요구사항 명세서

<팀 명>

- 변 경 이 력 -

|  |  |  |  |
| --- | --- | --- | --- |
| 일자 | **버전** | **변경 내역** | **작 성 자** |
| 2018.04.06 | 1.0 | 시스템 개요 작성, 비기능 요구사항작성 | 박주원 |
| 2018.04.09 | 1.1 | 유스케이스 목록 및 유스케이스 다이어그램 작성 | 조금비 |
| 2018.4.10. | 1.2 | 사용자 인터페이스 요구사항, 화면 기술 작성 | 이지홍 |
| 2018.04.11 | 1.3 | 기능 요구사항, 유스케이스 기술 작성 | 이은선 |
| 2018.04.12 | 1.4 | 사용자 분석 및 목차 추가 | 권이안 |
|  |  |  |  |
|  |  |  |  |

- 목 차 -

1. 시스템 개요

2. 사용자 분석

2.1 액터 정의

2.2 액터 다이어그램

3. 요구사항 분석

3.1 기능 요구사항

3.2 유스케이스 목록

3.2.1 Use Case 다이어그램

3.2.2 Use Case 기술

3.3 사용자 인터페이스 요구사항

3.3.1 화면 목록

3.3.2 화면기술

4. 비기능 요구사항

1. 시스템 개요
2. **기능**
3. -TO DO LIST 프로그램은 과목별로 할 일을 입력할 수 있다.
4. -입력한 할 일에 대해 중요 여부를 체크 할 수 있다.
5. -입력한 항목들을 과목명, 마감 기한, 실제 마감일, 완료 여부로 정렬이 가능하다.
6. -과목 등록시에는 과목명, 담당 교수,
7. -강의 요일/시간, 수강 년도/ 학기를 입력해야 한다.
8. -모든 과목과 등록된 할 일 들은 수정과 삭제가 가능 하다.
9. **특징**
10. -디자인의 메인 색은 와인색(빨간 색 계통) 이며, 깔끔함을 추구하는 디자인을 목표로한다.
11. -GUI의 가독성 중시 //과목별 색상 지정
12. **효과**
13. -과목별로 일정을 Listing하여, 할 일 들을 잊지 않고 더욱 체계적으로 관리 할 수 있다.
14. 사용자 분석
    1. 액터 정의

|  |  |
| --- | --- |
| 액터 | **설명** |
| 사용자 | 소프트웨어를 사용하여 스케줄을 관리하는 주체. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* 1. 액터 다이어그램



1. 요구사항 분석
   1. 기능 요구사항

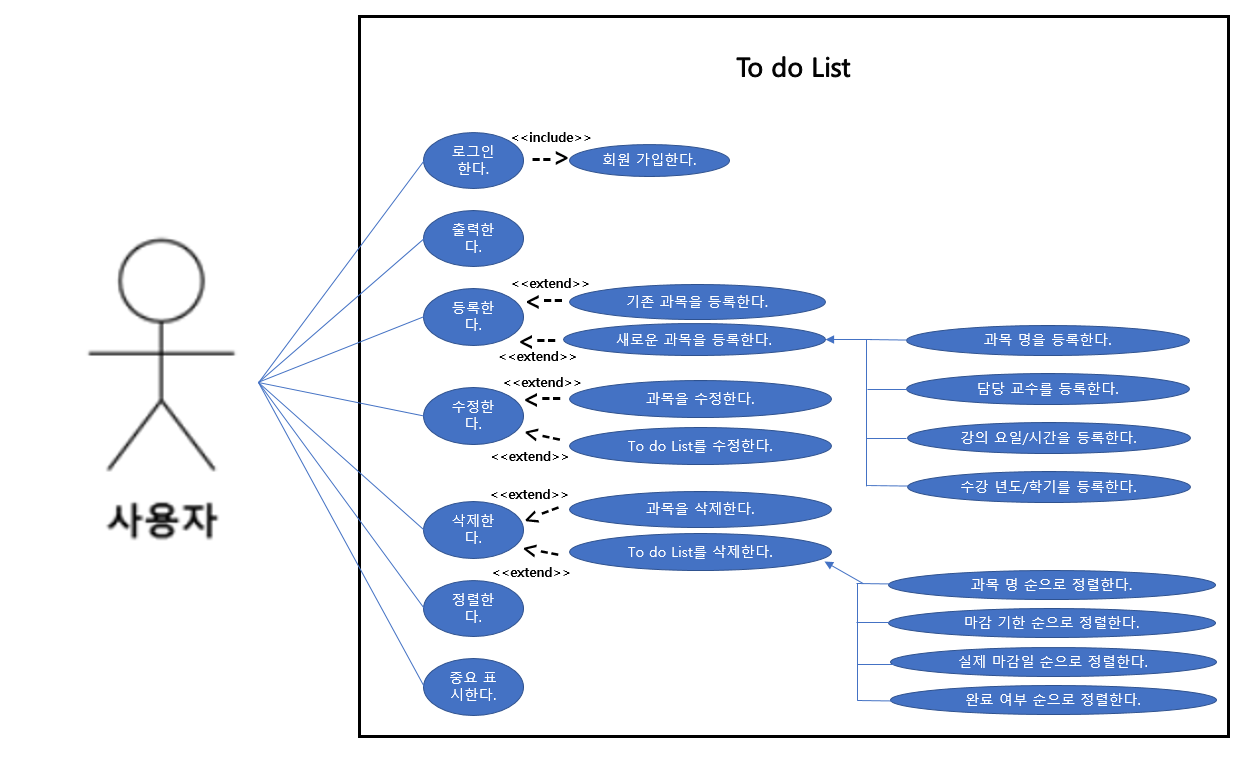
|  |
| --- |
| 1.로그인  i.화면 한가운데에 작은 로그인 창을 띄운다.  ii.로그인시 아이디 비밀번호 입력할 수 있는 빈칸을 한 줄로 길게 띄운다.  iii.아이디 비밀번호를 기존 DB data와 비교하여 둘 다 맞으면 로그인 창이 닫히고 To do List 기본화면으로 접속하게 한다.  iv.아이디, 비밀번호 오류가 나는 경우 다시 입력할 수 있도록 한다. (무한반복)  v.아이디가 오류가 난 경우는 아이디입력 창 바로 아래에 “계정 오류” 알림과 함께 입력한 아이디와 비밀번호가 지워진다.  vi.비밀번호가 오류가 난 경우에는 비밀번호입력 창 바로 아래에 “비밀번호 오류” 알림과 함께 비밀번호가 지워진다. 이때 계정아이디는 지우지 않고 커서는 비밀번호 창에 뜬다.  vii.로그인 줄 밑에 회원가입 버튼 만든다.  i.회원가입: 아이디\n비밀번호\n비밀번호재확인\n닉네임\n본인확인이메일\n을 다 입력하고 가입하기 버튼을 누르면 DB에 새 계정이 추가되고 “회원가입 완료”라는 작은 메시지 창이 출력되고 확인버튼을 누르면 자동으로 로그인 초기화면으로 돌아간다.  2. To do List 기본화면  i.등록되어 있는 list 목록을 출력한다.  ii.목록 아래에 등록 수정 삭제 정렬 버튼이 있다.  3. 등록  i.등록버튼 누르면 ‘새과목에 등록’과 ‘기존과목에 등록’으로 나뉨  ii.‘새과목에 등록’은 과목 명, 담당 교수, 강의 요일/시간, 수강 년도/학기 창을 띄워 기입하게 한다. 등록하기 누르면 과목table의 새로운 tuple이 추가되고 바로 to do list작성 화면으로 넘어간다. 예) 소프트웨어공학, 한혁수 교수님, 월/3-6pm, 2018/1  iii.‘기존과목에 등록’은 번호로 과목 목록에서 과목을 선택하고 to do list 작성화면으로 넘어간다.  iv.뒤로 가고 싶으면 취소 창을 누른다.  v.To do List 작성화면에는 To do 항목 명, 마감 기한, 실제 마감일, 완료 여부, 중요 여부 창을 띄워 기입하게 하고 완료를 누르면 To do list 기본화면으로 돌아가게 한다.  예) 요구사항명세서 작성, 2018/04/13, 2018/04/12, O, (중요 여부)  vi.각 과목 table과 To do list table의 DB에 새로운 tuple이 입력된다.  4. 수정  i.수정버튼 누르면 ‘과목수정’과 ‘To do List 수정’으로 나뉨.  ii.‘과목수정’누르면 기존 존재하는 과목 리스트 출력하고 수정하고 싶은 과목 앞의 버튼을 누르면 수정할 수 있는 입력창이 뜨고 수정하기를 누르면 완료버튼이 뜨고 To do List 기본화면으로 돌아간다.  iii.‘To do List 수정’누르면 기존 존재하는 To do list 출력하고 수정하고 싶은 리스트 앞의 버튼을 누르면 수정할 수 있는 입 각 과목table과 To do list table의 DB도 수정되어 입력된다. 그리고 To do List 기본화면으로 돌아간다.  iv.력창이 뜨고 수정하기를 누르면 완료버튼이 뜨고 To do List 기본화면으로 돌아간다.  v.각 과목 table과 To do list table의 DB도 수정되어 입력된다.  5. 삭제  i.삭제버튼 누르면 ‘과목삭제’와 ‘To do List 삭제’로 나뉨.  ii.‘과목삭제’누르면 기존 존재하는 과목 리스트 출력하고 삭제하고 싶은 과목 앞의 버튼을 누르면 ‘정말 삭제하시겠습니까?’라는 문구가 나오고 ‘YES’선택 시 해당 tuple과 연결된 to do list목록을 전부 삭제한다. 그리고 ‘삭제완료’가 출력된 후 To do List 기본화면으로 돌아간다. ‘NO’선택 시 To do List 기본화면으로 돌아간다.  iii.‘To do List 삭제’누르면 기존 존재하는 To do list 출력하고 삭제하고 싶은 리스트 앞의 버튼을 누르면 ‘정말 삭제하시겠습니까?’라는 문구가 나오고 ‘YES’선택 시 해당 리스트만 삭제한다. 그리고 ‘삭제완료’가 출력된 후 To do List 기본화면으로 돌아간다. ‘NO’선택 시 To do List 기본화면으로 돌아간다.  6. 정렬  i.정렬버튼을 누르면 과목 명, 마감 기한, 실제 마감일, 완료 여부 중하나를 선택하게 하고 누른 상태로 정렬시킨 후 To do List 기본화면이 정렬된 상태로 유지되게 한다.  7.중요 To do 항목에는 별도 표시  i.To do List 기본화면의 리스트의 앞의 버튼을 누르면 리스트의 내용이 빨간색으로 출력되면서 유지된다. |

|  |  |  |
| --- | --- | --- |
| 구분 | **상세** | **비고** |
| 로그인 | 아이디계정과 비밀번호를 입력하고 To do List 초기화면으로 접속한다. 회원가입 기능과 로그인 오류시 재로그인을 가능하게 하는 기능을 넣는다. |  |
| 등록 | 새 과목에 To do List를 넣거나 기존 과목에 To do List를 넣는다. |  |
| 수정 | 기존과목을 수정하여 저장하거나 기존 과목의 To do List를 수정하여 저장한다. |  |
| 삭제 | 기존 과목을 삭제하고 하위 To do List를 삭제하거나 개별 To do List를 삭제한다. |  |
| 정렬 | 과목 명, 마감 기한, 실제 마감일, 완료 여부 중 하나를 선택하면 과목명의 가나다순으로, 기한이 빠른 것부터, 실제 마감일이 빠른 것부터 완료 여부를 누른것의 가나다 순으로 정렬이 된다. |  |
| 중요 To do 항목에는 별도 표시 | 중요한 항목을 별도로 표시 가능하다. |  |

* 1. 유스케이스 목록

|  |  |  |  |
| --- | --- | --- | --- |
| ID | **유스케이스 명** | **설명** | **우선**  **순위** |
| UC\_Login\_001 | 로그인한다. | 아이디와 비밀번호를 입력하여 로그인 할 수 있도록 한다. | 1 |
| UC\_Login\_  002 | 회원 가입한다. | 로그인 화면에 회원 가입을 할 수 있는 버튼을 만든다. | 1 |
| UC\_Output\_001 | 출력한다. | 등록되어 있는 To do List 목록을 출력한다. | 2 |
| UC\_Reg\_001 | 기존 과목을 등록한다. | 기존 과목을 선택하여 To do List를 작성한다. | 3 |
| UC\_Reg\_002 | 새로운 과목을 등록하며 과목 명을 설정한다. | 과목 명을 부여하며 새로운 과목을 등록한다. | 3 |
| UC\_Reg\_003 | 새로운 과목을 등록하며 담당 교수를 설정한다. | 담당 교수를 부여하며 새로운 과목을 등록한다. | 3 |
| UC\_Reg\_004 | 새로운 과목을 등록하며 강의 요일/시간을 설정한다. | 강의 요일 및 시간을 부여하며 새로운 과목을 등록한다. | 3 |
| UC\_Reg\_005 | 새로운 과목을 등록하며 수강 년도/학기를 설정한다. | 수강 년도 및 학기를 부여하며 새로운 과목을 등록한다. | 3 |
| UC\_Modify\_001 | 과목을 수정한다. | 과목 리스트를 띄워 원하는 과목을 선택하여 수정할 수 있도록 한다. | 4 |
| UC\_Modify\_002 | To do List를 수정한다. | To do List를 띄워 원하는 리스트를 선택하여 수정할 수 있도록 한다. | 4 |
| UC\_Delete\_001 | 과목을 삭제한다. | 과목 리스트를 띄워 원하는 과목을 선택하여 삭제한다. | 4 |
| UC\_Delete\_002 | To do List를 삭제한다. | To do List를 띄워 원하는 리스트를 선택하여 삭제한다. | 4 |
| UC\_Sort\_001 | 과목 명 순으로 정렬한다. | 출력된 리스트를 과목 명 순으로 정렬한다. | 5 |
| UC\_Sort\_002 | 마감 기한 순으로 정렬한다. | 출력된 리스트를 마감 기한 순으로 정렬한다. | 5 |
| UC\_Sort\_003 | 실제 마감일 순으로 정렬한다. | 출력된 리스트를 실제 마감일 순으로 정렬한다. | 5 |
| UC\_Sort\_004 | 완료 여부 순으로 정렬한다. | 출력된 리스트를 완료 여부 순으로 정렬한다. | 5 |
| UC\_Important\_001 | 중요한 To do 항목을 표시한다. | 중요한 To do 항목을 별도로 표시하여 출력한다. | 6 |

* + 1. 유스케이스 다이어그램



* + 1. 유스케이스 기술
       1. ID: UC\_Login\_001

|  |  |  |
| --- | --- | --- |
| 설명 | 로그인을 한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 |  | |
| 사후 조건 | 로그인되어 To do List 초기화면으로 접속된다. | |
| 기본 흐름 | **B01** | 시스템은 사용자에게 로그인을 요구한다. |
| **B02** | 사용자는 아이디 비밀번호를 입력한다. |
| **B03** | 사용자는 로그인버튼을 누른다. |
| **B04** | 시스템은 To do List초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 아이디 오류시 아이디, 비밀번호가 지워진다. |
| **E02** | 비밀번호 오류시 비밀번호만 지워진다. |
| **E03** | 계정 등록이 안되어 있을 경우 회원가입으로 계정을 새로 등록한다. |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01->B02->B03->B04 |
| **SN003** | B01->B02->E02->B02->B03->B04 |
| **SN004** | B01->E03->B01->B02->B03->B04 |

* + - 1. ID: UC\_Login\_002

|  |  |  |
| --- | --- | --- |
| 설명 | 회원가입을 한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 사용자의 계정이 등록되어 있지 않다. | |
| 사후 조건 | 계정이 새로 등록되어 사용자계정 DB에 추가생성된다. | |
| 기본 흐름 | **B01** | 빈 회원가입창을 띄운다. |
| **B02** | 사용자는 아이디, 비밀번호, 비밀번호 재확인, 닉네임, 본인확인 이메일을 기입한다. |
| **B03** | 사용자는 회원가입완료 버튼을 누른다. |
| **B04** | 시스템은 로그인 창으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 계정이 존재할 경우 회원가입이 완료되지 않고 이미 존재하는 계정이라는 메시지 창이 출력된다. |
| **E02** | 비밀번호와 비밀번호 재확인의 내용이 같지 않을 경우 비밀번호가 다르다는 메시지 창이 출력된다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->B03->E01->B04 |
| **SN003** | B01->B02->B03->B02->B03->B04 |
|  |  |

* + - 1. ID: UC\_Output\_001

|  |  |  |
| --- | --- | --- |
| 설명 | 출력한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 |  | |
| 사후 조건 | To do List 초기화면에 등록되어 있는 list 목록을 출력한다. | |
| 기본 흐름 | **B01** | 시스템은 To do List 초기화면에 등록되어 있는 List 목록을 출력한다. |
|  |  |
|  |  |
|  |  |
| 대안 흐름 | **A01** | To do List에 등록되어 있는 항목이 없는 경우 ‘비어있음’을 출력한다. |
| A02 | To do List에 등록되어 있는 항목이 있는 경우 전부 출력한다. |
| **예외 흐름** |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->A01 |
| **SN002** | B01->A02 |
|  |  |
|  |  |

* + - 1. ID: UC\_Reg\_001

|  |  |  |
| --- | --- | --- |
| 설명 | 기존과목에 등록한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존과목에 과목이 등록되어 있다. | |
| 사후 조건 | 기존 과목의 To do list table에 새로운 tuple이 추가 저장된다. | |
| 기본 흐름 | **B01** | 사용자는 기존과목에 등록 버튼을 누른다. |
| **B02** | 사용자는 과목을 선택한다. |
| **B03** | 사용자는 To do list 항목을 추가한다. |
| **B04** | 시스템은 To do List초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 과목이 기존 과목에 없을 경우 뒤로 가서 새로운 과목을 등록을 선택한다. |
| **E02** | To do List 항목을 잘못 기입했을 경우 초기화면에서 수정을 누른다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01->B04 |
| **SN003** | B01->B02->B03->E02->B04 |
|  |  |

* + - 1. ID: UC\_Reg\_002

|  |  |  |
| --- | --- | --- |
| 설명 | 새로운 과목을 등록하며 과목명을 설정한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 새로운 과목 등록 버튼을 누른다. | |
| 사후 조건 | 과목명이 입력된다. | |
| 기본 흐름 | **B01** | 사용자는 새로운 과목 등록 버튼을 누른다. |
| **B02** | 사용자는 과목명을 기입한다. |
|  |  |
|  |  |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 기존에 등록된 과목일 때 뒤로 가서 기존 과목에 등록을 선택한다. |
| **E02** | 과목명을 잘못 기입했을 경우 초기화면으로 돌아가서 수정 버튼을 눌러서 다시 입력한다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02 |
| **SN002** | B01->E01 |
| **SN003** | B01->B02->E02 |
|  |  |

* + - 1. ID: UC\_Reg\_003

|  |  |  |
| --- | --- | --- |
| 설명 | 새로운 과목을 등록하며 담당교수를 설정한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 새로운 과목 등록 버튼을 누른다. | |
| 사후 조건 | 담당교수가 입력된다. | |
| 기본 흐름 | **B01** | 사용자는 새로운 과목 등록 버튼을 누른다. |
| **B02** | 사용자는 담당교수를 기입한다. |
|  |  |
|  |  |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 기존에 등록된 과목일 때 뒤로 가서 기존 과목에 등록을 선택한다. |
| **E02** | 담당교수를 잘못 기입했을 경우 초기화면으로 돌아가서 수정 버튼을 눌러서 다시 입력한다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02 |
| **SN002** | B01->E01 |
| **SN003** | B01->B02->E02 |
|  |  |

* + - 1. ID: UC\_Reg\_004

|  |  |  |
| --- | --- | --- |
| 설명 | 새로운 과목을 등록하며 강의 요일 및 시간을 설정한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 새로운 과목 등록 버튼을 누른다. | |
| 사후 조건 | 강의 요일 및 시간이 입력된다. | |
| 기본 흐름 | **B01** | 사용자는 새로운 과목 등록 버튼을 누른다. |
| **B02** | 사용자는 강의 요일 및 시간을 기입한다. |
|  |  |
|  |  |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 기존에 등록된 과목일 때 뒤로 가서 기존 과목에 등록을 선택한다. |
| **E02** | 강의 요일 및 시간을 잘못 기입했을 경우 초기화면으로 돌아가서 수정 버튼을 눌러서 다시 입력한다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02 |
| **SN002** | B01->E01 |
| **SN003** | B01->B02->E02 |
|  |  |

* + - 1. ID: UC\_Reg\_005

|  |  |  |
| --- | --- | --- |
| 설명 | 새로운 과목을 등록하며 수강년도/학기를 설정한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 새로운 과목 등록 버튼을 누른다. | |
| 사후 조건 | 수강년도/학기가 입력된다. | |
| 기본 흐름 | **B01** | 사용자는 새로운 과목 등록 버튼을 누른다. |
| **B02** | 사용자는 수강년도/학기를 기입한다. |
| **B03** | 사용자는 입력버튼을 누른다. |
| **B04** | 시스템은 To do List의 초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 기존에 등록된 과목일 때 뒤로 가서 기존 과목에 등록을 선택한다. |
| **E02** | 수강년도/학기를 잘못 기입했을 경우 초기화면으로 돌아가서 수정 버튼을 눌러서 다시 입력한다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->E01 |
| **SN003** | B01->B02->E02 |
|  |  |

* + - 1. ID: UC\_Modify\_001

|  |  |  |
| --- | --- | --- |
| 설명 | 과목을 수정한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 과목 list가 존재하며 수정 버튼을 누른다. | |
| 사후 조건 | 기존 과목이 수정되어 DB에 저장된다. | |
| 기본 흐름 | **B01** | 사용자는 수정 버튼을 누른다. |
| **B02** | 사용자는 수정할 과목을 선택한다. |
| **B03** | 사용자는 해당 과목의 과목명, 담당 교수, 강의 요일/시간, 수강년도/학기 중 수정하고 싶은 항목을 수정하고 저장을 누른다. |
| **B04** | 시스템은 To do List의 초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 수정할 과목이 없는 경우 뒤로 가서 기존 과목에 등록을 선택한다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01 |
|  |  |
|  |  |

* + - 1. ID: UC\_Modify\_002

|  |  |  |
| --- | --- | --- |
| 설명 | To do List를 수정한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재하며 수정 버튼을 누른다. | |
| 사후 조건 | 기존 To do List가 수정되어 DB에 저장된다. | |
| 기본 흐름 | **B01** | 사용자는 수정 버튼을 누른다. |
| **B02** | 사용자는 수정할 To do List를 선택한다. |
| **B03** | 사용자는 해당 To do List의 내용을 수정하고 저장을 누른다. |
| **B04** | 시스템은 To do List의 초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 수정할 To do List가 없는 경우 뒤로 가서 등록을 선택한다. |
| **E02** | 과목을 수정하고 싶은 경우 뒤로 가서 수정버튼의 과목수정을 누른다. |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01 |
| **SN003** | B01->B02->E02 |
|  |  |

* + - 1. ID: UC\_Delete\_001

|  |  |  |
| --- | --- | --- |
| 설명 | 과목을 삭제한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 과목이 존재하며 삭제 버튼을 누른다. | |
| 사후 조건 | 기존 과목이 삭제되어 DB에서 제거된다. | |
| 기본 흐름 | **B01** | 사용자는 삭제 버튼을 누른다. |
| **B02** | 사용자는 삭제할 과목을 선택한다. |
| **B03** | 사용자는 해당 과목과 하위 To do List를 삭제하고 저장을 누른다. |
| **B04** | 시스템은 To do List의 초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 삭제할 과목이 없는 경우 뒤로 간다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01 |
|  |  |
|  |  |

* + - 1. ID: UC\_Delete\_002

|  |  |  |
| --- | --- | --- |
| 설명 | To do List를 삭제한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재하며 삭제 버튼을 누른다. | |
| 사후 조건 | 기존 To do List가 삭제되어 DB에 저장된다. | |
| 기본 흐름 | **B01** | 사용자는 삭제 버튼을 누른다. |
| **B02** | 사용자는 삭제할 To do List를 선택한다. |
| **B03** | 사용자는 해당 To do List의 내용을 삭제하고 저장을 누른다. |
| **B04** | 시스템은 To do List의 초기화면으로 접속한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 삭제할 To do List가 없는 경우 뒤로 간다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01 |
|  |  |
|  |  |

* + - 1. ID: UC\_Sort\_001

|  |  |  |
| --- | --- | --- |
| 설명 | 과목명 순으로 정렬한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재하며 정렬 버튼을 누른다. | |
| 사후 조건 | To do List가 과목명 순으로 정렬되어 출력된다. | |
| 기본 흐름 | **B01** | 사용자는 정렬 버튼을 누른다. |
| **B02** | 사용자는 과목명을 정렬 기준으로 선택한다. |
| **B03** | 시스템은 존재하는 To do List들을 과목의 가나다 순으로 정렬시킨다. |
| **B04** | 시스템은 To do List의 초기화면에 출력되는 목록이 정렬된 상태를 유지하게 한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 과목명이 같은 경우에는 해당 To do List 내용에서 문자열의 크기가 작은 순서대로 나열하게 한다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01->B03->B04 |
|  |  |
|  |  |

* + - 1. ID: UC\_Sort\_002

|  |  |  |
| --- | --- | --- |
| 설명 | 마감기한 순으로 정렬한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재하며 정렬 버튼을 누른다. | |
| 사후 조건 | To do List가 과목명 순으로 정렬되어 출력된다. | |
| 기본 흐름 | **B01** | 사용자는 정렬 버튼을 누른다. |
| **B02** | 사용자는 과목명을 정렬 기준으로 선택한다. |
| **B03** | 시스템은 존재하는 To do List들을 마감기한이 빠른 순으로 정렬시킨다. |
| **B04** | 시스템은 To do List의 초기화면에 출력되는 목록이 정렬된 상태를 유지하게 한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 마감기한이 같은 경우에는 과목명 순서대로 나열하게 한다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01->B03->B04 |
|  |  |
|  |  |

* + - 1. ID: UC\_Sort\_003

|  |  |  |
| --- | --- | --- |
| 설명 | 실제 마감일 순으로 정렬한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재하며 정렬 버튼을 누른다. | |
| 사후 조건 | To do List가 과목명 순으로 정렬되어 출력된다. | |
| 기본 흐름 | **B01** | 사용자는 정렬 버튼을 누른다. |
| **B02** | 사용자는 과목명을 정렬 기준으로 선택한다. |
| **B03** | 시스템은 존재하는 To do List들을 실제 마감일이 빠른 순으로 정렬시킨다. |
| **B04** | 시스템은 To do List의 초기화면에 출력되는 목록이 정렬된 상태를 유지하게 한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 실제마감일이 같은 경우에는 과목명 순서대로 나열하게 한다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01->B03->B04 |
|  |  |
|  |  |

* + - 1. ID: UC\_Sort\_004

|  |  |  |
| --- | --- | --- |
| 설명 | 완료여부 순으로 정렬한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재하며 정렬 버튼을 누른다. | |
| 사후 조건 | To do List가 과목명 순으로 정렬되어 출력된다. | |
| 기본 흐름 | **B01** | 사용자는 정렬 버튼을 누른다. |
| **B02** | 사용자는 과목명을 정렬 기준으로 선택한다. |
| **B03** | 시스템은 존재하는 To do List들을 완료가 된 것과 되지 않은 것 순서로 정렬시킨다. |
| **B04** | 시스템은 To do List의 초기화면에 출력되는 목록이 정렬된 상태를 유지하게 한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 완료여부가 같은 경우에는 과목명 순서대로 나열하게 한다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->E01->B03->B04 |
|  |  |
|  |  |

* + - 1. ID: UC\_Important-001

|  |  |  |
| --- | --- | --- |
| 설명 | 중요한 To do 항목을 표시한다. | |
| 관련 액터 | 사용자 | |
| 사전 조건 | 기존 To do List가 존재한다. | |
| 사후 조건 | To do List의 중요 표시가 되어 list가 출력된다. | |
| 기본 흐름 | **B01** | 시스템은 To do List 초기화면에서 list 목록을 출력한다. |
| **B02** | 사용자는 중요 표시할 To do List를 선택한다. |
| **B03** | 시스템은 해당 List의 내용을 빨간색으로 변경한다. |
| **B04** | 시스템은 To do List 초기화면에서 list 목록을 재출력한다. |
| 대안 흐름 |  |  |
|  |  |
| **예외 흐름** | **E01** | 중요 표시를 취소할 경우에는 해당 To do List를 다시 선택한다. |
|  |  |
|  |  |
|  |  |
|  |  |
| 시나리오 | **SN001** | B01->B02->B03->B04 |
| **SN002** | B01->B02->B03->E01->B04 |
|  |  |
|  |  |

* 1. 사용자 인터페이스 요구사항
     1. 화면 목록

|  |  |  |  |
| --- | --- | --- | --- |
| ID | **화면 명** | **관련**  **유스케이스 ID** | **설명** |
| SC001 | 로그인화면 | UC\_Login\_  001 | 아이디와 비밀번호를 입력하여 로그인 할 수 있도록 한다. |
| UC\_Login\_  002 | 로그인 화면에 회원 가입을 할 수 있는 버튼을 만든다. |
| SC002 | 과목 출력 화면 | UC\_Output\_  001 | 등록되어 있는 To do List 목록을 출력한다. |
| UC\_Reg\_001 | 기존 과목을 선택하여 To do List를 작성한다. |
| UC\_Modify\_  001 | 과목 리스트를 띄워 원하는 과목을 선택하여 수정할 수 있도록 한다. |
| UC\_Delete\_001 | 과목 리스트를 띄워 원하는 과목을 선택하여 삭제한다. |
| SC003 | 과목 등록 화면 | UC\_Reg\_002 | 과목 명을 부여하며 새로운 과목을 등록한다. |
| UC\_Reg\_003 | 담당 교수를 부여하며 새로운 과목을 등록한다. |
| UC\_Reg\_004 | 강의 요일 및 시간을 부여하며 새로운 과목을 등록한다. |
| UC\_Reg\_005 | 수강 년도 및 학기를 부여하며 새로운 과목을 등록한다. |
| SC004 | List 출력 화면 | UC\_Sort\_001 | 출력된 리스트를 과목 명 순으로 정렬한다. |
| UC\_Sort\_002 | 출력된 리스트를 마감 기한 순으로 정렬한다. |
| UC\_Sort\_003 | 출력된 리스트를 실제 마감일 순으로 정렬한다. |
| UC\_Sort\_004 | 출력된 리스트를 완료 여부 순으로 정렬한다. |
| UC\_Delete\_  002 | To do List를 띄워 원하는 리스트를 선택하여 삭제한다. |
| UC\_Important\_001 | 중요한 To do 항목을 별도로 표시하여 출력한다. |
| SC005 | List 등록 화면 | UC\_Modify\_  002 | To do List를 띄워 원하는 리스트를 선택하여 수정할 수 있도록 한다. |

* + 1. 화면 기술

|  |  |  |  |
| --- | --- | --- | --- |
| 화면 ID | SC001 | 화면 명 | 로그인 화면 |
| 화면 |  | | |
| 요구사항 | * 프로그램을 실행하면 로그인 화면이 나온다. * 회원가입 버튼을 누르면 가입 화면이 나온다. * 아이디와 비밀번호를 올바르게 입력하면 SC002로 넘어간다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 화면 ID | SC002 | 화면 명 | 과목 출력 화면 |
| 화면 |  | | |
| 요구사항 | * 과목 추가 버튼을 누르면 SC003으로 넘어간다. * 과목 삭제 버튼을 누르면 등록된 과목정보가 삭제된다. * 정렬 버튼을 누르면 선택한 기준으로 정렬한다. * 과목 정보를 누르면 SC004로 넘어간다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 화면 ID | SC003 | 화면 명 | 과목 등록 화면 |
| 화면 |  | | |
| 요구사항 | * 과목명, 교수명을 직접입력하고 나머지 옵션을 선택한 후 등록 버튼을 누르면 SC002에 과목 정보가 등록된다. * 종료버튼을 누르면 추가사항이 저장되지 않고 SC002로 돌아간다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 화면 ID | SC004 | 화면 명 | List 출력 화면 |
| 화면 |  | | |
| 요구사항 | * List작성 버튼을 누르면 SC005로 넘어간다. * List 삭제 버튼을 누르면 등록된 To do list가 삭제된다. * 중요도 체크 버튼을 누르면 등록된 To do list에 ★이 작성된다. * List 수정 버튼을 눌러 완료, 진행 여부를 수정할 수 있다. * 화면의 왼쪽 위에 기본적인 과목 정보가 출력된다. * 정렬 기준을 선택하면 기준에 따라 정렬된다. * 종료 버튼을 누르면 SC002로 돌아온다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 화면 ID | SC005 | 화면 명 | List 수정 화면 |
| 화면 |  | | |
| 요구사항 | * To do list에 내용을 적고 예정일과 완료여부를 선택한 후 등록 버튼을 누르면 SC004에 To do list의 내용이 등록된다. * 종료 버튼을 누르면 적은 내용이 저장되지 않고 SC004로 넘어간다. | | |

1. 비기능 요구사항

|  |  |
| --- | --- |
| 요구항목 | **설명** |
| 편리성 | 사용하는데 어렵지 않게 누구나 금방 익힐 수 있는 화면을 만든다. |
| 편리성 | 깔끔한 디자인을 추구한다 |
| 응답시간 | 시스템은 사용자의 요구에 1초만에 반응해야 한다 |
| 정확성 | 단어의 배치에 오해가 없도록 정확한 뜻을 전달 할 수 있게 한다 |
| 안정성 | 프로그램을 오래 사용해도 오류 없이 돌아가게 한다 |
| 보안 | 아이디 비밀번호를 5번 이상 틀리면 일정 기간 동안 접근이 불가능하게 한다 |