

## ДОДАТОК А

## Лістинг коду веб-інтерфейсу

○ index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Master Paper Project</title>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

○ main.js:

```
import Vue from 'vue';
import App from './App';
import VueFire from 'vuefire';
import * as VueGoogleMaps from 'vue2-google-maps';

Vue.use(VueFire);
Vue.use(VueGoogleMaps, {
  load: {
    key: 'some_secret_key',
    v: '3.26',
  },
});
new Vue({
  el: '#app',
  render: app => app(App),
});
```

○ Firebase.js:

```
import Firebase from 'firebase';

const config = {
  messagingSenderId: '1234567890',
  apiKey: 'some_secret_key',
  storageBucket: 'master-paper-project.appspot.com',
  authDomain: 'master-paper-project.firebaseio.com',
  databaseURL: 'https://master-paper-project.firebaseio.com',
};
const firebase = Firebase.initializeApp(config);

export default firebase.database();
```

- package.json:

```
{
  "name": "master-paper-project",
  "version": "0.0.1",
  "description": "Master Paper Project",
  "author": "Andriy Byk",
  "private": true,
  "scripts": {
    "dev": "node build/dev-server.js",
    "build": "node build/build.js"
  },
  "dependencies": {
    "firebase": "^3.6.7",
    "marker-clusterer-plus": "^2.1.4",
    "vue": "^2.1.0",
    "vue2-google-maps": "^0.5.2",
    "vuefire": "^1.3.0"
  },
  "devDependencies": {...},
  "engines": {
    "node": ">= 4.0.0",
    "npm": ">= 3.0.0"
  }
}
```

- Map.vue:

```
<template>
  <gmap-map :zoom="zoom" :center="center" :map-type-id="mapType">
    <gmap-marker
      v-for="marker in activeMarkers"
      v-if="marker.enabled"
      :position="marker.position"
      :opacity="marker.opacity"
      :draggable="marker.draggable"
      @click="markerClicked(marker)"
    >
      <gmap-info-window
        :opened="marker.ifw"
        :content="marker.ifw2text"
      >
      </gmap-info-window>
    </gmap-marker>
    <gmap-polyline
      v-if="polylineVisible"
      :path="polylinePaths"
      :options="{strokeColor: '#F00'}"
    >
    </gmap-polyline>
  </gmap-map>
</template>
```

```

<script>
  import Firebase from './Firebase';

  export default {
    name: 'map',
    firebase: {
      locations: Firebase.ref('locations'),
    },
    data() {
      return {
        zoom: 7,
        center: {lat: 49.5646126, lng: 25.6349196},
        mapType: 'roadmap',
        mapStyle: 'green',
        scrollwheel: true,
        polylineVisible: true,
        polylinePaths: [],
        markers: [],
      };
    },
    computed: {
      activeMarkers() {
        this.polylinePaths = [];
        this.markers = [];

        for (const location in this.locations) {
          if (this.locations.hasOwnProperty(location)) {
            const position = {
              lat: this.locations[location].lat,
              lng: this.locations[location].lng,
            };

            this.polylinePaths.push(position);
            this.markers.push({
              enabled: true,
              position,
              ifw: false,
              ifw2text:
this.locations[location].city,
            });
          }
        }

        return this.markers;
      },
    },
    methods: {
      markerClicked(marker) {marker.ifw = !marker.ifw; },
    },
  };
</script>

```

- App.vue:

```

<template>
  <section class="content">
    <aside class="sidebar">
      <h1 class="slogan">Master Paper Project</h1>
    </aside>
    <googleMap/>
  </section>
</template>

<script>
  import Map from './components/Map';

  export default {
    name: 'app',
    components: {
      googleMap: Map,
    },
  };
</script>

<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }
  body {
    width: 100vw;
    height: 100vh;
    cursor: default;
    font: 300 14px Arial, sans-serif;
  }
  .content {
    display: flex;
    flex-direction: row;
  }
  .sidebar {
    padding: 25px;
    flex: 1 0 300px;
    background: #f6f6f6;
  }
  .slogan {
    font-size: 18px;
    text-align: center;
    margin-bottom: 25px;
    padding-bottom: 25px;
    border-bottom: 1px solid;
  }
  .vue-map-container { flex: 1 0 calc(100% - 300px); }
</style>

```

## ДОДАТОК Б

## Лістинг коду Android додатку

○ AndroidManifest.xml:

```

    <?xml version="1.0" encoding="utf-8"?>
    <manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.andriybyk.masterpaper">

        <uses-permission android:name="android.permission.INTERNET"/>
        <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>

        <application
            android:allowBackup="true"
            android:icon="@mipmap/ic_launcher"
            android:label="@string/app_name"
            android:supportsRtl="true"
            android:theme="@style/AppTheme">
            <activity android:name=".MainActivity">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN"/>
                    <category
android:name="android.intent.category.LAUNCHER"/>
                </intent-filter>
            </activity>
        </application>

    </manifest>

```

○ build.gradle:

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.example.andriybyk.masterpaper"
        minSdkVersion 17
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
}

```

```

        buildTypes {
            release {
                minifyEnabled false
                proguardFiles getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro'
            }
        }
    }

    dependencies {
        compile fileTree(dir: 'libs', include: ['*.jar'])
        androidTestCompile('com.android.support.test.espresso:espresso-
core:2.2.2', {
            exclude group: 'com.android.support', module: 'support-
annotations'
        })
        compile 'com.android.support:appcompat-v7:25.1.1'
        compile 'com.google.firebase:firebase-core:10.0.1'
        compile 'com.google.firebase:firebase-database:10.0.1'
        testCompile 'junit:junit:4.12'
    }

    apply plugin: 'com.google.gms.google-services'

```

○ activity\_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="com.example.andriybyk.masterpaper.MainActivity">

    <TextView
        android:text="Andriy Byk © 2017"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/copyright"
        android:textSize="12sp"
        android:fontFamily="sans-serif"/>

```

```

        <Switch
            android:text="Automatically  "
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
android:layout_centerVertical="true"
            android:layout_centerHorizontal="true"
android:id="@+id/auto"/>
        <TextView
            android:text="Please, Press the Button Below\nto Send
Your Current Location:"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
android:layout_alignBottom="@+id/requestLocation"
            android:layout_centerHorizontal="true"
android:layout_marginBottom="70dp" android:id="@+id/textView"
            android:textSize="18sp"
android:textAlignment="center"/>
        <Button
            android:text="S e n d   L o c a t i o n"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/requestLocation"
            android:layout_marginBottom="19dp"
android:layout_above="@+id/auto" android:layout_centerHorizontal="true"/>
    </RelativeLayout>

```

o MainActivity.java:

```

package com.example.andriybyk.masterpaper;

import android.*;
import android.content.pm.PackageManager;
import android.os.Build;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class MainActivity extends AppCompatActivity {

    DatabaseReference mRootRef = FirebaseDatabase
        .getInstance().getReference();
    DatabaseReference mTestRef = mRootRef.child("locations");
    Button btnShowLocation;
    GPSTracker gpsTracker;
    Locations locations;
    Toast toast;

```

```

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

    btnShowLocation = (Button)
findViewById(R.id.requestLocation);

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION,
android.Manifest.permission.INTERNET}, 10);

            return;
        }
    } else {
        sendRequest();
    }
}

@Override
public void onRequestPermissionsResult(int requestCode,
@NonNull String[] permissions, @NonNull int[] grantResults) {
    switch (requestCode) {
        case 10:
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                sendRequest();
            }

            return;
        }
    }

private void sendRequest() {

    btnShowLocation.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            gpsTracker = new GPSTracker(MainActivity.this);

            if (gpsTracker.canGetLocation()) {
                double latitude = gpsTracker.getLatitude();
                double longitude = gpsTracker.getLongitude();
                locations = new Locations(latitude, longitude);
                mTestRef.push().setValue(locations);
            }
        }
    });
}

```

```

        toast = Toast.makeText(
            getApplicationContext(),
            "Your Location:\nLat: " + latitude +
"\nLng: " + longitude, Toast.LENGTH_LONG
        );
        toast.show();
    } else {
        gpsTracker.showSettingsAlert();
    }
    });
}

}
}
}

```

- Tracker.java:

```

package com.example.andriybyk.masterpaper;

import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.support.annotation.Nullable;

public class GPSTracker extends Service
implements LocationListener {

    private final Context context;

    boolean isGPSEnabled = false;
    boolean isNetworkEnabled = false;
    boolean canGetLocation = false;

    Location location;

    double latitude;
    double longitude;

    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 5;
    private static final long MIN_TIME_BETWEEN_UPDATES = 5000;

    protected LocationManager locationManager;

```

```

public GPSTracker(Context context) {
    this.context = context;
    getLocation();
}

public Location getLocation() {
    try {
        locationManager = (LocationManager)
context.getSystemService(LOCATION_SERVICE);

        isGPSEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

        isNetworkEnabled =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnabled && !isNetworkEnabled) {
        } else {
            this.canGetLocation = true;

            if (isNetworkEnabled) {
                locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_TIME_BETWEEN_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES,
                    this
                );

                if (locationManager != null) {
                    location =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }

            if (isGPSEnabled) {
                if (location == null) {
                    locationManager.requestLocationUpdates(
                        LocationManager.GPS_PROVIDER,
                        MIN_TIME_BETWEEN_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES,
                        this
                    );
                }
            }
        }
    }
}

```

```

        if (locationManager != null) {
            location
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);

            if (location != null) {
                latitude = location.getLatitude();
                longitude
location.getLongitude();
            }
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}

return location;
}

public void stopUsingGPS() {
    if (locationManager != null) {
        locationManager.removeUpdates(GPSTracker.this);
    }
}

public double getLatitude() {
    if (location != null) {
        latitude = location.getLatitude();
    }

    return latitude;
}

public double getLongitude() {
    if (location != null) {
        longitude = location.getLongitude();
    }

    return longitude;
}

public boolean canGetLocation() {
    return this.canGetLocation;
}

public void showSettingsAlert() {
    AlertDialog.Builder alertDialog = new
AlertDialog.Builder(context);
    alertDialog.setTitle("GPS settings!");

```

```

        alertDialog.setMessage("GPS is not enabled. Do you want to
go to settings menu?"); alertDialog.setPositiveButton("Settings", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
{Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                context.startActivity(intent);
            }
        });
        alertDialog.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
{dialog.cancel();}
        });
        alertDialog.show();
    }

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

    @Override
    public void onLocationChanged(Location location) {}

    @Override
    public void onStatusChanged(String provider, int status) {}

    @Override
    public void onProviderEnabled(String provider) {}

    @Override
    public void onProviderDisabled(String provider) {}
}

```

o Locations.java:

```

package com.example.andriybyk.masterpaper;

public class Locations {
    public double lat;
    public double lng;
    public String city;

    public Locations(double lat, double lng, String city) {
        this.lat = lat;
        this.lng = lng;
        this.city = city;
    }
}

```