**소프트웨어 요구사항 분석 명세서**

**수강신청 도우미**

**시대 플랜메이트**

2019920026 서웅진

2019920011 김민서

2020920018 류재욱

2021910033 정명훈

2021920017 김영신

**변 경 이 력**

| **버전** | **일자** | **변경 내역** | **작 성 자** |
| --- | --- | --- | --- |
| A\_001 | 2024-10-12 | 1차 명세서 | 서웅진 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[***1.***](#_30j0zll) ***소개 3***

[**1.1**](#_1fob9te) **목적 3**

[**1.2**](#_3znysh7) **프로젝트 범위 (Scope) 3**

[**1.3**](#_2et92p0) **요구기능 추출 방법 (Methods) 3**

[**1.4**](#_tyjcwt) **정의 및 약어 4**

[***2.***](#_3dy6vkm) ***제약사항 (Constraints) 4***

[***3.***](#_1t3h5sf) ***사용자 분석 4***

[**3.1 액터 정의 4**](#_4d34og8)

[***4.***](#_2s8eyo1) ***요구사항 분석 5***

[**4.1**](#_17dp8vu) **기능 요구사항 (Functional Requirements) 5**

[**4.2**](#_3rdcrjn) **Use case 목록 7**

[4.2.1](#_26in1rg) Use case Diagram 8

[4.2.2](#_lnxbz9) Use case 명세 9

[***5.***](#_1ksv4uv) ***비기능 요구사항 (Non-Functional Requirements) 16***

# 소개

## 목적

소프트웨어 시스템의 목적, 사용자에게 주는 value (어떤 문제를 해결하는가)에 대해서 기술

본 서비스는 서울시립대학교 구성원의 수강신청 관련 업무를 돕기 위한 소프트웨어이다. 기존의 서울시립대학교 대학행정정보시스템만으로는 수강신청 과정에서 어려움이 따랐다. 대학행정정보시스템의 수강신청 페이지를 이용하면 학생은 본인이 원하는 조건의 강의를 상세하게 검색할 수 없다. 그 때문에 방대한 강의 목록 중 자신이 원하는 조건의 강의를 육안으로 일일이 물색해야 했다.

따라서 학생의 시간표 상 공백 시간대에 추가될 수 있는 강의의 목록을 보여주는 기능의 필요성이 제기되었다. 또한 원하는 조건에 해당하는 강의를 검색하는 기능의 필요성도 제기되었다. 본 서비스는 기존 대학행정정보시스템이 가진 한계를 해결하여 서울시립대학교 구성원의 수강신청 업무상 편의를 증진하는데 그 목적을 둔다.

## 프로젝트 범위 (Scope)

 본 프로젝트의 개발 범위는 다음과 같다.

* 회원 인증
* 서울시립대학교 개설강의 목록 검색 기능
* 예상시간표 표출 기능
* 개설강의 목록에서 예상시간표로의 강의 추가 기능
* 예상시간표상 빈 시간에 추가될 수 있는 강의 목록 표출 기능

## 요구기능 추출 방법 (Methods)

본 서비스가 필요로 하는 기능을 추출하기 위해 브레인스토밍을 진행하였다. 브레인스토밍은 팀 구성원 각자가 수강신청 과정에서 겪었던 불편을 제시하는 방식으로 진행하였다. 브레인스토밍을 통해 얻은 결과물 중에서 중복되는 내용을 제거하고, 비슷한 내용은 통합하였다. 이를 바탕으로 요구기능을 정리하였다. 이후 서울시립대학교 대학행정정보시스템이 이미 제공하고 있는 기능은 제외하였고, 실효성이 현저히 떨어지는 기능 역시 제외하여 Functional Requirements를 정리하였다.

## 정의 및 약어

| 예상시간표 | 본 서비스를 통해 학생이 계획한 시간표. 서비스 화면에 표출된다. 예상시간표의 형식은 서울시립대학교 학사규정이 정하는 대로 월~토요일, 1~15교시의 각 시간을 표 형태로 나타낸 것이다. |
| --- | --- |
| 개설 강의 목록 | 서울시립대학교에서 각 학기별로 개설되는 강의의 목록. |
| 검색 목록 | 서울시립대학교에서 각 학기별로 개설되는 강의 중, 사용자가 원하는 조건 필터를 선택하여 검색한 강의의 목록. |
| 공백 시간대 | 예상시간표 상에서 2학점 이상의 강의가 추가될 수 있는 시간 공백. |
| 추천 강의 목록 | 예상시간표 상에서 공백 시간대에 추가할 수 있는 강의의 목록. 강의의 개설 요일과 시간 데이터를 통해 검색하여 표출된다. |
| 수강 금지 과목 | 사용자(학생)의 소속(학부, 과)에 따라 수강이 금지된 과목. |

# 제약사항 (Constraints)

 본 서비스의 제약사항은 다음과 같다.

보안 및 학사규정 등의 이유로 수강신청 기능 그 자체를 구현하지는 않는다. 본 서비스는 사용자가 수강신청을 계획하고 준비하는 과정에서의 어려움을 해결하는데 그 중점을 둔다. 이에 사용되는 학사 DB는 강좌의 경우 서울시립대학교 대학행정정보시스템이 제공하는 강좌 정보를 바탕으로 제작할 것이며 학생 정보의 경우 임의로 설정하여 사용할 것이다.

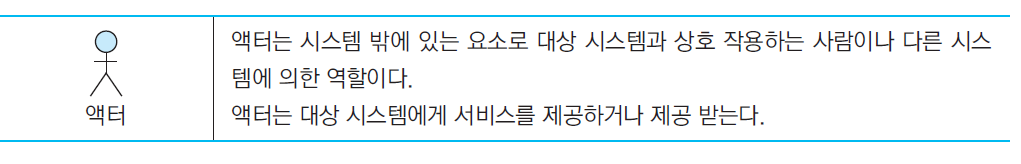
서울시립대학교 대학행정정보시스템이 자체적으로 제공하고 있는 수강신청 관련 기능은 구현하지 않는다. 그 예시로는 졸업요건을 충족하기 위해 수강해야 하는 영역별 학점 수를 조회하는 기능 등이 있다. 본 서비스는 대학행정정보시스템에 구현되어 있지 않으나, 사용자의 수강신청에의 편의를 위해 필요한 기능에 초점을 맞춘다.

본 프로젝트의 구현을 위해 사용한 기술 스택은 다음과 같다.

* 프론트엔드: React
* 백엔드: java spring, Oracle DB

# 사용자 분석

## 3.1 액터 정의



| 액터 ID | 액터 명 | 설명 |
| --- | --- | --- |
| AC\_01 | 학생 | 수강 신청 계획을 수립하기 위해 본 시스템을 이용하는 사용자 |
| AC\_02 | 개발자 | 해당 시스템의 보안, 서버 관리 등의 유지보수를 수행하는 관리자 |
| AC\_03 | db관리자 | 학생, 교수 및 과목 정보들과 관련된 테이블과 레코드들을 관리 및 담당하는 관리자 |
| AC\_04 | 서울시립대포털시스템 | 과목에 해당하는 정보들을 제공하는 시스템 |
|  |  |  |

# 요구사항 분석

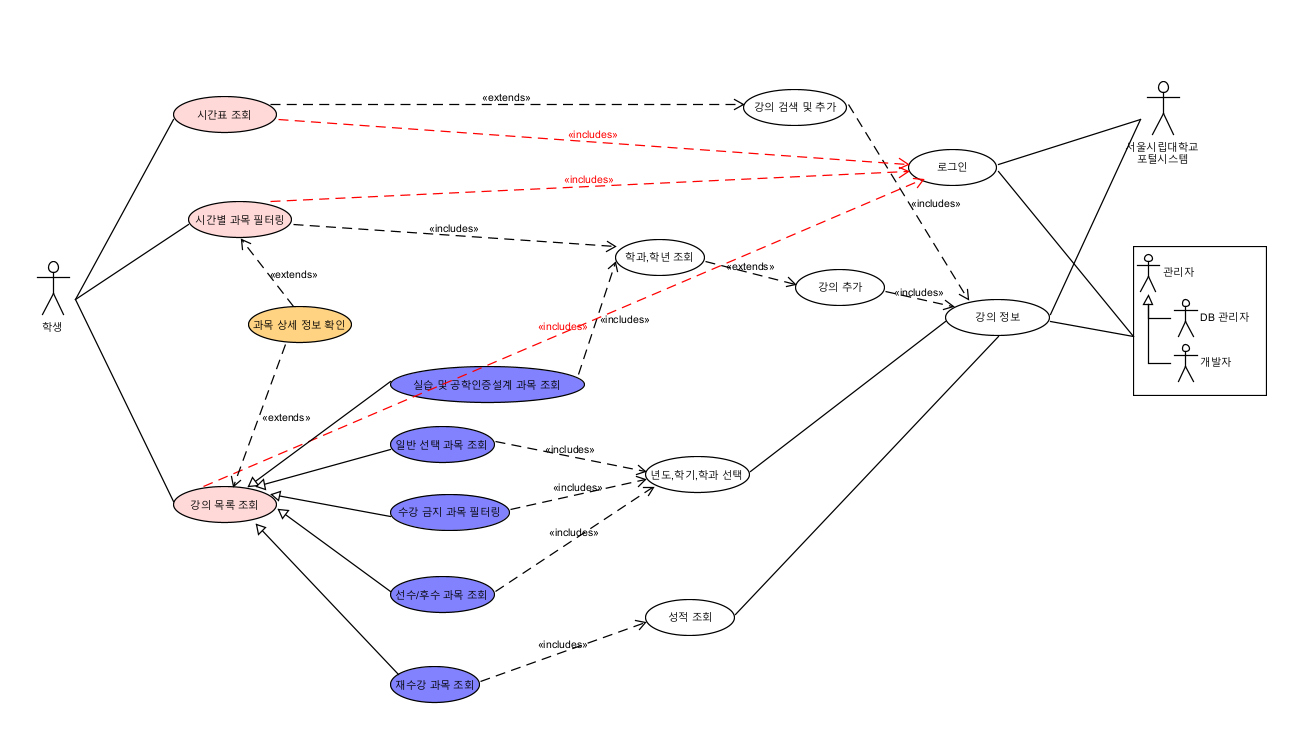
## 기능 요구사항 (Functional Requirements)

| ID | 구분 | 상세 | 비고 |
| --- | --- | --- | --- |
| FR001 | 일반 선택 과목 정렬 | - - -시스템은 학생이 선택한 학과의 일반 선택 과목 리스트를 제공해야 한다.  - 일반 선택 과목은 DB에서 학과 정보에 따라 가져와야 한다.  -과목들은 학기별로 정렬되어 보여야 한다.  -학생은 과목의 상세 정보를 확인할 수 있어야 한다. |  |
| FR002 | 수강 금지 과목 필터링 | - - -시스템은 학생이 선택한 학과에서 수강할 수 없는 과목을 자동으로 필터링하여 제공해야 한다.  -수강 불가 과목은 DB에서 학과 정보에 따라 가져와야 한다.  -필터링 된 과목은 과목 추천 목록에서 제거되어야 한다. |  |
| FR003 | 선수/후수 과목 정렬 | -시스템은 학생이 선택한 학과의 선수 과목과 후수 과목을 정렬해서 보여줘야 한다.  -선수과목의 학점이 F일 경우, 후수과목을 선택했을 때 알려줘야 한다.  -학생은 과목의 상세 정보를 확인할 수 있어야 한다. |  |
| FR004 | 실습 과목 및 공학인증설계 과목 정렬 | -시스템은 학생이 속한 학과의 실습 과목과 공학인증설계 과목에 대해 정렬된 리스트를 제공해야 한다.  -학 -학생은 과목의 상세 정보를 확인할 수 있어야 한다.  - -시스템은 학생이 추가한 강의를 시간표에 반영해야 한다. | 공학인증 프로그램 시행하지 않는 학과의 경우  설계의 경우 기초설계->요소설계->종합설계 순서로 이수 |
| FR005 | 재수강 과목조회 | -시스템은 학생이 로그인했을 때, C+ 이하의 성적을 받은 과목 리스트를 제공해야 한다.  -C+ 이하의 학점에 해당하는 과목은 재수강 가능 과목으로 분류되며, 수강 가능한 과목 목록에서 확인할 수 있어야 한다.  -학생은 과목의 상세 정보를 확인할 수 있어야 한다.  -폐지된 교과목이 있을 경우 대체 교과목을 안내해야 한다. |  |
| FR006 | 시간별 과목 필터링 | -시스템은 학생이 전공 필수/선택과 교양필수 과목이 모두 선택한 후, 빈 시간대에 수강 가능한 전공 필수, 선택 및 교양 또는 일반 선택 과목 리스트를 보여줘야 한다.  -학생은 과목의 상세 정보를 확인할 수 있어야 한다.  - 시스템은 학생이 추가한 강의를 시간표에 반영해야 한다. |  |
| FR007 | 시간표 | - 학생이 추가한 과목들을 시간표 형태로 보여준다.  - 시스템은 학생이 강의 검색을 통해 추가한 강의를 반영해야 한다.  -시간표는 학생의 시간표를 디스플레이 하는 기능을 제공해야 한다.  - 시스템은 시간표에 겹치는 과목을 추가하지 못하도록 해야 한다. |  |

## Use case 목록

| ID | 유스케이스 명 | 설명 | 우선 순위 |
| --- | --- | --- | --- |
| UC001 | 일반 선택 과목 조회 | 학생이 년도,학기,학과를 선택하면 해당 학과의 일반 선택 과목을 조회하고, 정렬된 리스트에서 과목을 선택할 수 있다. |  |
| UC002 | 수강 금지 과목 조회 | 학생이 년도,학기,학과를 선택하면 해당 학과에서 수강할 수 없는 과목을 자동으로 필터링하여 리스트를 보여준다. |  |
| UC003 | 선수 및 후수 과목 조회 | 학생이 년도,학기,학과를 선택하면 해당 학과의 선수/후수 과목을 정렬된 리스트로 확인하고, 관련 정보를 확인할 수 있다. |  |
| UC004 | 실습 및 공학인증설계 과목 조회 | 학생이 속한 학과의 실습 과목과 공학인증설계 과목에 대해 정렬된 리스트를 제공해야 하고 강의를 시간표에 추가할 수 있다. |  |
| UC005 | 재수강 과목 조회 | 학생이 로그인하면 C+ 이하의 성적을 받은 과목의 리스트를 보여준다. |  |
| UC006 | 시간별 과목 필터링 | 학생이 전공 필수/선택과 교양필수 과목이 모두 선택한 후, 빈 시간대에 수강 가능한 전공 필수, 선택 및 교양 또는 일반 선택 과목 리스트를 보여준다. |  |
| UC007 | 시간표 | 학생이 로그인하면 시간표를 보여준다. 강의 검색 후 추가할 수도 있다. |  |

### Use case Diagram



### Use case 명세

| 시스템 제목 | 수강신청 도우미 |
| --- | --- |
| UC\_001 | 일반 선택 과목 조회 |
| 액터 | 학생 |
| 시작 조건 | 학생이 로그인이 되어야 하고, 학과와 학년을 입력하여야 함. |
| 기본 흐름 | 1. 학생이 “일반 선택 과목 조회”를 선택한다.  2. 학생이 학과와 학년을 입력하고, 학생이 강의 리스트를 강의 목록에서  검색한다.  3. 시스템이 일반 선택 과목을 검색한다.  4. 학생이 강의 리스트를 조회한다. |
| 대안 흐름 | 2A  1. 학생의 학과와 학년이 입력되지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2B  1. 시스템이 학생의 학과와 학년을 접근할 수 없는 경우 일정 횟수로  반복 시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2C  1. 시스템이 강의 목록에 접근할 수 없는 경우 일정한 횟수로 반복  시도하고 이후에도 불가능하면 오류메시지를 디스플레이한다.  2. 학생이 오류 메시지를 인식하고 유스케이스를 종료한다.    3A  1. 일반 선택 과목이 없어서 조회가 되지 않을 경우,   오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    4A  1. 학생이 시간표에 추가하고 싶은 강의를 선택한다. 이후 ‘등록’ 버튼을   누른다.  2. 시스템이 해당 강의를 학생의 시간표에 추가한다.    4A-1A  1. 시간표에 추가한 강의가 기존에 시간표에 추가된 강의와 시간대가  겹칠 경우, 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다 |
| 종료 조건 | 과목 리스트를 확인하고 닫기를 누른다. |
| 관련 유스케이스 | UC\_007 |

| 시스템 제목 | 수강신청 도우미 |
| --- | --- |
| UC\_002 | 수강 금지 과목 필터링 |
| 액터 | 학생 |
| 시작 조건 | 학생이 로그인이 되어야 하고, 학과와 학년을 입력하여야 함. |
| 기본 흐름 | 1. 학생이 “수강 금지 과목 필터링”을 선택한다.  2. 학생이 학과를 입력하고, 학생이 강의 리스트를 강의 목록에서  검색한다.  3. 시스템은 수강 금지 과목을 강의 리스트에서 제외한 후  강의 리스트를 검색한다.  4. 학생이 강의 리스트를 조회한다. |
| 대안 흐름 | 2A  1. 학생이 학과를 입력지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2B  1. 시스템이 강의 목록에 접근할 수 없는 경우 일정한 횟수로 반복  시도하고 이후에도 불가능하면 오류메시지를 디스플레이한다.  2. 학생이 오류 메시지를 인식하고 유스케이스를 종료한다.    3A  1. 수강 금지 과목이 없어서 필터링 되지 않을 경우,   오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    4A  1. 학생이 시간표에 추가하고 싶은 강의를 선택한다. 이후 ‘등록’ 버튼을   누른다.  2. 시스템이 해당 강의를 학생의 시간표에 추가한다.    4A-1A  1. 시간표에 추가한 강의가 기존에 시간표에 추가된 강의와 시간대가  겹칠 경우, 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다. |
| 종료 조건 | 과목 리스트를 확인하고 닫기를 누른다. |
| 관련 유스케이스 | UC\_007 |

| 시스템 제목 | 수강신청 도우미 |
| --- | --- |
| UC\_003 | 선수 및 후수 과목 조회 |
| 액터 | 학생 |
| 시작 조건 | 학생이 로그인이 되어야 하고, 학과와 학년을 입력하여야 함. |
| 기본 흐름 | 1. 학생이 “선수 및 후수 과목 조회”를 선택한다.  2. 학생이 학과와 학년을 입력하고, 학생이 강의 리스트를 강의 목록에서  검색한다.  3. 시스템이 선수 및 후수 과목을 검색한다.  4. 학생이 강의 리스트를 조회한다. |
| 대안 흐름 | 2A  1. 학생의 학과와 학년이 입력되지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2B  1. 시스템이 학생의 학과와 학년을 접근할 수 없는 경우 일정 횟수로  반복 시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2C  1. 시스템이 강의 목록에 접근할 수 없는 경우 일정한 횟수로 반복  시도하고 이후에도 불가능하면 오류메시지를 디스플레이한다.  2. 학생이 오류 메시지를 인식하고 유스케이스를 종료한다.    3A  1. 선수 및 후수 과목이 없어서 조회가 되지 않을 경우,   오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    4A  1. 학생이 시간표에 추가하고 싶은 강의를 선택한다. 이후 ‘등록’ 버튼을   누른다.  2. 시스템이 해당 강의를 학생의 시간표에 추가한다.    4A-1A  1. 시간표에 추가한 강의가 기존에 시간표에 추가된 강의와 시간대가  겹칠 경우, 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다. |
| 종료 조건 | 과목 리스트를 확인하고 닫기를 누른다. |
| 관련 유스케이스 | UC\_007 |

| 시스템 제목 | 수강 신청 도우미 |
| --- | --- |
| UC\_004 | 실습 및 공학인증설계 과목 조회 |
| 액터 | 학생 |
| 시작 조건 | 학생이 로그인이 되어야 하고, 학과와 학년을 입력하여야 함. |
| 기본 흐름 | 1. 학생이 “실습 및 공학인증설계 과목 조회”를 선택한다.  2. 학생이 학과와 학년을 정보를 입력하고, 학생이 강의 리스트를  강의 목록에서 검색한다.  3. 시스템이 실습 및 공학인증설계 과목을 검색한다.  4. 학생이 강의 리스트를 조회한다. |
| 대안 흐름 | 2A  1. 학생의 학과와 학년이 입력되지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2B  1. 시스템이 학생의 학과와 학년을 접근할 수 없는 경우 일정 횟수로  반복 시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2C  1. 시스템이 강의 목록에 접근할 수 없는 경우 일정한 횟수로 반복  시도하고 이후에도 불가능하면 오류메시지를 디스플레이한다.  2. 학생이 오류 메시지를 인식하고 유스케이스를 종료한다.    3A  1. 실습 및 공학인증설계 과목이 없어서 조회가 되지 않을 경우,   오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    4A  1. 학생이 시간표에 추가하고 싶은 강의를 선택한다. 이후 ‘등록’ 버튼을   누른다.  2. 시스템이 해당 강의를 학생의 시간표에 추가한다.    4A-1A  1. 시간표에 추가한 강의가 기존에 시간표에 추가된 강의와 시간대가  겹칠 경우, 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다. |
| 종료 조건 | 학생이 유스케이스를 종료한다. |
| 관련 유스케이스 | UC\_007 |

| 시스템 제목 | 수강 과목 도우미 |
| --- | --- |
| UC\_005 | 재수강 과목 조회 |
| 액터 | 학생 |
| 시작 조건 | 학생이 이전 학기 성적을 입력하여야 하고, 로그인이 되어야 함. |
| 기본 흐름 | 1. 학생이 “재수강 과목 조회”를 선택한다.  2. 시스템이 학생의 성적표를 불러온 후, 재수강 가능 과목을 검색한다.  3. 시스템이 재수강 가능 과목을 강의 목록에서 검색한다.  4. 학생이 강의 리스트를 조회한다. |
| 대안 흐름 | 1A  1. 학생의 성적표가 입력되지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    1B  1. 시스템이 학생의 성적표에 접근할 수 없는 경우 일정 횟수로 반복   시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    2A  1. 재수강 가능 과목이 없는 경우, 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    3A  1. 시스템이 강의 목록에 접근할 수 없는 경우 일정한 횟수로 반복   시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다. |
| 종료 조건 | 학생이 유스케이스를 종료한다. |
| 관련 유스케이스 | UC\_007 |

| 시스템 제목 | 수강 과목 도우미 |
| --- | --- |
| UC\_006 | 시간별 과목 필터링 |
| 액터 | 학생 |
| 시작 조건 | 학생이 로그인 되어야 하고, 학과와 학년, 시간표가 입력되어 있어야 함. |
| 기본 흐름 | 1. 학생이 “시간별 과목 필터링”을 선택한다.  2. 학생이 학과와 학년 정보를 선택하고, 학생이 수강 가능한 강의   리스트를 강의 목록에서 검색한다.  3. 시스템이 학생의 시간표와 성적표를 불러온 후, 학생의 시간표와   겹치거나, 재수강 할 수 없는 강의를 필터링한다.  4. 학생이 필터링 된 강의 리스트를 조회한다. |
| 대안 흐름 | 2A  1. 학생의 학과와 학년이 입력되지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2B  1. 시스템이 학생의 학과와 학년을 불러올 수 없는 경우 일정 횟수로   반복 시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    2C  1. 시스템이 강의 목록에 접근할 수 없는 경우 일정한 횟수로 반복   시도하고 이후에도 불가능하면 오류메시지를 디스플레이한다.  2. 학생이 오류 메시지를 인식하고 유스케이스를 종료한다.    3A  1. 학생의 시간표와 성적표가 입력되지 않았을 경우, 오류메시지를   디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    3B  1. 시스템이 학생의 시간표와 성적표에 접근할 수 없는 경우 일정   횟수로 반복 시도하고 이후에도 불가능한 경우  디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다.    4A  1. 학생이 시간표에 추가하고 싶은 강의를 선택한다. 이후 ‘등록’ 버튼을   누른다.  2. 시스템이 해당 강의를 학생의 시간표에 추가한다.    4A-1A  1. 시간표에 추가한 강의가 기존의 시간표에 추가된 강의와 시간대가   겹칠 경우, 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고, 유스케이스를 종료한다. |
| 종료 조건 | 학생이 유스케이스를 종료한다. |
| 관련 유스케이스 | UC\_007 |

| 시스템 제목 | 수강 과목 도우미 |
| --- | --- |
| UC\_007 | 시간표 |
| 액터 | 학생 |
| 시작 조건 | 학생이 로그인 되어야 함. |
| 기본 흐름 | 1. 학생이 “시간표”를 선택한다.  2. 시스템이 학생의 시간표를 불러온다.  3. 학생이 시간표를 조회한다. |
| 대안 흐름 | 2A  1. 시스템이 학생의 시간표에 접근할 수 없는 경우, 일정한 횟수로 반복   시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다.    3A  1. 학생이 시간표에 강의를 추가하고 싶은 경우, ‘강의 검색’ 버튼을   누른다.  2. 시스템이 강의 리스트를 강의 목록에서 검색한다.  3. 학생이 강의 리스트에서 추가하고 싶은 강의를 검색하고, 해당  강의를 선택하고, ‘등록’ 버튼을 누른다.    3A-2A  1. 시스템이 강의 목록에 접근할 수 없는 경우, 일정한 횟수로 반복  시도하고 이후에도 불가능한 경우 오류메시지를 디스플레이한다.  2. 학생이 오류메시지를 인식하고 유스케이스를 종료한다. |
| 종료 조건 | 학생이 유스케이스를 종료한다. |
| 관련 유스케이스 | UC\_.001, UC\_002, UC\_003, UC\_004, UC\_005, UC\_006 |

# 비기능 요구사항 (Non-Functional Requirements)

| ID | 요구 항목 | 설명 |
| --- | --- | --- |
| NF001 | 성능 | 서울시립대 학생 수는 약 9,000명이므로 5%인 450명의 사용자가 동시에 사용할 경우에도 시스템은 3초 내로 사용자가 요구한 서비스에 대한 응답을 해야 한다. 특히 많은 요청이 예상되는 장바구니 수강 신청 기간에도 부하를 견딜 수 있어야 한다. |
| NF002 | 보안 | 사용자는 다른 사용자가 수강한 과목 리스트나 학과 등의 정보에 접근할 수 없어야 한다. 사용자의 아이디, 패스워드 및 개인 정보를 외부에서 접근할 수 없어야 한다. 사용자의 패스워드의 평문은 시스템 내부에서도 접근할 수 없도록 해시 형태로 저장해야 한다. |
| NF003 | 신뢰 | 사용자에게 정확한 정보를 제공하기 위해 시스템에 등록된 강의 정보나 학과 정보는 실제 시립대의 학사 정보와 정확히 일치해야 한다. 사용자가 등록한 자신의 수강 정보는 시스템 장애가 발생해도 유실되지 않아야 한다. |
| NF004 | 사용성 | 도움말이나 사용 가이드가 없어도 사용자가 원하는 정보를 얻는 방법을 알 수 있어야 하며, 4번의 클릭 이내로 원하는 정보를 얻을 수 있어야 한다. |
| NF005 | 유지보수 | 수강 신청 기간 중 강의 정보가 변경되었을 때는 시스템에 등록된 강의 정보도 1일 이내에 반영되어야 한다. 새로운 학기가 시작되었을 때는 해당 학기의 장바구니 수강 신청 기간 전까지 강의 정보가 추가되어야 한다. |
| NF006 | 가용성 | 시스템에 새 학기의 수강 정보가 등록된 이후부터 수강 신청 기간 전까지는 언제나 서비스를 이용할 수 있어야 한다. 서비스 이용이 불가한 유지보수 작업이 필요하다면 이 기간을 피해 작업해야 한다. |
| NF007 | Interoperability | iOS, Android, PC 등의 플랫폼에 상관없이 유저가 편하게 사용할 수 있어야 한다. |