Team Project - OOAD

Timetable

(instructor :: 유준범 교수님)

Konkuk univ. dept of CSE

Team 4

200611517 정훈섭

200711420 권준수

200711448 오희수

200710118 유희찬

Team Project - Timetable development

Contents

Motivation Draft plan Requirement Prototype Analysis

지난 여름 건국대학교 컴퓨터공학부 1학년 학생들이 timetable 프로그램 '암고나메잌ㅋ' 을 배포한 이후 게시판 및 블로그에 많은 댓글이 달렸습니다. 대부분 '잘 쓰겠다' 라는 내용이었지만 중간 중간 '차후 버전에서는 ~해 달라' 는 의견이 종종 눈에 띄어, 이번 S/E 수업 team project 진행 겸 평소 눈여겨보았던 사항이라 이 의견들을 수렴하여 timetable을 제작하게 되었습니다.

Motivation **Draft plan** Requirement Prototype Analysis

Goal

'간편하고 쓰기 쉬운' 시간표 프로그램을 목표로 하고 있습니다. 복잡하지 않고 직관적으로 사용이 가능한 시간표 프로그램입니다.

Determining

프로젝트 착수 여부를 결정하기 전에 현재 내부 상황을 확인합니다.

인력	4명으로 구성된 팀으로 앞서 팀 과제를 수행을 통해 팀워크와 개개인의 실력이 입증 됨.
기간	충분한 편은 아니지만 프로젝트 범위를 크게 잡지 않을 계획이어서 계획상으로는 기간 안에 끝낼 가능성이 높다.
범위	비교 대상인 '암고나메잌ㅋ' 을 보다 간략화 해서 개발할 것이므로 범위는 좁아진다.

프로젝트의 범위가 너무 크거나 진행 기간이 짧으면 완성 기간을 지키지 못할뿐더러 개발 자체를 끝내지 못할 수도 있습니다. 또한 개발 기간에 맞춰서 무리하게 진행하다보면 전반적으로 software의 품질이 떨어질 수 있는 바, 현재 상황을 꼼꼼하게 확인했습니다.

이번 프로젝트는 진행에 무리가 없다고 판단하여 이번 프로젝트를 진행하기 결정합니다.

Scope

프로젝트의 목표가 기존의 프로그램인 '암고나메일ㅋ' 을 수정 및 보완하는 것으로 시작하였습니다. 물론 가장 큰 위험으로 프로젝트 진행 기간이 부족하다는 것을 고려하여 '암고나메일ㅋ' 과는 다르게 제공되는 강의는 컴퓨터공학부에 한정하기로 했습니다. 또한 화려한 GUI는 생략하고 간략하고 깔끔한 인터페이스를 제작하였습니다.

User Characteristic

시간표 프로그램을 사용하게 될 주사용자는 건국대학교 학생입니다. 시간표 프로그램은 늘 상 쓰이는 프로그램도 아니어서 항상 필요가 있는 것은 아니지만 수강신청 같이 특정 날에 대해 있으면 편리한 프로그램이기 때문에 엄청난 수요를 형성합니다.

대부분이 컴퓨터에 익숙하여 왠만한 프로그램은 무리없이 쓰지만, 사용하기 복잡한 프로그램은 기피합니다. 또한, 잘 갖춰진 프로그램을 주로 쓰다가 시간표 프로그램처럼 그냥 배포되는 잘 갖춰지지 않은 프로그램에 대해 금방 불만을 가지기 쉽습니다.

Scheduling

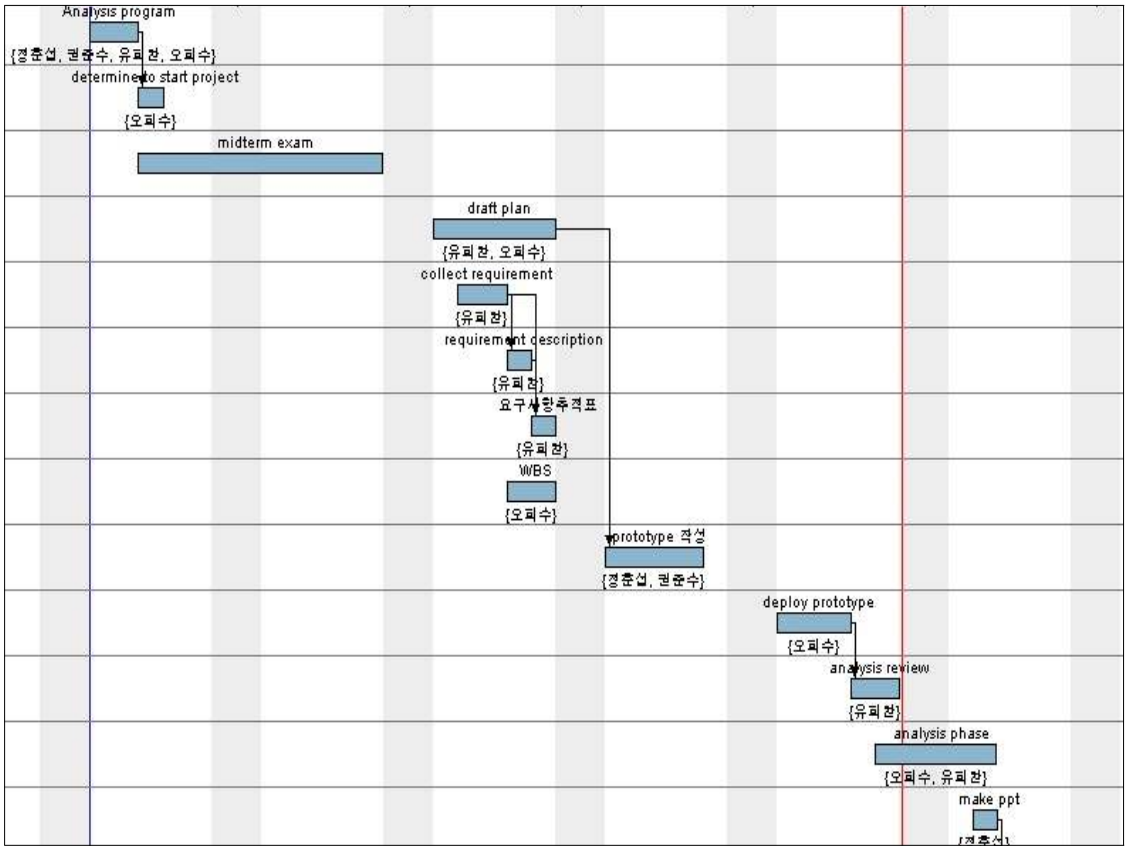
업무 분담

본격적인 프로젝트 착수에 앞서 프로젝트 내 과업를 세세히 정의하여 진행하는 동안 누락되는 과업이 없도록 WBS(**W**ork **B**reak **S**tructure)을 작성합니다. 먼저 4명의 팀원의 업무를 분담했습니다.

오희수	Project Manager
권준수	Developer
정훈섭	Developer
유희찬	Analyst

WBS

이번 프로젝트에서 수행되어야할 과업들을 정리하여 표로 정리했습니다. 이를 바탕으로 각 과업의 시작 날, 끝난 날, 과업을 맡은 인원을 한 눈에 알아볼 수 있도록 프로젝트 관리 프로그램을 이용하여 WBS를 최종적으로 완성했습니다.

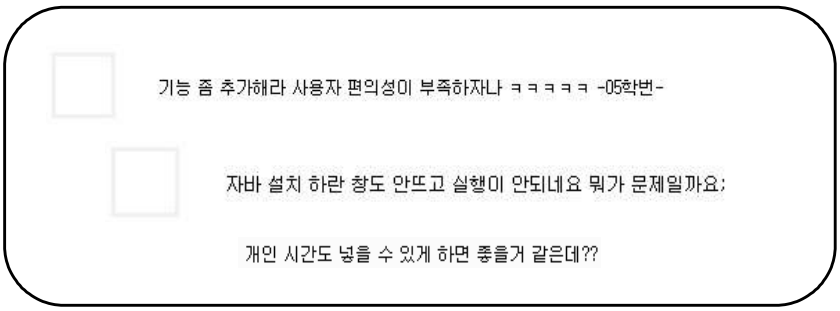


WBS를 작성하면 앞서 말했듯 과업의 누락을 예방할 수 있고, 팀원 개개인이 해야 할 업무를 할당할 수 있다는 면에서 이점이 있습니다. 또한 WBS를 통해서 프로젝트 시간 관리도 할 수 있습니다.

Motivation **Draft plan** **Requirement** **Prototype** **Analysis**

프로젝트 진행 첫 단계로 우선 '암고나메일ㅋ' 을 배포한 사이트인 '건이네' 와 해당 블로그에서 게시글에 달린 요구사항 댓글을 수집합니다. 1차적으로 댓글을 통하여 요구사항을 수집하고, 이를 토대로 prototype을 만들어 간단한 설문과 함께 주위에 배포하여 feedback을 받습니다.

게시판 댓글에서 수집한 요구사항들은 다음과 같습니다.



이를 토대로 requirement specification을 만듭니다.

Requirement specification

1. 학교 강의 외 개인 일정(언교원, 학원, 봉사활동 등) 추가 가능
2. Java의 jdk 추가 설치 없이 가볍게 파일 하나로 실행
3. 직관적으로 알 수 있는 사용 방법

이중에서 1번은 system의 기능적인 면을 요구하는 functional requirement이고 2번과 3번은 non-functional requirement에 속합니다. 2번은 system의 이식성과 관련된 portability와 연관되어 있습니다.

요구사항들이 프로젝트 진행동안 제대로 반영되었는지 확인하기 위하여 요구사항 추적표를 작성합니다. 후에 개발 과정 동안 추가되는 요구사항들 역시 요구사항 추적표에 추가하여 관리합니다.

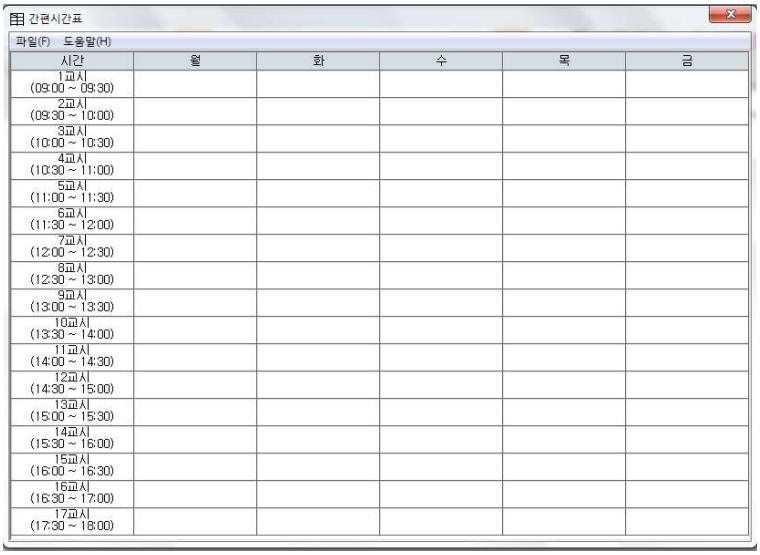
번호	Analysis phase	Design phase	Implement phase	비고
R.1	Use case 작성	class diagram 작성	function test	
R.2	N/A	class diagram 작성	UML tool에서 C++로 generation	
R.3	Use case 작성	class diagram 작성	function test	

요구사항 추적표를 작성하여 각 요구사항이 언제 어떻게 적용되어 system에 구현되고 있는지 알 수 있습니다. 예로 들어 R.1 즉, '개인 일정 추가' 라는 요구사항은 Analysis 단계에서는 개별 기능에 대한 Use case를 작성함으로써 system의 한 기능으로 정의되고 있고, Design 단계에서 작성된 class diagram에서 timetable class의 한 method로써 설계될 것입니다. 마지막으로 implement 단계에서 functional test를 통하여 '개인 일정 추가' 기능이 제대로 수행되고 있는지 확인을 할 것입니다.

Motivation **Draft plan** **Requirement** **Prototype** **Analysis**

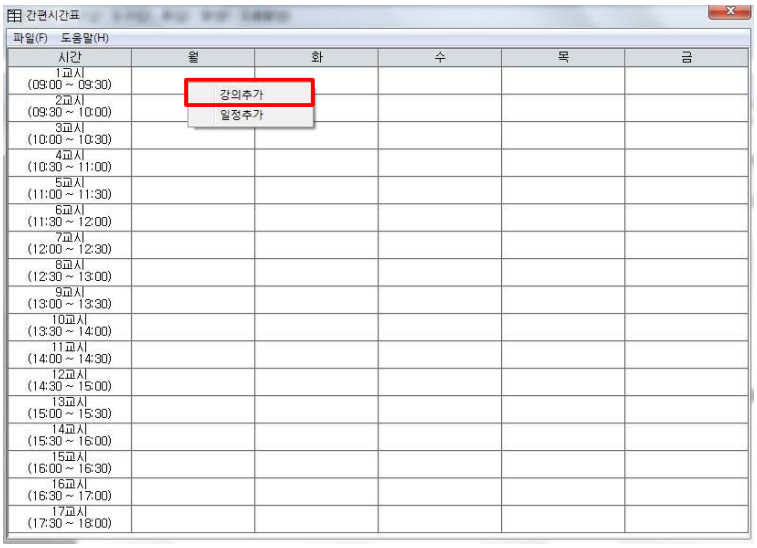
prototype

지금까지 나온 요구사항들을 통해, 우리가 생각하는 timetable을 prototype을 제작했습니다. 내부적으로는 팀원끼리는 물론, 지인들에게 배포하여 prototype을 통해 요구사항들이 정확히 구현되어 있는지 확인을 해봅니다.

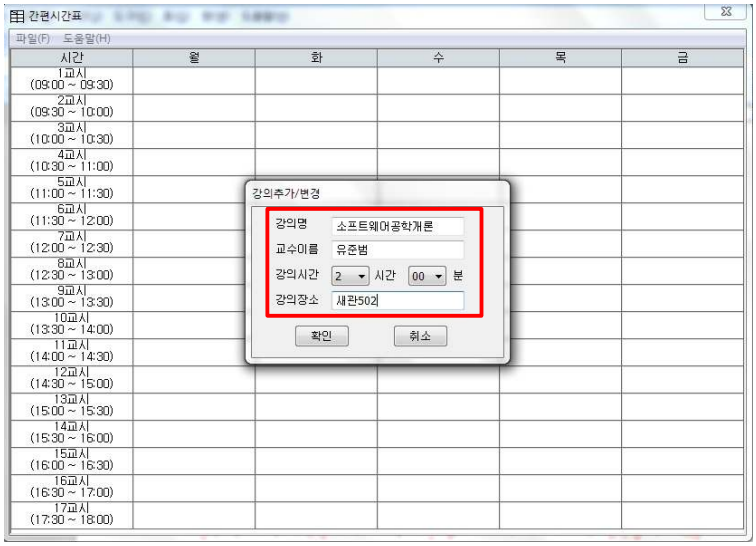


위의 요구사항에 나온 기능을 하나씩 해봄으로써 제대로 역할을 수행하는지 시험해 봤습니다.

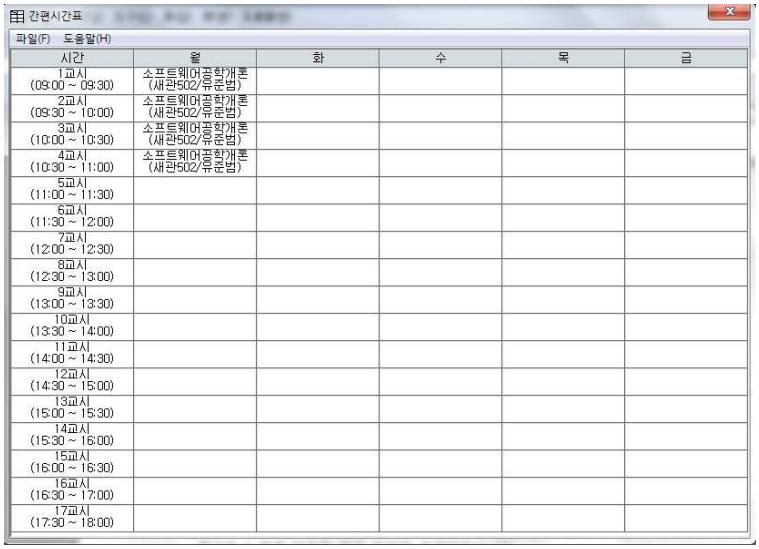
월요일 1교시 '소프트웨어 공학 개론' 수업을 추가해보겠습니다. 월요일 1교시에 커서를 대고 오른쪽 클릭을 하여 '강의 추가' 를 선택합니다.



강의 이름에 소프트웨어 공학 개론, 교수명에 유준범, 장소는 새천년관 502호, 시간은 2시간을 선택합니다.

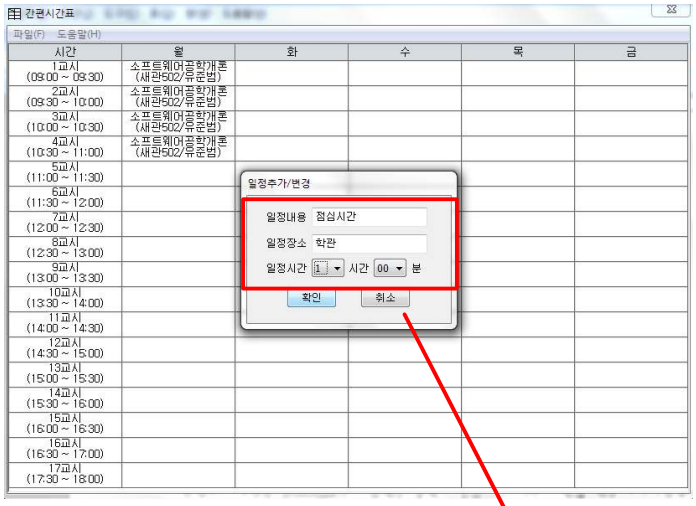


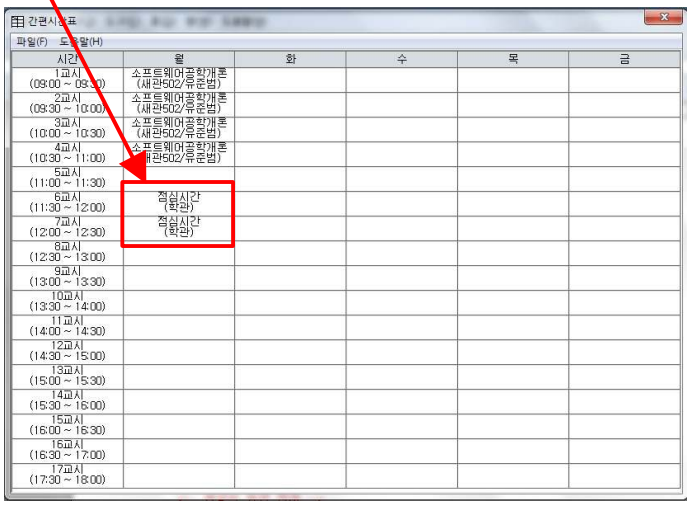
확인을 누르면 다음과 같이 강의가 추가되었습니다.



시간	월	화	수	목	금
1교시 (09:00 ~ 09:30)	소프트웨어공학개론 (새관502/유준범)				
2교시 (09:30 ~ 10:00)	소프트웨어공학개론 (새관502/유준범)				
3교시 (10:00 ~ 10:30)	소프트웨어공학개론 (새관502/유준범)				
4교시 (10:30 ~ 11:00)	소프트웨어공학개론 (새관502/유준범)				
5교시 (11:00 ~ 11:30)					
6교시 (11:30 ~ 12:00)					
7교시 (12:00 ~ 12:30)					
8교시 (12:30 ~ 13:00)					
9교시 (13:00 ~ 13:30)					
10교시 (13:30 ~ 14:00)					
11교시 (14:00 ~ 14:30)					
12교시 (14:30 ~ 15:00)					
13교시 (15:00 ~ 15:30)					
14교시 (15:30 ~ 16:00)					
15교시 (16:00 ~ 16:30)					
16교시 (16:30 ~ 17:00)					
17교시 (17:30 ~ 18:00)					

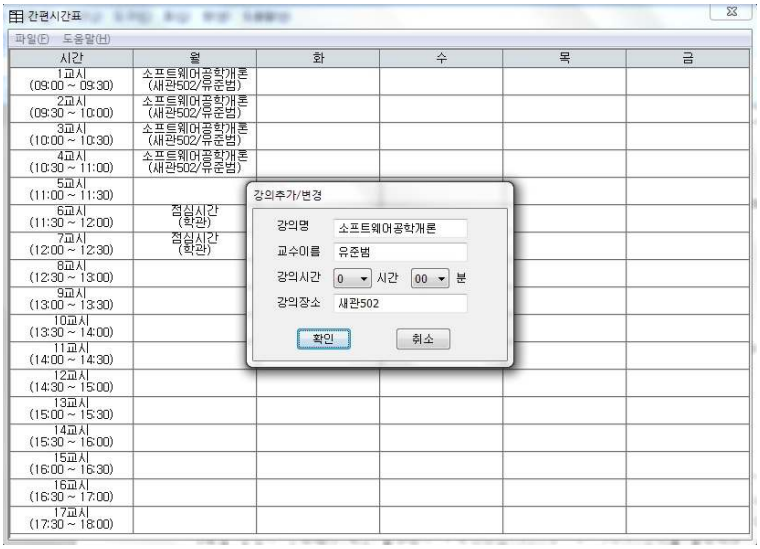
같은 방식으로 수업이 끝나고 점심을 먹으러 가는 일정을 추가해봅니다.



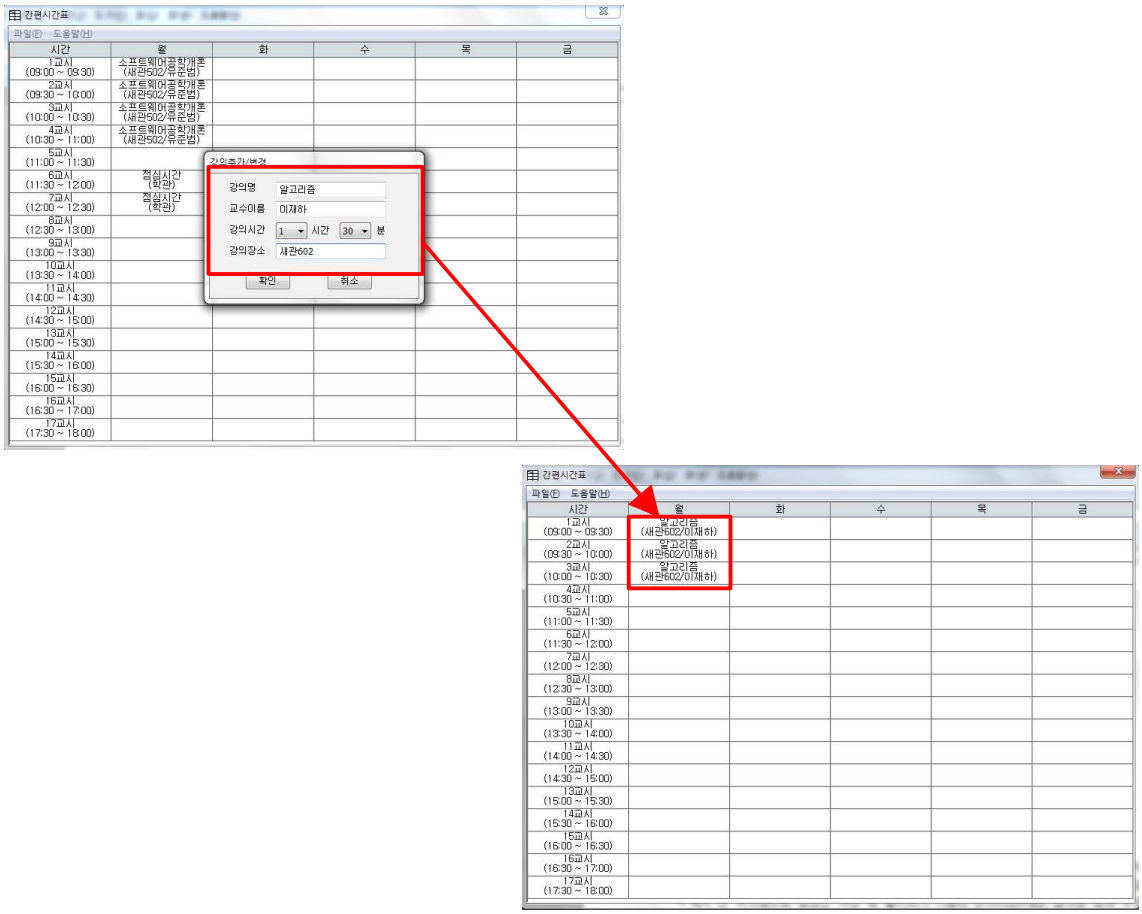


시간	월	화	수	목	금
1교시 (09:00 ~ 09:30)	소프트웨어공학개론 (새관502/유준범)				
2교시 (09:30 ~ 10:00)	소프트웨어공학개론 (새관502/유준범)				
3교시 (10:00 ~ 10:30)	소프트웨어공학개론 (새관502/유준범)				
4교시 (10:30 ~ 11:00)	소프트웨어공학개론 (새관502/유준범)				
5교시 (11:00 ~ 11:30)					
6교시 (11:30 ~ 12:00)	점심시간 (학관)				
7교시 (12:00 ~ 12:30)	점심시간 (학관)				
8교시 (12:30 ~ 13:00)					
9교시 (13:00 ~ 13:30)					
10교시 (13:30 ~ 14:00)					
11교시 (14:00 ~ 14:30)					
12교시 (14:30 ~ 15:00)					
13교시 (15:00 ~ 15:30)					
14교시 (15:30 ~ 16:00)					
15교시 (16:00 ~ 16:30)					
16교시 (16:30 ~ 17:00)					
17교시 (17:30 ~ 18:00)					

다음으로 일정 변경을 해봅니다. 월요일 1교시 '소프트웨어 공학 개론' 수업을 '알고리즘' 수업으로 변경해보겠습니다. 강의를 오른쪽 클릭하여 '강의 변경' 메뉴를 선택합니다.



변경될 내용(알고리즘 수업)을 입력하고, 확인을 클릭하면 다음과 같이 변경 됩니다.



현재까지 구현된 prototype의 기능에는 문제가 없습니다. 다만 시험을 해본 지인으로부터 몇가지 요구사항이 더 나왔고, 팀 내에서도 새로운 아이디어가 나왔습니다. 우선, '암고나메일ㅋ'에서 큰 아쉬움으로 남았던 저장 및 불러오기 기능이 prototype에는 없다는 점과 수업 과목을 일일이 입력해야 하는 불편함이 지적되었습니다. 여기서 나아가 강의를 클릭하면 강의계획서를 볼 수 있는 기능을 추가하자는 팀 내 아이디어가 나왔습니다.

새로운 3가지 요구 사항을 접수하고, 3가지 요구사항이 모두 프로그램의 목적인 '편리한'에 부합하고, 프로젝트 마감일까지의 기간이 남아있어 모두 수렴하기로 했습니다. 단, 학과별 강의 목록은 프로젝트 범위 상 컴퓨터공학부의 강의 목록은 한정하였습니다. requirement description, 요구사항 추적표를 새로 수정합니다.

Revised Requirement specification

1. 학교 강의 외 개인 일정(언교원, 학원, 봉사활동 등) 추가 가능
2. Java의 jdk 추가 설치 없이 가볍게 파일 하나로 실행
3. 직관적으로 알 수 있는 사용 방법
4. 저장/불러오기 기능 추가
5. 학과별 강의 목록 출력(컴퓨터공학부 강의로 한정)
6. 강의계획서 보여주기

<< 수정된 요구사항 추적표 >>

번호	Analysis phase	Design phase	Implement phase	비고
R.1	Use case 작성	class diagram 작성	function test	
R.2	N/A	class diagram 작성	UML tool에서 C++로 generation	
R.3	Use case 작성	class diagram 작성	function test	
R.4	Use case 작성	class diagram 작성	function test	
R.5	Use case 작성	class diagram 작성	function test	
R.6	Use case 작성	class diagram 작성	function test	

수집된 요구사항 들을 가지고 use case를 작성합니다.

Revised Requirement specification

1. 사용자가 프로그램을 실행 한다
2. 사용자는 저장된 시간표를 불러오거나 새로이 작성한다.
 2-1 사용자가 시간표를 새로이 작성한다.
  2-1-1 사용자는 등록하고 싶은 시간의 칸을 선택한다.(right click)
   2-1-1-1 사용자가 선택한 칸에 일정이 있다.
    2-1-1-1-1 '강의 등록' 을 선택한다.
     2-1-1-1-1-1 강의명, 교수명, 강의실, 시간을 입력한다.
     2-1-1-1-1-2 입력을 다했으면 저장한다.
    2-1-1-1-2 '일정 등록' 을 선택한다.
     2-1-1-1-2-1 활동명, 장소, 시간을 입력한다.
     2-1-1-1-2-2 입력을 다했으면 저장한다.
  2-1-2 사용자가 선택한 칸이 비어있다.
   2-1-2-1 '변경' 을 선택한다.
   2-1-2-2 '삭제' 를 선택한다.
 2-2 사용자가 저장된 시간표를 불러온다.
  2-2-1 위의 내용과 동일한 작업 반복
3. 사용자가 시간표를 저장한다.
4. 사용자가 시간표를 종료한다.

위의 use case를 보면 프로그램을 이용하게 되는 사람은 사용자 한 명입니다. 따라서 use case의 actor는 user 한 명이고, system이 수행하는 performance로 save/load, add schedule, change schedule, add lecture, delete lecture, delete schedule이 있습니다.

use case diagram을 그리기 전에 system이 수행하는 performance를 개별 use case를 통해서 보다 자세하게 알아봅니다.

기능별 use case

Use case	save
actor	user
type	default
Description	User는 시간표를 작성하고, 후에 작성된 시간표를 다시 볼 수 있도록 'save' 버튼을 눌러 저장한다.

Use case	load
actor	user
type	default
Description	사용자는 'load' 버튼을 클릭한다. 저장되어 있던 기존 시간표를 선택하여 불러온다.

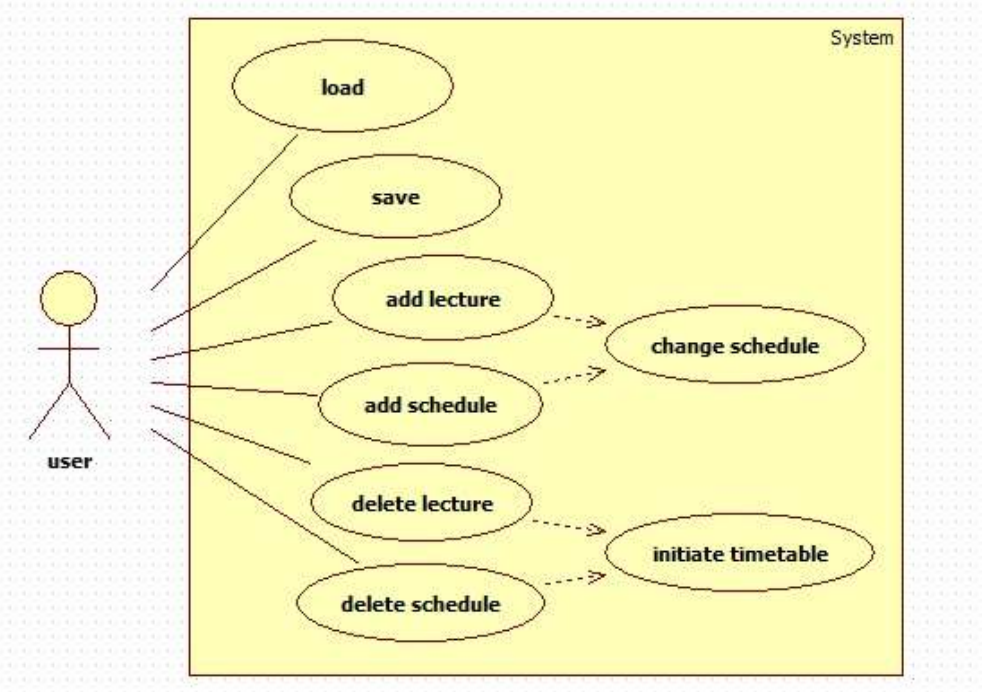
Use case	add shedule
actor	user
type	default
Description	사용자는 원하는 시간에 해당하는 칸에 커서를 두고 오른쪽 클릭을 해서 나오는 일정 추가 메뉴를 선택한다. 일정명, 시간, 장소를 입력하고 저장한다.

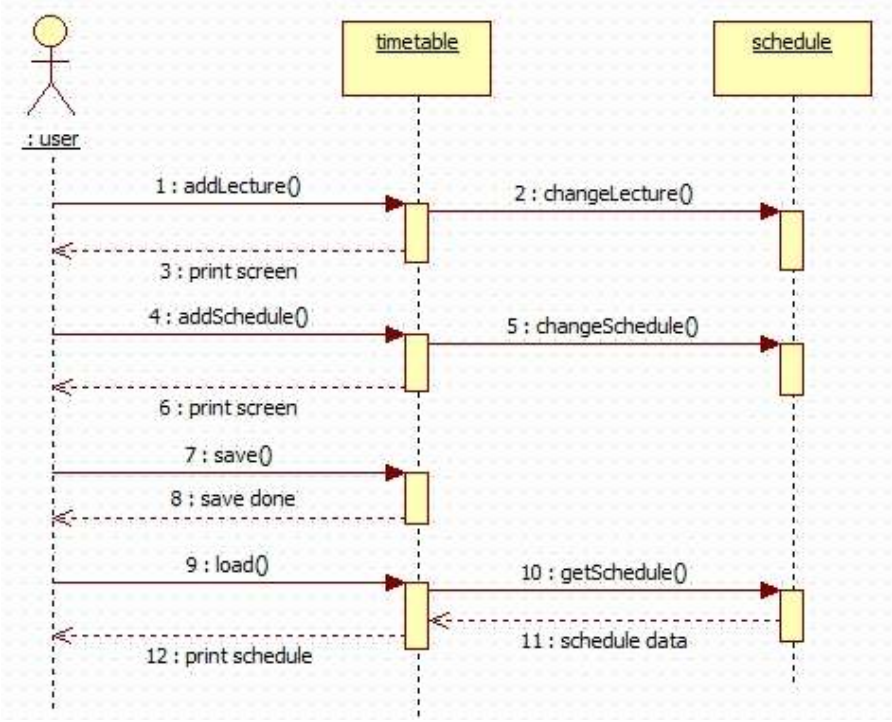
Use case	add lecture
actor	user
type	default
Description	사용자는 원하는 시간에 해당하는 칸에 커서를 두고 오른쪽 클릭을 해서 나오는 강의 추가 메뉴를 선택한다. 강의명, 교수명, 시간, 강의실을 입력하고 저장한다.

Use case	change schedule
actor	user
type	default
Description	사용자가 변경으 원하는 일정/강의를 선택하고 오른쪽 클릭을 해서 나오는 변경 버튼을 누른다. 변경된 내용을 수정하고 저장한다.

Use case	delete schedule
actor	user
type	default
Description	사용자가 삭제를 원하는 일정/강의를 선택하고 오른쪽 클릭을 해서 나오는 삭제 버튼을 누른다.

use case diagram을 통하여 system의 경계 및 user가 system과 어떤 관계를 맺는지 알아봅니다.





use case를 작성하고 use case diagram 그리고 sequential diagram까지 완성함으로써 프로젝트의 Analysis 단계까지 진행되었습니다. 지금까지의 활동으로 프로그램의 목적을 설정하였고, 범위를 정하여 프로젝트가 성공적으로 진행될 수 있도록 하였습니다. 또한 use case를 작성함으로써 프로그램이 수행하는 기능을 정의 및 확인할 수 있었고, 요구사항과 비교하여 우리가 요구사항들을 만족해가며 제대로 구현을 하고 있는지 알 수 있었습니다.

이제 다음 design 단계에서 use case 및 use case diagram에서 class 모델들을 추출하고 이들 관계를 나타내는 class diagram을 만들게 됩니다.

Reference

Head first OOAD - 브랫 맥리프란 외 (O'REILY)
UML 객체지향 분석 · 설계 - 조완수 (홍릉과학출판사)
Software Engineering (8th) - Ian Sommerville (Addison-Wesley)