DELIVERABLE 6.2
Performance evaluation of the Monitoring & Management Central

Project Acronym	**ENTOMATIC**
Project Reference:	**605073**
Project Title:	**Novel automatic and stand-alone integrated pest management tool for remote count and bioacoustic identification of the Olive Fly (Bactrocera oleae) in the field**

Deliverable 6.2 – Performance Evaluation of the Monitoring & Management Central

Revision: v3.0

Authors:

Albert Bel, Toni Adame, Maddalena Nurchis, Gabriel Martins, Boris Bellalta (UPF)

Ward Bryssinckx, Tom Matheusen, Mina Petric (Avia-GIS)

Received amendments and comments from:

Antonio Estévez (NUT)

Frank Spiller (IMMS)

Ilyas Potamitis (TEIC)

Michel Chamoun (MTSYSTEM)

Eda Ciner (AEGEAN)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
1	9/12/2016	Albert Bel	UPF	General structure
2	20/12/2016	Ward Bryssinckx	Avia-GIS	Structural revisions
3	28/12/2016	Albert Bel	UPF	Amendments and comments
4	10/3/2017	Toni Adame	UPF	Update of communication protocol and visualization
5	29/3/2017	Tom Matheusen	Avia-GIS	Revision of web page visualization and communication protocol
6	31/3/2017	Albert Bel	UPF	Final review
7	31/7/2017	Albert Bel	UPF	Review after comments
8	31/8/2018	Boris Bellalta, Toni Adame, Albert Bel	UPF	Review v3.0

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

This deliverable reflects the work performed for the development of the architecture that makes up the monitoring software of ENTOMATIC according to the design requirements of ***Task 6.1: Design of the Olive fly Monitoring & Management Central***.

Consequently, a software layer has been developed in order to receive data from traps deployed on the field, and the corresponding database architecture has been defined for the proper storage of gathered data. Thus, the main function of the developed platform is to read data from the field and store it (after pre-processing it) in the system database. It also incorporates the ability of performing system checks, detecting failures in nodes and sensors, including a troubleshooting guide, performing auto-calibration processes and incorporating "plug and play" devices. A user friendly graphical interface has been developed for the user to interact with all the above described modules.

All the developed modules follow the Model-View-Controller standard architecture, thus dividing the application into 3 interconnected parts: one managing the data, logic and rules, another one responsible for representing information, and the final one accepting inputs and converting them to commands.

The current deliverable includes the design of the ENTOMATIC monitoring and management central in Section 2, and describes how to use the whole web platform in Section 3. As for the GPRS communication between the hardware deployed on the field and the Monitoring and Management Central, Section 4 defines the two different approaches (single GPRS link and ENTOMATIC WSN + GPRS link). Lastly, Section 5 compiles the conclusions regarding the use of the GPRS communication protocols, the improvements incorporated to the data receiver server, and a discussion on the validation of the resulting platform.

2 PLATFORM ACCESS AND EDITION

2.1 MONITORING AND MANAGEMENT CENTRAL

The ENTOMATIC Monitoring and Management Central consists of a data receiver server, a web map server, the IPM (Integrated Pest Management) software and a database, as it can be seen in Figure 2.1.

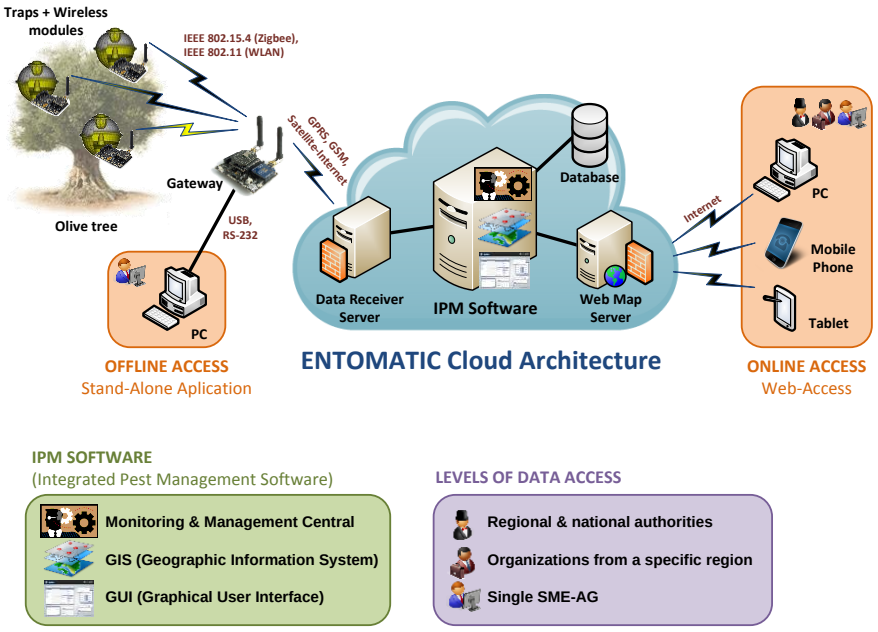

Figure 2.1: Cloud architecture of the ENTOMATIC Monitoring and Management Central

All these elements are hosted in UPF premises into different hardware components, so that the whole system is split into a backend and a frontend, as shown in Figure 2.2.

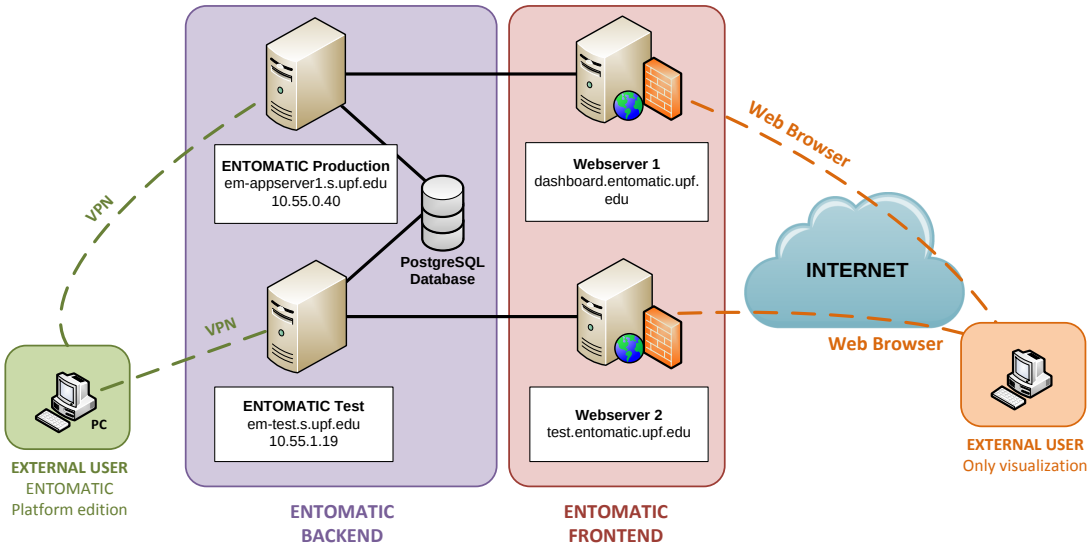

Figure 2.2: Diagram of ENTOMATIC platform access and edition

2.1.1 FRONTEND

As the name states, "Front-End" is the part of the code that is on the front of the application. It is usually visible to users in the form of an interface inviting to interact with the services. The main purpose of the front-end code is to interact with users, as well as to present data in a well-defined style and matter.

The ENTOMATIC frontend consists of two virtual servers responsible for dealing with the connections of external users who want to access the platform via internet. These servers maintain a secure communication with the backend servers and provide users with the requested information.

2.1.2 BACKEND

The back-end is the brain of any app. It is the part of the application that is never visible to the user. It is built using server-side language and database. In simpler words, front-end code interacts with users in real-time while back-end code interacts with a server to return user ready results. Anything displayed on the website is because of the query performed on the server returning data to the front-end.

In the ENTOMATIC case, two virtual servers exist in the back-end, each one of them with the following characteristics:

- Core, 6 GB RAM
- 145 GB Storage
- Debian GNU/Linux
- Apache
- PostgreSQL
- PHP
- Tomcat

The database will be supported by the PostgreSQL server using SQL standard ways of storing information and maintaining data consistency, which include:

- Data tables with rows for basic data storage.
- Primary Keys and Foreign keys for data consistency.
- Sequences for primary key Id generators.
- Indexes, as way for increasing querying performance.

Both servers have direct access to the PostgreSQL database and can be accessed from the outside by means of a secure VPN connection to UPF.

2.2 WEB APPLICATION WALKTHROUGH

This chapter contains information regarding the user interface and usage of the developed web application. It forms the basis of the whole ENTOMATIC system and is fully accessible from the following links:

ENTOMATIC Test Platform: http://test.entomatic.upf.edu/

ENTOMATIC Production Platform: http://dashboard.entomatic.upf.edu/

During the project development, both websites were available; one was intended to test the platform improvements and the other one to publish the consolidated versions. However, it is worth noting here that ***only the ENTOMATIC Test Platform webpage is currently available*** and contains the final version of the platform[1].

[1] In order to access to the test platform the following user and password can be used:
User: toni.adame | **Password:** entomatic

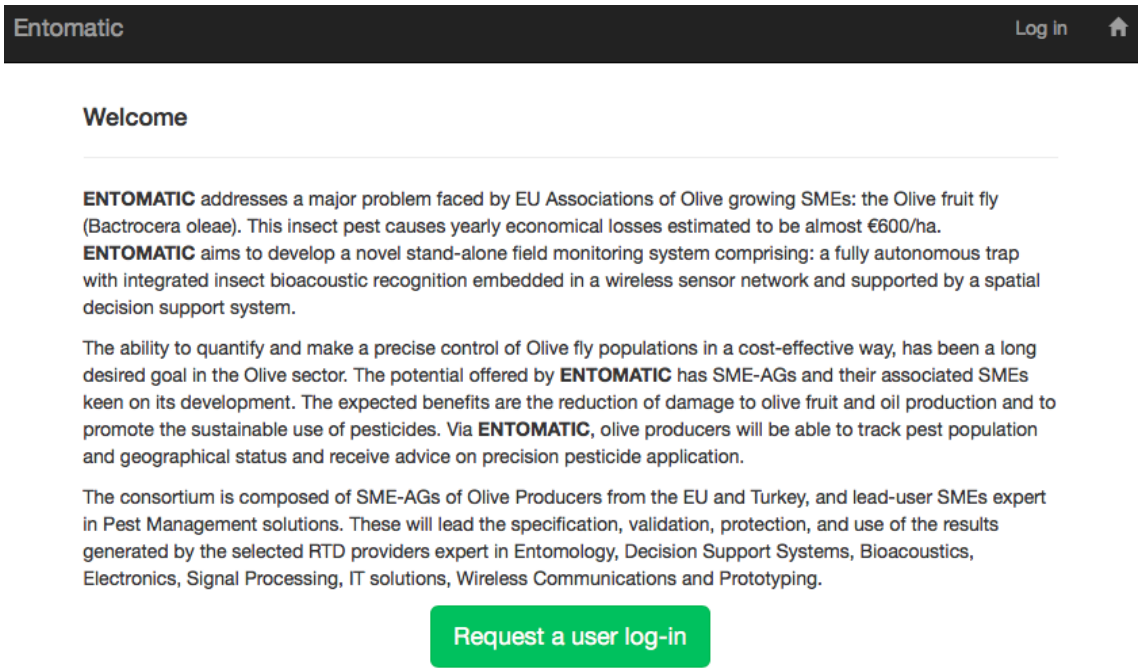

Figure 2.3: Screenshot of the ENTOMATIC home page

When accessing the test.entomatic.upf.edu web page, you can either log-in by clicking on "Log in" in the upper-right corner of the page if you already have an account, or request a user log-in by clicking on the "Request a user log-in" button in the middle of the page (Figure 2.3).

If the authentication fails, either because the user does not exist or the password is wrong, an error message is shown (Figure 2.4). If the authentication process is successful, then the browser redirects to the dashboard page.

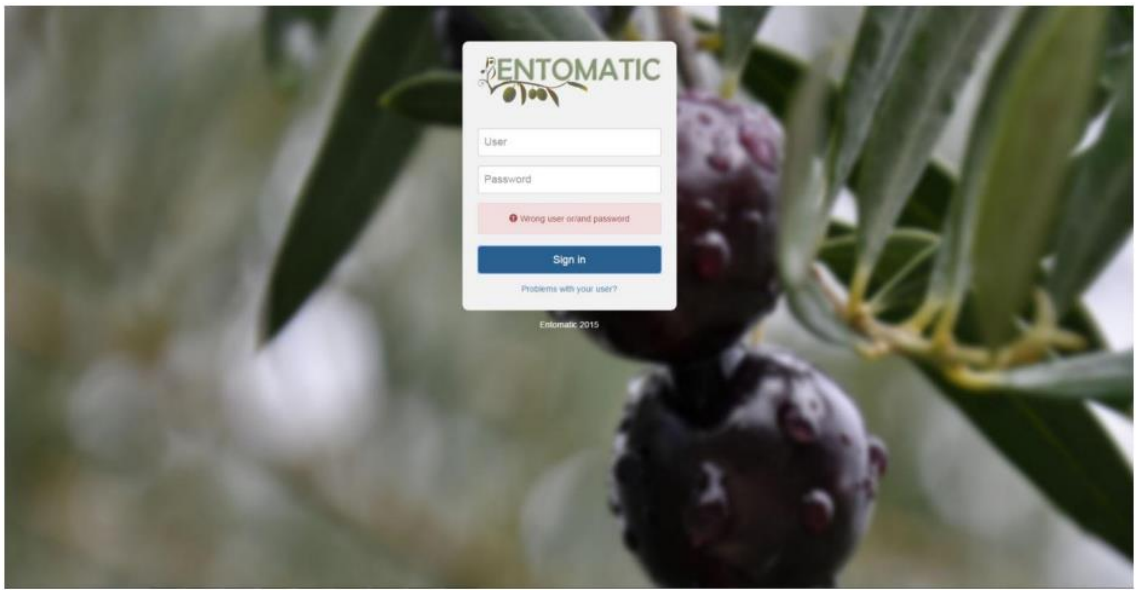

Figure 2.4: Screenshot of a failed authentication

When requesting a user log-in, one is asked to fill in a series of identification text fields (Figure 2.5). By using the organisation drop-down list, one selects the organisation he/she is directly part of (i.e., the lowest level structure from which the user is a member). If a mistake is made, it can still be changed later on by a user with administrator rights in an upper organization. Therefore, in case of mistake, a new request for a user log-in is not needed.

The username has to be unique; in an attempt to make a second user log-in request, a different username must be chosen. The e-mail address is used to send notifications as well as to reset the password (e.g. when the previous password was lost). Initially, a password can be freely chosen by entering it twice in the respective text fields.

To prevent robots from automatically generating data on the ENTOMATIC database and preventing our system from being used to spread spam e-mails, the symbol corresponding to the random green word in a bold font must be clicked prior to signing-up. After signing-up for a user account, an administrator of the organization that you selected, must confirm that you are allowed to use the web platform.

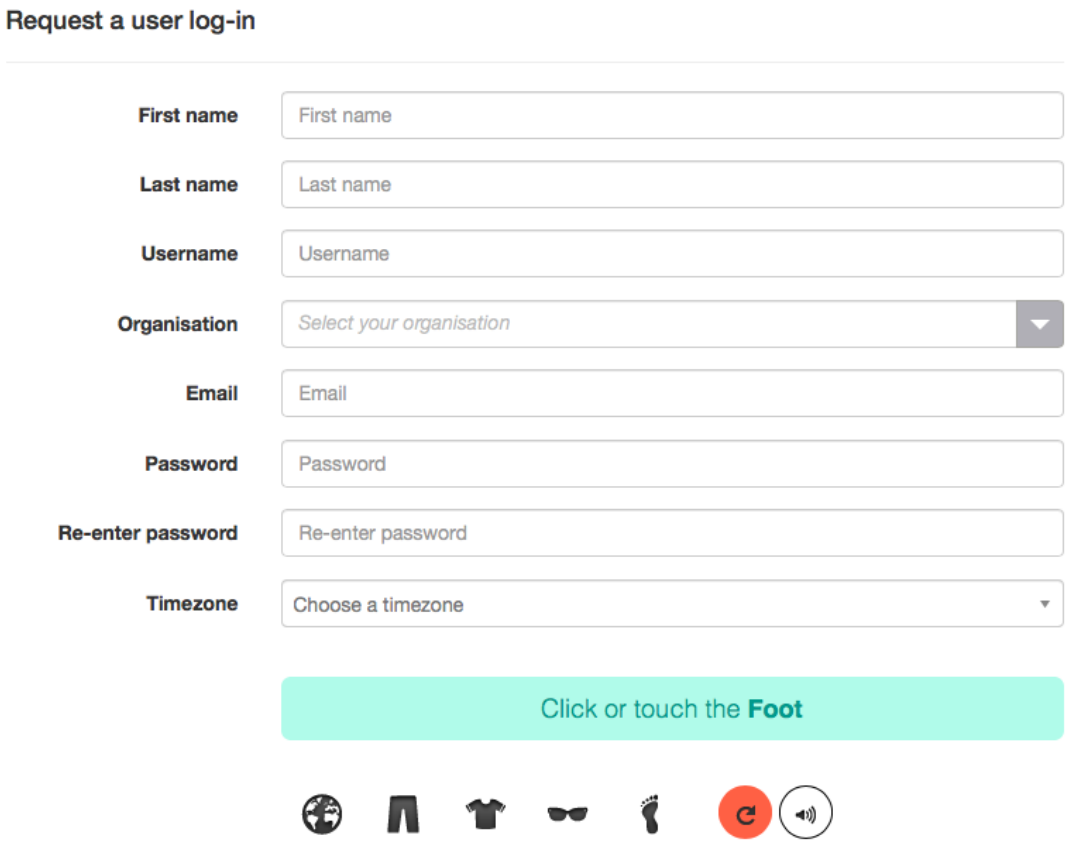

Figure 2.5: Questions asked when requesting a new user credentials

3 OVERVIEW OF THE WEB-APP

The ENTOMATIC web application gives users access to a variety of tools that are classified into five different web pages: (i) administration, (ii) pest management, (iii) analysis, (iv) network performance and (v) configuration (Figure 3.1).

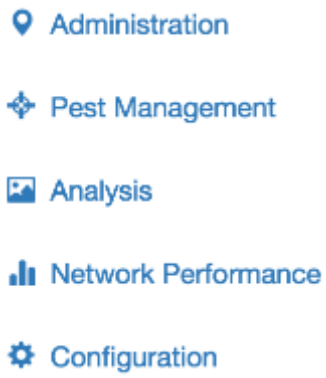

Figure 3.1: Symbols used to navigate through the different web app functionalities

Depending on the user type, a different set of functionalities is enabled. There are four different user types:

1. **Farmer**: has access to data from the sensors that are assigned to him/her;
2. **Administrator**: can assign sensors to users and perform user administration, when an administrator is assigned to an SME, he/she can view all data from sensors that are assigned to farmers within the SME;
3. **Supervisor**: can view the same data as an administrator but has no rights to edit data;
4. **Super user**: can view all data on the server and has the right to change the behaviour and appearance of the web app through the configuration page.

The different steps to enter the web app will be shown in the following sections. In addition, a guide to navigate among the different sections and options will be provided.

3.1 THE HOME SCREEN

The first screen that a user gets to see after successfully logging in is the home screen (Figure 3.2). This screen gives a summary of the information given in the other pages, so that the user is provided with a clear and quick overview of the most important data:

1. The number of active gateways and nodes
2. The latest recommendations
3. Alarms

All tables can be downloaded in a .csv file format by clicking “Download table as CSV”.

Home

System Status	Download table as CSV
Active Gateways	**Active Nodes**
8	25

Latest Recommendations — Download table as CSV

Date	Organisation	User	Type	Comment	Orchard
22-07-2016	National	ktheodorakis	Spraying		East

View more

Latest alarms — Download table as CSV

Date	Organisation	User	Sensor	Alarm Type	Value
19-12-2016	TEI of Crete	herc	209	Deactivated type	temp
19-12-2016	TEI of Crete	herc	209	Deactivated type	Hum
19-12-2016	TEI of Crete	herc	209	Deactivated type	insects
19-12-2016	TEI of Crete	herc	209	Deactivated type	batt
19-12-2016	TEI of Crete	herc	206	Deactivated type	Hum

View more

Figure 3.2: The home screen shows information summaries from other pages

3.2 ADMINISTRATION

In order for users to be able to select their organisation from the dropdown list when requesting a user log-in, organisations must first be added to the ENTOMATIC database. This can only be done by users with administrator or super user rights.

By clicking on the “Create new” button, a new row is added to the list with text fields in which the user can specify the name of the new organisation and the name of the upper organisation to which the new organization belongs. An existing organisation can be edited or deleted as well. Prior to deleting an organisation, all users within that organisation must be removed or linked to another organisation (Figure 3.3)

When one of the organisations is selected from the Organisations-section (Figure 3.4), the users that are linked to that organisation or to its “children” appear in the Users-section. When no organisation is selected, the Users-section remains empty. All attributes from a user can be edited by an administrator or a super user after clicking on the button that has a pencil.

Changes can be confirmed by using the confirm button on the right side of the row being edited, or cancelled by clicking on the cancel button. When no changes are made in the edit mode, only the cancel button can be used to exit the edit mode. The “Active” field contains a checkbox that is unchecked by default when a new user log-in request is made. Before the new user can access the application, this checkbox must be checked by the administrator.

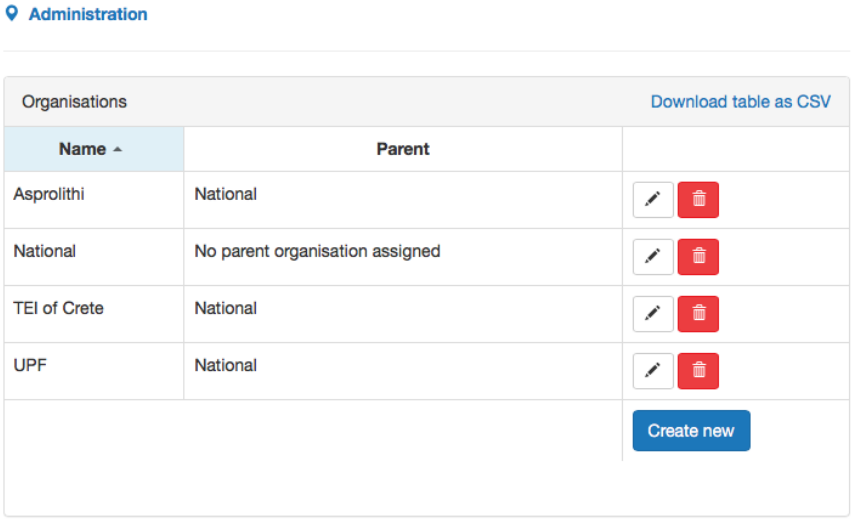

Figure 3.3: Organisations defined at the web-app

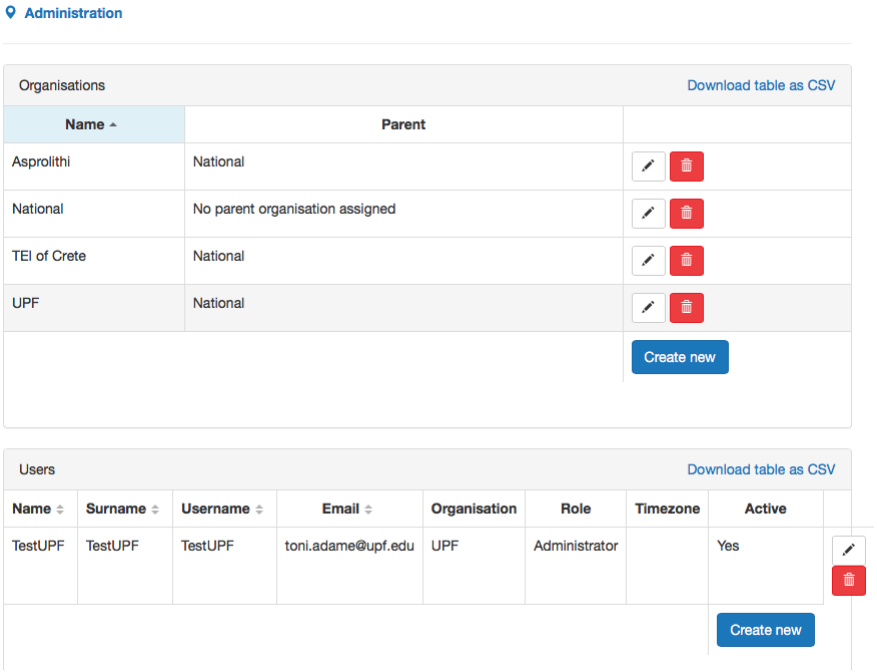

Figure 3.4: Users of the selected organisation

The Sensors-section shows all sensors registered in the ENTOMATIC database, whose gathered data has been received (Figure 3.5). This also includes sensors that were powered-on but not yet assigned to a user. The administrator may also assign new sensors to users by editing the data and selecting a user from the dropdown list.

In addition, users can be changed in case a sensor is moved from one user to another. The “View on Map” button allows the user to see where the sensor is located on a map together with the locations of other sensors that may or may not be already assigned to a user (see Figure 3.6).

Sensors — Download table as CSV

ID	Date of Last Power-On	User	Orchard	Action	
4	11/06/2016 - 14h10	Ipotamitis	testH	View on Map	
3	02/06/2016 - 12h28	Ipotamitis	Unassigned	View on Map	
9	22/07/2016 - 10h12	wbryssinckx	Unassigned	View on Map	
10	22/07/2016 - 10h12	wbryssinckx	Unassigned	View on Map	

Figure 3.5: Sensors that were powered on and detected by the system

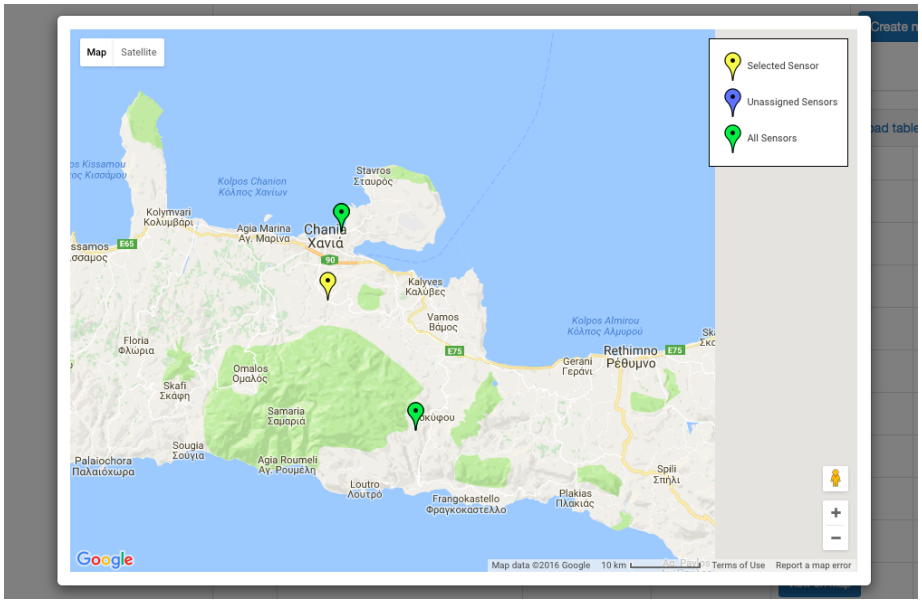

Figure 3.6: "View on Map" screen

After selecting an organisation and a user, a new section is shown in the administration page, which is called “Orchards”. Orchards need to be created first, before a user could assign their own sensors to a specific orchard. The only information that is needed to create a new orchard is a unique name. After this is done, one can select the orchard and click the “View Sensors” button which brings the user to the orchard’s administration page. For getting back to the main administration page, it is necessary to click the “Back” button on the upper right corner (Figure 3.7).

In the main administration page, a farmer, administrator and super user are allowed to assign sensors to orchards after which they will appear on the orchard administration page. In the orchard’s administration page there are three different sections: (i) a Sensors section, (ii) a Map section that shows the orchard’s sensors and (iii) a Last sensor readings section. Once a sensor is selected from the Sensors-section, this sensor is highlighted on the map and its sensor readings are given in the Last sensor readings-section.

The label of a sensor can be changed to a text, which is easier to remember than the MAC address (specific to the sensor hardware) or the ID (linked to the ENTOMATIC database). The text that is chosen to enter in this field could, for example, also be written on a physical label that can be attached to the trap for easy reference.

The last sensor readings-section shows all sensor readings for the selected sensor. As there can be many readings that make it hard to interpret the data, readings can be filtered by checking or unchecking the boxes in front of the different variables that are being monitored. For each reading the date and time of acquisition is given, as well as the Measurement Type and the Sensor Reading itself.

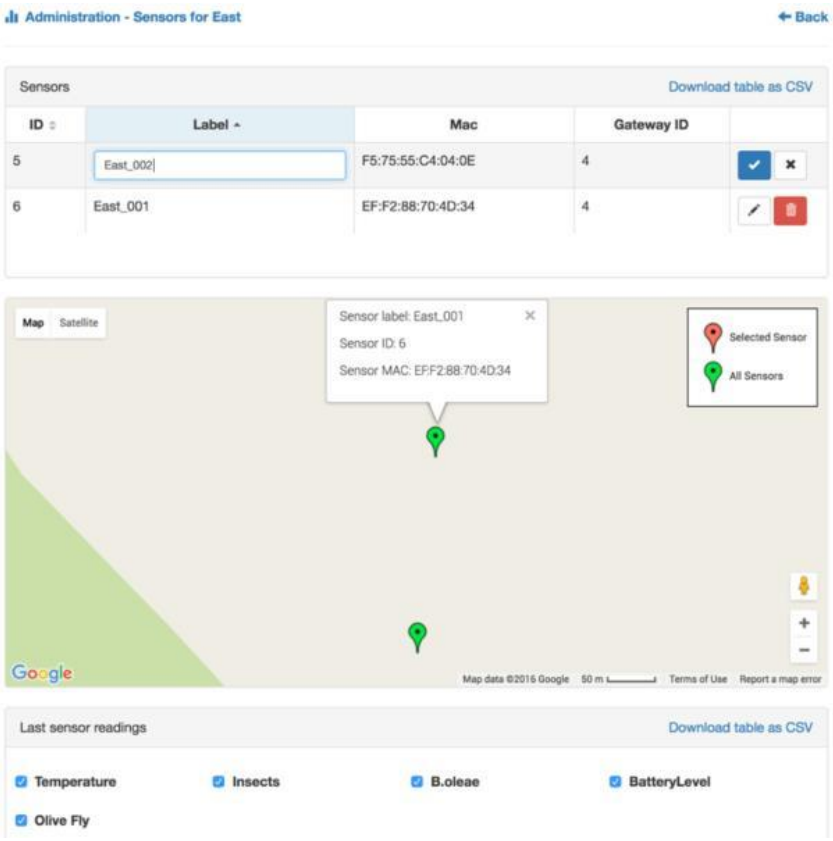

Figure 3.7: Sensor details screen [2]

A super user has the required rights to disable certain Measurement Types in the configuration page. Once disabled, they will still be stored in the ENTOMATIC database but users will no longer be able to see their Sensor Readings until it is enabled again.

3.3 PEST MANAGEMENT

The pest management page shows all information related to Recommendations that are automatically generated by the ENTOMATIC spatial decision support system (SDSS) and Control Actions that are manually entered by the user (Figure 3.8).

The Recommendations-section shows following information:

1. **Date**: the date on which the recommendation was generated. The ENTOMATIC system runs an algorithm each day to decide whether new recommendations should be given;
2. **Type**: the type of control action that is recommended at the moment;
3. **Comment**: a brief text describing why a certain recommendation was given;
4. **Orchard**: the orchard to which a recommendation applies.

[2] After selecting an orchard, details of sensors that are assigned to that orchard are shown. One can label the sensors to recognise them more easily on a map.

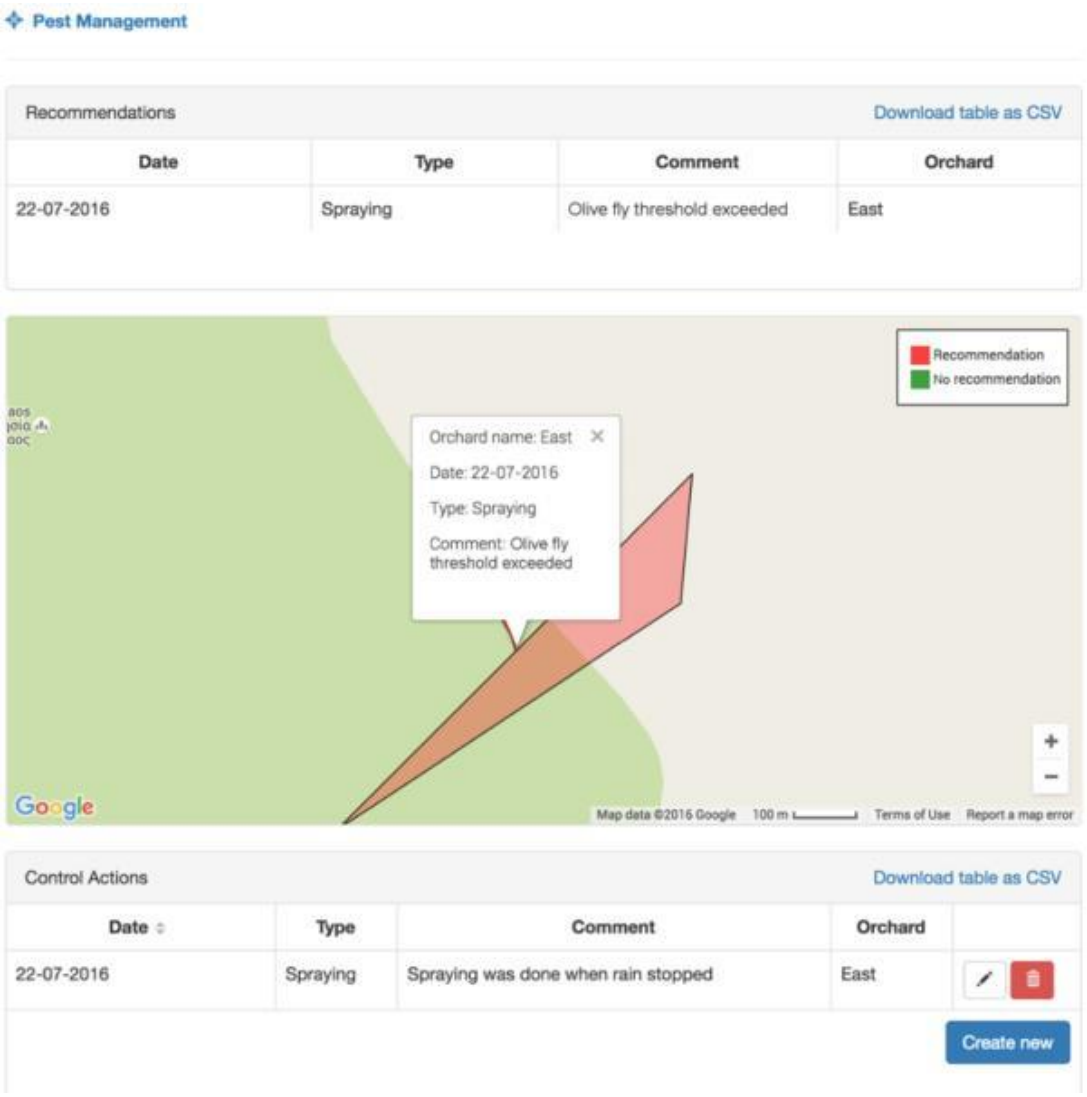

Figure 3.8: Pest managament screen [3]

While the Recommendations-section reflects the data generated by the ENTOMATIC system and visible to administrators, supervisors and super users, the Control Actions section only shows information that was manually entered by the farmer. The information uploaded by this way from a farmer is not visible to other farmers, administrators, supervisors and super users.

While the information on Control Actions can be edited or deleted, this cannot be done for Recommendations.

[3] The pest management page shows recommendations that are automatically generated by the ENTOMATIC spatial decision support system (SDSS) and control actions that are manually entered by the user.

3.4 ANALYSIS

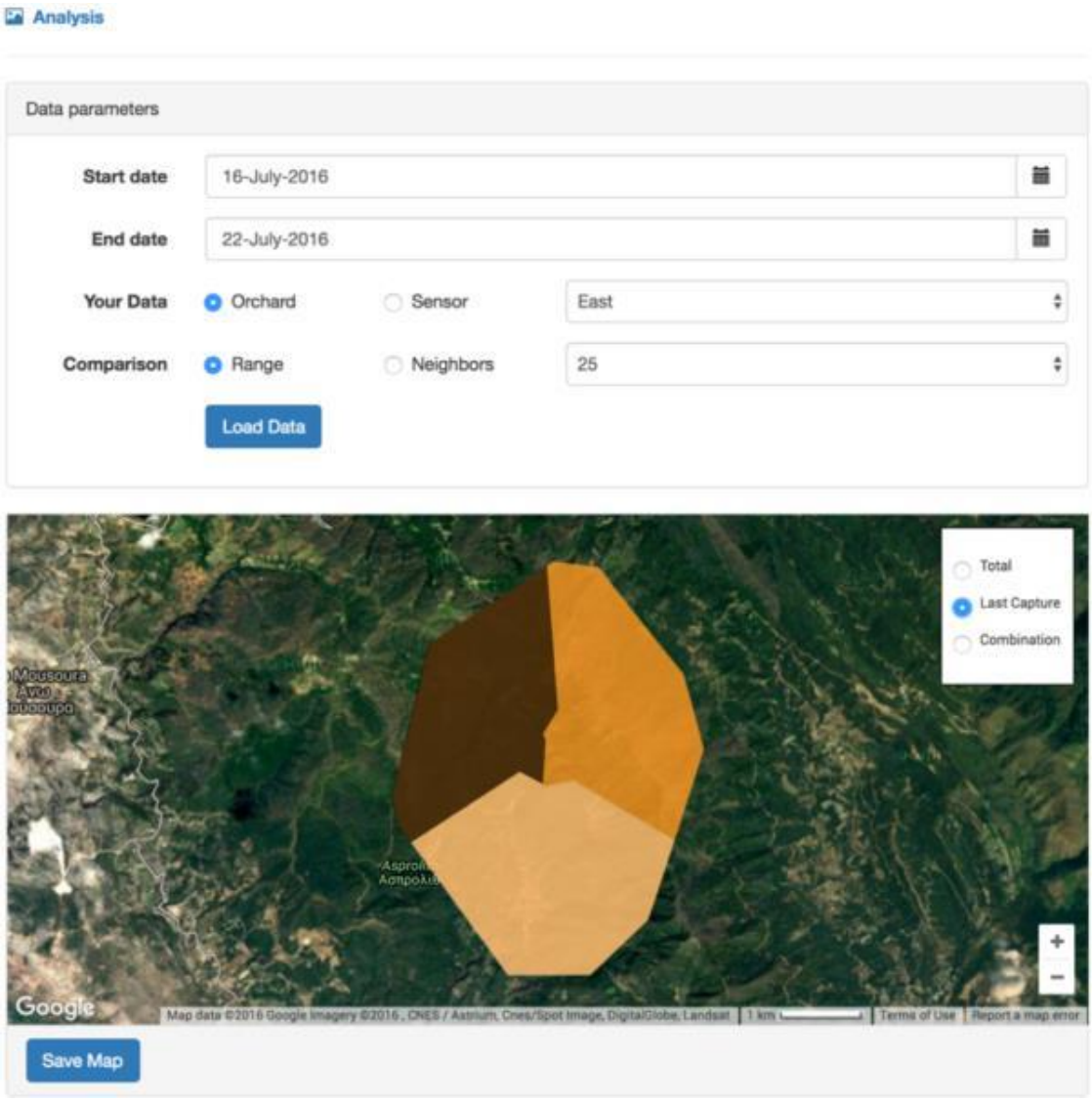

Figure 3.9: Analysis screen [4]

For those users interested to go further than viewing recommendations, the web platform also offers the possibility of analysing sensor readings both in space and time (Figure 3.9).

In the Data parameters-section, data can be loaded for analysis, based on a timeframe and sensor selection. The sensor selection is done for the own data by selecting one orchard or one sensor. After clicking one of these radio buttons, the name of the orchard or sensor can be selected from the drop-down list. This data may be used by a user to compare it with their own data, and can be selected either based on a range around the selected own data, or by giving a number of nearest sensors that must be taken into consideration. After clicking the "Load Data" button, the map and graph are updated. Both the map and the graph can be saved locally as a *.png* image.

The map shows the total number of captured olive flies on the last full day, the number of days since the last olive flies were captured or a combination of both metrics. When "Combination" is selected, the number of olive flies captured on the last day is given by using different colours, whereas the number of days since the last olive flies were captured is represented by a range of different opacity levels.

[4] The analysis page allows the user to compare data from its own orchards or sensors to data from the nearest sensors (also from other users) and sensors within a given radius.

The graph shows a time-series of boxplots representing the distribution of sensor reading values that were registered for the day given on the X-axis or the period around it (Figure 3.10). The amount of days for which data is aggregated in one box depends on the length of the selected timeframe to keep the graph easy to interpret.

Figure 3.10: Graphical representation of sensor readings on a timeline [5]

If one would prefer to have other graph types instead of this boxplot to visualize own data, data on sensor level can be saved in a *.csv* file format from the orchard administration page. Next, data can be then opened in other software platforms such as Microsoft Excel.

3.5 NETWORK PERFORMANCE

In the network performance page users can monitor the status of their sensors in terms of e.g. battery level. When the battery voltage drops below a certain level, the user receives a warning so he/she knows that batteries need to be replaced (Figure 3.11).

[5] The users' own data are shown using both boxplots as averages. The average of the comparison data is also given.

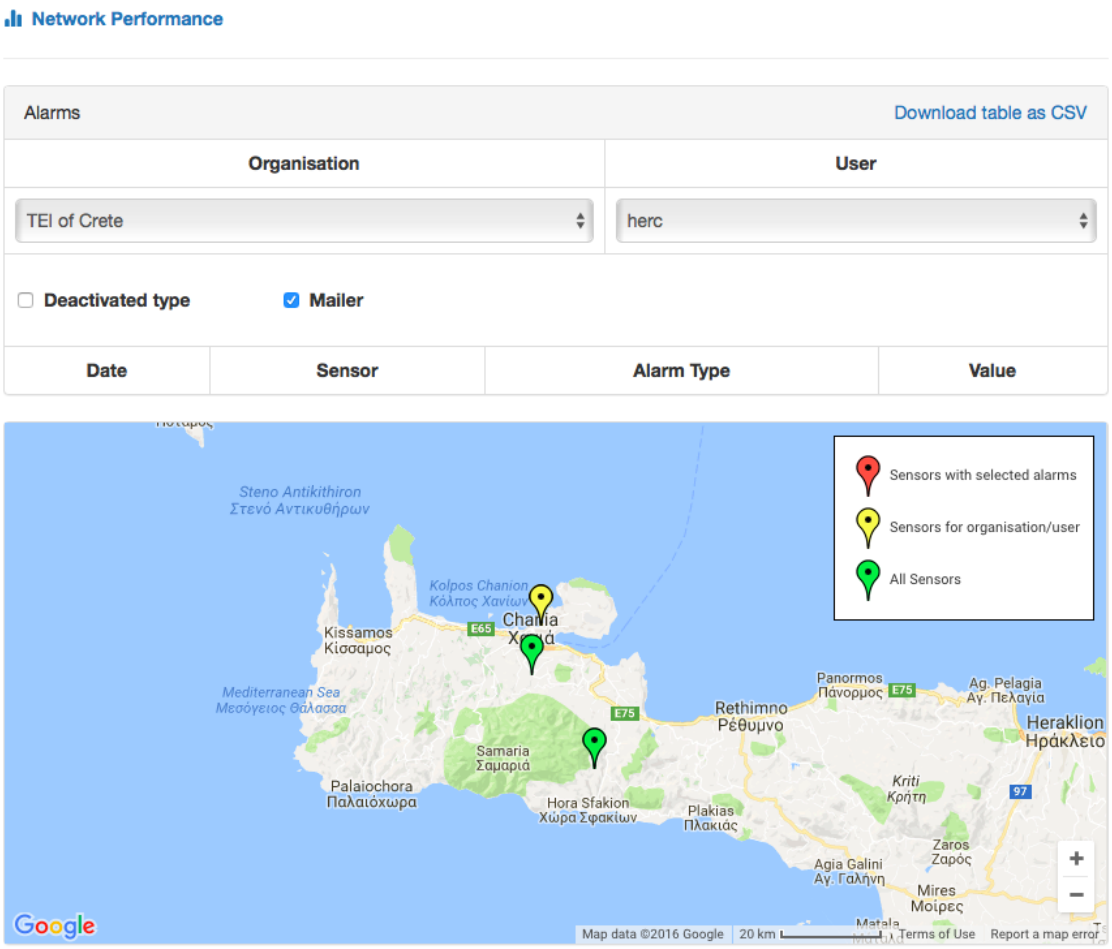

Figure 3.11: Network performance screen

Administrators, supervisors and super user can see alarms from all sensors of farmers that are members of their own organisation.

The “Deactivated type” is a special case of alarm that shows what data was received for a deactivated measurement types. The “value” column specifies the measurement type the alarm is referring to.

As an administrator, supervisor or super user, it is also possible to see an Error log -section (Figure 3.12) that shows all alarms from sensors that are not directly assigned to a user. Using this error log, it is possible to troubleshoot erroneous sensors. Sensors that were powered on at the time given in the error log were able to contact the server; however, for some other reason, they could not be registered in the database and therefore they could not be assigned to a user.

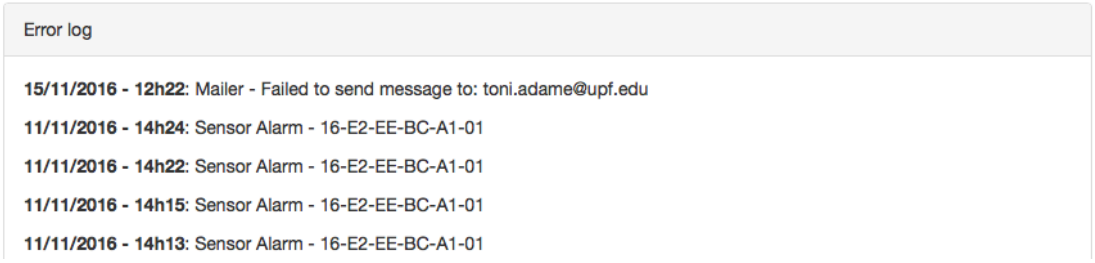

Figure 3.12: Error log of sensors not assigned to users.

3.6 CONFIGURATION

The main configuration page shows three main sections:

1. Alarm Type
2. Control Action Type
3. Measurement Type

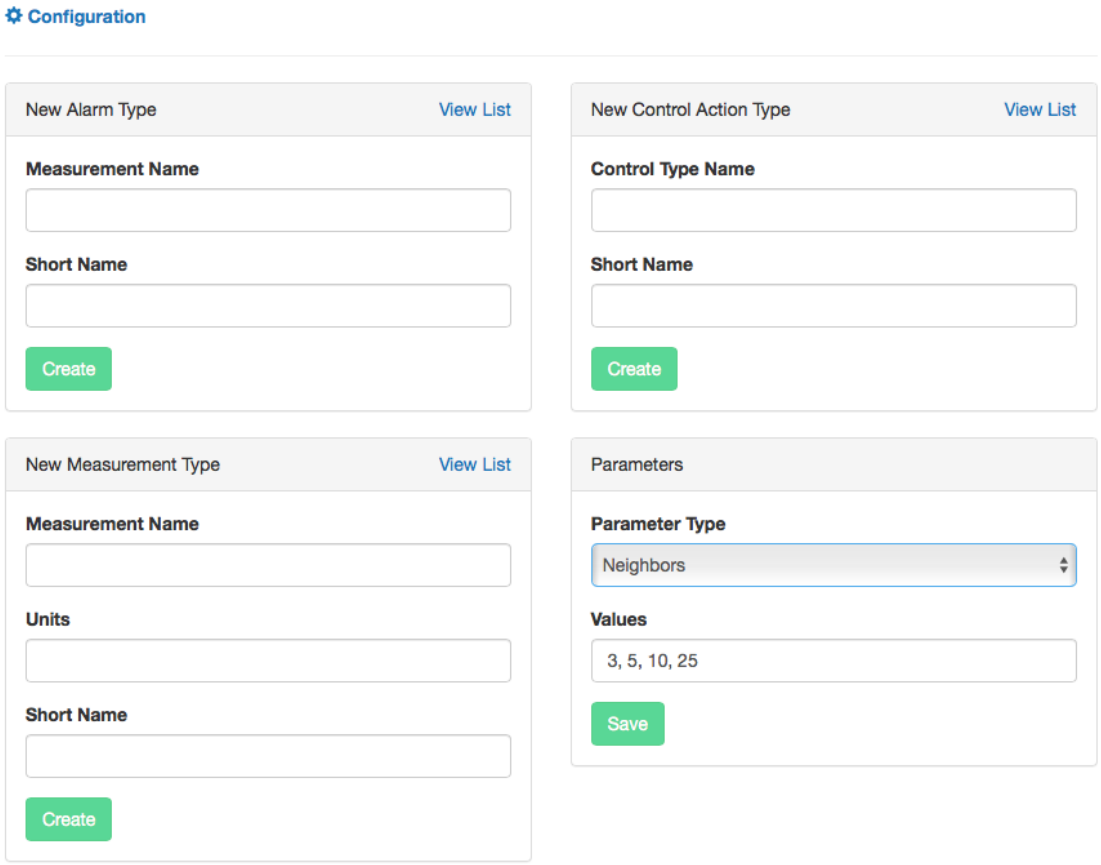

Figure 3.13: The main configuration page

In the Parameters-section, a parameter can be selected from the drop-down list and a value or multiple values can be given to that parameter (Figure 3.13). When a parameter requires multiple values, they are separated by a comma. Changing these parameters will alter the behaviour of the web platform, so that it will now show different classes of legend items in the map.

For the different list-sections, new items can be quickly generated by entering values (for e.g. name and short name) and clicking the “Create” button. Existing list items can also be viewed by clicking on “View List”. After doing so, the list’s configuration page is opened and one can activate or deactivate certain list items (Figure 3.14). Measurement types will not be visible to the users when deactivated but administrators, supervisors and super users will notice through the error-log on the Network Performance page when values are received for a deactivated measurement type.

All Types | Download table as CSV

Name	Short Name	Units	
B.ole	bol	amount	Deactivate
B.oleae	fly	Amount	Activate
BatteryLevel	batt	%	Activate
Humidity	Hum	%	Activate
Insects	insects	Amount	Activate
Luminance	lum	%	Activate
Olive Fly	ofly	Number	Deactivate
Temperature	temp	°C	Activate

'Short Name'

Figure 3.14: The measurement type list's configuration type

4 GPRS COMMUNICATION PROTOCOLS

The current version of the ENTOMATIC system has two possible ways to reach the data receiver server of the Monitoring and Management Central depending on its availability of having equipped a GPRS module:

1. **By means of a single GPRS link (between an embedded GPRS and the webserver)**

   Direct connection of the trap by means of a GPRS module embedded in the trap itself. By using this method, any trap is equipped with all the required electronics to perform a GPRS connection to the webserver. No communication between traps is required.

2. **By means of the ENTOMATIC WSN and a GPRS link (between a gateway and the webserver)**

   The Zolertia RE-Mote gateway centralizes gathered data from various traps and sends the information by a GPRS module. The gateway is the single element of the deployed network on the field that is equipped with all the required electronics to perform a GPRS connection to the webserver. Consequently, traps must not be equipped with a GPRS module and their gathered data is transmitted via 868 MHz links to the gateway, which will proceed to its retransmission to the webserver.

In any of the two aforementioned approaches, where a GPRS link is always necessary to perform the data transmissions, the communication with the data receiver server of the Monitoring and Management Central platform is based on GET requests encapsulated within the HTTP protocol, through a request-response approach between a client (the GPRS module) and a server (the ENTOMATIC data receiver server).

The GET method is used to retrieve data from a web server by specifying parameters in the URL part of the request. It is the main method for retrieving data from a web server. The structure of a request via GET method is defined as follows,

```
/test/demo_form.asp?name1=value1&name2=value2
```

where the URL is specified before the '?' character and next are attached the set of pairs of variable names and their values.

As for the ENTOMATIC system, whenever a GPRS module has any messages to transmit to the data receiver server, it will make a GET request which will include the parameters of the preconfigured actions as well as the corresponding data (see Figure 4.1).

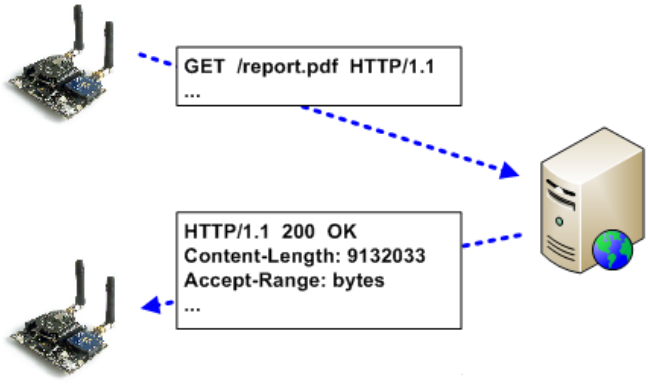

Figure 4.1: Operation diagram of a GET method request and response

4.1 SINGLE GPRS LINK BETWEEN TRAP AND WEBSERVER

Four different GET methods have been defined to fulfil the system requirements: ***newGateway***, ***newSensor***, ***newMeasurement***, and ***newAlarm***.

The developed protocol is defined in the following subsections, and as any GET method is based in two differentiated parts: the HTTP request and the Server response.

4.1.1 NEWGATEWAY

The ***newgateway*** method is responsible for registering a new gateway at the data receiver server. When a new gateway is connected, it is automatically registered at the server to be able to visualize all the data that the traps deployed on the field are collecting.

- **HTTP request:** The gateway sends an HTTP message containing the following information.
    - **idgw:** MAC address of the gateway.
    - **lat:** Latitude coordinate of the gateway.
    - **lon:** Longitude coordinate of the gateway.
- **Server response:** The server has to possible answers.
    - **OK:** The server answers with a 0, plus the current server time to synchronize the gateway clock and the MAC address of the gateway.
    - **NO OK:** The server answers with a 1, plus the MAC address of the gateway. The gateway is not registered and it is necessary to reset the device.

4.1.2 NEWSENSOR

The ***newsensor*** method is responsible for registering a new trap at the data receiver server. When a new trap is detected by the gateway, it is automatically registered at the server to be able to visualize all the data that it is collecting.

- **HTTP request:** The gateway sends an HTTP message containing the following information.
    - **idgw:** MAC address of the gateway.
    - **id:** MAC address of the sensor.
    - **lat:** Latitude coordinate of the gateway.
    - **lon:** longitude coordinate of the gateway.
- **Server response:** The server has to possible answers.
    - **OK:** The server answer with a 0, plus the MAC address of the sensor, the web address of the sensor provided by the system and the current server time.
    - **NO OK:** The server answers with a 1, plus the MAC address of the sensor. The sensor is not registered and it is necessary to reset the device.

4.1.3 NEWMEASUREMENT

The ***newmeasurement*** method introduces new data collected by the traps connected to a registered gateway.

- **HTTP request:** The gateway sends an HTTP message containing the following information.
    - **counter:** Message counter to keep control of possible failures.
    - **insects:** Number of insects/events detected.
    - **fly:** Number of olive flies detected.
    - **temp:** Temperature in Celsius.
    - **hum:** Percentage of humidity.
    - **lum:** Percentage of luminance.

- **batt:** Percentage of battery.

- **Server response:** The server has to possible answers.
  - **OK:** The server answers with a 0 and the web address of the sensor provided by the system.
  - **NO OK:** The server answers with a 1 and the web address of the sensor. The measurement is not registered in the platform and the sensor discards its message.

4.1.4 NEWALARM

The ***newalarm*** method is responsible for informing the server of some possible problems that the network may suffer[6].

- **HTTP request:** The gateway sends an HTTP message containing the following information.
  - **sensor:** Web address of the sensor.
  - **alarm_type:** Shortname of the alarm.

- **Server response:** The server has to possible answers.
  - **OK:** The server answer with a 0 and the web address of the sensor provided by the system.
  - **NO OK:** The server answers with a 1 and the web address of the sensor. The alarm is not registered in the platform and the sensor discards its message.

4.2 ENTOMATIC WSN + GPRS LINK

The proposal of a new GPRS communication protocol between the ENTOMATIC gateways and the ENTOMATIC Monitoring and Management Central is mainly motivated by the low speed of GPRS transmissions. The different tests performed by UPF at the lab show the need of modifying the communication protocol as it has been observed that ***a single GPRS transmission can last up to 50 seconds*** (including the establishment of the connection and the exchange of data and acknowledgement information). Those tests are reported in ***Deliverable 4.2: Development and performance evaluation of the ENTOMATIC network*** as the GPRS communication protocol is part of the whole ENTOMATIC communication protocol designed and developed in ***WP4: Design & Development of Wireless Sensor Network***.

While this operation delay is not a major inconvenient when sending data from a single trap (like in the McPhail trap version developed by TEIC), it ***produces non-negligible delays when using a centralized approach, where a designated device like the gateway transmits via GPRS the information from all its associated traps. For instance, the case of the ENTOMATIC system, where up to 30 traps can be managed by a single gateway***. In addition, the energy consumption of the gateway is directly proportional to the time it remains active for sending or receiving data.

In the following lines, a new GPRS communication protocol between the Zolertia RE-Mote gateway device developed by UPF and the data receiver server of the Monitoring and Management Central managed by AVIA-GIS is proposed. At this point it is worth noting that ***this new protocol does not set out to replace the existing one used in those traps sending data directly to the data receiver server through GPRS connections (which is kept in use by TEIC), but offer an efficient solution for the ENTOMATIC traps that require of the existence of the ENTOMATIC gateway***.

Taking into account that the maximum payload that can be sent by the Zolertia RE-Mote gateway via GPRS is of 255 bytes, the main innovation of this communication protocol is to reduce the size of variables used to define the different data frames. As a secondary goal, in this second protocol, the ***timestamps provided by the server will be used to continuously synchronize the network with a real clock, and the name of data fields in the messages has been homogenized***.

[6] These alarms must be previously defined in the webpage according to thresholds applied on collected data.

4.2.1 NEW GATEWAY (GA)

With the same purpose than the ***newGateway*** method, the ***Ga*** method is proposed.

4.2.1.1 Ga HTTP request

The ***Ga*** method registers a new Gateway in the data receiver server and the data is the one shown in Table 4.1.

Field	Purpose	Maximum size (bytes)	Example
mg	MAC address of the Gateway	16	00124b0006160f60
la	Latitude coordinate of the Gateway	3 + 3 = 6	41.40
lo	Longitude coordinate of the Gateway	3 + 3 = 6	2.202

Table 4.1: Proposed *Ga* method request fields

An example of a ***Ga HTTP request*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take values like the ones of the column 'Example' from Table 4.1.

Example: http://test.entomatic.upf.edu/Ga?mg=⑯&la=③.③&lo=③.③

4.2.1.2 Ga Server response

The server responds, in turn, with the following data frame as the example inside Table 4.2. Two different responses can be emitted: OK, or No OK. In case a "No OK" is received, the gateway is not registered and it is necessary to reset the device.

OK 0|timestamp|wg|
No OK 1|timestamp|

Field	Purpose	Maximum size (bytes)	Example
timestamp	Timestamp with the current server time: *Year/month/day/hour /minute/second*	14	20170126173655
wg	Web address of the Gateway provided by the server	5	103

Table 4.2: Proposed *Ga* method response fields

4.2.2 NEW SENSOR (SE)

With the same purpose than the ***newSensor*** method, the ***Se*** method is proposed. However, in this case, up to 4 different sensors can be registered with a single request. **It is worth noting here that the number of association payloads per HTTP request is variable, depending on the number of associated stations, but this will be always confined between 1 and 4.**

4.2.2.1 Se HTTP request

The ***Se*** method registers a new Sensor in the data receiver server. It consists of a common part and a variable number of association payloads (from 1 to 4) as shown in Table 4.3, whose value is defined by the new number (*n)* field. The different fields that the method has are summarized in Table 4.4.

Common part	Association payload STA #1	Association payload STA #2	Association payload STA #3	Association payload STA #4
46 bytes	45 bytes	45 bytes	45 bytes	45 bytes
226 bytes				

Table 4.3: Frame structure of a *Se* HTTP request

Field	Purpose	Maximum size (bytes)	Example
wg	Web address of the Gateway	5	103
n	Number of sensor requests	2	3
ms1, ms2, ms3, ms4	MAC address of each sensor	16	00124b000615ab18
la1, la2, la3, la4	Latitude coordinate of each sensor	3+3 = 6	41.40
lo1, lo2, lo3, lo4	Longitude coordinate of each sensor	3+3 = 6	2.202

Table 4.4: Proposed *Se* method request fields

An example of a ***Se HTTP request*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take values like the ones of the column 'Example' from Table 4.4.

Example: http://test.entomatic.upf.edu/Se?wg=⑤&n=②&ms1=⑯&la1=③.③&lo1=③.③

&ms2=⑯&la2=③.③&lo2=③.③&ms3=⑯&la3=③.③&lo3=③.③&ms4=⑯&la4=③.③&lo4=③.③

As stated in the frame structure of a Se HTTP request from Table 4.3, it consists of a common part and an association payload for each new STA associating to the network:

1. Common part format:
   http://test.entomatic.upf.edu/Se?wg=⑤&n=② ➔ 46 bytes
2. Association payload format:
   &ms1=⑯&la1=③.③&lo1=③.③ ➔ 45 bytes

4.2.2.2 Se Server response

The server responds, in turn, with the following data frame (see Table 4.5 for more details). Two different responses can be emitted: OK, or None OK. Those stations that have not been acknowledged in the response of the server will try to join the network again in the next beacon period.

OK 0|timestamp|n*|ws1|ws2|ws3|ws4|
None OK 1|timestamp|

Field	Purpose	Maximum size (bytes)	Example
timestamp	Current server time with the format: *Year/month/day/hour /minute/second*	14	20170126173655
n*	Number of accepted sensor association requests	2	3

ws1, ws2, ws3, ws4	Web address of each sensor provided by the server	5	103

Table 4.5: Proposed *Se* method response fields

If all or some stations could be appropriately registered in the platform, the server would respond to the request by filling the *n** field with the number of newly associated stations, and including in the *ws* fields attached next their allocated web address. For those that could not be associated, the value of their *ws* field would be 0.

Example 1:

If all stations were properly associated, the server would respond with:

0|timestamp|4|ws1|ws2|ws3|ws4|

being *ws1*, ws2, ws3 and *ws4* the values of their allocated web address.

Example 2:

If the server received a request of 4 associations, but it only could give an address to the first three ones, this would be the response:

0|timestamp|3|ws1|ws2|ws3|0|

being *ws1*, *ws2*, and *ws3* the values of their allocated web address.

Example 3:

If not a single association could be handled by the server, it would respond with:

1|timestamp|

4.2.3 NEW MEASUREMENT (ME)

With the same purpose than the ***newMeasurement*** method, the ***Me*** method is proposed. However, in this case, data from up to 3 different sensors can be accepted. **It is worth noting here that the number of data payloads per HTTP request is variable, depending on the number of associated stations, but this will be always confined between 1 and 3.**

4.2.3.1 Me HTTP request

The ***Me*** method introduces new information obtained from sensors in the data receiver server. It consists of a common part and a variable number of data payloads (from 1 to 3) as shown in Table 4.6, whose value is defined by the new number (*n)* field.

Common part	Data payload STA #1	Data payload STA #2	Data payload STA #3
46 bytes	**65 bytes**	**65 bytes**	**65 bytes**
241 bytes			

Table 4.6: Frame structure of a *Me* HTTP request

Field	Purpose	Maximum size (bytes)	Example
wg	Web address of the Gateway	5	103
n	Number of data payloads from different sensors	2	3
ws1, ws2, ws3	Web address of the sensor provided by the server	5	213
co1, co2, co3	Message counter to	3	3

	keep control of possible failures		
in1, in2, in3	Number of insects/events detected	3	10
fl1, fl2, fl3	Number of olive flies detected	3	2
te1, te2, te3	Temperature in °C	2+2 = 4	22.24
hu1, hu2, hu3	Humidity (%)	2	41
lu1, lu2, lu3	Luminance (%)	2	20
ba1, ba2, ba3	Battery level (%)	2	98

Table 4.7: Proposed *Me* method request fields

An example of a ***Me HTTP request*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take values like the ones of the column 'Example' from Table 4.7.

Example: http://test.entomatic.upf.edu/Me?wg=⑤&n=②&ws1=⑤&co1=③&in1=③&fl1=③&te1=②.②&hu1=②&lu1=②&ba1=②&ws2=⑤&co2=③&in2=③&fl2=③&te2=②.②&hu2=②&lu2=②&ba2=②&ws3=⑤&co3=③&in3=③&fl3=③&te3=②.②&hu3=②&lu3=②&ba3=②

As stated in the frame structure of a Me HTTP request from Table 4.6, it consists of a common part and a data payload for each new STA sending data:

1. Common part format:
   http://test.entomatic.upf.edu/Me?wg=⑤&n=② ➔ 46 bytes
2. Data payload format:
   &ws1=⑤&co1=③&in1=③&fl1=③&te1=②.②&hu1=②&lu1=②&ba1=② ➔ 65 bytes

4.2.3.2 Me Server response

The server responds, in turn, with the following data frame (see Table 4.8 for more details). Two different responses can be emitted: OK, or None OK. Those measurements not acknowledged by the server will be discarded by the gateway.

OK 0|timestamp|n*|ws1|ws2|ws3|
None OK 1|timestamp|

Field	Purpose	Maximum size (bytes)	Example
timestamp	Current server time with the format: *Year/month/day/hour /minute/second*	14	20170126173655
n*	Number of data payloads properly received	2	3
ws1, ws2, ws3	Web address of each sensor provided by the server	5	103

Table 4.8: Proposed *Me* method response fields

Similarly to the proposed Se method, if the server receives data payload from only a subset of stations, it will respond by filling the n* field with the number of received payloads. Data packets lost are signalled with 0.

Example 1:

In case all packets were properly received, the response of the server would be:

0|timestamp|3|ws1|ws2|ws3|

being *ws1, ws2,* and *ws3* the values of the stations' web address, whose measurements have been properly received.

Example 2:

If the server received a transmission of 3 data payloads, but it only could properly decodify the first and the last one, this would be its response:

0|timestamp|2|ws1|0|ws3|

being *ws1* and *ws3* the values of the stations' web address, whose measurements have been properly received.

Example 3:

If not a single data transmission could be handled by the server, it would respond with:

1|timestamp|

4.2.4 NEW ALARM (AL)

4.2.4.1 Al HTTP request

With the same purpose than the ***newAlarm*** method, the ***Al*** method is proposed. An example of an ***Al HTTP request*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take values like the ones of the column 'Example' from Table 4.9.

Example: http://test.entomatic.upf.edu/Al?ty=①&wg=⑤

Field	Purpose	Maximum size (bytes)	Example
ty	Type of alarm. Used to distinguish between different kind of alarms	1	2
wg	Web address of the Gateway	5	103

Table 4.9: Proposed *Al* method request fields

The type of alarm determines the value of the type (***ty***) field as well as the presence of possible additional fields (Table 4.10 compiles the codification of the ty field according to the related alarm). The alarms considered in the ENTOMATIC system are described in the following lines.

Value of type (*ty*) field	Purpose
1	Malfunctioning network
2	Malfunctioning device
3	Device running out of battery
4	High population of olive flies

Table 4.10: Summary of possible alarms depending on the type (*ty*) field value

4.2.4.2 Malfunctioning network

This alarm is set off if the gateway detects a high ratio of lost packets or instability in the association mechanism of stations. The web address of the gateway is also attached (*wg*).

An example of a ***malfunctioning network alarm*** is shown here, where the number within circle corresponds to the maximum size (in bytes) contained in each data field and in real operation takes the value of the web address of the gateway.

Example: http://test.entomatic.upf.edu/AI?ty=1&wg=⑤

4.2.4.3 Malfunctioning device

This alarm is set off if the gateway detects that an associated device is sending values out of normal operation thresholds. After the type (*ty*) field, the web address of the gateway (*wg*) and the one of the malfunctioning device (*ws*) will be attached.

An example of a ***malfunctioning device alarm*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take the value of the web address of the gateway and the web address of the malfunctioning device, respectively.

Example: http://test.entomatic.upf.edu/AI?ty=2&wg=⑤&ws=⑤

4.2.4.4 Device running out of battery

This alarm is set off if a device (one of the associated stations or the gateway itself) is running out of battery. Again, after the type (*ty*) field, the web address of the gateway (*wg*), the web address of the malfunctioning device (*wg* when being the gateway or *ws* when being an associated station) will be attached. Lastly, the battery level of the affected device will be also included.

An example of a ***device running out of battery alarm*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take the value of the web address of the gateway, the web address of the malfunctioning device, and the battery level of the affected device.

Example: http://test.entomatic.upf.edu/AI?ty=3&wg=⑤&wg/ws=⑤&ba=②

4.2.4.5 High population of olive flies

Lastly, if the population of olive flies detected by a station during the sensing period exceeds a pre-established threshold, an alarm is set off. After the type (*ty*) field, three fields are attached: the web address of the gateway (*wg*), the web address of the corresponding station (*ws*) and the number of olive flies detected (*fl*).

An example of a ***high population of olive flies alarm*** is shown here, where numbers within circles correspond to the maximum size (in bytes) contained in each data field and in real operation take the value of the web address of the gateway, the web address of the corresponding station, and the number of olive flies detected.

Example: http://test.entomatic.upf.edu/AI?ty=4&wg=⑤&ws=⑤&fl=②

4.2.5 AL SERVER RESPONSE

In all alarm cases, the server responds, in turn, with the following data frame (see Table 4.11 for more details). Two different responses can be emitted: OK, or No OK. Those alarms not acknowledged by the server will be discarded by the gateway.

OK	0|timestamp|
No OK	1|timestamp|

Field	Purpose	Maximum size (bytes)	Example
timestamp	Current server time with the format: *Year/month/day/hour /minute/second*	14	20170126173655

Table 4.11: Proposed *AI* method response fields

5 CONCLUSIONS

5.1 GPRS COMMUNICATION PROTOCOLS

Two different versions of the GPRS communication protocol to send data to the ENTOMATIC Monitoring and Management Central were designed and developed: one intended for communication through a single GPRS link between an isolated trap and the data receiver server, and an advanced one by means of the ENTOMATIC WSN and a GPRS link, where an intermediate gateway is responsible for gathering, aggregating and transmitting data from traps to the data receiver server also using a GPRS module.

Laboratory results using in the ENTOMATIC gateway the same GPRS communication protocol as in the traps (single GPRS link) demonstrated that the system was not able to offer enough reliability when a gateway aggregates data from different traps, due to time-constraints and hardware limitations inherent to the GPRS module.

For that reason, a second version of the GPRS communication protocol was designed and developed to be used in the ENTOMATIC gateway, where data and requests from multiple sensors / traps are aggregated to reduce the time the gateway is active transmitting or receiving data. The different request frames from the traps and the response frames from the server are in-depth described in this document, as well as some examples of GET requests/responses from each available functionality in the ENTOMATIC system.

5.2 UPDATE OF THE DATA SERVER AND WEB APP

Tests performed in a controlled lab environment not only revealed that the GPRS communication protocol between the gateway and the server should be redesigned due to hardware constraints of the GPRS module, but also the data visualization in the webserver dashboard had to be adapted to the information provided by the traps.

The experience achieved by different consortium members at the time of using the web app led to the introduction of some changes in order to benefit the web-user experience. Hence a new visualization of the data has been designed and developed. All these novelties have been uploaded and can be observed in http://test.entomatic.upf.edu.

The improvements introduced are summarized in the following points, and shown in Figure 5.1:

- A table of the available gateways is now shown. In this table users can modify the GPS coordinates of the gateway or remove it.
- An option to completely remove any sensor trap is also included.
- The collected data is now shown in a more visual format. All data reported by a sensor is shown in a single row and is classified by the reporting date.
- The location of the sensors or the gateway can be manually done by the user, introducing the latitude and the longitude of the gateway/sensor, thus facilitating this task and avoiding the installation of a GPS module in each trap, which would increase the final cost of the solution and only be used once to determine its initial position.

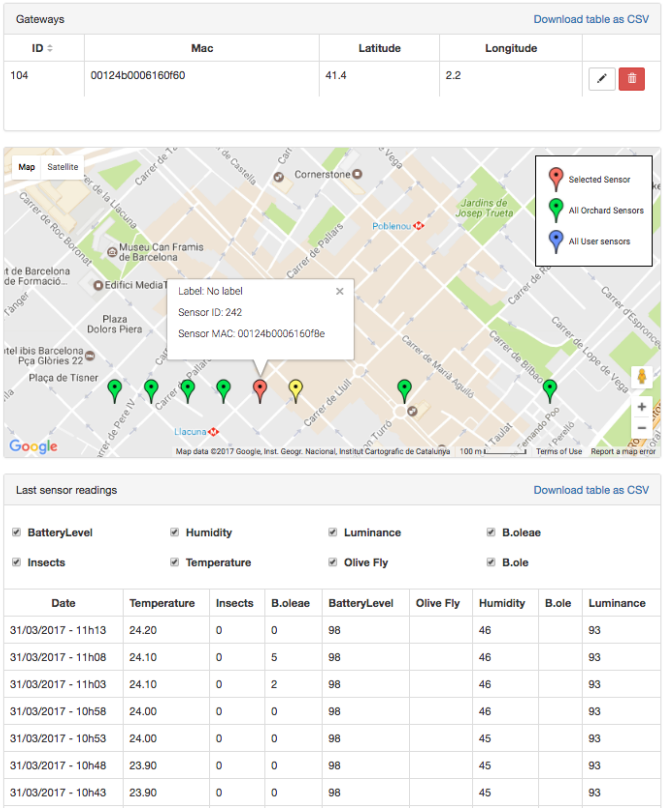

Figure 5.1: The new aspect of the web app with all the described new functionalities

5.3 PLATFORM VALIDATION

The current web app is completely available at test.entomatic.upf.edu after obtaining access to the platform via the creation of a new user by the administrators. First results of trap measurements have been uploaded and a first set of tests have been performed.

Main platform functionalities have been tested (namely; data transmission, store and visualization). We have not observed any problem with the data upload from the ENTOMATIC gateway via defined GPRS messages. Data available at the receiver server has been generated by the traps and has been correctly processed in the server. These tests have helped us to update the server according to performance observed. New modifications (see Section 5.2) have been included with regard to the first submitted version of the deliverable.

Moreover, the final set of tests will be performed during the final tests of the prototype manufactured by the end of 2016. These tests have suffered from some delay and will be performed during next olive fruit fly season. The final results obtained from those tests will be included in following reports. The validation of the whole system will be reported, as planned, in the ***WP8: Demonstration of the ENTOMATIC system***.

6 BIBLIOGRAPHY

[1] J.A. Pérez Cano. Web-based Wireless Sensor Network Platform. Technical report, 2015

[2] PostgreSQL, “The world's most advanced open source database,” [Online]. Available: http://www.postgresql.org/.

[3] GitHub, “Where software is built,” [Online]. Available: https://github.com/.

[4] AngularJS, “Superheroic JavaScript MVW Framework,” [Online]. Available: https://angularjs.org/.

[5] Node.js, “JavaScript runtime built on Chrome's V8 JavaScript engine,” [Online]. Available: https://nodejs.org/en/.

[6] Sails.js, “Realtime MVC Framework for Node.js,” [Online]. Available: http://sailsjs.org/.