

**변 경 이 력**

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **일자** | **변경 내역** | **작 성 자** |
| 1.0.0 | 2024.10.09 | 1차 작성 | 일조매 |
| 1.0.1 | 2024.10.13 | use case 명세서 추가  프로토타입 UI 추가 | 일조매 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[***1.***](#_30j0zll) ***소개 4***

[**1.1**](#_1fob9te) **목적 4**

[**1.2**](#_3znysh7) **프로젝트 범위 (Scope) 4**

[**1.3**](#_2et92p0) **요구기능 추출 방법 (Methods) 4**

[**1.4**](#_tyjcwt) **정의 및 약어 4**

[***2.***](#_3dy6vkm) ***제약사항 (Constraints)*** 5

[***3.***](#_1t3h5sf) ***사용자 분석*** 5

[**3.1 액터 정의**](#_4d34og8) 5

[***4.***](#_2s8eyo1) ***요구사항 분석*** 7

[**4.1**](#_17dp8vu) **기능 요구사항 (Functional Requirements)** 7

[**4.2**](#_3rdcrjn) **Use case 목록** 9

[4.2.1](#_26in1rg) Use case Diagram 10

[4.2.2](#_lnxbz9) Use case 명세 11

[***5.***](#_35nkun2) ***비기능 요구사항 (Non-Functional Requirements) 18***

[***6.***](#_1ksv4uv) ***SRS Standards 를 참고하여 필요한 사항을 자유롭게 추가 19***

***6.1*** [***Low Quality 프로토타입***](#_7jzrky2ri281)***....19***

# **소개**

## 목적

협동심을 중시하는 최근 사회에서, 팀원과 함께 협력하여 과제를 수행하는 것은 매우 가치 있는 활동이 되었다. 이에 따라 팀을 꾸려 수행하는 과제가 점점 늘어나는 추세이다. 하지만 시간 조율, 직무 유기 등, 조별 과제 진행을 방해하는 상황이 지속적으로 발생하고 있다.

본 프로젝트는 조별 과제를 진행하면서 팀원들 간 협력을 더 쉽게 하는 데에 초점을 두고 있다. 조별 과제를 경험하면서 다른 학생들이 주로 언급했던 문제점인 시간 조율부터, 현재 누가 어떤 일을 언제까지 수행하고 있는지 한눈에 볼 수 있게 하여, 고질적인 직무 유기 문제를 해결할 수 있도록 한다. 이외에도 *“별로 친하지 않은 팀원에게 알림을 보내고 싶다.”* , *“전반적으로 과제가 어떻게 진행되고 있는지 한눈에 보고 싶다.”* 등 조별 과제를 수행하는 데 있어 불편했던 점을 해결할 수 있는 시스템을 개발하는 것을 목표로 한다.

## 프로젝트 범위 (Scope)

개발하는 프로젝트의 범위는 다음과 같다.

1. 로그인 기능
   1. 계정 생성
   2. 로그인 시 자신의 팀 프로젝트 불러오기.
2. 팀 프로젝트 생성
   1. 팀 프로젝트 생성
   2. 팀원 초대
3. 시간 조율
   1. 각 팀원은 자신의 ‘이번 주 불가능한 회의 시간’을 기입.
   2. 팀장은 종합된 ‘팀 프로젝트 스케줄’을 기반으로 ‘회의 시간’을 설정.
4. Task 관리
   1. Task 별 마감 기한 설정
   2. 날짜 별 Task 확인
   3. Task 진행 중/ 완료 상태 확인
5. ‘독촉 알림’ 전송 가능
   1. 진행 중인 Task에 대해, 해당 Task를 수행해달라고 요청
6. 팀 프로젝트 저장 데이터 베이스
   1. 프로젝트 단위로 클라우드에 저장

※회의 자체를 개발한 시스템 내에서 하는 기능은 제공하지 않을 예정이다.

※로그인에 대한 보안 문제는 이번 개발 범위에 포함하지 않았다.

## 요구기능 추출 방법 (Methods)

* 문제 확인

“조별과제를 돕는 앱”이 제공해야 할 Use Case를 확인하기 위해서, 1차적으로 브레인스토밍을 통해, 팀원 각자가 “조별과제에서 겪었던 어려움”을 확인했다. 이어서, 브레인스토밍으로 나온 각 문제를 온라인 미팅을 통해 비슷한 문제끼리 그룹화 한 다음, 몇 가지로 요약된 문제를 분석했다.

* 요구사항 분석

각 문제를 겪은 사람과, 그 문제를 해결하기 위해 개입하는 외부 시스템을 액터로 정의했다. 다음으로, 그리고, 우리가 요약한 문제들과, 그 문제들을 해결하기 위한 기능을 각 Use case로 만들었다. Use case를 효율적으로 해결할 수 있는 시스템의 기능적 요구사항을 정리하기 위해서, 직접 앱 화면 배치를 논의하는 과정에서 다른 방법으로 해결할 수 있거나, 필요 없다고 판단한 Use case를 삭제했다. 정리된 Use case를 바탕으로, 각 액터와 Use case 간의 관계를 도식화하기 위해 Use case Diagram을 작성하고, 구체화하기 위해 상세한 내용을 팀원 간에 논의하여 추가했다.

## 정의 및 약어

* 팀원: 팀장을 포함한 팀 프로젝트 참여자 전원을 의미한다.
* Task: 각 팀원이 배정받는 업무를 말한다. 주로 팀 자체의 목표보다는 팀원 개인의 목표이다.
* 캘린더: 각 Task 수행 기간, 그리고 확정된 회의의 시작 시각을 표기한 달력이다.
* 팀 프로젝트 스케줄: 팀원 모두가 가능한 회의 시간대만 보여주는 일주일 시간표. 이를 확인하여

모두가 가능한 회의 시간을 쉽게 설정할 수 있다.

* 독촉 알림: Task를 완료하지 않은 팀원에게 팀 프로젝트 수행을 독촉하는 알림을 보낼 수 있다.

# **제약사항 (Constraints)**

팀 프로젝트 진행 상황, 스케쥴 관리 등등을 위해 웹 브라우저와 모바일 앱 모두 구현해야하는 ‘크로스플랫폼’이 적절할 것으로 예상된다. 이를 위해 웹과 앱의 효율적인 프론트엔드 개발을 위해 자바스크립트를 기반으로 한 React, React Native로 구현할 예정이다. 백엔드 또한 RESTful API가 가능하면서 React와의 호환성을 위해 Node.js로 구현할 예정이다.

데이터베이스는 mySQL로 구현할 예정이며, 프론트엔드, 백엔드, 데이터베이스 구현을 위한 서버는 가상 서버를 사용할 예정이며, Amazon Web Services의 EC2 인스턴스를 사용할 것이다.

# **사용자 분석**

## 3.1 액터 정의

| 액터 ID | 액터 명 | 설명 |
| --- | --- | --- |
| P01 | 팀장 | 팀 프로젝트를 만든 사용자   * 팀 프로젝트 생성 및 수정   + 팀 프로젝트 생성   + 팀 프로젝트 삭제   + 캘린더에 프로젝트 기한 설정   + 팀 프로젝트 스케줄 시간 범위 요청 * 팀원 관리   + 팀장 위임   + 팀원 추방(하고 이름 빼기) * 팀 프로젝트 수행   + 미팅 일정 등록 * Task 관리   + Task 등록   + Task 마감 시한 연장 |
| P02 | 팀원  (팀장을 포함한 모든 참가자) | * 팀 프로젝트 인원 관련   + 팀 프로젝트 참여   + 팀 프로젝트 초대 * 팀 프로젝트 수행 관련   + 팀 프로젝트 스케줄 시간 범위 요청 시 자신의 가능 시간대 작성 * 팀 프로젝트 수행   + 자신의 팀 프로젝트 진행 상황(진행중, 완료) 표기 * Task 관련   + 독촉 알림 전송 가능 |

| 액터 ID | 액터 명 | 설명 |
| --- | --- | --- |
| PHONE01 | 스마트폰 기본 권한 | 팀 프로젝트 앱은 알림, 전화 등을 위해 스마트폰의 권한을 받아야 함.   * 시스템에서 특정 인원에게 독촉 알림을 전송하면, 해당 팀원에게 알림이 전송됨. * 디데이를 설정하면, 알림창에 알림이 생김. |
| DB01 | 유저 데이터베이스 | 유저 아이디, 비밀번호, 닉네임, 팀 프로젝트 배열이 담겨있는 데이터베이스.  새로운 계정이 생성될 때마다 레코드가 추가됨. |
| DB02 | 팀 프로젝트  데이터베이스 | 생성된 모든 팀 프로젝트가 담겨있는 데이터베이스.  하나의 팀 프로젝트에는   * 팀 프로젝트 번호 * 팀장 * 팀 프로젝트 참여하는 인원 * 스케줄 * 캘린더 * Task정보   이며, 사용자가 팀 프로젝트를 생성할 때마다 레코드가 추가됨. |
| DB03 | Task 데이터베이스 | 생성된 모든 Task가 담겨있는 데이터베이스.  하나의 Task에 포함되는 내용   * 받은 인원의 아이디(여러 명일 경우 #으로 구분) * 팀 프로젝트 번호 * Task 내용(string) * 완료상태 * 마감기한   이며, 사용자가 팀 프로젝트에서 Task 생성할 때마다 레코드가 추가됨. 다만 팀 프로젝트 별로 Task 테이블이 생성되는 것은 아님. |

# **요구사항 분석**

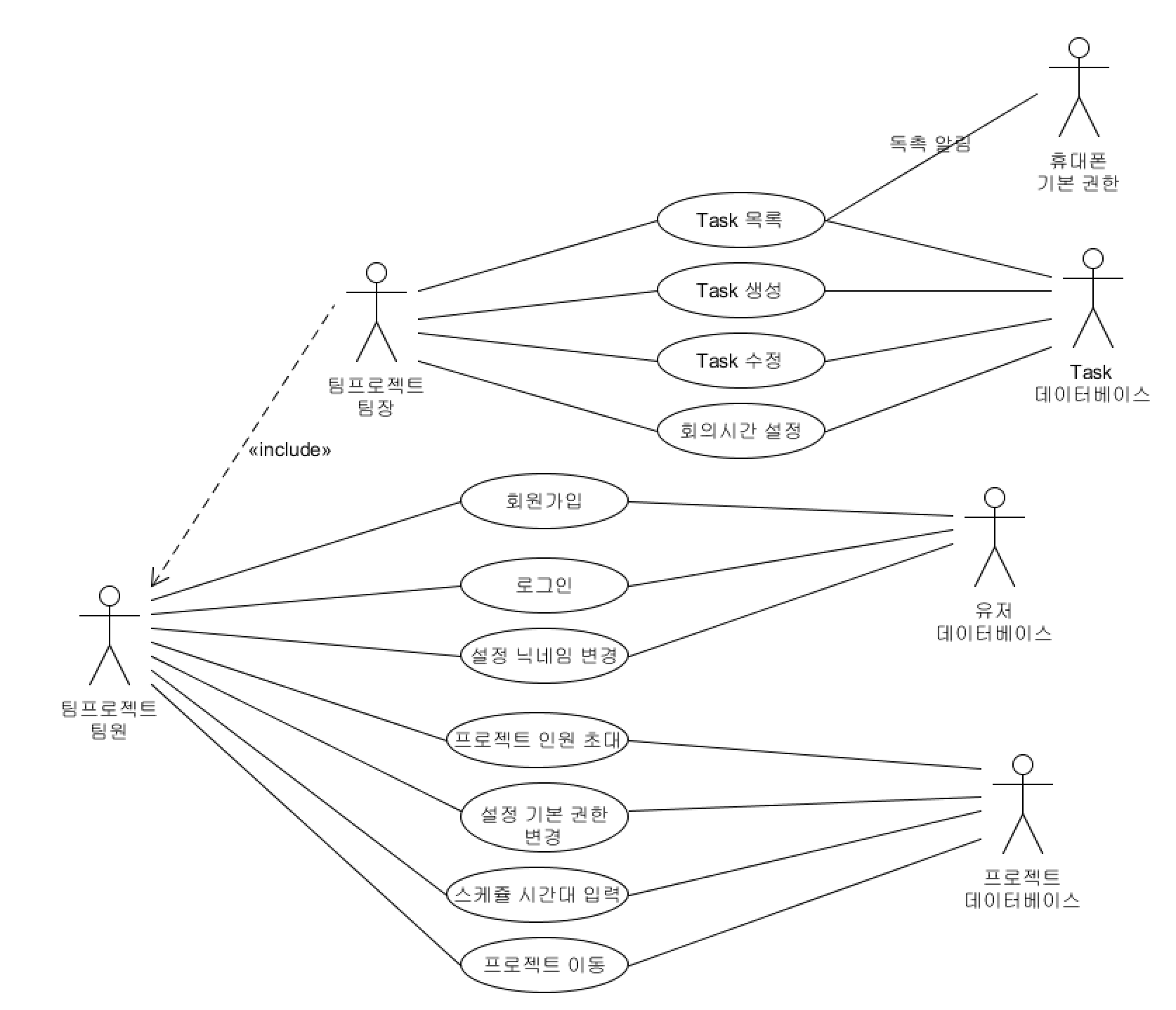
## 기능 요구사항 (Functional Requirements)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 구분 | 상세 | 비고 |
| MAIN\_00 | 로그인 및  메인 화면 | * 웹/앱을 실행했을 때 기본으로 제공되는 화면. 로그인 후 자신 계정에 저장되어 있는 팀 프로젝트 목록 확인 가능. * 로그인 기능, 팀 프로젝트 목록 등은 데이터베이스에서 저장하고 있어야 한다. 데이터베이스에서 로그인이 정상적으로 매칭되면, 팀 프로젝트 목록을 표시한다. * 팀 프로젝트 목록에서 자신이 원하는 팀 프로젝트로 이동할 수 있다. * 메인화면에서 설정으로도 이동할 수 있다. |  |
| MAIN\_01 | 설정 | * 알림, 전화, 문자 등 스마트폰 기본 권한을 관리하는 기능이 있다. * 닉네임을 확인하고 변경할 수 있다. |  |
| PROJ\_00 | 팀 프로젝트  화면 | * 메인 화면에서 팀 프로젝트 목록 중 팀 프로젝트를 눌렀을 때 나오는 화면이다. * 팀 프로젝트 화면에서는 참여자 목록을 확인할 수 있다. * 팀 프로젝트 화면에서는 팀원 추가 버튼을 누를 수 있다. * 팀 프로젝트 화면에서는 스케줄로 이동할 수 있다. * 팀 프로젝트 화면에서는 캘린더로 이동할 수 있다. | 6.1 Low Quality 프로토타입 항목 참고 |
| PROJ\_01 | 팀원 추가  버튼 | * 팀원 추가 버튼을 눌렀을 시, id를 기반으로 팀원 추가가 가능하다. |  |
| PROJ\_02 | 스케줄 | * 이번주, 다음주 시간표를 표시한다. * 시간표는 월화수목금토일, 1시간 단위로 구성된다. * 시간 등록 버튼을 이용해 각자 여유로운 시간을 입력할 수 있다. * 다른 팀원이 입력한 시간을 확인할 수 있다. * 모든 팀원이 가능하다고 입력한 시간대만 뽑아서 확인할 수 있다. | 6.1 Low Quality 프로토타입 항목 참고 |
| PROJ\_03 | 캘린더 | Task를 생성하고 관리, 확인하는 영역.   * 화면에는 기본적으로 달력, Task 목록을 확인할 수 있는 화면으로 구성된다. * 달력(화면 상단)   + 날짜별로 특이 사항을 표기한다.     - 미팅 일정     - Task 마감 시한   + 본인의 Task는 노란색으로 표시된다.   + 타인의 Task는 검정색으로 표시된다.   + 완료된 Task는 취소선이 추가로 표시된다.   + 캘린더에서 해당하는 날짜를 눌러, 해당 날짜의 상세 내용(PROJ\_04)을 확인할 수 있다. * Task 진행 화면 (화면 하단)   + 본인이 수행중인 Task 목록을 표시한다.   + 하단 좌측에는 진행중인 Task 목록이 표시된다.   + 하단 우측에는 알림 목록이 표시된다. * Task 생성 버튼   + Task를 생성하는 창(PROJ\_05) 으로 넘어갈 수 있다.   + 생성 권한을 가진 팀장만 버튼이 표시된다. | 6.1 Low Quality 프로토타입 항목 참고 |
| PROJ\_04 | 날짜 확인 화면 (상세) | * 해당 날짜를 왼쪽 상단에 표시하고, 배정된 Task를 추가한다. * 타인의 Task와 배정된 인원을 확인할 수 있다. * 타인의 Task가 완료되지 않은 경우, 배정된 인원에게 독촉 알림을 보낼 수 있다. * 뒤로 가기 버튼을 누를 경우, 캘린더 (PROJ\_03)으로 넘어간다. * Task 생성 버튼   + 해당 날짜를 마감시한으로 입력한(수정 가능) Task를 생성하는 창(PROJ\_05)으로 넘어갈 수 있다. | 6.1 Low Quality 프로토타입 항목 참고 |
| PROJ\_05 | Task 생성 화면 | * 날짜를 선택하여 Task를 생성한다. * 마감시한은 년/월/일/시 형태로 되어 있다. * Task에 내용, 인원을 할당한다. * 각 숫자를 눌러, 직접 날짜를 수정할 수 있다. (스마트폰 내 날짜 입력 api 사용) * 뒤로 가기 버튼을 눌렀을 경우, 입력된 내용이 삭제된다는 알림을 한 번 띄우고, “네”를 누를 경우 이전 화면으로 넘어간다. * Task 생성이 완료되면 이전 화면으로 넘어간다. |  |

## Use case 목록

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 유스케이스 명 | 설명 | 우선 순위 |
| LOGIN01 | 로그인 | 모든 사용자는 앱에 접속시 로그인을 한다. 유효한 아이디와 비밀번호시, 메인화면으로 이동한다. | 1 |
| LOGIN02 | 회원가입 | 로그인 화면에서 계정이 없을 경우, 회원가입 버튼을 누른다. 새로운 아이디와 비밀번호를 만든다. | 1 |
| SET01 | 설정  닉네임 변경 | 설정 화면에서 닉네임을 변경할 수 있다. 기본값은 아이디이다. | 3 |
| SET02 | 설정  기본 권한 변경 | 설정 화면에서 알림 여부를 선택할 수 있다. | 3 |
| PROJ01 | 팀 프로젝트 이동 | 모든 사용자는 팀 프로젝트 목록에서 특정 팀 프로젝트를 클릭하면, 해당 팀 프로젝트로 이동한다. | 1 |
| PROJ02 | 팀 프로젝트 인원 초대 | 팀원은 팀 프로젝트 화면에서 인원 초대 버튼을 누른 후 아이디를 기준으로 초대할 수 있다. | 1 |
| SCHE01 | 스케쥴  시간대 입력 | 팀원은 주간 스케줄에서 여유로운 시간대를 입력할 수 있다.  다른 팀원이 입력한 여유로운 시간대를 확인할 수 있다. | 1 |
| SCHE02 | 회의시간 설정 | 팀장은 여유로운 시간대를 바탕으로 회의시간을 설정할 수 있다.  설정한 회의시간은 캘린더에 기록된다. | 2 |
| CAL01 | Task 목록 및 독촉 알림 전송 | 팀원은 날짜를 선택하여 세부 Task를 확인, 특정 Task 독촉 알림을 전송할 수 있다. | 1 |
| CAL02 | Task 완료 | Task 할당 받은 팀원은 Task 완료 버튼을 사용할 수 있다. | 1 |
| CAL03 | Task 생성 | 팀장은 Task생성 버튼을 눌러 Task를 생성한다. | 1 |
| CAL04 | Task 수정 | 수정 버튼을 눌러 수정한다. | 1 |

### Use case Diagram



### Use case 명세

|  |  |
| --- | --- |
| 시스템 제목 | 로그인 시스템 |
| 유스케이스 이름 | 로그인 |
| 유스케이스 ID | LOGIN01 |
| 액터 | 팀원 |
| 시작 조건 | 사용자가 앱에 접속한다. |
| 기본 흐름 | 1. 사용자가 아이디와 비밀번호를 입력한다. 2. 유저 데이터베이스에 접속하여 입력받은 아이디, 비밀번호와 일치하는 계정이 있는지 확인한다. 3. 일치할 경우 로그인에 성공하여 4.1.의 PROJ\_00으로 넘어간다. |
| 대안 흐름 | 일치하지 않을 경우 로그인 실패 메세지를 띄우고 아이디와 비밀번호를 다시 입력받는다. |
| 종료 조건 | 사용자가 입력한 아이디, 비밀번호가 유저 데이터베이스에 기록된 것과 일치하여 로그인이 성공한다. |

|  |  |
| --- | --- |
| 시스템 제목 | 로그인 시스템 |
| 유스케이스 이름 | 회원가입 |
| 유스케이스 ID | LOGIN02 |
| 액터 | 팀원 |
| 시작 조건 | 사용자가 로그인 화면에서 회원가입을 선택한다. |
| 기본 흐름 | 1. 새로운 닉네임을 입력받는다. 2. 새로운 아이디를 입력받는다. 3. 입력받은 아이디가 유저 데이터베이스에 존재하는지 아닌지 판별한다. 4. 입력받은 아이디가 유저 데이터베이스에 존재하지 않을 경우 새로운 비밀번호를 입력받는다. 5. 입력받은 닉네임과 아이디, 비밀번호를 유저 데이터베이스에 기록한다. |
| 대안 흐름 | 4A   1. 입력받은 아이디가 유저 데이터베이스에 존재할 경우 “이미 존재하는 아이디입니다”를 출력한다. 2. 유저가 사용 가능한 아이디를 새로 입력하거나 로그인 화면으로 되돌아간다.   4B   1. 입력받은 아이디에 사용 불가능한 특수문자가 들어갈 경우 아이디 재입력을 요청한다. 2. 유저가 사용 가능한 아이디를 새로 입력한다. |
| 종료 조건 | 새로운 닉네임, 아이디, 비밀번호를 유저 데이터베이스가 기록된다. |

|  |  |
| --- | --- |
| 시스템 제목 | 설정 시스템 |
| 유스케이스 이름 | 닉네임 변경 |
| 유스케이스 ID | SET01 |
| 액터 | 팀원 |
| 시작 조건 | 사용자가 회원가입이 되어있고 로그인 중이어야 한다 |
| 기본 흐름 | 1. 사용자가 닉네임 변경 버튼을 누른다 2. 입력창에 변경 후의 닉네임을 입력한다 3. 입력한 닉네임이 정확한지 확인한다 4. 사용자가 결정 버튼을 누른다 |
| 대안 흐름 | 1. 입력받은 닉네임에 사용 불가능한 문자가 포함된 경우 메세지를 출력한 뒤 다시 입력받는다 2. 입력받은 닉네임이 최소 글자수보다 모자라거나 최대 글자수보다 많으면 결정 버튼을 비활성화하고 메세지를 출력한다 |
| 종료 조건 | 입력받은 새 닉네임을 데이터베이스에 반영한다 |

|  |  |
| --- | --- |
| 시스템 제목 | 설정 시스템 |
| 유스케이스 이름 | 기본 권한 변경 |
| 유스케이스 ID | SET02 |
| 액터 | 팀원 |
| 시작 조건 | 사용자가 회원가입이 되어있고 로그인 중이어야 한다 |
| 기본 흐름 | 1. 사용자가 기본 권한 변경 버튼을 누른다 2. 사용자가 필요한 권한 설정을 변경한다 3. 사용자가 결정 버튼을 누른다 |
| 대안 흐름 | 1. 앱 사용에 필수적인 권한이 비활성화될 경우, 경고 메세지를 출력한다 |
| 종료 조건 | 변경된 권한을 시스템에 반영한다 |

|  |  |
| --- | --- |
| 시스템 제목 | 팀 프로젝트 시스템 |
| 유스케이스 이름 | 팀 프로젝트 이동 |
| 유스케이스 ID | PROJ01 |
| 액터 | 팀원 |
| 시작 조건 | 로그인 되어 있어야 함. |
| 기본 흐름 | 1. 팀원이 메인 화면에서 팀 프로젝트 목록에서 하나의 프로젝트의 버튼을 누른다. 2. 선택된 팀 프로젝트 화면으로 이동한다.   a. 화면에는 참여자 목록이 디스플레이된다.  b. 팀원 추가 버튼이 디스플레이된다. |
| 대안 흐름 | 1. 팀 프로젝트 화면에서 뒤로 가기 버튼을 팀원이 누른다. 2. 메인 화면으로 이동한다. |
| 종료 조건 | 팀 프로젝트 화면이 완전히 디스플레이 된다. |

|  |  |
| --- | --- |
| 시스템 제목 | 팀 프로젝트 시스템 |
| 유스케이스 이름 | 팀 프로젝트 인원 초대 |
| 유스케이스 ID | PROJ02 |
| 액터 | 팀원 |
| 시작 조건 | 팀 프로젝트 화면에서 인원 초대 버튼을 누른다. |
| 기본 흐름 | 1. 초대하고자 하는 유저의 아이디를 입력할 수 있는 창이 디스플레이 된다. 2. 팀원이 유저의 아이디를 입력하고, 입력 완료시 확인 버튼을 누른다. 3. 팀 프로젝트 화면이 업데이트되어, 추가한 유저의 아이디가 참여자 목록에 추가된다. |
| 대안 흐름 | 2A.   1. 초대하고자 하는 유저의 아이디를 찾을 수 없는 경우 “존재하지 않는 유저입니다”라는 오류 메시지를 디스플레이한다. 2. 팀원이 오류 메시지를 확인하고 유스케이스를 종료한다.   2B.   1. 유저가 이미 현재 팀 프로젝트 데이터베이스에 등록된 팀원일 경우 “이미 추가된 유저입니다”라는 오류 메시지를 디스플레이한다. 2. 팀원이 오류 메시지를 확인하고 유스케이스를 종료한다. |
| 종료 조건 | 유저 정보가 팀 프로젝트 데이터베이스에 저장된다. |

|  |  |
| --- | --- |
| 시스템 제목 | 스케쥴 시스템 |
| 유스케이스 이름 | 스케쥴 시간대 입력 |
| 유스케이스 ID | SCHE01 |
| 액터 | 팀원 |
| 시작 조건 | 스케쥴 화면에서 시간등록 버튼을 누를 시 |
| 기본 흐름 | 1. 팀원이 시간을 입력할 수 있는 창이 디스플레이 된다. 2. 팀원이 여유있는 시간을 입력한다. 입력이 완료되면 확인 버튼을 누른다 3. 선택한 시간대가 시간표에 추가된다. 4. 시스템이 시간표를 디스플레이한다. |
| 대안 흐름 | 2A.   1. 이미 등록되어 있는 시간과 겹치게 입력 후 완료를 누르면 “이미 등록된 시간대 입니다.”를 출력한다. 2. 팀원이 시간대를 수정하고 확인버튼을 누른다. |
| 종료 조건 | 입력 된 시간이 팀 프로젝트 데이터베이스에 저장된다. |

|  |  |
| --- | --- |
| 시스템 제목 | 스케쥴 시스템 |
| 유스케이스 이름 | 회의 시간 설정 |
| 유스케이스 ID | SCHE02 |
| 액터 | 팀장 |
| 시작 조건 | 스케쥴 화면에서 미팅 시간 정하기 버튼을 누를 시 |
| 기본 흐름 | 1. 팀장이 등록된 시간표를 확인한다. 2. 팀장이 모두가 여유있는 시간대를 지정한 후 확인 버튼을 누른다. 3. 선택한 시간대가 시간표에 “미팅”이라 적힌 시간표로 바뀐다. 4. 팀장을 제외한 팀원 전원에게 알림이 전송된다. |
| 대안 흐름 | 2A.   1. 모든 팀원이 여유있는 시간이 아닐 시 “‘닉네임’님이 해당 시간대에는 안된다고 되어있습니다. 확인하시겠습니까?”라는 창이 디스플레이된다. 2. 확인을 누른다.   2B.   1. 모든 팀원이 여유있는 시간이 아닐 시 “‘닉네임’님이 해당 시간대에는 안된다고 되어있습니다. 확인하시겠습니까?”라는 창이 디스플레이된다. 2. 취소를 누르고 유스케이스를 마친다. |
| 종료 조건 | 미팅을 하는 시간대가 팀 프로젝트 데이터베이스에 저장된다. |

|  |  |
| --- | --- |
| 시스템 제목 | 캘린더 시스템 |
| 유스케이스 이름 | Task 목록 및 독촉 알림 전송 |
| 유스케이스 ID | CAL01 |
| 액터 | 팀원 |
| 시작 조건 | 캘린더 화면에서 날짜를 누를 시 |
| 기본 흐름 | 1. 팀원은 캘린더의 상단에 원하는 날짜에서 Task를 확인할 수 있다.    1. 이때 본인의 Task는 노란색, 그렇지 않으면 회색이다.    2. 이때 완료된 Task는 취소선이 그어진다. 2. 팀원은 원하는 날짜를 클릭시 해당 날짜에 등록되어있는 모든 Task를 확인할 수 있으며, 각 Task마다 독촉 알림버튼을 사용할 수 있다. 3. 팀원이 독촉 알림버튼을 사용시, 해당 Task가 할당된 팀원에게 스마트폰 알림을 전송한다. |
| 대안 흐름 | 3A   1. 독촉 알림버튼을 사용하는 것에는 문제가 발생하지는 않지만, 독촉 알림을 받아야하는 팀원의 스마트폰 기본 권한이 꺼져있는 경우가 있다. 2. 이 경우 독촉 알림버튼을 누른 팀원에게는 ‘{Task가 할당된 팀원의 아이디}의 기본 권한이 꺼져있어 독촉 알림에 실패했어요’ 라는 팝업을 표시한다. |
| 종료 조건 | 독촉 알림을 받아야 하는 팀원의 스마트폰에 독촉 알림이 표시된다. |

|  |  |
| --- | --- |
| 시스템 제목 | 캘린더 시스템 |
| 유스케이스 이름 | Task 완료 |
| 유스케이스 ID | CAL02 |
| 액터 | 팀원 |
| 시작 조건 | 캘린더 화면 하단 좌측에서 완료버튼을 누를 시 |
| 기본 흐름 | 1. 팀원은 캘린더의 하단 좌측에는 본인의 미완료 Task를 확인할 수 있다. 2. 팀원은 각 Task 옆에 있는 완료 버튼을 누른다. 3. 완료버튼이 눌린 Task는 완료된 Task로 변경되므로, 캘린더 상단에는 취소선으로 표시되며, 하단 좌측에는 목록에서 삭제, Task 데이터베이스도 완료상태를 반영한다. |
| 대안 흐름 | 2A   1. Task에 할당된 팀원이 2명 이상일 경우, Task 완료 버튼을 연달아 누르는 경우가 있을 수 있다. 2. 이때 완료버튼을 늦게 누른팀 팀원은 ‘이미 완료된 Task입니다.’ 팝업을 표시하며, 해당 Task가 완료된 캘린더를 보여준다. |
| 종료 조건 | 완료버튼이 눌린 Task는 완료상태가 되며, 캘린더의 상단에는 취소선으로 표시되며, 하단 좌측에는 목록에서 삭제, Task 데이터베이스도 완료상태를 반영한다. |

|  |  |
| --- | --- |
| 시스템 제목 | Task 관리 시스템 |
| 유스케이스 이름 | Task 생성 |
| 유스케이스 ID | CAL03 |
| 액터 | 팀장 |
| 시작 조건 | Task 만들기 버튼을 눌렀을 때 |
| 기본 흐름 | 1. 팀장이 Task의 마감시한을 설정한다.    1. PROJ\_03 캘린더 화면에서 Task 만들기 버튼을 눌렀을 경우, 날짜 와 시간 칸의 기본값은 입력하지 않는다.    2. PROJ\_04 날짜 상세 화면에서 Task 만들기 버튼을 눌렀을 경우, 해당 날짜를 기본값으로 입력한다. 시간 칸은 입력하지 않는다.    3. 스마트폰 내 날짜 입력 api를 사용하여, 사용자가 원하는 날짜와 시간을 설정할 수 있다.   2. 팀장이 Task의 이름을 설정한다.  3. 팀장이 Task에 대한 설명을 작성한다.  4. 팀장이 Task를 수행할 팀원을 설정한다. 자신을 포함한 팀원의 이름 옆에 있는 체크박스를 눌러 선택할 수 있다.  5. 팀장이 화면 내에 있는 Task 만들기 완료 버튼을 누른다.  6. 만들어진 Task가 캘린더에 추가된다.  7. 시스템이 만들어진 Task를 Task 데이터베이스에 저장한다. |
| 대안 흐름 | 뒤로 가기 버튼이 눌릴 경우   1. 시스템이 Task 생성을 취소합니다. Y/N 알림을 디스플레이한다. 2. 팀장이 Y/N 알림를 인식하고 유스케이스를 마친다.    1. 팀장이 Y를 누를 경우, Task 생성 이전의 화면으로 돌아간다.    2. 팀장이 N를 누를 경우, 알림창을 닫는다. |
| 종료 조건 | 만들어진 Task가 Task 데이터베이스에 저장된다. |

|  |  |
| --- | --- |
| 시스템 제목 | Task 관리 시스템 |
| 유스케이스 이름 | Task 수정 |
| 유스케이스 ID | CAL04 |
| 액터 | 팀장 |
| 시작 조건 | Task 수정 |
| 기본 흐름 | 1. 팀장이 Task의 이름. 및 내용을 수정할 수 있다. 2. 팀장이 Task의 마감 시한을 변경할 수 있다.    1. 시스템은 마감 시한을 변경할 경우, 기존 마감 시한을 임시로 저장한다.    2. 팀장은 간단한 마감 시한 변경 사유를 입력할 수 있다. 3. 팀장이 Task를 수행할 팀원을 변경할 수 있다. 자신을 포함한 팀원 이름 옆에 있는 체크박스를 눌러 설정한다.    1. 시스템은 기존 Task를 수행할 인원이 변경될 경우, 기존 인원의 이름을 임시로 저장한다. 4. 팀장이 화면 내에 있는 Task 수정 완료 버튼을 누른다. 5. 시스템이 Task의 내용이 일부 수정되었을 경우, 모든 팀원에게 알림을 보낸다.    1. 마감시한이 변경된 경우, “기존 마감시한 -> 수정된 마감시한” 과 변경 사유를 출력한다.    2. Task를 할당받은 팀원이 변경된 경우, “기존 팀원 -> 수정된 팀원” 과 변경 사유를 출력한다. |
| 대안 흐름 | 5A   1. Task가 수정되지 않았을 경우, “Task 수정 사항이 없습니다. 이전 화면으로 돌아가시겠습니까?” 라는 안내를 띄운다. 2. 팀장이 Y/N 알림를 인식하고 유스케이스를 마친다.    1. 팀장이 Y를 누를 경우, Task 생성 이전의 화면으로 돌아간다.    2. 팀장이 N를 누를 경우, 알림창을 닫는다.   뒤로 가기 버튼이 눌릴 경우   1. 시스템이 Task 생성을 취소합니다. Y/N 알림을 디스플레이한다. 2. 팀장이 Y/N 알림를 인식하고 유스케이스를 마친다.    1. 팀장이 Y를 누를 경우, Task 생성 이전의 화면으로 돌아간다.    2. 팀장이 N를 누를 경우, 알림창을 닫는다. |
| 종료 조건 | 수정된 Task가 Task 데이터베이스에 저장된다. |

# **비기능 요구사항 (Non-Functional Requirements)**

|  |  |  |
| --- | --- | --- |
| ID | 요구 항목 | 설명 |
| NF001 | 성능 | 500명의 사용자가 동시에 사용할 경우, 시스템은 1초 내로 사용자가 요구한 서비스에 대한 응답을 해야한다. |
| NF002 | 신뢰성 | 서비스를 제공하는 도중, 데이터베이스의 오류는 10000개당 1건 이하를 유지한다. |
| NF003 | 가동률 | 서버는 업데이트를 제외한 시간을 제외하고는 24시간 접속되어야 한다.  데이터와 서버의 안전성 때문에 중단 업데이트를 하며, 최대 1시간 안에 다시 서버에 접속할 수 있어야 한다. |
| NF004 | 보안 | 로그인 기능을 구현할 시, 브루트포스로 남의 계정에 로그인이 가능하면 안된다. 이를 위해 같은 아이디로 비밀번호를 5번 틀리면 5분동안 로그인을 할 수 없다. |
| NF005 | 보안 | 데이터베이스의 백업 기간은 1달에 한 번 전체 백업, 1주일에 한 번 차등 백업한다. 백업 데이터는 HDD에 저장한다. |

# **SRS Standards 를 참고하여 필요한 사항을 자유롭게 추가**

## Low Quality 프로토타입

|  |  |
| --- | --- |
| 사용자 인터페이스 | |
|  |  |
| PROJ\_00 | PROJ\_02 |
|  |  |
| PROJ\_03 | PROJ\_04 |