Real-Time Case Study Example: Bluetooth, Wi-Fi, and NFC in Android Java

Here's a case study where we integrate **Bluetooth**, **Wi-Fi**, and **NFC** functionalities in a simple Android app. We will create an app that:

1. Scans for Bluetooth devices, Employees can share files between devices using Bluetooth
2. Connects to Wi-Fi. Automatically connects if Wifi available
3. Reads NFC tags. Employees can use this NFC enabled ID cards to check their office attendance

---

1. Setting Up Permissions

In your AndroidManifest.xml, declare the required permissions:

xml

Copy code

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.NFC" />

<uses-feature android:name="android.hardware.nfc" android:required="true" />
<uses-feature android:name="android.hardware.bluetooth" android:required="true" />
<uses-feature android:name="android.hardware.wifi" android:required="true" />
```

2. Layout File (activity_main.xml)

In your res/layout/activity_main.xml, create buttons for the functionalities and a TextView to display results.

xml

Copy code

```
<?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"
    android:padding="16dp">

    <Button
        android:id="@+id/btnBluetooth"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Scan Bluetooth Devices" />

    <Button
        android:id="@+id/btnWifi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Connect to Wi-Fi" />

    <Button
        android:id="@+id/btnNfc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Read NFC Tag" />

    <TextView
        android:id="@+id/txtResults"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Results will appear here"
        android:paddingTop="20dp" />
</LinearLayout>
```

3. MainActivity Java Code (MainActivity.java)

In your MainActivity.java, handle Bluetooth, Wi-Fi, and NFC operations:

java

Copy code

```
package com.example.bluetoothwifinfc;


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.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.nfc.NfcAdapter;
import android.nfc.NfcEvent;
import android.nfc.NfcManager;
import android.nfc.Tag;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.Set;

public class MainActivity extends AppCompatActivity {

    private BluetoothAdapter bluetoothAdapter;
    private WifiManager wifiManager;
    private NfcAdapter nfcAdapter;
    private TextView resultsTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize views
        Button bluetoothButton = findViewById(R.id.btnBluetooth);
        Button wifiButton = findViewById(R.id.btnWifi);
        Button nfcButton = findViewById(R.id.btnNfc);
        resultsTextView = findViewById(R.id.txtResults);
        // Initialize Bluetooth adapter
        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

        // Initialize Wi-Fi manager
        wifiManager = (WifiManager) getSystemService(WIFI_SERVICE);

        // Initialize NFC adapter
        nfcAdapter = NfcAdapter.getDefaultAdapter(this);

        // Bluetooth scanning functionality
        bluetoothButton.setOnClickListener(v -> scanBluetoothDevices());

        // Wi-Fi connection functionality
        wifiButton.setOnClickListener(v -> connectToWifi());

        // NFC reading functionality
        nfcButton.setOnClickListener(v -> readNfcTag());
    }

    private void scanBluetoothDevices() {
        if (bluetoothAdapter == null) {
            resultsTextView.setText("Bluetooth is not supported on this device.");
            return;
        }
        if (!bluetoothAdapter.isEnabled()) {
            resultsTextView.setText("Bluetooth is off. Please turn it on.");
            return;
        }

        Set<BluetoothDevice> pairedDevices = bluetoothAdapter.getBondedDevices();
        StringBuilder deviceList = new StringBuilder("Paired Bluetooth devices:\n");
        for (BluetoothDevice device : pairedDevices) {
            deviceList.append(device.getName()).append("\n");
        }
        resultsTextView.setText(deviceList.toString());

        // Discover new devices
        IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        registerReceiver(bluetoothReceiver, filter);
        bluetoothAdapter.startDiscovery();
    }

    private void connectToWifi() {
        WifiInfo currentWifi = wifiManager.getConnectionInfo();
        String currentNetwork = currentWifi.getSSID();
        resultsTextView.setText("Connected to Wi-Fi: " + currentNetwork);
    }

    private void readNfcTag() {
        if (nfcAdapter == null || !nfcAdapter.isEnabled()) {
            resultsTextView.setText("NFC is not available or turned off.");
            return;
        }
        resultsTextView.setText("Please scan an NFC tag.");
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (nfcAdapter != null) {
            nfcAdapter.enableForegroundDispatch(this, PendingIntent.getActivity(this, 0, new Intent(this, getClass()), 0),
                    null, null);
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
        if (nfcAdapter != null) {
            nfcAdapter.disableForegroundDispatch(this);
        }
    }

    // Bluetooth receiver to handle found devices
    private final BroadcastReceiver bluetoothReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();
            if (BluetoothDevice.ACTION_FOUND.equals(action)) {
                BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                String deviceName = device.getName();
                String deviceAddress = device.getAddress(); // MAC address
                resultsTextView.append("Found device: " + deviceName + " (" + deviceAddress + ")\n");
            }
        }
    };

    // Handle NFC tags when scanned
    @Override
    public void onNewIntent(Intent intent) {
        super.onNewIntent(intent);
        if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent.getAction())) {
            Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
            String tagId = bytesToHex(tag.getId());
            resultsTextView.setText("NFC Tag Scanned: " + tagId);
        }
    }

    // Convert NFC tag byte array to Hex
    private String bytesToHex(byte[] bytes) {
        StringBuilder hexString = new StringBuilder();
        for (byte b : bytes) {
            hexString.append(String.format("%02X", b));
        }
        return hexString.toString();
    }
}
```

Key Points

1. **Bluetooth:**
   - We check if Bluetooth is supported and enabled on the device.
   - We list paired Bluetooth devices and can scan for new devices.
2. **Wi-Fi:**
   - We use WifiManager to check the current connected Wi-Fi network.
3. **NFC:**
   - We use NfcAdapter to detect NFC tags and display the tag's ID.

Testing the App:

1. Ensure Bluetooth, NFC, and Wi-Fi are enabled on the device.
2. Use the buttons in the app to:
   - Scan for Bluetooth devices.
   - Display the current Wi-Fi connection.
   - Read an NFC tag when scanned.

Conclusion:

This example shows a real-time case study where Bluetooth, Wi-Fi, and NFC functionalities are integrated into a single Android app, enabling basic operations such as scanning Bluetooth devices, connecting to Wi-Fi, and reading NFC tags.