

SKKU MAP

Software Requirement Specification

2024.05.26.

Introduction to Software Engineering TEAM 11

Team Leader 이주형

Team Member 김민수

Team Member 박정은

Team Member 정승훈

Team Member Nariman Abubakirov

CONTENTS

1. Introduction

- 1.1. Purpose
- 1.2. Scope
- 1.3. Definitions, Acronyms and Abbreviation
- 1.4. References
- 1.5. Overview

2. Overall Description

- 2.1. Product perspective
 - 2.1.1. Market status
 - 2.1.2. Overall Structure
- 2.2. Product Function
 - 2.2.1. 로그인 및 회원가입
 - 2.2.2. 메인 페이지(강의실 번호 입력)
 - 2.2.3. 건물 외부 네비게이션
 - 2.2.4. 건물 내부 네비게이션
 - 2.2.5. 검색 기록
 - 2.2.6. 즐겨찾기
 - 2.2.7. 나의 프로필
- 2.3. User Classes and Characteristic
 - 2.3.1. User
 - 2.3.2. System Manager
- 2.4. Operation Environment
 - 2.4.1. Hardware
 - 2.4.2. Software
- 2.5. Design and Implementation Constraints
- 2.6. User Documentation
- 2.7. Assumptions and Dependencies

3. Specific Requirements

- 3.1. External Interface Requirements
 - 3.1.1. User Interface

- 3.1.2. Hardware Interfaces
 - 3.1.3. Software Interfaces
 - 3.1.4. Communication Interfaces
 - 3.2. Function Requirements
 - 3.2.1. Use case
 - 3.2.2. Use-Case Diagram
 - 3.2.3. Data Dictionary
 - 3.3. Non-Functional Requirements
 - 3.3.1. Product Requirements
 - 3.3.1.1. Usability Requirements
 - 3.3.1.2. Efficiency Requirements
 - 3.3.1.3. Dependability Requirements
 - 3.3.1.4. Security Requirements
 - 3.3.2. Organization Requirements
 - 3.3.2.1. Environmental Requirements
 - 3.3.2.2. Development Requirements
 - 3.3.3. External Requirements
 - 3.3.3.1. Regulatory Requirements
 - 3.3.3.2. Ethical Requirements
 - 3.3.4. Organizing the Specific Requirements
 - 3.3.4.1. Context Model
 - 3.4. API Specifications
 - 3.5. Main Chain Architecture
 - 3.6. Database Architecture
 - 3.7. Server Architecture
 - 3.8. Infrastructure Architecture
 - 3.8.1. 인프라 설계
 - 3.8.2. 인프라 구축
 - 3.8.3. 인프라 배포
 - 3.8.4. 인프라 운영
 - 3.9. Project Management
 - 3.10. System Evolution
 - 3.10.1. Limitation and Assumption
 - 3.10.2. Anticipated Changes
- 4. Supporting Information
 - 4.1. Software Requirement Specification

LIST OF TABLES

[표 1] definitions.....	7
[표 2] abbreviation and acronyms.....	7
[표 3] Interface of Login page.....	15
[표 4] Interface of Register page.....	16
[표 5] Interface of Top Tab Navigator.....	16
[표 6] Interface of Bottom Tab Navigator.....	17
[표 7] Interface of the Search Lecture Room page.....	18
[표 8] Interface of the external map page.....	19
[표 9] Interface of First Floor Page.....	20
[표 10] Interface of Destination Floor page.....	21
[표 11] Interface of History Page.....	22
[표 12] Interface of Favorites page.....	23
[표 13] Interface of My Profile page.....	24
[표 14] hardware Interface.....	25
[표 15] Software Interface.....	25
[표 16] communication interface.....	26
[표 17] use case for register.....	27
[표 18] use case for login.....	27
[표 19] use case for Password change.....	28
[표 20] use case of inserting the lecture room number.....	28
[표 21] use case of finding the most efficient route - outside the building.....	28
[표 22] use case of finding the most efficient route - inside the building.....	29
[표 23] use case of favorite.....	29
[표 24] use case of Recent.....	30
[표 25] User Data Dictionary.....	32
[표 26] Favorite Data Dictionary.....	32
[표 27] History Data Dictionary.....	33
[표 28] 길찾기 API의 parameter, response 예시.....	36
[표 29] 강의실 리스트 API의 parameter, response 예시.....	37

LIST OF FIGURES

[그림 1] 프로그램 구조.....	10
[그림 2] use case diagram.....	31
[그림 3] RDB ERD.....	32
[그림 4] System Context Model.....	36
[그림 5] 주요 API 명세.....	36
[그림 6] Main Chain 구조.....	38
[그림 7] 데이터베이스 구조.....	39
[그림 8] 서버 구조.....	40
[그림 9] Infrastructure Architecture.....	42
[그림 10] 프로젝트 진행 상황 공유 예시.....	44
[그림 11] 버전, 코드 관리 관련 브랜치.....	45

1.Introduction

1.1 Purpose

본 문서는 'SKKU Map" 애플리케이션 서비스를 제공하기 위한 SRS(Software Requirements Specification)이다. SKKU Map은 기존 네비게이션과 같이 건물 외부에서의 길 찾기 뿐만 아니라, 보다 복잡한 건물 내부의 강의실을 빠른 시간에 최적의 경로로 효율적으로 가기 위해 건물 내부 길 찾기 기능을 제공한다. 해당 서비스는 이를 위한 요구 사항을 요약, 분석하고 본 내용을 바탕으로 성균관대학교 소프트웨어공학개론 team11에서 디자인 및 구현된다. 이 문서의 목적은 새로운 어플리케이션의 Requirement Specification을 개요화하고 공개하는 것이다. 이 애플리케이션은 사용자들이 성균관대학교 캠퍼스의 강의실을 효율적으로 찾을 수 있도록 도움을 주는 것이 목적이다. 기존 네비게이션은 건물 위치까지만 파악할 수 있었던 것과 달리, 건물 내부에서 가려는 강의실의 위치와, 최적의 동선을 보여주어 사용자들이 이동하는데 있어 시간과 노력을 절약할 수 있다.

1.2 Scope

성균관대학교에 방문하는 많은 사람들은 강의실 번호만 보고 해당 강의실의 위치가 어디 있는지 찾기 어려워 하며, 해당 강의실로 가기 위해서는 어떤 길을 거쳐야 하는지 알기 어렵다. 현재 성균관대학교는 강의실을 26312등의 숫자로 표기한다. 여기서 26, 즉 맨 앞 두 숫자는 건물 번호를 나타내고, 세번째 숫자는 층수를 표현한다. 이를 바탕으로, 26동 3층에 있는 강의실이라는 것을 알 수 있다. 하지만, ㄷ자 모형의 건물에 입구는 각 방향으로 여러개 존재하며, 계단과 엘리베이터의 위치가 어디인지는 건물에 들어가면 붙어있는 층별 도면을 찾거나 직접 걸어가서 확인하지 않는 이상 쉽게 파악하기 어렵다. 심지어 기울어진 지면 탓에 건물 연결통로를 거쳐 다른 건물로 이동하면, 층이 바뀌는 일도 일어난다. 정리하자면, 어느 입구로 들어가야 최적의 경로로 쉽게 강의실을 찾아 가는지, 또 3층에 도착해서는 어디에 강의실이 있는지 파악하기 어려운 상황이다. 따라서 이러한 강의실의 위치를 쉽게 찾을 수 있고, 강의실로 가는 길 또한 쉽게 찾을 수 있는 도구의 필요성을 느꼈다. 이 소프트웨어는 성균관대학교를 방문하는 사람들에게 강의실을 보다 쉽게 찾아갈 수 있도록 돕는 것을 목표로 한다.

첫번째로 길찾기 기능은 가고싶은 강의실 번호를 입력하면, 현재 위치에서 강의실까지 도달하는 길을 알려준다. 이는 단지 건물로 향하는 길을 알려주는 것 뿐만

아니라, 건물 내의 경로까지 알려주어 건물 내부에서도 빠르게 강의실까지 찾아갈 수 있도록 한다.

두번째로 즐겨찾기 기능은 자신이 자주 가는 강의실(가령, 매주 있는 수업의 강의실)을 등록할 수 있는 기능이다. 즐겨찾기에 등록해둔 강의실은 강의실 번호를 입력할 필요 없이 손쉽게 강의실로 향하는 길을 알아낼 수 있도록 하는 것을 목표로 한다.

세번째로 최근 방문한 강의실을 볼 수 있는 기능이다. 사용자가 최근 검색한 강의실은 자동으로 저장되어, 사용자가 최근 방문한 강의실을 다시 한번 방문할 때 빠르게 강의실까지 가는 길을 탐색할 수 있도록 하는 것을 목표로 한다.

1.3 Definitions, Acronyms and Abbreviation

아래 표는 이 문서에서 사용된 약어에 대한 설명이다.

[표 1] definitions

약어	설명
유저 / user	“SKKU MAP”의 사용자
System administrator	로그인, 회원가입 및 전체적인 시스템 관리자
Client	서버에 연결된 사용자 장치/사용자
server 서버	네트워크에서 중앙 자원 또는 서비스에 대한 접근 권한을 관리하는 컴퓨터 프로그램
software	컴퓨터가 사용하는 프로그램 및 기타 작동 정보
DB	Database

[표 2] abbreviation and acronyms

약어	설명
OS	operating system
backend	사용자가 볼 수 없는 앱의 부분
frontend	사용자가 직접적으로 보는 앱의 부분
UI/UX	User Interfaces / User Experience

API	Application Programming Interface
HTTP	Hypertext Transfer protocol
PK	Primary Key(on database)
FK	Foreign Key(on database)

1.4 References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>
- “Software Requirement Specification”. SKKU 2022 Fall Introduction to Software Engineering Team7.
https://github.com/skkusal/SWE3002_42/blob/main/2022f/Team7_SRS.pdf

1.5 Overview

본 소프트웨어 요구사항 명세서는 총 3개의 **chapter**로 구성되어 있다. **chapter 2**에서는 **SKKU MAP** 애플리케이션의 전반적인 구조와 설명을 제공한다. 이 챕터에서는 이 애플리케이션의 목적과 실제 시장에서의 사용 용도를 설명하고, 사용자 및 시스템 관리자, 애플리케이션의 기능 및 기반이 되는 환경, 그리고 운영 과정에서의 시스템의 제한 사항과 가정, 다른 시스템과의 의존성이 설명된다. **chapter 3**에서는 구체적인 요구 사항을 기술한다. 먼저 외부 인터페이스 요구사항으로 유저, 하드웨어, 소프트웨어, 커뮤니케이션 인터페이스에 대해 분석한다. 그 다음 다양한 시스템 기능들을 기술하고, 데이터베이스 및 시스템 구조, 소프트웨어 시스템의 특성 및 시스템 제약사항에 대해 설명된다. 네 번째 및 마지막 챕터에서는 본 소프트웨어 요구 사항 명세서 문서를 준비하는 데 참고한 정보가 기록된다. 또한, 본 소프트웨어 요구 사항 명세서 문서를 작성하는 동안 수정할 항목에 대한 설명, 수정된 버전 및 수정한 사람의 이름이 기록되어야 한다.

2. Overall Description

2.1 Product perspective

본 애플리케이션은 성균관대학교를 방문한 학생 또는 외부인들을 위한 소프트웨어로, 자연과학캠퍼스(수원) 캠퍼스의 강의실 번호를 입력하면 해당 강의실의 위치와 그에 대한 내비게이션 기능을 제공한다. 애플리케이션은 강의실 번호를 기반으로 사용자에게 건물의 위치와 몇 층에 있는지에 대한 정보를 제공한다. 또한, 건물 외부와 건물 내부 경로 모두 제공한다. 건물 내의 경로는 첫 번째 층, 목적 층으로 나누어 시각적으로 표시된다. 이러한 기능은 사용자들이 캠퍼스 내에서 강의실로의 경로를 쉽게 찾을 수 있도록 돕는데 중점을 두고 있다. 이는 사용자의 편의성을 높이고, 시간을 절약하는 데 도움이 된다.

2.1.1 Market status

성균관대학교의 건물 구조나 강의실 위치를 알 수 있는 비슷한 애플리케이션은 두가지가 있다.

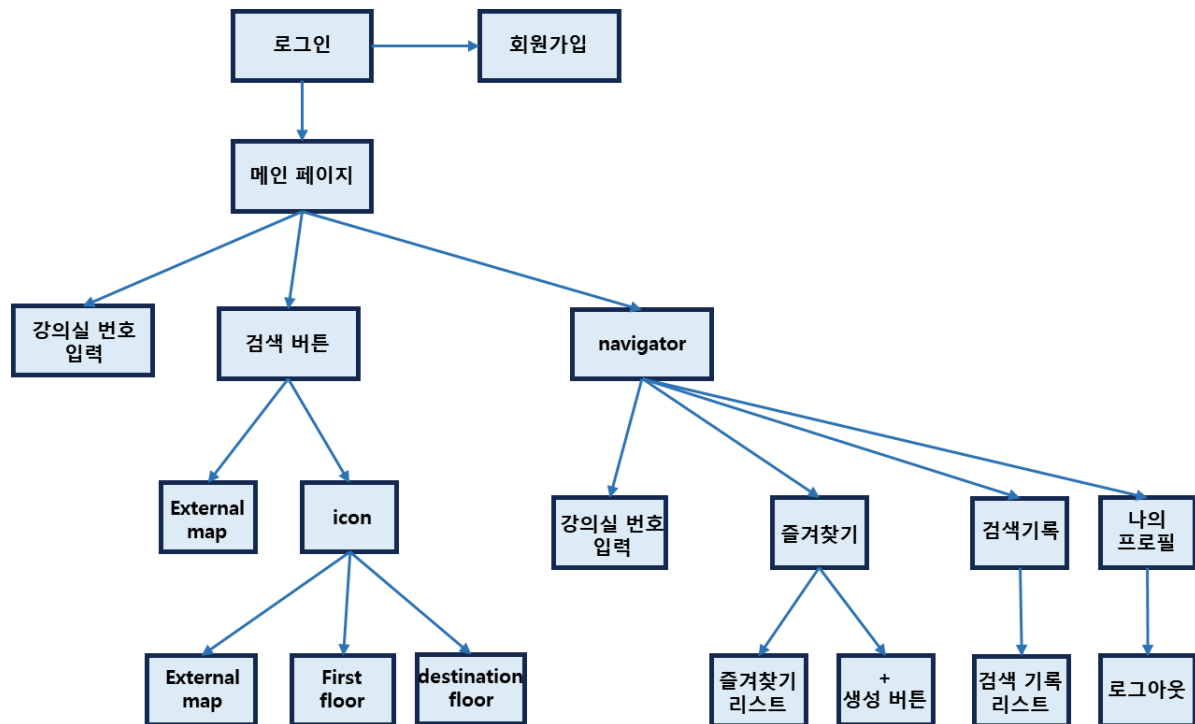
- **SKKU CAMPUSMAP**

SKKU CAMPUSMAP은 skku.edu 에 나와있는 정보로, 학교 자체에서 제공하는 애플리케이션이다. 인문사회과학캠퍼스와 자연과학캠퍼스의 건물 번호가 표시된 kakaomap을 사용하여, 건물 까지의 길을 찾게 해준다. 하지만 강의실의 위치 등 건물 내부에서의 정확한 위치를 찾을 수 없다. 이로써, SKKU MAP과의 주요 차이점은 강의실까지의 내비게이션 기능에 있다.

- **SKK 강의종합안내도**

SKK 강의 종합안내도는 역시 학교 자체에서 제공하는 정보이다. 각 건물의 층별 위치가 있으나, 우리가 필요한 강의실 외에 다른 필요없는 정보들이 포함되어 있으며, 업데이트되지 않아, 잘못 표기된 강의실 번호가 많았다. 또한, 강의실의 위치를 대략적으로 알 수는 있지만, 경로를 안내해주는 기능은 없어 건물의 엘리베이터나 계단의 위치 등 건물 구조에 익숙하지 않은 사람들에게는 효율적인 길을 찾기 쉽지 않았다.

2.1.2 Overall Structure



[그림 1] 프로그램 구조

2.2 Product Function

2.2.1 로그인 및 회원가입

프로그램을 시작하면 맨 처음으로 나오는 화면은 로그인 페이지이다. 즐겨찾기, 검색 기록을 유저별로 보이기 위해서 로그인을 해야한다. 유저의 아이디와 비밀번호를 넣고, 버튼을 눌러 로그인을 한다. 등록된 계정이 없는 경우, 회원가입을 한 후, 로그인을 다시 진행한다.

2.2.2 메인 페이지(강의실 번호 입력)

로그인을 하고 나서, 유저는 메인 페이지를 볼 수 있다. 메인 페이지에는 강의실 번호를 입력할 수 있는 칸과 검색 버튼이 있다. 찾아갈 강의실 번호를 입력하고 검색 버튼을 누르면, 곧바로 건물 외부 네비게이션 화면으로 넘어가게 된다. 모든 페이지의 기본 네비게이션 바에는 검색기록, 즐겨찾기 기능이 있다. 이는 2.2.6과 2.2.7에서 더 자세히 설명하도록 한다.

2.2.3 건물 외부 네비게이션

강의실 번호를 입력하고 나면, 건물 외부 네비게이션 화면이 나온다. 이를 통해 현재 내 위치 혹은 설정할 출발 지점에서부터 입력한 강의실 번호가 있는 건물까지의 건물 외부 경로가 표시된다.

2.2.4 건물 내부 네비게이션

건물 외부 네비게이션이 보이는 스크린의 하단 바에는 세개의 아이콘이 있다. 하나는 'show external map', 두번째는 'first floor', 마지막 아이콘은 'destination floor' 이다. 'show external map'에서는 2.2.3에 기술된 건물 외부 네비게이션을 볼 수 있다. 'first floor'아이콘을 누르면, 목적 강의실이 있는 건물의 1층 구조가 보이게 된다. 또한, 강의실까지의 가장 효율적인 경로에 있는 출입구와 엘리베이터, 계단의 경로를 알 수 있다. 해당층에 도착하면, 'destination floor' 아이콘을 누른다. 해당 아이콘을 클릭하면, 목적 강의실이 있는 층의 사진을 볼 수 있다. 엘리베이터나 계단을 통해 올라온 후, 해당 층에서는 최종 목적지인 강의실을 어떻게 갈 수 있는지 동선이 표시된다.

2.2.5 검색기록

사용자는 최근 검색한 10개의 항목을 리스트 형식으로 확인할 수 있다. 사용자는 리스트에서 원하는 검색 항목을 선택할 수 있으며, 사용자가 선택한 검색 항목을 클릭하면, 건물 외부 경로부터 하단 바의 아이콘들을 이용하여 쉽게 다시 확인할 수 있다. 이와 같이, 검색 기록은 사용자가 편리하게 이전 검색 항목에 다시 접근하고, 선택한 항목을 클릭하여 해당 장소로 이동할 수 경로를 다시 보여주는 기능을 제공한다.

2.2.6 즐겨찾기

사용자는 자주 찾는 강의실을 미리 등록할 수 있다. 이를 통해 매번 검색할 필요 없이 자주 이용하는 장소로 쉽게 접근할 수 있습니다. 스크린 하단의 + 버튼을 눌러 자주 가는 강의실 번호를 입력하여 즐겨찾기 리스트를 생성할 수 있다.

2.2.7 나의 프로필

나의 프로필을 클릭하면 사용자의 로그인 정보가 뜬다.

2.3 User Classes and Characteristic

2.3.1 User

이 프로그램의 사용자는 특정 대학교를 방문객과 대학생으로 제한된다. 예를 들어, 성균관대학교 방문객 또는 학생들일 수 있다. 사용자가 일반적으로 대학 캠퍼스 구조에 대한 지식이 제한적이거나 익숙하지 않다고 가정한다. 원활한 사용을 위해 한국어 또는 영어에 기본적인 지식이 있다고 가정하며 기기에서 애플리케이션을 다운로드하고 사용할 수 있는 능력을 갖추고 있다고 가정한다.

2.3.2 System Manager

본 애플리케이션의 시스템 관리자는 시스템에 대한 전문 지식을 갖춘 사람으로 제한된다. 그들은 시스템에 대해 충분히 알고 전체 시스템을 이해할 수 있어야 한다. 시스템 오류가 발생하거나 그러한 문제를 감지할 수 있는 능력을 전제로 한다. 그들은 해당 버그나 문제를 처리할 수 있어야 한다. 시스템 관리자는 새로운 시스템에 변경 사항을 통합할 수 있는 능력을 가정한다. 이러한 요구 사항을 충족하기 위해 시스템 관리자는 컴퓨터 공학, 네트워크 또는 시스템에 대한 전문 지식이나 해당 지식과 자격증을 갖추거나 그와 동등한 지식을 가져야 하며, 또한 소프트웨어 윤리의식도 가져야 한다.

2.4 Operation Environment

2.4.1 Hardware

- **Client:** 사용자는 운영체제, 프로세서 종류에 제한 없이 사용할 수 있다.

- **Server:** AWS EKS에 의해 노드가 관리 되고, 이 노드와 그 아래서 운영되는 Pod은 아래 스펙을 갖고 있다.
 - AMD 기반 Architecture
 - t3.medium 타입 인스턴스
 - 최소 .5 vCPU 이상, 250 MiB Mem 이상 (per Pod)

2.4.2 Software

- **Client:**

사용자는 운영체제, 브라우저 종류에 제한 없이 사용할 수 있다. 하지만 정상적으로 서비스를 사용하기 위해 아래 조건을 만족 시켜야 한다.

- 가용 메모리 최소 128MB 이상
- 가용 저장 공간 64MB 이상

- **Server:**

AWS EKS에 의해 관리되는 인스턴스와 운영 환경은 아래 조건을 만족시킨다.

- Linux 기반 운영 체제 (Amazon Linux 2)

이외에 Python, Docker, Database 등 운영은 AWS에 의해 자동으로 이루어진다.

2.5 Design and Implementation Constraints

시스템 설계 및 구현 방향에 대한 세부 사항은 다음과 같다.

1. 외부 건물 길 찾기는 개발 시간을 줄이기 위해 외부 네비게이션 API를 사용한다.
2. 백엔드 프레임워크는 FastAPI 0.110.0를 사용하고 Python 3.9 이상의 버전에서 구동되어야 한다.
3. 이외 라이브러리는 파이썬 라이브러리 관리 툴을 이용하여 일괄 설치가 가능해야 한다.
4. 데이터베이스는 용도에 따라 관계형 데이터베이스와 비관계형 데이터베이스를 같이 사용한다.
5. 배포되는 서버 / 컨테이너는 최소 1 vCPU, 1GiB RAM 이상의 리소스를 가지고 있어야 하고, 1GB의 저장소 가 연결되어 있어야 한다.
6. 서버가 동작하는 운영체제는 Linux 기반이어야 한다.

7. 클라이언트와 서버는 HTTP(S) 프로토콜로 통신한다.
8. 프론트엔드와 백엔드는 독립적으로 운영되어 서로의 환경에 영향을 받지 않도록 해야 한다.

2.6 User Documentation

1. 운영체제 최소 요구사항을 명시해야 한다.
2. 기기 최소 하드웨어 요구 사항을 명시해야 한다.
3. 개인 정보 동의 정책을 명시해야 한다.
4. 보안적으로 데이터가 안전하게 보관된다는 것을 명시해야 한다.
5. 주요 기능과 같은 핵심 유스케이스에 대하여 스크린샷과 함께 사용법을 설명해야 한다.
6. 개발자 연락처나 게시판을 열어 문제가 개선 방안을 받을 수 있도록 한다.

2.7 Assumptions and Dependencies

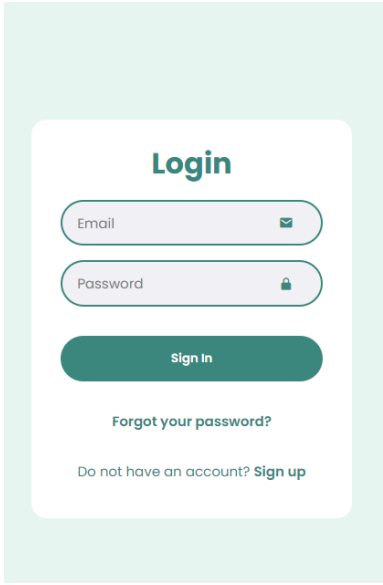
테스트 운영체제 환경은 Windows 10, MacOS 14, Ubuntu 22.04로 구성되며, 테스트 브라우저로는 Chrome 118.0.5993.88, Safari 17.4.1, Arc 1.40.1을 사용한다. 배포 서버 환경은 Amazon Linux 2이며, EC2 인스턴스(t3.medium, 2 vCPU, 4 GiB Mem.)에서 운영된다.

3. Specific Requirements

3.1 External Interface Requirements

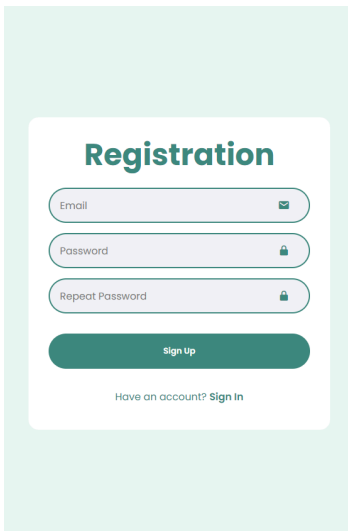
3.1.1 User Interface

[표 3] Interface of Login page

Name	로그인
Purpose / Description	웹사이트를 이용할 때, 자신의 계정을 이용하기 원하는 경우 로그인을 할 수 있도록 한다. 사용자는 이 페이지에서 이메일과 비밀번호를 입력하여 로그인을 하거나, 비밀번호를 잊었다면 비밀번호를 찾을 수 있고, 계정이 없다면 신규 회원가입을 할 수 있다.
input source / output destination	Client/Server
Time / Velocity	로그인은 1초 이내에 이루어져야 한다.
Relationship with other inputs and outputs	강의실 번호 입력 화면에서의 전환
GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 계정을 갖고있는 사용자는 이메일과 비밀번호를 입력한 후, “sign in” 버튼을 눌러 로그인을 시도할 수 있다. 2. 계정을 갖고 있는 사용자 중 비밀번호를 잊은 사용자는 “Forgot your password?”를 클릭하여 이메일을 입력하고, 이메일을 통해 비밀번호를 전송받을 수 있다. 3. 시스템은 “sign in” 버튼이 클릭되었을 때 이메일과 비밀번호가 올바르게 입력되었는지 확인한다. 4. 사용자가 계정을 가지고 있지 않다면, “Sign up”을 클릭하여 회원가입 페이지로 이동할 수 있다.
Data type	Text

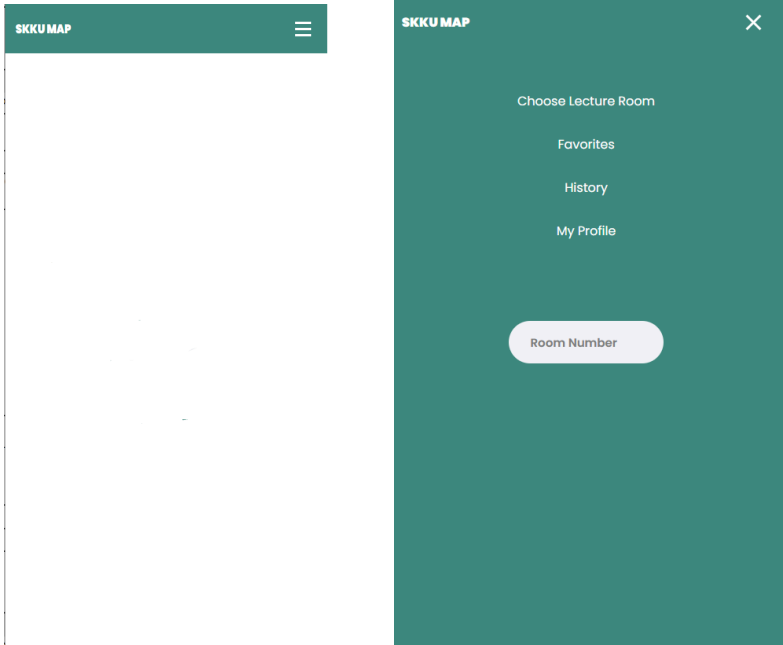
Instruction type	API 호출, 화면 전환, 이메일 전송
------------------	-----------------------

[표 4] Interface of Register page

Name	회원가입
Purpose / Description	사용자는 이 페이지에서 신규 회원가입을 할 수 있다.
input source / output destination	User Input/API Server
Time / Velocity	이메일 전송은 10초 이내에 이루어져야 한다. 계정 생성은 1초 이내에 이루어져야 한다.
Relationship with other inputs and outputs	회원가입 화면에서의 전환
GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 회원가입을 위해 이메일과 비밀번호, 비밀번호 확인 칸을 채울 수 있다. 2. 사용자가 Sign up 버튼을 클릭했을 때, 시스템은 비밀번호와 비밀번호 확인이 올바르게 입력되었는지 확인하고, 회원가입 정보를 저장한다. 3. 사용자는 이 페이지에서 Sign in 버튼을 클릭하여 다시 로그인 화면으로 돌아갈 수 있다.
Data type	user input text
Instruction type	save registration info


[표 5] Interface of Top Tab Navigator

Name	Top Tab Navigator
Purpose / Description	사용자는 상단에 배치된 이 Navigator를 통해 페이지를 전환할 수 있다.

input source / output destination	사용자의 클릭/알맞은 화면으로의 전환
Time / Velocity	화면 전환은 1초 이내에 이루어져야 한다.
Relationship with other inputs and outputs	None
GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 상단의 3줄로 되어있는 네비게이션 버튼을 클릭하여 탭을 열 수 있다. 2. 이 화면에서 강의실 선택, 즐겨찾기, 최근 기록, 내 프로필 버튼을 클릭하여 해당하는 페이지로 이동할 수 있다.
Data type	None

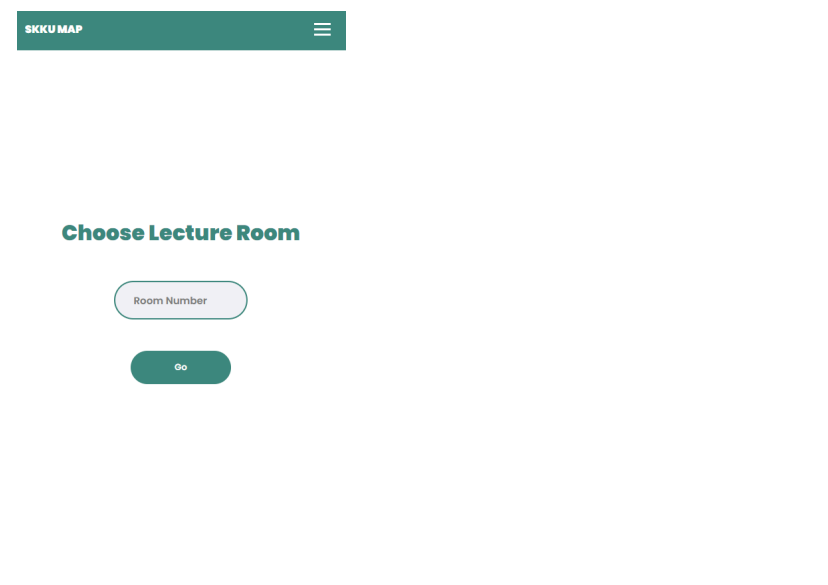
[표 6] Interface of Bottom Tab Navigator

Name	Bottom Tab Navigator
Purpose / Description	사용자는 길을 찾을 때 하단의 옵션을 선택하여 지도를 전환할 수 있다. 외부지도/목적지 1층의 지도/ 목적지 층의 지도의 3가지 옵션을 선택할 수 있다.
input source / output destination	사용자의 클릭/알맞은 화면으로의 전환
Time / Velocity	지도 화면의 전환은 1초 이내에 이루어져야 한다.
Relationship with other inputs and outputs	User's Destination input

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 하단의 3가지 옵션중에서 하나를 선택할 수 있다. 2. 사용자가 Show External Map 버튼을 누르면 외부지도 페이지를 보여준다. 3. 사용자가 First Floor 버튼을 누르면 목적지 건물의 1층 지도를 보여준다 4. 사용자가 Destination Floor 버튼을 누르면 목적지 층의 지도를 보여준다. 한편, 목적지가 1층이라면 이 옵션은 나타나지 않는다.

[표 7] Interface of the Search Lecture Room page

Name	강의실 번호 검색
Purpose / Description	사용자는 강의실 번호를 입력하고, Go 버튼을 눌러 시스템에게 강의실 번호를 넘겨주고, 외부 지도 페이지로 넘어갈 수 있다.
input source / output destination	사용자 입력
Time / Velocity	Go 버튼을 누르고 1초 이내에 지도화면으로 전환이 되어야 한다.
Relationship with other inputs and outputs	None

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 자신이 향하는 강의실 번호를 입력하고, Go 버튼을 누른다. 2. Go 버튼을 누르면 외부 지도 페이지로 넘어간다.
Data type	Number

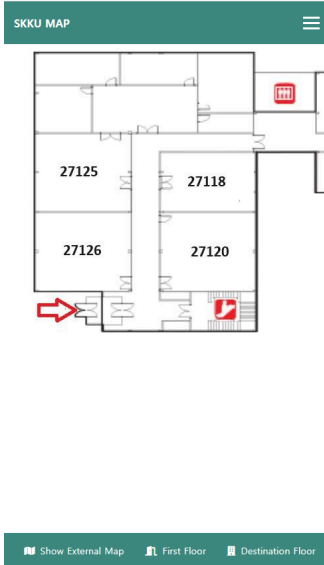
[표 8] Interface of the external map page

Name	외부 지도 페이지
Purpose / Description	사용자는 외부 지도를 보고 자신이 원하는 강의실이 있는 건물까지 가는 길을 안내 받을 수 있어야 한다.
input source / output destination	None
Time / Velocity	외부 지도는 1초 내로 페이지에 나타나야 한다.
Relationship with other inputs and outputs	사용자의 강의실 번호 입력을 사용한다.

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 자신이 선택한 강의실이 있는 건물까지 가는 길을 안내받을 수 있다. 2. 외부 지도는 Kakao Map을 사용한다.
Data type	None

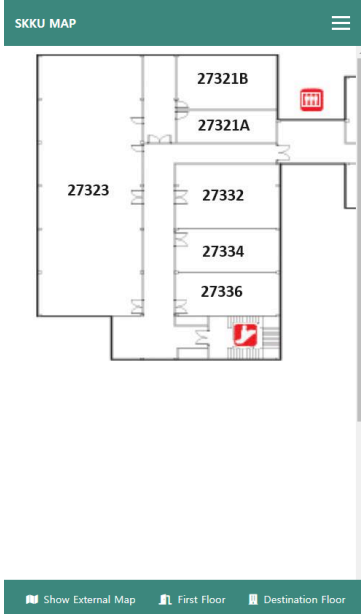
[표 9] Interface of First Floor Page

Name	1층 페이지
Purpose / Description	사용자가 목적지의 건물에 입장하면, 첫번째 층의 지도와 강의실 까지 가는 길을 화면에 보여주어 사용자가 목적지 까지 가는 길을 안내받을 수 있도록 한다.
input source / output destination	None
Time / Velocity	지도가 화면에 1초 이내에 나타나야 한다.
Relationship with other inputs and outputs	목적지 건물 정보를 알기 위해 User input Number를 사용한다.

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 건물 내부 지도를 보며 자신이 선택한 강의실까지 향하는 길을 안내받을 수 있다. 2. 지도는 확대 및 축소가 가능하다.
Data type	Map Image

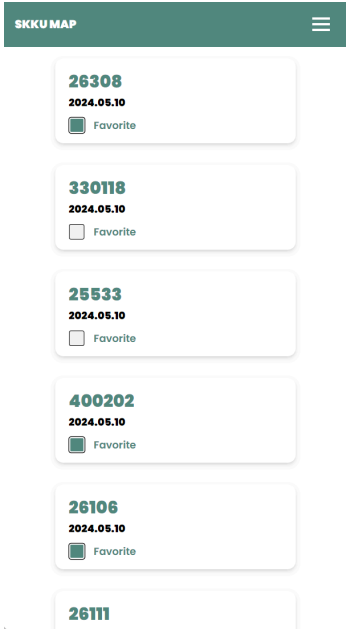
[표 10] Interface of Destination Floor page

Name	목적지 경로 페이지
Purpose / Description	사용자가 건물의 목적지 층에 도착하면, 목적지 층의 지도와 강의실까지 가는 길을 화면에 보여주어 사용자가 목적지 까지 가는 길을 안내받을 수 있도록 한다.
input source / output destination	None
Time / Velocity	지도가 화면에 1초 이내에 나타나야 한다.
Relationship with other inputs and outputs	목적지 건물 정보를 알기 위해 User Input number을 사용한다.

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 건물 내부 지도를 보며 자신이 선택한 강의실까지 향하는 길을 안내받을 수 있다. 2. 지도는 확대 및 축소가 가능하다.
Data type	Map Image

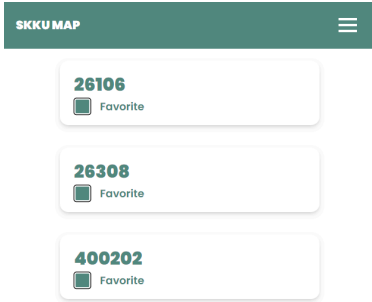
[표 11] Interface of History Page

Name	이전 기록 페이지
Purpose / Description	로그인을 한 사용자가 자신이 최근에 검색한 강의실 번호를 빠르게 재검색 할 수 있도록 돕는 페이지이다. 이 곳에는 사용자가 최근 검색한 강의실 번호의 목록이 나타나며, 사용자는 강의실 번호를 바로 선택하여 목적지까지의 경로를 알아낼 수 있다.
input source / output destination	사용자의 클릭/목적지 까지 안내하는 지도 화면
Time / Velocity	사용자의 최근 검색기록은 1초 이내에 화면에 나타나야 한다.
Relationship with other inputs and outputs	사용자가 새롭게 강의실 번호를 입력하면, 이 페이지는 업데이트 되어 새로운 검색 정보를 반영해야 한다.

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 자신이 최근에 검색한 강의실 번호의 목록을 열람할 수 있다. 2. 사용자는 강의실 번호의 목록중 하나를 선택하여 해당 강의실까지 향하는 경로를 안내하는 페이지로 이동할 수 있다.
Data type	Text

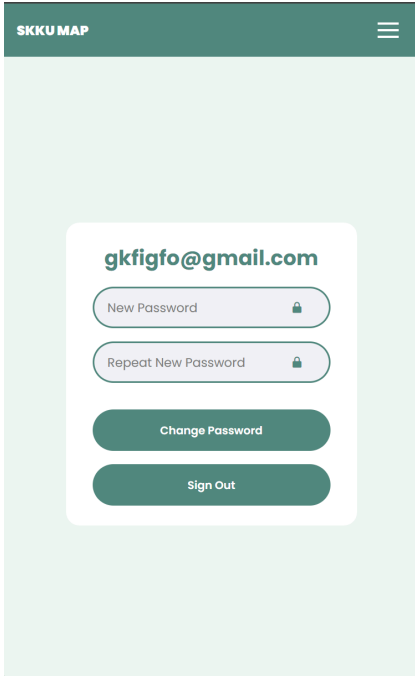
[표 12] Interface of Favorites page

Name	즐거찾기 페이지
Purpose / Description	로그인을 한 사용자가 자신이 자주 가는 강의실 번호를 등록하고 그 강의실 번호를 빠르게 재검색 할 수 있도록 돕는 페이지이다. 이 곳에는 사용자가 즐겨찾기로 등록해 둔 강의실 번호의 목록이 나타나며, 사용자는 강의실 번호를 바로 선택하여 목적지까지의 경로를 알아낼 수 있다.
input source / output destination	사용자의 클릭/목적지 까지 안내하는 지도 화면
Time / Velocity	사용자의 즐겨찾기 목록은 1초 내에 나타나야 한다.
Relationship with other inputs and outputs	None

GUI Sketch	
Format and configuration of screen	<ol style="list-style-type: none"> 1. 사용자는 자신이 등록해둔 즐겨찾기 강의실의 목록을 열람할 수 있고, 즐겨찾기 해제 및 등록이 가능하다. 2. 사용자는 강의실 번호의 목록중 하나를 선택하여 해당 강의실까지 향하는 경로를 안내하는 페이지로 이동할 수 있다.
Data type	Text

[표 13] Interface of My Profile page

Name	내 프로필 페이지
Purpose / Description	사용자는 계정 비밀번호와 같은 자신의 정보를 이 화면에서 열람할 수 있고, 정보를 변경할 수 있다.
input source / output destination	None
Time / Velocity	사용자의 정보는 1초 이내에 화면에 나타나야 한다.
Relationship with other inputs and outputs	회원가입시 입력한 정보는 이 페이지에서 열람 및 수정될 수 있다.

GUI Sketch	
Format and configuration of screen	사용자는 이 페이지에서 자신의 계정 정보를 열람할 수 있다. 사용자가 비밀번호 변경을 선택하면, 자신의 계정의 비밀번호를 변경할 수 있다.
Data type	Text

3.1.2 Hardware Interfaces

[표 14] hardware Interface

Name	Applicable device for the system
Purpose / Description	사용자는 64bit CPU, 512MB의 RAM, 100MB drive space 이상의 하드웨어 사양을 가진 모바일 기기 또는 PC로 접속해야 한다.

3.1.3 Software Interfaces

[표 15] Software Interface

Name	AWS Database
Purpose / Description	데이터 관리를 위한 쿼리 입력 및 출력을 위해 사용된다.
Input Source / Output destination	Host Server/User, User/Host Server, User/User
Accuracy	Serverless 기반 AWS 데이터베이스로 99.99999% 이상의 응답 성공 보장

Unit	Query
Time / Velocity	Instant reaction
Relationship with other input/outputs	Related to all input/output from server
Format and configuration of screen	None
Format and configuration of window	None
Data type	Query
Instruction type	Query statement
Exit Message	None

3.1.4 Communication Interfaces

[표 16] communication interface

Name	Client and Server
Purpose / Description	클라이언트가 입력받은 강의실 번호를 서버에 전달하면, 서버는 해당 강의실 까지 가는 길을 안내하는 지도를 클라이언트로 전달한다.
Input Source / Output destination	Client/Server
Unit	packet
Time / Velocity	0.1초 이내
Relationship with other input/outputs	Related to all input/output from server
Format and configuration of screen	None
Format and configuration of window	None
Data type	Query
Instruction type	Query Statement
Exit Message	None

3.2 Function Requirements

3.2.1 Use case

[표 17] use case for register

Name	Applicable device for the system
Actor	회원가입을 하지 않은 사용자
Description	회원가입을 하지 않은 사용자가 회원 가입을 한다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자는 로그인 페이지에서 Register 버튼을 누른다. 2. 사용자가 회원가입 페이지에 들어오면, 자신의 이메일과 비밀번호, 비밀번호 확인란에 알맞은 값을 입력한다. 3. “Sign up” 버튼을 눌러 사용자는 회원가입을 완료한다. 만일 비밀번호와 비밀번호 확인란에 적힌 값이 일치하지 않는다면, 2번 과정을 다시 시도한다. 4. 사용자의 계정 정보는 시스템으로 보내진다.
Pre-condition	<p>네트워크 사용이 가능해야 한다.</p> <p>사용자는 자신의 이메일 주소를 가지고 있어야 한다.</p> <p>사용자는 해당 이메일로 가입되어 있지 않은 상태여야 한다.</p>
Post-condition	사용자의 계정 정보가 데이터베이스에 저장되어야 한다.
Assumptions	None

[표 18] use case for login

Name	Applicable device for the system
Actor	회원가입을 한 사용자
Description	사용자가 로그인을 시도한다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자가 로그인 페이지에 들어온다. 2. 사용자는 자신의 이메일과 비밀번호를 입력한다. 3. 사용자는 로그인 버튼을 클릭한다. 4. 사용자가 입력한 정보는 시스템으로 전송된다. 5. 시스템은 해당 정보가 유효한지 확인한다. 유효한 정보란, 이메일이 데이터베이스에 저장되어 있고, 데이터베이스에 저장된 해당 이메일을 키로 하는 비밀번호 값이 사용자가 입력한 비밀번호 값과 일치함을 의미한다. 6. 만일 사용자가 입력한 정보가 유효하다면, 로그인에 성공하고, 그렇지 않다면, 로그인에 실패하여 2번과정으로 돌아간다.
Pre-condition	<p>사용자는 회원가입을 완료한 상태여야 한다.</p> <p>사용자가 자신의 계정 정보에 대하여 알고 있어야 한다.</p> <p>네트워크 사용이 가능해야 한다.</p>
Post-condition	사용자는 자신의 계정으로 로그인되어, 계정을 이용한 서비스에 접근할 수 있어야 한다.

Assumptions	None
-------------	------

[표 19] use case for Password change

Name	Applicable device for the system
Actor	로그인 한 사용자
Description	사용자는 자신의 계정 정보를 열람할 수 있고, 이 곳에서 비밀번호를 변경할 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자는 자신의 이메일 주소를 확인한다. 2. 사용자가 비밀번호 변경을 원한다면, new password 칸에 새로 변경하고 싶은 비밀번호를 입력한다. 3. new password check 칸에 새로 변경할 비밀번호를 재확인한다. 4. 사용자가 Password change 버튼을 클릭하면, 비밀번호가 변경된다. 5. 만약 사용자가 logout 버튼을 누르면, 사용자는 로그아웃된다.
Pre-condition	네트워크 사용이 가능해야 한다.
Post-condition	사용자의 변경된 비밀번호가 데이터베이스에 올바르게 저장되어야 한다.
Assumptions	None

[표 20] use case of inserting the lecture room number

Name	Applicable device for the system
Actor	모든 사용자
Description	사용자가 자신이 가고자 하는 강의실의 번호를 입력한다.
Normal Course	<ol style="list-style-type: none"> 1. 메인 화면에서, 사용자는 자신이 가고자 하는 강의실의 번호를 입력한다. 이때, 강의실의 번호는 실제로 존재하는 강의실의 번호여야 한다. 2. 사용자는 Go 버튼을 클릭한다.
Pre-condition	사용자는 존재하는 강의실의 번호를 입력해야 한다.
Post-condition	사용자가 입력한 강의실의 번호가 시스템에 전달되어야 한다.
Assumptions	None

[표 21] use case of finding the most efficient route - outside the building

Name	Applicable device for the system
------	----------------------------------

Actor	사용자
Description	사용자가 외부에서의 길찾기 지도를 열람한다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자는 메인 페이지에서 강의실의 번호를 입력하고, Go 버튼을 눌러 외부 지도 페이지로 넘어간다. 2. 사용자는 외부에서 강의실이 있는 건물까지 가는 빠른 길을 볼 수 있다. 3. 만일 건물에 도착하여 건물 내부 지도를 열람하고 싶다면, Inside 버튼을 클릭하여 지도 화면을 전환할 수 있다.
Pre-condition	Map API가 사용가능한 상태여야 한다. 네트워크 사용이 가능해야 한다.
Post-condition	사용자가 지도 화면 전환을 시도하면, 화면이 올바르게 전환되어야 한다.
Assumptions	None

[표 22] use case of finding the most efficient route - inside the building

Name	Applicable device for the system
Actor	사용자
Description	사용자가 건물 내부에서의 길찾기 지도를 열람한다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자는 외부 지도 열람 중 내부 지도 열람 버튼을 눌러 내부 지도 열람 페이지로 넘어간다. 2. 사용자는 건물 입구에서 강의실까지 가는 빠른 길을 볼 수 있다. 3. 만일 건물 외부 지도를 다시 열람하고 싶다면, Outside 버튼을 클릭하여 지도 화면을 전환할 수 있다.
Pre-condition	Map API가 사용가능한 상태여야 한다. 네트워크 사용이 가능해야 한다.
Post-condition	사용자가 지도 화면 전환을 시도하면, 화면이 올바르게 전환되어야 한다.
Assumptions	None

[표 23] use case of favorite

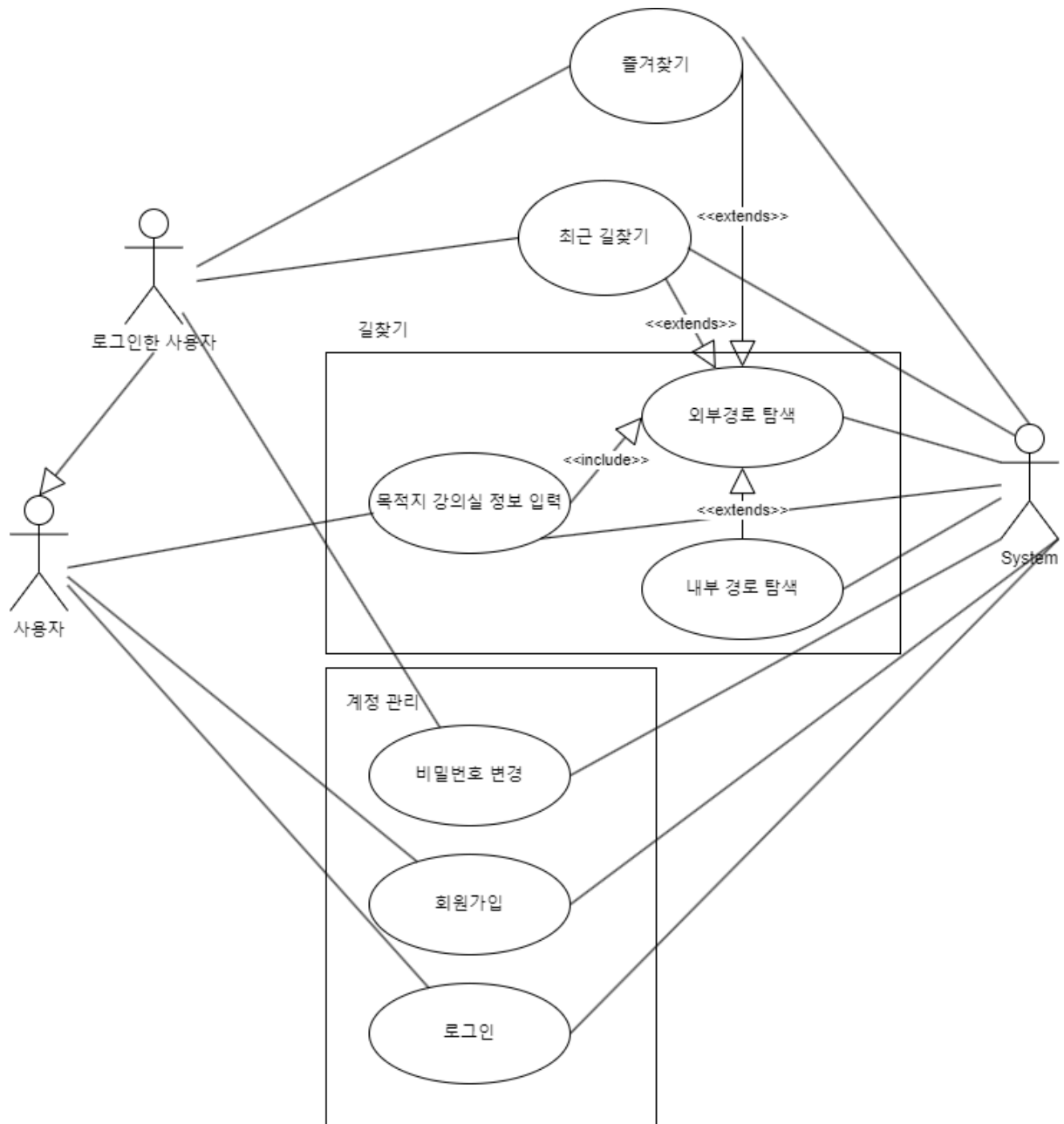
Name	Applicable device for the system
Actor	로그인한 사용자
Description	사용자가 즐겨찾기를 설정하거나 즐겨찾는 강의실까지 가는 길을 빠르게 찾을 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 로그인한 사용자가 상단 바에서 Favorite을 클릭한다. 2. 사용자는 Favorite 목록을 열람할 수 있고, Favorite 목록에

	강의실을 추가하거나 삭제할 수 있다. 3. Favorite 목록에 있는 강의실을 클릭하면, 해당 강의실까지 가는 길을 안내하는 페이지로 넘어간다.
Pre-condition	네트워크 사용이 가능해야 한다.
Post-condition	사용자가 Favorite 목록을 업데이트 하면, 해당 정보가 데이터베이스에 저장되어야 한다.
Assumptions	None

[표 24] use case of Recent

Name	Applicable device for the system
Actor	로그인한 사용자
Description	사용자가 최근 길을 찾은 강의실을 열람하고 해당 강의실까지 가는 길을 빠르게 찾을 수 있도록 한다.
Normal Course	<ol style="list-style-type: none"> 1. 로그인한 사용자가 상단 바에서 Recent를 클릭한다. 2. 사용자는 최근 길을 찾은 강의실 목록을 열람할 수 있다. 3. Recent 목록에 있는 강의실을 클릭하면, 해당 강의실까지 가는 길을 안내하는 페이지로 넘어간다.
Pre-condition	네트워크 사용이 가능해야한다.
Post-condition	None
Assumptions	사용자가 길을 찾은 정보가 데이터베이스에 저장되어있다.

3.2.2 Use-Case Diagram

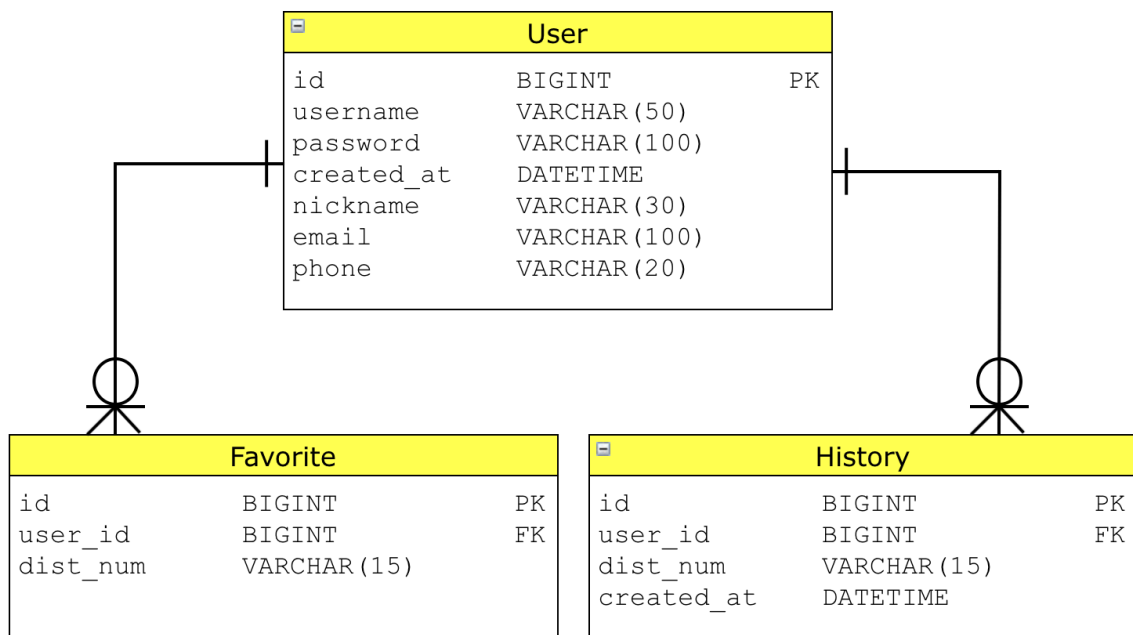


[그림 2] use case diagram

3.2.3 Data Dictionary (RDB)

RDB에는 사용자 정보와 경로 탐색 기록, 자주 찾는 강의실 정보를 주로 저장한다.

아래는 ERD와 테이블 설명 표이다.



[그림 3] RDB ERD

[표 25] User Data Dictionary

Field	Type	Constraint	Description
id	bigint	PK Not null	User Identifier
username	varchar(50)	Not Null	ID
password	varchar(100)	Not Null	PW (Encrypted)
nickname	varchar(30)		Display name
email	varchar(100)		User Information
phone	varchar(20)		Phone
created_at	datetime	Not Null	Sign up time

User Data Dictionary는 사용자의 정보를 담고있다.

[표 26] Favorite Data Dictionary

Field	Type	Constraint	Description
id	bigint	PK Not null	Identifier

user_id	bigint	FK Not null	Reference ID
dist_num	varchar(15)	Not Null	Lecture Room No.

Favorite Data Dictionary는 사용자가 자주 찾는 강의실 정보가 들어있으며, User : Favorite가 1:N 관계이다.

[표 27] History Data Dictionary

Field	Type	Constraint	Description
id	bigint	PK Not null	Identifier
user_id	bigint	FK Not null	Reference ID
dist_num	varchar(15)	Not Null	Lecture Room No.
created_at	datetime	Not Null	Request Time

History Data Dictionary는 사용자가 경로를 탐색한 기록을 저장하며, User : History가 1:N 관계이다.

3.3 Non-Functional Requirements

3.3.1 Product Requirements

3.3.1.1 Usability Requirements

- 사용자는 별도의 설명을 확인할 필요 없이 기능을 사용할 수 있도록 직관적인 인터페이스를 제공해야 한다. 또한 전반적으로 녹색과 흰색을 기반으로 구성하여 시스템 전체적으로 일관된 인터페이스를 제공한다.
- 강의실 번호를 기반으로 길을 찾는 외국인을 대상으로 고려하였으므로 영어를 기본 언어로 지원한다.
- 계정을 사용하여 사용자마다 “favorites”와 “history”를 제공함으로써 시스템의 접근성과 편의성을 제공한다.

3.3.1.2 Efficiency Requirements

건물 내부의 강의실 정보를 반환하는 API의 응답 시간은 1,000ms 이하이어야 한다.

내부 길 찾기 API의 응답 시간은 **3,000ms** 이하이어야 한다. 데이터가 차지하는 용량이 크지 않고, 인메모리에서 관리해도 되므로 데이터는 서버 실행 시 인메모리에 모두 로딩해둔다.

- 이는 곧 사용자가 접속할 때는 이미 데이터가 로딩된 상태이기 때문에 API 호출에 있어서 CPU, 메모리 부담을 덜 수 있다.

사용자가 최근에 검색해본 경로는 캐시해서 더 빨리 제공할 수 있도록 한다.

- Cache Miss : 150 ms ~ 600 ms
- Cache Hit : 15 ms ~ 50 ms

- 캐시를 사용하면 80% 정도 응답 시간을 개선할 수 있다.

3.3.1.3 Dependability Requirements

존재하는 강의실에 대해서는 항상 최단 경로가 포함된 사진을 받을 수 있어야 한다. 사용자가 없는 강의실을 입력했을 때에도 오류가 나면 안 되고 적절한 응답을 할 수 있어야 한다. 자동화된 인프라 관리를 테스트하여 자동 인프라 증설이 되는지 확인해야 한다.

3.3.1.4 Security Requirements

모든 API는 인증된 사용자만이 사용할 수 있어야 한다. 사용자의 개인 정보는 데이터베이스에 안전하게 보호되어야 한다. 사용자의 비밀번호는 특히 단방향 암호화를 거쳐 역추적이 불가능하도록 해야 한다. CORS 이슈를 고려하여 일부 Origin은 허용하되, 프론트 엔드 서버의 Origin만 허용하여 불필요한 접속을 막아야 한다.

3.3.2 Organization Requirements

3.3.2.1 Environmental Requirements

해당 사이트는 일반적인 운영체제, 브라우저에서는 모든 기능을 제공해야 한다.

- Linux 기반, Window 기반, MacOS, iOS, Android 등
- Chrome, Safari, Firefox, Arc, Microsoft Edge 등

3.3.2.2 Development Requirements

- 웹 UI/UX 의 전체적인 디자인은 **Figma**를 통해 진행한다.
- 사용성을 1순위로 하여 길 찾기 기능을 직관적으로 쉽게 이용할 수 있도록 한다.
- **React** 를 사용해 해당 사이트의 **Frontend** 개발을 진행하도록 한다.
 - **FastAPI** 를 사용해 해당 사이트의 **Backend** 개발을 진행하도록 한다.
 - 데이터베이스는 용도에 따라 **RDB**와 **NoSQL**을 같이 사용한다.
 - 데이터베이스는 사용자 정보와 캐시 정보를 주로 저장한다.
 - 사용자 정보는 **RDB**로 관리하고, **NoSQL**은 경로 캐시용으로 사용한다.

3.3.3 External Requirements

3.3.3.1 Regulatory Requirements

개발 시 저작권 보호법을 위반 해서는 안 되며 원작자의 요청 시 이를 삭제해야 한다.
 사용자의 개인정보 보관 동의가 반드시 이루어진 후 정보를 저장해야 한다.

- 사용자의 아이디는 인증용으로만 사용됨을 보장해야 한다.
- 사용자의 이메일은 동의 하에 관련 푸시 알림이 갈 수 있다는 사실을 고지해야 한다.
- 비밀번호는 암호화되어 안전하게 보관된다는 사실을 고지해야 한다.
- 사용자의 경로 찾기 기록은 공유되지 않는 다는 사실을 고지해야 한다.

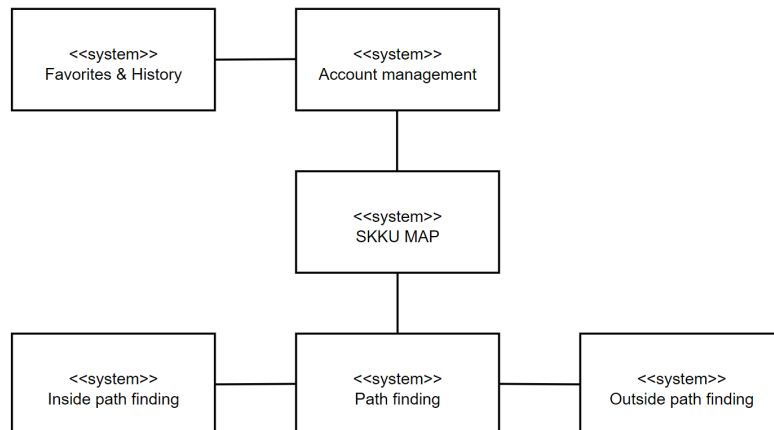
3.3.3.2 Ethical Requirements

- 사용자 데이터를 수집, 처리, 저장할 때 개인정보 보호법 및 관련 규정을 철저히 준수한다.
- 개인정보는 전송 및 저장 시 암호화하여 보호한다.
- 데이터에 대한 접근은 엄격히 통제되어야 하며, 오직 권한이 부여된 사용자만이 접근할 수 있다.

3.3.4 Organizing the Specific Requirements

해당 장에서는 통합 모델링 언어, UML(Unified Modeling Language)을 통해 시스템 모델을 설명한다.

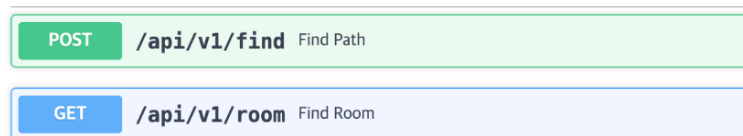
3.3.4.1 Context Model



[그림 4] System Context Model

3.4 API Specification

다음 두 API는 길찾기 기능에 대한 주요 API 명세이다.



[그림 5] 주요 API 명세

1. 길 찾기 API

찾고 싶은 강의실 이름을 넣으면 경로가 포함된 사진의 경로를 반환한다. 또한, 입구 아이디도 같이 반환하여 어떤 입구로 들어가야 하는지도 알 수 있다.

HTTP POST /api/v1/find

[표 28] 길찾기 API의 parameter, response 예시

	Example
Parameters	destination_room (string)
Response	<pre> { "stair": ["", "7p7zzx5xauqaycsjm0t9lyp4u234b9", "", "", "", "e52iu3oa0y5frn8er5dn89dz95y6jh"], "elevator": ["", "scn1n0tvb43jjpe5n2ctbv3vyuzum", "", "", "", "uugklzkeegegmullmq40ty184fp3sh"], "entrance_id": { "stair": 98, "elevator": 98 } }</pre>

2. 강의실 리스트 API

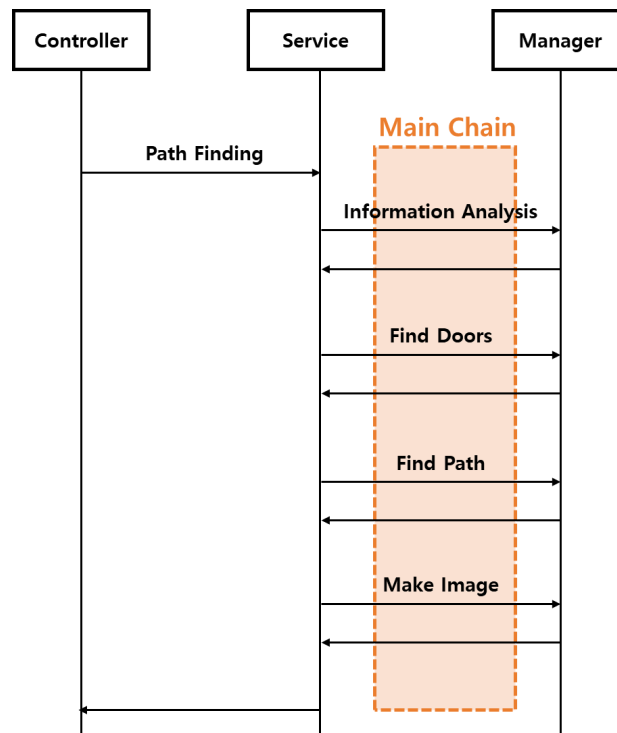
특정 건물 번호에 있는 강의실들을 총별로 나누어 볼 수 있다.

HTTP GET /api/v1/room

[표 29] 강의실 리스트 API의 parameter, response 예시

	Example
Parameters	building_number (string)
Response	<pre> [["26119B", "26119A", "26117", ...], ...]</pre>

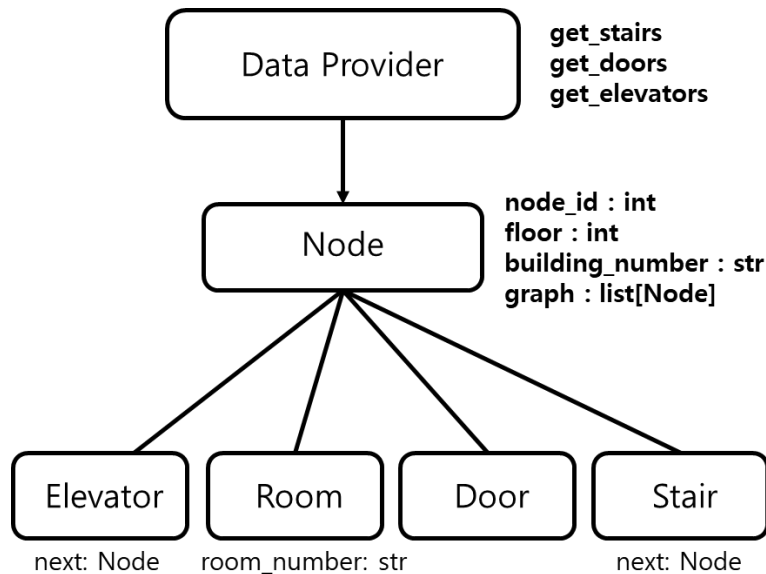
3.5 Main Chain Architecture



[그림 6] Main Chain 구조

계층 구조와 결합시켜 매니징 계층의 기능을 하나의 체인으로 껴서 하나의 기능을 제공하는 체인(Chain)을 만들었다. 각 기능은 독립적이지만, 유기적으로 연결되고 맞물려 하나의 완결된 기능을 제공한다. 각 기능의 동작으로 인해 다른 기능 내부의 상태가 바뀌면 안 되며, (애초에 상태가 없게 설계 되어야 한다.) 같은 데이터에 대해서는 항상 같은 결과가 나와야 한다.

3.6 Database Architecture



[그림 7] 데이터베이스 구조

백엔드에서는 프론트엔드의 요청을 통해 목표 강의실로의 최단 경로를 구하고, 그 경로를 따라 선을 이어 지도에 표시해주는 역할을 한다. 장소 데이터는 위와 같이 크게 다섯 가지로 이루어져 있고, 이 상위(부모) 클래스로 **Node**가 있다. 이 **Node**들은 **Data Provider**를 거쳐 제공된다. 하위(자식) 클래스는 상위 클래스에 의해 공통 필드(`node_id`, `floor`, `building_number`, `graph`)를 가지고, 몇몇 클래스는 특성에 따라 추가적인 데이터를 갖는다.

- Parent Class Fields

- `node_id` : 해당 건물, 층 안에서 유일한 노드의 아이디이다. 이 아이디는 프론트엔드와 공유되어 어떤 입구로 들어가는 것이 최적인지 찾아내는 역할을 한다.
- `floor` : 해당 노드에 몇 층에 위치한 것인지 나타낸다.
- `building_number` : 해당 노드가 어떤 건물에 위치한 것인지 나타낸다.

`graph` : 해당 노드와 연결된 노드 객체를 담고 있는 필드이다. 그래프에서 간선을 나타낸다. 간선을 양방향으로 연결되어야 한다.

- Child(Sub) Class

- **Elevator, stairs** : 건물 내에 위치한 엘리베이터와 계단을 나타낸다. 다음 층의 어떤 노드와 연결되어 있는지 `next_id` 또한 존재한다.

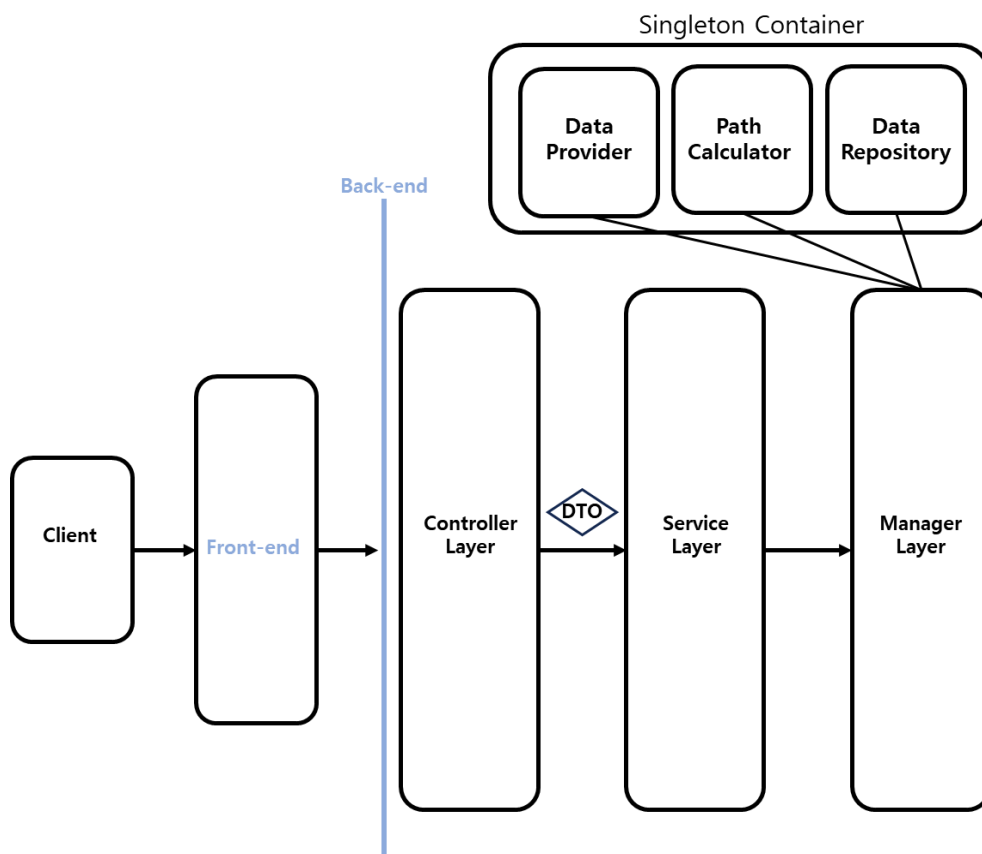
- **Room** : 강의실을 의미한다. 강의실 번호도 함께 저장한다.
- **Node** 클래스에 상속받지 않은 클래스도 존재할 수 있는데, 이 경우에는 특별한 장소, 문 등을 나타내지 않고 경로의 일부로써 경로를 완성시켜주는 노드(복도)이다.

- **Data Provider**

- **get_stairs**: Node 중 계단(stair)를 담아 반환하는 함수이다.
- **get_doors**: Node 중 출입구(door)를 담아 반환하는 함수이다.
- **get_elevators**: Node 중 엘리베이터(elevator)를 담아 반환하는 함수이다.

우리 서비스는 계단과 엘리베이터 두 가지 경우에 대한 최단 경로를 제공하는데, 각 출입구에서 계단과 엘리베이터까지의 최단 경로를 구하는 역할을 한다.

3.7 Server Architecture



[그림 8] 서버 구조

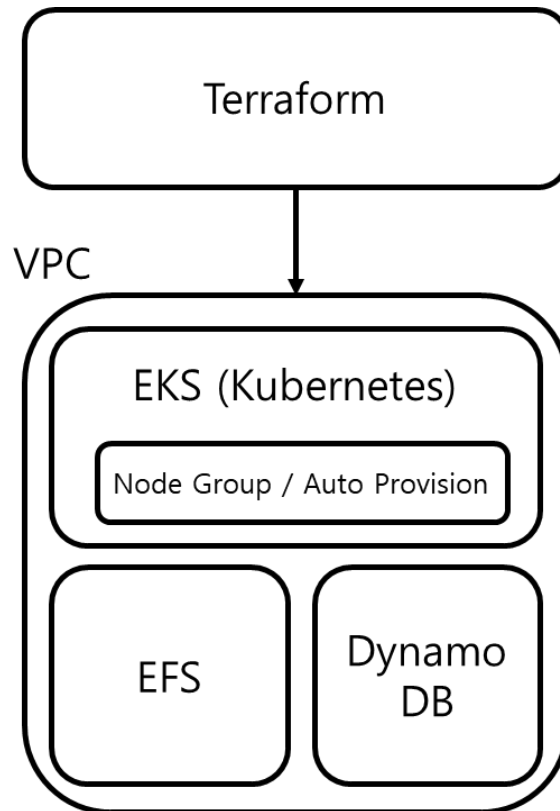
프론트엔드와 백엔드는 완전 독립적으로 운영되도록 하기 위해 **API**로 통신하도록 한다. 백엔드 서버는 계층 구조(**Layered Architecture**)로 이루어져 있어 각 계층은 독립적인 역할을 수행한다.

- 컨트롤러 계층은 사용자의 요청을 직접 받아 메인 로직이 담겨 있는 서비스 계층으로 인계한다.
- 이 과정에서 **DTO**를 이용해 불필요한 정보는 넘기지 않고 데이터의 독립성을 확보하도록 한다.
- 서비스 계층에서는 여러 매니저 계층의 기능을 활용하여 메인 로직을 실행한다.
- 각 계층을 설계할 때는 설계 원칙에 따라 수정에는 닫혀있고 확장에는 열려있게 설계하도록 하며, 수정되더라도 각 계층에는 영향을 끼치지 않도록 완전 독립적인 형태로 개발하도록 한다.
- 특히, 데이터 같은 민감한 정보는 매니저 계층에서만 접근할 수 있도록 하여 각 계층의 역할을 확실히 하고 보안성을 챙길 수 있도록 한다.
- 데이터베이스 접근 객체나 데이터베이스 쿼리 로직은 서비스 계층에서 알 필요 없이 추상화된 기능만 접근하면 되므로 **Repository Pattern**을 활용하여 데이터베이스 관련 기능은 추상화하여 제공하도록 한다.
- 최단 경로를 계산하는 객체는 상태를 가지지 않도록 설계하여 싱글톤 패턴에 의해 관리될 수 있도록 하고, 빠른 응답 속도를 낼 수 있도록 효율적인 알고리즘을 사용하여 구현한다.

데이터를 관리하는 객체와 경로를 계산하는 객체는 상태가 없기 때문에 같은 객체를 모든 사용자가 공유하는 것이 리소스 효율적이다. 따라서 이런 특성을 갖는 객체는 싱글톤 패턴을 이용해 모든 사용자가 같은 객체를 공유하여 효율적으로 리소스를 사용할 수 있도록 한다. 이때 싱글톤으로 관리되는 객체는 **Provider Pattern**을 활용한 싱글톤 컨테이너가 따로 관리하여 필요할 때 적절한 객체를 제공할 수 있도록 한다.

3.8 Infrastructure Architecture

3.8.1 인프라 설계



[그림 9] Infrastructure architecture

인프라의 설계 방향은 개발에 집중할 수 있도록 많은 과정을 자동화 시키는 것을 1원칙으로 한다. 따라서 인프라 구축, 배포, 운영, 인프라 추가 증설을 모두 자동화하여 개발 / 테스트에 집중할 수 있도록 한다.

3.8.2 인프라 구축

- Terraform을 이용하여 IaC(Infrastructure as a Code) 방식으로 인프라를 구축할 수 있도록 한다.
- 계정이 바뀌거나 플랫폼이 바뀌어도 큰 변경 없이 쉽게 동일한 배포 환경을 구축할 수 있다.
- 인프라 추가 증설 : 전통적인 노드 그룹을 사용하지 않고 노드 프로비저닝 오픈소스 **Karpenter**를 이용하여 서버에 부담이 생겨도 자동으로 배포를 할 수 있도록 한다.

3.8.3 인프라 배포

모든 서비스는 도커 컨테이너화 시켜 쉽고 빠르게 배포할 수 있도록 한다.

- 이미지는 도커 허브나 **AWS ECR**에 저장하여 언제든지 내려받아 배포할 수 있도록 한다.
 - **Python**의 가상 환경을 활용해 서비스 환경을 독립적으로 관리하고 **Requirements.txt**로 쉽게 같은 운영 환경을 만들 수 있도록 한다.
 - **Github Action**을 이용하여 이미지 자동 빌드 / 배포를 하여 빠른 배포를 가능하도록 한다.

Helm을 이용하여 서비스에 운영되는 모든 **Pod**들을 쉽고 빠르게 자동 배포 할 수 있도록 한다.

3.8.4 인프라 운영

- 본 서비스는 **AWS(Amazon Web Service)**에서 운영, 관리된다.
- 애플리케이션의 빠른 배포와 통합적인 관리를 위해 쿠버네티스를 사용한다.
- **AWS**에서 제공하는 쿠버네티스 통합 관리 환경인 **EKS**를 사용하여 **AWS** 서비스와 유기적으로 연결될 수 있도록 한다.
- 데이터베이스는 자체 배포를 하되, 추후에 **AWS**에서 제공하는 데이터베이스인 **DynamoDB**를 사용하도록 한다. 또한, 경로 데이터는 **EFS**에 저장하여 모든 인스턴스가 특별한 데이터 로딩 과정 없이 공유할 수 있도록 한다.
- 가능한 서비스는 같은 **VPN** 안에 위치시켜 네트워크 지연 시간을 줄이고, 보안적으로 안전하게 운영하도록 한다.

3.9 Project Management

프로젝트 진행 상황 공유는 Linear를 이용한다. 세 분야 Data Preparation, Front-end, Back-end로 나누어 각 파트를 맡은 담당자가 이슈를 올리고 진행 상황을 공유할 수 있도록 한다. 이슈를 정의할 때는 아래 내용이 명확히 정의되어야 한다.

Simple task name in one sentence clearly here

- Why you have to do this task?
 -
- What you have to do?
 -
- What is goal of this task?
 -

In Progress...

-

After finishing

- What you have learned
 -
- Final output
 -
- Opened related issues
 -

Backlog

Priority

Assignee

Project

Cycle

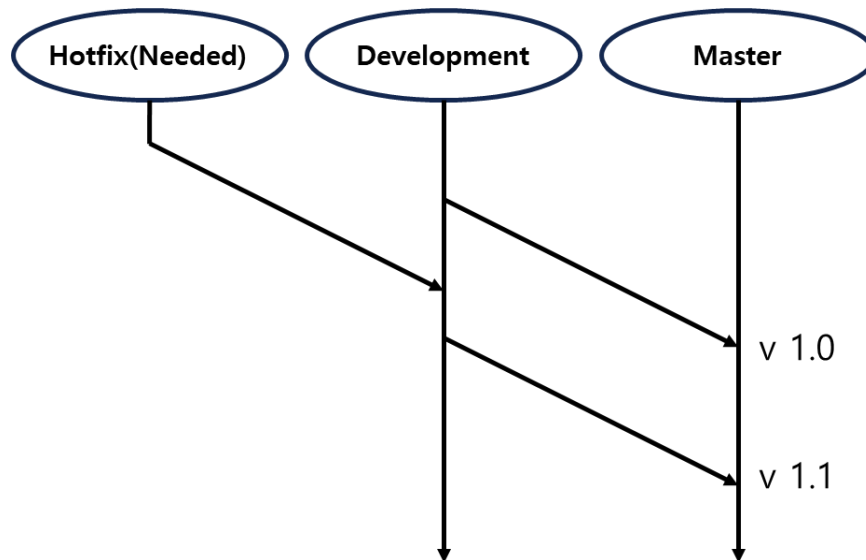
Create more

Create issue

[그림 10] 프로젝트 진행 상황 공유 예시

- 진행 전
 - 왜 이 태스크를 해야 하는가?
 - 해야 할 내용이 무엇인가? / 달성하고자 하는 목표가 무엇인가?
- 진행 중
 - 진행 중 내용 공유
- 진행 후
 - 무엇을 배웠는가?
 - 산출물
 - 이 이슈로 인해 열린 다른 이슈

위 과정을 거쳐 불필요한 태스크에 시간을 쓰지 않도록 조절하여 개발 생산성을 높일 수 있다. 일주일에 두 번, 간단한 회의를 진행하여 프로젝트 총 관리자는 전체 진행 상황을 파악하고, 다른 분야와 협업을 해야 하는 소요를 파악하도록 한다.



[그림 11] 버전, 코드 관리 관련 브랜치

버전 관리, 코드 관리는 **GitHub**을 이용한다. 테스트 중 개선해야 할 문제가 있으면 **Linear**와 **Github Issue**로 등록하여 해결할 수 있도록 한다. 두 개의 브랜치 (**Development**, **Master**) 브랜치로 나누어 진행하도록 한다.

- 작은 버전은 **Development**로 커밋하고, 점차 큰 버전을 만들어나간다.
- 하나의 완결된 버전은 **Master**에 배포한다.
- 이외에 다른 이슈를 처리해야 하면 **Hotfix** 브랜치를 만들어 별도 해결하도록 한다.
- 과정 중에 충돌이 없도록 해야 한다.

3.10 System Evolution

3.10.1 Limitation and Assumption

본 시스템은 강의실 찾기를 중점 기능으로 개발되었으나, 기타 편의시설(휴게실, 스터디 룸, 화장실, 식당 등)에 대해서는 길찾기를 제공하고 있지 않다. 실제로 강의실 뿐만 아니라 기타 시설의 위치를 모르는 학생이 많아 어려움을 겪는 일이 많기 때문에 강의실만을 대상으로 하는 것은 기능이 부족하다고 느낄 가능성이 있다. 또한, 이 서비스의

사용자는 기본적으로 길찾기를 어려워하는 신입생을 대상으로 하기 때문에 사용자 풀이 좁을 것이라는 한계도 있다.

3.10.2 Anticipated Changes

첫 번째로 강의실 뿐만 아니라 기타 편의 시설에 대해서도 길찾기를 제공하면 이에 어려움을 겪고 있는 학생들의 니즈를 만족시킬 수 있을 것이라 생각한다. 하지만 여러 시설에 대한 길찾기를 제공할 때 쉽게 조작할 수 있는 사용자 인터페이스는 추가적인 고안과 개발이 필요할 것으로 예측된다. 다음으로 다양한 기능을 제공하여 성균관대학교 신입생에서 성균관대학교 학생으로 사용자 풀을 넓힐 필요가 있다. 지도 기반의 운동장 예약 시스템, 식당 선주문 시스템 등 여러 편의 기능과 결합된다면 많은 사용자가 유용하게 사용할 수 있을 것이라고 예측한다.

4. Supporting Information

4.1. Software Requirement Specification

이 소프트웨어 요구 사항(SRS) 규격은 IEEE 권장 사항(IEEE 권장 소프트웨어 요구 사항 규격, IEEE-Std-830)을 참조하여 작성되었다.