2019학년도 제 10회 MDP 과제발표회 작품보고서

SSG
(Smart Street Generate)

M D P 명	정보통신 프로젝트 실무 네트워크 프로젝트 실무
학 과 학 년	정보통신기기과 3학년
학 생 성 명	**조승섭**, 선예은, 신동준, 정회훈, 황민혁
지 도 교 사	김진섭, 최보람, 최윤희

전자·통신분야 마이스터고등학교
인천전자마이스터고등학교
http://intec.icehs.kr/

Creative Commons

저작자표시(by)-비영리(nc)-변경금지(nd) 3.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다.

저작자표시: 귀하는 원저작자를 표시하여야 합니다.

비영리: 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지: 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들을 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

- https://creativecommons.org/licenses/by-nc-nd/3.0/deed.ko

2019학년도 제 10회 MDP 과제발표회 작품보고서
SSG

1. 연구 윤리 서약서

2019학년도 정보통신기기과 8기 학생으로서 이 작품보고서 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교사의 지도를 받아 정직하고 엄정한 연구를 수행하여 작품보고서를 작성한다.

둘째, 작품보고서 작성 시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

2019년 11월 20일

학과: 정보통신기기과
지도교사: 김진섭, 최보람, 최윤희
성명: 조승섭 선예은
신동준 정회훈
황민혁

2. 작품보고서 제출

이 보고서를 제 10회 MDP 과제발표회 SSG 작품보고서로 제출합니다.

2019년 11월

인천전자마이스터고등학교
정보통신기기과
조승섭, 선예은, 신동준, 정회훈 황민혁

이 보고서를 제 10회 MDP 과제발표회 SSG 작품보고서로 인준함.

2019년 11월

학과장: 조 우 현 (성명)
교 사: 김 진 섭 (성명)

목 차

목 차 ····· i
표 목 차 ····· ii
그림목차 ····· iii
국문요지 ····· iv

Ⅰ. 서론

1. 연구 배경 ····· 1
2. 연구 방법 ····· 1
3. 관련 전공 ····· 2

Ⅱ. 설계 및 구현

1. 기구물 ····· 3
2. 하드웨어 ····· 4
3. 소프트웨어 ····· 6
4. 서버 ····· 29
5. 웹페이지 ····· 32

Ⅲ. 결론

1. 실무결과 ····· 43
2. 시사점 ····· 44
3. 느낀점 ····· 45

Ⅳ. 활동사진 ····· 48

※ 참고문헌

표 목 차

<표 Ⅰ-1> 작품계획 ····· 2
<표 Ⅰ-2> 학과 교육과정 ····· 3
<표 Ⅰ-3> 학과 교육과정 외 관련 전공 ····· 3

그 림 목 차

[그림Ⅱ-1] 플랫폼 구성 ····· 3
[그림Ⅱ-2] 인도 설계도 ····· 3
[그림Ⅱ-3] 차도 설계도 ····· 3
[그림Ⅱ-4] 기구물 ····· 4
[그림Ⅱ-5] 하드웨어 블록도 ····· 4
[그림Ⅱ-6] 신호등 회로도 ····· 5
[그림Ⅱ-7] 번호인식 순서도 ····· 6
[그림Ⅱ-8] 구급차인식 순서도 ····· 7
[그림Ⅱ-9] 서버 상태도 ····· 8
[그림Ⅱ-10] 안드로이드 순서도 ····· 8
[그림Ⅱ-11] 서버 알고리즘 ····· 9
[그림Ⅱ-12] 디스플레이 화면 ····· 28
[그림Ⅱ-13] 로그인 화면 ····· 28
[그림Ⅱ-14] 계정 선택 화면 ····· 28
[그림Ⅱ-15] 서버 구축도 ····· 29
[그림Ⅱ-16] 비콘 사용자 테이블 ····· 29
[그림Ⅱ-17] 미세먼지 테이블 ····· 30
[그림Ⅱ-18] 불법주차 테이블 ····· 30
[그림Ⅱ-19] 사용자 정보 테이블 ····· 30
[그림Ⅱ-20] 클라우드 서비스 ····· 31
[그림Ⅱ-21] Main Page Index.html ····· 32
[그림Ⅱ-22] Main Page join.html ····· 33
[그림Ⅱ-23] Main Page login.html ····· 35
[그림Ⅱ-24] Main Page findid.html ····· 37
[그림Ⅱ-25] Main Page findpw.html ····· 37
[그림Ⅱ-26] 8조 parking.php ····· 39
[그림Ⅱ-27] 8조 dust 엑셀파일 ····· 39
[그림Ⅱ-28] 5조 실시간 영상 페이지 ····· 42
[그림Ⅳ-1] 활동사진 ····· 48

국문요지

최근 많은 기업들이 자율주행자동차 개발과 상용화에 힘쓰고 있지만 자동차의 기반이 되는 도로의 개발에 대한 관심은 부족한 상황입니다. 그래서 저희는 도로에 자율사물 기능을 구축한 SmartRoad를 주제로 잡아 사람들이 더욱 더 안전하고 편리하게 이용할 수 있도록 하는 도로의 플랫폼을 구축하기로 하였습니다. 물체 인식을 이용한 지능형 교통신호를 구현하여 구급차가 인식되면 구급차가 빠르게 사고 현장에 도착할 수 있도록 상황에 알맞은 신호로 변환시킵니다. 번호 인식 기능을 이용하여 특정 구역의 불법주차가 감지되면 웹과 디스플레이에서 확인할 수 있습니다. 비콘을 이용한 미아 찾기 시스템은 안드로이드 앱을 통해 비콘이 인식되면 인식 데이터를 통해 이동 경로를 생성하여 실시간 경로를 확인할 수 있으며, 디스플레이에서 실종아이를 확인할 수 있습니다. 공개 API를 통해 미세먼지 데이터에 대한 빅데이터를 구축하고 데이터통계를 내어 여러 방법으로 활용할 수 있도록 미세먼지 기능을 구현했습니다. 모든 것은 TCP/IP통신과 소켓통신으로 구현하였습니다. 또한 웹과 앱을 통해 작품에 쉽게 접근할 수 있도록 하였습니다. 이 작품을 통해 더욱 편리하고 쾌적한 도로를 기대할 수 있습니다.

Ⅰ. 서론

1 연구 배경

최근 여러 기업들은 2020년 자율주행자동차의 상용화를 목표로 힘쓰고 있습니다. 자율주행자동차는 가장 주목하는 IT 기술인 자율사물과 자동차를 접목한 기술입니다. 하지만 자동차의 기반이 되는 도로의 개발은 부족한 상황입니다. 그래서 도로에 자율사물 기능을 구축한 SmartRoad를 주제로 잡아 사람들이 더욱 더 안전하고 편리하게 이용할 수 있도록 하는 도로의 플랫폼을 구축하기로 하였습니다.

2 연구 방법

〈표 Ⅰ-1〉 작품계획

내 용	4월				5월				6월				7월				8월				9월				10월				11월			
프로젝트 주제 선정 및 상세계획																																
아이디어 구체화, 무품선정, 역할 배분																																
DB서버, 웹서버 기반 다지기																																
하드웨어 프로토타입 모델 제작																																
하드웨어 설계 및 개발, 펌웨어 개발																																
번호인식, 영상인식 구현																																
앱개발																																
외관제작																																
SW/HW 연동 테스트																																
전체 디버깅, 최적화																																

3 관련 전공

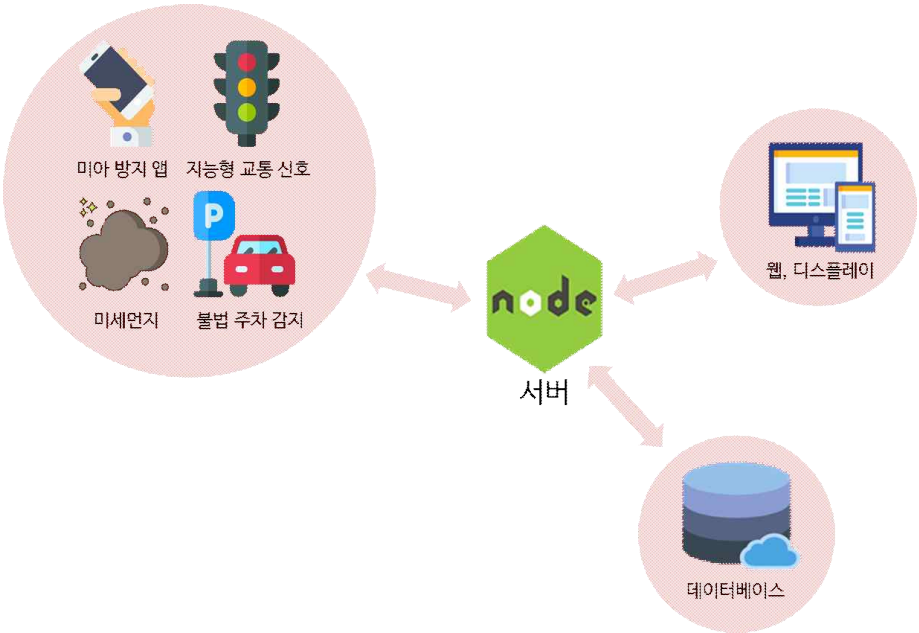
〈표 Ⅰ-2〉 학과 교육과정

학년	과 목		내 용	비고
1	전자 통신 기초		회로 설계에 대한 기초	
	프로그래밍(JAVA)		C언어 및 JAVA의 기초	
2	마이프로프로세서 기초	MPU	CodeVision을 활용한 ATmega128a 펌웨어 개발	
		Android	Android Studio를 활용한 앱개발	
	정보통신	Packet Tracer	OSI 7계층 및 TCP/IP 4계층을 비롯한 네트워크에 대한 전반적인 내용	
	정보통신 기기 설계		회로 설계 및 PCB 제작	
3	정보통신기기 소프트웨어 개발		APM을 통한 웹 제작 및 Android 앱과의 연동 JQuery, JavaScrpit를 통해 웹 프로그램 제작	
	네트워크 구축		정보보호와 암호 및 문제해결과 통신보안	
	마이크로프로세서 제어		CodeVision을 활용한 ATmega128a 펌웨어 개발	
	정보통신기기		프로젝트를 통한 작품 제작	

〈표 Ⅰ-3〉 학과교육과정 외 관련 전공

제 목	내 용	비고
Python	이미지 인식과 번호인식 학습에 사용데이터 분석, 이미지 소켓통신에 활용	
Node.js	서버 구축에 활용	

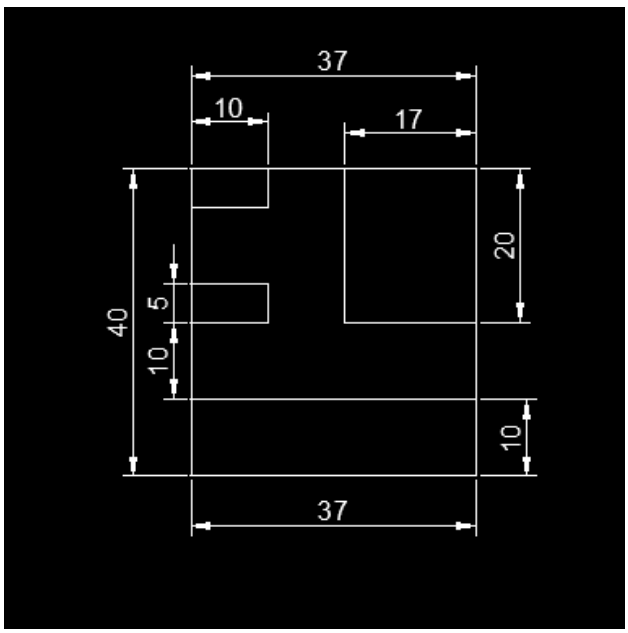
Ⅱ. 설계 및 구현



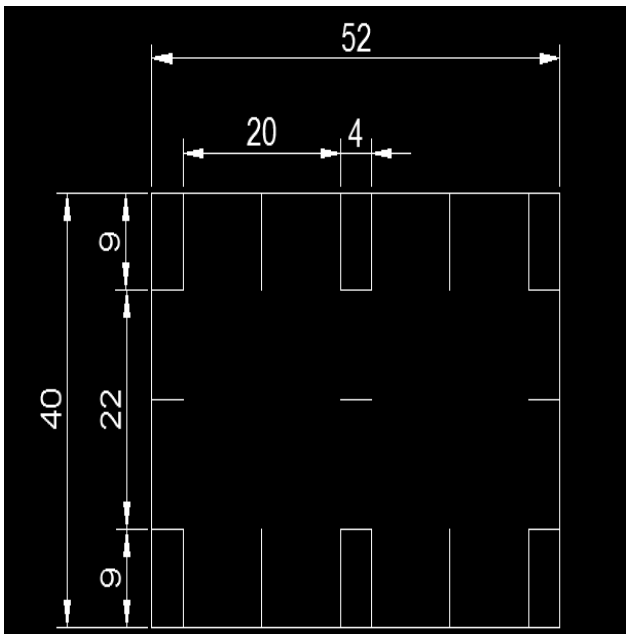
[그림Ⅱ-1]플랫폼 구성

1 기구물

1. 설계도



[그림Ⅱ-2]인도 설계도



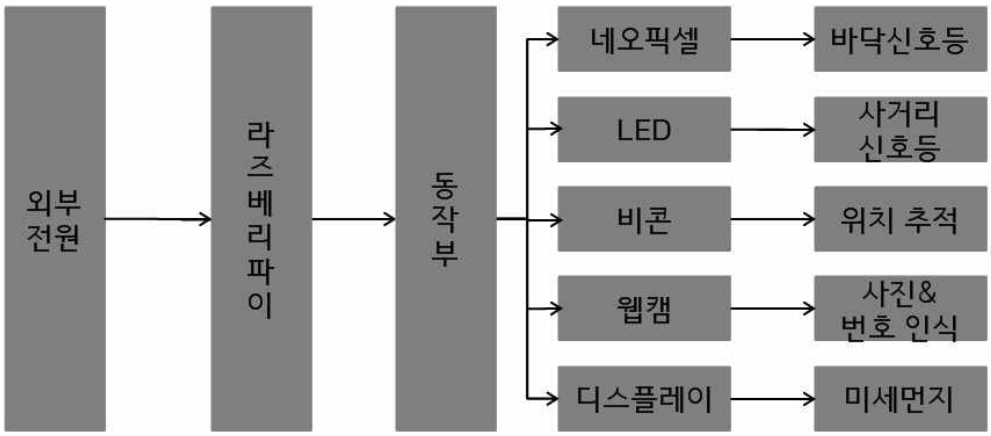
[그림Ⅱ-3]차도 설계도

2. 기구물 사진

[그림Ⅱ-4] 기구물

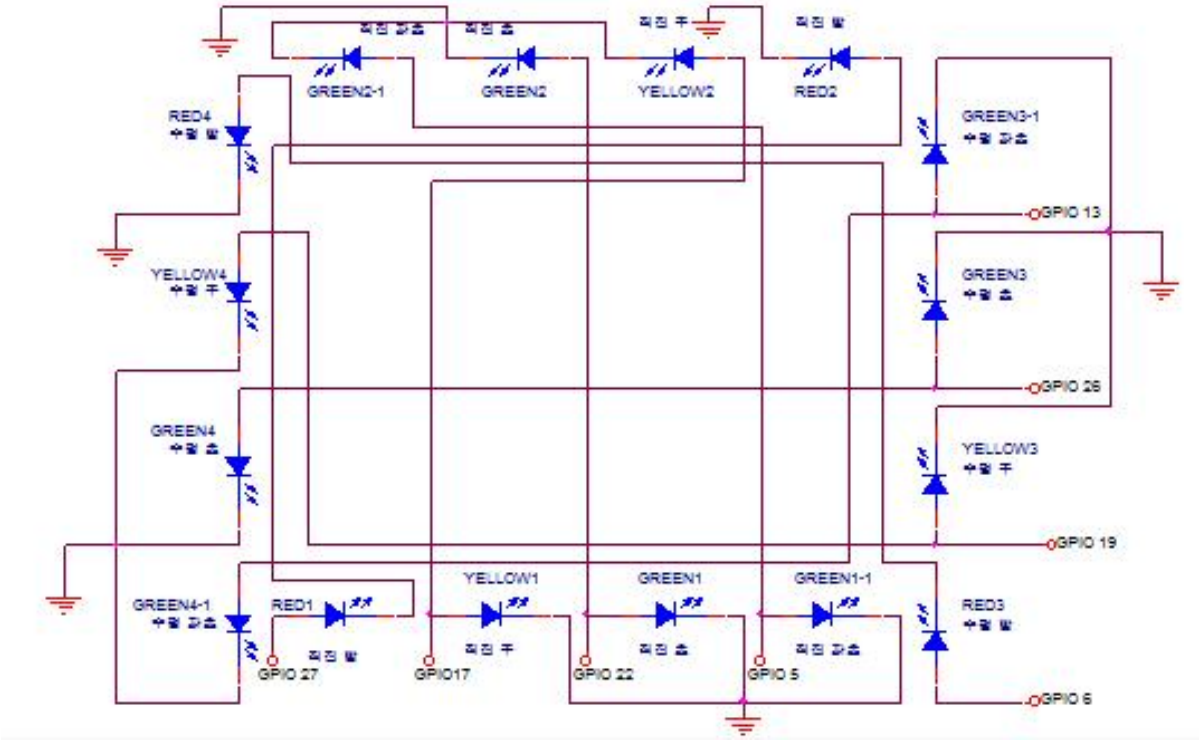
2 하드웨어

1. 블록도



[그림Ⅱ-5]하드웨어 블록도

2. 회로도



[그림Ⅱ-6]신호등 회로도

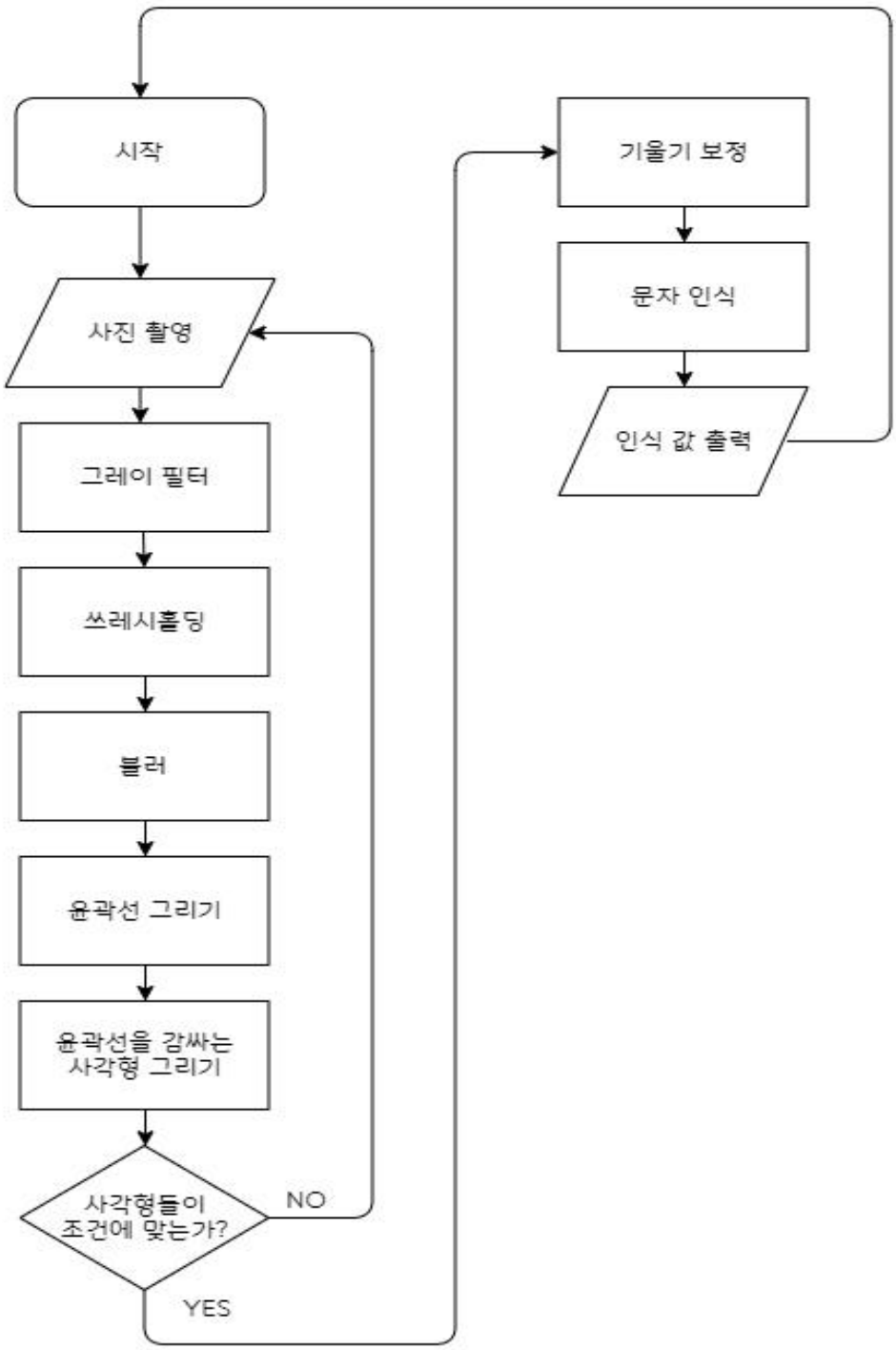
3. 사용된 모듈

NeoPixel : 네오픽셀 WS2812

Beacon : HM-10 블루투스 4.0

3 소프트웨어

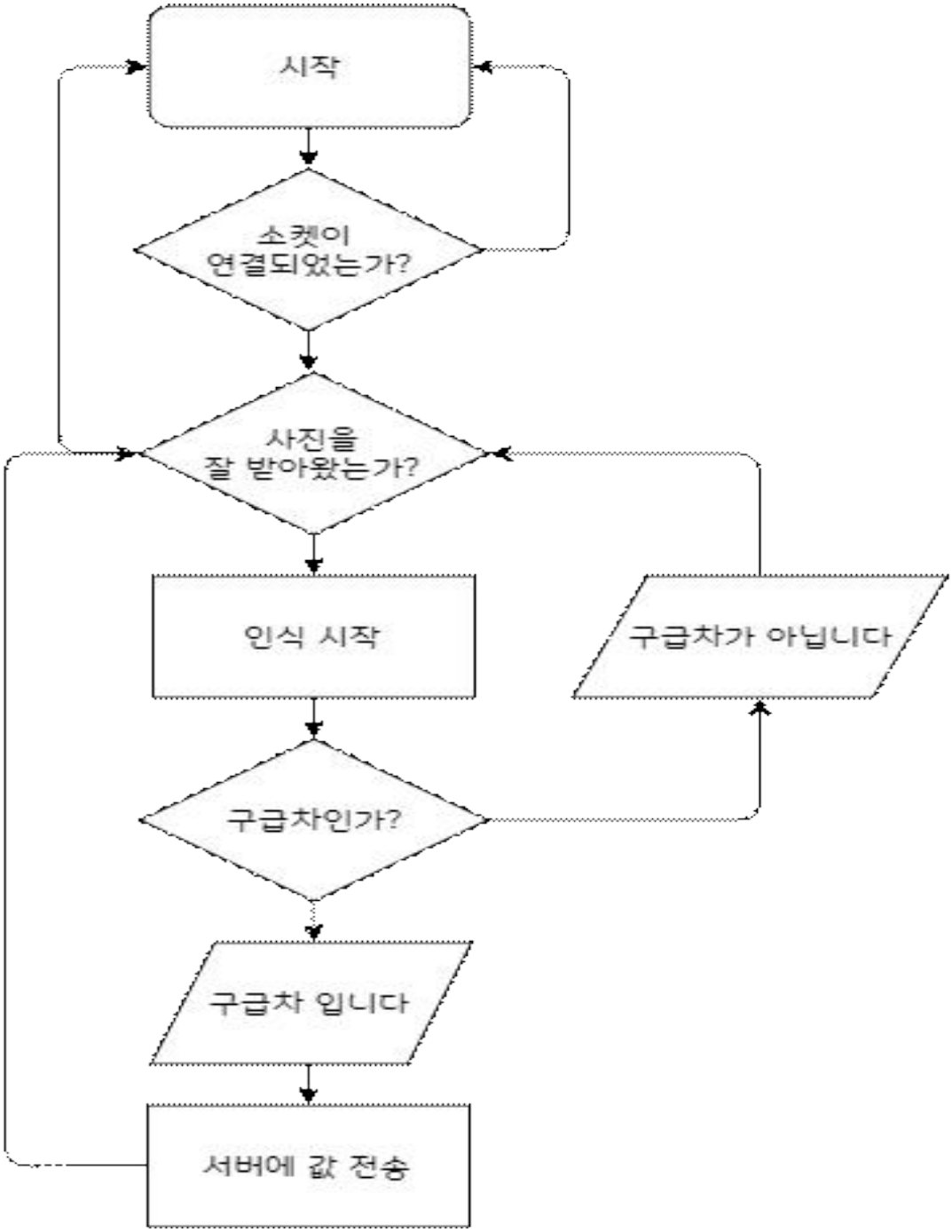
1. 알고리즘



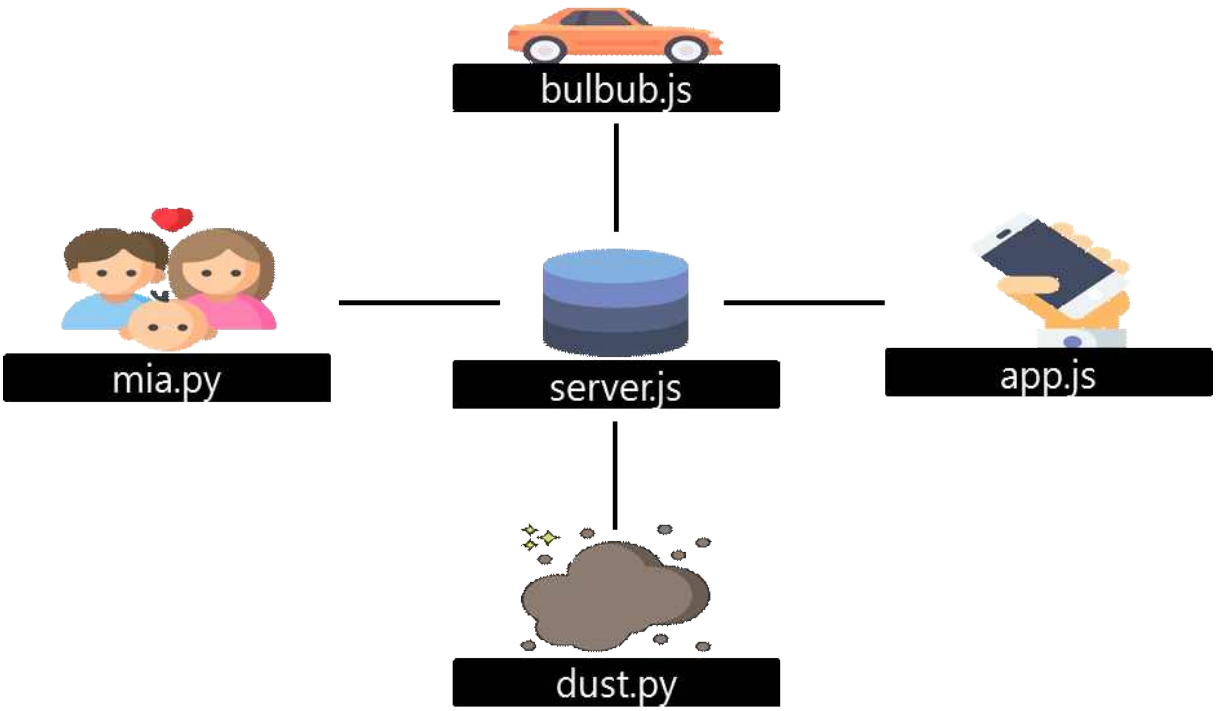
조건 :

사각형들의 너비
사각형들의 높이
사각형 크기의 최소값
사각형 크기의 최대값

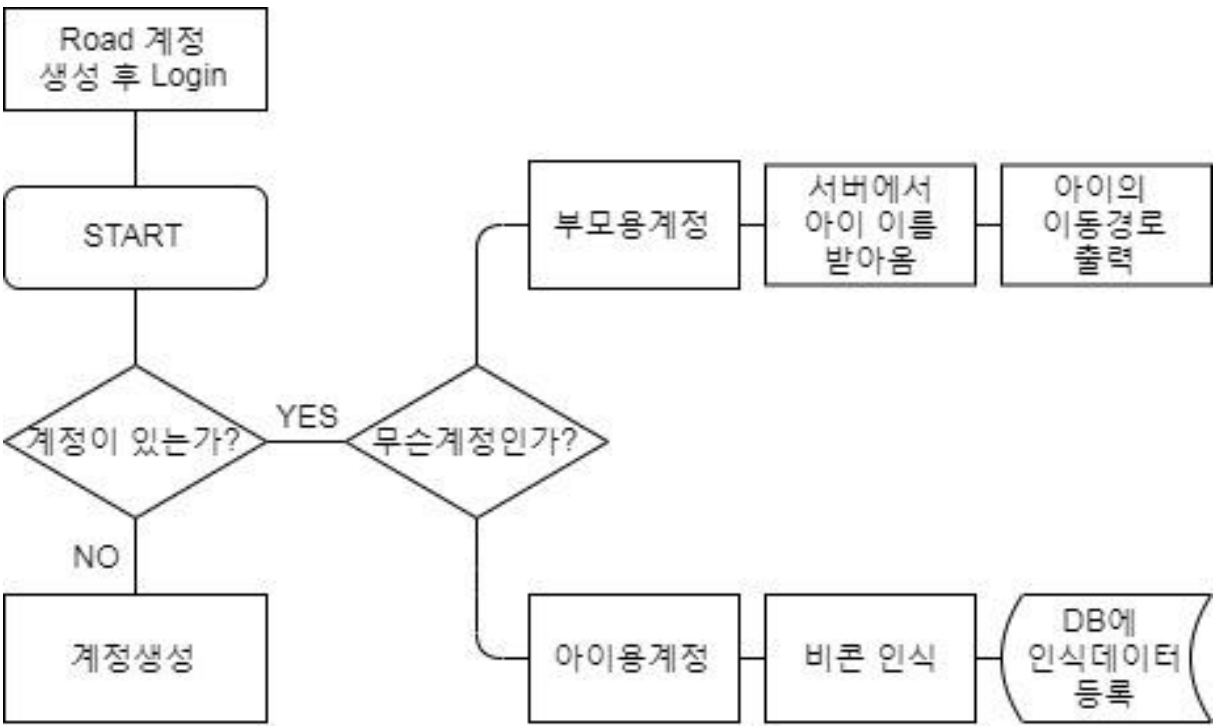
[그림Ⅱ-7]번호인식 순서도



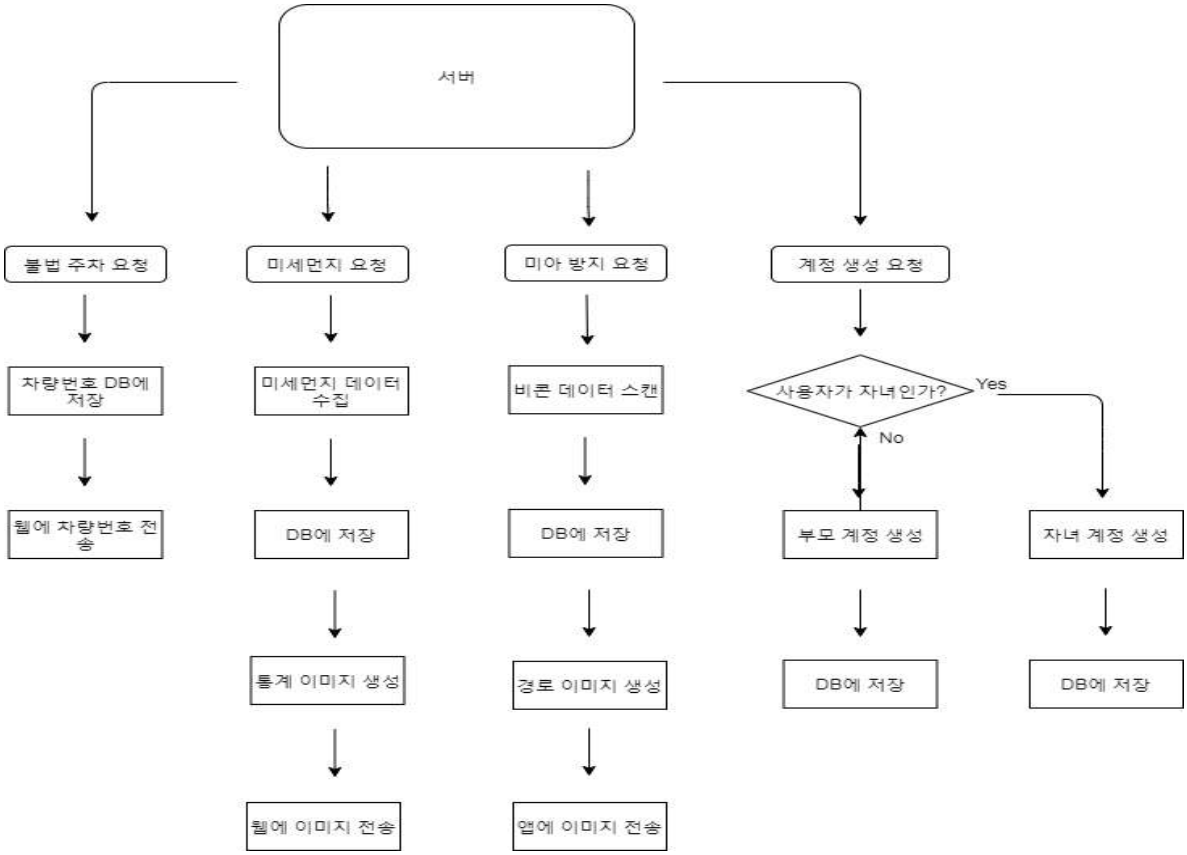
[그림Ⅱ-8]구급차인식 순서도



[그림Ⅱ-9]서버 상태도



[그림Ⅱ-10]안드로이드 순서도



[그림Ⅱ-11]서버 알고리즘

3. 코드

JAVA : JAVA가 기반인 안드로이드 스튜디오에 활용

HTML : 웹 문서를 만들기 위하여 사용하는 기본적인 웹 언어의 한 종류
PHP : HTML과 SQL의 연동을 위함. 정적언어인 HTML에 동적인 요소를 생성할 수 있는 서버측 스크립트 언어이자 범용 프로그래밍 언어
SQL : 데이터베이스에 접근할 수 있는 데이터베이스 하부 언어로 데이터베이스를 구축, 개발할 수 있음
CSS : 웹 문서의 전반적인 스타일을 미리 저장해 둔 스타일시트
JavaScript : 객체지향 스크립트 언어로 웹페이지의 동작을 담당

Python : 객체지향적인 프로그래밍 언어로 데이터 분석과 이미지 소켓통신을 위해 사용함

구급차 인식 함수

```
import numpy as np #라이브러리 임포트
import os
import threading
```

```
import tensorflow as tf

from matplotlib import pyplot as plt
from PIL import Image
from PIL import ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True
from object_detection.utils import ops as utils_ops

from utils import label_map_util
from utils import visualization_utils as vis_util

from socket import *
from sys import exit
import socket
HOST = '192.168.0.9' #서버와 연결할 소켓
PORT = 8602
so = socket.socket(AF_INET, SOCK_STREAM)
def amb(): #소스 전체를 임포트 하기 위해 함수로 묶어줌
    PATH_TO_CKPT  =  'D:/object/export_dir/faster_rcnn_inception_v2_coco_2018_01_
28/frozen_inference_graph.pb' #내가 교육한 모델
    PATH_TO_LABELS = os.path.join('D:/object/label_map.pbtxt')
    NUM_CLASSES = 3 #SUV, 세단, 구급차
    detection_graph = tf.Graph()
    with detection_graph.as_default(): #그래프에 노드 추가
      od_graph_def = tf.GraphDef()
      with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid: #모델을 불러옴
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')
    label_map = label_map_util.load_labelmap(PATH_TO_LABELS) #라벨을 불러옴
    categories=label_map_util.convert_label_map_to_categories(label_map, max_num_cl
asses=NUM_CLASSES, use_display_name=True)
    category_index = label_map_util.create_category_index(categories)
    def load_image_into_numpy_array(image): #사진을 넘파이 배열로 불러옴
      (im_width, im_height) = image.size
      return np.array(image.getdata()).reshape(
        (im_height, im_width, 3)).astype(np.uint8)
    try: #사진이 들어오면
      so.connect((HOST, PORT)) #소켓 연결
      PATH_TO_TEST_IMAGES_DIR='D:/deeplearning/models-master/research/object_d
etection/test_images'
      #TEST_IMAGE_PATHS = os.listdir(PATH_TO_TEST_IMAGES_DIR)
      TEST_IMAGE_PATHS = [os.path.join(PATH_TO_TEST_IMAGES_DIR, 'image.jpg')]
     #image.jpg를 불러온다
      IMAGE_SIZE = (12, 8)
      def run_inference_for_single_image(image, graph): #불러온 이미지에 대해 실행
        with graph.as_default():
          with tf.Session() as sess:
            # Get handles to input and output tensors
            ops = tf.get_default_graph().get_operations()
            all_tensor_names = {output.name for op in ops for output in op.outputs}
            tensor_dict = {}
            for key in [ #각 키 값에 대해서
              'num_detections', 'detection_boxes', 'detection_scores',
```

```
                'detection_classes', 'detection_masks' ]: #이름을 지정
                tensor_name = key + ':0'
                if tensor_name in all_tensor_names:
                  tensor_dict[key]=tf.get_default_graph().get_tensor_by_name(tensor_na
me)
                if 'detection_masks' in tensor_dict:
                  # The following processing is only for single image
                  detection_boxes = tf.squeeze(tensor_dict['detection_boxes'], [0])
                  #사이즈가 1인 차원을 제거
                  detection_masks = tf.squeeze(tensor_dict['detection_masks'], [0])
                  real_num_detection = tf.cast(tensor_dict['num_detections'][0], tf.int32)
                  #새로운 형태로 바꾸어줌
                  detection_boxes=tf.slice(detection_boxes,[0,0],[real_num_detection, -1])
                  #특정 부분 추출
               detection_masks=tf.slice(detection_masks,[0,0,0],[real_num_detection,-1,  -
1])
               detection_masks_reframed = utils_ops.reframe_box_masks_to_image_mask
s(detectio                n_masks,detection_boxes, image.shape[0], image.shape[1])
               detection_masks_reframed=tf.cast(tf.greater(detection_masks_reframed,
0.5),
                                                tf.uint8)
               tensor_dict['detection_masks'] = tf.expand_dims(detection_masks_reframe
d, 0)
               image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:
0')
               output_dict=sess.run(tensor_dict,feed_dict={image_tensor:np.expand_dims
(image,0)})
               output_dict['num_detections']=int(output_dict['num_detections'][0])
               output_dict['detection_classes']=output_dict['detection_classes'][0].astype
(np.uint8)
               output_dict['detection_boxes']=output_dict['detection_boxes'][0]
               output_dict['detection_scores']=output_dict['detection_scores'][0]
                if 'detection_masks' in output_dict:output_dict['detection_masks'] = out
put_dict['d                  etection_masks'][0]
                 return output_dict
         for image_path in TEST_IMAGE_PATHS: #이미지에 대해 실행
           image = Image.open(image_path) #이미지를 불러옴
           image_np = load_image_into_numpy_array(image) #이미지를 넘파이 배열로 바꿈
           image_np_expanded = np.expand_dims(image_np, axis=0)
           output_dict = run_inference_for_single_image(image_np, detection_graph)
           vis_util.visualize_boxes_and_labels_on_image_array(
             image_np, #이미지에 대해서
             output_dict['detection_boxes'], #테두리 상자
             output_dict['detection_classes'], #추측한 종류
             output_dict['detection_scores'], #정확도
             category_index,
             instance_masks=output_dict.get('detection_masks'),
             use_normalized_coordinates=True,
             line_thickness=8)
           plt.figure(figsize=IMAGE_SIZE)
           plt.imshow(image_np)
           print ('img',image_path)
           plt.savefig(image_path[:-3]+'png')
         add = output_dict['detection_classes']
```

```
        print(add[0])
        if(add[0] == 1): #만약 구급차로 추측되었다면
            print("구급차 입니다")
            so.send('1*1*E'.encode()) #서버로 값 전송
        else:
            print("구급차가 아닙니다") #구급차가 아니라면
            so.send('1*1*N'.encode()) #서버로 값 전송
    except: #사진을 불러오지 못했을 때
        print("예외 처리")
```

이미지를 수신하는 서버 소스

```
print("Import...")
import socket
import datetime as dt
import os
import threading
import json
import sys
import time as t
from obyy import amb
print("Import Finish!")
def threaded(client_socket): #사진을 받는 쓰레드 함수
  while True:
    print("대기중")
    data = client_socket.recv(65536)
    print(type(data))
    print(data)
    # image_len = int(client_socket.recv(1024).decode())
    image_len = int(client_socket.recv(1024))
    print(image_len)
    total_image = b''
    total_len = 0
    while True:
      print("total -> ", total_len)
      if total_len >= image_len:
        break
      print("이미지 받는중...")
      image_data = client_socket.recv(image_len)
      print("buf -> ", sys.getsizeof(image_data))
      if sys.getsizeof(image_data) == 33:
        break
      total_len += sys.getsizeof(image_data)
      total_image += image_data
      print("이미지 전송 끝")
image_path='D:/deeplearning/models-master/research/object_detection/test_images/image.jpg'
        try:
          image = open(image_path, 'wb')
          image.write(total_image)
        except FileNotFoundError:
          print("파일을 찾을 수 없습니다.")
        if os.path.isfile(image_path):
```

```
            amb()
          else:
            client_socket.send("Fail".encode())
def Main(): #서버를 바인딩하는 메인 함수
  server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  # server_socket.bind(('127.0.0.1', 10117))
  server_socket.bind(('192.168.0.30', 8801))
  server_socket.listen(5)
  print("socket is listening")

  while True:
    client_socket, addr = server_socket.accept()
    ip, port = str(addr[0]), str(addr[1])
    print("Connected with " + ip + ":" + port)
    threading.Thread(target=threaded, args=(client_socket, )).start()
if __name__ == 'main':
    Main()
```

이미지를 송신하는 클라리언트 소스

```
# -*- coding: utf-8 -*-
import socket
import json
import cv2
import sys
import time

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('192.168.0.30', 8801))
# img_path = "./evalimg/img_" + time + ".jpg"
img_path = "/home/pi/image.jpg"
cam = cv2.VideoCapture(0)
ret, frame = cam.read()
cv2.imwrite(img_path, frame)
cam.release()
#cv2.destroyAllWindows()
sock.send(img_path.encode())
image = open(img_path, 'rb')
image_send = image.read(1048576)
print(image_send)
print(type(image_send))
print(sys.getsizeof(image_send))
sock.send(str(sys.getsizeof(image_send)).encode())
time.sleep(10)
# sock.send(input_data.encode())
sock.send((image_send.decode() + "₩n").encode())
data = sock.recv(65536).decode('utf-8')
data = json.loads(data)
print("데이터를 돌려받았다 : ", data)
```

자동차 번호인식 소스

```
import cv2
#opencv 라이브러리
import numpy as np
#numpy 라이브러리
import matplotlib.pyplot as plt
#결과물 출력을 위한 matplot라이브러리
import pytesseract
#글씨를 읽어내기 위한 pytesseract 라이브러리
#plt.style.use('dark_background')

cap = cv2.VideoCapture(0)
#웹캠 호출
cap.set(3, 320)
cap.set(4, 240)
#320*240의 사이즈 지정

while True:
#무한반복
  ret, frame = cap.read()
  #영상 사진을 불러온다
  cv2.imshow('webcam', frame)
  #화면 출력
  if cv2.waitKey(1) == ord('q'):
    #q를 누르면
    image = frame
    #영상 사진 저장
    height, width, channel = image.shape
    #높이, 너비, 채널 정하기
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    #그레이 스케일 변환
    img_blurred = cv2.GaussianBlur(gray, ksize=(5, 5), sigmaX=0)
    #가우시안블러로 노이즈 없애기
      img_thresh = cv2.adaptiveThreshold(
        #이미지 threshold 처리
      img_blurred,
      #블러
      maxValue=255.0,
      #threshold 최대값
      adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
      #검은색과 흰색으로 이미지 나누기
      thresholdType=cv2.THRESH_BINARY_INV,
      #threshold타입
      blockSize=19,
      #threshold값을 계산하기 위해 사용되는 블록 크기
      C=9
      #계산된 평균으로부터 뺄 상수값
    )
    _, contours, _ = cv2.findContours(
      #윤곽선 찾기
      img_thresh,
      #threshold한 이미지
      mode = cv2.RETR_LIST,
```

```
      #윤곽 검색 모드 설정
      method=cv2.CHAIN_APPROX_SIMPLE
      #윤곽 근사방법 )
    temp_result = np.zeros((height, width, channel), dtype=np.uint8)
    #크기가 height, width, channel인 배열 생성
    contours_dict = []
    #배열선언
    for contour in contours:
      #for 문
      x, y, w, h = cv2.boundingRect(contour)
      #윤곽선을 감싸는 사각형 구하기
      #x값, y값, 너비, 높이
      cv2.rectangle(temp_result, pt1=(x, y), pt2=(x+w, y+h), color=(255,255,255), thickne
ss=2)
      #사각형을 그린다
      contours_dict.append({
      #배열에 값들을 각각 저장
      'contour': contour, 'x': x, 'y': y, 'w': w, 'h': h, 'cx' : x + (w/2), 'cy' : y + (h/2) })
      MIN_AREA = 60
      #bounding rect의 최소 넓이
      MIN_WIDTH, MIN_HEIGHT = 2, 8
      #최소 높이와 너비
      MIN_RATIO, MAX_RATIO = 0.25, 1.0
      #가로 대비 세로 비율의 최소 최대 값
      possible_contours = []
      #조건 만족한 번호판을 저장하는 배열
      cnt = 0
      #변수
      for d in contours_dict:
      #for 문
        area = d['w'] * d['h']
        #넓이
        ratio = d['w'] / d['h']
        #가로 대비 세로 비율
        if area > MIN_AREA ₩
          and d['w'] > MIN_WIDTH and d['h'] > MIN_HEIGHT ₩
          and MIN_RATIO < ratio < MAX_RATIO:
          #조건과 비교
          d['idx'] = cnt
          #인덱스 저장
          cnt += 1
          #변수 + 1
          possible_contours.append(d)
          #배열에 저장
    temp_result = np.zeros((height, width, channel), dtype=np.uint8)
      #크기가 height, width, channel인 배열 생성
    for d in possible_contours:
      #cv2.drawContours(temp_result, d['contour'], -1, (255, 255, 255))
      cv2.rectangle(temp_result, pt1=(d['x'], d['y']), pt2=(d['x'] + d['w'], d['y'] + d['h']),
      color=(255, 255, 255), thickness=2)
      #배열에 들어간 번호판 후보들 그리기

      MAX_DIAG_MULTIPLYER = 5
      #사각형과 사각형 사이의 거리 제한
```

```
        #각 사각형들의 중심점 거리

        MAX_ANGLE_DIFF = 90.0
        #중심점들이 이루는 각의 최대크기
        MAX_AREA_DIFF = 0.5
        #면적차이
        MAX_WIDTH_DIFF = 0.8
        #너비차이
        MAX_HEIGHT_DIFF = 0.2
        #높이차이
        MIN_N_MATCHED = 3
        #위 조건을 만족하는 사각형들이 3개 이하일때 제외
        def find_chars(contour_list):
        #함수 선언
          matched_result_idx = []
          #최종 결과물의 인덱스 저장
          for d1 in contour_list:
          #for문 1
            matched_contours_idx = []
            #조건에 맞는 결과 저장
            for d2 in contour_list:
            #for문 2
            if d1['idx'] == d2['idx']:
             #contour가 같으면 똑같은 사각형이므로 비교x
             continue
             #넘기기
             dx = abs(d1['cx'] - d2['cx'])
             #중심점들의 가로 길이
             dy = abs(d1['cy'] - d2['cy'])
             #중심점들의 세로 길이
             diagonal_length1 = np.sqrt(d1['w'] ** 2 + d1['h'] ** 2)
             #d1사각형의 대각길이
             distance=np.linalg.norm(np.array([d1['cx'],d1['cy']])-np.array([d2['cx'],d2['cy']
]))
             #벡터a와 벡터b의 거리 구하기
             if dx == 0:
             #x값이 같을때
               angle_diff = 90
               #90으로 준다
               #예외처리
             else:
               angle_diff = np.degrees(np.arctan(dy / dx))
               #아크 탄젠트 값을 구한다
               #'라디안'을 '도'로 변경

             area_diff = abs(d1['w'] * d1['h'] - d2['w'] * d2['h']) / (d1['w'] * d1['h'])
             #면적의 비율
             width_diff = abs(d1['w'] - d2['w']) / d1['w']
             #너비의 비율
             height_diff = abs(d1['h'] - d2['h']) / d1['h']
             #높이의 비율
             if distance < diagonal_length1 * MAX_DIAG_MULTIPLYER ₩
             and angle_diff < MAX_ANGLE_DIFF and area_diff < MAX_AREA_DIFF ₩
             and width_diff < MAX_WIDTH_DIFF and height_diff < MAX_HEIGHT_DIFF:
```

```
            #기준에 맞다면
              matched_contours_idx.append(d2['idx'])
              #결과물의 d2 인덱스만 저장
              matched_contours_idx.append(d1['idx'])
              #결과물의 d1 인덱스 저장
              if len(matched_contours_idx) < MIN_N_MATCHED:
              #번호판 후보의 윤곽선 개수가 3보다 작으면
              #번호판은 7글자이므로
                continue
                #제외
              matched_result_idx.append(matched_contours_idx)
              #최종 후보에 넣어준다
              unmatched_contour_idx = []
              #후보에서 제외된 사각형들을 넣어주는 공간
              for d4 in contour_list:
                if d4['idx'] not in matched_contours_idx:
                #후보가 아니라면
                unmatched_contour_idx.append(d4['idx'])
                #리스트에 넣어준다
              unmatched_contour = np.take(possible_contours, unmatched_contour_idx)
              #unmatched_contour_idx에서 possible_contours와 같은 인덱스만 추출
              recursive_contour_list = find_chars(unmatched_contour)
              #함수에 넣어서 재귀함수로 돌린다
              for idx in recursive_contour_list:
              #재귀함수로 걸러진 후보들로 for문 돌리기
                matched_result_idx.append(idx)
                #살아남은 후보들을 추가해준다
                break
                 #정지
              return matched_result_idx
              #값 반환
          result_idx = find_chars(possible_contours)
          #결과물들을 함수에 넣어서 돌린다
          matched_result = []
          #결과물 저장
          for idx_list in result_idx:
            matched_result.append(np.take(possible_contours, idx_list))
            #결과물들을 주어진 index로 정렬
          temp_result = np.zeros((height, width, channel), dtype=np.uint8)
          #크기가 height, width, channel인 배열 생성
          for r in matched_result:
            for d in r:
            #cv2.drawContours(temp_result, d['contour'], -1, (255, 255, 255))
            cv2.rectangle(temp_result,pt1=(d['x'],d['y']),pt2=(d['x']+d['w'],d['y']+d['h']),  colo
r=(255, 255, 255),thickness=2)
              #후보들을 다시 그려본다

          PLATE_WIDTH_PADDING = 1.3
          #너비 패딩값
          PLATE_HEIGHT_PADDING = 1.5
          #높이패딩값
          MIN_PLATE_RATIO = 3
          #최소값
          MAX_PLATE_RATIO = 10
```

```
        #최대값
        plate_imgs = []
        #배열선언
        plate_infos = []
        #배열선언
        for i, matched_chars in enumerate(matched_result):
        #최종 후보들에 대해 루프 돌림
          sorted_chars = sorted(matched_chars, key=lambda x: x['cx'])
          #x방향에 순차적으로 정렬
          plate_cx = (sorted_chars[0]['cx'] + sorted_chars[-1]['cx']) / 2
          #센터x좌표
          plate_cy = (sorted_chars[0]['cy'] + sorted_chars[-1]['cy']) / 2
          #센터y좌표
          plate_width=(sorted_chars[-1]['x']+sorted_chars[-1]['w']-sorted_chars[0]['x']) *
PLATE_WIDTH_PADDING
          #너비 구하기
          sum_height = 0
          #높이의 합
          for d in sorted_chars:
            sum_height += d['h']
            #정렬한 후보들의 높이의 합을 구한다
            plate_height=int(sum_height / len(sorted_chars) * PLATE_HEIGHT_PADDING)
            #높이 구하기
            triangle_height = sorted_chars[-1]['cy'] - sorted_chars[0]['cy']
            #삼각형의 높이 구하기
            triangle_hypotenus = np.linalg.norm(
            #삼각형의 빗변 구하기
              np.array([sorted_chars[0]['cx'], sorted_chars[0]['cy']]) -
              #첫번째 사각형의 중심 좌표
              np.array([sorted_chars[-1]['cx'], sorted_chars[-1]['cy']])
              #마지막 사각형의 중심 좌표
            angle = np.degrees(np.arcsin(triangle_height / triangle_hypotenus))
            #아크사인으로 빗변분의 높이의 라디안 값을 구하고 '도'로 변환한다
            rotation_matrix=cv2.getRotationMatrix2D(center=(plate_cx,plate_cy), angle=
angle, scale=1.0)
            #로테이션 매트릭스를 구한다
            img_rotated = cv2.warpAffine(img_thresh, M=rotation_matrix, dsize=(width,
height))
            img_cropped = cv2.getRectSubPix(
            #회전된 이미지에서 원하는 부분만 잘라낸다
              img_rotated,
              #이미지 변형
              patchSize=(int(plate_width), int(plate_height)),
              center=(int(plate_cx), int(plate_cy))
              #번호판 부분만 잘라낸다
            )
            ifimg_cropped.shape[1]/img_cropped.shape[0]<MIN_PLATE_RATIOor img_cro
pped.shape[1]/img_cropped.shape[0] < MIN_PLATE_RATIO > MAX_PLATE_RATIO:
                continue
            #조건과 비교하여 통과
            plate_imgs.append(img_cropped)
            #img_cropped 추가
            plate_infos.append({
            #plate_infos에 각각의 정보 추가
```

```
                'x': int(plate_cx - plate_width / 2),
                #x값
                'y': int(plate_cy - plate_height / 2),
                #y값
                'w': int(plate_width),
                #너비값
                'h': int(plate_height)
                #높이값
              })
        longest_idx, longest_text = -1, 0
        plate_chars = []
        for i, plate_img in enumerate(plate_imgs):
          plate_img=cv2.resize(plate_img,dsize=(0,0),fx=1.6,fy=1.6)_,
          plate_img=  cv2.threshold(plate_img,thresh=0.0,maxval=255.0,type=cv2.THRESH
_BINARY|                                cv2.THRESH_OTSU)
            #threshold
            _,contours,_ = cv2.findContours(plate_img,mode=cv2.RETR_LIST, method=cv
2.CHAIN_                                    APPROX_SIMPLE)
            #contours 찾기
            plate_min_x, plate_min_y = plate_img.shape[1], plate_img.shape[0]
            plate_max_x, plate_max_y = 0, 0
            for contour in contours:
              x, y, w, h = cv2.boundingRect(contour)
              #boundingRect 구하기
              area = w * h
              #면적 구하기
              ratio = w / h
              #비율 구하기
              if area > MIN_AREA ₩
              and w > MIN_WIDTH and h > MIN_HEIGHT ₩
              and MIN_RATIO < ratio < MAX_RATIO:
              #기준에 맞는지 체크
                if x < plate_min_x:
                  plate_min_x = x
                if y < plate_min_y:
                  plate_min_y = y
                if x + w > plate_max_x:
                  plate_max_x = x + w
                if y + h > plate_max_y:
                  plate_max_y = y + h
                  #번호판의 x, y의 최대 최소값 구하기
            img_result = plate_img[plate_min_y:plate_max_y, plate_min_x:plate_max_x]
            #번호판 부분만 잘라내기
            img_result = cv2.GaussianBlur(img_result, ksize=(3, 3), sigmaX=0)
            #노이즈 없애기
            _,img_result = cv2.threshold(img_result,thresh=0.0,maxval=255.0, type=cv2.T
HRESH_B                                     INARY | cv2.THRESH_OTSU)
            #threshold
            img_result = cv2.copyMakeBorder(img_result,top=10,bottom=10,left=10, righ
t=10,bor                                    derType=cv2.BORDER_CONSTANT, value=(0, 0, 0))
            #이미지에 패딩을 준다
            chars = pytesseract.image_to_string(img_result,lang='kor',config='--psm7 --
oem 0')
            #이미지에서 글자 읽기
```

```
            result_chars = ''
            #정답글자
            has_digit = False
            for c in chars:
              if ord('가') <= ord(c) <= ord('힣') or c.isdigit():
              #숫자나 한글이 포함되어 있는지 체크
                if c.isdigit():
                #숫자가 한개라도 있는지 체크
                  has_digit = True
                  result_chars += c
                  #최종 결과물에 추가
            print(result_chars)
            #최종 결과물 출력
            plate_chars.append(result_chars)
            if has_digit and len(result_chars) > longest_text:
            #가장 긴 문자열 구하기
              longest_idx = i
             #번호판으로 저장
```

<SSG_parsing.js> : 미세먼지를 공공 api에서 받아오는 소스

```
var convert = require('xml-js');     // xml 데이터를 json으로 파싱할 수 있는 모듈
var request = require('request');   // 미세먼지 API에 요청을 보낼 때 사용하는 모듈

// MariaDB와 연동하는 소스
var sql = mysql.createConnection({
    host: "localhost",
    database: "road",
    port: "3306",
    user: "root",
    password: "1234"
})

 var request =
'http://openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforInqireSvc/getCtprvnM
esureSidoLIst?serviceKey=Xq64Np67pvkDgHDDaTQ0gc6h3sjTC3PfJOIMRPaFwstU5h%2B
CKeBUbRL3%2FWurv%2FyfcDClbXGsB3jlHm01ShvNEg%3D%3D&numOfRows=9&pageNo=1
&sidoName=%EC%9D%B8%EC%B2%9C&searchCondition=HOUR' // 미세먼지 공공 API
데이터를 받아올 때 사용하는 인증 Key

console.log("미세먼지 데이터 수신 대기중입니다...");

var scheduler = schedule.scheduleJob('05 * * * *', function(){   //매 시 05분마다 실행되는 함수
request.get(requestUrl, (err, res, body) => { //API에서 데이터 얻어오는 함수
```

```
        if(err) {
            console.log(`err => ${err}`);
        }
        else {
            //받아온 데이터 데이터 베이스에 삽입
       }
```

<SSG_app.js> : 어플로 들어오는 요청을 처리하는 소스

```
var net = require('net');        // 소켓 서버를 위한 모듈
var mysql = require("mysql");    // MariaDB와 연동을 위한 모듈

var client = new net.Socket();  //경로분석해주는 파일에 데이터 전달
client.setEncoding('utf8');
client.connect('8604','192.168.0.2',function(){
});

// 서버 생성
var server = net.createServer(function(socket){ // 콜백함수 : 클라이언트 소켓 연결

socket.setEncoding("utf8");  // 데이터 UTF-8로 인코딩
socket.on('data', function(data){   //: 'data' 이벤트가 왔을 때 콜백함수 실행
    var command = data.split("*"); // 들어온 데이터를 " "로 구분해 배열로 저장
    console.log(socket.remoteAddress.toString().substring(7) + " : " + data);
    // 해당 소켓에서 들어온 데이터 콘솔에 출력

    if(command[0] == "delete"){      //앱에서 삭제 요청이 들어왔을 때

            var Delete = "DROP TABLE "+ command[1];

            sql.query(Delete,function(err,result,field){     //테이블 삭제

                if(!err){
                    console.log(command[1] + " 테이블 삭제 성공 ...");
                }
                else{
                    socket.write("error : " + err);
                    console.log("error : " + err);
```

```
                }
            })
    }
```

<SSG_dust.py> : 미세먼지 데이터를 시각화하는 소스

```
#-*- coding:utf-8 -*- //파이썬 한글 인코딩

import schedule #일정시간마다 실행할 수 있게 해주는 모듈
import time    #시간에 관한 함수를 쓰는 모듈
import numpy as np #다차원 배열을 쉽게 처리하는 모듈

from matplotlib import pyplot as plt #그래프 표시할 수 있게 해주는 모듈
from socket import * #소켓통신 모듈

HOST='192.168.0.2' #서버주소
PORT=8603          #서버포트
c = socket(AF_INET, SOCK_STREAM) #소켓 객체 생성
c.connect((HOST, PORT))           #소켓 연결

file_transferred = 0
def job():       #일정시간 마다 실행하는 함수
    day = [] #배열 초기선언
    c.send("munzi")     #munzi라는 데이터를 보낸다
    u = str(c.recv(1024)).encode("utf-8")  #데이터 수신대기
    a = u.split('*') #*로 문자열 구분

    day.append(a[8]) #날짜 삽입
    plt.figure() #새로운 figure 생성
    x= np.arange(3) #x와 y를 일정구간으로 나누어줌
    y= np.arange(0,210,20)

    p1 = plt.bar(x-0.25,nam,width,label="namgu") #막대그래프 생성
    plt.xticks(x,day) #x, y축의 현재 눈금 위치 및 레이블을 설정함
    plt.yticks(y)
    plt.legend() #라벨 및 범례를 표시함

    plt.xlabel('day') #x축 제목 설정
    plt.ylabel('%') #y축 제목 설정
    img = plt.gcf() #figure에 접근할 수 있는 객체 생성
    img.savefig('munzi.jpg') #표를 munzi.jpg라는 이름의 파일로 저장

schedule.every(10).seconds.do(job) #job()함수를 10초마다 실행
while 1: #무한루프를 돌면서 스케줄을 유지함
```

```
        schedule.run_pending()
c.close()
```

<SSG_image.py> : 이미지 소켓통신_서버

```
# -*- coding: utf-8 -*-
import SocketServer #소켓서버 모듈
import sys #입출력 모듈
from os.path import exists #경로설정 모듈

HOST ='192.168.0.2' #IP
PORT = 8605 #PORT

class MyTcpHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        data_transferred = 0
        print('[%s] 연결됨' %self.client_address[0])
        filename = self.request.recv(1024) #클라이언트로 부터 파일이름을 전달받음
        #filename = filename.decode() #파일이름 이진바이트스트림 데이터를 일반 문자열로

        if not exists(filename): #파일이 해당 디렉터리에 존재하지 않으면
            return #handle()함수를 빠져 나온다.

        print('파일[%s] 전송 시작...' %filename)
        with open(filename,'rb') as f: #요청온 데이터의 이름으로 된 파일 참조
            try:
                data = f.read(1024) #파일을 1024바이트 읽음
                while data: #파일이 빈 문자열일때까지 반복
                    data_transferred += self.request.send(data) #파일 전송
                    data = f.read(1024)
            except Exception as e: #예외 처리
                print(e)

        print('전송완료[%s], 전송량[%d]' %(filename,data_transferred))

def runServer(): #서버 함수
    print('++++++파일 서버를 시작++++++')
    print("+++파일 서버를 끝내려면 'Ctrl + C'를 누르세요.")

    try:
        server = SocketServer.TCPServer((HOST,PORT),MyTcpHandler)
        server.serve_forever()
    except KeyboardInterrupt:
```

```
        print('++++++파일 서버를 종료합니다.++++++')

runServer()
```

<munzi.py> : 이미지 소켓통신_클라이언트

```
# -*- coding: utf-8 -*-

import schedule
import socket
import time

HOST = '192.168.0.9'
PORT = 8605

def getFileFromServer(filename):
    data_transferred = 0

    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        sock.connect((HOST,PORT))
        sock.sendall(filename.encode())

        data = sock.recv(1024)
        if not data:
            print('파일[%s]: 서버에 존재하지 않거나 전송중 오류발생' %filename)
            return

        with open('/var/www/html/jbts/8/' + filename, 'wb') as f:

            try:
                while  data:
                    f.write(data)
                    data_transferred += len(data)
                    data = sock.recv(1024)
            except Exception as e:
                print(e)

    print('파일[%s] 전송종료. 전송량 [%d]' %(filename, data_transferred))

schedule.every(10).seconds.do(getFileFromServer,'mia.jpg')
```

```
while 1:
    schedule.run_pending()
    time.sleep(1)
```

안드로이드_비콘스캐너

```
import java.util.ArrayList;
            ...

public class road_beacon extends AppCompatActivity implements BeaconConsumer {

  private SocketManager m_SocketManager;
  private final static int SOCKET_CREATE_SUCCESS = 0;
  private final static String IP = "192.168.0.2";
  private final static int PORT = 8600;

  String data1;            //비콘의 값을 받는다
  private BeaconManager beaconManager;    // 감지된 비콘들을 임시로 담을 리스트
  private List<Beacon> beaconList = new ArrayList<>();
  TextView textView;

  protected void onCreate(Bundle savedInstanceState) {   //앱 시작시  초기화설정
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_road_beacon);

    m_SocketManager = new SocketManager(IP, PORT, m_Handler);

    ImageView rabbit = (ImageView) findViewById(R.id.gif_image);
    GlideDrawableImageViewTarget gifImage = new GlideDrawableImageViewTarget(rabbit);
    Glide.with(this).load(R.drawable.road_background).into(gifImage);

    // 실제로 비콘을 탐지하기 위한 비콘매니저 객체를 초기화
    beaconManager = BeaconManager.getInstanceForApplication(this);
    textView = (TextView)findViewById(R.id.Textview);

    // setBeaconLayout 안에 iBeacon의 식별번호를 넣어준다.
    beaconManager.getBeaconParsers().add(new BeaconParser().setBeaconLayout("m:2-
```

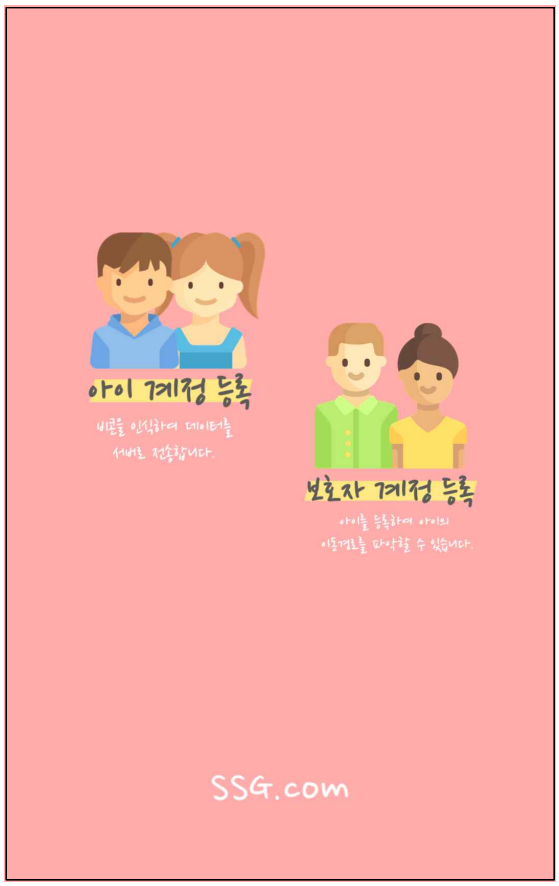
```
                                        3=0215,i:4-19,i:20-21,i:22-23,p:24-24,d:25-25"));

    // 비콘 탐지를 시작한다. 실제로는 서비스를 시작하는것.
    beaconManager.bind(this);
    handler.sendEmptyMessage(0);     //후에 textView에 비콘의 정보들을 뿌린다.
    handler2.sendEmptyMessage(0);    //서버에 textView에 있는 정보를 보낸다.
  }

  protected void onStop() {  //앱 종료시
    super.onStop();
    m_SocketManager.closeSocket();   //소켓을 닫는다.
  }

  protected void onResume(){
    super.onResume();
    SystemRequirementsChecker.checkWithDefaultDialogs(this);
  }

  protected void onDestroy() {
    super.onDestroy();
    beaconManager.unbind(this);
  }

  public void onBeaconServiceConnect() {
    beaconManager.setRangeNotifier(new RangeNotifier() {
    // 비콘이 감지되면 해당 함수가 호출된다.
    // region에는 비콘들에 대응하는 Region 객체가 들어온다.
    public void didRangeBeaconsInRegion(Collection<Beacon> beacons, Region region) {
      if (beacons.size() > 0) {
        beaconList.clear();
          for (Beacon beacon : beacons) {
            beaconList.add(beacon);
          }
        }
      }
    });
    try {
      beaconManager.startRangingBeaconsInRegion(new Region("myRangingUniqueId",
                                                                        null, null, null));
    } catch (RemoteException e) {   } }
```

```
  Handler handler = new Handler() {
    public void handleMessage(Message msg) {
      textView.setText(""); // 비콘의 아이디와 거리를 측정하여 textView에 넣는다.
      for(Beacon beacon : beaconList){
        textView.append( "" + beacon.getId1());      //비콘의 UUID 값을 추출한다.
      } handler.sendEmptyMessageDelayed(0, 5000); // 자기 자신을 1초마다 호출
    }
  };

  Handler handler2 = new Handler() {
    public void handleMessage(Message msg) {
      data1 = textView.getText().toString(); // 데이터를 String 형태로 받아서 data에 저장
      if (data1.length() > 0) {        //데이터가 공백이 아니라면
        m_SocketManager.sendData("beacon*" + Index.name + "*" + data1.substring(9, 18));
      } handler2.sendEmptyMessageDelayed(0, 5000); // 자기 자신을 1초마다 호출
    }
  };

  Handler m_Handler = new Handler() {
    public void handleMessage(Message msg) {
      switch(msg.what) {
        case SOCKET_CREATE_SUCCESS: // 소켓 연결 성공
        break;
      }
    }
  };
}
```

4. UI/UX

[그림Ⅱ-12]디스플레이 화면

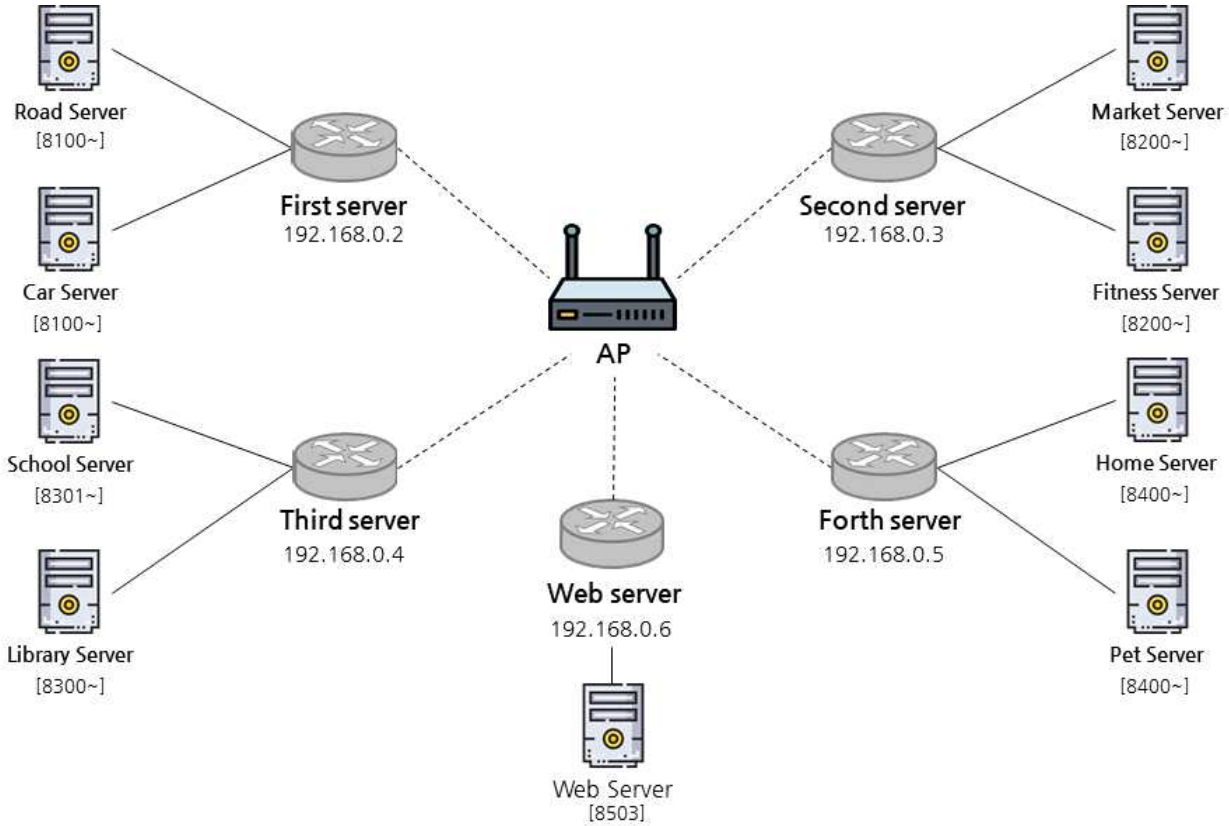
[그림Ⅱ-13]로그인 화면



[그림Ⅱ-14]계정 선택화면

4 서버

1. 네트워크 시스템



[그림Ⅱ-15] 서버구축도

2. Data Base

통신방법 : TCP/IP

road.예은: 252 행 (총) (대략적)

DATE	beacon
2019-10-30 09:40:50	b222-6654
2019-10-30 09:40:52	b222-6654
2019-10-30 09:40:54	b222-6654
2019-10-30 09:40:56	b111-3652
2019-10-30 09:40:58	b111-3652
2019-11-04 11:48:06	b333-6524
2019-11-04 11:48:34	b333-6524
2019-11-04 11:48:39	b333-6524
2019-11-04 11:49:33	b333-6524
2019-11-04 11:49:38	b333-6524
2019-11-04 11:49:43	b333-6524

[그림Ⅱ-16]비콘 사용자 테이블

road.munzi: 510 행 (총) (대략적) 다음 모두 보기 정렬 (1)

DATE	CITY	MISAE	STATUS1	CHOMISAE	STATUS2
2019-08-30 11:02:00.923348	남구	15	좋음	2	좋음
2019-08-30 11:02:00.951332	남동구	20	좋음	7	좋음
2019-08-30 11:02:00.985757	서구	17	좋음	9	좋음
2019-08-30 12:02:00.583880	남구	11	좋음	2	좋음
2019-08-30 12:02:00.613549	남동구	19	좋음	4	좋음
2019-08-30 12:02:00.642153	서구	18	좋음	9	좋음
2019-09-02 10:02:01.104527	남구	35	보통	20	보통
2019-09-02 10:02:01.141885	남동구	36	보통	16	보통
2019-09-02 10:02:01.186519	서구	34	보통	16	보통
2019-09-02 11:02:00.228506	남구	32	보통	12	좋음
2019-09-02 11:02:00.763712	남동구	39	보통	16	보통
2019-09-02 11:02:01.797414	서구	37	보통	19	보통
2019-09-03 11:02:00.709612	남구	38	보통	24	보통

[그림Ⅱ-17]미세먼지 테이블

road.bulbub: 5 행 (총) (대략적) 다음

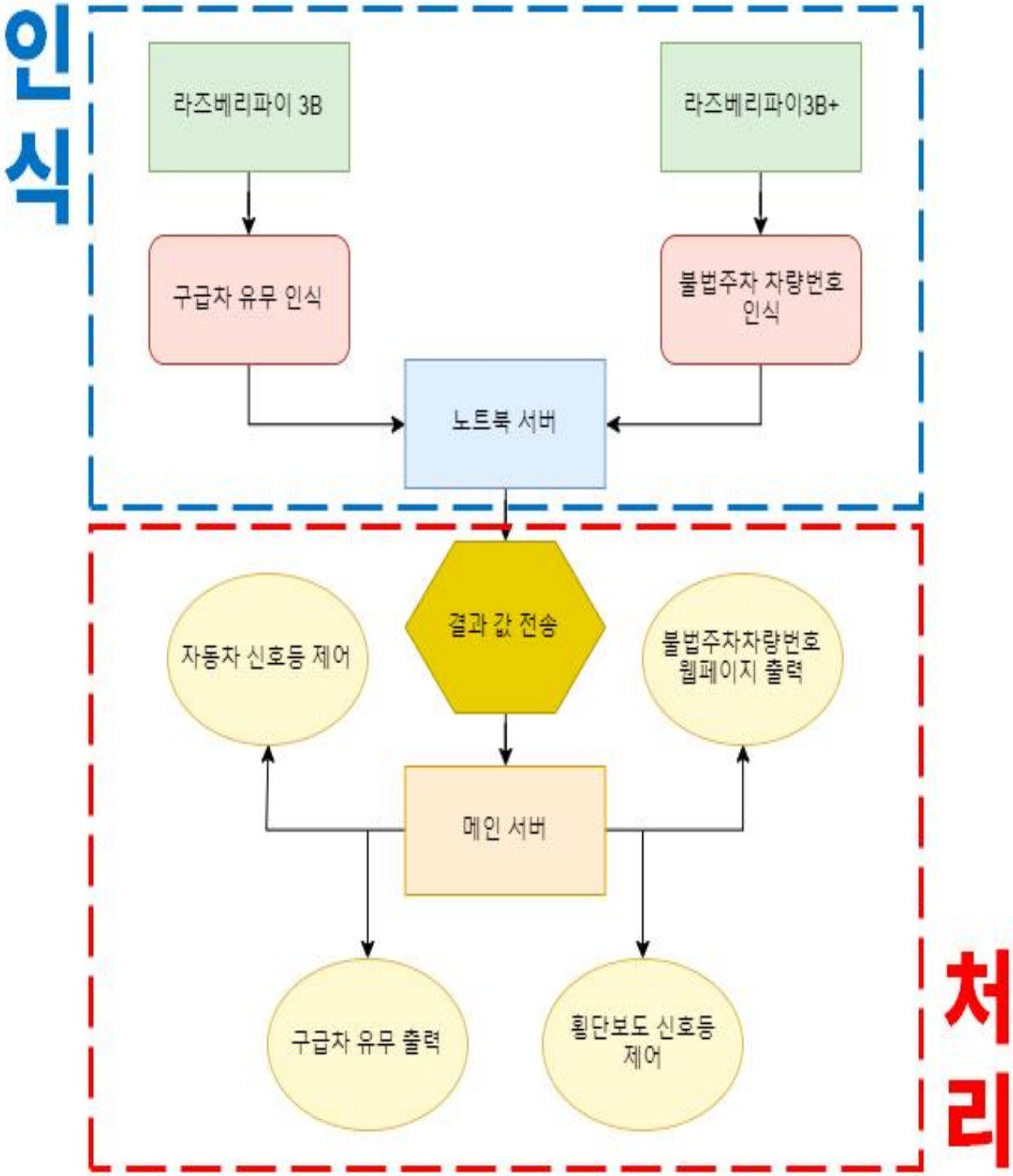
DATE	NUMBER	ZONE
2019-10-16 13:42:24.786598	74황 2001	A
2019-10-16 13:42:29.000000	53선 2001	A
2019-10-17 09:51:37.979969	28조 2001	A
2019-10-16 15:42:30.000000	24신 2001	A
2019-10-17 10:26:09.888869	98정 2003	A

[그림Ⅱ-18]불법주차 테이블

id	pw	name	phone	address	department	age	gender	height	weight	kcal	email	bankname	account	pin_num	balance	image	kid	parents	class_num	UUID	addi	rentdenydate
yeeun	0503	예은	01049145474	인천 미추홀구 한나루로	road	0		0	0	0	syedk@naver.com	MG	5161651	123456	9,999,999	927		FALSE		(NULL)	(NULL)	0
chan	0117	찬	01027739122	인천 남동구 만수로	mychan	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	chanbabo@naver.com	(NULL)	(NULL)	(NULL)	(NULL)	927	(NULL)	(NULL)		(NULL)	(NULL)	0
dahun	1221	다훈	01049419748	인천 남동구 구월로	schools	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	dahunbabo@naver.com	(NULL)	(NULL)	(NULL)	(NULL)	927	(NULL)	(NULL)	33102	2147483647	(NULL)	0
hyeongjoo	0516	형주	01062707026	인천 미추홀구 인하로	home	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	hyeongjoobabo@naver.com	(NULL)	(NULL)	(NULL)	(NULL)	927	(NULL)	(NULL)		(NULL)	(NULL)	0
subeen	0723	수빈	01065007381	인천 부평구 경인로	mart	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	subeenbabo@naver.com	kakaobank	2147483647	211521	9,913,523	927	(NULL)	(NULL)		(NULL)	(NULL)	0
seungseop	0728	승섭	01049146722	인천 미추홀구 주안로	road	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	seungseopbabo@naver.com	(NULL)	(NULL)	(NULL)	(NULL)	927	예은	TRUE		(NULL)	(NULL)	0
soohwa	1024	수화	01022307234	인천 계양구 봉오대로	dog	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	soohwababo@naver.com	kakaobank	4645645615	150152	1,000,000	927	(NULL)	(NULL)		(NULL)	(NULL)	0
hyeonggyun	0213	형균	01032837936	경기도 시흥시 포도원로	library	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	hyeonghyunbabo@naver.com	(NULL)	(NULL)	(NULL)	(NULL)	927	(NULL)	(NULL)		(NULL)	(NULL)	2019-11-08
minwoo	0704	민우	01051612986	인천 미추홀구 석정로	library	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	927	(NULL)	(NULL)		(NULL)	(NULL)	0
dasol	0927	최민우	01056395534	인천 남동구 호구포로	fitness	25	M	198	85	2,059	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	4	(NULL)	(NULL)		(NULL)	(NULL)	0
yeoungchae	0222	정영채	01024658185	서산	mart	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	927	(NULL)	(NULL)		(NULL)	(NULL)	0
hmh0704	010704	황민혁	01020380169	고양	road	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
qqqq	qwer	정희준	01089582162	인마고	road	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)				(NULL)	(NULL)	0
yennnn	010503	선예은	0101111111	인천	road	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
Deongjin	010203	허성진	01040512321	Bucheon	dog	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
cheol247	abc8870	김상철	01082224684	경인로 335번길 16	mychan	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
dudco	0220	영채	01024658185	서산시	mart	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
Hong	1111	홍길동	01012345678	인천전자마이스터고등학교	road	0	남자	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	FALSE		(NULL)	아버지를 아버지라 부르지 못함	0
guswns	0707	현준	01024390224	쌍라	mart	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
guswns2	0707	현준2	01024390224	쌍라동	mart	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)		(NULL)	(NULL)	0
gyungbin	0306	김경빈	01011111111	인천	schools	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	33101	708331629689	(NULL)	0
hosulake	1004	정호수	01011111111	인천	schools	(NU...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	33114	(NULL)	(NULL)	0

[그림Ⅱ-19]사용자정보 테이블

3. Cloud



[그림Ⅱ-20]클라우드 서비스

5 웹 페이지

통신방법 : TCP/IP

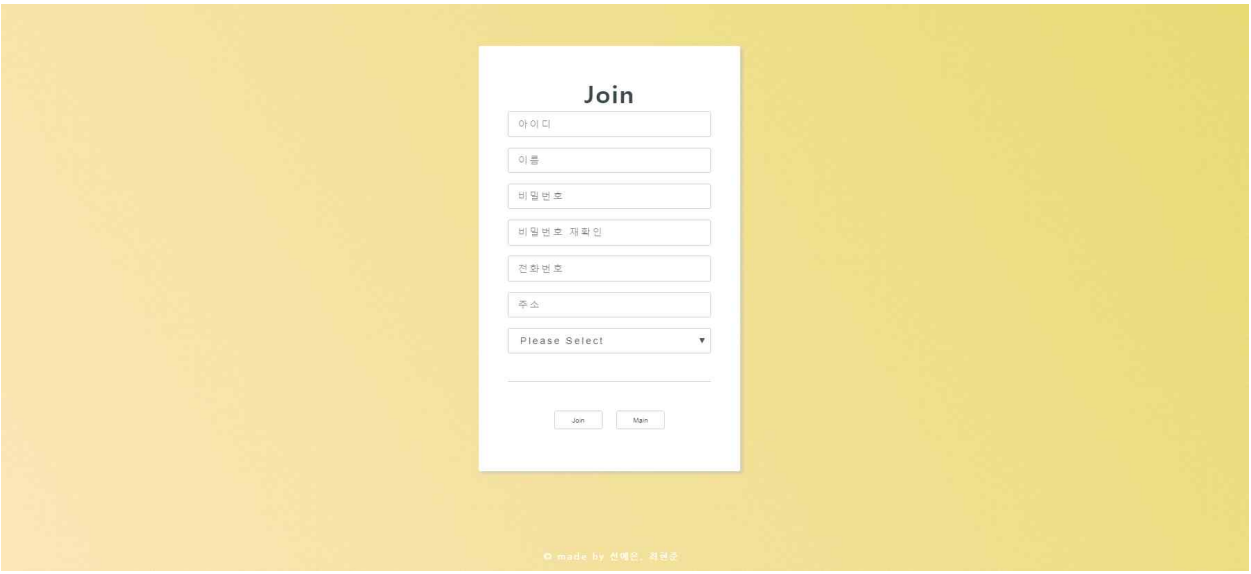
가. Main Page

<Index.html>

[그림Ⅱ-21]Main Page index.html

```
<header>
  <span><img src="images/meister.png" alt="" /></span><br><br>
  <h1>Smart City</h1>
  <p>정보통신기기과 3학년</p><hr/>
</header>

<ul class="actions special">
  <li><a href="login.html" class="button">Login</a></li>
                                <!-- Login 이라고 쓰여진 버튼을 누르면 login.html 로 이동 -->
</ul>

<ul class="actions special">
  <li><a href="join.html" class="button">Join</a></li>
                                  <!-- join 이라고 쓰여진 버튼을 누르면 join.html 로 이동 -->
</ul>
```

〈Join.html〉〈Join.php〉



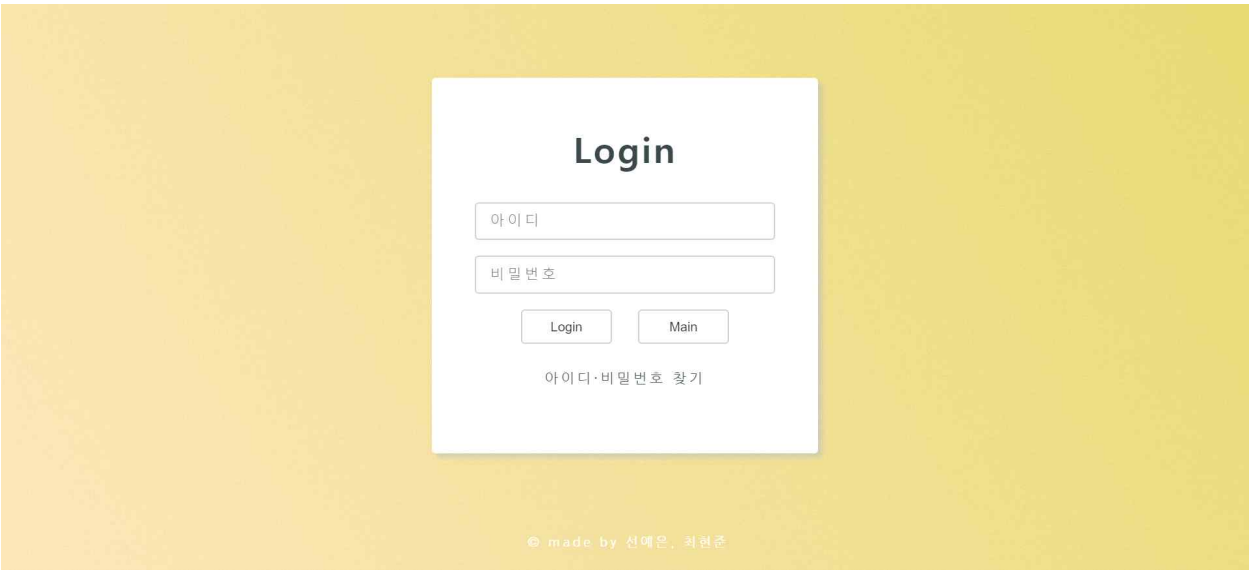
[그림Ⅱ-22]Main Page join.html

```
〈form method="post" action="join.php"〉
                                〈!-- Post 방식으로 sign.php에 전송하는 form 태그--〉
  〈div class="fields"〉
    〈div class="field"〉
      〈input type="text" name="id" id="id" placeholder="아이디" /〉
                                                〈!-- 변수 id에 id값 저장하여 전송--〉
    〈/div〉
                                        ...
    〈div class="field"〉
      〈select name="department" id="department"〉
          〈!-- select 형식을 이용하여 변수 department 에 선택한 값 저장하여 전송--〉
      〈option value=""〉Please Select〈/option〉
                            ...
      〈option value="road"〉Smart Road〈/option〉
    〈/select〉 〈/div〉 〈/div〉 〈hr /〉

     〈div class="field"〉
    〈button type = "submit" onclick = "location.href='join.php'" 〉Join〈/a〉
    〈button type = "button" onclick = "location.href='index.html'" 〉Main〈/a〉
  〈/div〉
〈/form〉
```

```
〈?php
  $id = $_POST['id']; //변수 id에 post방식으로 받아온 id값 저장
```

```
                                                ...
  $department = $_POST['department']; //post방식으로 받아온 department값 저장

  if($pw!=$pwc) //비번, 비번 확인 다르면
  {
   echo "<script>alert('비밀번호가 서로 일치하지 않습니다.');
   history.back(); </script>"; exit();
  }

  if($id == NULL || $pw == NULL || $pwc == NULL || $name == NULL || $phone
    == NULL || $address == NULL || $department == NULL ) //한칸이라도 빈칸일 시
  { echo "<script>alert('빈칸을 모두 채워주세요.); history.back(); </script>"; exit(); }

  $mysqli = mysqli_connect( '192.168.0.6', 'root', '1234', 'jbts' );
  mysqli_set_charset($mysqli,"utf8"); //DB 문자 셋을 utf8로 설정

  $checkid="SELECT * from user WHERE id='$id'";
  $result=$mysqli->query($checkid); //DB쿼리문을 실행하고 나온 값을 변수에 저장
  if($result->num_rows==1){ //결과가 있을 경우
   echo "<script>alert('중복된 ID 입니다.'); history.back(); </script>"; exit();
  }

  $checkname="SELECT * from user WHERE name='$name'";
  $result=$mysqli->query($checkname); //DB쿼리문을 실행하고 나온 값을 변수에 저장
  if($result->num_rows==1){
   echo "<script>alert('중복된 이름 입니다.'); history.back(); </script>"; exit();
  }

  $signup=mysqli_query($mysqli," INSERT INTO user (id,pw,name,phone,address,
      department) values('$id','$pw','$name','$phone','$address','$department') ");

  if($signup){ //signup의 값이 있을 경우
   echo "<script>alert('회원가입 성공!');</script>"; //회원가입 성공 팝업창 생성
   echo "<script>location.replace('index.html'); </script>"; exit();
  }
?>
```

‹login..html› ‹lo_check.php› ‹pageflip.php›



[그림Ⅱ-23]Main Page login.html

```
<!-- Main -->
  <section id="main">
    <h1>Login</h1><br>
    <form method="post" action="lo_check.php">
      <div class="fields">

        <div class="field">
          <input type="text" name="id" id="id" placeholder="아이디" />
        </div>

        <div class="field">
                                                    ...
        </div> </hr>

        <div class="field">
          <button type = "submit"  class = "button" onclick="location.href=
                                                        'lo_check.php'">Login</a>
          <button type = "button"  class = "button" onclick="location.href=
                                                            'index.html'">Main</a>
        </div>
      </div>

    <a href="findid.html">아이디</a> <h7>·</h7> <a href="findpw.html">비밀번호 찾기</a>
   </form> </hr> </section>
```

```
<?php session_start(); ?> //세션 시작하기 위해 제일 먼저 선언되야 함
<?php @ob_start(); ?>
<?php
  $id = $_POST['id'];
  $pw = $_POST['password'];
  $mysqli = mysqli_connect("192.168.0.6","root","1234","jbts"); //DB연결 함수
  $_SESSION['id'] = $id; //session 변수에 id값 저장

  $check="SELECT * from user WHERE id='$id'";
  $result=$mysqli->query($check); //쿼리문을 실행하고 결과값을 변수 result에 저장
  if($result->num_rows==1) { //쿼리문이 실행되면
    $row=$result->fetch_array(MYSQLI_ASSOC); //row에 배열로 값 저장
    if($row['pw']==$pw){ //비밀번호가 맞으면
      if(isset($_SESSION['id'])) //세션이 있다면
      {
        $_SESSION['departresult']=$result_row[5];
        echo("<script>location.replace('pageflip.php');</script>"); exit();
      } else{ echo "세션 저장 실패"; }
    } else{ echo "<script>alert('아이디 혹은 비밀번호를 확인해주세요.'); history.back();</script>"; exit(); }
  } else{ echo "<script>alert('아이디 혹은 비밀번호를 확인해주세요'); history.back();</script>"; exit(); }
?>
```

```
<? session_start();?> <!--세션 변수를 사용하기 위함-->
<?php @ob_start();?>
<?php
  switch($_SESSION['departresult'])
    {
      CASE "home": //departresult가 home일 경우
        echo("<script>location.replace('../1');</script>"); break;
                                    ...
    }
?>
```

‹findid.html› ‹findid.php› ‹findpw.html› ‹findpw.php›

[그림Ⅱ-24]Main Page findid.html

[그림Ⅱ-25]Main Page findpw.html

```
<?php
  $name = $_POST['name']; //변수 nick에 post방식으로 받아온 nickname값 저장
  $mysqli = mysqli_connect( '192.168.0.6', 'root', '1234', 'jbts' );
  mysqli_set_charset($mysqli,"utf8"); //DB 문자 셋 utf8로 설정

  if($name == ""){ echo '<script> alert("항목을 입력하세요"); history.back(); </script>';
  } else {
    $data = array();
    $check = "SELECT * from user WHERE name = '$name'";
    $result = $mysqli->query($check); //쿼리문을 실행하고 결과값을 변수에 저장
    if($result->num_rows==1) { //쿼리문이 실행되면
      $row = mysqli_fetch_array($result);
      echo "<script>alert('회원님의 아이디는 ".$row['id']."입니다.');</script>";
      echo("<script>location.replace('login.html');</script>");
    } else { echo "<script>alert('이름을 확인해 주세요.'); history.back();</script>";
    }
  }
?>
```

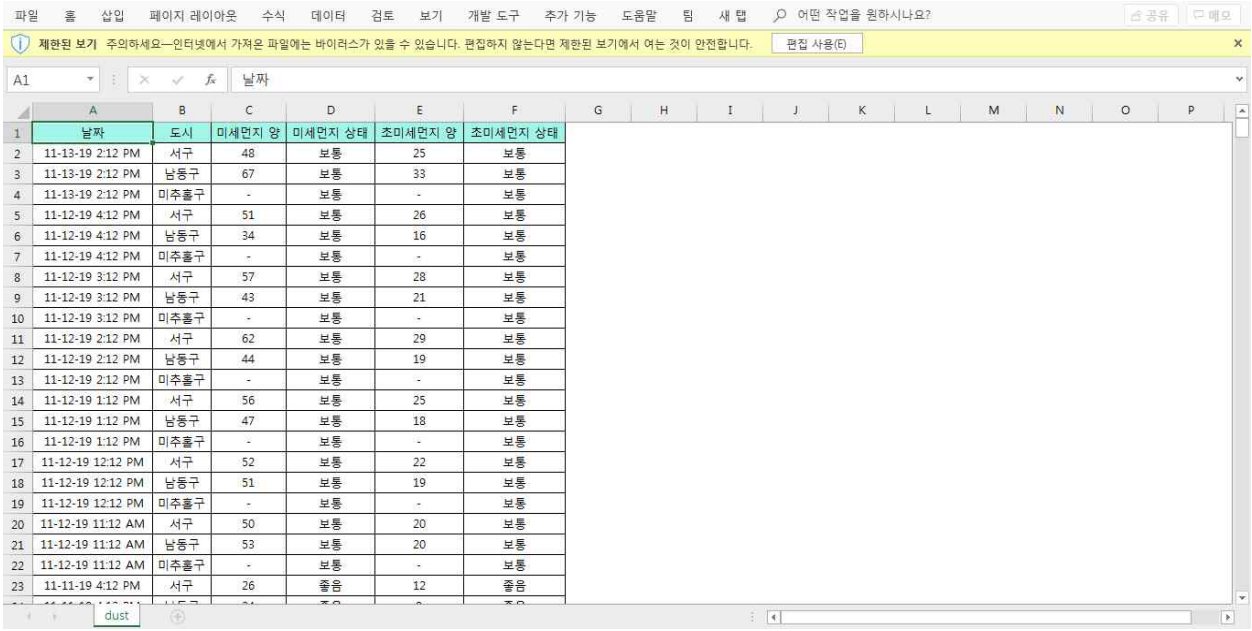
```
<?php
  $id = $_POST['id']; //변수 id에 post방식으로 받아온 id값 저장
                                                    ...
  $mysqli = mysqli_connect( '192.168.0.6', 'root', '1234', 'jbts' );
  mysqli_set_charset($mysqli,"utf8"); //DB 문자 셋

  if($id == "" || $name == "" || $phone == ""){
    echo '<script> alert("항목을 입력하세요"); history.back(); </script>';
  }else{
    $data = array();
    $check = "SELECT * from user WHERE id = '$id'";
    $result=$mysqli->query($check);
    if($result->num_rows==1) { //쿼리문이 실행되면
      $row = mysqli_fetch_array($result);
      if($row['name'] == $name && $row['phone'] == $phone) {
        "<script>alert('회원님의 비밀번호는 ".$row['pw']."입니다.');</script>";
         echo("<script>location.replace('login.html');</script>");
      } else if($row["name"] != $name && $row["phone"] == $phone){
        echo "<script>alert('이름을 확인해 주세요'); history.back();</script>";
      } else if($row["name"] == $name && $row["phone"] != $phone){
        echo "<script>alert('전화번호를 확인해 주세요'); history.back();</script>";
      } else if($row["name"] != $name && $row["phone"] != $phone){
        echo "<script>alert('이름과 전화번호를 확인해 주세요'); history.back();</script>"; }
   } else { echo "<script>alert('id를 확인해 주세요'); history.back();</script>"; } } ?>
```

나. 데이터베이스의 데이터 가져오기

[그림Ⅱ-26]8조 parking.php

```
<?php
  $jb_conn = mysqli_connect( '192.168.0.67', 'root', '1234', 'road' );
  mysqli_set_charset($jb_conn,"utf8");
  $sql = "SELECT * FROM bulbub";
  $result = mysqli_query( $jb_conn, $sql );
?>
                                                    ...
<thead>
  <tr>
    <th>Car Num.</th>
    <th>Time</th>
  </tr>
</thead>
<tbody>
  <?php
    while($row = mysqli_fetch_row($result)){
      echo "<tr>";
        echo "<td>".$row[1]."</td>";
        echo "<td>".$row[0]."</td>";
      echo "</tr>“;
    }
  ?>
</tbody>
```

다. 데이터베이스 다운



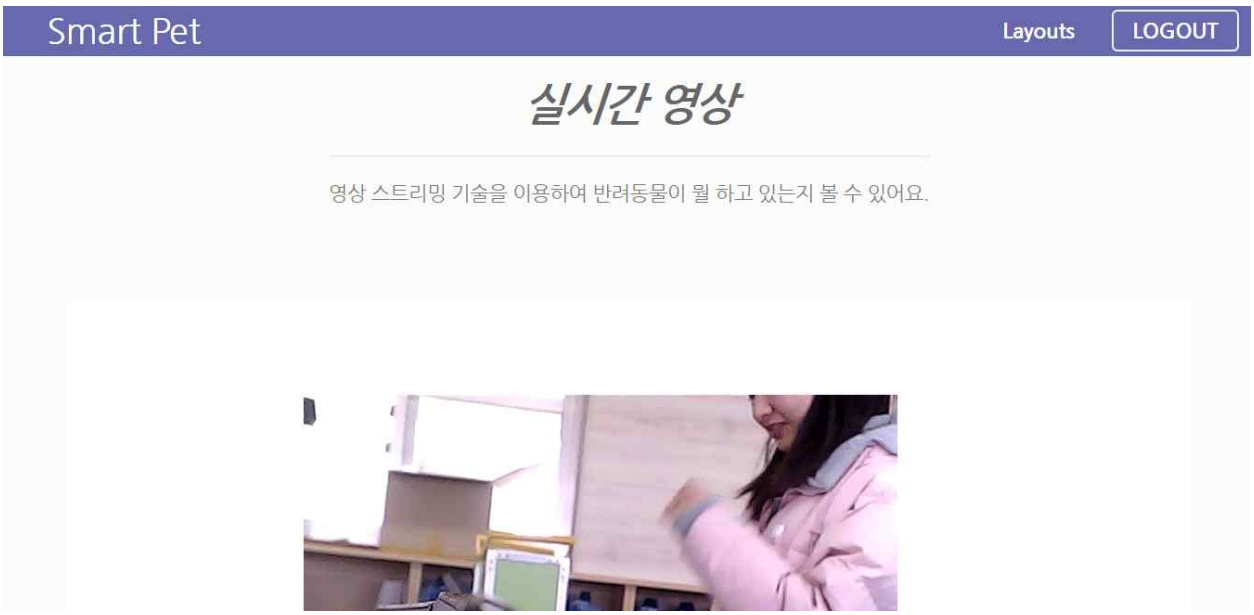
날짜	도시	미세먼지 양	미세먼지 상태	초미세먼지 양	초미세먼지 상태
11-13-19 2:12 PM	서구	48	보통	25	보통
11-13-19 2:12 PM	남동구	67	보통	33	보통
11-13-19 2:12 PM	미추홀구	-	보통	-	보통
11-12-19 4:12 PM	서구	51	보통	26	보통
11-12-19 4:12 PM	남동구	34	보통	16	보통
11-12-19 4:12 PM	미추홀구	-	보통	-	보통
11-12-19 3:12 PM	서구	57	보통	28	보통
11-12-19 3:12 PM	남동구	43	보통	21	보통
11-12-19 3:12 PM	미추홀구	-	보통	-	보통
11-12-19 2:12 PM	서구	62	보통	29	보통
11-12-19 2:12 PM	남동구	44	보통	19	보통
11-12-19 2:12 PM	미추홀구	-	보통	-	보통
11-12-19 1:12 PM	서구	56	보통	25	보통
11-12-19 1:12 PM	남동구	47	보통	18	보통
11-12-19 1:12 PM	미추홀구	-	보통	-	보통
11-12-19 12:12 PM	서구	52	보통	22	보통
11-12-19 12:12 PM	남동구	51	보통	19	보통
11-12-19 12:12 PM	미추홀구	-	보통	-	보통
11-12-19 11:12 AM	서구	50	보통	20	보통
11-12-19 11:12 AM	남동구	53	보통	20	보통
11-12-19 11:12 AM	미추홀구	-	보통	-	보통
11-11-19 4:12 PM	서구	26	좋음	12	좋음

[그림Ⅱ-27]8조 dustdb 엑셀파일

```
<?php
  $file_name = "dust.xls";
  header( "Content-type: application/vnd.ms-excel; charset=utf-8");
  header( "Cache-Control: must-revalidate, post-check=0,pre-check=0" );
  header( "Content-Disposition: attachment; filename=$file_name" );
  header( "Content-Description: PHP4 Generated Data" );
  //header() 함수를 쓰기 전 어떠한 출력도 있어선 안됨.

  $db = isConnectDb($db);

  function isConnectDb($db)
  {
    $conn = mysqli_connect("host", "user", "pw", "dbname");
    mysqli_set_charset($conn, "utf8");
    if (mysqli_connect_errno()) {
      echo "Failed to connect to MySQL: " . mysqli_connect_error(); exit;
    } else {
      return $conn;
    }
  }
```

```
    $tblName = munzi;     // 테이블명
    $sql ="SELECT * FROM $tblName ORDER BY date DESC";
    // DESC 내림차순 정렬, ASC 오름차순 정렬
    $result = mysqli_query($db,$sql);

    // 테이블 상단 만들기
    $EXCEL_STR = "
    <table border='1'>
    <tr>
    <td align=center BGCOLOR='#9DEEE1'>날짜</td>
                                            ...
    <td align=center BGCOLOR='#9DEEE1'>초미세먼지 상태</td>
    </tr>";

    // 테이블 내용
    while ($row = mysqli_fetch_row($result) ){
      $EXCEL_STR .= "
        <tr>
          <td style=₩"mso-number-format:m₩/d₩/yy₩ h₩:mm₩ AM₩/PM;₩"
                                                    align=center>".$row[0]."</td>
          <td align=center>".$row[1]."</td>
                                            ...
          <td align=center>".$row[5]."</td>
        </tr>
      ";
    }

    $EXCEL_STR .= "</table>";
    echo $EXCEL_STR;
    exit;
?>
```

라. 동영상 실시간 스트리밍



[그림Ⅱ-28]5조 실시간 영상 페이지

```
<section class="box special">
  <div class="box">
    <p align="center"><iframe width="700" height="480"
              src="http://192.168.0.54:8409/javascript_simple.html"></iframe></p>
  </div>
</section>
```

Ⅲ. 결론

1 결과

1. 실무결과

<미아 방지>

미아 방지를 위한 어플리케이션과 데이터베이스를 활용해 사용자 계정을 생성하고 삭제하며 제어하는 기능을 구현하였고, 자녀 계정과 부모 계정으로 나누어 자녀의 이동경로를 부모의 계정에서 확인할 수 있게 하였다. 또 사용자의 계정으로 인식된 비콘 데이터를 스캔하여 서버로 데이터가 전송되면 데이터를 가시화해주는 Python의 Matplotlib 패키지를 이용해 데이터베이스에 있는 데이터들을 분석해 사용자의 이동경로를 분석해 어플에 띄우는 기능을 구현하였다.

<미세 먼지>

미세 먼지 빅 데이터를 받아오기 위해 공공 데이터 포털(https://www.data.go.kr/)에서 공개 API 인증 키를 받아와 정각마다 실행되는 스케줄러로 미세먼지 정보를 서버를 통해 데이터베이스에 저장하고, 미아 방지와 마찬가지로 파이썬의 Matplotlib 패키지를 이용하여 일별 미세먼지 평균으로 통계내어 웹과 LCD 디스플레이에 띄우는 기능을 구현하였다.

<불법 주차>

번호인식은 파이테서랙트라는 OCR API를 사용한 번호 인식 딥러닝을 통해 인식된 차량의 번호가 서버로 전송되면 데이터베이스에 감지된 구역과 차량의 번호를 저장하고, 데이터베이스에 저장된 내용을 웹과 LCD 디스플레이에 실시간으로 띄워 불법주차를 방지하는 기능을 구현하였다.

<지능형 교통신호>

물체 인식 딥러닝을 통해 구급차를 인식해 서버로 신호가 오면 구급차가 빠르게 사고 현장에 도착할 수 있도록 차량 신호를 바꾸어 지나갈 수 있도록 차량신호를 제어하고 구급차가 없어지면 정상적인 신호체계로 돌아오는 기능을 구현하였다. 구급차 인식은 Tensorflow의 Object detection api를 사용하였다. 처음에는 이미 만들어진 데이터를 이용하여 각종 물체를 인식하고 화면에 띄우는데 성공하였고, 이후 구급차, 세단, SUV사진을 100~200개 정도 수집한 후 라벨링 하여 학습시켜 구급차 인식의 기능을 구현할 수 있었다.

2. 기대효과

가. 스마트 로드를 구현함으로써 더욱 더 편리한 도로를 이용할 수 있다.
나. 미아 발생률은 낮추고 구급차의 원활한 이동을 기대할 수 있다.

2 시사점

직책	성명	시사점
조장	조승섭	맨 처음에 생각했던 지능형 교통신호는 너무 장벽이 높아 거리가 멀어진 것이 아쉬웠고, 내가 더 많은 노력을 했으면 더 좋은 작품이 나왔을 것 같아서 아쉽고 조원들에게 미안하다. 그리고 소프트웨어와 하드웨어 사이 소통이 중요한데 다른 조들에 비해 괜찮다고 생각하지만 조원들 사이 소통이 부족했던 것 같다.
조원	선예은	웹페이지를 제작하면서 제일 아쉬웠던 점은 시간에 쫓겨 다양한 언어를 활용하지 못했다는 점이다. 웹프로그램을 만드는 언어도 정말 많고 웹서버를 구축하는 방법도 여러 가지다. 처음 목표가 다양한 언어를 다뤄보는 것이었는데 프로젝트 진행을 너무 여유롭게 했던 탓인지 베이스 구축을 한 후에는 시간에 쫓겨 목표는 이루지도 못하고 한정적인 언어로만 웹페이지 제작을 했던 것이 너무 아쉽고 웹서버 구축도 node.js로 해보고 싶었는데 apache2로 끝낸 것이 정말 아쉽다.
조원	신동준	처음으로 툴을 셋팅하고, 라이브러리를 설치하는 과정을 좀 더 자세히 알아보고 정확하게 숙지하여야 겠다. 계속 삭제 후 설치를 진행하니 다른 라이브러리와 충돌이 일어나고 임포트가 제대로 이루어지지 않았다. 개발 환경을 윈도우가 아닌 리눅스에서 진행하는 것이 훨씬 많은 정보가 있어 아쉬웠다. 노트북에 리눅스를 설치 할수 없던 것이 큰 걸림돌이었다. 각 라이브러리의 함수들을 알아보고 사용하였으면 코딩 시간이 단축되었을 것 같다.
조원	정회훈	주변의 비콘장치를 인식하여서 서버에 시각과 비콘의 아이디를 전송하여 미아의 경로를 어느정도 예상해 보고 그것을 지도에 띄운다는 것은 정말 좋았습니다. 하지만 미아의 이동경로 예측보다는 현재 미아의 위치를 표시했으면 더 정확한 어플을 만들 수 있지 않았을까 하는 아쉬움이 있습니다. 이와 같은 경우는 GPS를 이용하여 실시간 위치를 서버로 전송, 그것을 지도에 띄우는 것을 성공시킨다면 가능 할 것입니다. 기회가 된다면 한번 도전해 보고 싶습니다.
조원	황민혁	처음 접해보는 라즈베리파이라 사용하는데 어려움을 많이 겪었다. 스마트로드의 기능중 하나인 바닥 신호등 기능을 구현하기 위해 원래 WS2812 스티커로 접착되는 스트립 네오픽셀을 사용하려고 했지만 라즈베리파와의 연동의 어려움으로 결국 LED바로 구성된 네오픽셀을 사용하게 된 것이 조금 아쉬움에 남는다. 또한 비콘 역시 블루투스모듈을 목걸이로 구성하지 못한 것이 아쉽게 느껴진다.

3 느낀점

직책	성명	느낀점
조장	조승섭	지난 2년 동안 프로젝트를 해오면서 아쉬웠던 점이 있었다. 1학년 때는 적응이 안 돼 팀에게 해가 되면 안 되는 데 조원들에게 누를 끼쳤다고 스스로 생각해서 아쉬웠었고, 2학년 때는 뭔가 더 많은 비중을 맡고 싶었지만 능력 부족으로 내 할 일만 해내서 아쉬웠다. 하지만 올 해 3학년 프로젝트는 1, 2학년 때 아쉬웠던 점을 모두 이뤄낸 것 같았다. 올해 프로젝트에는 2년 동안 해보지 못한 꼭 조장이 되어 이끌고 싶었다. 프로젝트를 진행해오면서 많이 힘들고 조장을 선택했음을 후회한 적이 많았지만, 조원들과 같이 협동하고 으쌰으쌰하며 성장해나감을 느꼈다. 스마트 시티라는 대 주제로 짜인 주제 중 나는 스마트 로드가 제일 마음에 들었다. 스마트 피트니스, 스쿨 등등 작품이 있었지만 이 작품들 모두 스마트 로드안에 포함되어 있다고 생각했기 때문이다. 2년 동안 배워보지 못한 서버라는 역할을 자진해서 맡았다. 여태 프로젝트에는 내가 남는 역할을 하거나 쉬운 역할을 담당하여 뭔가 조의 깍두기가 된 느낌이었다. 올해에는 팀에서 중요한 역할을 담당해보고 싶어서 자진해서 맡게 되었다. 각 조의 서버 담당자들 모두가 처음이어서 낯설었던 역할이라 기본을 익히고 적응하는 데 오랜 시간이 걸렸다. 이러다 팀에 또 폐를 끼치겠구나. 라는 생각이 들어 열심히 공부하고 노력하여 완성할 수 있었다. 나보다 훨씬 노력 많이 해준 조원들에게 고마웠다. 3년 동안의 프로젝트를 되돌아보면서 성장해진 나를 보는 것 같아 보람차고 값진 경험이었던 것 같다.
조원	선예은	매년 MDP를 진행하면서 느끼는 것이지만 계획부터 세워서 팀원들끼리 전공 지식을 나누며 하나의 작품을 만든다는 것은 언제나 설레는 기분이다. 어떤 작품을 만들지 구상하는 과정이 조금 힘들긴 하지만 그것을 확정하여 회로를 설계하고 소프트웨어 기반을 다지고 서버를 구축하고 그 위에 살을 붙여가며 완성에 가까워질수록 마음이 붕 뜨게 된다. 하루 최소 4시간, 최대 11시간을 모두 작품을 만드는데 소비하는 탓에 몸과 머리가 매우 힘들지만 “오늘 하루는 이만큼이나 했구나, 오늘도 고생했어.”라며 뿌듯함을 느낀다. 우리는 완성만을 바라보며 열심히 작품을 진행했고, 발표회 3일 전, 완성된 우리의 작품을 봤을 땐 너무 기뻤다. 우리 팀의 1년간의 노력이 담긴 우리 작품. 작품을 완성으로 이끌어준 선생님과 팀원들에 너무 감사했다. 특히 우리 팀은 1년 동안 작품 진행에 필요한 역할에 있어 구멍 없이 모두가 열심히 해준 덕에 모두 자기의 역할에 충실할 수 있었다. 그래서 큰 트러블 없이 작품을 완성할 수 있었다. 우리 조의 작품에 대한 과

		정과 결과에 있어 모두 만족한다. 그래도 아쉬운 점이 있다면, 처음 역할을 배정받을 때 작년부터 담당해온 웹서버를 담당하게 되어 작년보단 완성도 높은 페이지를 만드리라 다짐을 했었다. 하지만 웹서버라는 게 다른 DB 서버와 회의를 통해 서로의 의견을 조율해야 하고 속도도 DB 서버에 맞춰야 한다. DB가 구축이 안 되어 있으면 만들 수가 없으며 DB의 내용이 바뀔 때마다 소스 내용도 계속 바꿔줘야 한다. 그래서 프로젝트 진행을 빨리빨리 했어야 했는데 너무 여유만 부려서인지 웹페이지 제작에 있어 많은 시간을 들일 수밖에 없었다. 그렇기 때문에 정작 조 작품에는 신경을 못 쓴 것이 조원들에게 좀 미안했다
조원	신동준	1, 2학년 과정에서 배운 모든 전공을 활용하고 응용하는 3학년 MDP는 주제 그 자체만으로도 나에게 있어서 다신 경험할 수 없는 프로젝트 과정이었다. 스스로 코드를 짜고, 디버깅하는 과정에서 나는 성장할 수 있었다. 처음으로 파이썬을 공부하면서 '내가 과연 성공할 수 있을까?' 라는 생각이 머릿속을 가득 채웠다. 하지만 성공이 실패를 밀어내고 가득 채우기 시작하자, 나에게는 자신감 이라는 것이 생겼고, 가장 믿음직한 무기가 되었다. 딥러닝을 공부하여 팀 프로젝트의 기반을 세워주었고, 나의 팀원들이 그 기반을 딛고 날아오를 수 있게 해주었다. 나는 우리 조에게 날개를 달아주는 역할이었다. 딥러닝을 처음 접하였을 때는 도통 무슨 소리인지 이해조차 할 수 없었다. 계속 연구하고 확실히 인공지능 쪽으로 넘어가니 내 머리로는 알 수가 없었다. 그래도 나는 도전하였다. 실패하고, 또 실패하고, 마침내 성공하였다. 성공하였을 때의 그 기쁨은 말로 표현할 수 없을 정도로 좋았다. 나뿐만 아니라 같은 팀원들과 함께 기쁨을 나누어 배가 되었다. 학습을 시키면서 틈틈이 공부하였고, 이해하기위해 노력하였다. 웹캠으로 스스로 테스트 해보고, 서버와 통신을 하면서 디버깅을 하였다. 이러한 과정이 나는 너무 즐거웠다. 그리고 나의 능력이 어디까지 인지 알 수 있었다. 나의 딥러닝이 핵심이 되는 이번 프로젝트로 이제 팀원과 프로젝트의 관계성에대해서 알게 되었고, 프로젝트의 진행에 있어서 계획의 수립과 진행척도를 어떻게 짜야할지 조금은 알게 되었다. 이번 포트폴리오는 온전히 완전완성이라고 이야기하지 못하지만, 실패를 맛 본 것으로 배울 수 있다면 충분한 1년이었다고 생각한다. 다만 아쉬운 것은 내가 처음부터 그런 마인드를 가지고 임했다면 실패하지 않고 성공했지 않을까하는 기대이다.
조원	정회훈	이번 프로젝트를 통해서 어떤 팀 과제 간에 팀의 개인의 역할도 중요하지만 팀의 협력 또한 중요하다는 것을 알게 되었다. 아무래도 서로 맡은 곳에서의 일도 있고 다른 친구와 연결해서 하는 기능도 있기 때문에 알게 된

		것 같다. 또한 비콘을 인식해서 인식한 비콘의 정보를 화면에 표시하는 부분이 있었는데 개인적으로 생각하기에는 지금까지 했던 프로젝트 중에서 이 부분이 가장 힘들었던 것 같다. 하지만 포기하지 않으면 언젠가는 된다는 것을 느끼게 되었고, 프로젝트를 진행하다 보니 나의 코딩 실력이나 안드로이드 스튜디오 활용도 또한 많이 성장한 것 같다. 이번 프로젝트를 계기로 안드로이드에 대해서 더 많은 것을 배우게 되었고, 아직 배우지 못한 부분들을 더 배우고 싶은 생각이 들었다. 그리고 이번 프로젝트에서 팀원 친구들을 많이 도와주지 못해서 많이 미안하고 아쉬운 것 같다. 그런데도 불구하고 끝까지 열심히 자신이 맡은 부분뿐만 아니라 다른 부분까지 신경써가며 열심히 해준 조원들에게 많은 고마움을 느낄 수 있는 프로젝트였던 것 같다.
조원	황민혁	작년 3학년 선배님들의 작품을 보면서 올해에는 어떤 주제로 작품을 만들게 될지 궁금하였고 1년 동안 프로젝트를 할 생각에 막막했다. 우리 8조는 스마트 로드라는 주제를 맡게 되며 처음에는 도로로 어떤 기능을 통해 작품을 전시하게 될지 걱정이었다. 스마트 로드의 기능을 회의할 때도 좋은 기능이 나오지 않아 고민이 되었는데 평소 스마트폰을 자주 보는 현대인들을 위해 바닥 신호등을 설치하자는 의견과 구급차일 때의 신호등을 제어하는 의견들이 나왔고 김진섭 선생님께서 비콘을 이용하여 미아방지 찾기를 해보자는 의견으로 다양한 기능들을 구현하기로 하였다. 처음 나는 회훈이와 도로 외형과 기능들을 어떤 방식으로 구현하게 될지 고민을 하였다. 나는 하드웨어와 펌웨어를 하게 되면서 스마트 로드의 신호등인 사거리신호등과 바닥 신호등을 담당하게 되었다. 처음 접해보는 파이썬과 라즈베리파이로 어려움을 겪었지만 반 친구들의 도움과 김진섭 선생님, 최윤희 선생님, 최보람 선생님, 채광원 선생님의 도움으로 사거리신호등 제어와 바닥 신호등을 만드는 데 성공을 하였다. 여러 선생님과 친구들의 도움을 받았지만 포기하지 않고 스스로 노력하여 성공한 것이 더욱 뿌듯하게 느껴진 것 같았다. 프로젝트를 하면서 자신의 역할이 아닌데도 하드웨어를 도와준 우리 8조 조원들에게 고마움을 느꼈다. 또한, 선 정리를 도와준 정성운과 하드웨어를 만드는 데 도와준 엄지현 등등 8조를 도와준 다른 조 친구들에게도 고맙다고 전하고 싶다. 매번 프로젝트를 하면서 느끼는 것은 나 혼자의 성과가 중요한 것이 아닌 친구들의 소통과 협력으로 비로소 하나의 작품이 만들어지는 것이라는 것을 다시 한 번 깨닫게 되었고 이번 3학년 프로젝트는 많이 힘들었지만, 여러모로 잊지 못할 경험으로 남을 추억이 될 것 같다.

Ⅳ. 활동사진

[그림Ⅳ-1] 활동사진

참고문헌

1. HTML 개념 : 두산백과, “HTML”, https://terms.naver.com/entry.nhn?docId=1168097&cid=40942&categoryId=32838, 2019.10.27.
2. PHP 개념 : NAVER 지식백과, “PHP”, https://terms.naver.com/alikeMeaning.nhn?query=E00274971, 2019.10.27.
3. SQL 개념 : 두산백과, “SQL”, https://terms.naver.com/entry.nhn?docId=1180957&cid=40942&categoryId=32838, 2019.10.27.
4. CSS 개념 : 두산백과, “CSS”, https://terms.naver.com/entry.nhn?docId=1179641&cid=40942&categoryId=32854, 2019.10.27.
5. JavaScript 개념 : 네이버캐스트, “javascript”, https://terms.naver.com/entry.nhn?docId=3580851&cid=59088&categoryId=59096, 2019.10.27.
6. 위키백과, “파이썬”, 2019-10-29 https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%B4%EC%8D%AC
7. 위키백과, “Node.js", 2019-10-29 https://ko.wikipedia.org/wiki/Node.js
8. 위키백과, “Python.matplotlib", 2019-10-29
9. https://ko.wikipedia.org/wiki/Matplotlib
10. 위키백과, “Python.Numpy", 2019-10-29 https://ko.wikipedia.org/wiki/NumPy
11. 위키백과, “라즈베리파이”, 2019-10-29
12. https://ko.wikipedia.org/wiki/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC_%ED%8C%8C%EC%9D%B4
13. 블로그, “파이썬”, 2019-10-31 https://tworab.tistory.com/35
14. 위키독스, “파이썬”, 2019-10-31 https://wikidocs.net/6
15. 블로그, “파이썬”, 2019-10-31 https://jjinfotech.tistory.com/21
16. 블로그, “Node.js", 2019-10-31 https://javacpro.tistory.com/62
17. 블로그, “Node.js", 2019-10-31 http://- 블로그, “Node.js", 주소, 2019-10-31
18. 블로그, “Node.js", 2019-10-31 https://medium.com/day34/node-js-노드는-무엇이고-어떠한-기능들이-있는가-1-98e49e1100ab
19. 블로그, “Node.js", 2019-10-31 https://kdydesign.github.io/2017/07/15/nodejs-npm-tutorial/

20. HTML 개념 : 두산백과, “HTML”, https://terms.naver.com/entry.nhn?docId=1168097&cid=40942&categoryId=32838, 2019.10.27.
21. PHP 개념 : NAVER 지식백과, “PHP”, https://terms.naver.com/alikeMeaning.nhn?query=E00274971, 2019.10.27.
22. SQL 개념 : 두산백과, “SQL”, https://terms.naver.com/entry.nhn?docId=1180957&cid=40942&categoryId=32838, 2019.10.27.
23. CSS 개념 : 두산백과, “CSS”, https://terms.naver.com/entry.nhn?docId=1179641&cid=40942&categoryId=32854, 2019.10.27.
24. JavaScript 개념 : 네이버캐스트, “javascript”, https://terms.naver.com/entry.nhn?docId=3580851&cid=59088&categoryId=59096, 2019.10.27.
25. 조대협, 조대협의 블로그, “Tensorflow object detection api”,https://bcho.tistory.com/1192, 2019.09
26. OTAKU, GITHUB, “Tensorflow object detection api”, https://github.com/5taku/tensorflow_object_detection_helper_tool, 2019.08
27. OTAKU, GITHUB, “Custom object detection”, https://github.com/5taku/custom_object_detection, 2019.08