**IAS Project – Team 3 - Module-1**

**Index:**

* [**Description**](#Description)
* [**Devices used in Actual implementation**](#Devices_used)
* [**Technologies used**](#Technologies_used)
* [**Data Flow and Data Format**](#Data_flow)
* [**Screen Shots**](#Scree_shots)
* [**Notes**](#notes)

**Description:**

Simulating **Multiple** **Sensors** devices sending data over **Bluetooth** to a nearby **Gateway** device, which has **Internet** capabilities and can send that data to anywhere in the world over a **Socket** connection, given the corresponding IP address and port number.

**Devices used in Actual implementation**:

* Senor devices 🡪 Android mobile.
* Gateway device 🡪 Android mobile.
* Data Receiver server 🡪 Node.js server on laptop.

**Technologies used**:

* Cordova for the Senor and Gateway Mobile Application(**all coding done in java script, as required**)
* Node.js server library on Laptop for Socket connection.

**Data Flow and Data Format:**

* At the boot up of the **Sensor** application, it will list all the available Bluetooth devices within its range, which includes the **Gateway** application.
* After connecting to the Gateway, Sensor will send its **Configuration** details in **JSON** format as below:

**{'ID':'Sensor1','Type':'Temp','Location':'IASClass'}**

Here,

**ID** 🡪 unique for each Sensor device

**Type** 🡪 Indicates Type of Sensor i.e. temperature Sensor

**Location** 🡪 The actual location of the sensor.

* After successfully sending its Configuration details, sensor now starts sending the actual **Data** in **JSON** format as below:

**{'ID':'Sensor1','DateTime:'Tue Apr 14 2015 20:03:21 GMT+0530 (IST)','value':'25'}**

Here,

**ID** 🡪 Unique ID for identifying the sensor device, using which we can get all its details from its previously stored configuration details.

**DateTime** 🡪 The actual time at which the corresponding reading was sensed

**Value** 🡪corresponding sensor reading at that time.

* **Multiple Sensors** will be sending their corresponding reading to the Gateway Application (one at a time).
* The Gateway device (mobile application) is capable to distinguish between Configuration Data or the actual Sensed Data sent to it from these multiple sensors, and forwards it to the Repository Server or Filter Server respectively over the internet using socket connection.

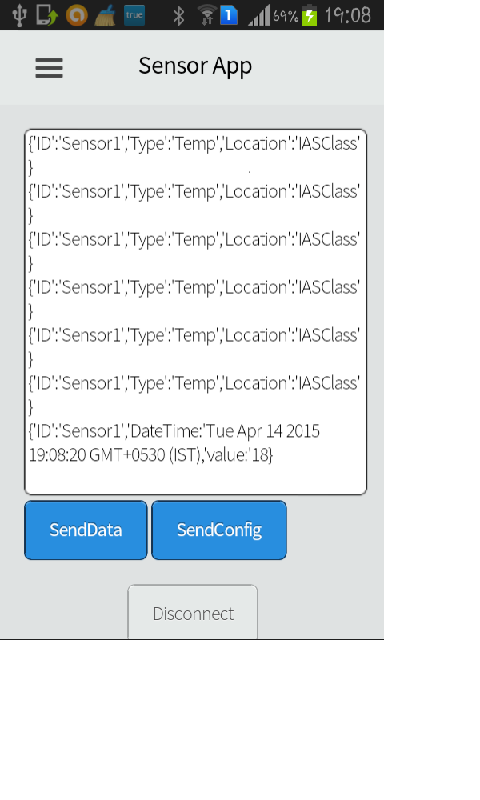
(Of course the corresponding IP address and port numbers of both these servers are required as an input to this module).

**Few Screen Shots:**

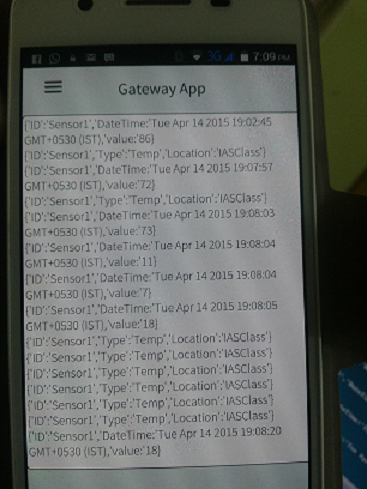
* Sensor application at boot up, detecting all nearby Bluetooth devices

****

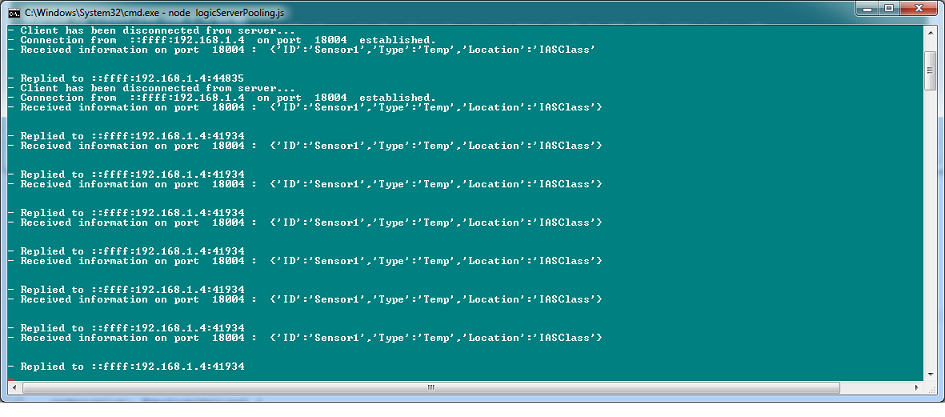
* The Sensor after connecting to the Gateway Application, sending the configuration and Data, all this can be viewed on the mobile device



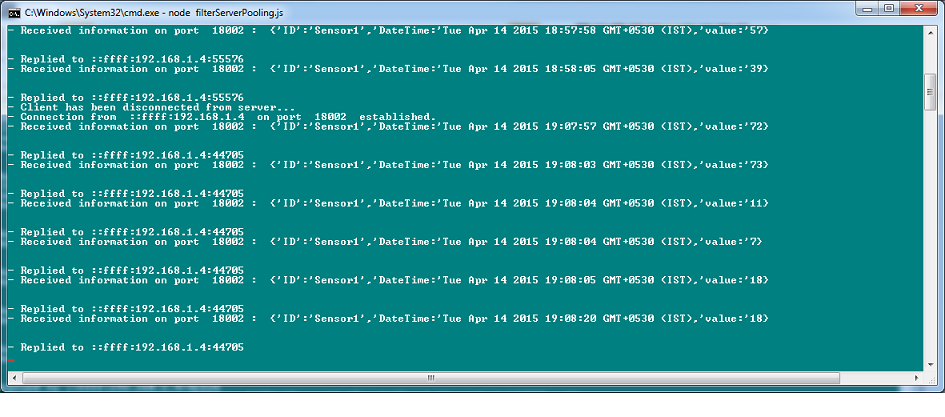
* The Gateway application showing all the Data/Configuration it is receiving from all the sensors



* The Gateway application identifying and sending all the **Sensor configuration** data to the **repository server**.



* The Gateway application identifying and sending all the **Sensed data** from multiple Sensors to the **filter** server.



**Note:**

* **Multiple Sensors** can be registered at any given time and can data to the Gateway.
* **BLE** was suggested for communication between Sensor and Gateway, but only latest mobile devices have this capability, due to lack to availability of devices we have used simple **Bluetooth** communication.
* **Bluetooth** is one to one communication i.e. only one sensor can be connected to the Gateway and can send data at a given time.

But since we had to simulate multiple sensors, we implemented a work around, as soon as the sensor is done sending the data, it automatically disconnect, allowing other sensors to connect to Gateway.