

Capstone Design

Final report



교 수 님: 손 용 석 교수님

팀 명: 재건축 (team #7)

조 원: 이동훈 20216018

한동희 20210189

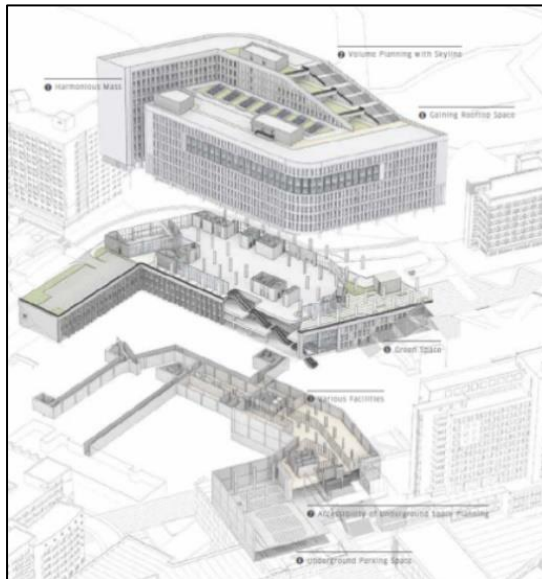
민보욱 20215744

제 출 일: 2023년 12월 9일

1. Introduction

○ 문제설정

- ❖ 큰 규모의 시설이나 건물의 경우 내부에서 길을 잃는 경우가 있음
- ❖ 실내 길 찾기를 지원하는 보편적인 서비스가 없음
- ❖ 일반 방문객들은 건물 내 안내도를 얻기도 쉽지 않다(지정된 위치에 적은 개수로만 존재)
- ❖ 사람들이 어느 건물에서도 쉽게 길을 찾을 수 있도록 하는 서비스를 플랫폼화 하는 것은 어떨까?



초행길에 길을 찾기 어려운
건물 중 하나인
중앙대학교 310관 전경

2. Motivation

현재 존재하는 실내 길 찾기 서비스는 장소가 매우 제한적이다. 그 이유는 AR 기반의 서비스이기 때문에 구현 난이도가 높기 때문이다. 이러한 문제를 해결하며 사람들이 더 쉽고 넓은 방향으로 사용할 수 있는 길 찾기 플랫폼을 만들어 보고자 한다.

단순한 길 찾기 앱이지만 화장실이 급할 때, 화재 등의 비상상황 혹은 더 나아가 시력이나 거동이 불편한 장애인들에게도 도움이 될 것이라고도 판단하였다. 이러한 앱들이 실용성이 있으려면 주변에 서비스를 지원하는 실내 공간이 충분해야 한다. 이를 보완하기 위해 내린 결론이 ‘플랫폼’화이다. 여력상 앱 제작 업체나 팀이 모든 공간의 데이터를 얻기는 힘들다. 그렇기에 사용자들에게 다른 사람들이 필요로 할 만한 복잡한 공간들에 대하여 데이터를 제공받는 시스템을 구축하는 것이다.

3. Related Work

○ AR 도보 네비



실내 측위 기술, AR 등 최신 기술이 집약된 길 안내 서비스

➤ 단점

- ❖ 실내 측위 기술, AR, 건물 스캔 기술은 돈 및 시간이 많이 필요한 작업
- ❖ 현재위치를 정확하게 알 수 있지만 지원 가능한 건물이 한정적이게 됨

➤ 개선 가능한 방안

- ❖ 현재 위치를 실시간으로 알 수 있는 기술을 제외
- ❖ 길 찾기를 위해 비교적 얻기 쉬운 데이터인 도면(안내도) 데이터 활용
- ❖ 데이터를 제공받고 가공 후 최대한 많은 건물을 서비스할 수 있는 플랫폼

4. Functional Description

○ Frontend

➤ Flutter

- ❖ Dart 언어 기반의 구글에서 개발한 멀티플랫폼 개발 프레임워크

➤ Fultter을 선택한 이유

- ❖ 개발 속도가 빠르다고 알려져 있고 개발자 경험에 있어서도 좋은 평가를 받음
- ❖ 멀티플랫폼 개발이 가능하므로 추후 다른 플랫폼에서의 배포 용이
- ❖ 핫 리로드 기능을 사용해 코드 수정 후 디버그 과정에서 시간 단축

➤ Flutter 내부 주요 기능

- ❖ Naver Map API를 사용하여 지원 가능 건물 표시, 새로운 건물정보 저장 가능(위도, 경도 기반)
- ❖ 현재 위치 정보 (GPS) 권한 요구 및 현 위치 기반 주변 건물 탐색 가능
- ❖ 자체 API 서버와 지속적인 통신을 통해 사용자에게 정보 제공 또는 업로드 지원



- **Backend (DB)**

➤ Firestore

- ❖ 구글에서 지원하는 NoSQL 데이터베이스 서비스
- ❖ 실시간 리스너를 통해 사용자와 기기간 실시간 데이터 동기화
- ❖ Flutter과의 좋은 연동성

➤ 사용 용도

- ❖ 각 건물의 위치정보, 층 정보 저장
- ❖ 어플리케이션에서 길 찾기 전, 방문자에게 실시간으로 건물정보, 각 층별 객체 정보 전달

[illegible]

- **Server**

➤ Flask API

- ❖ Flask를 사용하여 Rest API를 주고받는 서버 제작
- ❖ Image Segmentation, 길 찾기 알고리즘 모두 Python기반이므로 Flask로 API 제작

➤ **배포**

- ❖ AWS의 EC2 인스턴스를 생성하여 Ubuntu 환경에 배포

- ❖ 타 서비스와 통신 도중 보안 관련 제약사항이 많아 HTTPS 및 로드밸런스를 적용하여 커스텀 도메인에 배포 중
- ❖ 개발자의 개입 없이 모든 과정이 자동화될 수 있도록 설계

[illegible]

○ AI

➤ Image Segmentation

- ❖ 이미지에서 픽셀 단위로 관심 객체를 추출하는 방법

➤ 딥러닝 학습관련 정보

- ❖ 데이터셋: 건축 도면 데이터(아파트) 약 2만장
- ❖ Deeplab V3 모델 사용

➤ 학습 결과 모델 사용 image segmentation 결과



- ❖ 높은 정확도가 필요한 서비스인 만큼 검수가 필요할 것은 불가피
- ❖ Image segmentation 결과 보정 작업이 필요(수작업)
- ❖ AI의 역할은 사람의 작업 시간을 줄여주는 것에 한함

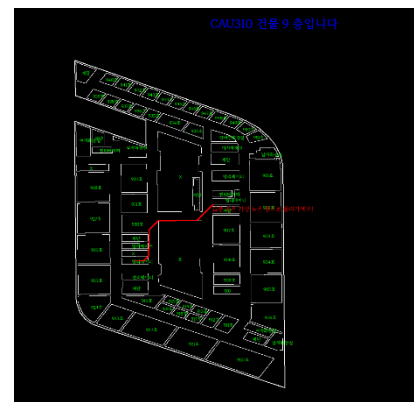
○ Main Function

➤ 길찾기 알고리즘

- ❖ Image Segmentation 결과를 기반으로 길 찾기 알고리즘 제작
- ❖ BFS 기반 탐색을 통한 경로 설정

➤ 주요 고려사항

- ❖ 다른 층으로 이동할 경우 이동수단 객체를 탐색
- ❖ 각 객체 사이의 작은 빈 틈을 벽으로 인식하도록 설계
- ❖ 위, 아래층으로 이동하다가 더 이상 올라가지 못할 때 그 층에서 다른 이동수단 객체를 찾도록 설계



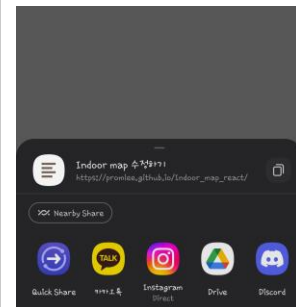
○ Sub Function

➤ 객체 정보 채우는 웹사이트

- ❖ 기본적으로 비워져 있는 각 객체의 이름을 지정해줄 수 있도록 하는 웹사이트

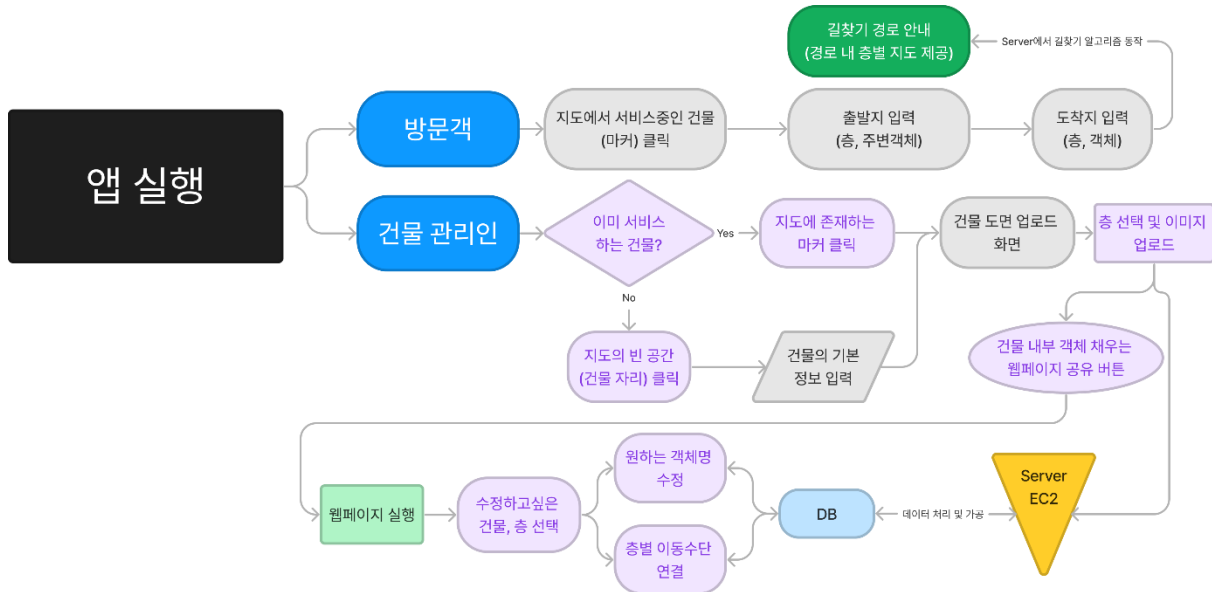
➤ 웹사이트 제작 이유 및 기능

- ❖ 본 서비스의 단점인 객체의 이름을 인식할 수 없다는 점을 보완하기 위해 제작
- ❖ 큰 사진과 가독성을 위해 큰 화면에서도 객체정보를 채울 수 있도록 웹페이지 기반 제작
- ❖ 모바일에서도 브라우저를 통해 누구나 접속 가능
- ❖ 실시간으로 mask 파일을 바꿔주어 현재 상태 확인 가능
- ❖ 도면 이미지 업로드 화면에서 링크 공유 가능

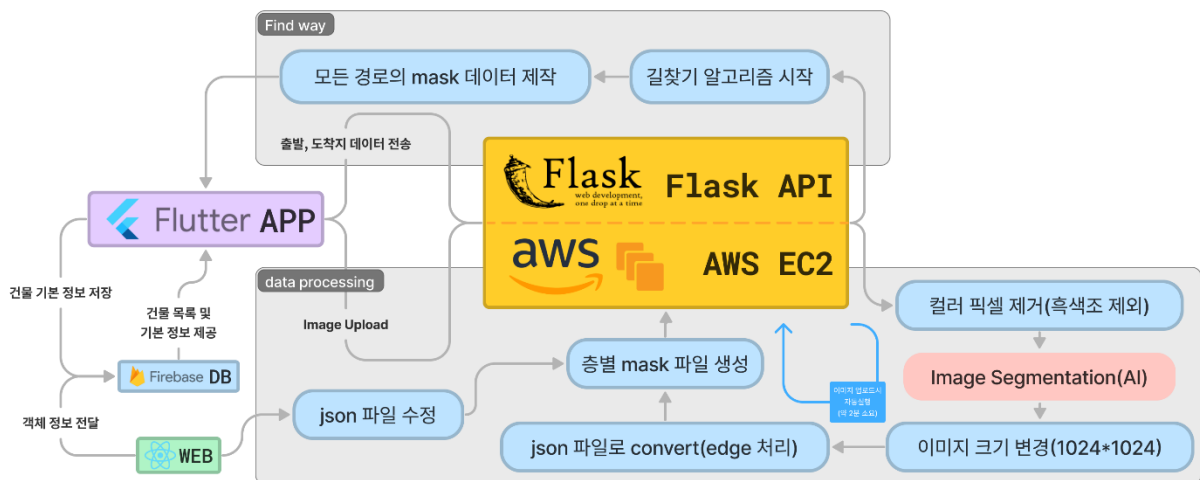


5. Flowchart

○ Workflow – Frontend(User)



○ Workflow – Backend(Server)



6. Conclusion

○ Member Role and Schedule

➤ 한동희

- ❖ 팀장
- ❖ 길찾기 알고리즘 구현
- ❖ 시연 시나리오 작성 및 연기

➤ 민보욱

- ❖ Flutter 기반 메인 앱 제작 및 디자인
- ❖ Naver Map API 사용
- ❖ Firebase 연동 및 데이터 관리

➤ 이동훈

- ❖ 딥러닝 모델 생성 및 이미지 처리
- ❖ Flask API 제작 및 배포
- ❖ React 기반 부가기능 웹페이지 제작

➤ Schedule

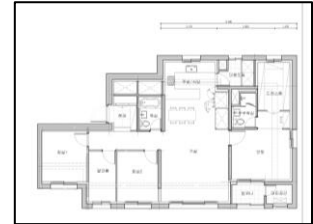
| 주차 | 주요 프로젝트 스케줄 |
|-------|---|
| 1~2 | 아이디어 선정 및 Proposal Report 작성 |
| 3~7 | AI 관련 딥러닝 학습(Image Segmentation) 및 백엔드 구상 |
| 8 | 중간고사 |
| 9~10 | Frontend 개발 시작(Flutter), Firebase 연동 |
| 11~12 | 알고리즘 길 찾기를 위한 데이터 가공 |
| 13 | 길 찾기 알고리즘 제작, 데이터 처리 자동화 |
| 14 | Flask API 제작 및 배포, Frontend 연동 |
| 15 | 객체 내 정보를 채울 수 있는 웹페이지 제작 |
| 16 | 최종 보고서 작성 |

○ Feedback

아쉬웠던 AI분야(Image Segmentation) 결과

➤ 원인분석

- ❖ 팀원 모두 AI(딥러닝) 관련 지식이 전무함에도 무리한 아이디어 선정
 - 개발 일정이 전체적으로 늦어 짐, 모델 결과 성능 개선 미약
- ❖ 아이디어에 정확하게 맞는 데이터셋을 찾지 못함에 따라 정확도 하락
 - 아이디어를 먼저 선정하고 데이터셋을 찾았기 때문에 일반적인 건물이 아닌 아파트의 데이터셋 활용



➤ 팀원 회고

- ❖ 처음에 걱정했던 것에 비하면 이미지에 따라 꽤 관찰은 결과물을 얻을 수 있음
- ❖ 결과적으로 충분히 도전해볼만 했던 기능이었음
- ❖ OCR 기술을 적용하지 못한 것은 여전히 아쉬움

○ 결과분석

- ❖ 초기 목표였던 플랫폼화를 구축하는 것은 성공
 - 모든 과정 자동화 (AI 기술이 완벽 해졌다는 가정하)
- ❖ 각 건물의 도면(안내도) 데이터를 활용하여 길 찾기 부분적 구현
 - 정확도를 위해 중간 과정에 수작업 필요
- ❖ 각 객체에 정보 또한 유저를 통해 입력 받는 방식 구현
 - 여러 유저의 정보를 업로드 받는 방식으로 대형 데이터베이스 구축 가능
 - 변경사항이 있을 시 실시간 수정 가능
- ❖ 많은 투자 없이도 간단하게 모든 건물의 지도 열람과 길 찾기를 가능하게 한다는 점을 장점으로 삼아 다른 플랫폼과의 연계 등 발전 가능성이 높음

7. Release and Source Code

○ Release

- ❖ Demo APK Release _ IndoorMap.apk

https://github.com/PROMLEE/Indoor_map_flutter/releases/tag/Release

- ❖ Website Release _ Indoor Map 수정하기

https://promlee.github.io/Indoor_map_react/

- ❖ API _ Flask 서버 통신 URL (서버는 2023년 12월 31일까지 가동)

<https://indoormap.store>

ex) get 요청: mask 파일 반환

https://indoormap.store/mask/CAU310_02

○ Source Code Link (github repo)

- ❖ Indoor_map_flutter: flutter 기반 Application 제작 code

https://github.com/PROMLEE/Indoor_map_flutter

- ❖ Indoor_map_algorithm: Flask 기반 API 서버(AI모델, 길 찾기 algorithm 포함) - 현재 AWS EC2 인스턴스로 배포 중 (서버는 2023년 12월 31일까지 가동)

https://github.com/PROMLEE/Indoor_map_algorithm

example.h5: 딥러닝 학습모델(용량 문제로 별도로 관리), scripts 폴더에 저장

https://drive.google.com/file/d/1Rvcbt_1Dbu_joxERk2QEQd50t-UhX0x3/view?usp=drive_link

- ❖ Indoor_map_react: React 기반 데이터 수정용 웹 페이지

https://github.com/PROMLEE/Indoor_map_react