



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

UNIVERSITÉ ECHAHID HAMMA LAKHDAR
EL OUED

FACULTÉ DES SCIENCES ET DE TECHNOLOGIE

Mémoire de fin d'étude

LICENCE ACADEMIQUE

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Informatique fondamentale

Thème

**Créer un composant Joomla de gestion de
Catalogue pour une bibliothèque universitaire**

Présenté par:

DIB Zakia et KEDADRA Khaoula

Sous la supervision de :

Mr Othmani samir.Encadreur

Année universitaire 2014 – 2015

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicace

A la mémoire de mon père.

A ma mère.

A mon père.

A mon Frère et mes sœurs.

A ceux et celle qui ont été tout

Pour moi.

A toute ma famille et mes amis.

Zakia Dib

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrions témoigner toute notre reconnaissance.

Nous voudrions tout d'abord adresser toute notre gratitude à notre encadreur, Othmani Samir pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion.

Nous désirons aussi remercier les enseignants de la filière de d'informatique qui ont fourni les outils nécessaires à la réussite de nos études universitaires.

Nous voudrions exprimer notre reconnaissance envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Un Grand Merci
Pour Tous

ملخص :

نقترح في مشروع تخرجنا هذا انجاز مكون جوملا لموقع جامعة الوادي للبحث في فهرس كتب المكتبة المركزية بالجامعة باعتبارها واحد من اهم العوامل التي لها تأثير مباشر على موقع الجامعة والتي لا تقل اهميتها عن باقي العوامل الاخرى، مما يوفر تناسق بين موقع الجامعة والموقع الحالي الخاص بفهرس الكتب الذي يعمل بصفة منفصلة عن ذلك الموقع، لهذا السبب اخترنا ان نركز عملنا هذا على:

- تطوير و انشاء مكون جوملا من اجل موقع الجامعة خاص بإدارة فهرس المكتبة.

Résumé :

Nous proposons dans notre travail de fin d'étude une création d'un composant Joomla pour le web site de l'université d'El-Oued, et pour chercher dans le catalogue de la bibliothèque centrale de l'université que a un grand impact sur le site l'université et qui fournit une harmonie entre le site de l'université et le site actuel du catalogue que est séparé de site.

Et pour accomplir notre but, nous allons faire la mise au point :

★ Le développement et la création d'un composant Joomla dans le catalogue de la bibliothèque pour le site de l'université.

Abstract:

In our Graduation Project; we propose Joomla's Component for the web site of university of El-Oued and order to research in a catalogue of the Central Library and it is considered one of the most significant agents that has a great impact on the University's Web Site. Moreover, it gives a harmonious relation between the University's Web Site and the actual Site of the Library's Catalogue that works separately from that Site. For such reason, we have chosen to focus our work on:

★ The development and the creation of Joomla's Component for a University's Web Site; precisely; for a Library's Catalogue management.

Sommaire

Introduction General	
contexte général	01
Problématique.....	01
Objective.....	01
Organisation de mémoire.....	02

Chapitre I

Joomla

1. Introduction.....	04
2 Définition Joomla.....	04
3. Caractéristiques de Joomla	04
4. Installation de Joomla.....	05
5. Terminologie de Joomla.....	10
5.1. Menus.....	10
5.2. Contenus	10
5.3. Extensions.....	10
5.4. Template	11
5.5. Plugins.....	12
5.6. Module	12
5.7. Composants	14
6. Architecture (MVC) : (Composant, Module)	14
6.1. Modèle	14
6.2. Vue	15
6.3. Contrôleur	15
7. Conclusion	21

Chapitre II

La gestion de catalogue pour une bibliothèque universitaire

1. Introduction.....	23
2. Définition la bibliothèque.....	23
2.1. Rôle des bibliothèques universitaire.....	23
4. Catalogue	23
4.1 .Types de Catalogues	24
4.2 Les Différents Accès	24
4.3 Gestion des Notices Bibliographiques	24
5. Conclusion	24

Chapitre II

La Conception et L'implémentation du système

PARTIE1 : LA CONCEPTION

1. Introduction.....	26
2. Langage de conception (UML).....	26
3. Pourquoi nous utilisons UML ?.....	26
4. Acteur de Système	26
4.1 Identification des acteurs.....	26
4.2 Identification des activités.....	26
5. Les diagrammes de casd'utilisation	26
5.1. Description textuelle des cas d'utilisation	27
6. Les diagrammes de séquence	28
7. Diagramme des class.....	29
8. Conclusion.....	30

PARTIE 2 : LA REALISATION

1. Introduction.....	34
2 .Environnement detravail.....	34
2.1. Wamp Server.....	34
2.2. Langages utilisés.....	36
2.2.2. PHP (PHP hyper text préprocessor)	36
2.2.3. XML (eXtended Markup Language).....	36
3. Implémentation.....	37
3.1 La page Accueil.....	37
3.2. La Pag Recherche.....	37
3.3. La page d'Affichage des livres.....	38
3.4. La page d'Affichage d livre.....	38
4. Conclusion.....	39
Conclusion Général.....	40

Liste des figures

Liste des tableaux

Reference Bibliographique

1. Contexte générale :

L'enseignement et la recherche sont les deux activités principales de l'université que ne sucratent atteindre, d'une manière efficace, leurs objectifs sans l'appui de l'information Scientifique et Technique disponible au niveau des bibliothèques universitaires.

La bibliothèque fait partie intégrante de l'enseignement où la documentation joue un grand rôle. La bibliothèque universitaire située au cœur de l'enseignement supérieur et n'a jusqu'ici bénéficié de politiques adéquates, ni de moyens qui lui permettent de jouer pleinement le rôle social de développement qualitatif et quantitatif.

Dans ce cadre, s'inscrit notre projet de fin d'études qui consiste à réaliser un composant Joomla suivant les normes pour la Gestion de catalogue pour une bibliothèque universitaire.

2. Problématique :

La Bibliothèque Central de l'Université d'El-Oued a un noter de recherche de livres séparé site de l'Université. De pour recherche cet problème, nous allons voir une nanisère pour faire le site actuel de catalogue de la bibliothèque au forme d'un composant Joomla.

3. L'Objectifs :

L'objectif de ce projet est :

- ✓ nous allons concevoir un composant Joomla pas intégrer un composant dans Joomla (nous n'étés pas l'équipe de développement de Joomla).

4. Organisation de mémoire :

Le rapport suivent est constitué de trois chapitres :

✓ Chapitre 1 :

Le premier chapitre nous donnons une idée générale sur l'organisation, les procédures et les standards pour le site Joomla.

✓ Chapitre 2 :

Le deuxième chapitre nous discutons un système d'un catalogue d'une bibliothèque universitaire.

✓ Chapitre 3 :

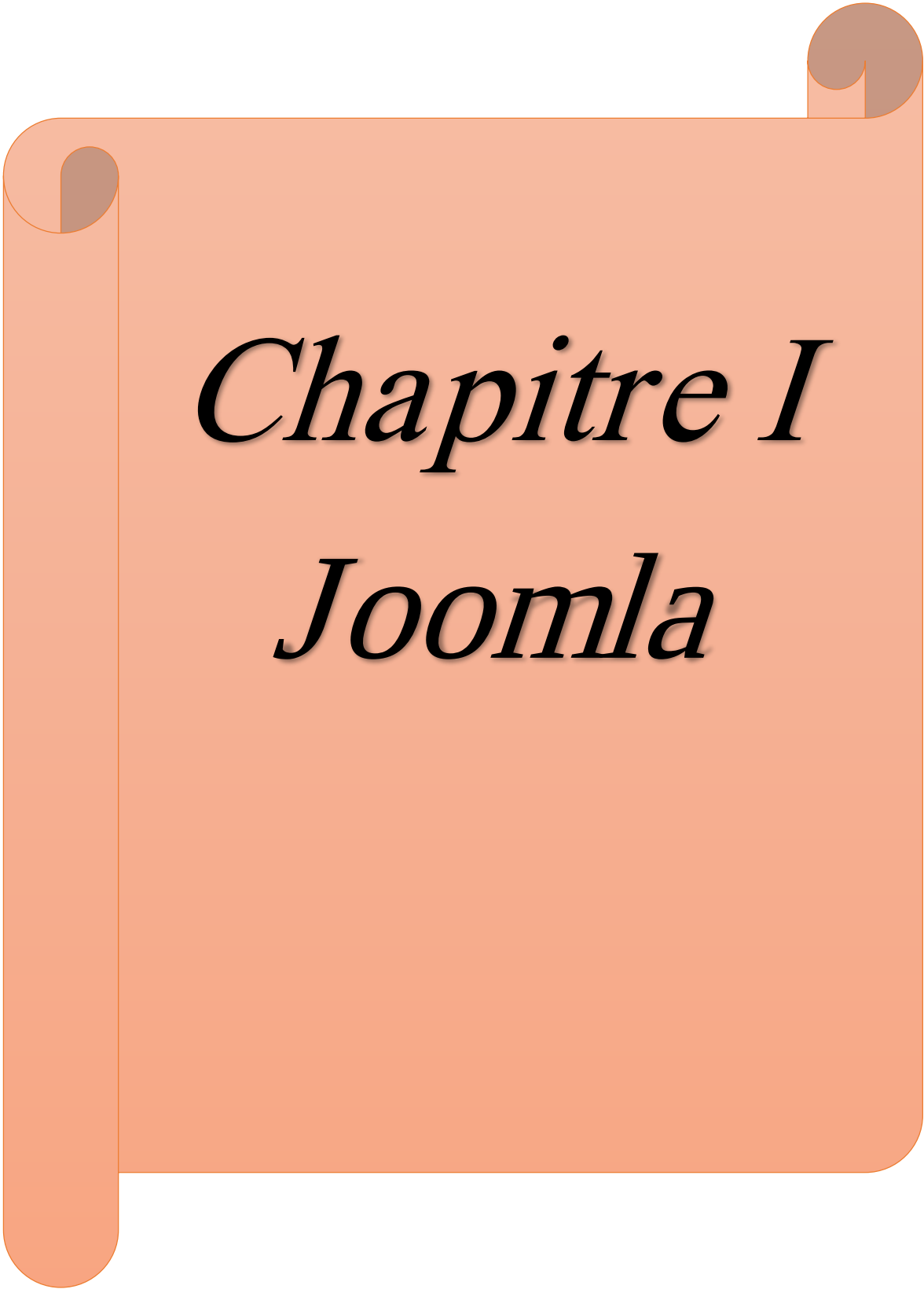
- ❖ Le troisième chapitre concerne la conception du système à l'aide des diagrammes UML :

1. Diagramme de cas d'utilisation.
2. Diagramme de séquence.

3. Diagramme de classe.

- ❖ Le dernier chapitre sera consacré à la réalisation de notre travail, comportant la conception et la mise en œuvre d'un composant Joomla pour la gestion catalogue de bibliothèque universitaire et à présenter les interfaces et les fonctionnalités de notre système.

Enfin, nous concluons ce rapport par une conclusion générale.

The background of the page is a light orange scroll with rounded corners and a vertical strip on the left side. The text is centered on the scroll.

Chapitre I

Joomla

1. Introduction :

Dans ce chapitre on va identifier Joomla, qui présente un système de management de contenu d'un site web et à savoir les systèmes des présentations de contenu .les gestions des extensions, sur tout l'installation, la personnalisation et nous avant remarqué sur tous les catégories : module, composant, plugin en détails sur l'objectif dans ce projet.

2. Définition Joomla :



Joomla est un système de gestion de contenu (en anglais, CMS, pour Content Management system) Open Source sous licence GNU/GPL créé par une équipe internationale de développeurs récompensée à maintes reprises.

Joomla est chapeauté par Open Source Matters, Inc., organisation à but non lucratif basée aux Etats- Unis dont la mission est de fournir une structure légale et financière au projet. [1]

3. Caractéristiques de Joomla :

Résumée des caractéristiques actuelles de Joomla :

- code source disponible ;
- grande base installée d'utilisateurs et de développeurs ;
- système de gestion de workflow simple ;
- système de diffusion des contenus ;
- gestionnaire de téléchargement et la gestion des fichiers distants ;
- exploitation des contenus dans le format RSS ;
- corbeille pour les contenus effacés ;
- adresses URL compatibles avec les moteurs de recherche ;
- gestion de bannières publicitaires ;
- frontal et administration disponibles en de nombreuses langues ;
- interface d'administration séparée de l'accès Web normal ;
- langage de macros pour traiter les contenus (Mambos) ;
- mécanisme de cache pour assurer une génération rapide des pages choisies ;
- installation facile des modèles et composants complémentaires ;
- système de modèles simple mais complet (HTML, CSS, PHP) ;
- groupes d'utilisateurs hiérarchisés ;
- statistiques de fréquentation simples ;
- éditeur de contenus visuel (WYSIWYG) ;
- gestion de sondages ;
- mécanisme d'évaluation des contenus.

Vous trouverez de nombreuses extensions sur le site <http://extensions.joomla.org>.

Citons notamment

- gestion de forums ;
- gestion de galeries de photos ;
- système de gestion documentaire ;
- calendriers.

Et presque 2 000 autres...

4. Installation de Joomla :

Avant d'installer Joomla, nous devons le télécharger. Il n'existe qu'un seul site officiel sur lequel nous pouvons télécharger Joomla. Cependant, la communauté Française propose des packs entièrement Francisés. Si nous sommes des débutants, il est conseillé d'utiliser ces packs car ils contiennent également les données. Ces données d'exemple nous aideront à nous familiariser avec le CMS. Dans ce cas nous devons préalablement installer un serveur local sur notre ordinateur. Il existe plusieurs serveurs différents : Easyphp, Wampserver, Wabmatrix...etc. Dans ce cas, nous aurons simplement besoin de nos identifiants pour nous. Connecter par FTP et à la base de données. Une fois Joomla téléchargé, nous devons dé-zipper le pack et en transférer le contenu par FTP à l'aide d'un client FTP sur le serveur mis à disposition par notre hébergeur. Une fois le transfert terminé, nous accédons à l'URL de notre nom de domaine et nous arrivons directement sur la page d'installation. [2]

❖ Les septes étapes de l'installation :

Étape 1 : Choix de La langue :

Joomla nous propose le Français comme langue par défaut. Si cela nous convient, appuyons sur le bouton Suivant en haut à droite. Sinon, nous choisissons une autre langue avant d'appuyer sur le bouton Suivant.



Fig1 : Choix de La langue.

Étape 2 : Pré-installation :

A cette étape est effectué un contrôle de la configuration du serveur web. Pour une Installation locale sur notre micro-ordinateur, ne nous tenons pas compte des avertissements en rouge pour l'instant. Nous continuons en cliquant sur le bouton Suivant.



Fig2 : Pré-installation.

Étape 3 : acceptation de la licence :

Nous Prenons la connaissance de la licence d'utilisation GNU/GPL puis cliquons sur le bouton Suivant. La version Française (non officielle) est consultable à cette adresse: <http://www.gnu.org/licenses/translations.fr.html>



Fig3 : acceptation de la licence.

Étape 4 : base de données :

A cette étape l'assistant d'installation va automatiquement créer la base de donnée utilisée par Joomla. Renseignez les Paramètres de base selon les recommandations fournies ci-après Puis cliquez sur le bouton Suivant.

Fig4 : base de données.

Étape 5 : Configuration FTP :

Joomla peut prendre en compte le transfert de fichiers, mais en local vous n'en n'avez pas besoin ... Alors cliquez sur le bouton Suivant.

Fig 5 : Configuration FTP.

Étape 6 : Configuration générale :

- ✓ Dans le premier bloc, saisissez le nom du site en clair : c'est ce nom qui apparaîtra dans le titre de la page de votre navigateur.
- ✓ Dans le second bloc, saisissez l'adresse email de l'administrateur ; ceci vous Permettra de tester certaines fonctions comme l'envoi de mail vers un contact.
- ✓ Saisissez aussi le mot de passe pour l'administration du site.

Attention Il s'agit du mot de passe associé au compte Admin, le super Administrateur Sans lui vous ne pourrez pas accéder à l'interface d'administration de Votre site.

- ✓ dans le troisième bloc, cliquez sur le bouton Installer des données d'exemple et cliquez sur le bouton Suivant.

Fig 6 : Configuration.

Étape7 : Terminer :

Vous êtes arrivés à la dernière étape de l'assistant d'installation. Avant de cliquer sur les boutons Site ou Admin, un message vous invite à supprimer le répertoire d'installation Joomla!... Supprimez donc le répertoire site Joomla!/installation avec le gestionnaire de Fichiers et retournez dans votre navigateur Web et cliquez sur le bouton Site pour visiter votre tout nouveau site Web.

Fig7 : Terminer.

5. Terminologie de Joomla.

5.1. Menus :

Quand on clique sur Menus dans la console d'administration, on a accès aux liens vers la «Gestion des menus» (affiche la liste complète de tous les menus existants), la «Corbeille de menu» (pour supprimer un menu), «main Menu» (le menu principal et tous les menus secondaires créés par le concepteur du Site.

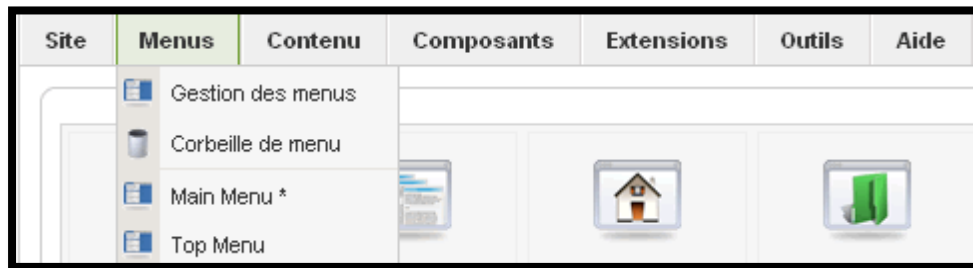


Fig8 : La gestion des Menus

5.2. Contenus :

Gestion des sections, gestion des catégories, gestion des articles, corbeille article.



Fig 9 : La gestion des contenus.

5.3. Extensions :

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 4 catégories : les composants, les modules, les plugins et les Template. La version standard de Joomla Intègre un certain nombre d'extensions mais vous en trouverez quelques milliers sur le net pour personnaliser votre site. [3]

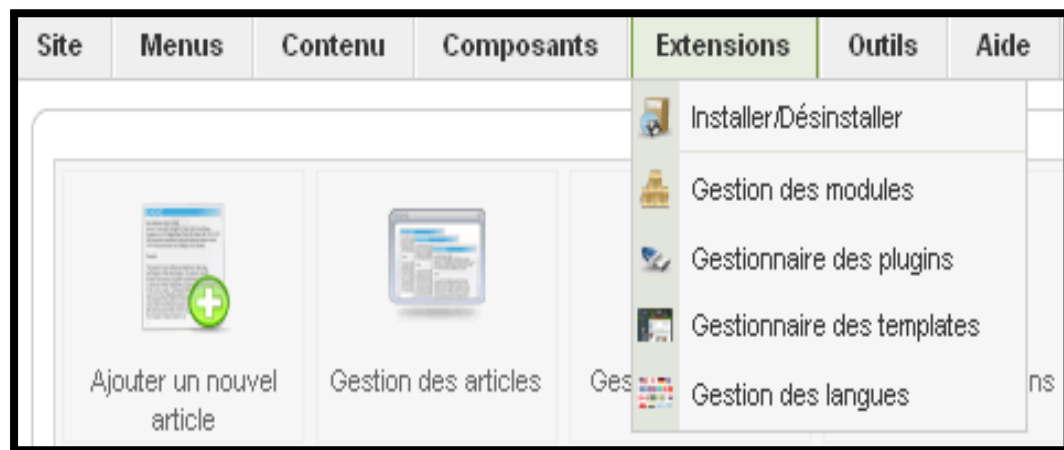


Fig10 : La gestion des extensions.

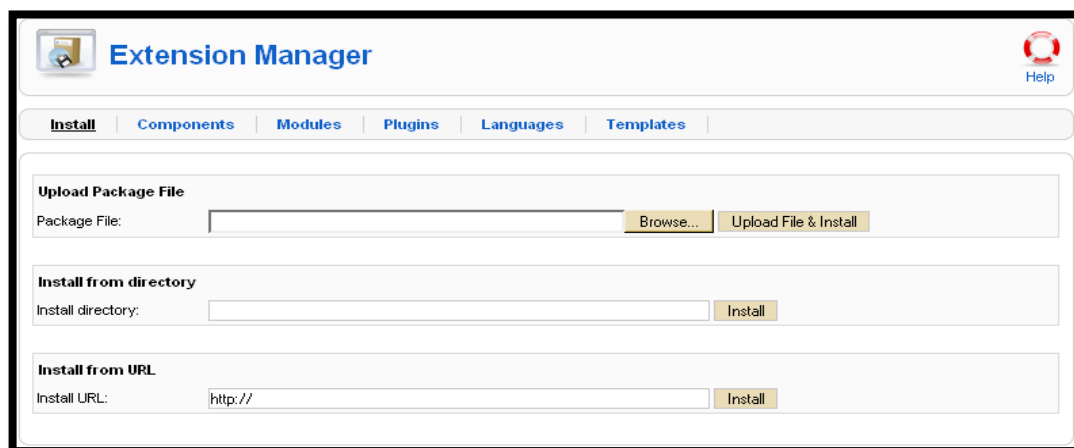


Fig 11 : Install, Components, Modules, Plug-ins, Langages, et Template

5.4. Template :

Le Template correspond à toute la partie présentation de la page Web. Il en existe des centaines qui circulent sur Internet et la plupart sont disponibles gratuitement. Les Template gèrent la position des différents modules, l'affichage des différents menus, sections, catégories, articles et le design de la page Web (couleurs, images, présentation, etc.).

Comme les modules, il existe 3 différents types de Template :

- Les Template préinstallés.
- Les Template installés par l'administrateur
- Les Template créés par l'administrateur.

Un Template Joomla est toujours composé d'une feuille de style CSS et d'un répertoire d'images ainsi qu'un programme en PHP intitulé « index. PHP », lequel va définir la présentation par défaut du site Web. Il existe aussi des Template permettant de modifier la présentation des pages de l'administration (back end) du site, mais ceux-là sont beaucoup moins nombreux que les Template du site...

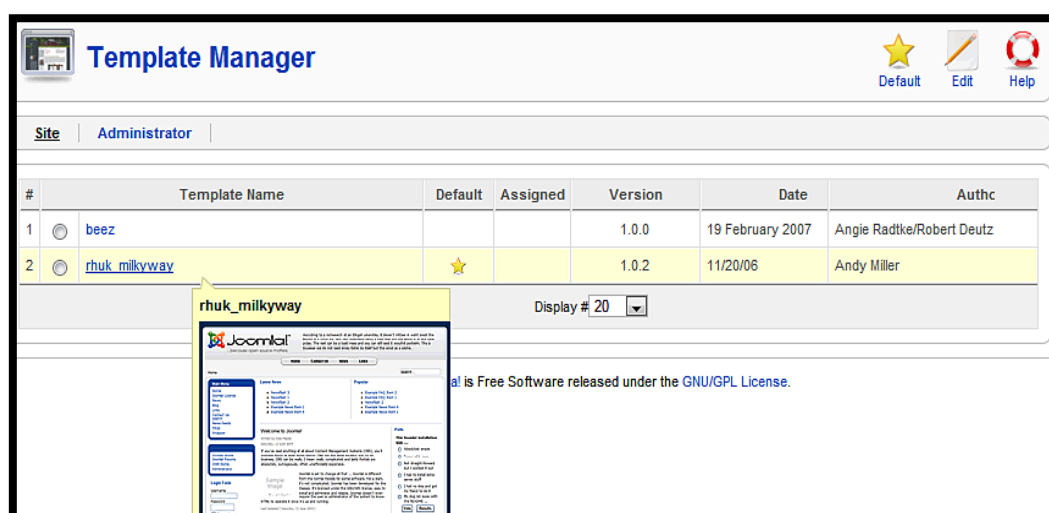


Fig 12 : Template Manager.

5.5. Plugins :

Un plugin est un bloc de code qui vient se brancher en un point précis de l'infrastructure de Joomla pour en modifier le fonctionnement. Vous pouvez par exemple utiliser un plugin dans un contenu pour charger le contenu d'un module dans un texte. Pour pouvoir étendre la fonction de recherche d'un site Web à un composant supplémentaire, il faut lui associer un plugin. Les plugins peuvent parfois être vus comme une langage de macros pour Joomla.

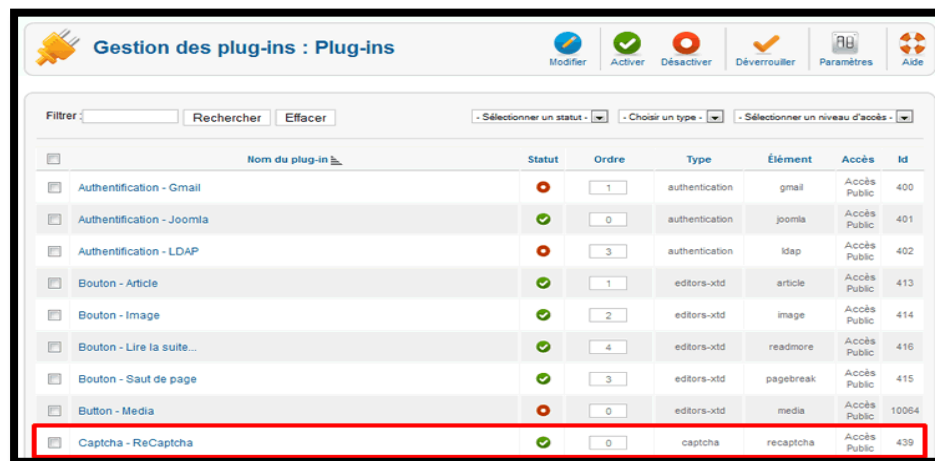


Fig 13 : Ajout d'un Plug-in

5.6. Module :

Un module est une fonctionnalité qui affiche quelque chose dans le site, par exemple un sondage, des news, un questionnaire d'identification ... Certains modules fonctionnent parfois avec un composant.

Un module est un bloc de la page Web. Il existe différents blocs dans une page qui vont permettre à l'utilisateur de choisir l'endroit où chaque module de son site doit

être placé. Le nombre de positions diffèrent en fonction du Template utilisé. Les positions

Les plus reconnaissables sont les colonnes de gauche et de droite d'une page Web. Il existe trois types de modules :

- Les modules créés par l'administrateur.

Pour définir la place de chaque module sur la page souhaitée, il faut pour cela établir l'ordre des modules dans chaque position auxquelles ils appartiennent. Par exemple, si l'on souhaite placer différents modules dans la colonne de gauche, il suffit de donner le numéro 1 au module que l'on souhaite voir le plus en haut à gauche. Le numéro 2 sera en-dessous du module numéro 1 et ainsi de suite pour tous les modules.

- Les modules préinstallés par Joomla : Menu principal, module de présentation.
- Les modules installés par l'administrateur.

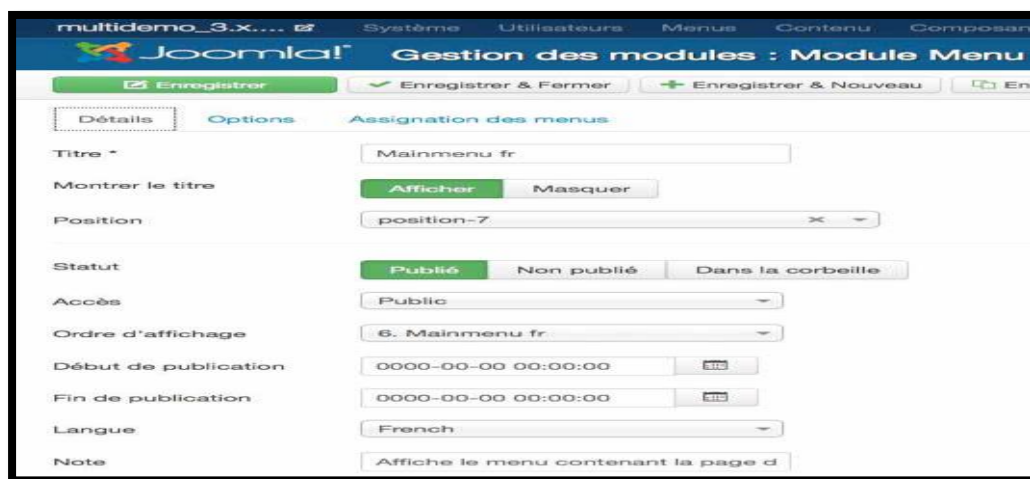


Fig 14 : la gestion des modules

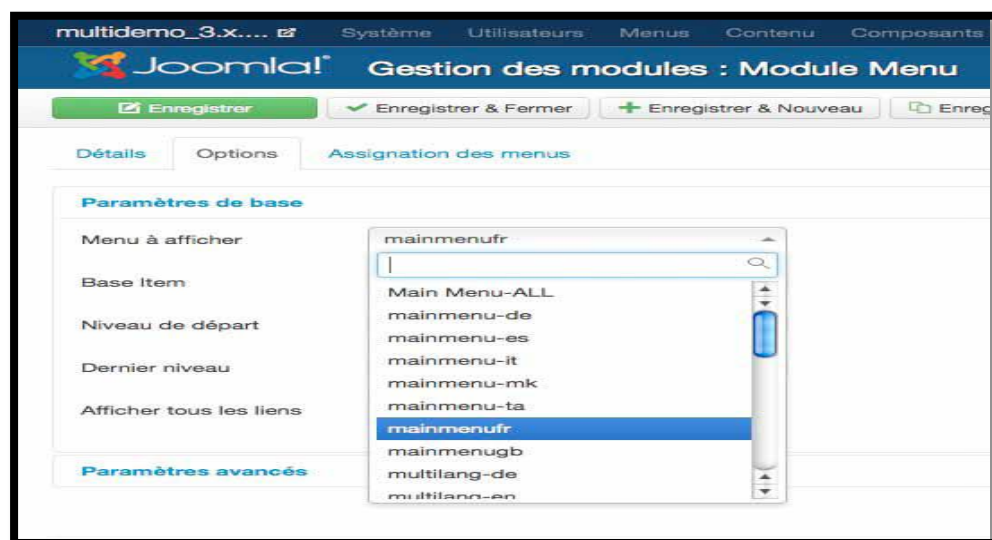


Fig 15 : la gestion modules

5.7. Composants :

Un composant est une application qui est intégrée à Joomla et qui dispose de sa propre interface de configuration sur la console d'administration de Joomla. Les composants font aussi partie du contenu de Joomla. Il existe différentes sortes de composants. Le composant « Contact », par exemple, va permettre à n'importe quelle internaute de communiquer avec l'un des administrateurs du site. Et ce composant va permettre à l'administrateur d'enregistrer et d'afficher ses coordonnées sur le site afin que les internautes puissent communiquer avec lui. Ce composant est préinstallé par Joomla. Il reste à l'administrateur d'afficher ou non ce composant et de mettre à disposition des internautes ses données personnelles. Même s'il y a déjà des composants installés par défaut par Joomla, l'administrateur peut en intégrer de nouveaux sur son site. Par exemple, il existe des composants très utiles que l'on peut télécharger gratuitement sur Internet tel qu'un calendrier d'événements. Pour installer de nouveaux composants, il suffit juste de se connecter à la console d'administration et de se rendre sous l'onglet « Installation ». Ensuite, il suffit de cliquer sur « Composants », et de rentrer le chemin absolu jusqu'à l'archive du composant correspondant et ensuite de cliquer sur « Installer ». Si l'opération est un succès, l'administrateur pourra alors gérer son composant à partir de la console d'administration, sous l'onglet « Composants ». [4]

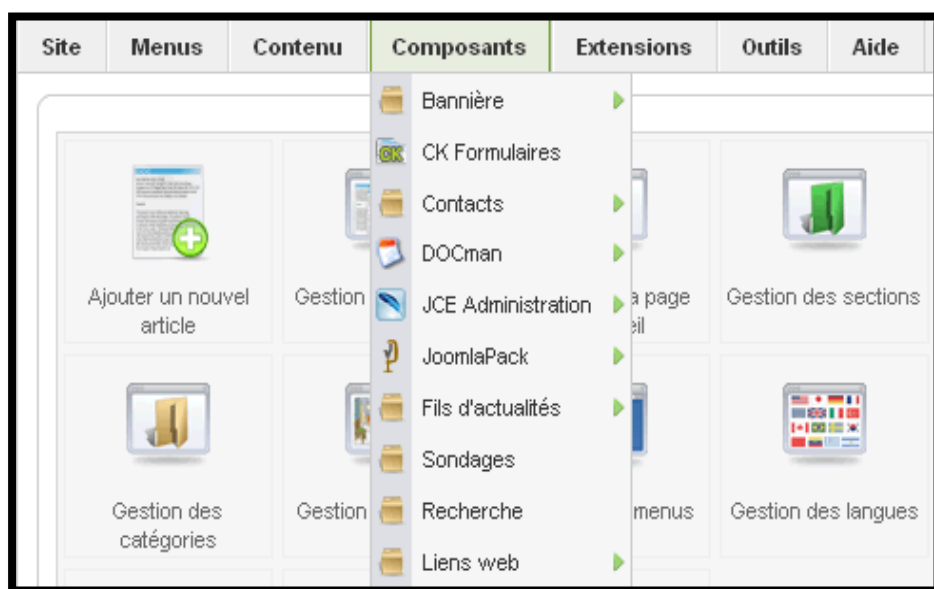


Fig16 : La gestion des composants.

6. Architecture (MVC) : (Composant, Module)

L'organisation d'une interface graphique est délicate. L'architecture MVC ne prétend pas en éliminer tous les problèmes, mais fournit une première approche pour ce faire. Offrant un cadre normalisé pour structurer une application, elle facilite aussi le dialogue entre les concepteurs. L'idée est de bien séparer les données, la présentation et les traitements. Il en résulte les trois parties énumérées plus haut : le modèle, la vue et le contrôleur.

6.1. Modèle :

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées Par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes Standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Le modèle ne contient aucun lien direct vers le contrôleur ou la vue. Sa communication avec la vue s'effectue au travers du patron Observateur. Le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir tous les cours d'une Salle, tous les cours d'une personne ou tous les cours d'un groupe de TD.

6.2. Vue :

Ce avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur. Plusieurs vues peuvent afficher des informations partielles ou non d'un même modèle. Par exemple si une application de conversion de base a un entier comme unique donnée, ce même entier peut être affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir à l'utilisateur la possibilité de changer de vue.

6.3. Contrôleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de La vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est lui « observable ». Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande. Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base Des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne. Quand un même objet contrôleur reçoit les événements de tous les

composants, il lui faut déterminer l'origine de chaque événement. (Un énorme Ce tri des événements peut s'avérer fastidieux et peut conduire à un code peu élégant switch). C'est pourquoi le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements D'une partie des composants. [5]

❖ Pour les articles homonymes, voir MVC.

Le patron modèle-vue-contrôleur (en abrégé MVC)

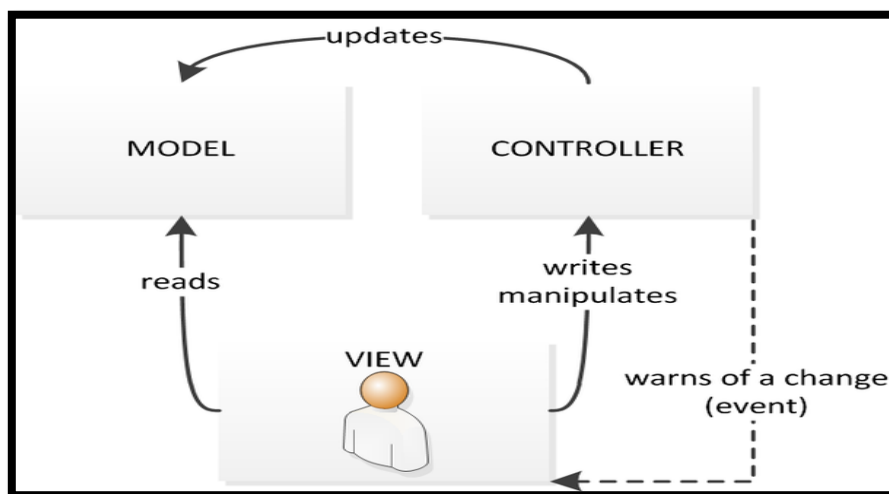


Fig17 : la gestion (MVC).

Le schéma de cette figure résume les différentes interactions entre le modèle, la vue et le contrôleur. Les lignes pleines indiquent une dépendance (importation) tandis que les pointillés sont des Événements (cf. observateur). De l'anglais model-view-Controller), tout comme les patrons modèle-vue-présentation ou Présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques Liées aux différents composants au sein de leur architecture respective. Ce paradigme regroupe les fonctions nécessaires en trois catégories :

1. un modèle (modèle de données).
2. une vue (présentation, interface utilisateur).
3. un contrôleur (logique de contrôle, gestion des événements, synchronisation).

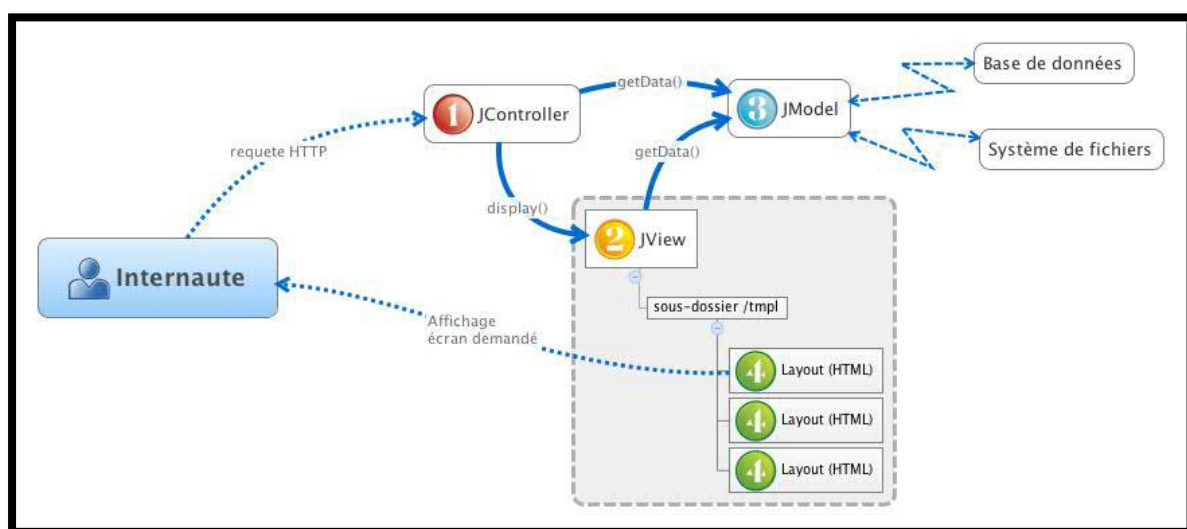


Fig 18 : Structure du composant(MVC).

- ❖ Les scripts seront regroupés dans un ZIP selon une arborescence spécifique :

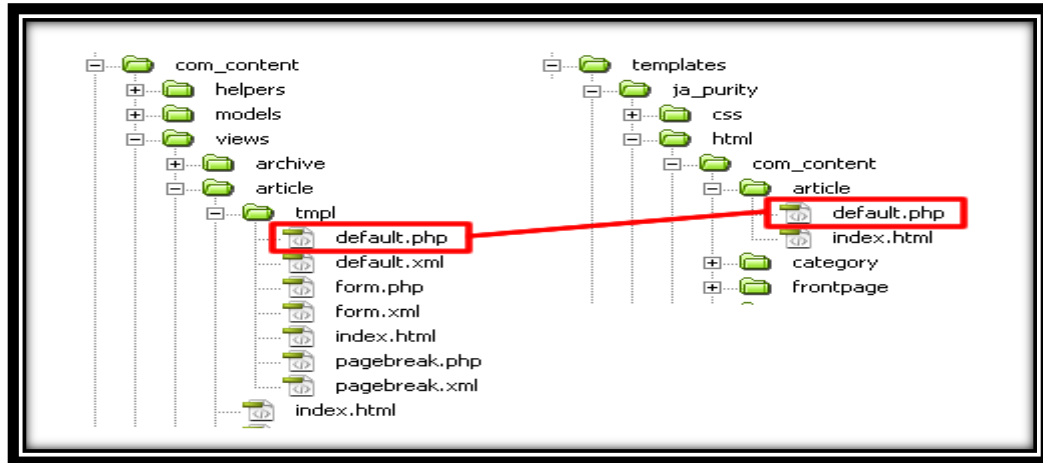


Fig 19 : Composant (Mvc).

Le contrôleur principal, la vue, le Modèle :

```

?>php
//No direct access
defined('_JEXEC') or die('Restricted access');

//import Joomla table library
jimport('joomla.database.table');

/**
 * Hello Table class
 */
class HelloWorldTableHelloWorld extends JTable
{
    /**
     * Constructor
     *
     * @param object Database connector object
     */
    function __construct($db)
    {
        parent::__construct('#helloworld', 'id', 'db');
    }

    /**
     * Overloaded bind function
     *
     * @param array named array
     * @return null|string null is operation was satisfactory, otherwise returns
     * an error
     * @see JTable:bind
     * @since 1.5
     */
    public function bind($array, $ignore = false)
    {
        if (isset($array['params']) && (!isset($array['params'])) {
            //Convert the params field to a string.
            $parameter = new JRegistry;
            $parameter->loadArray($array['params']);
            $array['params'] = (string)$parameter;
        }
        return parent::bind($array, $ignore);
    }
}

```

```

*Overloaded load function
*
@ *param    int $pk primary key
@ *param    boolean $reset reset data
@ *return    boolean
@ *see JTable:load
/*
public function load($pk = null, $reset = true (
)
    if (parent::load($pk, $reset)) {
        //Convert the params field to a registry.
        $params = new JRegistry ;
        //      loadJSON is @deprecated 12.1 Use loadString passing JSON
        //      as the format instead.
        $ //      params<-loadString($this-<-item<-params, 'JSON'('
        " //      item" should not be present.
        $      params<-loadJSON($this-<-params'(
            $this-<-params$ = params;
            return true;
        {
        else
        }
            return false;
        {
    {
    {

```

```

<?php
//No direct access to this file
defined('_JEXEC') or die('Restricted access');

//import Joomla view library
jimport('joomla.application.component.view');

/**
 * HTML View class for the HelloWorld Component
 */
class HelloWorldViewHelloWorld extends JView
{
    //Overwriting JView display method
    function display($tpl = null)
    {
        //Assign data to the view
        $this->item = $this->get('Item');

        //Check for errors.
        if (count($errors) > 0)
        {
            JError::raiseError(500, implode('<br>', $errors));
            return false;
        }
        //Display the view
        parent::display($tpl);
    }
}

```

```

?>php
//No direct access to this file
defined('_JEXEC') or die('Restricted access');

//import Joomla modelitem library
jimport('joomla.application.component.modelitem');

/**
 * HelloWorld Model
 */
class HelloWorldModelHelloWorld extends JModelItem
{
    /**
     * @var object item
     */
    protected $item;

    /**
     * Method to auto-populate the model state.
     *
     * This method should only be called once per instantiation and is designed
     * to be called on the first call to the get State () method unless the model
     * configuration flag to ignore the request is set.
     *
     * Note. Calling get State in this method will result in recursion.
     *
     * @return void
     * @since 2.5
     */
    protected function populateState ()
    {
        $app = JFactory::getApplication();
        //Get the message id
        $input = JFactory::getApplication->input;
        $id = $input->getInt('id');
        $this->setState('message.id', $id);

        //Load the parameters.
        $params = $app->getParams();
        $this->setState('params', $params);
        parent::populateState();
    }
    ....etc.

```

7. Conclusion :

Espérant que notre travail et cet effort modeste d'avoir satisfait l'objectif Tout en recommandant Joomla dans les travaux universitaires et la création des sites web.

Chapitre II

La gestion de
catalogue pour une
bibliothèque
universitaire

1. Introduction :

L'objectif de ce chapitre est de présenter bibliothèque universitaire dans le cadre de notre étude. Ce chapitre concerne précisément la présentation de l'organisation des études au sein d'un temple du savoir ; socle du donner et du recevoir universitaire. Se rapporte aux fonctions classiques caractéristiques de toute bibliothèque comme la gestion, la référence, le choix et l'acquisition des documents, leur traitement, ainsi qu'à la prestation de services, à l'organisation de l'information et à la diffusion des connaissances.

2. Définition la bibliothèque :

La bibliothèque est une collection organisée de livres. Il existe des bibliothèques privées et des bibliothèques proposent souvent d'autres documents (journaux, patriotiques, enregistrements sonores, enregistrements vidéo, cartes et plans, partition ...) ainsi que des accès à l'Internet et sont parfois appelées médiathèques ou informatiques. La majorité des bibliothèques (municipales ou universitaires), autorisent le prêt de leurs documents et autres (par exemple la bibliothèque publique d'information) leur consultation sur place seulement. [6]

3. Rôle des bibliothèques universitaires :

Le rôle premier des bibliothèques est d'accompagner et de soutenir les activités d'enseignement et de recherche. Plus importantes que jamais dans le monde de l'information numérique, elles identifient, acquièrent et rendent accessibles les ressources documentaires indispensables aux étudiants et aux chercheurs. Elles ont également pour mission de former à l'utilisation des ressources documentaires. Gardiennes de la mémoire et du patrimoine documentaire de l'établissement, elles occupent une place centrale dans la vie sociale et culturelle d'une université : la bibliothèque universitaire est non seulement un lieu de travail, d'étude et de recherche, mais aussi de convivialité. [7]

4. Catalogue :

Un catalogue de bibliothèque est destiné à faciliter la recherche des livres qui sont contenus dans une bibliothèque, ou de plusieurs bibliothèques dans le cas des catalogues collectifs et des catalogues virtuels. Dans sa version informatisée, la plus courante, le catalogue est une base de données signalant tout ou partie des documents disponibles dans une ou plusieurs bibliothèques. Le catalogue est destiné à identifier les documents (décrits dans des notices) et à faciliter leur recherche ou leur localisation. Le catalogage consiste à analyser le document en tant que support. Le catalogage fait partie des deux opérations intellectuelles nécessaires à la constitution d'un catalogue : catalogage (Description physique) et indexation (Description intellectuelle). Traditionnellement, on distingue le catalogue, instrument de recherche documentaire, de l'inventaire, liste des documents possédés par la bibliothèque classés par ordre d'entrée et destinée à attester la propriété de la bibliothèque sur ces documents. Cette

distinction a beaucoup perdu de sa pertinence avec l'informatisation des catalogues, la même base de données servant à produire à la fois l'inventaire et les différents accès publics du catalogue.

4.1. Types de catalogues :

- Un catalogue local décrit les documents d'une seule bibliothèque.
- Un catalogue commun référence les documents de plusieurs bibliothèques relevant d'un même organisme, comme une bibliothèque centrale et des bibliothèques de quartier.
- Un catalogue collectif décrit les collections de plusieurs bibliothèques indépendantes
- Un catalogue virtuel est un outil permettant d'interroger plusieurs catalogues simultanément.

4.2. Les différents accès :

Un catalogue peut permettre plusieurs types d'accès aux documents.

❖ Sujet

Le catalogue alphabétique matières : permet un accès par sujet, exprimé sous forme de vedette-matière. C'est le plus difficile

❖ Auteur

Le catalogue alphabétique auteurs : permet une recherche par nom d'auteur. Lorsque le catalogue n'est pas informatisé, il est de coutume de classer à leur Titre dans ce catalogue le document n'ayant par d'auteur identifiable. Le catalogue est alors appelé alphabétique auteurs et titres d'anonymes

4.3. Gestion des notices bibliographiques :

- Catalogage de l'ensemble des types de documents (livres, mémoires...etc.) Avec la personne au minimum de champs (ISBN, titre, auteur, fonction d'auteur, éditeur, année d'édition, collection, Notes, Résumé, Indices systématiques).
- Dé doublonnage systématique des documents par leur ISBN (International Standard Bibliographic Description) ou par des auteurs, titre. [8]

5. Conclusion :

Nous avons abordé, dans ce chapitre, les normes et les standards utilisés pour l'organisation des bibliothèques pour en tirer profit. Nous avons parlé des procédures de gestion de catalogue adoptées par la bibliothèque de notre centre universitaire. Dans les chapitres que se suivent, nous allons aborder des concepts plus détaillés de notre travail.

Chapitre III

*La conception et la
réalisation du
système*

PARTIE1:LA CONCEPTION

1. Introduction :

Ce chapitre a pour but d'analyser les fonctionnalités de l'application, de définir les droits d'accès pour l'acteur et de présenter les différents diagrammes et modèle de Conception en utilisant le langage UML. Nous allons procéder comme suit : En premier lieu, nous présentons les approches de conception. Dans un second temps, nous présentons le cycle de développement d'un logiciel. Enfin et dans la dernière partie nous présentons les différents diagrammes relatifs à chaque acteur.

2. Langage de conception (UML) :

Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié" est une notation permettant un problème de façon standard. Pour cela UML se base sur une sémantique précise conception de programmes, ainsi que leur description pour des non informaticiens. Ce mode de conception repose donc sur les principes de la programmation objet. UML modélise les objets et leurs liens au moyen de vues constituées de diagrammes. [9]

3. Pourquoi nous utilisons UML ?

Depuis plus de 30 ans, la conception des bases de données est réalisée à l'aide du modèle entité-association (Modèle de P. Chen et le MCD de Merise). Ce modèle conceptuel a fait ses preuves et la plupart des outils informatiques de modélisation (destinés principalement aux concepteurs français) l'utilisent encore aujourd'hui.

La notation UML (Unified Modeling Language) s'est imposée depuis quelques années pour la mode modélisation et le développement écrites dans un langage objet (C++ et Java principalement). Cette notation n'a pas été initialement pensée pour les bases de données mais elle permet d'offrir un même formalisme aux concepteurs d'objets et aux concepteurs de base de données.

4. Acteurs de Système :

4.1 Identification des acteurs :

Un acteur représente un rôle joué par une entité externe qui interagit directement avec le système étudié. On distingue deux intervenants qui interagissent avec l'interface : l'adhérent (Etudiant ou Enseignants).

4.2 Identification des activités :

Décrit l'ensemble des activités effectuées par les acteurs du système en les décomposant en sous activités et en spécifiant les contraintes relatives à l'enchaînement de ces activités.

❖ L'Adhérent recherche :

Recherche simple des livres : pour rechercher un ou plusieurs livres
(Par titre, par auteur, par sujet).

Recherche avancée des livres : pour rechercher un éditeur ou plusieurs livres

(Par titre, Par auteur, Par sujet, Par éditeur, Par Année, Par type de document par catégories) Ces activités peuvent être résumées par le diagramme suivant :

5. Les diagrammes de cas d'utilisation :

On utilise les diagrammes des cas d'utilisation pour représenter et structurer au niveau conceptuel, les besoins de l'utilisateur et les objectifs correspondants du système. Le but est d'identifier les acteurs du domaine et leurs interactions avec l'interface. Ce diagramme permet de déterminer le modèle objet sur lequel le système reposera. [10]

5.1. Description textuelle des cas d'utilisation :

- ✓ Diagramme de cas d'utilisation pour la Adhérent :

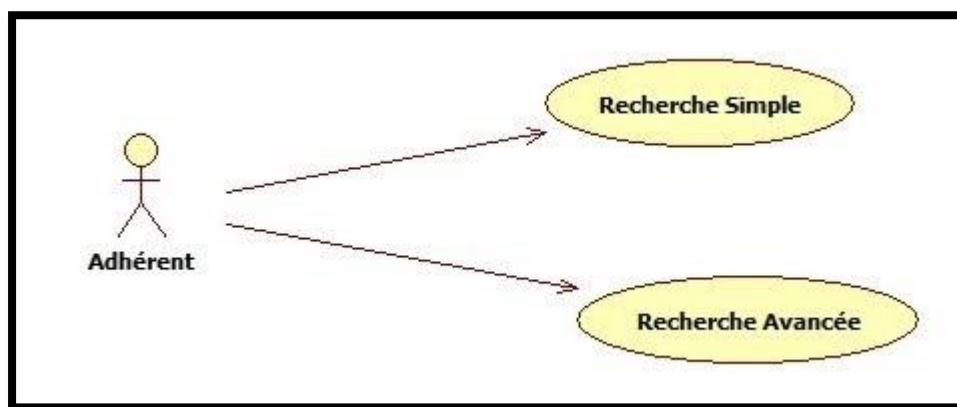


Fig20 : Diagramme de cas d'utilisation pour la Adhérent.

- ✓ Cas d'utilisation <<Recherche Simple>> :

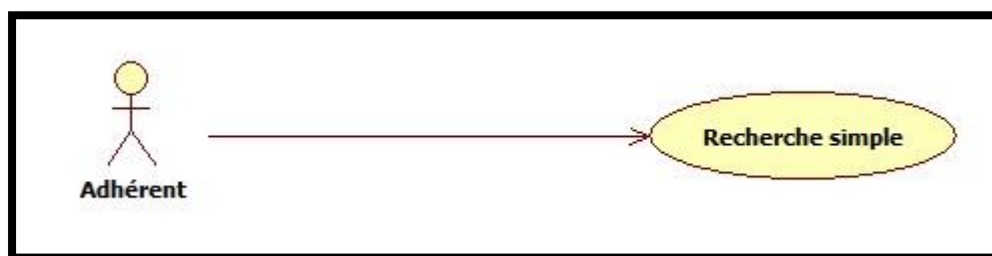


Fig 21 : Diagramme de cas d'utilisation "Recherche Simple "

- ✚ **Tite** : Recherche simple.
- ✚ **Acteur principal** : Adhérent.
- ✚ **Objectif** : faire une recherche un livre par (titre, auteur, sujet).
- ✓ Cas d'utilisation <<Recherche Avancée>> :

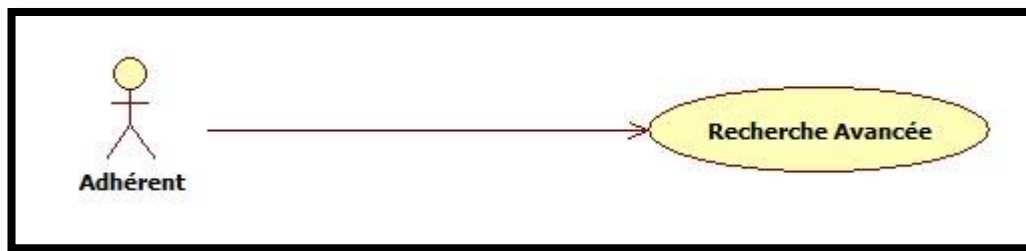


Fig22 : digramme des cas d'utilisation "Recherche Avancée"

🚦 **Titre : Recherche Avenacée.**

🚦 **Objectif :** faire une recherche un livre par (titre, auteur, sujet, Par éditeur, Par Année, Par type de document).

Scénarios :

Action des Acteurs	Réaction du Système
1) Adhérent dépend de connecte au système.	2) le système affiche le formulaire de recherche.
3) sélectionner type de recherche	4) le système l'affiche le formulaire de recherche
5) adhérent saisit les informations relative à la recherche.	6) le système affiche le résultat de recherche.
7) l'adhérent sélectionne les livres de recherche.	8) afficher le résultat de recherche (sommaire, image.....)

Tableau1 : Recherche.

6. Les diagrammes de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling langage. [11]

Dans ce qui suit, nous présentons toutes les fonctions de l'utilisateur à travers des

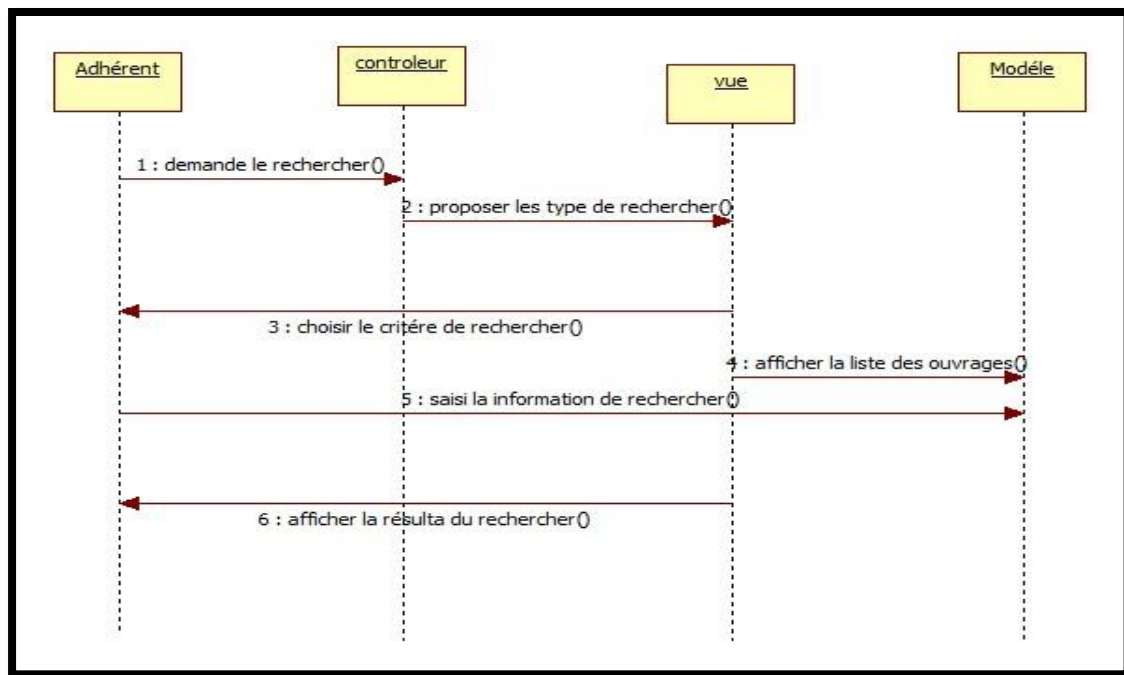


Fig 23 : Digamma de séquence pour recherche.

7. Diagramme des class :

Le diagramme de la classe en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. [12]

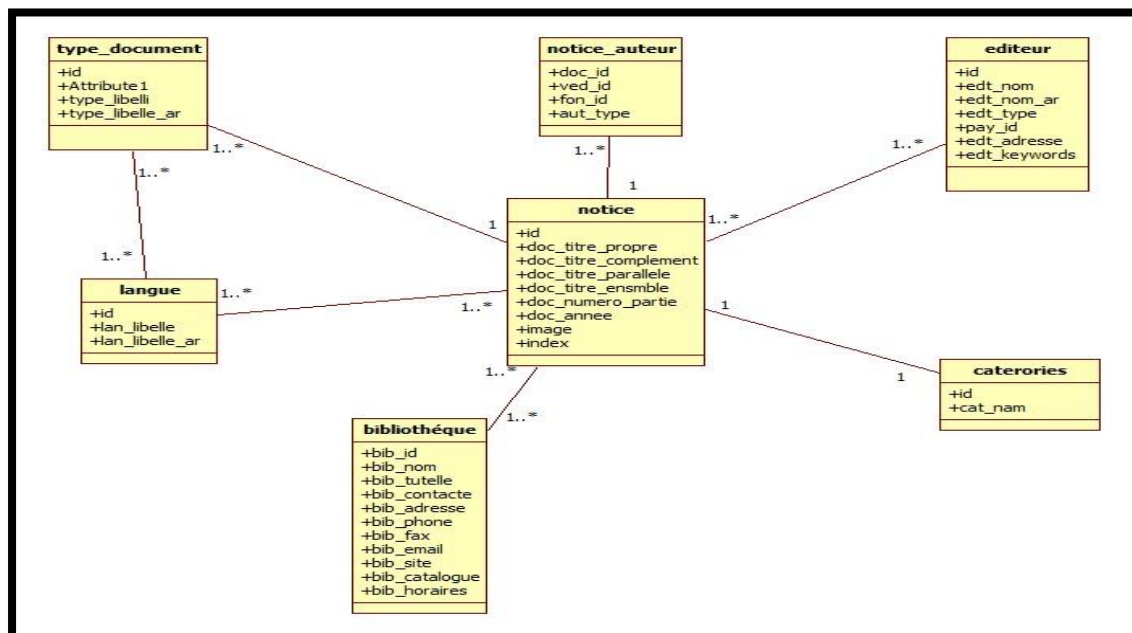


Fig 24 : digramme de classe.

8. Conclusion :

Tout au long de ce chapitre nous avons mené une conception détaillée du système d'information selon une approche objet afin de garantir la fiabilité et l'efficacité de la phase de réalisation de l'application.

Nous avons dressé une liste des acteurs constituant le système en exprimant leurs besoins avec le diagramme de cas d'utilisation, puis nous l'avons détaillé en précisant comment les objets et les acteurs doivent collaborer ensemble selon une dimension temporelle par l'utilisation des diagrammes de séquence. Finalement, nous avons décrit l'aspect statique avec les diagrammes des classes.

A l'aide de l'étude de notre cas nous avons déterminé l'environnement de développement de notre application qui sera présentée dans le chapitre suivant.

PARTIE2:LA REALISATION

1.Introduction :

Ce chapitre présente les différents outils et techniques informatique qui ont été utilisés pour la réalisation de notre composant Joomla. Le premier paragraphe est parler le langage de programmation que nous utilisé dans notre système avec les outils logiciel. En dernier lieu, on présente l'architecture de notre composant Joomla avec un aperçu sur les maquettes des interfaces, ainsi que les différentes fonctionnalités de composant Joomla illustrées par des images écrans.

2. Environnement de travail :

2.1. Wamp Server :

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale. [13]

✱ Apache :

Est un serveur http. C'est aussi un logiciel libre. Rappelons que le http (HyperText Transfer Protocol) est comme son nom l'indique un protocole de transmission dédié aux clients et serveurs du web. Un serveur le donc un logiciel permettant de rendre accessibles à de nombreux ordinateurs (les clients) de pages web stockées sur le disque.

✱ MySQL :

Une base de données (BDD) permet d'enregistrer, de stocker, de ranger des

Données de façon organisée et hiérarchisée. SQL est le langage qui permet de manipuler les bases de données. Les SGBD (Systèmes de Gestion de Base de Données) sont les programmes qui vous permettent de gérer vos données directement sans utiliser de script PHP.

Les plus connus sont :

MySQL : libre et gratuit, c'est probablement le SGBD le plus connu. Nous

L'utiliserons PostgreSQL : libre et gratuit comme MySQL, avec plus de fonctionnalités mais SQLite : libre et gratuit, très limité en fonctionnalités.

Oracle : utilisé par les entreprises, un des SGBD les plus complets mais payant.

Microsoft SQL Server : le SGBD de Microsoft.

* PHPmyadmin :

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données phpMyAdmin peut un serveur MySQL (cela nécessite un compte super-utilisateur) aussi bien qu'une seule base de données. Pour pouvoir administrer une seule base de données, il est nécessaire d'avoir un droit de lecture/ écriture sur la base de données concernée. Il vous appartient de consulter la section appropriée dans le manuel de MySQL.

IL permet de :

- Parcourir et supprimer base de données, tables, vues, champs et index.
- Créer, copier, supprimer, renommer et modifier bases de données, tables, vues, champs et index.
- Maintenir serveur, bases de données et tables en proposant une configuration serveur.
- Supprimer, modifier bases de données, tables en proposant une configuration serveur.
- Supprimer, modifier et ajouter des champs.
- Exécuter, modifier et créer un signet pour toute requête SQL, même les requêtes par lot. [14]

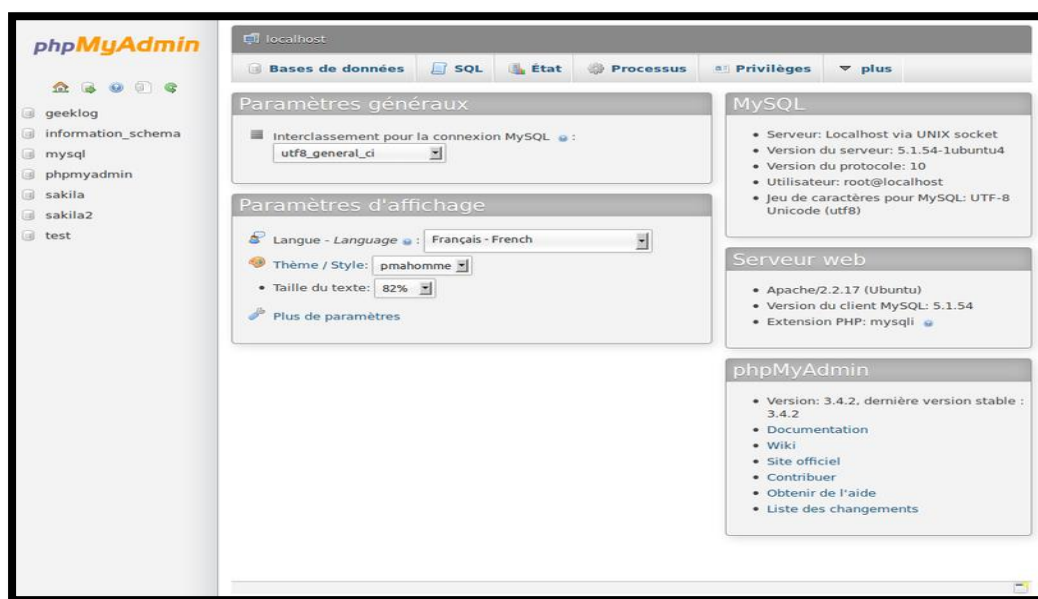


Fig25 : Interface phpmyadmin.

2.2. Langages utilisés :

On a bien utilisé deux différents langages :

Langage html et langage de programmation PHP

2.2.1 Langage HTML (HyperText Markup Language) ou XHTML :

Toute page web comprend une base de langage HTML ou XHTML. Il s'agit d'un langage de balisage qui définit essentiellement la **structure** de la page web (titres, tableaux, paragraphes, etc.).

C'est un langage qui permet de créer des hyperliens, à savoir des liens d'un document à un autre ou d'un endroit d'un document à un autre endroit du même document (identificateur de fragment).

Au départ ce langage ne définissait que la structure de la page mais très vite se sont rajoutées des balises (ou tags) qui s'occupaient de mettre en forme le document (mise en gras, en italique, choix d'une police de caractère, ...).

Depuis sa version 4, le langage HTML s'accompagne des CSS (Cascading Style Sheets - Feuilles de style en cascade) qui sont des styles qui se chargent de la **mise en** forme de la page. Le XHTML est une évolution du HTML basée sur XML.

2.2.2. PHP (PHP hyper text préprocesor) :

PHP (PHP : Hypertext Preprocessor) est un langage de script interprété côté serveur. Le code PHP est lu par le serveur et interprété pour produire une page HTML (ou d'autres types de fichier comme des images ou des documents PDF) à chaque fois que la page sera demandée. Ce langage a été créé initialement dans le cadre des logiciels (Open Source). De nombreux développeurs ont ensuite profité de l'Open Source pour développer de nombreux modules prêts à l'usage et paramétrables à volonté.

- Portabilité : Les modules serveurs de PHP sont implémentés sur les serveurs Apache et Unix mais PHP peut être implémenté sur d'autres plates-formes via des CGI. PHP travaille généralement avec une base de données MySQL.
- Mise en place : PHP est un langage de script, donc relativement facile à apprendre. La syntaxe est simplifiée, mais part d'une base de C.
- Performance : Très bonnes performances. La seule limite serait l'évolution rapide des modules et donc parfois le besoin de "rafraîchir" la programmation.
- Utilisation : Tout type d'application web peut être créé avec PHP.

2.2.3 XML (Extended Markup Language) :

Le langage de balisage XML peut également servir de base à une page web. Le fichier XML contiendra le contenu de la page web. Ce fichier sera lié à un fichier XSL (Feuilles de style XML) qui mettra en forme les données du fichier XML.

La plupart du temps, la page web ne sera pas basée sur un fichier XML natif mais celui-ci sera généré dynamiquement par un des langages de programmation web ci-dessous. [15]

3. Implémentation :

Nous allons présenter dans cette section le travail réalisé dans le cadre de ce projet par l'intermédiaire de captures d'écrans de quelques pages de notre composant Joomla.

3.1 La page Accueil :

Lors de lancement de notre composant, la Page d'accueil comme illustre la fig.



Fig 26 : l'accueil de site.

3.2. La Page Recherche :

Cette page est destinée pour tout adhérent (étudiant ou enseignant) voulant Recherche sur un ouvrage donné selon plusieurs critères comme illustré la fig.

Fig 27 : Recherche d'un livre.

3.3. La page d'Affichage des livres :

Notre page assure l'affichage de l'information détaillée sur chaque livre cette étape permet à l'utilisateur, de consulter les disponibles dans la bibliothèque.

Numéro	Année	Titre	Auteur	Année	Image de la couverture	Bibliothèque	Catégories	Type de document
2	2006	étude sur Caligula Albert Camus	Houel Florence	2006		El oued	English	Livre
6	2006	أفكار تطوير الإدارة والقيادة التي تربية في الذات العربية بالإضافة من التجارب والمناهج العلمية المتبعة	مؤتمن صمد الدين مني	2006		El oued		Livre
3	2006	الدليل المائي التاريخي ومسرح الخزيمية	علي شخوور، حسين	2006		El oued	علوم انسانية	Livre
4	1998	رأى في تاريخ العالم العربي والإسلامي الحديث والمعاصر	عثمان أبانعة ، فاروق	1998		El oued	علوم انسانية	Livre
5	2004	نشاط الطلبة الجزائريين إبان حرب التحرير 195	مائل، صابر	2004		El oued		Livre

Fig28 :l'affichage des livres.

3.4.La page d'Affichage de livre :

Notre page assure l'affichage d'information un livre

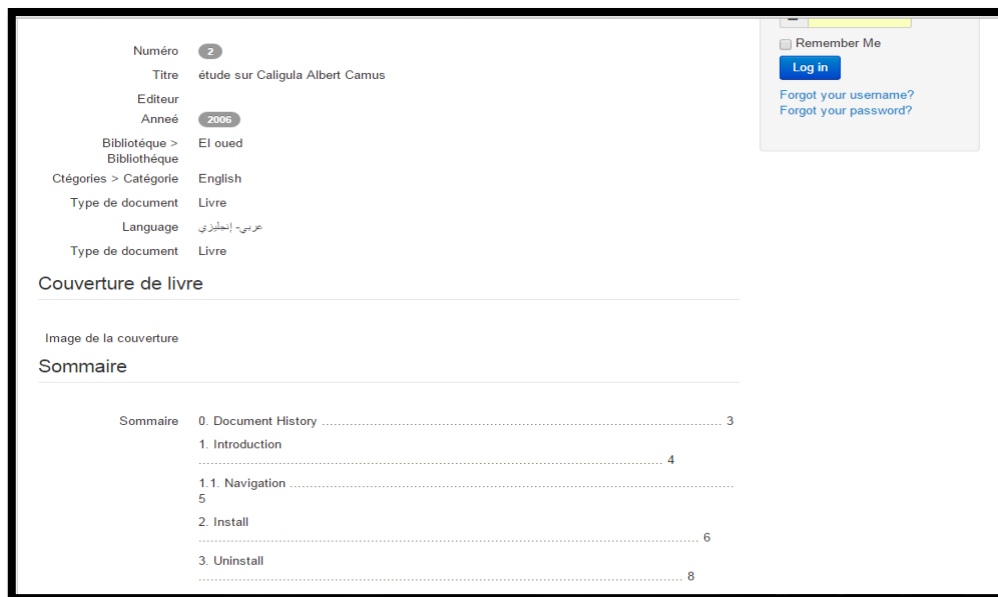


Fig 29 : couverture de livre.

4.Conclusion :

Nous avons présenté dans ce chapitre la réalisation de notre composant Joomla. Différentes fonctionnalités ont été développées afin de faciliter l'accès et l'exploitation des données relatives à la gestion catalogue de la bibliothèque.

Conclusion Générale :

L'objectif de notre projet de fin d'études est réaliser un composant Joomla pour la gestion d'un catalogue en ligne d'une bibliothèque universitaire de l'université d'EL-Oud .cet composant Joomla va permettre faciliter la recherche dans un catalogue universitaire en ligne.

Ce projet nous a été bénéfiques car il nous a permis de bain nous familiariser à programmation wampserver et les langages, PHP, HTML et XML.

Enfin, il nous a été très bénéfique car il nous a permis de concrétiser nos connaissances théoriques issues des trois années d'études, de connaitre de prés un système d'une bibliothèque universitaire et d'acquérir et pratiquer de nouvelles connaissances techniques, outils, programmation qui nous seront sans nul doute utiles dans notre vie professionnelle.

LISTE DES FIGURES

Nº	DÉSIGNATION	PAGE
Fig1	Choix de La langue	06
Fig2	Pré-installation	06
Fig3	acceptation de la licence	07
Fig4	base de données.	08
Fig5	Configuration FTP.	08
Fig6	Configuration.	09
Fig7	Terminer	10
Fig8	La gestion des Menus	10
Fig9	La gestion des contenus.	10
Fig10	La gestion des extensions	11
Fig11	Install, Components, Modules, Plug-ins, Langages, et Template	11
Fig12	Template Manager.	12
Fig13	Ajout d'un Plug-in	12
Fig14	la gestion des modules	13
Fig15	la gestion modules	13
Fig16	La gestion des composants.	14
Fig17	la gestion (MVC).	16
Fig18	Structure du composant(MVC)	16
Fig19	Composants (MVC)	17

Fig20	Diagramme de cas d'utilisation pour la Adhérent.	27
Fig21	Diagramme de cas d'utilisation "Recherche Simple "	27
Fig22	Digramme des cas d'utilisation "Recherche Avancée"	28
Fig23	Digamma de séquence pour Indentification	29
Fig24	digramme de classe	29
Fig25	Inteface phpMyAdmine	32
Fig26	l'accueil de site.	34
Fig27	Recherche d'un livre.	35
Fig28	l'affichage des livres	36
Fig29	couverture de livre.	36

LISTE DES TABLEAUX

Nº	DÉSIGNATION	PAGE
Tableau 1	Recherche	28

Bibliographie

[1] <http://creativecommons.org/licenses/by-sa/2.0/fr/deed.fr> Joomla pour les nuls [max_Hradware.com] (PDF)

Date de visite : 2014/12/08

[2] Internet dynamique avec Joomla(PDF)

Date de visite : 2014/05/08

[3] Rapport de stage(PDF)

Date de visite : 2015/05/08

[4] Accessible directement sur le site d'Amazon.fr TUTO_JOOMLA(PDF)

Date de visite : 2015/05/08

[5] Modele_Vue_Controleur(PDF)

Date de visite : 2015/04/25

[6]<file:///C:/Users/mc/Desktop/uml/Biblioth%C3%A8que%20universitaire%20%E2%80%94%20Wikip%C3%A9dia.html>

Date de visite : 2015/04/10

[7] Bibliothèque universitaires paris_ Académie

Date de visite : 2015/04/14

[8] http://fr.wikipedia.org/wink/Catalogue_de_biblioth%C3%A8que

Date de visite : 2015/04/12

[9] http://fr.wikipedia.org/wink/Unified_Modeling_Language

Date de visite : 2015/04/12

[10] <http://laurent-audibert.developpez.com/Cours-UML/html/Cours-UML008.html>

Date de visite : 2015/04/13

[11] http://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence

Date de visite : 2015/04/13

[12] <http://laurent-audibert.developpez.com/Cours-UML/html/Cours-UML015.html>

Date de visite : 2015/04/12

[13] tuto_MySQL_WampServer. (PDF)

Date de visite :2015/05/19

[14]<file:///C:/Users/mc/Desktop/2/phpmyadmin%20-%20Documentation%20Ubuntu%20Francophone.html>

Date de visite : 2015/05/15

[15]<file:///C:/Users/mc/Desktop/2/htiml ou xml %20-%20Documentation%20Ubuntu%20Francophone.html>

Date de visite : 2015/05/15