

# CLASS MATE

User-friendly Timetable Program  
Software Design Description

TEAM 9

200911385 박기남

200911425 조서경

200911426 조성완

200911427 조아라

# Index

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
2. Reference.....	3
3. User interface design description.....	3
- Real Use case Descriptions & Storyboards	
4. Software architecture description.....	23
- Package diagram	
5. Interaction diagram.....	24
- Sequence diagram	
6. A Design Class Diagram.....	30
7. Database design description.....	31
7.1 Class 변수 및 메소드.....	31
7.2 Database schema.....	33

# 1. Introduction

## 1.1 Purpose

- 이 문서는 시간표 관리 시스템(명칭 : CLASS MATE)의 설계 내용을 서술하는데 그 목적이 있다.
- 프로젝트의 Plan & Analysis 단계에서의 Use case와 다이어그램을 기초로 하여 클래스 및 메소드를 보다 자세히 구체화하고자 한다.

## 1.2 Scope

- 이 시스템은 '시간표 관리' 기능과 더불어 '성적 관리' '일정 관리' '학교 정보보기' 의 기능을 추가로 포함한다.
- 이 시스템은 개인 사용자를 위한 것이며 암호를 이용한 사용자 인증을 통하여 다수의 사용자가 이용할 수 있다.
- 사용자의 PC는 Microsoft Visual C++ 6.0을 이용하여 만든 실행파일이 실행 가능하고 Internet Explorer 사용이 가능하다고 가정한다.

# 2. Reference

- IEEE Standard 1016-1998 - Recommended Practice for Software Design Descriptions
- Software Modeling & Analysis OSP Summary

# 3. User interface design description

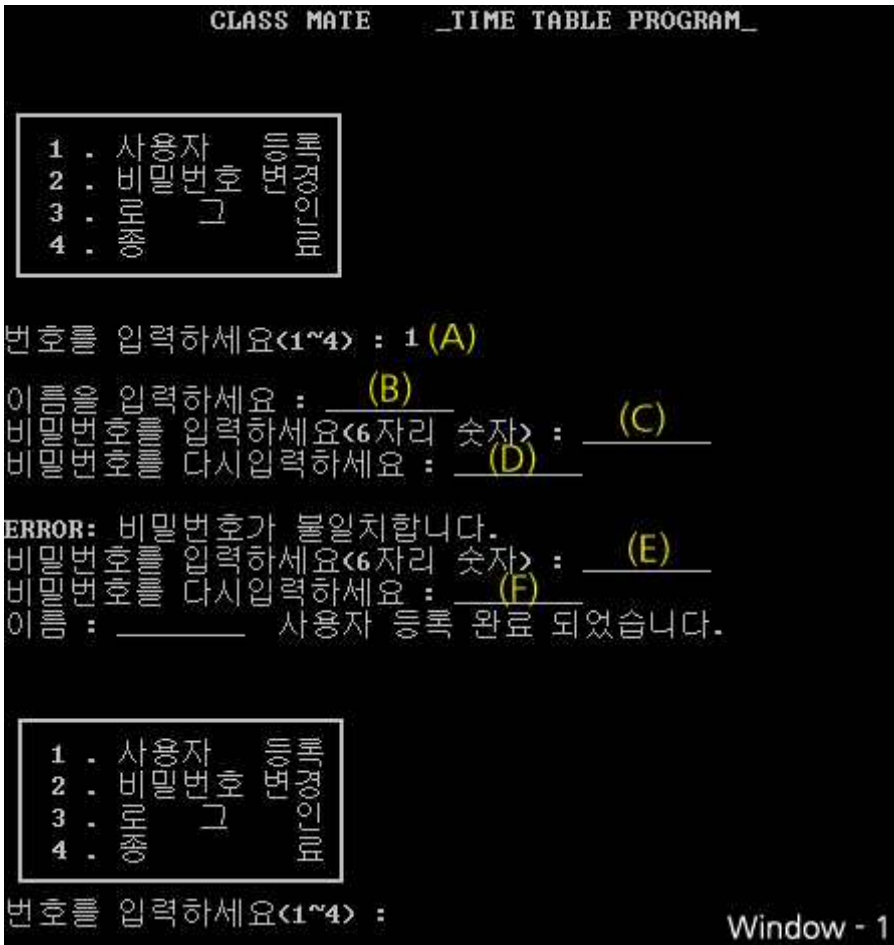
## Real Use case Descriptions & Storyboards

- 공통적인 Exceptional Courses of Events 설명

E1 : 이 시스템은 프로그램 실행에 필요한 모든 정보를 Text형태의 Database에서 읽어오고 프로그램 실행 중에 변경된 모든 사항을 다시 Database에 저장하므로 Text파일을 읽고 쓰기위해 DB에 접근할 때에는 반드시 '파일 열기' 과정이 제대로 실행되었는지를 확인하고 에러메시지를 출력하는 예외처리 과정을 거친다.

E2 : 이 시스템은 메뉴를 출력하고 사용자로부터 원하는 메뉴의 번호를 입력받는 형식으로 실행되므로 사용자로부터의 모든 입력은 유효한지 확인하고 유효하지 않은 입력일 경우 에러 메시지를 출력한 후 메뉴 번호를 다시 입력받는 예외처리 과정을 거친다.

### 3.1 사용자 등록



1: 사용자 등록 UI

Use Case	사용자 등록
Actor	사용자
Purpose	프로그램을 처음 사용하는 사용자의 정보 등록
Overview	사용자가 프로그램을 사용하기 위하여 자신의 이름과 비밀번호를 입력하고 시스템은 사용자 DB에 사용자가 입력한 정보를 저장한다.
Type	Primary
Cross Reference	Classes : FileIO, Identify Functions : InFile_Open(), OutFile_Open(), InFile_Close(), OutFile_Close(), Regist_User()
Pre-Requisites	N/A
UI Widgets	Window-1
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 사용자가 프로그램을 처음 실행시킨 후 '사용자 등록' 메뉴(A)를 선택하면 시작된다. (E2) 2. (S) 시스템이 사용자에게 이름과 비밀번호 입력을 요청한다. 이 때, 비밀번호는 2회 입력 받는다. 3. (A) 사용자는 B, C, D에 이름 및 패스워드를 입력한다.

	<p>4. (S) 시스템은 2회 입력한 비밀번호가 서로 일치하는지 확인하고(E3) 사용자 DB에 사용자 정보를 저장한다. (E1)</p> <p>5. (A) 사용자는 정보 등록이 완료 되었으며 앞으로 로그인을 통하여 시간표 관리 프로그램을 이용할 수 있다.</p> <p>6. (S) 사용자가 요청한 모든 과정을 수행한 후에는 메뉴를 다시 출력하여 사용자가 로그인하거나 프로그램을 종료할 수 있도록 한다.</p> <p>위와 같은 상위메뉴 재 출력은 사용자가 선택한 서브메뉴의 기능이 끝나면 항상 실행한다.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>E1, E2</p> <p>E3: 2회 입력한 비밀번호가 일치하지 않을 경우 에러 메시지를 출력하고 비밀번호를 다시 입력받는다.</p>

### 3.2 비밀번호 변경

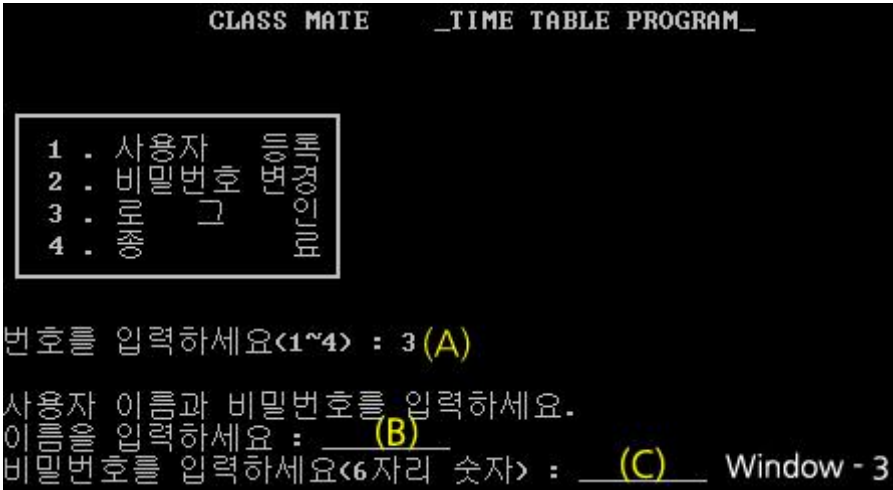


2: 비밀번호 변경 UI

Use Case	<b>비밀번호 변경</b>
Actor	사용자
Purpose	사용자의 비밀번호 변경
Overview	프로그램을 실행하여 사용자 등록을 했던 사용자의 비밀번호를 변경한다.
Type	Optional
Cross Reference	<p>Classes : FileIO, Identify</p> <p>Functions : InFile_Open(), OutFile_Open(), InFile_Close(), OutFile_Close(), Input_ID(), Change_Password(), Check_Password()</p>
Pre-Requisites	사용자는 비밀번호를 변경하기 위하여 사용자 등록을 통하여 사용자 정보를 입력하는 과정이 필요하다.
UI Widgets	Window-2
Typical Courses of Events	(A) : Actor, (S) : System

	<ol style="list-style-type: none"> <li>1. (A) 이 use case는 사용자가 프로그램을 처음 실행시킨 후 ‘비밀번호 변경’ 메뉴(A)를 선택하면 시작된다. (E2)</li> <li>2. (S) 시스템이 사용자에게 이름과 기존의 비밀번호 입력을 요청한다.</li> <li>3. (A) 사용자는 B, C에 이름 및 비밀번호를 입력한다. (E2)</li> <li>4. (S) 시스템은 사용자가 입력한 이름, 비밀번호가 사용자 DB에 저장되어 있는지 확인한다.(E1, E3)</li> <li>5. (S) 시스템이 사용자에게 새로운 비밀번호 2회 입력을 요청한다.</li> <li>6. (A) 사용자는 D, E에 비밀번호를 2회 입력한다.</li> <li>7. (S) 시스템은 사용자가 입력한 비밀번호가 일치하는지 확인한다.(E4)</li> <li>8. (S) 시스템이 비밀번호 변경완료 메시지를 출력한다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>E1, E2</p> <p>E3: 사용자 DB에서 사용자가 입력한 정보를 가진 사용자를 찾을 수 없는 경우, 에러 메시지를 출력하고 다시 입력받는다.</p> <p>E4: 사용자가 2회 입력한 비밀번호가 서로 일치하지 않을 경우, 에러 메시지를 출력하고 다시 입력받는다.</p>

### 3.3 사용자 인증



3: 로그인 UI

Use Case	사용자 인증
Actor	사용자
Purpose	사용자가 이름, 비밀번호 입력을 통해 자기 자신을 인증
Overview	사용자가 자신의 시간표 관리 프로그램에 접근하기 위해서 이름과 패스워드를 입력하여 로그인한다.
Type	Primary
Cross Reference	Classes : FileIO, Identify Functions : InFile_Open(), InFile_Close() Input_ID(), Check_Password()
Pre-Requisites	사용자는 로그인하기 위해서 자신의 이름과 비밀번호를 setting

	하는 과정이 필요하다.
UI Widgets	Window-3
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 사용자가 프로그램을 처음 실행시킨 후 '로그인' 메뉴(A)를 선택하면 시작된다. (E2) 2. (S) 시스템이 사용자에게 이름과 비밀번호 입력을 요청한다. 3. (A) 사용자는 B, C에 이름 및 비밀번호를 입력한다.(E2) 4. (S) 시스템은 사용자DB에서 입력한 정보를 가진 사용자가 있는지 확인한다. (E1, E3) 5. (A) 사용자 DB에 있는 정보와 사용자가 입력한 이름, 비밀번호가 일치하면 시스템에 로그인 된다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1, E2 E3: 사용자가 입력한 정보와 일치하는 사용자를 찾을 수 없을 경우 에러 메시지를 출력하고 다시 입력받는다.

### 3.4 프로그램 종료



4: 프로그램 종료 UI

Use Case	<b>프로그램 종료</b>
Actor	사용자
Purpose	사용자가 프로그램 종료
Overview	사용자가 프로그램 이용을 마치고 종료하고자 할 경우 종료 메시지 출력과 함께 프로그램을 종료한다.
Type	Primary
Cross Reference	Function : exit()
Pre-Requisites	N/A
UI Widgets	Window-4
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 사용자가 프로그램을 처음 실행시킨 후 '종료' 메뉴(A)를 선택하거나 또는 각 서브메뉴에서 '종료'

	메뉴를 선택할 경우에도 실행된다. (E2) 2. (S) 시스템은 프로그램 종료 메시지를 출력하고 프로그램을 종료한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E2

### 3.5 수강과목 추가



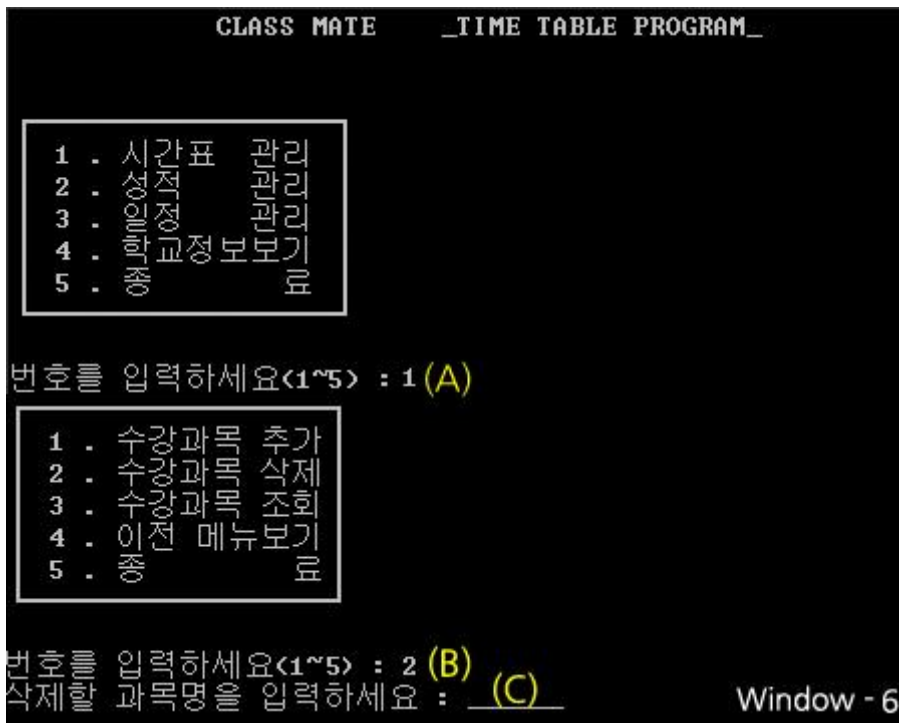
5: 수강과목 추가 UI

Use Case	수강과목 추가
Actor	사용자
Purpose	사용자가 자신이 수강하고 있는 과목을 시간표에 추가
Overview	사용자는 이 프로그램의 과목별 일정관리, 성적관리 등의 기능을 이용하기 위해서 자신이 수강하고 있는 과목을 시간표에 추가하고, 그 과목의 정보를 입력한다. (과목 정보: 과목명, 교수명, 강의시간, 강의장소, 학점)
Type	Primary
Cross Reference	Classes : FileIO, Identify, Subject Functions : InFile_Open(), OutFile_Open(),



	InFile_Close(), OutFile_Close(), Input_ID(), Check_Password() Add_SubInfo() Usecase : 사용자인증
Pre-Requisites	N/A
UI Widgets	Window-5
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 인증단계를 거친 사용자가 '시간표관리' 메뉴(A)를 선택 후 '수강과목추가' 서브메뉴(B)를 선택하면 시작된다. (E2) 2. (S) 시스템이 사용자에게 과목 정보의 입력을 요청한다. 3. (A) 사용자는 자신이 수강하고 있는 과목의 정보를 입력한다(C, D, E, F, G, H). 4. (S) 시스템은 사용자에게 입력받은 과목의 정보를 '과목정보 저장용 DB'에 저장한다. (E1)
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1, E2

### 3.6 수강과목 삭제



6: 수강과목 삭제 UI

Use Case	수강과목 삭제
Actor	사용자
Purpose	사용자가 자신이 시간표에 추가했던 과목을 삭제
Overview	사용자가 시스템에게 관리가 필요하지 않은 과목이나 잘못 추가한 과목을 삭제 요청한다.
Type	Primary

Cross Reference	Classes : FileIO, Identify, SubManager Functions : InFile_Open(), OutFile_Open(), InFile_Close(), OutFile_Close(), Input_ID(), Check_Password() Delete_Subject() Usecases : 사용자인증, 과목검색
Pre-Requisites	사용자는 자신이 기존에 추가했던 과목만 삭제를 요청 할 수 있다.
UI Widgets	Window-6
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 인증단계를 거친 사용자가 '시간표관리' 메뉴(A)를 선택 후 '수강과목삭제' 서브메뉴(B)를 선택하면 시작된다. (E2) 2. (S) 시스템이 사용자에게 삭제하고 싶은 과목의 이름을 입력받는다. 3. (A) 사용자는 삭제하고자 하는 과목의 이름을 입력한다(C). 4. (S) 시스템은 '과목정보 저장용 DB'에 사용자가 입력한 과목이 존재하는 지 검색한다. (E1,E3) 5. (S) 시스템은 '과목정보 저장용 DB'에서 사용자에게 입력받은 과목에 관한 모든 정보를 지운다.(E1)
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1, E2 E3: '과목정보 저장용 DB'에 사용자가 입력한 과목이 존재하지 않으면(사용자가 추가한 적 없는 과목을 삭제하고자 할 때) 에러 메시지를 출력한다.

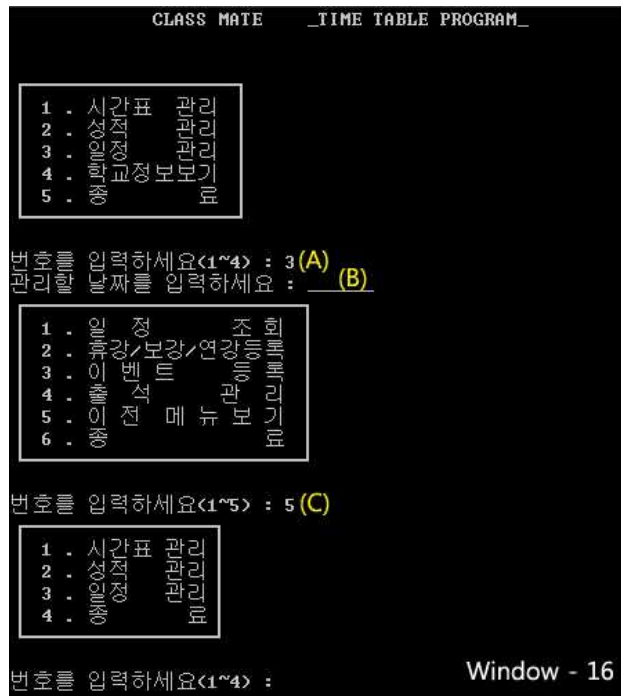
### 3.7 수강과목 조회



7: 수강과목 조회 UI

Use Case	수강과목 조회
Actor	사용자
Purpose	사용자가 자신이 시간표에 추가한 과목을 조회
Overview	사용자가 자신이 시간표에 추가한 과목을 조회하고자 하면 시스템이 콘솔에 보기 좋은 형태로 출력한다.
Type	Primary
Cross Reference	Classes : FileIO, Identify, Subject Functions : InFile_Open(), OutFile_Open(), InFile_Close(), OutFile_Close(), Input_ID(), Check_Password() Show_List() Usecases : 사용자인증, 과목검색
Pre-Requisites	N/A
UI Widgets	Window-7
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 인증단계를 거친 사용자가 '시간표관리' 메뉴(A)를 선택 후 '수강과목조회' 서브메뉴(B)를 선택하면 시작된다.(E2) 2. (S) 시스템은 '과목정보 저장용 DB'에서 이 사용자가 전에 추가한 과목들의 정보를 읽고 콘솔에 출력한다.(E1)
Alternative Courses of Events	N/A

3.8 이전 메뉴보기

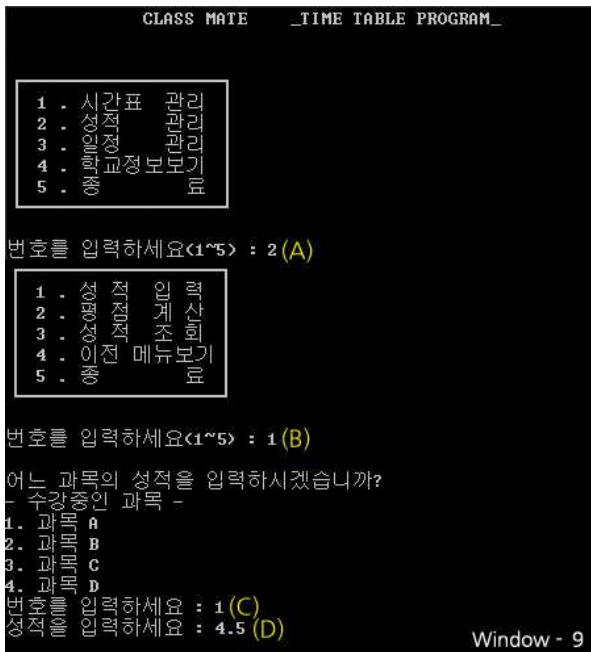


8 이전 메뉴보기 UI

Use Case	이전 메뉴 보기
Actor	사용자
Purpose	사용자가 상위 메뉴를 조회
Overview	사용자가 서브 메뉴를 선택했다가 다시 이전의 상위 메뉴를 보고자 할 경우 이전 메뉴를 다시 출력한다.
Type	Primary
Cross Reference	Function : by while

Pre-Requisites	N/A
UI Widgets	Window-8, Window-12, Window-16
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) 이 use case는 인증단계를 거친 사용자가 ‘시간표관리’ ‘성적관리’ ‘일정관리’ ‘학교정보보기’ 메뉴(A)를 중 어느 하나를 선택 후 ‘이전 메뉴보기’ 서브메뉴(B)를 선택하면 시작된다.(E2)</li> <li>2. (S) 시스템은 ‘시간표관리’ ‘성적관리’ ‘일정관리’ ‘학교정보보기’ ‘종료’ 메뉴를 포함하는 이전의 메뉴를 다시 출력하고 사용자가 메뉴번호를 입력할 수 있도록 한다.</li> <li>3. (A) 사용자는 원하는 메뉴 번호를 입력하여 새로운 기능을 실행할 수 있다. (E2)</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	E2

### 3.9 성적 관리



9: 성적관리 UI

Use Case	성적관리
Actor	사용자
Purpose	사용자가 자신이 시간표에 추가한 과목들의 성적을 관리
Overview	사용자가 수강하고 있는 과목들의 성적을 입력하여 한 눈에 확인할 수 있고, 평점을 확인할 수 있다.
Type	Primary
Cross Reference	Classes : FileIO, Identify, Grade Functions : InFile_Open(), OutFile_Open(), InFile_Close(), OutFile_Close(), Input_ID(), Check_Password() Input_Grade(), Read_FGrade(), Cal_Grade(),

	<p>Show_Grade()  Usecases : 사용자인증, 과목검색</p>
Pre-Requisites	<p>평점을 계산하기 위해서는 사용자가 '수강과목추가'를 할 때 과목정보 중 '학점'에 해당과목이 몇 학점짜리 과목인지 입력해야 한다.</p>
UI Widgets	<p>Window-9, Window-10, Window-11</p>
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>(A) 이 use case는 인증단계를 거친 사용자가 '성적관리' 메뉴(A)를 선택하면 시작된다. (E2)</li> <li>(A) 사용자가 서브메뉴를 선택(B)한다. (E2)</li> <li>(A) 사용자가 '성적 입력' 서브메뉴를 선택(A-1)  사용자가 '평점 계산' 서브메뉴를 선택(A-2)  사용자가 '성적 조회' 서브메뉴를 선택(A-3)  사용자가 '이전 메뉴보기' 서브메뉴를 선택(A-4)  사용자가 '시스템 종료' 서브메뉴를 선택(A-5)</li> <li>(S) 시스템은 사용자가 선택한 기능을 수행한다.</li> </ol>
Alternative Courses of Events	<p>A-1: 사용자가 '성적 입력' 서브메뉴를 선택할 경우</p> <ol style="list-style-type: none"> <li>(S) 사용자에게 과목 리스트를 보여주고 어떤 과목의 성적을 입력 받을 건지 선택을 요청한다. (E1)</li> <li>(A) 과목을 선택(Window-9 C)하고 성적을 입력(Window-9 D)한다. (E2)</li> <li>(S) 선택한 과목이 DB에 존재 하는지 확인 하고 입력받은 성적을 DB에 저장한다. (E1, E3)</li> </ol> <p>A-2: 사용자가 '평점 계산' 서브메뉴를 선택할 경우</p> <ol style="list-style-type: none"> <li>(S) DB에 저장된 성적 정보를 가지고 시스템 내부에서 평점 계산을 수행한다. (E1)</li> <li>(S) 계산된 평점을 사용자에게 보여준다.</li> </ol> <p>A-3: 사용자가 '성적 조회' 서브메뉴를 선택할 경우</p> <ol style="list-style-type: none"> <li>(S) DB에 저장된 모든 과목의 성적정보만을 읽어온다.(E1)</li> <li>(S) 사용자에게 읽어온 성적 정보를 리스트화하여 보여준다.</li> </ol> <p>A-4: 사용자가 '이전 메뉴보기' 서브메뉴를 선택할 경우  이전 메뉴 보기 Use case 실행, 이전 메뉴를 다시 출력한다.  :Window-12</p> <p>A- 5: 사용자가 '종료' 서브메뉴를 선택할 경우  프로그램 종료 Use case 실행, 프로그램을 종료한다.</p>
Exceptional Courses of Events	<p>E1, E2  E3 : DB에 저장되어있지 않은 과목의 성적을 입력하려고 할 경우 에러메시지를 출력한 후 다시 입력받는다.</p>

### 3.10 일정관리

CLASS MATE \_TIME TABLE PROGRAM\_

- 1 . 시간표 관리
- 2 . 성적관리
- 3 . 일정관리
- 4 . 학교정보보기
- 5 . 종료

번호를 입력하세요(1~4) : 3 (A)  
 관리할 날짜를 입력하세요 : (B)

- 1 . 일 정 조 회
- 2 . 휴강/보강/연강등록
- 3 . 이 벤 트 등 록
- 4 . 출 석 관 리
- 5 . 이 전 메 뉴 보 기
- 6 . 종

번호를 입력하세요(1~5) : 2 (C)

- 1 . 휴강 등록
- 2 . 보강 등록
- 3 . 연강 등록

번호를 입력하세요(1~3) : 1 (D)  
 \_\_\_\_\_ 수강과목입니다.  
 1. 과목 A  
 2. 과목 B  
 3. 과목 C  
 4. 과목 D  
 휴강처리 할 과목번호를 입력하세요 : (E)  
 휴강처리가 완료 되었습니다. Window - 13.1

CLASS MATE \_TIME TABLE PROGRAM\_

- 1 . 시간표 관리
- 2 . 성적관리
- 3 . 일정관리
- 4 . 학교정보보기
- 5 . 종료

번호를 입력하세요(1~4) : 3 (A)  
 관리할 날짜를 입력하세요 : (B)

- 1 . 일 정 조 회
- 2 . 휴강/보강/연강등록
- 3 . 이 벤 트 등 록
- 4 . 출 석 관 리
- 5 . 이 전 메 뉴 보 기
- 6 . 종

번호를 입력하세요(1~5) : 2 (C)

- 1 . 휴강 등록
- 2 . 보강 등록
- 3 . 연강 등록

번호를 입력하세요(1~3) : 2 (D)  
 보강처리 할 과목명을 입력하세요 : (E)  
 보강의 시작 시간을 입력하세요 : (F)  
 보강의 종료 시간을 입력하세요 : (G)  
 보강처리가 완료 되었습니다. Window - 13.2

CLASS MATE \_TIME TABLE PROGRAM\_

- 1 . 시간표 관리
- 2 . 성적관리
- 3 . 일정관리
- 4 . 학교정보보기
- 5 . 종료

번호를 입력하세요(1~4) : 3 (A)  
 관리할 날짜를 입력하세요 : (B)

- 1 . 일 정 조 회
- 2 . 휴강/보강/연강등록
- 3 . 이 벤 트 등 록
- 4 . 출 석 관 리
- 5 . 이 전 메 뉴 보 기
- 6 . 종

번호를 입력하세요(1~5) : 2 (C)

- 1 . 휴강 등록
- 2 . 보강 등록
- 3 . 연강 등록

번호를 입력하세요(1~3) : 3 (D)  
 \_\_\_\_\_ 수강과목입니다.  
 1. 과목 A  
 2. 과목 B  
 3. 과목 C  
 4. 과목 D  
 연강처리 할 과목번호를 입력하세요 : (E)  
 연강의 종료 시간을 입력하세요 : (F)  
 연강처리가 완료 되었습니다. Window - 13.3

CLASS MATE \_TIME TABLE PROGRAM\_

- 1 . 시간표 관리
- 2 . 성적관리
- 3 . 일정관리
- 4 . 학교정보보기
- 5 . 종료

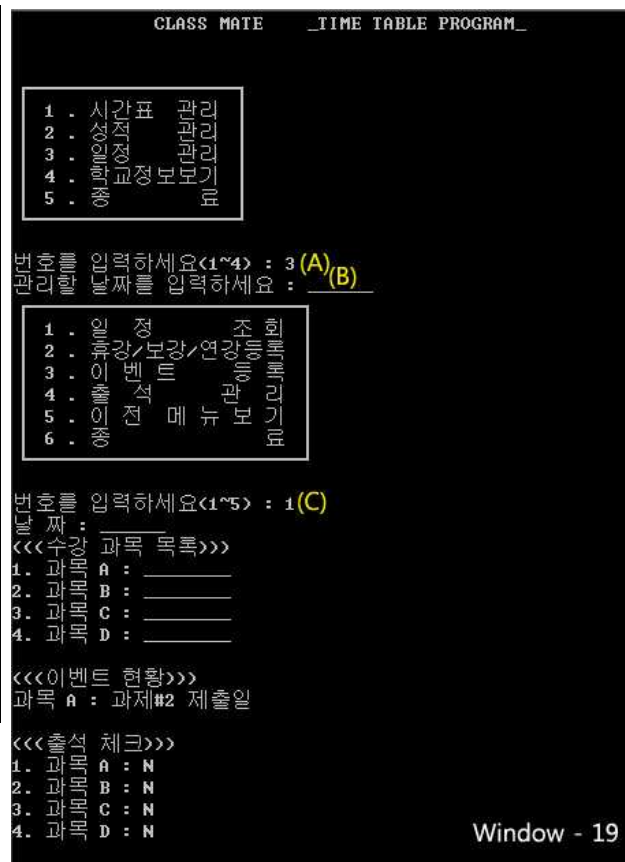
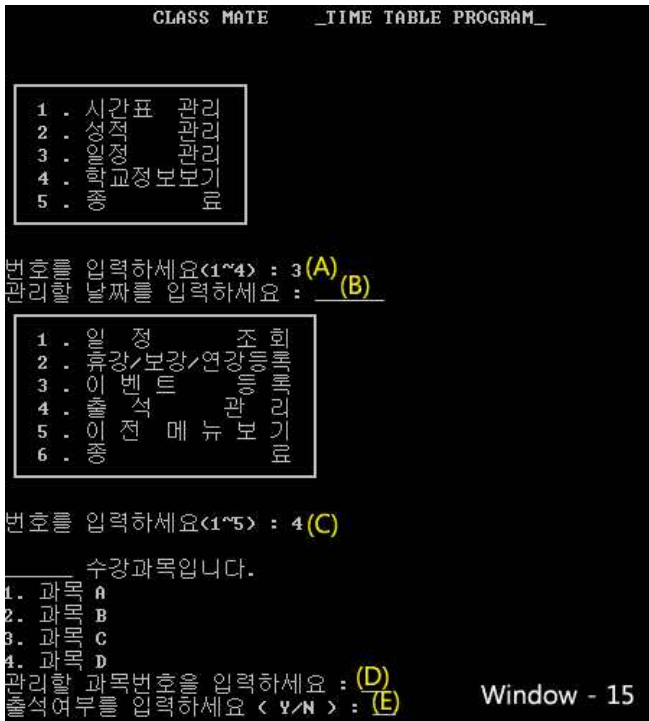
번호를 입력하세요(1~4) : 3 (A)  
 관리할 날짜를 입력하세요 : (B)

- 1 . 일 정 조 회
- 2 . 휴강/보강/연강등록
- 3 . 이 벤 트 등 록
- 4 . 출 석 관 리
- 5 . 이 전 메 뉴 보 기
- 6 . 종

번호를 입력하세요(1~5) : 3 (C)  
 \_\_\_\_\_ 수강과목입니다.  
 1. 과목 A  
 2. 과목 B  
 3. 과목 C  
 4. 과목 D  
 관리할 과목번호를 입력하세요 : (D)  
 이벤트 정보를 입력하세요 .  
 이벤트 이름 : (E)  
 이벤트 시작시간 : (F)  
 이벤트 종료시간 : (G)  
 이벤트 종료 : (H)

Window - 14





10\_2: 일정관리\_2 UI

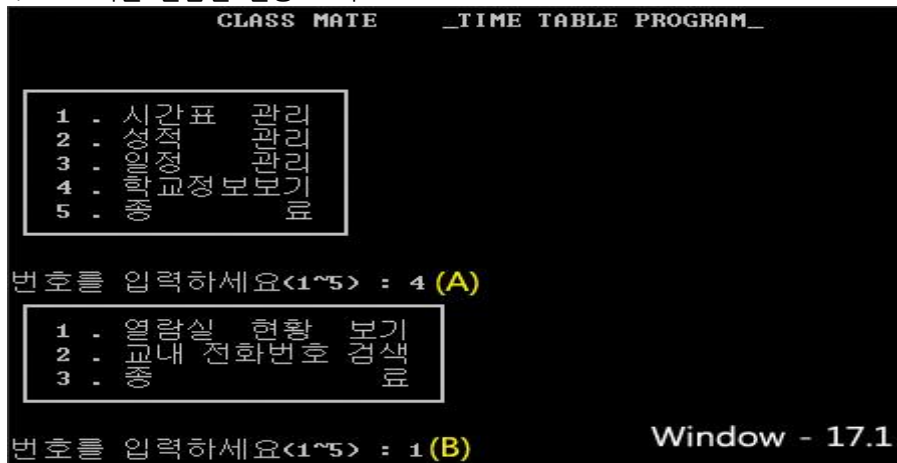
Use Case	일정관리
Actor	사용자
Purpose	사용자가 자신이 시간표에 추가한 과목에 관한 일정을 관리
Overview	사용자가 수강하고 있는 과목들의 휴강, 보강, 연강 등의 일정 등록하거나 이벤트 정보 입력, 또는 과목별 출석상태 등의 일정을 조회하고 관리할 수 있다. (이벤트의 예: 시험일정, 과제정보 및 기한 등록 등등)
Type	Primary
Cross Reference	Classes : FileIO, Identify, Date Functions : InFile_Open(), OutFile_Open(), InFile_Close(), OutFile_Close(), Input_ID(), Check_Password Set_StartDate(), Set_Subject(), Set_Date(), Show_Schedule(), Add_Schedule(), Add_ASubject(), Del_ASubject(), Extend_ASubject(), Check_Attend() Usecases : 사용자인증, 과목검색, 날짜검색
Pre-Requisites	사용자가 '수강과목추가' 기능을 통해 입력한 과목에 한해서 일정을 관리할 수 있다.
UI Widgets	Window-19, Window-13.1, Window-13.2, Window-13.3 Window-14, Window-15

<p>Typical Courses of Events</p>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) 이 use case는 인증단계를 거친 사용자가 '일정관리' 메뉴(A)를 선택하면 시작된다. (E2)</li> <li>2. (S) 시스템이 사용자에게 일정을 관리하고 싶은 날짜를 입력 요청한다.</li> <li>3. (A) 사용자가 원하는 날짜를 입력(B)한다. (E2)</li> <li>4. (S) 시스템은 서브메뉴를 출력하고 사용자로부터 메뉴 번호 선택을 요청한다.</li> <li>5. (A) 사용자가 원하는 메뉴의 번호를 선택(C)한다. (E2)</li> <li>6. (A) 사용자가 '일정 조회' 서브메뉴를 선택(A-1)        사용자가 '휴강/보강/연강등록' 서브메뉴를 선택(A-2)        사용자가 '이벤트 등록' 서브메뉴를 선택(A-3)        사용자가 '출석 관리' 서브메뉴를 선택(A-4)        사용자가 '이전 메뉴보기' 서브메뉴를 선택(A-5)        사용자가 '시스템 종료' 서브메뉴를 선택(A-6)</li> <li>7. (S) 시스템은 사용자가 선택한 기능을 수행한다.</li> </ol>
<p>Alternative Courses of Events</p>	<p>A-1: 사용자가 '일정 조회' 서브메뉴를 선택할 경우 Window-19        (S) 시스템은 '일정 관리 DB'에서 사용자가 입력한 날짜의 수강과목, 이벤트 현황, 출석 현황을 읽어 와서 콘솔창에 출력한다. (E1)</p> <p>A-2: 사용자가 '휴강, 보강, 연강처리' 서브메뉴를 선택할 경우</p> <ol style="list-style-type: none"> <li>1. (S) 시스템이 '휴강등록', '보강등록', '연강등록' 의 서브메뉴를 사용자에게 보여주고 선택을 요청한다.</li> <li>2. (A) 사용자가 원하는 메뉴(D)를 선택한다. (E2)</li> <li>3. (A) 사용자가 '휴강등록'을 선택(A-2-1)        사용자가 '보강등록'을 선택(A-2-2)        사용자가 '연강등록'을 선택(A-2-3)</li> <li>3. 시스템은 사용자가 선택한 기능을 수행한다.</li> </ol> <p>A-2-1: 사용자가 '휴강등록' 서브메뉴를 선택할 경우 Window 13.1</p> <ol style="list-style-type: none"> <li>1. (S) 시스템이 사용자가 선택한 날짜의 '일정관리용DB'에서 해당 날짜의 수강과목 리스트를 출력한다. (E1)</li> <li>2. (S) 사용자로부터 휴강처리 할 과목번호 입력을 요청한다.</li> <li>3. (A) 사용자는 휴강처리 하고자하는 과목 번호 (Window-13.1 E)를 입력한다. (E2)</li> <li>4. (S) 시스템은 '일정관리DB'에서 해당날짜에 사용자가 입력한 과목 정보를 삭제한다. (E1, E3)</li> <li>5. (S) 시스템은 휴강처리 완료 메시지를 출력한다.</li> </ol> <p>A-2-2: 사용자가 '보강등록' 서브메뉴를 선택할 경우 Window 13.2</p>

	<ol style="list-style-type: none"> <li>1. (S) 시스템이 사용자에게 보강 등록할 과목명, 강의 시작시간, 종료시간 입력을 요청한다.</li> <li>2. (A) 사용자는 보강 등록할 과목명(Window-13.2 E) 과 강의 시작시간(Window-13.2 F), 종료시간(Window-13.2 G)을 입력한다. (E2)</li> <li>3. (S) 시스템이 사용자가 선택한 날짜의 '일정관리용 DB'에 해당과목의 정보(과목명 및 강의시간)를 추가한다. (E1)</li> <li>4. (S) 시스템은 보강처리 완료 메시지를 출력한다.</li> </ol> <p>A-2-3: 사용자가 '연강등록' 서브메뉴를 선택할 경우 Window 13.3</p> <ol style="list-style-type: none"> <li>1. (S) 시스템이 '일정관리 DB'에서 사용자가 선택한 날짜의 수강과목 리스트를 읽어 와서 출력한다.(E1)</li> <li>2. (S) 시스템이 연강처리 할 과목의 번호, 강의 종료시간 입력을 요청한다.</li> <li>3. (A) 사용자가 연강처리 하고자하는 과목번호(Window-13.3 E), 강의 종료시간(Window-13.3 F)을 입력한다. (E2)</li> <li>4. (S) 시스템이 사용자가 선택한 날짜의 '일정관리용 DB'에서 해당과목의 강의시간을 연장하여 저장한다.(E1, E3)</li> <li>5. (S) 시스템은 연강처리 완료 메시지를 출력한다.</li> </ol> <p>A-3: 사용자가 '이벤트 등록' 서브메뉴를 선택할 경우 Window 14</p> <ol style="list-style-type: none"> <li>1. (S) 사용자에게 이벤트의 정보를 입력할 것을 요청한다.</li> <li>2. (A) 사용자가 이벤트의 정보(이벤트의 이름, 시작 및 종료시간, 메모)를 입력(Window-14 E, F, G, H)한다.(E2)</li> <li>3. (S) 시스템은 사용자가 입력한 정보를 '일정관리용 DB'에 저장한다.(E1)</li> </ol> <p>A-4: 사용자가 '출석 관리' 서브메뉴를 선택할 경우 Window 15</p> <ol style="list-style-type: none"> <li>1. (S) 시스템이 사용자가 입력한 날짜의 수강과목 리스트를 출력한다. (E1)</li> <li>2. (S) 시스템이 사용자에게 출석관리 하고자하는 과목의 번호 입력을 요청한다.</li> <li>3. (A) 사용자가 출석관리 할 과목의 번호를 입력한다.(Window-15 D) (E2)</li> <li>4. (S) 시스템이 사용자에게 출석여부를 묻는다.</li> <li>5. (A) 사용자는 출석여부를 Y/N로 입력한다.(Window-15 E) (E2)</li> <li>6. (S) 시스템은 사용자가 입력한 출석여부를 '일정관리용 DB'에 저장한다. (E1)</li> </ol>
--	---

	<p>A-5: 사용자가 '이전 메뉴보기' 서브메뉴를 선택할 경우 :Window - 16 이전 메뉴 보기 Use case 실행, 이전 메뉴 를 다시 출력한다.</p> <p>A-6: 사용자가 '종료' 서브메뉴를 선택할 경우 프로그램 종료 Use case 실행, 프로그램을 종료한다.</p>
Exceptional Courses of Events	<p>E1, E2</p> <p>E3 : 사용자가 과목추가를 하지 않은 과목을 휴강, 또는 연강 등록하려고 하는 경우 (DB에서 해당하는 과목을 찾지 못한 경우) 에러 메시지를 출력하고 다시 입력받는다.</p>

### 3.11 도서관 열람실 현황 보기



11\_1: 열람실 정보보기\_1 UI

건국대학교 상허도서관 열람실별 실시간 좌석 정보

2010-11-23 오후 1:29:39 현재

순번	열람실명	전체 좌석수	사용 좌석수	잔여 좌석수	이용률(%)
1	<a href="#">제1A 열람실</a>	300	165	135	55 %
2	<a href="#">제1B 열람실</a>	294	124	170	42.18 %
3	<a href="#">제2 열람실</a>	184	71	113	38.59 %
4	<a href="#">제3A 열람실</a>	332	120	212	36.14 %
5	<a href="#">제3B 열람실</a>	284	103	181	36.27 %
6	<a href="#">제4 열람실</a>	172	65	107	37.79 %
7	<a href="#">제5 열람실</a>	168	36	132	21.43 %
8	<a href="#">제6 열람실</a>	252	129	123	51.19 %
9	<a href="#">국가고시연구실</a>	100	0	100	0 %
10	<a href="#">6층노트북열람실</a>	72	45	27	62.5 %
11	<a href="#">1층노트북열람실</a>	46	36	10	78.26 %
12	-	1	0	1	0 %

※ 본 페이지는 1분에 한 번씩 자동으로 갱신됩니다.

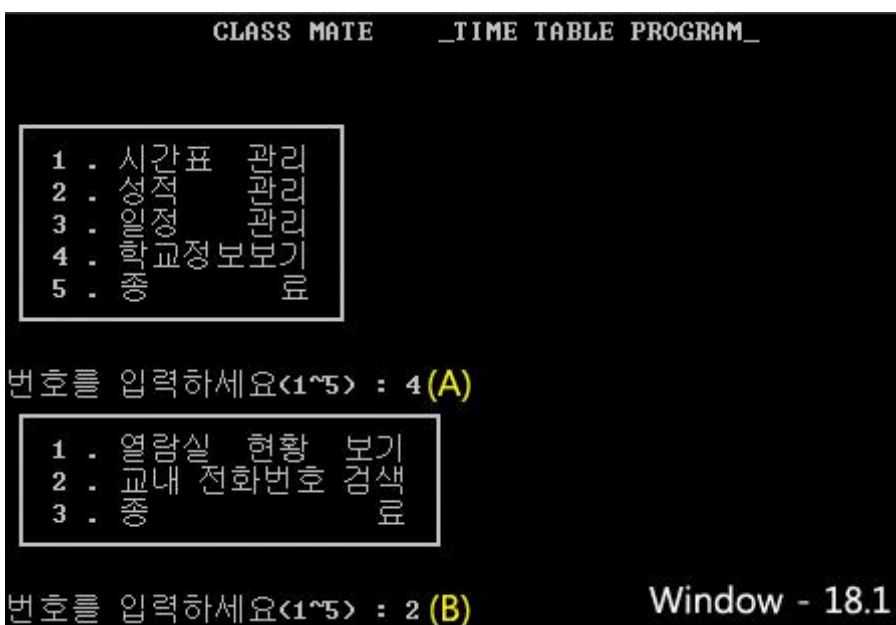
Copyright © 2006 건국대학교 상허도서관. All rights reserved.

Window - 17.2

11\_2: 열람실 정보보기\_2 UI

Use Case	도서관 열람실 현황 보기
Actor	사용자
Purpose	사용자가 현재 도서관 열람실 상황 조회
Overview	사용자가 현재 도서관 열람실 정보를 조회하고자 할 때 학교 홈페이지 내의 도서관 상황 웹페이지를 링크 오픈한다.
Type	Primary
Cross Reference	Classes : FileIO, Identify, Link Functions : InFile_Open(), InFile_Close(), Input_ID(), Check_Password() Show_Library() Usecase : 사용자인증
Pre-Requisites	사용자의 컴퓨터가 인터넷 연결이 되어있어야 한다.
UI Widgets	Window-17.1 , Window-17.2
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 인증단계를 거친 사용자가 '학교정보 보기' 메뉴(A)를 선택 후 '열람실 현황 보기' 서브메뉴(B)를 선택하면 시작된다. (E2) 2. (S) 시스템이 Explorer을 실행하여 도서관 현황 웹페이지인 <a href="http://203.252.130.231/domian5.asp">http://203.252.130.231/domian5.asp</a> 를 링크 오픈한다. (Window-17.2와 같은 웹페이지가 오픈된다.) 3. (A) 사용자는 도서관 열람실 상황을 볼 수 있다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E2

### 3.12 교내 전화번호 검색



12\_1: 교내 전화번호 검색\_1 UI



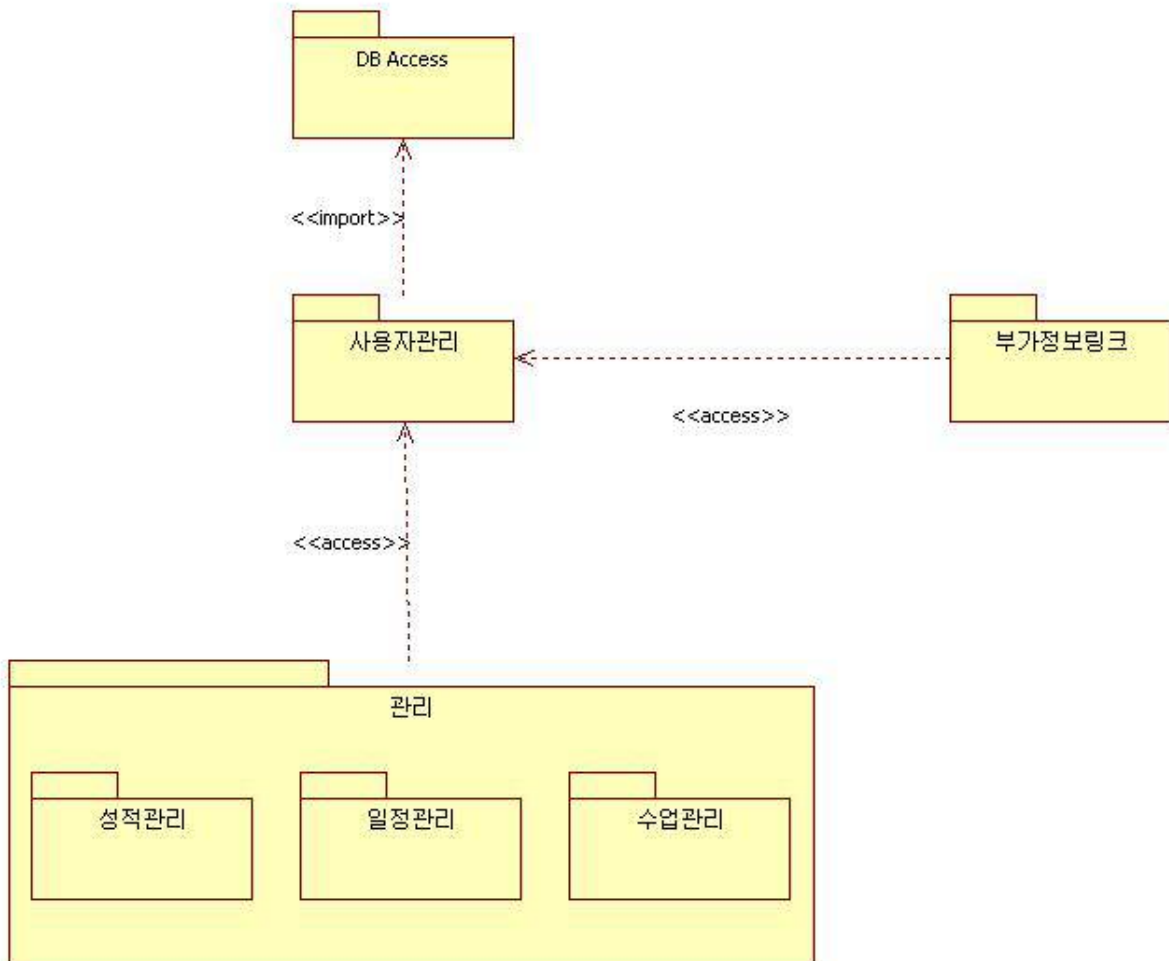
12\_2: 교내 전화번호 검색\_2 UI

Use Case	교내 전화번호 검색
Actor	사용자
Purpose	사용자가 교내 시설 전화번호 검색
Overview	사용자가 교내 시설 전화번호를 알고자 할 경우 학교 홈페이지의 교내 전화번호 안내/검색 페이지를 링크오픈 한다.
Type	Primary
Cross Reference	Classes : FileIO, Identify, Link Functions : InFile_Open(), InFile_Close(), Input_ID(), Check_Password() Show_Facilities() Usecase : 사용자인증
Pre-Requisites	사용자의 컴퓨터가 인터넷 연결이 되어있어야 한다.
UI Widgets	Window-18.1, Window-18.2
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) 이 use case는 인증단계를 거친 사용자가 '학교정보 보기' 메뉴(A)를 선택 후 '교내 전화번호 검색' 서브메뉴(B)를 선택하면 시작된다. (E2) 2. (S) 시스템이 Explorer을 실행하여 학교 홈페이지 내에서 교내 시설 전화번호 검색이 가능한 <a href="http://www.konkuk.ac.kr/do/Intro/TelNmMng.do">http://www.konkuk.ac.kr/do/Intro/TelNmMng.do</a> 를 링크 오픈한다. (Window-17.2와 같은 웹페이지가 오픈된다.)

	3. (A) 사용자는 검색어를 입력하여(Window-18.2 A) 전화번호를 검색 할 수 있다. 필요시, (Window-18.2 B)를 클릭하여 카테고리 선택 할 수 있다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E2

## 4. Software architecture description

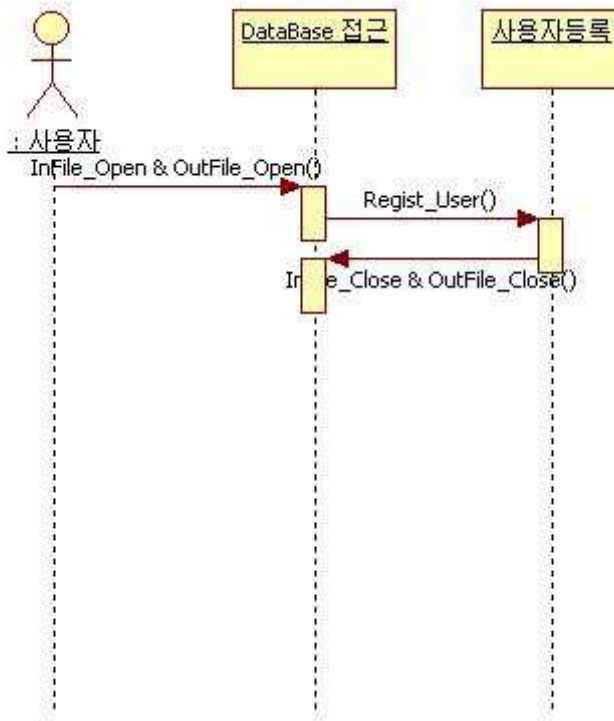
- Package diagram



## 5. Interaction diagram

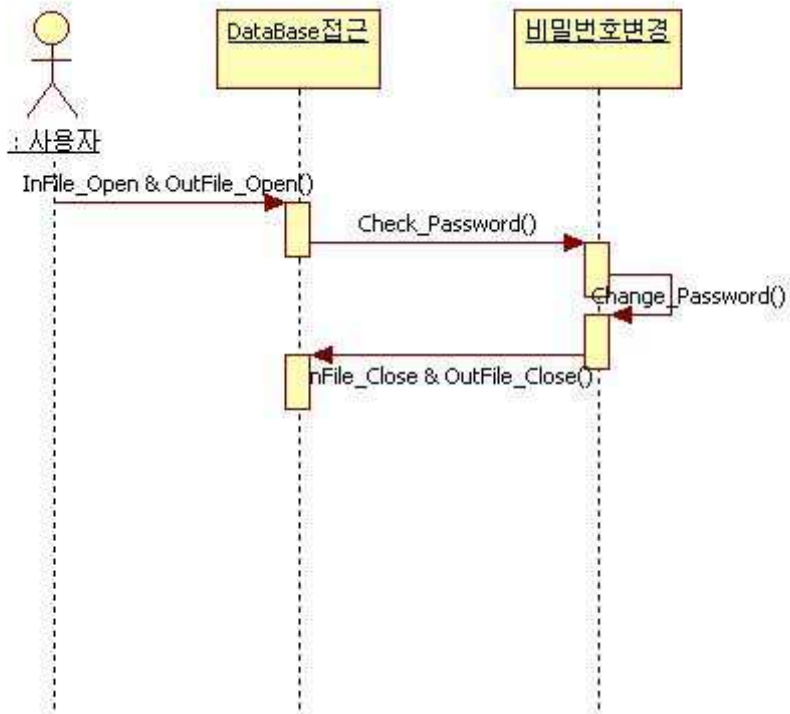
- Sequence diagram

### 5.1 사용자 등록



14: 사용자 등록 Sequence diagram

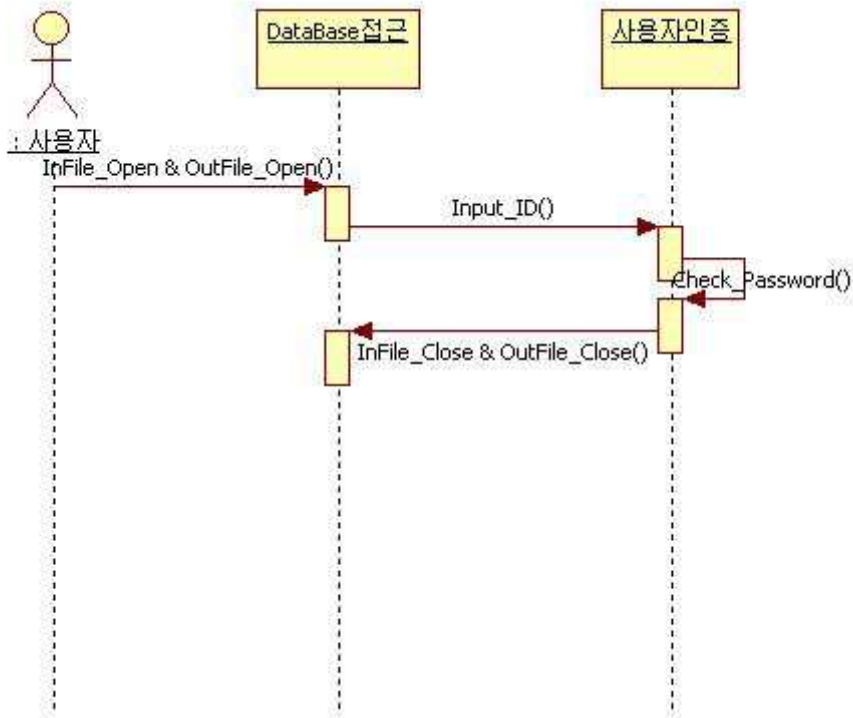
### 5.2 비밀번호 변경



15: 비밀번호 변경 Sequence diagram

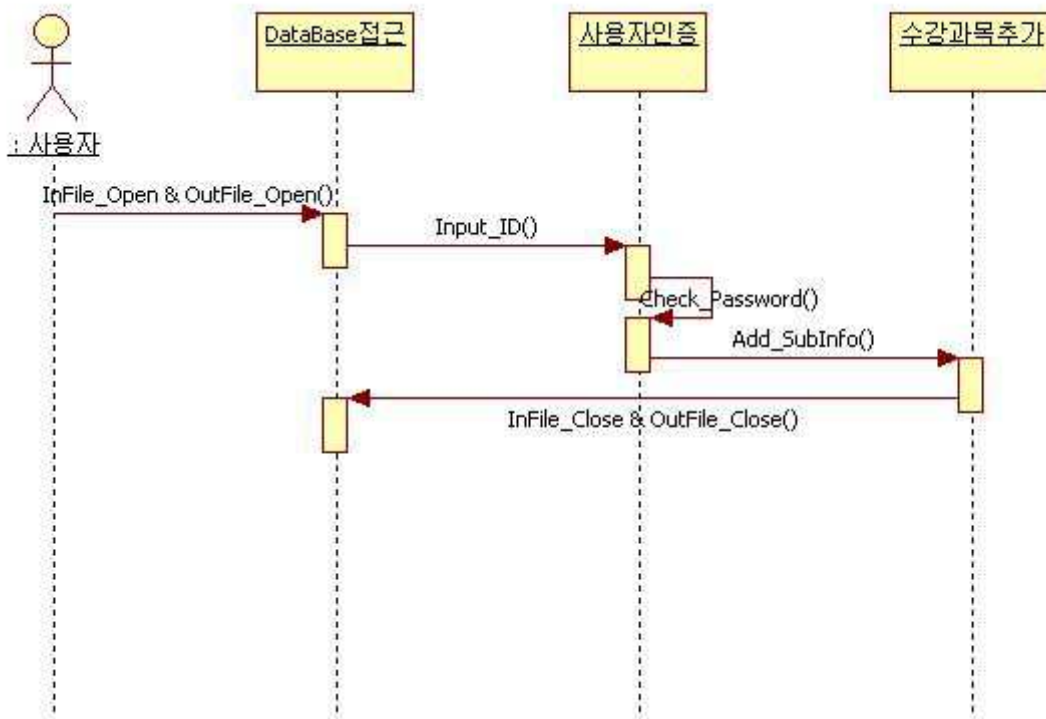


### 5.3 사용자 인증



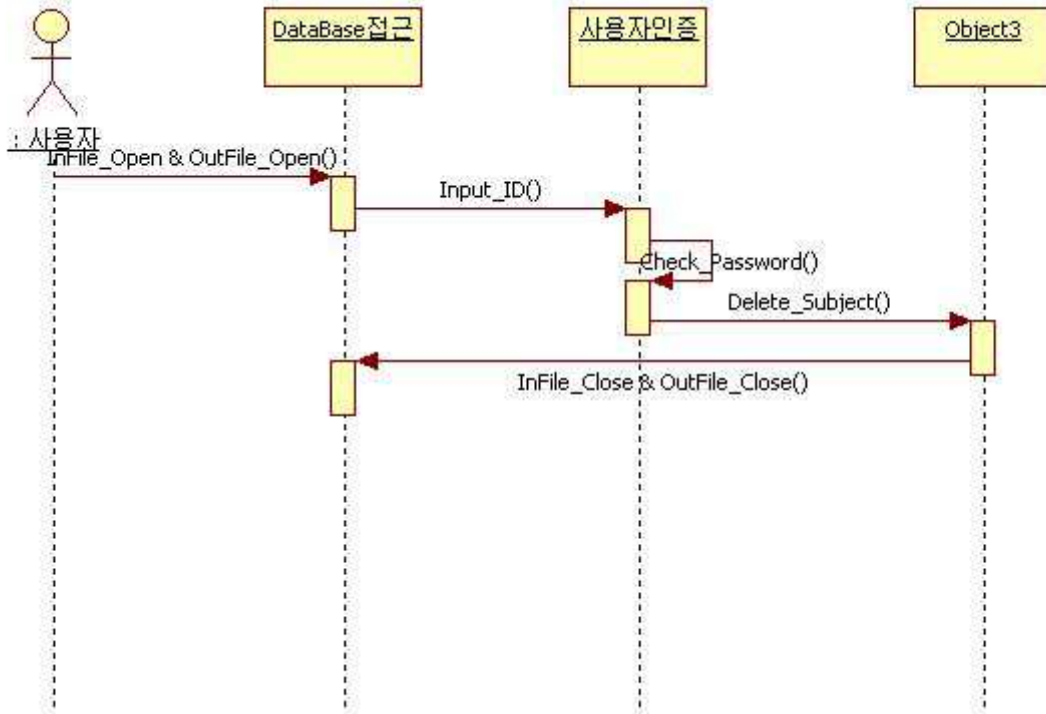
16: 사용자 인증 Sequence diagram

### 5.4 수강과목 추가



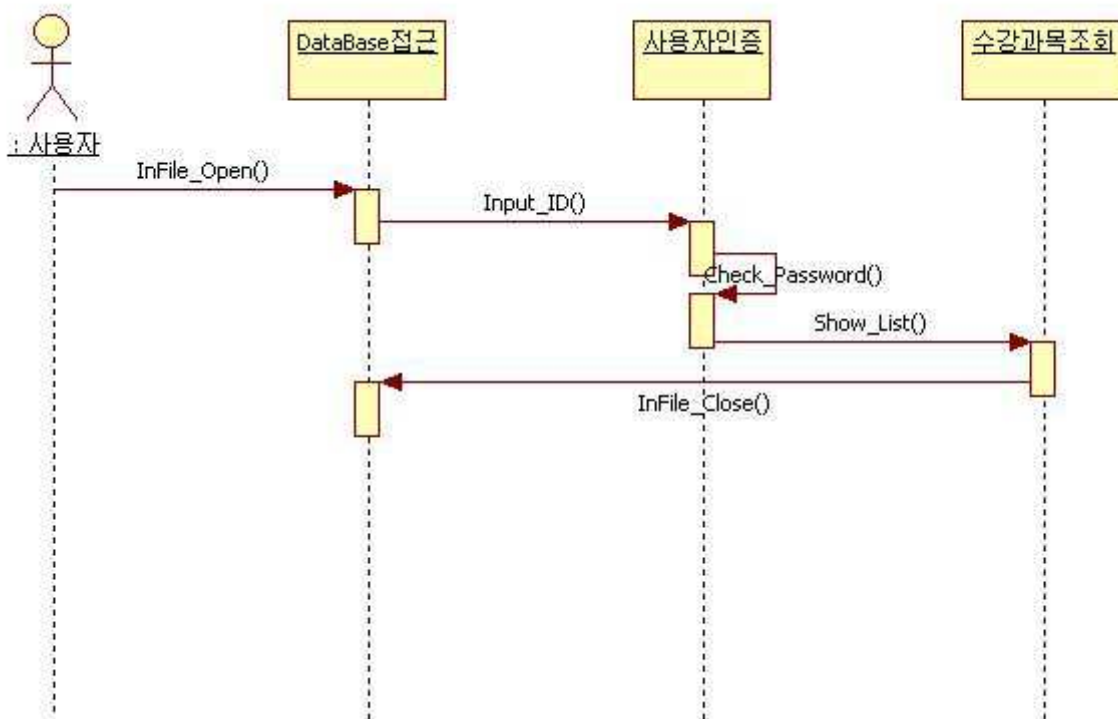
17: 수강과목 추가 Sequence diagram

### 5.5 수강과목 삭제



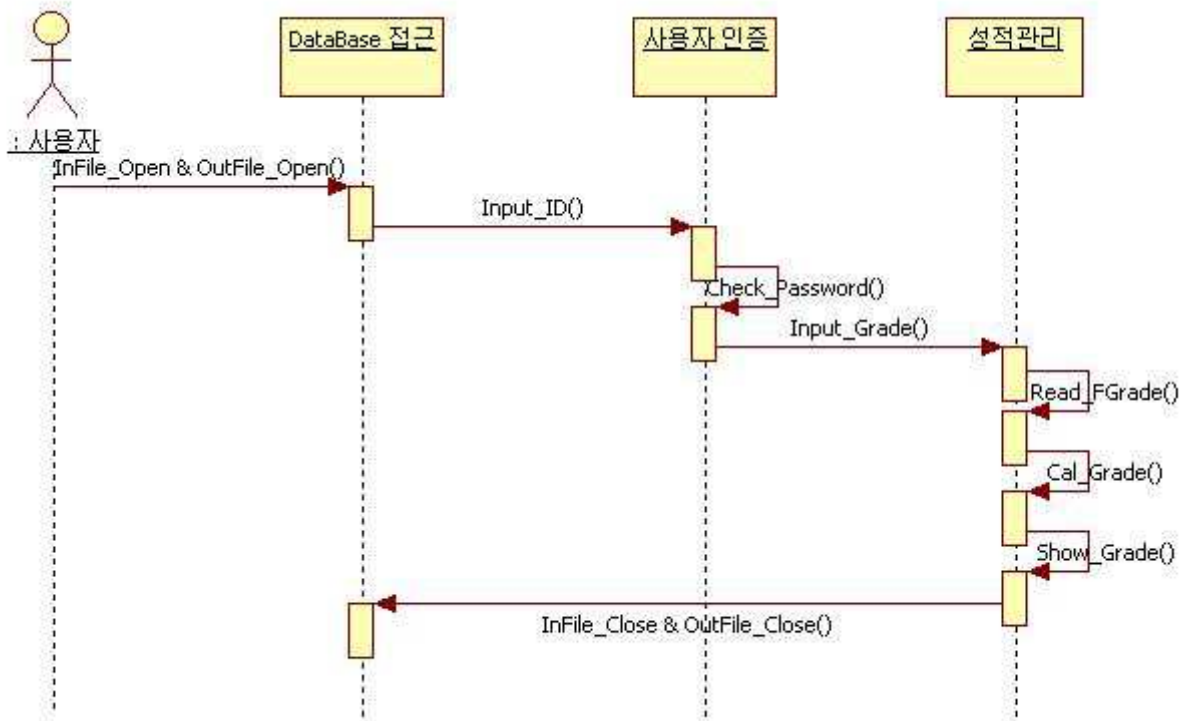
18: 수강과목 삭제 Sequence diagram

### 5.6 수강과목 조회



19: 수강과목 조회 Sequence diagram

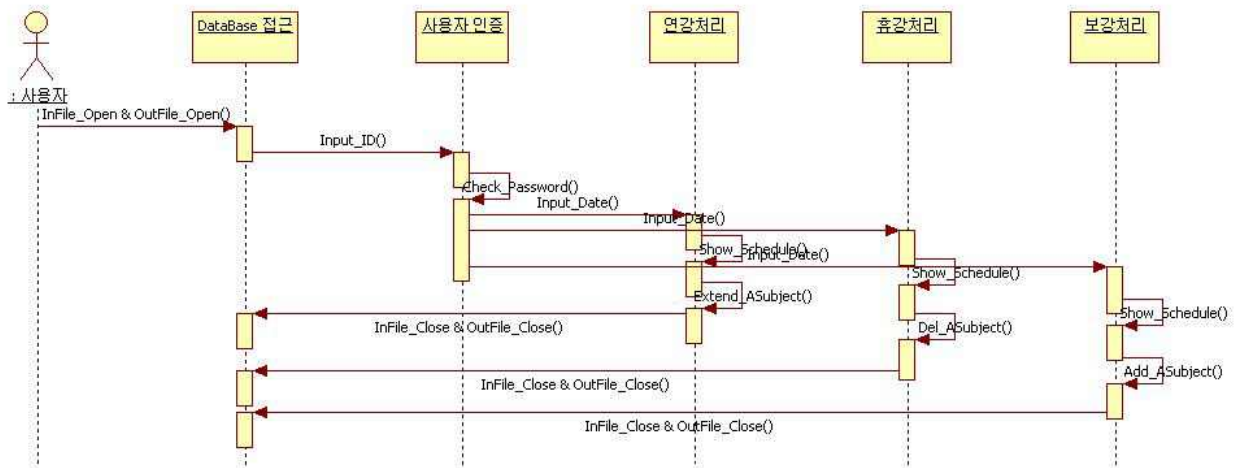
## 5.7 성적관리



20: 성적관리 Sequence diagram

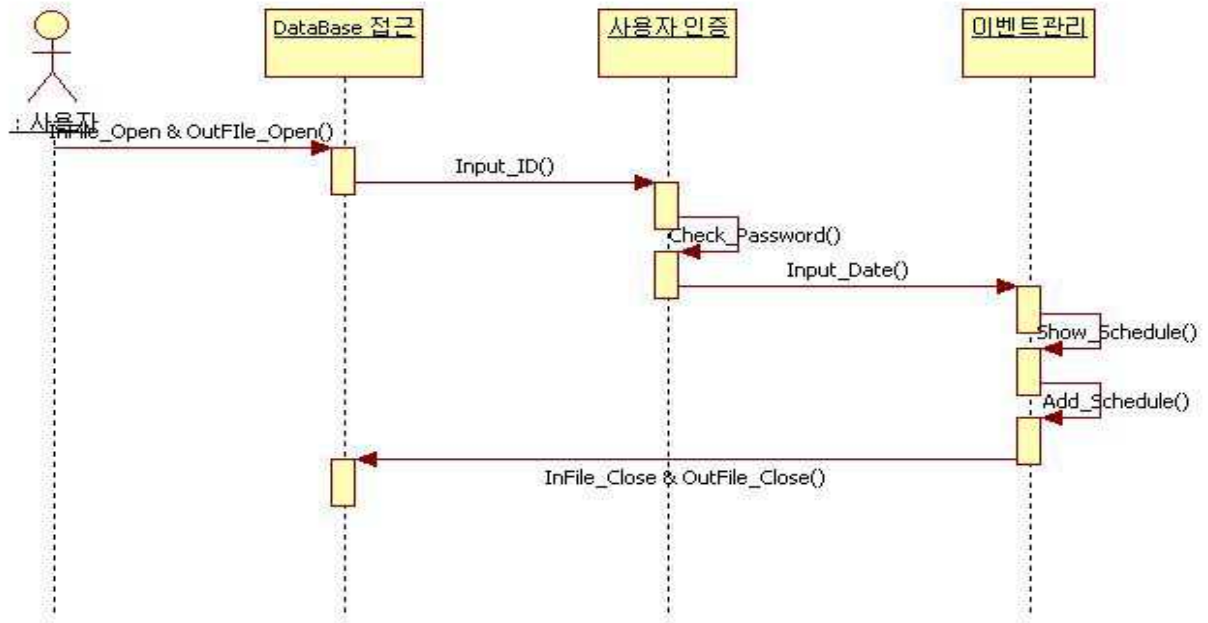
## 5.8 일정관리

### 5.8.1 휴강/보강/연강등록



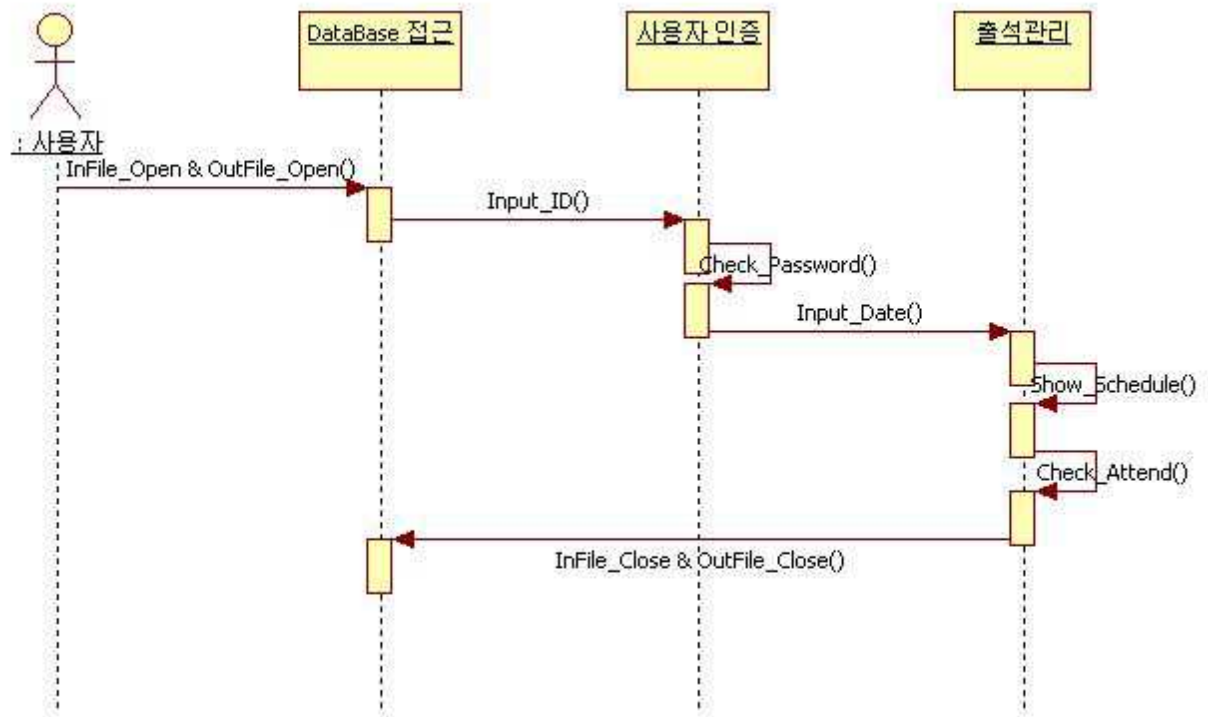
21\_1: 일정관리\_1 Sequence diagram

### 5.8.2 이벤트 등록



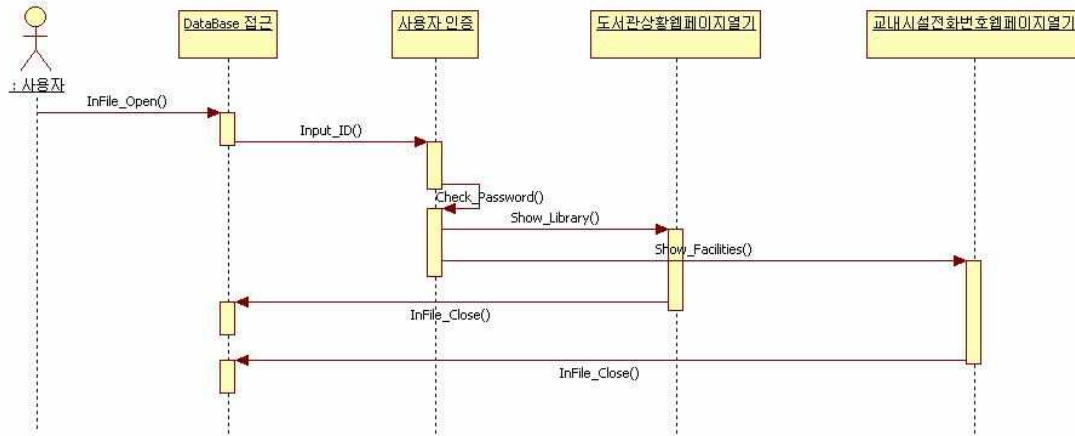
21\_2: 일정관리\_2 Sequence diagram

### 5.8.3 출석관리



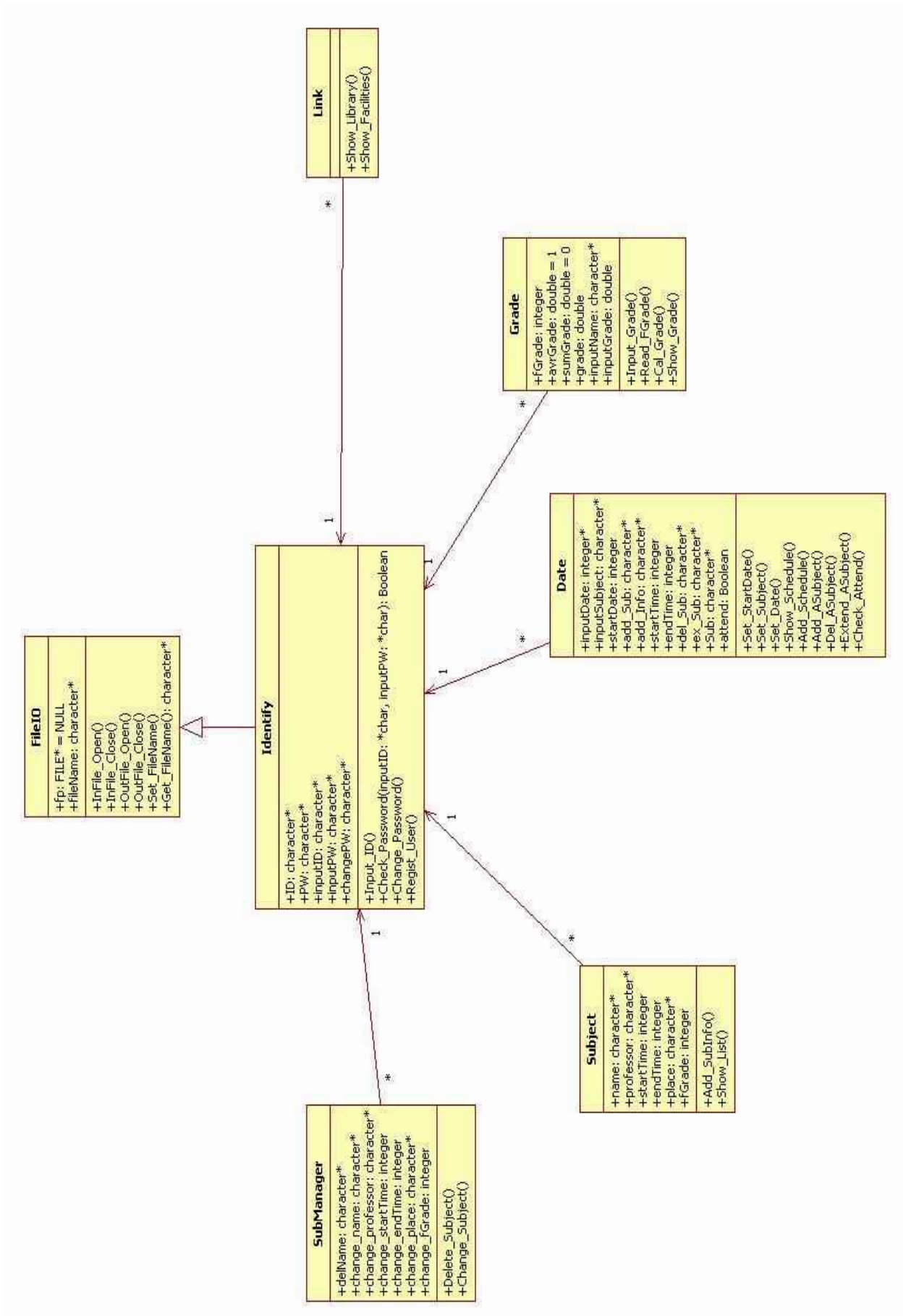
21\_3: 일정관리\_3 Sequence diagram

### 5.9 웹페이지 링크(도서관 열람실 현황 조회/ 교내시설 전화번호 안내, 검색)



22: 웹페이지 링크 Sequence diagram

## 6. A Design Class Diagram



23: Class Diagram

## 7. Database design description

### Database schema

#### 7.1 Class 변수 및 메소드

##### 7.1.1 class Identify

```
char *ID; // 파일에서 읽어온 ID Buffer
char *PW; // 파일에서 읽어온 PW Buffer

void Input_ID()
    char *inputID;
    char *inputPW; // 사용자가 입력한 비밀번호
bool Check_Password(char *inputID, char *inputPW)
void Change_Password()
    char *inputID;
    char *inputPW;
    char *changePW; // 변경할 비밀번호
void Regist_User()
    char *inputID;
    char *inputPW;
```

##### 7.1.2 class FileIO

```
FILE *fp; // 파일 포인터
char *FileName; // 파일이름 (저장 / 반환)

void InFile_Open()
void InFile_Close()
void OutFile_Open()
void OutFile_Close()
void Set_FileName()
char *Get_FileName()
```

##### 7.1.3 class SubManager

```
void Delete_Subject()
    char *delName; // 지우고자 하는 과목의 과목명
void Change_Subject()
    char *change_name; // 정보변경하고자 하는 과목의 과목명
    // 바꾸고자하는 세부정보
    char *change_professor; // 담당교수명
    int change_startTime; // 강의가 시작하는 시간
    int change_endTime; // 강의가 끝나는 시간
    char *change_place; // 강의실
    int change_fGrade; // 학점단위
```

#### 7.1.4 class Subject

```
void Add_SubInfo()
    char *name;           // 과목명
    char *professor; // 담당교수명
    int startTime;       // 강의가 시작하는 시간
    int endTime;        // 강의가 끝나는 시간
    char *place;        // 강의실
    int fGrade;         // 학점단위
void Show_List()
```

#### 7.1.5 class Grade

```
int fGrade;           // 학점단위
double avrGrade;     // 학점평균 (계산값)
char *inputName;     // 사용자가 입력한 과목명

void Input_Grade()
    double inputGrade; // 성적으로 받은 학점
void Read_FGrade()
void Cal_Grade()
void Show_Grade()
```

#### 7.1.6 class Date

```
int *inputDate;     // 사용자가 조회하거나 변경할 날로 입력한 날짜
char *inputSubject; // " " " " " " 과목명

void Set_StartDate()
    int startDate; // 한 학기가 시작되는 날짜
void Set_Date()
void Set_Subject()
void Show_Schedule()
void Add_Schedule()
    char *add_Sub; // 정보가 추가될 과목의 과목명
    char *add_Info; // 추가할 정보
void Add_ASubject()
    char *add_Sub; // 보강하는 과목의 과목명
    int startTime; // 보강이 시작하는 시간
    int endTime; // 보강이 끝나는 시간
void Del_ASubject()
    char *del_Sub; // 휴강되는 과목의 과목명
void Extend_ASubject()
    char *ex_Sub; // 연장되는 과목의 과목명
    int endTime; // 연장이 끝나는 시간
void Check_Attend()
    char *Sub; // 과목명
    bool attend; // 출석여부
```



### 7.1.7 class Link

void Show_Library()	//도서관 상황 웹페이지 링크오픈
void Show_Facilities	//교내시설 전화번호 웹페이지 링크오픈

## 7.2 Database schema

### 7.2.1 Identify 클래스

- 프로그램 시작 시, 비밀번호를 입력/확인/변경하는 보안관련 클래스

Key name	Data type	NULL	Default value	Key information
ID	char *	X	DB에서 Load	파일에서 읽어온 ID
PW	char *	X	DB에서 Load	파일에서 읽어온 비밀번호
inputID	char *	X		사용자가 입력한 ID
inputPW	char *	X		사용자가 입력한 비밀번호
changePW	char *	X		바뀔 비밀번호

### 7.2.2 FileIO 클래스

- Database를 여닫고, Database의 이름을 입력받는 클래스

Key name	Data type	NULL	Default value	Key information
FileName	char *	X	DB에서 Load	Database 텍스트파일의 이름
fp	FILE *	O	NULL	파일 포인터

### 7.2.3 SubManager 클래스

- 과목을 삭제하거나 그 안의 정보를 변경하는 클래스

Key name	Data type	NULL	Default value	Key information
delName	char *	X		사용자가 지우고자하는 과목명
change_name	char *	X		과목정보를 변경할 과목의 이름
change_professor	char *	X		변경될 담당교수이름
change_startTime	int	X		변경될 강의 시작시간
change_endTime	int	X		변경될 강의 종료시간
change_place	char *	X		변경될 강의실
change_fGrade	int	X		변경될 학점단위

#### 7.2.4 Subject 클래스

- 과목의 입력/출력을 하는 클래스

Key name	Data type	NULL	Default value	Key information
name	char *	X		과목의 이름
professor	char *	X		담당교수이름
startTime	int	X		강의 시작시간
endTime	int	X		강의 종료시간
place	char *	X		강의실
fGrade	int	X		학점단위

#### 7.2.5 Grade 클래스

- 학점 입/출력, 평점 계산/출력을 하는 클래스

Key name	Data type	NULL	Default value	Key information
fGrade	int	X		학점단위
avrGrade	double	X	1	평점
sumGrade	double	X	0	학점의 합
grade	double	X		학점
inputName	char *	X		사용자가 입력한 과목명
inputGrade	double	X		사용자가 입력한 학점

#### 7.2.6 Date 클래스

- 날짜별로 과목의 정보를 열람하고 수정/추가/관리를 하는 클래스

Key name	Data type	NULL	Default value	Key information
inputDate	int	X		사용자가 입력한 날짜
inputDate	char *	X		사용자가 입력한 과목명
startDate	int	X		학 학기의 시작날짜
add_Sub	char *	X		정보를 추가할 과목명
add_Info	char *	X		추가할 정보
startTime	int	X		보강 시작시간
endTime	int	X		보강 및 연강 종료시간
del_Sub	char *	X		휴강 과목명
ex_Sub	char *	X		연강 과목명
Sub	char *	X		출석체크할 과목명
attend	bool	X	true / false	출석여부

#### 7.2.7 Link 클래스

- 웹페이지를 링크하는 클래스(변수 존재X)

Key name	Data type	NULL	Default value	Key information
X	X	X		X