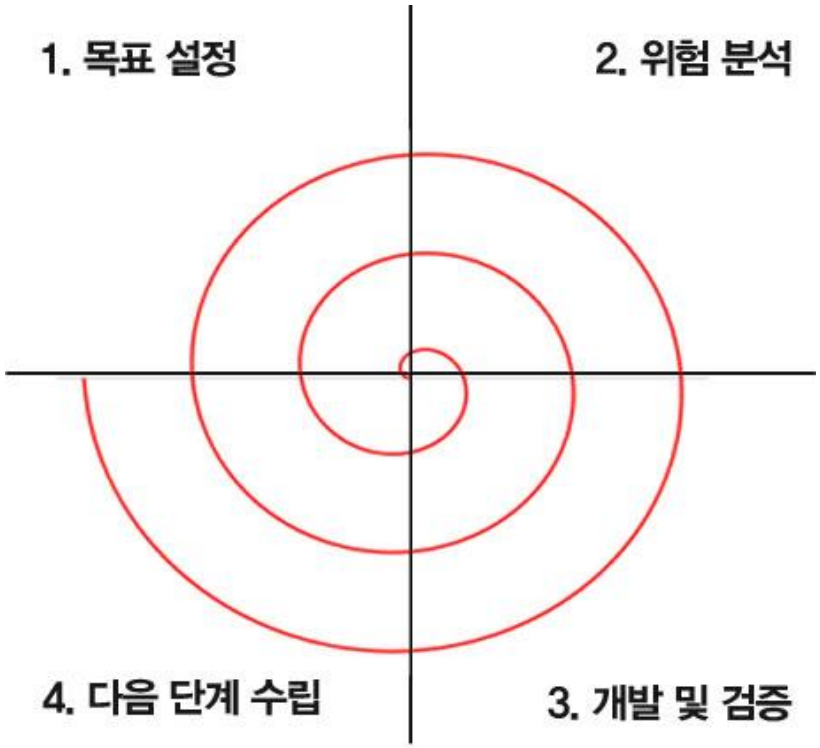
# LMS프로젝트 – 비트캠프 구리점

| | |
|---|---|
| **Write Dates** | 2020-02-04~2020-02-05 |
| **Writer** | 김덕수 |
| **Member** | 고재현, 김덕수, 장인영 |

비트캠프

# 0. 개발 모델 채택 – 나선형 모델




## 작업 순서

- 메뉴 설계 / 클래스 설계
- 스토리보드 설계 / DB 설계

↓

- 메인페이지 프론트엔드 개발 / 메인페이지 백엔드 개발

↓

- 학생 로그인 프론트엔드 개발 / 학생 로그인 백엔드 개발

↓

- LMS 부서별 프론트엔드 개발 / LMS 부서별 백엔드 개발

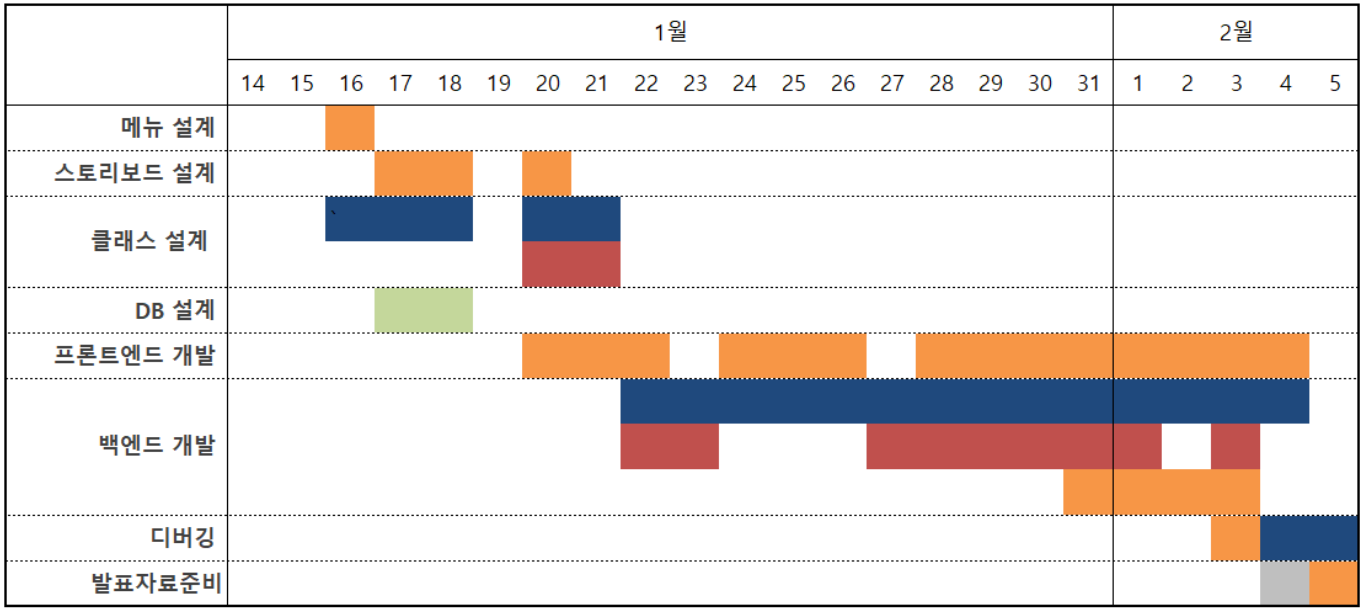
↓

- 프론트엔드 디버깅 / 백엔드 디버깅

# 1. 일정

김덕수 | 고재현 | 장인영 | 유명효 | 전 체

수정일정은 표시 X




| | |
|---|---|
| 고재현 | 클래스 설계 / DB 설계 / 백엔드 개발 |
| 김덕수 | 메뉴 설계 / 스토리보드 설계 / 프론트엔드 개발 / 백엔드 개발 |
| 장인영 | 클래스 설계 / 백엔드 개발 |
| 유명효 | DB 설계 |

**설계기간 총 5일**

- 메뉴 설계 1일
- 스토리보드 설계 3일
- 클래스 설계 5일
- DB 설계 2일

**개발기간 총 14일**

- 프론트엔드 개발 14일
- 백엔드 개발 14일
- 디버깅 2일

# 2_1. 프론트엔드 – 메뉴설계

# 2_2_1. 프론트엔드 - 스토리보드 - 메인

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 3 |
|---|---|---|---|---|---|---|---|---|---|

## 스토리보드

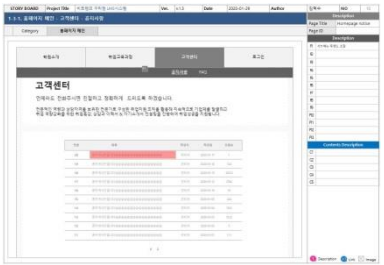
### 1. 홈페이지 메인

1-1. 인덱스

1-2. 학원소개

1-3. 취업교육소개

1-4. 고객센터



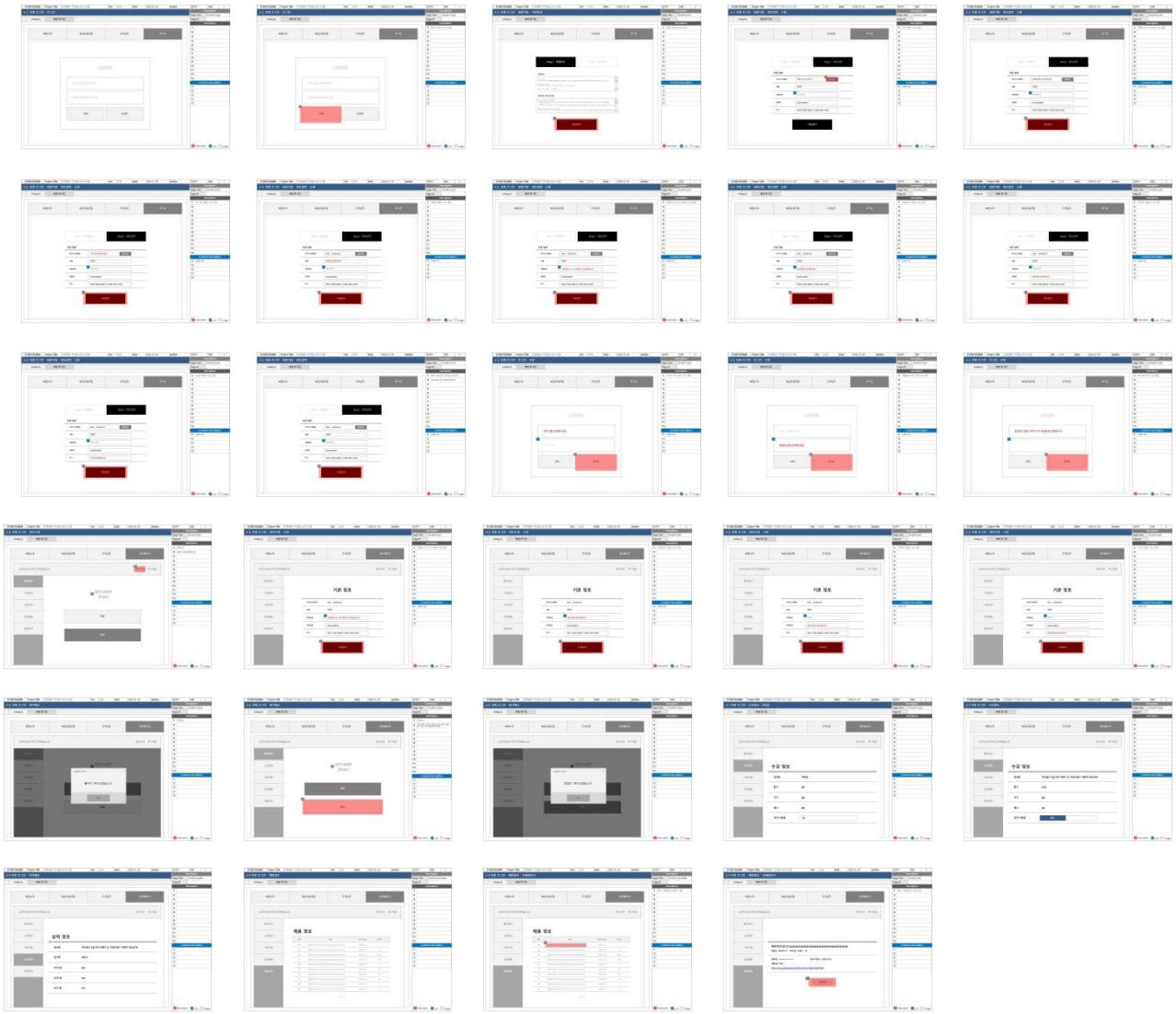
# 2_2_2. 프론트엔드 – 스토리보드 – 학생 로그인

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김목수 | 13 |
|---|---|---|---|---|---|---|---|---|---|

## 스토리보드

### 2. 학생 로그인

2-1. 로그인
2-2. 회원가입
2-3. 대시보드
2-4. 정보수정
2-5. 로그아웃
2-6. 출석체크
2-7. 수강정보
2-8. 시험조회
2-9. 성적열람
2-10. 채용정보



# 2_2_3. 프론트엔드 – 스토리보드 – LMS 로그인

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 53 |
|---|---|---|---|---|---|---|---|---|---|

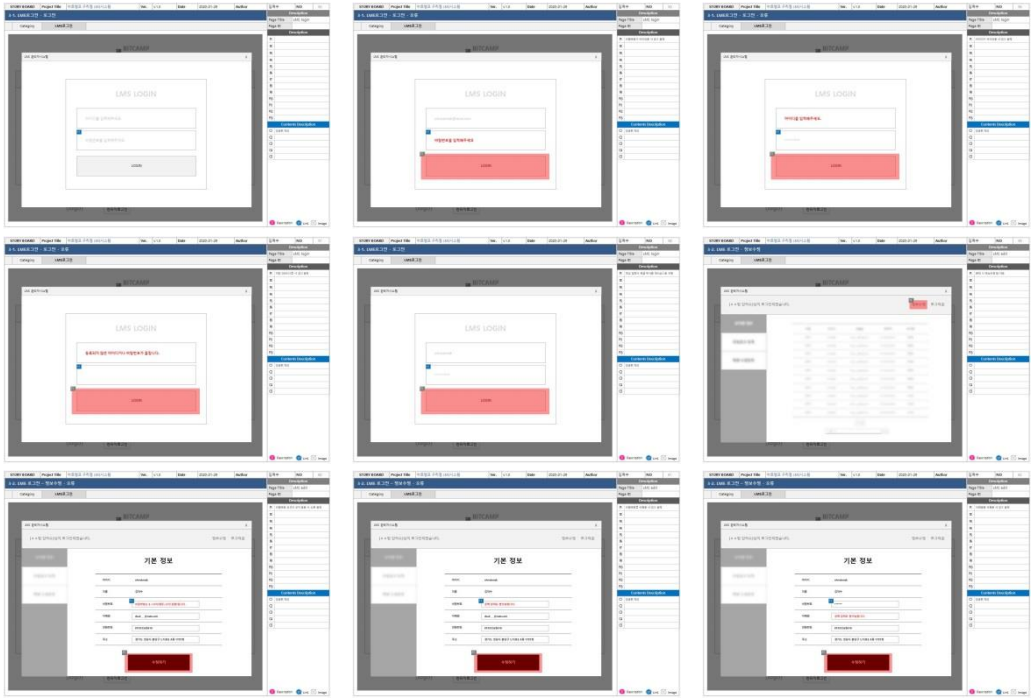
## 스토리보드

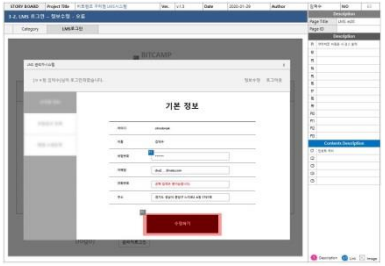
### 3. LMS 로그인

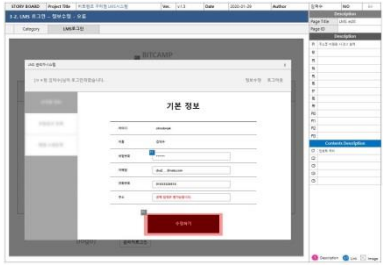
3-1. 일반 로그인

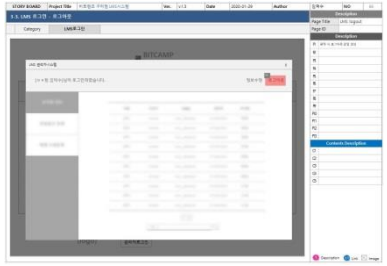
3-2. 정보수정

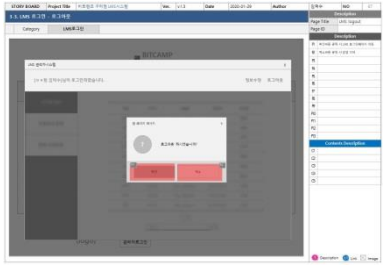
3-3. 로그아웃

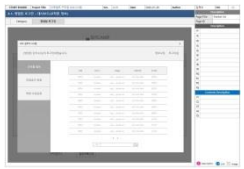


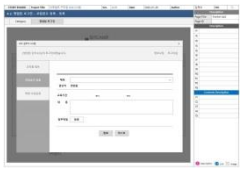


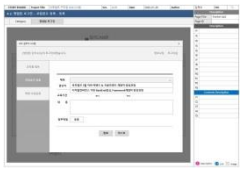


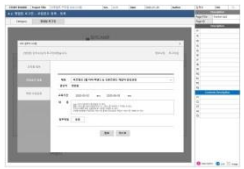


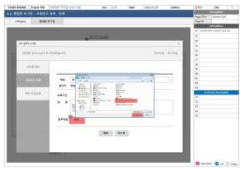


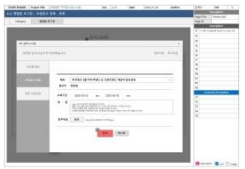


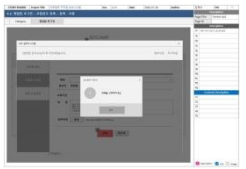


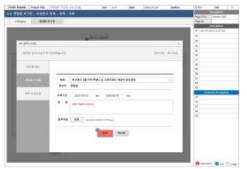


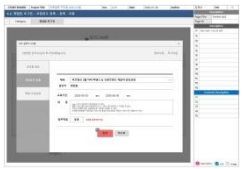


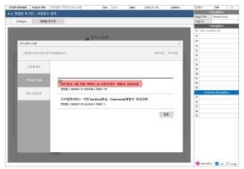


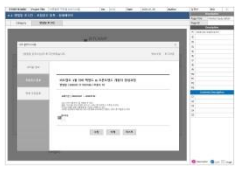


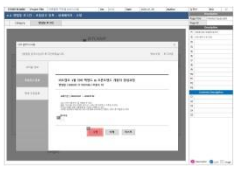


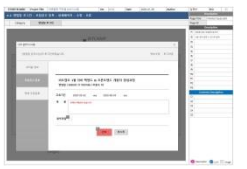


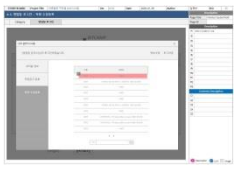


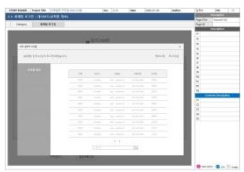












# 2_2_4. 프론트엔드 – 스토리보드 – 영업팀 로그인 & 회계팀 로그인

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 68 |
|---|---|---|---|---|---|---|---|---|---|

## 스토리보드

### 4. 영업팀 로그인

4-1. 대시보드(교직원 정보)
4-2. 모집공고 등록
4-3. 학생 수강등록

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 94 |
|---|---|---|---|---|---|---|---|---|---|

## 스토리보드

### 5. 회계팀 로그인

5-1. 대시보드(교직원 정보)

# 2_2_5. 프론트엔드 – 스토리보드 – 행정팀 로그인

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 96 |
|---|---|---|---|---|---|---|---|---|---|

## 스토리보드

### 6. 행정팀 로그인

6-1. 대시보드(교직원 관리)
6-2. 강좌 개설
6-3. 수강생 관리
6-4. 공지사항 관리
6-5. FAQ 관리

# 2_2_6. 프론트엔드 – 스토리보드 – 취업팀 로그인

| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 165 |
|---|---|---|---|---|---|---|---|---|---|

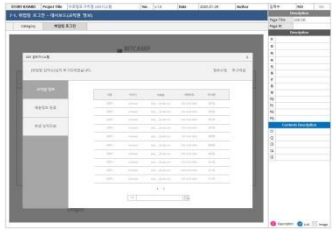
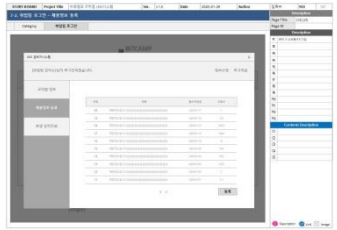
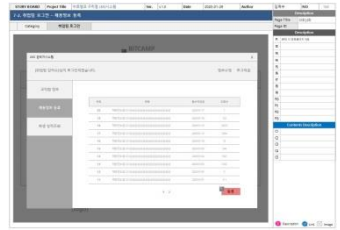
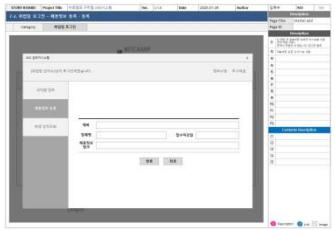
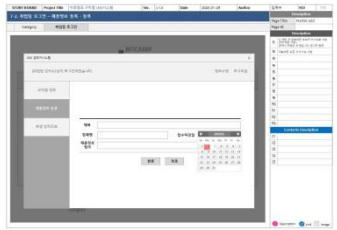
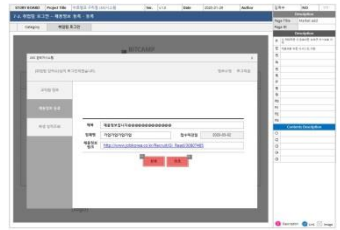
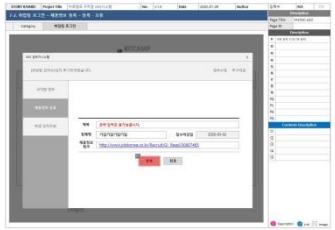
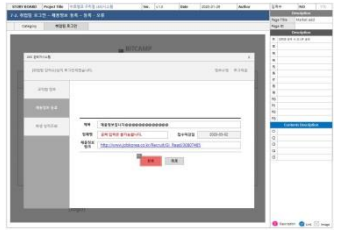
## 스토리보드

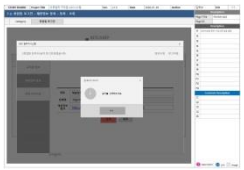
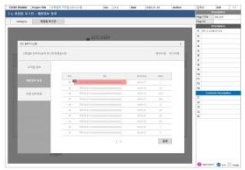
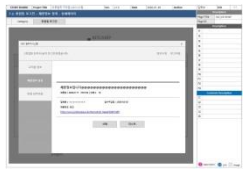
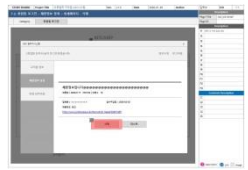
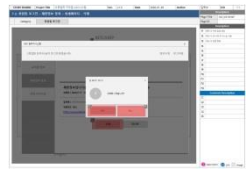
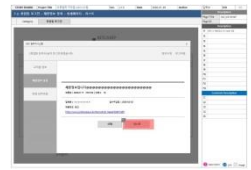
### 7. 취업팀 로그인

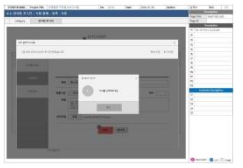
7-1. 대시보드(교직원 정보)

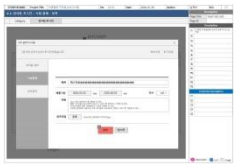
7-2. 채용정보 등록

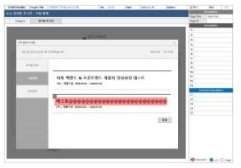
7-3. 학생 성적조회

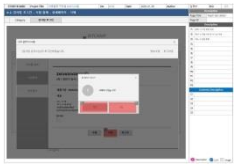









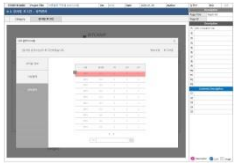

















# 2_2_7. 프론트엔드 – 스토리보드 – 강사팀 로그인

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STORY BOARD | Project Title | 비트캠프 구리점 LMS시스템 | Ver. | v.1.3 | Date | 2020-01-29 | Author | 김덕수 | 183 |

## 스토리보드

### 8. 강사팀 로그인

8-1. 대시보드(교직원 정보)
8-2. 시험출제
8-3. 성적관리

# 2_3_1. 프론트엔드 – 주요 기능 – 회원가입 시 유효성 검사

```
$('.checkId').on("click",function(){
    var id=$('#id').val();
    var id=$.trim(id);
    var regex=/^[-A-Za-z0-9_]+[-A-Za-z0-9_.]*[@]{1}[-A-Za-z0-9_]+[-A-Za-z0-9_.]*[.]{1}[A-Za-z]{1,5}$/;
    if(regex.test(id) && id.length>10){
         $.post('memberduplicate.html','id='+id, function(data){
        if(data=='fail'){
            alert('중복 처리 에러 다시 접속하시길 바랍니다.');
        }else{
            if(data==0){
               alert('사용 가능한 아이디 입니다.');
               $('.checkId').prop("disabled", true).css("cursor", "default");
               $('#id').attr("readonly",true);
               $('#id').css("background-color","silver");

               var pw=document.getElementById('pw').value;
               var pw2=document.getElementById('pw2').value;
               if(pw == pw2 && pw !="" && pw2 !=""){
                  $('.btn3').prop("disabled", false);
                  $('.btn3').css({
                     'background-color': '#313436', 'color': 'white', 'cursor': 'pointer'
                        });
                  }
               }else if(data==1){
                  $('#id').val('');
                  $('#id').addClass("error");
                  $('#id').attr('placeholder', '중복된 아이디입니다.');
               }
            }
         });
        }else{
            $('#id').val('');
            $('#id').addClass("error");
            $('#id').attr('placeholder', '형식이 맞지 않습니다.');
        }
    });
```

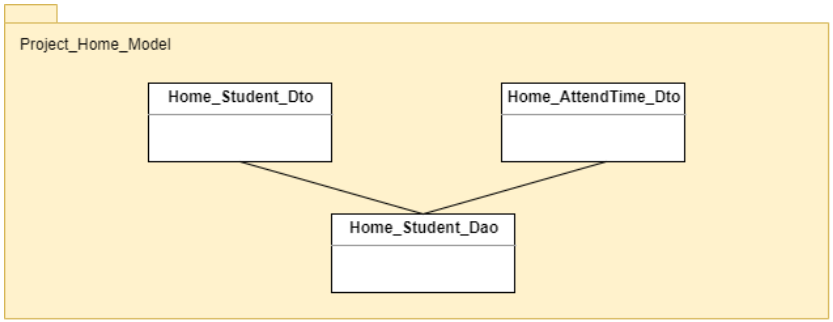
```
if(!/^[a-zA-Z0-9]{8,13}$/.test(pw)){
    $('#pw').val('');
    $('#pw2').val('');
    $('#pw').addClass("error");
    $('#pw').attr('placeholder', '8~13자리로 입력해주세요.');
    return false;
}
var chk_num = pw.search(/[0-9]/g);
var chk_eng = pw.search(/[a-z]/ig);
if(chk_num < 0 || chk_eng < 0){
    $('#pw').val('');
    $('#pw2').val('');
    $('#pw').addClass("error");
    $('#pw').attr('placeholder', '숫자와 영문을 혼용해야합니다.');
    return false;
}
```

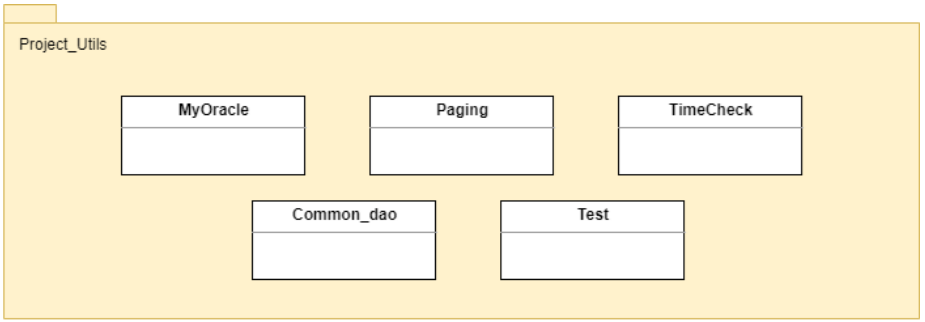
# 2_3_2. 프론트엔드 – 주요 기능 – CSS, 자바스크립트 분할 관리

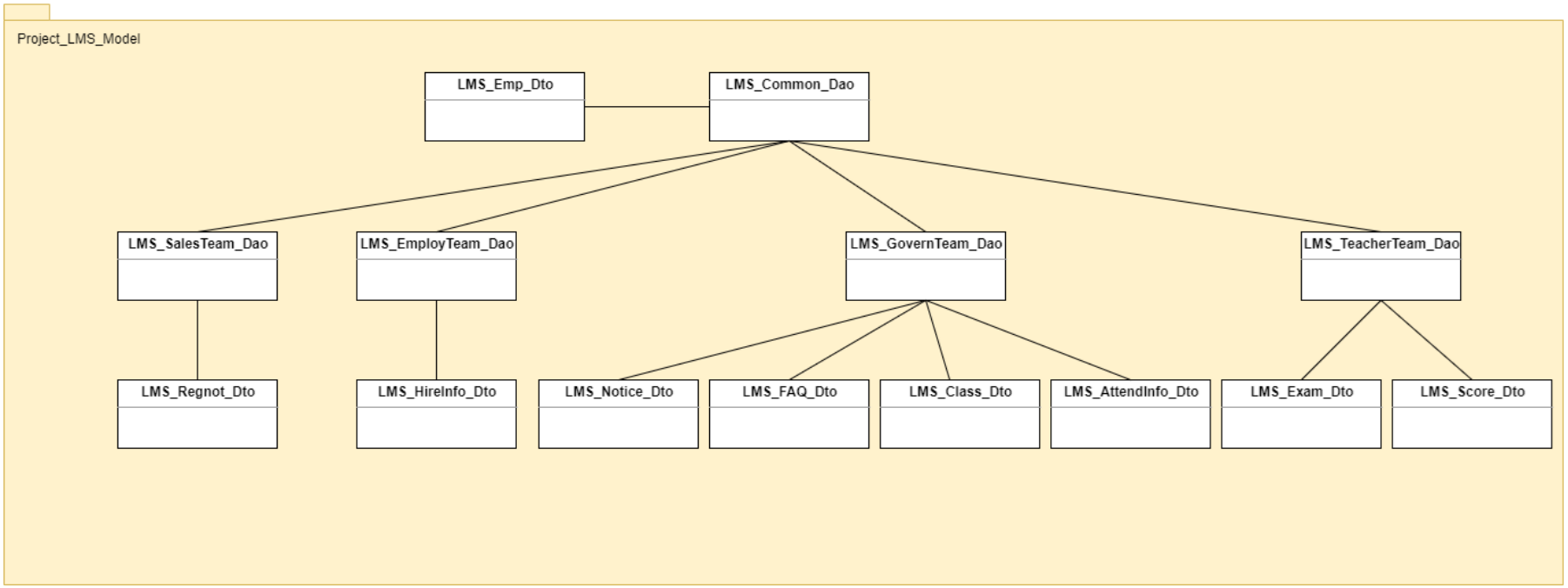
```
<link rel="stylesheet" href="../css/animate.css" />                    <!--    외부 애니메이션        -->
<link rel="stylesheet" href="../css/reset.css" />                      <!--    초기 css설정 초기화    -->
<link rel="stylesheet" href="../css/index.css" />                      <!--    메인 홈페이지&학생     -->
<link rel="stylesheet" href="../css/lms.css" />                        <!--    LMS 로그인&회계팀      -->
<link rel="stylesheet" href="../css/lms/employ.css" />                 <!--    취업팀 LMS             -->
<link rel="stylesheet" href="../css/lms/govern.css" />                 <!--    행정팀 LMS             -->
<link rel="stylesheet" href="../css/lms/sales.css" />                  <!--    영업팀 LMS             -->
<link rel="stylesheet" href="../css/lms/teacher.css" />                <!--    강사팀 LMS             -->

<script type="text/javascript" src="../js/jquery-1.12.4.js"></script>  <!--    제이쿼리               -->
<script type="text/javascript" src="../js/jquery-ui.js"></script>      <!--    제이쿼리 기본 UI       -->
<script type="text/javascript" src="../js/jquery.bxslider.js"></script><!--    메인 슬라이더          -->
<script type="text/javascript" src="../js/wow.js"></script>            <!--    외부 애니메이션        -->
```

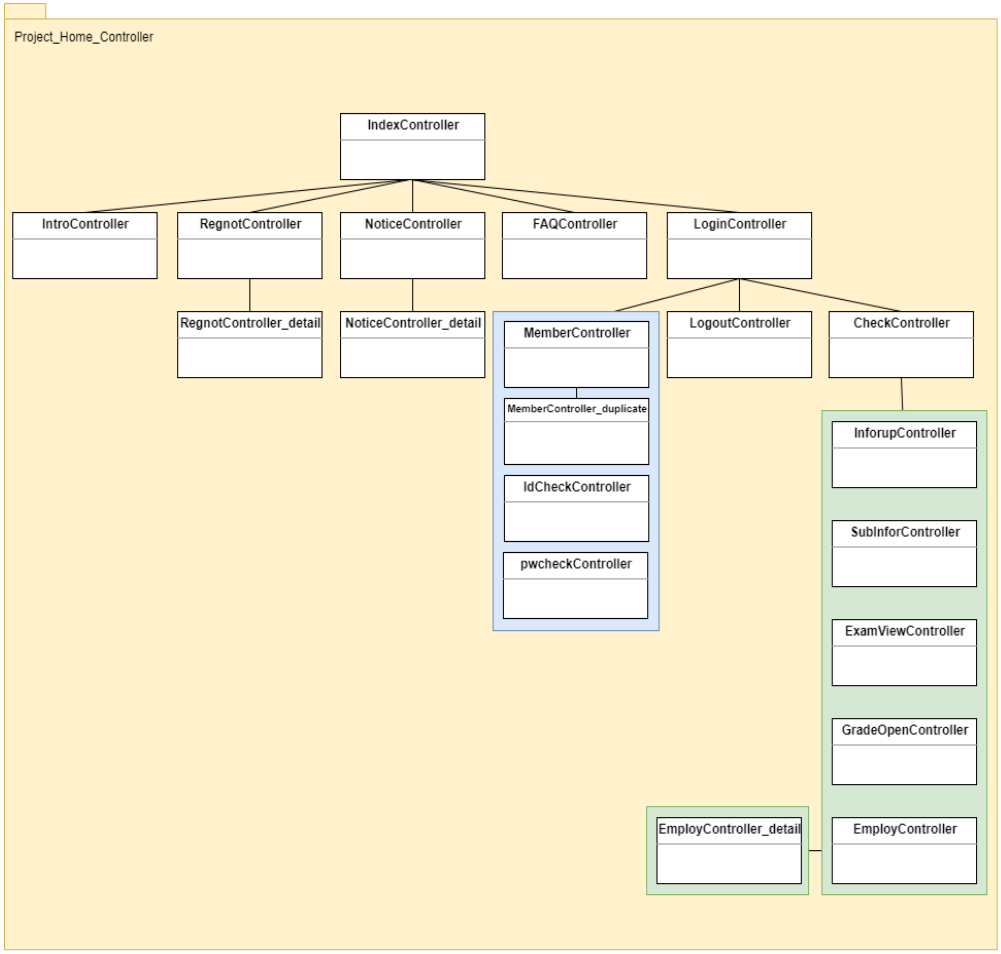
# 3_1_1. 백엔드 – 클래스 설계 – 모델 설계

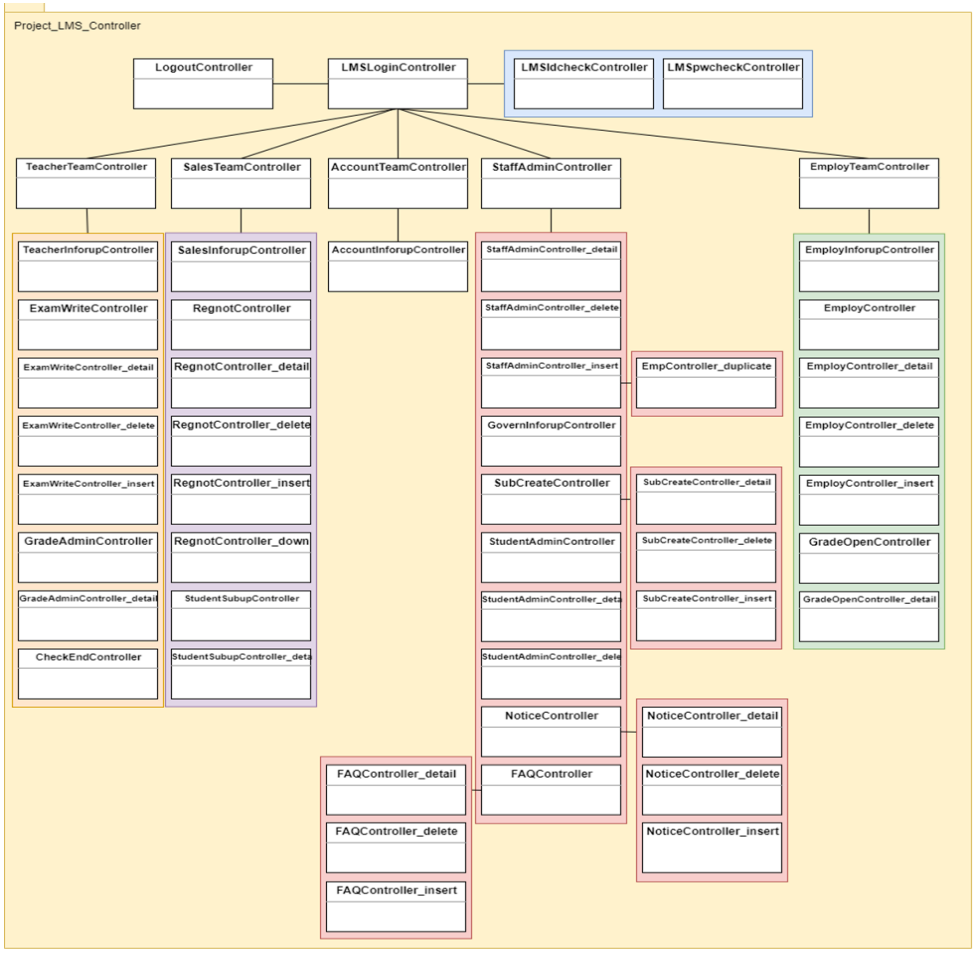


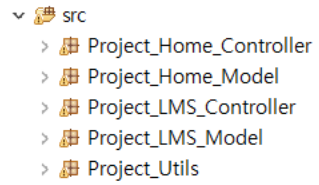



# 3_1_2. 백엔드 – 클래스 설계 – 컨트롤러 설계



# 3_1_3. 백엔드 – 클래스 설계 – 패키지 설계




**HOME컨트롤러**

-고객과 학생이 사용하는 모든 컨트롤러
-공통 (모집공고조회, 공지사항조회,
FAQ조회, 인트로, 인덱스, 로그인, 로그아웃)
-학생 (출석체크, 시험조회, 성적열람, 취업공고조회, 수강정보)

**LMS컨트롤러**

-직원이 사용하는 모든 컨트롤러
-공통 (직원정보출력, 로그인, 로그아웃)
-행정 (교직원관리, 강좌개설, 수강생관리, FAQ, 공지사항)
-취업 (채용정보등록, 학생성적조회)
-영업 (모집공고등록, 학생수강등록)
-강사 (시험등록, 성적관리, 출석마감)

**HOME모델**

-고객과 학생이 사용하는 모든 정보처리 클래스
-HOME_DAO (홈 화면, 학생 화면에서 처리되는 모든 정보출력 담당)
-STUDENT_DTO (학생정보를 담고있는 클래스)
-ATTENDTIME-DTO (학생의 입실,퇴실시간을 담고있는 클래스)

**LMS모델**

-직원이 사용하는 모든 정보처리 클래스
-GOVERNTEAM_DAO (행정팀 전용으로 사용하는 정보처리 클래스)
-TEACHERTEAM_DAO (강사팀 전용으로 사용하는 정보처리 클래스)
-SALESTEAM_DAO (영업팀 전용으로 사용하는 정보처리 클래스)
-EMPLOYTEAM_DAO (취업팀 전용으로 사용하는 정보처리 클래스)
-CLASS_DTO (강좌정보를 담고있는 클래스)
-REGNOT_DTO (모집공고 정보를 담고있는 클래스)
-EXAM_DTO (시험정보를 담고있는 클래스)
-ATTENDINFO_DTO (학생출결정보를 담고있는 클래스)
-EMP_DTO (교직원 정보를 담고있는 클래스)
-HIREINFO_DTO (취업공고 정보를 담고있는 클래스)
-NOTICE_DTO (공지사항 정보를 담고있는 클래스)
-FAQ_DTO (FAQ정보를 담고있는 클래스)
-SCORE_DTO (강좌점수를 담고있는 클래스)

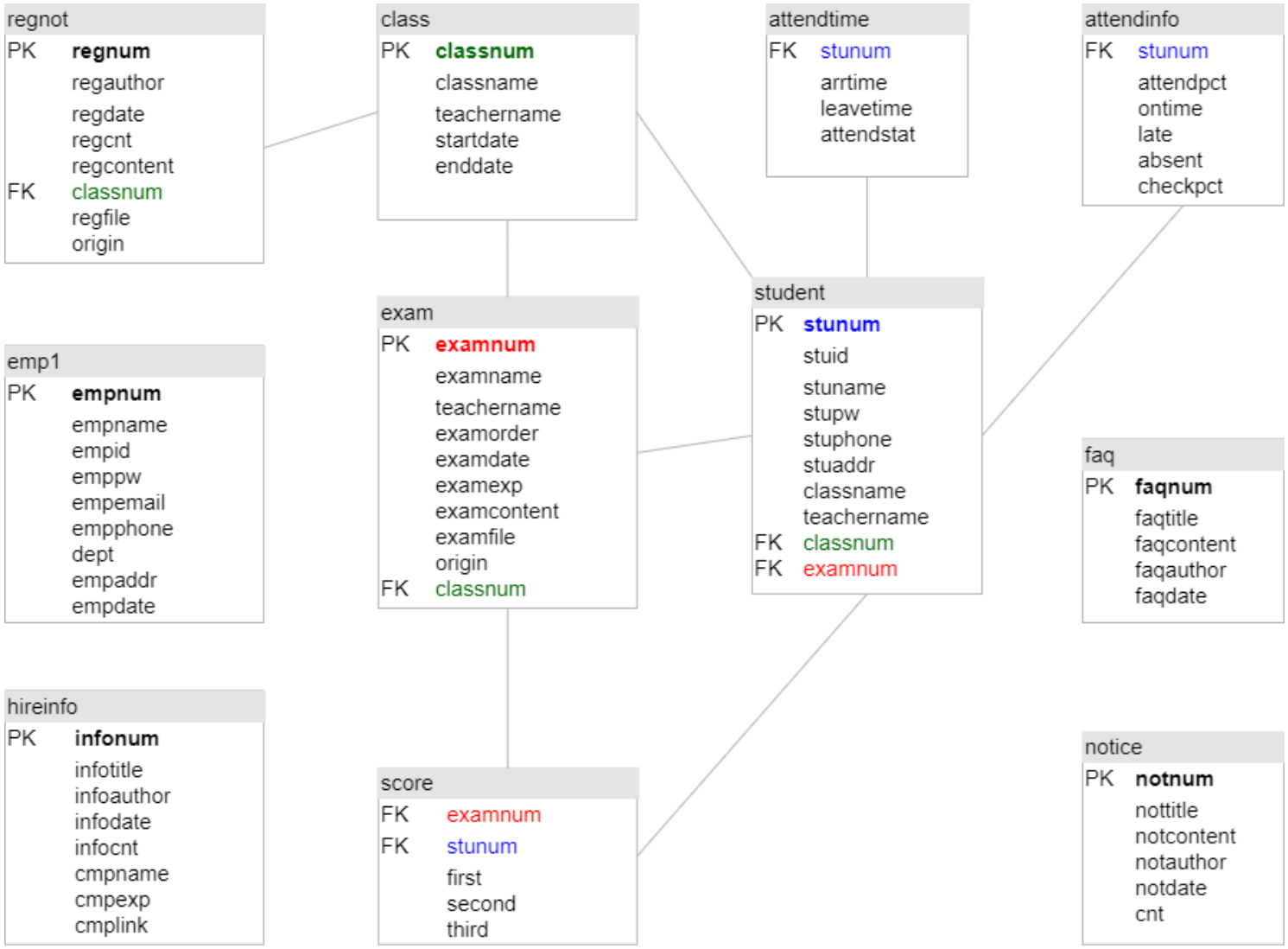
**UTILS**

-전체가 공통으로 사용하는 정보처리 클래스
-MYORACLE (DB와 연동해주는 작업을 처리)
-TIMECHECK (출석정보와 날짜관련된 데이터를 처리)
-PAGING (게시판의 페이지를 처리)
-COMMON_DAO (학생,직원 둘다 사용되는 작업을 처리)

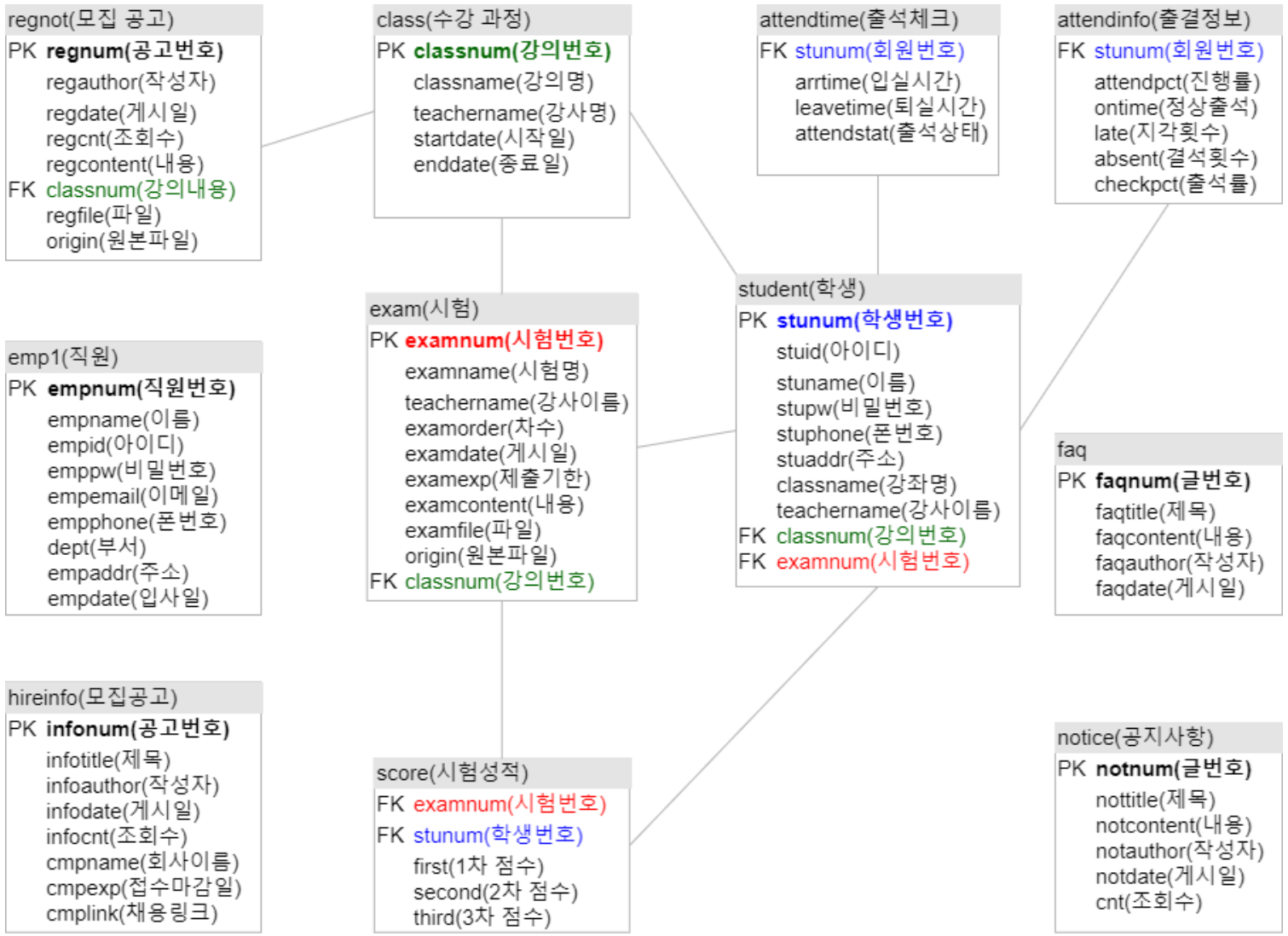
# 3_2. 백엔드 – 뷰 설계


index.jsp
Home
학원소개
intro.jsp
취업 교육과정
regnot.jsp
regnot_detail.jsp
공지사항
notice.jsp
notice_detail.jsp
FAQ
faq.jsp
로그인
login.jsp
회원가입
member.jsp
아이디 찾기
idcheck.jsp
비밀번호 찾기
pwcheck.jsp
MyPage
출석체크
check.jsp
수강정보
subInfor.jsp
시험조회
examView.jsp
성적열람
gradeOpen.jsp
채용정보
employ.jsp
employ_detail.jsp
정보수정
inforup.jsp
LMS
강사팀(Teacher)
교직원 정보
teacherTeam.jsp
시험출제
examWrite.jsp
examWrite_detail.jsp
examWrite_insert.jsp
성적관리
gradeAdmin.jsp
gradeAdmin_detail.jsp
정보수정
inforup.jsp
영업팀(Sales)
교직원 정보
salesTeam.jsp
모집공고
regnot.jsp
regnot_detail.jsp
regnot_insert.jsp
학생 수강등록
studentSubUp.jsp
studentSubUp_detail.jsp
정보수정
inforup.jsp
회계팀(Account)
교직원 정보
accountTeam.jsp
정보수정
inforup.jsp
lmsLogin.jsp
아이디 찾기
idcheck.jsp
비밀번호 찾기
pwcheck.jsp
행정팀(Govern)
교직원 관리
staffAdmin.jsp
staffAdmin_detail.jsp
staffAdmin_insert.jsp
강좌개설
subCreate.jsp
subCreate_detail.jsp
subCreate_insert.jsp
수강생 관리
studentAdmin.jsp
gradeAdmin_detail.jsp
공지사항 관리
notice.jsp
notice_detail.jsp
notice_insert.jsp
FAQ 관리
faq.jsp
faq_detail.jsp
faq_insert.jsp
정보수정
inforup.jsp
취업팀(Employ)
교직원 정보
employTeam.jsp
채용정보 등록
employ.jsp
employ_detail.jsp
employ_insert.jsp
학생 성적조회
gradeOpen.jsp
gradeOpen_detail.jsp
정보수정
inforup.jsp

# 3_3. 백엔드 – DB 설계(영문)



# 3_3. 백엔드 – DB 설계(한글)



# 3_4_1. 백엔드 – 주요 기능

**학생 로그인**

회원가입

회원가입 시 학생정보 로우, 학생출석정보 로우, 학생출결정보 로우, 학생성적 로우 생성

---

**LMS 행정팀**

강좌 개설

현재 등록되어있는 강사팀 목록을 선택해 등록하고 강좌 로우, 강좌시험 로우 생성
수정 처리시 강좌에 묶여있는 학생정보(강사명, 강좌명), 강좌시험정보(강사명), 강좌정보(강사명, 강좌명)이 수정됨
삭제 처리시 강좌에 묶여있는 정보 순서대로 리셋, 삭제

1. [삭제] 학생성적 테이블
2. [삭제] 학생출석정보 테이블
3. [삭제] 학생출결정보 테이블
4. [삭제] 학생정보 테이블

# 3_4_1. 백엔드 – 주요 기능

**LMS 행정팀**

수강생 관리

현재 등록되어 있는 학생정보를 순서대로 지움

1. [삭제] 학생성적 로우
2. [삭제] 학생출석정보 로우
3. [삭제] 학생출결정보 로우
4. [삭제] 학생정보 로우

---

**LMS 영업팀**

학생 수강등록

수정 처리시 학생정보(강좌명, 강좌번호, 강사명, 시험번호)가 수정되고,
학생과 엮여있는 학생성적 테이블(시험번호,1차,2차,3차)이 리셋됨

[리셋] 학생성적 테이블 (시험번호=수강선택된 시험번호, 1차시험=0, 2차시험=0, 3차시험=0)

# 3_4_2. 백엔드 – 주요 기능 – 학생 출석체크(CONTROLLER)

```
//파라미터 처리
String data=req.getParameter("data");
System.out.println("넘어온 파라미터 : "+data);

String[] check1=data.split(",");
System.out.println("파라미터 스플릿 : "+check1[0]+" and "+check1[1]);

String check=check1[1].trim();
System.out.println("check 상태 : "+check);
int stunum=Integer.parseInt(check1[0]);
System.out.println("학번 : "+stunum);


//현재시간 뽑는곳
TimeCheck tc=new TimeCheck();
String time=tc.getTime();
System.out.println(time);

//출결상태 체크하는곳
int attendstat=0;
attendstat=tc.attendTimeReturn();

Home_Student_Dao dao=new Home_Student_Dao();
Home_AttendTime_Dto bean=null;
PrintWriter out=resp.getWriter();
HttpSession session=req.getSession();
```

```
//입실처리
if(check.equals("arrive")){
    try {
        bean=dao.checkInsertArrive(stunum,time,attendstat);
        session.setAttribute("check", bean);
        System.out.println("입실완료");
    } catch (SQLException e) {
        out.print("fail");
        out.flush();
        out.close();
    }
//퇴실처리
}else if(check.equals("leave")){
    try {
        bean=dao.checkInsertLeave(stunum, time);
        session.setAttribute("check", bean);

        System.out.println("퇴실완료");
    } catch (SQLException e) {
        out.print("fail");
        out.flush();
        out.close();
    }
}
out.print("ok");
out.flush();
out.close();
```

# 3_4_3_1. 백엔드 – 주요 기능 – 강사 출석마감(CONTROLLER)

```
System.out.println("컨트롤러 접근");
Common_dao dao=new Common_dao();
TimeCheck check=new TimeCheck();
ArrayList<Object[]> list=new ArrayList<Object[]>();
int diffDays=0;
System.out.println("객체생성 성공");

//파라미터 받아오는곳
String teachername=req.getParameter("name");
String startdate=req.getParameter("start");
String enddate=req.getParameter("end");
System.out.println("받아온 파라미터값 ? "+teachername+","+startdate+","+enddate);

//강좌진행일 받아서 그에맞는 총 강좌일 계산하는곳
diffDays=check.subAttendTime(startdate,enddate);


//학생들의 각각의 출결상태를 리턴받는곳 -list타입에
//LMS_AttendInfo_Dto와 Home_AttendTime_Dto객체 내장
try {
    list=dao.attendTimeSelectAll(teachername);
    System.out.println("리스트값 받아오기 성공");
} catch (SQLException e) {
    System.out.print("출석마감 처리-학생 출결상태리턴 에러");
    resp.sendRedirect("../lmsLogin.html");
    req.getSession().setAttribute("lmslogin", null);
    return;
}
```

```
//각 강사팀의 학생들 명수 구하는곳
int tableRow=0;
try {
    tableRow=dao.tableRowCheckReturn(teachername);
    System.out.println("테이블 갯수 받아오기 성공");
} catch (SQLException e) {
    System.out.println("출석마감 처리-테이블갯수 에러");
    resp.sendRedirect("../lmsLogin.html");
    req.getSession().setAttribute("lmslogin", null);
    return;
}

int[] result1=new int[tableRow];
int[] result2=new int[tableRow];
```

# 3_4_3_2. 백엔드 – 주요 기능 – 강사 출석마감(CONTROLLER)

```
//모든 학생정보 출석처리
PrintWriter out=resp.getWriter();
try{
    for(int i=0;i<tableRow;i++){
        Object[] obj=list.get(i);
        System.out.println("반복문 접근 성공");

        //인트배열[attendinfo의 정보를 attendtime에있는 attendstat기준으로 계산]
        int[] checking=check.attendInfoReturn(obj,diffDays);
        int stunum=checking[0];
        int ontime=checking[1];
        int late=checking[2];
        int absent=checking[3];
        int attendpct=checking[4];
        int checkpct=checking[5];
        System.out.println("받아온 attendinfo값 : "+stunum+","+ontime+","+late+","+absent+","+attendpct+","+checkpct);
        System.out.println("attendinfo값 처리 성공");

        result1[i]=dao.attendInfoUpdate(stunum, ontime, late, absent, checkpct, attendpct);
        System.out.println("attendinfo 업데이트 성공");
        result2[i]=dao.attendTimeUpdate(stunum);
        System.out.println("attendtime 업데이트 성공");
        Thread.sleep(300);
    }
```

```
}catch(Exception e){
    out.print("Exception");
    out.flush();
    out.close();
    return;
}
if(result1.length==tableRow&&result2.length==tableRow){
    System.out.println("출석마감 성공");
    out.print("ok");
    out.flush();
    out.close();
}else{
    System.out.println("출석마감 실패");
    out.print("fail");
    out.flush();
    out.close();
}
```

# 3_4_3_3. 백엔드 – 주요 기능 – 강사 출석마감(METHOD & DAO)

```
//현재 진행중인 강좌 시작,마감날짜 가지고 총 일수 구하는곳
public int subAttendTime(String startdate,String enddate){
    this.startdate=startdate;
    this.enddate=enddate;
    int diffDays=0;

    SimpleDateFormat formatter = new SimpleDateFormat("yyyy/MM/dd");
    try {
        Date beginDate = formatter.parse(startdate);
        Date endDate = formatter.parse(enddate);

        long diff = endDate.getTime() - beginDate.getTime();
        diffDays = (int) (diff / (24 * 60 * 60 * 1000));
        System.out.println(diffDays);

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

    return diffDays;
}
```

```
//출석마감에 사용될 각 강사팀의 학생들의 로우갯수 리턴
public int tableRowCheckReturn(String teachername) throws SQLException{
    String sql="select count(*) as cnt from student inner join emp1 on student.teachername=emp1.empname"
            + " inner join attendtime on student.stunum=attendtime.stunum inner join attendinfo"
            + " on student.stunum=attendinfo.stunum where student.teachername=?";
    int result=0;
    try{
        conn=MyOracle.getConnection();
        pstmt=conn.prepareStatement(sql);
        pstmt.setString(1, teachername);
        rs=pstmt.executeQuery();
        if(rs.next()){
            result=rs.getInt("cnt");
        }
    }finally{
        if(rs!=null)rs.close();
        if(pstmt!=null)pstmt.close();
        if(conn!=null)conn.close();
    }

    return result;
}
```

# 3_4_3_4. 백엔드 – 주요 기능 – 강사 출석마감(DAO)

```java
//출석마감시 모든 attendinfo정보 업데이트
public int attendInfoUpdate(int stunum,int ontime,int late,int absent,int checkpct,int attendpct){
    String sql="update attendinfo set ontime=?,late=?,absent=?,checkpct=?,attendpct=? where stunum=?";
    int result=0;
    System.out.println("attendinfo 메소드 접근성공");
    try {
        conn=MyOracle.getConnection();
        conn.setAutoCommit(false);
        pstmt=conn.prepareStatement(sql);
        pstmt.setInt(1, ontime);
        pstmt.setInt(2, late);
        pstmt.setInt(3, absent);
        pstmt.setInt(4, checkpct);
        pstmt.setInt(5, attendpct);
        pstmt.setInt(6, stunum);
        result=pstmt.executeUpdate();
        System.out.println("attendinfo 작업 성공");
    } catch (SQLException e) {
        try {
            conn.rollback();
            return result;
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }finally{
        try {
            if(pstmt!=null)pstmt.close();
            if(conn!=null)conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return result;
}
```

```java
//출석마감시 모든 attendtime정보 업데이트
public int attendTimeUpdate(int stunum){
    String sql="update attendtime set arrtime='',leavetime='',attendstat=3 where stunum=?";
    int result=0;
    System.out.println("attendtime 메소드 접근성공");
    try {
        conn=MyOracle.getConnection();
        conn.setAutoCommit(false);
        pstmt=conn.prepareStatement(sql);
        pstmt.setInt(1, stunum);
        result=pstmt.executeUpdate();
        System.out.println("attendtime 작업 성공");
    } catch (SQLException e) {
        try {
            conn.rollback();
            return result;
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }finally{
        try {
            if(pstmt!=null)pstmt.close();
            if(conn!=null)conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return result;
}
```

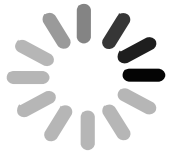
# 3_4_3_5. 백엔드 – 주요 기능 – 강사 출석마감(METHOD)

```
//출결상태 리턴하는곳(정상출석,지각,결석)
public int[] attendInfoReturn(Object[] obj1,int diffDays){
    this.diffDays=diffDays;
    Object[] obj=obj1;
    LMS_AttendInfo_Dto check1=(LMS_AttendInfo_Dto)obj[0];
    Home_AttendTime_Dto check2=(Home_AttendTime_Dto)obj[1];
    int stunum=check2.getStunum();
    int ontime=check1.getOntime();
    int late=check1.getLate();
    int absent=check1.getAbsent();
    int attendstat=check2.getAttendstat();
    String leavetime=check2.getLeavetime();
    int attendpct=0;
    int checkpct=0;

    int[] checking=null;
```

```
//입실버튼시 정상출석이고 퇴실버튼 눌렀을때      -처리 정상출석+1
if(attendstat==1&&!leavetime.equals("")){
    System.out.println("정상출석 접근");
    ontime+=1;
    try{
        attendpct=(int) Math.round((((double)(ontime+late+absent)/this.diffDays)*100));
        checkpct=(int) Math.round((((double)(ontime+late)/(ontime+late+absent))*100));
    }catch(java.lang.ArithmeticException e){
        attendpct=0;
        checkpct=0;
    }
    checking=new int[]{stunum,ontime,late,absent,attendpct,checkpct};

    return checking;
```



# 시연