

# 요구사항 분석서

대학 내 동행 모집 앱: CampusMeetup

전공심화프로젝트 2조

조영탁, 이준혁, 박연우

# 목차

## 1. 개요

- 1.1. 프로젝트 기획 배경
- 1.2. 기술 동향 및 핵심 기술
- 1.3. 프로젝트 주요 기능 및 특징
- 1.4. 조원 구성 및 역할 분담
- 1.5. 일정

## 2. 기능적 요구사항

- 2.1. Use Case Diagram
- 2.2. Use Case Document

## 3. 비기능적 요구사항

- 3.1. 사용편리성
- 3.2. 신뢰성
- 3.3. 성능
- 3.4. 이식성
- 3.5. 유지관리
- 3.6. 구현상 제약사항
- 3.7. 인터페이스

# 1. 개요

## 1.1 프로젝트 기획 배경

대학 생활에서 학생들은 스터디, 공연 관람, 운동, 전시회부터 간단한 식사나 술자리 등 다양한 활동을 함께할 동행인을 필요로 하지만, 모집 과정에서 홍보·관리·소통의 어려움을 겪습니다.

한국청소년정책연구원의 「청년의 사회적 고립 실태 및 지원 방안 연구」에 따르면, 코로나19 시기 대학생들은 “학교가 문을 닫고 비대면 수업으로 전환되면서 관계가 얕고 느슨해졌다”고 응답하였으며, 학업 스트레스와 더불어 소속감 부재로 인한 어려움을 호소하는 비율이 높게 나타났습니다. 엔데믹 전환 이후에도 여전히 많은 학생이 대면 관계 형성에 어려움을 겪고 있습니다.

현재 오픈채팅방 혹은 에브리타임 등의 커뮤니티를 통해 동행을 구하는 학생들이 있지만, 이러한 플랫폼에는 정보의 휘발성이 높고 신뢰도 부재와 잠재적 위험 가능성성이 있다는 근본적인 한계가 존재합니다.

우선 다양한 주제의 게시글 혹은 채팅방이 실시간으로 쏟아져 올라오기 때문에 많은 이용자에게 노출이 되지 않는 경우가 잦습니다. 모집 글을 계속해서 다시 올리는 일명 '끌올 행위'는 번거로움을 야기합니다. 또한 익명성에 기반하기에 상대방의 신원을 확인할 방법이 전무합니다. 실제로 각종 커뮤니티에서는 약속을 갑자기 파투 내는 '노쇼' 사례는 물론, 금전 사기나 불미스러운 사건에 대한 우려가 끊임없이 제기되고 있습니다. 즉, 학생들을 체계적으로 연결해주는 플랫폼이 부재하다고 볼 수 있습니다.

이러한 환경 속에서 캠퍼스 동행은 대학생들의 교류 단절 문제를 해결하기 위해 기획된 재학생 전용 미팅 매칭 플랫폼입니다. 학교 이메일 인증 및 프로필 정보를 통해 사용자 신뢰도를 확보하고, 관심사 기반 매칭 및 통합 미팅 관리(미팅 생성·참여·채팅·일정 조율)를 제공함으로써 기존의 불편함을 해소하고자 합니다. 이를 통해 같은 캠퍼스 내 학생들이 안전하고 효율적으로 동행자를 찾을 수 있도록 하는 것을 목표로 합니다.

## 1.2 기술 동향 및 핵심 기술

최근 웹·모바일 서비스 분야에서는 대규모 트래픽을 안정적으로 처리하고, 사용자에게 즉각적인 반응성을 제공하며, 신속한 개발 및 배포 주기를 유지하는 방향으로 기술이 발전하고 있습니다. 본 ‘캠퍼스 동행’ 프로젝트는 이러한 기술적 흐름에 발맞추어, 신뢰성 강화 및 다음과 같은 주요 기술 동향을 반영하고 이를 핵심 기술로 선정하였습니다.

### 1.2.1. 마이크로서비스 아키텍처 (MSA)

기존의 모놀리식(Monolithic) 구조는 모든 기능이 하나의 서버에 통합되어 있어, 일부 기능의 수정·확장 시 전체 시스템에 영향을 미치는 한계가 존재합니다. 이에 따라 최근의 서비스들은 기능 단위를 독립적으로 분리하여 운영할 수 있는 마이크로서비스 아키텍처(MSA)를 도입하는 추세입니다.

카카오, 네이버와 같은 대형 플랫폼 역시 MSA를 기반으로 한 서비스 확장 방식을 채택하여 트래픽 분산과 안정성을 확보하고 있습니다. 이러한 점에서 본 프로젝트 역시 현대적인 백엔드 구조를 따르고 있다는 공통점이 있습니다.

기존 서비스들은 방대한 사용자 규모를 위한 복잡한 인프라에 초점을 맞춘 반면, 본 프로젝트는 대학 단위의 소규모 커뮤니티 환경에 최적화된 MSA 설계를 적용합니다. 각 기능(예: 인증, 미팅 관리, 채팅)을 도메인 중심으로 명확히 분리하여 개발자가 기능별로 빠르게 수정·배포할 수 있도록 하며, 초기 개발 및 유지보수 효율성에 중점을 두었습니다.

이를 통해 향후 특정 기능에 트래픽이 집중되더라도 해당 서비스만 개별적으로 확장할 수 있으며, 전체 시스템의 안정성과 유지보수성을 확보할 수 있을 것으로 기대됩니다.

### 1.2.2 실시간 통신

현대의 사용자들은 페이지 새로고침 없이도 즉각적인 데이터 반영을 기대합니다. 이에 따라 서버와 클라이언트 간의 지속적인 연결을 유지하는 WebSocket 및 SSE(Server-Sent Events) 기술이 실시간 채팅과 알림 기능의 표준으로 자리 잡고 있습니다.

본 프로젝트에서는 실시간 상호작용이 필요한 채팅 기능에 WebSocket을, 미팅 일정 변경이나 알림과 같은 단방향 정보 전달에는 SSE를 적용할 계획입니다.

이러한 실시간 통신 기술의 도입을 통해 사용자에게 즉각적이고 몰입감 있는 서비스 경험을 제공할 수 있을 것으로 예상됩니다.

### 1.2.3. JWT 통합 인증

마이크로서비스 환경에서는 여러 서버 간에 인증 정보를 일관되게 유지하는 것이 중요합니다. 기존의 세션(Session) 기반 인증은 서버 측에서 상태를 관리해야 하기 때문에 확장성 측면에서 제약이 있습니다. 이에 따라 최근에는 JWT(JSON Web Token)을 활용하여 클라이언트가 인증 정보를 직접 보유하는 Stateless(상태 비저장) 구조가 널리 채택되고 있습니다.

Spring Security와 JWT를 결합한 인증 시스템을 도입할 계획이며, 이를 통해 각 서비스 간의 일관된 인증 체계를 구축할 수 있습니다. 또한, Access/Refresh 토큰 방식을 통해 보안성과 확장성을 동시에 확보하고, 분산 환경에서도 효율적인 사용자 인증이 가능하도록 설계할 예정입니다.

대부분의 최신 웹서비스는 이미 JWT 기반 인증을 채택하고 있으며, 분산 서버 환경에서 보안성과 효율성을 입증받은 방식입니다.

기존 서비스는 일반적인 사용자 로그인 인증에만 초점을 두는 반면, 본 프로젝트의 JWT 시스템은 학교 이메일 기반 인증 체계와 결합하여 재학생 신원 보장형 인증 구조를 제공합니다. 즉, 단순 로그인 기능이 아닌, 학교 이메일 인증 → JWT 발급 → 서비스 간 통합 접근 제어로 이어지는 구조를 통해 캠퍼스 단위 신뢰 네트워크를 형성합니다.

이를 통해 익명 기반 커뮤니티의 불신 문제를 기술적으로 보완합니다.

## 1.3 프로젝트 주요 기능 및 특징

### 1.3.1 인증 기능

‘캠퍼스 동행’은 재학생 전용 서비스를 지향하기 때문에, 사용자의 신원을 검증하기 위한 학교 이메일 인증 기반의 가입 절차를 제공합니다. 이를 통해 익명 커뮤니티에서 자주 발생하는 허위 계정 문제를 예방하고, 플랫폼 내 신뢰도를 확보할 수 있습니다.

또한 JWT(Json Web Token) 기반의 인증 방식을 적용하여, 로그인 이후에도 각 서비스 간 일관된 사용자 인증이 유지되도록 설계합니다. 이를 통해 세션 관리의 복잡성을 줄이고, 모바일·웹 환경 모두에서 안정적인 인증 흐름을 보장합니다.

### 1.3.2 마이페이지 기능

사용자는 마이페이지를 통해 자신의 프로필, 관심사, 활동 이력, 참여 미팅 기록 등을 관리할 수 있습니다. 프로필 정보는 공개 범위를 설정할 수 있으며, 참여한 미팅 내역이나 작성한 모집글, 신뢰도 점수 등의 데이터를 통합적으로 조회할 수 있습니다.

또한 이러한 사용자 데이터는 향후 추천 알고리즘에 활용되어, 맞춤형 미팅 추천 기능을 제공할 수 있도록 합니다.

이 기능을 통해 사용자는 자신에게 최적화된 커뮤니티 활동 환경을 경험할 수 있으며, 플랫폼은 사용자 데이터를 기반으로 지속적인 서비스 개선이 가능합니다.

### 1.3.3 신고 및 신뢰도 기능

플랫폼 내의 신뢰성과 건전성을 유지하기 위해 사용자 평가 및 신고 시스템을 제공합니다. 미팅 종료 후, 사용자는 상대 사용자를 평가할 수 있으며, 평가 결과는 동행 지수로 누적 반영됩니다.

이 점수는 참여율, 신고 이력, 이용 태도 등의 다양한 지표를 바탕으로 산출되며, 일정 기준 이하로 하락한 사용자는 자동으로 미팅 생성 제한, 일시 이용 정지, 신고 우선 심사 대상 지정 등의 제재를 받을 수 있습니다.

이를 통해 ‘캠퍼스 동행’은 단순한 매칭 서비스를 넘어, 신뢰 기반의 사용자 관리 시스템을 구현함으

로써 안전하고 지속 가능한 커뮤니티 문화를 조성하는 것을 목표로 합니다.

#### 1.3.4 컨텐츠/검색 기능

사용자는 관심 있는 주제나 키워드를 통해 미팅, 이벤트, 게시글 등 다양한 콘텐츠를 탐색할 수 있습니다.

검색 서비스는 최근 검색어 저장 및 일정, 인원 등 필터 기능을 제공하여 사용자 편의성을 높입니다. 또한 검색 및 탐색 결과는 사용자 프로필 데이터와 신뢰도 지수를 반영하여 더욱 정교하고 개인화된 추천 결과를 제공하며, 무한 스크롤 형태의 실시간 콘텐츠 피드를 구현합니다.

#### 1.3.5 미팅 및 이벤트 관리 기능

본 시스템의 핵심인 미팅 기능은 사용자가 새로운 미팅을 생성하거나 참여할 수 있도록 지원합니다. 미팅 생성 시 제목, 일정, 소개, 모집 인원 등을 입력하며, 미팅장은 참여자 승인, 일정 조율, 공지 관리 등의 기능을 수행할 수 있습니다.

용자는 미팅의 성격을 명확히 나타낼 수 있도록 제목, 소개, 모집 인원, 시간, 장소 등 필수 정보를 직관적인 UI를 통해 손쉽게 등록할 수 있습니다. 참여자는 일목요연하게 정리된 정보를 통해 미팅의 내용을 빠르게 파악하고 참여를 결정할 수 있습니다. 또한 단순 공지를 넘어, 이용자는 일정 기능을 통해 여러 날에 걸친 상세한 타임라인을 구성할 수 있습니다.

모든 미팅에는 독립된 게시판과 채팅 기능이 제공됩니다. 참여자들은 투표 기능을 통해 활동에 대한 의견을 모으거나, 이미지와 글을 공유하며 하나의 커뮤니티를 형성할 수 있습니다. 이는 일회성 만남의 아쉬움을 해소하고 지속적인 관계 형성의 기반이 됩니다.

이벤트 등록 및 변경 정보는 실시간으로 BroadCast 서비스를 통해 다른 사용자에게 전파됩니다.

#### 1.3.6 알림 기능

데이터 변경이 발생할 때마다 사용자는 즉각적인 피드백을 받습니다. 예를 들어, 채팅에 새로운 메시지가 도착하거나 미팅장이 게시판에 중요 공지를 올리면, 관련자에게 즉시 푸시 알림이 전송됩니다.

또한 새로운 멤버가 미팅 참여를 신청하거나 승인될 때, 미팅장과 기존 참여자들은 이를 즉시 인지할 수 있습니다.

이러한 실시간 상호작용은 WebSocket 또는 SSE(Server-Sent Events) 기술을 기반으로 구현됩니다. 이를 통해 사용자는 앱을 수동으로 새로고침할 필요 없이 항상 최신 상태의 정보를 확인할 수 있습니다.

특히, 오프라인 상태에서도 중요 알림을 놓치지 않도록 오프라인 큐(Offline Message Queue)를 활용합니다. 예를 들어, 지하철에서 잠시 인터넷이 끊겨도 다시 연결되는 즉시 그동안 쌓인 중요 알림을 누락 없이 전달받아, 안정적인 커뮤니케이션을 보장합니다.

### 1.3.7 사용자 환경 설정 기능

사용자는 자신의 환경에 맞게 알림 수신 여부, 공개 범위, 테마(라이트/다크모드) 등을 설정할 수 있습니다.

또한 다국어 지원 기능을 통해 사용자 지역이나 언어 설정에 따라 UI가 자동으로 전환되며, 이 모든 설정은 개별 사용자 단위로 저장됩니다.

이 기능은 사용자 경험(UX)을 개인화하고, 불필요한 알림이나 정보 노출을 최소화하여 서비스에 대한 만족도 및 재사용률을 향상시키는 역할을 합니다.

### 1.3.8 관리자 기능

관리자는 관리자 콘솔을 통해 신고 내역 관리, 사용자 통계, 로그 분석, 제재 조치 등의 작업을 수행 할 수 있습니다.

## 1.4 조원 구성 및 역할 분담

박연우	1. 어플리케이션 UI/UX 디자인 2. 문서화 총괄 3. Flutter를 이용한 클라이언트(Web/Android) 개발 참여
이준혁	1. Flutter을 이용한 클라이언트(web/Android) 개발 총괄
조영탁	1. 서버 개발 총괄 A. 전체 서버 아키텍처 설계 및 기술 스택 결정 B. 서버 배포 환경 구성 2. 핵심 기능 개발 A. 실시간 채팅 및 알림 기능 B. 미팅, 유저, 이벤트 관리 기능

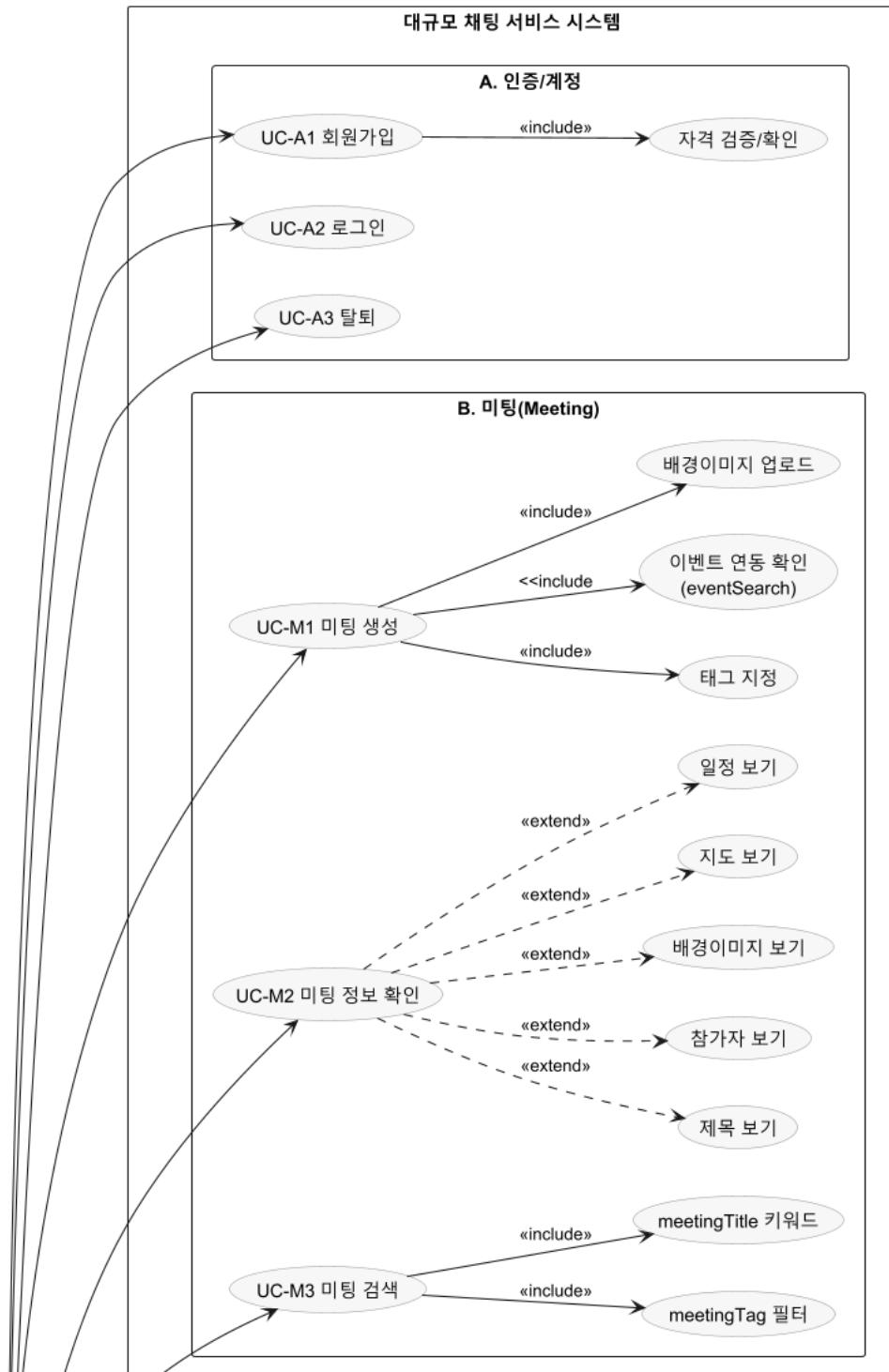
## 1.5 일정

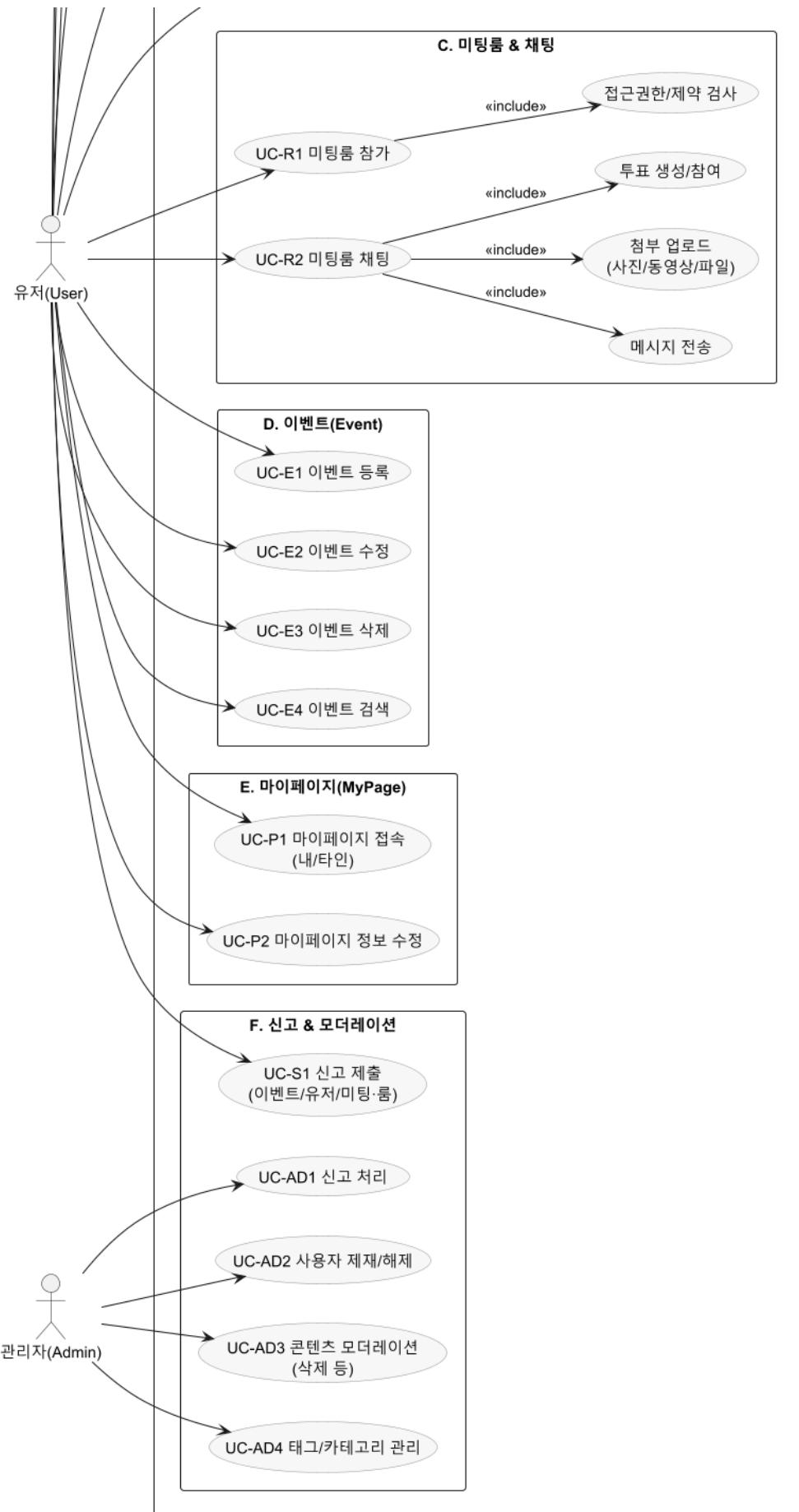
9월	- 프로젝트 준비 및 설계 - 기획 확정
10월	- 요구사항 분석 - 시스템 아키텍처 및 UI/UX 설계
11월	- 핵심 기능 개발 - 부가 기능 개발 및 고도화
12월	- 테스트, 배포 - 최종 문서화 및 발표



## 2. 기능적 요구사항

### 2.1 Use Case Diagram





## 2.2 Use Case Document

### 2.2.1 인증/계정

Name	A1. 회원가입
Description	신규 사용자가 서비스 이용에 필요한 계정을 생성하고, 인증된 사용자로 시스템에 등록되기 위한 기능
Actors	Primary: 사용자 Secondary: User Database, Verification Service (Email/SMS)
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자의 디바이스에 어플리케이션이 설치되어 있다.</li> <li>2. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> <li>3. 시스템의 사용자 데이터베이스가 정상 작동 중이다.</li> <li>4. 인증 발송을 위한 외부 이메일/SMS 서비스 API가 정상 상태이다.</li> </ol>
Post-condition	해당 어플리케이션의 로그인 권한이 부여된다.

Main Flow	
Actor Actions	System Responses
1. 가입에 필요한 정보(이메일, 비밀번호, 닉네임 등)를 입력한다.	1.1 각 입력 필드의 형식 유효성을 실시간으로 검증한다.
	1.2 이메일, 닉네임의 중복 여부를 확인한다.
2. 이메일 인증 요청 후 인증 코드를 입력한다.	2.1 시스템이 자격 검증(이메일 인증)을 수행한다.
	3.1 입력된 정보를 바탕으로 회원정보를 등록한다.
3. 가입 요청을 제출한다.	3.2 로그인 페이지 또는 서비스 메인 페이지로 리디렉션한다.

Alternate Flow	
Actor Actions	System Responses
1.a 중복된 정보 입력: 기가입된 이메일 또는 닉네임을 입력한 경우	
	1.a.1 "이미 사용 중인 이메일(닉네임)입니다."라는 오류 메시지를 해당 입력 필드 하단에 즉시 표시한다.
2.a 불일치하는 인증 코드를 입력하거나 만료 시간 이후에 입력한 경우	
	2.a.1 "인증 코드가 올바르지 않거나 만료되었습니다." 메시지를 표시한다.
2.b 인증 코드 재전송 요청	
	2.b.1 해당 이메일에 대한 이전 인증 코드를 무효화처리한다.
	2.b.2 새로운 코드를 생성하고 발송한다.

Exception Flow	
Actor Actions	System Responses
	2.1.a 인증 메일 발송 API 오류 시 사용자에게 메일 발송 실패 및 대기 메시지를 표시한다.
	3.1.a 시스템 오류(DB 혹은 네트워크 문제) 시 오류 메시지 표시 및 재시도를 권고, 문제가 지속 시 고객지원을 안내한다.

Name	A2. 로그인
Description	시스템에 등록된 계정 정보를 이용하여 사용자의 신원을 인증하고, 서비스 접

	근 권한을 발급받기 위한 기능
Actors	Primary: 사용자 Secondary: User Database, Auth Server
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 시스템에 정상적으로 가입된 '활성' 상태의 계정을 가지고 있다.</li> <li>2. 사용자는 로그인 페이지에 접속한 상태이다.</li> <li>3. 사용자의 디바이스는 인터넷에 안정적으로 연결되어 있다.</li> </ol>
Post-condition	사용자 신원이 확인되고 로그인된 상태로 메인 페이지로 이동한다. 어플리케이션의 다른 기능 사용 가능.

Main Flow	
Actor Actions	System Responses
1. 이메일과 비밀번호를 입력하고 로그인 요청을 제출한다.	
	1.1 각 입력 필드의 형식 유효성을 실시간으로 검증한다.
	1.2 계정 존재 여부 및 자격(비밀번호 일치, 계정 상태 활성화 등)을 검증한다.
	1.3 인증이 성공하면 세션 또는 액세스 토큰(JWT 등)을 발급하고 로그인 상태로 리다이렉트한다.

Alternate / Exception Flow	
Actor Actions	System Responses
	1.2.a 인증 정보 불일치 시 오류 메시지 표시 및 재시도 유도 (시도 횟수 제한이 있을 시 제한 안내 및 계정 잠금)
	1.2.b 사용자 계정이 비활성화, 탈퇴, 제제 상태 일 시 접근 거부 메시지 및 문의 안내

	1.3.a 시스템 오류 시 오류 메시지 표시, 재시도 또는 고객지원 안내.
--	---

Name	A3. 회원 탈퇴
Description	사용자가 서비스 이용 계약을 해지하고, 자신의 계정 및 관련 개인 데이터를 시스템에서 영구적으로 삭제(또는 비활성화)하기 위한 기능
Actors	Primary: 사용자 Secondary: User Database
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 사용자는 '회원 탈퇴' 페이지에 접근한 상태이다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	해당 계정의 상태가 탈퇴로 변경, 사용자는 강제 로그아웃 처리

Main Flow	
Actor Actions	System Responses
1. 회원 탈퇴 페이지에 접근한다.	1.1 시스템에 정의된 탈퇴 관련 정책 안내문을 화면에 표시한다.
2. 탈퇴 사유 선택 및 본인 확인을 위해 인증(비밀번호 재입력 또는 MFA) 후 탈퇴 요청을 보낸다.	2.1 계정 인증 후 계정 상태를 확인한다.
	2.2 계정 상태를 비활성화 또는 영구삭제 처리 후 확인 이메일을 발송한다.

Alternate / Exception Flow	
Actor Actions	System Responses

	2.1.a 인증 정보 불일치 시 오류 메시지 표시 및 재시도 유도
3.a 사용자가 탈퇴 철회	
	3.a.1 진행 중이던 탈퇴 프로세스를 중단한다.

## 2.2.2 미팅

Name	M1. 미팅 생성
Description	사용자가 상세 정보를 포함한 신규 미팅을 생성하여 다른 사용자들이 참여할 수 있도록 등록하는 기능
Actors	Primary: 사용자 Secondary: Meeting Database
Pre-condition	1. 사용자는 서비스에 로그인된 상태이다. 2. 사용자는 미팅 생성 권한을 가지고 있다. 3. 사용자의 디바이스는 인터넷에 연결되어 있다.
Post-condition	신규 미팅 정보가 DB에 저장되고, 사용자는 생성된 미팅 상세 페이지로 이동 가능하다.

Main Flow	
Actor Actions	System Responses
1. 미팅 생성 페이지에 접근한다.	
	1.1 미팅 생성에 필요한 입력 폼(제목, 설명, 태그, 이미지 업로더 등)을 렌더링한다.
2. 미팅 생성에 필요한 기본 정보를 입력한다.	
	2.1 각 입력 필드에 대해 실시간으로 형식 유효성 검증을 수행하고 피드백을 표시한다.
3. 태그 입력란에 키워드를 입력한다.	
	3.1 입력된 키워드로 기존 태그를 검색하여 추천 태그 목록을 제공한다.

4. 배경 이미지 업로드를 수행한다.	
	4.1 선택된 파일의 형식과 크기를 클라이언트 사이드에서 사전 검증한다.
	4.2 파일 비동기 업로드 후 저장된 이미지 URL 반환 및 이미지 미리보기 표시
5. 필요 시 이벤트 연동 요청한다.	
	5.1 이벤트의 연동 가능 여부를 확인 후 해당 이벤트 정보를 미팅에 포함한다.
6. 모든 정보 입력을 완료 후 생성 요청을 보낸다.	
	6.1 전송된 모든 데이터에 대한 서버 사이드 최종 유효성 검증을 수행한다.
	6.2 미팅 생성 완료 후 메시지와 함께 생성된 미팅 상세 페이지로 리디렉션한다.

Alternate / Exception Flow	
Actor Actions	System Responses
2.a 필수 입력 요소 누락 또는 형식 오류	
	2.a.1 오류 메시지 표시로 사용자가 수정 후 재시도하도록 유도한다.
4.a 네트워크 오류나 파일 크기/형식 제한 조건으로 업로드가 실패한 경우	
	4.a.1 사용자에게 다른 파일 선택 혹은 기본 이미지 사용 유도한다.

Name	M2. 미팅 정보 확인
Description	사용자가 특정 미팅의 상세 정보(참여 여부를 결정하는 데 필요한 모든 정보)

	를 조회하는 기능
Actors	Primary: 사용자 Secondary: Meeting Database
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 조회하려는 미팅이 데이터베이스에 존재한다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	사용자는 미팅 상세 정보를 성공적으로 조회하고 화면에서 내용을 확인, 미팅에 참여 요청을 보낼 수 있다.

Main Flow	
Actor Actions	System Responses
1. 미팅 목록(또는 검색 결과)에서 특정 미팅 카드를 클릭한다.	1.1 선택된 미팅의 고유 식별자(Meeting ID)를 포함하여 상세 페이지로 이동하는 요청을 보낸다.
.	1.2 전달받은 ID를 기준으로 해당 미팅의 기본 정보를 제공한다.
	1.3 해당 ID와 매핑된 참석자 목록 및 호스트 정보를 조회한다.
	1.4 조회된 모든 정보를 클라이언트에 전달한다.
	1.5 서버로부터 받은 데이터로 상세 페이지(제목, 설명, 옵션 정보 등)를 표시한다.

Alternate / Exception Flow	
Actor Actions	System Responses
1.a 사용자가 참여 권한이 없는 미팅에 접근을 시도한 경우	

	1.a.1 정보 조회 시, 공개 범위와 사용자의 참여 여부를 먼저 확인한다.
	1.a.2 권한이 없다고 판단되면, 상세 정보 대신 접근 거부 코드와 메시지를 반환한다.
1.b 사용자가 조회한 미팅의 상태가 '종료' 또는 '취소'인 경우	
	1.b.1 페이지 상단에 상태 배너를 표시 후 상호 작용 버튼 비활성화 처리
	1.1.a 전달된 Meeting ID가 DB에 존재하지 않는 경우 404 Not Found 에러를 반환한다.

Name	M3. 미팅 검색
Description	사용자가 키워드, 태그, 날짜 등 다양한 필터 조건을 조합하여 원하는 미팅을 탐색하고, 검색 결과 목록을 확인하는 기능
Actors	Primary: 사용자 Secondary: Meeting Database
Pre-condition	1. 사용자는 서비스에 로그인된 상태이다. 2. 조회하려는 미팅이 데이터베이스에 존재한다. 3. 사용자의 디바이스는 인터넷에 연결되어 있다.
Post-condition	사용자가 입력한 검색 조건에 부합하는 미팅 목록을 성공적으로 조회 후 미팅을 선택해 상세 페이지로 이동하거나 재검색할 수 있다.

Main Flow	
Actor Actions	System Responses
1. 검색 페이지의 검색 바에 키워드를 입력한다.	
2. 사용자가 필터 옵션을 선택한다.	
	2.1 사용자가 선택한 필터 옵션을 내부 상태에 저장한다.

3. 검색 버튼 혹은 Enter 키 입력	
	3.1 현재 설정된 모든 검색어와 필터 조건을 취합하여 검색 API 요청을 서버로 전송한다.
	3.2 수신된 쿼리의 유효성을 서버 사이드에서 검증 후 키워드와 태그에 대한 전체 검색을 수행한다.
	3.3 DB를 대상으로 필터링을 수행한다.
	3.4 요청된 정렬 기준을 적용하여 최종 결과 목록에 대해 페이지네이션 또는 무한 스크롤 로직을 적용하여 보여준다.

Alternate / Exception Flow	
Actor Actions	System Responses
	3.2.a 사용자의 조건에 부합하는 미팅이 없을 시 결과 없음 상태를 반환한다.
	3.2.b 사용자의 검색 경험을 돋기 위해 관련 키워드 추천 혹은 전체 목록 보기 버튼 제공
3.a 사용자 입력에 허용되지 않은 특수문자 포함 혹은 너무 긴 쿼리를 전송한 경우	
	3.a.1 에러 메시지를 검색 바 하단에 표시한다.

Name	R1. 미팅룸 참가
Description	사용자가 접근 권한 검사를 통과한 후, 특정 미팅의 실시간 상호작용 공간에 입장하는 기능
Actors	Primary: 사용자 Secondary: Meeting Database, User Database

Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 조회하려는 미팅이 데이터베이스에 존재한다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	사용자는 미팅 UI에 진입하여 미팅 내 상호작용이 가능하다.

Main Flow	
Actor Actions	System Responses
1. 미팅 상세 페이지에서 참여 버튼을 클릭한다.	1.1 해당 미팅의 고유 ID를 포함하여 서버에 입장 가능 여부 확인을 요청한다.
	1.2 미팅의 제약 조건(모집 기간, 중복 요청 여부 등)을 확인한다.
	1.3 미팅에 호스트가 설정한 품이 존재할 시 사용자에게 품을 표시한다.
2. 품이 존재할 시 품의 각 항목을 작성하고 제출 버튼을 클릭한다.	2.1 제출된 데이터의 유효성을 검증한다.
	2.2 사용자의 요청을 '승인 대기(Pending)' 상태로 데이터베이스에 저장한다.
	2.3 호스트에게 새로운 참여 요청 알림을 발송한다
	2.4 사용자에게 요청 전달 완료 안내 메시지를 보여준다.
3. 사용자가 승인 알림을 확인하고, 미팅 상세 페이지의 입장하기 버튼을 클릭한다..	
	3.1 사용자의 상태가 승인 완료인지 최종 확인 후 입장 허가

Alternate / Exception Flow	
Actor Actions	System Responses
	1.1.a 사용자가 서비스 제재 상태인 경우 입장을 거부하고 거부 사유를 명시하여 응답한다.
	1.2.a 정원 초과 또는 종료된 미팅의 경우 사유 응답

Name	R2. 미팅룸 채팅
Description	미팅에 입장한 사용자들이 텍스트, 파일 첨부, 투표 등의 기능을 통해 실시간으로 정보를 교환하고 상호작용하는 기능
Actors	Primary: 사용자 Secondary: Real-time Chat Server (WebSocket), Database, File Storage Server
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 사용자는 미팅에 성공적으로 입장한 상태이다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	사용자의 메시지 또는 생성된 콘텐츠가 채팅룸의 다른 참여자들에게 실시간으로 전송되고 DB에 기록된다.

Main Flow	
Actor Actions	System Responses
1. 채팅 입력창에 텍스트를 입력 후 전송	

요청한다.	
	1.1 메시지 내용을 담은 데이터 객체(payload)를 WebSocket을 통해 채팅 서버로 전송한다.
	1.2 수신한 메시지에 대해 유효성(금지어, 길이) 검사를 수행한다.
	1.3 해당 미팅룸에 연결된 모든 클라이언트에게 메시지를 브로드캐스트한다.
	1.4 전송된 메시지를 대화 기록으로 저장한다.
2. 파일 첨부 버튼을 눌러 사진/동영상/문서 등을 선택한다.	
	2.1 선택된 파일의 용량/형식을 클라이언트 단에서 사전 검증한다.
	2.2 텍스트 메시지와 파일 URL을 포함한 데이터 객체를 WebSocket을 통해 채팅 서버로 전송한다.
	2.3 메시지를 해당 룸의 모든 클라이언트에게 브로드캐스트 후 수신한 다른 클라이언트들은 파일 URL을 이용해 이미지를 렌더링하거나 다운로드 링크를 제공한다.
	2.4 타임스탬프/보낸이/전달 상태(보냄/수신/표시)를 관리하고, 클라이언트는 UI를 갱신한다.

Alternate / Exception Flow	
Actor Actions	System Responses
	1.1.a 실시간 연결이 끊어진 경우 자동으로 재연결을 시도, 성공 시 연결이 끊어져 있던 동안 놓친 메시지를 서버에 요청하여 동기화. 최종 실패 시 서버 연결 끊김 메시지를 표시한다.
	1.2.a 유효성 검사에 맞지 않는 경우 메시지 전송을 차단하고 오류 메시지를 보낸다.

	2.1.a 파일 용량이 너무 크거나 네트워크 오류 시 파일 업로드 실패 사유 안내 메시지와 함께 재시도 옵션을 제공한다.
--	---

### 2.2.3 이벤트

Name	E1. 이벤트 등록
Description	생성 권한이 있는 사용자가 신규 이벤트를 시스템에 등록하는 기능. 등록된 이벤트는 미팅과 연동되거나, 공개 목록에 노출될 수 있다.
Actors	Primary: 사용자, 관리자 Secondary: Event Database, Meeting Database
Pre-condition	1. 사용자는 서비스에 로그인된 상태이다. 2. 사용자에게 이벤트 생성 권한이 있다. 3. 사용자의 디바이스는 인터넷에 연결되어 있다.
Post-condition	신규 이벤트 정보가 DB에 저장되고 고유 ID가 생성된다. 사용자는 등록 완료 메시지를 확인하고 생성된 이벤트 상세 페이지로 이동할 수 있다.

Main Flow	
Actor Actions	System Responses
1. 이벤트 등록 페이지로 이동한다.	1.1 이벤트 등록에 필요한 UI 컴포넌트(입력 폼, 날짜 선택기, 파일 업로더 등)를 렌더링한다.
2. 이벤트 등록 시 필수 정보를 입력한다.	2.1 각 입력값에 대해 실시간으로 유효성(글자 수, 날짜 선후 관계 등)을 검증하고 피드백을 표시한다.
3. 이미지 업로드	

	3.1 이미지 파일의 형식/크기를 사전 검증 후 파일 스토리지 서버로 업로드하고, 반환된 URL 을 저장한다..
4. 모든 정보 입력을 확인하고 이벤트 등록 요청을 보낸다.	
	4.1 전송된 모든 데이터의 유효성을 서버 사이드에서 최종 검증한다.
	4.2 테이블에 이벤트 정보를 저장 후 등록 성공 메시지를 반환하며, 생성된 이벤트의 상세 페이지로 리디렉션한다.

Alternate / Exception Flow	
Actor Actions	System Responses
	2.1.a 필수 입력 요소 누락 혹은 형식 오류 시 요청을 거부하고, 구체적인 오류 정보를 반환한다.
	3.1.a 이미지 및 첨부파일 업로드 실패 시 업로드 실패 메시지 표시 후 사용자에게 재업로드를 요청한다.
	4.2.a DB 서버 장애로 저장 실패 시 진행 중이던 트랜잭션을 즉시 롤백하여 데이터 불일치를 방지하고, 등록 실패 메시지 표시한다.

Name	E2. 이벤트 수정
Description	사용자 또는 이벤트 관리자가 이미 등록된 이벤트의 정보를 변경하여 최신 상태로 유지하는 기능

Actors	<p>Primary: 사용자, 관리자</p> <p>Secondary: Event Database</p>
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 수정하려는 이벤트가 시스템에 정상적으로 등록되어 있다.</li> <li>3. 사용자는 해당 이벤트의 상세 정보 페이지에 접근한 상태이다.</li> <li>4. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	해당 이벤트 정보가 사용자가 수정한 내용으로 갱신된다. 사용자는 변경 완료 메시지를 확인한다.

Main Flow	
Actor Actions	System Responses
1. 자신이 생성한 이벤트의 상세 페이지에서 '수정하기' 버튼을 클릭한다.	1.1 이벤트 ID를 포함하여 수정 페이지로 이동하는 요청을 보낸다.
	1.2 DB에서 해당 이벤트의 기존 정보를 조회하여 클라이언트로 전달, 전달받은 데이터로 수정 폼의 각 항목(제목, 설명 등)을 채워서 표시한다.
2. 제목, 설명, 기간, 첨부 파일 등 변경할 내용을 수정한다.	2.1 각 항목의 유효성을 실시간으로 검증한다.
3. 모든 수정을 완료하고 '저장하기' 버튼을 클릭한다.	3.1 변경된 데이터와 함께 수정 요청을 수신한다. 3.2 요청을 보낸 사용자가 해당 이벤트의 소유자 또는 관리자인지 확인한다 3.3 변경된 값들이 비즈니스 규칙(e.g., 이미 종료된 이벤트의 날짜 변경 불가)에 맞는지 검증한다

	다.
	3.4 (파일 변경 시) 기존 파일을 삭제하고 새로 업로드된 파일의 URL로 교체한다.
	3.5 이벤트 정보 수정 완료 확인 메시지를 표시하고, 수정된 내용이 반영된 이벤트 상세 페이지로 리디렉션한다.

Alternate / Exception Flow	
Actor Actions	System Responses
	3.2.a 이벤트 생성자 혹은 관리자가 아닌 사용자가 수정을 시도한 경우, 요청을 즉시 거부하고 수정 권한 없음 에러를 반환한다.
	3.3.a 유효하지 않은 변경 시도 시 요청 거부, 구체적인 실패 사유 반환 후 입력 수정을 유도한다.
	3.5.a 사용자가 수정 페이지에 머무는 동안 다른 관리자가 먼저 해당 이벤트를 수정하고 저장한 경우, 데이터의 최종 수정 시간을 비교하여 충돌을 감지한다.
	3.5.b 요청을 거부하고 에러를 반환하여 사용자의 페이지 갱신을 유도한다.

Name	E3. 이벤트 삭제
Description	이벤트 소유자 또는 관리자가 등록된 이벤트를 시스템에서 삭제(또는 비활성화)하여, 더 이상 사용자에게 노출되거나 참여할 수 없도록 처리하는 기능
Actors	Primary: 사용자, 관리자 Secondary: Meeting Database, Event Database, Notification

Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 삭제하려는 이벤트가 시스템에 정상적으로 등록되어 있다.</li> <li>3. 사용자는 해당 이벤트의 상세 페이지 또는 관리자 대시보드에 접근한 상태이다.</li> <li>4. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	해당 이벤트의 상태가 삭제 또는 비활성으로 변경되며, 연동된 미팅의 정보가 변경된다.

Main Flow	
Actor Actions	System Responses
1. 이벤트 상세 페이지 또는 관리 화면에서 '삭제하기' 버튼을 클릭한다.	1.1 삭제 여부를 확인하는 모달(팝업) 창을 띄운다.
2. 확인 모달 창에서 삭제 버튼을 최종적으로 클릭한다.	<p>2.1 삭제 요청을 수신하고, 요청을 보낸 사용자가 해당 이벤트의 소유자 또는 관리자인지 권한 검증을 수행한다.</p> <p>2.2 해당 이벤트의 상태(진행 중, 종료 등)를 확인하여 삭제 가능한 조건인지 비즈니스 규칙 검증을 수행한다</p>
	2.3 모든 검증을 통과하면 해당 이벤트 레코드 상태를 삭제 상태로 업데이트한다.
	2.4 삭제된 이벤트와 연결된 모든 미팅들의 이벤트 연동 정보를 변경 후 해당 미팅의 호스트에게 이벤트 취소 관련 시스템 알림을 발송한다.
	2.5 삭제 완료 메시지 표시 후 이벤트 목록 페이지로 리디렉션한다.

Alternate / Exception Flow	
Actor Actions	System Responses
1.a 사용자의 취소 입력	
	1.a.1 모달 창을 닫고 삭제 프로세스를 모두 중단한다.
	1.a.2 사용자는 이벤트 상세 페이지에 그대로 머무른다.
	2.1.a 이벤트 소유자나 관리자가 아닌 사용자가 삭제를 시도한 경우 요청을 즉시 거부 후 예외를 반환한다.
	2.2.a 삭제가 불가능한 경우 요청을 거부하고 구체적인 사유를 반환한다.

Name	E4. 이벤트 검색
Description	사용자가 키워드, 태그, 날짜 등 다양한 필터 조건을 조합하여 이벤트를 탐색하고, 검색 결과 목록을 확인하는 기능
Actors	Primary: 사용자 Secondary: Event Database
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 조회하려는 이벤트가 데이터베이스에 존재한다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	사용자가 입력한 검색 조건에 부합하는 이벤트 목록을 성공적으로 조회 후 이벤트를 선택해 상세 페이지로 이동하거나 재검색할 수 있다.

Main Flow	
Actor Actions	System Responses

1. 검색 페이지의 검색 바에 키워드를 입력한다.	
2. 사용자가 필터 옵션을 선택한다.	2.1 사용자가 선택한 필터 옵션을 내부 상태에 저장한다.
3. 검색 버튼 혹은 Enter 키를 입력한다.	3.1 현재 설정된 모든 검색어와 필터 조건을 취합하여 검색 API 요청을 서버로 전송한다. 3.2 수신된 쿼리의 유효성을 서버 사이드에서 검증 후 키워드와 태그에 대한 전체 검색을 수행한다.
	3.3 DB를 대상으로 필터링을 수행한다.
	3.4 요청된 정렬 기준을 적용하여 최종 결과 목록에 대해 페이지네이션 또는 무한 스크롤 로직을 적용하여 보여준다.

Alternate / Exception Flow	
Actor Actions	System Responses
	3.2.a 사용자의 조건에 부합하는 이벤트가 없을 시 결과 없음 상태를 반환한다.
	3.2.b 사용자의 검색 경험을 돋기 위해 관련 키워드 추천 혹은 전체 목록 보기 버튼 제공
3.a 사용자 입력에 허용되지 않은 특수문자 포함 혹은 너무 긴 쿼리를 전송한 경우	3.a.1 에러 메시지를 검색 바 하단에 표시한다.

## 2.2.4 마이페이지

Name	P1. 마이페이지 접속 (본인/타인)
Description	사용자가 자신 또는 다른 사용자의 프로필, 활동 내역, 소개 등 종합 정보를 조회하는 기능
Actors	Primary: 사용자 Secondary: User Database
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 조회하려는 대상 사용자가 시스템에 존재하는 계정이다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	사용자는 대상 유저의 공개 설정 범위 내에서 프로필 정보를 조회한다.

Main Flow	
Actor Actions	System Responses
1. 사용자 목록 또는 내 정보의 닉네임 링크를 클릭하여 특정 사용자의 마이페이지로 이동을 요청한다.	
	1.1 요청을 수신하고, 페이지를 요청한 유저 (Viewer)와 페이지의 주인(Owner)이 동일 인물인지 판별한다.
	1.2 Owner의 기본 정보(닉네임, 프로필 이미지, 자기소개 등)를 조회한다.
	1.3 Owner의 공개 범위 설정을 확인하고, Viewer의 접근 권한에 따라 노출할 활동 데이터 범위를 결정한다.
	1.4 조회된 모든 정보를 종합하여 마이페이지 UI를 렌더링한다. (Owner일 경우에만 '프로필 편집', '설정' 액션을 추가로 노출)

Alternate / Exception Flow	
Actor Actions	System Responses

	1.2.a 요청한 사용자가 탈퇴 등의 이유로 존재하지 않는 경우 에러 페이지를 표시한다.
--	---

Name	P2. 마이페이지 정보 수정
Description	사용자가 자신의 프로필 정보(닉네임, 소개), 공개 범위 등 개인화된 설정을 변경하고 시스템에 최신 상태로 저장하는 기능
Actors	Primary: 사용자 Secondary: User Database
Pre-condition	<ol style="list-style-type: none"> <li>1. 사용자는 서비스에 로그인된 상태이다.</li> <li>2. 사용자는 자신의 마이페이지 또는 프로필 편집 페이지에 접근한 상태이다.</li> <li>3. 사용자의 디바이스는 인터넷에 연결되어 있다.</li> </ol>
Post-condition	해당 사용자 정보가 수정한 내용으로 갱신된다. 사용자는 변경된 프로필을 즉시 확인할 수 있다.

Main Flow	
Actor Actions	System Responses
1. 자신의 마이페이지에서 프로필 편집 버튼을 클릭한다.	
	1.1 DB에서 현재 로그인된 사용자의 최신 정보를 모두 조회하여 클라이언트로 전달한다.
	1.2 전달받은 데이터로 편집 폼의 각 항목(닉네임, 소개, 설정 토글 등)을 채워서 표시한다.
2. 닉네임을 변경하고, 프로필 이미지를 새로 업로드하며, 설정을 변경한다.	
	2.1 닉네임 입력 시 실시간으로 글자 수 제한을 체크한다
	2.2 새 프로필 이미지 선택 시, 파일 형식/크기를 사전 검증 후 임시 저장한다.

3. 모든 수정을 완료하고 저장 버튼을 클릭한다.	
	3.1 변경된 모든 데이터와 함께 수정 요청을 수신한다.
	3.2 요청을 보낸 사용자의 인증 토큰(JWT)을 확인하여, 수정하려는 프로필의 소유주가 본인이 맞는지 검증한다.
	3.3 데이터 유효성 검증 후 변경된 내용을 업데이트한다.
	3.4 프로필 저장 토스트 메시지를 표시하고, 변경된 내용이 반영된 마이페이지로 리디렉션한다.

### 2.3.5 신고/모더레이션

Name	S1. 신고 제출
Description	사용자가 서비스 내 부적절한 콘텐츠나 사용자, 미팅 등을 발견했을 때, 운영 팀에 알리고 조치를 요청하는 기능
Actors	Primary: 사용자 Secondary: Report Database, Notification Service
Pre-condition	1. 사용자는 서비스에 로그인된 상태이다. 2. 신고하려는 대상(콘텐츠, 유저 등)이 시스템에 존재한다. 3. 사용자의 디바이스는 인터넷에 연결되어 있다.
Post-condition	신고 내용이 저장되고, 관리자에게 알림이 전송된다. 사용자는 신고 접수 완료를 확인한다.

Main Flow	
Actor Actions	System Responses
1. 부적절한 게시물/사용자 프로필 등에서 '더보기' 메뉴를 누르고 '신고하기'를 선택한다.	

	1.1 신고 사유를 선택할 수 있는 모달(팝업) 창을 띄운다.
	1.2 기타 선택 시 구체적인 내용을 입력할 수 있는 텍스트 입력란을 함께 제공한다.
2. 신고 유형을 선택하고, 필요한 경우 상세 사유를 입력한 뒤 '제출' 버튼을 클릭한다.	
	2.1 신고 요청을 수신하고, 필수값(신고 유형)이 포함되었는지 유효성을 검증한다.
	2.2 신고자 ID, 피신고자 ID(또는 콘텐츠 ID), 신고 유형, 상세 내용 등을 저장 후 관리자 대시보드에 신고 접수 알림을 보낸다.
	2.3 사용자에게 신고 접수 확인 메시지를 표시 후 모달을 닫는다.

Alternate / Exception Flow	
Actor Actions	System Responses
1.a 사용자가 이미 신고한 대상을 다시 신고하려고 하는 경우	
	1.a.1 DB에서 '신고자 ID'와 '피신고자 ID' 조합을 확인하여 중복 여부를 검사한다.
	1.a.2 중복일 경우 요청을 거부하고 안내 메시지를 반환한다.
2.a 신고 유형을 선택하지 않고 제출 버튼을 누른 경우	
	2.a.1 제출을 막고, 오류 메시지를 표시한다.

Name	AD1. 신고 처리
------	------------

Description	관리자가 접수된 신고 내용을 검토하고, 정책에 따라 경고, 제재, 보류, 기각 등의 적절한 조치를 취하는 기능
Actors	Primary: 관리자 Secondary: Report Database, User Database, Meeting Database, Event Database
Pre-condition	1. 관리자는 관리자 계정으로 시스템에 로그인한 상태이다. 2. 처리해야 할 신고 내역이 시스템에 1건 이상 존재한다.
Post-condition	신고 내역의 처리 상태가 갱신되고, 정책에 따른 후속 조치(사용자 제재 등)가 시스템에 반영된다.

Main Flow	
Actor Actions	System Responses
1. 관리자 페이지에서 '신고 관리' 메뉴로 이동 한다.	1.1 테이블에서 처리 상태가 '대기중'인 신고 목록을 조회하여 표시한다.
2. 처리할 신고 항목을 클릭하여 상세 내용을 확인한다.	2.1 해당 신고의 상세 내용, 신고된 콘텐츠 원문, 피신고자의 과거 제재 이력 등을 한 화면에서 조회할 수 있도록 데이터를 조합하여 제공한다.
3. 내용을 검토한 후, 처리 방법을 선택한다.	3.1 선택된 조치 내용을 시스템에 반영한다.

Name	AD2. 사용자 제재/해제
Description	관리자가 서비스 규칙을 위반한 사용자에 대해 이용 정지(기간/영구) 조치를

	취하거나, 제재를 해제하는 기능
Actors	Primary: 관리자 Secondary: User Database, Sanction Log Database
Pre-condition	1. 관리자는 관리자 계정으로 시스템에 로그인한 상태이다. 2. 제재 또는 해제하려는 대상 사용자가 시스템에 존재한다.
Post-condition	대상 사용자의 계정 상태가 변경되고, 모든 조치 내역이 로그에 기록된다.

Main Flow	
Actor Actions	System Responses
1. 사용자 관리 페이지로 이동 후, 제재할 사용자를 닉네임 등으로 검색한다.	1.1 검색 조건에 맞는 사용자 목록과 현재 상태를 조회하여 표시한다.
2. 대상 사용자를 선택하고 '제재하기' 버튼을 클릭한다.	2.1 제재 유형(기간 정지, 영구 정지), 기간, 사유를 입력할 수 있는 팝업 창을 띄운다.
3. 제재 기간과 사유를 입력 후 적용한다.	3.1 해당 사용자의 상태 및 제재 만료일을 업데이트한다. 3.2. 처리 완료 메시지 표시 후, 사용자 목록을 갱신한다.

Name	AD3. 콘텐츠 모더레이션
Description	관리자가 신고되었거나 자동 탐지된 비정상 콘텐츠(게시물, 댓글, 미팅 등)를 검토하고 삭제 또는 비공개 처리하는 기능
Actors	Primary: 관리자

	Secondary: Meeting Database, Event Database, Report Database
Pre-condition	<ol style="list-style-type: none"> <li>1. 관리자는 관리자 페이지에 로그인한 상태이다.</li> <li>2. 검토가 필요한 콘텐츠가 신고 목록 또는 자동 탐지 목록에 존재한다.</li> </ol>
Post-condition	대상 콘텐츠의 상태가 삭제 또는 비공개로 변경되고, 처리 기록이 로그에 남는다.

Main Flow	
Actor Actions	System Responses
1. 관리자 페이지의 신고된 콘텐츠 목록으로 이동한다.	1.1 처리 대기 중인 신고 내역과 해당 콘텐츠의 미리보기를 목록 형태로 조회하여 표시한다.
2. 처리할 콘텐츠를 선택하여 원문과 신고 사유를 상세히 검토한다.	
3. 위반이 명확하다고 판단하여 '콘텐츠 삭제' 버튼을 클릭하고 사유를 입력한다.	3.1 해당 콘텐츠의 상태를 업데이트한다. 3.2 테이블에서 관련된 모든 신고 건의 상태를 '처리 완료'로 변경한다. 3.3 목록에서 해당 콘텐츠가 사라지고, 처리 완료 메시지를 표시한다.

Name	AD4. 태그 관리
Description	관리자가 서비스 내 콘텐츠 분류 체계의 기준이 되는 태그와 카테고리를 생성, 수정, 삭제하는 기능
Actors	Primary: 관리자 Secondary: Tag/Category Database, Meeting Database, Event

	Database
Pre-condition	1. 관리자는 관리자 페이지에 로그인한 상태이다.
Post-condition	태그 정보가 시스템에 생성, 수정, 또는 삭제된다.

Main Flow	
Actor Actions	System Responses
1. 관리자 페이지의 태그 관리 메뉴로 이동한다.	1.1 현재 시스템에 등록된 모든 태그 목록과 각 태그가 사용된 콘텐츠 개수를 조회하여 표시한다.
2. 태그 이름 입력 후 추가한다..	2.1 입력된 태그 이름이 이미 존재하는지 중복 검사를 한다.
	2.2 중복이 아닐 시 새로운 태그를 추가 후, 태그 목록을 갱신한다.
3. 기존 태그를 선택 후 삭제한다.	3.1 해당 태그를 참조하는 콘텐츠가 있는지 확인한다.
	3.2 참조하는 콘텐츠가 없으면 태그 목록에서 해당 태그가 사라진다.

### 3. 비기능적 요구사항

#### 3.1 사용편리성

- 모든 아이콘과 버튼은 기능이 명확히 드러나도록 디자인
- 무한 스크롤 및 검색/필터 기능을 제공해 사용자가 컨텐츠 내용을 쉽게 확인할 수 있도록 한다.
- 라이트/다크 모드를 모두 지원하여 사용자의 시스템 설정에 자동으로 동기화되도록 함으로써 다양한 환경에서의 가독성을 제공한다.
- 메인 화면의 미팅 목록 스크롤 시 60fps를 유지하여 부드러운 경험을 제공하고, 버튼 클릭 시 1초 이내에 시각적 피드백을 제공한다.
- 모바일/웹/데스크톱 클라이언트 간에 일관된 UX를 제공해야 한다.

#### 3.2 신뢰성 (Reliability)

- 채팅 메시지 전송 시, 일시적 네트워크 단절 발생 시 메시지가 유실되지 않고 자동 재전송 및 복구 기능을 제공해야 한다.
- 채팅 기록은 영구 저장 및 필요 시 복구가 가능해야 한다.
- 특정 기능(예: 채팅 서버)에 장애가 발생하더라도 다른 기능(미팅 목록 조회 등)은 정상 동작 해야 하며, 장애 발생 시 관리자에게 즉시 알림이 전송되어야 한다.
- 방에 존재하는 인원 수를 실시간으로 정확하게 갱신해야 한다.

#### 3.3 성능 (Performance)

- 시스템은 동시에 10,000명 이상의 실시간 사용자를 처리할 수 있어야 한다.
- 최소 1,000개의 채팅방이 동시에 운영 가능해야 한다.
- 채팅 메시지/이벤트 처리 지연 시간은 200ms 이하, UI 응답은 300ms 이내여야 한다.

- 서버의 CPU, 메모리 사용률을 실시간으로 모니터링하고, 자동 스케일링이 가능해야 한다.

### 3.4 이식성 (Portability)

- 서버는 클라우드 환경(AWS, GCP, Azure 등)과 온프레미스 환경 모두에서 동작 가능해야 한다.
- 클라이언트는 Web, Android, iOS를 동시 지원하며, 공통 API를 활용하여 플랫폼 독립성을 보장해야 한다.

### 3.5 유지관리 (Maintainability)

- 서버와 클라이언트를 병렬적으로 개발할 수 있도록 API 명세(Swagger/OpenAPI)를 제공해야 한다.
- 코드 변경 시 자동 테스트 및 CI/CD 파이프라인을 통해 빠른 배포와 안정성을 보장해야 한다.
- 로깅/모니터링을 통해 장애 발생 시 원인 분석이 가능해야 한다.
- 아키텍처는 모듈화/계층화하여 신규 기능 추가 시 기존 서비스 영향도를 최소화해야 한다.

## 3.6 구현상 제약사항 (Constraints)

### 3.6.1 개발 환경 제약

#### 3.6.1.1 개발 언어

Backend: Spring Boot (Java)를 사용하여 개발

Frontend: Flutter를 사용하여 크로스플랫폼(Web, Android) 클라이언트를 개발

#### 3.6.1.2 프레임워크

Backend: Spring Security를 활용하여 JWT 기반의 인증 및 인가 시스템을 구현한다.

Frontend: Flutter 프레임워크의 표준 라이브러리 및 가이드라인을 준수한다.

### 3.6.2 데이터 및 보안 제약

데이터 영속성을 위해 PostgreSQL을 사용하며, 다른 RDBMS는 허용하지 않는다.

### 3.6.3 성능 제약

- 실시간 메시지 처리 지연 시간은 200ms 이하를 유지해야 한다.
- 주요 기능의 UI 응답 시간은 1초 이내를 만족해야 한다 (전체 요청의 95% 기준).
- 목표 트래픽: 시스템은 아래의 트래픽을 안정적으로 처리할 수 있도록 설계 및 구축되어야 한다.
  - 동시 실시간 유저 수: 10,000명
  - 동시 채팅방 수: 1,000개

### 3.6.4 외부 연동 제약

- 외부 이메일 발송은 SMTP 프로토콜을 사용한다.
- 소셜 로그인(구글/깃허브) 기능은 표준 OAuth 2.0 프로토콜을 준수하여 구현해야 한다.

## 3.7 인터페이스 (Interface)

### 3.7.1 사용자 인터페이스(UI)

<https://www.figma.com/design/W2IDCt3EJwXe5RtATB0Ziw/%EC%A0%84%EC%8B%AC%ED%94%84-3?node-id=4013-254&t=INDCEq5gL40KCfRK-1>

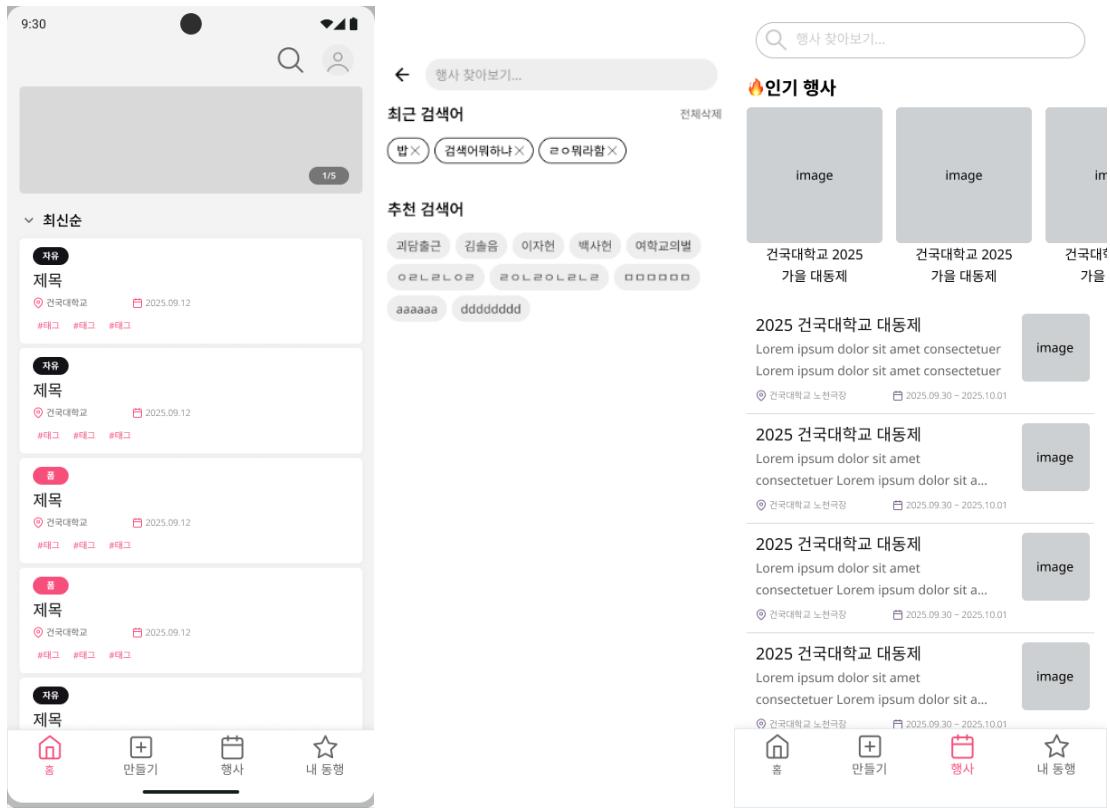
#### 3.7.1.1 온보딩 및 사용자 인증

The wireframe shows a registration form with the following fields:

- Left side:
  - 이름 \* (Name) - Input field with placeholder "성이름" (Last name).
  - 이메일 \* (Email) - Input field with placeholder "example@konkuk.ac.kr". To the right is a small "인증요청" (Verification request) link.
  - 인증코드 \* (Verification code) - Input field with placeholder "000000".
  - 비밀번호 \* (Password) - Input field with placeholder "\*\*\*\*\*".
  - 생년월일 \* (Birthdate) - Input field with placeholder "2025년 9월 28일".
  - 성별 (Gender) - Buttons for "남성" (Male) and "여성" (Female).
- Right side:
  - 프로필을 설정하세요 (Set profile) - Text above a large empty circular profile placeholder with an '@' icon.
  - 닉네임 \* (Nickname) - Input field with placeholder "닉네임을 입력하세요".
  - 학과 (Major) - A dropdown menu labeled "학과 선택" (Select major).
  - 소개 (Introduction) - Input field with placeholder "나를 소개하는 글을 적어주세요".
  - A large pink "완료" (Complete) button at the bottom.

로그인 옵션과 명확한 단계별 가이드라인을 제공하는 이메일 회원가입 프로세스로 구성됩니다. 각 입력 필드에는 플레이스홀더와 유효성 검사 피드백을 제공하여 가입 과정의 이탈을 최소화합니다.

#### 3.7.1.2 동행 미팅, 이벤트 탐색 및 검색



사용자는 소셜미디어 피드와 유사한 스크롤 방식의 인터페이스를 통해 다양한 동행 미팅을 직관적으로 탐색할 수 있습니다. 상단의 검색 버튼을 통해 원하는 조건의 미팅을 빠르고 정확하게 찾을 수 있도록 지원합니다.

### 3.7.1.3 미팅 생성 및 관리

모임 생성하기

제목 \*

제목을 입력해 주세요

설명

설명을 입력해 주세요

날짜

2025.09.28 - 2025.09.30

시간

00:00 ~ 선택 안함

장소

지도에서 위치 주기

최대 참여 인원 수

태그 \*

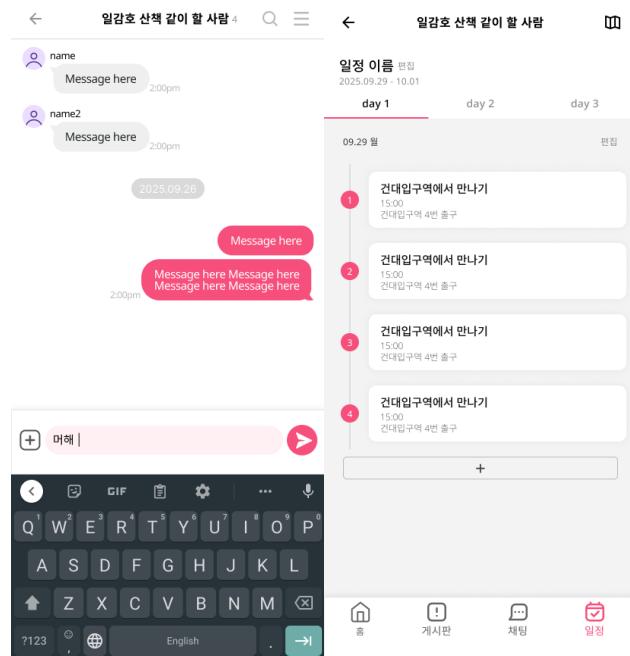
#으로 태그 구분

참여 조건 설정

등록

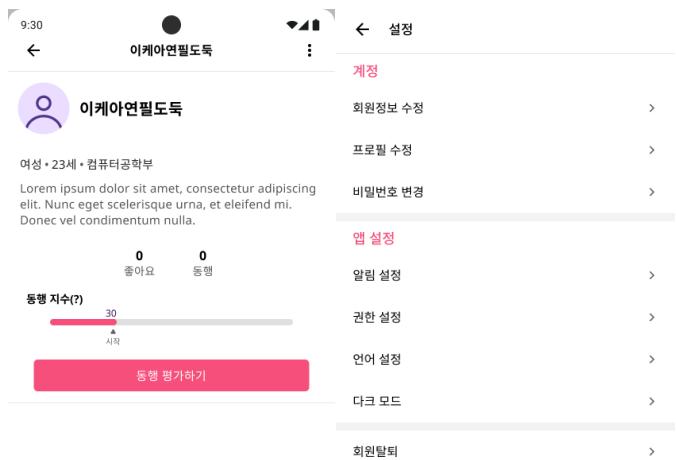
날짜, 시간, 장소(지도 연동), 상세 설명 등 각 단계별로 명확한 과업을 제시합니다. 미팅 생성 후에는 관리 페이지를 통해 참여자 확인, 일정 수정 등 미팅 운영에 필요한 기능들을 통합적으로 제공합니다.

### 3.7.1.4 커뮤니케이션



미팅 참여자 간의 원활한 소통을 위해 대중적인 메신저 앱의 UI/UX를 채택한 실시간 채팅 기능, 게시판, 일정 기능을 제공합니다.

### 3.7.1.5 프로필 및 설정



사용자는 자신의 프로필 정보를 확인하고 수정할 수 있으며, 앱 관련 설정(알림, 계정 관리 등)을 할 수 있는 통합 메뉴를 제공합니다. 정보들을 명확하게 그룹화하여 사용자가 원하는 메뉴를 쉽게 찾을 수 있도록 구성했습니다.

### 3.7.2 소프트웨어 인터페이스

#### 3.7.2.1 Google Maps API

구글에서 제공하는 지도 및 위치 기반 서비스 API

#### 3.7.2.2 Firebase Cloud Messaging (FCM)

Google에서 제공하는 크로스플랫폼 메시징 솔루션. 서버가 특정 사용자 또는 그룹의 디바이스 (Android, iOS, Web)로 직접 푸시 알림을 보낼 수 있도록 지원함. 채팅 메시지 도착, 미팅 참여 승인, 중요 공지사항 등 사용자가 즉시 알아야 할 정보를 앱이 꺼져 있거나 백그라운드 상태일 때도 푸시 알림으로 전달하기 위함.

#### 3.7.2.3 Simple Mail Transfer Protocol (SMTP) 기반 이메일 서비스

이메일 전송을 위한 표준 통신 규약(SMTP)을 기반으로, 대량의 이메일을 안정적으로 발송할 수 있도록 돋는 외부 서비스. 회원가입 시 사용자가 입력한 학교 이메일 주소로 인증 코드 또는 인증 링크를 발송하여 재학생 신원을 확인하기 위함.

#### 3.7.2.4 OAuth 2.0 API

사용자가 기존에 사용하던 소셜 계정(구글, 깃허브 등) 정보를 이용해 제3자 서비스 (CampusMeetup)에 안전하게 로그인할 수 있도록 하는 인증 표준 프로토콜.