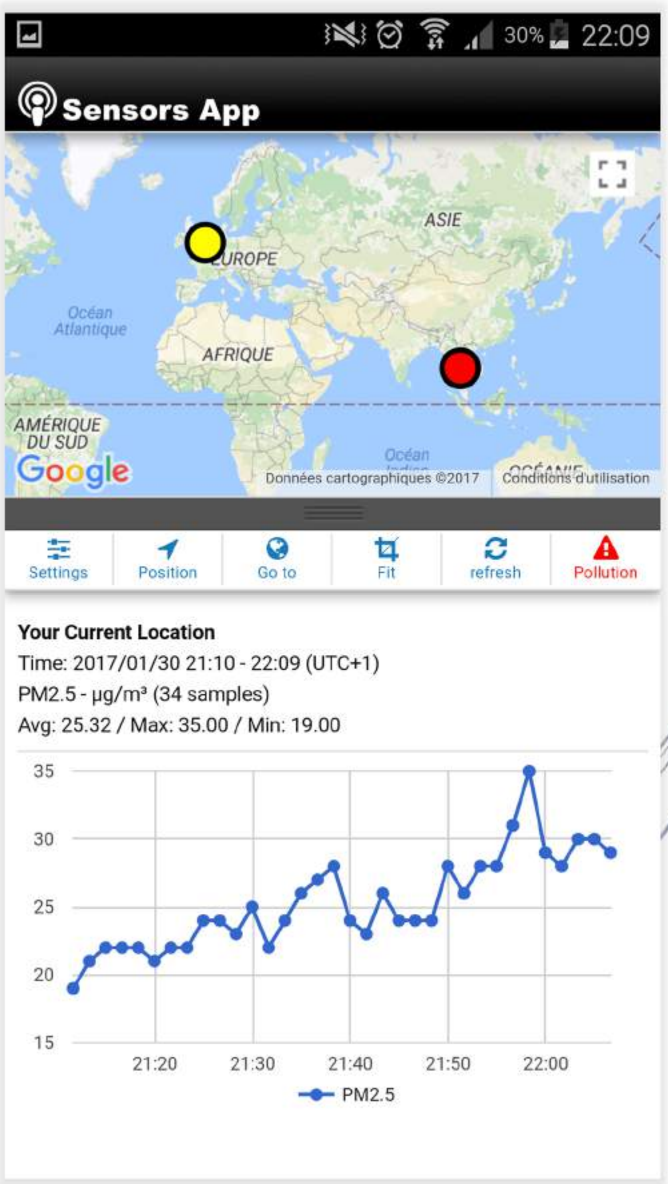
SENSORS APP

SMS Project



Your Current Location

Time: 2017/01/30 21:10 - 22:09 (UTC+1)

PM2.5 - µg/m³ (34 samples)

Avg: 25.32 / Max: 35.00 / Min: 19.00

Realized by:

DIOUBATE Sékou
BELLAL Mohamed El Moustapha
KERKOUCHE Raouf
DJOUDI Aghiles

1. Introduction

University Pierre and Marie Curie of Paris and Macao Polytechnic Institute have developed pollution sensors to predict pollution based on data sent by sensors network implanted in the worldwide (France, Italy, Thailand …). We design this mobile application to operate the sensor network and apply Data Mining & Deep Learning to all sensors data. This application analyze, represent and display the correlations between traffic, speed of pollution travel and exposure.

Our application named **SensorsApp**, operates on mobile devices with Android operating system. Users will be asked to enable GPS on their device to get their current position in order to receive data of the closest sensors.

2. Technical specifications

2.1. Developing application

This application is hybrid and developed using the **Apache Cordova** development tool. The code of our application can be compiled for the whole operating system such as Android, IOS, BlackBerry, Firefox OS, Ubuntu, WP8 (Windows Phone 8), Windows (8.0, 8.1, 10, Phone 8.1) or paciarelli.

Apache Cordova is an open-source mobile development framework. It allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc.

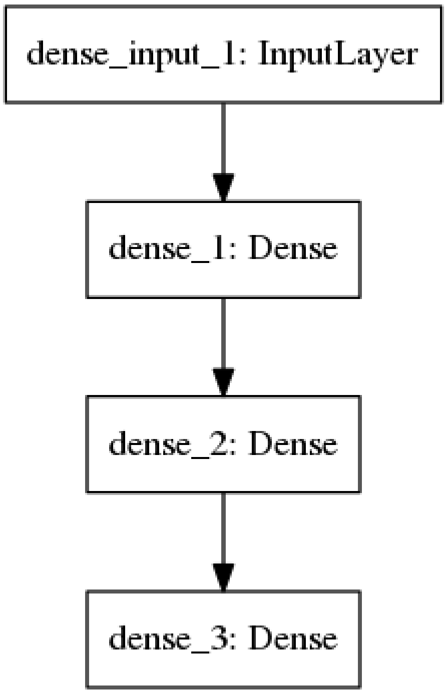
Our application connects to the API available on the server http://52.3.255.145/sensor-map. The information is transmit from the server to the application, through JSON files.

2.2. Data Mining

For the data mining, we used neural networks and the **Keras** library of python.

Keras is a deep Learning library for Theano and TensorFlow. It is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

As shown in Figure bellow, we builded a model of 4 layers. One input layer, 2 hidden layer and one output layer, this model is sequential.



This application predict the pollution of the next day by analyzing temperature, humidity, pressure and pollution of the previous days.

We used **Flask** to develop the learning code that provide our mobile application through a JSON file.

Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions.

3. User Manuel

3.1. Installation

SensorsApp operates on mobile devices with Android operating system. It is compatible with Android 5.0 (Lollipop) or higher version, and for API level 14.

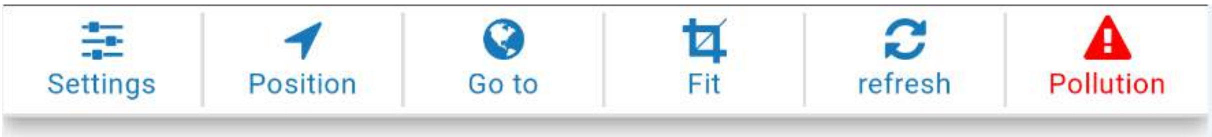
The application requires connection to Internet in order to get data from server's database, and internal GPS receiver to obtain coordinates automatically.

The application is available on play store from the link:
https://play.google.com/store/apps/details?id=fr.upmc.sensors_app

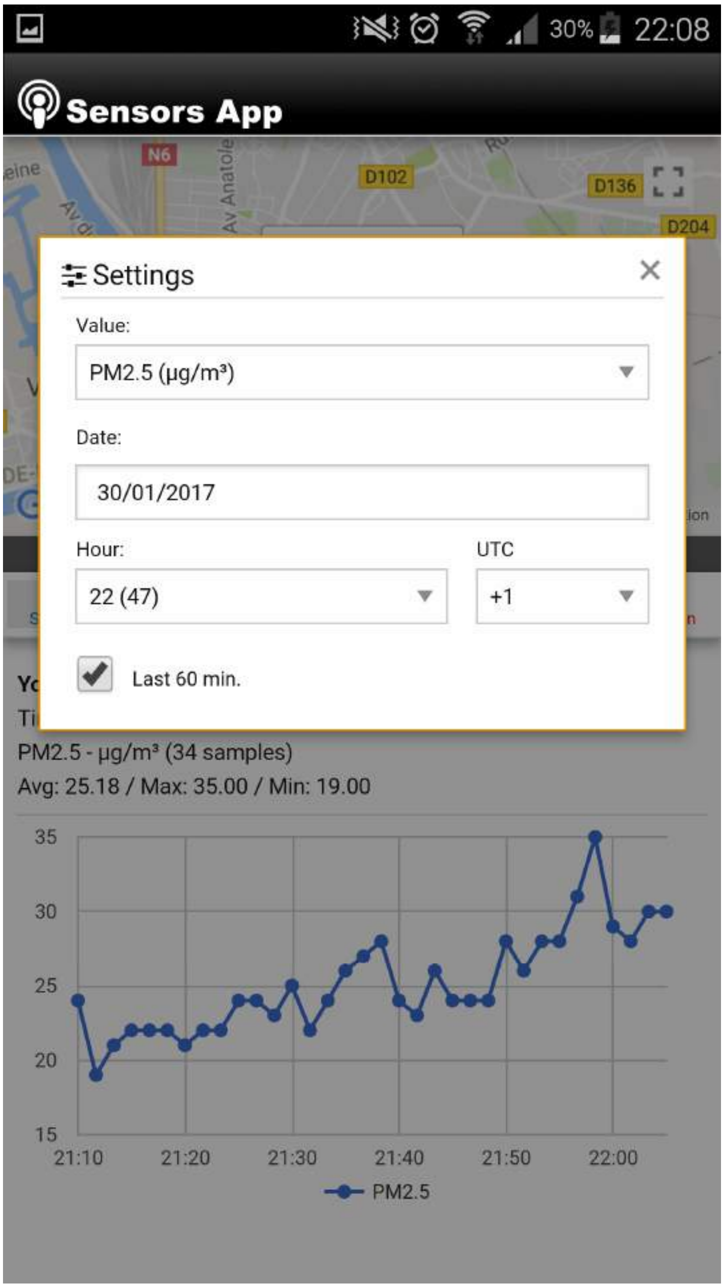
After installation on the device, SensorsApp can be used immediately without any further configuration.

3.2. Getting Started

SensorsApp is tabbed application, which consists of 6 tabs.



- **Settings tab:** The Setting tab allows configure application settings, like time, date, temperature, Air pressure, etc...



Settings

Value: PM2.5 (µg/m³)

Date: 30/01/2017

Hour: 22 (47)

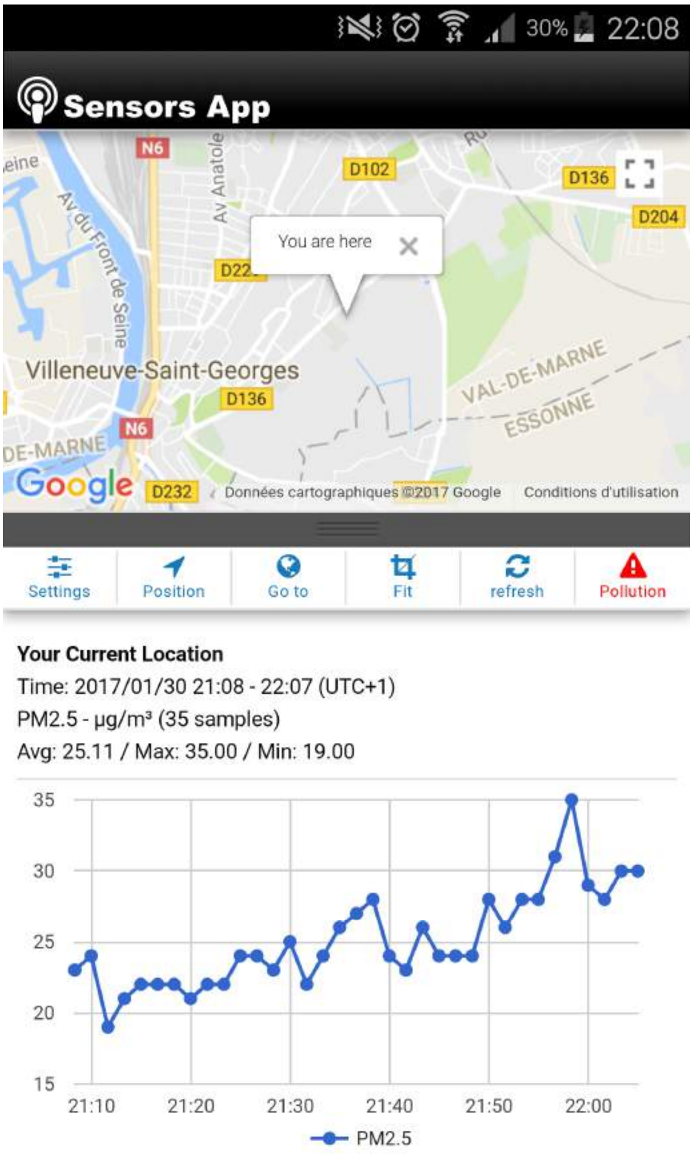
UTC +1

☑ Last 60 min.

PM2.5 - µg/m³ (34 samples)

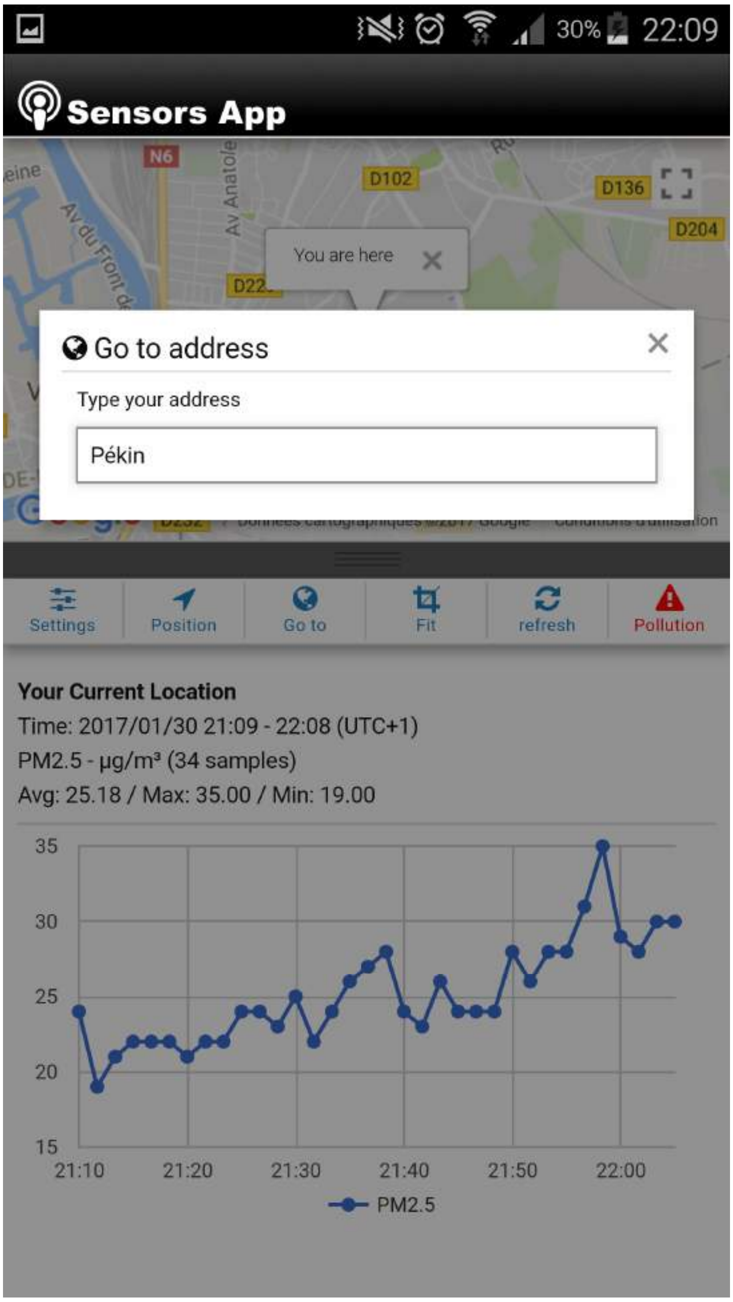
Avg: 25.18 / Max: 35.00 / Min: 19.00

- **Position tab:** Application will take user position and give information about closest active sensor.



Your Current Location
Time: 2017/01/30 21:08 - 22:07 (UTC+1)
PM2.5 - µg/m³ (35 samples)
Avg: 25.11 / Max: 35.00 / Min: 19.00

- **Go to tab:** This functionality allows user to get new position by putting address, and show information of closest sensor to this position.



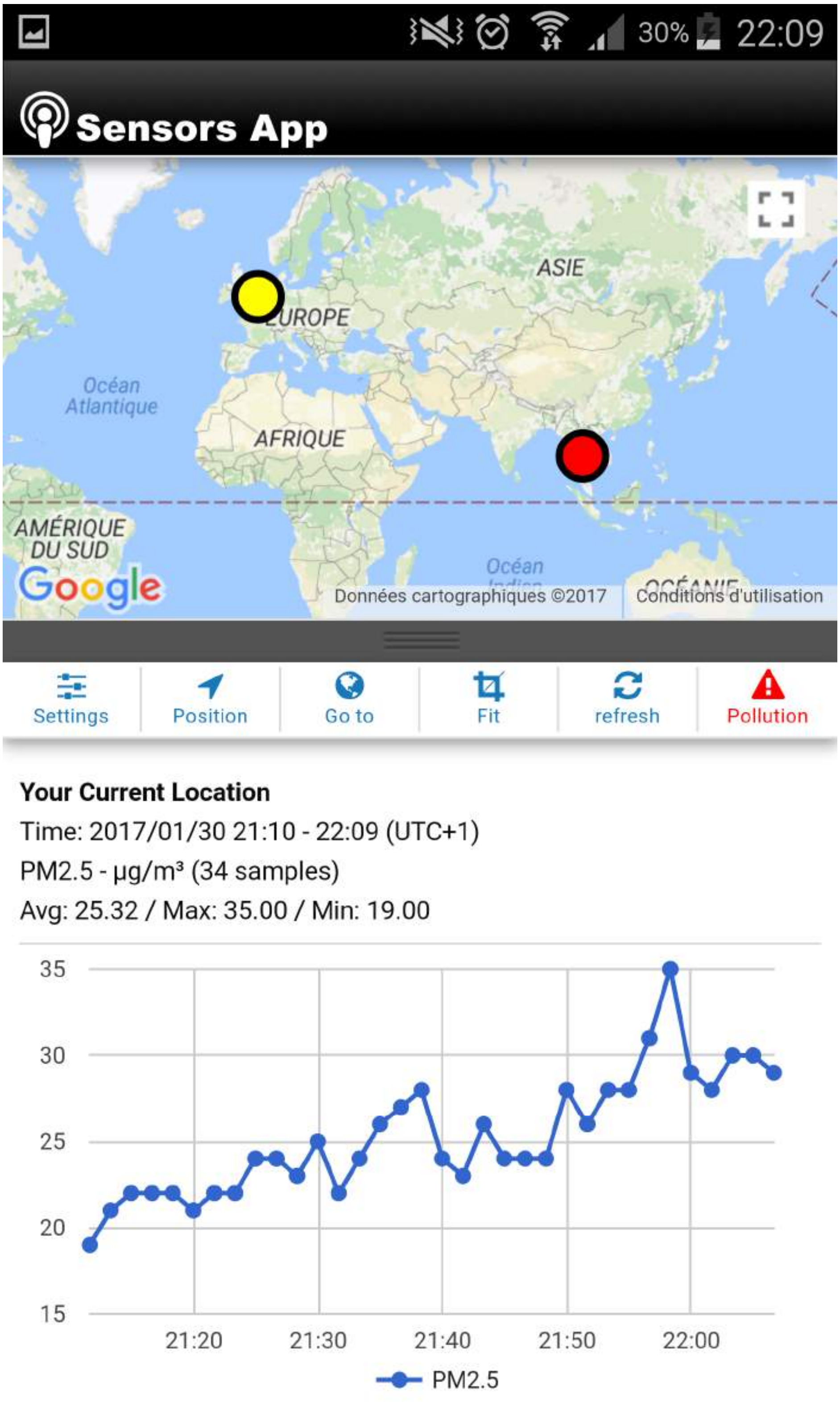
Go to address

Type your address

Pékin

Your Current Location
Time: 2017/01/30 21:09 - 22:08 (UTC+1)
PM2.5 - µg/m³ (34 samples)
Avg: 25.18 / Max: 35.00 / Min: 19.00

- **Fit tab:** The fit button adjusts the map to better describe the sensors on the map.



Your Current Location

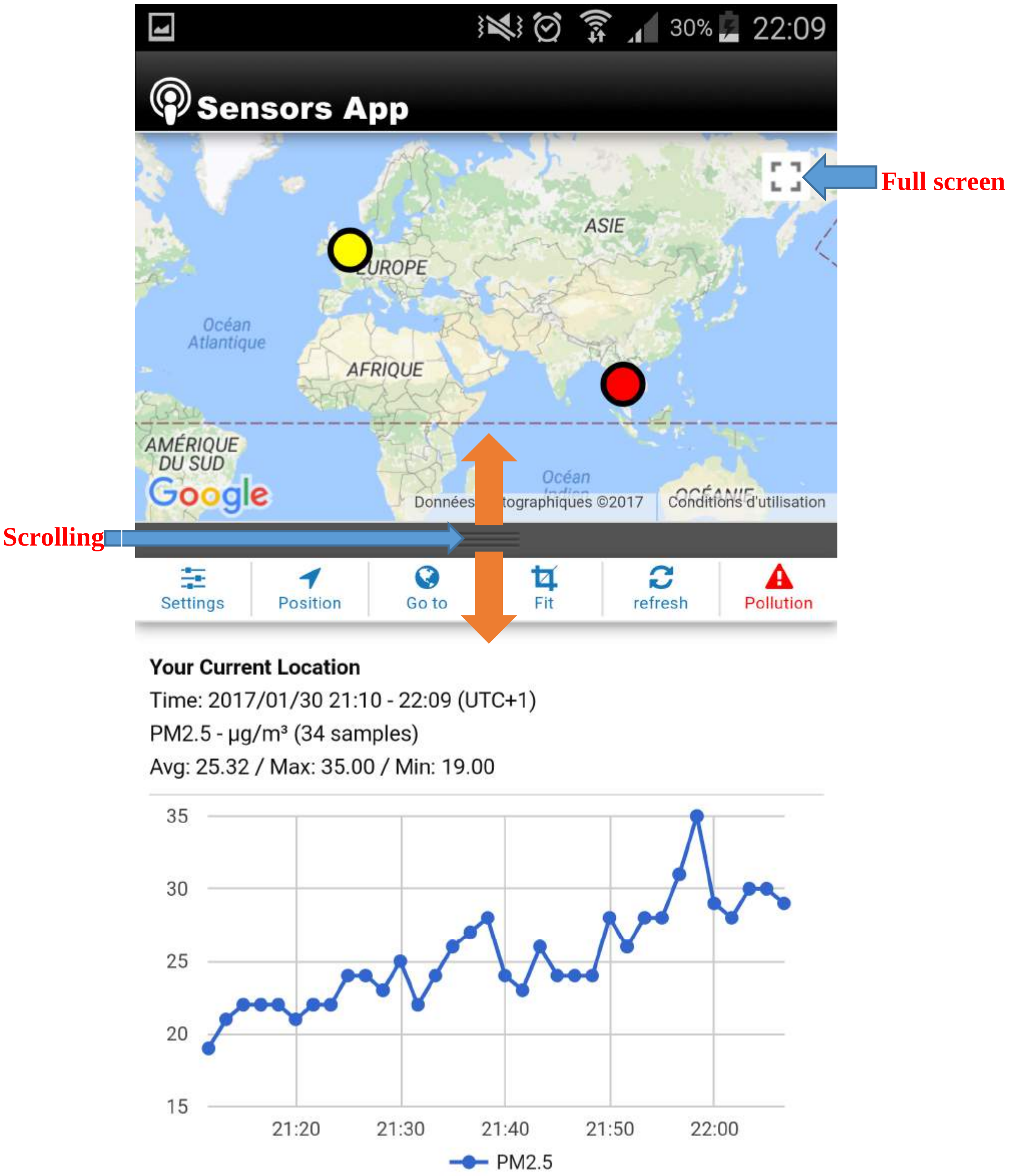
Time: 2017/01/30 21:10 - 22:09 (UTC+1)

PM2.5 - µg/m³ (34 samples)

Avg: 25.32 / Max: 35.00 / Min: 19.00

- **Refresh tab:** This refresh application at the default values.

- **Scrolling and full screen:** The Scrolling allows user to well view the map and sensors information. Full screen allows to get map on full screen.



Your Current Location

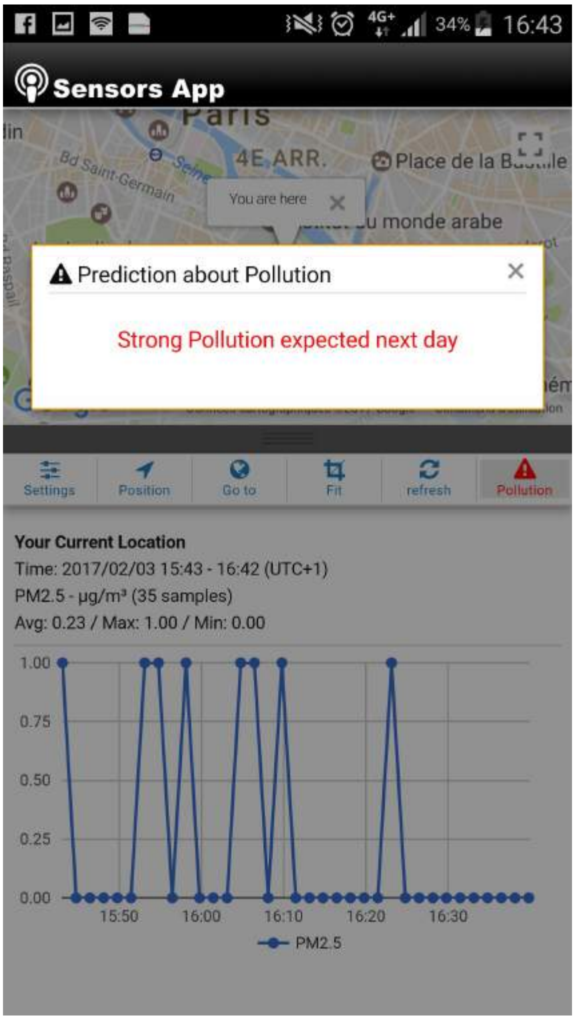
Time: 2017/01/30 21:10 - 22:09 (UTC+1)

PM2.5 - µg/m³ (34 samples)

Avg: 25.32 / Max: 35.00 / Min: 19.00

- **Predicting pollution :** The application will show three cases about the prediction of pollution:

If for next day our model predict that will be pollution in the area of user



Sensors App

Prediction about Pollution

Strong Pollution expected next day

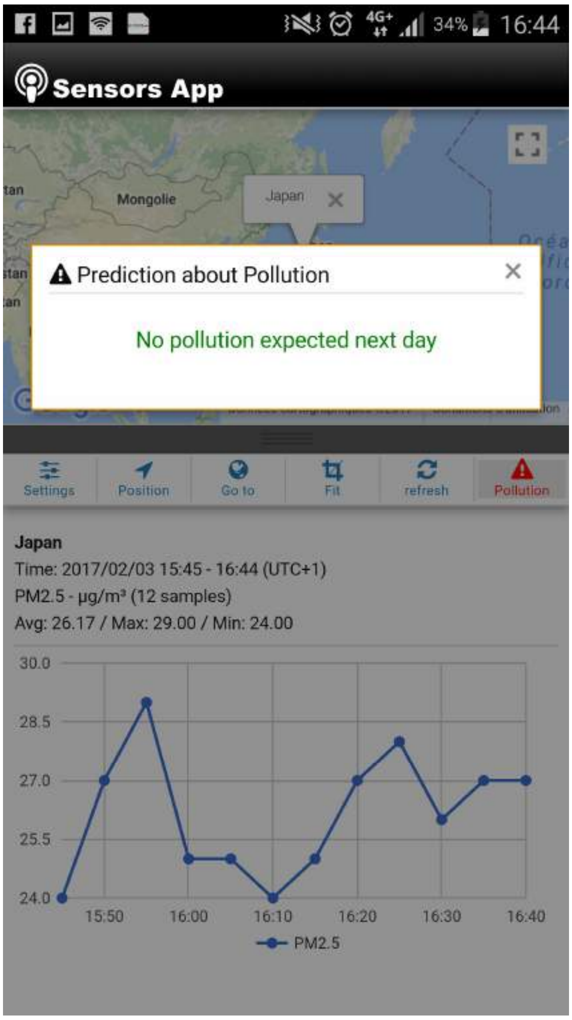
Your Current Location

Time: 2017/02/03 15:43 - 16:42 (UTC+1)

PM2.5 - µg/m³ (35 samples)

Avg: 0.23 / Max: 1.00 / Min: 0.00

If for next day our model predict that will be not pollution in the area of user



Sensors App

Prediction about Pollution

No pollution expected next day

Japan

Time: 2017/02/03 15:45 - 16:44 (UTC+1)

PM2.5 - µg/m³ (12 samples)

Avg: 26.17 / Max: 29.00 / Min: 24.00