좋아하면 울리는
기초모바일실험2 프로젝트

타조: 32210799 김수진
32211623 박민주
32215080 박유빈
32211768 박윤아
32212398 심예린

역할 소개

아두이노
H/W

박윤아

백엔드
데이터베이스

김수진
박유빈

박민주
심예린

프론트엔드
UI

목차

개발 동기

저희는 수업시간에 배운 ECG와 심장박동을 의학적인 부분이 아닌, 대학생들이 쉽게 흥미를 가질 수 있을만한 주제와 연관지어 어플을 개발하고 싶다는 생각이 들었고, 심박수를 이성과 만나는 때에 활용하는 상황을 떠올리게 되었습니다.

어플 주요 기능

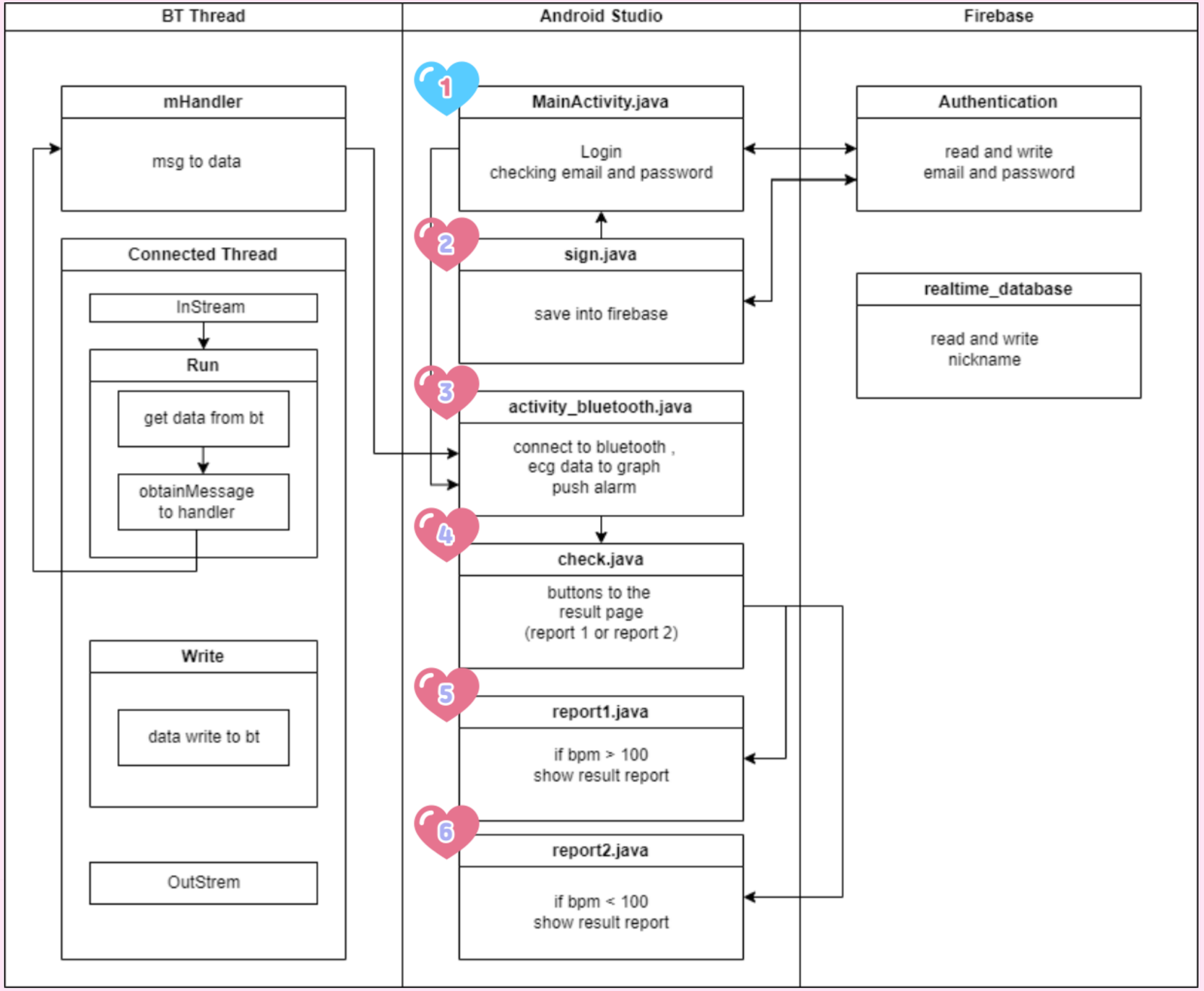
'좋아하면 울리는' 어플은 심박수가 일정 수준 이상 올라가면 상대방에게 호감을 느꼈다고 판단하여 이를 알림을 통해 확인시켜주는 것이 가장 큰 기능입니다.

* 의료용 목적이 아니기 때문에 진지하게 받아들이지 말 것

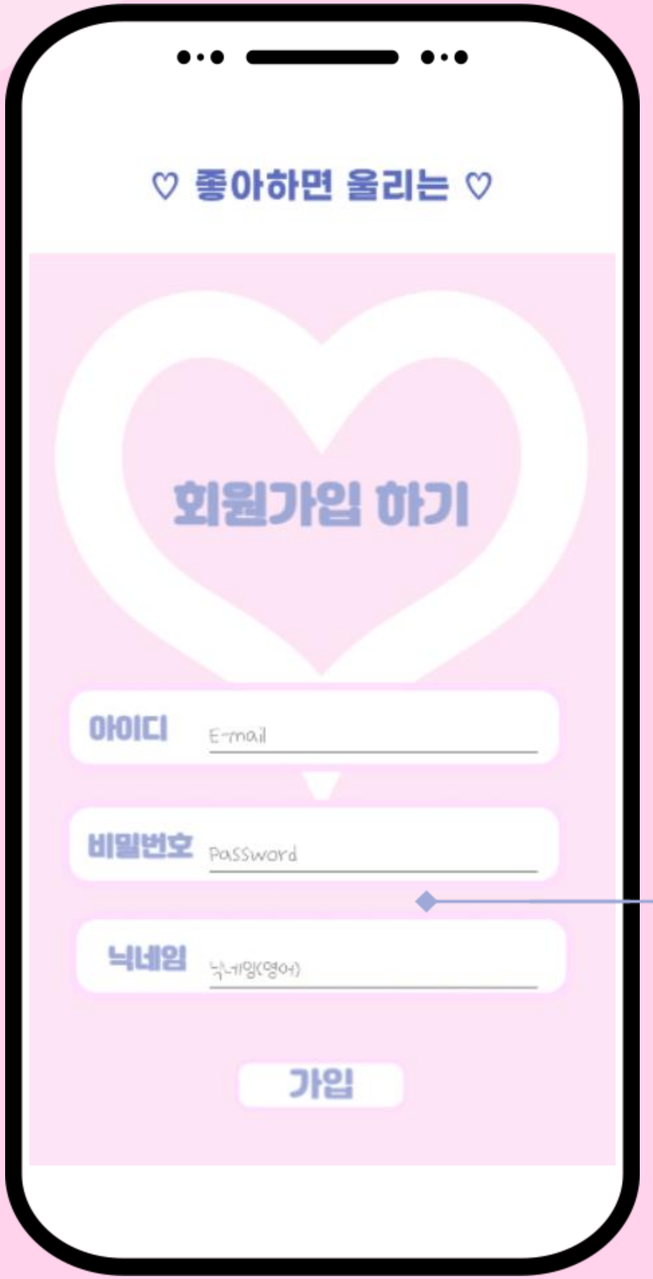
웹툰 '좋아하면 울리는'

어플 이름은 웹툰 '좋아하면 울리는'의 제목에서 따와 정하였습니다. 원작에 나오는 '좋알람' 어플은 호감을 가지는 사람이 반경 10m 안에 들어오면 알림이 울리는 기능을 가지고 있습니다. 저희는 원작 속 '좋알람' 어플에서 '**호감가는 이성**'과 '**알림**'이라는 키워드를 따와 기능을 구성해보았습니다.

UI 및 흐름 소개


BT Thread
Android Studio
Firebase
mHandler
msg to data
Connected Thread
InStream
Run
get data from bt
obtainMessage
to handler
Write
data write to bt
OutStrem
1
MainActivity.java
Login
checking email and password
2
sign.java
save into firebase
3
activity_bluetooth.java
connect to bluetooth ,
ecg data to graph
push alarm
4
check.java
buttons to the
result page
(report 1 or report 2)
5
report1.java
if bpm > 100
show result report
6
report2.java
if bpm < 100
show result report
Authentication
read and write
email and password
realtime_database
read and write
nickname

UI #1 #2

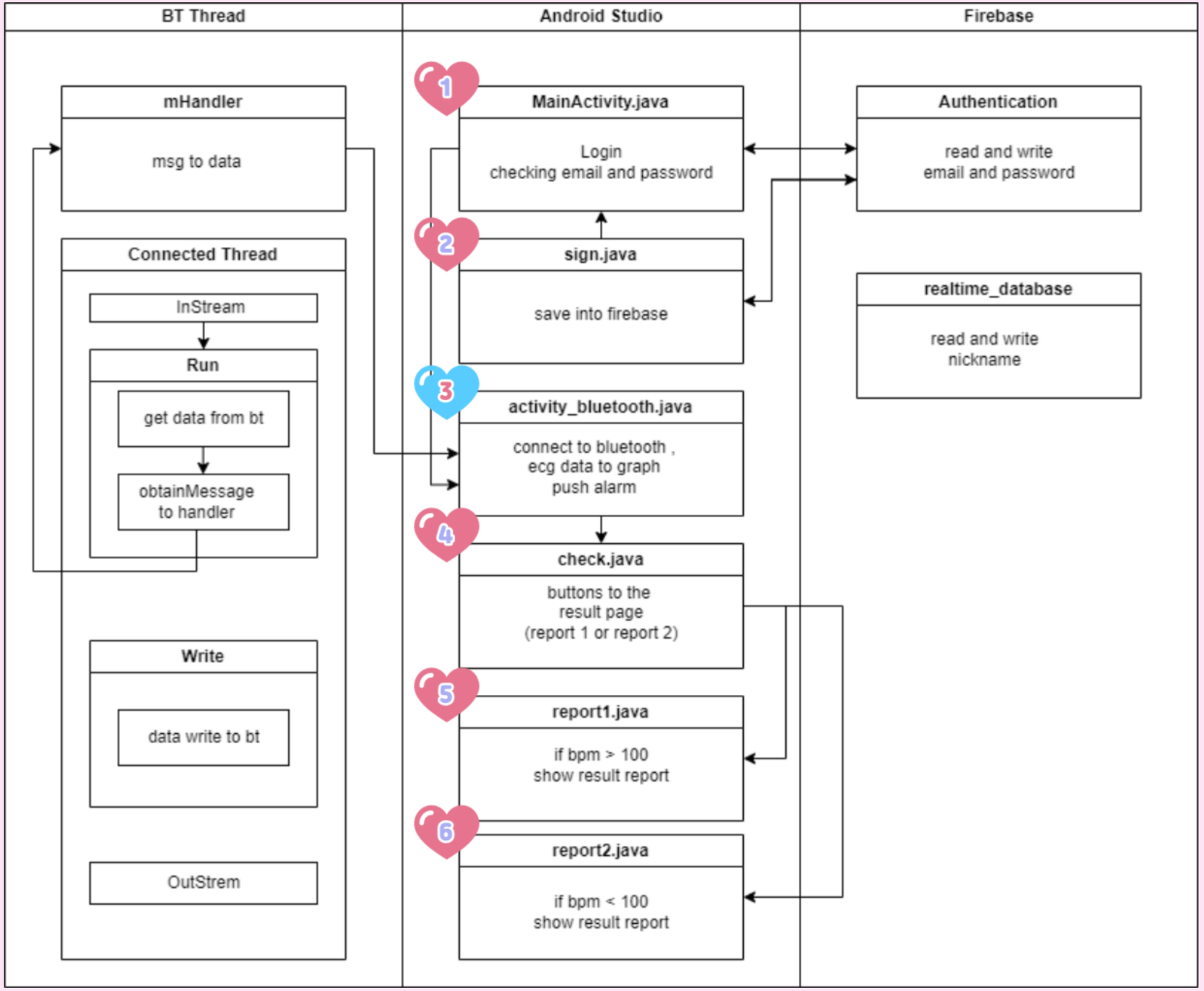


기존 회원은
아이디와 비밀번호를
입력하여 로그인

회원이 아닐 경우
회원가입

아이디와 이메일, 닉네임을
입력받아
회원가입 진행

UI 및 흐름 소개


BT Thread
Android Studio
Firebase
mHandler
msg to data
Connected Thread
InStream
Run
get data from bt
obtainMessage
to handler
Write
data write to bt
OutStrem
1
MainActivity.java
Login
checking email and password
2
sign.java
save into firebase
3
activity_bluetooth.java
connect to bluetooth ,
ecg data to graph
push alarm
4
check.java
buttons to the
result page
(report 1 or report 2)
5
report1.java
if bpm > 100
show result report
6
report2.java
if bpm < 100
show result report
Authentication
read and write
email and password
realtime_database
read and write
nickname

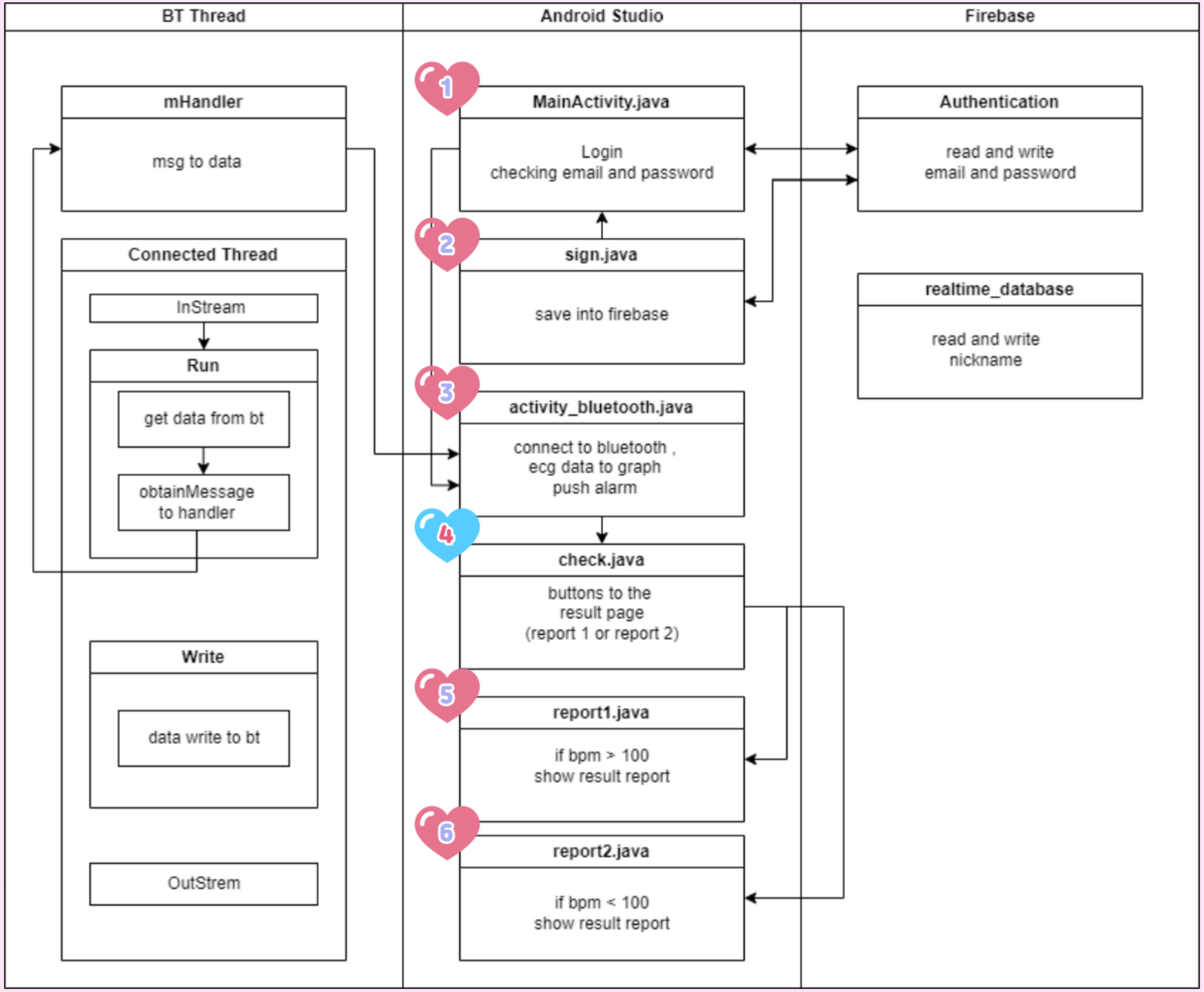
UI #3

블루투스 연결
+
심전도 그래프 출력

회원가입시 입력받은
닉네임 정보 불러옴

BPM >=100
count 값 푸시 알림

UI 및 흐름 소개


BT Thread
Android Studio
Firebase
mHandler
msg to data
Connected Thread
InStream
Run
get data from bt
obtainMessage
to handler
Write
data write to bt
OutStrem
1
MainActivity.java
Login
checking email and password
2
sign.java
save into firebase
3
activity_bluetooth.java
connect to bluetooth ,
ecg data to graph
push alarm
4
check.java
buttons to the
result page
(report 1 or report 2)
5
report1.java
if bpm > 100
show result report
6
report2.java
if bpm < 100
show result report
Authentication
read and write
email and password
realtime_database
read and write
nickname

UI #4 #5 #6

[report 1]
BPM> =100

[report 2]
BPM< 100

클릭 시 화면 이동

구현 세부과정 설명

파이어베이스

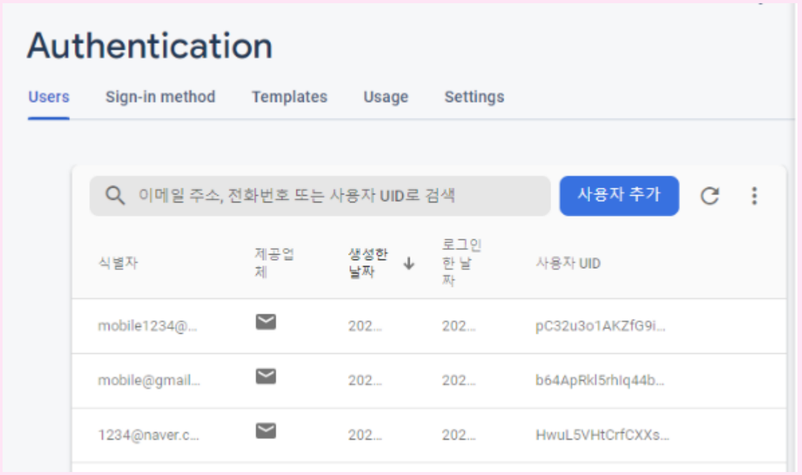
핸들러&스레드

블루투스

그래프

푸시 알림

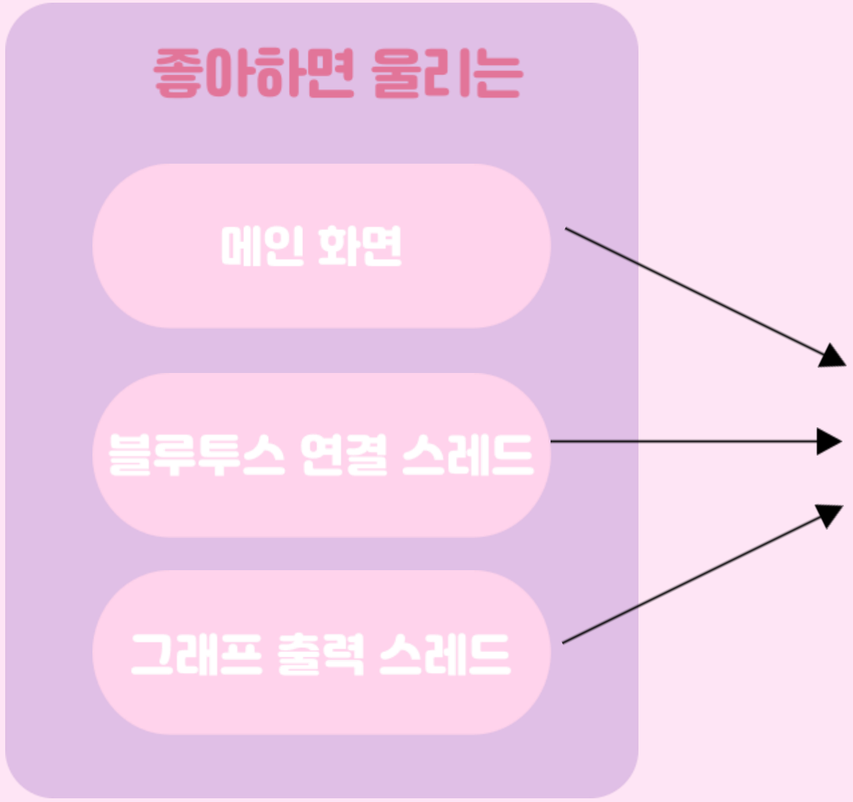
파이어베이스




```java
package com.example.loving;

public class table{
    String name;

    public table(){};

    public String getname() {
        return name;
    }

    public void setname(String name) { this.name = name; }


    public table(String name) { this.name = name; }
}
```

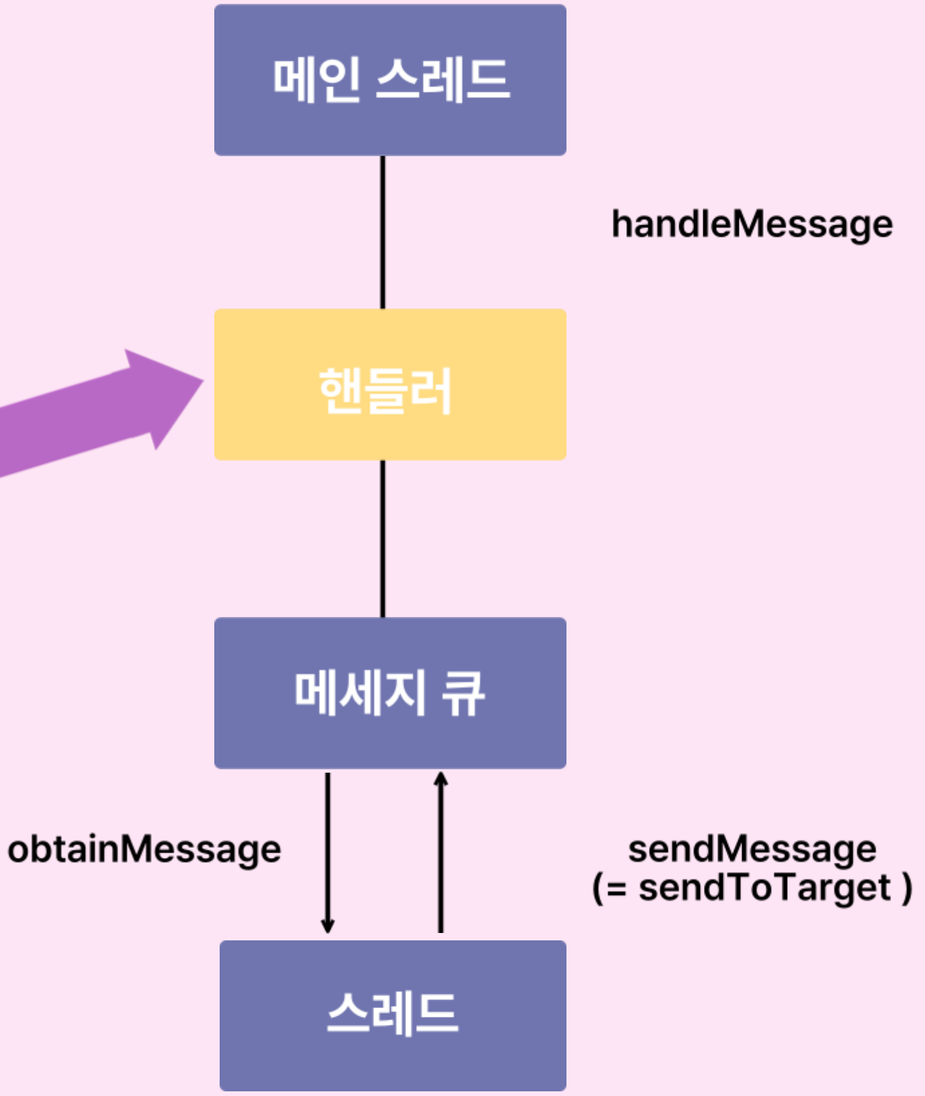
```java
//닉네임 출력
private void readUser(String uid){
    mDatabase.child("zoo").child(uid).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            table table = dataSnapshot.getValue(table.class);
            name_=table.getname();
            name.setText(name_);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText(getApplicationContext(), text: "데이터를 가져오는데 실패했습니다" , Toast.LENGTH_LO
        }
    });
}
```

핸들러&스레드



UI
데드락 발생 !

문제 해결



데드락(Dead Lock) : 동시에 어떤 리소스에 접근을 했을 때,
시스템이 어떤 스레드의 작업을 먼저
처리할지 몰라 작업을 멈추게 되는 상황

핸들러&스레드

obtainMessage() : 호출의 결과로 메시지 객체 를 리턴 받게함

sendMessage() : Message Queue 에 넣음

```
new Thread() //블루
{
    @SuppressLint("MissingPermission")
    public void run() {
        boolean fail = false;

        BluetoothDevice device = mBTAdapter.getRemoteDevice(address);

        try {
            mBTSocket = createBluetoothSocket(device);
        } catch (IOException e) {
            fail = true;
            Toast.makeText(getBaseContext(), text: "소캣 생성 실패", Toast.LENGTH_SHORT).show();
        }
        // Establish the Bluetooth socket connection.
        try {
            mBTSocket.connect();
        } catch (IOException e) {
            try {
                fail = true;
                mBTSocket.close();
                mHandler.obtainMessage(CONNECTING_STATUS, arg1: -1, arg2: -1)
                        .sendToTarget();
            } catch (IOException e2) {
                //insert code to deal with this
                Toast.makeText(getBaseContext(), text: "소캣 생성 실패", Toast.LENGTH_SHORT).show();
            }
        }
```

handleMessage
자동호출

```
// 받은 값 String 형태로 보여주기, status에 연결된 장치 보여주기
mHandler = new Handler() { //handler:서로 다른 thread간의 통신을 위한 장치
    public void handleMessage(Message msg) {
        if (msg.what == MESSAGE_READ) { //MESSAGE_READ = 2 => to identify message
            int retValue =0;
            retValue = ((byte[]) msg.obj)[0]&(0xFF);
            if(retValue>=100) {
                count++;
                if(count/20>0){
                    if(count%20==0){
                        createNotificationChannel(DEFAULT, channelName: "default chan
                        createNotification(DEFAULT, id: 1, title: "좋아하면 울리는",("당신
                    }
                }
            }
            ECG.add(retValue);

        }
```

블루투스

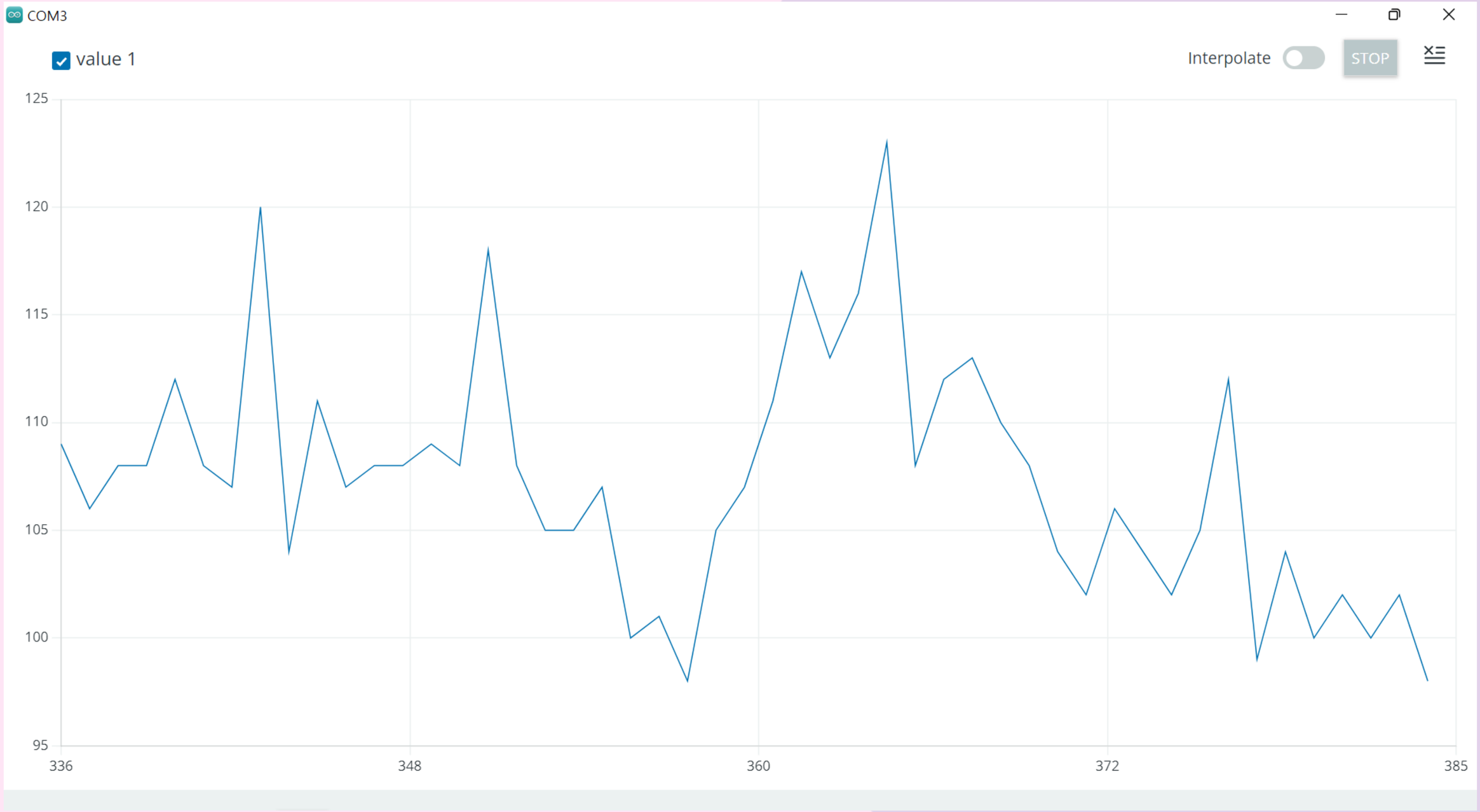
블루투스 연결

HC-06 블루투스 모듈 사용


```
int BTreadV= 0;
void loop(){

  rvalue = analogRead(0) / 4; //write가 최대 256까지 보낼 수 있기 때문
  if(BTSerial.available()==true){ //블루투스로 신호를 받으면
    rvalue = analogRead(0) / 4;
    BTreadV = BTSerial.read();
    if(BTreadV==49){ // 그 신호가 1(49)인경우( 시작 버튼 누를 경우 )
      Serial.println("start");
      start = true;

    }
```

시작 : 1 전송

종료 : 0 전송


COM3
value 1
Interpolate
STOP
125
120
115
110
105
100
95
336
348
360
372
385

블루투스


```
if(start == false){
  empty = 500-count;
}


delay(100); //0.1초 간격으로 측정
```

데이터 전송

ECG 큐 에
데이터 저장

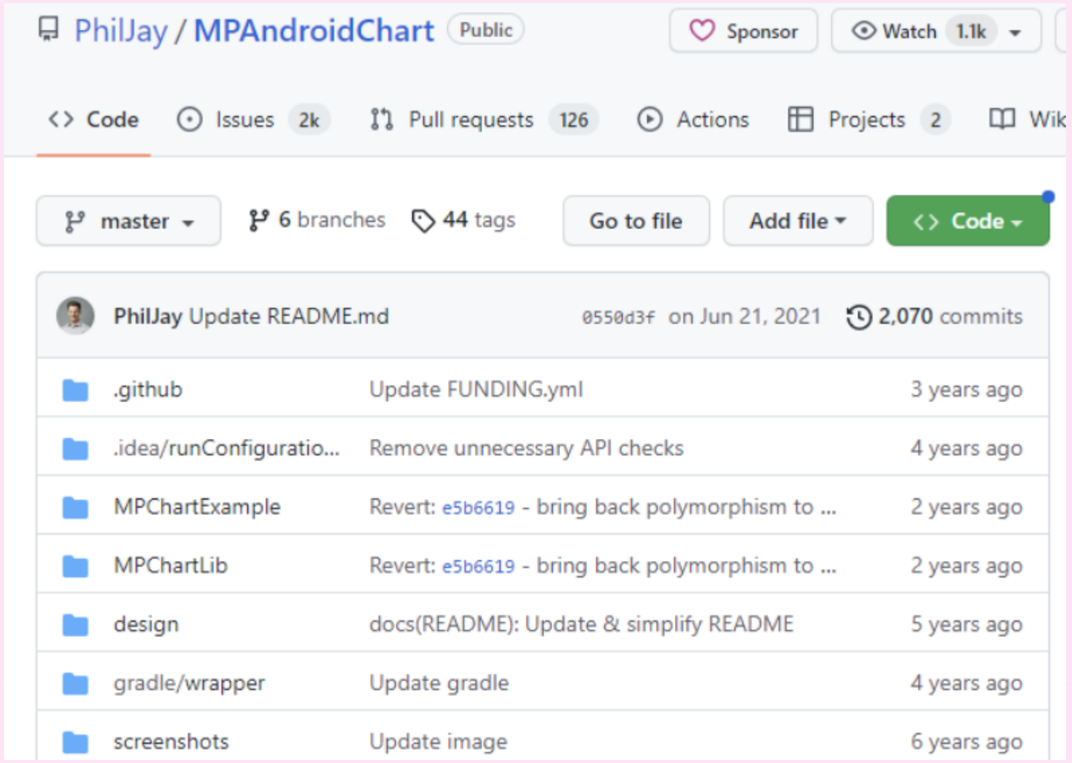
```
private Queue<Integer> ECG = new LinkedList<>();
```

큐에서 데이터를 하나씩 꺼내서 그래프를 출력!!

그래프 출력

MPAndroidChart
created by Philipp Jahoda

핸들러& 스레드




```
private Queue<Integer> ECG = new LinkedList<>();
```

```
// 받은 값 string 형태로 보여주기, status에 연결된 상태 보여주기
mHandler = new Handler() { //handler:서로 다른 thread간의 통신을 위한
    public void handleMessage(Message msg) {
        if (msg.what == MESSAGE_READ) { //MESSAGE_READ = 2 => to i
            int retValue =0;
            retValue = ((byte[]) msg.obj)[0]&(0xFF);
            if(retValue>=100) {
                count++;
                if(count/20>0){
                    if(count%20==0){
                        createNotificationChannel(DEFAULT, channelNa
                        createNotification(DEFAULT, id: 1, title: "졸아
                    }
                }
            }
            ECG.add(retValue);

        }
```

그래프 출력

```
private void addEntry() {
    LineData data = chart.getData();
    if (data != null) {
        ILineDataSet set = data.getDataSetByIndex(0);

        if (set == null) {
            set = createSet();
            data.addDataSet(set);
        }
        if(ECG.isEmpty()){
            data.addEntry(new Entry(set.getEntryCount(),  y: 60),  dataSetIndex: 0);
        }
        else {
            data.addEntry(new Entry(set.getEntryCount(), ECG.poll()),  dataSetIndex: 0);
            //ECG.poll() queue 첫번째 값을 반환

        }
        data.notifyDataChanged();

        chart.notifyDataSetChanged();
        chart.setVisibleXRangeMaximum(10);
        chart.moveViewToX(data.getEntryCount());
    }
}
```

기본 값 60으로 설정

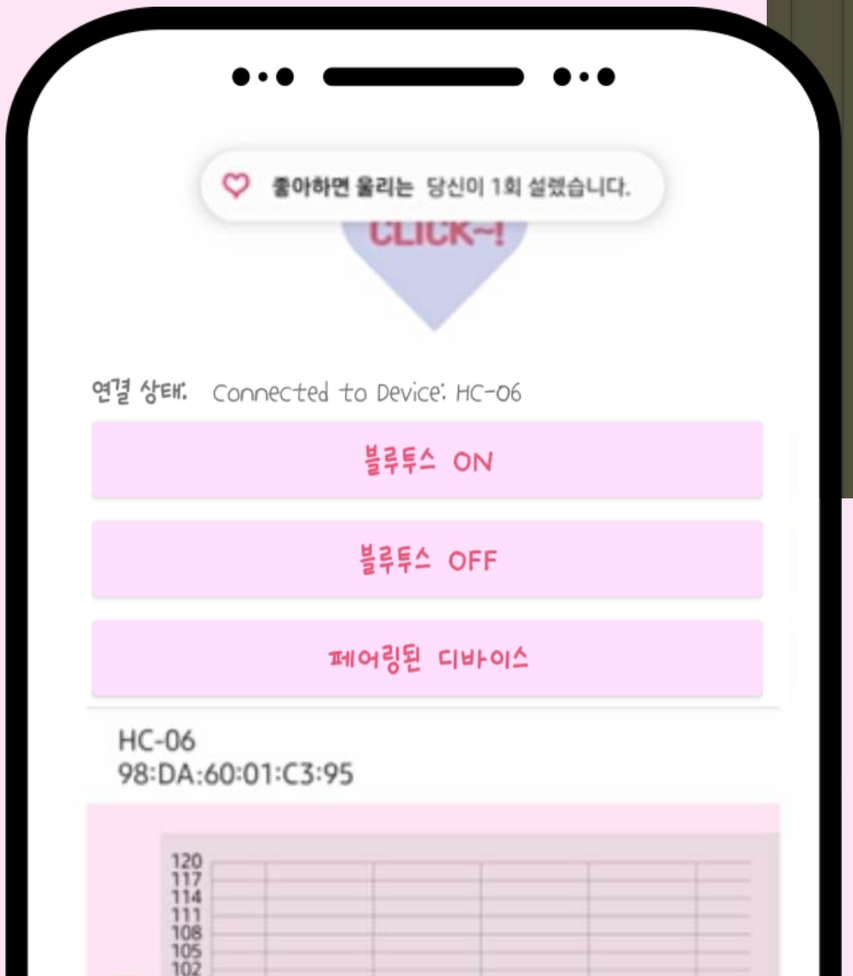
그래프 출력

```
private void addEntry() {
    LineData data = chart.getData();
    if (data != null) {
        ILineDataSet set = data.getDataSetByIndex(0);

        if (set == null) {
            set = createSet();
            data.addDataSet(set);
        }
        if(ECG.isEmpty()){
            data.addEntry(new Entry(set.getEntryCount(),  y: 60),  dataSetIndex: 0);
        }
        else {
            data.addEntry(new Entry(set.getEntryCount(), ECG.poll()),  dataSetIndex:
            //ECG.poll() queue 첫번째 값을 반환

        }
        data.notifyDataChanged();

        chart.notifyDataSetChanged();
        chart.setVisibleXRangeMaximum(10);
        chart.moveViewToX(data.getEntryCount());
    }
```

푸시 알림

notification 기능 활용

심박수가 100이상 넘어가면 푸쉬 알림

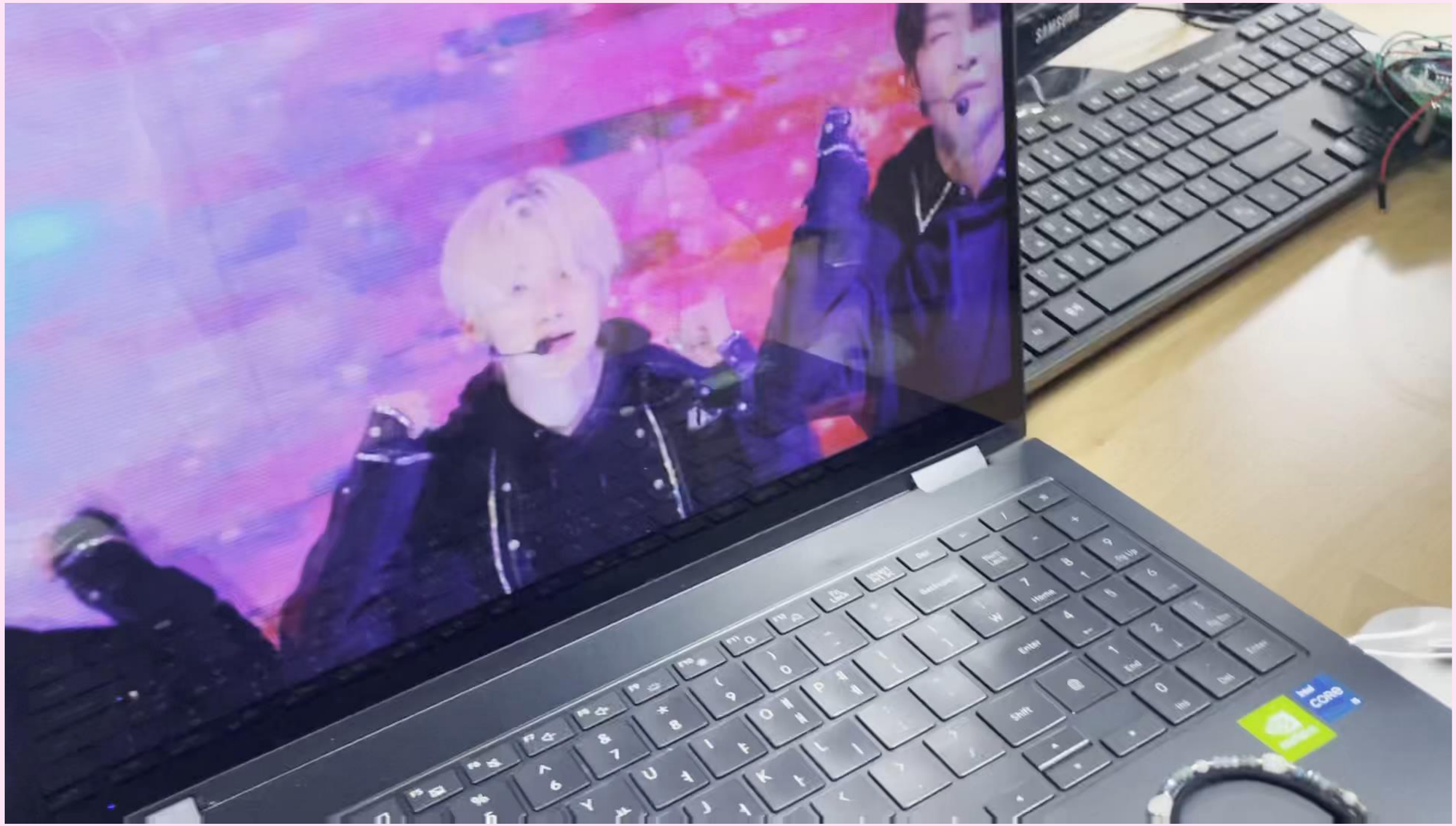



```
mHandler = new Handler() { //handler:서로 다른 thread간의 통신을 위한 장치
    public void handleMessage(Message msg) {
        if (msg.what == MESSAGE_READ) { //MESSAGE_READ = 2 => to identify message update
            int retValue =0;
            retValue = ((byte[]) msg.obj)[0]&(0xFF);
            if(retValue>=100) {
                count++;
                if(count/20>0){
                    if(count%20==0){
                        createNotificationChannel(DEFAULT, channelName: "default channel", NotificationManager.IMPORTANCE_HIGH);
                        createNotification(DEFAULT, id: 1, title: "좋아하면 올리는",("당신이 "+count/20+"회 설했습니다."),intent_pop);
                    }
                }
            }
            ECG.add(retValue);

        }
```

시연 영상

bpm< 100

bpm>=100


SAMSUNG
intel
CORE

보완할 점/한계

1. 사람마다 평균 심박수가 다름 → 개인별 평균 심박수를 먼저 구한 후, 맞춤형 푸시알림

2. 심박수 그래프의 피크점을 찾아 피크일때만 알람을 울리게 했으면 좋았을 듯함