Wi-Fi based Indoor Navigation System

Team 13

201935018 김민성

201935068 신민철

201935112 이지원

202035317 김선정

Contents

1 Introduction

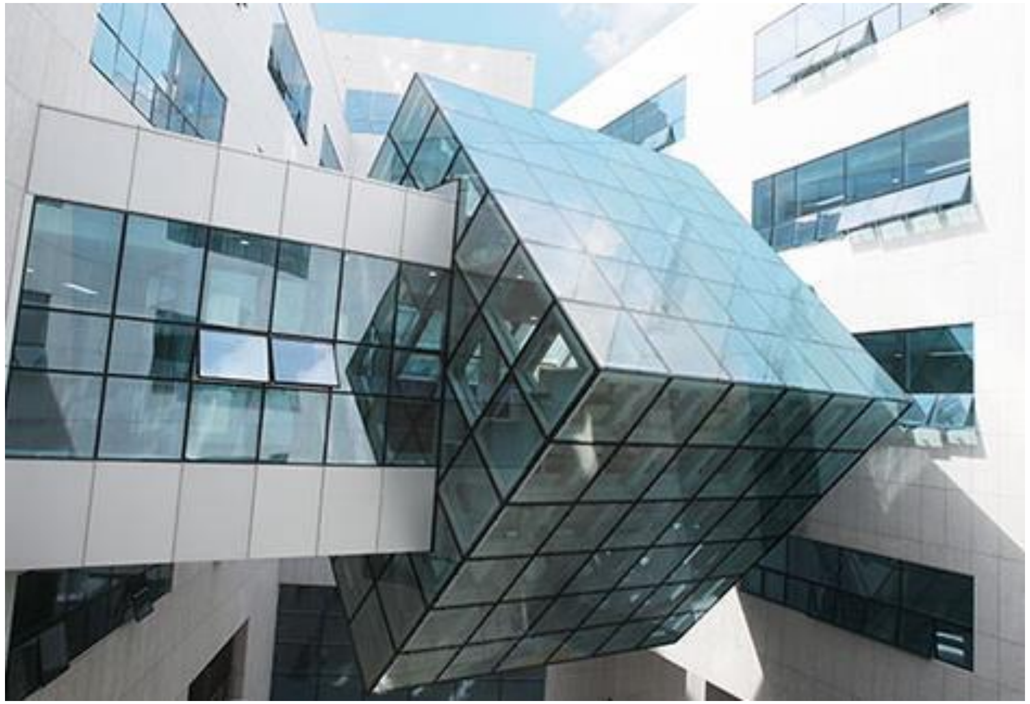
2 System architecture

3 Progress

4 Remaining task

Part 1

Introduction



Project Goal

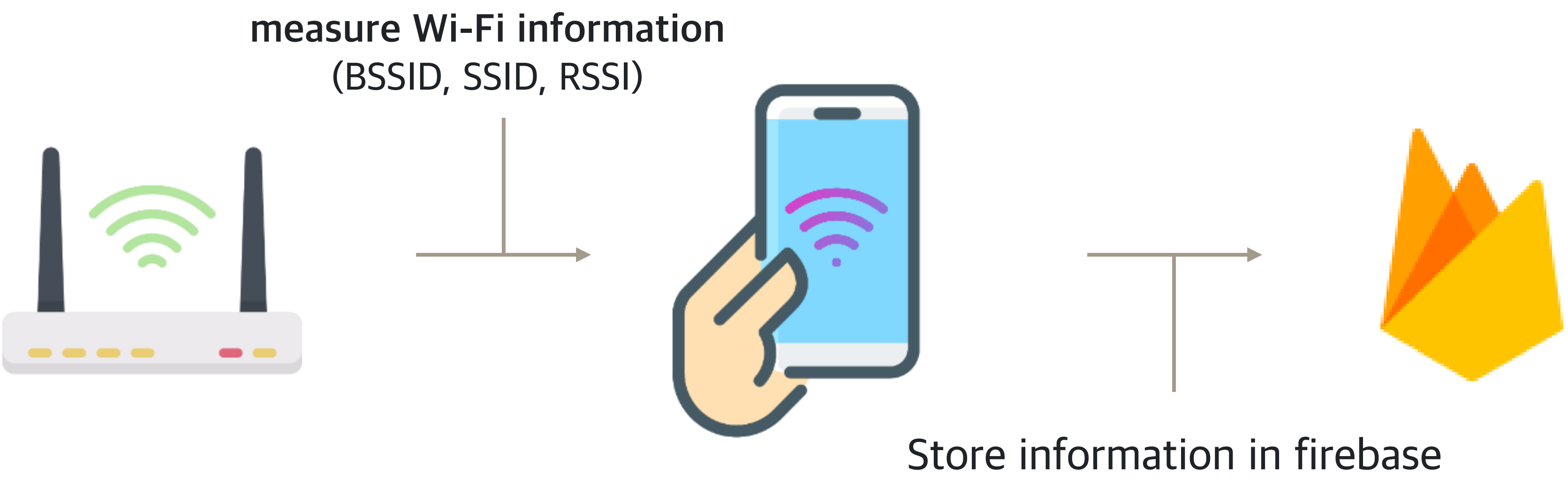
- Provides Wi-Fi based indoor navigation services using Android devices.
- Provides navigation functions for the 4th & 5th floor of AI building

Part 2

System architecture

1. Admin app

Overall architecture



measure Wi-Fi information
(BSSID, SSID, RSSI)

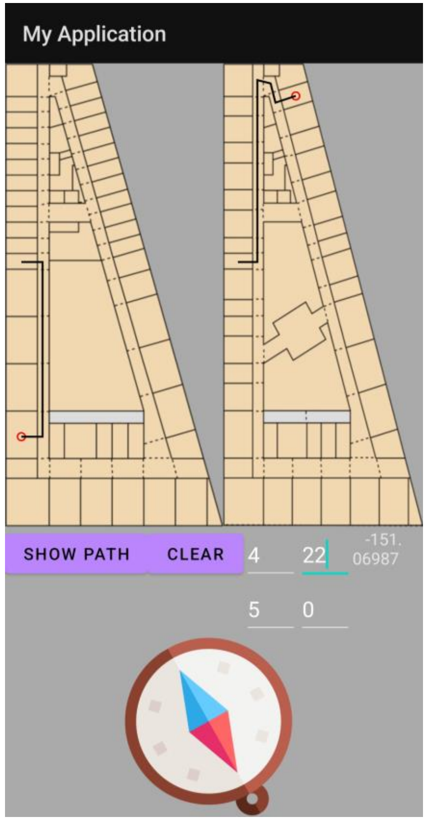
Store information in firebase

Part 2

System architecture

2. User app

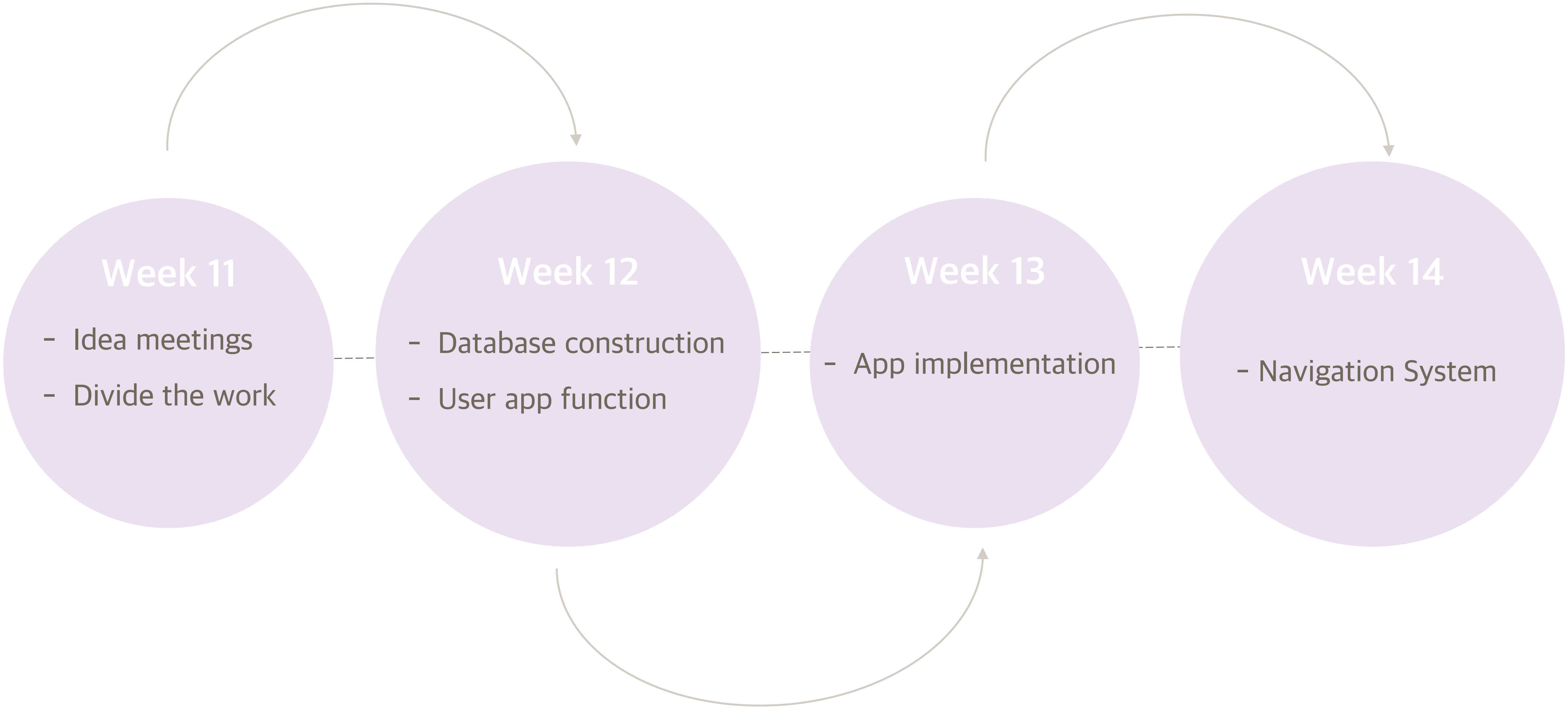
Overall architecture



- Use map to recognize a coordinate of each lecture room
- When you set up the start and arrival points, the map shows you the route
- The compass indicates where the user is looking within the AI building

Part 3

Progress



Week 11

- Idea meetings
- Divide the work

Week 12

- Database construction
- User app function

Week 13

- App implementation

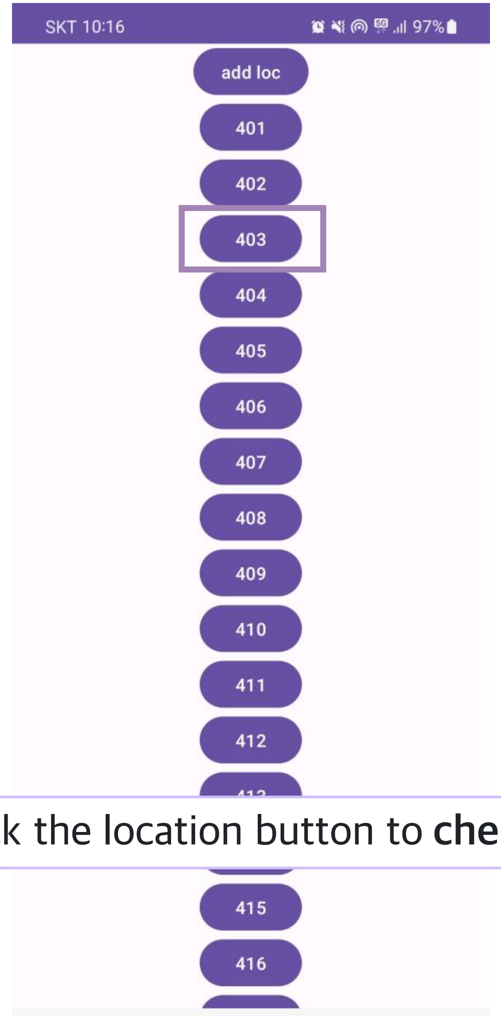
Week 14

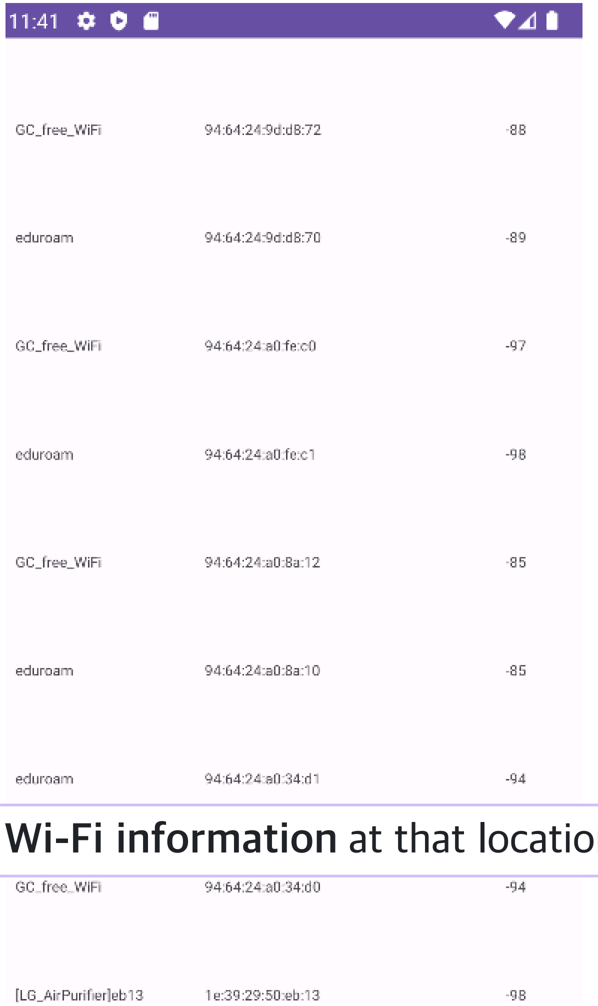
- Navigation System

Part 3

Progress

1. Admin app



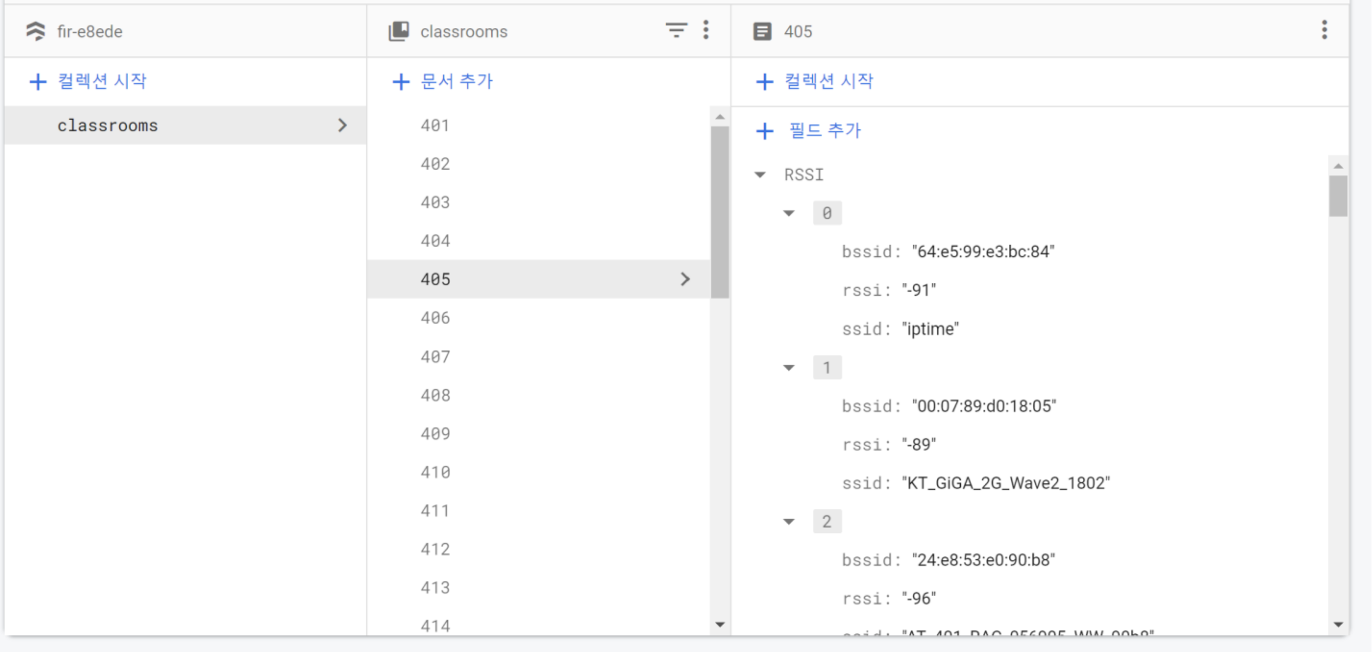


```
GC_free_WiFi          94:64:24:9d:d8:72          -88
eduroam               94:64:24:9d:d8:70          -89
GC_free_WiFi          94:64:24:a0:fe:c0          -97
eduroam               94:64:24:a0:fe:c1          -98
GC_free_WiFi          94:64:24:a0:8a:12          -85
eduroam               94:64:24:a0:8a:10          -85
eduroam               94:64:24:a0:34:d1          -94
GC_free_WiFi          94:64:24:a0:34:d0          -94
[LG_AirPurifier]eb13  1e:39:29:50:eb:13          -98
```

Click the location button to **check the measured Wi-Fi information** at that location

Progress

1. Admin app



```
classrooms
401
402
403
404
405
406
407
408
409
410
411
412
413
414
```

```
RSSI
  0
    bssid: "64:e5:99:e3:bc:84"
    rssi: "-91"
    ssid: "iptime"
  1
    bssid: "00:07:89:d0:18:05"
    rssi: "-89"
    ssid: "KT_GiGA_2G_Wave2_1802"
  2
    bssid: "24:e8:53:e0:90:b8"
    rssi: "-96"
    ssid: "AT_401_RAC_056005_WW_00b8"
```

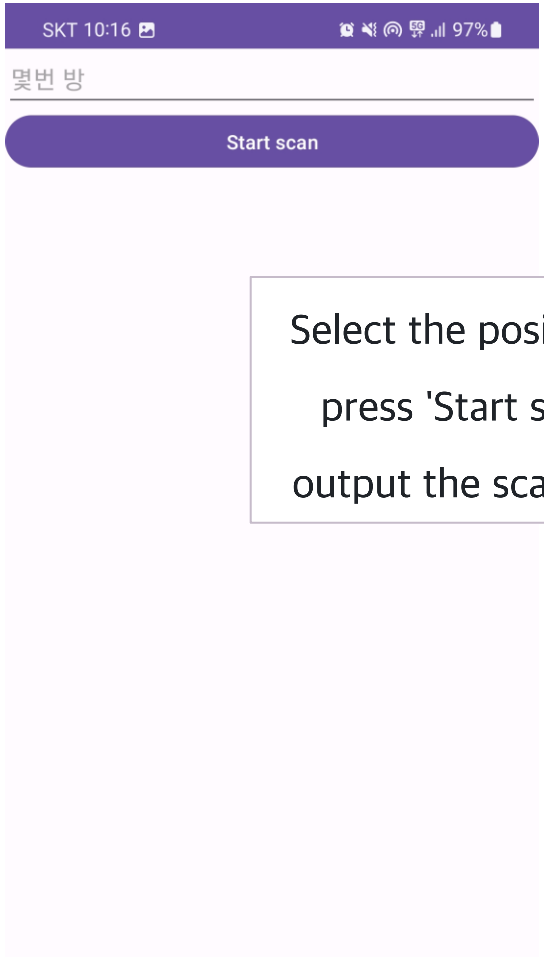
Store the Wi-Fi information measured for each classroom in the database

Stored information ⇨ BSSID, SSID, RSSI

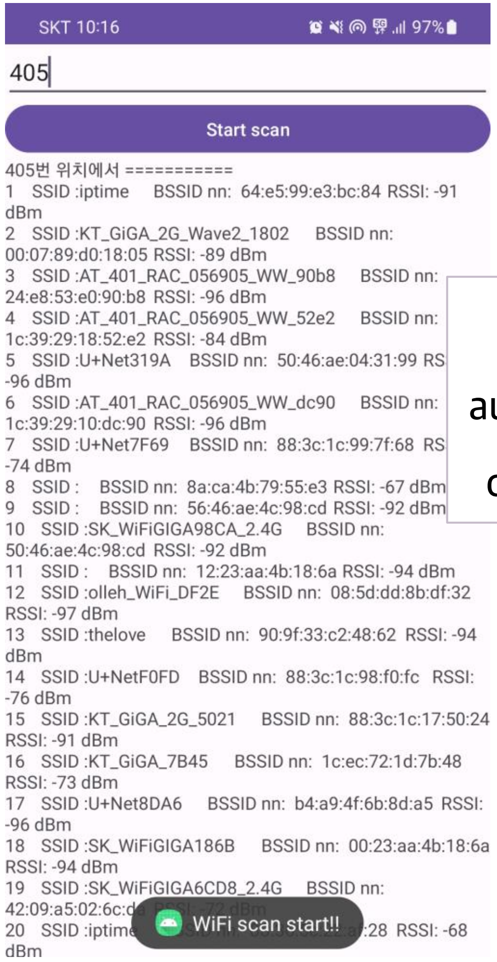
Part 3

Progress

1. Admin app



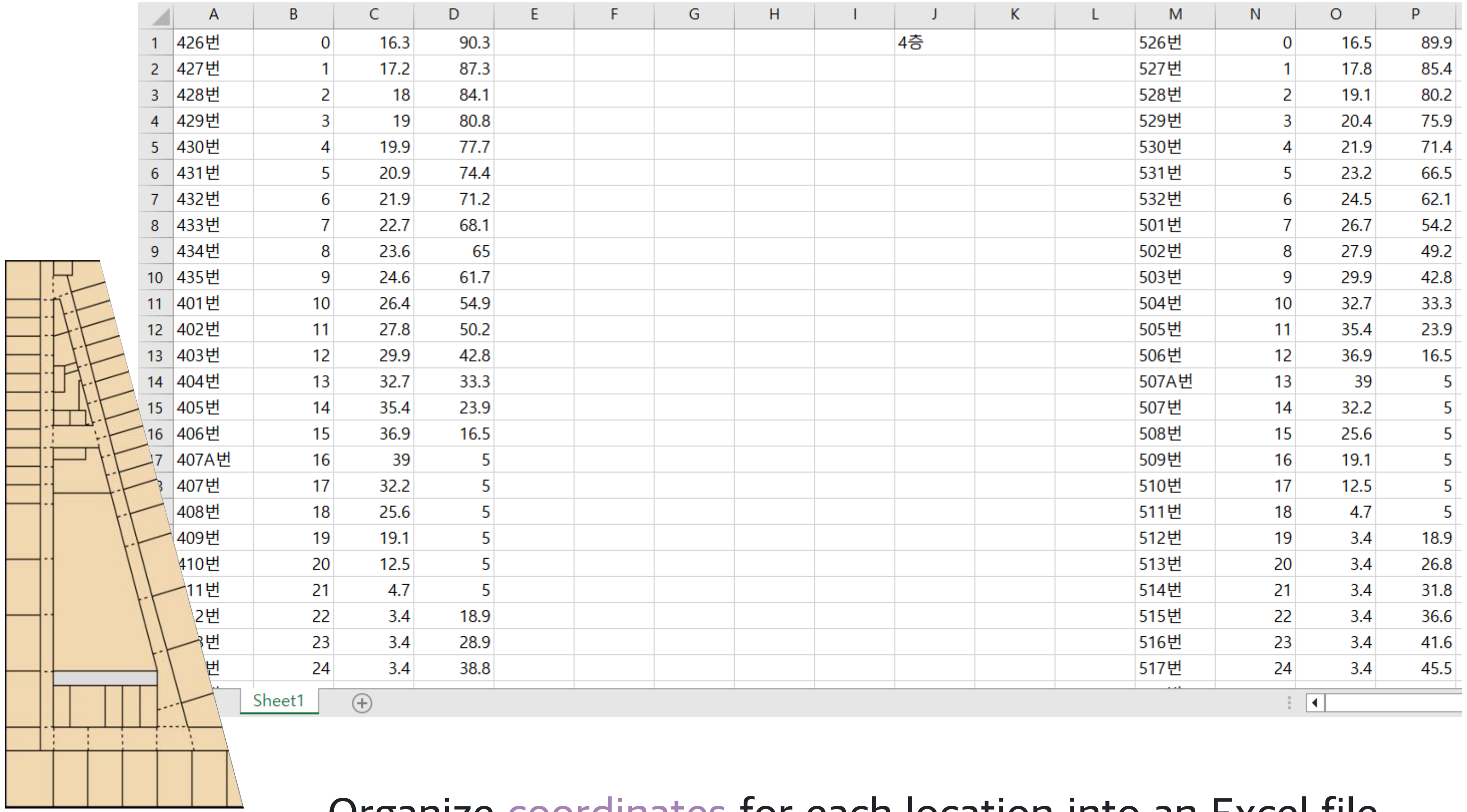
Select the position and press 'Start scan' to output the scan values



```
405번 위치에서 ===========
1   SSID :iptime    BSSID nn:  64:e5:99:e3:bc:84 RSSI: -91
dBm
2   SSID :KT_GiGA_2G_Wave2_1802     BSSID nn:
00:07:89:d0:18:05 RSSI: -89 dBm
3   SSID :AT_401_RAC_056905_WW_90b8     BSSID nn:
24:e8:53:e0:90:b8  RSSI: -96 dBm
4   SSID :AT_401_RAC_056905_WW_52e2     BSSID nn:
1c:39:29:18:52:e2  RSSI: -84 dBm
5   SSID :U+Net319A    BSSID nn:  50:46:ae:04:31:99 RS
-96 dBm
6   SSID :AT_401_RAC_056905_WW_dc90     BSSID nn:
1c:39:29:10:dc:90  RSSI: -96 dBm
7   SSID :U+Net7F69    BSSID nn:  88:3c:1c:99:7f:68  RS
-74 dBm
8   SSID :    BSSID nn:  8a:ca:4b:79:55:e3 RSSI: -67 dBm
9   SSID :    BSSID nn:  56:46:ae:4c:98:cd RSSI: -92 dBm
10   SSID :SK_WiFiGIGA98CA_2.4G     BSSID nn:
50:46:ae:4c:98:cd  RSSI: -92 dBm
11   SSID :    BSSID nn:  12:23:aa:4b:18:6a RSSI: -94 dBm
12   SSID :olleh_WiFi_DF2E     BSSID nn:  08:5d:dd:8b:df:32
RSSI: -97 dBm
13   SSID :thelove     BSSID nn:  90:9f:33:c2:48:62  RSSI: -94
dBm
14   SSID :U+NetF0FD    BSSID nn:  88:3c:1c:98:f0:fc   RSSI:
-76 dBm
15   SSID :KT_GiGA_2G_5021     BSSID nn:  88:3c:1c:17:50:24
RSSI: -91 dBm
16   SSID :KT_GiGA_7B45     BSSID nn:  1c:ec:72:1d:7b:48
RSSI: -73 dBm
17   SSID :U+Net8DA6     BSSID nn:  b4:a9:4f:6b:8d:a5  RSSI:
-96 dBm
18   SSID :SK_WiFiGIGA186B     BSSID nn:  00:23:aa:4b:18:6a
RSSI: -94 dBm
19   SSID :SK_WiFiGIGA6CD8_2.4G     BSSID nn:
42:09:a5:02:6c:da  RSSI: -72 dBm
20   SSID :iptime [illegible]:28  RSSI: -68
dBm
```

The scan value is automatically stored in the database at that location

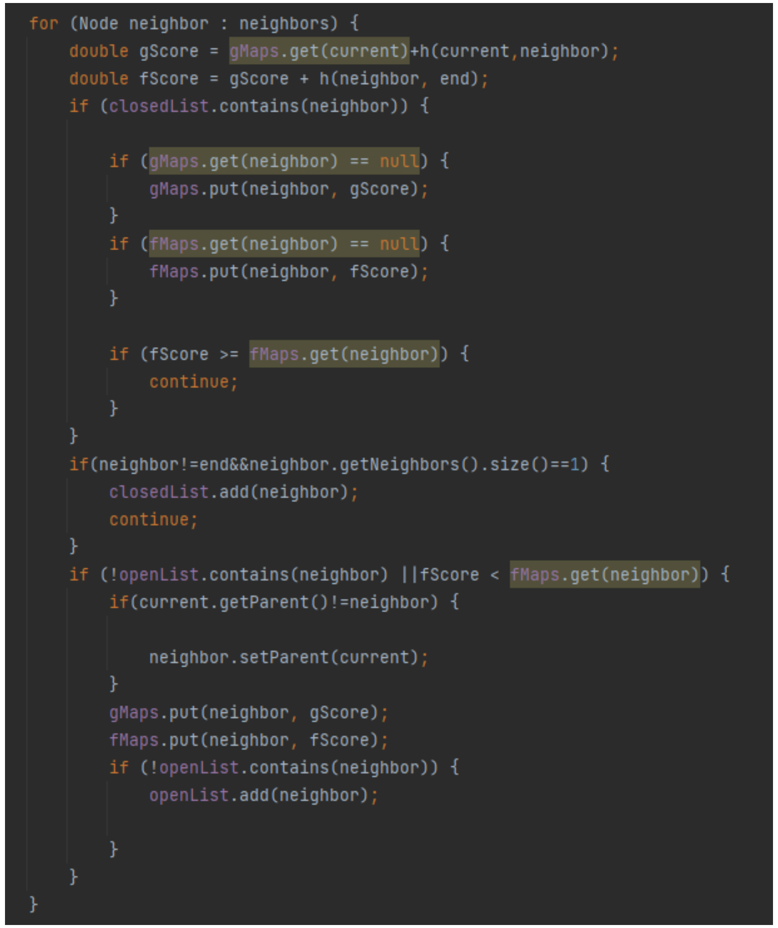
2. User app



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	426번	0	16.3	90.3						4층			526번	0	16.5	89.9
2	427번	1	17.2	87.3									527번	1	17.8	85.4
3	428번	2	18	84.1									528번	2	19.1	80.2
4	429번	3	19	80.8									529번	3	20.4	75.9
5	430번	4	19.9	77.7									530번	4	21.9	71.4
6	431번	5	20.9	74.4									531번	5	23.2	66.5
7	432번	6	21.9	71.2									532번	6	24.5	62.1
8	433번	7	22.7	68.1									501번	7	26.7	54.2
9	434번	8	23.6	65									502번	8	27.9	49.2
10	435번	9	24.6	61.7									503번	9	29.9	42.8
11	401번	10	26.4	54.9									504번	10	32.7	33.3
12	402번	11	27.8	50.2									505번	11	35.4	23.9
13	403번	12	29.9	42.8									506번	12	36.9	16.5
14	404번	13	32.7	33.3									507A번	13	39	5
15	405번	14	35.4	23.9									507번	14	32.2	5
16	406번	15	36.9	16.5									508번	15	25.6	5
17	407A번	16	39	5									509번	16	19.1	5
18	407번	17	32.2	5									510번	17	12.5	5
19	408번	18	25.6	5									511번	18	4.7	5
20	409번	19	19.1	5									512번	19	3.4	18.9
21	410번	20	12.5	5									513번	20	3.4	26.8
22	11번	21	4.7	5									514번	21	3.4	31.8
23	2번	22	3.4	18.9									515번	22	3.4	36.6
24	3번	23	3.4	28.9									516번	23	3.4	41.6
25	번	24	3.4	38.8									517번	24	3.4	45.5

Organize coordinates for each location into an Excel file

2. User app



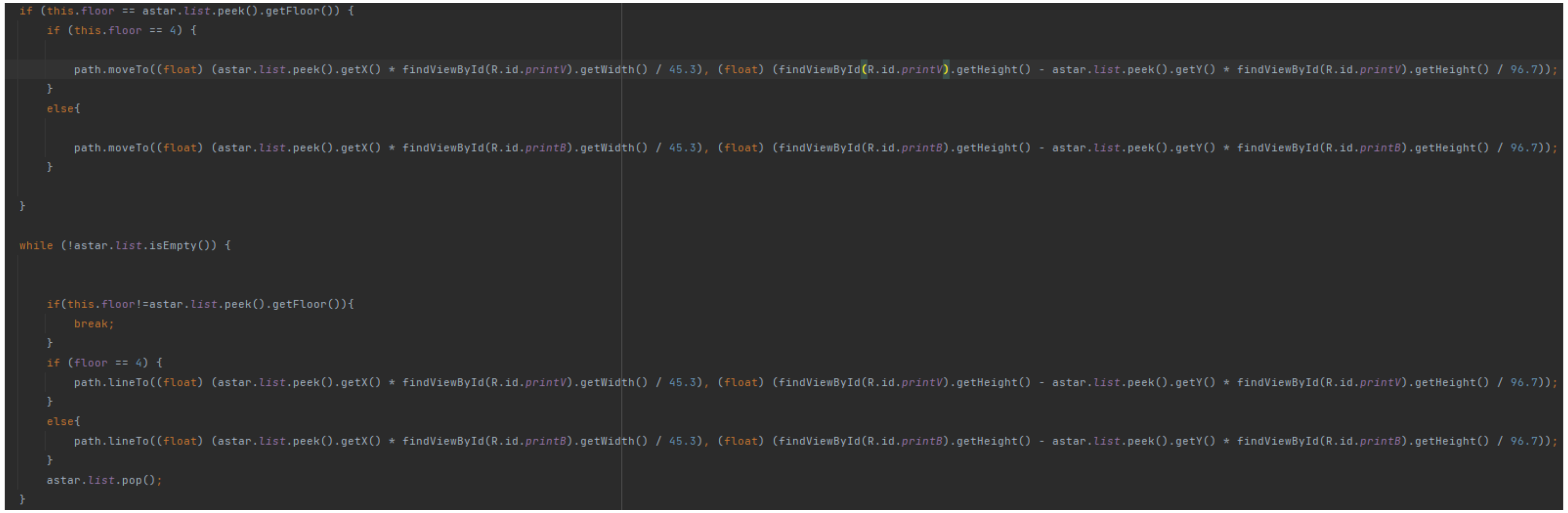
```java
for (Node neighbor : neighbors) {
    double gScore = gMaps.get(current)+h(current,neighbor);
    double fScore = gScore + h(neighbor, end);
    if (closedList.contains(neighbor)) {

        if (gMaps.get(neighbor) == null) {
            gMaps.put(neighbor, gScore);
        }
        if (fMaps.get(neighbor) == null) {
            fMaps.put(neighbor, fScore);
        }

        if (fScore >= fMaps.get(neighbor)) {
            continue;
        }
    }
    if(neighbor!=end&&neighbor.getNeighbors().size()==1) {
        closedList.add(neighbor);
        continue;
    }
    if (!openList.contains(neighbor) ||fScore < fMaps.get(neighbor)) {
        if(current.getParent()!=neighbor) {

            neighbor.setParent(current);
        }
        gMaps.put(neighbor, gScore);
        fMaps.put(neighbor, fScore);
        if (!openList.contains(neighbor)) {
            openList.add(neighbor);

        }
    }
}
```

Once the start and end points are entered, find the **shortest path** by using A* algorithm

Part 3

Progress

2. User app



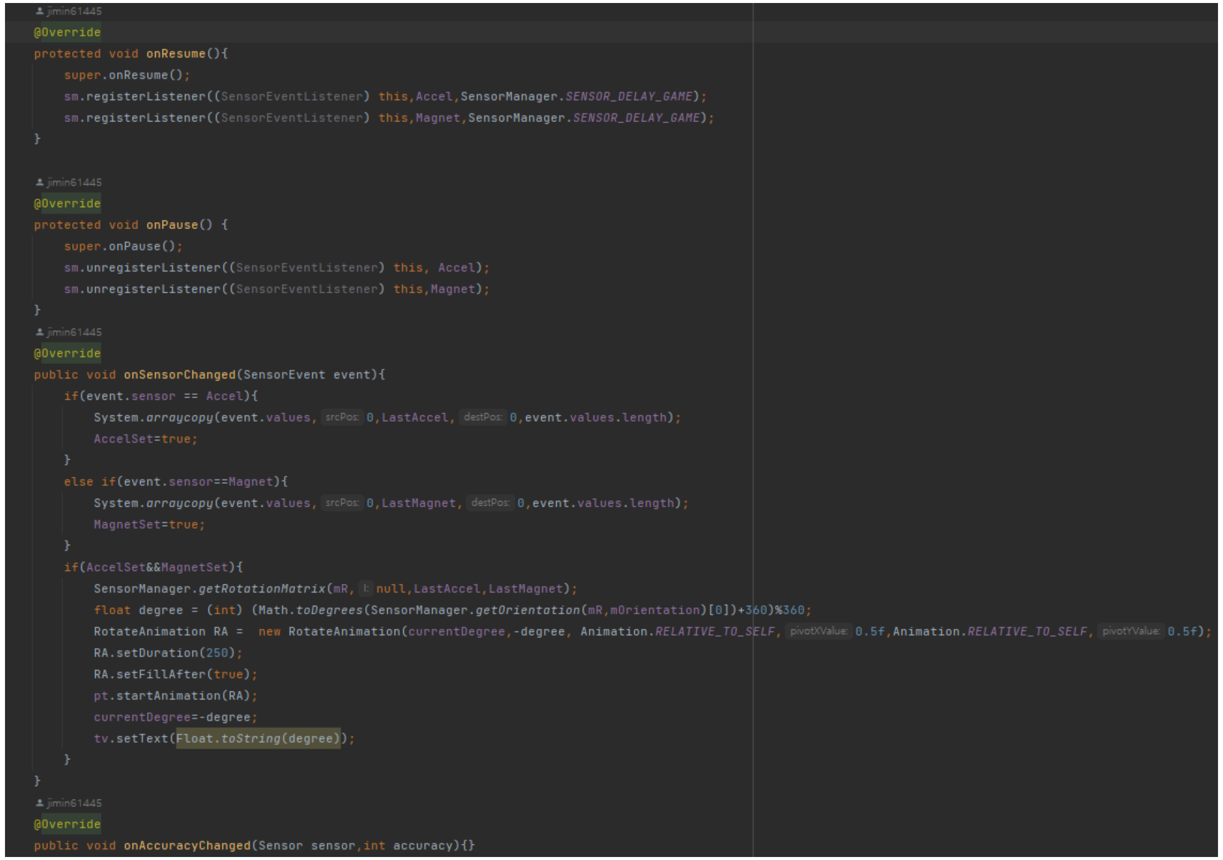
```
if (this.floor == astar.list.peek().getFloor()) {
    if (this.floor == 4) {

        path.moveTo((float) (astar.list.peek().getX() * findViewById(R.id.printV).getWidth() / 45.3), (float) (findViewById(R.id.printV).getHeight() - astar.list.peek().getY() * findViewById(R.id.printV).getHeight() / 96.7));
    }
    else{

        path.moveTo((float) (astar.list.peek().getX() * findViewById(R.id.printB).getWidth() / 45.3), (float) (findViewById(R.id.printB).getHeight() - astar.list.peek().getY() * findViewById(R.id.printB).getHeight() / 96.7));
    }

}

while (!astar.list.isEmpty()) {


    if(this.floor!=astar.list.peek().getFloor()){
        break;
    }
    if (floor == 4) {
        path.lineTo((float) (astar.list.peek().getX() * findViewById(R.id.printV).getWidth() / 45.3), (float) (findViewById(R.id.printV).getHeight() - astar.list.peek().getY() * findViewById(R.id.printV).getHeight() / 96.7));
    }
    else{
        path.lineTo((float) (astar.list.peek().getX() * findViewById(R.id.printB).getWidth() / 45.3), (float) (findViewById(R.id.printB).getHeight() - astar.list.peek().getY() * findViewById(R.id.printB).getHeight() / 96.7));
    }
    astar.list.pop();
}
```

Use canvas "moveTo" and "lineTo" to draw a path on the map

Part 3

Progress

2. User app



```
jimin61445
@Override
protected void onResume(){
    super.onResume();
    sm.registerListener((SensorEventListener) this,Accel,SensorManager.SENSOR_DELAY_GAME);
    sm.registerListener((SensorEventListener) this,Magnet,SensorManager.SENSOR_DELAY_GAME);
}

jimin61445
@Override
protected void onPause() {
    super.onPause();
    sm.unregisterListener((SensorEventListener) this, Accel);
    sm.unregisterListener((SensorEventListener) this,Magnet);
}
jimin61445
@Override
public void onSensorChanged(SensorEvent event){
    if(event.sensor == Accel){
        System.arraycopy(event.values, srcPos: 0,LastAccel, destPos: 0,event.values.length);
        AccelSet=true;
    }
    else if(event.sensor==Magnet){
        System.arraycopy(event.values, srcPos: 0,LastMagnet, destPos: 0,event.values.length);
        MagnetSet=true;
    }
    if(AccelSet&&MagnetSet){
        SensorManager.getRotationMatrix(mR, I: null,LastAccel,LastMagnet);
        float degree = (int) (Math.toDegrees(SensorManager.getOrientation(mR,mOrientation)[0])+360)%360;
        RotateAnimation RA =  new RotateAnimation(currentDegree,-degree, Animation.RELATIVE_TO_SELF, pivotXValue: 0.5f,Animation.RELATIVE_TO_SELF, pivotYValue: 0.5f);
        RA.setDuration(250);
        RA.setFillAfter(true);
        pt.startAnimation(RA);
        currentDegree=-degree;
        tv.setText(Float.toString(degree));
    }
}
jimin61445
@Override
public void onAccuracyChanged(Sensor sensor,int accuracy){}
```

Compass implementation using the ACCELEROMETER, MAGNETIC FIELD sensor

Part 4

Remaining task

Modifying UI

Add 5th floor information

Compass Enhancements

Find Current Location

THANK YOU