 너무 즐겁게 잘 지내고 수료 하는 것 같다. 특히 친구들과는 많이 친해져서 적응을 더 빨리 할 수 있었던 것 같다. 제가 파이널 프로젝트를 할 때 조에서 막내였는데 언니 오빠들이 저를 잘 챙겨주셔서 정말 즐겁게 생활 했던 것 같다. 매일 같이 코딩을 하면서 처음에는 어려웠던 문제들도 점점 해결해 나갈 수 있는 방향성이 생기기도 앞으로도 내가 무엇을 잘 할 수 있는지 깊게 생각해 볼 수 있는 시간이 된 것 같다.

- 김은영

저는 원래 전공자도 아니고 문과 출신 이여서 코딩을 배우는데 처음에는 많이 생소하고 어려웠는데 점차 시간이 지나고 잘 하는 분들과 함께 배우다 보니까 도움도 받게 되고 내가 가지고 있는 강점도 발견하게 될 수 있었던 거 같다. 조원들이 작은 일을 해내어도 칭찬을 해주고 사기를 불러 일으켜 주셔서 더 의욕을 가지고 많은 일들을 할 수 있었던 것 같다. 앞으로도 코스타에서 배운 것들을 바탕으로 많은 것에 도전 할 수 있는 사람이 될 수 있을 것 같아서 기분 좋게 마무리를 하게 되는 것 같다. 프로젝트 하면서 좋은 언니, 오빠, 동생들을 만났고 또 많은 것들을 배우고 코딩 실력이 늘게 되어 좋았다. 막바지에 모두 함께 한 3박4일의 합숙은 수면부족으로 괴롭지만 서도 잊지 못 할 재미있고 행복한 경험 이었다.

- 위진학

1번째 2번째 조 에서 부터 계속 배워왔고 중간중간 작은 프로젝트를 하면서 많은 시행착오를 겪었었다. 마지막 프로젝트는 이를 바탕으로 만들고 싶었다. 아무리 이런 생각을 혼자 한들 프로젝트는 혼자 할 수 없는 것이기에 조원 각자의 힘이 크다는 것을 많이 느껴볼 수 있는 코스타에서 마지막 조이자 프로젝트였다. 열심히 하면 결과가 어떠한 아쉬움이 남는 건 맞지만 이 아쉬움을 가지고 더 나은 프로젝트를 수행할 수 있는 계기.동기 부여가 될 수 있을 것 같다. 마지막 후기에서 팀원들 모두에게 감사하고 고맙다는 말을 전하고 싶다.

- 최석환

간단해 보이는 기능에도 많은 로직과 노력이 들어간다는 것을 알았다. 앞으로는 겉으로 보여지는 규모만 보고 쉽게 판단하는게 아니라 세세한 부분까지 염두하고 설계 해야 겠다. 스프링 프레임워크를 통한 프로젝트를 통해 많은 공부가 되었고, 특히 시나리오를 보고 로직을 설계하는 부분이 향상된 것 같다.

PART 12

참조 문헌

프로젝트를 진행하면 참고하였던 사이트들을 제시하였다.

PART 12. 참조 문헌

■ 참조 문헌

- 해피아카데미

http://www.happyaca.com/manager_web

- ACADAY

https://demo.acaday.com/atlantis_demo/main/today

- ipad 어플 (교실)

<https://itunes.apple.com/kr/app/%EA%B5%90%EC%8B%A4/id1085319084?mt=8>

- 교실관리 솔루션 모음

<https://websetnet.com/ko/8-classroom-management-apps-for-tech-savvy-teachers/>

- 쪽지기능 벤치마킹

<https://www.inflearn.com/>

- 이해도 체크 벤치마킹 어플

<https://www.plickers.com/>

본 자료는  국가인적자원개발컨소시엄 2014년도 채용예정자 학습 산출물 자료입니다.



한국소프트웨어기술진흥협회

KOSTA Korea Software Technology Association

경기도 성남시 분당구 삼평동 대왕판교로 670길
유스페이스2 B동 8층

Tel: 070-7090-4972~5 / Fax: 02-408-2599

웹사이트: <http://edu.kosta.or.kr>