



NO-TOUCH S.E (Smart Entrance)

과목명	인터넷응용
담당	홍진표 교수님
제출일	2019년 06월 일
학과	정보통신공학과
팀명	Knock Knock (4팀)
팀장	박주영 (201601364)
팀원	김수빈 (201700716)
	김세진 (201500530)
	김혜원 (201701138)



한국외국어대학교

머리말

본 문서는 라즈베리파이와 안드로이드의 소켓통신을 이용한 노터치 방식의 공동 현관문 시스템에 관해 기술한다. 거주자가 애플리케이션에 정보를 등록하고 승인되면 공동 현관문을 출입 할 때 키패드 터치 없이 현관문이 자동으로 열린다.



목 차

1. 제품 개요 -----	4
1.1 제품 소개 -----	4
1.2 제품 개발 배경 및 필요성 -----	5
1.3 제품 특징 및 장점 -----	6
1.4 용어 및 약어 -----	6
2. 제품 내용 -----	8
2.1 제품 구성도 -----	8
2.2 제품 흐름도 -----	8
2.3 제품 설계 -----	9
2.3.1 서버/클라이언트 연결 -----	9
2.3.2 애플리케이션 정보 등록 -----	9
2.3.3 서비스 실행 -----	9
2.3.4 안드로이드 코드 설명 -----	10
2.3.5 아두이노 회로 설계 -----	16
2.4 제품 수행 -----	17
2.4.1 순차 다이어그램 -----	17
2.4.2 애플리케이션 등록 순서 -----	17
2.5 주요 적용 기술 -----	19
2.6 제품 개발 환경 -----	19
3. 기대효과 및 사업성 -----	20
3.1 기대효과 -----	20
3.2 사업성 -----	20
4. 개발 수행 내용 -----	21
4.1 업무 분장 -----	21
4.2 프로젝트 수행일정 -----	21
5. 참고 문헌 및 사이트 -----	22
5.1 참고 문헌 -----	22
5.2 참고 사이트 -----	22
6. 소스코드 -----	23

1. 제품 개요

본 장에서는 라즈베리파이(Raspberry pi), 아두이노 그리고 초음파 센서(Ultrasonic sensor)와 DC 모터(DC motor)를 이용하여 제작한 NO-TOUCH S.E(Smart Entrance)에 대하여 소개한다. NO-TOUCH S.E는 손을 자유롭게 쓸 수 없는 상황이나 치매로 인해 공동 현관문의 비밀번호를 잊은 경우, 또는 공동 현관문 키패드에 손이 닿지 않는 사람을 타겟으로 한다. 이 제품은 손을 사용하지 않아도 편하게 공동 현관문을 출입 할 수 있는 서비스를 제공한다. 거주자(사용자)는 애플리케이션에 설치해 정보를 등록하고 정보가 승인되면 자동으로 이용가능하다. 거주자와 라즈베리파이가 일정 거리 이하로 떨어진 경우, 거주자의 휴대폰 속 애플리케이션이 라즈베리파이로 정보를 보내고 라즈베리파이는 이 정보를 바로 웹으로 보낸다. 웹에서는 입력된 정보를 확인하고 거주자 데이터베이스에 정보가 있는 경우 문을 열어달라는 신호를 라즈베리파이로 보낸다. 신호를 받은 라즈베리파이는 공동 현관문을 연다. 공동 현관문 개폐 장치는 DC 모터로 대체한다. 보고서를 통해 본 개발의 필요성, 정보 등을 제공한다.

1.1 제품 소개

NO-TOUCH S.E(Smart Entrance)는 제품의 특징인 'No-Touch'와 Smart Entrance가 합쳐진 것이 제품명이다. 이 제품은 스마트폰 애플리케이션을 이용한 아파트 공동 현관문의 자동 개폐 시스템을 제공한다.

공동 현관문에 이 제품을 설치하고 애플리케이션을 다운받아 정보를 등록하면 패스워드를 입력하지 않아도 공동 현관문을 열 수 있다. NO-TOUCH S.E는 양손을 자유롭게 사용하기 힘든 상황이나 패스워드를 잊어버리기 쉬운 치매 환자들, 키패드에 손이 닿지 않는 어린 아이들에게 편리한 출입 서비스를 제공하려는 목적으로 개발되었다.

1.2 제품 개발 배경 및 필요성

우리가 일반적으로 사용하는 공동 현관문은 문 앞의 키패드에 비밀번호를 입력하고 일치하면 문이 열리는 방식이다. 이 방식은 뒤에서 유심히 쳐다보기만 해도 비밀번호를 유추할 수 있을 정도로 보안이 취약하다. 이와 관련된 내용은 다음 뉴스를 통해 알 수 있다.



MBC
떠도는 아파트 공동 현관 비밀번호, 주민 '불안'

기사입력 2017.01.06 오후 8:28
최종수정 2017.01.06 오후 8:38

187 242

가 가

270P 480P

[뉴스데스크] <앵커>
보통 아파트 1층엔 공동현관문이 있죠?
비밀번호를 눌러야만 들어갈 수 있는데요.

이 아파트 비밀번호 자료가 광고업체들 사이에서 떠돌고 있어 경찰이 수사에 나섰습니다.

보도에 조성식 기자입니다.

<리포트>

강원도 원주의 한 아파트, 세대 문마다 광고 전단지가 덕지덕지 붙어 있습니다.

아파트 안으로 들어오려면 1층 공동현관문의 비밀번호를 알아야만 하는 상황. 주민들은 어떻게 들어와 전단지를 붙이는지 의아해합니다.

[주민]

"모르겠다. 항상 집 앞에 외출하고 들어오면 전단지가 붙여 있으니깐 보통 서너 개씩..."

왜 그런가 했더니 원주의 열 곳 넘는 아파트 단지의 공동 현관 비밀번호가 유출된 겁니다.

아파트 이름과 동별 비밀번호는 물론 입력하는 방법까지 자세히 정리한 자료가 광고업체들을 중심으로 떠돌고 있습니다.

유출된 비밀번호를 가지고 직접 아파트 현관을 열어보겠습니다.

일부 아파트는 개별 현관문 비밀번호까지 노출돼 절도 등 범죄 우려를 낳고 있습니다.

[관리사무소 직원]

"우리가 누르면 뒤에서 보고 '아 저거구나' 하고 누르고 들어간다. 비밀번호를 빨리빨리 바꾼다고 해도 안 되더라."

경찰은 비밀번호 자료의 정확한 출처와 유보자 조사에 착수했습니다.

자료를 공유한 광고업체들은 물론 관리사무소 등을 상대로 수사를 벌일 계획입니다.

MBC뉴스 조성식입니다.

위의 뉴스 내용을 보면 아파트 이름과 동별 비밀번호는 물론 입력하는 방법까지 자세히 정리한 자료가 광고업체를 중심으로 떠돌고 있다는 것을 확인 할 수 있다. 이렇게 공동 현관문의 보안이 취약하다 보니 거주자가 아니어도 쉽게 들어갈 수 있다.

뿐만 아니라 2009년 집 앞 현관문에 정체를 알 수 없는 표시가 범죄자들이 해 놓은 표시라는 괴담이 떠돌았다. 공동 현관문의 보안이 확실하다면 이러한 괴담이 돌지 않았을 것이다. 만약 그 표시가 정말 범죄자들이 해 놓은 표시라면 공동 현관문의 낮은 보안으로 인해 범죄에 노출 될 수도 있다는 의미이다.

우리는 이러한 범죄의 노출 가능성으로부터 멀어지기 위해 확실한 보안을 가진 현관문이 필요하다고 생각했다. 키패드를 직접 입력하지 않아 공동 현관문의 비밀번호의 유출이 어려워지면서도 누구나 편하게 사용할 수 있는 방식인 NO-TOUCH S.E를 통해 거주자들에게 안전함과 편리함을 제공하고자 하였다.

1.3 제품 특징 및 장점

NO-TOUCH S.E의 대표적인 특징이자 장점은 키패드를 터치할 필요가 없다는 것이다. 그렇기 때문에 양손에 짐이 있거나 키패드에 손이 닿지 않아도 문을 열 수 있다. 비밀번호를 잊어 먹기 쉬운 치매 환자들도 애플리케이션만 있으면 된다. 애플리케이션에 정보를 입력하고 승인되면 자동으로 문이 개폐되며, 거주자라면 누구나 이용가능하다. 거주자만 이용가능하기 때문에 우수한 보안성을 지니고 있다.

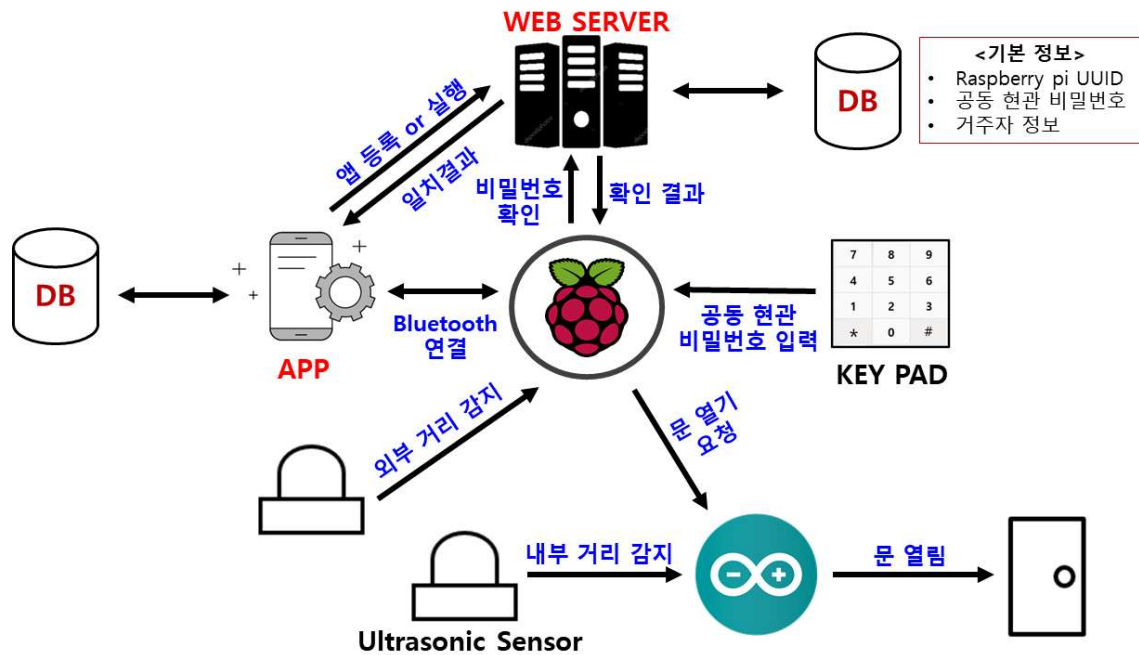
1.4 용어 및 약어

용어 및 약어	설명
Raspberry Pi 3 (라즈베리)	라즈베리파이는 영국 잉글랜드의 라즈베리파이 재단이 학교와 개발도상국에서 기초 컴퓨터 과학의 교육을 증진시키기 위해 개발한 신용카드 크기의 싱글 보드 컴퓨터이다. 그래픽 성능이 뛰어나면서도 가격이 저렴한 것이 특징이다. 라즈베리파이는 모두 동일한 비디오코어 IV GPU와 싱글 코어 ARMv6에 호환되는 CPU 또는 신형의 ARMv7에 호환되는 쿼드 코어, 1GB의 RAM 또는 256MB의 메모리를 포함한다. Micro SD 카드를 이용하여 Raspberry Pi 공식 홈페이지에서 운영체제를 다운받아 사용할 수 있다.
Arduino UNO (아두이노)	아두이노는 오픈 소스를 기반으로 한 단일 보드 마이크로컨트롤러로 완성된 보드와 관련 개발 도구 및 환경을 말한다. 2005년 이탈리아의 IDII (Interaction Design Institutelvera)에서 하드웨어에 익숙지 않은 학생들이 자신들의 디자인 작품을 손쉽게 제어할 수 있게 하려고 고안되었다. 아두이노는 다수의 스위치나 센서로부터 값을 받아들여, LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어 낼 수 있다. 임베디드 시스템 중의 하나로 쉽게 개발할 수 있는 환경을 이용하여, 장치를 제어할 수 있다.
Ultrasonic Sensor (초음파 센서)	초음파 센서는 가까운 거리에 있는 물체 혹은 사람의 유·무, 거리 측정, 속도 측정 등에 사용된다. 초음파 센서는 2개의 압전소자로 되어 있으며, 초음파가 입사되면 압전 소자는 진동해서 전압이 발생되고 역으로 압전 소자에 전압을 인가하면 초음파가 발생된다. 초음파 센서는 물체에 반사된 음파를 그대로 센서로 입력해 발생한 전압을 회로에서 처리하여 거리를 계산한다.
DC Motor (직류 전동기)	DC모터는 기계적인 동력으로 직류 전력을 전기 기계로 변환하는 장치이다. 가장 일반적인 유형은 자기장이 생성한 힘에 의존한다. 대부분의 DC 모터는 주기적인 모터 부분의 전류 흐름 방향을 변경하기 위해 전기 또는 전자기구의 내부 중 일부를 가지고 있다. DC 모터는 기존의 직류 점등 전원 분배 시스템으로부터 전원을 공급 받을 수 있기 때문에, 광범위하게 사용되는 제 1유형이다. 속도는 가변 공급 전압 또는 계자 권선의 전류 가도

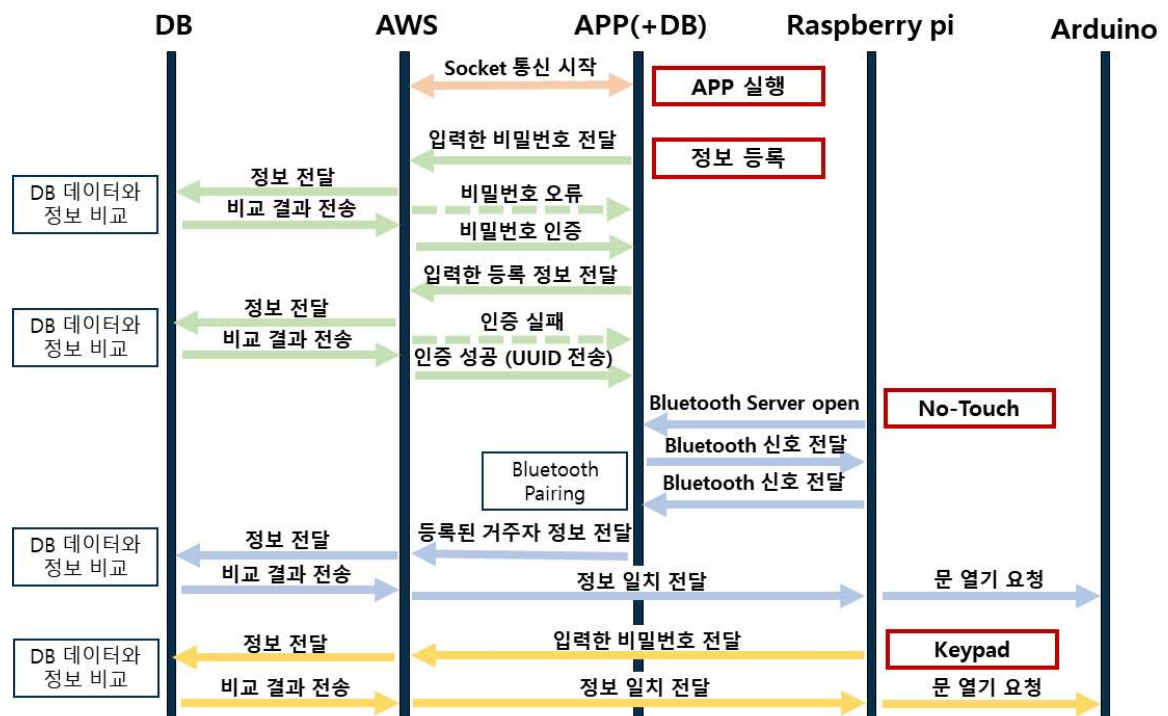
	변화를 사용하여, 넓은 범위에서 제어될 수 있다. 소형 DC 모터는 도구, 장난감 및 가전제품에서 사용된다.
AWS	AWS는 아마존 웹 서비스(Amazon Web Services)로, 아마존닷컴이 제공하는 각종 원격 컴퓨팅 서비스(웹 서비스)이다. AWS는 다른 웹 사이트나 클라이언트 측 응용 프로그램에 대해 온라인 서비스를 제공하고 있다. 이러한 서비스의 상당수는 최종 사용자에게 직접 공개되는 것이 아니고, 다른 개발자가 사용 가능한 기능을 제공하는 플랫폼을 제공하는 PaaS이다. AWS의 각종 서비스는 REST 프로토콜 및 SOAP 프로토콜을 통해 접근, 이용 및 관리가 가능하다. 비용은 실제 사용량에 따라 결정되며, 일부 서비스의 경우 미리 고정된 금액을 지불하는 형태도 있다.
SQLite 3	SQLite는 MySQL이나 PostgreSQL과 같은 데이터베이스 관리 시스템이지만, 서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스이다. 일반적인 RDBMS에 비해 대규모 작업에는 적합하지 않지만, 중소규모라면 속도에 손색이 없다. 또 API는 단순히 라이브러리를 호출하는 것만 있으며, 데이터를 저장하는 데 하나의 파일만을 사용하는 것이 특징이다. 구글 안드로이드 운영 체제에 기본 탑재된 데이터베이스이기도 하다.

2. 제품 내용

2.1 제품 구성도



2.2 제품 흐름도



2.3 제품 설계

2.3.1 서버/클라이언트 연결

```
def echoserver(port):
    serverSock = socket(AF_INET, SOCK_STREAM)
    serverSock.bind(('', port))
    serverSock.listen(5)
    print('%d번 포트로 접속 대기중...'%port)
    while True:
        connectionSock, addr = serverSock.accept()
        recvData = connectionSock.recv(1024).decode('utf-8')
        if recvData == "P2\n" or recvData == "P3\n" or recvData == "DOOR\n": #앱에서 페이지 정보 받으면 pass
            #print(recvData)
            pass
        else: #라즈베리파이 정보 받으면 라즈베리파이 connection정보 저장
            #print(recvData)
            raspberry[recvData] = connectionSock
        print('Connected by {}'.format(connectionSock))
        handler = threading.Thread(target=run, args=(connectionSock, addr))
        handler.daemon = True
        handler.start()
```

서버에서 thread를 통해 앱과 라즈베리파이 각각 소켓 통신을 연결해준다

2.3.2 애플리케이션 정보 등록

```
if '?' in recvData: #앱에서 동 비밀번호 입력시 서버 db와 비교해줄
    a = recvData.split("?")[1]
    print("a=" + a)
    d_pass = Apart.objects.filter(door_password=a) #서버db에서 동 비밀번호 인증
    if d_pass:
        print("db_ok")
        sendData = 'confirm' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
    else:
        print("db_np")
        sendData = 'no' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
```

앱에서 동 비밀번호를 입력하면 서버에 저장된 DB와 입력된 비밀번호가 일치하는지 비교한다.

```
if '/' in recvData: #앱에서 거주자 정보를 입력하여 서버 db와 비교해 인증한다/앱에서 자동 인증 절차
    b = recvData.split('/')
    ho = int(b[0])
    db = Apart.objects.filter(ho=ho, name=b[1], phone=b[2], identification=b[3][:-2])
```

앱에서 입력한 정보를 받아와 DB와 비교한다. (앱 실행 시 인증 절차의 시작과 동일)

```
elif '#' in b[3][:-1]:
    if db:
        print("db ok2")
        p = Apart.objects.get(UUID__contains="@", name=b[1], identification=b[3][:-2])
        p = p.UUID
        sendData = 'confirm' + p + '\n'
        print(sendData)
        connectionSock.sendall(sendData.encode("utf-8"))
    else:
        print("db no2")
        sendData = 'noooooo@1' + '\n'
        sendData = 'noooooo@1' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
# connectionSock.sendall(sendData.encode("utf-8"))
        sendData = 'noooooo@1' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
# connectionSock.sendall(sendData.encode("utf-8"))
```

DB와 비교해서 정보가 일치하면 서버는 라즈베리파이 고유 ID인 UUID를 앱에 보낸다.

2.3.3 서비스 실행

```
if '/' in recvData: #앱에서 거주자 정보를 입력하여 서버 db와 비교해 인증한다/앱에서 자동 인증 절차
    b = recvData.split('/')
    ho1 = int(b[0])
    db = Apart.objects.filter(ho=ho1, name=b[1], phone=b[2], identification=b[3][:-2])
```

앱을 실행하면 입력한 정보를 받아와 DB와 비교한다.

```
if '&' in b[3][:-1]:
    #print(raspberry)
    con = raspberry['33333$'\n'] #라즈베리파이에게 인증 확인을 보내줌
    #print(type(con))
    #print(con)
    if db:
        print("db ok1")
        sendData = "yes1"
        con.sendall(sendData.encode("utf-8"))
        print("send ok")
        sendData2 = "confirm\n"
        connectionSock.sendall(sendData2.encode("utf-8"))
    else:
        print("db no1")
        sendData2 = "noooooo\n"
        connectionSock.sendall(sendData2.encode("utf-8"))
        # sendData = "no1"
        # con.sendall(sendData.encode("utf-8"))
```

DB와 비교해서 정보가 일치하면 서버는 라즈베리파이에게 인증 확인을 보내준다.

2.3.4 안드로이드 코드 설명

<APP등록 시>

안드로이드 스튜디오 코드는 총 5개의 창으로 이루어져 있고 각 창을 Main, Main0, Main1,

Main2, Main3, Main4 라고 한다.

Main창 이외의 모든 창에는 Onclick을 이용한 간단한 뒤로가기 버튼이 있다.

Main*Activity.java

```
public void buttonback(View v) { finish(); }
```

activity_main*.xml

```
<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="@android:color/white"
    android:onClick="buttonback"
    android:text="뒤로가기"
    android:textColor="@android:color/black"
    android:textStyle="normal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.04"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.023" />
```

Main창에서는 app거주자등록버튼구현, 라즈베리파이 자동 문 열림 구현을 위한 블루투스 및 소켓 연결이 있다.

MainActivity.java

```
//다음창으로 넘어가기 버튼
public void apart(View v) {
    Intent intent = new Intent( packageContext: this, Main0Activity.class);
    startActivity(intent);
}
```

위의 방법과 같이 Onclick을 이용하여 버튼을 누르면 apart함수가 실행되게 한다, intent 모듈을 이용하여 다음페이지인 Main0Activity.class로 넘어가게 해준다.

Main0창에서는 app거주자등록 1단계 : 아파트선택

Main1창에서는 app거주자등록 2단계: 동 선택

위의 방법과 같이 버튼설정과, 버튼 실행 시 다음페이지로 넘어간다.

Main2창에서는 app거주자등록 3단계: 동 비밀번호

이 동 비밀번호는 확인 버튼을 누르면 서버로 입력받은 동 비밀번호가 전송되고, 속 값과 일치 시 confirm메세지, 불일치시 no메세지를 앱으로 전송해준다. confirm메세지를 받으면 intent를 이용하여 다음 창으로 넘어가고 아니면 비밀번호가 맞지 않는다는 팝업창을 띄우며 페이지 이동이 없다.

Oncreate 밖에서 필요한 서버 변수를 선언해주고, Oncreate 안에서 서버와 통신을 시작한다.
통신을 위해 객체 Handler와 Thread를 선언하고 Thread를 실행시켜준다.

Main2Activity.java

```
intent2 = new Intent(getApplicationContext(),Main3Activity.class);//비밀번호를눌렀을때 페이지 넘기는 intent

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

//서버와 통신시작
try {
    clientSocket = new Socket(ip, port);
    socketIn = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
    socketOut = new PrintWriter(clientSocket.getOutputStream(), autoFlush: true);
} catch (Exception e) {
    e.printStackTrace();
}
socketOut.println("P2");//서버에게 페이지 알림

myHandler = new MyHandler();
myThread = new MyThread();
myThread.start();
```

Button모듈을 이용하여 비밀번호확인 버튼을 누르면 EditText창의 값을 소켓으로 보낸다.

```
Button btn = (Button) findViewById(R.id.password);//버튼 변수 설정
btn.setOnClickListener((v) -> {
    //패스워드확인버튼 눌렀을때 : textView의 값받아와서 비밀번호와 확인!
    EditText text_pw = (EditText) findViewById(R.id.pwtext);//안드로이드 위젯(글자입력)을 실행
    String pw = text_pw.getText().toString();//pw문자열변수에 위의 text_pw에서 텍스트를 가져와 문자열로 바꿔서 사용
    socketOut.println("?" + pw);
});
```

Oncreat밖에서 Thread클래스에서는 Handler를 이용하여 서버로부터 받은 메시지의 데이터를 다룬다.

```
class MyThread extends Thread {
    boolean stop = false;

    public void stopThread() { stop = true; }

    @Override
    public void run() {
        while (true) {
            try {
                // InputStream의 값을 읽어와서 data에 저장
                String data = socketIn.readLine();
                // Message 객체를 생성, 핸들러에 정보를 보낼 땐 이 메시지 객체를 이용
                Message msg = myHandler.obtainMessage();
                msg.obj = data;
                myHandler.sendMessage(msg);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

Handler클래스는 서버로부터 받은 데이터가 confirm이면 다음페이지로 아니면 팝업창이 띄워진다.

```
class MyHandler extends Handler {
    @Override
    public void handleMessage(Message msg) {
        if(msg.obj.toString().equals("confirm")) { //보낸 비번이 인증확인되면,
            startActivity(intent2); //다음페이지로 넘어감
        }
        else {
            new AlertDialog.Builder( context: MainActivity.this)
                .setTitle("오류 : 동 비밀번호가 틀립니다")
                .setMessage("WnWn 비밀번호를 다시 입력하십시오")
                .setNeutralButton( text: "닫기", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dlg, int sumthin) {
                    }
                })
                .show();
        }
    }
}
```

Main3창에서는 app거주자등록 4단계: 거주자 정보 입력(호수,이름,전화번호,주민번호) 거주자 정보는 확인 버튼을 누르면 서버로 입력받은 정보가 전송되고, 서버의 DB 속 값들과 모두 일치 시 confirm + UUID 메시지를 받으면 인증된 거주자정보와 UUID가 모두 앱 DB에 저장되고, 다음 창으로 넘어간다. 불일치 시 noooooo메세지를 받아 팝업창을 띄우며 페이지 이동이 없다.

서버와의 소켓통신은 위와 같은방법이며 Handler함수에서 서버로부터 확인 메시지를 받아 실행하는 코드이다.

Handler함수 속 handleMessage함수에서 서버에서 준 데이터를 servermsg라 선언하고 이는 인증결과 + UUID이므로 이를 쪼개어 servermsg_confirm과 servermsg_uuid로 나눠준다.

```
String servermsg = msg.obj.toString(); //서버에서 준 데이터는 인증결과+UUID값이다.
String servermsg_confirm=servermsg.substring(0,7); //서버에서 준 데이터중 인증결과만 뜻하는 변수
String servermsg_uuid = servermsg.substring(8);
```

servermsg_confirm이 confirm이고 입력받은 inf*값들이 null이 아니라면 dbHelper 속에 inf*와 uuid가 저장되고, 다음페이지로 넘어간다.

```
//보낸 거주자 정보가 웹디비랑 인증확인이면,
if(servermsg_confirm.equals("confirm") && inf1 != null && inf2 != null && inf3 != null && inf4 != null) {
    dbHelper.insert(inf1, inf2, inf3, inf4, servermsg_uuid);
    System.out.println(dbHelper.getResult());
    startActivity(intent3); //다음페이지로 넘어감
}
```

Main4창에서는 app거주자등록이 완료

<APP등록 후 라즈베리파이와 블루투스연결 + 서버와 소켓통신>

APP거주자등록이 완료되었다면 APP을 다시 실행시킨다.

1. APP을 다시 실행 시키면 Main창에서 라즈베리파이와 Bluetooth 연결이 실행된다.
(bluetooth가 인식되고 앱에서 bluetooth가 활성화되어있는 조건에서)
2. 라즈베리파이와 APP의 블루투스 소켓 통신으로 라즈베리파이가 보낸 UUID가 APP DB의 UUID와 같다면, 서버 소켓 통신을 연결하여 APP이 서버로 APP DB 속 거주자정보를 보내준다.
서버가 APP이 보낸 거주자정보와 서버 속 DB의 거주자 정보가 일치하면 라즈베리 파이에게 신호를 주어 문이 열린다.

*소켓통신을 하기 위해서는 통신 환경설정 + Thread + Handler로 이루어진다.
(블루투스 소켓 또한 같다. Handler는 데이터를 받아오는 함수를 뜻한다)

블루투스기능을 수행하기 위해 manifest에 bluetooth기능을 권한을 준다.
AndroidManifest.xml

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

블루투스기능은 블루투스소켓을 이용하는 것으로 위의 소켓통신방법과 유사하다.
onCreate 밖에 블루투스소켓통신을 위해 선언해주고

```
//선언
BluetoothAdapter mBluetoothAdapter;
Set<BluetoothDevice> mPairedDevices;
Handler mBluetoothHandler;
ConnectedBluetoothThread mThreadConnectedBluetooth;
BluetoothDevice mBluetoothDevice;
BluetoothSocket mBluetoothSocket;
final static int BT_MESSAGE_READ = 2;
final static UUID BT_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
```

onCreate 안에 실행코드 함수를 수행해준다.
bluetoothrasberry()는 블루투스연결을 위한 실행코드이고
mThreadConnectedBluetooth.write()은 연결된 블루투스에 메시지를 보내는 코드이다.

```
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

bluetoothrasberry();//실행코드
mThreadConnectedBluetooth.write( str: "hello I'm App");
```

실행코드는 앱이 실행되고 있는동안 'rasberry'라는 라즈베리파이 블루투스 이름을 가진
디바이스 장치가 있다면 항상 블루투스를 연결시키는 코드이다.
블루투스 연결이 가능한 디바이스 mPairedDevices 중에서 raspberry이면 블루투스 연결 할
디바이스 mBluetoothDevice에 라즈베리파이 장치 주소가 입력되어 블루투스 소켓 통신을 한다.

```
//실행코드
void bluetoothrasberry() {
    if (mBluetoothAdapter.isEnabled()) {
        mPairedDevices = mBluetoothAdapter.getBondedDevices();
        //라즈베리파이가 있으면 항상 연결
        System.out.println("check!");
        connectSelectedDevice( selectedDeviceName: "rasberry");
    }
}
```

실질적으로 블루투스가 연결되는 코드이다.
라즈베리파이가 APP의 페어링 목록 mPairedDevices에 있다면 블루투스 소켓 통신을 시작한다.
소켓통신과 같이

1. 소켓열기
2. Handler객체 선언
3. Thread객체 선언
4. Thread 실행 하여 블루투스를 연결한다.

```
//블루투스 연결하는 메소드
void connectSelectedDevice(String selectedDeviceName) {
    for (BluetoothDevice tempDevice : mPairedDevices) {
        if (selectedDeviceName.equals(tempDevice.getName())) {
            mBluetoothDevice = tempDevice;
            System.out.println(tempDevice.getName());
            break;
        }
    }
}

//실제로 블루투스 장치와 연결하는 부분이다.
// 우리가 연결에 필요한 값은 장치의 주소이다.
// 따라서 for 문으로 피어싱된 모든 장치를 검색을 하면서 매개 변수 값과 비교하여 같다면 그 장치의 주소 값을 얻어오는 부분이다.
try {
    mBluetoothSocket = mBluetoothDevice.createRfcommSocketToServiceRecord(BT_UUID);
    mBluetoothSocket.connect();
    mBluetoothHandler = new ConnectedBluetoothHandler();
    mThreadConnectedBluetooth = new ConnectedBluetoothThread(mBluetoothSocket);
    mThreadConnectedBluetooth.start();
    Toast.makeText(getApplicationContext(), text: "라즈베리파이와 연결되었습니다.", Toast.LENGTH_LONG).show();
} catch (IOException e) {
    Toast.makeText(getApplicationContext(), text: "블루투스 연결 중 오류가 발생했습니다.", Toast.LENGTH_LONG).show();
}

//mBluetoothDevice를 통해 createRfcommSocketToServiceRecord(UUID)를 호출하여 mBluetoothSocket을 가져온다. (참고로 여기서 사용된 UUID 값은 시리얼 통신용이다.)
//그러면 mBluetoothDevice에 연결 될 mBluetoothSocket이 초기화된다. 그 후 connect()를 호출하여 연결을 시작한다.
```

블루투스 소켓 Thread 코드

```
public ConnectedBluetoothThread(BluetoothSocket socket) {
    mSocket = socket;
    InputStream tmpIn = null;
    OutputStream tmpOut = null;
    try {
        tmpIn = socket.getInputStream();
        tmpOut = socket.getOutputStream();
    } catch (IOException e) {
        Toast.makeText(getApplicationContext(), text: "소켓 연결 중 오류가 발생했습니다.", Toast.LENGTH_LONG).show();
    }
    mInStream = tmpIn;
    mOutStream = tmpOut;
}
```

블루투스 소켓 Handler코드

이 Handler 클래스는 handleMessage로 라즈베리파이에서 UUID값을 받아오는데 이용된다. 따라서 APP DB인 dbHelper를 통해 라즈베리파이에서 받아온 receiveduuid와 dbHelper 속 getUuid를 비교하여 일치한다면 runserver()를 실행시켜준다.

```
//ConnectedBluetoothHandler 핸들러
private class ConnectedBluetoothHandler extends Handler{
    public void handleMessage( android.os.Message msg) {
        if (msg.what == BT_MESSAGE_READ) {
            String readMessage = null;
            try {
                readMessage = new String((byte[]) msg.obj, charsetName: "UTF-8");
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
            System.out.println("블루투스핸들러");
            String receiveuuid = readMessage.substring(0,36);
            if (dbHelper.getUuid().equals(receiveuuid)){
                runserver();
            }
        }
    }
}
```

앞서 말한 서버소켓 통신과 같이 서버를 위해 환경설정을 해주고 서버통신을 시작하는 함수 runserver()는 소켓을 열어주고 서버소켓Handler, Thread를 선언해주고 실행한다. 소켓통신이 된 후 dbHelper 속의 거주자정보를 서버로 보내준다.

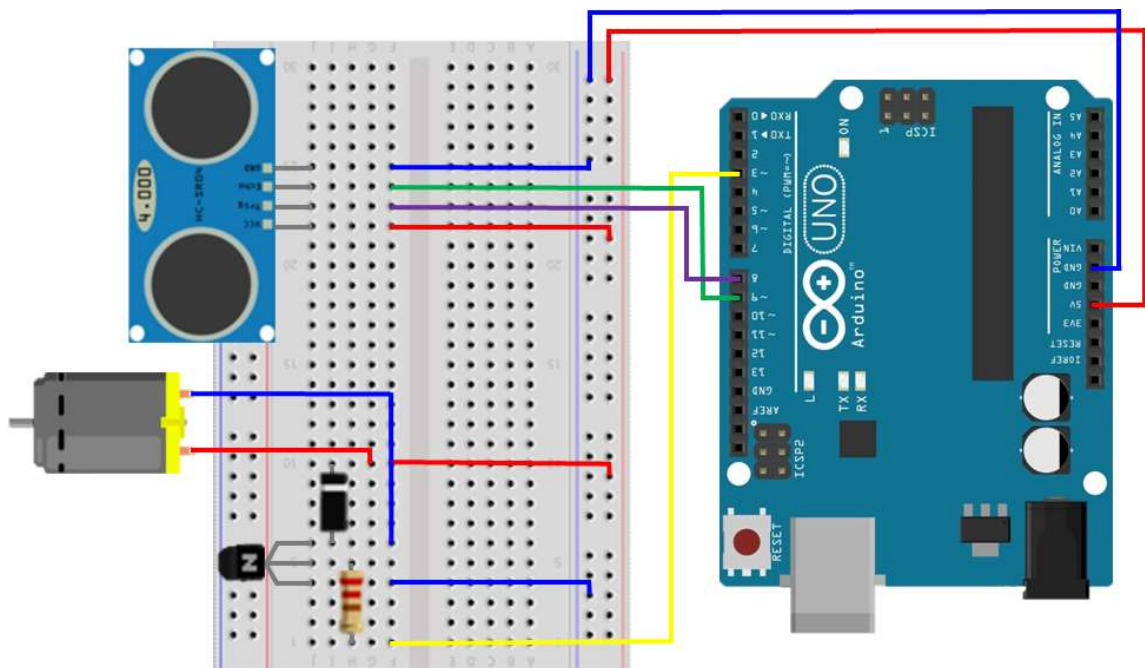
```
//라즈베리파이에서 받아온 데이터값이 DB값이면 실행되는 서버소켓통신코드
void runserver(){
    //readMessage가 DB의 uuid값과 일치하면
    //서버와 통신시작
    try {
        clientSocket = new Socket(ip, port);
        socketIn = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), autoFlush: true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    socketOut.println("DOOR");//서버에게 페이지 알림

    myHandler = new MyHandler();
    myThread = new MyThread();
    myThread.start();
    System.out.println(dbHelper.openDoor());
    socketOut.println(dbHelper.openDoor());
    // socketOut.println("1"+"/"+KlmsubIn+"/"+010-1111-1111+"/"+111111-111111+"&");
}
```

서버통신 Handler클래스에서 서버의 확인 메시지를 받는다.

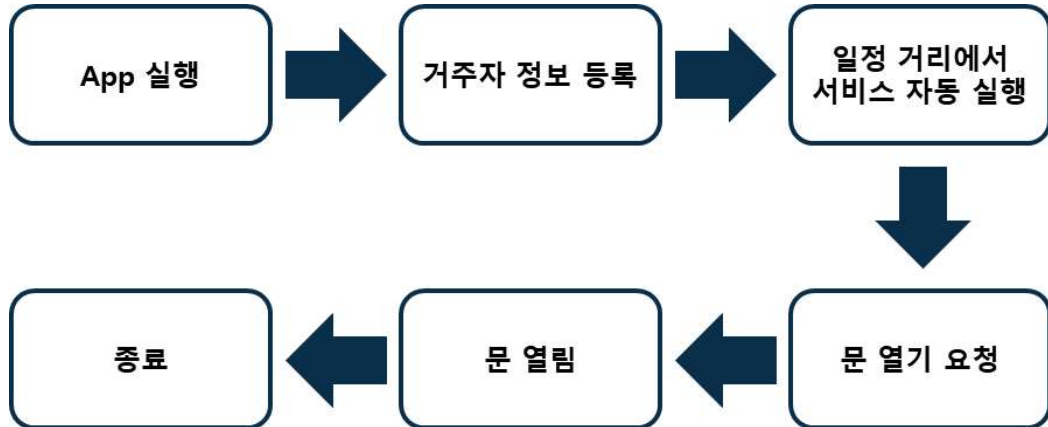
```
//서버소켓통신 Handler
class MyHandler extends Handler {
    @Override
    public void handleMessage(Message servermsg) {
        String a = servermsg.obj.toString();
        String reala = a.substring(0,7);
        if(reala.equals("confirm")){//보낸 비번이 인증확인이면,
            System.out.println(reala);
        }
    }
}
```

2.3.5 아두이노 회로 설계



2.4 제품 수행

2.4.1 순차 다이어그램



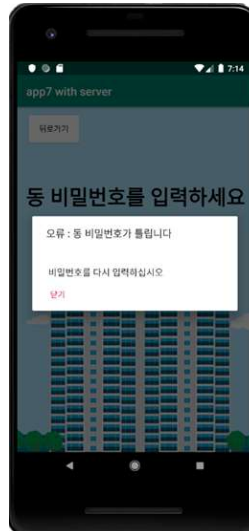
2.4.2 애플리케이션 등록 순서



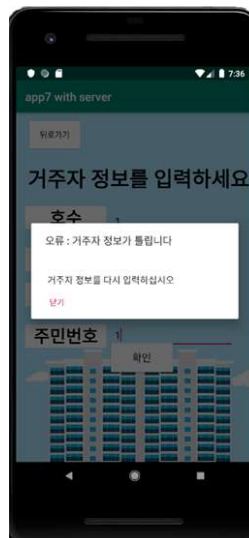
NO-TOUCH S.E 애플리케이션 등록을 위해 애플리케이션을 실행하면 다음과 같은 화면이 뜬다. 거주자 APP 등록을 누르면 애플리케이션에 정보를 등록할 수 있다.



애플리케이션을 실행하면 아파트를 고르는 화면이 나온다. 아파트를 선택하고 나면 동을 선택하는 화면이 나온다.



동을 선택하고 나면 동 비밀번호를 입력하는 화면이 나온다. 입력한 동 비밀번호가 데이터베이스에 저장된 것과 다를 경우, 그림과 같이 오류 문구가 뜬다.



올바른 동 비밀번호를 입력하면 거주자 정보를 입력하는 창이 나온다. 입력한 거주자 정보가 데이터베이스에 등록된 정보와 다를 경우, 그림과 같이 오류 문구가 뜬다.



올바른 거주자 정보를 입력하면 앱 등록이 완료된다.
앱 등록이 완료되면 NO-TOUCH S.E를 사용 할 수 있다.

2.5 주요 적용 기술

S/W	<ul style="list-style-type: none"> · Java를 이용하여 SQLite 3 DB 구축 · BLE 신호를 이용한 애플리케이션과 라즈베리파이 Bluetooth 통신 구축 · 애플리케이션과 서버의 Socket 통신을 통한 정보 등록 시스템 구현 · Android studio를 활용한 애플리케이션 DB 구축
H/W	<ul style="list-style-type: none"> · 아두이노에 Ultrasonic Sensor를 활용한 내부 개폐 시스템 구현 · 라즈베리파이에 키패드 입력을 활용한 외부 개폐 시스템 구현 · 애플리케이션과 라즈베리파이의 Bluetooth 통신 및 웹 서버와 라즈베리파이의 Socket 통신을 통한 노터치 방식의 외부 개폐 시스템 구현

2.6 제품 개발 환경

구분		상세 내용
S/W 개발환경	OS	Linux, Android
	개발환경 (IDE)	Android Studio, sqlite 3
	개발 언어	Python, Java
H/W 구성장비	디바이스	Raspberry Pi 3, Arduino UNO
	Sensor 및 Actuator	Ultrasonic sensor, DC motor
	통신	Serial, Socket, Bluetooth
	개발 언어	Python, C

3. 기대효과 및 사업성

3.1 기대효과

NO-TOUCH S.E 설치함으로써 키패드를 사용할 확률이 감소하므로 비밀번호 유출 가능성이 줄어 보안이 강화 될 것이다. 뿐만 아니라 애플리케이션 등록 및 사용이 거주자만 가능하다는 점에서 거주자들은 보다 안전한 주거 생활을 누릴 것으로 보인다. 손으로 키패드에 비밀번호를 입력하지 않아도 되므로 손을 쓸 수 없는 상황이나 상대적 약자, 치매 환자도 현관문 출입을 편하게 할 수 있다. 그리고 기존의 현관문 시스템 사용 시 발생하던 물리적 작동, 육체적 노동 단계가 제거 되므로 공동 현관문 출입 시 발생하는 시간이 감소되어 시간이 절약된다.

3.2 사업성

NO-TOUCH S.E는 스마트폰 애플리케이션만 있으면 되기 때문에 카드 키 제작이 불필요하므로 비용이 감소될 것으로 보인다. 그리고 기존의 현관문 시스템과 다르게 비밀번호 유출 가능성이 적고 편리하기 때문에 다양한 곳에서 활용 가능하다. 보안이 중요한 미술관, 은행 등의 직원 전용 출입구나 회사, 아파트 등에서 활용 가능하므로 수익이 창출 될 것으로 예상된다. 더 나아가 앞으로의 스마트 홈 시대에 도어락 시스템에서 큰 역할을 할 것으로 보인다.

4. 개발 수행 내용

4.1 업무분장

이름	학과	학번	역할	담당업무
박주영	정보통신공학과	201601364	팀장	Web Server
김수빈	정보통신공학과	201700716	팀원	Raspberrypi & Arduino
김세진	정보통신공학과	201500530	팀원	Application
김혜원	정보통신공학과	201701138	팀원	Application

4.2 프로젝트 수행일정

구분	추진내용	수행기간			
		1주	2주	3주	4주
계획	아이디어 선정 및 자료조사				
분석	Arduino Sensor 및 Actuator 분석				
	Database 구조 분석				
설계	전체 구성도 작성				
	App 수행 시나리오 작성				
	Arduino 회로 설계				
개발	Raspberry pi와 Arduino Serial 통신 구축				
	NO-TOUCH S.E 내부 개폐 시스템				
	Application 디자인				
	Application 구축				
	Server 구축				
	Socket 통신 구축				
	Application 정보 등록 시스템 구현				
	NO-TOUCH S.E 외부 개폐 시스템 구현(Keypad 방식)				
	NO-TOUCH S.E 외부 개폐 시스템 구현(No-touch 방식)				
	Bluetooth 통신 구축				
테스트	연동 테스트 실행				
종료	제품 발표				

5. 참고 문헌 및 사이트

5.1 참고 문서

출판사	문서 제목
HANBACK(에서 제공)	IoT-Smart-Server를 이용한 클라우드 구축 및 활용

5.2 참고 사이트

Title	URL
Ultrasonic Sensor 회로	http://blog.naver.com/PostView.nhn?blogId=kjycu&logNo=220639957952
DC Motor 회로	https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm
용어 약어 검색 (위키백과 이용)	https://ko.wikipedia.org/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%EB%8C%80%EB%AC%B8
Raspberry pi, Arduino Serial 통신	https://creamp.tistory.com/3
Raspberry pi Bluetooth 및 Beacon 설치	https://codeanalysis.tistory.com/2
파이썬 socket	https://soooprmx.com/archives/8737
bluetooth server	https://webnautes.tistory.com/1137
aws ec2 ubuntu설정	https://www.google.com/search?q=aws+ec2+ubuntu%EC%84%A4%EC%A0%95&rlz=1C1CHBD_koKR839KR839&oq=aws+ec2+ubuntu%EC%84%A4%EC%A0%95&aqs=chrome..69i57.26900j0j4&sourceid=chrome&ie=UTF-8
장고 설치	https://developer.mozilla.org/ko/docs/Learn/Server-side/Django/development_environment
장고(sqlite3) db	https://beomi.github.io/2017/03/01/HowToMakeWebCrawler-Save-with-Django/
장고 쿼리셋	https://wayhome25.github.io/django/2017/07/25/tsd7-django-query-database/

6. Source Code

arduino code

```
int motor_pin = 3;
int trig_pin = 8;
int echo_pin = 9;
int motor_speed = 255;
int motor_state = 1;
char pwd_ok = '.';

void setup() {
  Serial.begin(9600);
  pinMode(trig_pin, OUTPUT);
  pinMode(echo_pin, INPUT);
  pinMode(motor_pin, OUTPUT);
}

void loop() {
  digitalWrite(trig_pin, LOW);
  digitalWrite(echo_pin, LOW);
  delayMicroseconds(2);
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);

  if (Serial.available() > 0) { //키패드 사용하는 경우
    pwd_ok = Serial.read();
    analogWrite(motor_pin, motor_speed);
    delay(500);
    Serial.println("pwd_ok");
  }

  unsigned long duration = pulseIn(echo_pin, HIGH);

  float distance = duration / 29.0 / 2.0;

  //초음파 센서를 사용하는 경우
  if(distance < 10) {
    if (motor_state == 0){
      motor_state = 1;
      analogWrite(motor_pin, motor_speed);
      Serial.println(distance);
    }
  }
  else{
```



```

    if (motor_state == 1){
        motor_state = 0;
        analogWrite(motor_pin, LOW);
        Serial.println(distance);
    }
}

delay(500);
}

```

sever code(AWS)

```

import os
from socket import *
import django
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "mysite.settings")
django.setup()
from apartment.models import Apart
import threading
raspberry={}

def echoserver(port):
    serverSock = socket(AF_INET, SOCK_STREAM)
    serverSock.bind(('', port))
    serverSock.listen(5)
    print('%d번 포트로 접속 대기중...'%port)
    while True:
        connectionSock,addrs = serverSock.accept()
        recvData=connectionSock.recv(1024).decode('utf-8')
        if recvData == "P2\n" or recvData=="P3\n" or recvData=="DOOR\n": #앱에서 페이지 정보 받으면
pass
            #print(recvData)
            pass
        else: #라즈베리파이 정보 받으면 라즈베리파이 connection정보 저장
            #print(recvData)
            raspberry[recvData]=connectionSock
            print('Connected by{}'.format(connectionSock))
            handler = threading.Thread(target=run, args=(connectionSock, addrs))
            handler.daemon = True
            handler.start()

def run(connectionSock,addrs):
    while True:
        try:
            recvData = connectionSock.recv(1024).decode('utf-8')
            print(recvData)
            if '$' in recvData: #라즈베리파이 키패드에서 비밀번호 입력 받는 코드
                d = recvData.split("$")[0] # 비밀번호에 포함된 #을 삭제한다
                print(d)

```




```

db_pass = Apart.objects.filter(door_password=int(d)) #db에서 동 비밀번호와 라즈베리파이 입
력값을 비교하는 인증 코드
if db_pass: # 만약 db 비밀번호가 일치하면
    print("yes")
    sendData = "yes1"
    connectionSock.sendall(sendData.encode("utf-8"))
else:
    print("no")
    sendData = "no1"
    connectionSock.sendall(sendData.encode("utf-8"))
connectionSock.sendall(sendData.encode("utf-8"))
if '?' in recvData: #앱에서 동 비밀번호 입력시 서버 db와 비교해줌
    a = recvData.split("?")[1]
    print("a=" + a)
    d_pass = Apart.objects.filter(door_password=a) #서버db에서 동 비밀번호 인증
    if d_pass:
        print("db_ok")
        sendData = 'confirm' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
    else:
        print("db_np")
        sendData = 'no' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
if '/' in recvData: #앱에서 거주자 정보를 입력하여 서버 db와 비교해 인증한다/앱에서 자동 인증
절차
    b = recvData.split('/')
    ho1 = int(b[0])
    db = Apart.objects.filter(ho=ho1, name=b[1], phone=b[2], identification=b[3][:-2])#입력값은
호/이름/폰번호/주민번호 순으로 들어온다
    if '&' in b[3][:-1]:
        #print(raspberry)
        con = raspberry['3333$\n'] #라즈베리파이에게 인증 확인을 보내줌
        #print(type(con))
        #print(con)
        if db:
            print("db ok1")
            sendData = "yes1"
            con.sendall(sendData.encode("utf-8"))
            print("send ok")
            sendData2 = "confirm\n"
            connectionSock.sendall(sendData2.encode("utf-8"))
        else:
            print("db no1")
            sendData2 = "noooooo\n"
            connectionSock.sendall(sendData2.encode("utf-8"))
            # sendData ="no1"
            # con.sendall(sendData.encode("utf-8"))
    elif '#' in b[3][:-1]:

```



```

        if db:
            print("db ok2")
            p = Apart.objects.get(UUID__contains="@", name=b[1], identification=b[3][:-2]) #만약
인증이 성공하면 라즈베리파이 고유id인 UUID를 앱에 보내준다
            p = p.UUID
            sendData = 'confirm' + p + '\n'
            print(sendData)
            connectionSock.sendall(sendData.encode("utf-8"))
        else:
            print("db no2")
            sendData = 'noooooo@1' + '\n'
            sendData = 'noooooo@1' + '\n'
            connectionSock.sendall(sendData.encode("utf-8"))
        # connectionSock.sendall(sendData.encode("utf-8"))
        sendData = 'noooooo@1' + '\n'
        connectionSock.sendall(sendData.encode("utf-8"))
        # connectionSock.sendall(sendData.encode("utf-8"))
    except socket.error as e:
        print('socket error:{}'.format(e))
        break
    connectionSock.close()

if __name__ == '__main__':
    echoserver(5001)

```

bluetooth server

```

import socket, time
import serial

def test():
    input_pwd = input("password: ")
    return input_pwd

def pwd_ok():
    print("pwd_ok")
    s = serial.Serial('/dev/ttyACM0', 9600)
    s.flushInput()
    ok = "ok"
    s.write(ok.encode())
    read_pwd = s.readline()
    return read_pwd[:-2]

def run_client(host='18.220.63.77', port=5001):
    with socket.socket() as sock:
        sock.connect((host, port))
        data = "3333" + "$\n"
        print("data")

```



```

sock.sendall(data.encode('utf-8'))
print("send")
while True:
    if input()=="*":
        data = test() + "$"
        print(data)
        sock.sendall(data.encode('utf-8'))
        print(data)
        res = sock.recv(1024)
        pwd = res.decode('utf-8')
        print(pwd)
        if (pwd == 'yes1'):
            print("password ok")
            pwd_ok()
        elif (pwd == 'no1'):
            print("FAIL!!!")

if __name__ == '__main__':
    run_client()

```

raspberry pi client code

```

import socket, time
import serial

def test():
    input_pwd = input("password: ")
    return input_pwd

def pwd_ok():
    print("pwd_ok")
    s = serial.Serial('/dev/ttyACM0', 9600)
    s.flushInput()
    ok = "ok"
    s.write(ok.encode())
    read_pwd = s.readline()
    return read_pwd[:-2]

def run_client(host='18.220.63.77', port=5001):
    with socket.socket() as sock:
        sock.connect((host, port))
        data = "3333" + "$\n"
        print("data")
        sock.sendall(data.encode('utf-8'))
        print("send")
        while True:
            if input()=="*":
                data = test() + "$"

```



```

        print(data)
        sock.sendall(data.encode('utf-8'))
        print(data)
        res = sock.recv(1024)
        pwd = res.decode('utf-8')
        print(pwd)
        if (pwd == 'yes1'):
            print("password ok")
            pwd_ok()
        elif (pwd == 'no1'):
            print("FAIL!!!")

if __name__ == '__main__':
    run_client()

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.app8">
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="app8 with DB"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Main4Activity"> </activity>
        <activity android:name=".Main3Activity" />
        <activity android:name=".Main2Activity" />
        <activity android:name=".Main1Activity" />
        <activity android:name=".Main0Activity" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>

```

DBHelper

```
package com.example.app8;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper {

    // private static volatile DBHelper dbHelper;
    //
    // public static DBHelper getInstance(Context context) {
    //     if (dbHelper == null) {
    //         synchronized (DBHelper.class) {
    //             if (dbHelper == null) {
    //                 dbHelper = new DBHelper(context, "Resident", null, 1);
    //             }
    //         }
    //     }
    //     return dbHelper;
    // }

    // DBHelper 생성자로 관리할 DB 이름과 버전 정보를 받음
    public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    // DB를 새로 생성할 때 호출되는 함수
    @Override
    public void onCreate(SQLiteDatabase db) {
        // 새로운 테이블 생성
        /* 이름은 Resident이고, 자동으로 값이 증가하는 _id 정수형 기본키 컬럼과
        item 문자열 컬럼, price 정수형 컬럼, create_at 문자열 컬럼으로 구성된 테이블을 생성. */
        db.execSQL("CREATE TABLE Resident (_id INTEGER PRIMARY KEY AUTOINCREMENT, ho TEXT, name TEXT, phone TEXT, ide TEXT, uuid TEXT);");
    }

    // DB 업그레이드를 위해 버전이 변경될 때 호출되는 함수
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }

    public void DBClear() {
        SQLiteDatabase db = getReadableDatabase();
        db.execSQL("DROP TABLE Resident");
    }
}
```



```

        onCreate(db);
    }

    public void insert(String ho, String name, String phone, String ide, String uuid) {
        // 읽고 쓰기가 가능하게 DB 열기
        SQLiteDatabase db = getWritableDatabase();
        // DB에 입력한 값으로 행 추가
        db.execSQL("INSERT INTO Resident VALUES(null, '" + ho + "', '" + name + "', '" + phone + "', '" + ide + "', '" + uuid + "');");
        // db.execSQL("INSERT INTO Resident (ho, name, phone, ide, uuid) Values ('" + ho + "', '" + name + "', '" + phone + "', '" + ide + "', '" + uuid + "');");
        db.close();
    }

    public void update(String ho, String name, String phone, String ide, String uuid) {
        SQLiteDatabase db = getWritableDatabase();
        // 입력한 항목과 일치하는 행의 가격 정보 수정
        db.execSQL("UPDATE Resident SET ho='" + ho + "', phone='" + phone + "', ide='" + ide + "', uuid='" + uuid + "' WHERE name='" + name + "';");
        db.close();
    }

    public void delete(String name) {
        SQLiteDatabase db = getWritableDatabase();
        // 입력한 항목과 일치하는 행 삭제
        db.execSQL("DELETE FROM Resident WHERE name='" + name + "';");
        db.close();
    }

    public void deleteAll() {
        SQLiteDatabase db = getReadableDatabase();
        db.execSQL("Delete from Resident");
        db.close();
    }

    public String getUuid() {
        //uuid 조회
        SQLiteDatabase db = getReadableDatabase();
        String result = "";
        //DB에 있는 데이터를 쉽게 처리하기 위해 Cursor를 사용하여 테이블에 있는 uuid 출력
        Cursor cursor = db.rawQuery("SELECT * FROM Resident", null);
        while (cursor.moveToNext()) {
            result = cursor.getString(5);
        }
        return result;
    }
}

```



```
public String getResult() {
    // 읽기가 가능하게 DB 열기
    SQLiteDatabase db = getReadableDatabase();
    String result = "";

    // DB에 있는 데이터를 쉽게 처리하기 위해 Cursor를 사용하여 테이블에 있는 모든 데이터 출력
    Cursor cursor = db.rawQuery("SELECT * FROM Resident", null);
    while (cursor.moveToNext()) {
        result += cursor.getString(0)
            + ". "
            + cursor.getString(1)
            + "호 - 이름 : "
            + cursor.getString(2)
            + ", 전화번호 : "
            + cursor.getString(3)
            + ", 주민등록번호 : "
            + cursor.getString(4)
            + ". UUID : "
            + cursor.getString(5)
            + "\n";
    }

    return result;
}

public String openDoor() {
    // 읽기가 가능하게 DB 열기
    SQLiteDatabase db = getReadableDatabase();
    String result = "";

    // DB에 있는 데이터를 쉽게 처리하기 위해 Cursor를 사용하여 테이블에 있는 모든 데이터 출력
    Cursor cursor = db.rawQuery("SELECT * FROM Resident", null);
    while (cursor.moveToNext()) {
        result += cursor.getString(1)
            + "/"
            + cursor.getString(2)
            + "/"
            + cursor.getString(3)
            + "/"
            + cursor.getString(4)
            + "&";
    }

    return result;
}
}
```



MainActivity

```
package com.example.app8;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.os.SystemClock;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.net.Socket;
import java.util.Set;
import java.util.UUID;

public class MainActivity extends AppCompatActivity {

    //블루투스 소켓통신 환경설정
    BluetoothAdapter mBluetoothAdapter;
    Set<BluetoothDevice> mPairedDevices;
    Handler mBluetoothHandler;
    ConnectedBluetoothThread mThreadConnectedBluetooth;
    BluetoothDevice mBluetoothDevice;
    BluetoothSocket mBluetoothSocket;
    final static int BT_MESSAGE_READ = 2;
    final static UUID BT_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    //서버 소켓통신 환경설정
    private Socket clientSocket;
    private BufferedReader socketIn;
    private PrintWriter socketOut;
    private int port = 5000;
    private final String ip = "18.220.63.77";
    private MyHandler myHandler;
```



```
private MyThread myThread;

// DB를 위한 환경설정
public static final String DB_name = "Resident.db";
public DBHelper dbHelper;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    dbHelper = new DBHelper(getApplicationContext(), DB_name, null, 1);

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

    mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    bluetoothrasberry();//실행코드
    mThreadConnectedBluetooth.write("hello I'm App");

}

//실행코드
void bluetoothrasberry() {
    if (mBluetoothAdapter.isEnabled()) {
        mPairedDevices = mBluetoothAdapter.getBondedDevices();
        //라즈베리파이가 있으면 항상 연결
        System.out.println("check1");
        connectSelectedDevice("raspberrypi");
    }
}

//라즈베리파이에서 받아온 데이터값이 DB같으면 실행되는 서버소켓통신코드
void runserver(){
    //readMessage가 DB의 uuid값과 일치하면
    //서버와 통신시작
    try {
        clientSocket = new Socket(ip, port);
        socketIn = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    socketOut.println("DOOR");//서버에게 페이지 알림

    myHandler = new MyHandler();
}
```



```

myThread = new MyThread();
myThread.start();
System.out.println(dbHelper.openDoor());
socketOut.println(dbHelper.openDoor());
//      socketOut.println("1"+"/"+"Kimsubin"+"/"+"010-1111-1111"+"/"+"111111-111111"+"&");
}

//서버소켓통신 Thread
class MyThread extends Thread {
    boolean stop = false;

    public void stopThread() {
        stop = true;
    }
    @Override
    public void run() {
        while (true) {
            try {
                // InputStream의 값을 읽어와서 data에 저장
                String data = socketIn.readLine();
                // Message 객체를 생성, 핸들러에 정보를 보낼 땐 이 메시지 객체를 이용
                Message servermsg = myHandler.obtainMessage();
                servermsg.obj = data;
                myHandler.sendMessage(servermsg);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

//서버소켓통신 Handler
class MyHandler extends Handler {
    @Override
    public void handleMessage(Message servermsg) {
        String a = servermsg.obj.toString();
        String reala = a.substring(0,7);
        if(reala.equals("confirm")) { //보낸 비번이 인증확인이면,
            System.out.println(reala);
        }
    }
}

//블루투스 연결하는 메소드
void connectSelectedDevice(String selectedDeviceName) {
    for (BluetoothDevice tempDevice : mPairedDevices) {
        if (selectedDeviceName.equals(tempDevice.getName())) {

```

```
mBluetoothDevice = tempDevice;
System.out.println(tempDevice.getName());
break;
}
}
//실제로 블루투스 장치와 연결하는 부분이다.
// 우리가 연결에 필요한 값은 장치의 주소이다.
// 따라서 for 문으로 페어링된 모든 장치를 검색을 하면서 매개 변수 값과 비교하여 같다면 그 장치의
주소 값을 얻어오는 부분이다.
try {
    mBluetoothSocket = mBluetoothDevice.createRfcommSocketToServiceRecord(BT_UUID);
    mBluetoothSocket.connect();
    mBluetoothHandler = new ConnectedBluetoothHandler();
    mThreadConnectedBluetooth = new ConnectedBluetoothThread(mBluetoothSocket);
    mThreadConnectedBluetooth.start();
    Toast.makeText(getApplicationContext(), "라즈베리파이와 연결되었습니다.",
    Toast.LENGTH_LONG).show();
    // mBluetoothHandler.obtainMessage(BT_CONNECTING_STATUS, 1, -1).sendToTarget();
} catch (IOException e) {
    Toast.makeText(getApplicationContext(), "블루투스 연결 중 오류가 발생했습니다.",
    Toast.LENGTH_LONG).show();
}
}
//mBluetoothDevice를 통해 createRfcommSocketToServiceRecord(UUID)를 호출하여 mBluetoothSocket을
가져온다. (참고로 여기서 사용된 UUID 값은 시리얼 통신용이다.)
//그러면 mBluetoothDevice에 연결 될 mBluetoothSocket이 초기화된다. 그 후 connect()를 호출하여 연결을
시작한다.

//ConnectedBluetoothThread 스레드
private class ConnectedBluetoothThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    //ConnectedBluetoothThread 스레드의 시작이며 이 스레드에서 사용할 전역 객체들을 선언하였다.
    // 위에서 사용한 소켓이 이미 메인 액티비티 자체의 소켓이니 그대로 사용해도 되지만 스레드 내부 자
체에서만 사용할 소켓 객체를 추가하였다.
    //
    public ConnectedBluetoothThread(BluetoothSocket socket) {
        mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;
        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "소켓 연결 중 오류가 발생했습니다.",
            Toast.LENGTH_LONG).show();
        }
        mmInStream = tmpIn;
```

```
        mmOutputStream = tmpOut;
    }
    public void run() {
        byte[] buffer = new byte[1024];
        int bytes;
        while (true) {
            try {
                bytes = mmInStream.available();
                if (bytes != 0) {
                    SystemClock.sleep(100);
                    bytes = mmInStream.available();
                    bytes = mmInStream.read(buffer, 0, bytes);
//                Message msg = mBluetoothHandler.obtainMessage();
                mBluetoothHandler.obtainMessage(BT_MESSAGE_READ,
                    bytes,
                    -1,
buffer).sendToTarget();
            }
            } catch (IOException e) {
                break;
            }
        }
    }
    public void write(String str) {
        byte[] bytes = str.getBytes();
        try {
            mmOutputStream.write(bytes);
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "데이터 전송 중 오류가 발생했습니다.",
Toast.LENGTH_LONG).show();
        }
    }
    public void cancel() {
        try {
            mmSocket.close();
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "소켓 해제 중 오류가 발생했습니다.",
Toast.LENGTH_LONG).show();
        }
    }
}

//ConnectedBluetoothHandler 핸들러
private class ConnectedBluetoothHandler extends Handler{
    public void handleMessage( android.os.Message msg) {
        if (msg.what == BT_MESSAGE_READ) {
            String readMessage = null;
            try {
                readMessage = new String((byte[]) msg.obj, "UTF-8");
            } catch (UnsupportedEncodingException e) {
```

```
        e.printStackTrace();
    }
    System.out.println("블루투스핸들러");
    String receiveuuid = readMessage.substring(0,36);
    if (dbHelper.getUuid().equals(receiveuuid)){
        runserver();
    }
}
}

//다음창으로 넘어가기 버튼
public void apart(View v) {
    Intent intent = new Intent(this, Main0Activity.class);
    startActivity(intent);
}
}
```

activity_main.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="#21529C"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView5"
        android:layout_width="337dp"
        android:layout_height="118dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="No touch Entrance"
        android:textAlignment="center"
        android:textColor="@android:color/white"
        android:textIsSelectable="false"
        android:textSize="40sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.439"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>
```

```
        app:layout_constraintVertical_bias="0.175" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="471dp"
    android:layout_height="962dp"
    android:layout_marginTop="88dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/apartment" />

<Button
    android:id="@+id/appregist"
    android:layout_width="226dp"
    android:layout_height="75dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:backgroundTint="@android:color/background_light"
    android:onClick="apart"
    android:text="거주자 APP 등록"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="normal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.404" />

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

Main0Activity

```
package com.example.app8;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class Main0Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```



```

        setContentView(R.layout.activity_main0);
    }
    public void apart1(View v){
        Intent intent0 = new Intent(this, Main1Activity.class);
        startActivity(intent0);
    }
    public void buttonback(View v){
        finish();
    }
}

```

activity_main0.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="@android:color/holo_blue_light"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView5"
        android:layout_width="322dp"
        android:layout_height="59dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="아파트를 고르세요"
        android:textAlignment="center"
        android:textColor="@android:color/white"
        android:textIsSelectable="false"
        android:textSize="36sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.444"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.134" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="471dp"
        android:layout_height="962dp"
        android:layout_marginTop="136dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

```

```
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/apartment" />
```

```
<Button
```

```
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="@android:color/white"
    android:onClick="buttonback"
    android:text="뒤로가기"
    android:textColor="@android:color/black"
    android:textStyle="normal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.04"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.023" />
```

```
<Button
```

```
    android:id="@+id/button"
    android:layout_width="@android:dimen/thumbnail_width"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:backgroundTint="@android:color/background_light"
    android:onClick="apart1"
    android:text="왕산 현대아파트"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="normal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.528" />
```

```
<Button
```

```
    android:id="@+id/button6"
    android:layout_width="@android:dimen/thumbnail_height"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
```



```
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:backgroundTint="@android:color/background_light"
android:onClick="apart1"
android:text="왕산 롯데캐슬"
android:textColor="@android:color/black"
android:textSize="24sp"
android:textStyle="normal"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.402" />
```

<Button

```
android:id="@+id/button4"
android:layout_width="@android:dimen/thumbnail_width"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:backgroundTint="@android:color/background_light"
android:onClick="apart1"
android:text="왕산 푸르지오"
android:textColor="@android:color/black"
android:textSize="24sp"
android:textStyle="normal"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.296" />
```

</android.support.constraint.ConstraintLayout>

Main1Activity

```
package com.example.app8;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class Main1Activity extends AppCompatActivity {
```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main1);
}
public void button101(View v){
    Intent intent1 = new Intent(this, Main2Activity.class);
    startActivity(intent1);
}
public void button102(View v){
    Intent intent1 = new Intent(this, Main2Activity.class);
    startActivity(intent1);
}
public void button103(View v){
    Intent intent1 = new Intent(this, Main2Activity.class);
    startActivity(intent1);
}
//뒤로가기 버튼
public void buttonback(View v){
    finish();
}
}

```

activity_main1.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="#C698E0FF"
    tools:context=".Main1Activity">

    <Button
        android:id="@+id/button2"
        android:layout_width="@android:dimen/thumbnail_height"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:background="@android:color/white"
        android:onClick="button102"
        android:text="@string/button102"
        android:textColor="@android:color/black"
        android:textSize="24sp"
        android:textStyle="bold"

```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.418" />
```

<ImageView

```
android:id="@+id/imageView2"
android:layout_width="471dp"
android:layout_height="962dp"
android:layout_marginTop="88dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/apartment" />
```

<Button

```
android:id="@+id/button3"
android:layout_width="@android:dimen/thumbnail_height"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:onClick="button103"
android:text="@string/button103"
android:textColor="@android:color/black"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.541" />
```

<TextView

```
android:id="@+id/textView"
android:layout_width="253dp"
android:layout_height="56dp"
android:text="동을 선택하세요"
android:textAlignment="center"
android:textColor="@android:color/black"
android:textSize="36sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
```

```
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.12" />
```

<Button

```
android:id="@+id/password"
android:layout_width="@android:dimen/thumbnail_height"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:onClick="button101"
android:text="@string/button101"
android:textColor="@android:color/black"
android:textColorHighlight="#D81B60"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.283" />
```

<Button

```
android:id="@+id/button5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:onClick="buttonback"
android:text="뒤로가기"
android:textColor="@android:color/black"
android:textStyle="normal"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.04"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.023" />
```

</android.support.constraint.ConstraintLayout>

Main2Activity

```
package com.example.app8;
```



```

import android.os.Handler;
import android.os.Message;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.StrictMode;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
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.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class Main2Activity extends AppCompatActivity {
    //    public static final String EXTRA_MESSAGE = "com.example.APP5.PASSWORD"
    Intent intent2;
    //서버 소켓통신 환경설정
    private Socket clientSocket;
    private BufferedReader socketIn;
    private PrintWriter socketOut;
    private int port = 5555;
    private final String ip = "18.220.63.77";
    private MyHandler myHandler;
    private MyThread myThread;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        intent2 = new Intent(getApplicationContext(),Main3Activity.class); //비번버튼눌렀을때 페이지 넘기는
intent

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

        //서버와 통신시작
        try {
            clientSocket = new Socket(ip, port);

```



```

        socketIn = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    socketOut.println("P2");//서버에게 페이지 알림

    myHandler = new MyHandler();
    myThread = new MyThread();
    myThread.start();

    Button btn = (Button) findViewById(R.id.password);//버튼 변수 설정
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //패스워드확인버튼 눌렀을때 : textview의 값받아와서 비밀번호와 확인!
            EditText text_pw = (EditText) findViewById(R.id.pwtext);//안드로이드 위젯(글자입력)을 쓸꺼다
            String pw = text_pw.getText().toString();//pw문자열변수에 위의 text_pw에서 텍스트를 가져와
문자열로 바꿔서 사용
            socketOut.println("?" + pw);
        }
    });
}
class MyThread extends Thread {
    boolean stop = false;

    public void stopThread() {
        stop = true;
    }
    @Override
    public void run() {
        while (true) {
            try {
                // InputStream의 값을 읽어와서 data에 저장
                String data = socketIn.readLine();
                // Message 객체를 생성, 핸들러에 정보를 보낼 땐 이 메세지 객체를 이용
                Message msg = myHandler.obtainMessage();
                msg.obj = data;
                myHandler.sendMessage(msg);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
class MyHandler extends Handler {
    @Override

```

```
public void handleMessage(Message msg) {
    if(msg.obj.toString().equals("confirm")) { //보낸 비번이 인증확인이면,
        startActivity(intent2); //다음페이지로 넘어감
    }
    else{
        new AlertDialog.Builder(Main2Activity.this)
            .setTitle("오류 : 동 비밀번호가 틀립니다")
            .setMessage("\n\n 비밀번호를 다시 입력하십시오")
            .setNeutralButton("닫기", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dlg, int sumthin) {
                }
            })
            .show();
    }
}

public void buttonback(View v){
    finish();
}
}
```

activity_main2.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="#C698E0FF"
    tools:context=".Main2Activity">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="471dp"
        android:layout_height="962dp"
        android:layout_marginTop="88dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/apartment" />

    <TextView
        android:id="@+id/viewtext3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="동 비밀번호를 입력하세요"
```

```
android:textColor="@android:color/black"
android:textSize="36sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.545"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.243" />
```

<Button

```
android:id="@+id/password"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:text="확인"
android:textSize="18sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.912"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.407" />
```

<EditText

```
android:id="@+id/pwtext"
android:layout_width="224dp"
android:layout_height="45dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:ems="10"
android:hint="@string/password"
android:inputType="number"
android:textAlignment="center"
android:textColorHint="#FF1744"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.181"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.405" />
```




```

<Button
    android:id="@+id/main2backbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="@android:color/white"
    android:onClick="buttonback"
    android:text="뒤로가기"
    android:textSize="14sp"
    android:textStyle="normal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.04"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.011" />

```

```

</android.support.constraint.ConstraintLayout>

```

MAin3Activity

```

package com.example.app8;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Arrays;

public class Main3Activity extends AppCompatActivity {
    Intent intent3;

```



```

//서버 소켓통신 환경설정
private Socket clientSocket;
private BufferedReader socketIn;
private PrintWriter socketOut;
private int port = 5555;
private final String ip = "18.220.63.77";
private MyHandler myHandler;
private MyThread myThread;

// handler에서 사용하기위한 전역변수
String inf1, inf2, inf3, inf4;

// DB를 위한 환경설정
public static final String DB_name = "Resident.db";
public DBHelper dbHelper;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main3);
    dbHelper = new DBHelper(getApplicationContext(), DB_name, null, 1);

    intent3 = new Intent(getApplicationContext(),Main4Activity.class);//다음페이지로 넘어가는 intent

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

    //서버와 통신시작
    try {
        clientSocket = new Socket(ip, port);
        socketIn = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
        socketOut = new PrintWriter(clientSocket.getOutputStream(), true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    socketOut.println("P3");//서버에게 페이지 알림

    myHandler = new Main3Activity.MyHandler();
    myThread = new Main3Activity.MyThread();
    myThread.start();

    Button infbut = (Button) findViewById(R.id.infbutton);//버튼 변수 설정
    infbut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //거주자정보확인버튼 눌렀을때 : textview의 값받아와서 비밀번호와 확인!

```

```
        EditText text_inf1 = (EditText) findViewById(R.id.inf1);
        inf1 = text_inf1.getText().toString();
        EditText text_inf2 = (EditText) findViewById(R.id.inf2);
        inf2 = text_inf2.getText().toString();
        EditText text_inf3 = (EditText) findViewById(R.id.inf3);
        inf3 = text_inf3.getText().toString();
        EditText text_inf4 = (EditText) findViewById(R.id.inf4);
        inf4 = text_inf4.getText().toString();

        //서버로 보내기
        socketOut.println(inf1+"/"+inf2+"/"+inf3+"/"+inf4+"#");
    }
});

}

class MyThread extends Thread {
    @Override
    public void run() {
        while (true) {
            try {
                // InputStream의 값을 읽어와서 data에 저장
                String data = socketIn.readLine();
                // Message 객체를 생성, 핸들러에 정보를 보낼 땐 이 메세지 객체를 이용
                Message msg = myHandler.obtainMessage();
                msg.obj = data;
                myHandler.sendMessage(msg);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

class MyHandler extends Handler {
    @Override
    public void handleMessage(Message msg) {
        String servermsg = msg.obj.toString();//서버에서 준 데이터는 인증결과+UUID값이다.
        String servermsg_confirm=servermsg.substring(0,7);//서버에서 준 데이터중 인증결과만 뜻하는 변수
        String servermsg_uuid = servermsg.substring(8);

        //보낸 거주자 정보가 웹디비랑 인증확인이면,
        if(servermsg_confirm.equals("confirm") && inf1 != null && inf2 != null && inf3 != null && inf4
        != null) {
            dbHelper.insert(inf1, inf2, inf3, inf4, servermsg_uuid);
            System.out.println(dbHelper.getResult());
            System.out.println(dbHelper.getUuid());
            System.out.println("dbconfirm");
            startActivity(intent3);//다음페이지로 넘어감
        }
    }
}
```

```

    }
    else{
        new AlertDialog.Builder(Main3Activity.this)
            .setTitle("오류 : 거주자 정보가 틀립니다")
            .setMessage("WnWn 거주자 정보를 다시 입력하십시오")
            .setNeutralButton("닫기", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dlg, int sumthin) {
                }
            })
            .show();
    }
}
}
public void buttonback2(View v){
    finish();
}
}
}

```

activity_main3.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="#C698E0FF"
    tools:context=".Main3Activity">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="471dp"
        android:layout_height="962dp"
        android:layout_marginTop="88dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/apartment" />

    <Button
        android:id="@+id/infbbutton"
        android:layout_width="107dp"
        android:layout_height="50dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:background="@android:color/white"

```

```
android:text="확인"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.541"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.726" />
```

<Button

```
android:id="@+id/main3backbutton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:onClick="buttonback2"
android:text="뒤로가기"
android:textSize="14sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.049"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.011" />
```

<TextView

```
android:id="@+id/viewtext3"
android:layout_width="140dp"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:text="@string/ph"
android:textAlignment="center"
android:textColor="#020000"
android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.031"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.53" />
```

<TextView

```
    android:id="@+id/viewtext4"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="@android:color/white"
    android:text="주민번호"
    android:textAlignment="center"
    android:textColor="#020000"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.031"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.659" />
```

<EditText

```
    android:id="@+id/inf1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:hint="@string/hosu"
    android:inputType="number"
    android:textColorHighlight="#FF1744"
    android:textColorHint="#FF1744"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.857"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.29000002" />
```

<TextView

```
    android:id="@+id/viewtext2"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="@android:color/white"
```

```
android:text="이름"
android:textAlignment="center"
android:textColor="#020000"
android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.031"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.42" />
```

<TextView

```
android:id="@+id/viewtext1"
android:layout_width="140dp"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@android:color/white"
android:text="호수"
android:textAlignment="center"
android:textColor="#020000"
android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.031"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.29" />
```

<EditText

```
android:id="@+id/inf2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:ems="10"
android:hint="@string/name"
android:inputType="text"
android:textColorHint="#FF1744"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.857"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.41" />
```

<TextView

```
android:id="@+id/viewtext"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:text="거주자 정보를 입력하세요"
android:textColor="@android:color/black"
android:textSize="36sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.164" />
```

<EditText

```
android:id="@+id/inf3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:ems="10"
android:hint="@string/phonenummer"
android:inputType="text"
android:textColorHint="#FF1744"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.857"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.526" />
```

<EditText

```
android:id="@+id/inf4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
```




```

        android:ems="10"
        android:hint="@string/jumin"
        android:inputType="text"
        android:textColorHint="#FF1744"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.857"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.653" />

```

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

MAin4Activity

```

package com.example.app8;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class Main4Activity extends AppCompatActivity {

    DBHelper dbHelper = new DBHelper(this, "Resident", null, 1);
    //    DBHelper dbHelper = new DBHelper(getApplicationContext(), "Resident", null, 1);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main4);

        System.out.println("check");
        System.out.println(dbHelper.getResult());
        TextView result = (TextView) findViewById(R.id.result);
        result.setText(dbHelper.getResult());
    }
}

```

activity_main4.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="#C698E0FF"
    tools:context=".Main4Activity">

```

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="471dp"
    android:layout_height="962dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/apartment" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="322dp"
    android:layout_height="70dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="앱 등록 완료"
    android:textAlignment="center"
    android:textColor="#E53935"
    android:textSize="36sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.196" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="322dp"
    android:layout_height="70dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="거주자로 확인되었습니다"
    android:textAlignment="center"
    android:textColor="@android:color/black"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.462" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="322dp"
    android:layout_height="70dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="축하드립니다!"
    android:textAlignment="center"
    android:textColor="@android:color/black"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.351" />

<TextView
    android:id="@+id/result"
    android:layout_width="316dp"
    android:layout_height="229dp"
    android:layout_marginStart="32dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="32dp"
    android:gravity="center"
    android:text="TextView"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

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

//-----

dc motor가 문제가 생겨 LED로 대체했습니다.

```
int trig_pin = 8;
int echo_pin = 9;
char pwd_ok = '.';
int ledPin = 5; // LED를 제어할 디지털IO 포트번호를 설정합니다.

void setup() {
    Serial.begin(9600);
    pinMode(trig_pin, OUTPUT);
    pinMode(echo_pin, INPUT);
```



```
pinMode (ledPin, OUTPUT); // LED가 연결된 디지털IO포트를 출력으로 설정
}

void loop() {
    digitalWrite(trig_pin, LOW);
    digitalWrite(echo_pin, LOW);
    delayMicroseconds(2);
    digitalWrite(trig_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_pin, LOW);

    if (Serial.available()>0) { //키패드 사용하는 경우
        pwd_ok = Serial.read();
        digitalWrite (ledPin, HIGH); // 5번핀을 HIGH(약 5V)로 출력
        delay (2000); // 딜레이 설정(2000=2초)
        delay(500);
        Serial.println("pwd_ok");
    }

    unsigned long duration = pulseIn(echo_pin, HIGH);

    float distance = duration / 29.0 / 2.0;
    Serial.println(distance);
    //초음파 센서를 사용하는 경우
    if(distance < 10) {
        digitalWrite (ledPin, HIGH); // 5번핀을 HIGH(약 5V)로 출력
        delay (2000); // 딜레이 설정(2000=2초)
    }
    else{
        digitalWrite (ledPin, LOW); // 5번핀을 LOW(약 0V)로 출력
        delay (2000); // 딜레이 설정(2000=2초)
    }

    delay(500);
}
```