Android based Wireless Access Point Tracker (WAP Tracker)

MINT 709: CAPSTONE PROJECT REPORT

Ishant Dhanani
Supervisor: Mr. Leonard Rogers

Acknowledgment

I would like to express my gratitude to Mr. Leonard Rogers for being my project supervisor. His constant guidance and willingness to share his vast knowledge has helped me in enhancing my knowledge and accomplishment of assigned tasks to perfection. Without his effort and full support, this project may not have succeeded.

I would also like to thank Dr. Michael Macgregor and Mr. Shahnawaz Mir for allowing me to carry out this project as a part of CAPSTONE Project.

Ishant Dhanani

Table of Contents

Table of Figures

Brief Description

WAP Tracker will be useful to network analysts to manage, track and identify Wireless Access Points within a specific area. The basic features are:

- Identify all Wireless Access Point signals within the measuring devices range and list them on screen with measured signal strength
- All identified WAPs will display their SSID.
- All unidentified WAPs (not broadcasting its SSID) will be listed as unknown.
- If a WAP is selected, the screen switches to display information about the specific WAP only and continuously scan for the selected WAP to indicate the moving close or further away from a selected Access Point

Project Details

This project has come about due to a lack of effective, low-cost handheld tools that can be used to track and manage Wireless Access Points in the field. Most importantly, this includes rogue Access Points which are often physically hidden and not broadcasting their SSIDs, representing a serious security issue to corporate networks.

The issue that started this project in the first place was the need to track down and eliminate just such a physically hidden rogue Access Point. After much searching, it was determined that there are very few of these applications available to use, with the majority of them being laptop or tablet-based, only two (at the time) being android-based.

Of course, of the applications that did track Wireless Access Points none of them identified access points that were not broadcasting their SSIDs (unknown APs). The end result was 3 days of wandering around with a wireless packet analyser to capture all wireless traffic in an area, then filtering out all known traffic until only the unknown traffic was left and then trying to identify where the signal was coming from by determining if the traffic for a device was being picked up in a measured area.

With the growth of Android-based smartphones and the increased processing capabilities of these devices, it only makes sense that these devices are the perfect solution for development and implementation of such an android application.

The formula to calculate signal level received for a signal of Wavelength λ (nm) after traveling distance d (km) is expressed in Free Space Path Loss (FSPL) as below [1]:

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2$$

$$= \left(\frac{4\pi df}{c}\right)^2$$

$$= 20\,log_{10}(d) + 20\,log_{10}(f) + 20\,log_{10}\left(\frac{4\pi}{c}\right)$$

$$= 20\,log_{10}(d) + 20\,log_{10}(f) - 147.55$$

$$= 20\,log_{10}(d) + 20\,log_{10}(f) - 27.55$$

So the final formula to calculate distance is as below:

$$d = 10^{\left(\frac{FSPL\ (dB) + 27.55 - 20\,log_{10}(f)}{20}\right)}$$

Where:
f – Signal Frequency (Mega Hertz MHz)
d – Distance from the transmitter (meters m)
c – Speed of light in Vacuum (3 x 10^8 m/s)

There could be several obstacles in the path of radio waves between transmitter and receiver. There are three different types of waves that could be received at receiver:

1. Direct waves: Waves received directly by receiver
2. Reflected waves: Waves received by receiver after hitting the obstacle
3. Diffracted waves: Waves received by the receiver which passes from one medium to another.

Even if the signal from Access Point is not directly received by Smartphone, the signal could be received momentarily due to reflected and diffracted waves. At receiver, it might happen that direct, reflected and diffracted signals are combined, as a result of which the received signal level becomes variously stronger or weaker. So the distance measured as per above formula is an approximate value. [2]

Application Details

Application consists of below class files:

- MainActivity
- ListAccessPoints
- AccessPointDetails
- DatabaseHelper
- SettingsActivity
- AboutActivity

MainActivity [3] [4] [5] [6] [7] [8]

This class is responsible for scanning all the WAP including those not broadcasting the SSID and list them on Screen. It shows below details of Access Points:

- SSID – Name of WAP (if broadcasting SSID) else it would show Unknown AP# <Number in list>
- BSSID – MAC Address of Wireless Radio Interface
- Signal Strength

The list is refreshed periodically based on the preference set in Settings for **List Scan Interval**, by default the value is set to 10 Seconds. Access Points are displayed in a selectable list using ListAccessPoints class.

ListAccessPoints [9]

This class is responsible for fetching SSID, BSSID and Signal Strength and populate data in the selectable list using Wi-Fi scan list passed by MainActivity.

AccessPointDetails [3] [4] [5] [7] [8]

This class is responsible for listing the details of WAP selected in List on the main screen. It shows below details of Access Points:

- SSID – Name of WAP (if broadcasting SSID) else it would show Unknown AP# <Number in list>
- BSSID – MAC Address of Wireless Radio Interface
- Signal Strength
- Distance To
- Vendor

The list is refreshed periodically based on the preference set in Settings for **Details Scan Interval**, by default the value is set to 5.0 Seconds. In a scenario where there are a large number of access points, if the interval is too small compared to the process of going through all access points, it might happen next periodic scan is due. So before the next periodic scan is started, completeFlag is checked to see if the previous scan was completed or not. If not, the scan is skipped until the previous scan completed. In this way, it doesn't hog the CPU and produce no result. The distance increases if the mobile device is moving away from the access point whereas distance decreases if it is moving closer to the access point. If distance doesn't change, it might be possible that the mobile device is moving along the radius.

DatabaseHelper [10] [11] [12]

This class is responsible for setting up a database for Vendor Data. It copies the database from asset folder of android to local app directory. It is also responsible for finding the company name based on the OUI derived from BSSID.

SettingsActivity [14] [15]

This class is responsible for showing and setting up values for preferences: **List Scan Interval** and **Details Scan Interval.** The values of these preferences are listed in form of Radio List. Below are the minimum and maximum values for preferences:

List Scan Interval:
Minimum Value: 2.5 Seconds
Maximum Value: 9.0 Seconds

Details Scan Interval
Minimum Value: 5 Seconds
Maximum Value: 30 Seconds

AboutActivity

This class is responsible for showing information of the application.

Future Scope

This application is developed using API exposed for Android OS which doesn't provide direct access of wireless network interface card. When direct access to hardware is available, packets could be sent and received from access point to extract password information, IP Address and Ethernet MAC Address of rouge access point. This allows to login into access point which could be trying to get access to a network for which user is not authorized to. The decision of using API for application development is due to lack of driver availability of wireless NIC for android devices. For future development, it might be possible to develop a generic driver which allows access to wireless NIC or if API itself allows access to hardware, above things could be achieved in future.

References

[1] https://en.wikipedia.org/wiki/Free-space_path_loss
[2] http://www.cdt21.com/resources/guide2.asp
[3] https://developer.android.com/reference/android/net/wifi/WifiManager.html
[4] https://developer.android.com/reference/android/net/wifi/ScanResult.html
[5] https://developer.android.com/reference/android/content/BroadcastReceiver.html
[6] https://developer.android.com/reference/android/content/SharedPreferences.html
[7] https://developer.android.com/reference/java/util/Timer.html
[8] https://developer.android.com/reference/java/util/TimerTask.html
[9] http://theopentutorials.com/tutorials/android/listview/android-custom-listview-with-image-and-text-using-baseadapter
[10] https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html
[11] https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html
[12] https://developer.android.com/reference/android/provider/BaseColumns.html
[13] https://developer.android.com/reference/android/preference/PreferenceActivity.html
[14] https://developer.android.com/reference/android/preference/PreferenceFragment.html

Appendix – A

MainActivity.java

```
package com.waptracker;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.os.Handler;
import android.preference.PreferenceManager;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.Toast;
import com.waptracker.about.AboutActivity;
import com.waptracker.settings.SettingsActivity;
import com.waptracker.vendor.DatabaseHelper;
import org.apache.commons.lang3.StringUtils;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    private final static String listRefresh = "main_scan_interval";
    long listRefreshRate;
    WifiManager wifiManager;
    WifiReceiver wifiReceiver;
    ListView wifiListView;
    List<ScanResult> wifiList;
    ListAccessPoints adapter;
    Context context;
    DatabaseHelper databaseHelper;
    private Timer timer;
    private TimerTask timerTask;
    private Handler handler = new Handler();
    private ProgressBar spinner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        PreferenceManager.setDefaultValues(this, R.xml.preferences, false);
        setContentView(R.layout.activity_main);

        setRefreshRate();

        databaseHelper = new DatabaseHelper(this);
        File database =
```

```
getApplicationContext().getDatabasePath(DatabaseHelper.DATABASE_NAME);
        if (false == database.exists()) {
            databaseHelper.getReadableDatabase();
            if (copyDatabase(this)) {
                Toast.makeText(this, "Copy database success",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Copy database failed",
Toast.LENGTH_SHORT).show();
                return;
            }
        }

        wifiListView = (ListView) findViewById(R.id.lvWifiDetails);
        wifiManager = (WifiManager) getSystemService(context.WIFI_SERVICE);
        wifiReceiver = new WifiReceiver();
        registerReceiver(wifiReceiver, new
IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));

        if (wifiManager.isWifiEnabled() == false) {
            wifiManager.setWifiEnabled(true);
            Toast.makeText(this, "WiFi is disabled, Enabling WiFi",
Toast.LENGTH_LONG).show();
        }

        wifiListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
                String[] wifiListArray = new String[2];
                wifiListArray[0] = wifiList.get(position).SSID;
                wifiListArray[1] = wifiList.get(position).BSSID;
                stopTimer();
                Intent accessPointDetails = new Intent(MainActivity.this,
AccessPointDetails.class);
                accessPointDetails.putExtra("wifiList", wifiListArray);
                startActivity(accessPointDetails);
            }
        });
        startTimer();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.action_about) {
            startActivity(new Intent(this, AboutActivity.class));
            return true;
        }
        if (id == R.id.action_settings) {
            startActivity(new Intent(this, SettingsActivity.class));
        }
        return super.onOptionsItemSelected(item);
    }

    private boolean copyDatabase(Context context) {
```

```
        try {
            byte[] buff = new byte[1024];
            int length = 0;
            InputStream inputStream =
context.getAssets().open(DatabaseHelper.DATABASE_NAME);
            String outFileName = DatabaseHelper.DATABASE_LOCATION +
DatabaseHelper.DATABASE_NAME;
            OutputStream outputStream = new FileOutputStream(outFileName);
            while ((length = inputStream.read(buff)) > 0) {
                outputStream.write(buff, 0, length);
            }
            outputStream.flush();
            outputStream.close();
            return true;
        } catch (Exception ex) {
            ex.printStackTrace();
            return false;
        }
    }

    protected void onPause() {
        unregisterReceiver(wifiReceiver);
        super.onPause();
    }

    protected void onResume() {
        registerReceiver(wifiReceiver, new
IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
        super.onResume();
    }

    private void setAdapter() {
        int counter = 1;
        for (int index = 0; index < wifiList.size(); index++) {
            if (StringUtils.isBlank(wifiList.get(index).SSID)) {
                wifiList.get(index).SSID = "Unknown AP#" + String.valueOf(counter);
                counter++;
            }
        }
        adapter = new ListAccessPoints(getApplicationContext(), wifiList);
        wifiListView.setAdapter(adapter);
    }

    private void scanWifiList() {
        wifiManager.startScan();
        wifiList = wifiManager.getScanResults();
        setAdapter();
    }

    private void stopTimer() {
        if (timer != null) {
            timer.cancel();
            timer.purge();
        }
    }

    private void startTimer() {
        timer = new Timer();
        timerTask = new TimerTask() {
            public void run() {
                handler.post(new Runnable() {
                    public void run() {
                        wifiManager.startScan();
```

```
                        scanWifiList();
                    }
                });
            }
        };

        timer.schedule(timerTask, 1, listRefreshRate);
    }

    private void setRefreshRate() {
        SharedPreferences sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(this);
        String listIndex = sharedPreferences.getString(listRefresh, null);
        String[] mainArray = getResources().getStringArray(R.array.main_by_array);
        String strListRate = mainArray[Integer.parseInt(listIndex)];
        Double listRate = Double.parseDouble(strListRate.replace("Seconds",
"").trim());
        listRefreshRate = (long) (listRate * 1000);
    }

    public class WifiReceiver extends BroadcastReceiver {

        public void onReceive(Context context, Intent intent) {
        }
    }
}
```

ListAccessPoints.java

```
package com.waptracker;

import android.content.Context;
import android.net.wifi.ScanResult;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
import java.util.List;

public class ListAccessPoints extends BaseAdapter {
    private final Context context;
    private final List<ScanResult> wifiList;
    private final LayoutInflater inflater;

    public ListAccessPoints(Context context,  List<ScanResult> wifiList){
        this.context = context;
        this.wifiList = wifiList;
        inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return wifiList.size();
    }

    @Override
    public Object getItem(int position) {
        return wifiList.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent){
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        String strSSID, strBSSID, strLevel;
        View rowView = convertView;
        if (rowView == null){
            rowView = inflater.inflate(R.layout.access_points_list, null);
        }

        TextView txtSSID = (TextView) rowView.findViewById(R.id.SSID);
        TextView txtBSSID = (TextView) rowView.findViewById(R.id.BSSID);
        TextView txtStrength = (TextView) rowView.findViewById((R.id.Strength));
        strSSID = wifiList.get(position).SSID;
        strBSSID = wifiList.get(position).BSSID;
        strLevel = String.valueOf(wifiList.get(position).level);
        txtSSID.setText(strSSID);
        txtBSSID.setText(strBSSID);
        txtStrength.setText(strLevel + " dBm");
        return rowView;
    }
}
```

AccessPointDetails.java

```
package com.waptracker;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.os.Handler;
import android.preference.PreferenceManager;
import android.support.v4.app.NavUtils;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;
import com.waptracker.vendor.DatabaseHelper;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;
import org.apache.commons.lang3.StringUtils;

public class AccessPointDetails extends AppCompatActivity {

    private static final double DISTANCE_MHZ_M = 27.55;
    private DatabaseHelper databaseHelper;
    private Timer timer;
    private TimerTask timerTask;
    private Handler handler = new Handler();
    private boolean completeFlag = true;

    long detailsRefreshRate;
    double distance;
    String[] APDetails = new String[2];
    WifiManager wifiManager;
    WifiReceiver wifiReceiver;
    List<ScanResult> wifiList;
    String strSSID, strBSSID, strSignal, strDistance, strVendor;
    TextView txtSSID, txtBSSID, txtDistance, txtSignal, txtVendor;

    private final static String detailsRefresh = "details_scan_interval";

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

        setRefreshRate();

        APDetails = getIntent().getStringArrayExtra("wifiList");
        strSSID = APDetails[0];
        strBSSID = APDetails[1];
        databaseHelper = new DatabaseHelper(this);

        txtSSID = (TextView) findViewById(R.id.SSID);
        txtBSSID = (TextView) findViewById(R.id.BSSID);
        txtSignal = (TextView) findViewById(R.id.Strength);
        txtDistance = (TextView) findViewById(R.id.Distance);
        txtVendor = (TextView) findViewById(R.id.VendorName);

        txtSSID.setText(strSSID);
        txtBSSID.setText(strBSSID);
```

```
        wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
        wifiReceiver = new WifiReceiver();
        registerReceiver(wifiReceiver, new
IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
        startTimer();
        return;
    }

    @Override
    public void onBackPressed() {
        stopTimer();
        NavUtils.navigateUpFromSameTask(this);
        finish();
    }

    public void setDetails() {
        completeFlag = false;
        String checkSSID, checkBSSID;
        wifiManager.startScan();
        wifiList = wifiManager.getScanResults();

        for (int index = 0; index < wifiList.size(); index++) {
            checkSSID = wifiList.get(index).SSID.toString();
            checkBSSID = wifiList.get(index).BSSID.toString();

            if (StringUtils.isBlank(checkSSID)) {
                if (strBSSID.compareTo(checkBSSID) == 0) {
                    strSignal = String.valueOf(wifiList.get(index).level) + " dBm";
                    strDistance = calculateDistance(wifiList.get(index).frequency,
wifiList.get(index).level);
                    strVendor = databaseHelper.find(checkBSSID.replace(":",
"").substring(0, 6));
                }
            } else {
                if ((strSSID.compareTo(checkSSID) == 0) &&
(strBSSID.compareTo(checkBSSID) == 0)) {
                    strSignal = String.valueOf(wifiList.get(index).level) + " dBm";
                    strDistance = calculateDistance(wifiList.get(index).frequency,
wifiList.get(index).level);
                    strVendor = databaseHelper.find(checkBSSID.replace(":",
"").substring(0, 6));
                }
            }
            if (strSignal != null) {
                txtSignal.setText(strSignal);
            } else {
                txtSignal.setText(getResources().getString(R.string.signal_noData));
            }
            if (strDistance != null) {
                txtDistance.setText(strDistance);
            } else {

txtDistance.setText(getResources().getString(R.string.distance_noData));
            }
            if (strVendor != null) {
                txtVendor.setText(strVendor);
            } else {
                txtVendor.setText(getResources().getString(R.string.vendor_noData));
            }
        }
        completeFlag = true;
        return;
```

```
    }

    public String calculateDistance(int frequency, int level) {
        distance = Math.round((Math.pow(10.0, ((DISTANCE_MHZ_M - (20 *
Math.log10(frequency)) + Math.abs(level)) / 20.0))) * 100D) / 100D;
        return (String.valueOf(distance) + " m");
    }

    protected void onPause() {
        unregisterReceiver(wifiReceiver);
        super.onPause();
    }

    protected void onResume() {
        registerReceiver(wifiReceiver, new
IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
        super.onResume();
    }

    private void stopTimer() {
        if (timer != null) {
            timer.cancel();
            timer.purge();
        }
    }

    private void startTimer() {
        timer = new Timer();
        timerTask = new TimerTask() {
            public void run() {
                handler.post(new Runnable() {
                    public void run() {
                        if (completeFlag){
                            setDetails();
                        }
                    }
                });
            }
        };

        timer.schedule(timerTask, 1, detailsRefreshRate);
    }

    private void setRefreshRate() {
        SharedPreferences sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(this);
        String detailsIndex = sharedPreferences.getString(detailsRefresh, null);
        String[] listArray = getResources().getStringArray(R.array.details_by_array);
        String strListRate = listArray[Integer.parseInt(detailsIndex)];
        Double detailsRate = Double.parseDouble(strListRate.replace("Seconds",
"").trim());
        detailsRefreshRate = (long) (detailsRate * 1000);
    }

    public class WifiReceiver extends BroadcastReceiver {

        public void onReceive(Context context, Intent intent) {
        }
    }
}
```

DatabaseHelper.java

```
package com.waptracker.vendor;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.provider.BaseColumns;

public class DatabaseHelper extends SQLiteOpenHelper implements BaseColumns {

    private static final String TABLE_NAME = "oui";
    private static final String COLUMN_MAC = "mac";
    private static final String COLUMN_NAME = "company";
    public static final String DATABASE_NAME = "android_oui.db3";
    public static final String DATABASE_LOCATION =
"/data/data/com.waptracker/databases/";
    private Context sContext;
    private SQLiteDatabase sqLiteDatabase;

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
        this.sContext = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }

    public void openDatabase() {
        String dbPath = sContext.getDatabasePath(DATABASE_NAME).getPath();
        if ((sqLiteDatabase != null) && (sqLiteDatabase.isOpen())) {
            return;
        }
        sqLiteDatabase = SQLiteDatabase.openDatabase(dbPath, null,
SQLiteDatabase.OPEN_READWRITE);
    }
    public void closeDatabase() {
        if (sqLiteDatabase != null) {
            sqLiteDatabase.close();
        }
        return;
    }
    public String find(String mac) {
        String result = null;
        openDatabase();
        Cursor cursor = sqLiteDatabase.rawQuery("SELECT * FROM " + TABLE_NAME + "
WHERE " + COLUMN_MAC + " = " + "'" + mac.toUpperCase() + "'", null);
        if (cursor.moveToFirst()) {
            result = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAME));
        }
        cursor.close();
        closeDatabase();
        return result;
    }
}
```

SettingsActivity.java

```
package com.waptracker.settings;

import android.os.Bundle;
import android.preference.PreferenceFragment;
import android.support.v4.app.NavUtils;
import android.support.v7.app.AppCompatActivity;
import com.waptracker.R;

public class SettingsActivity extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getFragmentManager().beginTransaction().replace(android.R.id.content, new
SettingPreferenceFragment()).commit();
    }

    @Override
    public void onBackPressed() {
        NavUtils.navigateUpFromSameTask(this);
        finish();
    }
    public static class SettingPreferenceFragment extends PreferenceFragment {

        public static final String KEY_MAIN_INTERVAL = "main_scan_interval";

        @Override
        public void onCreate(final Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            addPreferencesFromResource(R.xml.preferences);
        }
    }
}
```

AboutActivity.java

```
package com.waptracker.about;

import android.os.Bundle;
import android.support.v4.app.NavUtils;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.AppCompatDelegate;
import com.waptracker.R;

public class  AboutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.content_about);
    }
    @Override
    public void onBackPressed() {
        NavUtils.navigateUpFromSameTask(this);
        finish();
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <ListView
            android:id="@+id/lvWifiDetails"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="6dp"
            android:layout_marginBottom="6dp"
            android:layout_marginLeft="6dp"
            android:layout_marginRight="6dp"
            android:dividerHeight="8dp"
            android:divider="@null"
            android:clickable="true">
        </ListView>
    </LinearLayout>
</LinearLayout>
```

access_points_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="horizontal">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:background="@drawable/rounded_corner">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/SSID"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/textColor"
                android:textSize="9pt"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/BSSID"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/textColor"/>

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="right"
            android:orientation="vertical">

            <TextView
                android:id="@+id/Strength"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:textSize="9pt"
                android:textStyle="bold"
                android:textColor="@color/textColor"/>

        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

access_points_details.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="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingRight="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingBottom="@dimen/activity_vertical_margin">

            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:orientation="horizontal">

                <LinearLayout
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="SSID: "
                        android:textSize="10pt"
                        android:textStyle="bold"
                        android:textColor="@color/labelColor"/>

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="BSSID: "
                        android:textSize="10pt"
                        android:textStyle="bold"
                        android:textColor="@color/labelColor"/>

                </LinearLayout>

                <LinearLayout
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">

                    <TextView
                        android:id="@+id/SSID"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:textSize="10pt"
```

```
                    android:textStyle="bold"
                    android:textColor="@color/labelColor"/>

                <TextView
                    android:id="@+id/BSSID"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textSize="10pt"
                    android:textStyle="bold"
                    android:textColor="@color/labelColor"/>

            </LinearLayout>

        </LinearLayout>
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/rounded_corner"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/text_signal_strength"
        android:textColor="@color/textColor"
        android:textSize="@dimen/label_textSize"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/Strength"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textColor="@color/textColor"
        android:textSize="@dimen/value_textSize" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingBottom="@dimen/padding_textview">

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/rounded_corner"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin">
```

```
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="@string/text_distance"
            android:textColor="@color/textColor"
            android:textSize="@dimen/label_textSize"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/Distance"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textColor="@color/textColor"
            android:textSize="@dimen/value_textSize" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingBottom="@dimen/padding_textview">

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/rounded_corner"
        android:orientation="vertical"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_vertical_margin"
        android:paddingRight="@dimen/activity_vertical_margin"
        android:paddingTop="@dimen/activity_vertical_margin">


        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="@string/text_vendor"
            android:textColor="@color/textColor"
            android:textSize="@dimen/label_textSize"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/VendorName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textColor="@color/textColor"
            android:textSize="@dimen/value_textSize" />

    </LinearLayout>

</LinearLayout>
```

content_about.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="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingRight="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:paddingBottom="@dimen/activity_vertical_margin">

            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:contentDescription="@+id/app_name"
                android:src="@drawable/ic_access_point"/>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:paddingLeft="@dimen/activity_vertical_margin"
                android:paddingRight="@dimen/activity_vertical_margin">

                <TextView
                    android:id="@+id/about_app_name"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/app_name"
                    android:textStyle="bold"
                    android:textColor="@color/labelColor"/>

                <TextView
                    android:id="@+id/about_package_name"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:visibility="gone"/>

                <TextView
                    android:id="@+id/about_version_info"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/app_version"
                    android:textColor="@color/labelColor"/>

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/company_name"
```

```
                android:textStyle="bold"
                android:textColor="@color/labelColor"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/copyright"
                android:textColor="@color/labelColor"/>
        </LinearLayout>

    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Idea:"
        android:textStyle="bold|italic"
        android:paddingBottom="6px"
        android:textColor="@color/labelColor"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/app_idea"
        android:textColor="@color/labelColor"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/app_idea_email"
        android:textColor="@color/labelColor"/>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Developed by:"
        android:textStyle="bold|italic"
        android:paddingBottom="6px"
        android:textColor="@color/labelColor"/>

    <TextView
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/app_developer"
            android:textColor="@color/labelColor"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/app_developer_email"
            android:textColor="@color/labelColor"/>

    </LinearLayout>
</LinearLayout>
```

preferences.xml

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <ListPreference
        android:defaultValue="@integer/main_scan_interval_default"
        android:summary="@string/main_scan_interval_summary"
        android:entries="@array/main_by_array"
        android:entryValues="@array/main_by_index_array"
        android:key="@string/main_scan_interval_key"
        android:persistent="true"
        android:title="@string/main_scan_interval_title" />

    <ListPreference
        android:defaultValue="@integer/details_scan_interval_default"
        android:summary="@string/details_scan_interval_summary"
        android:entries="@array/details_by_array"
        android:entryValues="@array/details_by_index_array"
        android:key="@string/details_scan_interval_key"
        android:persistent="true"
        android:title="@string/details_scan_interval_title" />
</PreferenceScreen>
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/action_about"
        android:orderInCategory="1"
        android:title="@string/action_about"
        app:showAsAction="never" />

    <item
        android:id="@+id/action_settings"
        android:orderInCategory="2"
        android:title="@string/action_settings"
        app:showAsAction="never" />

</menu>
```

arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string-array name="main_by_array">
        <item>5 Seconds</item>
        <item>10 Seconds</item>
        <item>15 Seconds</item>
        <item>20 Seconds</item>
        <item>25 Seconds</item>
        <item>30 Seconds</item>
    </string-array>

    <string-array name="main_by_index_array" translatable="false">
        <item>0</item>
        <item>1</item>
        <item>2</item>
        <item>3</item>
        <item>4</item>
        <item>5</item>
    </string-array>

    <string-array name="details_by_array">
        <item>2.5 Seconds</item>
        <item>3.0 Seconds</item>
        <item>3.5 Seconds</item>
        <item>4.0 Seconds</item>
        <item>4.5 Seconds</item>
        <item>5.0 Seconds</item>
        <item>5.5 Seconds</item>
        <item>6.0 Seconds</item>
        <item>6.5 Seconds</item>
        <item>7.0 Seconds</item>
        <item>7.5 Seconds</item>
        <item>8.0 Seconds</item>
        <item>8.5 Seconds</item>
        <item>9.0 Seconds</item>
    </string-array>

    <string-array name="details_by_index_array" translatable="false">
        <item>0</item>
        <item>1</item>
        <item>2</item>
        <item>3</item>
        <item>4</item>
        <item>5</item>
        <item>6</item>
        <item>7</item>
        <item>8</item>
        <item>9</item>
        <item>10</item>
        <item>11</item>
        <item>12</item>
        <item>13</item>
    </string-array>

</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#337AFF</color>
    <color name="colorPrimaryDark">#337AFF</color>
    <color name="colorAccent">#337AFF</color>
    <color name="textColor">#FFFFFF</color>
    <color name="labelColor">#000000</color>
</resources>
```

dimens.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="padding_textview">10pt</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="label_textSize">9pt</dimen>
    <dimen name="value_textSize">8pt</dimen>
</resources>
```

styles.xml

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

strings.xml

```
<resources>
    <string name="app_name">WAP Tracker</string>

    <!-->Strings for Menu Items</!-->
    <string name="action_about">About</string>
    <string name="action_details">Details</string>
    <string name="action_settings">Settings</string>

    <!-->Strings for About Page</!-->
    <string name="company_name">Wolf Raven Technologies</string>
    <string name="app_version">1.0.0</string>
    <string name="copyright">Copyright \u00A9 2016 \u002D 2017</string>
    <string name="app_idea">Leonard Rogers</string>
    <string name="app_idea_email">lenr@shaw.ca</string>
    <string name="app_developer">Ishant Dhanani</string>
    <string name="app_developer_email">ishant@ualberta.ca</string>

    <!-->Strings for Access Point Details Page</!-->
    <string name="text_signal_strength">Signal Strength</string>
    <string name="text_distance">Distance To</string>
    <string name="text_vendor">Vendor</string>

    <!-->Strings for No Data Available</!-->
    <string name="signal_noData">No Data Available</string>
    <string name="distance_noData">No Data Available</string>
    <string name="vendor_noData">No Data Available</string>

    <!-->Values for Settings Page</!-->
    <string name="button_ok">OK</string>
    <string name="button_cancel">Cancel</string>

    <integer name="main_scan_interval_default" translatable="false">1</integer>
    <string name="main_scan_interval_title">List Scan Interval</string>
    <string name="main_scan_interval_summary">%s between each scan</string>
    <string name="main_scan_interval_key"
translatable="false">main_scan_interval</string>

    <integer name="details_scan_interval_default" translatable="false">5</integer>
    <string name="details_scan_interval_title">Details Scan Interval</string>
    <string name="details_scan_interval_summary">%s between each scan</string>
    <string name="details_scan_interval_key"
translatable="false">details_scan_interval</string>

</resources>
```

Appendix – B

Screenshots

Figure 1. Screenshot depicting List of Access Points

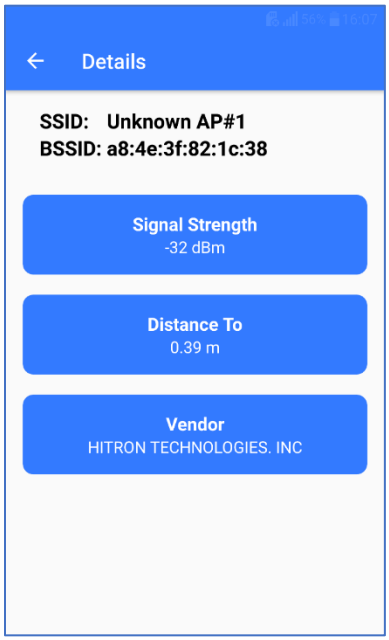

Figure 2. Screenshot depicting Details of Access Point not broadcasting SSID

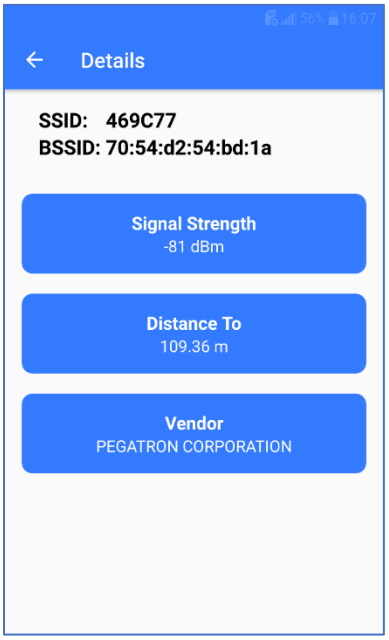

Figure 3. Screenshot depicting Details of Access Point broadcasting SSID

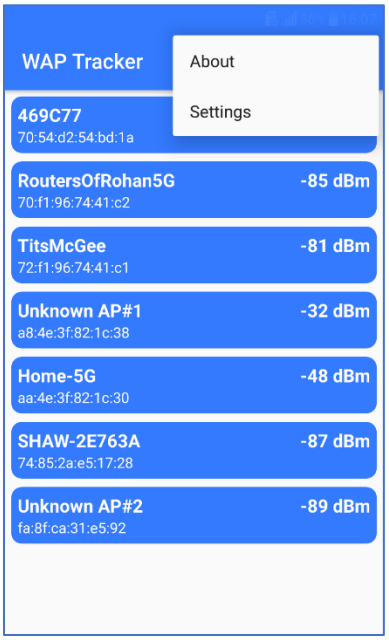

Figure 4. Screenshot depicting Menu

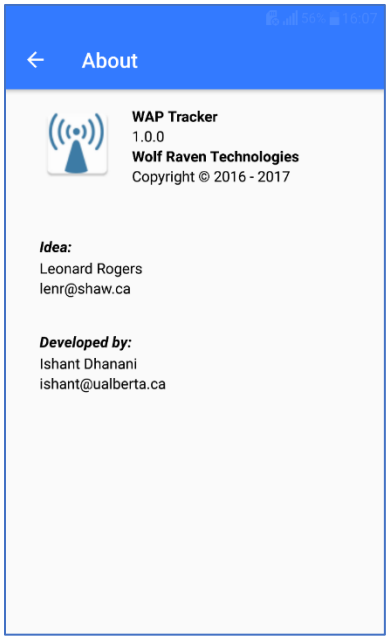

Figure 5. Screenshot depicting About Page

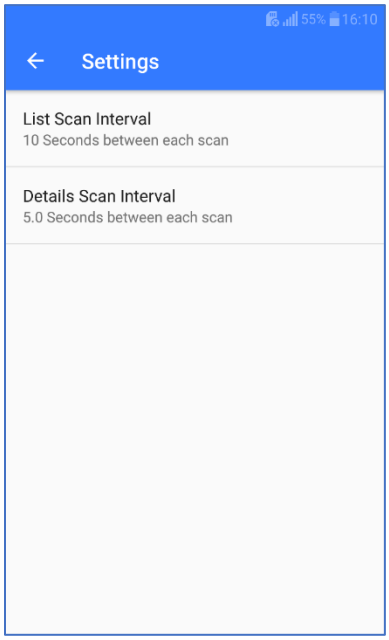

Figure 6. Screenshot depicting Settings Page

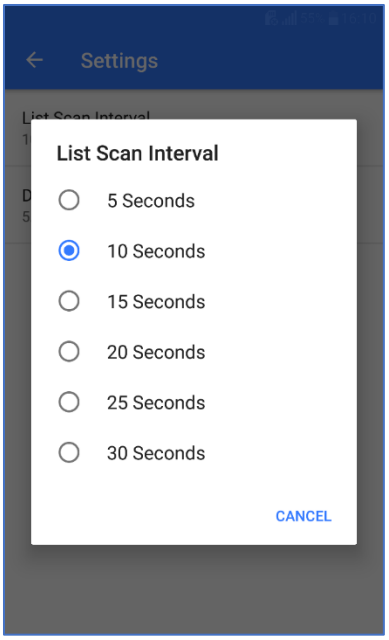

Figure 7. Screenshot depicting List Scan Interval

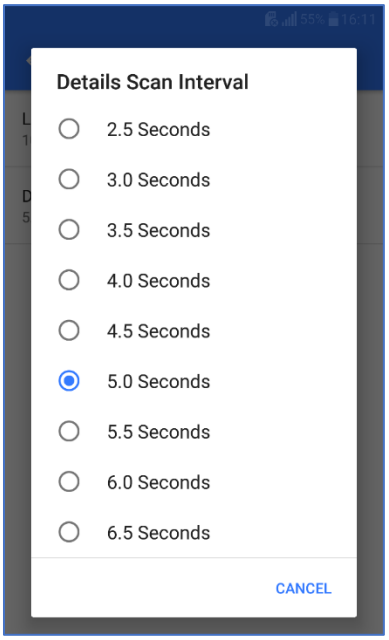

Figure 8. Screenshot depicting Details Scan Interval