

ANSIM

(안심 귀가 서비스)

Ver. 1.0

2018. 06. 25.

한국외국어대학교
정보통신공학과

LAP 팀

201401059 류형오

201402927 전진우

201402097 유한석

201400264 김강연

201601289 박슬희

201600979 김혜지

머리말

본 문서는 라즈베리파이와 안드로이드의 소켓통신을 이용한 귀가를 보다 안전하고 보호자가 확실히 확인할 수 있는 알림 시스템을 기술한다. 보호자는 어플 사용자로 보호가 필요로 하는 사람이 귀가할 경우 사진을 통해 알림을 받고 확인할 수 있다.

목 차

| | | |
|----------|---|-----------|
| 1 | 개요..... | 4 |
| 1.1 | 목적 | 4 |
| 1.2 | 용어 및 약어 | 5 |
| 2 | 제품 소개..... | 7 |
| 2.1 | 제품 개요 | 7 |
| 2.2 | 제품 필요성..... | 7 |
| 3 | 시스템 구성 및 설계..... | 8 |
| 3.1 | 전체 시스템 구성 | 8 |
| 3.2 | 사용자/서버 구성도 | 8 |
| 3.3 | 카메라/서버 구성도 | 9 |
| 3.4 | 흐름도 | 9 |
| 3.5 | 상세 설계 | 10 |
| 3.5.1 | 서버/클라이언트 연결..... | 10 |
| 3.5.2 | 어플에서의 회원가입 | 12 |
| 3.5.3 | 어플에서의 로그인 | 13 |
| 3.5.4 | 서비스 실행..... | 15 |
| 4 | 시스템 시나리오 | 21 |
| 4.1 | 순차 다이어그램..... | 21 |
| 4.2 | 시나리오..... | 21 |
| 4.3 | 시나리오 어플 수행 순서 | 22 |
| 5 | 기대효과..... | 24 |
| 6 | 프로젝트 세부일정..... | 25 |
| 7 | 팀원 담당업무 | 26 |
| 8 | 참고 문헌..... | 27 |
| 8.1 | 관련 문서 | 27 |
| 8.2 | 참고 사이트..... | 27 |
| 9 | [부록]ANSIM_SourceCode..... | 28 |
| 9.1 | Server(AWS) source code | 28 |
| 9.2 | Android source code | 30 |
| 9.3 | Sensor/Camera(Raspberry pi) source code | 57 |
| 9.4 | DB(MySQL)연결 source code..... | 58 |

1 개요

본 장에서는 라즈베리 파이(Raspberry pi)와 센서[초음파 센서(HC-SR04), 파이카메라(Pi camera)]를 이용하여 제작한 ANSIM에 대하여 소개한다. ANSIM은 집에 귀가를 확인 보호해야 하는 어린 자녀들이나 치매 노인과 함께 사는 보호자들을 타깃으로 하여 보호자가 외부에 있어도 이들의 안전귀가를 어플로 실시간 확인할 수 있는 서비스를 제공한다. 사용자는 어플을 이용하여 자신의 집 현관에 부착되어있는 라즈베리파이 카메라를 제어할 수 있으며 움직임을 감지할 시 알림과 함께 사진을 서비스로 받는다. 보고서를 통해 본 개발의 목적을 제공하고, 그와 관련된 문서, 용어 및 약어 등을 소개한다.

1.1 목적

최근 아이들이 학업을 위해 학원을 다니면서 늦은 귀가가 일상이 되어가고 있다. 하지만 맞벌이 부모의 경우 아이들이 언제 귀가했는지 등을 집에 없는 경우 확인 할 수가 없다. 또한 집에 치매노인과 같이 귀가를 확인 할 필요가 있는 경우에 외부에서도 보호자들이 어플을 통해 바로 사진과 함께 알림 서비스를 받을 수 있게 하는 상품이다. CCTV 와 같이 계속 작동 시켜놓으며 계속 확인해야 하는 번거로움이 있는 것과 달리 보호자가 외부에 나가있을 경우에만 서비스를 실행 시켜놓고 보호대상이 귀가를 하면 사진과 함께 알림을 받아 계속 확인을 하지 않아도 되어서 귀가 확인을 보다 편리하게 할 수 있을 것이다.

1.2 용어 및 약어

| 용어 및 약어 | 풀이 | 비고 |
|---|--|----|
| Raspberry Pi3 [라즈베리 파이] | 영국의 라즈베리파이 재단이 학교에서 기초 컴퓨터 과학 교육을 증진시키기 위해 만든 싱글 보드 컴퓨터. 손바닥만한 크기지만 램 1G, 1.2GHz quad-core ARM Cortex-A53 CPU를 갖춘 매우 저렴한 가격대비 성능을 갖추고있다. Micro SD 카드를 이용하여 Raspberry Pi 공식 홈페이지에서 운영체제를 다운받아 사용할 수 있다. | |
| sensor [초음파센서] | 초음파 센서는 일정한 간격으로 높은 주파수 사운드 펄스를 통하여 짧게 소리를 발사합니다. 이러한 것들은 소리의 속도를 통하여 공기 중에 전파됩니다. 전파범위 내부에 물체가 존재할 경우 초음파가 되돌아오게 되는데 이때 걸린 시간을 반환해준다. 전파각은 대략 15°이며 전파 범위는 2 ~ 400cm이다. 본 제품의 설치장소를 고려했을 때 전체적인 감지가 가능하다고 판단하여 이 센서를 채택하였다. | |
| Raspberry Pi Camera module | 라즈베리 파이 본체에 연결하여 사용하는 카메라 모듈이다. 소니의 IMX219 이미지 센서를 탑재하였다. 3g정도의 무게이며 정지화상에서 3280 x 2464 픽셀의 성능으로 사진을 찍는다. | |
| SCP | 리눅스 서버에서 다른 서버로 파일을 전송할 때 사용하는 리눅스 명령어(Security CoPy) | |
| AWS EC2 (Amazon Web Service Elastic Compute) | 아마존에서 제공하는 클라우드에서 운용이 가능한 웹 서비스이다. | |

문서명: 인터넷응용 최종보고서

| 용어 및 약어 | 풀이 | 비고 |
|---------|---|----|
| MySQL | 표준 데이터베이스 질의언어인 구조화 질의언어 (SQL: Structured Query Language)를 사용하는 공개소스 의 관계형 데이터 베이스 시스템(RDBMS). 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 스레드(thread)를 지원하고, C, C++, 에펠(Eiffel), 자바, 펄, PHP, 파이썬(Python) 스크립트 등을 위한 응용 프로그램 인터페이스(API)를 제공한다. 유닉스나 리눅스, 윈도우 운영체제 등에서 사용할 수 있다. 램프(LAMP), 즉 리눅스 운영 체제와 아파치 (Apache) 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 공개 소스(오픈 소스)로 개발되는 무료 프로그램이어서 홈페이지 나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다. | |
| Apache | 1995년 발표된 월드 와이드 웹 서버용 소프트웨어. 유닉스, 윈도우를 비롯하여 거의 모든 운영체제와 시스템에서 운영이 가능하다. | |

2 제품 소개

2.1 제품 개요

ANSIM제품은 안심귀가 서비스를 상징하는 '안심'이라는 뜻이 제품명이다.
제품을 현관에 설치하여 정보를 제공 받음으로써 보호자가 보다 확실하게 귀가를 확인 받을 수 있게 서비스를 제공한다.

2.2 제품 필요성

'세계 실종아동의 날' 맞은 5월, 매일 아이 62명 사라져

▶ 좋아요 0개 공유하기

전남 강진 여고생이 실종된 가운데 대전지역에서도 매년 아동·청소년 실종신고가 1000건 이상 발생하고 있는 것으로 나타났다.

20일 대전지방경찰청에 따르면 최근 3년 간 실종 아동·청소년 신고 건수는 2015년 1161건, 2016년 1147건, 지난 해 1169건이다. 대부분 경찰의 수색이나 자진귀가 등으로 집으로 돌아갔으나 아직 16명은 발견되지 못했다.

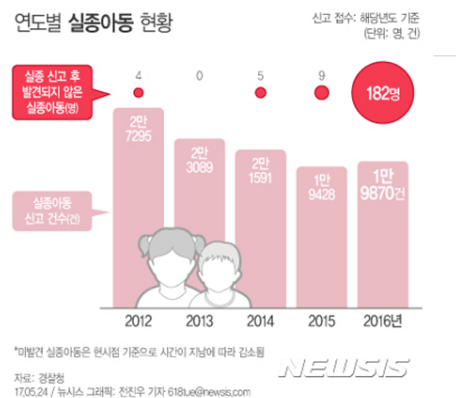
이 가운데 18세 미만 아동 실종 건수도 2015년 606건, 2016년 596건, 지난 해 628건으로 늘어나고 있다.

18세 이상 가출인도 매년 증가하고 있다.

최근 5년 간 18세 이상 가출인 현황은 2013년 1477명, 2014년 1542명, 2015년 1548명, 2016년 1627명, 지난 해 1715명으로 15% 증가했다. 이 중 187명은 가족의 품으로 돌아가지 못하고 있다.

대전지역에선 2016년 지역 여대생이 실종됐다 11일만에 타지역에서 발견되기도 했다.

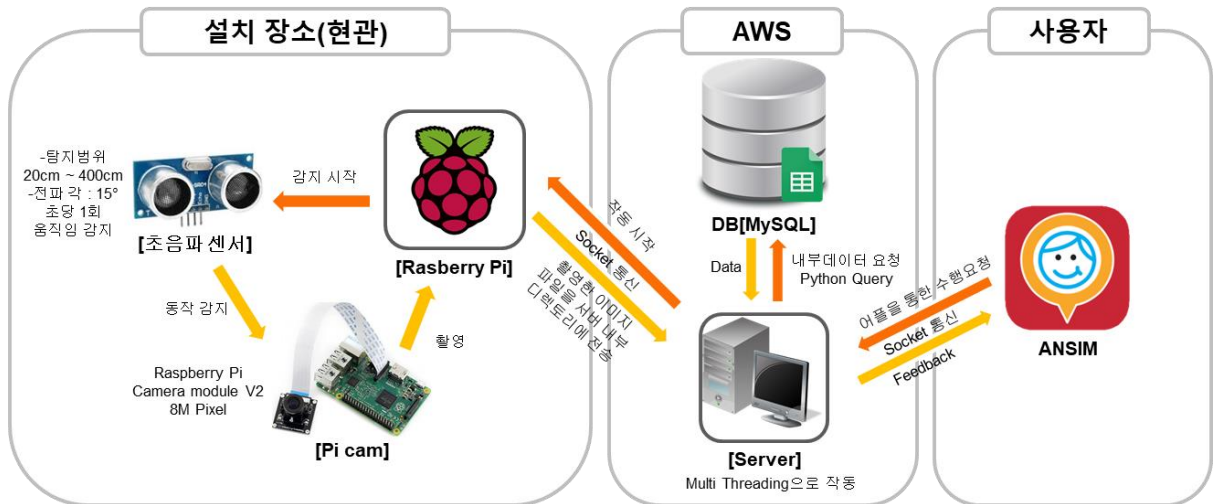
연도별 실종아동 현황



통계에서 보듯 많은 아이들이 하루에 62명씩 사라진다고 한다.
요즘은 맞벌이 부모가 대다수 이다. 이때 부모가 집에 없을때, 제때 들어와야 하는 아이가 안 들어온다면 부모는 불안에 떨며 지내기 십상이다. 이를 예방하기 위해 아이가 언제 집에 들어오는지를 체크하고 보호자가 그에 따라 확인을 받을 수 있는 서비스를 어플을 통해 제공하고자 하였다.

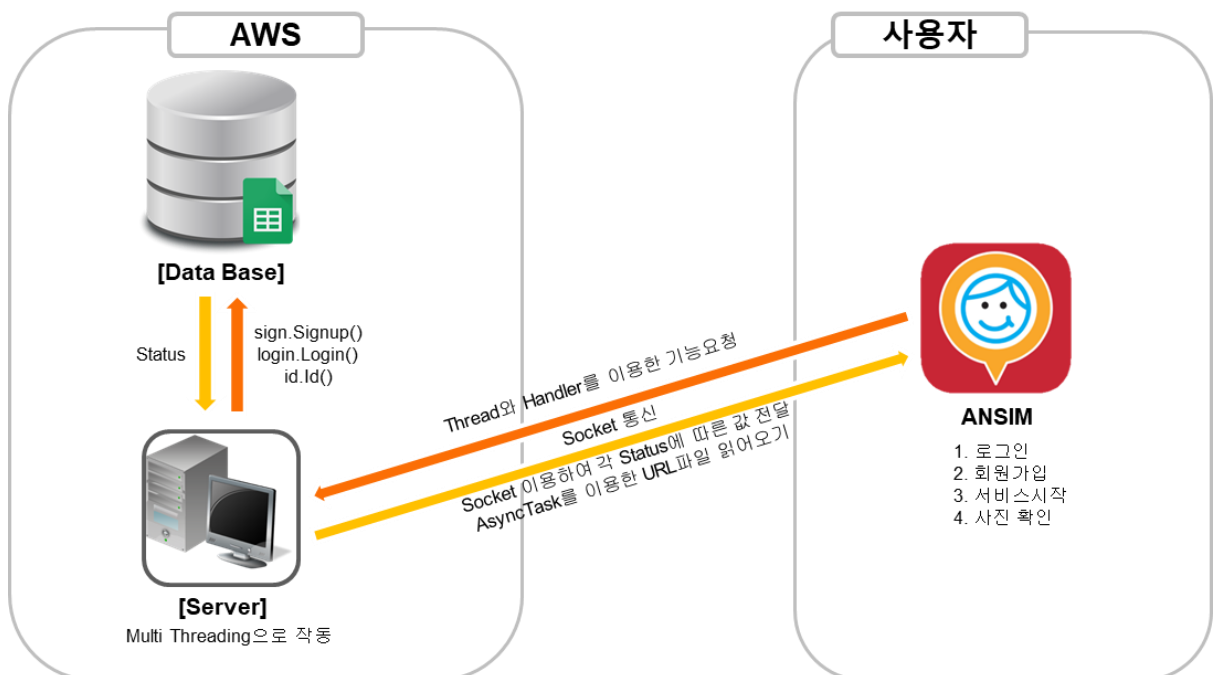
3 시스템 구성 및 설계

3.1 전체 시스템 구성

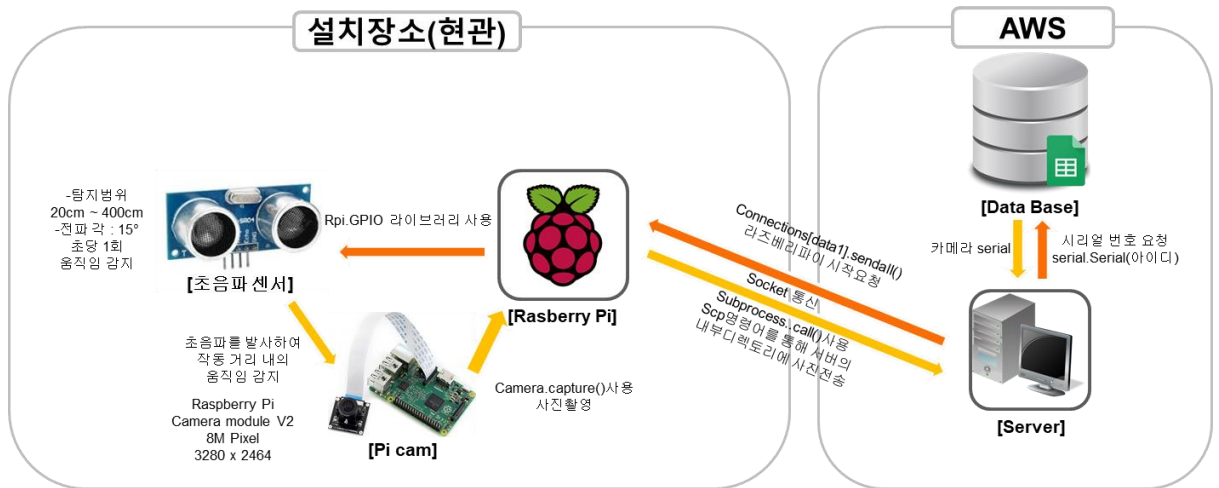


사용자는 어플을 사용해서 socket통신을 통해 서버에 서비스 수행 요청을 하고 서버에서는 DB에 있는 사용자 정보와 비교해서 해당하는 설치장소에 있는 라즈베리파이와 socket통신을 통해 초음파 센서로 동작을 시키고 초음파센서에서 움직임을 감지하면 파이캠을 작동시킨다. 파이캠에서 이미지가 찍히면 서버로 전송해서 사용자는 어플을 통해 feedback을 받게 된다.

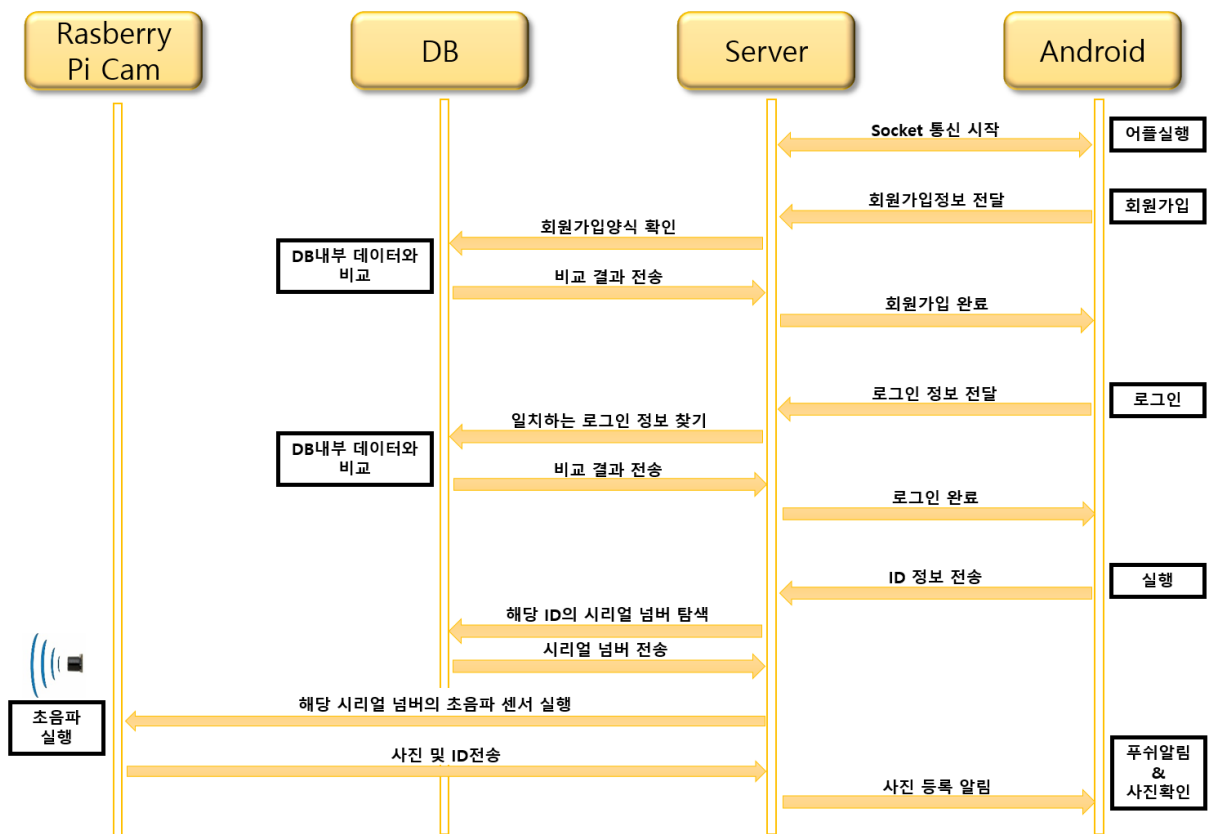
3.2 사용자/서버 구성도



3.3 카메라/서버 구성도



3.4 흐름도



3.5 상세 설계

3.5.1 서버/클라이언트 연결

```
def echo_server(my_port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.bind(('', my_port))
    sock.listen(100)
    print('Server started')
    while True:
        conn, cli_addr = sock.accept()
        socid = conn.recv(1024).decode('utf-8')
        if socid == "L\n" or socid == "S\n":
            pass
        else:
            connections[socid] = conn
            print('Connected by {}'.format(cli_addr))
            handler = threading.Thread(target=echo_handler, args=(conn, cli_addr))
            handler.daemon = True
            handler.start()
```

서버를 열어서 들어오는 소켓들을 연결해주고 demon스레드를 사용하여 여러 client도 연결할 수 있도록 설정해준다.(Server)

```
serial = "4141\n"
def echo_client(server_addr):
    """Echo client"""
    # make TCP/IP socket obj
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect(server_addr) # connect to server process
    sock.send(serial.encode('utf-8'))
if __name__ == '__main__':
    echo_client(('52.79.181.10', 5001))
```

카메라 client에서 연결할 포트번호와 ip주소를 설정해서 연결한다.
연결되면 바로 서버로 serial번호를 보내주게 된다.(Camera client)

```
32     private Socket clientSocket;
33     private BufferedReader socketIn;
34     private PrintWriter socketOut;
35     private int port = 5001;
36     private final String ip = "52.79.181.10";
37     private MyHandler myHandler;
38     private MyThread myThread; //서버 환경 설정
```

어플 client에서 포트번호와 ip주소를 설정해서 연결한다.
실행창 연결할 때는 서버로 로그인한 아이디를 보내주게 된다.(어플 client)

문서명: 인터넷응용 최종보고서

```

59      StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
60      StrictMode.setThreadPolicy(policy);
61
62      try{
63          clientSocket = new Socket(ip ,port);
64          socketIn = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
65          socketOut = new PrintWriter(clientSocket.getOutputStream(), autoFlush: true);
66      }
67      catch (Exception e){
68          e.printStackTrace();
69      } //서버와 통신 시작
70
71      socketOut.println("L"); //서버에게 페이지 알림

```

서버와 통신을 위와 같이 설정해준다. (어플 client)

```

73      myHandler = new MyHandler();
74      myThread = new MyThread(); //서버로부터 데이터를 받는 스레드
75      myThread.start(); //스레드 시작

```

서버로부터 메시지를 받기 위해 Handler와 Thread객체를 선언해준 뒤,

```

99      class MyThread extends Thread{
100          public void run(){
101              while(true){
102                  try{
103                      String data = socketIn.readLine();
104                      Message msg = myHandler.obtainMessage();
105                      msg.obj = data;
106                      myHandler.sendMessage(msg);
107                  }
108                  catch (Exception e){
109                      e.printStackTrace();
110                  }
111              }
112          }
113      }

```

Thread 클래스를 설정해준다. (어플 client)

3.5.2 어플에서의 회원가입

```

68 btn.setOnClickListener((view) -> {
71     String id_s = id.getText().toString();
72     String pwd_s = pwd.getText().toString();
73     String num_s = number.getText().toString();
74     if (id_s.contains("?")){
75         Toast.makeText( context: join.this, text: "?"를 포함한 아이디는 생성 불가합니다", Toast.LENGTH_SHORT).show();
76     }else if (id_s.contains("/")){
77         Toast.makeText( context: join.this, text: "/"를 포함한 아이디는 생성 불가합니다", Toast.LENGTH_SHORT).show();
78     }else if (id_s.contains("-")){
79         Toast.makeText( context: join.this, text: "-"를 포함한 아이디는 생성 불가합니다", Toast.LENGTH_SHORT).show();
80     }else {
81         socketOut.println(id_s + "/" + pwd_s + "/" + num_s); //서버로 메시지 보내기
82     }
83 });

```

어플에서 id, password, camera serial를 입력한 후 회원가입 버튼을 눌렀을 때
id에 몇가지 특수문자를 포함하고 있는 경우를 제외하고
id/password/camera serial의 형태로 연결되어 있는 서버로 보내주게 된다.
(어플 client)

```

elif "/" in data:
    sign = signup.Signup(data) + "\n"
    conn.sendall(sign.encode('utf-8')) # 회원가입 디비 들어갈 곳

```

서버에 "/"를 포함한 data가 들어오면 signup.py의 Signup함수를 불러와서
결과를 어플로 보내준다. (Server)

```

import pymysql

def Signup(str):
    conn = pymysql.connect(host='localhost', user='root', password='1234', db='login', charset='utf8', )
    str = str.replace("\n", "")
    curs = conn.cursor(pymysql.cursors.DictCursor)
    list = str.split('/')

    usr_id = list[0]
    usr_pwd = list[1]
    input_serial = list[2]
    sql = """SELECT serial from user"""
    curs.execute(sql)
    results = curs.fetchall()

    try:
        bool = False
        for sNum in range(len(results)): #시리얼이 등록되었는지 for문으로 돌아가면서 확인
            if results[sNum]['serial'] == int(input_serial):
                sql = ("SELECT id FROM user")
                curs.execute(sql)
                idresults = curs.fetchall()
                for idNum in range(len(idresults)): # 사용중인 아이디를 for문으로 확인하는부분
                    if idresults[idNum]['id'] != None:
                        if idresults[idNum]['id'] == usr_id:
                            return "idError"
                sql = ("SELECT id FROM user WHERE serial = %s"%input_serial)
                curs.execute(sql)
                endResults = curs.fetchall() # 이제 내가 내가 쓸 카메라의 시리얼이 이미 회원가입이 된건지 확인
                if endResults[0]['id'] == None: # 등록하고 싶은 시리얼에 가입이안되서 id가 none인경우
                    bool = True
                    break
                else:
                    return "idError2"
                    break
        if bool == False:
            raise Exception("no serial")
    except Exception as e:
        return "noSerial"
    else:
        sql = ("UPDATE user SET id = '%s' , pwd = '%s' WHERE serial = %s"%(usr_id,usr_pwd ,input_serial))
        curs.execute(sql.encode('utf8'))
        conn.commit()
        return "good"
    finally:
        conn.close()

```

Signup.py 의 Signup 함수로 제대로 회원가입이 되었는지 여부를 결과를 return 하거나 여러가지 예외가 있을 경우 그에 맞는 메시지를 결과를 return 한다.(DB Server)

```

104 class MyHandler extends Handler{
105     public void handleMessage(Message msg){
106         if (msg.obj.toString().equals("good")) { //받은값이 good일 경우
107             Toast.makeText( context.join.this, text: "회원가입이 완료되었습니다!", Toast.LENGTH_SHORT).show();
108             finish(); //창 끝내기
109         }else if(msg.obj.toString().equals("idError")) {
110             Toast.makeText( context.join.this, text: "이미 사용중인 아이디입니다.", Toast.LENGTH_SHORT).show();
111         }else if(msg.obj.toString().equals("idError2")) {
112             Toast.makeText( context.join.this, text: "이미 등록된 시리얼넘버입니다.", Toast.LENGTH_SHORT).show();
113         }else if(msg.obj.toString().equals("noSerial")) {
114             Toast.makeText( context.join.this, text: "시리얼넘버를 잘 못 입력하셨습니다..", Toast.LENGTH_SHORT).show();
115         } //잘못된 회원가입 방식일 경우 토스트로 알람
116     }
117 }

```

서버로부터 결과메세지를 전송받아 Handler 에서 받은 메시지를 판별하여
회원가입여부를 알려준다. (어플 client)

3.5.3 어플에서의 로그인

```

86 btnlogin.setOnClickListener((v) → {
91     sId = editId.getText().toString();
92     sPw = editPassword.getText().toString();
93     intent_sub.putExtra( name: "i_id", sId);
94     socketOut.println(sId + "?" + sPw);
95 }); //로그인하자

```

어플에서 id, password를 입력한 후 로그인 버튼을 눌렀을 때
Id?password의 형태로 연결되어 있는 서버로 보내주게 된다.
(어플 client)

```
elif "?" in data:
    if data.split("?")[0]+"\\n" in connections:
        conn.sendall("already\\n".encode('utf-8'))
    elif data.split("?")[0]+":\\n" in connections:
        conn.sendall("already\\n".encode('utf-8'))
    else:
        log = login.Login(data)          # 로그인 디비 들어갈 곳
        log = log + "\\n"
        conn.sendall(log.encode('utf-8'))
```

서버에 "?"를 포함한 data가 들어오면 연결되어 있는 기기가 있다면 already를 어플 client에 보내주고 연결되어 있는 기기가 없다면 login.py의 Login함수를 불러와서 결과를 어플로 보내준다. (Server)

```
import pymysql

def Login(str1):
    con = pymysql.connect(host='localhost',user='root',password='1234',db='login',charset='utf8')
    cur = con.cursor()
    str1 = str1.replace("\\n","")
    usr2 = str1.split('?')

    usr_id = usr2[0]
    usr_pwd = usr2[1]
    query = ("select pwd from user where id='%s'" %usr_id)
    cur.execute(query)

    results = cur.fetchall()
    result = results[0][0]
    if usr_pwd == result:
        return "success"
    else:
        return "fail"
```

login.py 의 Login 함수로 제대로 DB 안에 입력한 정보가 있는지 없는지 확인한 결과를 return 시켜준다..(DB Server)

```

115 class MyHandler extends Handler{
116     public void handleMessage(Message msg) {
117         if (msg.obj.toString().equals("success")) { //받은값이 true일 경우
118             Toast.makeText( context: MainActivity.this, text: "로그인되었습니다!", Toast.LENGTH_SHORT).show();
119             startActivity(intent_sub); //sub페이지로 넘어감
120         }else if(msg.obj.toString().equals("fail")){
121             Toast.makeText( context: MainActivity.this, text: "아이디 혹은 비밀번호를 확인해주세요 ", Toast.LENGTH_SHORT).show();
122         }else if(msg.obj.toString().equals("already")){
123             Toast.makeText( context: MainActivity.this, text: "이미 접속중인 아이디입니다.", Toast.LENGTH_SHORT).show();
124         }
125     }
126 }

```

회원가입과 마찬가지로 서버로부터 결과메세지를 전송받아 Handler 에서 받은 메시지를 판별하여 로그인여부를 알려준다. (어플 client)

3.5.4 서비스 실행

어플에서 Start 버튼을 눌러주면 id 를 서버로 보내주게 된다.

```

else:
    if "_" not in data:
        data = serial.Serial(data)
        data1 = data + "\n"
        data2 = data + "-\n"
        if data1 in connections:
            connections[data1].sendall(data2.encode('utf-8')) # 아이디 받아서 카메라번호 디비
            conn.sendall("start\n".encode('utf-8'))
        else:
            conn.sendall("finish\n".encode('utf-8'))

```

서버에 id data가 들어오면 딕셔너리에서 해당 카메라 시리얼을 찾아서 value값인 해당 소켓으로 camera serial-을 보내주게된다. (Server)

```
while True:
    gpio.setmode(gpio.BCM)
    data = sock.recv(1024).decode('utf-8')

    if data == (a.replace("\n", "-\n")):
        trig = 13
        echo = 19
        gpio.setup(trig,gpio.OUT)
        gpio.setup(echo,gpio.IN)

        try:
            while True:

                gpio.output(trig,False)
                time.sleep(0.5)
                gpio.output(trig,True)

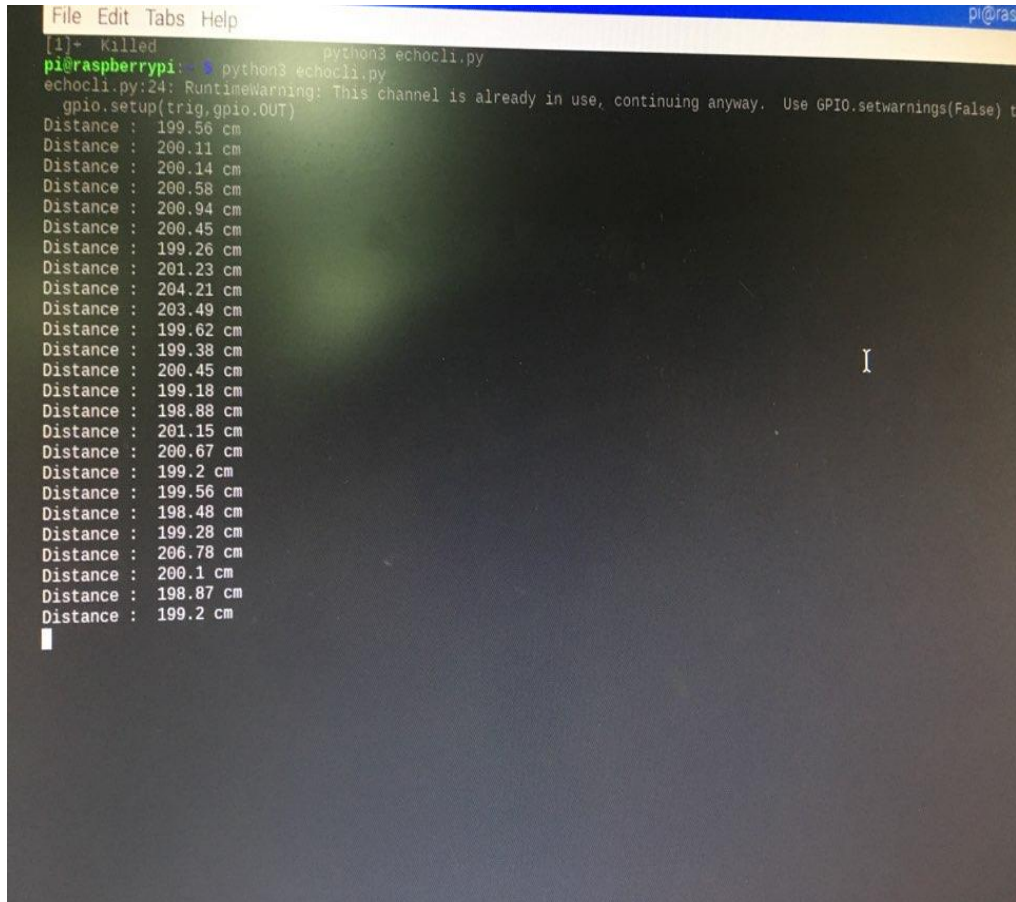
                time.sleep(0.00001)
                gpio.output(trig,False)

                while gpio.input(echo) == 0:
                    pulse_start=time.time()
                while gpio.input(echo) == 1:
                    pulse_end = time.time()

                pulse_duration = pulse_end - pulse_start
                distance = pulse_duration * 17000
                distance = round(distance,2)
```

서버로부터 camera serial-의 data를 받으면 초음파 센서를 실행시킨다.(Camera client)

문서명: 인터넷응용 최종보고서



```

File Edit Tabs Help
[1]- Killed python3 echocli.py
pi@rasberrypi:~$ python3 echocli.py
echocli.py:24: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to
  gpio.setup(trig, gpio.OUT)
Distance : 199.56 cm
Distance : 200.11 cm
Distance : 200.14 cm
Distance : 200.58 cm
Distance : 200.94 cm
Distance : 200.45 cm
Distance : 199.26 cm
Distance : 201.23 cm
Distance : 204.21 cm
Distance : 203.49 cm
Distance : 199.62 cm
Distance : 199.38 cm
Distance : 200.45 cm
Distance : 199.18 cm
Distance : 198.88 cm
Distance : 201.15 cm
Distance : 200.67 cm
Distance : 199.2 cm
Distance : 199.56 cm
Distance : 198.48 cm
Distance : 199.28 cm
Distance : 206.78 cm
Distance : 200.1 cm
Distance : 198.87 cm
Distance : 199.2 cm

```

초음파가 실행되고 있는 화면이다.

```

if distance < 100:
    with picamera.PiCamera() as camera:
        camera.start_preview()
        time.sleep(1)
        camera.capture('/var/www/html/4141.jpg')
        camera.stop_preview()
    subprocess.call(['scp -i /home/pi/tean2/AWS_enova_psvd.pem /var/www/html/4141.jpg ubuntu@52.79.181.10:/var/www/html', shell = True])
    print(data)
    sock.send(data.encode('utf-8'))
    data = sock.recv(1024).decode('utf-8')

```

초음파의 거리가 100m 이내로 들어왔을 때 사진을 찍어서 연결되어 있는 서버로 보내주고 서버에 찍은 camera serial-를 보내주게 된다.(Camera client)

문서명: 인터넷응용 최종보고서



100m 이내로 들어왔을 때 파이캠이 찍히는 화면이다.

초음파 센서는 허용 거리 범위가 400 ~ 500cm 로 현관문에서 사용하는 현 서비스에서 적합하다.

```
if "-" in data:
    chid = data.split("-")
    chid = chid[0] + "\n"
    data = id.Id(chid)
    data = data + "\n"
    connections[data].sendall(chid.encode('utf-8')) # 카메라번호 받아서 아이디 디비
```

서버에 "-"를 포함한 data가 들어오면 id.py의 Id함수를 불러와서 data에 id를 저장시킨 후 해당 아이디로 연결되어 있는 어플로 보내준다. (Server)

```
import pymysql

def Id(serial):
    serial = serial.replace("\n", "")
    conn = pymysql.connect(host='localhost', user='root', password='1234', db='login', charset='utf8')

    curs = conn.cursor(pymysql.cursors.DictCursor)

    query = ("select id from user where serial =%s" %serial)
    curs.execute(query)

    results = curs.fetchall()

    result = results[0]['id']
    return result
```

id.py 의 Id 함수로 들어온 serial 에 저장되어 있는 아이디를 찾아서 return 해준다.

(DB Server)

```

100      class MyThread extends Thread{...}
115
116      class MyHandler extends Handler{
117          public void handleMessage(Message msg) {
118              serial = msg.obj.toString();
119              temp = msg.obj.toString();
120              if ((serial != null) && (!serial.equals("start"))){
121                  notificationManager();
122                  serial = null;
123                  fin.setVisibility(View.GONE);
124                  chk.setVisibility(View.VISIBLE);
125              }
126          }
127      }

```

서버로부터 사진등록 알림을 받고 어플리케이션에서 notification함수를 이용하여 push알림을 준다. 동시에 사용자가 추가 조작할 수 있도록 버튼을 설정해준다.
(어플 client)

```

128      private class back extends AsyncTask<String, Integer, Bitmap>{
129
130          @Override
131          protected Bitmap doInBackground(String... urls) {
132              // TODO Auto-generated method stub
133              try{
134                  URL myFileUrl = new URL(urls[0]);
135                  HttpURLConnection conn = (HttpURLConnection)myFileUrl.openConnection();
136                  conn.setDoInput(true);
137                  conn.connect();
138
139                  InputStream is = conn.getInputStream();
140
141                  bmlmg = BitmapFactory.decodeStream(is);
142
143              }catch(IOException e){
144                  e.printStackTrace();
145              }
146              return bmlmg;
147          }
148
149          protected void onPostExecute(Bitmap img) { imageView.setImageBitmap(bmlmg); }
150
151      }
152
153      } //url 확인 클래스

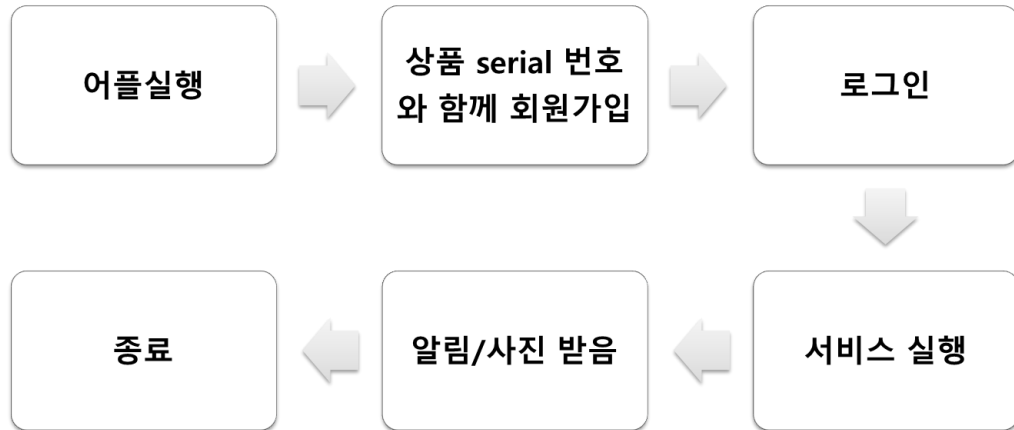
```

```
case R.id.chk:
    task = new back(); //여기부터 3줄은 사진읽어오기
    imageView = (ImageView) findViewById(R.id.imageView1);
    task.execute(imgUrl+ temp + ".jpg"); //읽어올 파일의 이름
    imageView.setVisibility(View.VISIBLE);
    c.setVisibility(View.VISIBLE);
    f.setVisibility(View.VISIBLE);
    break;
```

알림을 받는다면 클라이언트에서 사진확인버튼을 눌러 사진을 확인 할 수 있다.
웹서버에 올려진 url 을 확인하는 class 를 만들어준 뒤 버튼클릭이벤트가 일어날
시 imageView 가 url 의 사진을 가져온다.(어플 client)

4 시스템 시나리오

4.1 순차 다이어그램



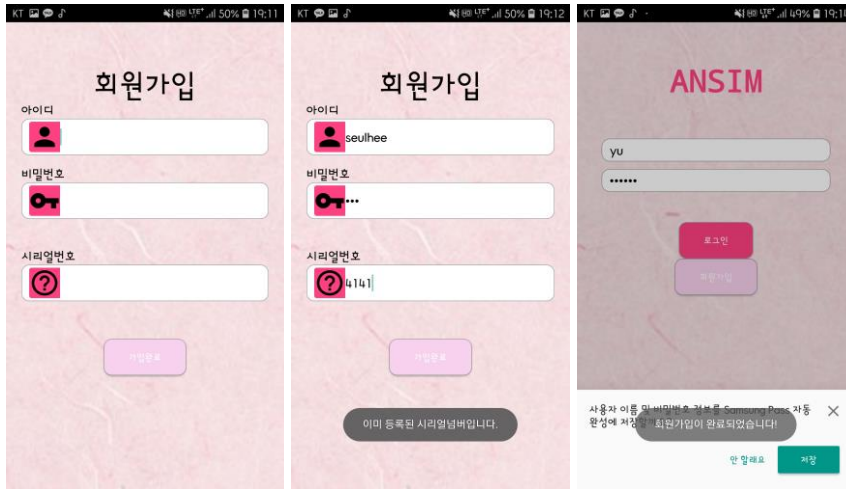
4.2 시나리오

10살 아이를 둔 맞벌이 부모가 자녀의 안전한 귀가를 확인하기 위해 상품을 구입하고 해당 상품의 serial 번호를 받았다.

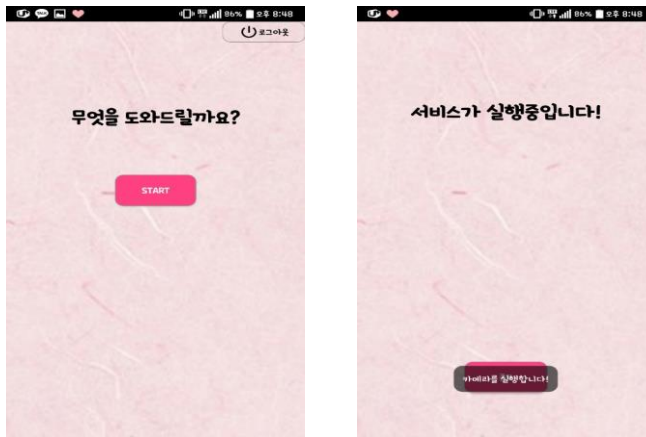
사용자인 부모가 serial 번호와 함께 어플에서 회원가입을 하고 서비스 실행 권한을 받았다.

하루는 맞벌이 부모가 모두 늦게 귀가 하기 때문에 아이가 학원에 갔다가 언제 잘 도착했는지 확인하고 싶어서 어플을 켜서 서비스를 실행하였다. 아이가 학원이 끝나고 집에 돌아가는 길에 핸드폰 배터리가 방전되어서 보호자에게 연락을 하지 못하였다. 보호자는 항상 아이에게 연락이 오는 시간에도 연락 오지 않자 불안해 하던 와중에 어플 알림을 통해 사진을 확인하여 아이가 집에 잘 도착했다는 걸 확인하고 안심하였다.

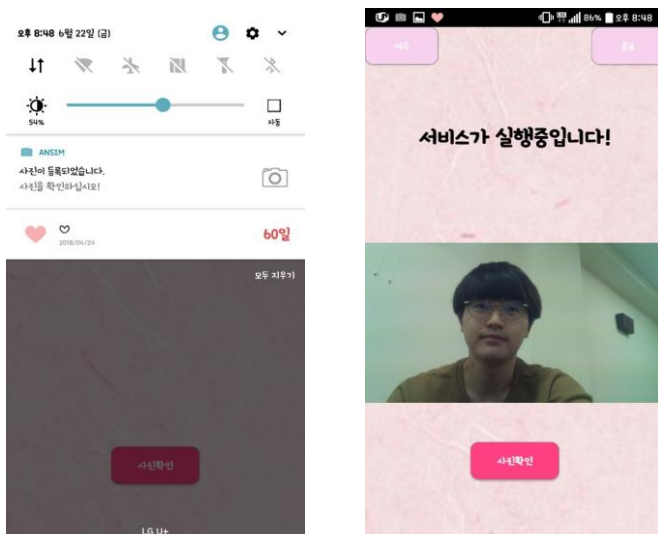
4.3 시나리오 어플 실행 순서



왼쪽사진은
회원가입 버튼을
누른 후의
모습이다.
가운데 사진은
ID 와

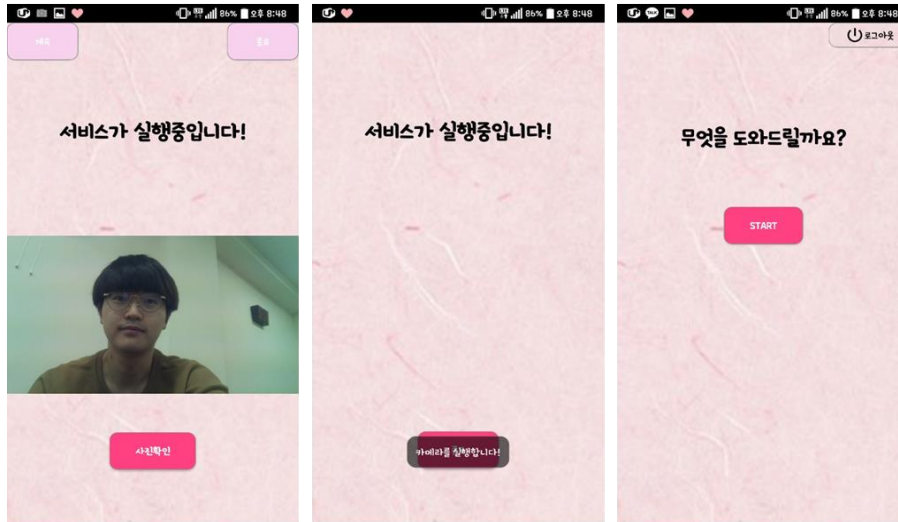


왼쪽사진을 보면 로그인을 한 뒤에
나오게 되는 화면이다. 여기서
START 버튼을 누르면 오른쪽사진으로
넘어가게 된다.
오른쪽 사진은 START 버튼을 누르면
초음파센서가 실행되며, 서비스가
실행된다는 토스트와 함께 구동된다.



왼쪽사진은 초음파센서에 물체가
인식되어, Raspberry Cam 으로
사진을 찍은 후 보호자에게
푸시알림이 오게 된 모습이다.
푸시알림을 누르게 되면,
오른쪽 사진창으로 넘어오게
되는데, 사진확인 버튼을 누르면
Raspberry Pi Cam 이 촬영한
사진이 업로드 된다.

문서명: 인터넷응용 최종보고서



왼쪽사진 상단의
계속버튼을 누르게 되면,

사진이 사라진 뒤
서비스가 재시작 되는
모습이다.(가운데사진)

왼쪽사진 상단의
종료버튼을 누르게 되면,

어플을 구동하는
버튼(START)이 있는 전
페이지로 넘어간다.
(오른쪽사진)

5 기대효과

현관문에서 초음파센서로 움직임을 감지해서 사진을 전송하는 시스템이기 때문에 안전 귀가 뿐 아니라 보호자가 자녀나 / 치매노인이 언제 외출을 시도 했는지 역시 확인 할 수 있다. 또한 집에 아무도 없을 때 외부 침입이 있었는지 실시간으로 알림을 받아 볼 수 있으므로 이 상품은 단순히 안심귀가 서비스 뿐 아닌 집의 출입 서비스를 제공할 수 있다.

6 프로젝트 세부일정

| 기간 개발내용 | 개발 단계 | | | |
|----------------------------------|-----------|------------|------------|-------------|
| | 5월 | | 6월 | |
| | 5.18~5.25 | 5.26 ~ 6.1 | 6.2 ~ 6.10 | 6.21 ~ 6.23 |
| 아이디어 회의 및 역할 분담 | | | | |
| 제안서 작성 및 검토 | | | | |
| Raspberry Pi 및 센서 분석 | | | | |
| 어플 구성도 작성 | | | | |
| 아이디어 재회의 및 제안서 작성 | | | | |
| Server 구축 | | | | |
| Client(App) 구축 | | | | |
| Socket 통신 구축 | | | | |
| Raspberry Pi 및 센서 개발 | | | | |
| Raspberry Pi 와 Server connection | | | | |
| Database를 이용한 로그인/회원가입 구현 | | | | |
| 코드 추합 및 전체 연결 | | | | |
| App Design | | | | |
| 최종 작동 확인 | | | | |
| 최종보고서 작성 | | | | |
| PPT 제작 | | | | |

문서명: 인터넷응용 최종보고서

7 팀원 담당업무

| 담당자 | 상세내용 |
|-------|-------------------------------|
| 류 형 오 | 총괄 및 서버/클라이언트 소켓 통신 |
| 유 한 석 | 중간 서버/DB 구축 |
| 김 혜 지 | 중간 서버/DB 구축 |
| 전 진 우 | Android 클라이언트 기능 및 서버 간 소켓 통신 |
| 박 슬 희 | Android 클라이언트 UI 디자인 및 제작 |
| 김 강 연 | Android 클라이언트 UI 디자인 및 제작 |

8 참고 문헌

8.1 관련 문서

| 출판사 | 문서 제목 |
|---------|----------------------------|
| 이지스퍼블리싱 | Do It! 안드로이드 앱 프로그래밍-7.0버전 |
| 한빛미디어 | 데이터베이스 첫걸음 |

8.2 참고 사이트

| Title | URL |
|-------------------------------------|---|
| Android 스레드 | http://itmining.tistory.com/4 |
| Android url읽기 | http://cholol.tistory.com/154 |
| 파이썬 서버와 Android 소켓통신 | http://mititch.tistory.com/36 |
| Android push알림 | http://www.joshi.co.kr/index.php?mid=board_QBE_S95&document_srl=13912 |
| Android intent | http://foradun.tistory.com/entry/Activity-Single-Instance-%EC%9C%A0%EC%A7%80%ED%95%98%EA%B8%B0 |
| Android UI 설정 | https://blog.naver.com/eominsuk55/220222752164 |
| 라즈베리파이 subprocess 명령어로 shell 명령어 치기 | http://noplanlife.com/?p=949 |
| aws scp 사용법 | http://ourcstory.tistory.com/162 |
| 라즈베리파이 초음파센서 | https://m.blog.naver.com/PostView.nhn?blogId=roboholic84&logNo=220319850312&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F |
| 파이썬 파이썬 사용 | http://pinkwink.kr/847 |
| 파이썬과 MySQL 연동 | http://pythonstudy.xyz/python/article/202-MySQL-%EC%BF%BC%EB%A6%AC |
| EC2에 Database구축 | http://ljs93kr.tistory.com/42 |

9 [부록]ANSIM_source code

9.1 Server(AWS) source code

```
#aws npserver.py
#라즈베리파이캠 과 어플에서 정보를 받고 그에 맞는 정보를 보내줌.
import socket
import threading, logging
import login
import signup
import serial
import id
import subprocess

connections = {}
socid = ''

def echo_server(my_port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.bind(('', my_port))
    sock.listen(100)
    print('Server started')
    while True:
        conn, cli_addr = sock.accept()
        socid = conn.recv(1024).decode('utf-8')
        if socid == "LWn" or socid == "SWn":
            pass
        else:
            connections[socid] = conn
            print('Connected by {}'.format(cli_addr))
            handler = threading.Thread(target=echo_handler, args=(conn, cli_addr))
            handler.daemon = True
            handler.start()
def echo_handler(conn, cli_addr):
    while True:
        try:
            data = conn.recv(1024).decode('utf-8')
            if not data:
                try:
                    for name, value in connections.items():
                        if value == conn:
                            data = name
                if data in connections:
                    if ":" not in data:
                        x = serial.Serial(data)
                        subprocess.call('sudo rm -r /var/www/html/' + x + '.jpg',
shell=True)
            del connections[data]
```

```

        print('Client closing: {}'.format(cli_addr))
        break # 연결 끝길때 서버에 사진지우고 끝는 부분
    except:
        print('Client closing: {}'.format(cli_addr))
        break # 카메라 연결 끝인 부분
print('Received: {}'.format(data))
if "-" in data:
    chid = data.split("-")
    chid = chid[0] + "\n"
    data = id.Id(chid)
    data = data + "\n"
    connections[data].sendall(chid.encode('utf-8')) # 카메라번호 받아서
아이디 디비
elif "/" in data:
    sign = signup.Signup(data) + "\n"
    conn.sendall(sign.encode('utf-8')) # 회원가입 디비 들어갈 곳
elif "?" in data:
    if data.split("?")[0]+"\n" in connections:
        conn.sendall("already\n".encode('utf-8'))
    elif data.split("?")[0]+"\n" in connections:
        conn.sendall("already\n".encode('utf-8'))
    else:
        log = login.Login(data) # 로그인 디비 들어갈 곳
        log = log + "\n"
        conn.sendall(log.encode('utf-8'))
elif "logout" in data:
    for name, value in connections.items():
        if value == conn:
            data = name
    if data in connections:
        del connections[data]
    data = data.replace(":\n", "\n")
    if data in connections:
        del connections[data] # logout 시 디렉터리에서 카메라 소켓 제거
else:
    if "_" not in data:
        data = serial.Serial(data)
        data1 = data + "\n"
        data2 = data + "-\n"
        if data1 in connections:
            connections[data1].sendall(data2.encode('utf-8')) # 아이디
            받아서 카메라번호 디비
            conn.sendall("start\n".encode('utf-8'))
        else:
            conn.sendall("finish\n".encode('utf-8'))
    elif "_" in data:
        data = data.replace('_\n', '\n')
        data = serial.Serial(data)
        subprocess.call('sudo rm -r /var/www/html/'+data+'.jpg',
shell=True)
        data1 = data + "\n"

```

```
data2 = data + "\n"
connections[data1].sendall(data2.encode('utf-8')) # 종료
    눌렀을시 대기상태

except socket.error as e:
    print('socket error: {}'.format(e))
    break
conn.close()

if __name__ == '__main__':
    echo_server(5001)
```

9.2 Android source code

```
MainActivity.java
package com.example.ruldy.test1;

import android.content.Context;
import android.content.Intent;
import android.os.Vibrator;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.InputType;
import android.text.method.PasswordTransformationMethod;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import java.io.BufferedReader;
import java.io.IOException;
```

```
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class MainActivity extends AppCompatActivity {

    Button btn;

    private Socket clientSocket;
    private BufferedReader socketIn;
    private PrintWriter socketOut;
    private int port = 5001;
    private final String ip = "52.79.181.10";
    private MyHandler myHandler;
    private MyThread myThread; //서버 환경 설정

    EditText editId, editPassword, editPasswordchk;
    String sId, sPw; //여러곳에서 쓰게 전역변수로 설정
    Intent intent_join, intent_sub;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editId = (EditText) findViewById(R.id.edit_id); //버튼 변수 설정
        editPassword = (EditText) findViewById(R.id.edit_password);
        editPassword.setInputType(
            InputType.TYPE_CLASS_TEXT
            InputType.TYPE_TEXT_VARIATION_PASSWORD );

        editPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());

        Button btnlogin = (Button) findViewById(R.id.btn_login);
        Button btnjoin = (Button) findViewById(R.id.btn_join); //변수 설정 마침

        intent_sub = new Intent(this, SubActivity.class);
        intent_join = new Intent(this, join.class); //페이지 넘기는 intent 설정

        StrictMode.ThreadPolicy
            policy
            =
            new
```

```
StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    try{
        clientSocket = new Socket(ip ,port);
        socketIn = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
    }
    catch (Exception e){
        e.printStackTrace();
    } //서버와 통신 시작

    socketOut.println("L"); //서버에게 페이지 알림

    myHandler = new MyHandler();
    myThread = new MyThread(); //서버로부터 데이터를 받는 스레드
    myThread.start(); //스레드 시작

    btnjoin.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            startActivity(intent_join);
        }
    }); //회원가입창으로 이동

    btnlogin.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            sId = editId.getText().toString();
            sPw = editPassword.getText().toString();
            intent_sub.putExtra("i_id", sId);
            socketOut.println(sId + "?" + sPw);
        }
    })
```



```

    }); //로그인하자

}

class MyThread extends Thread{
    public void run(){
        while(true){
            try{
                String data = socketIn.readLine();
                Message msg = myHandler.obtainMessage();
                msg.obj = data;
                myHandler.sendMessage(msg);
            }
            catch (Exception e){
                e.printStackTrace();
            }
        }
    }
}

class MyHandler extends Handler{
    public void handleMessage(Message msg) {
        if (msg.obj.toString().equals("success")) { //받은값이 true일 경우
            Toast.makeText(MainActivity.this, "로그인되었습니다!",
                Toast.LENGTH_SHORT).show();
            startActivity(intent_sub); //sub페이지로 넘어감
        } else if(msg.obj.toString().equals("fail")){
            Toast.makeText(MainActivity.this, "아이디 혹은 비밀번호를 확인해주세요.",
                Toast.LENGTH_SHORT).show();
        } else if(msg.obj.toString().equals("already")){
            Toast.makeText(MainActivity.this, "이미 접속중인 아이디입니다.",
                Toast.LENGTH_SHORT).show();
        }
    }
}

}

activity_main.xml
<?xml version="1.0" encoding="utf-8"?>

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:orientation="vertical"
    android:background="@drawable/background" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="50dp"
        android:text="ANSIM"
        android:textColor="@color/colorAccent"
        android:textSize="40dp"
        android:textStyle="bold"
        android:typeface="monospace" />

    <EditText
        android:id="@+id/edit_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="32dp"
        android:layout_marginRight="32dp"
        android:background="@drawable/edit_main"
        android:ems="10"
        android:hint="아이디"
        android:textColorHint="@color/colorGray"
        android:textColor="@color/colorBlack"/>

    <EditText
        android:id="@+id/edit_password"
        android:layout_width="match_parent"

```

```
android:layout_height="wrap_content"
android:layout_marginLeft="32dp"
android:layout_marginRight="32dp"
android:layout_marginTop="10dp"
android:background="@drawable/edit_main"
android:ems="10"
android:hint="비밀번호"
android:textColorHint="@color/colorGray"
android:textColor="@color/colorBlack"
android:inputType="textPassword"/>
```

<Button

```
android:id="@+id/btn_login"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginLeft="32dp"
android:layout_marginRight="32dp"
android:layout_marginTop="40dp"
android:background="@drawable/button_main"
android:text="로그인"
android:textColor="@color/colorWhite" />
```

<Button

```
android:id="@+id/btn_join"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:background="@drawable/button_three"
android:text="회원가입"
android:textAppearance="@style/TextAppearance.AppCompat" />
```

</LinearLayout>

SubActivity.java

package com.example.ruldy.test1;

import android.content.Intent;

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import org.w3c.dom.Text;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class SubActivity extends AppCompatActivity {

    private Socket clientSocket;
    private BufferedReader socketIn;
    private PrintWriter socketOut;
    private int port = 5001;
    private final String ip = "52.79.181.10";
    private MyHandler myHandler;
    private MyThread myThread; //서버환경설정

    Intent intent_run;
    String id_data;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub);
    }
}
```

```
Intent intent_id = getIntent(); //intent해서
id_data = intent_id.getStringExtra("i_id"); //아이디값 받아오기
intent_run = new Intent(this, RunActivity.class); //run페이지로 넘김
intent_run.putExtra("i_id_run", id_data);

StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);

try{
    clientSocket = new Socket(ip ,port);
    socketIn = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
    socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
}
catch (Exception e){
    e.printStackTrace();
} //서버와 통신시작

socketOut.println(id_data+ ":"); //서버에게 페이지 알림

myHandler = new MyHandler();
myThread = new MyThread(); //서버의 값을 받는 스레드 설정
myThread.start(); //스레드 시작

Button btn_run = (Button) findViewById(R.id.button_run);
Button btn_logout = (Button) findViewById(R.id.button_logout);

btn_run.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        socketOut.println(id_data); //
    }
}); //실행창으로 넘어감
```

```

        btn_logout.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                socketOut.println("logout"); //
                finish();
            }
        }); //로그아웃

    }

    class MyThread extends Thread{
        public void run(){
            while(true){
                try{
                    String data = socketIn.readLine();
                    Message msg = myHandler.obtainMessage();
                    msg.obj = data;
                    myHandler.sendMessage(msg);
                }
                catch (Exception e){
                    e.printStackTrace();
                }
            }
        }
    }

    class MyHandler extends Handler{
        public void handleMessage(Message msg){
            if (msg.obj.toString().equals("start")) { //받은값이 true일 경우
                Toast.makeText(SubActivity.this, "카메라를 실행합니다!",
                    Toast.LENGTH_SHORT).show();
                startActivity(intent_run); //sub페이지로 넘어감
            } else if(msg.obj.toString().equals("finish")) {
                Toast.makeText(SubActivity.this, "카메라의 연결상태를 확인하세요.",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```
}

activity_sub.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background" >

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="127dp"
        android:textStyle="bold"
        android:text="무엇을 도와드릴까요?"
        android:textSize="30sp"
        android:textColor="@color/colorBlack"/>

    <Button
        android:id="@+id/button_logout"
        android:layout_width="wrap_content"
        android:layout_height="32dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentTop="true"
        android:background="@drawable/button_sub"
        android:drawableLeft="@drawable/ic_power_settings_new_black_24dp"
        android:text="로그아웃"
        android:textColor="@color/colorBlack"/>

    <Button
        android:id="@+id/button_run"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="236dp"
android:background="@drawable/button_main"
android:text="START"
android:textColor="@color/colorWhite"
android:textStyle="bold" />
```

</RelativeLayout>

```
Join.java
package com.example.ruldy.test1;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.text.InputType;
import android.text.method.PasswordTransformationMethod;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class join extends AppCompatActivity {

    Button btn, btn_idchk;
    EditText number, id, pwd;
    Intent intent_main;
```



```
private Socket clientSocket;
private BufferedReader socketIn;
private PrintWriter socketOut;
private int port = 5001;
private final String ip = "52.79.181.10";
private MyHandler myHandler;
private MyThread myThread;

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

    intent_main = new Intent(this, MainActivity.class);

    StrictMode.ThreadPolicy policy = new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    try{
        clientSocket = new Socket(ip ,port);
        socketIn = new BufferedReader(new
        InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
    }
    catch (Exception e){
        e.printStackTrace();
    } //서버와 연결시작

    socketOut.println("S"); //서버에게 페이지 알림

    myHandler = new MyHandler();
    myThread = new MyThread(); //서버로부터 데이터를 받는 스레드
    myThread.start(); //스레드 시작

    btn = (Button) findViewById(R.id.btn);
    id = (EditText) findViewById(R.id.id);
    pwd = (EditText) findViewById(R.id.pwd);
```

```

        pwd.setInputType(                InputType.TYPE_CLASS_TEXT
        InputType.TYPE_TEXT_VARIATION_PASSWORD );

        pwd.setTransformationMethod(PasswordTransformationMethod.getInstance());
        number = (EditText) findViewById(R.id.number); //변수설정

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String id_s = id.getText().toString();
                String pwd_s = pwd.getText().toString();
                String num_s = number.getText().toString();
                if (id_s.contains("?")){
                    Toast.makeText(join.this, "'?'를 포함한 아이디는 생성 불가능합니다",
                    Toast.LENGTH_SHORT).show();
                }else if (id_s.contains("/")){
                    Toast.makeText(join.this, "'/'를 포함한 아이디는 생성 불가능합니다",
                    Toast.LENGTH_SHORT).show();
                }else if (id_s.contains("-")){
                    Toast.makeText(join.this, "'-'를 포함한 아이디는 생성 불가능합니다",
                    Toast.LENGTH_SHORT).show();
                }else {
                    socketOut.println(id_s + "/" + pwd_s + "/" + num_s); //서버로 메시지
                    보내기
                }
            }
        });
    }

    class MyThread extends Thread{
        public void run(){
            while(true){
                try{
                    String data = socketIn.readLine();
                    Message msg = myHandler.obtainMessage();
                    msg.obj = data;
                    myHandler.sendMessage(msg);
                }
            }
        }
    }

```

```

        catch (Exception e){
            e.printStackTrace();
        }
    }
}

class MyHandler extends Handler{
    public void handleMessage(Message msg){
        if (msg.obj.toString().equals("good")) { //받은값이 good일 경우
            Toast.makeText(join.this, "회원가입이 완료되었습니다!",
                Toast.LENGTH_SHORT).show();
            finish(); //창 끝내기
        } else if(msg.obj.toString().equals("idError")) {
            Toast.makeText(join.this, "이미 사용중인 아이디입니다.",
                Toast.LENGTH_SHORT).show();
        } else if(msg.obj.toString().equals("idError2")) {
            Toast.makeText(join.this, "이미 등록된 시리얼넘버입니다.",
                Toast.LENGTH_SHORT).show();
        } else if(msg.obj.toString().equals("noSerial")) {
            Toast.makeText(join.this, "시리얼넘버를 잘 못 입력하셨습니다..",
                Toast.LENGTH_SHORT).show();
        } //잘못된 회원가입 방식일 경우 토스트로 알림
    }
}

activity_join.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background">

    <EditText

```

```

android:id="@+id/id"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:layout_marginTop="127dp"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:background="@drawable/edit_main"
android:drawableLeft="@drawable/username"
android:hint="아이디"
android:textColor="@color/colorBlack"/>

```

<EditText

```

android:id="@+id/pwd"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:layout_marginTop="211dp"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:background="@drawable/edit_main"
android:drawableLeft="@drawable/password"
android:hint="비밀번호"
android:textColor="@color/colorBlack"
android:inputType="textPassword"/>

```

<EditText

```

android:id="@+id/number"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:layout_below="@+id/textView5"
android:background="@drawable/edit_main"
android:drawableLeft="@drawable/number"

```

```
android:hint=" 시리얼번호"  
android:textColor="@color/colorBlack"/>
```

```
<Button
```

```
    android:id="@+id/btn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_centerHorizontal="true"  
    android:layout_gravity="center_horizontal"  
    android:layout_marginBottom="151dp"  
    android:background="@drawable/button_three"  
    android:text="가입완료" />
```

```
<TextView
```

```
    android:id="@+id/textView2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="55dp"  
    android:text="회원가입"  
    android:textSize="40dp"  
    android:textColor="@color/colorBlack"/>
```

```
<TextView
```

```
    android:id="@+id/textView3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_above="@+id/id"  
    android:layout_alignStart="@+id/id"  
    android:layout_marginBottom="-127dp"  
    android:textSize="17dp"  
    android:text="아이디"  
    android:textColor="@color/colorBlack"/>
```

```
<TextView
```

```
    android:id="@+id/textView4"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/pwd"
        android:layout_alignStart="@+id/id"
        android:layout_marginBottom="-211dp"
        android:textSize="17dp"
        android:text="비밀번호"
        android:textColor="@color/colorBlack"/>

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/id"
    android:layout_centerVertical="true"
    android:textSize="17dp"
    android:text="시리얼번호"
    android:textColor="@color/colorBlack"/>
</RelativeLayout>

```

```

RunActivity.java
package com.example.ruldy.test1;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.os.Message;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

```

```
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.HttpURLConnection;
import java.net.Socket;
import java.net.URL;
import java.net.URLConnection;
//import org.apache.http.util.ByteArrayBuffer;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.widget.ImageView;
import org.json.JSONObject;

import static android.service.autofill.Validators.and;

public class RunActivity extends AppCompatActivity {

    ImageView imageView;
    String imgUrl = "http://52.79.181.10/"; //읽어올 파일의 주소
    Bitmap bmlmg;
    back task;
```

```
String serial, id_data, temp;
View f, c, chk, fin;

private Socket clientSocket;
private BufferedReader socketIn;
private PrintWriter socketOut;
private int port = 5001;
private final String ip = "52.79.181.10";
private MyHandler myHandler;
private MyThread myThread; //서버 환경 설정

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

    Intent intent_id = getIntent(); //intent해서
    id_data = intent_id.getStringExtra("i_id_run"); //아이디값 받아오기

    f = findViewById(R.id.finish);
    c = findViewById(R.id.contin);
    chk = findViewById(R.id.chk);
    fin = findViewById(R.id.close);

    StrictMode.ThreadPolicy policy = new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    try{
        clientSocket = new Socket(ip ,port);
        socketIn = new BufferedReader(new
        InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
    }
    catch (Exception e){
        e.printStackTrace();
    } //서버와 통신 시작
```



```
socketOut.println(id_data); //서버에게 페이지 알림

myHandler = new MyHandler();
myThread = new MyThread(); //서버로부터 데이터를 받는 스레드
myThread.start(); //스레드 시작
}

class MyThread extends Thread{
    public void run(){
        while(true){
            try{
                String data = socketIn.readLine();
                Message msg = myHandler.obtainMessage();
                msg.obj = data;
                myHandler.sendMessage(msg);
            }
            catch (Exception e){
                e.printStackTrace();
            }
        }
    }
}

class MyHandler extends Handler{
    public void handleMessage(Message msg) {
        serial = msg.obj.toString();
        temp = msg.obj.toString();
        if ((serial != null) && (!serial.equals("start"))){
            notificationManager();
            serial = null;
            fin.setVisibility(View.GONE);
            chk.setVisibility(View.VISIBLE);
        }
    }
}

private class back extends AsyncTask<String, Integer, Bitmap>{
```

```
@Override
protected Bitmap doInBackground(String... urls) {
    // TODO Auto-generated method stub
    try{
        URL myFileUrl = new URL(urls[0]);
        HttpURLConnection conn =
(HttpURLConnection)myFileUrl.openConnection();
        conn.setDoInput(true);
        conn.connect();

        InputStream is = conn.getInputStream();

        bmlmg = BitmapFactory.decodeStream(is);

    }catch(IOException e){
        e.printStackTrace();
    }
    return bmlmg;
}

protected void onPostExecute(Bitmap img){
    imageView.setImageBitmap(bmlmg);
}

} //url 확인 클래스

public void onClick(View view){
    switch (view.getId()){

        case R.id.close:
            finish();
            break;

        case R.id.chk:
            task = new back(); //여기부터 3줄은 사진읽어오기
            imageView = (ImageView) findViewById(R.id.imageView1);
            task.execute(imgUrl+ temp + ".jpg"); //읽어올 파일의 이름
            imageView.setVisibility(View.VISIBLE);
```

```

        c.setVisibility(View.VISIBLE);
        f.setVisibility(View.VISIBLE);
        break;

    case R.id.contin:
        imView.setVisibility(View.GONE);
        c.setVisibility(View.GONE);
        f.setVisibility(View.GONE);
        fin.setVisibility(View.VISIBLE);
        chk.setVisibility(View.GONE);
        socketOut.println(id_data); //서버에게 페이지 알림
        break;

    case R.id.finish:
        socketOut.println(id_data + "_"); //서버에게 페이지 알림
        finish();
        break;
    }
}

public void notificationManager(){
    NotificationManager notificationManager = (NotificationManager)
    getSystemService(Context.NOTIFICATION_SERVICE);

    Intent intent = new Intent(this, RunActivity.class); //다른 액티비티로 해보고 저 내
    용들을 출력해봐
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
    intent,PendingIntent.FLAG_UPDATE_CURRENT); //찾아보고
    Notification.Builder builder = new Notification.Builder(this);

    builder.setLargelcon(BitmapFactory.decodeResource(getResources(),android.R.drawable.ic_me
    nu_camera));
    builder.setSmallIcon(android.R.drawable.ic_menu_camera);
    builder.setTicker("사진이 등록되었습니다.");
    builder.setTitle("사진이 등록되었습니다.");
    builder.setText("사진을 확인하십시오!");
    builder.setWhen(System.currentTimeMillis());
    builder.setDefaults(Notification.DEFAULT_SOUND | Notification.DEFAULT_VIBRATE);
    builder.setContentIntent(pendingIntent);
    builder.setAutoCancel(true);

```

```

        builder.setNumber(999);

        notificationManager.notify(0, builder.build());
    }    //푸쉬알림설정
}

activity_run.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background" >

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="119dp"
        android:text="서비스가 실행중입니다!"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="@color/colorBlack"/>

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="match_parent"
        android:layout_height="211dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="164dp" />

    <Button
        android:id="@+id/contin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:background="@drawable/button_three"
android:onClick="onClick"
android:text="계속"
android:visibility="gone" />

```

<Button

```

android:id="@+id/finish"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
android:layout_alignParentTop="true"
android:background="@drawable/button_three"
android:onClick="onClick"
android:text="종료"
android:visibility="gone" />

```

<Button

```

android:id="@+id/close"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:layout_marginBottom="71dp"
android:layout_weight="1"
android:background="@drawable/button_main"
android:onClick="onClick"
android:text="종료"
android:textColor="@color/colorWhite"
android:textStyle="bold"
android:visibility="visible" />

```

<Button

```

android:id="@+id/chk"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"

```

```
android:layout_centerHorizontal="true"
android:layout_marginBottom="71dp"
android:layout_weight="1"
android:background="@drawable/button_main"
android:onClick="onClick"
android:text="사진확인"
android:textColor="@color/colorWhite"
android:textStyle="bold"
android:visibility="gone" />
```

</RelativeLayout>

```
IntroActivity.java
package com.example.ruldy.test1;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class IntroActivity extends Activity {

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

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                Intent intent = new Intent(IntroActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, 1500);
    }
}
```

```
activity_intro.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/security"
    tools:context=".IntroActivity">

</android.support.constraint.ConstraintLayout>
```

```
AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ruldy.test1">

    <uses-permission android:name="android.permission.VIBRATE"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/icon"
        android:label="ANSIM"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.AppCompat.NoActionBar">
        <activity android:name=".IntroActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity"/>
        <activity android:name = ".join"/>
        <activity android:name = ".SubActivity"/>
```

```

<activity android:name = ".RunActivity"
    android:launchMode="singleInstance"
    android:excludeFromRecents="true"
    android:taskAffinity="" />
</application>

<uses-permission android:name="android.permission.INTERNET" />
</manifest>

```

colors.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#c0c1d9</color> //background color
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="colorWhite">#ffffff</color>
    <color name="colorGray">#909090</color>
    <color name="coloransim">#c954ab</color>
    <color name="colorBlack">#000000</color>
    <color name="colorback">#23e7cfcf</color>
    <color name="colorthree">#f7d1ed</color>

</resources>

```

styles.xml

```

<resources>

<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>

```


</style>

</resources>

9.3 Sensor/Camera(Raspberry pi) source code

```
#
-
# pyclient.py
# 안드로이드에서 정보를 받아서 초음파 실행하고 사람 들어오면 사진 찍어서 서버로 보내줌

import sys, socket
import RPi.GPIO as gpio
import time
import picamera
import subprocess

serial = "4141Wn"
def echo_client(server_addr):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect(server_addr)
    sock.send(serial.encode('utf-8'))

    while True:
        gpio.setmode(gpio.BCM)
        data = sock.recv(1024).decode('utf-8')

        if data == (serial.replace("Wn", "-Wn")):
            trig = 13
            echo = 19
            gpio.setup(trig, gpio.OUT)
            gpio.setup(echo, gpio.IN)

            try :
                while True:

                    gpio.output(trig, False)
                    time.sleep(0.5)
                    gpio.output(trig, True)

                    time.sleep(0.00001)
                    gpio.output(trig, False)

                    while gpio.input(echo) == 0:
                        pulse_start=time.time()
                    while gpio.input(echo) == 1:
                        pulse_end = time.time()
```

```

pulse_duration = pulse_end - pulse_start
distance = pulse_duration * 17000
distance = round(distance,2)
if distance < 100:
    with picamera.PiCamera() as camera:
        camera.start_preview()
        time.sleep(1)
        camera.capture('/var/www/html/4141.jpg')
        camera.stop_preview()
    subprocess.call('scp -i /home/pi/team2/AWS_enoma_pswd.pem
/var/www/html/4141.jpg ubuntu@52.79.181.10:/var/www/html',shell = True)
    print(data)
    sock.send(data.encode('utf-8'))
    data = sock.recv(1024).decode('utf-8')
    if data.decode('utf-8') == serial.replace('\n', '-\n'):
        pass
    elif data.decode('utf-8') == serial.replace('\n', '_\n'):
        break
    print("Distance : " , distance, "cm")

except:
    gpio.cleanup()

sock.close()

if __name__ == '__main__':
    echo_client(('52.79.181.10', 5001))

```

9.4 DB(MySQL)연결 source code

-

```

#id.py
# serial번호를 받아서 DB에서 해당 serial의 id를 보내주는 함수

import pymysql

def Id(serial):
    serial = serial.replace("\n","")
    conn =
pymysql.connect(host='localhost',user='root',password='1234',db='login',charset='utf8
')

    curs = conn.cursor(pymysql.cursors.DictCursor)

    query = ("select id from user where serial =%s" %serial)

```

```

curs.execute(query)

results = curs.fetchall()

result = results[0]['id']
return result

```

```

#signup.py
#id password serial번호를 받아서 DB와 비교해서 회원가입 시켜주는 함수
import pymysql

def Signup(str):
    conn =
    pymysql.connect(host='localhost',user='root',password='1234',db='login',charset='utf8
    ',)
    str = str.replace("\n","")
    curs = conn.cursor(pymysql.cursors.DictCursor)
    list = str.split('/')

    usr_id = list[0]
    usr_pwd = list[1]
    input_serial = list[2]
    sql = """SELECT serial from user"""
    curs.execute(sql)
    results = curs.fetchall()
    try:
        bool = False
        for sNum in range(len(results)): #시리얼이 등록된건지 for 문으로 돌아가면서
확인
            if results[sNum]['serial'] == int(input_serial):
                sql = ("SELECT id FROM user")
                curs.execute(sql)
                idresults = curs.fetchall()
                for idNum in range(len(idresults)): # 사용중인 아이디를 for 문으로
확인하는부분
                    if idresults[idNum]['id'] != None:
                        if idresults[idNum]['id'] == usr_id:
                            return "idError"
                sql = ("SELECT id FROM user WHERE serial = %s"%input_serial)
                curs.execute(sql)
                endResults = curs.fetchall() # 이제 내가 내가 쓸 카메라의 시리얼이
이미 회원가입이 된건지 확인
                if endResults[0]['id'] == None: # 등록하고싶은 시리얼에 가입이안되서
id가 none 인경우
                    bool = True

```

```

        break
    else:
        return "idError2"
        break
    if bool == False:
        raise Exception("no serial")
except Exception as e:
    return "noSerial"
else:
    sql = ("UPDATE user SET id = '%s' , pwd = '%s' WHERE serial
= %s"%(usr_id,usr_pwd ,input_serial))
    curs.execute(sql.encode('utf8'))
    conn.commit()
    return "good"
finally:
    conn.close()

```

```

#serial.py
#id를 받아서 DB에서 id에 해당하는 serial번호를 보내주는 함수
import pymysql

def Serial(usr_id):
    usr_id = usr_id.replace("\n","")
    conn =
pymysql.connect(host='localhost',user='root',password='1234',db='login',charset='utf8
',)

    curs = conn.cursor(pymysql.cursors.DictCursor)

    query = ("select serial from user where id='%s'" %usr_id)
    curs.execute(query)

    results = curs.fetchall()
    result = results[0]['serial']
    return str(result)

```

```

#login.py
#id 와 password를 받아서 DB에서 비교해서 login해주는 함수
import pymysql

def Login(str1):
    con =
pymysql.connect(host='localhost',user='root',password='1234',db='login',charset='utf8

```

문서명: 인터넷응용 최종보고서

```
'')
cur = con.cursor()
str1 = str1.replace("Wn","")
usr2 = str1.split('?')

usr_id = usr2[0]
usr_pwd = usr2[1]
query = ("select pwd from user where id='%s'" %usr_id)
cur.execute(query)

results = cur.fetchall()
result = results[0][0]
if usr_pwd == result:
    return "success"
else :
    return "fail"
```