

# 프로젝트 최종 보고서

1733022 정현수

## 1. 프로젝트 개요

### ■ 프로젝트명

- 학사 관리 프로그램

### ■ 프로젝트 설명

상기 프로젝트는 학생과 교수의 학사 관련 업무를 보다 편리하고 효율적으로 사용할 수 있도록 개발하는 것을 주요 목표로 설정하고 있다. 즉, 프로젝트 주제가 학사 관리 프로그램이기 때문에 잠재적인 이용 대상자는 대학생과 교수로 선정하였다. 프로젝트를 통해 개발된 프로그램에 대한 잠재적 사용자의 특성은 다음과 같다. 학생, 교수로 구분하여 서비스를 제공한다. 여기서 교수는 학생들을 가르치는 것을, 학생은 학업을 수행하는 것을 주요 목적으로 하고 있다. 그러므로 교수는 강의 및 학생 관리에 대한 기능을 제공하고 학생은 학업을 수행하는 주제이므로 학업 수행에 필요한 정보를 열람하고 수강과 및 성적 확인과 관련된 기능을 제공하게 될 것이다.

## 2. 시스템 개요

### ■ 시스템 주요 기능

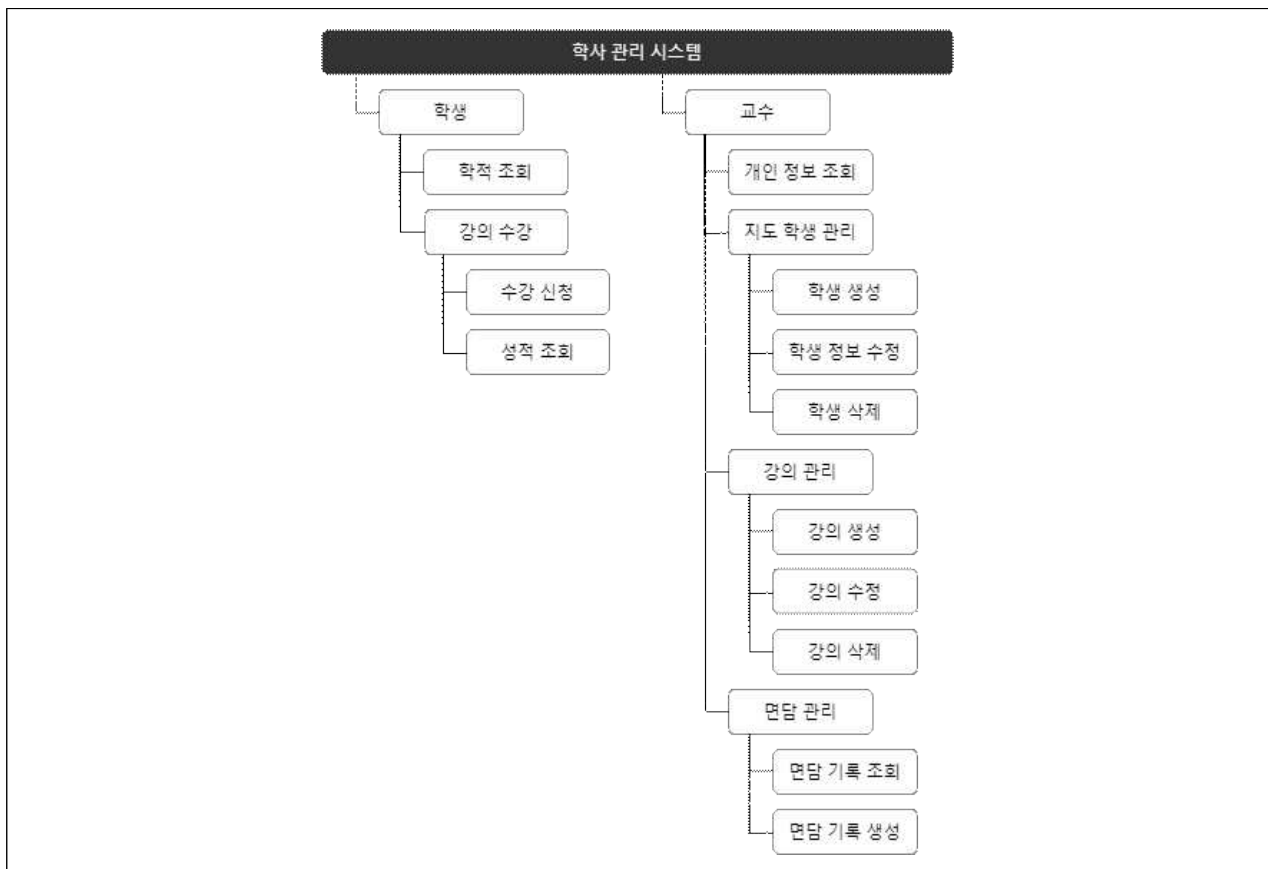
#### 1. 학생 - 수강 신청을 할 수 있으며 개인 정보와 강의에 관한 이력을 확인할 수 있다.

=> 학생 관련 정보 / 수강신청 및 확인 / 과목별 성적조회

#### 2. 교수 - 강의 및 학생들을 관리하며 개인 정보를 확인할 수 있다.

=> 지도 학생 관리 / 학생 정보 조회 / 학생 성적 관리(부여) / 강의 관리(생성/수정/삭제) / 면담 기록

### ■ 프로젝트 기능 구성도



### 3. 시스템 요구 분석

#### ■ 시스템 요구 분석

##### √공통 사항

- 사용자의 신분에 따라 교수 번호가 부여되어야 한다. (교수는 교수 번호/학생은 학번)
- 사용자는 자신의 신상 정보를 조회할 수 있어야한다.
- 사용자는 반드시 학과에 소속되어야 한다.

##### √교수 부문

- 교수는 강의 관리와 관련된 기능을 제공받을 수 있어야한다.
- 교수는 과목별 수강 학생을 조회할 수 있어야 한다.
- 교수는 수강 학생의 성적을 부여할 수 있어야 한다.
- 교수는 학생(지도 학생)에 대해 생성/수정/삭제 작업을 수행할 수 있어야 한다.
- 교수는 지도 학생의 정보를 조회할 수 있어야 한다.
- 교수는 지도 학생의 면담 기록을 관리할 수 있어야 한다.

##### √학생 부문

- 학생은 수강 신청을 할 수 있어야한다.
- 학생은 강의 이력과 성적을 조회할 수 있어야 한다.
- 학생은 반드시 학과에 소속되어야 한다.
- 학생은 반드시 지도교수를 가져야 한다.

#### ■ 요구 사항 명세서

업무	요구사항	요구사항ID	비고
교수 관리	각 교수는 식별을 위한 교유 번호가 부여된다.	Pro-100	
	교수는 자신의 신상 정보를 조회할 수 있다.	Pro-101	
	교수는 강의 관리에 대한 기능을 제공 받는다.	Pro-200	
	교수는 과목별 수강 학생을 조회할 수 있다.	Pro-201	
	교수는 과목별 수강 학생의 성적을 부여할 수 있다.	Pro-202	
	교수는 지도 학생의 대한 정보를 조회할 수 있다.	Pro-300	
	교수는 학생(지도 학생)을 생성/수정/삭제를 할 수 있다.	Pro-301	
	교수는 지도 학생에 대한 면담 기록을 관리할 수 있다.	Pro-302	
	교수는 반드시 학과에 소속되어야 한다.	Pro-400	
학생 관리	각 학생은 학번으로 구분할 수 있어야 한다.	Stu-100	
	학생은 자신의 학적에 관한 정보를 조회할 수 있어야 한다.	Stu-101	
	학생은 수강 신청과 이력을 조회할 수 있다.	Stu-200	
	학생은 수강한 과목의 성적을 조회할 수 있다.	Stu-201	
	학생은 수강 과목을 검색할 수 있어야 한다.	Stu-202	
	학생은 반드시 학과에 소속되어야 한다.	Stu-301	
	학생은 반드시 지도교수를 가진다.	Stu-302	

## ■ 프로젝트 전체 ERD



```
CREATE TABLE "CONSULTING"
(
    "STU_NUM" NUMBER(8,0),
    "CONSULT_NO" NUMBER,
    "CONSULT_DATE" DATE,
    "CONSULT_CONTENT" NVARCHAR2(200)
);

CREATE TABLE "DEPT"
(
    "DEPT_CODE" NUMBER(8,0),
    "DEPT_NAME" NVARCHAR2(12)
);

CREATE TABLE "LECTURE"
(
    "LEC_CODE" NUMBER(8,0),
    "LEC_NAME" NVARCHAR2(12),
    "LEC_CREDIT" NUMBER(1,0),
    "LEC_ROOM" CHAR(4 CHAR),
    "LEC_CLASSIFY" NCHAR(2)
);
```

```

CREATE TABLE "LECTURE_LEARNER"
(
    "LEC_CODE" NUMBER(8,0),
    "STU_NUM" NUMBER(8,0),
    "LEARNER_GRADE" CHAR(2 BYTE)
);

CREATE TABLE "LECTURE_TEACH"
(
    "PROF_NUM" NUMBER(8,0),
    "LEC_CODE" NUMBER(8,0)
);

CREATE TABLE "PROFESSOR"
(
    "PROF_NUM" NUMBER(8,0),
    "PROF_NAME" NVARCHAR2(4),
    "DEPT_CODE" NUMBER(8,0),
    "PROF_ADDRESS" NVARCHAR2(20),
    "PROF_TEL" CHAR(13 BYTE),
    "PROF_OFFICE" CHAR(4 BYTE),
    "PROF_PASSWORD" VARCHAR2(20 BYTE),
    "PROF_BIRTH" DATE
);

CREATE TABLE "STUDENT"
(
    "STU_NUM" NUMBER(8,0),
    "STU_NAME" NVARCHAR2(4),
    "STU_ADDR" NVARCHAR2(20),
    "STU_TEL" CHAR(13 BYTE),
    "STU_CONDITION" NCHAR(2),
    "STU_PASSWORD" VARCHAR2(20 BYTE),
    "STU_MAIN_PROF" NUMBER(8,0),
    "DEPT_CODE" NUMBER(8,0),
    "STU_GRADE" NCHAR(3),
    "STU_BIRTH" DATE,
    "STU_ADMISSION_DATE" DATE
);

Insert into CONSULTING values (20185932,2,to_date('20/12/04','RR/MM/DD'),'면담기록 중');
Insert into CONSULTING values (20185932,1,to_date('20/12/04','RR/MM/DD'),'면담기록생성 테스트중입니다. ');
Insert into CONSULTING values (20191004,1,to_date('19/07/01','RR/MM/DD'),'진로에 대한 면담을 진행');

Insert into CONSULTING values (20185932,3,to_date('20/12/04','RR/MM/DD'),'테스트 중입니다. ');
Insert into CONSULTING values (20191004,2,to_date('20/12/05','RR/MM/DD'),'123123');
Insert into CONSULTING values (20185932,4,to_date('20/12/08','RR/MM/DD'),'안녕하십니까');
Insert into CONSULTING values (20185932,5,to_date('20/12/09','RR/MM/DD'),'탕수육은 역시 부먹입니다. ');

Insert into DEPT values (20000001,'컴퓨터소프트웨어과');
Insert into DEPT values (20000002,'정보보안과');

Insert into LECTURE values (10000001,'생활과법률',2,'K702','교양');
Insert into LECTURE values (10000002,'음악산책',2,'K704','교양');
Insert into LECTURE values (10000003,'파이썬',3,'K405','전공');
Insert into LECTURE values (10000004,'C언어',3,'K406','전공');

```

```

Insert into LECTURE values (10000005,'서비스의이해',2,'K507','교양');
Insert into LECTURE values (10000009,'중국어1',3,'K801','교양');
Insert into LECTURE values (10000007,'자바',3,'K802','전공');
Insert into LECTURE values (10000011,'영어회화',1,'K804','교양');
Insert into LECTURE_LEARNER values (10000009,20171125,null);
Insert into LECTURE_LEARNER values (10000001,20171125,null);
Insert into LECTURE_LEARNER values (10000011,20171125,null);
Insert into LECTURE_LEARNER values (10000001,20171165,null);
Insert into LECTURE_LEARNER values (10000004,20181445,null);
Insert into LECTURE_LEARNER values (10000007,20181445,null);
Insert into LECTURE_LEARNER values (10000005,20181463,null);
Insert into LECTURE_LEARNER values (10000002,20181463,null);
Insert into LECTURE_LEARNER values (10000007,20201119,null);
Insert into LECTURE_LEARNER values (10000002,20201119,null);
Insert into LECTURE_LEARNER values (10000009,20201119,null);
Insert into LECTURE_LEARNER values (10000002,20181445,'A+');
Insert into LECTURE_LEARNER values (10000003,20171125,null);
Insert into LECTURE_LEARNER values (10000005,20181445,null);

Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40000011,10000001);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40000011,10000005);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40000011,10000007);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40000011,10000009);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40001111,10000002);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40001111,10000003);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40001111,10000004);
Insert into LECTURE_TEACH (PROF_NUM,LEC_CODE) values (40001111,10000011);

Insert into PROFESSOR values (40000011,'강수재',20000002,'서울 용산구','010-1234-1111','E209','1q2w3e4r',to_date('70/08/21','RR/MM/DD'));
Insert into PROFESSOR values (40001111,'정바스',20000001,'경기 안산','010-1234-2222','E304','1q2w3e4r',to_date('75/06/21','RR/MM/DD'));

Insert into STUDENT values (20191004,'김영화','경기 부천','010-1234-1234','휴학','1q2w3e4r',40001111,20000001,'2학년',to_date('99/07/12','RR/MM/DD'),to_date('19/03/05','RR/MM/DD'));
Insert into STUDENT values (20181445,'정노루','경기 김포','010-1234-1111','재학','1q2w3e4r',40000011,20000002,'1학년',to_date('96/11/05','RR/MM/DD'),to_date('18/03/06','RR/MM/DD'));
Insert into STUDENT values (20181463,'김미화','경기 수원','010-1234-2222','휴학','1q2w3e4r',40000011,20000002,'2학년',to_date('98/06/05','RR/MM/DD'),to_date('18/03/06','RR/MM/DD'));
Insert into STUDENT values (20171125,'이매자','경기 부천','010-1234-3333','재학','1q2w3e4r',40000011,20000002,'3학년',to_date('97/05/30','RR/MM/DD'),to_date('17/03/04','RR/MM/DD'));
Insert into STUDENT values (20181543,'사미자','경기 성남','010-1234-1029','재학','1q2w3e4r',40001111,20000001,'1학년',to_date('98/08/06','RR/MM/DD'),to_date('18/03/06','RR/MM/DD'));
Insert into STUDENT values (20201119,'조영준','서울 중구','010-1234-5555','재학','1q2w3e4r',40000011,20000002,'1학년',to_date('00/07/08','RR/MM/DD'),to_date('20/03/02','RR/MM/DD'));
Insert into STUDENT values (20208676,'조영순','인천 남동구','010-1234-6058','휴학','1q2w3e4r',40001111,20000001,'1학년',to_date('00/10/12','RR/MM/DD'),to_date('20/03/02','RR/MM/DD'));
Insert into STUDENT values (20185932,'박사균','경기 안양','010-1234-3847','재학','1q2w3e4r',40001111,20000001,'2학년',to_date('98/06/05','RR/MM/DD'),to_date('18/03/06','RR/MM/DD'));
Insert into STUDENT values (20171165,'강나라','경기 부천','010-1234-4444','재학','1q2w3e4r',40000011,20000002,'3학년',to_date('96/04/21','RR/MM/DD'),to_date('17/03/04','RR/MM/DD'));

ALTER TABLE "CONSULTING" ADD CONSTRAINT "CONSULTING_PK" PRIMARY KEY ("STU_NUM", "CONSULT_NO");
ALTER TABLE "CONSULTING" MODIFY ("CONSULT_CONTENT" NOT NULL ENABLE);
ALTER TABLE "CONSULTING" MODIFY ("CONSULT_DATE" NOT NULL ENABLE);
ALTER TABLE "CONSULTING" MODIFY ("CONSULT_NO" NOT NULL ENABLE);
ALTER TABLE "CONSULTING" MODIFY ("STU_NUM" NOT NULL ENABLE);

```

```

ALTER TABLE "DEPT" MODIFY ("DEPT_NAME"NOT NULL ENABLE);
ALTER TABLE "DEPT" ADD CONSTRAINT "DEPT_PK"PRIMARY KEY ("DEPT_CODE");
ALTER TABLE "DEPT" MODIFY ("DEPT_CODE"NOT NULL ENABLE);
ALTER TABLE "LECTURE" MODIFY ("LEC_CLASSIFY"NOT NULL ENABLE);
ALTER TABLE "LECTURE" MODIFY ("LEC_ROOM"NOT NULL ENABLE);
ALTER TABLE "LECTURE" MODIFY ("LEC_CREDIT"NOT NULL ENABLE);
ALTER TABLE "LECTURE" MODIFY ("LEC_NAME"NOT NULL ENABLE);
ALTER TABLE "LECTURE" MODIFY ("LEC_CODE"NOT NULL ENABLE);
ALTER TABLE "LECTURE" ADD CONSTRAINT "LECTURE_PK"PRIMARY KEY ("LEC_CODE");
ALTER TABLE "LECTURE_LEARNER" MODIFY ("STU_NUM"NOT NULL ENABLE);
ALTER TABLE "LECTURE_LEARNER" MODIFY ("LEC_CODE"NOT NULL ENABLE);
ALTER TABLE "LECTURE_LEARNER" ADD CONSTRAINT "LECTURE_LEARNER_PK"PRIMARY KEY ("LEC_CODE", "STU_NUM");

ALTER TABLE "LECTURE_TEACH" MODIFY ("LEC_CODE"NOT NULL ENABLE);
ALTER TABLE "LECTURE_TEACH" MODIFY ("PROF_NUM"NOT NULL ENABLE);
ALTER TABLE "LECTURE_TEACH" ADD CONSTRAINT "LECTURE_TEACH_PK"PRIMARY KEY ("PROF_NUM", "LEC_CODE");

ALTER TABLE "PROFESSOR" MODIFY ("DEPT_CODE"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" MODIFY ("PROF_BIRTH"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" MODIFY ("PROF_PASSWORD"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" MODIFY ("PROF_OFFICE"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" MODIFY ("PROF_TEL"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" MODIFY ("PROF_ADDRESS"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" MODIFY ("PROF_NAME"NOT NULL ENABLE);
ALTER TABLE "PROFESSOR" ADD CONSTRAINT "PROFESSOR_PK"PRIMARY KEY ("PROF_NUM");
ALTER TABLE "PROFESSOR" MODIFY ("PROF_NUM"NOT NULL ENABLE);

ALTER TABLE "STUDENT" MODIFY ("DEPT_CODE"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_MAIN_PROF"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_ADDR"NOT NULL ENABLE);
ALTER TABLE "STUDENT" ADD CONSTRAINT "TABLE1_PK"PRIMARY KEY ("STU_NUM");
ALTER TABLE "STUDENT" MODIFY ("STU_ADMISSION_DATE"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_BIRTH"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_GRADE"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_PASSWORD"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_CONDITION"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_TEL"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_NAME"NOT NULL ENABLE);
ALTER TABLE "STUDENT" MODIFY ("STU_NUM"NOT NULL ENABLE);

ALTER TABLE "CONSULTING" ADD CONSTRAINT "CONSULTING_FK1"FOREIGN KEY ("STU_NUM") REFERENCES "STUDENT" ("STU_NUM") ON
DELETE CASCADE ENABLE;

ALTER TABLE "LECTURE_LEARNER" ADD CONSTRAINT "LECTURE_LEARNER_FK1" FOREIGN KEY ("STU_NUM") REFERENCES "STUDENT"
("STU_NUM") ON DELETE CASCADE ENABLE;
ALTER TABLE "LECTURE_LEARNER" ADD CONSTRAINT "LECTURE_LEARNER_FK2" FOREIGN KEY ("LEC_CODE") REFERENCES "LECTURE"
("LEC_CODE") ON DELETE CASCADE ENABLE;

ALTER TABLE "LECTURE_TEACH" ADD CONSTRAINT "LECTURE_TEACH_FK1" FOREIGN KEY ("LEC_CODE") REFERENCES "LECTURE"
("LEC_CODE") ON DELETE CASCADE ENABLE;
ALTER TABLE "LECTURE_TEACH" ADD CONSTRAINT "LECTURE_TEACH_FK2" FOREIGN KEY ("PROF_NUM") REFERENCES "PROFESSOR"
("PROF_NUM") ON DELETE CASCADE ENABLE;

ALTER TABLE "PROFESSOR" ADD CONSTRAINT "PROFESSOR_FK1" FOREIGN KEY ("DEPT_CODE") REFERENCES "DEPT" ("DEPT_CODE")
ON DELETE SET NULL ENABLE;

```

```
ALTER TABLE "STUDENT" ADD CONSTRAINT "STUDENT_FK1" FOREIGN KEY ("DEPT_CODE") REFERENCES "DEPT" ("DEPT_CODE") ON DELETE CASCADE ENABLE;
ALTER TABLE "STUDENT" ADD CONSTRAINT "STUDENT_FK2" FOREIGN KEY ("STU_MAIN_PROF") REFERENCES "PROFESSOR" ("PROF_NUM") ON DELETE CASCADE ENABLE;
```

## 5. 개발 환경

### ■ 프로젝트 개발 환경



이번 프로젝트인 학사 관리 프로그램은 윈도우 10 운영체제에서 개발하였다. 또한, 프로젝트에 필요한 데이터베이스는 오라클을 사용하였으며 C#을 이용하여 프로그램을 구현하였다. 이에 따라 데이터베이스 언어는 PL/SQL을 사용하였으며 C#에 대한 개발 프로그램은 비주얼 스튜디오를 사용하였다.

## 6. 시스템 주요 기능

### ■ 공통 기능

해당 프로그램을 실행하면 처음으로 로그인 페이지가 나타나게 된다. 로그인 페이지는 학생, 교수 모두 로그인이라는 공통 기능을 사용하여 서비스 제공 받게 되므로 로그인 페이지는 공통 기능과 관련된 공간이라고 할 수 있다. 각 구성요소와 로그인 페이지가 가진 기능에 대한 설명은 아래와 같다.

#### - 로그인 및 학생/교수 학번 비밀번호 목록 화면

	
<p>처음으로 진입했을 때 나타나는 로그인 화면이다. 사용자는 자신의 신분에 맞는 서비스를 제공받기 위해 화면 상단 부분의 교수 전용/학생 전용을 선택하게 되며 그에 따라 아이디와 비밀번호를 입력하여 로그인을 하게 된다.</p>	<p>로그인 화면에서 하단의 학생/교수 학번 비밀번호 목록 버튼을 클릭했을 때 나타나는 화면이다. 사용자는 해당 화면을 통해서 자신의 학번(교수 번호)와 비밀번호를 확인하고 로그인 화면에서 로그인을 진행하게 된다. 상기 기능은 시연의 편리성을 위해 구현한 기능이다.</p>

## ■ 사용자 전용 기능

로그인 화면에서 학생 전용을 클릭하고 학번 및 비밀번호를 통해 로그인을 진행되었을 때 제공하는 기능이다. 학생 관련 기능을 제공하기 위해 학생 전용 폼을 구현하였으며 해당 폼의 구성 요소와 제공 기능에 대한 설명은 아래와 같다.

메인 화면

SchoolManagementSystem - [학생전용]
메인 화면   로그아웃

**학적기본정보**

학 번	<input type="text" value="20181445"/>	학생이름	<input type="text" value="정노루"/>
학 년	<input type="text" value="1학년"/>	학 과	<input type="text" value="정보보안과"/>
지도교수	<input type="text" value="강수재"/>	입 학 일	<input type="text" value="2018-03-06"/>
생년월일	<input type="text" value="1996-11-05"/>	전화번호	<input type="text" value="010-1234-1111"/>
학 적	<input type="text" value="재학"/>	주 소	<input type="text" value="경기 김포"/>

**수강신청**

강의코드	강의명	학점	강의실	강의구분
10000009	중국어1	3	K801	교양
10000011	영어회화	1	K804	교양
10000003	파이썬	3	K405	전공
10000001	생활과학	2	K702	교양

**수강현황 및 성적조회**

☐ 평점이 부여된 강의만 조회

강의명	강의학점	강의실	강의구분	강의평점
음악산책	2	K704	교양	A+
C언어	3	K406	전공	
서비스의...	2	K507	교양	
자바	3	K802	전공	

강의명 입력

수강신청 카테고리에서 조회되는 과목은 해당 학생이 수강하지 않은 강의만 조회됩니다.

### 1. 상단 메뉴 스트립

사용자가 자유롭게 화면을 전환하기 위해 제공되는 메뉴들이 들어가 있다. 학생 전용 폼에서는 메인 화면과 로그 아웃에 대한 메뉴가 제공되며, 로그 아웃을 클릭할 경우 해당 폼이 종료 되고 로그인 화면이 나타나게 된다.

### 2. 학적기본정보 (상단 왼쪽)

학생 전용 폼에서 제공되는 메인 기능중에 하나로서 학생의 이름, 학과, 지도교수 등 학생과 관련된 정보가 출력되게 된다.

### 3. 수강현황 및 성적 조회 (하단 왼쪽)

학생이 수강하고 있는 과목과 성적을 조회할 수 있는 공간이다. 초기에는 모든 과목(성적이 없는 과목 포함)이 출력되며, 상단의 체크 박스를 클릭할 경우, 성적이 있는 과목만 리스트 뷰에 출력되게 된다. 또한, 텍스트 박스에 강의명을 입력하고 검색 버튼을 누르면 해당 과목에 대한 정보가 검색되어 리스트 뷰에 출력되게 된다. 여기서 만약 텍스트 박스에 빈칸을 입력할 경우, 빈칸을 입력했다는 오류 문구가 출력된 대화상자가 나타나면서 검색이 되지 않으므로 주의를 하여야 한다. 마찬가지로 상단의 체크박스를 누른 상태에서 검색을 하면 성적이 입력된 과목에 한해서 검색되게 된다. 이외에도 오른쪽의 전체 조회 버튼을 누르면 상단의 체크 박스 및 검색 조건(텍스트 박스의 내용 제거)이 초기화되어 전체 과목이 출력되게 된다.

### 4. 수강신청 (오른편)

학생이 원하는 강의에 대해 수강 신청할 수 있는 공간이다. 여기서 수강신청은 해당 학생이 듣지 않은 과목을 기준으로 출력이 되며 원하는 과목을 선택하고 강의 수강 버튼을 누르면 하단 왼쪽의 수강현황 및 성적 조회 리스트 뷰에 자동으로 추가 되게 된다. 이외에도 교수가 강의를 추가할 경우, 추가된 강의에 대한 내용이 DB에 업데이트되어 추후에 학생이 로그인을 하였을 때, 수강신청 카테고리에 새로운 강의를 포함한 강의들이 출력되게 된다.



## ■ 교수 전용 기능 - 메인화면

로그인 화면에서 교수 전용을 클릭하고 학번 및 비밀번호를 통해 로그인을 진행되었을 때 제공하는 기능이다. 교수 전용 기능을 제공하기 위해 교수 전용 폼을 구현하였으며 해당 폼의 구성 요소와 제공 기능에 대한 설명은 아래와 같다.

메인 화면

SchoolManagementSystem - [교수 전용]
— □ ×

메인 화면
강의 관리
로그 아웃

**교수기본정보**

교수번호	40001111	교수이름	정미수
연구실	E304	학과	컴퓨터소프트웨어과
생년월일	1975-06-21	전화번호	010-1234-2222
주소	경기 안산		

**면담기록조회**

**지도학생관리**

**교육강의**

강의명	학점	강의실
음악산책	2	K704
파이썬	3	K405
C언어	3	K406
영어회화	1	K804

**지도학생목록**

학번	이름	학년	전화번호	주소	입학일	생일
20181543	서미자	1학년	010-1234-1029	경기 성남	2018-03-06	1998-08-06
20185932	박사군	2학년	010-1234-3847	경기 안양	2018-03-06	1998-06-05
20191004	김영희	2학년	010-1234-1234	경기 부천	2019-03-05	1999-07-12
20208676	조영순	1학년	010-1234-6058	인천 남동구	2020-03-02	2000-10-12

### 1. 상단 메뉴 스트립

사용자가 자유롭게 화면을 전환하기 위해 제공되는 메뉴들이 들어가 있다. 교수 전용 폼에서는 메인 화면과 강의 관리, 그리고 로그 아웃에 대한 메뉴가 제공되며, 로그 아웃을 클릭할 경우 해당 폼이 종료 되고 로그인 화면이 나타나게 된다.

### 2. 교수기본정보 (상단 왼쪽)

교수 전용 폼에서 제공되는 메인 기능중에 하나로서 학생의 이름, 학과, 지도교수 등 학생과 관련된 정보가 출력되게 된다.

### 3. 교육 강의 (하단 왼쪽)

교수가 가르치고 있는 강의(개설한 강의 포함)에 대한 정보가 나타나는 곳이다.

### 4. 면담기록 조회(상단 오른쪽)

학생의 면담 기록을 조회하기 위해 제공하는 검색 기능이다. 사용자(교수)는 지도학생목록에 한해서 학생을 검색할 수 있으며 빈칸 입력일 경우, 학생 전체가 검색되어 면담 기록 폼에 학생의 전체 정보가 전달되게 되며 지도학생목록에 있는 학생 이름에 한해서 일부 이름만 입력할 경우, 해당 글자가 들어간 이름과 관련된 학생들의 정보가 전달되게 된다.

### 5. 지도학생관리(중단 오른쪽)

학생을 추가하고 수정하고 삭제할 수 있는 기능을 제공한다. 이때 학생을 생성할 경우, 지도학생목록에 해당 학생의 정보가 자동적으로 추가가 되며 지도학생목록의 특정 학생을 누르고 수정 버튼을 누르고 수정 폼의 내용을 입력하여 확인 버튼을 누를 경우, 해당 학생의 정보가 자동적으로 수정되게 된다. 이외에도 삭제버튼을 누를 경우, 해당 학생에 대한 내용이 DB에서 삭제되어 지도학생목록에서 사라지게 된다.

### 6. 지도학생목록(오른쪽 하단)

해당 교수가 지도하고 있는 학생의 목록이 출력되는 곳이다. 해당 카테고리는 지도학생관리의 생성, 수정, 삭제에 따라 내용이 자동적으로 변동되게 된다.

■ 교수 전용 기능 - 메인화면 추가 설명(지도학생 관리 폼)

### 메인 화면 보충 설명 (지도학생 관리 폼 -> 생성/수정)

메인 화면의 학생 생성을 눌렀을 때 나타나는 폼이다. 학생 생성과 관련된 정보를 입력하고 제출을 누르면 학생이 생성되게 된다.

메인 화면의 학생 수정을 눌렀을 때 나타나는 폼이다. 해당 폼에서는 주소와 전화번호만 변경할 수 있게끔 하였으며 제출 버튼을 누르면 해당 학생의 정보가 갱신되어 지도학생 목록에 나타나게 된다.

■ 교수 전용 기능 - 면담 기록 조회

### 메인 화면 관련 폼 (면담 기록 조회)

학번	이름	학년	전화번호	입학일	생일
20171165	강나라	3학년	010-1234-4444	2017-03-04	1996-04-21
20181463	김미희	2학년	010-1234-2222	2018-03-06	1998-06-05
20171125	이매지	3학년	010-1234-3333	2017-03-04	1997-05-30
20181445	정노루	1학년	010-1234-1111	2018-03-06	1996-11-05
20201119	조영준	1학년	010-1234-5555	2020-03-02	2000-07-08

회차    면담 진행 날짜

면담내용생성

닫기

#### 1. 학생기본정보(왼편)

학생에 관한 정보가 출력되는 곳이다. 초기 화면에서는 내용이 없는 빈칸이 출력되며, 상단 오른편의 학생들 중에 하나를 선택할 경우, 그에 대한 정보가 출력되게 된다.

#### 2. 학생 리스트 (상단 오른편)

학생의 목록이 나타나는 리스트 뷰이다. 메인 화면 폼에서 검색한 내용을 토대로 출력되며, 만약 지도학생목록에 없는 학생을 검색하여 검색 버튼을 누를 경우, 빈 내용이 출력되게 된다. 이외에도 한글자나 두글자를 검색할 경우, 지도학생목록에 있는 학생에 한해서 해당 글자가 들어간 학생의 정보가 출력되게 된다.

#### 4. 면담 회차 및 진행 날짜(하단 중앙)

학생 리스트에서 선택된 학생이 진행한 면담에 대한 기록이 나타나는 곳이다. 해당 리스트 뷰에는 면담 회차 및 진행 날짜가 나타나게 된다. 또한, 면담 생성 폼에 가려졌지만, 해당 리스트뷰에서 조회를 원하는 회차를 클릭할 경우, 해당 면담의 내용이 하단 오른편에 출력되게 된다.

#### 5. 면담 기록 생성 폼(하단 오른편)

학생 리스트에서 학생을 선택하고 면담내용생성을 누르면 나타나는 폼으로 날짜와 면담 내용을 입력하고 완료 버튼을 누르면 해당 내용이 DB에 갱신되어 추후에 학생 리스트에서 해당 학생을 클릭했을 때 면담 회차 및 진행 날짜 리스트뷰에서 내용이 갱신되어 출력되는 것을 확인할 수 있다.

### ■ 교수 전용 기능 - 강의 관리

상단의 메뉴 스트립에서 강의 관리를 선택했을 때 나타나는 폼이다. 해당 폼에서는 강의와 관련된 기능을 제공하고 있으며 강의 생성, 수정, 삭제와 관련된 메뉴는 컨텍스트 메뉴로 구현하였다. 이외에도 해당 과목을 수강하는 학생들에 대한 성적을 부여할 수 있으며, 성적이 부여된 학생 계정으로 로그인을 하여 확인해보면 해당 과목에 대한 성적이 교수 계정에서 입력한 성적으로 갱신된 것을 확인 할 수 있다.

## 강의 관리 화면

강의코드	강의명	학점	강의실	강의구분
10000001	생물과...	2	K702	교양
10000005	서비스...	2	K507	교양
10000009	중국어1	3	K801	교양
10000007	자바	3	K802	전공

강의생성

강의수정

강의삭제

학번	이름	학년	생일	입학일	평점
20171125	이매지	3학년	1997-05-30	2017-03-04	
20171165	강나라	3학년	1996-04-21	2017-03-04	

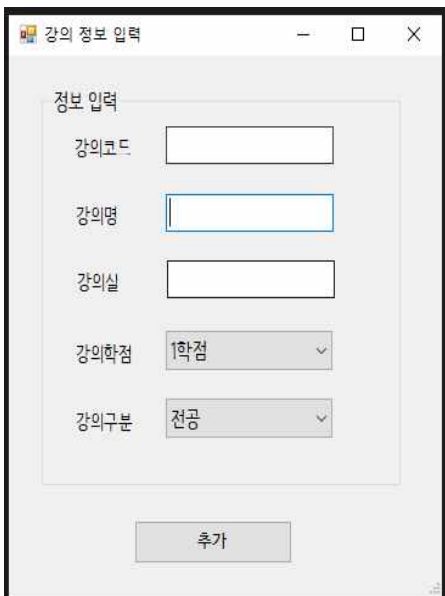

## 1. 강의 관리(목록 / 원편)

교수가 강의하고 있는 강의들이 출력되는 곳이다. 교수는 해당 데이터 그리드뷰에 컨텍스트 메뉴 기능으로 강의를 생성, 수정, 삭제할 수 있다.

## 2. 수강학생관리(오른편)

강의를 수강하고 있는 학생의 목록이 나타나는 곳으로 원편의 강의 관리 데이터 그리드뷰에서 조회를 원하는 강의를 클릭할 경우, 해당 강의를 수강하고 있는 학생의 목록으로 갱신하여 수강학생관리 카테고리의 데이터 그리드뷰에 출력하게 된다. 이외에도 학생을 클릭하고 상단의 콤보박스에서 원하는 성적을 고르고 확인 버튼을 누르면 성적이 부여되게 된다.

### ■ 교수 전용 기능 - 강의 관리 화면 추가 설명(강의 생성, 수정 폼)

강의 관리 화면 추가 설명 (강의 생성, 수정 폼)	
	
강의관리 카테고리의 데이터 그리드뷰에 마우스 오른쪽을 클릭하고 강의 생성을 선택했을 때 나타나는 입력 폼이다. 사용자는 해당 내용들을 입력하고 추가버튼을 눌러 강의를 생성하게 된다.	강의관리 카테고리의 데이터 그리드뷰에 수정을 원하는 과목을 선택하고 마우스 오른쪽을 클릭하여 강의 수정을 선택했을 때 나타나는 수정 폼이다. 보이는 바와 같이 강의 코드, 강의명, 강의실은 수정할 수 없으며 강의 학점과 강의 구분을 변경하고 업데이트 버튼을 누르면 해당 강의의 내용이 수정되게 된다.

## 7. 보완 사항 및 개발 후기

### ■ 보완 사항

첫 프로젝트인 만큼 완벽한 프로그램을 만들고 싶은 열의에 가득찼었다. 그러나 상대적으로 미숙했던 프로그램 구현과 그로 인해 생겨난 에러 등을 해결하는 과정에서 구현하고 싶었던 기능들을 많이 구현하지 못했던 것 같다. 따라서 추후에 보완할 수 있는 기회가 주어진다면 프로그램에 대한 기능을 더 많이 구현하여 프로그램의 완성도를 높이고 싶다. 개인적으로 좋은 프로그램은 단편적인(부분적인) 기능만 제공하는 것보다는 언제, 어디서든, 무엇이든지(Anytime, Anywhere, Anything) 사용자의 목적에 맞게 도움을 줄 수 있는 프로그램이 좋은 프로그램이라고 생각하고 있기 때문이다. 다음으로는 코드 최적화 작업을 수행하고 싶다. 물론 프로그램이 잘 구동되는 상황에서 코드를 최적화하는 작업은 무의미해 보일 수 있다. 그러나 2명의 프로

그래머가 같은 프로그램을 구현한다 했을 때, 상대적으로 짧은 코드로 구현하는 프로그래머가 더 좋은 프로그래머라고 할 수 있을 것이다. 이는 유지보수와 프로그램 효율성 관리 측면에서 더 좋은 코드라고 볼 수 있기 때문이다. 이외에도 사용자에게 더욱 친숙한 프로그램이 될 수 있도록 UI를 개선해 나가고 싶다. 특히, 기회가 주어진다면 가시성과 직관성있는 UI를 개발함으로써 사용자가 좀 더 쉽게 원하는 기능을 사용할 수 있도록 프로그램을 보완해 나갈 것이다.

## ■ 개발 후기

2020년 9월 2학기가 시작되었다. 그리고 12월, 2학기의 마지막을 향해 달려가고 있다. 시험이라는 과정이 남아있지만 개발 후기를 작성하고 제출하면 C#과 DB의 과목이 끝난다고 생각하니 감회가 새로운 것 같다. 내게 이번 프로젝트는 많은 것을 느끼게 해준 프로젝트였다. 1인 프로젝트였지만 프로그램이라는 결과물을 만들기 위해서 어떤 과정으로 개발을 하고 DB는 어떻게 구축해야 하는지를 알 수 있었던 계기가 되었기 때문이다. 과거에는 프로젝트가 간단한 과정을 통해 진행되는 줄로만 알고 있었다. 그러나 프로그램을 구현하는 과정에서 나타나는 에러들은 프로젝트를 진행하는데 많은 어려움을 가져다 주었다. 이러한 에러는 이론적인 부분과는 분명 거리가 존재했다. 따라서 나는 프로젝트가 이론만으로 프로그램을 만들 수 없다는 것을 알게 되었다.

개발 후기를 작성하다 보니 2차 점검일이 생각난다. 나는 프로그램 구현을 거의 끝난 상태에서 회의에 참여하였다. 그래서 인지 자신감이 있었던 것 같다. 하지만 시간이 흐르면서 나의 자신감은 조금씩 사그라들었다. 이는 더 좋은 기획과 UI 디자인으로 개발된 프로그램들이 많았기 때문일 것이다. 다들 나만큼, 혹은 나보다 더 프로젝트에 진지하게 임하고 프로그램을 개발했다는 것을 느꼈다. 물론 나의 프로젝트에 대해 실망을 느꼈냐고 묻는다면 그렇지 않다고 대답할 것이다. 오히려 다른 사람이 개발한 프로그램들은 나에게 좋은 동기 부여가 되었다. 혹자는 자신을 발전시키기 위해서는 남과 비교를 자주 해야 한다고 말을 한다. 남과의 비교를 통해 자신의 부족함을 깨닫고 발전하기 위한 노력을 한다면 자신의 분야에서 한 단계 도약할 수 있는 계기가 될 수 있기 때문이다. 이처럼 프로젝트 기간 중에서 2차 점검일은 나에게 큰 전환점을 마련해주었다고 볼 수 있다.

프로젝트를 진행하면서 문득 이런 생각이 들었다. 나에게 있어 프로그래밍은 무엇일까? 겨울 방학을 앞두고 프로젝트를 마무리할 때까지 내내 들었던 생각이다. 프로젝트를 마무리하면서 내가 생각해 낸 답은 '미술 작품'과 같다는 것이다. 미술 작품은 작가가 재료들을 선택하고 자신만의 방법에 따라 만들어 가는 과정을 통해 완성된다. 이렇게 해서 완성된 작품은 만드는 사람의 경험, 재료의 특성, 제작 기법에 따라 작품의 완성도가 달라지게 된다. 프로그램도 마찬가지다. 자신이 원하는 프로그램이라는 작품을 만들기 위해 언어와 개발 프로그램이라는 재료를 선택하고 자신만의 코딩 스타일을 통해 프로그램을 구현해 나가기 때문이다. 여기서 중요한 점은 프로그래밍 언어라는 재료를 사용하는 법과 프로그램을 만들어가는 과정(혹은 기법)을 배워야 비로소 프로그램이라는 결과물을 만들어 낼 수 있다는 것이다. 낚시대를 갖고 있는 아이에게 낚시를 하는 법을 가르쳐주지 않는다면 그 아이는 아무리 좋은 낚시대를 갖고 있어도 낚시라는 행위를 하지 못할 것이다. 즉, 이번 프로젝트는 나에게 프로그램을 개발하는 과정에 대해서 알려주었다고 볼 수 있다. 어쩌면 졸업 작품 프로젝트도 잘 만들 수 있을 것 같다는 생각이 든다.

담담하게 써내려간 개발 후기가 어느덧 마침표를 향해 달려가고 있다. 지금 생각하면 나는 운이 매우 좋았다. 좋은 교수님들에게 프로그래밍에 대해서 배울 수 있었기 때문이다. 그러나 스스로도 더 많이 노력해야 한다고 생각한다. '늙은 소가 가장 곧은 고랑을 판다'라는 J.하우어의 말처럼 앞으로 많은 프로젝트와 프로그래밍 실습에 대한 경험을 많이 하면 분명 좋은 프로그래머에 한 발짝 다가설 수 있을 것이라고 생각한다. 물론, 여기서 말하는 프로젝트 중에서는 졸업 작품 프로젝트도 포함될 것이다. 대학교 생활의 마지막 프로젝트라고 볼 수 있는 만큼 최선을 다해 좋은 프로그램이라는 결과물을 남기고 싶다. 이를 위해선 더 많은 프로그래밍 지식이 쌓여야 할 것이다. 그러므로 많은 공부를 해야 할 것 같다. 이번 겨울 방학이 많이 바빠질 것 같다.