수업 일정 관리

소프트웨어 요구사항 명세서

작성자: 8조

20193098 최영은

20193051 김수진

20173127 안상욱

20113310 이희원

**목차**

[1. 문서 개요 4](#_Toc87920867)

[2. 시스템 개요 5](#_Toc87920868)

[2.1. 시스템 컨텍스트 5](#_Toc87920869)

[2.2. 입/출력 개요 5](#_Toc87920870)

[3. 소프트웨어 기능 요구사항 7](#_Toc87920871)

[3.1. 기능 요구사항 목록 7](#_Toc87920872)

[3.2. 기능 흐름 8](#_Toc87920873)

[3.3. 개념적 데이터 모델 11](#_Toc87920874)

[3.4. 기능 요구사항 정의 13](#_Toc87920875)

[3.4.1. U1 – 로그인 13](#_Toc87920876)

[3.4.2. U2 – 과목 추가 14](#_Toc87920877)

[3.4.3. U3 – 과목 삭제 15](#_Toc87920878)

[3.4.4. U4 – 시험과제 등록 16](#_Toc87920879)

[3.4.5. U5 – 시험과제 삭제 17](#_Toc87920880)

[3.4.6. U6 – 수업일정 조회 18](#_Toc87920881)

[3.4.7. U7 – 친구 목록 조회 19](#_Toc87920882)

[3.4.8. U8 – 친구 수업일정 조회 20](#_Toc87920883)

[3.4.9. U9 – 친구 추가 21](#_Toc87920884)

[3.4.10. U10 – 친구 삭제 23](#_Toc87920885)

[3.4.11. U11 - 알림 추가 23](#_Toc87920886)

[3.4.12. U12 - 알림 삭제 24](#_Toc87920887)

[4. 소프트웨어 품질 요구사항 26](#_Toc87920888)

[4.1. 품질 요구사항 목록 26](#_Toc87920889)

[4.2. 품질 요구사항 정의 27](#_Toc87920890)

[4.2.1. QR-001 – 수업일정 조회 속도 27](#_Toc87920891)

[4.2.2. QR-002 – 친구 목록 조회 속도 27](#_Toc87920892)

[4.2.3. QR-003 – 친구 수업일정 조회 속도 28](#_Toc87920893)

[4.2.4. QR-004 – 로그인 유지 28](#_Toc87920894)

[5. 인터페이스 요구사항 29](#_Toc87920895)

[6. 제약 사항 30](#_Toc87920896)

**변경 이력**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 버전 | 변경 내역 | 작성 | | 검토 | | 승인 | |
| 일자 | 담당자 | 일자 | 담당자 | 일자 | 담당자 |
| 1.0 | 명세서 작성 | 10/8 | 8조 | 10/8 | 8조 |  |  |
| 2.0 | 명세서 수정 | 11/15 | 8조 | 11/15 | 8조 |  |  |
|  |  |  |  |  |  |  |  |

# 문서 개요

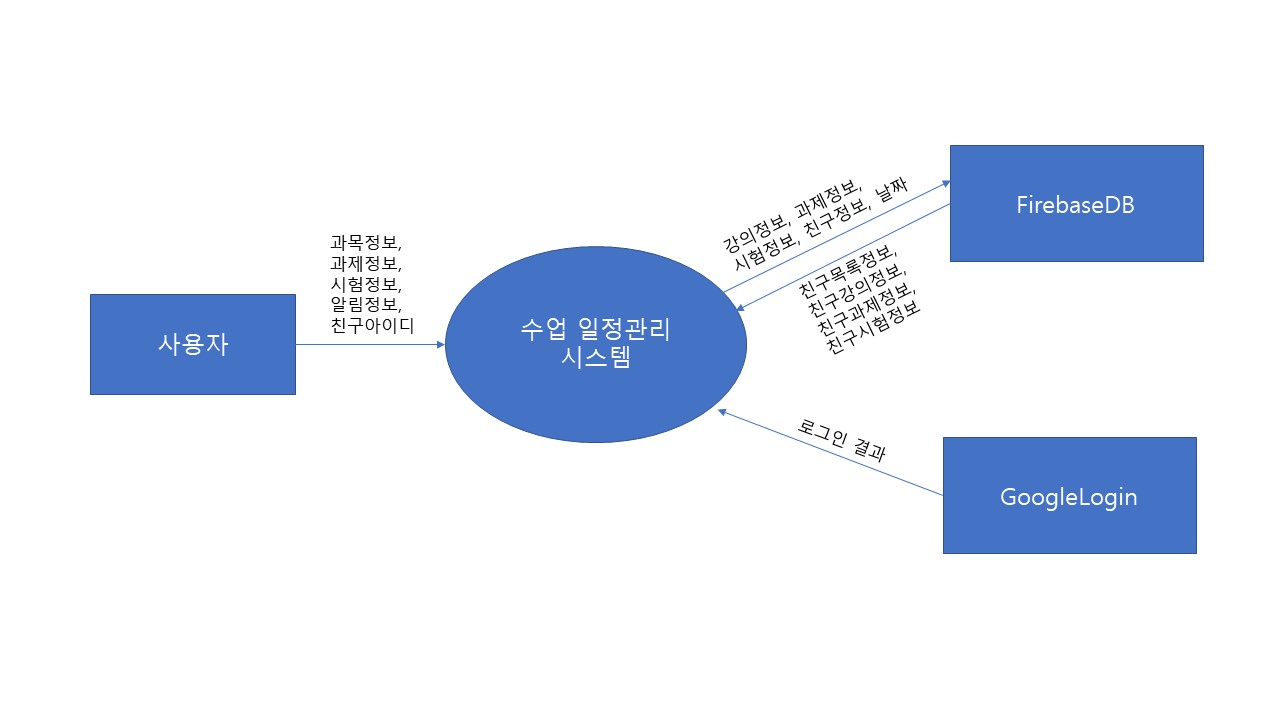
본 문서는 고객 및 이해관계자의 관점으로 기술된 고객 요구사항을 바탕으로 소프트웨어 제품 관점의 요구사항을 구체화한 결과로서 소프트웨어 요구사항을 기술한다.

본 문서의 주요 구성 항목은 다음과 같다.

* 시스템 개요: 시스템의 Context로서 개발 대상 시스템과 외부와의 입/출력을 정의한다.
* 소프트웨어 기능 요구사항: 소프트웨어 기능 요구사항을 정의한다.
* 소프트웨어 품질 요구사항: 소프트웨어 품질 요구사항을 정의한다.
* 인터페이스 요구사항: 시스템과 외부 및 내부 요소 간의 인터페이스에 대한 요구사항을 정의한다.
* 제약 사항: 표준, 법규, 규정 등과 같이 프로젝트 외부의 조직에 의해서 요구되는 제약을 정의한다.

# 시스템 개요

## 시스템 컨텍스트

개발 대상 시스템은 수업 일정관리 시스템이며, 외부 시스템으로는 로그인 시스템과 DB시스템이 있다. 시스템과 외부와의 입/출력은 다음과 같다.

## 입/출력 개요

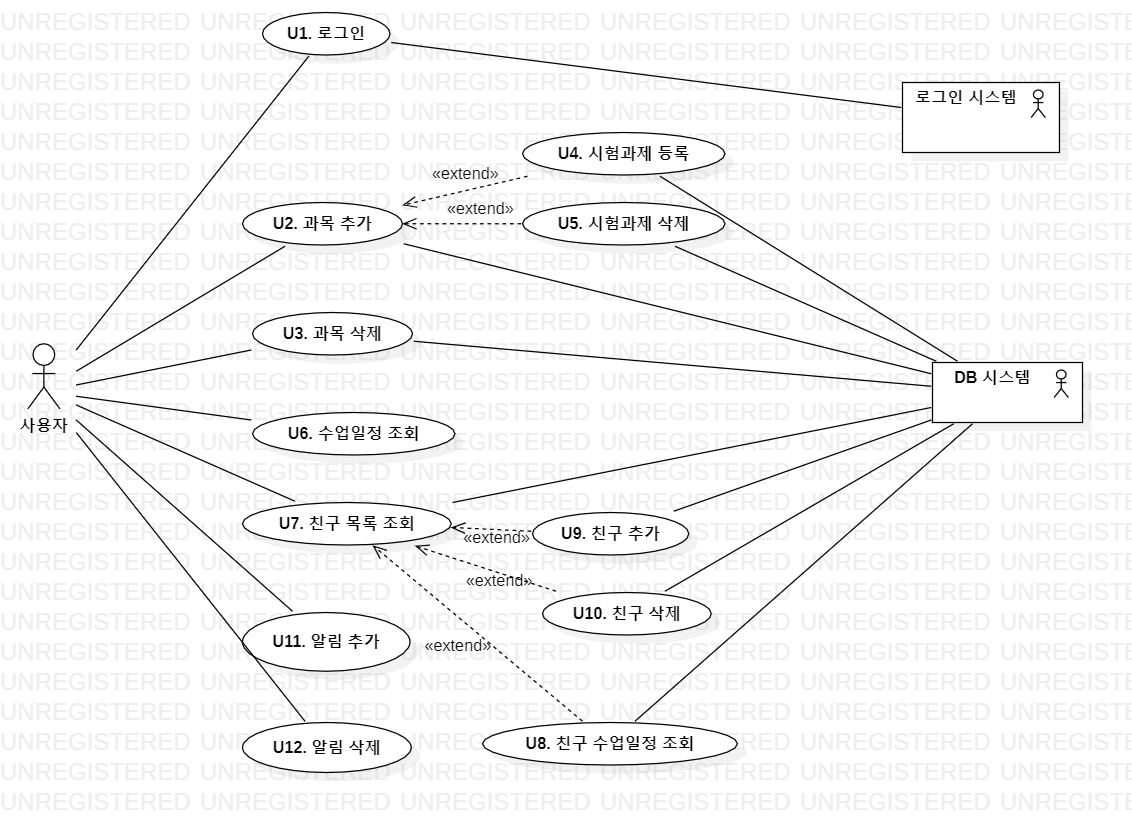
시스템의 주요 입/출력 데이터는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **데이터 이름** | **유형** | **설명** |
| 과목 정보 | 입력 | 사용자가 강의를 추가하기 위하여 입력하는 정보 (과목명, 갯수, 시작요일, 시작시간, 종료요일, 종료시간) |
| 알림 정보 | 입력 | 사용자가 알림을 추가하기 위하여 입력하는 정보(과목명, 시험알림시점, 과제알림시점, 동영상강의알림시점, 실시간강의알림시점) |
| 과제 정보 | 입력, 출력 | 사용자가 과제를 추가하기 위하여 입력하는 정보이자 firebaseDB에 저장하는 정보 (과목명, 과제명, 시작날짜, 시작시간, 종료날짜, 종료시간, 완료여부) |
| 시험 정보 | 입력, 출력 | 사용자가 시험을 추가하기 위하여 입력하는 정보이자 firebaseDB에 저장하는 정보 (과목명, 시험명, 날짜, 시간) |
| 친구 아이디 | 입력 | 사용자가 친구를 추가하기 위하여 입력하는 정보 (친구 ID) |
| 강의 정보 | 출력 | 사용자가 입력한 과목 정보를 이용하여 생성한 강의에 대한 정보로, firebaseDB에 저장하는 정보 (과목명, 강의명, 시작날짜, 시작시간, 종료날짜, 종료시간, 완료여부) |
| 친구 정보 | 출력 | 사용자가 firebaseDB에서 친구 신청을 하거나, 일정 정보를 받아오기 위하여 입력하는 친구의 ID, 또는 UID 정보 |
| 날짜 | 입력 | 사용자가 firebaseDB에서 친구의 강의정보, 과제정보, 시험정보를 얻고자 하는 날짜정보 |
| 친구 목록 정보 | 입력 | firebaseDB에서 가져온 사용자의 친구 목록 정보(친구ID, 친구 UID) |
| 친구 강의 정보 | 입력 | firebaseDB에서 가져온 친구의 강의 정보 (과목명, 강의명, 완료구분) |
| 친구 과제 정보 | 입력 | firebaseDB에서 가져온 친구의 과제 정보 (과목명, 과제명, 완료구분) |
| 친구 시험 정보 | 입력 | firebaseDB에서 가져온 친구의 시험 정보 (과목명, 시험명) |
| 로그인결과 | 입력 | GoogleLogin에서 로그인 한 결과 정보 (사용자 정보(ID, UID) |

# 소프트웨어 기능 요구사항

## 기능 요구사항 목록

각각의 유스케이스에 대한 명세서는 3.4 기능 요구사항 정의와 같다.

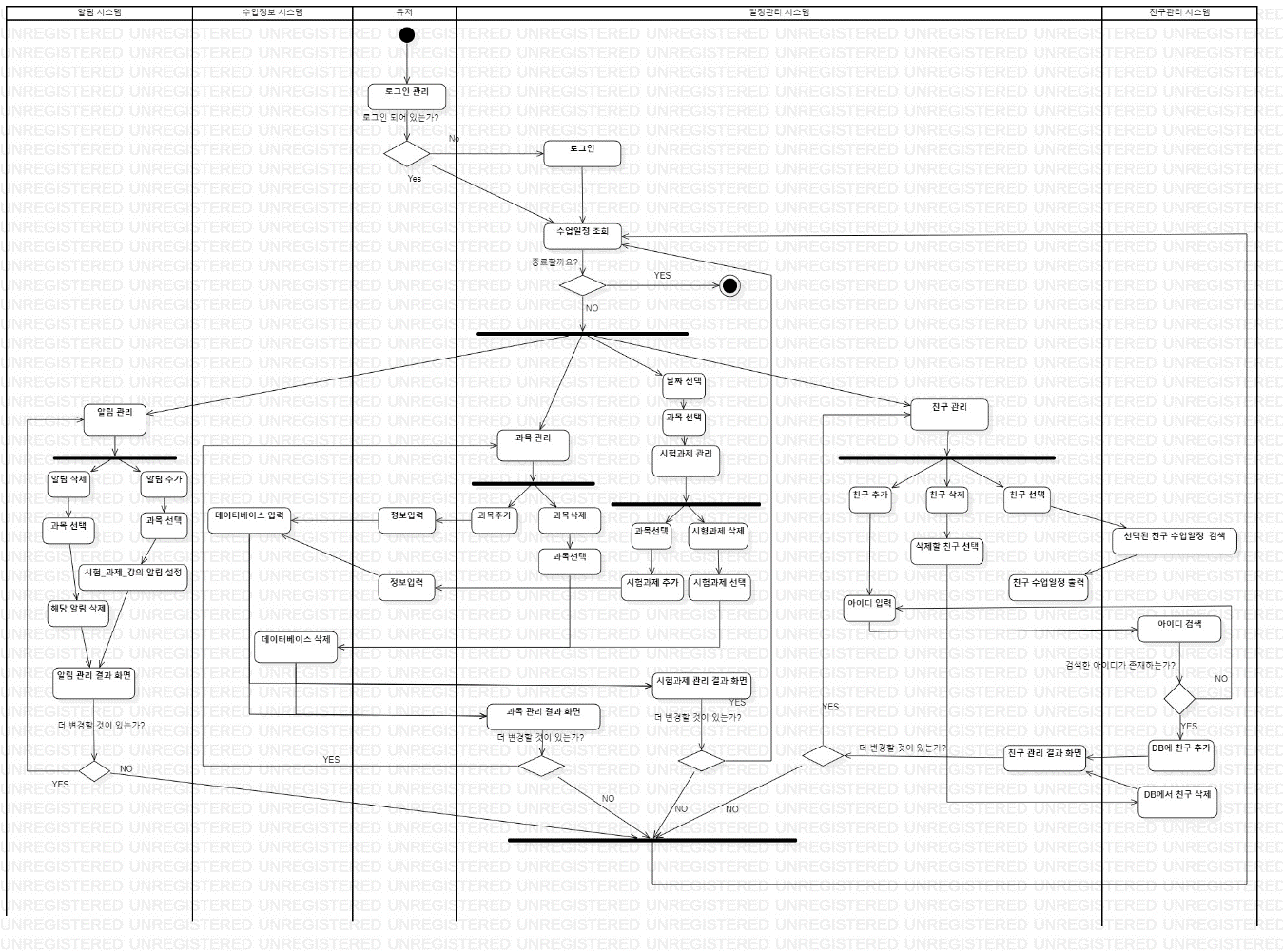


<전체 유즈케이스 다이어그램>

## 기능 흐름

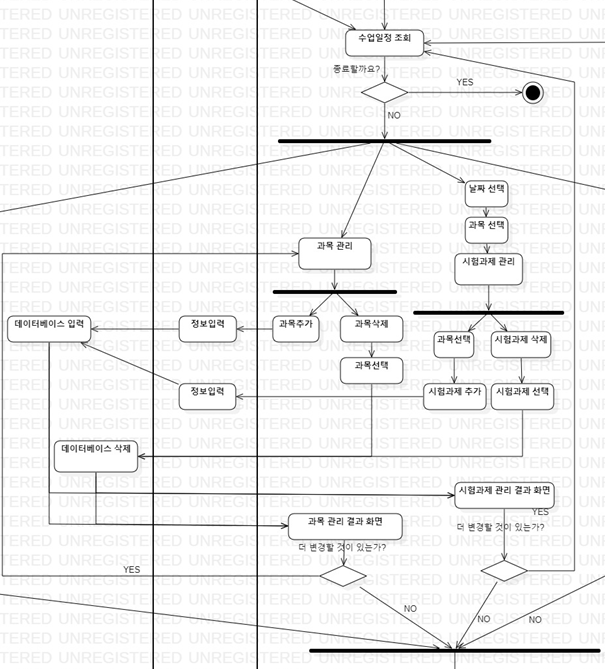
각 요구사항 간의 흐름은 다음의 액티비티 다이어그램과 같다.

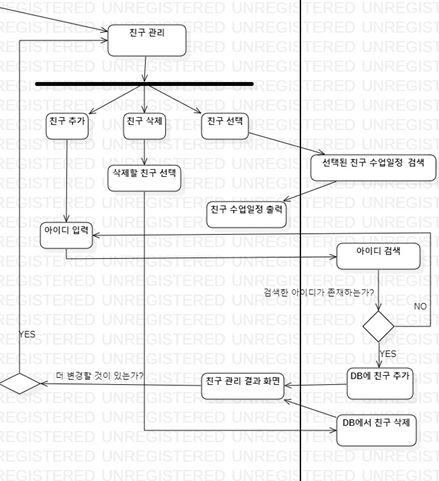
<전체 액티비티 다이어그램>



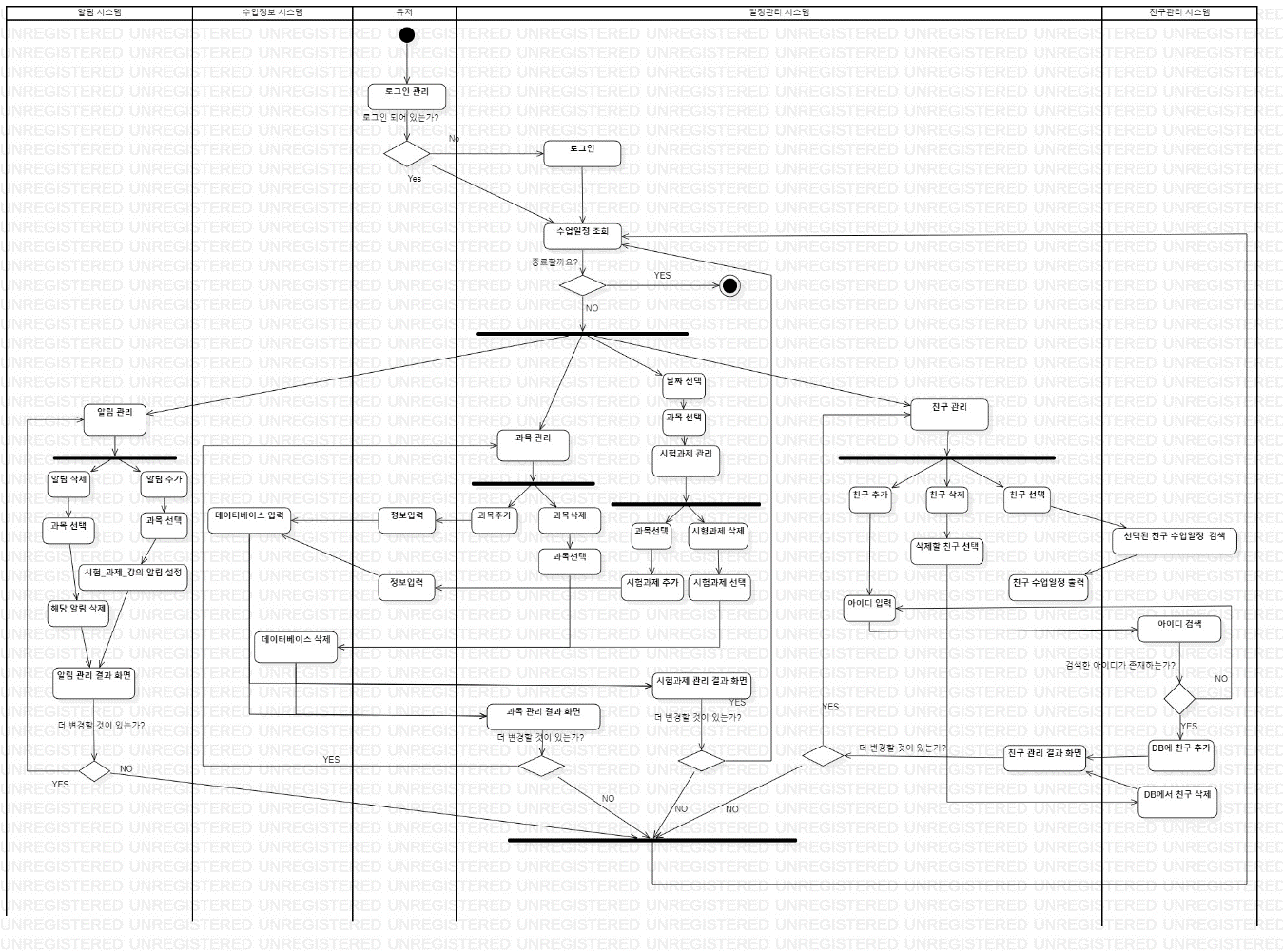
<알림 설정>

<과목 관리, 시험과제 관리>



<친구 관리>  


<친구 관리>



## 개념적 데이터 모델

시스템에서 저장/관리되는 데이터는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **데이터 명** | **속성** | **설명** |
| 강의 정보, 과제 정보 | 과목명 | 강의 또는 과제가 포함되는 과목 이름 |
| 강의명, 과제명 | 강의 또는 과제의 이름 |
| 시작 날짜 | 강의 또는 과제의 시작 날짜 |
| 시작 시간 | 강의 또는 과제의 시작 시간 |
| 종료 날짜 | 강의 또는 과제의 종료 날짜 |
| 종료 시간 | 강의 또는 과제의 종료 시간 |
| 완료 구분 | 강의 또는 과제를 완료하였는지 Boolean 정보 |
| 과목 정보 | 과목명 | 과목 이름 |
| 기본 강의 개수 | 매주 생성되는 강의의 개수 |
| 강의 생성 요일 | 매주 강의가 생성되는 요일 |
| 강의 생성 시간 | 매주 강의가 생성되는 시간 |
| 강의 종료 요일 | 매주 강의가 종료되는 요일 |
| 사용자 정보 | ID | 사용자가 로그인한 정보의 ID |
| UID | 사용자가 로그인한 정보의 UID |
| 친구 정보 | 서로 친구 목록 | 나와 친구인 아이디들의 목록 |
| 받은 친구 신청 목록 | 나를 친구로 추가하여 수락을 대기중인 아이디들의 목록 |
| 시험 정보 | 과목명 | 과목 이름 |
| 시험 날짜 | 시험의 날짜 |
| 시험 시간 | 시험의 시간 |
| 알림 정보 | 과목명 | 알림이 설정된 과목의 이름 |
| 시험 알림 시점 | 사용자가 추가한 시험 알림을 시험날짜 며칠 전에 받을지에 대한 정보 |
| 과제 알림 시점 | 사용자가 추가한 과제 알림을 과제마감 몇 시간 전에 받을지에 대한 정보 |
| 동영상 강의 알림 시점 | 사용자가 추가한 강의 알림을 출석기간 종료 몇 시간 전에 받을지에 대한 정보 |
| 실시간 강의 알림 시점 | 사용자가 추가한 강의 알림을 수업 시간 몇 분 전에 받을지에 대한 시간 정보 |

## 기능 요구사항 정의

3.1의 유스케이스 다이어그램에서 각각의 유스케이스에 대한 명세는 다음과 같다.

### U1 – 로그인

|  |  |
| --- | --- |
| **개요** | 사용자가 수업일정관리 시스템에 로그인을 하기 위한 기능이다. |
| **선행조건** | * 사용자가 수업일정관리 시스템에 접속한다. |
| **후행조건** | * 로그인 후에는 로그인 상태가 유지되어야 한다. |
| **기본**  **시나리오** | 1. 사용자가 수업일정관리 시스템에 접속한다. 2. 시스템은 사용자가 로그인을 한 결과에 대한 정보가 존재하는지 확인한다. 3. 만약, 로그인을 한 결과 정보가 없다면 시스템은 외부 로그인 시스템을 불러 로그인을 실행한다. 4. 로그인 결과 정보가 맞는 경우, firebaseDB의 사용자 목록에 사용자를 등록한다. 5. 시스템은 수업일정조회 화면으로 이동한다. |
| **대안**  **시나리오** | **로그인 결과 정보 존재 시나리오**   1. 만약 로그인을 한 결과 정보가 존재한다면, 시스템은 결과 정보에서 사용자 정보를 저장한다. 2. 수업일정조회 화면으로 이동한다. |

### U2 – 과목 추가

|  |  |
| --- | --- |
| **개요** | 사용자가 수업 일정 관리 시스템에 과목을 추가하기 위한 기능이다. 사용자가 과목을 추가하면, 과목의 매주 생성되는 강의 개수에 따라 시스템이 강의를 생성하고 수업일정 조회 화면에서 그것을 확인할 수 있다. |
| **선행조건** | * 사용자는 시스템에 로그인을 한 상태이어야 한다. |
| **후행조건** | * 시스템은 과목정보에 맞춰 강의를 시스템(sqlite)과 firebaseDB에 추가한다. * 추가된 과목은 과목 관리 화면의 과목 목록에서 확인 가능하다. |
| **기본**  **시나리오** | 1. 사용자는 과목 관리 화면에서 “과목 추가” 버튼을 선택한다. 2. 시스템은 과목 추가 창을 출력한다. 과목 추가 창은 과목 정보 입력을 위해서 과목명, 생성 년도, 학기, 강의 개수, 생성 요일, 생성 시간, 종료 요일, 종료 시간 입력 칸이 존재한다. 3. 사용자는 과목 정보를 입력하고 “취소”, “확인” 버튼 중에서 “확인” 버튼을 선택한다. 4. 과목이 중복되지 않은 경우, 시스템은 시스템(sqlite)에 과목정보를 기록하고 과목정보에 맞춰 강의를 시스템(sqlite)과 firebaseDB에 추가한다. 5. 시스템은 과목 추가 창을 닫고 성공 메시지를 보여준다. 갱신된 과목 관리 화면에는 추가한 과목이 과목 목록에 포함되어 출력된다. |
| **대안**  **시나리오** | **과목 추가 실패 시나리오**   1. 사용자가 과목 정보를 입력하고 “취소”, “확인” 버튼 중에서 “확인” 버튼을 선택한다. 2. 과목이 이미 등록되어 있는 경우, 시스템에 정보를 추가하지 않고 과목 추가 실패 메시지를 보여준다. |

### U3 – 과목 삭제

|  |  |
| --- | --- |
| **개요** | 사용자가 수업일정관리 시스템에서 과목을 삭제하기 위한 기능이다. 과목을 삭제하면 시스템(sqlite)에 추가 되어있던 해당 과목의 강의와 시험, 과제, 알림과 firebaseDB에 추가 되어있던 해당 과목의 강의와 시험, 과제들이 모두 제거된다. 그리고 시스템 알림이 삭제된다. |
| **선행조건** | * 사용자는 시스템에 로그인을 한 상태이어야 한다. |
| **후행조건** | * 일정에서 해당 과목의 강의, 시험, 과제, 알림이 모두 제거되어야 한다. * 삭제된 과목은 과목 관리 화면의 과목 목록에서 제거된다. |
| **기본**  **시나리오** | 1. 사용자는 과목 관리 화면에서 “과목 삭제” 버튼을 선택한다. 2. 시스템은 과목 관리 화면의 과목 목록 옆에 체크박스, 맨 아래에 “삭제” 버튼을 띄운다. 3. 사용자는 삭제할 과목을 선택한 후 “삭제” 버튼을 선택한다. 4. 시스템은 과목목록에서 해당과목을 제거하고 시스템(sqlite)과 firebaseDB에서 선택된 과목과 관련된 정보들을 제거한다. 5. 시스템은 삭제 성공 메시지를 보여주고 과목 관리 화면을 갱신한다. 갱신된 과목 관리 화면은 사용자가 선택했던 과목이 과목 목록에서 제거되어 나타난다. |

### U4 – 시험과제 등록

|  |  |
| --- | --- |
| **개요** | 사용자는 하나의 과목을 선택하여 해당 과목에 시험 또는 과제를 추가한다. 시험을 추가하는 경우에는 시험의 날짜 및 시간을 추가정보로, 과제를 추가하는 경우에는 시작시간, 종료시간을 추가정보로 저장한다. 만약 과목에 알림이 추가되어 있다면 해당 시험과제에 시스템 알림을 등록한다. |
| **선행조건** | * 사용자는 로그인을 한 상태여야 한다. * 사용자는 과목을 하나 이상 등록한 상태여야 한다. |
| **후행조건** | * 시험 또는 과제의 정보를 시스템(sqlite)과 firebaseDB에 저장한다. |
| **기본**  **시나리오** | 1. 사용자는 수업일정조회 화면에서 과목 하나를 선택하여 해당 과목의 “시험과제 추가” 버튼을 선택한다. 2. 시스템은 시험과제 정보 입력 화면을 출력한다. 시험과제 정보 입력 화면은 과목명과 시험, 과제 선택 버튼, 추가정보(시험: 날짜 및 시간, 과제: 시작시간, 종료시간) 입력창을 포함한다. 3. 사용자는 시험/과제 선택 및 추가정보를 입력하고, “등록” 버튼을 선택한다. 4. 시스템은 입력된 정보를 시스템(sqlite)과 firebaseDB에 기록하고, 시험 과제 등록 성공 메시지를 보여준다. 5. 수업일정조회 화면을 갱신하고 표시한다. |
| **대안**  **시나리오** | **시험과제 취소 시나리오**   1. 사용자가 “등록” 버튼이 아닌 “취소” 버튼을 누를 경우 2. 시스템은 입력된 정보를 저장하지 않고 수업일정 조회 화면으로 돌아간다. |

### U5 – 시험과제 삭제

|  |  |
| --- | --- |
| **개요** | 사용자는 자신이 생성한 시험 또는 과제에 대하여 삭제를 선택한다.  만약 선택한 시험과제의 과목에 알림이 추가되어 있다면 삭제하는 시험 또는 과제에 대한 시스템 알림을 삭제한다. |
| **선행조건** | * 사용자는 로그인을 한 상태여야 한다. * 사용자는 시험 또는 과제를 하나 이상 등록한 상태여야 한다. |
| **후행조건** | * 삭제를 진행한 시험 또는 과제의 정보를 시스템(sqlite)과 firebaseDB에서 삭제한다. |
| **기본**  **시나리오** | 1. 사용자는 수업일정 조회 화면에서 삭제할 시험 또는 과제의 삭제 버튼을 누른다. 2. 사용자가 “예” 버튼을 선택한 경우, 시스템은 선택된 시험 또는 과제를 시스템(sqlite)과 firebaseDB에서 삭제한다. 3. 시스템은 시험과제 삭제 성공 메시지를 보여준다. 4. 수업일정조회 화면을 갱신하고 표시한다. |
| **대안**  **시나리오** | **시험과제 삭제 취소 시나리오**   1. 사용자가 “아니오” 버튼을 선택한 경우, 시스템은 수업일정 조회 화면으로 돌아간다. |

### U6 – 수업일정 조회

|  |  |
| --- | --- |
| **개요** | 사용자는 조회하고자 하는 수업일정을 조회한다.  사용자가 선택한 날짜의 수업일정이 화면에 출력된다. |
| **선행조건** | * 사용자가 로그인한 상태이어야 한다. |
| **후행조건** | * 사용자가 선택한 날짜의 수업일정이 화면에 출력된다. * 추가된 과목의 일정은 수업일정으로 조회 가능한 상태가 된다. |
| **기본**  **시나리오** | 1. 시스템에서 수업일정 정보를 가져온다. 2. 시스템은 수업일정 조회 화면을 출력한다. 3. 사용자는 조회하고자 하는 날짜를 선택한다. 4. 시스템은 시스템(sqlite)에서 해당 날짜의 일정을 가져온다. 5. 시스템은 과목별로 수업일정을 화면에 보여준다. |
| **대안**  **시나리오** | **일정완료 표시 시나리오**   1. 사용자는 수업일정 조회 화면에서 일정의 완료 여부를 체크한다. 2. 시스템은 시스템(sqlite)과 firebaseDB의 일정 데이터를 업데이트 한다. 3. 시스템은 수업일정 조회 화면을 갱신하고 완료한 일정에 체크 표시를 한다. |

### U7 – 친구 목록 조회

|  |  |
| --- | --- |
| **개요** | 사용자가 받은 친구신청을 수락하거나 사용자가 보낸 친구신청을 상대가 수락한 경우의 친구들을 “서로 친구 목록” 창에서 확인하고, 사용자에게 친구신청을 보낸 사용자들의 이름을 “받은 친구 신청” 창에서 확인하는 기능이다. |
| **선행조건** | * 사용자는 로그인을 한 상태이어야 한다. |
| **후행조건** | * 친구 관리 화면에서 서로 친구 목록 또는 받은 친구 신청 목록을 출력한다. |
| **기본**  **시나리오** | 1. 사용자는 일정조회 화면에서 “친구 관리” 버튼을 선택한다. 2. 시스템은 친구 관리 화면을 출력한다. 친구 관리 화면은 서로 친구 목록창과 받은 친구 신청 창을 선택하는 버튼과, 목록을 출력하는 화면, 친구 추가, 친구 삭제 버튼을 가진다. 3. 친구 관리 화면에 접속하거나, 친구 목록 버튼을 누르는 경우, 시스템은 firebaseDB에 기록된 서로 친구 목록 정보를 받아온다. 4. 시스템은 서로 친구 목록을 화면에 출력한다. 친구 목록은 친구의 아이디의 목록이다. 5. 친구 아이디를 선택하면 친구 일정 조회 화면으로 이동한다. |
| **대안**  **시나리오** | **친구 신청 목록 조회 시나리오**   1. 사용자는 친구 관리 화면에서 “받은 친구 신청” 버튼을 선택한다. 2. 시스템은 firebaseDB에서 받은 친구 신청 정보를 가져온다. 3. 시스템은 받은 친구 신청을 화면에 출력한다. 목록은 친구 이름과 “수락” 버튼을 포함한다. 4. 친구 이름 옆의 “수락” 버튼을 누르면 시스템은 firebaseDB에 정보를 업데이트하고, 시스템의 서로 친구 목록에 추가한다. 5. 시스템은 수락 성공 메시지를 보여주고, 받은 친구 신청 목록을 갱신하여 출력한다. 갱신된 화면에서는 해당 친구가 제거된다. |

### U8 – 친구 수업일정 조회

|  |  |
| --- | --- |
| **개요** | 사용자는 친구 목록에서 친구를 선택하면 해당 친구의 일정을 firebaseDB에서 불러와 출력하는 기능이다. |
| **선행조건** | * 사용자는 로그인 한 상태이어야 한다. * 수업 일정을 보려고 하는 친구와 서로 친구 상태이어야 한다. |
| **후행조건** | * 사용자가 선택한 날짜의 친구 수업일정을 화면에 출력한다. |
| **기본**  **시나리오** | 1.    사용자는 친구 관리 화면의 서로 친구 목록에서 친구를 선택한다.  2.   시스템은 firebaseDB에서 친구 아이디를 검색하여 친구의 수업일정 정보를 받아온다.  3. 친구 수업일정 조회 화면을 출력한다.  4. 사용자는 조회하고자 하는 날짜를 선택한다.  5. 시스템은 과목별로 수업일정을 화면에 보여준다. |

### U9 – 친구 추가

|  |  |
| --- | --- |
| **개요** | 사용자가 친구로 등록하고자 하는 다른 사용자의 아이디를 입력하여 친구신청을 보내는 기능이다. 시스템은 입력된 아이디가 firebaseDB에 등록된 사용자인지 확인하고, 등록된 사용자라면 firebaseDB에 친구 신청을 기록한다. |
| **선행조건** | * 사용자는 로그인을 한 상태이어야 한다. |
| **후행조건** | * firebaseDB에 친구신청을 기록한다. |
| **기본**  **시나리오** | 1. 사용자는 친구 관리 화면에서 친구 추가 버튼을 선택한다. 2. 시스템은 친구 추가 창을 출력하고, 아이디 입력을 요구한다. 3. 사용자는 친구로 등록하고자 하는 다른 사용자의 아이디를 입력하고 “확인” 버튼을 누른다. 4. 시스템은 firebsaeDB에 친구 아이디가 사용자로 등록되어 있는지 확인한다. 5. 해당 아이디가 존재할 시, firebaseDB에 친구신청을 기록한다. 6. 시스템은 친구 추가 창을 닫고 친구 신청 성공 메시지를 보여준다. |
| **대안**  **시나리오** | **친구 추가 실패 시나리오**   1. 입력된 아이디가 사용자로 등록되어 있지 않은 경우, ‘대상을 찾을 수 없습니다.’ 라는 메시지를 보여준다. 2. 시스템은 친구 추가 창의 아이디 입력창을 출력하고, 아이디 입력을 다시 요구한다.   **친구 추가 취소 시나리오**   1. 사용자가 “취소” 버튼을 누른 경우, 친구 추가 창을 닫고 친구 관리 화면으로 돌아간다. |

### U10 – 친구 삭제

|  |  |
| --- | --- |
| **개요** | 사용자가 친구 목록에서 친구를 선택하여 삭제하는 기능이다. 시스템은 firebaseDB에서 정보를 삭제하고 친구 관리 화면을 갱신한다. |
| **선행**  **조건** | * 사용자는 로그인을 한 상태이어야 한다. * 친구가 한 명 이상 존재해야 한다. |
| **후행**  **조건** | * 친구 목록에서 선택한 친구의 이름이 삭제된다. |
| **기본**  **시나리오** | 1. 사용자는 “친구 삭제” 버튼을 누른다. 2. 시스템은 친구 목록에 체크박스를 띄운 화면을 출력한다. 3. 사용자는 삭제할 친구들을 선택하여 “삭제” 버튼을 선택한다. 4. 시스템은 firebaseDB에서 선택된 친구들을 삭제한다. 5. 친구 관리 화면의 친구 목록을 갱신하여 보여준다. |

### U11 - 알림 추가

|  |  |
| --- | --- |
| **개요** | 사용자가 과목별로 알림이 언제 울릴지 설정하는 기능이다. 시험, 과제, 강의에 대한 알림을 언제 울릴 지 선택할 수 있고, 알림을 추가할 시 시스템(sqlite)에 정보를 기록한다. |
| **선행**  **조건** | * 사용자는 시스템에 로그인을 한 상태이어야 한다. |
| **후행**  **조건** | * 시스템은 시스템(sqlite)에 알림 정보를 기록한다. * 추가된 알림은 알림 목록에서 확인 가능하다. |
| **기본**  **시나리오** | 1. 사용자는 수업일정 조회 화면에서 “알림” 버튼을 선택한다. 2. 시스템은 알림 관리 화면을 보여준다. 알림 관리 화면은 알림이 설정된 과목 목록과 알림 추가, 알림 삭제 버튼을 가진다. 3. 사용자는 “알림 추가” 버튼을 선택한다. 4. 시스템은 알림 추가 창을 띄운다. 알림 추가 창에서는 과목명, 시험 알림 시점, 과제 알림 시점, 강의 알림 시점, 강의 유형을 선택할 수 있다. 5. 사용자는 정보를 입력하고 “확인” 버튼을 선택한다. 6. 시스템은 시스템(sqlite)에 알림 정보를 기록하고 정보에 맞는 시스템 알림을 설정한다, 7. 알림 추가 창을 닫고 알림 추가 성공 메시지를 보여준다. 8. 알림 관리 화면이 갱신되어 출력되고 추가한 알림이 알림 목록에 포함되어 표시된다. |
| **대안**  **시나리오** | **알림 추가 취소 시나리오**   1. 사용자가 “취소” 버튼을 누른 경우, 알림 관리 화면으로 돌아간다. |

### U12 - 알림 삭제

|  |  |
| --- | --- |
| **개요** | 사용자가 생성되어 있는 알림을 삭제하는 기능이다. |
| **선행**  **조건** | * 사용자는 시스템에 로그인을 한 상태이어야 한다. * 사용자는 알림을 하나 이상 등록한 상태여야 한다. |
| **후행**  **조건** | * 시스템은 시스템(sqlite)에 기록된 알림 정보를 삭제한다. * 삭제된 알림은 알림 목록에서 제거된다. |
| **기본**  **시나리오** | 1. 사용자는 알림 관리 화면에서 “알림 삭제” 버튼을 선택한다. 2. 시스템은 알림 관리 화면의 과목별 알림 목록 옆에 체크박스, 맨 아래에 “삭제” 버튼을 띄운다. 3. 사용자는 삭제할 알림을 선택한 후 화면 아래의 “삭제” 버튼을 선택한다. 4. 체크된 알림에 대하여 시스템은 시스템(sqlite)에 기록된 알림 정보를 삭제하고, 설정된 시스템 알림을 제거한다. 5. 알림 삭제 성공 메시지를 보여준다. 6. 시스템은 알림 관리 화면을 갱신한다. 갱신된 알림 관리 화면은 사용자가 선택했던 알림이 과목별 알림 목록에서 제거되어 나타난다. |

# 소프트웨어 품질 요구사항

## 품질 요구사항 목록

소프트웨어에 요구되는 품질 요구사항의 목록은 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **품질 요구사항 ID** | **유형** | **품질 요구사항 명** | **우선순위** | | | **관련 고객**  **요구사항 ID** |
| **중요도** | **난이도** | **우선순위** |
| QR-001 | 성능 | 수업일정 조회 속도 | 5 | 3 | 2 |  |
| QR-002 | 성능 | 친구목록 조회 속도 | 3 | 5 | 3 |  |
| QR-003 | 성능 | 친구 수업일정 조회 속도 | 3 | 5 | 4 |  |
| QR-004 | 편의성 | 로그인  유지 | 5 | 2 | 1 |  |

## 품질 요구사항 정의

4.1의 품질 요구사항 목록의 각각의 요구사항에 대한 정의는 다음과 같다.

### QR-001 – 수업일정 조회 속도

|  |  |
| --- | --- |
| **항목** | **내용** |
| 설명 | 선택한 날짜의 수업일정을 빠르게 불러오는 성능에 대한 요구사항 |
| 전제 조건 | 전체 일정 수: 500개  검색 대상 일정 수: 20건  조회 기준: 선택한 날짜 |
| 충족 기준 | 날짜 선택 후 조회 결과가 1초 이내에 화면에 출력 |

### QR-002 – 친구 목록 조회 속도

|  |  |
| --- | --- |
| **항목** | **내용** |
| 설명 | 친구 목록을 선택했을 때 저장된 친구 목록을 빠르게 불러오는 성능에 대한 요구사항 |
| 전제 조건 | 전체 친구 수: 100 |
| 충족 기준 | 친구 목록 선택 후에 조회 결과가 3초 이내에 화면에 출력 |

### QR-003 – 친구 수업일정 조회 속도

|  |  |
| --- | --- |
| **항목** | **내용** |
| 설명 | 친구 목록에서 선택한 친구의 수업일정을 빠르게 불러오는 성능에 대한 요구사항 |
| 전제 조건 | 전체 일정 수: 500개  검색 대상 일정 수: 20건  조회 기준: 선택한 날짜 |
| 충족 기준 | 날짜 선택 후 조회 결과가 3초 이내에 화면에 출력 |

### QR-004 – 로그인 유지

|  |  |
| --- | --- |
| **항목** | **내용** |
| 설명 | 시스템에 재 접속했을 때 로그인이 유지되어야 하는 편의성에 대한 요구사항 |
| 전제 조건 | 로그인에 적어도 한 번은 성공해야 한다. |
| 충족 기준 | 시스템에 다시 접속했을 때 로그인이 유지되어 있어야 한다. |

# 인터페이스 요구사항

시스템과 외부 및 내부 요소 간의 인터페이스에 대한 요구사항은 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **인터페이스**  **요구사항 ID** | **인터페이스**  **요구사항** | **인터페이스**  **요구사항 설명** | **우선순위** | | | **관련고객**  **요구사항ID** |
| **중요도** | **난이도** | **우선순위** |
| IR-001 | 과목 당 일정 개수 제한 | 한 과목당 입력할 수 있는 일정의 개수는 50개 이하로 한다. | 2 | 2 | 3 |  |
| IR-002 | 최대 친구 수 제한 | 계정 당 등록할 수 있는 친구의 수를 50명 이하로 한다. | 2 | 2 | 3 |  |
| IR-003 | DB 연결 | FirebaseDB  사용 | 5 | 3 | 2 |  |
| IR-004 | SNS로그인 연동 | 구글 로그인 API를 사용 | 5 | 3 | 1 |  |

# 제약 사항

프로젝트 외부의 조직에 의해서 요구되는 제약 사항들은 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **제약사항ID** | **제약사항** | **제약사항**  **설명** | **우선순위** | | | **관련고객**  **요구사항**  **ID** |
| **중요도** | **난이도** | **우선순위** |
| CR-001 | 구글 로그인 API지침 | 패키지 이름과 SHA-1 서명 인증서 디지털 지문을 추가해야 한다. | 5 | 2 | 1 |  |
| CR-002 | 사용자 정보 보호 | 개인정보 보호법에 따라서 사용자 인적사항 정보 보호 필요 | 5 | 3 | 1 |  |
| CR-003 | Firebase 제약사항 | 동시 연결 수는 200,000개 미만이어야 한다. | 5 | 2 | 2 |  |