

# 연구논문/작품 중간보고서

2020 학년도 제 2 학기

제목	초등교사를 위한 시간표 계획 웹 서비스	○ 논문( ) 작품( √ ) ※해당란 체크
GitHub URL	<a href="https://github.com/hyeonjang/class_schedule_maker">https://github.com/hyeonjang/class_schedule_maker</a>	
평가등급	지도교수 수정보완 사항	팀원 명단
A,B,F 중 택 1 (지도교수가 부여)	○  ○  ○	안현장 (학번:2013314597)  ○ ○ ○ (인) (학번: )  ○ ○ ○ (인) (학번: )  ○ ○ ○ (인) (학번: )

2020 년 9 월 14 일

지도교수 : 한 환 수 서명

## ■ 요약

본 연구 작품은 초등교사를 위한 시간표를 계획 웹 서비스를 제공한다. 해당 서비스는 초등담임교사의 학사계획업무를 체계적으로 수행하도록 도와, 초등학교 학년별 시간표를 더 체계적으로 계획하도록 돕는 것을 목표로 한다.

이를 통해 초등교사의 학사계획업무 부담을 더는 것뿐만 아니라, 세심한 시간표 계획을 가능하게 하여 학생들의 성장 발달 단계와 사회적 계기를 고려한 수업을 계획할 수 있도록 한다.

수업을 맡는 초등교사의 역할은 각각 담임교사, 교과전담교사, 초청강사로 나뉘며, 프로그램 내에서도 사용자의 권한을 역할과 사용자별로 한정시켜, 개인이 맡아야 할 업무를 분명히 한다.

초등교사 시간표 계획의 핵심은 한정된 조건 안에서 담임교사로 하여금 시간표에 대한 유동성을 발휘할 수 있도록 하는 것이다. 또한 이 과정 속에서 고려해야 할 제약을 분명히 하여, 불필요한 과정과 의사소통의 문제를 해결한다.

## ■ 서론

### 1) 제안배경 및 필요성

현재 초등학교의 시간표 계획은 비체계적으로 이루어지고 있다. 시간표에 관하여 교과별 시수나 교과과정에 대한 교육청의 지침은 존재하지만, 세부적인 내용은 각 학교에 재량에 맡기기 때문이다. 또한 각 지방의 교육청이 실시하고 있는 혁신학교 사업은 학교별로 학사계획업무를 다르게 만든다.

이 때문에, 담임교사들이 학기 초마다 시간표 계획에 많은 시간을 할애하고 있다. 초등담임교사들이 시간표 계획에 고려해야하는 문제는 고정된 교과과목 교사의 시

간표와 유동적인 외부 강사의 시간과 요일 조건, 학년 군과 학년, 각 학기 별 수업 시수 충족이다.

반복되는 대면회의의 내용 때문에 피곤함을 호소하는 교사도 종종 나타난다. 이러한 시간표 짜기의 문제는 학생 수와 초등교사의 수가 많은 규모가 큰 학교로 갈수록 문제가 점점 불명확해지고 이를 결정하기 위한 대면회의가 늘어나기 때문에 점점 어려워진다.

이에 소프트웨어 프로그램을 적용하는 것은 복잡해 보이는 시간표 계획을 단순화하는데 좋은 해결책이 될 수 있다. 직접적으로는 대면회의의 횟수를 줄여줄 뿐만 아니라, 고려해야하는 문제를 분명히 하여 의사소통에서 불필요하게 발생하는 논쟁과 에너지 소모를 줄여줄 수 있다. 또한 학기 초마다 직접 반복해야 하는 업무를 일정부분 자동화할 수 있다면, 초등담임교사들이 더욱 세심하게 시간표를 계획하도록 도울 수 있다. 이는 학생들의 성장 발달 단계와 사회적 계기까지 고려하여 공교육의 질을 높이는 문제까지 연관된다.

## 2) 연구작품의 목표

앞서 밝힌 바와 같이 웹 서비스를 제공하여 초등교사의 시간표 계획 업무 부담을 줄이는 것을 목표로 한다. 현재 대면회의로 정해지는 학년별 시간표 계획 업무를 웹으로 일부 전환하여 합리적인 방법으로 하여금 시간표를 짜는 노고와 시간을 줄인다. 관계형 데이터베이스와 그 특성을 적절히 이용한다면 시간표를 체계적으로 관리하는데 일조할 것이다.

또한 이를 웹으로 서비스하는 것은 보안으로 인해 제한요소가 존재하는 학급 컴퓨터에서 프로그램에 대한 접근가능성을 높이는 일이다.

최종에 작품은 초등담임교사들의 시간표 생성에 들어가는 시간과 수고를 절약하여 초등학생이 받을 수 있는 교육의 질을 향상시키도록 돕는다.

### 3) Overview

초등학교의 시간표는 중고등학교와 달리, 대부분의 과목을 초등담임교사가 수업한다. 때문에 초등담임교사가 직접 본인 학급의 수업 시수를 조정하는 역할도 맡는다. 이에 프로그램이 시간표 내에서 유동가능성을 쉽게 찾을 수 있도록 돕는 일이 필요하다. 즉, 제약이 분명한 시간표를 제공하되, 변동사항을 빠르게 알려 담임교사가 고민해야 하는 문제의 수를 줄이는 것이 프로그램의 핵심이 된다.

본 프로그램은 다음과 같은 순서에 의해 작동한다.

1. 필수 정보가 학년 관리자에 의해 우선적으로 입력되어야 한다.
2. 이후 교과전담교사가 본인의 시간표를 생성하여, 반 별로 전담교과 수업이 입력된다.
3. 자동 시간표 생성 혹은 담임교사의 수동 입력으로 각 반 별의 시간표가 완성된다.
4. 이 후 학기 시작 전 후로 결정되는 초청강사의 수업의 유무에 따라 초청강사의 반별 시간표를 배정하고
5. 담임교사에게 그 변동사항을 알려 부족한 교과와 수업 시수가 무엇인지 명확히 알린다.

본 연구작품의 구현부는 크게 두 부분으로 나눌 수 있다. 첫째는, 전체 CRUD(Create, Read, Update, Delete) 생성이다. Django 프레임워크를 바탕으로 템플릿부터 데이터베이스까지 설계 및 구현한다. 이 때 단순히 객체 각각의 CRUD 만을 구현하는 것이 아니라, 같은 시간, 같은 요일, 같은 반에 해당하는 시간표는 서로 연동되도록 한다.

둘째는, 자동 시간표 생성 로직이다. 담임교사의 시간표와 초청강사에 대한 각각 자동시간표 생성 로직이 필요하다. 대학생 시간표와 다르게 시간표에 대한 선호도를 파악하는 것은 불가능하기에 추천시스템의 방법을 적용할 수 없다. 대신 교과전담

과목, 요일, 시간의 제약을 고려하고, 교과 시수에 대한 정보를 바탕으로 한 우선순위를 큐를 생성한다. 이를 바탕으로 월요일 1 교시부터 금요일 8 교시까지 빈 시간에 과목을 순차적으로 배정하도록 한다.

이번 중간 보고서는 다음과 같은 순서와 내용으로 작성되었다. 관련연구에서는 선행된 시간표 계획 프로그램을 소개하고 그 특징을 파악한다. 또한 Django 에서의 구현을 돕는 ORM 의 개념과 그 배경을 파악한다. 시간표 계획의 ~를 소개할 것이다.

제안 작품 소개에서는 UML 과 UML 시퀀스 프로세스 도표를 바탕으로 설계를 중심으로 하여 모델이 구성된 방식을 밝힌다.

구현 및 결과분석에서는 실제 웹페이지의 템플릿을 소개하며 만들어진 템플릿 뷰를 참조하여 구현이 어떻게 되었는지를 밝힌다.

## ■ 관련연구

### 1) 컴시간 주간시간표 프로그램

컴시간 주간시간표 프로그램은 컴시간 홈페이지(<http://comcigan.co.kr>)에서 배포하고 있는 중고등학교 주간시간표를 생성하는 프로그램이다. 학교 당 1 개에 한해서 사용기간 1 년을 두고 유료로 배포한다. 윈도우 실행 파일이며 온라인 컴시간 서버에 학교별 데이터베이스를 등록하여 그 내용을 학교 내에서 공유하는 방식이다(컴시간, 2003).

교사의 성명과 과목 학급을 등록한 후 요일 별 시수를 바탕으로 자동으로 시간표를 생성하고 수정하도록 하는 기능이 제공되어 있다. 이외에도 특별실에 관한 시간표와, 교과순배를 검색하여 문제점을 수정할 수 있는 기능도 가지고 있어, 중고등학교 시간표 생성에 적합한 프로그램이다. 또한 교육행정정보시스템(Neis)에 대한 대응도 갖추고 있어, Neis 에서 기초시간표 정보를 받거나 다시 업로드할 수도 있다. 다만 중고등학교에서는 초등학교와는 달리 학급의 담임교사가 대부분의 수업을 도맡아 하지 않는다. 다시 말해, 교과전담교사의 시간표만 존재할 뿐이며, 학년 혹은 학교 전체가 전체 시간표를 공유하는 형식이다. 따라서 이를 직접적으로 초등학교 시간표에 적용하기는 어렵다.

### 2) Object Relational Mapping (ORM)

Django에서는 자체적으로 ORM 기능을 제공하고 있다. Object Relational Mapping은 프로그래밍 언어에서 객체로 존재하는 모델을 데이터베이스의 실제 데이터와 매핑하는 일을 말한다. 이는 프로그래머로 하여금 직접적으로 데이터베이스에 접근할 여지를 줄여, 개발 비용을 낮추는데 일조한다. ORM은 처음 2002 년도에 Java 시스템에서 시작된 오픈 소스 프로젝트 Hibernate에서 시작되었으며 Microsoft의 .Net 시스템에 Entity Data Model에 합류되었다(O'Neil, 2008)

### 3) 시간표 계획

시간표 계획은 보통 NP-Complete Problem 으로 파악된다. 이에 시간표 계획에 유전알고리즘을 적용한 선행연구가 있다. 시간표 계획 문제를 효과적으로 자원을 분배하는 문제로 해석하여 이를 분배하는 방법으로 유전 알고리즘을 적용한 연구이다. 시간표를 시간과 날짜, 공간을  $x, y, z$  축으로 구분한 3 차원 표현으로 해석하여 유전알고리즘을 적용하였으며, 해당 알고리즘이 짧은 시간 안에 문제를 해결할 수 있음을 증명하였다. 또한 Intelligent Operators 를 추가할 때, 그 결과가 향상되는 양상 또한 내보였다(Sigl et al., 2003)

이외에도 시간표 계획에 대하여, 대학생 시간표에 추천시스템을 적용하는 연구가 다수 진행되었다. 추천 시스템이란 평점이나 과거 정보를 활용, 사용자의 선호를 파악해서 사용자에게 선호에 관한 대안을 제시하여 선택에 도움을 주는 것이다. (Ahmad & Sadiq, 2015) 추천 시스템 중 Collaborated 필터링은 User-based (Resnick et al., 1994), Item-based (Sarwar et al., 2001)로 나누어질 수 있다. User-based 는 다른 사용자를 참고하여 추천하는 것이고, Item-based 는 아이템을 선택한 사용자들의 양상을 비교하여 비슷한 아이템을 추천해 주는 것이다.

## ■ 제안 작품 소개

### 1) 작품 설계

해당 서비스에서는 다른 역할의 사용자(교사)에게 각기 다른 뷰를 보여주면서, 공통된 시간표를 공유하도록 하는 데이터베이스를 구성해야 한다. 이를 위한 UML 은 다음과 같다(그림 1)

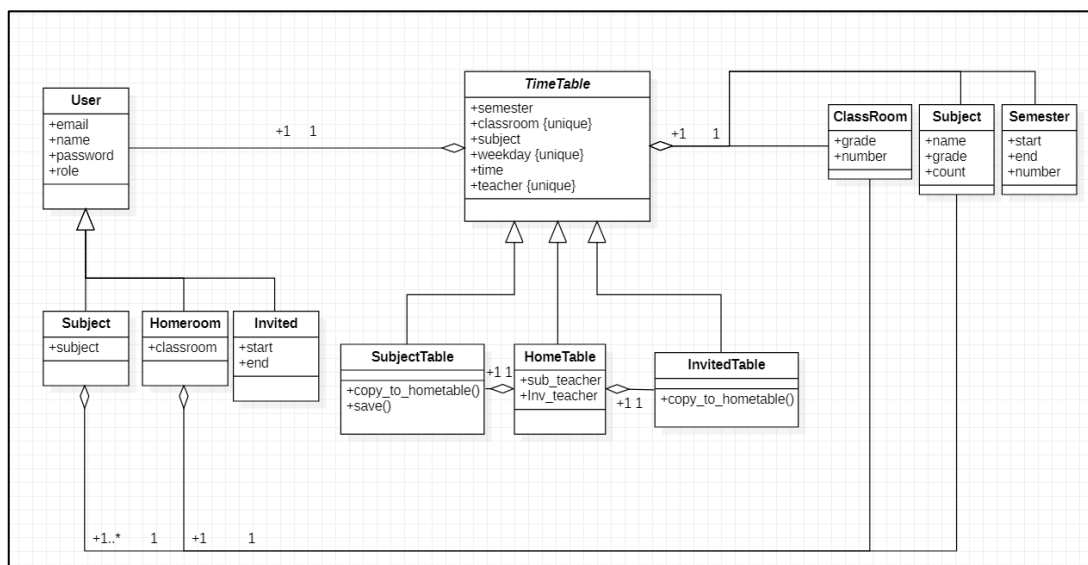


그림 1. 시간표 계획 프로그램의 UML

추상클래스인 TimeTable 을 제외한 모든 클래스는 데이터베이스 상에서 각각의 테이블이 존재한다. 이는 Django 와 PostgreSQL 의 자동화된 ORM 에 의해 생성된 결과이다. UML 에서 나타난 관계도는 데이터베이스 상에서도 Entity Model 로 일치하여 나타난다. TimeTable 을 상속하여 구현된 각각의 Table 들은 ClassRoom, Subject, Semester 를 외래키로 가진다. 또한 User 를 상속받아 구현된 Subject, Homeroom 도 각각 다대다 관계, 일대일 관계로, ClassRoom 과 Subject 테이블을 참조한다.

User 를 상속받아 구현된 Subject, Homeroom, Invited 는 role 에 기반하여 각 자신의 시간표인 SubjectTable, HomeTable, InvitedTable 에 관해서만 Transaction 을 할 수 있다.



앞서 밝힌 바와 같이 반(ClassRoom)에 대한 시간표는 반의 담임교사(Homeroom)를 기준으로 생성되며, 반과 담임교사는 일대일 관계로 매칭되어 있다. 반면 과목(Subject)과 교과전담교사(SubjectTeacher)는 다대다 관계로 구성되며, 교과전담교사가 복수의 학년별 교과를(예를 들면 3,4 학년 영어 등) 가질 수 있도록 한다.

초청강사는 학기 중 특정시간에만 학교에 출근하고, 담임교사가 정해진 시간표에 따라 수업을 하는 특징을 지닌다. 초청강사는 기존의 수업에 특별히 참가하여, 담임교사의 수업을 대체하므로, 초청강사 유형에 해당하는 사용자는 시작과 끝으로 출근 가능 기간의 데이터를 가진다.

해당 모델에서 가장 중점적인 반별 시간표는(HomeTable)은 교과전담교사의 시간표와 초청강사의 시간표를 일대일 관계로 참조한다. 일대일 참조 필드는 Null로 설정될 수 있으며, 이 경우 담임교사가 본 반의 수업을 맡는 것으로 생각한다. 항상 동일한 시간과 요일, 반 학년번호를 참조해야만 한다.

## 2) 시퀀스 다이어그램

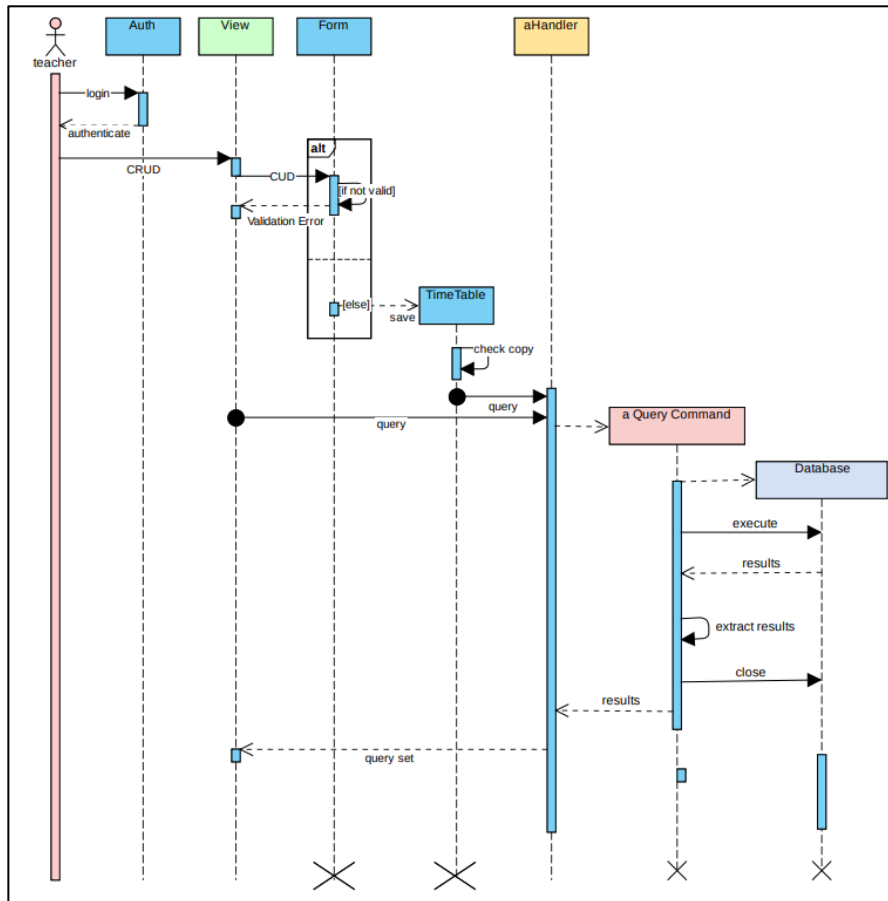


그림 2. Django를 기반으로 만든 시간표 계획 시퀀스 다이어그램

본 프로그램의 시퀀스 다이어그램은 그림 1 과 같다. 사용자인 Authentication 모듈로부터 인증을 받으면, 각 역할에 따라 자신의 시간표에 관한 CRUD(Create, Read, Update, Delete)를 할 수 있게 된다. 사용자가 입력한 데이터의 타당성 검사는 Form 객체에서 이루어지게 된다. 입력에 대한 제약은 더 추가할 필요가 있으며 현재까지 구현된 제약은 3)번 문항에서 밝히도록 한다.

시간표의 생성, 업데이트, 삭제 시에는 시간표의 해당 시간, 요일, 반과 일치하는 다른 시간표도 참조한다. 예를 들어, 교과전담교사의 시간표를 업데이트할 때 동시에, 교사가 수업하는 반의 시간표, 즉 담임교사의 시간표도 업데이트 한다.

### 3) 타당성 검증

앞서 밝힌 대로 사용자가 입력한 데이터는 Form 객체에서 그 타당성을 검사한다. 현재까지 생성한 제약은 다음과 같다.

1. 사용자, 시간표 시간, 시간표 요일의 Unique Constraint
2. 담임교사가 본인 반에 복사된 교과전담교사의 시간표를 수정할 수 없음
3. 담임교사가 특정 과목의 시수를 기준보다 적게 하여 시간표를 생성할 수 없음
4. 다른 교과전담교사에 의해 이미 참조된 담임교사 시간표가 존재할 때, 교과전담교사는 해당시간표를 중복해서 생성하거나 수정, 삭제할 수 없음

### 3) 데이터 저장방법

시간표가 사용자 역할별로 존재하고 서로 참조하기 때문에, 시간표 객체의 인스턴스화가 모두 동시에 진행될 수 없다. 시간표의 인스턴스화 순서는 차례대로 반 별 시간표(HomeTable), 과목별 시간표(SubjectTable), 초청강사 시간표(InvitedTable)이다.

1. 학기에 해당하는 반 별 시간표(HomeTable)를 과목을 비운 채로 반이 생성될 때 인스턴스화한다.
2. 과목별 시간표(SubjectTable)를 생성할 때, 입력 받은 반 정보를 바탕으로 동 요일, 동 시간 대의 HomeTable 을 참조하여 업데이트 한다.
3. 반 별 시간표(HomeTable)가 과목별 시간표(SubjectTable)를 참조하고 있다면 해당 시간표를 삭제, 수정하는 것은 불가능하다.
4. 초청강사 시간표(InvitedTable)는 반 별 시간표가 생성된 이후에 생성 가능하다.

## ■ 구현 및 결과분석

### 1) View

Scheduler

개인정보

학기관리

반관리

교과관리

Your are logged in as 3\_1담임 [Log out](#)

Homeroom

#	Mon	Tue	Wed	Thr	Fri
1		영어(3G) 교과선생님1	영어(3G) 교과선생님1	영어(3G) 교과선생님1	
1			영어(3G) 교과선생님1	예술(음악/미술)(3G) 교과선생님2	영어(3G) 교과선생님1
1					영어(3G) 교과선생님1
1					
1					
1				영어(3G) 교과선생님1	영어(3G) 교과선생님1
1		영어(3G) 교과선생님1	영어(3G) 교과선생님1		

create update

그림 3. 교과전담교사의 시간표가 적용된 담임교사 시간표 뷰

시간표를 볼 수 있는 템플릿 뷰는 그림 3 과 같다. 월-금과 1-8 교시를 기본으로 한 5x8 테이블로 주간시간표를 보여준다. 좌측에 고정된 네비게이션에 보이는 링크로 사용자와 학교에 관한 정보에 접근할 수 있으며, 링크 해당 정보를 생성, 수정, 삭제할 수 있는 템플릿과 연동되어 있다. 좌측 네비게이션에 위치한 링크는 필수정보로, 정보가 입력되지 않았다면 테이블 뷰가 아니라 관리 페이지로 리다이렉션 된다.

상단 네비게이션에는 홈페이지로 가는 제목 링크가 좌측에 로그인 정보가 우측에 표기된다.

하단의 create, update 버튼은 각각 현재 보이는 해당시간표의 생성 및 수정할 수 있는 템플릿으로 연결되어 있다.

## 2) Form

Scheduler

Your are logged in as 교과선생님1 [Log out](#)

Subject

#	Mon	Tue	Wed	Thr	Fri
1	Classroom <input type="text"/>	Classroom <input type="text"/>	Classroom <input type="text" value="3-1"/>	Classroom <input type="text" value="3-1"/>	Classroom <input type="text"/>
	Subject <input type="text"/>	Subject <input type="text"/>	Subject <input type="text" value="영어(3G)"/>	Subject <input type="text" value="영어(3G)"/>	Subject <input type="text"/>
	Id	Id	Id	Id	Id
	Teacher	Teacher	Teacher	Teacher	Teacher
1	Classroom <input type="text"/>	Classroom <input type="text"/>	Classroom <input type="text" value="3-1"/>	Classroom <input type="text"/>	Classroom <input type="text" value="3-1"/>
	Subject <input type="text"/>	Subject <input type="text"/>	Subject <input type="text" value="영어(3G)"/>	Subject <input type="text"/>	Subject <input type="text" value="영어(3G)"/>
	Id	Id	Id	Id	Id
	Teacher	Teacher	Teacher	Teacher	Teacher
1	Classroom <input type="text"/>	Classroom <input type="text"/>	Classroom <input type="text"/>	Classroom <input type="text"/>	Classroom <input type="text" value="3-1"/>
	Subject <input type="text"/>	Subject <input type="text"/>	Subject <input type="text"/>	Subject <input type="text"/>	Subject <input type="text" value="영어(3G)"/>

그림 4. 교과전담교사의 시간표 업데이트 뷰

Form 은 각 객체 모델별로 CRUD 총 네 가지 존재한다. 시간표로 보여줘야할 form 은 8x5 테이블의 총 40 개이므로, 한 html 폼에서 한 번에 생성, 수정할 수 있도록 하기 위해 Django 내 존재하는 Formset 함수를 이용하여 관리한다(부록 참조). Set 안에 존재하는 Form 들은 html 에서의 <tr> <td> 문법을 따라 행을 중심으로 렌더링 된다. 즉, Formset 에서 시간표는 월 1 교시, 화 1 교시, 수 1 교시, ... 월 2 교시 순서로 저장되어 있으며, 이는 데이터베이스에 저장할 때도 같은 순서이다.

업데이트 시에는 form 에 초기값을 입력하기 위해, 기존에 있는 데이터에 ORM 을 통해 쿼리문을 보내고, 이를 바탕으로 form 을 인스턴스화 하여 수정되기 이전의 데이터 값을 템플릿에도 보일 수 있도록 한다(그림 4)

## 2) 데이터베이스의 구성

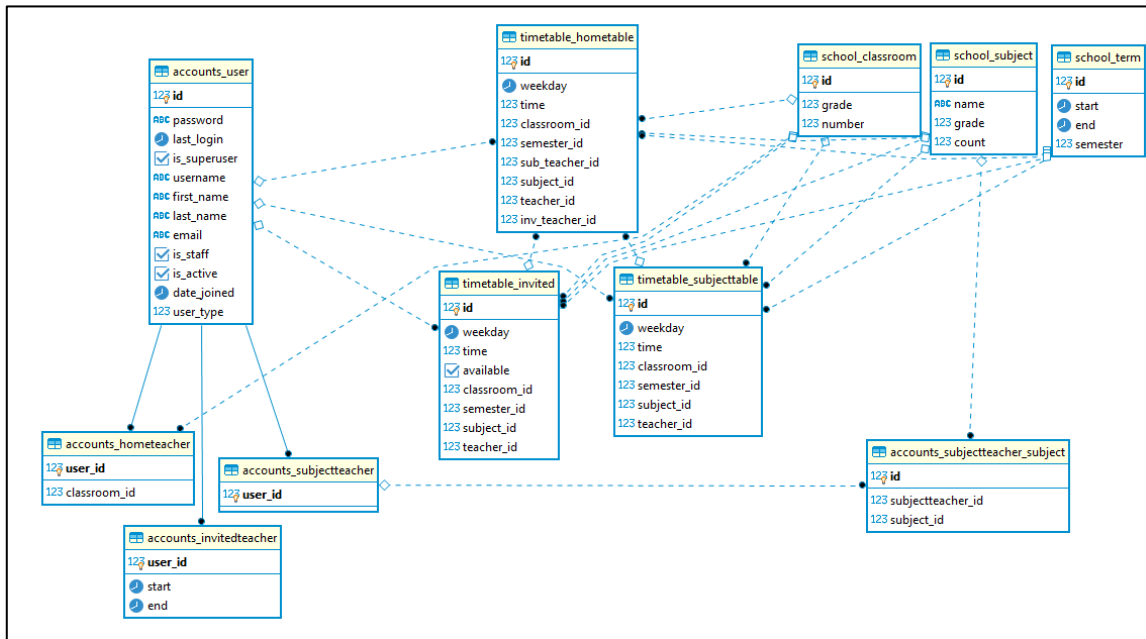


그림 5. ER Diagram

앞선 UML 을 바탕으로 Django 에서 Migrate 한 데이터베이스의 ER Diagram 은 다음과 같다(그림 5). TimeTable 을 기반으로 각각 timetable\_hometable, timetable-subjecttable, timetable\_invited 가 생성되었음을 알 수 있다. 또한 다대다 관계로 구성된 SubjectTeacher 와 Subject 는 중간 테이블 accounts\_subjectteacher\_subject 에 의해 관리된다.

사용자는 Django 에서 제공하는 기본 사용자 모델을 Proxy 모델로 확장하고 각각의 역할별로 구분하였다. 이는 Django 에서 기본으로 제공하는 Authentication 을 적극활용하기 위함이다.

### 3) 자동시간표 생성

#### 3-1) 담임교사 시간표 생성 – 우선순위 힙

우선순위가 내림차순이라 가정하여 알고리즘을 작성한다. 우선순위 큐는 해당 학년의 교과전담으로 배정된 과목을 제외한 전부로 구성된다. 시간표에 교과가 배정되면 이는 주마다 반복되므로 해당 요일의 행사와 쉬는 날을 뺀 주 횟수를 곱한 만큼 수업 시수가 채워지게 된다. 이에 해당 교과는 채워진 수업 시수를 교과 시수에서 뺀 만큼의 우선순위를 갖게 된다. 그러나 미술과 같은 교과 외에는 과목이 중복되면 안되므로 특정 감쇠 값을 곱해 이미 한 번 배정되었음을 나타낸다. 새로이 계산되는 수식은 다음과 같이 나타낼 수 있다.

$$\text{priority} = c * (\text{subject count} - (\text{priority} * \text{weeks} - \text{count of holis and events}))$$

이렇게 업데이트 된 우선순위 힙을 바탕으로 월요일 1 교시부터 금요일 8 교시까지 비어 있는 시간에 순차적으로 과목을 적용한다.

#### 3-1) 초빙교사 자동 시간표 생성

초빙교사는 시간표 배정은 우선순위보다 제약조건을 먼저 고려하게 된다. 초빙교사는 출근 가능 시간만 제공하며, 학년별 담임교사가 할당된 시간에 배정가능성을 확인하여 초빙교사의 시간표를 계획한다. 이 때 담임교사는 학급 시간표를 자율적으로 유동성 있게 변경할 수 있으나, 교과전담시간과는 교체할 수 없다. 이를 고려하여 초청 강사의 수업 시수를 반 별로 공평하게 분배하는 것이 해당 자동시간표 생성의 핵심이다.

초빙교사의 시간표는 단순히 1 반부터 순차적으로 배정하되 해당 시간표에 SubjectTable 이 Null 이 아닌 경우에만 가능하도록 한다. 배정된 후에는 해당하는 시간의 과목과 시수를 변경한다.

## ■ 결론 및 소감

현재 프로젝트는 절반 진행되었다. 앞으로 해야 할 일은 자동 시간표 생성을 완성하고, 감수를 받아 피드백을 반영한 후, 배포하는 것이다. 완성된 프로그램을 9 월 중에 마무리하고 감수 후 수정하면서 배포한다.

현재 초등교사들이 시간표 계획 프로그램을 사용한다면 가장 불편할 요소는 교육행정정보시스템 Neis 와의 연동이다. 엑셀 파일을 출력하지 않는 한 Neis 상에 주마다 수기로 시간표를 입력해야 하기 때문에 시간표를 이중으로 운용해야 하는 불편함을 낳는다. 다만 이는 목표로 했던 것에서의 추가 기능이기에 배포와 동시에 기능의 구현가능성을 고민해본다.

진행하면서, 기능을 코드 어느 부분에 구현해야 기존 Django 문법에 맞는 코드인지, 어떻게 구성해야 프로젝트가 명료해지는지 고민하는데 시간을 많이 소모했다. 이는 Django 문법과 개발에 친숙하지 않은 본인 역량의 미흡함이라고 할 수 있겠다.

이외에 실제로 구현을 하다 보니 필요하다고 생각되는 기능들이 계속해서 추가된다. 또한 이 과정에서 개발자와 사용자의 입장의 구분이 명확하지 못하여, 실제로 필요한 기능을 파악하는 일이 모호해졌다. 이러한 문제는 프로그램에 대한 청사진을 미리 계획하지 않아서 벌어진 일로 파악한다.

사전조사가 미흡한 부분도 이러한 문제가 발생하는데 일조한다. 구현하고자 하는 프로그램이 무엇인지 분명히 하고 꼭 필요한 기능을 우선적으로 구현하는 능력이 필요하다.



## ■ 참고문헌

[1] 컴시간, "컴시간 주간시간표 프로그램", <http://comcigan.co.kr/xe/jdata/19458>, 2016

[2] 컴시간, "컴시간 주간시간표 프로그램 사용설명서", 2020.09.08., <http://comcigan.co.kr/xe/jdata/19458>

[3] Django Software Foundation, "Django Official Documentation", 2020.09.08., <http://doc.Djangoproject.com>

[4] O'Neil, Elizabeth J. "Object/relational mapping 2008: hibernate and the entity data model (edm)." Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008.

[5] Sigl, Branimir, Marin Golub, and Vedran Mornar. "Solving timetable scheduling problem using genetic algorithms." Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.. IEEE, 2003.

[6] Shahnawaz Ahmad, Mohd. Sadiq, "Recommender Systems for Software Requirements Negotiation and Prioritization", International Journal of Computer Applications (0975-8887) Volume 117-No.13, 2015.

[7] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J., "Grouplens: An open architecture for collaborative filtering of netnews", Conference on Computer Supported Cooperative Work, 1994.

## ■ 부록 (원천코드)

### 1. Formset 의 인스턴스화와 그 활용

```
class SubjectUpdate(LoginRequiredMixin, generic.UpdateView):
    """
    class doc
    """
    template_name = 'sub_update.html'
    model = SubjectTable
    form_class = SubjectTableForm

    def get_context_data(self, **kwargs):
        context = super(SubjectUpdate, self).get_context_data(**kwargs)
        qs = SubjectTable.objects.filter(teacher=self.request.user,
        weekday__range=("2020-08-31", "2020-09-04"))
        if self.request.POST:
            context['TimeTables'] =
SubjectTableUpdateFormset(self.request.POST, instance=self.request.user,
        queryset=qs)
        else:
            context['TimeTables'] =
SubjectTableUpdateFormset(instance=self.request.user, queryset=qs)
        return context

    def form_valid(self, form):
        context = self.get_context_data()
        formset = context['TimeTables']
        if formset.is_valid():
            for form in formset:
                instance = form.save(commit=False)
                instance.save()
                #if instance.classroom and instance.subject:
                #    copy_sub_to_home(instance)
            messages.success(self.request, 'success', extra_tags='alert')
            return redirect(self.get_success_url())
        else:
            messages.warning(self.request, formset.errors)
            return
        self.render_to_response(self.get_context_data(form=form))

    def form_invalid(self, form):
        context = self.get_context_data()
        formset = context['TimeTables']
        print(formset.errors)
        return self.render_to_response(self.get_context_data(form=form))
```