ECG Kit App Development
Final Presentation

조정희, 최윤호, 김준형, 박범순

Contents

Member Introduction

Front End Developer

Park Beom Soon

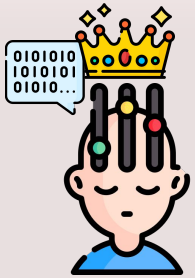
- UI Design, Documentation,
- Chart Implementation

Choi Yun Ho

- BT Connection, Data Transmission,
- Overall Function Implementation

Member Introduction

Back End Developer



Cho Jeong Hui

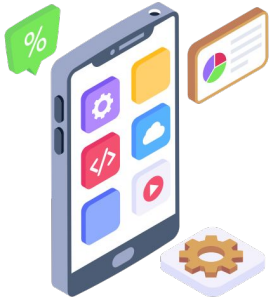
- Main Director, Full-Stack Developer

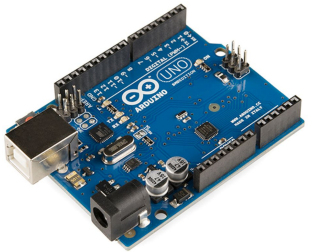
Kim Jun Hyeong

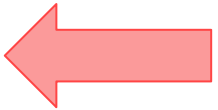
- Server Management

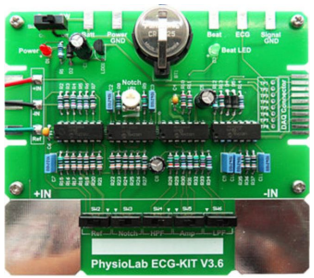
1st Mission

Transfer the measured data to the app through BT connection using Arduino UNO

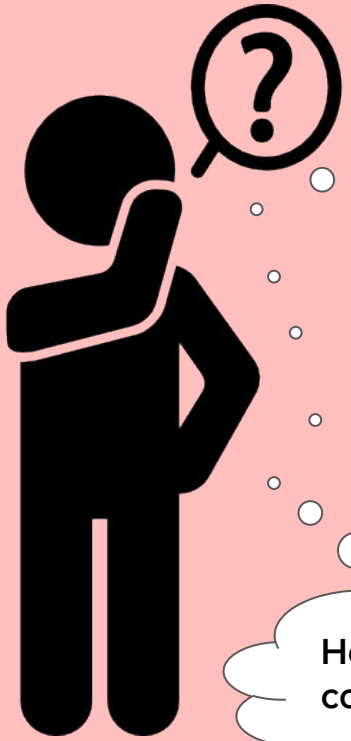








Initial Plan



A week left until the deadline..

What programming language?

What should do with those data?

How can we transfer the data through BT?

How do we compose the app?

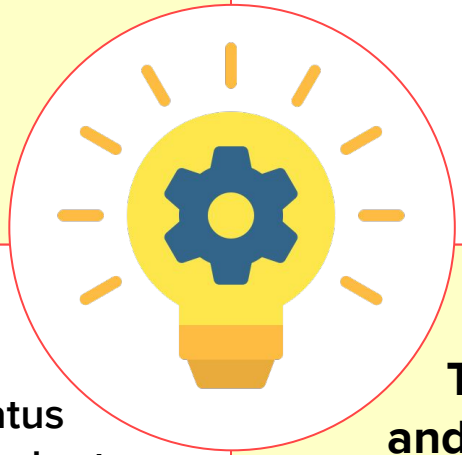
Initial Solution

Divide into 2 teams
(Front-End / Back-End)

One and only RULE
: **Always be punctual and responsible**

App Development : **Android Studio**

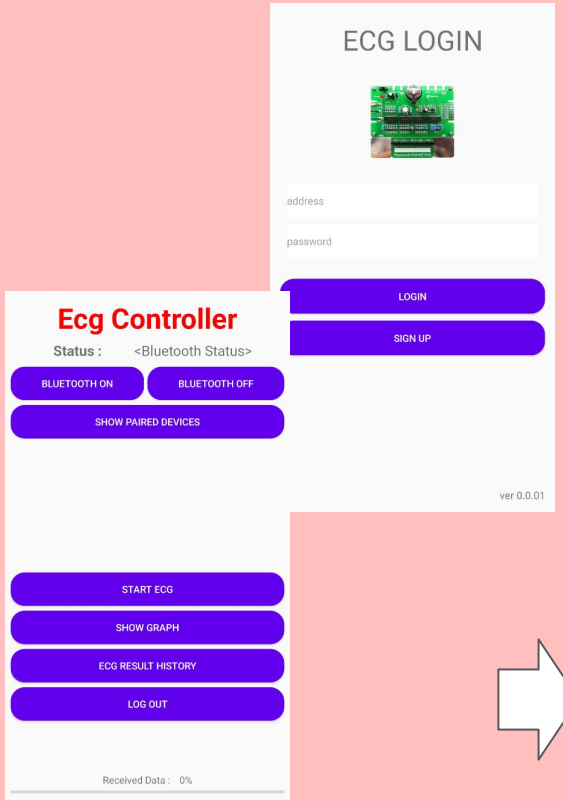
Server (Database)
: **Django**



App Contents
① Examine the user's status
② Show the result with the chart
③ Store the user's results

The most significant and difficult to implement
: **BT Connection**

But..



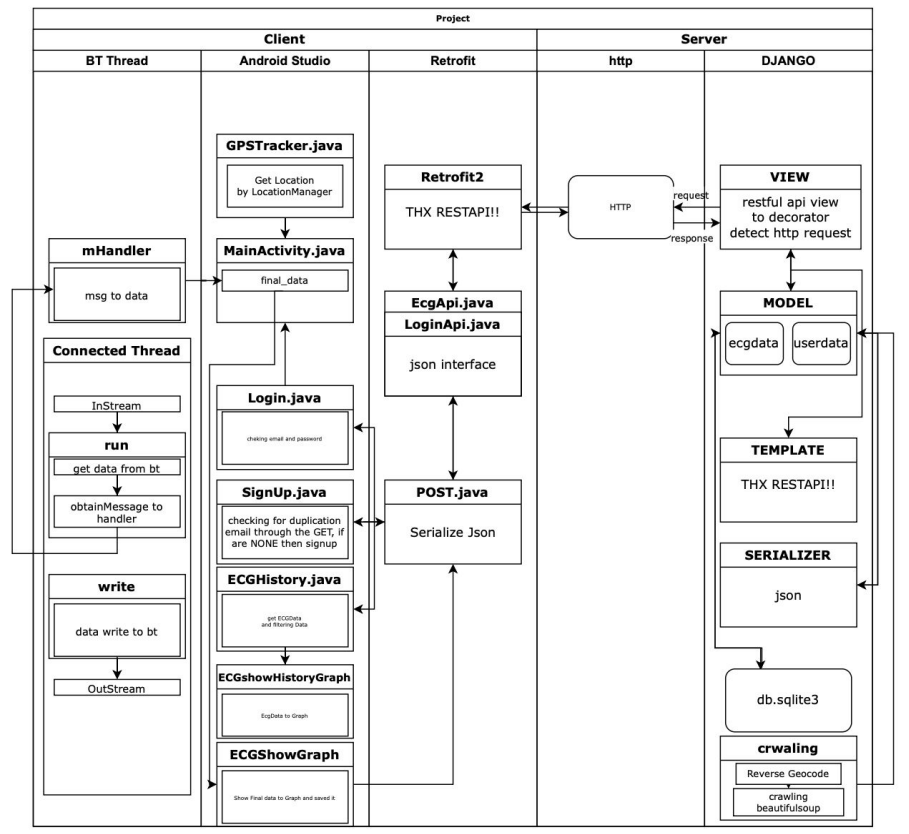
Prof. Kim

Well done within the short time

But, **POOR DESIGN**

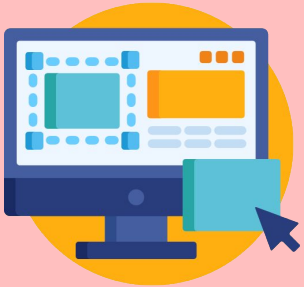
Your team adds more information such as meteorological information and upgrade the app

Our Final Application

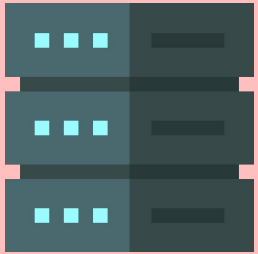


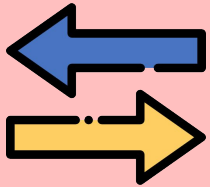
Terminology

Client



Server



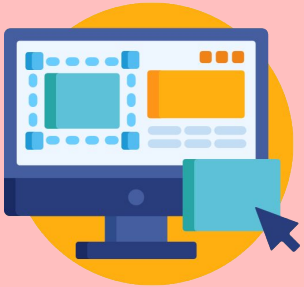


Request for service

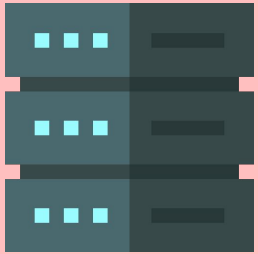
Provide Service in response

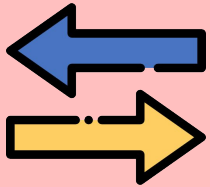
Terminology

Client



Server





Request for service

Provide Service in response

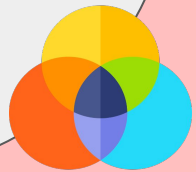
= Application

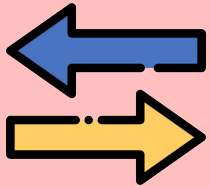
= Database

Terminology

Front-end

What you CAN see from the display (Overall Layout)





Back-end

What you CANNOT see from the display (Data Management)

Terminology

Front-end

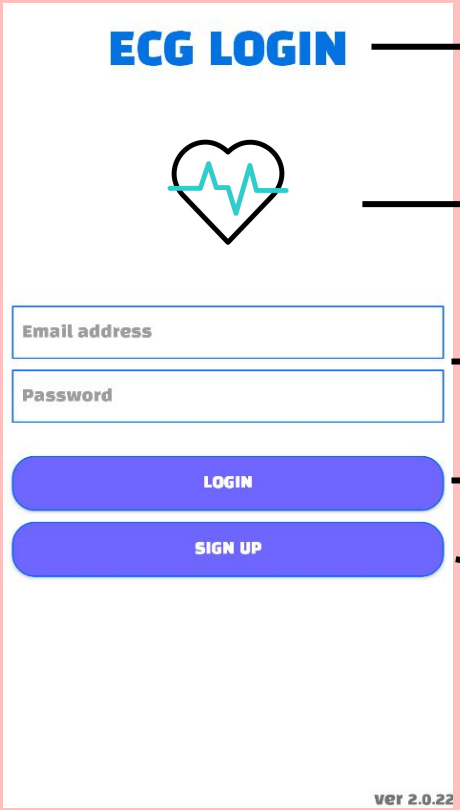
Back-end

What you
from the
(Overall

u CANNOT
the display
anagement)

Full-stack

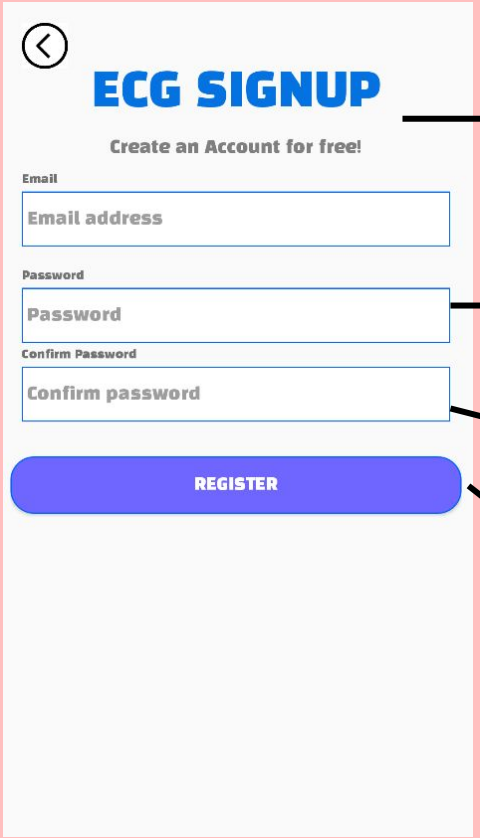
Login Activity



- Title : TextView
- GIF File (ImageView)
- EditTextView (Enterring Email / PW)
- Login (Button)
- Register (Button)

Relative Layout

SignUp Activity



(Sub)Title : TextView

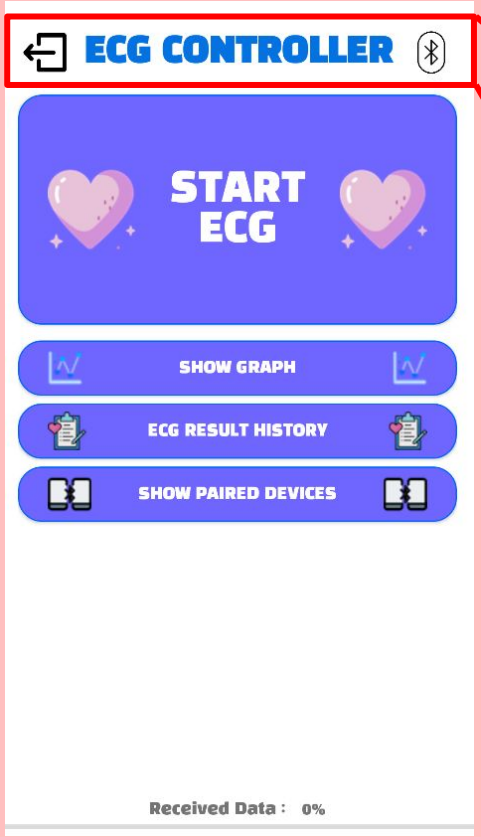
EditTextView
(Enterring Email / PW)

Enter PW 1 more time

Register (Button)

Relative Layout

MainActivity



Title (TextView)

Logout Button (ImageButtonView)

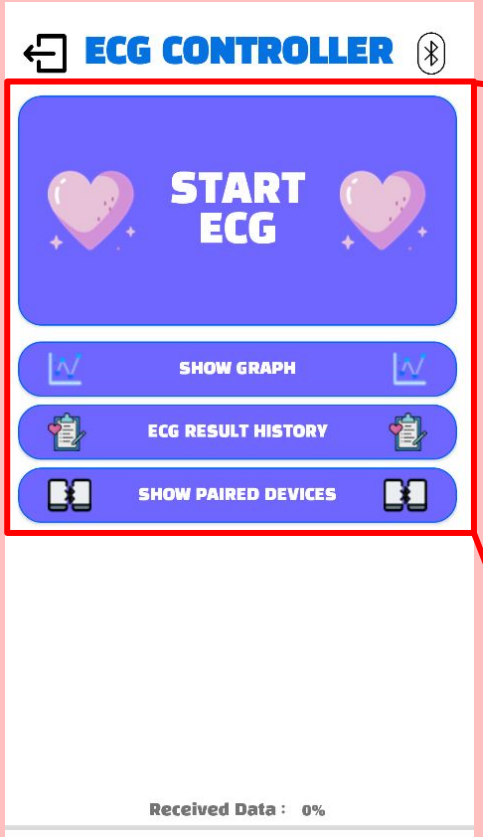
Setting Image View

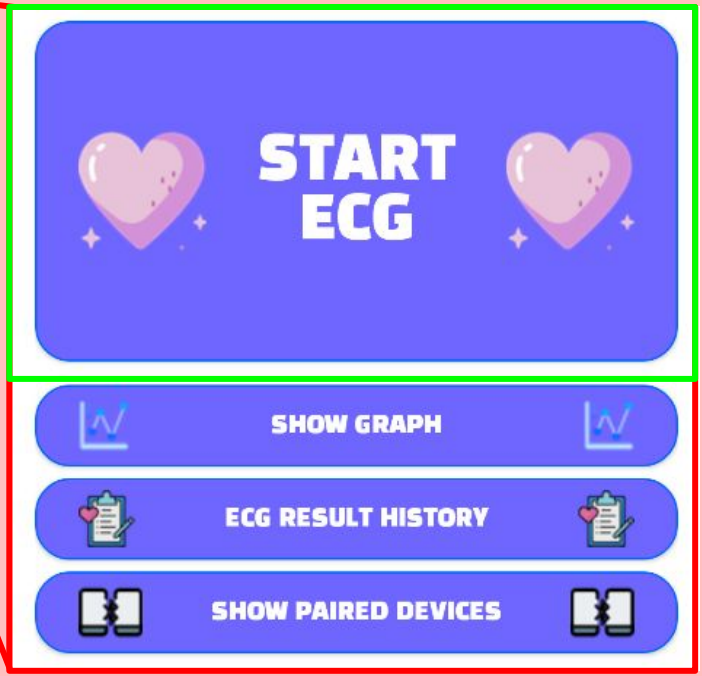
: Bluetooth Off

: Bluetooth On

: Pairing Success

MainActivity

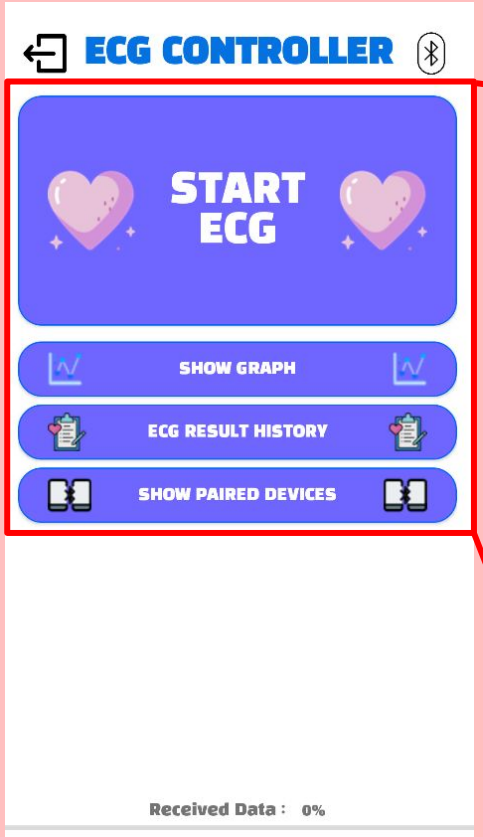


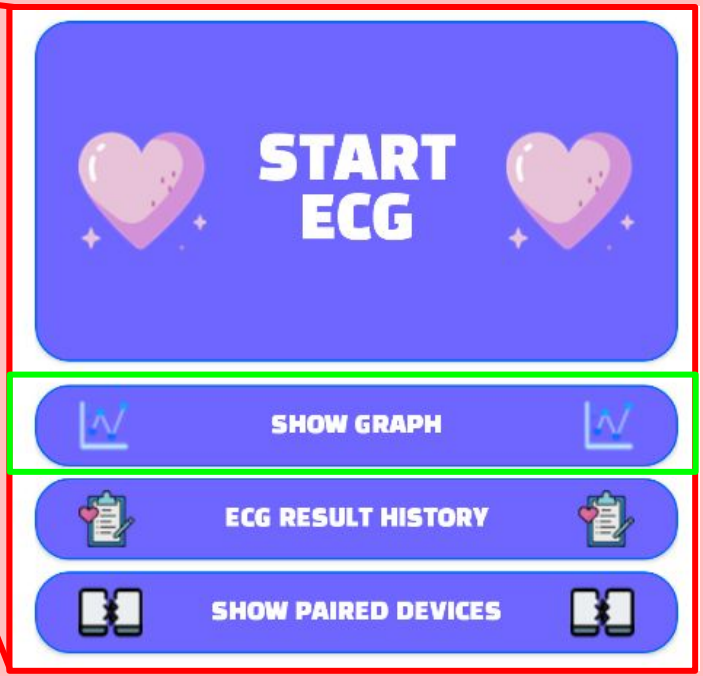


Start ECG (Button)

① Get Location Access Permission
② Start getting data from arduino

MainActivity



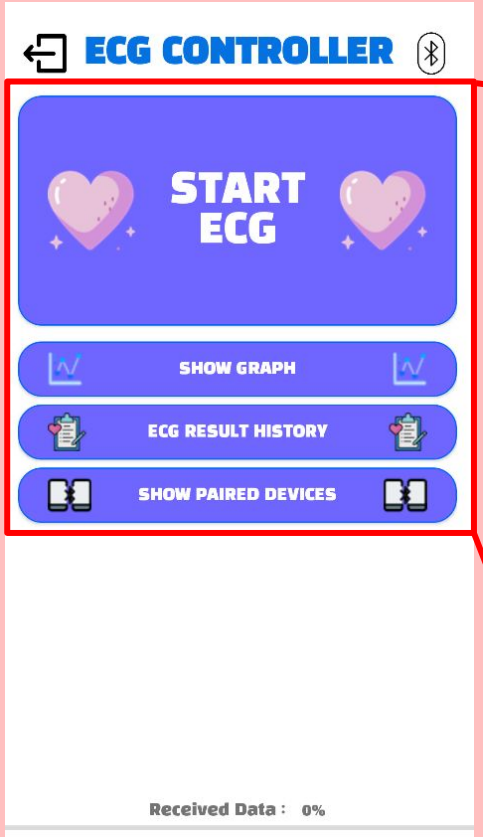


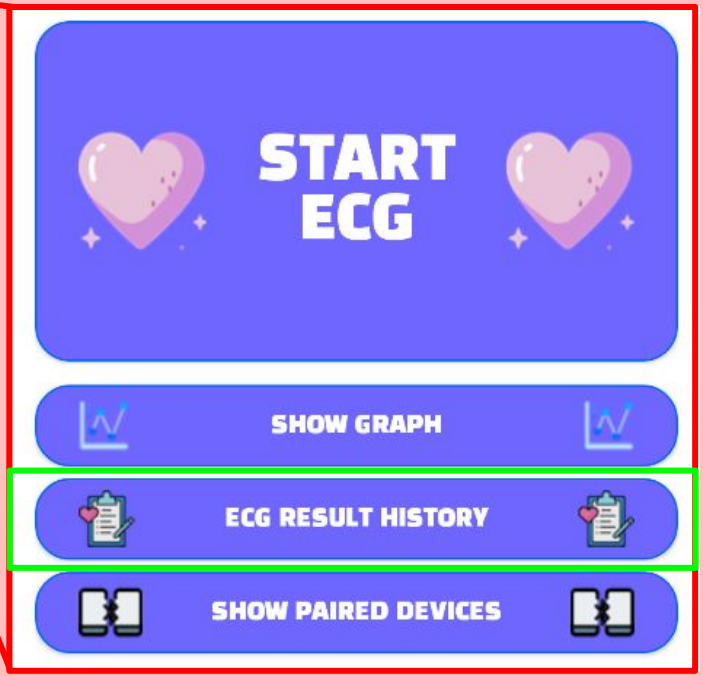
Show Graph (Button)

(After receiving data from arduino)

- Transmit the result to make the chart
- Go to another activity (Use intent)

MainActivity

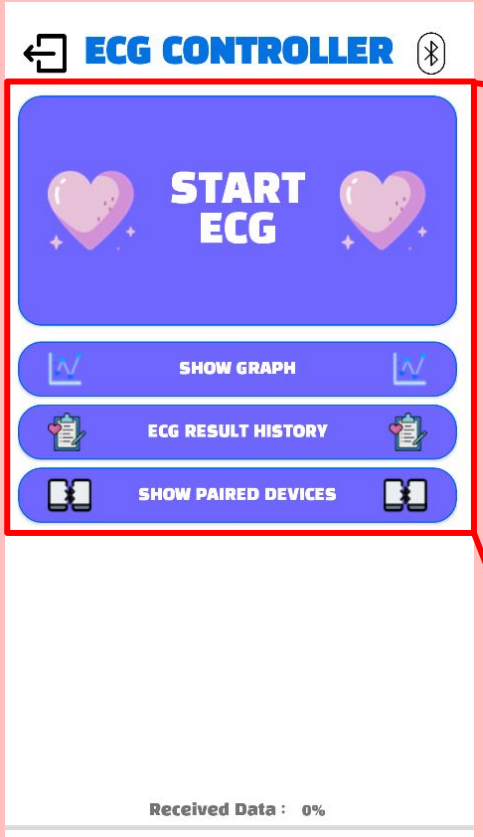


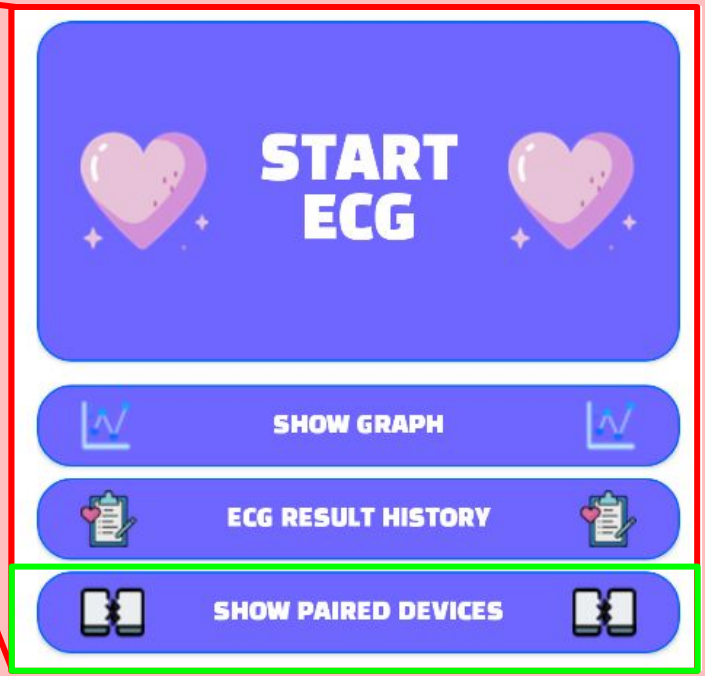


ECG Result History (Button)

- Go to another activity (Use intent)
- Get the user's previous data from database

MainActivity





Show Paired Device (Button)

- Only clickable when BT turned on
- Pairable devices shown (Use Broadcast Receiver)
- Click specific remote device → Pair Success

ECG Result Activity

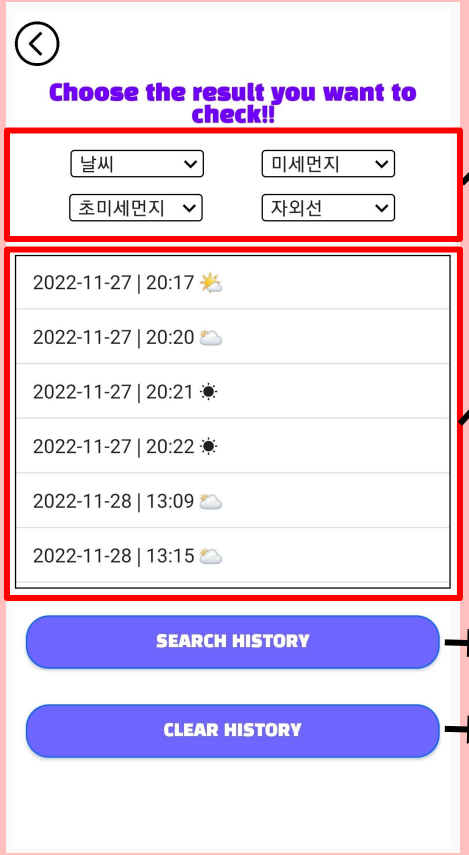
Result Chart (LineChart)

(※ Refer to PhilJay/MPAndroidChart)

Save (Button)

· By pressing Save Button, Not only measured data, but also data-collected time and location are transmitted and stored at the server

ECG History Activity



Weather Filter (Spinner)

History List (ListView)

Search History (Button)

Clear History (Button)

(※ Temporarily remove data)

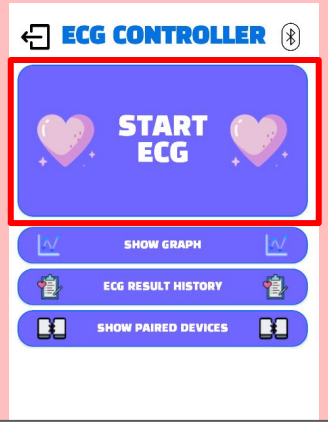
· When Search History Button Clicked, user's all previous results are displayed.

· User can filter the specific results according to the weather conditions that he/she has selected.

Weather information

Android Studio provides classes accessing the system location services

(Location, LocationManager, LocationListener)



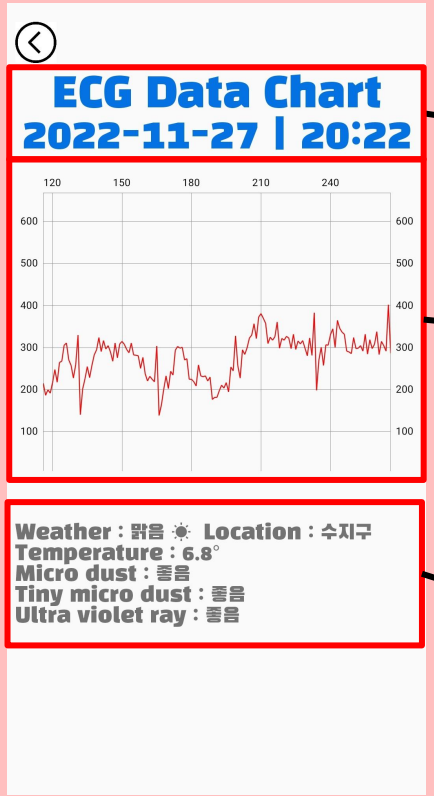
Allow access to manage device location

Measured data & Location Information transferred to the server

Location Information (Latitude / Longitude)

Converted into Korean address

ECG History Graph Activity



Title/Date (TextView)

Result Chart (LineChart)

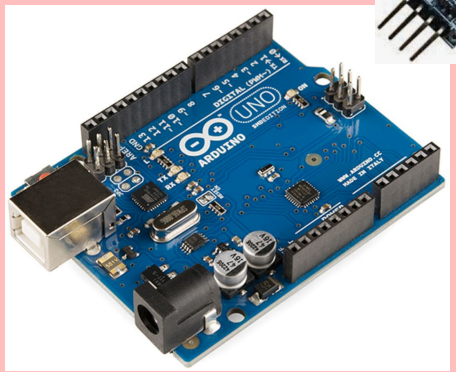
Weather (TextView)

· Get the weather information of the location of selected data from the server and display it on the screen

※ Weather information were retrieved by using ‘Web Crawling’

Bluetooth Connection

HC06



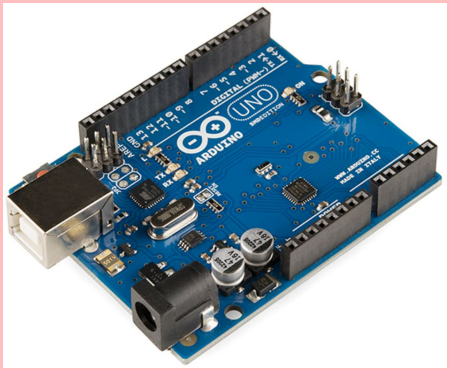
Arduino UNO

How to transfer the data?

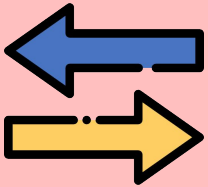
Application

Bluetooth Connection

Complete the initial setting of HC06



Arduino UNO



HC06

UART Communication

1. Make reverse of the pin numbers of Tx & Rx in Arduino and those in HC06

(At Serial Monitor)
2. Set BT module name & PIN Number using ‘AT’ Command

3. Activate BT and check the connection between HC06 & the device

Bluetooth Connection

Client

Application

Bluetooth Adapter

+Adapter View

Bluetooth Socket

BroadcastReceiver

Thread

Handler

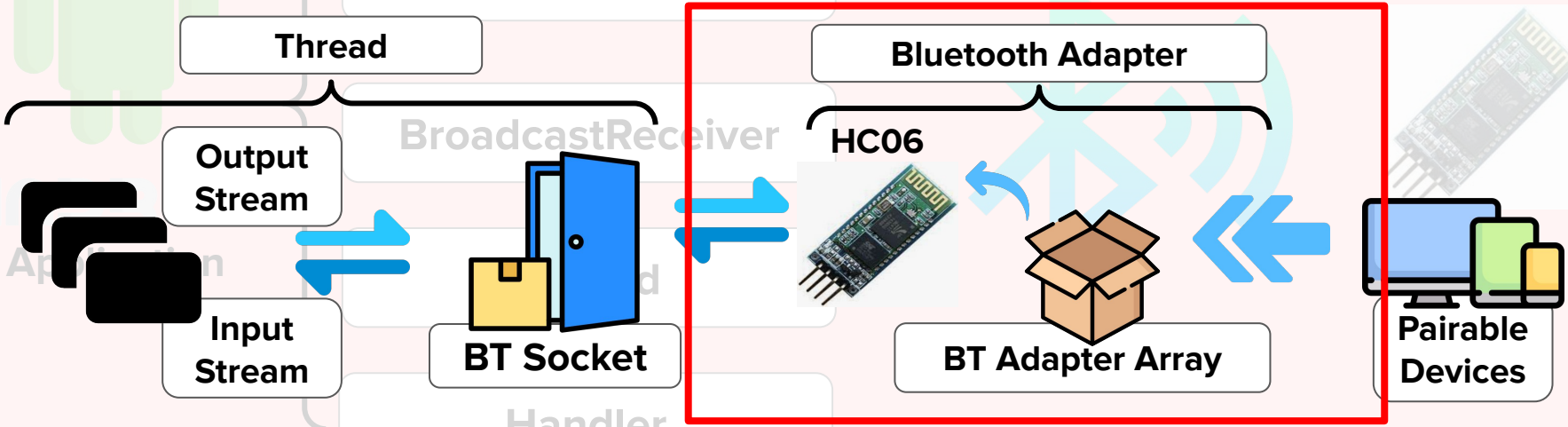
Server

HC06

Bluetooth Connection

Bluetooth Adapter

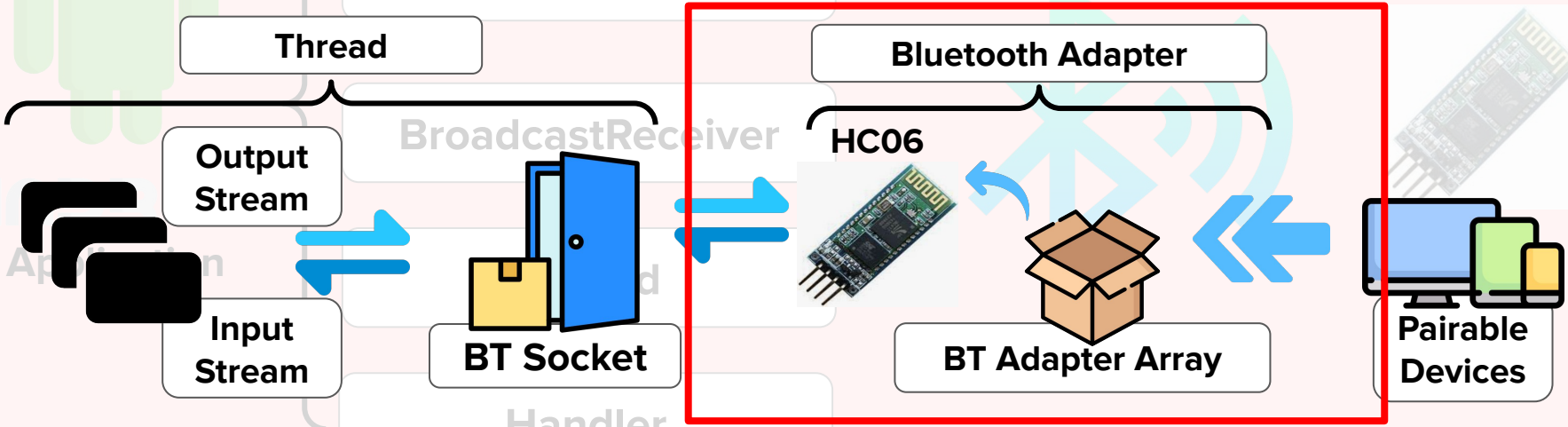
- Basically android studio contains ‘Bluetooth’ class
- Help to utilize the fundamental Bluetooth functions



Bluetooth Connection

BT Adapter Array

- Array for storing the information of pairable devices
- When you activate BT of your device,
it receives the data of each pairable device



Bluetooth Connection

BroadcastReceiver

It receives any change about Bluetooth events happening in the app

➜ Define the tasks depending on the given actions

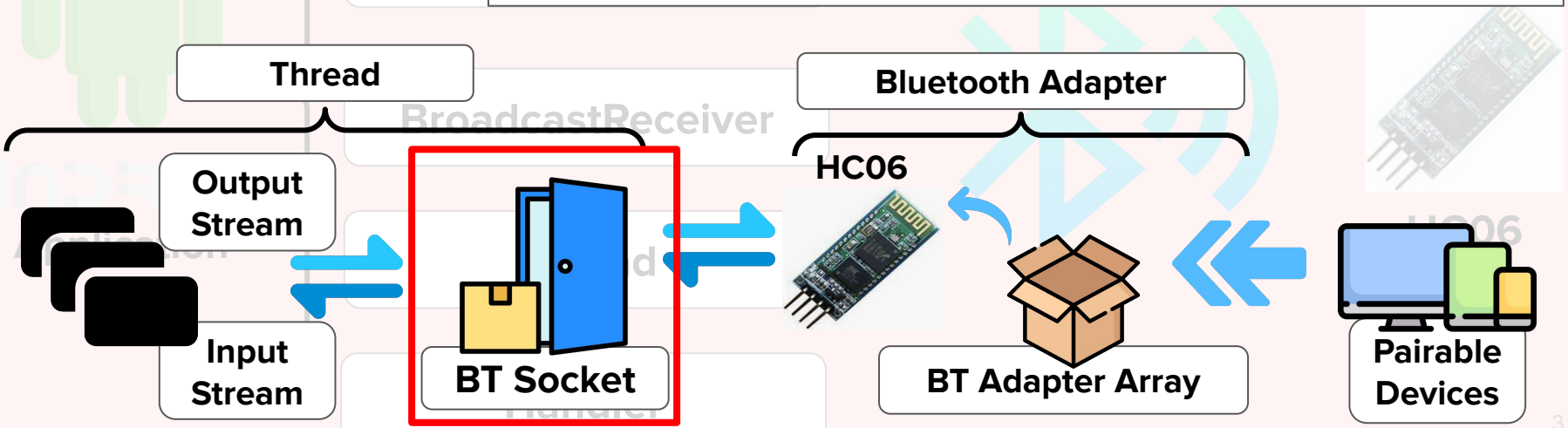
```
// 블루투스 이벤트 관련 변경사항을 전달받아 주어진 액션에 따라 수행할 기능을 정의한다.
final BroadcastReceiver blReceiver = (context, intent) -> {
        String action = intent.getAction();
        if(BluetoothDevice.ACTION_FOUND.equals(action)){
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            // add the name to the list
            mBTArrayAdapter.add(device.getName() + "\n" + device.getAddress());
            mBTArrayAdapter.notifyDataSetChanged();
        }

        if(action.equals(BluetoothAdapter.ACTION_STATE_CHANGED)) {
            if (mBTAdapter.getState() == BluetoothAdapter.STATE_ON) {
                Toast.makeText(getApplicationContext(), text: "Bluetooth on",Toast.LENGTH_SHORT).show();
                btImg.setImageResource(R.drawable.bton);
            }
            if (mBTAdapter.getState() == BluetoothAdapter.STATE_OFF) {
                btImg.setImageResource(R.drawable.btoff);
                Toast.makeText(getApplicationContext(), text: "Bluetooth offed",Toast.LENGTH_SHORT).show();
                // The user bluetooth is already disabled.
                return;
            }
        }
```

Bluetooth Connection

Bluetooth Socket

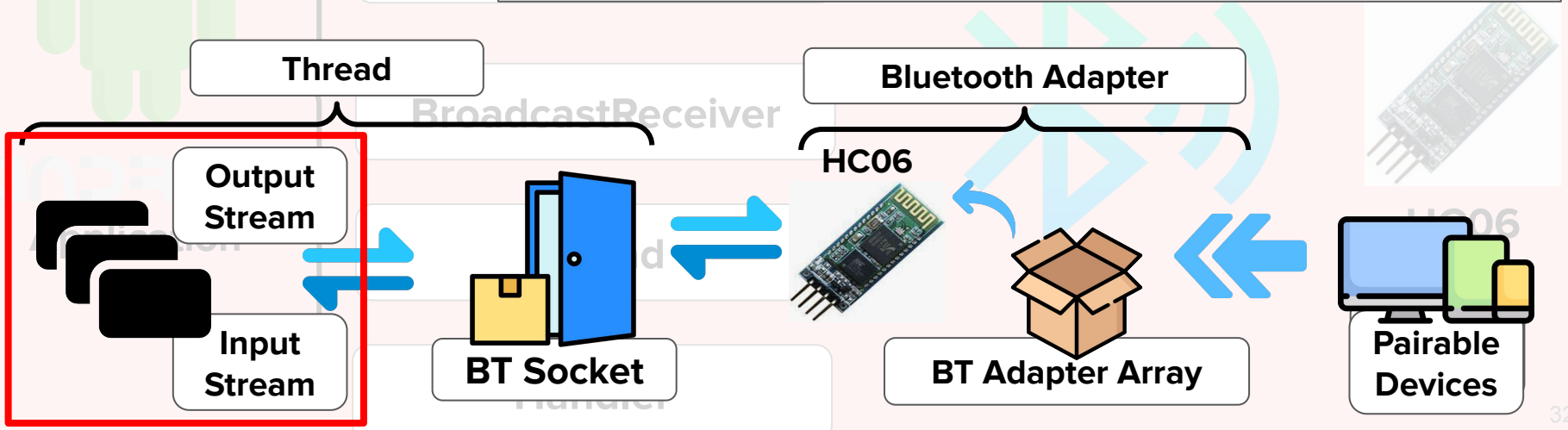
- Interface for the bluetooth communication
- Connection is created through the socket
- After the connection is created,
Transmission & Receipt becomes available (like TCP)



Bluetooth Connection

Input / Output Stream

- Through the socket, BT communication is available
- Input Stream : Sequence of received data (Kind of Buffer)
- Output Stream : Sequence of data to be transmitted (Same as above)



Bluetooth Connection

- Through the socket, BT communication is available
- Input Stream : Sequence of received data (Kind of Buffer)
- Output Stream : Sequence of data to be transmitted

Input / Output Stream

We are going to deal with these data
(Transferred from Arduino UNO
through BT communication)

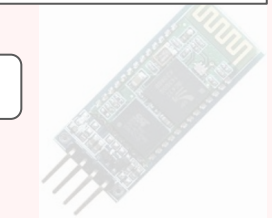
Thread

Output Stream

Input Stream

BT Socket

BT Adapter Array



Pairable Devices

Bluetooth Connection

Client

Thread

- The segment of a process (Program being executed in the main memory)
- One process has one or more threads
- Main Activity in Android Studio
  ➔ One process executed with a single thread (Default)

Application

ceiver

Handler

HC06

Want more various functions to be executed at background?

Create and define more threads on your own!

Bluetooth Connection

Problems with multi threads

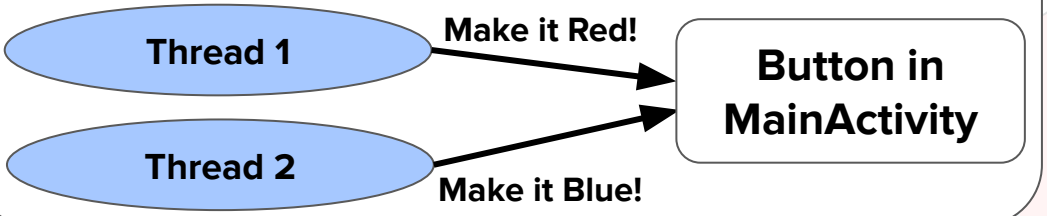
We can do more with many threads!!

Then how would you HANDLE those threads?

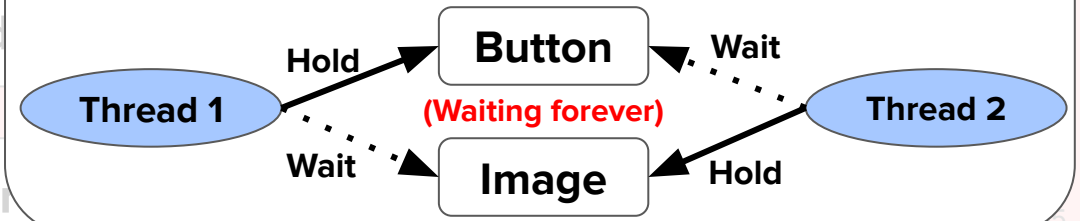
Race Condition

Simultaneously

: 2 or more threads try to access shared resource



DeadLock : 2 or more threads hold certain resource respectively and waiting for other thread to release the lock of their resource



Bluetooth Connection

Handler

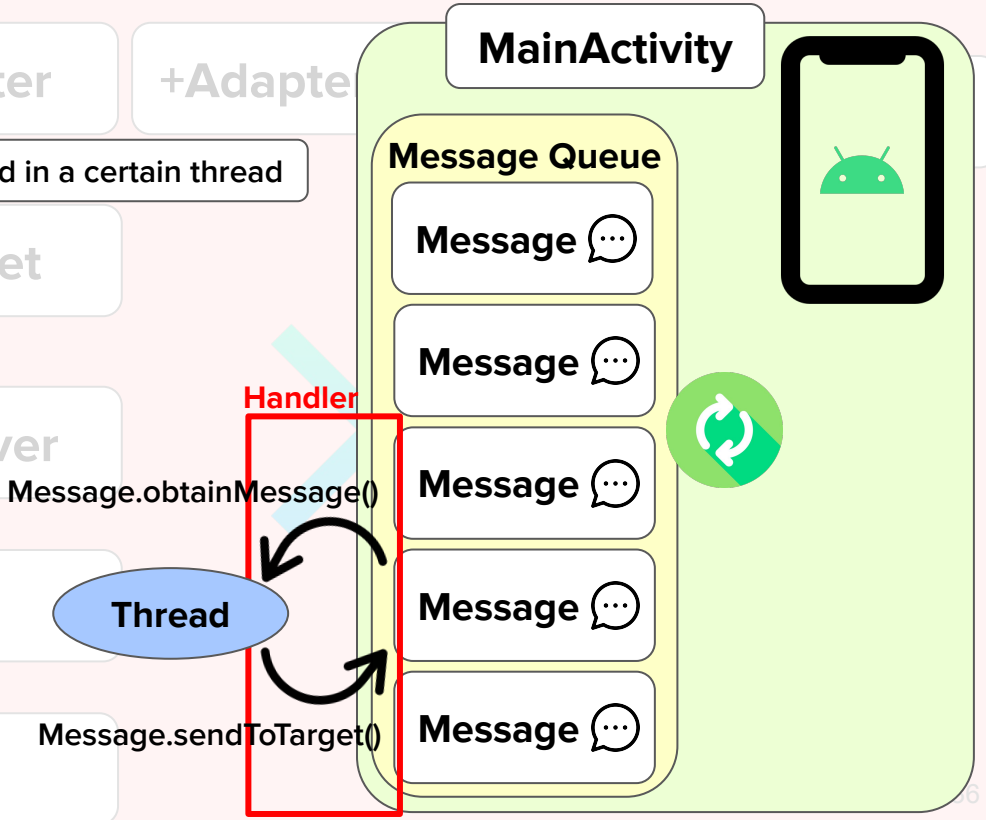
(Message, Runnable object)

A queue composed of **tasks** that will be executed in a certain thread

- 1 Message Queue is assigned to 1 Thread

- Send & process message or runnable object associated with the thread (Tasks represented in thread should be conducted **through the handler**)

- To avoid the undesirable problems in the previous slide, 1 handler to 1 thread is necessary



Data Transfer

Byte Array

Transmission is ONLY available using byte array

Arduino IDE

```
#include <SoftwareSerial.h>

int Tx = 6; //전송 보내는핀  8
int Rx = 7; //수신 받는핀
int count = 512; //테스트 횟수
SoftwareSerial bluetooth(Tx,Rx);
String A;
int tmp1, tmp2, tmp3;
byte ecg_data[1024];
```

```
tmp1 = (tmp2 & 0xff00);
tmp2 = (tmp2 & 0x00ff);

ecg_data[count*2] = (tmp1>>8)+1;
ecg_data[count*2+1] = tmp2+1;

delay(20);
count++;
```

Android Studio

```
// 실제 블루투스로 측정값을 가져오는 인터페이스 부분
public void run() {
    // 넘어오는 측정값을 받기 위한 버퍼 (buffer store for the stream)
    byte[] buffer = new byte[1024];
    int bytes; // bytes returned from read()
    // 예외 발생할 때까지 넘어오는 측정값을 받을 준비를 하고 있는다
    // Keep listening to the InputStream until an exception occurs
    while (true) {
        try {
            // Read from the InputStream
            bytes = mmInStream.available();
```

Data Transfer

How to transfer data with this data type?

Data Stream : [124, 235, 82, 351, 455, 92, 90, 43, ...]

1 Byte = 8 bit → (0 ~ 2^8 - 1)

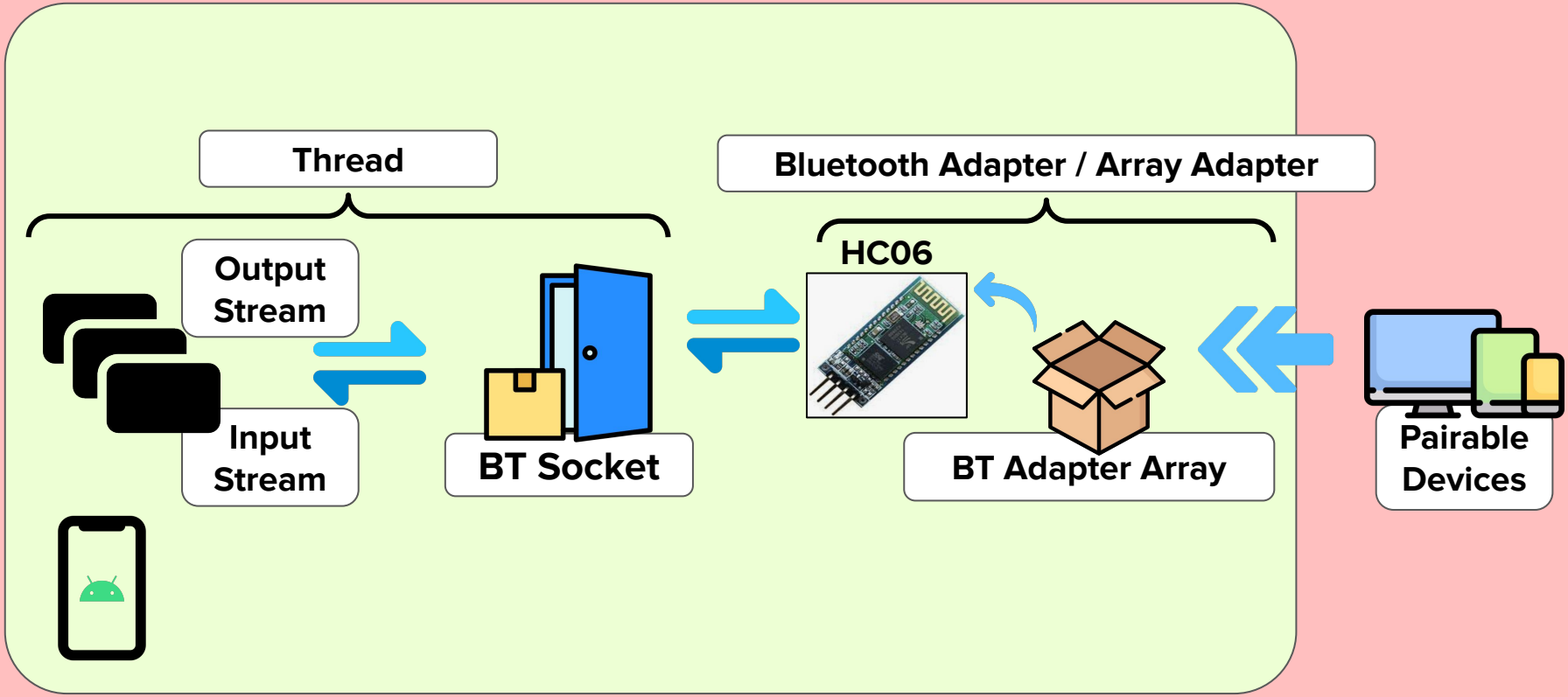
2 bytes are needed to represent value over 255

Value > 255 : n * 2^8 + m
0 ≤ Value ≤ 255 : 0 * 2^8 + m
(n, m : consecutive 2 elements in byte array)

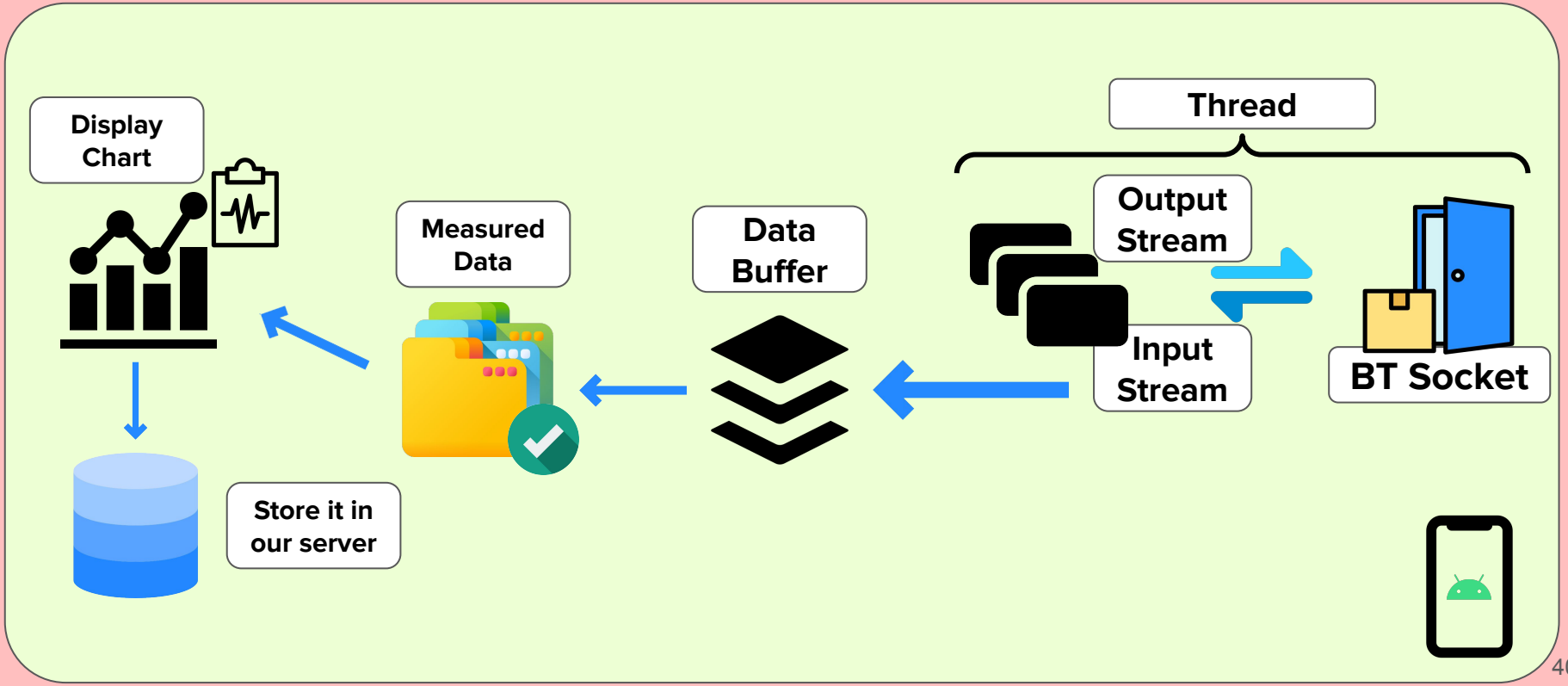
```
// 블루투스 핸들러 동작 알고리즘 (ECG 키트에서 안드로이드 앱으로 데이터 수신하는 알고리즘)
mHandler = (Handler) handleMessage(msg) → {
        // START ECG 버튼을 누르면 키트로부터 데이터를 읽어온다
        if (msg.what == MESSAGE_READ) {
            // Back GettingData to START ECG
            mECGStartBtn.setText("START ECG");
            mECGStartBtn.setBackgroundResource(R.drawable.btn_blue);
            // 측정 데이터 가져오기 전 초기화 (Initialize Data Variable of ECG)
            int tmpNum = 0, tmpNum1 = 0, tmpNum2 = 0;

            /*
            측정 데이터 값의 범위가 0 ~ 1000 이다.
            하지만 블루투스를 통해서 데이터를 넘기기 위해서는 Byte 단위로 전달해야하는데,
            1 byte = 8 bit 이므로, 0 ~ 255 까지만 표현이 가능하다.
            따라서 256 ~ 1000 까지의 데이터 또한 표현 가능하도록 해야 측정 데이터의 그래프 표현이 가능해지므로
            1 byte 두 부분을 받아서 값을 원래 측정된 값으로 다시 변환하여 앱 상 데이터 목록에 추가한다.
            */
            for (int i = 0; i < 512; i++) {
                tmpNum1 = ((byte[]) msg.obj)[i * 2] & 0xff;
                tmpNum2 = ((byte[]) msg.obj)[i * 2 + 1] & 0xff;
                if (tmpNum1 != 0) {
                    tmpNum = ((tmpNum1 << 8) & 0xff00) + (tmpNum2 & 0x00ff);
                    final_data.add(tmpNum - 256);
                } else {
                    sum += i;
                    System.out.println(sum);
                    mProgressBar.setProgress(sum);
                    mProgressData.setText(Integer.toString( i: sum * 100 / 512) + "%");
                    mShowGraphBtn.setClickable(true);
                    break;
                }
            }
```

Data Transfer



Data Transfer



Basic Concept

django : OpenSource Python Web framework
(Support common Web Development needs)

- MVT Pattern
  - Model : Hold data interacting with database
  - View : Handle HTTP Request/Response directly and work as a bridge of Model & Template
  - Template : provide user interface

➡ **HOWEVER**, we will use it as a web server by using **REST API**

Basic Concept

REST API : helps Django to be implemented as either client or server

↳ Replace the role of 'View'

(Handle responses directly with HTTP request 'method' (Parameter) C.R.U.D.)

C.R.U.D. = Create / Read / Update / Delete

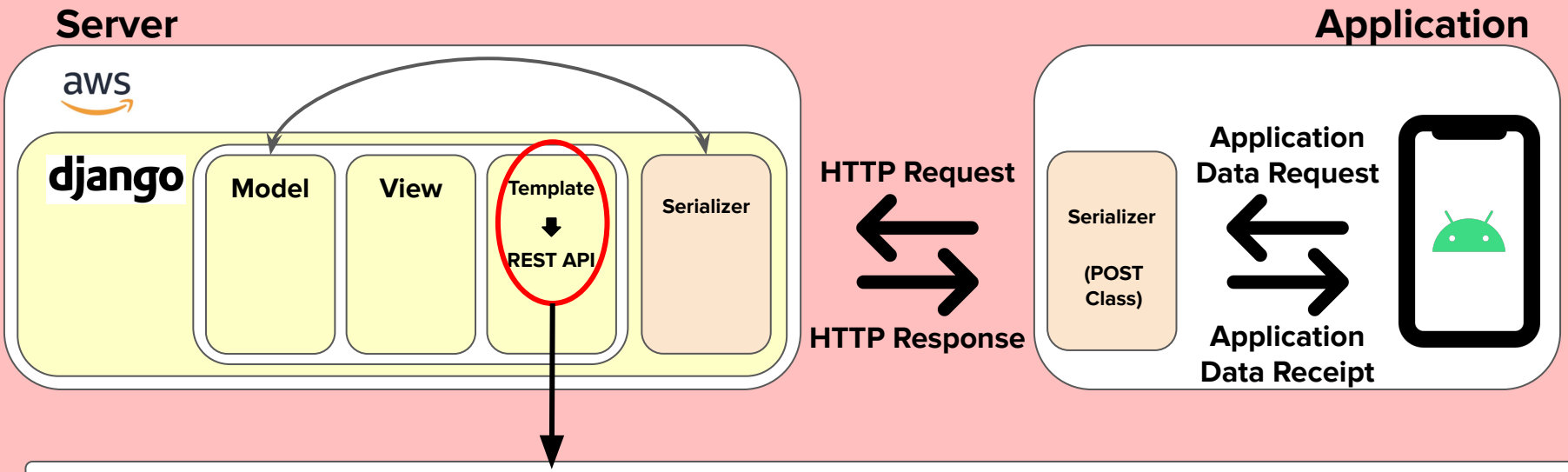
- POST(Create) : Store data at the server

- GET(Read) : Return the requested data

- PUT(Update) : Update data

- DELETE(Delete) : Delete data

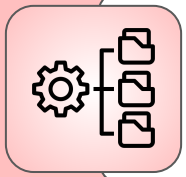
Overall Data Flow



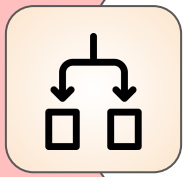
Thanks to REST API, we can observe the logic and entire data stored in our web server through template

※ Serializer : Change the form of data into JSON file and conduct transfer through HTTP protocol

Lessons from this project

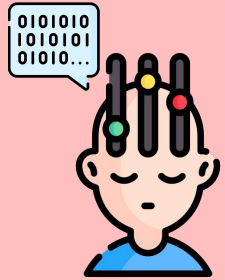


1. Importance of sophisticated design



2. (At most) 1 event assignment per function

3. Necessity of searching skill for valid information



Thank you