DISTANCE VIOLATION RECOGNITION (BLUETOOTH) ANDROID APPLICATION (COVID-19)

Hochschule Schmalkalden
Schmalkalden, Thüringen
MSc Informatik SoSe 2020/21
Prof. Dr.-Ing. Dr.phil. Michael Cebulla

Madhan Raj Moorthy
309937
Hochschule Schmalkalden
Schmalkalden, Thüringen
MSc Informatik SoSe 2020/21

30.09.2020

Table of Contents:

1. Abstract

Basically, android architecture is based on Linux and the applications gives major key points where several complex tasks are made easier using Android oriented applications. It supports to deal with heterogenous hardware components as there are several android devices available globally. Android deals with all the aspects of the hardware components as per the devices which are communicated using sensors majorly. It handles most of the resources as per the device's hardware components. Bluetooth is important component used to send or receive data from one device to other devices wirelessly with its framework.

Keywords
Bluetooth, Distance Violations, Logging, RSSI

2. Introduction

The goal of this project is to create an android application (APK) to find the nearby devices using Bluetooth to know the violation of the distance where we can measure the distance between using the signal strength. There are few ways to find this distance using some features in Android devices. This application uses Bluetooth to fulfill the requirements. Using this feature, the application is developed to detect the nearby devices using Bluetooth. It can detect the devices whichever is nearby as per the signal strength (RSSI) of other devices.

3. Bluetooth

Bluetooth is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the industrial, scientific and medical radio bands, from 2.402 GHz to 2.480 GHz, and building personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables.

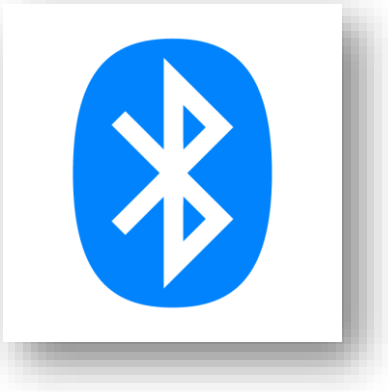

Fig 1: Bluetooth

3.1.Communication of Bluetooth:

For Bluetooth-enabled devices to transmit data between each other, they must first form a channel of communication using a pairing process. One device, a discoverable device, makes itself available for incoming connection requests. Another device finds the discoverable device using a service discovery process. A Bluetooth® device works by using radio waves instead of wires or cables to connect with your cell phone, smartphone or computer. Bluetooth is a wireless short-range communications technology standard.

Android provides Bluetooth API to perform these different operations.

- Scan for other Bluetooth devices
- Get a list of paired devices
- Connect to other devices through service discovery

Android provides **BluetoothAdapter** class to communicate with Bluetooth.

3.2.Uses

The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs. The Bluetooth has some key benefits or uses for communicating as below,

- Hands-Free
- File Transfers

- Internet Tethering
- Pairing devices to communicate

The uses also extend to the following,

* Robustness - It uses a fast acknowledgement and frequency hopping scheme to make a radio link robust.

* Low complexity- The necessary transceiver components present in the devices are simple.

* Low cost- A wireless device with this technology is available at an affordable price.

4. Architecture/Function

A Bluetooth application communicates with the Bluetooth process through Binder. The Bluetooth process uses JNI to communicate with the Bluetooth stack and provides developers with access to various Bluetooth profiles. This diagram shows the general structure of the Bluetooth stack:

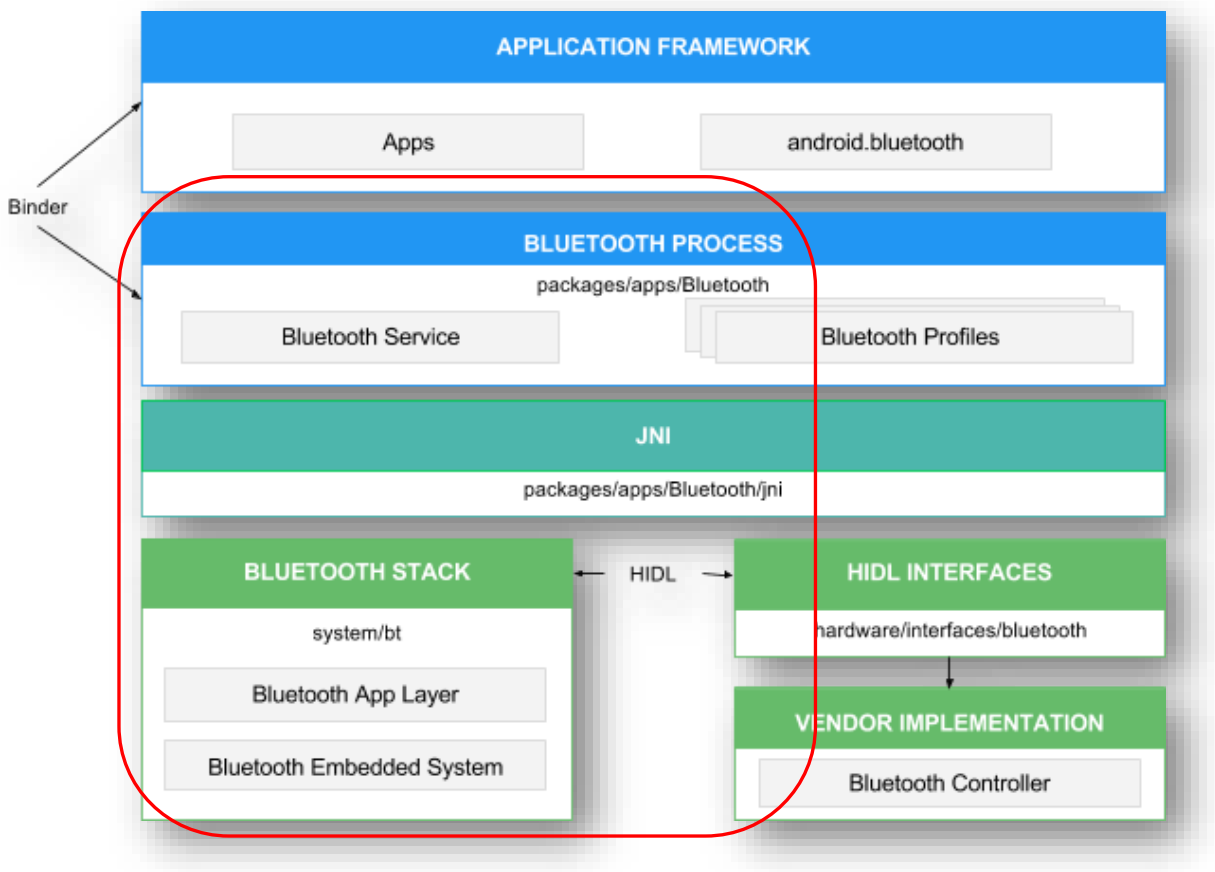

Fig 2: Bluetooth Architecture

Based on this project, firstly the Bluetooth should scan the nearby devices. So that when the Bluetooth or Bluetooth scanning is enabled, a report containing a list of nearby Bluetooth beacons.

4.1.RSSI

RSSI is the signal strength at the Bluetooth receiver. The **Received Signal Strength Indicator** (RSSI) is a measure of the power level at the receiver. When a device scans for Bluetooth devices, the Bluetooth radio inside the device provides a measurement of the RSSI for each seen device. It's measured in decibels, dBm, on a logarithmic scale and is negative. A more negative number indicates the device is further away. For example, a value of -20 to -30 dBm indicates the device is close while a value of -120 indicates the device is near the limit of detection.

As per RSSI values can be used to infer the distance. The accuracy of the distance can be dependent on many factors like,

- Type of sending device
- Output power
- Capability of receiving device
- Acceptance of receiving devices

4.2.Signal Strength

For Bluetooth, it is the strength of the beacon's signal as seen on the receiving device, **e.g**. a smartphone. **The signal strength depends on distance and Broadcasting Power value**. Bluetooth works with broadcasting signals and that broadcasting power value is around 2–4 dBm — and due to that, the signal RSSI strength will be around -26 (a few inches) to -100 (40–50 m distance). The distance calculated will be approximate as per the signal strength.

Formula

- Distance
- Measured Power
- RSSI
- N (Constant depends on the Environmental factor. Range 2–4, low to-high strength as explained above)

Distance = 10 ^ ((Measured Power — RSSI)/(10 * N))

Values Example:
Distance for RSSI -75 = 10 ^ ((-69 - (-75))/(10 * 2)) = 1.995 meters ≅ 2 meters

Distance for RSSI -69 = 10 ^ ((-69 - (-69))/(10 * 2)) = 1 meter

5. User Interface

Manual prototype is drawn for the UI as per the functionalities which are planned to add it in the project. **Layout Used**: Linear Layout
There are 2 activities in the application. First activity contains the major function of the application.

Fig 3: **Main Activity**

Activity 1 is named in the button as (FIND DEVICES) is for finding the nearby devices as per the Bluetooth signal strength. Activity 2 named in the button as (INFO) shows awareness information related to COVID-19. The screenshots of the UI is given below,

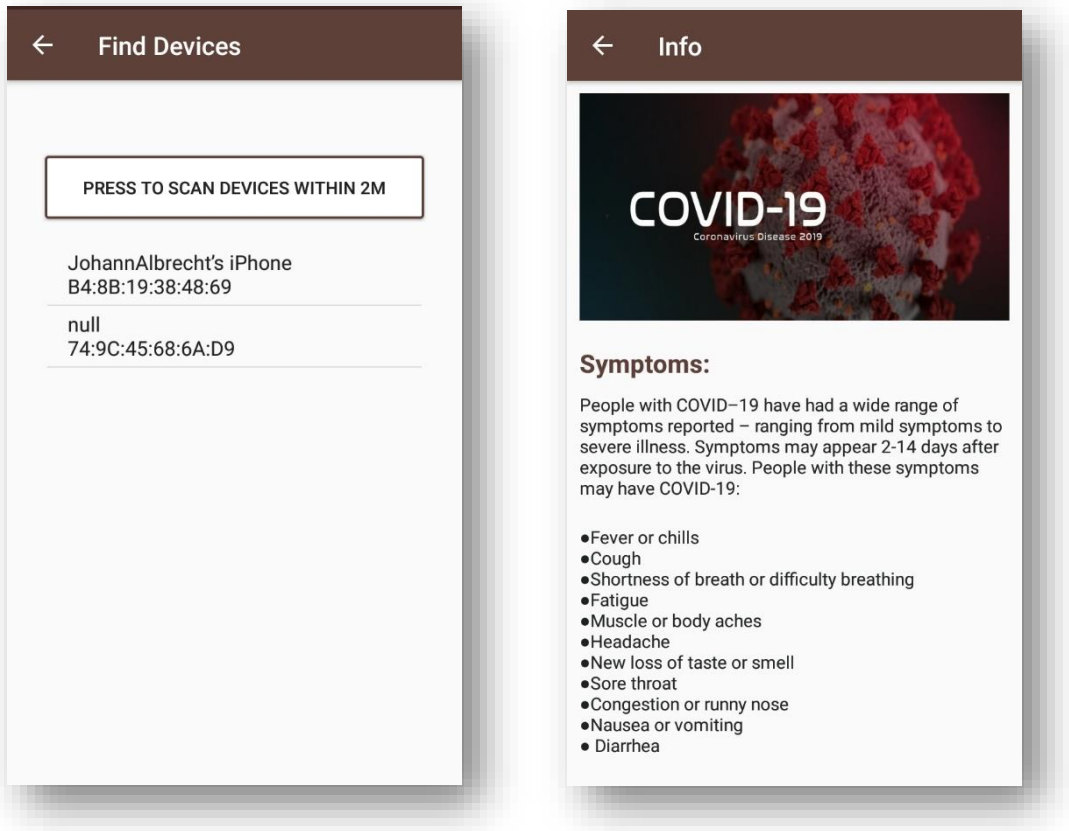

Fig 4: **Activity 1 & 2**

6. Implementation:

The application has been implemented to work based on scanning and detecting nearby Bluetooth devices and log it into the application. Where, it is added to receive the deviceName, deviceAddress, and also to show the value.

Permissions used:

- BLUETOOTH
- BLUETOOTH_ADMIN
- ACCESS_COARSE_LOCATION

Bluetooth permissions are important to make it usable for the feature of this application. At first time, application request for the device location as a permission to the user. Then,

the user can decide it whether to allow or deny the location to make the application uses it for detecting the nearby devices using signal strength as per the defined RSSI value.

Functions:

This application used Bluetooth adapter, RSSI, BroadcastReceiver functions for declaring the Bluetooth functions.

- BluetoothAdapter
  - Used to initiate device discovery, instantiate a BluetoothDevice using a known MAC address, and create a BluetoothServerSocket to listen for connection requests from other devices, and start a scan for Bluetooth.

```
public void scanNewDevices(View view) {
    BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    boolean isEnabled = bluetoothAdapter.isEnabled();
    if (!isEnabled) {
        setBluetooth(true);
    } else {
        Toast.makeText( context: this,  text: "Scanning nearby devices...", Toast.LENGTH_SHORT).show();
        //Scan for unpaired available bluetooth devices when "scan button" is clicked
        if (!isDiscoverable) {
          discoverOn();//Set bluetooth to discoverable
            isDiscoverable = true;
        }

        allDevices.clear();
        if (btAdapter.isDiscovering()) {
            //Bluetooth is already in mode discovery mode, we cancel to restart it again
            btAdapter.cancelDiscovery();
        }
        btAdapter.startDiscovery();                                    //Start scanning

        IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        isReceiverRegistered = true;
        registerReceiver(receiver, filter);
    }
}
```

Fig 5: Bluetooth Adapter

- BroadcastReceiver
  - It is an inactive component of android that listens to system-wide broadcast events or intents. When any of these events occur, it brings the application into action by either creating a status bar notification or performing a task.

- RSSI
    - It is the measurement of how well your device can receive the signal from other devices using its values as range.

```
//Create a BroadcastReceiver for ACTION_FOUND.
//RSSI
private final BroadcastReceiver receiver = new BroadcastReceiver() {
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            //Discovery has found a device. Get the BluetoothDevice
            //object and its info from the Intent.
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);

            if (newDevices != null) {
                newDevices.clear();                                 //Clear earlier list
            }

            if (device != null) {
                int rssi = intent.getShortExtra(BluetoothDevice.EXTRA_RSSI, Short.MIN_VALUE);
                if (rssi > -62 && rssi < -40) {
                    newDevices.add( device.getName() + '\n' + device.getAddress() + '\n' + (" RSSI Value: " +rssi));
                }
                allDevices.addAll(newDevices);                      //Update new devices in final display list

                showList();                                         //Display list
            }

        }
    }
};
```

Fig 6: RSSI

RSSI:

- It is based on the signal strength where we can define as the range. As per the range the distance can be approximately assigned.

Values Example:
Distance for RSSI -75 = 10 ^ ((-69 - (-75))/ (10 * 2)) = 1.995 meters ≅ 2 meters
Distance for RSSI -69 = 10 ^ ((-69 - (-69))/ (10 * 2)) = 1 meter

7. Conclusion

Overall, as per the scenario the application works for finding nearby devices which is fetched based on the signal strength of Bluetooth (RSSI). The values can be modified as per the required distances. The range is defined in the Receiver function with the RSSI values then the information fetched are devices name, device-Address, RSSI value. All the nearby devices those are on the range as defined will be scanned and shown in the list view of the activity. In the list view shows the name, addresses of the detecting devices and the range

value to know as an information. RSSI helps to detect the approximate distances of the nearby devices. The range can be set as per the requirement. Lastly, the RSSI function used in this application would be useful to the end users to know the nearby devices and the users will be notified.

References

1) Development of an Android APK for Bluetooth localization, 2011/2012, https://upcommons.upc.edu/bitstream/handle/2099.1/17357/memoria.pdf?sequence=4&isAllowed=y

2) Distance Estimation and Positioning Based on Bluetooth Low Energy Technology, Kth Royal Institute of Technology, 2015, https://www.diva-portal.org/smash/get/diva2:859549/FULLTEXT01.pdf

3) Distance Measurements and Classification Test Report, Dipl.-Inf. Steffen Meyer, Fraunhofer IIS

4) https://en.wikipedia.org/wiki/Bluetooth

5) https://www.bluetooth.com/blog/proximity-and-rssi/

6) https://developer.android.com/guide/topics/connectivity/bluetooth

7) http://www.swiftutors.com/bluetooth-introduction.html

8) https://bluetoothle.wiki/rssi