UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A DIGITAL SIGNAGE SYSTEM BASED ON THE CONTENT MANAGEMENT SYSTEM JOOMLA

NICOLÁS LÓPEZ CANO
2020

TRABAJO DE FIN DE GRADO

Título:	Diseño y desarrollo de un sistema de Digital Signage basado en el Sistema de Gestión de Contenidos Joomla
Título (inglés):	Design and Development of a Digital Signage System based on the Content Management System Joomla
Autor:	Nicolás López Cano
Tutor:	Carlos A. Iglesias Fernandez
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	——
Vocal:	——
Secretario:	——
Suplente:	——

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A DIGITAL SIGNAGE SYSTEM BASED ON THE CONTENT MANAGEMENT SYSTEM JOOMLA

Nicolás López Cano

2020

Resumen

El objetivo de este proyecto ha sido el diseño y la implementación de una extensión de Señalización Digital, que permita visualizar la información multimedia de los artículos presentes en las paginas web del Sistema de Gestion de Contenidos llamado Joomla!.

Para realizar esto, se estudió las ofertas del mercado y los diferentes tipos de extensiones presentes en Joomla, para decantarse por realizar un Módulo.

Con todo esto, se ha desarrollado un Módulo que permite, además de visualizar las imágenes de los artículos, visualizar el Autor, el Título del artículo y las primeras líneas del texto, de un forma clara y bonita.

Para ello, obtenemos la información de los artículos en forma de un fichero JSON, el cual, nuestro modulo parsea a datos. Una vez con estos datos crea unas diapositivas formadas por las imágenes y el resto de datos.

Estas diapositivas se crean de forma dinámica en función de los parámetros que se introduzcan en la parte de administración del Modulo, permitiendo escoger entre múltiples opciones. Se actualizan automáticamente cada un cierto tiempo, configurable también.

Desde la configuración, podemos escoger qué datos, como el título, autor, texto; y de que forma queremos que se muestren en pantalla y cuáles no.

Este proyecto permite visualizar la información de cualquier articulo de una forma clara, con los datos que se escojan en su amplia configuración, de forma que sea una herramienta útil y sencilla de utilizar.

Palabras clave: Sistema de Gestion de Contenido, Joomla!, Módulo, JSON, Diapositiva, Configuración

Abstract

The objective of this project has been the design and implementation of a Digital Signage extension, which allows the visualization of the multimedia information of the articles present in the web pages of the Content Management System called Joomla!

To do this, we studied the market offers and the different types of extensions present in Joomla, to decide on a Module.

With regard to all this, a Module has been developed that allows, besides visualizing the images of the articles, to visualize the Author, the Title of the article and the first lines of the text, in a clear and beautiful way.

To do this, we obtain the information of the articles in the form of a JSON file, which our module parses to data. Once this data is obtained, it creates a slide show formed by the images and the rest of the data.

These slides are created dynamically according to the parameters that are introduced in the administration part of the module, allowing to choose between multiple options. They are automatically updated every certain time, configurable as well.

From the configuration, we can choose which data, such as the title, author, text; and in what way we want to be shown on screen and which not.

This project allows to visualize the information of any article in a clear way, with the data chosen in its wide configuration, so that it is a useful and easy to use tool.

Keywords:Content Management System, Joomla!, Module, JSON, Slide, Configuration

Agradecimientos

En esta parte toca hablar de todas las personas que han hecho posible que haya llegado hasta aquí.

En primer lugar a Julia, mi mayor apoyo en mi día a día. Por animarme a no rendirme nunca, seguir esforzándome, ayudarme a ser mejor todos los días y soportarme de vez en cuando. Por todas esas tardes en Skype, apoyándonos para lo que sea.

Después a mi Madre, por estar siempre ahí, pase lo que pase y para lo que la necesite.

Por supuesto esto no habría sido posible sin todos mis amigos y compañeros de la Escuela.

A Maite, Mario y Jaime, por ser el mayor *Team* posible, para estudiar y para divertirnos.

A Luís, por enseñarme más de programación y de la carrera de lo que nunca nadie ha hecho.

A Charlas, por ser la luz en muchas partes que estaban muy oscuras y destinarme siempre que puede un hueco de su agenda.

A todo mi grupo en definitiva que, luchando como un equipo entre todos, hemos llegado increíblemente lejos.

Doy las gracias también a Carlos, por confiar en mí, pese a ir siempre muy justo de tiempo. Por estar disponible para cualquier duda y su ayuda en muchos aspectos.

Os debo mucho a todos y siempre lo tendré en cuenta.

En definitiva, *gracias*.

Contents

Resumen **I**

Abstract **III**

Agradecimientos **V**

Contents **VII**

List of Figures **XI**

1 Introduction **1**

1.1 Context 2

1.2 Project goals 2

1.3 Structure of this document 3

2 Enabling Technologies **5**

2.1 Definition 6

2.1.1 Introduction 6

2.1.2 Major Uses 7

2.2 Relation with CMS 8

2.2.1 What a Content Management System is 8

2.2.2 Digital Signage in CMS 9

2.3 Programming languages 10

2.4 Joomla 11

2.4.1 Extensions 12

2.4.1.1 Components 12

2.4.1.2 Modules 13

2.4.1.3 Plugins 14

2.4.1.4 Templates 15

3 Requirement Analysis **17**

3.1 Introduction 18

3.1.1 System actors 18

3.2 Use cases 19

3.2.1 Display and view multimedia information 20

3.2.2 Manage Module configuration 21

3.3 Functional and non-functional 22

4 Architecture **25**

4.1 Introduction 26

4.2 Joomla 27

4.3 DSJoomla 28

4.4 DSJoomla files 31

4.4.1 mod_DSJoomla.xml 32

4.4.2 index.html 33

4.4.3 mod_DSJoomla.php 33

4.4.4 default.php 33

4.4.5 helper.php 34

4.4.6 DSJoomla.css 35

5 Case study **37**

5.1 Introduction 38

5.2 Case Study: Display and view multimedia information 38

5.2.1 Overview of the Use Case 38

5.2.2 Display and View multimedia information 39

5.3 Case Study: Display and view multimedia information 41

5.3.1 Overview of the Use Case 41

5.3.2 Manage Module Configuration 43

5.4 Summary 46

6 Conclusions and future work 47

6.1 Conclusions 48

6.2 Achieved goals 48

6.3 Future work 49

Appendix A Impact of this project i

A.1 Social impact i

A.2 Economical impact ii

A.3 Environmental impact ii

Appendix B Economic budget iii

B.1 Physical resources iii

B.2 Human resources iii

Bibliography v

List of Figures

3.1 Use cases 20

3.2 Display and view multimedia information Sequence Diagram 21

3.3 Manage Module configuration Sequence Diagram 22

4.1 Architecture 26

4.2 Site sequence diagram 29

4.3 Admin sequence diagram 30

4.4 User interaction 31

4.5 Structure 32

4.6 Files Diagram 36

5.1 Display and view multimedia information 39

5.2 Example article 40

5.3 Example of Slide 1 40

5.4 Example of Slide 2 41

5.5 Example of Last Slide 41

5.6 Manage Module configuration 42

5.7 Administration Site 43

5.8 Configuration View 43

5.9 Category options 44

5.10 Content options 44

5.11 Example with all the options active 45

5.12 Example with only the Readmore activated and changed 45

5.13 Last Slide configuration 45

5.14 Last Slide 46

5.15 Last Slide without background and text changed 46

CHAPTER 1

Introduction

In this first chapter, we will begin by looking at an introduction to the current context of the project, an outline of the project's goals and detailing the structure of the document.

1.1 Context

Today, we find screens in almost every area of our lives. From walking down the street, at home, at work, in restaurants, in the subway, we see screens with advertising, information or offers and more. This is due to the great growth of the last years in Digital Signage [32]. Showing multimedia content, choosing what is shown from a central computer that does not have to be even in the same country, is something very usable by companies, so that, now walking around Times Square or Gran Vía is equivalent to seeing ads constantly through Digital Signage.This is because, through this technology, it is possible to capture more attention and that consumers remember more what they have seen, than if it were a traditional poster or public [6]. In addition, it is more convenient to manage the menu of a restaurant franchise from a central office than from each of them.

Because of this, the number of companies that are dedicated to this service grows every day. There are companies that manage the content you show, in what way, with what style and even some that provide directly the hardware itself.

And this is where the Content Manager System or CMS [26] comes in. All this information to show is managed much more easily from a centralized management system. That is why many companies offer their Digital Signage service in the CMS, with monthly subscriptions of 15€ or by payments of 800€ at a time to have the service for life, from where we can deduce that it is a service with high demand on the rise. Our extension for Joomla allows you to display the content of the articles you choose from the Joomla administration panel.

1.2 Project goals

The main objective of the project is to create an extension for Joomla that shows audiovisual content in an attractive way.

This extension will be validated in the GSI department, showing content related to the topics of the talks given in the department. To achieve this, several goals have been set.

- Create an extension that displays the contents of a folder.
- Obtain the content of the department's articles.
- Display all content in a clean and beautiful style.
- Add all administration options to make it configurable.

1.3 Structure of this document

In this section we provide a brief overview of the chapters included in this document. The structure is as follows:

Chapter 1. Introduction Brief description of the current context, project objectives and structure of the document.

Chapter 2. Enabling technologies Presentation of the concept of Digital Signage and its uses in the most used Content Management Systems actually.Also, an introduction of the programming languages used and description of the *CMS Joomla* is given.

Chapter 3. Requirements

In this chapter the necessary requirements are analyzed, by means of Use Cases and seeing the UML diagrams. we will also see, the Functional and non-functional Requirements of our System.

Chapter 4. Architecture

In this chapter, we will see a detailed description of the architecture of the system and the files that compose it.

Chapter 5. Case Study

In this chapter, we will describe the use cases in depth by looking at all the utilities of the system for the users.

Chapter 6. Conclusions and future work

In this chapter, we will see the conclusions of the project and future work to continue the development.

CHAPTER 2

Enabling Technologies

In this chapter, we will explain what is Digital Signage. Next, we will describe what a CMS is, naming the most important ones and their relation with Digital Signage and some examples. Moreover, we will see the programming languages that have been necessary to carry out the work and explain the CMS in which it has been done.

2.1 Definition

2.1.1 Introduction

Digital signage [32] is a tool used to show multimedia information on screens, which can be both outdoor and indoor of the establishments. It is widely used in marketing and advertising. These screens can be a single screen or a group of screens that can work as one to give the feeling of a large single screen, or be managed each individually showing different information, even in different formats. It is common to see this technology in restaurants, showing the menu, in shops, showing offers almost any kind of business, we will go into detail later on. Another use that is increasing recently is that the screens are installed on the street, to give useful information about the city, events, weather to people who circulate on the street.

Focusing more on our case in particular, in most companies in the meeting rooms there are screens using this technology, whether they work all together to show a presentation, or adorning with the logo of the company.

As we've seen, this technology has two main parts:

- **Hardware**. Screens are the most characteristic hardware component to present Digital Signage. The great development and expansion of this technology goes very much in hand with the increasing quality of the screens and their lowering in price. Along with the evolution of the screens, Digital Signage has gone through 3 stages [18]. The first one with LED screens, in which it was difficult to show multimedia information. Later, in the second stage, content players are incorporated, allowing the reproduction of multimedia content. The last and current stage is the interactive one, in which the user can interact with the screen, getting a better customer service. Digital Signage has been very present for more than 15 years, for example on the flight information screens in airports. Today, more than 20 screens can be combined to simulate a giant screen, and all of them work together in perfect synchronization. Another great advance was made with touch screens, allowing not only to display information, but also for the user to interact with it. At first the cost of this type of screen was too high and the investment was risky, but today they are more affordable, are found in many places, highlighting the kiosks.
- **Software**. The software part was almost non-existent at first. The information was stored in a database manually and transmitted to the screen for display. When advances are made, the software is updated by entering the content managers, and

the information and configuration of all the screens can be updated from a central computer, which makes this task much easier. In the last stage, the software part is already very developed and the interaction of the users is implemented, allowing to obtain information directly from them.

Digital Signage is a very powerful and booming tool, due to its high flexibility, easy configuration and good results, added to the fact that the screens are getting cheaper and cheaper. The impact created to the client is much greater than with paper posters, even more so when a video is shown instead of a static image.

2.1.2 Major Uses

Let's see now the most common uses of digital signage today, the reasons and benefits of these.

- Restaurants

  Digital signage has become an indispensable tool for food establishments. We can see screens outside the establishments announcing the menu and the offers of the day, or inside the establishment with the complete menu. This allows businesses to update all the information they want from a centralized office that does not even have to be in the same city, thus greatly reducing the time that employees had to invest in updating this information manually. On the other hand, and more importantly, interactive touch screens, also known as kiosks, appear. These have significantly reduced queues when ordering, since it automates the process and, nowadays, customers prefer to order by this method than the traditional one [23]. The client can see what they are ordering and pay for it without having to interact with workers who can perform other tasks. In addition, it has been shown that visualizing food can have an impact on what customers are asking for and can encourage better ways of eating [30].

- Shopping malls

  In shopping malls, we find the largest number of screens with digital signage. It has been demonstrated the benefits for businesses of this technology [6], increasing sales and customer service, in addition to facilitating in many cases their location, with an interactive map or their decisions when choosing where to eat.

- Public areas

It is a growing trend, to find more and more information panels of the city council itself, with news, the weather forecast and other articles of interest to the inhabitants of the city. Also, of course, advertising of major brands, in increasingly larger screens and more relevant locations in the city, as it has shown the greatest attraction of interest and remains longer in the memory of people who see it [2].

- Transport

  We find digital signage on all modes of transportation, whether it's a news video on the subway, information screens on buses and planes, or even television screens in taxis and companies like *Uber*. Whether it's for entertainment, event information or advertising, we find this technology every day in our lives.

- Companies

  Joining several screens together to make one big screen is an everyday concept in business. During a meeting it can be implemented in such a way that all the necessary information is displayed in the way that is needed to capture more attention.

- Educational establishments

  In schools and universities, we find many examples of digital signage. The screens become interactive, there are posts with news of the center updated in real time, even menu boards in the cafeteria. In the process of digitization and modernization of educational centers this is a good step [4].

As we have seen, Digital Signage is a really useful technology, which is being used in more and more places. And it is a trend that will continue to grow in the coming years.

2.2 Relation with CMS

Now we will see another fundamental part of the work and that is also a very powerful technology and more used every day around the world.

2.2.1 What a Content Management System is

A Content Management System or CMS [26], is a software application designed to manage content, mainly digital. The most used and that we are going to treat are Web Content

Management, which are oriented to develop and manage web pages. In addition, they allow you to create web pages in a simpler and more accessible way for everyone, without requiring the necessary programming knowledge to create a website. To do this, the systems are divided into two parts [3]. Content Management Application (CMA), this is the visual part of the system, allows the user to create, manage and operate both the front and the end of the web. The Content Delivery Application (CDA) compiles and executes the web code.

Some common characteristics of all the CMS for which they have succeeded are [3]:

- Create web pages without high knowledge of programming, based on previously designed and functional blocks developed by other users or companies.
- Manage the content with a configurable administration panel, its design and the way it is shown to the website visitor, through customize templates. It is also remarkable the facilities it offers regarding SEO positioning, file managers, logs registers and much more facilities.

2.2.2 Digital Signage in CMS

Now let's see what the relationship is between digital signage and CMS. Digital signage is about showing information, so a content management system is a very useful complement. With the components that exist in today's CMS, you can configure what information is displayed, in what form, how long, where to obtain the information and many other options. Following, we will see what each of the most used CMS offers us about this technology.

- Wordpress [37] is the most used CMS worldwide [33], and also the one with the biggest number of Digital Signage plugins. In Wordpress there are two clearly differentiated types: those that are purposefully created for Wordpress and those that are created by external companies adaptable to any platform, in which some even include hardware devices.

  Let's review one created for Wordpress, **Foyer** [16], which is the one with the best rewievs on the web of Wordpress for plugins [36]. One of the most important features of this plugin is that it is free and open source. The design is very neat and easy to use. You have to add the contents manually from Wordpress. It has online documentation and the creators actively solve doubts in the forum.

  Other examples of free plugins are EngagePHD [11] and Intuiface [19]. They have On the payment side, we find a wide range of prices ranging from monthly subscriptions of

$10 a month to product licenses for $800 from a single purchase. All are fully adaptable to user needs. Some include some added hardware to improve the management of the different screens Examples are: CommandCenterHD [7], Enplug DisplayOS [12], etc.

- Joomla does not count with a plugin like the previous ones seen for Wordpress specifically. The most similar extension that we found in this case are the slideshows, in which images are shown one after another, but it is far from a digital signage extension. The options of external companies that adapt their services to this platform, as in Wordpress, are still present.

- Drupal [10] is the last we are going to see, here the best we find is **Open Y Digital Signage** a module that allows us to create screens, screen schedules, content, modify sessions manually. This module is free and you can consult its versions and commits.

From all of the above, it can be concluded that digital signage and CMS are an optimal combination. You have the platform to show the content and get to improve your business and information of the users themselves, on the one hand, and on the other hand, a powerful tool to manage all the content you show, how you do it and what to do with the information you receive. Therefore, Digital Signage in CMS is a combination that has come to stay and grow more every day. As more and better extensions are developed for all the CMS's out there, it will be an unstoppable expansion to improve the way you see ads, interact in restaurants, get information and all the other features this powerful combination has.

2.3 Programming languages

Now we move on to look at the four programming languages that have been used for this project and that are the most used today in web development.

- **HTML**.

  HTML [28] or *HyperText Markup Language* is a language designed for the creation of web pages. This is not a programming language in itself, it is a language made up of tags, content and attributes that web browsers interpret.

  HTML does not include the design part, it simply indicates how and where the content is arranged for the browser to display.

- **CSS**.

CSS [27] or Cascading Style Sheets is a complementary design language to HTML in web page design. Although we have seen before that HTML tells us the order of the elements, CSS tells us their appearance, through styles. It is called cascading because the styles are applied from top to bottom. This concept is called inheritance, in which the children inherit the properties of their parents, especially those related to the text.

In the CSS document we find everything related to the presentation, while in the HTML we find the content, so that we can make changes in the visual part without altering the design and vice versa.

- **JavaScript**.

  JavaScript [14] is a client-side programming language. It completes, together with CSS and HTML, the pillars for the development of web pages. It allows you to create dynamic content, manage multimedia content, create animations and many more features.

- **PHP**.

  PHP [25] or (hypertext preprocessor) is a server-side programming language, used to generate dynamic HTML and, consequently, to create dynamic web pages. In these, when the request arrives at the server, the answer does not originate directly, as in static web pages, but rather, a data collection from the database and data processing is carried out, to send answers according to the request received. This produced a breakthrough in web development.

2.4 Joomla

As we have seen above, Joomla [35] is a Content Management System (CMS) that allows you to develop dynamic and interactive websites. From your administration panel you can manage all the content you want from the web. It stands out from the rest of CMS that is open source software released under GPL [31], so it is constantly improving and evolving by developers everywhere who are dedicated to it. The administration panel can be accessed via browser and must be running an Apache server and have a SQL database. It is a very powerful software that allows us not only to create webs in a simple way, but also intranets in the companies to manage the information.

For this, Joomla is based on php scripts. By means of these scripts the consultations and modifications in the data base are made, resulting in an updated web page, so that the browsers show it.

On the other hand, as we have already mentioned is the MySQL [8] database. This database is open source and the most used in the world in this category and in general. In this database all the information of the system is stored in tables so that it is easily consultable.

Joomla uses the Model-View-Controller or MVC [17] architecture, which allows for dynamic page modifications and reuse of code to optimize resources.

2.4.1 Extensions

To make content management as easy as possible, Joomla is composed of two parts:

- **Extensions**.

  Extensions [20] are a fundamental part of Joomla. They add or improve some functionality. The most important that we will develop later are the components, modules, plugins and on the other hand but important also the templates.

- **Content**.

  The content consists of articles. In these articles you can add text, images, videos, everything that can be in a web page. In addition, according to the extensions installed the possibilities of customization are endless.

2.4.1.1 Components

Components in Joomla are like small applications with full functionality, making it a CMS for much more than only articles or news. They have their own section in the menu of the administration part of Joomla, from where we can see all the components installed for our website.

They are divided into two parts, the administrator and the site. In the administrator part we can configure and manage different aspects of the component. In the site part, the pages requested by the visitors are render, with the required functionality.

To access them in an easy way, in the public part of the web, we can add to the menu a new item of this component.There can only be one component per page.

The structure of any component [21] is as follows

- **com_name.xml**.

In this file we can find the files and folders that *Joomla! installer* must install to make the component work correctly.

- **Site**.

  In the folder site we find the public part of the component. Inside, there are two files:

  - **name.php**.

    In this file, the component is initialized, requiring all the necessary files and parameters.

  - **index.html**.

    These are blank files, which are used to prevent the contents of our server's directory from being accessed externally.

- **Admin**.

  In this folder we can find the administration part, which is not publicly accessible. Inside we find the same two files as in site, but for the administration part and the sql folder.

  - **sql**.

    Here are the database configuration files, with an schema of the versions.

  - **index.html**.
  - **name.php**.

All components must be named with a com_ and the name of the component.

2.4.1.2 Modules

The modules are simpler extensions than the components. They are usually associated with the concept of a box, since their arrangement in the template is similar to this. There can be as many modules as there are holes in the template and some modules can occupy more than one hole, which gives us a wide range of possibilities for style of the pages.

In the administration part, in the tab of extensions we find the option modules, from where you can choose which are published, in which page, and the configuration of each module.

The basic structure [22] of a module is

- **mod_name.xml**.

This file describes the module. Here, we can find the files and folders that *Joomla! installer* must install and all the configurable parameters that the module has.

- **mod_name.php**.

  It is the entry point for the module, since all the initial processes are done there. All the necessary files for the correct functioning of the module are called, the template is loaded and the necessary data is obtained from the configuration.

- **helper.php**.

  In this file is where most of the communication with the database takes place, retrieving and processing the information through the helper class, to finally show it.

- **tmpl/default.php**.

  In this file is generated the HTML that will show the page, depending on the parameters that passes the mod_name.php. It is located inside the tmpl folder, due to it is the template of the module.

All modules must be named with a mod_ and the name of the module.

2.4.1.3 Plugins

Plugins are small pieces of code that are executed when a certain trigger is triggered. For this, Joomla uses JEventDispatcher as an event controller, where all activated triggers are registered and which plugin they activate.

They are always executed before showing any other output, allowing this way to make modifications in other modules or components. Therefore, there are two different types of plugins, those that are basic for the correct operation of other extensions or the Joomla core, and those that add functionality to other extensions but are not critical for their operation. The typical plugin consists of two files:

- **plg_name.xml**.

  In the XML file we configure the plugin and put the fields we want to be filled in or the options we want to be chosen.

- **plg_name.php**.

  In the php file, the requested parameters, in the xml, are loaded and write the syntax of the code we want to run.

All plugins must be named with a plg_ and the name of the plugin.

2.4.1.4 Templates

In the templates you define the style of the screens that are displayed. They can be responsive pages, specific to a component or a full screen image, for example.They're divided into gaps, where the modules will go. These gaps are personalizable, and you can configure your layout to make the page as attractive as possible.

Once this chapter is finished, we can state that our system is a very current and demanded technology, developed with the most used programming languages in web development.

CHAPTER 3

Requirement Analysis

In this chapter we will see a fundamental part of software development, the obtaining of requirements. We will also see cases of uses and the actors involved.

3.1 Introduction

Requirements analysis [1] or requirements engineering is a fundamental part of software development these days. Requirements are a feature or quality factor that a system needs to be useful and valuable to users. Therefore, requirements engineering tries to systematize the analysis of the requirements, so that when they are changed, the impact is minimized. It helps us to identify what we have to develop and in what way, being able to highlight the most relevant features and those that are secondary to optimize the work. In addition, it serves us as a source for quality control with KPIs [15], for example.

There are many things that can be identified in this analysis. We will focus on functional and non-functional requirements, main actors of our system and use cases.

3.1.1 System actors

The system's actors are all the elements that intervene in it, either internally, as another Joomla extension, or externally, as a user. Each one plays a specific role and has its own unique name. They have great importance in the requirements analysis, especially for use cases.

The actors we have identified are:

- **User.**

  The ordinary user. This actor is the target of the content shown by the Module in Joomla. It cannot manage the content but can only view it.

- **Admin.**

  It's one of the main actors in the system. It can configure and visualize the content, choosing the best options for it, from the Joomla administration panel.

- **Joomla.**

  This actor is the front end of our application. The module runs in Joomla, showing the information for the user to see. It also allows the administrator to access the control panel, from which the module is configured.

- **Articles.**

  From this actor, the System obtains the information to be shown.

3.2 Use cases

Use cases are different scenarios in which the system performs a particular set of actions, producing value for certain actors.

These use cases are useful to obtain the functional requirements of the application. In its diagram it allows to see in an easy way which actors participate in which use case and which cases there are. With the actors identified above, we formed these diagrams, looking at the relationships between them and the system.

To visualize them, we use UML [24] diagrams or Unified Modeling Language. This language is very useful when representing a system with standard drawings that everyone understands. There are many different UML diagrams depending on which part of the development process or the application we want to represent. These types are grouped into two categories:

The *Structure diagrams* which are focused on showing the static structure of the objects in the system. Some examples are class diagrams, component diagrams, object diagrams, etc.

And the *Behavior diagrams* which shows the interaction and dynamic behavior of objects with the system. They are the most known and used. We highlight the state machines, interaction diagrams, activities and, the best known and most interesting, the use cases.

Now let's look at the two main use cases and they will be explained in depth later on. The first one is done by the user and is to visualize the information. The second one is done by the administrator and consists in managing the content to be shown. The administrator can also see the information but the user cannot manage it.

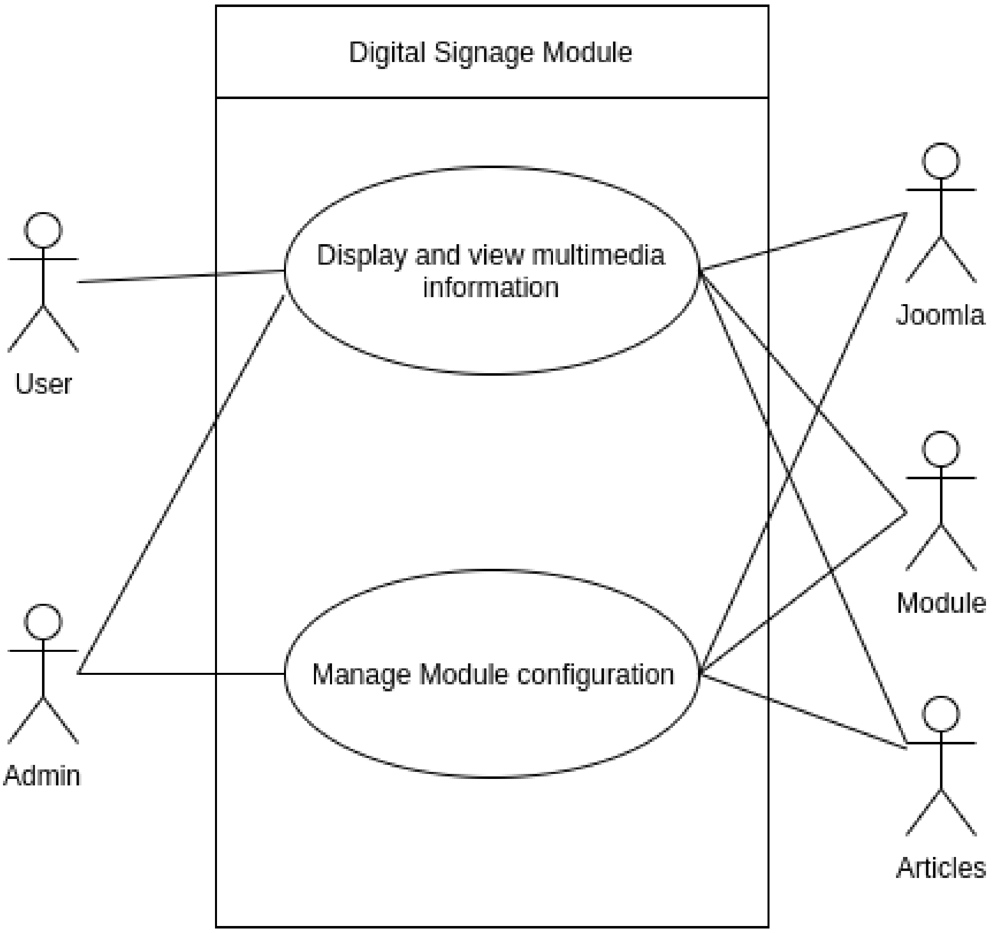

Digital Signage Module
Display and view multimedia information
Manage Module configuration
User
Admin
Joomla
Module
Articles

Figure 3.1: Use cases

On the left side we see the main actors of the use cases, *User* and *Admin*. On the right side we see the secondary actors but without which it would not be possible to do this, which are *Joomla* and the *Articles*. And in the center the two use cases **Display and view multimedia information** and **Manage Module configuration**. As we see all, the secondary actors are involved in all cases because to visualize the information you need to access Joomla, use the System and read the information in the articles, so everyone is involved. To manage the module the same thing.

Below are the details of these two cases.

3.2.1 Display and view multimedia information

This use case starts when the user or admin opens Joomla to view the content in the articles.

Joomla accesses the System and displays it in the template slot assigned to it, with the settings and information that had been saved from the previous session.

Meanwhile, the System obtains from the articles the new information, in case there is any, to show it too. So, the System always has information to show and it is updated whenever it is activated, making sure to show always updated information.

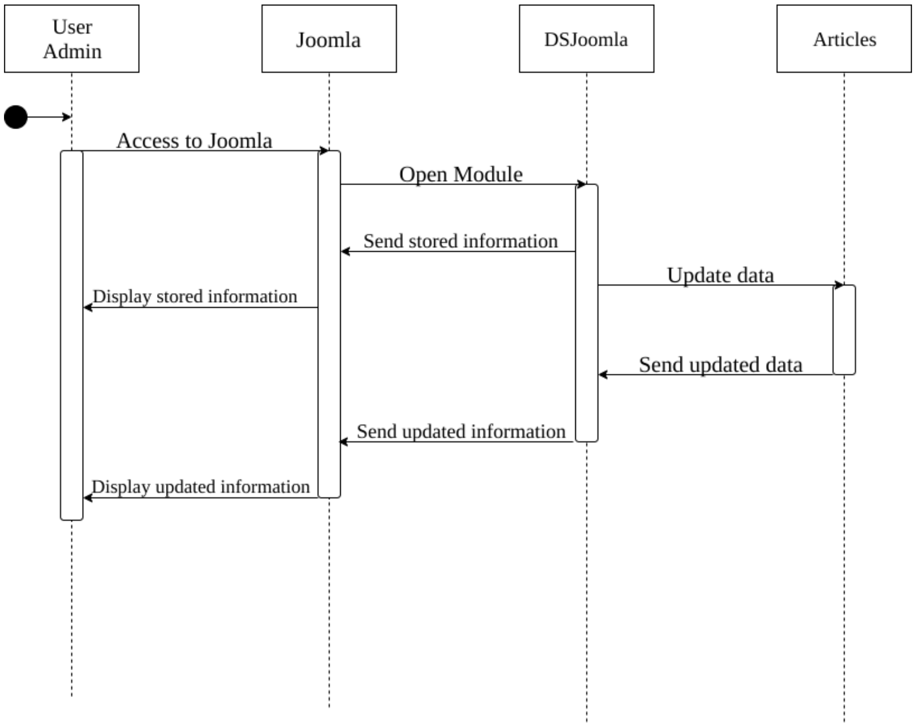


Figure 3.2: Display and view multimedia information Sequence Diagram

3.2.2 Manage Module configuration

This case of use starts when the administrator opens Joomla to change the configuration of the module.

In this case, when registering in Joomla as an administrator, he accesses the Joomla administration panel, instead of the normal view for the rest of the users. Joomla accesses the administration panel of the module and shows the options to configure it. The administrator chooses the options he wants to modify and selects them. Joomla changes the parameters of the module, which checks the information and obtains the data of the articles with this new configuration. Once you have the data updated the System processes it so Joomla can show the new configuration.

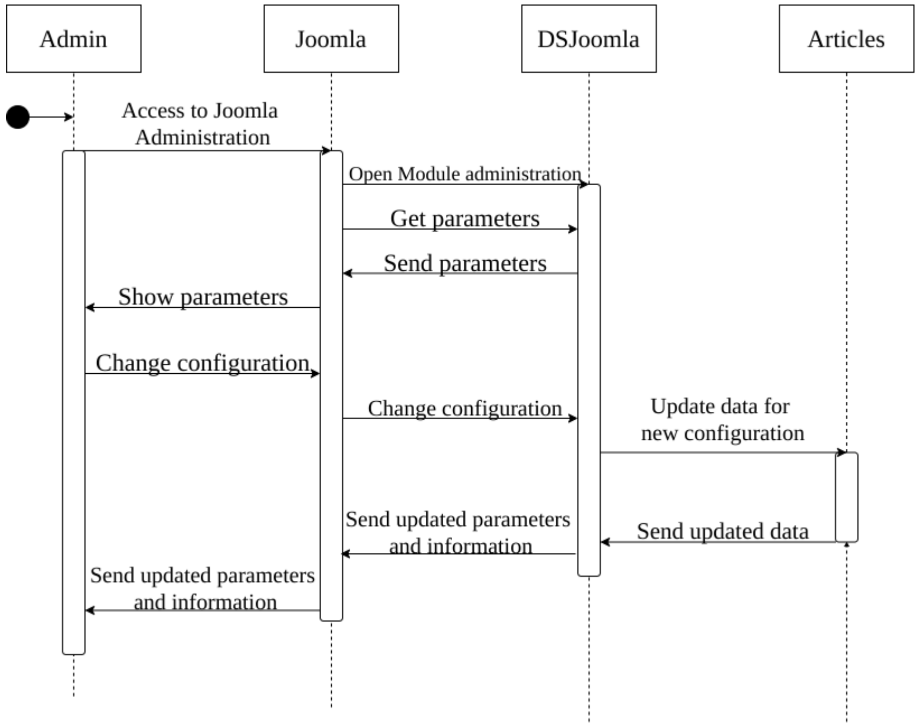

Admin
Joomla
DSJoomla
Articles
Access to Joomla Administration
Open Module administration
Get parameters
Send parameters
Show parameters
Change configuration
Change configuration
Update data for new configuration
Send updated data
Send updated parameters and information
Send updated parameters and information

Figure 3.3: Manage Module configuration Sequence Diagram

3.3 Functional and non-functional

Once we have identified the actors and seen the use cases, we move on to identify the functional and non-functional requirements of our system.

- **Functional requirements.**

  Functional requirements are functions or characteristics that describe what the system has and what adds value to our product. Here we will see ours.

  - **Show content of articles.**

    This is the main requirement of the system. It could be divided into more, but the functionality that brings real value is this. Our system must show the content you get from the articles on the screens, in a clear and simple way.

  - **Configurable.**

    The system must be configurable, allowing you to choose what you want to display, in what form and other options. With a simple and intuitive interface to make it easy to use.

- **Non-functional requirements.**

  Non-functional requirements are defined as quality attributes or restrictions of the system. They also describe how the system will perform its functions. We use the ESA-SRD classification [13].

  - **Interface.**
    - The system must comply with the interfaces (data and protocols) of the external systems with the ones it interacts with: Articles and other extensions.
  - **Security.**
    - Only authorized users can access the administration panel, to configure the module.
  - **Operation.**
    - The user interface is based on Joomla module.
  - **Portability.**
    - The system is exportable in a .zip file, in case we want to install it in another Joomla system. systems.

CHAPTER 4

Architecture

In this chapter, we will look in depth at the development of the system. We will start with an overview of the entire system. Later, we will explain the architecture of the system, going into depth in each part of the module.

4.1 Introduction

Now let's take a look at the architecture of our system.

We clearly differentiate two parts. Joomla and DSJoomla, our system. Each of these parts is divided in internal blocks, which are each responsible for a function of the system.

Next, we are going to see a diagram with all the processes that the system carries out in all its possible uses. We will explain how each one is activated, which path it follows and what the result is.

To do this, we will first explain each part with its separate blocks and then explain the overall process of the system.

Finally, we'll look at the files that make up the module to see how it is formed and how each one interacts with the system.

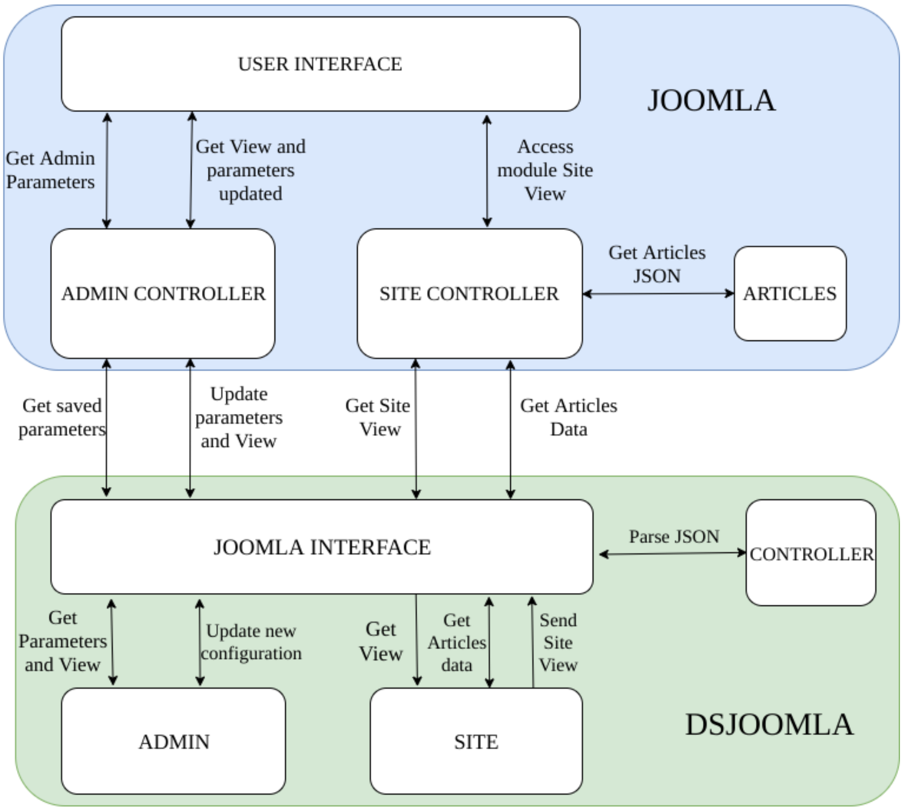


Figure 4.1: Architecture

In the diagram we see clearly the two parts and their blocks, and, by means of arrows, we express the relations between them. Now that we have seen the general description of the system, let's go into more detail with each part.

4.2 Joomla

As we have already said in chapter 2, Joomla is a fundamental part of our system.

Joomla is the visible part of our system. It communicates the users with the module, gets the JSON with the information of the articles and sends it to the module for processing. JavaScript Notation Object [34] or JSON is a language for the simple exchange of information. It is an alternative to XML [5], but its syntax is clearer and it is easier to parse, that is, to transform a text into usable data.

Let's see what blocks it's divided into and what each one does.

- **User interface.**

  As we said before, Joomla is the access point for users. This block is responsible for managing all user requests.

  There are two types of user requests. From the public part, looking for the module site or from the administration part, looking for the module administration part. Depending on the type of the User Interface you will communicate with the Admin Controller or with the Site Controller.

  With the Admin Controller to ask for the administration part of the module or to send the parameters that the administrator has introduced for the configuration of the module.

  With the Site Controller to obtain the view of the Module Site, that is to say to visualize the multimedia content of the articles.

- **Admin Controller.**

  The Admin Controller, once it has received the request of the User Interface to interact with the administration part of the module, communicates with *Joomla Interface* of DSJoomla, in charge of managing the requests of Joomla to the Module.

  There can be two requests as we have already seen. Get parameters and View, to see the configuration part with the parameters that are already configured in the system. And Update parameters, to update the configuration and return the updated View.

- **Site Controller.**

The Site controller is in charge of the requests to see the module's site. It also communicates with the Joomla Interface for this purpose.

The Site Controller makes the request of the site to the Joomla Interface and before sending it, it asks for the data of the Articles. At this point, the Site Controller communicates with the Articles within Joomla to obtain this data in a JSON file. Then, it sends it to the Joomla Interface, which as soon as it processes the information, sends the View.

- **Articles.**

  The Articles of our website, communicate on request with the Site Controller. Send a JSON with all your data to the Site Controller, when requested.

Also as we have seen in the Introduction, our system will be tested in the GSI website, which is also made in Joomla, so we will install the module and use it with the GSI articles.

4.3 DSJoomla

Now let's see how he acts and what blocks DSJoomla is made of.

Although we have separated Joomla and DSJoomla to make it easier to understand and see, our module is hosted in Joomla, but for the description of the architecture we will study it independently.

Our module is waiting for Joomla to send you a request, to respond with the views and data that are requested.

Let's see your blocks to understand how the system works.

- **Joomla Interface.**

  It is in charge of the communication between Joomla and the Module.

  It attends to the requests of both the Admin Controller and the Site Controller and redirects them to the Admin or Site part, depending on their origin.

  It also transmits the request of the site of the data of the articles, receives the JSON and sends it to the controller to parse the data. Once it receives the parsed data, it sends it back to the Site to generate the View and, when the Site responds, it sends the View to Joomla.

- **Admin.**

The administration part saves the configuration of the module.

It also generates the Administration View when prompted and updates the configuration parameters when the Joomla Interface sends them.

- **Site.**

  The Site part is in charge of the public part of the Module.

  When it receives a request from the Joomla Interface to display the View, before responding, it asks for the data of the articles.

  Once it has sent them, it generates the view with the updated data and the configuration parameters that the Module has and sends it to the Joomla Interface.

- **Controller.**

  The Controller is responsible for parsing the JSON when it arrives and sending the data back so that the Module can display it.

Now we'll look at two sequence diagrams of the system. One for the administration part and another for the site part.

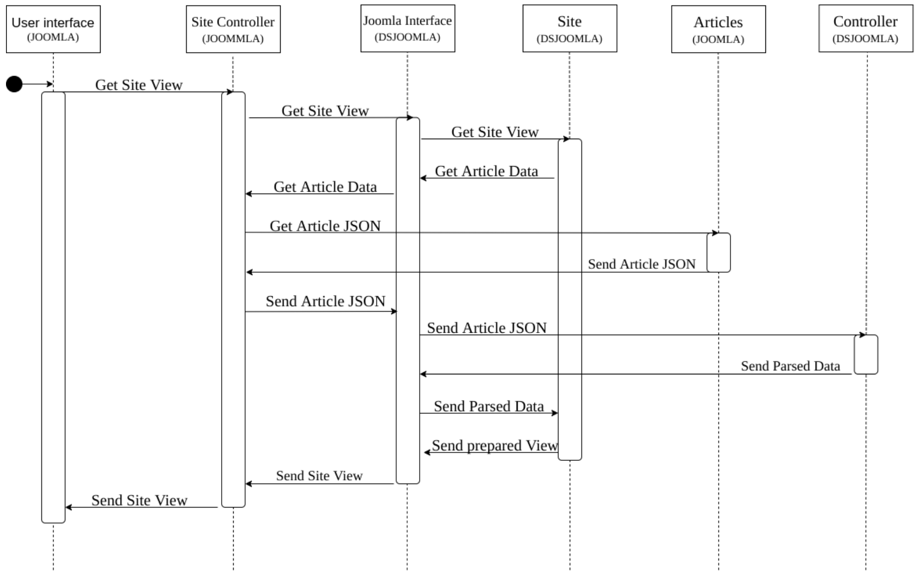


Figure 4.2: Site sequence diagram

In this diagram we see the process from the moment a request arrives to Joomla to see the Site View until it arrives.

First Joomla communicates with the Module and requests it. The module then makes a request to obtain the data of the article. Joomla responds with the data, our system parses it and generates the view that has been requested.

Once everything is ready, the Module responds to Joomla with the Site View.

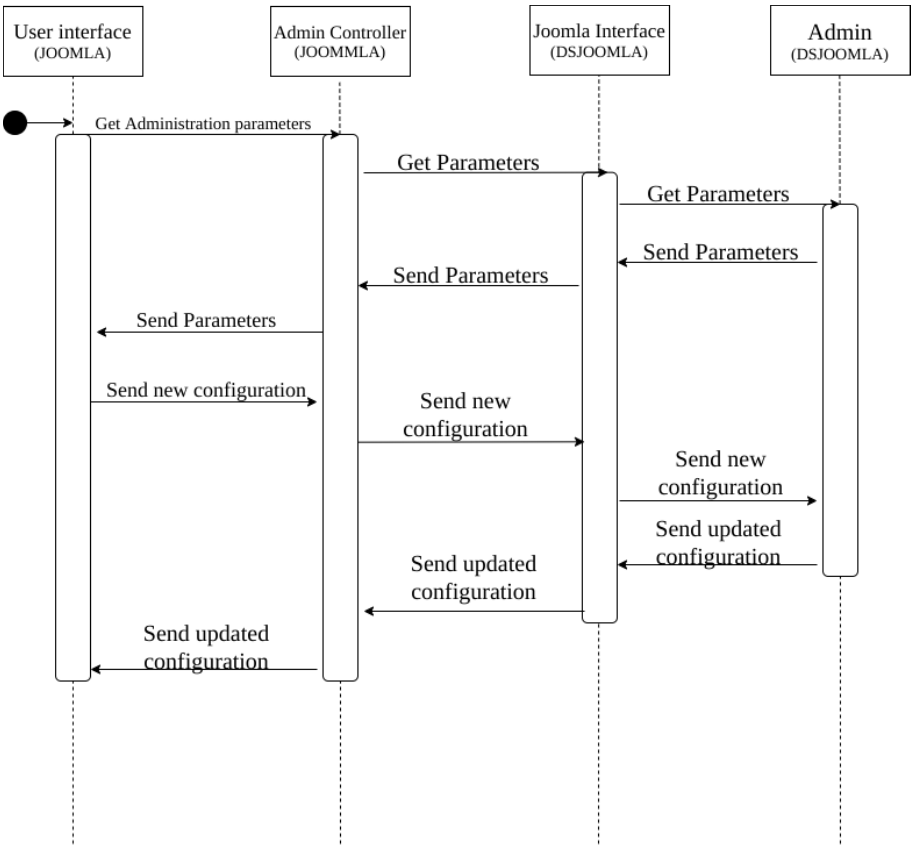


Figure 4.3: Admin sequence diagram

In this one we see the process on Admin's side.

First the Joomla sends the Administrator's request to the module to see the current configuration. The module responds with its saved parameters.

When Joomla receives the new parameters to be configured from the Administrator, it sends them to the Module to update them. This does so, and returns the new configuration to Joomla.

With the following diagram we'll see a summary of how the user interacts with the system.

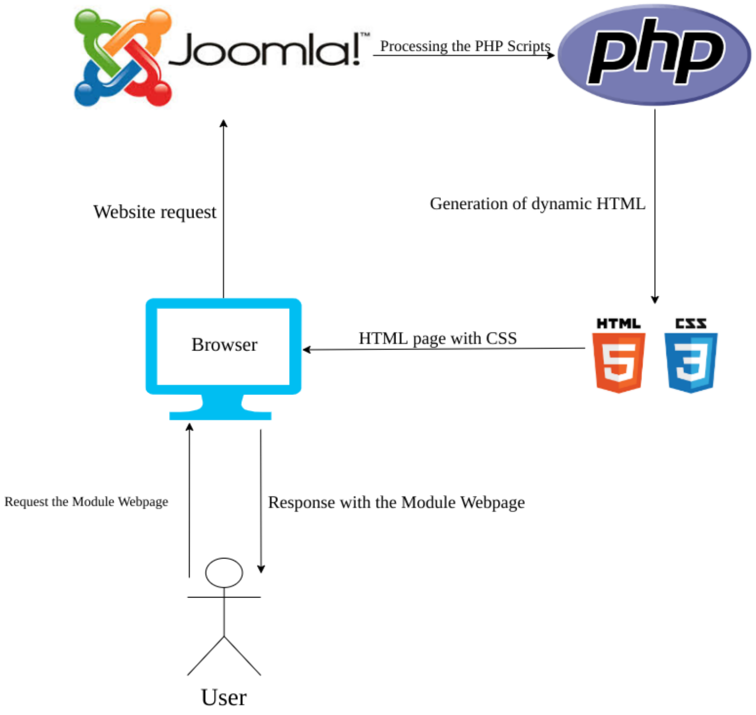

Joomla!
Processing the PHP Scripts
php
Website request
Generation of dynamic HTML
Browser
HTML page with CSS
HTML
CSS
Request the Module Webpage
Response with the Module Webpage
User

Figure 4.4: User interaction

The user through his browser makes a request to the Joomla server, to see our system. The browser asks Joomla for the HTML page of our Module.

Joomla accesses the file mod_DSJoomla.php. This file loads all the associates, including the default.php and CSS, so a dynamic HTML is generated based on the PHP scripts and the parameters obtained from the configuration. PHP accesses the database and obtains all the necessary data to create this file.

This HTML page is sent to the browser, which finally shows it to the user.

In this way, dynamic HTML files are generated according to the parameters in the module configuration and the data of the articles in the database, always showing the updated data when the module is accessed.

4.4 DSJoomla files

In this part, we will see all the files that make up the Module, explaining how each one works and then seeing how they relate to each other.

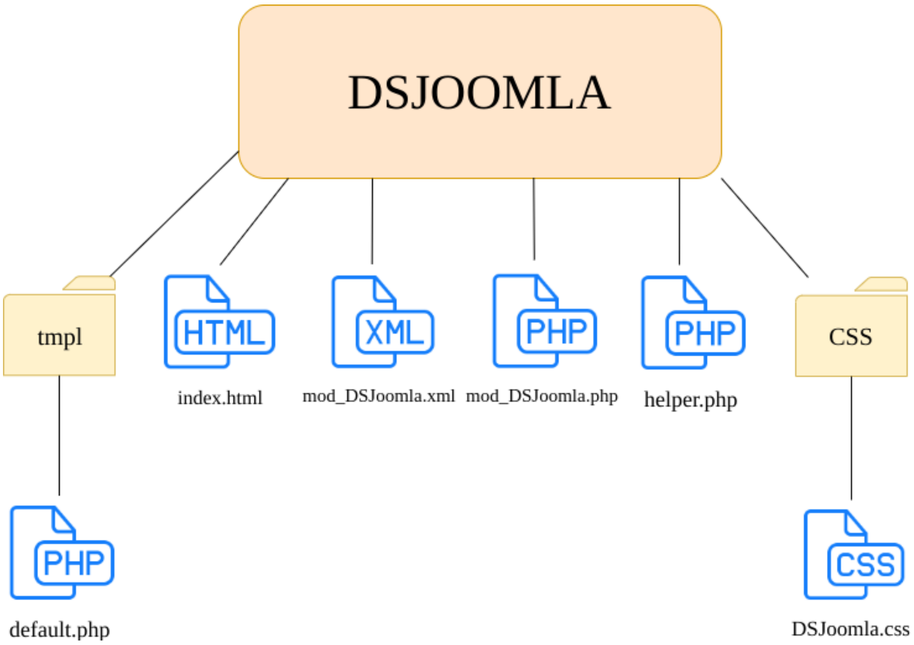


Figure 4.5: Structure

4.4.1 mod_DSJoomla.xml

This is one of the most important files in our module. It is written in XML. Extensible Markup Language or XML [5], is a language for the representation of data in a structured and standardized way. This way it is easier to write, read and manipulate this information.

As we saw, this file tells Joomla how to install the module. It indicates the type of extension, version, client, name, author, description and the files included in the module or folders.

This is the first function of this file, but in our case has another very important, create all the configuration parameters of our module. From here we can create all the options of the administration part of the module. You can create buttons or text boxes and then, in the mod_DSJoomla we will get these elements through PHP.

Therefore, this file has been of vital importance in the development of our module.

Now we are going to see the XML code of a configuration parameter of our module. With this, we create a yes or no button that allows to choose if we want to filter by category or not.

```
<field
    name="category" type="radio" class="btn-group" default="1" label="Filter
         by category" description="Filter articles by category">
    <option
        value="0">JNO</option>
    <option
        value="1">JYES</option>
</field>
```

Listing 4.1: Parameter Code

4.4.2 index.html

This file is used to make the module inaccessible from outside Joomla. In this case it is blank.

4.4.3 mod_DSJoomla.php

This is the entry point to our module. In this file we load the css file, the helper.php, the default.php and we load all the necessary variables so that later the Script is executed correctly and of administration.

4.4.4 default.php

This file is the one that generates the HTML code that we will see later in Joomla. This inside the folder tmpl, template, as it is the one that gives the template to the page itself.

In addition, this is where you get the JSON with all the data from all the articles. It is parsed to obtain the ones we want, in our case the images, the title and the URLs, to later create the Slides. In this way it has been easy for us, to get everything we want from the articles by parts, which is a great advantage to choose what to display and how.

Below, you will find the PHP code to parse the JSON of each article into a loop and obtain the parameters that interest us. In our case the images, the author, title and introtext.

```php
<?php
  foreach ($items as $key => $item)
  {
  $image = JURI::root() . json_decode($item->images)->image_fulltext;
  $author = JFactory::getUser($item->created_by)->name;
  $text = $item->title;
  $image1 = JURI::root() . json_decode($item->images)->image_intro;
  $intro = $item->introtext;
?>
```

Listing 4.2: JSON parsing

The base design of our application are the images obtained from the articles to full screen, with the title of the article visible in a rectangle. This is the base design, since the design is highly configurable. The background image is configured to occupy the maximum without repeating itself and centered. The title, which can be made visible or not from the administration, is placed in a rectangle with a low opacity background to allow the image to be seen.

You can also include a small text or description, which is also activated or deactivated from the administration and configure what it says. This text goes in a box similar to the title but on the opposite side.

There are other configuration options but they do not affect the design to a great extent, so we will see them later.

Also, since the system will be tested in the GSI meetings and talks, the last Slide is a configurable text to put the messages that the administrator wants, setting by default ”Please, do not forget to turn off the lights”.

4.4.5 helper.php

This file contains the Helper class, which is responsible for making the change of Slides.

In the first non-configurable version, this file was JavaScript, since what we did was a Script that changed the opacity of the Slides from visible to invisible every 3 seconds, with an interval.

In the configurable version, we changed the file to PHP since we need to get the new variable of time between Slides that the administrator has configured, but its operation is the same Script. This variable is obtained through PHP, obtaining the parameters of the XML file.

Following is the final Script, which allows the slides to change automatically.

```
const slides = document.getElementsByClassName('slide');
let index = 0;
index = nextIndex();
setOpacity(index, 1);
setInterval(() => {
  setOpacity(index, 0);
  index = nextIndex();
  setOpacity(index, 1);
}, time);

function setOpacity(index, opacity) {
  slides[index].style.opacity = opacity;
}

function nextIndex() {
  if(index === slides.length - 1) {
    return 0;
  }
  return index + 1;
}
```

Listing 4.3: Script for changing Slides

With nextIndex(), we move through the array of Slides.

setOpacity(), changes the opacity of the slides from 1 to 0, making them visible and invisible.

Finally, with setInterval(), we make this change every certain time interval, based on the time parameter that is chosen by the Administrator.

4.4.6 DSJoomla.css

In this file you can find the whole design part of the module.

The style of the images and texts is controlled from this file, as well as the fade effect and the size of the images. Slides should have an attractive style with the text visible and without it.

In digital signage, design is vital, as it has to be visually appealing to cause the greatest possible impact. We've created a class for part of the Slides. The full-screen image on the page. The Article title in the lower left corner with a text box. The Author in the upper left corner, the Readmore in the lower right corner and the Intro Text is in the upper right corner.

Also, there is a style for the Ultima Slide, with an activable background.

Next, in the following diagram we will see the interplay of the files in a diagram.

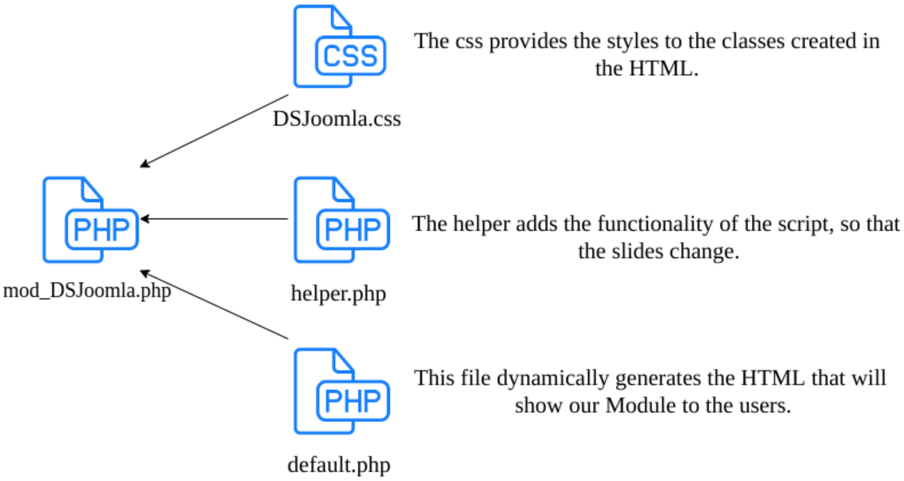

CSS
The css provides the styles to the classes created in the HTML.
DSJoomla.css
PHP
mod_DSJoomla.php
PHP
The helper adds the functionality of the script, so that the slides change.
helper.php
PHP
This file dynamically generates the HTML that will show our Module to the users.
default.php

Figure 4.6: Files Diagram

CHAPTER 5

Case study

In this chapter, we will explain in detail the two use cases, Display and view multimedia information and Manage Module Configuration. We will see how our system operates in each use case and how it interacts with the actors.

5.1 Introduction

Next, we will see in detail the use of our system through an in-depth description of the use cases.

Let's see how the prototype created, based on the previously defined architecture acts in each use case. To do this, first we will describe the actors and how they intervene, then we will explain the process and illustrate everything with diagrams and images of the prototype to illustrate and exemplify everything.

5.2 Case Study: Display and view multimedia information

5.2.1 Overview of the Use Case

In this use case the actors involved are:

- **User.**

  The user accesses the page where the module is located to see the multimedia information to be displayed. This case can also be started by the Administrator if he wants to see the information, but since he is more related to the second case, we will only put the User in this use case.

- **Joomla.**

  Joomla is the entry point for the user. Through his browser the user enters the Joomla page. It houses our system, the article and the database where the configuration is, so it is of great interest.

- **Articles.**

  The articles provide the system with all the information about the articles to be displayed in our system. The articles are organized by categories. They are composed of a title, a main text, the images and the URLs that you can attach to them.

In the following figure we will see how our system works in this Use Case and how each part interacts with the others.

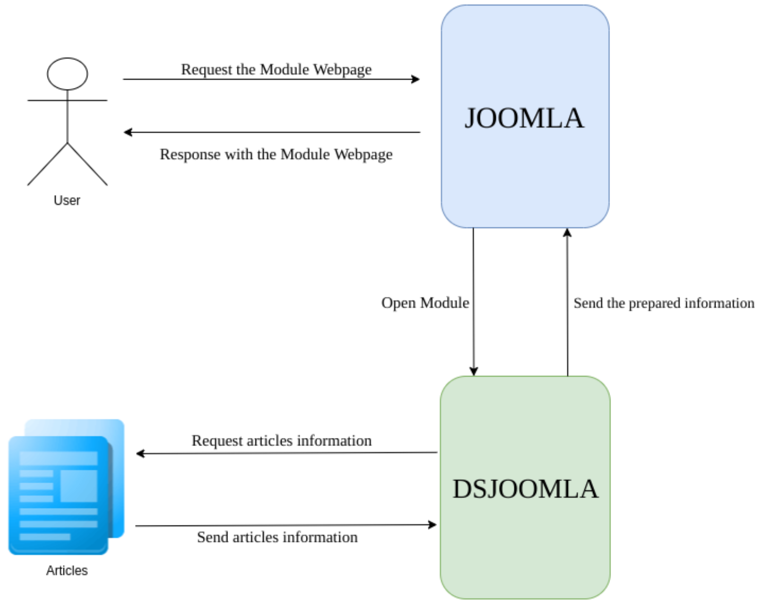

User
Request the Module Webpage
Response with the Module Webpage
JOOMLA
Open Module
Send the prepared information
Articles
Request articles information
Send articles information
DSJOOMLA

Figure 5.1: Display and view multimedia information

The user accesses the Joomla page to use the module, looking to view the multimedia information of the articles.

Joomla opens it and the module obtains the data of the articles that interest him. It processes them according to the configuration and sends them to Joomla to show the information requested by the user.

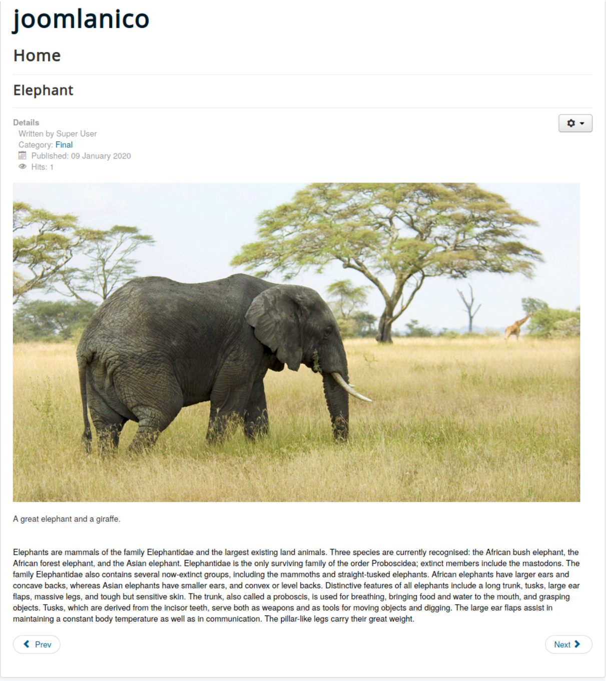
5.2.2 Display and View multimedia information

In this case of use the user will want to see or show the multimedia information that is in the articles of the page in which the module is. Therefore, this Use Case starts when the User or the Administrator accesses the Module.

For this, the module will have to be published and visible by the administrator of the page, and the user must access its URL.

At this point the module starts to create the view of the Site part. To do this it needs the information of the articles, so it makes a request to Joomla to get the JSON with all the data and parse it.

Once it has the data, depending on the configuration set by the administrator, at that time, it creates the view for Joomla to show the user.



joomlanico

Home

Elephant

Details
Written by Super User
Category: Final
Published: 09 January 2020
Hits: 1

A great elephant and a giraffe.

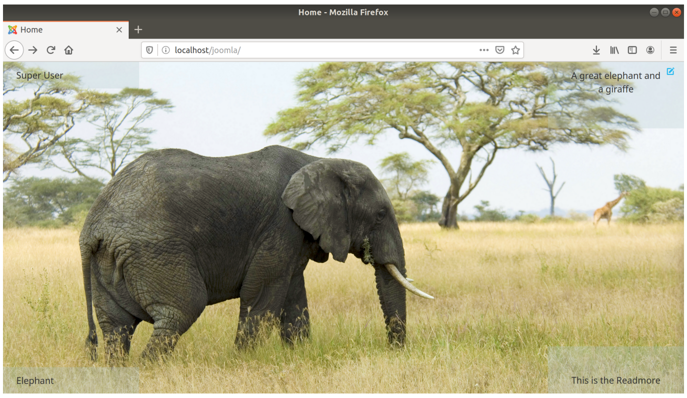
Elephants are mammals of the family Elephantidae and the largest existing land animals. Three species are currently recognised: the African bush elephant, the African forest elephant, and the Asian elephant. Elephantidae is the only surviving family of the order Proboscidea; extinct members include the mastodons. The family Elephantidae also contains several now-extinct groups, including the mammoths and straight-tusked elephants. African elephants have larger ears and concave backs, whereas Asian elephants have smaller ears, and convex or level backs. Distinctive features of all elephants include a long trunk, tusks, large ear flaps, massive legs, and tough but sensitive skin. The trunk, also called a proboscis, is used for breathing, bringing food and water to the mouth, and grasping objects. Tusks, which are derived from the incisor teeth, serve both as weapons and as tools for moving objects and digging. The large ear flaps assist in maintaining a constant body temperature as well as in communication. The pillar-like legs carry their great weight.

Figure 5.2: Example article

In this image we can see what format the Joomla articles have.

At the top we see, the name of the Joomla page, Title of the article and author. Below the fulltext photo and, finally, the text.

Now we will see this article in our system.



Super User

A great elephant and a giraffe

Elephant

This is the Readmore

Figure 5.3: Example of Slide 1

We can see the Title, Author, IntroText and the Readmore, in addition to the main image.

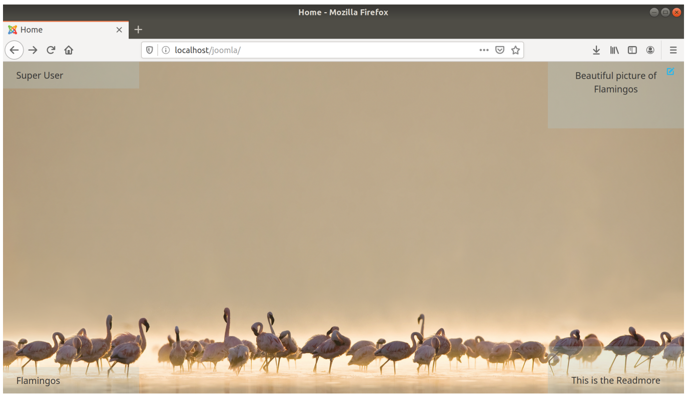
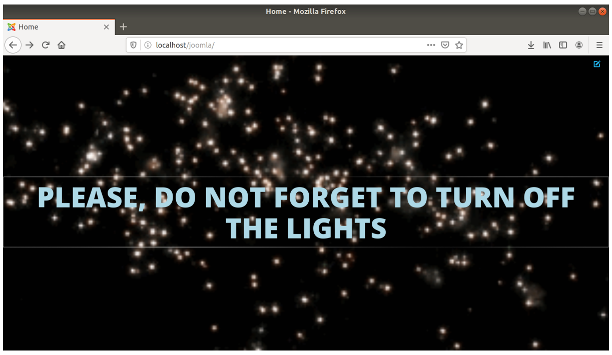


Figure 5.4: Example of Slide 2

These two images are shown with all configuration parameters active, so that the User can see the Title, author, the beginning of the article text and the Readmore added by the Administrator.

In case there is no article, only the last slide will be shown, which is a message configurable by the Administrator. By default it will say ”Don’t forget to turn off the lights, please”.



PLEASE, DO NOT FORGET TO TURN OFF THE LIGHTS

Figure 5.5: Example of Last Slide

5.3 Case Study: Display and view multimedia information

5.3.1 Overview of the Use Case

Since this use case has been less explained above and is more complex than the previous one, we will explain it in detail.

In this use case the actors involved are:

- **Administrator.** In this case the Administration, enters with its credentials of Administrator of the Joomla site. Once there, access the administration part of our module and configure the parameters to your choice.

- **Joomla.** In this case of use, Joomla is responsible for sending the configuration that the Administrator has chosen to the module. Joomla has a menu to access the configuration of the modules and, from there, we can change all the parameters of each one.

- **Articles.** The articles send the information of the new form that this configured the module to our system.

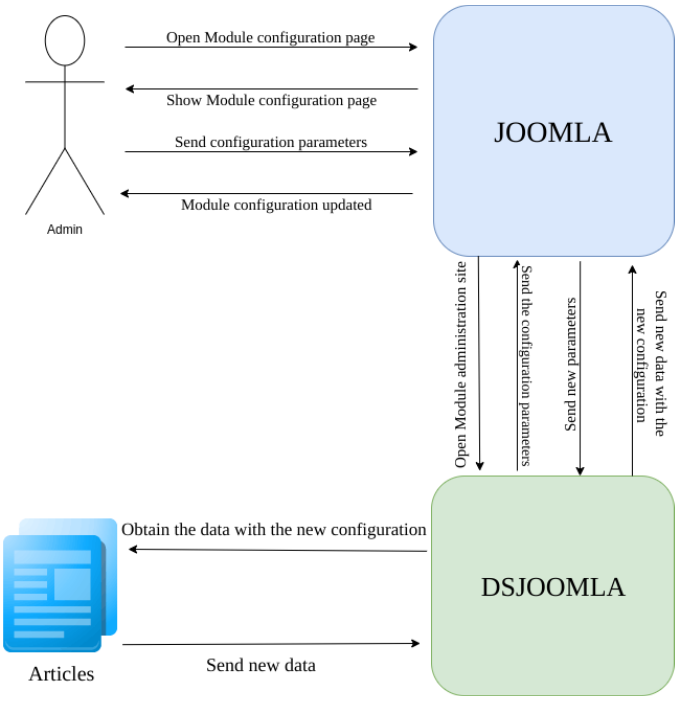

Open Module configuration page
Show Module configuration page
Send configuration parameters
Module configuration updated
Admin
JOOMLA
Open Module administration site
Send the configuration parameters
Send new parameters
Send new data with the new configuration
Obtain the data with the new configuration
DSJOOMLA
Articles
Send new data

Figure 5.6: Manage Module configuration

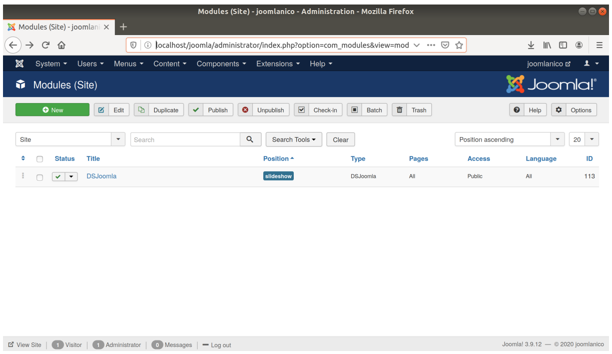
In this case it is not just any user, but it has to be an Administrator who accesses the Administration part of Joomla.

Joomla accesses the module and it shows the last parameters that were introduced.

The administrator can now change the parameters to his liking. Joomla will send them to the module to be updated and this will get the information with the new configuration.

5.3.2 Manage Module Configuration

In this second case of use, the main actor is the administrator. The use case starts when the web administrator accesses the administration part with his credentials.

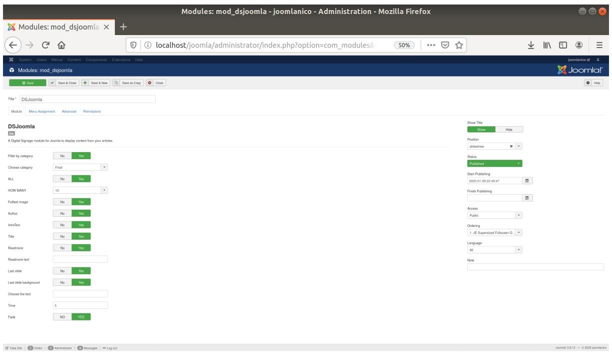


Status	Title	Position	Type	Pages	Access	Language	ID
✓	DSJoomla	slideshow	DSJoomla	All	Public	All	113

Figure 5.7: Administration Site

As we can see, from here we can manage the external part of the module. We can publish it, delete it, duplicate it, see the position in which it is, in which pages it is published, etc.

Clicking now on DSJoomla, the configuration menu will be displayed.



Title * DSJoomla

DSJoomla

Site

A Digital Signage module for Joomla to display content from your articles.

Filter by category: No [✓ Yes]
Choose category: Final
ALL: No [✓ Yes]
HOW MANY: 10
Fulltext image: No [✓ Yes]
Author: No [✓ Yes]
IntroText: No [✓ Yes]
Title: No [✓ Yes]
Readmore: No [✓ Yes]
Readmore text:
Last slide: No [✓ Yes]
Last slide background: No [✓ Yes]
Choose the text:
Time: 5
Fade: NO [✓ YES]

Show Title: [✓ Show] Hide
Position: slideshow
Status: Published
Start Publishing: 2020-01-09 22:49:47
Finish Publishing:
Access: Public
Ordering: 1. JE Supersized Fullscreen G...
Language: All
Note:

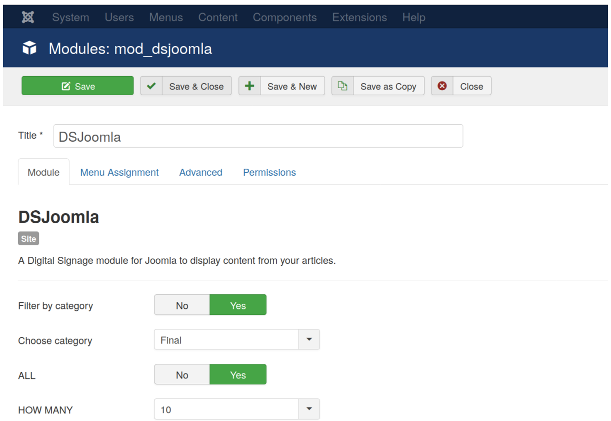
Figure 5.8: Configuration View

In this screen we can differentiate two parts. On the left we find all the configuration parameters of our module. On the right we find common options for all the modules, like publishing them, space in the template, date, language and some more. In this last part we are only interested in publishing and assigning the slot of the template Slideshow.

Now we will focus on the parameters of our own module. Here you will see the parameters that were set last time. In case it's the first time, all of them will be with their default options.

From this menu, all the options that we have configured in the XML file of our System, can be activated, changed or deactivated. Or in case of a text box, add the text that the Administrator chooses.

The first configurable option and one of the most relevant for the user, is the filtering by category. As we have seen, the articles are grouped in categories, so filtering by them, depending on the topic of the talk, for example, is a very interesting option. In addition, it allows you to select if you want all the articles in the category or a limited number of them.



Modules: mod_dsjoomla

Title * DSJoomla

DSJoomla

Site

A Digital Signage module for Joomla to display content from your articles.

Filter by category No [✓ Yes]

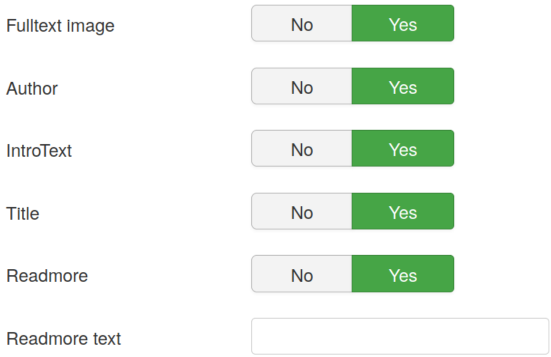
Choose category Final

ALL No [✓ Yes]

HOW MANY 10

Figure 5.9: Category options

The following parameters allow us to configure the content we show. It allows to choose between the Intro Image or the Fulltext Image of the article. It also allows to choose if we want to show the Title, Author and the text intro. Finally we have the Readmore, which is a small text that can be added by the Administrator. The Readmore if empty displays a message that says ”This is the Readmore”.



Fulltext Image No [✓ Yes]

Author No [✓ Yes]

IntroText No [✓ Yes]

Title No [✓ Yes]

Readmore No [✓ Yes]

Readmore text

Figure 5.10: Content options

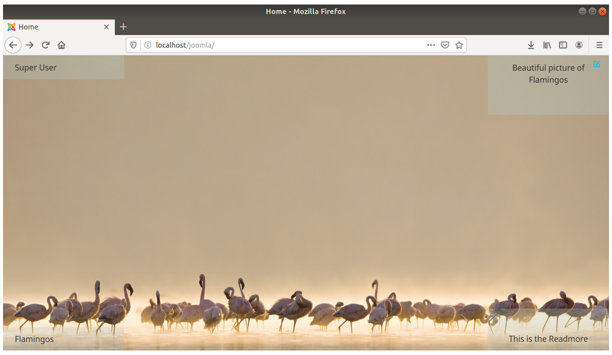


Figure 5.11: Example with all the options active

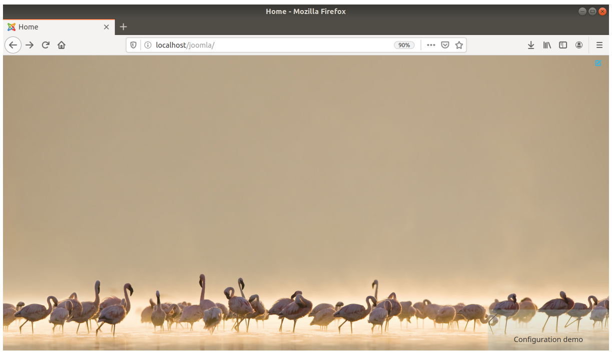
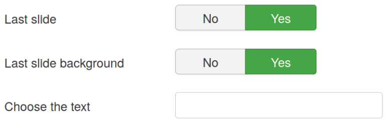


Figure 5.12: Example with only the Readmore activated and changed

Here are the parameters on the last slide. This is a Slide customizable by the Administrator, in order to put a message that will be displayed as a Slide, the last of all. By default the message is "Please, do not forget to turn off the lights", intended for a meeting or conference in a room.

With these options the administrator can activate or deactivate this slide, change the text and add or remove the background.

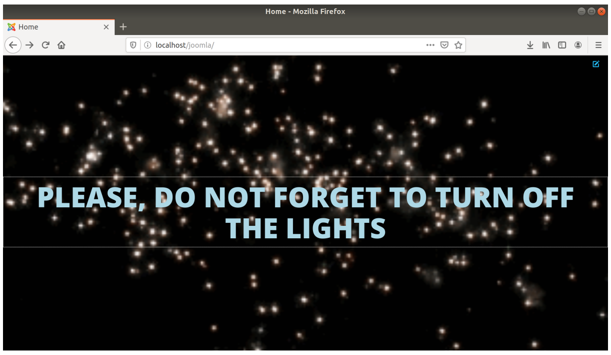


Last slide: No [✓ Yes]

Last slide background: No [✓ Yes]

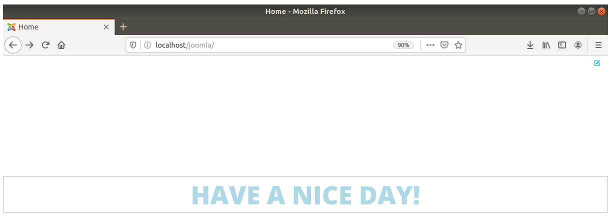
Choose the text:

Figure 5.13: Last Slide configuration



PLEASE, DO NOT FORGET TO TURN OFF THE LIGHTS

Figure 5.14: Last Slide



HAVE A NICE DAY!

Figure 5.15: Last Slide without background and text changed

Finally we find the parameters referred to the style of the transitions. We can choose how often we want a slide change to occur and if we want a fade effect or not.

5.4 Summary

As we have seen despite it seems a simple system, we have created a very robust module with high configurability and great utility for users. Thanks to the study of the Use Case we can understand how the System works in depth.

CHAPTER 6

Conclusions and future work

In this chapter we will see the conclusions obtained after carrying out the work, as well as the objectives fulfilled. Moreover, we will see what future work could be developed.

6.1 Conclusions

Throughout this project, we have developed a Digital Signage extension for Joomla. We will see below what conclusions have been drawn from this work.

First, we did some research on what is Digital Signage, Joomla and CMS. This has allowed us to have a greater knowledge of how to approach the project and know where to go to find information on how to develop an extension. As we saw, Digital Signage is a booming technology. We find it in many parts of our daily life, which makes the project more interesting to carry out, seeing the usefulness and actuality of the subject.

Later, we made an analysis of requirements that our system had to meet. All this has served to advance the project more quickly and effectively. Seeing that could be a priority at the time of development.

Our extension is a very interesting tool since there are not many Digital Signage extensions for Joomla and those that are, are not free. Seeing the success of Wordpress plugins, we can conclude that this extension for Joomla would also be a great success.

To conclude, our module, although it may seem a simple tool, is proven and, as it appears every day in more and more places, is really useful and causes a positive effect on people. As in everything in life, style has a great influence on the perception that you have of things, and a nice and careful style is vital for this extension. Another aspect that has been key is to be intuitive when setting up, to avoid future problems.

6.2 Achieved goals

Now let's go back to the objectives set out above and see if they have been met.

- **Create an extension that displays the contents of a folder.**

  This point was the first to be fulfilled and the beginning of the project. To start getting familiar with the creation of Joomla extensions, we made a test to get the files locally, eliminating the complexity of getting the information from the articles of the department.

- **Obtain the content of the department's articles.**

  The next step was to get the information from the articles. This was done by means of a JSON, being able to show the images that they contained.

- **Display all content in a clean and beautiful style.**

  Style is very important in Digital Signage so it was the next point to take care of. Changing completely the basic style of the first module, to make it visually attractive.

- **Add all administration options to make it configurable**.

  Finally, we had to make the module configurable, adding options to show what and how the administrator wants.

In this way we complete all the points previously established for our system.First we made a study of the most typical options that would be most useful and implemented them, so that the configuration is simple. Despite having fulfilled all the objectives, we have had to face certain problems.

The first and most obvious was the first contact with Joomla, which requires a lot of documentation, but it was easy to overcome. Another problem was the choice of functionalities to implement in the system, because there is a huge range of options and implementing them all was unfeasible. Thanks to the requirements analysis we could optimize the most necessary things and make a first screening of the selected ones. An analysis of the options that offer extensions for other content management systems, helped us to see the most demanded options and finish choosing.

6.3 Future work

- **Add more configuration items.**

  A possible improvement to be developed is to continue adding elements to the administration part of the module so that it has more functionalities. It could even be made to measure so that it is customizable according to who wants to use it

- **Implementing video display.**

  In Joomla 3.x version still do not exist the videos embedded in the articles must be introduced with a link, so when this changes in version 4.x of Joomla, we will have to introduce this functionality.

- **Improve the style.**

  The style can be modified to be optimal for the screen on which it will be displayed. You can even add completely different styles depending on who is going to use it.

As a final conclusion, our system offers many features but can always offer more and will have to be adapted to the demand at that time. Maybe in a few years all the screens that use it become touch screens and you have to modify everything, or do it with a filter for people with colorblindness.

APPENDIX A

Impact of this project

This appendix reflects on the possible impact nowadays.

A.1 Social impact

Digital Signage is a technology increasingly present in our society. We are used to seeing screens with information constantly. We have already seen the positive effect this has on advertising [6], but it also has an effect on our view of places, information and the way we get information. It can turn a conventional meeting room into a much smarter one, with interactive screens, completely changing the way we interact and the aesthetics of the meeting.

The way of getting information is changing completely, in a few years, there will be more and more news channels that we will see walking on the street or advertising while we go in the subway, becoming an everyday element.

A.2 Economical impact

A Digital Signage system is very much appreciated by companies, as it can be enormously beneficial.

We have already seen all the places where this technology is applied, in restaurants to make the menu more attractive, to have kiosks to order, being able to reduce the queues and the personnel, increasing the benefits and the satisfaction of the clients [23].(cite) In the field of advertising it is an optimal tool as well. The effect of dynamic screen advertising on people is much greater than that of paper ads. And even more so if they are interactive [2].

Therefore, economically speaking, digital signage is a technology that brings huge benefits to businesses. The cost of companies that are dedicated exclusively to this is moderate but the investment will be worth it, in most cases.

A.3 Environmental impact

In the environmental aspect, being a not especially polluting technology, it is worth highlighting some aspects.

The electricity needed to maintain the screens and the computers that manage them is obviously greater than that of having a paper poster. But in turn, this reduces the number of trees felled to obtain the paper needed for advertising or menus.

On the other hand, there is a critical area with the screens because of the damage that can be done to the view all day long by looking at the screens, even when one eats in a restaurant, walks on the street or is shopping. Although there is no study of the damage that digital signage can do to the screens, the screens are becoming more and more safe to look at, so this does not present a problem.

APPENDIX B

Economic budget

This appendix details an adequate budget for the project.

B.1 Physical resources

This project was carried out on a computer with an i7 processor 8th generation, 16 GB of RAM, a 100 GB SSD hard disk and an NVIDIA GeForce GTX 1050 Ti graphic card. Approximately, the cost of this laptop is 1600€.

On the software side, everything used has been open-source, so the cost is zero.

B.2 Human resources

This project has taken approximately 320 hours. If we estimate an average of 12 euros per hour for an undergraduate telecommunication engineer, it results 3840 euros.

Bibliography

[1] Isabel María del Águila Cano. *Ingeniería de requisitos: Material didáctico. Cuaderno de teoría*. Vol. 35. Universidad Almería, 2019.

[2] Florian Alt, Jorg Muller, and Albrecht Schmidt. "Advertising on public display networks". In: *Computer* 45.5 (2012), pp. 50–56.

[3] Deane Barker. *Web content management: systems, features, and best practices*. O'Reilly Media, Inc., 2016.

[4] Jonathan Blackwood. "University of Minnesota Now Has 275 Digital Signage Displays with Carousel 7.0". In: *Commercial Integrator* (2017).

[5] Tim Bray et al. *Extensible markup language (XML) 1.0*. 2000.

[6] Raymond R Burke. "Behavioral effects of digital signage". In: *Journal of Advertising Research* 49.2 (2009), pp. 180–185.

[7] Command Center. *CommandCenterHD Reviews*. URL: `https://www.capterra.com/p/149315/CommandCenterHD/` (visited on 10/20/2019).

[8] Ángel Cobo. *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos, 2005.

[9] Charles Dennis et al. "The mediating effects of perception and emotion: Digital signage in mall atmospherics". In: *Journal of Retailing and Consumer services* 17.3 (2010), pp. 205–215.

[10] Drupal. *Drupal Website*. 2001. URL: `https://www.drupal.org/` (visited on 10/19/2019).

[11] EnagagePHD. *Intuiface website*. URL: `https://www.engagephd.com/` (visited on 10/20/2019).

[12] Enplug. *Enplug website*. URL: `https://www.enplug.com/` (visited on 10/20/2019).

[13] ESA. *System Requirements Document*. URL: `http://www.esa-sealevel-cci.org/webfm_send/99` (visited on 11/03/2019).

[14] David Flanagan. *JavaScript: the definitive guide*. " O'Reilly Media, Inc.", 2006.

[15] Farnaz Fotrousi et al. “Kpis for software ecosystems: A systematic mapping study”. In: *International Conference of Software Business*. Springer. 2014, pp. 194–211.

[16] Foyer. *Foyer Website*. URL: `https://foyer.tv/` (visited on 10/20/2019).

[17] Yanette Díaz González and Yenisleidy Fernández Romero. “Patrón Modelo-Vista-Controlador.” In: *Revista Telemática* 11.1 (2012), pp. 47–57.

[18] John V Harrison and Anna Andrusiewicz. “Enhancing digital advertising using dynamically configurable multimedia”. In: *2003 International Conference on Multimedia and Expo. ICME’03. Proceedings (Cat. No. 03TH8698)*. Vol. 1. IEEE. 2003, pp. I–717.

[19] Intuiface. *Intuiface website*. URL: `https://www.intuiface.com/` (visited on 10/20/2019).

[20] Joomla. *Joomla documentation*. URL: `https://docs.joomla.org/Extension` (visited on 10/01/2019).

[21] Joomla. *Joomla extensions documentation*. URL: `https://docs.joomla.org/J3.x:Developing_an_MVC_Component/Developing_a_Basic_Component/en` (visited on 10/05/2019).

[22] Joomla. *Joomla Module documentation*. URL: `https://docs.joomla.org/J3.x:Creating_a_simple_module/Developing_a_Basic_Module/en` (visited on 10/05/2019).

[23] Jungsun Kim, Natasa Christodoulidou, and Yunjeong Choo. “Factors influencing customer acceptance of kiosks at quick service restaurants”. In: *Journal of Hospitality and Tourism Technology* 4.1 (2013), pp. 40–63.

[24] Craig Larman. *UML y Patrones*. Pearson Educación ˆ eMadrid Madrid, 2003.

[25] Red Gráfica Latinoamérica. *El lenguaje de programación PHP*. 2010.

[26] Susan McKeever. “Understanding Web content management systems: evolution, lifecycle and market”. In: *Industrial management & data systems* 103.9 (2003), pp. 686–692.

[27] Eric A Meyer. *Cascading style sheets: The definitive guide.* ” O’Reilly Media, Inc.”, 2004.

[28] Chuck Musciano, Bill Kennedy, et al. *HTML, the definitive Guide*. O’Reilly & Associates, 1996.

[29] Savan K Patel, VR Rathod, and Jigna B Prajapati. “Performance analysis of content management systems-joomla, drupal and wordpress”. In: *International Journal of Computer Applications* 21.4 (2011), pp. 39–43.

[30] Anicia Peters and Brian Mennecke. "Can Digital Signage Help Consumers Eat Healthier?" In: *International Conference on Human-Computer Interaction.* Springer. 2011, pp. 443–447.

[31] Dan Rahmel. *Beginning Joomla!: From novice to professional.* Apress, 2008.

[32] Jimmy Schaeffler. "What Is Digital Signage?" In: *Digital Signage.* Routledge, 2012, pp. 61–92.

[33] Jimmy Schaeffler. "What Is Digital Signage?" In: *Digital Signage.* Routledge, 2012, pp. 61–92.

[34] Charles Severance. "Discovering javascript object notation". In: *Computer* 45.4 (2012), pp. 6–8.

[35] Ric Shreves. *Joomla! bible.* Wiley Online Library, 2013.

[36] Wordpress. *Foyer in Wordpress.* URL: `https://wordpress.org/plugins/foyer/` (visited on 10/20/2019).

[37] Wordpress. *Wordpress Website.* URL: `https://es.wordpress.com/` (visited on 10/19/2019).