**ASSIGNMENT- 2**

1. Write a program that identifies the Bluetooth devices in the wireless range.

***Discover\_Bluetooth.java***

**package** com.example.madpractical;  
  
**import** androidx.annotation.RequiresApi;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.Manifest;  
**import** android.bluetooth.BluetoothAdapter;  
**import** android.bluetooth.BluetoothDevice;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
  
**import** java.util.ArrayList;  
  
**public class** Discover\_Bluetooth **extends** AppCompatActivity {  
 **private static final** String ***TAG*** = **"MainActivity"**;  
 BluetoothAdapter **bluetoothAdapter**;  
 **public** DeviceListAdapter **deviceListAdapter**;  
 ListView **listView**;  
 ArrayList<BluetoothDevice> **deviceArrayList**;  
  
 */\*\*  
 \* Broadcast Receiver for changes made to bluetooth states such as:  
 \* 1) Discoverability mode on/off or expire.  
 \*/* **final** BroadcastReceiver **btReceiver** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 **if** (action.equals(**bluetoothAdapter**.***ACTION\_STATE\_CHANGED***)) {  
 **final int** state = intent.getIntExtra(BluetoothAdapter.***EXTRA\_STATE***, **bluetoothAdapter**.***ERROR***);  
  
 **switch** (state) {  
 **case** BluetoothAdapter.***STATE\_OFF***:  
 Log.*d*(***TAG***, **"btReceiver: STATE OFF"**);  
 **break**;  
 **case** BluetoothAdapter.***STATE\_TURNING\_OFF***:  
 Log.*d*(***TAG***, **"btReceiver: STATE TURNING OFF"**);  
 **break**;  
 **case** BluetoothAdapter.***STATE\_ON***:  
 Log.*d*(***TAG***, **"btReceiver: STATE ON"**);  
 **break**;  
 **case** BluetoothAdapter.***STATE\_TURNING\_ON***:  
 Log.*d*(***TAG***, **"btReceiver: STATE TURNING ON"**);  
 **break**;  
 }  
 }  
 *//if(BluetoothDevice.ACTION\_FOUND.equals(action)){  
 //BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA\_DEVICE);  
 // add the name to the list  
 //BTArrayAdapter.add(device.getName() + "\n" + device.getAddress());  
 //mBTArrayAdapter.notifyDataSetChanged();  
 //}* }  
 };  
  
 */\*\*  
 \* Broadcast Receiver for listing devices that are not yet paired  
 \* -Executed by btnDiscover() method.  
 \*/* **final** BroadcastReceiver **enable\_discovery\_receiver** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 **if** (action.equals(**bluetoothAdapter**.***ACTION\_SCAN\_MODE\_CHANGED***)) {  
 **final int** state = intent.getIntExtra(BluetoothAdapter.***EXTRA\_STATE***, **bluetoothAdapter**.***ERROR***);  
  
 **switch** (state) {  
 **case** BluetoothAdapter.***SCAN\_MODE\_CONNECTABLE\_DISCOVERABLE***:  
 Log.*d*(***TAG***, **"enable\_discovery\_receiver: Discoverability Enabled"**);  
 **break**;  
 **case** BluetoothAdapter.***SCAN\_MODE\_CONNECTABLE***:  
 Log.*d*(***TAG***, **"enable\_discovery\_receiver: Discoverability Enabled. Able to receive Connect"**);  
 **break**;  
 **case** BluetoothAdapter.***SCAN\_MODE\_NONE***:  
 Log.*d*(***TAG***, **"enable\_discovery\_receiver: Discoverability Disabled. Not able to receive Connect"**);  
 **break**;  
 **case** BluetoothAdapter.***STATE\_CONNECTING***:  
 Log.*d*(***TAG***, **"enable\_discovery\_receiver: Connecting..."**);  
 **break**;  
 **case** BluetoothAdapter.***STATE\_CONNECTED***:  
 Log.*d*(***TAG***, **"enable\_discovery\_receiver: Connected."**);  
 **break**;  
 }  
 }  
 *//if(BluetoothDevice.ACTION\_FOUND.equals(action)){  
 //BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA\_DEVICE);  
 // add the name to the list  
 //BTArrayAdapter.add(device.getName() + "\n" + device.getAddress());  
 //mBTArrayAdapter.notifyDataSetChanged();  
 //}* }  
 };  
  
 */\*\*  
 \* Broadcast Receiver for listing devices that are not yet paired  
 \* -Executed by btnDiscover() method.  
 \*/* **private** BroadcastReceiver **mBroadcastReceiver3** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 **final** String action = intent.getAction();  
 Log.*d*(***TAG***, **"onReceive: ACTION FOUND."**);  
  
 **if** (action.equals(BluetoothDevice.***ACTION\_FOUND***)) {  
 BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.***EXTRA\_DEVICE***);  
 Log.*d*(***TAG***, **deviceArrayList**.size() + **""**);  
  
 Log.*d*(***TAG***, **"onReceive: "** + device.getName() + **": "** + device.getAddress());  
 **if** (!**deviceArrayList**.contains(device)) **deviceArrayList**.add(device);  
 Log.*d*(***TAG***, **deviceArrayList**.size() + **""**);  
 **deviceListAdapter**.notifyDataSetChanged();  
 }  
 }  
 };  
  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 Log.*d*(***TAG***, **"onDestroy called"**);  
 unregisterReceiver(**btReceiver**);  
 unregisterReceiver(**enable\_discovery\_receiver**);  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_discover\_bluetooth***);  
 Button on\_off = findViewById(R.id.***on\_off***);  
 Button enable\_discover = findViewById(R.id.***enable\_discover***);  
 **listView** = findViewById(R.id.***device\_list***);  
 **deviceArrayList** = **new** ArrayList<>();  
 **deviceListAdapter** = **new** DeviceListAdapter(Discover\_Bluetooth.**this**, R.layout.***device\_adapter\_list***, **deviceArrayList**);  
 **listView**.setAdapter(**deviceListAdapter**);  
 enable\_discover.setOnClickListener(v -> {  
 btnEnableDisable\_Discoverable(v);  
 });  
 **bluetoothAdapter** = BluetoothAdapter.*getDefaultAdapter*();  
 on\_off.setOnClickListener(v -> {  
 enableBluetooth();  
 });  
 Button discover = findViewById(R.id.***discover\_devices***);  
 discover.setOnClickListener(v -> btnDiscover(v));  
 }  
  
 **public void** enableBluetooth() {  
 **if** (**bluetoothAdapter** == **null**) {  
 Log.*d*(**"enableBlueTooth"**, **"enabledisableBT: your device doesn't have BT"**);  
 }  
 **if** (!**bluetoothAdapter**.isEnabled()) {  
 Intent intent = **new** Intent(BluetoothAdapter.***ACTION\_REQUEST\_ENABLE***);  
 startActivity(intent);  
  
 IntentFilter BTintent = **new** IntentFilter(BluetoothAdapter.***ACTION\_STATE\_CHANGED***);  
 registerReceiver(**btReceiver**, BTintent);  
 }  
 **if** (**bluetoothAdapter**.isEnabled()) {  
 **bluetoothAdapter**.disable();  
  
 IntentFilter BTintent = **new** IntentFilter(BluetoothAdapter.***ACTION\_STATE\_CHANGED***);  
 registerReceiver(**btReceiver**, BTintent);  
 }  
 }  
  
 **public void** btnEnableDisable\_Discoverable(View view) {  
 Log.*d*(***TAG***, **"btnEnableDisable\_Discoverable: Making device discoverable for 300 seconds"**);  
  
 Intent intent = **new** Intent(BluetoothAdapter.***ACTION\_REQUEST\_DISCOVERABLE***);  
 intent.putExtra(BluetoothAdapter.***EXTRA\_DISCOVERABLE\_DURATION***, 300);  
 startActivity(intent);  
  
 IntentFilter intentFilter = **new** IntentFilter(**bluetoothAdapter**.***ACTION\_SCAN\_MODE\_CHANGED***);  
 registerReceiver(**enable\_discovery\_receiver**, intentFilter);  
 }  
  
 **public void** btnDiscover(View view) {  
 Log.*d*(***TAG***, **"btnDiscover: Looking for unpaired devices."**);  
  
 **if** (**bluetoothAdapter**.isDiscovering()) {  
 **bluetoothAdapter**.cancelDiscovery();  
 Log.*d*(***TAG***, **"btnDiscover: Canceling discovery."**);  
  
 *//check BT permissions in manifest* checkBTPermissions();  
  
 **bluetoothAdapter**.startDiscovery();  
 IntentFilter discoverDevicesIntent = **new** IntentFilter(BluetoothDevice.***ACTION\_FOUND***);  
 registerReceiver(**mBroadcastReceiver3**, discoverDevicesIntent);  
 }  
 **if** (!**bluetoothAdapter**.isDiscovering()) {  
  
 *//check BT permissions in manifest* checkBTPermissions();  
  
 **bluetoothAdapter**.startDiscovery();  
 IntentFilter discoverDevicesIntent = **new** IntentFilter(BluetoothDevice.***ACTION\_FOUND***);  
 registerReceiver(**mBroadcastReceiver3**, discoverDevicesIntent);  
 }  
 }  
  
 */\*\*  
 \* This method is required for all devices running API23+  
 \* Android must programmatically check the permissions for bluetooth. Putting the proper permissions  
 \* in the manifest is not enough.  
 \* <p>  
 \* NOTE: This will only execute on versions > LOLLIPOP because it is not needed otherwise.  
 \*/* **private void** checkBTPermissions() {  
 **if** (Build.VERSION.***SDK\_INT*** > Build.VERSION\_CODES.***LOLLIPOP***) {  
 **int** permissionCheck = **this**.checkSelfPermission(**"Manifest.permission.ACCESS\_FINE\_LOCATION"**);  
 permissionCheck += **this**.checkSelfPermission(**"Manifest.permission.ACCESS\_COARSE\_LOCATION"**);  
 **if** (permissionCheck != 0) {  
  
 **this**.requestPermissions(**new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***, Manifest.permission.***ACCESS\_COARSE\_LOCATION***}, 1001); *//Any number* }  
 } **else** {  
 Log.*d*(***TAG***, **"checkBTPermissions: No need to check permissions. SDK version < LOLLIPOP."**);  
 }  
 }  
  
}

***DevicelistAdapter.java***

**package** com.example.madpractical;  
  
**import** android.bluetooth.BluetoothDevice;  
**import** android.content.Context;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
  
**public class** DeviceListAdapter **extends** ArrayAdapter<BluetoothDevice> {  
  
 **private** LayoutInflater **mLayoutInflater**;  
 **private int mViewResourceId**;  
  
 **public** DeviceListAdapter(Context context, **int** tvResourceId, ArrayList<BluetoothDevice> devices) {  
 **super**(context, tvResourceId, devices);  
 **mLayoutInflater** = LayoutInflater.*from*(context);  
 **mViewResourceId** = tvResourceId;  
 }  
  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 convertView = **mLayoutInflater**.inflate(**mViewResourceId**, parent, **false**);  
  
 BluetoothDevice device = getItem(position);  
 TextView deviceName = convertView.findViewById(R.id.***tvDeviceName***);  
 TextView deviceAdress = convertView.findViewById(R.id.***tvDeviceAddress***);  
 deviceName.setText(device.getName());  
 deviceAdress.setText(device.getAddress());  
  
 **return** convertView;  
 }  
  
}

***device\_adapter\_list.xml***

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/tvDeviceName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="00:00"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
  
 <**TextView  
 android:id="@+id/tvDeviceAddress"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginBottom="20dp"  
 android:text="00:00"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
  
</**LinearLayout**>

***activity\_discover\_bluetooth.xml***

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/gradient"  
 tools:context=".Discover\_Bluetooth"**>  
  
 <**ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
  
 <**Button  
 android:id="@+id/on\_off"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="ON/OFF"  
 android:textSize="24sp"** />  
  
 <**Button  
 android:id="@+id/enable\_discover"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="Enable discover"  
 android:textSize="24sp"** />  
  
 <**Button  
 android:id="@+id/discover\_devices"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="discover devices"  
 android:textSize="24sp"** />  
  
 <**ListView  
 android:id="@+id/device\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="400dp"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginTop="20dp"  
 android:foregroundGravity="center\_horizontal"** />  
  
 </**LinearLayout**>  
 </**ScrollView**>  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

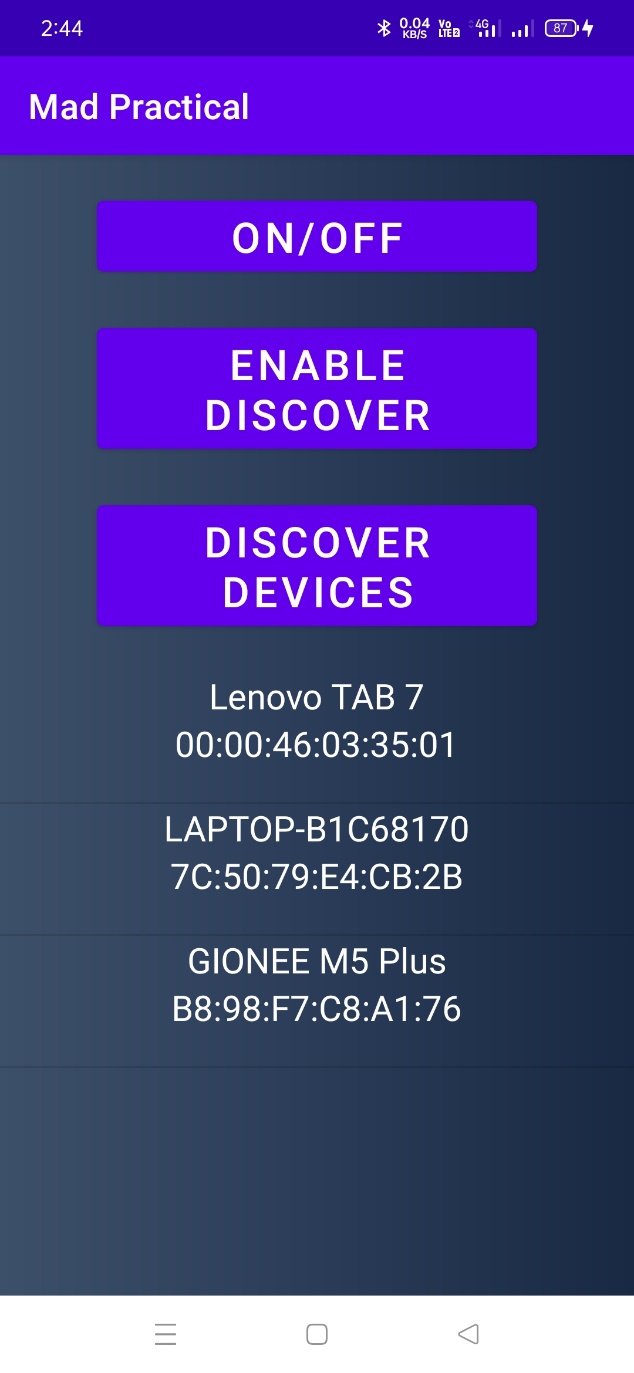
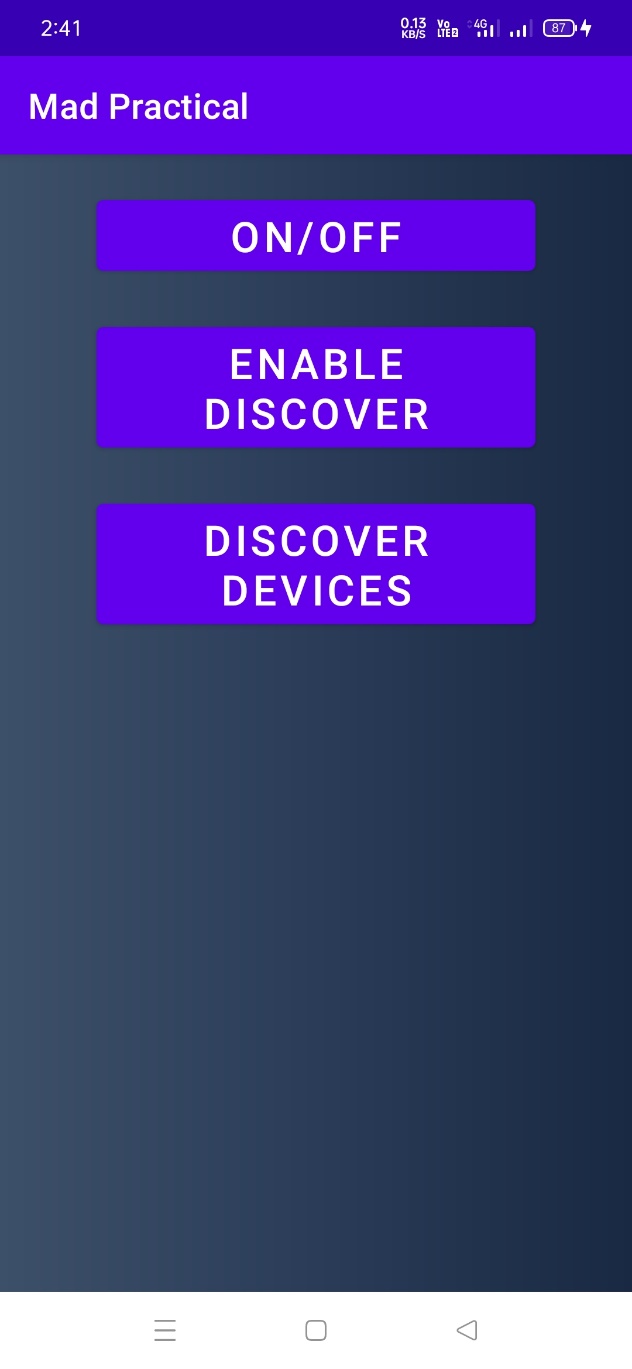


Fig 1 – Bluetooth options Fig 2 – Devices Discovery

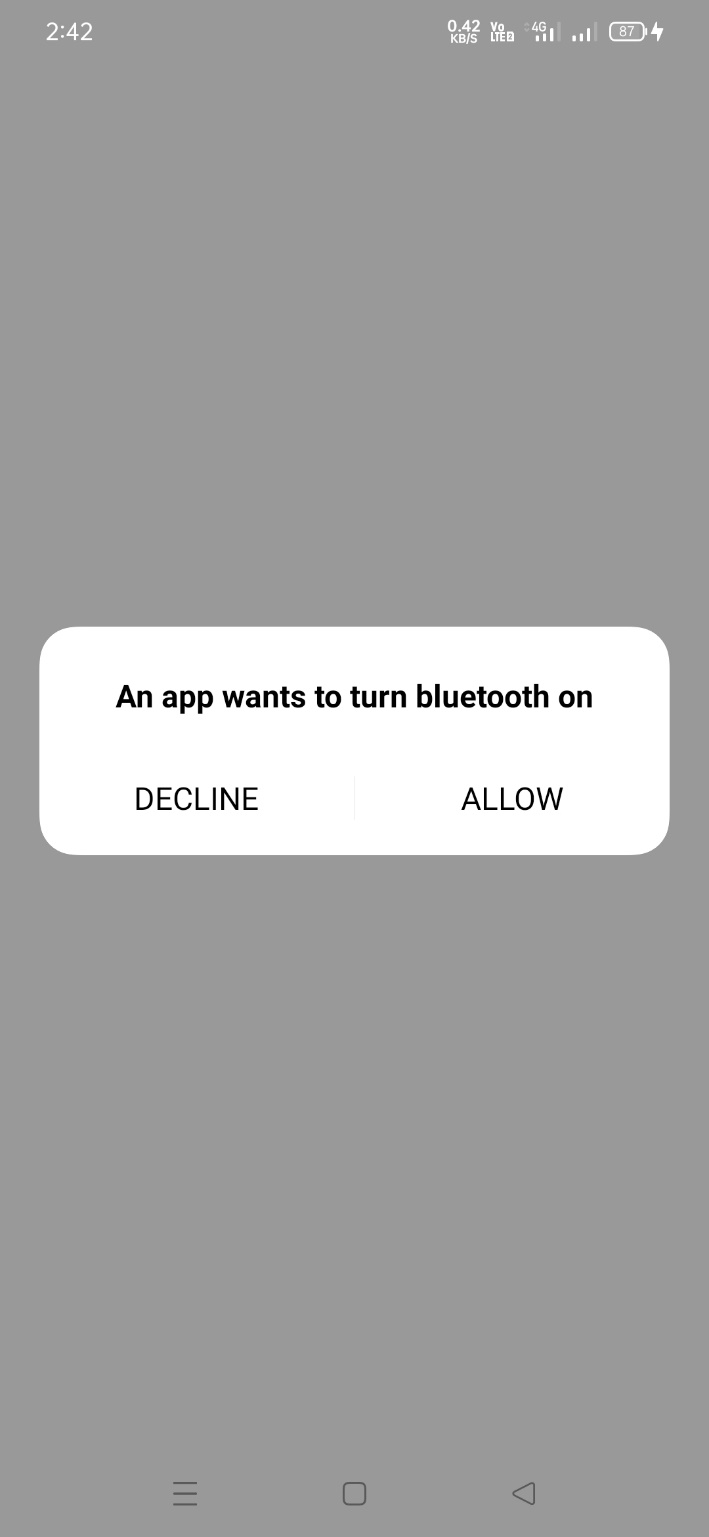
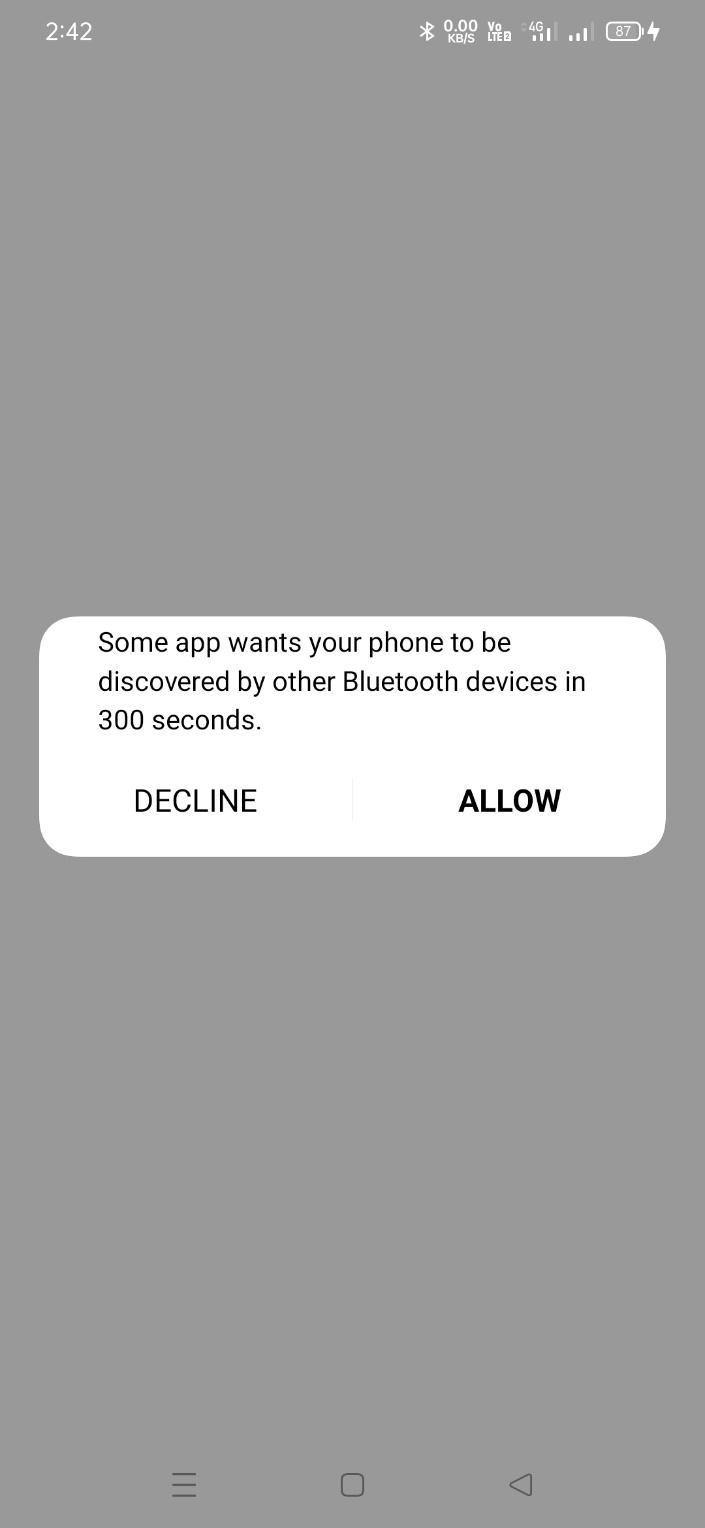


Fig 3 & 4 Take Permission

1. Write a program that prints the signal strength of WiFi connection of the given computer/mobile.

***Wifi\_Signal.java***

**package** com.example.madpractical;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.app.ActivityCompat;  
  
**import** android.Manifest;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.content.pm.PackageManager;  
**import** android.net.wifi.ScanResult;  
**import** android.net.wifi.WifiManager;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** Wifi\_Signal **extends** AppCompatActivity {  
  
 **private** WifiManager **wifiManager**;  
 **private** WifiReceiver **wifiReceiver**;  
 **private** List<ScanResult> **result** = **new** ArrayList<>();  
 ScanResultsAdapter **scanResultsAdapter**;  
 ListView **deviceList**;  
 StringBuilder **sb** = **new** StringBuilder();  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_wifi\_signal***);  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***ACCESS\_WIFI\_STATE***,  
 Manifest.permission.***CHANGE\_WIFI\_STATE***, Manifest.permission.***ACCESS\_FINE\_LOCATION***, Manifest.permission.***ACCESS\_COARSE\_LOCATION***}, PackageManager.***PERMISSION\_GRANTED***);  
  
 **wifiManager** = (WifiManager) getApplicationContext().getSystemService(Context.***WIFI\_SERVICE***);  
 **wifiReceiver** = **new** WifiReceiver();  
 registerReceiver(**wifiReceiver**, **new** IntentFilter(WifiManager.***SCAN\_RESULTS\_AVAILABLE\_ACTION***));  
 Log.*d*(**"MAinActivity"**, **"Starting scan"**);  
 **wifiManager**.startScan();  
  
 Button on\_off = findViewById(R.id.***on\_off\_wifi***);  
 on\_off.setOnClickListener(v -> {  
 **if** (!**wifiManager**.isWifiEnabled()) {  
 **wifiManager**.setWifiEnabled(**true**);  
 Log.*d*(**"Wifi state"**, **"wifi enable"**);  
 } **else** {  
 **wifiManager**.setWifiEnabled(**false**);  
 Log.*d*(**"Wifi state"**, **"wifi disable"**);  
 }  
 });  
  
 Button scan = findViewById(R.id.***enable\_discover\_wifi***);  
 scan.setOnClickListener(v -> {  
 **wifiManager**.setWifiEnabled(**true**);  
 Boolean success = **wifiManager**.startScan();  
  
 **if** (success) Log.*d*(**"startscan"**, **"started"**);  
 **else** Log.*d*(**"startscan"**, **"not started"**);  
  
 Log.*d*(**"Buttton"**, **"Scanning..."**);  
 });  
 }  
  
 **private void** setadapter() {  
 **deviceList** = findViewById(R.id.***device\_list***);  
 **scanResultsAdapter** = **new** ScanResultsAdapter(Wifi\_Signal.**this**, R.layout.***network\_list\_row***, **result**, **wifiManager**);  
 **deviceList**.setAdapter(**scanResultsAdapter**);  
 }  
  
 **protected void** onResume() {  
 **super**.onResume();  
 registerReceiver(**wifiReceiver**, **new** IntentFilter(  
 WifiManager.***SCAN\_RESULTS\_AVAILABLE\_ACTION***));  
 }  
  
 **protected void** onPause() {  
 **super**.onPause();  
 unregisterReceiver(**wifiReceiver**);  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 unregisterReceiver(**wifiReceiver**);  
 }  
  
 **public class** WifiReceiver **extends** BroadcastReceiver {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 **sb** = **new** StringBuilder();  
 **result** = **wifiManager**.getScanResults();  
 Log.*d*(**"WifiReceiver"**, **"Received Something"**);  
 *// prepare text for display and CSV table* **sb**.append(**"Number of APs Detected: "**);  
 **sb**.append((Integer.*valueOf*(**result**.size())).toString());  
 **sb**.append(**"\n\n"**);  
 **for** (**int** i = 0; i < **result**.size(); i++) {  
 *// sb.append((Integer.valueOf(i + 1)).toString() + ".");  
 // SSID* **sb**.append(**"SSID:"**).append((**result**.get(i)).**SSID**);  
 **sb**.append(**"\n"**);  
 *// BSSID* **sb**.append(**"BSSID:"**).append((**result**.get(i)).**BSSID**);  
 **sb**.append(**"\n"**);  
 *// capabilities* **sb**.append(**"Capabilities:"**).append(  
 (**result**.get(0)).**capabilities**);  
 **sb**.append(**"\n"**);  
 *// frequency* **sb**.append(**"Frequency:"**).append((**result**.get(i)).**frequency**);  
 **sb**.append(**"\n"**);  
  
 *// level* **sb**.append(**"Level:"**).append((**result**.get(i)).**level**);  
 **sb**.append(**"\n\n"**);  
 }  
 Log.*d*(**"WifiReceiver"**, **sb**.toString());  
 Log.*d*(**"WifiReceiver"**, **"setting adapter"**);  
 setadapter();  
 *// notify that Wi-Fi scan has finished* }  
 }  
  
  
}

***ScanResultAdapter.java***

**package** com.example.madpractical;  
  
**import** java.util.List;  
  
**import** android.content.Context;  
**import** android.net.wifi.ScanResult;  
**import** android.net.wifi.WifiManager;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.TextView;  
  
**public class** ScanResultsAdapter **extends** ArrayAdapter<ScanResult> {  
  
 **private final** Context **context**;  
 **private final** List<ScanResult> **results**;  
 **private int mViewResourceId**;  
 **private** LayoutInflater **mLayoutInflater**;  
 WifiManager **wifiManager**;  
 **private final** String[] **signalStrength** = {**"Weak"**, **"Fair"**, **"Good"**, **"Excellent"**};  
  
 **public** ScanResultsAdapter(Context context, **int** tvResourceId, List<ScanResult> results, WifiManager wifiManager) {  
 **super**(context, tvResourceId, results);  
 **this**.**context** = context;  
 **this**.**results** = results;  
 **mViewResourceId** = tvResourceId;  
 **mLayoutInflater** = LayoutInflater.*from*(context);  
 **this**.**wifiManager** = wifiManager;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 convertView = **mLayoutInflater**.inflate(**mViewResourceId**, parent, **false**);  
 **if** (**results**.size() > 0) {  
 ScanResult result = getItem(position);  
 *// Get textview fields* TextView txtSSID = convertView.findViewById(R.id.***txtSSID***);  
 TextView txtBSSID = convertView.findViewById(R.id.***txtBSSID***);  
 TextView txtCapabilities = convertView.findViewById(R.id.***txtCapabilities***);  
 TextView txtFrecuency = convertView.findViewById(R.id.***txtFrecuency***);  
 TextView txtLevel = convertView.findViewById(R.id.***txtLevel***);  
  
 **int** level = **wifiManager**.calculateSignalLevel(result.**level**);  
 *// Set fields values* txtSSID.setText(convertView.getContext().getString(R.string.***ssid\_msg***, result.**SSID**));  
 txtBSSID.setText(convertView.getContext().getString(R.string.***bssid\_msg***, result.**BSSID**));  
 txtCapabilities.setText(convertView.getContext().getString(R.string.***capabilities\_msg***, result.**capabilities**));  
 txtFrecuency.setText(convertView.getContext().getString(R.string.***frecuency\_msg***, Integer.*toString*(result.**frequency**)));  
 txtLevel.setText(convertView.getContext().getString(R.string.***signal\_level\_msg***, **signalStrength**[level - 1]));  
 }  
 **return** convertView;  
 }  
  
  
}

***activity\_wifi\_signal.xml***

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/gradient"  
 tools:context=".Wifi\_Signal"**>  
  
 <**ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
  
 <**Button  
 android:id="@+id/on\_off\_wifi"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="ON/OFF"  
 android:textSize="24sp"** />  
  
 <**Button  
 android:id="@+id/enable\_discover\_wifi"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="Enable discover"  
 android:textSize="24sp"** />  
  
 <**ListView  
 android:id="@+id/device\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="400dp"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginTop="20dp"  
 android:foregroundGravity="center\_horizontal"** >  
  
  
 </**ListView**>  
  
 </**LinearLayout**>  
 </**ScrollView**>  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

***network\_list\_row.xml***

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/network"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="left|center"  
 android:orientation="vertical"  
 android:paddingLeft="5dp"  
 android:paddingTop="5dp"  
 android:paddingBottom="5dp"**>  
  
 <**TextView  
 android:id="@+id/txtSSID"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/ssid\_msg"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
  
 <**TextView  
 android:id="@+id/txtBSSID"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
  
 <**TextView  
 android:id="@+id/txtCapabilities"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
  
 <**TextView  
 android:id="@+id/txtFrecuency"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
  
 <**TextView  
 android:id="@+id/txtLevel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/white"  
 android:textSize="20sp"** />  
</**LinearLayout**>

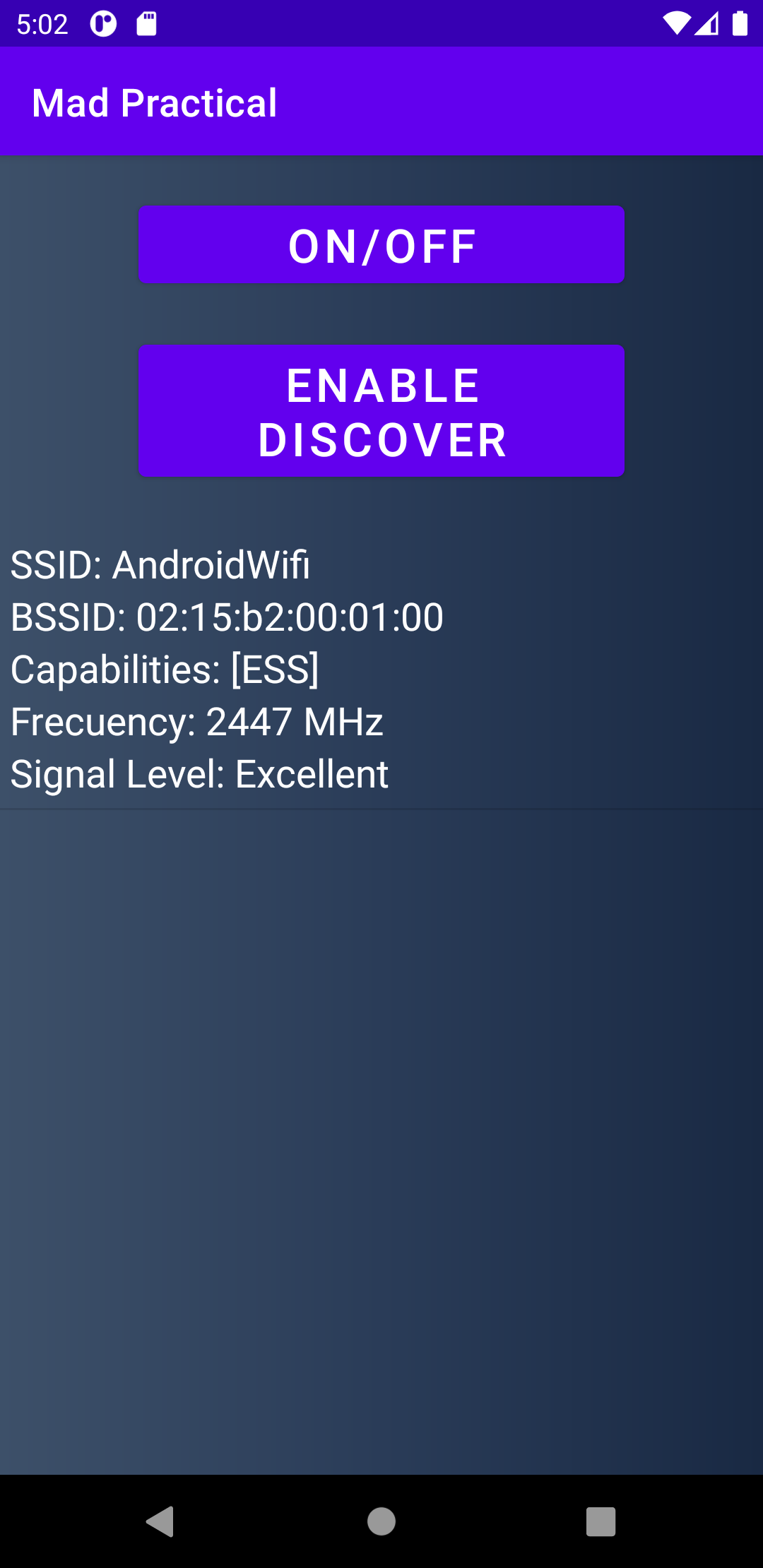


Fig 5 – Wi-Fi Scanner shows Signal strength with extra details