

# SOFTWARE DESIGN SPECIFICATION

“한양대학교 학부생  
시간표 조합  
웹 애플리케이션”

2012027390  
김명재

2014004511  
김호중

# TABLE OF CONTENTS

<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
<b>2. Overall View</b>	<b>5</b>
<b>3. Features to be Designed</b>	<b>7</b>
<b>4. Design: Developing Environment</b>	<b>9</b>
4.1 Scraper and Parser	9
4.2 Database Management System	10
4.3 Server	10
4.4 Front-end	10
<b>5. Design: Preparing Lecture Data</b>	<b>11</b>
5.1 Scraper	11
5.2 Parser	11
<b>6. Design: Storing Data</b>	<b>12</b>
<b>7. Design: Server</b>	<b>13</b>
7.1 Connection to the DBMS	13
7.2 Web server to handle HTTP Requests	13
7.3 Pseudo Codes of the server	14
<b>8. Design: Front-end</b>	<b>15</b>
8.1 Objects of DB Accessing Layer	16
8.1.1 DBAccessor	16
8.2 Objects of Data Controlling Layer	17
8.2.1 LectureDataController	17
8.2.2. SelectedLectureDataController	18
8.3 Objects of User Interface	19
8.3.1 MainHeader	19
8.3.2 MajorCulturalButtons	20

8.3.3 SearchBar	21
8.3.4 LectureTable	22
8.3.5 LectureBasket	23
8.4 Objects of Time Table Calculating	25
8.4.1 CombinationCondition	25
8.4.2 TimeTable	27
8.5 Objects for Additional Features and User Interfaces	29
8.5.1 LoginWindow	29
8.5.2 MyPage	30
8.5.3 PopularLectures	31

# 1. Introduction

## 1.1 Purpose

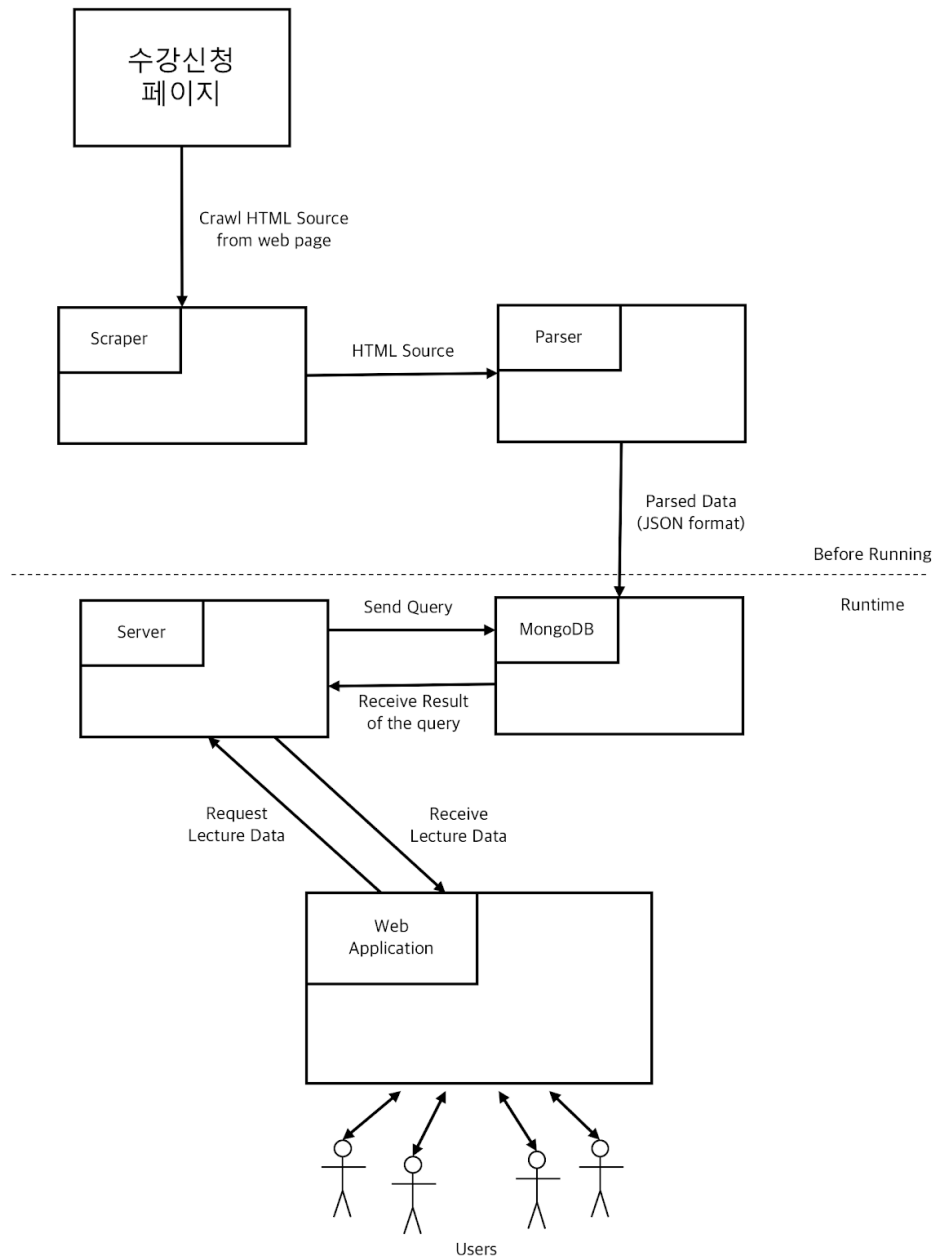
이 문서(Software Requirements Specification)의 목적은 '한양대학교 학부생 시간표 조합 웹 애플리케이션(이하 시간표 앱)의 설계를 자세하게 표현하는 것이다.

문서의 내용 순서는 다음과 같다.

- Overall View
- Features to be Designed
- Design: Developing Environment
- Design: Preparing Lecture Data
  - Scraper
  - Parser
- Design: Storing Data
  - MongoDB
- Design: Server
  - Sending HTML
  - Seding Lecture Data
- Design: Front-end
  - Classes for User Interface
  - Classes for features

먼저 '시간표 앱'의 전체적인 구성을 Diagram으로 나타내고, Design해야 할 기능들을 표현할 것이다. 이후에 Design의 결과물들을 서술할 것이다.

## 2. Overall View



<Overview Diagram>

프로그램의 전체적인 흐름은 위의 다이어그램과 같다. 'Before Running'에서 Data를 준비하고, 준비된 데이터를 바탕으로 프로그램을 실행한다. 'Runtime'에서 사용자의 입력을 받아 처리 결과를 보여준다.

## **Before running**

Data준비는 스크레이핑과 파싱으로 이뤄진다. Scraper를 통해 HTML Source를 받아오고, 받아온 source를 parser에 집어넣어서 json 형태의 data를 추출한다. 추출한 data를 NoSQL DB인 MongoDB에 보관한다. MongoDB를 선택한 이유는 json 포맷의 지원이 우수하며, '시간표 앱'은 복잡한 query를 사용할 일이 없기 때문이다. SQL을 사용할 수 없지만 json을 지원하면서 가볍게 사용할 수 있는 DBMS이기 때문에 MongoDB를 선택했다.

## **Runtime**

데이터가 준비가 되면 웹 server를 실행할 수 있다. 웹 server를 실행하면 server가 MongoDB 데몬에 새로운 세션을 열어 접속하게 되고, 사용자는 웹 앱에 접속할 수 있다. 사용자가 웹 앱에 접속해서 과목 검색을 하면, server는 사용자의 입력을 받아서 DBMS에 요청을 해 데이터를 받아오고, 받아온 데이터로 웹 페이지를 만들어 사용자에게 보여준다.

## 3. Features to be Designed

'시간표 앱'의 기능은 'Core features'와 'Additional Features'로 구분할 수 있다. 'Core features'는 앱의 정상 작동을 위해 필수로 구현해야 할 기능들이고, 'Additional Features'는 프로젝트의 시간 제한에 맞추어 구현할 수 있을 만큼만 구현하면 된다. 'Additional Features'는 앱의 동작이 필수적인 기능은 아니기 때문에, 앱을 런칭(Launching)한 뒤에 구현해도 늦지 않다.

### Core Features

1. 강의 데이터 스크레이핑 & 파싱
  - a. '시간표 앱' 제작을 위해서 가장 먼저 구현해야 할 기능
  - b. 한양대학교 수강신청 페이지를 스크레이핑해서 HTML Source 획득
  - c. HTML Source를 파싱해서 의미있는 데이터 추출
2. 전공 강의, 교양 강의 검색
  - a. 구글 메인 페이지처럼 검색 바(search bar)가 메인페이지의 중간에 있다.
  - b. 전공 강의 검색: 전공을 선택하면 해당 전공의 모든 학년 강의를 볼 수 있다.
  - c. 교양 강의 검색: 조건을 선택하면 조건에 맞는 교양 강의를 카드 형태(=Metro ui)로 보여준다. (시간, 요일, 학점 등 복수의 조건 선택 가능)
3. 강의 정보 표시
  - a. 선택한 강의에 대한 상세정보를 알 수 있다. (교수명, 강의 시간, 강의실 등등)
  - b. 강의 이름 위에 마우스를 올리면 정보를 마우스 커서 근처에 띄워준다.
4. 강의 장바구니
  - a. 강의를 장바구니에 담거나 삭제할 수 있다.
  - b. 강의가 담겨있는 순서를 바꿔 우선순위를 조절할 수 있다.
5. 조합 조건 설정
  - a. 조건 설정을 통해 원치 않는 시간표를 제거할 수 있다.
  - b. 조건들은 강의 수, 총 학점, 공강날 설정 등등이 있다.
6. 시간표 조합

- a. 선택한 강의들과 조건을 통해서 성립 가능한 모든 경우의 시간표를 만든다.

## **Additional Features**

### 7. 로그인

- a. 한양대학교, 구글, 페이스북 계정으로 로그인 할 수 있다.
- b. 한양대학교 API를 통해 로그인 하면 전공 정보를 받아와 자동으로 전공 검색을 해준다.

### 8. 시간표 저장

- a. 로그인을 한 상태에서 제작한 시간표를 저장할 수 있다.

### 9. 시간표 수정하기

- a. 저장한 시간표를 수정할 수 있다.
- b. 수정 버튼을 누르면 시간표에 담겨있는 강의가 담겨있는 시간표 제작 화면으로 이동한다.

### 10. 인기 강의 Hashtag

- a. 사용자들이 많이 선택한 강의를 메인 페이지에 띄워준다.

### 11. 시간표 동선

- a. 만들어진 시간표를 바탕으로 한양대학교 지도 위에 시간표를 따라 동선을 그려준다.



## 4. Design: Developing Environment

'시간표 앱'은 웹 애플리케이션으로서, 사용자의 웹 브라우저 위에서 동작한다. 이 앱은 무거운 계산을 하거나 화려한 그래픽을 사용하지 않는 가벼운 앱이다. 웹 표준에 따라서 구현하기 때문에 사용자의 운영체제와 상관없이 웹 브라우저를 쓸 수 있다면 모든 곳에서 앱을 사용할 수 있다.

Chrome, Firefox, Safari, Edge, Opera 총 5개의 웹 브라우저에서 동일한 User Experience(UX)를 보장할 것이며, Internet Explorer는 최신의 웹 표준을 지키지 못하기 때문에 지원 대상에서 제외했다.

프로젝트를 큰 묶음으로 나뉘보면 Crawler와 Parser, DBMS, Server, Front-end로 나뉜다. 여기서는 각각의 부분들을 어떤 환경에서 개발할 것인지 서술한다.

### 4.1 Scraper and Parser

#### Scraper

Front-end를 제외한 모든 영역에서 기본적으로 Golang을 사용한다. Golang은 맥, 윈도우, 리눅스를 모두 훌륭하게 지원하는 멀티플랫폼 언어이다. Python을 사용해도 충분히 구현할 수 있겠지만 새롭게 떠오르는 언어를 공부하는 겸 해서 Golang으로 언어를 정했다. 언어차원에서 지원해주는 기능이 많아 간단한 웹 Scraping 프로그램은 물론이고 웹 Server도 손쉽게 만들 수 있다.

#### Parser

HTML Source에서 정보를 뽑아내는 것도 Golang에서 처리할 수 있다. Golang의 HTTP Package를 활용하면 HTML Tag 단위로 정보를 읽어와 원하는 attribute나 value를 얻을 수 있다. 정보를 뽑아 낸 뒤 JSON Package를 활용해서 MongoDB에 보관할 Data를 생성한다.

## 4.2 Database Management System

DBMS는 MongoDB를 사용한다. MongoDB는 json 형태로 데이터를 저장한다. Javascript의 구조체와 형태가 같기 때문에 웹 앱을 개발할 때 데이터를 편하게 다룰 수 있다. 게다가 MongoDB는 NoSQL DBMS다. '시간표 앱'은 복잡한 SQL이 필요한 앱이 아니기 때문에 NoSQL DBMS로도 충분하다.

## 4.3 Server

Golang의 HTTP Package를 활용하면 서버를 쉽게 만들 수 있다. 애플리케이션을 만드는 김에 웹 서버도 Golang으로 구현해 보려고 한다. RESTFul API에 대한 기본적인 이해만 있으면 어렵지 않게 서버를 구현할 수 있다.

## 4.4 Front-end

Front-end는 웹페이지다. 웹을 개발하기 위해 Javascript와 CSS Framework를 사용할 것이다. 최근에 들어서 ReactJS나 AngularJS같은 깊이 있는 Javascript user interface framework가 등장했지만, 그중에서도 진입장벽이 낮은 VueJS로 개발을 할 것이다. CSS Framework는 Semantics CSS를 선택했다. Ajax(Asynchronous Javascript and XML) 기법을 활용해서 One page application을 구현해 사용자에게 웹이 아니라 Native app(운영체제 API를 활용해서 만든 앱)과 같은 User Experience를 줄 것이다.

## 5. Design: Preparing Lecture Data

### 5.1 Scraper

Scraper의 목적은 간단하다. 한양대학교 수강신청 페이지의 HTML Source code를 가져오는 것이다. 구글에서는 chromedriver라는 프로그램을 제공해서 사용자가 code로 Chrome web browser를 통제할 수 있도록 돕는다. 아래 코드는 chromedriver를 사용해서 스크레이핑 하는 프로그램을 디자인 한 것이다. 스크레이핑이 끝나면 수강신청 페이지의 source code를 얻을 수 있다.

#### Pseudo Code

```
Run a chrome browser using chromedriver
Go to the course registration site of Hanyang University
Login
Go to the course catalog
Click search button
Until the end of search result page
    Save each page's HTML source code
End Until Loop
Program exit
```

### 5.2 Parser

Scraper가 얻어온 HTML Source에서 의미 있는 데이터를 추출하는 과정은 Parser가 담당한다. Parsing이 끝나면 json 형태의 강의 정보 data가 만들어진다. Parser에는 강의 정보를 저장할 수 있는 적절한 형태의 structure가 있어야 한다.

Parser의 Pseudo code는 다음과 같다.

### **Pseudo Cdoe**

Iterate All of HTML source files

    Find lecture table

    Until at the end of the table

        Add each lecture data to an array

    Until loop end

Iteration End

Print parsed data to json file

---

## **6. Design: Storing Data**

---

json 포맷으로 만들어진 data를 MongoDB에 넣으면 데이터 보관이 끝난다. 데이터를 넣는 것은 간단한 shell command로 수행할 수 있다. 후에 MongoDB server를 실행한 뒤 웹 server에서 DB Server에 접속해 데이터를 사용할 것이다.

## 7. Design: Server

웹 server는 브라우저에 HTML 페이지를 제공하고, 사용자의 입력을 받아서 DBMS와 연동해 사용자가 원하는 정보를 브라우저에게 보내준다. 웹 Server의 동작을 위해서는 첫 번째로 DBMS와의 연결, 둘째로 브라우저의 HTTP Request를 처리할 수 있는 handler, 세 번째로 사용자의 입력(Ajax call)을 처리할 수 있는 handler가 필요하다.

### 7.1 Connection to the DBMS

웹 server의 초기화 단계에서 DBMS와의 연결이 필요하다. DBMS와 연결되지 않으면 웹 server는 동작할 수 없다.

### 7.2 Web server to handle HTTP Requests

웹 Front-end와의 communication을 위해서 다음과 같은 3개의 handler가 필요하다.

- Root handler
  - 기본적인 handler이다. 접근이 허용된 url 경로에 해당하는 file을 browser에 보내준다. 이 file들을 html, css, javascript 혹은 image나 기타 모든 파일이 될 수 있다.
- Major Lecture Data Handler
  - 해당하는 Major의 모든 Lecture Data를 브라우저에 보내준다.
  - DBMS에 접속해 해당하는 data를 받아와 브라우저에 보낸다.
- Cultural Lecture Data Handler
  - 교양 선택 조건에 해당하는 모든 교양 Lecture Data를 브라우저에 보내준다.
  - DBMS에 접속해 해당하는 data를 받아와 브라우저에 보낸다.

## 7.3 Pseudo Codes of the server

### **Main**

```
Check whether DB server is on
    If DB server is off, Exit
Register root handler
Register Major Lecture Data Handler
Register Cultural Lecture Data Handler
While at the end of the program
    Wait user request and run appropriate handler
End while
Exit
```

### **Root handler**

```
Parse input URL
Read a file that the URL indicates
Return the file to the browser
End of the handler
```

### **Major Lecture Data Handler**

```
Read major name from POST method
Send request for DBMS to find the major's lecture
Return structure array of lectures
```

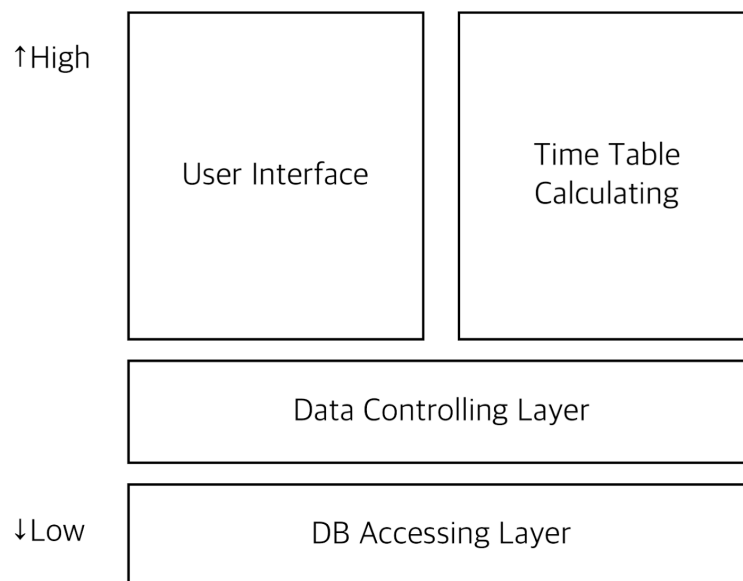
### **Cultural Lecture Data Handler**

```
Read cultural search condition from POST method
Send request for DBMS to find the major's lecture
Return structure array of lectures
```

## 8. Design: Front-end

Front-end의 design은 객체지향 패러다임을 따를 것이다. 모든 기능들은 객체 단위로 wrapping해서 관리한다. 각각의 객체들은 VueJS의 Object이다.

Front-end의 layer구성은 다음과 같다.



<Front-end Layer>

이 다이어그램은 Front-end만 나타냈다. Low layer의 아랫부분에는 Server가 있고, High layer의 위에는 사용자가 있다.

모든 Data는 Data controlling layer와 DB Accessing layer를 통해서 주고받게 된다. 실제 서버에 데이터 요청을 하는 곳은 DB Accessing layer이고, 요청한 데이터를 받아 저장하는 곳은 Data Controlling Layer이다. Data controlling layer의 data를 가지고 User Interface에서 user에게 강의 정보를 표시해주고, Time Table을 계산할 때도 Data controlling layer의 data를 가지고 계산한다.

## 8.1 Objects of DB Accessing Layer

### 8.1.1 DBAccessor

이 객체는 서버에 Data를 요청하는 역할을 한다. 멤버 변수는 존재하지 않고, LectureData의 배열을 return하는 메서드를 가진다. LectureData는 구조체인데, 학년, 반, 수업번호, 학수번호, 교과목명, 교강사, 학점, 수강정원, 수업시간, 강의실등의 정보를 가진다. 이 부분은 차후 구현 과정에서 명확히 할 것이다.

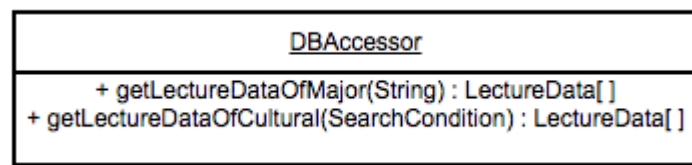
#### Member Variables

None

#### Methods

LectureData[] getLectureDataOfMajor(String majorName)

LectureData[] getLectureDataOfCultural(SearchCondition searchCondition)



<Class diagram of the DBAccessor>



## 8.2 Objects of Data Controlling Layer

### 8.2.1 LectureDataController

이 객체는 DBAccessor의 단순한 Wrapper object이다. 다른 객체는 DBAccessor를 직접 호출할 수 없고 이 객체를 통해 호출해야 한다. 이렇게 함으로써 DBAccessor와 High layer간의 거리를 넓히고, 미래에 기능을 추가할 수 있는 자유도를 높였다.

현재는 DBAccessor의 method를 부르는 정도지만 프로그램이 버전업하면서 여러가지 다른 기능이 추가될 것으로 예상된다.

Just a wrapper of the DBAccessor. Every objects must request lecture data through this object, not the DBAccessor. It is an abstraction of DBAccessor.

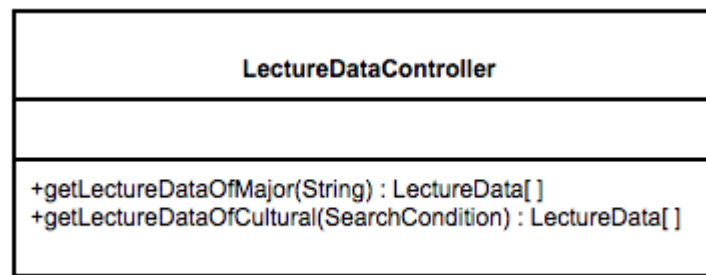
#### Member Variables

None

#### Methods

LectureData[] getLectureDataOfMajor(String majorName)

LectureData[] getLectureDataOfCultural(SearchCondition searchCondition)



<Class diagram of the LectureDataController>

### 8.2.2. SelectedLectureDataController

이 객체는 사용자가 선택한 강의의 Data들을 담는다. 이 Data들을 UI(User Interface)의 장바구니에서 사용자가 어떤 강의를 선택했는지 보여주는 곳에도 쓰이고, Time Table Calculating 과정에서도 사용된다.

멤버변수로는 LectureData의 배열이 있다. addLectureData와 removeLectureData 메서드로 멤버변수에 데이터를 추가, 제거할 수 있다.

#### Member Variables

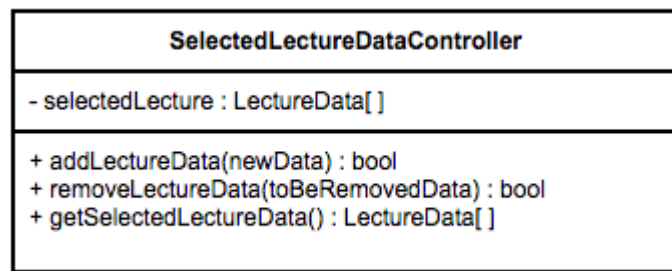
LectureData[] selectedLecture

#### Methods

bool addLectureData(LectureData newData)

bool removeLectureData(LectureData toBeRemovedData)

LectureData[] getSelectedLectureData(void)

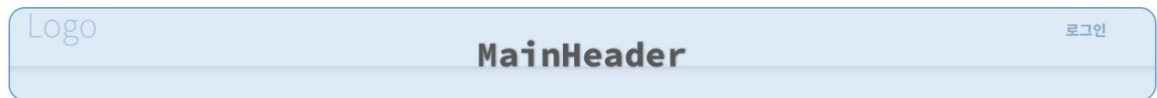


<Class diagram of the SelectedLectureDataController>

## 8.3 Objects of User Interface

### 8.3.1 MainHeader

이 객체는 메인 페이지의 header를 의미한다. Additional feature인 login과 my page의 버튼이 여기 헤더에 들어갈 것이다. 아래는 mockup(모ック업)틀로 만든 메인 페이지에서 MainHeader Interface의 영역을 나타낸 것이다.



Lorem ipsum dolor sit amet, consectetur adipiscing elit.



<Mockup Image of the MainHeader>

#### Member Variables

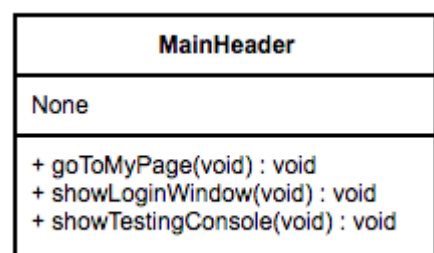
None

#### Methods

void goToMyPage(void)

void showLoginWindow(void)

void showTestingConsole(void)



<Class Diagram of the MainHeader>

### 8.3.2 MajorCulturalButtons

이 객체는 전공과 교양의 검색을 전환할 때 쓰이는 버튼이다. 목업 이미지에서 다음과 같이 나타난다.



<Mockup Image of the MajorCulturalButtons>

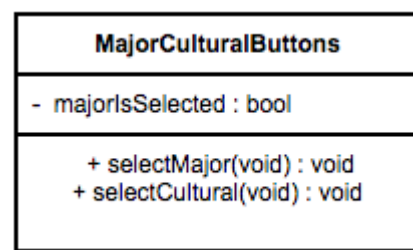
#### Member Variables

bool majorIsSelected

#### Methods

void selectMajor(void)

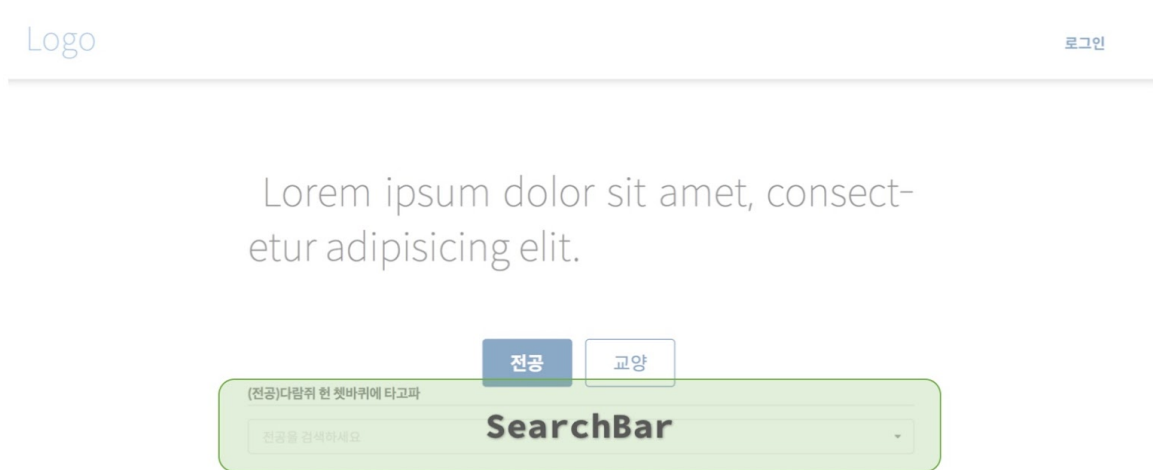
void selectCultural(void)



<Class Diagram of the MajorCulturalButtons>

### 8.3.3 SearchBar

이 객체는 검색바를 나타낸다. 사용자가 검색바에 입력을 하면 자동으로 DBMS에서 사용자 입력에 상응하는 데이터를 받아오게 구현할 것이다. 사용자 입력을 의미하는 DOM(Document Object Model)의 값이 바뀌는 event가 발생할 때 Server에 lecture data를 요청하도록 event handler를 등록할 것이다..



<Mockup Image of the SearchBar>

#### Member Variables

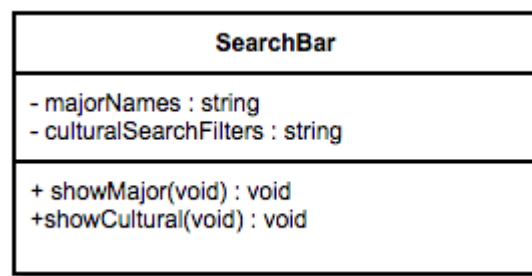
String[] majorNames

String[] culturalSearchFilters

#### Methods

void showMajor(void)

void showCultural(void)



<Class Diagram of the SearchBar>

### 8.3.4 LectureTable

이 객체는 Lecture Data를 받아서 강의가 담겨있는 테이블을 그려준다. 해당 강의를 클릭하면 강의 정보를 볼 수 있는 창이 나타난다. 이 객체에서 강의를 선택하면 위에서 설명했던 SelectedLectureDataController 객체에 강의가 담긴다.



<Mockup Image of the LectureTable>

#### Member Variables

LectureData[] lectures

#### Methods

void show(void)

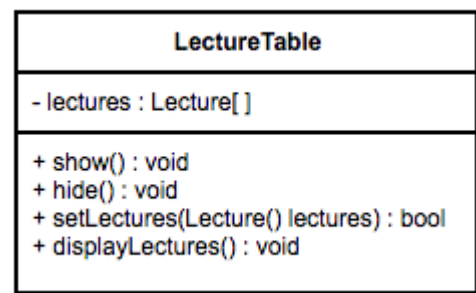
void hide(void)

bool setLectures(LectureData[] lectures)

void displayLectures(void)

void popupLectureInfo(void)

void selectLecture(void)



<Class diagram of the LectureTable>

### 8.3.5 LectureBasket

이 객체는 SelectedLectureDataController에 담겨있는 객체들을 표현하는 객체다. MetaData는 시간표를 조합할 때 해당 강의를 넣을지 말지 결정하는 boolean 변수를 담고 있다. 사용자가 LectureBasket에서 강의를 disable하면 시간표를 조합할 때 해당 강의를 포함하지 않는다. MetaData 구조체에는 필요에 따라 더 많은 데이터를 추가할 수 있다.

showConditionPanel()은 조합 조건을 설정할 수 있는 panel을 띄운다. generateTimeTable() 메서드를 호출하면 TimeTable 객체에 LectureData를 보내고 TimeTable 객체의 generate() 메서드를 호출한다.



<Mockup Image of the LectureBasket>

## Inner Struct

```
struct MetaData {  
  
    bool blocked;  
  
}
```

## Member Variables

MetaData[] lectureMetaData

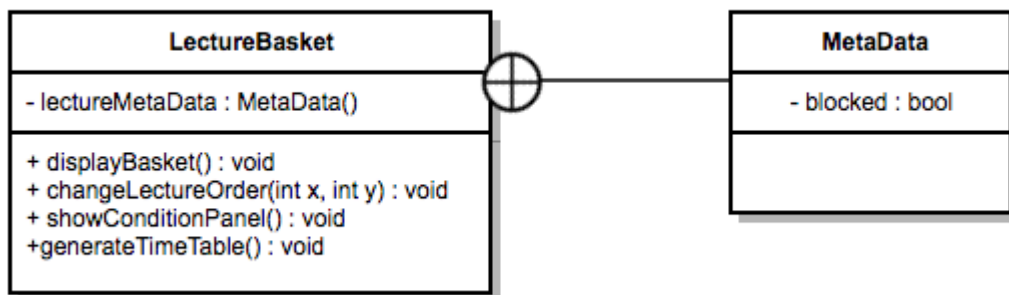
## Methods

void displayBasket(void)

void changeLectureOrder(int x, int y)

void showConditionPanel(void)

void generateTimeTable(void)



<Class diagram of the LectureBasket>

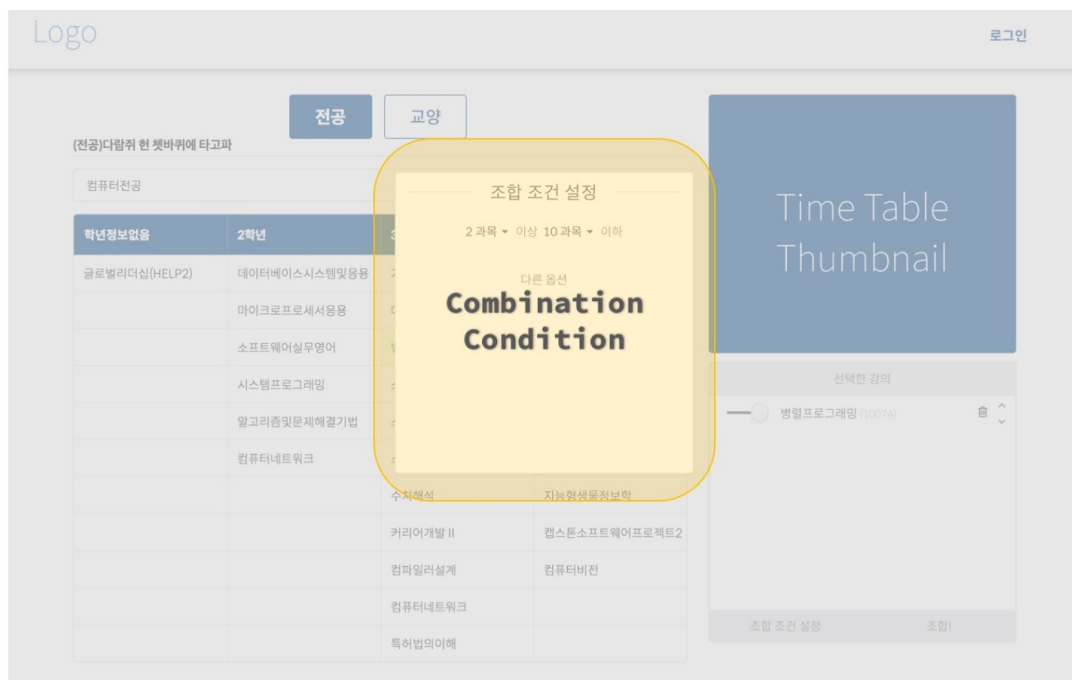


## 8.4 Objects of Time Table Calculating

시간표 조합에 쓰이는 2가지 객체는 CombinationCondition과 TimeTable 객체다. CombinationCondition객체는 LectureBasket을 통해서 값을 수정할 수 있다. LectureBasket 객체의 generateTimeTable() 메서드를 호출하면 그 안에서 TimeTable 객체의 generate() 메서드를 호출한다.

### 8.4.1 CombinationCondition

시간표 조합에 쓰이는 조건을 설정할 수 있는 객체다. 최대/최소 강의 수, 최대/최소 강의 학점을 설정할 수 있다. 기본적인 구현이 완료되면 공강날이나 아침선호, 저녁 선호 등의 조건도 추가될 것이다.



<Mockup Image of the CombinationCondition>

**Inner Struct**

```

struct CCondition {
    int numLecturesMin;
    int numLecturesMax;
    int numCreditMin;
    int numCreditMax;
}

```

**Member Variables**

CCondition condition

**Methods**

```

void show(void)
void hide(void)
void setNumLecturesMin(int num);
void setNumCreditMax(int num);
void setNumLecturesMax(int num);
void setNumCreditMin(int num);

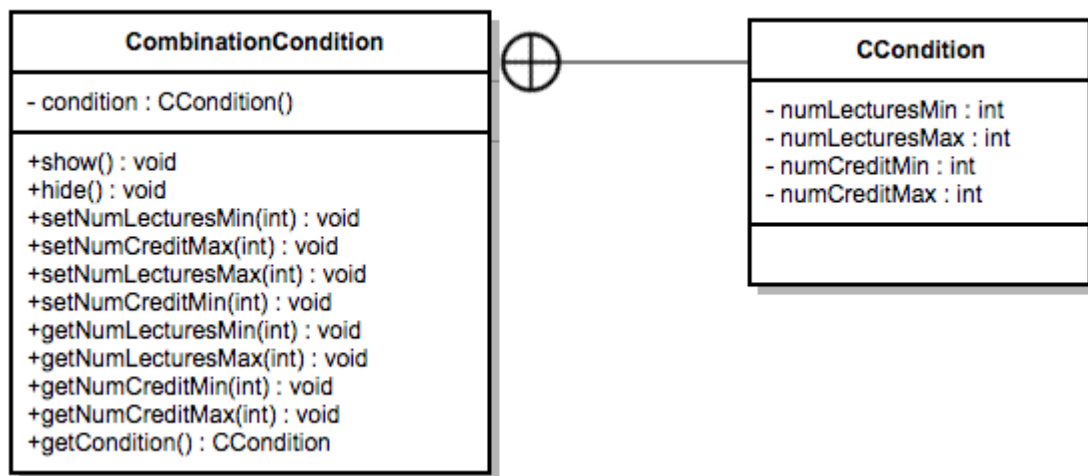
```

```

void getNumLecturesMin(int num);
void getNumLecturesMax(int num);
void getNumCreditMin(int num);
void getNumCreditMax(int num);

```

CCondition getCondition(void);



<Class diagram of the CombinationCondition>

### 8.4.2 TimeTable

이 객체는 LectureData 배열과 MetaData 배열을 받아서 TimeTable 배열을 출력한다. generate() 메서드 내부에 시간표 조합 알고리즘이 들어간다.

LectureData 배열에 시간표가 n개 있다면, 기계적으로 만들 수 있는 시간표는  $2^n$ 개이다. 모든 부분집합을 구하는 멍집합(PowerSet) 알고리즘을 통해서 모든 경우를 찾을 수 있다.

모든 경우에 대해서, CCondition의 최대/최소 강의수, 학점 조건에 맞지 않는 시간표를 걸러내고, 남아 있는 시간표중에서 강의 시간의 중복이 있는 시간표를 걸러낸 뒤 시간표들을 TimeTable 배열에 입력한다.

TimeTable 배열에 데이터가 생기면 사용자는 조합된 시간표를 볼 수 있다.

다음은 시간표 조합 알고리즘의 Pseudo code이다.

#### Pseudo Code

Find all subset of LectureData Array using PowerSet algorithm

Discard time tables that do not meet the condition

Discard time tables that have overlapped lecture time

Insert result to TimeTables array

	일	월	화	수	목	금	토
09:00			소프트웨어공학		소프트웨어공학		
09:30							
10:00							
10:30							
11:00							
11:30							
12:00							
12:30							
13:00							
13:30							
14:00							
14:30							
15:00		데이터베이스시스템 및 응용					
15:30							
16:00							
16:30							
17:00					데이터베이스시스템 및 응용		

<Mockup Image of the TimeTable>

### Member Variables

LectureData[] lectureData

MetaData[] MetaData

TimeTable[] timeTables

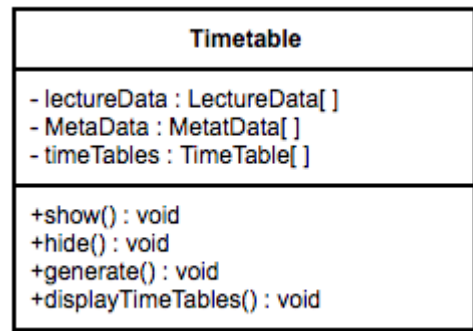
### Methods

void show(void)

void hide(void)

void generate(void)

void displayTimeTables(void)

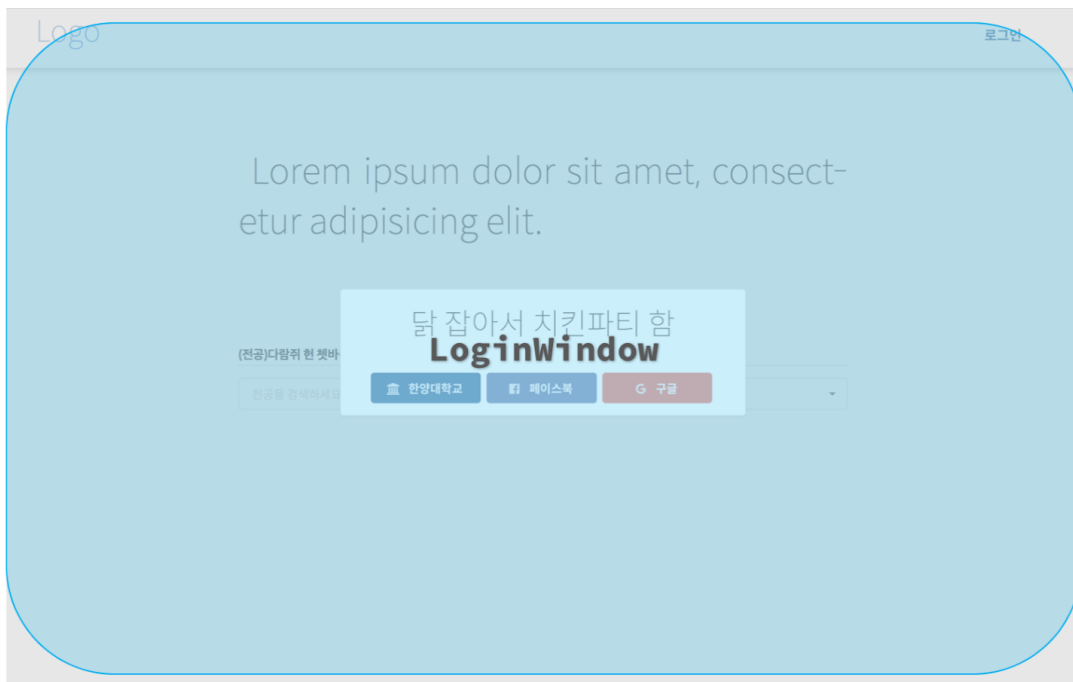


<Class diagram of the TimeTable>

## 8.5 Objects for Additional Features and User Interfaces

### 8.5.1 LoginWindow

로그인 버튼을 눌렀을 때 등장하는 창이다. ID, Password 입력창은 없고 로그인 API를 선택할 수 있는 버튼이 있다.



<Mockup Image of the LoginWindow>

#### Member Variables

None

#### Methods

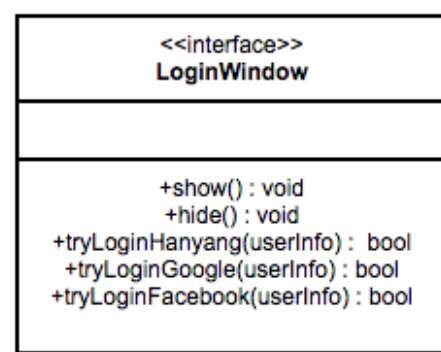
void show(void)

void hide(void)

bool tryLoginHanyang(NeededData userInfo)

bool tryLoginGoogle(NeededData userInfo)

bool tryLoginFacebook(NeededData userInfo)



<Class Diagram of the LoginWindow>

### 8.5.2 MyPage

이 객체는 로그인했을 때 로그인한 사용자의 My page를 보여준다. My page 객체에는 사용자가 이전에 저장했던 시간표가 담겨있고, 수정 버튼을 누르면 저장한 시간표의 상태에서 수정할 수 있다.

#### Member Variables

TimeTable[] savedTimeTables

#### Methods

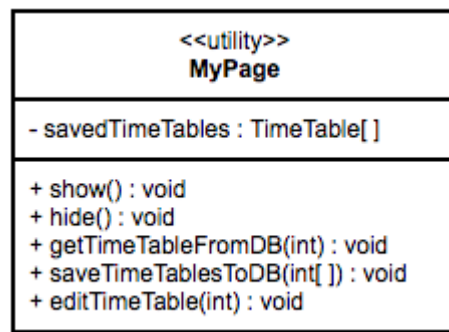
void show(void)

void hide(void)

void getTimeTableFromDB()

void saveTimeTablesToDB(int[] timeTableNumbers)

void editTimeTable(int timeTableNumber)



<Class Diagram of the MyPage>

### 8.5.3 PopularLectures

이 객체는 사용자가 많이 선택한 강의를 Main page에 띄워준다. DB에 사용자가 어떤 과목을 얼마나 선택했는지에 대한 정보가 있어야 하고, 해당 정보를 바탕으로 인기 있는 과목을 사용자에게 보여준다.

#### Member Variables

LectureData[] popularLectures;

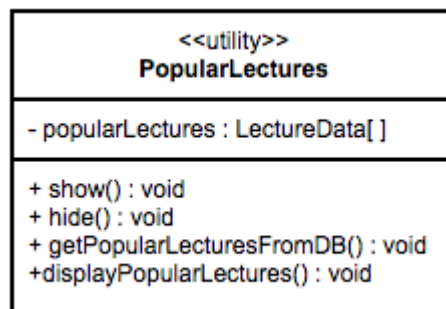
#### Methods

void show(void)

void hide(void)

void getPopularLecturesFromDB(void)

void displayPopularLectures(void)



<Class Diagram of the PopularLectures>

