

SW개발/HW제작 설계서

프로젝트 명 : IoT 센서와 인공지능을 이용한 이상징후 차단
보안관리 시스템

수행 단계별 주요 산출물



환경분석	시장/ 기술 환경 분석서	○	○	○
	인터뷰 결과서	--	--	--
요구사항 분석	요구사항 정의서	○	○	○
	유즈 케이스 정의서	○	○	○
아키텍처 설계	서비스 구성도	○	○	○
	서비스 흐름도	○	○	○
	UI/UX	○	○	--
기능 설계	알고리즘 명세서	○	○	○
	하드웨어 설계도	--	--	○
	기능처리도(기능 흐름도)	○	○	○
개발 /구현	핵심 소스 코드	--	○	--

시장/ 기술 동향 분석



가택 불법 침입,
공공기관 절도등의
범죄 증가



신속한 알림 및
예방 중요성 증대



공공기관의
CCTV 설치 및
운영 대수 증가

DATA ANALYSIS



불법 침입 방지를 위한 솔루션
사례 증가



대용량 영상의 등장, 보관 난해성



각종 해킹으로 인한 보안의
중대성 인식 증가



센서의 보다 높은 정밀도 추구

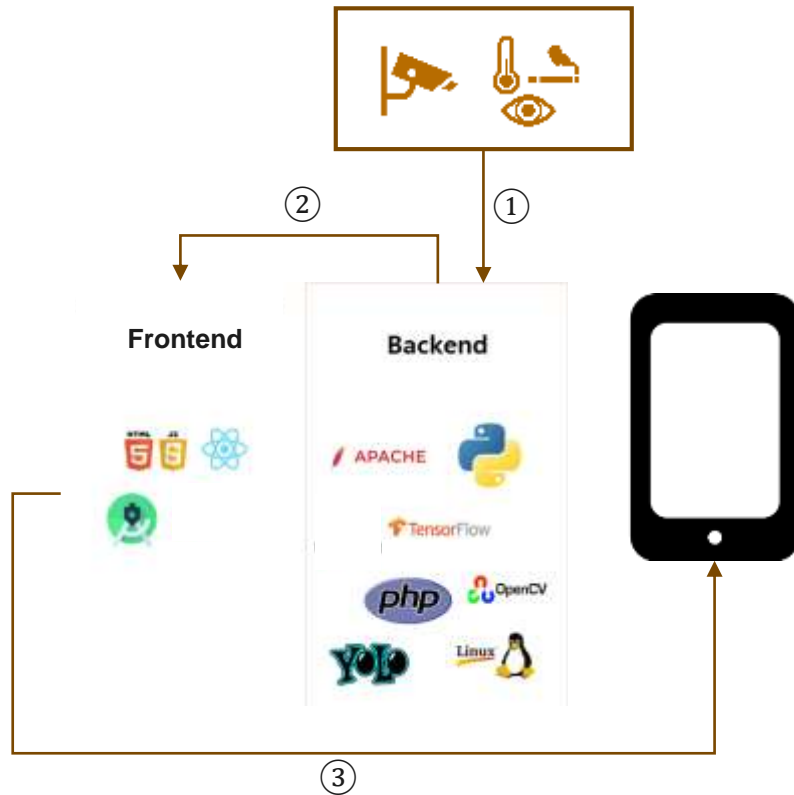
요구사항 정의서



요구사항 ID	요구사항명	기능ID	기능명	세부사항	예외사항
Fire	화재 경고 알림	fire_01	온습도 감지	범위 내 온도와 습도를 입력 받아 특정 온습도 이상이 될때 경고	특정 값 미만 탐지
		fire_02	일산화탄소 감지	범위 내 일산화탄소를 감지하여 특정 일산화 탄소 이상 값이 감지 될 경우 경고	특정 값 미만 탐지
		fire_And	화재 감지	범위내 온습도, 일산화탄소, 연기가 모두 특정 값 이상 감지되어 화재로 인식하여 관리자에게 알림	특정 값 미만 탐지
Real_time	실시간 영상 전송	rt_01	실시간 영상 전송	실시간 영상을 서버로 전송함	--
		rt_02	Edge detecting	서버로 전송 받은 실시간영상의 화질을 낮추고 edge detecting처리	--
Invasion	불법 침입 알림	inv_01	불법 침입 감지	실시간 영상에서 특정 물체가 감지될 경우 관리자에게 알림	변화가 없을 경우

서비스 흐름도

서비스 시나리오



- **Back-end**

- ① 센서/CCTV 이상징후 발견
- ② 침입/화재감지 알고리즘: 평소 측정되는 value가, 정해 놓은 threshold보다 높은 경우 화재 혹은 침입으로 판단
- ③ 현재 측정 데이터를 아파치, php로 구축해 놓은 서버로 전송

- **Front-end**

- ④ 측정된 데이터를 프론트엔드 쪽에서 처리
- ⑤ 관리자의 스마트폰 앱에서 처리된 데이터 화면으로 display

서비스 흐름도

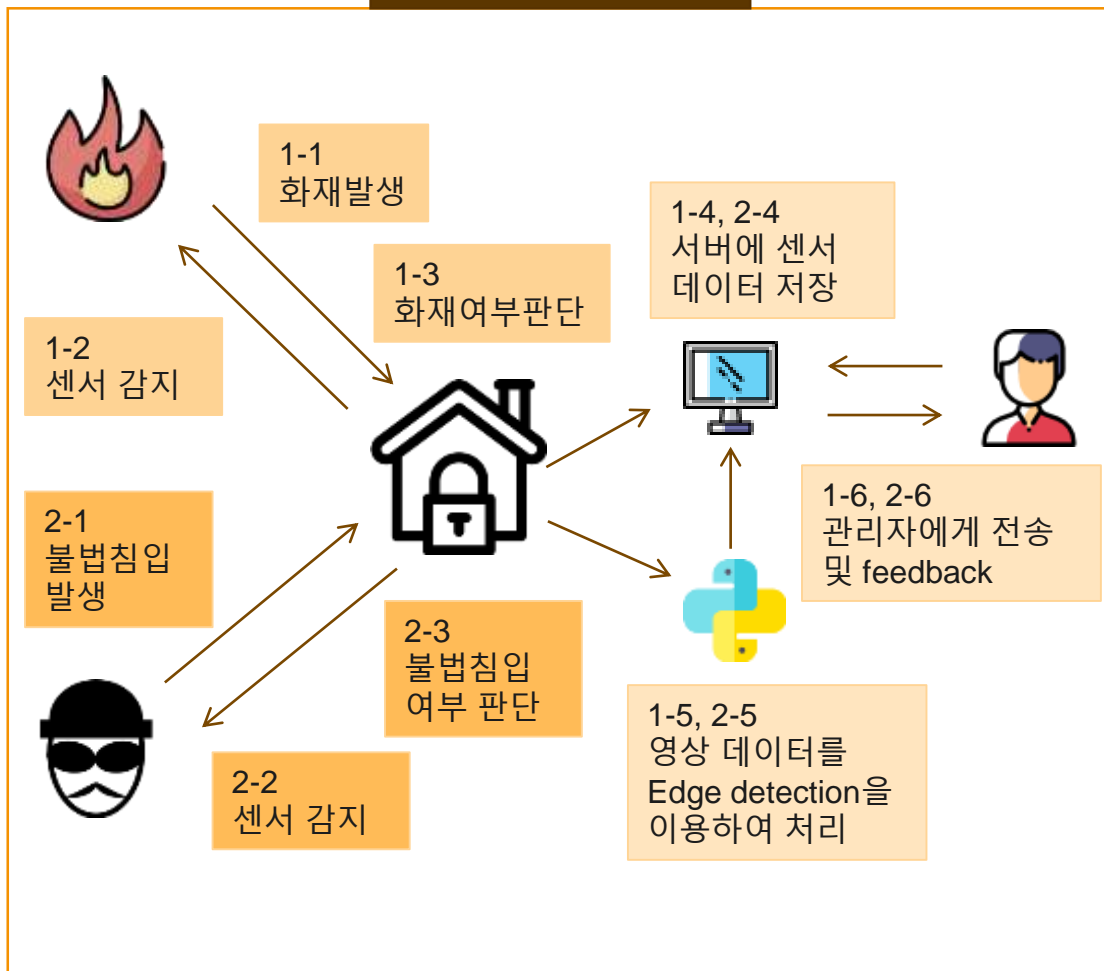
서비스 시나리오



서비스 흐름도

시나리오 별 서비스 구성 및 흐름

서비스



시나리오

* 화재발생

화재 시 발생하는 일산화탄소와 뜨거운 열기를 가스감지센서와 온,습도감지센서를 이용하여 감지한다

센서 값이 특정 임계 값을 기준으로 기준 이상이면 화재로 판단하고 기준에 못 미치면 화재로 판단하지 않으며, 측정된 센서 값들은 모두 서버에 저장된다.

만약 화재로 판단된다면 관리자에게 즉시 알림이 갈 수 있도록 하고, 관리자가 앱을 이용하여 화재발생 구역을 실시간으로 모니터링 하고 측정된 센서 값들을 확인할 수 있다.

* 불법침입발생

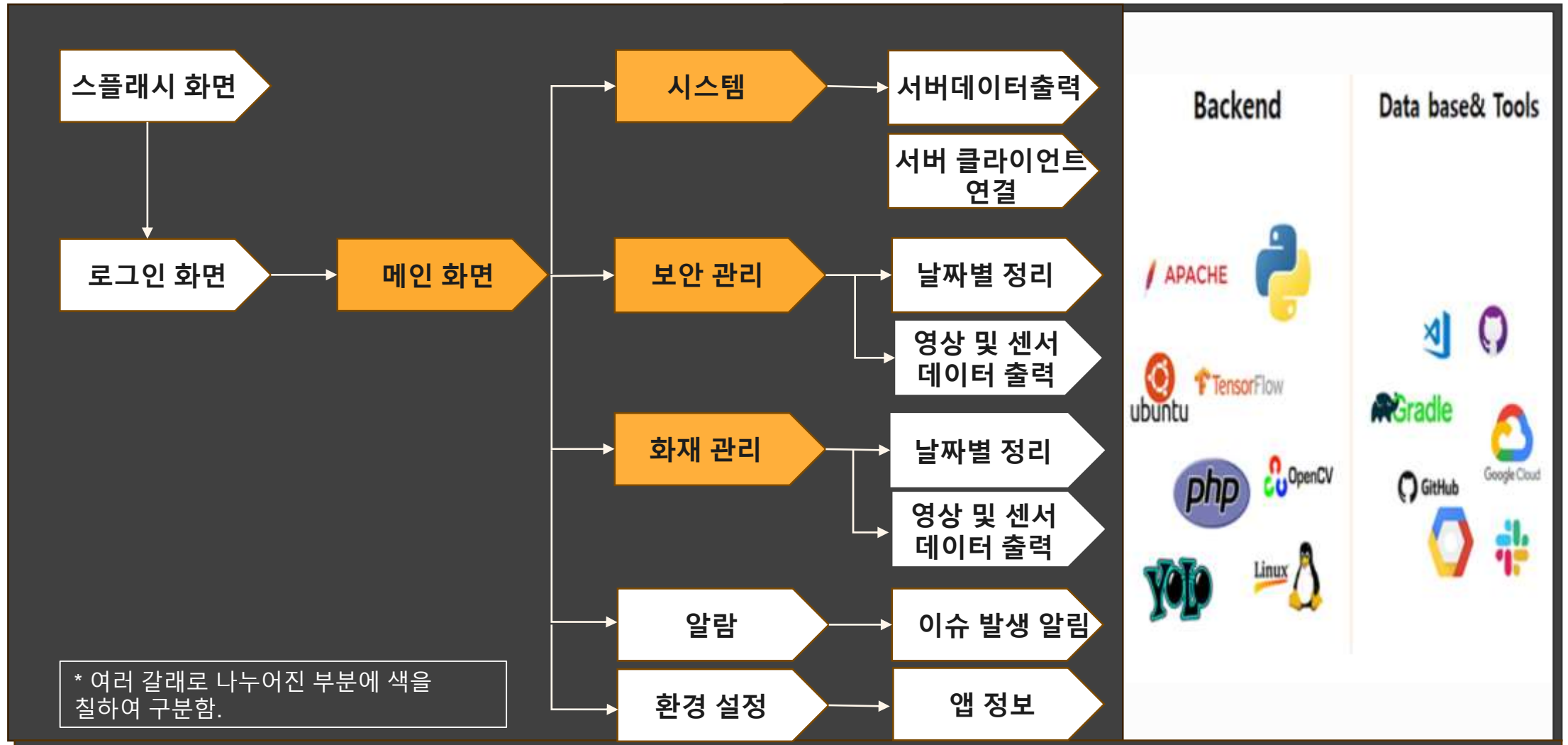
인체감지센서와 카메라를 이용하여 특정 시간 동안에 불법침입을 감지한다. 이때, 카메라로 얻은 영상 데이터의 양을 줄이고 고속 데이터 처리를 위해 edge detection을 이용하여 영상처리를 한다.

센서 값이 특정 임계 값을 기준으로 기준 이상일 때, 센서 값과 영상데이터를 종합하여 불법침입 발생 여부를 판단한다.

만약 불법침입으로 판단된다면 관리자에게 즉시 알림이 갈 수 있도록 하고 관리자가 앱을 이용하여 불법침입 발생 구역을 실시간으로 모니터링하고 측정된 센서 값들을 확인할 수 있다.

화면구성도

화면구성요소 마인드맵



화면설계서

관리자 화면 설계 및 기능설명



장소와 무관하게
관리자가 담당하는
곳에서 어떠한
이벤트가 발생했는지
실시간으로 정보를
받을 수 있음.

또한 실시간
모니터링을 기반으로
시스템 보안을
적절하게 관리 가능.

Intuition

직관적인 구성

사람들이 좀 더 편하게 사용하기 위해 단순하고 직관적인 형태로 버튼과 화면을 구성.

Real time

실시간 알림

보안 구역 내 이벤트가 발생함에 따라 실시간 알림이 동작하여 관리자에게 정보 전달. 실시간 알림은 메시지와 진동의 형태로 전달.

System

시스템 관리

실시간으로 받은 데이터를 한눈에 알아볼 수 있도록 스크롤로 정렬하여 관리자가 시스템을 관리하기에 편리하도록 설계.

manage

보안 및 화재관리

불법침입 및 화재발생 이슈에 대해 테이블 형식으로 정리하여 실시간으로 영상 및 센서 값 등의 데이터를 관리하는 업무를 할 수 있음.
또한 과거의 영상 및 센서 값 등의 데이터들을 확인할 수 있음.

화면설계서

관리자 화면 세부설명



시스템 버튼

System button

온도, 습도, 일산화탄소 농도 등의 데이터가 실시간으로 데이터베이스에 저장되어 관리자가 편리하게 시스템을 관리하도록 함.

보안 버튼

Security button

불법침입 이슈에 대해 테이블 형식으로 정리하여 실시간으로 영상 및 센서 값 등의 데이터를 관리하는 업무를 할 수 있음.
또한 과거의 영상 및 센서 값 등의 데이터들을 확인할 수 있음.

이벤트 발생 알림

Event alarm

해당 어플로 업무 중에 중요한 이벤트 발생 시, 텍스트 띄우기로 이벤트 알림을 전달.
또한 사용하고 있는 해당 기기에 이벤트 내용을 메시지로 전송.

알림 및 메시지 아이콘

Alarm & message icon

이벤트 발생 시, 우측 상단에 알림 아이콘 및 메시지 도착 아이콘 형성.

검색 및 환경설정 버튼

Search & Settings button

검색기능을 통해 원하는 날짜의 데이터를 찾을 수 있도록 함.
언어, 텍스트 크기, 알림 금지시간 설정 등 관리자 스타일에 맞게 변경 가능.

알림 버튼

Alarm button

현재 및 과거의 이슈를 날짜 별, 시간 별로 정리하여 직관적으로 정보를 얻을 수 있도록 구성.

화재 버튼

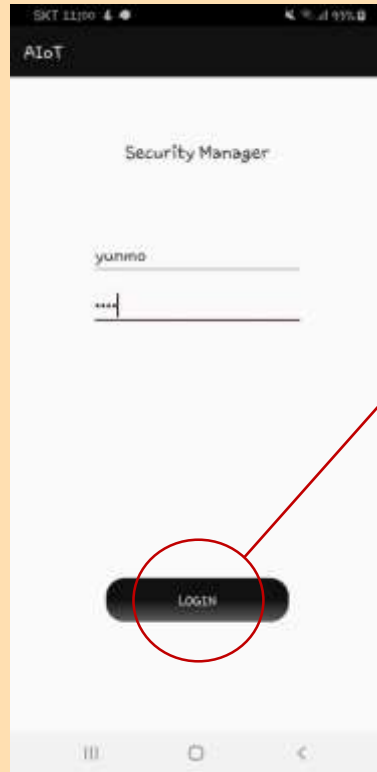
Fire button

화재발생 이슈에 대해 테이블 형식으로 정리하여 실시간으로 영상 및 센서 값 등의 데이터를 관리하는 업무를 할 수 있음.
또한 과거의 영상 및 센서 값 등의 데이터들을 확인할 수 있음.

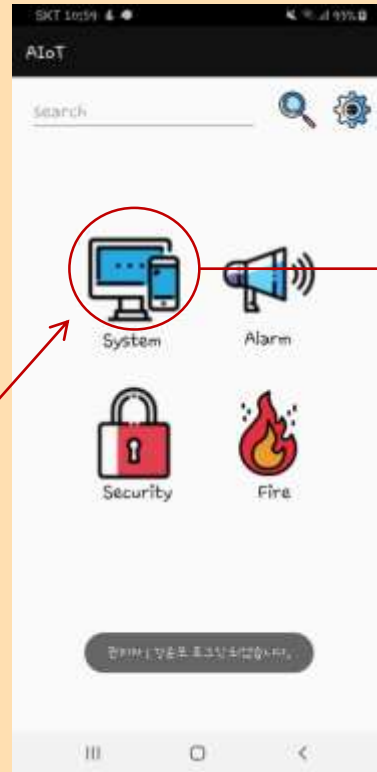


화면설계서

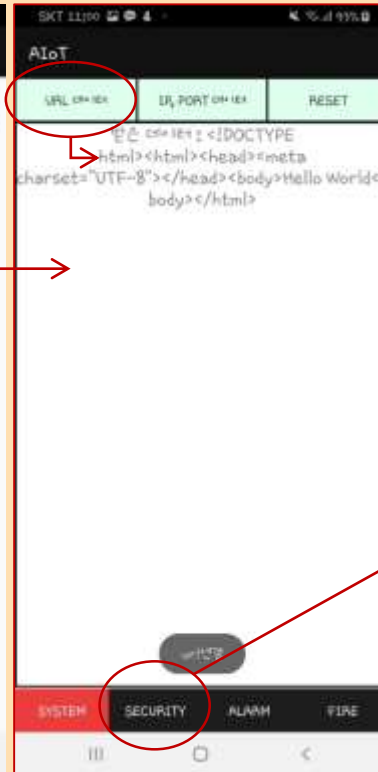
사용 예시 및 입출력 데이터



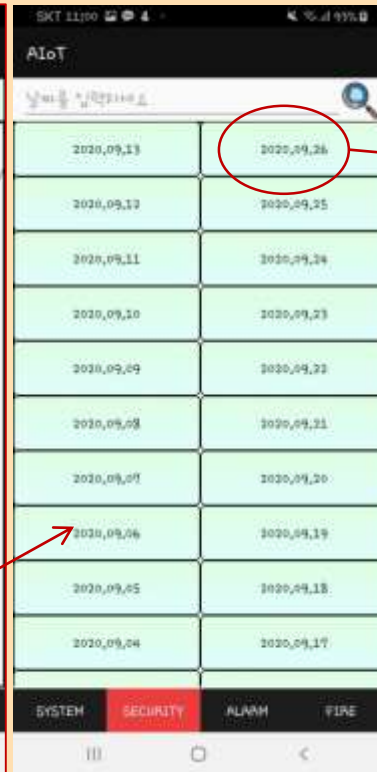
로그인 화면



메인 화면



시스템 화면(서버에서 데이터를 받은 모습)



보안관리화면(날짜별로 정리되어있음)



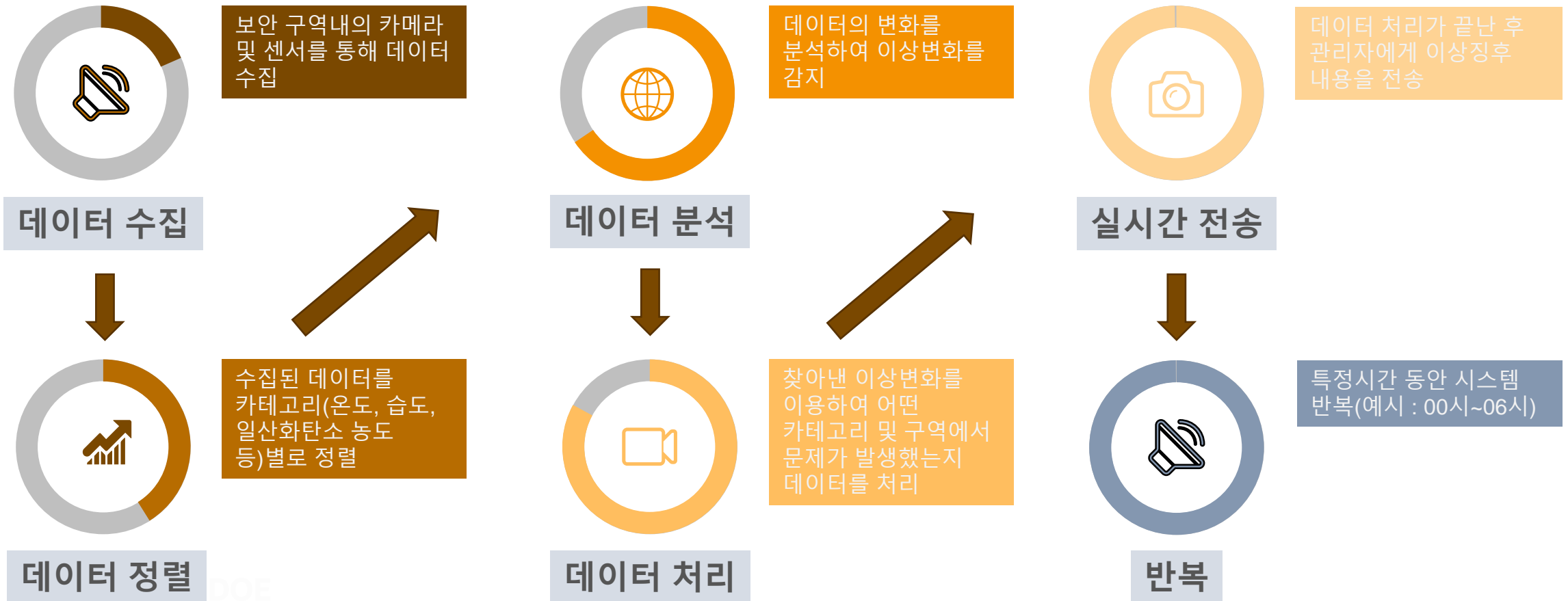
9월 26일에 해당하는 data출력 화면

기능 처리도

기능처리 구성의 알고리즘

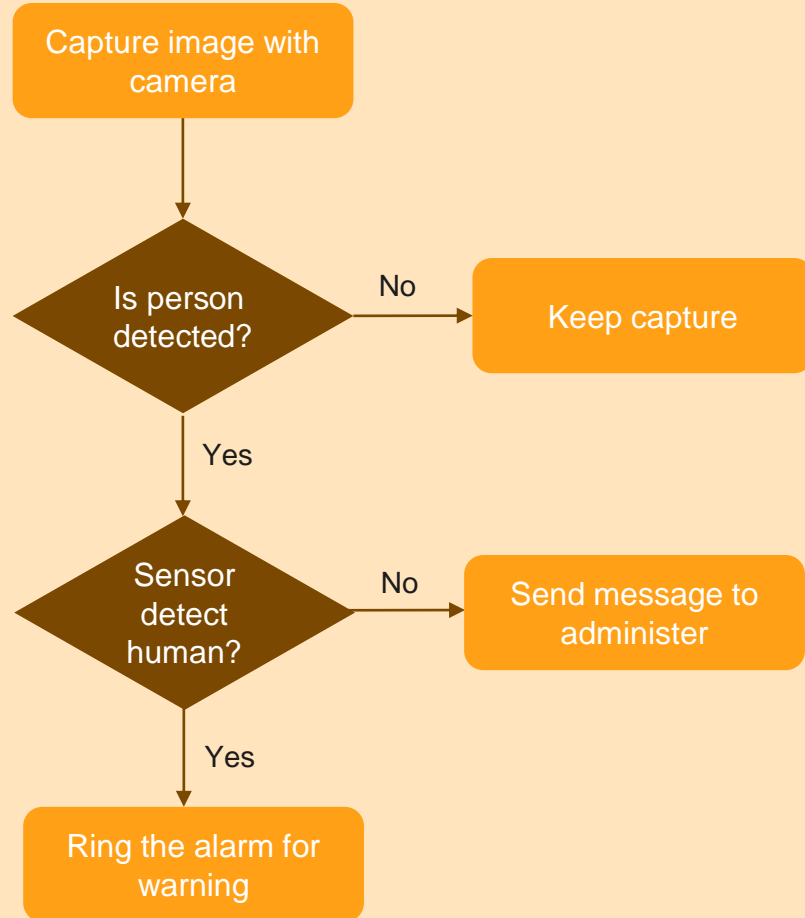


시스템 관리 (예시)



알고리즘 명세서

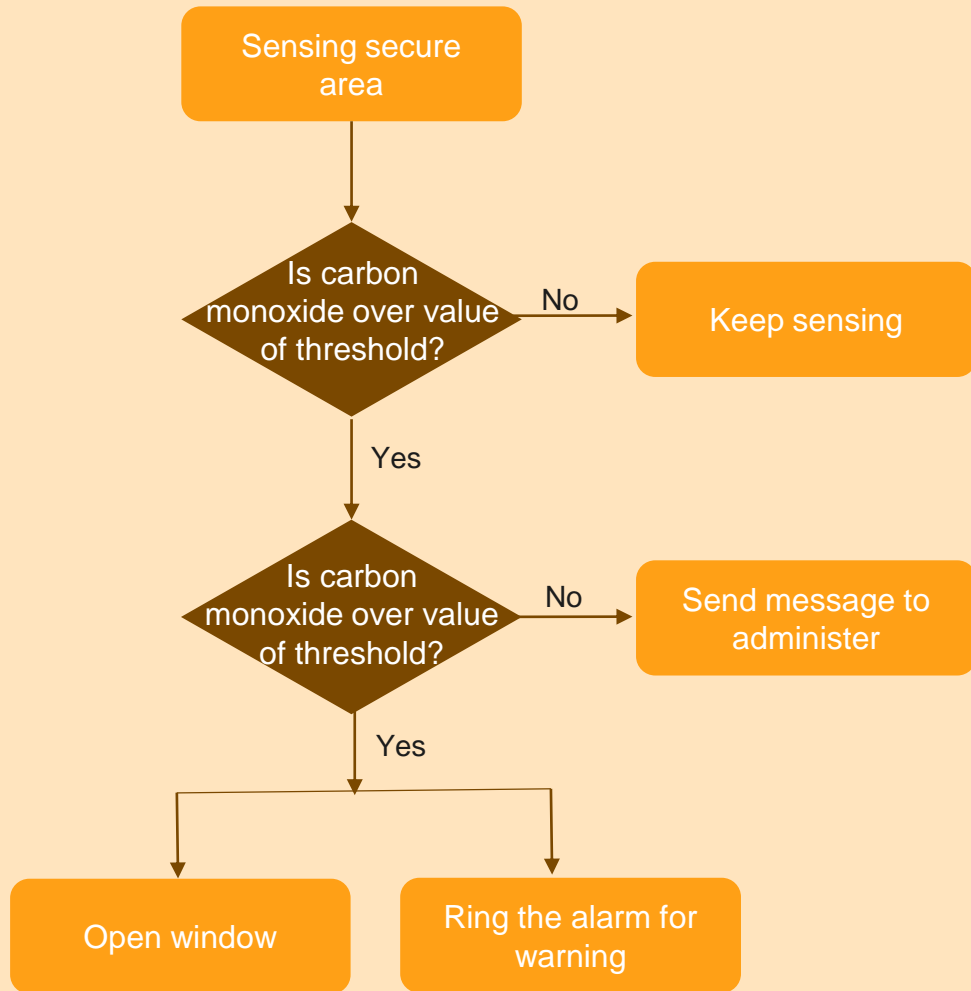
침입자 방지



1. 캠으로 보안구역 영상을 촬영한다.
2. 영상 촬영 중 사람이 포착되었는 지를 yolo deep learning detection을 이용하여 판단한다.
3. 센서에 인체가 감지 되었는지 체크
4. 센서에 감지 되지 않았을 경우, 관리자에게 침입자 의혹 알림을 보냄, 감지되었을 경우 부저를 울려 침입을 알림

알고리즘 명세서

화재감지



1. 센서로 보완구역을 센싱
2. 일산화탄소 센서를 통해 일산화탄소가 임계 값 이상으로 감지되었는지를 판단
3. 해당 구역의 온도와 습도가 임계 값을 넘었는지 확인
4. 임계 값을 넘지 않았을 경우, 관리자에게 경고 메시지를 보내고, 임계 값을 넘었을 경우, 부저와 사이키를 통해 화재가 났음을 알림

알고리즘 상세 설명서



침입자 감지

보안구역을 카메라를 이용하여 촬영해서 yolo deep learning를 통해 사람인지 여부를 판단한다. 만약 yolo에서 사람이라 판단이 되면 인체감지 센서에서 사람이 감지되었는지 확인하고 사람이 감지된 기록이 있을 경우, 부저와 사이키를 작동 시켜 침입자가 있음을 알림과 동시에 관리자에게 침입 경보를 보냄. 만약 인체 감지센서에 기록이 없을 경우, 관리자에게 침입자가 의심된다는 메시지를 보냄

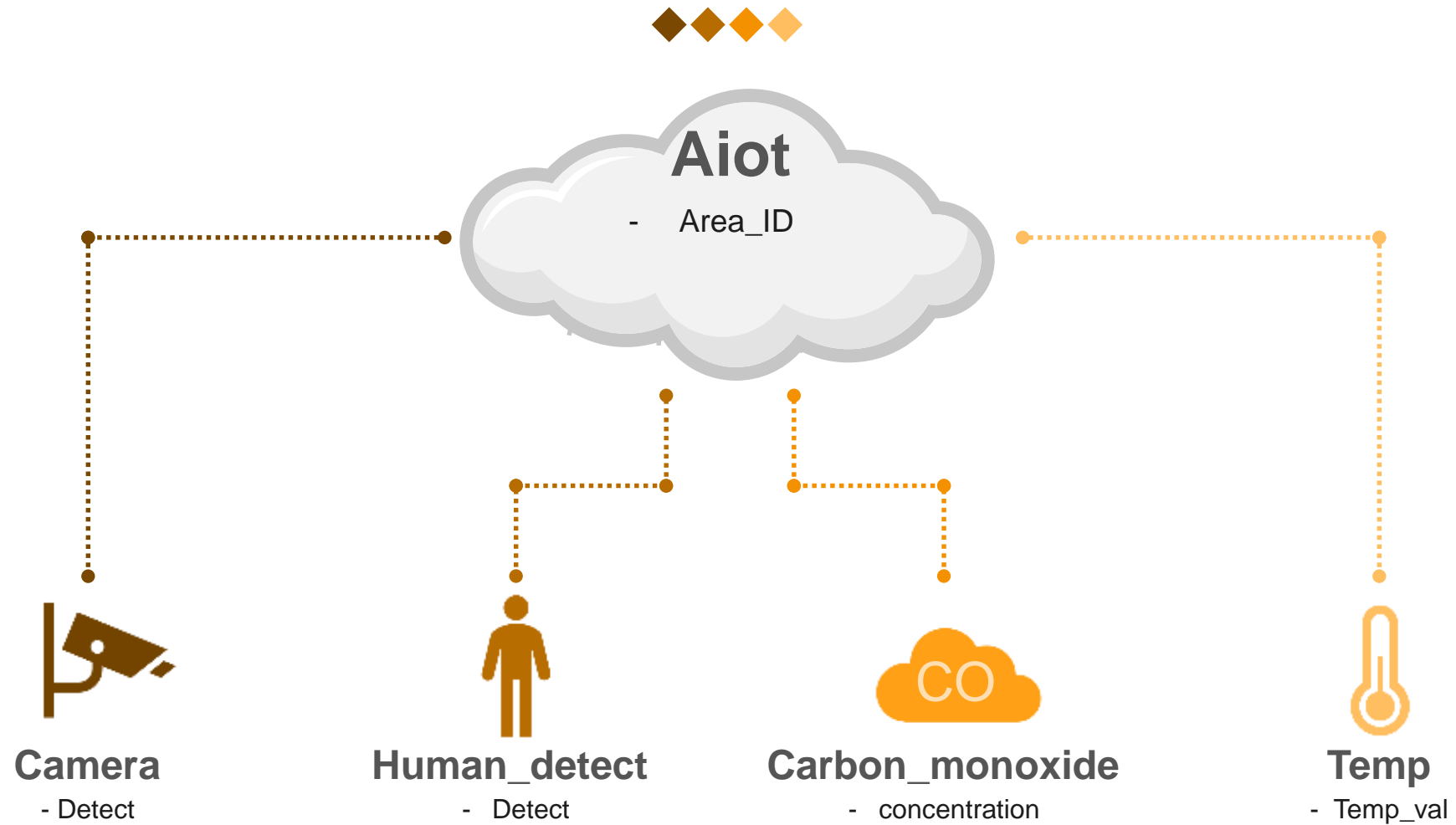


화재 감지

보안구역 내의 일산화 탄소를 측정하여 임계 값을 넘을 경우, 온습도를 체크하여 온습도 센서 값 역시 임계 값을 넘으면 화재가 났다고 판단하여 부저와 사이키를 울림으로써 화재가 났음을 알리고 동시에 관리자에게 메시지를 전송하며, 자동으로 창문을 열어 연기가 보안 구역 밖으로 빠져나갈 수 있게 한다. 온습도 센서가 임계 값을 넘지 않는 경우, 관리자에게 화재 의심 메시지를 관리자에게 보낸다.



테이블 정의서



핵심소스코드

URL을 이용한 http 통신 코드



```
/**
 * url요청을 위한 클래스 통신을 위해 AsyncTask로 구현
 */

public class NetworkTask extends AsyncTask<Void, Void, String> {

    private String url;
    private ContentValues values;

    public NetworkTask(String url, ContentValues values) {
        this.url = url;
        this.values = values;
    }

    @Override
    protected String doInBackground(Void... voids) {
        String result;
        RequestHttpURLConnection requestHttpURLConnection = new RequestHttpURLConnection();
        result = requestHttpURLConnection.request(url, values); // 해당 URL로부터 결과를 받아온다

        return result;
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);

        // doInBackground()로 부터 리턴한 값이 onPostExecute()의 매개변수로 넘어오므로 s를 출력한다.
        output.append("받은 데이터 : " + s + "\n");
    }
}
```

```
/**
 * 2. HttpURLConnection을 통해 web의 데이터를 가져온다.
 */
try{
    URL url = new URL(_url);
    urlConnection = (HttpURLConnection) url.openConnection();

    // [2-1]. urlConnection 설정.
    urlConnection.setRequestMethod("POST"); // URL 요청에 대한 포스트 설정 : POST
    urlConnection.setRequestProperty("Accept-Charset", "UTF-8"); // Accept-Charset 설정
    urlConnection.setRequestProperty("Content-type", "application/x-www-form-urlencoded; charset=UTF-8");

    // [2-2]. parameter 전달 및 데이터 읽어오기.
    String strParams = strParams.toString(); // strParams의 정리된 파라미터들을 스트림으로 저장. 예)id=1 & pw=123
    OutputStream outputStream = urlConnection.getOutputStream();
    outputStream.write(strParams.getBytes("UTF-8")); // 출력 스트림에 출력
    outputStream.flush(); // 출력 스트림을 플러시(보낸다)하고 버퍼링 된 모든 출력 바이트를 강제 실행
    outputStream.close(); // 출력 스트림을 닫고 모든 시스템 자원을 해제.

    // [2-3]. 연결 요청 확인
    // 실패 시, null을 리턴하고 예외를 종료
    if (urlConnection.getResponseCode() != HttpURLConnection.HTTP_OK)
        return null;

    // [2-4]. 읽어온 결과를 리턴.
    // 요청한 URL의 출력물을 BufferedReader로 받는다.
    BufferedReader reader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream(), "UTF-8"));
}
```

HTTP 통신을 하기 위해 AsyncTask 함수 사용
AsyncTask 함수 사용시 networktask,
doInBackground, onPostExecute 매소드 생성됨

1. NetworkTask 사용하여 어떤 url 받아올지 선정
2. doInBackground 사용하여 함수 호출하여 응답에 대한 결과를 result 변수에 넣음. url, value 값으로 requestHttpURLConnection 함수 통해 가지고옴
3. onPostExecute 통하여 실행하고자하는것 string 형식으로 받아와 output.append 함수를 통해 한줄 씩 시현

* URLconnection 안에 있 메소드 정의시 예외 발생
제외시키기 위해 try~catch 문 사용

핵심소스코드

서버에 저장된 영상 데이터 재생 코드



```
VideoView videoView;
public static String url = "http://sites.google.com/site/vbaccessmobile/sample_video.mp4";
//public static String url = "https://www.youtube.com/watch?v=Cl1cqh1B32o";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_data200926);

    Toast.makeText(getApplicationContext(), "잠시만 기다려주세요.", Toast.LENGTH_SHORT).show();
    // 직접 컨트롤할 수 있도록
    videoView = (VideoView) findViewById(R.id.videoView);

    MediaController controller = new MediaController(this);
    videoView.setMediaController(controller);
    videoView.setVideoURI(Uri.parse(url));
    videoView.requestFocus(); // 파일의 위치를 가져옴. 준비과정의 끝이나 영상줄 가져올 수 있게

    // 준비과정을 알려주는 메소드
    videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mediaPlayer) {
            Toast.makeText(getApplicationContext(), "동영상 재생준비 완료", Toast.LENGTH_SHORT).show();
        }
    });

    Button startbutton = (Button) findViewById(R.id.startbutton);
    startbutton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // 처음위치로 돌아갈수 있는거
            videoView.seekTo(0);
            videoView.start();
        }
    });
}
```

(1)영상이 등재돼 있는 링크를 'url'이라는 문자열 형태의 변수로 저장하고 (2)Controller를 새로 생성해서 영상에 대한 재생 및 일시정지가 가능하도록 해준다. 그리고 (3)각종 컨트롤을 사용자가 조작할 수 있도록 버튼을 선언해준다.

onPrepared 함수를 통해 영상이 재생되기 전까지 대기 상태 돌입.

핵심소스코드

Edge detection을 이용한 영상처리 코드



```
import numpy as np
from matplotlib import pyplot as plt
from scipy import ndimage

# input image (gray)
inimg = plt.imread('boats_512_high.png')
inimg = np.uint8(255*inimg)
plt.imshow(inimg, 'gray', vmin=0, vmax=255)
plt.show()

# =====
# # edge detection
# =====
x,y = inimg.shape
edge = np.zeros(256)
sobel1 = 1/4*np.array([[1,0,-1],
                       [2,0,-2],
                       [1,0,-1]])

sobel2 = 1/4*np.array([[-1,-2,-1],
                       [0,0,0],
                       [1,2,1]])

inimg_x = ndimage.convolve(inimg[:,:], sobel1)
inimg_y = ndimage.convolve(inimg[:,:], sobel2)
```

```
#x gradient
inimg_x = np.uint8(inimg_x)
plt.imshow(inimg_x, 'gray', vmin=0, vmax=255)
plt.show()

#y gradient
inimg_y = np.uint8(inimg_y)
plt.imshow(inimg_y, 'gray', vmin=0, vmax=255)
plt.show()

# sobel_inimg = abs(inimg_x) + abs(inimg_y)
sobel_inimg = (inimg_x**2 + inimg_y**2)**0.5
sobel_outimg = np.zeros([x,y])
threshold = 5
|
for i in range(x):
    for j in range(y):
        if (sobel_inimg[i,j] >= threshold):
            sobel_outimg[i,j] = 0
        else:
            sobel_outimg[i,j] = 255

sobel_outimg = np.float32(sobel_outimg)
plt.imshow(sobel_outimg, 'gray', vmin=0, vmax=255)
plt.show()
```

기본 python 파일로 matplotlib 라이브러리를
통해 그림 시현시킬 환경 구축
input image-> plt.imread를 통해 png 파일을
읽으며 255를 곱하여 gray scale 파일의
명암비를 설정

x축과 y 축을 통해 모형 지정하여 edge detecting.
edge 라는 변수를 설정해 초기화시켜 준다.
Sobel이라는 변수는 원 영상보다 작은 필터로서
한 픽셀씩 움직여서 값을 곱해주는 역할을 한다.
(convolution 함수를 통해 각각의 x,y 축)
X와 Y 축 한번씩 곱했으니 파일 2개가 생성되므로
각각을 제공해서 더하면 한 개의 파일이 생성된다.
(0~255 범위를 가지는 픽셀값 들을 양수를 만들어주는
기능도 있음)
Threshold 를 5로 한정해야 육안으로 보기 좋게 나오며
encoding (1or0)
값이 0 아님 1 로 나옴-> edge detecting 됨을 확인

개발 환경 및 설명



구분		항목	적용내역
S/W 개발환경	스마트폰 App 개발	Android Studio (1.2.2)	Android Studio를 이용한 application 개발
		안드로이드 OS(5.0.1)	스마트폰 운영체제
	서버 애플리케이션 개발	PHP(5.3.3)	서버 관리자 웹 페이지 처리 모듈 작성
		MySQL(5.1.73)	자율주행에 필요한 데이터를 저장, 관리하는 데이터베이스
		서버운영체제	Linux Ubuntu, Window CentOS 6.6, GCP(Google Cloud Platform)
		Putty	클라우드 서버 CentOS 원격 접속용 툴 , RSA 키 생성(Putty gen)
	GCP (Google Cloud Platform)	VM instance	가상 서버 설계 저장소, 앱과 서버 연결시 필요한 터미널 구축
		메타데이터 및 SSH	보안을 위한 RSA 키 저장
	자율주행 소프트웨어 개발	Python (3.7)	EdgeDetectiong 알고리즘 개발
		MATLAB	신호처리 프로그램 개발
	웹 서버, 데이터 교류 가시화	ATOM, HTML, CSS	서버와 클라이언트의 데이터 교류가 잘 이루어지고 있는지 보여주기 위한 중간 지점 구축
	딥러닝, 영상처리	YOLO v3, OpenCV	웹캠에서 촬영되는 실시간 영상에 물체 감지 효과를 넣음
	NAS 시스템	Portainer, Next Cloud	크기가 큰 영상 혹은 대용량 업데이트 시 빠른 파일 전송 하기 위한 추가 서버 구축

참조-개발 환경 및 설명



구분		항목	적용내역
H/W 구성장비	Sensor	온도 및 습도 센서	화재 시, 열 발생으로 인한 공기 온도 및 습도의 변화 탐지
		일산화탄소 센서	화재 시 발생하는 유독가스 중 일산화탄소의 농도 측정
		인체감지 센서	인체를 감지하여 불법침입 여부를 판단하기 위한 센서
	Device & Module	스마트폰 (안드로이드, 아이폰)	사용자에게 서비스를 직접적으로 제공하는 End Device
		카메라	센서 값과 더불어 영상 데이터를 화재 및 불법침입 등에 활용하기 위한 Device
		아두이노보드(UNO)	센서 및 기타 하드웨어 작동에 필요, 포트가 많이 필요한 경우 메가(Mega) 사용
	기타	부저	화재 및 불법침입 발생 시, 위험발생을 소리로 알리기 위해 필요
		사이키(LED)	화재 및 불법침입 발생 시, 위험발생을 시각정보로 알리기 위해 필요

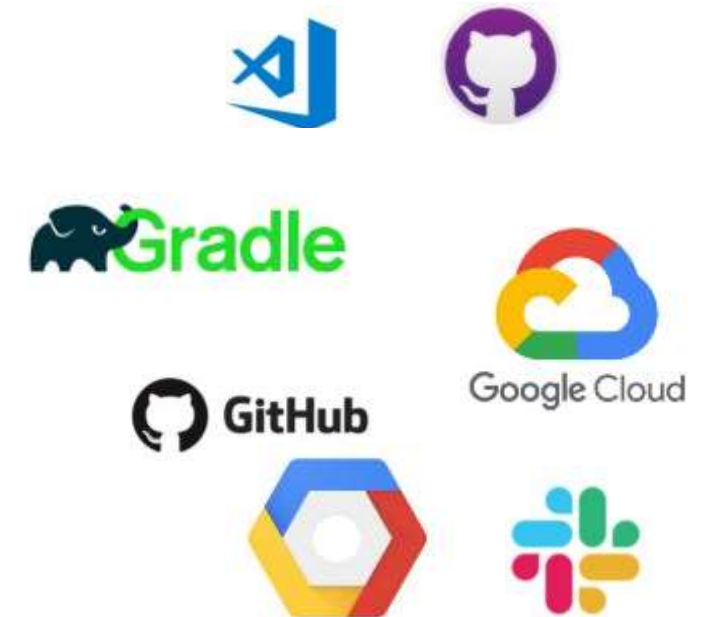
Frontend



Backend



Data base & Tools



참조 - S/W 기능 실사 사진



참조 - H/W 기능 실사 사진



<동작중인 Smart Home >



<부저>

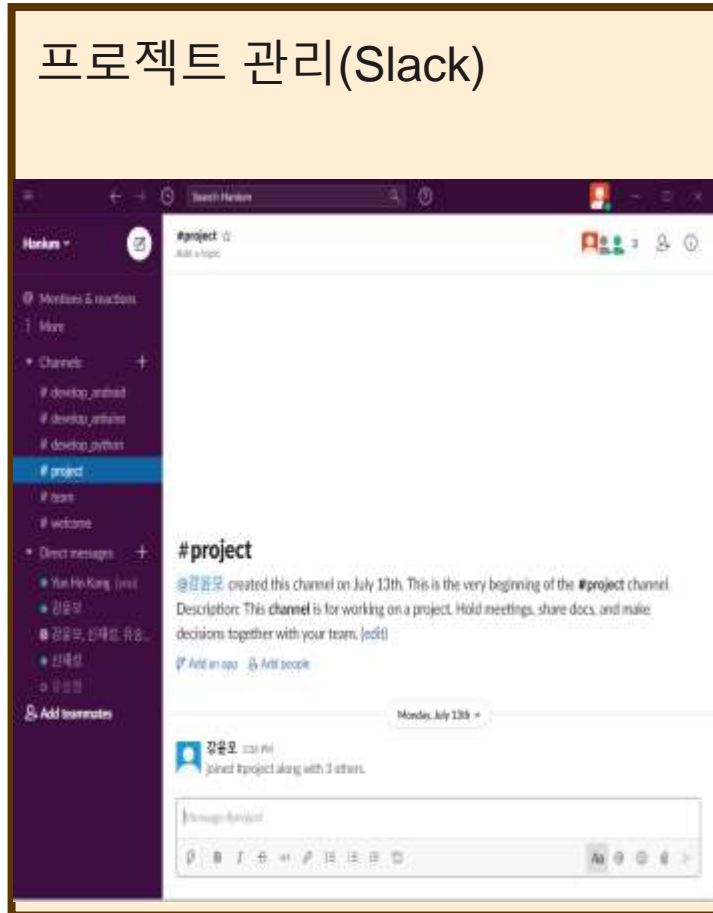


<티비근처에 부착된 웹캠>

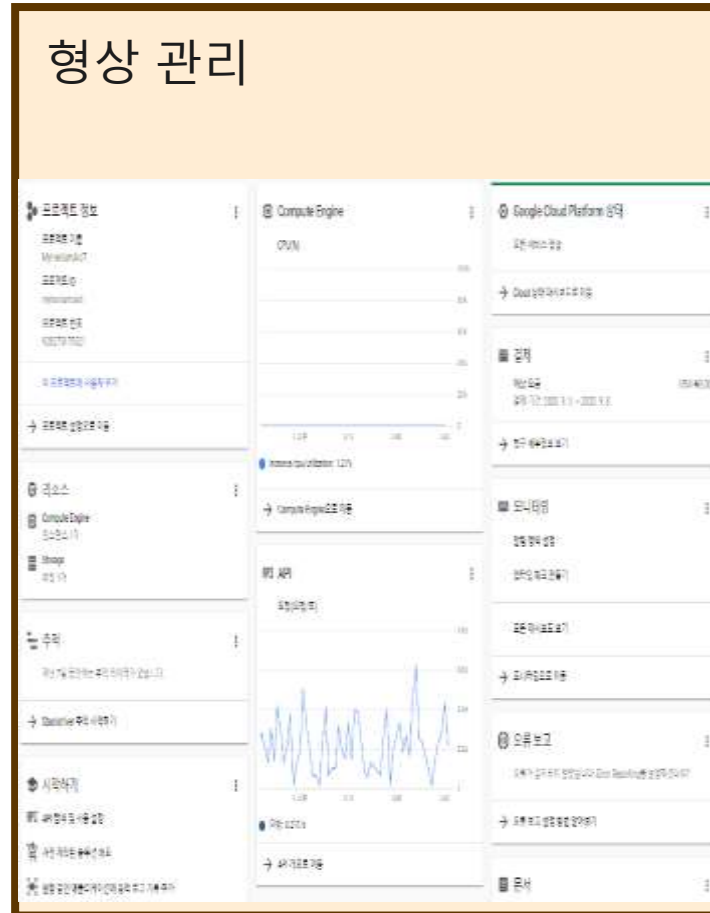
참조 - 프로젝트 관리



프로젝트 관리(Slack)



형상 관리



회의록 및 기획 관리

The screenshot shows a meeting and planning management system. The top section displays a list of meetings with columns for 'Meeting ID', 'Meeting Title', 'Meeting Date', 'Meeting Time', 'Meeting Location', and 'Meeting Status'. Below this, there is a detailed view of a specific meeting, showing the 'Meeting Details' and 'Meeting Agenda'. The 'Meeting Details' section includes a table with columns for 'Meeting ID', 'Meeting Title', 'Meeting Date', 'Meeting Time', 'Meeting Location', and 'Meeting Status'. The 'Meeting Agenda' section includes a table with columns for 'Meeting ID', 'Meeting Title', 'Meeting Date', 'Meeting Time', 'Meeting Location', and 'Meeting Status'.

Thank you

