TEAM PROJECT
OOAD

TARGET SYSTEM : TIME TABLE PROGRAM
_비교와 COMMENT 기능을 포함한 시간표
#2 DESIGN

Class B Team 1

200711437 성하진

Contents

- Introduction , Definitions
- Design description information content
- Design description organization

Introduction , Definitions

Object Oriented Analysis & Design(OOA/D) 단계 중 Design Phase부분

Design Entity의 기능과 Entity간 관계, 전체적인 Software Design Description(SDD)를 IEEE1016-1998 에 준하여 문서 작성 및 시스템 설계

Design description information content

-5.1 Introduction

-5.2 Design Entities

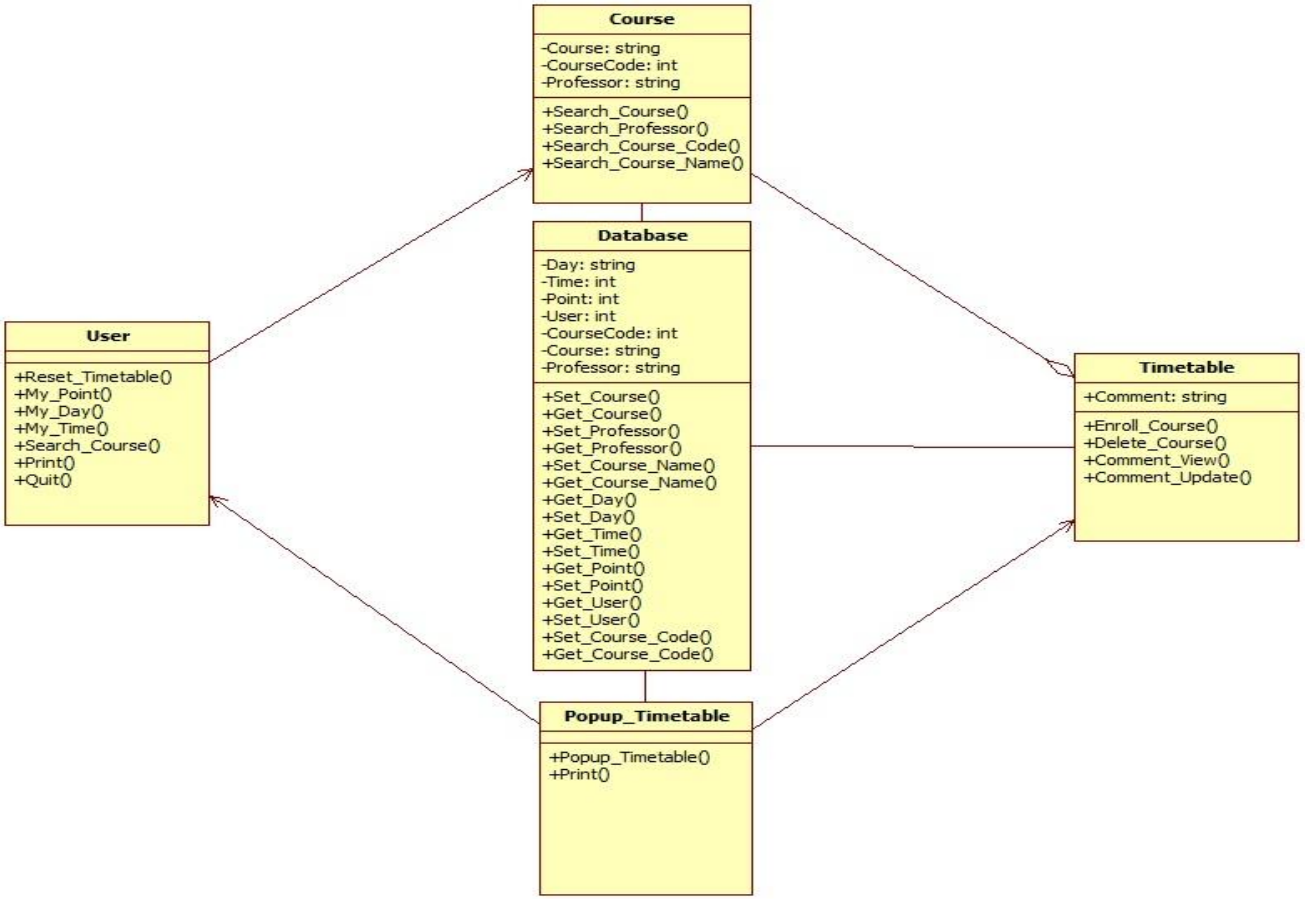
-5.3 Design Entities Attribute

-5.1 Introduction

SDD는 생성할 소프트웨어 시스템의 모형 또는 표현이다. 이 모형은 소프트웨어 시스템의 계획, 분석, 구현에 필요한 정확한 설계 정보를 제공해야 한다. 이는 시스템이 분할된 설계 Entity들로 표현되고 이들의 중요한 속성과 이들 Entity 간의 관계를 기술해야 한다.

하나의 소프트웨어 시스템을 표현하는데 사용되는 설계기술서 모형은 설계 Entity들의 집합, 각 소유 속성들과 관계들로 표현될 수 있다. 이 모형을 단순화하기 위하여 각 설계 Entity의 속성과 관계들은 속성들의 표준 집합으로 기술되어진다. 프로젝트에 필요한 설계 정보들은 Entity의 식별과 그들 간의 관련 속성들로 구성된다. SDD는 모든 설계 Entity에 대한 속성들이 명시되었을 때 완성된다.

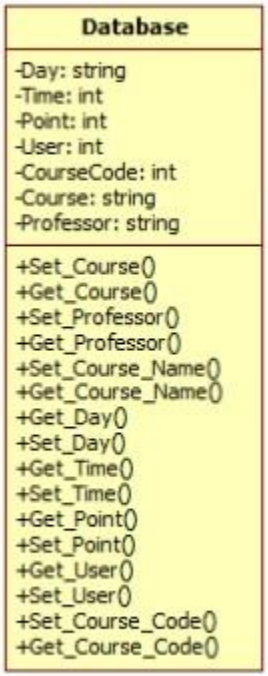
-5.2 Design Entities



-5.3 Design Entities Attribute

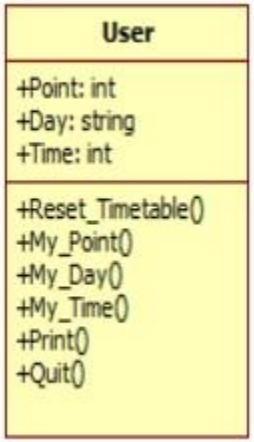
-5.3.1 Identification
-5.3.2 Type
-5.3.3 Purpose
-5.3.4 Function
-5.3.5 Subordinates
-5.3.6 Dependencies
-5.3.7 Interface
-5.3.8 Resources
-5.3.9 Processing
-5.3.10 Data

-5.3.1~5.3.4
(Identification , Type , Purpose , Function)



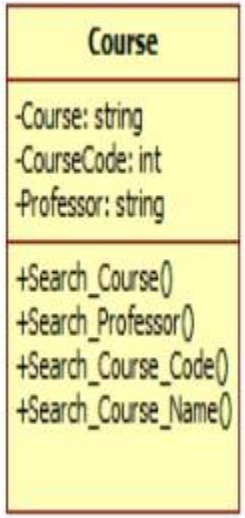
Identification	Database
Type	데이터 저장: 수강 가능한 과목명 , 과목별 요일 및 시간 , 이수학점 , 현재 등록된 학생 수, 과목 코드 , 교수명 등을 저장함. 절차: 과목명을 중심으로 , 과목을 구성하는 각종 정보들을 연결 시켜, 하나의 집합체로서 자료들을 저장하며, 필요시 각 부분들을 호출하여 따로 사용 가능하게 함.
Purpose	수강신청의 기본 자료들을 저장하는 클래스로서 , 수강 가능한 과목들의 정보(학점, 시간, 요일, 교수명 등)를 저장하고 그 내용을 표현 할 수 있게 하기 위한 클래스이다.
Function	타 클래스가 각 메소드를 호출하는데 있어, 여러 과목들에 대한 정보들을 공유할 수 있게 하며, 과목별 정보를 저장하는 기능을 한다.

-5.3.1~5.3.4
(Identification , Type , Purpose , Function)



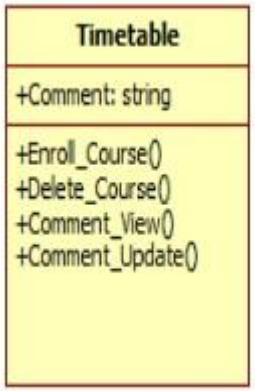
Identification	User
Type	데이터 저장: 현재 학생의 한도 학점 , 수강한 과목의 날짜 , 시간을 저장하는 변수를 중심으로 현재 자신의 시간표를 저장함. 절차: 사용하는 사용자를 기준으로 자신의 시간표 구성한도에 맞추어 해당시간표를 초기화 하거나 출력 , 종료를 수행하며, 현재 가능한 과목 및 학점한도를 표현할 수 있도록 함.
Purpose	실제로 수강신청을 수행하는 사용자 클래스 이며 , 해당 시간표 초기화 , 현재 자신의 시간표 , 출력 및 프로그램 종료 등 사용자 입장에서 필요한 기능을 수행할 수 있도록 하는 클래스 이다.
Function	현재 시간표 초기화, 한도학점, 시간 , 날짜 표기 및 해당 변수들과 함께 시간표를 구성할 수 있도록 하는 기능을 가지며, 출력및 종료가 가능하다.

-5.3.1~5.3.4
(Identification , Type , Purpose , Function)



Identification	Course
Type	데이터 저장: 과목명 , 해당 과목 코드 , 교수명의 변수들로 과목을 검색하는데 중점을 두고, 검색하는 방법들에 대한 메소드들을 저장함. 절차: 사용자가 과목을 검색할 시 교수명 검색 , 과목코드검색 , 과목명 검색 부분을 각각 호출했을 시 과목에 대한 기본 정보가 한번에 호출될 수 있도록 구성함.
Purpose	수강신청을 하는 과목에 대한 검색하는 방법들과 각 방법에 따라 표현하는 메소드들을 표현하기 위한 클래스 이며 , 이 클래스를 통해 사용자가 과목검색을 수월하게 할 수 있도록 한다.
Function	과목 검색시 과목명뿐 아니라 과목코드나 교수명으로도 검색이 가능하도록 하며 , 어떤 방법으로 검색을 하더라도 , 각 과목에 대한 상세 정보가 표현될 수 있도록 하는 기능을 갖는다.

-5.3.1~5.3.4
(Identification , Type , Purpose , Function)

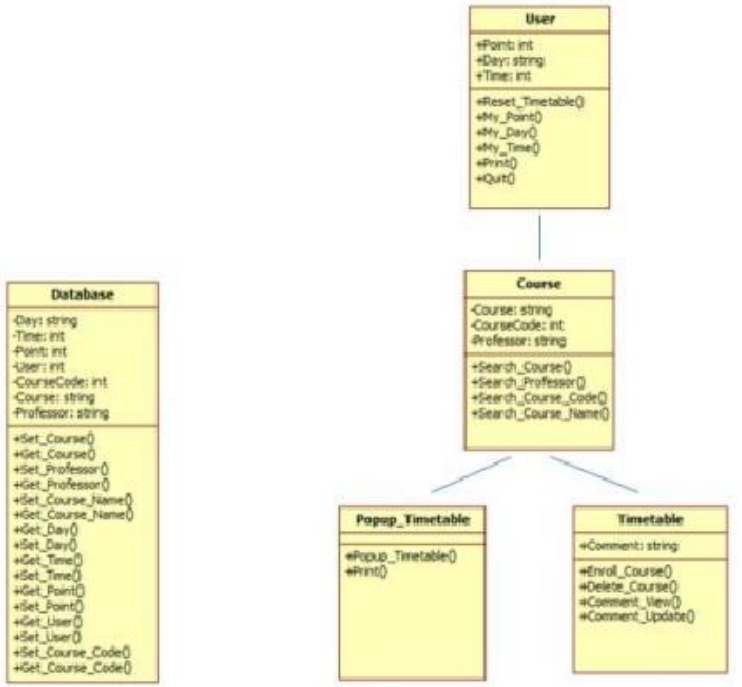


Identification	Timetable
Type	데이터 저장: 시스템의 주 기능인 comment 작성에 필요한 변수와 과목을 등록, 삭제 하는 부분 , comment를 보거나 새로 등록을 하는 메소드들을 저장함. 절차: comment 등록시 변수로 입력을 받아 기존 comment에 추가하며, 수강과목에 대해 등록하거나 삭제를 하는 메소드를 호출함으로 실제 시간표상에 올려줄 수 있도록 한다.
Purpose	사용자가 실제 시간표상에 과목들을 등록하며 삭제하는 기능을 하며 , comment를 보거나 새로 등록을 하기 위해 따로 시간표 자체를 저장하는 클래스 이다.
Function	comment 등록 및 보기 , 실제 과목들에 대한 등록 및 삭제 기능을 수행하며 그 내용들을 개인 시간표로 저장을 하는 기능을 한다.

-5.3.1~5.3.4
(Identification , Type , Purpose , Function)

Identification	Popup_Timetable
Type	데이터 저장: 과목을 등록하거나 삭제하는데 실제 수행 전 새로운 창으로 보여줄 수 있는 메소드, 현재 시간표를 출력하는 메소드를 갖는다. 절차: 사용자가 시간표를 등록하거나 삭제를 수행할 시 , 새로운 창으로 누적된 시간표 상황과 비교를 할 수 있도록 하며, 바로 출력이 가능할 수 있다.
Purpose	본 시스템의 두 번째 기능으로서 , 비교를 쉽게 할 수 있기 위해 새로운 창으로 과목을 등록하거나 삭제하기전의 시간표상의 변화를 표현하며, 출력도 가능하게 하는 클래스이다.
Function	실시간으로 사용자가 등록하거나 삭제를 하는 부분에 있어 수행 이전에 쉽게 비교할 수 있는 기능을 갖는 새로운 창을 올려주는 기능과 현재 상태의 시간표를 출력할수 있는 기능을 갖는다.

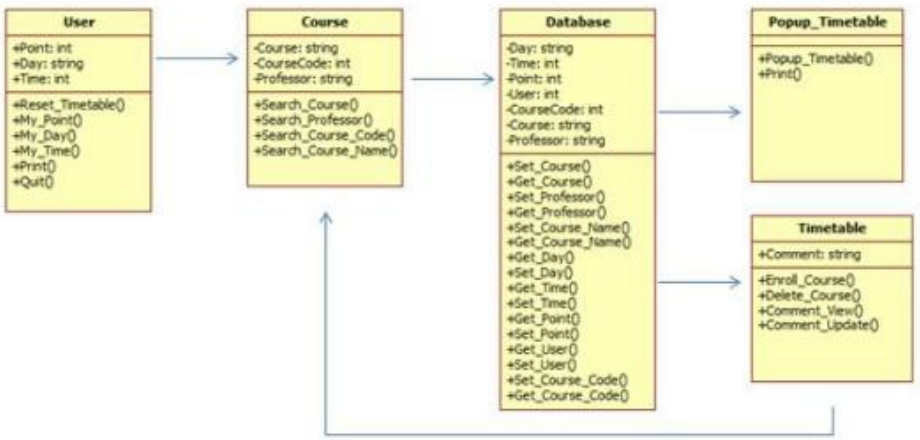
-5.3.5 Subordinates



Parent 클래스는 User 클래스이고, Child 클래스는 Course 클래스이다. 그리고 Popup_Timetable 클래스와 Timetable 클래스는 Grand Child 클래스이다. User 클래스의 하위 구성 클래스로서 존재하는 Course 클래스에서의 기능을 또다시 Pop up_Timetable 클래스와 Timetable 클래스가 구성하고 있기 때문이다.

Database 클래스는 다른 클래스들과는 독립적으로 존재한다. 다른 클래스에 종속되지 않고 독립적으로 Course 클래스, Timetable 클래스에 Data를 제공한다.

-5.3.6 Dependencies



User가 Course 클래스에 접근하여

Professor, Course_Code, Course_Name중

선택하여 과목을 검색하도록 한다.

Search_Course는 Database 클래스에 접근하여,

과목 선택시 Popup_Timetable 클래스와 Timetable

클래스에 접근하여 시간표를 (화면에)출력하거나

Comment_Update를 가능하게 한다.

Timetable 클래스에서는 과목을 등록하거나 삭제한 후에

다시 과목을 검색할 수 있도록 Course 클래스로 돌아온다.

사용자가 원하는 결과를 얻을 때 까지 이러한 순환이 반복된다.

-5.3.7 Interface
-5.3.8 Resources

<물리적 장치>

프린터 - 해당 프린터의 Driver가 Install된 PC를 통해 개인 시간표 출력 가능

메모리 - 본 시스템을 설치 및 실행 가능하고, 개인 시간표를 저장할 수 있는 용량(약10M)

<소프트웨어 서비스>

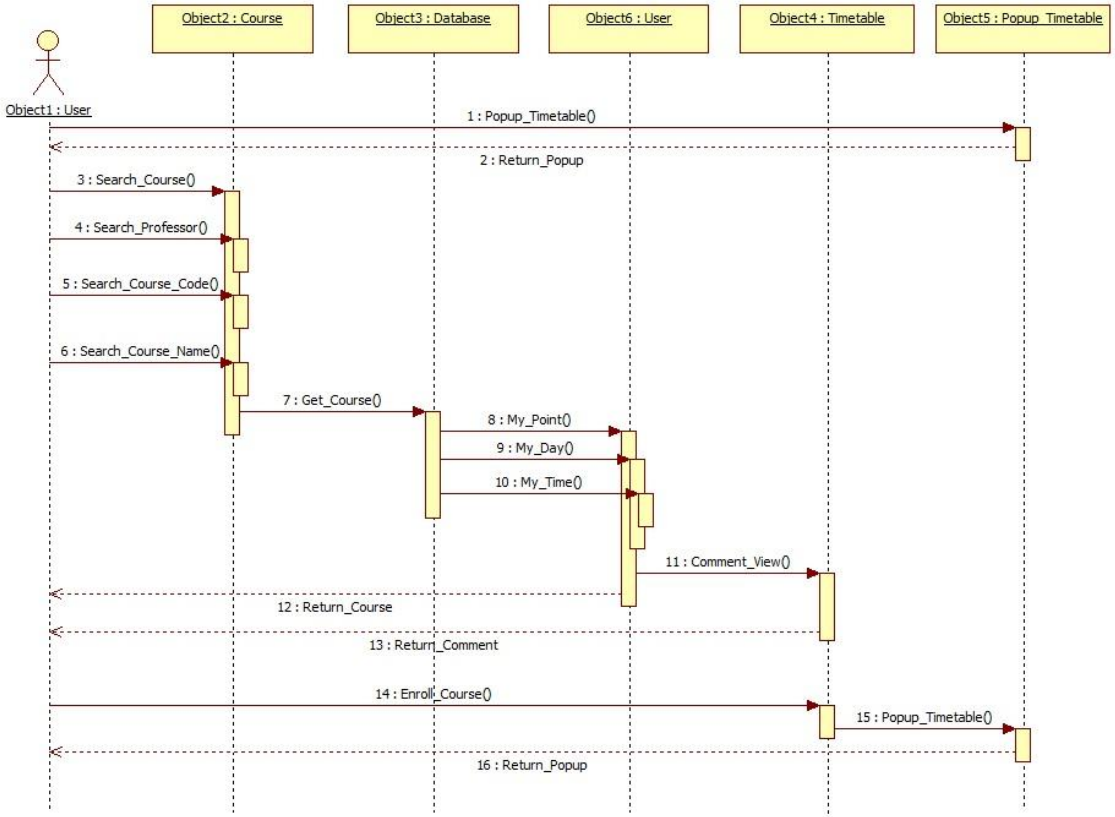
운영체제 서비스 - Windows 기반의 OS 및 JAVA Code 실행이 가능한 프로그램 설치

<프로세스 서비스>

메모리 할당 - 프로그램의 원활한 실행을 위하여 최소 64MB의 메모리 할당이 필요

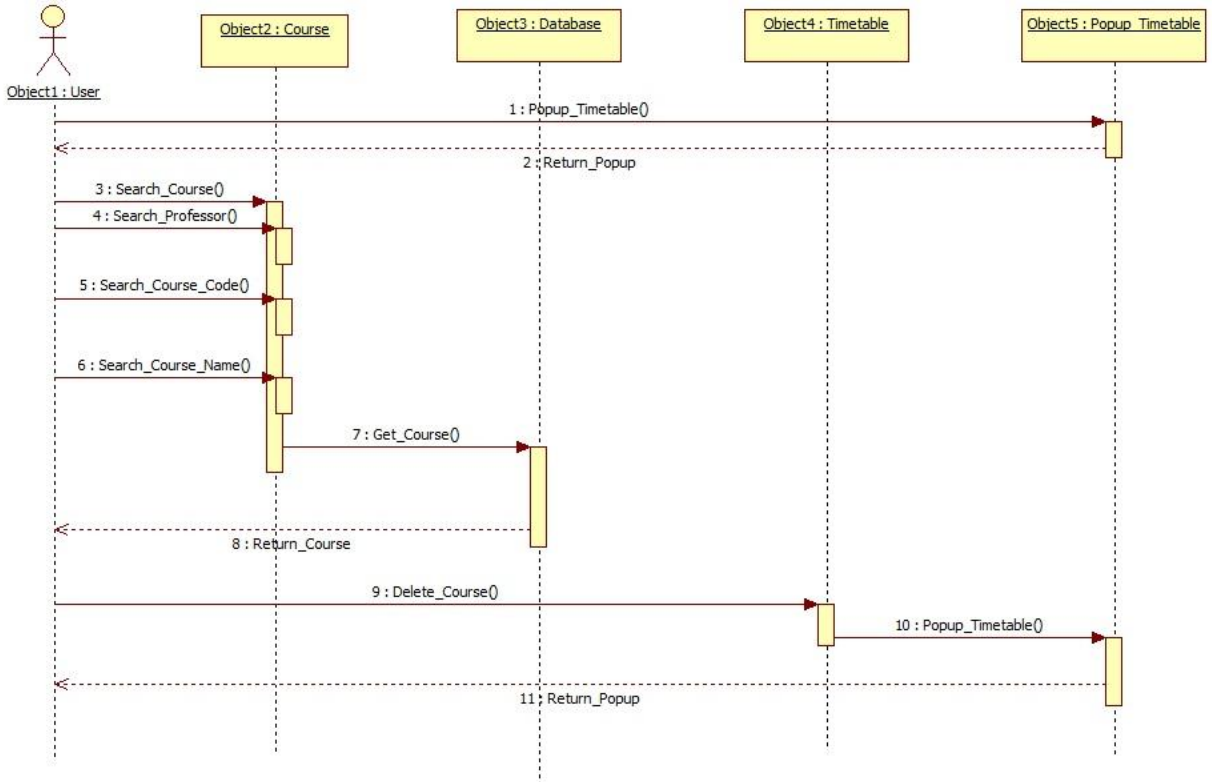
-5.3.9 Processing

<과목 등록 기능>



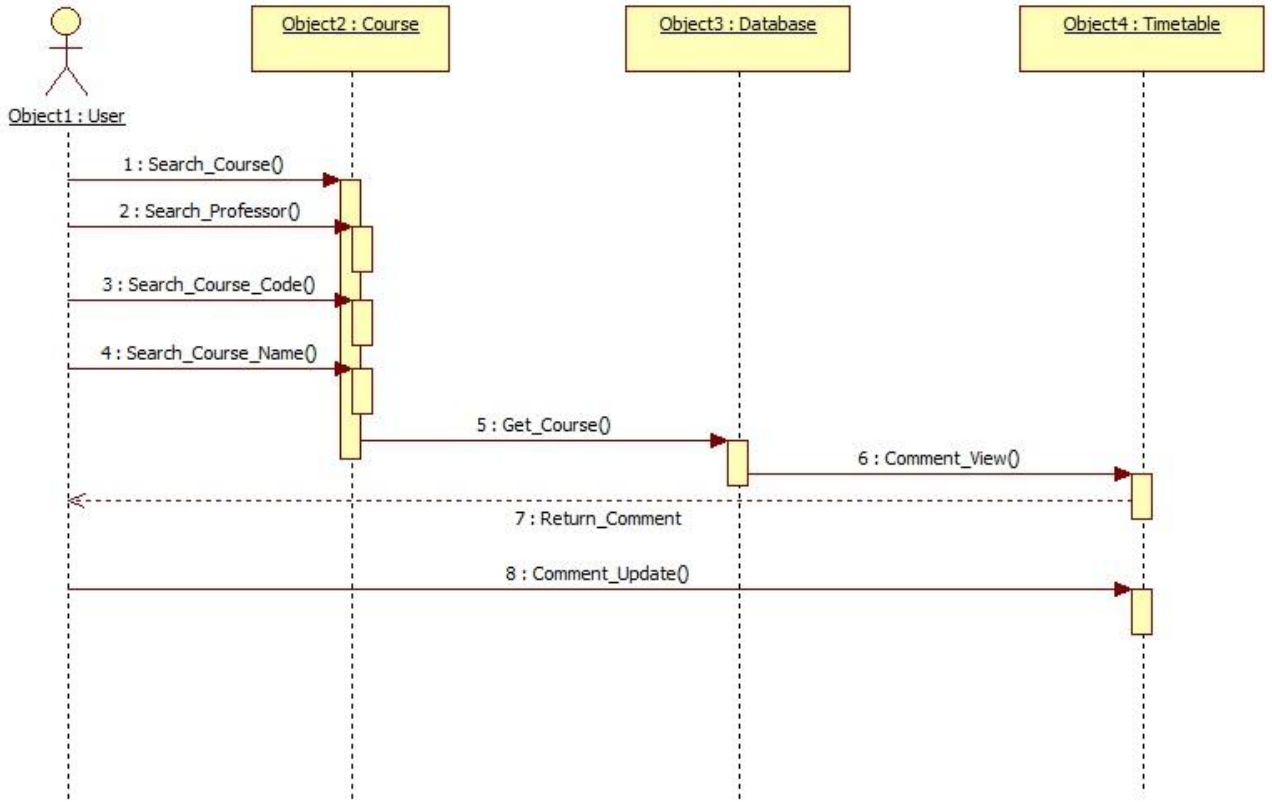
-5.3.9 Processing

<과목 삭제 기능>



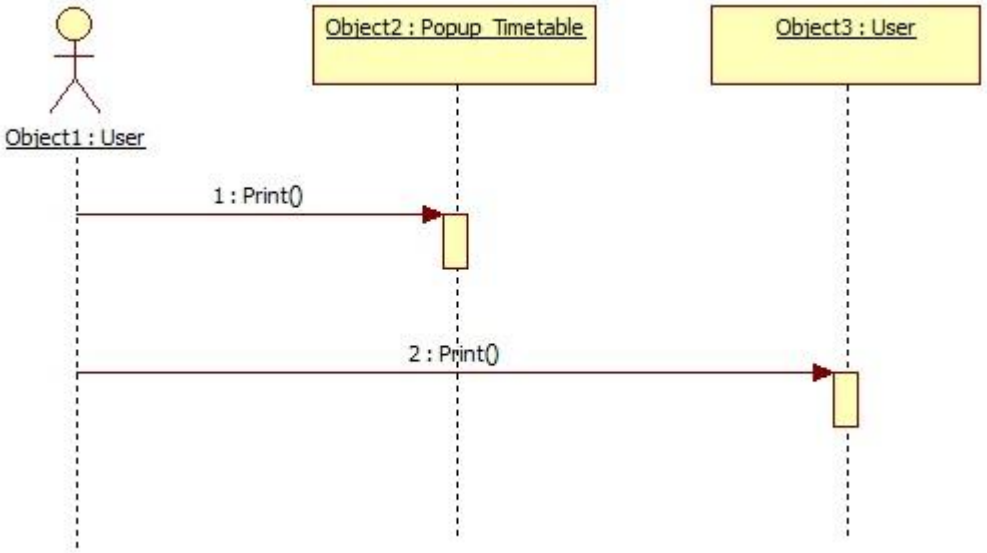
-5.3.9 Processing

<Comment 등록 및 보기 기능 >

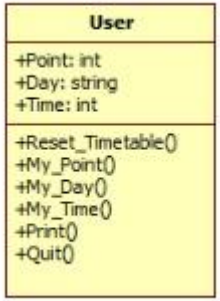


-5.3.9 Processing

<프린트 기능>

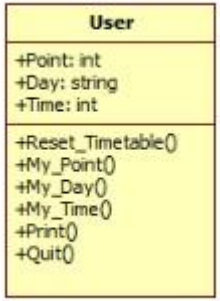


-5.3.10 Data



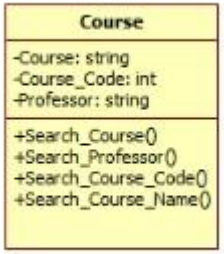
변수	
Point: int	Point는 현재 수강 가능한 User의 총 학점을 뜻한다. 이 때, Point(학점)을 표현할 변수는 1~3같은 식의 숫자로 표현되므로 int형 변수를 사용한다.
Day: string	Day는 User가 수강하려고자 하는 과목의 수업 요일을 뜻한다. 이 때, Day(요일)을 표현할 변수는 문자가 되어야 하므로 string형 변수를 사용한다.
Time: int	Time은 User가 수강하려고자 하는 과목의 수업 시간을 뜻한다. 이 때, Time(시간)을 표현할 변수는 11시~1시와 같은 방법으로 표현되므로 int형 변수를 사용한다.

-5.3.10 Data



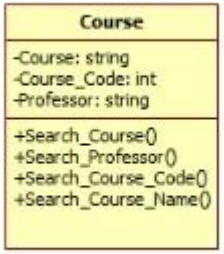
메소드	
Reset_Timetable()	현재 수강신청을 완료 혹은 진행 중인 User의 시간표에 문제가 있거나, 혹은 User의 계획이 변경되었을 시에 User는 새로운 시간표를 재구성하길 원한다. 이 때, 각 과목별로 하나하나 과목 삭제를 하여 시간표를 초기화하는 대신 Reset_Timetable이란 메소드로 한 번에 시간표를 초기화할 수 있다.
My_Point()	과목 등록을 하기 전, User는 자신의 수강 가능한 학점을 확인할 수 있어야 한다. My_Point메소드는 한도학점을 초과해서 발생할 수 있는 오류나, 학점 미달로 발생할 문제를 막을 수 있다.
My_Day()	My_Day는 현재 수강하려는 과목이, 이 전에 수강하려는 과목과 요일이 중복되는 경우를 피할 수 있게 해준다.
My_Time()	My_Time은 현재 수강하려는 과목이 이 전에 수강하려는 과목과 요일은 다르지만, 시간이 중복되는 경우를 피할 수 있게 해준다.
Print()	현재 User가 신청한 과목들로부터 만들어진 시간표를 Print메소드를 통해 PC와 연결 된 프린터로 출력할 수 있다.
Quit()	Quit은 프로그램 종료 메소드이다.

-5.3.10 Data



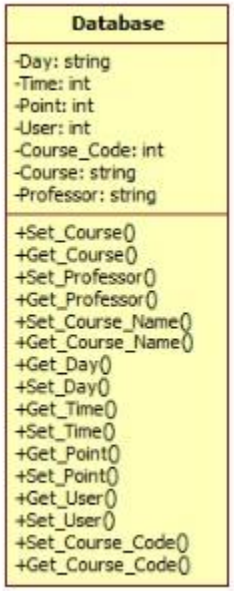
변수	
Course: string	Course는 과목을 의미한다. 과목명의 경우는 문자로 되어 있으므로 string변수를 사용한다.
Course_Code: int	Course_Code는 과목의 고유 과목 번호를 의미한다. 과목 번호의 경우 4자리 숫자로 이루어져 있으므로 string변수를 사용한다.
Professor: string	Professor는 과목의 교수님를 의미한다. 교수님의 성함은 문자로 입력되어 있으므로 string변수를 사용한다.

-5.3.10 Data



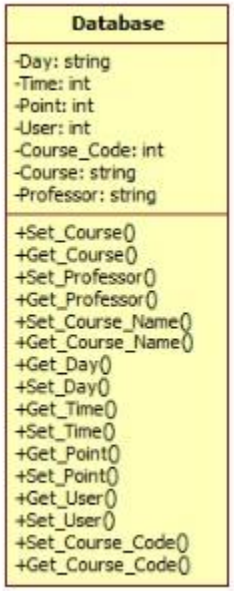
메소드	
Search_Course()	Search_Course()는 User가 과목검색을 할 때 이용할 각각의 방법들을 아우르는 메소드이다.
Search_Professor()	Search_Professor()는 User가 수강하려고 하는 과목을 검색할 때, 교수명으로 검색할 수 있는 메소드이다.
Search_Course_Code()	Search_Course_Code()는 User가 수강하려고 하는 과목의 고유 과목번호를 알고 있을 때, 그 번호만으로 과목을 검색할 수 있는 메소드이다.
Search_Course_Name()	Search_Course_Name()은 User가 수강하려고 하는 과목의 이름으로 해당 과목을 검색할 수 있는 메소드이다.

-5.3.10 Data



변수	
Day: string	Day는 User가 수강하려고자 하는 과목의 수업 요일을 뜻한다. 이 때, Day(요일)를 표현할 변수는 문자가 되어야 하므로 string형 변수를 사용한다.
Time: int	Time은 User가 수강하려고자 하는 과목의 수업 시간을 뜻한다. 이 때, Time(시간)을 표현할 변수는 11시~1시와 같은 방법으로 표현되므로 int형 변수를 사용한다.
Point: int	Point는 현재 수강가능 한 User의 총 학점을 뜻한다. 이 때, Point(학점)을 표현할 변수는 1~3같은 식의 숫자로 표현되므로 int형 변수를 사용한다.
User: int	User는 과목의 수강 인원수를 말한다. 학생 수는 int형 변수를 사용하여 표현 될 수 있다.
Course_Code: int	Course_Code는 과목의 고유 과목 번호를 의미한다. 과목 번호의 경우 4자리 숫자로 이루어져 있으므로 string변수를 사용한다.
Course: string	Course는 과목을 의미한다. 과목명의 경우는 문자로 되어 있으므로 string변수를 사용한다.
Professor: string	Professor는 과목의 교수님를 의미한다. 교수님의 성함은 문자로 입력되어 있으므로 string변수를 사용한다.

-5.3.10 Data



메소드	
Set_Course() **Get_Course()**	User가 프로그램을 실행하기 전에 Set_Course를 통해 클래스 내 대항하는 과목정보를 저장한다. User가 프로그램을 실행하면 Get_Course를 통해 해당 과목에 대한 정보를 가져온다.
Set_Professor() **Get_Professor()**	User가 프로그램을 실행하기 전에 Set_Professor를 통해 과목의 해당 교수정보를 저장한다. User가 프로그램을 실행하면 Get_Professor를 통해 해당 과목의 해당 교수정보를 가져온다.
Set_Course_Name() **Get_Course_Name()**	User가 프로그램을 실행하기 전에 Set_Course_Name을 통해 과목의 과목명을 저장한다. User가 프로그램을 실행하면 Get_Course_Name을 통해 과목명을 가져온다.
Set_Day() **Get_Day()**	User가 프로그램을 실행하기 전에 Set_Day를 통해 과목의 수강 요일을 저장한다. User가 프로그램을 실행하면 Get_Day를 통해 해당 과목의 수강 요일을 가져온다.
Set_Time() **Get_Time()**	User가 프로그램을 실행하기 전에 Set_Time을 통해 과목의 수강 시간을 저장한다. User가 프로그램을 실행하면 Get_Time을 통해 해당 과목의 수강 시간을 가져온다.
Set_Point() **Get_Point()**	User가 프로그램을 실행하기 전에 Set_Point를 통해 과목의 수강학점을 저장한다. User가 프로그램을 실행하면 Get_Point를 통해 해당 과목의 학점정보를 가져온다.
Set_User() **Get_User()**	User가 프로그램을 실행하기 전에 Set_User를 통해 과목의 수강인원정보를 저장한다. User가 프로그램을 실행하면 Get_User를 통해 과목의 수강인원정보를 가져온다.
Set_Course_Code() **Get_Course_Code()**	User가 프로그램을 실행하기 전에 Set_Course_Code를 통해 과목의 해당 과목번호를 저장한다. User가 프로그램을 실행하면 Get_Course_Code를 통해 과목의 과목번호를 가져온다.

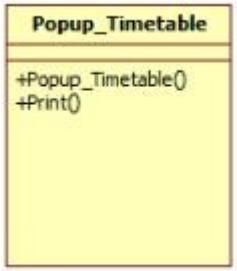
-5.3.10 Data

변수	
Comment: string	User들이 입력한 Comment, 즉 해당 과목에 대한 댓글을 이야기한다. 문자로 작성되므로 string변수를 이용한다.

-5.3.10 Data

메소드	
Enroll_Course()	User가 과목 검색 뒤 해당 과목을 시간표에 등록하려고 한다면 Enroll_Course메소드를 통해 시간표에 해당 과목을 등록할 수 있다.
Delete_Course()	User가 이미 신청한 과목 중 삭제를 원하는 과목이 있다면 Delete_Course메소드를 통해 시간표에서 해당 과목을 삭제할 수 있다.
Comment_View()	User가 신청할 과목을 선택함에 있어서 이전 User들이 해당 과목에 남긴 댓글들을 확인할 수 있는 메소드가 Comment_View이다.
Comment_Update()	User가 특정 과목에 댓글을 남길 수 있게 해주는 것이 Comment_Update메소드 이다.

-5.3.10 Data



메소드	
Popup_Timetable()	Popup_Timetable메소드는 User가 팝업창으로 현재 시간표를 보거나, 어떠한 과목을 등록 혹은 삭제를 했을 때 변화한 상태를 팝업창에 시간표를 나타냄으로써 User에게 제공한다.
Print()	현재 User가 신청한 과목들로부터 만들어진 시간표를 Print메소드를 통해 PC와 연결 된 프린터로 출력할 수 있다.

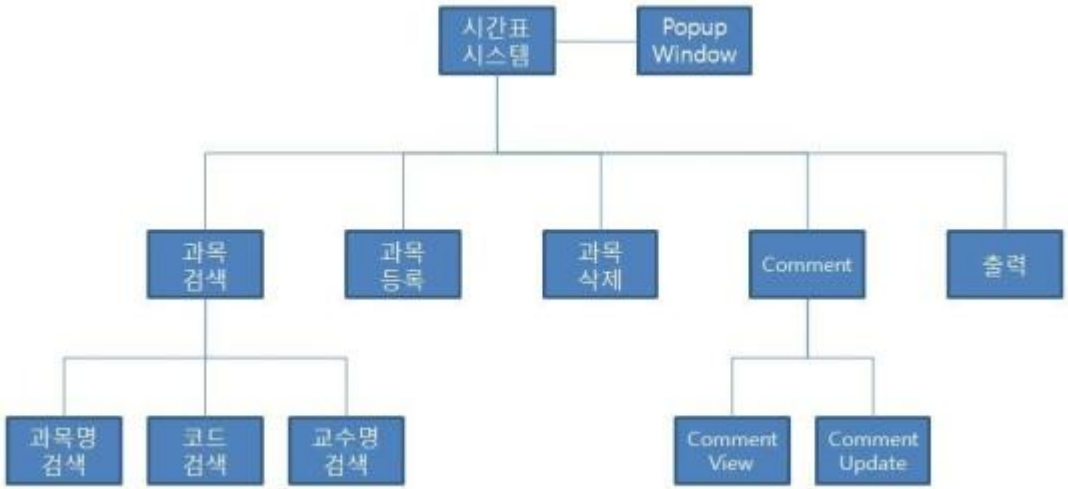
Design description organization

-6.1 introduction

-6.2 View

- -6.2.1 Decomposition Description
- -6.2.2 Dependency Description
- -6.2.3 Interface Description
- -6.2.4 Detailed Design Description

-6.2.1 Decomposition Description



과목검색 : 과목명, 코드, 교수명중 한 가지를 택하여 과목을 검색함.

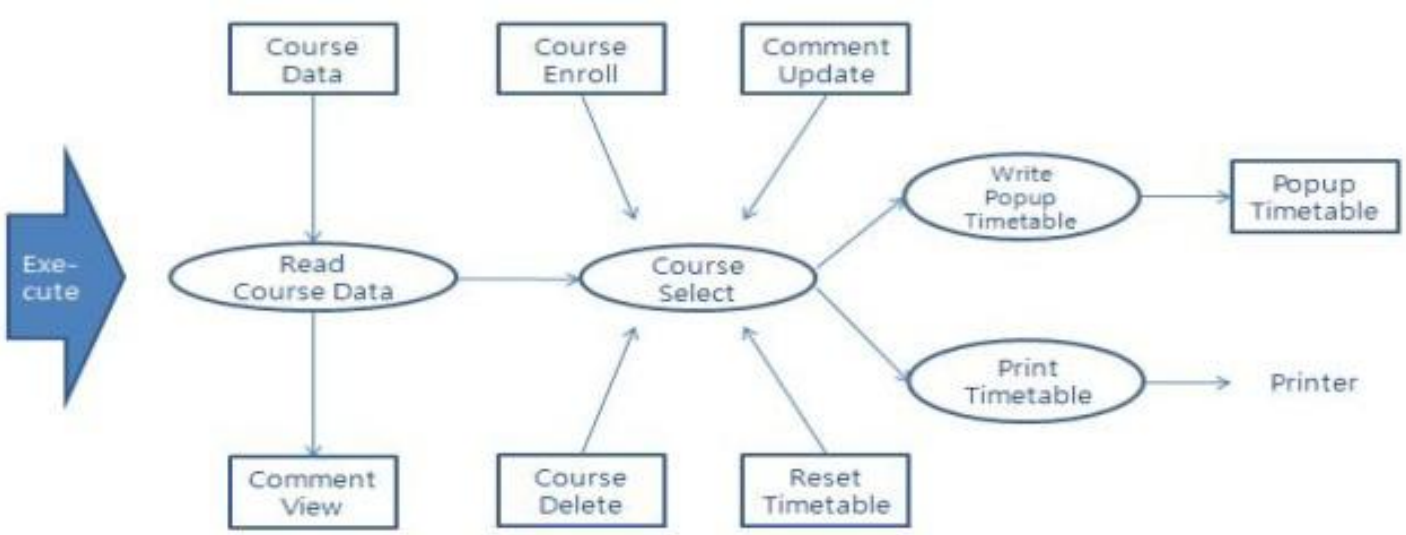
과목등록 : 검색하여 선택한 과목을 데이터베이스에 저장함.

과목삭제 : 선택한 과목을 사용자가 등록한 시간표에서 삭제함.

Comment : Comment View, Comment등록 두 가지 기능을 실행함.

출력 : 선택한 시간표를 Output device(Printer)로 출력함.

-6.2.2 Dependency Description

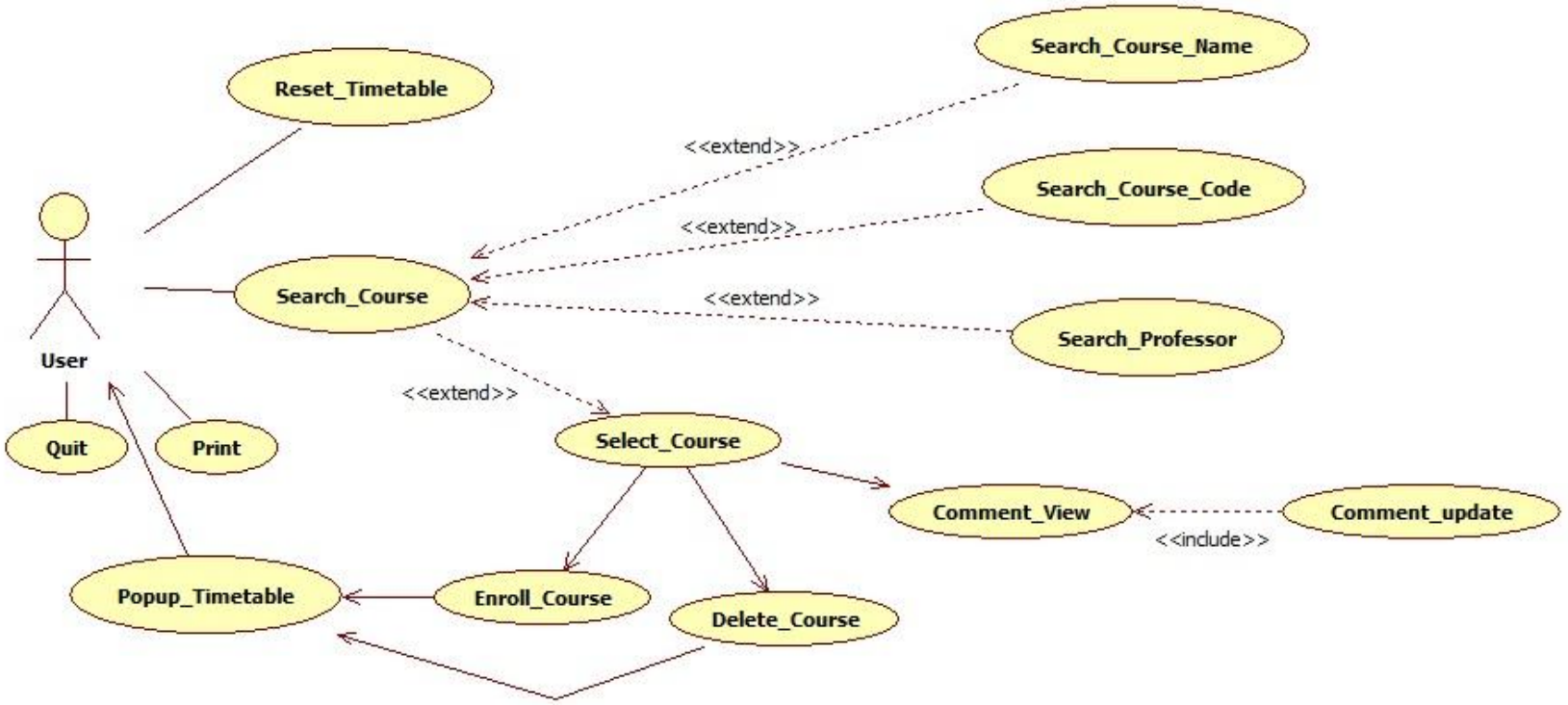


Rectangle : Data(Input/Output)

Oval : Transformations

Arrows : Data-flows

-6.2.3 Interface Description



-6.2.3 Interface Description

Use Case ID	Use Case 명	Use Case 설명	
U1	Search_Course	과목을 검색한다.	
		사용자	학생
		조건	과목검색 버튼을 입력
		흐름설명	1. 과목검색 버튼 입력 2-1. 과목 이름 입력 2-2. 과목 코드 입력 2-3. 교수 이름 입력 3. 서버에서 과목에 대한 데이터를 받아온다 4. 받아온 정보를 화면에 출력

Use Case ID	Use Case 명	Use Case 설명	
U2	Reset_Timetable	현재 등록되있는 시간표의 초기화 한다.	
		사용자	학생
		조건	리셋 버튼 입력
		흐름설명	1.리셋 버튼 입력 2.Timetable의 초기화

-6.2.3 Interface Description

Use Case ID	Use Case 명	Use Case 설명	
U3	Print	현재 나의 시간표를 프린터로 출력 한다.	
		사용자	학생
		조건	출력 버튼 입력
		흐름설명	1. 출력 버튼 입력 2. 현재 나의 Timetable을 프린터로 출력

Use Case ID	Use Case 명	Use Case 설명	
U4	Quit	현재 사용하고 있는 시간표 프로그램을 종료 한다.	
		사용자	학생
		조건	종료 버튼 입력
		흐름설명	1. 종료 버튼 입력 2. 현재 사용하고 있는 시간표 프로그램 종료

-6.2.3 Interface Description

Use Case ID	Use Case 명	Use Case 설명	
U5	Select_Course	검색한 과목을 선택한다.	
		사용자	학생
		조건	과목 검색후 선택
		흐름설명	1. 검색된 과목중 하나를 선택 2. 서버에서 불러온 데이터를 현재 나의 상황 (시간,학점,요일 등)과 비교 3-1. 불가능: 1로 이동 3-2. 가능: 과목을 선택

Use Case ID	Use Case 명	Use Case 설명	
U6	Enroll_Course	선택한 과목을 나의 시간표에 등록 한다.	
		사용자	학생
		조건	과목 선택후 등록 가능
		흐름설명	1. 등록 버튼 입력 2. 나의 시간표 데이터로 이동

-6.2.3 Interface Description

Use Case ID	Use Case 명	Use Case 설명	
U7	Delete_Course	등록되어있는 과목중 원치 않는 과목을 등록 삭제한다.	
		사용자	학생
		조건	나의 시간표에 미리 등록 되어있어야 가능
		흐름설명	1. 삭제 버튼 입력 2. 변경된 나의 시간표 데이터로 이동

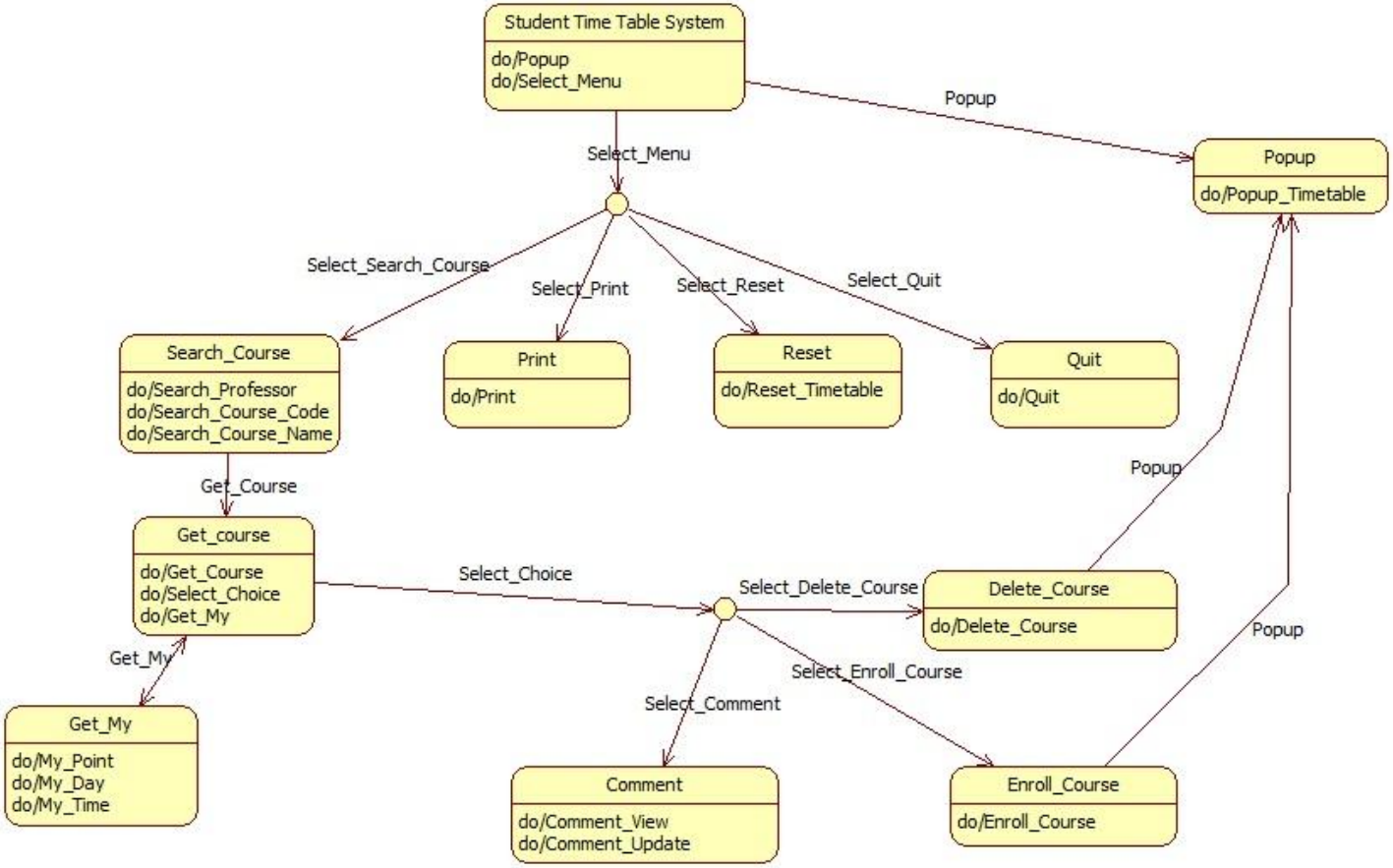
Use Case ID	Use Case 명	Use Case 설명	
U8	Comment_View	현재 선택한 과목에 대한 타 학생들이 적어놓은 코멘트를 본다.	
		사용자	학생
		조건	미리 작성된 코멘트가 존재 없을 경우: 공백이 뜨거나 Comment_Update로 이동
		흐름설명	1. 코멘트 보기 버튼 입력 2-1. 작성되어있는 코멘트를 화면에 출력 2-2. 코멘트가 없을 경우 비어있는 게시판이 출력

-6.2.3 Interface Description

Use Case ID	Use Case 명	Use Case 설명	
U9	Comment_Update	현재 과목에 대한 나의 코멘트를 작성 한다.	
		사용자	학생
		조건	욕설 및 은어 불가능
		흐름설명	1. 코멘트 페이지에 있는 나의 코멘트작성 버튼을 입력 2. 욕설 및 은어를 배제한 코멘트 작성 3. 서버에 저장

Use Case ID	Use Case 명	Use Case 설명	
U10	Popup_Timetable	현재 나의 시간표를 팝업하여 화면에 출력 한다.	
		사용자	학생
		조건	프로그램 실행시 자동 팝업 나의 데이터 변경시 업데이트
		흐름설명	1. 프로그램이 실행되면 자동으로 팝업 2-1. 과목 등록 2-2. 과목 삭제 2-3. 시간표 초기화 3. 나의 시간표 데이터로 이동 4. 업데이트 된 나의 시간표를 화면에 출력

-6.2.4 Detailed Design Description



감사합니다.