



INSIDE CAMPUS

Software Requirement Specification

2020.04.24

Team 13(Inside Campus)

Team Leader	황 석진
Team Member	김 규용
Team Member	김 승호
Team Member	신 승환
Team Member	천 주형
Team Member	한 지명

목차

1. 소개

- 1.1 문서의 목적
- 1.2 시스템의 목적
- 1.3 단어 정의
- 1.4 참고 문헌
- 1.5 개요

2. 전체 설명

- 2.1 제품 구조
 - 2.1.1. 시스템 인터페이스
 - 2.1.2. 유저 인터페이스
 - 2.1.3. 하드웨어 인터페이스
 - 2.1.4. 소프트웨어 인터페이스
 - 2.1.5. 커뮤니케이션 인터페이스
 - 2.1.6. 메모리 제약사항
 - 2.1.7. 운용
 - 2.1.7.1. 시스템관리자
 - 2.1.4.2. 유저
- 2.2 제품 기능
 - 2.2.1. 경로 탐색
 - 2.2.2. 경로 즐겨찾기
 - 2.2.3. 프로필

2.2.4. 로드맵

2.3 유저 특성

2.3.1. 시스템 관리자

2.3.2. 유저

2.4 제약

2.5 설계에서의 가정과 의존

3. 세부 요구 사항

3.1 외부 인터페이스 요구 사항

3.1.1. 유저 인터페이스

3.1.2. 하드웨어 인터페이스

3.1.3. 소프트웨어 인터페이스

3.1.4. 통신 인터페이스

3.2 기능적 요구 사항

3.2.1. 사용 예시

3.2.2. 사용 예시 다이어그램

3.3 성능 요구 사항

3.4 데이터베이스 요구 사항

3.5 디자인 제약

3.6 개발 환경

3.7 소프트웨어 시스템 특성

3.7.1. 제품 요구사항

3.7.1.1 사용성 요구사항

3.7.1.2 효율성 요구사항

- 3.7.1.3. 이식성 요구사항
 - 3.7.2. 조직 요구사항
 - 3.7.2.1. 구현 요구사항
 - 3.7.2.2. 운영 요구사항
 - 3.7.3. 외부 요구사항
 - 3.7.3.1. 안정성 요구사항
 - 3.7.3.2. 법적 요구사항
- 3.8 요구사항 상세
 - 3.8.1. Context model
 - 3.8.2. process model
 - 3.8.3. Interaction model
 - 3.8.4. Behavior model – 각 기능에 따른 Sequence model 들
 - 3.8.4.1. 경로 검색 기능
 - 3.8.4.2. 건물 내부 로드뷰 기능
- 3.9 System 구조
- 3.10 유지 보수
 - 3.10.1 제한 사항과 예상
 - 3.10.2 Hardware 유지 보수 측면과 User requirements 변경에 대한 측면

4. 참고 정보

- 4.1 요구 사항 명세서
- 4.2 문서 히스토리

표 목차

[표 0] 기술적 단어 정의

[표 1] 소프트웨어 인터페이스 - Flutter

[표 2] 소프트웨어 인터페이스 – Firebase Realtime Database

[표 3] 통신 인터페이스

[표 4] 사용 예시 1

[표 5] 사용 예시 2

[표 6] 사용 예시 3

[표 7] 사용 예시 4

[표 8] 사용 예시 5

[표 9] 사용 예시 6

[표 10] 사용 예시 7

[표 11] 문서 인터페이스

그림 목차

[그림 1] 앱 실행 화면

[그림 2] 메인 화면

[그림 3] 건물 선택 시 내부 로드맵 출력 사진

[그림 4] 검색 화면

[그림 5] 검색 완료 시 화면(건물 검색)

[그림 6] 검색 완료 시 화면(강의실 검색)

[그림 7] 경로 검색 화면

[그림 8] 경로 검색 완료 화면

[그림 9] 옵션 화면

[그림 10] 사용 예시 다이어그램

[그림 11] Context model

[그림 12] process model

[그림 13] Sequence model – Path guide function

[그림 14] Sequence model – Road view function

[그림 15] System architecture

1. 소개

1.1. 문서의 목적

위 문서는 inside campus(학교 소개를 위한 지도) 시스템의 요구 사항을 명세하기 위한 요구 사항 명세서이다. 이 시스템은 2021년 봄학기 성균관대학교 소프트웨어 공학 개론 13팀에 의하여 디자인, 구현되었다. 시스템에 필요한 요구 사항의 분석, 요약 그리고 시스템 디자인, 구현은 '목차' 항목에 적힌 순서대로 기술되어 있다.

위 문서에서 예상되는 주요 독자로는 시스템 설계, 구현, 요구 사항 작성을 위한 소프트웨어 공학 개론 13팀, 그리고 추가적으로 검수를 위한 소프트웨어 공학 개론 교수, 조교이다.

위 문서가 쓰인 목적은 소프트웨어 공학 설계 13팀 프로젝트인 사용자를 위한 학교 소개 및 경로 안내를 위한 모바일 앱의 요구사항 명세서이다. 코로나-19를 겪고 신입생은 학교를 나오지 못해 학교에 대해서 잘 알지 못하며, 재학생 또한 제한된 형태로 개방된 학교의 출입구를 찾는데 불편함을 겪고 있다. 그렇기 때문에 다른 지도앱과 달리 inside campus는 학교에 대해 잘 알지 못하는 신입생 및 외부인을 위해 건물 내부의 모습 또한 볼 수 있도록 사진을 제시한다. 또, 경로에 대해서도 단순히 어떤 건물로 가기 위한 빠른 경로를 제시하는 것이 아니라, 코로나-19로 인해 닫혀 있는 출입구, 혹은 시간이 늦어서 폐쇄한 출입구를 피해 출입 가능한 출입구를 경로로써 제시한다.

1.2. 시스템의 목적

Inside campus는 기존 지도에 없었던 코로나-19로 인한 불편함, 차후에도 이러한 전염병이 발병했을 시 생길거라 예상되는 불편함을 개선하기 위해 만든 지도 앱이다. 감염자의 동선, 소독구역, 제한된 출입구들을 고려한 최적의 경로, 혹은 사회적 거리두기로 인한 원격 학교 탐방 등을 가능케하는 것이 목표이다. 시스템은 데이터베이스와 다익스트라 알고리즘을 사용을 기반으로 하고 있다. 이와 같이 전염병에 안전하고 효율적인 경로 제시로 인한 사용자들의 편안한 학교생활과 원격 학교 탐방으로 조금이나마 신입생의 불안을 해소하는 것을 목표로 삼고있다.

1.3. 단어 정의

위 문서에서는 지도 앱의 요구 사항을 명세하기 때문에 앱 내에서 사용하는 단어 들 보다 위 문서에서 기술적으로 사용하는 단어에 대해서 정의 한다.

[표 0] 기술적 단어 정의

단어	의미
유저	시스템을 사용할 사람들
디바이스	시스템의 구동하는데 필요한 도구
데이터 베이스(DB)	시스템의 기능 구현을 위한 모든 데이터(정보)를 모아 두는 곳
Path calculating engine (알고리즘)	최적의 경로를 찾아내는 방법을 제시하기 위한 연산을 하는 알고리즘
API	Application programming interface, 어플리케이션 소프트웨어의 개발 및 통합에 용되는 정의 및 프로토콜 세트

1.4. 참고 문헌

- IEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>
- Team 1. "Software Requirement Specification". SKKU, Last Modified : May. 13, 2020
2020spring_41class_team1/SRS_TEAM1.pdf at master · skkuse/2020spring_41class_team1 (github.com)
- Team 13. "Team 13 - Proposal". SKKU,
https://github.com/skkuse/2021spring_41class_team13/blob/master/Team%2013%20-%20Proposal.pdf
- 15조, 요구 사항 명세서, 2020, Konkuk
[\[T15\]SRS.pdf \(konkuk.ac.kr\)](#)

1.5. 개요

목차 항목에 의하면 이후에 소개될 내용은 크게 3챕터로 이루어져 있다. 두번째 챕터에서는 몇가지 인터페이스, 다른 시스템과의 상호 작용, 시스템의 기능을 포함한 제품의 전체적인 특성이 소개 될 것이다. 추가적으로 두번째 챕터에서 제품을 제작하는데 걸린 제약과 제품 설계에 있어서의 가정 또한 소개 될 것이다. 세번째 챕터에서는 더 세부적으로 들어가, 시스템의 특성과 함께 기능적 요구사항이 명세화 될 것이다. 또 그 다음으로 시스템 성능과 같은 비기능적 요구사항들이나, 기능적 요구사항들에 의해 생겨진 비기능적 요구사항, 디자인 제약 등에 대해서 소개할 것이다. 네번째 챕터에서는 프로젝트의 진행 상황, 문서 작성에서의 정보 등이 담겨 있다.

2. 전체 설명

2.1. 제품 구조

이 어플리케이션은 성균관대학교의 지리에 익숙하지 않은 사람들에게 캠퍼스, 건물 내부 지리 안내를 위해 디자인되었다. 처음 가보는 건물일지라도 경로 탐색을 통해 최적 경로를 안내 받을 수 있다. 코로나 바이러스로 인해 통제된 출입구를 제외한 경로를 안내하여 캠퍼스 내에서 보다 편리하게 사용할 수 있다.

2.1.1. 시스템 인터페이스

유저의 경로 탐색 결과, 히스토리, 즐겨찾는 경로는 SQLite 를 사용하여 로컬에 저장된다. SQLite 에 저장된 정보들은 API 를 통해 불러올 수 있다.

지도와 로드맵에 대한 데이터는 JSON 형식으로 firebase 에 저장되며 API 를 통해 불러올 수 있다.

2.1.2. 유저 인터페이스

인터페이스는 스마트폰의 화면에 표시된다. 유저의 경로 검색은 스마트폰의 텍스트 입력을 통해 이루어진다. 혹은 지도 화면의 터치를 통한 출발지, 목적지 설정을 통해 이루어질 수 있다. 경로 탐색 옵션을 인풋으로 전달할 수 있다. 탐색 결과는 건물 내부 로드뷰 또는 지도 경로로 유저에게 전달된다.

시스템 관리자는 지도와 로드맵 데이터베이스에 접속할 수 있다. 관리자는 firebase 의 GUI 를 사용하여 상대적으로 쉽게 정보를 관리할 수 있다.

2.1.3. 하드웨어 인터페이스

이 시스템은 스마트 디바이스에 최적화되어 있다. 디바이스는 최소 1GB 의 RAM 과 1GHz 싱글 코어를 포함해야 한다.

2.1.4. 소프트웨어 인터페이스

안드로이드 운영체제를 사용하는 디바이스의 경우 Android 6.0(API 23) 이상 버전에 최적화되어 있다.

IOS 를 사용하는 디바이스의 경우 iOS 10 버전 이상에 최적화되어 있다.

2.1.5. 커뮤니케이션 인터페이스

유저 디바이스와 서버는 API 를 통해 JSON 형식으로 정보를 주고 받는다.

2.1.6. 메모리 제약사항

이 시스템의 원활한 구동을 위해 디바이스는 1GB 의 RAM 과, 실행과 설치를 위해 적어도 512MB 의 저장 공간이 요구된다.

2.1.7. 운용

2.1.7.1. 시스템 관리자

- 지도 정보 변경

새로 추가되는 길, 건물과 없어지는 길, 건물 등에 대한 정보를 변경할 수 있다.

2.1.7.2. 유저

- 현재 위치

유저의 현재 위치를 표시한다.

- 경로 탐색

목적지, 출발지를 강의실 번호로 검색할 수 있다.

출발지, 목적지 설정을 통해 경로를 탐색할 수 있다.

- 옵션

사운드를 조절, 온/오프할 수 있다.

해당 강의실의 시작 예정 강의의 표시 여부를 선택할 수 있다.

경로 탐색 기준을 선택할 수 있다.

- 경로 즐겨찾기

경로 탐색을 통해 탐색된 경로를 자주 찾는 경로로 저장할 수 있다.

- 로드맵

건물 내부 로드맵을 확인할 수 있다.

- 프로필

검색 히스토리를 확인할 수 있다.

자주 찾는 경로를 저장, 삭제할 수 있다.

2.2. 제품 기능

2.2.1. 경로 탐색

경로 탐색을 위해서는 기본적으로 출발지, 목적지를 설정해야 한다. 출발지 목적지 설정은 크게 2 가지 방법이 있다. 첫 번째 방법은 아래와 같다. 지도의 한 부분의 짧은 터치를 통해 팝업이 뜨게 된다. 이 팝업은 세부 주소 정보를 포함하고, 출발지, 목적지, 취소 3 가지 중 선택할 수 있도록 한다. 기숙사를 제외한 교내 건물을 선택한다면, 건물 내부에 존재하는 룸 번호를 선택할 수 있도록 한다. 룸 번호를 선택하지 않고 출발지 혹은 목적지로 설정한다면 건물의 정문 입구를 기본으로 한다. 두 번째 방법은 건물의 룸 번호 검색이다. 검색 창에 룸 번호 검색을 하면 그 위치를 출발지 혹은 목적지로 설정할 수 있다. 출발지와

목적지를 설정하였다면 옵션을 선택하여 경로를 탐색한다. 옵션에는 최단 경로, 엘리베이터 포함, 도보(계단) 등이 있다. 선택하지 않는다면 엘리베이터를 포함한 최단 경로를 검색한다. 코로나로 인해 통제된 출입구를 포함한 경로는 최종 경로에서 제외된다.

2.2.2. 경로 즐겨찾기

경로 탐색을 완료한 경우 이 정보를 자주 찾는 경로로 저장할 수 있다. 또는 프로필의 자주 찾는 경로 메뉴 선택을 통해 설정 화면으로 진입할 수 있다. 이 메뉴에서 출발지, 목적지, 옵션 설정을 통해 경로를 최대 10 개까지 등록할 수 있다.

2.2.3. 프로필

검색 히스토리를 확인, 삭제할 수 있다. 자주 찾는 경로를 등록, 삭제할 수 있다. 자주 찾는 경로를 등록하기 위해 목적지와 도착지를 설정하여야 하며, 옵션으로 최적 경로, 최단 경로, 계단/엘리베이터 포함 경로를 옵션으로 선택할 수 있다.

2.2.4. 로드맵

유저는 건물 내부 로드맵을 볼 수 있다. 로드맵은 화면에 표시되는 화살표를 터치하여 원하는 방향으로 이동할 수 있다. 만약 로드맵 상의 가까운 거리에 강의실이 있다면 강의실의 번호를 입구의 앞에 표시하여 준다.

2.3. 유저 특성

2.3.1. 시스템 관리자

시스템 관리자는 이 시스템에 대한 이해를 하고 있는 자에 한정된다. 또한 시스템 관리자는 시스템에 발생하는 문제를 해결할 수 있는 역량을 가졌다고 가정한다. 관리자는 컴퓨터 공학을 전공한 사람 혹은 이와 비슷한 분야에서 공부를 한 사람이라고 가정하며, 이 시스템을 다룰 수 있는 기술을 갖고 있어야 한다.

2.3.2. 유저

이 문서에서 표현하는 유저란 일반적으로 이 시스템을 사용하는 고객이다. 유저는 캠퍼스, 건물 내부의 경로가 궁금한 사람들이다. 유저는 목적지에 대한 특정 정보를 갖고 있다고 가정한다. 예를 들면 건물 번호, 룸 번호 등이 있다. 또한 스마트폰의 사용, 어플의 설치, 사용에 큰 제약이 없는 사람이라고 가정한다. 주요 사용층으로는 신입생, 복학생, 캠퍼스 투어를 온 외부인이다.

2.4. 제약

이 시스템은 이 문서에 언급된 콘텐츠를 바탕으로 디자인, 구현된다. 세부적인 디자인, 구현은 개발자에 의해 정해진다. 다음의 사항들은 세부적인 디자인, 구현에서의 제약 사항들이다.

- 경로 탐색의 경우 5 초 이상 걸리면 안된다.
- 비용(저작권 등)을 지불해야하는 소프트웨어는 가능하다면 사용을 피한다.

- 가능하다면 오픈소스 소프트웨어를 사용한다.
- 시스템 비용과 유지 비용을 고려한다.
- 미래의 기술 발전을 고려하여 확장성을 높인 시스템을 개발한다.
- 개발 환경 OS 는 윈도우 10 이다.
- 개발 툴은 안드로이드 스튜디오 4.1.3 버전과 Xcode 12.5 버전이다.
- 구동을 위한 최소 안드로이드 버전은 Android version 6.0 (API 23), iOS 버전은 iOS 10 이다.
- 테스트를 위한 에뮬레이터 버전은 Android version 10 (API 29)이다.

2.5. 설계에서의 가정과 의존

이 문서는 모든 시스템은 안드로이드, iOS 디바이스와 오픈 소스를 사용하여 디자인, 구현된다는 가정과 함께 작성되었다. 안드로이드와 iOS를 기반으로 한 디바이스에 중점을 두고 작성되었기 때문에 이 외의 다른 운영 체제에서는 정상적으로 작동하지 않을 수 있다.

3. 세부 요구 사항

3.1 외부 인터페이스 요구사항

3.1.1 유저 인터페이스

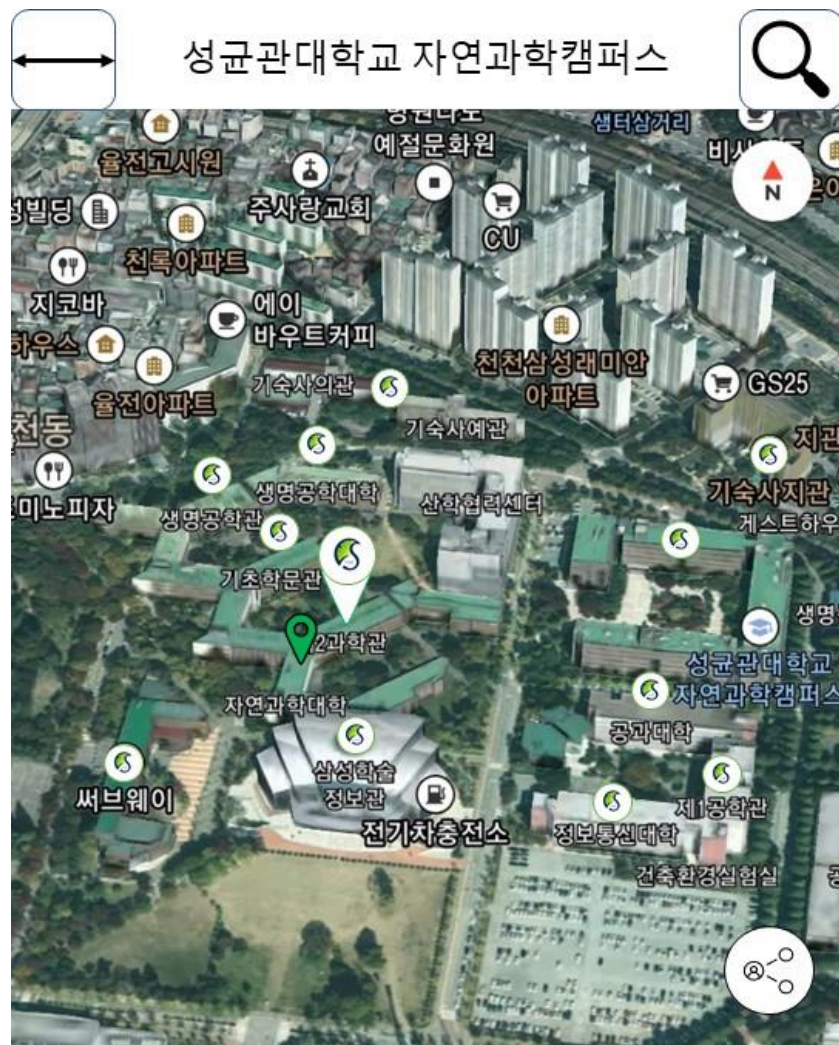
[그림 1] 앱 실행 화면

INSIDE CAMPUS



성균관대학교
SUNG KYUN KWAN UNIVERSITY

[그림 2] 메인 화면



☆ 제 2과학관 – 건물번호 : 32

열린 출입문 : 정문

1층 : 32101~32146

2층 : 32201~32246

3층 : 32301~32346

자연과학대학행정실, 자연과학대학학생회실, 생명과학학생회실


[그림 3] 건물 선택 시 내부 로드맵 출력 사진




☆ 제 1과학관 - 건물번호 : 31

열린 출입문 : 없음
(32동 주출입문 이용)

[그림 4] 검색 화면



성균관대학교 자연과학캠퍼스



검색어를 입력하십시오.

(검색어 입력 전) 즐겨찾는 위치 표시
(검색어 입력 시작 시) 추천 및 유사 검색어 표시

이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X

[그림 5] 검색 완료 시 화면(건물 검색)



[그림 6] 검색 완료 시 화면(강의실 검색)



[그림 7] 경로 검색 화면



성균관대학교 자연과학캠퍼스



출발지

검색어를 입력하십시오.

도착지

검색어를 입력하십시오.

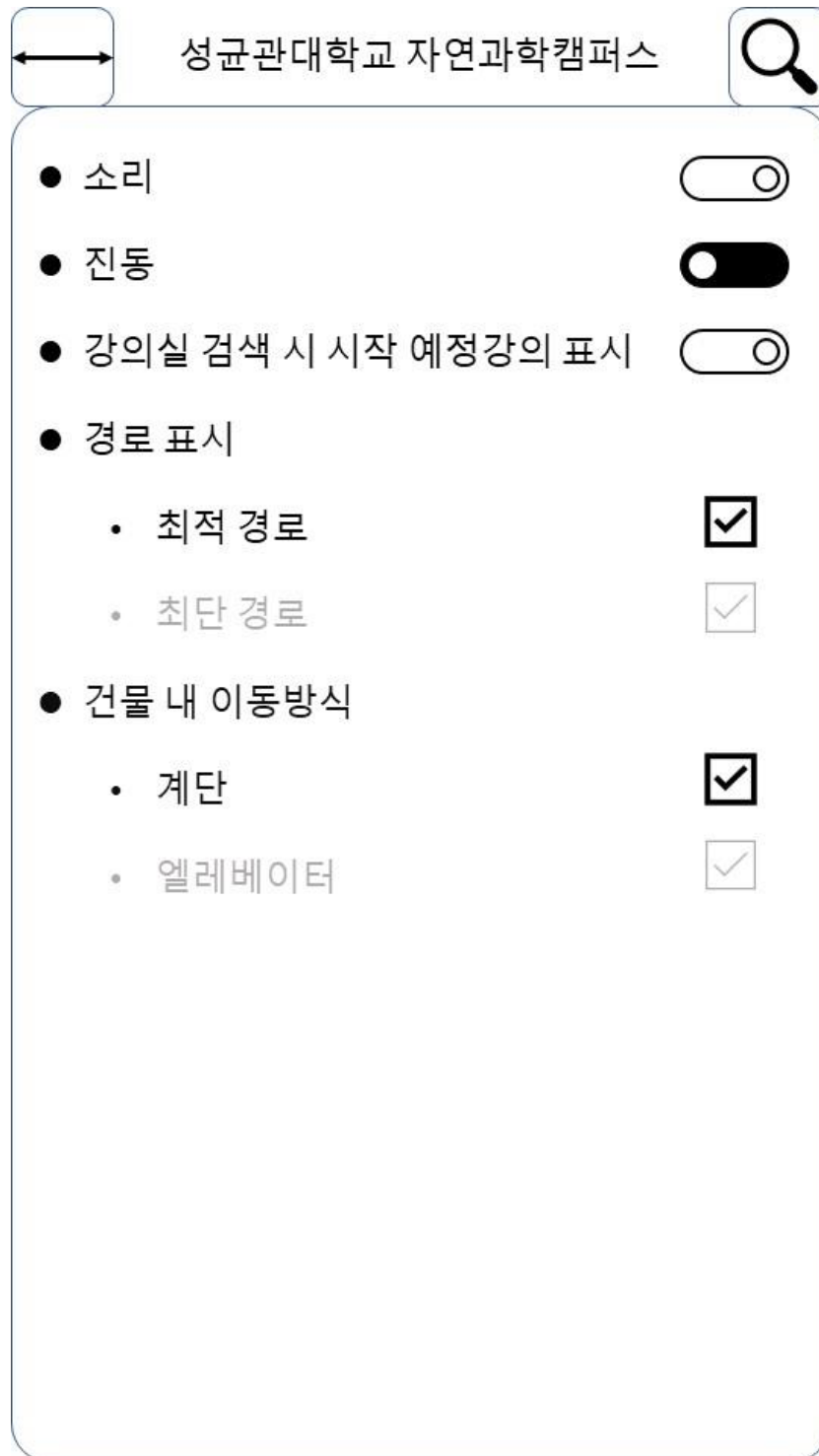
(검색어 입력 전) 즐겨찾는 위치 표시
 (검색어 입력 시작 시) 추천 및 유사 검색어 표시

이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X
이전 검색 기록들	X

[그림 8] 경로 검색 완료 화면



[그림 9] 옵션 화면



3.1.2. 하드웨어 인터페이스

Android OS(6.0 이상)이나 IOS(10 이상)을 채용한 디바이스가 요구된다.

3.1.3. 소프트웨어 인터페이스

[표 1] 소프트웨어 인터페이스 – Flutter

이 름	Flutter
목적/내용	앱의 전체적인 인터페이스 제작
입력 주체/ 출력 목적지	유저 혹은 시스템 관리자/디바이스의 화면
범위/정확도 /허용오차	Flutter의 성능에 따라 다름
단위	해당 없음
시간/속도	즉각적인 반응
타 입출력과 의 관계	어플리케이션의 모든 입출력과 관련
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음S
데이터 형식	다트(Dart)
명령 형식	다트(Dart) statement
종료 메시지	해당 없음

[표 2] 소프트웨어 인터페이스 – Firebase Realtime Database

	Firebase Realtime Database
목적/내용	멀티미디어 데이터 및 메타 데이터의 관리를 위한 쿼리 입출력
입력 주체/ 출력 목적지	호스트서버 또는 유저/유저 또는 호스트서버
범위/정확도 /허용오차	Firebase 의 성능에 따라 다름
단위	쿼리
시간/속도	즉각적인 반응
타 입출력과의 관계	호스트서버의 모든 입출력과 관련
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식	쿼리
명령 형식	쿼리 statement
종료 메시지	해당 없음

3.1.4. 통신 인터페이스

[표 3] 통신 인터페이스

이 름	클라이언트와 호스트 서버
목적/내용	각 클라이언트에서 호스트 서버에 접속을 요청하고, 검색 위치 요청 및 경로 탐색 요청, 현 위치 공유 요청 호스트 서버에서 각 클라이언트에 현 위치 제공 및 현 위치에서 검색된 위치로 가는 최적의 경로 제공, 출입이 가능한 문 정보 제공
입력 주체/ 출력 목적지	클라이언트와 호스트 서버
단위	패킷
시간/속도	최소 10Mbps 이상
타 입출력과 관계	서버와 관련된 모든 입출력
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식	쿼리
명령 형식	쿼리 statement
종료 메시지	해당 없음

3.2. 기능적 요구사항

3.2.1. 사용 예시

[표 4] 사용 예시 1

이 름	현 위치
설 명	사용자의 현 위치를 지도에 표시한다.
행위자	사용자
선행조건	GPS 센서가 켜져 있어야 하며, 네트워크에 연결되어 있어야 한다.
후행조건	해당 없음
기본 동작	<ol style="list-style-type: none"> 1. 현 캠퍼스 지도를 지도 API 혹은 호스트서버를 통해 출력 2. 사용자가 현 위치 버튼을 클릭 3. 사용자의 현 위치를 지도 위에 표시 4. 지도를 사용자의 현위치가 중앙에 오게끔 배치
추가 동작	해당 위치를 즐겨찾기로 지정을 원할 시 별 버튼을 누르면 저장

[표 5] 사용 예시 2

이 름	건물 내부 출력 및 내부 이동
설 명	건물의 실제 내부 사진을 로드뷰 방식으로 출력하고 건물 내부를 버튼을 통해 이동한다.
행위자	사용자

선행조건	GPS 센서가 켜져 있어야 하며, 네트워크에 연결되어 있어야 한다.
후행조건	해당 없음
기본 동작	<ol style="list-style-type: none"> 원하는 건물을 선택 후 층수 선택 주출입구(1 층) 혹은 해당 층 중앙 계단(1 층 제외) 사진 출력 및 실제 이동이 가능한 경로로 이동하기 위한 버튼 출력 (로드뷰) 이동 버튼 클릭시 사진 이동 및 이동 가능한 경로 재탐색 후 버튼 생성
추가 동작	해당 없음.

[표 6] 사용 예시 3

이 름	위치 검색
설 명	사용자가 특정 위치를 검색한다.
행위자	사용자
선행조건	GPS 센서가 켜져 있어야 하며, 네트워크에 연결되어 있어야 한다.
후행조건	검색 히스토리를 기기 DB 에 저장한다.
기본 동작	<ol style="list-style-type: none"> 검색버튼을 클릭 사용자가 검색하고 싶은 강의실번호나 이름을 입력 검색한 내용에 따라 출력 <ol style="list-style-type: none"> 1) 검색한 위치 데이터가 존재하면, 해당 위치를 출력하고 지도를 검색 위치가 중앙에 오게끔

	배치 2) 검색한 위치 데이터가 없으면, 재검색 요청 팝업을 출력하고, 현 화면을 유지 4. 지도를 사용자의 검색 위치가 중앙에 오게끔 배치
추가 동작	도착지 버튼 혹은 출발지 버튼 선택 시, 경로 탐색을 시행 해당 위치에 대해 즐겨찾기를 원할 시 별 버튼을 눌러 선택.

[표 7] 사용 예시 4

이름	위치 공유
설명	사용자의 현 위치 혹은 검색 위치를 공유한다.
행위자	사용자
선행조건	GPS 센서가 켜져 있어야 하며, 네트워크에 연결되어 있어야 한다.
후행조건	공유 방식을 선택하여야 한다.
기본 동작	1. 공유 버튼을 선택 시 현 위치를 출력하는 url 링크를 생성 2. 링크 복사, 카카오톡 등의 공유방식 선택창 팝업 3. 현 위치 공유 1) 링크 복사 선택 시, 링크 주소를 디바이스 클립보드에 저장 2) 카카오톡 등 어플 선택 시 해당 어플의

	공유방식으로 현 위치 공유
추가 동작	팝업창 내의 버튼 이외의 화면을 클릭시 해당동작 취소

[표 8] 사용 예시 5

이 름	경로 탐색
설 명	사용자가 지정한 출발지 및 도착지의 최적 경로를 출력한다.
행위자	사용자
선행조건	GPS 센서가 켜져 있어야 하며, 네트워크에 연결되어 있어야 한다.
후행조건	검색한 경로 히스토리를 기기 DB 에 저장한다.
기본 동작	<ol style="list-style-type: none"> 출발지와 도착지를 설정 <ol style="list-style-type: none"> 미리 검색한 위치를 출발지 혹은 도착지로 설정한 경우, 도착지 혹은 출발지만 검색 두 곳 전부 검색하는 경우 중 같은 곳을 설정한 경우, 후에 검색한 곳에 위치 설정하고 전에 설정한 위치 초기화 해당 두 지점을 직선거리(점선)과 실제 통행거리(실선)으로 표시 실제 통행거리로 예상 소요시간 설정 (축적과 사람의 평균도보 속도인 4km/h 로 측정)
추가 동작	해당 없음.

[표 9] 사용 예시 6

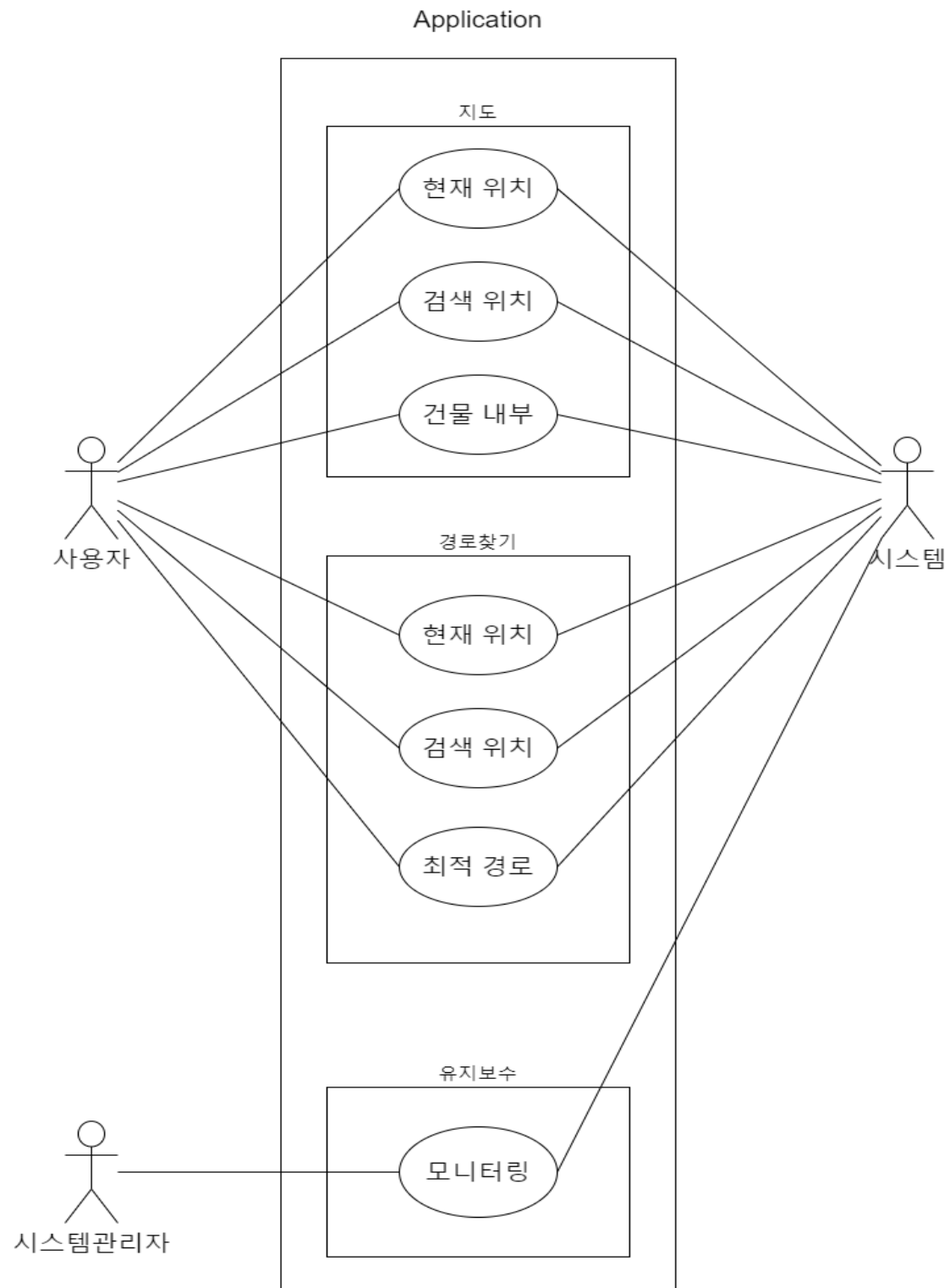
	옵션 설정
설 명	사용자가 앱의 옵션을 설정한다.
행위자	사용자
선행조건	상단 중앙을 누르면 옵션 설정창을 활성화한다.
후행조건	해당 없음
기본 동작	<ol style="list-style-type: none"> 1. 소리와 진동을 사용할지 안할지 설정 2. 검색한 강의실의 시작 예상 강의를 표시할지 설정 3. 검색 경로의 종류 설정 <ul style="list-style-type: none"> - 최단 경로 혹은 최적 경로 선택 시 다른 쪽의 버튼을 비활성화 - 둘 다 설정하지 않을 시 최단 경로로 기본 설정 4. 건물 내의 이동방식 설정 <ul style="list-style-type: none"> - 계단 / 엘리베이터 중 선택 시 다른 쪽의 버튼을 비활성화 - 둘 다 설정하지 않을 시 엘리베이터로 기본 설정 - 건물 내에 엘리베이터가 없는 경우 엘리베이터로 설정하더라도 계단으로 인식되도록 설정
추가 동작	해당 없음.

[표 10] 사용 예시 7

이 름	모니터링
설 명	관리자가 앱 오작동 및 오류사항을 확인한다.
행위자	관리자
선행조건	해당 없음
후행조건	해당 없음
기본 동작	기본적인 앱 동작이나 지도 데이터의 오류, 지도에 표시되는 데이터들의 오류 등을 확인하고 유지 및 보수가 필요한 곳은 수정합니다.
추가 동작	해당 없음.

3.2.2. 사용 예시 다이어그램

[그림 10] 사용 예시 다이어그램



3.3 성능 요구사항

이 소프트웨어는 약 100 명의 동시 접속자를 생각하고 있기 때문에 100 명 이상의 클라이언트에 안정적인 QoS 를 제공하여야 한다.

모든 요청과 명령은 2 초 이내로 응답해야 하며, 검색을 진행하면 3 초 이내에 검색 결과값을 출력해야 된다. 경로 탐색의 경우, 5 초 이내로 최적 경로와 예상 소요시간, 예상 도착시간을 출력해야 한다.

각 클라이언트 내부에서의 각 사용자 명령에 대해 1 초 이내에 응답하여야 한다.

3.4 데이터베이스 요구 사항

Firebase에 건물 및 방의 위치 정보는 물론, 이용 가능한 경로 및 입출구 정보를 저장한다. 또한 정확한 로드뷰 제공을 위해, 각 로드뷰 위치의 최신 사진을 서버에 저장해야 한다. 저장한 정보는 추후 최단경로를 구해주는 알고리즘 및 머신러닝 모델을 구현하는데 쓰인다. 데이터베이스는 항상 무결성 제약조건을 준수해야 하며, 정규화를 통해 중복을 제거하고 처리 속도를 향상시켜야 한다.

3.5 디자인 제약

시스템은 MIT license에 명시된 의무사항을 준수하여 만들어져야 한다. 시스템은 다양한 모바일 기기 및 운영체제에서 접속 및 사용이 가능해야 한다. 데이터베이스 및 백엔드 처리를 위해, 시스템은 파이어베이스를 이용할 수 있도록 디자인이 되어야 한다. 사용자가 로드뷰 및 경로를 쉽게 이해할 수 있도록 디자인이 되어야 한다.

3.6 개발 환경

Photoshop을 이용하여 지도의 로고와 아이콘을 디자인하며, XD를 이용하여 직관

적인 UI/UX를 디자인한다. Flutter와 Github을 이용하여 여러 개발자들과 협업하여 크로스 플랫폼 앱개발을 진행한다. Firebase로 데이터베이스 및 백엔드 처리를 한다.

3.7 소프트웨어 시스템 특성

소프트웨어 시스템 특성은 주로 비기능적 요구사항을 통해서 드러나게 된다. 따라서 이 문단에서는 시스템의 비기능적 요구사항을 설명하고자 한다. 비기능적 요구사항에는 크게 제품 요구사항, 조직 요구사항, 외부 요구사항이 존재한다. 하나씩 설명을 하면 다음과 같다.

3.7.1 제품 요구사항

제품 요구사항은 소프트웨어의 기능, 목적 등을 설명한 문서로, 소프트웨어가 어떻게 동작하는지를 명시한다. 구체적으로, 다음과 같은 요구사항을 만족해야 한다.

3.7.1.1 사용성 요구사항

사용성 요구사항은 사용자들이 소프트웨어를 쉽게 사용할 수 있어야 한다는 원칙을 지키기 위해 필요한 요구사항이다. 전문용어는 최소화하고, 필요할 경우 쉽게 설명을 해야 한다. 사용자는 복잡한 매뉴얼을 배우지 않고도 시스템의 기능을 전부 어렵지 않게 사용할 수 있어야 한다. 즉, 사용하기 쉽고 직관적인 UI를 만들어야 한다.

3.7.1.2 효율성 요구사항

연산 및 메모리 사용을 효율적으로 하는 것은 모든 소프트웨어에서 중요하다. 최단경로를 구해주는 알고리즘은 이 시스템에서 가장 시간 및 공간 복잡도가 높은 연산의 하나다. 또한 경로를 지도로 보여주므로, 효율적인 이미지 처리 기술이 요구된다. 사용성을 높이기 위해, 최단경로를 구해 지도로 보여주는 것까지 5초 안

에 이루어져야 한다.

3.7.1.3 신뢰성 요구사항

시스템은 유저로부터 요구 받은 기능을 정확하게 제공해야 한다. 최단경로를 추천할 때, 이용이 불가능한 경로를 추천하면 안 된다. 최단경로는 유저의 기대를 만족시켜야 하며, 에러 가능성이 1%를 넘으면 안 된다. 유저로부터 입력을 받았을 때, 결과물을 반드시 유저에게 보여줘야 한다.

3.7.1.4 이식성 요구사항

소프트웨어는 가능하면 다양한 플랫폼 위에서 똑같이 작동해야 한다. 안드로이드와 ios에서 시스템은 똑같이 동작해야 한다. 또한 GPS 및 와이파이 기능을 지원하는 디바이스라면, 기종에 무관하게 정상적으로 동작해야 한다.

3.7.2 조직 요구사항

조직 요구사항은 소비자 및 개발자가 속해있는 조직의 규정과 절차를 따르기 위한 요구사항이다.

3.7.2.1 구현 요구사항

개발 방법론으로 폭포수 모델(waterfall model)을 사용한다. Photoshop을 이용하여 지도의 로고와 아이콘을 디자인하며, XD를 이용하여 직관적인 UI/UX를 디자인한다. Flutter와 Github를 이용하여 여러 개발자들과 협업하여 크로스 플랫폼 앱개발을 진행한다. Firebase로 데이터베이스 및 백엔드 처리를 한다.

3.7.2.2 운영 요구사항

만들어진 시스템은 스마트폰 앱으로 언제든지 접속, 사용이 가능해야 한다. 사용자가 입력을 하고 5초 안에는 결과를 보여줘야 한다. 시스템은 유저에게 가장 적절한 최단경로를 보여줘야 한다.

3.7.3 외부 요구사항

외부 요구사항은 시스템 및 개발 과정의 외부에 대한 요구 사항을 말한다.

3.7.3.1 안전성 요구사항

모든 경로 및 위치 정보는 암호화 및 백업을 통해 안전하게 보관되어야 한다. 유저의 개인정보는 유출되어서도, 타 목적으로 악용되어서도 안 된다. 시스템의 모든 중요 정보는 외부의 접근을 허용해서는 안 된다.

3.7.3.2 법적 요구사항

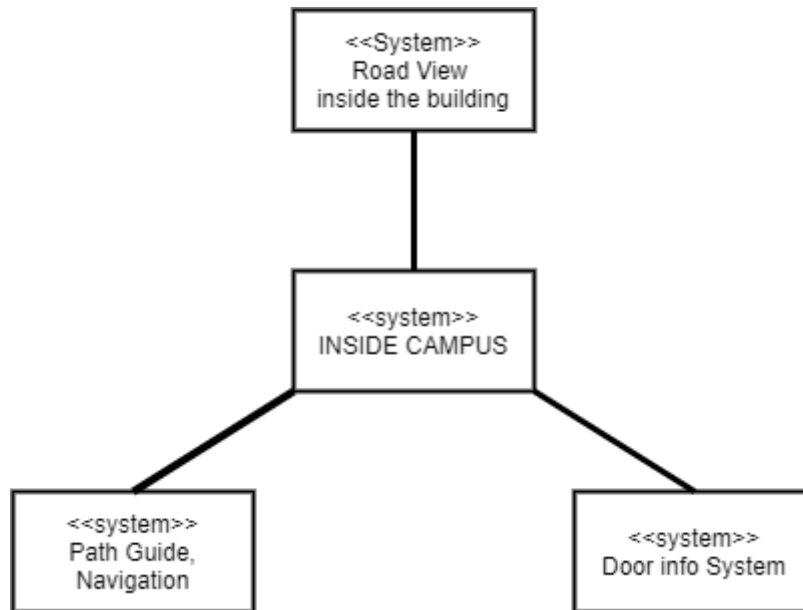
이 시스템은 어떤 범죄에도 악용되어서는 안 된다. 유저의 개인정보는 개인정보 보호법에 따라 처리해야 한다.

3.8 Organizing the Specific Requirements

이 목차에서는 Unified Modeling Language 와 tabular form 을 이용하여 system model 을 설명한다. 각 단계에서 system 의 abstract model 들을 확인할 수 있으며, 각 단계마다 다른 관점에서 접근하여 표현한다.

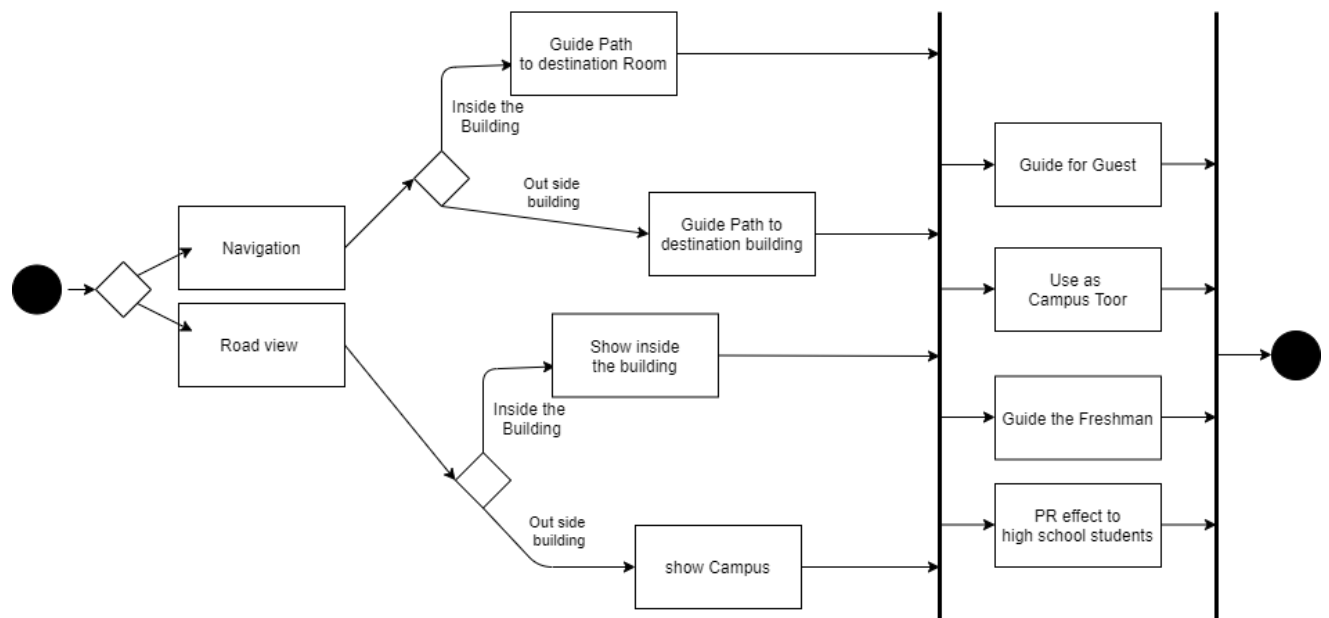
3.8.1 Context model.

[그림 11] Context model



3.8.2 process model

[그림 12] process model



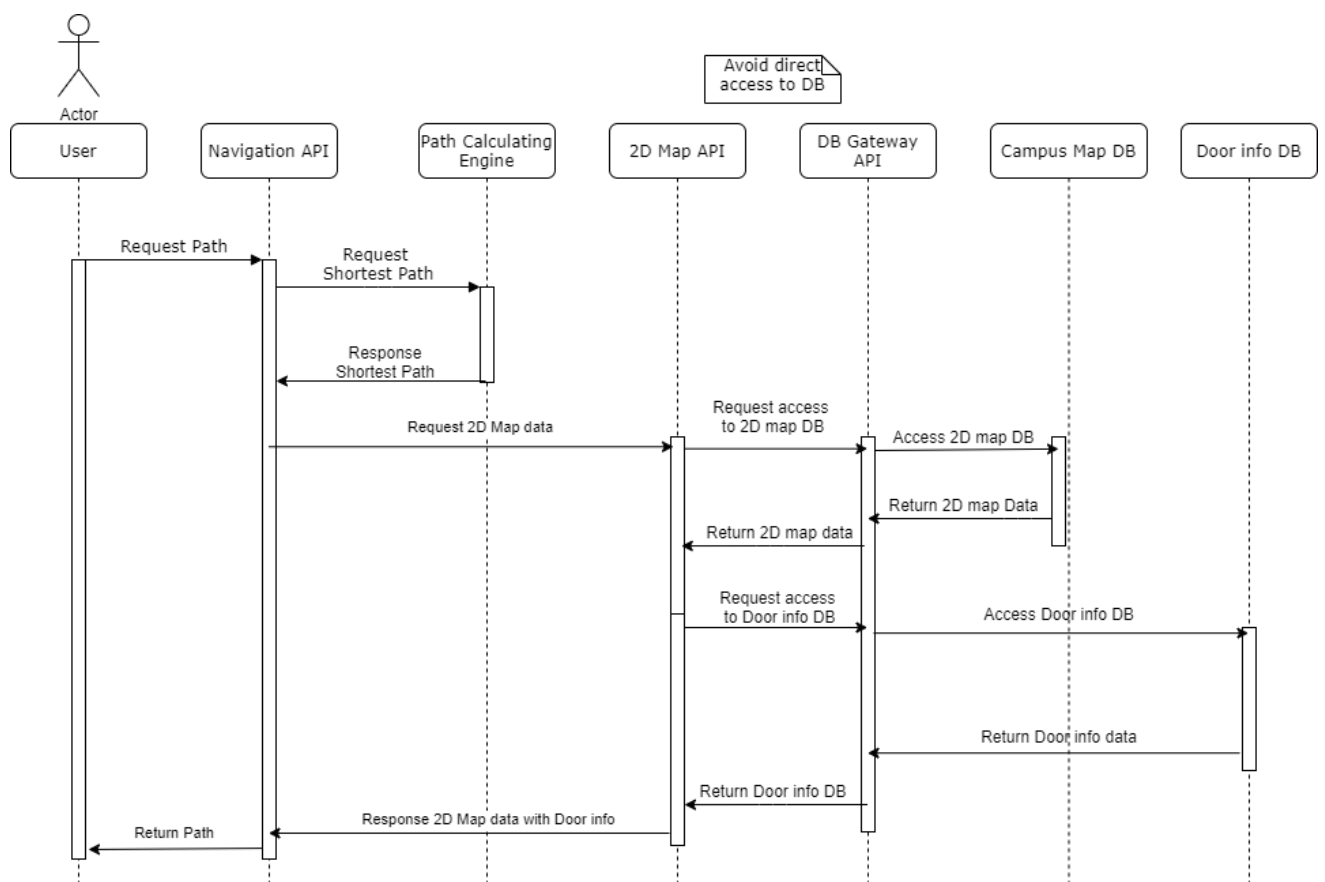
3.8.3 Interaction Model

3.2.2. 사용 예시 다이어그램 참고

3.8.4 Behavior model – 각 기능에 따른 Sequence model 들

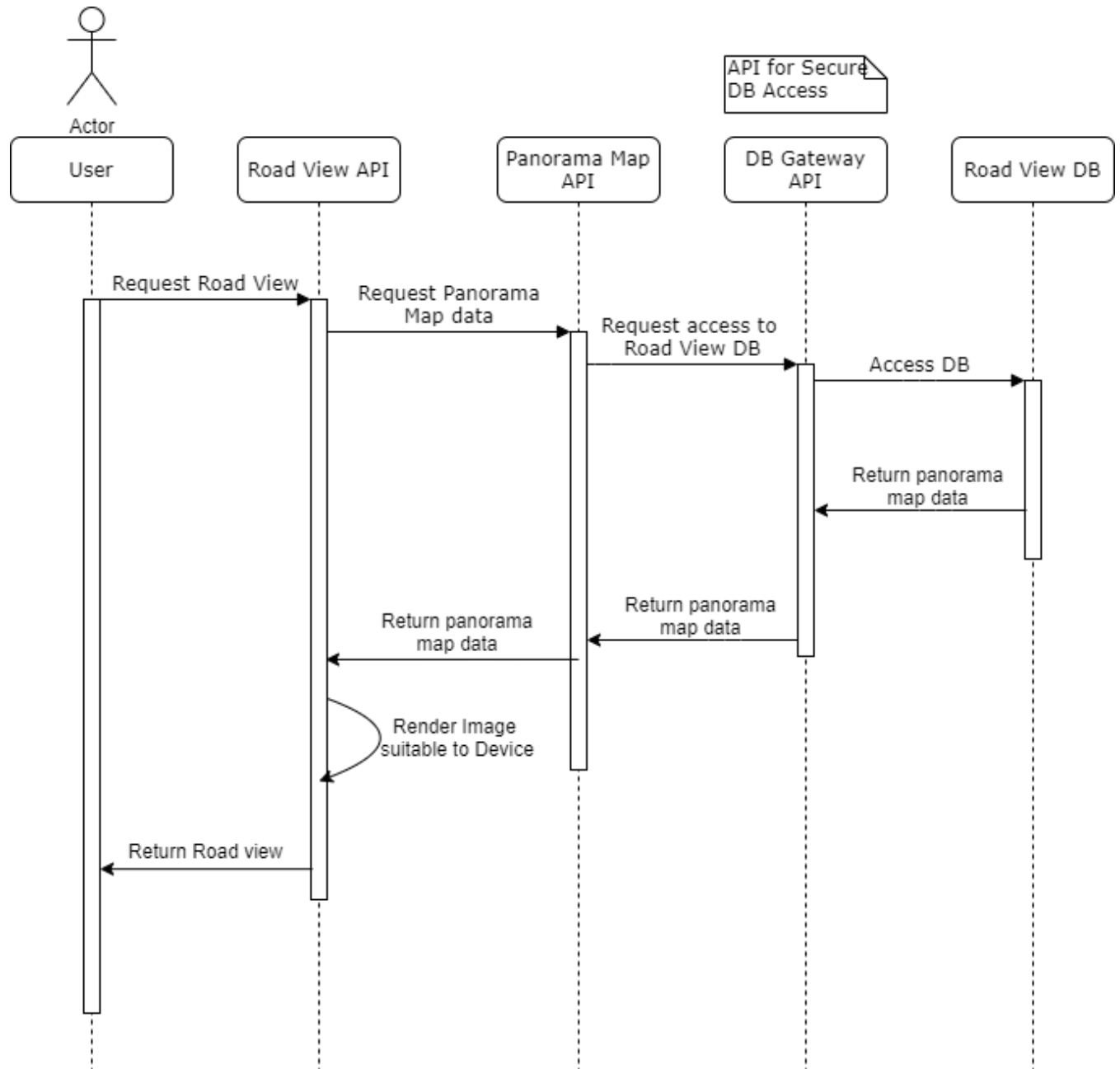
3.8.4.1. 경로 검색 기능

[그림 13] Sequence model – Path guide function



3.8.4.2 건물 내부 로드뷰 기능

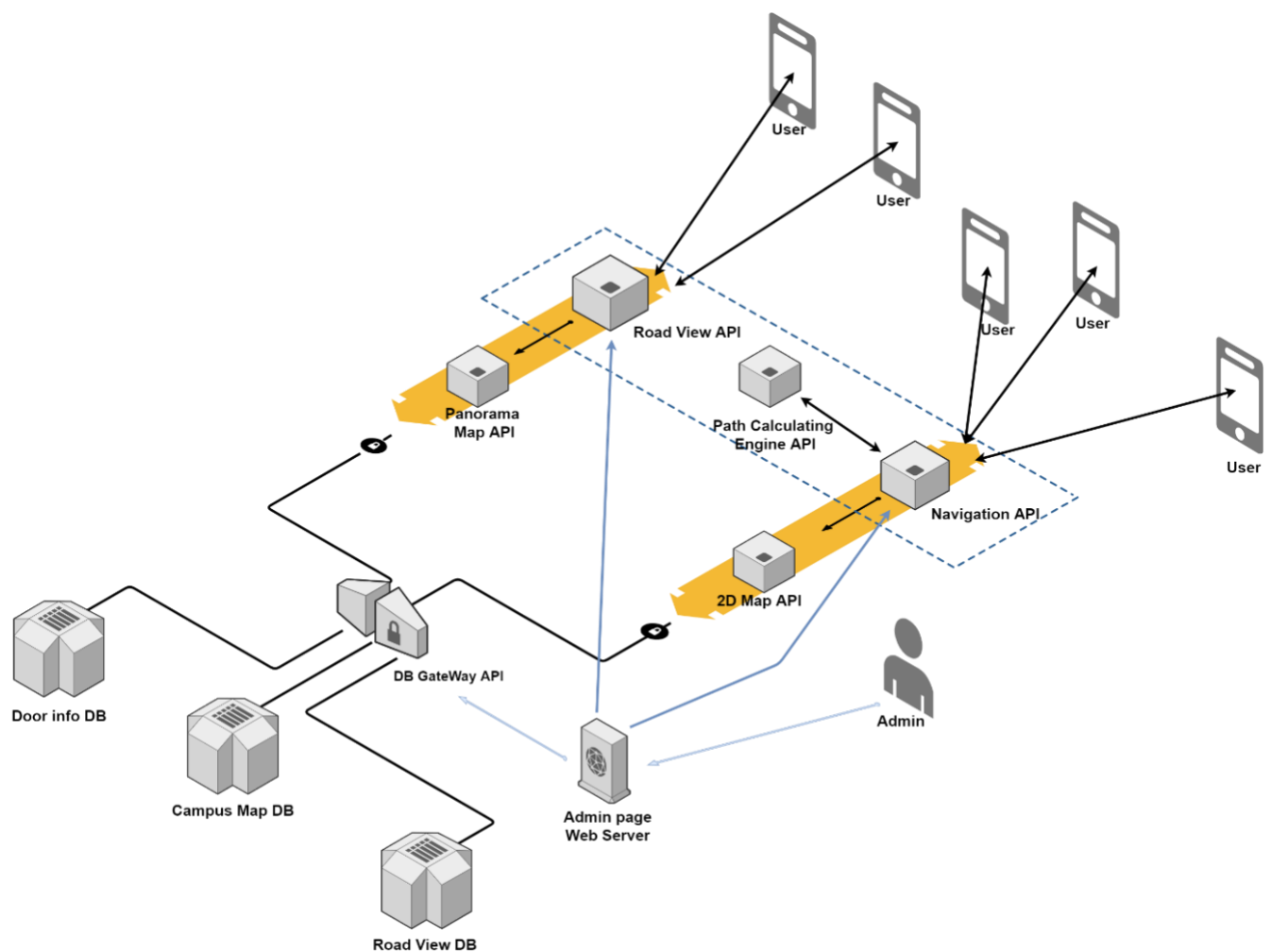
[그림 14] Sequence Model – Road view function



3.9 System Architecture

다음은 전체적인 개발 계획에서의 시스템의 구조를 도식화로 표현한 것이다. 각 기능 별로 API 서버들이 분리되어 구성이 되며, 궁극적으로는 Cloud-Native 한 환경에서 운영하는 것을 목표로 한다. 또한 유지보수 측면에서 긍정적인 효과를 볼 수 있는 Micro Service Architecture 를 도입하여 각 API 서버들을 Docker 로 유지한다.

[그림 15] System architecture



3.10 System Evolution

이 목차에서는 시스템이 기반이 되는 환경에 근본적인 추정들을 다룬다. 또한 하드웨어 유지보수나 요구사항의 변화로 인한 예상되는 수정 사항들도 다룬다. 이 목차는 미래의 유지보수를 책임지는 개발자와 디자이너에게 유용할 것이다.

3.10.1 Limitation and Assumption

현재 우리의 앱은 성균관대학교 하나의 학교에만 서비스를 하게 된다. 따라서 다른 학교로도 추가적으로 제공할 가능성도 존재한다.

3.10.2 Evolutions of Hardware and Change of User Requirements

하드웨어가 바뀌거나, 교체가 필요한 경우에도 해당 교체 디바이스와 연결되는 API 또는 서버는 문제없이 작동할 것이다. Restful API 구조와 Micro service Architecture 로 구성을 하므로, Component 간의 Communication 에 있어서 형식만 지켜서 요청 & 응답을 하게 된다면 문제는 없을 것이다. 추가적인 User requirements 로 예상되는 것은 현재 기능은 학교 캠퍼스로 제한이 되어있지만, 학교 근처 맛집 거리, 카페, 등등 주변의 장소들로 가는 길안내와 로드뷰 기능도 요구될 수 있다. 따라서 우리 DB 구조를 설계할 때, 학교 건물만이 아닌, 다른 건물도 포함될 가능성을 염두에 두고 설계해야 하며, 학교 주변의 파노라마 사진들도 촬영해서 로드뷰 기능으로 사용할 준비를 해야 할 것이다.

4. 참고 정보

4.1 요구 사항 명세서

위 소프트웨어 요구 사항 명세서는 IEEE Recommendation (IEE Recommended Practice for Software Requirements Specifications, IEE-Std-830)의 기준을 참고하여 작성되었다.

4.2 문서 히스토리

[표 11] 문서 히스토리

일자	버전	설명	작성자
2021.04.10	0.1	디자인, 목차	신승환
2021.04.12	1.0	1.1, 1.2, 1.3, 1.4, 1.5 추가	신승환
2021.04.12	1.1	2.1, 2.2, 2.3, 2.4, 2.5 추가	김승호, 한지명
2021.04.12	1.2	3.1, 3.2, 3.3 추가	김규용
2021.04.14	1.3	3.4, 3.5, 3.6, 3.7 추가	천주형
2021.04.14	1.4	3.8, 3.9, 3.10 추가	황석진
2021.04.21	1.5	모든 파트 검토	김규용, 김승호, 신승환, 천주형, 한지명, 황석진
2021.04.22	1.5	4.1, 4.2 추가	신승환
2021.04.22	2.0	모든 항목 병합	신승환
2021.04.23	2.1	최종 검토, 표지 작성	황석진