2. Data analysis (데이터 분석)

과제명 (Project)	2.1 WiFi based people counting (이지은)
제 목 (Title)	2.1.1 WiFi based people counting survey (김시영)
Continued from page	

2.1.1.1 Intro (강민희 · 김시영)

✓ Intro (강민희)

Mobile devices regularly broadcast Wi-Fi probe requests in order to discover available proximal Wi-Fi access points for connection.
A probe requests, sent automatically in the active scanning mode, consisting of useful information such as the MAC address of the device, received signal strength indication (RSSI) and so on. A "SPOT" is the a real-time wireless sniffing system which is able to sense Wi-Fi probe requests.
Each mobile device has on unique MAC address. This provides an opportunity to obtain the mobile devices and the environment.
The basic assumption is that, with a high probability an adult person in our world will carry a smartphone on their body.
For somebody not carrying, there is another one carrying ~~two~~ two.
Therefore, we can equate a mobile phone with a visitor

✓ Use cases (김시영)

- Indoor: Every business with a physical space should count customer traffic to see the bigger picture of what is going on in their business. Whether you are a shopping center, retail chain, museum, library, sporting venue, bank, hospital or other. people counting data will help you make well-informed decisions about your business.

- Disaster : People Counting technology is used in disaster-safe area, allowing firefighters to check where a peron is in a building when a disaster such as a fire occurs.

Continued to page

기록자(Recorded by)	작성일(Date)	서 명(Signature)
김 시영	2020. 06. 19	
점검자(Witnessed by) 이 지은	점검일(Date) 2020. 06. 19	서 명(Signature)

2. Data analysis (데이터 분석)

Page 2

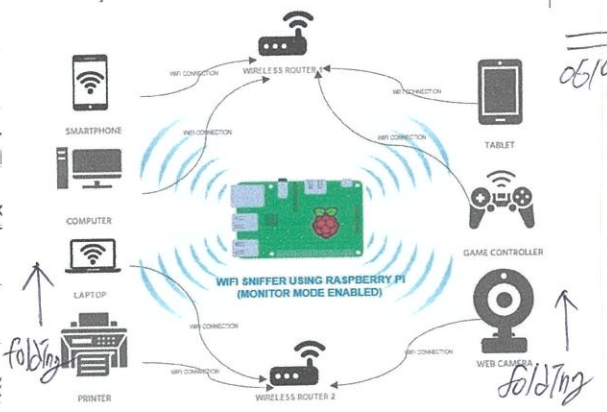
과제명 (Project)	2.1 WiFi based people counting (이진은)
제 목 (Title)	2.1.2 Benefits and drawbacks of the technique (장단점)
Continued from page	2.1.2.1 Advantages (비자수)

Most existing people counting systems are sensor-based and can only work in some fixed gateways or checkpoints where sensors installed.
This high dependence on the exact locations of sensor leads to low accuracy
"Counting people without counting people!"

Every smartphone has Wi-Fi and sends out Wi-Fi signals
↳ send out probe requests to communicate with a Wi-Fi access point.
"Wi-Fi based people counting technique" does not require any action from people

Advantages of Wi-Fi based people counting technique over sensors and camera

1) Very cheap devices are possible.
2) Little instalment costs.
3) Zero daily effort for cleaning or mai
4) No matter how many doors / entrances
5) By linking several devices, department and controlled separately.
6) All data is anonymous (most MACs are
7) Independent from outside conditions like humidity and temperature
8) Additional, anonymous data about movements inside the location



06/19

Continued to page

기록자(Recorded by) 김 주 연	작성일(Date) 2020. 06. 19	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 06. 19	서 명(Signature)

3. Monitoring Platform (플랫폼)

과제명 (Project)	3.1 Visualing Collected Information (김상경)
제 목 (Title)	3.1.1 Visualing using Dashboard (홍의명)
Continued from page	

(박지영)

3.1.1.1 Determine dashboard objectives (강혜영)

✓ Abstract

Improving the quality and safety performance of Smart cities have become Increasingly important In recent years. Researchers and developers creating different kinds of sensors using high-end technologies to tackle Smart City problems like safety. In fact these sensors collect huge amount of raw data which is overwhelming to analyze with bare eyes. So, to solve this problem, we need user-friendly secure dashboard that delivers most important real time data fast and In a secure way. Our main goal is to develop a dashboard to provide emergency services with exact numbers of people In the buildings who are facing a danger. However, there are a number of elements should be met before these dashboards are valid to use as Emergency Management Dashboard. Research shows that thes Dash-board should :

1) provide content that aligns with the needs of emergency services
2) be designed In such a way that the contents is easily reliable.
3) display timely, complete and correct data to be perceived as valid and reliable by users.

✓ Operational dashboards : aim to impact critical Information quickly to users as they are engaged In time sensitive tasks.

✓ Analytical dashboards : provide the user with at a glance Information used for analysis and decision making, but don't have the same level of the time sensitivity as operational dashboards.

Continued to page

기록자(Recorded by) 김 경	작성일(Date) 2020. 06. 19	서 명(Signature)
점검자(Witnessed by) 이 재 형	점검일(Date) 2020. 06. 19	서 명(Signature)

과제명 (Project)	2.1 Wi-Fi based people canting (이라는)
제 목 (Title)	2.1.2 Benefits and drawbacks of technics (장단점)
Continued from page	2.1.2.2 Drawbacks (방식의)

✓ Drawbacks (possibe solutions)

The people counter based on smartphones will work but not not accurately or comfortable.
There are a lot of things to consider.

1) Signals outside my area might be received. : Wi-Fi probe request sniffers usually can sniff around 100 meter (300 ours), that implies smartphones out of our Interest area can be captured.

2) The people canter has a more or less bowl-shaped reception area but most locations have corners Instead : As a result people In the corners can be missed or in contrast, people out of the Interest area can be captured

3) The visit was too short to receive a ping from a certain Phone : Devices tend to send more Probe requests when they are active. on the other hand, send less when they remain untouched

4) There are up to 14 Wi-Fi channels - the ping could be on any of them (even all) : Devices send probe request on different Wi-Fi channels to find access point sooner which can cause that Wi-Fi sniffers to miss the probe.

Continued to page

기록자(Recorded by) 정 희 연	작성일(Date) 2020. 06. 26	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 06. 26	서 명(Signature)

과제명 (Project)	3.1 Visualizing Collected Information (김성호)
제 목 (Title)	3.1.2 UI/UX Design Planning (홍이영)
Continued from page	

3.1.2.1 Designing Layouts (홍이영) (김지수)

✓ Designing Layouts.
Creating a dashboard that clearly comunicates your key goals and metrics is not an easy task. It requires the ability to combine a smooth user experience with appealing UI to make important operational data easy to read and perceive.

✓ 'stick to the five-second rule': It should take no more than five seconds for user to find the most important information on the dashboard.

✓ 'Take care of creating a clear and logical layout.': Divide all the information into three parts in descending order of importance.

✓ 'Display only key metrics': not overwhelm the user with too many details.

✓ 'visualize data in an appropriate way': we should organize all the information to make it easy to understand.

✓ 'chose the right fonts' for both app or web UI and UX design, the proper font is of the utmost importance. When creating a dashboard, we no more than two fonts styles to make your UI dashboard. design more legible

✓ 'Pick suitable colors' A useful dashboard must be designed with colors in mind and any potential confusion must be avoided.

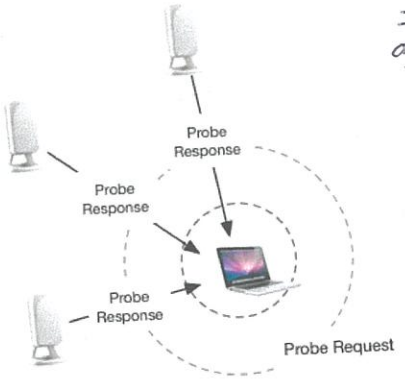
Continued to page

기록자(Recorded by) 홍이영	작성일(Date) 2020. 06. 26	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020 06. 26	서 명(Signature)

과제명 (Project)	2.2 Research Wi-Fi data characteristics (이자은)
제 목 (Title)	2.2.1 Feature extraction from Wi-Fi probes (강지영)
Continued from page	

2.2.1.1 Exploring Wi-Fi probe request (강민채)

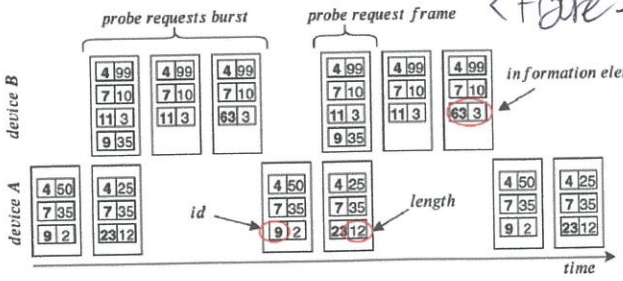
Wi-Fi devices searches for wireless network by sending messages called probe request (shown in figure.A) They send a burst of these messages with associated a time limit within which it must receive reply to connect to the network. Every Access Points (APs) that receive these frames replies to the device by sending a probe Response Frame with the information necessary to establish the connection.



<Figure 1>

The Probe Request and the probe Response are two sub-types of a particular frame called Management Frame which is divided into MAC header and frame body. Fields within the frame body could have fixed lengths, called fixed fields, or variable lengths called Information Elements (IEs) All IEs are labeled with an identification number and it's size. The first octet of the Information Element is reserved for the Element ID, the second defines the whole information element's length as shown in Figure 2.

<Figure 2>



The remaining bits contain the information. Accordigly, each IE conveyed in the probe request frame is identified by it's ID and it's length

Continued to page 7페이지 계속

기록자(Recorded by) 강 조 영	작성일(Date) 2020. 07. 10	서 명(Signature)
점검자(Witnessed by) 이 재 희	점검일(Date) 2020. 07. 10	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	李 6.페이지 에서.

which Indicates the number of octets used by the IE content.

Each Wi-Fi device has a MAC address that uniquely identifies itself in the local network.

Fig A) shows the structure of the MAC address,
which is composed of six octets of bits. The first three octects are assigned by the IEEE to the device manufacturer and constitute the Organization Unique Identifier (OUI).
The remaining three octets are called Network Interface Controller (NIC) and are assigned by the manufacturer.
An interesting part concerns the second least bit in the first octet highlighted in red in the fig.A

<fig.A>

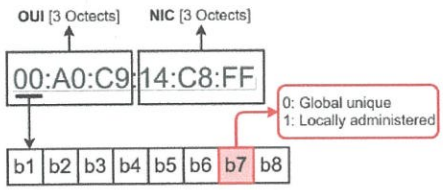


Figure 1: Global unique and Locally administered bit detail of a MAC address.

0이어야

In Wi-Fi applications if this is set to 0, then the MAC address should be globally unique and it is kept constant over the time Otherwise, when this bit is set to 1, the MAC address should be randomly generated and may change from one session to another.

Continued to page

기록자(Recorded by) 김 [illegible]	작성일(Date) 2020. 07. 10	서 명(Signature)
점검자(Witnessed by) 이 [illegible]	점검일(Date) 2020. 07. 10	서 명(Signature)

3. Monitoring Platform. (플랫폼).

과제명 (Project)	3.1 visualizing Collected Information (김성조)
제 목 (Title)	3.1.3 charts, Graphs Visualization Planning. (박경원)
Continued from page	

3.1.3.1 charting Libraries. (김경수)

There are numerous JavaScript libraries out there, each with their specific pros and cons as with any tools. We picked several open source charting libraries and will look at their pros and cons and ~~desi~~ decide which one to use.

1) ChartJS 2) ApexChart 3) D3JS

1) Let's start with 'ChartJS'. this is an open source JavaScript library supporting 8 chart types. It is a small js library at just 60 kb. Types include line charts, bar charts, radar, pie chart, bubble, scatter plots, and mixed. A time series is also supported. Chart.Js is an open source library and free to use for personal and commercial use which is a plus. The limited number of types can be an issue for more advanced dashboard ~~the~~ requirements.

2) Apexcharts is one of the most comprehensive and popular JavaScript charting libraries base on HTML5, rendering in SVG/VML. It is lightweight. supports a wide range of diverse chart types and ensures high performance. It can export chart to SVG PNG or CSV formats. ApexChart scales gracefully in desktops, tablets as well as mobiles, You can set responsive queries and can have different layouts for different screen sizes. Zoom, pan, scroll through data, toggle datasets visibility in multiple series, show information ~~tool~~ tool tips. When user hovers over datpoints. Each of these options helps to communicate data more effectively.

3) D3JS is a #powerful open-source JavaScriot library for data visualization Basically, D3 is more like a framework than a library. It may well be not that simple to work with, which can look quite critical at the beginning. but there are a lot of helpful information resource available out there. D3 is an open surce which is free all kinds of use.

Continued to page

기록자(Recorded by) 김 경 수	작성일(Date) 2020. 07. 10	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 07. 10	서 명(Signature)

2. Data analysis (데이터 분석)

과제명 (Project)	2.2 Research Wi-Fi data charitеristics (이자은)
제 목 (Title)	2.2.1 Feature extraction from WIFI probes. (김지원)
Continued from page	

2.2.1.2 Feature extraction (김지원)

The probe request send snippets of information such as a unique MAC address (Wi-Fi) device MAC address, RSSI (Received signal strength Indication) and a list of previous SSIDs encountered (most SSIDs are hidden) Following data is just piece of the sniffed Wi-Fi probe request collecting:

```
1 spot_mac, push_time, device_mac, device_time, rssi, signal_type
2 d86595062780,2020-07-03 10:43:15,d013fd3d15d1,2020-07-03 10:36:17,-48,WIFI
3 d86595062780,2020-07-03 10:43:15,24dbedfa4040,2020-07-03 10:36:19,-38,WIFI
4 d86595062780,2020-07-03 10:43:15,22e722d066d1,2020-07-03 10:36:36,-36,WIFI
5 d86595062780,2020-07-03 10:43:15,859d36fe473e,2020-07-03 10:37:05,-50,WIFI
```

✓ spot_mac : Wi-Fi sniffer mac address
(Our wi-fi sniffer also has wi-Fi ap mode)
✓ push_time : The time when probe request is collected and sent to the sever
✓ device_mac : Mac address of the daice, which the probe request is sent
✓ device_time : The time probe request sent off from device.
✓ rssi : Received signal strength Indication of wi-Fi sniffer.
✓ signal_type : probe request type, It can be BLE or wi-Fi

Device Mac address and RSSI are the core information needed for people counting. Device Mac address is unique.

Continued to page

기록자(Recorded by) 조정명	작성일(Date) 2020. 07. 14	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020. 07. 14	서 명(Signature)

3. Monitoring Platform (플랫폼)

과제명 (Project)	3.1 visualizing Collected information (정보화)
제 목 (Title)	3.1.3 charts, Graphs, visualization planning (박진영)
Continued from page	

3.1.3.1 Identifying suitable Chart Types. (강석)

There are a lot of chart types are available, each with different use cases. Usually, it is hard to figure out which type of chart to use while visualizing given data and developers usually try to add charts as many as possible and they try to use very complex charts which usually leads to poor user experience Choice of chart type usually depend on several factors.

- ✓ what are the type of metrics, features, or other variables, plan on plotting.
- ✓ who is the audience that plan on presenting to.
- ✓ what is the kind of conclusion that user to draw.

✓ propose method.

since, we are developing operational dashboard for emergency situations in safety management platform. the charts must be simple yet. It should include detailed information about building population like how many people are in a building, how many people in each floor and so on.

✓ The Proposed Charts

- ✓ Indicator to show how many people remain in an emergency situation.
- ✓ Pie chart to show how many percentage of people escaped and how many still remaining
- ✓ Grouped bar chart to show how many people in each floor escaped.
- ✓ Line Chart to check daily people count.

Continued to page

기록자(Recorded by) 김지영	작성일(Date) 2020. 07. 17	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020. 07. 17	서 명(Signature)

과제명 (Project)	2.2 Research Wi-Fi data characteristic (이지은)
제 목 (Title)	2.2.1 Feature extraction from Wi-Fi probes (김지영)
Continued from page	2.2.1.3 Mac address randomization (배지수)

✓ Mac address randomization. (1)

Mac address are identifiers of radio chips that is unique worldwide address. The chip uses the MAC address for network communications which, In wireless, are sent over the air for to see.

Randomization of Mac address is the process of generating random Mac address by devices during the active scanning for access point. Mac address is randomized to prevent tracking issues. when the device is connected to the AP the device uses It's real mac.

But, latest OS systems still use randomized, but fixed mac address There are no standard algorithms for this process, and each OS implements Its custom randomization

Apple Introduced the MAC address radomization In IOS since ver. 8 Based on experiments with various models, iOS devices randomize the MAC address for each burst of probe requests. In iOS 14 for each unique wireless network, the device will choose a new randomized address and use that private address for the network and this address was also randomized every 24 hours.

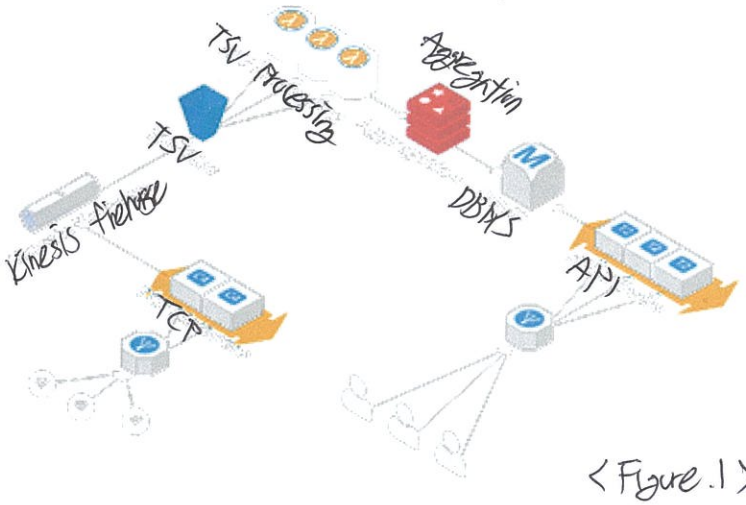
Continued to page

기록자(Recorded by) 조 주 영	작성일(Date) 2020. 07. 24	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 07. 24	서 명(Signature)

과제명 (Project)	3.2 Gathering and processing data (데이터 수집)
제 목 (Title)	3.2.1 Accessing raw data (원시 데이터)
Continued from page	3.2.1.1 Collecting spot Data. (실내 위치 정보)

✓ propose

Spot Data is the collection of mobile device MAC address, RSSI, and collected time information collected from Access point routers and this data is processed in lambda (Code Snippet 1) before storing to the cloud Database (Figure 1) Spot data is stored in Mina database and we need several data tables to fresh Spot data : 1. Buildings 2. Collectors (WiFi Probe request receivers) 3. collector_population (raw collector data are storage)



<Figure.1>

✓ Result

First we need to check if building ID is in registered building list 面 if yes we use this building ID to fetch collectors mac addresses. then we can use this collector mac addreses to fetch spot raw data from collector population table (Code snippet 1) and after successfully fetching data we can proceed to filtering and processing.

13 Continued to page

기록자(Recorded by) 정정명	작성일(Date) 2020. 07. 20	서 명(Signature)
점검자(Witnessed by) 이재현	점검일(Date) 2020. 07. 20	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	12 페이지 연결

```python
def get_nipa_connection():
    return pymysql.connect(
        host='host',
        port=port,
        user='user',
        password='password',
        db='nipa',
        cursorclass=pymysql.cursors.DictCursor
    )
def get_spot_raw_data(request):
    date_str = request.GET.get('date_1', None)
    date = datetime.strptime(date_str, "%Y-%m-%d")
    tomorrow_date = date + timedelta(hours=24)
    query = f"SELECT count, collector_mac, updated_at FROM collector_population_detail /
            WHERE collector_mac in {col_macs} AND counted_at = '{latest_counted_at}'"
    read_gate = get_gate_connection()
    with read_gate.cursor() as cursor:
        cursor.execute(query)
        gate_datas = cursor.fetchall()
```

0220

<code snippet # 1>

Continued to page

기록자(Recorded by) 문종명	작성일(Date) 2020. 01. 20	서 명(Signature)
점검자(Witnessed by) 이재현	점검일(Date) 2020. 01. 20	서 명(Signature)

과제명 (Project)	2.0 Research WiFi data characteristics (이지은)
제 목 (Title)	2.2.1 Feature extraction from WiFi probes (김지영)
Continued from page	

2.2.1.4 Mac address randomization (이) and RSSI (박수진)

✓ Mac address randomization

Apple introduced the MAC address randomization in iOS since version 8. Based on experiments with various models, iOS devices randomization the MAC address for each burst of probes requests, In iOS 14 for each unique wireless network, the device will choose a new randomized address and use that private address for the network and this address was also randomized every 24 hours. (page 11).

Android devices, starting from Android 8.0 use randomized MAC addresses when probing for new networks while not currently associated with a network. In android 9, you can enable a developer option (this disabled by default) to cause the device to use a randomized MAC address when connecting to a Wi-Fi network. In Android 10, MAC randomization is enabled by default for client mode.

✓ RSSI

RSSI (Received Signal Strenth Indicator) is an estimated mensure of power level that a client device is receiving from an access point. At larger distances, the signal gets weaker and the wireless data rates set slower leading to poor connection. RSSI can be used to estimate how far the device is located from AP

<excellent>	<Good>	<Fair>	<Poor>
±50 dBM	-50 to -60 dBM	-60 to -70 dBM	≤ -70dBM

Continued to page

기록자(Recorded by) 윤 주 영	작성일(Date) 2020. 07. 31	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 07. 31	서 명(Signature)

3. Monitoring Platform (플랫폼)

과제명 (Project)	3.2 Gathering and processing Data. (아부바크르)
제 목 (Title)	3.2.1 Accessing raw data. (홍이명)
Continued from page	

3.2.1.2 Collecting GATE Data (윤인정)

✓ Propose

GATE Data is the collected ~~the~~ from GATE sensors installed at the top the doors. GATE sensors calculate entrances and exits for 30 seconds and sends it to server sever stores data in GATE Database. In order to fetch gate information when need several tables and attributes.

1. Buildings 2. GATE_sensors 3. GATE_data.

✓ Result

First we need to check if bilding id is in registered building list if yes we use this building id to fetch GATE Mac addresses. Then we can use this GATE sensor mac addresses to fetch gate raw data from gate data table and after successfully fetching data we can proceed to filtering and processing (code snipped 2)

```
def get_gate_data(self, mac_place, current_date):
        gate_macs = tuple(mac_place.keys())
        if len(gate_macs) == 1:
            query = f"SELECT mac_address, sum(count) AS count, max(updated_at) "
                    "as updated_at FROM gate_data where mac_address='{gate_macs[0]}' and "
                    "updated_at >= '{current_date}' GROUP BY mac_address"
        else:
            query = f"SELECT mac_address, sum(count) AS count, max(updated_at) "
                    "as updated_at FROM gate_data where mac_address in {gate_macs} "
                    " and updated_at >= '{current_date}' GROUP BY mac_address"

        read_gate = get_gate_connection()
        try:
            with read_gate.cursor() as gate_cursor:
                gate_cursor.execute(query)
                gate_data = gate_cursor.fetchall()
        finally:
            read_gate.close()
```

<code snippet 2>

GATE → SERVER → DATA BASE → DATA BASE ACCESS → DEVELOPER

Continued to page

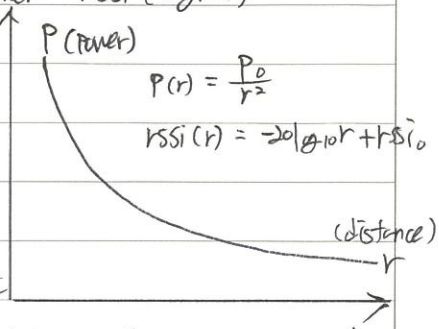
기록자(Recorded by) 윤인정	작성일(Date) 2020. 01. 31	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020. 01. 31	서 명(Signature)

과제명 (Project)	2.3 WiFi data processing (이지은)
제 목 (Title)	2.3.1 Algorithm ~~survey~~ development (강지영)
Continued from page	2.3.1.1 Algorithm survey (강 민제)

✓ RSSI experiment

As shown below Figure1, the larger the distance, the weaker rssi (signal) The signal strenth depends on distance and boardcasting power value. At maximum board casting power (+4.2Bm) the rssi range from -26 (a few inches) to -100 (50~60 meters).

the rssi can be used to approximate distance between the device and wi-fi sniffer. We conducted an experiment to see the behavior of RSSI at different ranges.



(Fig 1) The Inverse square law

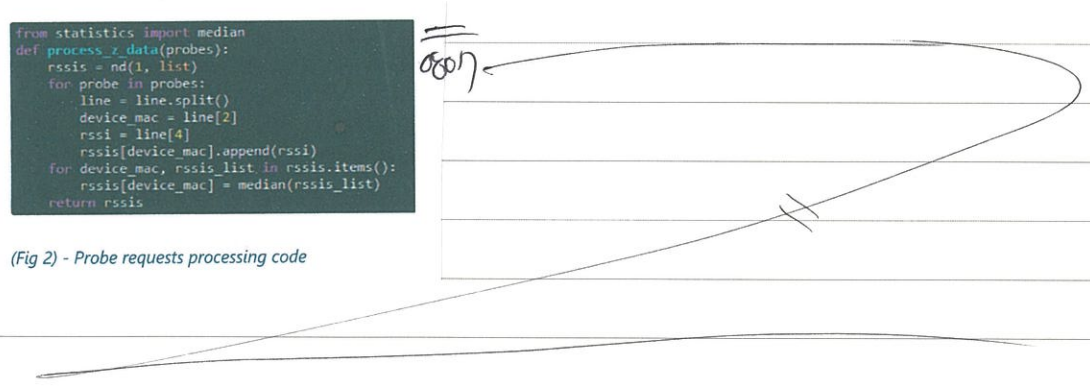
✓ Experiment settings

Devices : Galaxy s6, s9, A5 / iPhone 8s, 10s / LG G6, G8

Above devices were placed at every 5 meters of 30 meters. During the experiment devices were connected to wi-fi to distinguish the probe requests by their MAC addresses in the below python code, RSSIs are grouped by device MAC addresses and median of each group taken as a final result. Finally, it is plotted as shown below the code.

```
from statistics import median
def process_z_data(probes):
    rssis = nd(1, list)
    for probe in probes:
        line = line.split()
        device_mac = line[2]
        rssi = line[4]
        rssis[device_mac].append(rssi)
    for device_mac, rssis_list in rssis.items():
        rssis[device_mac] = median(rssis_list)
    return rssis
```

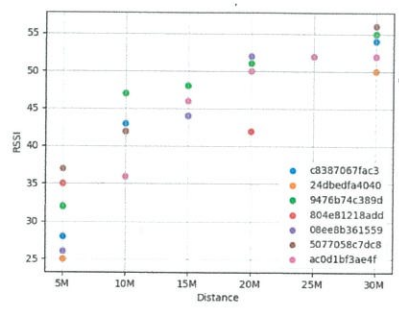
(Fig 2) - Probe requests processing code



Continued to page

기록자(Recorded by) 강 민 제	작성일(Date) 2020. 08. 01	서 명(Signature)
점검자(Witnessed by) 이 지 은	점검일(Date) 2020. 08. 01	서 명(Signature)

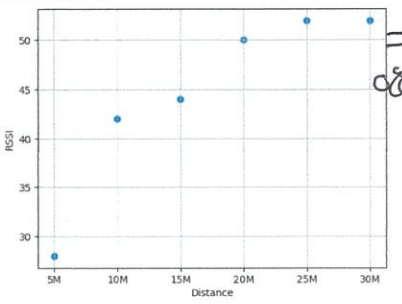
과제명 (Project)	
제 목 (Title)	
Continued from page	16페이지 연결



(Fig 3) - RSSIs VS Distance

Devices mac are used for labeling for convenience. As it can be seen that, as the distance increases the RSSI is decreased. Moreover the analyses shwed that, RSSI is highly dependent on the phone model, too

We took the average of all devices RSSIs median and depicted as shown in Fig. 4 below



(Fig 4) - RSSIs median VS Distance

Experiment result showed that, it is easir to distinguish devices located at shorter distance compared to longer distance. thus, the best practice would be portioning wi-fi sniffers to cover 20 meters of diameters.

Continued to page

기록자(Recorded by)	작성일(Date)	서 명(Signature)
조 종 명	2020. 08. 07	
점검자(Witnessed by) 이 재 희	점검일(Date) 2020 08. 07	서 명(Signature)

과제명 (Project)	2.3 Wifi data processing (이지은)
제 목 (Title)	2.3.1 Algorithm development (김지영)
Continued from page	

2.3.1.2 Wi-Fi probe request interval (김지영)

✓ Wi-Fi probe request interval

According to some research papers, devices can send from 1 to 100 probe requests with random MAC address per second. We wanted to experiment and verify wether it is correct or not.

It is difficult to group probes when devices are randomizing their MAC address. But, to overcome this problem we placed one device for each probe request sniffer right beside them. This way, we can filter out probe request sent by other devices by filtering probes by their RSSIs.

✓ Experiments setup

Devices : Galaxy S20, S8, S10, S9, A5, J5, Q8, Note3, 4 / iPhone 8, 11 / LG G7, G8, V50, V30

phones and Wi-Fi sniffers are placed right next to each other as shown below

Positioning wi-fi probe request sniffer and device

0814

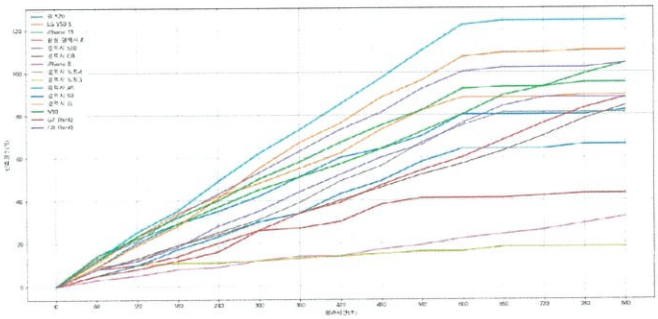
14페이지 이동 Continued to page

기록자(Recorded by) 김지영	작성일(Date) 2020.08.14	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020.08.14	서 명(Signature)

과제명 (Project)	
제 목 (Title)	

Continued from page 18페이지 연결.

When devices is located close to the sniffer as shown above, we can filter out other probe requests using rssi.
After experimenting with all above devices, we plotted all result as time vs probe request frequency as shown in Figure 5.



각 디바이스별 신호 개수 / 경과 시간 그림

<Fig. 5>

As it can be seen in above Figure 5 that, devices and average of 10 probe requests per minute.

Continued to page

기록자(Recorded by)	작성일(Date)	서 명(Signature)
점검자(Witnessed by)	점검일(Date)	서 명(Signature)

기록자: 김건영 — 작성일: 2020. 08. 14
점검자: 이재현 — 점검일: 2020. 08 14

과제명 (Project)	3.0 Gathering and Processing Data (데이터 수집)
제 목 (Title)	3.2.2 Processing raw Data (박진영)
Continued from page	

3.2.2 Filtering and Processing ~~Gate Data~~ GATE/SPOT data (김경수)

✓ propose

We need to create structred databases datasets using Spot Raw Data so what we can use them on charts. To do so, we need to research what type of datasets our charts can handle and pick the most appropriate. After choosing data-set type we can proceed to filtering our Spot Raw data. and GATE Raw data.

✓ result

We need to create Building people count and floor people count and base on ~~this~~ these datasets we create our data-set for charts and it is going to be dictionaries in python and every dictionary includes 'x' and 'y' date-time and count of mac addresses respectively.

```python
def process_spot_data(raw_spot_data, days_list):
    data = nd(1, int)
    for row in raw_spot_data:
        data[row.counted_at] += row.count
    data = dict(sorted({**time_dict, **data.to_dict()}.items()))
    dataset = []
    for i in range(len(days_list)-1):
        temp = {'x': days_list[i], 'y': list(data.values())[i]}
        dataset.append(temp)
    return dataset
```

08/14

<spot data python code>

```python
def process_gate_data(gate_raw_data):
    gate_data = nd(1, int)
    for row in gate_raw_data:
        gate_data[date_str + ' ' + row['created_at']] += row['count']
    gate_d = gate_d.to_dict()
    g_data = {**labels, **gate_d}
    dataset = []
    for label in range(labels):
        temp = {'x': label, 'y': gate_data[i]}
        dataset.append(temp)
    return dataset
```

0814

<gate data python code>

Continued to page

기록자(Recorded by) 김경수	작성일(Date) 2020. 08. 14	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020. 08. 14	서 명(Signature)

2. Data analysis (데이터 분석)

과제명 (Project)	2.3. Wi-Fi data processing (이지은)
제 목 (Title)	2.3.1 Algorithm development (김지영)
Continued from page	

2.3.1.3 Algorithm flow chart / develop (박재우)

Counting people based on wi-Fi probe request emitted by their smartphones requires some from of filtering and processing. Following is the flowchart of our algorithm.

✓ Probe requests

Probe requests are scanned in every 2 seconds and sent to our server in every 30 seconds.

✓ Distance based filtering.

Distance-based filtering assumes a correlation of the RSSI and the distance between the sending phone an the scanner. A strong RSSI would indicate that the phone is close to the wi-Fi ~~sniffer~~ sniffer where a weak signal would mean it is futher away. clear RSSI thresholds would allow for filtering for wi-fi probes that were sent from a particular area.

Since we are interested in devices only in a particular area (20 meters circle), we would exclude any probe that is sent out of interest area. Based on result of RSSI experiment above, we decided to filter out any Probes RSSI of which is smaller than -50

Continued to page

기록자(Recorded by) 김지영	작성일(Date) 2020. 08. 25	서 명(Signature)
점검자(Witnessed by) 이지은	점검일(Date) 2020. 08. 25	서 명(Signature)

과제명 (Project)	2.3 Wi-Fi data processing (이자은)
제 목 (Title)	2.3.1 Algorithm development (김지명)
Continued from page	

2.3.1.4 Algorithm survey (박수진)

✓ Temporary filtered-data collection

Filtered data by distance-based filtering is stored and merged with upcoming data for 5 minutes. Then, the data is passed to MAC address filter

✓ MAC addes filter / normalization

according to this artide, as of July 2020 mobile vendor market shre of South korea as follow.

- Samsung 63.92% / Apple 25.69% / LG 7.15%
- Unknown 1.64% / Xiaomi 0.81% / Huawei 0.29%

collected MAC address are filtered based on their manufacture above. If the vendor of the MAC address is unknown. It is dussified as a random generated MAC address and colected separately for later normalization.

As a result, we will have ~~two~~ two group of MAC addess, namely random macs and real macs.

✓ People counting

number of randomised MAC address is normalized based on the experiment we had conducted above. Number of people is equal to the number of real macs and the number of normalized random macs.

✓ Insert Into DB.

The count of people is inserted to a Database along with It's counted time.

Continued to page

기록자(Recorded by) 김지명	작성일(Date) 2020.08.28	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020.08.28	서 명(Signature)

과제명 (Project)	3.2 Gathering and Processing Data (데이터 수집 및 처리)
제 목 (Title)	3.2.3 Charts data response (응답)
Continued from page	3.2.3.1 Selecting data response method (응답 방식)

✓ Propose

We need to feed our charts with data and we need to choose how to delivering data to charts. There are mainly two method available.

1) Django's native method sending data with template.

2) Django's JSON response method.

✓ Result

- Django's native method sending data with template

  Pros : Easy development (We only need to develop one Django view function to create whole page and sending data)

  cons : page is loaded after all data is ready (directly related to data process time)

  poor UX (in order to update data when need to reload page, it is not good approache to building dashboard)

- Django's JSON response method

  Pros : page loads instantly and request for data separately

  → Excellent UX ~~3) Hard~~

  cons : Hard development (we need to create separate Django view functions for every Json response method)

To conclude, we will pick Django's JSON response method despite being hard to develop it gives great advantages in developing front-end of the dashboard

Continued to page

기록자(Recorded by) 오 승 아	작성일(Date) 2020. 08. 29	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 08. 29	서 명(Signature)

과제명 (Project)	3.2 Gathering and processing data (아부바크르)
제 목 (Title)	3.2.3 charts Data response (홍이명)
Continued from page	3.2.3.2 Line-chart API development (김해연)

✓ Propose

Line chart is used to represents daily people count. and to develop line chart we need date attribute send from front-end which is used to process data which we are going to convert into line chart data.

After that send it via Json response to front-end

✓ Result

set code.

```
1  def get_line_data(request):
2      date = datetime.strptime(request.GET.get('date_1', None), "%Y-%m-%d")
3      collectors_data, labels = process_spot_data(date)
4      data = []
5      for label in sorted(labels):
6          label1 = label.split()[1][:5]
7          for row in collectors_data:
8              count = 0
9              if row.counted_at != label:
10                 continue
11             for i in range(len(collector_macs)):
12                 if row.collector_mac == collector_macs[i]:
13                     count = row.count
14         data.append({'x': label1, 'y': count})
15
16     return JsonResponse({'data': data, 'labels': sorted(labels)})
17
```

Continued to page

기록자(Recorded by) 홍준영	작성일(Date) 2020. 08. 28	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 08. 28	서 명(Signature)

과제명 (Project)	3.2 Gathering and processing data (아부바크르)
제 목 (Title)	3.2.3 charts Data response (홍미영)
Continued from page	

3.2.3.3 Bar chart API development (강건수)

✓ Propose

Bar chart is used to display how many people in every floor and to develop Bar chart we need date attribute send from front-end which is used to process data which we are going to convert into bar chart data.

After that send it via Json response to front-end

✓ Result

set code.

```
1  def get_bar_chart(request):
2      time_dict = {}
3      now = datetime.now()
4      floors = get_floors(request.building_id)
5      for i in range(1,4):
6          time_dict[str(round_mnt(now-timedelta(days=i)))[:16]+':00'] = 0
7          print(datetime.now())
8
9      raw_data = process_gate_data(time_list)
10     data = nd(1, int)
11     for i in raw_data:
12         data[i.counted_at] += i.count
13     data = dict(sorted({**time_dict, **data.to_dict()}.items()))
14     dataset = []
15     for i in range(len(floors)-1):
16         temp = {'x': floors[i], 'y': list(data.values())[i]}
17         dataset.append(temp)
18     dataset.append({'x': floors[-1], 'y': x})
19
20     return JsonResponse(dataset, safe=False)
21
```

Continued to page

기록자(Recorded by) 오조영	작성일(Date) 2020. 09. 04	서 명(Signature)
점검자(Witnessed by) 이제현	점검일(Date) 2020. 09. 04	서 명(Signature)

과제명 (Project)	3.2 Gathering and process data (아부바크르)
제 목 (Title)	3.2.3 charts data response (홍미영)
Continued from page	

3.2.3.4 Table data API development (김혜영)

✓ propose

To develop Table chart we need date attribute send from front-end which is used to process data which we are going to convert into bar chart data.

After that send it via Json response to front-end.

```
def get_table_data(request):
    raw_data = []
    now = datetime.now()
    t = str(round_mnt(now-timedelta(minutes=10)))[:16]+':00'
    for mac in collector_macs:
        raw_data +=Collector_population_detail.objects.filter(collector_mac=mac).filter(counted_a
    dataset = []
    date_time = []
    c = []
    for row in raw_data:

        dev_macs = json.loads(row.device_macs)
        c.append(row.count)
        date_time.append(row.counted_at)
        for key, value in dev_macs.items():
            for i in range(0, len(key), 2):
                if i < len(key) / 2:
                    a += key[i:i + 2].upper() + ":"
                else:
                    a += '**' + ":"
            a = a[:17]
            dataset.append({
                'SCANNED DATE': row.counted_at[:16],
                'MAC ADDRESS': a,
                'RSSI': int(sum(value)/len(value)) * (-1)
            })

    data = {'dataset': dataset, 'counts': c, 'time': date_time}

    return  JsonResponse(data, safe=False)
```

09.11

09.11

기록자(Recorded by) 홍미영	작성일(Date) 2020. 09. 11	서 명(Signature)
점검자(Witnessed by) 이재희	점검일(Date) 2020. 09. 11	서 명(Signature)

과제명 (Project)	2.3 wifi data processing (이지은)
제 목 (Title)	2.3.2 coding (김지영)
Continued from page	27~30

2.3.2.1 coding algorithm (Pseudo code) (강선희, 김지영)

✓ coding algorithm (Pseudo code)

We have used Python programming language for data collection and processing

09.18

```
1  from datetime import datetime as datetime2, timedelta
2  from model import median
3  import calendar
4  import datetime
5  import hashlib
6  import os
7  import random
8  import time
9  import json
10 import urllib
11 #from OuiLookup import OuiLookup
12 from nested_dict import nested_dict as nd
13 import pickle as pk
14 import boto3
15 import decimal
16 import time
17 import pymysql
18 import math
19 from redis import ConnectionPool, StrictRedis
20
21
22 rssi_threshold = 50
23 mac_vendor_db = pk.load(open('oui.pk', 'rb'))
24 random_mac_normalizer = 8
25
26 def round_mnt(dt):
27     given_time = dt
28     delta = timedelta(minutes=5)
29     rounded = dt + (datetime2.min - dt) % delta
```

27~30 Continued to page

기록자(Recorded by) 윤준영	작성일(Date) 2020. 09. 18	서 명(Signature)
점검자(Witnessed by) 이제원	점검일(Date) 2020. 09. 18	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	27~30

```
30      return rounded
31
32  def process_logs(probes):
33      scan_data = nd(2, list)
34      for row in device_logs:
35          if not row:
36              continue
37          row = row.split('\t')
38          collector_mac = row[0]
39          push_time = int(row[1])
40          device_mac = row[2]
41          device_time = int(row[3])
42          signal_type=row[5]
43          rssi=abs(int(row[4]))
44          if rssi > 55:
45              continue
46          counted_at = round_mnt((device_time)   # rounidng minute to the closest 5th minute
47          collector_key = collector_mac + '_'+ 'nipa' +'_' + str(counted_at)
48          scan_data[collector_key][device_mac].append(rssi)
49      return scan_data
50
51  def get_rid_of_overlaps(device_logs):
52      overlaps_cleaned = nd(2, int)
53      for counted_at, group_uuids in device_logs.items():
54          for group_uuid, collectors in group_uuids.items():
55              all_devices = get_all_devices_macs(collectors)
56              added_devices = []
57              for col_mac, scanned_devices in collectors.items():
58                  for device_mac, rssis in scanned_devices.items():
59
60                      if device_mac in added_devices: #The device has already been attached to the closest co
61                          continue
62                      device_rssi = sum(rssis)/len(rssis)
63                      if device_rssi > rssi_threshold and col_mac.decode("utf-8").split('_')[0] not in hospit
64                          continue
65                      if device_rssi != all_devices[device_mac]:
66                          continue
67                      overlaps_cleaned[col_mac][device_mac] = rssis
68                      added_devices.append(device_mac)
```

27~30	Continued to page

기록자(Recorded by) 조종현	작성일(Date) 2020. 09. 18	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 09. 18	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	21 ~ 30

```
69      return overlaps_cleaned
70
71  def update_dict(stored_dict, scanned_devices):
72      stored_dict = json.loads(stored_dict)
73      for device_mac, rssi_list in scanned_devices.items():
74          if device_mac in stored_dict:
75              stored_dict[device_mac] += rssi_list
76          else:
77              stored_dict[device_mac] = rssi_list
78      return stored_dict
79
80  def get_count(devices):
81      devices2 = {}
82      real, random = set(), set()
83      for device_mac, rssis in devices.items():
84          if device_mac[:6] in mac_vendor_db:
85              device_mac_vendor = mac_vendor_db[device_mac[:6]]
86              if device_mac_vendor not in mac_vendors:
87                  continue
88              real.add(device_mac)
89          else:
90              if len(rssis) > probe_freq:
91                  real.add(device_mac)
92              else:
93                  random.add(device_mac)
94      count = len(real) + math.ceil(len(random)/random_mac_normalizer)
95      return count
96
97  def filter_macs(scanned_devices):
98      filtered_scanned_devices = {}
99      for device_mac, rssi_list in scanned_devices.items():
100         vendor_mac = device_mac[:6]
101         if vendor_mac in mac_vendor_db: #filtering real mac, but not in ['Samsung Electronics Co.,Ltd', 'Ap
102             device_mac_vendor = mac_vendor_db[vendor_mac]
103             if device_mac_vendor not in mac_vendors:
104                 continue
105         filtered_scanned_devices[device_mac] = rssi_list
106     return filtered_scanned_devices
```

09.18

21 ~ 30	Continued to page

기록자(Recorded by) 문 준 영	작성일(Date) 2020. 09. 18	서 명(Signature)
점검자(Witnessed by) 이 재 현	점검일(Date) 2020. 09. 18	서 명(Signature)

과제명 (Project)

제 목 (Title)

Continued from page 27~30

0918

```
107
108 def insert_to_rds():
109     collector_uuid = get_collector_group_uuid()
110     collector_group = nd(3, list)
111
112     with write_conn.cursor() as cursor:
113         for collector_key in redis.scan_iter():
114             scanned_devices = redis.get(collector_key)
115             if not scanned_devices:
116                 continue
117             scanned_devices = json.loads(scanned_devices)
118             collector_mac, _, counted_at = collector_key.decode("utf-8").split('_')
119
120             try:
121                 group_uuid = collector_uuid[collector_mac]
122             except:
123                 print(f'Collector_mac:{collector_mac} has no group_uuid')
124                 continue
125
126             collector_group[counted_at][group_uuid][collector_key] = scanned_devices
127     overlaps_cleaned = get_rid_of_overlaps(collector_group)
128     with write_conn.cursor() as cursor:
129         for collector_key, scanned_devices in overlaps_cleaned.items():
130             collector_key = collector_key.decode("utf-8")
131             collector_mac, _, counted_at = collector_key.split('_')
132             count = get_count(scanned_devices)
133             query = f"INSERT INTO collector_population"
134             cursor.execute(query, (collector_mac, counted_at, collector_key))
135         write_conn.commit()
136
137 def main():
138      with redis.pipeline(transaction=False): #redis is a temporary data base
139          for collector_key, scanned_devices in scan_data.items():
140              stored_dict = redis.get(collector_key)
141              if stored_dict:
142                  updated_stored_dict = update_dict(stored_dict, scanned_devices)
143                  redis.set(collector_key, json.dumps(updated_stored_dict))
144              else:
145                  redis.set(collector_key, json.dumps(scanned_devices))
146                  redis.expire(collector_key, 60*15)
147
148     if datetime2.now().minute%5==0:
149         insert_to_rds()
```

27~30

기록자(Recorded by) 문준영	작성일(Date) 2020. 09. 18	서 명(Signature)
점검자(Witnessed by) 이 재 현	점검일(Date) 2020. 09. 18	서 명(Signature)

과제명 (Project)	3.3 Building Dashboard (대시보드)
제 목 (Title)	3.3.1 Gathering technology (기술)
Continued from page	

3.3.1.1 Styling technology (인증)

To make the our dashboard user friendly, it's wise to keep all the rules that determine the key elements of the website in one place - we're talking about the style guide for the visualization of the website.

✓ Elements

Inside the style guide we determine

- colors : primary and secondary colors, typography lay out on colored background. primary colors for highlights, link and button colors, etc.
- typography : the size of the main titles (H1 - H5), the size of the body text. quotes, spacing between ~~this~~ titles and body text, typography color and background colors where the text can appear.
- Grid : determining the spacing inside the elements, styles.
- Graphics : drawing the key icons that will represent the main content.
- other : determining the main style of the website, the color variations of pictures, button style, contact from style, call to action section, testimonials, etc.

There are several CSS Libraries which can help us a style our dashboard. One of them is Bootstrap 5. It is a free and open-source CSS framwork directed at responsive mobile-first front-end web development. It contains CSS design tendates for typography forms, buttons, navigation and other interface components.

In our project we will be using bootstrap library since it is one of the best styling libraries on the market.

Continued to page

과제명 (Project)	2_3 Wi-Fi data processing (이지은)
제 목 (Title)	2.3.2 coding (김지명)
Continued from page	32 ~ 36

2.3.2.2 Testing / Debugging (배지석, ~~김수진~~ 박수진)

✓ Testing Debugging.

We had to adjust rssi_threashold and random mac randomization as below

rssi_threashold : -50 → -55

random_mac_normalizer : 8 → 10

average(rssis) → median(rssis)

```
1  from datetime import datetime as datetime2, timedelta
2  from model import median
3  import calendar
4  import datetime
5  import hashlib
6  import os
7  import random
8  import time
9  import json
10 import urllib
11 #from OuiLookup import OuiLookup
12 from nested_dict import nested_dict as nd
13 import pickle as pk
14 import boto3
15 import decimal
16 import time
17 import pymysql
18 import math
19 from redis import ConnectionPool, StrictRedis
20
21 rssi_threshold = 55
22 mac_vendor_db = pk.load(open('oui.pk', 'rb'))
23 random_mac_normalizer = 10
24
25 def round_mnt(dt):
26     given_time = dt
27     delta = timedelta(minutes=5)
28     rounded = dt + (datetime2.min - dt) % delta
29     return rounded
30
```

0930

기록자(Recorded by) 김 조명	작성일(Date) 2020. 09. 30	서 명(Signature)
점검자(Witnessed by) 이 재현	점검일(Date) 2020. 09. 30	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	32 ~ 36

```
31  def process_logs(probes):
32      scan_data = nd(2, list)
33      for row in device_logs:
34          if not row:
35              continue
36          row = row.split('\t')
37          collector_mac = row[0]
38          push_time = int(row[1])
39          device_mac = row[2]
40          device_time = int(row[3])
41          signal_type=row[5]
42          rssi=abs(int(row[4]))
43          if rssi > 55:
44              continue
45          counted_at = round_mnt((device_time)   # rounidng minute to the closest 5th minute
46          collector_key = collector_mac + '_'+ 'nipa' +'_' + str(counted_at)
47          scan_data[collector_key][device_mac].append(rssi)
48      return scan_data
49
50  def get_rid_of_overlaps(device_logs):
51      overlaps_cleaned = nd(2, int)
52      for counted_at, group_uuids in device_logs.items():
53          for group_uuid, collectors in group_uuids.items():
54              all_devices = get_all_devices_macs(collectors)
55              added_devices = []
56              for col_mac, scanned_devices in collectors.items():
57                  for device_mac, rssis in scanned_devices.items():
58
59                      if device_mac in added_devices: #The device has already been attached to the closest co
60                          continue
61                      device_rssi = median(rssis)
62                      if device_rssi > rssi_threshold and col_mac.decode("utf-8").split('_')[0] not in hospit
63                          continue
64                      if device_rssi != all_devices[device_mac]:
65                          continue
66                      overlaps_cleaned[col_mac][device_mac] = rssis
67                      added_devices.append(device_mac)
68      return overlaps_cleaned
69
```

0930

		Continued to page
	32 ~ 36	

기록자(Recorded by) 홍 종 영	작성일(Date) 2020. 9. 30	서 명(Signature)
점검자(Witnessed by) 이 재 현	점검일(Date) 2020. 09. 30	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	32~36

0930

```
70  def update_dict(stored_dict, scanned_devices):
71      stored_dict = json.loads(stored_dict)
72      for device_mac, rssi_list in scanned_devices.items():
73          if device_mac in stored_dict:
74              stored_dict[device_mac] += rssi_list
75          else:
76              stored_dict[device_mac] = rssi_list
77      return stored_dict
78
79  def get_count(devices):
80      devices2 = {}
81      real, random = set(), set()
82      for device_mac, rssis in devices.items():
83          if device_mac[:6] in mac_vendor_db:
84              device_mac_vendor = mac_vendor_db[device_mac[:6]]
85              if device_mac_vendor not in mac_vendors:
86                  continue
87              real.add(device_mac)
88          else:
89              if len(rssis) > probe_freq:
90                  real.add(device_mac)
91              else:
92                  random.add(device_mac)
93      count = len(real) + math.ceil(len(random)/random_mac_normalizer)
94      return count
95
96  def filter_macs(scanned_devices):
97      filtered_scanned_devices = {}
98      for device_mac, rssi_list in scanned_devices.items():
99          vendor_mac = device_mac[:6]
100         if vendor_mac in mac_vendor_db: #filtering real mac, but not in ['Samsung Electronics Co.,Ltd', 'Ap
101             device_mac_vendor = mac_vendor_db[vendor_mac]
102             if device_mac_vendor not in mac_vendors:
103                 continue
104         filtered_scanned_devices[device_mac] = rssi_list
105     return filtered_scanned_devices
106
```

32~36 Continued to page

기록자(Recorded by) 조 진 영	작성일(Date) 2020. 09. 30	서 명(Signature)
점검자(Witnessed by) 이 재 현	점검일(Date) 2020. 09. 30	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	32~36

```python
def insert_to_rds():
    collector_uuid = get_collector_group_uuid()
    collector_group = nd(3, list)

    with write_conn.cursor() as cursor:
        for collector_key in redis.scan_iter():
            scanned_devices = redis.get(collector_key)
            if not scanned_devices:
                continue
            scanned_devices = json.loads(scanned_devices)
            collector_mac, _, counted_at = collector_key.decode("utf-8").split('_')

            try:
                group_uuid = collector_uuid[collector_mac]
            except:
                print(f'Collector_mac:{collector_mac} has no group_uuid')
                continue
            collector_group[counted_at][group_uuid][collector_key] = scanned_devices

    overlaps_cleaned = get_rid_of_overlaps(collector_group)

    with write_conn.cursor() as cursor:
        for collector_key, scanned_devices in overlaps_cleaned.items():

            collector_key = collector_key.decode("utf-8")
            collector_mac, _, counted_at = collector_key.split('_')

            count = get_count(scanned_devices)
            query = f"INSERT INTO collector_population"
            cursor.execute(query, (collector_mac, counted_at, collector_key))
        write_conn.commit()
```

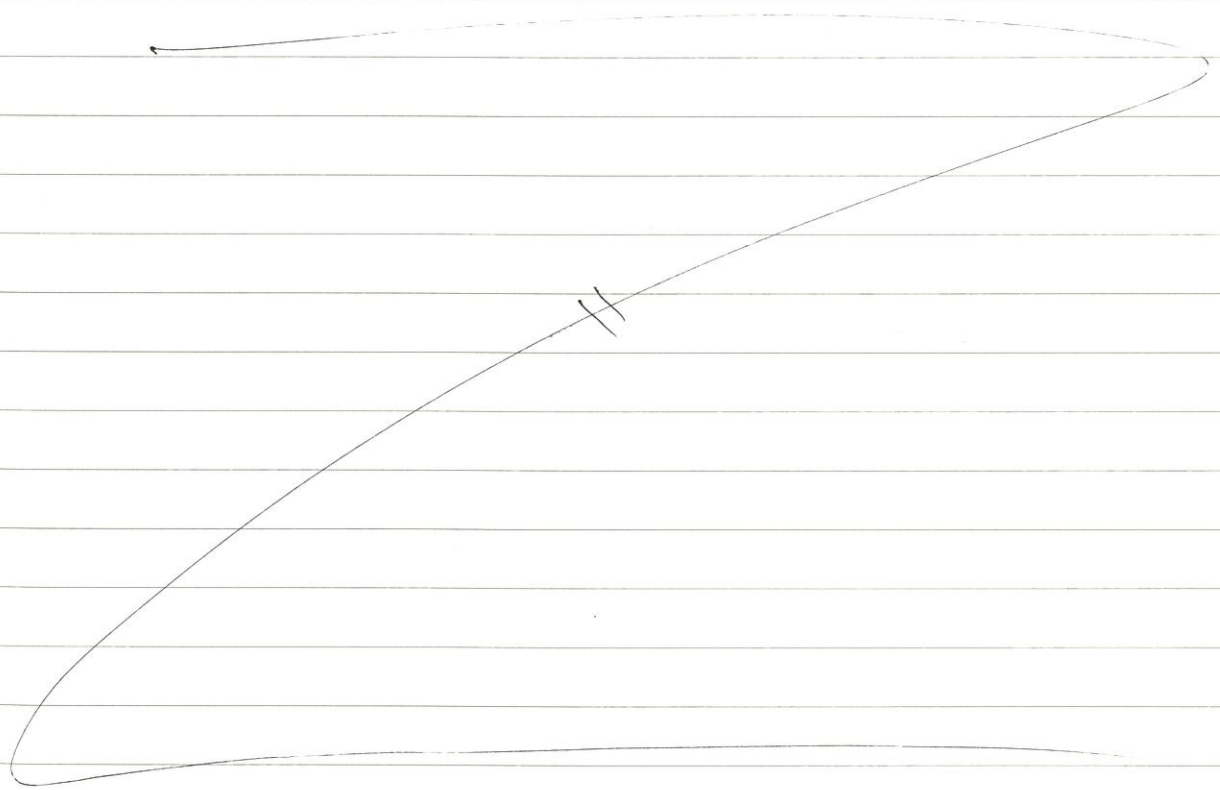
09 30

기록자(Recorded by) 윤 종 영	작성일(Date) 2020. 09. 30	서 명(Signature)
점검자(Witnessed by) 이 재 현	점검일(Date) 2020. 09. 30	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	32-36

```
139  def main():
140       with redis.pipeline(transaction=False): #redis is a temporary data base
141           for collector_key, scanned_devices in scan_data.items():
142               stored_dict = redis.get(collector_key)
143               if stored_dict:
144                   updated_stored_dict = update_dict(stored_dict, scanned_devices)
145                   redis.set(collector_key, json.dumps(updated_stored_dict))
146               else:
147                   redis.set(collector_key, json.dumps(scanned_devices))
148                   redis.expire(collector_key, 60*15)
149
150      if datetime2.now().minute%5==0:
151          insert_to_rds()
```

09.30



32-36 Continued to page

기록자(Recorded by) 윤 종 명	작성일(Date) 2020.09.30	서 명(Signature)
점검자(Witnessed by)	점검일(Date)	서 명(Signature)

과제명 (Project)	3.3 Building Dashboard (아부바크르)
제 목 (Title)	3.3.1 Gathering technologies (홍미영)
Continued from page	

3.3.1.2 Scripting Technologies (김진수)

✓ JavaScript and Technologies

Javascript is very very important in front-end development. It gives the site additional functionality that isn't otherwise achievable with HTML and CSS alone. Java script allows web-pages to respond to user activity and dynamically update themselves, and all without requiring a page reload to change appearance.

We will be using some Javascript libraries like Jquery which helps us to write less code do more jobs.

Another one of the most Important technology is AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes.

This means that it is possible to update parts of web page, without reloading the whole page.

Continued to page

기록자(Recorded by) 김진수	작성일(Date) 2020.09.30	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020.09.30	서 명(Signature)

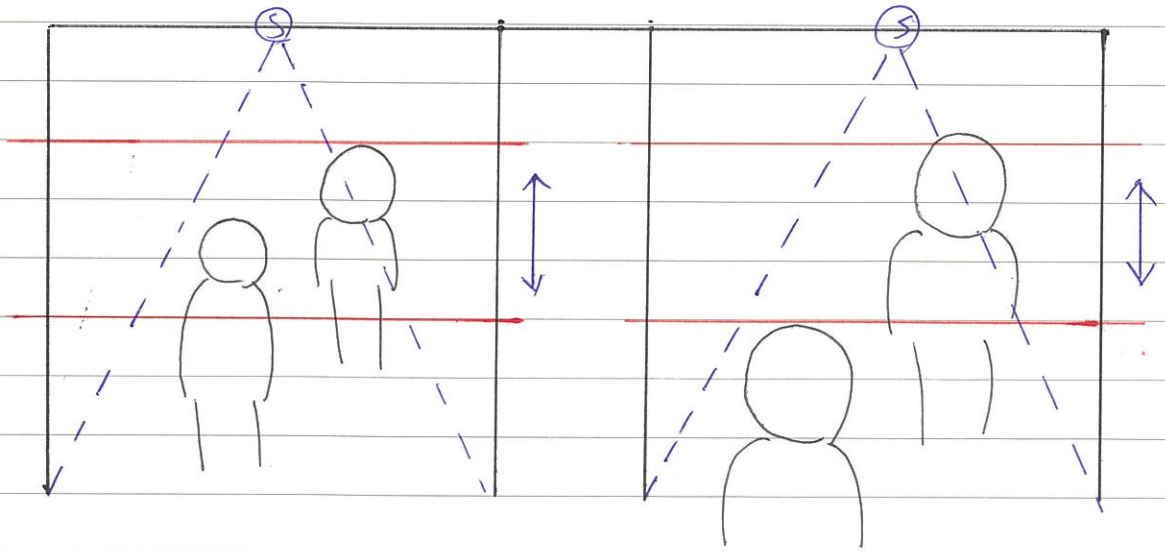
과제명 (Project)	2.4 long-distance ranging TOF sensor (이자은)
제 목 (Title)	2.4.1 overview (김지영)
Continued from page	2.4.1.1 Intro (강민채)

Tof sensor can be detect people as they enter and exit a room The ue of a TOF sensor can help bilding managers keep track of occupation and control their spaces for a very affordable cost.

Although called a people counter, the TOF simply detects a shape under the door jam and determines the direction of travel.

The device is then able to use this information to determine whether the shape is entering or exiting the room.

The TOF (VL53L1X) is a state-of-the-art, Time-of-Flight (TOF) laser, ranging sensor.



Continued to page

기록자(Recorded by) 김지영	작성일(Date) 2020.10.09	서 명(Signature)
점검자(Witnessed by) 이재희	점검일(Date) 2020.10.09	서 명(Signature)

과제명 (Project)	3.3 Building Dash board (아부바그드)
제 목 (Title)	3.3.2 Developing Web Pages and Navigation
Continued from page	

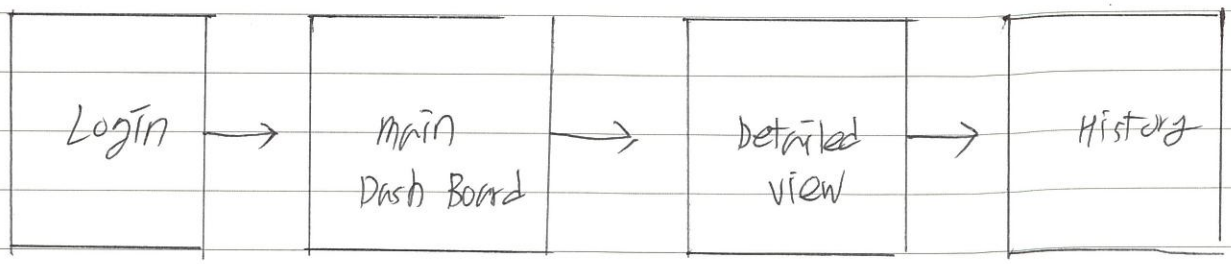
3.3.2.1 Developing Web pages.

Our dash board going to have the following Web Pages.

- ✓ Login
- ✓ Main Dash board
- ✓ Detailed view
- ✓ History

We will start with developing our login page it will have main Logo and username and password from to authenticate user.

After authentication user will be redirected to Main dashboard. In main dash board page we will have main navigation and main dash board charts to show how many people are in the building And there will detailed view of every chart and there will be page for checking history of records.



< Web Dash Board flow >

Continued to page

기록자(Recorded by) 김 [illegible]	작성일(Date) 2020. 10. 14	서 명(Signature)
점검자(Witnessed by) 이 [illegible]	점검일(Date) 2020. 10. 14	서 명(Signature)

2. Data analysis (데이터 분석)

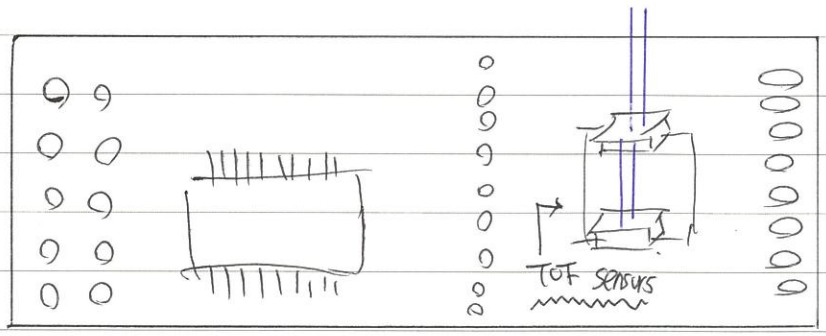
과제명 (Project)	2.4 long Distance ~~ranging kit~~ ranging Time-of-flight (이자은)
제 목 (Title)	2.4.2 VL53L1X algorithm development. (김지영)
Continued from page	

2.4.2.1 Introduction. (강민재)

✓ How to use a VL53L1X long-distance ranging Time-of-Flight (TOF) sensors to count people ~~crossing~~ crossing entrance.

It also describes an algorithm used to count the people. In addition, we present the details of a counting application where a sensor is installed on the top, not the side, of the area to be tracked.

VL53L1X is the fast miniature ToF sensor with accurate ranging up to 4m and fast ranging frequency up to 50 Hz. Housed in a miniature and reflowable package, it integrates a SPAD receiving array, a 940 nm invisible class 1 laser emitter, physical infrared filters, and optics to achieve the best ranging performance in various ambient lighting conditions with a range of cover window options.



Continued to page

기록자(Recorded by) 김지영	작성일(Date) 2020. 10. 16	서 명(Signature)
점검자(Witnessed by) 이재원	점검일(Date) 2020. 10. 16	서 명(Signature)

과제명 (Project)	3.3 Building Dash Board (대시보드)
제 목 (Title)	3.3.2 Developing Web Pages and Navigations (웹페이지)
Continued from page	41-42

3.3.2.2 Styling Web Pages. (웹스타일)

✓ The basic components that create a style guide designed for dashboards : Grid

A Grid is a technical ~~term~~ term that defines a system of lines in which the designer can consistently organize graphic elements (e.g headlines, graphs, etc.) It is recomended to design all dashboards from the same series using an identical grid in order to create similarity and consistency.

✓ color ~~pallet~~ pt Palette.

A consistent colors palette should be used in all dashboards.
Best practice would be to decide the colors into main and secondary colors and define a "rule" for each color : background color, headings colors graph's colors.

✓ Typography.

A specific font (or fonts) should be used in all dashboards using different weights. some weights will be used for headings while others will be used for text or dingrams. It is important to understand that a typeface is a basic design component. It is one of the ingredients which determines what will be the "feel" of the dashboard.

✓ Logo

Usually, when planning a style guide for the organization's dashboards one should use an ~~exi~~ existing logo or brand visuals.

41 - 42 Continued to page

기록자(Recorded by) 윤 국 영	작성일(Date) 2020. 10. 16	서 명(Signature)
점검자(Witnessed by) 이 재 원	점검일(Date) 2020. 10. 16	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	41 - 42

It is important to educate the developers about the Importance the Logo and the company's brand - using the logo as is without changing its colors, proportions or shape. The style guide also determines where to place the logo on the screen and in what size to use it

✓ Icons

Icons can emphasize KPIs or any other dashboard components. They can also create visual interest and break the monotonous pattern of diagrams and graphs. However, it is extremely important to use a specific icon set that is visually connected to the general visual language of the brand.

* Using this all tips we will style our dashboard.

41 - 42 Continued to page

기록자(Recorded by)	작성일(Date)	서 명(Signature)
조 준영	2020. 10. 16	
점검자(Witnessed by) 이 재희	점검일(Date) 2020. 10. 16	서 명(Signature)

과제명 (Project)	2.4 Counting people with the long-distance ranging TOF sensor (이재원)
제 목 (Title)	~~2.4.1 Overview~~ 2.4.2 VL53LX ToF algorithm development. (김지영)
Continued from page	43 - 44

2.4.2.2 overview (김지영)

For the ease of reading we have made the Acronyms and abbreviations table as below.

< Acronym / abbreviation >	< Definition >
✓ FOV	✓ field of view
✓ SPAD	✓ single photon avalanche diode
✓ SW	✓ software
✓ TOF	✓ Time-of-flight.

Counting people with the VL53L1X consists of using the multiple zones of the sensor receiving single photon avalanche diode area, and of configuring it with two distinct fields of view (FoV), to alternatively get a ranging distance from them and consequently recognise the movements of a person.

Using this method, the number of peole occupying a meeting room, accessible from a reasonably narrow access, is known at all times by detecting the entrances and exits of the attendees. By mensuring and analyzing the distances of targets within the FoVs of a front and back zone (see figure below), a simple aboritthm can detect the direction a person crosses the area under the two FoVs. This algorithm "understands" that someone is under one of the FoV as long as the distance measured by the sensor under this FoV is between 0 and threshold value specified in mm

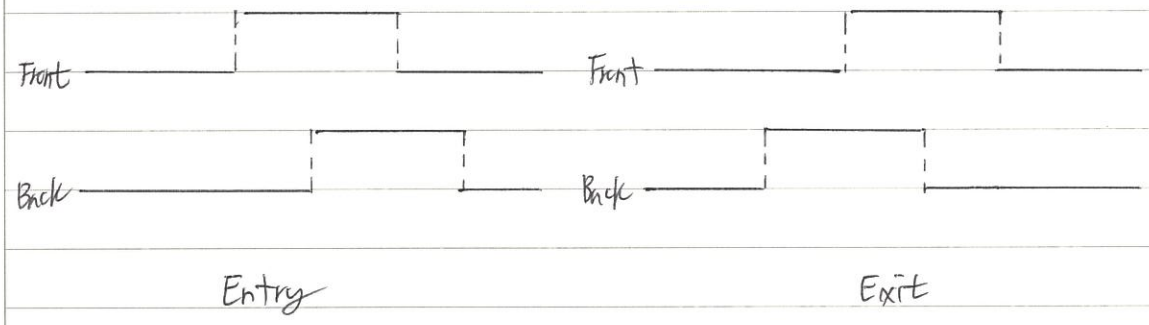
43 - 44 Continued to page

기록자(Recorded by) 김 지영	작성일(Date) 2020. 10. 23	서 명(Signature)
점검자(Witnessed by) 이 재원	점검일(Date) 2020. 10. 25	서 명(Signature)

과제명 (Project)	
제 목 (Title)	
Continued from page	43 - 44

From a timing perspective, the sensor alternatively ranges on each of the two zones, for a very short period of time in milliseconds.

It is possible to determine in which direction a person crosses the area, depending in which order this person has been detected in the two zones, as shown in the figure below.



< person counting chronogram >

43 - 44 Continued to page

기록자(Recorded by) 오정명	작성일(Date) 2020. 10. 23	서 명(Signature)
점검자(Witnessed by) 이재희	점검일(Date) 2020. 10. 23	서 명(Signature)

과제명 (Project)	3.3 Building Dashboard (대쉬보드)
제 목 (Title)	3.3.3 Populating pages with charts (홍대영)
Continued from page	

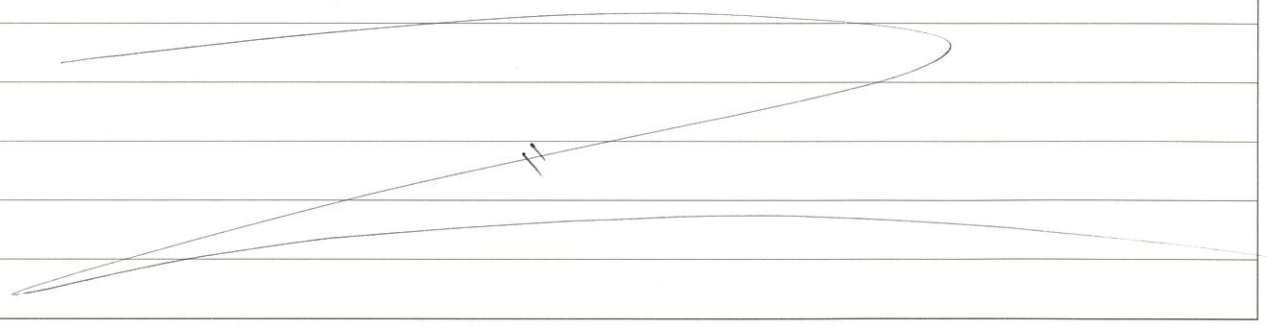
3.3.3.1 Populating charts with data (김혜영)

In populating charts with data we use JavaScript because our charting libraries are built in Javascript. It give us full control over our data which is also Javascript object. We use Ajax to send and retriever data from server.

Ajax work on background does not interfere with our interface and while retrieving data it will not ~~reload~~ reload our page and can update data on charts in real-time which improves user interface.

We are using Apexcharts as our charting library and this library can handle several data feeding methods such as separate single value arrays of data for both x-axis and y-axis or array of javascript object with x and y keys.

In our dashboard we will be using second method because it is easy to use with our response data.



Continued to page

기록자(Recorded by) 홍준영	작성일(Date) 2020. 10. 23	서 명(Signature)
점검자(Witnessed by) 이재영	점검일(Date) 2020. 10. 23	서 명(Signature)

과제명 (Project)	2.4 Counting people with the long distance TOF (이지은)
제 목 (Title)	2.4.2 VL53L1X TOF sensor algorithm (강지원)
Continued from page	

2.4.2.3 Algorithm description · flow chart. (배지수)

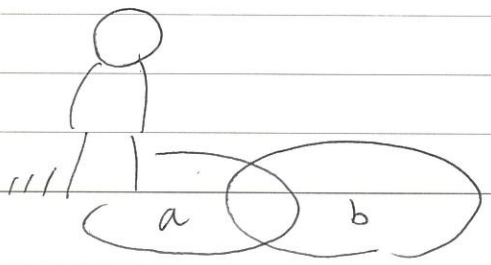
The counting algorithm relies on a list of states (numbers) that have to occur in a certain order to detect if a person has crossed the specified area and in which direction this area has been crossed.

These states are stored in a list and compared to two default lists of states that represent how the area is crossed in two different directions. #

When no-one is seen in either of the two zones, the list of state is reset.

If we consider that a person detected in the front zone equals 2 and a person detected in the back zone equals 1, the algorithm adds the value of the two ~~zones~~ states and stores the result as soon as it changes.

Eventually, if the ~~consecq~~ consecutive states in the list are 0,1,3,2 or 0,2,3,1,0 this means a person has been detected in one direction or the other, as described in the figures below.



- a: ~~first zone~~ front Zone
- b: Back zone.

Continued to page

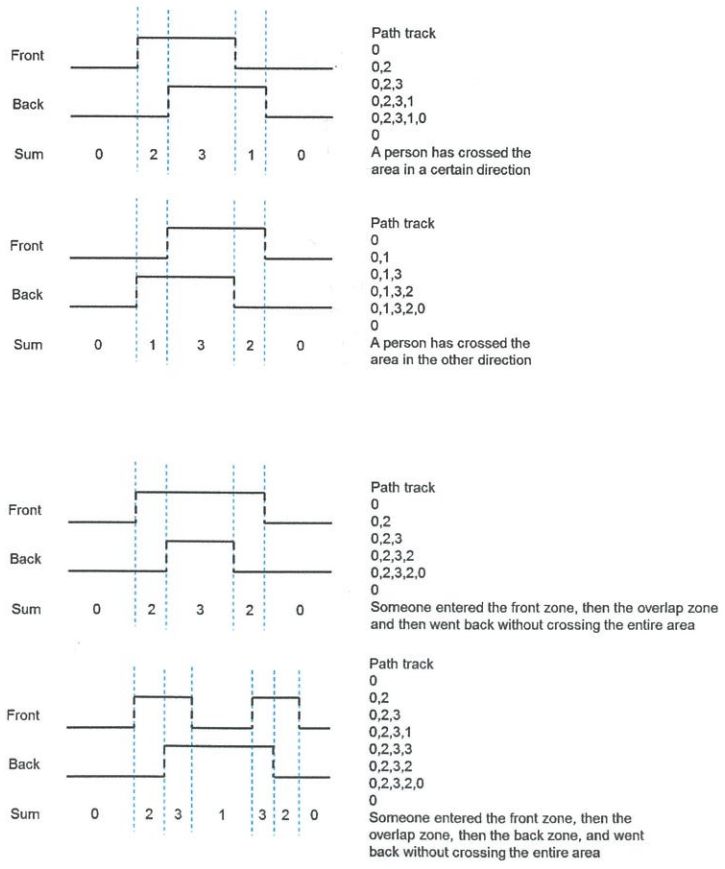
기록자(Recorded by) 강지원	작성일(Date) 2020. 10. 30	서 명(Signature)
점검자(Witnessed by) 이지은	점검일(Date) 2020. 10. 30	서 명(Signature)

과제명 (Project)	2.4 Counting people with the long-Distance TOF (이자은)
제 목 (Title)	2.4.2 VL53L1X TOF sensor Algorithm development (김지영)
Continued from page	

2.4.2.4 Algorithm validation (박수진)

The algorithm validates a crossing event only when a person has fully crossed the two zones. It does not validate the event when the person remains for a long time under the FoV or when the person decides to return from the place he came from.

This is illustrated in the figure below : the algorithm stop and the list of states is reset as soon as no-one is detected in any of the two FoVs.



<Figure 1. Validating principle>

Continued to page

기록자(Recorded by) 김 지영	작성일(Date) 2020. 11. 10	서 명(Signature)
점검자(Witnessed by) 이 재원	점검일(Date) 2020. 11. 10	서 명(Signature)

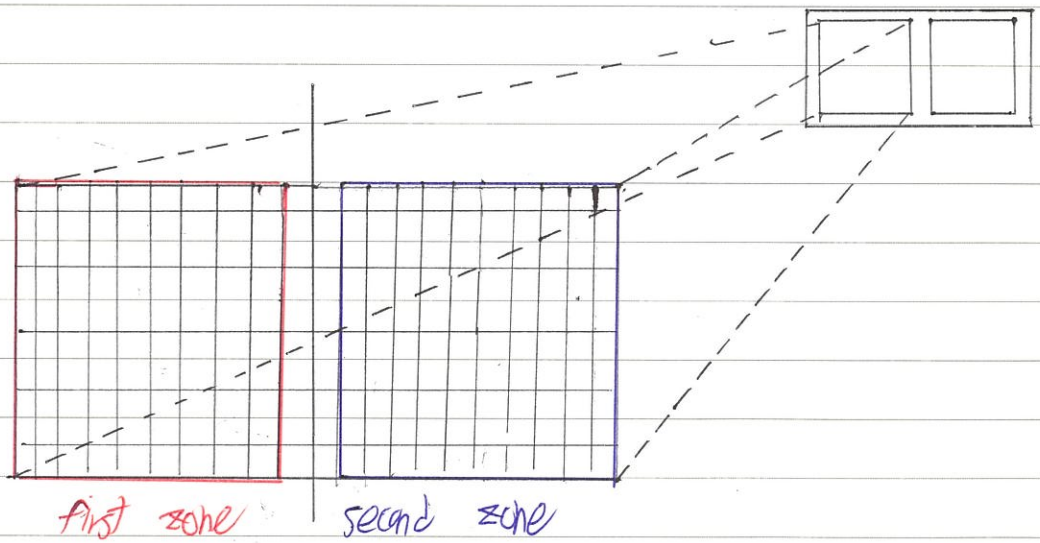
과제명 (Project)	2.4 Counting people with long-Distance TOF (이자은)
제 목 (Title)	2.4.2 VL53L1X TOF sensor Algorithm (김지명)
Continued from page	

2.4.2.5 Sensor / Algorithm configuration (김지명)

It is sufficient to set the two FoVs by dividing the 16/16 SPAD array of the VL53L1X sensor into two sub N16 arrays. For example, N can be chosen within the interval [4, 8].

This means that it is not necessary to use the same SPADs for front and back ranging.

Alternatively, enabling 1×16 SPADs is sufficient to see a person in the two zones at the same time when this person stands directly under the sensor, in the middle of the two FoVs.



<Example of configuration of the SPAD array>

Continued to page

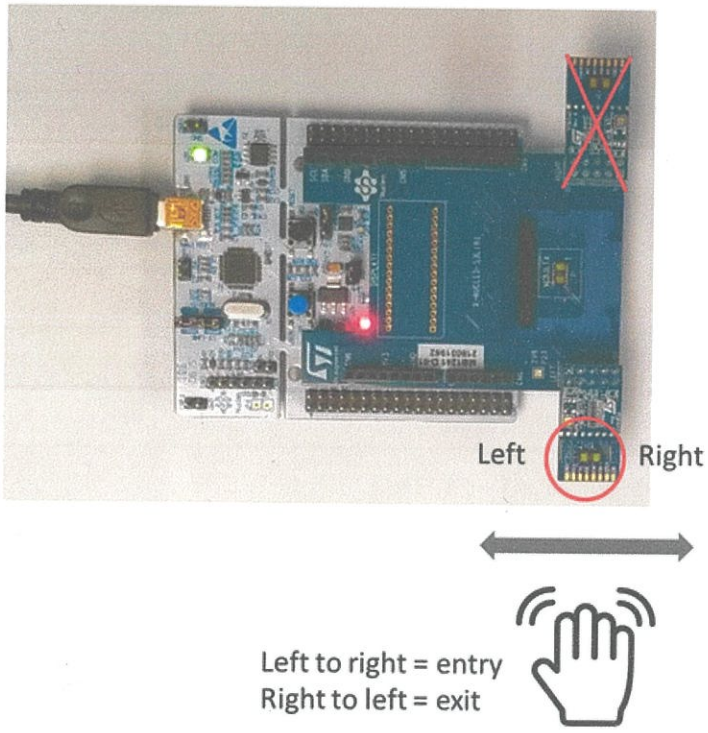
기록자(Recorded by) 김지명	작성일(Date) 2020. 11. 20	서 명(Signature)
점검자(Witnessed by) 이 재	점검일(Date) 2020. 11. 20	서 명(Signature)

과제명 (Project)	2.4 Counting people with the long-distance TOF sensor (이지은)
제 목 (Title)	2.4.2 VL53L1X TOF sensor based algorithm (김지영)
Continued from page	

2.4.2.6 Algorithm implementation.

Implementing the algorithm as described in the sections above the software runs on a NUCLEO F401RE board and is accompanied by one X-NUCLEO 53L1A1 expansion board.

The VL53L1X "left" satelite (see figure below) is the one enabled by the software example.



= 11시

To test software, you can move your hand from right to left or from left to right, at a distance of around 30cm above the sensor at a reasonable speed. Moving youre hand simulates a person crossing a specified area which gives an output on a counter. The serial speed is set at the 460800 bauds per second. The figure below shows the captured distances from two zones, where the VL53HX is set at 2600 mm from the floor. Three crossings are visible.

기록자(Recorded by) 이 지 은	작성일(Date) 2020. 11. 27	서 명(Signature)
점검자(Witnessed by) 이 재 형	점검일(Date) 2020. 11. 27	서 명(Signature)

과제명 (Project)	2.4. Counting people with the long distance ToF sensor
제 목 (Title)	2.4.2 VL53HX ToF sensor Algorithm
Continued from page	

Coding (peseudo coding) / Test / Debugging

We begin by creating a SFEVL53HX object called distanceSensor with our wire port, wire, and then our shutdown and interrupt pins. Then we initialized our sensor object in the setup() loops. The code to do this is shown below and is repeated in some from in all of the examples.

```
//Optional interrupt and shutdown pins.
#define SHUTDOWN_PIN 2
#define INTERRUPT_PIN 3

SFEVL53L1X distanceSensor(Wire, SHUTDOWN_PIN, INTERRUPT_PIN);

void setup(void)
{
  Wire.begin();

  Serial.begin(9600);
  Serial.println("VL53L1X Qwiic Test");

  if (distanceSensor.init() == false)
    Serial.println("Sensor online!");

}
```

11
1130

```
void loop(void)
{
  distanceSensor.startRanging(); //Write configuration bytes to
initiate measurement
  int distance = distanceSensor.getDistance(); //Get the result of the
measurement from the sensor
  distanceSensor.stopRanging();

  Serial.print("Distance(mm): ");
  Serial.print(distance);

  float distanceInches = distance * 0.0393701;
  float distanceFeet = distanceInches / 12.0;

  Serial.print("\tDistance(ft): ");
  Serial.print(distanceFeet, 2);

  Serial.println();
}
```

Once we've initialized our sensor, we can start grabbing measurements from it. To do this, we send some configuration bytes to our sensor using distanceSensor.startRanging() to initiate the measurement.

=1130

We then wait for data to become available and when it does, we read it in, convert it from millimeters to feet, and print it out over serial. The void loop() function that does this is shown below.

50 ~ 51 Continued to page

기록자(Recorded by) 이 지 은	작성일(Date) 2020. 11. 30	서 명(Signature)
점검자(Witnessed by) 이 재 현	점검일(Date) 2020. 11. 30	서 명(Signature)