universidade de aveiro deti departamento de eletrónica, telecomunicações e informática

IoT City: Visualizing and Interacting with the city through an IoT Platform

Milestone 1

1 May, 2017

Project in Informatics Engineering

Aveiro University

Team Members	
Ana Patrícia Gomes da Cruz	*76351*
Diogo Daniel Soares Ferreira	*76504*
Inês Filipa Mendonça da Cruz	*72725*
José Pedro Baião Castanheira	*76545*
Luís Davide Jesus Leira	*76514*
Miguel Ângelo Pereira da Silva	*76450*

Mentor
Ilídio Oliveira

Collaborators
Susana Sargento
Filipe Cabral Pinto

Index

Context

In the modern world, the *Smart Cities* concept is becoming more popular every day and telecommunications operators have very high expectations on the future market associated with this concept. Also, the *Internet of Things* (IoT), a concept closely related with *Smart Cities*, is similarly a paradigm with growing interest as time goes by.

There are already some cities around the world with well-established systems that interact with the city environment in a digital way through sensors and actuators.

Currently it is already possible in some cities to see and analyze the data of the city environment and to understand better its' dynamic and its habitants' movements, through several types of sensors like luminosity sensors, waste management, air pollution sensors and many other metrics.

Altice Labs, our industrial collaborator in this project, and IT (Instituto de Telecomunicações) provided us a wide range of sensors to work with, and proposed that the main goal of this project would be to gather and analyze data from several sensors and build a web application that makes use of that information to give feedback about their metrics to the city managers, as well as providing an interface to interact with the sensors in a secure and safe way. Besides the web portal, and considering that a very significant part of the population uses smartphone, we also intend to build a mobile application that displays the city information for the regular end-user, and an administrator portal to be easier to add sensors and other information to the platform.

Figure 1 - Drawing that represents the Internet of Things. Includes connected objects like drones, plants, wearables, houses and many more. Source: flickr.com/wilgengebroed

Introduction to the Report

This milestone is currently on the version 1.5.

On this first report, our main goal is to define the scope of our project and to establish common ground with the client/stakeholders regarding success criteria.

The first section, ***IoT City project – Vision***, has the aim of explaining the project on a high-level basis. It explains the solution adopted, with focus on the information flow and the user applications.

The third section, ***Use Cases***, describes the identified required interactions between the users and the system.

The fourth section, ***Technical Requirements***, clarifies the requirements for this project, including the hardware needs, the software needs for our platform and other non-functional requirements.

The fifth section, ***Overall Architecture***, gives insight into how we plan to achieve the goals, and differentiates between the software modules already existent from the software modules that we must build from scratch, describing their main functions.

The sixth section, ***Main Features***, explains the main features that we plan to present in both our applications, web portal and mobile application.

The seventh section, ***Usage and Screenshots***, illustrates and explains the end-user interface, and how the applications (Web Portal, Mobile Application and Administrator Portal) will look, with screenshots to help understand the main features.

The eighth section, ***Project Timeline***, maps directly the features and important events described earlier into a timeline, explaining what is scheduled to be completed on each milestone.

The ninth section addresses the possible ***risks and issues*** on building the solution, and suggests preventive and corrective measures on how to minimize their impact on the overall project.

The last section, **Report Versions**, presents the major changes made to this report, with the date and the version number.

The project IoT City – Vision

The main goal of the project is to provide the city manager a web application easy to learn and to use for gathering city metrics and act upon them. Other goals are to provide the city users a mobile application that allows them to track in real-time the state of the city, and to provide the city administrator a platform where he can manage the sensors in the platform, and other information.

To aggregate city metrics, we will have sensors that measure illumination conditions, air quality, UV radiation, waste management, temperature, sound and people counter. The data collected by these sensors will show both the city manager and the regular user information about the city current state and history. It will also be available illumination actuators, for the city manager to turn on or off public illumination Not all users will have access to the same data. For example, a city local government should not have access information from another city.

The project has a wide range of scenarios to aim for. It can help to create optimized routes for waste collectors, or to determine which parts of the city should have better lighting, or to know the levels of pollution on each street, during the day. With the actuators, the city manager can turn off the public illumination when it gets dark. It is also possible to correlate different city metrics during a certain period and research for ways to make the city a better place.

There are already some platforms that provide information from city metrics to city managers and regular users, such as "Array of Things" (Chicago, USA) and “IoTsense” (Castellón, Spain). The main advantage of *IoT City* is to be able to congregate the information available and allow the city manager to define thresholds and alarms and act immediately. Other advantages are the statistics stored in the system, that can be used to do analytics in the future.

Use cases

On this section, we expose the use cases gathered, reflecting all system activities that have significance to the users. All use cases are mapped with and identification code.

Web Portal

- Web_UC1 – Visualize sensors' streams values in real-time;
- Web_UC2 – Visualize sensors' stream history;
- Web_UC3 – Visualize sensors' location on a map;
- Web_UC4 – Send values to actuator;
- Web_UC5 – Create an alert;
- Web_UC6 – Visualize User Reports;
- Web_UC7 – Create a Note.

Mobile Application

- App_UC1 –Visualize sensors' streams values in real-time;
- App_UC2 – Visualize sensors' location on a map;
- App_UC3 – Send report to city manager.

Administrator Portal

- AP_UC1 – Add Sensor;
- AP_UC2 – Add Stream;
- AP_UC3 – Add stream to sensor;
- AP_UC4 – Create subscription group;
- AP_UC5 – Add sensor to subscription group;
- AP_UC6 – Add user to subscription group.

Technician

- T_UC1 – Add sensor through an API;
- T_UC2 – Add stream through an API.

Technical Requirements

On this section, we explore the technical requirements identified for our platform to work as intended. All the requirements are mapped with an identification code.

- The platform needs at least one server with public access to be provided by Altice Labs or IT;
- The Web Portal should require user authentication;
- The Admin Portal should require user authentication;
- Passwords should not be stored in the database, or should be encrypted;
- The web application must be fully functional in Google Chrome (version 41 or superior), Firefox (version 14 or superior) and Safari (version 7 or superior) and Edge (version 25 or superior) browsers;
- The mobile application should be compatible with Android 4.0 or superior;
- The sensor aggregators should be able to run python 2.x;
- The IoT City Web Server must have at least: 1 Gb of Ram and 1 Gb of Disk Drive;
- The IoT City Web server should have a public IP, and be addressable in port 80;
- The system should handle, at least, 100 online users simultaneously;
- The system should handle, at least, 100 mobile users simultaneously;
- The system should handle, at least, 50 sensors of any type simultaneously;
- When an alarm defined by the user is triggered, if an actuator is associated, the IoT Platform should trigger the actuator in less than one minute;
- The mobile app should be self-adjustable to the screen size (mobile and tablet regular screens);
- The notes and alerts defined by each user should not appear for another user rather that the creator of the note or alert;
- A user can only see the streams associated to the allowed groups;
- A user can only see the sensors that have streams associated to the allowed groups;
- The types and subtypes of sensors should be: temperature, lighting (illumination values and actuator), air (pressure and CO2), waste containers (fullness percentage, internal temperature and volume), noise, radiation (UV Radiation, Visible Radiation and Infrared Radiation) and people counter;
- Sensors can have static or dynamic location.

Overall Architecture

This section briefly describes the planned overall architecture of our solution. Our architecture is divided into 3 layers: (1) Distributed data sources (Data Generation Layer); (2) IoT Platform (Business Layer) and (3) Applications (Presentation Layer).

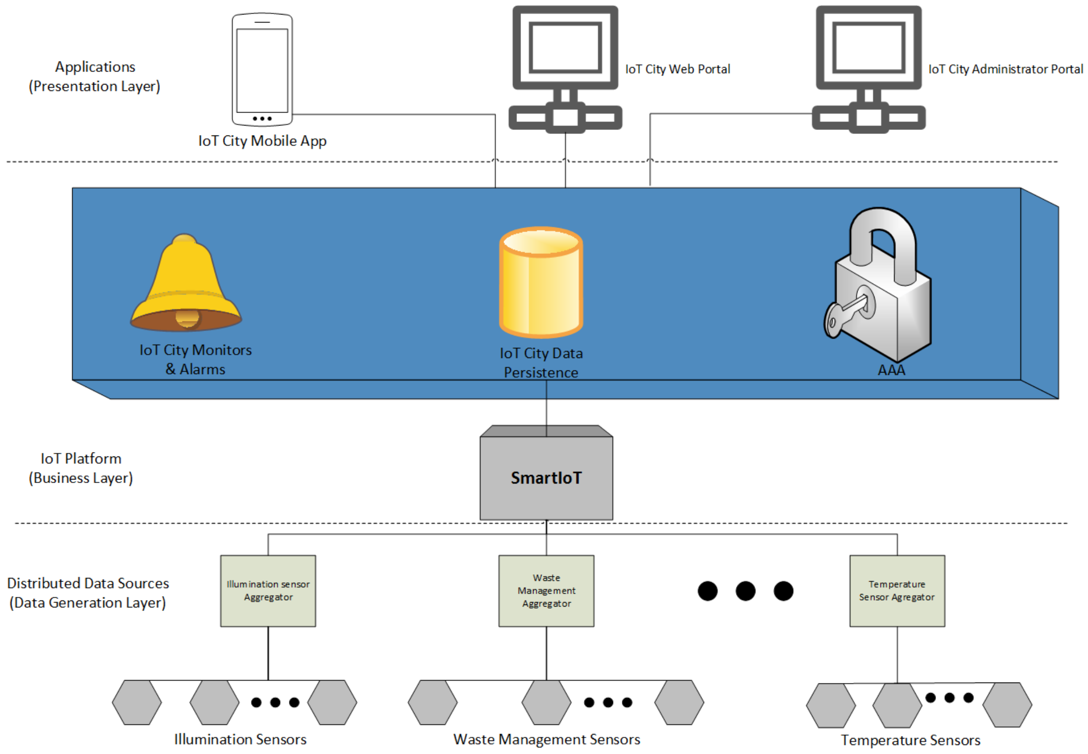


Figure 1 - The architecture of our solution. It has three layers: Distributed data sources (Data Generation Layer), IoT Platform (Business Layer) and Applications (Presentation Layer).

On the “Distributed data sources" layer, we have a set of sensors for each type of data (lighting, waste, temperature...). These sensors will collect data from the environment and send to an aggregator. The aggregators will send the information to a gateway, which will be the point of contact with the broker in the “IoT Platform”.

On the “IoT Platform”, the broker will receive the data from all the aggregators. The broker already exists in Altice Labs, and goes by the name of SmartIoT. SmartIoT is a platform that collects the data from the Altice Labs sensors that will be used in our project. The "AAA" module, where AAA stands for "Authentication, Authorization and Accounting" will be responsible for authenticate registered users. The "IoT City Monitors & Alarms" module will allow the user to add alerts to the data streams, with optional actuators defined. If any threshold is crossed, this module will notify the user and act accordingly. The module "IoT City Data Persistence" will store all the data from the IoT Platform. In some cases, it is also possible for the "IoT City Web App" to send information for the sensors (actuators). On that case, the information is sent to the SmartIoT broker, and then is sent to the sensor (actuator).

On the "Applications" layer we have the web portal, the administrator portal and the mobile application. The web portal is the interface for the city manager to see the current state of the city. The web portal uses the AAA module to authenticate its users, and receives the data from the IoT City Data Persistence. It also can add, change or delete alarms, notes and chage the state of user reports. The Administrator portal has the capability of adding sensors, streams, users, groups of users, and to do other changes on the IoT City Data Persistence Module. The IoT City Mobile Application requires no authentication, and it communicates with the IoT City Data Persistence to gather the values from sensors and create user reports.

Main Features

On this section, we explore the main features of our applications (Web Portal and Mobile Application). All the features are mapped with an identification code.

Web Portal

- WP_F1 - Check all the sensors' streams values in real-time;
- WP_F2 - Check the history of sensors' streams;
- WP_F3 - Check the location of all the sensors allowed to the user on a map;
- WP_F4 - Turn on/off public lighting and send values to actuators;
- WP_F5 - Add alerts with actuators to sensors' streams;
- WP_F6 - Check the users reports;
- WP_F7 - Add notes associated to the user.

Mobile Application

- MA_F1 - Check the sensors' streams values in real-time and history;
- MA_F2 - Check the location of the sensors in real-time;
- MA_F3 - Send reports of problems to the city manager.

Administrator Portal

- AP_F1 - Add sensors to the platform;
- AP_F2 - Add streams to the sensors previously added;
- AP_F3 - Add streams to subscription groups;
- AP_F4 - Add users to subscription groups;
- AP_F5 - Create, edit and delete data on the IoT City Data Persistence (database).

Technician

- T-F1 - API to add sensors dynamically;
- T-F2 - API to add streams dynamically;
- T-F3 - Dynamic location of sensors.

Usage and Screenshots

Web portal

The first view of the web portal is the authentication page. If the user is not authenticated, it doesn't have the permission to see any information of the metrics gathered. If the username and password entered by the user match with a valid account, the webpage will redirect the user to the main dashboard page. Each account can have streams associated. A stream is a flow of data send by the sensors (for example: pressure of air sensor x, or waste container percentage of waste sensor y). If an account doesn't have any streams, it will not be possible to see any information of the sensors.

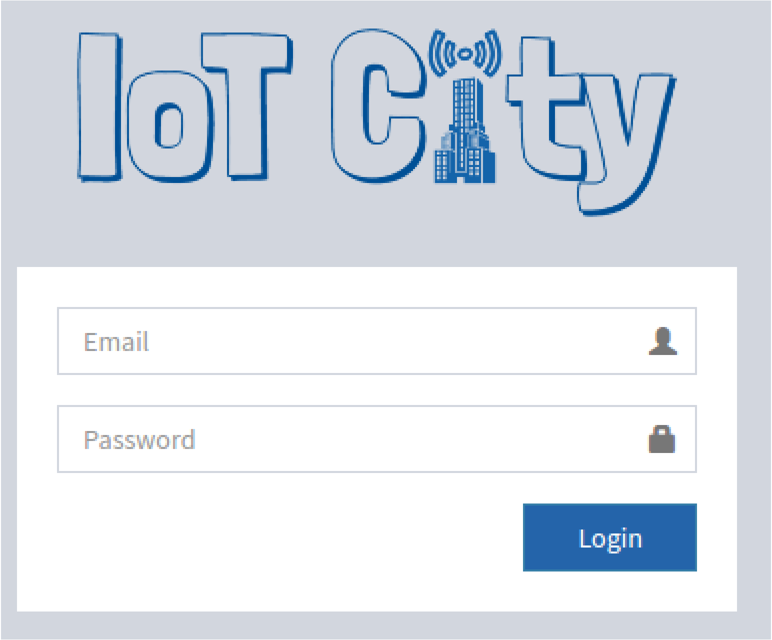


Figure 2 - Authentication View

The main page of the web portal contains a dashboard with information about the type of metrics gathered by the sensors (temperature/lighting/air/waste/noise/radiation/people). For each type of metric, there is a card with information about the last values retrieved by the sensors.

- For the temperature sensors, will be displayed the average temperature of all temperature sensors allowed to the user, as well as the maximum and minimum value.
- For the lighting sensors, will be displayed the minimum value of lighting received by the sensors, as well as its location.
- For the air sensors, will be displayed the average carbon dioxide values, as well as the maximum and minimum values.

- For the waste containers sensors, it will be displayed the number of containers that have a waste volume higher than 75%.
- For the noise sensors, it will be displayed the maximum noise value detected, as well as the average for the allowed streams
- For the radiation sensors, it will be displayed the number of sensors with UV Radiation index above 7. It will also be displayed the number of sensors with UV Radiation index between 4 and 7.
- For the people counter sensors, it will be displayed the maximum percentage of location of a place, alongside the name of the place.

For all the cards, it will also be shown the number of active and inactive streams and the date from last reading.

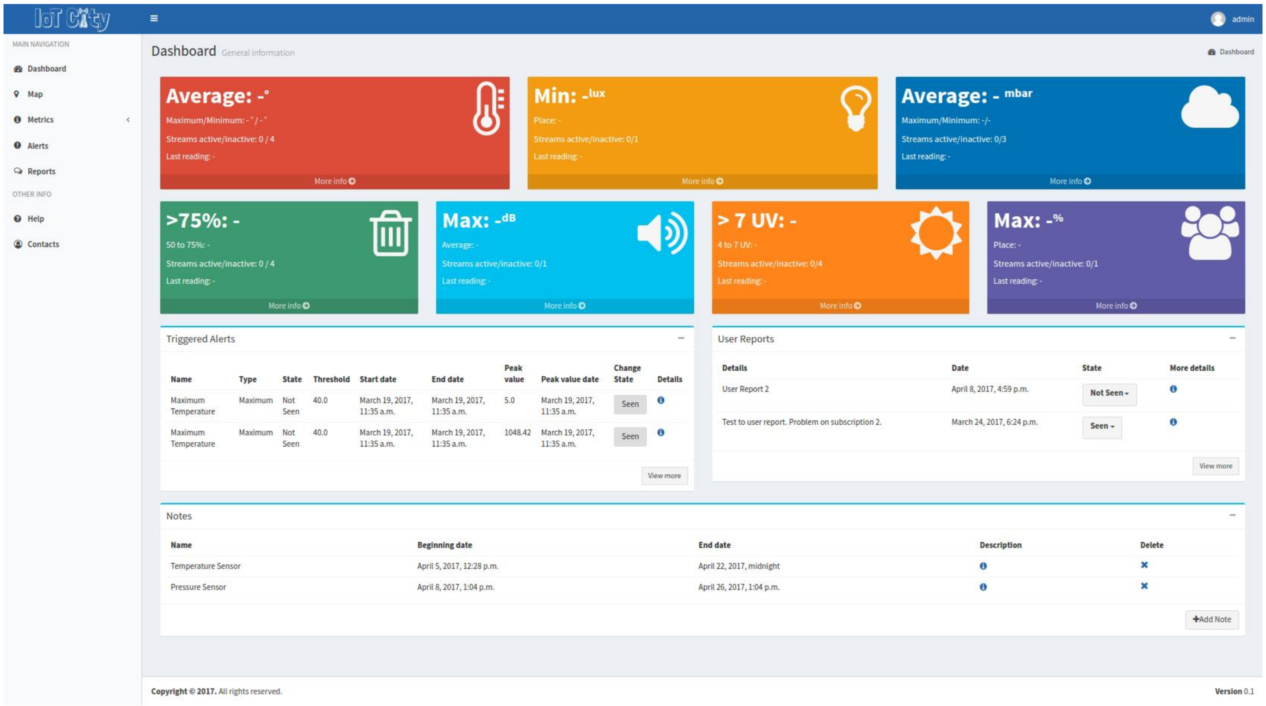
Bellow the cards, there is a list of all the triggered alerts, alongside with its information. An alert can be added by the user on the sensor metrics page. An alert is a maximum or minimum threshold for several subscriptions, that can be turned on or off according to a schedule. It is also possible to act when an alarm is triggered (e.g. when it is 7 p.m., turn all public lighting), sending a message to all the actuators.

The triggered alerts have three states: active, not seen and seen. If an alert is active, it means that the sensor values are currently crossing a threshold. After the active state, the alert goes on to the not seen state. It is possible to change its state in the button seen. After the alert is marked as seen, he will no longer be in the "Triggered alerts" table.

The "User reports" table shows the information about the reports sent by the citizens through the mobile app. As with the triggered alerts, it is also possible to change its state, being the possible values not seen, seen, working on it and solved. When the state is changed to solved, it will no longer be shown on the table.

Finally, there is also a table with notes added by the user. If the user wants to add a note, he can click on "Add note" and fill the popup form with the adequate information.

On the left sidebar, we can access different pages of the web portal.



Dashboard General information

Average: -°
Maximum/Minimum: - ° / - °
Streams active/inactive: 0 / 4
Last reading: -

Min: -lux
Place: -
Streams active/inactive: 0/1
Last reading: -

Average: - mbar
Maximum/Minimum: -/-
Streams active/inactive: 0/3
Last reading: -

>75%: -
50 to 75%: -
Streams active/inactive: 0 / 4
Last reading: -

Max: -dB
Average: -
Streams active/inactive: 0/1
Last reading: -

> 7 UV: -
4 to 7 UV: -
Streams active/inactive: 0/4
Last reading: -

Max: -%
Place: -
Streams active/inactive: 0/1
Last reading: -

Triggered Alerts

Name	Type	State	Threshold	Start date	End date	Peak value	Peak value date	Change State	Details
Maximum Temperature	Maximum	Not Seen	40.0	March 19, 2017, 11:35 a.m.	March 19, 2017, 11:35 a.m.	5.0	March 19, 2017, 11:35 a.m.	Seen	🛈
Maximum Temperature	Maximum	Not Seen	40.0	March 19, 2017, 11:35 a.m.	March 19, 2017, 11:35 a.m.	1048.42	March 19, 2017, 11:35 a.m.	Seen	🛈

User Reports

Details	Date	State	More details
User Report 2	April 8, 2017, 4:59 p.m.	Not Seen ▾	🛈
Test to user report. Problem on subscription 2.	March 24, 2017, 6:24 p.m.	Seen ▾	🛈

Notes

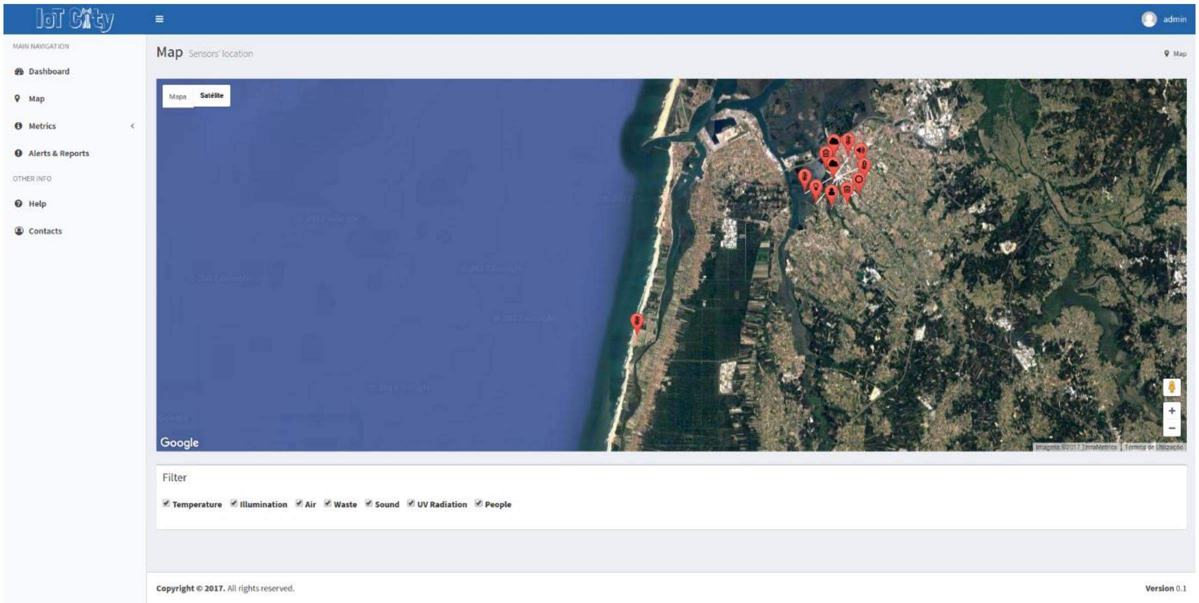
Name	Beginning date	End date	Description	Delete
Temperature Sensor	April 5, 2017, 12:28 p.m.	April 22, 2017, midnight	🛈	✖
Pressure Sensor	April 8, 2017, 1:04 p.m.	April 26, 2017, 1:04 p.m.	🛈	✖

Copyright © 2017. All rights reserved.

Version 0.1

Figure 3 - Main Dashboard Page

The map view contains a map with the location of all the sensors allowed to that user, with a filter to select the sensor types that are visible on the map. When the user clicks on top of a sensor, it will appear a popup box with information about that sensor and a toggle switch to turn on or off the streams of that sensor (the toggle will NOT be used to turn on or off the sensor, but to turn on or off the retrieval of data from streams of that sensor). If several sensors are too close to use separate markers only one will appear and clicking on it will expand all the sensors' markers.



Map Sensors' location

Filter

☑ Temperature ☑ Illumination ☑ Air ☑ Waste ☑ Sound ☑ UV Radiation ☑ People

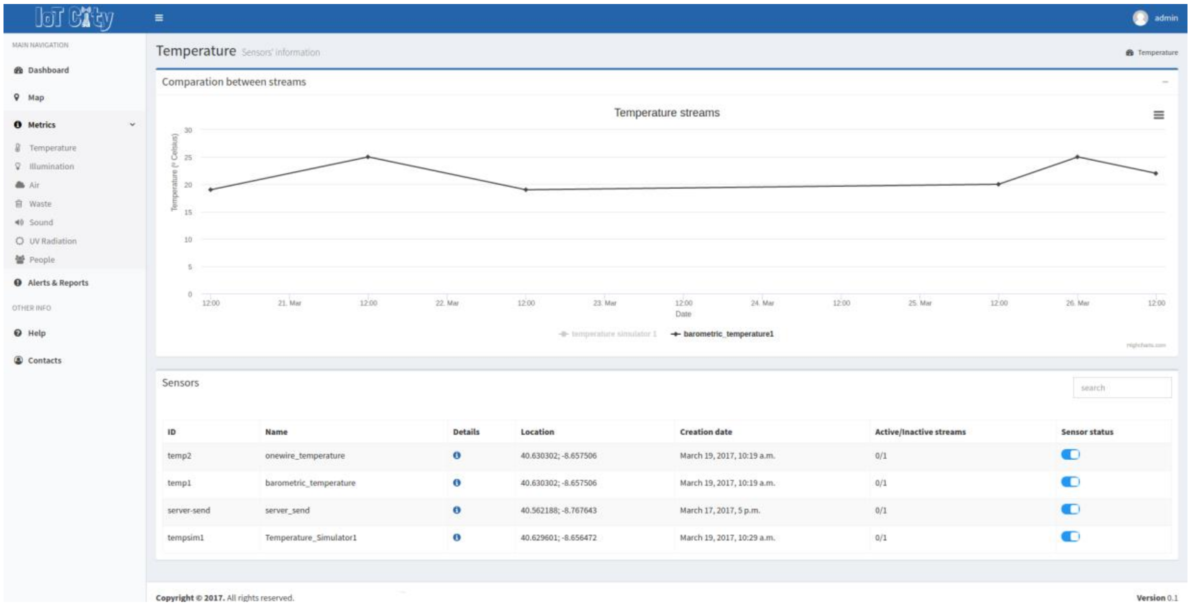
Copyright © 2017. All rights reserved.

Version 0.1

Figure 4 - Map with markers that shows the allowed sensors for that user

There is a view for each one of the seven types of sensors. That view will have a chart with the values of the all the sensors of that type. In case of the sensors have more than one subtype (e.g. waste sensors can have streams of container percentage, internal temperature and volume), each subtype will have a chart (separated by tabs). It is possible to zoom in on a date on the chart, and filter by the subscriptions wanted.

Below there is a table with all the sensors of that type, and associated information. There is also a toggle switch to turn on or off the streams of that sensor (the toggle will NOT be used to turn on or off the sensor, but to turn on or off the retrieval of data from streams of that sensor).

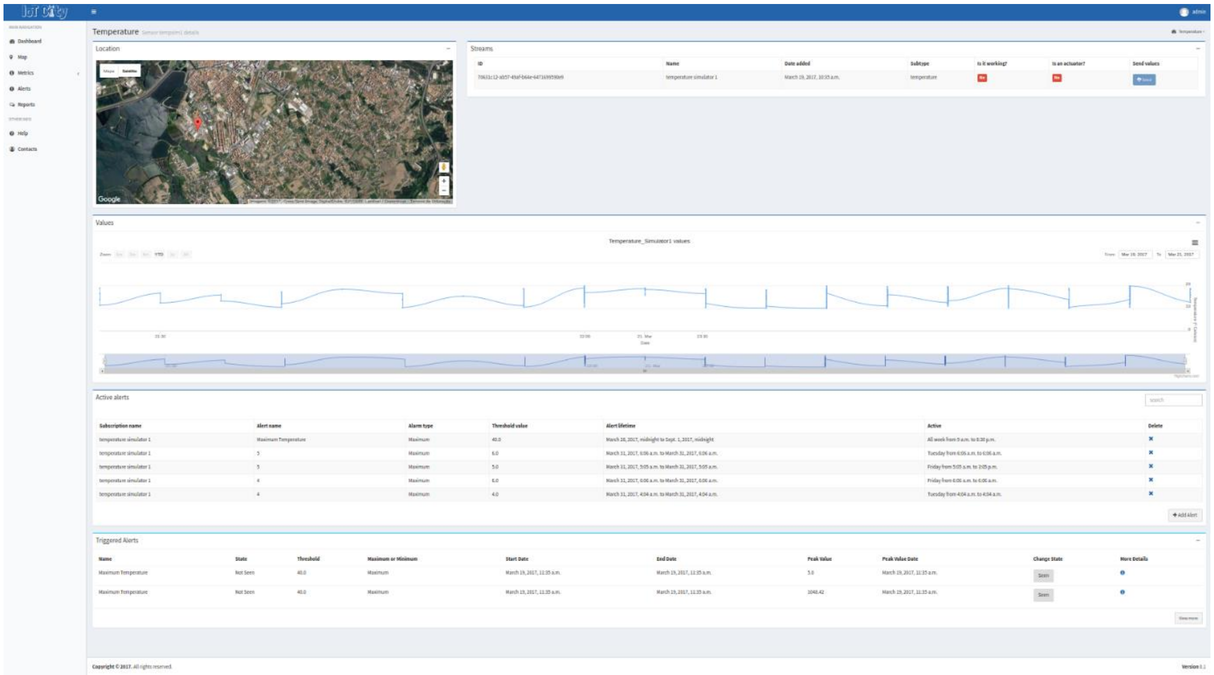


ID	Name	Details	Location	Creation date	Active/Inactive streams	Sensor status
temp2	onewire_temperature	ℹ	40.630302; -8.657506	March 19, 2017, 10:19 a.m.	0/1	●
temp1	barometric_temperature	ℹ	40.630302; -8.657506	March 19, 2017, 10:19 a.m.	0/1	●
server-send	server_send	ℹ	40.562188; -8.767643	March 17, 2017, 5 p.m.	0/1	●
tempsim1	Temperature_Simulator1	ℹ	40.629601; -8.656472	March 19, 2017, 10:29 a.m.	0/1	●

Figure 5 - Temperature Page

There is a page dedicated to every sensor associated. That page has a map with the location of the sensor, as well as a table with all the streams associated to that sensor, with the information associated, an indicator that shows if the stream is working, and other indicator that tells if the stream is an actuator (if it is possible to send data for that stream). If the stream is an actuator, it is possible to send data, by clicking on the "send" button on the "Send Value" column. After that, a popup is open, where the user can insert a numeric value to send to the stream.

Below the map and table, we have a chart with the values of the streams associated to the sensor. There is also a table with all the alerts and the associated information. It is possible to create alerts, by clicking on the Button "Add Alert", and filling the popup form. Finally, we can also see the triggered alerts, alongside with its information.



Streams

ID	Name	Date added	Subtype	Is it working?	Is an actuator?	Send values
70631c12-ab57-49af-b64e-6471699590e9	temperature simulator 1	March 19, 2017, 10:35 a.m.	temperature	No	No	Send

Active alerts

Subscription name	Alert name	Alarm type	Threshold value	Alert lifetime	Active	Delete
temperature simulator 1	Maximum Temperature	Maximum	40.0	March 28, 2017, midnight to Sept. 1, 2017, midnight	All week from 9 a.m. to 8:30 p.m.	✖
temperature simulator 1	5	Maximum	6.0	March 31, 2017, 6:06 a.m. to March 31, 2017, 6:06 a.m.	Tuesday from 6:06 a.m. to 6:06 a.m.	✖
temperature simulator 1	5	Maximum	5.0	March 31, 2017, 5:05 a.m. to March 31, 2017, 5:05 a.m.	Friday from 5:05 a.m. to 2:05 p.m.	✖
temperature simulator 1	4	Maximum	6.0	March 31, 2017, 6:06 a.m. to March 31, 2017, 6:06 a.m.	Friday from 6:06 a.m. to 6:06 a.m.	✖
temperature simulator 1	4	Maximum	4.0	March 31, 2017, 4:04 a.m. to March 31, 2017, 4:04 a.m.	Tuesday from 4:04 a.m. to 4:04 a.m.	✖

+ Add Alert

Triggered Alerts

Name	State	Threshold	Maximum or Minimum	Start Date	End Date	Peak Value	Peak Value Date	Change State	More Details
Maximum Temperature	Not Seen	40.0	Maximum	March 19, 2017, 11:35 a.m.	March 19, 2017, 11:35 a.m.	5.0	March 19, 2017, 11:35 a.m.	Seen	●
Maximum Temperature	Not Seen	40.0	Maximum	March 19, 2017, 11:35 a.m.	March 19, 2017, 11:35 a.m.	1048.42	March 19, 2017, 11:35 a.m.	Seen	●

Copyright © 2017. All rights reserved.

Figure 6 - Sensor Page

The Alerts & Reports view is still under construction.

Admin Portal

The users, marked as "Staff Users", have access to the admin portal, where they can add, change or remove users, sensors, subscriptions, and have access to all the data saved on the IoT City Data Persistence. On this section, we are going to exemplify the process of adding a sensor, subscriptions and associate the subscriptions to specific users.

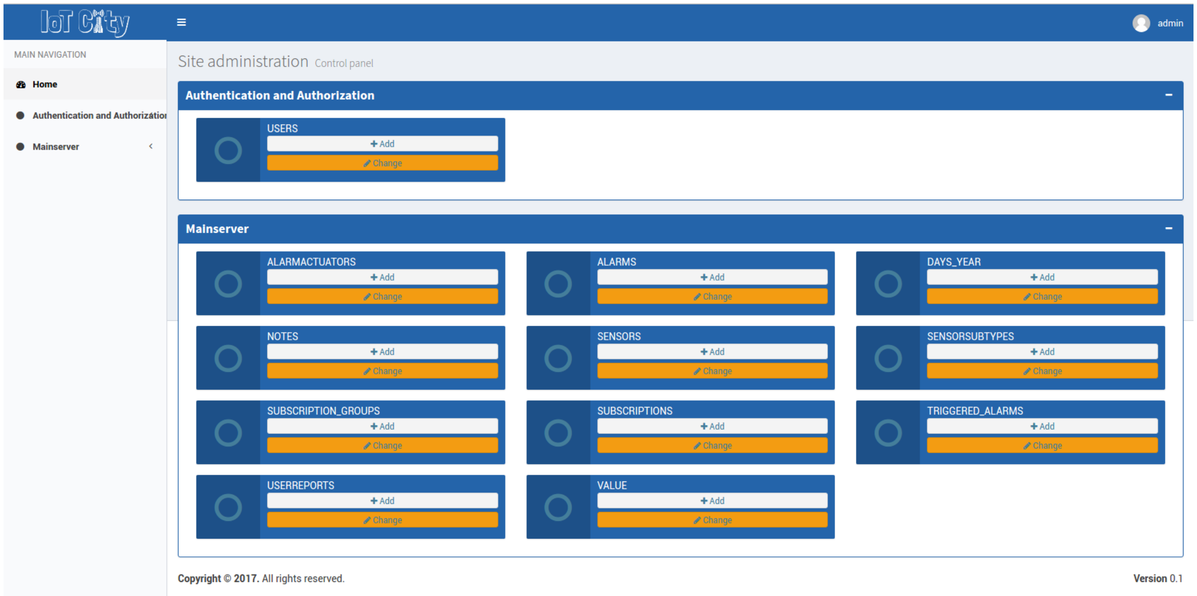
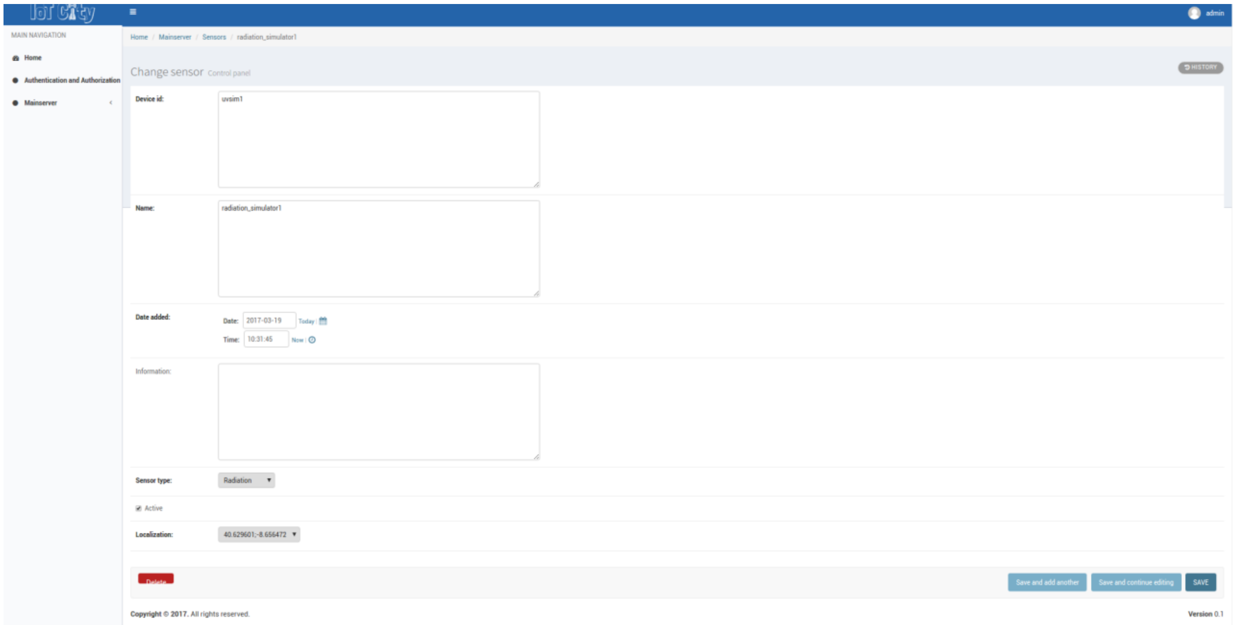


Figure 7 - Admin View

After we log in, we want to add a sensor, so we click on "Add" button, on the "Sensors" card. Then, we can fill the form with the description of the sensor: Id, Name, date added to the IoT Platform, other information, sensor type, its' localization and if it is active.


IoT City
admin
MAIN NAVIGATION
Home
Authentication and Authorization
Mainserver
Home / Mainserver / Sensors / radiation_simulator1
HISTORY
Delete
Save and add another
Save and continue editing
SAVE

Change sensor Control panel

Device id: uvsim1

Name: radiation_simulator1

Date added: Date: 2017-03-19 Today | Time: 10:31:45 Now |

Information:

Sensor type: Radiation

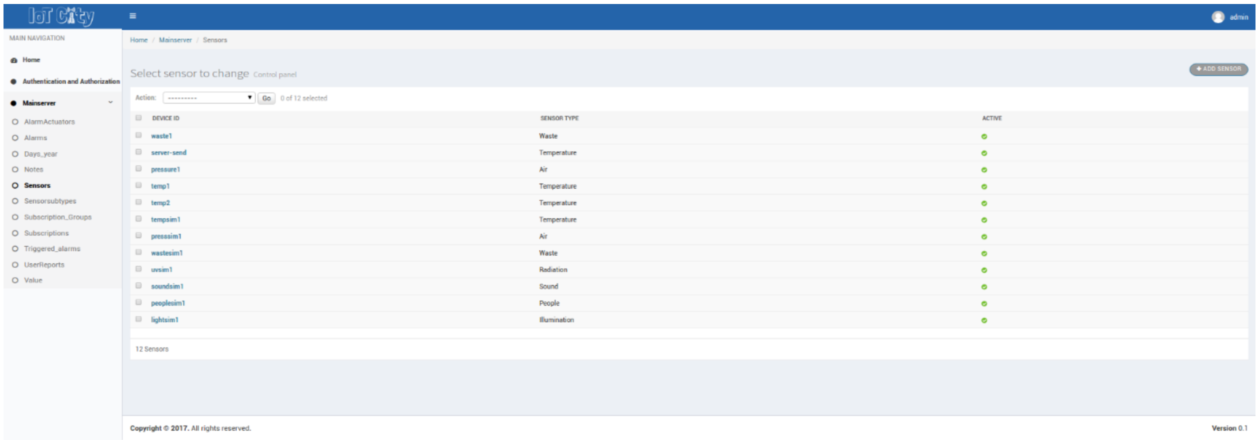
☑ Active

Localization: 40.629601;-8.656472

Copyright © 2017. All rights reserved. Version 0.1

Figure 8 - Add a sensor

After we add a sensor, we can see all the sensors in the platform, including the recently added one.



Select sensor to change Control panel

Action: --------- 0 of 12 selected

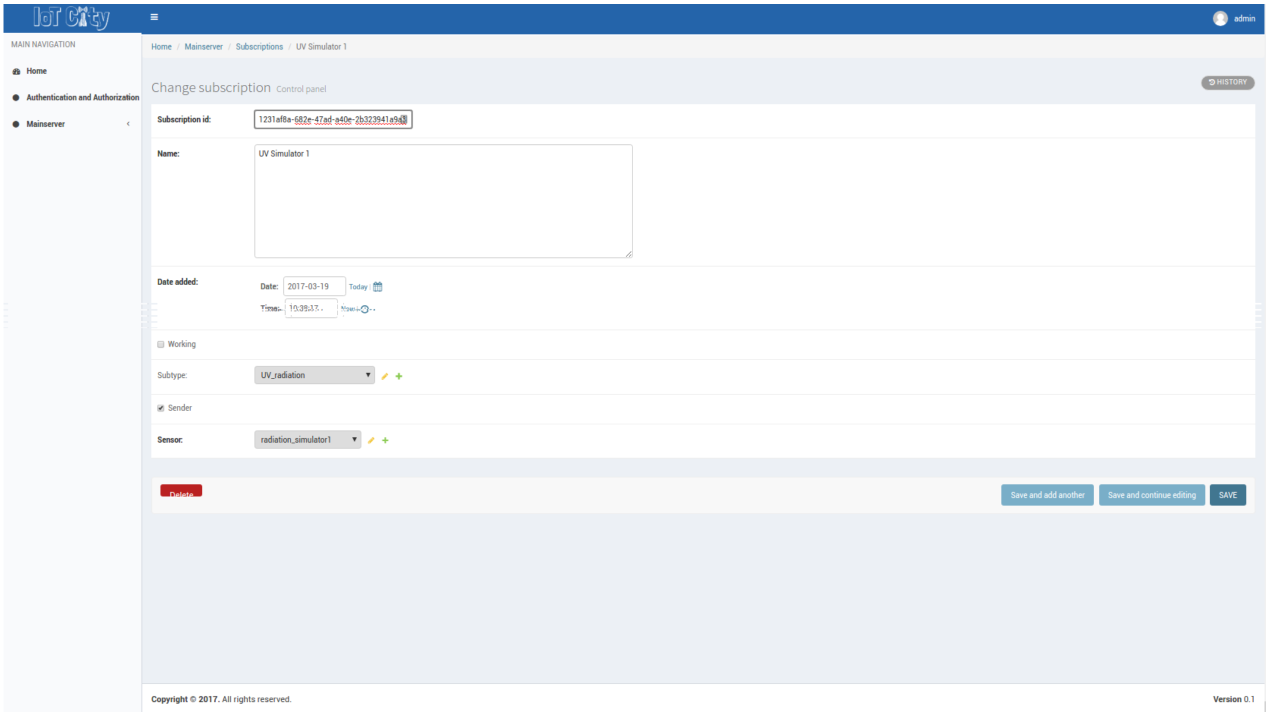
	DEVICE ID	SENSOR TYPE	ACTIVE
□	waste1	Waste	✓
□	server-send	Temperature	✓
□	pressure1	Air	✓
□	temp1	Temperature	✓
□	temp2	Temperature	✓
□	tempsim1	Temperature	✓
□	pressim1	Air	✓
□	wastesim1	Waste	✓
□	uvsim1	Radiation	✓
□	soundsim1	Sound	✓
□	peoplesim1	People	✓
□	lightsim1	Illumination	✓

12 Sensors

Copyright © 2017. All rights reserved. Version 0.1

Figure 9 - All the sensors in the platform

Then, we want to add the subscriptions of that sensor to the platform. For that, we click on "Add" Button, on the "Subscriptions" card, and we enter the information of the subscription in the form. The subscription id is the id returned by the broker when adding the IoT City Web Portal as a subscription to the stream. The sensor subtype is the subtype of that subscription (e.g. For the waste container type there are the subtypes of waste percentage of container, internal container temperature and waste volume). We can also specify if the subscription sends or receives values (if receives values is an actuator), by checking or unchecking the "sender" checkbox.


IoT City
≡
admin
MAIN NAVIGATION
Home
Authentication and Authorization
Mainserver
Home / Mainserver / Subscriptions / UV Simulator 1
HISTORY
Today
Now
Delete
Save and add another
Save and continue editing
SAVE

Change subscription Control panel

Subscription id: 1231af8a-682e-47ad-a40e-2b323941a9a3

Name: UV Simulator 1

Date added: Date: 2017-03-19 Time: 10:38:37

□ Working

Subtype: UV_radiation

☑ Sender

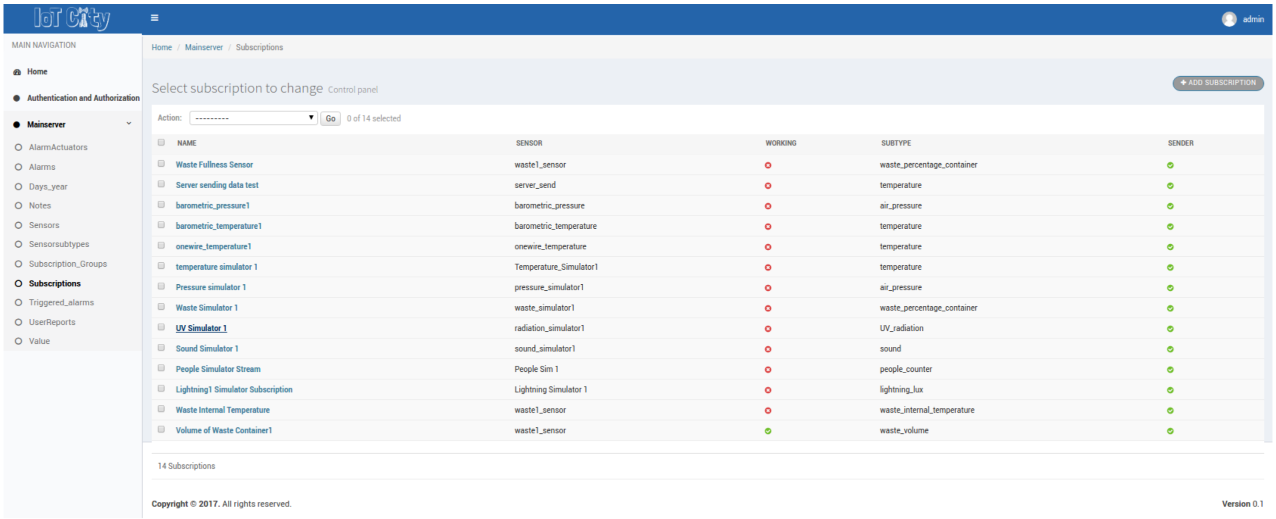
Sensor: radiation_simulator1

Copyright © 2017. All rights reserved.

Version 0.1

Figure 10 - Add a Subscription

After we add a subscription to a sensor, we can see all the subscriptions in the platform, including the recently added one.



Select subscription to change Control panel

0 of 14 selected

□	NAME	SENSOR	WORKING	SUBTYPE	SENDER
□	Waste Fullness Sensor	waste1_sensor	✗	waste_percentage_container	✓
□	Server sending data test	server_send	✗	temperature	✓
□	barometric_pressure1	barometric_pressure	✗	air_pressure	✓
□	barometric_temperature1	barometric_temperature	✗	temperature	✓
□	onewire_temperature1	onewire_temperature	✗	temperature	✓
□	temperature simulator 1	Temperature_Simulator1	✗	temperature	✓
□	Pressure simulator 1	pressure_simulator1	✗	air_pressure	✓
□	Waste Simulator 1	waste_simulator1	✗	waste_percentage_container	✓
□	UV Simulator 1	radiation_simulator1	✗	UV_radiation	✓
□	Sound Simulator 1	sound_simulator1	✗	sound	✓
□	People Simulator Stream	People Sim 1	✗	people_counter	✓
□	Lightning1 Simulator Subscription	Lightning Simulator 1	✗	lightning_lux	✓
□	Waste Internal Temperature	waste1_sensor	✗	waste_internal_temperature	✓
□	Volume of Waste Container1	waste1_sensor	✓	waste_volume	✓

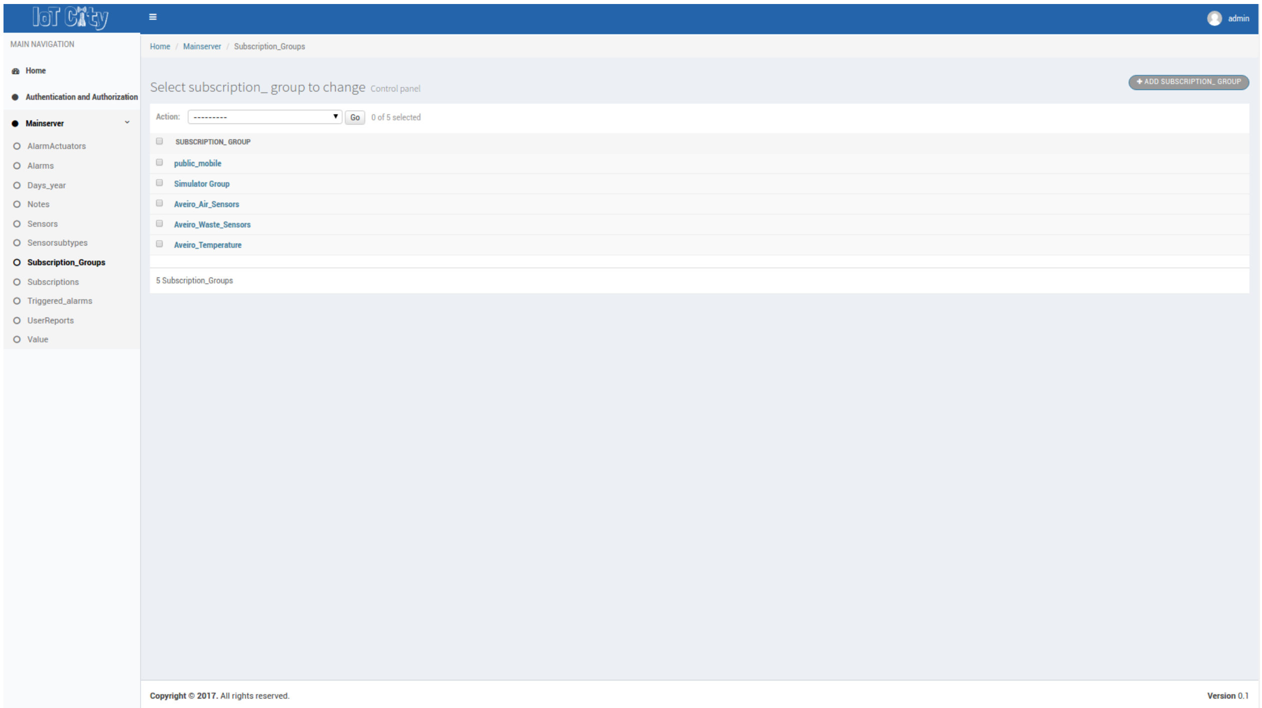
14 Subscriptions

Copyright © 2017. All rights reserved.

Version 0.1

Figure 11 - All subscriptions in the platform

Finally, we want to allow all the users in a group to see the values of the subscriptions. For that, we go to the "Subscription_group" to see the available groups.



Select subscription_ group to change Control panel

Action: --------- Go 0 of 5 selected

□ SUBSCRIPTION_GROUP

□ public_mobile

□ Simulator Group

□ Aveiro_Air_Sensors

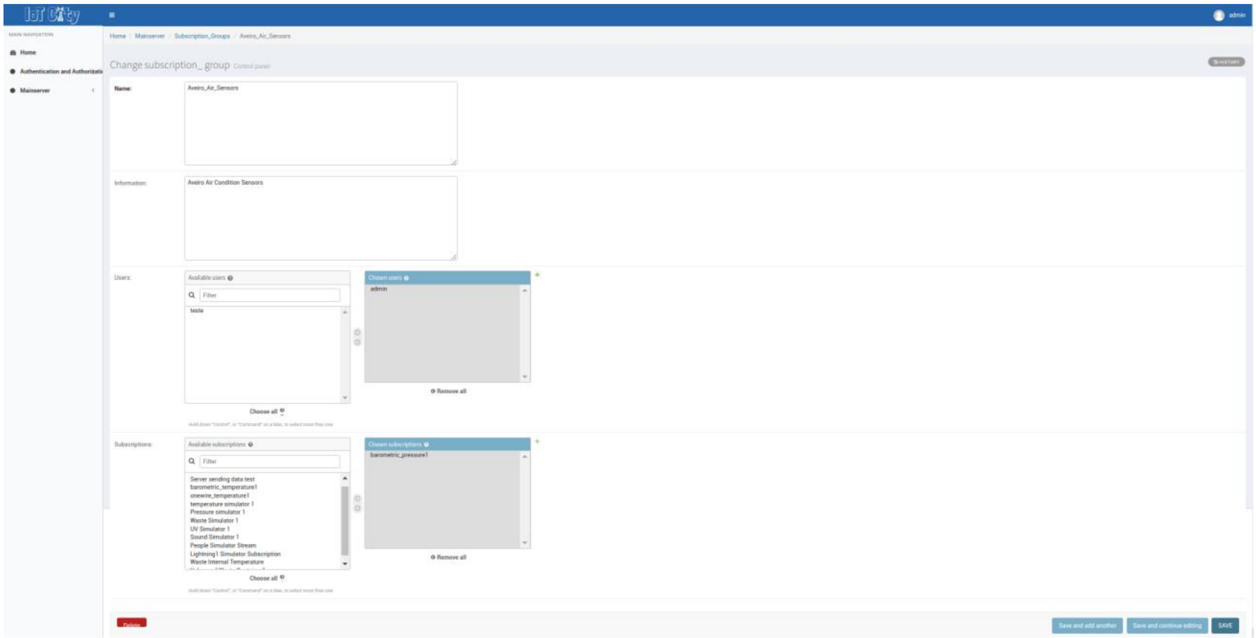
□ Aveiro_Waste_Sensors

□ Aveiro_Temperature

5 Subscription_Groups

Figure 12 - All Subscription Groups

After we select a group, on "Subscriptions", we add the new subscription to that group.



Change subscription_ group Control panel

Name: Aveiro_Air_Sensors

Information: Aveiro Air Condition Sensors

Users:
Available users: teste
Chosen users: admin

Hold down "Control", or "Command" on a Mac, to select more than one.

Subscriptions:
Available subscriptions:
Server sending data test
barometric_temperature1
onewire_temperature1
temperature simulator 1
Pressure simulator 1
Waste Simulator 1
UV Simulator 1
Sound Simulator 1
People Simulator Stream
Lightning1 Simulator Subscription
Waste Internal Temperature

Chosen subscriptions: barometric_pressure1

Hold down "Control", or "Command" on a Mac, to select more than one.

Figure 13 - Add subscription to subscription group

Mobile Application

On the initial view of the mobile application, the user can see a dashboard with seven cards, one for each type of sensor. Each card will have the values gathered from the sensor that is currently closer to the user in real-time.

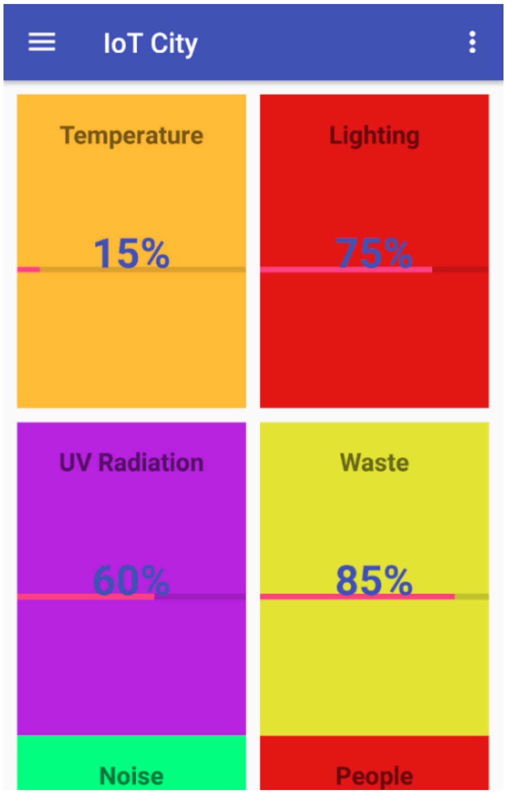


Figure 14 - Dashboard View

The application will have a left sidebar menu, which the user can access by sliding the finger from the left to right in the application, or by tapping on the top left icon. The menu has the following items: "Dashboard", "Map", "Metrics", "Reports", and "Help & Contacts". The "Dashboard" option redirects us to the dashboard, the "Map" takes us to the map view. The "Metrics" option retrieves a list with all the types of values collected by the app. The "Report" option takes us to a screen where the user can report a problem through text and also adding photos/videos. The "Help & Contacts" options shows a page with a help text and contacts of the app developers.

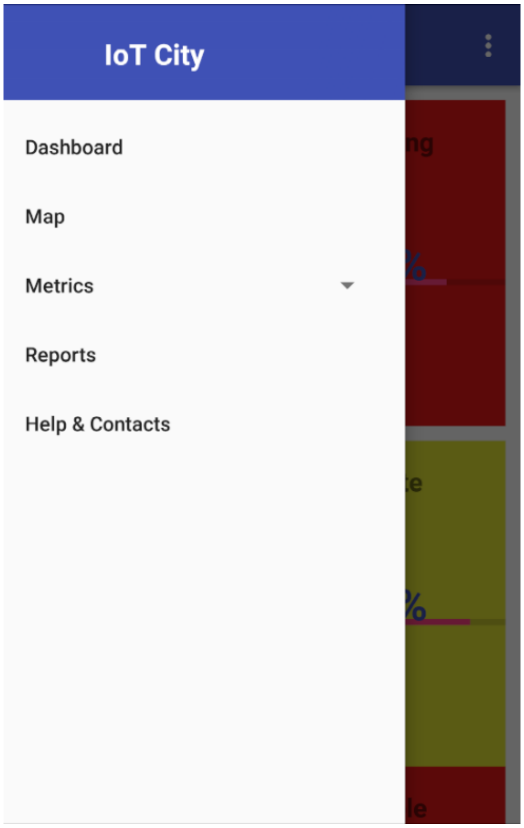


Figure 15 - Sidebar Menu

The map page will show a map with markers on the location of the sensors. By clicking on the markers, the user will see a toolbar with information about that sensor.

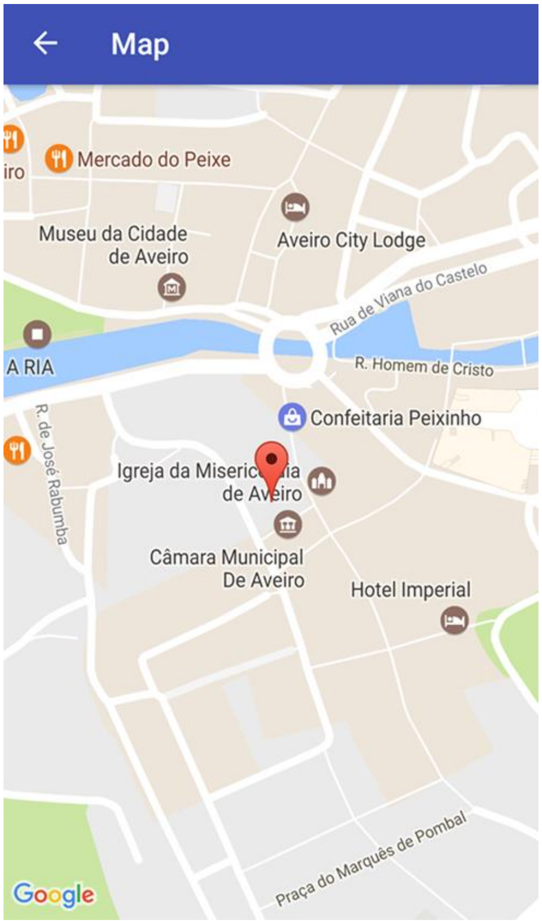
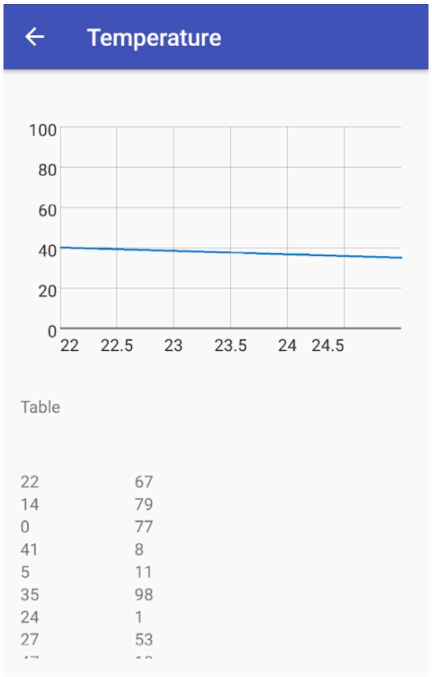


Figure 16 - Map View

Clicking on a given type of data (e.g. temperature), the user is redirected to a page with a chart showing the data retrieved by all the temperature sensors and a table with the same information. The user can zoom in or zoom out in the graphic to see the data with better accuracy. All the pages have a back button to go to the previous page.



Temperature

Table

22	67
14	79
0	77
41	8
5	11
35	98
24	1
27	53

Figure 17 - Temperature View

Technician

When the sensor technician sets up the sensors, he can use the APIs available on the platform to automatically add sensors and streams. To add a sensor to the platform, the technician can send a 'POST' request to /ws/addsensor. The body of the request must contain the following fields:

- ID - Id of the device
- password - A Password for device authentication
- name - Name of the device on the Platform
- description - Description of the device
- type - Type of the device (Available types: 'TE' (Temperature), 'AI' (Air), 'WA' (Waste), 'SO' (Noise), 'PE' (People), 'IL' (Lighting), 'RA' (Radiation))
- lat - Latitude of the device
- lon - Longitude of the device

To facilitate the changes of the parameters, exists a default scripts for adding sensors, it is only needed a file "configadd.json" on the same folder than the script, with the parameters configured (you can see an example on the folder sensors).

After adding a sensor, there is also an API for adding streams, with a 'POST' request to /ws/addsubscription. The body of the request must contain the following fields:

- sensor_id - Id of the sender device
- subtype - Subtype of the stream (Available subtypes: 'C2' (CO2), 'VR' (Visible Radiation), 'IR' (Infrared Radiation), 'WV' (Waste Volume), 'WT' (Waste Internal Temperature), 'LI' (Lighting Illumination), 'PC' (People Counter), 'NL'

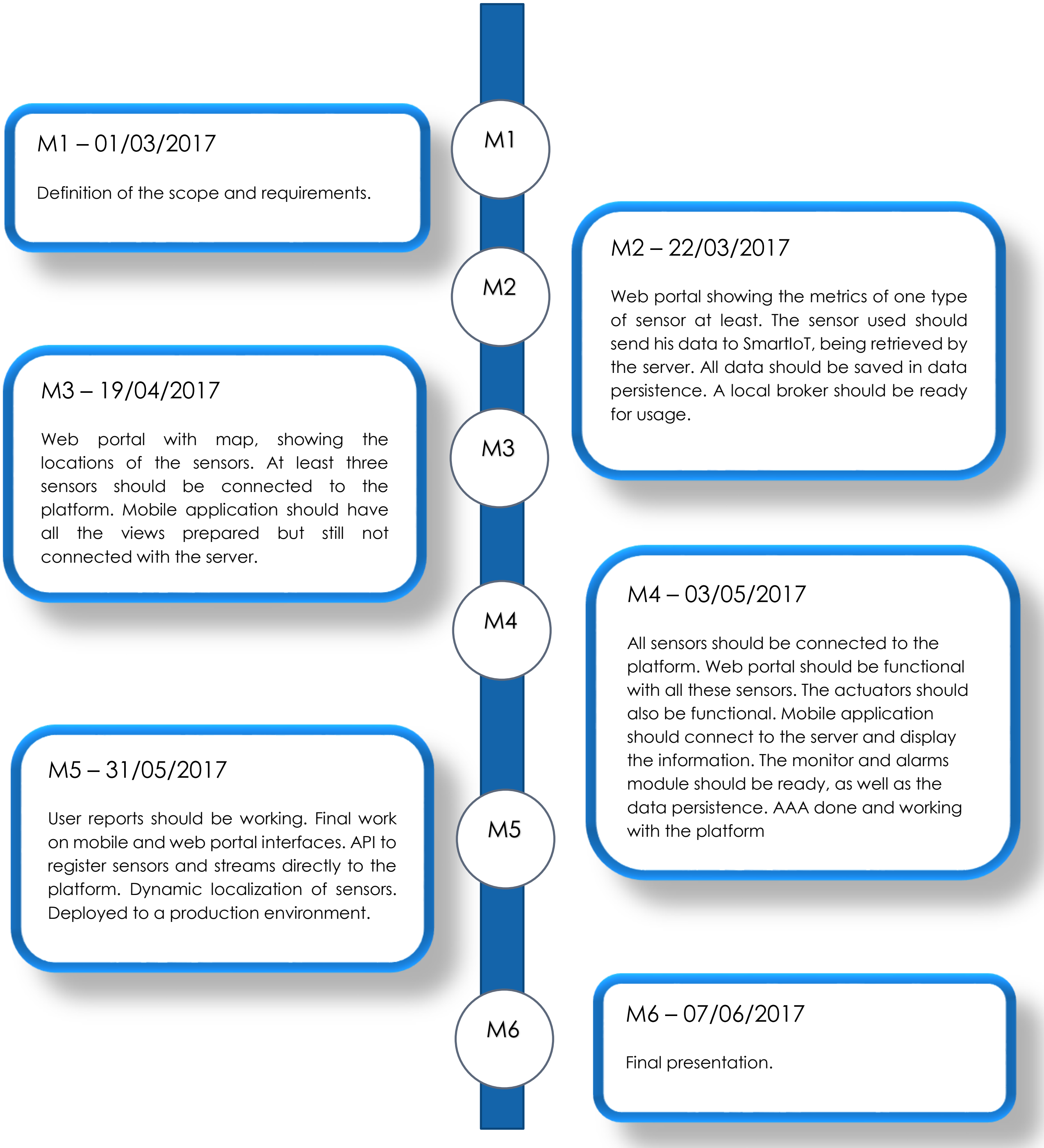
(Noise Level), 'UV' (UV Radiation), 'AP' (Air Pressure), 'WP' (Waste Fullness Percentage), 'TE' (Temperature), 'LT' (Latitude), 'LO' (Longitude))

- subs_name - Name of the stream
- sub_description - Description of the stream

To facilitate the changes of the parameters, exists a default scripts for adding streams, it is only needed a file "configstream.json" on the same folder than the script, with the parameters configured (you can see an example on the folder sensors).

Project timeline

This section sets the main goals for each milestone, clearly indicated the objectives and features that must be achieved on each date.



M1 – 01/03/2017

Definition of the scope and requirements.

M2 – 22/03/2017

Web portal showing the metrics of one type of sensor at least. The sensor used should send his data to SmartIoT, being retrieved by the server. All data should be saved in data persistence. A local broker should be ready for usage.

M3 – 19/04/2017

Web portal with map, showing the locations of the sensors. At least three sensors should be connected to the platform. Mobile application should have all the views prepared but still not connected with the server.

M4 – 03/05/2017

All sensors should be connected to the platform. Web portal should be functional with all these sensors. The actuators should also be functional. Mobile application should connect to the server and display the information. The monitor and alarms module should be ready, as well as the data persistence. AAA done and working with the platform

M5 – 31/05/2017

User reports should be working. Final work on mobile and web portal interfaces. API to register sensors and streams directly to the platform. Dynamic localization of sensors. Deployed to a production environment.

M6 – 07/06/2017

Final presentation.

Risks and Issues

On this section, we address the risks and issues identified in the project and possible solutions that can minimize the damage.

Sensors not provided

If some sensors are not provided to us by Altice Labs or IT we might not be able to measure some specific metrics. On that case, we have two possible solutions: (1) We can simulate the sensors or (2) we can eliminate that sensor type from our platform.

SmartIoT not available

If Altice Labs SmartIoT platform is not operational, we cannot read sensors' data neither can we operate the sensors. On that case, we must address the issue with *Altice Labs* and work with our developed broker until the platform is operational.

Sensors not working properly

If the sensors are not working properly then the gathered data will not be accurate. We can detect it by observation of strange data retrieved, that doesn't match with the actual perception. If that happens, we must speak with our mentor and collaborators so that they can help us fix the underlying problem.

Report Versions

This section describes the main changes between each version of the report.

Version	Date	Main Changes
1.0	March 1, 2017	Initial version of the report
1.1	March 5, 2017	“Context” section extended. “Vision” section added. “Use cases” section added. “Risks and Issues” section changed. Added Features “WebP_F17” and “WebP_F18”. Architecture changed, with “Gateways” added.
1.2	March 12, 2017	“Timeline” changed (M2, M3 and M4).
1.3	March 24, 2017	Added Features "WebP_F19" and "App_F7" Added Use Cases "Web_UC18" and "App_UC7"
1.4	23 April, 2017	Administrator Portal Added. Analytics Module removed. Screenshots of interfaces added. Total Rewrite of Use Cases and features.
1.5	1 May, 2017	Dynamic Location of sensors added. API’s for the technician to add streams and sensors added.