Internship Report

Master in Informatics Engineering Mobile Computing.

Quantum Portal

Pooja Pujari

Leiria, October 2016

Internship Report

Master in Informatics Engineering Mobile Computing.

Quantum Portal

Pooja Pujari

Report developed under the supervision of **Professor. Nuno Costa**, Professor of the School of Technology and Management of the Polytechnic Institute of Leiria.

Leiria, October 2016

Acknowledgements

Currently I am completing a Master's degree in Computer Science-Mobile Computing from the School of Technology and Management of the Polytechnic Institute of Leiria. I am in the second year of Master's therefore I have to conduct a 9-months internship. The reason I choose to do my internship was to develop new sets of programming skills.

For this opportunity, I thank:

Samuel Silva, who is the CEO of Domatica Global Solution's. I want to thank him for giving me the opportunity to do my internship at his company. He had the kindness to accept me in his company and guide me through my internship with advice, feedback and tips despite his busy schedule.

Pedro Pina, who is the Administrator of the Domatica Global Solution's. I want to thank him for giving me a support in the work time. He uses to motivate and appreciate the work I have done in the internship and that motivates me to learn more and more.

Carlos, who is our colleague working on Hardware part in the Domatica Global solution's. I want to thank him for help. He is so friendly that he uses to take us every week to our main Branch, which is in Lisbon, used to feel it as week trip for us. He uses to help at any time during holidays.

Carlos Grilo, who is the master's Coordinator. He helped me a lot in the first year and also gave me some assignments to do in JAVA because my background skills are Electrical. Hence, all high-level programming languages are unknown to me. I can say that my first step towards real software

development started during internship. He helped me a lot by teaching me how to handle the situations. He had always time to answer all my emails during his busy schedule.

Nuno Costa, who is my school's internship coach. He helped and coached me during my internship by giving me feedback and tips on how to handle and approach situations. He had always time to answer all my questions about my internship, specially my internship report.

Resumo

Este relatório enquadra-se no curso de Mestrado em Engenharia Informática - Computação Móvel da Escola de Tecnologia e Gestão do Instituto Politécnico de Leiria.

O objetivo deste relatório é descrever todo o processo em relação ao estágio de 9 meses que foi realizado na empresa Domatica, bem como o trabalho que foi desenvolvido durante o estágio. Este relatório apresenta informações sobre o local do estágio, o trabalho desenvolvido durante o período de estágio e as conclusões extraídas após a conclusão do estágio.

Todo o trabalho realizado foi dedicado ao portal Quantum node. Quantum Node Portal funciona como um Portal de Gestão de Informação e Dispositivos. Ele fornece aos clientes instalações como monitorar nós que estão localizados em diferentes locais.

O nosso portal de projetos lida com informação sobre os Quantum Nodes dos vários clientes. A informação inclui o status (on-line / off-line), a localização, a visão dos detalhes dos nós, para reivindicar os nós pelo processo de autenticação e também o gerenciamento dos nós e contas de usuários, armazenando a informação atual do nó atualizada pelos clientes, focando principalmente na apresentação de dados visuais sob a forma de tabelas e gráficos e fornece segurança para o Portal Quantum.

Palavras-chave: (Laravel, Web Site, PHP, JavaScript, Framework, Bootstrap)

List of figures

List of tables

List of Abbreviations

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
IPQ	Instituto Portuguese Quality Institute
JSON	JavaScript Object Notation
KPI	Key Performance Indicators
MVC	Model-View-Controller
OEE	Overall Equipment Effectiveness
PHP	Hypertext Preprocessor
REST	Representational State Transfer
SQL	Structured Query Language
VPN	Virtual Private Network
WAMP	Windows, Apache, MySQL, PHP

Table of Content

1. Introduction

As a full time student at Computer Engineering - Mobile Computing from the School of Technology and Management of the Polytechnic Institute of Leiria I passed the last 9 Months in an internship. The programming languages and development tools used at the Internship were all new for me, hence I had to quickly review and learn them. This was very beneficial for me as at the end I could compare what I have learnt with what I already knew and find a connection between the two. Since I find myself more interested in the area of web development and design, I was happy to be assigned the position of a trainee in web development in a Domatica Corporation solution.

Next in this chapter details of my team activities are given. Afterwards, I am going to explain my role and tasks as trainee and give specific technical details about my main tasks.

1.1. Stage Characterization

In this section, a presentation is made about the company where the internship took place, informing about its history and working conditions, as well as the general aims of the internship.

1.1.1. Company

Domatica Global Solution is a technology company founded in 2002, specialized in engineering solutions, with a highly qualified team of different backgrounds with the strong focus on development of a robust framework for M2M/IoT solutions.

Domatica has created a middleware for the IoT (Internet of Things), the Domatica Quantum System, and an Application Enablement Platform (AEP) that enables the fast deployment of industrial M2M/IoT solutions. This IoT framework can be used by non-specialist developers and facilitates bridging between devices and software applications, by enabling connectivity, remote

device management, flawless data collection activities and local processing power capabilities.

Tailored made M2M/IoT solutions that address key challenges for our societies are now possible to be quickly deployed at 10 times faster, reducing engineering complexity and costs. Domatica's IoT framework is suitable for Software Developers, Solution Providers, Hardware Manufacturers and System Integrators, whom pursue to deliver disruptive services, solutions and new business models in key areas such as Agriculture, Building Automation, Energy Management, Industry 4.0 or Smart Cities, among other vertical markets.

Domatica's technology has been implemented in several countries from different continents for instance, Portugal, Spain, Ireland, Italy, UK, Russia, Emirates, Kuwait, South Africa, China, Brazil, Mexico and EUA.

Domatica Software Partners are "iRidium mobile, SYSDEVD KALIPSO, EXAKOM AUTOMATION, iDom Live"

Skilled programmers are able to develop applications for software platforms, using high level programming languages and databases. However, they usually face significant barriers to get out of the computing environment when they want to control physical devices. Imagine a world where units of measurement appear directly on your application, no matter what sensor you use and not having to deal with different electrical signals depending on the sensor or actuators.

Now Developers can focus on what they do best by using the Domatica IoT Framework to create automation applications in an easy way and gain competitive advantages by improving quality while speeding delivery time and reducing costs.

On my first day, I was surprised when I was brought to the internship company. It was a house with small garden with an office. I remember Carlos

who is our colleague so kind hearted. He treats interns as his friends. At the beginning, I definitely had to accept the idea of working in a house, with all the smalls spaces and the house environment. But after a while I got used to it and felt comfortable.

The colleagues that I have worked with were very nice and helpful. Domatica staff came from different studies and have different specialties. Some of them are also international students like me. This is very useful because this way we can learn and share the knowledge with each other.

The tasks and project that I was assigned to were very interesting. At the beginning, I noticed that the work and communication was done very fast. I also noticed the different communications tools they were working with such as Asana, Cisco Spark and Skype. This was a great opportunity for me to learn because I had little knowledge of these communication tools.

Overall my introduction with the company went really well. The work fits my education and I felt I could learn a lot. I felt really accepted in the company.

1.1.2. Purpose of the internship

Now a days IoT development projects are everywhere, and affordable, and advanced technology is the driving force behind this fast-growing phenomenon. Smaller, more accessible hardware and flexibility to use common programming languages make it easier than ever before to develop these embedded IoT systems. From hobbyists programming their own single-board computers to companies developing devices that we can control from our mobile devices shows that IoT is rapidly expanding. IoT is very useful nowadays no matters if you're creating a quick prototype or an entire IoT-powered business application. Here's a look at the small but incredibly smart technology of IoT development. We can cover IoT data, hardware, and software considerations plus the most popular IoT skills on the rise. Cloud technologies are based on the Internet. Where in the past, people would run applications or

programs from software downloaded on a physical computer or server in their building, cloud computing allows people access to the same kinds of applications through the Internet. These are flexible, work from anywhere and sensitive to automatic software updates. Hence many advantages are there.

Coming to web programming, it refers to the writing, markup and coding involved in web development, which includes web content, web client and server scripting and network security. The most common languages used for web programming are XML, HTML, JavaScript, Perl and PHP. Web programming is different from just programming, which requires interdisciplinary knowledge on the application such as, client and server scripting, and database technology. Now a days these technologies are key technologies to the companies.

One of the main objective of an internship is to expose you to a particular job and profession in industry. While it might have an idea about what a job is like, you won't know until you actually perform it. If it's what you thought it was, internship give you a taste of what a profession is like, and more over I was interested in software development and found it is a good opportunity to explore my career.

The reason I choose to do internship was because I wanted to improve myself in the software development field and to benefit from the experience. I wanted a new challenge in order to learn, improve and develop new sets of skills. During my internship, I got some major competences. Working as a Project Developer performing duties and activities and analyzing them from different perspectives. I was Electrical student in my Bachelors, hence starting the internship was difficult. So I started with small things on HTML then I felt I can do developing and I have done. And finally, I had to set up personal targets to develop and improve of my skills.

1.1.3. Working Conditions

The work carried out during the internship was developed on a laptop computer provided by Domatica with the Windows 10 operating system.

At the beginning of the stage a PHP (Hypertext Preprocessor) server was installed for the development and hosting of WebPortal using WAMP software (Windows, Apache, MySQL, PHP) [1][2]. This server was setup on a company computer that was accessed through the local network.

In an early phase of the development process the exchange of data between the PHP server and the database was performed using a REST (Representational State Transfer) API. This API was accessed through a proxy software developed by Domatica that was also hosted on the PHP server computer. Several databases with real data from Domatica clients were also used to validate the operation of the application and the data representation.

For the code writing the text editor Sublime Text was chosen and as a project management tool the Asana was used since it was the tool already used in the company.

1.2. Motivation and Objectives

I am currently pursuing my Masters in Mobile Computing and at our "Instituto Politécnico de Leiria" we understand the importance of gaining practical knowledge which will complement the textbook knowledge and help a student to gain a wider perspective of the subjects. Internships are designed to expand the depth and breadth of academic learning. I felt that it is an opportunity for me to receive experience in applying theories learned in the classroom to specific experience in the community and work world. I felt that it can also heighten awareness of community issues. I had little knowledge about HTML, CSS, and JavaScript which will be the boost for the project so I felt that can make something better and can learn new things.

The specific goals to be achieved with this internship are the following:

1. Learning web technologies (Apache, Node.js, Laravel Web Framework, REST API's)
2. Implementation of a web portal to handle quantum nodes and service system file from the server. It must handle user information which is stored in the cloud using REST API's. The portal must also handle and manage node details, and must be implemented using Laravel Framework

In order to achieve the previous main goals the work process must follow the following guidelines.

- Each feature to the website will be planned as a small project, with a time line. Responsible people will be identified for creating the solution, creating documentation and for adding and maintaining the content and for monitoring the progress.
- Each feature must be described as a business case with a clear description of the benefits to the contributors. This will help to have a clear goal and to motivate the users to add the content to the portal. This also helps to measure progress and manage the development in achieving the envisioned goals.
- For each function on the portal there will be stepwise procedures for the users in order to achieve how they can perform the tasks to maintain the content.
- People use multiple computer system, each with a different username and password. It must be made clear to the users which username and password to use (not local network or Promise

password etc.), how to request an account and get permissions and how to change password, etc.

1.3. Internship Planning

Officially this internship started in November with the initialization phase. This document has been written to capture all project's relevant basic information and principles in order to manage it in the right way. At the time of training period (in October). I started to learn new technologies. Below there is a table that summaries all the tasks held during internship.

Table 1 - Assigned Tasks

1	Task Scheduled	Start	End
2	Learning Node.Js	21/10/2016	25/10/2016
3	Learning JavaScript	26/10/2016	28/10/2016
4	Learning PHP	31/10/2016	2/11/2016
5	Learning SQL	3/11/2016	4/11/2016
6	Learning Laravel	7/11/2016	9/11/2016
7	Learning Basics HTML	10/11/2016	11/11/2016
8	Learning CSS	14/11/2016	16/11/2016
9	Learning Bootstrap	17/11/2016	19/11/2016
10	Apache Server	21/11/2016	23/11/2016
11	Register-Page	24/11/2016	28/11/2016
12	Login-Page	29/11/2016	2/12/2016
13	RecoveryPassword	5/12/2016	9/12/2016
14	Authentication	12/12/2016	19/12/2016
15	ListNodes	20/12/2016	23/12/2016
16	UserManagement	28/12/2016	30/12/2016
17	CreateUser	2/1/2017	2/1/2017
18	Functions to create	3/1/2017	20/01/2017
19	Design Interface	23/1/2017	24/02/2017
20	ListNodes-Node	27/02/2017	14/03/2017
21	Functions in Maincontroller	15/03/2017	28/04/2017
22	Implementation	1/5/2017	Still going on

Here we used Asana for the task schedule, but before we used Skype for the conversation. Then we felt that we were not able to comment each other on our project and mistakes, then we thought of using the Asana which is helpfull for the task assigning and group conversation.

Since the internship beginning there was no detailed specification for the functionalities to be implemented (although there is a general idea of the final product). It was established that the implementation would be done as the order of priorities of the internship, such as an overview of front-end, back-end,

testing, implementation, etc. It was also expected that the back-end and front-end implementation would be tailored to the perceived need, primarily to test the validity and functionality of data presentation and API methods.

2. Used Technologies

This section describes the background technologies used directly or indirectly in the internship in order to develop the Quantum Portal. Every year, technological devices become faster, smaller, and smarter [30]. Cell phone holds more information than room-sized computers that took man to the moon.

2.1. The World Wide Web

Originally, the World Wide Web was imagined as a mean for sharing information where documents are linked together in an inter-network [3]. These documents were basically static [25]. When forms were introduced they helped users to interact with servers. Soon after, the first Web application was created with the born of server-side scripting language making dynamic generation of HTML documents possible [18]. Until now, the web technology has reached to a new standard where web applications have full abilities of desktop applications. A group of dynamically generated web pages is most common way to represent web applications [19]. The online market is emerging requiring developers to produce better featured application within less amount of time. Web application frameworks are created with this purpose in mind [20] [21]. As companies have applied and succeeded in saving time and boosting application's features, frameworks have proven their ability to cope with the market demands [4] [5].

2.2. PHP

PHP: Hypertext Preprocessor is a well-known scripting language often associated with web development even though it has other areas of usages [34] [35]. According to w3techs.com PHP is the most common used scripting languages on the Internet [40]. Many frameworks based on PHP popped up during the last decade [41]. Frameworks like Code Igniter, Symfony, Phalcon and Laravel are widely used and according to sitepoit.com they are four of the most promising frameworks in 2014.

When visiting Laravel's website, the first thing we see isslogans like beautiful code, rapidity and speed. Phalcon's Team claims that their framework is the fastest. Symfony's website says "High performance PHP framework for the web development". CodeIgniter's team states that their framework is "Powerful with a very small print".

A development aiming to develop a project where PHP is required as the development language and high performance is one of the highly prioritized requirements in the requirements specification. So how would the decisions be made on whether choosing the plain PHP stack or a PHP framework in the development? In the case of using the framework, which one of these above mentioned would be suitable bearing in mind the performance requirements? To answer these questions this study has been conducted in which two functionally equivalent blogging web applications have been developed and subjected to an experiment in which the performance of each of the mentioned frameworks is measured and evaluated. The first web application will be developed in plain PHP. The other one will be developed in the CodeIgniter PHP framework [31] [32]. After implementing the applications, performance metrics were measured on all these versions of the web application. Those measurements consist of (i) Execution time of CRUD-functionality, (ii) Memory usage for each of them. The result from the experiments were analyzed and interpreted in order to become the basis of the decision and conclusions of a study. Our study included more or less different types of strings like "web development evaluation" and "web development performance". In the findings, most of the literature are about the evaluation in general, not about performance in web development. The search for data was about performance evaluation and the selection of literature research in this field was not plentiful but easy to find [16]. The important is a good way on how they did the research to find out the information in the data [17].

2.3. XAMPP

Whether you're on Windows, Linux, or even Mac, the easiest way to get all of the necessary pieces of software for this tutorial is to install XAMPP, which includes a web server, PHP, and the MySQL database engine [7]. If you choose to go this route it will be easy [8]. In our Quantum portal development we used Apache and MySQL processor [14] [15]. Both work together a step which is already taken care of by XAMPP.

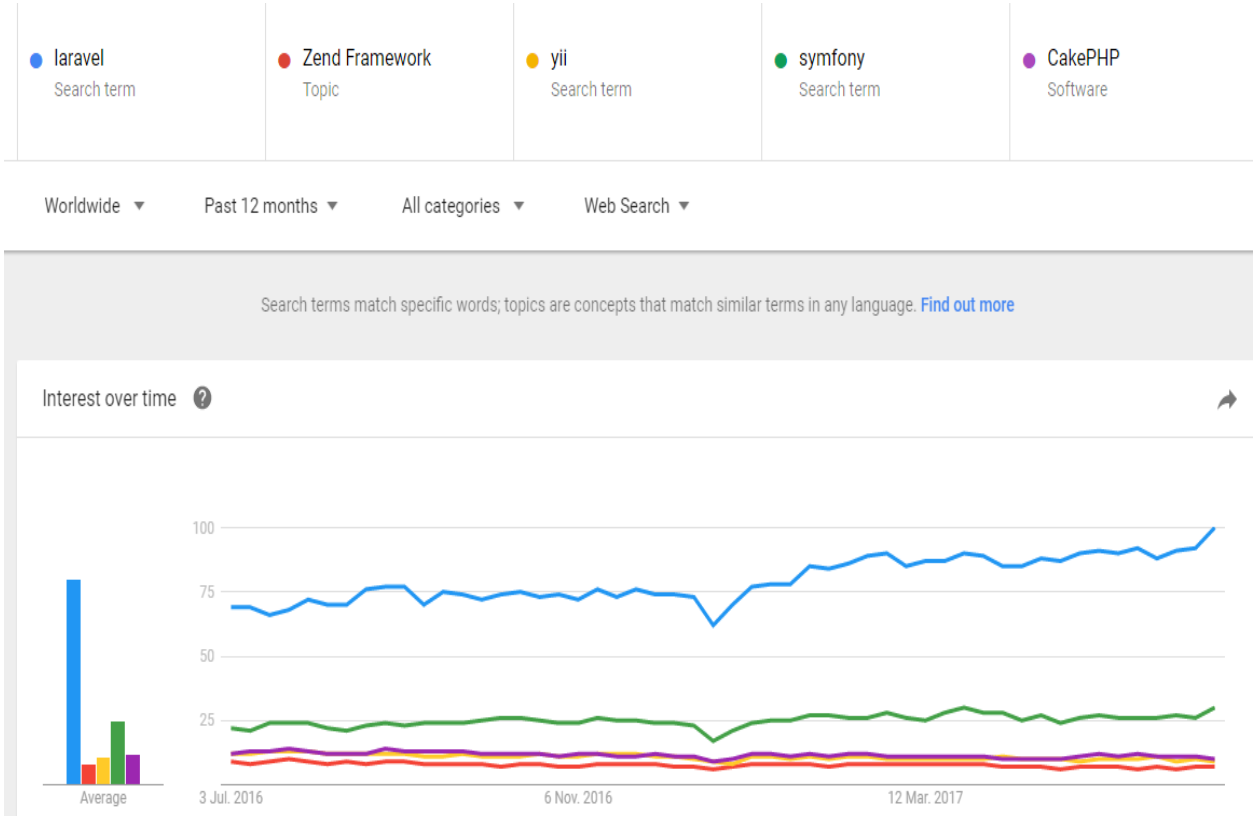
2.4. MySQL

Part of this project involves saving data to a database, so we need one of database where we can save the information [10]. In order to build our database on our PC and later export it to website, it requires setting up a localhost server on PC and finding an HTML editor that will synchronizes up with the server [9]. As we are using Windows, the process of setting up and running a localhost so that we can test and edit scripts and build databases is very easy [12] [13]. So, we have chosen the MySQL database.

2.5. Laravel

Laravel is an open-source PHP framework [71] for Web application development created by Taylor Otwell designed for the creation of system based on the MVC architectural standard.

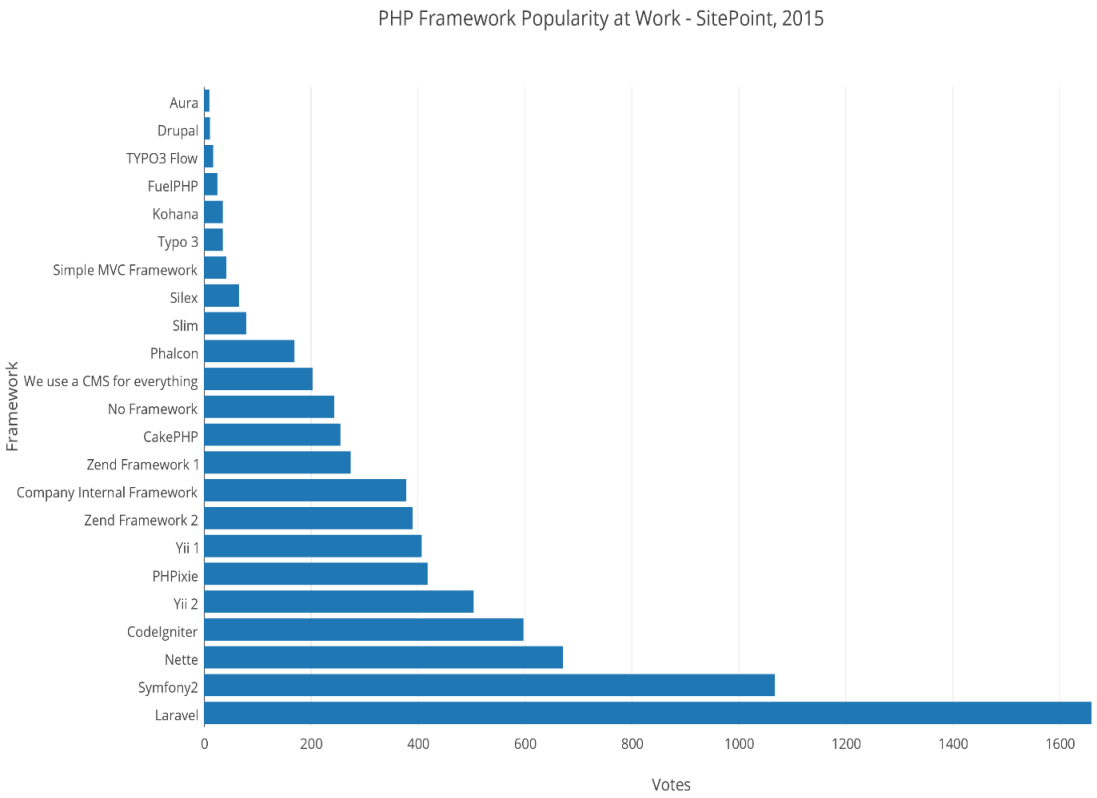
Figure 1 - Search trend of PHP frameworks in Google search engine.



Although there are other PHP frameworks, Laravel has enjoyed a great growth in popularity since its launch in 2011 (the above figure generated through google trend site), making it the best rated framework attributed by users of the GitHub website during its version in 2017 [72]. Figure above shows the trend of searches in the google search engine of some PHP frameworks from July 2016 to March 2017.

Since the Laravel framework was born in 2011, it is possible to perceive from the previous image that the growth of the interest of the programmers/software engineers was very fast that compared to a questionnaire made by the Site Point site [73], which lasted for one month and had a participation of around 7800 people. Laravel is the PHP framework most used both in personal projects and professional projects, as shown in figure below.

Figure 2 - PHP frameworks most used in professional Projects according to Site Point site questionnaire

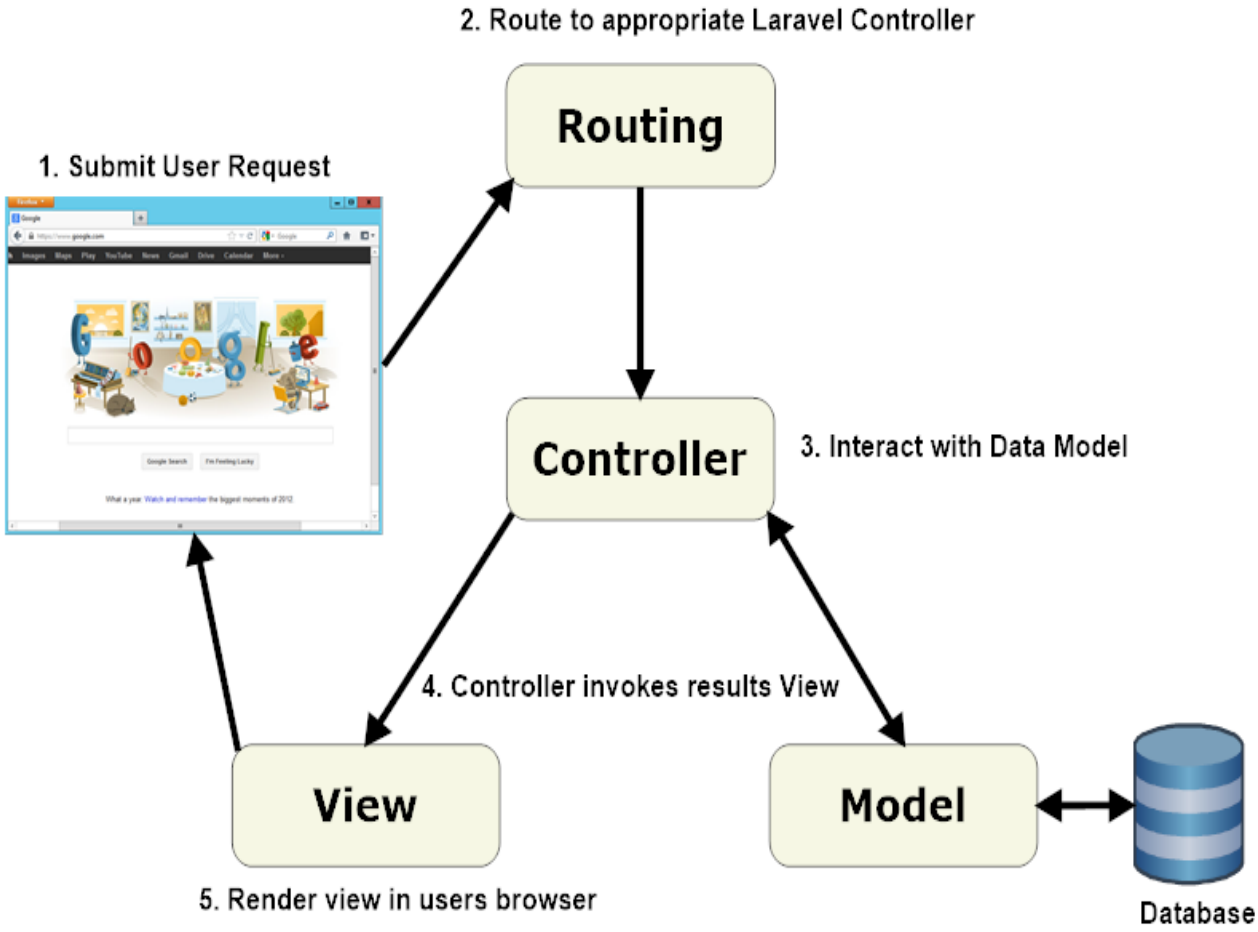


- **Operation**

The basic communication between Client and Server of Laravel is represented in the following figure and represents the following sequence;

1. The user makes an HTTP request to the server;
2. The request is handled by a routing mechanism that redirects to the appropriate controller (Main Controller);
3. The controller interacts with the model to obtain information;
4. The model accesses the database and responds to the controller with the necessary information;
5. The controller creates the view passing the data previously received;
6. The view is rendered and generates HTML that is sent to the client;

Figure 3 - Default MVC user by Laravel.



Laravel has an HTTP request filtering feature called Middleware (The above figure adapted from the Laravel book site). The middleware allows, for each request, to verify that certain conditions occur and to act according to the result. During the development of QUANTUM PORTAL this functionality was used to ensure that with each HTTP request the application language was always the same. This was necessary because the language was changed to the application default to each received request, as identified by several other programmers [74]. The use of Middleware is detailed in the next chapter.

Laravel features used

This section introduces some of the native features and extension of Laravel that were used during Web server development.

- **Blade**

The blade template system combines one or more templates and views with a dataset in order to create web pages. Blade has its own syntax that simplifies controller operations, such as conditions and loops, as well as a cache system of server-side views to improve performance. In the initial phase of the project, since the views only contained HTML code, it was necessary to adapt the views to the Blade syntax, such as imports of other files.

- **Artisan**

Artisan is a command line interface that allows you to perform various actions in a Laravel project such as viewing existing routes or creating files automatically, such as drivers or templates. In the development of QUANTUM PORTAL, the use of this feature mainly served to create the files for the configuration of the database through the use of the package.

- **Web Application Authentication and Authorization.**

Since access to the application required prior authentication by the user, Laravel's authentication and authorization system was used. This system uses filters that allow you to check if a certain condition occurs before calling a route. In this way, whenever a route is called, it is checked whether an authenticated user exists before redirecting the processing to a controller. If there are no authenticated users, the browser is redirected to the login window.

- **Laravel-crud-generator**

In order to manage the back office, a “Laravel-crud-generator” package was used [75]. This package is available for version 5 of Laravel and allows a fast creation of the necessary files for the managements of the database table (views, models, controllers, routes, etc.…) through artisan commands [75].

In contrast to the front end, the Backoffice communication with the database was done directly through tables or MySQL views instead of using the API developed by Domatica. Although it was understood that the method adopted was not correct since it was against the architecture previously defined, the decision was made at the time the development of this functionality began. The REST API did not provide the necessary functionalities at the time of the need.

In order to make the connection between the PHP server and the MySQL Server Database, it was necessary to install and configure specific external drivers as they were not included in the installed version of the WAMP software. Thus, the drivers provided by Microsoft [76] were downloaded and followed the instructions to configure the server correctly. Initially there were some doubts regarding the version of the drivers to use, as well as some errors during the configuration. However, after a process of debugging the errors I found it was possible to properly install the drivers on the server and perform the queries to the database.

Since Laravel-crud-generator is only prepared to generate MySQL commands, it was necessary to adapt the drivers to use SQL Server queries. It was also necessary to change the external file imports of the views once they were created through a template, as well as some general settings like the primary keys and the route names, among others, since the names of these fields in the database are different from the names expected by the plugin. Initially the files were changed manually, however, after it was considered

advantageous, the templates that Laravel-crud-generator used to create the files were adjusted, and then only small changes specific to each table were necessary.

- **jQuery**

jQuery [77] is an open-source cross-platform JavaScript library designed to simplify JavaScript code writing, focusing primarily on Document Object Model (DOM) manipulation, event manipulation (for example, clicks), Animation, CSS and AJAX requests.

It was decided to use this library because it is not only the most popular [70] [77] but also includes all the necessary features for the development of Quantum Portal. In addition, many of the plugins used during the internship are jQuery based, including many that were made available in the template provided at the beginning of the stage.

- **Date Range Picker**

A date range picker, pop-up calendar, date and time picker is a graphical user interface widget, which allows the user to select a date from a calendar or time from a time range. The typically practice is to provide a text box field which, when clicked upon to enter a date, pops up a calendar next to or below the field, allowing the user to populate the field with an appropriate date. Thus, it became necessary to use a solution that would allow it to do so intuitively. To do this, the jQuery date range picker [76] was used. This component uses three other frameworks/plugins: bootstrap for visual, jQuery for functionality and times.

In the initial development phase, there was also a problem with the status of the date and time since whenever the page was reloaded the dates shown were changed to a statically defined date. The solution was to maintain the state

by placing the chosen data in a hidden field. When requesting the server, the value of that date was recent, allowing the state (date value) to be maintained between the requests. Then the server would send that date back in the response and it would be written into the hidden fields in order to update the date in the Date Range Picker. At a later stage of development this problem was no longer relevant since the request was made by AJAX, and it is not necessary to reload the page.

- **jQuery – Locate(find)**

Given the need for support, it was necessary to implement a client-side translation solution since the pages could be upgraded through AJAX, and some client-side data needs to be translated. After some comparisons between various libraries and plugins, we chosen the "jQuery". Its easy implementation and operation were the factors that contributed most to this choice.

The method of operation of this plugin is by assigning the attribute data-locate to an HTML element that will contain text. Within this attribute the path to the desired string should be placed. All strings are defined in a JSON file.

The translation method should receive the name of the JSON file that contains the translated strings. The second parameter of this method is the language of the translation, and the characters used must be the same as those used in the JSON file name. So, considering that the language variable has the "pt" value, when calling the method of the previous source code excerpt, the library will try to find a file named "fileNamedpt.json".

In order to determine which language the user is currently using, a "meta" tag containing the current language (received by the server) has been created. Whenever translation method is called, the existing string in the HTML element is used to know which language the user is using, as shown in the below source code.

```
//HTML
<meta name='language' content='<?php echo App::getLocale() ?>'/>
```

```
//Javascript
var currentLanguage = $('meta[name="language"]').attr('content');
```

2.6. Web server

Web server respond to the client request in either of the following two ways:

- Sending file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database.

When client sends request for web page, the web server searches for the requested page. If requested page is found then it will send it to client with an HTTP response. If requested web page is not found, web server will send an HTTP response with a 404 error not found. If client has requested for some other resources then the web server will contact the application server and data store to construct the HTTP response.

Web server follows the approaches:

1. Concurrent approach
2. Single process event driven approach.

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by the following methods [72].

- Multi process
- Multi-threaded
- Hybrid method

Multi process

Web Server initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

Multi-threaded

It creates multiple single threaded process.

Hybrid method

It is the combination of above two approaches. In this approach, multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

The following web servers are most leading today:

Table 2 - Most leading web servers now a days

1. **Apache HTTP Server:**	This is the most popular web server in the world by the apache software foundation. It is open for all operating systems like Linux, Unix, Windows, Mac OSx and more
2. **Internet information services IIS:**	The internet information server IIS is a high-performance web server from Microsoft. This runs on windows NT/2000, 2003 and other networked platforms.

3. Sun java system web server:	This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free but not open source, it however, runs on windows, Linux and UNIX platforms.

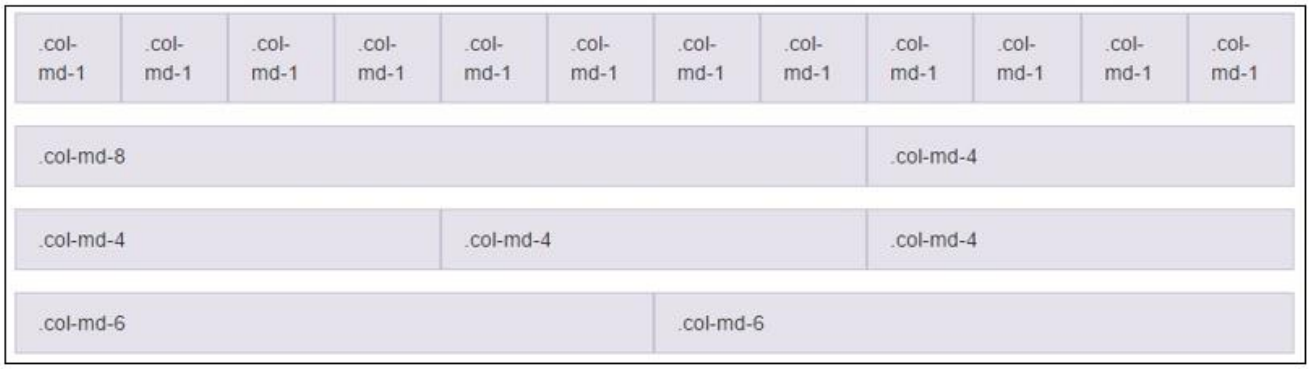
We used Apache server which is very convenient for the collection and sending request or data to the front office. The methods provided in the API developed by Domatica were used. For the back-office managements, the development team opted for a direct connection to the database since the API did not support the necessary methods and it was not expected that they would be completed in a timely manner. Thus, in the back-end office, communication with database was directly through Apache server through MySQL commands.

- **Bootstrap**

Bootstrap is the most popular front-end framework for constructing responsive web content. The purpose of this framework is to provide preprogrammed formations and layouts through CSS classes so that the programmer only needs to assign classes to the elements, knowing at the outset what these elements will look like. During the development of Quantum Portal various features and components made available by Bootstrap were used such as buttons, text boxes, tables, etc. One of the most used features and at the same time one of the most difficult features was the grid system [78]. Those difficulties were, however, overcome after a meeting with one of the collaborators of the company media web Creation. After that the grid system was adapted in almost every page of the project.

The grid system allows to divide a container up to twelve columns with equal space between them, being that when using the maximum division each column will occupy about 8.33% of the width of the parent element. The grid system offers specific classes for each screen size. The name of these classes always begins with "col-" and depending on the size you want, you can do this by "xs-" for very small screens (less than 768px), "sm-" for small screens (more than 768px) "For medium screens (over 992px) and" lg- "for large screens (over 1200px). Finally, a number between 1 and 12 should be used to determine the size that the element will occupy. Figure below shows an example of the use of this system, using various types of sizes for the elements.

Figure 4 - Grid System Example



If the element classes exceed 12 columns (eg col-md-6 and col-md-7), a new line will be created and the elements that do not have space in the top line will move to the bottom line.

2.7. JavaScript

All functionality of the modules has been programmed in JavaScript, including jQuery and AJAX. jQuery is a fast and small JavaScript library that offers many useful features that make event handling among other things much simpler with an easy to use API that works across a multitude of browsers.

AJAX, although not another programming language or library is a way of using existing standards. It is the art of exchanging data with a server and updating parts of a web page, without reload the entire web page.

2.8. Quantum Node

Quantum Node is the ultimate industrial IoT gateway. It narrows the distance between the physical and the logical world, easing real-time monitoring and control applications. It also empowers programming through the cloud, but keeping processing local, removing load from servers with the next generation IoT architecture, the fog computing.

Moreover, because of its dynamic data pipe, which adjusts traffic to demands, it is able to ensure remote real-time data acquisition with very low latency and very little data consumption.

Besides the embedded I/O capability, it allows creating a network of devices, either by adding Quantum Node Extenders or third-party models using standard protocols. Our aim is to create the portal for the Node in the cloud. This section presents the details about hardware Quantum Node as a communication concentrator, functioning as the bridge between the control devices and the SDK. In addition, it also handles communication with several peripheral and sub system protocols.

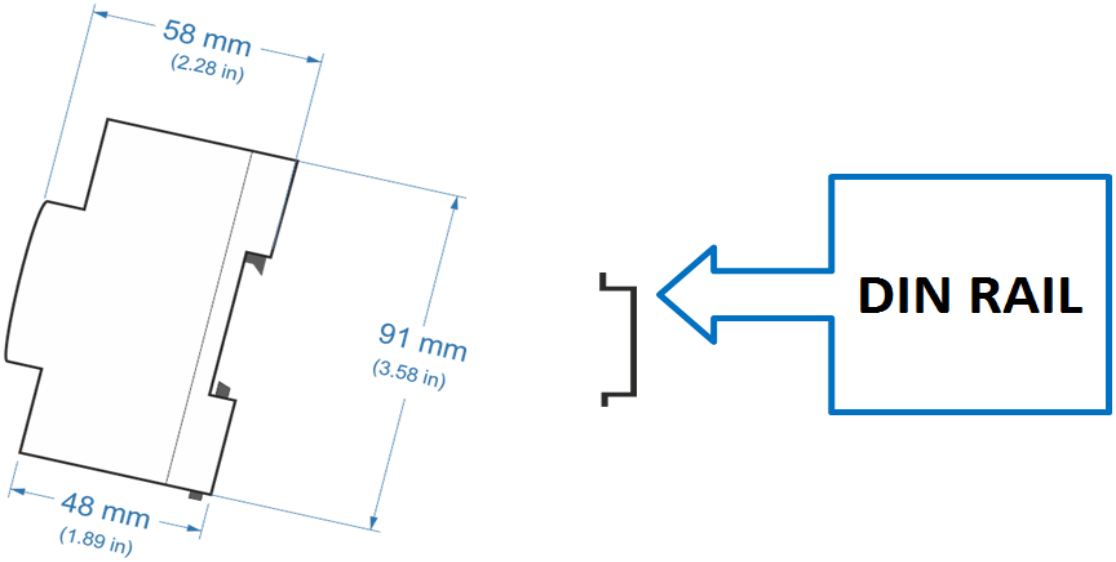
Although it is very common but not mandatory to include only one Node per network, it should not be mistaken with central system. Any Quantum Node module works in a distributed intelligent architecture, where each module has its program memory, object data memory (devices, timers, variables) as well as high processing power.

Key features:

- Object Server
- 6-channel inputs
- 6-channel outputs
- 1 Ethernet port

- 1 RS-485 port and 1 RS-232 port
- Domatica BUS for I/O expansion (up to 1 Mbit)
- RTC – Real Time Clock
- 150 000 register log (circular log) *
- Local data processor/Off line processor
- Built-in protocols*
- Modbus (up to 1000 Modbus registers)
- KNX (up to 1000 KNX group addresses)
- DMX512
- SNMP
- HDL Buspro
- WebSockets
- REST API
- API/SDK available for Windows/Linux/Android/iOS

Figure 5 - Quantum Node Extender



The Quantum Node Extender is a programmable device which is controlled by programming and includes a control unit, with universal inputs and outputs for sensing and actuating, which works either alone or in a distributed intelligence network.

Quantum Node Extender is suitable for industrial Automation, Distributed Control System (DCS), Building Management System (BMS), Energy Management System (EMS), Home Automation (HA), other general monitoring/control solutions and M2M/IoT application.

The Quantum Node Extender works in a distributed intelligence architecture, where each module has its own program memory, object data memory (devices, timers, variables) and a dedicated high-power processor. When a module is added to a network, it adds power and resources to that network, instead of consuming resources from other modules. The Quantum Extender allows to virtually connect any device for reading or control. With internal processing capacity, it can operates in standalone mode or connected to other system via DomatiCAN Bus.

Domatica quantum node Gateway Pro3 is the ultimate industrial IoT gateway that narrows the distance between devices and software applications, easing real-time monitoring and control solutions.

Built in a Fog Computing architecture, DQN Gateway Pro3 empowers programming through the cloud, keeping processing locally, removing load from your servers. Its dynamic data pipe adjusts traffic to demands, ensuring remote real-time data acquisition with very low latency and very little data consumption.

Besides the embedded I/O capability, it allows creating a network of devices, either by adding DQN Extenders or third-party modules using standard protocols. It is suitable for building Automation, Energy Management,

Industrial 4.0, Smart Agriculture, Smart Cities, and other general M2M/IoT applications.

2.9. Cloud Protocols

The most of IoT solutions, even those ones living almost entirely on the edge need to integrate with cloud service or other IoT solution based on it. Since it is a requirement, we need to communicate using a cloud protocol as listed below:

- **MQTT:** Message Queue Telemetry Transport (MQTT) was introduced by IBM in 1999 and standardized by OASIS in 2013. It is designed to provide embedded connectivity between applications and middleware on one side and networks and communications on the other side. It follows a publish/subscribe model, where the system consists of three main components: publishers, subscribers, and a broker.
- **AMQP:** The Advanced Message Queuing Protocol (AMQP) is a protocol that was designed for financial industry. It runs over TCP and provides a publish/subscribe model which is similar to that of MQTT. The difference is that the broker is divided into two main components: exchange and queues. The exchange is responsible for receiving publisher messages and distributing them to queues based on pre-defined roles and conditions. Queues basically represent the topics and subscribed by subscribers which will get the sensory data whenever they are available in the queue.
- **CoAP:** The Constrained Application Protocol (CoAP) is another session layer protocol designed by IETF Constrained RESTful Environment (Core) working group to provide lightweight RESTful (HTTP) interface. Representational State Transfer (REST) is the standard interface between HTTP client and servers. However, for lightweight applications such as IoT, REST could result in significant overhead and power consumption.

CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints. It is built over UDP, instead of TCP commonly used in HTTP and has a light mechanism to provide reliability. CoAP architecture is divided into two main sublayers: messaging and request/response. The messaging sublayer is responsible for reliability and duplication of messages while the request/response sublayer is responsible for communication. As in HTTP, CoAP utilizes GET, PUT, POST, and DELETE messages requests to retrieve, create, update, and delete, respectively.

- **HTTP:** This is the standard protocol for web services and still will be used in IoT solutions. The overhead of this protocol is well known but we will continue use this protocol in some case when latency and bandwidth are not issues. We need also consider HTTP/2 and other protocols such as Google Protobuf and even CoAP which are based on HTTP. The most popular architectural style called RESTFul is widely used on mobile and web application and must be considered on IoT Solutions.

2.10. Development Methodology

The development methodology used during project development was Prototyping software. This methodology can be divided into 4 subtypes, with in which the prototyping software specifications were followed. This section presents a study on the used methodology and its subtypes.

The Prototyping software is a development methodology that aims at the rapid development of software to test the validity of requirements [50]. This methodology focuses on the creation of prototypes to test the various functionalities expected and follows the following steps [51].

1. Identify the basic requirements in terms of functionality and interface;
2. Develop the initial prototype for users/client's UI Design;

3. Examine the prototype and collect information about its status from the consumers;

Based on the step 3, the prototype may be back to the implementation phase until the final result matches the customer's expectations.

This methodology helps customers and system creators understand the requirements of the system. Through the use of the prototypes, the end user of the system can know how it will work, as well as detect failures or errors that may not have been found in the requirements phase. This allows a reduction of risks since a large part of the flaws can be found in an early phase of the project [52].

The types of developed prototypes can generally follow 2 dimensions [53]. The horizontal dimension is more oriented to the visual component of the software, seeking to give a wider view of the whole visual aspect that final product will take. These prototypes can also be used, for example, as business-level demonstrations to find potential customers or possible financing. The vertical dimension serves to elaborate a function in a more specific subsystem of the product. Vertical prototypes are intended to obtain exact details about a functionality so as to acquire data about its requirements, such as for the creation of the database, interception between systems, data processing, inputs and outputs, among others.

The Prototyping Software can be divided into 4 types, Throwaway Prototyping, Evolutionary Prototyping, Incremental Prototyping and Extreme Prototyping, which will be analyses next.

2.10.1. Throwaway Prototype Design

Throwaway Prototyping [54] refers to the creation of incomplete prototype designs quickly which will eventually be discarded instead of being part of the final product. After the preliminary requirements are collected, a

functional model is constructed in order to show the customer the expected final appearance.

The most important aspect of this method of prototype designing is the rapid creation of functional models, being able, from an early stage, to validate the requirements and interface of the system. This allows you to avoid costs and time in any changes to the system as well as address some of the problems related to identifying the requirements.

2.10.2. Evolutionary Prototype Design

The goal of evolutionary prototyping is to create a robust prototype that will serve as the basis for all other prototypes developed [55]. This initial prototype contemplates only the most important and well defined initial requirements, which is then improved as new requirements are defined and understood.

This methodology allows adding and changing functionalities that are not possible at an early stage of the development, either because the requirements are not well defined or understood or because other functionalities on which they depend are not developed. Thus, the initial prototype undergoes changes and evolved throughput the development, avoiding risks caused by the poor knowledge of the requirements.

2.10.3. Incremental Prototype Design

In this methodology prototype designs are created separately and independently for the various functionalities of the system [56]. After the prototypes are developed, they are, or are being integrated with each other in order to create the final product.

An advantage of this method is that the customer or end users have the opportunity to test the components developed and their functionality and can

provide feedback while other components are still in development influencing the end product.

2.10.4. Extreme Prototype Design

Extreme Prototyping is a methodology used for the creation of applications, especially oriented to Web applications [57] [58]. This methodology divides the development of a Web application in essentially 3 steps:

1. Creating static pages with HTML elements only;
2. Scheduling screens and HTML elements using data Simulated;
3. Implementation of services and functionalities and integration with the interface.

The goal of this methodology is to focus on the second step, developing a fully functional page before implementing the services. Once the page is fully functional it is possible to get feedback on your status and make any changes. In addition, this allows development teams to focus on their tasks separately, and make decisions in advance to ensure successful integration between the pages and services.

This was the methodology used during the development of Quantum Portal. The use of this methodology was due to the fact that, with in the Prototyping software, it is most suitable for creating web applications [59]. In the case of Quantum Portal, the tasks to be performed were usually oriented towards the representation of data, so they usually required the creation of new pages or the alternative of previously existing pages (usually from the template).

In practical terms, the creation of tasks was done in the preferred order of Domatica. At the beginning of each task it was discussed the functionalities, inputs / outputs and visual aspect of the window as per the UI. In terms of plugins for data representation, it was tried whenever possible to use those that

were already included in the base project, since they presented the desired visual aspect initially [60].

1. Domatica created the task and attributed it to the employee (trainee);
2. A task planning was done and the necessary input and outputs were studied;
3. If plugins and / or external extensions were needed, several possibilities were studied and tested, otherwise the alternatives already in the template were used;
4. If it was necessary to draw a new page or redesigned a page already existing in the template. The necessary changes a revision was made by Domatica. If the result was satisfactory, progress to the development phase, otherwise the necessary changes would be discussed and made until the desired result is achieved;
5. With the page created, functionalities were implemented using the chosen plugins (if necessary) making use of static data for testing purposes;
6. A review was again made by Domatica and made the necessary changes until the expected result is reached;
7. The code needed to load dynamic API and / or database data was implemented;
8. After the functionality was implemented, a final analysis was made to identify the necessary changes and to make the corresponding corrections;
9. After any corrections, the task was terminated, moving on to the next task.

3. Requirements and System Architecture

This chapter describes the analysis of system requirements, software architecture and user interface. The objective of the portal is to provide an innovative solution which facilitate the frequent and recurring use of online services by users for obtaining information and interacting with Quantum Node.

3.1. Functional requirements

Whenever a software application is built, be it for the web or not, the development team has to acquire certain knowledge about the problem domain and application's requirements. The elicitation and specification of these requirements is a complex process as it is necessary to identify the functionality that the system has to fulfill in order to satisfy the user's and customers' needs.

Although there is a lack of a standardized process supporting requirements handling and guaranteeing the quality of the results, best practice in the development of general software applications provide a set of techniques. Such techniques are also recommended by some web methodologies for requirements specification of web applications.

Portal Information

The portal should enable users to access published information and content about nodes.

Portal Browser Support

The portal must support different screen resolutions and the latest two version of most common internet browser (e.g. Internet Ex-plorer, Mozilla Firefox etc.) and it should also provide all graphics and images in the portal.

Portal Registration

Portal should support various registration models for different users including online identity verification. Registration should allow creating profile of user based on their information and usage of portal and applications. The portal should support single account creation for all users without duplicate accounts.

Authentication

The portal should provide authentication and single sign-on functionality with single username/password allowing various registered users groups to access authorized information and services/applications. The portal should provide this capability to manage user authentication (login), authorization to access content and services/application based on user profiles, roles, and access control. The portal should enable single sign-on across applications providing controlled access to services according to their profiles and roles.

Authentication is the process in which the credentials provided are compared to those on file in database of authorized users' information on a local operating system or within an authentication server. If the credentials match, the process is completed and the user is granted authorization for access.

Personalization

The portal should support personalization to select the node list and view the information about node. This should be provided for only the registered users. For unregistered users only the login page is presented where user can go for register page for his/her registration.

Web channel

The portal should allow various groups to access portal content and information about nodes using different web browsers via HTTP and HTTPS protocols.

Portal Search operations

The portal should provide users with basic and advanced search capabilities to help user to find information. User should be able to see his node status whether online/offline and to retrieve content of information about node.

Maintain customer profile

The portal should allow user to create his profile and set credential and also should allow user to update the profile information.

Data Storage

The customer's web browser shall never display a customer's password. It shall always be echoed with special characters representing typed characters. Password may be reset but never shown. The system back end databases shall only be accessible to authenticated administrators.

Documents on the portal are the most up to date version

The current collaboration on documents through sending the information about the nodes and status of each node should be an up to date.

Documents on the portal can be edited by multiple users, without conflicts in versions

Document repository can be opened for editing. The user can check out a document, so that user can edit his information in the portal, providing authority to the client through authentication method for safety use.

Personal Information

User register with the application providing all the necessary details. Then the user or the customer will have to provide all the necessary details. After all the details entered in the customer registration database will be verified. Then accepted and stored it in the database

Entity Information

Description about the Entity. Then entity name can be entered it should need VAT-ID for the authentication Serial number and registration code have to enter.

3.2. Non-functional Requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work with in. Some of them are shown below.

- Performance Requirements.
- Safety Requirements.
- Security Quality Attributes.
- Software Quality Attributes.
- Business Rules

After presenting functional requirements for the portal we specify a number of additional nonfunctional requirements. These requirements are user-visible aspects of the portal that are not directly related with the functional behavior.

3.2.1. Performance Requirements

To ensure meaning full data, the portal should not slow down the host-application to the point that its usability is affected.

3.2.2. Safety Requirements

All logged information, updates and user activities are backup at each day automatically.

3.2.3. Security Requirements

Any modification (insert, delete and update) for the Database shall be synchronized and done only by the System administrator.

3.2.4. Software Quality Attributes

Describe in the same way as previous! Why apply different format for the same thing? Remove these four bullets!

The system must be available during 24 hours of the day. And system shall provide the capability to back-up the data. User can log in to the system at any time. The system will not be broken down easily and will not wholly affect by a single application failure.

3.2.5. Reliability

In order to provide meaningful results, the portal should not make the host-application behave differently. Any side effects on the host-application should therefore be minimized and influence on the test user's behavior should be as low as possible.

3.2.6. Privacy

Recorded user interactions and other events should only be accessible to an authorized developer.

3.2.7. Supportability

Full source code comments should be provided for all portal code. The portal should be structured so that it can be easily extended with new capturing abilities, interpretation algorithms, and report section types.

Implementing the portal will start with the essential and keep the system simple. Initial version of the portal will contain the basic html pages and document download. Each new feature added to the website will be planned as a small project, with a time line. Responsible people will be identified for creating the solution, creating documentation and for adding and maintaining the content and for monitoring the progress. Each new feature must be described as a business case with a clear description of the benefits to the contributors. This will help to have a clear goal and to motivate the users to add

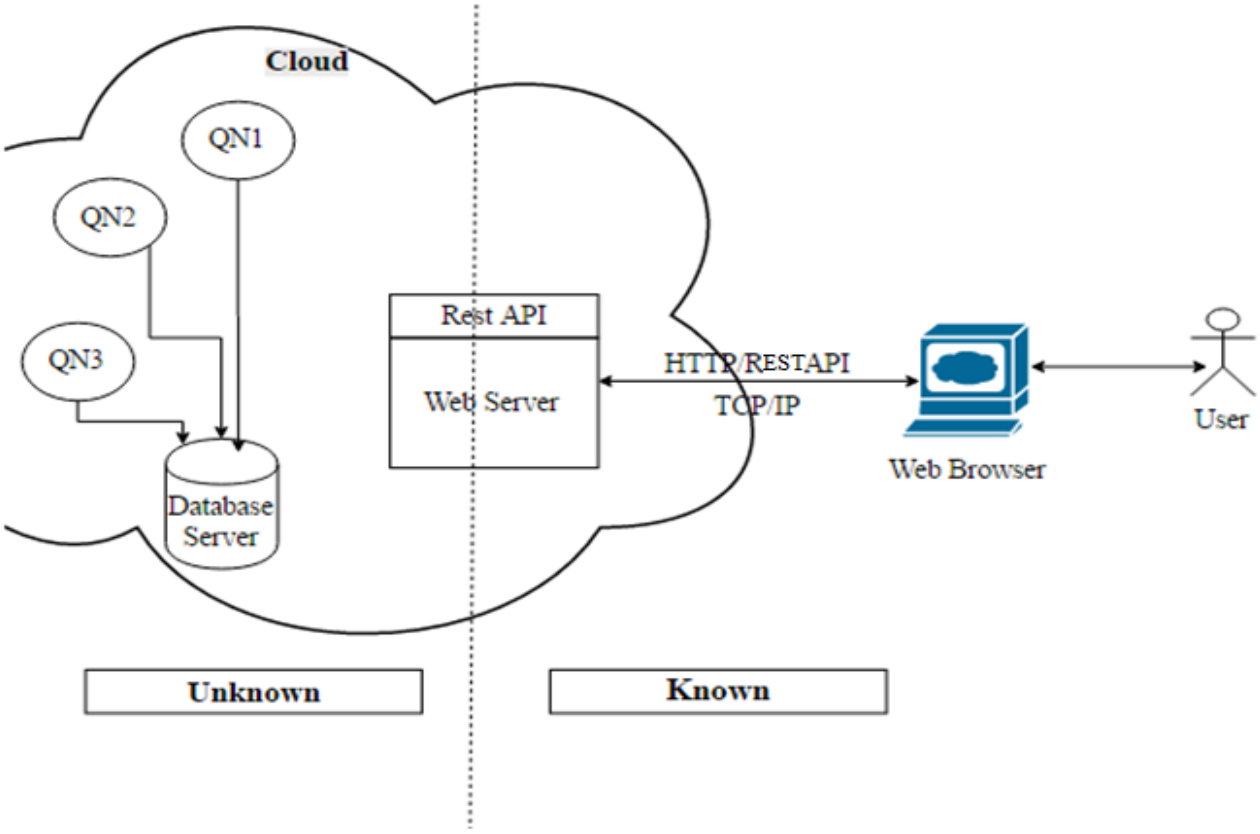
the content to the portal. This also helps to measure progress and manage the development in achieving the envisioned goals. For each function on the portal there will be stepwise procedures for the users on how they can perform the tasks to maintain the content. There should be adequate manuals, training and instructions for the editors of the web system on how they can use the system and what information they will put on the web site.

Changes in work processes should be easily implemented as changes in workflow in the software. Editing the texts on the pages should be simple. Committees should be able to maintain the content of basic web pages without the need for training. It should be intuitive and a simple manual should be sufficient. Advanced features such as workflow might require more instructions, though the aim is to keep content management as simple as possible. Different users have different tasks and permissions on the system. It should be clear to the user what tasks they are required to perform. The website should therefore present a clear list of tasks that user can perform on the initial page or in the common parts, so he does not have to search the site [71].

3.3. System architecture

This section is entitled system architecture and deals with concepts, design, and architecture of a content management system. Figure below explains the project work flow.

Figure 6 - System Architecture



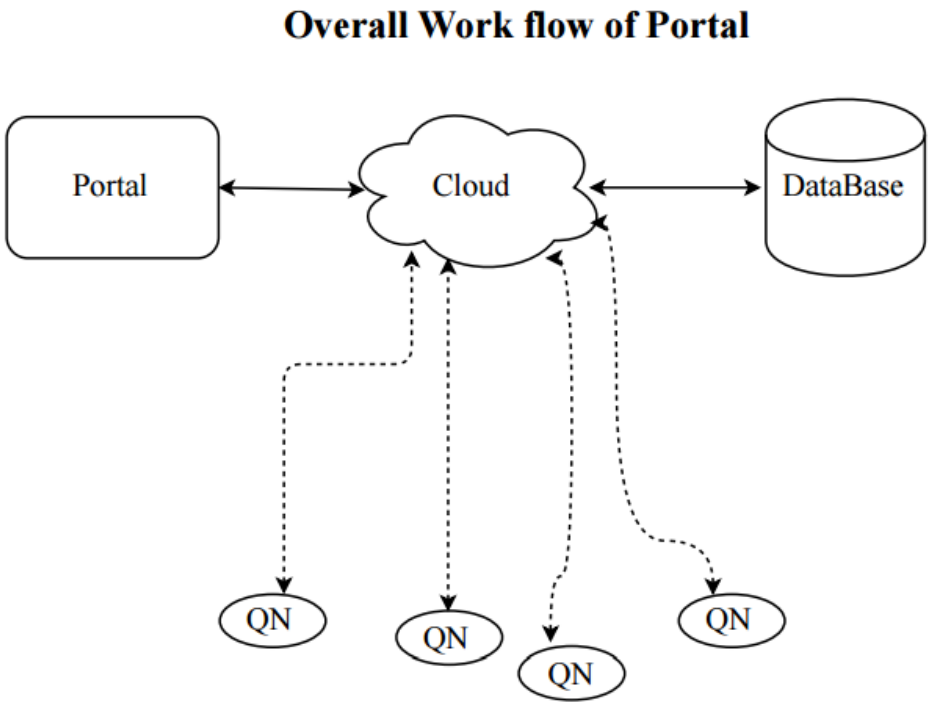
Information services have been growing rapidly with the rise of new technological innovations. The latest trend on the web is collaboration of users. One common approach in this direction is personalization. The aim of our project that is Quantum Portal is making a user feel as an integral part of a website.

Above figure can see user using the portal through the Internet in order to find information's about the node, which is may be stored in the cloud or local database. The request is done by using AJAX URL specification. By

adding the “url” to the code in controller, user will be getting information of the node.

Main aims of these systems were to make users easy access to the portal with a high security. An Internet portal is enabled by software executing on an Internet connected server. The portal, in response to a log on by user, presents a secure and personalized page to the user. The personalized page having listed plural internet destinations enabled by hyperlinks, wherein upon invocation of a hyperlink by the subscriber, such as by a point and click technique, the portal invokes a URL for the destination, and upon connection with the destination, transparently provides any required log on information for user access at the destination. In an enhanced embodiment, a search function is provided where in a user may configure searches. Once user enter into the portal with an authentication and secured email and password he will get information which is stored in the cloud (unknown). Here I am just imagining node information saved in a cloud. Administrator can create, add, delete or add new user. All functions are done by using cloud support.

Figure 7 – Quantum Portal Node Work Flow



Here I am thinking that Quantum nodes are directly connected to the cloud. In the figure above we can be see Quantum nodes connected to the cloud and hence their information will be stored in the cloud. When user browses the portal about the Nodes, cloud will send it to the user. Administrator of the portal can create, add, delete the user. All information will be saved into database which has been done by the administrator or clients. If any changes take place in the quantum node then it will be updated to the cloud and database simultaneously (unknown).

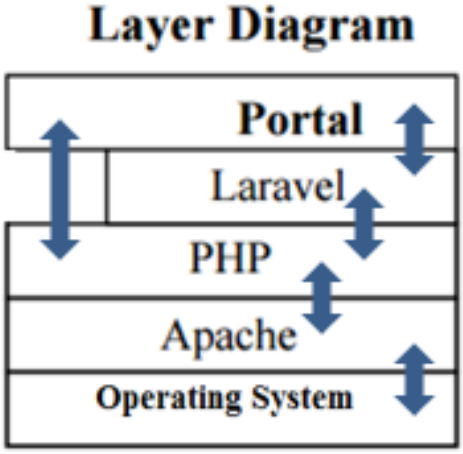
Quantum Node Portal works as an Information and Device Management Portal. It provides the clients with facilities like monitoring their nodes that are located at different places. Our project portal delivers information about the Quantum Nodes owned by the various Clients. Information includes status (online/offline), location, view of the details of the nodes, to claim the nodes by the authentication process and also, management of the nodes and user accounts, storing the node's current information updated by Clients, focusing

mainly on the presentation of visual data in the form of tables and graphs and provides high security for the Quantum Portal.

3.4. Layer Diagram

Layer Diagram helps us to visualize the high level, logical architecture of the system. A layer diagram organizes the physical artifacts in the system into logical, abstract groups called layers. These layers describe major tasks that the artifacts perform or the major components of system. Each layer can also contain layers that describe more detailed tasks.

Figure 8 - Layer Diagram of Quantum Portal.



3.4.1. Portal

A Web portal is a specially designed to bring information together from diverse sources in a uniform way like emails, forums, and search engines etc. Usually, each information source gets its dedicated area on the page for displaying information, our portal is to bring information about the specific node (which is situated in the cloud) saved in the cloud. Portal contents information's includes status, tariff, Billing, how many nodes are online and offline.

3.4.2. Laravel

Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, and caching.

- **Managing Database**

Migration can be considered as a form of version control for our database. They allow us to change the database schema and describe and record all those specific changes in a migration file. Each migration is usually associated with a schema builder to effort easily manage our portal database. A migration can also be reverted or "rolled back" using the same said file.

Using our terminal, we can issue the following commands to create or drop tables in our database:

Table 3 - A collection of commands related to migrations

Command	**Description**
php artisan migrate : install	Creates the migration repository
php artisan migrate:make	Creates a new migration file
php artisan migrate:refresh	Resets and reruns all the migrations
php artisan migrate:reset	Rollback all the database migrations
php artisan migrate:rollback	Rollback the last database migration

These are some commands which are helpful in future for the project to create database table to store the information of nodes or users. Before saving

the data, first we are thinking to save data and information locally using MySQL and Apache server.

3.4.4. Apache

Apache HTTP server is cross platform, meaning that it is built for Unix-like systems (e.g., macOS, Linux as well as windows). Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interface support Perl, Python, Tcl and PHP. Popular authentication module mod_access, mod_auth, mod_dihest, the successor to mod_digest. It has a feature that includes Secure Sockets Layer and Transport Layer Security support, custom logfiles, filtering support etc.

3.4.5. Web Server

A web server is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the world wide web. The term can refer to the entire system, or specifically to the software that accepts and supervises the HTTP requests.

The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheet and scripts in addition to content.

Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. This means that behavior of the web server can be scripted in separate files, while the actually server software remains unchanged. Web servers are not only used for serving the World Wide Web. They can also be embedded in devices such as printers,

routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring or administering the device in question. This usually means that no additional software has to be installed on the client computer, since only a web browser is required.

3.5. Quantum Portal Design

A key factor for a Web portal to succeed is the support it offers to its users in completing their tasks satisfactorily. We have discussed before start the work and I have made some designs before start representing the project. Quantum portal includes all information about the nodes mainly it's user interface design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. User interface brings together concepts from interaction design, visual design, and information about nodes.

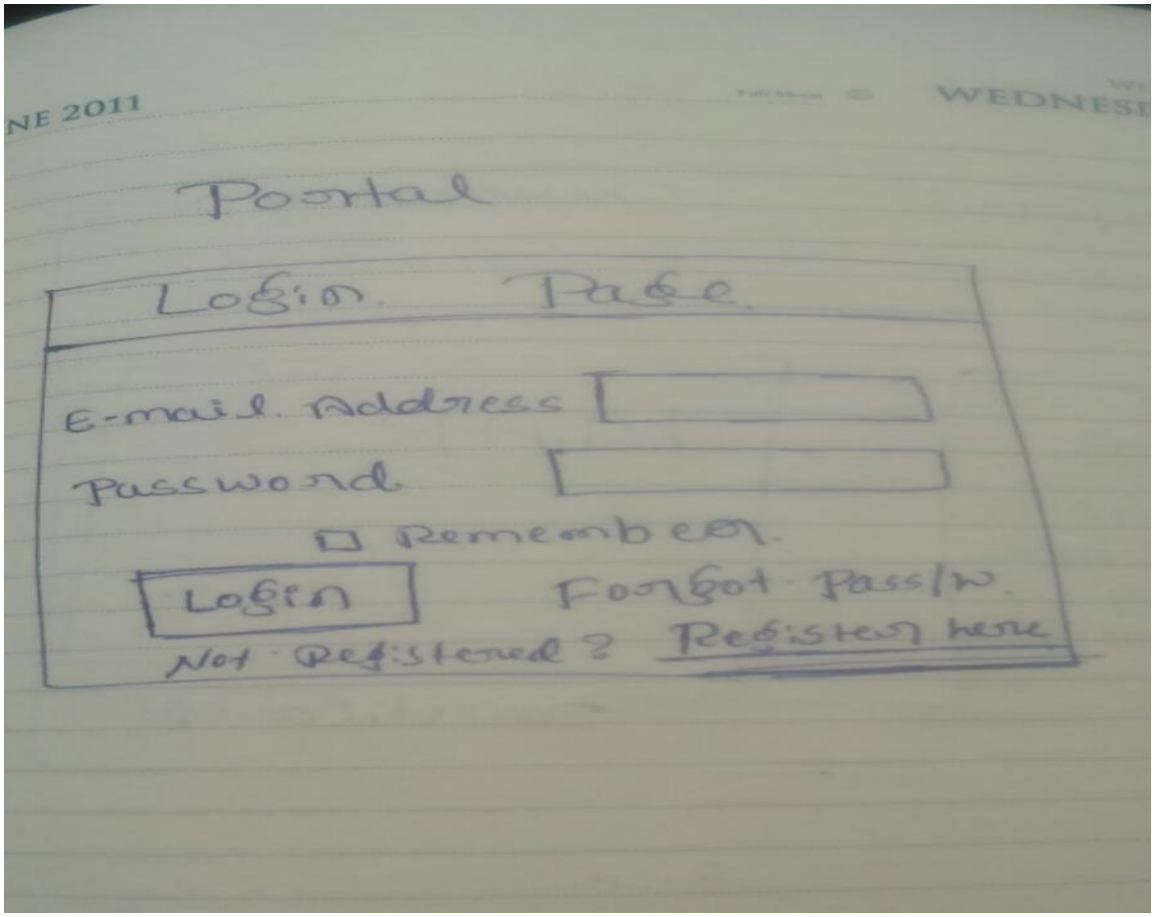
Quantum Node Portal works as Information and Device Management Portal. It provides the clients with facilities like monitoring their nodes that are located at different places.

Our project portal delivers information about the Quantum Nodes owned by the various Costumers. Information includes status (online/offline), location, view of the nodes' details, to claim the nodes by the authentication process and also management of the user accounts and the nodes, storing the node's current information updated by Costumers, focusing mainly on the presentation of visual data in the form of tables and graphs.

To achieve successful design and implementation of the Web Portal based user interface, it is essential to understand the requirements of the application and the technical background of the system. In order to achieve successful web application design process, technical details, which meet the requirements has

to be specified. This section analyzes the system's component and portal design with draws before going to the implementation.

Figure 9 - Draw for Login Form



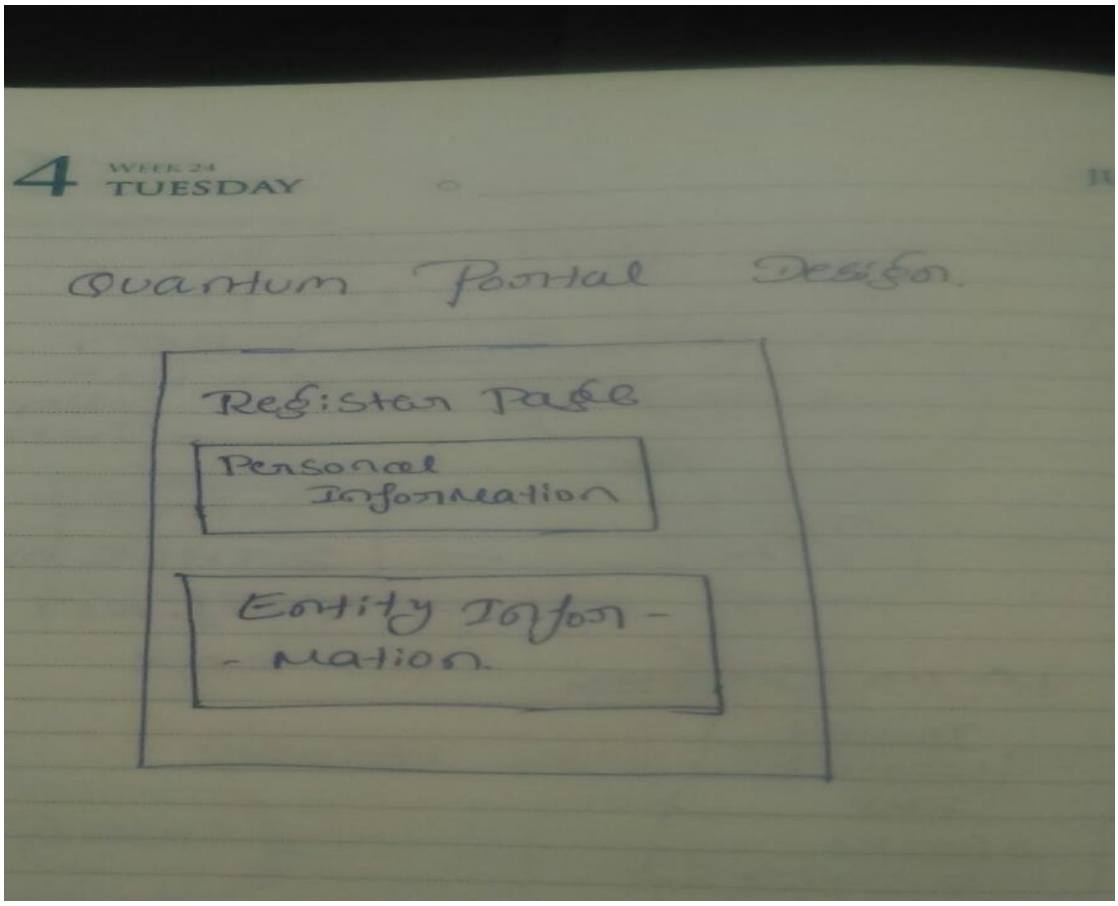
Login page includes email address, password, remember the password with check box, forgot password, not yet registered. These sketches are done for the personal improvements before starting the project

When building a login page or system for a website, there are many factors to take into consideration. For example, will the site support single sign-on? If so, for which email clients or services? Should we require users to setup security questions? Before delve into the details, it can be helpful to set up a high-level view of actors, use cases, and the system itself.

The Quantum Portal User Interface is built from components. The components are the following: the web portal design content and semantics are

HTML and CSS. JavaScript is used for the client-side scripts. The server-side technology is jQuery, MySQL database and REST API are used for the server-side and backend process.

Figure 10 - Draw for Registration Form



If the user is not registered before it must mention that he is not registered yet and should go to the registered page for the registration. In registration form we have divided it into two parts: the one is related to personal information and the second is related to Entity information. Personal information has content like First Name, Last Name, Phone Number, City, Country etc. The other one, entity information, consists of entity number, Vat, Serial number and register number, etc.

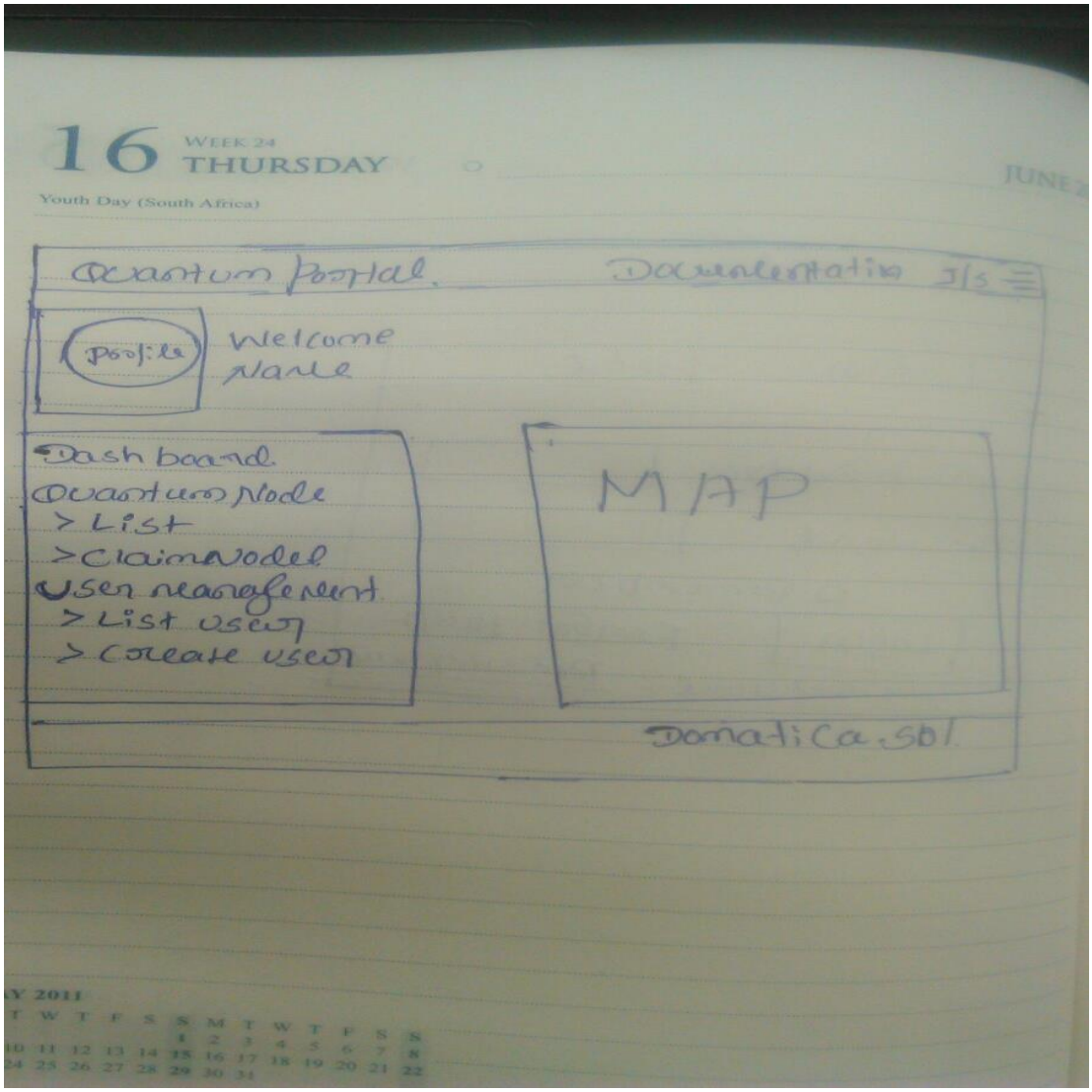
Here we have thought of making two types of login: one for admin and another one for User. Logging in is usually used to enter a specific page. Once the user is logged in, the login token may be used to track what actions the user

has taken while connected to the site. In system there are two types of login operations. One is administrator and second in the user. When administrator logged in to the portal he can perform some changes like below:

- Add New User
- Remove User
- Reset User
- Reset Password

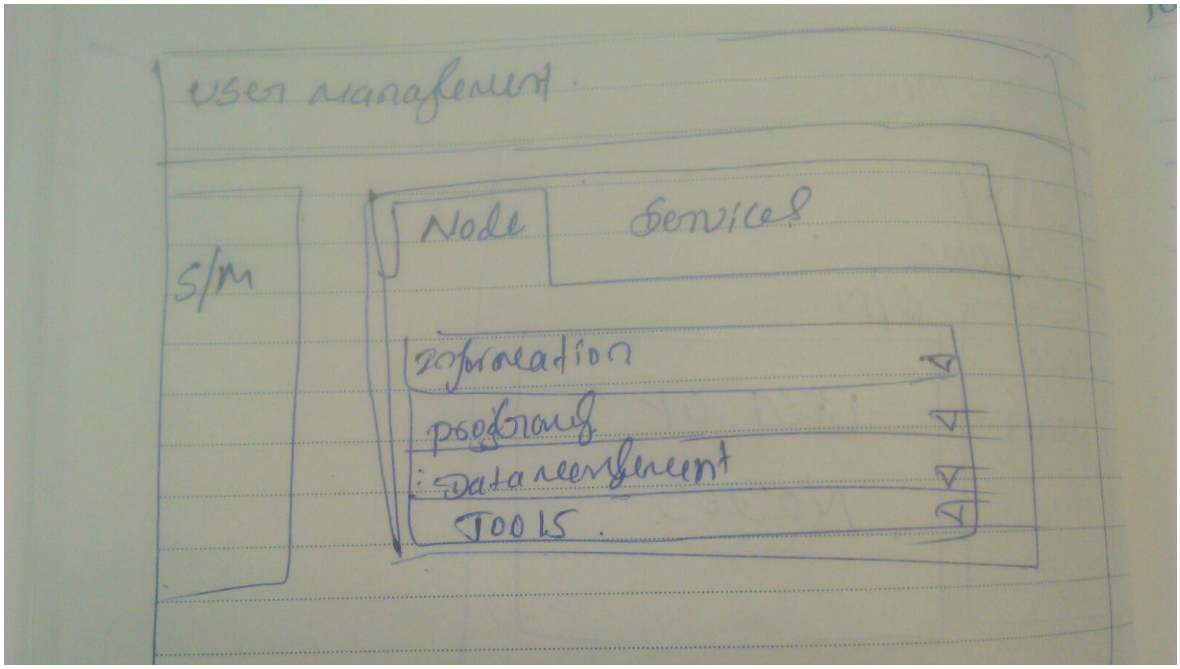
Administrator can add users directly or synchronize users. Once added, users can enroll devices and access company resources. The above figure explains the fields to be filled, such as the name and display name, email and password. Here comes the regular user or administrator. If the person has authority then has to choose the admin role, when details are filled it will compile with the database stored data. If data matches then it will allow the permission to be an admin. Then the admin will have authority to add new user, remove user and reset the information about the user.

Figure 11 - Draw for Welcome Page



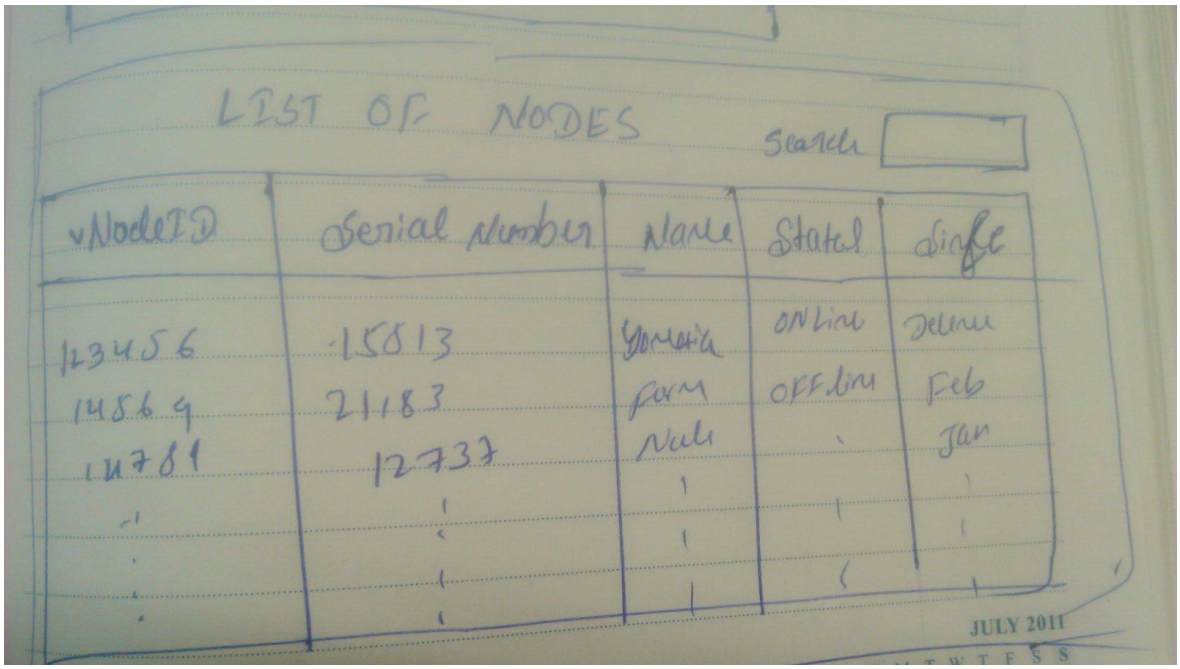
After login in the site, user can see the dashboard and Map where nodes are located. Connection between node and information view is done by clicking on the view node button. In the above draw we can see that dashboard contains the list, clime node, user management etc.

Figure 12 - Draw for User Management



Node information is shown by click on the view node button. The above image should open to see the nodes information and services provided by the node.

Figure 13 - Sketch for List Nodes



This is the page where we can see the list of nodes when we click on the LIST button in the dash board which is showed in the above. These lists of

nodes will have a pagination ruler because the number of nodes could be more that the screen space. There we can also find the search button which in the right corner and used to search specific node from the list.

4. Quantum Portal Implementation

This chapter presents the implementation details of the Quantum Portal, including the Web application and the database structure.

4.1. Database

In our project, we used MySQL Database which is a central component of the LAMP open-source software stack. LAMP is an acronym for “Linux, Apache, MySQL, Perl/PHP/Python”. LAMP is very easy to install, and setting up an instance of MySQL is a relatively simple task. In addition, it’s also an easy database to work with known languages. We used MySQL version “**ver 15.1 Distrib 10.1.16-MariaDB**”.

Couple of advantages of using MySQL;

1. It’s easy to use.
2. Support is readily available whenever necessary.
3. MySQL is inexpensive.
4. MySQL is an Open-Source.

4.1.1. Database Structure

Figure 14 - Structure of Users Table from the MySQL admin

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)		UNSIGNED	No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	firstName	varchar(191)			No	None		Change Drop Primary Unique Index More
3	lastName	varchar(191)			No	None		Change Drop Primary Unique Index More
4	telephone	varchar(191)			No	None		Change Drop Primary Unique Index More
5	email	varchar(191)			No	None		Change Drop Primary Unique Index More
6	password	varchar(191)			No	None		Change Drop Primary Unique Index More
7	remember_token	varchar(100)			Yes	NULL		Change Drop Primary Unique Index More
8	created_at	timestamp			Yes	NULL		Change Drop Primary Unique Index More
9	updated_at	timestamp			Yes	NULL		Change Drop Primary Unique Index More

In the above figure we can see the structure of our database with user table which has the fields:- Id, firstName, lastName, telephone, email, and password, remember_token, created_at and updated_at. By the way, remember_token field is used for remember me in the login page, which created_at and updated_at are used when row is created and updated. The database has just one table.

4.1.2. MySQL Server Configuration

Laravel supports MySQL Server out of the box; however, it needs to add the connection configuration for the database to the configuration file(config/database.php).

```
'mysql' => [
        'driver' => 'mysql',
        'host' => env('DB_HOST', 'localhost'),
        'port' => env('DB_PORT', '3306'),
```

```
'database' => env('DB_DATABASE', 'forge'),
'username' => env('DB_USERNAME', 'forge'),
'password' => env('DB_PASSWORD', ''),
'charset' => 'utf8',
'collation' => 'utf8_unicode_ci',
'prefix' => '',
'strict' => true,
'engine' => null,
    ],
```

Above we can see the database configuration for username, password, host and port in ".env" file, which stores database details. In the below code we can see that our project has a database called "devportal" and username as "root" no password set.

Source code from .env file

```
APP_ENV=local
APP_KEY=base64:/F8m2EsAx6PkaytkRyFCSLXl61SEerl+ZaFAx3nYX9U=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=devportal
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

```
PUSHER_APP_ID=
PUSHER_KEY=
PUSHER_SECRET=
```

Previous presented information is related with the used database, its structure and configuration. Next, it is presented the details of the Web application implementation.

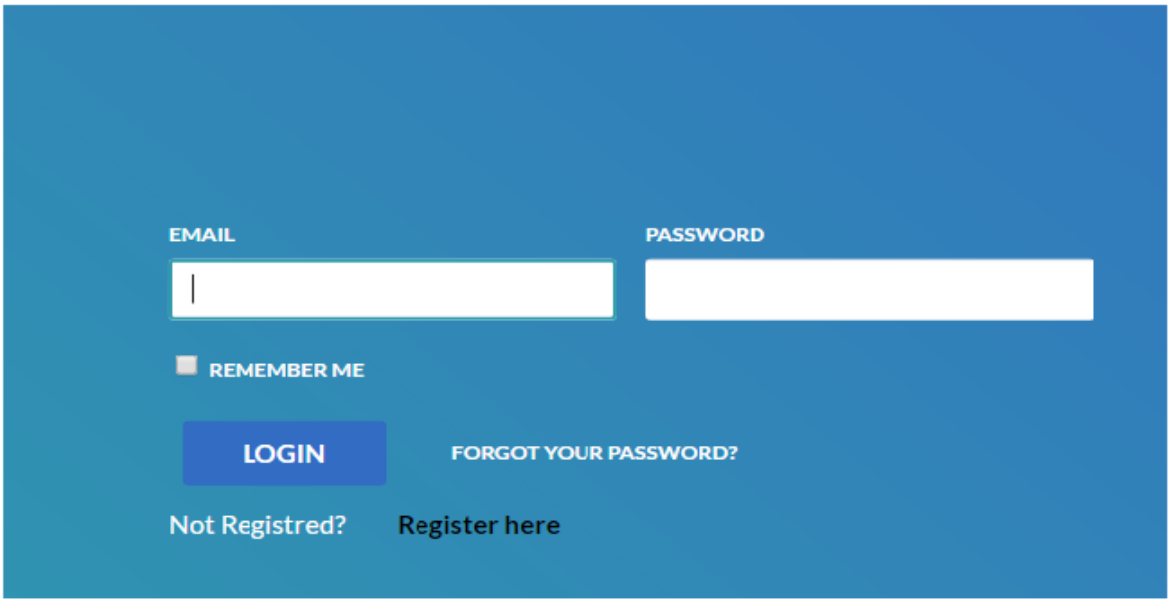
4.2. Quantum Portal Web application

Quantum Portal application has the main aim of handling information about the Quantum Node owned by the various customers. Information includes status(online/offline), location, view of the nodes, and management of the user accounts and nodes, storing the node's current information updated by clients, focusing mainly on the presentation of visual data in the form's tables.

4.2.1. Form Login

In general login is the procedure used to get access to an application. In Quantum Portal application login form is used for the user to login and view the nodes information, like - node status, list of nodes, billing information, monthly traffic, status whether the node is online or offline.

Figure 15 - Login Form



4.2.1.1. Format details

In the login form, there are two edit boxes, one checkbox, two links and one button. Two edit boxes are email and password, where user can enter his email address and password who is already registered before. Check box is to remember the email address and password in-to the web portal. The button called Login is used to grab email and password and send them to the server side in order to find out if the specified user can or cannot use the application. The two links are – one is for forgot password if the user forgot his/her password he/she can click and enter his/her email and, it will send a link for email to recreate the new password. The second one is for users who are not register yet. They can click on the link and register themselves.

4.2.1.2. Associated View

Login form is linked to the view **login.blade.php** which contains the HTML served by the application (views are stored in the resources/views directory). The complete HTML code for the login form can be seen in **login.blade.php**.

4.2.1.3. Associated Controller

This portal relies in just one controller called Main-Controller. Next it is presented the Controller source code to handle the form login functionality.

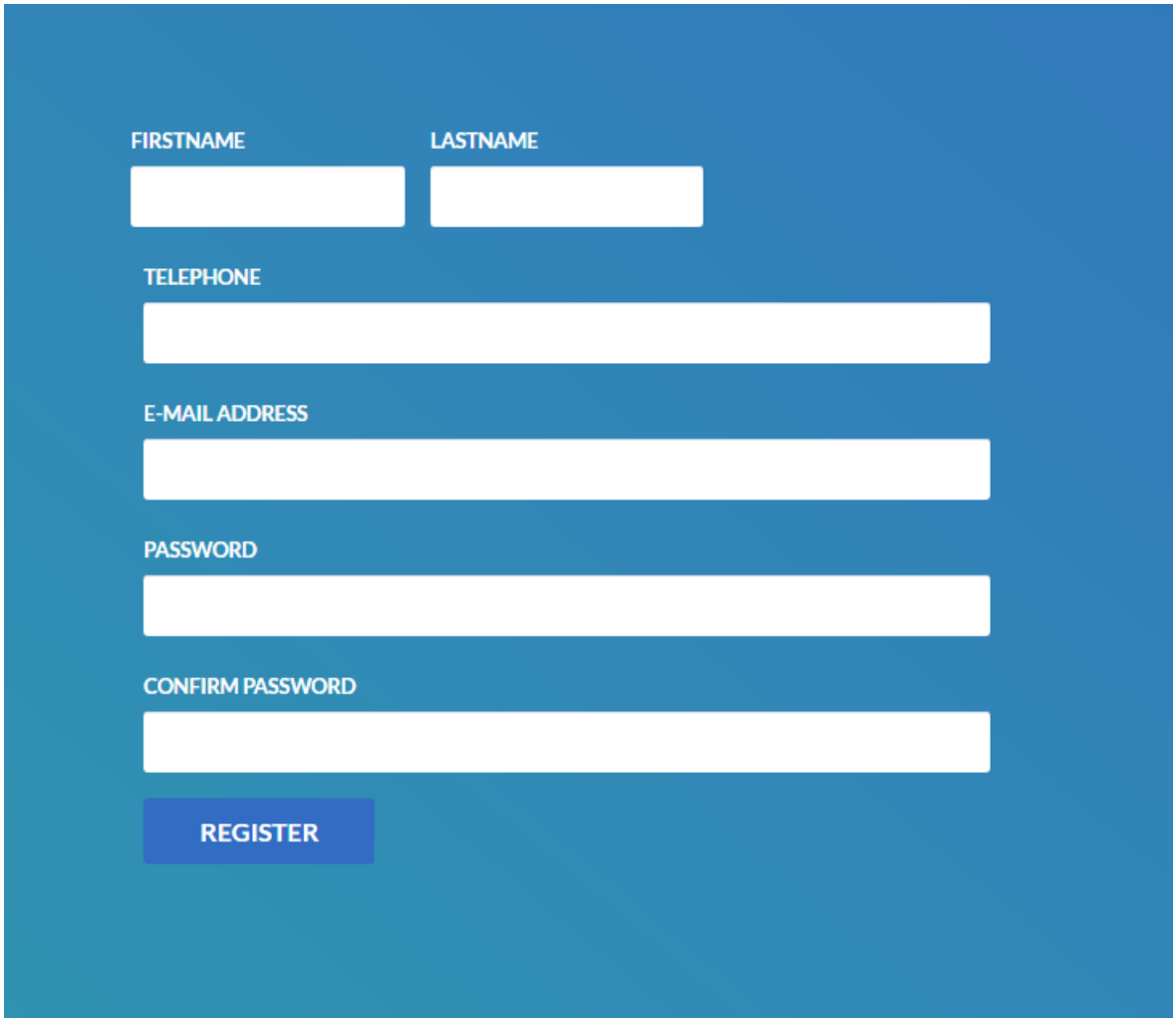
Source code for the form login

```
public function Login(Request $request)
  {
      /**
      * Function Login
      *
      * Code to login the user with the email and password
      *
      * @param $request -> string $mail (email of user)
      * @param $request -> string $password (password of user)
      * @return boolean (true or false)
      */
      $mail = $request->get('email');
      $pass = $request->get('password');
      if(Auth::attempt(['email' => $mail, 'password' => $pass]))
  {
      //return true
      //Need to pass the userId to the function vNodeInformations
      $dataDashboard =  $this->vNodeInformations();
      return view ('index')->with(['dataDashboard' => $dataDashboard]);
   }
    else
          {
      return redirect()->back()->withErrors(['Login failed! There is a problem with your email
address or password. Please try again.', '']);
        }
  }
```

4.2.2. Form for User Registration

This form is used to register user in to the portal. If user wants to participate in a Quantum portal application user needs to be registered before. Hence, register page has only personal information's like -FirstName, LastName, Telephone, Email address, Password, Confirm Password.

Figure 16 - Form Registration



4.2.2.1. Format details

Registration form contains six edit boxes for entering user personal information and one button. Once all user information is entered in the edit boxes, user will press the register button in order to send the information to the database, via server side.

4.2.2.2. Associated View

Register form is associated in the view **register.blade.php** which includes the HTML served by the application. The complete code for the register form is located in **register.blade.php** file.

4.2.2.3. Associated Controller

Upcoming is presented the Controller source code to grasp the form Register functionality.

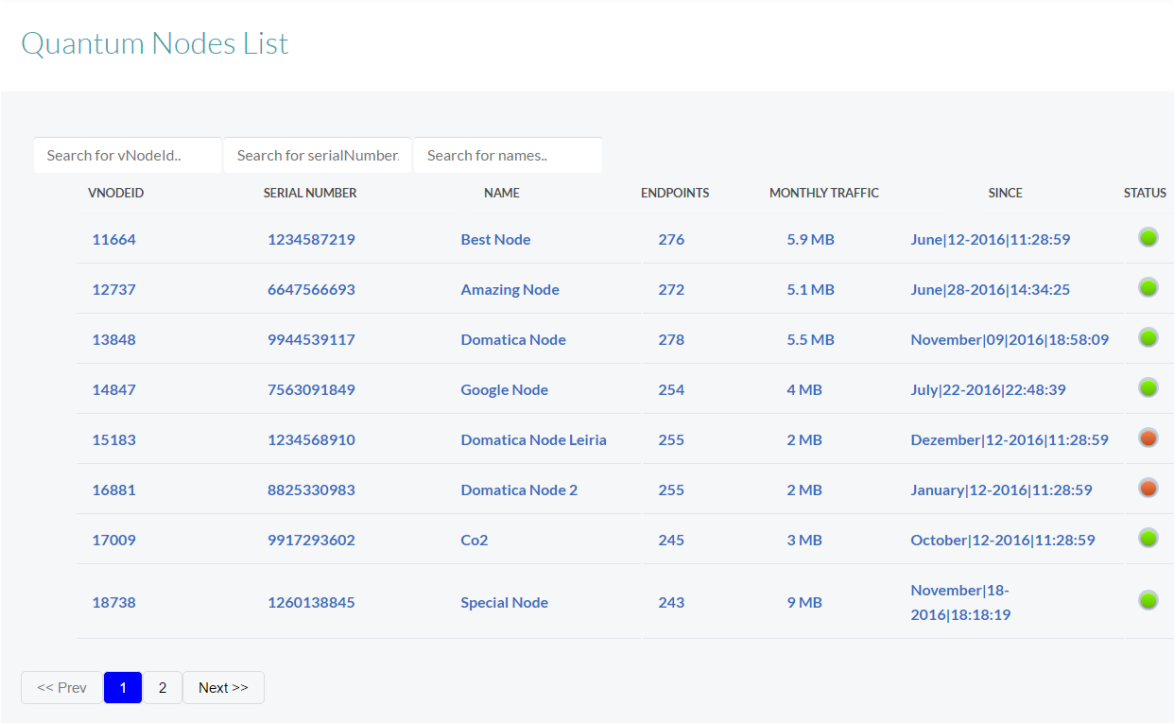
Source code for the Form Register

```
Public function Register(Request $request) {
        /**
         * Function Register
         *
         * Code to register the user with yours information and the entity information
         *
         * @param $request ->  All user and entity information and the authentication
method
         * @return boolean (true or false)
         */
        //CODE to get all informations (user and entity)
        $firstName = $request - > get('FirstName');
        $email = $request - > get('email');
        $password = $request - > get('password');
        $password_confirmation = $request - > get('password_confirmation');

        User::create([
           'FirstName' => $request - > get('FirstName'),
           'LastName' => $request - > get('LastName'),
           'Telephone' => $request - > get('Telephone'),
           'email' => $request - > get('email'),
           'password' => bcrypt($request - > get('password'))
        ]);

        return view('welcome');
      }
```

4.2.3. List Nodes

Here user can see the list of nodes and their details like NodeId, Serial Number, name, Endpoints, Monthly Traffic, Since and status. It also contains the pagination. Once there is no node's records yet (due to hardware unavailability), it was manually created 15 nodes records in the controller in order to test this view.

Figure 17 - List of nodes Form



The view is able to search data and to sort data using columns heading.

4.2.3.1. Format details

List nodes relies in a data table object in order to show nodes information following a tabular approach. The last column, the status column, is shown with a color: Red status means that node is offline, while green color means that node is online. Besides this, on the bottom, there is pagination object in order to split information for different pages in case of lots of nodes.

4.2.3.2. Associated View

List of nodes is located in the view folder with a name called **nodes_list.blade.php**.

4.2.3.3. Associated Controller

The List nodes background controller is the Main-Controller and the source code can be seen in the next listing,

Source code for the List Nodes option

```
public function vNodeInformations(/* $userId */)
{
/**
* Function vNodeInformations
*
* Code to get data about nodes
*
* @param  string $userId (user id to identify the user)
* @return array $nodesArray (array that contains all information about the nodes)
*/
// TEST ARRAY
$nodesArray = [
0 =>
[
'serialNumber' => '1234587219',
'vNodeId' => '11664',
'name' => 'Best Node',
'since' => 'September, 12-2016 11:28:59',
'client' => 'Cisco',
'status' => 'Online',
'geoLat' => '41.15', //Porto
'geoLong' => '-8.6166667',
'endpoints' => '276',
'monthlytraffic' => '5.9 MB'
],

1 =>
[
'serialNumber' => '6647566693',
'vNodeId' => '12737',
'name' => 'Amazing Node',
'since' => 'June, 28-2016 14:34:25',
'client' => 'EDP',
'status' => 'Online',
'geoLat' => '37.010327', //Faro
'geoLong' => '-7.937495',
'endpoints' => '272',
'monthlytraffic' => '5.1 MB'
],
2 =>
[
'serialNumber' => '9944539117',
'vNodeId' => '13848',
'name' => 'Domatica Node',
'since' => 'November, 09-2016 18:58:09',
```

```
'client' => 'Domatica',
'status' => 'Online',
'geoLat' => '39.766667', //Leiria
'geoLong' => '-8.883333',
'endpoints' => '278',
'monthlytraffic' => '5.5 MB'
],
3 =>
[
'serialNumber' => '7563091849',
'vNodeId' => '14847',
'name' => 'Google Node',
'since' => 'July, 22-2016 22:48:39',
'client' => 'Google',
'status' => 'Online',
'geoLat' => '40.204829', //Coimbra
'geoLong' => '-8.440357',
'endpoints' => '254',
'monthlytraffic' => '4 MB'
],
4 =>
[
'serialNumber' => '1234568910',
'vNodeId' => '15183',
'name' => 'Domatica Node Leiria',
'since' => 'Dezember, 12-2016 11:28:59',
'client' => 'Domatica-Leiria',
'status' => 'Offline',
'geoLat' => '38.722252', //Lisbon
'geoLong' => '-9.139337',
'endpoints' => '255',
'monthlytraffic' => '2 MB'
],
5 =>
[
'serialNumber' => '8825330983',
'vNodeId' => '16881',
'name' => 'Domatica Node 2',
'since' => 'January, 12-2016 11:28:59',
'client' => 'Domatica',
'status' => 'Offline',
'geoLat' => '40.416775', //Madrid
'geoLong' => '-3.703790',
'endpoints' => '255',
'monthlytraffic' => '2 MB'
],
```

```
6 =>
[
'serialNumber' => '9917293602',
'vNodeId' => '17009',
'name' => 'Co2',
'since' => 'October, 12-2016 11:28:59',
'client' => 'Domatica',
'status' => 'Online',
'geoLat' => '41.385064', //Barcelona
'geoLong' => '2.173403',
'endpoints' => '245',
'monthlytraffic' => '3 MB'
],

7 =>
[
'serialNumber' => '1260138845',
'vNodeId' => '18738',
'name' => 'Special Node',
'since' => 'November, 18-2016 18:18:19',
'client' => 'EDP',
'status' => 'Online',
'geoLat' => '48.856614', //Paris
'geoLong' => '2.352222',
'endpoints' => '243',
'monthlytraffic' => '9 MB'
],

8 =>
[
'serialNumber' => '3488902531',
'vNodeId' => '19198',
'name' => 'Quantum Node',
'since' => 'February, 28-2016 05:58:19',
'client' => 'Domatica',
'status' => 'Online',
'geoLat' => '51.507351', //London
'geoLong' => '-0.127758',
'endpoints' => '255',
'monthlytraffic' => '2 MB'
],

9 =>
[
'serialNumber' => '7719374021',
'vNodeId' => '20832',
'name' => 'Quantum Node 2',
```

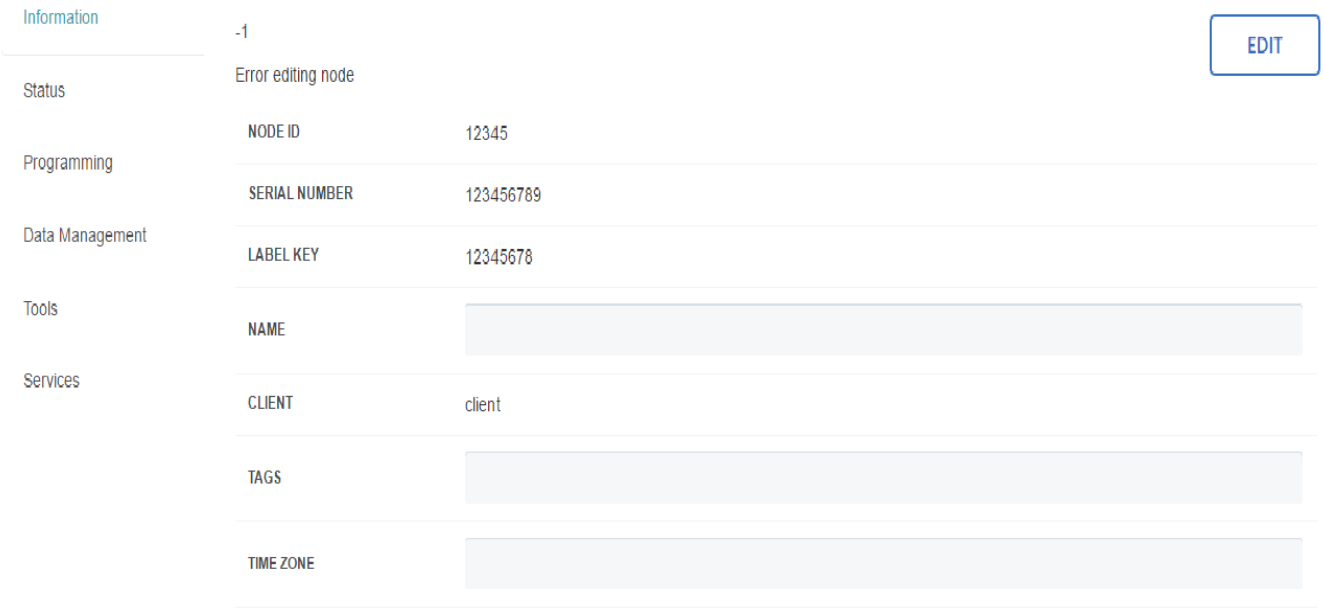
```
'since' => 'January, 02-2017 08:28:21',
'client' => 'Domatica',
'status' => 'Online',
'geoLat' => '52.520007', //Berlim
'geoLong' => '13.404954',
'endpoints' => '235',
'monthlytraffic' => '2.2 MB'
],

10 =>
[
'serialNumber' => '1348572384',
'vNodeId' => '21746',
'name' => 'Farm Control One',
'since' => 'August, 01-2016 15:48:49',
'client' => 'Farm Control',
'status' => 'Online',
'geoLat' => '38.570712', //Évora
'geoLong' => '-7.909307',
'endpoints' => '215',
'monthlytraffic' => '2.9 MB'
],

11 =>
[
'serialNumber' => '12348827332',
'vNodeId' => '22999',
'name' => 'Farm Control Two',
'since' => 'August, 18-2016 06:55:39',
'client' => 'Farm Control',
'status' => 'Offline',
'geoLat' => '41.902783', //Roma
'geoLong' => '12.496366',
'endpoints' => '243',
'monthlytraffic' => '32.5 MB'
],

12 =>
[
'serialNumber' => '12348827392',
'vNodeId' => '22999',
'name' => 'Farm Control Two',
'since' => 'August, 18-2016 06:55:39',
'client' => 'Farm Control',
'status' => 'Offline',
'geoLat' => '41.902783', //Roma
'geoLong' => '12.496366',
```

```
'endpoints' => '240',
'monthlytraffic' => '72.5 MB'
],

13 =>
[
'serialNumber' => '123488273382',
'vNodeId' => '22999',
'name' => 'Farm Control Two',
'since' => 'August, 18-2016 06:55:39',
'client' => 'Farm Control',
'status' => 'Offline',
'geoLat' => '41.902783', //Roma
'geoLong' => '12.496366',
'endpoints' => '240',
'monthlytraffic' => '12.5 MB'
],

14 =>
[
'serialNumber' => '12348827372',
'vNodeId' => '22999',
'name' => 'Farm Control Two',
'since' => 'August, 18-2016 06:55:39',
'client' => 'Farm Control',
'status' => 'Offline',
'geoLat' => '41.902783', //Roma
'geoLong' => '12.496366',
'endpoints' => '240',
'monthlytraffic' => '32.5 MB'
],
15 =>
[
'serialNumber' => '12348827322',
'vNodeId' => '22999',
'name' => 'Farm Control Two',
'since' => 'August, 18-2016 06:55:39',
'client' => 'Farm Control',
'status' => 'Offline',
'geoLat' => '41.902783', //Roma
'geoLong' => '12.496366',
'endpoints' => '240',
'monthlytraffic' => '86.5 MB'
],
];
return $nodesArray;
}
```

4.2.4. Nodes Information Form

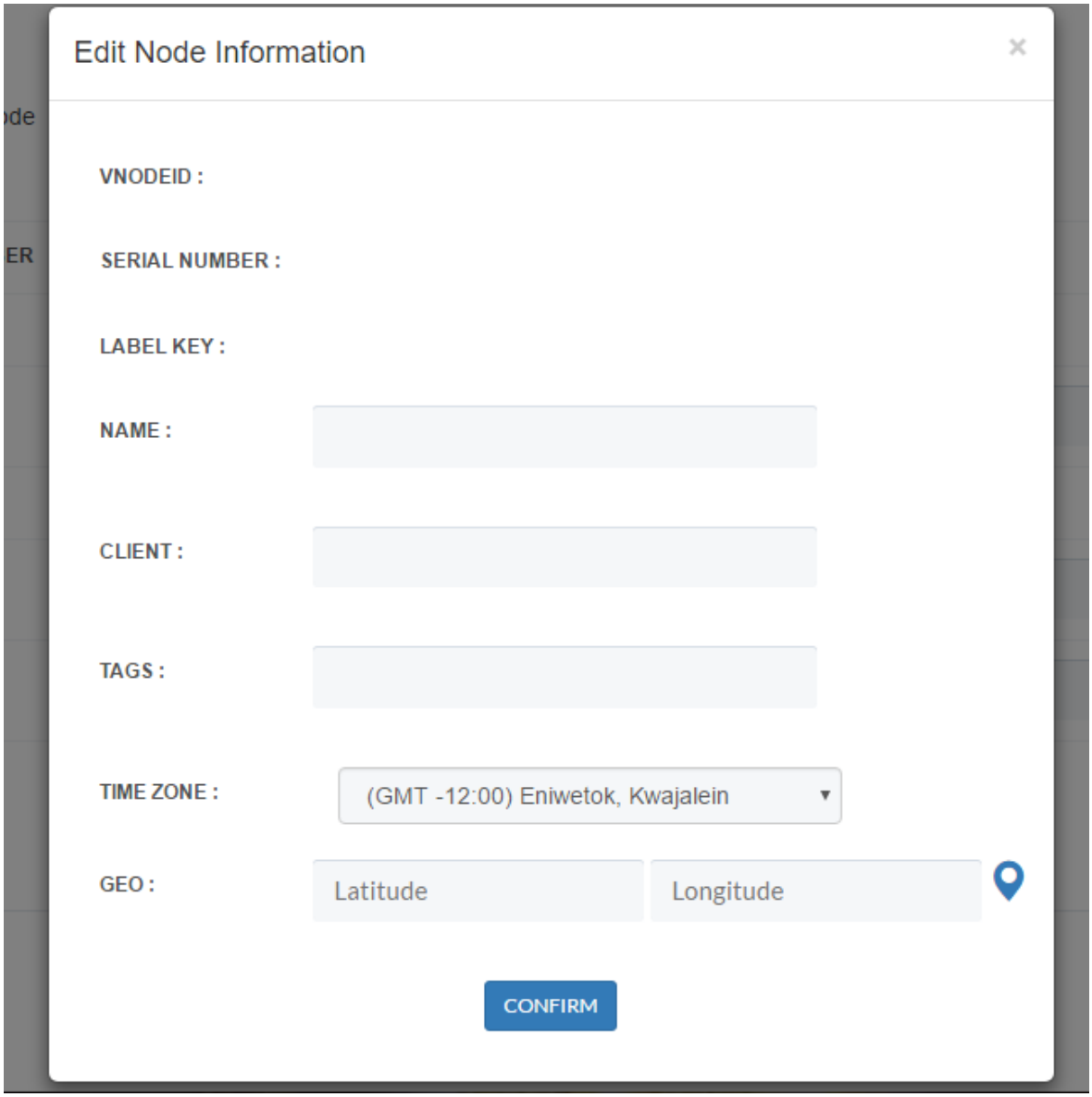
This form presents information about nodes such as NodeID, Serial Number, Label key, Name, Client and services and Tools Data management for the users. Besides information presentation, it is also possible to update information.

Figure 18 - Nodes Information Form



Above we can see the edit button useful for updating information,

Figure 19 - Edit Node Information



4.2.4.1. Format details

In nodes information form we can see three edit boxes and one assemble box. Edit boxes are Name of the user, Tags and Time zone for the particular time information about the node, where user can enter name, tags and Time zone. One is assemble mentioned as EDIT. If user click on EDIT button figure 6 form will appear. This form has again Name, Tags, Time zone and geo (Latitude and longitude). After entering all details user should press confirm button to see the changes.

4.2.4.2. Associated View

Nodes Information form is located in the view folder with a name **nodes_details.php** and complete code for the Nodes information form is located in the same file.

4.2.4.3. Associated Controller

Next is presented the source code for the nodes information form from the **"Main Controller"** to handle the nodes information's functionalities.

Source code for the Nodes Information

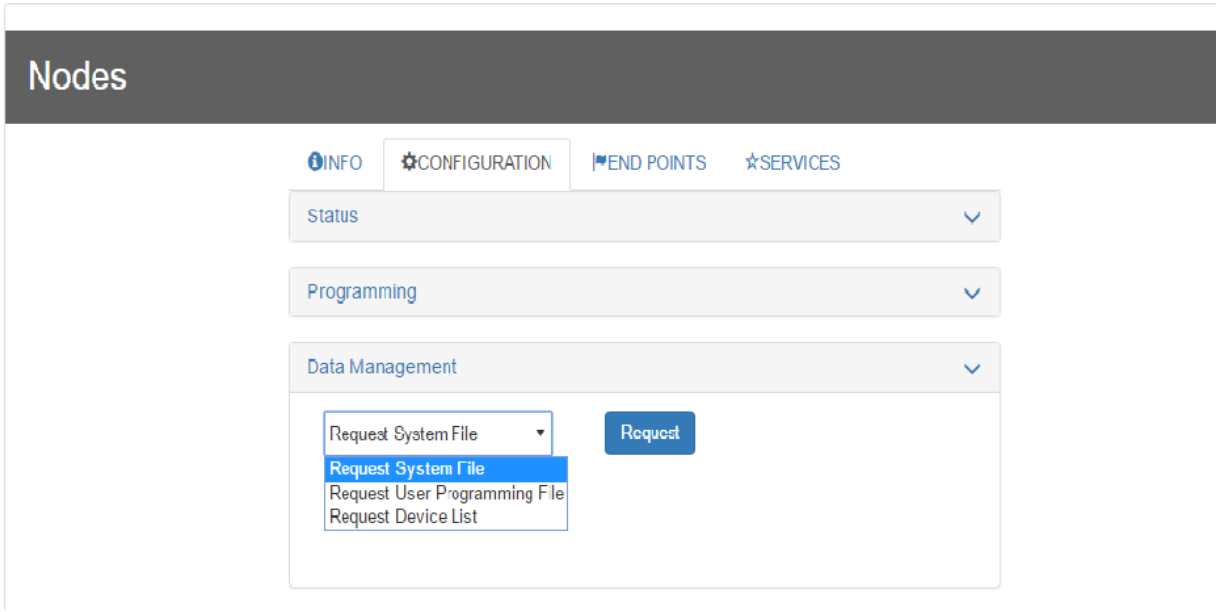
```
Public function GetNodeInfo( /* $userId */ ) {
   /**
    * Function GetNodeInfo
    *
    *
    * @param string $userId (user id to identify the user)
    * @return (type) (name)
    */
   //CODE (not made yet) to send all informations with rest and save on remote
database

   /*
   $url =  "url to register";
   $body = "array with all content";

   $this->ConnectionRest($url,$body);

   */

}
Public function showEditNode() {

   /**
    * Function showEditNode
    *
    * Code used to show the view to edit nnodes
    *
    * User type -> Owner + Admin
    * @param
    * @return (type) (name)
    */
   return view('pages.EditNode');
}
```

```
Public function WriteNodeInfo(Request $request) {
   /**
    * Function WriteNodeInfo
    *
    *
    * @param $request -> $name (new name of node)
    * @param $request -> $tags (new tags associated to node)
    * @param $request -> $timeZone (new node time zone)
    * @param $request -> $geoLat (new latitude of node)
    * @param $request -> $geoLong (new longitude of node)
    * @return (type) (name)
    */
   $name = $request - > get('newNodeName');
   $client = $request - > get('newClientName');
   $timeZone = $request - > get('timezone');
   $geoLat = $request - > get('geoLat');
   $geoLong = $request - > get('geoLong');
   //CODE (not made yet) to send all informations with rest and save on remote
database
   /*
   $url =  "url to register";
   $body = "array with all content";
   $this->ConnectionRest($url,$body);
   */
   return redirect() - > route('ListNodes');
}
Public function showConfigureNode($vNodeId = '') {
   /**
    * Function showConfigureNode
    *
    * Code used to show configure node pages
    *
    * @param (type) (name) about this param
    * @return configure node view with the vNodeId from node selected by user
    */
   $data = new Controller();
   $nodeStatus = $data - > nodeStatus();
   $nodesServices = $data - > nodesServices();
   return view('pages.nodes') - > with(['vNodeId' => $vNodeId, 'nodeStatus' =>
$nodeStatus, 'nodesServices' => $nodesServices]);
}
```

4.2.5. Node Configuration Form

In this form user can request for files like Request System file, Request User Program File, and Request Device List. Here because of some internal enterprise issues we are not able to request all of three files. The only that succeeds is the Request System File.

Figure 20 - Node Configuration Form



4.2.5.1. Format Details

In configuration form we can see drop down box with naming three types of file Request System file, Request User Programming File and Request Device List, and request button. After choosing file from the drop box user can request the file by clicking on request button.

4.2.5.2. Associated View

Node configuration form is located in to the **views/pages.** This form is created at the time of starting so in that application we have one more folder

called pages because, in new design we don't have that folder_(pages). The complete code is located in the file called "**nodes_details.blade.php"**.

4.2.5.3. Associated Controller

This Web application form also relies in the controller **"Main Controller"** which handles the configuration and actions.

Source Code for the Node Configuration

```
Public function GetSystemFile( /* Request $request */)
        {
        /**
         * Function GetSystemFile
         *
         * Code used users see the system file associated to your nodes
         *
         * @param $request ->
         * @return (type) (name)
         */
        $service_url =
"https://dcs.domaticasolutions.com/api/v1/operation/monitorControl/requestSyste
mFile";

        // request some information according to the specific gateway and token

        $curl_post_data = array (
                "vNodeId" => "2775777578137622031",

                // "gToken" => "39992D9EF2E8928039BA0BAF8D02A8A0",

                "parameters" => ""

                // to system file it is not necessary to specific a parameter

        );

        // encode the data (json format)

        $data_string = json_encode($curl_post_data);

        // start curl

        $curl = curl_init($service_url);
```

```
        // some curl configuration

        curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false); //this line of code solve
the certificate error. Read more about ssl certificate
        curl_setopt($curl, CURLOPT_HEADER, false);
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

        // post method

        curl_setopt($curl, CURLOPT_POST, true);

        // send the data string (gateway and token on post method)

        curl_setopt($curl, CURLOPT_POSTFIELDS, $data_string);

        // execute the request to the domatica

        $curl_response = curl_exec($curl);

        // echo $curl_response; //to show all json object
        // some error verifications

        if (curl_exec($curl) === false)
                {

                // in case of failure, show the specific error message in browser

                echo 'Curl error: ' . curl_error($curl);
                }
         else
                {

                // in case of sucess (put a message if you want)

                echo ' ';
                }

        // close the conection

        curl_close($curl);

        // [1]First step - cut the message from json object
        // decode the data

        $json = json_decode($curl_response, true);

        // get only the message to be decompress
```

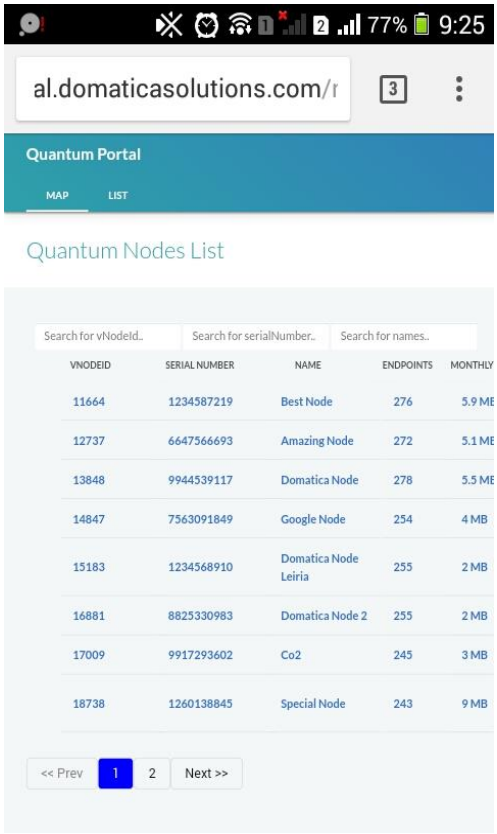
```
        $json_cut = $json['resultMessage'];

        // echo $json_cut; // to show the cut message
        // [2]Second step - decompress and show the cut message in final format
(xml)

        return $json_cut;
        }

Public function showPagesNodes()
        {
        return view('node_details');
        }

Public function getRequestDataManagement(Request $request)
        {
        $data = $request->get('selectData');
        if ($data == 'SystemFile')
                {
                $json_cut = $this->GetSystemFile();
                }
         else
        if ($data == 'ProgramFile')
                {
                }
         else
        if ($data == 'DeviceList')
                {
                }

        return response()->json($json_cut);
}
```

5. Test and Evaluation

To conduct extensive testing, there is a need for a number of different environments. The tester should configure a representative sample of all possible operating systems, browser, and different security settings.

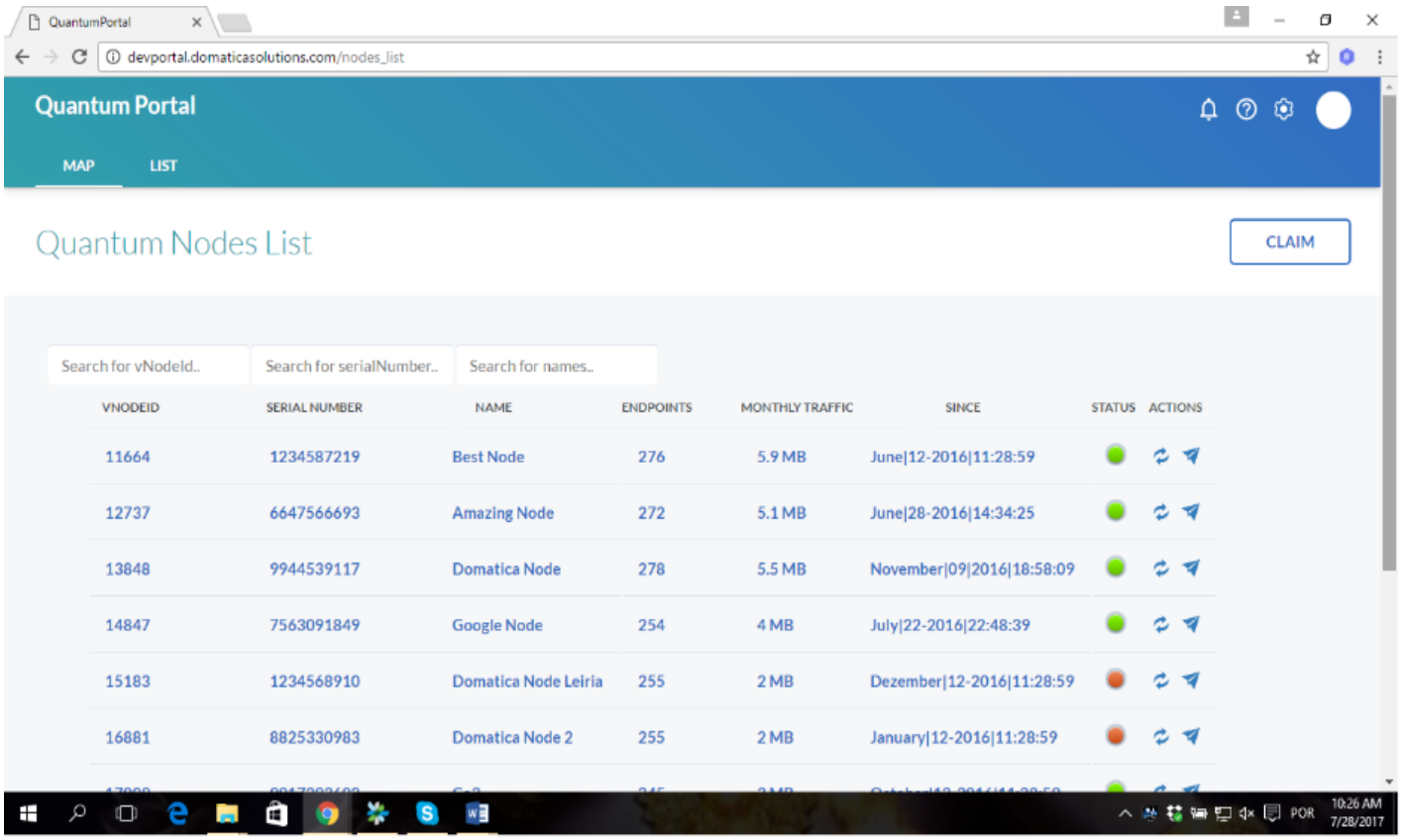
There is an emerging need to test the web page in different browsers to check if the content is displayed properly. The webpage should be optimally displayed on different screen sizes as it is difficult to predict what devices the visitors will use. Responsive web design approach can be used to create a universal web application for various devices, including smartphones and tablets. Here are some screens for various devices below.

Figure 21 - Node List in Mobile Phone



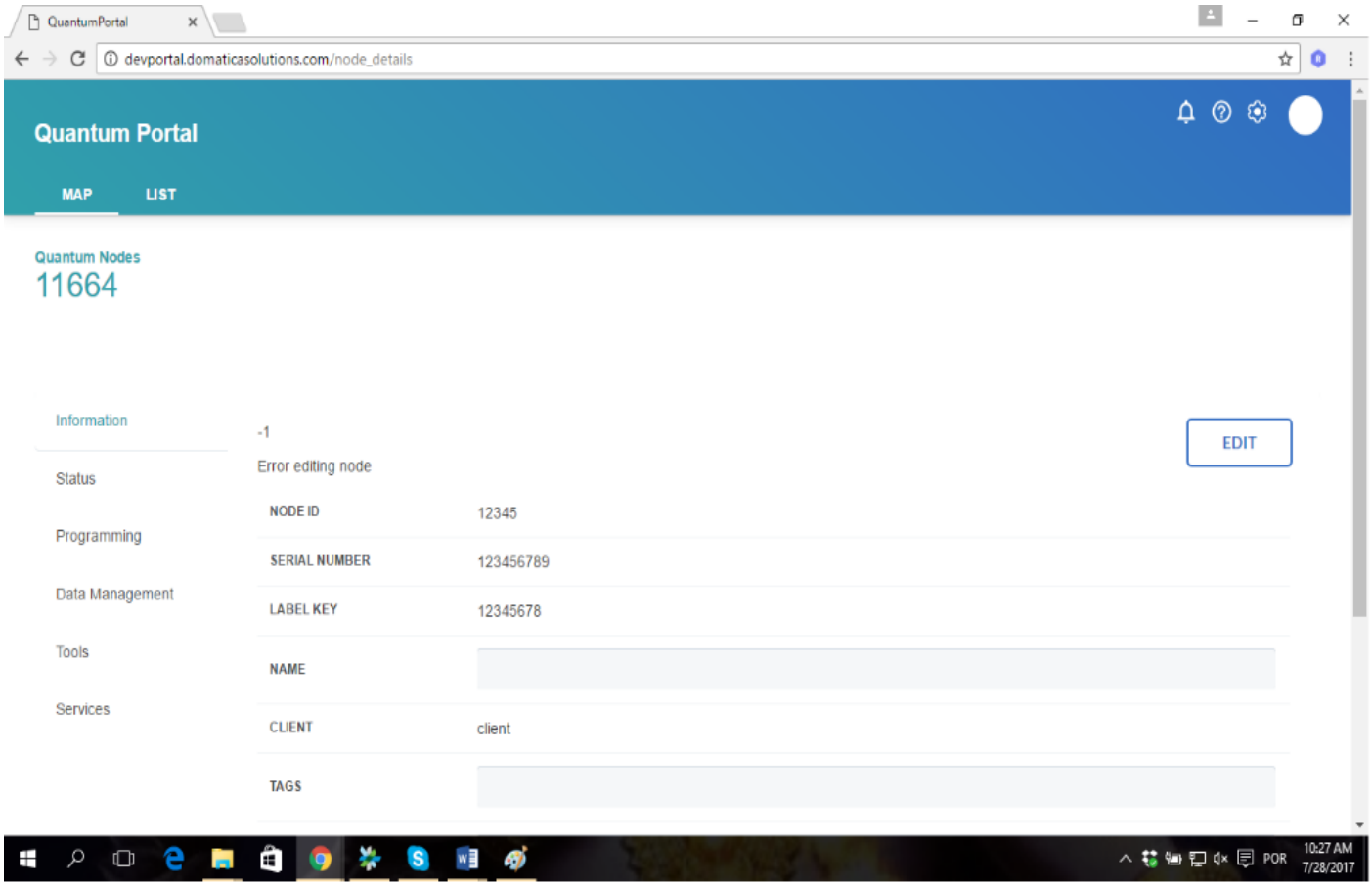
The above figure 21 shows that web application is not completely responsive and uses must zoom in and zoom out facing some difficulties. The used device was a phone Sony Xperia M model.

Figure 22 - Node list in small screen Laptop



The above image shows that web application looks great for the small system lap tops. Next will be tested for big screen.

Figure 23 - Node information with big screen



As can be seen in the above image, the web application does not look great on big screen devices. I recently started to work on it so needs to be done in future.

Email Verification

If the user register in the register page he must get a notification that send him a link to the user's mail id which he/her used for the registration.

Figure 24 - Registration page for verification

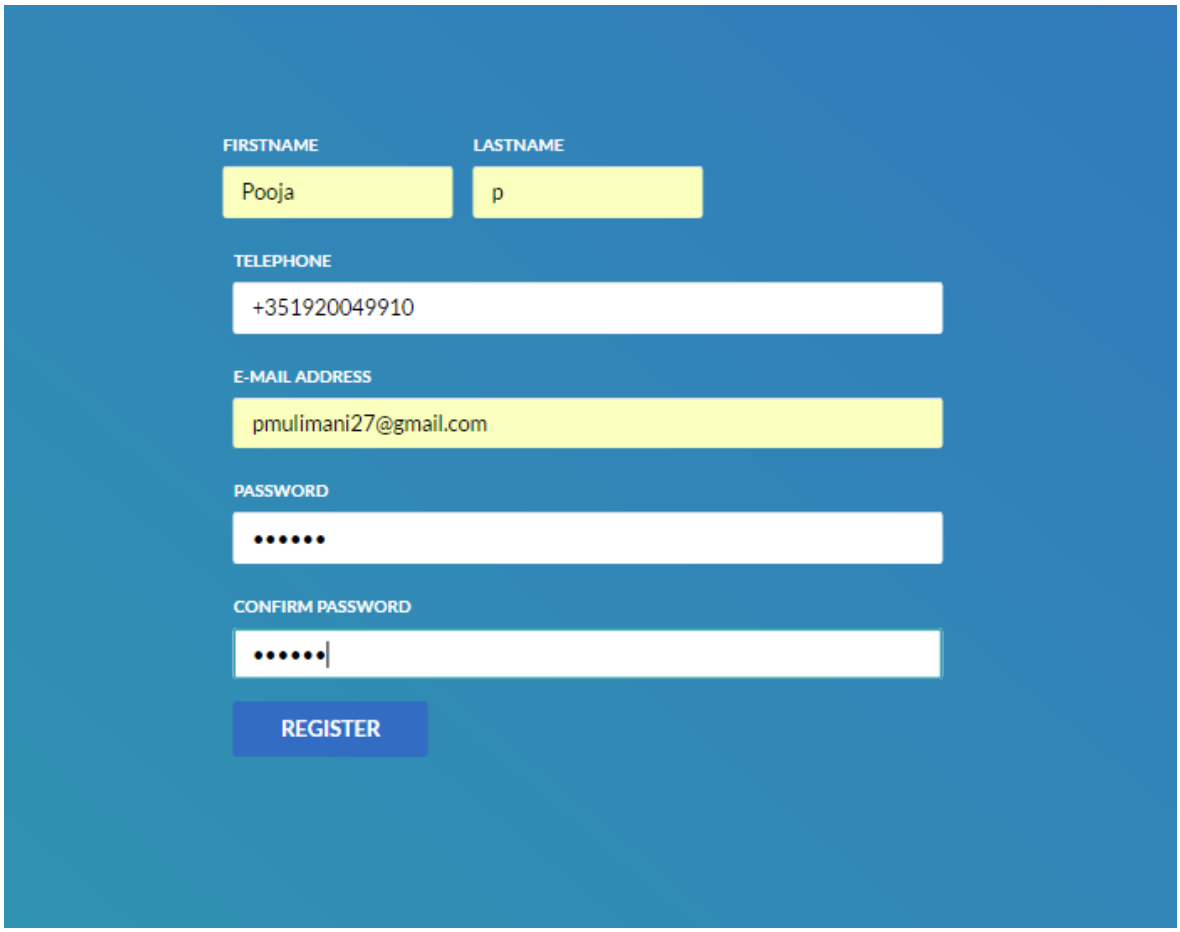
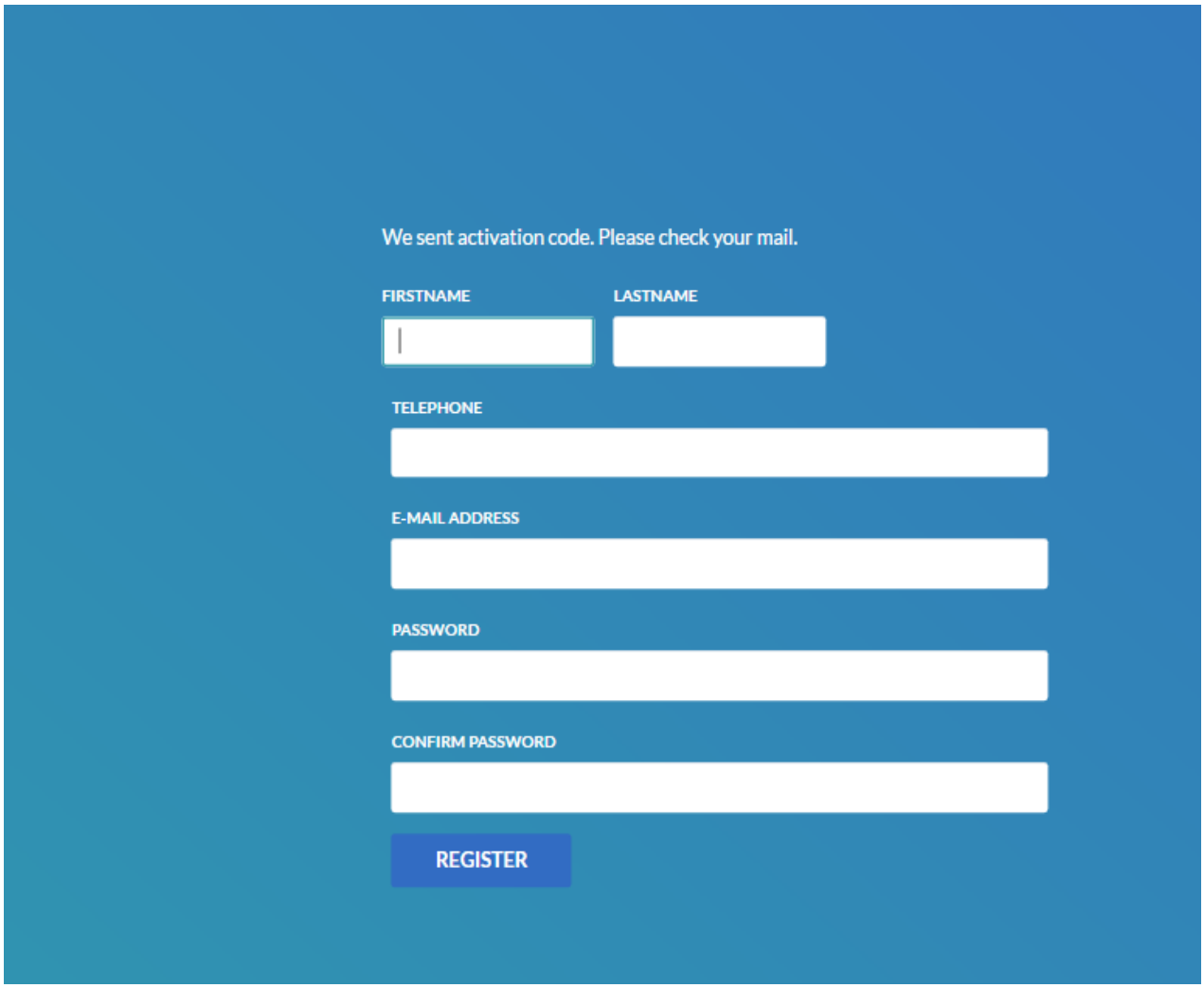
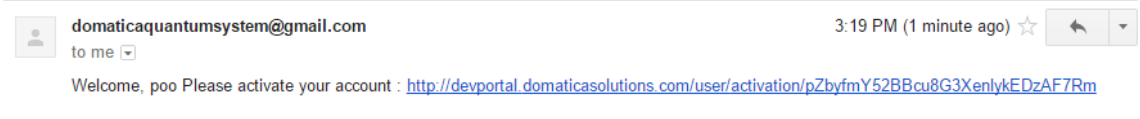


Figure 25 - Getting activation code or link



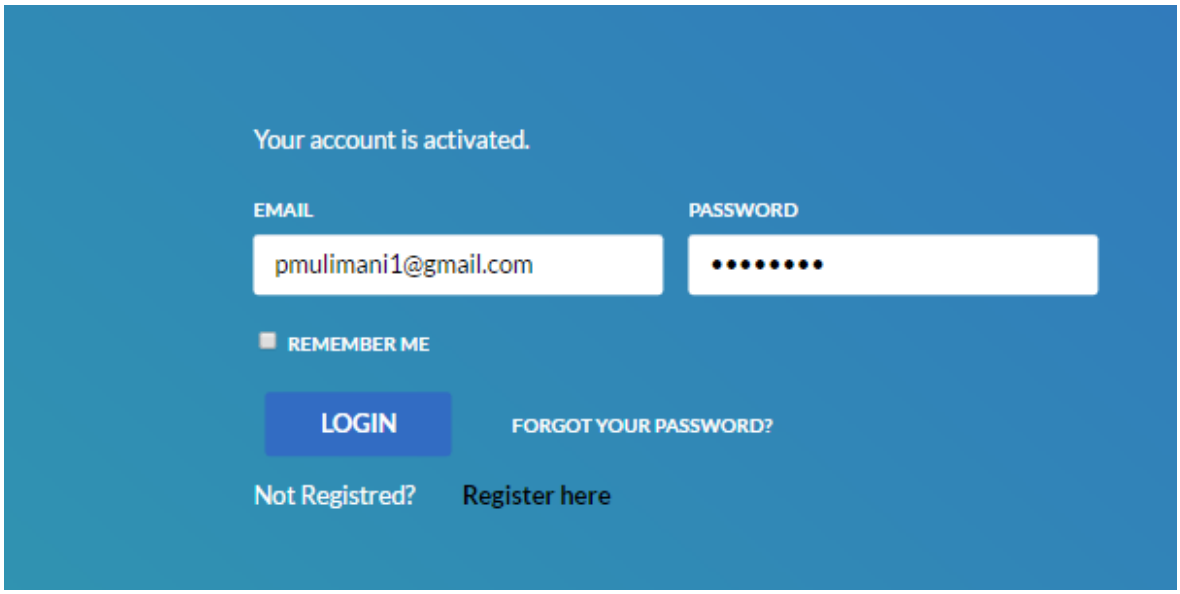
When user enters to his email account he will be getting the link or code as shown below.

Figure 26 - User account after registration



When user clicks on the link he will go to the Quantum Portal, and is ready to issue the first login operation.

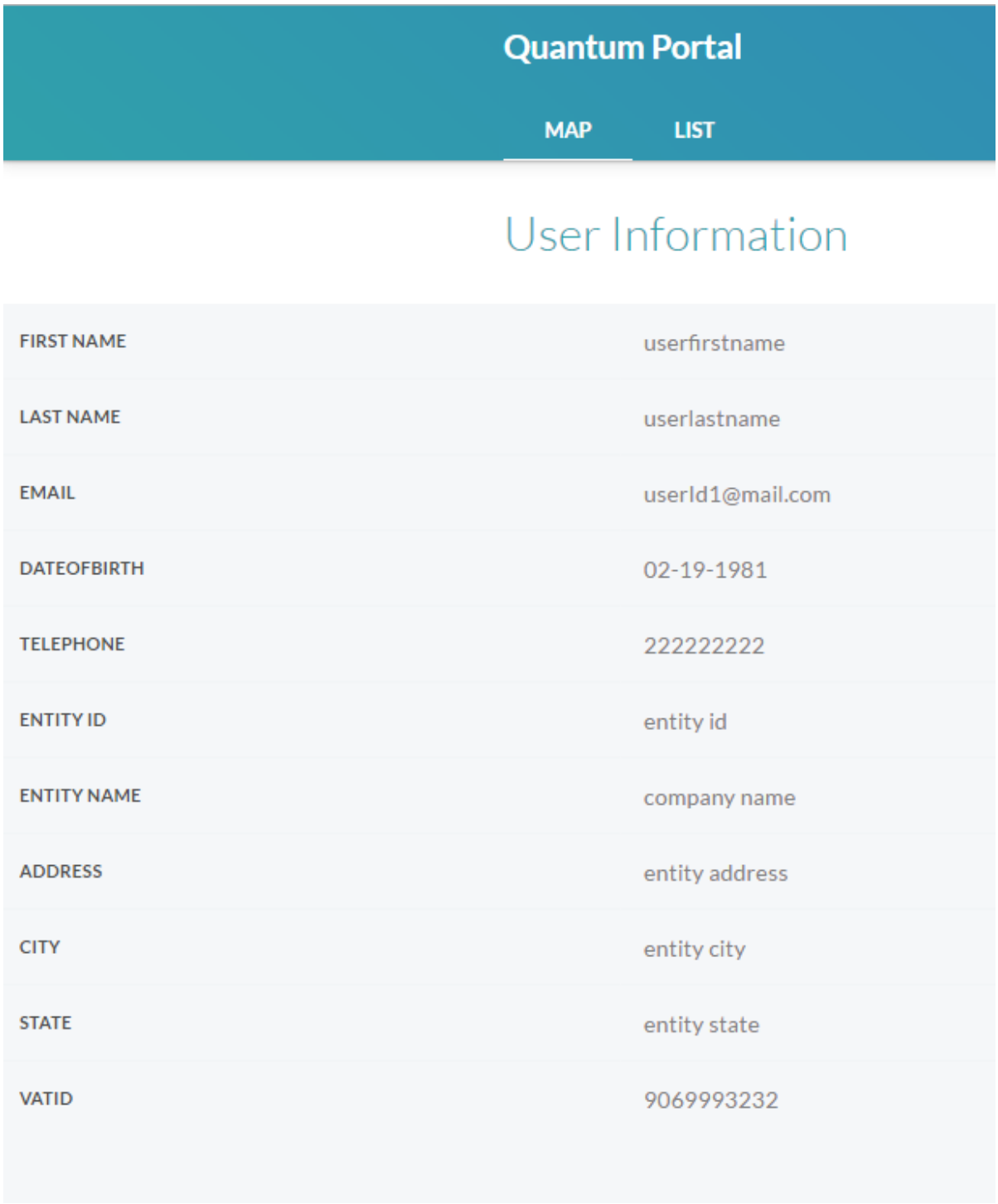
Figure 27 - login to the main page



User Information

When client/admin need to check user information like first name, last name, email etc. he/she need to view the user information page.

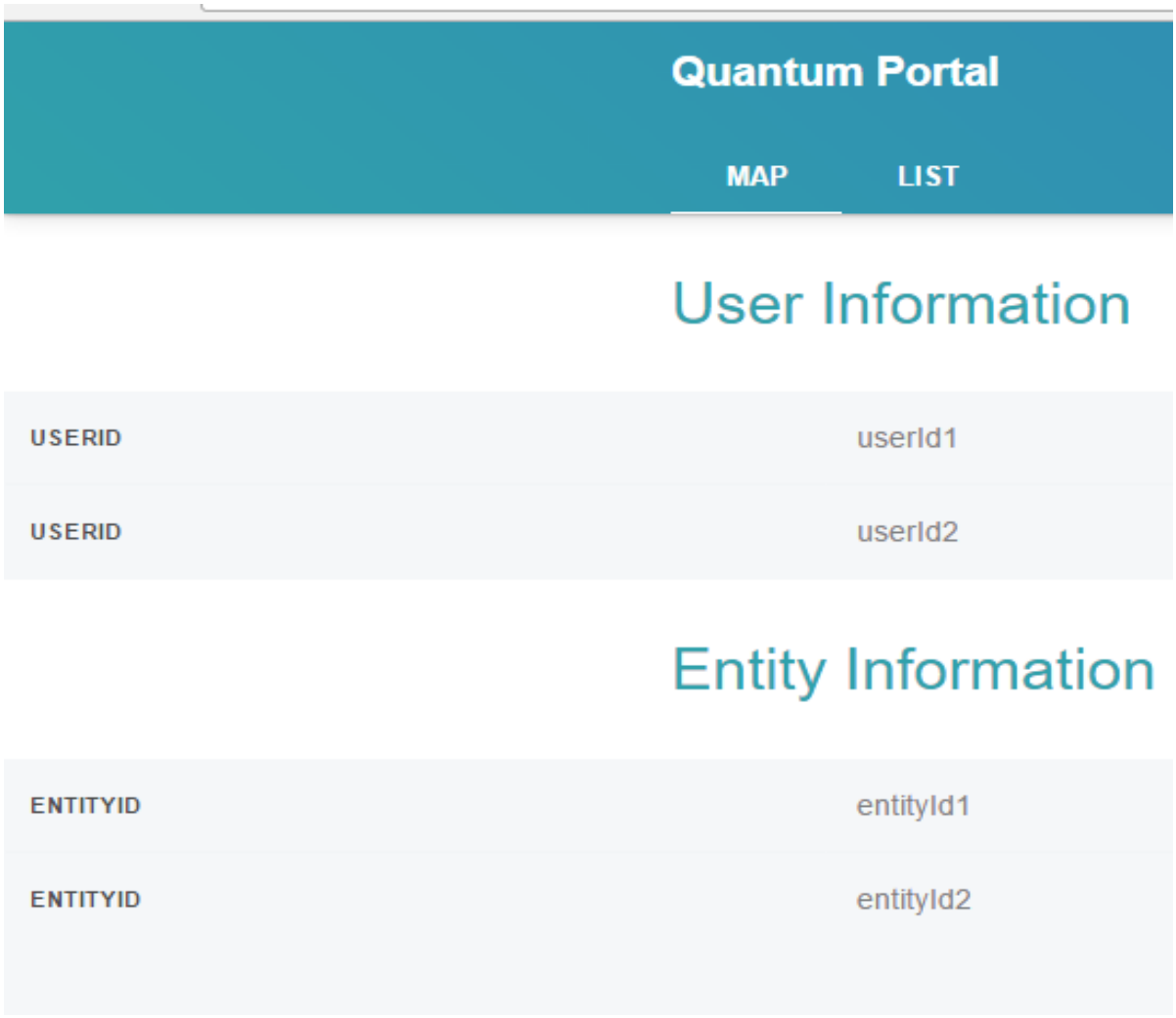
Figure 28 - User information viewed by admin/client



Delete User

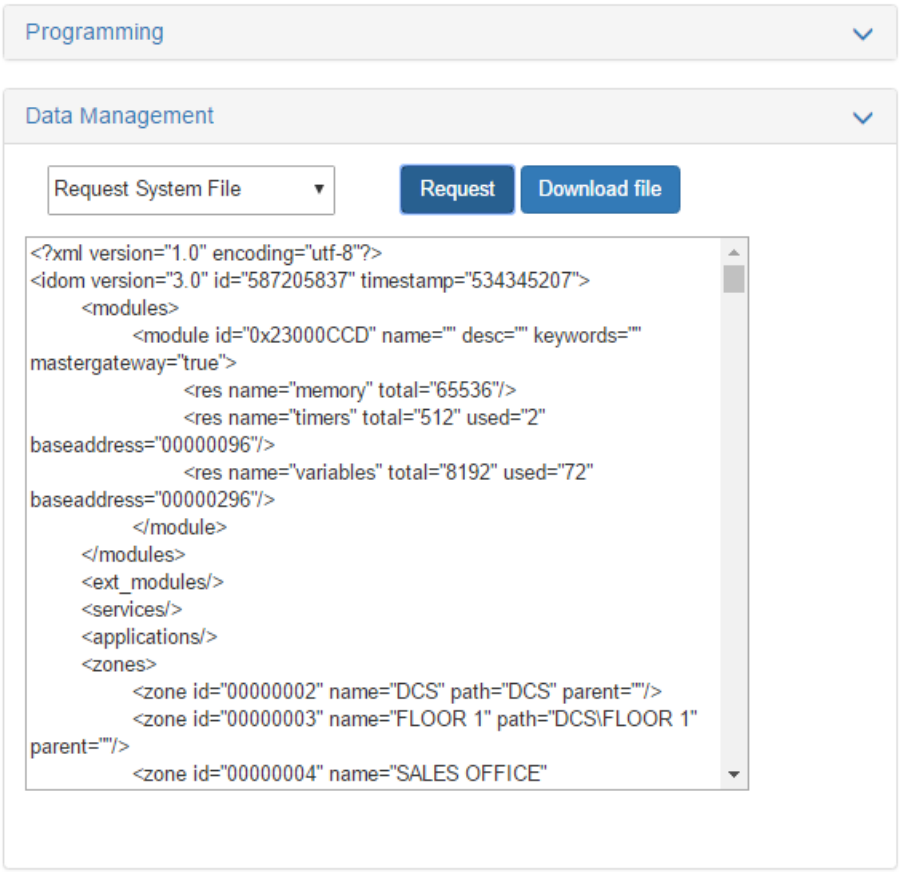
When admin/client wants to delete user information or user from the Quantum portal (or node) he needs to view the information like user information or entity information which is stored in the DeletUI form.

Figure 29 - DeleteUI form



System File.

Figure 30 - Request System File



When user requests a system file, it receives a file that contains memory total, timer total and time stamp for the node, as well model id.

Evaluation is that our application is not yet ready to work, and some file like Request System file, User Program files will work only in the Domatica network.

Recently, they have given some URL to get System File and logfile from the cloud. In future I need to work on that URL to get information like, User Information, List nodes, Delete UI and Delete user information. These kinds of work have to be done in the upcoming days. This happens because REST API team hasn't the API finished yet at the time of this project report delivery.

6. Conclusion and Future Work

Taking stock of the work during the 9-months internship, I believe that the project is practically in the idealized state at the beginning of its development. At the time of starting I was not knowing about technologies as my back ground is Electrical and Electronics. Internship gave me a big flat form to learn something better which will be helpful for my future.

Taking into account the planning of the project carried out in the initial phase, I considered that the proposed objectives have been fulfilled, except for "web project availability and the responsive". The reason this objective was not full filled is due to the fact that the functionalities to be implemented have not been explicitly described. As we planned all back-end functionalities could be in the cloud before the time we expect. It will be good to work for front end people to test their functionalities which have been working in the local host, but the cloud functionalities took more time from the backend work, and also to get logfile and system files an public URL is needed which is given by the Higher authorities but unfortunately Domatica could not provide any URL to get file from cloud. But local host work and gets files work perfectly. Since planning only contemplated generic tasks (such as back-end and frontend implementation), the range of possibilities for the various functionalities to be implemented was left open. This has led to the development of functionalities without having defined at which point the project should be made available to the web.

In relation to the encountered difficulties in the development of the project, I consider that the greatest challenges were due to the lack of specific knowledge, mainly in terms of front-end languages as well as the level of Quantum Node Portal concepts. That lead to the design changing every week and making same thing with different style sheets. There were also some difficulties in the initial planning of the project, since it was not possible to

carry out a survey of requirements as specific as the idealized one (since the tasks that existed at the beginning of the stage were quite generic and were being specified as required by Domatica).

The plans for the future of the project are to continue to develop new features and improve existing ones, taking into account the requirements of Domatica and its customers, as well as their availability of the public URL and functionalities on the web when it is considered appropriate.

7. References

[1] Samisa Abeysingha , “Php Team Development”, packt publishing, October 2001

[2] Mohammed Musthafa, “Analysis of Model based MVC framework for php development”, Academics Education conference,2008

[3] Peter Sevcik and John Bartlett, “Understanding Web Performance”, Business Communications Review, October 2001

[4] Martin Artlitt, “A performance Comparison of Dynamic Web Technologies”, ACM sigmetrics,December 2003

[5] Nguyen Minh Thanh, “Adopting Web Framework in Web Application Development”, Spring 2012

[6] David Díaz Clavijo, “A practical comparison of Agile Web Frameworks”, 2014 [21-45].

[7] M. Alsadat Kazemi and S. Eskandari, Web-based application for Collaborative Ethical Decision Making” Dep. of Information Technology, Uppsala Universitet, 2013.

[8] The Open Group Base Specifications Issue 6, IEEE Std 1003.1. ”clock - report CPU time used” opengroup.org http://pubs.opengroup.org/onlinepubs/000095399/functions/clock html

[9] C. Wang, "Web Services Integration Method" Dep. of Computer Science and Engineering, Chalmers University of Technology, 2010.

[10] PHP Documentation. “PHP’s memory_get_usage function.” http://php.net/manual/en/function.memory-getusage.php

[11] Xdebug documentation. “Xdebug’s stack trace feature” xdebug.org. http://xdebug.org/docs/stack_trace

[12] PHP documentation. “History of PHP” php.net. http://php.net/manual/en/history.php.php

[13] D. Powers, “PHP Solutions: Dynamic Web Design Made Easy”. Apress, 2006.

[14] W3techs. “W3Techs - World Wide Web Technology Surveys”, w3techs.com.

[15] Packagist. “PHP’s package manager” Packagist.org.

[16] Su Chengjun, He Pilian. Applied Research on Developing Web Database with PHP [J]. Computer engineering, 2000.9:184-185.

[17] Sarker and K. Apu, "MVC Architecture Driven Design and Implementation of Java Framework for Developing Desktop Application," International Journal of Hybrid Information Technology, pp. Vol.7, No.5, 317-322, 2014.

[18] Chhikara, "A Web Architectural Study of HTML5 with MVC Framework," International Journal oj Advanced Research in Computer Science and Software engineering, pp. 451 - 454, 2013.

[19] F. K. Frantz, "A Taxonomy of Model Abstraction Techniques," in Simulation Conference Proceedings, 1995. Winter, Arlington, 1995.

[20] V. Tran, 1. Vanderdonckt, M. Kolp and S. Faulkner, "Generating User Interface from Task, User and Domain Models," in Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2009. CENTRIC '09. Second International Conference on , Porto, 2009.

[21] U. Zdun, "Dynamically Generating Web Application Fragments from Page Templates," in SAC '02 Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, 2002.

[22] A. Loh and M. Robey, "Generating Web Applications from Use Case Scenarios," in ASWEC '04 Proceedings oj the 2004 Australian Software Engineering Conference, Australia, 2002.

[23] H. M. Hao and A. Jaafar, "Tracing User Interface Design Pre-requirement to Generate Interface Design Specification," in Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference on (Volume: 01), Selangor, 2009.

[24] A. Ginige and S. Murugesan, "Web engineering: An Introduction," IEEE MultiMedia (Volume:8 , Issue: 1), pp. 14 - 18, 200l.

[25] Daintith, IT: A Dictionary of Physics, Oxford, UK: Oxford University Press, 2009.

[26] Bll.Ijah, Laravel: A free book covering the Laravel 4 Oflicial Documentation, Vancouver, BC, Canada: Leanpub, 2014, [Online]. Available: http://leanpub.com/14. Accessed: Apr 21, 2014.

[27] M.F. Awaj, "Sistem Pengukur Suhu dan Kelembaban Ruang Server," Eprints Undip, Feb. 2014, Art. 10. 42290.

[28] F. Vitallenko, "Prototipe Alat Pengukur Suhu dan Kelembaban Berbasis Web," Digilib UNS, Jul. 2011, Art. 10. 5434. 978

[29] S.B. Wibowo, et aI., "Development of Embedded Gateway for Wireless Sensor Network and Internet Protocol Interoperability ", Proceeding of 2014 6'h International Conference on Information Technology and Electrical Engineering TS 1- 28, October 2014, [6'h ICITEE Indonesia, p.150,2014].

[30] D.V. Rama, Sistem Informasi Akuntansi, Jakarta, Indonesia: Salemba,2008.

[31] JL. Whitten and L.D. Bentley, Systems Analysis & Design Methods, 7 h Edition, New York, USA: McGraw-Hill,2 007.

[32] V. N. Cartcnsen P., "Design of a Web-Based Lnformation System - New Challenges for System Development," in 9th Conference on European Conference on Information Systems, Bled, Slovenia, 2001.

[33] V. Dedik, 'Thesis Management System for industrial Partner Red Hat," 2013. [Onlinc]. Available: http://is,muni,cz)thl374278/fi_blthesis-texLpdC [Accessed 2013].

[34] Alexa. Alexa internet, inc [online]. 2015. [cit. 2015-3-5]. Dostupné na internete: http://alexa.com

[35] Seel, N. M. 2012. Encyclopedia of the Scienes of Learning. New York: Springer Science & Business Media, 2012. 1109 s. ISBN 978-14-419142-8-6.

[36] Kozik, T., Šimon. M. 2013: Preparing and Managing the Remote Experiment in Education. International Journal of Online Engineering (iJOE) 9.1 (2013): pp. 21-25.

[37] Mišút, M., Pribilová, K. 2013: Communation impact on project oriented teaching in technology supported education. In Lecture Notes in Electrical Engineering, 152 LNEE, pp. 559-567.

[38] Mišútová, M., Mišút, M. 2012: Impact of ICT on the quality of mathematical education 2012 IMSCI 2012 - 6th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings, pp. 82-86.

[40] Toman, J. Michalík, P. Possibilities of Implementing Practical Teaching in Distance Education. International Journal of Modern Education Forum, 2 (4), 2013, pp. 77-83

[41] Ölvecký, M., and Gabriška, D. Motion Capture as an Extension of Web-Based Simulation. In Applied Mechanics and Materials, 2014, Vol. 513, pp. 827-833.

[42] HORVÁTHOVÁ, Z., ZLÁMAL, J. Metódy uplatnenia informačných a komunikačných technológií vo vzdelávaní. In Technológia vzdelávania. Nitra : PF UKF, 2007. ISSN 1335- 033X.

[43] HOSŤOVECKÝ, M., ŠTUBŇA, J., STANKOVSKÝ, J. 2012. The potential implementation of 3D technology in science education. In ICETA 2012 : 10th IEEE International Conference on Emerging eLearning Technologies and Applications, November 89, 2012, Stará Lesná, The High Tatras, Slovakia. - Danvers : Copyright Clearance Center, 2012. - ISBN 978-146735122-5, P. 135-138

[44] O´reilly. Webportal [online]. 2015. [cit. 2015-2-7]. Dostupné na internete: http://archive.oreilly.com/pub/a/java/archive/what-is-aportlet.html?page=2

[45] J. Pearsall – B. Trumble. 1996. Oxford English Reference Dictionary. Published: Oxford University Press. ISBN-10: 019860050X; ISBN-13: 978-0198600503.

[46] A.Tatnall. 2005. Web Portals: The New Gateways to Internet Information and Services. Published in U.S by Idea Group. pp. 349. ISBN 1-59140-439-8

[47] K. Sotáková. 2010. Príprava žiakov a študentov na ich budúcu profesiu začína v škole. Zborník z medzinárodnej konferencie Generácia Y vstupuje na trh práce. Trnava: NKOS, 2010. 77 s. ISBN 978-80-970464-3-9.

[48] P. Yang, Y. Cao, and J. Evans, "Web Map Server perfonnance and client design principles," The journal of Geolnformation Science and Remote Sensing, vol. 44(4), 2007, pp. 320-333.

[49] P09as, J. Gon9alves, B. Marcos, J. Alonso, P. Castro and P. Honrado,"Evaluating the fitness for use of spatial data sets to promote quality in ecological assessment and monitoring," International Journal of Geographical Information Science,voI.28, 2014,pp. 2356-2371.

[50] “Software Prototyping,” 23 07 2008. [Online]. Available: http://pt.slideshare.net/drjms/software-prototyping. [Acedido em 19 05 2016].

[51]“Teach-ICT A2 Level ICT OCR exam board - The System Life Cycle, prototyping, rapid application development, RAD,” [Online]. Available: http://www.teachict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg4.ht m. [Acedido em 29 05 2016].

[52]“Teach-ICT A2 Level ICT OCR exam board - The System Life Cycle, prototyping, rapid application development, RAD,” [Online]. Available: http://www.teachict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg3.ht m. [Acedido em 29 05 2916].

[53] Media, “Incremental Prototyping,” 2007. [Online]. Available: http://www.sqa.org.uk/elearning/IMAuthoring01CD/page_09.htm. [Acedido em 29 05 2016].

[54] S. Komatineni, “Reshaping IT Project Delivery Through Extreme Prototyping,”11152006.[Online].Available: http://archive.oreilly.com/pub/a/onjava/2006/11/15/reshaping-it-projectdelivery-through-extreme-prototyping.html . [Acedido em 29 05 2016].

[55] T. Otwell, “http://laravel.com/,” [Online]. Available: http://laravel.com/. [Acedido em 10 12 2015].

[56] “GitHub - kEpEx/laravel-crud-generator: php artisan command to generate fully working crud with grid paginated server side only by having database tables,” [Online]. Available: https://github.com/kEpEx/laravel-crud-generator. [Acedido em 31 03 2016].

[57] Yuan Jing, Cao Yaoqin, Wang Wenhai, Li Jidong. Net Based Office Automation System Based on PHP Technology [J]. Microcomputer development. 2003.8:61-63

[58] Zhou Qihai, Liu Yunqiang. Design and implementation of personnel management system based on PHP, MYSQL and Apache [J]. Computer era. 2004. 8:23-24.

[59] Nan Di. Design of hospital website based on PHP technology [J]. China health vision: medical edition. 2014. 3:175-176

[60] Zheng Hao. Design and Realization of Online Voting System through PHP [J]. Jilin Normal University Journal (Natural Science Edition). 2014. 1:78-80

[61].https://trends.google.com/trends/explore?date=all&q=yii,CodeIgniter,Zend%20Framework,Cakephp,Laravel&hl=en-US

Https://trends.google.com/trends/explore?date=all&q=yii,CodeIgniter,Zend%20Framework,Cakephp,Laravel&hl=en-US

[62] Jeremy J. Sydik, Design Accessible Web Sites, Pragmatic Bookshelf,2007 www.elug.ca/reviews/design_accessible_websites.shtml

[63] Nigel Chapman, Jenny Chapman, "Web design a complete introduction", John Wiley & Sons Ltd, 2006.

[64] V. N. Cartcnsen P., "Design of a Web-Based Lnformation System - New Challenges for System Development," in 9th Conference on European Conference on Information Systems, Bled, Slovenia, 2001.

[65] V. Dedik, 'Thesis Management System for industrial Partner Red Hat," 2013. [Onlinc]. Available: http://is,muni,cz)thl374278/fi_blthesis-texLpdC [Accessed 2013].

[66] N. B. Sadri, tIe-Thesis Management System (eTMS)," 2011. [Onlinc]. Available:http://umpir.ump.edu.my/4311I1INURSY AZWANA_BfNTl_SADRl.PDF. [Acccssed 2013].

[67] "Ldea Source: Sharing Ideas and Promotong Lnnovations," [Onlinc]. Available: http://ideasource.blankdots.com. [Accesscd 2013].

[68] S. S. Mohamed Sayeh, "Web-Based PhD Thesis Management System for Tripoli Faculty of Computer Technology in Libya (WBPTMS)," Univcrsiti Utara Malaysia, Kcdah, Malaysia, 2014.

[69] A. P. Pantola, "Thesis Coordinator System ver. 1," Manila, Philippines, 200!.

[70] F. B. T. lsip, "Encouraging the Use of Slovin's Formula in Computing Sample Sizes in DMS Survey Related Projects," 2015.

[71] P. Sparrow, "What is Spiral Life Cycle Model?," [Online]. Available: http://www.ianswer4u.com/20 11/12/spiral-model-advantages-and.html#axzz33v.rllCzsVN. [Accessed 2015].

[72] "What is Prototyping Model?," [On line]. Available: http://searchcio.techtarget.comldefinitionlPrototyping-ModeI.

[73] "What is Waterfall Model?," [On line]. Available: http://searchsoftwarequality.techtarget.com/definitionlwaterfall-model.

[74] Department of Health and Human Services, Centers for Medicare and Medicaid Services, March 2008. [On line]. Available: ht tps:l/www .cms.gov/research-statist ics-data-and-systems/cmsinformation-technology/x1c/downloads/selectingdevelopmentapproach.pdf.

[75] 2015.[Online] Available: http://associationdatabase.com/awsINCDAlasset_managerlget_file13404.

[76] T. P. Andrew, "Theses Alive! : an E-theses management system for the UK," 2005. [Online]. Available: https:l/www.era.lib.ed.ac. uklbitstreamlhandle/1842/4231 ASSIGN _ t hesesal ive. PDF?sequence= I &isA Ilowed=y.

[77] M. S. Bandor, "Quantitative Methods for Software Selection and Evaluation," 2006.

[78] A.-M. Aadamsoo, "Web-Based Project Management System," Vaasa, Finland, 2010.