Mobile Application Development
Report On
Bluetooth And WiFi

1) Rationale:-

- This project is an android application for Bluetooth and WiFi.
- In this application, a switch button is used for changing the Bluetooth and WiFi state (i.e. ON or OFF).

2) Aims of the Project:-

- To create a Mobile Application using Android Studio for turning ON and OFF the Bluetooth and WiFi of mobile.

Benefits of project are:-

- The project can be used in different applications for which Bluetooth and WiFi switching (ON and OFF) is required.

3) Course Outcomes Addressed:-

- Interpret features of Android operating system.
- Configure Android environment and development tools.
- Develop rich user interfaces by using layouts and controls.
- Use User Interface components for android application development.

4) Literature Review:-

- Referred following website:-
- https://android.jlelse.eu/connect-android-device-with-wifi-within-android-studio-3b1bc00c1e17?gi=e56da83e7ba
- https://www.tutlane.com/tutorial/android/android-bluetooth-turn-on-or-off-with-examples

5) Actual Methodology Followed:-

a)Finalization of Micro-Project topic.
b)Preparation of proposal.
c)Collecting information on:-

1.WiFi Service:

Android provides WiFi API to perform these different operations:

Scan for other WiFi networks.

Get a list of WiFi networks.

- Connect to other network through service discovery.

Android provides WiFi Manager class to communicate with WiFi. Create an object of this calling for switching state of WiFi as given below syntax:

```
WifiManager Wifion =
    (WifiManager)getApplicationContext().getSystemService(Context.WIF
    I_SERVICE);
```

In order to enable the WiFi of your device, set the if condition with the following WiFi constant .getSystemService(Context.WIFI_SERVICE). Its syntax for turning service on and off is:

```
 if(wifi.isChecked())
      {
     WifiManager wifion =
    (WifiManager)getApplicationContext().getSystemService(Context.WIF
    I_SERVICE);
     wifion.setWifiEnabled(true);
        }
     else
      {
     WifiManager wifioff =
    (WifiManager)getApplicationContext().getSystemService(Context.WIF
    I_SERVICE);
         wifioff.setWifiEnabled(false);
                    }
```

2.Bluetooth Service:

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. Create an object of this calling by calling the static method getDefaultAdapter(). Its syntax is given below.

```
privateBluetoothAdapter BA;
BA =BluetoothAdapter.getDefaultAdapter();
```

In order to enable the Bluetooth of your device, call the intent with the following Bluetooth constant ACTION_REQUEST_ENABLE. Its syntax for turning service on is.

```
Intent turnOn
    =newIntent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
 startActivityForResult(turnOn,0);
```

3. Switch:

It is a two-state toggle switch widget that can select between two options. The user may drag the switch or simply just click on it. It is a subclass of CompoundButton. It is basically used for turning off/on button which decides the current state of Switch. In this project, it is use for turning Bluetooth and WiFi On and Off.

d)Designing interface for the application:-

 activity_main.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:orientation="vertical"
 tools:context=".MainActivity">

<Switch
        android:id="@+id/bluetooth"
        android:layout_width="208dp"
        android:layout_height="45dp"
        android:layout_marginStart="92dp"
        android:layout_marginLeft="92dp"
       android:layout_marginBottom="459dp"
        android:text="Bluetooth"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent" />
```

```
<Switch
        android:id="@+id/wifi"
        android:layout_width="208dp"
        android:layout_height="45dp"
        android:layout_marginStart="92dp"
        android:layout_marginLeft="92dp"
        android:layout_marginTop="80dp"
        android:text="Wifi"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/bluetooth" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

e)Implementing code required for Bluetooth and WiFi switching:-

 MainActivity.java

```
package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.bluetooth.BluetoothAdapter;
import android.content.Context;
import android.content.Intent;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Switch;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {


    Switch bluetooth,wifi;
    privateBluetoothAdapter BA;

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

```
        bluetooth=(Switch)findViewById(R.id.bluetooth);
        wifi=(Switch)findViewById(R.id.wifi);
        bluetooth.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
               BA = BluetoothAdapter.getDefaultAdapter();
                if (bluetooth.isChecked()) {
                    Intent turon = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                    startActivity(turon);
                    Toast.makeText(getApplicationContext(), "Turning Bluetooth
ON", Toast.LENGTH_SHORT).show();
                 }
              else
               {
                    BA.disable();
                    Toast.makeText(getApplicationContext(), "Bluetooth OFF",
Toast.LENGTH_SHORT).show();
                }
            }
        });
        wifi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(wifi.isChecked())
                {
                    WifiManager wifion =
(WifiManager)getApplicationContext().getSystemService(Context.WIFI_S
ERVICE);
                    wifion.setWifiEnabled(true);
                }
                e  l  s  e
                {
                    WifiManager wifioff =
(WifiManager)getApplicationContext().getSystemService(Context.WIFI_S
ERVICE);
                    wifioff.setWifiEnabled(false);
                    Toast.makeText(getApplicationContext(), "Wifi OFF",
Toast.LENGTH_SHORT).show();
```

```
            }
        }
    });
  }
}
```

f)Setting up permission in manifest for Bluetooth and WiFi:-

 AndroidManifest.xml

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

g)Testing the application and preparing report:-

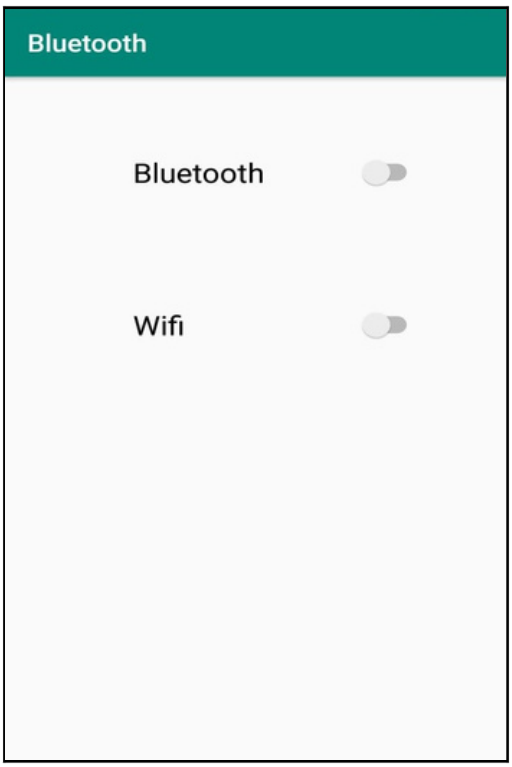
6) Actual Resources Used:-

Sr. No.	Name of Resource/material	Specifications	Qty
1)	Microsoft Word.	Any Version.	1
2)	Android Studio	Arduino 3.5	1
3)	Mobile Phone.	Any Android Version above 5.	1

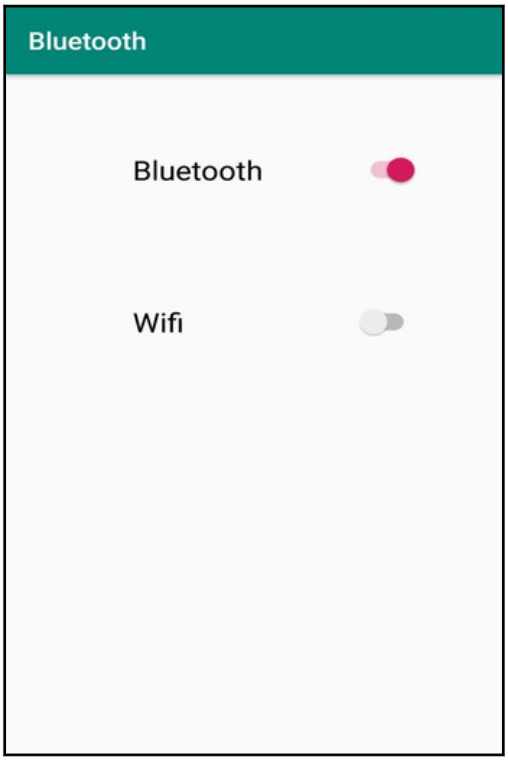
7) Outputs of the Micro-Projects:-

- Interface designs of an application:

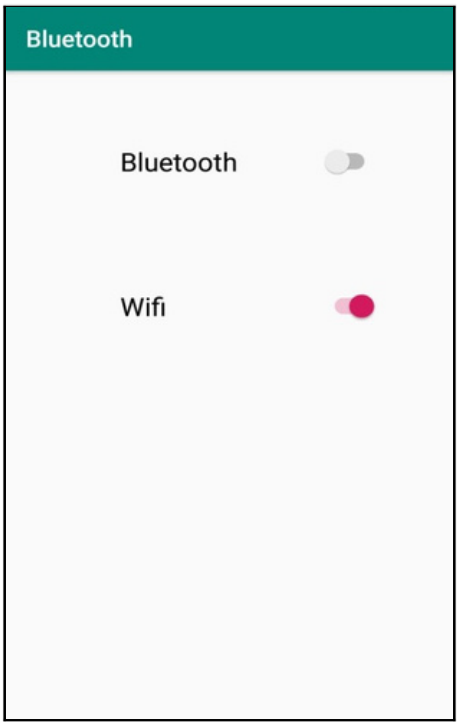
First Screen



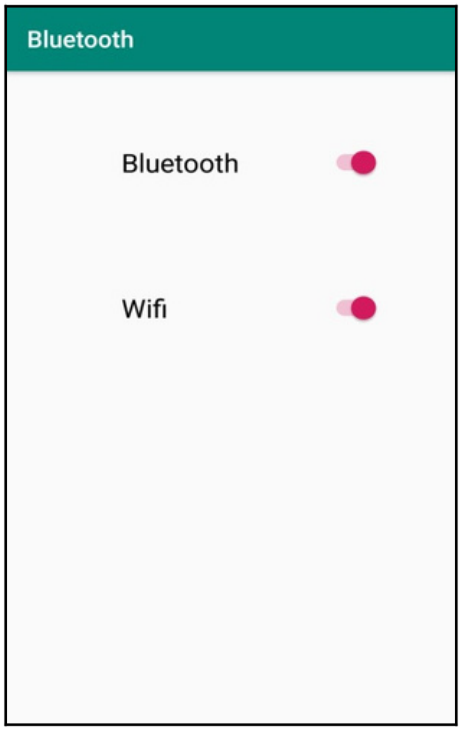
Second Screen



Third Screen



Fourth Screen



8) Skill Developed / Learning outcome of this Project:-

- To develop an application using Android Studio.

9) Applications of this Project:-

- It can be used in application which requires both Bluetooth and WiFi Connectivity.

****