TRAINEESHIP REPORT

Responsive HTML website presenting the graphical representation of data collected from a server

Trainees

Lucian-Mihai Ciugudean
Roxana-Daniela Sustic

Mentor

Prof. Jaume Segura

Special thanks to Adolfo Pastor for all the help.

Table of contents

1. Introduction

Approximately 70 percent of the world's population is expected to live in cities by 2050 according to Gartner. This rapid urban growth is already placing a considerable strain on the existing infrastructure, and with more people making the move to urban living, it's only going to get worse in the coming years.

To accommodate this new demand on cities, municipalities around the globe are turning to the Internet of Things (IoT) for new opportunities to use data to manage traffic, cut pollution, make better use of infrastructure and keep citizens safe.

In the acoustics research field, IoT has a series of applications and benefits such as having more control of noise levels by permanent and real-time control, detecting new noise sources or using these tools as a showcase for informing the citizens among others.

2. Project requirements

This paper describes the steps taken to collect data from a local data base which contains and processes real-time data from 4 Raspberry Pi located in the cafeteria of the University of Valencia in order to graphically represent various parameters such as loudness, sharpness, fluctuation, etc. and develop an user-friendly interactive web page to display them.

For the implementation of the project both hardware and software components are used:

- 4 Raspberry Pi that collect the physical data
- A MongoDB database where the collected data is stored
- A Bitvise client profile to connect to the local server
- HTML5 and CSS to build the web page
- JavaScript and JQuery libraries in order to plot the charts

3. Data processing

3.1. Data collection

The main component of this project is the processing unit, which is also used for the data acquisition and connectivity. For achieving the established requirements, the design of the noise monitoring device was based on a Raspberry Pi 3 Model B+ single board computer. The Raspberry Pi platform offers a number of advantages like good computing power, high versatility and low power consumption resulting in affordable and durable nodes. Those qualities, together with the continuous upgrades that the platform undergoes over time makes Raspberry Pi the best option for the development of a noise monitoring prototype. There are 4 such devices in different parts of the cafeteria presented in Figure 3.1.1. which collect data of 6 parameters: loudness, sharpness, SPL, roughness, fluctuation and PA.

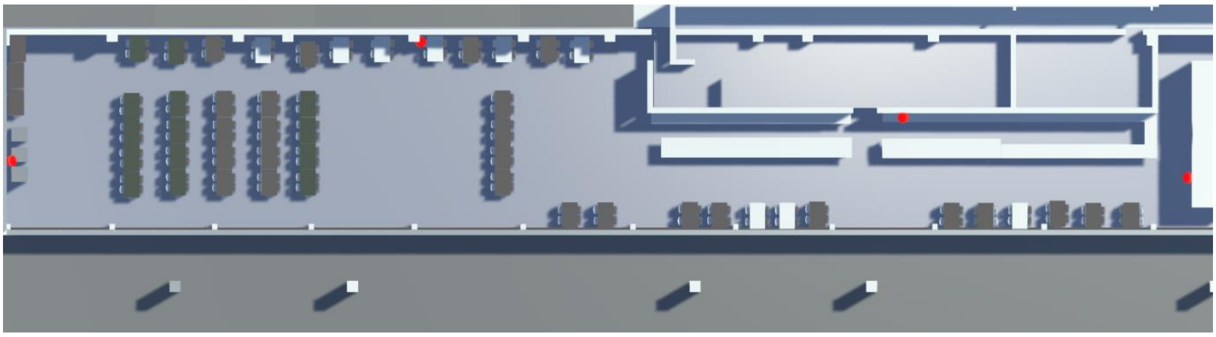

Figure 3.1.1. The placement of the noise monitoring devices in the cafeteria represented by red

3.2. Data storage

Data collected from the noise monitoring devices is stored in a remote Secure Shell (SSH) server accessed through the virtual machine mariola.uv.es. The server implements a MongoDB document-oriented database due to its rich and expressive query language that allows filtering and sorting by any field, a very important aspect that must be taken into account when working with a big volume of data.

3.3. Data access

The SSH connection is implemented using a client-server model. In order to access the local server, a Bitvise client profile is established which includes graphical as well as command-line SFTP support, a FTP-to-SFTP bridge and powerful tunneling features including dynamic port forwarding through integrated proxy and remote administration for the server.

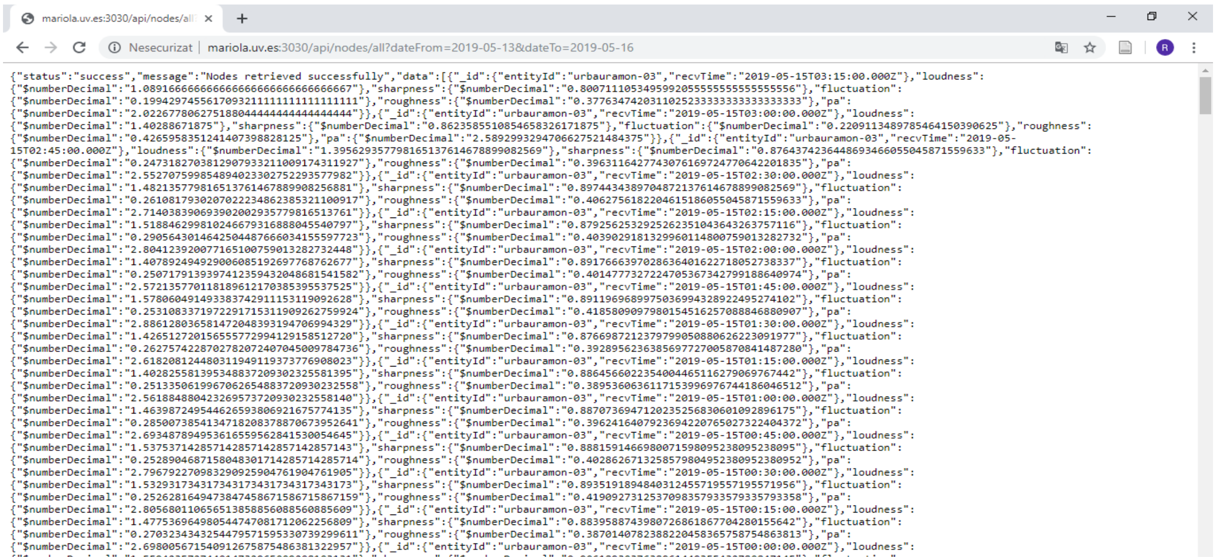

Figure 3.3.1 Data displayed on http://mariola.uv.es:3030/api/nodes/all?dateFrom=2019-05-13&dateTo=2019-05-16

4. Website interface

In order to simplify how the user interacts with data, a responsive website was build using HTML5 and CSS. For the user interface a Bootstrap 4 admin template was used, which provides a collection of ready to use code snippets and utilities, custom pages, charts and some useful widgets.

The website is structured as presented in Figure 4.1 having a left pannel containing a side menu and a right pannel which includes the header, content and footer.

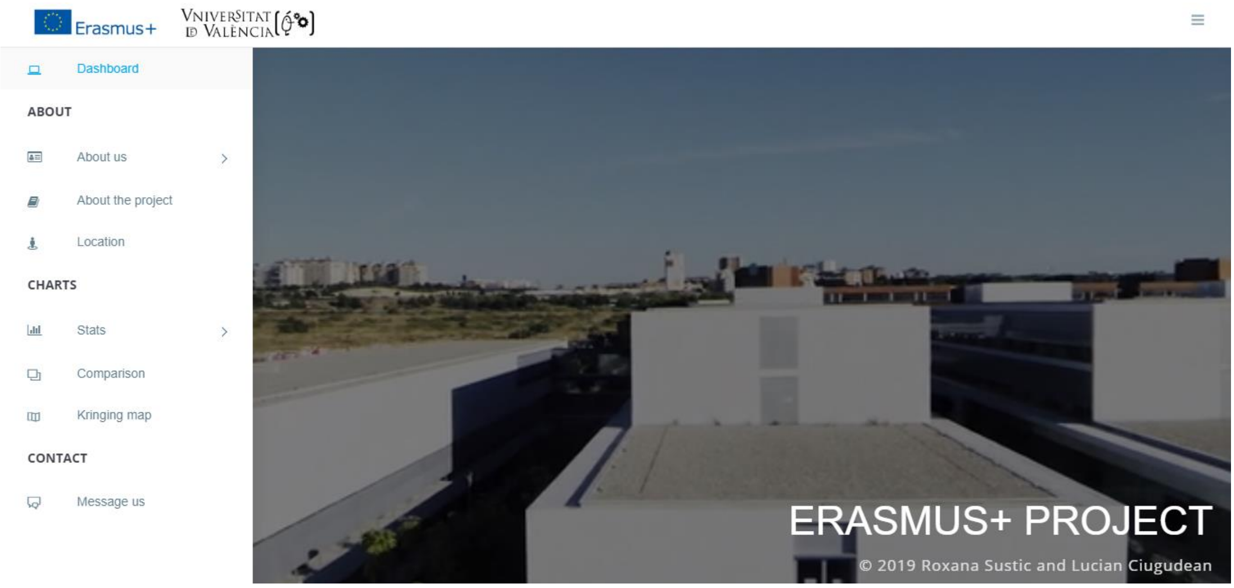

Figure 4.1 Structure of the website interface

4.1. Side Menu

The side menu is a collapsible vertical navigation bar divided into 3 main menus: ABOUT, CHARTS and CONTACT as shown in Figure 4.1.1.

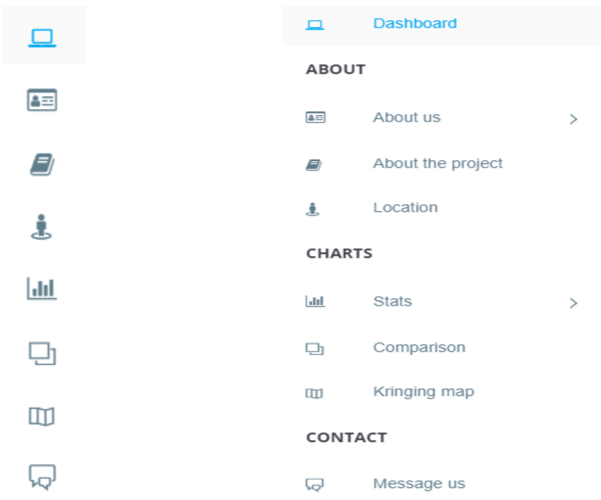

Figure 4.1.1. Structure of the navigation bar: collapsed and expanded

In the ABOUT section, there is a dropdown item called About us which contains a sub-menu with the items Roxana Sustic and Lucian Ciugudean offering details about the creators of the website. This section also contains 2 other items: About the project, presenting some stats and a short description of the project, and Location which enables the user to access the exact location of the noise monitoring devices on an embedded map.

The CHARTS section contains the dropdown menu Stats where a list of all 4 Raspberry Pi is displayed in order to access the corresponding graphs. The second item of this section is Comparison and it allows the user to plot a comparison graph between the parameters of each Raspberry Pi. The last item is called Kringing map item and it displays a geostatistic method that predicts the value in the geographic area of the cafeteria given measurements from the noise monitoring devices.

The CONTACT section has a single item Message us represented by the contact page.

For all items in the navigation bar, predefined icons from the Temify theme have been used to improve the overall design of the user interface.

4.2. Header

The header presented in figure 4.2.1. contains in the left side the logos of Erasmus+ and University of Valencia with hyperlinks to their official webpages and in the right side an icon responsable with toggling between the collapsed or expanded version the side menu.

Figure 4.2.1. Structure of the header

On each webpage, excepting the main webpage, breadcrumb navigation is used to enhance the way users find their way around the interface. This method is an effective visual aid that indicates the location of the user within the website's hierarchy, making it a great source of contextual information for landing pages. An example of breadcrumb navigation bar can be seen on the Location webpage, as represented in Figure 4.2.2.

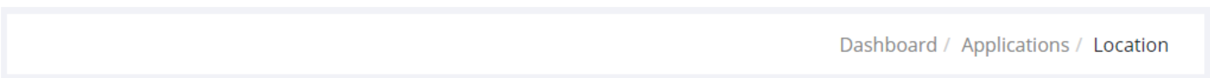

Figure 4.2.2. Example of breadcrumb navigation

4.3. Footer

The footer is divided into 1 row and 2 columns as displayed in Figure 4.3.1. It is present on all pages excluding the dashboard page.

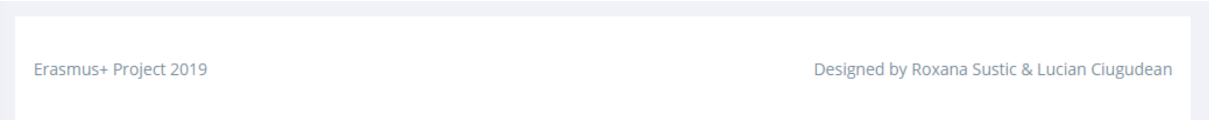

Figure 4.3.1. Structure of the footer

4.4. Content

4.4.1. Dashboard

The main webpage is represented by the Dashboard, displayed in the figure below, and it contains an embedded video background element which is a great way to grab attention through motion. The video behaves like a section so content could be placed in the lower right side.

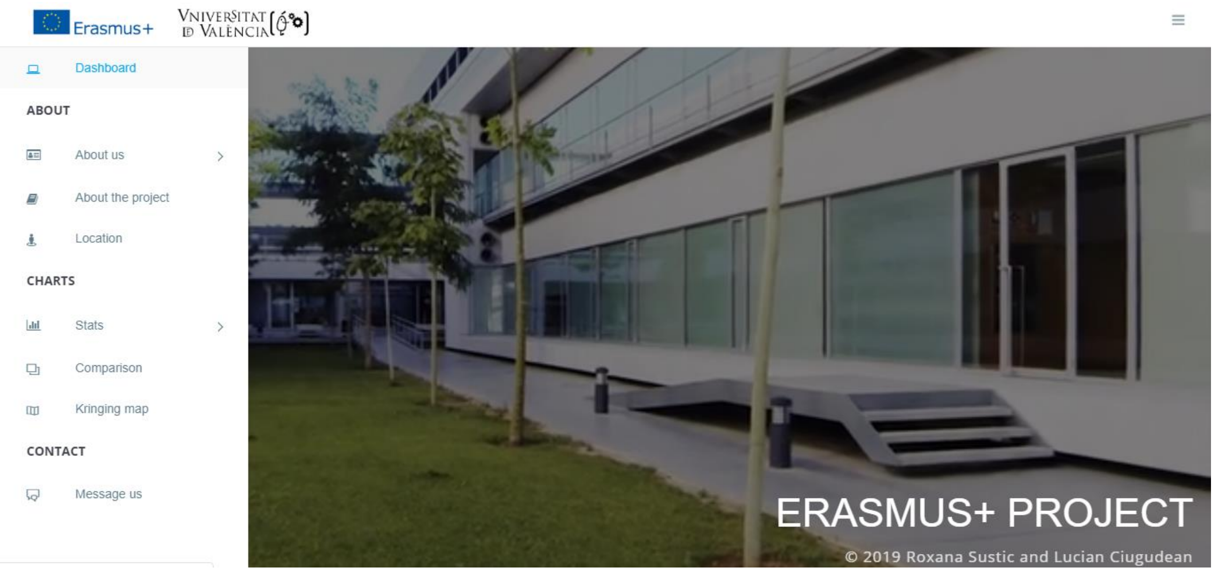

Figure 4.4.1.1 Dashboard

4.4.2. About us

This section of the user interface consists of 2 webpages, Roxana Sustic and Lucian Ciugudean, offering details about the creators of the website. Both pages are build in a similar way, displaying content composed of different types of objects using cards. Icons are used as hyperlinks to the corresponding Facebook and Linkedin pages and the CVs of the two. Photos with captions were also added for extra contextual information without losing the interactive interface feeling. A preview of the webpages is presented in the figures below.

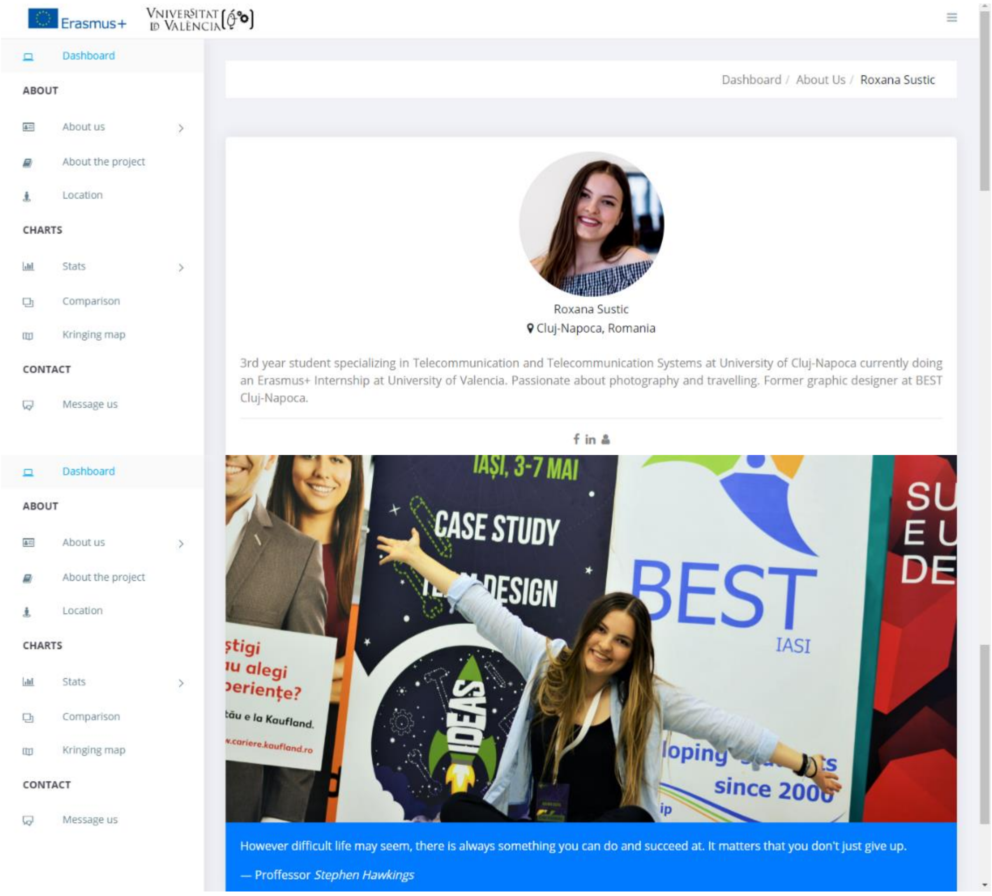

Figure 4.4.2.1 Roxana Sustic

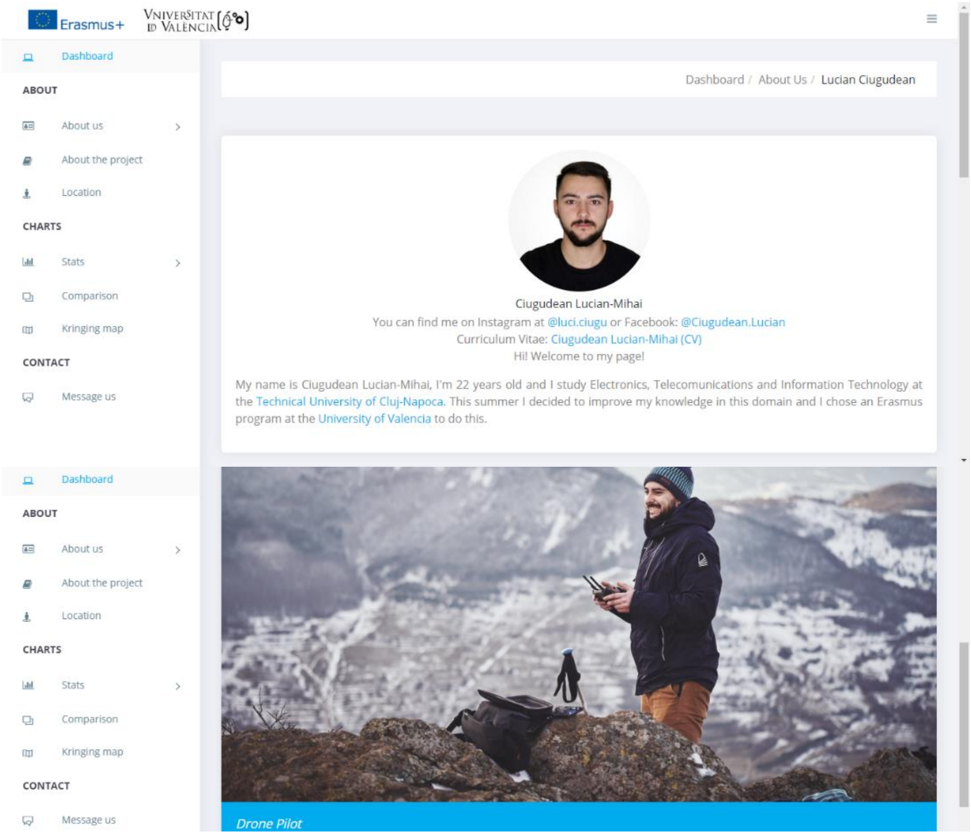

Figure 4.4.2.2 Lucian Ciugudean

4.4.3. About the project

On this webpage a short description of the project is presented, as seen in the following figure.

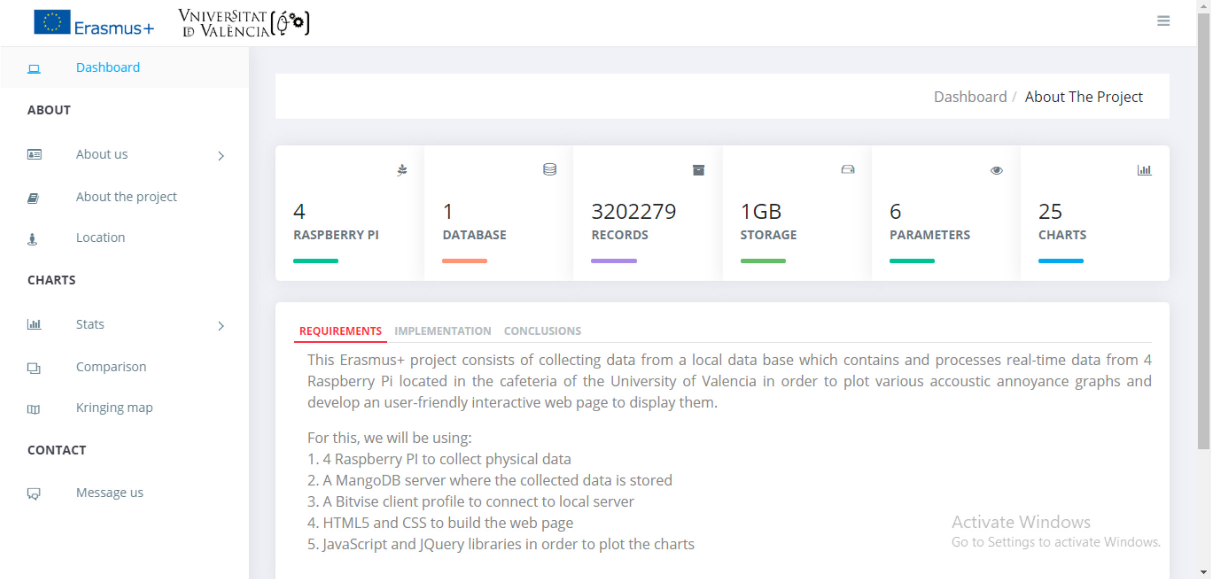

Figure 4.4.3.1 About the project

Animated widgets are used in order to engage the user, making the inteface more dynamic while providing important figures related to the project. More information has been included in tab cards, being a great way to display more content without loosing important screen space.

4.4.4. Location

An inline frame was used to embed an interactive Google map within the webpage. The map shows the location of the ETSE Cafeteria where the noise monitoring devices are installed, presented in t he figure below. It comes with both street view and satellite view.

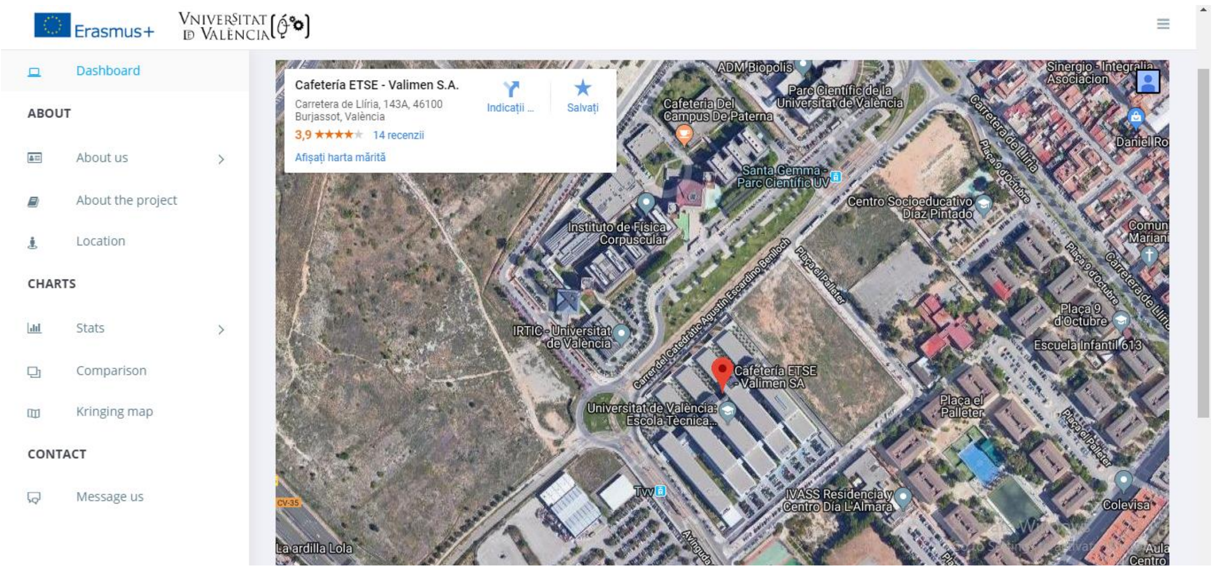

Figure 4.4.4.1 Location

4.5. Charts

4.5.1. Stats

Here, on this section, using a dropdown menu, the user can choose between the 4 Raspberry PI located in the Cafeteria.

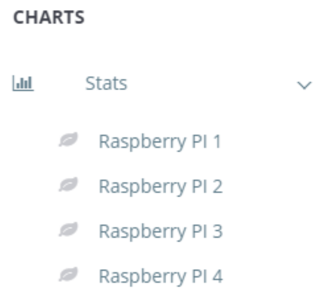

Figure 4.5.1.1 Dropdown menu for switching between each Raspberry PI

Using the data stored in the MongoDB database, it is created a graph showing the change of each parameter throughout an entire day, or between an interval that can be choose by the user.

Figure 4.5.1.2 Stats Chart example showing the loudness of the first Raspberry PI

Using the data stored in the MongoDB database, it is created a graph showing the change of each parameter throughout an entire day, or between an interval that can be choose by the user.

Each page has 6 sections for all 6 paramters, so that the user can check evolution depending on what parameter he's interested in, for any of the Raspberry PI.

Figure 4.5.1.2 Parameter selection section

The normal aggrecation for a new set of data is around 1 second, for each Raspberry PI and for each parameter. In order to not use too much data from the database, preventing a possible crash or slow loading of the web page, the algorithm that extracts the data is designed to not get all the data for the selected interval, such that:

If you select 1 day, the aggregation it's 5min.

If you select 2,3,4 days, the aggregation it's 15min.

If you select 5 or more days the aggregation its 1hour.

The PLOT button reloads the graph with the new interval specified by the user.

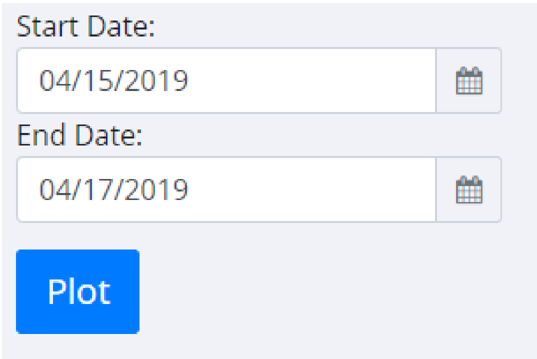

Figure 4.5.1.3 Interval date selection

4.5.2. Comparison

This section offers a live comparison of each parameter average value of a day.

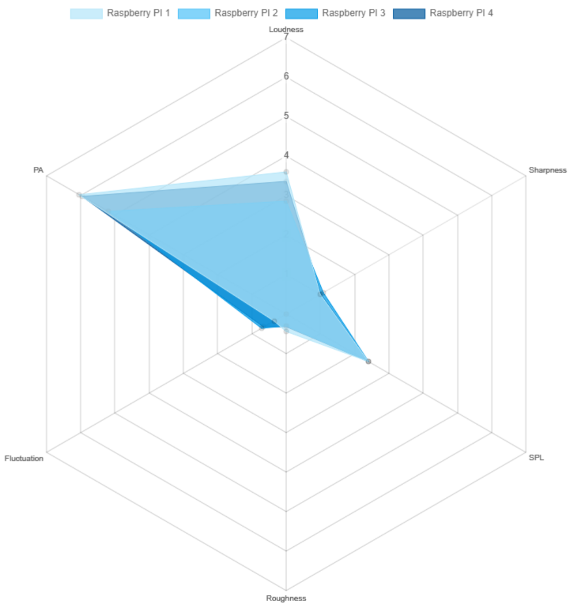

Figure 4.5.2.1 Comparison Graph

The user can eliminate the unwanted Raspberry PI by selecting which one he's not interested in and the graph will show only the parameters from the remaining Raspberry PI.

The program automatically extracts all the data from the database for the selected day and it calculates the average medium of each parameter of that day.

4.5.3. Kringing map

Kriging is a group of statistical techniques to interpolate the value of a random field at an unobserved location from observations of its value at nearby locations. The method is widely used in the domain of spatial analysis and computer experiments.

Using this method, the user can see and understand easily what those Raspberry PI actually give us informations about. The web page is showing the map of the cafeteria(Figure 3.1.1.) and the position of eash 4 Raspberry PI.

The web page offers the user a dropdown menu for selecting the parameters that he's interested in.

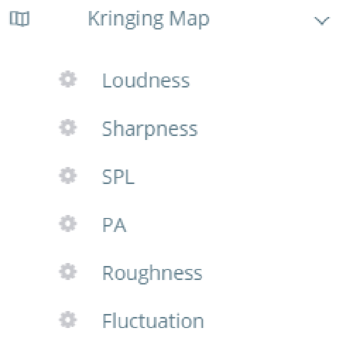

Figure 4.5.3.1 Dropdown menu for selecting the parameter of the Kringing map

The program calculates the average medium of the paramteter for which the user wants to see the Kringing map of the chosen day. It does that for all 4 Raspberry PI and is placing the value at the exact position where the Raspberry PI is situated.

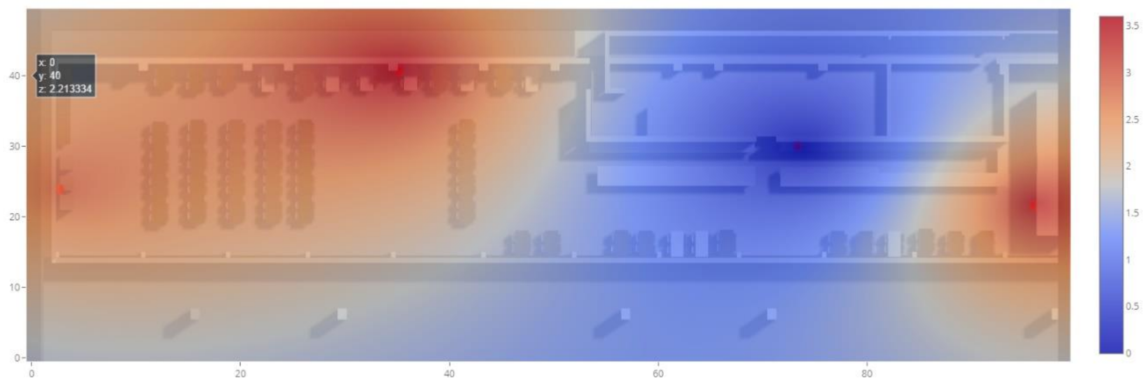

Figure 4.5.3.2 Kringing map example for loudness at 04.15.2019

This user-friendly method of expressing the meaning of the paramters can help anybody understand way much easier why are the Raspberry PI used for this kind of operations.

In order to plot this matrix of data (contour/heatmap), it was used an external library, called plotly.js, which is a charting module for Python. It can create publication-quality charts. It supports many types of charts/plots including line charts, bar charts, bubble charts and many more.