

## 0. 작성 시 주의사항

※아래의 작성 양식(제출분량, 폰트, 크기, 줄 간격 등)을 미준수 시 서류 평가의 감점요인됨

- ※ 제출 분량 : A4 용지 상세내용 포함 30 page 이내
- ※ 작성 양식 (폰트 : 맑은 고딕 / 폰트 크기 : 10pt / 자간 : 0% / 장평 : 100% / 줄 간격 : 130%)
- ※ 제출 포맷 : pdf

## 1. 팀 정보

팀명	KOBOT	팀장	서명희
팀원	김명진	팀원	박대영
팀원	이정하	팀원	

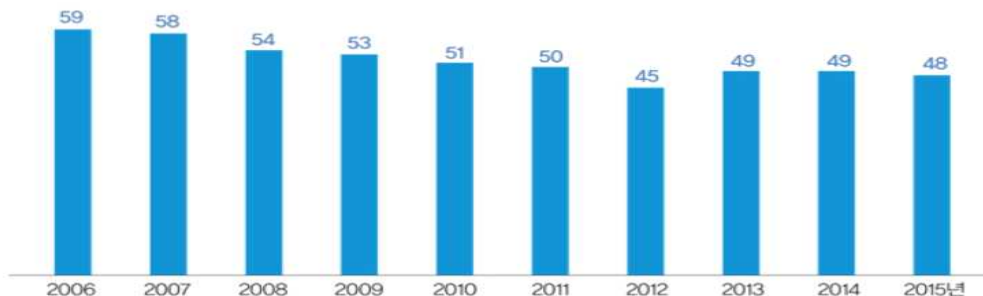
## 2. 개발완료보고서

### 0. 작품명 : Air Police

#### 1. 개요

##### 1.1. 작품 개요

2006년 이후 전국 미세먼지(PM10) 연평균 농도



최근 3년간 1~4월 전국 초미세먼지(PM2.5) 평균 농도

시기	2015년	2016년	2017년	국내 연간 환경 기준	세계보건기구(WTO) 연간 권장 기준
농도	28	30	31	25	10

단위: ug/m<sup>3</sup>, 자료: 국립환경과학원

그림1. 우리나라 전국 미세먼지 연평균 농도

중국발 미세먼지를 포함한 대기 상태의 심각성이 대두되면서 많은 사람이 실외 대기 상태에 민감한 태도를 보이지만 정작 하루 12시간 이상을 보내는 실내 공간의 공기 상태에 대해서는 안일하게 생각하고 있다. 하지만 실내 공기 상태는 미세먼지와 같은 외부적 요인 외에도 건축자재로부터 발생하는 오염물질의 방출, 요리, 각종 생활용품, 침구류 등의 내부적인 요인에 의해 실외 오염 수치보다 더 높아질 수 있다. 특히 노약자, 호흡기 질환을 가지고 있거나 면역력이 떨어지는 사람들의 경우에는 일반 성인보다 많은 시간을 실내에서 보내고 있으므로 실내 공기 오염이 인체에 많은 영향을 준다.

국가에서도 미세먼지에 대한 심각성을 인지하고 이를 예방하고 해결하기 위해 측정소를 운영하여 실외 미세먼지를 측정하고 미세먼지를 줄이기 위한 다양한 방안들을 내고 있다. 하지만 현재 우리가 위치한 곳과 측정소의 위치가 정확하게 일치하는 것이 아니므로 아래의 그림 2과 같이 실내외 미세먼지 수치 또한 차이가 클 수밖에 없다. (왼쪽은 실제 거주지와 가장 가까운 측정소의 미세먼지 수치이고 오른쪽은 실내에서의 측정 수치이다.)

## 우리동네 대기 정보

지역명 검색: 서울특별시 성북구 정릉로 77(정릉동) 검색

선택	측정소명	측정소 주소	거리	측정항
●	성북구	서울 성북구 삼양로2길 70 갈음2동 주민센터	2.9km	도시대기
○	강북구	서울 강북구 삼양로 139길 49 우이동 주민센터	4.1km	도시대기
○	서대문구	서울 서대문구 세경정로4길 32(중제3동 주민센터)	4.6km	도시대기

※ 거주지역의 대표 대기질은 "도시대기" 측정자료를 참고하시기 바랍니다.

측정자료 조회: 통합대기 환경지수 조회

데이터 구분: ● 시간 ○ 일평균  
 조회기간: 2019-09-17 ~ 2019-09-17 검색 그래프 보기 열람

측정자료(수치)  
 ※ 최종인증된 실시간자료는 경정색 글씨로 표시되며, 최종확정자료는 파란색 글씨로 표시됩니다.

날짜 (월-일시)	PM <sub>10</sub> ( $\mu\text{g}/\text{m}^3$ )	PM <sub>2.5</sub> ( $\mu\text{g}/\text{m}^3$ )	오존 (ppm)	이산화질소 (ppm)	일산화탄소 (ppm)	아황산가스 (ppm)
	1시간	1시간	1시간	1시간	1시간	1시간
09-17 01	19	8	0.013	0.030	0.6	0.003

```

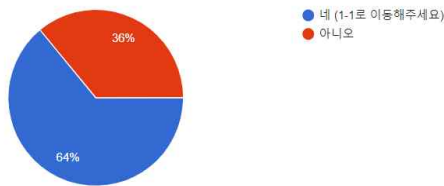
20:46:15.983 -> gas = 8.03
20:46:15.983 -> humidity = 44.20
20:46:16.018 -> temperature = 25.09
20:46:16.816 -> PM 1.0 (ug/m3): 7
20:46:16.850 -> PM 2.5 (ug/m3): 15
20:46:16.850 -> PM 10.0 (ug/m3): 16
20:46:16.885 ->
20:46:16.885 -> gas = 8.00
20:46:16.920 -> humidity = 44.03
20:46:16.920 -> temperature = 25.10
20:46:17.722 -> PM 1.0 (ug/m3): 6
20:46:17.757 -> PM 2.5 (ug/m3): 14
20:46:17.792 -> PM 10.0 (ug/m3): 15
20:46:17.792 ->
20:46:17.792 -> gas = 8.00
20:46:17.826 -> humidity = 44.97
20:46:17.826 -> temperature = 25.11
20:46:18.661 -> PM 1.0 (ug/m3): 6
20:46:18.661 -> PM 2.5 (ug/m3): 13
20:46:18.696 -> PM 10.0 (ug/m3): 14
20:46:18.696 ->
20:46:18.696 -> gas = 8.00
20:46:18.730 -> humidity = 44.42
20:46:18.730 -> temperature = 25.11
20:46:19.559 -> PM 1.0 (ug/m3): 6
20:46:19.594 -> PM 2.5 (ug/m3): 12
20:46:19.594 -> PM 10.0 (ug/m3): 13
20:46:19.627 ->
20:46:19.627 -> gas = 7.97
20:46:19.627 -> humidity = 44.98
20:46:19.661 -> temperature = 25.13
20:46:20.411 -> PM 1.0 (ug/m3): 6
20:46:20.411 -> PM 2.5 (ug/m3): 12
20:46:20.446 -> PM 10.0 (ug/m3): 13
20:46:20.446 ->
    
```

그림2. 실내미세먼지와 실외미세먼지 비교

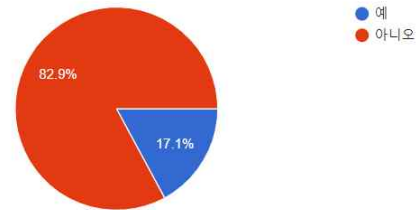
또한 실내는 미세먼지뿐만 아니라 가스누출에 대한 위험도 있다. 가스 누출 관련 사고는 여전히 지속해서 일어나고 있고 대부분의 사고에서 사망자가 발생하는 실정이다. 그러나 우리가 스스로 가스 누출을 인지하기 어렵고 인지하여도 이미 늦은 상황일 확률이 높다. 이러한 상황에 대한 경각심을 일깨우고 해결하기 위해 우리는 전체적인 실내 공기 상태에 대한 수치를 직접 확인하고 효율적으로 실내 공기질을 관리, 개선할 수 있는 서비스인 Air Police를 개발하였다.

### 1.1.1 프로젝트 배경 및 사전 조사 내용

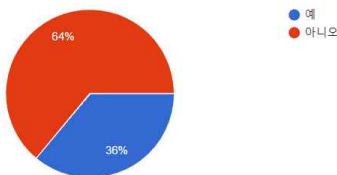
1. 공기 청정기를 집에 보유 하고 있으신가요?



1-1 보유하고 있다면 각 방마다 설치하고 있으신가요?



2. 공기청정기가 청정 할 수 있는 구역이 정해져 있다는 것을 알고 있나요?



3. 한 대의 공기청정기가 집안의 모든 구역을 관리 할 수 있다면 좋다고 생각하시나요?

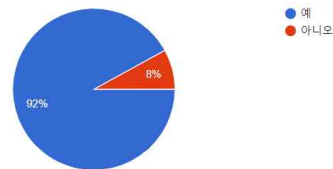


그림3. 공기질 및 공기청정기에 대한 인식 조사

이 프로젝트를 시작하기에 앞서 사람들의 공기질 및 공기청정기에 대한 인식을 조사하기 위해 간단한 설문 조사를 진행하였고, 결과는 위의 그림.3의 결과와 같았다. 추가로 '1-2. 방마다 없다면

그 이유는 무엇인가요?’의 질문 같은 경우에는 총 비율의 83%가량의 많은 사람들이 비용적인 문제를 답해주었고 10%가량의 사람들은 공간적인 문제로 설치하지 않은 이유를 답해주었다. 본 프로젝트의 배경은 다음과 같다.

- (1) 2013년부터 미세먼지 문제가 부각되기 시작한 이유는 여러 가지로 제시되고 있다. 2013년에 PM2.5를 대기환경 기준물질로 지정하여 기준을 설정하고 미세먼지 예보제를 실시할 예정이라고 정부에서 발표하였고(MOE, 2013), 2013년 이후 언론에서 미세먼지를 보도한 건수가 증가한 것 또한 국민들의 관심과 불안을 증가시킨 것으로 보인다.
- (2) 최근 서울을 위시한 대도시지역의 미세먼지의 오염도는 지속적해서 증가하고 있는 것으로 나타나고 있으며, 광역화되는 특성을 보인다.
- (3) 미세먼지는 한반도 전역에 걸쳐 나타나고 있는 광범위한 환경 문제이다. WHO는 전세계인의 92%가 대기오염의 영향을 받으며, 이로 인해 해마다 700만 명의 사람이 목숨을 잃는다고 밝혔다(2014 기준).
- (4) 미세먼지는 피부와 점막에 직접 접촉하여 자극을 유발하고 호흡기와 혈관을 통해 인체에 영향을 준다. 질병관리본부에 따르면 미세먼지(PM 10) 농도가 공기 1당 10 $\mu$ g 증가할 때마다 '만성 폐쇄성 폐 질환(COPD)'으로 인한 입원율은 2.7%, 사망률은 1.1% 증가했다. 그보다 작은 초미세먼지(PM2.5)의 경우 농도가 1당 10 $\mu$ g 증가할 때마다 폐암 발생률은 무려 9.0% 늘어나는 것으로 나타났다.
- (5) 최근에는 가스온수보일러, 기름 온수보일러, 가스온수기, 이동형가스난방기 등 개별 난방 보급이 확대되면서 화재, 폭발, 중독사고가 지속적으로 발생하고 있으며, 중독사고의 경우에는 물적피해는 없지만 일시적으로 모두에게 피해를 주기 때문에 사회적으로 문제제기가 되고 있다.

### 1.1.2 기술 시장 현황

국립환경과학원(2013)에 따르면 인간은 하루에 섭취하는 음식물과 물보다 10배 이상의 많은 공기를 흡입하고 있는 것으로 알려져 있다. 청정공기는 실외 대기와 달리 자정 능력이 없는 밀폐된 실내환경으로, 삶의 질의 중요사항으로 인식이 확대되는 중으로 갈수록 악화하고 있는 국내외 미세/초미세먼지 이슈 등으로 인해 관련 수요는 급증하고 시장도 확대되고 있다. 한국과학기술정보연구원(2015)은 공기청정기의 세계시장 규모를 2014년 기준으로 약 41억 달러에 이르는 것으로 추정하였으며, 국가별 비중은 일본 26%, 미국, 22%, 중국 22%, 캐나다 10%, 한국 9%, 기타(유럽 등) 11%으로 보았다. 그림 4에 나온 바와 같이 공기청정기의 세계시장 규모는 꾸준히 증가하고 있다.

구분	2013	2014	2015	2016	2017	2018	2019	(단위: 억 달러, 억 원, %)
								연평균성장률(CAGR)
세계	38	41	48	57	67	78	92	16.1
국내	3,000	5,000	5,600	6,272	7,025	7,868	8,812	12.0

그림4. 2014년 기준 국가별 공기청정기 세계시장 규모

중소기업기술정보진흥원(2016)의 중소·중견기업 기술로드맵에 따르면 국내시장은 삼성, LG 등 대기업의 참여와 함께 코웨이, 대유위니아, 위닉스 등 중견기업이 높은 시장점유율을 확보하고 있다. 최근 초미세먼지 이슈로 소비자의 구매력이 높아지고 고성능 제품 수요가 증가하면서 삼성, LG 등 대기업의 프리미엄 제품이 시장점유율을 확대하고 있으며 중국의 샤오미 등 수입 중저가 제품의 판매량도 함께 증가하는 등 시장 경쟁이 갈수록 심화하고 있다.

## 1.2. 개발 목표

실외에서 생활하는 동안 높은 농도의 미세먼지에 노출되어 피로감을 느끼는 사람들이 증가하고 있다. 지속적으로 미세먼지에 노출될 경우 각종 질환의 위험성이 높아질 수 있다. 특히 노약자, 임산부, 호흡기 질환을 가지고 있는 사람들에게는 더욱 취약하다. 그러나 실외에서 유입되는 미세먼지뿐만 아니라 실내에서 생활하면서 발생하는 미세먼지의 영향도 무시할 수 없다. 우리가 주로 생활하는 실내에서 음식을 조리하거나 옷을 털거나 청소를 할 때도 미세먼지가 발생한다.

하지만 실내 모든 공간에 공기청정기를 설치하는 것은 현실적으로 어려운 일이다. 그래서 우리는 본 프로젝트를 통해 좀 더 저렴한 비용으로 실내 미세먼지 수치를 측정하고 공기질을 꾸준히 관리할 수 있는 Air Police를 개발하게 되었다.

구체적인 개발 목표는 다음과 같다.

(1) 실외 공기질의 상태는 쉽게 알 수 있지만, 우리가 주로 생활하는 실내의 공기질 오염 상태에 대한 정보는 정확하게 알 수 없다. 이에 KOBOT팀은 집의 방마다 다양한 센서를 통해 공기질(미세먼지, 온/습도, 가스)을 측정하고 사용자에게 신뢰할 수 있는 정보를 UI를 통해 시각적으로 제공한다.

(2) KOBOT팀은 집을 크게 안방, 주방, 거실 3부분으로 나누었고 해당 방마다 센서를 설치하였다. 방마다 공기청정기가 있는 것이 가장 이상적이겠지만 현실적으로는 많은 어려움이 따른다. 따라서 가장 공기질이 나쁜 장소부터 우선순위로 두어 스마트 공기청정봇인 'POLI'로 공기를 정화하여 쾌적한 실내 환경을 유지할 수 있도록 도울 수 있다.

(3) 일반적으로 가스레인지가 있는 주방에서 가스가 누출된다면 미미한 농도라도 일산화탄소가 누출되었을 때 일산화탄소 중독의 위험이 있으며, 높은 농도의 일산화탄소에 노출되면 단 시간 내에 사망할 수 있다. 이를 예방하기 위해 추가로 가스 센서를 설치함으로써 webOS 기반 웹앱을 통해 수치 정보를 쉽게 확인하고 위험한 상황에 경보를 울릴 수 있다.

2. 개발 환경 설명

2.1. Hardware 구성

본 프로젝트 Air police에서 HW는 Poli(Turtlebot3), RaspberryPI3(WebOS, Sensor kit 3개, Server)로 구성되어 있다. 하드웨어의 전체적인 구성은 다음과 같다.

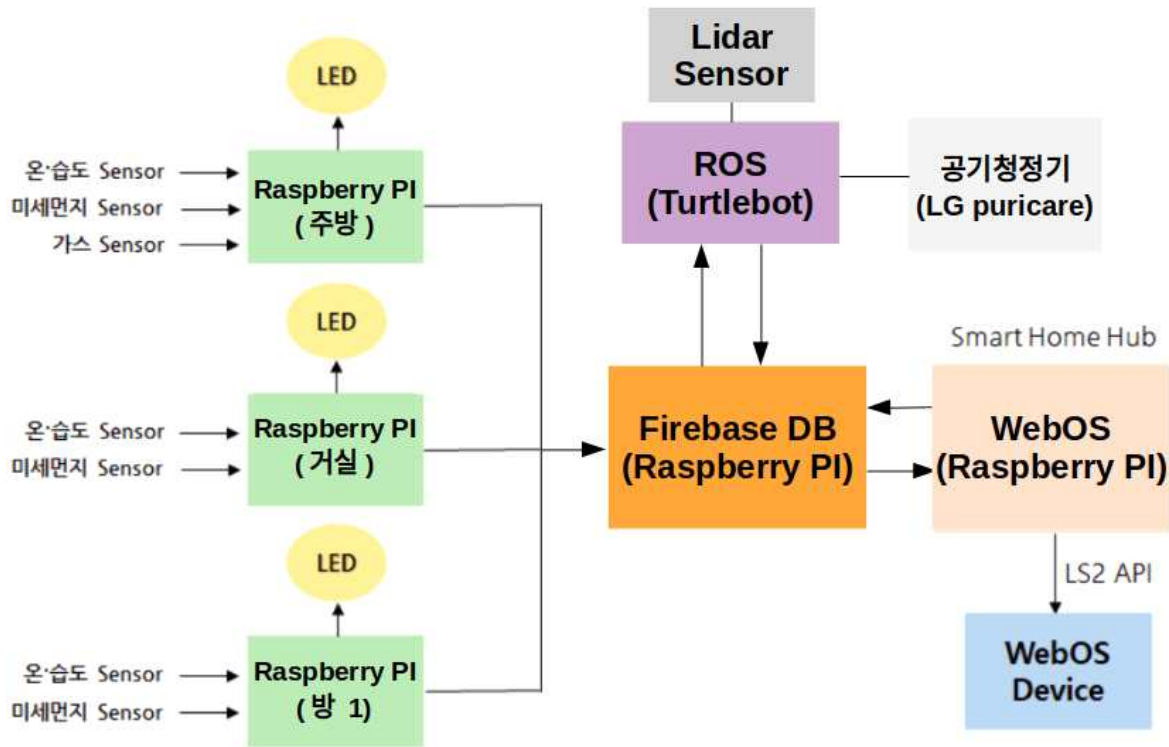


그림5. HW 구성도

(1) Poli (Turtlebot3)

Firestore Realtime DB에 저장된 값을 호출하여 PM2.5, PM10의 농도가 30이상인 곳이 있다면 그 위치로 이동하여 공기를 청정하도록 한다. 그리고 요리모드가 켜져있을때는 Poli의 전원을 끄고

요리모드가 꺼진다면 다시 공기청정을 위한 준비를 한다. 만약 Poli가 Base camp의 위치가 아니라면 Base camp의 위치로 이동하게 한다. 또한 사용자가 원하는 위치로 이동하고 싶으면 web app에서 설정하여 사용자가 원하는 위치로 보낸다.

(2) WebOS (Raspberry pi3)

WebOS가 탑재된 라즈베리파이로, AirPolice web app이 실행되는 중심역할을 한다. web app에서는 Firebase DB에 저장된 공기질 수치값들을 받아 보여주고, 가스수치를 감시하여 위험 기준 초과 시 luna api로 notification과 경보를 보내준다. 필요에 따라서는 사용자에게 알려야 할 상황마다 notification을 띄워준다. 그리고 Air Police의 모드를 조절할 수 있고, 이에 따라 Firebase DB를 통해 터틀봇의 주행을 조절한다.

(3) Server (Raspberry pi3)

Crawler 프로그램이 탑재된 라즈베리파이로, 실외 공기질에 대한 정보를 API 호출을 통해 가져오며 이를 Firebase Realtime DB에 저장하여 데이터를 관리한다. Raspberry PI가 전원이 켜지면 프로그램이 자동으로 실행되며 이는 서버와 같은 역할을 한다.

(4) 실내 공기질 측정 키트 (Raspberry pi3)

각 방에 Raspberry pi3를 기반으로 구성된 실내 공기질 측정 키트를 설치한다. 주방에 설치하는 센서 키트에는 미세먼지 센서와 온·습도 센서, 가스 센서가 부착되어있고 주방을 제외한 다른 공간의 Raspberry pi3에는 가스센서를 제외한 미세먼지 센서, 온·습도 센서가 부착되어있다. 그리고 각 키트마다 미세먼지 농도를 한 눈에 쉽게 볼 수 있도록 현재 공기질의 상태를 LED를 이용하여 표현한다.

2.2. Software 구성

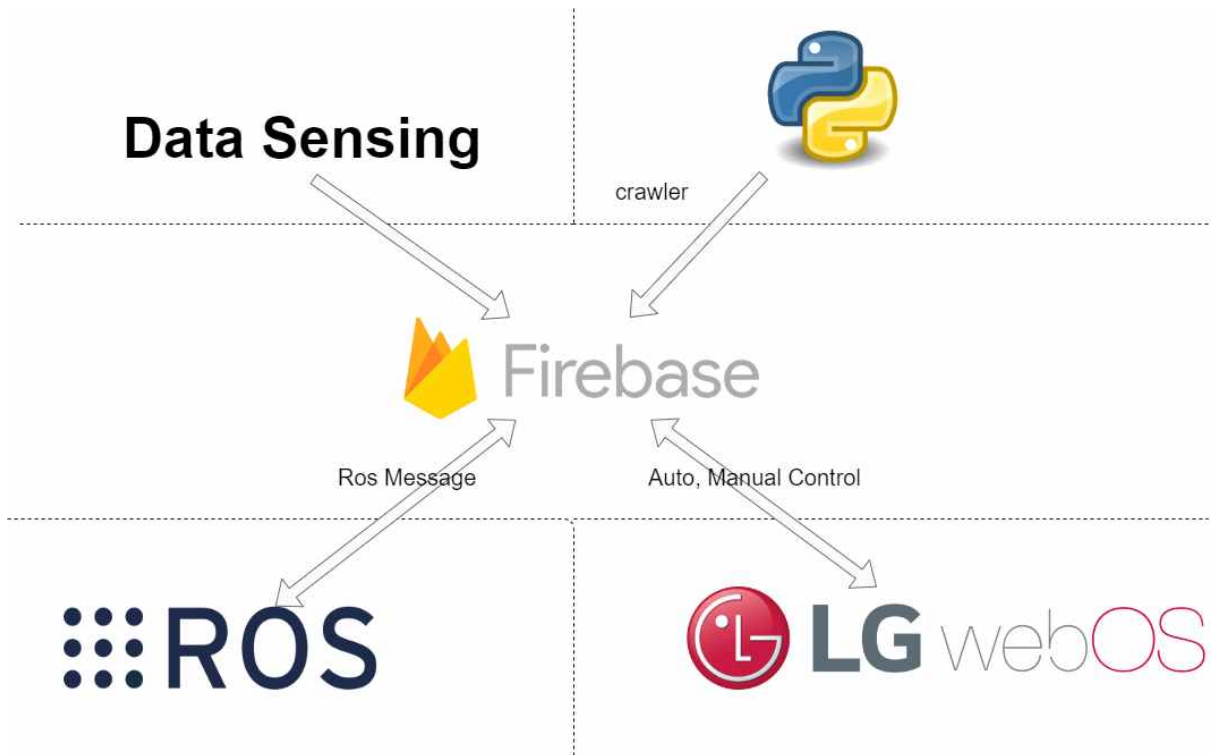


그림6. software 구성 흐름도

### 2.3. Software 설계도

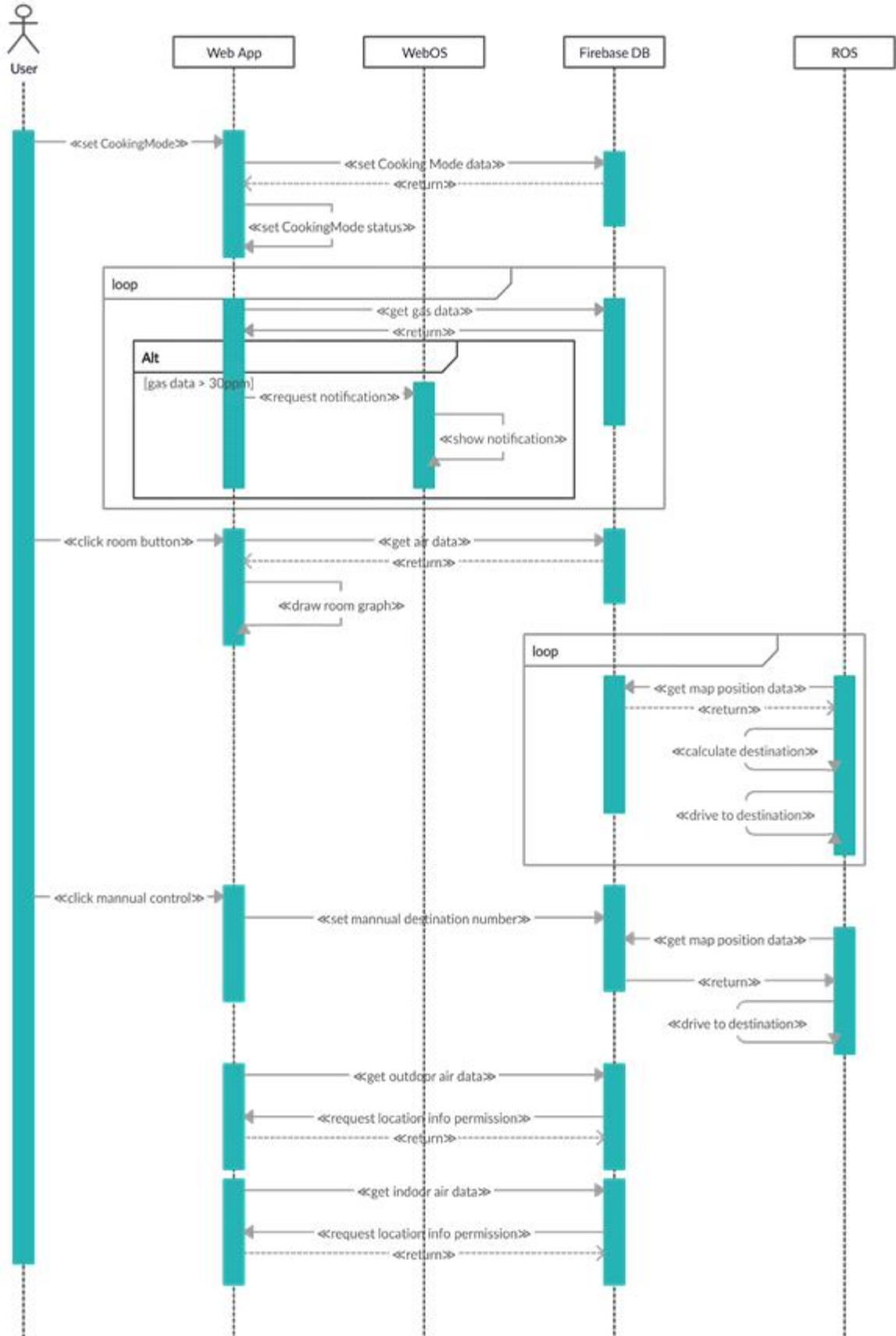


그림7. 클래스 다이어그램

## 2.4. Software 기능

### 2.4.1 알고리즘

#### (1) Poli(Turtlebot) 이동 판별 알고리즘

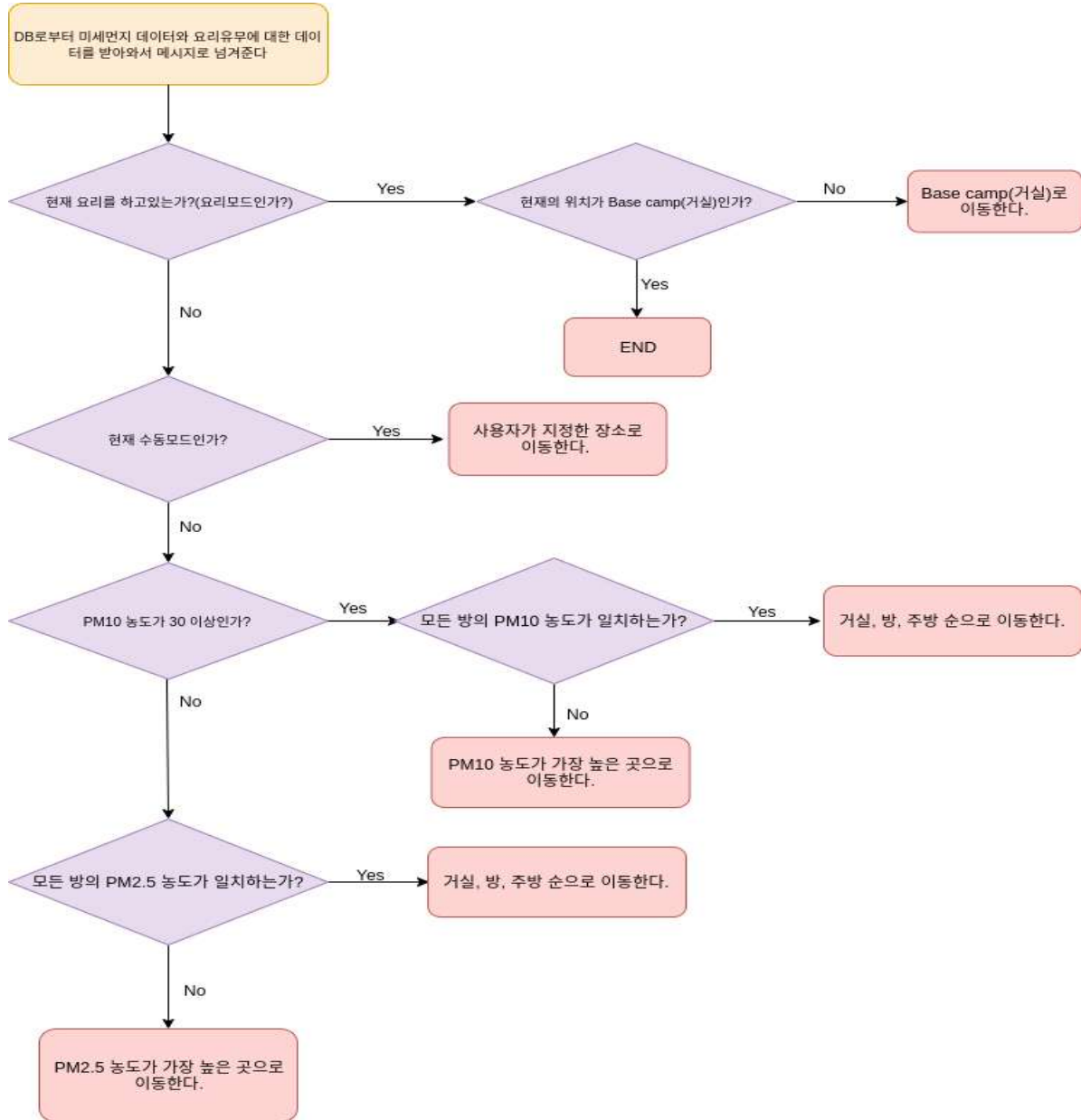


그림8. 터틀봇 이동 판별 알고리즘

Firestore Realtime DB로부터 요리의 유무와 수동이동의 유무, 이동해야하는 위치에 대한 데이터를 받아온다. 처음에 요리유무에 대한 데이터부터 체크한다. 요리 중이라면 요리모드를 사용자가 켜서 모든 공기청정을 중단한다. 요리중이 아니라면 요리모드는 해제되어 있을것이다. 그 다음 수동이동의 유무의 값을 확인한다. 수동모드일 때는 자동으로 공기청정을 중지하고 사용자가 지정한 장소로 이동한다. 만약 수동모드가 해제된다면 자동 공기청정을 시작한다.

표1. WHO 미세먼지 기준

	좋음	보통	나쁨	매우나쁨
미세먼지(PM10)	0~30	31~50	51~100	101~
초미세먼지(PM2.5)	0~15	16~25	26~50	51~

자동 모드의 경우, WHO 미세먼지 기준이 보통일 경우에 공기청정을 시작한다.

만약 PM10의 농도가 30 이상이라면 모든 방의 농도가 일치하지 않으면 PM10농도가 가장 높은 곳으로 이동하지만 만약 모든 방의 농도가 일치하는 경우에는 거실, 방, 주방 순으로 이동한다. 만약 PM10 농도가 30 이상이 아니라면 더 작은 입자인 PM2.5의 농도로 판단한다. 만약 모든 방의 농도가 일치한다면 거실, 방, 주방 순으로 이동하고 아니라면 PM2.5 농도가 가장 안 좋은 곳으로 이동한다.

(2) 실내 공기질 측정 알고리즘

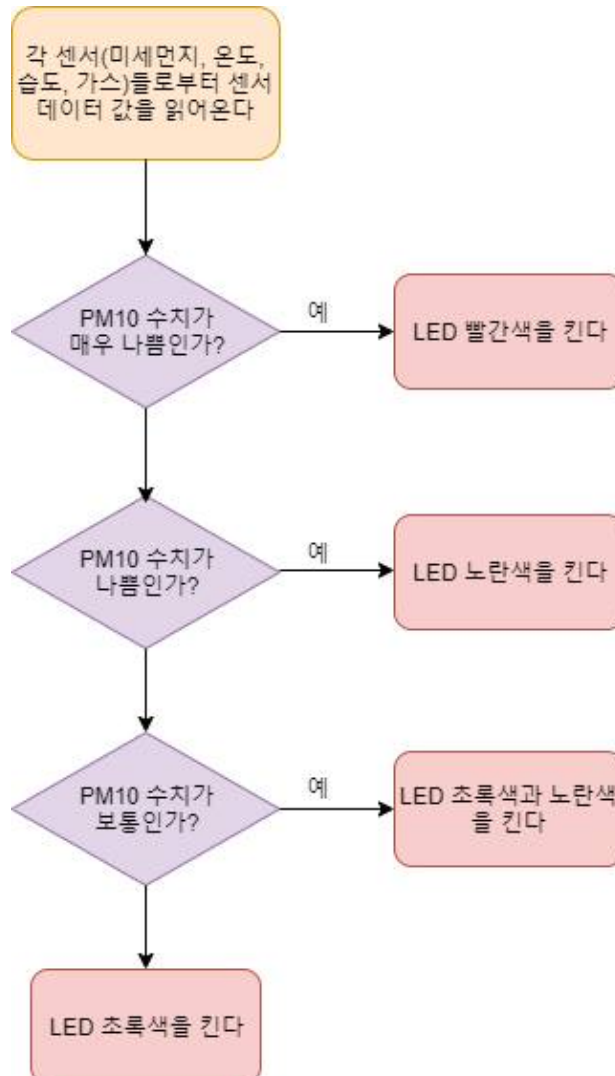


그림9. 실내 미세먼지에 따른 LED 점등 알고리즘

실내 공기질 측정 알고리즘은 각 센서들로부터 값들을 취합하여 Firabase Realtime DB에 값들을 전송해주고 미세먼지 값(PM10)들을 측정하여 WHO 기준(표 1)에 의하여 LED 색을 표출한다.

(3) 실외 공기질 측정 알고리즘

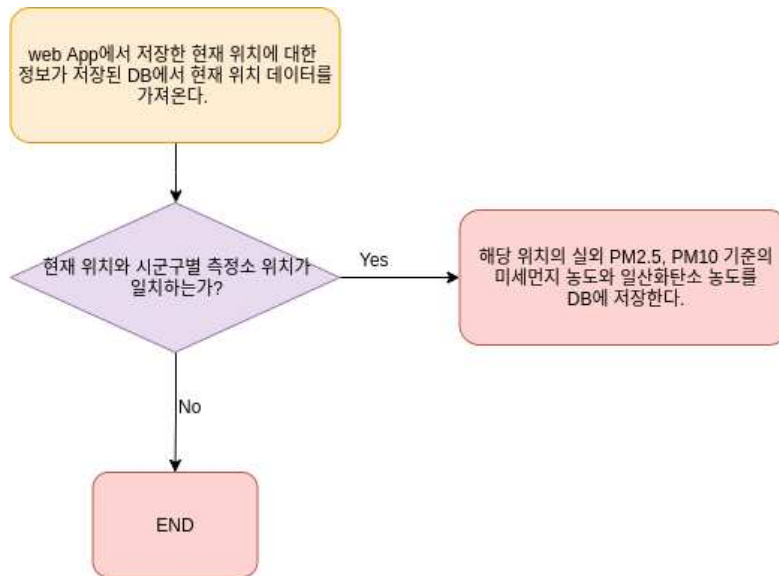


그림10. 실외 미세먼지, 일산화탄소 측정 알고리즘

실외 공기질 측정 알고리즘은 일단 web app에서 GPS로 받아온 현재 위치에 대한 정보를 DB에 저장하고 저장된 데이터를 가져온다. 만약 현재 위치와 시군구별 측정소 위치가 일치한다면 해당 위치의 실외 PM2.5, PM10 기준의 미세먼지 농도와 일산화탄소 농도를 DB에 저장한다. 만약 일치하지 않는다면 정보를 가져올 수 없다.

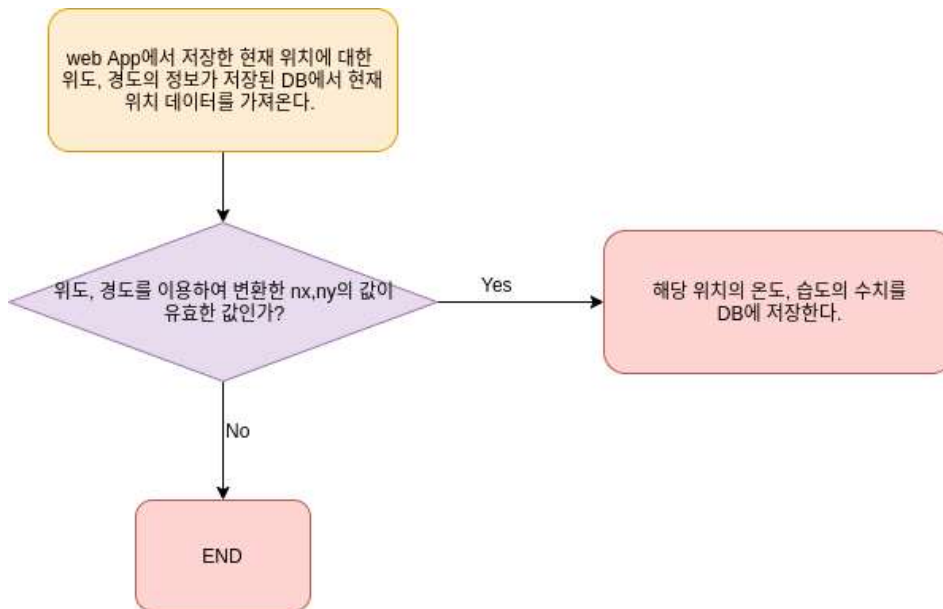


그림11. 실외 온도, 습도 측정 알고리즘

web app에서 저장한 현재 위치에 대한 위도, 경도의 정보를 DB에서 가져온다. 그리고 위 값을 기반으로 날씨 API에서 필요한 nx, ny 값으로 변환하고 이 값이 유효한지 확인한다. 만약 유효하다면 해당 위치의 온도, 습도의 수치를 DB에 저장한다. 만약 유효하지 않은 값이라면 정보를 가져올 수 없다.

(4) 실내 이산화탄소 수치 알림 및 경보 알고리즘

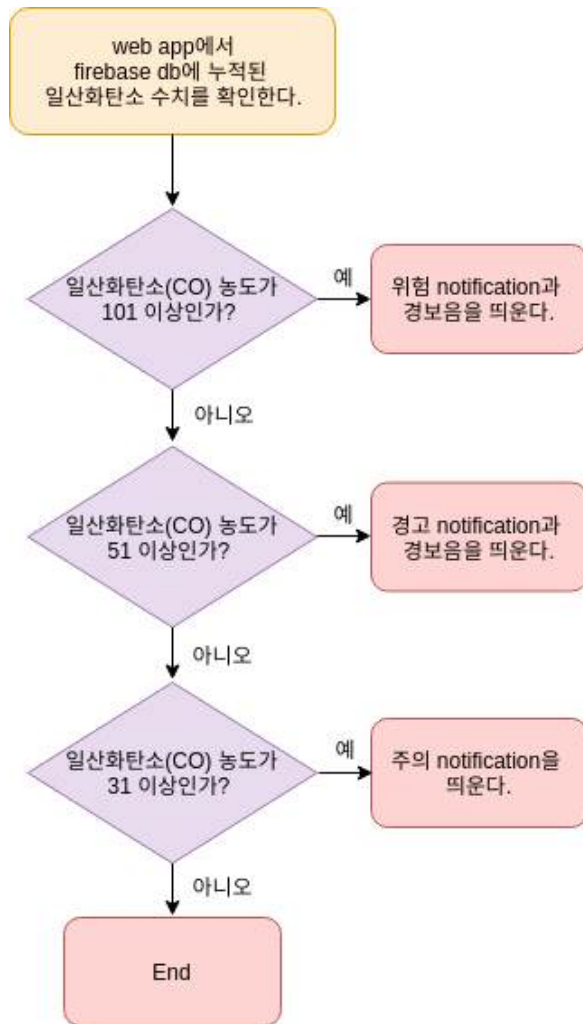


그림12. 이산화탄소 수치에 따른 notification 및 경보음 알고리즘

주방의 가스 센서를 통해 측정된 가스수치의 값을 web app에서 수시로 확인하여 특정 수치 이상의 값이 감지되었을 때, luna api 중 하나인 notification을 이용하여 디바이스에서 시스템 notification을 띄워준다. 수치가 '주의' 단계에 해당되면 단순히 주의 notification을 띄워준다. 수치가 증가하여 '경고' 단계로 넘어가면 경고 notification과 함께 경보음을 띄워준다. 수치가 '위험' 단계로 넘어간다면, 위험 notification과 함께 경보음을 띄운다. (알람은 75dB 이상으로 울리게 하고 기준은 유럽연합 경보농도시험 기준을 따른다)

표2. 일산화탄소 기준

	안전	주의	경고	위험
CO농도(ppm)	0~30	31~50	51~100	101~

## 2.4.2 데이터 관리

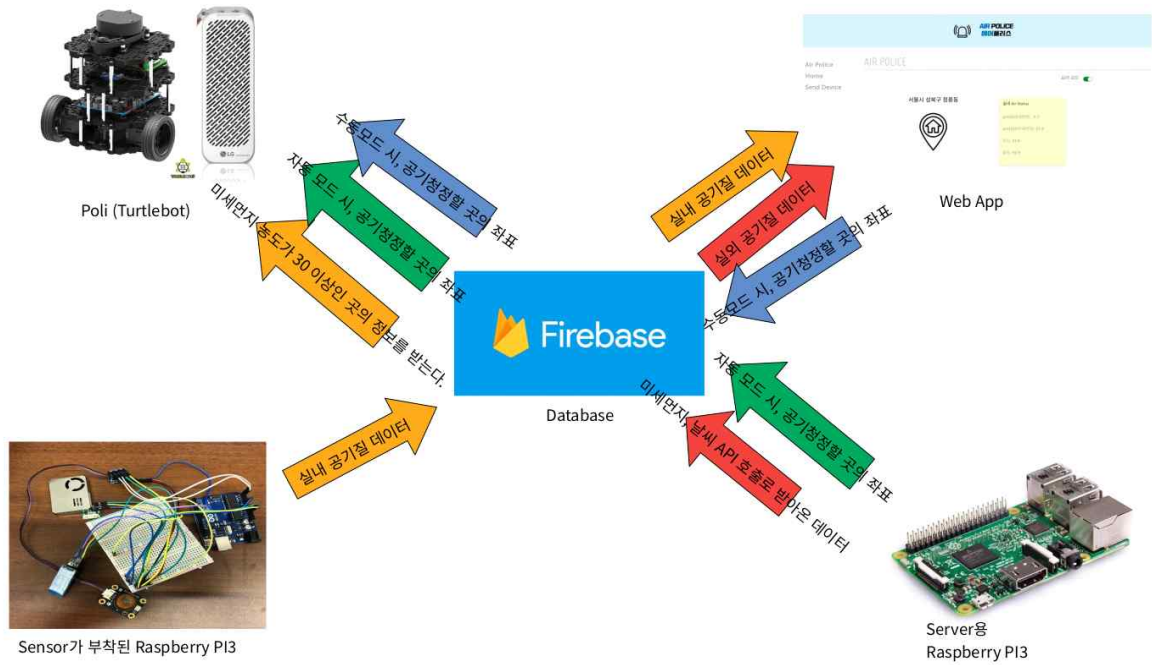


그림13. 데이터 흐름도

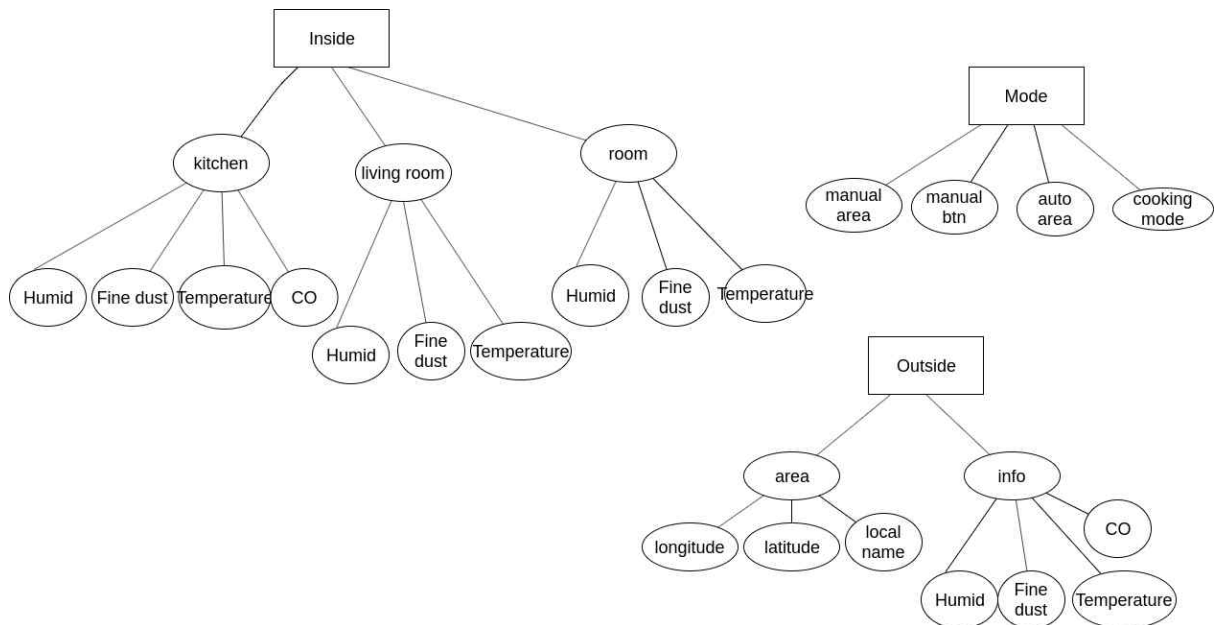


그림14. DB E-R 다이어그램

본 프로젝트에서는 Firebase Realtime DB를 이용하여 데이터를 관리한다. 날씨, 미세먼지 관련 API를 호출하여 실외 데이터를 DB에 저장하고 실내 데이터는 센서키트에서 바로 DB에 저장한다. 그리고 web app에서 DB에 접근하여 데이터를 가져온 후 출력한다.

POLI(Turtlebot3)은 web app에서 만약 사용자가 수동 조작을 선택한다면 원하는 위치를 DB에서 받아와 이동하여 공기 청정을 시작한다. 그리고 자동 조작이라면 DB에 저장된 미세먼지 농도가 가장 좋지 않은 방으로 순서대로 이동하여 공기 청정을 시작한다. 또한 web app에서 사용자가 요리모드를 선택한다면 1, 요리모드를 해제하면 0 값을 받아 DB에 저장한다.

```

{
  "inside" : {
    "kitchen" : {
      "date":{
        "time": {
          "coValue" : "일산화탄소 농도",
          "humid" : "습도",
          "pm10Value" : "pm10 미세먼지 농도",
          "pm25Value" : "pm2.5 미세먼지 농도",
          "temp" : "온도"
        }
      }
    },
    "living room" : {
      "date":{
        "time": {
          "humid" : "습도",
          "pm10Value" : "pm10 미세먼지 농도",
          "pm25Value" : "pm2.5 미세먼지 농도",
          "temp" : "온도"
        }
      }
    },
    "room" : {
      "date":{
        "time": {
          "humid" : "습도",
          "pm10Value" : "pm10 미세먼지 농도",
          "pm25Value" : "pm2.5 미세먼지 농도",
          "temp" : "온도"
        }
      }
    }
  },
  "mode" : {
    "auto_area" : 1,
    "cooking_mode" : 1,
    "manual_area" : 1,
    "manual_btn" : 0
  }
},

```

```

"outside" : {
  "area": {
    "latitude" : "현 위치의 위도",
    "longitude" : "현 위치의 경도",
    "local" : "현위치 지리명"
  },
  "info" : {
    "coValue" : "일산화탄소 농도",
    "humid" : "습도",
    "pm10Value" : "pm10 미세먼지 농도",
    "pm25Value" : "pm2.5 미세먼지 농도",
    "temp" : "온도"
  }
}

```

그림 15. Firebase Realtime DB(JSON형식)

DB는 크게 inside, mode, outside로 나누어진다. inside는 실내 공기질 데이터, mode는 자동모드, 요리모드, 수동모드에 대한 신호를 가지고 있다. manual\_btn는 수동모드 버튼이며 1이면 수동모드 on, 0이면 자동모드이며 자동모드일 때 auto\_area, 수동모드일 때 manual\_area의 value의 경우 공기청정을 할 방 위치에 대한 번호를 나타낸다. (참고로 0: 거실(base camp), 1: 주방, 2: 방 1이다.)

outside는 현재 위치에 기반한 실외 공기질 데이터를 가지고 있으며 area는 현재 위치에 대한 위도, 경도에 대한 데이터이며 위 데이터를 기반으로 local 이라는 현위치 지리명을 가져와 저장한다. info에서 coValue는 일산화탄소 농도, humid는 습도, pm10value은 pm10 단위의 미세먼지 농도, pm25Value는 pm2.5 단위의 미세먼지 농도, si는 현재 위치의 시 이름, temp는 온도이다.

inside에는 kitchen, living room, room으로 세 구역으로 나누어지며 하루씩 데이터를 분류하고 관리하여 web app에서 일별, 주별 단위의 그래프를 그릴 때 데이터 접근의 효율성을 높였다. 그리고 일별 정보로 들어가면 0시부터 23시까지 5분단위로 센서가 측정한 값을 받아와서 DB에 저장하고 관리한다. 5분 단위로 측정하여 더 정확하고 신뢰성 있는 데이터를 구축해나간다. 5분

단위로 측정값을 저장하는데 주방의 경우는 일산화탄소, 습도, 온도, pm2.5, pm10단위의 미세먼지 농도, 주방 이외의 거실, 방의 경우에는 일산화탄소를 제외한 습도, 온도, pm2.5, pm10단위의 미세먼지 농도가 있다. coValue는 일산화탄소 농도, humid는 습도, pm10Value는 pm10 단위의 미세먼지 농도, pm25Value는 pm2.5 단위의 미세먼지 농도, temp는 온도를 가져오는 key 값이다.

### 2.4.3 통계 관리

데이터 시각화 라이브러리인 Rechart를 이용하여 미세먼지 수치, 온도, 습도, 가스 수치를 꺾은선 그래프로 구현하였다. Firebase DB에서 필요한 방의 특정 공기질 수치에 대해 필요한 일(day), 주(week)의 값들을 가져온 후, data 배열에 저장하고 Rechart에 인자로 전달하여 통계 그래프를 그린다. 시각화 된 그래프는 web app을 통해 사용자에게 보여주고, 사용자는 web app을 통해 집의 공기질에 관한 누적 데이터를 일, 주 별로 직접 확인할 수 있다.

### 2.4.4 ROS Navigation 기능과 LIDAR를 이용한 실내 구조 모델링

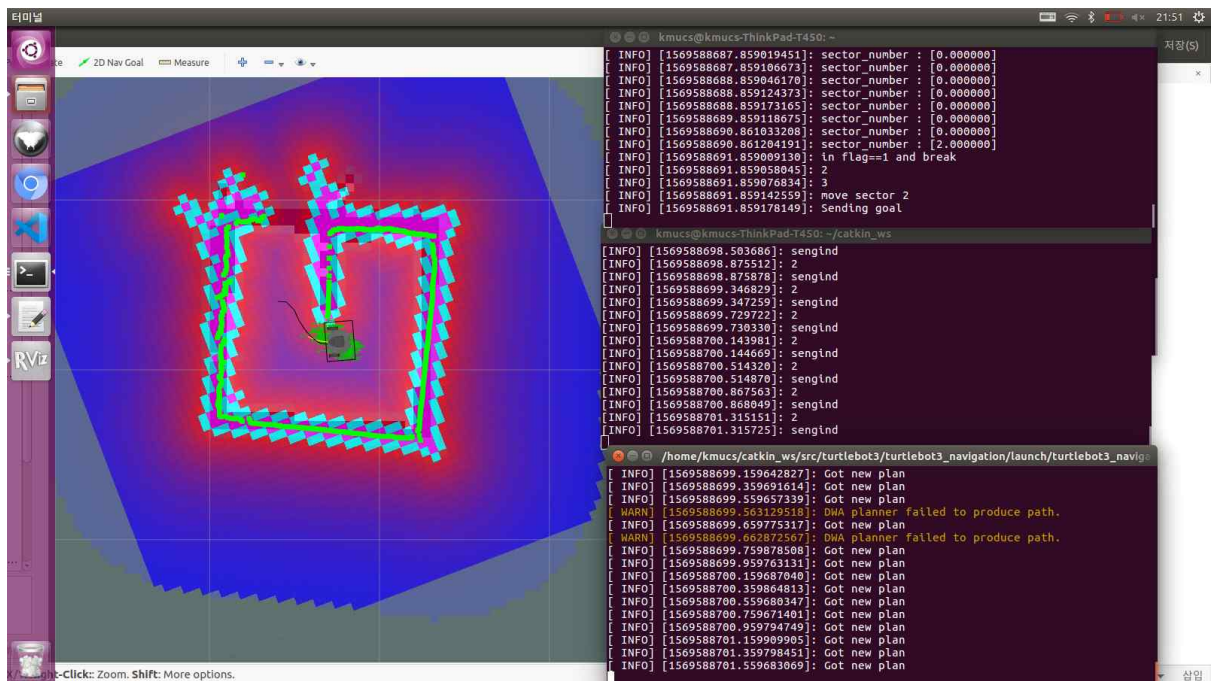


그림16. ROS Navigation 기능과 LIDAR를 이용한 실내 구조 모델링

POLI(Turtlebot3)는 LIDAR센서에서 받아온 값을 이용하여 집 내부를 돌며 Rviz를 사용하여 맵을 그린다. Remote-PC에 저장된 맵을 바탕으로 ROS Navigation노드를 실행한다. 원래 Navigation노드는 마우스를 사용하여 맵을 이동하여야 하지만, ROS의 SendingSimpleGoals 라이브러리를 사용함으로써 원하는 위치의 좌표값을 계산하여 이동하도록 개선하였다.

## 2.5. UI 사용법

### <메인 화면(Air Police)>



그림 17-1. 메인 페이지



그림 17-2. 일산화탄소 수치가 기준을 넘었을 때 notification가 뜨면서 메시지를 보여준다. (ls2)

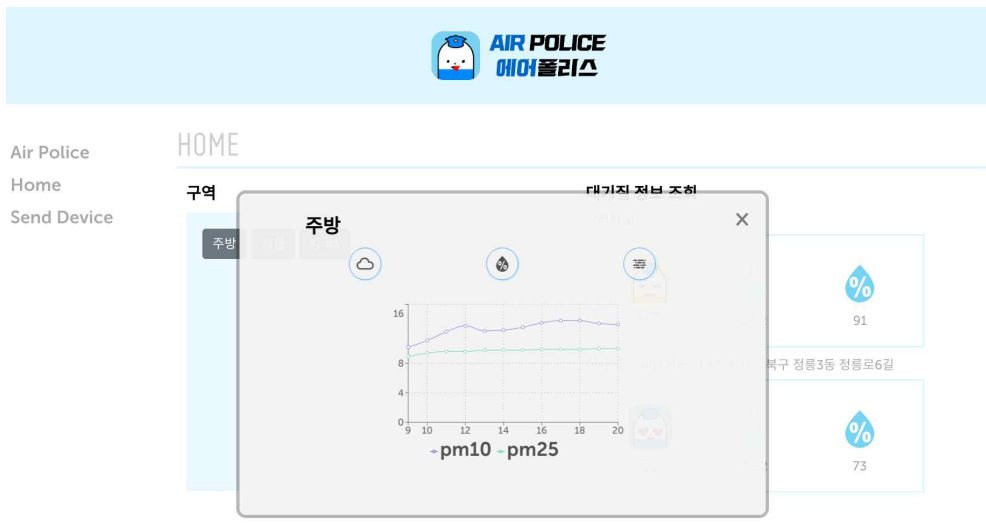


그림 17-3. 요리모드 toggle이 on으로 바뀐 상태

<Home 화면>



그림 17-4. Roomgraph를 볼 수 있는 방 버튼으로 구성된 구역 부분과 실내, 실외의 공기질 상태를 시각화하여 볼 수 있는 Airbox로 구성되어 있다.



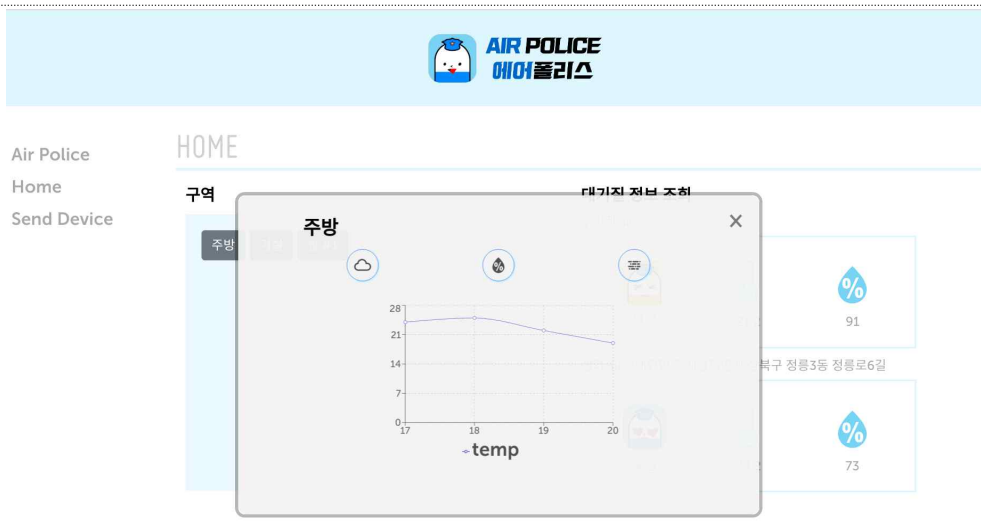


그림 17-5, 17-6, 17-7. 구역에서 원하는 방의 버튼을 누르면 해당 방에 대한 미세먼지(pm10, pm2.5), 습도, 온도에 대한 그래프를 볼 수 있다.

<Send Device 화면>

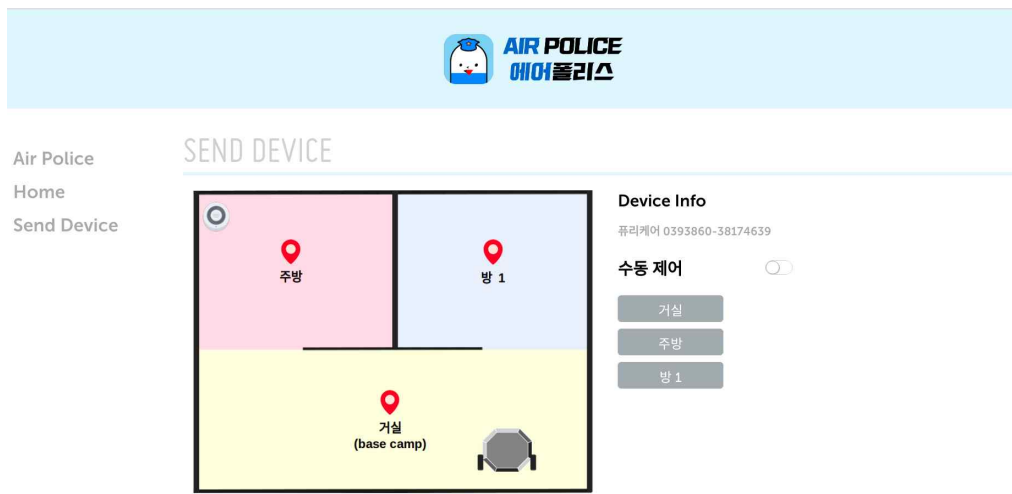


그림 17-8. 터틀봇이 이동하려는 위치를 시각적으로 볼 수 있는 우리 집의 map 부분, 연결된 공기청정기 device에 대한 정보와 수동제어를 할 수 있는 부분으로 구성되어 있다.

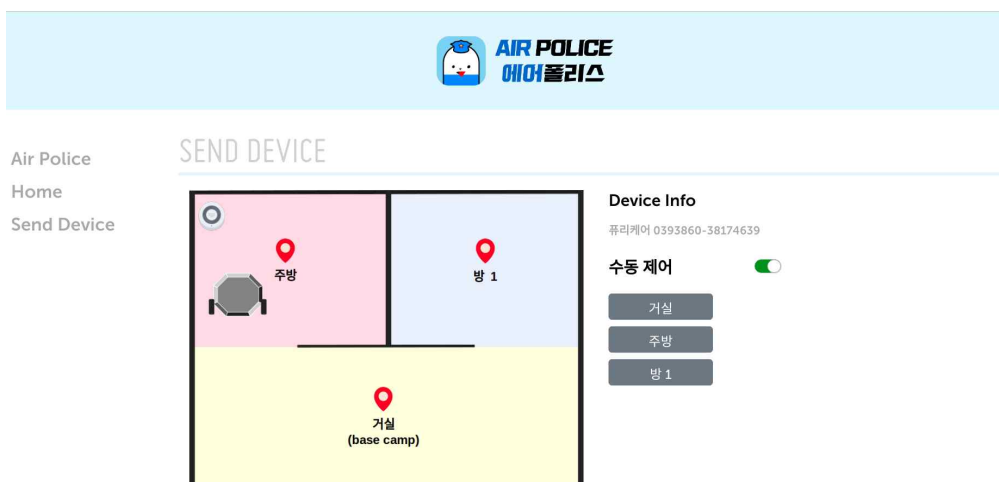


그림 17-9. 수동 제어 toggle을 on으로 바꾸고 주방으로 turtlebot을 수동으로 보내어, 왼쪽의 map에 turtlebot의 위치가 반영된 모습이다.

## 2.6. 개발환경

### [하드웨어 개발환경]

				
그림18. Raspberry PI 3	그림19. 미세먼지 센서 (PMS7003M)	그림20. 온 / 습도 센서 (DHT22)	그림21. 가스센서 (CO) (MQ-7B)	그림22. Arduino LED 3/5/8/10 파이
				
그림23. TurtleBot3		그림24. LG 퓨리케어 미니 공기 청정기		

제품명	주요정보	
Raspberry Pi 3B	Chip	Broadcom BCM2837
	Cpu	Quad Cortex A53 @ 1.2GHz
	Memory	1GB SDRAM
	Operating System	Boots from Micro SD, Linux operating system
미세먼지 센서 (PMS7003M)	Range of measurement	0.3~1.0; 1.0~2.5; 2.5~10.0; Micrometer( $\mu$ m)
	DC Power supply	4.5 / 5.5 Volt (V)
	Effective Range	0 ~ 500 Micrometer( $\mu$ m)
	Working Temperature Range	-10 ~ 60 Celsius( $^{\circ}$ C)
온 / 습도 센서 (BME280)	Temperature Sensing	-40 ~85 Celsius( $^{\circ}$ C)
	Operating Voltage	3.3 / 5 Volt(V)
가스센서 (CO) (MQ-7B)	Detection Range	10 ~ 500 ppm
	Operation Voltage	1.5 / 5 Volt(V)
Arduino LED 3/5/8/10 파이	Rated Voltage	2.0 ~ 3.2v
TurtleBot3	Embedded Controller	OpenCR(32-bit ARM <sup>®</sup> Cortex <sup>®</sup> -M7)
	Sensor	360 <sup>°</sup> Laser Distance Sensor LDS-01
LG 퓨리케어 미니 공기 청정기	공기청정화능력(CADR)	13m <sup>3</sup> /hr(강기준)
	정격전압	DC 5V
	소비전력	12.0W
	배터리	일체형 배터리(3,600mAh)
	센서	포터블 PM1.0

## [소프트웨어 개발환경]

구분	내용
개발환경	Linux, WebOS, Windows
개발언어	C++, JavaScript, Python
개발 Tool	CLI, Pycharm, Visual Studio,
개발 framework	Enact

### 3. 개발 프로그램 설명 (최대한 자세하게 기술)

#### 3.1. 파일 구성

##### (1) Web App

<src/App/>

- App.js: web app의 큰 틀을 구성하는 파일이며 헤드로고와 메뉴, 메뉴선택에 따라 해당하는 view를 보여준다.

- View.js: web app에서 index에 해당하는 메뉴 영역을 보여준다.

<src/components/>

- AirboxInside.js: firebase에 저장된 현재의 실내 공기질 데이터를 불러와 아이콘과 함께 보여주는 박스 영역에 해당하는 파일

- AirboxOutside.js: firebase에 저장된 현재의 실외 공기질 데이터를 불러와 아이콘과 함께 보여주는 박스 영역에 해당하는 파일

- CookingSwitch.js: 요리모드의 상태를 조절하고 db에 해당 상태값을 넘겨주는 toggle switch component

- Headname.js: web app의 헤드로고를 구성하는 파일

- Roomgraph.js: 특정 방의 특정 공기질 항목에 해당하는 데이터에 대해 누적된 수치를 그래프로 그려주는 파일

<src/views/>

- Main.js: Airpolice web app에서 가장 첫 번째 메뉴인 Air Police에 해당하는 view영역으로 센서가 설치된 실내 공기질 수치를 보여주며, 요리모드를 설정할 수 있다.

- Home.js: 구역(방)마다 notification를 볼 수 있는 버튼 영역과 AirboxInside, AirboxOutside로 구성되어 있는 view영역이다.

- Device.js: 방의 구조와 함께 센서와 디바이스가 설치된 위치를 보여주며, 연결된 디바이스 정보를 알 수 있다. 또한 수동제어/자동제어를 조절할 수 있으며, 수동 제어를 통해 원하는 위치로 Airpolice를 보낼 수 있는 view영역으로 이루어져 있다.

- LunaNotification.js: ls2 중 notification의 createToast method를 사용하여 특정 수치 이상의 가스가 감지되었을 때, toast를 띄운다.

- Room.js: 특정 방에 해당하는 Roomgraph 컴포넌트를 불러온다.

<패키징 시 생성되는 파일>

- appinfo.json: 패키징한 web app에 대한 정보, 사용하려는 luna api의 permission을 위한 값들이 정의되어 있다.

## (2) Raspberry PI 3b (Server)

- auto.py : 크롤러 파일을 자동으로 실행시킨다.

## (3) Raspberry PI 3b (Sensor)

- sensors.py : pms7003m, dht22, mq-7 , 3개의 센서를 통해 값을 받고 데이터베이스에 값을 집어 넣어준다.

## (4) Crawler

- kFirebase.py: Firebase Realtime DB에 데이터를 저장하거나 가져오는 것이 가능하다.

- openApi\_call.py : 현재 위치로 시군구별 미세먼지 농도를 크롤링하여 미세먼지(PM10), 초미세먼지(PM2.5) 데이터를 Firebase Realtime DB에 저장할 수 있도록 한다.

- trans.py: 날씨 정보를 가지고 있는 동네예보조회 API를 사용하기 위해서 위도, 경도 데이터를 API parameter인 nx, ny로 변환할 수 있도록 한다.

- weather\_api.py: 동네예보조회API를 사용해 현재 위치의 온도와 습도를 크롤링하여 데이터를 Firebase Realtime DB에 저장할 수 있도록 한다.

- main.py: 필요한 모듈을 모두 실행할 수 있도록 한다.

## (5) ROS (Turtlebot)

- Msg.msg : ROS내에서 새로운 형태의 메시지를 전달하기 위해서 정의해준다.

- chatter.py : 센서를 이용하여 저장된 데이터를 Firebase Realtime DB에서 판별하여 저장되어 있는 요리의 유무와 수동 이동의 유무 그리고 이동해야하는 위치를 받아와서 ROS의 Publisher를 사용하여 메시지 형태로 보내준다.

- navigation\_goals.cpp : ROS의 Subscriber를 사용하여 받은 메시지를 확인하여 어디로 이동하여야 하는지 확인하고 터틀봇이 이동하도록 navigation노드에 메시지를 보내준다. 정상적으로 도착 후에는 ROS의 Publisher를 사용하여 도착유무에 대한 메시지를 넘겨준다. 도착을 했으면 1, 이동중이라면 0을 넘겨준다.

- sending.py : navigation\_goals.cpp에서 ROS의 Subscriber를 사용하여 받은 메시지를 확인하고 Firebase Realtime DB에 값을 갱신해준다.

- CMakeLists.txt : CMake를 이용하여 빌드환경을 기술하고 있다. 이는 실행 파일 생성, 의존성 패키지 우선 빌드, 링크 생성 등을 설정하게 되어 있다.

- package.xml : 패키지 정보를 담은 XML 파일로써 패키지의 이름, 저작자, 라이선스, 의존성 패키지 등을 기술하고 있다.

- map.pgm : 맵의 형태가 이미지 형태로 저장되어 있다.

- mpa.yaml : Navigation 노드를 실행하기 위해서 호출하는 파일로 map.pgm의 경로 맵의 크기 및 초기위치가 저장되어 있다.

## 3.2. 함수별 기능

### (1) Web App

- showAirbox(): 현재 시간을 기준으로 state에 받아온 공기질 수치에 따라 아이콘과 함께 상태를 시각적으로 보여준다.
- update\_data(num): main view에서 현재 선택된 방의 미세먼지, 온도, 습도 값을 state에 각각 저장한다.
- toggleSwitch() - in CookingSwitch.js: 요리모드를 조절하기 위한 state값을 변경하고 firebase에 넘겨준다.
- luna\_api(): webosose에서 제공하는 luna api를 사용하기 위해 조건문에 따라 요청을 보내는 함수이다.
- handleOpen(expNum): home view에서 클릭한 방에 대한 notification을 보여주기 위해 state를 변경한다.
- handleClose(expNum): home view에서 보고있는 방에 대한 notification을 닫기 위해 state를 변경한다.
- toggleSwitch() - in Device.js: 수동모드/자동모드를 조절하기 위한 state값을 변경하고 firebase에 넘겨준다.
- manual\_turtlebot(room\_number): firebase에 수동제어를 하려는 방 번호를 넘겨주고 state 값을 변경한다.
- onRadioBtnClick(rSelected): ph를 그리기 위해 선택된 방 state를 변경한다.
- push\_data(pm10\_ave, pm25\_ave): roomgraph를 그리기 위해 firebase db에서 미세먼지 수치를 불러와 data 배열에 저장한다.
- drawGraph(data): 저장된 일정량의 data를 기반으로 특정 공기질 데이터에 대해 꺾은선 그래프를 그려 보여준다.
- read\_db(): 특정 방, 시간에 대한 공기질 수치 데이터를 firebase db에서 읽어온다.

### (2) Raspberry PI 3b(Sensor)

- data.PM() : PM(1.0 , 2.5 10) 의 데이터 값을 읽고 가공한다.
- DHT.read() : 온도 및 습도의 데이터 값을 읽고 가공한다.
- mq-7.read() : 가스센서의 데이터값을 읽고 가공한다.
- update(dataset) : 각센서의 데이터값을 firebase realtime DB에 저장한다.

### (3) Crawler

- firebase\_db(): firebase realtime DB의 사용자 인증키와 DB url을 관리한다.
- load(path): firebase realtime DB에 저장된 값을 가져온다.
- update(dataset): 데이터를 firebase realtime DB에 저장한다.

- mise\_url(sidoName, pageNo): 공공데이터 포털에서 미세먼지관련 api를 가져오기 위해 서비스키와 url 정보를 관리하고 리턴한다.
- air\_info(tmp\_data): firebase DB에서 PM2.5, PM10 농도의 미세먼지, 일산화탄소 농도를 가져온다.
- get\_address(country, lat, lng, api\_key): 구글 geocode를 이용하여 현재 위치값을 가져온다.
- trans\_coordinate(latitude, longitude): 위도와 경도를 이용하여 날씨 api에서 필요로 하는 nx, ny 값을 계산해서 리턴한다.

#### (4) ROS (Turtlebot)

- db\_chk() : Firebase Realtime DB에서 실시간으로 값을 받아온다.
- sector\_chk() : 이 함수에서는 3가지를 판단한다. 요리의 유무(cooking\_mode)와 수동이동의 유무(manual\_btn) 그리고 미세먼지를 판별하여 나온 이동해야하는 위치(auto\_area)이다. 그 중에서 우선순위를 판별하여 이동한다. 순서대로 cooking\_mode와 manual\_btn 그리고 auto\_area이다. 만약 auto\_area의 값이 0이 아닌 값이 있더라도 cooking\_mode의 값이 1이라면 auto\_area의 이동을 무시하고 0번 위치인 거실(Base camp)의 위치로 돌아온다.
- pose\_msg() :ROS의 Publisher를 생성해주고 메시지를 보내주기 위해 Firebase Realtime DB에 있는 값을 받아오기 위해서 db\_chk()와 sector\_chk() 함수를 실행한다. 실행 후에 메시지를 publish하여 ROS로 보내준다.
- callbackpose(const kobot::Msg::ConstPtr& msg) : ROS에 있는 메시지를 Subscriber를 통하여 받아온다. 메시지의 내용으로는 chatter.py에서 판별한 이동해야하는 위치와 각 위치에 대한 좌표값이 있다.
- check\_sector() : 현재 위치에서 이동해야 하는 위치까지의 이동량을 구해주는 함수이다. 현재의 좌표에서 이동해야하는 위치의 좌표를 빼주고 dx, dy값에 넣어준다. 그리고 현재의 위치를 이동한 위치의 좌표로 바꿔준 후, 현재의 위치를 이동한 위치로 변경해준다.

### 3.3. 주요 함수의 흐름도

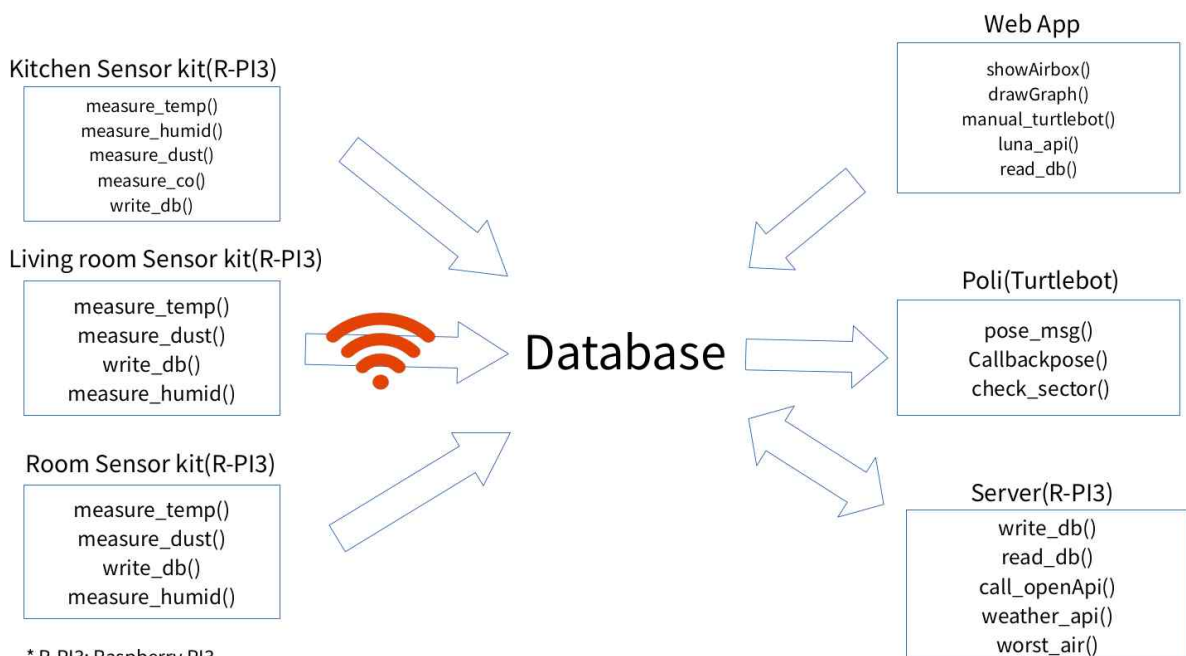


그림25. 주요 함수의 흐름도

- Sensor kit

미세먼지, 온습도, 가스센서 활성화를 시킨다. 활성화가 된 센서를 통해서 센서값을 읽어낸다. 읽은 센서값들을 Firebase Realtime DB에 다시 보낸다. 그리고 미세먼지 값을 확인하여 LED의 색을 변환시켜준다.

- Web app

Firebase Realtime DB에 누적된 데이터를 가져와 사용자에게 시각화하여 보여주며, 사용자에게 의한 수동조작을 통해 터틀봇을 원하는 위치로 이동시키는 신호를 넘겨준다. 또한 일산화탄소 수치를 수시로 체크하여 luna\_api로 알람을 띄운다.

- Turtlebot3

Turtlebot3에서는 실내 공기질 측정 키트에서 측정하여 Firebase Realtime DB에 쌓아서 판별한 데이터를 기반으로 상태를 확인한다. 상태를 확인하여 이동해야하는 위치를 판별한 후에 이동한다.

- Crawler

Open API를 이용하여 실외 공기질(미세먼지, 온도, 습도, 일산화탄소)을 측정하고 이를 DB에 저장하고 관리한다.

### 3.4. 기술적 차별성

(1) 'Air police'는 스마트홈 시스템을 총칭하며 POLI라는 스마트 공기청정봇과 각 방에 설치된 공기질 측정 센서들로 구성되어있다. Air police에는 Raspberry PI3를 와이파이를 이용하여 데이터를 DB에 넘겨 주고받을 수 있게 한다. 일반적으로 하나의 보드만을 이용하여 값을 주고 받지만, Air police는 다량의 보드를 이용함으로써 하나의 기기에서 통합하여 관리할 수 있도록 효율성을 높였다.

(2) Firebase를 이용하여 데이터를 효과적으로 관리하였다. 서버를 비효율적으로 만들면 DB관리와 각종 보드와의 연결 및 데이터 교류에 어려움이 있지만 Firebase를 이용하여 서버와 DB 관리 문제를 한 번에 해결하여 구조를 더 깔끔하게 만들었다.

(3) DB에 저장된 실내, 실외 공기질 데이터에 webOS에서 작동하는 web app이 접근하여 웹 상에서 현재 위치의 실외 공기질과 실내 공기질 데이터를 한번에 확인할 수 있다. 이에 따라 현재 위치와 측정소의 위치의 불일치로 인한 측정값의 오차를 줄이고 정확한 실내 공기질을 볼 수 있다.

(4) Turtlebot3를 이용하여 공기청정기 한 대로 집안 곳곳의 공기질을 개선해 줌으로써 경제적으로도 효율적이고 기존의 공기청정기의 이동 문제를 쉽게 해결하였다.

(5) Turtlebot3의 LIDAR센서에서 받아온 값을 Rviz에서 실내 구조를 모델링한다. 모델링한 맵을 토대로 Navigation노드를 사용하여 실내 집 내부에서 이동해야하는 위치를 자동으로 이동하도록 하였다.

(6) Luna API를 사용하여 system notification에 접근함으로써 WebOS에서만 실행이 되는 기능을 구현하도록 하였다.

### 3.5 버전관리 이력

소스코드들의 버전관리를 위해서 Git을 사용했고, 버전관리 현황은 다음과 같다.

jjolbo / Kobot8-ESWcontest Private

Branch: master

Commits on Sep 26, 2019

- ros modify  
DaeYoung96 committed 6 hours ago Verified c84f02a

Commits on Sep 25, 2019

- Update README.md  
jjolbo committed yesterday Verified 2716516
- airpolice webapp created by enact  
Myounghee96 committed yesterday Verified 37bab2b
- Create README.md  
Myounghee96 committed yesterday Verified e57c7cb
- Update README.md  
jjolbo committed yesterday Verified 2d941f7
- Update README.md  
Myounghee96 committed yesterday Verified 0ce9413
- Update README.md  
kmj-audwls committed yesterday Verified a986a5f
- move mode modify  
DaeYoung96 committed yesterday Verified 9b40ee5

Commits on Sep 24, 2019

- turtlebot3 code modify  
DaeYoung96 committed 2 days ago Verified ef78806

Commits on Sep 23, 2019

- Update README.md  
jjolbo committed 3 days ago Verified 760f61b
- Create sensor.ino ...  
kmj-audwls committed 3 days ago Verified ef17d0f
- Update README.md  
jjolbo committed 3 days ago Verified f182437
- Calculate worst air and save the area  
jjolbo committed 3 days ago Verified 0364934
- Call openAPI about weather  
jjolbo committed 3 days ago Verified 296cb02
- Translate from coordinate to nx,ny  
jjolbo committed 3 days ago Verified cd02156
- Call openAPI about fine dust  
jjolbo committed 3 days ago Verified 1ac72b1
- Call open API about fine dust  
jjolbo committed 3 days ago Verified 1980366
- Manage Firebase Realtime DB  
jjolbo committed 3 days ago Verified 148f4cf

Commits on Sep 21, 2019

<https://github.com/jjolbo/Kobot8-ESWcontest/commits/master>

그림26. Git 버전 관리 이력

#### 4. 개발 중 장애요인과 해결방안

- (1) web APP에서 공공데이터포털의 open API를 바로 호출하여 사용하려 하였지만 API 호출 과정에서 cors policy 문제로 직접 사용하지 못하였다. 이를 해결하기 위해 python으로 API를 호출하고 이를 통해 얻은 실외 미세먼지 및 날씨 정보에 대한 데이터를 Firebase Realtime DB에 업로드하여 web에서는 DB에 접근하여 데이터를 활용할 수 있도록 바꾸어 web에서의 API 호출 문제를 우회하여 해결하였다.
- (2) Navigation 노드를 실행하고 이동하는 방법으로는 마우스를 사용하는 방법 뿐이 없었다. 하지만 자동적으로 이동하는 해결방안을 찾기 위해서 ROS의 SendingSimpleGoals 라이브러리를 사용하여 마우스를 사용하지 않고도 원하는 위치로 이동하도록 해결하였다.
- (3) Turtlebot의 Raspberry Pi 내 navigation 노드 알고리즘: Turtlebot3의 Raspberry Pi에서 Navigation 노드를 사용하려고 했으나 Raspberry Pi의 성능적인 한계로 인하여 Navigation 노드를 수행하는 Rviz가 실행되지 않았다. 그래서 본래 Turtlebot3 E-manual에서 안내한 방법인 Remote-PC를 사용해 제어를 해결하였다.
- (4) Navigation 노드를 사용 중 costmap2dros transform timeout의 warning에러가 발생하여 계속 Navigation 노드의 실행을 방해하였다. 이 문제는 Turtlebot3와 Remote-PC간의 timesync가 맞지 않아서 발생한 문제이다. ntp서버의 시간을 사용하여 서로의 timesync를 맞춰주어 해결하였다.
- (5) 아두이노에서 전력부족으로 인한 센서값의 불안정 문제 및 센서 값 전달 불안정, 추가전력을 추가해줬는데도 값의 전송과정에서의 불안정 문제가 해결되지 않아서 너무 많은 추가전력이 들어가야 했다. 이러한 문제는 보드를 아두이노 우노에서 라즈베리파이로 변경을 하였다.

#### 5. 개발결과물의 차별성

- (1) 제품과의 호환성 향상 : 스마트홈 구축 장려  
LG의 다양한 제품에서 사용되는 운영체제인 WebOS의 web app 개발에 적합한 Enact와 Luna API(LS2)를 적극적으로 활용하였고 이를 통해 스마트 TV, 스마트워치 등 webOS가 탑재된 디바이스에서의 호환이 가능하다.
- (2) 각 방마다 있는 실내 공기질 측정 키트 : 현재 위치와 측정소의 측정값의 차이 해결  
미세먼지 측정소는 각 자치구마다 있지만 현재 위치한 곳과 거리가 먼 경우가 굉장히 많다. 하나의 동 단위의 동네에서도 측정값이 차이가 나기 때문에 정확한 값을 알 수 없다. 하지만 Air police 스마트홈 시스템은 실내의 각 방마다 공기질 측정 키트가 있기 때문에 정확한 수치를 알고 이를 대처할 수 있도록 한다.
- (3) 움직이는 스마트 공기청정기 "POLI" : 경제적인 문제 해결  
자율주행자동차인 Turtlebot3에 LG퓨리케어 미니공기청정기를 부착하여 스마트 공기청정기인 Poli를 개발하였다. 하나의 공기청정기만으로도 넓은 집을 관리 할 수 있어 경제적으로도 효율적이다. 또한 각 방마다 공기청정기를 설치한다면 전기세 부담과 전반적인 실내 공기질에 대한 정보를 정확히 알 수 없지만 Air police 시스템 하나만으로 각 방의 정확한 미세먼지 수치 또한 알 수 있다.
- (4) 실내 데이터 축적 및 관리  
통계에서 일별, 주별로 자료를 제시함으로써 추후 실내 공기질 관리에도 도움이 될 수 있는 데이터를 제공한다. 현재 실내 공기질의 정보를 제공하는 것에 그치지 않고, 많은 양의 데이터를 축적해놓음으로써 개인이 생활하는 실내에 맞는 공기질 관리에 기여를 할 수 있도록 하였다.
- (5) 귀여운 캐릭터 레니(RenY)  
미세먼지 농도에 따라 좋음은 파란색, 보통은 초록색, 나쁨은 노란색, 매우 나쁨은 빨간색 캐릭터로 표현하였으며 미세먼지 농도를 감시하며 상황에 따라 시각적으로 이목을 끄는 사이렌(Siren)과 비슷하여 레니(RenY)라는 이름을 고안해 내었다. Air police가 독자적으로 만든 귀여운 캐릭터인 레니(RenY)를 이용하여 미세먼지 농도에 대한 정보를 시각적으로 나타냄으로써 사용자에게 더 친근하게 다가갈 수 있도록 하였다.



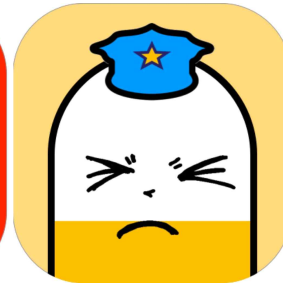
좋음



보통



매우 나쁨



나쁨

그림 27. 캐릭터 레니(RenY)

## 6. 단계별 개발계획 및 실제 참여인원 및 업무 분장

### 6.1 제작자 정보

No.	구분	성명	소속(학교)	부서(학과)	입학년도	담당업무
1	팀장	서명희	국민대학교	소프트웨어학부	2017	Web app 개발 및 WebOS, ls2
2	팀원	김명진	국민대학교	컴퓨터공학부	2015	보드 제어, 통신
3	팀원	박대영	국민대학교	컴퓨터공학부	2015	ROS 통신 및 맵 제작
4	팀원	이정하	국민대학교	소프트웨어학부	2017	Web app back-end 개발 및 DB관리

### 6.2 단계별 개발계획 및 실제 참여인원

구분	내용	이름	6월	7월	8월	9월~
조사	사전조사	전원				
	연구 결과 수집	전원				
메인보드	아두이노 이해	김명진				
	아두이노 제어	김명진				
	라즈베리파이 이해	김명진, 박대영				
	라즈베리파이 제어	김명진, 박대영				
	OpenCR보드 이해	박대영				
	OpenCR보드 제어	박대영				
센서	센서 테스트	김명진				
	센서값 분석 및 관리	김명진, 이정하				
알고리즘	실내 이산화탄소 수치 알림 및 경보 알고리즘	서명희				
	실내, 실외 데이터 측정 및 관리 알고리즘	이정하				
	DB관리 알고리즘	이정하				
	Turtlebot 이동 제어 및 관리	박대영				
Web App	Front-end 개발	서명희				
	Back-end 개발	이정하, 서명희				
하드웨어	Poli 제작	박대영				
	Airpolice 맵 제작	전원				
	실내 센서 키트 제작	김명진				
테스트	작동 확인	전원				
보고서 작성	개발 완료 보고서	전원				

### 6.3 참고 자료

- [1] 최은석, 임효선, 이진렬, 「공기청정기의 품질속성분류 및 속성별 트렌드분석 연구」, Archives of Design Research, 31(3), 69-87
- [2] 김준범 (2017), 「국내 산업 및 시도별 대기오염물질 배출량자료를 이용한 미세먼지 형성 가능성 및 인체 호흡기 영향 평가추정」, 대한환경공학회지, 39(4), p.220-228
- [3] 김승혁, 이선재, 박지용, 유민우, 장범진, 박준석, 홍경준, 장준, 윤수혁 (2018), 「미세먼지와 호흡기질환의 상관관계」, 한국환경교육학회 학술대회 자료집, p.231-235
- [4] 여민주, 김용표 (2019), 「우리나라 미세먼지 농도 추이와 고농도 발생 현황」, 한국대기환경학회지(국문), 35(2), p.249-264
- [5] 김신도 (2004), 「우리나라 미세먼지 현황 및 문제점」, 한국대기환경학회 학술대회논문집, p.41-56
- [6] ROS.org, 「Sending Goals to the Navigatio Stack」,  
<http://wiki.ros.org/navigation/Tutorials/SendingSimpleGoals>
- [7] pms 7003 datasheet , [https://eleparts.co.kr/data/goods\\_old/data/DS\\_PMS7003.pdf](https://eleparts.co.kr/data/goods_old/data/DS_PMS7003.pdf)
- [8] dht-22 datasheet ,  
<https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>
- [9] mq-7 datasheet, <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>