11.10.2020

ERSTELLUNG EINER ANDROIDAPP ZUR ERFASSUNG VONEDUROAM NETZWERKEN

Mit der Google Maps API

Falah Askar
HOCHSCHULE BOCHUM

Inhaltsverzeichnis

1 Einleitung

Im Rahmen des Masterstudienganges Geoinformatik an der Hochschule Bochum, wird das Modul „mobile mapping applications“ angeboten . In Modul wurden Inhalte vermittelt, die dazu dienen mobile Kartenanwendungen umzusetzen. Unter anderem wurden verschiedene Mapping APIs vorgestellt und getestet. Als Zielplattform wurde das mobile Betriebssystem Android gewählt und entsprechende Grundlagen erarbeitet. Zum Abschluss des Moduls soll eine App entwickelt werden, mit der es möglich ist, dass Hochschulnetzwerk „Eduroam“ zu kartieren und die Verfügbarkeit in einer Karte darzustellen.

1.1 Aufgabenstellung

Die Aufgabe bestand darin, eine mobile Kartenanwendung für die Android Plattform zu entwickeln.Mit dieser soll das Hochschulnetzwerk „Eduroam“ kartiert werden und die Verfügbarkeit des Netzwerks in einer Karte grafisch dargestellt werden. Für die Umsetzung ist es freigestellt, welche Mapping API verwendet wird. Die Umsetzung soll in drei Iterationen erfolgen, wobei die ersten beiden zwingend erforderlich sind.

1.1.1 Iteration 1

Während der ersten Iteration soll das Grundgerüst der Anwendung aufgebaut werden. Nach dieser Iteration wird ein Hintergrunddienst erstellt, der die Verfügbarkeit eines bestimmten Wifi-Netzes prüfen soll. Sobald das entsprechende Netzwerk erkannt wurde, soll der Nutzer durch eine beliebige Ausgabemethode darauf aufmerksam gemacht werden.

1.1.2 Iteration 2

Die zweite Iteration sieht vor, dass die Position des Smartphones gespeichert wird, sobald das Netzwerk erkannt wurde. Die Speicherung wird dazu zunächst lokal vorgenommen werden. Dies kann zum Beispiel mit Hilfe einer SQLite Datenbank geschehen. Außerdem sollen die gespeicherten Positionen als Punkte in einer Karte dargestellt werden.

2 Grundlagen

2.1 Android Betriebssystem

Android ist sowohl ein Betriebssystem als auch eine Software-Plattform für mobile Geräte wie Smartphones, Tabletcomputer, Fernseher, Mediaplayer, Netbooks und Autos, die von der von Google gegründeten Open Handset Alliance entwickelt werden. (wikipedia)

Um eine Android-App herzustellen , hat der Entwickler nach seiner Wahl die zwei Programmiersprachen Java und Kotlin Für die Entwicklung existiert eine spezielle Entwicklungsumgebung Namens „Android Studio“, welche auf der Intellij IDEAbasiert.

2.1.1 Services

Android hat eine App-Komponente, die Android Service, dieser Komponente wird im Hintergrund ausgeführt, um permanent eine bestimmte Operation durchzuführen, diesen Operationen können Internet-Downloads, Überprüfen auf neue Daten, Datenverarbeitung sein. (Big App)

```java
class LocationService extends Service implements LocationListener {
    protected LocationManager locationManager;
    Location location;

    private static final long MIN_DISTANCE_FOR_UPDATE = 0;
    private static final long MIN_TIME_FOR_UPDATE = 1000;

    public LocationService(Context context) {
        locationManager = (LocationManager) context.getSystemService(LOCATION_SERVICE);
    }

    @SuppressLint("MissingPermission")
    public Location getLocation(String provider) {
        if (locationManager.isProviderEnabled(provider)) {
            locationManager.requestLocationUpdates(provider, MIN_TIME_FOR_UPDATE, MIN_DISTANCE_FOR_UPDATE,  listener: this);
            if (locationManager != null) {
                location = locationManager.getLastKnownLocation(provider);
                return location;
            }
        }
        return null;
    }

    @Override
    public IBinder onBind(Intent intent) { return null; }
```

Abbildung 1 Services in Android Studio

2.2 Google Maps API

Google Maps wird aufgrund seiner hohen Genauigkeit, Qualität und einzigartigen Effizienz als dass am häufigsten verwendete im Internet angesehen. Auch der Name des riesigen Google-Unternehmens, das dahintersteht, spielt eine große Rolle bei der Besetzung dieser Position durch die große Unterstützung, die ihm gewährt wird. Wenn Sie in einem professionellen Projekt arbeiten, wird die Verwendung der Google Maps API-Plattform in vielen Fällen zu einer unvermeidlichen Notwendigkeit.

Um Google Maps API in eine Android Projekt verwendet werden , muss der Nutzer erst sich in Google Maps Plattform anmelden und eine kostenloser Schlüssel API erreichen und dieser API wird in Android Projekt gespeichert , Danach hat der Nutzer Zugriff Google Karte zu Nutzen .Eine Anmeldung wird unter dieser URL

https://console.cloud.google.com/apis/credentials?hl=DE&_ga=2.194268269.1948315339.1602428204-1400977891.1590406106&project=upbeat-palace-277923&folder=&organizationId=

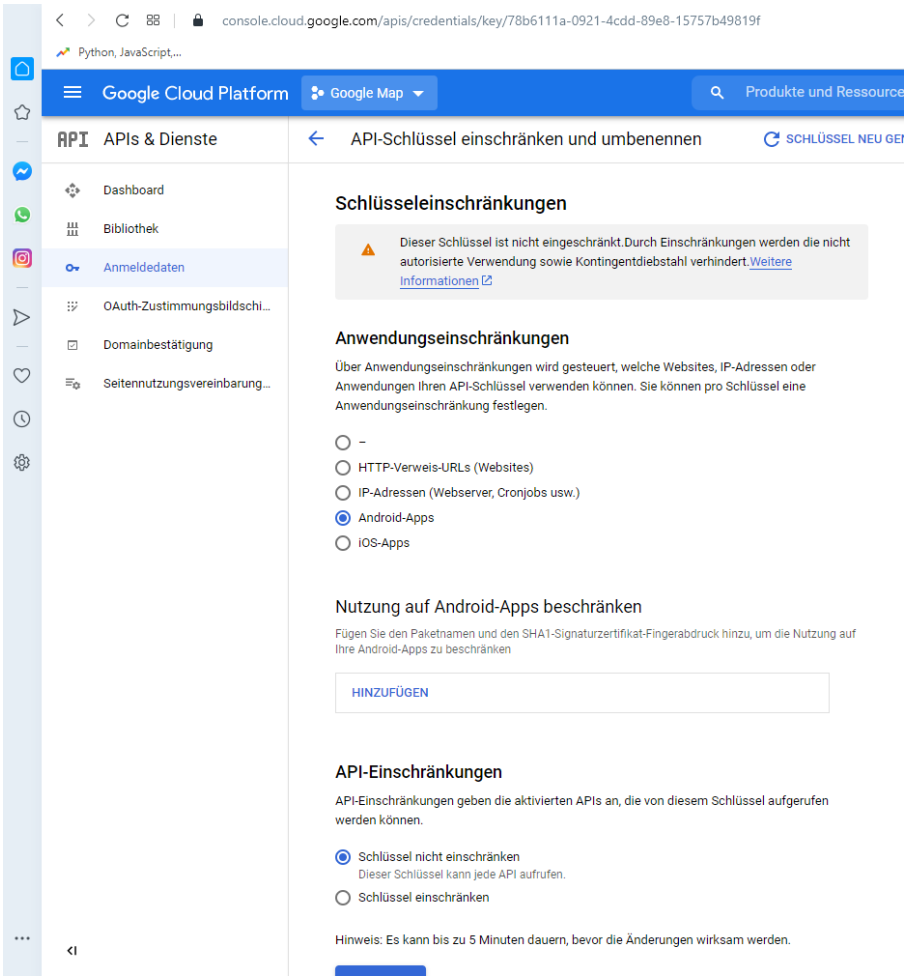


Abbildung 2 Anmeldung für Google Maps API

Am besten ist die Auswahl Android Studio in Anwendungenseinschränkungen und Schlüssel nicht einschränken in API-einstellungen aktivieren, damit das Arbeit Fehlerfrei in Android Studio implementiert werden.

Das Google Maps API muss in einer Value-Ordner in google_maps_api.xml gespeichert werden .

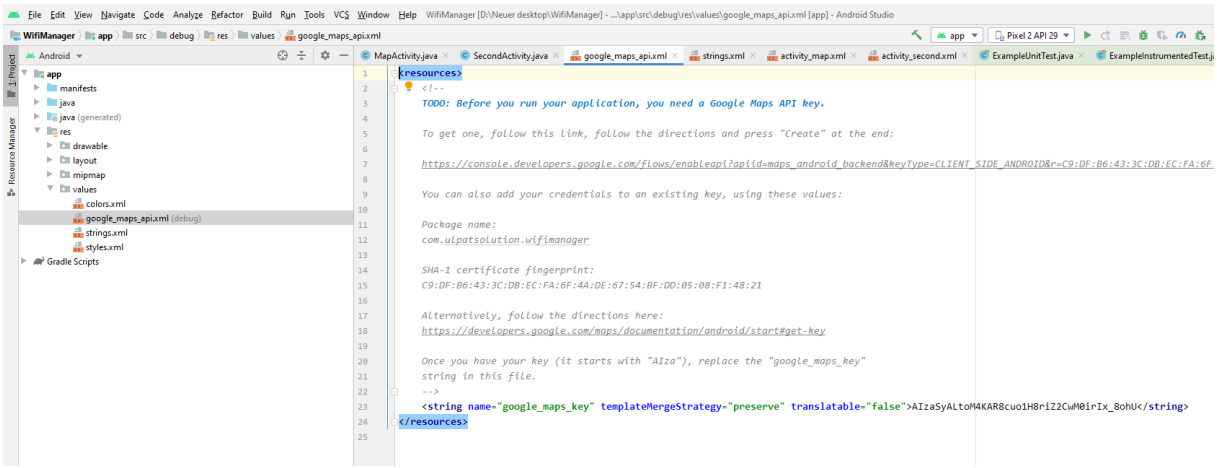
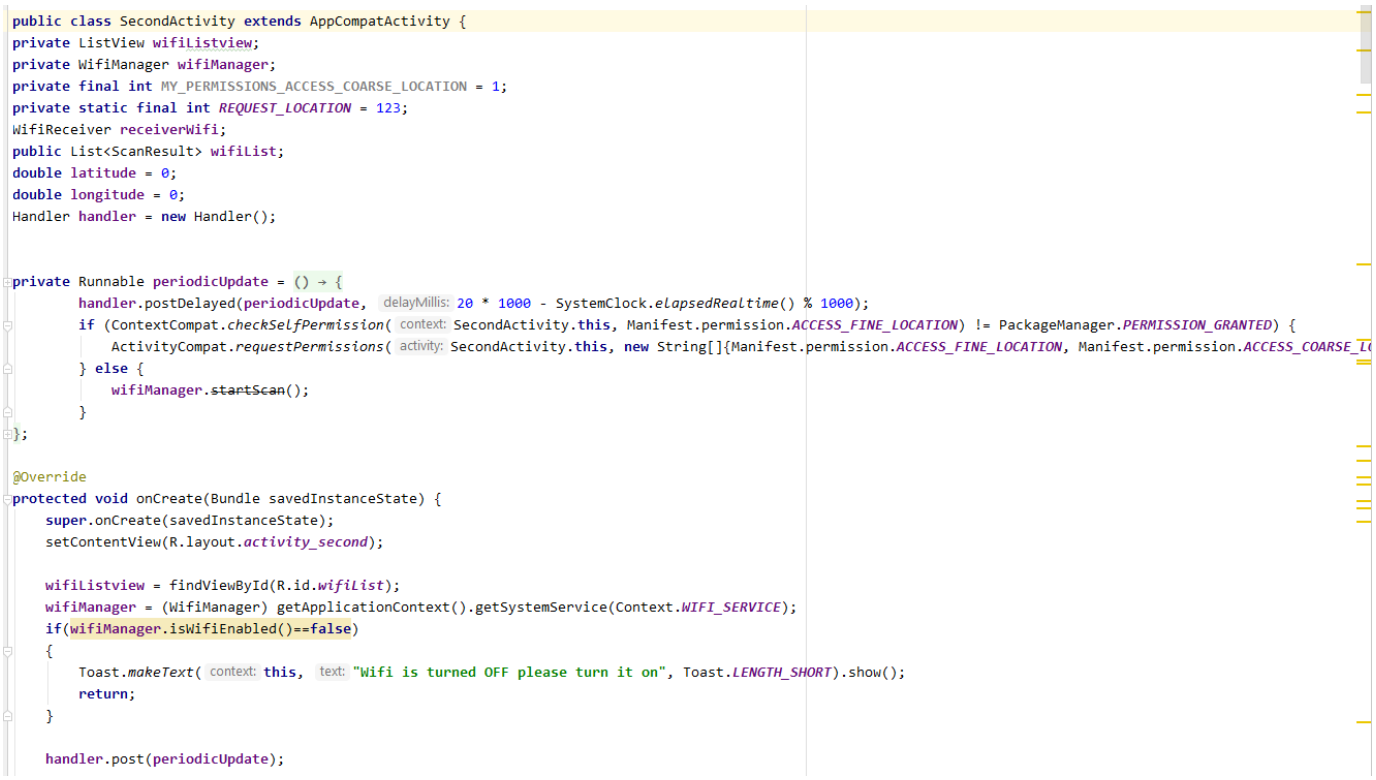


Abbildung 3 google_maps_api

3 Implementierung

3.1 Iteration 1 - sucht nach Eduroam-Netzwerk

Die erste Iteration soll dazu dienen ein Grundgerüst zu bilden, welches die grundlegenden Funktionen abdeckt. Dazu soll ein Hintergrundservice erstellt werden, der die verfügbaren Wifi-Netzwerke prüft. Dabei sollen die SSIDs der Netzwerke mit der SSID des Eduroam-Netzwerkes verglichen werden, um zu prüfen ob dieses verfügbar ist. Sollte das Netzwerk verfügbar sein, soll eine entsprechende Mitteilung ausgegeben werden.



```
public class SecondActivity extends AppCompatActivity {
private ListView wifiListview;
private WifiManager wifiManager;
private final int MY_PERMISSIONS_ACCESS_COARSE_LOCATION = 1;
private static final int REQUEST_LOCATION = 123;
WifiReceiver receiverWifi;
public List<ScanResult> wifiList;
double latitude = 0;
double longitude = 0;
Handler handler = new Handler();


private Runnable periodicUpdate = () -> {
        handler.postDelayed(periodicUpdate, delayMillis: 20 * 1000 - SystemClock.elapsedRealtime() % 1000);
        if (ContextCompat.checkSelfPermission( context: SecondActivity.this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions( activity: SecondActivity.this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LO
        } else {
            wifiManager.startScan();
        }
};

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

    wifiListview = findViewById(R.id.wifiList);
    wifiManager = (WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE);
    if(wifiManager.isWifiEnabled()==false)
    {
        Toast.makeText( context: this,  text: "Wifi is turned OFF please turn it on", Toast.LENGTH_SHORT).show();
        return;
    }

    handler.post(periodicUpdate);
```

Abbildung 4 WiFi-Scann

Damit die Funktionalität abdecken, wird zwei Klassen DBHelper und SecondActivity , die SecondActivity ist dafür zuständig die verfügbaren WiFi in Hintergrunddienst mit Eduroam zu vergleichen .

Android ermöglicht Anwendungen den Zugriff, um den Zugriff auf den Status der drahtlosen Verbindungen auf sehr niedriger Ebene anzuzeigen. Die Anwendung kann auf fast alle Informationen einer WLAN-Verbindung zugreifen. Um die verfügbaren WiFi zu finden, ist Fore-Service zu nutzen

Android bietet die WifiManager-API zur Verwaltung aller Aspekte der WIFI Konnektivität. Wir können diese Klasse instanziieren, indem wir die Methode **getSystemService** aufrufen (siehe Abb 5) (tutorialspoint-getService)

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

    wifiListview = findViewById(R.id.wifiList);
    wifiManager = (WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE);
    if(wifiManager.isWifiEnabled()==false)
    {
        Toast.makeText( context: this,  text: "Wifi is turned OFF please turn it on", Toast.LENGTH_SHORT).show();
        return;
    }

    handler.post(periodicUpdate);
```

Abbildung 5 getSystemService

Um eine Liste von Verfügbaren zu scannen, muss Ihren Broadcast Receiver registrieren, das kann mit der registerReceiver-Methode mit dem Argument Ihres Empfängerklassenobjekts registriert werden.

(sieh Abb 6) (tutorialspoint-broadcast_receivers)

```java
class WifiReceiver extends BroadcastReceiver {
    WifiManager wifiManager;
    StringBuilder sb;
    ListView wifiDeviceList;

    public WifiReceiver(WifiManager wifiManager, ListView wifiDeviceList) {
        this.wifiManager = wifiManager;
        this.wifiDeviceList = wifiDeviceList;
    }

    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (WifiManager.SCAN_RESULTS_AVAILABLE_ACTION.equals(action)) {

            wifiList = wifiManager.getScanResults();
            ArrayList<String> deviceList = new ArrayList<>();
            for (ScanResult scanResult : wifiList) {
                deviceList.add(scanResult.SSID);
            }
            ArrayAdapter arrayAdapter = new ArrayAdapter(context, android.R.layout.simple_list_item_1, deviceList.toArray());
            wifiDeviceList.setAdapter(arrayAdapter);
            SaveButton();
```

Abbildung 6 BroadcastReceiver

WiFi Receiver erbt BroadcastReceiver und in dieser inneren Klasse wird onReceive überschrieben. Der Zugriff auf die Netzwerke erfolgt über den Systemservice namens

WIFI_SERVICE , Systemservice scannt im Hintergrund nach WiFis Netzwerke , wenn eine neue WiFI gefunden wird , wirft einen Broadcast *SCAN_RESULTS_AVAILABLE_ACTION* .
Wenn WiFi gescannt wird , wird ihrer SSID abgefragt und mit geforderten WiFi(Eduroam) verglichen , wenn es passt , wird der Benutzer darauf benachrichtigen und seine Standort gespeichert und auf der Karte gezeigt (vgl. 3.2Iteration 2) .

3.2 Iteration 2–lokale Speicherung von Eduroam-Positionen

Wenn die Netzwerke verfügbar sind , sollte es möglich sein die Standort abzufragen Die entsprechende Position soll dann lokal gespeichert werden. In Rahmen dieses Arbeit wird Standort von Eduroam in eine SQLite Daten Bank gespeichert , dazu wird eine besonderes Klasse DBHelper erstellt , dieser Klasse erbt SQLiteOpenHelper und wird Attribute für Latitude , longitude ID usw. erzeugt (sieh Abb 7) : (Android-Dokument SQLite)

```
public class DBHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "MyDB.db";
    public static final String DATA_TABLE_NAME = "DATA";
    public static final String DATA_COLUMN_ID = "ID";
    public static final String DATA_COLUMN_WIFIRESULT = "WIFIRESULT";
    public static final String DATA_COLUMN_DATE = "MYDATE";
    public static final String DATA_COLUMN_GLAT = "GLAT";
    public static final String DATA_COLUMN_GLON = "GLON";
    public static final String DATA_COLUMN_NLAT = "NLAT";
    public static final String DATA_COLUMN_NLON = "NLON";
    public DBHelper(Context context) {
        super(context, DATABASE_NAME , factory: null, version: 1);
    }
```

Abbildung 7 SQLite Daten Bank

Und in Klasse SecondActivity wird eine Objekt dafür erstellt Eduroam Position in Daten Bank zu speichern , nach dem die Verfügbaren WiFis geprüft werden und mit Eduroam verglichen werden ,(sieh Abb 8).

```
if (wifiList.size() > 0) {
    int eduromfound=0;
    for (int i = 0; i < wifiList.size(); i++) {
        ScanResult scanResult = wifiList.get(i);
        if(scanResult.SSID.contains("eduroam"))
        {
            eduromfound++;
        }
    }
    if(eduromfound==0){
        Toast.makeText( context: this, text: "eduroam Network Not Found", Toast.LENGTH_SHORT).show();
        LoadMap();
        return;
    }
```

Abbildung 8 Eduroam

wird eine neue Daten Bank für Eduroam erstellt und die Lokale Positionen gespeichert (sieh Abb 9) , Danach wird durch Google Maps API wird die gespeicherten Positionen auf die Karte angezeigt .

Abbildung 9 Ergebnis

Fazit und Ausblick

Zum Abschloss dieser Aufgabe wird eine Android App erstellt, mit dem kann Standort von Eduroam-Netzwerke in eine SQLite Daten-Bank gespeichert werden. Als Grundlagen in dieser Arbeit wird Fore-Service genutzt. In dieses App soll es möglich sein, dass die verfügbare WiFi-Netzwerke gesucht wird und mit einer Eduroam SSID verglichen wird, falls Eduroam verfügbar ist, wird das automatisch auf der Karte angezeigt. Die Standorte werden gespeichert bleiben, wenn der Nutzer die Standorte löschen möchte, muss das App erneut installiert wird.

Literaturverzeichnis

Big App: Service. Available online at https://www.big-app.de/alles-zu-services-unter-android/.

tutorialspoint-broadcast_receivers. Available online at https://www.tutorialspoint.com/android/android_broadcast_receivers.htm.

tutorialspoint-getService. Available online at https://www.tutorialspoint.com/android/android_wi_fi.htm.

wikipedia. Available online at https://de.wikipedia.org/wiki/Android_(Betriebssystem).

Abbildungsverzeichnis