

Université Saad Dahleb de Blida



Faculté de Sciences

Département informatique

mémoire présenté par :

DJEBLI Mohamed

LAOUER Youcef

En vue d'obtenir le diplôme de MASTER

Domaine : Mathématique et Informatique

Filière : Mathématique et Informatique

Spécialité: Informatique

Option : Ingénierie des Logiciels

Sujet:

Développement de composants pour donner de la valeur ajoutée à Joomla

Soutenue le : /0../2013, devant le jury composé de :

M.	Président
M.	Examineur
M.	Examineur
M. Cherif Zahar Amine	Promoteur

Promotion : 2012/2013

Remerciements

En tout premier lieu, nous remercions Allah le tout puissant, à la sagesse et au savoir infinis, Gloire à Toi! Nous n'avons de savoir que ce que Tu nous as appris. Certes c'est Toi l'Omniscient, le Sage.

(Sourate al-Baqarah, verset 32).

Nous devons une reconnaissance particulière à notre promoteur, Mr. Cherif Zahar Amine, pour nous avoir proposé ce sujet intéressant et d'actualité, et pour ces nombreux conseils, encouragements, relectures et corrections.

Nous devons aussi une reconnaissance à tous nos enseignants à l'USDB.

Enfin, nous n'oublions pas de remercier nos amis d'études et tous ceux, qui ont contribué, de près ou de loin à notre formation et à l'aboutissement de ce projet.

Dédicaces

A tous nos parents,

Nos frères et sœurs,

Et tous nos amis

Résumé

Avec l'évolution du domaine web, une nouvelle notion est apparue celle des systèmes de gestion de contenu (SGC) qui sont des logiciels permettant d'administrer des sites web que ce soit sur internet ou sur un réseau local. Un SGC vous permettra de créer un site internet dynamique en toute simplicité, parmi les SGC nous trouvons Joomla qui est open source avec lequel vous pourrez mettre en ligne des contenus et mettre à la disposition de vos visiteurs des services, le tout sans connaissance technique particulière.

Notre travail consiste à concevoir et à réaliser des extensions qui donnent de la valeur au système Joomla afin de résoudre certaines problématiques, parmi lesquels nous avons choisi de résoudre le problème relatif à l'affichage aléatoire des articles, ainsi que la création d'une extension de statistique dédiée à rendre le processus de décisions plus simple à gérer. Cela nous a entraînés dans la nécessité d'étudier l'outil Joomla et ses fonctionnalités ainsi son architecture.

Durant le processus de développement de notre projet nous avons constaté que la quantité importante des données à traiter peut compliquer certaines fonctionnalités. C'est pour cela, avoir une bonne architecture est important afin de limiter ces problèmes, d'où la nécessité d'une architecture de développement spécifique dont le Modèle-Vue-Contrôleur (MVC) en est l'exemple. Nous avons opté pour une architecture MVC qui nous a permis de séparer chaque rôle de nos applications, de programmer à plusieurs (les règles étant communes) et de faire évoluer le code en toute simplicité, ceci nous a permis aussi de garder en tête le cheminement des requêtes, de travailler avec des diagrammes, de simplifier indirectement la gestion de notre projet qui nous a permis d'atteindre nos objectifs.

Mots clés : SGC, Joomla, extensions, aléatoire, statistiques, MVC.

Abstract

With the development of web domain a new concept appeared, a content management systems (CMS) are software used to manage websites either on the Internet or a local network. A CMS allows you to create a dynamic website with ease among, between the CMS we found Joomla that is open source that you can get online content and make it available services to your visitors, without any technical knowledge.

Our job is to design and implement extensions that gives Joomla a value to solve some problems among them we chose to solve the problem according to the random display of items, and to create a statistical extension dedicated to make the process of decisions so easier to manage. This has led us to study Joomla and these features and architecture tool.

while the development of this project process we found that the large amount of data to be processed can complicate things, this is why having a good architecture is important to minimize these problems, therefore a need to solve these problems based on the MVC model that allowed us to separate each part of our applications, program several (the rules are common) and to easily change the code, this also allows us to keep in mind the routing of requests, working with charts, indirectly simplify the management of our project that ended us to achieve our goals.

Keywords: CMS, Joomla, extensions, MVC, random, statistics.

ملخص:

مع تطور مجال شبكة الإنترنت و ظهور مفهوم جديد " أنظمة إدارة المحتوى". (ن.إ.م) هي عبارة عن برمجيات مستخدمة لإدارة المواقع إما على الإنترنت أو الشبكة المحلية. (ن.إ.م) تسمح لك بإنشاء موقع ديناميكي وبكل سهولة, من بين ن.إ.م نجد جملة مفتوحة المصدر التي تمكنك من نشر المحتوى عبر الإنترنت وجعله متاح الخدمة للزائرين, كل هذا دون أي معرفة تقنية.

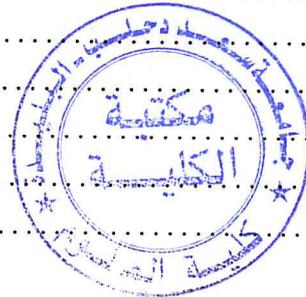
مهمتنا هي تصميم وتنفيذ تمديدات تعطي قيمة إلى جملة من اجل حل بعض المشاكل من بينهم اخترنا حل مشكلة العرض العشوائي للمقالة وخلق امتدادا للإحصائيات مكرسة لجعل عملية إدارة القرارات أسهل. وقد أدى هذا بنا لدراسة جملة و ميزاتنا وأداة هندستها.

و من خلال عملية تطوير هذا المشروع تبين أن الكمية الكبيرة من البيانات التي سيتم تجهيزها يمكنها تعقيد الأمور ولهذا من المهم وجود بنية جيدة للحد من هذه المشاكل، وبالتالي ضرورة حل هاته المشاكل على أساس نموذج عرض وحدة التحكم التي سمحت لنا بفصل كل جزء من طلباتنا، البرمجة المتعددة (قواعد شائعة) وبسهولة تطوير تعليمات البرمجة وهذا سمح لنا أيضا أن نأخذ بعين الاعتبار توجيه الطلبات والعمل مع الرسوم البيانية، مع تبسيط إدارة مشروعنا بشكل غير مباشر مما أنهى الأمر بنا إلى الوصول إلى أهدافنا.

مفاتيح الكلمات: ن.إ.م, جملة, التمديدات, العشوائي, إحصائية, نموذج عرض وحدة التحكم

Sommaire

Introduction général	1
Contexte général.....	1
Problématique.....	3
Objectifs.....	3
Organisation du mémoire.....	6
Partiel : Etat de l'art sur les systèmes de gestion de contenu CMS	7
Chapitre 1 : CMS et JOOMLA	7
1 Introduction.....	7
2 Et un CMS, c'est quoi ?.....	8
2.1 Principales caractéristiques.....	9
3 Qu'est-ce que Joomla?.....	9
3.1 Avantages de Joomla 2.5.....	11
4 Conclusion.....	11
Chapitre 2 : Étude du système Joomla	15
1 Introduction.....	15
2 Principes de fonctionnement.....	15
3 Architecture Joomla.....	18
3.1 Les extensions dans Joomla.....	18
3.2 Application.....	19
3.2.1 JInstallation.....	20
3.2.2 Back-end.....	20
3.2.3 Font-end.....	21
3.3 FRAMWORK.....	22
3.3.1 FRARWOMK JOOMLA.....	22
3.3.2 PLUGINS.....	23
4 Conclusion.....	24



Partie II : Mise en œuvre du système	26
Chapitre 3 : Le modèle MVC	26
1 Introduction au domaine MVC	26
2 Avantages et inconvénients	27
3 Principe du modèle MVC	28
4 Utiliser l'architecture MVC dans les applications Web/PHP	30
5 Conclusion	34
Chapitre 4 : Analyse et conception	35
1 Introduction	35
2 Conception uml d'une application	35
3 Analyse	36
3.1 Diagramme de classe	39
3.2 Diagrammes des cas d'utilisation	40
3.3 Diagrammes de séquence	43
3.4 Diagramme d'états transition	45
4 Conception	47
4.1 Diagramme de classe	47
4.2 Diagrammes de séquence	48
4.3 Diagramme d'états transition	54
5 Conclusion	55
Chapitre 5 : Implémentation d'application	56
1 Introduction	56
2 Présentation des outils de développement	56
2.1 Le langage PHP	56
2.2 Description du serveur de base des données MySQL	57
2.3 Description du serveur Web	57
3 Fonctionnement de composants sous Joomla	58
3.1 Fonctionnement de composant d'aléatoire	58
3.2 Fonctionnement de composant de statistiques	60
3.2.1 Description des méthodes utilisées dans le composant de statistiques	61
4 Les interfaces du composant	62
4.1 Composant d'aléatoire	62
4.2 Composant de statistiques	67
5 Conclusion	71

Conclusion générale.....	72
Sources	73
Références webographie.....	73
Programmes utiles.....	75
Glossaire.....	76

Table de figures

Figure 1 : Exemple d'interface d'administration Joomla 2.5.....	10
Figure 2 : Fonctionnement de pages web en PHP [5].....	12
Figure 3 : Les droits d'accès dans Joomla [6].....	13
Figure 4 : Les templates dans Joomla [7].....	15
Figure 5 : Interface de la gestion des extensions dans Joomla.....	15
Figure 6 : Les différentes couches dans Joomla [8].....	16
Figure 7 : Les Templates dans Joomla [9].....	18
Figure 8 : Interface installation Joomla [10].....	19
Figure 9 : Backend de Joomla [10].....	20
Figure 10 : Exemple d'un Frontend Joomla [11].....	21
Figure 11 : Schéma du modèle MVC [4].....	28
Figure 12 : Diagramme de séquence du modèle MVC [4].....	28
Figure 13 : Fonctionnement du MVC dans le web [14].....	31
Figure 14 : Fonctionnement de MVC [14].....	32
Figure 15 : Schéma Joomla 2.5 [15].....	37
Figure 16 : Diagramme de classe - Analyse -.....	38
Figure 17 : Diagramme de cas d'utilisation globale.....	40
Figure 18 : Premier niveau d'authentification.....	41
Figure 19 : Premier niveau de gestion de site.....	41
Figure 20 : Second niveau de gestion de site.....	42
Figure 21 : Second niveau de gestion de contenu.....	42
Figure 22 : Diagramme de séquence - Identification -.....	43
Figure 23 : Diagramme de séquence - installation des extensions -.....	44
Figure 24 : Diagramme états transition - cas d'articles -.....	45
Figure 25 : Diagramme de classe- après conception -.....	46
Figure 26 : Diagramme de séquence - installation des extensions -.....	48
Figure 27 : Diagramme de séquence - manipulation composant statistique -..	49
Figure 28 : Diagramme de séquence - manipulation composant aléatoire -.....	50
Figure 29 : Diagramme de séquence création de nouvelle configuration d'aléatoire.....	51
Figure 30 : Diagramme de séquence modification d'une configuration d'aléatoire.....	52

Figure 31 : Diagramme de séquence - ajouter un lien d'un menu -	53
Figure 32 : Diagramme d'états transition- configurations d'aléatoire -	54
Figure 33 : Traitement d'une requête d'un internaute avec le modèle MVC [17].....	58
Figure 34 : Fonctionnement de composant d'aléatoire.	59
Figure 35 : Fonctionnement de composant de statistiques.	60
Figure 36 : Installation du composant aléatoire.	62
Figure 37 : Guide d'utilisation du comopsant aléatoire.	63
Figure 38 : Configuration d'aléatoire.	63
Figure 39 : Nouvelle configuration d'aléatoire.	64
Figure 40 : Modification d'une configuration d'aléatoire.	64
Figure 41 : Nouveau lien vers composant d'aléatoire.	65
Figure 42 : Test de la notion d'aléatoire sur frontend.	66
Figure 43 : Sous menu du composant des statistiques.	67
Figure 44 : Statistiques des articles(stat-articles).....	68
Figure 45 : Statistiques des utilisateurs enregistrés (stat-enregistred-users)....	68
Figure 46 : Statistiques des utilisateurs (stat-fresh-users).....	69
Figure 47 : Choisir un type de lien vers composant des statistiques.	70
Figure 48 : Statistiques globaux sur frontend atteint à partir d'un lien d'un menu.....	71

Liste de tableaux

Tableau 1 : Paquets Joomla[12].....	23
Tableau 2 : Plugins de base de Joomla[12].....	24
Tableau 3 : Logique Applicative[14].....	33

Liste des abréviations

CMS	<i>Content Management System</i>
MVC	<i>Model View Controller</i>
IP	<i>Internet protocol</i>
PHP	<i>Hypertext Preprocessor</i>
HTML	<i>Hypertext Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
SGBD	<i>Système de gestion de base de données</i>
FTP	<i>File Transfer Protocol</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
URI	<i>Uniform Resource Identifier</i>
WYSIWYG	<i>What You See Is What You Get</i>
RPC	<i>Remote procedure call</i>
XML	<i>Extensible Markup Language</i>
XSLT	<i>Extensible Stylesheet Language Transformations</i>
XSL	<i>Extensible Stylesheet Language</i>
PDF	<i>Portable Document Format</i>
AJAX	<i>Asynchronous JavaScript and XML</i>
HTTP	<i>Hypertext Transfer Protocol</i>

Contexte général

Avec la progression rapide du domaine informatique le monde subit des changements de vues et de besoins, l'évolution sur l'échelle web permettait d'une part de faciliter le processus d'échange des documents et des fichiers à travers les réseaux, mais d'autre part cela a entraîné l'apparition d'un problème de facilité la gestion de ces documents tout en préservant la notion de la rapidité du service. D'où la naissance des systèmes de gestion de contenu (SGC) pour content management system (CMS) en anglais.

Les SGC sont des outils de gestion dynamique de site web et d'application multimédia de manière interactive à l'aide de divers modules. Ils permettent à plusieurs personnes selon des droits prédéfinis de gérer leur site web en local ou à distance, en leur fournissant une chaîne de publication (workflux) pour la mise en ligne, la mise à jour, la suppression, tout cela leur assurant une indépendance entre les opérations de gestion de la forme et de contenu.

De nombreuses CMS existent sur le marché, nous trouvons parmi eux Joomla qui reflète un outil populaire et gratuit de gestion de contenu dès le départ orienté utilisateur dans le but d'assurer l'extensibilité et la compatibilité des produits. Joomla permet de concevoir des extensions sous forme de packages sans avoir besoin de modifier le code source de l'application, afin de personnaliser le CMS. Il suffit de créer un composant, un plugin ou un module selon l'architecture MVC et l'installer, cela implique une grande flexibilité et une capacité de mise à jour évolutive. Mais, nous nous trouvons des fois devant un cas où l'extensibilité est coûteuse, cela nous entraîne dans l'essai de concevoir et de réaliser nos propres extensions selon nos besoins.

Durant la majorité des recherches concernant des travaux réalisés et basés sur le CMS, nous constatons que lorsque nous travaillons sur un gros projet tel qu'une application web, la quantité importante des données à traiter peut compliquer certaines fonctionnalités, C'est pourquoi avoir une bonne architecture est primordial pour limiter ces problèmes et c'est là que le modèle MVC entre en scène, cette notion de Modèle Vue Contrôleur est un modèle d'architecture qui sépare les données,

les traitements et la vue (présentation) de l'application, pour une plus grande clarté structurelle.

Nous trouvons intéressant de travailler sur ce domaine web et être à jour afin de suivre son évolution. De même, avoir une vue sur ces systèmes de gestion de contenu ainsi que la possibilité d'appliquer nos études de conception sur des exemples réels des sites web réussis, et de ces certains critères. Ces systèmes nous offrent une aide précieuse pour le développement et la mise en oeuvre des applications web afin de satisfaire les besoins.

Problématique

Les articles constituent l'essentiel du contenu d'un site Joomla. Ils représentent également un bon exemple de fonctionnement global de Joomla en ce qui concerne les listes, la manipulation d'éléments (copie, suppression, droits d'accès, ..ect). Actuellement dans la version Joomla 2.5, l'affichage des articles d'un blog ne dispose que de deux possibilités. Affichage du plus récent et l'affichage des articles les plus lus en premier, l'un et l'autre sont mauvais. Le premier, dans le cas de site qu'on ne met pas à jour fréquemment, pour le visiteur qui y vient régulièrement prend l'aspect d'un site statique. Deuxième cas, les articles qui s'affichent les premiers ont une audience croissante et les derniers arrivés risquent de n'être jamais suivis.

En plus, à chaque fois qu'un utilisateur est entrain de palier le site il remarquera qu'il existe une sorte de standardisation relative aux manières d'exposition de contenu de la page d'accueil, cela peut mal influencer sur le comportement ainsi sur la partie éducative, car cette standardisation peut causer une sorte de routine si cette personne est considérée comme un visiteur régulier.

Objectifs

- i. Les articles dans Joomla représentent toute la partie rédactionnelle du site, elle est donc très importante. Certaines extensions aident à personnaliser ces articles mais quelques fois il est difficile d'en trouver un qui satisfait nos besoins, cela nous amène à résoudre ce problème en réalisant une extension Joomla qui donnera un nouvel aspect d'affichage des articles relatifs à une catégorie, donc utiliser la notion d'aléatoire afin de permettre aux visiteurs de parcourir différents articles en temps limité, et donner la possibilité au administrateur de créer et de modifier les paramètres de cette notion, comme : le nom de la catégorie ainsi le temps de rafraichissement de la page contenant cet article aléatoire pour afficher un autre afin de parcourir un maximum d'articles en peu de temps.
- ii. Quel que soit le type de notre site : référencement, publicité,..ect. nous devons le maîtriser afin de le mettre en valeur. Mais afin d'optimiser un maximum de nos efforts, pourquoi ne pas faire des statistiques ? Autrefois, le suivi

des statistiques était un parcours de combattant. Les plateformes n'offraient que peu d'outils et on effectuait le référencement à l'aveuglette, donc il nous faut l'utilisation des outils qui analysent nos visiteurs afin de savoir si notre site est visité ou non (popularité) afin de l'améliorer. Les sites d'outils de statistiques sont là pour nous aider à connaître la fréquentation de notre site avec notamment le nombre de visiteurs, le nombre de pages vues,..ect.

Cela nous ramène à un besoin de réaliser un outil (extension Joomla) qui rend le contrôle facile a géré et qui permet de prendre une vue globale pour aider l'administrateur à déterminer les différentes décisions. Une stratégie couronnée de succès passe inévitablement par une bonne compréhension de notre public. Mieux nous aurons réussi à cerner nos spectateurs, plus il sera simple de créer un contenu adapté qui saura le captiver. Cette extension doit assurer ces différents points :

- a. **Statistiques relatives aux visiteurs** : nous donneront une idée sur notre popularité selon la thématique. Nous devons déterminer le nombre de visiteurs ainsi que leurs informations relatives
 - Type de visiteur (membre inscrit ou bien un visiteur non inscrit)
 - Son adresse IP
 - Son pays
 - Son type de système d'exploitation
 - Son type de navigateur
 - Sa date de visite

Chaque site possède une limite de visiteurs possibles et si nos chiffres sont inférieurs à la normale, alors nous devons augmenter notre référencement et c'est là que notre véritable travail commence.

- b. **Statistiques des articles** : donnent la possibilité à l'administrateur de poursuivre l'état des articles :
 - *Informations sur leur créateurs* : puisqu'en Joomla nous avons la possibilité d'avoir plusieurs éditeurs.

- *Informations propres à l'article* : son identifiant, son titre, sa catégorie, sa première date de mise en ligne.
- *Informations liées à la publication* : le nombre de fois où l'article à été consulté.

La plupart du temps, si nous découvrons que seules quelques pages sont populaires, nous devons revoir l'état des autres pages et modifier l'architecture en amont.

c. **Informations relatives aux membres inscrits** : permettent à l'admin d'avoir une idée à propos des membres inscrits

- Son nom et son pseudo
- Son email
- Sa date d'enregistrement
- Sa dernière date de consultation

Organisation du mémoire

Notre mémoire est organisé autour de deux parties:

La 1^{ère} partie, reflète une vue globale sur le domaine des systèmes CMS. Cette partie est constituée de deux chapitres :

Dans le 1^{er} chapitre, nous exposons les généralités sur les CMS ainsi que leurs fonctionnalités et caractéristiques ensuite en nous limitons sur un exemple qui est Joomla.

Le second chapitre sert à étudier le système Joomla en expliquant ses principes de fonctionnement en se basant sur son architecture et ses différentes parties.

Dans la 2^{ème} partie, nous allons essentiellement nous intéresser à la mise en œuvre du système afin d'atteindre nos objectifs. Cette partie est composée de trois chapitres :

Le 3^{ème} chapitre introduit la notion d' MVC, tout en se basant sur ses principes de fonctionnement afin de faciliter sa compréhension ainsi que sa manipulation pour obtenir une bonne architecture.

Le 4^{ème} chapitre concerne l'étude détaillée de système en interprétant des différents diagrammes pour mieux décrire les différentes fonctionnalités. Ce chapitre se décompose en deux niveaux : au premier niveau nous trouvons l'analyse qui exprime l'état actuel du système, au deuxième niveau nous commençons notre conception en rajoutant les nouvelles fonctionnalités résolvant nos problèmes traités.

Vous trouverez dans le 5^{ème} et dernier chapitre, une présentation d'implémentation de notre système.

Partie I :

Etat de l'art sur les systèmes de gestion de contenu CMS

Chapitre 1

Cms

et

joomla

1 Introduction

Quand vous travaillez dans une petite structure, voire seul, vous combinez bien souvent plusieurs métiers : celui dans lequel vous êtes spécialisé, pour lequel vous offrez des services et auquel vous aimeriez principalement vous consacrer, mais vous jouez aussi le rôle de comptable, secrétaire, standardiste... ou encore d'informaticien.

Par exemple vous souhaitez construire une nouvelle maison mais vous ne savez trop comment vous y prendre. Vous n'avez pas de connaissances en gros œuvre, en électricité ou encore en décoration, mais vous en avez tellement envie de cette nouvelle maison ! Vous pourriez tout apprendre vous-même ! Vous avez donc pris contact avec plusieurs maîtres d'œuvre et l'un d'eux a particulièrement retenu votre attention: il s'occupe du gros œuvre et vous livre une maison modulable où vous pourrez choisir vous-même l'emplacement des cloisons (pour faire autant de pièces que vous voulez), et la décoration. Il ne vous reste plus qu'à meubler. C'est là qu'intervient Joomla!, qui vous sera précieux pour créer votre indispensable présence, et pour utiliser ce logiciel libre, il suffit de disposer d'un ordinateur connecté à internet , La maison c'est votre site web, le maître d'œuvre c'est Joomla , le gros œuvre c'est l'environnement de travail PHP/MySQL, les cloisons c'est précisément la modularité de Joomla (qui vous permettra notamment d'ajouter des composants et des modules à la structure de site), la décoration c'est le Template (le design de votre site), quant aux meubles, vous l'aurez deviné, il s'agit du contenu même de votre site[1].

Sa prise en main et son apprentissage sont évolutifs. Vous pouvez en effet le découvrir progressivement et déjà créer les premières pages de votre site grâce aux fonctionnalités élémentaires du système. Par la suite, en fonction du temps dont vous disposez, de votre envie d'améliorer votre espace sur le Web et de vos besoins, vous pouvez découvrir ses multiples possibilités, des plus simples au plus sophistiquées. De plus, c'est un logiciel qui permet aussi bien de travailler seul que de collaborer avec une ou plusieurs autres personnes grâce aux différents droits d'accès à l'espace de travail. Il est destiné spécialement

à ceux qui souhaitent avoir un site internet bien à eux mais qui n'ont pas l'opportunité de confier la création à un professionnel ou qui ont simplement très envie de mettre la main à la pâte, il est accessible à tout le monde et il existe de bons outils pour faire en sorte que les sites internet y soient bien conçus.

C'est le cas de Joomla!, bien entendu, qui est accessible à tout le monde, et ne requiert pas de connaissance préalable en informatique.

2 Et un CMS, c'est quoi ?

Un système de gestion de contenu (CMS) est un programme informatique qui permet la publication, l'édition et la modification de contenu ainsi que la maintenance d'une interface centrale. Ces systèmes de gestion de contenu fournissent des procédures pour gérer les flux de travail dans un environnement collaboratif. Ces procédures peuvent être des étapes manuelles ou une cascade automatisée. Le premier système de gestion de contenu (CMS) a été annoncé à la fin des années 1990, Ce CMS a été conçu pour simplifier la tâche complexe de l'écriture de nombreuses versions de code et de rendre le processus de développement du site Web plus flexible. En plus les plates-formes CMS permettent aux utilisateurs de centraliser les données d'édition, de publication et de modification sur une interface back-end unique et souvent les plates-formes CMS sont utilisés comme logiciel de blog.

En effet, un CMS est un logiciel web qui permet de créer un site internet dynamique en toute simplicité, sans connaissances techniques particulières, l'idée étant de séparer le volet technique du logiciel de la forme et du contenu qui seront eux la partie visible, votre site tel qu'il sera vu par vos futurs visiteurs. Bon ce n'est toujours pas clair ? Prenons une bagnole, n'importe qui est capable de la conduire sans pour autant être mécanicien non ? CQFD. S'il est d'abord orienté gestion de contenu, c'est-à-dire destiné à mettre en ligne des articles et des news, CMS web peut en quelques clics être " détourné " en portail, site de boutique en ligne, site communautaire, site personnel, ect., ce grâce aux centaines d'extensions disponibles sur le web [1].

Cet outil permet une gestion facile et complète des rubriques, images.. depuis votre site.

2.1 Principales caractéristiques

La fonction de base et l'utilisation des systèmes de gestion de contenu est de présenter des informations sur les sites Web. Les fonctionnalités CMS varient considérablement d'un système à l'autre. Des systèmes simples présentent une poignée de fonctionnalités, tandis que les autres versions, notamment les systèmes d'entreprise, offrent des fonctions plus complexes et puissantes. La plupart des CMS comprennent l'édition sur le Web, la gestion de format, contrôle de révision (contrôle de version), l'indexation, la recherche et la récupération. L'incrément CMS est le numéro de version lorsque de nouvelles mise à jour seront ajoutées à un fichier déjà existant. Un CMS peut servir de référentiel central contenant des documents, films, photos, numéros de téléphone, des données scientifiques, CMS peuvent être utilisés pour le stockage, le contrôle, la révision, sémantiquement enrichissement et la publication des documents.

3 Qu'est ce que Joomla?

Joomla est un système de gestion de contenu libre, open source et gratuit, il est écrit en PHP et utilise une base de données MySQL, Joomla inclut des fonctionnalités telles que les news, une version imprimable des pages, des blogs, des sondages, des recherches.

Joomla, déformation du mot swahili Jumla, veut dire : tous ensemble, en bloc ou en tout, en moré restons unis. Il est créé à partir du CMS Open Source Mambo en août 2005, suite aux désaccords d'une majorité des développeurs open source avec la société propriétaire du nom de Mambo. Celle-ci, pour accroître sa notoriété, voulait utiliser ce nom pour le CMS propriétaire qu'elle développait en parallèle. La très grande majorité des utilisateurs a rapidement suivi le mouvement [3].

Joomla C'est un moyen fantastique pour créer un site internet sans toucher de code. De base, Joomla fait partie des systèmes de gestion de contenu, il vous permet de créer, d'organiser du contenu et d'étendre les fonctionnalités de votre site avec des fonctions déjà intégrées. Joomla vous permettra de gérer votre site internet sans avoir besoin de connaissances de programmation particulières. Le contenu des pages est stocké dans une base de données et non pas dans des pages HTML, si vous avez l'habitude de travailler avec des outils comme dreamweaver, vous devez d'abord créer vos pages sur votre ordinateur et ensuite les envoyer sur le serveur qui héberge votre site. Avec Joomla vous avez accès à une partie administration de votre site qui vous permet de créer vos pages et ensuite de les sauvegarder dans votre base de données, en plus de vous permettre d'ajouter les pages Web, il vous offre la possibilité d'installer des modules et composants par exemple pour gérer vos albums photos ou mettre en place un magasin en ligne. L'installation de ces fonctionnalités supplémentaire est très simple car il vous suffit de télécharger sur votre poste les fichiers concernés puis de les installer depuis l'administration de votre site. Plus qu'il est gratuit ainsi qu'il est maintenu à jour par une communauté de développeurs ce qui assure sa progression rapide d'amélioration.

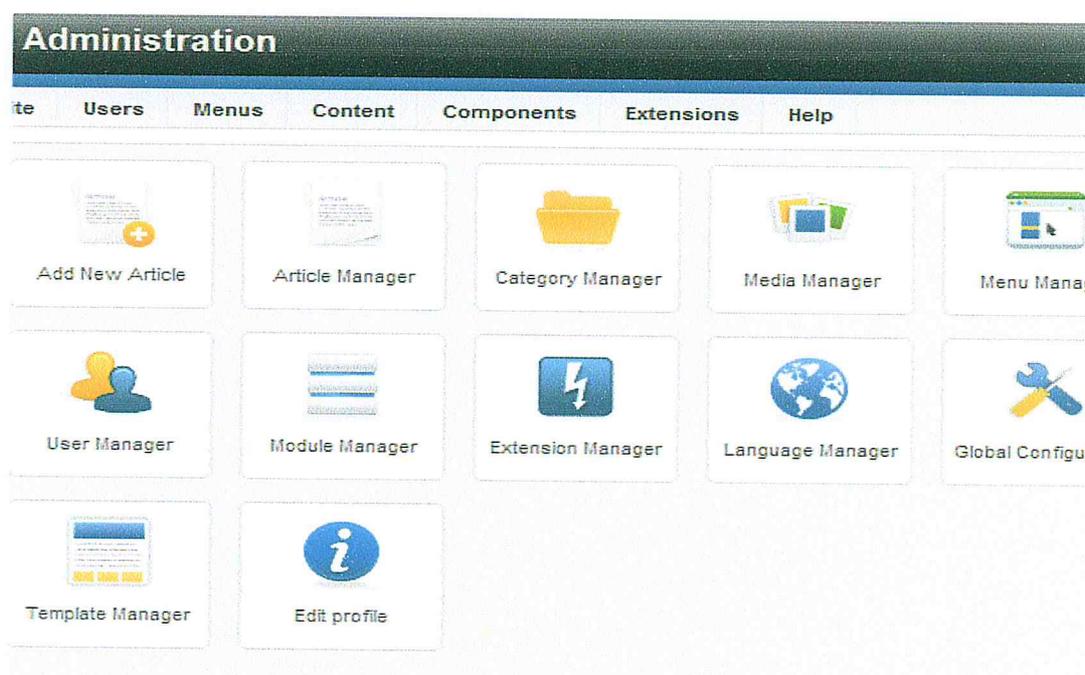


Figure 1 : Exemple d'interface d'administration Joomla 2.5

3.1 Avantages de Joomla 2.5

- Plus de sécurité avec des mises à jour régulières et une stratégie de développement transparente de Joomla
- Un système simple et fiable de mises à jour des versions de Joomla intégré à l'administration
- Une feuille de route jusqu'en 2015 avec Joomla 3.0 et Joomla 3.5 à l'été 2013
- Support d'autres bases de données telles Microsoft SQL Server
- Intégration du moteur de recherche "Finder"
- Un système de notification des mises à jour de Joomla et des extensions
- Un système de catégories imbriquées et illimitées, plus de sections...
- Captcha et multilangue intégrés en natif à Joomla 2.5
- Attribution des modules par éléments de menu
- Paramètre "note" pour les éléments de menu et les modules très utile. [4]

4 Conclusion

Joomla est rapidement devenu un acteur incontournable dans la réalisation de portail de diffusion de ressources. Basée sur des licences ouvertes, cette solution Open Source est une référence en la matière basée sur un corps CMS.

Chapitre 2

Étude de

systeme

Joomla

1 Introduction

Il est très utile d'aborder l'architecture Joomla avant de développer des extensions. Avec cette connaissance de base, il est possible de mieux comprendre la structure des composants, des modules et plugins. Durant ce chapitre nous allons décrire l'architecture Joomla à un niveau très abstrait, Par conséquent, constituer un bon point de départ dans le développement de Joomla.

2 Principes de fonctionnement

Joomla est un logiciel écrit en langage PHP et fonctionne avec une base de données MYSQL. Son fonctionnement peut être résumé par le schéma suivant :

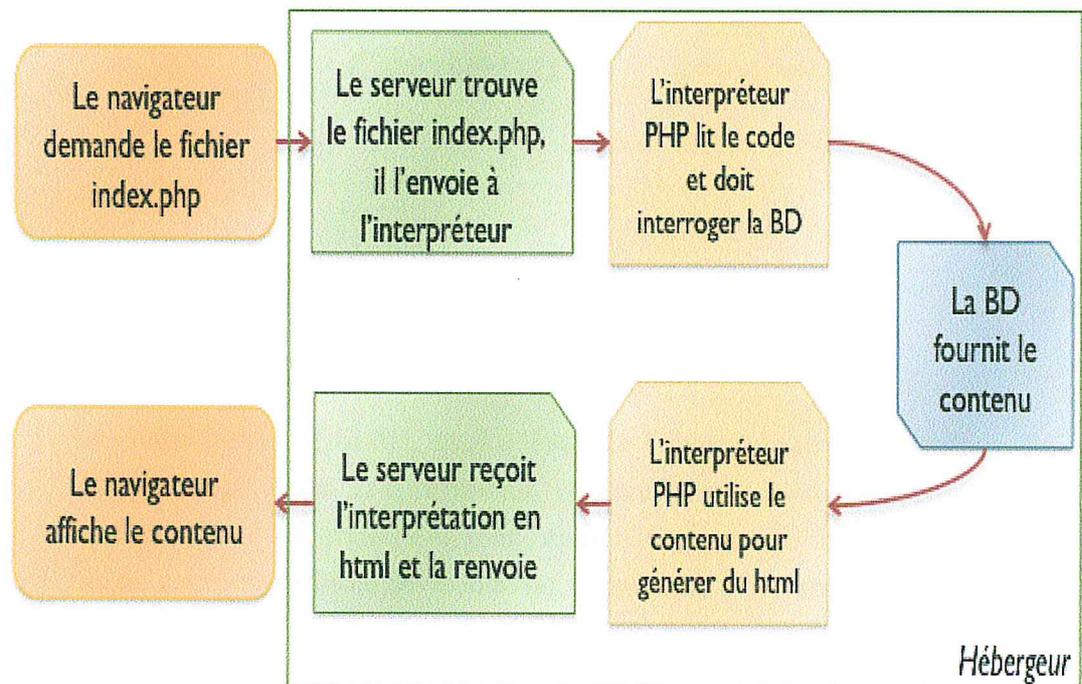


Figure 2 : Fonctionnement de pages web en PHP [5]

L'idée principale à retenir de ce schéma est la dissociation entre aspect du site et contenu du site. Le contenu est stocké de manière brute dans la base de données, l'aspect est géré par le logiciel (à l'aide des templates).

- des Utilisateurs
- des Groupes
- des Niveaux d'accès

Dans une architecture de droits d'accès Joomla (Figure 3), nous avons :

- a) **Des utilisateurs** : les comptes des membres de votre site.
 - b) **Des groupes** : qui regroupent les utilisateurs par fonction (auteurs, gestionnaires, administrateurs ..) un utilisateur peut désormais faire partie de plusieurs groupes.
 - c) **Des niveaux d'accès** : qui définissent quel groupe a accès (public, enregistré, spécial) a quel contenu (articles, catégories, modules...).
 - d) **Des permissions** : qui accordent des actions (login, création, suppression, configuration, publication ...) a des groupes d'utilisateurs.
 - e) **Des contenus** : toutes les données de votre site qui peuvent soumis à la gestion des droits de Joomla (articles, catégories, modules ...).
- III. **L'aspect visuel** : habillage du site, position des éléments, affichage des contenus.(Figure 4)
- IV. **Les extensions** : installation, publication, accès, affichage..(Figure 5)

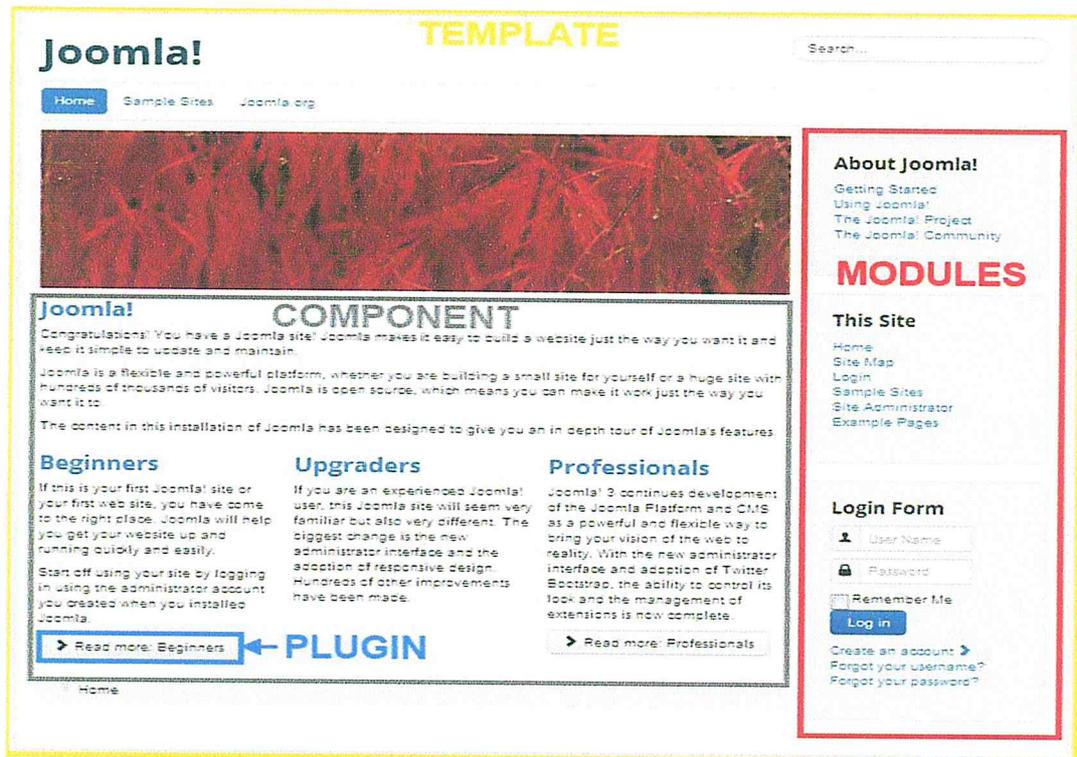


Figure 4 : Les templates dans joomla [7]

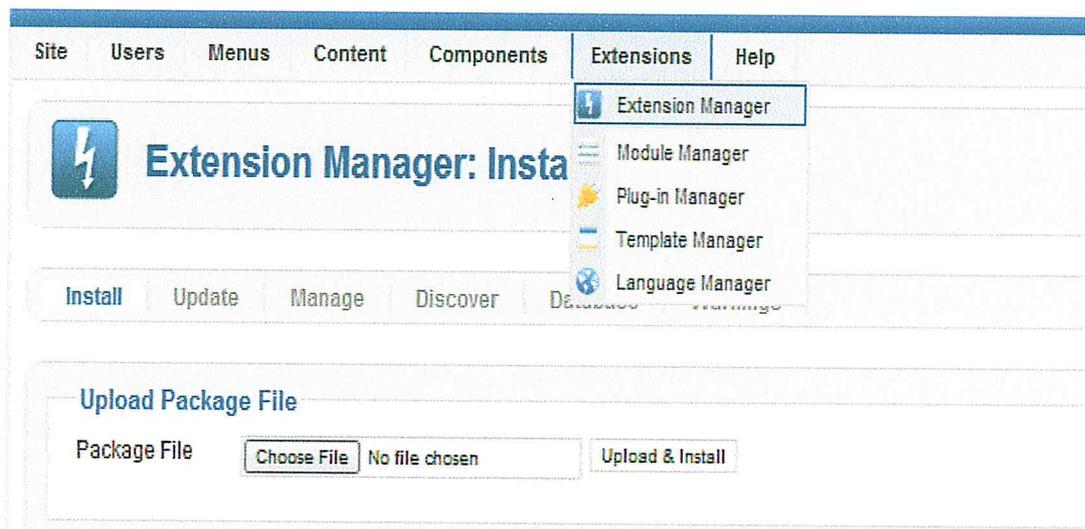


Figure 5 : Interface de la gestion des extensions dans Joomla

3 Architecture Joomla

Il est structuré autour d'une architecture en 3 couches [8] :

- i. Une couche permettant de gérer des extensions fournies par la communauté des développeurs (Module, Composant, Templates).
- ii. Une couche incluant le niveau applicatif, afin que qualifier le comportement de Joomla selon l'application en cours d'exécution (Frontend, Backend, Installation)
- iii. La couche de socle technique qui contient tout le Core de Joomla, les bibliothèques additionnelles et les Plugins qui sont communément reconnues comme étant des extensions additionnelles, leur place est bien localisée aux côtés du Framework. Ils sont effectivement très fortement liés à la gestion des évènements du système fourni par le Fmamework ... indépendamment de l'application qui est en cours d'exécution.

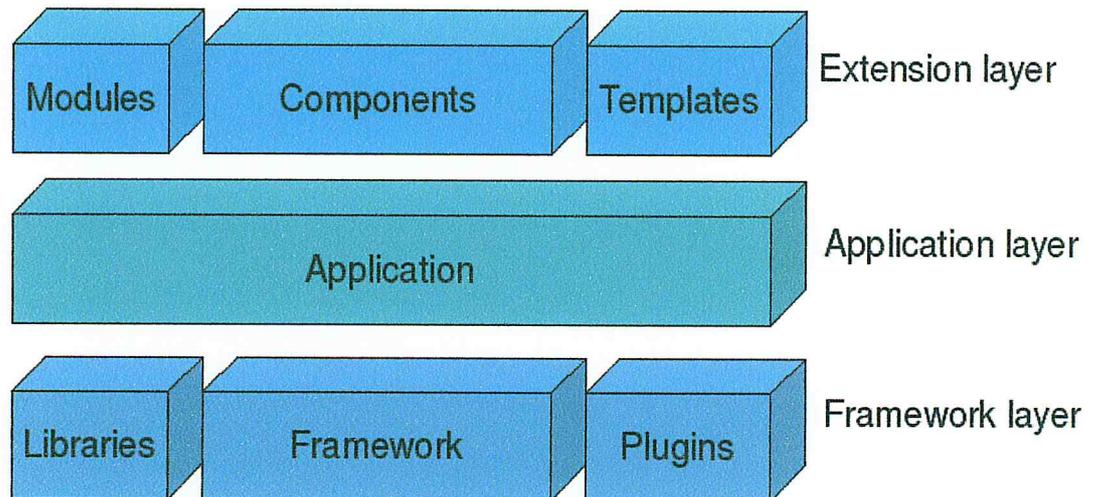


Figure 6 : Les différentes couches dans Joomla [8]

3.1 Les extensions dans Joomla

Joomla est un outil de gestion de contenu assez sophistiqué qui s'appuie sur des extensions, c'est-à-dire des programmes complémentaires pour gérer la mise en forme ou ajouter des nouveaux services. Ces extensions sont classées en 3 catégories : les composants, les modules, et les Templates. La version standard de Joomla intègre un certain nombre d'extensions mais vous en trouverez quelques milliers sur le net pour personnaliser votre site.

- a) **Composant** : c'est une mini application intégrée à votre site Joomla, qui dispose de sa propre interface de configuration dans la console d'administration Joomla. Ainsi à chaque fois qu'une page est chargée, Joomla fait appel à un composant pour générer le corps de la page, de même, il existe un composant pour authentifier les utilisateurs ... Les composants de base sont fournis avec Joomla. D'autres composants peuvent être facilement installés par la suite (forums, galerie d'images, gestionnaire de newsletter, gestionnaire de formulaires... et bien d'autres encore).

exemple : com_content (gestion des contenus) et com_registration (enregistrement des utilisateurs).

- b) **Module** : en effet, un module est un bloc que l'on trouvera généralement autour du corps de la page web, par exemple dans la colonne de gauche ou la colonne de droite de notre site, ainsi le menu de gauche de votre site est placé dans un module. De même que la bannière en haut de votre site, le bas de page ou le module d'identification...

Les modules sont souvent associés à des composants, comme par exemple le module qui affiche une photo aléatoire tirée d'une galerie d'images gérées par un composant.

Exemple : mod_banners (affichage des bannières), mod_mainmenu (affichage d'un menu).

- c) **Template** : un Template gère toute la partie graphique de votre site : les couleurs des caractères et des fonds, la police des caractères, les cadres, les menus ...

En changeant de Tamplate, vous changez le style de votre site. Il en existe des centaines, disponibles gratuitement sur des sites qui se sont spécialisés dans cette activité.

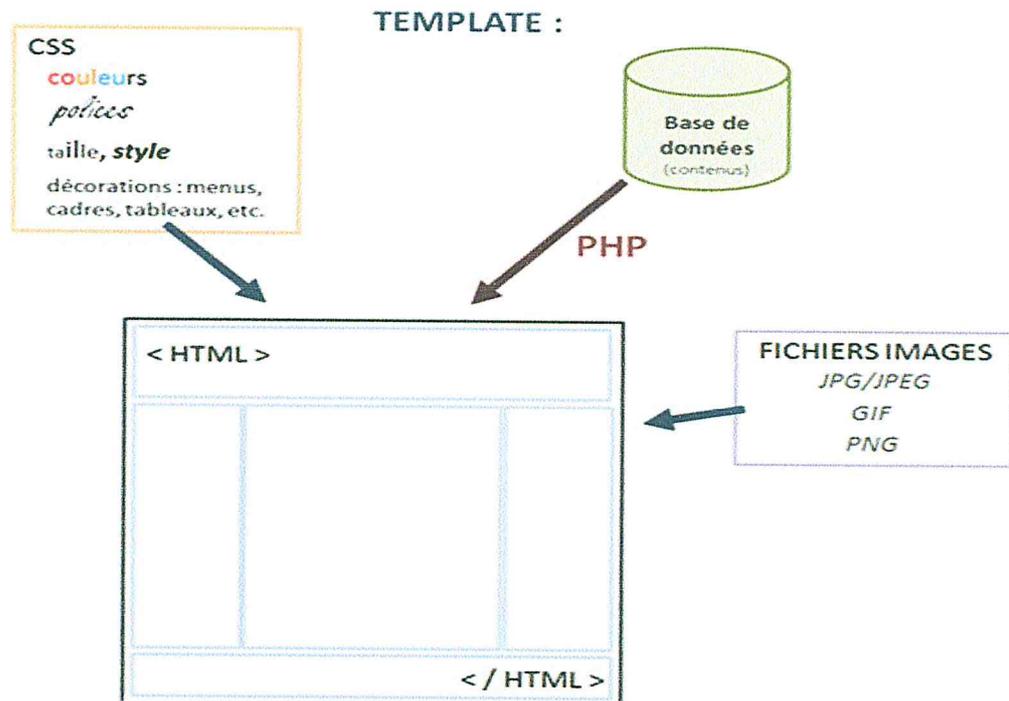


Figure 7 : Les Templates dans Joomla [9]

3.2 Application

S'appuyant sur le niveau de cadre pour trouver le niveau de l'application (couche application). La couche d'application se compose de trois applications, toutes les classes de cadre pour Joomla 2.5 Japplication avec un héritage. La première application est JInstallation, qui est responsable de l'installation de Joomla. Les deux autres sont JAdministrator et jSite, comme leurs noms l'indiquent déjà JAdministrator est responsable de l'interface back-end de Joomla et jSite pour le front-end.

3.2.1 JInstallation

L'application JInstallation s'exécute lorsque vous installez Joomla. Après avoir réussi à l'installation de Joomla, vous devez supprimer le dossier d'installation, qui contient l'application de JInstallation, avant de continuer. L'installation d'extensions (composants, modules, plugins, templates et langues) est réalisée à l'aide de la fonctionnalité d'installation de l'application JAdministrator.



Figure 8 : Interface installation Joomla [10]

3.2.2 Back-end

Joomla permet de développer rapidement des sites très puissants où les informations présentées peuvent être changées facilement et rapidement par des utilisateurs non informaticiens.

Son menu d'administration présente les éléments de gestion du contenu plutôt

3.3 FRAMWORK

Joomla Framework est une plateforme pour écrire des applications web en ligne et commande en PHP, c'est un logiciel libre et open source. Framework comprend un code développé à l'origine pour le système de gestion de contenu Joomla et s'inscrit dans la continuité ou projet de plate-forme. Framework Joomla et le CMS Joomla sont des projets distincts avec les équipes de développement distinctes. Contrairement à la plate-forme, Framework n'est pas principalement conçu comme une plate-forme pour le CMS Joomla, et le CMS Joomla n'est pas actuellement en construction sur le Framework.

3.3.1 FRAMWORK JOOMLA

Le Framework est constitué d'un ensemble de classes qui fournit de nouvelles fonctionnalités. Une liste de classes peut être trouvée dans les annexes ou vous pouvez naviguer sur le site officiel de Joomla. Les classes qui ratrapent le cadre, étant regroupées en paquets afin de faciliter la classification et l'identification.



Les paquets qui composent le Framework Joomla sont énumérés dans le tableau suivant :

Package	Description
Application	JApplication and related classes
Base	Base classes
Cache	Cache classes
Client	FTP and LDAP classes
Database	JDatabase and related classes
Docunemt	Classes for creating and rendering pages
Environment	URI, Request/Response handling, and browser classes
Error	Error handling, logging, and profiling classes
Event	Dispatch and event classes
Filesystem	Classes supporting file access
Filter	Input and output filter classes
HTML	Classes for rendering HTML
Installer	Classes for installing extensions
Language	Language translation classes
Mail	E-mail related classes
Plugin	Core Plugin classes
Registry	Configuration classes
Session	User session handler and storage of session data classes
User Site	User classes
Utilities	Miscellaneous classes

Tableau 1 : Paquets Joomla [12]

3.3.2 PLUGINS

Les plugins sont créés pour étendre les fonctionnalités du Framework, ce sont des morceaux de code activés sur un évènement. L'exécution de n'importe quelle partie de Joomla, que ce soit le noyau, en module ou un composant, peut déclencher un évènement et alors les plugins associés à cet évènement s'exécuteront. Par exemple, le plugin de recherche content *Searchbot* est un script qui permet d'effectuer une recherche sur tout votre site.

Autre exemple, quand vous saisissez le texte d'un article, le plugin *TinyMCE* permettra de convertir le texte que vous saisissez en code HTML interprétable par votre navigateur : un évènement est l'appui sur une touche du clavier, l'action est la génération du code html adéquat.

Joomla est livré avec huit noyaux, groupes de greffons, chacun conçu pour gérer un ensemble spécifique d'évènements.

Le tableau suivant décrit les différents types de plugin de base :

Plugin Type	description
Authentication	Authenticate users during the login process
Content	Process content items before they are displayed
Editors	WYSIWYG editors that can be used to edit content
Editors-xtd	Editor extensions (additional editor buttons)
Search	Search data when using the search component
System	System event listeners
User	Process a user when actions are performed
xmlrpc	Create XML-RPC responses

Tableau 2 : Plugins de base de Joomla [12]

4 Conclusion

Une bonne connaissance de l'environnement Joomla nous donne la possibilité d'ajuster nos propres solutions selon nos besoins tout en s'appuyant sur des bases fondamentales de cet espace de travail.

Partie II

Mise en œuvre du système

Chapitre 3

Le modèle

MVC

1 Introduction au domaine MVC

Plusieurs architectures sont proposées pour les projets informatiques de plus en plus complexes à réaliser mais surtout à maintenir. Une bonne organisation du code est fortement recommandée afin de faciliter ces deux aspects. Parmi ces architectures nous trouvons MVC qui donne une vue très courante pour la réalisation des gros projets surtout ceux nécessitant beaucoup d'interactions avec les utilisateurs telles que les CMS et les bibliothèques graphiques des langages évolués. Le MVC tout comme l'orienté objet du code, semble être devenu un standard dans le développement d'applications web, avec la réputation d'être une bonne pratique de conception.

De nombreux Framework utilisent aujourd'hui MVC, car le but principal de ce motif est de séparer les couches logiques d'une application. Nous allons voir qu'il n'est pas toujours judicieux d'utiliser MVC, ce motif sera plutôt réservé aux sites dans lesquels beaucoup de redondances de code apparaissent en général, des sites souvent mis à jour, souvent remodelés.

Dans le cadre d'une petite application web la complexité en termes de code apportée par MVC, ne sera pas justifiée, en revanche pour tout autre projet web, la séparation en plusieurs couches permet à différentes équipes de travailler chacune sur une couche, indépendamment des autres, et c'est aussi un gros avantage lorsqu'une couche doit évoluer sans que les autres n'en aient besoin. A chaque fois que nous traitons un simple code publié sur le web, nous pouvons comprendre probablement à quoi il est destiné, l'inconvénient, c'est que les couches sont mélangées. Si nous essayons de décomposer n'importe quel code avec un gros volume dans le traitement d'informations sur le web, généralement nous distinguons 3 parties qui sont mélangées : une qui gère la manière d'accès aux données, l'autre qui se charge de présenter les données, et la dernière est l'étape intermédiaire de sélection et fusion des données. En effet, il est plus facile de dire de quoi est constituée une vinaigrette lorsque les ingrédients sont tous posés sur la table, plutôt que d'essayer d'analyser leur mélange au fond du saladier, nous pouvons donc factoriser le code pour bien séparer ces 3 parties, Et voilà si nous appliquons le modèle MVC nous allons bien séparer le code en 3 couches distinctes, qui se connaissent l'une et l'autre. Dans ce code-là,

nous pouvons facilement changer la manière d'accéder aux données (changer le SGBD, par exemple), sans pour autant se soucier le moins du monde de la présentation, ou de la manipulation future de ces données.[13]

Cependant il n'est pas toujours facile de bien distinguer les couches, et le motif MVC reste sans aucun doute un motif complexe car, les développeurs habitués à ce modèle apprécient particulièrement la lisibilité du code, mais comme toujours, la séparation des entités logiques, réalisable grâce à la programmation orientée objet, simplifie et clarifie la conception d'un logiciel (et dans notre cas d'une application web) donc nous pouvons ainsi créer une quantité innumérable de motifs MVC, répondant chacun à un problème particulier. Cependant la plupart du temps, les développeurs utiliseront des modèles MVC déjà écrits, dans des Framework ayant fait leurs preuves.

2 Avantages et inconvénients

D'une part un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose, cela rend le processus de maintenance ou d'amélioration plus facile, en séparant la couche du traitement des données et les traitements des événements et d'affichages cela diminue le taux d'erreurs qui ramène à un gain de temps d'implémentation, de maintenance et d'évolution des applications surtout qu'il facilite la notion du travail en groupe sur un seul projet cela assure une grande souplesse afin d'organiser le développement entre différents développeurs.

D'autre part cette méthode augmente le nombre de lignes du code. Cela rend sa complexité plus élevée pour un débutant non expert, mais à force du temps ce dernier va s'habituer, ce qui lui donne une vue très vaste afin de réaliser ses projets en se basant sur cette notion d'MVC.

3 Principe du modèle MVC

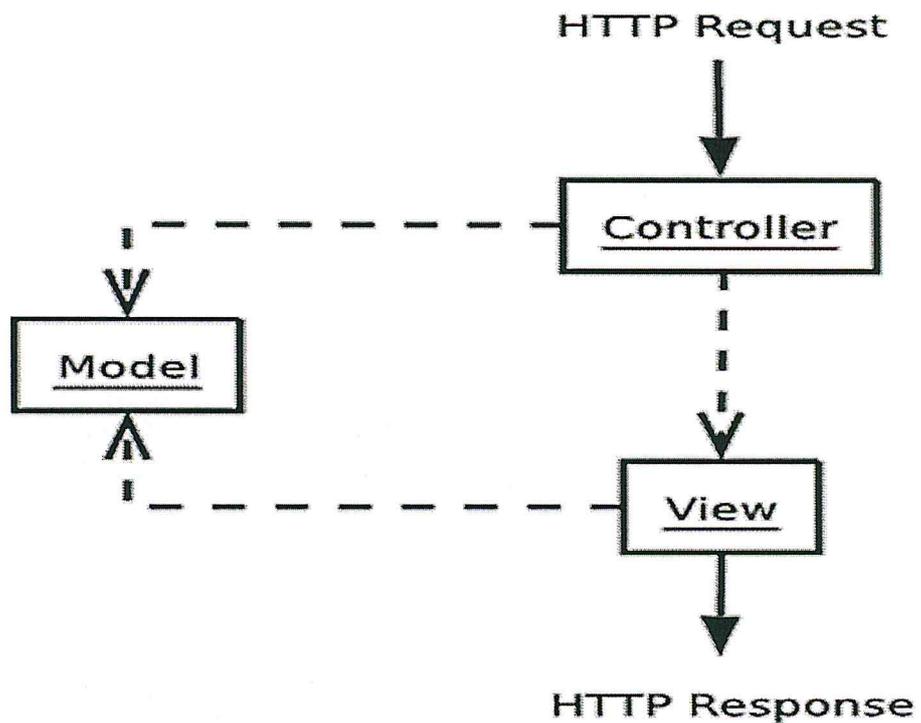


Figure 11 : Schéma du modèle MVC [13]

Diagramme de séquence d'un MVC simple :

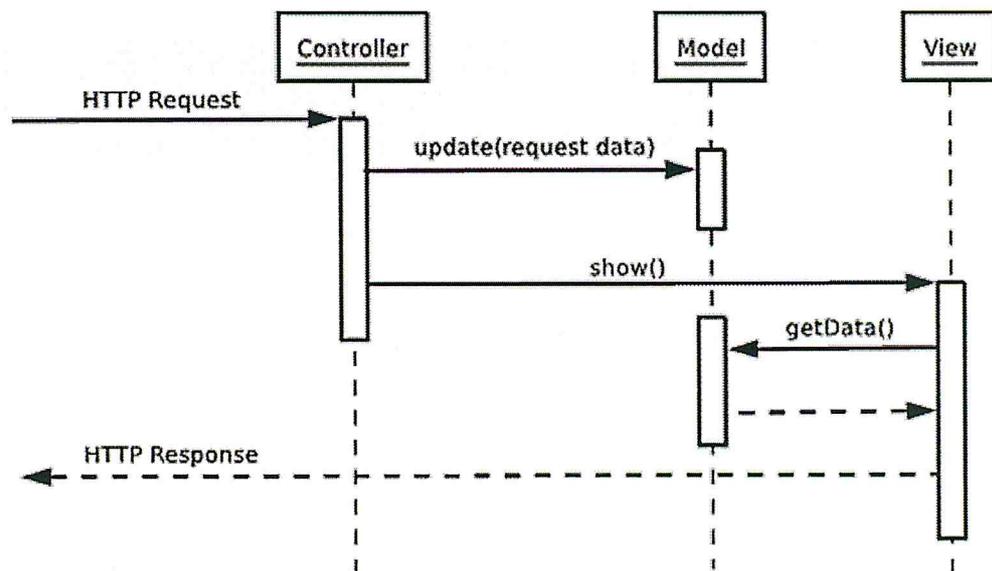


Figure 12 : Diagramme de séquence du modèle MVC [13]

Le but de MVC est de séparer les couches d'une application (en 3 couches distinctes, au minimum et le plus souvent). Nous allons donc distinguer : [13]

- i. **Le modèle** : qui encapsule le logique métier et la manipulation des sources de données. Ici, des designs patterns peuvent intervenir.
- ii. **La vue** : présente les données au client les demandant (un utilisateur humain ou un autre programme). Ici aussi on peut introduire des motifs : `TemplateView` par exemple.
- iii. **Le contrôleur** : c'est tout le reste, c'est lui qui analyse la requête du client, accède aux données, formate le tout et l'envoi à la partie vue qui va présenter tout ça.

nous allons détailler un peu. **La vue**, c'est un `TemplateView`. Nous utilisons PHP lui-même pour écrire nos variables dans du HTML, ou alors un moteur de `Template`.

Le `TransformView` peut aussi être utilisé pour sortir en transformant les données renvoyées par les modèles, qui sont en XML, grâce à XSLT et les feuilles de style XSL. Une sortie en PDF peut aussi être utilisée ou encore une présentation JSON des données, dont les clients javascript et Ajax sont très efficaces.

Il est très important de noter que la vue ne fait que présenter des données et à aucun moment elle ne connaît leur provenance, c'est le modèle qui va fournir toutes les méthodes nécessaires à la récupération, et le contrôleur qui se chargera de cette récupération et de sa transformation éventuelle afin de la coller à la vue.

Du côté du **modèle**, il est très important que celui-ci ne connaisse ni les contrôleurs, ni les vues. Les liens vont des contrôleurs vers le modèle, de temps en temps, des vues vers le modèle, mais jamais dans le sens inverse.

Le modèle étant totalement indépendant du reste, il pourra faire l'objet d'une réflexion et d'un développement à part. Il pourra de même être utilisé par d'autres composants, tels que des web services ou d'autres applications locale. Les données manipulées par le modèle doivent persister entre toutes les requêtes HTTP, ainsi un support de stockage devra être utiles, dans la majeure partie des cas, la source de données est une base de données relationnelle.

Il est possible que la vue accède au modèle, c'est un choix architectural qui devra être bien réfléchi, car il impose un plus fort couplage et un mélange des couches, nous appelons cela l'approche *pull*. C'est souvent réservé aux opérations légères, comme par exemple afficher une liste de catégories d'éléments quelconques et aux opérations de lecture uniquement. L'approche inverse, appelée *push*, interdit à la vue d'accéder au modèle. En général, nous utilisons les 2 choix en même temps : une instruction de lecture simple (récupérer la liste des membres) sera demandée directement par la Vue, ou un de ses sous-composants. En revanche une opération de modification du modèle, sera demandée par un contrôleur.

Et nous arrivons enfin au **contrôleur**, qui est le composant principal du modèle MVC, il gère les événements et la synchronisation de l'application, il récupère les événements venant de la vue, déterminer quel traitement à effectuer et appelle les modèles et les vues nécessaires à leur réalisation en prenant le soin de faire la correspondance entre les données reçues par la vue et celles définies dans le modèle.

4 Utiliser l'architecture MVC dans les applications Web/PHP

Le modèle MVC (Modèle-Vue-Contrôleur) cherche à séparer nettement les couches présentation, traitement et accès aux données.

Une application web respectant ce modèle sera architecturée de la façon suivante :[14]

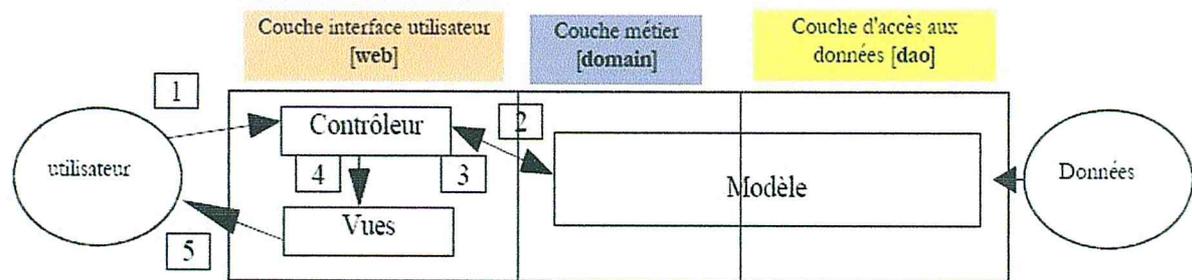


Figure 13 : Fonctionnement du MVC dans le web [14]

Le traitement d'une demande d'un client se déroule selon les étapes suivantes :

- 1) Le client fait une demande au contrôleur. Ce contrôleur fait passer toutes les demandes des clients, c'est la porte d'entrée de l'application. C'est le C de MVC.
- 2) Le contrôleur traite cette demande, pour se faire, il peut avoir besoin de l'aide de la couche métier, ce qu'on appelle le modèle M dans la structure MVC.
- 3) Le contrôleur reçoit une réponse de la couche métier. La demande du client a été traitée, celle-ci peut appeler plusieurs réponses possibles. Un exemple classique est:
 - Une page d'erreurs si la demande n'a pu être traitée correctement
 - Une page de confirmation sinon
- 4) Le contrôleur choisit la réponse (= vue) à envoyer au client, celle-ci est le plus souvent une page contenant des éléments dynamiques. Le contrôleur fournit ceux-ci à la vue.
- 5) La vue est envoyée au client. C'est le V de MVC .

Une telle architecture est souvent appelée "architecture 3-tier" ou à 3 niveaux.

L'interface utilisateur est souvent un navigateur web mais cela pourrait être également une application autonome qui via le réseau enverrait des requêtes HTTP au service web et mettrait en forme les résultats que celui-ci lui envoie. La logique applicative est constituée des scripts traitant les demandes de l'utilisateur, des classes métier et d'accès aux données.

La source de données est souvent une base de données mais cela peut être aussi de simples fichiers plats, un service web distant, Le développeur intérêt à maintenir une grande indépendance entre ces trois entités afin que si l'une d'elles change, les deux autres n'aient pas à changer ou peu .

L'architecture MVC est bien adaptée à des applications web écrites avec des langages orientés objet. Le langage PHP (4.x) n'est pas orienté objet donc nous pouvons néanmoins faire un effort de structuration du code et de l'architecture de l'application afin de se rapprocher du modèle MVC :

On mettra le logique métier de l'application dans des modules séparés des modules chargés de contrôler le dialogue demande-réponse. L'architecture MVC devient la suivante :[14]

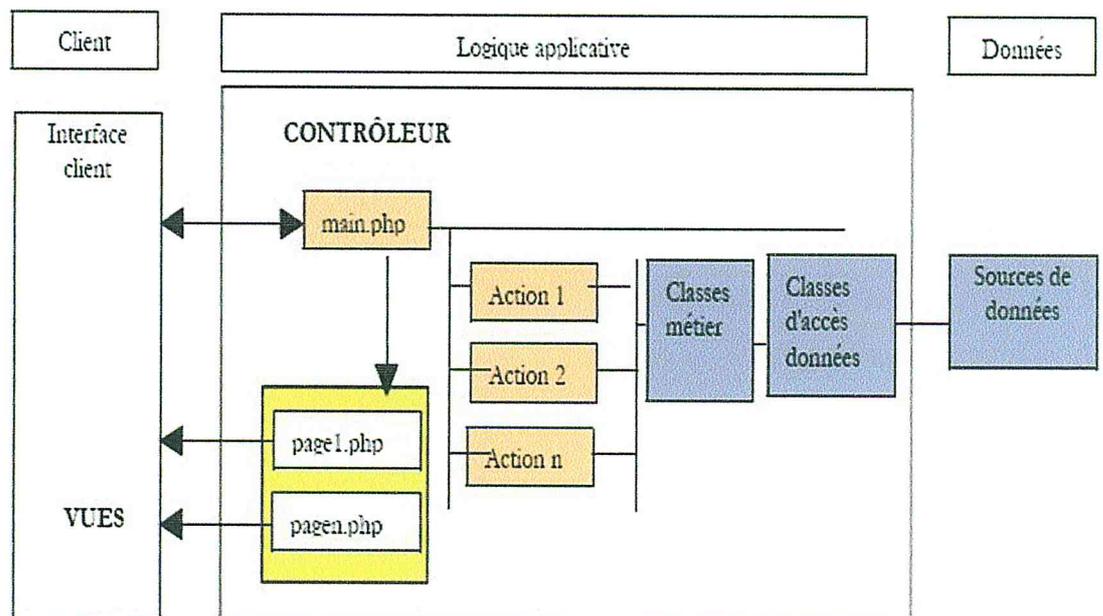


Figure 14 : Fonctionnement de MVC [14]

M=modèle	les classes métier, ses classes d'accès aux données de la base de données
V=vues	les pages PHP
C=contrôleur	le script PHP de traitement des requetes clientes, les scripts PHP [Action] de traitement des actions.

Tableau 3 : Logique Applicative[14]

Dans le bloc [Logique Applicative], on pourra distinguer :

- 1) Le programme principal ou contrôleur [main.PHP], qui est la porte d'entrée de l'application.
- 2) Le bloc [Actions], ensemble de scripts PHP chargés d'exécuter les actions demandées par l'utilisateur.
- 3) Le bloc [Classes métier] qui regroupe les modules PHP nécessaires à la logique de l'application. Ils sont indépendants du client. Par exemple, la fonction permettant de calculer un impôt à partir de certaines informations qui lui sont fournies en paramètres, n'a pas à se soucier de la façon dont ont été acquises celles-ci.
- 4) Le bloc [Classes d'accès aux données] qui regroupe les modules PHP qui obtiennent les données nécessaires au contrôleur, souvent des données persistantes (BD, fichiers)
- 5) Les générateurs [pagex.php] des vues envoyées comme réponse au client .

Dans les cas les plus simples, la logique applicative est souvent réduite à deux modules :

- i) Le module [contrôle] assurant le dialogue client-serveur : traitement de la requête, génération des diverses réponses.
- ii) Le module [métier] qui reçoit du module [contrôle] des données à traiter et lui fournit en retour des résultats. Ce module [métier] gère alors lui-même l'accès aux données persistantes .

5 Conclusions

L'architecture MVC est une solution très intéressante et très puissante, à mettre en œuvre dans de grands projets de préférence. L'indépendance des trois couches est très utile, car c'est grâce à cette différenciation que l'architecture réalise sa puissance et sa logique, il est ainsi facile de mettre à jour le design de notre site, sans toucher aux données ni à la façon dont elles sont organisées. Par contre, il est aussi facile de modifier la structure de stockage des données ou la manière dont elles sont gérées sans bouleverser l'affichage, ceci permet aussi de garder en tête le cheminement des requêtes, de travailler avec des diagrammes, de simplifier indirectement la gestion de projet.

Chapitre 4

Analyse

et

Conception

1 Introduction

Avant d'entamer n'importe quel processus de développement d'un projet, il est indispensable de se baser sur une bonne conception pour désigner l'ensemble des étapes préparatoires nécessaires au lancement du projet. Il s'agit donc de définir précisément ce que sera le projet afin d'obtenir de bons résultats, cette validation nous donnera l'assurance que la version de notre site soit conforme aux meilleures pratiques et que la flexibilité que nous intégrons nous permettra d'ajouter facilement et de manière efficace des caractéristiques et des fonctionnalités aux versions ultérieures.

2 Conception uml d'une application

Les techniques de programmation n'ont cessé de progresser depuis l'époque de la programmation en langage à nos jours. Cette évolution a toujours été dictée par le besoin de concevoir et de maintenir des applications toujours plus complexes.

Avec la mise en commun de l'expérience et la maturation des savoir-faire, nous voyons se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes. UML, qui se voit comme un instrument de capitalisation des savoir-faire puisqu'il propose un langage qui soit commun à tous les experts du logiciel, va dans le sens de cet assouplissement des contraintes méthodologiques.

Du point de vue des notations employées en UML, les différences entre l'analyse et la conception se traduisent avant tout par des différences de niveau de détail dans les diagrammes utilisés, toutes les vues proposées par UML sont complémentaires les unes des autres, elles permettent de mettre en évidence différents aspects d'un logiciel réaliser. Nous pouvons organiser une présentation d'UML autour d'un découpage en vues, ou bien en différents diagrammes, selon que nous séparons plutôt les aspects fonctionnels des aspects architecturaux, ou les aspects statiques des aspects dynamiques.

3 Analyse

Cette partie reflète l'état actuel du système ainsi que ses fonctionnalités. Durant la procédure d'installation de Joomla 61 tables vont être créées dans la base de données que vous avez spécifiée, l'ensemble du contenu sera géré dans ces tables. à partir du schéma complet de Joomla (Figure 15), nous constatons la volumétrie de la base de données de ce dernier pour cela nous allons nous limiter dans notre étude juste sur les tables qui possèdent une relation (interagissent) avec nos problématiques à résoudre.

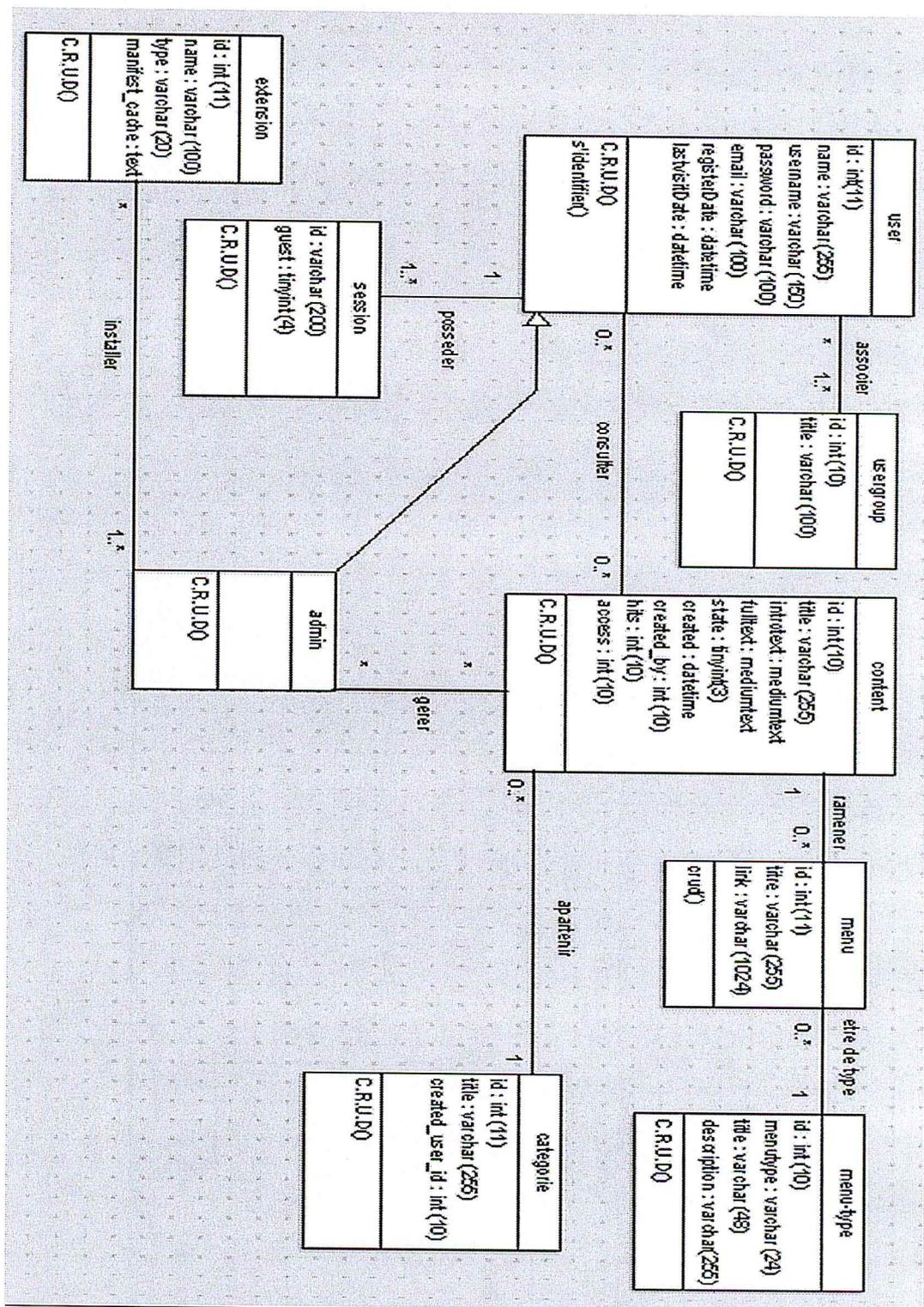


Figure 16 : Diagramme de classe - Analyse -

3.1 Diagramme de classe

Tout d'abord nous allons commencer par faire un diagramme de classe simplifié, montrant les tables nécessaires pour que le site fonctionne avec des fonctionnalités de base et un aperçu général de l'utilisation du CMS.

Dans ce diagramme de classe (Figure 16), nous avons des utilisateurs enregistrés (user) chaque utilisateur est caractérisé par plusieurs attributs: un identifiant, un nom ainsi qu'un pseudo et un mot de passe afin de pouvoir s'identifier, un email, la date d'enregistrement et la date de la dernière visite. Il pourra consulter les articles (content) du site, chaque utilisateur est associé à un groupe d'utilisateurs par exemple : un utilisateur a le droit d'écrire des articles donc il sera associé au groupe (auteurs).

La table user-groupe possède plusieurs attributs : un identifiant, un titre du type de groupe.

A chaque connexion d'un utilisateur une session lui est associée avec un attribut identifiant et un attribut guest indiquant la nature de cet utilisateur entre un utilisateur enregistré ou non.

Une table content (article), chaque article possède plusieurs attributs : un identifiant, un titre, une introduction du texte ainsi que le corps du texte, un attribut *state* qui indique si l'article sera affiché ou non, une date de création, l'identifiant du créateur et un attribut *hits* qui sert à déterminer le nombre de fois où l'article a été consulté, un attribut *Access* qui sert à déterminer le niveau d'accès. Ce niveau d'accès permet de savoir quel utilisateur pourra voir la page : soit les invités et les membres, soit uniquement les membres.

Les articles sont catégorisés selon des catégories, chaque catégorie possède comme attributs : un identifiant, un titre ainsi qu'un identifiant du créateur.

Chaque article peut être atteint avec un lien menu qui est caractérisé par un identifiant, un titre ainsi qu'un attribut Link qui contient le chemin vers cet article. Chaque menu possède un type-menu caractérisé par un identifiant, un menu-type, un titre et une description du type.

En suite, une table Admin qui hérite de classe user. Un administrateur pourra créer, modifier et supprimer des pages ou des articles, ainsi qu'installer des extensions pour performer le site où chaque extension est caractérisée par : un identifiant, un nom, un manifeste-cache qui englobe des informations relatives à cette extension ainsi qu'un type.

3.2 Diagrammes de cas d'utilisation

Un diagramme de cas d'utilisation, aussi appelé Use Case, permet de définir les droits de chaque utilisateur du logiciel, dans notre cas du site.

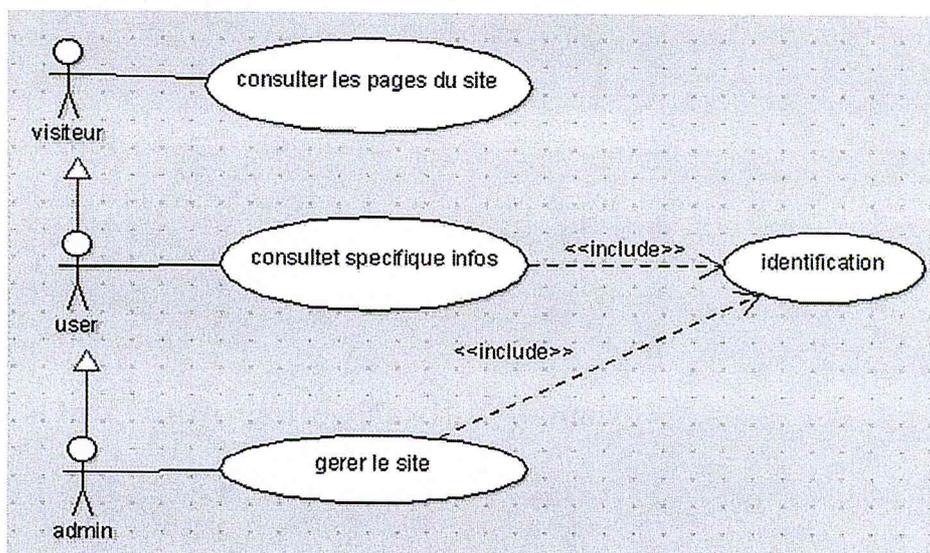


Figure 17 : Diagramme de cas d'utilisation globale

Nous avons donc trois types d'utilisateurs : les visiteurs, les membres (users) et l'administrateur. Les visiteurs ne peuvent que consulter les pages sur le site, via les différents menus.

Les membres peuvent visiter les pages du site, mais aussi consulter des pages qui possèdent un niveau d'accès défini par l'administrateur pour chaque membre. L'administrateur peut visiter les pages ainsi que s'occuper de la gestion du site.

Par suite nous allons entamer les diagrammes de cas d'utilisation avec notion de niveau qui facilite la compréhension des droits.

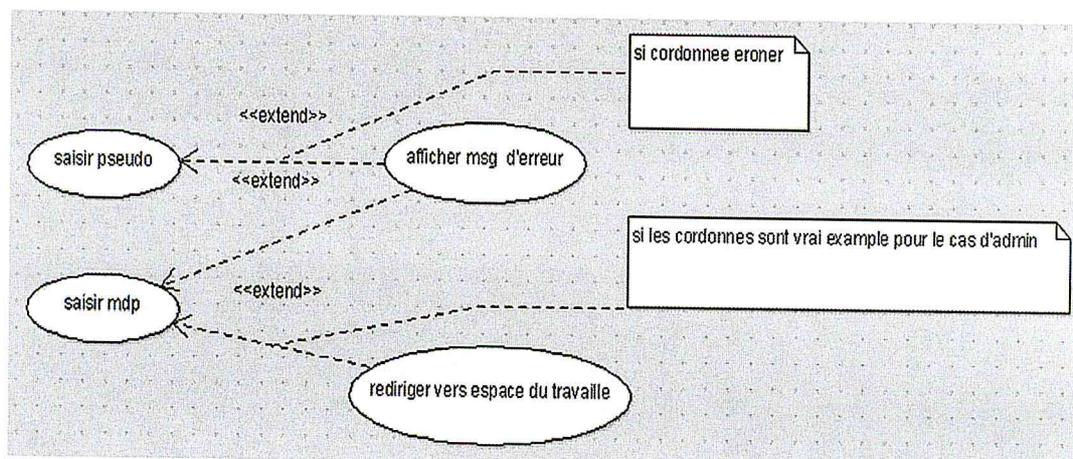


Figure 18 : Premier niveau d'authentification

A ce niveau les membres peuvent s'authentifier en utilisant leur identifiant ainsi que leur mot de passe, Au cas où les coordonnées soient fausses, un message d'erreur va apparaître, sinon le membre aura la possibilité à accéder aux pages qui lui sont autorisées. Si ce membre est un administrateur et qu'il est connecté au côté administration et que ces coordonnées sont valides il sera dirigé vers l'espace d'administrateur.



Figure 19 : Premier niveau du gestion du site

Dans cette partie l'administrateur peut gérer le site des différents cotés : améliorer ses performances à l'aide des extensions, ou bien gérer le contenu du site, exemple : articles, gérer la notion des membres de leur affecter des niveaux d'accès.

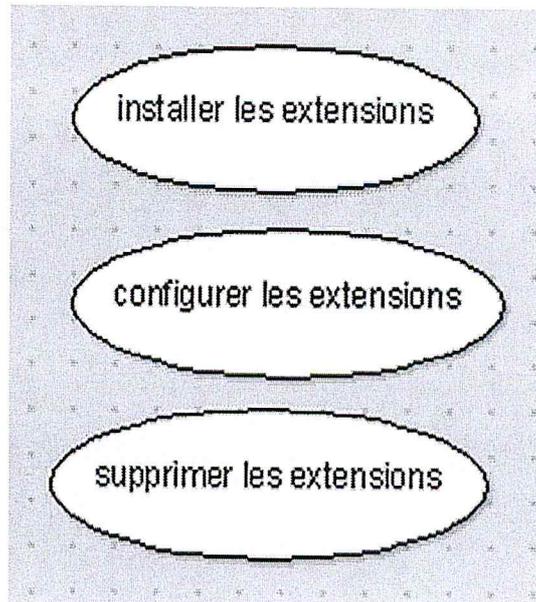


Figure 20 : Second niveau de gestion de site

Pour améliorer Joomla, l'administrateur a besoin des extensions pour donner de la valeur au site. Pour cela il est ramené à installer des extensions, ensuite de les configurer selon ses besoins ainsi que de les supprimer s'il nous n'avons pas besoin pour ne pas alourdir la rapidité d'interaction du site.

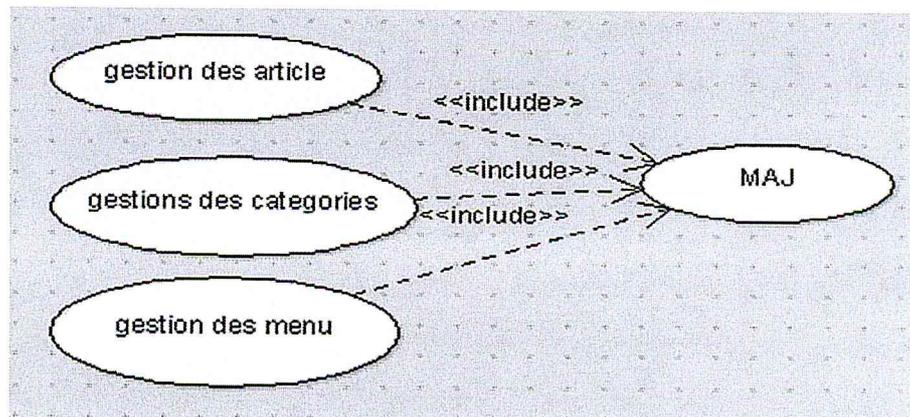


Figure 21 : Second niveau de gestion de contenu

La gestion de contenu s'exprime en différentes pages qui vont contenir des différents articles qui seront créés, modifiés ou supprimés, cela est de même pour les catégories qui contiendront ces articles. Nous accédons aux différents articles souvent par des liens dans des menus donc cela provoque un besoin pour la gestion des menus ainsi que leurs liens.

3.3 Diagrammes de séquence

Les diagrammes de séquence permettent de schématiser l'interaction entre les objets, ils prévoient ce qui va se passer à chaque action de l'utilisateur. Dans notre cas pour installer les extensions, l'administrateur doit se connecter au coté administration cela nous amène à traiter son authentification comme suit:

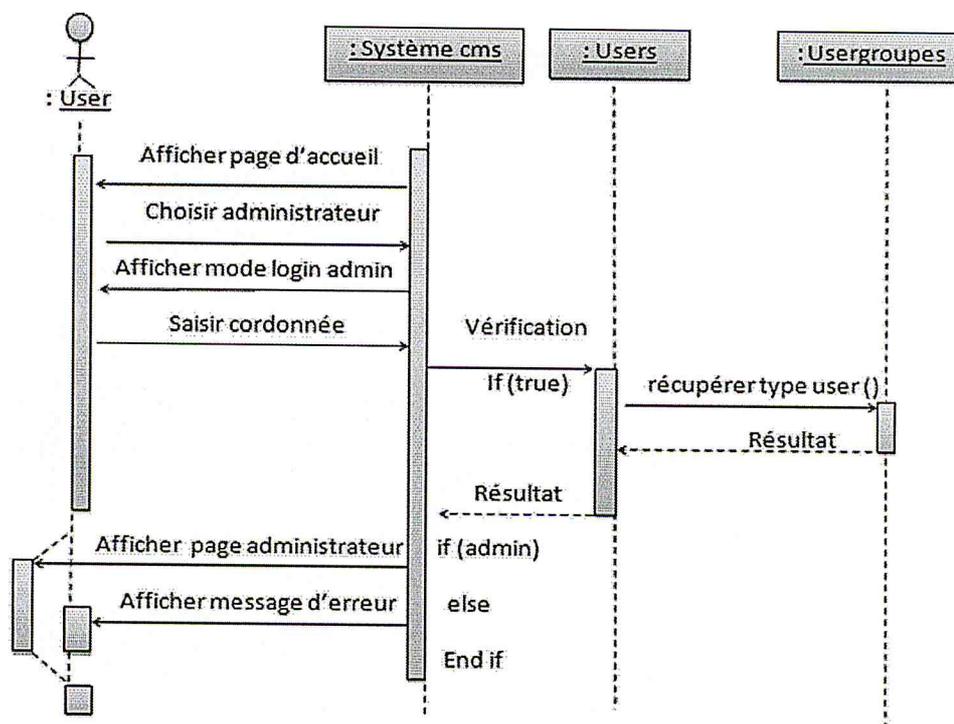


Figure 22 : Diagramme de séquence - Identification -

Lors de connexion d'administrateur au coté administration, un angle d'authentification va apparaître, il doit saisir son identifiant, en d'autres termes son login ou son adresse mail et son mot de passe, et le système se connecte à la

base de données pour vérifier si les informations rentrées sont exactes. Pour cela, il va regarder dans la table users si le login existe, puis vérifier que le mot de passe entré y est bien associé, ensuite il va consulter la table usergroupe pour vérifier et que ce membre fait partie de groupe administrateur, Si c'est bon, alors l'administrateur est connecté et il sera renvoyé au coté administration.

Après l'authentification si l'administrateur veut améliorer les performances du site à l'aide des extensions il doit les installer en suivant cette procédure :

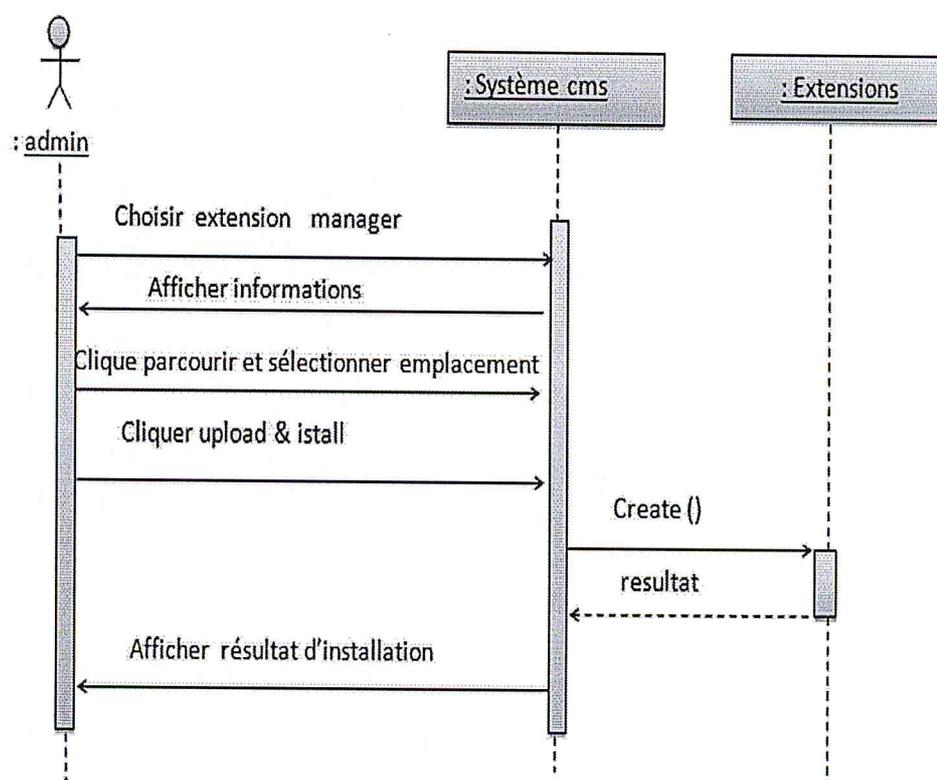


Figure 23 : Diagramme de séquence - installation des extensions -

Lorsque l'administrateur veut installer une extension, il doit cliquer sur l'icône *extension manager*, une page contenant des différentes informations va apparaître. Le premier angle est associé à l'installation des extensions, l'administrateur doit parcourir des dossiers afin d'atteindre l'emplacement d'une extension à installer ensuite il doit valider l'installation, ensuite le système l'effectue, donc

il se connecte à la base de données pour insérer des nouvelles données dans la table extensions. À la fin de l'installation l'administrateur va recevoir un message indiquant le résultat d'installation.

3.4 Diagramme d'états transition

Ces diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres qui provoquent un changement de comportement. Puisque nous allons réaliser un composant de statistique qui prend en charge des articles, nous devons parler donc de leurs transition d'état.

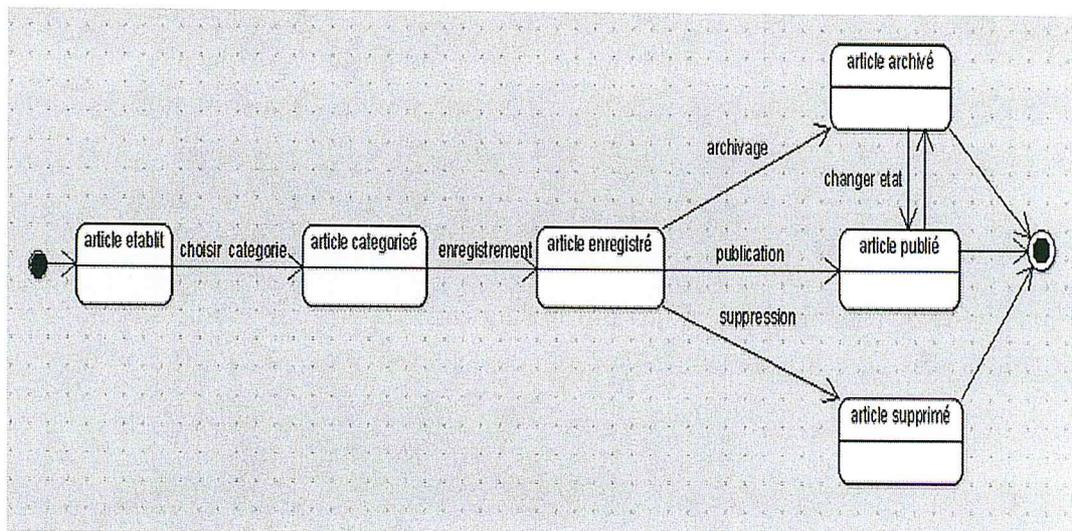


Figure 24 : Diagramme états transition - cas d'articles -

Lors d'une création d'un nouvel article, l'administrateur doit l'associer à une catégorie donc il sera classifié selon sa catégorie, ensuite il faut remplir ces différentes informations relatives à cet article puis valider l'enregistrement pour que son état puisse être enregistré, au fur et à mesure l'état d'article peut être changé entre article archivé, publié ou supprimé.

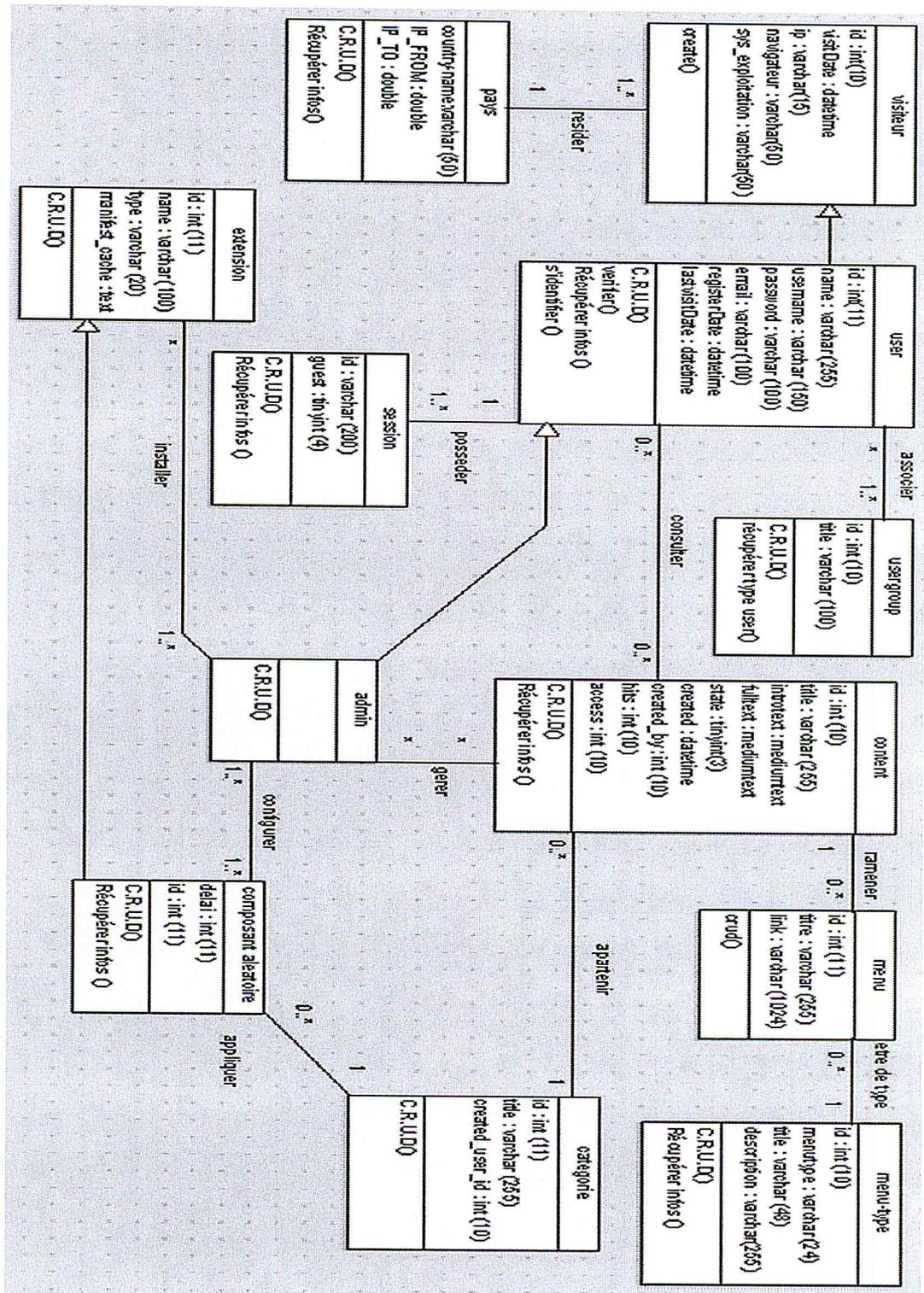


Figure 25 : Diagramme de classe - conception -

4 Conception

Cette partie traite les nouvelles fonctionnalités ajoutées au système afin d'atteindre nos objectifs pour résoudre les problèmes posés.

4.1 Diagramme de classe

Dans cette partie vous trouvez un diagramme similaire à ceux des précédents mais il y a des classes en plus, ainsi que des diagrammes décrivant les fonctionnalités supplémentaires (Figure 25).

De plus pour le diagramme de classe précédent, nous avons ajouté de nouvelles tables afin que nous puissions atteindre les objectifs visés. Parmi ces tables, nous avons une table visiteur qui est considérée comme une classe mère de user, elle est caractérisée par : adresse IP, date de visite, un système d'exploitation, un navigateur. Grâce à cette adresse IP, il sera possible de déterminer le pays de ce visiteur.

Une table pays qui contient un nom de pays et un intervalle traduit en deux attributs IP-from et IP-to qui servent à déterminer le pays en fonction d'IP du visiteur. Pour arriver à remplir cette table, nous avons utilisé une base de données (ip-to-country[16]), l'utilisation la plus populaire de cette base de données consiste à localiser le pays d'origine d'un visiteur à l'aide des intervalles basés sur les adresses IP.

Une table composant aléatoire qui hérite de la classe extensions et qui contiendra la configuration exprimée en temps (délai) de rafraîchissement de la page, c'est là précisément où nous voulons utiliser la notion d'aléatoire, ainsi qu'un identifiant qui sera attribué à chaque configuration.

4.2 Diagramme de séquence

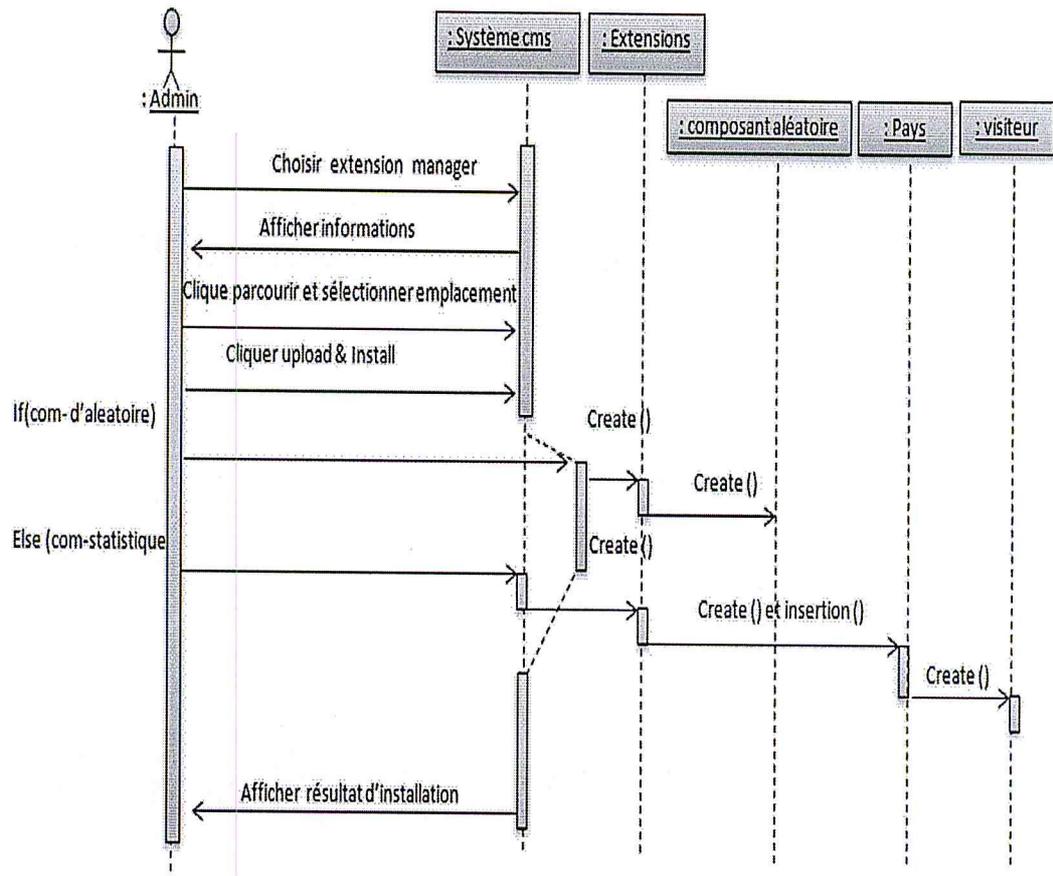


Figure 26 : Diagramme de séquence - installation des extensions -

Dans le cas d'une nouvelle installation d'un composant parmi les notre, le nom de ce dernier sera ajouté à la table extensions. Ainsi, de nouvelles tables seront créées dans la base de données. Par exemple, dans le cas de composant de statistiques, la table pays sera créée ensuite elle sera remplie dès l'installation.

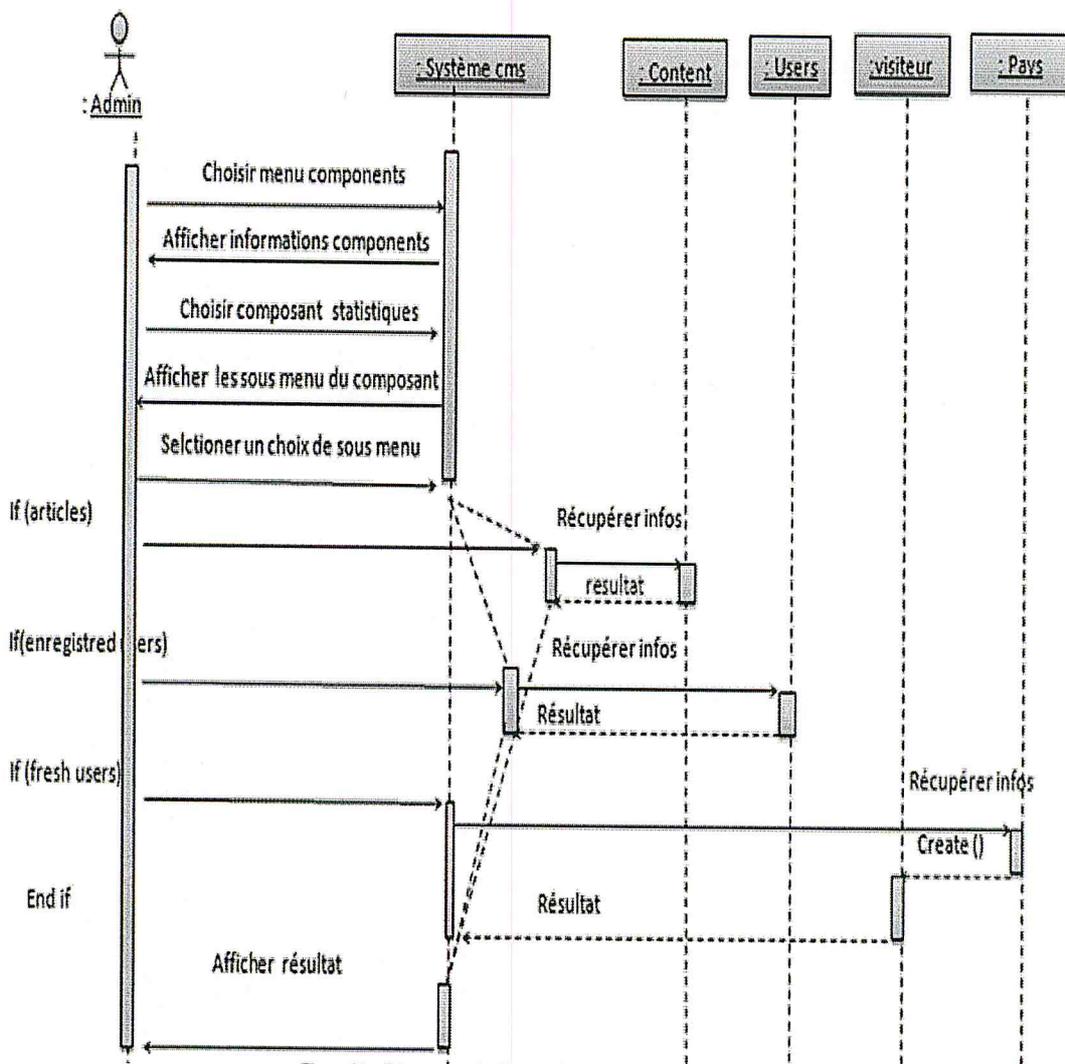


Figure 27 : Diagramme de séquence - manipulation composant statistique -

Après un succès dans l'installation d'extension (dans notre cas composant de statistique), ce composant sera prêt à être utiliser. il suffit que l'administrateur choisit le menu *component*, le système affichera les différents composants installés, ensuite il lui faut sélectionner composant de statistique, le système va afficher un sous menu de ce dernier qui contiendra les différentes utilités, l'administrateur va choisir l'une d'elle, ensuite le système va consulter sa base de données afin de collecter les informations (détails) puis les afficher .

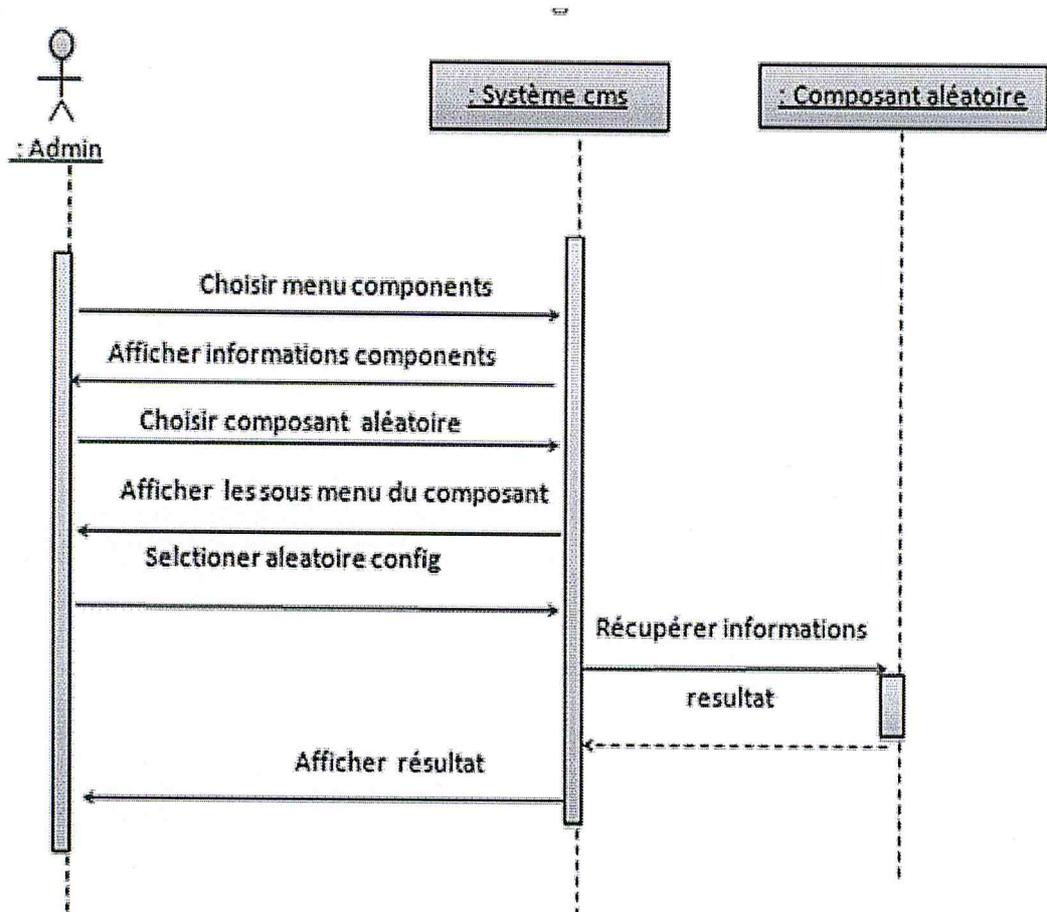


Figure 28 : Diagramme de séquence - manipulation composant aléatoire -

Même chose pour le composant d'aléatoire, après un succès d'installalion, il sera prêt à être utilisé. Dès que l'administrateur choisit le composant d'aléatoire à partir du menu *components*, le système va apparaitre un sous menu de ce composant. Après la sélection de configuration d'aléatoire le système va consulter la base de données afin de récupérer les informations puis les afficher.

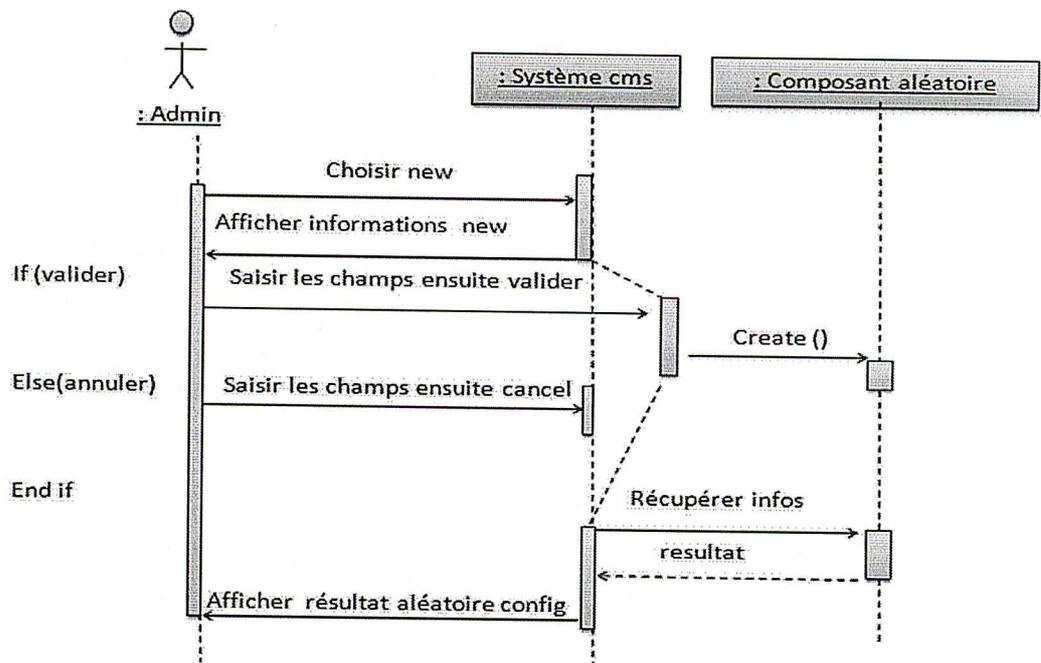


Figure 29 : Diagramme de séquence - création de nouvelle configuration d'aléatoire -

Le temps où l'administrateur se trouve sur la page de configuration d'aléatoire, deux cas peuvent être traités : la création d'une nouvelle configuration, ou bien celle d'une modification relative à une ancienne configuration.

Si l'administrateur traite le premier cas, il doit sélectionner le bouton *New* pour pouvoir créer une nouvelle configuration, le système va afficher une page contenant les informations relatives à une nouvelle configuration, après la saisie de ces informations l'administrateur doit confirmer et sauvegarder cette configuration en sélectionnant *Save*, A cette étape le système établit une connexion avec la base de données afin d'enregistrer ces données de configuration, ensuite un résultat de confirmation d'enregistrement sera transmis à l'administrateur qui a aussi la possibilité d'annuler la configuration avant de sauvegarder ou quitter l'espace d'une nouvelle configuration avec le bouton *cancel*.

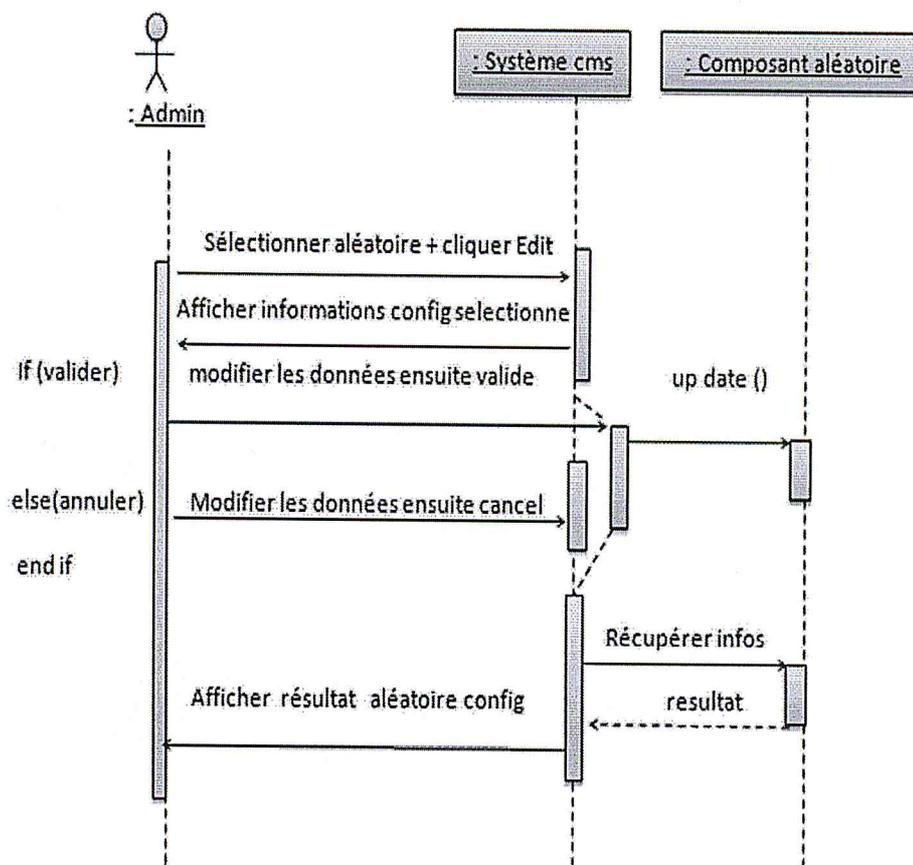


Figure 30 : Diagramme de séquence - modification d'une configuration d'aléatoire -

D'autre part, si l'administrateur veut réaliser une modification relative à une configuration d'aléatoire, il n'a qu'à sélectionner la configuration à modifier sur la page *aléatoire config*, ensuite cliquer sur le bouton *Edit*, le système va afficher les informations de cette configuration, l'administrateur fait ses modifications ensuite il les sauvegarde en cliquant *Save*, dans ce cas le système consulte la base de données pour enregistrer ces changements. L'administrateur peut annuler la modification avant de sauvegarder ou quitter l'espace d'une modification de la configuration en question avec le bouton *cancel*.

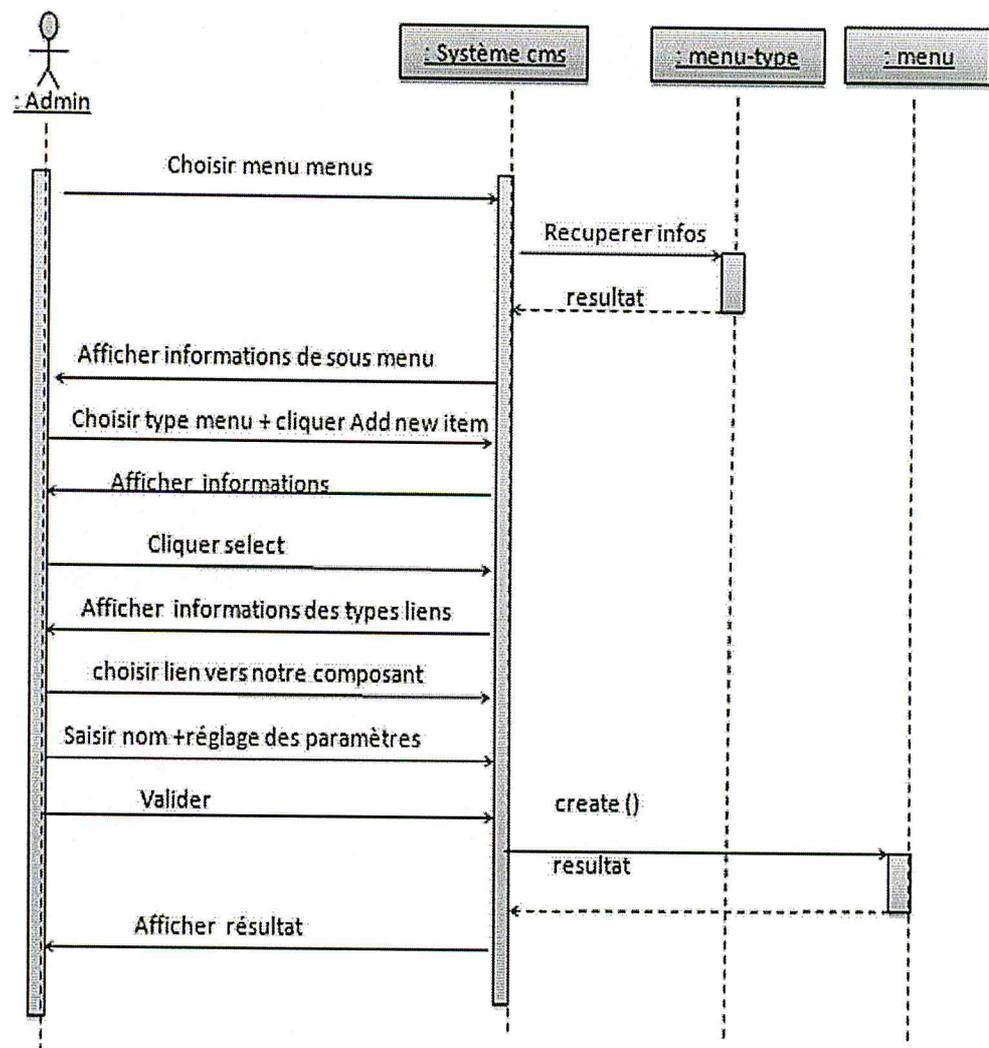


Figure 31 : Diagramme de séquence - ajouter un lien d'un menu -

La majorité des sites prend en considération la gestion des menus comme une clé qui sert à faciliter aux visiteurs le déplacement dans le site, cela permet d'accéder aux données d'une façon rapide. C'est le cas de nos composants, l'administrateur doit créer des liens de menu qui vont apparaitre sur la partie visible aux visiteurs afin qu'ils puissent accéder aux différentes fonctionnalités réalisées par nos composants. Par exemple nous voulons avoir un lien d'un menu principal (*main menu*), tout d'abord, l'administrateur sélectionne des menus dans le menu d'administration, le système va consulter la table *menu-type*, ensuite il affichera un sous menu contenant les différents types de menus

existants. Dans notre cas l'administrateur doit sélectionner (*main menu*) ainsi que *Add new menu item*, le système affichera les informations relatives à un nouveau lien, l'administrateur a besoin de choisir un type de lien en cliquant sur le bouton *select*, le système va afficher les différents types de liens existant, l'administrateur doit choisir le lien relatif au composant qu'il veut atteindre du coté visiteurs, ensuite il remplit les autres champs afin de terminer l'enregistrement, puis il choisit le Bouton *Save* pour le valider. Le système consulte la base de données pour enregistrer les informations de ce lien, puis il transmet un message indiquant le résultat d'opération.

4.3 Diagramme d'états transition

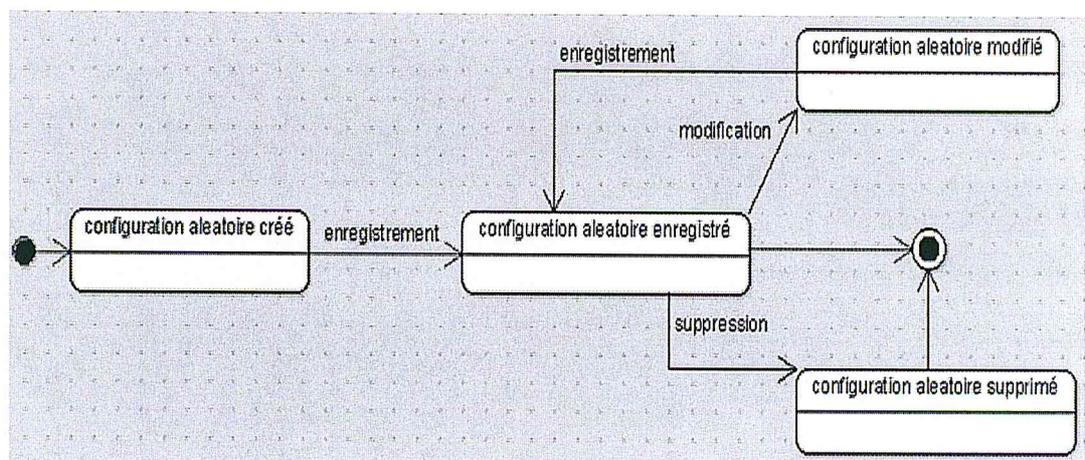


Figure 32 : Diagramme d'états transition - configurations d'aléatoire -

Après l'installation de composant d'aléatoire, il faut le configurer par la suite. L'administrateur devra créer une nouvelle configuration (créer aléatoire) en précisant les différentes valeurs des champs relatifs à cette configuration. L'administrateur doit la valider, dans ce cas son état passe à aléatoire enregistré. Au fur et à mesure, l'état de cette configuration d'aléatoire peut être changé entre aléatoire modifié afin qu'il subisse des changements de valeurs de configuration ou bien aléatoire supprimé pour détruire cette configuration.

5 Conclusion

En passant par cette étape de conception nous avons obtenu une vue exacte, claire et des idées bien organisées afin de faciliter le processus de développement qui nous sert à atteindre nos besoins.

Chapitre 5

Implémentation

d'application

1 Introduction

Dans l'étude conceptuelle, nous avons présenté notre système avec ses acteurs et cas d'utilisation. Maintenant, nous allons donner le détail de l'implémentation des différents modèles du système. En outre, la réalisation touche aussi le déploiement de l'application dans son environnement réel. Nous allons présenter l'implémentation et la mise en œuvre de cette plateforme, nous présenterons en premier le langage et le SGBD utilisés dans la réalisation, puis les interfaces de différents utilisateurs, ainsi que chacune de ses fonctionnalités.

2 Présentation des outils de développement

2.1 Le langage PHP

Ces dernières années, il n'est plus question de faire sa " page Web perso " comme c'était la mode il y a quelques années auparavant. Le HTML statique est mort depuis longtemps, tandis qu'aujourd'hui le dynamique a pris son envol avec AJAX. Plus personne ne code son site directement en HTML, mais utilise plutôt des scripts qui génèrent des pages HTML selon des informations qui sont à la base de données, ce qui introduit l'utilisation du php.

Php est un langage open source, donc libre d'utilisation et sert à produire des pages web de façon dynamique pour sites internet. Au fur et à mesure, les versions de php ont évolué, le code d'exception s'est révélé plus rapide. Php reste le langage de programmation web le plus utilisé au monde, malgré ses failles de sécurité. Php 5 permet de parcourir, manipuler un fichier XML, faire de la programmation orienté objet.

2.2 Description du serveur de base de données MySQL

MySQL est un véritable serveur de base de données SQL multiutilisateur et multithread, basé sur SQL qui est le plus populaire langage de base de données dans le monde, SQL est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information.

MySQL est idéal pour le web, suffisamment rapide et flexible pour gérer des historiques et des images, par exemple, vous pouvez utiliser MySQL pour récupérer des informations sur un produit ou stocker des informations clients sur un site web. Il a avantage d'être portable, c'est-à-dire, il est compilé sur plusieurs plateformes comme Windows, Linux, OS,...etc. de plus, il est facile à utiliser, standard (utilise SQL) et open source.

2.3 Description du serveur Web

Un serveur Web est un outil qui permet la publication d'information sur un réseau intranet ou bien internet. Celui accède à une base de données pour fournir des résultats issus de la demande de l'utilisateur.

Nous avons utilisé pour notre serveur http le serveur Apache, ce logiciel offre plusieurs avantages, parmi eux :

- les bonne performances.
- nécessite peu de ressources.
- portable sur plusieurs plateformes (Windows (98/2000/XP), linux, Nombreuse version d'Unix, MacOS X).
- gratuit.
- stable est en cours d'évolution.

3 Fonctionnement de composants sous Joomla

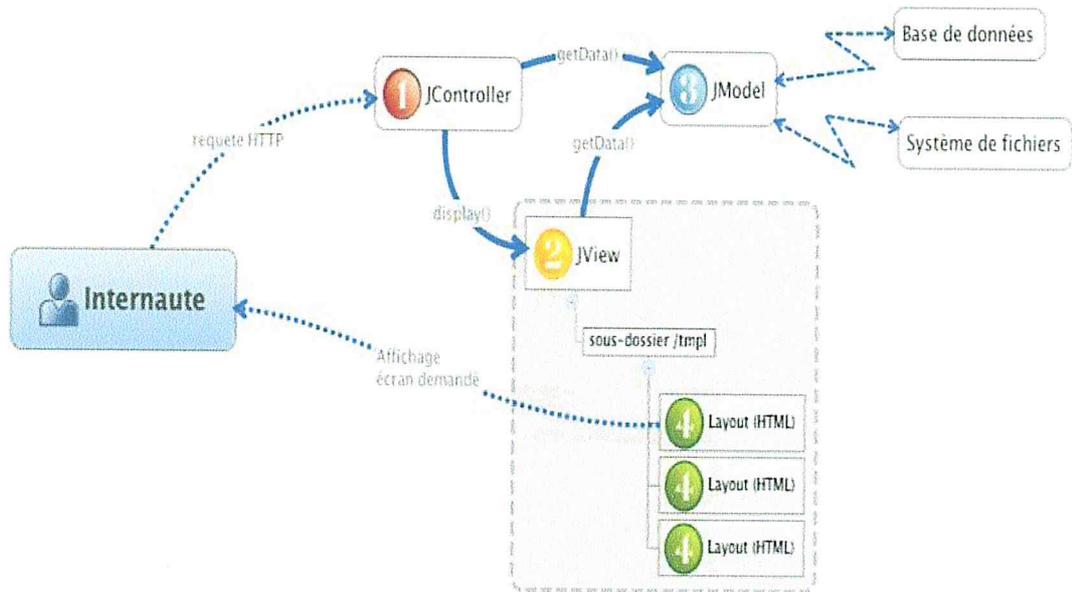


Figure 33 : Traitement d'une requête d'un internaute avec le modèle MVC [17]

Lorsque nous avons entamé le chapitre concernant le modèle MVC, nous avons constaté qu'il correspond au découpage d'un script en 3 scripts spécialisés : le Modèle de données, la Vue et le Controlleur. Seule particularité de Joomla, la vue de son MVC est découpée en une Vue (JView) et en un ou plusieurs Layouts (calque ou couche d'écran) selon ce qu'on souhaite afficher comme rendu (Liste, table, blog).

3.1 Fonctionnement de composant d'aléatoire

Le composant aléatoire qui donne la possibilité d'afficher des articles sur site d'une façon aléatoire et d'une durée choisie par l'administrateur du site.

Le déroulement du composant se passe comme suit :

- Le remplissage des paramètres de configuration du composant d'aléatoire qui sont le délai de rafraîchissement et le nom de la catégorie.
- Le contrôleur joue un rôle de lien entre le modèle et la vue qui va s'afficher sur l'écran.
- Le modèle sélectionne les articles relatifs à la catégorie choisie au paravent et

les renvoie dans des objets.

- Le script du fichier *view.html* situé dans la vue *articleAl* fait un appel au modèle pour obtenir les articles sélectionnés.
- La page *default.php* est chargée d'afficher d'une façon aléatoire l'article sur site avec un rafraîchissement de la page suivant le délai choisi.
- En dernier le navigateur affiche au visiteur l'article renvoyé.

Avec le schéma suivant nous pouvons voir le déroulement du composant aléatoire :

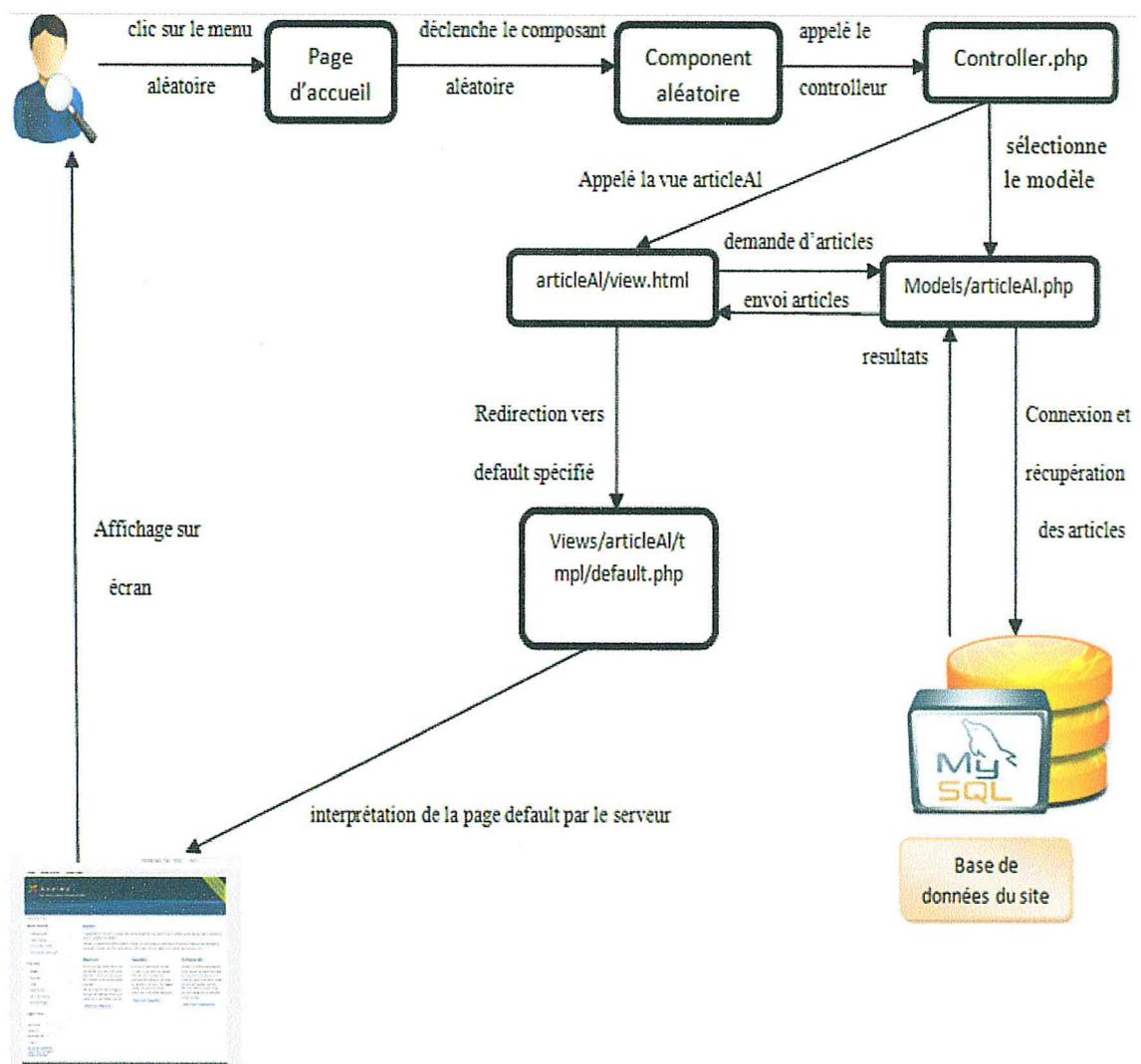


Figure 34 : Fonctionnement de composant d'aléatoire

3.2 Fonctionnement de composant de statistiques

Le schéma suivant va simplifier comment fonctionne notre composant de statistiques concernant la récupération des informations relatives au visiteur :

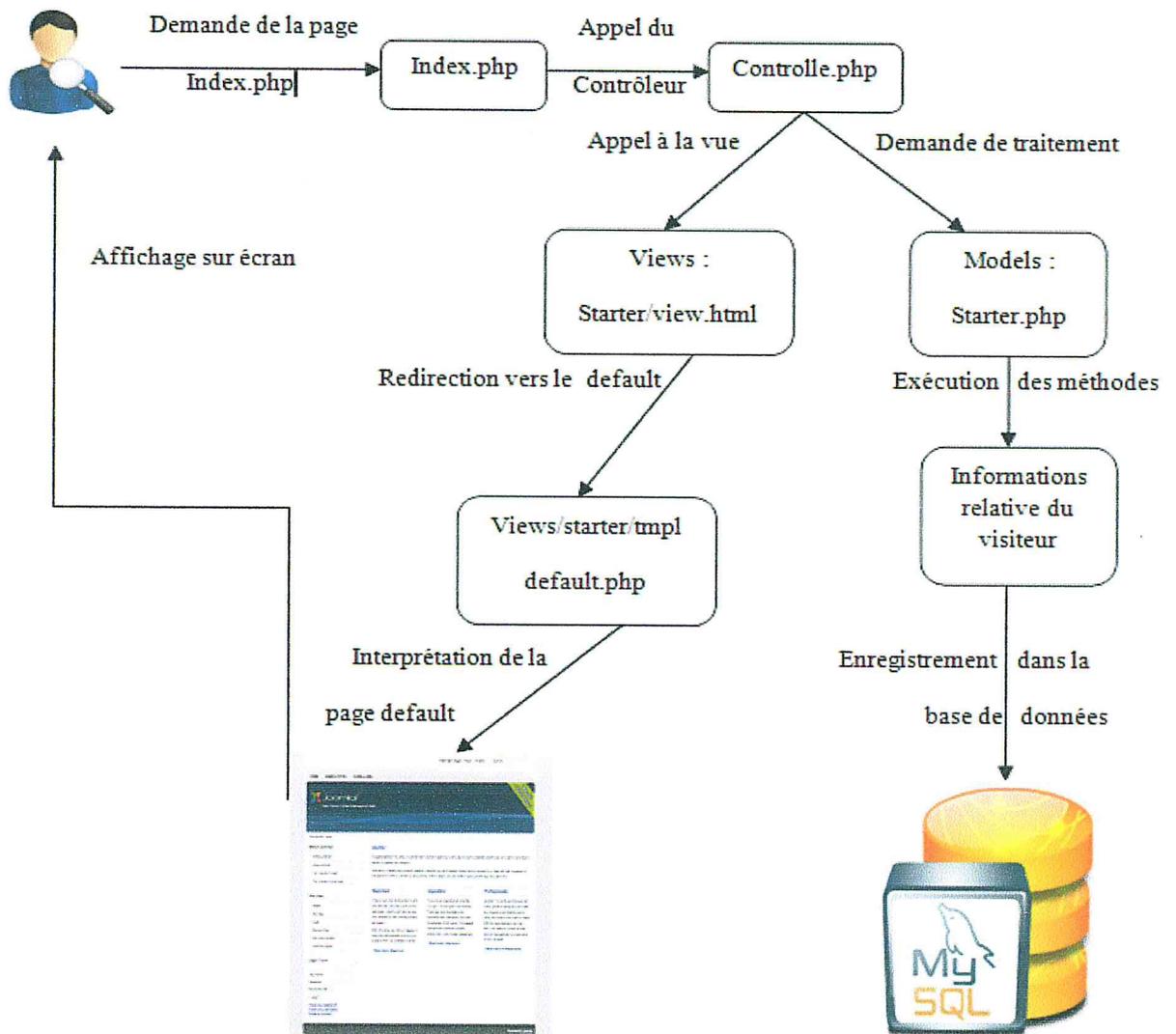


Figure 35 : Fonctionnement de composant de statistiques

Le principe du fonctionnement du composant est le suivant :

- Une fois le visiteur fait un clic sur le site, il y aura un déclenchement de plusieurs méthodes qui vont s'exécuter:
- Le contrôleur fait un appel au modèle pour exécuter la fonction *getinformations* située dans le fichier *models/starter.php*.

- L'exécution de celle-ci va permettre au serveur l'enregistrement d'informations retenues dans la table *visiteur* de la base de données du site.
- En parallèle le contrôleur demande l'affichage de la vue *starter*.
- Ce dernier va redirectionner vers la page *default.php* et la renvoie au serveur.
- Le serveur interprète la page *default.php* et la renvoie au navigateur sous forme de HTML.
- Le navigateur affiche la page d'accueil du site au visiteur.

3.2.1 Description des méthodes utilisées dans le composant de statistiques

Dans notre composant, nous avons utilisé différentes fonctions pour arriver à retenir les différentes informations relatives au visiteur ou même d'un membre inscrit dans le site.

Dans le modèle *starter.php* nous trouvons la principale fonction *getInformation* qui est chargée de récupérer les informations (adresse ip , date visite, le navigateur...etc) à l'aide des scripts PHP spéciale pour cela, nous devons citer les scripts comme suit :

- *Detectip()* : la fonction chargée pour obtenir l'adresse IP du visiteur une fois que ce dernier aura fait un clic sur le site .
- *Navigateur()* : une fonction qui récupère le type du navigateur utilisé par le visiteur.
- *Sys-exploitation()* : la fonction nous donne une idée sur le système d'exploitation utilisé par le visiteur.
- *Date()* : elle nous renvoie la date avec l'heure précise de la visite d'un internaute.
- *GetUser()*: cela est un script PHP situé dans la classe *JFactory* qui peut distinguer si le visiteur est un étranger ou bien un membre dans le site.
- *Recu-name-country(IP)* : cette dernière nous fait connaître le pays du visiteur à l'aide du paramètre *IP*. Une fois l'adresse IP obtenue par la méthode *detectip()* nous la convertissons en nombre décimal par la fonction *ipAdressNumber*. Avec la table *pays* dans la base des données du site nous pouvons connaître le pays

du visiteur.

- Après l'exécution des différentes méthodes et l'obtention des informations, nous les stockons dans la table *visiteur*.

4 Les interfaces

4.1 Composant d'aléatoire

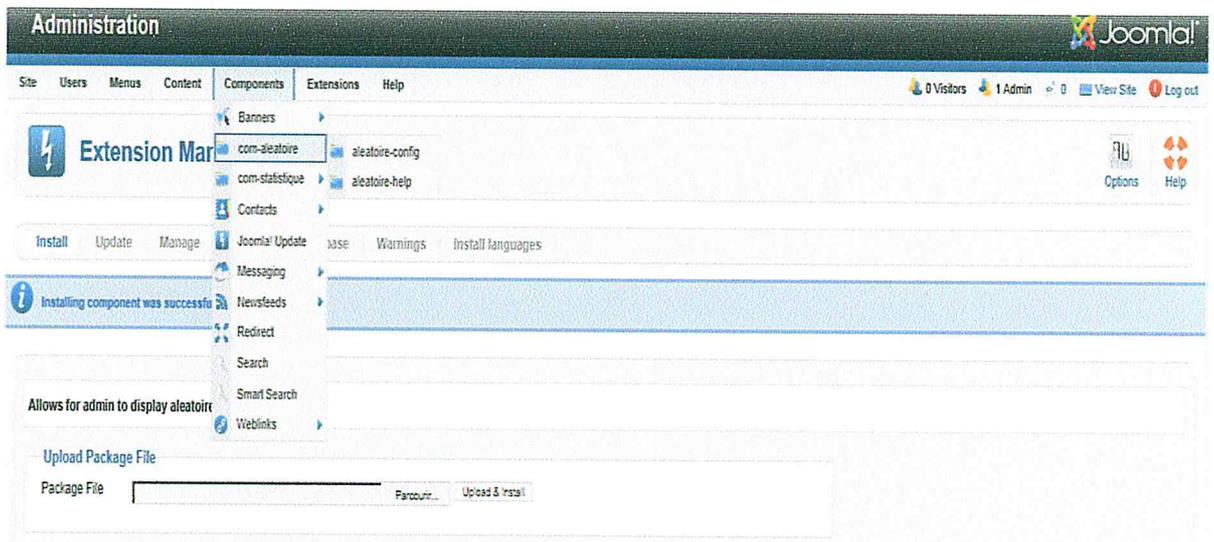


Figure 36 : Installation de composant aléatoire

Après une certaine installation avec succès, le composant d'aléatoire va apparaître dans le menu *components* dans le sous menu de ce composant. Nous avons le choix entre une configuration d'aléatoire (*aleatoire-config*) afin de gérer cette notion ou bien un *help* afin d'obtenir des informations facilitant la manipulation de ce composant ainsi qu'une démarche à suivre pour mieux réaliser nos besoin d'aléatoire .

The screenshot shows the Joomla! Administration interface. At the top, there is a navigation menu with 'Site', 'Users', 'Menus', 'Content', 'Components', 'Extensions', and 'Help'. The main content area is titled 'ALEATOIRE_HELP' and contains the following text:

ce composant est realise pour permettre de faire un affichage aleatoire des articles relative a une categorie:

etapes a suivre :

- 1- pour commencer il faut avoir deja une categorie sinon vous devez creer une
- 2- il faut avoir des articles qui appartiennent au categorie precedante sinon vous devez les creer
- 3- il faut configurer ce composant selon votre choix : composants-> Explore Aleatoire -> aleatoire_config
- * si cela est votre premiere utilisation donc vous devez creer un nouveau config : en specifiant le nom de categorie s'elle qu'on veut afficher ces article en aleatoire ainsi qu'il faut specifier le temps de rafraichissement qui est en 'second' , ensuite il est possible juste de faire un 'edit' pour le modifier.
- 4- en dernier etape il faut ajoute un lien sur le site 'front end' qui pointera ce composant : ADD New Menu Item -> type : regarder Explore Aleatoire , selectionner 'article aleatoire view' , donc les visiteurs puissent acceder a un article aleatoire en cliquant juste sur ce lien

remarques :

* en cas ou l'admin a installer ce composant et il a mal configurer , les utilisateur resvoient un msg 'cette option n'est pas encors gerer par l'admin' .
 * il faut garder toujours que ce composant est pointer a fonctionner juste avec id config '1'.

Joomla! 2.5.7
 Joomla! is free software released under the GNU General Public License.

Figure 37 : Guide d'utilisation de composant aléatoire

The screenshot shows the Joomla! Administration interface for the 'Aleatoire Config' component. The page title is 'Aleatoire Config'. There are three action buttons: 'New', 'Edit', and 'Trash'. Below the buttons is a table with the following data:

ID	categorie name	delai
1	el hadith	50
2	el souna	60

Joomla! 2.5.7

Figure 38 : Configuration d'aléatoire

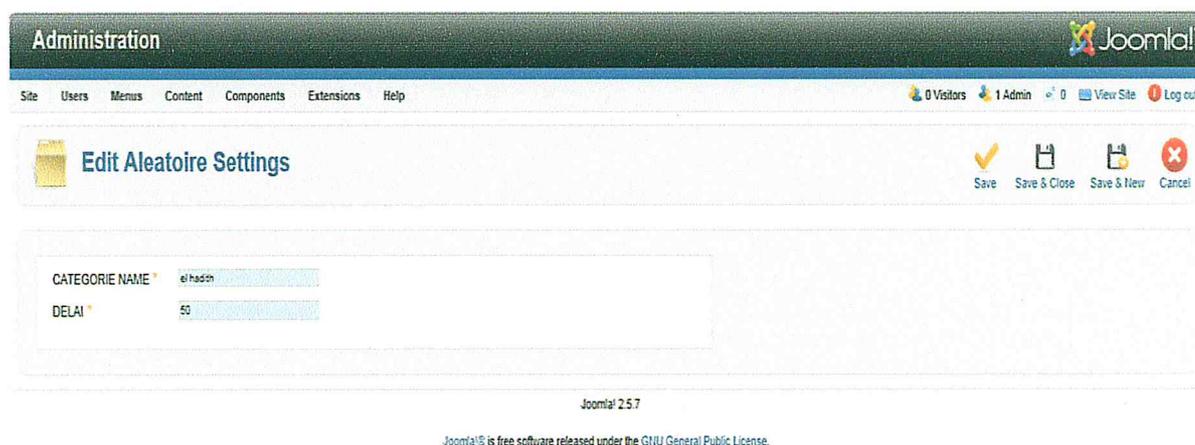
Cette partie est relative à l'affichage des différentes configurations d'aléatoire existantes. Elle contient aussi des différents boutons facilitant la gestion des configurations d'aléatoire : un bouton *New* permet de créer une nouvelle configuration, un autre *Édit* permet de modifier une configuration et un autre *Trash* permet de supprimer une configuration.



The screenshot shows the Joomla! Administration interface. At the top, there is a dark green header with the word 'Administration' on the left and the Joomla! logo on the right. Below the header is a navigation menu with items: Site, Users, Menus, Content, Components, Extensions, and Help. On the right side of the header, there are statistics: 0 Visitors, 1 Admin, 0, View Site, and Log out. The main content area is titled 'Add Aleatoire Settings'. It features two input fields: 'CATEGORIE NAME' and 'DELAI'. To the right of these fields are four buttons: 'Save' (with a checkmark), 'Save & Close' (with a floppy disk icon), 'Save & New' (with a floppy disk and plus icon), and 'Cancel' (with an 'X' icon). At the bottom of the page, it says 'Joomla! 2.5.7' and 'Joomla! is free software released under the GNU General Public License.'

Figure 39 : Nouvelle configuration d'aléatoire

Cette partie est responsable de la création d'une nouvelle configuration d'aléatoire. Nous trouvons deux paramètres qui contrôlent cette notion d'aléatoire : un nom de catégorie pour laquelle nous voulons afficher l'un de ces articles ainsi qu'un second paramètre *délai* qui exprime le temps de rafraichissement de l'article, délai est exprimé en second. Après le remplissage des champs de paramètres nous devons enregistrer cette configuration avec l'un des boutons de gestion.



The screenshot shows the Joomla! Administration interface. At the top, there is a dark green header with the word 'Administration' on the left and the Joomla! logo on the right. Below the header is a navigation menu with items: Site, Users, Menus, Content, Components, Extensions, and Help. On the right side of the header, there are statistics: 0 Visitors, 1 Admin, 0, View Site, and Log out. The main content area is titled 'Edit Aleatoire Settings'. It features two input fields: 'CATEGORIE NAME' with the value 'el hadon' and 'DELAI' with the value '50'. To the right of these fields are four buttons: 'Save' (with a checkmark), 'Save & Close' (with a floppy disk icon), 'Save & New' (with a floppy disk and plus icon), and 'Cancel' (with an 'X' icon). At the bottom of the page, it says 'Joomla! 2.5.7' and 'Joomla! is free software released under the GNU General Public License.'

Figure 40 : Modification d'une configuration d'aléatoire

Dans cette partie nous pouvons attribuer des modifications propre à une configuration, il suffit juste de redéfinir les paramètres que nous souhaitons modifier, puis enregistrer ces modifications de configuration avec l'un des boutons de gestion.

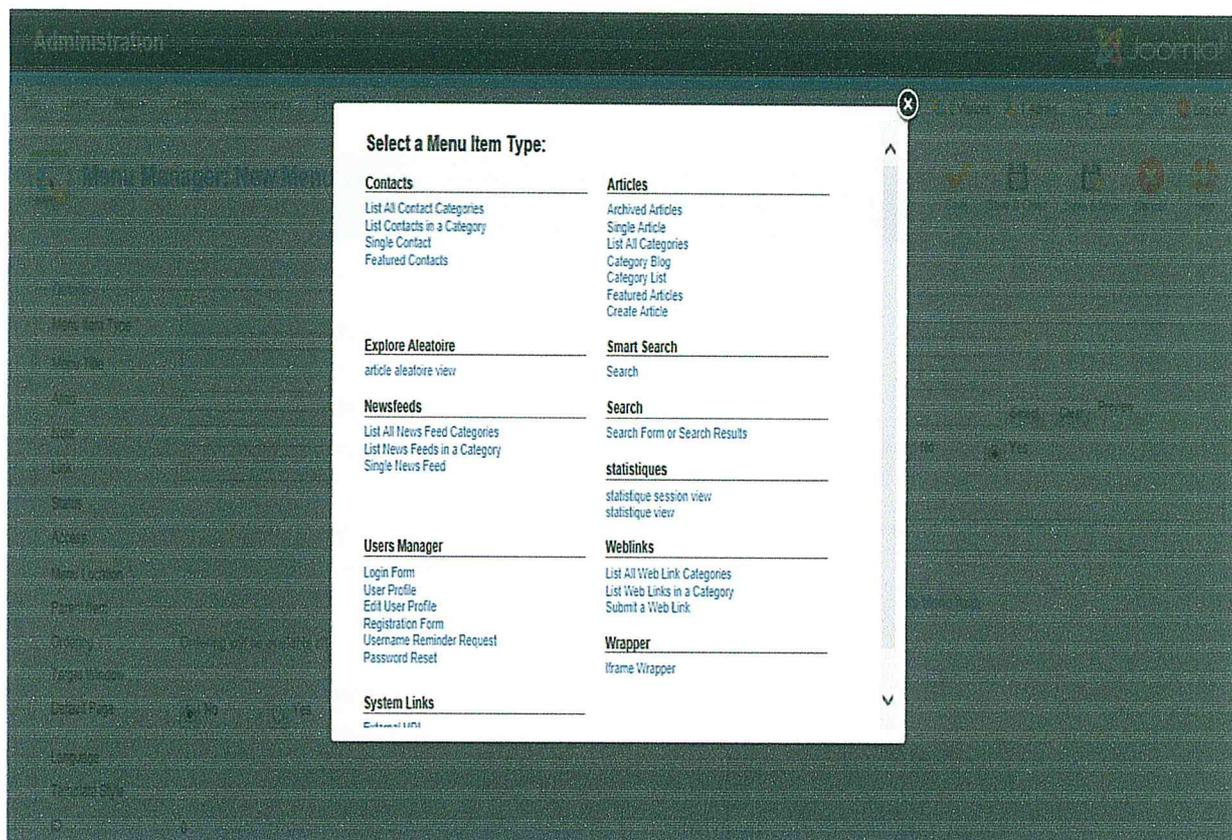


Figure 41 : Nouveau lien vers composant d'aléatoire

Pour permettre aux visiteurs de profiter de la fonctionnalité d'aléatoire il faut qu'un lien d'un menu soit établi vers cette configuration de ce composant, donc nous devons sélectionner un type de lien qui pointe vers notre composant d'aléatoire (*article aléatoire view*).

4.2 Composant de statistiques

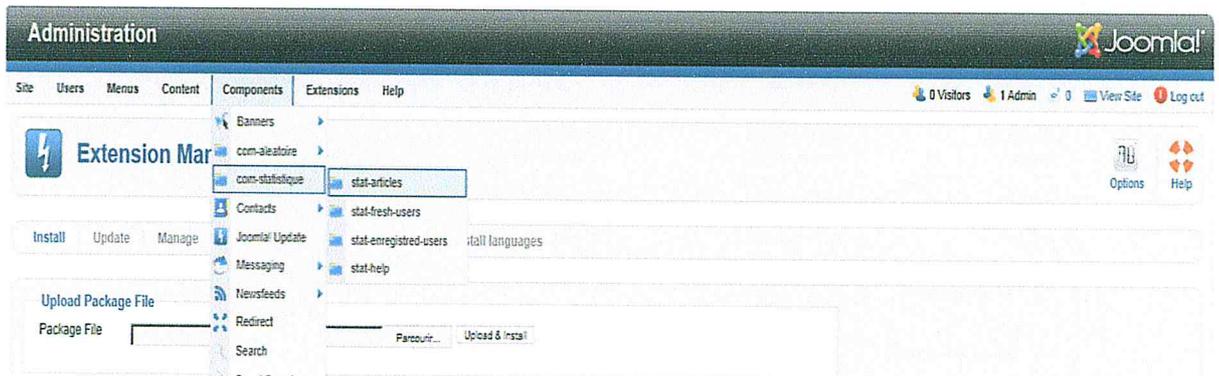


Figure 43 : Sous menu de composant de statistiques

Après une installation avec succès du composant statistiques, nous pouvons l'apercevoir sur le menu composants et il sera prêt à être utilisé. Dans le sous-menu de ce composant nous pouvons distinguer des différentes fonctionnalités de statistiques :

- i. Statistiques d'articles (*stat-articles*) : récupère toutes les informations relatives aux articles.
- ii. Statistiques des utilisateurs enregistrés (*stat-enregistred-users*) : récupère toutes les informations des utilisateurs enregistrés.
- iii. Statistiques des utilisateurs actuels (*stat-fresh-users*) : récupère toutes les informations des utilisateurs actuels, que se soit des visiteurs ou de membres.

Nous trouvons aussi un *help* afin d'obtenir des informations facilitant la manipulation de ce composant de statistiques.

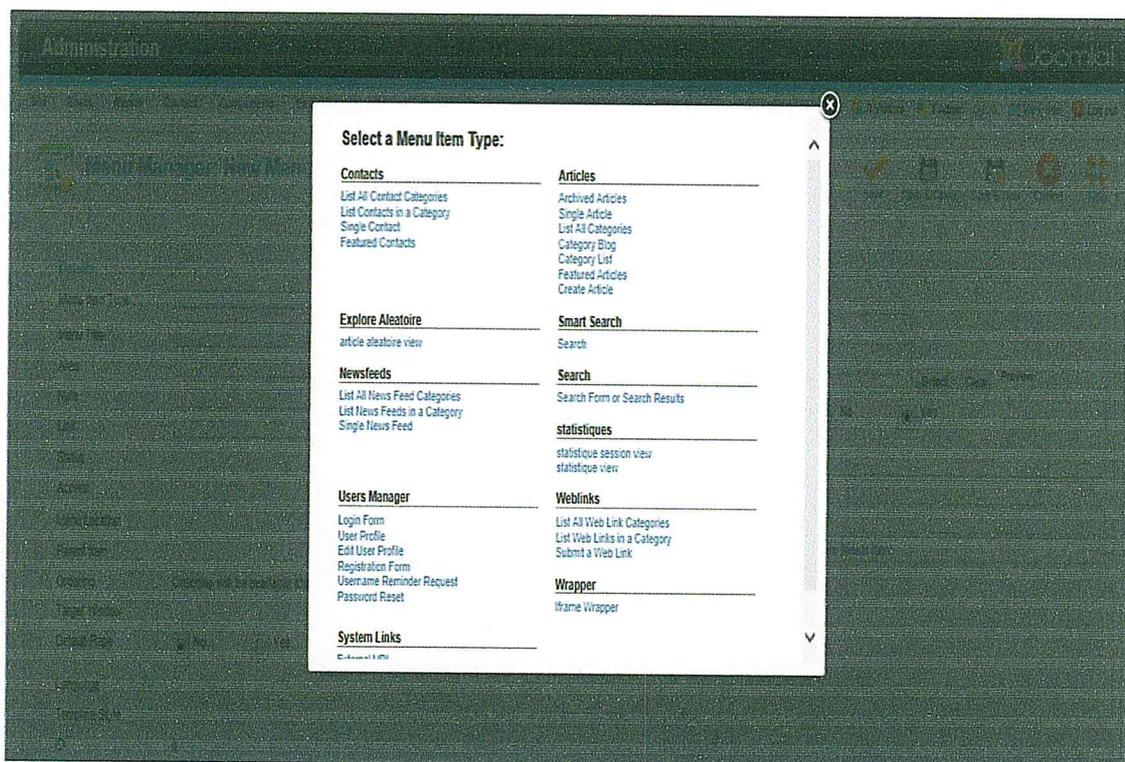


Figure 47 : Choisir un type de lien vers composant de statistiques

nous pouvons donner la chance aux visiteurs du site sur la partie visible afin d'en profiter ces fonctionnalités de statistiques à l'aide des liens de menu établit par l'administrateur afin de pouvoir pointer vers des articles spécifiques propres au composant de statistiques, par exemple :

The screenshot displays a Joomla! frontend dashboard. On the left, there is a 'Main Menu' with links for Home, ip session, aleatoire hadith, and statistiques des articles. Below the menu is a 'Login Form' with fields for User Name and Password, a 'Remember Me' checkbox, and a 'Log in' button. There are also links for 'Forgot your password?', 'Forgot your username?', and 'Create an account'.

On the right side, there are three sections:

- Global Statistic :** A link to view global statistics.
- Visitors :** A link to view visitors, with a sub-link indicating 'en ce moment il existe : 2 visiteurs'.
- Most 6 View Articlcs :** A link to view the most viewed articles.

Below these sections is a table showing the most viewed articles:

article title	created-by	publish-up	categorie name	hits
أخباريت عن التحلي بالصديق وترتك الخب	Super User	2013-07-01 15:22:45	el hadith	8
الحياء	Super User	2013-07-01 15:24:04	el hadith	5
الوفاء بالوعد	Super User	2013-07-01 15:24:42	el hadith	3
article 1	Super User	2013-06-26 09:17:12	Uncategorised	1
article 2	Super User	2013-07-01 18:42:07	Uncategorised	0

Figure 48 : Statistiques globaux sur frontend atteint a partir d'un lien d'un menu

5 Conclusion

Le processus de développement est un processus difficile à réaliser et qui nécessite des connaissances approfondies dans le domaine. Durant notre réalisation, nous avons rencontré des problèmes reletifs à la maitrise de modèle MVC et sa logique de programmation ce qui influence négativement sur la compréhension du système MVC Joomla. Cela rend le processus d'adaptation aux étapes de programmeurs difficiles, par contre avoir une bonne conception nous a permis de mieux suivre de plus proche notre implémentation et d'avoir une vue globale et claire qui nous a amené à des résultats qui arrivent à satisfaire nos besoins.

Conclusion générale :

Un processus de conception et de réalisation de fonctionnalités qui assurent l'extensibilité d'un système de gestion de contenu est un travail difficile qui nécessite de gros efforts de recherche ainsi que de patience afin d'obtenir des bons résultats.

Durant notre travail nous avons présenté l'aspect d'un CMS et son importance dans le domaine web, ensuite nous avons limité notre étude sur CMS Joomla qui satisfait des besoins spécifiques de chaque utilisateur, par la suite, nous avons présenté le concept de modèle MVC et son rôle dans le développement et la réalisation des grands projets.

Nous avons ensuite entamé une étude conceptuelle de notre système où nous avons mis en évidence les besoins des différents acteurs et déroulement des différents processus ainsi que l'aspect technique du système.

A la fin, nous sommes parvenus à réaliser des extensions qui satisfont nos besoins. L'une des extensions permet une gestion assez complète des statistiques avec différentes fonctionnalités telles que : afficher les adresses IP de nos visiteurs, leur pays et beaucoup d'autres informations concernant le contenu Joomla. Une seconde extension permet de gérer la notion d'aléatoire des articles, ce qui constitue une nouveauté dans le système de notre site.

Sources

Références webographie

- [1] Site de - developpez.com -, Créer et administrer en toute zénitude un site Web avec Joomla!(consulté entre le 10 et 30 avril) < <http://phc.devepoppez.com/pours/joomla/?lage=concepts> > [en ligne]
- [2] Site de - joeltrebi.wordpress.com -, Qu'est ce que JOOMLA? (consulté entre le 1 et 10 avril) < <http://joeltrebi.wordpress.com/iag/joomla-creer-defntition/> > [en ligne]
- [3] Site de - wikipedia.org -, Joomla! (consulté entre le 15 et 20 mars) < <http://fr.wikipedim.org/wiki/Joomla!> > [en ligne]
- [4] Site de - seonul.com -, Les nouveautés de Joomla! 2.5 (consulté entre le 3 et 20 février) < <http://www.seonul.com/2012/01/lts-nouveautes-de-joomla-25.html> > [en ligne]
- [5] Stéphan Cammarata, Document d'accompagnement de la formation Joomla!, Académie de Strasbourg,(2012)
- [6] Site de - systrio.fr -, ACL, comment personnaliser la gestion des droits utilisateurs ? (consulté entre le 1 et 5 mars) < <http://www.systrio.fr/blog-systgio/item/27-jopmla-16-acl-comment-personnaliser-la-gestion-des-droits-utilisateurs-> > [en ligne]
- [7] Site de - joomla.org -, Extension types (général definitions) (consulté entre le 10 et 20 février) < [http://docs.joomla.org/Extension_types_\(teneral_definigions\)](http://docs.joomla.org/Extension_types_(teneral_definigions)) > [en ligne]
- [8] Site de - joomla.org -, Archived:CMS Architecture in 1.5 and 1.6 (consulté entre le 15 et 20 mars) < http://docs.joomia.org/Archlved:CMS_Architecture_in_1.5_and_1.6 > [en ligne]
- [9] Site de - support-joomla.com -, Templates Charte graphique (consulté entre le 1 et 25 avril) < <http://www.support-joomla.com/construire-son-site/les-templates> > [en ligne]

- [10] Site de - [templatemonster.com](http://blog.templatemonster.com/2012/01/12/joomla-2-5-new-features/) - Two And A Half of Joomla New Features of Joomla 2.5 (consulté entre le 5 et 10 avril) < <http://blog.templatemonster.com/2012/01/12/joomla-2-5-new-features/> > [en ligne]
- [11] Site de - [cocoate.com](http://cocoate.com/fr/Jtomla!%201.5%20-%20Fran%C3%A7ais/Visite%20guid%C3%A9e%20du%20sioe%20d%E2%80%99exemple/partie-publique-frontend) -, Partie publique Frontend (consulté entre le 13 et 20 avril) < <http://cocoate.com/fr/Jtomla!%201.5%20-%20Fran%C3%A7ais/Visite%20guid%C3%A9e%20du%20sioe%20d%E2%80%99exemple/partie-publique-frontend> > [en ligne]
- [12] Chuck Lanham James Kennard, Mastering Joomla! 1.5 Extension and Framework Development, Birmingham,(2007), p.12-560
- [13] Site de - [developpez.com](http://julien-ppuli.developpez.com/tutoniels/pha/mvc-cortroleur/) -, Le modèle MVC et le controleur sous PHP (consulté entre le 10 et 25 avril) < <http://julien-ppuli.developpez.com/tutoniels/pha/mvc-cortroleur/> > [en ligne]
- [14] Site de - [developpez.com](http://tahe.developpez.com/web/php/mvc/?page=page_1) -, Méthodologie de développement MVC d'une application PHP (consulté entre le 10 et 25 avril) < http://tahe.developpez.com/web/php/mvc/?page=page_1 > [en ligne]
- [15] Site de - [torkiljohnsen.com](http://www.torkiljohnses.com/wp-content/uploads/2010/07/joomla_1.6_databanh_ochema.png) -, diagramme de classes joomla (consulté entre le 10 et 20 mars) < http://www.torkiljohnses.com/wp-content/uploads/2010/07/joomla_1.6_databanh_ochema.png > [en ligne]
- [16] Site de - [tonyarchambeau.com](http://tonyarchambeau.com/blog/329-connaître-pays-adresse-ip/) -, Connatre le pays d'une adresse IP (consulté entre le 10 et 25 avril) < <http://tonyarchambeau.com/blog/329-connaître-pays-adresse-ip/> > [en ligne]
- [17] Site de - [joomla.fr](http://dev.joomla.fr/tutoriels/item/56-creation-dun-composant-mvc-hello-world) -, Création d'un composant MVC - Hello World (consulté entre le 10 et 30 fevrier) < <http://dev.joomla.fr/tutoriels/item/56-creation-dun-composant-mvc-hello-world> > [en ligne]

Programmes utiles

- **analyse et conception**
 - Microsoft Office Word
 - AgroUML
- **traitement de textes :**
 - Microsoft Office Word
 - LaTeX
- **Design et traitement d'images :**
 - Paint
 - Microsoft powerPoint
- **Codage :**
 - Adobe Dreamweaver
 - Notepad++
- **Autres :**
 - Internet explorer
 - WampServer
 - Joomla
 - Prezi

Glossaire

CMS : Système de Gestion de Contenu (SGC), famille de logiciels destinés à la création et à la mise à jour dynamique de site web ou d'application multimédia.

Joomla : CMS libre, open source et gratuit écrit en PHP et utilise une base de données MySQL.

PHP : Langage de programmation principalement utilisé pour créer des pages Web dynamiques.

MYSQL : Système de gestion de bases de données (S.G.B.D.), basé sur le SQL.

Open Source : Signifie source ouverte ou code source libre , s'applique aux logiciels dont la licence permet la libre distribution, l'accès au code source et les travaux dérivés.

Framework : Ensemble de composants qui serent à créer l'architecture et les grandes lignes d'une application. Vous pouvez le voir comme une boîte à outils géante, conçue par un ou plusieurs développeurs et mise à disposition d'autres développeurs, afin de faciliter leur travail.

MVC : C'est un modèle permet d'architecturé un projet.

Backend : Espace sur le site invisible aux visiteurs, généralement réservé pour l'administrateur du site.

Frontend : Espace sur le site, visible par tous les visiteurs. nous trouvons également dedans la partie réservée aux membres s'il existe.

Miscellaneous : Terme anglais, composée de ou contenant une variété de choses, mélangé.

XML : Est un langage informatique de balisage générique qui permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun de leur vocabulaire et leur grammaire.

XSLT : Est un langage de transformation XML, il permet de transformer un document XML dans un autre format comme : PDF ou encore HTML afin d'être affiché comme une page web.

XSL : Est un fichier qui décrit comment doivent être transformés les documents XML basés sur un même schéma.

Jason : C'est une plateforme de développement multi agent systèmes.

JavaScript : Langage de programmation de script utilisé principalement pour les pages web interactives.

AJAX : Permet de construire des applications web et des sites dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web.

fichiers plats : C'est un fichier texte de type .csv