

최종 보고서

0. 프로젝트 참여자

15011048 배경준

19011043 홍지민

1. 개발 요구사항 정리

1. MySQL 계정 정보는 소스코드내에 정의한다.
2. 프로그램 실행 시 연결을 시도한다.
3. 프로젝트 이름은 학번으로 만든다.
4. MySQL 접속 코드는 아래 소스코드를 사용한다.

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class MainFrame extends JFrame implements ActionListener {
    static Connection con;

    String Driver = "";
    String url = "jdbc:mysql://localhost:3306/madang?&serverTimezone=Asia/Seoul&useSSL=false";
    String userid = "madang";
    String pwd = "madang";

    public MainFrame() {
        super("학사정보 시스템");
        conDB();
    }

    public void conDB() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }

        try {
            con = DriverManager.getConnection(url, userid, pwd);
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }
}
```

```

@Override
public void actionPerformed(ActionEvent e) {
}

public static void main(String[] args) {
    new MainFrame();
}
}

```

1. 프로그램 실행 시의 창에 아래와 같은 버튼들을 구성한다. 각 버튼에 따라 새로운 창으로 넘어간다.
 - 관리자 버튼
 - 교수 버튼
 - 학생 버튼
2. 관리자, 교수, 학생 버튼의 분기에 상관없이 **madang** 계정을 이용하여 데이터베이스에 연결해둔 상태여야 한다.
3. 관리자 메뉴 창에 아래와 같은 요소들을 구성한다. 각 요소에 따라 기능을 실행하거나 새로운 창으로 넘어간다.
 - 결과 출력 텍스트 박스
 - 데이터베이스 초기화 버튼
 - "데이터베이스 초기화 완료" 결과 출력
 - 테이블 목록이 담긴 콤보 박스
 - 선택된 테이블에 따라 테이블 입력/삭제/변경 버튼 클릭 시 연결
 - 테이블 입력 버튼
 - 테이블 입력 창에 아래와 같은 요소들을 구성한다.
 - 선택된 테이블의 속성 값마다 텍스트 박스 구성
 - 저장 버튼
 - 7, 8번 검사 후 결과 출력
 - 취소 버튼
 - 테이블 삭제 버튼
 - 테이블 삭제 창에 아래와 같은 요소들을 구성한다.
 - 조건식을 입력받을 텍스트 박스
 - 저장 버튼
 - 7, 8번 검사 후 결과 출력

- 취소 버튼
- 테이블 변경 버튼
 - 테이블 변경 창에 아래와 같은 요소들을 구성한다.
 - 조건식을 입력받을 텍스트 박스
 - 저장 버튼
 - 7, 8번 검사 후 결과 출력
 - 취소 버튼
- 전체 테이블 보기 버튼

4. 교수 메뉴 창에 아래와 같은 요소들을 구성한다. 각 요소에 따라 기능을 실행하거나 새로운 창으로 넘어간다.

- 교수 번호 입력 버튼
 - 교수 번호가 유효하지 않은 데이터이면 오류메시지 출력
- 학년과 학기 입력 버튼
 - 학년과 학기가 유효하지 않은 데이터이면 오류메시지 출력
- 강의 과목 정보 버튼
 - 버튼을 클릭하면 해당 교수가 해당 학기에 강의했던 과목에 대한 모든 정보 출력
 - 과목 정보 클릭할 수 있는 버튼 포함
 - 과목 정보 중 하나 클릭 시 해당 과목을 수강했거나 수강하는 모든 학생에 대한 정보 출력
- 지도 학생 버튼
 - 현재 본인이 지도하는 학생에 대한 정보 출력
 - 학생 정보 클릭할 수 있는 버튼 포함
 - 학생 정보 중 하나 클릭 시 해당 학생이 수강했거나 수강하고 있는 모든 과목에 대한 성적 정보 출력
- 소속 학과 버튼
 - 소속 학과의 모든 정보 출력
 - 학과장 정보 출력
- 현재 학기의 강의 시간표 버튼

- 해당 교수가 현재 학기 강의하는 과목을 시간표 형태로 표시 - 요일/교시 정보 포함
- 해당 교수가 강의하는 과목에 대한 성적 입력 버튼

5. 학생 메뉴 창에 아래와 같은 요소들을 구성한다. 각 요소에 따라 기능을 실행하거나 새로운 창으로 넘어간다.

- 학생 번호 입력 버튼
 - 학생 번호가 유효하지 않은 데이터이면 오류메시지 출력
- 연도와 학기 입력 버튼
 - 연도와 학기가 유효하지 않은 데이터이면 오류메시지 출력
- 수강 과목 버튼
 - 버튼을 클릭하면 본인이 수강했던 과목의 모든 정보를 보여준다.
- 시간표 버튼
 - 현재 학기에 해당 학생이 수강하는 모든 과목을 시간표 형태로 출력
- 수강내역 신규 삽입 버튼
 - 신청 학점은 10학점으로 제한
 - 신규 수강내역이 10학점이 넘어가면 오류메시지 출력
 - 해당 학생의 등록금 납입 여부 확인
 - 등록금을 전부 납입하지 않았으면 오류메시지 출력
- 동아리 버튼
 - 해당 학생이 소속된 동아리 정보 출력
 - 해당 학생이 동아리 회장인 경우, 동아리에 속한 모든 학생들의 정보 출력
- 성적표 버튼
 - 과목 번호, 과목명, 취득 학점, 평점, GPA 출력

6. 하나의 GUI윈도우 안에서 모든 데이터를 입력받는다.

- 입력해야 하는 부분에서는 하나의 창에 모든 입력 객체 생성

7. 저장/취소 버튼을 구현하여 입력 실행, 입력 취소 기능을 실행해야 한다.

- OK 버튼으로 입력 실행

- Cancel 버튼으로 입력 취소
8. 제약조건을 벗어난 입력값을 받으면 오류 메시지를 표시하고 저장하지 않는다.
- 질의문에 문제가 생기거나, 문제가 생길 것 같은 입력 값에 대해서 검증
9. 오류 메시지를 표시한 후 프로그램이 종료되면 안된다.
- 오류가 발생해도 프로그램이 비정상적으로 종료되지 않도록 알맞게 예외 처리를 해야 한다.
10. 데이터베이스 초기화 기능은 다음 SQL을 이용해 구현한다.
1. `drop table if exists`
 2. `create table`
 3. `insert into`

2. 데이터베이스 요구사항 정리

- 모든 학생은 학생번호를 부여하여 식별하며 학생이름, 학생주소, 학생전화번호, 학생이메일, 전공 학과, 지도교수 정보, 등록금 납부 계좌 정보를 가진다. 학생은 부전공을 1개까지 신청할 수 있다. 학생은 한학기에 최대 10학점까지만 수강 할 수 있다.
- 학생 번호, 이름, 주소, 전화번호, 이메일, 전공 학과 번호, 지도교수 번호, 등록금 납부 계좌, 부전공 학과 번호 필요
- 당학기 신청 학점은 `lecture` 테이블에서 조회하여 구할 수 있어 `student` 테이블의 컬럼에서는 제거

```
CREATE TABLE student (
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(40),
    address VARCHAR(100),
    phone VARCHAR(20),
    email VARCHAR(100),
    major INTEGER,
    professor INTEGER,
    tuition_address VARCHAR(20), // 등록금 납부 계좌
    sub_major INTEGER,

    FOREIGN KEY (major) REFERENCES major(id),
    FOREIGN KEY (sub_major) REFERENCES major(id),
    FOREIGN KEY (professor) REFERENCES professor(id),
);
```

- 프로그램에서 신청 학점 컬럼의 데이터 값과 신청하려는 과목의 학점의 합이 10을 초과하는 경우 올바른 메시지 출력 필요
- 동아리 가입 현황 테이블 생성

```
CREATE TABLE club_history (
    student INTEGER,
    club INTEGER,

    FOREIGN KEY (student) REFERENCES student(id);
    FOREIGN KEY (club) REFERENCES club(id);
);
```

- 교수는 소속학과가 반드시 있어야 하고, 학생을 반드시 담당해야 하며, 강좌에 대한 강의를 반드시 해야 한다. 교수는 1개 이상의 복수 학과에 소속이 가능하다.
- 교수 번호, 이름, 주소, 연락처, 이메일

```
CREATE TABLE professor (
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(40),
    address VARCHAR(100),
    phone VARCHAR(20),
    email VARCHAR(100)
);
```

- 1개 이상의 복수 학과

```
CREATE TABLE professor_major (
    professor INTEGER,
    major INTEGER,

    FOREIGN KEY (professor) REFERENCES professor(id),
    FOREIGN KEY (major) REFERENCES major(id)
);
```

- 교수와 학생은 학생 번호와 교수 번호를 이용한 지도 관계를 가지며, 해당 관계에는 현재 연도/학기 정보를 포함한다.
- 학생 번호와 교수 번호를 이용해 지도 관계 구성
- 해당 관계에 연도, 학기 정보 포함

```
CREATE TABLE relationship (
    student INTEGER PRIMARY KEY,
    professor INTEGER,
    year INTEGER,
    semester INTEGER,

    FOREIGN KEY (student) REFERENCES student(id),
    FOREIGN KEY (professor) REFERENCES professor(id)
);
```

- 학과는 학과번호를 부여하여 식별하며 학과명, 학과전화번호, 학과사무실 정보를 가진다. 해당 학과에서 개설하는 강좌가 반드시 1개 이상 존재해야 하며, 동시에 학과장이 1명 있어야 한다.
- 학과 번호, 이름, 연락처, 사무실 정보, 학과장

```
CREATE TABLE major (
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    phone VARCHAR(20),
    place VARCHAR(100),
    professor INTEGER NOT NULL, // 학과장

    FOREIGN KEY (professor) REFERENCES professor(id)
);
```

- 강좌는 강좌번호, 분반번호, 강의하는 교수 번호, 강좌이름, 강의요일, 강의교시, 취득학점 (1~4), 강좌시간 (1~6), 개설 학과 번호, 강의실 정보가 필요하다.
- 강좌 번호, 분반 번호, 강의하는 교수 번호, 강좌 이름, 강의 요일, 강의 교시, 취득 학점, 강좌 시간, 개설 학과 번호, 강의실 정보

```
CREATE TABLE lecture (
    id INTEGER PRIMARY KEY,
    class_no INTEGER,
    professor INTEGER,
    name VARCHAR(100),
    day VARCHAR(20),
    period VARCHAR(10),
    unit INTEGER,
    course_time INTEGER,
    major INTEGER,
    place VARCHAR(100),

    FOREIGN KEY (professor) REFERENCES professor(id),
    FOREIGN KEY (major) REFERENCES major(id)
);
```

- 수강내역은 학생번호, 강좌번호. 교수번호를 부여하여 식별하며 출석점수, 중간고사점수, 기말고사점수, 기타 점수, 총점 (0 ~ 100), 평점 (A ~ F) 정보를 가진다.
- 학생 번호, 강좌 번호, 교수 번호, 연도, 학기, 출석 점수, 중간고사 점수, 기말고사 점수, 기타 점수, 총점, 평점

```
CREATE TABLE course (
    student INTEGER,
    lecture INTEGER,
    professor INTEGER,
    year INTEGER,
    semester INTEGER,
    attendance_score DOUBLE,
    midterm_score DOUBLE,
    final_score DOUBLE,
    extra_score DOUBLE,
    total_score DOUBLE,
    grade VARCHAR(10),

    FOREIGN KEY (student) REFERENCES student(id),
    FOREIGN KEY (lecture) REFERENCES lecture(id),
    FOREIGN KEY (professor) REFERENCES professor(id)
);
```

- 학생은 1개 이상의 동아리에 가입이 가능하다. 동아리는 동아리 번호, 동아리 이름, 소속 학생 숫자, 회장 학생 정보, 동아리 지도 교수 정보, 동아리방 정보 등을 가져야 한다.
- 번호, 이름, 회장 학생 번호, 동아리 지도 교수 번호, 장소

```
CREATE TABLE club (
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    student INTEGER, // 회장 학생 정보
    professor INTEGER,
    place VARCHAR(50),

    FOREIGN KEY (student) REFERENCES student(id),
    FOREIGN KEY (professor) REFERENCES professor(id)
);
```

- 학생은 1개 이상의 동아리에 가입 가능
- 소속 학생 숫자는 `club_history` 테이블에서 조회할 수 있어 `club` 테이블의 컬럼에서 는 제거

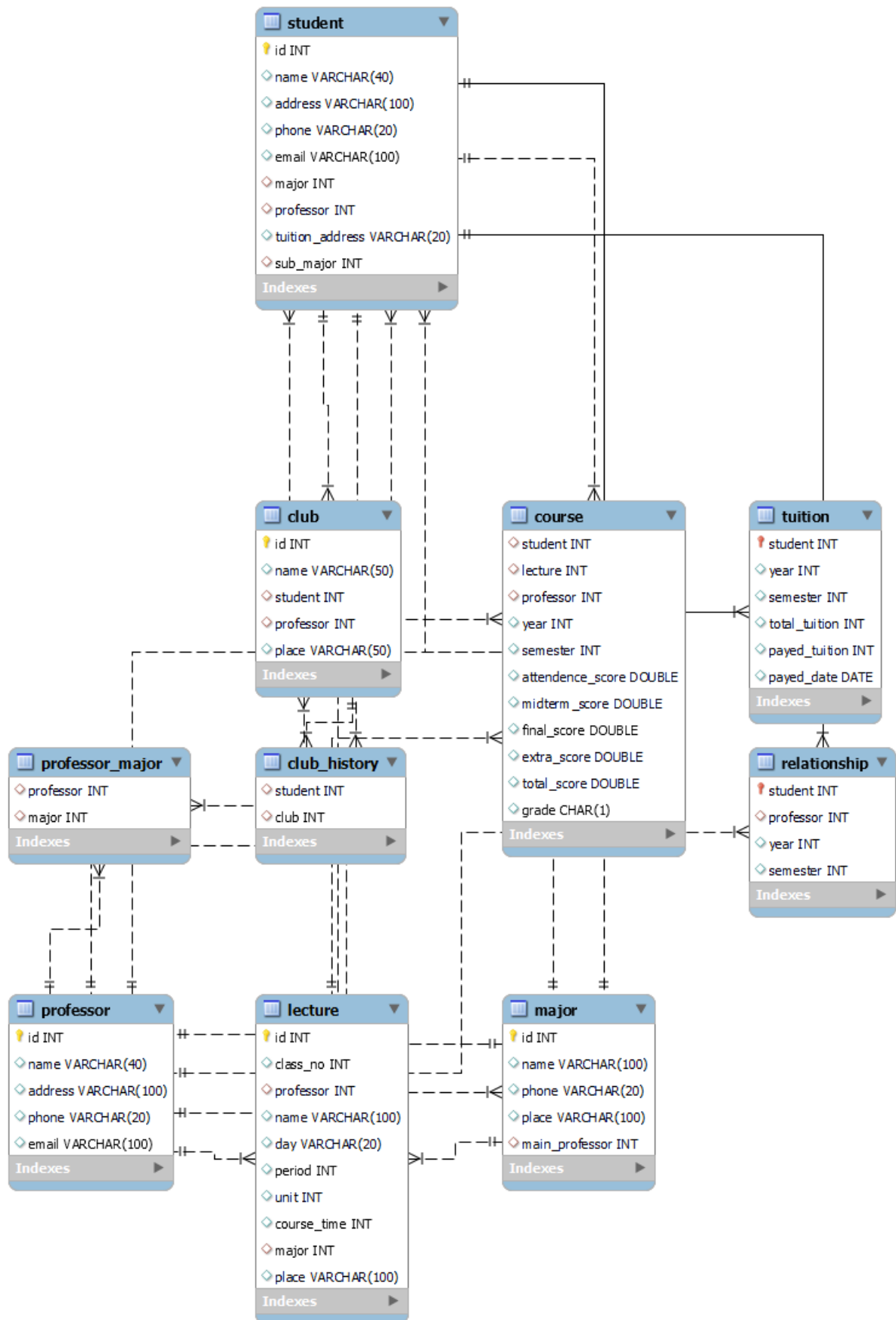
- 학생별 등록금 납부 내역을 기록해야 한다. 학생 번호, 등록금 납부 연도, 등록금 납부 학기, 등록금 총액, 납부 총액, 마지막 납부 일자가 기록되어야 한다. 납부 총액이 등록금 총액보다 작을 경우에는 수강내역 “신규 삽입”에서 “등록금 미납”오류를 표시해야 한다.
- 학생 번호, 등록금 납부 연도, 등록금 납부 학기, 등록금 총액, 납부 총액, 마지막 납부 일자

```
CREATE TABLE tuition (
    student INTEGER,
    year INTEGER,
    semester INTEGER,
    total_tuition INTEGER,
    pated_tuition INTEGER,
    payed_date DATE,

    FOREIGN KEY (student) REFERENCES student(id)
);
```

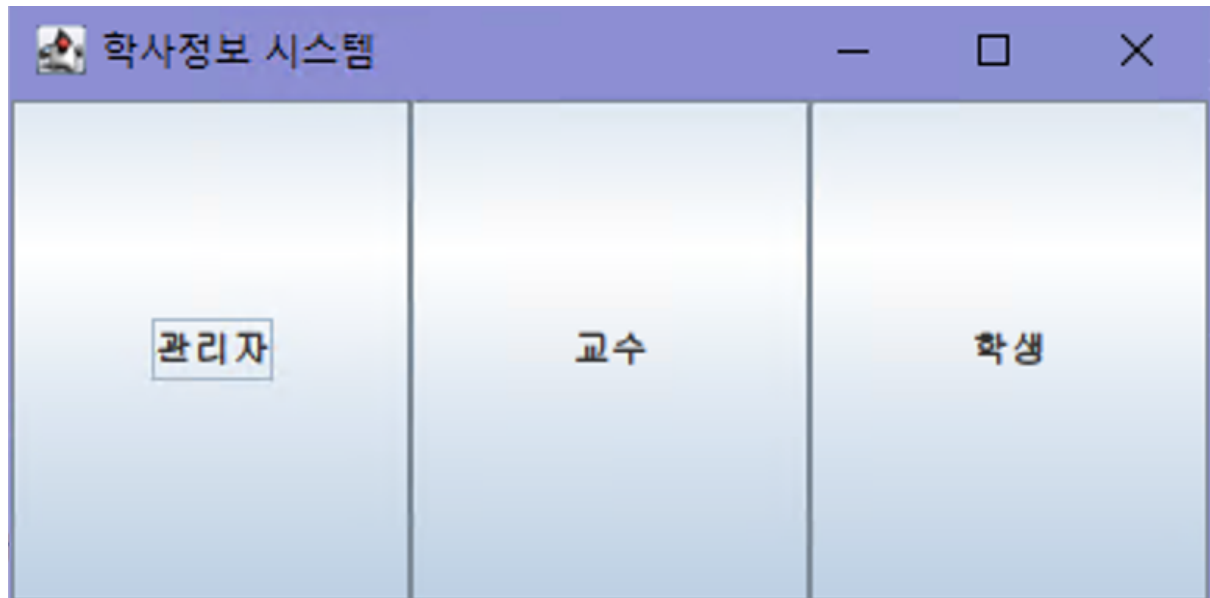
- 납부 총액이 등록금 총액보다 작을 경우, 수강내역 "신규 삽입"에 "등록금 미납" 오류 표시

3. 스키마 설계



4. 프로그램 구조

4-1. 프로그램 시작 창



- 관리자 버튼: 관리자 메뉴 창을 생성합니다.
- 교수 버튼: 교수 메뉴 창을 생성합니다.
- 학생 버튼: 학생 메뉴 창을 생성합니다.

4-2. 관리자 메뉴 창

관리자 메뉴				
club	테이블 조회	테이블 입력	테이블 변경	테이블 삭제
id	name	student	professor	place
1	cluba	1	1	aa
2	clubb	2	2	bb
3	clubc	3	3	cc
4	clubd	4	4	dd
5	clube	5	5	ee
6	clubf	6	6	ff
7	clubg	7	7	gg
8	clubh	8	8	hh
9	clubi	9	9	ii
10	clubj	10	10	jj
11	clubk	11	11	kk
12	clubl	12	12	ll
13	clubm	13	13	mm
14	clubn	14	14	nn
15	clubo	15	15	oo
16	clubp	16	16	pp
17	clubq	17	17	qq
18	clubr	18	18	rr
19	clubs	19	19	ss
20	clubt	20	20	tt
21	clubu	21	21	uu
22	clubv	22	22	vv
23	clubw	23	23	ww
24	clubx	24	24	xx
25	cluby	25	25	yy

- TableBox (화면 상 club 이 쓰여있는 객체): 데이터베이스에 생성된 테이블을 선택할 수 있습니다.
- 테이블 조회 버튼: TableBox 에서 선택된 테이블의 정보를 조회합니다.
- 테이블 입력 버튼: TableBox 에서 선택된 테이블에 정보를 입력합니다.
- 테이블 변경 버튼: TableBox 에서 선택된 테이블의 정보를 변경합니다.
- 테이블 삭제 버튼: TableBox 에서 선택된 테이블의 정보를 삭제합니다.
- DB 초기화 버튼: 데이터베이스를 초기화하고, 샘플 데이터를 구성합니다.

4-3. 교수 메뉴 창

교수 메뉴

교수 번호 입력	연도/학기 입력	담당 강좌 조회	지도 학생	소속 학과	강의 시간표	성적 입력
lecture	day	period	unit	course_time	major	
1	봄	1	1	3	1	

- 교수 번호 입력 버튼: 교수 메뉴의 다른 기능에 필요한 교수 번호 정보를 입력합니다.
- 연도/학기 입력 버튼: 교수 메뉴의 다른 기능에 필요한 연도/학기 정보를 입력합니다.
- 담당 강좌 조회 버튼: 입력된 교수 번호가 입력된 연도/학기에 담당하는 강좌 정보를 조회합니다. 조회된 강좌 정보 중 한 row 를 선택하면 그 강좌를 수강하는 학생들의 정보를 조회합니다.
- 지도 학생 버튼: 입력된 교수 번호가 지도하고 있는 학생의 정보를 조회합니다. 조회된 학생 정보 중 한 row 를 선택하면 그 학생의 강좌 수강 내역 정보를 조회합니다.
- 소속 학과 버튼: 입력된 교수 번호가 소속된 학과의 정보를 조회합니다.
- 강의 시간표 버튼: 입력된 교수 번호가 입력된 연도/학기에 담당한 강좌들을 시간표 형태로 조회합니다.
- 성적 입력 버튼: 입력된 교수 번호가 담당한 강좌를 수강하는 학생의 성적 정보를 입력합니다.

4-4. 학생 메뉴 창

교시	월	화	수	목	금
1					
2	lecturea				
3	lecturea				
4	lecturea				
5					
6					
7					
8					
9					
10					
11					
12					

- 학생 번호 입력 버튼: 학생 메뉴의 다른 기능에 필요한 학생 번호 정보를 입력합니다.
- 연도/학기 입력 버튼: 학생 메뉴의 다른 기능에 필요한 연도/학기 정보를 입력합니다.
- 시간표 버튼: 입력된 학생 번호가 입력된 연도/학기에 수강한 강의들을 시간표 형태로 조회합니다.
- 수강 과목 버튼: 입력된 학생 번호가 수강한 과목의 정보를 조회합니다
- 수강 내역 신청 버튼: 입력된 학생 번호가 입력된 연도/학기에 신규로 과목을 수강 신청합니다. 수강 과목 학점은 한 학기당 10학점으로 제한하며, 해당 버튼을 눌렀을 때 입력된 학생 번호가 등록금을 전부 납부했는지 확인한 뒤 납부하지 않으면 에러 메세지 창을 띄우고, 납부했다면 정상적으로 정보를 입력합니다.

- 동아리 버튼: 입력된 학생 번호가 소속된 동아리 정보를 조회합니다. 해당 학생이 동아리 회장인 경우, 동아리에 속한 모든 학생들의 정보를 조회합니다.
- 성적 버튼: 입력된 학생 번호가 수강한 과목에서 받은 성적을 조회합니다.

5. 테스트 방법

- 프로그램 실행
- 관리자 버튼 클릭
 - DB 초기화 버튼 클릭
 - 데이터베이스 테이블마다 입력 버튼 클릭
 - 테이블마다 필요한 속성 값들에 올바른 값 입력 후 OK 버튼 클릭
 - ⇒ 테이블 조회하여 입력 완료 확인
 - 테이블마다 필요한 속성 값들에 올바르지 않은 값 작성 후 OK 버튼 클릭
 - ⇒ 입력되지 않고 입력 값이 잘못 되었다는 오류 메시지 창 확인
 - Cancel 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 데이터베이스 테이블마다 변경 버튼 클릭
 - 테이블마다 올바른 조건식을 입력 후 OK 버튼 클릭
 - ⇒ 테이블 조회하여 변경 완료 확인
 - 테이블마다 올바르지 않은 조건식을 입력 후 OK 버튼 클릭
 - ⇒ 변경되지 않고 입력 값이 잘못 되었다는 오류 메시지 창 확인
 - Cancel 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 데이터베이스 테이블마다 삭제 버튼 클릭
 - 테이블마다 올바른 조건식을 입력 후 OK 버튼 클릭
 - ⇒ 테이블 조회하여 삭제 완료 확인
 - 테이블마다 올바르지 않은 조건식을 입력 후 OK 버튼 클릭
 - ⇒ 삭제되지 않고 입력 값이 잘못 되었다는 오류 메시지 창 확인
 - Cancel 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인

- 데이터베이스 테이블마다 조회 버튼 클릭
 - ⇒ 테이블이 정상적으로 조회되는 것 확인
- 관리자 메뉴 창 닫기
- 교수 버튼 클릭
 - 교수 번호 입력 버튼 클릭
 - 올바른 교수 번호를 입력 후 OK 버튼 클릭
 - ⇒ 입력이 완료되었다는 메시지 창 확인
 - 올바르지 않은 교수 번호를 입력 후 OK 버튼 클릭
 - ⇒ 입력 값이 잘못되었다는 오류 메시지 창 확인
 - 존재하지 않는 교수 번호를 입력 후 OK 버튼 클릭
 - ⇒ 존재하지 않는 교수 번호라는 오류 메시지 창 확인
 - CANCEL 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 연도/학기 입력 버튼 클릭
 - 올바른 연도/학기를 입력 후 OK 버튼 클릭
 - ⇒ 입력이 완료되었다는 메시지 창 확인
 - 올바르지 않은 연도/학기를 입력 후 OK 버튼 클릭
 - ⇒ 입력 값이 잘못되었다는 오류 메시지 창 확인
 - CANCEL 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 담당 강좌 버튼 클릭
 - 교수 번호가 입력되지 않은 상태인 경우
 - ⇒ 교수 번호를 입력하라는 오류 메시지 창 확인
 - 연도/학기가 입력되지 않은 상태인 경우
 - ⇒ 연도/학기를 입력하라는 오류 메시지 창 확인
 - 교수 번호, 연도/학기가 입력된 상태인 경우
 - ⇒ 입력된 교수 번호가 입력된 연도/학기에 담당하고 있는 강좌의 정보를 확인
 - ⇒ 담당 강좌 정보 중 한 row 를 클릭하여 해당 강좌를 수강하는 모든 학생의 정보를 확인

- 지도 학생 버튼 클릭
 - 교수 번호가 입력되지 않은 상태인 경우
 - ⇒ 교수 번호를 입력하라는 오류 메시지 창 확인
 - 교수 번호가 입력된 상태인 경우
 - ⇒ 교수 번호의 지도 학생 정보를 확인
 - ⇒ 지도 학생 정보 중 한 row 를 클릭하여 해당 학생이 수강했던 내역을 확인
- 소속 학과 버튼 클릭
 - 교수 번호가 입력되지 않은 상태인 경우
 - ⇒ 교수 번호를 입력하라는 오류 메시지 창 확인
 - 교수 번호가 입력된 상태인 경우
 - ⇒ 교수 번호의 소속 학과 정보를 확인
- 강의 시간표 버튼 클릭
 - 교수 번호가 입력되지 않은 상태인 경우
 - ⇒ 교수 번호를 입력하라는 오류 메시지 창 확인
 - 연도/학기가 입력되지 않은 상태인 경우
 - ⇒ 연도/학기를 입력하라는 오류 메시지 창 확인
 - 교수 번호, 연도/학기가 입력된 상태인 경우
 - ⇒ 입력된 교수 번호의 입력된 연도/학기 강의 시간표 정보를 확인
- 성적 입력 버튼 클릭
 - 교수 번호가 입력되지 않은 상태인 경우
 - ⇒ 교수 번호를 입력하라는 오류 메시지 창 확인
 - 교수 번호가 입력된 상태인 경우
 - 속성 값들에 올바른 값 입력 후 OK 버튼 클릭
 - ⇒ 성적 입력이 완료되었다는 메시지 창 확인
 - 속성 값들에 올바르지 않은 값 작성 후 OK 버튼 클릭
 - ⇒ 입력되지 않고 입력 값이 잘못 되었다는 오류 메시지 창 확인
 - Cancel 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인

- 교수 메뉴 창 닫기
- 학생 버튼 클릭
 - 학생 번호 입력 버튼 클릭
 - 올바른 학생 번호를 입력 후 OK 버튼 클릭
 - ⇒ 입력이 완료되었다는 메시지 창 확인
 - 올바르지 않은 학생 번호를 입력 후 OK 버튼 클릭
 - ⇒ 입력 값이 잘못되었다는 오류 메시지 창 확인
 - 존재하지 않는 학생 번호를 입력 후 OK 버튼 클릭
 - ⇒ 존재하지 않는 학생 번호라는 오류 메시지 창 확인
 - CANCEL 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 연도/학기 입력 버튼 클릭
 - 올바른 연도/학기를 입력 후 OK 버튼 클릭
 - ⇒ 입력이 완료되었다는 메시지 창 확인
 - 올바르지 않은 연도/학기를 입력 후 OK 버튼 클릭
 - ⇒ 입력 값이 잘못되었다는 오류 메시지 창 확인
 - CANCEL 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 시간표 버튼 클릭
 - 학생 번호가 입력되지 않은 상태인 경우
 - ⇒ 학생 번호를 입력하라는 오류 메시지 창 확인
 - 연도/학기가 입력되지 않은 상태인 경우
 - ⇒ 연도/학기를 입력하라는 오류 메시지 창 확인
 - 학생 번호, 연도/학기가 입력된 상태인 경우
 - ⇒ 입력된 학생 번호의 입력된 연도/학기의 시간표 정보 확인
- 수강 과목 버튼 클릭
 - 학생 번호가 입력되지 않은 상태인 경우
 - ⇒ 학생 번호를 입력하라는 오류 메시지 창 확인

- 연도/학기가 입력되지 않은 상태인 경우
 - ⇒ 연도/학기를 입력하라는 오류 메시지 창 확인
- 학생 번호, 연도/학기가 입력된 상태인 경우
 - ⇒ 입력된 학생 번호의 입력된 연도/학기의 수강 과목 정보 확인
- 수강 내역 신규 삽입 버튼 클릭
 - 학생 번호가 입력되지 않은 상태인 경우
 - ⇒ 학생 번호를 입력하라는 오류 메시지 창 확인
 - 연도/학기가 입력되지 않은 상태인 경우
 - ⇒ 연도/학기를 입력하라는 오류 메시지 창 확인
 - 학생 번호, 연도/학기가 입력된 상태인 경우
 - 등록금 납부가 되어있지 않은 경우
 - ⇒ 등록금이 미납되었다는 오류 메시지 창 확인
 - 입력된 학생 번호, 연도/학기의 총 수강 학점과 신청하려는 강의의 학점의 합이 10을 초과하는 경우
 - ⇒ 한 학기에 10 학점을 초과해서 신청할 수 없다는 오류 메시지 창 확인
 - 위 두 조건에 해당하지 않는 경우
 - 속성 값들에 올바른 값 입력 후 OK 버튼 클릭
 - ⇒ 입력이 완료되었다는 메시지 창 확인
 - 속성 값들에 올바르지 않은 값 작성 후 OK 버튼 클릭
 - ⇒ 입력되지 않고 입력 값이 잘못 되었다는 오류 메시지 창 확인
 - Cancel 버튼 클릭
 - ⇒ 아무 행위하지 않고 입력 창이 닫힘을 확인
- 동아리 버튼 클릭
 - 학생 번호가 입력되지 않은 상태인 경우
 - ⇒ 학생 번호를 입력하라는 오류 메시지 창 확인
 - 학생 번호가 입력된 상태인 경우
 - 학생 번호가 동아리 회장으로 속한 동아리가 있는 경우
 - 학생 번호가 속한 동아리 정보를 왼편에서 확인

- 학생 번호가 동아리 회장으로 속한 동아리의 구성원 정보를 오른쪽에서 확인
 - 학생 번호가 동아리 회장으로 속한 동아리가 없는 경우
 - 학생 번호가 속한 동아리 정보를 확인
- 성적 버튼 클릭
 - 학생 번호가 입력되지 않은 상태인 경우
 - ⇒ 학생 번호를 입력하라는 오류 메시지 창 확인
 - 학생 번호가 입력된 상태인 경우
 - ⇒ 학생 번호의 수강 내역 정보와 마지막 줄의 GPA 정보 확인
- 학생 메뉴 창 닫기
- 프로그램 종료