

MA-004-194-1

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية العلوم
Faculté des Sciences

قسم الإعلام الآلي
Département d'Informatique



Mémoire de Projet de Fin d'Études

présenté par

TOUATI Abdelhak

&

ZITOUN Abderrazak



pour l'obtention du diplôme de Master en l'Informatique option Ingénierie du Logiciel

Thème

Conception et réalisation d'un générateur de charte graphique pour Joomla

Proposé par : Mr.CHERIF ZAHAR Amine

Année Universitaire 2012-2013

MA-004-194-1

Remerciements

Allah le tout puissant, qui nous a aidés à mener à terme ce modeste travail.

Monsieur Cherif Zahar Amine, pour l'honneur qu'il nous a fait en acceptant d'être notre promoteur et pour tous les conseils et l'aide qu'il nous a donnés, afin de pouvoir mener à terme ce travail.

Nous tenons à remercier l'ensemble des enseignants du département d'informatique pour avoir assuré notre formation, le long des cinq années d'études, et pour nous avoir transmis leur savoir sans réserve.

Nous tenons aussi à remercier les membres du jury pour l'honneur qu'ils nous font en acceptant de juger ce travail.

Enfin, nous tenons à remercier tous nos amis et collègues pour leur soutien moral tout au long de la préparation de ce mémoire.

Zitoun Abderrazak

Touati Abdelhak

Résumé

Les sites web actuels utilisent deux parties distinctes : le contenu et le contenant. Le contenant est composé de détails sur la structure (apparence générale) et sur l'apparence de chaque objet de cette structure : images, fonte, bouton,...

Le présent mémoire traite de la génération automatique d'une charte graphique pour un site web basé sur Joomla. La charte comprendra les deux aspects signalés : la structure et le détail de l'apparence des éléments de la structure.

L'intérêt que revêt le travail est de rendre plus aisée la génération de chartes agréables et donc d'agir sur l'apparence d'un site web basé sur Joomla sans avoir à connaître HTML, CSS, PHP.

Mots clés : joomla, CMS, Template, utilisateur, site web.



Sommaire

Introduction générale.....	1
Problématique	2
Objectif.....	2

PARTIE 1 : ETAT DE L'ART

Chapitre 1 : Le gestionnaire de contenu

Objectifs.....	5
1. Présentation de CMS Joomla	5
2. Architectures d'un CMS	5
3. Principes et technologie	7
4. Caractéristique.....	9
5. Avantages	10
6. Inconvénients	11

Chapitre 2 : Charte Graphique web (Template)

Introduction.....	12
1. Présentation des données.	12
2. Utilisation dans les systèmes de gestion de contenu web.	13
3. Présentation de l'architecture.	13
4. Exemple de générateur de Template : Artisteer.	15
4.1. Historique.	15
4.2. Définition.	15
4.3. Caractéristique.	15
4.4. Artisteer.exe et Artisteer.net.	16

Chapitre 3 : Eléments nécessaire à la génération d'un template

Introduction.....	18
1. Les fichiers nécessaires au Template.....	18
2. Création du Template.....	19
3. Création du répertoire et des fichiers.....	20
4. Renseigner le fichier templateDetails.xml.....	21
5. Modification du fichier index.php.....	23
6. Modification du fichier template.css.....	25
7. Installer notre Template.	27

PARTIE 2 : CONCEPTION ET REALISATION

Chapitre 1 : Conception

Introduction	32
1- Analyse des besoins.....	32
1.1- Diagrammes de cas d'utilisation.....	32
1.2- Explication	34
2- Description des cas d'utilisation	36
2.1- Générer modèle de site	36
2.2- Générer position.....	39
2.3- Générer style.....	42
2.4- Générer support technique de Template.....	44
3- Diagramme de classe.....	45
4- Dictionnaire de données	46
5- Le schéma relationnel Base de données de l'application.....	48
6- Conclusion.....	48

Chapitre 2 : Réalisation

Introduction.....	49
1- Les outils d'implémentation.....	49
1.1. HTML.....	49
1.2 CSS.....	49
1.3. PHP.....	50
1.4. JavaScript.....	50
1.5. WAMP.....	50
1.6. SQL.....	52
2- Présentation.....	53
2.1. Présentation de l'application.....	53

La liste des figures

Figures		P
Fig.1 :	Contenant et contenu	6
Fig.2 :	position de l'interface web d'un CMS	8
Fig.3 :	fonctionnement de CMS	9
Fig.4 :	Présentation de l'architecture	14
Fig.5 :	les fichiers de template	19
Fig.6 :	la structure d'un template	20
Fig.7 :	Description l'information dans joomla	21
Fig.8 :	le choix d'une position	22
Fig.9 :	fichier présente le template.css	25
Fig.10 :	Gestion des extensions dans joomla	27
Fig.11 :	Transfère d'un archive	27
Fig.12 :	Gestion des template dans joomla	28
Fig.13 :	Présentable d'une ligne pour mon_template	28
Fig.14 :	Une template par défaut	28
Fig.15 :	Présentable d'un site de démonstration	29
Fig.16 :	Gestion des modules pour joomla	29
Fig.17 :	Diagra Présentable de Menu principal on joomla	29
Fig.18 :	Sélectionnée la position de template	30
Fig.19 :	Présentation de votre site	30
Fig. 20 :	Page d'accueil de l'application. (Bouton pour lancer l'application)	53
Fig.21 :	Etape (Choix entre Template pris ou création d'une nouvelle).	53
Fig.22 :	Répertoire des Templates Complets pour installer directement	54
Fig.23 :	Exemple de Template complet	54
Fig.24 :	Page de saisir les informations de nouvelle Template	55
Fig.25 :	Les pages de chois des modèles des Templates.	56
Fig.26 :	Exemple de Modèle choisie	56
Fig.27 :	Exemple de Modèle choisie	57
Fig.28 :	Page pour générer le style de l'entête	57
Fig.29 :	Page pour générer le style de position 5	58

La liste des tableaux

Tableau		Pages
Tab.1 :	Structure du CMS	8

Introduction générale :

Au début des années 1990, les sites web étaient directement développés par manipulation du langage HTML de manière statique par les développeurs en employant des éditeurs de textes ou des logiciels pas assez performants, cela donnait des résultats satisfaisant du côté création notamment au niveau esthétique.

En revanche, la maintenance n'était pas nécessaire avec ses sites statiques où seuls de rares développeurs très expérimentés pouvait minimiser le temps de maintenance et pour des raisons de coût il n'y avait qu'une seule personne : le webmaster qui entretenait le site.

Le web s'améliore et le besoin de diffusion et d'interaction avec les utilisateurs augmente de façon exponentielle : le webmaster ne supporte plus la maintenance des sites à lui seul, ces derniers ne satisfaites plus les besoins de la majorité d'utilisateurs.

Pour cela, les sites dynamiques on vu le jour afin de faciliter la maintenance et la mise à jour en organisant les données dans des bases de données indépendamment de leur présentation avec une génération de pages web à la demande.

Une première solution pour séparer le contenu du contenant est apparue avec l'apparition des langages web, le contenu étant stocké dans des bases de données celle-ci est reliev aux pages à l'aide de scripts écrit dans un langage de programmation comme PHP ce qui a permis de résoudre les problèmes de maintenance et l'affichage se fait par des scripts utilisés pour la mise en page.

Cette approche apporte quand même des inconvénients : les sites ne sont valables que pour un certain style du site et si on voulait ajouter ou modifier des fonctionnalités il fallait modifier une parties ou le script en entier, ajouter d'autre adaptés à ce qui existe ou le contraire sans oublier le cout engendré.

Une deuxième alternative est apparus dans le but de créer un logiciel accessible par tous le monde et qui gère les aspects liés à la fusion du contenus au contenant. On a commencé à les concevoir par des développeurs isolés ou dans des entreprises, puits en 1999 les premier CMS fut apparaitre comme des CMS propriétaires assez chers. Deux ans après, les CMS sont devenus des logiciels libres (on cite par exemple joomla) ce qui rendra la création des applications web public facile.

Le terme Web design au sens large englobe de nombreuses compétences et disciplines différentes qui sont mises en œuvre dans le cadre de la production et la maintenance de sites web.

La maîtrise de compétences techniques telles que les langages HTML, CSS, JavaScript, PHP, la retouche d'image et bien d'autres est un impératif. Joomla n'est qu'un outil supplémentaire" dans la boîte à outils des designers web.

Un Template Joomla de qualité ne se contente pas de traiter des couleurs et du graphisme. La mise en forme et le positionnement du contenu sont tout aussi importants. Le Template est une des pièces les plus importantes d'un site Web. Il donne l'apparence, le design. Les visiteurs réguliers et les utilisateurs apprécient d'être sur un site avec un design superbe et pratique. Pensez à d'autres produits, par exemple : Artisteer.

Problématique :

Nous nous sommes principalement intéressés au problème de création de propre template pour un site web Joomla, il n'y a pas de répertoire central pour les templates gratuits et commerciaux.

Dans tous les cas, il y a beaucoup de clubs ou de professionnels produisant des templates Joomla. Il suffit de chercher sur le web et vous trouverez très certainement quelque chose qui vous convient.

Objectif :

Un Template est un modèle de mise en forme, permettant d'appliquer la même charte graphique (habillage du site, taille et forme des polices, aspect des boutons et des modules qui composent le site, ...) à tout ou partie du site. Il permet la normalisation de la présentation, de la mise en forme.

Notre objectif est de réaliser un générateur de charte graphique pour Joomla.

Organisation de mémoire :

Afin de répondre à la problématique présentée précédemment, ce mémoire est organisé comme suit :

PARTIE 1 : ETAT DE L'ART

Elle est consacrée pour les différents concepts liés à cette étude.

Chapitre 1 : Les gestionnaires de contenu.

Chapitre 2 : Charte Graphique web (Template).

Chapitre 3 : Eléments nécessaire à la génération d'un template

PARTIE 2 : CONCEPTION ET REALISATION

Chapitre 1 : Conception.

Chapitre 2 : Réalisation.

Première Partie

L'état de l'art

Objectif

Dans cette partie nous allons introduire les concepts de base des CMS, leurs architectures, caractéristiques générales, et les avantages et inconvénient. Nous avons parlé aussi d'un générateur des chartes web et par exemple nous avons pris Artisteer.

Chapitre 1 : Le gestionnaire de contenu

1. Présentation de CMS Joomla [DOC_01]

Un système de gestion de contenu (Content Management System - CMS) appelé aussi Web Content Management (WCM), est un outil qui permet la création, la validation et la publication de contenu en ligne.

Toutes ces actions sont réalisées depuis un simple navigateur Web et ne nécessitent aucune compétence technique particulière. Qu'il soit dédié à un site Internet, Intranet ou Extranet, l'outil de CMS permet d'administrer le site de manière interactive.

Du point de vue architecture, un système de gestion de contenu se compose d'un front office (votre site ou l'application Web) et d'un back-office (l'interface de gestion et d'administration du site et de ses utilisateurs).

La caractéristique principale d'un outil de gestion de contenu Web est de pouvoir séparer le contenu du contenant, autrement dit de séparer la mise en page (la présentation) et les documents, textes, images... etc (le contenu).

Ainsi, il est possible de mettre à jour l'un indépendamment de l'autre et vice versa.

2. Architecture d'un CMS [DOC_01]

- **Utilisation d'une interface Web classique :**

Les logiciels de gestion de contenu Web utilisent comme interface un navigateur Web classique. Les navigateurs Web offrent l'avantage de fonctionner dans tous les environnements et sur toutes les plateformes informatiques.

Associées à des méthodes de rangement de l'information (liens hypertextes, moteur de recherche, tris), les données pourront ainsi être gérées facilement et réutilisées à plusieurs endroits du site, évitant les ressaisies et les erreurs.

3. Principes et technologies [DOC_02]

➤ **Front Office :**

Le terme Front Office est terme d'architecture logicielle. Il désigne la partie qui prend en charge l'interface d'une application, par opposition du back office qui lui regroupe la partie gestion.

On retrouve typiquement le front office dans les sites web commerciaux qui permettent de commander en ligne. Dans ce cas, le front office correspond à la partie hypertexte (web) de l'application, donc ce qui est visible par le consommateur. Elle lui permet d'interagir avec le reste du site.

➤ **Back Office :**

Le back office est une partie d'un site web ou d'un système informatique. Elle concerne la partie qui permet d'administrer et de gérer le site. On pourra retrouver comme service par exemple :

- Modification de paramètres.
- Administration d'un forum.
- Gestion des utilisateurs.

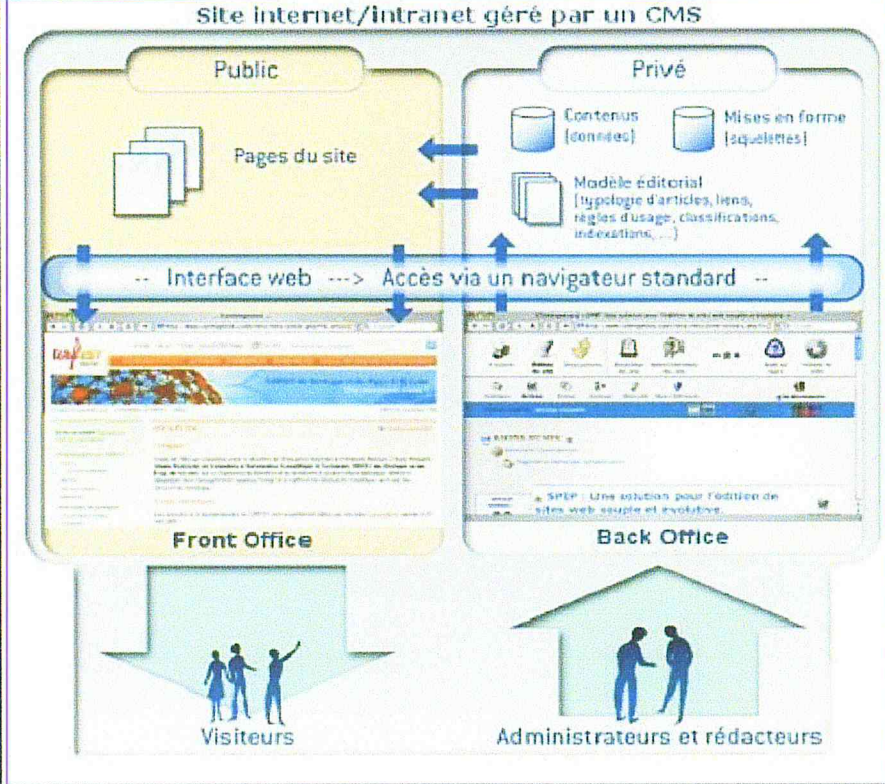
CMS	Front Office	Back Office
 <p>The diagram illustrates the architecture of a CMS. It is divided into a 'Public' section (Front Office) and a 'Privé' section (Back Office). The Front Office displays 'Pages du site' to 'Visiteurs'. The Back Office manages 'Contenus (connaiss)' and 'Mises en forme (requêtes)' using a 'Modèle éditorial'. Both sections are accessed via a standard browser through an 'Interface web'. The Back Office is used by 'Administrateurs et rédacteurs'.</p>	<p>Partie <<publique>> du site internet géré par le CMS</p> <p>Le contenu est mis en page et présenté au visiteur du site</p>	<p>Partie <<Privé>> nécessitant une authentification</p> <p>Interface permettant la gestion du contenu par les utilisateurs autorisés</p>

Tableau 1: Structure du CMS

➤ Utilisation d'interfaces Web

L'accès au Back Office se fait via un navigateur Internet classique et la gestion du contenu se fait à l'aide de formulaires Web .Ces formulaires une fois validés sont traités par le serveur qui enregistre et organise le contenu en conséquence.

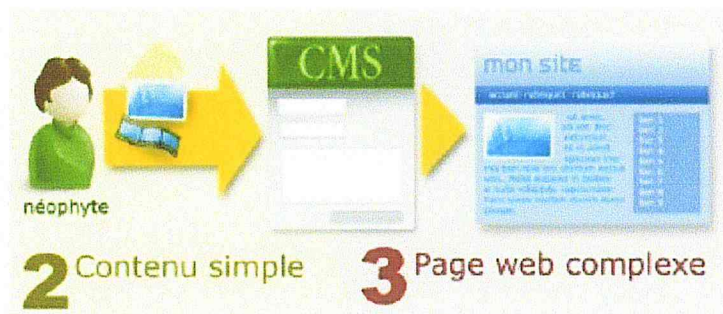


Figure 2 : position de l'interface web d'un CMS

➤ Séparation contenu/présentation

Le contenu est stocké dans une base de données mais la structure et la présentation sont définies dans des fichiers physiques de type : (modèles, templates, gabarits, squelettes...).

Lors de la consultation d'une page, le CMS intègre le contenu dans le modèle de page et affiche le résultat au visiteur et le système de mise en cache permet de ne pas calculer la page à chaque affichage.

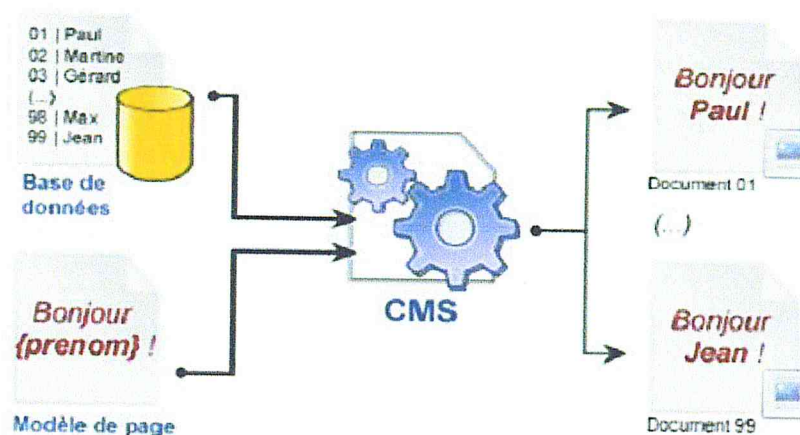


Figure 3 : fonctionnement de CMS

4. Caractéristiques [DOC_03]

Parmi les principales caractéristiques d'un CMS on peut citer:

- Facile à utiliser, ne requière pas de connaissances particulières tant pour les concepteurs du site que pour les rédacteurs et les lecteurs .
- Accès depuis un navigateur web via le réseau (internet, intranet, extranet) depuis n'importe quel point de celui-ci et avec n'importe quel système d'exploitation.
- Création et publication dans un front-office (qui est le site web) à travers un back- office fourni à chaque utilisateur selon ses droits.
- Séparation du contenu et du contenant.
- Propose une gestion de cycle de vie des documents (workflow).

5.4. Travail collaboratif

- ✓ Travail plus efficace : aucun conflit lié aux interactions.
- ✓ Le workflow assure la qualité du contenu.

5.5. Décentralisation de la gestion de l'information

- ✓ Chaque contributeur peut travailler où il veut et quand il veut.
- ✓ Un simple accès à Internet suffit pour gérer le contenu du site internet.

5.6. Contrôle de l'accès à l'information

- ✓ L'accès à l'information peut être conditionnée et limitée.
- ✓ L'historique des versions permet de connaître l'auteur de modifications.
- ✓ Les statistiques donnent des informations pertinentes sur la consultation du contenu.

5.7. Sécurité

- ✓ Les CMS ont des dispositifs permettant de résister aux attaques.
- ✓ La rigueur d'utilisation qu'imposent les CMS réduit les risques d'erreurs humaines.
- ✓ Les droits protègent les ressources.
- ✓ Fonctions de sauvegarde intégrées.
- ✓ Correctifs de sécurités régulièrement diffusés.

6. Inconvénients [DOC_04]

6.1. Code accessible à tous

- ✓ Faite régulièrement de mise à jour.

6.2. Lenteur d'accès à la base de données

- ✓ Visible à l'affichage des pages (selon l'hébergeur).

Chapitre 2 : Charte Graphique web (Template)

Introduction

Un **gabarit**, souvent nommé à tort avec les anglicismes **Template** ou **Layout**, est un patron de mise en page où l'on place images et textes. Aussi, il est souvent utilisé de manière répétitive pour créer des documents présentant une même structure.

1- Présentation des données [DOC_05]

Par extension, en informatique, un template est un moyen de séparer le fond (le contenu informationnel) de la forme (la manière dont il est présenté). Très utilisé dans la conception de sites web, un template agit comme un modèle (une structure) dans lequel seuls certains éléments sont modifiables (le contenu et le style). Cela facilite la conception et la mise à jour des sites, aussi bien sur le contenu que sur la présentation :

- changer la charte graphique du site revient à changer le template et cela met à jour toutes les pages du site.
- ajouter une page ne consiste plus qu'à en écrire le contenu.

SPIP, phpBB, Zope, Joomla, WordPress, Postnuke, PHPBoost, Phorum, KwsPHP, Ixprim, eZ Publish et de nombreux CMS utilisent nativement des gabarits.

Les moteurs de templates sont les outils utilisant les templates pour en générer les fichiers textes (HTML, code source, etc.). Chacun dispose de son propre langage de programmation limité permettant de décrire le template, à l'aide de structures de contrôle (telles que if, foreach, etc.) ainsi que de réaliser des opérations simples sur les données (récupérer la sous-chaîne d'une chaîne de caractères, supprimer des caractères indésirables pour éviter des attaques de type cross-site scripting sur un site web, etc.). Ces langages étant la plupart du temps interprétés et dynamiques, ils sont considérés comme des langages de script.

2- Utilisation dans les systèmes de gestion de contenu web [DOC_05]

De plus en plus, la présentation d'une page Web est reléguée vers les feuilles de style (CSS : Cascading Style Sheets ou feuilles de style en cascade) externes à la page. Les templates ne sont donc plus seulement censés permettre une séparation entre la structure et la présentation mais aussi une séparation entre le code HTML ou XHTML et le code PHP (ou autre langage de développement côté serveur). On obtient ainsi une structure de site possédant d'un côté les traitements de données (calculs, connexion à des bases de données, sécurité) et de l'autre le « langage » de restitution des données produites. Cette séparation facilite le travail en équipe et les mises à jour soit du côté du code PHP soit du côté du code HTML puisque chacune des deux parties peut être modifiée sans affecter l'autre.

Les templates utilisés conjointement à des feuilles de style efficaces, permettent de gérer la publication d'informations sur plusieurs supports différents.

3- Présentation de l'architecture [DOC_05]

Un template est une sorte de gabarit visuel prêt à recevoir des contenus. Il détermine les couleurs, les polices, les tailles de caractère, l'image d'arrière-plan, les espacements et la distribution des sous éléments d'une page. Il constitue donc une mise en page. Un template réunit au moins un fichier HTML décrivant la structure de la page et un fichier CSS décrivant les styles d'affichage. Un template peut être beaucoup plus complexe et rendre (afficher) des données de Joomla! Accessibles de multiples manières ou en autoriser le recyclage.

Comment ça marche ?

La page qui apparaît sur le navigateur s'appuie sur un Template.



Le template définit différentes zones dans la page autour du corps de la page

Le template s'appuie sur des feuilles de style pour l'affichage du texte sur les différentes pages du site

Dans chaque zone, on trouve différents modules. Selon les cas, les modules de la page font appel à un composant, un plug-in, un menu, un article.

La valeur des différents libellés est stockée dans des fichiers.

Les composants, plug-ins, menus, articles vont chercher leurs données dans la base de données.

4- Exemple de générateur de Template : Artisteer [WEB_01]

Artisteer est le premier et le seul produit automatisant la conception Web à créer instantanément des modèles de sites Web et de blogs au look professionnel. C'est un logiciel puissant qui fait un concepteur Web professionnel de sites Web et de blogs. Il vous permet de créer des conceptions et des modèles fantastiques de sites Web en seulement quelques minutes, sans avoir à connaître quoique ce soit sur l'édition de graphiques ou du HTML.

4.1. Définition [WEB_01]

Artisteer.net est un Système de Gestion de Contenu (CMS) basé sur WordPress et un service d'hébergement gratuit pour exécuter et publier en ligne vos sites web. Combinés, Artisteer.net et Artisteer offrent une boîte à outils complète et conviviale pour créer et gérer des projets de sites web de petite et de moyenne taille. Vous pouvez gérer à la fois la conception et le contenu de votre site web en créant des templates, en ajoutant des pages et des billets, en insérant des images et en publiant le site web modifié sur le web avec un seul logiciel.

4.2. Caractéristique [WEB_01.02]

- ✓ **Exemples de sites web prêts à l'emploi.** Vous pouvez télécharger des projets de sites web gratuits, réalisés par des concepteurs web professionnels et les modifier dans Artisteer. Il y a de nombreuses catégories de sites web avec des mises en page prêtes à l'emploi, qui peuvent servir pour vous donner des idées ou en tant que base pour vos projets de sites web. Vous avez le droit d'utiliser toutes les images et les icônes disponibles sur le site de l'éditeur, mais vous ne pouvez pas redistribuer, ou revendre une template web sous quelque forme que ce soit. De même vous n'êtes pas autorisé à proposer au téléchargement sur un autre site web la template web qu'elle soit ou non modifiée. Les exemples sont accessibles depuis la boîte de dialogue de démarrage Artisteer
- ✓ **Edition de Contenu en ligne et hors ligne**
La majorité des CMS implique que vous éditiez en ligne le contenu en utilisant un éditeur HTML simplifié, alors qu'avec Artisteer vous pouvez éditer le contenu hors ligne en utilisant le logiciel Artisteer sur votre poste de travail, ou en ligne avec Artisteer.net.

✓ Options de Mise en Page variées

Vous avez la main sur le nombre de colonnes dans vos articles et billets, leur taille et leur position.

✓ Hébergement gratuit (200 Mo)

Vous n'avez pas besoin de payer pour héberger votre site web et le publier sur le web.

4.3. Artisteer.exe et Artisteer.net [WEB_01.03]

Artisteer.exe est le logiciel que vous exécutez sur votre poste de travail pour créer vos pages web et définir un style (ou votre « charte graphique »). Lorsque vous êtes prêt, vous pouvez publier votre site web sur Artisteer.net, ce qui rend alors votre site visible sur le web. Vous pouvez éditer votre contenu web avec Artisteer, mais Artisteer sur votre poste de travail est principalement dédié à la définition du style, à savoir le rendu graphique de vos textes, images, boutons, menus et autres éléments variés qui s'affichent dans le navigateur. Par exemple, lorsque vous ajoutez du texte à une page, vous pouvez définir un certain nombre d'attributs graphiques comme la police, la taille de la police et la couleur. La plupart du temps, le « style » que vous créez est conservé dans ce que l'on appelle une template (ou un thème). Lorsque vos pages publiées sont copiées sur Artisteer.net, Artisteer envoie la template et le contenu (c'est-à-dire le texte). Sur le web, les internautes voient vos pages avec le style (template) appliqué à votre contenu.

Artisteer sur le poste de travail se réfère à Artisteer.exe, et Artisteer web à Artisteer.net. Une fois que vous avez édité votre site web, vous pouvez également créer/éditer/supprimer le contenu sur le web ou sur votre poste de travail. La différence majeure entre la version web et la version poste de travail est la suivante. Le style de votre contenu (c'est-à-dire la template) peut être créé sur votre poste de travail. Vous pouvez toujours appliquer des styles à certains éléments sur le web, mais le rendu visuel de vos pages est principalement réalisé avec Artisteer.exe. La version web est utilisée pour gérer les fonctionnalités de votre site web au travers du paramétrage côté administrateur d'Artisteer.net.

Vous pouvez créer du contenu pour votre site web (le texte et les images) aussi bien

avec Artisteer sur votre poste de travail qu'avec Artisteer.net, mais il est fortement déconseillé de les utiliser en même temps. Lorsque vous publiez votre site web à partir du poste de travail, Artisteer tente de synchroniser le contenu entre le web et le poste de travail, ce qui vous permet de visualiser les mêmes pages, billets, etc. Vous devez juste garder à l'esprit que quelque soit l'endroit depuis lequel vous effectuez des modifications (web ou poste de travail), vous devez à chaque fois republier votre site web à partir du poste de travail. Cela permet de maintenir synchronisée les deux versions. Lorsque vous enregistrez votre projet Artisteer sur le poste de travail, vous enregistrez à la fois le thème (template) et le contenu dans votre fichier projet.

Artisteer.net n'est pas uniquement un espace d'hébergement pour votre site web. Il fournit également le « backend » pour gérer votre site. Une fois votre site publié et que vous avez un compte utilisateur, vous pouvez aller sur www.artisteer.net, vous connecter, et commencer à ajouter, régler, et modérer votre contenu. Vous pouvez également ajouter ou paramétrer des fonctionnalités liées à la manière dont les internautes interagissent avec vos pages web, comme par exemple autoriser les commentaires.

Chapitre 3 : Eléments nécessaire à la génération d'un template

Introduction

Dans ce chapitre, nous avons expliqué comment créer un template Joomla.

Avant de commencer, il faut savoir que le template c'est le gabarit du site, la structure. Il détermine l'apparence visuelle du site.

C'est le template qui va déterminer la couleur du site, la taille du texte et des titres, l'affichage des pages, les positions des modules, les couleurs du texte...

En modifiant le template, on change l'aspect du site.

Avec Joomla 2.5.x, on peut attribuer un template spécifique pour une catégorie ou pour un article uniquement.

Le template est constitué au minimum d'un fichier php et d'un fichier au format xml. On peut aussi rajouter une feuille de style au format css, ainsi que des images.

Sous Joomla 1.5 on pouvait mettre directement le template dans le dossier templates et il était automatiquement reconnu par Joomla.

Sous Joomla 1.6, 1.7 et 2.5.x, cela ne fonctionne plus. On est maintenant obligé d'installer le template comme une extension. Le template devra être au format zip

1- Les fichiers nécessaires au Template [DOC_06]

Deux fichiers sont indispensables pour un Template, les fichiers **index.php** et **templateDetails.xml**.

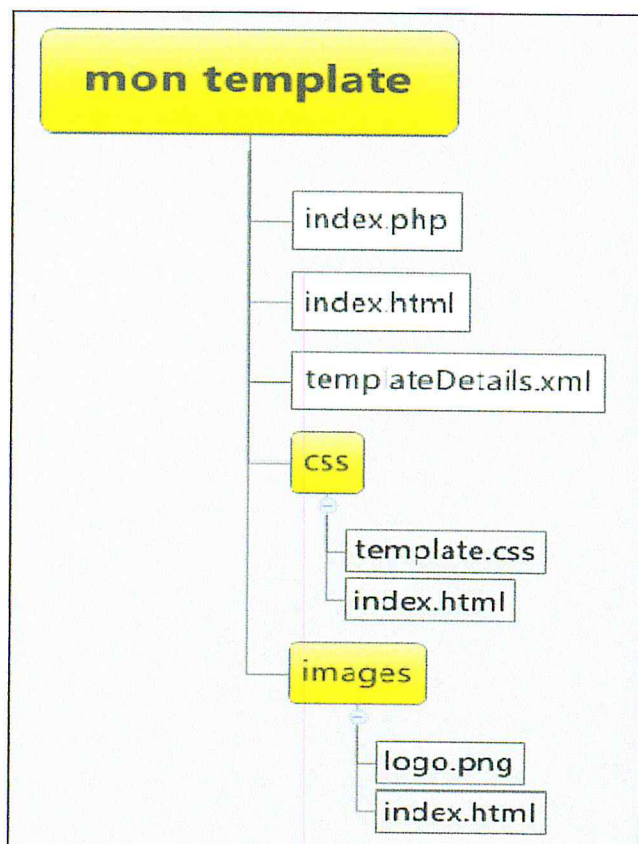
Avec juste ces 2 fichiers, un Template est opérationnel. Mais vous allez voir que l'on va utiliser d'autres fichiers optionnels, pour améliorer le graphisme et l'organisation du Template.

Voici une vue type d'un Template, une fois que l'on aura rentré tous les fichiers :

Css : contient le fichier css du Template (que l'on va appeler **template.css**). C'est dans ce fichier que l'on mettra tous les styles, couleurs...

Images : contient toutes les images du Template.

Racine : Ce dossier contient :



index.html : c'est un fichier pour éviter qu'une personne vienne fouiner dans le répertoire. Le fichier est vide. Vous pouvez y insérer le code que vous voulez.

index.php : c'est ce fichier qui contient toutes les positions du Template

templateDetails.xml : c'est le fichier qui sert à l'installation du Template et qui va indiquer les positions disponibles dans la partie des modules.

template_thumbnail.png : c'est une copie-écran du Template.

template_preview.png : c'est une copie écran plus grande. Elle est affichée pour avoir un aperçu du Template.

Figure 4 : les fichiers de template

2- Création du template [DOC_06]

Avant de se lancer dans la création du template, on va quand même en préciser les grandes lignes :

On va concevoir un template avec :

- Une partie haute, pour y insérer le logo.
- Une colonne à gauche.
- Une partie centrale pour le texte.
- Une colonne à droite.
- Une partie pied de page.
- Les colonnes s'adapteront à la taille du navigateur.
- On aura un espace sur les coins du navigateur, pour ne pas coller les colonnes au bord de l'écran.

Le template aura cette structure :

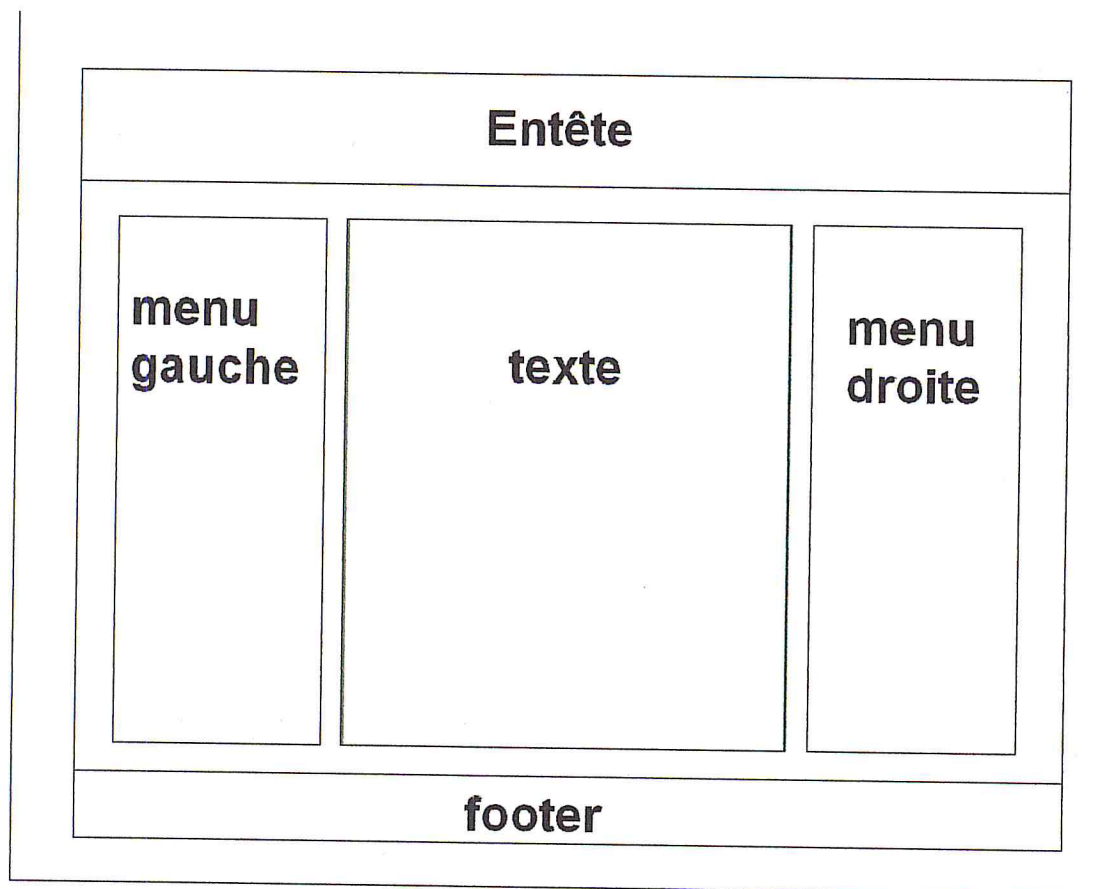


Figure 5 : la structure d'un template

3- Création du répertoire et des fichiers [WEB_08]

Pour la création de ce template que l'on va appeler `mon_template`, On va créer un dossier dans le répertoire `templates` de notre site.

1. il faut créer un dossier, avec le nom que nous voulons. Pour notre exemple on va utiliser : `mon_template` (ne pas mettre d'espace) ne pas utiliser le nom de notre template, sinon quand nous voudrions l'installer, nous aurons un message nous indiquant qu'il y a déjà un répertoire avec le même nom.

2. Dans le dossier, nous allons créer les dossiers : **images**, **css**.

Dans chacun des dossiers, nous allons créer un fichier `index.htm` : ce fichier restera vide ou nous pourrions y mettre le code que nous voudrions. Il sert juste à empêcher de consulter le contenu du répertoire.

3. à la racine de `mon_template`, il faut créer les fichiers : **index.html**, **index.php**, **templateDetails.xml**.

Maintenant, nous allons renseigner le fichier `templateDetails.xml`, sans ce fichier, on ne pourra pas installer le template.

4- Renseigner le fichier `templateDetails.xml` [WEB_08]

C'est ce fichier qui va indiquer à Joomla que c'est un template. Il comporte le nom du template, ainsi que les positions que l'on peut choisir lorsqu'on veut affecter une position à un module.

Explication de code :

`<? xml version="1.0" encoding="utf-8" ?>` : Indique que c'est un fichier xml avec un encodage utf-8

`<extension version="2.5" type="template" client="site">` : indique à Joomla que c'est un template

`<name>mon_template</name>` : indique le nom du template. Ne pas mettre d'espace. Lors de la l'installation du template, le dossier sera créé avec le nom du template. On appellera le template « mon_template ».

`<creationDate>juin 2013</creationDate>`

`<author>ZITOUN&TOUATI</author>`

`<authorEmail>TOUATI@gmail.com</authorEmail>`

`<authorUrl>http://touati.fr</authorUrl>`

`<copyright>Copyright (C) touati.fr</copyright>`

`<license>GNU General Public License version 2 ou supérieure</license>`

`<version>1.0</version>`

`<description>Template Joomla 2.5</description>`

Ce sont des options, c'est informations s'affichent dans Joomla, dans la partie gestion des templates.

Indique la date de création, nom de l'auteur, adresse courriel, site, copyright, licence, la version, la description.

Voici ce que cela donne dans Joomla :



Figure 6 : Description l'information dans Joomla

```
<files>
<folder>css</folder>
<folder>images</folder>
<filename>index.php</filename>
<filename>index.html</filename>
<filename>templateDetails.xml</filename>
</files>
```

Ce sont les fichiers qui composent le template. S'il en manque un ou si le nom n'est pas exact, vous aurez un message d'erreur à l'installation du template.

```
<positions>
<position>Entête</position>
<position>Gauche</position>
<position>Texte</position>
<position>Droite</position>
<position>Footer</position>
</positions>
```

Ce sont les positions qui seront affichées dans le menu déroulant, lors du choix d'une position dans le module

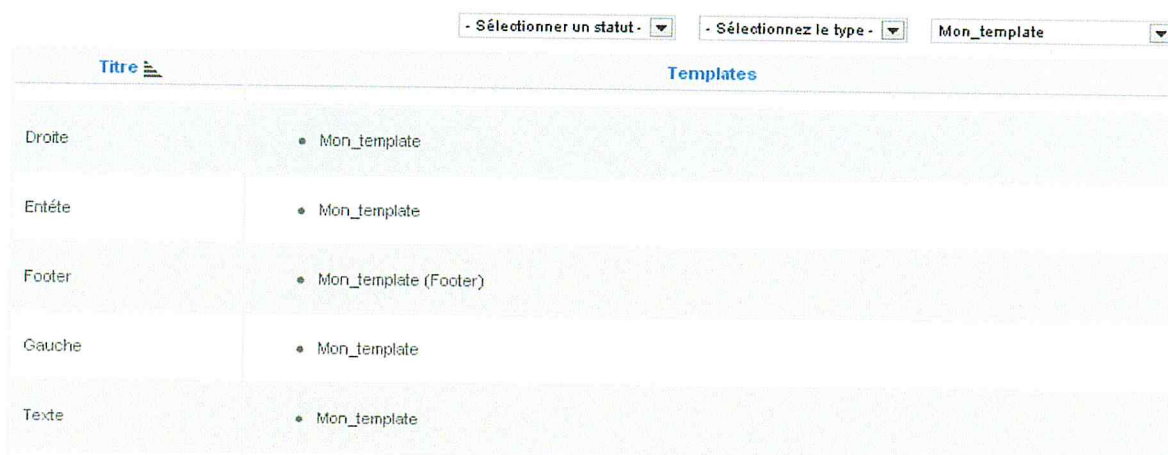


Figure 7 : le choix d'un position

Nous allons maintenant modifier le fichier index.php

5- Modification du fichier index.php [WEB_08]

```
<? php defined('_JEXEC') or die;?>
```

C'est une ligne de commande pour sécuriser le système. Il vérifie que les données proviennent de Joomla.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language; ?>"
```

```
lang="<?php echo $this->language; ?>" >
```

Ce sont 2 lignes standard. Elles ne sont pas propres à Joomla.

`<head>` : On ouvre la commande Head, l'entête du fichier html.

`<jdoc:include type="head" />` : On dit à Joomla d'ouvrir les balises

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/system/css/system.css" type="text/css" />
```

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/system/css/general.css" type="text/css" />
```

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo $this->template ?>/css/template.css" type="text/css" />
```

On dit à Joomla d'ouvrir les 3 fichiers templates. Les 2 premiers sont les fichiers standards de Joomla. Le 3e c'est notre fichier. Au lieu d'utiliser le nom du site et le nom du templates on utilise cette commande

`$this->baseurl` : utilise l'adresse du site

`$this->template` : utilise le nom du template.

`</head>` : On ferme la commande head

`<body>` : On ouvre body, la partie central de la page

On va utiliser des id pour identifier les différents blocs : `<div id= "nom du bloc">`. Vous pouvez changer les noms comme vous voulez. Il faut juste que le nom dans le fichier.php corresponde au nom qui se trouve dans template.css. J'essaie d'utiliser des noms significatifs, qui sont parlants.

`<div id="cadre_exterieur">` : on indique d'ouvrir le bloc cadre_exterieur. C'est le cadre qui permet de centrer toutes les colonnes et de ne pas coller les colonnes au bord de l'écran. Les informations seront rentrées dans le fichier template.css.

`<div id="logo"></div>` : on indique d'ouvrir le bloc logo et de le refermer juste après.

Joomla affichera le logo du site

```
<div id="cadre_gauche"><jdoc:include type="modules" name="gauche" style="xhtml" />
</div> :
```

On va détailler la ligne :

`<div id="cadre_gauche">` : on indique que l'on veut utiliser le style (couleur, fond, retrait...) `cadre_gauche` pour le bloc qui arrive.

`<jdoc:include type="modules" name="gauche" style="xhtml" />` : `jdoc:include` est la ligne de commande qui indique à Joomla d'afficher un module à cet endroit. La position de ce module aura le nom de gauche. On utilisera le style `xhtml`.

`</div>` : ferme le bloc

```
<div id="cadre_texte"><jdoc:include type="component" style="xhtml" /></div> :
```

Cette ligne indique que l'on va afficher le texte (component) dans le bloc que l'on appelle `cadre_texte`

```
<div id="cadre_droite"><jdoc:include type="modules" name="droite" style="xhtml" />
</div> :
```

C'est la même ligne que pour la colonne gauche, sauf que l'on met les paramètres pour la droite.

Il faut retenir ceci:

`<jdoc:include type="modules" name="nom de la position" style="xhtml" />` : indique le nom d'un module

`<jdoc:include type="component" style="xhtml" />` : Affiche le texte

Ce sont ces 2 commandes qu'il faut connaître. Ces 2 lignes sont spécifiques à Joomla. C'est cela qui permet d'afficher le texte et les modules

`<div class="clr"></div>` : c'est une commande que l'on utilise pour réinitialiser les positions

```
<div id="footer"><jdoc:include type="modules" name="pied_de_page" style="xhtml"
/></div> :
```

comme avant. On indique à Joomla d'afficher un module qui à la position `pied_de_page` en utilisant le style `footer`.

`</div>` : ferme le bloc `cadre_exterieur`

`</body>` : ferme `body`, le corps du fichier.

Nous pouvons enregistrer, le format `index.php` est fini.

6- Modification du fichier template.css [DOC_06]



Nous allons maintenant voir ce que cela donne sur notre site. Si on compresse le dossier de notre template maintenant et on installe le template (se qu'on va expliquer dans le partie installation) .Aller voir sur votre site et voici ce que cela donne:

Menu principal

- [Accueil](#)
- [Catégorie 1](#)
- [Catégorie 2](#)
- [Catégorie 3](#)

Accueil

[essai \(2\)](#)

- 
- 

Détails

Catégorie : [catégorie 1](#)
Publié le mercredi 25 avril 2012 10:58
Écrit par Super Utilisateur
Affichages : 0

Lorem ipsum dolor sit amet, consectetur adipiscing eros quis ultrices. In vitae urna et nunc pulvinar malesuada ante lectus in purus. Vivamus interdum porta. Ut rutrum mollis vulputate.

Cela ne correspond pas à ce que je veux. Tout est mis l'un en dessous de l'autre. C'est normal, car on n'a pas encore rentré les données de style. On va le faire maintenant. Nous allons dans le répertoire de notre template/ « mon_template, nous ouvrons le dossier : css, nous ouvrons le fichier template.css avec notre éditeur. C'est du code css. On va maintenant rentrer nos premières lignes :

```
/* General */
html {margin:0;padding:0}
body {background:#EBEBEB;font-family: Arial,sans-serif;font-size:12px;width:100%}
img {border:none}
.clr {clear:both}
```

Figure 8 : fichier présente le template.css

`/* General */` : c'est un commentaire. Dans la partie générale, je mets des informations générales.

Tous les commentaires sont à rentrés comme ceci : `/* commentaire */`

`html {margin:0;padding:0}` : on indique que les marges et les retraits sont à 0

`body {background:#EBEBEB;font-family: Arial,sans-serif;font-size:12px;width:100%}` :

Chapitre 3 : Eléments nécessaire à la génération d'un template

on va utiliser un fond gris (EBEBEB), la police arial ou sans empattement si la police arial n'est pas installée sur le poste, les caractères seront de la taille 12 pouces, on utilisera toute la largeur du navigateur

`img {border:none}` : on indique que l'on ne veut pas de cadre autour des images

`.clr {clear:both}` : commande pour réinitialiser les positions

Maintenant on va s'attaquer aux id :

```
/* Id */
```

```
#cadre_exterieur{background:#FFF;width:90%;min-width:800px;margin:40px auto;padding:10px;border:5px outset #808080}
```

```
#logo{width:100%;height:97px;margin:0 auto;background:url(..images/logo.png) center norepeat}
```

```
#cadre_gauche, #cadre_droite{width:20%;float:left}
```

```
#cadre_texte{width:60%;float:left}
```

```
#pied_de_page {margin:10px 0;text-align:center}
```

```
#cadre_exterieur{background:#FFF;width:90%;min-width:800px;margin:40px auto;padding:10px;border:5px outset #808080} :
```

on crée un cadre, dans lequel les autres blocs seront imbriqués. Il a un fond blanc (&FFF), il a une taille de 90% de la taille du navigateur, avec une dimension minimum de 800 pixels. En hauteur et bas de page, il a une marge de 40 pixels et sur les côtés, c'est calculer automatiquement. A l'intérieur du cadre, il y a un retrait de 10 pixels sur chaque côté. Le cadre est en relief.

```
#logo{width:100%;height:97px;margin:0 auto;background:url(..images/logo.png) center norepeat}
```

: C'est le logo. Mon image à une hauteur de 97 pixels. Il faudra indiquer la hauteur de votre logo

`#cadre_gauche, #cadre_droite{width:20%;float:left}` :C'est le style que l'on a indiqué pour le module gauche (dans le fichier index.php). On indique que le bloc aura une largeur de 20% de la taille de `cadre_exterieur`. On indique qu'il est positionné sur la gauche de la page. Si on n'indique pas cette position, les 3 blocs (gauche, droite et texte) seront affichés l'un en dessous de l'autre.

Comme le `cadre_gauche` et le `cadre_droit` ont les mêmes infos, on les rentre sur la même ligne, pour éviter de retaper les informations.

`#cadre_texte{width:60%;float:left}` : C'est le cadre du texte. Il occupera 60 % de la taille de

cadre_exterieur, logique vu que les 2 autres colonnes font $20 + 20 = 40\%$ de la taille.

`#pied_de_page {margin:10px 0;text-align:center}` : on met une marge de 10 pixels au dessus et en dessous du bloc et on centre le texte.

Maintenant que les fichiers `templateDetails.xml`, `index.php` et `template.css` sont créés, on va créer notre fichier compressé.

Compresser le dossier

Maintenant, nous revenons à la racine de notre dossier. Compressant le dossier `mon_template`.

7- Installer notre template [DOC_06]

Pour installer le template, il faut aller dans la partie administration du site :

Extensions → Gestion des extensions

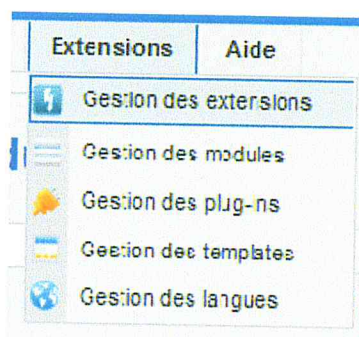


Figure 9 : Gestion des extensions dans joomla

Il faut aller chercher le fichier `mon_template.zip`

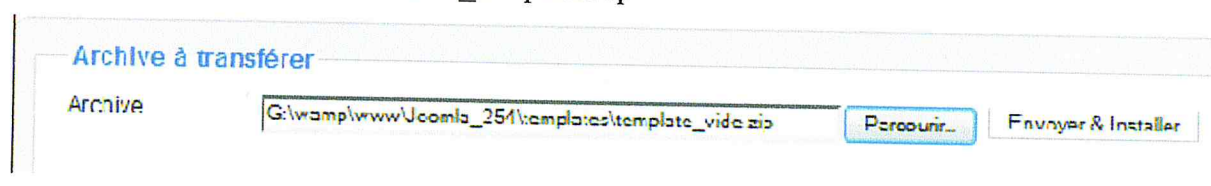


Figure 10 : Transfère d'un archive

Chapitre 3 : Eléments nécessaire à la génération d'un template

Cliqué sur Envoyer & Installer.

Le template est installé. Pour le vérifier, cliquer sur : Extensions -> Gestion des templates

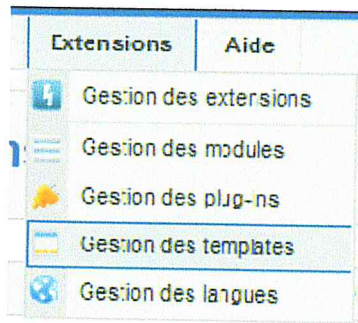


Figure 11 : Gestion des template dans joomla

Le template mon_template doit être affiché. Cliquer sur l'étoile blanche qui se trouve sur la ligne du template

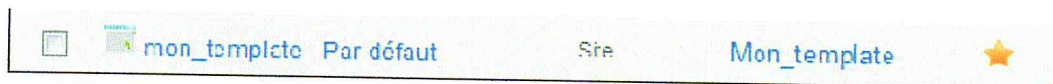


Figure 12 : Présentable d'une ligne pour montemplate

L'étoile devient jaune. Le template est maintenant le template par défaut.

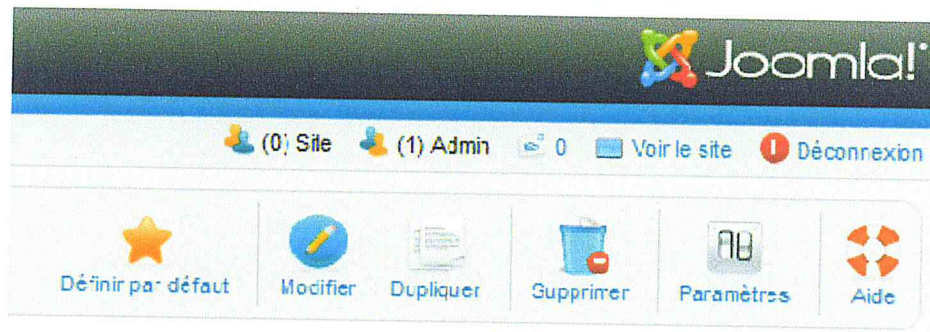


Figure 13 : Une template par défaut

Si on clique sur l'icône voir le site (en haut à droite de l'écran) nous allons faire apparaître le site.



Figure 14 : Présentable d'un site de demonstration

Mes modules ne sont pas affichés. Normal, nous n'avons pas indiqué les positions dans Joomla!. Il faut aller dans le module administration de Joomla! → gestion des modules :

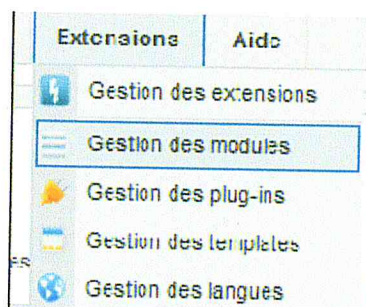
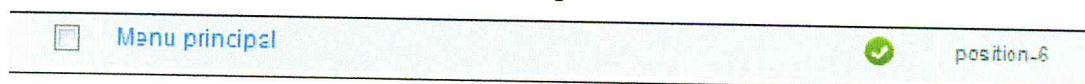


Figure 15 : Gestion des modules pour Joomla!

Pour notre menu, on voit qu'il est affiché à une position x



Cliquer sur Menu principal :

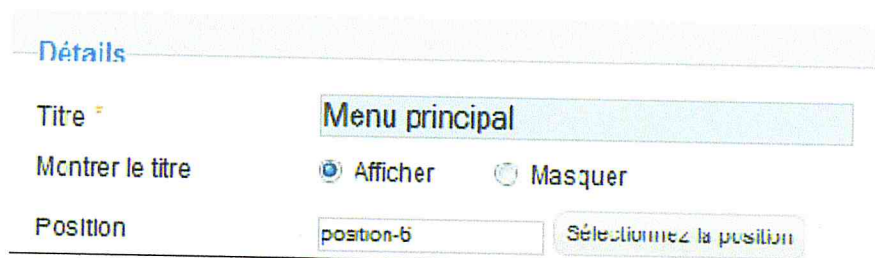


Figure 16 : Présentable de Menu principal on Joomla!

Cliquer sur Sélectionner la position :

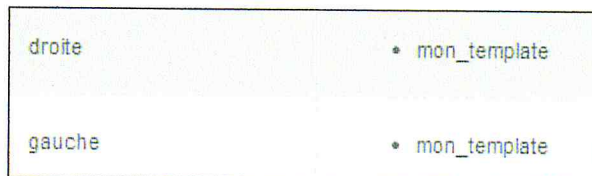


Figure 17 : Sélectionnée la position de template

Il nous affiche nos 2 positions que nous avons rentrées dans le fichier templateDetails.xml.

Cliquer sur gauche

Faites la même chose pour le module connexion, mais choisissez droite.

Après, allez de nouveau consulter votre site, il devrait ressembler à cela :



C'est ce que l'on avait prévu. Si vous supprimer la class clr, voici ce que l'on obtient :



Figure 18: Présentation de votre site

Deuxième Partie :

Conception et Réalisation

Chapitre 1 : Conception

Introduction :

Dans ce chapitre nous avons analysés les besoins de notre système c'est-à-dire que faire notre future système, et nous avons donnés une vue générale sur les différents fonctions de notre application ainsi que les cas d'utilisation de chacun d'entre eux et nous allons par la suite détaillés chaque cas d'utilisation.

Pour construire un système informatique le passage par l'étape de modélisation est obligatoire. La modélisation orienté objet consiste à modéliser informatiquement un ensemble d'éléments qui appartiennent au monde réel en un ensemble d'entités informatique (objet).

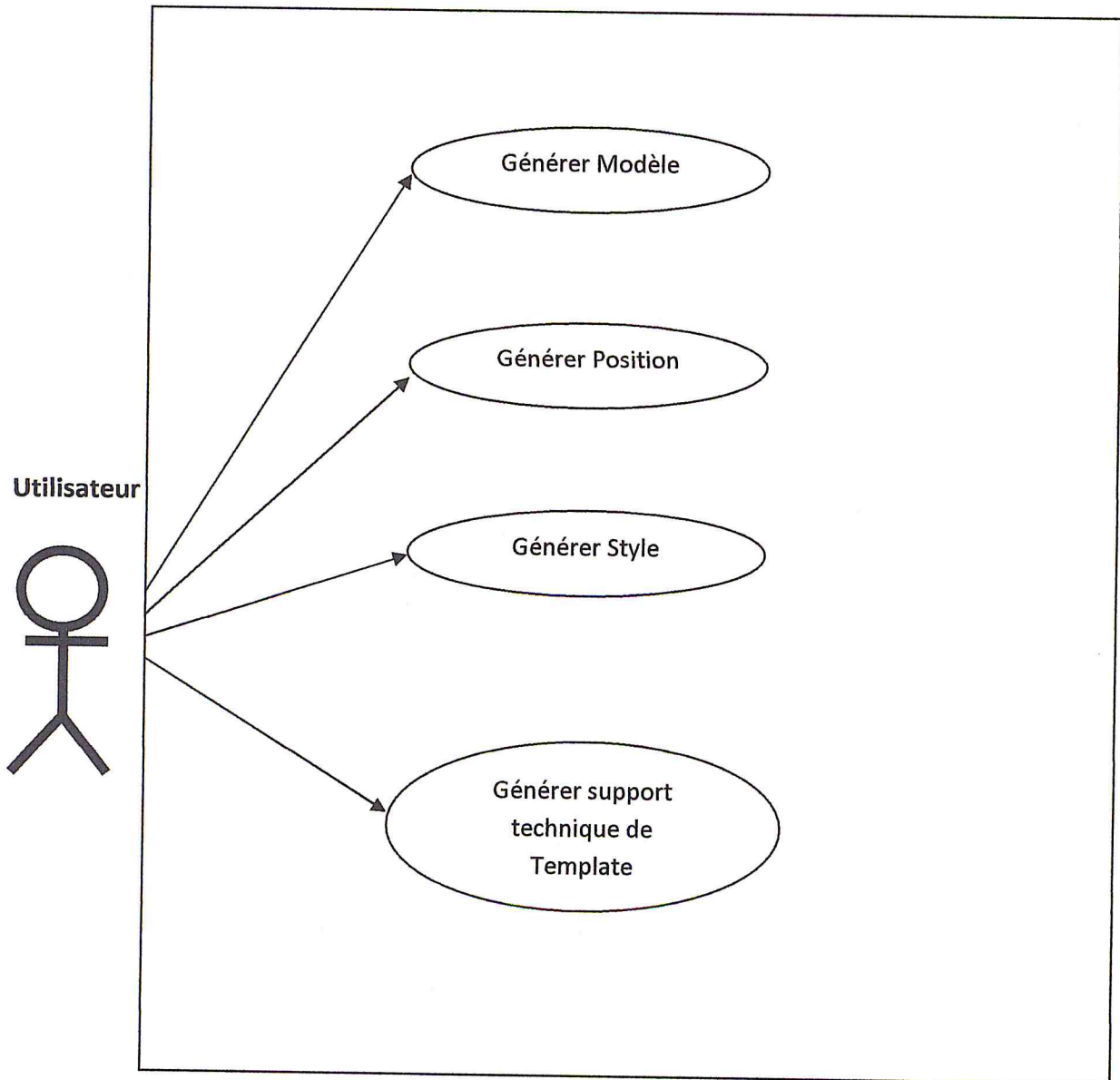
La modélisation consiste tout d'abord à écrire un problème (les besoins à propos d'un système informatique à construire), puis à décrire la solution de ce problème (la conception du système à construire).

Pour modéliser notre système nous utiliserons le langage UML.

1- Analyse des besoins

1.1- Diagrammes de cas d'utilisation :

Un diagramme de cas d'utilisation permet de représenter les relations entre les acteurs et les fonctionnalités du système, ainsi c'est une vue externe de la façon d'utiliser un système, que ce soit application, un sous-système, une fonction, un composant.



Un diagramme de cas d'utilisation : création du Template avec l'application

1.2- Explication :

Nous montrons dans ce diagramme que l'utilisateur crée son template à travers différents étapes ordonnées :

En commençant par le choix de modèle de template réaliser souvent à lequel est classé son modèle site, puis si les modèles de template sélectionné ça marche pas avec le besoin d'utilisateur, il choisit d'autres modèles du site qui présente la base pour générer son futur template.

Le system permet à l'utilisateur de personnaliser sa vue externe par la mise en forme de différent position qui existe dans le modèle choisi, puis il génère le style de différent position réalisé souvent.

A la fin, l'utilisateur génère le support technique de template (les différents fichiers qui constituent le template).

2- Description des cas d'utilisation :

Après la présentation des besoins identifiés de manière formelle avec les diagrammes des cas d'utilisation présentés dans la partie précédente, Dans cette partie nous allons présenter les diagrammes d'UML que nous avons utilisés lors de la conception de notre système. Toutefois, nous donnerons ici les différents scénarios associés à chaque cas d'utilisation, en premier temps de manière générale, du point de vue de l'acteur à travers les diagrammes de séquences, qui permettent de montrer la chronologie des interactions et les informations échangées, ensuite nous décrirons ces scénarios, en identifiant les objets du système ainsi que leurs collaborations pour réaliser les cas d'utilisation à travers le diagramme de collaboration qui permet de mettre les interactions entre les différents objets du système étudié, ainsi que les messages qu'ils échangent entre eux.

2.1- Générer modèle de site

Diagramme de séquence

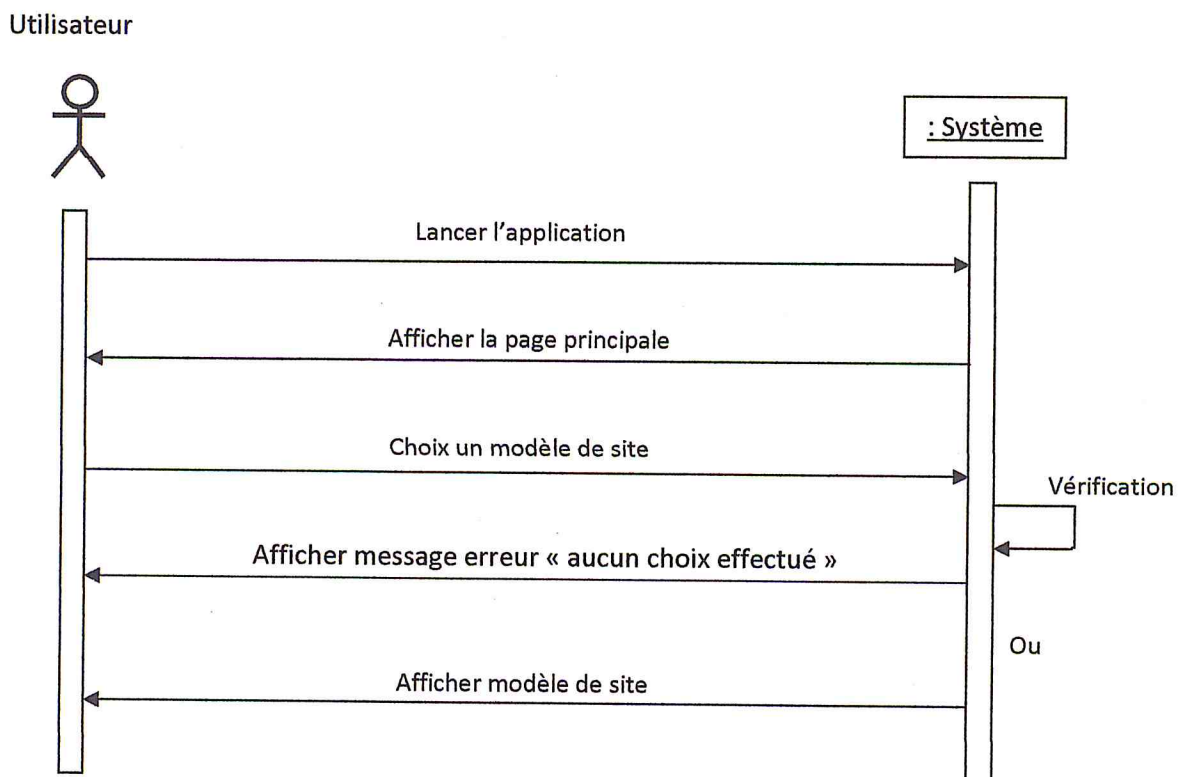


Diagramme de collaboration

Utilisateur

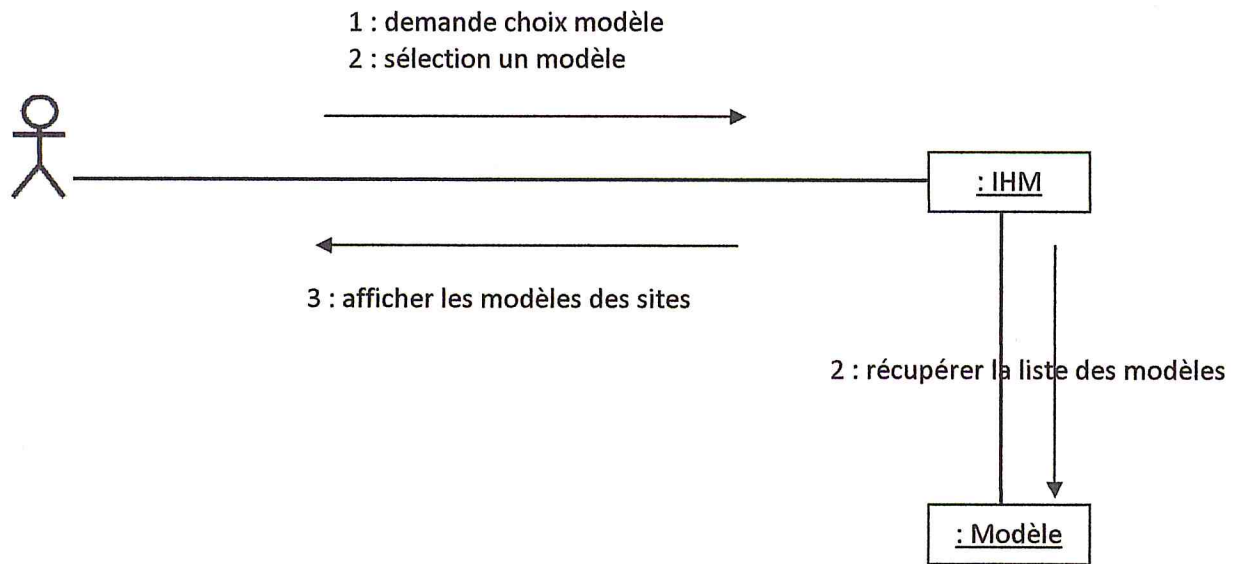
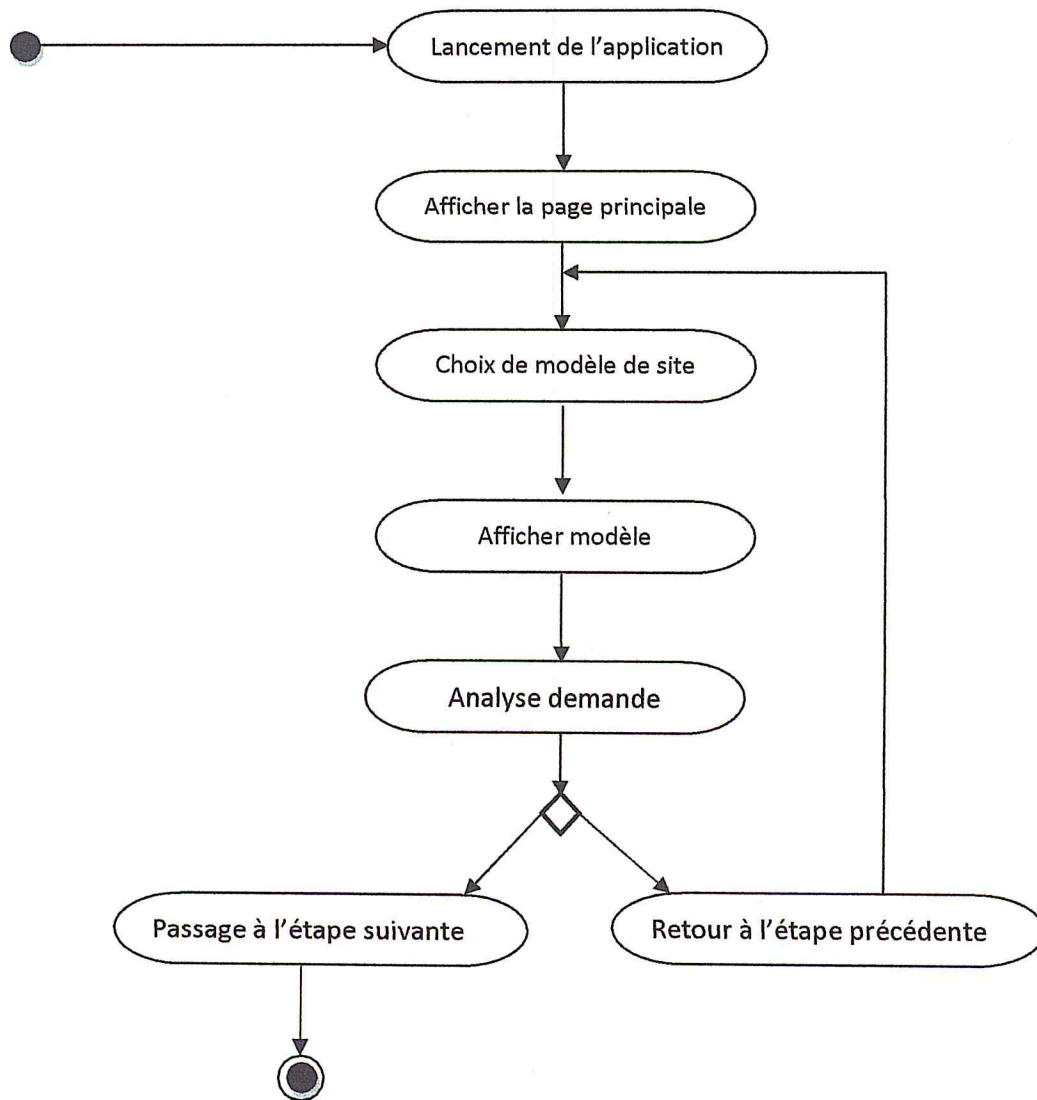


Diagramme d'activité



2.2- Générer position

Diagramme de séquence

Utilisateur

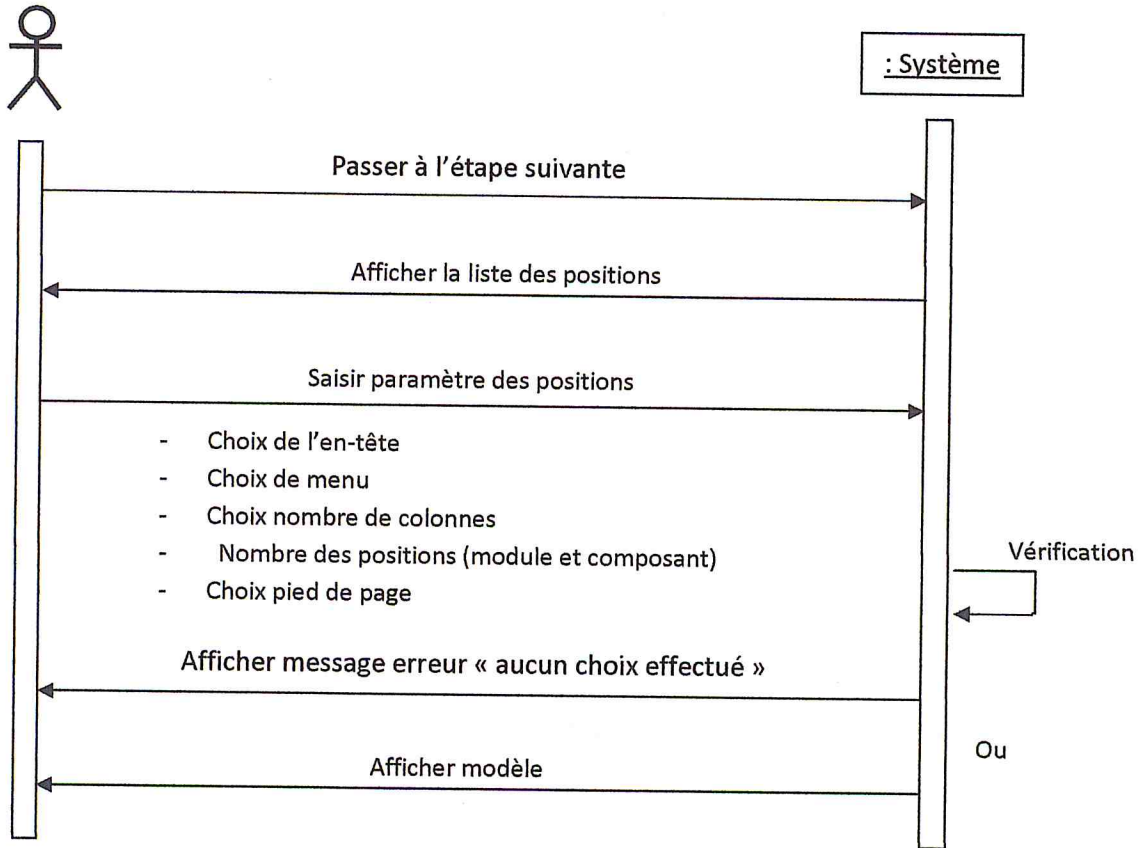


Diagramme de collaboration

Utilisateur

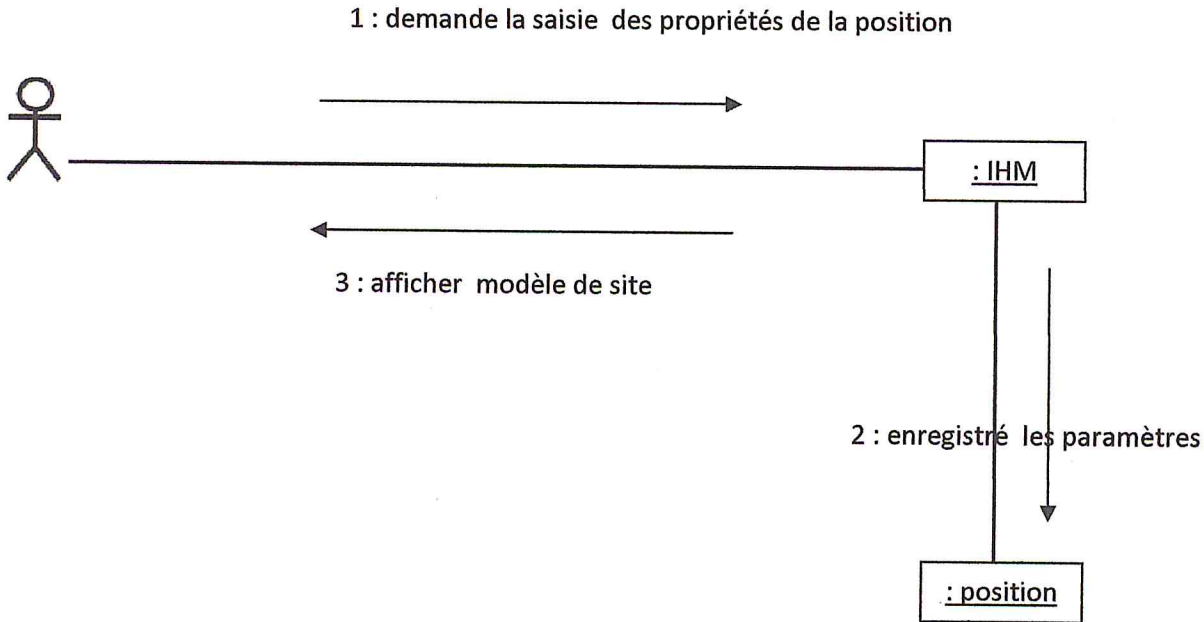
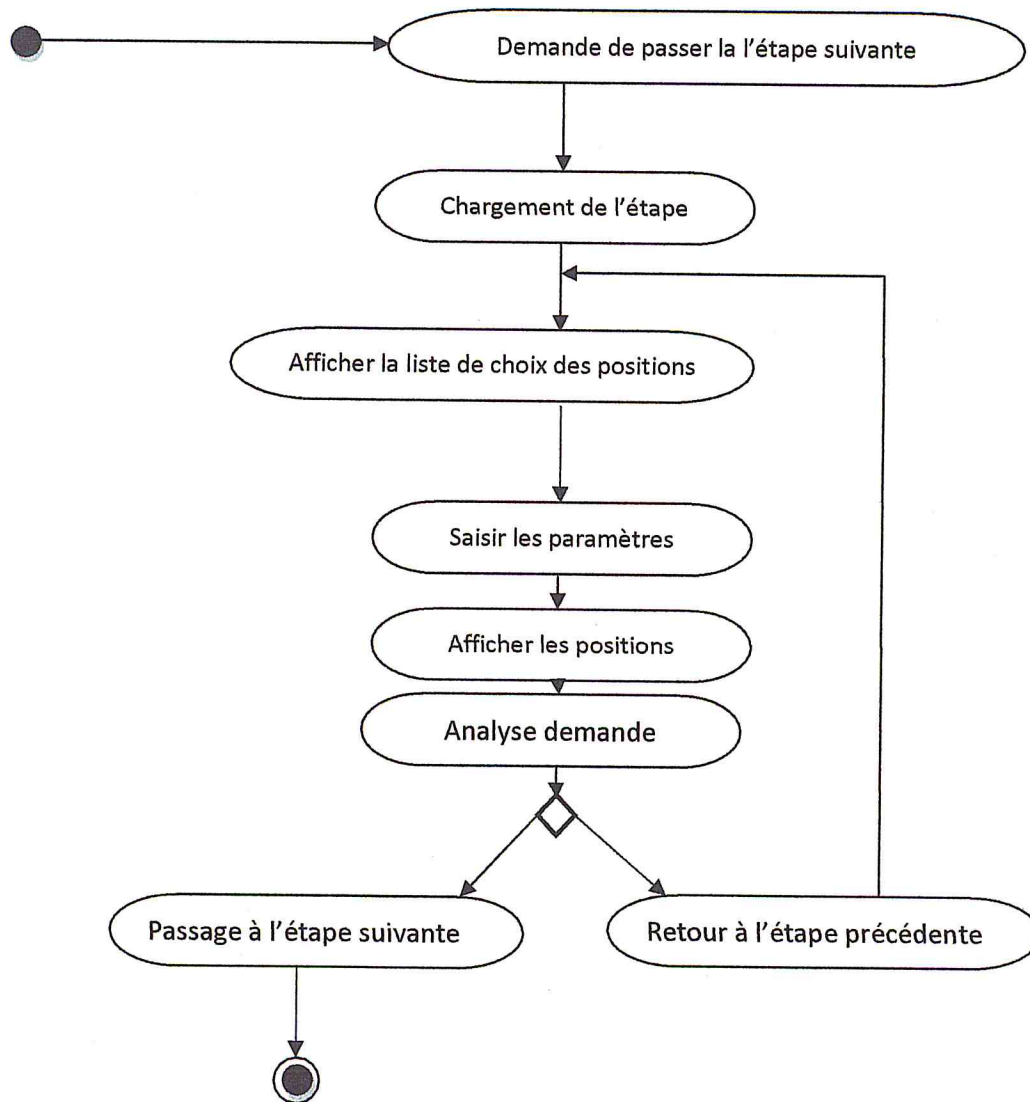


Diagramme d'activité



2.3- Générer style

Diagramme de séquence

Utilisateur

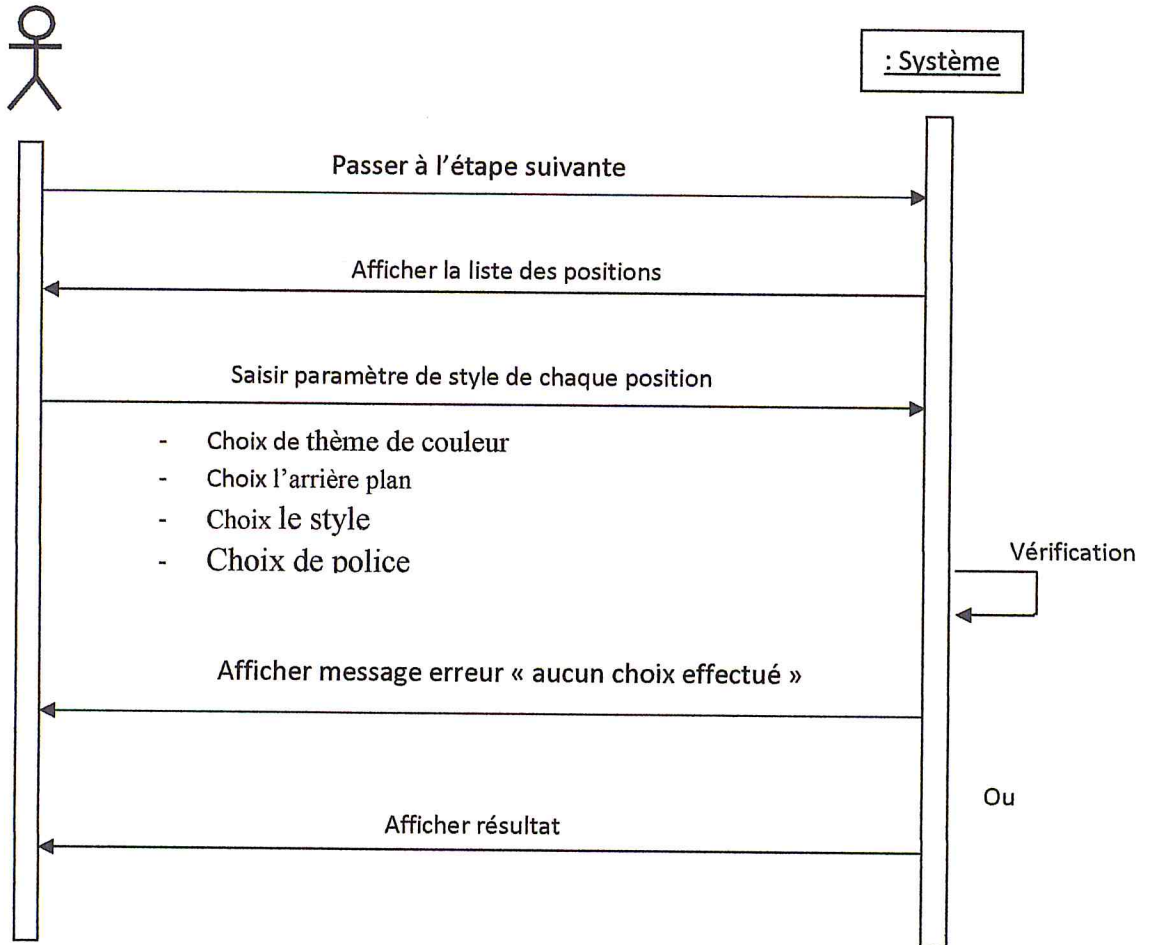
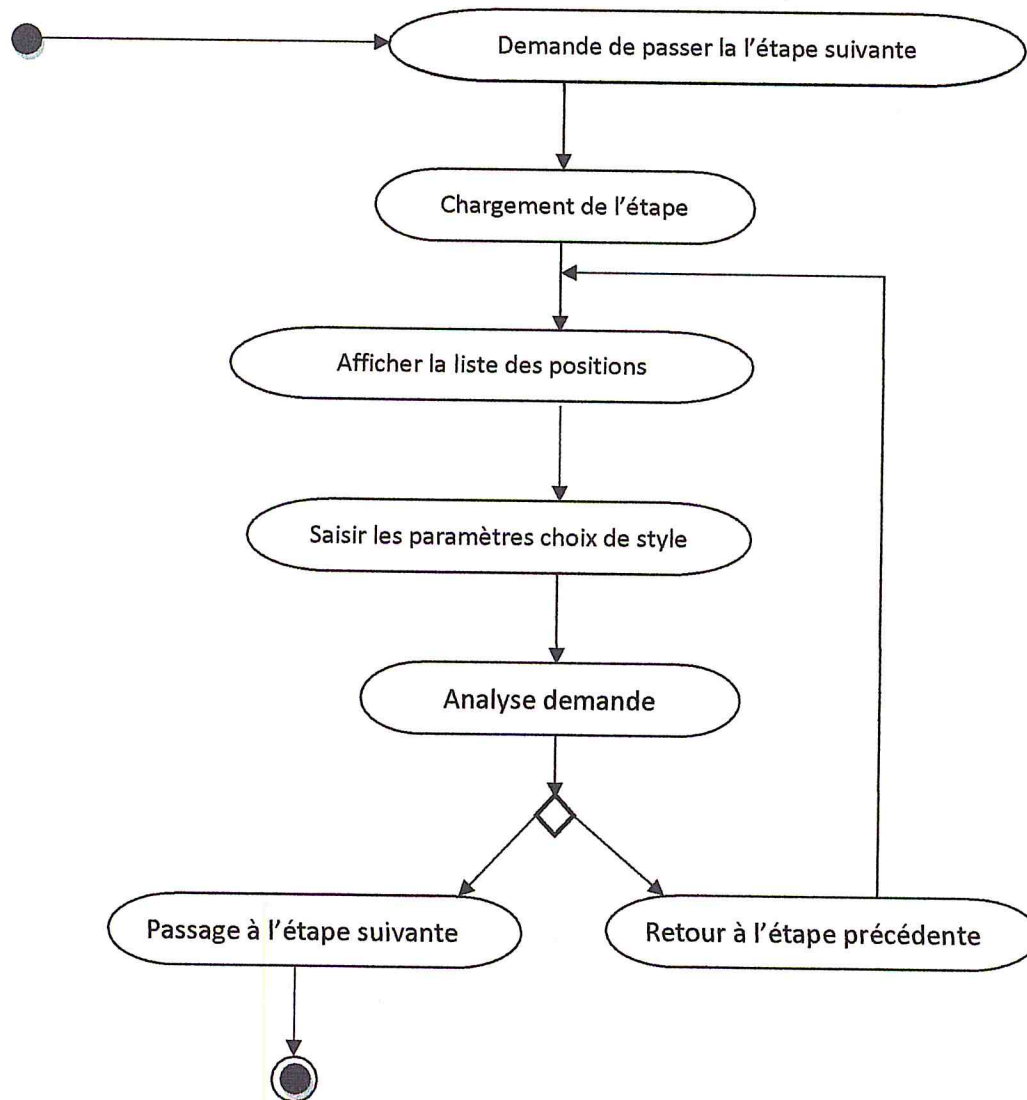


Diagramme d'activité



2.4- Générer support technique de Template

Diagramme de séquence

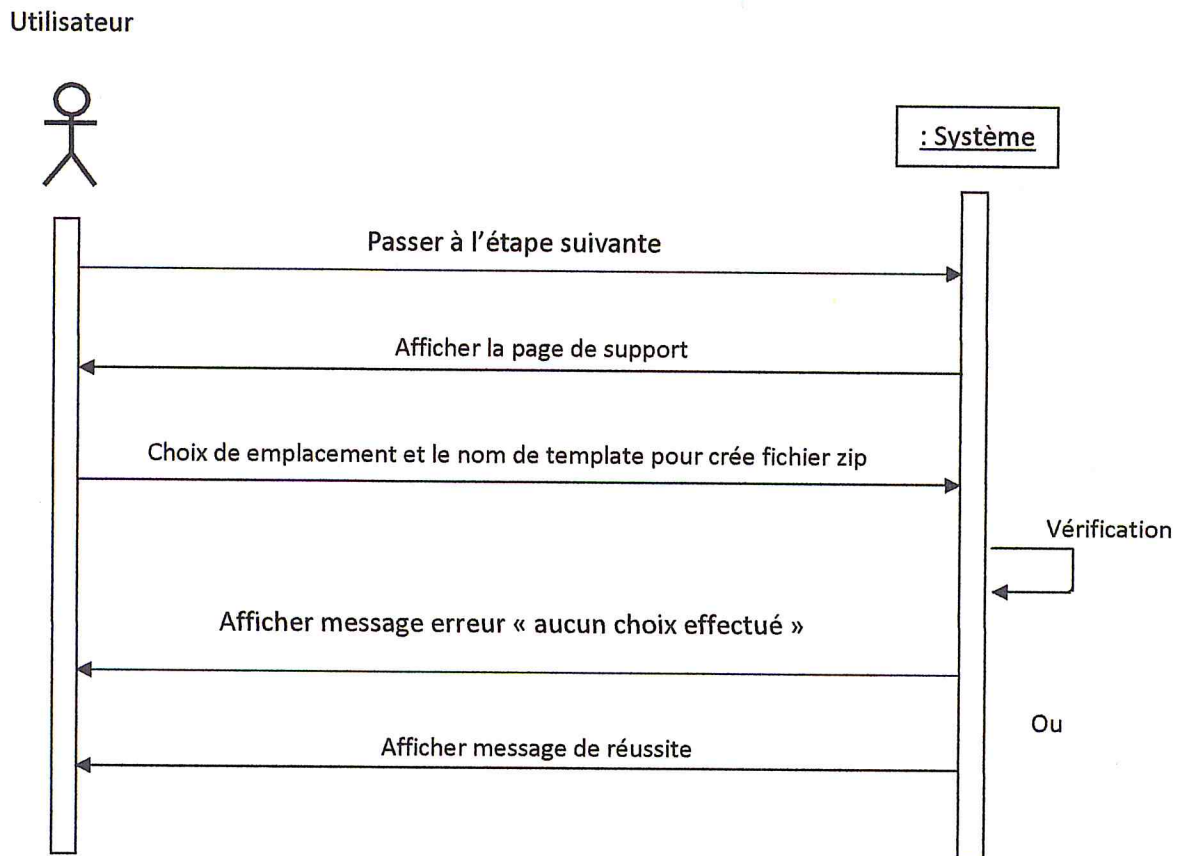
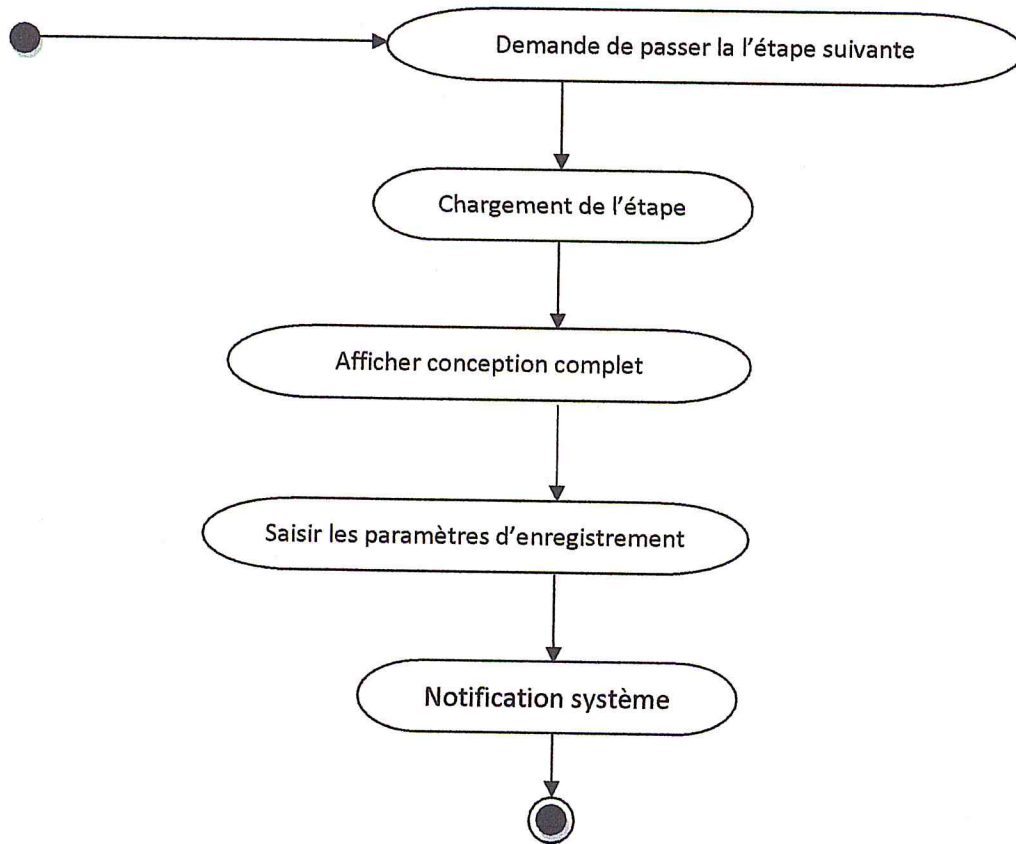
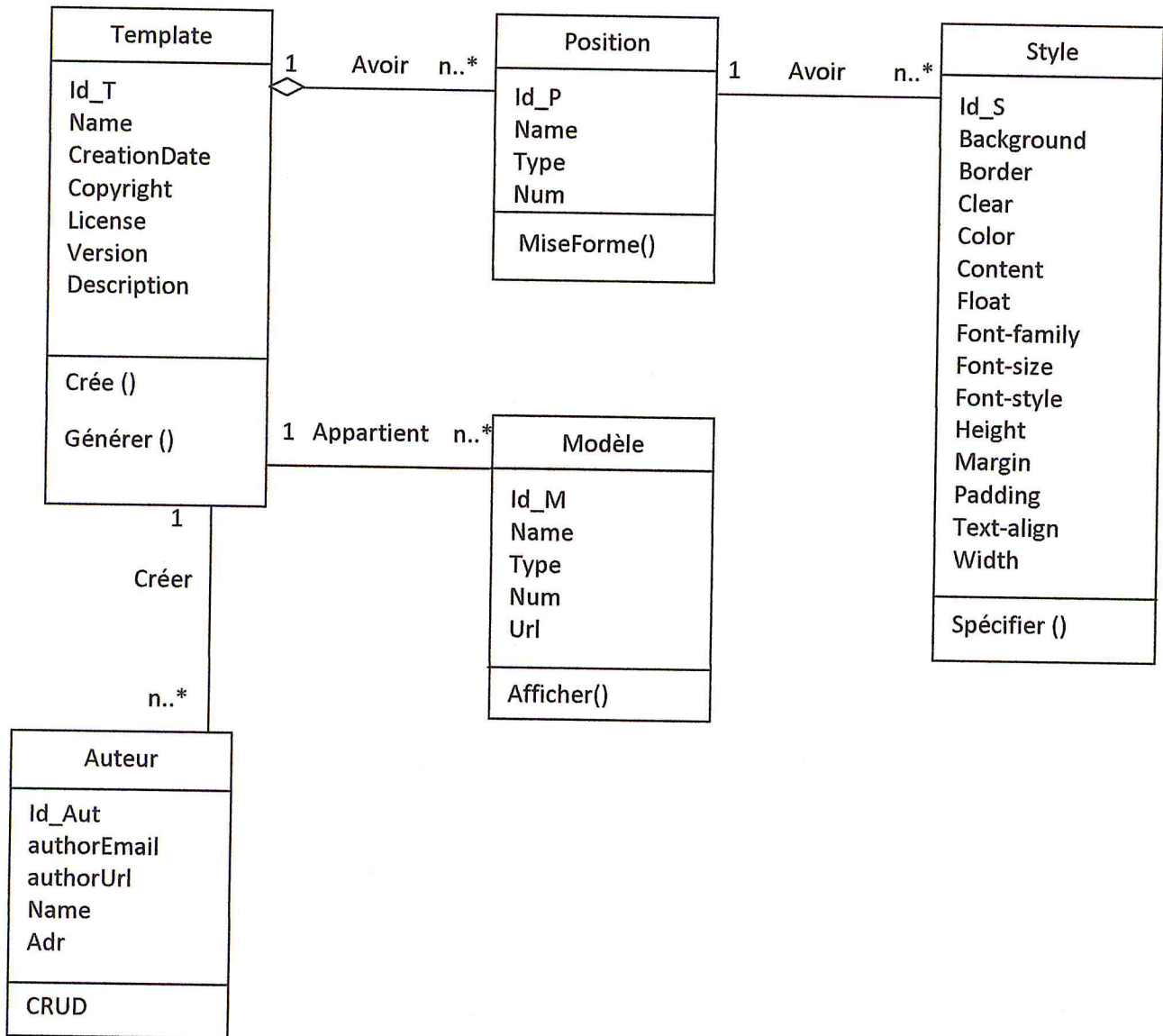


Diagramme d'activité



4- Diagramme de classe :

Le diagramme de classe est un modèle permettant de décrire d'une manière abstraite et générale les liens entre objets, comme il permet d'exprimer d'une manière générale la structure statique d'un système, en terme de classes et de relations entre ces classes. Le diagramme de classe ci-dessous correspond à notre application.



4.1- Dictionnaire de données

Nom classe	Rôle	Termes	Signification
Template	Enregistrer dans la base de données JOOMLA associée au site les différents templates créés par l'utilisateur.	Id_T	Identification d'un template.
		Name	Indique le nom du template.
		CreationDate	Indique la date de création.
		Copyright	Indique Copyright
		License	Indique License.
		Version	Indique la version
		Description	Description
Position	Enregistrer les informations des positions existantes dans le template.	Id_P	Identification d'une position.
		Name	Indique le nom de la position.
		Type	Indique le type de position. « composant » ou bien « module »
		Num	Indique le numéro de position dans le template.
Modèle	Enregistrer les informations des Modèles existants dans l'application.	Id_M	Identification d'un Modèle.
		Name	Indique le nom d'un Modèle.
		Type	Indique le type de Modèle.
			Indique le numéro de

		Url	Modèle dans le répertoire.
Style	Enregistrer les informations de configurations de style acquis par l'utilisateur durant la création.	Id_S	Identification d'un style.
		Background	Indique la couleur de fond
		Border	Indique la bordure
		Clear	Indique l'alignement
		Color	Indique la couleur
		Content	Indique contenant
		Float	Indique l'alignement
		Font-family	Indique police de thème
		Font-size	Indique la taille de police
		Font-style	Indique le style de font
		Height	Indique l'auteur
		Margin	Indique la marge
		Padding	Indique le retrait
		Text-align	Indique l'alignement
Width	Indique la largeur		
Auteur	Enregistrer les informations correspondent au l'auteur.	Id_Aut	Identification d'auteur
		authorEmail	E-mail de l'auteur
		authorUrl	URL de l'auteur
		Name	Indique le nom d'auteur.
		Adr	Indique l'adresse de l'auteur.

5- Le schéma relationnel Base de données de l'application

Template (Id_T, Name, creationDate, Copyright, License, Version, Description, Id_M*, Id_Aut*).

Position (Id_P, Name, Type, Num, Id_T*, Id_S*).

Style (Id_S, Background, Border Clear, Color, Content, Float, Font-family, Font-size, Font-style, Height, Margin, Padding, Text-align, Width).

Modèle (Id_M, Name, Type, Num, Url).

Auteur (Id_Aut, authorEmail, authorUrl, Name, Adr).

6- Conclusion :

Dans ce chapitre nous avons détaillé les cas d'utilisation de notre application d'où nous avons expliqué chaque cas avec ces différents diagrammes selon le langage UML pour faire sortir le diagramme de classe générale qui englobe les classes de notre application puis nous avons fait le passage vers le modèle relationnel.

Chapitre 2 : Réalisation

Introduction :

Dans ce chapitre nous présenterons les principales parties d'une application basée sur le langage PHP. Nous commencerons par les outils d'implémentations et on dernier nous exposerons quelque pages de notre application.

1- Les outils d'implémentation :

1.1- HTML : [WEB-02]

Pour faire court, HTML a été inventé en 1990 par un scientifique nommé Tim Berners-Lee. L'objectif était de faciliter l'accès par des scientifiques d'universités différentes aux documents de recherche de chacun. Le projet eut un succès inespéré, plus grand que tout ce que Tim Berners-Lee avait pu imaginer. En inventant HTML, il posa les fondations du Web tel que nous le connaissons aujourd'hui.

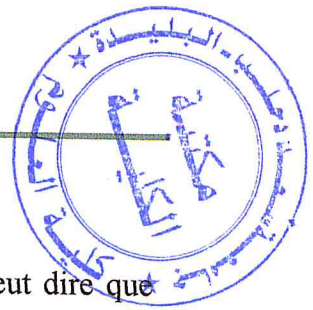
HTML est un langage qui permet de présenter des informations (par exemple, des recherches scientifiques) sur Internet. Ce que vous voyez quand vous regardez une page sur Internet est l'interprétation par votre navigateur du code HTML. Pour voir le code HTML d'une page sur Internet, cliquez simplement sur la rubrique « Vue » dans le menu supérieur de votre navigateur et sélectionnez « Source ».

1.2- CSS : [WEB-03]

CSS signifie en anglais « Cascading Style Sheets » et se traduit « feuilles de style en cascade » en français.

On utilise le langage CSS pour décrire la présentation d'un document écrit en htm ou xml. La CSS décrit les couleurs, la police...

La CSS permet de rendre un document plus agréable à lire, mieux présenté... La CSS a donc pour rôle d'améliorer la mise en page, de séparer la structure de sa présentation et d'améliorer l'accessibilité des pages web. Le langage CSS n'est autre que des feuilles de style en cascade.



1.3- PHP : [WEB-04]

Le langage PHP est un langage de programmation web côté serveur, ce qui veut dire que c'est le serveur qui va interpréter le code PHP (langage de scripts) et générer du code HTML qui pourra être interprété par votre navigateur.

Le PHP permet d'ajouter des fonctionnalités de plus en plus complexe, d'avoir des sites dynamiques, de pouvoir gérer une administration de boutique en ligne, de modifier un blog, de créer des réseaux sociaux.

Le PHP fut créée en 1994 par Rasmus Lerdorf, c'est un langage libre et gratuit, avec une grande communauté mondiale.

1.4- Javascript : [WEB-05]

JavaScript est un langage de programmation de scripts principalement utilisé pour les pages web interactives.

C'est une extension du langage HTML qui est incluse dans le code. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes.

Ce code est directement écrit dans la page HTML, c'est un langage peu évolué qui ne permet aucune confidentialité au niveau des codes (ceux-ci sont effectivement visibles).

1.5- Wamp : [WEB-06]

Wampserver est ce qu'on pourrait appeler un serveur web local pour windows. Par abus de langage, Wampserver (auparavant nommé WAMP5) est souvent désigné par WAMP.

Acronyme signifiant « Windows Apache MySQL PHP (dans la majorité des cas mais aussi parfois, « Perl » ou « Python ») », il comprend un programme destiné à se comporter comme un serveur web sur votre ordinateur.

Lorsque vous demandez à votre navigateur (comme Firefox, IE8, Chrome ou Opera par exemple) d'afficher une page web, celui-ci envoie une requête au serveur possédant cette page qui le lui envoie. Wampserver se comporte exactement de la même manière sauf qu'il se trouve directement sur votre machine. Il y a donc aucune information transmise sur l'extérieur

et vous pouvez donc tester votre site sans même avoir un hébergement ni même accès à internet ! C'est comme s'il se trouvait en ligne sur le web.

C'est ça la force de Wampserver !

Il comprend la suite Apache, MySQL et PHP. Voilà ce que nous dit Wikipedia sur son fonctionnement :

Les rôles de ces quatre composants sont les suivants :

- Apache est le serveur web « frontal » : il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur)
- Le langage de script PHP sert la logique et permet des traitements (calculs, vérification, test, etc.).
- MySQL stocke toutes les données de l'application (c'est une base de données).
- Windows assure l'attribution des ressources à ces trois composants.

Il existe de la même manière son homologue sous linux désigné par LAMP ou XAMPP (le X correspondant à Unix) et sous Mac Os avec MAMP.

Ce logiciel est donc parfait pour tous les développeurs de sites web qui aimeraient tester leurs créations avant leur mise en ligne, tester des scripts mais aussi définir un site disponible sur le réseau local comme on en trouve dans beaucoup de grandes entreprises pour tenir au courant le personnel.

1.6- SQL : [WEB-07]

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel qu'Access ou par les produits plus professionnels tels qu'Oracle ou Sybase. Il a fait l'objet de plusieurs normes ANSI/ISO dont la plus répandue aujourd'hui est la norme SQL2 qui a été définie en 1992. Nous décrivons ici les principaux aspects de cette norme.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution. Le langage SQL propose un langage de requêtes ensembliste et assertionnel. Néanmoins, le langage SQL ne possède pas la puissance d'un langage de programmation : entrées/sorties, instructions conditionnelles, boucles et affectations. Pour certains traitements il est donc nécessaire de coupler le langage SQL avec un langage de programmation complet au sens Turing du terme.

Le langage SQL comporte :

- Une partie sur la définition des données :

Le langage de définition des données (LDD) qui permet de définir des relations, des vues externes et des contraintes d'intégrité;

- Une partie sur les requêtes :

Le langage de manipulation des données (LMD) qui permet d'interroger une base de données sous forme déclarative sans se préoccuper de l'organisation physique des données;

- Une partie sur le contrôle des données :

Le langage de contrôle des données (LCD) qui permet de contrôler la sécurité et les accès aux données.

2- Présentation :

2.1- Présentation de l'application :

Présentation de l'application :



Figure 32 : Page d'accueil de l'application. (Bouton pour lancer l'application)



Figure 33 : Etape (Choix entre Template pris ou création d'une nouvelle).

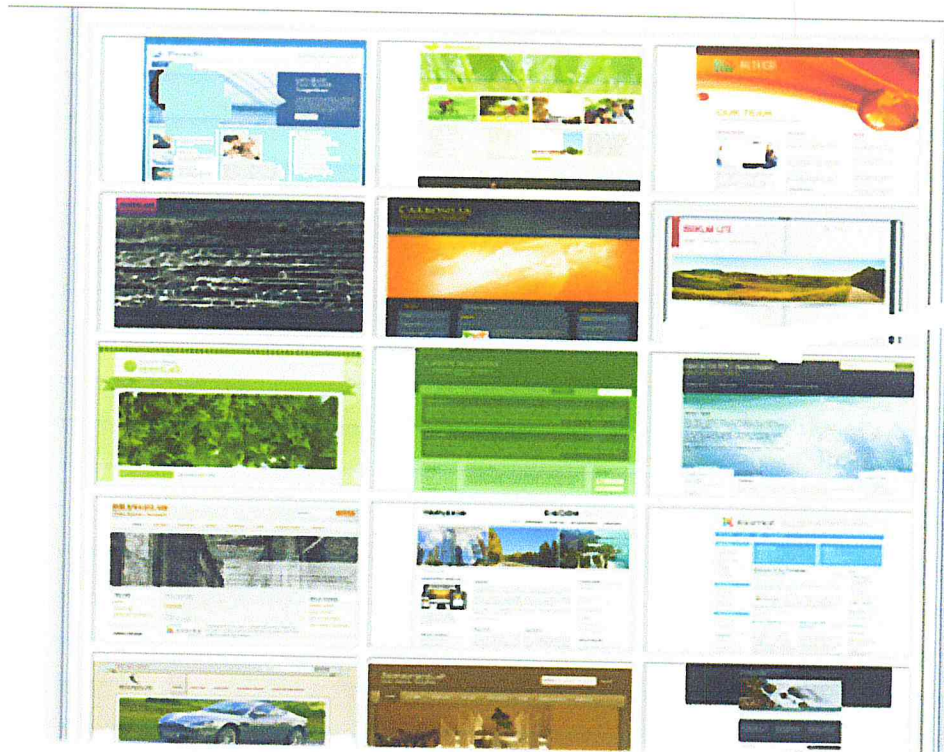
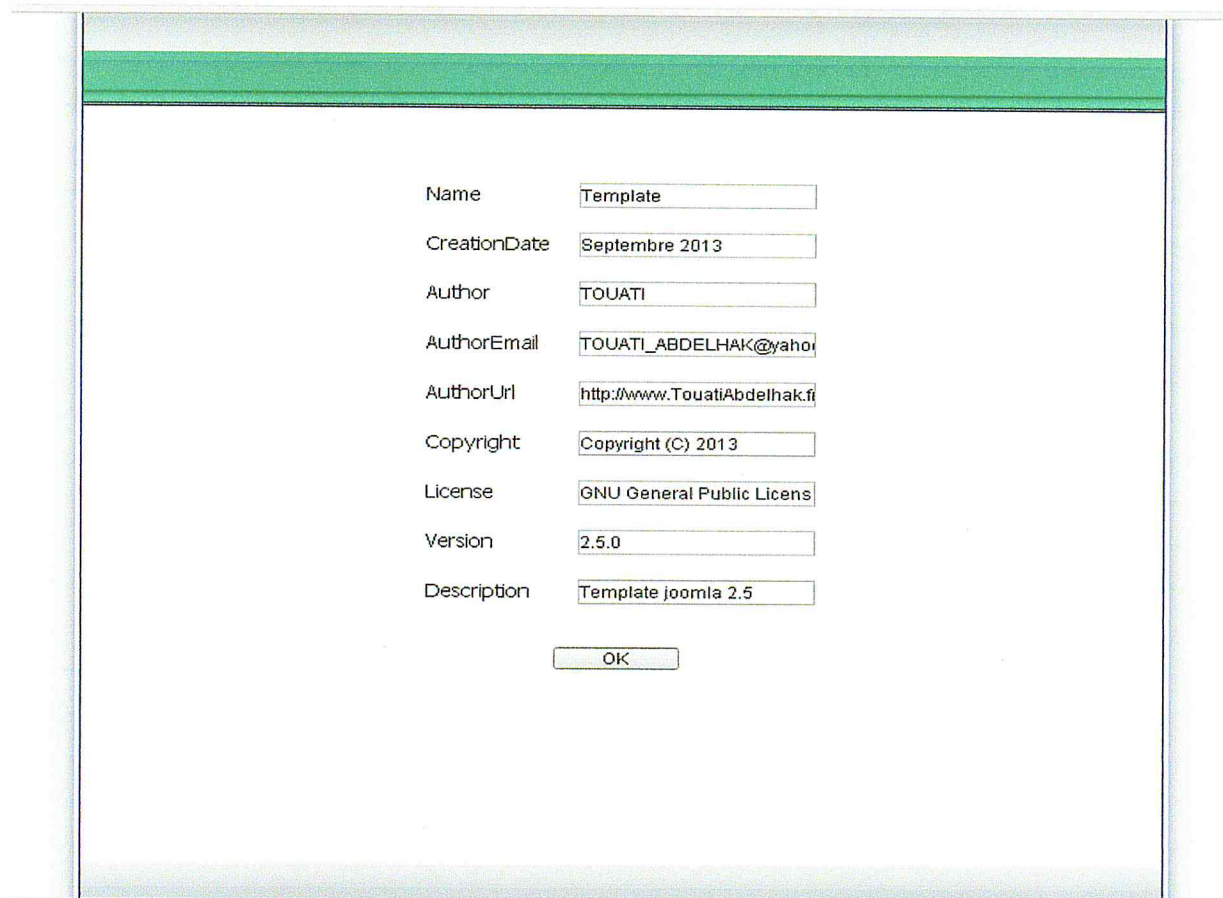


Figure 34 : Répertoire des Templates Complètes pour installer directement.



Figure 35 : Exemple de Template complet



The image shows a screenshot of a Joomla! administration interface for creating a new template. The form is contained within a window with a green header bar. It features several input fields for metadata, each with a label on the left and a text box on the right. The fields are: Name (Template), CreationDate (Septembre 2013), Author (TOUATI), AuthorEmail (TOUATI_ABDELHAK@yahoo), AuthorUrl (http://www.TouatiAbdelhak.fr), Copyright (Copyright (C) 2013), License (GNU General Public Licens), Version (2.5.0), and Description (Template joomla 2.5). An OK button is located at the bottom center of the form.

Name	Template
CreationDate	Septembre 2013
Author	TOUATI
AuthorEmail	TOUATI_ABDELHAK@yahoo
AuthorUrl	http://www.TouatiAbdelhak.fr
Copyright	Copyright (C) 2013
License	GNU General Public Licens
Version	2.5.0
Description	Template joomla 2.5

OK

Figure 36 : Page de saisir les informations de nouvelle Template

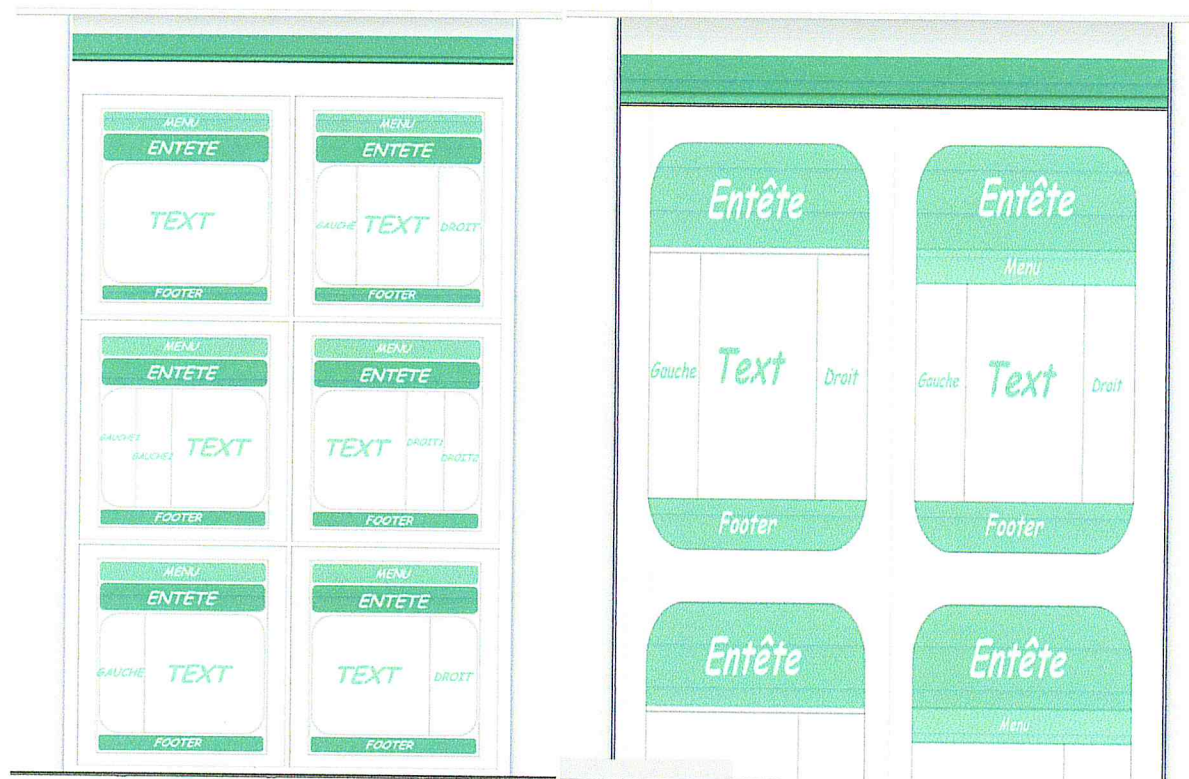


Figure 37 : Les pages de choix des modèles des Templates.

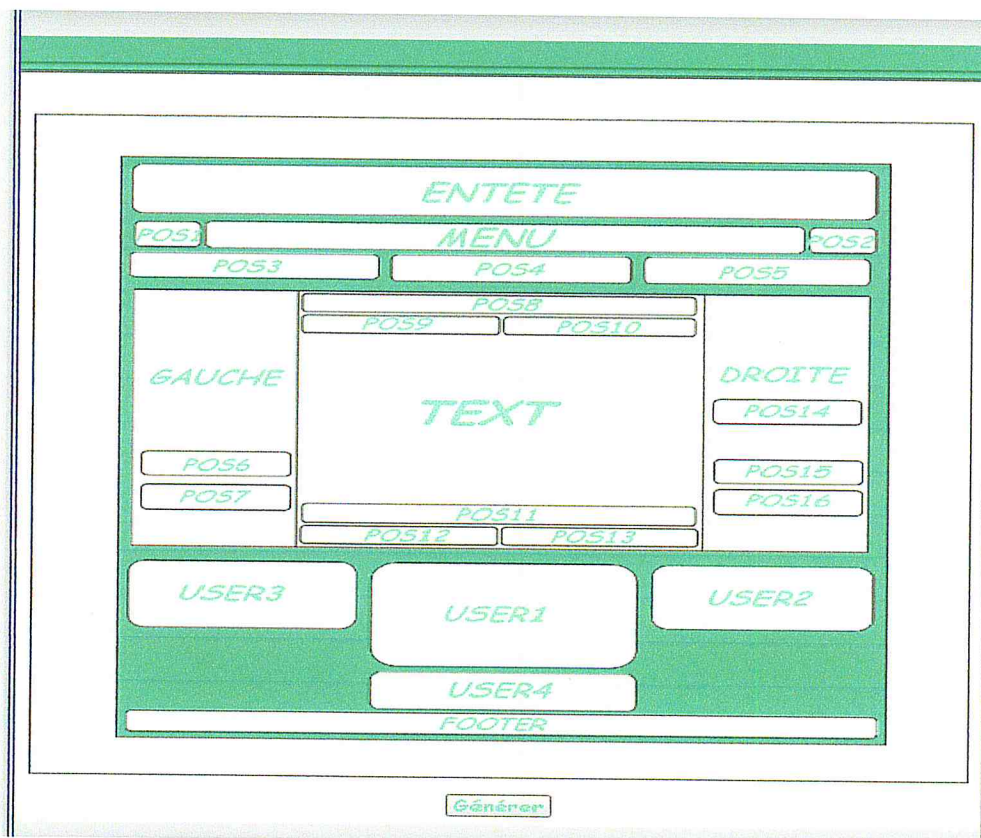


Figure 38 : Exemple de Modèle choisie

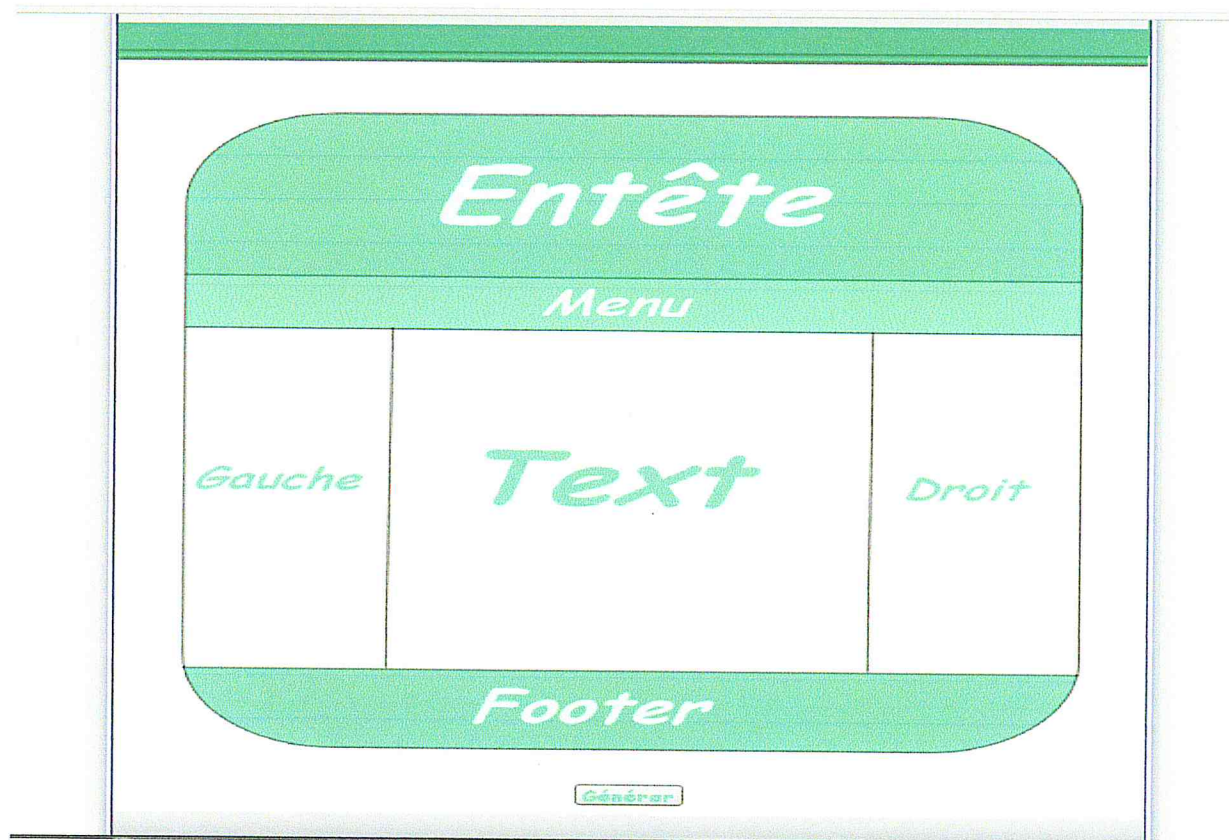


Figure 39 : Exemple de Modèle choisie

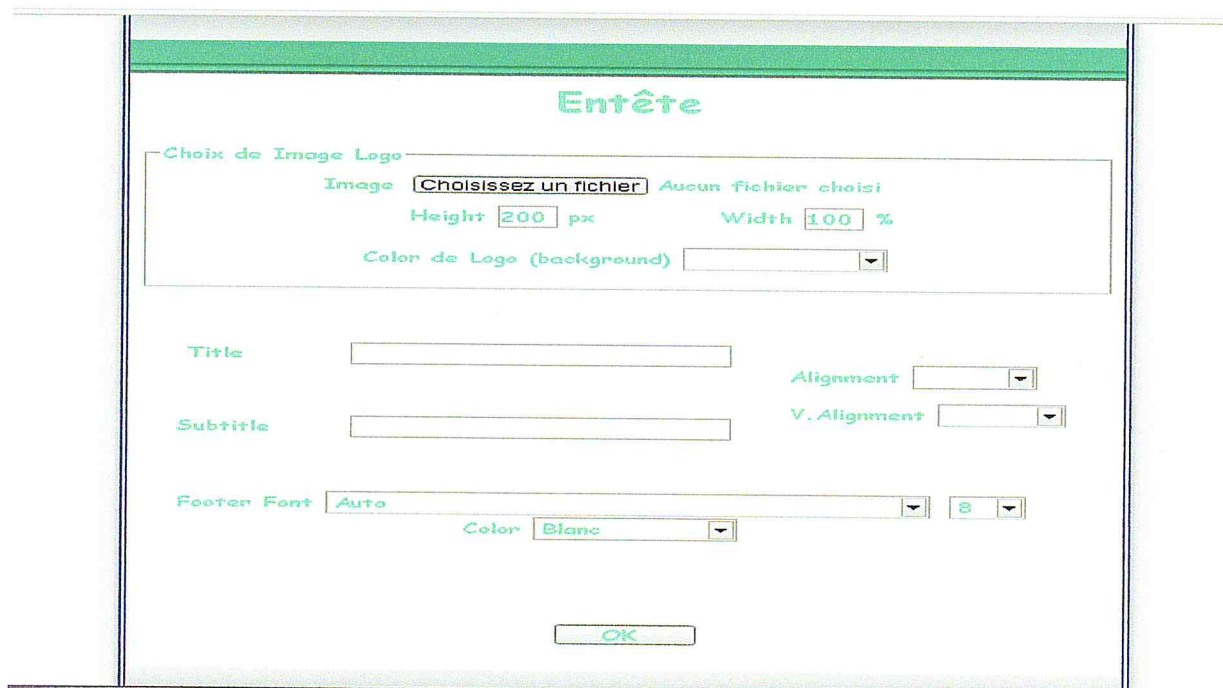


Figure 40 : Page pour générer le style de l'entête

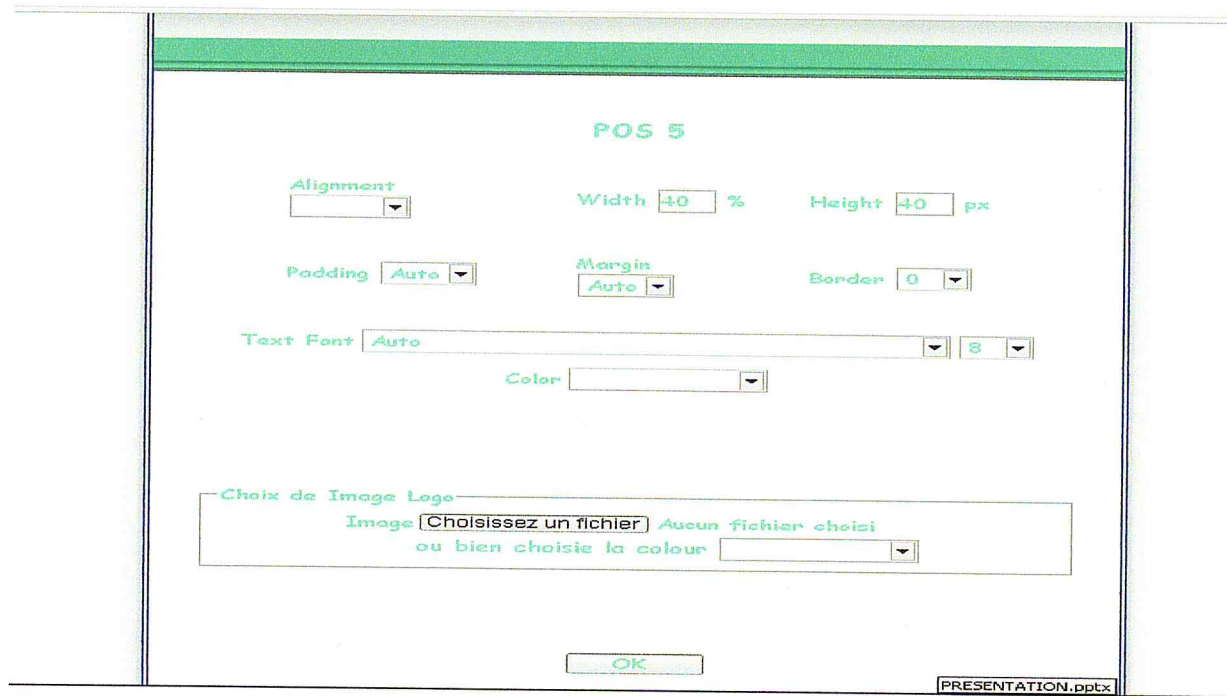


Figure 41 : Page pour générer le style de position 5

Conclusion :

Le thème pour ce mémoire est la conception et réalisation d'un générateur de charte graphique basé pour le CMS JOOMLA et PHP qui permet à un utilisateur de créer son propre template qui répond à son besoin sans avoir besoin à une connaissance technique en programmation.

Un site web possède un bon template est vraiment très nécessaire, il encourage les visiteurs à rester sur ce site et à l'explorer. Les visiteurs réguliers et les utilisateurs apprécient d'être sur un site avec un design superbe et pratique.

Enfin, la réalisation de ce projet de travail sur une durée limitée est un bon entraînement pour futur métier.



Références

[DOC_01] : «Mémoire PFE Master» Conception et réalisation d'un système d'information relatif à la mise en place d'un système de gestion de contenu web. << Mr. Ferhat ZAHOUANI et Mr. Mohammed GHAZAL OULAD DAOUD 2009. >>

[DOC_02] : «Mémoire PFE Master» Etude Comparative et mise en place d'une solution CMS (system de gestion de contenu). << ECHCHEIKH EL ALAOUI Adnane, JADDAD Alaa Eddine et SALIM Adil 2009/2010. >>

[DOC_03] : «Mémoire PFE Master» Conception et réalisation d'un système de gestion de contenu dédié aux conférences. <<Dellys Hachemi Nabil et Artabaz Saliha 2008/2009. >>

[DOC_04] : Thierry Duboise <<Tous pour réussir un site web, tome 2, Septembre 2010>>

[DOC_05] : Joomla! 2.5 Le Guide Pour Débutant<<Hagen Graf, Milena Mitova>>

[DOC_06] : Tutoriel de création de Template pour joomla 2.5.x <<Philippe STEINER>>

[WEB_01] :] <http://Articles sur Artisteer - Générateur de templates Joomla et Blogger, thèmes WordPress et Drupal> artisteer-france.fr.htm

[WEB_01.02] <http://Artisteer-France.fr/> - Part 2.htm

[WEB_01.03] <http://Artisteer-France.fr/> - Part 4.htm

[WEB-02] <http://fr.html.net/tutorials/html/lesson2.php>

[WEB-03] <http://www.graphic-evolution.fr/definition/css-40.html>

[WEB-04] <http://www.webmasterfrance.fr/Webmaster-php.html>

[WEB-05] <http://www.webmasterfrance.fr/Webmaster-javascript.html>

[WEB-06] http://www.craym.eu/tutoriels/developpement/site_local_avec_wamp.html

[WEB-07] <http://membres-liglab.imag.fr/Herve.Martin/HTML/SQL.htm>

[WEB-08] <http://joomla.buldozer.fr>