



# 좋아하면 물리는 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



## 동기 & 어플 소개

- 개발 동기 및 아이디어
- 주요 기능 소개



## 역할 분담

- 아두이노 & 하드웨어
- 백엔드 & 데이터베이스
- 프론트엔드 & UI



## UI 소개

- UI



## 흐름 및 이론

- UI
- 세부 기능의 코드 소개



## 세부 기능 코드

- 아두이노
- 로그인 & 회원가입
- 블루투스
- ECG 그래프
- 푸시 알림

# 개발 동기

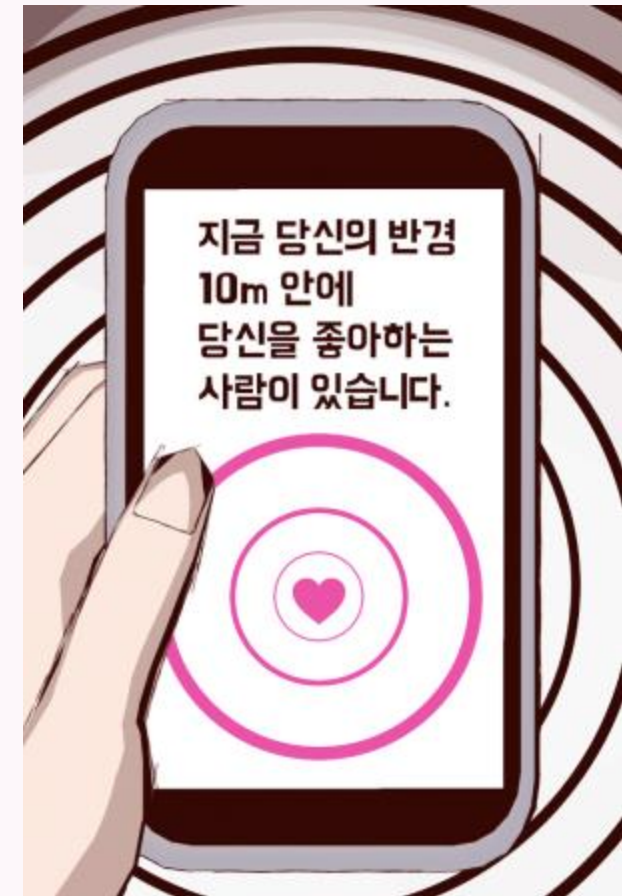
저희는 수업시간에 배운 ECG와 심장박동을 의학적인 부분이 아닌, 대학생들이 쉽게 흥미를 가질 수 있을만한 주제와 연관지어 어플을 개발하고 싶다는 생각이 들었고, 심박수를 이성과 만나는 때에 활용하는 상황을 떠올리게 되었습니다.

# 어플 주요 기능

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

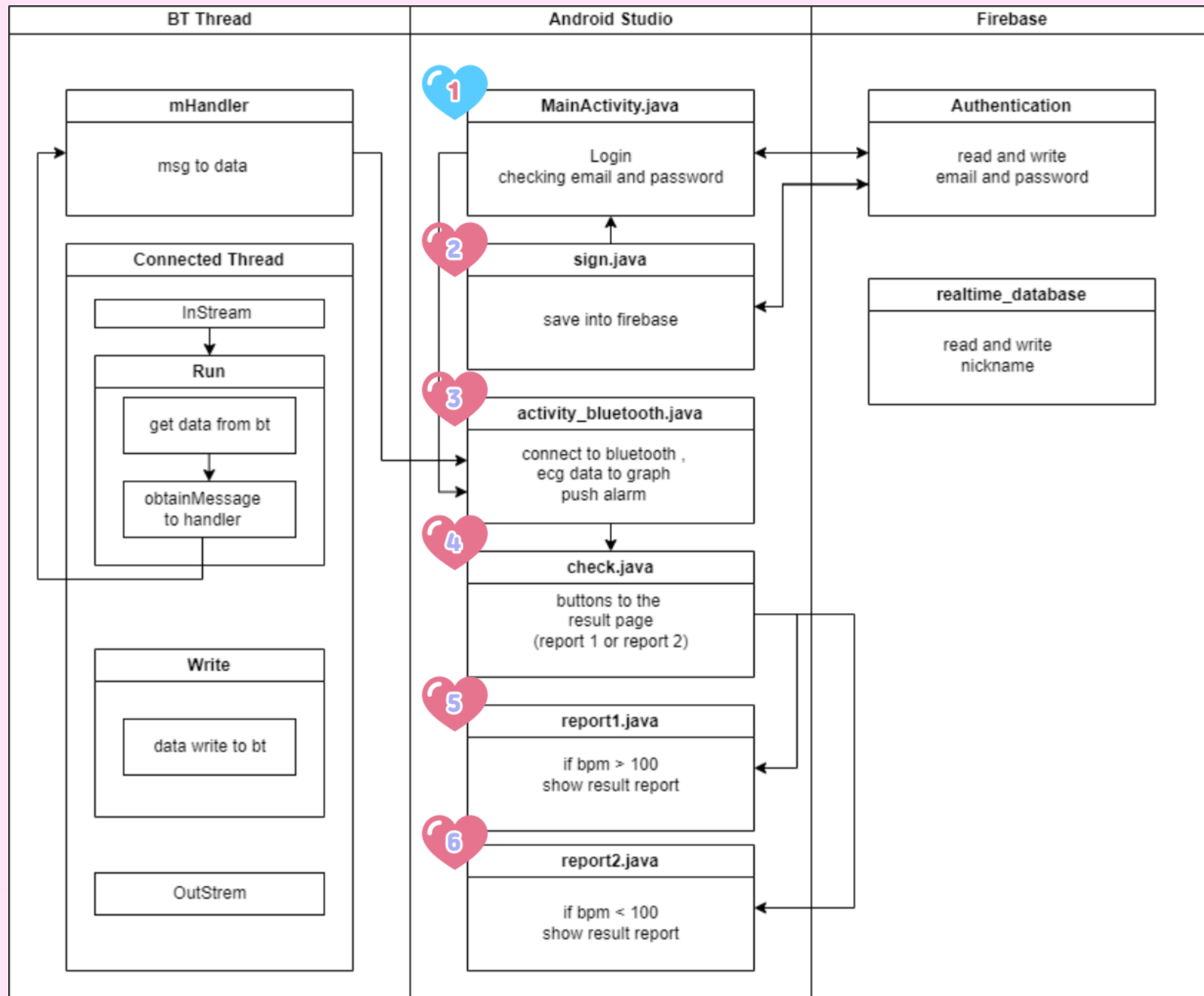
\* 의료용 목적이 아니기 때문에 진지하게 받아들이지 말 것

# 웹툰 '좋아하면 울리는'



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

# UI 및 흐름 소개





# UI #1 #2

... — ...

♡ 좋아하면 울리는 ♡

♡

♡

로그인

비밀번호

로그인  
회원가입

♡ 좋아하면 울리는 ♡

회원가입 하기

아이디 E-mail

비밀번호 password

닉네임 닉네임(영어)

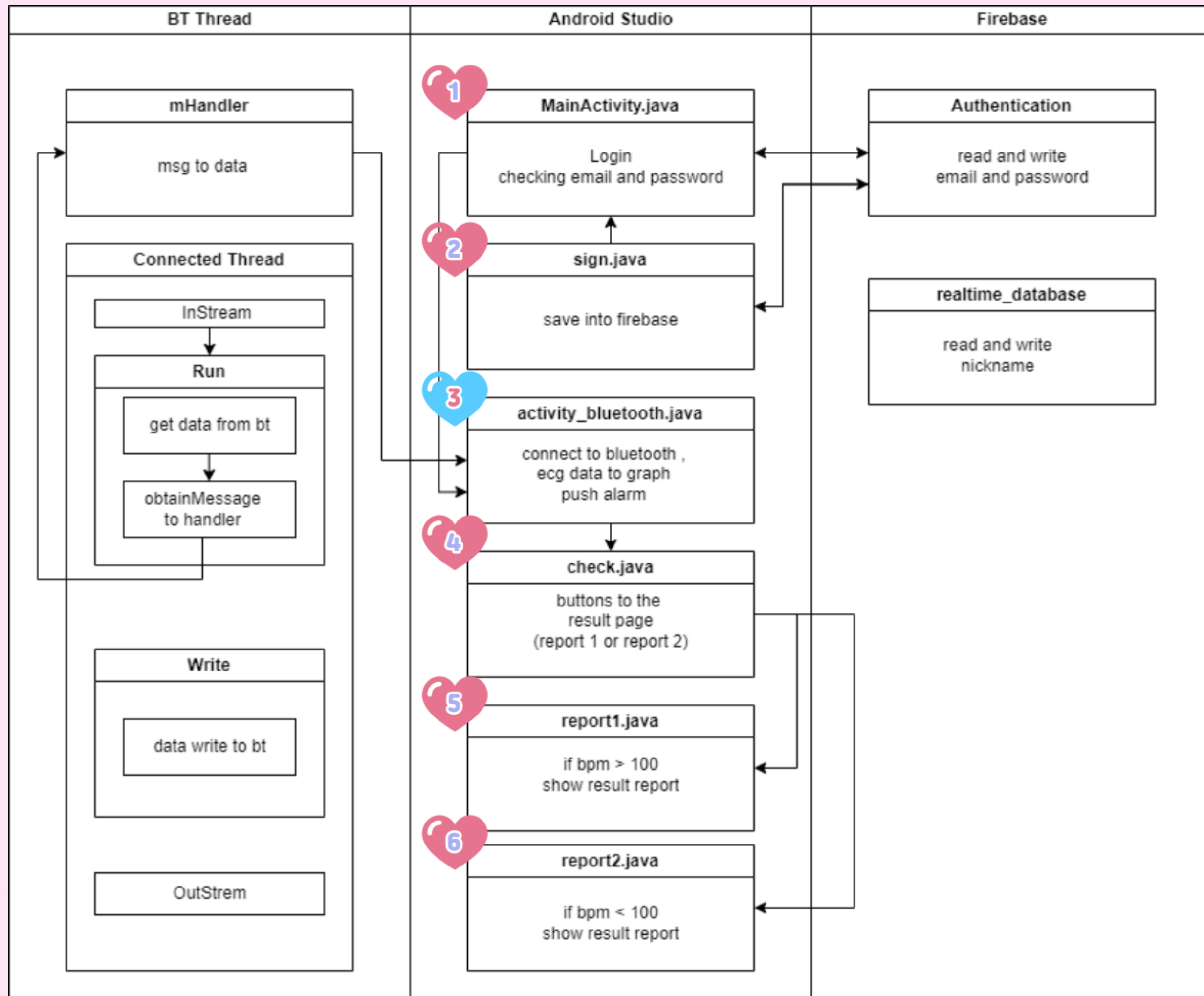
가입

기존 회원은  
아이디와 비밀번호를  
입력하여 로그인

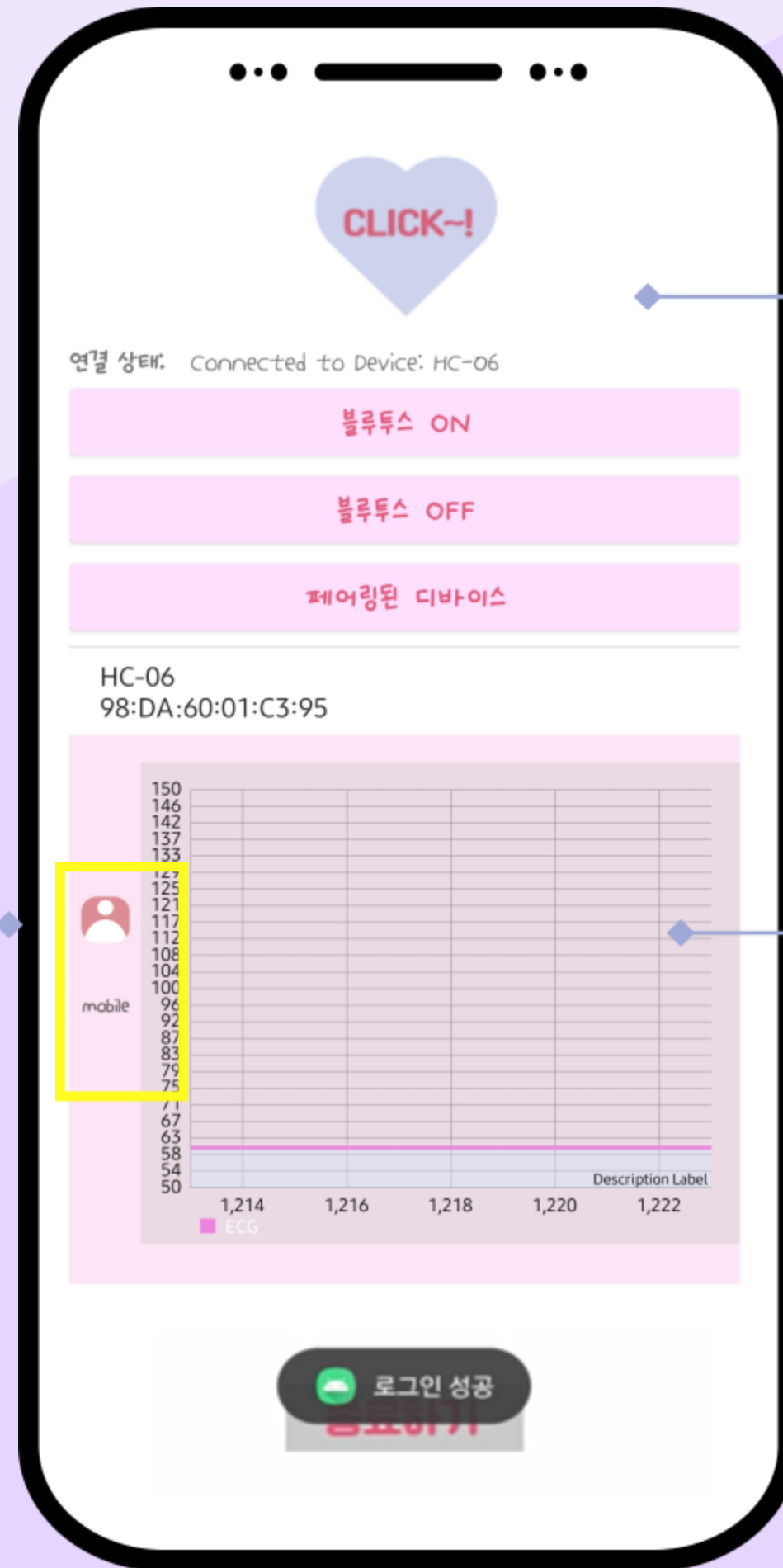
회원이 아닐 경우  
회원가입

아이디와 이메일, 닉네임을  
입력받아  
회원가입 진행

# UI 및 흐름 소개



# UI #3



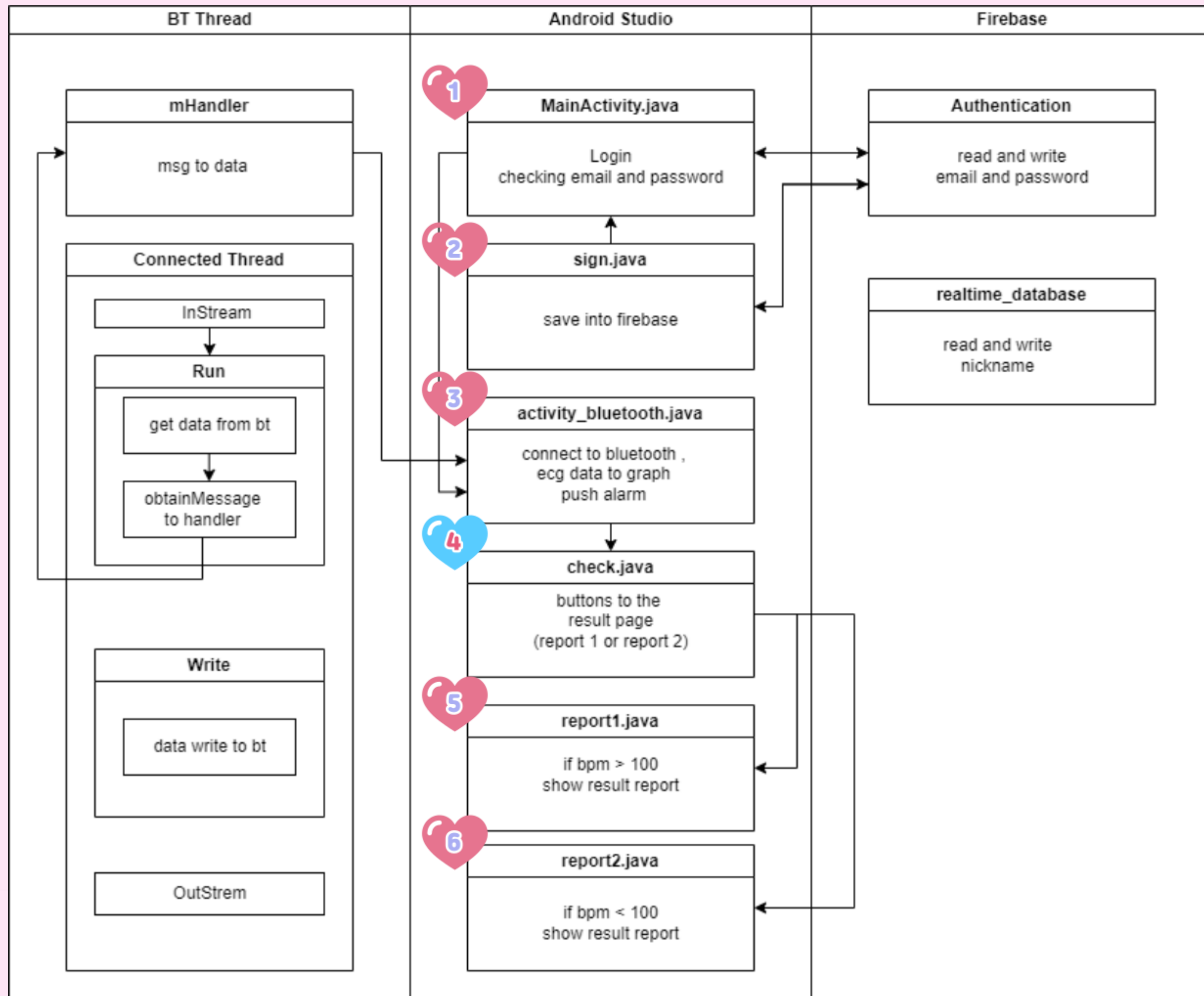
블루투스 연결  
+  
심전도 그래프 출력

회원가입시 입력받은  
닉네임 정보 불러옴

BPM  $\geq 100$   
count 값 푸시 알림



# UI 및 흐름 소개



# UI #4 #5 #6

[report 1]  
BPM > =100

[report 2]  
BPM < 100



# 구현 세부과정 설명



파이어베이스



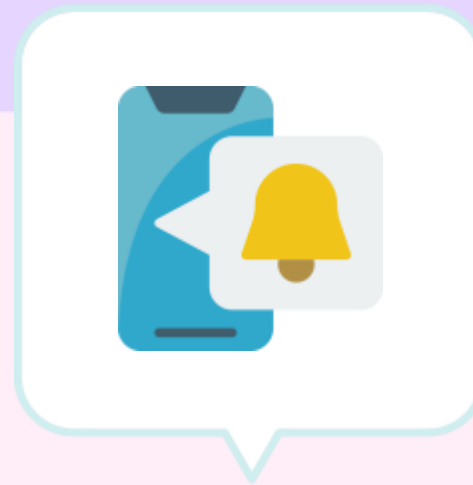
핸들러&스레드



블루투스



그래프



푸시 알림

# 파이어베이스

## Authentication

Users Sign-in method Templates Usage Settings

🔍 이메일 주소, 전화번호 또는 사용자 UID로 검색 사용자 추가 🔄 ⋮

| 식별자             | 제공업체 | 생성한 날짜 | 로그인한 날짜 | 사용자 UID            |
|-----------------|------|--------|---------|--------------------|
| mobile1234@...  | 📧    | 202... | 202...  | pC32u3o1AKZfG9i... |
| mobile@gmail... | 📧    | 202... | 202...  | b64ApRkl5rhq44b... |
| 1234@naver.c... | 📧    | 202... | 202...  | HwuL5VHtCrCXXs...  |

## 실시간 데이터베이스

데이터 규칙 백업 사용량

🛡️ 결제 사기나 피싱과 같은 악용으로부터 Realtime Database 리소스를 보호하세요. [앱 체크 구성](#)

🔗 <https://lovingu-18c4c-default-rtdb.firebaseio.com> > zoo

```
zoo
├── 2PQ6UjQTfAWsEQuzs0PKTtK9xE02
│   └── name: "minju123"
└── AHrvUurzjwXinYKhhiidSWtRfo1
    └── name: "yuba"
```

```
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; }
}
```

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

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



# 핸들러&스레드



문제 해결

메인 스레드

handleMessage

핸들러

메세지 큐

obtainMessage

sendMessage  
(= sendToTarget)

스레드

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

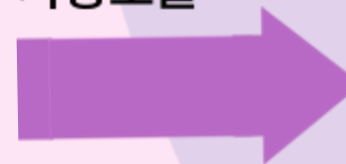
```
new Thread() { //블루투스 스레드
{
    @SuppressWarnings("MissingPermission")
    public void run() {
        boolean fail = false;

        BluetoothDevice device = mBTAdapter.getRemoteDevice(address);

        try {
            mBTSocket = createBluetoothSocket(device);
        } catch (IOException e) {
            fail = true;
            Toast.makeText(getBaseContext(), "소켓 생성 실패", Toast.LENGTH_SHORT).show();
        }

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

handleMessage  
자동호출



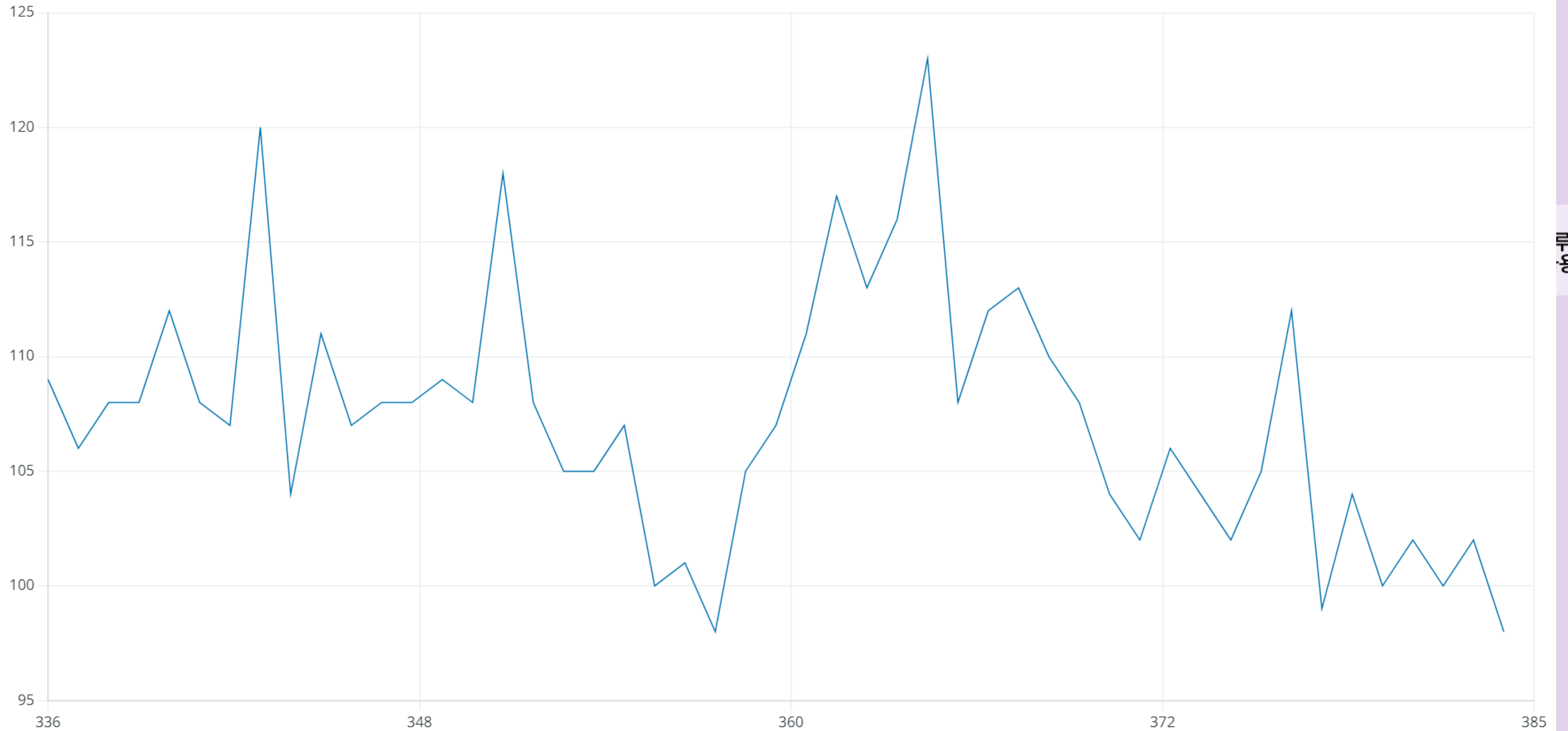
```
// 받은 값 String 형태로 보여주기, status에 연결된 장치 보여주기
mHandler = new Handler() { //handler:서로 다른 thread간의 통신을 위한 장치
    public void handleMessage(Message msg) {
        if (msg.what == MESSAGE_READ) { //MESSAGE_READ = 2 => to identify message
            int retValue = 0;
            retValue = ((byte[]) 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 올리는", ("당신
                    }
                }
            }
            EC6.add(retValue);
        }
    }
}
```



☒ value 1

Interpolate ☐

STOP



루투스  
용



```
if(start == false){  
  empty = 500-count;  
}  
  
delay(100); //0.1초 간격으로 측정
```

데이터 전송

ECG 큐에  
데이터 저장

```
private Queue<Integer> ECG = new LinkedList<>();  
private int count = 0;
```



큐에서 데이터를 하나씩 꺼내서 그래프를 출력!!

# ♥ 그래프 출력

MPAndroidChart

created by Philipp Jahoda

핸들러&스레드

PhilJay / MPAndroidChart Public

Sponsor Watch 1.1k

Code Issues 2k Pull requests 126 Actions Projects 2 Wik

master 6 branches 44 tags Go to file Add file Code

PhilJay Update README.md 0550d3f on Jun 21, 2021 2,070 commits

|                          |  |             |
|--------------------------|--|-------------|
| .github                  | Update FUNDING.yml                               | 3 years ago |
| .idea/runConfiguratio... | Remove unnecessary API checks                    | 4 years ago |
| MPChartExample           | Revert: e5b6619 - bring back polymorphism to ... | 2 years ago |
| MPChartLib               | Revert: e5b6619 - bring back polymorphism to ... | 2 years ago |
| design                   | docs(README): Update & simplify README           | 5 years ago |
| gradle/wrapper           | Update gradle                                    | 4 years ago |
| screenshots              | Update image                                     | 6 years ago |

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

```
// 받은 값 string 형태로 보여주기, status에 연결된 장치 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: "좋아하
                    }
                }
            }
            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.moveToX(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: 0);
            //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 푸시 알림

♡ 좋아하면 올리는 당신이 1회 섰했습니다.

CLICK~!

연결 상태: Connected to Device: HC-06

블루투스 ON

블루투스 OFF

페어링된 디바이스

HC-06

98:DA:60:01:C3:95

120  
117  
114  
111  
108  
105  
102

```
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 섰했습니다."), intent_pop);
                    }
                }
            }
            ECG.add(retValue);
        }
    }
}
```



# 시연 영상





# bpm < 100

♡

좋아하면 물리는

♡

로그인

비밀번호

로그인

회원가입



# bpm ≥ 100

CLICK~!

연결 상태: 블루투스 연결상태

블루투스 ON

블루투스 OFF

페어링된 디바이스

mobile

120

117

114

111

108

105

102

100

97

94

91

88

85

82

79

76

73

70

68

65

62

59

56

53

50

240

242

244

246

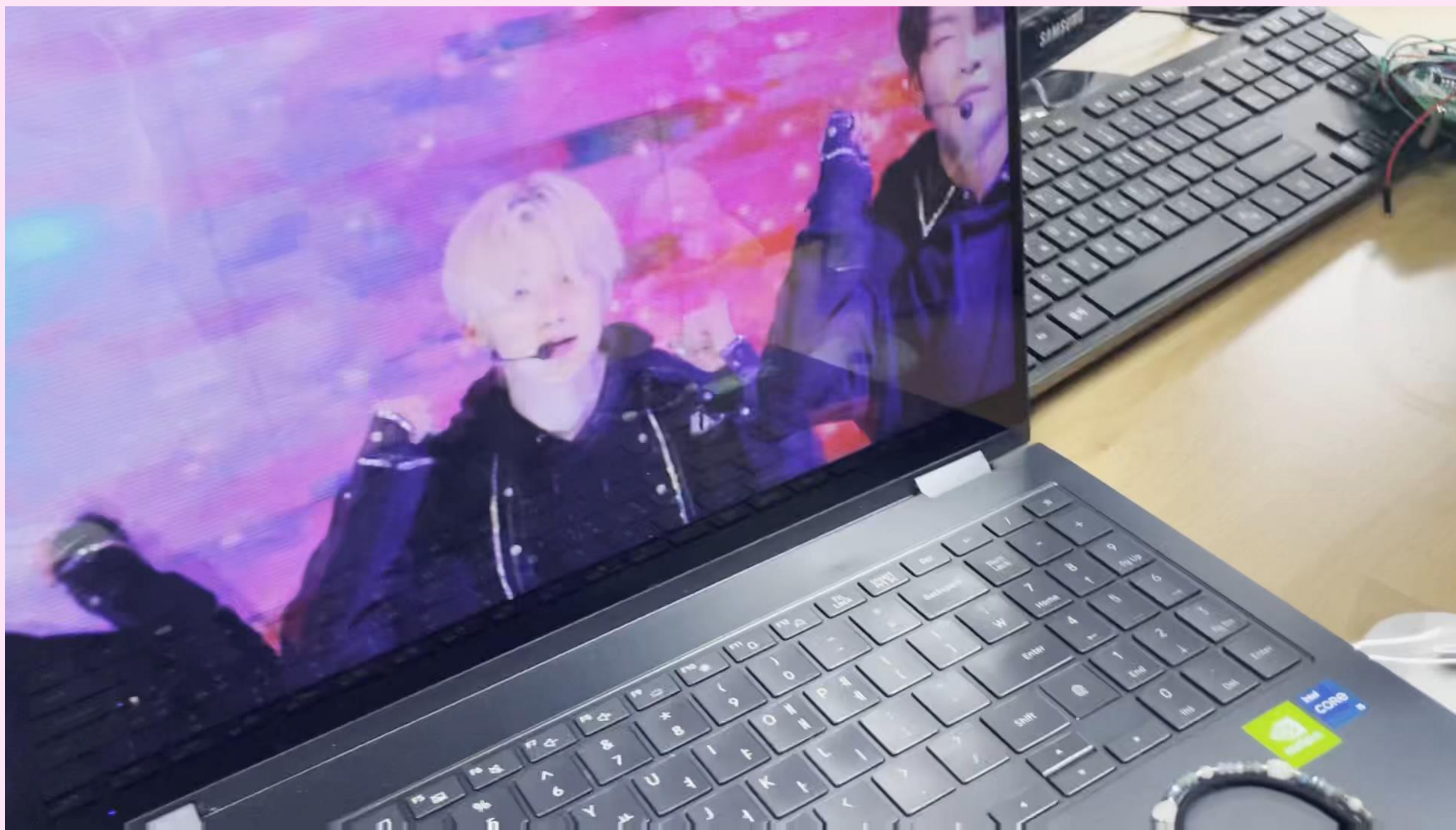
248

ECG

Description Label

종료하기







# 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

