

PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화 및
실증서비스 개발연계지원

Adnroid 앱 개발 가이드

(주) 디케이아이테크놀로지

문서번호: PHRP_KEIT-DE-01

Ver : 1.4

개정 이력

개정일자	문서번호	개정 내역	작성자	확인자
2019-07-08	1.0	최초 작성	김창환	
2019-08-20	1.1	데모 앱 업데이트 사항 반영	김창환	
2020-01-30	1.2	데모 앱 업데이트 사항 반영, build gradle 내용 추가, Json view 설명 추가	김창환	
2020-03-12	1.3	build gradle 내용추가, Omron 혈압 측정 설명 추가	김재중	
2022.03.23	1.4	build gradle 내용추가, MiBand 데이터 연동 설명 추가	김재중	

문서 목차

1. 앱 개발 환경 구성.....	1
1.1. 개발 환경 요구 사항.....	1
1.2. ANDROID STUDIO 개발 환경 설정	1
1.2.1. <i>Project Structure</i>	1
1.2.2. <i>New Module</i>	2
1.2.3. <i>Module dependency</i>	3
1.2.4. <i>build.gradle</i>	3
2. OAUTH 2.0 샘플소스.....	8
2.1. 서비스 ID 설정.....	8
2.1.1. <i>MainActivity.java</i>	9
2.2. 인증요청 소스코드	9
2.2.1. <i>OAuthAsyncTask.java</i>	9
2.2.2. <i>인증서버가 제공하는 로그인 페이지</i>	10
2.3. 실행결과.....	10
3. REDWOOKHEALTH SDK 샘플소스.....	11

3.1. PATIENT LIST	11
3.1.1. <i>PatientListActivity.java</i>	11
3.1.2. 실행결과	13
3.2. PATIENT CREATE / EDIT.....	13
3.2.1. <i>PatientAddActivity.java</i>	13
3.2.2. 실행결과	16
3.3. PATIENT DELETE	17
3.3.1. <i>PatientListAdapter.java</i>	17
3.3.2. 실행결과	18
3.4. JSON VIEW	19
3.5. BLUETOOTH PARING	20
3.5.1. <i>PrefsFragmentSettings.java</i>	20
3.5.2. 실행결과	21
3.6. ADD BLOOD PRESSURE DEVICE	22
3.6.1. <i>PrefsFragmentSettingsPlatform.java</i>	22
3.6.2. 실행결과	23
3.7. MEASURE BLOOD PRESSURE	24
3.7.1. <i>Fragment_2_Connect_BP.java</i>	24

3.7.2. 실행결과	26
3.8. 혈압 측정결과	27
3.8.1. <i>Fragment_3_Read_BP.java</i>	27
3.8.2. 실행결과	32
3.9. 기기 찾기.....	33
3.9.1. <i>DiscoveryActivity.java</i>	33
3.9.2. 실행결과	34
3.10. MiBAND 블루투스 페어링	35
3.10.1. <i>MiBandParingActivity.java</i>	35
3.10.2. 실행결과.....	37
3.11. 심박수 측정결과.....	38
3.11.1. <i>HeartRateDialog.java</i>	38
3.11.2. 실행결과.....	41
3.12. 활동데이터 수집.....	42
3.12.1. <i>GBDeviceAdapterv2.java</i>	42
3.12.2. 실행결과.....	44

1. 앱 개발 환경 구성

모바일용 SDK를 사용하여 앱을 개발하기 위한 개발 환경 요구 사항과 개발 환경 설정에 대해 설명합니다.

1.1. 개발 환경 요구 사항

모바일용 SDK를 사용하여 앱을 개발하기 위한 환경은 아래와 같습니다.

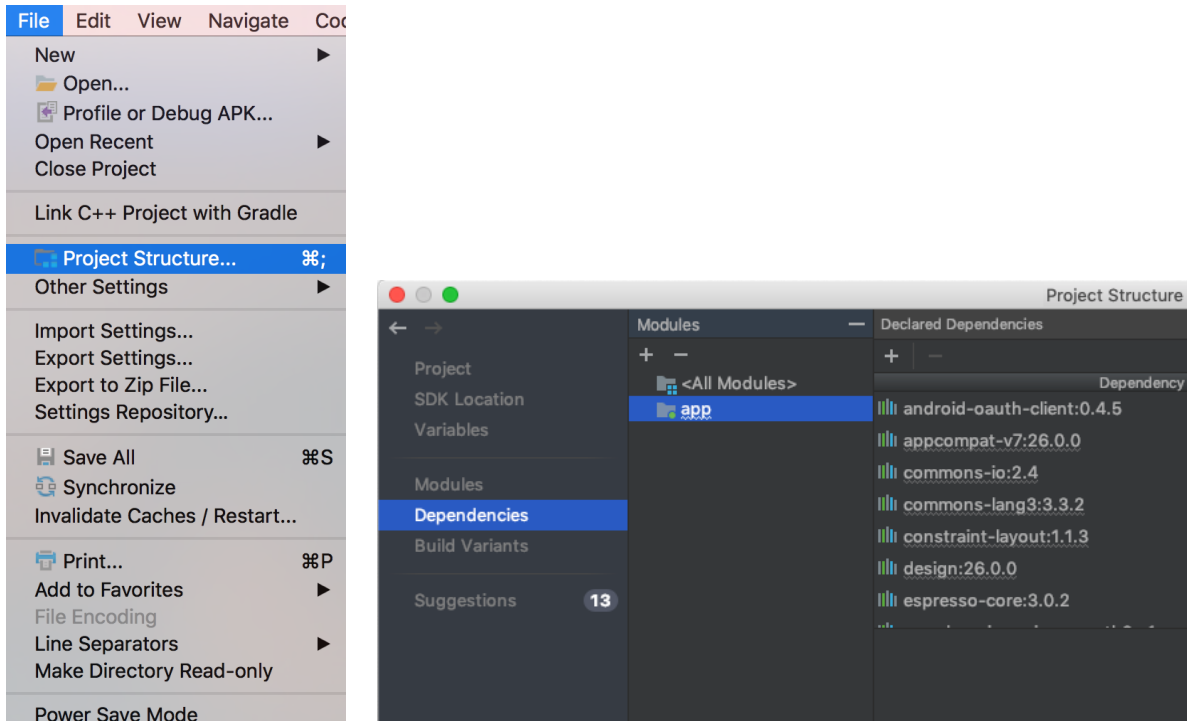
- Android Studio 개발 도구
- API Level 19 (안드로이드 4.4 킷캣) 이상 설치된 모바일
- RedwoodHealth-SDK.aar : 제공되는 aar 라이브러리

1.2. Android Studio 개발 환경 설정

제공되는 RedwoodHealth-SDK.aar 라이브러리를 import 하여 프로젝트 환경을 구성하는 방법입니다.

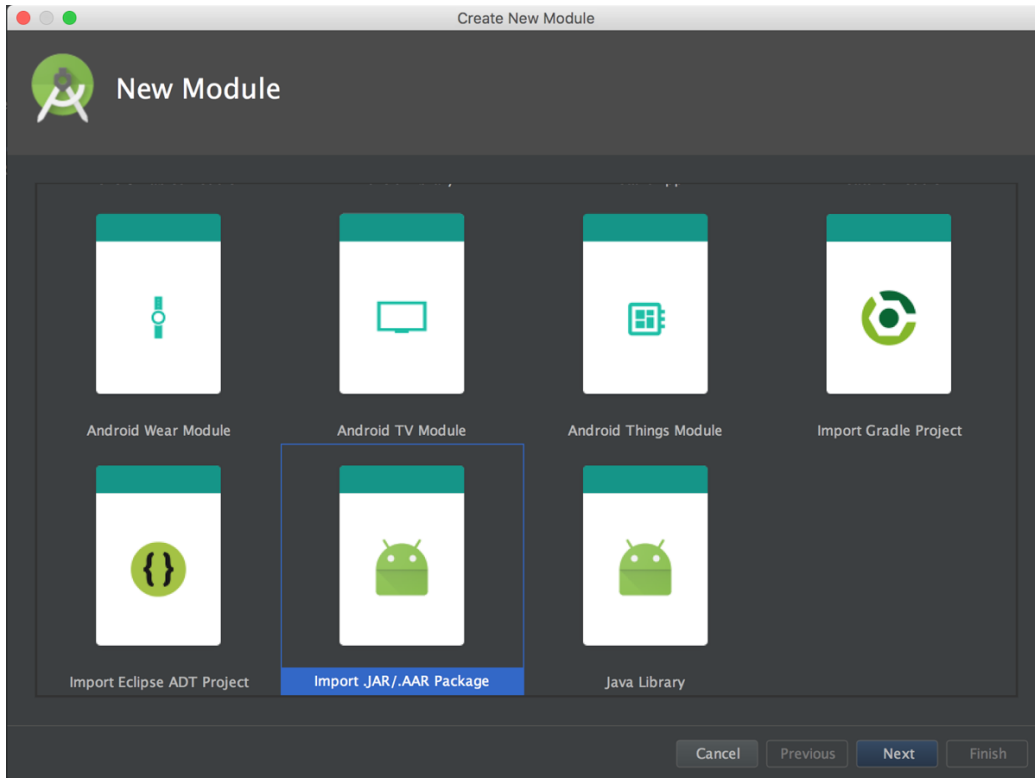
1.2.1. Project Structure

Project Structure에서 왼쪽 pane에서 + 버튼을 눌러 모듈을 추가합니다.

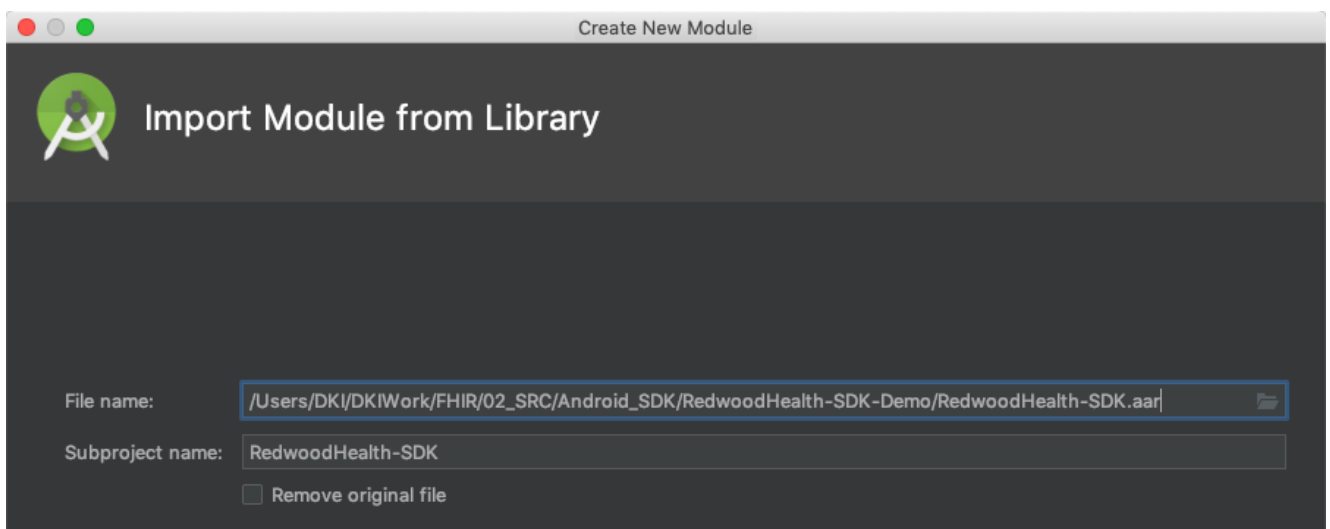


1.2.2. New Module

새 모듈의 형식을 선택하는 곳이 나오는데 import JAR/.AAR Package 를 선택합니다.



RedwoodHealth-SDK.aar 파일을 선택하고 Subproject name 은 원하는데로 수정가능.(모듈 이름)

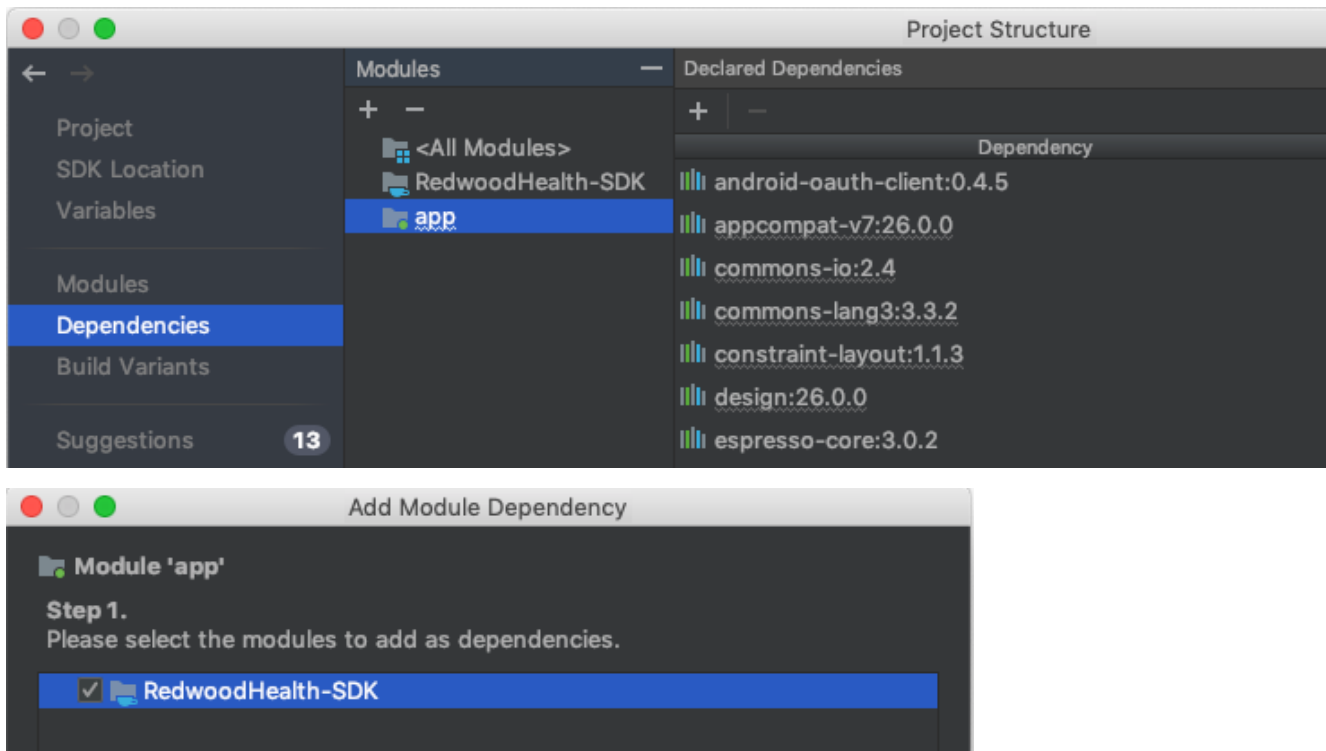


1.2.3. Module dependency

위 작업이 끝나면 Modules 에 app 말고 추가한 모듈이 보입니다.

Module 의 app 을 선택하고 Dependencies 탭에서 플러스 버튼을 눌러

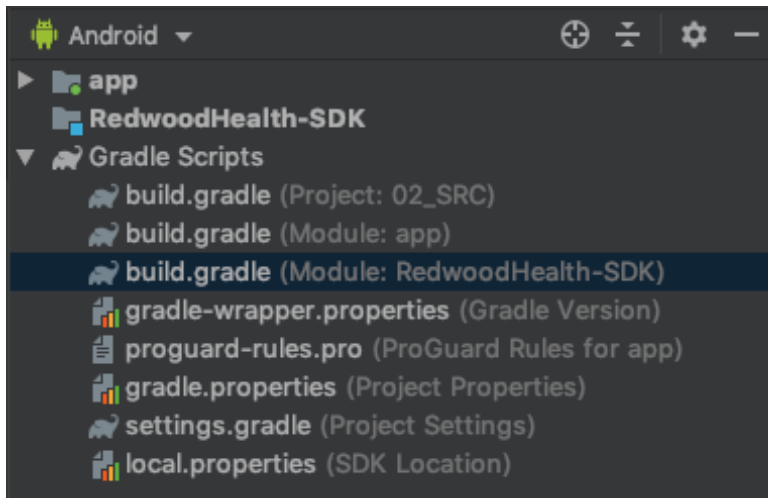
Module dependency 를 클릭하여 app 에 추가된 라이브러리 모듈을 연결합니다.



1.2.4. build.gradle

위의 과정으로 RedwoodHealth-SDK.aar 라이브러리를 import 하고 나면 프로젝트 구조와 build.gradle 구조가 자동으로 변경되어 SDK 라이브러리를 사용할 수 있는 프로젝트 구조가 됩니다.

프로젝트 구조



build.gradle (Module: PHRSDKLib)

```
1 configurations.maybeCreate("default")
2 artifacts.add "default", file('RedwoodHealth-SDK.aar')
```

settings.gradle (Project Settings)

```
1 include ':app', ':RedwoodHealth-SDK'
2
```

build.gradle (Module: app)

```
android {
    compileSdkVersion 28
    buildToolsVersion "27.0.0"
    defaultConfig { ... }
    buildTypes { ... }

    configurations { ... }

    dependencies {
        implementation fileTree(include: ['*.jar'], dir: 'libs')
        androidTestCompile('androidx.test.espresso:espresso-core:3.1.0', {
            exclude group: 'com.android.support', module: 'support-annotations'
        })
        implementation 'androidx.appcompat:appcompat:1.0.0'
        implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
        implementation 'androidx.recyclerview:recyclerview:1.0.0'

        //PHR 라이브러리 추가
        implementation project(path: ':RedwoodHealth-SDK-debug')

        //fhir 라이브러리
        implementation 'ca.uhn.hapi.fhir:hapi-fhir-android:2.5:dstu3@jar'

        //SDK 관련 라이브러리
        implementation 'org.slf4j:slf4j-api:1.7.12'
        implementation 'org.slf4j:slf4j-android:1.7.12'
        implementation 'commons-io:commons-io:2.4@jar'
        implementation 'org.apache.commons:commons-lang3:3.8'
        implementation 'com.squareup.okhttp3:okhttp:3.4.2'
        implementation 'com.google.code.gson:gson:2.7'
        implementation 'javax.servlet:servlet-api:2.5'

        //OAuth2 Lib
        implementation 'com.wu-man:android-oauth-client:0.4.5@aar'
        implementation 'com.google.apis:google-api-services-oauth2:v1-rev155-1.25.0'

        //Google Fit 설정
        implementation 'com.google.android.gms:play-services-fitness:18.0.0'
        implementation 'com.google.android.gms:play-services-auth:17.0.0'
    }
}
```

Build.gradle(app) 에 RedwoodHealth-SDK 모듈이 추가되어있습니다.

추가적으로 hapi 라이브러리, OAuth 라이브러리, Google Fit 라이브러리와 SDK 관련 라이브러리를 명시해 줍니다.

- HAPI 라이브러리

```
compile "ca.uhn.hapi.fhir:hapi-fhir-android:2.5:dstu3@jar"
```

- OAuth2 라이브러리

```
compile 'com.wu-man:android-oauth-client:0.4.5@aar'
```

```
compile 'com.google.apis:google-api-services-oauth2:v1-rev155-1.25.0'
```

- Google Fit 라이브러리

```
implementation 'com.google.android.gms:play-services-fitness:18.0.0'
```

```
implementation 'com.google.android.gms:play-services-auth:17.0.0'
```

- SDK 관련 라이브러리

```
implementation 'org.slf4j:slf4j-api:1.7.12'
```

```
implementation 'org.slf4j:slf4j-android:1.7.12'
```

```
implementation 'commons-io:commons-io:2.4@jar'
```

```
implementation 'org.apache.commons:commons-lang3:3.8'
```

```
implementation 'com.squareup.okhttp3:okhttp:3.4.2'
```

```
implementation 'com.google.code.gson:gson:2.7'
```

```
implementation 'javax.servlet:servlet-api:2.5'
```

- Omron 관련 라이브러리

```
- //Omron 설정
```

```
implementation 'com.bluelinelabs:logansquare:1.3.7'
```

```
annotationProcessor 'com.bluelinelabs:logansquare:1.3.7'
```

```
//plugin for icons
```

```
compile 'com.mikepenz:iconics-core:2.8.1@aar'
```

```
compile 'com.mikepenz:fontawesome-typeface:4.7.0.0@aar'
```

```
compile 'com.mikepenz:google-material-typeface:2.2.0.3.original@aar'
```

```
//plugin for dialogbox
```

```
compile 'com.afollestad.material-dialogs:core:0.9.2.1'
```

```
//plugin for leakcanary
```

```
debugCompile 'com.squareup.leakcanary:leakcanary-android:1.5'
```

```
releaseCompile 'com.squareup.leakcanary:leakcanary-android-no-op:1.5'
```

```
testCompile 'com.squareup.leakcanary:leakcanary-android-no-op:1.5'
```

```
//Material Drawer
compile('com.mikepenz:materialdrawer:5.8.1@aar') {
    transitive = true
}

// add rxAndroid
compile 'io.reactivex:rxandroid:1.2.1'
compile 'io.reactivex:rxjava:1.1.6'
```

- dstu3 에서 r4로 마이그레이션

```
- //fhir r4
implementation "ca.uhn.hapi.fhir:hapi-fhir-android:5.5.1"
implementation "ca.uhn.hapi.fhir:hapi-fhir-structures-r4:5.5.1"
implementation "ca.uhn.hapi.fhir:org.hl7.fhir.r4:5.5.3"
compile("commons-validator:commons-validator:1.5.1") {
    exclude group: 'commons-logging'
    exclude group: 'commons-collections'
    exclude group: 'commons-digester'
}
compile 'org.slf4j:slf4j-api:1.7.12'
compile 'org.slf4j:slf4j-android:1.7.12'
compile 'commons-io:commons-io:2.4@jar'
compile 'javax.servlet:servlet-api:2.5'
compile 'org.apache.commons:commons-lang3:3.8'
compile 'com.google.code.gson:gson:2.7'
// okhttp3 버전을 4.0 이상으로 올릴 경우 "okhttp3.internal.Version" class 가 미존재
// "ca.uhn.fhir.okhttp.client.OkHttpRestfulClient" 에서 Version class를 참조하지 못해 예
리 발생함
implementation 'com.squareup.okhttp3:okhttp:3.14.9'
```

```
// logging-interceptor는 반환된 모든 응답에 대해 로그 문자열을 생성합니다.
implementation 'com.squareup.okhttp3:logging-interceptor:3.14.9'
```

- MiBand 라이브러리

```
- implementation "com.github.PhilJay:MPAndroidChart:v3.1.0"
implementation 'com.jaredrummler:colorpicker:1.0.2'
implementation "androidx.preference:preference:1.1.1"
implementation "de.cketti.library.changelog:ckchangelog:1.2.2"
implementation "org.greenrobot:greendao:2.2.1"
```

2. OAuth 2.0 샘플소스

소스코드는 RedwoodHealth-SDK.aar 라이브러리와 함께 제공되는 테스트앱에 구현되어 있습니다.

RedwoodHealth-SDK.aar 라이브러리를 import 한 프로젝트에서 OAuth 2.0 API 을 사용하여 PHR 플랫폼의 인증서버로부터 access token 을 발급받는 샘플소스에 대한 설명입니다.

2.1. 서비스 ID 설정

OAuth 2.0 API 를 통해서 access token 을 발급받으려면 PHR 플랫폼에 가입한 서비스정보를 설정해야 합니다.

- user ID : 사용자 ID.
- login server url : 사용자 인증을 위한 url
- token request url : 토큰 요청을 위한 url
- callback uri : access token 처리 위한 callback url.
- client ID : 서비스 공급자에게 부여 받은 client ID
- client secret : 서비스 승인후 발급받은 client secret

2.1.1. MainActivity.java

```
//OAuth2 인증을 위한 설정 값
String userID = "test001";
String loginServerUrl = "http://www.test.com/auth";
String tokenRequestUrl = "http://www.test.com/auth/token";
String callbackUrl = "http://127.0.0.1:7000/phrtest/receiveCode.html";
String clientID = "testClient001";
String clientSecret = "781aff68-9296-403e-9884-949350ba9s7v";

PHROAuth2.configOAuth2Client(userID,loginServerUrl,tokenRequestUrl,callbackUrl,clientID,clientSecret);
```

2.2. 인증요청 소스코드

RedwoodHealth-SDK 의 PHROAuth2 에 PHR 플랫폼에 가입한 서비스정보를 설정한 후 인증요청하는 소스코드입니다.

인증요청을 하면 PHR 플랫폼의 인증서버에서 제공하는 로그인 페이지를 통해서 서비스에 가입한 사용자 정보를 입력하여 로그인하면 인증서버가 사용자를 확인하여 access token 을 발급하게됩니다.

2.2.1. OAuthAsyncTask.java

```
private Context context; //Application Context (Activity 에서 전달)
private FragmentManager fragmentManager; //OAuth UI Fragment (Activity 에서 전달)
PHROAuth2 phroAuth = new PHROAuth2(context,fragmentManager);
```

2.2.2. 인증서버가 제공하는 로그인 페이지

5:01 오후

REDWOOD HEALTH
DK1 / TECHNOLOGY

아이디

비밀번호

로그인

5:08 오후

테스트서비스001 서비스가 PHR 플랫폼 계정에 액세스 하려고 합니다.

· 개인 건강정보 조회 ☒ 승인 ☐ 거절

· 개인 건강정보 등록 ☒ 승인 ☐ 거절

테스트서비스001 서비스가 다음 작업을 할 수 있도록 허용하시겠습니까?

취소 허용

FragmentManager 에서 자동으로 Dialog 팝업을 띄워줍니다.
해당 페이지에 디자인은 html 파일을 수정하여 바꿀 수 있습니다.

2.3. 실행결과

로그인 페이지에서 입력한 사용자 정보가 정상적이면 PHR 플랫폼의 인증서버가 access token 를 보내옵니다.

SDK에서는 인증서버로 받은 access token 값을 내부 스토리지에 저장하여 access token 이 필요한 리소스 서버에 접근할 때 사용됩니다.

3. RedwookHealth SDK 샘플소스

소스코드는 RedwoodHealth-SDK.aar 라이브러리와 함께 제공되는 데모앱에 구현되어 있습니다. Patient 에 대한 조회, 신규 등록, 내용 수정, 삭제 등의 기능을 설명합니다.

3.1. Patient List

Patient List 는 환자의 목록을 조회하는 API 호출과 리스트 UI 에 표시하는 샘플코드입니다.

access token 이 필요한 API 입니다.

API 호출시 스토리지에 저장된 access token 이 없으면 SDK 내부에서 자동으로 access token 을 발급받고 자동으로 API 다시 호출합니다.

3.1.1. PatientListActivity.java

Patient 출력을 위한 리소스를 할당하고 AsyncTask 를 통해 Patient 값을 Intent 값으로 받아와서 출력합니다.

```
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_resource_list_v2);
    setTitle(MainActivity.MENU_LIST_00);

    try {

        // get the patients sent from the async task
        ArrayList<PHRPatient> patientList = (ArrayList<PHRPatient>)
            getIntent().getExtras().getSerializable("patient_list");

        // initialise listview and data array
        ListView patientListView = findViewById(R.id.resource_list);
        if(patientList != null && patientList.size() > 0) {
            mPatientId = patientList.get(0).getPatientId();
            PHRParser jsonParser = PHRManager.getJsonParser();
            jsonParser.setPrettyPrint(true);
```



```
        PHRPatientViewModel patientViewModel = new
        PHRPatientViewModel();

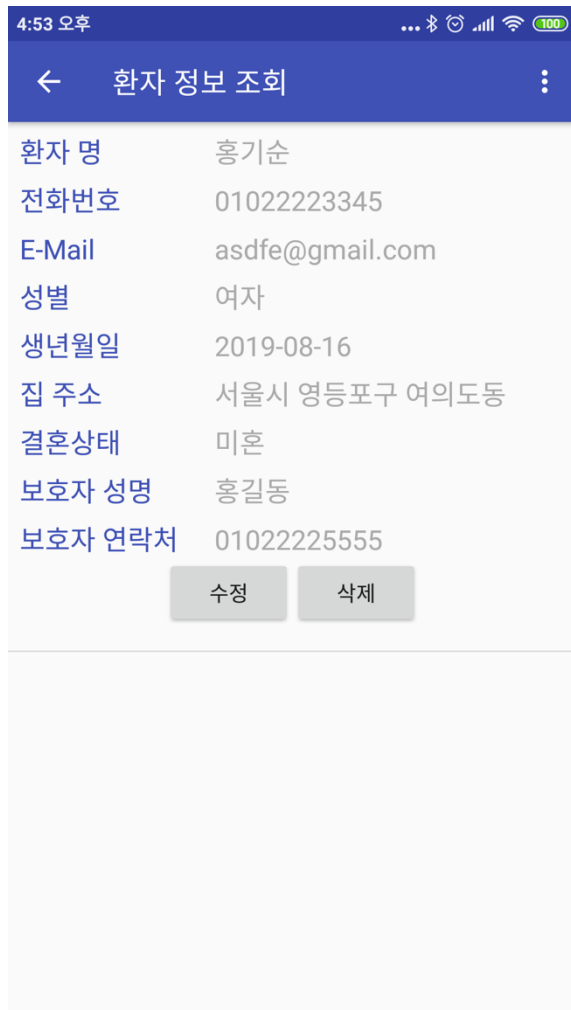
        mJsonString =
        jsonParser.encodeResourceToString(patientViewModel.createResource(patientList.get(0))
        );

        // attach data to a the new adapter and then attach adapter to the ListView
        PatientListAdapter adapter = new PatientListAdapter(PatientListActivity.this, patientList,
        patientListView);
        patientListView.setAdapter(adapter);
    }else{
        Log.d(this.getClass().getName(),"patient_list size error!");
    }

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setHomeAsUpIndicator(null);

} catch (Exception ex){
    ex.printStackTrace();
}
```

3.1.2. 실행결과



3.2. Patient Create / Edit

Patient Create/ Edit 는 환자에 정보를 생성/수정하는 API 호출과 UI 를 표시하는 샘플코드입니다.

3.2.1. PatientAddActivity.java

Patient 정보를 입력하기 위한 UI 를 만들고 AsyncTask 를 통해 Patient 정보를 등록합니다.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_patient_add_v2);
    setTitle("환자 정보 등록");
}
```

```
btnCreate = (Button) findViewById(R.id.btnPatientOk);
editTextFamilyName = (EditText) findViewById(R.id.etxPatientFamilyName);
editTextGivenName = (EditText) findViewById(R.id.etxPatientGivenName);
editTextPhoneNumber = (EditText) findViewById(R.id.etxPatientPhoneNumber);
editTextEmail = (EditText) findViewById(R.id.etxPatientEmail);
spinnerGender = (Spinner) findViewById(R.id.spnPatientGender);
editTextDateOfBirth = (EditText) findViewById(R.id.etxPatientDateOfBirth);
editTextAddress = (EditText) findViewById(R.id.etxPatientAdress);
spinnerMaritalStatus = (Spinner) findViewById(R.id.spnPatientMaritalStatus);
editTextGuardianFamilyName = (EditText) findViewById(R.id.etxPatientGuardianFamilyName);
editTextGuardianGivenName = (EditText) findViewById(R.id.etxPatientGuardianGivenName);
editTextGuardianPhoneNumber = (EditText)
findViewById(R.id.etxPatientGuardianPhoneNumber);
initData();

btnCreate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (btnCreate.getText().toString().equalsIgnoreCase("등록")) {
            createResouce();
        } else if (btnCreate.getText().toString().equalsIgnoreCase("수정")) {
            updateResource();
        }
    }
});

Intent intent = new Intent(this.getIntent());

String result_type = intent.getStringExtra("result_type");

if (result_type != null && result_type.equalsIgnoreCase("show_patient_list")) {
    showPatientList();
}

String created_patient_id = intent.getStringExtra("created_patient_id");
```

```
if (created_patient_id != null) {
    createPatientBlockChain(created_patient_id);
}

PHRPatient patient = (PHRPatient) getIntent().getExtras().getSerializable("patient");

if (patient != null) {
    setTitle("환자 정보 수정");
    btnCreate.setText("수정");

    showPatient(patient);
} else {
    btnCreate.setText("등록");
}

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setHomeAsUpIndicator(null);
```

```
protected void initData()
{
    String[] genderList = PHRGender.getGenderList();

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(
        this,
        R.layout.support_simple_spinner_dropdown_item,
        genderList
    );

    spinnerGender.setAdapter(adapter);
    spinnerGender.setSelection(0);

    String[] maritalStatusList = PHRPatient.getMaritalStatusList();
```

```

    ArrayAdapter<String> adapterMaritalStatus = new ArrayAdapter<String>(
        this,
        R.layout.support_simple_spinner_dropdown_item,
        maritalStatusList
    );

    spinnerMaritalStatus.setAdapter(adapterMaritalStatus);
    spinnerMaritalStatus.setSelection(0);    }

```

3.2.2. 실행결과

4:56 오후

← 환자 정보 수정

사용자명 홍기순

전화번호 01022223345

E-Mail asdfs@gmail.com

성별 여자 ▼

생년월일 2019-08-16

집 주소 서울시 영등포구 여의도동

결혼상태 미혼 ▼

보호자명 홍길동

보호자 연락처 01022225555

수정

3.3. Patient Delete

Patient Delete 는 환자에 정보를 삭제하는 API 호출과 UI 를 표시하는 샘플코드입니다.

3.3.1. PatientListAdapter.java

Patient 정보를 삭제하기 위한 UI 를 만들고 AsyncTask 를 통해 Patient 정보를 삭제합니다.

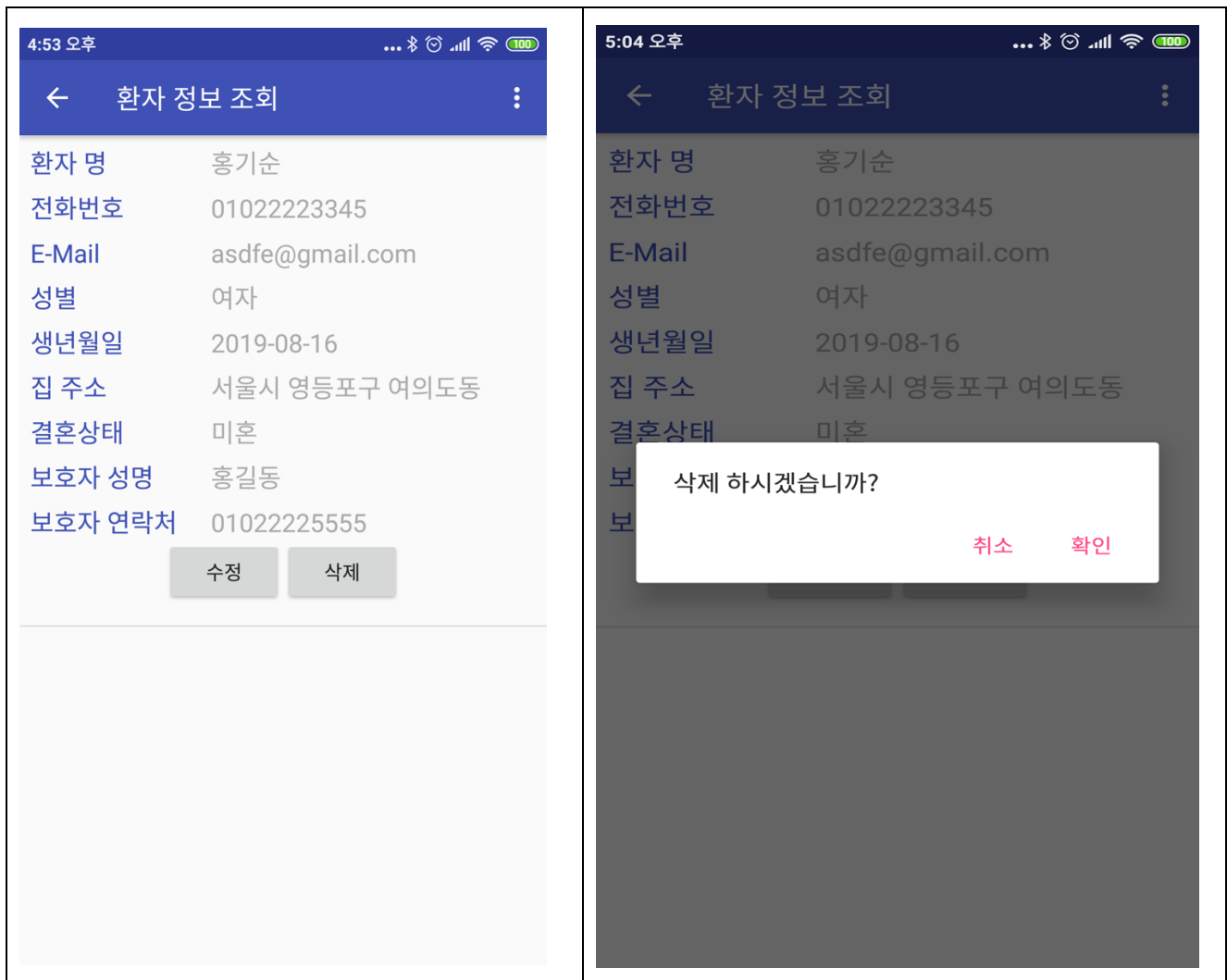
```
public boolean deleteResource(String patientId)
{
    try {

        PHRPatient obj = new PHRPatient();
        obj.setPatientId(patientId);

        PatientAddAsyncTask addAsyncTask = new PatientAddAsyncTask(inActivity,obj,
            PHRUtils.OperationType.DELETE);
        addAsyncTask.execute();
    }catch (Exception ex){
        ex.printStackTrace();
        return false;
    }

    return true;
}
```

3.3.2. 실행결과



3.4. JSON View

실행 결과에 대해 Json 형태로 결과를 확인 할 수 있도록 기능을 제공 합니다.
우측 상단 메뉴를 통해 접근 할 수 있습니다.



3.5. Bluetooth Paring

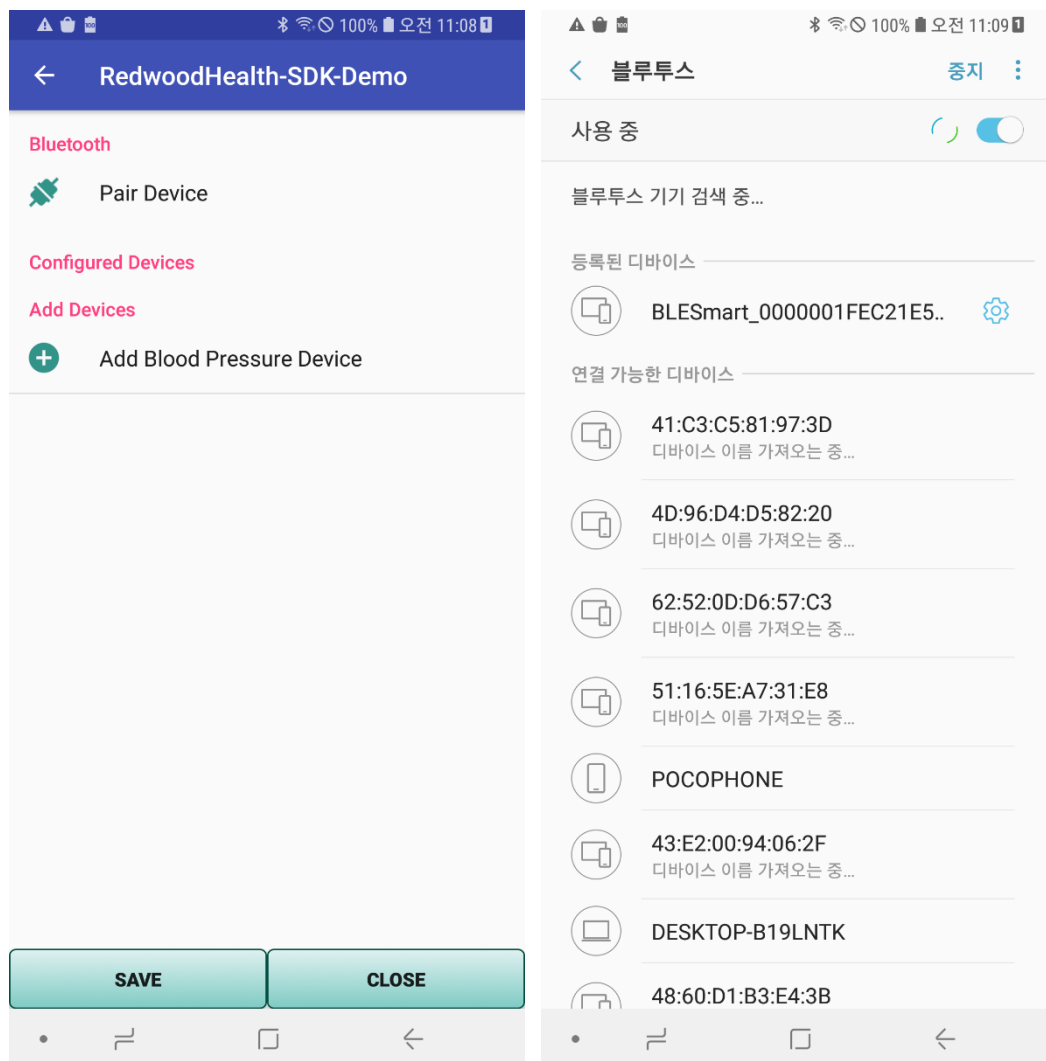
Omron 혈압측정 디바이스와 Bluetooth 페어링을 하는 샘플 코드입니다.

3.5.1. PrefsFragmentSettings.java

디바이스와 Bluetooth 페어링을 합니다.

```
private void setPreferenceBluetoothPair() {  
    Preference preference = findPreference("key_bluetooth_pair");  
    preference.setOnPreferenceClickListener(new Preference.OnPreferenceClickListener() {  
        @Override  
        public boolean onPreferenceClick(Preference preference) {  
            Intent intent = new Intent(Intent.ACTION_MAIN, null);  
            intent.setAction(android.provider.Settings.ACTION_BLUETOOTH_SETTINGS);  
            startActivity(intent);  
            return true;  
        }  
    });  
}
```

3.5.2. 실행결과



3.6. Add Blood Pressure Device

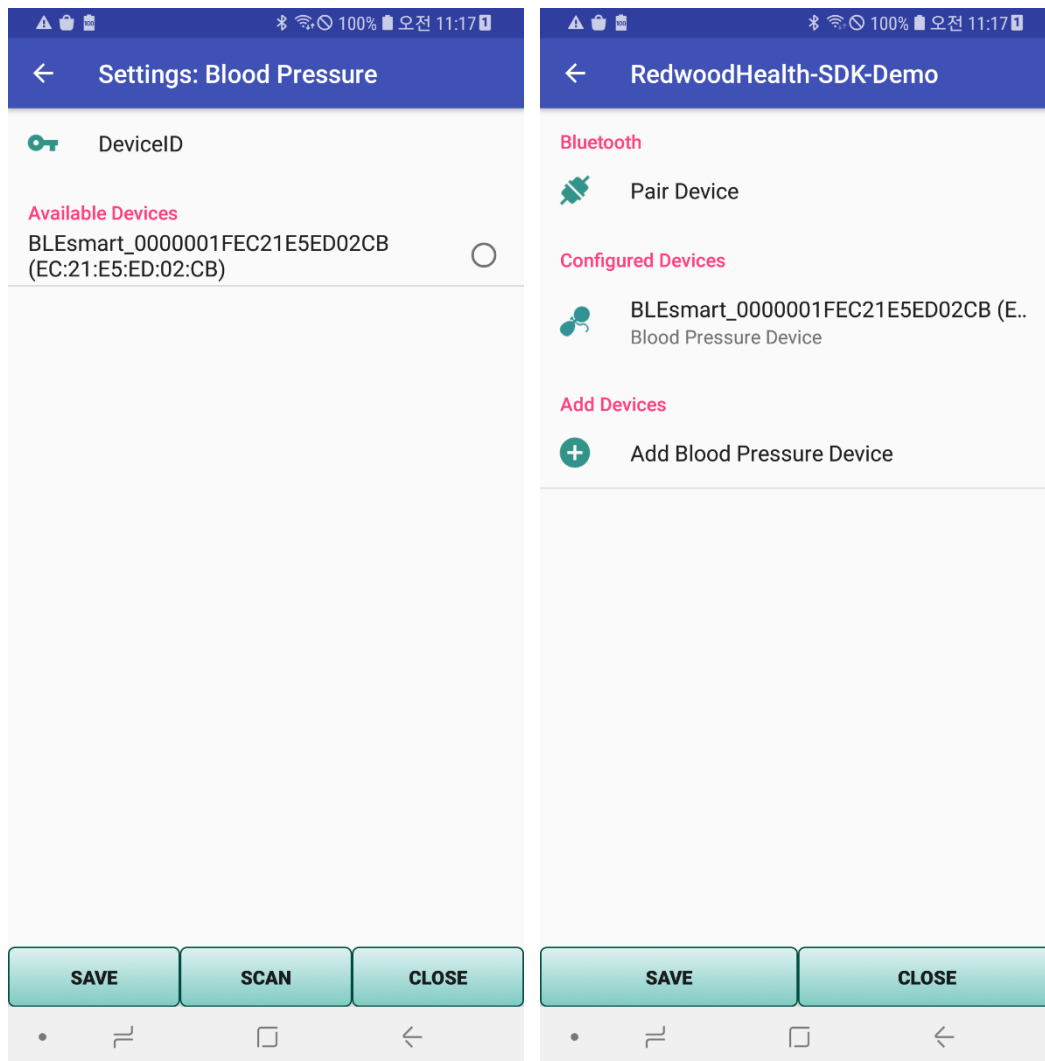
Omron 혈압측정 디바이스를 추가하기 위해 Scanning 하는 샘플 코드입니다.

3.6.1. PrefsFragmentSettingsPlatform.java

디바이스를 추가하기 위해 Scanning 을 합니다.

```
private void setScanButton() {
    try {
        final Button button = (Button) getActivity().findViewById(R.id.button_2);
        if (isScanning) {
            button.setText("Scanning...");
            button.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    isScanning = false;
                    myBluetooth.scanOff();
                }
            });
        } else {
            button.setText("Scan");
            button.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    scanLeDevice();
                }
            });
        }
    } catch (Exception ignored) {
    }
}
```

3.6.2. 실행결과



3.7. Measure Blood Pressure

디바이스에서 혈압측정을 한 후 측정 데이터를 받는 샘플 코드입니다.

3.7.1. Fragment_2_Connect_BP.java

혈압 측정을 한 후 데이터를 받습니다.

```
OnReceiveListener onReceiveListener = new OnReceiveListener() {
    @Override
    public void onReceived(Message msg) {
        byte[] data;
        byte[] buf = new byte[2];
        ByteBuffer byteBuffer;
        switch (msg.what) {
            case MyBlueTooth.MSG_BPM_DATA_RECV:
                int idx = 0;
                data = (byte[]) msg.obj;

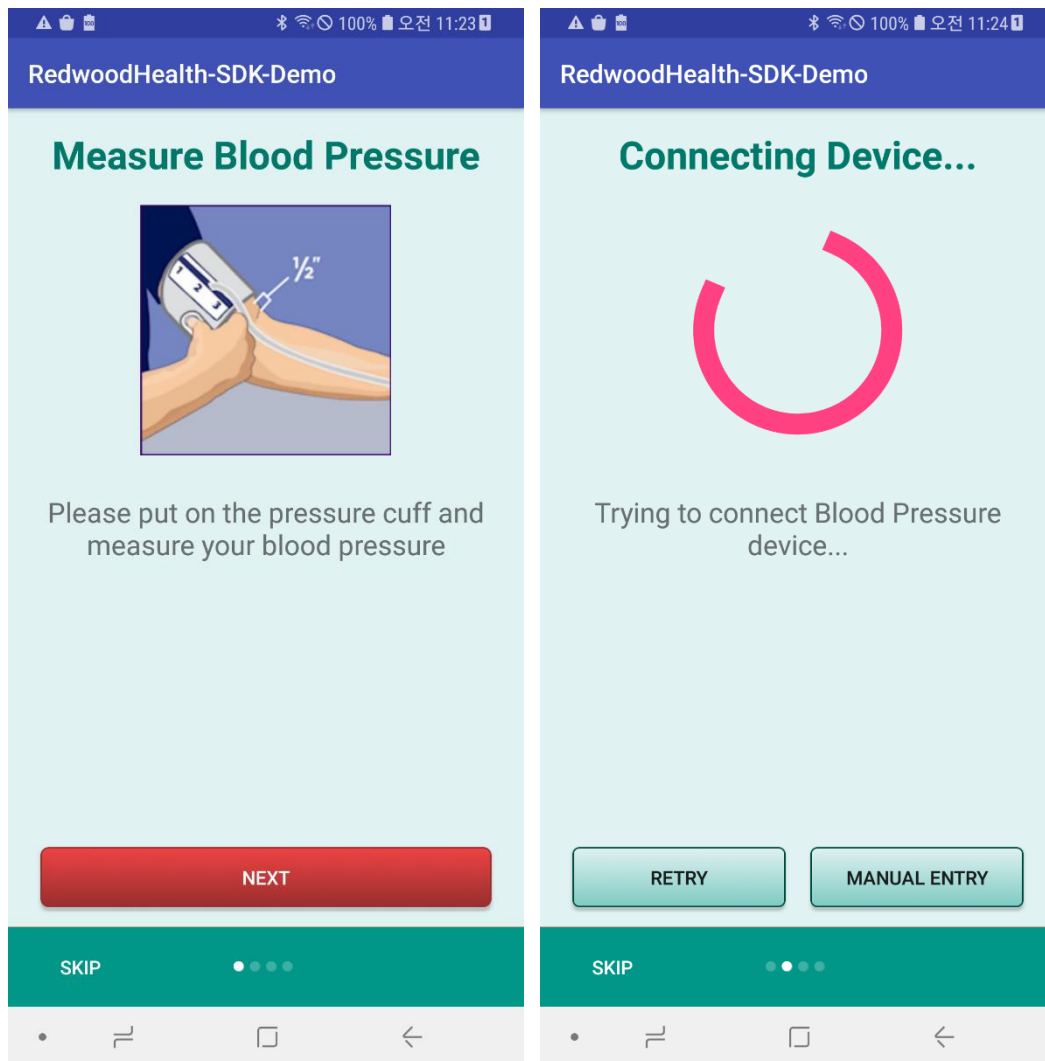
                byte flags = data[idx++];

                // 0: mmHg 1: kPa
                boolean kPa = (flags & 0x01) > 0;
                // 0: No Timestamp info 1: With Timestamp info
                boolean timestampFlag = (flags & 0x02) > 0;
                // 0: No PlseRate info 1: With PulseRate info
                boolean pulseRateFlag = (flags & 0x04) > 0;
                // 0: No UserID info 1: With UserID info
                boolean userIdFlag = (flags & 0x08) > 0;
                // 0: No MeasurementStatus info 1: With MeasurementStatus info
                boolean measurementStatusFlag = (flags & 0x10) > 0;

                // Set BloodPressureMeasurement unit
                String unit;
                if (kPa) {
                    unit = "kPa";
```

```
    } else {  
        unit = "mmHg";  
    }  
  
    // Parse Blood Pressure Measurement  
    short systolicVal = 0;  
    short diastolicVal = 0;  
    short meanApVal = 0;  
  
    System.arraycopy(data, idx, buf, 0, 2);  
    idx += 2;  
    byteBuffer = ByteBuffer.wrap(buf);  
    byteBuffer.order(ByteOrder.LITTLE_ENDIAN);  
    systolicVal = byteBuffer.getShort();  
  
    System.arraycopy(data, idx, buf, 0, 2);  
    idx += 2;  
    byteBuffer = ByteBuffer.wrap(buf);  
    byteBuffer.order(ByteOrder.LITTLE_ENDIAN);  
    diastolicVal = byteBuffer.getShort();  
  
    System.arraycopy(data, idx, buf, 0, 2);  
    idx += 2;  
    byteBuffer = ByteBuffer.wrap(buf);  
    byteBuffer.order(ByteOrder.LITTLE_ENDIAN);  
    meanApVal = byteBuffer.getShort();  
    activity.bloodPressure = new double[3];  
    activity.bloodPressure[0] = systolicVal;  
    activity.bloodPressure[1] = diastolicVal;
```

3.7.2. 실행결과



3.8. 혈압 측정결과

디바이스로 전송된 혈압데이터를 화면에 출력하고 서버에 전송하는 샘플 코드입니다.

3.8.1. Fragment_3_Read_BP.java

혈압 측정 데이터를 화면에 표시하고 서버에 결과를 전송합니다.

```
//혈압 측정 데이터를 화면에 표시
private void updateView() {
    if (activity.bloodPressure == null) {
        textViewSystolic.setText("---");
        textViewDiastolic.setText("---");
        textViewPulse.setText("---");
    } else {
        if(activity.bloodPressure[0]!=-1)
            textViewSystolic.setText(String.valueOf(activity.bloodPressure[0]));
        if(activity.bloodPressure[1]!=-1)
            textViewDiastolic.setText(String.valueOf(activity.bloodPressure[1]));
        if(activity.bloodPressure[2]!=-1)
            textViewPulse.setText(String.valueOf(activity.bloodPressure[2]));
    }
    if (activity.heartRate == null) {
        textViewPulseRate.setText("---");
        textViewIrregularPulse.setText("---");
    } else {
        if(activity.heartRate[0]!=-1)
            textViewPulseRate.setText(String.valueOf(activity.heartRate[0]));
        if(activity.heartRate[1]!=-1)
            textViewIrregularPulse.setText(String.valueOf(activity.heartRate[1]));
    }
    if (activity.battery == null) {
        textViewBattery.setText("---");
    } else {
        textViewBattery.setText(String.valueOf(activity.battery[0]));
    }
}
```



```
if (activity.activity == null)
    textViewMovement.setText("---");
else textViewMovement.setText(String.valueOf(activity.activity[0]));
setupButton();
}
```

//서버에 혈압측정 결과를 전송합니다.

```
public PHRObservation makeResource(){
    PHRObservation obj = new PHRObservation();
    try{
        obj.setStatusText("preliminary");//초기 등록
        obj.setSubjectText(MainActivity.gPatientId);
        obj.setObservationId(mResourceId);

        long now = System.currentTimeMillis();
        Date mDate = new Date(now);
        obj.setEffectiveDateTime(mDate);
        obj.setBodySiteText(spinner.getSelectedItem().toString());
        List<PHRObservationComponentComponent> componentComponentList =
            new ArrayList<PHRObservationComponentComponent>();

        //code 할당
        {
            PHRCodeableConcept phrCode = new PHRCodeableConcept();
            List<PHRCoding> phrCodings = new ArrayList<PHRCoding>();
            PHRCoding phrCoding = new PHRCoding();
            phrCoding.setSystem("https://www.hins.or.kr");
            phrCoding.setCode("H02138365");
            phrCoding.setDisplay("Blood pressure systolic & diastolic");
            phrCodings.add(phrCoding);
            phrCode.setCoding(phrCodings);
            phrCode.setText("혈압(수축기&이완기)");
            obj.setCode(phrCode);
        }
    }
}
```

```

    }

    //////////// 이완기(확장기) 혈압 ////////////
    {
        PHRObservationComponentComponent componentComponent =
            new PHRObservationComponentComponent();

        //code
        PHRCodeableConcept codeDiastolic = new PHRCodeableConcept();
        List<PHRCoding> codingsDiastolic = new ArrayList<PHRCoding>();
        PHRCoding coding = new PHRCoding();
        coding.setSystem("https://www.hins.or.kr");
        coding.setCode("H02171908");
        coding.setDisplay("DBP");
        codingsDiastolic.add(coding);
        codeDiastolic.setCoding(codingsDiastolic);
        codeDiastolic.setText("혈압(이완기)");
        componentComponent.setCode(codeDiastolic);

        //value[x]
        PHRQuantity quantity = new PHRQuantity();
        quantity.setValue(Double.parseDouble(String.valueOf(activity.bloodPressure[1])));
        quantity.setUnit("mmHg");
        quantity.setCode("mm[Hg]");
        quantity.setSystem("http://unitsofmeasure.org");
        componentComponent.setValue(quantity);

        //interpretation
        PHRCodeableConcept interpretation = new PHRCodeableConcept();
        interpretation.setText("Normal");
        PHRCoding codingInterpretation = new PHRCoding();
        codingInterpretation.setSystem("http://hl7.org/fhir/v2/0078");
        codingInterpretation.setCode("N");
        codingInterpretation.setDisplay("normal");
        interpretation.addCoding(codingInterpretation);
    }

```

```
        componentComponent.setInterpretation(interpretation);

        componentComponentList.add(componentComponent);
    }

    //////////// 수축기 혈압 ////////////
    {
        PHRObservationComponentComponent componentComponent =
            new PHRObservationComponentComponent();

        //code
        PHRCodeableConcept codeSystolic = new PHRCodeableConcept();
        List<PHRCoding> codings = new ArrayList<PHRCoding>();
        PHRCoding coding = new PHRCoding();
        coding.setSystem("https://www.hins.or.kr");
        coding.setCode("H02178351");
        coding.setDisplay("SBP");
        codings.add(coding);
        codeSystolic.setCoding(codings);
        codeSystolic.setText("혈압(수축기)");
        componentComponent.setCode(codeSystolic);

        //value[x]
        PHRQuantity quantity = new PHRQuantity();
        quantity.setValue(Double.parseDouble(String.valueOf(activity.bloodPressure[0])));
        quantity.setUnit("mmHg");
        quantity.setCode("mm[Hg]");
        quantity.setSystem("http://unitsofmeasure.org");
        componentComponent.setValue(quantity);

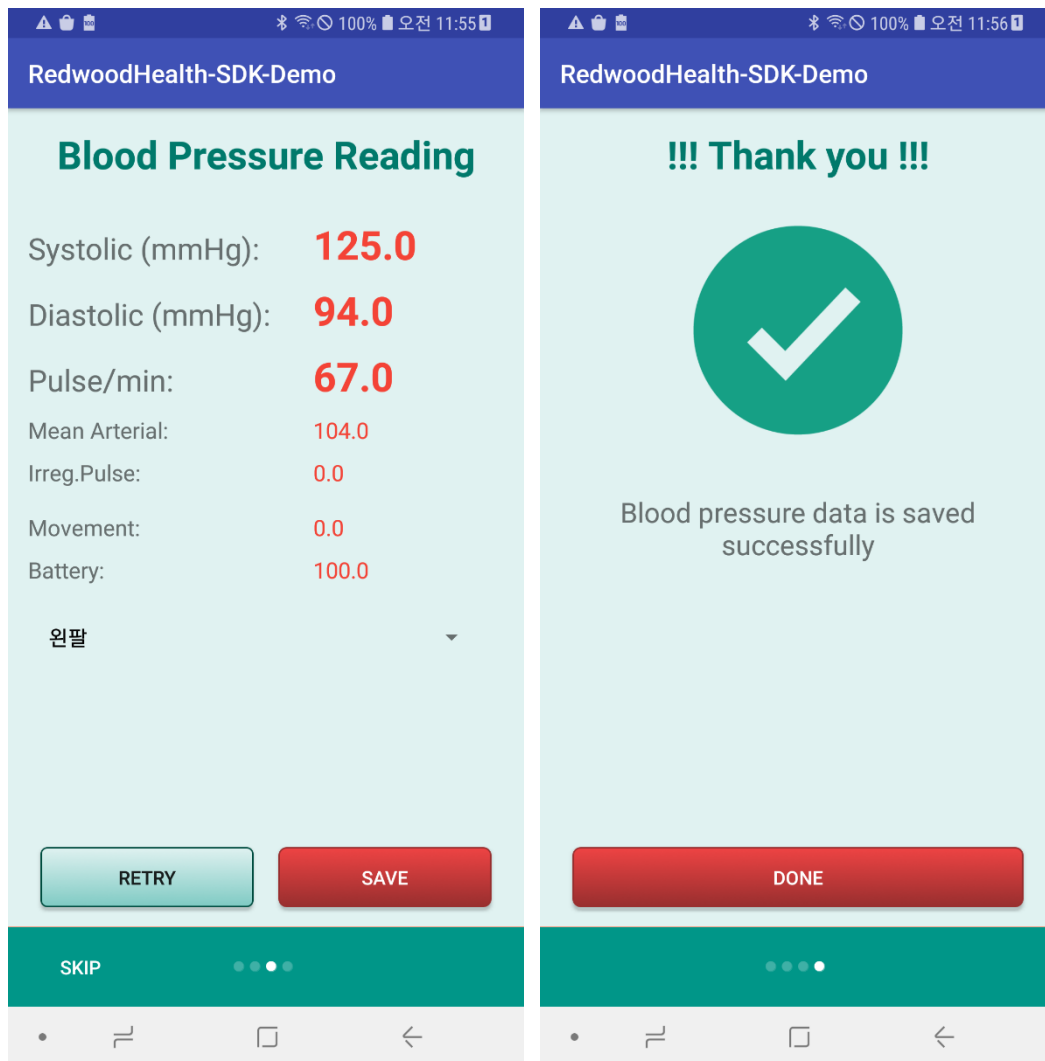
        //interpretation
        PHRCodeableConcept interpretation = new PHRCodeableConcept();
        interpretation.setText("Below low normal");
        PHRCoding codingInterpretation = new PHRCoding();
        codingInterpretation.setSystem("http://hl7.org/fhir/v2/0078");
```

```
        codingInterpretation.setCode("L");
        codingInterpretation.setDisplay("low");
        interpretation.addCoding(codingInterpretation);
        componentComponent.setInterpretation(interpretation);
        componentComponentList.add(componentComponent);
    }

    obj.setComponentList(componentComponentList);

} catch (Exception ex) {
    ex.printStackTrace();
}
return obj;
}
```

3.8.2. 실행결과



3.9. 기기 찾기

미밴드와 연결하기 전에 다음과 같은 작업이 필요합니다.

- 비공식 미핏 앱 다운로드(<https://www.freemyband.com/2019/08/mi-band-4-auth-key.html>) 후 설치
- 비공식 미핏 앱으로 로그인
- 미핏4하고 연결한 후 미핏 앱 삭제
- 폰스토리지 Phone/freemyband 폴더 내 auth key를 포함한 텍스트 파일이 저장되어 있는지 확인
- 기기 스캔 후 미밴드를 검색 한 후 Auth Key에 아까 받은

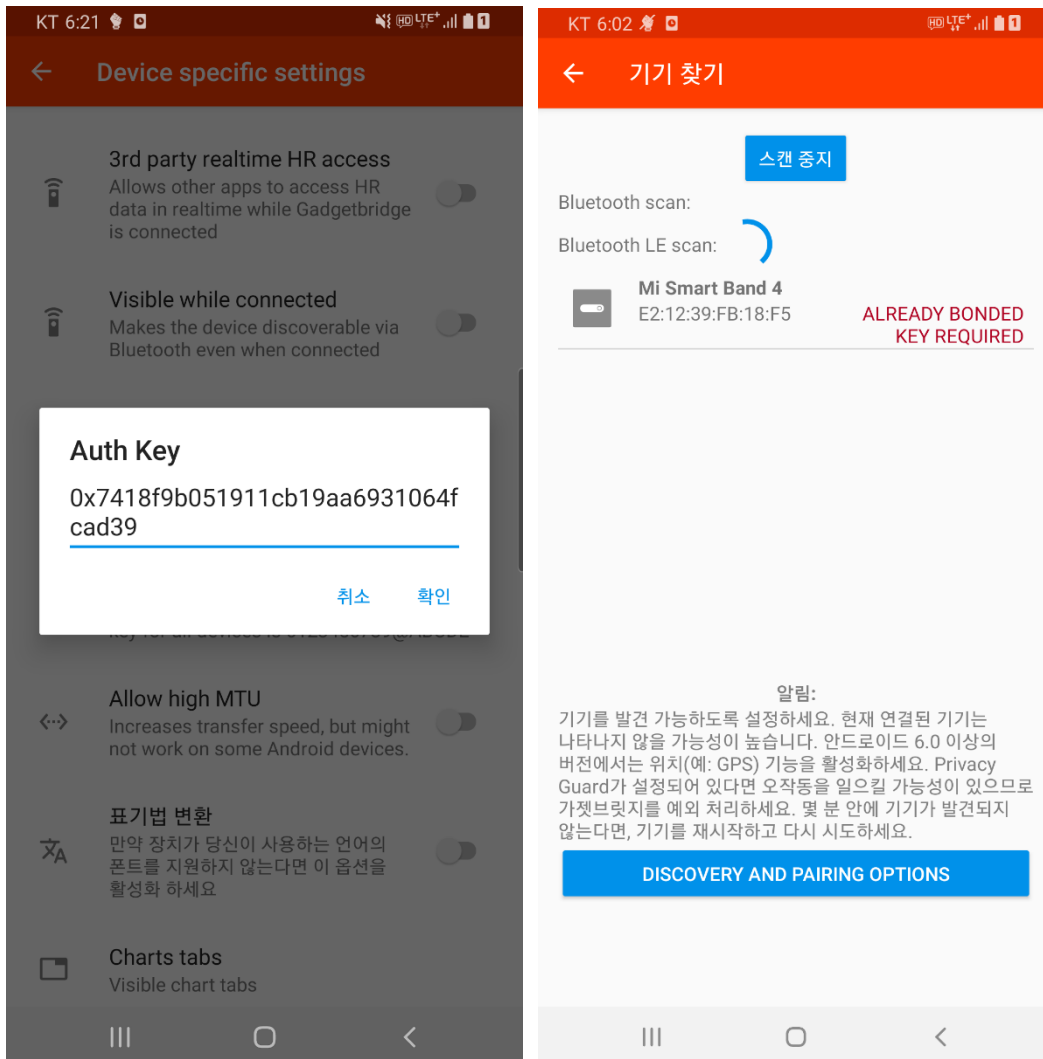
텍스트 파일인 E2:12:39:FB:18:F5:[authkey]에서 제일 앞에 0x를 추가한 0x[authkey] 키를 입력하면 연결완료

3.9.1. DiscoveryActivity.java

기기 목록을 검색하는 샘플 코드입니다.

```
BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
if (device != null) {
    int bondState=intent.getIntExtra(BluetoothDevice.EXTRA_BOND_STATE
    , BluetoothDevice.BOND_NONE);
    if (bondState == BluetoothDevice.BOND_BONDED) {
        BondingUtil.handleDeviceBonded((BondingInterface)context,getCandidateFromMAC(device));
    }
}
```

3.9.2. 실행결과



3.10. MiBand 블루투스 페어링

Miband 디바이스와 Bluetooth pairing 을 연결하는 샘플소스입니다.

3.10.1.MiBandParingActivity.java

디바이스와 블루투스 페어링을 합니다.

```
private void startPairing() {
    isPairing = true;
    message.setText(getString(R.string.pairing, deviceCandidate));

    if (!BondingUtil.shouldUseBonding()) {
        BondingUtil.attemptToFirstConnect(getCurrentTarget().getDevice());
        return;
    }

    BondingUtil.tryBondThenComplete(this, deviceCandidate);
}

public static void tryBondThenComplete(BondingInterface
bondingInterface, GBDeviceCandidate deviceCandidate) {
    bondingInterface.registerBroadcastReceivers();
    BluetoothDevice device = deviceCandidate.getDevice();
    int bondState = device.getBondState();

    if (bondState == BluetoothDevice.BOND_BONDED) {
        GB.toast(bondingInterface.getContext().getString(R.string.pairing_already_bonded,
device.getName(), device.getAddress()), Toast.LENGTH_SHORT, GB.INFO);

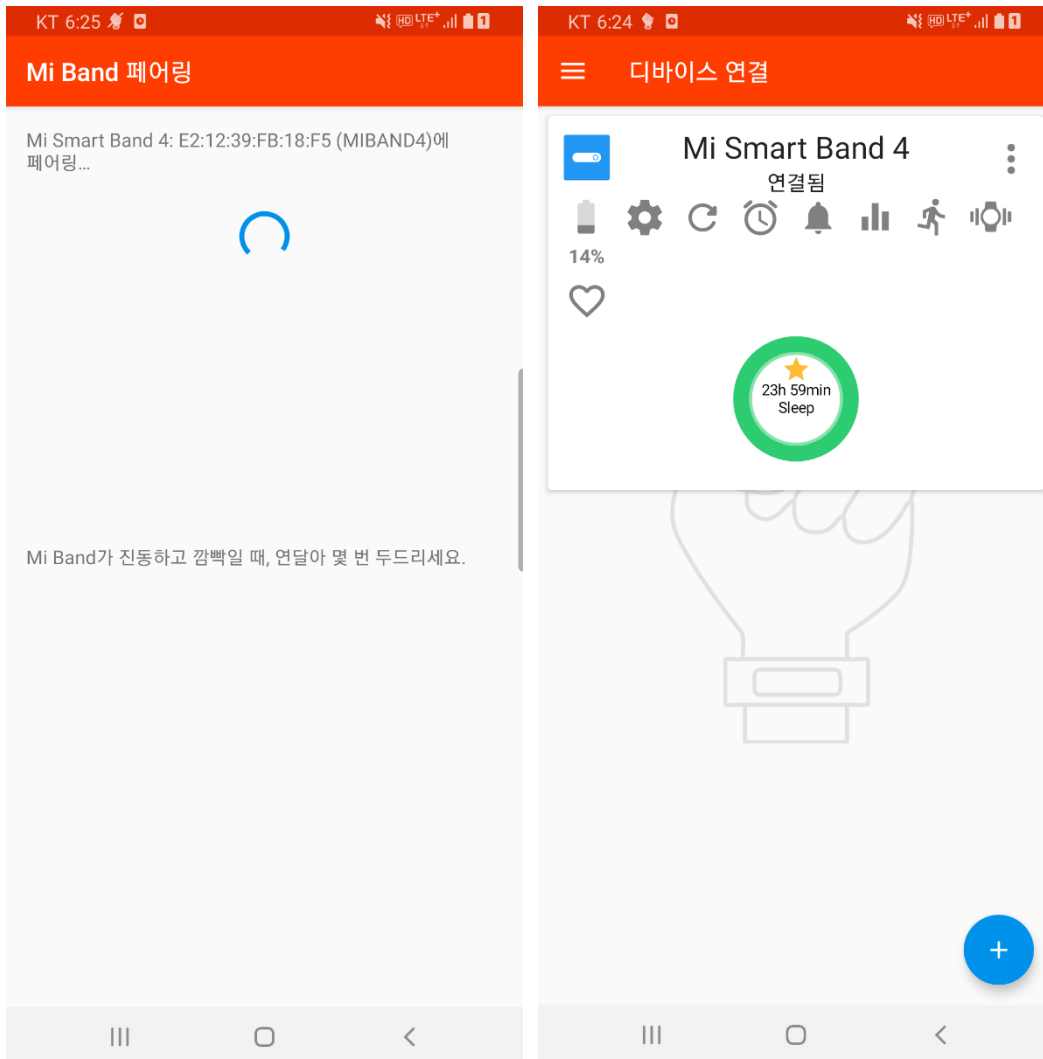
        //noinspection StatementWithEmptyBody
        if (GBApplication.getPrefs().getBoolean("enable_companiondevice_pairing", true)
&& Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
```



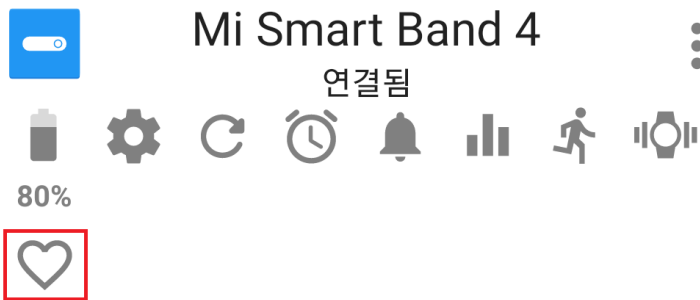
```
// If CompanionDeviceManager is enabled, skip connection and go bond
// TODO: It would theoretically be nice to check if it's already been granted,
// but re-bond works
    } else {
        attemptToFirstConnect(bondingInterface.getCurrentTarget().getDevice());
        return;
    }
} else if (bondState == BluetoothDevice.BOND_BONDING) {
    GB.toast(bondingInterface.getContext(),
        bondingInterface.getContext().getString(R.string.pairing_in_progress,
            device.getName(), device.getAddress()), Toast.LENGTH_LONG, GB.INFO);
    return;
}

GB.toast(bondingInterface.getContext(),
    bondingInterface.getContext().getString(R.string.pairing_creating_bond_with,
        device.getName(), device.getAddress()), Toast.LENGTH_LONG, GB.INFO);
toast(bondingInterface.getContext(),
    bondingInterface.getContext().getString(R.string.discovery_attempting_to_pair,
        deviceCandidate.getName()), Toast.LENGTH_SHORT, GB.INFO);
if (GBApplication.getPrefs().getBoolean("enable_companiondevice_pairing", true)
    && Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
    companionDeviceManagerBond(bondingInterface, deviceCandidate);
} else {
    bluetoothBond(bondingInterface, deviceCandidate);
}
}
```

3.10.2. 실행결과



3.11. 심박수 측정결과



화면 상단에 하트 모양의 아이콘을 선택하여 심박수를 측정합니다.

디바이스로 전송된 심박수데이터를 화면에 출력하고 서버에 전송하는 샘플 코드입니다.

3.11.1.HeartRateDialog.java

심박수 측정 데이터를 화면에 표시하고 서버에 결과를 전송합니다.

```
private void setMeasurementResults(Serializable result) {
    heart_rate_dialog_results_layout.setVisibility(View.VISIBLE);
    heart_rate_dialog_loading_layout.setVisibility(View.GONE);
    heart_rate_dialog_label.setText(getContext().getString(R.string.heart_rate_result));

    if (result instanceof ActivitySample) {
        ActivitySample sample = (ActivitySample) result;
        heart_rate_hr.setVisibility(View.VISIBLE);
        if (HeartRateUtils.getInstance().isValidHeartRateValue(sample.getHeartRate())) {
            heart_rate_widget_hr_value.setText(String.valueOf(sample.getHeartRate()));
            if (isFirst) {
                createResource(String.valueOf(sample.getHeartRate()));
                isFirst = false;
            }
        }
    }
}

public boolean createResource(String heartRate){
```

```
try{
    PHRObservation obj = makeResource(heartRate);
    //혈압측정
    ObservationAddAsyncTask addAsyncTask =
    new ObservationAddAsyncTask((ControlCenterv2)context,
    obj, PHRUtils.OperationType.CREATE);
    addAsyncTask.setResourceType(mResourceType);
    addAsyncTask.execute("Miband");
}catch (Exception e){
    e.printStackTrace();
    return false;
}
return true;
}
```

```
public PHRObservation makeResource(String heartRate){
```

```
    PHRObservation obj = new PHRObservation();
    try{
        obj.setSubjectText(MainActivity.gPatientId);
        obj.setStatusText("preliminary");//초기 등록
        obj.setObservationId(mResourceId);
        long now = System.currentTimeMillis();
        Date mDate = new Date(now);
        obj.setEffectiveDateTime(mDate);

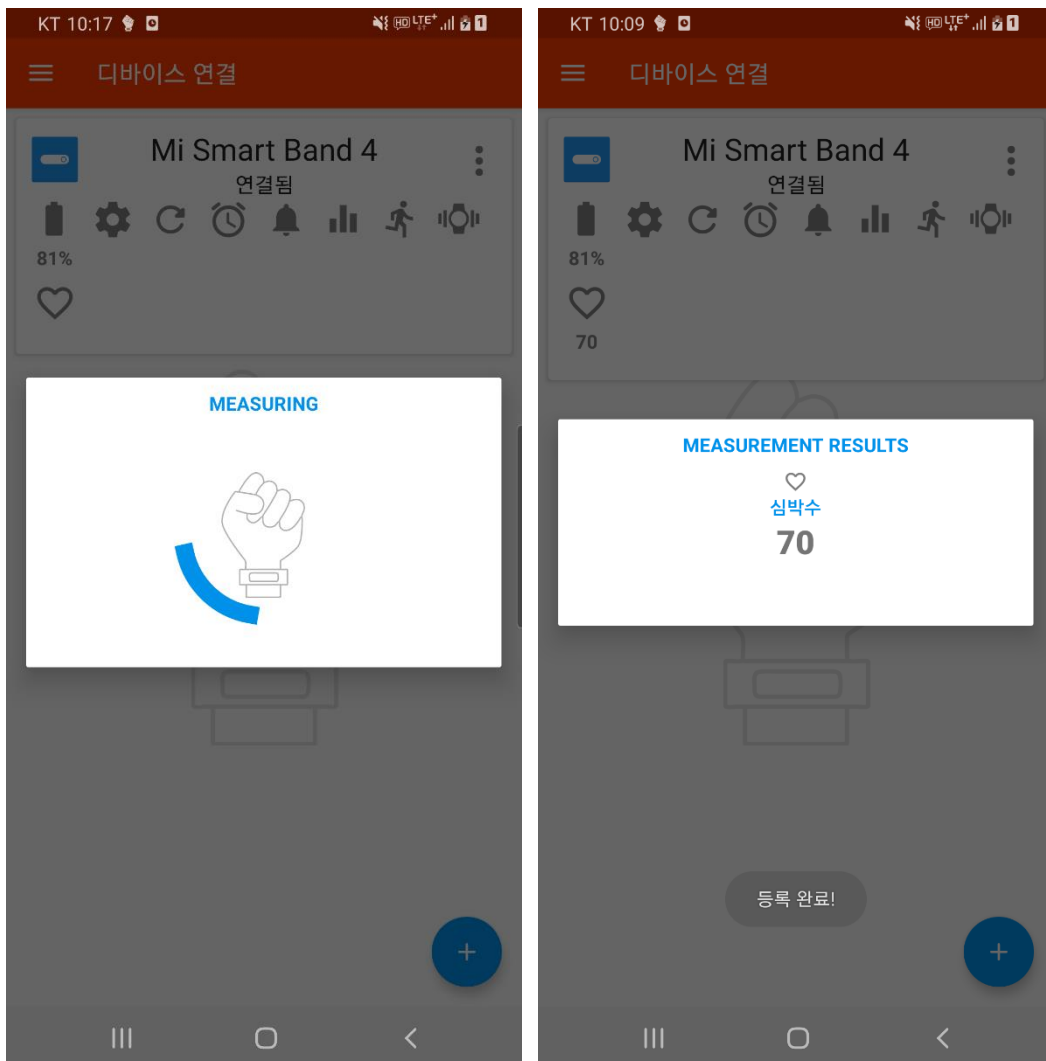
        //code 할당
        {
            PHRCodeableConcept code = new PHRCodeableConcept();
            List<PHRCoding> codings = new ArrayList<PHRCoding>();
            PHRCoding coding = new PHRCoding();
            coding.setSystem("https://www.hins.or.kr");
            coding.setCode("H00734513");
            coding.setDisplay("Heart rate");
            codings.add(coding);
        }
    }
}
```

```
        code.setCoding(codings);
        code.setText("Heart rate");

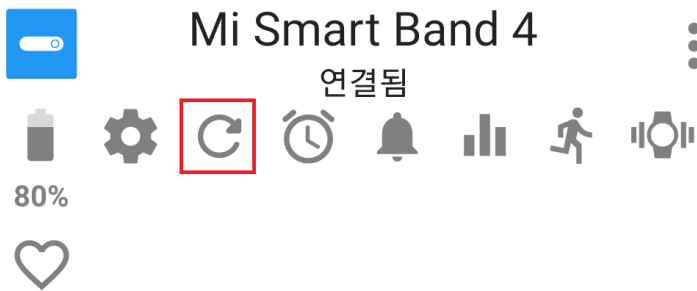
        obj.setCode(code);
    }

    //value[x]
    PHRQuantity quantity = new PHRQuantity();
    quantity.setValue(Double.parseDouble(heartRate));
    quantity.setUnit("beats/minute");
    quantity.setCode("/min");
    quantity.setSystem("http://unitsofmeasure.org");
    obj.setValue(quantity);
} catch (Exception ex) {
    ex.printStackTrace();
}
return obj;
}
```

3.11.2. 실행결과



3.12. 활동데이터 수집



화면 상단에 왼쪽에서 세번째 리프레쉬 아이콘을 선택하여 활동데이터를 가져옵니다.

디바이스로 전송된 활동데이터(걸음수, 거리, 수면)를 화면에 출력하고 서버에 전송하는 샘플 코드입니다.

3.12.1. GBDeviceAdapterv2.java

```
public boolean createResource(String steps){
    try{
        PHRObservation obj = makeResource(steps);
        //혈압측정
        ObservationAddAsyncTask addAsyncTask = new ObservationAddAsyncTask((ControlCenterv
2)context, obj, PHRUtils.OperationType.CREATE);
        addAsyncTask.execute("Miband");
    }catch (Exception e){
        e.printStackTrace();
        return false;
    }
    return true;
}

public PHRObservation makeResource(String steps){

    PHRObservation obj = new PHRObservation();
    try{
        obj.setSubjectText(MainActivity.gPatientId);
```

```
obj.setStatusText("preliminary");//초기 등록

long now = System.currentTimeMillis();
Date mDate = new Date(now);
obj.setEffectiveDateTime(mDate);

//code 할당

{
    PHRCodeableConcept code = new PHRCodeableConcept();
    List<PHRCoding> codings = new ArrayList<PHRCoding>();
    PHRCoding coding = new PHRCoding();
    coding.setSystem("https://www.hins.or.kr");
    coding.setCode("H01818711");
    coding.setDisplay("Walking");
    codings.add(coding);
    code.setCoding(codings);
    code.setText("Walking");

    obj.setCode(code);
}

//value[x]
PHRQuantity quantity = new PHRQuantity();
quantity.setValue(Double.parseDouble(steps));
quantity.setUnit("STEPS");
quantity.setCode("{STEPS}");
quantity.setSystem("http://unitsofmeasure.org");
obj.setValue(quantity);
} catch (Exception ex) {
    ex.printStackTrace();
}
return obj;
}
```


3.12.2. 실행결과

