



# Project #1

## Web Application Development





# Web Application Development

- 이번 프로젝트의 목표는 아래와 같습니다

csv는 그냥 예시이고, 수정해도 됨

## 가상 수강 신청 사이트 구축

1. 주어진 수강 신청 데이터를 기반으로 데이터 베이스를 생성 (Back-End)
2. 데이터 베이스에 연결 가능한 사용자 어플리케이션을 제작 (Front-End)

- Data Set을 참고하여 Table을 만들고 Data를 Database에 추가합니다
  - 어디까지나 참고일 뿐이며, 자유롭게 스키마 변경 및 추가 가능
  - Wiki에 본인의 Schema 디자인 명시 필수 (제공된 Data Set이어도 필수)
- 각각의 Data csv파일은 HYU LMS에서 다운받을 수 있습니다
- Assignment #1과 실습 수업 때 배운 내용을 잘 활용하시길 바랍니다

# Web Application Development

## 아래의 요구사항을 모두 포함해야 합니다

2022년까만 검색이 된다. 왜지? 모르겠다.  
하지만 2022년까만 필요하다. 그래서 상관이 없다.  
2022년의 모든 수업시간 검색까지 구현함

- All
  - 지속적인 트랜잭션이 가능한 인터페이스
  - 수강 편람 기능
    - ✓ 수업번호, 학수번호, 교과목명으로 지정 검색 가능
    - ✓ 신청버튼, 수업번호, 학수번호, 교과목명, 교강사이름, **수업시간**, 신청인원/수강정원, 강의실(건물+호수) 표시
- Admin(관리자)
  - 관리자 로그인/로그아웃
  - 학생 정보 / 과목 상태 조회 및 변경
    - ✓ 학생 정보 조회 및 변경 : 1)금학기 시간표 조회, 2)성적 조회, 3)지도교수 조회, 4)재학/휴학/자퇴 등 상태 변경
    - ✓ 과목 정보 조회 및 변경 : 1)과목 증원 가능, 2)수강 허용 반영: **정원 늘리기**
  - 과목 설강 및 폐강 과목x 수업 O -> class
  - **OLAP** (**필요시** 스키마 변경/추가 가능)
    - ✓ **평점 평균**과 **특정 과목의 학점** 간 차이(평점평균-과목학점)가 가장 큰 Top10 과목 추출
- User(사용자)
  - 사용자 로그인/로그아웃
  - 수강 신청 및 취소
    - ✓ **재수강 여부** 표시
    - ✓ 수강 신청 전에 **희망 수업 신청** 가능 (장바구니 기능)
  - 수업 시간표 생성

2022년만 사용해도됨

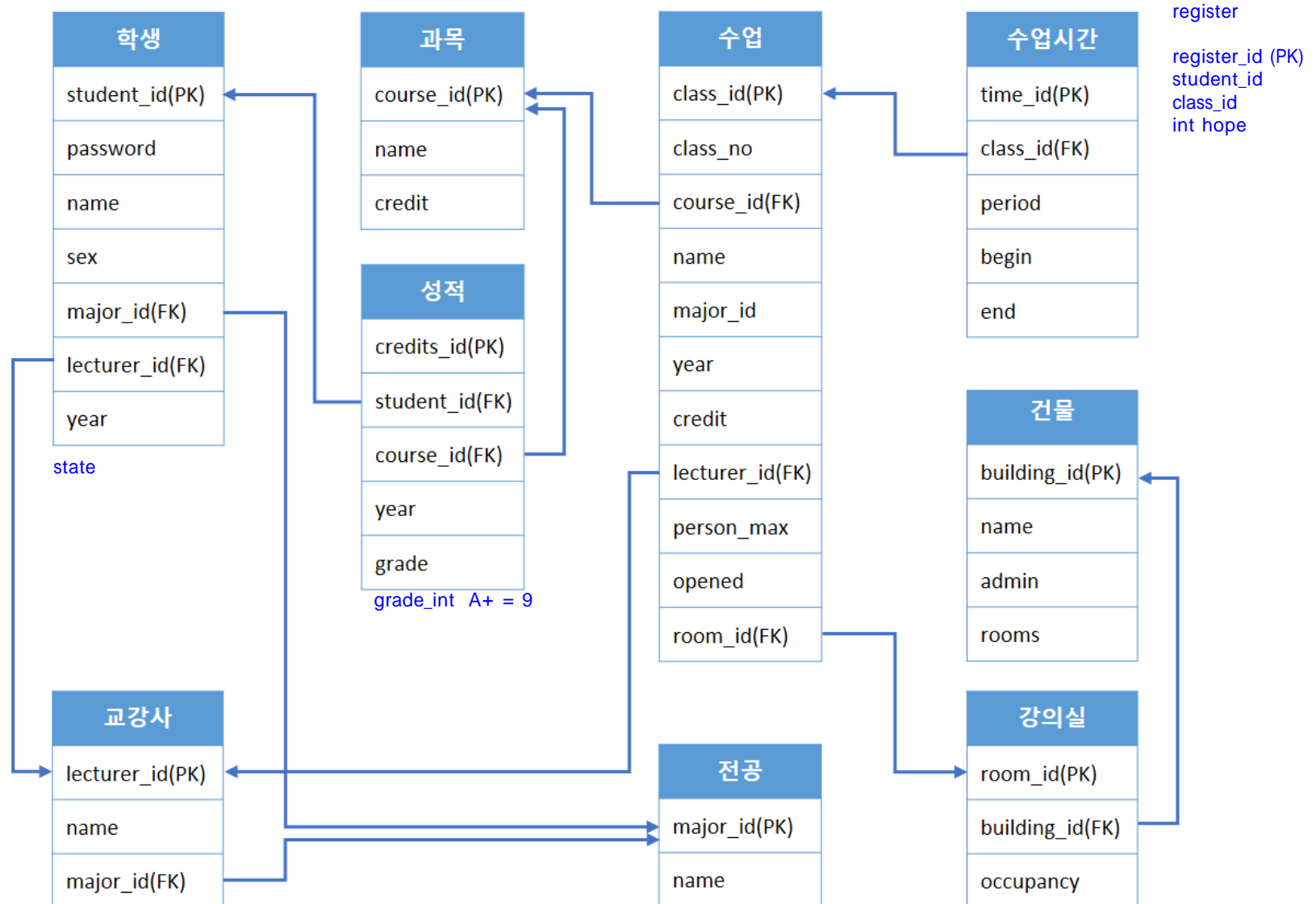
평점 평균 : 사진에있음  
x, y, z과목에 대해  
a, b, c 학생

# Web Application Development

## 아래의 요구사항을 모두 포함해야 합니다

- 검색 기능
  - 수업번호, 학수번호는 완전 일치 검색
  - 교과목명은 키워드 포함 검색
- 관리 기능
  - 설강 및 폐강, 학생 정보 조회, 통계 조회는 관리자만 가능
    - ✓ 조건1: 수강 정원이 강의실 수용 인원보다 초과할 경우 과목 개설 불가능
    - ✓ 조건2: 수업이 폐강되면 해당 수업을 신청/희망 한 학생들의 목록에서 과목 내역 삭제
    - ✓ 조건3: 토요일과 평일 18시 이후 수업은 E-러닝 강의로 분류, 일요일은 과목 개설 불가능
- 수강신청 기능
  - 수강 신청 내역의 삽입, 삭제, 갱신
  - 수강 신청 시 발생하는 제약 조건 처리
    - ✓ 조건1: 이전 성적이 B0이상일 경우 수강 신청 불가능 credits.grade
    - ✓ 조건2: 정원이 다 찼을 경우 해당 과목 수강 신청 불가능
    - ✓ 조건3: 동일 시간대에 2개 이상의 과목 수강 신청은 불가능
    - ✓ 조건4: 최대 학점은 18학점으로 제한하여 초과 신청은 불가능
- 확장 기능
  - 사용자(학생)별로 자신의 수업 시간표 생성
    - ✓ E-러닝 강의는 신청내역/희망내역에는 표시되지만 시간표에는 표시되지 않음
  - 관리자는 모든 사용자의 수업 시간표를 조회 가능

# Schema Example



# Schema Example

## • 제공된 Data Set에 대한 설명은 아래와 같습니다

### ▪ 강의실(room)

강의실 개수의 합과 건물의 총 강의실 개수는 같아야 함

### ▪ 건물(building)

건물의 총 강의실 개수는 강의실 개수의 합과 같아야 함

### ▪ 과목(course)

### ▪ 전공(major)

### ▪ 성적(credits)

이수 년도(year)에는 학기를 구분하지 않음

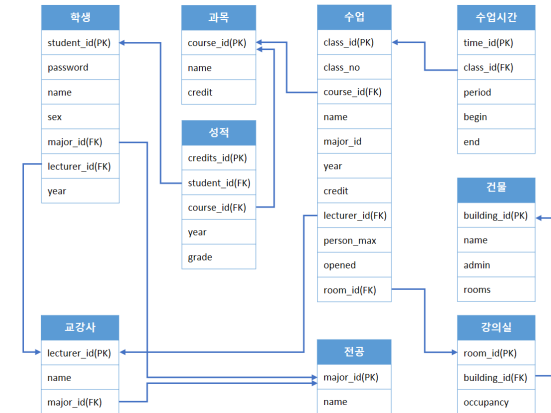
### ▪ 교강사(lecturer)

### ▪ 학생(student)

학생의 지도교수는 교강사 목록에 실제로 존재해야 함

### ▪ 수업(class)

신청 수강 정원이 강의실의 수용 인원을 넘을 수 없음



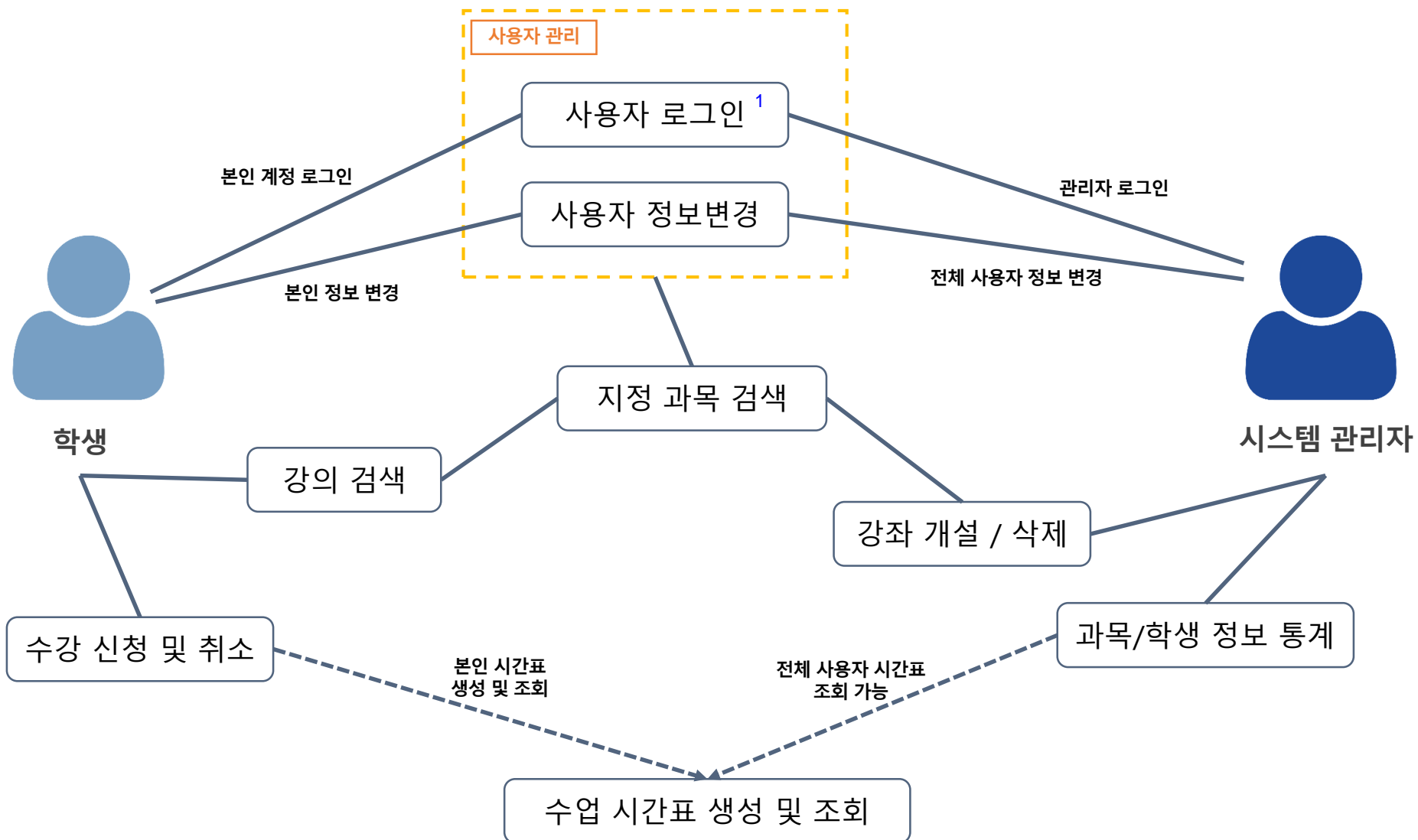
### ▪ 수업시간(time)

“시작시간”, “종료시간” 필드 : \*빨간색을 제외한 필드는 필요 없음

1900-01-0DTHH:MM:00.000Z

필드명	내용
D	월요일부터 토요일까지 각 정수로 표시 (1~6)
HH	시
MM	분

- 현재 시간의 기준은 2022년
- ex) 수요일 오후 2시 30분 수업 → 1900-01-03T02:30.000Z





# Judging System

- 기본 점수(90)
  - 검색 기능 : 수강 편람 --- 15
  - 관리 기능
    - 관리자 - 로그인 / 로그아웃 --- 5
    - 수강 개설 및 삭제 가능 --- 10
    - 학생/과목 상태 조회 및 변경 --- 10
    - 통계 기능 (OLAP) --- 25
  - 수강 신청 기능
    - 사용자 - 로그인 / 로그아웃 --- 5
    - 수강 신청 및 취소 가능 --- 10
    - 수업 시간표 생성 가능 --- 10
- 추가 점수(10) (요구된 기능을 모두 만족했다는 전제 하에 추가 점수가 부과됩니다)
  - GUI
  - 그 외 추가 기능 구현
 

희망수업 인원
제일 결과목





# Judging System

- 이번 프로젝트의 채점 환경(**Default Version**)은 아래와 같습니다

**IntelliJ IDEA 2022.2**  
**MySQL 8.0**  
**Tomcat 9.0.65**  
**SDK Amazon Corretto version 15.0.2**  
**Connector mysql-connector-java-8.0.20.jar**

- Correto-15 **JDK**는 IntelliJ 내에서 [SDK추가 → JDK 다운로드]에서 다운받을 수 있습니다
- Connector **.jar파일**은 HYU LMS에서 다운받을 수 있습니다
- ❖ 특별한 이유로 **이 외의 Version을 사용할 경우 Wiki에 명시**하고 조교가 실행할 수 있도록 **실행 과정을 상세히 서술** 하시길 바랍니다
- ❖ 그렇지 않을 경우 **위의 환경에서 채점이 진행**되는데, Version차이로 인하여 생기는 문제는 조교가 책임지지 않으며 **빌드와 실행이 안 될 경우 점수는 0점** 입니다



- DB name은 'DB+학번'으로 합니다
- 제출 하는 dbURL에서의 포트 번호는 3307으로 맞춥니다  
ex) String dbURL = "jdbc:mysql://localhost:3307/DB20xx123456?serverTimezone=Asia/Seoul";

- Database와 Table을 만든 SQL Script 파일을 함께 제출합니다  
조교의 local 환경에서 채점할 수 있도록 해야 하며,

```
DROP DATABASE IF EXISTS 데이터베이스이름;  
CREATE DATABASE IF NOT EXISTS 데이터베이스이름;  
USE 데이터베이스이름;
```

Script 첫 부분에는 위의 쿼리가 시작되어야 합니다

- Wiki에는
  - 1) 자신의 Schema 디자인, 2) UI 디자인, 3) 본인 Schema에 맞는 Input Data Set 설명,
  - 4) 구현된 코드 설명, 5) 빌드 과정 및 실행 과정, 6) 각 요구사항에 대한 실행 결과,
  - 7) (구현시) 요구사항 외 추가 기능 및 실행 결과위의 내용이 필수로 포함되어야 합니다



## ❖ Code & Script

- **Completeness** : 명세의 요구 조건을 모두 올바르게 구현해야 합니다
- **Defensiveness** : 발생할 수 있는 예외 상황에 대처할 수 있어야 합니다
- **Comment** : 코드에는 반드시 주석이 있어야 합니다

➤ Submission via HYU GitLab “git push”

## ❖ Wiki

- **Design** : 명세에서 요구하는 조건에 대해서 어떻게 사이트를 구성할 것인지에 대한 **UI계획**과, 주어진 Data Set을 보고 설계한 **DB Schema**를 함께 설명합니다
- **Implement** : 이번 요구사항에 대하여 어떤 **SQL문**을 작성하였는지 서술합니다
- **Result** : 해당 명세에서 요구한 부분이 정상적으로 동작하는 **실행 결과를 캡처**하고, 이에 대한 동작 과정에 대해 설명합니다
- **Trouble shooting** : 과제를 수행하면서 마주하였던 문제와 이에 대한 **해결 과정**을 서술합니다. 혹여 문제를 해결하지 못하였다면 어떤 문제였고 어떻게 해결하려 하였는지에 대해서 서술합니다

➤ Submission via HYU LMS      request 헤더      time\_t  
html 공부

## ❖ Demo

- 최종적으로 개발된 사이트에 대해 정상적으로 결과를 출력하는 과정들을 시연 및 녹화한 별도의 **영상 파일**을 제출합니다

➤ Submission via HYU LMS



### ❖ Milestone1

- 제공 드린 Data Set을 참고하여 DB Schema를 작성하고,  
사이트 디자인 구성 계획과,  
요구사항이 반영된 SQL문 (수도)코드가 포함된 report(wiki) 제출
  - 최종본과 달라도 괜찮습니다
- Deadline : ~10/31 11:59 pm .md

### ❖ Milestone2

- 전체 project#1 code가 담긴 프로젝트 폴더를 압축하여 제출
- SQL script 파일 제출
- 최종 Wiki 제출
- Demo 영상 제출
- Deadline : ~11/8 11:59 pm