

학원 학생 관리 프로그램

multicampus

제출일	2021.12.10		
과목	데이터베이스		
담당교수	정현희, 소인록	이름	정태환, 임주영, 김희태, 최우혁

목차

1. 서론

1.1 주제선정 및 배경	4
---------------------	---

2. 계획

2.1 시나리오 및 문제의 정의	5
2.2 일정계획	5
2.3 노력 추정	7
2.4 조직 계획	8

3. 요구 분석

3.1 요구 사항	9
3.2 구조적 분석	10

4. 설계

4.1 화면 설계	11
4.2 객체지향 분석과 설계	12
4.2.1 순서 다이어그램	12
4.2.2 ER 다이어그램	13

5. 구현

5.1 소스코드	14
----------------	----

표 목차

표 1. 학생 관리 프로그램 일정 계획	6
표 2. 소요 예산	7
표 3. 인건비 산정 내역	7
표 4. 조직의 업무 기능	8

그림 목차

그림 1. 작업 분해 (WBS)	5
그림 2. 간트차트	6
그림 3. 조직의 구성	8
그림 4. ERD 설계	10
그림 5. 화면설계	11
그림 6. 순서 다이어그램	12
그림 7. ER 다이어그램	13

■ 1. 서론

1.1 주제선정 및 배경

학생 관리 프로그램은 학교, 학원 등에서의 사무를 컴퓨터 처리 시스템으로 만든 것이다. 좋은 학원의 요건 중 잘 가르치는 것과 학원의 환경이 있겠지만, 학생을 잘 관리한다는 것 또한 하나의 요소가 될 것이다. 작은 규모에 학원이라면 이러한 프로그램이 필요치 않겠지만 규모가 있는 학원에서의 학생의 정보를 잘 다루는 것은 필수요건이기도 하다.

학생 관리 프로그램을 이용하면 학원 운영에 있어서 들게 되는 많은 시간과 비용(인력)을 절약할 수 있다는 점에서 큰 장점이 있다. 따라서 학생관리프로그램을 간편하게 이용하여 원생관리에 도움을 줄 수 있는 학원 원생관리프로그램을 주제로 선정하였다.

■ 2. 계획

2.1 시나리오 및 문제의 정의

관리자 입장에서 학생들에 대한 데이터베이스를 관리 할 수 있는 학생 관리 업무시스템을 의뢰받아 제작하였다. 학생 뿐 아니라 교사들에 대한 정보도 추가하고 확인, 수정할 수 있도록 한다.

2.2 일정계획

2.2.1 작업 분해

전체 프로젝트의 일정을 예측하기 위하여 소작업들로 분류하였다. WBS를 작성함으로써, 프로젝트 진행에서 일어나는 모든 작업을 찾고 큰 단위의 일을 관리하기 쉬운 작은 일로 나누기 위하여 작성되었다.



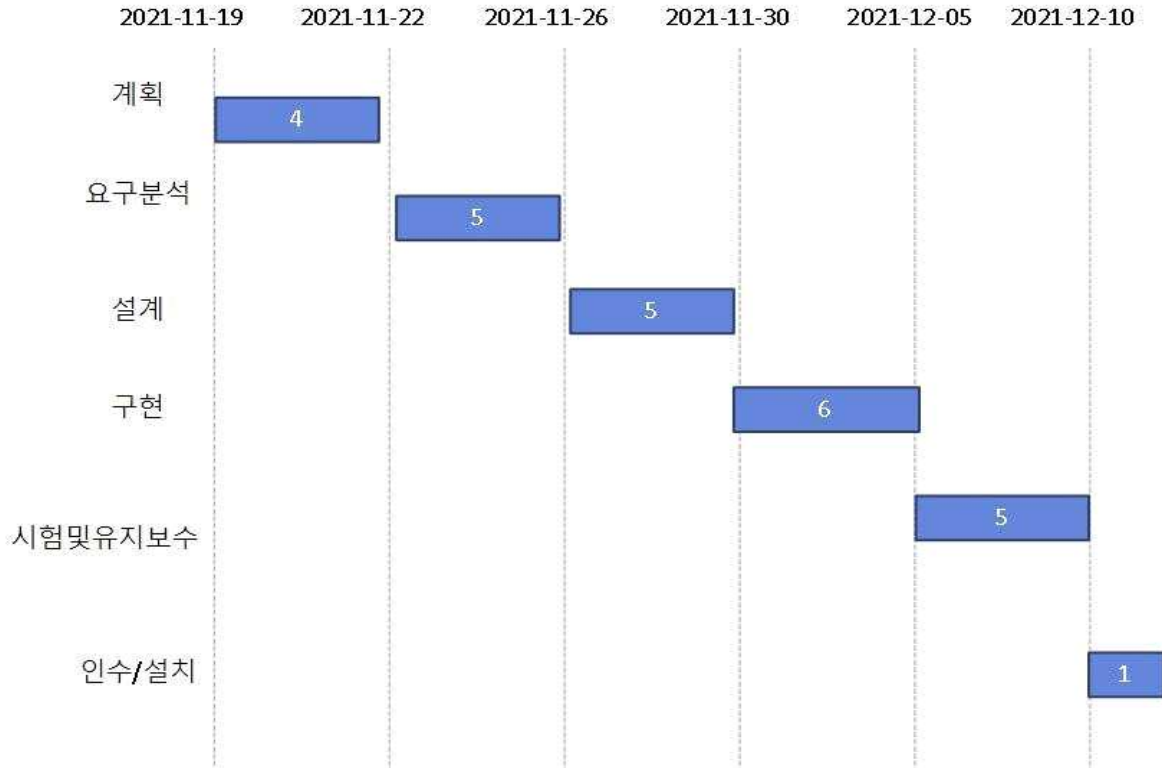
<그림 1. 작업 분해 (WBS)>

2.2.2 간트 차트

간트 차트(Gantt chart)는 프로젝트 일정 관리를 위한 바(bar) 형태의 도구이다. 프로젝트의 주요 활동을 파악한 후, 각 활동의 일정을 시작하는 시점과 끝나는 시점을 연결한 막대 모양으로 표시하여 전체 일정을 한눈에 볼 수 있다. 아래처럼 활동은 막대로 표시하고 그 길이는 활동의 작업 시간과 비례한다. 프로젝트를 이루는 소작업 별로 언제 시작되고 언제 끝나야 하는지를 한눈에 볼 수 있도록 프로젝트 일정표를 그렸다.

총 개발 기간	1개월 2021.11.19. ~ 2021.12.10	
소 작업	학원 학생 관리 프로그램	
소 작업별 개발 기간	소작업	개발 기간
	계획	11월 19일 ~ 11월 22일
	요구 분석	11월 22일 ~ 11월 26일
	설계	11월 26일 ~ 11월 30일
	구현	11월 30일 ~ 12월 05일
	시험 및 유지보수	12월 05일 ~ 12월 09일
	발표	12월 10일
개발 순서	계획 → 요구 분석 → 설계 → 구현 → 시험 및 유지 보수	
필요 자원	개발용 PC, 개발 공간	

<표 1. 학생 관리 프로그램 일정 계획>



<그림 2. 간트차트>

2.3 노력추정

학생 관리 프로그램을 개발하기 위하여 드는 비용이 얼마이며 어느 정도의 기간이 걸리는지를 추정할 필요가 있다. 소프트웨어 개발 비용의 대부분은 소요되는 인력 자원이기 때문에 대부분의 비용 예측은 월 인원(Man-Month)으로 표현하는 노력의 추정에 초점을 두고 있다.

소요내역	세부내역	산출금액
① 인건비	별첨 참조	7,314,408
② 재경비	① × 100%	7,314,408
③ 기술료	(① + ②) × 10%	1,000,000
④ 직접경비	회의비: 10,000원×4명×7회	280,000
⑤ VAT	(①+②+③+④)×10%	859,440
총 계		8,594,408

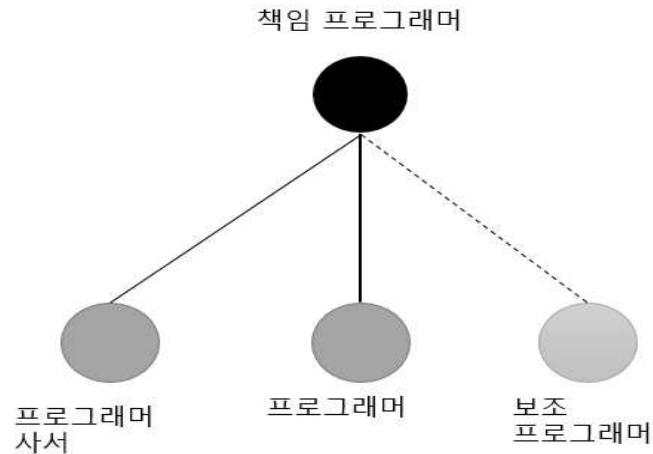
<표 2. 소요 예산>

구분	인원	개월	일/월	참여율	단 가	금액
고급기술사	1		18	80%	210,654	3,791,772
중급기능사	1		18	50%	87,234	1,570,212
초급기능사	2		18	50%	54,234	1,952,424
계						7,314,408

<표 3. 인건비 산정 내역>

2.4 조직 계획

책임 프로그래머 팀 구성은 팀 구성원이 관리자의 명령을 따라 일하고 결과를 보고하는 방식이다. 학생관리 서비스는 별도의 창의력을 요하지 않기에 책임 프로그래머 팀 구성을 계획하였다. 개발기간이 단기간에 수행되어야하기에 협의와 빠른 의사결정이 필요함에 판단했다.



<그림 3. 조직의 구성>

위의 노력 추정을 통하여 고급 기능사 1명과 중급기능사 1명 초급 기능사 2명이 필요하다. 각 기능사들의 업무 기능을 아래의 표로 정리 하였다.

조직 단위	업무 기능
고급 기능사	요구 분석 및 설계, 중요한 기술적 판단, 프로그래머에게 작업 및 지시 배분
중급 기능사	고급 프로그래머의 지시에 따른 원시 코드 작성, 테스트, 디버깅, 문서 작성 등
초급 기능사	고급 프로그래머의 업무 지원, 여러 가지 기술적인 문제에 대한 자문, 사용자와 품질 보증 담당자 등의 섭외, 고급 기능사 감독 하에 분석, 설계, 구현 담당

<표 4. 조직의 업무 기능>

3. 요구 분석

3.1 요구 사항

1. 학원 운영, 관리자만이 이 프로그램을 수정, 삭제, 조회, 추가 할 수 있어야 하며 관리자의 아이디와 비밀번호가 부여되어야한다.
2. 학생의 이름, 연락처, 출신학교, 주소, 반 등의 정보가 필요하다.
3. 선생님들의 이름, 담당 과목, 담당 반, 연락처 등의 정보가 필요하며 조회,수정,삭제, 추가를 할 수 있어야 한다.
- 4 .학생정보인 이름, 연락처, 출신학교, 주소, 반을 입력하여 해당 학생의 점수를 조회 할 수 있어야 한다.
- 5 .국어, 영어, 수학, 한국사 과목의 평균 점수를 구해 자동적으로 반 배정이 가능해야 한다. 90점 이상: A반, 80점 이상:B반, 70점이상:c반 70점 미만은 D반으로 배정한다.
6. 국어,영어,수학,한국사의 점수는 0점~100점 사이여야 한다.
7. 교사정보 추가에서 교사는 담당 반이 존재하지 않고 담당 과목만 있을 수 있다. 이 경우 담당반정보가 null로 추가된다.
8. 학생, 교사정보의 전화번호는 -을 제외하고 입력되도록 하며 11자리여야한다.
9. 추가(신규)학생의 정보가 저장되는 파일이 있어야한다.
10. 학생정보 전체 조회시 학생번호, 이름, 전화번호, 주소, 출신학교, 과목별 점수, 평균, 반, 담당교사의 이름이 출력되도록 한다.
11. 학생정보추가,수정시 저장되어있는 전화번호와의 중복이 있어서는 안된다.
12. 학생, 교사정보 수정시 이름은 수정이 불가능하다.

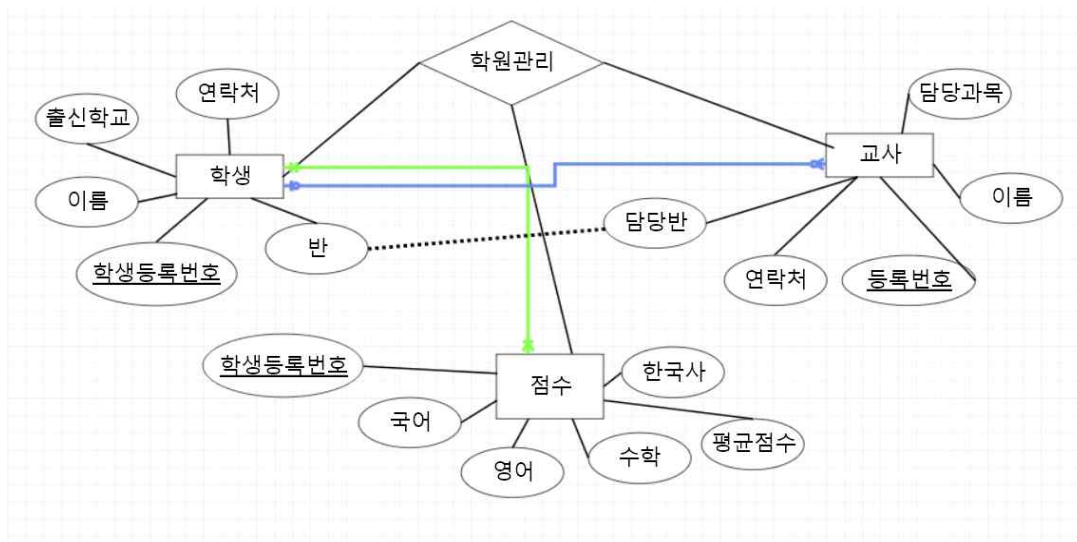
개체	Student(학생)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	등록번호	studentID	Int(Auto Increment)	int	Not Null
속성	이름	studentname	varchar	varchar(5)	Not Null
	연락처	studentnumber	varchar	varchar(15)	Not Null
	출신학교	studentSchool	varchar	varchar(20)	Not Null
	주소	studentaddr	varchar	varchar(20)	Not Null
	반	studentclass	varchar	varchar(4)	

개체	Report(점수)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키 외래키	등록번호	studentID	int(Auto Increment)	int	Not Null
속성	학생이름	studentname	varchar	varchar(5)	Not Null
	국어	korscore	int	int	Not Null
	영어	engscore	int	int	Not Null
	수학	mathscore	int	int	Not Null
	한국사	historyscore	int	int	Not Null
	평균	avgg	double	double	Not Null

개체	Teacher(교사)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	교사번호	teacherID	int(Auto Increment)	int	Not Null
속성	교사이름	teachername	varchar	varchar(5)	Not Null
	담당과목	teachersubject	varchar	varchar(10)	Not Null
	담당반	teacherclass	varchar	varchar(4)	
	연락처	teachernumber	varchar	varchar(20)	Not Null

3.2 구조적 분석

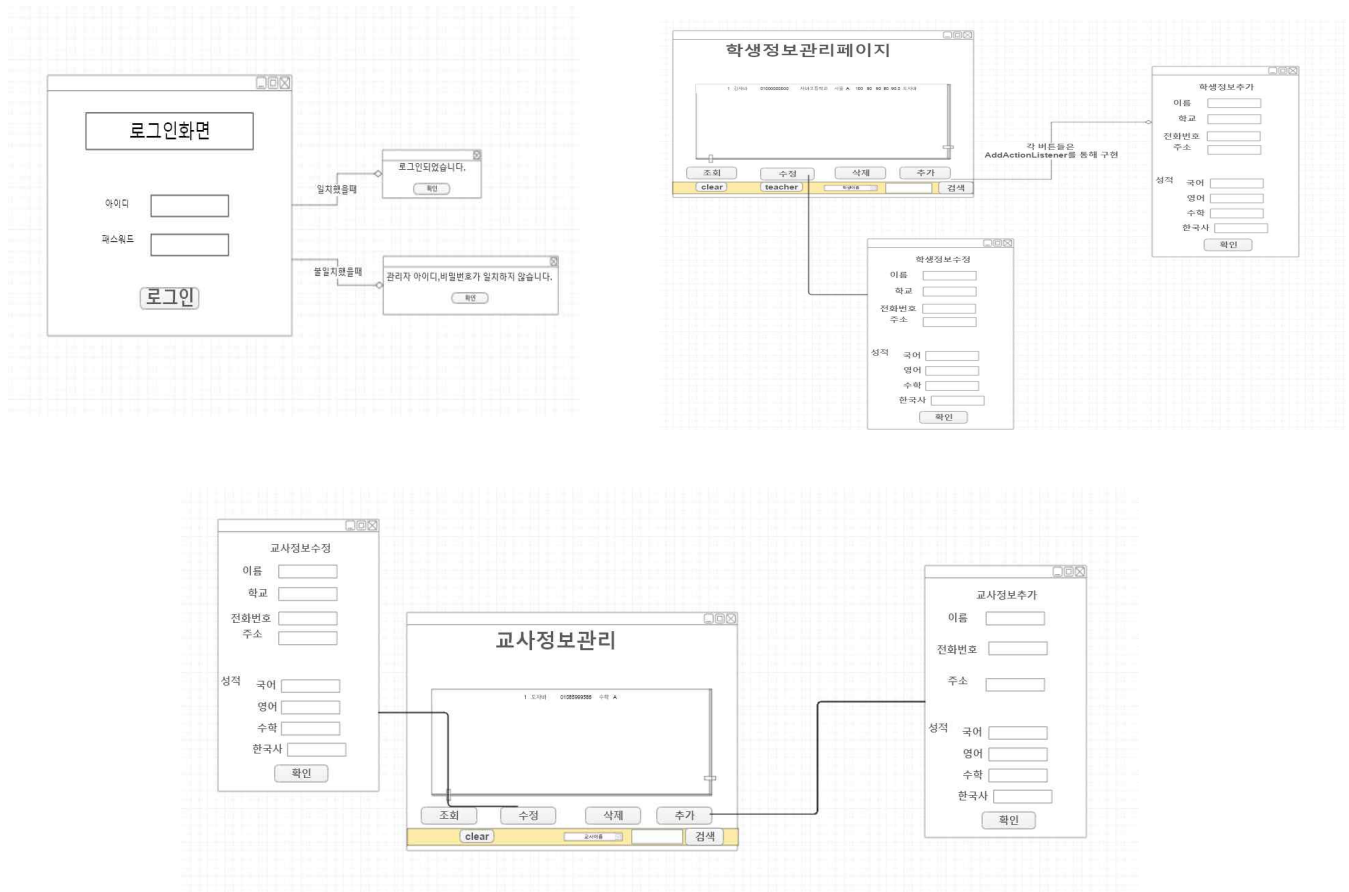
개념적설계 단계에서 ERD를 작성하였다. 학생, 교사, 점수 테이블이 구성되어 있고, 학생 테이블의 학생 등록번호로 점수 테이블과 연결하고 학생의 반과 교사의 담당반을 통해 교사 테이블을 연결할 수 있게 설계 하였다.



<그림 4. ERD 설계>

4.1 화면 설계

학원원생관리화면을 임의로 디자인하여 설계하였다. 로그인 화면을 통해 관리자 이외의 사람이 접근할 수 없게 설계하였으며 조회, 수정, 삭제, 추가 기능을 통해 학생정보를 관리할 수 있게 하였다. 또한 교사의 정보를 업데이트할 필요가 있기에 학생정보관리페이지에 교사버튼으로 교사정보관리페이지에 접근 할수 있도록 설계하였다. 학생정보관리페이지와 교사정보관리페이지를 유사하게 설계하여 사용에 어려움이 없도록 설계하였다.

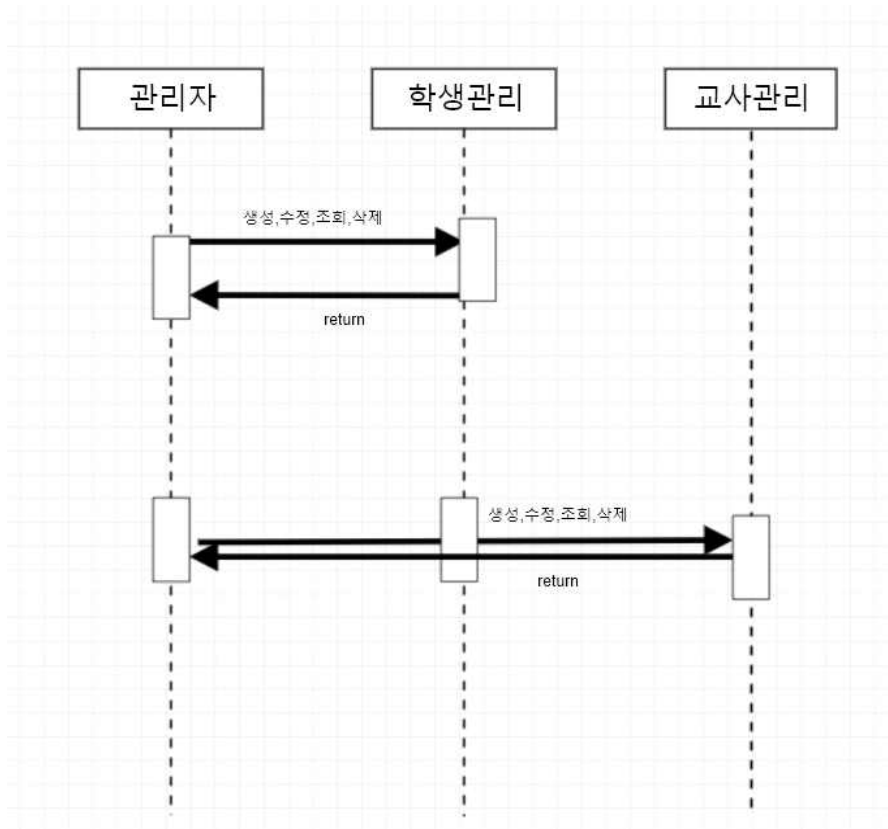


<그림 5. 시스템 화면 설계>

4.2 객체지향 분석과 설계

4.2.1 순서 다이어그램

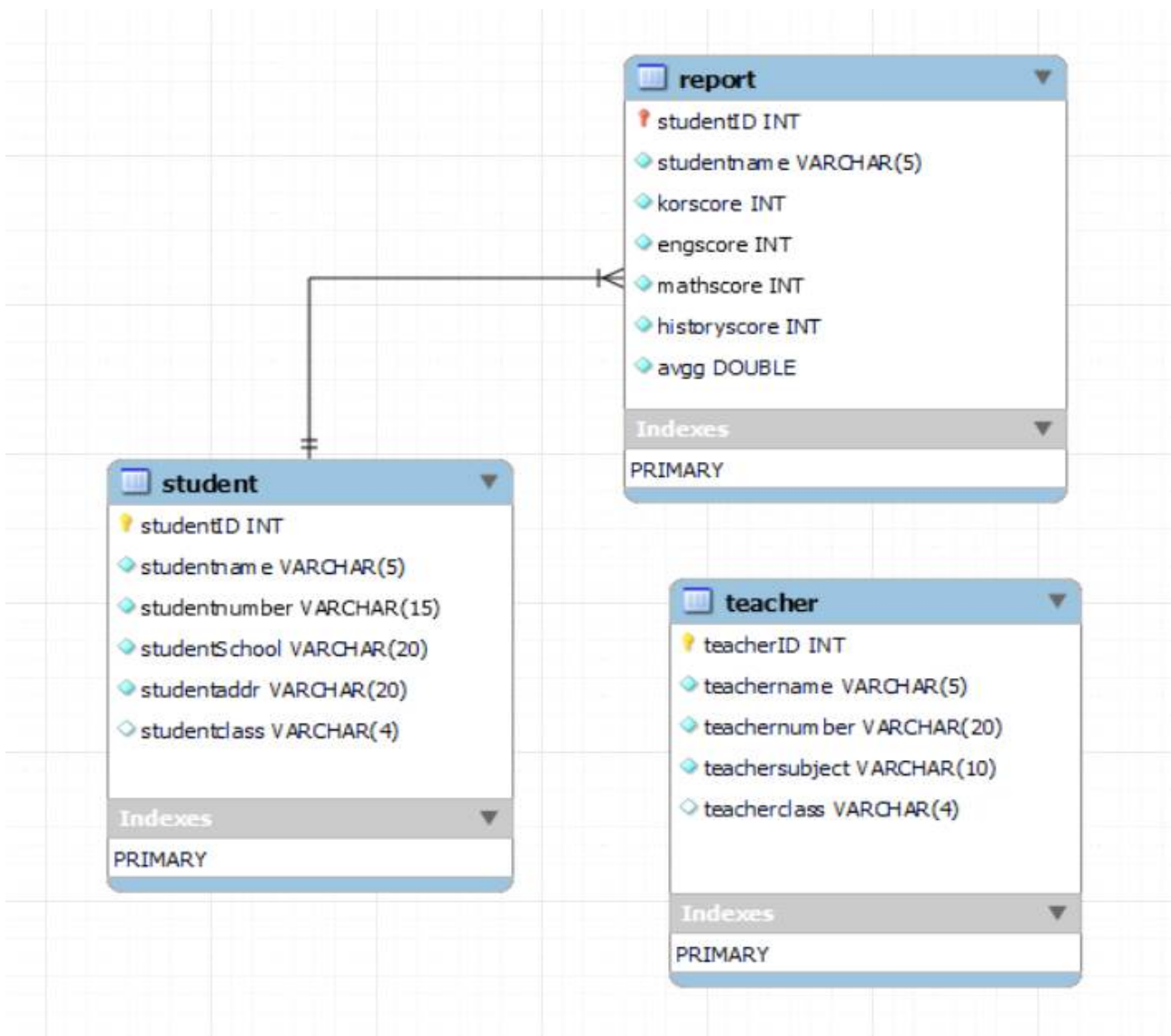
고객인 관리자가 해당 UI에서 학생관리를 하기위해 로그인을 통해 관리페이지에 접속하여 학생정보 생성, 수정 삭제, 조회를 할 수 있도록 설계하였다. 교사정보도 변동사항이 있을 수 있기에 학생관리 페이지에서 교사버튼을 통해 교사관리 페이지에 접속하여 교사정보 조회, 수정, 삭제, 생성을 할 수 있도록 설계하였다.



<그림 6. 시퀀스 다이어그램>




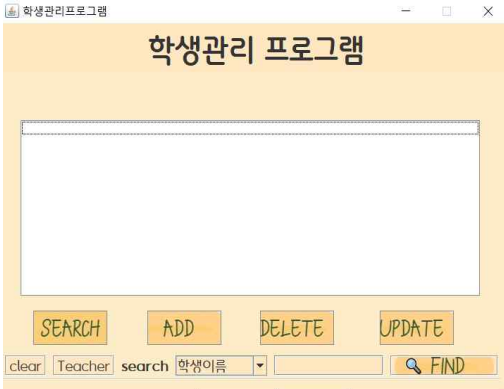
4.2.2 ER 다이어그램

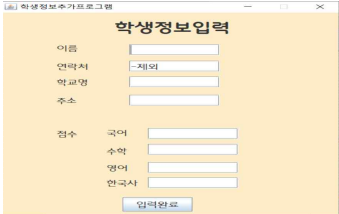

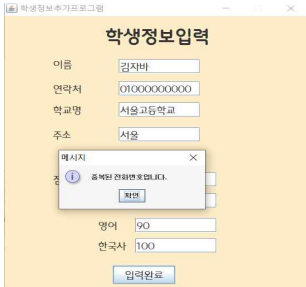
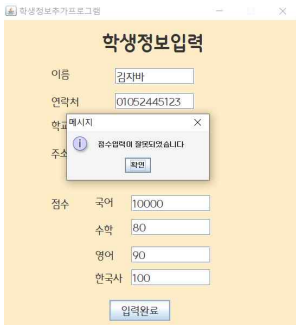
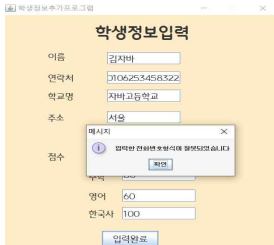
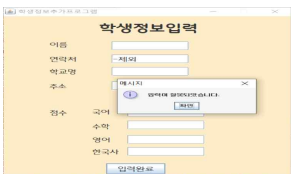
MySQL에서 정적 모델링의 표현 중 하나인 ER 다이어그램으로 시스템의 도메인 개념을 표현하는데 사용하였다. ER 다이어그램은 클래스 객체들의 공통 구조와 동작들을 추상화시킨 것이다. ER 다이어그램은 특히 도메인 개념과 속성 및 관계를 나타내는 중요한 모델이다. 아래와 같이 클래스들을 파악하고 연관 관계를 표현하였다



<그림 7. ER 다이어그램>

5. 소스 코드

	화면	설명
로그인		<ol style="list-style-type: none"> 1) : 아이디 입력 2) :비밀번호입력 3) Login: 로그인 버튼 4) 로그인되었습니다 메시지: 로그인성공 5) 아이디나 비밀번호가 틀렸습니다 메시지: 로그인실패
메인화면		<ol style="list-style-type: none"> 1) search: 전체조회화면 출력버튼 2) add: 학생정보추가버튼 3) delete: 학생정보 삭제버튼 4) update: 학생정보 수정버튼 5) clear: 콤보박스 옆 텍스트, 출력된 리스트 초기화버튼 6) Teacher: 교사정보관리페이지 이동 버튼 7) find: 콤보박스에 해당하는 텍스트 내용이 학생정보와 일치하는 것을 리스트로 출력시키는 버튼

	화면	설명
학생추가		
		1) 이름,연락처,학교명,주소, 점수-국어,수학,영어,한국사 정보 입력->입력완료 메시지출력
		2) 입력완료 클릭시 입력한 연락처가 기존정보와 중복되는 경우 '중복된 전화번호입니다' 오류메시지 출력
		3) 연락처가 11자리가 아니면 "입력된 전화번호형식이 잘못되었습니다"메시지 출력
		4) 점수-국어,수학,영어,한국사 정보가 0~100점 사이가 아니면 점수입력이 잘못되었습니다 메시지 출력
		5) 정보입력에 빈칸이있는 경우 오류메시지 출력

	화면	설명
학생삭제		1) 삭제할 데이터를 클릭 2) delete버튼 클릭 3) 클릭한 데이터가 없을 시 선택된 학생이 없습니다 메시지출력

화면

설명

학생검색

학생관리 프로그램

38	박유진	0102225555	대구고등학교
39	김민준	0102225555	대구고등학교
40	김진우	0102227777	C비고등학교
41	이민준	0102228888	C비고등학교
42	박재민	0102229999	남동고등학교
43	김수민	0103333000	C비고등학교
44	김서준	0103333111	대구고등학교
45	이민준	0103333222	C비고등학교
46	김진우	0103333444	C비고등학교
47	최준혁	0103333555	충청고등학교
48	최지훈	0103333666	충청고등학교
49	한지우	0103333777	대구고등학교

SEARCH ADD DELETE UPDATE

clear Teacher search 학생이름 FIND

학생관리 프로그램

학번번호	학생이름	학생전화번호	학교이름
1	김지민	0100000000	자비고등학교

SEARCH ADD DELETE UPDATE

clear Teacher search 학생이름 김지민 FIND

학생관리 프로그램

학번번호	학생이름	학생전화번호	학교이름
10	이도환	0104444444	충청고등학교

SEARCH ADD DELETE UPDATE

clear Teacher search 학생전화번호 0104444444 FIND

학생관리 프로그램

학번번호	학생이름	학생전화번호	학교이름
1	김지민	0100000000	자비고등학교
7	김서준	0101111111	자비고등학교
8	김진우	0102222222	자비고등학교
9	김수민	0102222222	자비고등학교
17	이민준	0100002222	자비고등학교
24	이민준	0100009999	자비고등학교
28	이민준	0101111555	자비고등학교
31	이민준	0101117777	자비고등학교
32	이민준	0102228888	자비고등학교
41	이민준	0102228888	자비고등학교
43	이민준	0102228888	자비고등학교

SEARCH ADD DELETE UPDATE

clear Teacher search 출신학교 자비고등학교 FIND

학생관리 프로그램

학번번호	학생이름	학생전화번호	학교이름
1	김지민	0100000000	자비고등학교
6	김지민	0100004444	자비고등학교
7	김서준	0101111111	자비고등학교
8	김진우	0102222222	자비고등학교
9	김수민	0102222222	자비고등학교
18	이민준	0100004444	자비고등학교
20	최준혁	0100005555	대구고등학교
21	최지훈	0100009999	C비고등학교
31	이민준	0101117777	자비고등학교
32	이민준	0101118888	자비고등학교
33	이민준	0102220000	자비고등학교

SEARCH ADD DELETE UPDATE

clear Teacher search 학생주소 서울 FIND

학생관리 프로그램

학번번호	학생이름	학생전화번호	학교이름	성	성
0100000000	김지민	0100000000	자비고등학교	김	지민
0100000111	김서준	0101111111	자비고등학교	김	서준
0100004444	김진우	0102222222	자비고등학교	김	진우
0100005555	김수민	0102222222	자비고등학교	김	수민
0100007777	이민준	0100002222	자비고등학교	이	민준
0100009999	이민준	0100009999	자비고등학교	이	민준
0101111555	이민준	0101117777	자비고등학교	이	민준
0101118888	이민준	0101118888	자비고등학교	이	민준
0102220000	이민준	0102220000	자비고등학교	이	민준





SEARCH ADD DELETE UPDATE

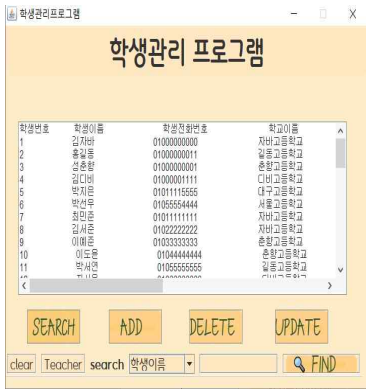
clear Teacher search 성 지민 FIND

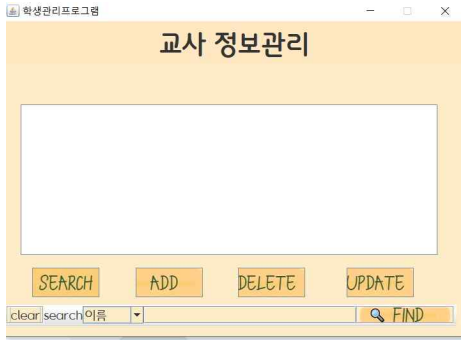
1) 검색 필드 선택

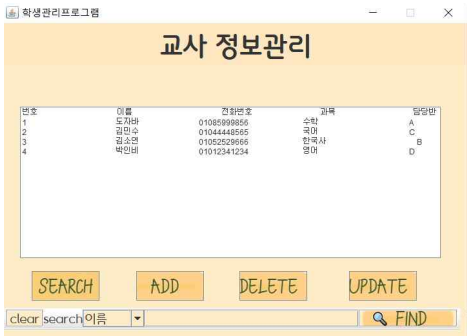
(학생이름,학생전화번호,출신학교,학생 주소,반 검색 가능)




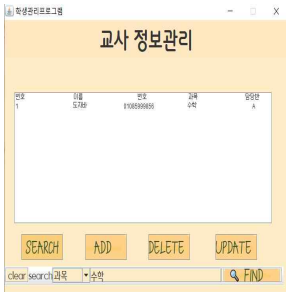

2) 검색 -> 결과에 해당하는 리스트 출력

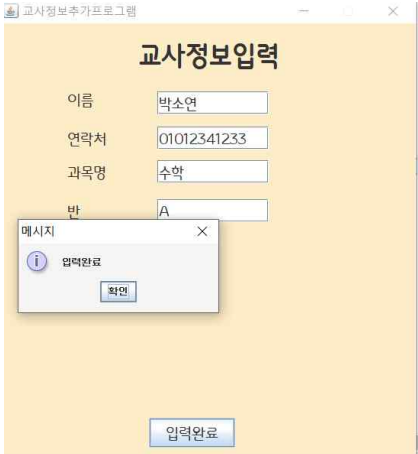
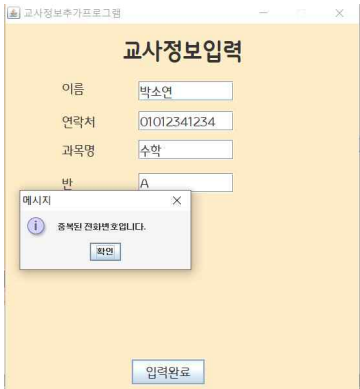

	화면	설명
학생 정보 수정		
		<p>1) 메인화면에서 수정할 데이터 선택한 후 update버튼 클릭시 해당 창이 뜸</p>
		<p>2)메인화면에서 클릭한 데이터가 없을 시 선택된 학생이 없습니다 메시지출력</p> <p>3)수정완료버튼 클릭시 수정완료메시지 출력</p>
		

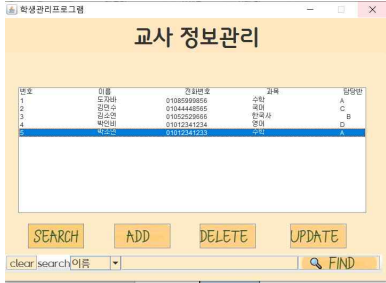
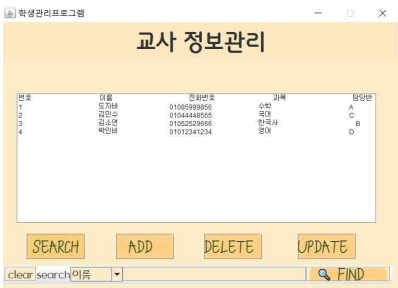

	화면	설명
학생전 체조회		<p>1) search버튼클릭</p> <p>2) 학생번호,이름,학생전화번호, 출신학교, 주소, 반, 국어점수, 수학점수, 영어점수, 한국사점수, 평균점수 리스트로 출력</p>





	화면	설명
교사관리 리페이지(메인)		<ol style="list-style-type: none"> 1) search: 전체조회화면 출력버튼 2) add: 교사정보추가버튼 3) delete: 교사정보 삭제버튼 4) update: 교사정보 수정버튼 5) clear: 콤보박스 옆 텍스트, 출력된 리스트 초기화버튼 6) find: 콤보박스에 해당하는 텍스트 내용이 교사정보와 일치하는 것을 리스트로 출력시키는 버튼

	화면	설명
교사전 체조회		<p>1) search버튼클릭</p> <p>2) 번호, 이름, 전화번호, 과목, 담당반 내용 리스트로 출력</p>

	화면	설명
교사검색		
		
		1) 검색 필드 선택 (교사이름,교사전화번호,과목,담당반 검색 가능)
		2) 텍스트에 검색필드에 해당하는 검색하고싶은 값 검색 -> 결과에 해당하는 리스트 출력
		

	화면	설명
교사추가	  	<p>1)이름,연락처,과목, 담당반 정보 입력->입력완료 메시지출력</p> <p>2)입력완료 클릭시 입력한 연락처가 기존정보와 중복되는 경우 '중복된 전화번호입니다.' 오류메시지 출력</p> <p>3)연락처가 11자리가 아니면 '입력된 전화번호형식이 잘못되었습니다.' 메시지 출력</p>

	화면	설명
교사삭제		
		
		1)삭제할 데이터를 클릭 2)delete버튼 클릭->삭제 3) 클릭한 데이터가 없을 시 선택된 학생이 없습니다 메시지출력

	화면	설명
교사 정보 수정		
		<p>1) 메인화면에서 수정할 데이터 선택한 후 update버튼 클릭시 해당 창이 뜨며 선택한 데이터내용이 창에 같이 뜨게된다.</p>
		<p>2)메인화면에서 클릭한 데이터가 없을 시 선택된 학생이 없습니다 메시지출력</p>
		<p>3)수정완료버튼 클릭시 수정완료메시지 출력 ->수정</p>

소스 코드

StuDAO.java

1. 오라클 연동

```
public class StuDAO {
    String driver="com.mysql.cj.jdbc.Driver";
    String url="jdbc:mysql://localhost:3306/project";
    String userid="root";
    String passwd="1234";
    Connection conn=null;
    Statement stmt=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;

    JPanel panel_addinfo;
    StuDAO(){
        try {
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } //드라이버로딩
    }
}
```

2. 학생전화번호 중복체크

```
String studentnumber;
    int cnt=0;
    boolean result=true;
    public boolean checknum(String stunum) {
        Connection conn=null;
        PreparedStatement pstmt=null;

        try {
            conn=DriverManager.getConnection(url, userid, passwd);
            String query="select * from student where studentnumber=?";
            pstmt=conn.prepareStatement(query);
            pstmt.setString(1, stunum);
            rs=pstmt.executeQuery();

            while(rs.next()) {

                cnt++;

            }
            if(cnt==1) {
                JOptionPane.showMessageDialog(null, "중복된 전화번호입니다.");

                result=false;
            }

        } catch (SQLException e) {

            e.printStackTrace();
        } finally {
            try {
                pstmt.close();
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        return result;
    }
}
```

3. 학생정보 전체출력

```
public ArrayList<StuDTO> select(){
    ArrayList<StuDTO> list=new ArrayList<StuDTO>();

    Connection conn=null;
    Statement stmt=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="select
s.studentID,s.studentname,s.studentnumber,s.studentSchool,s.studentaddr,s.studentclass,r
.korscore,r.mathscore,r.engscore,r.historyscore,r.avgg,t.teachername from student s left
outer join report1 r on s.studentname=r.studentname and s.studentID=r.studentID left
outer join teacher t on s.studentclass=t.teacherclass";
        pstmt=conn.prepareStatement(query);
        rs=pstmt.executeQuery();
        while(rs.next()) {
            StuDTO dto=new StuDTO();
            dto.setStudentID(rs.getInt("studentID"));
            dto.setStudentname(rs.getString("studentname"));
            dto.setStudentnumber(rs.getString("studentnumber"));
            dto.setStudentschool(rs.getString("studentSchool"));
            dto.setStudentaddr(rs.getString("studentaddr"));
            dto.setStudentclass(rs.getString("studentclass"));
            dto.setKorscore(rs.getInt("korscore"));
            dto.setMathscore(rs.getInt("mathscore"));
            dto.setEngscore(rs.getInt("engscore"));
            dto.setHistoryscore(rs.getInt("historyscore"));
            dto.setAvgg(rs.getDouble("avgg"));
            dto.setTeachername(rs.getString("teachername"));

            list.add(dto);

        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            rs.close();
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    return list;
}
```

4. 학생정보 insert/ 성적정보 insert

```
//학생정보추가
public void insertstu(String studentname, String studentnumber, String studentSchool,
String studentaddr, String stuclass) {
    Connection conn=null;
    PreparedStatement pstmt=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="insert into
student(studentname,studentnumber,studentSchool,studentaddr,studentclass)
values(?,?,?,?,?)";

        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, studentname);
        pstmt.setString(2, studentnumber);
        pstmt.setString(3, studentSchool);
        pstmt.setString(4, studentaddr);
        pstmt.setString(5, stuclass);

        int n=pstmt.executeUpdate();
        if(n==1) System.out.println("insert success");

    } catch (SQLException e) {
        e.printStackTrace();
    }finally {

        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }

    }
}
```

```

//점수부분 추가
public void insertscore(String studentname,String korscore,String mathscore,String engscore,
String historyscore,String avgg) {
    Connection conn=null;
    PreparedStatement pstmt=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="insert into
report1(studentname,korscore,mathscore,engscore,historyscore,avgg) values(?,?,?,?,?,?)";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, studentname);
        pstmt.setInt(2, Integer.parseInt(korscore));
        pstmt.setInt(3, Integer.parseInt(mathscore));
        pstmt.setInt(4, Integer.parseInt(engscore));
        pstmt.setInt(5, Integer.parseInt(historyscore));
        pstmt.setDouble(6,Double.parseDouble(avgg));

        int n=pstmt.executeUpdate();
        if(n==1) {JOptionPane.showMessageDialog(null, "입력완료");
        System.out.println("insert success");
        }
    }catch(SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}

```

5. 학생정보삭제: Delete

```

public void delete(int studentID) {
    Connection conn=null;
    PreparedStatement pstmt=null;
    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="delete from student where studentID=?";
        pstmt=conn.prepareStatement(query);
        pstmt.setInt(1,studentID);
        int n=pstmt.executeUpdate();
        if(n==1) System.out.println("삭제완료");
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}

```

6. 학생정보 콤보박스 조건에 따른 내용 검색: select

```
public ArrayList<StuDTO> search(String column, String item){
    ArrayList<StuDTO> list=new ArrayList<StuDTO>();//객체생성
    Connection conn=null;
    Statement stmt=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;
    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        switch(column) {
            case "studentname": String query="select
s.studentID,s.studentname,s.studentnumber,s.studentSchool,s.studentaddr,s.studentclass,r.korscore
,r.mathscore,r.engscore,r.historyscore,r.avgg from student s left outer join report1 r on
s.studentID=r.studentID WHERE s.studentname=?"; pstmt=conn.prepareStatement(query); break;
            case "studentnumber": query="select
s.studentID,s.studentname,s.studentnumber,s.studentSchool,s.studentaddr,s.studentclass,r.korscore
,r.mathscore,r.engscore,r.historyscore,r.avgg from student s left outer join report1 r on
s.studentID=r.studentID WHERE s.studentnumber=?";pstmt=conn.prepareStatement(query); break;
            case "studentSchool": query="select
s.studentID,s.studentname,s.studentnumber,s.studentSchool,s.studentaddr,s.studentclass,r.korscore
,r.mathscore,r.engscore,r.historyscore,r.avgg from student s left outer join report1 r on
s.studentID=r.studentID WHERE s.studentSchool=?";pstmt=conn.prepareStatement(query); break;
            case "studentaddr": query="select
s.studentID,s.studentname,s.studentnumber,s.studentSchool,s.studentaddr,s.studentclass,r.korscore
,r.mathscore,r.engscore,r.historyscore,r.avgg from student s left outer join report1 r on
s.studentID=r.studentID WHERE s.studentaddr=?";pstmt=conn.prepareStatement(query); break;
            case "studentclass": query="select
s.studentID,s.studentname,s.studentnumber,s.studentSchool,s.studentaddr,s.studentclass,r.korscore
,r.mathscore,r.engscore,r.historyscore,r.avgg from student s left outer join report1 r on
s.studentID=r.studentID WHERE s.studentclass=?";pstmt=conn.prepareStatement(query); break;}
        stmt=conn.createStatement();
        pstmt.setString(1, item);
        rs=pstmt.executeQuery();
        while(rs.next()) {
            StuDTO dto=new StuDTO();//객체생성
            dto.setStudentID(rs.getInt("studentID"));
            dto.setStudentname(rs.getString("studentname"));
            dto.setStudentnumber(rs.getString("studentnumber"));
            dto.setStudentschool(rs.getString("studentSchool"));
            dto.setStudentaddr(rs.getString("studentaddr"));
            dto.setStudentclass(rs.getString("studentclass"));
            dto.setKorscore(rs.getInt("korscore"));
            dto.setMathscore(rs.getInt("mathscore"));
            dto.setEngscore(rs.getInt("engscore"));
            dto.setHistoryscore(rs.getInt("historyscore"));
            dto.setAvgg(rs.getDouble("avgg"));
            list.add(dto);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            rs.close();
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return list;}
```


7. 학생정보 수정/ 점수정보 수정: Update

```

public void updatechangestu(String studentname, String studentnumber, String studentSchool,
String studentaddr,
        String studentclass) {
    Connection conn=null;
    PreparedStatement pstmt=null;
    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="update      student          set
studentnumber=?,studentSchool=?,studentaddr=?,studentclass=? where studentname=?";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, studentnumber);
        pstmt.setString(2, studentSchool);
        pstmt.setString(3, studentaddr);
        pstmt.setString(4,studentclass);
        pstmt.setString(5,studentname);

        int n=pstmt.executeUpdate();
        if(n==1) System.out.println("updatechange success");

    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

}

} //update끝 수정

//점수부분수정
public void updatescore(String korscore, String engscore, String mathscore,
String historyscore, String avgg, String studentname) {
    Connection conn=null;
    PreparedStatement pstmt=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="update      report1          set
korscore=?,engscore=?,mathscore=?,historyscore=?, avgg=? where studentname=? ";
        pstmt=conn.prepareStatement(query);
        pstmt.setInt(1, Integer.parseInt(korscore));
        pstmt.setInt(2, Integer.parseInt(engscore));
        pstmt.setInt(3, Integer.parseInt(mathscore));
        pstmt.setInt(4, Integer.parseInt(historyscore));
        pstmt.setDouble(5,Double.parseDouble(avgg));
        pstmt.setString(6,studentname);

        int n=pstmt.executeUpdate();
        if(n==1) System.out.println("updatechange success");

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

}

}

```

LoginSystem.java –로그인 창

1. 로그인화면

```
class ImagePanel1 extends JPanel{
private Image img;
public ImagePanel1(Image img) {
    this.img=img;
    setSize(new Dimension(img.getWidth(null), img.getHeight(null)));
    setPreferredSize(new Dimension(img.getWidth(null), img.getHeight(null)));
}
public void paintComponent(Graphics g) {
    g.drawImage(img,0,0,null);
}
}

public class Loginsystem extends JFrame{

    JPanel panelMTitle;
    JPanel panelMScreen;

    Loginsystem(){
        JFrame frame=new JFrame();
        setSize(480,400);
        frame.setBounds(100, 120, 450, 360);

        setTitle("로그인 화면");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        frame.setSize(480,450);
        frame.setResizable(false);
        Container ct=getContentPane();

        ImagePanel1 imgpanel=new ImagePanel1(new
        ImageIcon("image\\LoginUI.jpg").getImage());
        ct.add(imgpanel, BorderLayout.SOUTH);

        imgpanel.setLayout(null);

        TextField text_id =new TextField();
        text_id.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        text_id.setBackground(new Color(255, 250, 205));
        text_id.setBounds(166, 158, 230, 40);
        imgpanel.add(text_id);

        JPasswordField txt_password =new JPasswordField();
        txt_password.setBackground(new Color(255, 250, 205));
        txt_password.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txt_password.setBounds(166, 236, 230, 40);
        imgpanel.add(txt_password);

        JButton btnlogin =new JButton("");
        btnlogin.setIcon(new ImageIcon("image\\buttonLogin1.jpg"));

        btnlogin.setBounds(176, 297, 159, 79);
        imgpanel.add(btnlogin);

        JLabel lblNewLabel =new JLabel("학원관리시스템");
        lblNewLabel.setForeground(new Color(188, 143, 143));
        lblNewLabel.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN, 50));
        lblNewLabel.setBounds(82, 44, 353, 80);
        imgpanel.add(lblNewLabel);
    }
}
```

2.로그인 버튼클릭시 - 아이디비밀번호 일치여부 확인

```
btnlogin.addActionListener(new ActionListener() {  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        String id="abc";  
        String pw="1234";  
  
        if(text_id.getText().length()==0 || txt_password.getPassword().length==0) {  
            JOptionPane.showMessageDialog(null, "  
아이디나 비밀번호가 입력되지않았습니다.");  
        }  
  
        if(id.equals(text_id.getText())&&pw.equals(txt_password.getText())){  
            JOptionPane.showMessageDialog(null, "  
로그인되었습니다.");  
            new Manageprogram();  
        }else{  
            JOptionPane.showMessageDialog(null, "  
아이디나 비밀번호가 틀렸습니다.");  
        }  
    }  
});
```

1. 화면구성

```
public class Manageprogram extends JFrame{
    JPanel panel_addinfo;
    List list;
    JPanel panel_changeinfo;
    JPanel panelMTTitle;
    JPanel panelMScreen;

    Manageprogram(){
        setResizable(false);
        setTitle("학생관리 프로그램");
        setSize(700,700);
        setResizable(false);
        setBounds(100, 100, 725, 527);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Container ct= getContentPane();
        ct.setLayout(new BorderLayout(0, 0));
        JPanel panelMTTitle =new JPanel();
        ct.add(panelMTTitle, BorderLayout.NORTH);
        JLabel labelMTTitle =new JLabel("학생관리 프로그램");
        labelMTTitle.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN, 43));
        panelMTTitle.add(labelMTTitle);
        panelMTTitle.setBackground(Color.decode("#FBE1BD"));
        JPanel panelMScreen =new JPanel();
        ct.add(panelMScreen, BorderLayout.CENTER);
        panelMScreen.setLayout(null);
        panelMScreen.setBackground(Color.decode("#F7E6C2"));
        //툴바내용
        JToolBar toolBar=new JToolBar("toolbar menu");
        JComboBox<String> comboBoxfind =new JComboBox<String>();
        JButton btnclear=new JButton("clear");
        btnclear.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        JButton btnteacher=new JButton("Teacher");
        btnteacher.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        JLabel toollabelsearch=new JLabel("search");
        toollabelsearch.setFont(new Font("한컴 말랑말랑 Regular", Font.BOLD,
20));

        JTextField txtfind=new JTextField();
        txtfind.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        comboBoxfind.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN,
20));

        JButton toolFind=new JButton();
        toolFind.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        toolFind.setIcon(new ImageIcon("imgWWfind.jpg"));
```

```

btnClear.setBackground(Color.decode("#FBE1BD"));
btnTeacher.setBackground(Color.decode("#FBE1BD"));

txtFind.setBackground(Color.decode("#FBE1BD"));
comboBoxFind.setBackground(Color.decode("#FBE1BD"));
toolFind.setBackground(Color.decode("#FBE1BD"));

toolBar.setMargin(new Insets(1,1,1,1));
toolBar.setFloatable(false);
toolBar.add(btnClear);
toolBar.addSeparator();
toolBar.add(btnTeacher);
toolBar.addSeparator();
toolBar.add(toolLabelSearch);
toolBar.addSeparator();
toolBar.add(comboBoxFind);
toolBar.addSeparator();
toolBar.add(txtFind);
toolBar.addSeparator();
toolBar.add(toolFind);
toolBar.setBounds(0, 377, 707, 33);
toolBar.setBackground(Color.decode("#F7E6C2"));
comboBoxFind.addItem("학생이름");
comboBoxFind.addItem("학생전화번호");
comboBoxFind.addItem("출신학교");
comboBoxFind.addItem("학생주소");
comboBoxFind.addItem("반");

list = new List();
list.setBounds(25, 64, 652, 235);
list.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 15));

panelMScreen.add(list);
panelMScreen.add(toolBar);

```

2. 조회,추가,삭제,수정,검색, clear,teacher 버튼 이벤트연결

```

JButton btnSearch =new JButton();
btnSearch.setBounds(43, 319, 105, 46);
btnSearch.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
panelMScreen.add(btnSearch);
btnSearch.setIcon(new ImageIcon("image\Wsearch.jpg"));
//조회하기 버튼 누르면 전체조회
btnSearch.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        displayAll();

    }

});
private void displayAll() {
    list.removeAll();
    StuDAO dao=new StuDAO();

    list.add("학생번호"+" "+"학생이름"+" "+"학
    생주소"+" "+"반"+" "+"국어성적"+" "+"수학성적"+" "+"영어성
    적"+" "+"한국사성적"+" "+"평균"+" "+"교사이름");
    ArrayList<StuDTO> allData=dao.select();
    for(StuDTO dto : allData) {
        int studentID=dto.getStudentID();
        String studentname=dto.getStudentname();
        String studentnumber=dto.getStudentnumber();
        String studentschool=dto.getStudentschool();
        String studentaddr=dto.getStudentaddr();
        String studentclass=dto.getStudentclass();
        int korscore=dto.getKorscore();
        int mathscore=dto.getMathscore();
        int engscore=dto.getEngscore();
        int historyscore=dto.getHistoryscore();
        double avgg=dto.getAvgg();
        int teacherID=dto.getTeacherID();
        String teachername=dto.getTeachername();
        String teachernumber=dto.getTeachernumber();
        String teachersubject=dto.getTeachersubject();
        String teacherclass=dto.getTeacherclass();

        list.add(studentID+" "+"studentname+"
        "+"studentnumber+" "+"studentschool+"
        "+"studentaddr+" "+"studentclass+"
        "+"korscore+" "+"mathscore+" "+"engscore+"
        "+"historyscore+" "+"avgg+" "+"teachername");//화면에 있는 리스트에
        붙이기

    }

}

```

<삭제>

```
JButton btnDelete =new JButton("삭제");
btnDelete.setBounds(365, 319, 105, 46);
btnDelete.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
btnDelete.setIcon(new ImageIcon("image\delete.jpg"));
panelMScreen.add(btnDelete);

//삭제하기 버튼누르면 누른 내용이 삭제
btnDelete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try { //선택된항목없을때 예외처리
            String str=list.getSelectedItemAt();
            StringTokenizer st=new StringTokenizer(str);
            int studentID= Integer.parseInt(st.nextToken());
            StuDAO dao=new StuDAO();
            dao.delete(studentID);
            displayAll();
        }catch(NullPointerException ne) {
            JOptionPane.showMessageDialog(null, "삭제할 학생이 선택
되지않았습니다.");
        }
    }
});
```

<추가>

```
JButton btnAdd =new JButton("추가");
btnAdd.setBounds(205, 319, 105, 46);
btnAdd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
panelMScreen.add(btnAdd);
btnAdd.setIcon(new ImageIcon("image\add.jpg"));
btnAdd.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        new Addstuinfo();
        displayAll();
    }
});
```

<수정>

```
JButton btnUpdate =new JButton("수정");
btnUpdate.setBounds(535, 319, 105, 46);
btnUpdate.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
btnUpdate.setIcon(new ImageIcon("image\\Wwupdate.jpg"));
panelMScreen.add(btnUpdate);
btnUpdate.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            String str=list.getSelectedItemAt();

            StringTokenizer st=new StringTokenizer(str);

            st.nextToken();
            String stunamech=st.nextToken();
            String stunumberch=st.nextToken();
            String stuschoolch=st.nextToken();
            String stuaddrch=st.nextToken();
            String stuclassch=st.nextToken();
            String korscorech=st.nextToken();
            String mathscorech=st.nextToken();
            String engscorech=st.nextToken();
            String historyscorech=st.nextToken();

            ChangeStuinfo chinfo=new ChangeStuinfo();

            chinfo.change(studentIDch,stunamech,stunumberch,stuschoolch,stuaddrch,stuclassch,korscorech,mathscorech,engscorech,historyscorech);
        }catch(NullPointerException ne) {
            JOptionPane.showMessageDialog(null, "수정될 학생이 선택되지않았습니다.");
        }
    }
});//수정버튼누르면 화면나오게
```


<툴바검색버튼>

```
toolFind.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        list.removeAll();
        int index=comboBoxFind.getSelectedIndex();
        System.out.println(index);
        String column="";//column값 초기화
        //column값 인덱스번호로 읽어오기
        switch(index) {
            case 0: column="studentname"; break;
            case 1: column="studentnumber"; break;
            case 2: column="studentSchool"; break;
            case 3: column="studentaddr"; break;
            case 4: column="studentclass"; break;
        }

        String item=txtFind.getText(); //텍스트에 입력한 검색할내용 읽어오기
        StuDAO dao=new StuDAO();
        list.add("학생번호"+" "+ "학생이름"+" "+ "학교이름"+" "+ "수학점
        "+ "학생전화번호"+" "+ "반"+" "+ "국어점수"+" "+ "수학점
        "+ "학생주소"+" "+ "영어점수"+" "+ "한국사점수"+" "+ "평균");
        ArrayList<StuDTO> AllData=dao.search(column,item);
        for(StuDTO dto : AllData) {
            int studentID=dto.getStudentID();
            String studentname=dto.getStudentname();
            String studentnumber=dto.getStudentnumber();
            String studentschool=dto.getStudentschool();
            String studentaddr=dto.getStudentaddr();
            String studentclass=dto.getStudentclass();
            int korscore=dto.getKorscore();
            int mathscore=dto.getMathscore();
            int engscore=dto.getEngscore();
            int historyscore=dto.getHistoryscore();
            double avgg=dto.getAvgg();

            list.add(studentID+" "+ "studentnumber"+"
            "+studentschool+" "+ "studentaddr"+" "+ "studentclass"+"
            "+engscore+" "+korscore+" "+mathscore+"
            "+historyscore+" "+avgg);//
            화면에 있는 리스트에 붙이기
        }

    }

});

ct.add(panelMScreen);
setVisible(true);

}
```

<툴바 teacher, clear버튼 클릭시>

```
String btnteacherpw="abcd";
    btnteacher.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {

            String passwd=JOptionPane.showInputDialog("비밀번호를
입력하세요");

            try{
                if(passwd.equals(btnteacherpw)) {
                    new Teacherinfo();}
                else JOptionPane.showMessageDialog(null, "비밀번호 오류
접근불가");
            }catch(NullPointerException ne) {

            }

        }

    });

//clear버튼 누르면 검색텍스트, 리스트 리셋
btnclear.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        list.removeAll();
        txtfind.setText("");

    }

});
```

AddStuinfo.java-학생정보추가창

1. 화면

```

public class Addstuinfo extends JFrame{

    Addstuinfo(){
        setTitle("학생정보추가프로그램");
        setBounds(100, 100, 550, 600);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Container ct= getContentPane();
        ct.setLayout(null);

        JPanel panel_addinfo =new JPanel();
        panel_addinfo .setBounds(0, 0, 549, 567);
        ct.add(panel_addinfo);
        panel_addinfo.setLayout(null);
        panel_addinfo.setBackground(Color.decode("#F7E6C2"));

        JLabel label_titleadd=new JLabel("학생정보입력");
        label_titleadd.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN, 36));
        label_titleadd.setBounds(181, 1, 200, 76);
        panel_addinfo.add(label_titleadd);

        JLabel label_nameadd =new JLabel("이름");
        label_nameadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_nameadd.setBounds(87, 80, 48, 39);
        panel_addinfo.add(label_nameadd);

        JLabel label_phoneadd=new JLabel("연락처");
        label_phoneadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_phoneadd.setBounds(87, 131, 66, 39);
        panel_addinfo.add(label_phoneadd);

        JLabel label_school =new JLabel("학교명");
        label_school.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_school.setBounds(87, 175, 66, 39);
        panel_addinfo.add(label_school);

        JLabel label_addr =new JLabel("주소");
        label_addr.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_addr.setBounds(87, 226, 66, 39);
        panel_addinfo.add(label_addr);

        JTextField txtnameadd =new JTextField();
        txtnameadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txtnameadd.setBounds(205, 89, 146, 30);
        panel_addinfo.add(txtnameadd);
        txtnameadd.setColumns(10);
        JTextField txtphoneadd =new JTextField();
        txtphoneadd .setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txtphoneadd .setColumns(10);
        txtphoneadd.setText("-제외");
        txtphoneadd .setBounds(205, 135, 146, 30);
        panel_addinfo.add(txtphoneadd);
        JTextField txtschooladd=new JTextField();
        txtschooladd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txtschooladd.setColumns(10);
        txtschooladd.setBounds(205, 179, 146, 30);
        panel_addinfo.add(txtschooladd);
    }
}

```

```

JTextField txtAddradd=new JTextField();
txtAddradd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtAddradd.setColumns(10);
txtAddradd.setBounds(205, 230, 146, 30);
panel_addinfo.add(txtAddradd);

JLabel label_scoreadd =new JLabel("점수");
label_scoreadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_scoreadd.setBounds(87, 319, 48, 39);
panel_addinfo.add(label_scoreadd);

JLabel label_koradd =new JLabel("국어");
label_koradd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_koradd.setBounds(168, 316, 66, 39);
panel_addinfo.add(label_koradd);

JLabel label_mathadd =new JLabel("수학");
label_mathadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_mathadd.setBounds(168, 367, 66, 39);
panel_addinfo.add(label_mathadd);

JLabel label_engadd=new JLabel("영어");
label_engadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_engadd.setBounds(168, 414, 66, 39);
panel_addinfo.add(label_engadd);

JLabel label_historyadd=new JLabel("한국사");
label_historyadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_historyadd.setBounds(168, 456, 66, 39);
panel_addinfo.add(label_historyadd);

JTextField txtkoradd =new JTextField();
txtkoradd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtkoradd.setColumns(10);
txtkoradd.setBounds(235, 323, 146, 30);
panel_addinfo.add(txtkoradd);

JTextField txtmathadd =new JTextField();
txtmathadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtmathadd.setColumns(10);
txtmathadd.setBounds(235, 367, 146, 30);
panel_addinfo.add(txtmathadd);

JTextField txtengadd =new JTextField();
txtengadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtengadd.setColumns(10);
txtengadd.setBounds(235, 418, 146, 30);
panel_addinfo.add(txtengadd);

JTextField txthistoryadd =new JTextField();
txthistoryadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txthistoryadd.setColumns(10);
txthistoryadd.setBounds(235, 460, 146, 30);
panel_addinfo.add(txthistoryadd);

JButton btnaddsave =new JButton("입력완료");
btnaddsave.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
btnaddsave.setBounds(194, 516, 113, 39);
panel_addinfo.add(btnaddsave);

```

2. 입력완료 버튼 클릭시 insert

```

btnaddsave.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //번호는 자동생성이므로 뺀다.
        try {
            String studentname=txtnameadd.getText();
            String studentnumber=txtphoneadd.getText();
            String studentSchool=txtschooladd.getText();
            String studentaddr=txtAddradd.getText();
            String korscore=txtkoradd.getText();
            String mathscore=txtmathadd.getText();
            String engscore=txtengadd.getText();
            String historyscore=txthistoryadd.getText();
            S          t          r          i          n          g
avgg=String.valueOf((Integer.parseInt(korscore)+Integer.parseInt(engscore)+Integer.parseInt(mathscore)+Integer.parseInt(historyscore))/4);
            String stuclass="";

            //점수대로 반 나누기

            if(((Integer.parseInt(korscore)+Integer.parseInt(engscore)+Integer.parseInt(mathscore)+Integer.parseInt(historyscore))/4)>=90.0) stuclass="A";
            if(((Integer.parseInt(korscore)+Integer.parseInt(engscore)+Integer.parseInt(mathscore)+Integer.parseInt(historyscore))/4)>=80.0 ) stuclass="B";
            if(((Integer.parseInt(korscore)+Integer.parseInt(engscore)+Integer.parseInt(mathscore)+Integer.parseInt(historyscore))/4)>=70.0) stuclass="C";
            else stuclass="D";

            StuDAO dao=new StuDAO();

            if ( studentnumber.length() != 11 )
{JOptionPane.showMessageDialog(null,"입력한 전화번호형식이 잘못되었습니다");
            System.exit(DO_NOTHING_ON_CLOSE);}

            //점수 0점~100점사이 그외는 예외처리
            if (Integer.parseInt(korscore)>100 ||
Integer.parseInt(korscore)<0 || Integer.parseInt(mathscore)>100 ||
Integer.parseInt(mathscore)<0 || Integer.parseInt(engscore)>100 ||
Integer.parseInt(engscore)<0 || Integer.parseInt(historyscore)>100 ||
Integer.parseInt(historyscore)<0)
            {
                JOptionPane.showMessageDialog(null,"점수입력이 잘못되었습니다");

                //System.exit(DISPOSE_ON_CLOSE);
                check=false;
            }
        }
    }
});

```

```

if(dao.checknum(studentnumber)==true) {

    //파일로 저장
    try {

        FileWriter savefile=new FileWriter("studentinfo.csv",true);
        String studentnameap=txtnameadd.getText();
        String studentnumberap=txtphoneadd.getText();
        String studentSchoolap=txtschooladd.getText();
        String studentaddrap=txtAddradd.getText();
        String korscoreap=txtkoradd.getText();
        String mathscoreap=txtmathadd.getText();
        String engscoreap=txtengadd.getText();
        String historyscoreap=txthistoryadd.getText();
        S          t          r          i          n          g
        avggap=String.valueOf((Integer.parseInt(korscore)+Integer.parseInt(engscore)+Integer.parseInt(math
        score)+Integer.parseInt(historyscore))/4);

        savefile.write(studentnameap+","+studentnumberap+","+studentSchoolap+","+studentaddrap+","+kor
        scoreap+","+engscoreap+","+mathscoreap+","+historyscoreap+","+avggap+"\n");
        savefile.close();

    }catch(Exception ee) {

        JOptionPane.showMessageDialog(null, "파일 저장
        실패");System.exit(DISPOSE_ON_CLOSE);
    }//파일 저장부분끝

    dao.insertstu(studentname,studentnumber,studentSchool,studentaddr,stuclass);

    dao.insertscore(studentname,korscore,mathscore,engscore,historyscore,avgg);
    }
    //입력오류 예외처리
    }catch(NumberFormatException n) {
        JOptionPane.showMessageDialog(null, "입력이 잘못되었습니다.");
    }
    }

});

```

ChangeStuinfo.java- 학생정보 수정창

1.화면

```
public class ChangeStuinfo extends JFrame{
    JTextField txtnamechange =new JTextField();
    JTextField txtphonechange =new JTextField();
    JTextField txtschoolchange =new JTextField();
    JTextField txtAddrchange =new JTextField();
    JTextField txtchangeinfo =new JTextField();
    JTextField txtkorchange =new JTextField();
    JTextField txtmathchange =new JTextField();
    JTextField txtengchange =new JTextField();
    JTextField txthistorychange =new JTextField();

    ChangeStuinfo(){
        setTitle("학생정보수정창");
        setBounds(100, 100, 550, 600);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Container ct= getContentPane();
        ct.setLayout(null);

        JPanel panel_changeinfo=new JPanel();
        panel_changeinfo.setBounds(0, 0, 549, 567);
        ct.add(panel_changeinfo);
        panel_changeinfo.setLayout(null);
        panel_changeinfo.setBackground(Color.decode("#F7E6C2"));

        JLabel label_titlechange=new JLabel("학생정보수정");
        label_titlechange.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN, 36));
        label_titlechange.setBounds(181, 1, 200, 76);
        panel_changeinfo.add(label_titlechange);

        JLabel label_namechange =new JLabel("이름");
        label_namechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_namechange.setBounds(87, 80, 48, 39);
        panel_changeinfo.add(label_namechange);

        JLabel label_phonechange=new JLabel("연락처");
        label_phonechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_phonechange.setBounds(87, 131, 66, 39);
        panel_changeinfo.add(label_phonechange);

        JLabel label_classchange =new JLabel("반");
        label_classchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_classchange.setBounds(87, 268, 66, 39);
        panel_changeinfo.add(label_classchange);

        txtchangeinfo.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txtchangeinfo.setColumns(10);
        txtchangeinfo.setBounds(205, 274, 146, 30);
        panel_changeinfo.add(txtchangeinfo);

        JLabel label_schoolchange =new JLabel("학교명");
        label_schoolchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_schoolchange.setBounds(87, 175, 66, 39);
        panel_changeinfo.add(label_schoolchange);
        JLabel label_addrchange =new JLabel("주소");
        label_addrchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_addrchange.setBounds(87, 226, 66, 39);
        panel_changeinfo.add(label_addrchange);
        txtnamechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txtnamechange.setBounds(205, 89, 146, 30);
        panel_changeinfo.add(txtnamechange);
        txtnamechange.setColumns(10);
        txtnamechange.setEditable(false);
    }
}
```



```

txtphonechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtphonechange.setColumns(10);
txtphonechange.setBounds(205, 135, 146, 30);
panel_changeinfo.add(txtphonechange);
txtschoolchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtschoolchange.setColumns(10);
txtschoolchange.setBounds(205, 179, 146, 30);
panel_changeinfo.add(txtschoolchange);
txtAddrchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtAddrchange.setColumns(10);
txtAddrchange.setBounds(205, 230, 146, 30);
panel_changeinfo.add(txtAddrchange);
JLabel label_scorechange =new JLabel("점수");
label_scorechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_scorechange.setBounds(87, 319, 48, 39);
panel_changeinfo.add(label_scorechange);
JLabel label_korchange =new JLabel("국어");
label_korchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_korchange.setBounds(168, 316, 66, 39);
panel_changeinfo.add(label_korchange);
JLabel label_mathchange =new JLabel("수학");
label_mathchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_mathchange.setBounds(168, 367, 66, 39);
panel_changeinfo.add(label_mathchange);

JLabel label_engchange=new JLabel("영어");
label_engchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_engchange.setBounds(168, 414, 66, 39);
panel_changeinfo.add(label_engchange);

JLabel label_historychange=new JLabel("한국사");
label_historychange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
label_historychange.setBounds(168, 456, 66, 39);
panel_changeinfo.add(label_historychange);

txtkorchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtkorchange.setColumns(10);
txtkorchange.setBounds(235, 323, 146, 30);
panel_changeinfo.add(txtkorchange);
txtmathchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtmathchange.setColumns(10);
txtmathchange.setBounds(235, 367, 146, 30);
panel_changeinfo.add(txtmathchange);
txtengchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txtengchange.setColumns(10);
txtengchange.setBounds(235, 418, 146, 30);
panel_changeinfo.add(txtengchange);
txthistorychange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txthistorychange.setColumns(10);
txthistorychange.setBounds(235, 460, 146, 30);
panel_changeinfo.add(txthistorychange);
JButton btnchangesave =new JButton("입력완료");
btnchangesave.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
btnchangesave.setBounds(194, 516, 113, 39);
panel_changeinfo.add(btnchangesave);
txtnamechange.setText("");
txtphonechange.setText("");
txtschoolchange.setText("");
txtAddrchange.setText("");
txtchangechange.setText("");
txtkorchange.setText("");
txtengchange.setText("");
txtmathchange.setText("");
txthistorychange.setText("");

```

2.입력완료 버튼 클릭시 update

```
btnchangesave.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //번호는 자동생성이므로 뺀다.
        try { int studentID=stuid;
                String studentname=txtnamechange.getText();
                String studentnumber=txtphonechange.getText();
                String studentSchool=txtschoolchange.getText();
                String studentaddr=txtAddrchange.getText();
                String studentclass=txtchangechange.getText();
                String korscore=txtkorchange.getText();
                String mathscore=txtmathchange.getText();
                String engscore=txtengchange.getText();
                String historyscore=txthistorychange.getText();
                S t r i n g
                avgg=String.valueOf((Integer.parseInt(korscore)+Integer.parseInt(engscore)+Integer.
                parseInt(mathscore)+Integer.parseInt(historyscore))/4);
                StuDAO dao=new StuDAO();

                if(phonechange.equals(studentnumber)) {

                    dao.updatechangestu(stuid,studentname,studentnumber,studentSchool,studentaddr
                    ,studentclass);

                    dao.updatescore(korscore,engscore,mathscore,historyscore,avgg,stuid);
                    JOptionPane.showMessageDialog(null, "수정완
                    료");

                }
                else if(dao.checknum(studentnumber)==true) {

                    dao.updatechangestu(stuid,studentname,studentnumber,studentSchool,studentaddr
                    ,studentclass);

                    dao.updatescore(korscore,engscore,mathscore,historyscore,avgg,stuid);
                    JOptionPane.showMessageDialog(null, "수정완료");}

                }catch(NumberFormatException nume) {
                    JOptionPane.showMessageDialog(null, "수정할내용이 잘못입력
                    되었습니다. 다시입력하세요");
                }
            }
        });
```

3.수정창에 선택된 정보들이 텍스트에 출력되도록 하는 기능

```
public void change(int studentIDch,String stunamech, String stunumberch, String
stuschoolch, String stuaddrch, String stuclassch, String korscorech, String
mathscorech, String engscorech, String historyscorech) {
    stuid=studentIDch;
    txtnamechange.setText(stunamech);
    txtphonechange.setText(stunumberch);
    txtschoolchange.setText(stuschoolch);
    txtAddrchange.setText(stuaddrch);
    txtchangechange.setText(stuclassch);
    txtkorchange.setText(korscorech);
    txtmathchange.setText(mathscorech);
    txtengchange.setText(engscorech);
    txthistorychange.setText(historyscorech);
    phonechange=stunumberch;
}
```

Teacherinfo.java-교사정보관리창

1.화면

```
public class Teacherinfo extends JFrame {
    List list;
    JPanel panel_changetinfo;
    JTextField txtfind=new JTextField();
    Teacherinfo(){

        setTitle("학생관리프로그램");
        setSize(700,700);
        setResizable(false);
        setBounds(100, 100, 725, 527);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Container ct= getContentPane();
        ct.setLayout(new BorderLayout(0, 0));

        JPanel panelMTTitle = new JPanel();
        ct.add(panelMTTitle, BorderLayout.NORTH);

        JLabel labelMTTitle = new JLabel("교사 정보관리");
        labelMTTitle.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN, 43));
        panelMTTitle.add(labelMTTitle);
        panelMTTitle.setBackground(Color.decode("#FBE1BD"));

        JPanel panelMScreen = new JPanel();
        ct.add(panelMScreen, BorderLayout.CENTER);
        panelMScreen.setLayout(null);
        panelMScreen.setBackground(Color.decode("#F7E6C2"));

        JComboBox<String> comboBoxfind = new JComboBox<String>();
        JButton btnclear=new JButton("clear");
        btnclear.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));

        JLabel toollabelsearch=new JLabel("search");
        toollabelsearch.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));

        txtfind.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        comboBoxfind.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));

        JToolBar toolBar=new JToolBar("toolbar menu");
        toolBar.setFloatable(false);//툴바고정
        JButton tooltFind=new JButton();
        tooltFind.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        tooltFind.setIcon(new ImageIcon("image\\w\\find.jpg"));
        btnclear.setBackground(Color.decode("#F7E6C2"));
        toollabelsearch.setBackground(Color.decode("#F7E6C2"));
        txtfind.setBackground(Color.decode("#F7E6C2"));
        comboBoxfind.setBackground(Color.decode("#F7E6C2"));
        tooltFind.setBackground(Color.decode("#F7E6C2"));
        toolBar.setMargin(new Insets(1,1,1,1));

        toolBar.add(btnclear);

        toolBar.add(toollabelsearch);

        toolBar.add(comboBoxfind);
        toolBar.add(txtfind);
        toolBar.add(tooltFind);
        ct.add(toolBar);
        panelMScreen.add(toolBar);
        toolBar.setBounds(0, 377, 707, 33);

        comboBoxfind.addItem("이름");
        comboBoxfind.addItem("전화번호");
        comboBoxfind.addItem("과목");
        comboBoxfind.addItem("반");

        list = new List();
        list.setBounds(25, 64, 652, 235);
        panelMScreen.add(list);
    }
}
```

2.버튼이벤트

<추가,삭제버튼>

```

JButton btnAdd = new JButton("추가");
btnAdd.setBounds(205, 319, 105, 46);
btnAdd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
paneltMScreen.add(btnAdd);
btnAdd.setIcon(new ImageIcon("image\Wwadd.jpg"));
btnAdd.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        new AddteacherInfo();
        displayAll();
    }
});

JButton btnDelete = new JButton("삭제");
btnDelete.setBounds(365, 319, 105, 46);
btnDelete.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
btnDelete.setIcon(new ImageIcon("image\Wwdelete.jpg"));
paneltMScreen.add(btnDelete);

//삭제하기 버튼누르면 누른 내용이 삭제
btnDelete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try{
            String str=list.getSelectedItem();

            StringTokenizer st=new StringTokenizer(str);
            int teacherID= Integer.parseInt(st.nextToken());

            TeacherDAO dao= new TeacherDAO();
            dao.delete(teacherID);
            displayAll();
        }catch(NullPointerException ne) {
            JOptionPane.showMessageDialog(null, "삭제할 교사가
선택되지않았습니다.");
        }
    }
});
```

<수정버튼>

```

JButton btnUpdate = new JButton("수정");

btnUpdate.setBounds(535, 319, 105, 46);
btnUpdate.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
btnUpdate.setIcon(new ImageIcon("image\\Wwupdate.jpg"));
paneltMScreen.add(btnUpdate);
btnUpdate.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            String str=list.getSelectedItemAt();
            StringTokenizer st=new StringTokenizer(str);
            st.nextToken();
            String tnamech=st.nextToken();
            String tnumberch=st.nextToken();
            String tsubjectch=st.nextToken();
            String tclassch=st.nextToken();

            Changeteacherinfo chtinfo=new Changeteacherinfo();
            chtinfo.change(tnamech,tnumberch,tsubjectch,tclassch);
        }catch(NullPointerException ne) {
            JOptionPane.showMessageDialog(null, "수정될 교사가
선택되지않았습니다.");
        }
    }
});//수정버튼누르면 화면나오게

```

<툴바검색버튼>

```

tooltFind.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        list.removeAll();
        int index=comboBoxfind.getSelectedIndex();
        System.out.println(index);
        String column="";//column값 초기화

        switch(index) { //column값 인덱스값으로 검색해서 넣기
            case 0: column="teachername"; break;
            case 1: column="teachernumber"; break;
            case 2: column="teachersubject"; break;
            case 3: column="teacherclass"; break;
        }
        System.out.println(column);

        System.out.println(txtfind.getText());
        String item=txtfind.getText();
        TeacherDAO dao= new TeacherDAO();
        list.add("번호"+"
"+"이름"+"
"+"과목"+"
"+"교사전화번호"+"
"+"담당반");
        ArrayList<TeacherDTO> AllData=dao.search(column,item);
        for(TeacherDTO dto : AllData) {
            int teacherID=dto.getTeacherID();
            String teachername=dto.getTeachername();
            String teachernumber=dto.getTeachernumber();
            String teachersubject=dto.getTeachersubject();
            String teacherclass=dto.getTeacherclass();

            list.add(teacherID+"
"+teachername+"
"+teachersubject+"
"+teachernumber+"
"+teacherclass);
        }
    }
});

```

<조회버튼>

```

JButton btnSearch = new JButton();

    btnSearch.setBounds(43, 319, 105, 46);
    btnSearch.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
    paneltMScreen.add(btnSearch);
    btnSearch.setIcon(new ImageIcon("image\Wsearch.jpg"));
    //조회하기 버튼 누르면 전체조회
    btnSearch.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {

            displayAll();

        }
    });
private void displayAll() {
    list.removeAll();
    TeacherDAO dao=new TeacherDAO();

    list.add("번호"+"
"+"이름"+"
"+"과목"+"
"+"전화번호"+"
"+"담당반");
    ArrayList<TeacherDTO> allData=dao.select();
    for(TeacherDTO tdto : allData) {

        int teacherID=tdto.getTeacherID();
        String teachername=tdto.getTeachername();
        String teachernumber=tdto.getTeachernumber();
        String teachersubject=tdto.getTeachersubject();
        String teacherclass=tdto.getTeacherclass();

        list.add(teacherID+"
"+teachername+"
"+teachersubject+"
"+teacherclass);

    }

}

```

TeacherDAO.java

1.데이터베이스연결

```

public class TeacherDAO {
    String driver="com.mysql.cj.jdbc.Driver";
    String url="jdbc:mysql://localhost:3306/project";
    String userid="root";
    String passwd="1234";
    Connection conn=null;
    Statement stmt=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;

    TeacherDAO(){
        try {
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } //드라이버로딩
    }
}

```

2.전체조회

```
public ArrayList<TeacherDTO> select(){
    ArrayList<TeacherDTO> list=new ArrayList<TeacherDTO>();//객체생성

    Connection conn=null;
    Statement stmt=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="select * from teacher";
        //stmt=conn.createStatement();
        pstmt=conn.prepareStatement(query);
        rs=pstmt.executeQuery();
        while(rs.next()) {
            TeacherDTO tdto=new TeacherDTO();//객체생성
            tdto.setTeacherID(rs.getInt("teacherID"));
            tdto.setTeachername(rs.getString("teachername"));
            tdto.setTeachernumber(rs.getString("teachernumber"));
            tdto.setTeachersubject(rs.getString("teachersubject"));
            tdto.setTeacherclass(rs.getString("teacherclass"));
            list.add(tdto);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            rs.close();
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return list;
}
```


3.교사전화번호중복체크

```
String teachernumber;
int cnt=0;
boolean result=true;
public boolean checknum(String stunum) {
    Connection conn=null;
    PreparedStatement pstmt=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="select * from teacher where teachernumber=?";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, stunum);
        rs=pstmt.executeQuery();

        while(rs.next()) {
            cnt++; //같은번호 있으면 cnt값 추가
        }
        if(cnt==1) {
            JOptionPane.showMessageDialog(null, "중복된
전화번호입니다.");

            result=false;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return result;
}
```

4.교사정보삭제

```
public void delete(int teacherID) {
    Connection conn=null;
    PreparedStatement pstmt=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="delete from teacher where teacherID=?";
        pstmt=conn.prepareStatement(query);
        pstmt.setInt(1, teacherID);
        int n=pstmt.executeUpdate();
        if(n==1) System.out.println("삭제완료");

    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

5.교사정보삽입:insert

```
public void insert(String teachername, String teachernumber, String teachersubject,
String teacherclass) {
    Connection conn=null;
    PreparedStatement pstmt=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        String query="insert into
teacher(teachername,teachernumber,teachersubject,teacherclass) values(?,?,?,?)";

        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, teachername);
        pstmt.setString(2, teachernumber);
        pstmt.setString(3, teachersubject);
        pstmt.setString(4, teacherclass);

        int n=pstmt.executeUpdate();
        if(n==1) { System.out.println("insert
success");JOptionPane.showMessageDialog(null, "입력완료");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

6.교사정보수정

```
public void updatechangeget(String teachername, String teachernumber, String teachersubject, String
teacherclass) {

    Connection conn=null;
    PreparedStatement pstmt=null;

    try {    conn=DriverManager.getConnection(url, userid, passwd);
        String query="update teacher set
teachernumber=?,teachersubject=?,teacherclass=? where teachername=?";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, teachernumber);
        pstmt.setString(2, teachersubject);
        pstmt.setString(3,teacherclass);
        pstmt.setString(4, teachername);

        int n=pstmt.executeUpdate();
        if(n==1) System.out.println("updatechange success");

    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

7.교사정보 콤보박스해당내용 검색

```
public ArrayList<TeacherDTO> search(String column, String item){
    ArrayList<TeacherDTO> list=new ArrayList<TeacherDTO>();//객체생성

    Connection conn=null;
    Statement stmt=null;
    PreparedStatement pstmt=null;

    ResultSet rs=null;

    try {
        conn=DriverManager.getConnection(url, userid, passwd);
        switch(column) {
            case "teachername": String query="select * from teacher where
teachername=?"; pstmt=conn.prepareStatement(query); break;
            case "teachernumber": query="select * from teacher where
teachernumber=?";pstmt=conn.prepareStatement(query); break;
            case "teachersubject": query="select * from teacher where
teachersubject=?";pstmt=conn.prepareStatement(query); break;
            case "teacherclass": query="select * from teacher where
teacherclass=?";pstmt=conn.prepareStatement(query); break;
        }
        stmt=conn.createStatement();
        pstmt.setString(1, item);
        rs=pstmt.executeQuery();

        while(rs.next()) {
            TeacherDTO dto=new TeacherDTO();//객체생성
            dto.setTeacherID(rs.getInt("teacherID"));
            dto.setTeachername(rs.getString("teachername"));
            dto.setTeachernumber(rs.getString("teachernumber"));
            dto.setTeachersubject(rs.getString("teachersubject"));
            dto.setTeacherclass(rs.getString("teacherclass"));
            list.add(dto);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            rs.close();
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return list;
}
```

Addteacherinfo.java-교사정보추가창

1.화면

```
public class AddteacherInfo extends JFrame{

    AddteacherInfo(){
        setTitle("교사정보추가프로그램");
        setBounds(100, 100, 550, 600);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Container ct= getContentPane();
        ct.setLayout(null);
        JPanel panel_addtinfo = new JPanel();
        panel_addtinfo.setBounds(0, 0, 549, 567);
        ct.add(panel_addtinfo);
        panel_addtinfo.setLayout(null);
        panel_addtinfo.setBackground(Color.decode("#F7E6C2"));

        JLabel label_titleadd= new JLabel("교사정보입력");
        label_titleadd.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN, 36));
        label_titleadd.setBounds(181, 1, 200, 76);
        panel_addtinfo.add(label_titleadd);

        JLabel label_tnameadd = new JLabel("이름");
        label_tnameadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_tnameadd.setBounds(87, 80, 48, 39);
        panel_addtinfo.add(label_tnameadd);

        JLabel label_tphoneadd= new JLabel("연락처");
        label_tphoneadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_tphoneadd.setBounds(87, 131, 66, 39);
        panel_addtinfo.add(label_tphoneadd);

        JLabel label_tsubject= new JLabel("과목명");
        label_tsubject.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_tsubject.setBounds(87, 175, 66, 39);
        panel_addtinfo.add(label_tsubject);

        JLabel label_tclass = new JLabel("반");
        label_tclass.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        label_tclass.setBounds(87, 226, 66, 39);
        panel_addtinfo.add(label_tclass);

        JTextField txttnameadd = new JTextField();
        txttnameadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txttnameadd.setBounds(205, 89, 146, 30);
        panel_addtinfo.add(txttnameadd);
        txttnameadd.setColumns(10);

        JTextField txttphoneadd = new JTextField();
        txttphoneadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txttphoneadd.setColumns(15);
        txttphoneadd.setText("-제외");
        txttphoneadd.setBounds(205, 135, 146, 30);
        panel_addtinfo.add(txttphoneadd);

        JTextField txttsubjectadd= new JTextField();
        txttsubjectadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txttsubjectadd.setColumns(10);
        txttsubjectadd.setBounds(205, 179, 146, 30);
        panel_addtinfo.add(txttsubjectadd);
        JTextField txttclassadd= new JTextField();
        txttclassadd.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        txttclassadd.setColumns(10);
        txttclassadd.setBounds(205, 230, 146, 30);
        panel_addtinfo.add(txttclassadd);

        JButton btnaddsave = new JButton("입력완료");
        btnaddsave.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
        btnaddsave.setBounds(194, 516, 113, 39);
        panel_addtinfo.add(btnaddsave);
    }
}
```

2. 입력완료클릭시 이벤트

```
btnaddsave.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //번호는 자동생성이므로 뺀다.
        String teachername=txttnameadd.getText();
        String teachernumber=txttphoneadd.getText();
        String teachersubject=txttsubjectadd.getText();
        String teacherclass=txttclassadd.getText();

        TeacherDAO dao=new TeacherDAO();
        if(teachername.length()==0 || teachernumber.length()==3 || teachernumber.length()==0 |
        | teachersubject.length()==0) {
            JOptionPane.showMessageDialog(null,"입력된 값이
            없습니다."); System.exit(EXIT_ON_CLOSE);
        }
        else if(teachernumber.length()!=11)
        {JOptionPane.showMessageDialog(null,"입력한 전화번호형식이 잘못되었습니다");
        System.exit(DISPOSE_ON_CLOSE); }

        dao.checknum(teachernumber);

        if(dao.checknum(teachernumber)==true) {

            dao.insert(teachername,teachernumber,teachersubject,teacherclass);
        }

    }

});
```

Changeteacherinfo.java -교사정보수정창

1.화면

```
public class Changeteacherinfo extends JFrame{
    JTextField txttnamechange = new JTextField();
    JTextField txttphonechange = new JTextField();
    JTextField txttsubjectchange = new JTextField();
    JTextField txttchangeclass = new JTextField();

    Changeteacherinfo(){
        setTitle("교사정보수정창");
        setBounds(100, 100, 550, 600);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Container ct= getContentPane();
        ct.setLayout(null);

        JPanel panel_changetinfo= new JPanel();
        panel_changetinfo.setBounds(0, 0, 549, 567);
        ct.add(panel_changetinfo);
        panel_changetinfo.setLayout(null);
        panel_changetinfo.setBackground(Color.decode("#F7E6C2"));

        JLabel label_titlechange= new JLabel("교사정보수정");
        label_titlechange.setFont(new Font("한컴 말랑말랑 Bold", Font.PLAIN,
36));
        label_titlechange.setBounds(181, 1, 200, 76);
        panel_changetinfo.add(label_titlechange);

        JLabel label_tnamechange = new JLabel("이름");
        label_tnamechange.setFont(new Font("한컴 말랑말랑 Regular",
Font.PLAIN, 20));
        label_tnamechange.setBounds(87, 80, 48, 39);
        panel_changetinfo.add(label_tnamechange);

        JLabel label_tphonechange= new JLabel("연락처");
        label_tphonechange.setFont(new Font("한컴 말랑말랑 Regular",
Font.PLAIN, 20));
        label_tphonechange.setBounds(87, 131, 66, 39);
        panel_changetinfo.add(label_tphonechange);

        JLabel label_tsubjectchange = new JLabel("과목명");
        label_tsubjectchange.setFont(new Font("한컴 말랑말랑 Regular",
Font.PLAIN, 20));
        label_tsubjectchange.setBounds(87, 175, 66, 39);
        panel_changetinfo.add(label_tsubjectchange);

        JLabel label_taddrchange = new JLabel("반");
        label_taddrchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN,
20));
        label_taddrchange.setBounds(87, 226, 66, 39);
        panel_changetinfo.add(label_taddrchange);
    }
}
```

```

txttnamechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN, 20));
txttnamechange.setBounds(205, 89, 146, 30);
panel_changgetinfo.add(txttnamechange);
txttnamechange.setColumns(10);
txttnamechange.setEditable(false);

20));
txttphonechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN,
txttphonechange.setColumns(10);
txttphonechange.setBounds(205, 135, 146, 30);
panel_changgetinfo.add(txttphonechange);

20));
txttsubjectchange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN,
txttsubjectchange.setColumns(10);
txttsubjectchange.setBounds(205, 179, 146, 30);
panel_changgetinfo.add(txttsubjectchange);

20));
txttchangechange.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN,
txttchangechange.setColumns(10);
txttchangechange.setBounds(205, 230, 146, 30);
panel_changgetinfo.add(txttchangechange);

20));
JButton btnchangesave = new JButton("입력완료");
btnchangesave.setFont(new Font("한컴 말랑말랑 Regular", Font.PLAIN,
btnchangesave.setBounds(194, 516, 113, 39);
panel_changgetinfo.add(btnchangesave);

txttnamechange.setText("");
txttphonechange.setText("");
txttsubjectchange.setText("");
txttchangechange.setText("");

```

2.입력완료 클릭시 이벤트

```
btnchangesave.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //번호는 자동생성이므로 뺀다.
        try { String teachername=txttnamechange.getText();
                String teachernumber=txttphonechange.getText();
                String teachersubject=txttsubjectchange.getText();
                String teacherclass=txttchangechange.getText();

                TeacherDAO dao=new TeacherDAO();
                if(tphonechange.equals(teachernumber)) {

                    dao.updatechangeget(teachername,teachernumber,teachersubject,teacherclass);

                    JOptionPane.showMessageDialog(null, "수정완료");}

                    else if(dao.checknum(teachernumber)==true) {

                        dao.updatechangeget(teachername,teachernumber,teachersubject,teacherclass);
                        JOptionPane.showMessageDialog(null, "수정완
                        료");}

                }catch(NumberFormatException nume) {
                    JOptionPane.showMessageDialog(null, "수정할내용이 잘못입
                    력되었습니다. 다시입력하세요");
                }
            }
        });
```

3. 메인화면에서 수정할 교사선택후 수정버튼 눌렀을 때 해당내용이 수정창에 나타나게하기

```
public void change(String tnamech, String tnumberch, String tsubjectch, String
tclassch) {
    txttnamechange.setText(tnamech);
    txttphonechange.setText(tnumberch);
    txttsubjectchange.setText(tsubjectch);

    txttchangechange.setText(tclassch);
```