

# 다중운집지역 내 인파 밀집도 계산 및 실시간 위치 추적 기술 개발



천주희

장은서

지도교수 김태운

---

## 목 차

1. 과제 개요.....	1
1.1. 과제 배경.....	1
1.2. 기존 문제점.....	2
1.3. 과제 목표.....	3
2. 설계 과정.....	4
2.1. 요구 조건.....	4
2.1.1. 무선 신호 기반 위치 추정 기술.....	4
2.1.2. 비전 기반 위치 추정 기술 및 실시간 영상 스트리밍.....	4
2.1.3. 웹 서비스.....	5
2.2. 기존 제약사항에 대한 수정사항.....	12
2.2.1. 클라우드 서버 배포.....	12
2.2.2. 스트리밍용 웹 서버 구축.....	12
2.2.3. 무선 인터넷 관련.....	12
2.2.4. 카메라 설치 위치.....	14
2.2.5. 반응형 웹.....	14
2.2.6. 사용자 식별 기능.....	14
2.2.7. 카메라 사용 용도 변경.....	15
2.2.8. 이웃 기능 추가.....	15
2.2.9. YOLOv5n 사용.....	15
2.3. 시스템 구성.....	16
2.3.1. 무선 신호 기반 위치 추정.....	16
2.3.2. 비전 기반 위치 추정.....	16

---

2.3.3.	객체 인식 결과 영상 스트리밍.....	17
2.3.4.	웹 서비스.....	17
2.4.	개발 환경.....	17
3.	과제 설계.....	18
3.1.	과제 배경 지식.....	18
3.1.1.	무선 신호 기반 위치 추정 기술.....	18
3.1.2.	비전 기반 위치 추정 및 결과 영상 스트리밍.....	21
3.1.3.	공유 메모리.....	22
3.1.4.	웹 서비스.....	22
3.1.5.	반응형 웹.....	24
3.2.	기술 상세.....	25
3.2.1.	무선 신호 기반 실시간 위치 추정.....	25
3.2.2.	비전 기반 실시간 사람 인식.....	26
3.2.3.	카메라 인식 위치 보정.....	26
3.2.4.	실시간 영상 스트리밍.....	27
3.2.5.	밀집도 예측.....	28
4.	과제 결과 분석 및 평가.....	29
4.1.	메인 페이지.....	29
4.2.	로그인 페이지.....	31
4.3.	현재 위치 및 밀집도.....	32
4.4.	무선 통신 기반 밀집도.....	33
4.5.	밀집도 경고 알림.....	35
4.6.	비전 기반 실시간 위치 추정.....	36
4.7.	실시간 밀집도 통계 및 확인.....	37

---

4.8.	제보.....	39
4.9.	사용자 찾기 .....	43
4.10.	특정 구역 사용자 찾기 .....	45
4.11.	이웃 .....	46
4.12.	회원 관리.....	50
4.13.	설정 .....	51
4.14.	반응형 웹.....	55
5.	결론 및 향후 연구 방향 .....	58
5.1.	결론.....	58
5.2.	향후 연구 방향 .....	58
6.	개발 일정 및 역할 분담 .....	59
6.1.	개발 일정.....	59
6.2.	역할 분담 .....	59
7.	참고 문헌.....	60

## 1. 과제 개요

### 1.1. 과제 배경

작년 10월경에 발생한 이태원 참사를 통해 인원 과밀 상황에서 발생하는 압사 사고에 대한 대비책이 필요하다는 국민적 시선이 생겨났다. 하지만 OECD 인구밀도 1위 국가인 대한민국에서 실내 인구 과밀 상황에 노출되는 일은 여전히 불가피하며, 실외가 아닌 실내 다중밀집 사고에 대한 대비책은 여전히 부족한 실정이다.

서울의 지하철 9호선과 김포골드라인 등의 혼잡도가 높은 수도권 지하철 노선들은 출퇴근 시간마다 열차 내부가 승객들로 가득 찬다. 서울 9호선 급행열차의 1㎡당 밀집도는 5명 안팎으로 미 연방 재난관리청의 최대 밀집 권고 2~3명의 2배 수준이다. 김포골드라인의 경우에는 1㎡당 밀집도가 7~8명에 가까워 권고의 3배 수준에 육박한다. 이는 이태원 참사 당시의 밀집도와 유사한 수준으로, 일상 속에서도 실내 과밀 상황으로 인한 인파 사고 발생 가능성이 충분히 존재한다는 점을 시사한다.

서울연구원의 통계에 따르면 프랑스 파리, 일본 도쿄, 중국 베이징, 싱가포르 등 주요 도시와 비교해 볼 때, 대한민국 서울의 인구밀집도는 가장 높은 수준이다. 이러한 높은 인구밀집도에 익숙한 국민들은 인구 과밀 상황에 둔감해진다. 이는 위험 상황에 대한 대비 부족으로 이어지며, 사고의 규모를 키우는 원인이 된다.



그림 1-1. 서울 지하철 최대 밀집도 비교



그림 1-2. 세계 주요 시가지 인구밀집도

---

둔감함의 결과로 벌어진 이태원 참사와 같은 압사 사고를 되풀이하지 않기 위해서는 군중 밀집에 대한 국민적 경각심을 높이고, 과밀 상황으로 인한 사고를 대비해야 한다. 이를 위해 국민들은 군중 밀집도와 같은 지표를 가까이하고, 객관적인 수치를 통해 자신이 처한 상황의 심각성을 인지할 필요가 있다. 또한 정부와 기관에서 과밀 상황을 방지하고 통제할 수 있는 도구를 통해 사고를 예방하려는 노력이 필요하다. 본 과제는 사람들의 군중 밀집에 대한 경각심을 높이고 실시간 실내 밀집도 모니터링을 통해 과밀 상황에서의 사고를 대비하려 한다.

## 1.2. 기존 문제점

국내 최대 사상자를 낸 이태원 참사가 일어난 이후, 실외의 인파 밀집 사고를 예방하기 위한 방법들을 도입하고 있지만 그 시선이 실내로 향하지는 않고 있다.

(부산 불꽃축제의 밀집도 현황 스크린, 서울시 인파 계산 CCTV)

그러나 세계적인 주요 압사 사고 통계를 살펴보면 실외뿐만 아니라 실내 다중밀집사고가 꾸준히 발생하고 있는 실정이다. 아래는 실제 국내외 압사 사고 사례 중 일부이다.

- 1960년 1월 26일 서울역 계단 압사 사고: 설을 앞두고 승객들이 몰린 서울역 계단에서 사람들이 한꺼번에 넘어지며 31명 압사.
- 1992년 2월 17일 뉴키즈온더블록 내한 공연 압사 사고: 서울 잠실 올림픽 체조경기장에서 열린 미국 그룹 뉴키즈온더블록의 내한공연에서 공연 도중 팬들이 무대 앞에 몰려들어 여고생 1명 압사, 50여명 실신.
- 2023년 3월 5일 로체스터 공연장 압사 사고: 미국 로체스터 공연장에서 콘서트 도중 총을 쏘는 듯한 소리가 나면서 관객이 출입구로 몰려 2명이 압사, 8명 부상.

공연장, 지하철과 같이 많은 인파가 몰리는 장소는 대체로 실내가 많으며, 이에 따라 사고 가능성 또한 높아진다. 하지만 참사 이후 도입된 압사 사고 대비책은 모두 그 대상을 실외로 제한하고 있다. 이러한 점으로 미루어 볼 때, 실내 다중밀집사고에 대한 대비 및 대책의 부재는 시급히 해결되어야 하는 문제점으로 사료된다.

---

### 1.3. 과제 목표

본 과제는 다중 운집 지역에서의 실시간 밀집도 계산 및 위치 추적 웹 서비스를 제작하여 인파 사고를 대비하고자 한다.

- 기준 구역 내의 실시간 밀집도 계산 및 기준 구역 내 위치 추정
  - 무선 신호 기반으로 사용자 위치를 구역 단위로 추정한다.
  - 비전 기반으로 실내 인원의 정밀한 위치를 추정한다.
  - 위치 추정 정보를 통해 인원 수 측정 및 밀집도 등급을 결정한다.
  - 사용자에게 위 정보를 실시간으로 제공한다.
- 다중 운집 지역에서 재난/재해 또는 실종 사고 발생 시, 해당 사용자의 실내 존재 여부 및 위치 파악
  - 군중 밀집으로 인한 사고 이외에도 화재, 붕괴, 실종과 같은 사고 발생 시, 건물에 존재하는 사용자들의 위치를 빠르게 파악해 구조할 수 있다.
- 다중 운집 구역에서 인파 밀집도에 따른 경보/알림 시스템
  - 인파가 많이 몰린 곳에서 사용자가 위험 상황을 신속하게 인지 및 대비할 수 있도록 한다.

---

## 2. 설계 과정

### 2.1. 요구 조건

본 과제는 무선 신호 기반 위치 추정 기술과 비전 기반 위치 추정 기술 그리고 두 기술의 동작 결과를 활용해 사용자에게 밀집도 관련 정보를 제공하는 웹 서비스로 구성되어 있다. 따라서 무선 신호 기반 위치 추정 기술과 비전 기반 위치 추정 기술, 웹 서비스를 기준으로 요구 조건을 서술하고자 한다.

또한 모든 요구 조건은 부산대학교 제6공학관 2층을 기준으로 작성한다.

#### 2.1.1. 무선 신호 기반 위치 추정 기술

- 위치 추정 대상을 라즈베리파이로 가정한다.
- 가로, 세로 10m의 정사각형 공간을 단위로 영역을 구분하며, 6x4 형태로 총 24개의 좌표를 할당한다.
- 서버는 클라이언트에서 측정한 데이터를 전송받아 위치를 추정한다.
- 서버는 위치 추정 후, 해당 라즈베리파이의 MAC 주소와 좌표 정보를 웹 서버로 전송한다.
- 서버는 다중 클라이언트와 통신이 가능하다.
- 서버는 다중 클라이언트의 데이터 처리 및 위치 추정 계산이 가능하다.

#### 2.1.2. 비전 기반 위치 추정 기술 및 실시간 영상 스트리밍

##### 2.1.2.1. 비전 기반 위치 추정 기술

- 구동 환경을 카메라 모듈이 연결된 라즈베리파이로 가정한다.
- 카메라 모듈을 통해 촬영하는 실시간 영상을 기반으로 동작한다.
- 사람만을 객체로 인식한다.
- 인식된 객체의 좌표와 카메라의 설치 위치 및 각도 정보를 이용하여 해당 객체의 평면상 위치를 계산하고 좌표를 변환한다.
- 카메라 설치 위치 정보를 통해 평면상 좌표 값의 건물 평면도 내 위치를 계산한다.
- 인식된 모든 객체의 건물 평면도 내 위치 정보를 웹 서비스 서버로 전송한다.



---

#### 2.1.2.2. 실시간 영상 스트리밍

- 구동 환경을 카메라 모듈이 연결된 라즈베리파이로 가정한다.
- 카메라 모듈을 통해 촬영하는 실시간 영상을 기반으로 동작한다.
- 사람만을 객체로 인식한다.
- 객체 인식 결과를 실시간으로 송출하기 위한 웹 서버를 구동한다.
- 객체 인식 후 산출된 영상의 프레임을 공유 메모리에 저장한다.
- 웹 서버에서 공유 메모리에 저장된 영상의 프레임을 읽어와 화면에 출력한다.

#### 2.1.3. 웹 서비스

##### 2.1.3.1. 현재 위치에서의 밀집도

- 서비스 사용자에게 자신의 현재 위치 및 밀집도를 실시간으로 제공한다.
- [무선 신호 기반 위치 추정 기술](#)에서 측정한 위치 정보를 이용한다.
- 사용자가 서비스에 회원가입을 한 상태이며 라즈베리파이를 소지하고 있어야 한다.
- 사용자 정보에 소지한 라즈베리파이의 MAC 주소가 등록된 상태를 전제로 한다.
- 평면도와 현재 로그인한 사용자의 위치 정보, 현재 위치에 함께 있는 사용자 정보를 화면에 띄운다.
- 평면도에 현재 로그인한 사용자의 위치에 해당하는 부분을 색으로 표시한다.
- 현재 로그인한 사용자의 위치 정보에는 사용자의 마지막 위치와 기록 시간, 현재 해당 위치의 밀집도를 포함하며, 해당 정보를 1초마다 업데이트 한다.
- 현재 위치에 함께 있는 사용자들의 아이디와 프로필 이미지를 출력하며, 해당 정보를 1초마다 업데이트 한다.
- 라즈베리파이가 등록되지 않은 사용자의 경우 페이지 접속 시, 경고 창을 띄운 후 [무선 통신 기반 밀집도](#)로 이동한다.

---

#### 2.1.3.2. 무선 통신 기반 밀집도

- 서비스 사용자에게 건물 내 전반적인 실내 밀집도 정보를 제공한다.
- [무선 신호 기반 위치 추정 기술](#)에서 측정한 위치 정보를 이용한다.
- 밀집도 등급 및 인원 수는 100m<sup>2</sup>를 기준으로 측정한다.
- 평면도와 밀집도 표를 화면에 띄운다.
- 밀집도 표는 좌표와 실시간 인원 수 및 밀집도 등급을 포함한다.
- 평면도에 좌표 별 밀집도 등급을 색상을 통해 표현한다.
- 평면도에서 특정 좌표에 마우스를 올리면 해당 좌표와 실시간 인원 수를 평면도 중앙에 출력한다.
- 평면도에서 특정 좌표를 클릭하면 밀집도 표의 해당 좌표 정보가 담긴 행을 포커싱한다.

#### 2.1.3.3. 실시간 밀집도 알림

- 밀집도 등급에 기반하여 위험 상황을 판단해 서비스 사용자에게 이를 알린다.
- [무선 신호 기반 위치 추정 기술](#)에서 측정된 위치 정보를 이용한다.
- 현재 사용자가 로그인한 상태이며 라즈베리파이를 소지하여 서비스 등록이 완료된 상태를 전제로 한다.
- 사용자 위치와 해당 위치의 밀집도를 1초마다 확인하여 현재 밀집도가 위험 이상일 시 화면 우측 하단에 알림을 띄운다.
- 알림의 X 버튼을 누르면 해당 페이지에서는 재 출력하지 않는다.
- 밀집도가 위험 이상으로 지속된다는 가정 하에 특정 페이지에서 알림을 닫아도 새로운 페이지로 이동하면 알림을 화면에 재 출력한다.

---

#### 2.1.3.4. 비전 기반 밀집도

- 서비스 사용자에게 실내 체류 인원의 정밀한 위치 정보를 시각적으로 제공한다.
- 서비스 사용자에게 실내 상황 및 밀집도를 실시간 영상으로 제공한다.
- [비전 기반 위치 추정 기술](#)에서 계산된 위치 정보를 이용한다.
- 평면도와 카메라 별 인원 수 표를 화면에 띄운다.
- 평면도에 객체의 위치를 빨간 점으로 표현한다.
- 평면도에는 카메라가 설치된 위치에 카메라 아이콘이 함께 나타난다.
- 평면도의 카메라 아이콘을 클릭하면 해당 카메라의 [실시간 영상 스트리밍](#) 페이지 출력하여 객체 인식 결과를 사용자에게 제공한다.

#### 2.1.3.5. 실시간 밀집도 및 통계

- 서비스 사용자에게 시간에 따른 실내 밀집도의 변화 및 종합적인 밀집도 정보를 제공한다.
- [무선 신호 기반 위치 추정 기술](#)과 [비전 기반 위치 추정 기술](#)에서 측정 및 계산한 위치 정보를 이용한다.
- 공간을 4분할하여 가, 나, 다, 라로 구분한다.
- 평면도와 구역별 인원 수 표, 시간별 인원 수 그래프를 화면에 띄운다.
- 평면도의 구역별 인원 수는 무선 통신 기반 인원 수와 비전 기반 인원 수를 합한다.
- 평면도의 구역별 밀집도 등급을 평면도 위에 색상으로 표시한다.
- 평면도의 특정 구역을 클릭하면 해당 구역의 시간별 인원 수 그래프를 화면에 출력한다.
- 구역별 인원 수 표에는 구역 이름, 무선 통신 기반 인원 수와 비전 기반 인원 수 및 밀집도 등급을 포함한다.
- 시간별 인원 수 그래프는 최근 2시간 30분의 인원 수를 30분 단위로 차트에 표시한다.
- 시간별 인원 수 그래프는 30분 후의 인원 수를 예측하여 차트에 표시한다.
- 시간별 인원 수 그래프의 EXPORT TO EXCEL 이미지를 클릭하면 현재까지 데이터베이스에 저장된 모든 인원 수 정보를 엑셀 파일로 제공한다.

- 
- 엑셀 파일은 구역 별로 시트를 생성한다.
  - 엑셀 파일은 하나의 시트에 무선 신호 기반 인원 수, 비전 기반 인원 수, 총 인원 수 및 생성 시간을 포함한다.

#### 2.1.3.6. 제보하기

- 서비스 사용자가 위험 상황을 직접 알릴 수 있는 플랫폼을 제공한다.
- 사용자는 서비스에 회원가입 및 로그인을 해야 한다.
- 위치, 제목, 내용 작성 란과 제출 버튼을 화면에 띄운다.
- 위치는 사용자가 직접 입력할 수 없으며, 위치 선택 버튼을 눌러 나타나는 평면도 이미지에서 원하는 구역을 선택한다.
- 위치 선택을 하지 않거나 제목을 입력하지 않으면 제출을 거부하고 안내 창을 띄운다.
- 제출 완료 시 [제보 게시판](#)으로 이동한다.

#### 2.1.3.7. 제보 게시판

- 제보 데이터를 서비스 사용자에게 제공한다.
- 위치, 제목, 작성자 및 작성일자를 화면에 띄운다.
- 특정 게시물의 제목을 클릭하면 제보 세부 내용을 조회할 수 있다.
- 제보 세부 내용은 위치, 작성자, 제목 및 내용을 포함한다.
- 작성자의 닉네임 변경 시 작성된 게시물에 이를 반영한다.

#### 2.1.3.8. 제보 알림 기능

- 제보 작성 시 작성자를 제외한 모든 서비스 사용자에게 이를 알린다.
- 사용자는 서비스에 회원가입 및 로그인을 해야 한다.
- 제보 알림은 헤더의 우측 상단 종 모양 이미지에서 확인 가능하다.
- 제보 알림의 개수를 종 모양 이미지 위에 숫자로 출력한다.
- 종 모양 이미지에 마우스를 올리면 알림 창을 띄운다.
- 알림 창 내부에 제보 작성 일자 및 제목, 내용을 출력한다.

- 
- 특정 알림을 클릭하면 [제보 게시판](#)의 제보 세부 내용 페이지로 이동한다.

#### 2.1.3.9. 사용자 위치 조회

- 이메일을 통해 서비스 사용자의 최신 위치 정보를 조회한다.
- [무선 신호 기반 위치 추정 기술](#)에서 측정한 위치 정보를 이용한다.
- 기능 사용자를 서비스 관리자로 제한한다.
- 위치를 조회하고 싶은 이웃은 서비스에 회원가입을 한 상태이며 라즈베리파이를 소지하고 있어야 한다.
- 사용자의 이메일을 검색창에 입력하고 검색 버튼을 누르면 사용자의 최신 위치와 기록 시간, 기록 당시 해당 위치에서의 밀집도 등급을 화면에 띄우며, 해당 정보를 1초마다 업데이트 한다.
- 검색된 사용자의 위치를 평면도에 색으로 표시한다.
- 회원가입을 하지 않거나 라즈베리파이가 등록되지 않은 사용자 조회 시, 경고 창을 띄운 후 검색창이 리셋된다.

#### 2.1.3.10. 특정 구역 사용자 찾기

- 특정 구역에 머무르는 서비스 사용자들의 정보를 시간별로 조회한다.
- [무선 신호 기반 위치 추정 기술](#)에서 측정한 위치 정보를 이용한다.
- 기능 사용자를 서비스 관리자로 제한한다.
- 평면도를 화면에 띄운다.
- 평면도에서 검색을 원하는 구역을 클릭하면 해당 구역에 현재 시간을 기준으로 1분 내에 머물렀던 서비스 사용자들의 정보를 출력한다.
- 우측 상단의 드롭다운 리스트를 통해 1분, 3분, 5분, 10분으로 시간 옵션을 변경할 수 있다.

---

#### 2.1.3.11. 이웃 조회

- 서로 이웃 관계의 서비스 사용자 정보를 조회한다.
- 사용자는 서비스에 회원가입 및 로그인을 해야 한다.
- 프로필 이미지와 닉네임, 이메일 및 위치 보기 버튼과 삭제 버튼을 화면에 띄운다.
- 위치 보기 버튼을 누르면 [이웃 위치 조회](#)가 가능하다.
- 삭제 버튼을 누르면 이웃 목록에서 삭제된다.
- 우측 상단의 이미지 위에 [이웃 신청](#) 개수를 숫자로 출력한다.
- 우측 상단의 이미지를 클릭하면 [이웃 신청](#) 페이지로 이동한다.

#### 2.1.3.12. 이웃 위치 조회

- 서로 이웃 관계의 서비스 사용자의 최신 위치 정보를 제공한다.
- 사용자는 서비스에 회원가입 및 로그인을 해야 한다.
- 위치를 조회하고 싶은 이웃은 서비스에 회원가입을 한 상태이며 라즈베리파이를 소지하고 있어야 한다.
- 이웃의 최신 위치, 기록 시간 및 기록 당시 해당 위치에서의 밀집도 등급을 [이웃 조회](#) 화면의 우측에 띄우며, 해당 정보를 1초마다 업데이트 한다.
- 라즈베리파이가 등록되지 않은 이웃 조회 시, 경고 창을 띄운 후 이웃 위치 조회 화면이 닫힌다.

#### 2.1.3.13. 이웃 신청

- 서비스 이용자가 이웃 신청을 하거나 받은 이웃 신청을 수락/거절할 수 있다.
- 사용자는 서비스에 회원가입 및 로그인을 해야 한다.
- 이웃 신청을 원하는 사용자는 서비스에 회원가입을 해야 한다.
- 이메일 검색창과 이웃 신청 목록을 화면에 띄운다.
- 이메일 검색창에 이메일을 입력하고 추가 버튼을 누르면 이웃 신청이 가능하다.
- 본인에게 이웃 신청을 하거나 이미 이웃 신청을 사용자, 이웃인 사용자에게 이웃 신청 또는 회원가입을 하지 않은 사용자에게 이웃 신청 시 경고 창을 띄운다.

- 
- 이웃 신청 목록에는 프로필 이미지와 이름, 수락/거절 버튼을 출력한다.

#### 2.1.3.14. 회원 관리

- 회원의 라즈베리파이 소지 정보를 등록/변경/삭제할 수 있다.
- 기능 사용자를 서비스 관리자로 제한한다.
- 등록할 사용자는 서비스에 회원가입을 해야한다.
- 라즈베리파이의 MAC 주소와 부여된 ID, 소지한 사용자의 이메일 정보를 화면에 띄운다.
- 이미 회원 정보가 등록된 라즈베리파이의 경우, "변경"을 누르면 버튼이 "등록"으로 바뀌며 사용자 이메일 입력란이 활성화되어 수정이 가능하다.
- 회원 정보가 등록되지 않은 라즈베리파이의 경우 사용자 이메일 입력란이 기본적으로 활성화 되어있다.
- 이메일 입력 후 등록 버튼을 누르면 해당 정보가 반영되며 이메일 입력란이 비활성화된다.
- 만약 이미 등록된 사용자를 중복 등록하거나 회원 가입하지 않은 사용자를 등록하면 경고 창을 띄우며 이메일 변경 내역을 복구하고 입력란이 비활성화 된다.

#### 2.1.3.15. 설정

- 서비스 사용자에게 개인 정보 조회 및 수정 기능을 제공한다.
- 사용자는 서비스에 회원가입 및 로그인을 해야 한다.
- 현재 프로필 사진과 파일 선택/이미지 삭제/저장버튼, 닉네임 변경 란과 수정 버튼을 화면에 띄운다.
- 프로필 이미지를 수정한 뒤, 저장 버튼 누르면 적용된다.
- 닉네임을 수정한 뒤, 수정 버튼을 누르면 적용된다.

---

## 2.2. 기존 제약사항에 대한 수정사항

### 2.2.1. 클라우드 서버 배포

- 로컬 환경에서 서버 배포 시 물리적 서버 관리의 어려움이 있고, 트래픽 증가에 대한 대처가 불가능하다.
- 재난 상황의 트래픽 과부화에 대비하기 위해 부하 분산 및 서버 확장 용이한 클라우드 서버 배포를 진행했다.

### 2.2.2. 스트리밍용 웹 서버 구축

- 객체 인식 결과를 프레임단위로 웹 서비스에 전송해 페이지에서 직접 출력하려고 했으나, 프레임 데이터가 서버 측에서 처리할 수 있는 길이를 초과하는 이슈가 있었다. 따라서 객체 인식이 동작하는 라즈베리파이에 웹 서버를 구축하는 것으로 구현 방법을 변경했다.
- 이후 stateless한 서버의 특성으로 인해 http 통신을 이용한 실시간 데이터 교환이 현실적으로 불가능해 객체 인식 결과를 공유 메모리를 통해 전달하고, 웹서비스에서는 iframe을 통해 해당 서버에 접속하는 방식으로 스트리밍 기능을 구현했다.

### 2.2.3. 무선 인터넷 관련

- 스트리밍 웹 서버의 경우 사설 IP를 할당 받아 사용하는데, 클라우드 서버에 배포된 웹서비스가 해당 서버에 접근하기 위해서는 공인 IP로의 포트포워딩이 필수적이다.
- 학교에서 제공하는 사설 IP를 대상으로 정보화 본부 측에 공인 IP NAT를 신청했지만 학교 자체 공유기 및 무선 인터넷의 경우, 공인 IP 제공이 불가능하다는 정보화 본부의 안내를 받았다.
- 따라서 과 사무실의 동의를 얻어 개인 공유기를 창의마루 동아리실 B에 설치하고 랜선 분배기를 통해 해당 공유기에 사설 IP를 할당 받았다. 이후 해당 IP에 대해 정보화 본부에 공인 IP NAT를 요청해 정상적으로 할당 받았다.
- 이후 공인 IP에 스트리밍용 웹 서버를 위한 포트 포워딩을 진행했다.



### 공인 IP(NAT)/방화벽 신청서

교직원 번호				(교직원 번호 기재)
신청자 성명				
신청자 소속	정보의생명공학대학 정보컴퓨터공학부			(예: 인문대학 국어국문학과)
연락처	사무실 ☎		핸드폰 ☎	(사무실과 스마트폰 연락처 필수 기재)
E-Mail				(학교 이메일 주소 기재)
사용기간	2023.09.01. - 2023.11.30			(방화벽서비스 요청시 사용기간은 신청 후로부터 최대 5년 제한)
신청내용	IP 주소	사설 IP		
	외부에서 발급 받은 공인IP로 접근해야 할 경우 아래 내용 작성 필요			
	방화벽	출발지 IP	all	
		목적지 IP		
	서비스	포트 번호	22, 5000, 5001, 5002, 5003	
		적용 Rule	웹, SSH	
		신청목적	졸업 과제(카메라 스트리밍)	
	도메인 사용여부	X		
신청일자	2023년 9월 6일			
안내사항	▶ 신청방법 : 신청서 작성 후 정보화본부 공인IP 신청 게시판 업로드 ▶ 적용안내 : 공인 IP 적용 후 학교 이메일로 발송 ▶ 사용문의 : 정보화본부 (☎ 교내 7480, 7481)			
처리 일시	적용 일자	처리 담당자	적용 일자	처리 담당자
부산대학교 정보화본부장 귀하				

그림 2-1. 공인 IP 신청서

안녕하십니까 정보화본부 정보인프라팀입니다.

요청하신 1:1 NAT 공인IP 전달해드립니다.

1:1 NAT 공인 IP	사설 IP	개방 포트
		22, 5000, 5001, 5002, 5003

감사합니다.

그림 2-2. 공인IP 및 개방 포트

#### 2.2.4. 카메라 설치 위치

- 기존에는 카메라를 모든 복도의 모서리에 1대씩 설치할 예정이었지만 무선 인터넷 범위의 제약으로 위치를 변경하였다.

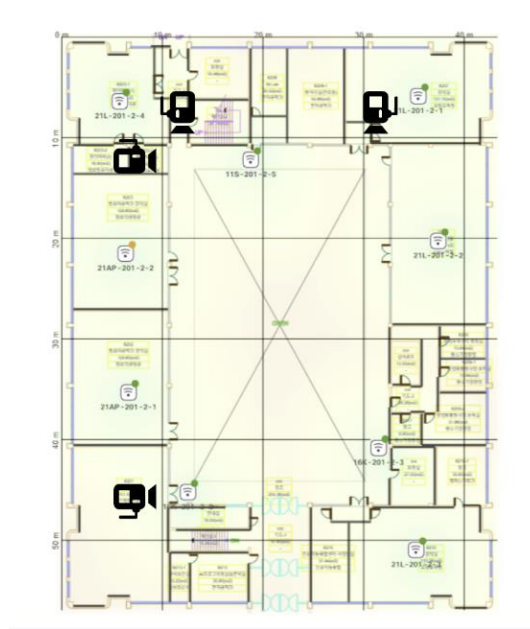


그림 2-3. 변경된 카메라 위치

#### 2.2.5. 반응형 웹

- 본 과제는 초기에 데스크탑에 최적화된 디자인으로 개발됐다. 하지만 해당 서비스는 사용자가 이동하면서 실시간으로 자신의 위치를 확인하여 위험 상황을 감지하는 것을 목표로 하므로 사용자가 휴대폰, 태블릿과 같이 이동 통신 매체를 통해 접속하는 상황을 고려해야 했다.
- 회의 결과 다양한 매체에서 접근을 할 수 있다는 점을 고려하여 모든 페이지에 반응형 웹을 적용했다.

#### 2.2.6. 사용자 식별 기능

- 서비스 사용자가 증가할수록 얼굴 마커만으로 사용자 구분이 힘들고 도면 내에 출력되는 마커의 사이즈가 크지 않아 얼굴의 특징을 뚜렷하게 나타내기 어렵다는 한계가 있었다.
- 라즈베리파이 소지자의 식별 기능을 유지하기 위해서 얼굴 마커 기능을 대체하여 관리자가 각 라즈베리 파이의 MAC 주소와 사용자 이메일을 매칭하여 등록하는 회원 관리 기능 및 페이지를 구현했다.

### 2.2.7. 카메라 사용 용도 변경

- [사용자 식별 기능](#)에서 언급한 제약사항과 함께 라즈베리파이 미 소지 인원을 포함한 실내 밀집도를 파악하고자 했다.
- 카메라와 딥러닝 기반의 실시간 객체 인식 툴을 이용하여 특정 구역내의 실내 인원 수 측정 및 정밀 위치 추정을 위해 사용하는 것으로 용도를 변경했다.

### 2.2.8. 이웃 기능 추가

- 서비스 이용자가 타 사용자의 위치를 찾기 위해서는 항상 서비스 관리자를 찾아 확인 요청을 해야 하는 것은 시간적 비용이 크다.
- 서비스 이용자가 상호 동의 하에 서로의 위치를 실시간으로 확인할 수 있는 이웃 추가 및 이웃 위치 파악 기능을 구현했다.

### 2.2.9. YOLOv5n 사용

- 라즈베리파이와 카메라 모듈을 사용하여 객체 인식을 진행하므로 일반적인 랩탑이나 데스크탑에 비해 성능이 낮아 속도 저하 이슈가 발생했다.
- 본 과제에서는 다양한 객체가 아닌 사람 인식만을 요구하므로 상대적으로 성능이 낮고 FPS가 가장 높은 경량화 모델 YOLOv5n을 사용하여 객체 인식을 하였다.

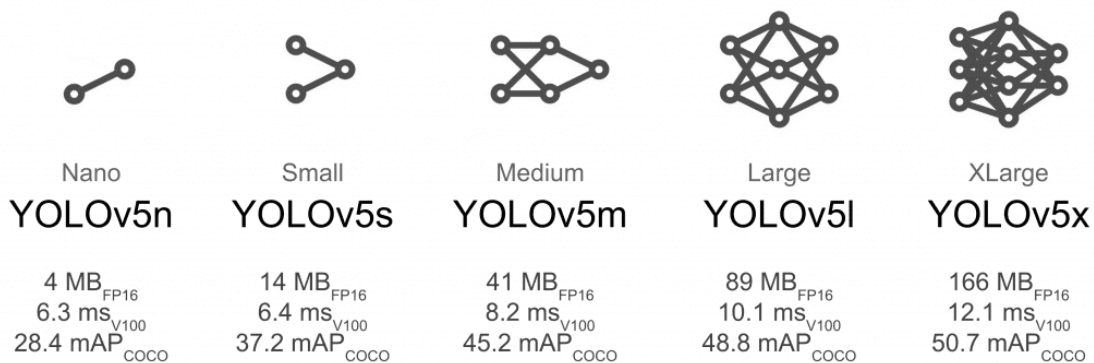


그림 2-4. YOLOv5

## 2.3. 시스템 구성

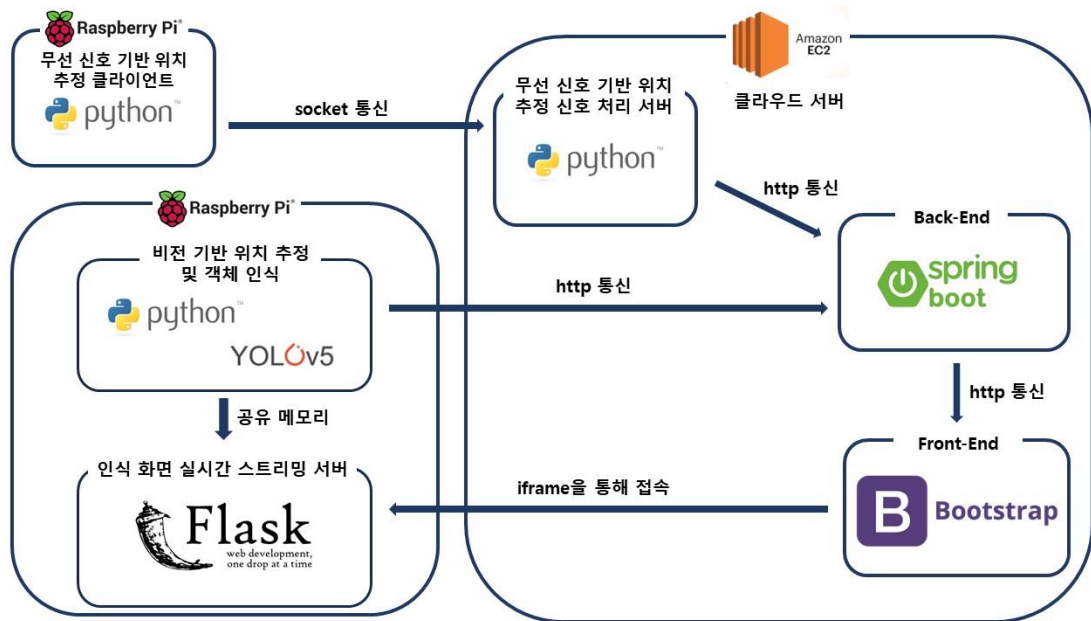


그림 2-5. 시스템 구성도

### 2.3.1. 무선 신호 기반 위치 추정

- 본 기술은 클라이언트와 서버로 나뉜다.
- 클라이언트는 무선 신호 값을 측정하여 서버로 전송한다.
- 서버는 클라이언트에게서 전송받은 신호 값을 통해 위치를 추정하고 http 통신을 통해 웹 서버로 위치 데이터를 보낸다.
- 클라이언트는 라즈베리파이 기반으로 동작하며 파이썬으로 작성된 프로그램이다.
- 서버는 클라우드 서버인 Amazon EC2를 기반으로 동작하며, 파이썬으로 작성된 프로그램이다.
- 서버는 select 모듈을 활용하여 여러 대의 클라이언트 요청을 동시에 처리 가능하다.

### 2.3.2. 비전 기반 위치 추정

- 라즈베리파이를 기반으로 동작하는 파이썬 프로그램이다.
- 카메라 모듈을 통해 위치 추정을 위한 영상을 제공한다.
- 객체 인식 결과 영상을 프레임 단위로 공유 메모리에 저장한다.
- 객체 인식을 통해 계산된 위치 데이터를 웹 서버에 http통신을 이용해 전송한다.

---

### 2.3.3. 객체 인식 결과 영상 스트리밍

- [비전 기반 위치 추정](#)을 구동하는 라즈베리파이에서 함께 동작한다.
- 파이썬과 Flask 프레임워크를 이용해 구현한 웹 서버이다.
- [비전 기반 위치 추정](#)에서 저장한 공유 메모리 데이터를 읽어와 화면에 출력한다.
- 웹 서비스에서 스트리밍 페이지에 접근이 가능하도록 포트 포워딩을 구현한다.

### 2.3.4. 웹 서비스

- 백엔드는 스프링 부트 프레임워크를 사용하며, 프론트엔드는 부트스트랩 라이브러리를 사용한다.
- 서버는 클라우드 서버인 Amazon EC2를 기반으로 동작한다.

## 2.4. 개발 환경

- Spring Boot 2.7.11
- JDK 11
- AWS EC2 Amazon Linux 2
- Python 3.7이상 (무선 기반 위치 추정 기술의 클라이언트)
- Python 3.11이상 (비전 기반 위치 추정 기술)
- Raspberry Pi 4 Model B
- YOLOv5
- Bootstrap

---

### 3. 과제 설계

#### 3.1. 과제 배경 지식

##### 3.1.1. 무선 신호 기반 위치 추정 기술

###### 3.1.1.1. Fingerprint

- 무선 신호 기반 위치 추정 시, 사용자의 위치 추정 기준이 되는 Radio Map을 구성하기 위해 핑거프린트 방식을 사용한다.
  - 관리자가 실내의 모든 위치마다 주변 AP로부터의 무선 수신신호 세기(Received Signal Strength, 이하 RSS)를 측정하여 핑거프린트 맵을 구성한다.
  - 사용자의 현재 위치에서 측정한 주변 AP들의 RSS 값을 이용해 가장 비슷한 값을 가진 위치를 핑거프린트 맵에서 추출하여 사용자의 위치를 추정한다.
  - 상기와 같은 핑거프린트 방식은 다른 추가 장비가 필요없이 실내에서 5m 이내의 오차로 위치를 추적할 수 있다.
  - 본 과제에서는 주어진 영역을  $6 * 4$ 의 cell-block으로 구분한 뒤, 각 cell-block에서 주변 AP들의 RSS 값을 듣고 기록한다. 총 N번(본 과제에서는 N을 10으로 설정)을 측정하고 각 AP의 RSS 값 중 median을 이용해 핑거프린트 맵을 구성한다.
  - 유클리드 거리를 기준으로 사용자의 RSS 값과 가장 유사한 값을 가지는 cell-block을 고르고 그 중심에 있다고 예측한다.

---

### 3.1.1.2. K-NN 알고리즘

- 사용자의 실시간 위치 추정 시, 서버에서 사용자의 위치 결정을 위한 알고리즘으로 사용한다.
- K-NN(K-Nearest Neighbor) 알고리즘은 지도 학습의 한 종류로, 거리 기반 분류(Classification) 알고리즘이며 비슷한 특성을 가진 데이터는 비슷한 범주에 속하는 경향이 있다는 가정 하에 사용한다.

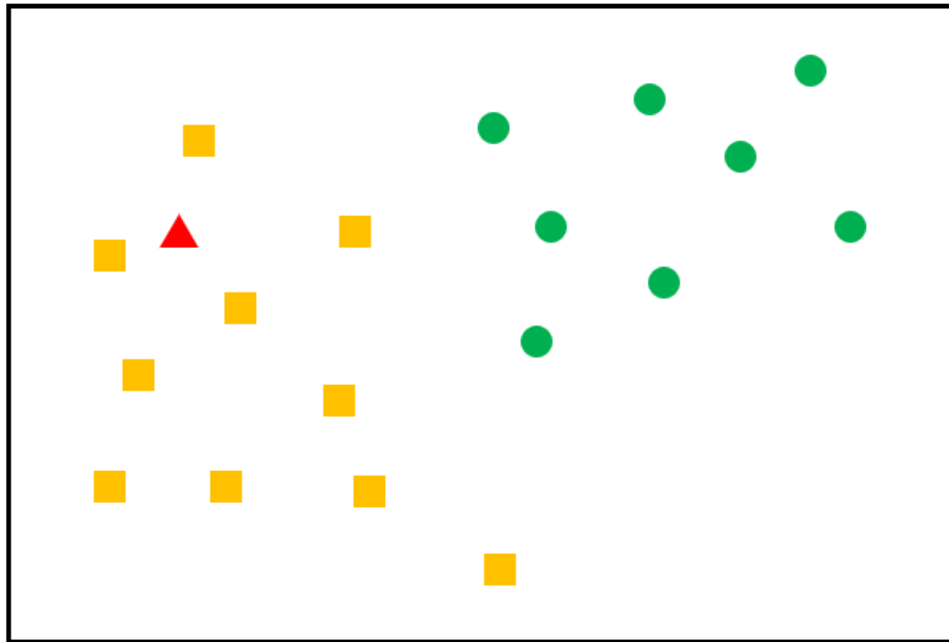


그림 3-1. K-NN 알고리즘 ①

- 예를 들어, 위와 같이 데이터가 주어져 있을 때, 빨간색인 세모 모양의 데이터는 초록색 그룹과 노란색 그룹 중 노란색 그룹에 속할 것이라고 추측할 수 있다. 이처럼 주변의 가장 가까운 K개의 데이터를 보고 데이터가 속할 그룹을 판단하는 알고리즘이 K-NN 알고리즘이다.
- 데이터 간의 거리를 측정할 때는 유클리드 거리(Euclidean distance) 계산법을 사용한다.

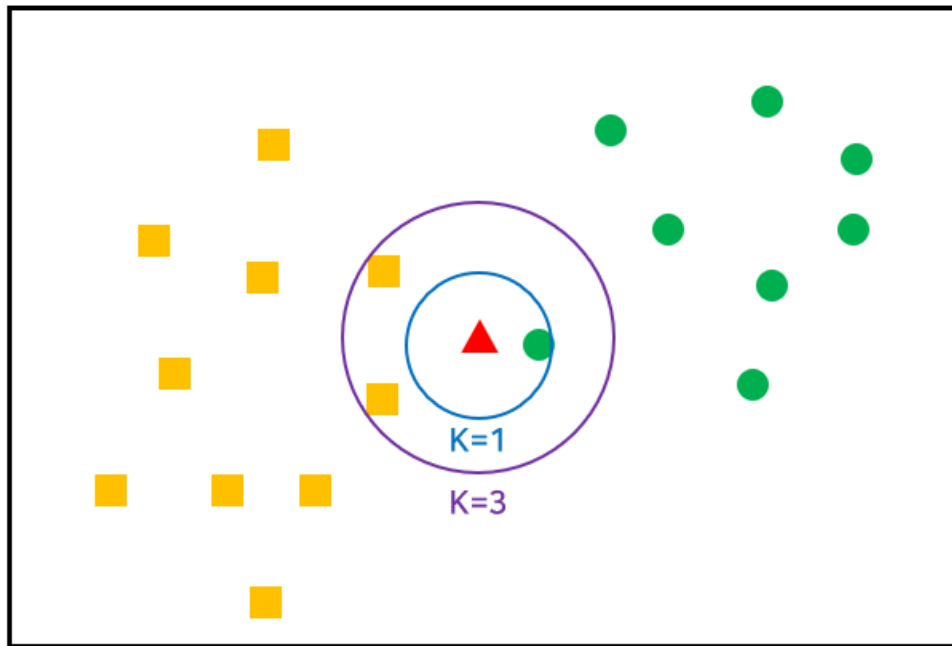


그림 3-2. K-NN 알고리즘 ②

- K-NN 알고리즘의 특징 중 하나는 K의 값에 따라 분류가 달라질 수 있다는 점이다. 예를 들어, 위와 같은 상황을 보면 K=1인 경우에는 빨간 삼각형을 초록색 그룹이라고 판단하고, K=3인 경우에는 노란색 그룹이라고 판단하게 된다.
- 항상 분류가 가능하도록 K는 홀수로 설정하는 것이 좋으며, 일반적으로 총 데이터 수의 제곱근 값을 사용한다.
- 본 과제에서는 클라이언트가 측정한 주변 AP들과의 RSS 값을 좌표별 기준 값과의 비교하여 위치를 추정하는데, 이때 가장 가까운 기준 값을 찾는 방식으로 K-NN 알고리즘이 사용된다.



### 3.1.2. 비전 기반 위치 추정 및 결과 영상 스트리밍

#### 3.1.2.1. YOLO

- 비전 기반 위치 추정을 위해 이미지 또는 비디오에서 특정 개체를 식별하는 알고리즘 모델인 YOLO를 사용했다.
- YOLO(You Only Look Once)는 대표적인 단일 단계 방식의 객체 탐지 알고리즘이다. 이미지 내에 존재하는 객체와 해당 객체의 위치는 이미지를 한 번만 보고 예측할 수 있다는 의미를 지닌다.

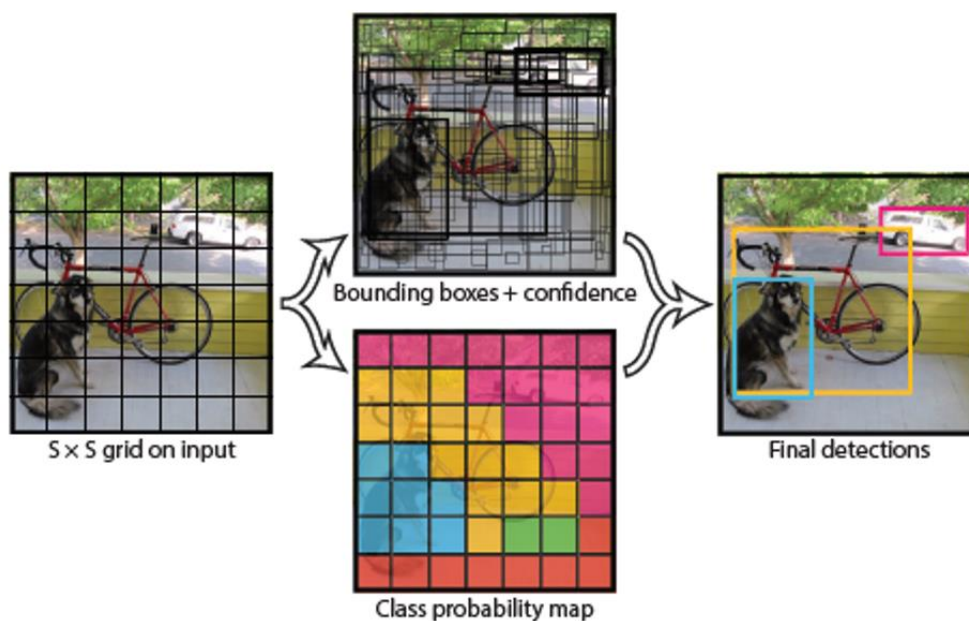


그림 3-3. YOLO의 동작 원리

- 알고리즘은 원본 이미지를 동일한 크기의 그리드(grid)로 나눈다. 각 그리드에 대해 그리드 중앙을 중심으로 미리 정의된 형태로 지정된 경계박스의 개수를 예측하고 이를 기반으로 신뢰도를 계산한다. 이미지에 객체가 포함되어 있는지, 또는 배경만 단독으로 있는지에 대한 여부가 포함되며 높은 객체 신뢰도를 가진 위치를 선택해 객체 카테고리를 파악한다.
- 본 과제에서는 건물 내 체류하는 인원의 수와 그 위치를 파악하기 위해 YOLO를 사용한다.

---

### 3.1.3. 공유 메모리

- YOLO를 이용한 객체 인식 결과를 실시간으로 스트리밍할 수 있는 웹 서버와 영상 프레임 데이터를 즉시 공유하기 위해 공유 메모리를 사용했다.
  - 공유 메모리는 IPC(Inter-Process Communication)의 한 가지 방법으로 프로세스 간 통신이 필요할 경우 사용한다. 보통 프로세스에서 사용되는 메모리 영역은 해당 프로세스만 접근 가능하다. 하지만 공유 메모리를 이용하면 둘 이상의 프로세스가 공통 메모리에 액세스 할 수 있다.
  - 한번 만들어진 공유메모리는 운영체제를 재시작 하거나 직접 공유메모리 공간을 삭제하지 않는 한 계속적으로 유지된다. 공유메모리를 사용하는 모든 프로세스가 없어졌다고 하더라도 제거되지 않으므로 주의해야 한다.
  - 본 과제에서는 YOLO의 객체 인식이 완료된 프레임 데이터를 공유 메모리에 저장하면 스트리밍용 웹 서버가 이 데이터를 읽어와 화면에 출력한다.

### 3.1.4. 웹 서비스

#### 3.1.4.1. SSE

- 밀집도가 일정 정도 이상 올라가거나 자신을 제외한 서비스 사용자가 제보 작성 시 사용자에게 알림을 제공하는 기능을 구현하기 위해 웹 서버와 클라이언트 간 SSE 통신을 사용했다.
  - SSE(Server-Sent-Events)는 단방향 통신 방식으로 서버에서 클라이언트로 실시간 이벤트를 전송하는 웹기술이다. 이를 통해 서버에서 발생하는 업데이트나 알림 등을 실시간으로 클라이언트에게 전달할 수 있다. SSE는 이런 특징으로 실시간 알림, 실시간 주가 업데이트 등에 사용된다.
  - SSE는 별도의 프로토콜을 사용하지 않고 http 프로토콜만으로 사용할 수 있기 때문에 구현이 용이하다.

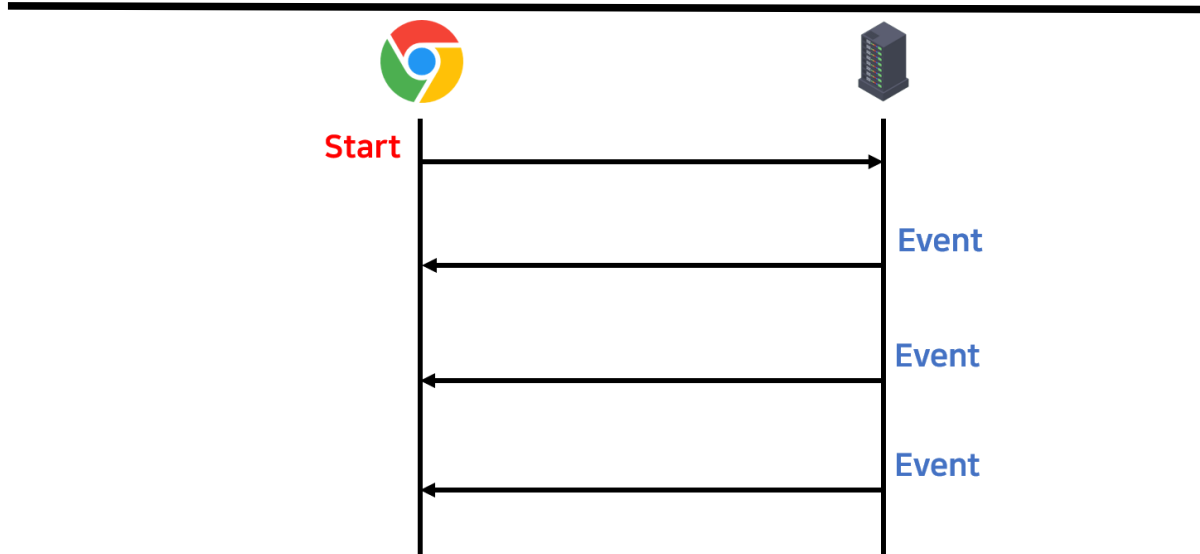


그림 3-4. SSE

- 본 과제에서는 알림 서비스 구현을 위해 서비스 사용자가 로그인을 하면 자동으로 SSE 통신을 연결하고, 실시간으로 변동사항에 대한 대응이 가능하도록 설계했다.

#### 3.1.4.2. 인구 밀집도

- 밀집도 등급을 결정하기 위해 일반적으로 사용되는 인구 밀집도의 정의와 밀집도 기준을 참고하였다.
  - 학계에서는 군중 밀집으로 인한 사고 위험성을 예측할 때, 1m<sup>2</sup> 당 몇 명이 있는지를 기준으로 삼는다. 평방미터로도 불리는 m<sup>2</sup>(제곱미터)는 기본 면적 단위로써, 한 변의 길이가 1m인 정사각형의 면적을 뜻한다.
  - '군중 행동 전문가' 키스 스틸 영국 서포크대 초빙교수는 논문 및 언론 인터뷰에서 1m<sup>2</sup>당 5명이 있을 경우 군중 사이에 접촉이 많아지면서 조금만 떠밀어도 문제가 될 수 있다고 분석했으며 6명에 이를 경우 사람들은 '신체 통제력'을 잃게 된다고 한다.
  - 위를 기준으로 본 과제에서는 인구 밀집도 등급을 아래와 같이 나누었다.
    - 안전: 기준면적 당 1-2명이 있을 경우
    - 주의: 기준면적당 3-4명이 있을 경우
    - 위험: 기준면적당 5-7명이 있을 경우
    - 고위험: 기준면적당 8명 이상이 있을 경우

- 실제 인구 밀집도의 면적 기준은 1m<sup>2</sup>이지만 본 과제에서는 여러 밀집도를 보여 주기 위해 기준 면적을 부산대학교 제6공학관 2층 평면도 상의 격자의 한 칸인 100m<sup>2</sup>으로 설정했다.

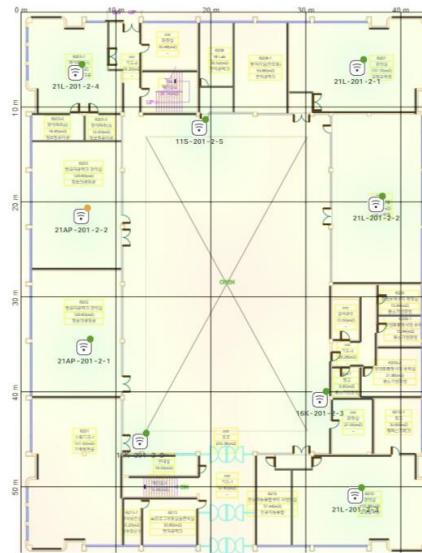


그림 3-5. 밀집도 측정 기준인 부산대학교 제6공학관 2층 평면도

### 3.1.5. 반응형 웹

- 사용자가 이동하며 사용하는 서비스이고, 장소에 구애받지 않고 사용이 용이해야 하므로 휴대폰, 태블릿, 노트북 등의 다양한 매체에서의 접근을 고려하여 반응형 웹으로 서비스를 제작했다.
  - 반응형 웹은 디바이스의 디스플레이의 종류에 반응하여 그에 맞도록 적절하게 UI 요소들이 유기적으로 배치되도록 설계된 웹이다.
  - 그리드 시스템(Grid System)
    - 그리드 시스템은 Bootstrap이 제공하는 layout과 content 정렬을 위한 그래픽 디자인 시스템이다.
    - 전체 화면을 격자로 나누고 그래픽 요소에 원하는 크기를 격자 단위로 할당하면 절대적인 크기와 상관없이 화면에 비례하여 요소들을 나타낼 수 있다.
    - 그리드 시스템은 플랫폼, 환경 및 화면 크기에 일관성을 부여하고 그래픽 요소의 체계적인 배열을 도와주어 접속 매체와 무관하게 일관된 UI를 얻을 수 있다.

- 
- 본 과제에서는 표에 반응형 웹을 적용하기 위해 그리드 시스템을 사용했다.
  - 미디어 쿼리(Media Query)
    - 미디어 쿼리는 디바이스의 폭이나 높이 등의 정보를 이용해 특정 화면 크기에서의 스타일을 개별적으로 설정할 수 있다.
    - 너비 및 높이는 min-이나 max-으로 최대 혹은 최소를 지정할 수 있다.
    - 본 과제에서는 screen의 너비를 고려하여 반응형 웹을 적용하기 위해 미디어 쿼리를 사용했다. 스타일 변경의 기준 너비는 Bootstrap에서 제공하는 그리드 시스템을 기준으로 한다. (576px 미만, 576px ~ 767px, 768px ~ 991px, 992px ~ 1199px, 1200px 이상)

## 3.2. 기술 상세

### 3.2.1. 무선 신호 기반 실시간 위치 추정

- 측정하고 싶은 전체 공간을 그리드로 나눈다. 측정을 원하는 cell-block마다 주변의 모든 Access Point들과의 RSS 값을 N회에 걸쳐 측정하여 저장한다. (본 과제에서는 N을 20으로 가정) 이를 서버 측에 전송하면 모든 데이터를 좌표별로 구분한 후, 각 AP마다 RSS 값의 median을 계산하여 radio map을 생성한다.
- 이후 라즈베리파이에서 실시간으로 RSS 값을 측정하고, 그 결과를 서버로 전송한다. 서버에서는 데이터를 이용해 현재 상태를 나타내는 radio map을 생성하고, K-NN 알고리즘에 따라 거리가 가장 가까운 cell-block을 찾는다.
- 하나의 서버가 여러 라즈베리파이와 통신해야 하므로 파이썬의 select 모듈을 사용하여 이를 구현한다. 스레드를 사용해서 해결할 수도 있지만, 스레드 방식은 스레드 수가 늘어날 수록 시스템에 부담이므로 효율적이지 못하다. select 방식은 처음에는 블로킹되어 있다가 특정 이벤트가 발생하면 그때 작동하는 방식이므로 스레드보다 효율적이다.
- 라즈베리파이의 위치 추정 결과를 http 통신을 이용해 웹 서버로 전송한다. 하나의 라즈베리파이 당 최대 1000개의 기록을 데이터베이스에 저장해 사용자의 실시간 위치 및 밀집도 계산, 위치/시간 별 사용자 검색 기능에 활용한다.

```

Address: ('172.30.1.41', 46140)
BEST: cell blocks (y,x) : [[0, 0]]
BEST: distances : [204]

Address: ('172.30.1.24', 51622)
BEST: cell blocks (y,x) : [[1, 0]]
BEST: distances : [228]

Address: ('172.30.1.41', 46140)
BEST: cell blocks (y,x) : [[0, 0]]
BEST: distances : [189]

Address: ('172.30.1.24', 51622)
BEST: cell blocks (y,x) : [[1, 0]]
BEST: distances : [196]

Address: ('172.30.1.41', 46140)
BEST: cell blocks (y,x) : [[0, 0]]
BEST: distances : [203]

Address: ('172.30.1.24', 51622)
BEST: cell blocks (y,x) : [[1, 0]]
BEST: distances : [232]

```

그림 3-6. 다수의 클라이언트(라즈베리파이)의 위치 추정 결과

### 3.2.2. 비전 기반 실시간 사람 인식

- 기본적으로 YOLOv5는 COCO dataset이 기본적으로 Pre-Trained 되어있는 모델을 제공하며 약 80개의 객체가 인식 가능하다. 이 과제에서는 기본 모델을 이용하여 객체를 인식한다.
- 객체 인식을 시작하기 위해서는 영상, 사진, 웹캠 등 소스 선택 및 다양한 옵션을 설정하여 터미널에서 detect.py라는 기본 제공 객체 탐지 스크립트를 실행하는 것이 일반적인 동작법이다.
- 하지만 본 과제에서는 인식 객체를 사람으로 한정하고 인식 결과를 저장하지 않는 등 모든 옵션들을 고정하기 위해 파라미터의 기본 값을 미리 설정한다.
- 일반적인 랩탑이나 데스크탑에 비해 저 성능인 라즈베리파이를 기반으로 동작하여 속도 저하 이슈를 고려해 본 과제에서는 상대적으로 성능이 낮지만 FPS가 가장 높은 경량화 모델인 YOLOv5n을 사용한다.

### 3.2.3. 카메라 인식 위치 보정

- YOLO는 인식된 객체의 위치를 알려주기 위해 객체의 둘레를 감싼 직사각형 사각

형 테두리로 위치를 표현하는데, 이를 Bounding box라고 한다. YOLO는 Bounding box의 정 중앙 좌표와 길이, 너비 정보를 제공해준다. 본 과제에서는 해당 정보를 통해 Bounding box의 하단 변 정 중앙 좌표를 계산한 후, 원근 변환 행렬을 이용해 실제 위치를 도출한다.

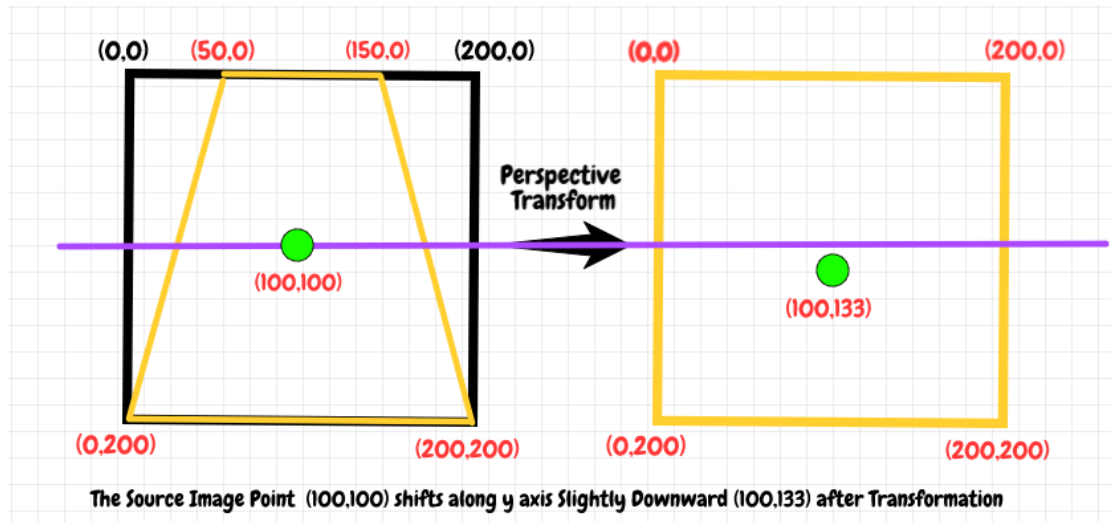


그림 3-7. 카메라 인식 위치 보정 원리

- 원근 변환 행렬을 구하기 위해서는 카메라에 나타나는 바닥, 즉 변환의 기준이 되는 객체의 네 꼭지점 좌표와 그에 대응하는 변환 후 평면의 네 꼭지점 좌표를 알아야 한다. 원본 이미지에서 평면으로 원근 변환을 하는 3x3의 행렬을 구한다. 인식 객체의 좌표를 행렬로 표현 시 차원이 2x1이므로 행렬의 길이가 서로 다르다. 그러므로 이를 맞추기 위해 객체 좌표 행렬에 1을 추가하여 3x1의 차원을 가지도록 한다. 그 후, 행렬 곱을 통해 wx, wy값을 구하고 각각을 w로 나누어 주어 x, y의 값을 구한다.
- 파이썬 OpenCV 라이브러리를 사용하여 구현했다.
- 계산된 객체의 위치를 http 통신을 통해 웹 서버에 전달한 후, "비전 기반 밀집도" 페이지에서 건물 단면도 위에 실시간으로 출력한다.

#### 3.2.4. 실시간 영상 스트리밍

- YOLO를 통한 객체 인식이 이루어지는 라즈베리파이에서 flask 라이브러리를 이용하여 스트리밍용 웹 서버를 구축하고 5000번 포트에 배포한다. 또한 외부에서의 접속을 고려하여 고정IP 할당 및 공인IP를 사용하는 지정 공유기의 포트 포워딩을 설정한다.

- 
- YOLO가 인식한 결과를 포함한 프레임 데이터를 바이트 형식으로 공유 메모리에 저장하고 이를 웹 서버에서 읽어와 화면에 출력하는 방식으로 동작한다.

### 3.2.5. 밀집도 예측

- 인원 수 및 밀집도 예측 기법은 이동평균법을 이용한다. 이동평균법은 표본평균처럼 관측값 전체에 동일한 가중값을 주는 대신, 최근  $n$ 개의 관측치를 이용하여 평균을 구하고 이를 이용해 예측을 하는 기법이다. 이동평균법은 평균의 계산 기간을 순차로 한 개의 항씩 이동시켜가며 기간별 평균을 계산하여 경향치를 구하는 방법으로서, 가장 오래된 데이터는 제거하고 가장 최초의 데이터로부터 평균을 구한다. 단순이동평균, 누적이동평균, 가중이동평균이 있다.
- 단순이동평균법은 이전  $n$ 개 데이터의 비가중 평균이다. 본 과제에서는 단순이동평균을 이용하여 인원 수를 예측했다. 전체 인원 수 데이터 중 최근 3개의 데이터의 평균을 내어 인원 수를 예측한다. 또한 예측된 인원 수를 기준으로 밀집도를 예측하였다.



## 4. 과제 결과 분석 및 평가

### 4.1. 메인 페이지



그림 4-1. 비회원 메인페이지



그림 4-2. 회원 메인페이지



그림 4-3. 관리자 메인페이지

- 권한에 따라 사용 가능한 기능의 차별을 두었으며, 이를 스프링 시큐리티를 이용해 통제한다.
  - 비회원이 회원만 사용 가능한 기능에 접근하면 로그인 페이지로 이동한다.
  - 회원 및 비회원이 관리자만 사용 가능한 기능에 접근하면 오류 페이지로 리다이렉트한다.
- 타임리프를 이용해 권한 별로 헤더를 다르게 구성했다. 각 화면은 순서대로 비회원, 회원, 관리자의 메인 페이지이다.
  - 비회원으로 접속 시, 메뉴바는 회원과 동일하게 출력되나 이웃 및 알림 아이콘이 보이지 않는다.
  - 회원으로 접속 시, 메뉴바의 찾기 버튼과 회원 관리 아이콘은 보이지 않는다.
  - 관리자로 로그인 했을 시, 메뉴바의 이웃 아이콘은 보이지 않는다.
- 본 과제의 주요 기능인 현재 위치 및 밀집도 등급 확인, 제보를 통한 사용자 참여, 이웃 추가 기능을 통한 위치 확인을 메인 페이지에 나타냄으로써 각 기능에 빠르게 접근할 수 있도록 하였다.
  - 위치 및 밀집도 등급 확인을 누르면 현재 위치 및 밀집도 페이지로 이동한다.
  - 제보를 통한 사용자 참여를 누르면 제보하기 페이지로 이동한다.

- 이웃 추가 기능을 통한 위치 확인을 누르면 이웃 목록 페이지로 이동한다.
- 본 과제의 핵심 기능인 위치 확인 및 밀집도 등급 확인 및 경고를 간단한 gif로 나타내 처음 사이트를 접한 사람도 쉽게 서비스의 목적을 이해할 수 있도록 했다.

## 4.2. 로그인 페이지

### 인파 밀집도 계산 및 실시간 위치 추적 웹서비스

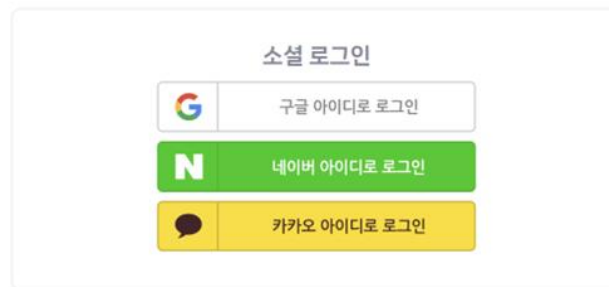


그림 4-4. 로그인 페이지

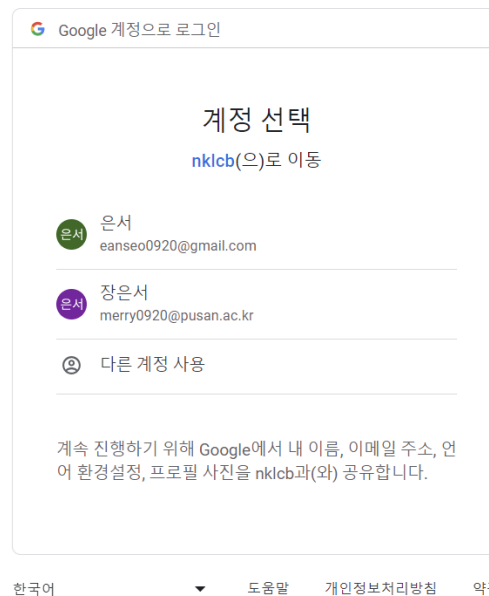


그림 4-5. 구글 로그인



그림 4-6. 구글 로그인 후 헤더

- 소셜 로그인을 통해 회원가입/로그인 과정을 단순화하여 사용성과 접근성을 높이 고자 했다.
- 스프링 시큐리티와 OAuth2.0 프로토콜을 이용하여 구글, 네이버 등 다양한 플랫폼 을 통한 소셜 로그인을 구현했다.
- 로그인한 리소스 서버의 닉네임과 프로필 사진을 가져와 헤더에 나타낸다.

### 4.3. 현재 위치 및 밀집도



그림 4-7. 현재 위치에서의 밀집도 페이지

- 사용자의 라즈베리파이 기반 위치를 보여주는 페이지이다. 사용자가 즉각적으로 현재 위치와 밀집도를 알 수 있도록 평면도 상에 현재 위치를 표시하고 밀집도 등급의 색상으로 표시한다.
- [무선 신호 기반 실시간 위치 추정](#) 기술을 통해 개발되었다.
- 현재 시간을 기준으로 1분 이내 같은 위치에 존재하는 사용자의 정보를 데이터베이스에서 가져와 밀집도 등급을 결정하고 현재 위치에 함께 있는 사용자의 목록을 보여준다.

- 1초에 한 번 데이터를 자동으로 업데이트해 실시간성을 유지한다.

#### 4.4. 무선 통신 기반 밀집도



그림 4-8. 무선 통신 기반 밀집도 페이지



그림 4-9. 평면도의 특정 구역을 선택했을 때의 무선 통신 기반 밀집도 페이지



그림 4-10. 밀집도 등급

- 라즈베리파이로 측정된 위치를 기반으로 각 구역의 인원 수 및 밀집도를 보여주는 페이지이다.
- 평면도 위의 각 구역에 마우스를 올리면 해당 구역의 실시간 밀집도가 색으로 표시되며 위치의 좌표와 실시간 인원 수를 보여준다.
- '무선 통신 기반 건물 전체의 밀집도' 옆의 물음표 버튼을 누르면 본 과제에서 가정한 밀집도 등급을 보여준다.
- [무선 신호 기반 실시간 위치 추정](#) 기술을 통해 개발되었다.
- 현재 시간을 기준으로 1분 이내 모든 위치 데이터를 데이터베이스에서 가져와 인원 수를 측정하고 밀집도 등급을 결정한다.
- 이 과정에서 중복적으로 사용자가 카운팅되지 않도록 Set 자료구조를 이용해 사용자 정보를 가져온다.
- 1초에 한 번 데이터를 자동으로 업데이트해 실시간성을 유지한다.

#### 4.5. 밀집도 경고 알림



그림 4-11. 밀집도 경고 알림창

- 사용자의 현재 위치의 밀집도 등급이 위험 이상으로 변경되었을 경우 경고창이 뜨게 된다.
- 밀집도 경고 알림은 [SSE](#) 및 SseEmitter 클래스를 통해 구현되었다.
- 사용자가 로그인을 하면 프론트 측에서 서버의 이벤트 구독을 위한 요청을 보낸다. 이후 사용자의 현재 위치 정보와 인원 수를 모니터링하며 밀집도 등급이 위험 이상으로 변경되면 해당 사용자에게 이벤트가 발생했다는 데이터를 전송한다.

#### 4.6. 비전 기반 실시간 위치 추정



그림 4-12. 비전 기반 실시간 위치 추정 페이지

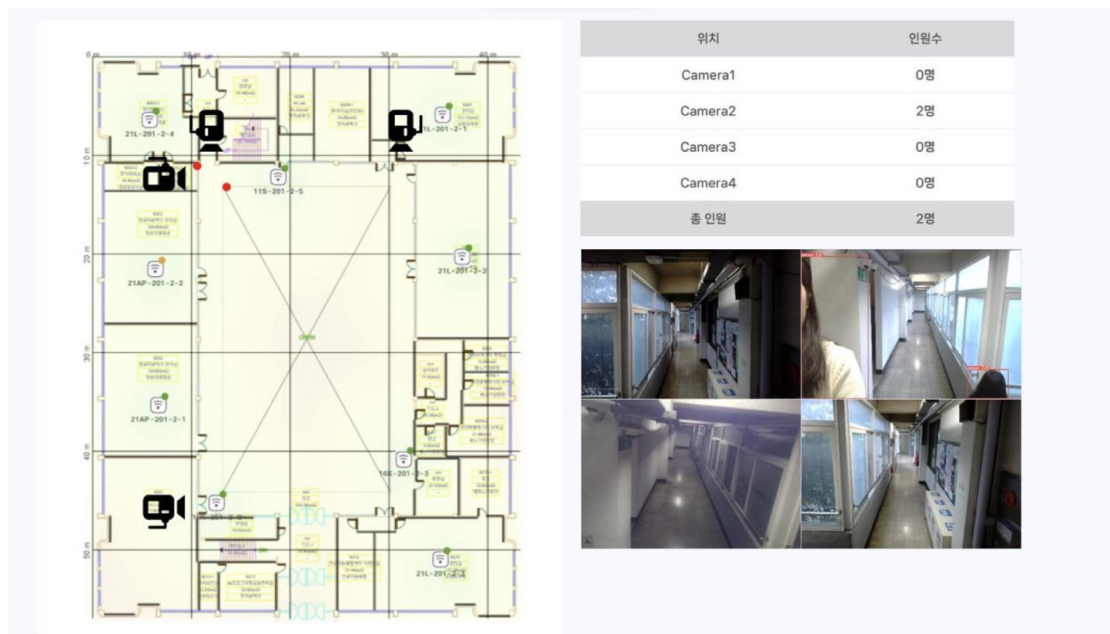


그림 4-13. 건물 내 카메라 전체 화면을 클릭했을 때의 비전 기반 실시간 위치 추정 페이지

- 비전 기반의 인원 수 정보 및 객체 인식 결과 영상 스트리밍 기능을 제공하는 페이지이다.
- 평면도 위에는 실시간으로 인식된 사람들의 위치가 빨간색 점으로 표시된다.
- 설치된 카메라에 인식된 인원 수를 표를 통해 나타낸다.



- 설치된 카메라의 위치를 도면도 위 아이콘으로 표시하여 보여주며 카메라 아이콘을 클릭하면 해당 위치의 카메라 화면을 이용한 객체 인식 결과 영상이 나타난다.
- '건물 내 카메라 전체 화면'을 클릭하면 4분할로 모든 카메라의 객체 인식 결과 영상을 볼 수 있다.
- [비전 기반 실시간 사람 인식](#) 기술과 [카메라 인식 위치 보정](#), [실시간 영상 스트리밍](#) 기술을 통해 구현되었다.

#### 4.7. 실시간 밀집도 통계 및 확인

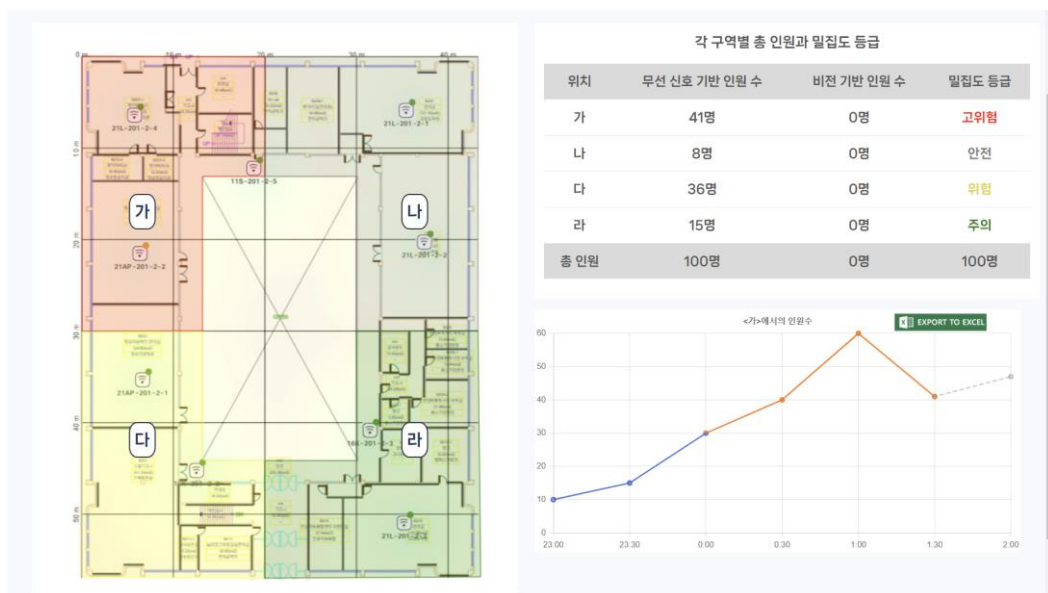


그림 4-14. 실시간 밀집도 통계 및 확인 페이지

	A	B	C	D
1	무선 신호 기반 인원 수	비전 기반 인원 수	총 인원 수	생성 시간
2	1	2	3	2023-10-12T00:10:49.223495
3	1	2	3	2023-10-12T00:10:49.248490
4	1	2	3	2023-10-12T00:10:49.249490
5	1	2	3	2023-10-12T00:10:49.250492
6	1	2	3	2023-10-12T00:10:49.251499
7	1	2	3	2023-10-12T00:10:49.252492
8	1	2	3	2023-10-12T00:13:11.153451
9	1	2	3	2023-10-12T00:13:11.155472
10	1	2	3	2023-10-12T00:13:11.157677
11	1	2	3	2023-10-12T00:13:11.158475
12	1	2	3	2023-10-12T00:13:11.159474
13	1	2	3	2023-10-12T00:13:11.160493
14	1	2	3	2023-10-12T00:34:04.892784
15	1	2	3	2023-10-12T00:34:04.895778
16	1	2	3	2023-10-12T00:34:04.900785
17	1	2	3	2023-10-12T00:34:04.905798
18	1	2	3	2023-10-12T00:34:04.910807

4-15. 인원 수 정보 엑셀 파일

- 
- 실내 공간을 4분할하여 전반적인 실내 밀집도 및 통계 정보를 제공하는 페이지이다.
  - 무선 신호 기반 인원 수와 비전 기반 인원수를 통합한 밀집도 등급을 도면도에 표시한다. 각 구역을 클릭하면 차트에 해당 구역의 시간별 인원수 변화가 나타난다.
  - 시간별 인원수가 밀집도 위험 등급 이상의 수라면 차트에 주황색으로 표시한다.
  - 30분 간격마다 인원수가 차트에 표시되며 예측한 다음 인원수를 회색 점선으로 나타낸다.
  - [무선 신호 기반 실시간 위치 추정](#)과 [비전 기반 실시간 사람 인식](#), [밀집도 예측](#) 기술을 통해 개발되었다.
  - static 변수를 이용하여 정각과 30분마다 자동으로 구역별 인원 수를 측정하는 기능을 구현하였다.
  - 카메라로 측정된 인원의 위치 데이터는 구역이 아닌 평면도 상의 좌표로 저장되는데, 이를 구역 별로 구분하기 위해 평면도 상 4분할의 기준이 되는 위치의 좌표 값을 기준으로 인식 객체의 좌표와 비교한다.
  - 시간별 인원 수 그래프의 EXPORT TO EXCEL 이미지를 클릭하면 현재까지 데이터베이스에 저장된 모든 인원 수 정보를 엑셀 파일로 제공한다.
  - 엑셀 파일 생성을 위해 Apache POI 라이브러리를 사용했다.

## 4.8. 제보

### 제보하기

위급한 상황을 제보해주세요.

위치	위치를 선택하세요 <span>위치 선택</span>
제목	제목을 입력하세요
내용	내용을 입력하세요

제출

그림 4-16. 제보하기 페이지

### 제보하기

위급한 상황을 제보해주세요.

위치	위치를 선택하세요 <span>위치 선택</span>
	

그림 4-17. 위치 선택을 눌렀을 때의 제보하기 페이지

- 사용자가 위급한 상황을 제보하기 위한 페이지이다.
- 위치 선택 버튼을 눌렀을 때 나타나는 평면도 상의 원하는 구역을 선택하면 위치 선택이 완료된다.

The screenshot shows a web form titled "제보하기" (Report) with the subtitle "위급한 상황을 제보해주세요." (Please report an emergency situation). The form has three main sections: "위치" (Location), "제목" (Subject), and "내용" (Content). The "위치" section has a text input field with the placeholder "위치를 선택하세요" (Select location) and a "위치 선택" (Select location) button. The "제목" section has a text input field with the placeholder "제목을 입력하세요" (Enter subject). The "내용" section is a large text area. A modal dialog box is displayed in the center, with the text "위치를 선택해주세요" (Please select location) and an "OK" button. At the bottom of the form is a "제출" (Submit) button.

그림 4-18. 위치 미선택 경고창

The screenshot shows the same "제보하기" (Report) form. In this state, the "위치" section has a dropdown menu open showing the option "A1", and the "위치 선택" button is active. The "제목" section has a text input field with the placeholder "제목을 입력하세요" (Enter subject). A modal dialog box is displayed in the center, with the text "제목을 입력해주세요" (Please enter subject) and an "OK" button. The "내용" section and the "제출" (Submit) button at the bottom are also visible.

그림 4-19. 제목 미입력 경고창

- 위치 선택을 하지 않았거나 제목을 입력하지 않았을 시에는 경고창을 띄우며 제출이 되지 않는다.

## 제보하기

위급한 상황을 제보해주세요.

위치	A1 <span style="float: right; background-color: #555; color: white; padding: 2px 5px; border-radius: 3px;">위치 선택</span>
제목	도움이 필요합니다.
내용	A1 구역에 사람이 많습니다.

제출

그림 4-20. 제보하기 글 작성

## 제보 게시판

#	위치	제목	작성자	작성일자
3	A1	도움이 필요합니다.	은서	2023-10-19 05:34
2	C3	도와주세요!	아무개1	2023-10-19 04:03
1	B0	확인 부탁드립니다.	아무개0	2023-10-19 04:03

이전
1
다음

그림 4-21. 제보 게시판

제보 게시판			
위치	A1	작성자	은서
제목	도움이 필요합니다.		
내용	A1구역에 사람이 많습니다.		

4-22. 제보 게시판 게시글

- 글을 작성한 뒤 제출 버튼을 누르면 제보 게시판으로 넘어간다.
- Pageable 객체를 정의해 쿼리 실행 시 적용하여 제보게시판의 게시물 출력 시 페이징 처리를 구현했다.
- 게시물 리스트의 템플릿 레이아웃을 개별적으로 정의하고 타임리프를 통해 동적으로 내용을 채운다. 이후 게시물 조회 요청 시 템플릿을 리턴하여 해당 템플릿이 제보 게시판의 게시물 영역을 대체하도록 한다.
- 게시물의 제목에 하이퍼링크를 정의해 제목을 누르면 해당 게시글 상세 내용 조회가 가능하다.



그림 4-23. 새로운 제보 등록 알림

- 제보 게시판에 새로운 제보가 등록되었을 경우 다른 사용자들에게 알림이 가게 된다. 제목과 내용의 일부가 보인다.
- 제보 알림은 [SSE](#) 및 SseEmitter 클래스를 통해 구현되었다.
- 사용자가 로그인을 하면 프론트 측에서 서버의 이벤트 구독을 위한 요청을 보낸다. 이후 제보가 작성되면 서버에서 작성자를 제외한 모든 사용자에게 이벤트가 발생했다는 데이터를 전송한다.

#### 4.9. 사용자 찾기



그림 4-24. 사용자 찾기 페이지



그림 4-25. 사용자 검색

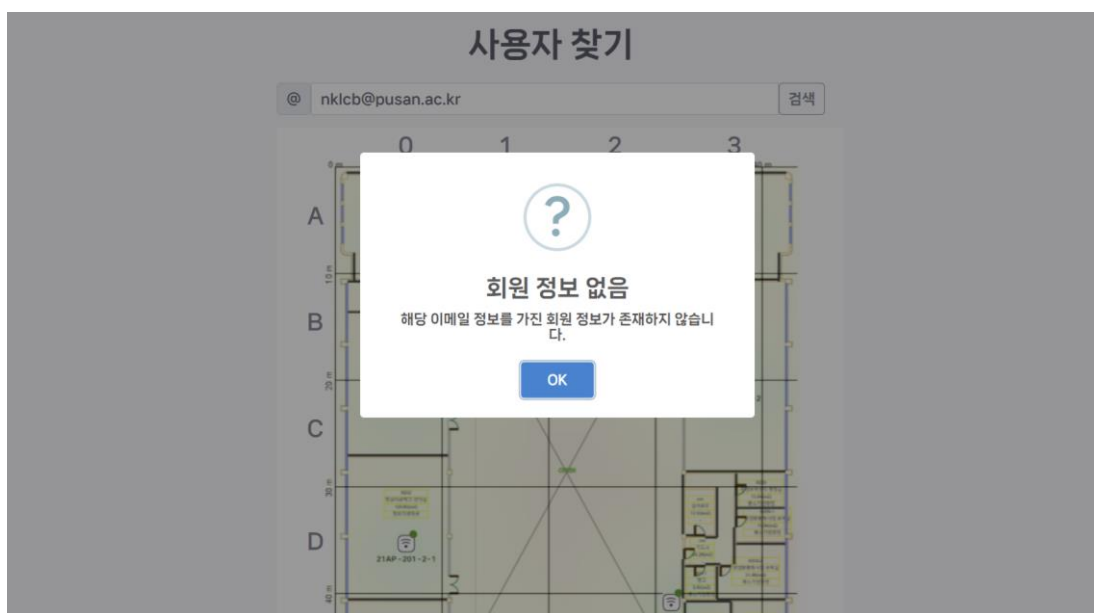


그림 4-26. 회원 정보 없음 알림창

- 관리자가 이메일을 검색하여 특정 사용자의 마지막 기록 위치와 기록 시간 및 밀집도 등급을 알 수 있는 페이지이다.
- [무선 신호 기반 실시간 위치 추정](#) 기술을 통해 개발되었다.
- 사용자 찾기의 밀집도 등급은 사고 당시 밀집도를 파악하기 위해서 기록 시간을 기준으로 1분 이내의 데이터를 활용한다.



- 회원가입이 되어 있지 않은 사용자의 이메일을 입력하거나 라즈베리파이를 소지하지 않은 회원을 검색할 경우 '회원 정보 없음'이라는 경고창으로 알려준다.

#### 4.10. 특정 구역 사용자 찾기



그림 4-27. 특정 구역 사용자 찾기 페이지

이메일	라즈베리파이 ID	라즈베리파이 MAC
nklcb25@pusan.ac.kr	26	2525.2525.2525
nklcb24@pusan.ac.kr	25	2424.2424.2424
nklcb23@pusan.ac.kr	24	2323.2323.2323
nklcb22@pusan.ac.kr	23	2222.2222.2222
nklcb21@pusan.ac.kr	22	2121.2121.2121
nklcb20@pusan.ac.kr	21	2020.2020.2020
nklcb19@pusan.ac.kr	20	1919.1919.1919
nklcb18@pusan.ac.kr	19	1818.1818.1818
nklcb17@pusan.ac.kr	18	1717.1717.1717

그림 4-28. 평면도 상의 특정 구역을 눌렀을 때 특정 구역 사용자 찾기 페이지

- 관리자가 선택한 구역에 있었던 사용자들의 목록을 확인할 수 있는 페이지이다.
- [무선 신호 기반 실시간 위치 추정](#) 기술을 통해 개발되었다.

- 현재 시간을 기준으로 최대 10분 이내의 데이터를 조회할 수 있으며, 위치 데이터 조회 과정에서 중복적으로 사용자가 카운팅되지 않도록 Set 자료구조를 이용해 사용자 정보를 가져온다.

#### 4.11. 이웃

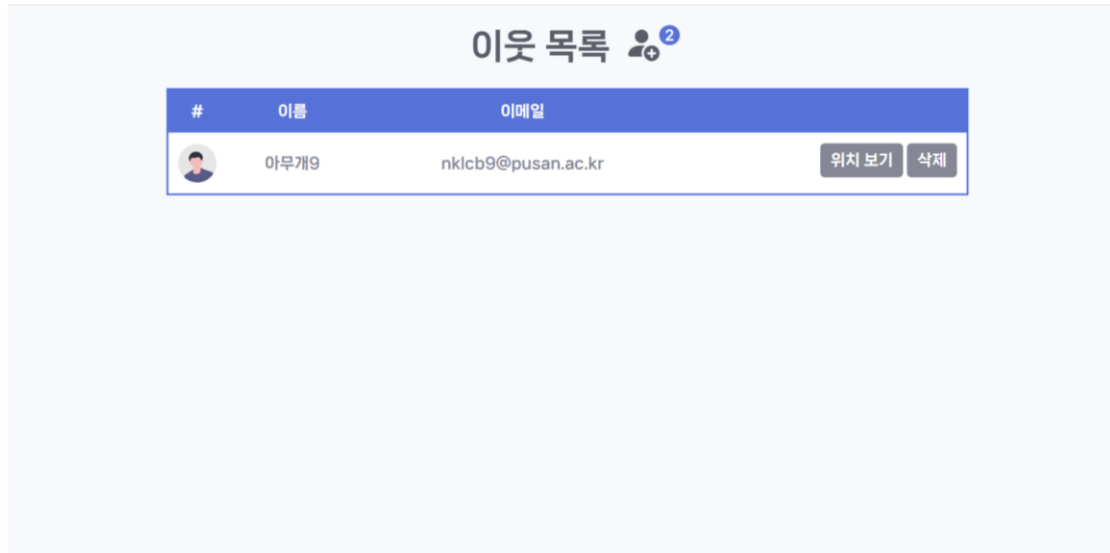


그림 4-29. 이웃 목록 페이지

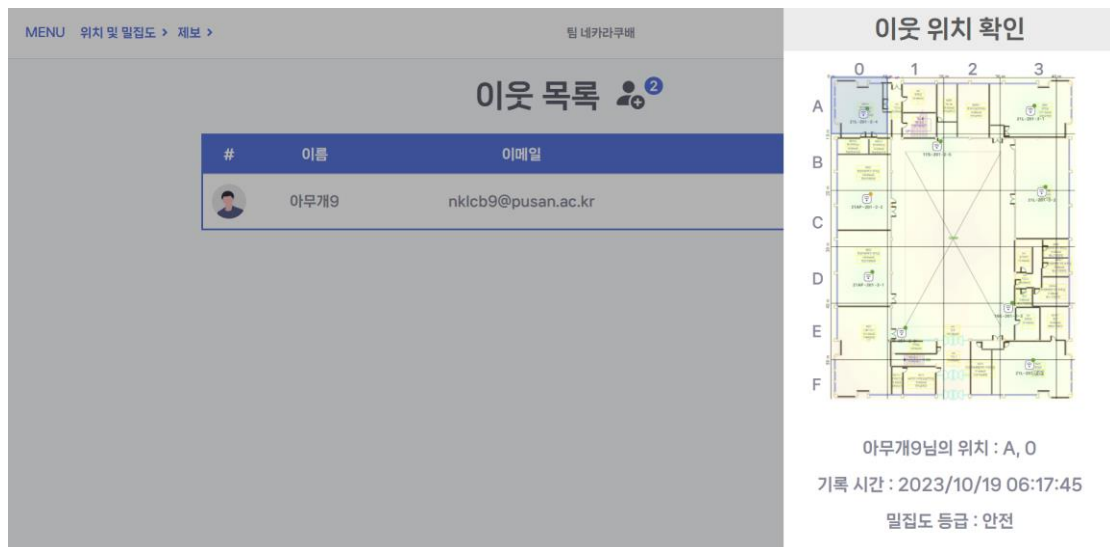


그림 4-30. 이웃 위치 확인

- 사용자로 로그인 시, 헤더의 사람 아이콘을 클릭하면 접속 가능하다.
- 이웃을 맺은 사용자들의 정보를 확인할 수 있으며, 제공되는 정보는 이웃의 프로필 이미지와 닉네임, 이메일이 있다.

- 위치 보기 버튼을 누르면 이웃의 가장 최근에 기록된 위치 정보가 표시되며, 기록 시간과 밀집도 등급을 보여준다.
- [무선 신호 기반 실시간 위치 추정](#) 기술을 통해 개발되었다.
- 이웃의 위치 보기의 밀집도 등급은 위치 정보 저장 당시의 밀집도 정보를 제공하기 위해서 기록 시간을 기준으로 1분 이내의 데이터를 활용한다.
- 이웃 위치 확인 부분을 제외한 화면을 누르면 이웃 위치 확인 부분이 슬라이딩되어 사라진다.
- 이웃 목록 옆 아이콘 클릭 시 이웃 추가 페이지로 이동하며, 이웃 신청 수가 숫자로 표시된다.

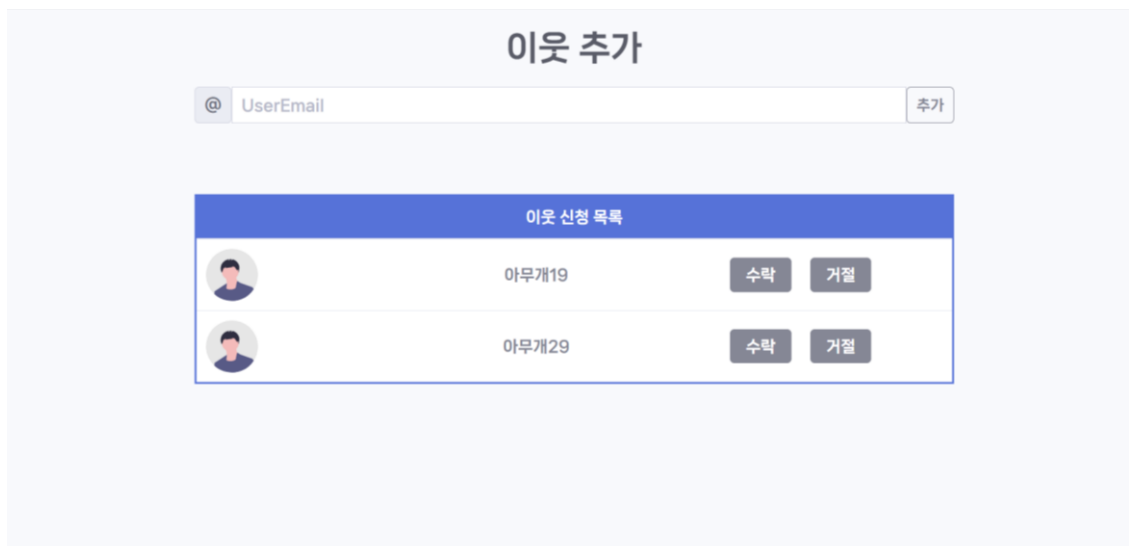


그림 4-31. 이웃 추가 페이지

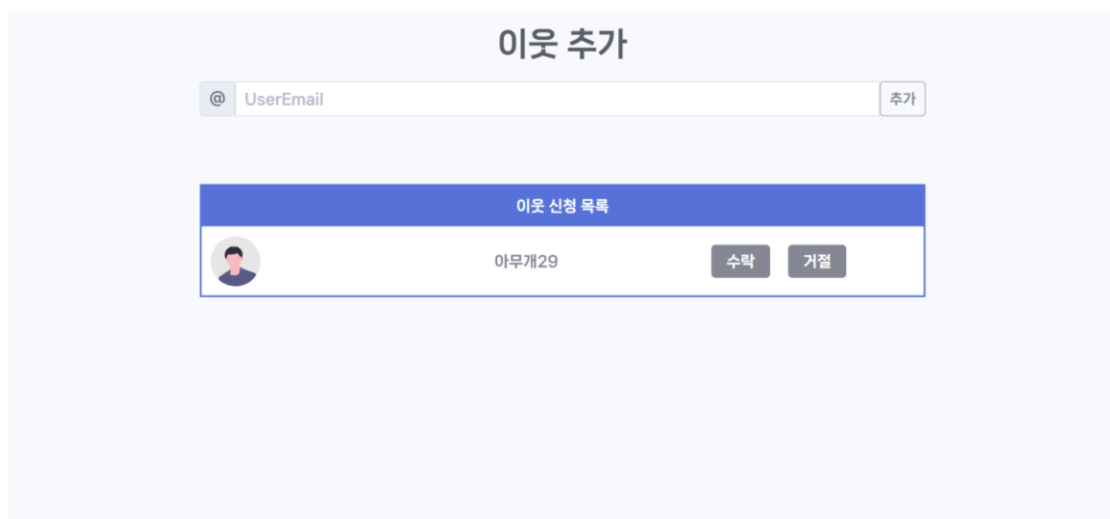


그림 4-32. 이웃 수락

- 이웃 추가 페이지에는 이웃 신청을 하고 싶은 사용자의 이메일을 입력하는 란과 이웃 신청 목록이 표시된다.
- 이웃 신청 수락 혹은 거절 버튼을 눌렀을 시, 이웃 신청 목록에서 삭제된다.
- 이웃 신청 수락을 하면 이웃 목록에 추가된다.



그림 4-33. 이웃 추가 이메일 검색

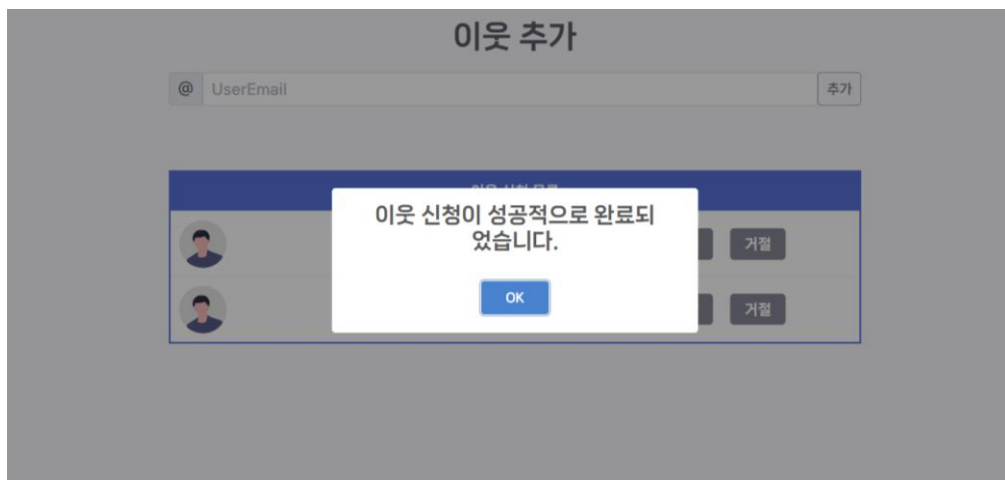


그림 4-34. 이웃 신청 완료 알림창

그림 4-35. 이웃 신청 목록 추가

- 이메일을 입력하여 추가 버튼을 누르면 '이웃 신청이 성공적으로 완료되었습니다.'라는 문구와 함께 이웃 신청이 된다.
- 이웃 신청을 한 사용자의 이웃 신청 목록에 추가된 것을 볼 수 있다.
- 만약 회원가입이 되지 않은 이메일을 입력하거나, 이미 이웃인 회원 혹은 이미 이웃 신청을 완료한 회원에게 이웃 신청을 하면 그에 맞는 오류 메시지를 출력하고 입력란이 리셋된다.
- 이웃 기능의 구현은 이웃 테이블의 신청자, 피신청자, mutual 컬럼을 이용했다. 이웃 신청을 하는 경우, 실제 신청자와 실제 피신청자의 정보를 (신청자: 실제 신청자, 피신청자: 실제 피신청자, mutual: true), (신청자: 실제 피신청자, 피신청자: 실제 신청자, mutual: false) 형태로 저장하고 이웃 신청을 수락하는 경우 false를 true로, 거절되는 경우 true를 false로 변경 후 두 행을 삭제한다.

## 4.12. 회원 관리

회원 관리		
라즈베리파이 MAC	라즈베리파이 ID	사용자 이메일
00.00.00	1	<input type="text" value="nklcb0@pusan.ac.kr"/> <input type="button" value="변경"/>
11.11.11	2	<input type="text" value="nklcb1@pusan.ac.kr"/> <input type="button" value="변경"/>
22.22.22	3	<input type="text" value="nklcb2@pusan.ac.kr"/> <input type="button" value="변경"/>
33.33.33	4	<input type="text" value="nklcb3@pusan.ac.kr"/> <input type="button" value="변경"/>
44.44.44	5	<input type="text" value="nklcb4@pusan.ac.kr"/> <input type="button" value="변경"/>

그림 4-36. 회원 관리 페이지

- 관리자가 라즈베리파이와 사용자의 정보를 매칭하는 페이지이다.
- 이미 회원 정보가 등록된 라즈베리파이의 경우, "변경"을 누르면 버튼이 "등록"으로 바뀌며 사용자 이메일 입력란이 활성화되어 수정이 가능하다.
- 회원 정보가 등록되지 않은 라즈베리파이의 경우 사용자 이메일 입력란이 기본적으로 활성화 되어있다.
- 이메일 입력 후 등록 버튼을 누르면 해당 정보가 반영되며 이메일 입력란이 비활성화된다.
- 만약 이미 등록된 사용자를 중복 등록하거나 회원 가입하지 않은 사용자를 등록하면 경고 창을 띄우며 이메일 변경 내역을 복구하고 입력란이 비활성화 된다.

## 4.13. 설정



그림 4-37. 설정 페이지

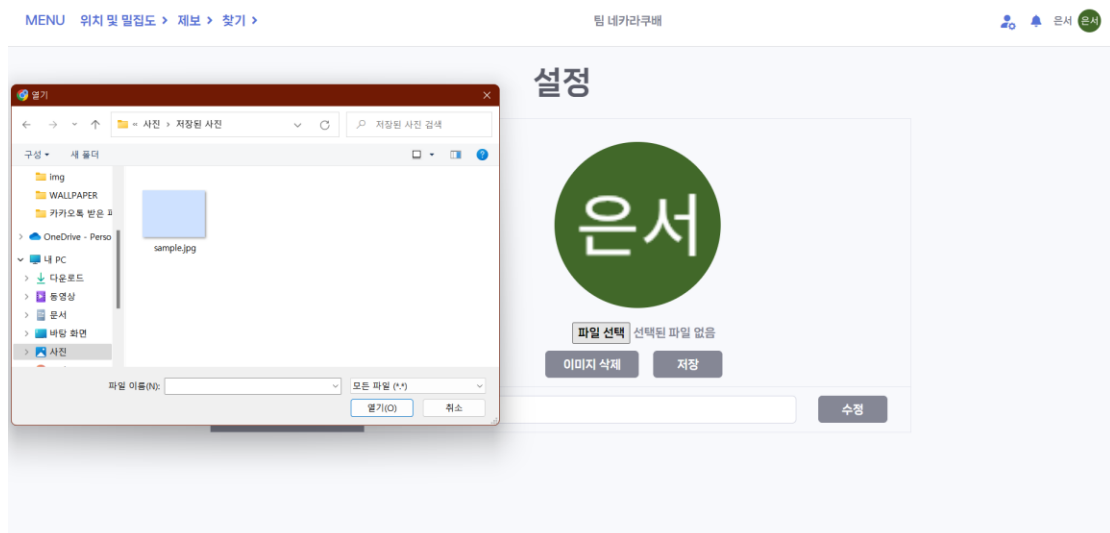


그림 4-38. 프로필 이미지 파일 선택



그림 4-39. 프로필 이미지 변경

- 사용자의 정보를 조회 및 수정할 수 있는 페이지이다.
- 파일 선택을 누르면 변경할 프로필 사진을 선택할 수 있는 창이 뜬다.
- 사진을 선택하고 저장 버튼을 누르면 즉각적으로 변경 내용이 반영되어 메뉴바의 프로필 사진이 바로 변경된 것을 볼 수 있다.
- 프로젝트의 정적 리소스 저장 폴더에 이미지를 저장하고 해당 파일의 위치와 파일명을 데이터베이스에 저장하는 방식으로 구현되었다.
- 프로젝트 재 시작없이 프로필 이미지의 변경을 반영하기 위해 WebMvcConfigurer 인터페이스를 구현, addResourceHandlers 메서드를 오버라이드하여 정적 리소스를 웹 애플리케이션에서 제공할 수 있도록 핸들러를 추가했다.



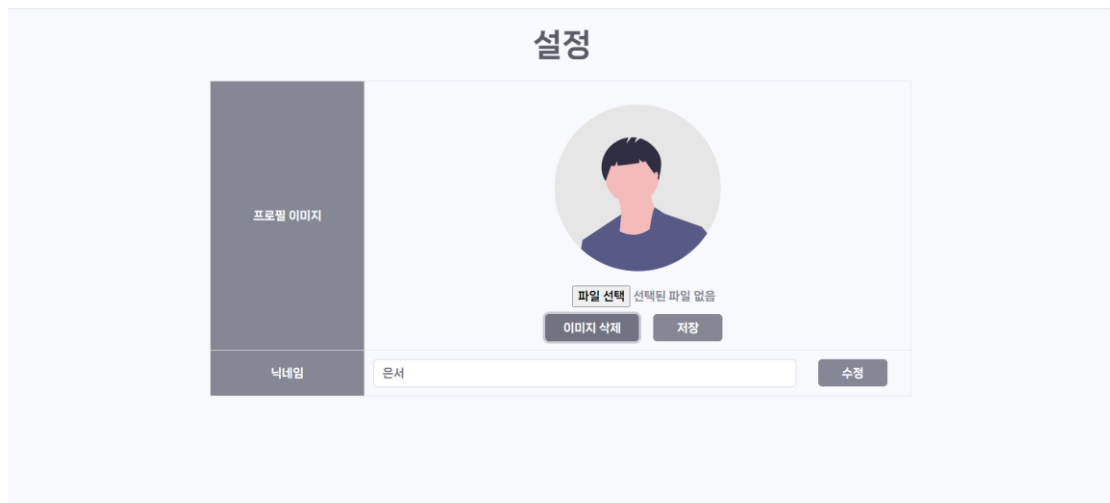


그림 4-40. 프로필 이미지 삭제

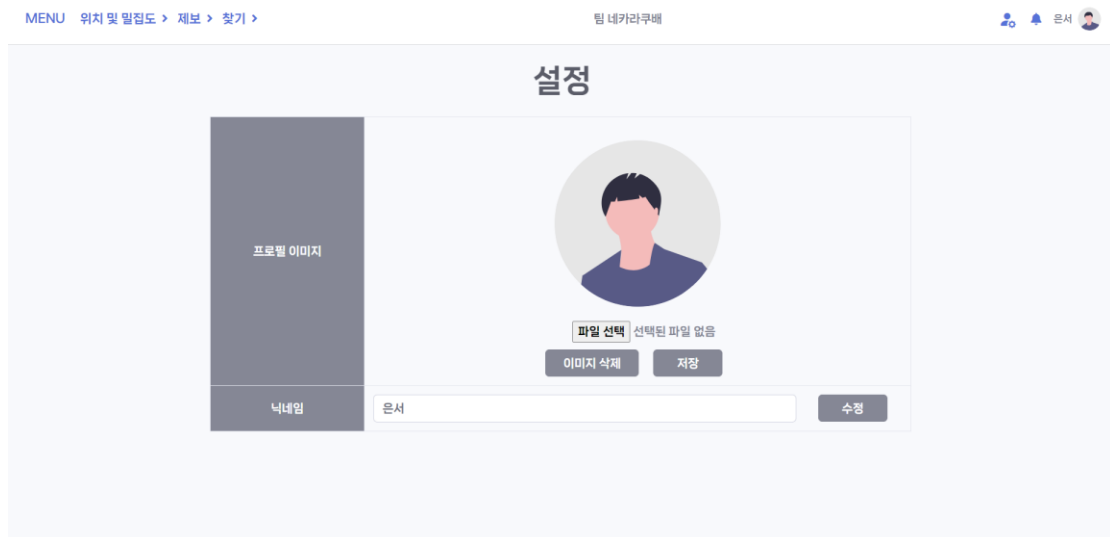


그림 4-41. 프로필 이미지 삭제 적용

- 이미지 삭제를 누르면 이미지가 설정되어 있는 기본 이미지로 변경되며, 저장을 누르면 적용된다.

### 설정

프로필 이미지

파일 선택 선택된 파일 없음

이미지 삭제 저장

닉네임

은서

수정

그림 4-42. 닉네임 수정

### 설정

프로필 이미지

파일 선택 선택된 파일 없음

이미지 삭제 저장

닉네임

관리자

수정

그림 4-43. 수정 닉네임 적용

- 닉네임도 사진과 마찬가지로 수정할 닉네임을 입력한 뒤 수정 버튼을 누르면 즉각적으로 닉네임이 변경되어 메뉴바에 반영된 것을 볼 수 있다.

#### 4.14. 반응형 웹

- 아이폰 12 Pro를 기준으로 보이는 화면이다.

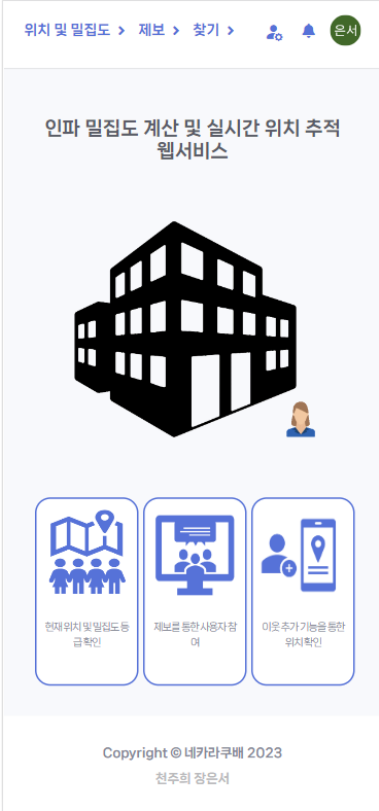


그림 4-44. 메인 페이지



그림 4-45. 실시간 밀집도 및 통계 확인



그림 4-46. 무선 신호 기반 실시간 밀집도



그림 4-47. 비전 기반 실시간 위치 추정



그림 4-48. 제보하기

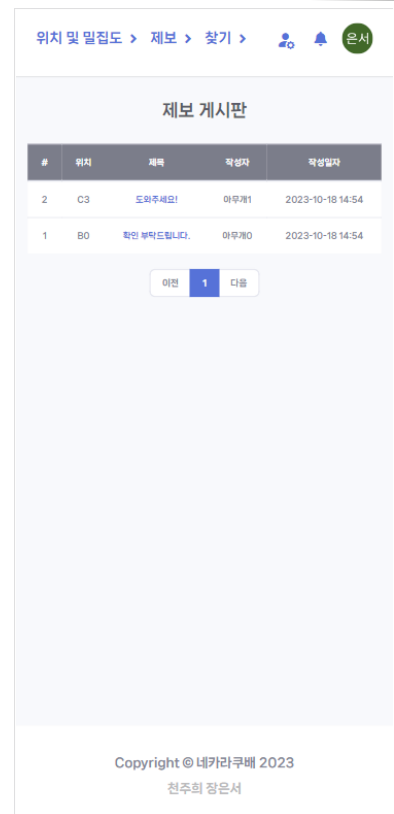


그림 4-49. 제보 게시판

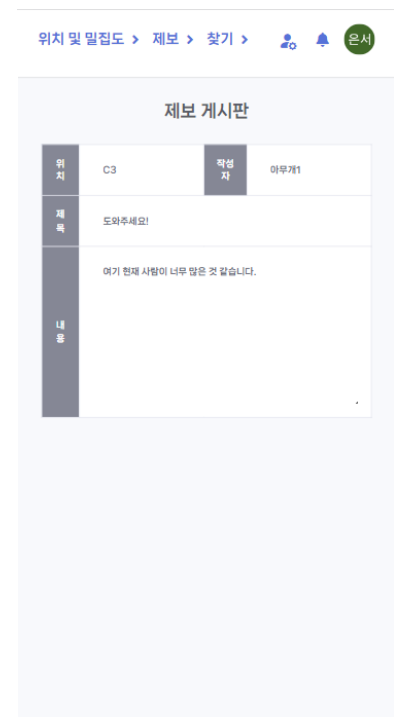


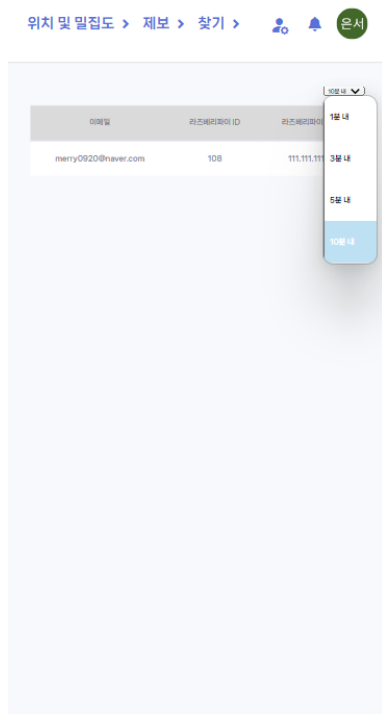
그림 4-49. 제보 게시판



그림 4-50. 사용자 찾기



그림 4-51. 특정 구역 사용자 찾기



Copyright © 네카라쿠배 2023  
천주의 장은서



Copyright © 네카라쿠배 2023  
천주의 장은서



Copyright © 네카라쿠배 2023  
천주의 장은서

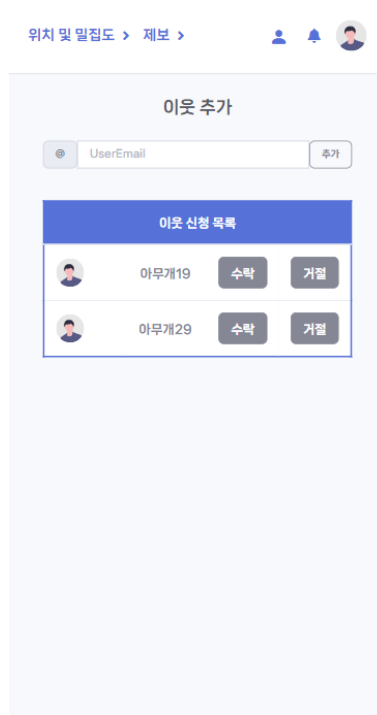
그림 4-52. 특정 구역 선택한 특정 구역 사용자 찾기

그림 4-53. 회원 관리

그림 4-54. 이웃 목록



그림 4-55. 이웃 위치 확인



Copyright © 네카라쿠배 2023  
천주의 장은서

그림 4- 56. 이웃 추가

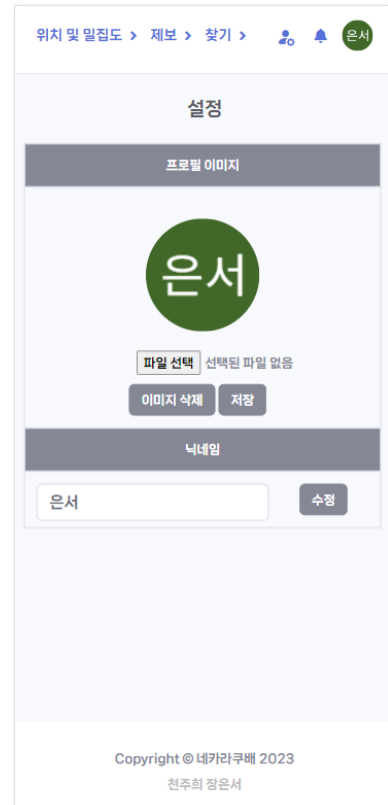


그림 4-57. 설정

---

## 5. 결론 및 향후 연구 방향

### 5.1. 결론

본 과제는 실내 다중밀집지역에서의 위급 상황의 예방 및 다중밀집사고 가능성이 높은 상황에 놓인 사람들이 스스로 위험 상황을 인지하여 사고를 대비할 수 있도록 정보와 서비스를 제공하는 플랫폼을 제작하고자 하였다.

사용자의 위치 추정 및 밀집도 측정, 밀집도에 따른 경보/알람 시스템을 구축해 사용자들은 실내 밀집도와 같은 안전 지표를 일상생활에서 마주하며 경각심을 높이고, 자신이 처한 상황의 심각성을 즉각적으로 인지할 수 있을 것으로 예상된다.

또한 사용자의 실내 존재 여부 및 실시간 위치를 관리자가 빠르게 파악할 수 있고, 사고 발생 직후에 기록된 데이터 및 로그를 통해 사고 원인 분석이 가능하므로 재난/재해 또는 실종 사고 발생 시 본 과제가 인명 사고 감소에 이바지할 것으로 기대된다.

인파가 많이 몰리는 공연장, 백화점, 지하철 등의 실내에서 밀집도 모니터링을 통해 사고 예방의 목적으로 일상에서도 사용할 수 있어 제품 활용도가 높으며, 공공 기관 및 민간 기업으로부터의 수요가 있을 것으로 예상된다.

### 5.2. 향후 연구 방향

- 비전 기반 밀집도
  - 카메라 화면을 가리는 현상이 발생한 경우, 가려진 화면 뒤에서 발생하는 상황을 감지하기 어렵고 밀집도의 정확한 계산이 불가능하다. 이를 개선하기 위하여 가장 최근의 인원수 정보를 활용하여 현재 밀집도를 예측하는 등의 방법을 고안할 수 있다.
- 인원 수 및 밀집도 예측
  - 본 과제에서는 단순 이동평균법을 이용하여 인원 수 및 밀집도를 예측했다. 인원 수 예측에 적합한 알고리즘을 연구, 적용하여 예측 정밀도 증대를 목표로 한다.
- 어플리케이션 개발
  - 현재는 라즈베리파이를 이용하여 무선 신호 기반 위치 추정을 구현했지만, 이후 휴대폰 어플리케이션으로 기능을 구현하여 사용자 편의성과 서비스 접근성을 높이려 한다.

● 시설물 제어 및 출동 서비스

- 위급 상황 발생 시, 관리자가 건물 내 시설물을 제어/관리할 수 있는 기능을 추가해 사고 직후 즉각적 조치에 도움을 줄 수 있다.
- 방법 업체와의 계약을 통해 출동 서비스를 지원하는 등의 서비스 확장이 가능할 것으로 예상된다.

## 6. 개발 일정 및 역할 분담

### 6.1. 개발 일정

5월					6월					7월					8월					9월					10월		
2주	3주	4주	5주		1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주
무선 네트워크 및 위치 추적 기술 학습																											
	실내 인파 밀집도 계산 알고리즘 개																										
		사용자 단말과 공유기 간의 연결 구현																									
					사람 인식 디 러닝																						
					웹 서비스 개발																						
															라즈베리파이 모듈에 사람 인식 시스템 적 용												
																서비스 공간에서 위치 추적과 카메라를 통한 사람 인식											
																											보고서 작성 및 최종 수정

### 6.2. 역할 분담

이름	역할
천주희	<ul style="list-style-type: none"> <li>- 다수의 라즈베리파이를 이용한 실시간 위치 추정 기술 개발</li> <li>- 영상 스트리밍 서버 및 웹 서버 기능 구현</li> <li>- 카메라를 이용한 정밀 위치 추정 기술 개발</li> <li>- 서버 간 통신 구현</li> <li>- 서버 배포</li> </ul>
장은서	<ul style="list-style-type: none"> <li>- 웹 UI 화면 구성</li> <li>- 웹 UI 디자인</li> <li>- 반응형 웹 제작</li> <li>- 인원수 및 밀집도 예측 모델 개발</li> </ul>
공동	<ul style="list-style-type: none"> <li>- 위치 추정 기술, 무선 네트워크 기술 학습 및 개발</li> <li>- YOLOv5를 통한 사람 인식 딥러닝 기술 학습</li> <li>- 보고서 작성</li> </ul>

---

## 7. 참고 문헌

- [1] ""여기도 호흡곤란" 9 호선 급행의 비명". 박지민, 김혜연, 오유진 (2023.05.09). 조선일보. Available: [https://www.chosun.com/national/national\\_general/2023/05/09/B7RTLZD6ZNDRDBE2ZLMN7MCGFM/](https://www.chosun.com/national/national_general/2023/05/09/B7RTLZD6ZNDRDBE2ZLMN7MCGFM/)
- [2] "인파 물리는 곳 불안감 확산... "군중밀집 재난문자 도입 필요" [이태원 헬러원 참사]". 안승진, 구윤모 (2022.11.03). 세계일보. Available: <https://m.segye.com/view/20221102515323>
- [3] "OpenCv Perspective Transformation". Raqueeb Shaikh. (2020.06.27). Medium. Available: <https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143>
- [4] "OpenCV python - 원근(투영) 변환 - perspective (projection) transform". House-of-E. (2022.06.07). tistory. Available: <https://house-of-e.tistory.com/entry/11-OpenCV-python-%EC%9B%90%EA%B7%BC%ED%88%AC%EC%98%81-%EB%B3%80%ED%99%98-perspective-projection-transform>
- [5] "여러 명이 동시에 접속하려면? — select". 박응용. (n.d.). wikidocs. Available: <https://wikidocs.net/125626>
- [6] "[Bootstrap] 부트스트랩 그리드 시스템 (Grid System)". Meeeeeejin. tistory. (2020.06.12). Available: <https://mjmmjmj98.tistory.com/19>
- [7] "국내 주요 압사사고". 원형민. (2022.10.30). 연합뉴스. Available: <https://www.yna.co.kr/view/GYH20221030000700044>
- [8] "美콘서트서 2명 압사..."총소리 들렸다" 관객들 출입구 몰려". 신유리. (2023.03.07). 연합뉴스. Available: <https://www.yna.co.kr/view/AKR20230307120300009>
- [9] "군중밀집도 임계점은..."m²당 5명부터 위험"". 연합뉴스. (2022.10.31). Available: <https://www.yna.co.kr/view/MYH20221031017400038>
- [10] "핑거 프린트 방식의 위치 추적 시스템 및 그 방법". 차호정, 김윤근, 전요한. (2011.03.15). Available: <https://patents.google.com/patent/KR20110026229A/ko>