

# 『Perfect』Time Table

(For Organization) SRS

- 학사관리 시스템 -

Team 8

200611458 김영승

200611478 성두훈

200611494 원스타

200611518 조민경

# 1. Introduction

## 1.1 Purpose

- 이 문서는 일반적인 Time Table의 의미를 넘어선, 학생, 교수, 수업관리자, 학사관리자로 나누어진 완전한 형태의 학사관리 시스템을 개발하기 위한 목적으로 작성되었다.
- 이 문서는 개발자와 사용자를 위해 작성 되었다.

## 1.2 Scope

- 이 시스템은 일반적인 Time Table의 기능을 4분류의 사용자 계층의 목적에 맞는 "Perfect(완전한)" Time Table이다.
- 이 시스템은 다양한 사용자로 구분되어 있고, 그 사용자에 따라 권한이 나눠져 있기 때문에 목적에 맞는 명확한 프로그램의 사용이 가능하다.
- 이 시스템은 개인용 Time Table이 아닌 단체의 사용에 그 목적이 있다.

## 1.3 Definitions, acronyms, and abbreviations

- 강의, 수업 : 같은 의미로, 사용되었으며 서로 다른 점은 존재하지 않는다.

## 1.4 References

- IEEE Std. 830-1998
- 이 문서는 IEEE 표준 830-1998을 따라서 작성되었다.

## 1.5 Overview

- 2. Overall description : 시스템이나 시스템의 요구사항에 영향을 미칠 수 있는 일반적인 사항들에 대해 기술한다. 상세한 내용은 section 3를 통해 이해하기 쉽도록 작성할 것이고, 여기서는 전반적인 내용을 다룬다.
  - 2.1 Product perspective
  - 2.2 Product functions
  - 2.3 User functions
  - 2.4 Constraints
  - 2.5 Assumptions and dependencies
- 3. Specific requirements : 이 section으로 디자인이 가능하고, 테스트가 가능할 수 있을 만큼 모든 시스템 요구사항에 관한 자세한 내용을 기술한다. Section 2와는 다르게 좀 더 구체적인 입력과 출력에 대한 내용을 기술한다.
  - 3.1 External interfaces
  - 3.2 Function requirements
  - 3.3 Performance requirements
  - 3.4 Logical database requirements

- 3.5 Design constraints
- 3.6 Software system attributes
- 3.7 Organizing the specific requirements
  
- 4. Supporting information : 본 문서를 읽고 사용하는데 유용한 정보를 첨부한다.
- 4.1 Tables of contents and index
- 4.2 Appendixes

## 2. Overall description

### 2.1 Product perspective

#### 2.1.1 System interfaces

- 본 시스템은 .xml 형식으로 정보를 저장하고, 읽어오는 방식을 취한다.
- 파일의 저장과 읽어오기는 실시간 방식이 아닌, 정적 방식으로 실행 시에는 시스템 버퍼를 이용하여 정보를 지니고, 프로그램 종료 시에 한번에 저장하는 방식을 취한다.

#### 2.1.2 User interfaces

- 본 시스템은 사용자에게 모니터, 키보드, 마우스를 통한 GUI를 제공한다.
- 로그인 화면은 동일하되, 사용자 별로 제한된 사항에 관해서는 활성화 방식과 화면 구성의 차이를 두어 구별한다.
- 화면의 구성은 일반적인 사용자에게 가장 익숙한 윈도우 방식을 따르며, 기존의 일반적인 학사관리 방식을 따른다.

#### 2.1.3 Hardware interfaces

- 본 시스템은 외부 하드웨어로 프린터를 사용한 시간표 출력 방식을 사용하여, 사용자가 시간표를 시스템 외부에서도 볼 수 있는 기능을 제공한다.

#### 2.1.4 Software interfaces

- 프린트를 하기 위한 프린터 장치 소프트웨어와 연관이 있다.

#### 2.1.5 Communications interfaces

- 향후, Evolution 단계에서 네트워크를 이용한 시스템으로 발전 가능성이 있으나, 본 시스템은 네트워크 기능을 사용하지 않는다.

#### 2.1.6 Memory constraints

- 실시간 저장 방식이 아닌 시스템 버퍼를 사용하였다 한번에 저장하는 방식으로 실시간 업데이트가 안 되는 제약사항이 존재한다.

#### 2.1.7 Operations

- 4분류로 나누어진 각각의 사용자에게 따라 각각 다르게 운영되도록 한다.
- 학사관리자 : 계정의 승인, 생성을 해준다. 그 외의 학생, 교수, 수업관리자의 개인 정보를 관리 할 수 있다.
- 수업관리자 : 강의 등록, 승인 및 강의실 관리, 강좌 정보 관리, 시간표 관리 등 수업에 관한 전반적인 내용을 관리한다.
- 교수 : 자신의 개인정보 관리, 강의 등록 신청, 자신의 강좌 관리를 할 수 있다.

- 학생 : 자신의 개인정보 관리와 수강정보 관리를 할 수 있다.

## 2.2 Product functions

- a. ID신청 : 사용자 별로 ID를 신청할 수 있다.
- b. 로그인 : 사용자 별 고유 ID로 접속할 수 있다.
- c. 개인정보관리 : (학생, 교수)자신의 개인정보를 수정할 수 있다.
- d. 공지사항 확인 : 공지사항을 확인할 수 있다.
- e. 학생
  - e.1 종합강의 시간표 조회 : 현재 등록되어 있는 수업들을 열람할 수 있다.
  - e.2 수강신청 : 종합강의 시간표상에 등록되어 있는 수업을 신청할 수 있다.
  - e.3 개인 시간표 조회 : 신청한 강의를 확인할 수 있다.
  - e.4 개인 시간표 인쇄 : 신청한 강의를 프린터를 통해 인쇄할 수 있다.
  - e.5 성적조회 : 자신의 강의 성적을 조회할 수 있다.
  - e.6 강의평가 : 자신이 수강한 과목의 강의를 평가할 수 있다.
- f. 교수
  - f.1 강의 등록 신청 : 자신의 강의를 등록할 수 있도록 신청할 수 있다.
  - f.2 강좌 시간표 조회 : 자신의 강의를 열람할 수 있다.]
  - f.3 각 과목별 출석부 조회 : 자신의 강의를 수강하는 학생들의 출석부를 확인할 수 있다.
  - f.4 성적처리 : 자신의 강의를 수강하는 학생들의 성적을 입력할 수 있다.
- g. 수업 관리자
  - g.1 수업생성 : 강의를 생성할 수 있다.
  - g.2 강의승인 : 교수가 신청한 강의 목록을 확인하여 승인할 수 있다.
  - g.3 강의실 관리 : 강의를 진행하는 강의실을 관리할 수 있다.
  - g.4 시간표 관리 : 전체 진행되는 강의에 관한 시간을 관리할 수 있다.
- h. 학사 관리자
  - h.1 계정 생성 및 승인 : 학생 및 교수의 계정을 생성 및 승인 할 수 있다.
  - h.2 개인정보관리 : 자신이 아닌 학생, 교수, 직원의 개인정보를 관리할 수 있다.
- i. 수업관리자와 학사관리자는 공지사항을 수정할 수 있다.
- j. 학생의 수강신청 시 시간이 중복되는 강의는 수강이 불가능하다.
- k. 강의 등록 시 동시간 대에 같은 강의실의 사용은 불가능하다.

## 2.3 User functions

- 본 시스템은 일반 대학의 사용을 기본으로 한다.
- 윈도우 기반의 컴퓨터를 사용할 수 있는 학생, 교수를 일반적인 대상으로 하며, 수업관리자와 학사관리자의 경우 사용자보다는 운영자의 입장으로 시스템에 대한 전반적인 이해가 필요하다.
- 시스템의 확장성을 고려했을 때, 일반 대학뿐 아니라, 사설 교육기관에서도 사용이 가능할 것이다.

## 2.4 Constraints

- 개발환경과 개발도구의 한계로 윈도우 사용을 기반으로 한다.
- 시스템의 한계로 네트워크의 사용이 불가능하다.
- 실시간 방식이 아니기 때문에 다른 사용자끼리의 즉각적인 정보 교환이 불가능하다.
- 개발 기간이 한정적이기 때문에 기본적인 기능만을 구현한다.
- 사용자의 편의를 위하여 GUI 기반 방식을 택해야 한다.
- 분석과 설계 방식은 객체 지향 방식으로 진행하며, UML tool을 반드시 사용한다.
- .NET Framework 3.5가 설치된 윈도우 환경에서 실행 가능하다.

## 2.5 Assumptions and dependencies

- 데이터 베이스파일이 초기화 될 경우, 기존의 정보가 삭제되므로 파일의 관리가 필요하다.
- 네트워크를 사용하지 않으므로, 한대의 컴퓨터에서 작업을 진행해야 한다.
- 학사관리자는 최고 운영자의 위치로 시스템 작동 시 임의로 지정된 ID와 password를 사용한다.

## 3. Specific requirements

### 3.1 External interfaces

3.1.1 사용자 별로 각기 다른 인터페이스를 제공해야 한다.

3.1.2 사용자 모두에게 첫 화면으로 공지사항을 보여준다.

3.1.3 개인정보 변경을 통해 수정 가능한 항목만을 수정할 수 있도록 입력 사항을 제한한다.

#### 3.1.4 학생

- a. 학생에게 등록된 수업 리스트와 시간표를 별도로 제공한다.
- b. 학생에게 강의 조회 시 과목명, 과목번호, 학과별, 담당교수로 검색할 수 있도록 한다.
- c. 학생이 수강신청을 할 시에 팝업을 통해 결과를 확인시켜 준다.
- d. 학생의 시간표를 보기를 실행할 시에 프린트버튼을 따로 띄워준다.
- e. 학생의 성적은 학점과 등급을 출력해주며, 설정이 되지 않았을 경우 설정이 되지 않았음을 알려준다.
- f. 신청학점, 취득학점, 평균 평점을 계산하여 보여준다.
- g. 학생이 강의 평가할 시에는 과목별로 강의 평가 창을 새로 띄워주도록 한다.

#### 3.1.5 교수

- a. 강의 중인 수업은 목록화 하여 보여준다.
- b. 목록을 선택하여 출석부를 확인할 수 있다.
- c. 강의 등록 신청은 대기중인 리스트를 보여주어, 처리상태를 알 수 있도록 한다.
- d. 성적처리 시 각 강의 별 학생들 리스트를 별도로 보여주어, 성적을 입력하도록 한다.

#### 3.1.6 수업관리자

- a. 강의조회를 통하여 현재 진행중인 강의 정보를 확인한다.

- b. 강의 승인을 통하여 승인 대기 중인 강의 정보를 확인하여 승인할 수 있도록 한다.
- c. 강의가 끝난 과목에 관하여 폐강을 할 수 있다.
- d. 공지사항 관리를 통하여 기존의 내용을 삭제시키거나 신규 내용을 입력한다.

#### 3.1.7 학사관리자

- a. 인원관리를 통하여 가입 승인 대기 중인 인원의 목록을 보여준다.
- b. 가입거부와 승인을 할 수 있도록 한다.
- c. 공지사항 관리를 통하여 기존의 내용을 삭제시키거나 신규 내용을 입력한다.

3.1.8 종료는 실행 프로그램의 'X'버튼을 클릭하여 실행한다.

## 3.2 Functions

### 3.2.1 프로그램 실행

- a. 실행파일의 더블클릭으로 실행한다.
- b. 로그인 화면이 실행되며, 아이디와 비밀번호를 입력 받는다.
- c. 로그인 아이디가 있을 경우, 로그인 후 데이터를 읽어오고, 로그인 아이디가 없을 경우, 메시지를 출력한다.

### 3.2.2 개인정보

- a. 변경 내용을 입력하고, 변경 적용을 클릭하여야 내용이 저장된다.
- b. 비밀번호는 두 번 입력을 받고, 암호화하여 직접 볼 수 없도록 한다.
- c. 비밀번호가 다를 시에는 메시지를 출력하여 동일할 시에만 적용한다.

### 3.2.3 학생 -> 수업관리

- a. 강의 추가를 통해 종합강의시간표를 조회할 수 있도록 하고, 과목 추가 시에 기존 강의와의 중복여부, 시간 등의 오류 사항을 파악하여, 수강 가능할 경우만 진행한다.
- b. 시간표 보기를 통하여 요일 별, 시간 별 표로 보여주고, 프린트가 가능하도록 프린터와 연동시킨다.
- c. 수강한 과목은 제거를 통해 지울 수 있다.

### 3.2.4 학생 -> 성적 및 강의 평가

- a. 성적 란을 통하여 수강중인 강의의 학점과 등급을 보여주며, 하단에 신청학점과, 취득학점, 평점평균을 계산하여 보여준다.
- b. 강의 평가 란을 통하여 수강한 강의의 평가를 하고, 평가는 점수를 통해 하도록 한다. 이 결과는 각 교수의 강의 평가 란에 보여진다.

### 3.2.5 교수 -> 담당 강의관리

- a. 수업을 통하여 현재 진행중인 강의의 정보를 확인할 수 있으며, 출석부를 확인할 수 있다.
- b. 목록에서 강의를 선택하고 출석부를 선택하면 출석부가 화면에 출력된다.
- c. 출석부는 외부 프린터와 연동하여 출력할 수 있도록 한다.
- e. 성적 처리는 별도의 성적 처리 란을 통하여 진행한다.
- f. 성적처리 시 과목별로 선택하면 분할된 창에 학생의 정보가 출력되고, 출력된 상태에서 곧바로 성적 처리가 가능하다.
- g. 성적은 등급으로 매기며, 등급은 현 대학의 시스템과 동일하다.

### 3.2.6 교수 -> 강의등록 신청

- a. 강의 등록 신청은 별도의 항목에서 진행하며, 이 항목에서 현재 신청중인 강의 목록을 확인할 수 있다.
- b. 강의 등록은 강의 이름, 학점, 최대인원, 전공분류, 요일, 시작교시, 끝 교시, 강의실 등의 정보를 입력하도록 한다.
- c. 잘못된 정보나 기존 강의와 중복된 정보에 관해서는 오류 메시지를 출력하도록 한다.

### 3.2.7 수업관리자

- a. 강의 조회를 통해 현재 진행중인 강의의 상태를 확인할 수 있다.
- b. 강의 승인은 목록에서 한 과목 한 과목 선택하여 승인하도록 하며, 삭제를 통해 승인을 거부할 수도 있다.
- c. 폐강을 통해 종료된 강의에 관해서는 폐지처리를 할 수 있다.

### 3.2.8 학사관리자

- a. 학사관리자는 시스템 생성시에 존재하는 아이디와 비밀번호를 통해 접속한다.
- b. 인원관리를 통하여 현재 회원가입을 대기중인 인원을 확인할 수 있고, 인원을 선별하여 받아들일 수 있다.

### 3.2.9 공지사항

- a. 공지사항은 모두가 볼 수 있지만, 수정은 학사관리자와 수업관리자만이 가능하다.
- b. 공지사항은 기존의 내용을 지우는 것과 신규 내용을 추가하는 것이 가능하다.

## 3.3 Performance requirements

- 3.3.1 4분류의 사용자가 사용한다. 엄밀히 말하여 1명의 최고 운영자, 1명의 준 운영자, 2명의 유저를 대상으로 한다.
- 3.3.2 생성할 수 있는 수업의 개수는 20개로 제한한다.
- 3.3.3 신청학점은 18학점으로 제한한다.
- 3.3.4 학생 계정의 생성은 100개로 제한한다.
- 3.3.5 교수 계정의 생성은 5개로 제한한다.
- 3.3.6 수업관리자 계정의 생성은 2개로 제한한다.

## 3.4 Logical database requirements

- 3.4.1 지정된 위치에 있어야 하며, .xml파일은 누구나 볼 수 있으므로, 별도의 암호화된 폴더에 보관하도록 한다.
- 3.4.2 데이터 베이스파일은 학사관리자가 관리해야 하며, 이 파일을 통하여 개인정보를 수정할 수 있다.

### 3.5 Design constraints

3.5.1 객체 지향 설계 방식을 사용하여 진행해야 한다.

3.5.2 분석단계를 완료하여 UML tool을 사용하여 클래스다이어그램을 완성하고, 기본 클래스를 code generation을 통하여 설계한다.

### 3.6 Software system attributes

#### 3.6.1. Reliability

3.6.1.1 학사관리자에 의한 개인정보 유출을 제외한 그 어떤 경우에도, 개인정보는 유출되어서는 안 된다.

3.6.1.2 권한을 넘어서는 기능을 제공해서는 안 된다.(ex : 학생의 학점관리, 교수의 무단 수업 생성 등등)

3.6.1.3 중복되는 강의를 대한 엄격한 제한이 있어야 한다.

3.6.1.4 비정상적인 종료는 없어야 한다.

#### 3.6.2 Availability

3.6.2.1 프린터와의 연동이 잘되어야 한다.

3.6.2.2 모든 기능을 구현해야 한다.

3.6.2.3 모든 기능은 사용자 선택 시 2초 이내에 반응하여야 한다.

#### 3.6.3 Security

3.6.3.1 .xml 파일은 모든 정보를 지니고 있으므로, 학사관리자에 의한 엄격한 관리가 필요하고, 필요 시에는 학사관리자 이외의 파일관리자를 고용할 수 있다.

#### 3.6.4 Maintainability

3.6.4.1 2명의 관리자(수업관리자, 학사관리자)를 통하여 지속적으로 사용할 수 있다.

#### 3.6.5 Portability

3.6.5.1 .NET Framework 3.5가 설치된 윈도우 환경에서 사용할 수 있다.

## 4 Supporting information

4.1 별도의 Use case 다이어그램, activity 다이어그램, sequence 다이어그램을 첨부.

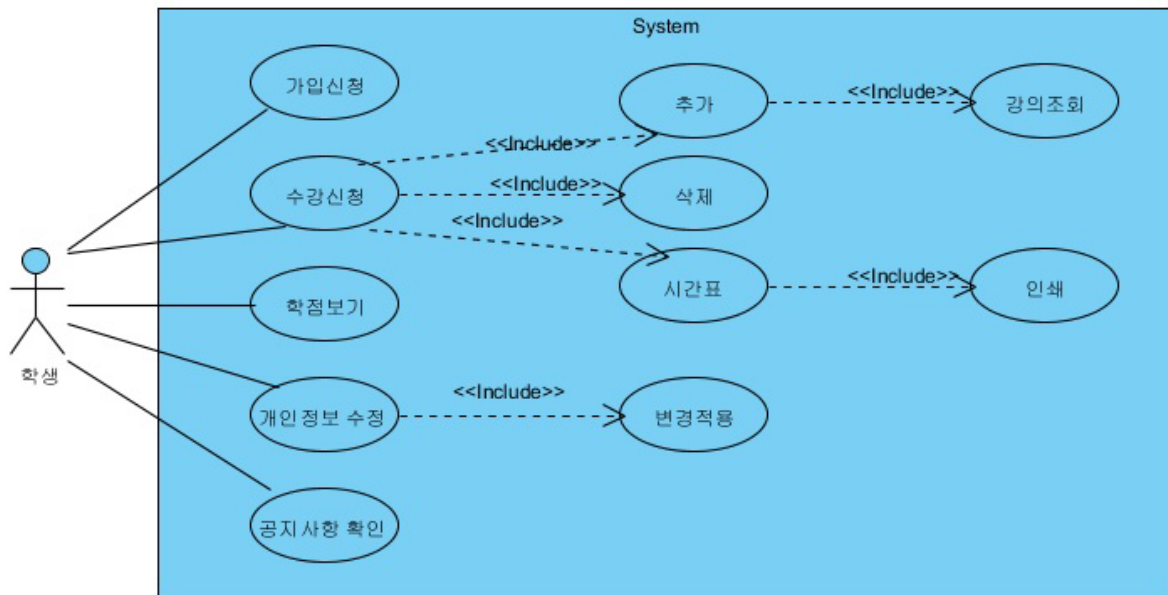
4.2 Use case 다이어그램을 통해 activity 다이어그램을 도출하였고, 이 activity 다이어그램을 통하여 세부 sequence 다이어그램을 도출하였다.

4.3 초기에 작성한 use case 다이어그램은 그대로 유지해 상세화 하지 않았다.

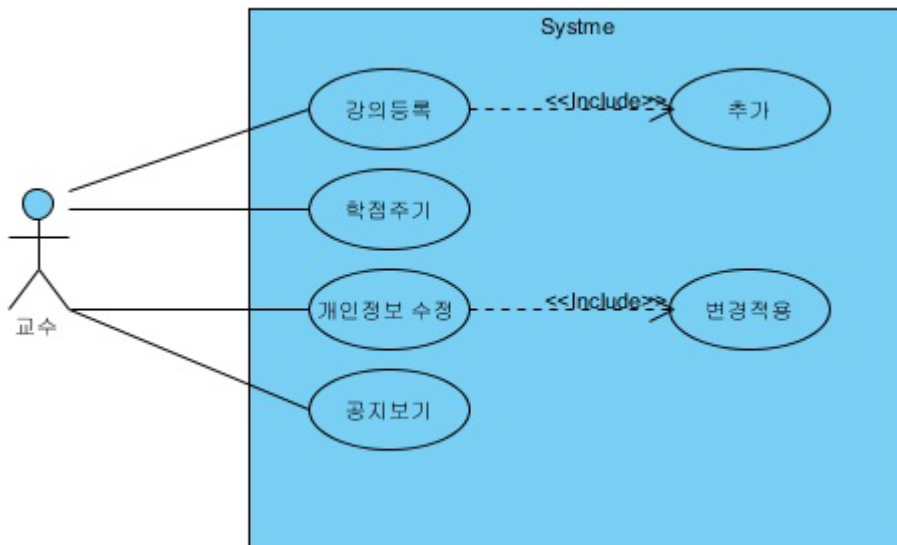


## 4.2 Usecase Diagram

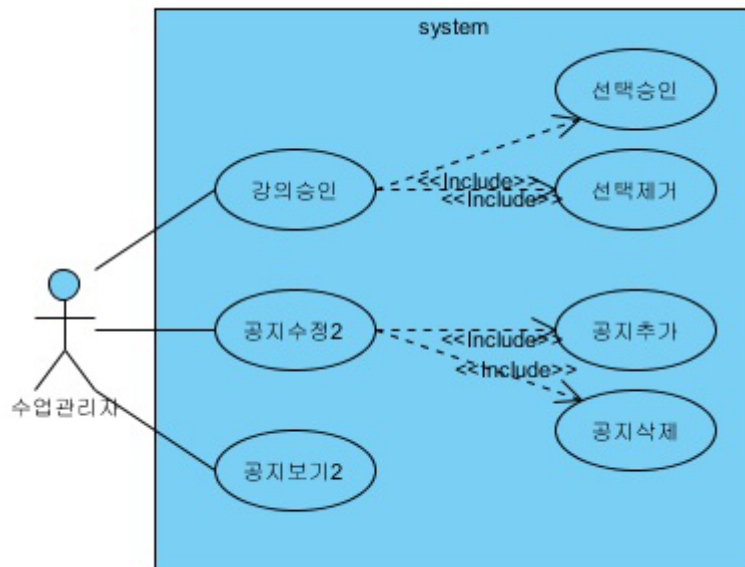
### 4.2.1 학생



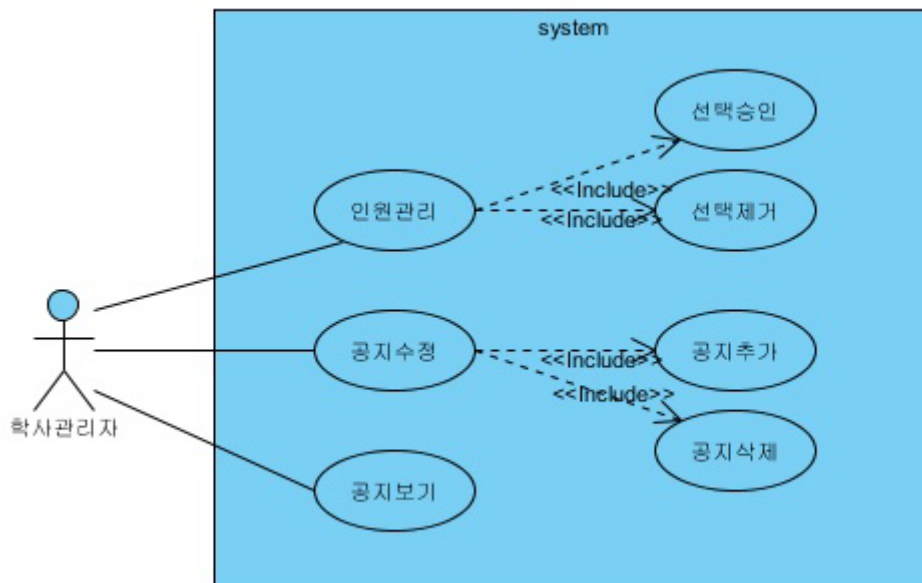
### 4.2.2 교수



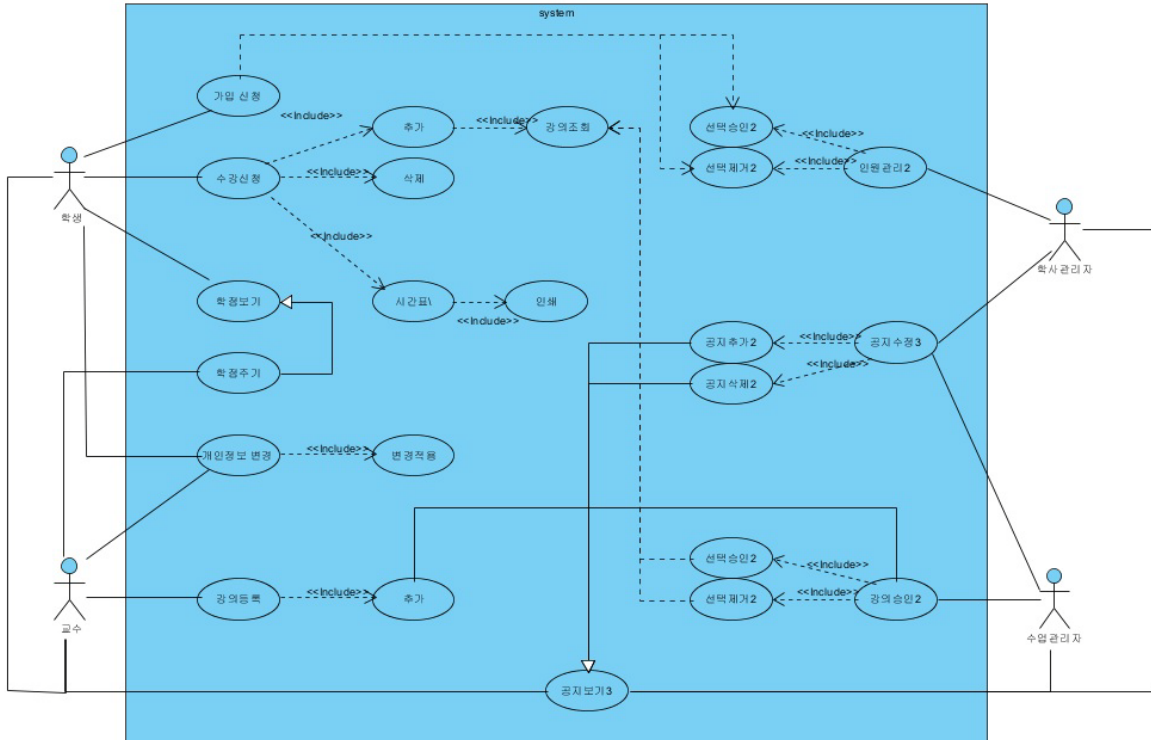
#### 4.2.3 수업관리자



#### 4.2.4 학사관리자

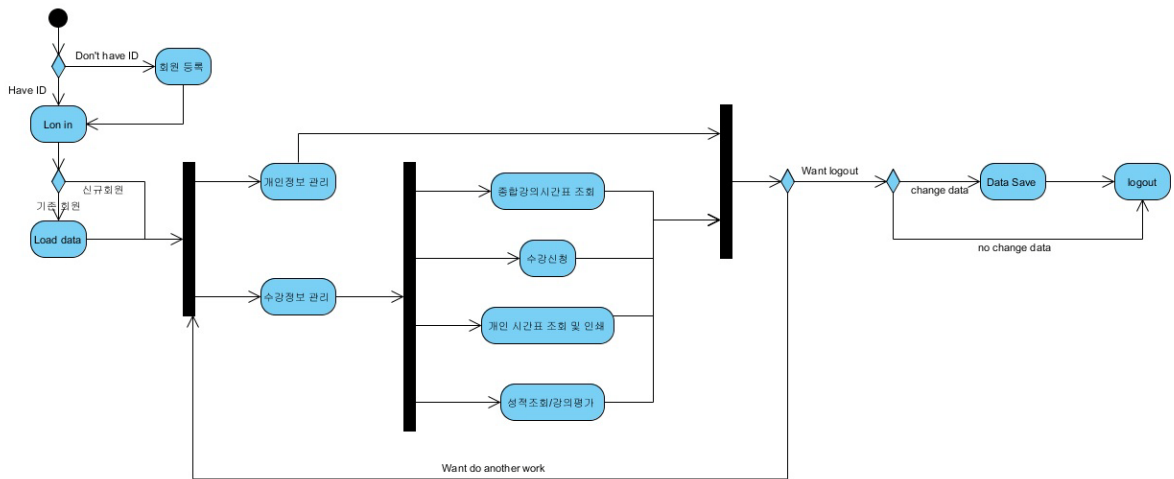


## 4.2.5 Overall

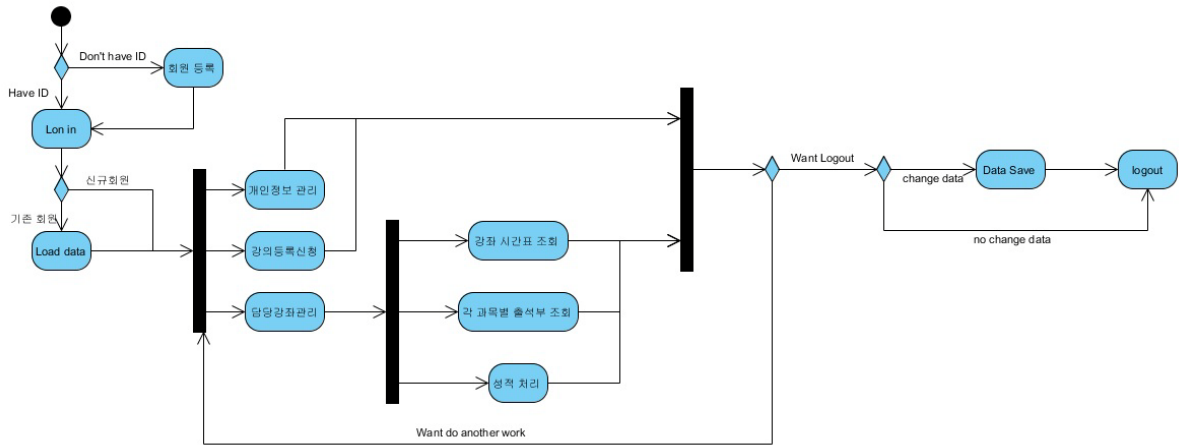


## 4.3 Activity Diagram

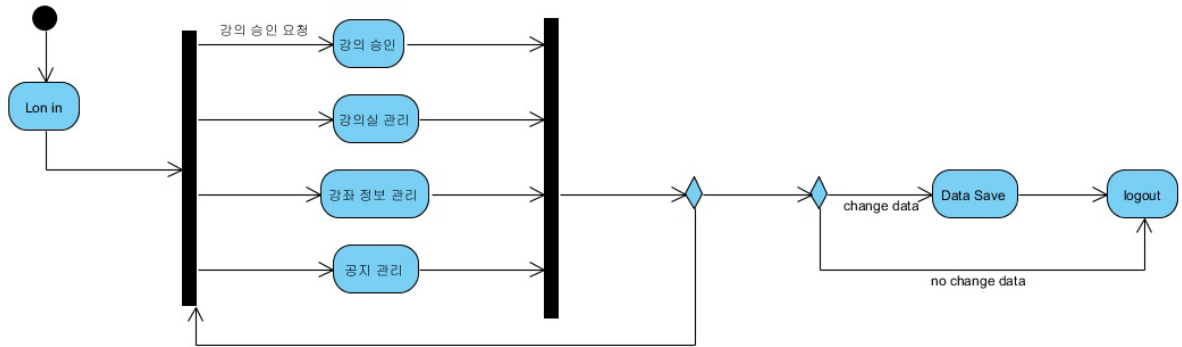
### 4.3.1 학생



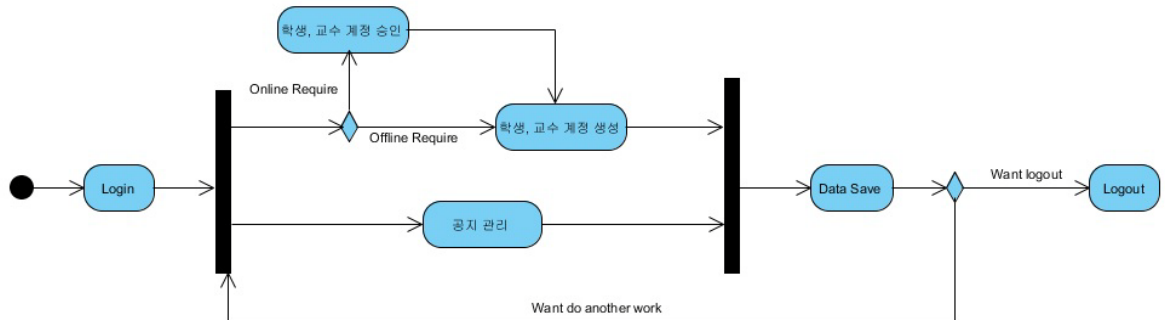
### 4.3.2 교수



### 4.3.3 수업관리자



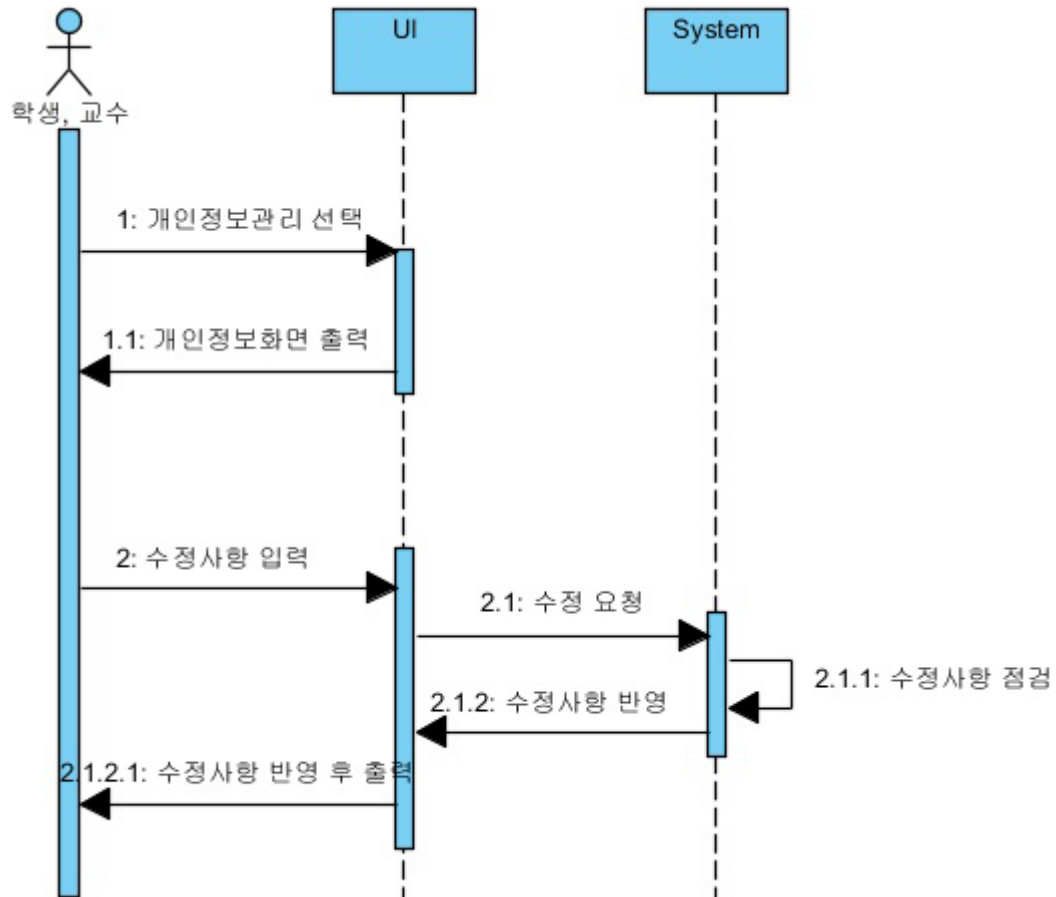
### 4.3.4 학사관리자



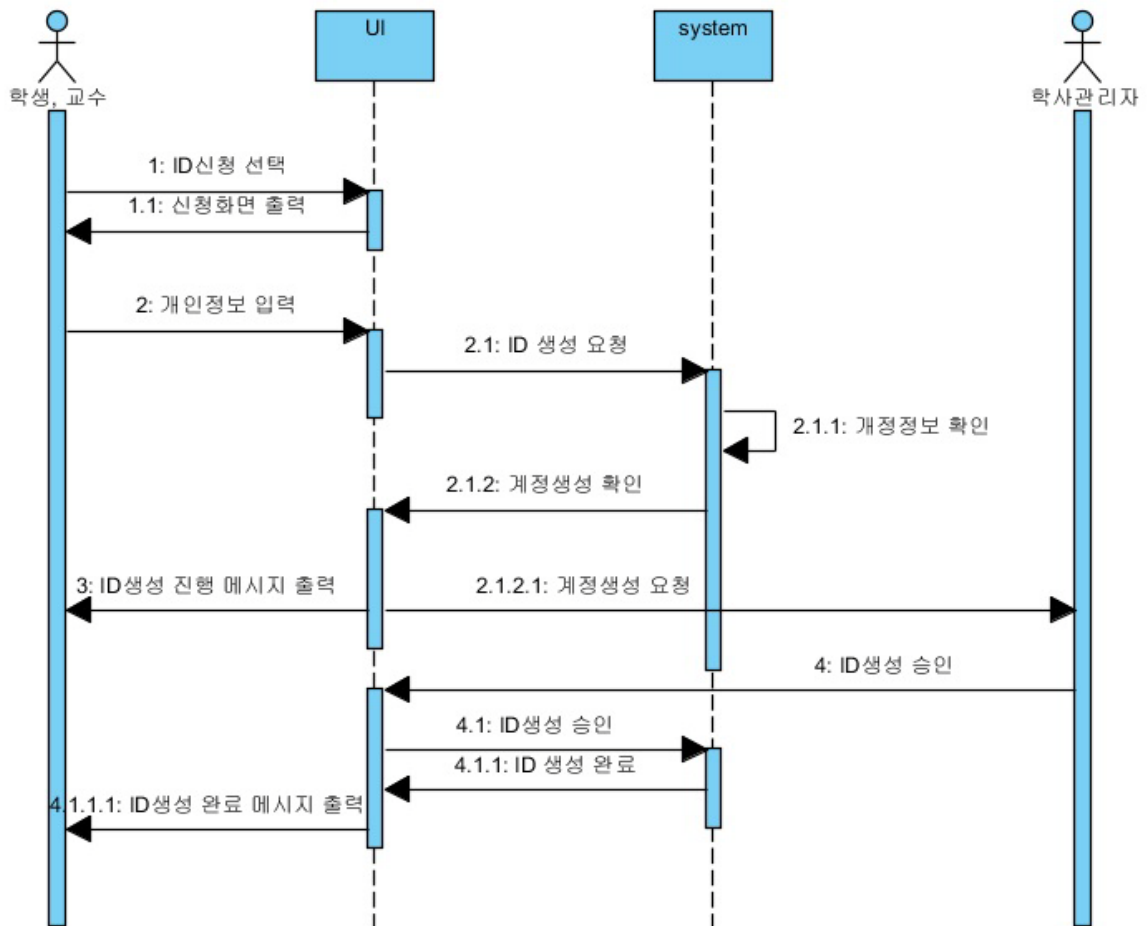
#### 4.4 Sequence Diagram

##### 4.4.1 공통사항

##### a. 개인정보 수정

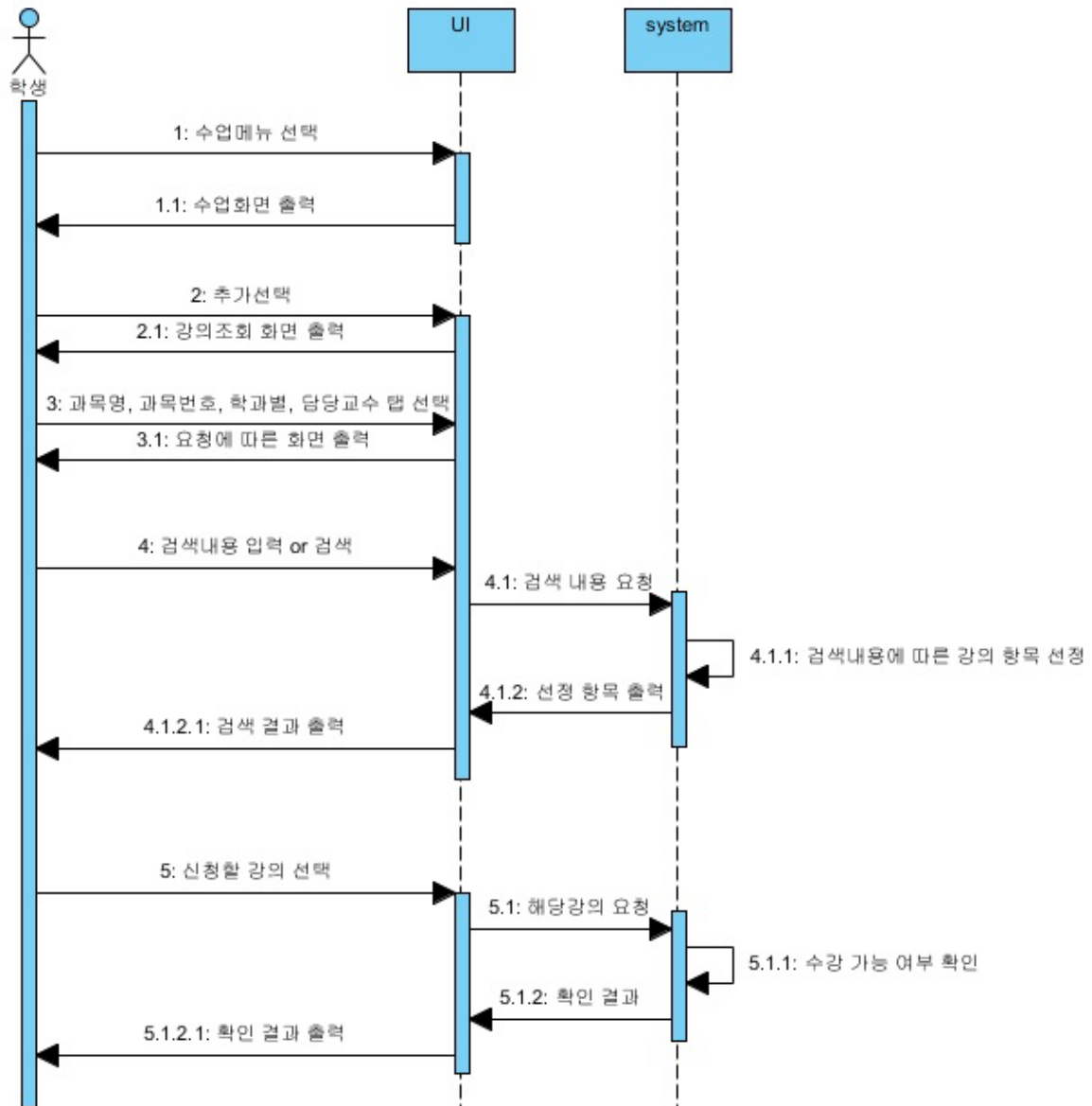


b. 계정생성

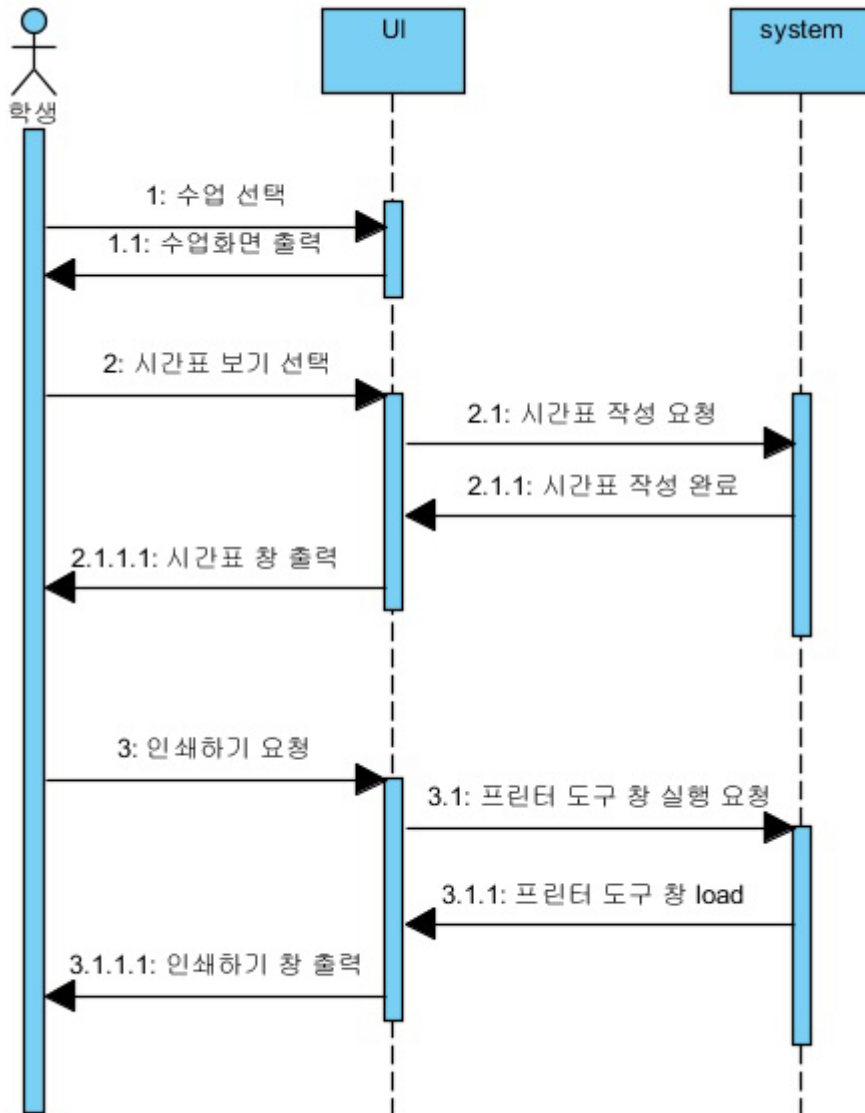


## 4.4.2 학생

### a. 수강 신청

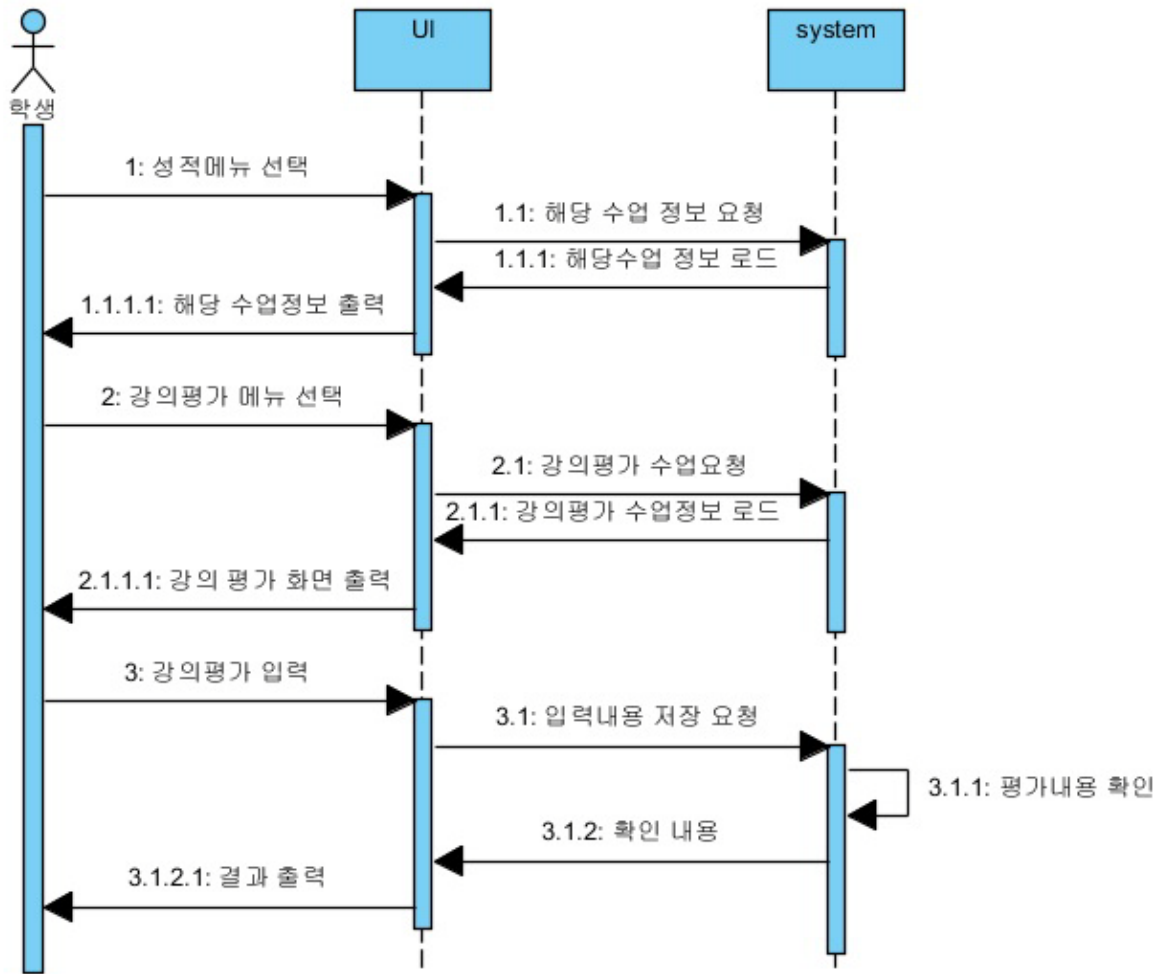


b. 개인강의 시간표 조회 및 인쇄



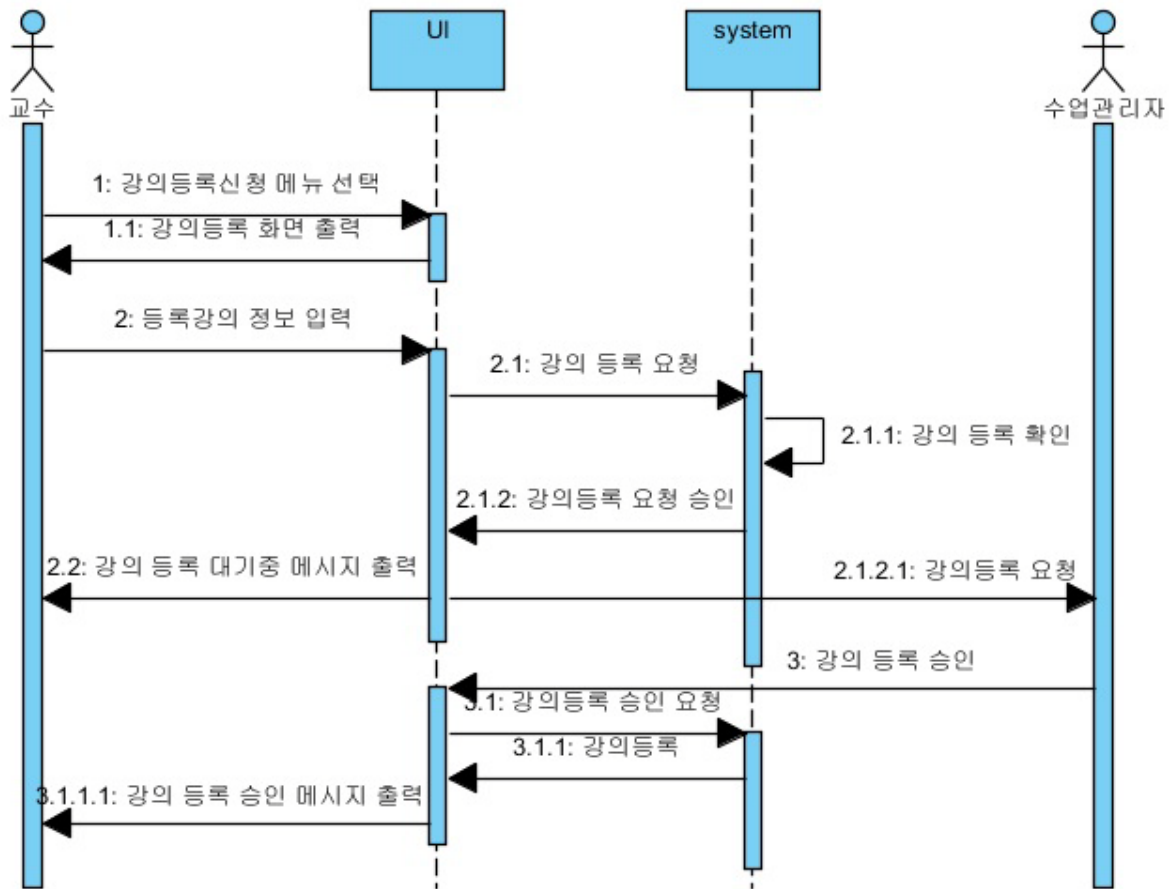


c. 성적확인, 강의평가

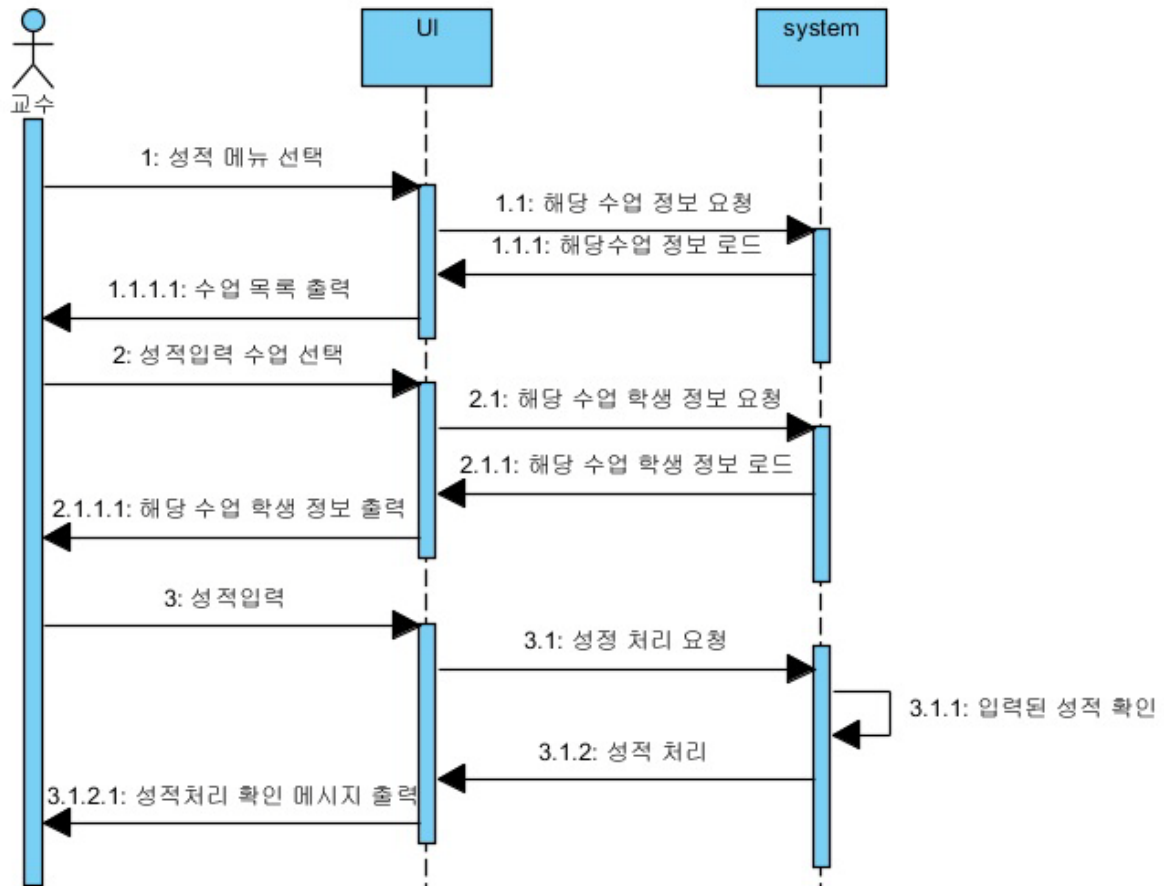


#### 4.4.3 교수

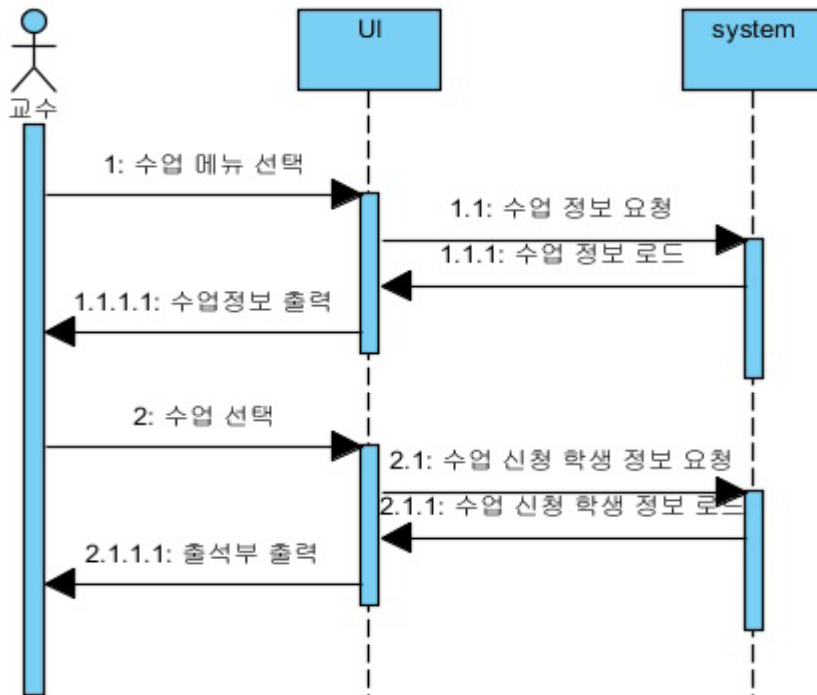
##### a. 강의등록 신청



b. 성적처리

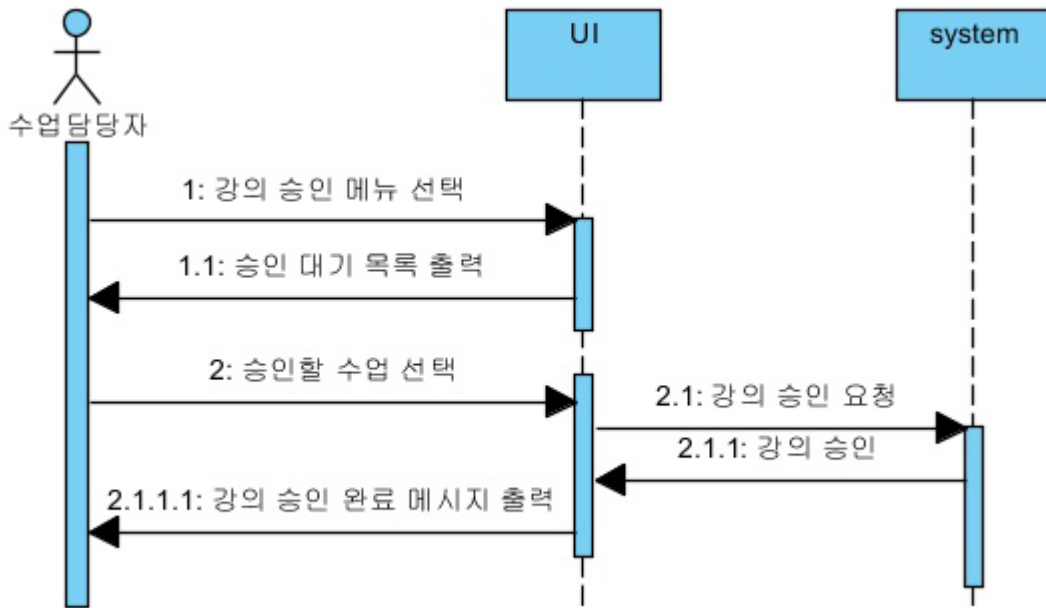


c. 출석부 확인 및



#### 4.4.4 수업관리자

##### a. 강의 승인



#### 4.4.5 학사관리자

##### a. 인원관리

