Team Project

OOAD

Target system : Time Table Program

#1 Plan & Analysis

Class B
Team 1

◇Plan

OOAD에 대한 개념과 방법을 이해하고, 실제적으로 target system을 정해 plan부터 implementation 까지 팀을 구성해 완성해 보도록 한다.

먼저, Target System으로 시간표 프로그램이 주어졌고, 그에 맞게 프로젝트를 시작하기 위해 어떠한 프로그램을 만들 것 인가, 어떤 방법과 어떤 도구를 사용할 것인지 계획 해보도록 한다.

단순하게 시간표를 정리해주는 프로그램이 아니라, 기존 학교 수강신청 포탈을 모태로 삼아, 이용하는 학생들의 요구사항과 추가적으로 더 필요한 기능을 수행하는 프로그램으로 만들기 위해, 앞서 요구사항들 먼저 정리를 해 보았다.

요구사항은 크게 2가지로 나뉘는데 살펴보면,
첫째로는 Functional Requirement 이다. 실제 사용하는 학생들이 요구하는 기능들을 정리하고, 실제적으로 구현이 가능한 것인지 판단하며, 정해진 시간 내에 할 수 있는 지에 대해 확인한다.

- 현재 수강 신청한 과목들이 타과목과 중복되는지 확인(실시간으로 쉽게 새로운 창으로 나타날 수 있도록)
- 현재 수강을 원하는 과목의 인원을 확인(클릭 하지 않고도 바로 확인이 가능하며 신청 가능할 수 있도록)

수강신청이 자신이 원하는 대로 잘 되고, 그에 맞게 정리 된 시간표가 나오는 것이 시간표 프로그램의 궁극적 목표이다. 현재 만들어져서 사용 중인 포탈 수강신청 시스템에서 위와 같은 불편한 점에 대해 요구사항이 발생 하였고, 이를 개선하고, 수정 하는 방법으로 더 유용하게 사용할 수 있는 프로그램으로 개발 하고자 한다.

두 번째로는 Non-Functional Requirement 이다.

실제 프로그램의 기능적 문제나 요구사항이 아니라, 제약적이고, 환경적인 여러 요소들을 말하는데, 해야하는 일정들을 살펴보면,

- 2010년 11월 9일까지 Plan & Analysis
- 2010년 11월 23일까지 Design
- 2010년 12월 8일까지 Implementation

일정의 제한 사항은 다음과 같으며, 이 프로젝트 이외에도 다른 수업의 과제나 기타 일정도 어느정도 수렴하기 위해 팀원이 모여 구체적인 계획을 세우게 되었다.

-그래픽을 기반 으로한 실제적 프로그램 구현
GUI를 이용하기 위해 JAVA라는 고급 언어를 사용하기로 결정 하였다.

-UML Tool 을 통해 각종 Diagram 사용
팀 레포트로 조사하였고, 유용하게 사용할 수 있는 UML Tool 인 StalUML 도구를 사용하며, 현재 진행중인 Plan & Analysis 부분에서는 주로 Use Case Diagram을 사용하도록 하였다.

-기타 팀원이 모두 모여 팀 프로젝트에 착수할 수 있는 여유 있는 시간일정
학생 신분이므로 타 과목 수업과 과제들, 그리고 개인적인 사정으로 인해, 전반적인 OOAD의 구성을 조사한 팀 레포트 제출 일 이후부터 본 프로젝트에 몰두하기 위해 팀원끼리 주말 위주로 일정을 맞추었다.

2010년 10월 30일 Team. 1 첫 미팅을 가짐(기본 틀 형성, 어떤 프로그램을 만들 것 인가에 대한 토의)
2010년 10월 31일 추가적 기능을 수행하는 간단하면서도 유용한 프로그램 개발로 결정 및 조사
2010년 11월 3일 요구사항들을 조사하며, 분석 , 고급언어 결정
2010년 11월 6일 더욱더 정확한 Analysis 를 위한 조사 및 계획수립
2010년 11월 7일 요구사항 분석을 통한 UML Tool 동작 및 UseCase Diagram 설계
2010년 11월 9일 Plan & Analysis 에 대한 발표 준비
2010년 11월 10일 Plan & Analysis 에 대한 발표

발표 후 평일 만남과 주말 미팅을 통해 구체적 프로그램 구현으로 가기 위한 Model, Process 등을 설계할 예정

OOAD를 접목시켜 실제로 Plan 단계에 대해서 각종 요구사항들을 수렴하면서, 크진 않지만 하나의 프로젝트를 부분 부분 나눠 정확하고 실용적인 프로그램으로 만들기 위해 노력한다.

◇Analysis

앞서 계획한 것들을 기반으로 실제적으로 더 구체적으로 분석하고
앞으로 나아가야 할 방향과 목적에 대해 다시금 조사한다.

어떤 프로그램을 구현할 것인지 정했고, 여러 요구사항들에 대해 더 심도 있게 분석하기 위해 IEEE의 원칙에 따라 질문에 답하는 형식으로 기술하였다.

1. Introduction

-1.1 Purpose (1.1 of the SRS)

a) Delineate the purpose of the SRS:
기존 시간표의 기능적 한계인 '실시간 비교'와 '시간표 확인'을 극복한 시간표의 구현

b) Specify the intended audience for the SRS:
현재 사용자가 어떤 과목을 신청했는지, 또 그 과목은 현재 사용자의 시간표상에서 어느 시간대에 위치했는지를 한 눈에 알아볼 수 있다. 그리고 수강신청 하려는 과목의 현재 수강신청 인원을 실시간으로 보여준다.

-1.2 Scope (1.2 of the SRS)

a) Identify the software product(s) to be produced by name:
'More Useful Time Table'

b) Explain what the software product(s) will, and, if necessary, will not do:
1. 창이 두개가 나와서, 우리의 수강신청 입력 화면과 실시간으로 그 결과를 보여준다.
2. 수강신청 화면에서 바로 현재 수강신청 인원을 볼 수 있게 해준다.

c) Describe the application of the software being specified, including relevant benefits, objectives, and goals:
우리의 시간표는 실시간으로 결과를 한 화면에 보여주기 때문에, 현재 사용자에게 남아있는 시간을 확인할 수 있어 보다 편리하게 수강신청이 가능하다. 또한, 수강신청 인원을 확인하기 위해 일일이 검색을 해야 했던 기존 수강신청 프로그램의 불편함을 덜어준다.

-1.3 Definitions, acronyms, and abbreviations (1.3 of the SRS)

피드백이 가능한 시간표

-1.4 References (1.4 of the SRS)

학교 수강신청 프로그램

-1.5 Overview (1.5 of the SRS)

a) Explain how the SRS is organized:
기존 시간표에 사용자의 입장에서 기능을 추가한다. 이것이 사용자의 요구사항이 되고, 그 요구 사항을 수용하여 프로그램을 구현한다.

2. Overall description (Section 2 of the SRS)

-2.1 Product perspective (2.1 of the SRS)

a) System interfaces:
JAVA

b) User interfaces:
Graphic User Interface (GUI)

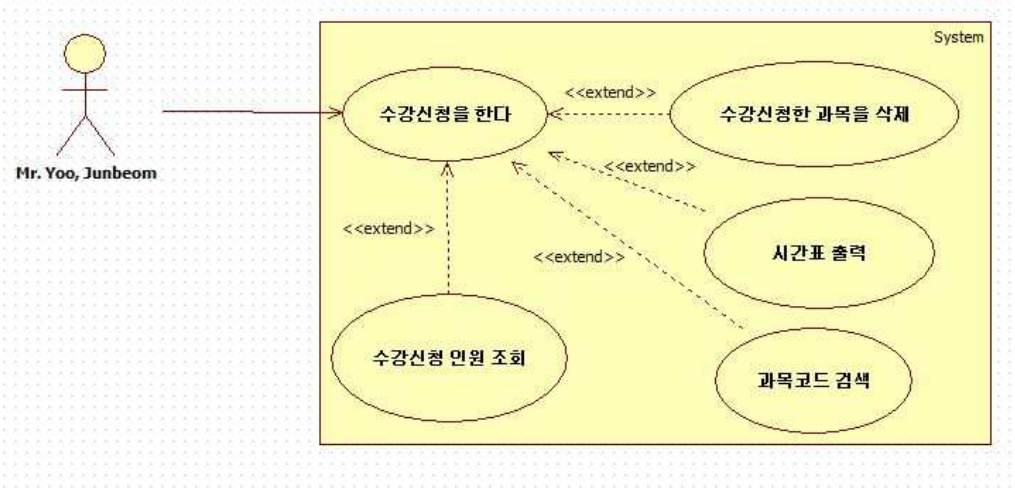
c) Software interfaces:
starUML, eclipse

d) Operations:
Backup and Recovery Operation: 사용자가 완료한 시간표를 따로 Backup할 수 있다, 그리고 수정이 가능하다.

e) Site adaptation requirements:
http://kupis.konkuk.ac.kr/sugang

-2.2 Product functions

a)
-어떤 과목을 수강신청 했을 때, 수강신청 화면이 있고, 그 옆에 현재 사용자의 시간표가 동시에 화면에 출력된다. 그래서 현재 사용자가 사용할 수 있는 시간표내의 잔여 시간을 실시간으로 보여준다.
-수강신청 과목코드 옆에 현재 수강신청 인원 현황을 표시하여, 사용자가 쉽게 확일할 수 있게 해준다.
b)



-2.3 User characteristics (2.3 of the SRS)

온라인으로 수강신청하는 건국대학교 학생

-2.4 Constraints (2.4 of the SRS)

a) Interfaces to other applications:
Internet Explore, Firefox, Chrome

b) Control functions:
한 사용자가 수강하는 한도 학점, 수강 가능한 교과목

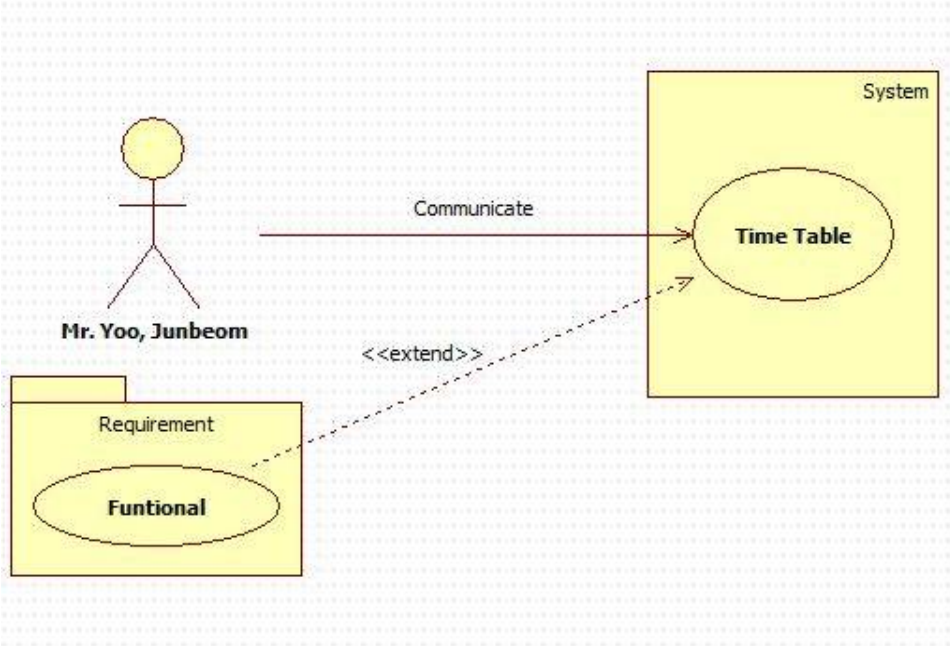
c) Higher-order language requirements:
JAVA

d) Reliability requirements:
사용자가 입력한 수강과목 코드가 정확히 시스템에 입력되어 신뢰할 수 있어야한다.

-2.5 Assumptions and dependencies (2.5 of the SRS)

학교 수강신청 서버의 변화

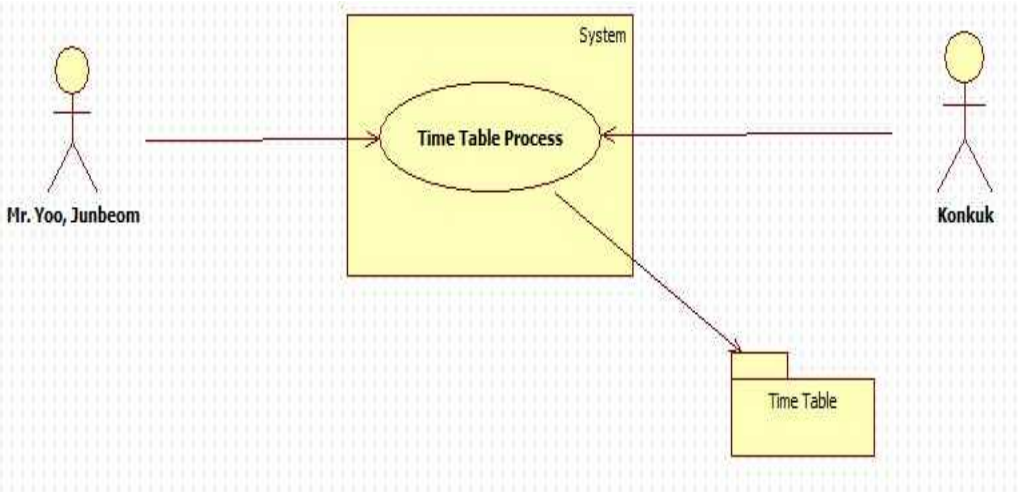
3. Specific requirements (Section 3 of the SRS)



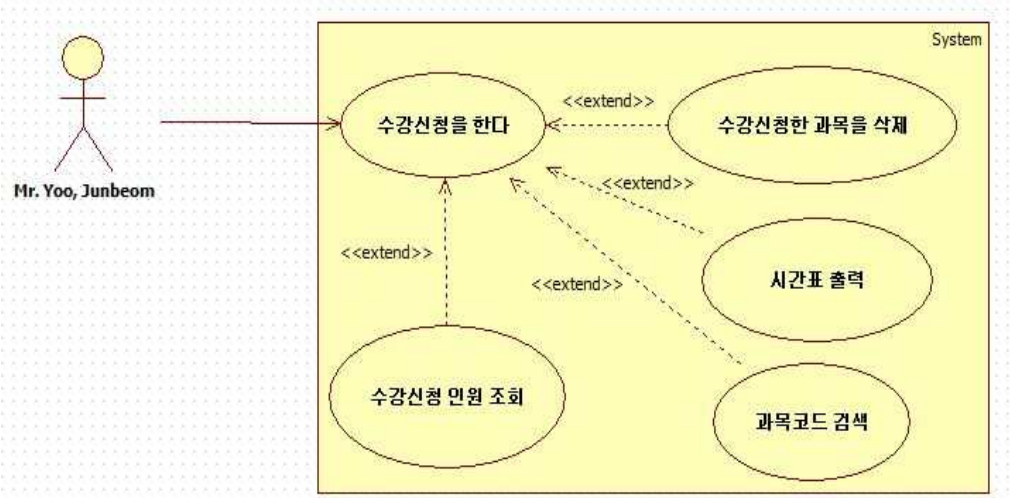
현재 수강신청 프로그램의 경우 발생하는 불편한 점

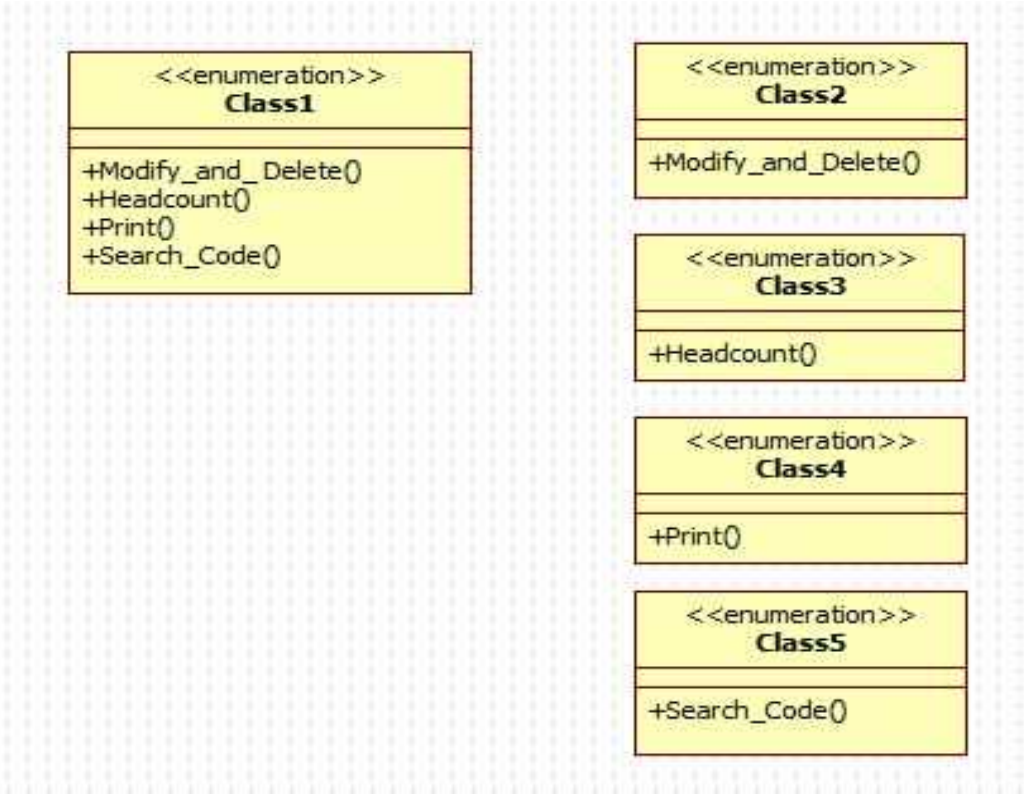
- 현재 수강신청 한 과목들이 타과목과 중복되는지 확인
- 현재 수강을 원하는 과목의 인원을 확인

이 두 가지가 현재 시간표에서 불편한 것이라 판단해, 이를 개선하는 소프트웨어를 개발한다.



사용자가 시스템상의 Time Table Process에 접근하여, Konkuk이라는 서버의 데이터를 불러와서 Time Table을 작성한다.





수강 신청이라는 기능 안에

-수정과 삭제
-수강 인원 조회
-시간표 출력
-과목코드 검색

이 같은 기능이 있어 사용자가 더욱 편리하게 수강신청이 가능하다.