Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Environmental Wireless Sensor Network

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Noelia Lostao Hervera

DIRECTOR: Manel Gasulla Forner

DATE: February 3 rd 2011

Títol: Environmental Wireless Sensor Network

Autor: Noelia Lostao Hervera

Director: Manel Gasulla Forner

Data: 3 de Febrer de 2010

Resum

'REALnet sensor network' és una xarxa de sensors sense fils que es troba a l'EPSC. Els sensors d'aquesta xarxa mesuren diferents magnituds com poden ser temperatura o voltatge. La informació mesurada pels sensors ha de ser recollida i mostrada a través d'Internet.

En aquest projecte, s'ha desenvolupat un lloc web on es pot trobar la informació dels sensors. Per fer aquest lloc web accessible des d'Internet, s'ha implementat un servidor web a l'EPSC. A més a més, el programa responsable d'obtenir la informació dels sensors i guardar-la en la base de dades, s'ha modificat per tal d'adaptar-lo als requeriments del projecte. Un cop, tot això ha estat desenvolupat, s'ha provat i testejat per tal d'assegurar que estava funcionant de manera correcta.

Title: Environmental Wireless Sensor Network

Author: Noelia Lostao Hervera

Director: Manel Gasulla Forner

Date: February, 3rd 2011

Overview

'REALnet sensor network' is a wireless network located in the EPSC. Sensors in the network measure different magnitudes as temperature or voltage. Sensors' measured data has to be collected and be shown in the Internet.

In this project it has been developed a website where the sensors information can be found. To make this website accessible from the Internet a Linux web server has been implemented in the EPSC. Besides, the program responsible for getting the information from the sensors and store it into a database, has been modified to adapt it to the project requirements. Once everything has been developed it has been tested to make sure it was working fine.

Gràcies a la meva família, per haver-me ajudat a arribar fins aquí. I gràcies a Dani per la seva ajuda i suport incondicionals.

INDEX

FIGURES INDEX

TABLES INDEX

INTRODUCTION

Nowadays, the Internet and the websites are becoming more and more important in our society. Most of the big companies have their own website which lets them be known all over the world.

When looking for some information in the network, it is not only important to find it, but also the way it is found. The appearance, usability and design are also becoming popular since people are getting more exigent day by day.

REALnet network is a wireless sensor network located in the EPSC (Escola Politècnica Superior de Castelldefels) and developed by the ISI (Instrumentation, sensors and interfaces group) group. Information coming from these sensors has been collected and shown in a simple website for some time. However the REALnet team needed to have a new website, where as well as finding the required information, they could manage its contents in an easy way.

This project aims not only to improve the web interface where the final user will find the sensors' information, but also to offer different functionalities to process the received data which have not been implemented so far.

All decisions when developing the project have been based on the specifications of the ISI group, thinking about their needs and also about the final user needs.

CHAPTER 1. GENERAL SCENARIO AND PROJECT STRUCTURE

The ISI group from the EPSC developed and designed an environmental wireless sensor network called REALnet. Nodes in this network, measure different environmental magnitudes (temperature, rain, solar radiation…) and they are located in different places in the 'Baix Llobregat Campus'. (1) Data from the sensors has to be collected and be shown in the Internet using a web browser http://realnet.upc.es/. In this project, it has been developed how to improve the collection of the data received from the sensors, how to process it and finally, how to show it by using a web browser. For this reason, the project has been divided in three different main sections. Each one explains the different goals that had to be achieved.

- Data Transmission
- Website
- Deployment in the EPSC server

1.1 General Scenario

The REALnet sensor network is composed by different elements. (2) On the one hand, it is composed by the sensors which measure different magnitudes. On the other hand it has different elements which allow these measurements to be stored in a database and afterwards shown in a website accessible from the Internet. These project aims to create a dynamic website where all the information coming from the sensors can be shown and managed by a final user. However, the project also aims to improve the way the measurements are stored in the database before being shown. A general idea of the general situation can be seen in Figure 1.1.
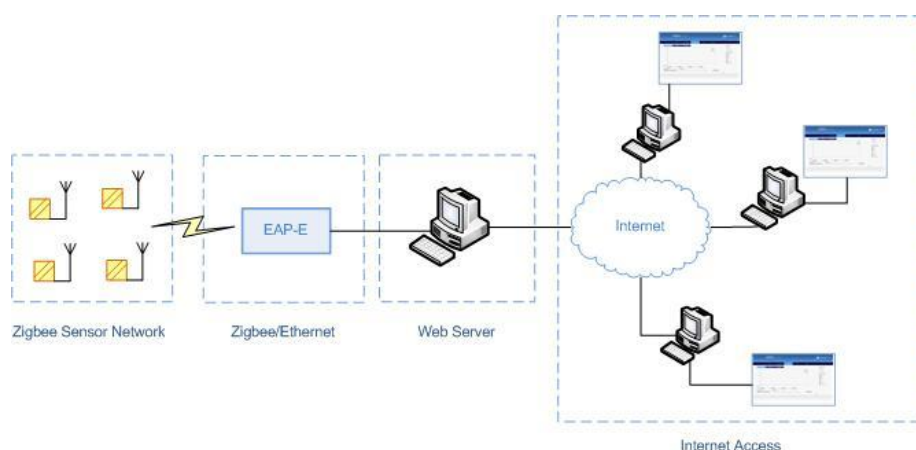
Figure 1.1 General Scenario

1.2 Project Structure

The sensor network is a Zigbee network. Through the EAP-E device from Telegesis, (3) measurements can be transmitted under Zigbee protocol and then transmitted to the database web server through the serial port by using Ethernet. So this ETRX-2 device lets bridging between Ethernet and Zigbee.

The way the measurements are transmitted to the database has been improved in this project and it will be explained in chapter CHAPTER 7. Once the information is in the database, it is shown by using a website. This will be explained in chapter CHAPTER 6. To allow users access to the website from the Internet a web server under Linux has been implemented in the EPSC. This implementation will be explained in chapter CHAPTER 8.

In chapters CHAPTER 2 and CHAPTER 4 the different technologies and tools used to develop the project are explained. And finally chapters CHAPTER 3 and CHAPTER 5 show the software patterns and the architecture model used.

CHAPTER 2. USED TECHNOLOGIES

Several technologies have been used in this project in order to satisfy the expected requirements. In this chapter each of these technologies are explained considering its main features. Most of these technologies belong to the Website part where most of them have used. It will also be considered how these technologies have been used in the project and how they have been structured. Technologies which are going to be commented below are: HTML, CSS, PHP, SQL and JavaScript.

2.1 HTML

2.1.1 HTML Definition

HTML (HyperText Markup Language), is the most used mark up language for creating web pages. (4) It is a hypertext and structured language composed by tags which show the beginning and the end of each part of the document. The browser displays the HTML documents as web pages and it uses these tags to interpret the content of the page.

Even HTML allows images and objects to be embedded and can be used to create interactive forms... It has some shortcomings that have been solved by using new technologies complementary to this one. Some of these technologies have been used in this project too, as PHP, JavaScript or CSS. (5) They will be explained deeply in this chapter and the improvements they have provided to HTML will be shown.

2.1.2 HTML in this project

HTML has been used mainly in the website part of this project even other technologies have been used too. As said, HTML is the most used language to create web pages. It has been essential in the project to create in general all the website part and in particular different elements which compose its framework.

Some remarkable features used are:

- Links

They have been responsible for redirecting from one page to another.

- Tables

Tables have been essential elements in order to display the information in a structured appearance.

- Forms

HTML forms have been used to collect information introduced by the user.

- Divs

A div defines a division or a section in the HTML document which has been significant when defining the different sections of the content of the page.

HTML and CSS are related to each other as they are responsible for showing the user the web page. HTML performs its structure and CSS performs the design. CSS is explained next.

2.2 CSS

2.2.1 CSS Definition

CSS (Cascading Style Sheet). This is a document format which provides a set of style rules which can then be incorporated in a HTML document. (6) CSS is used mainly so, to give 'style' to the HTML pages. HTML became limited when it was needed to give format to a document, then CSS appeared. The issue is to separate the content from formatting and presentation information. HTML is responsible for the content, while CSS style sheets define the format, colour, size, alignment... of this content. So, CSS is responsible for the way how the HTML page appears in the final user's screens.

CSS can be used to define the format of:

- An entire website
- An HTML document
- A part of the HTML document

CSS can be used in two different ways:

- Linking

A link in the head of the HTML page indicates which '.css' file has to be used. This link can be included in any document of the HTML page. Any change in the style sheet, applies to all the documents which have the link to that page.

- Embedding

The style sheet is not in a separated document but embedded in the HTML. However, this has some inconveniences as the code becomes heavier and longer... Besides, the code has to be included in all the HTML documents which

have to have that style and any modification in the style has to be changed in all the documents.

2.2.2 CSS in this project

CSS has been the essential element for the web design. With CSS it has been possible to give format to:

- Menus

The colour, size, position and some other features of the menus have been designed by CSS.

- Content

Background: CSS defines its colour, size…

Text: It can be specified the colour of the text, its size, the font family…

Position and size: It defines where the different sections of the document will be located. It can be specified the margin around the elements or the space between the element border and the element content. It is also used to determine the width and height of the section.

2.3 PHP

2.3.1 PHP Definition

PHP (Hypertext Preprocessor). It is a high-level interpreted language which can be included in the HTML pages. (7) There have been different PHP versions. From PHP 1.0 to the last stable one which is PHP 5.3.5. PHP is used mainly to make dynamic HTML pages in an easy way. One important feature is that PHP runs on a server on demand. It is one of the most common server languages on the web. It is a language of creation that has been growing in the webmaster community mainly due to the power and simplicity that it has.

PHP allows introducing its code into the HTML pages and performing different actions in an easy and effective way without having to program complicated code in a different language which cannot be included in HTML. So, PHP has improved significantly the power of HTML, for example considering the endless number of functions to operate with databases in a simple way with not a lot of complications.

2.3.2 PHP in this project

PHP is responsible for many scripts which perform different actions of the website part. PHP has also been used in the program responsible for inserting

the data received from the sensors into the database. PHP in this case got the information that came from the sensors and by using SQL it introduced the information into the required database.

As said, most of the actions that have been performed with PHP are related to SQL (Structured Query Language) sentences. In these cases, PHP works together with SQL. PHP sets the query to define the action and this query will have a SQL sentence to perform it. For example SQL will get information from the database and PHP will show it on the screen. But PHP has been also used to do other functions which are not related to the database. There are many of them, but the following ones could be defined as the most important ones in the project.

PHP no database actions
Create documents from user-entered text and show these documents on the screen.
Re-Direct from one page to another
Upload files to the server
Get files from the server
Create sessions for different kind of users
Open, validate and close session for each kind of user
Write HTML inside the PHP part of the HTML document

Table 2.1 PHP no database actions

2.4 SQL

2.4.1 SQL Definition

SQL is a standard language for accessing databases. Database management had been a complicated process, which was considerably rationalized by the SQL language. So, SQL is responsible for querying, controlling and editing information stored in a certain database management system. (8) SQL works by using sentences. There are three kinds of sentences considering their purpose.

- DDL (Data Description Language)

Language to define data. It includes orders to define, modify or delete tables where data is stored and the relations between them.

- DCL (Data Control Language)

Language to control data. It contains useful elements when working with protected data. It is used to control tables security and access restrictions or different ways of sharing information between users.

- DML (Data Manipulation Language)

Language of data manipulation. It allows getting stored data in the database and it also includes orders to let the user update the database adding new data, deleting old one, or modifying the stored one.

2.4.2 SQL in this project

PHP and SQL have been related very often in the project. Most of the PHP functions need information which is stored in the database. This information is requested by SQL sentences. But SQL is not only needed to get information from the database but also to introduce new data or remove some existing one that is not needed anymore. Data Manipulation Language sentences have been the most used ones. SQL sentences have had the main following functions:

SQL database actions
Select information from the database by using defined criteria
Insert information into the database
Delete information already existing in the database based on defined criteria
Modify information already existing in the database based on defined criteria

Table 2.2 SQL database actions

2.5 JavaScript, Prototype and Google Maps API

JavaScript, Prototype and Google Maps API (Application programming interface) are explained in the same section of the document as they are closely related to each other. JavaScript could be defined as the technology. Prototype and Google Maps use this technology in two different ways. Prototype is a JavaScript framework which makes easier working with JavaScript. And Google Maps API is a programming interface based on JavaScript.

2.5.1 JavaScript Definition

JavaScript is an interpreted programming language that is used to make web pages interactive. It has many possibilities and it is used to create small programs which are then inserted into a web page, and larger programs, object-oriented much more complex. While HTML is designed for defining static web page content, JavaScript is a programming language designed for performing dynamic tasks. (9) JavaScript is supported by all modern web browsers without requiring users to install new components on the system. It works from the client side. With client-side validation it will not be needed a whole page reload when executing a JavaScript functionality. (10)

JavaScript can be in the same file as the HTML but scripts will be more easily reused on multiple pages of the website if they are placed in separate files. In the same way as commented in CSS, it is just only needed to add the '.js' file in the HTML page. In this case by using a '<script>' tag. However, JavaScript can be also embedded directly into HTML pages

2.5.2 JavaScript in this project

JavaScript is a technology that has been used mainly to add functionalities on real time to the created application. As it is a client technology, it can be use to validate user actions for example, without the need of recharging the web page.

It has been used mainly when the user has clicked on a button of the web page and it requires an action that has to be confirmed. This confirmation pops up thanks to a JavaScript script which appears in the user screen without the need of recharging the page. Many of the actions in JavaScript are related to this. One example is when the administrator user wants to delete a user in the web page. After clicking on the button to delete, he or she receives a message automatically asking for confirmation to delete the user. This message has appeared so, directly in the user's screen through JavaScript. An example of the pop message can be seen in Figure 2.1:
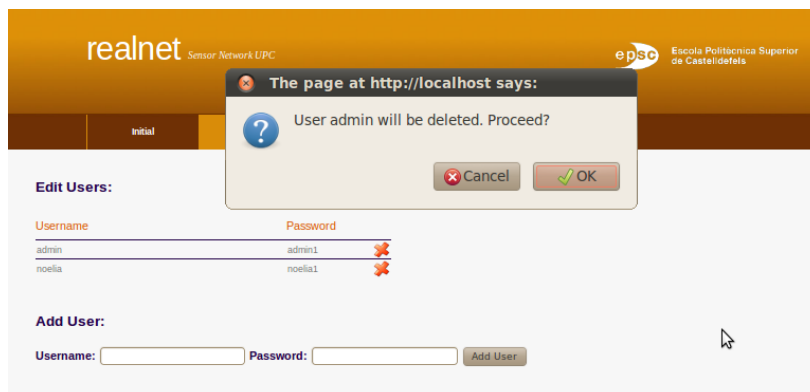

Figure 2.1 JavaScript confirmation message to delete user

Another application of JavaScript in this project appears in the sensors part, in the 'Temporal view' section where the data from the sensors is shown as a graphic. This graphic has many options that the user can change on line without the need of waiting to the graphic to be drawn again totally. Only by clicking on the option desired button the change is automatically performed. All these options will be explained in chapter CHAPTER 6. They can be seen in the right side of Figure 2.2.

Figure 2.2 Options in the graphic section

2.5.3 Prototype definition

Prototype is a JavaScript framework that aims to do an easier development of dynamic web applications. A framework is a conceptual structure and a defined support technology, usually with specific software modules, in which another software project can be based on. (11) Prototype provides enumerable object which implements number of useful operations to enhance JavaScript. By using prototype, some JavaScript functions become easier.

2.5.3.1 Protochart

ProtoChart is an opensource library that uses Prototype to create good looking charts. ProtoChart aims to be highly customizable and easy to use library. It allows the user create graphics adjusted to the needed requirements. (12)

2.5.4 Prototype in this project

ProtoChart has been essential in the graphic development of the charts in the Temporal View. Graphics have been made by using ProtoChart and it has helped to get attractive charts where information about the sensors is shown. It has been used to get:

- Lines Charts
- Points Charts
- Lines & Points Charts
- Area Charts
- Time Charts

It has also been essential in the design of these charts considering their colour, size... and many other features that will be shown in chapter CHAPTER 6.

2.5.5 Google Maps API definition

The Google Maps API gives developers several ways of embedding Google Maps into web pages, and allows for either simple use or extensive customization. (13) This API use as its base JavaScript. The API provides a number of utilities for manipulating maps (just like on the http://maps.google.com web page) and adding content to the map through a variety of services, allowing users to create robust maps applications on their websites.

2.5.6 Google Maps in this project

A Google map has been used to locate sensors in a map placed in the EPSC campus. As well as seeing the map with the sensors represented on it, some other features like having information about them have been included.

In the part of the Google map, as commented, JavaScript has been used. To see an example of a real time JavaScript function, it can be seen that when changing the kind of map to be shown (Map, Satellite and Hybrid) a JavaScript listener is working, as once the user clicks on the button to select the map, it automatically changes to the chosen kind of view. Besides, when the user clicks on an icon in the map (representation of a sensor), information about this sensor automatically pops up into the map too. All these specifications of the maps in the website and their appearance will also be shown later in the chapter CHAPTER 6.

CHAPTER 3. SOFTWARE DESIGN PATTERN

When the project was started and when developing the website part, it was thought it could be useful to use a defined design pattern. Design patterns are the base on looking for solutions to common problems in software development and interfaces design or interaction. In this project, part of the structure has been based on the Model-View-Controller pattern.

3.1 The Model View Controller

Model-View-Controller (MVC) (14) is a software architecture style which isolates application data from the user interface and the logic application for the user in three different concepts. It can be said that it separates the modelling of the domain, the presentation, and the actions based on user input.
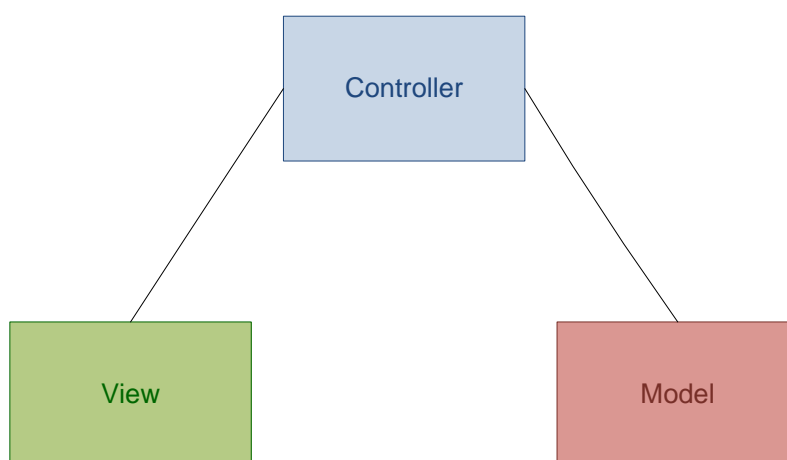
The diagram which represents this model is:

Figure 3.1 MVC diagram

- **The model** manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller). The model so, controls the view and its controller making easier complex visual presentations.

- **The view** shows the model in a format suitable to interact, usually the user interface.

- **The controller** responds to events, typically user actions, and initiates requests to the model and the view to perform actions based on that input.

Considering the different parts, it could be concluded that they have been implemented as follows in this project.
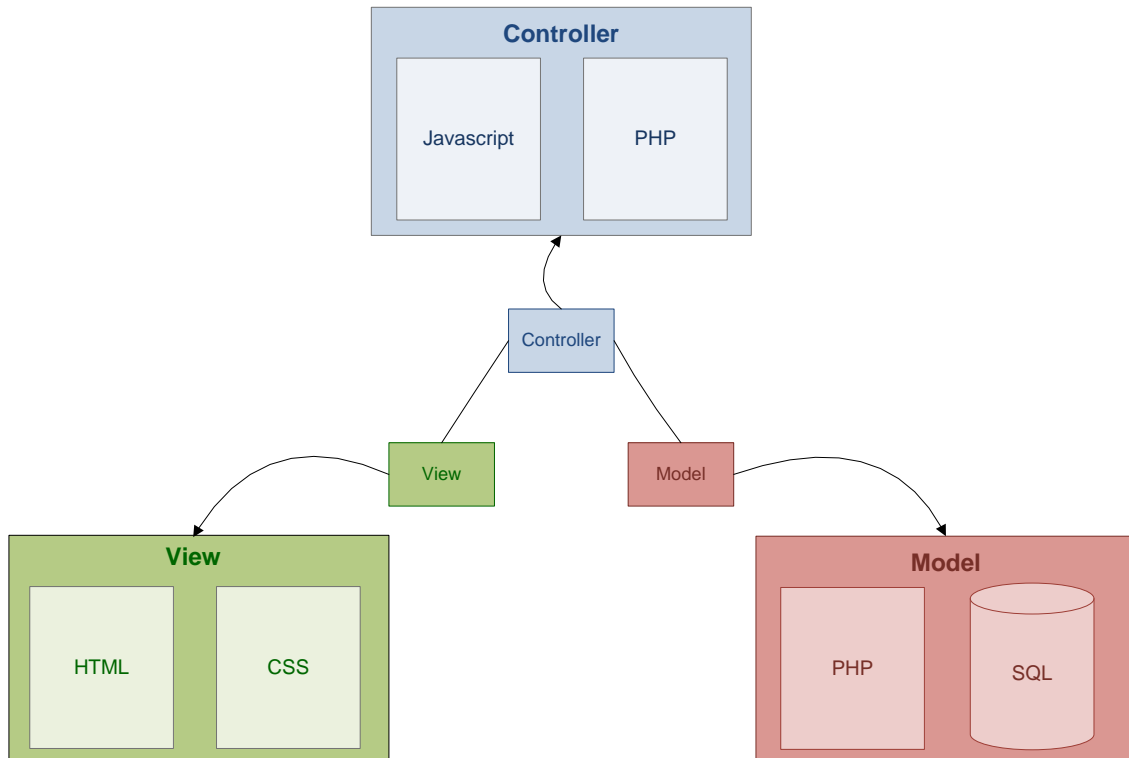

Figure 3.2 MVC in this project diagram

In the view part, the structure and the design part have been totally isolated. The HTML is responsible for the structure of the page which will appear in the user's browser, while CSS manages all the design features of the shown HTML page. By separating HTML from CSS it is possible to make modifications in the design and apply them only by changing the designing part. Besides, CSS can be defined to give format to different elements, or different pages. This separation means that, only by modifying one CSS instruction, many parts or even many HTML documents can be modified. In the same way, the HTML can be changed to get modifications into the structure of the document, but the design part can remain the same without any need of change it.

The controller part is carried out in sometimes by JavaScript, and also by PHP.

JavaScript interprets the mouse and keyboard inputs from the user, informing the model and the view to change as appropriate.

Regarding the model part in this project, it can be defined as the PHP functions that are activated from the controller (through JavaScript or trough PHP) part through the view layer. Access to the database is also granted by this part.

To understand the MVC applied to this project, some examples are included in annex A.

CHAPTER 4. USED TOOLS

Besides the technologies used to achieve the requirements of the project, also some tools to use these technologies have been needed. Some of them are mandatory when using the technology. Others only have made the work easier because they have been used as a framework. Tools used are different depending on the part of the project that it is considered. Two main stages can be differentiated regarding the use of tools. The first one when developing the project. The second one, when deploying the project in the university server.

4.1 Tools in the development of the project

The development of the project includes the part of programming the website and also the section of developing the solution to transmit the data from the sensors to the server.
When developing the project the scenario has been composed mainly by my workstation and all the tools that have been installed on it.

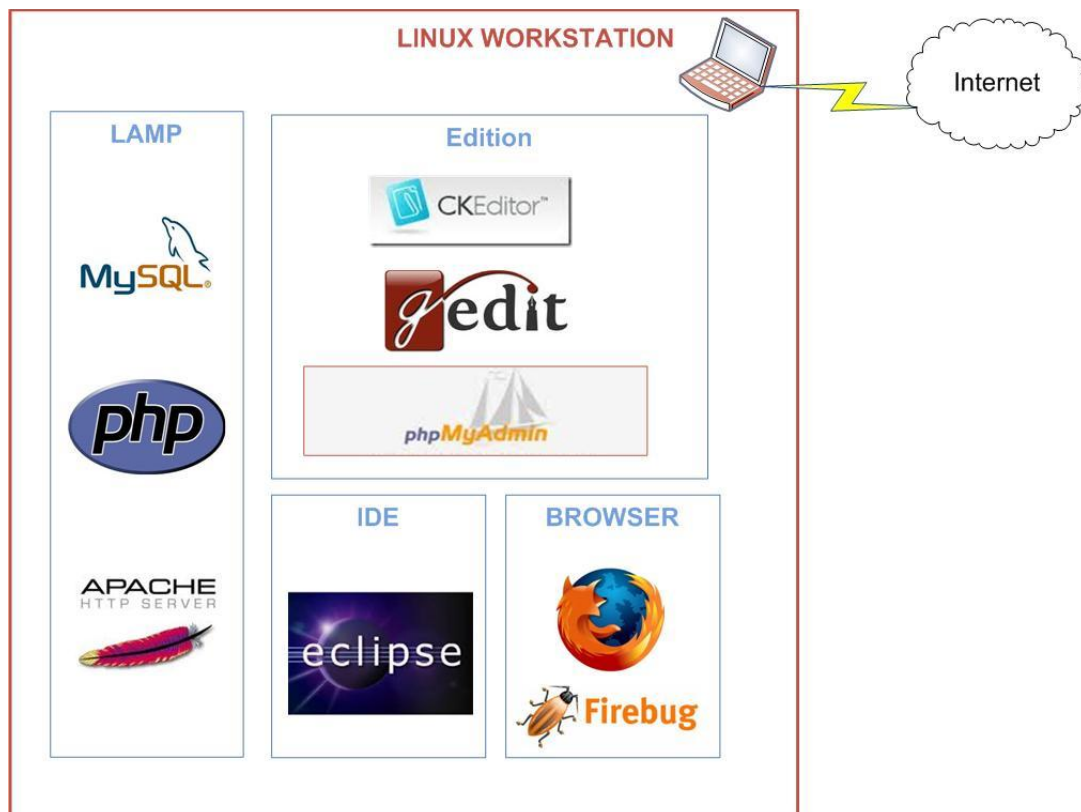To a better understanding of these tools, a diagram is shown in Figure 4.1:

Figure 4.1 Development tools

As commented, the workstation has been the base of all the development work. However, all the above shown tools have been essential to make easier every a development task.

4.1.1 Workstation

The workstation is the personal computer where all the tools have been installed. It has been connected to the Internet to get needed information. The operating system it uses is LINUX as it was one of the requirements of the project due to being free software.

4.1.2 LAMP

LAMP is composed by the first letters of **L**inux, **A**pache, **M**ySQL and **P**HP,Perl or Phyton. In this project this last, belongs to PHP. (15) LAMP is a package of open source software composed by the essential components to build a web server. Each element can be get separately as any of these programs was not designed to work specifically with each other. However, the development philosophy is shared and they were developed in close conjunction. When all the programs are used together, they form a solution stack of technologies that support application servers. So the package gives an effective solution as it is free of cost, and quite adaptable.

Each program has a different role.

- Linux is the **Operating System**
- Apache is the **Web Server**
- MySQL is the **Database Manager**
- PHP is the **Programming language**

4.1.2.1 Linux

Linux refers to the family of Unix-like computer operating systems using the Linux kernel. (16) Using Linux as operating system, was one the requirements of the project. The primary difference between Linux and many other popular contemporary operating systems is that the Linux kernel and other components are free and open source software. One of the advantages of Linux is its stability. It doesn't need to be rebooted periodically to maintain performance levels. It doesn't freeze up or slow down over time due to memory leaks... Another advantage is that it is one of the most secure operating systems nowadays. Maybe is not the most used operating system because becoming familiar with the Linux operating system requires patience as well as a strong learning curve.

The Linux distribution that has been used in the development part of the project is Ubuntu 9.10 – the Karmic Koala – released in October 2009 and supported until April 2011.

4.1.2.2 Apache

As commented Apache is the web server of the Project. (17) A web server can however refer to different things. When talking about a web server people may refer to the hardware, the computer or the software. In this case, Apache is the web server software. The application that helps to deliver the content accessed through the internet. Apache is developed and maintained by the developers under the supervision of the Apache Software Foundation. The application is available for a wide variety of operating systems, in our case Linux. It is characterized as open-source software.

Apache is used to serve static and dynamic content Web pages on the World Wide Web. In the case of the development of this project, a locally installed version of Apache has been used in order to preview and test code in the meanwhile it was developed. So the main feature of the software has been helping to build a local web server so that all the code and all the web pages developed could be tested, checked and visualized.

4.1.2.3 MySQL

MySQL is a relational database management system that runs as a server providing multi-user access to a number of databases. It must not be confused with SQL which is the language used to communicate with the databases. MySQL works on many different system platforms, in this case, as already said, Linux.

MySQL lets the user have control of the databases used for the developed project. (18) Sometimes, it is a bit annoying managing the databases and for this reason, there are several free graphical administration applications. They are available and they integrate with MySQL and enable users to work with database structure and data visually. It is a way of making easier the work of database management. Instead of using a console, (19) the user can create or modify information by a visual and an easier way.

Some well-known front ends are: phpMyAdmin, HeidiSQL, Adminer or DBEdit. PhpMyAdmin has been the chosen for this project. It is developed in PHP and requires the LAMP stack, so it was the one which best fitted with our features. It will be explained later.

4.1.2.4 PHP

PHP has been explained in the technologies part. It can be considered as a technology as it is a programming language but also a tool as it is part of the LAMP package application. The PHP interpreter is included when installing PHP so that all the *.php pages can be interpreted in the server side.

4.1.3 Eclipse

Eclipse is an IDE (Integrated Development Environment). (20) It is a multi language software development environment which can be used to develop applications in Java (Eclipse is most written in Java) and many other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ruby Scala, and Scheme. In this case, the most used languages have been PHP and JavaScript.

By using Eclipse some advantages have been achieved as it provides a consistent feature set on most platforms. And as it can work with different languages it has been possible to use Eclipse with all the programming languages used in the project. It is extensible and configurable so, it can be adapted to the user needs. While Apache has been used to test the web pages, Eclipse has been the main tool to develop them. That is, all the web pages which compose the project have been created using Eclipse. It has made easier the work as it helps the user with the language program. It gives the user suggestions, provides information about errors... In Figure 4.2 the main interface of Eclipse can be seen:
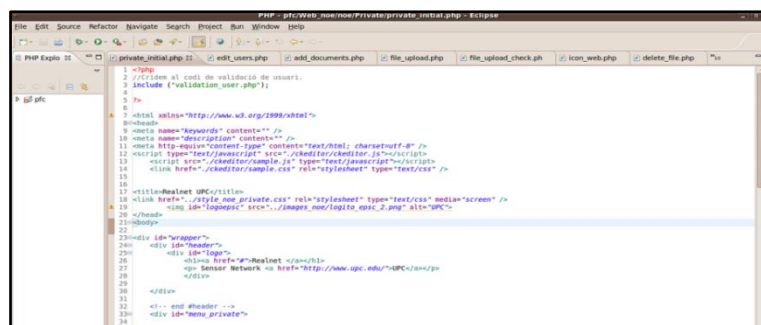
Figure 4.2 Eclipse interface

4.1.4 Mozilla Firefox

Mozilla Firefox is a free and open source web browser. It is managed by Mozilla Corporation. It is one of the most used browsers and it runs on various

operating systems including Microsoft Windows, Linux, Mac OS X, FreeBSD, and many other platforms. Firefox source code is free and open source software.

In this project it has been the browser used. As Linux has been used, not other common browsers were compatible and considering the ones which were, Firefox was the one which has more features to be chosen as its free license. After creating a web page, this was visualized in the web browser to see it as the user would see it and to check if all the requirements were achieved.

4.1.4.1 Firebug

Firebug is a plug-in for Firefox. (21) It allows the debugging, editing, and monitoring of any website's CSS, HTML and JavaScript, and provides other Web development tools. It puts a wealth of development tools at the user fingertips while he or she browses. As said so, can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page.

Firebug is also free and open source and it makes easier to the user the work of solving problems when the website does not look as expected or it has some errors. Firebug let the user debug all the problems he or she may have with CSS, HTML, and JavaScript…. Firebug appears as a complement in the part below the browser. It can be activated whenever the user wants.

Figure 4.3 Firebug appearance

4.1.5 PHP MyAdmin

PhpMyAdmin is a free software tool written in PHP intended to handle the administration of MySQL over the WWW. (22) PhpMyAdmin supports a wide range of operations with MySQL. The user interface lets do some actions as managing databases, tables, fields, relations, indexes, users, permissions, etc. This makes the user to have the ability to directly execute any SQL statement most of the times without even writing it. Sometimes SQL statements are very difficult to be written correctly and by using PhpMyAdmin, they become easier. The main MySQL feautures that can be achieved with PhpMyAdmin are

browse, create, copy and drop databases, tables, views, fields and indexes…
Besides this, data from databases can be exported to various formats: CSV,
SQL, XML, PDF… Also data can be imported from CSV or SQL files…

When writing a SQL statement, phpMyAdmin also provides the user information
about errors etc. which helps the user to know where he or she has the mistake.
So, even when the user is writing the SQL statement by himself it is better than
writing it in a common text editor or directly in the console. The most important
feature is so, making easier the database management to the user.

In this project it has been used for most of the database actions. When data
extracted from the sensors had to be included in the workstation where all the
development work has been done, all this data was imported from a SQL file by
using PhpMyAdmin. After that, all the database actions performed were done
also by using phpMyAdmin. With PhpMyAdmin, the control of the database of
the project as well as all the tables included in it, have been possible. In Figure
4.4 its appearance can be seen.
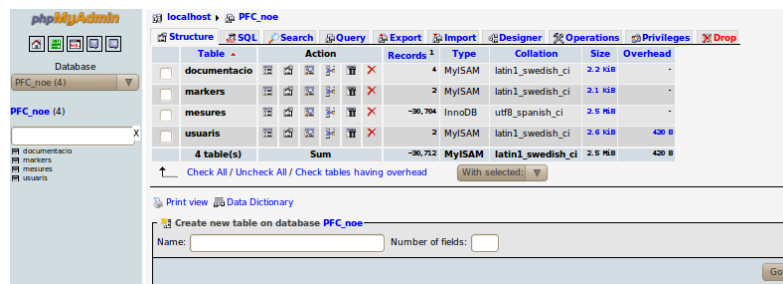
Figure 4.4 PhpMyAdmin appearance

4.1.6 Gedit

Gedit is the official text editor of the GNOME desktop environment. It is a very
complete text editor as it is configurable for various languages (C, C++, Java,
HTML, XML, Python, Perl and many others). Even in this project Eclipse has
been the essential tool of creating the web pages, sometimes when creating a
function or a small part of code, this was created by using Gedit and after that it
was included in the main code in the Eclipse interface. In Figure 4.5 Gedit
appearance can be seen.
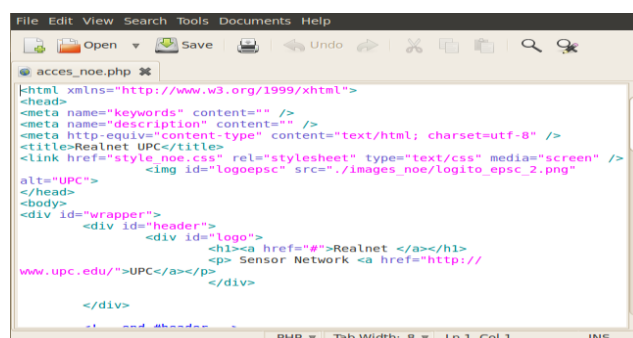
Figure 4.5 Gedit appearance

4.1.7 CK Editor

CKEditor is a text editor to be used inside web pages. It's a WYSIWYG (What you see is what you get) editor, which means that the text, images… being edited on this editor, look as similar as possible to the results that the user obtains when publishing it. (23) CKEditor gives the user the chance to edit texts with the common features that are usually found on desktop editing applications like for example the OpenOffice. Editing a text inside a website, without any editor help it is really complicated as it has to be written with HTML code. However these programs make this task for the users easier as they can work in the same way as they were editing a text in a desktop application.

As commented, the CK editor is to be used inside web pages. In this project it has been used in the Private Section, in the Home page edition section. The administrator of the website can change the content of the main page in the public part by using this editor. In Figure 4.6 it can be seen its appearance.
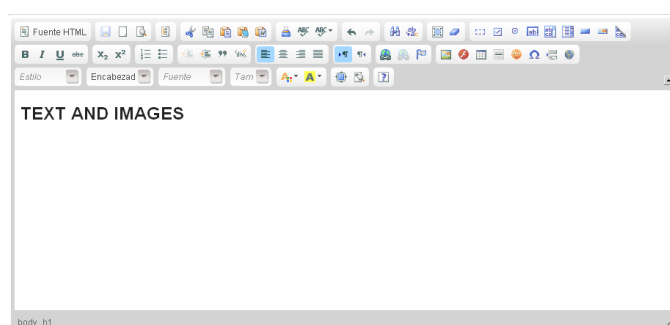
Figure 4.6 CK Editor appearance

4.2 Tools in the deployment of the project

The deployment part of the project includes the part of implementing all that has been created in the development part. This is, on the one hand, the implementation of the transmission of the data sent from the sensors to the server. On the other hand, the installation of the EPSC server to let the users access to the website through the Internet.

4.2.1 Data Transmission

Data transmission has been developed with PHP. Any additional tool has been needed as the program which performs the data transmission, it only needs a PHP interpreter and a database management system so that the information received can be inserted into the required database. Eclipse has been used to develop the code of the program and PhpMyAdmin has been essential when checking that the data was correctly transmitted and inserted to the database.

4.2.2 Implementation on the EPSC server

As commented, the deployment in the EPSC server has been prepared on the development workstation and then applied into the University computer where the server was going to be running.

The scenario used to implement the EPSC server can be shown below:
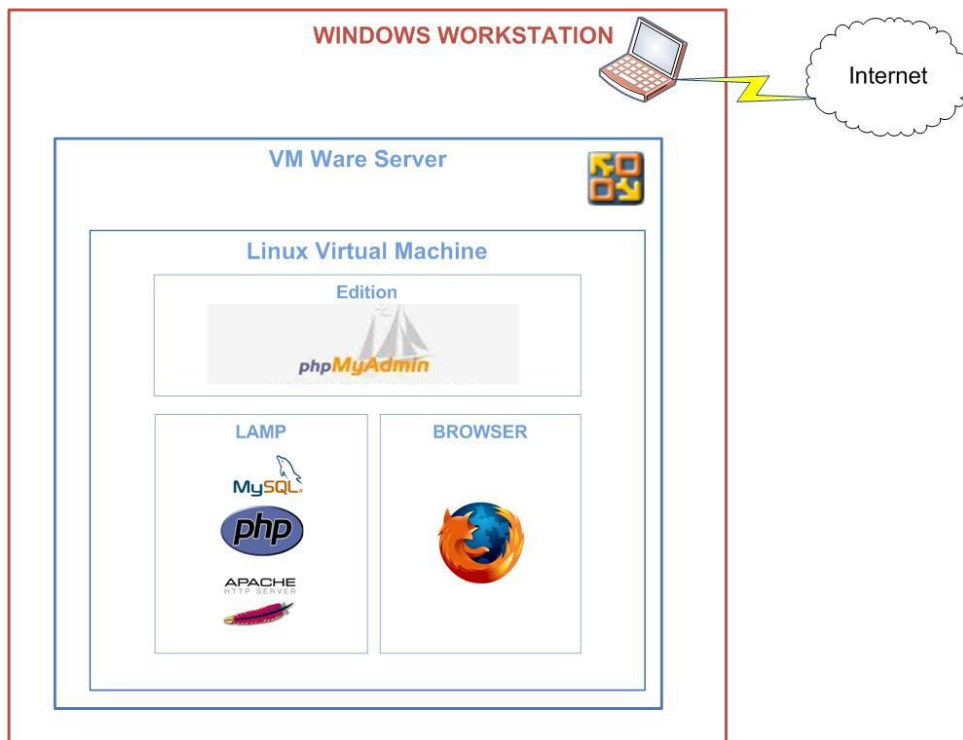
Figure 4.7 Deployment scenario

Despite in the development part of the project, the workstation used Linux as operating system; in this part the workstation worked with Windows. The reason is that the workstation had to simulate the computer in the EPSC which uses Windows as operating system and where a Linux virtual machine had to be installed. So, as it can be seen in the image, a Linux virtual machine using VMware has been installed in the workstation. There, all the needed tools to implement a web server have been installed too.

Some of the tools have been deeply explained in the previous point ('Tools in the development of the project'), so only the new ones will be explained here.

4.2.2.1 VMWARE

VMware Server is a free virtualization software server suite. There is also VMware software available for purchase; however VMware server can create, edit, and play virtual machines which were all our needs.

VMware Server so, is a hosted virtualization platform that installs like an application on any existing server hardware and partitions a physical server into multiple virtual machines. In this case, only a virtual machine is used. VMware server works with Windows, Solaris, Linux and NetWare, any or all of which can be used concurrently on the same hardware. In this project, as already said, the virtual machine works with Windows simulating a Linux machine. In Figure 4.8 an image of VMware can be seen.
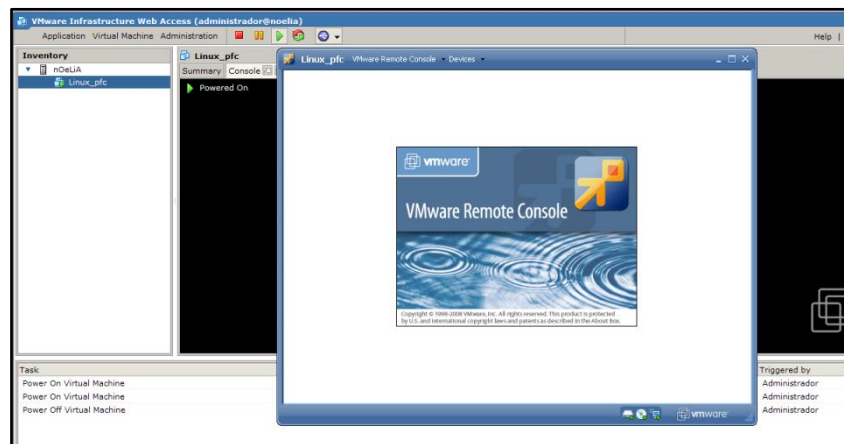
Figure 4.8 VMware appearance

4.2.2.2 LINUX

Despite Linux has already been commented before; it is also commented in this section as the Linux distribution used is a different one.

In this case, it has been used Ubuntu too, but a different version. Ubuntu 10.4 LTS –The Lucid Lynx- released in April 2010 and supported until April 2013.

This version has been chosen as is the last Long term support one which guarantees that will be supported longer. It does not mean that after April 2013 it cannot be used. It only means that there will be updates and solutions to some problems that can appear until then, which is more than the Ubuntu desktop editions which normally have one year and a half of support. LTS has three years.

CHAPTER 5. CLIENT/SERVER MODEL

5.1 Client-Server architecture

The project has been based on client-server architecture. (24) The client–server model is a distributed application structure that divides tasks between the providers of a resource or service, called servers, and service requesters, called clients. In our case the client is the user who is accessing the website and the server is the computer in the EPSC which is hosting the web and showing it to the client.

A diagram of the client-server architecture can be seen below:
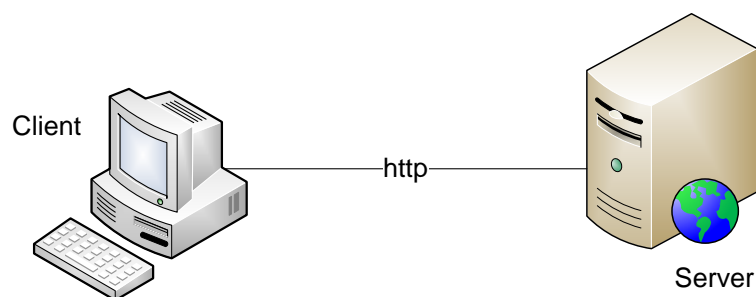
Figure 5.1 Client-server architecture

5.2 Technologies and tools in the client-server model

Technologies and tools commented above work in different sides of the model. They can be classified depending if they work in the server side or in the client side.

The browser is capable of interpreting HTTP (Hypertext Transfer Protocol) (25) orders received, and then transfer the HTML web page the final user sees.

When a HTTP request is done from the client side, asking for an HTML file, the server sends it and the browser shows it in the client side. This happens when the file is an HTML file. Other languages cannot be understood by the browser directly and they have to be interpreted in the server side. Once the code is interpreted, the server sends it to the client as HTML code. This is the case, for instance of PHP code that must be interpreted in the server side before it is sent to the client.

There are server side languages which are known and interpreted in the server side and then sent to the client as HTML code. And also about client side languages as HTML or JavaScript that do not need to be interpreted by the server as the client can understand them directly.

A schematic about the performance of this architecture is shown below:
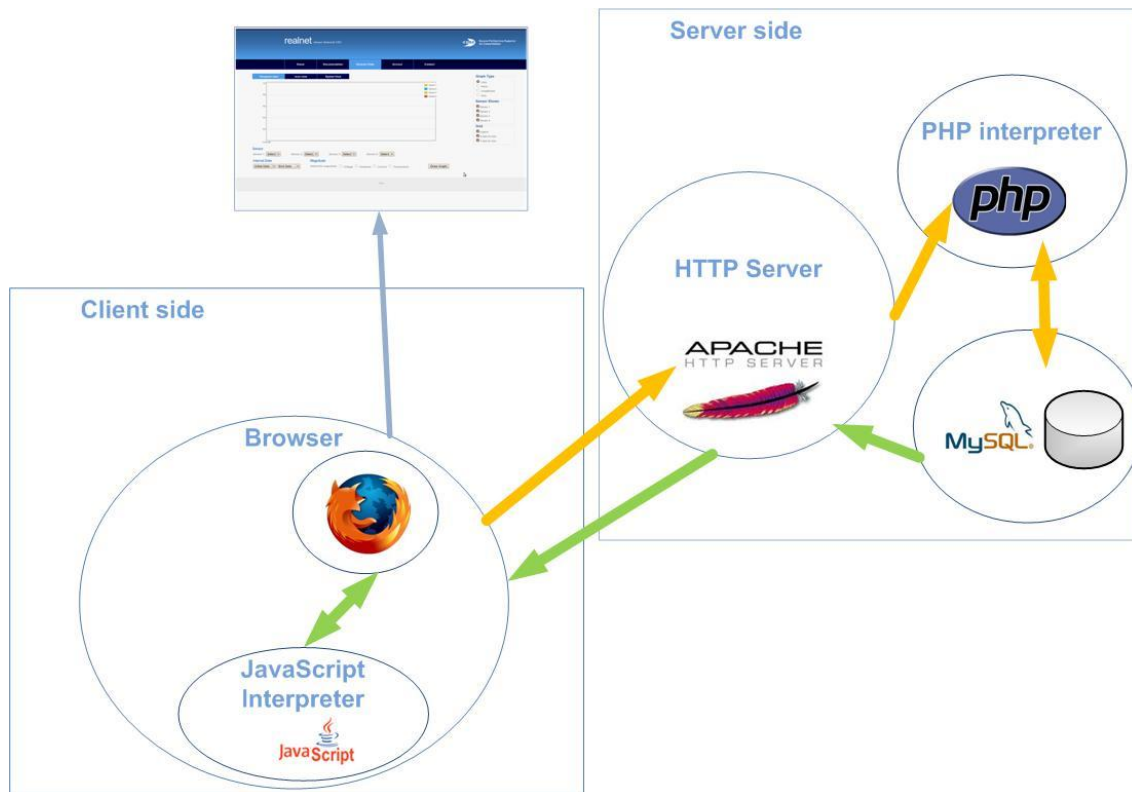
Figure 5.2 Client-server / Technologies and tools

As it can be seen, in Figure 5.2 the browser asks for a web page to the HTTP server. When this page contains PHP code, the HTTP server calls the PHP interpreter. The PHP interpreter executes the PHP scripts which sometimes need to access to the database. Once they are executed, it returns the generated document to the HTTP server. Then the HTTP server sends the document in HTML format so that it can be interpreted by the browser. When the document is on the client, it is interpreted, scripts are executed by the JavaScript Interpreter and the result is shown in the user's screen.

This is how the project works at the end, when everything is installed in the EPSEC server and the users access it from different sides of the network.

As commented, when developing the project, the scenario was not exactly this one. The difference was that both the client and the HTTP server were located in the same machine, the workstation where the entire project has been developed.

CHAPTER 6. REALNET WEBSITE

The REALnet website could be defined as the most visual part of the project. While the data transmission and the deployment in the EPSC server have not a visual impact on the final user, the REALnet website has. All the parts are needed to give the final user a result, but it is clear that this section, is responsible for attracting the attention of the user and give him or her the required information in an easy way. All the three parts have meant a lot of work; however the heaviest part of the project belongs to the REALnet Website. This is not also for the number of hours required to do it, but also for the investigation, and the previous work that had to be done to understand all the technologies that were going to be involved in the project and how to work with all of them together. Decisions about the design, colours, or even decisions about of the size of many elements of the page have been also important when deciding how to show the final result.

One of the requirements of the website was that it has to be composed by two different sections. On the one hand a public one accessible for everyone where it was possible to find information from the REALnet team, like documents related to them or their work. On the other hand a private one only accessible for administrator users where the administrator of the network could manage some contents of the public part. Even both, the public and the private sections of the website part had their own importance, one of the most important point or one of the main interests of the project belonged to the sensors section in the public part. There, the stored information from the sensors had to be shown in three different ways. Both parts and also all the sections of each part will be explained in this chapter.

6.1 Global Structure

To introduce the website a diagram of its global structure will be shown. Afterwards each section will be explained more deeply. As commented there are two main parts, the Public part and the Private part. Each of them has different sections that can be seen in Figure 6.1.
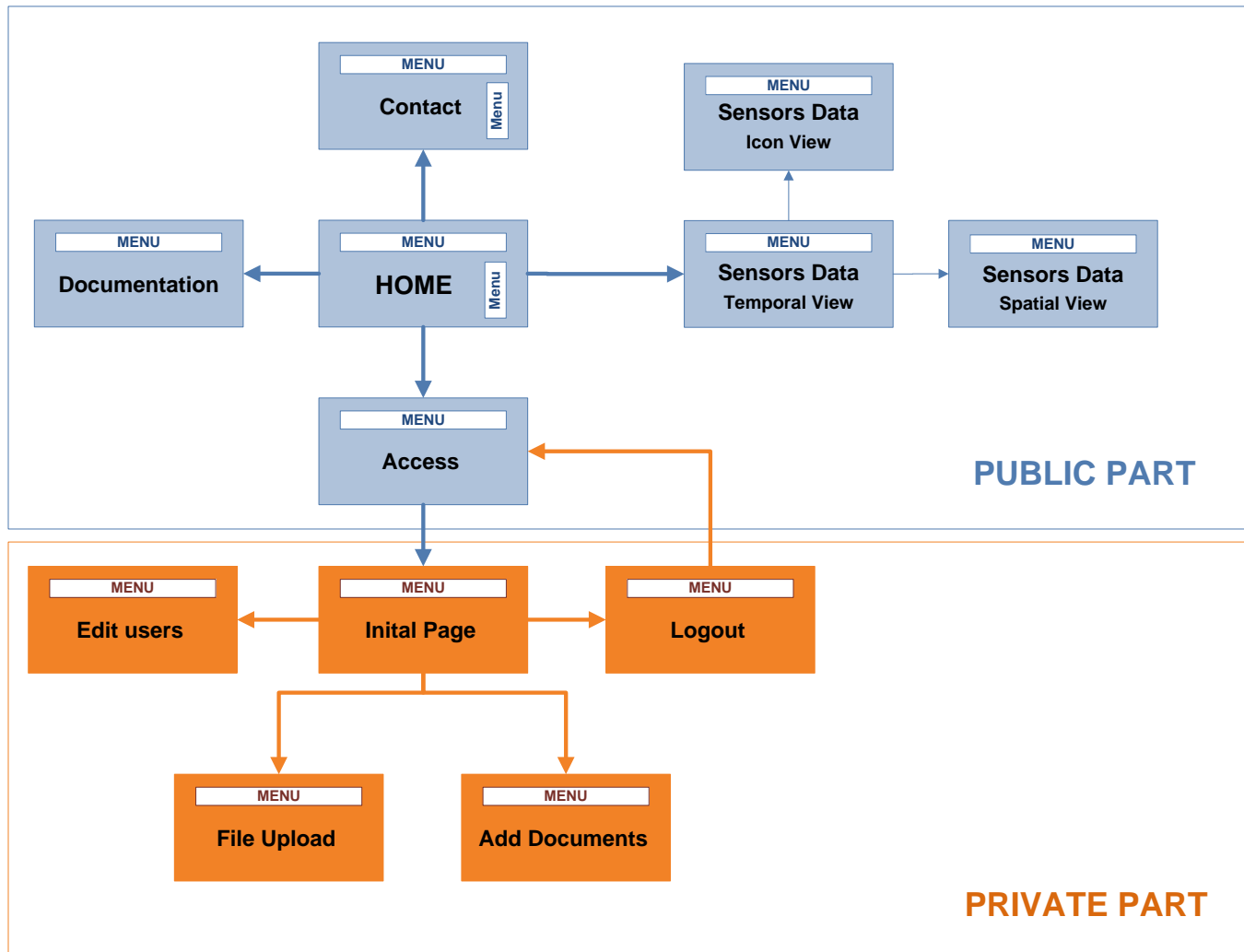
Figure 6.1 Diagram of the website

Both parts can be differentiated in the picture above. On the one hand, it can be seen the 'Public Part' where there are seven pages. The main one and also the default page when entering 'http://www.realnet.upc.es' is the 'Home' page. From the 'Home' page, the user can navigate to some other pages of the public part which are: 'Documentation', 'Access', 'Sensors Data' and 'Contact'.

When the user is on the 'Sensors Data' page, the default sensor view is the temporal one (Temporal view). However, from this page the user can go to two other possible views. The 'Icon View' and the 'Spatial View'. These are the three different ways of showing the information of the sensors that were commented above. It is important to say that, from each one of these pages, the user can navigate to all the other ones in the public part. However, not all these possibilities have been specified in the draw to avoid having a complicated diagram.

Private part is accessed by the user from the 'Access' page by introducing the appropriate user and password. The private part has some other pages accessible from the 'Initial Page' which are the 'Edit users', 'File Upload' and 'Add documents'. The logout page is not specifically a page because by doing click on the Logout menu, the user is redirected to the 'Access' page of the Public part and consequently he or she is logged out.

In the diagram different colours have been used to distinguish the public and the private part. It is also like this in the web page. Menus for the public page have different colour from menus from the public page. This lets the user know if he is logged or not.

6.2 Public Part

The public part had to be the website accessible for everybody where information from the REALnet sensor network could be found. Not only data from the sensors, (even this was one of the most important points) but also information about the REALnet team, useful documentation related to them or to any other remarkable and useful issues. It had to have also a gateway to the private part where the administrators users could introduce their user and password to get into it. Considering all these points, the REALnet public website part was developed to get a final result. All the pages of the public part of the website will be explained now. Images of each page can be found in annex B.

6.2.1 Home page

The home page is the default page when the user enters into 'http://www.realnet.upc.edu'. It could be shown as the default view of the website. It is composed by two menus. One in the top of the page, where the user can choose between the different sections of the website. Another one in the right side of the page, where there are different links related to the UPC.

In the Home page a 'Welcome message' with some information about REALnet can be seen by the final user. This information can be changed from the private part, but this will be explained in the private section.

6.2.2 Documentation Page

This is the page where the user can find documents related to REALnet and its main topics.   There are two sections in this page. The first one is related to documents that are on the Internet. In this page, only the link to the documents is shown. The document is not available in the website to be downloaded. The user is only redirected to the place where the document is stored and can be found. For instance, when looking for a final thesis, the user is re-directed to the corresponding university website. In the second section the user can download documents directly from the REALnet website.

6.2.2.1 Internet Links

Considering the first section related to the documents that are on the Internet, all documents are shown in the default view. The user can however, select only one kind of document to be shown.

6.2.2.2 Download files

The second section contains the files that can be downloaded from the REALnet page. All documents appear and they can be downloaded by clicking on them.

6.2.3 Sensors Data Pages

The ultimate aim of this part was showing the information collected from the sensors in different ways. Sensors data pages are the most important pages as they show the user the behaviour of the sensors which is the main requirement of the project. The user can have visual information about the data stored in the database along the time, about real time data or even the user can have visual alerts about the behaviour of the sensors. All these options can be found in the different sections of the Sensors Data Page.

6.2.3.1 Temporal view

Temporal View shows the temporal representation of sensor's data that has been stored in the database over time. It shows graphs of chronological information, being able to choose different sensors, magnitudes, dates intervals and some other options. There are different types of representing the graphs

and also several options to choose. Temporal View allows the user to choose between different options before drawing the desired graph. They are:

- Sensor
  - The user can select four different sensors whose information will be shown on the graph at the same time.

- Interval Date
  - The user can choose the 'Initial Date' and the 'End Date' of the graph. That is, the time interval in which the graph will be represented.

- Magnitude
  - The magnitude which is going to be shown in the graph can be selected. There are four different magnitudes: 'Voltage, Hysteresis, Current and Temperature'. Only one can be chosen each time.

Once all the options are selected, by clicking on the 'Draw Graph' button, the graph is represented. After the graph is drawn, the user can change different options of the plot where the graphic is shown.

These options are:

- Graph Type
  - It can be selected if the graph is shown as 'Lines', 'Points', 'Lines and Points' or 'Area'.

- Sensor Shown
  - It can be chosen if all the sensors or only some of them are shown in the graph.
  - It can also be selected which ones are going to be shown and which ones are not.

- Grid

Different options can be selected considering the grid where the graph is represented.
  - It can be selected if the:
    - Legend is shown or not
    - X-Axis and Y-Axis are shown or not.

The default options which appear when drawing a graph are:

  - Graph Type →Lines
  - Sensor Shown→All sensors selected
  - Grip Options→ Legend and all axis selected

The most important thing regarding the plot options is that they can be chosen and changed in real time. When the user selects for example, 'points' as graph

type, the graph is automatically changed to points without the need of clicking on the draw button again. This happens with all the 'Graph Type', 'Sensor Shown' and 'Grip' options.

6.2.3.2 Icon view

Icon View shows a visual representation of the status of the sensor in a chosen date. The user can select the date and the magnitude to be shown. Once both options are selected, and after clicking in the 'Draw button' an icon shows the status of all the sensors in the chosen moment. Depending of the value of the magnitude an 'ok' or a 'not ok' icon is shown.

The value of the magnitudes that the sensors show is measured based on a threshold. When this value is higher than the threshold, a 'Caution' icon is shown. When the value is lower than the threshold, a 'Ok' icon is shown. If there is no available information about the magnitude sensor in that date, a 'no info' icon is shown. This lets the user visually and quickly know if a sensor is having a good or a bad behaviour.

6.2.3.3 Spatial view

Spatial view shows a Google map where the sensors are placed. Each sensor is represented as a Google Icon and by clicking on each icon, real time information about all the magnitudes of the sensor chosen is shown. The information seen is the last information transmitted from the sensor to the database. The default map shown is an Hybrid map, but there are three different options to choose: Map, Satellite and Hybrid.

6.2.4 Access Page

The access page is the gateway to the Private part. By entering a correct user and password, the user is redirected to the Initial page of the Private section of the website. If the user or password are incorrect, the web asks the user to fill them again.

6.2.5 Contact Page

The contact page is the page where contact information about REALnet and the people who is working there can be found. It includes personal information and a map to indicate their location. Its page structure is similar to the 'Home Page' because it has two menus. One in the top of the page, where the user can choose between the different sections of the website. And another one in the right side of the page, where there are different links related to the UPC.

6.3 Private Part

Private part is only accessible for administrator users. As commented, the access page on the public part leads the user to the private part. In this part of the website, the administrator user can modify, control and edit some parts of the public part. Each page in the private part is related to another one in the public side. Not all can be controlled by the private side; even this would be the best and the ideal situation. However the main required issues can be managed from the private part.

Private part has a different colour from the public side. While the public part is blue, the private part is orange. By this change of colours it can be differentiated when the user is in the public part or in the private one. To understand what exactly can be done in the private side and how it affects to the public part; every page will be explained separately. Images about the private part section of the REALnet website can be found in annex C.

6.3.1 Initial Page

The initial or default page is the first page accessed when the user introduces the user and password in the access page of the public part. The 'Initial page' in the private side is related to the 'Home page' in the public one. Content of the 'Home page' can be modified from here. The 'Initial page' has a text editor where the administrator user can edit whatever he or she needs of the content which appears in the 'Home page'. Images can be added, text can be modified...

6.3.2 Edit users

Administrators user can be created and removed from the 'Edit users' page.
The page is divided in two sections. In the section above, all the existing users with their password appear. After each user and password there is a red icon. When clicking on it, a pop up message of confirmation ask the administrator to confirm that the user must be removed. If this is confirmed, that user is erased. In the section below, there are two text fields, where a new user and a new password can be introduced. By clicking on the 'Add user' button, that user is included in the database and it appears automatically in the section above of the page.

6.3.3 Add Documents

The 'Add documents' page works the same way as the 'Edit users page'. It is related to the Internet links of the 'Documentation page'. The documents that appear in the 'Documentation page' can be managed from this page of the private side. The page is also divided in two parts. The part above contains the

documents and the red button to erase them, while the part below contains four text fields to create a new document.

6.3.4 File Upload

In the 'File Upload' page the administrator user can add documents to the server. These files will appear in the 'Download files' of the 'Documentation page'. In the above part of the page the administrator can select the file which is going to be uploaded. Once is selected, by clicking on the 'Submit' button it is included in the server and it will appear in the below part of this page and also in the 'Documentation page' of the public part.

6.3.5 Logout

The Logout page maybe should not be defined as a page as even it is composed by different html pages, to the user's eyes is only seen as a button. When the administrator is in any section of the Private part, by clicking on the Logout of the menu he or she is logged out and redirected to the 'Access page' of the public part. So the Logout is only used to disconnect from the private part. Once the user has clicked on the Logout, it will be needed to Log in again to access to the private section.

6.4 Data model and web map

6.4.1 Data model

In this section it will be explained the structure of the database and the tables used in each case. The database is called 'PFC_noe' and it is composed by four tables:

- Documentacio
- Markers
- Mesures
-

The structure of the database and the tables can be seen in annex D.

6.4.1.1 Tables

- Documentacio

It is the table where all the documentation is added by the administrator. It is only referred to the documentation shown in the website to the user as an Internet link. It is composed by five fields: 'Id', 'Type', 'Title', 'Author', 'Link'.

Each one of them refers to a feature of the document. Administrators access to this table when they add a new document to the website or when they delete it. Normal users access to this table when they query for available documentation.

- Markers

It is the table where the location and some other information about the sensors can be found. It is composed by six fields: 'id', 'name', 'address', 'lat', 'lng' and 'type'. By using this information, the sensors are located in the Google map that appears in the 'Sensors Data' section.

- Mesures

It is the table where all the information that comes from the sensors is stored. There are seven fields in the table: 'sensor', 'tensio', 'hysteresis' 'corrent', 'temperature', 'data' and 'temps'. The table has information so about the sensor, the magnitudes it is measuring and also the date and time when they were measured. Graphics are based on this information to be created. Also the 'Icon view' in the 'Sensors Data' section makes queries to this table in the database.

- Usuaris

It is the table which has information about the administrator users. All the users that have administrator privileges are stored here. There are three fields in the table: 'Id', 'username' and 'password'. The user and the password of all the administrators can be found. When a user wants to get into the Private section of the website, he or she has to introduce username and password which are checked with the usernames and passwords stored in this table.

6.4.2 Web map

As well as the visual presentation and structure of the website seen in chapter CHAPTER 6, it is also important to understand how the different php, html, css... pages compose the site. In the structure website picture, it have been added the different pages that compose each section. In Figure 6.2 it can be seen the public part and all the pages that compose it. Inside each box, the pages that compose that section of the website, are included. On the top right hand of this picture it can be seen the style sheets that give design to the public part. In Figure 6.3 it can be seen the same for the private part. All pages will be explained deeply next.
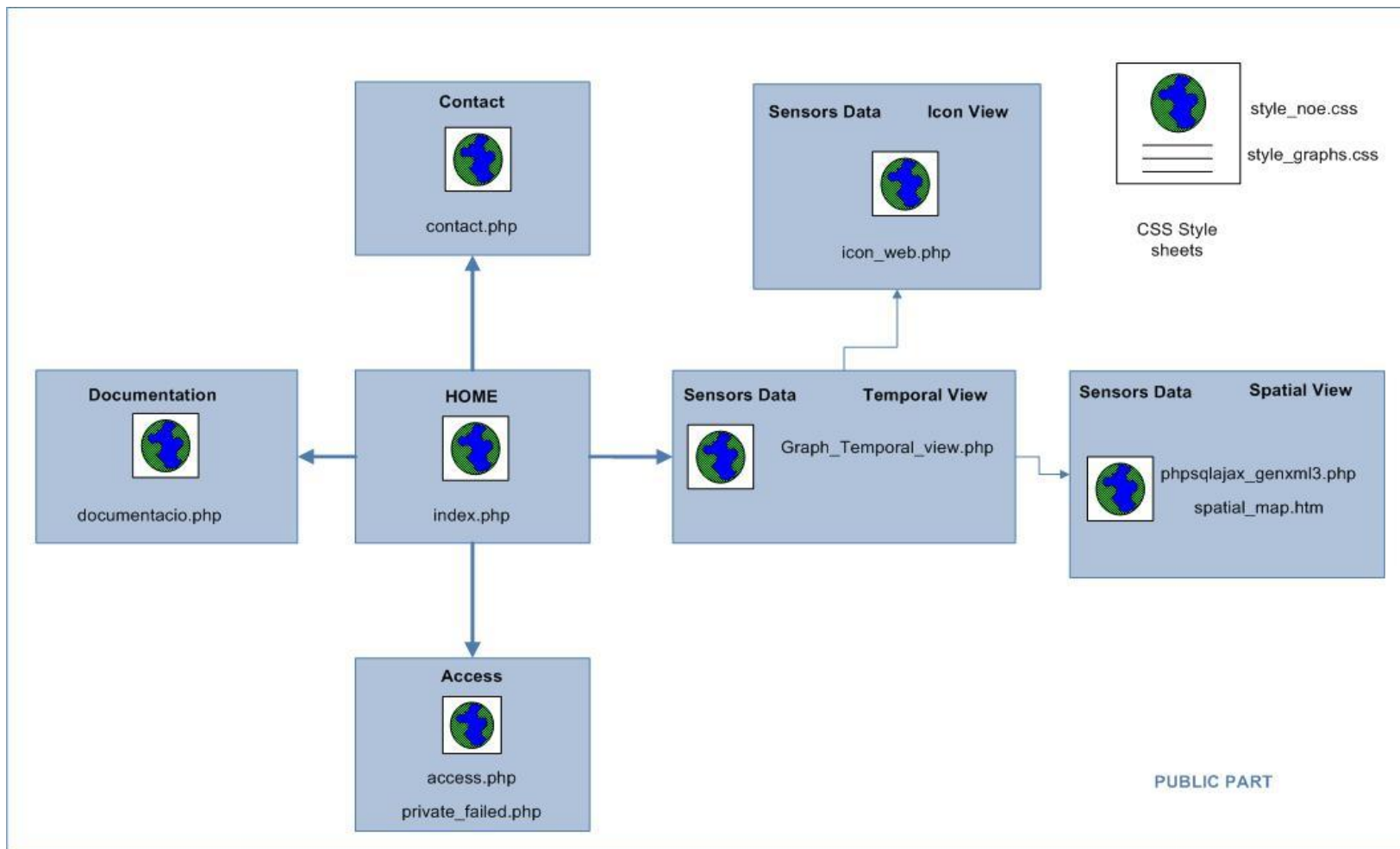
Figure 6.2 Data model of the public website
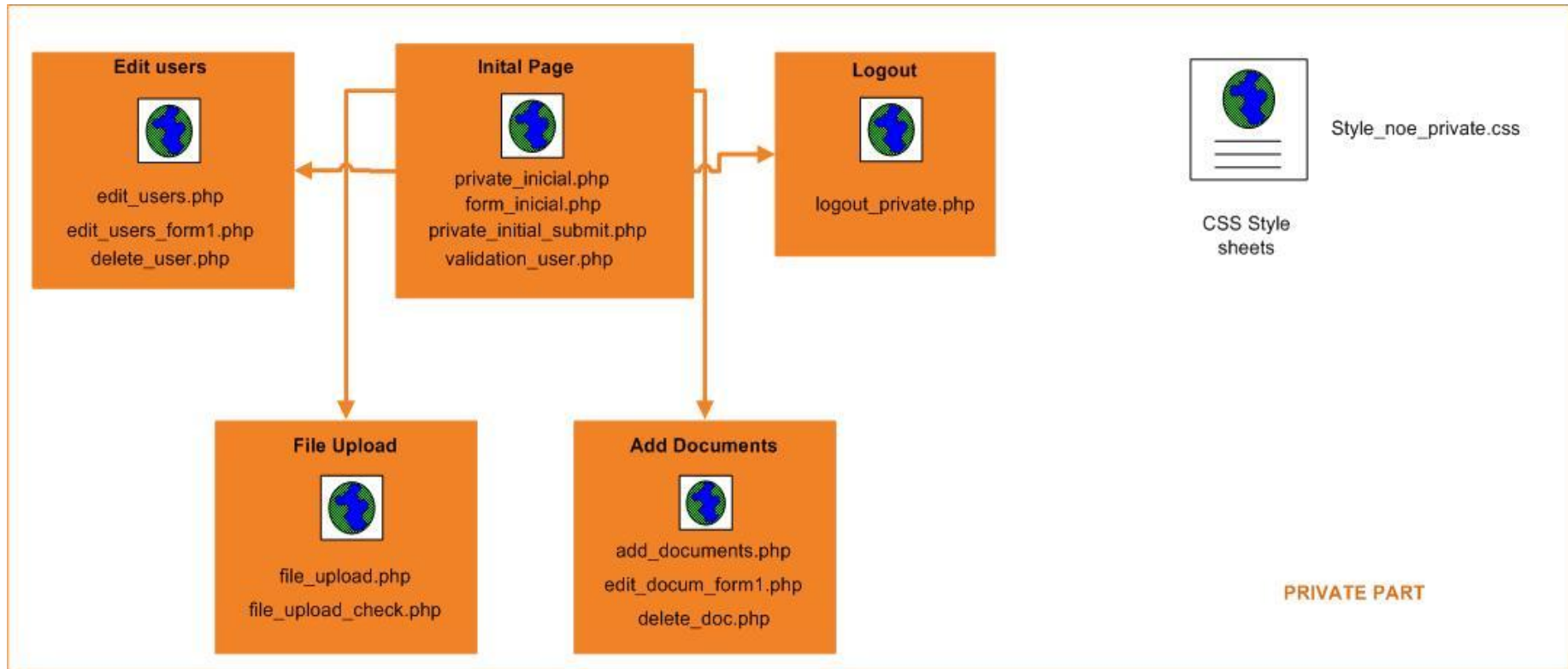
Figure 6.3 Data model of the private website

6.4.2.1 Public Part

- Index.php

It is the web page which has the content of the 'Home page'. The text from REALnet which appears in this page is get from a .txt file which is created from the 'Initial page' in the Private section.

- Contact.php

It has the contact information from REALnet, including a Google map. It is not directly related to any other page.

- Documentacio.php

It is the web page where all the documentation included from the Private section can be found. Both, the information included by the 'Add Documents' page and the files uploaded from the 'File Upload' page. To get the information included at 'Add Documents' must be a connection to the 'documentacio' table, as all these documents included from the private section are stored in this table. Uploaded files from the Private side are downloaded here from the directory where they are stored which is called 'upload'.

- Access.php

This page contains the form where the user introduces the username and password to get into the Private section. There is a connection to the database each time a user tries to get into the private side, to check if the username and password introduced are stored in the 'usuaris' table. If so, the user can access to the private side.

- Private_failed.php

It is the web page where the user is re-directed when the username and password introduced do not fit with the database. In this page, the user is informed that the login has not been correct and there is a link back to the 'access.php' page.

- Icon_web.php

Here icon information about the sensors is shown. There is a query to the database ('mesures' table) according to the parameters the user has introduced, 'date' and 'magnitude'. This page is not directly related to another one.

- Graph_Temporal_view.php

It is the page which creates the graphics according to the information found in the table 'mesures' from the database. The query in the database is

based on the parameters selected by the user: 'Sensors', 'Interval Date' and 'Magnitude'.

- Spatial_map.htm

It creates the Google map where the sensors will be located. The map is created in this page. However, the sensors positions, and all the information they provide when the user clicks on them comes from a XML file generated by another page.

- Phpsqlajax_genxml3.php

It is the web page that generates the XML file where the position and the last found information from the sensors is provided. To get this information, there are two connections to the database.
On the one hand, to get the coordinates from the sensor, there is a query to the 'markers' table.
On the other hand to get the last information in the database from that sensor, there is a query to the 'mesures' table. There, a request it is made asking for the last input added in the database.

Regarding the style sheets, in this public section, there are two of them:

- Style_noe.css

It contains all the design patterns of the public section.

- Style_graphs.css

It contains some particular design patterns related to the graphs part in the public section.

6.4.2.2 Private Part

- Validation_user.php

It is the web page called from all the pages in the private section. It checks that the user is logged as administrator.

- Private_inicial.php

It contains the CK editor with the information that appears in the initial page from the public side. The administrator can modify the content here and save changes. As said, the content of index.php is stored in a .txt file. (main_page.txt) which is created in another page.

- Private_initial_submit.php

This is the page where the 'main_page.txt' file is created. When the administrator 'Submit' the content of the index page from 'Private_incial.php', it is processed by this page. Once the information is received, the commented file is created here.

- Edit_users.php

Here is where the administrator users are shown. There is a table where the username and password can be seen. To show this information, there is a request to the table 'usuaris' from the database, where all the information from the table is shown. In this page, users can be added and also deleted. To perform these actions two more pages are needed.

- Edit_users_form1.php

This page performs the addition of a new user. When the administrator clicks on 'Add user', the page is re-directed to this one and it is here where the 'Insert' query to the database is made and the user is added.

- Delete_user.php

In the same way, when the administrator user proceeds with the removal of a user, the page is re-directed to this one and it is here where the 'Delete' query is performed and the user is deleted.

- Add_documents.php

This page has similar appearance to the 'Edit_users.php' page. It shows the documents stored in the 'documentacio' table. So instead of making a request to the table 'usuaris' from the database, the request is made to the 'documentacio' table. All the information from the table is required and it is then shown in the web page. In this page, documents can be added and also deleted. To perform these actions two more pages are needed.

- Edit_docum_form1.php

It is the page where the addition of a new document is performed. When the administrator clicks on 'Add document', the page is re-directed to this one and it is here where the 'Insert' query to the database is made and the user is added.

- Delete_doc.php

In the same way, when the administrator user proceeds with the removal of a document, the page is re-directed to this one and it is here where the 'Delete' query is performed and the document is deleted.

- File_upload.php

In this page a form to look for a file can be found. Besides, in the bottom of the page, there is a list of all the files. These files are stored in a directory called 'upload'. So, in this page a list of all the files stored in the directory are shown. A new file can be added by the administrator, but to do this, another page is needed.

- File_upload_check.php

This page is responsible for uploading the file to the 'upload' directory. After uploading a file, a message with some information about the file is shown to the user to let him know the file has been uploaded correctly.

- Logout_private.php

This page closes the user session and re-directs the user to the access public page.

CHAPTER 7. DATA TRANSMISSION

The information coming from the sensors arrives to the computer web server via the Ethernet. A device connected between the sensor network and the computer which manages the information is needed. This device is the Telegesis EAP-E. (3) The EAP-E receives the information coming from the sensors and it sends it to the management computer. There, a program is responsible to deal with the information, process it and convert it into an appropriate format to be included into the database.
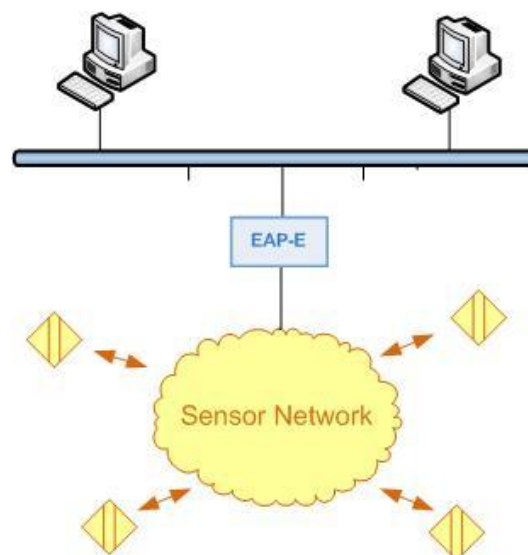
Figure 7.1 Data transmission diagram

To dot the storage of the information, there was a PHP program responsible for dealing with the received information and storing it into the database. This program had to be changed so that it could be adapted to the project needs. So, some changes had to be done so that the program could work properly in this project.

7.1 Program running in background

7.1.1 Requirements and solution

On the one hand, the program was working in foreground which was really annoying when working with the computer. This had to be changed so that the program could be executed in background. Besides, as the program was a .php file it was needed to execute it by using a web browser each first time. This had to be avoided by executing the program directly from the command line.

After some investigation the problem of executing the program from the command was solved by finding the script needed. This script so, was able to execute the PHP file from the command.

This is the script aspect:

php −f (php file route)

The main inconvenience however, was executing the file in background. A solution was found by using a .vbs and a .bat file. The .vbs file contains a script which calls the .bat file. The .bat file itself is responsible for executing the PHP file by using the above commented command script. By executing directly the .bat file, the program works in foreground. However, by using the .vbs file, the program starts working in background.

Another thing that had to be considered when thinking of executing the php program, was having the chance to have the program directly executed when turning on the computer. So, the .vbs file had to be started at the same time the computer was. To achieve this, the program was added to the following part of the registry.

HKEY_LOCAL_MACHINE/Software/Microsoft/Windows/CurrentVersion/Run

This allowed the program to be directly executed on the start up.

7.2 Connection to a remote database.

7.2.1 Requirements and solution

The PHP program was working established connections to a database which was located in the same machine where the program was running. So it connected to a local database. As this project works in Linux and the program runs in Windows, both the database and the program are not in the same place. So it was needed to adapt the program in order that it could be connected to the new required database.

Before explaining how this was solved it is important to clarify why the program was running in Windows instead of having it running in Linux. It would have

sense to consider that if the entire website and the web server were running in Linux, also the program responsible for inserting the information into the database should be. However this was not possible as the COM Port software of the EAP-E it is only supported by Windows. For that reason the program had to remain in the Windows machine.

The solution was adopted by changing the connection mode to the database. Instead of connecting to a local database, the connection was done to the IP address where the virtual machine was located. Moreover, the database permissions had to be changed. It was needed to add a new authorized database user which could insert new information. This was done by giving writing permission to the IP address of the Windows machine in the MySQL server.

CHAPTER 8. DEPLOYMENT IN THE EPSC SERVER

The deployment of the project in the EPSC server consisted of installing a virtual machine which emulated a Linux Operating System inside a computer that was using Windows. The final purpose was having the website created running in the EPSC sever so that everybody could access it by using the Internet.

Before installing the virtual machine in the university server, it was tested in the computer where the entire project had been developed. After that, it was installed in the EPSC computer.

8.1 Implementation

8.1.1 Installation of VMware server

Before creating the virtual machine, the needed software to create it had to be installed. As already commented, this software is VMware Server. *(26)* VMware is software that makes possible installing a virtual machine in a computer. Even the virtual machine can be moved or cloned from one computer to another, it is mandatory to have the VMware installed in each computer where the virtual machine is going to be installed. VMware was downloaded from its official page. The most complicated issue was creating the virtual machine and configuring it properly.

8.1.2 Virtual machine creation and configuration

Once VMware was installed it was needed to create the virtual machine.

8.1.2.1 Creation and configuration of the virtual machine

The created virtual machine was named 'Linux_pfc'. When creating it, some configuration parameters had to be defined. *(27)* A summary of the configured parameters can be shown below:

- Processors
    - It was configured one processor (1 x 1.948 GHz)

- Memory

  - The memory for the virtual machine was 277 MB as the University computer did not allow to install a virtual machine with more memory.

- Hard Disk

  - A 5GB Hard Disk was configured

- Network Adapter

  - The Network Adapter to allow having Internet access was configured in Bridge mode which lets the user have the virtual machine as another computer in the network.
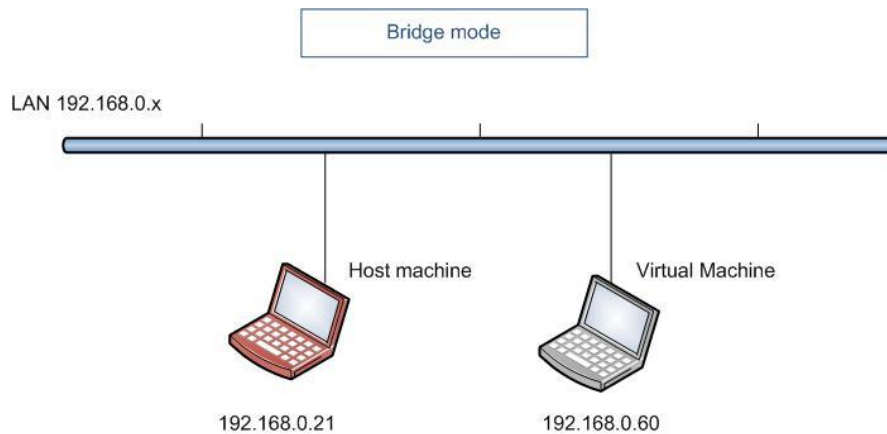
Figure 8.1 Bridge mode network adapter

Bridge mode is useful when the user wants the virtual machine to be part of the physical network and so, allow access to all network resources directly, or for example, allow access to resources that are in the machine as a server web...

- CD/DVD Drive

  - The CD/DVD Drive has been enabled. The main function of the CD/DVD Device was help to install the needed Ubuntu version in the virtual machine.

  - The Ubuntu version was downloaded from the Ubuntu web page as an ISO image.

- o For this reason, the type of connection for the CD/DVD drive was 'ISO image' which allowed installing the Ubuntu version directly from the downloaded ISO on the computer.

- USB Controller

  - o The USB controller has been enabled just in case in any moment is needed to install or copy something stored in an USB memory.

An image of the main configuration features can be seen in Figure 8.2.
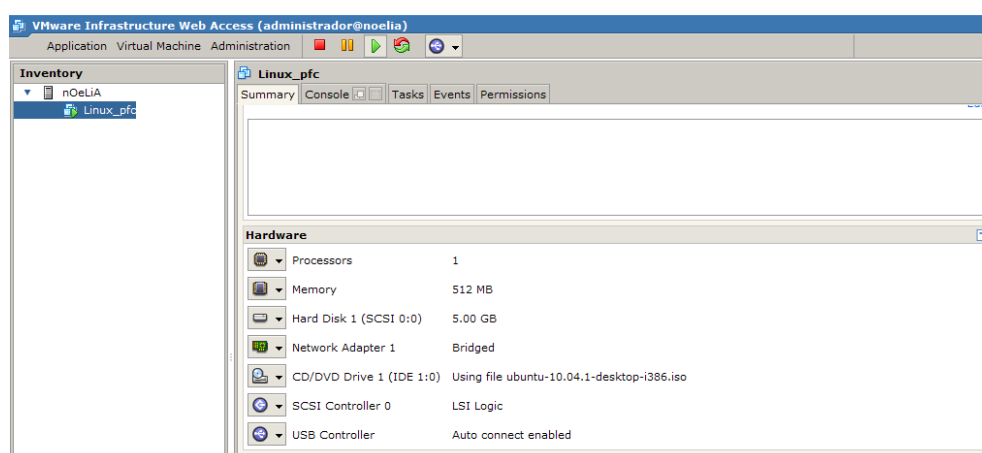
Figure 8.2 Configuration features of the virtual machine

Once the virtual machine was configured, it was powered on for the first time through the ISO image of Ubuntu. Thus, the operating system was installed in the virtual machine. Once Linux was running the network interface was configured to have access to the Internet.

8.1.2.2 Installing the web server in the virtual machine

After having installed Linux in the virtual machine all the necessary tools to have the web server running on it were needed to be installed. As the Internet connection had already been configured, it was only necessary to install LAMP. After that, PHPMyAdmin was installed too to have easier access to the databases. Once the web server was installed and the database editor was working too, it was needed to test everything was running.

8.1.2.3 Testing the website

It only remained to move the website from the development computer to the virtual machine and test that it can be accessed by another computer in the local network. In the picture below it can be seen the scenario used to test the

web server in the local network. From both computers, the virtual machine and another host in the network the REALnet website was accessible and could be seen.
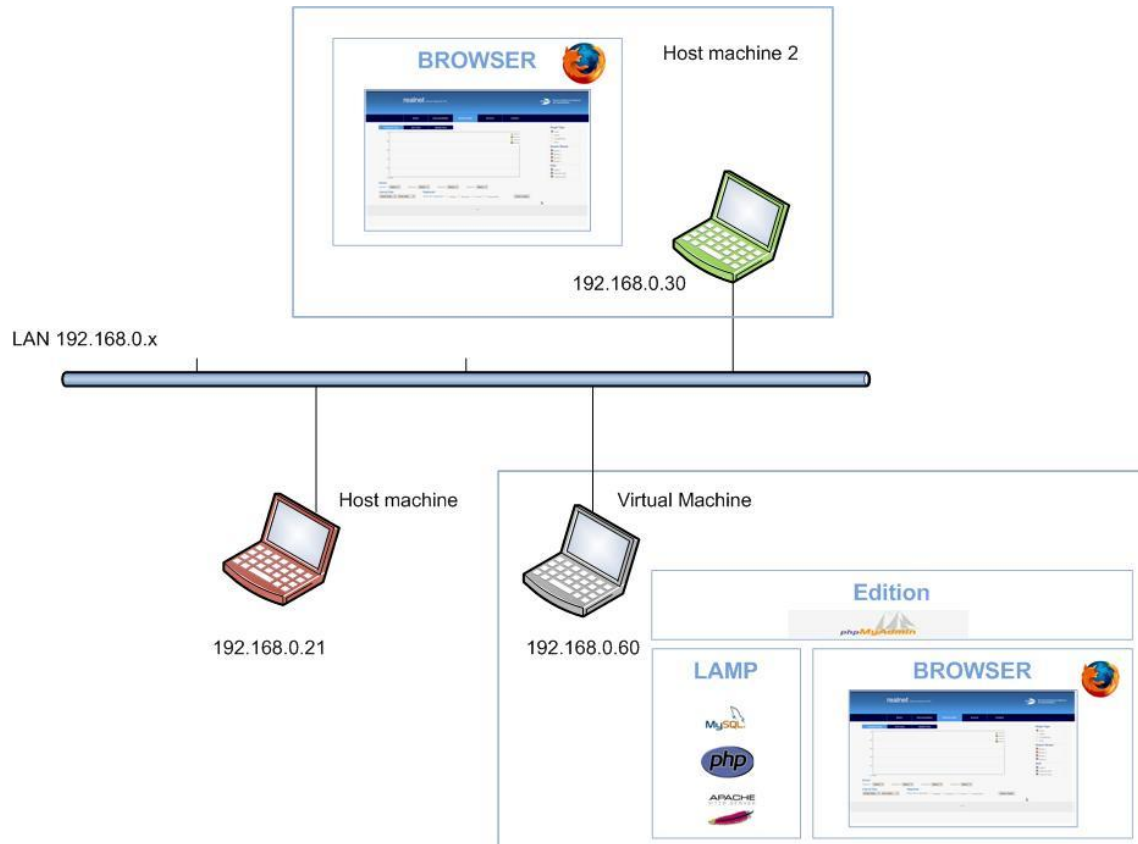
Figure 8.3 Connection and website test

So, the final part of this deployment was testing the created virtual machine in the EPSC computer and checking the website was accessible from the Internet.

CHAPTER 9. CONCLUSIONS

9.1   Achieved goals

A website accessible from the Internet has been developed. To make it accessible from the Internet, a web server based on Linux has been implemented in the EPSC. The website contains the information received from the REALnet sensor network, and also some other important information from the REALnet team. The website contains a public and a private section. The first one it is accessible for everyone. The second one it is only accessible for administrator users who can manage the public part.

All has been tested to make sure it was working correctly.

So, goals pursued in this project have been achieved. Although, the required technologies to be used, and the available resources meant some difficulties when considering the different options to perform the project, finally a solution to each problem was found.

9.2   Future improvements

In all projects there are many things that can be improved. Also in this project Not all is perfect and there are many things that can be better. Sometimes, these things are not good enough because lack of time to improve them, or simply because in the moment of the development of the project there were not enough resources to do them better. So, according to this, in this chapter it will be explained what could be improved and why it has not been improved while developing the project.

There are two main issues to consider when talking about what can be improved. One is some features of the website. Another point to take into consideration is the EPSC server where the website is now located.

9.2.1 Website part

One of the most important parts of the website is the 'Sensors Data' section. This part uses information from the database to create graphs... which gives data about the sensors in the REALnet website.

However, not all the information provided to create graphs is in the database. Some features are programmed directly in the code.

This is a shortcoming when the administrator of the website wants to have more control on it. That is making changes etc. If all the information is in the database, it is easier to have control on it. If not, it is more difficult. When the

website was almost finished it was thought that in a future, the website could be improved giving the administrator more freedom to take control of the user's interface through the private part.

9.2.2 Add/Delete sensors

One of the most important points for the administrator of the website is being able in future to increment or decrement the number of sensors of the network and reflect it in the website.

Sensors are not stored in an individual table in the database. They appear in the table where the measurements from the sensors are stored. So, the administrator cannot add or remove a sensor from the database by using the private interface in an easy way. It would be needed so, to add a table where all the active sensors were stored. Then, it would be also necessary to implement a new section in the private part of the website, where the administrator could manage the sensors, inserting or deleting them from the table in the database. In the same way as now the administrator can add or delete administrator users.

Another issue regarding adding or deleting sensors is that some information about them is stored in other tables of the database where the administrator cannot have access. For example the table markers, where the coordinates of the sensor are stored. If a new sensor would be added to the sensor network, it would have to be included in this table too. So, it would be need to create another administrator interface to do this.

9.2.3 Add/Delete magnitudes

In the same way, add a new sensor in the network could mean having a new magnitude that this sensor could measure and the others could not.

When the user selects a magnitude in the Sensors Data section, they are not stored in a table in the database. They appear in the web page as they are directly introduced in the programming code. So, today adding a new magnitude would mean introduce it directly in the code.

To avoid this, a future improvement, could be adding a new table in the database where available magnitudes could be managed. It would be just the same situation as with the sensors. Then a new interface in the private section would have to be added, so that the administrator could manage when to add or remove a magnitude in the table.

9.2.4 EPSC server

As already said in previous sections the web server which hosts the REALnet website, is installed in a Linux virtual machine. And it is installed in a computer running with Windows.

Even virtualising has some advantages as for example, energy savings, it is always better to have a dedicated computer to work as a server. A dedicated computer to work as a server is safer than having a computer used for some other purposes besides being a server. Besides, it would be interesting to have a computer with better resources than the one it is used now so that the website could be accessed more quickly from the Internet and it could have more users simultaneously.

However, the REALnet department resources did not allow using a dedicated equipment and due to this, the server is running in the virtual machine explained before.

9.3 Environmental Impact

The project has not a very significant environmental impact.
However a change can be noticed. Before performing this project, the EPSC computer where the web server has been installed, was used only a few hours a day. This means the computer was turned off the rest of the time. From now on, as it is hosting the REALnet website it must be always turned on which means higher energy consumption.

REFERENCES

1. **Albesa, Joan and Gasulla, Manel.** *REALnet:An environmental WSN testbed*

2. **Albesa, Joan.** *Xarxa de sensors sense fils per a la monitorització ambiental de l'estany del Campus del Baix Llobregat.* 2007

3. **Telegesis.** http://www.telegesis.com/downloads/general/TG-EAPE-PM-006-104.pdf

4. **W3.** http://www.w3.org/TR/html-markup/

5. **K. V. Chandrinos, P.E. Trahanias.** *Beyond HTML: Web-Based Information Systems.*

6. **w3chools.** *http://www.w3schools.com/css/css_intro.asp*

7. **PHP.** *http://www.php.net/*

8. **web, Desarrollo.** *http://www.desarrolloweb.com/manuales/9/*

9. **w3chools.** http://www.w3schools.com/JS/default.asp

10. **Mugur.** http://www.webdotdev.com/nvd/articles-reviews/javascript/javascript-and-other-client-side-technologies-1184-204.html

11. **Prototype.** *http://www.prototypejs.org/*

12. **Deensoft.** *http://www.deensoft.com/lab/protochart/*

13. **Google,Code.** *http://code.google.com/intl/ca/apis/maps/documentation/javascript/v2/*

14. **Microsoft.** http://msdn.microsoft.com/en-us/library/ff649643.aspx

15. **Linux.** http://www.linuxhelp.net/guides/lamp/

16. **Linux**. *http://www.linux-es.org/sobre_linux*

17. **Apache.** *http://www.apache.org/*

18. **MySQL.** *http://www.mysql.com/*

19. **Opensourcewebbook.** *http://www.opensourcewebbook.com/about/ch05.pdf*

20. **Eclipse.** http://www.eclipse.org/

21. **Firebug.** http://getfirebug.com/whatisfirebug

22.**phpMyAdmin.** http://www.phpmyadmin.net/home_page/index.php

23. **CKEditor.** http://ckeditor.com/

24.**Webdevelopersnotes.***http://www.webdevelopersnotes.com/basics/client_se rver_architecture.php3*

25. **Djce.** *http://djce.org.uk/dumprequest*

26. **VMWARE.** http://www.vmware.com/es/

27.**VmWare Tutorial.**http://forevergeeks.com/installing-ubuntu-810-on-vmware-server-20

BIBLIOGRAPHY

- Johann-Christian Hanke, Introducción a PHP (PC Cuadernos), Diciembre 2002

- Martin Baier, Iniciación a JavaScript (PC Cuadernos), Noviembre 2001

- John Maxwell, CSS: Webs con estilo (PC Cuadernos), Marzo 2002

- Spona Helma Marcombo, Programación de bases de datos con MYSQL y PHP, 2010

ANNEX

ANNEX A. MVC example

MVC in the access and the temporal view sections

The model view controller has been followed as design pattern. To have a better understanding of how this pattern has been used, an example will be explained next.

In the website part of the project, there are two sections. The public and the private one. To go from the public to the private part, the access page it is needed. In this page, it can be seen how the MVC has been followed.

The appearance of the page is as it can be seen in Figure A.1.
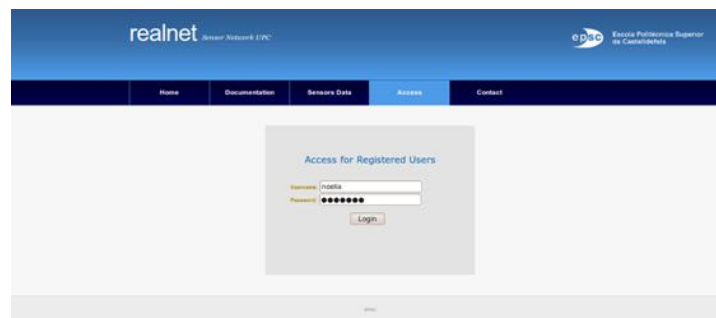
Figure A.1 Access section

The view part is composed by HTML and CSS. The CSS is performed by a file responsible for the appearance of the page. By this way the user interface is shown in a suitable format to interact. When the user wants to get into the private section he or she clicks on the Login button. When clicking on it, he or she is activating a 'php' function which activates another function responsible to get information from the database through a SQL sentence. In this case information about administrator users. The first php function belongs to the controller. The second one and the SQL sentence belong to the model part.

In the same way in other cases, the controller could be activated by a JavaScript function instead of a php one. It is the case of the 'temporal view' section. Temporal view is responsible for showing graphic information coming from the sensors which is stored in the database. Its appearance can be seen in Figure A.2.

Figure A.2 Temporal view appearance

Once the graphic is done the user can choose between different types of graphs to be shown. To do it, he or she click on the chosen option. By doing this he is activating a JavaScript function which performs the demanded action. In this case, so, the controller would be activated by a JavaScript function.

ANNEX B. REALnet public part website images

In this Annex, the visual appearance of the public part from the developed website can be seen. Each page will be explained in the same order as it has been explained in chapter CHAPTER 6.

Home Page

It can be seen the home page, (default page) appearance.

Figure B.3 REALnet Home Page

Documentation page

As said in chapter CHAPTER 6, the Documentation page can be separated in two parts. The first one where the Internet Links appear and the second one where the files to be downloaded appear.

Internet Links

In Figure B.4 can be seen the view that is shown to the user when the documentation page is selected. All kind of documents appear. In this case, the different types of documents stored in the website are: 'Thesis', 'Projects' and 'Papers'.
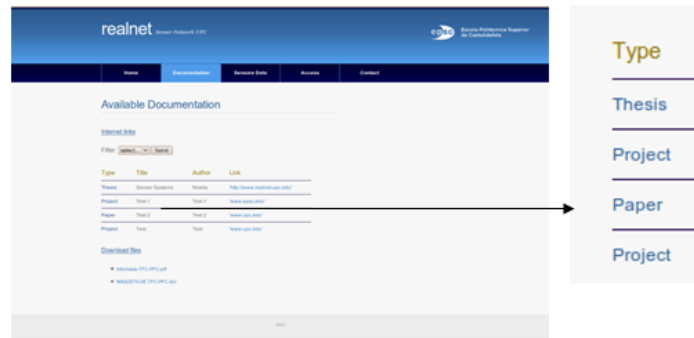
Figure B.4 Documentation default view

However, as commented, the user can select only one kind of document to be shown. This is achieved by using the filter which appears above the documents. For example, in Figure B.5, the selected document has been 'Project', and only the projects are shown to the user.
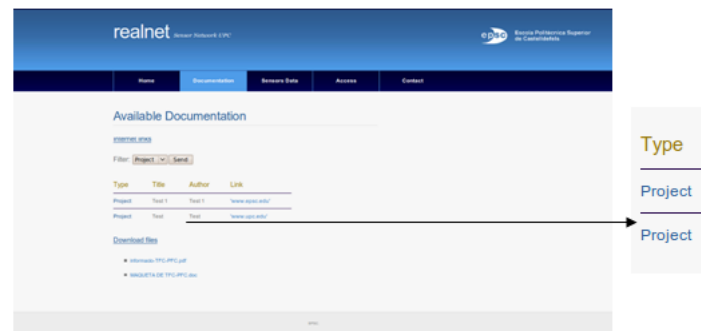
Figure B.5 Project selected in documentation view

Download Files

The files the user can be downloaded are shown in the bottom of the page.
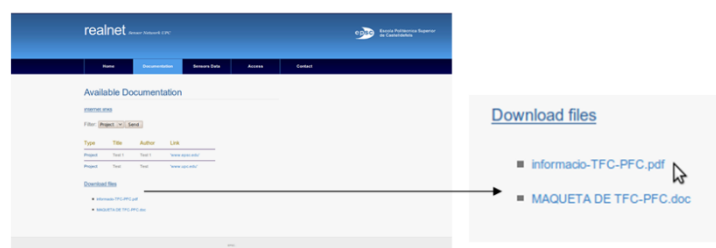
Figure B.6 Documentation 'Download files'

Sensors Data Page

All the views of this section will be shown.

Temporal View

This is the default page when selecting the 'Sensors Data' option in the main menu.
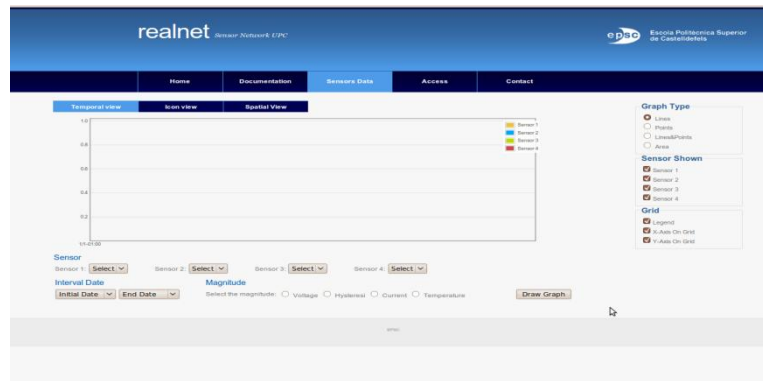
Figure B.7 Temporal View

By choosing the different options the graphic is drawn. X axis is temporal. It is automatically adapted depending on the interval that is represented. Y axis belongs to the magnitude value and it is adapted depending on the magnitude chosen.
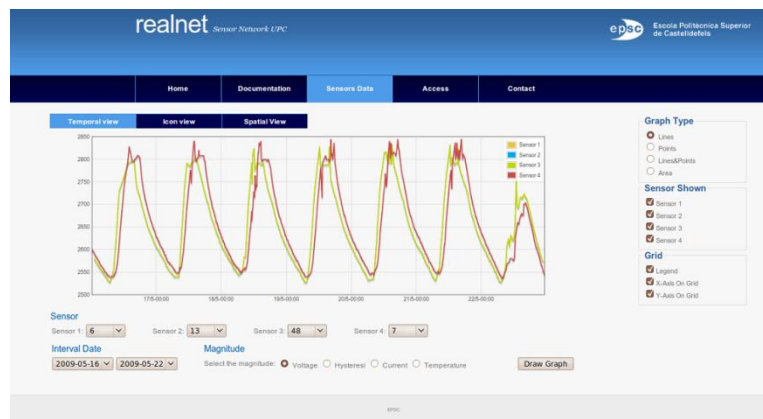
Figure B.8 Temporal View representation

Once it is represented, the user can choose the type of graph to be shown. One image of each graph type is shown below:

Lines

Figure B.9 Lines representation on Temporal view

Points

Figure B.10 Points representation on Temporal View

Lines and Points

Figure B.11 Lines & Points representation on Temporal View

Area

Figure B.12 Area representation on Temporal view

Some other options can be chosen. For example, in the following picture, only 'Sensor 3' is represented as it is the only one selected in the 'Sensor Shown'.

Figure B.13 Sensor 3 selected in Temporal view

In the following picture it can be seen as neither the Legend nor the X-Axis and the X-Axis are not shown as they are not selected.
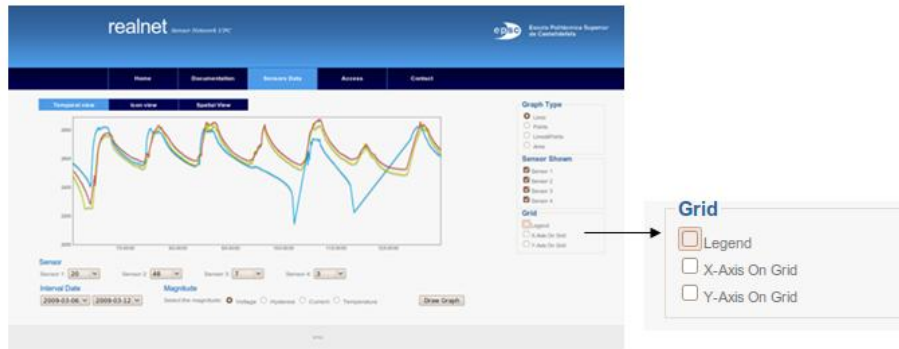
Figure B.14 No legend and no Axis in Temporal View

Icon View

The icon view is shown below. In Figure B.*15* it can be shown as 'Sensor 0' and 'Sensor 18' have a good value of voltage, while 'Sensor 7' and 'Sensor 48' have a higher value than expected. All the other sensors do not have information about Voltage in that moment.

Figure B.15 Icon view

Spatial View

The default view of the Saptial view is the Hybrid map.
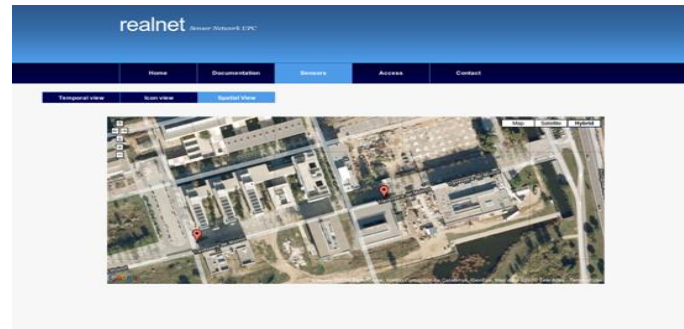
Figure B.16 Hybrid map in Spatial view

It can be changed to Satellite view and map view.
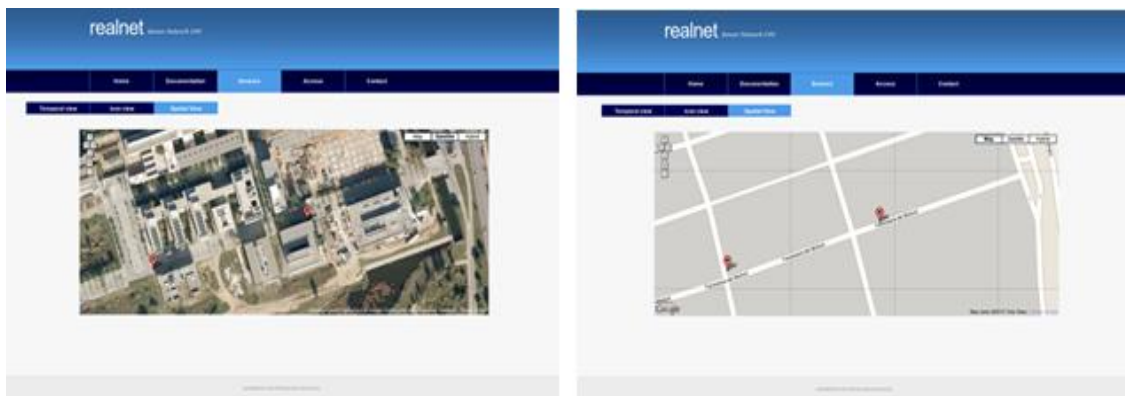
Figure B.17 Satellite and map view

Below it can be seen the information which is shown when clicking on an icon:
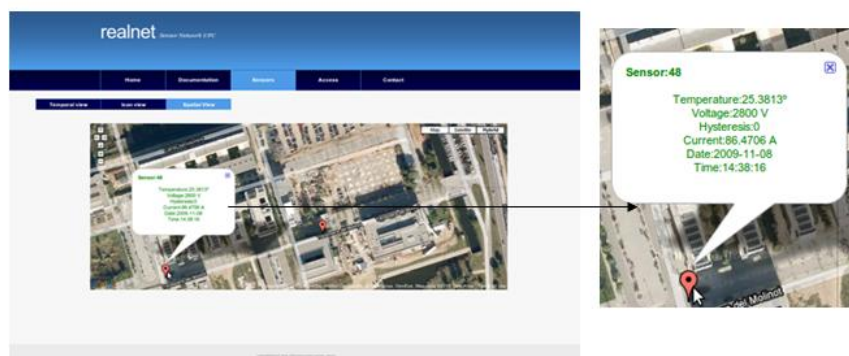
Figure B.18 Sensor information in Spatial view

Access page

The appearance of this page can be seen in Figure B.19:
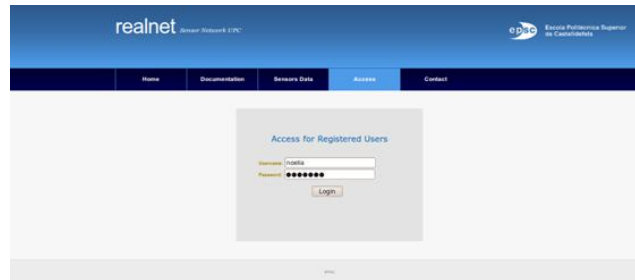
Figure B.19 Access page

Contact page

Contact page is shown in Figure B.20:
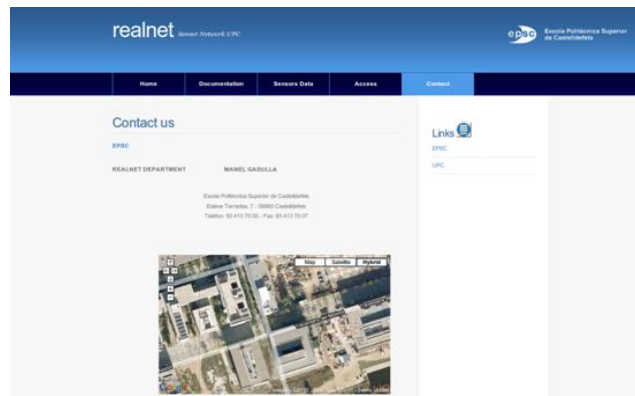
Figure B.20 Contact Page

ANNEX C. REALnet private part website images

Initial page

A picture of the appearance of the 'Initial page' can be seen in Figure C.21.
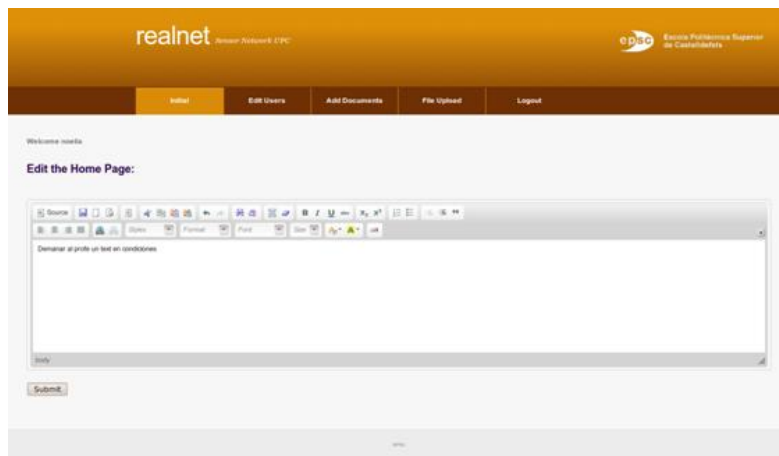
Figure C.21 Initial page of private site

Edit users

The appearance of Edit users page is shown in Figure C.22.
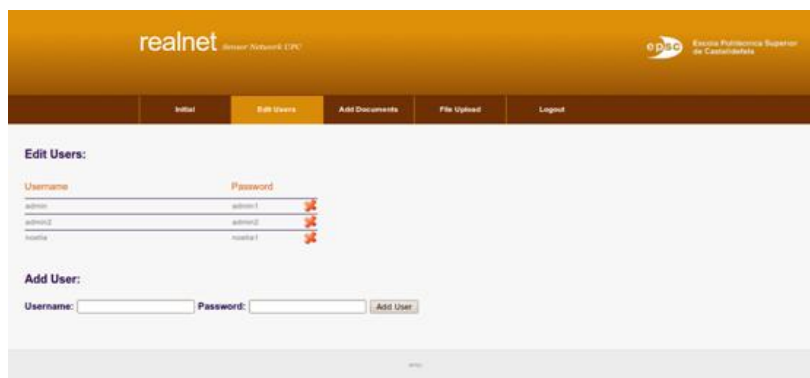
Figure C.22 Edit users page

Add Documents

The appearance of the page can be seen in Figure C.23.
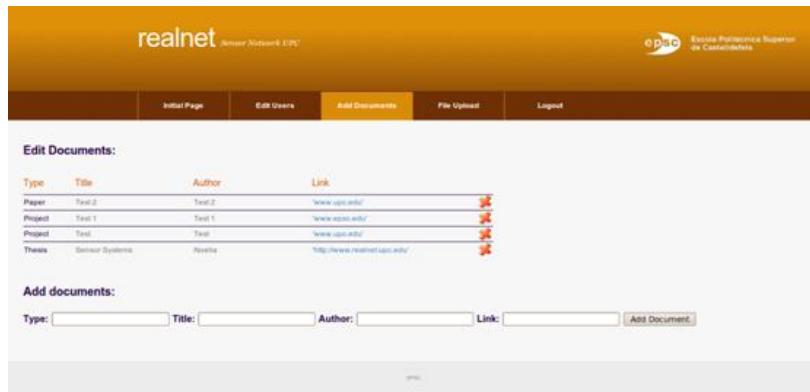
Figure C.23 Add documents page

File Upload

File upload page can be seen in Figure C.24
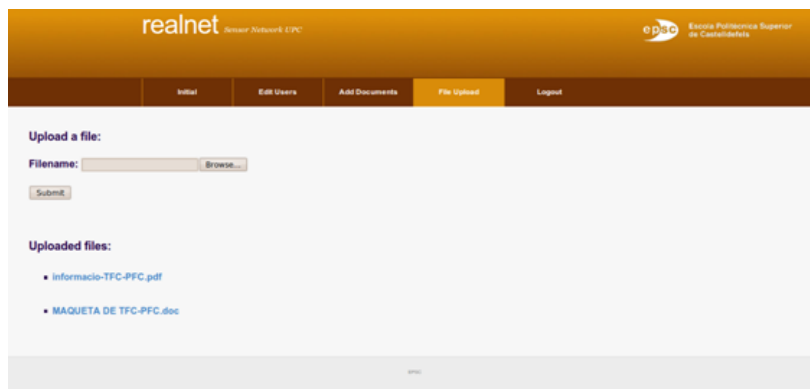
Figure C.24 File Upload page

Logout

The logout button is shown in Figure C.25.

Figure C.25 Logout

ANNEX D. Database and tables structure

Database structure

The database structure is shown in Figure C.26.
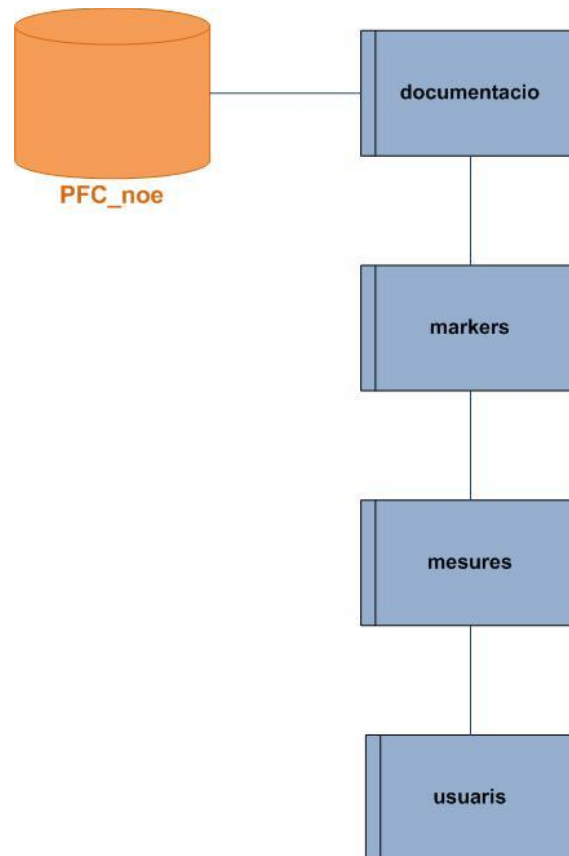
Figure C.26 Database structure

Tables structure

The tables structure can be seen in the following pictures. Figure C.27 shows the 'Documentation' table. Figure C.28 shows the 'Markers' table. In Figure C.29 an extract of 'Mesures' table can be seen. Finally, the 'usuaris' table can be seen in Figure C.30.

←T→	id	Type	Title	Author	Link
□ ✎ ✗	2	Thesis	Sensor Systems	Noelia	http://www.realnet.upc.edu
□ ✎ ✗	12	Project	Test 1	Test 1	www.epsc.edu
□ ✎ ✗	13	Paper	Test 2	Test 2	www.upc.edu
□ ✎ ✗	11	Project	Test	Test	www.upc.edu

Figure C.27 Documentation table

←T→	id	name	address	lat	lng	type
□ ✎ ✗	2	48	Farola 1	41.274864	1.986305	wireless sensor
□ ✎ ✗	1	13	Farola 2	41.275288	1.988321	wireless sensor

Figure C.28 Markers table

←T→	sensor	tensio	hysteresis	corrent	temperatura	data	temps
□ ✎ ✗	3	2840	0	85.2941	21.0018	2009-02-23	15:51:47
□ ✎ ✗	3	2840	0	83.2353	21.0018	2009-02-23	15:56:21
□ ✎ ✗	3	2840	0	83.8235	20.8193	2009-02-23	16:01:00
□ ✎ ✗	3	2840	0	79.1176	20.6368	2009-02-23	16:11:37
□ ✎ ✗	3	2840	0	77.3529	20.2718	2009-02-23	16:21:24
□ ✎ ✗	3	2840	0	72.3529	20.0893	2009-02-23	16:25:59
□ ✎ ✗	3	2840	0	68.8235	20.0893	2009-02-23	16:30:39
□ ✎ ✗	3	2836	0	60.2941	19.5419	2009-02-23	16:50:14
□ ✎ ✗	3	2832	0	53.8235	19.3594	2009-02-23	17:00:40
□ ✎ ✗	3	2828	0	34.7059	19.1769	2009-02-23	17:05:17
□ ✎ ✗	3	2828	0	47.3529	18.9945	2009-02-23	17:09:52
□ ✎ ✗	3	2812	0	6.76471	18.447	2009-02-23	17:20:17
□ ✎ ✗	3	2812	0	19.4118	18.082	2009-02-23	17:24:53
□ ✎ ✗	3	2812	0	35	17.3521	2009-02-23	17:35:14
□ ✎ ✗	3	2812	0	32.9412	17.1696	2009-02-23	17:39:56
□ ✎ ✗	3	2808	0	24.4118	16.9872	2009-02-23	17:44:39
□ ✎ ✗	3	2808	0	22.0588	16.8047	2009-02-23	17:49:16
□ ✎ ✗	3	2800	0	17.6471	16.4397	2009-02-23	17:59:48

Figure C.29 Mesures table

←T→	id	username	password
□ ✎ ✗	1	noelia	noelia1
□ ✎ ✗	23	admin	admin1

Figure C.30 Usuaris table