

Creative commons : Paternité - Pas d'Utilisation Commerciale -  
Pas de Modification 2.0 France (CC BY-NC-ND 2.0)



<http://creativecommons.org/licenses/by-nc-nd/2.0/fr>

UNIVERSITE CLAUDE BERNARD - LYON 1  
FACULTE DE PHARMACIE  
INSTITUT DES SCIENCES PHARMACEUTIQUES ET BIOLOGIQUES

2015

THESE n°75

**T H E S E**

pour le DIPLOME D'ETAT DE DOCTEUR EN PHARMACIE

Présentée et soutenue publiquement le 17 juillet 2015

par

M. CHENAL Pascal-andré

Né le 18 aout 1987

à Rennes (35)

\*\*\*\*\*

**CREATION D'UNE PLATEFORME WEB POUR LA FILIÈRE PHARMACIEN INGÉNIEUR**

\*\*\*\*\*

**JURY**

Mr Cyril PAILLER MATTEI, maitre de conférences  
Mme Stéphanie BRIANÇON, Professeure  
Mr Stéphane MICHEL, Développeur web  
Mr Rami MROUEH, Pharmacien industriel  
Mme Christine VINCIGUERRA, Professeure

## **UNIVERSITE CLAUDE BERNARD LYON 1**

- |   |                        |
|---|------------------------|
| • Président de l'Université                                       | M. François-Noël GILLY |
| • Vice-Président du Conseil d'Administration                      | M. Hamda BEN HADID     |
| • Vice-Président du Conseil Scientifique                          | M. Germain GILLET      |
| • Vice-Président du Conseil des Etudes et de la Vie Universitaire | M. Philippe LALLE      |

## **Composantes de l'Université Claude Bernard Lyon 1**

### **SANTE**

- |   |  |
|---|--|
| • UFR de Médecine Lyon Est  | Directeur : M. Jérôme ETIENNE          |
| • UFR de Médecine Lyon Sud Charles Mérieux                            | Directeur : Mme Carole BURILLON        |
| • Institut des Sciences Pharmaceutiques et Biologiques                | Directrice : Mme Christine VINCIGUERRA |
| • UFR d'Odontologie   | Directeur : M. Denis BOURGEOIS         |
| • Institut des Techniques de Réadaptation                             | Directeur : M. Yves MATILLON           |
| • Département de formation et centre de recherche en Biologie Humaine | Directeur : Anne-Marie SCHOTT          |

### **SCIENCES ET TECHNOLOGIES**

- |  |                                  |
|--|----------------------------------|
| • Faculté des Sciences et Technologies                                       | Directeur : M. Fabien DE MARCHI  |
| • UFR de Sciences et Techniques des Activités Physiques et Sportives (STAPS) | Directeur : M. Yannick VANPOULLE |
| • Ecole Polytechnique Universitaire de Lyon (ex ISTIL)                       | Directeur : M. Pascal FOURNIER   |
| • I.U.T. LYON 1  | Directeur : M. Christophe VITON  |
| • Institut des Sciences Financières et d'Assurance (ISFA)                    | Directeur : M. Nicolas LEBOISNE  |
| • ESPE   | Directeur : M. Alain MOUGNIOTTE  |

Décembre 2013

**UNIVERSITE CLAUDE BERNARD LYON 1**

**ISPB -Faculté de Pharmacie Lyon**

**Directrice : Madame la Professeure Christine VINCIGUERRA**

**Directeurs Adjoints : Madame S. BRIANCON, Monsieur P. LAWTON, Monsieur P. NEBOIS**

**Madame S. SENTIS, Monsieur M. TOD**

**Directrice Administrative : Madame P. GABRIELE**

**LISTE DES DEPARTEMENTS PEDAGOGIQUES**

**DEPARTEMENT PEDAGOGIQUE DE SCIENCES PHYSICO-CHIMIQUE  
ET PHARMACIE GALENIQUE**

• **CHIMIE ANALYTIQUE, GENERALE, PHYSIQUE ET MINERALE**

Monsieur Raphaël TERREUX (Pr – HDR)  
Monsieur Pierre TOULHOAT (Pr - PAST)  
Madame Julie-Anne CHEMELLE (MCU)  
Monsieur Lars-Petter JORDHEIM (MCU)  
Madame Christelle MACHON (AHU)

• **PHARMACIE GALENIQUE -COSMETOLOGIE**

Madame Stéphanie BRIANCON (Pr)  
Madame Françoise FALSON (Pr)  
Monsieur Hatem FESSI (Pr)  
Madame Joëlle BARDON (MCU - HDR)  
Madame Marie-Alexandrine BOLZINGER (MCU - HDR)  
Madame Sandrine BOURGEOIS (MCU)  
Madame Ghania HAMDI-DEGOBERT (MCU)  
Monsieur Plamen KIRILOV (MCU)  
Monsieur Fabrice PIROT (MCU - PH - HDR)  
Monsieur Patrice SEBERT (MCU - HDR)

• **BIOPHYSIQUE**

Monsieur Richard COHEN (PU – PH)  
Madame Laurence HEINRICH (MCU)  
Monsieur David KRYZA (MCU – PH)  
Madame Sophie LANCELOT (MCU - PH)  
Monsieur Cyril PAILLER-MATTEI (MCU)

**DEPARTEMENT PEDAGOGIQUE PHARMACEUTIQUE DE SANTE PUBLIQUE**

• **DROIT DE LA SANTE**

Monsieur François LOCHER (PU – PH)  
Madame Valérie SIRANYAN (MCU - HDR)

• **ECONOMIE DE LA SANTE**

Madame Nora FERDJAOUI MOUMJID (MCU - HDR)  
Monsieur Hans-Martin SPÄTH (MCU)  
Madame Carole SIANI (MCU – HDR)

• **INFORMATION ET DOCUMENTATION**

Monsieur Pascal BADOR (MCU - HDR)

• **HYGIENE, NUTRITION, HYDROLOGIE ET ENVIRONNEMENT**

Madame Joëlle GOUDABLE (PU – PH)



- **INGENIERIE APPLIQUEE A LA SANTE ET DISPOSITIFS MEDICAUX**

Monsieur Gilles AULAGNER (PU – PH)

Monsieur Daniel HARTMANN (Pr)

- **QUALITOLOGIE – MANAGEMENT DE LA QUALITE**

Madame Alexandra CLAYER-MONTEMBault (MCU)

Monsieur François COMET (MCU)

Monsieur Vincent GROS (MCU PAST)

Madame Pascale PREYNAT (MCU PAST)

- **MATHEMATIQUES – STATISTIQUES**

Madame Claire BARDEL-DANJEAN (MCU)

Madame Marie-Aimée DRONNE (MCU)

Madame Marie-Paule PAULTRE (MCU - HDR)

**DEPARTEMENT PEDAGOGIQUE SCIENCES DU MEDICAMENT**

- **CHIMIE ORGANIQUE**

Monsieur Pascal NEBOIS (Pr)

Madame Nadia WALCHSHOFER (Pr)

Monsieur Zouhair BOUAZIZ (MCU - HDR)

Madame Christelle MARMINON (MCU)

Madame Sylvie RADIX (MCU -HDR)

Monsieur Luc ROCHEBLAVE (MCU - HDR)

- **CHIMIE THERAPEUTIQUE**

Monsieur Roland BARRET (Pr)

Monsieur Marc LEBORGNE (Pr)

Monsieur Laurent ETTOUATI (MCU - HDR)

Monsieur Thierry LOMBERGET (MCU - HDR)

Madame Marie-Emmanuelle MILLION (MCU)

- **BOTANIQUE ET PHARMACOGNOSIE**

Madame Marie-Geneviève DIJOUX-FRANCA (Pr)

Madame Isabelle KERZAON (MCU)

Monsieur Serge MICHALET (MCU)

- **PHARMACIE CLINIQUE, PHARMACOCINETIQUE ET EVALUATION DU MEDICAMENT**

Madame Roselyne BOULIEU (PU – PH)

Madame Magali BOLON-LARGER (MCU - PH)

Madame Céline PRUNET-SPANO (MCU)

Madame Catherine RIOUFOL (MCU - PH)

**DEPARTEMENT PEDAGOGIQUE DE PHARMACOLOGIE, PHYSIOLOGIE ET TOXICOLOGIE**

- **TOXICOLOGIE**

Monsieur Jérôme GUITTON (PU – PH)

Monsieur Bruno FOUILLET (MCU)

Monsieur Sylvain GOUTELLE (MCU-PH)

Madame Léa PAYEN (MCU -HDR)

- **PHYSIOLOGIE**

Monsieur Christian BARRES (Pr)

Monsieur Daniel BENZONI (Pr)

Madame Kiao Ling LIU (MCU)

Monsieur Ming LO (MCU - HDR)

- **PHARMACOLOGIE**

Monsieur Bernard RENAUD (Pr)  
 Monsieur Michel TOD (PU – PH)  
 Monsieur Luc ZIMMER (PU – PH)  
 Madame Bernadette ASTIER (MCU - HDR)  
 Monsieur Roger BESANCON (MCU)  
 Madame Evelyne CHANUT (MCU)  
 Monsieur Nicola KUCZEWSKI (MCU)

Monsieur Olivier CATALA (Pr PAST)  
 Monsieur Pascal THOLLOT (MCU PAST)  
 Madame Corinne FEUTRIER (MCU-PAST)

## **DEPARTEMENT PEDAGOGIQUE DES SCIENCES BIOMEDICALES A**

- **IMMUNOLOGIE**

Monsieur Jacques BIENVENU (PU – PH)  
 Monsieur Guillaume MONNERET (PU-PH)  
 Madame Cécile BALTER-VEYSSEYRE (MCU - HDR)  
 Monsieur Sébastien VIEL (AHU)

- **HEMATOLOGIE ET CYTOLOGIE**

Madame Christine TROUILLOT-VINCIGUERRA (PU - PH)  
 Madame Brigitte DURAND (MCU - PH)  
 Monsieur Olivier ROUALDES (AHU)

- **MICROBIOLOGIE ET MYCOLOGIE FONDAMENTALE ET APPLIQUEE AUX BIOTECHNOLOGIE INDUSTRIELLES**

Monsieur Patrick BOIRON (Pr)  
 Monsieur Jean FRENEY (PU – PH)  
 Madame Florence MORFIN (PU – PH)  
 Monsieur Didier BLAHA (MCU)  
 Madame Anne DOLEANS JORDHEIM (MCU)  
 Madame Emilie FROBERT (MCU - PH)  
 Madame Véronica RODRIGUEZ-NAVA (MCU)  
 Madame Ghislaine DESCOURS (AHU)

- **PARASITOLOGIE, MYCOLOGIE MEDICALE**

Monsieur Philippe LAWTON (Pr - HDR)  
 Madame Nathalie ALLIOLI (MCU)  
 Madame Samira AZZOUZ-MAACHE (MCU - HDR)

## **DEPARTEMENT PEDAGOGIQUE DES SCIENCES BIOMEDICALES B**

- **BIOCHIMIE – BIOLOGIE MOLECULAIRE - BIOTECHNOLOGIE**

Madame Pascale COHEN (Pr)  
 Monsieur Alain PUISIEUX (PU - PH)  
 Monsieur Karim CHIKH (MCU - PH)  
 Madame Carole FERRARO-PEYRET (MCU - PH)  
 Madame Caroline MOYRET-LALLE (MCU – HDR)  
 Madame Angélique MULARONI (MCU)  
 Madame Stéphanie SENTIS (MCU)  
 Monsieur Olivier MEURETTE (MCU)  
 Monsieur Benoit DUMONT (AHU)

- **BIOLOGIE CELLULAIRE**  
Madame Bénédicte COUPAT-GOUTALAND (MCU)  
Monsieur Michel PELANDAKIS (MCU - HDR)
- **INSTITUT DE PHARMACIE INDUSTRIELLE DE LYON**  
Monsieur Philippe LAWTON (Pr - HDR)  
Madame Angélique MULARONI (MCU)  
Monsieur Patrice SEBERT (MCU – HDR)  
Madame Valérie VOIRON (MCU - PAST)
- **Assistants hospitalo-universitaires sur plusieurs départements pédagogiques**  
Madame Emilie BLOND  
Madame Christelle MOUCHOUX  
Madame Florence RANCHON
- **Attachés Temporaires d'Enseignement et de Recherche (ATER)**  
Monsieur Eyad AL MOUAZEN 85<sup>ème</sup> section  
Monsieur Boyan GRIGOROV 87<sup>ème</sup> section  
Madame Mylène HONORAT 85<sup>ème</sup> section  
Monsieur Abdalah LAOUINI 85<sup>ème</sup> section  
Madame Marine CROZE 86<sup>ème</sup> section

**Pr : Professeur**

**PU-PH : Professeur des Universités, Praticien Hospitalier**

**MCU : Maître de Conférences des Universités**

**MCU-PH : Maître de Conférences des Universités, Praticien Hospitalier**

**HDR : Habilitation à Diriger des Recherches**

**AHU : Assistant Hospitalier Universitaire**

**PAST : Personnel Associé Temps Partiel**

## TABLE DES MATIÈRES

<b>1. INTRODUCTION .....</b>	<b>11</b>
<b>2. ETAT DES LIEUX DE LA SITUATION .....</b>	<b>11</b>
<b>3. OBJECTIFS.....</b>	<b>12</b>
<b>4. CAHIER DES CHARGES .....</b>	<b>13</b>
<b>5. REFLEXIONS SUR LA MISE EN PLACE DES DIFFERENTS SERVICES .....</b>	<b>14</b>
5.1. CONDITIONS D'ACCES ET CONFIDENTIALITE .....	14
5.2. LA VERIFICATION DES DONNEES.....	15
5.3. LES STATISTIQUES.....	16
5.4. L'INTEGRATION DES RESEAUX SOCIAUX .....	17
5.5. LA GESTION DES COURS EN LIGNE .....	18
5.6. LES MEDIAS .....	18
<b>6. IDENTIFICATION DES PARTENAIRES POTENTIELS .....</b>	<b>18</b>
6.1. L'ICAP .....	18
6.2. L'ASSOCIATION ANCEPHAL.....	19
<b>7. INTERFACE ET DESIGN .....</b>	<b>20</b>
7.1. OBJECTIFS.....	20
7.2. TECHNOLOGIES UTILISEES.....	21
7.3. HTML .....	22
7.4. CSS.....	23
7.5. LA PROGRAMMATION .....	25
<b>8. PROGRAMMATION : ASPECTS TECHNOLOGIQUES.....</b>	<b>25</b>
8.1. CHOIX DES LANGAGES DE PROGRAMMATION .....	25
8.2. FONCTIONNEMENT DU BINOME PHP/AJAX .....	27
8.3. CMS VS NOUVEAU DEVELOPPEMENT PHP.....	28
8.4. CHOIX DE L'ENVIRONNEMENT DE PROGRAMMATION INTEGRE (IDE).....	29
8.5. L'ENVIRONNEMENT D'EXECUTION .....	30
<b>9. CONCEPTS DE PROGRAMMATION .....</b>	<b>32</b>
9.1. LES CLASSES .....	32
9.2. ECRITURE D'UNE METHODE .....	34
9.3. CONSTRUCTEUR DE CLASSE.....	34
9.4. PORTEE DES VARIABLES ET DES METHODES .....	35
9.5. LE PATTERN SINGLETON .....	35
9.6. L'HERITAGE.....	36
9.7. ARCHITECTURE ET FORMAT UML .....	37
9.8. LES VARIABLES SUPERGLOBALES .....	38
9.8.1. <i>Les paramètres de requête.....</i>	<i>38</i>
9.8.2. <i>Les sessions utilisateur .....</i>	<i>39</i>
<b>10. DEVELOPPEMENT PHP .....</b>	<b>39</b>
10.1. RESSOURCES PHP .....	39
10.2. ARCHITECTURE LOGICIELLE .....	40
10.2.1. <i>Généralités.....</i>	<i>40</i>
10.2.2. <i>Index et contrôleurs .....</i>	<i>41</i>
10.2.3. <i>Le contrôleur générique.....</i>	<i>42</i>
10.2.4. <i>Architecture des modèles.....</i>	<i>43</i>

10.2.5.	<i>Le contrôleur spécifique</i> .....	44
10.2.6.	<i>La vue</i> .....	45
10.2.7.	<i>Utilisation du JSON</i> .....	46
10.3.	PROCEDURES DE MODIFICATION DU PROGRAMME PHP .....	47
10.3.1.	<i>Modification de l'arborescence du site</i> .....	47
10.3.2.	<i>Ajout d'une nouvelle école</i> .....	48
10.3.3.	<i>Modification du choix des listes déroulantes</i> .....	48
10.3.4.	<i>Ajout d'un nouvel objet PHP rattaché à la base de données</i> .....	48
<b>11.</b>	<b>PROGRAMMATION JAVASCRIPT</b> .....	<b>49</b>
11.1.	AVANT PROPOS .....	49
11.1.1.	<i>Capacités de JavaScript</i> .....	49
11.1.2.	<i>Choix du framework principal</i> .....	49
11.1.3.	<i>Différences d'interprétation entre navigateurs</i> .....	50
11.1.4.	<i>Liste des ressources utilisées</i> .....	51
11.1.5.	<i>Connexion asymétrique et programmation événementielle</i> .....	52
11.1.6.	<i>Inclusion du JavaScript</i> .....	53
11.2.	ARCHITECTURE DES COMPOSANTES GRAPHIQUES DU SITE WEB .....	54
11.2.1.	<i>Organisation générale</i> .....	54
11.2.2.	<i>Le menu principal</i> .....	55
11.2.3.	<i>Les tableaux dans l'interface de gestion</i> .....	57
<b>12.</b>	<b>L'ARBORESCENCE DU PROJET</b> .....	<b>59</b>
<b>13.</b>	<b>MISE EN SERVICE DE LA PLATEFORME</b> .....	<b>60</b>
13.1.	CONTRAINTES TECHNOLOGIQUES.....	60
13.2.	INSTALLATION DE LA PLATEFORME SUR LE SERVEUR .....	61
13.2.1.	<i>Copie des fichiers du programme</i> .....	61
13.2.2.	<i>Enregistrement des paramètres du serveur</i> .....	62
13.2.3.	<i>Installation de la base de données</i> .....	63
<b>14.</b>	<b>BILAN DU DEVELOPPEMENT</b> .....	<b>65</b>
<b>16.</b>	<b>BIBLIOGRAPHIE</b> .....	<b>67</b>

## TABLE DES FIGURES

Figure 1 : Ancien site de la filière .....	12
Figure 2: méthode de visualisation de la localisation professionnelle des anciens étudiants.....	17
Figure 3 : interface utilisateur du site .....	21
Figure 4 : fonctionnement du binôme PHP/AJAX.....	27
Figure 5 : L'environnement NetBeans.....	30
Figure 6 : interface de gestion de MAMP .....	32
Figure 7 : exemple de schéma UML .....	37
Figure 8 : Pattern MVC .....	40
Figure 9 : Architecture MVC du code PHP.....	42
Figure 10 : Schéma UML de la gestion d'objets SQL.....	44
Figure 11 : Transfert de données au format JSON .....	46
Figure 12 : parts de marché des différents navigateurs en novembre 2012 .....	51
Figure 13 : Organisation générale du site web .....	54
Figure 14 : Insertion de l'interface d'administration.....	55
Figure 15: Schéma UML du widget de connexion.....	56
Figure 16 : tableau JavaScript .....	57
Figure 17 : schéma UML de la classe JsonTable .....	58
Figure 18 : écran de configuration de Filezilla.....	62
Figure 19 : Exportation de la base de données .....	64
Figure 20 : écran de requête de PHPmyadmin .....	65

## TABLE DES TABLEAUX

Table 1 : Droits d'accès au site .....	14
Table 2 : Exemple de transaction JavaScript.....	53
Table 3 : détail des statistiques du projet.....	66
Table 4 : résumé de la taille du projet .....	66

## TABLE DES ABBRÉVIATIONS

<b>AJAX</b>	(Asynchronous JavaScript and XML) outil de communication et de mise en forme JavaScript
<b>ANCEPHAL</b>	association des anciens de la faculté de pharmacie de Lyon
<b>CSS</b>	(Cascading stylesheet) : Outil de mise en forme des balises XHTML
<b>CMS</b>	(Content Management System) logiciel de création de sites web
<b>DOM</b>	(Document object model): Représentation permettant à un langage tiers d'interagir avec les balises HTML du document.
<b>FTP</b>	(File Transfer Protocol) Protocole de transfert de fichier
<b>GUI</b>	(Graphical user interface) : Interface utilisateur graphique
<b>ICAP</b>	Innovation, conception et accompagnement pour la pédagogie
<b>IDE</b>	(Integrated Development Environment) Outil de programmation
<b>JIT</b>	(just in time) : Outil de compilation de code à la volée
<b>JSON</b>	(JavaScript Object Notation) Format de données
<b>MAMP</b>	(Macintosh Apache MySQL PHP) Environnement de développement
<b>MVC</b>	(modèle Vue contrôleur) Architecture de programmation
<b>PHP</b>	(Hypertext Preprocessor) : Langage de programmation serveur
<b>PDO</b>	(PHP Data Objects) Classe PHP d'interface avec la base de donnée
<b>POO</b>	(Programmation orientée objet) Méthode de programmation
<b>SFTP</b>	Secured File Transfer Protocol
<b>SGBD</b>	Système de Gestion de Bases de Données
<b>SPIRAL</b>	Plateforme pédagogique de l'université de Lyon 1
<b>SQL</b>	(Structured Query Language): Langage de requête pour bases de données
<b>TICE</b>	Technologies de l'Information et de la Communication pour l'Enseignement
<b>UML</b>	(Unified Modeling Language) Modélisation graphique utilisée en développement logiciel.
<b>XHTML</b>	(Extensible HyperText Markup Language) : Langage balisé de présentation de données sur internet
<b>XML</b>	(Extensible Markup Language) Format de données balisé
<b>W3C</b>	(World Wide Web Consortium) Organisme de normalisation du web

## 1. Introduction

La filière pharmacien-ingénieur de l'ISPB offre la possibilité aux étudiants de suivre une formation en école d'ingénieur à la fin de leur cursus de pharmacie (en 5<sup>e</sup> et 6<sup>e</sup> année). Ces 6 ans d'études conduisent à l'obtention du diplôme de pharmacien et du titre d'ingénieur. Les étudiants en pharmacie suivent au cours de leur cursus 3 UE de 100 h chacune afin de les préparer à leur entrée en école.

À l'heure actuelle, des partenariats sont disponibles avec les écoles suivantes :

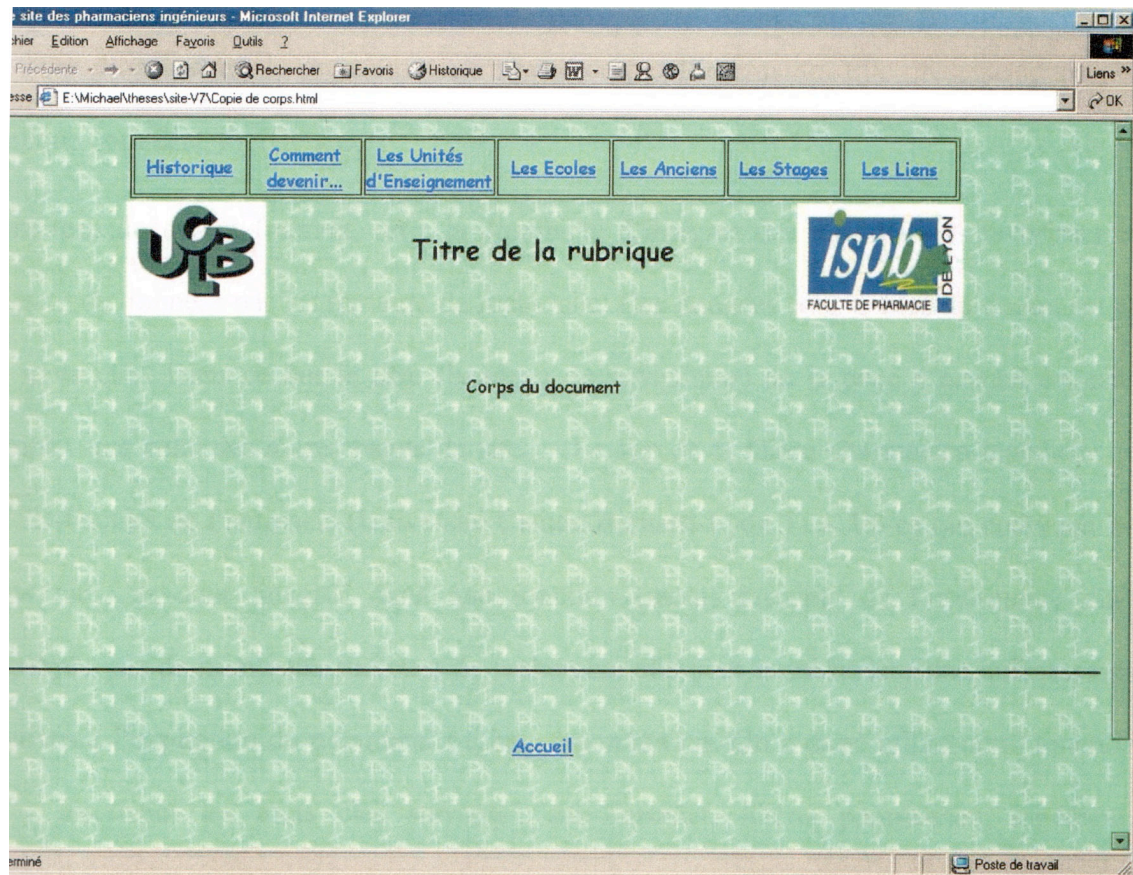
- L'école polytechnique de Montréal (EPM)
- L'école nationale des mines de Saint-Etienne (ENSMSE)
- L'institut national supérieur d'agriculture (INSA) Rhône-Alpes
- L'école supérieure de chimie physique électronique de Lyon (CPE)
- L'institut national polytechnique de Grenoble – Génie industriel (INP)
- Polytech Lyon

Afin de promouvoir cette filière auprès des étudiants et des professionnels, il était devenu indispensable de lui fournir une visibilité sur internet. Le but de ce projet est double ; fédérer une communauté d'anciens et diffuser de l'information sur la formation et ses débouchés.

## 2. Etat des lieux de la situation

Ce travail fait suite à celui réalisé par Michael Rabut (1), qui avait mis en place un premier site en 2003. Celui-ci avait essentiellement un but informatif. Il contenait aussi un annuaire des étudiants de 1996 à 2001. Voici une capture d'écran de ce site.





**Figure 1 : Ancien site de la filière**

Son inconvénient majeur est qu'il est totalement statique. La mise à jour de celui-ci doit être faite en éditant le code HTML. Depuis sa mise en service en 2003, aucune mise à jour n'a été effectuée.

L'objet de ce travail est donc de remplacer ce site afin de fournir un outil à jour et pouvant être facilement maintenu dans les années à venir. Un changement de la charte graphique a aussi été effectué par la même occasion.

### 3. Objectifs

Tout d'abord, la plateforme à venir doit permettre de connaître précisément les perspectives d'emploi qu'offre la formation. Une meilleure information permettrait d'augmenter l'attractivité de la filière et une meilleure orientation des étudiants. Ces derniers doivent faire un choix entre six écoles d'ingénieur pour le dernier cycle de leurs

études. Les enseignements dispensés ainsi que les postes occupés en sortie sont très différents d'une école à l'autre. Un choix avisé facilitera vraisemblablement l'insertion des étudiants dans le domaine de leur choix.

Les étudiants pharmacien-ingénieurs doivent réaliser un stage de fin d'études de six mois. C'est souvent à la suite de ce stage qu'ils obtiennent leur premier emploi, ou à défaut une première expérience significative dans leur domaine d'expertise. Cette période est donc cruciale pour l'insertion professionnelle. Une grande partie des offres circulent presque exclusivement par les réseaux présents dans les écoles. La mise en place de partenariats avec l'industrie est donc un facteur déterminant pour l'insertion. La plateforme réalisée dans ce projet doit être un support pour publier les offres de stage et d'emploi adressées aux membres de la filière.

La création d'un réseau professionnel est un point important au cours d'une carrière. Les anciens actuellement en poste sont vraisemblablement les meilleurs ambassadeurs pour la filière. Pouvoir contacter directement ces personnes peut permettre de se tenir informé des opportunités dans une entreprise. En outre, une candidature sera plus efficace si elle est relayée par une personne en interne. Maintenir le contact entre les anciens étudiants est donc indispensable. Le site internet aura donc aussi pour but de fournir un annuaire à tous les membres inscrits.

## **4. Cahier des charges**

Afin de remplir les objectifs précédemment énoncés, il a été décidé de fournir les services suivants :

- Maintenir une base de données à jour des anciens de la filière
- Permettre l'échange des supports de cours entre les enseignants et les étudiants
- Proposer une base de données d'offres d'emploi et de stages
- Générer des données chiffrées sur les débouchés de la filière
- Assurer une information à jour pour les professionnels et les étudiants

La partie suivante décrit de manière plus précise la mise en place de ces services.

## 5. Réflexions sur la mise en place des différents services

### 5.1. Conditions d'accès et confidentialité

Afin de fournir un service personnalisé et de créer de l'information sur la filière, il est nécessaire de stocker des données personnelles. Il va de soi que l'accès à ces données doit être restreint et que les utilisateurs doivent rester propriétaires des informations qu'ils soumettent. De plus, ils doivent pouvoir rétracter leur abonnement au site à tout moment.

Pour répondre à ces conditions, il a été décidé de créer plusieurs niveaux d'accès. Les simples visiteurs ne pourront voir que des données statistiques anonymes. En revanche, les utilisateurs du service pourront obtenir des informations sur toutes les personnes inscrites afin de pouvoir répondre au besoin de connexion entre les anciens étudiants. Pour des raisons de sécurité, en revanche, un compte ne peut être édité que par un administrateur ou le propriétaire dû dit compte. Voici un résumé des droits en fonction des profils.

**Table 1 : Droits d'accès au site**

		Administrateur	Professeur	Ancien	étudiant	visiteur
Comptes utilisateurs	voir	✓	✓	✓	✓	
	Editer	✓				
Offres d'emploi	voir	✓	✓	✓	✓	
	Editer	✓				
cours	voir	✓	✓		✓	
	Editer	✓	✓			
médias	voir	✓	✓	✓	✓	✓
	Editer	✓				
Statistiques	voir	✓	✓	✓	✓	✓
	Editer					

L'autre condition précédemment évoquée est l'accord des adhérents. Un compte contenant des informations ne peut être créé par un administrateur sans l'accord du principal concerné. Ainsi, un système d'invitations a été mis en place. L'administrateur entre l'adresse mail de la personne qu'il veut inviter. Le serveur génère alors un

identifiant et un mot de passe dans la base de données. Un mail d'invitation est envoyé au propriétaire de l'adresse comportant un lien d'activation du compte. L'invité est alors sollicité pour remplir un formulaire contenant ses informations personnelles. L'utilisateur fait lui-même la démarche de donner ses informations.

La désinscription peut se faire sur la page de gestion du compte utilisateur. Une nouvelle invitation devra être demandée aux administrateurs pour se réinscrire.

## **5.2. La vérification des données**

La vérification des données est une étape importante de la conception d'un site web dynamique; c'est peut-être celle qui demande le plus temps en terme de programmation. Chaque formulaire doit être vérifié pour que l'utilisateur soit dans l'incapacité d'insérer une erreur. Si une telle chose arrive, le programme peut rendre des résultats erronés (dans le cas des statistiques par exemple) et bien tout simplement ne plus fonctionner.

*Exemple :*

L'utilisateur rentre du texte à la place d'un nombre pour le temps d'inactivité à la fin de ses études. Pour des fins de statistiques, la moyenne de ce champ est réalisée pour tous les utilisateurs. Le programme peut alors essayer d'additionner des nombres avec du texte et renvoyer une erreur. La page ne peut alors plus s'afficher correctement.

La solution la plus simple est alors de restreindre le choix de réponse à quelques possibilités. Ceci se fait grâce à des listes déroulantes ou des cases à cocher. En revanche, le choix de certains champs ne peut être réduit de la sorte sans une perte substantielle d'information.

Par exemple, sélectionner la localisation du lieu de travail. Il est possible de proposer tous les pays, mais il sera difficile d'envisager toutes les adresses. On peut alors demander à un serveur externe de vérifier l'adresse. C'est ce qui est fait lors de l'affichage de la carte des anciens.

Les principaux critères d'acceptation pour un formulaire sont les suivants :

- un format cohérent (nombre, texte, date...etc.)
- le bon sens (date improbable, adresse mail sans nom de domaine...etc.)
- la nécessité ou non de l'information (ne pas créer de compte sans adresse mail par exemple)

L'ordinateur peut comparer différents formats de données, ou bien l'absence de données. En revanche, seul l'humain peut dicter ce qui a du sens ou non.

### **5.3. Les statistiques**

Un des points présents dans le cahier des charges est la génération de statistiques sur la filière. Deux choix sont possibles afin de les générer. La première option est de réaliser ces statistiques grâce aux données récoltées et de régulièrement les mettre à jour sur le site web. La seconde est de générer automatiquement les statistiques grâce aux données enregistrées sur le serveur. C'est cette dernière option qui a été privilégiée.

Le principal obstacle à l'obtention de statistiques fiables est ici le manque de données. Aucune trace des anciens élèves n'a été conservée, en dehors de leur adresse mail. Compte tenu du nombre de personnes concernées, il serait difficile de toutes les contacter afin de recueillir leur cheminement professionnel. C'est donc le serveur qui se chargera à la place des responsables de filière de contacter les membres et de leur faire remplir un formulaire. Le traitement des données sera de plus effectué en temps réel (en fonction des inscriptions).

Les informations statistiques qui ont été sélectionnées sont les suivantes :

- la répartition des étudiants dans les différentes écoles
- le nombre d'étudiants par années
- l'attente entre la fin des études et le premier emploi
- la localisation géographique de l'emploi
- le domaine professionnel (production, finance, marketing...)
- Le secteur d'activité (environnement, cosmétique...)

Les deux derniers points posent un problème pour le traitement automatique des données. La question étant ouverte, il a fallu restreindre le choix à une dizaine de possibilités.

La visualisation de la localisation géographique de l'emploi se fait par une carte. L'utilisateur rentre une adresse et un point apparaît à ces coordonnées. Cette partie a été réalisée grâce au service de localisation de Google. L'adresse est envoyée au service de localisation qui renvoie les coordonnées géographiques les plus probables. Ces coordonnées pourront ensuite être visualisées dans une fenêtre Google Maps.



**Figure 2: méthode de visualisation de la localisation professionnelle des anciens étudiants**

#### **5.4. L'intégration des réseaux sociaux**

Les réseaux sociaux sont à l'heure actuelle un élément incontournable d'internet. Facebook a franchi le milliard d'utilisateurs en 2012, LinkedIn® ou Viadeo® sont de plus en plus utilisés dans les processus de recrutement. Pour générer des informations pertinentes, il faut que les utilisateurs mettent à jour régulièrement leurs profils. Cela ne sera vraisemblablement pas le cas sur la plateforme pharmacien-ingénieur. Cette dernière permettra essentiellement d'obtenir des informations à la sortie d'école. Les réseaux sociaux se chargeront d'effectuer la mise à jour par la suite.

Il est actuellement impossible pour des raisons évidentes de confidentialité de mettre à jour le site web par le biais de ces réseaux. Par contre, il est possible d'inclure un lien vers les pages de chaque utilisateur.

### **5.5. La gestion des cours en ligne**

L'université dispose de la plateforme SPIRAL. Celle-ci offre déjà une possibilité de stocker des ressources pédagogiques en ligne. Il a cependant été décidé de mettre en place ce service. La principale raison est d'induire des inscriptions le plus tôt possible. Les étudiants doivent s'inscrire sur le site de la filière pour bénéficier de ce service. Ils pourront ainsi se familiariser plus rapidement avec la plateforme.

La mise en place du service nécessite de réguler l'accès au cours pour des raisons de propriété intellectuelle. Ainsi, chaque enseignant peut modifier uniquement les cours dont il est propriétaire. Les étudiants n'ont accès qu'aux cours de leur promotion. Les administrateurs peuvent éditer et voir les cours pour corriger d'éventuels problèmes sur le site web.

### **5.6. Les médias**

Un système d'upload de médias a été mis en place sur le site. Il permet aux administrateurs de mettre à disposition des visiteurs du contenu vidéo, audio ou photographique en rapport avec les différentes écoles proposées par la filière. Le but est de pouvoir enrichir l'information au cours du temps. Le contenu du site ne sera pas figé à sa création, mais pourra évoluer sans l'intervention d'un développeur.

## **6. Identification des partenaires potentiels**

### **6.1. L'ICAP**

La première intention a été de se tourner vers la faculté de pharmacie de Lyon afin d'héberger la plateforme. Le service en charge des TICE (technologies de l'information et de la communication pour l'enseignement) à la faculté est l'ICAP (innovation conception et accompagnement pour la pédagogie). Ce service assure notamment le

développement de la plateforme SPIRAL à l'université. Cette plateforme dispose de nombreuses fonctionnalités comme l'échange de cours, la diffusion d'enquêtes...

La dernière version, SPIRAL connect, fonctionne grâce à la technologie Adobe Flex®, une plateforme permettant de créer des programmes Flash®. Cette technologie ne permet pas d'intégrer un nouveau site sans avoir accès au programme initial. Une version plus ancienne du site (SPIRAL classique) fonctionne encore grâce à la technologie HTML. Une discussion a eu lieu avec les responsables de cette plateforme afin de créer un site intégré à SPIRAL classique. Ce logiciel ne permettait cependant pas de réaliser le cahier des charges précédemment énoncé. Le site a donc dû être développé de manière indépendante. La faculté ne pouvant pas héberger un programme tiers pour des raisons de sécurité, il a été décidé de se tourner vers l'association Ancephal afin d'obtenir un hébergement.

## **6.2. L'association Ancephal**

Ancephal (l'association des anciens de la faculté de pharmacie de Lyon) est une association qui a pour but de développer le réseau d'anciens pour toutes les filières de la faculté. Cet organisme organise des événements et conserve une base de données des anciens étudiants afin d'assurer leur communication. Les objectifs de cette thèse étant totalement en accord avec la vocation d'Ancephal, un partenariat s'est naturellement créé.

Un site internet a déjà été développé pour l'association. L'hébergement est réalisé par la société OVH sur un serveur mutualisé (plusieurs sites regroupés sur un même serveur). L'adresse du domaine est « [ancephal.fr](http://ancephal.fr) ». Ce site regroupe des informations relatives à l'association et assure la promotion des événements à venir. Il a été décidé de créer un sous-domaine (« [ingenieur.ancephal.fr](http://ingenieur.ancephal.fr) ») pour la filière. Il sera possible de créer un site pour chaque filière. Une base de données spécifique a aussi été créée pour stocker les données utilisateurs.



## **7. Interface et design**

### **7.1. Objectifs**

L'exercice est de réaliser une interface esthétique et simple d'utilisation. Pour ce faire, il a été décidé de dépouiller au maximum l'interface afin que l'utilisateur ne se concentre que sur l'essentiel.

Le menu principal qui a été développé en est un parfait exemple. Il fonctionne comme une arborescence. Cependant, en fonction du statut de l'utilisateur connecté (étudiant, administrateur ou simple visiteur), l'arborescence va changer pour ne montrer que ce qui lui est utile. Un simple visiteur n'ayant pas accès à l'échange de cours en ligne, il est inutile de lui montrer un lien qui le conduirait vers une page où l'accès lui sera refusé.

La bannière du site se veut peu intrusive pour libérer un maximum d'espace au contenu. Le site favorise au maximum l'utilisation de représentations graphiques pour présenter le contenu (images, graphiques...) afin de mieux capter l'attention de l'utilisateur.

Voici un aperçu du design global du site :



Figure 3 : interface utilisateur du site

## 7.2. Technologies utilisées

Le contenu ainsi que sa mise en page primaire est transporté grâce à XHTML (eXtensible HyperText Markup Language). Ce langage dérive directement du XML qui est un langage utilisant des balises. Il permet de définir des paragraphes, d'introduire des images, des sons ou des vidéos. Le site web est basé sur la version 4 de ce langage, pour des raisons de compatibilité. La version 5, bien que déjà implémentée dans les navigateurs récents, n'est à l'heure actuelle pas dans sa version définitive. Qui plus est, il est important de garder un site compatible avec les anciens navigateurs. La tendance actuelle est de passer vers cette technologie, notamment grâce à la mise à jour automatique des navigateurs. D'autant plus que HTML 5 permet de se passer de la technologie Flash® pour l'affichage de vidéos et de pistes audio. Dans ce projet, Flash® sera encore utilisé pour ces indications.

La réelle mise en page est apportée par CSS (cascading style sheets). La version utilisée sur le site est CSS 3, la dernière mise à jour disponible. Contrairement à la version précédente, celle-ci implémente les couches alpha<sup>1</sup> abondamment utilisées sur le site. Théoriquement, cette version est apportée par HTML 5. Cependant, le défaut de compatibilité n'est pas problématique. Les instructions non comprises par le navigateur seront simplement ignorées. De plus, il existe quelques possibilités pour contourner les incompatibilités grâce à des instructions propriétaires (instructions spécifiques à un navigateur, ne faisant pas officiellement partie du langage CSS). Le design global du site sera éventuellement altéré, mais le contenu sera identique.

### **7.3. HTML**

Il sera discuté dans cette partie de la structure du langage HTML. Comme énoncé précédemment, il s'agit d'un langage de balisage. La structure d'une balise est la suivante :

#### **HTML**

```
<html>
  <body>
    <div>texte</div>
  </body>
</html>
```

Elle s'ouvre par la séquence <balise> et se ferme par la séquence </balise>. Pour HTML, une balise représente souvent un bloc de texte (le texte étant contenu dans la balise entre les deux séquences). Si la balise ne contient rien, elle peut se fermer d'elle-même avec l'instruction <balise/>. Comme vu dans l'exemple, une balise peut-être imbriquée dans une autre.

Une balise peut se voir affecter des attributs par la séquence suivante :

---

<sup>1</sup> Couche alpha : couche représentant la transparence d'une image

## HTML

```
<balise attribut= 'valeur de l'attribut'></balise>  
<div width= '100' height= '100'>texte</div>
```

Dans ce cas, un rectangle de 100 pixels par 100 pixels contenant le mot «texte» est créé.

Toute page HTML est composée de trois parties, le doctype, l'entête et le corps. Le corps compose ce qui va être affiché dans le navigateur. Le doctype renseigne le navigateur sur la version d'HTML. L'entête (la balise «HEAD») renseigne sur les éléments externes à inclure dans la page (code JavaScript, feuilles de style...), les métadonnées et le titre de la page. On obtient ainsi la structure suivante :

## HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//FR"  
"http://www.w3.org/TR/html4/loose.dtd">  
<HTML>  
  <HEAD>  
    <META ...attributs des metadonnées/>  
    <TITLE>le titre de la page</TITLE>  
    <SCRIPT ...lien vers le javascript/>  
    <LINK ...lien vers une ressource externe/>  
  </HEAD>  
  <BODY>  
    Balises qui seront affichées sur la page  
  </BODY>  
</HMTL>
```

Une fois les balises et leur contenu mis en place, il reste alors à appliquer le style CSS pour finir la mise en page.

### 7.4. CSS

CSS est le langage permettant de modifier le style d'une balise. Voici une liste non exhaustive des paramètres qui peuvent être modifiés sur les différentes balises :

- la position
- la taille
- la typographie
- les bordures (couleurs, taille, type)
- le fond (image, couleur, transparence)
- la position verticale (l'ordre « d'empilement » des balises)

Il existe trois méthodes pour appliquer du CSS aux balises HTML. La première consiste à insérer le code dans une balise `<style>` dans l'entête du fichier HTML. Cette méthode est très peu utilisée. La seconde consiste à appliquer directement le style la balise concernée grâce à l'attribut `style`. Voici un exemple :

#### HTML

```
<div style='code CSS'></div>
```

Cette méthode est utile pour faire des modifications ponctuelles. Si en revanche le style veut pouvoir être réutilisé, la troisième méthode sera privilégiée. Cette dernière est de loin la plus utilisée dans le projet.

Premièrement il faut attribuer un nom de classe à la balise (si plusieurs balises partagent le même style) ou un id (si la balise doit être la seule à posséder cette mise en page). Ceci peut être réalisé par les attributs respectifs `id` et `class`.

#### HTML

```
<div id='id de la balise'></div>
<div class='classe partagée de la balise'></div>
```

Ainsi, les lignes de codes CSS s'appliqueront à toutes les balises contenant ces identifications.

La syntaxe d'un code CSS est la suivante :

#### CSS

```
#id ou .classe
{
    Attribut : valeur ;
    height : 100px ;
}
```

Les attributs peuvent être insérés de façon illimitée entre les accolades, à condition qu'ils soient séparés par un point-virgule.

Toutes ces mises en forme ont été regroupées dans le fichier `general.css` chargé au lancement du site. À titre indicatif, ce fichier contient environ 450 lignes de codes sans compter les fichiers annexes utilisés par les librairies JavaScript.

### **7.5. La programmation**

Les deux langages vus précédemment ne sont pas à proprement parler des langages de programmation, ce sont des formats de données. Ils se contentent de mettre en forme et d'organiser les données. Un programme (le navigateur) se charge de traduire les informations qu'apportent HTML et CSS en interface graphique (GUI : graphical user interface).

Pour effectuer les tâches incluses dans le cahier des charges, un vrai langage de programmation devra être utilisé. Celui-ci s'exprime en modifiant ces formats de données. En réalité, la programmation ne fait que créer des balises HTML, ou bien du code CSS.

Le résultat du calcul côté serveur est un texte HTML. Côté client, en revanche, la programmation permet de modifier le GUI après l'interprétation du fichier source envoyé par le serveur. La partie suivante décrit la réalisation de ces programmes.

## **8. Programmation : aspects technologiques**

### **8.1. Choix des langages de programmation**



Plusieurs langages sont actuellement utilisés dans le domaine du web. La technologie la plus employée est PHP. C'est d'ailleurs la technologie qui a été utilisée afin de réaliser ce site. En réalité, un autre langage (JSP, .NET) aurait pu être employé de manière presque indifférente. Le choix s'est arrêté sur PHP en raison de la plus grande disponibilité de ce langage chez les différents hébergeurs, ainsi que sa communauté très active sur le web. Les ressources et formations sont ainsi plus accessibles.



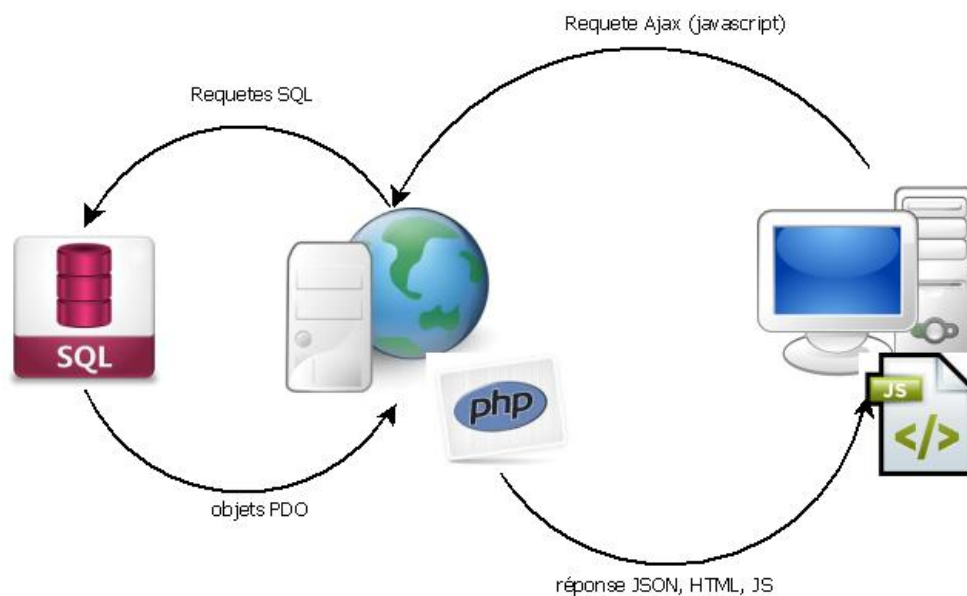
Pour ce qui est du langage côté client, deux concurrents majeurs s'imposent: Flash® et JavaScript. De ce côté, c'est JavaScript qui a été utilisé. Les raisons de ce choix sont multiples. Tout d'abord, à l'instar de Flash®, JavaScript est présent sur tous les navigateurs (y compris sur les smartphones), ce qui permet une plus grande accessibilité. De plus, il s'agit d'une solution totalement gratuite. L'utilisation de Flash® est aussi gratuite, mais les logiciels permettant de le développer étaient payants au moment où le projet a commencé. JavaScript (tout comme PHP) ne nécessite pas de compilation, c'est un langage de script. La compilation est l'opération de traduction du langage de programmation (langage humain) en langage machine (binaire). Elle permet d'obtenir un programme dit « exécutable ». Il est nécessaire de réaliser cette opération à chaque nouvelle version du logiciel. Utiliser un langage de script offre donc une certaine flexibilité dans le développement puisque le langage de programmation est seulement interprété par le moteur de rendu (donc directement exécutable). La contrepartie est que les scripts sont réputés moins rapides. Flash® nécessite une précompilation. Le compilateur rassemble tous les fichiers du programme au sein d'un seul fichier « .swf », mais ne transforme pas le code en binaire. Il sera interprété au sein du lecteur flash (flash player). Celui-ci peut compiler le code à la volée si un gain de performance est envisageable (2).



Le langage de requête pour la gestion de la base de données est SQL, le langage le plus courant et documenté sur le web. Il s'accorde parfaitement avec PHP qui se charge de créer la connexion, envoyer les requêtes et décoder la réponse de manière transparente pour le développeur. Le SGBD (système de gestion de base de données) est MySQL. Il est aussi très courant chez les hébergeurs. Il possède l'avantage d'être gratuit et multiplateforme. On notera que le SGBD importe peu pour la programmation puisque n'importe quel système pouvant recevoir des requêtes SQL peut être utilisé. Pour un hébergement Windows, SQL server aurait été privilégié.

## **8.2. Fonctionnement du binôme PHP/AJAX**

Les deux langages de programmation sélectionnés vont s'articuler de la manière suivante. Lorsque la page est chargée, le programme JavaScript permet de mettre à jour les différentes parties du site sans besoin de recharger l'intégralité de la page. Une requête est envoyée au serveur par l'intermédiaire de JavaScript. Le serveur calcule la réponse grâce à PHP en s'interfaçant avec la base de données SQL. Il retourne alors une réponse au format HTML/JavaScript au programme JavaScript exécuté sur l'ordinateur client. Ce dernier peut alors mettre à jour le contenu visible par l'utilisateur.



**Figure 4 : fonctionnement du binôme PHP/AJAX**



### **8.3. CMS vs nouveau développement PHP**

L'utilisation d'un CMS (content management system) est une voie possible pour la création d'un site web. Ces programmes (codés en PHP la plupart du temps) permettent de créer du contenu à partir d'une interface intuitive, sans nécessiter réellement de développeur. L'avantage de ce type de système est une création accélérée du site internet. La plupart peuvent être améliorés par l'ajout de plug-ins afin de réaliser certaines tâches. L'inconvénient de ce type de système est le manque de souplesse. Parmi les CMS les plus connus, on compte : WordPress, Joomla, Drupal, Xoops...etc.

Pour ce site, l'implémentation d'un CMS n'aurait pas été adéquate du fait de la complexité de la gestion des droits ; la présence de plusieurs types d'utilisateurs pouvant ou non administrer le site rend la tâche difficile. De plus, il faut connaître suffisamment le CMS pour pouvoir développer les plug-ins nécessaires. Il a donc été choisi de développer un framework PHP simple, robuste contenant des bases qui permettront un développement ultérieur accéléré.

Un framework est un ensemble coordonné de bibliothèques. Une bibliothèque est un morceau de code dédié à une tâche spécifique : gérer une base de données, faire de la retouche d'image... Le but est donc de permettre un développement plus rapide des applications. Il est en réalité impossible de travailler sans bibliothèques, car les langages de programmation contiennent tous des bibliothèques de base, ne serait-ce que pour faire une multiplication. Il existe des frameworks PHP libres de droits qui permettent d'augmenter la puissance du langage. Cependant, ces programmes répondent à une architecture très particulière et peuvent être difficiles à appréhender. Maîtriser un framework peut demander plus de temps que d'en écrire un plus léger qui répond tout juste aux besoins du projet. De plus, ces architectures sont souvent lourdes, du fait qu'elles doivent pouvoir être utilisées afin de créer tout type de sites (sites de e-commerce, informatifs, réseaux sociaux...). Ceci peut grandement impacter les performances du serveur.

Pour ces deux raisons, il a donc été choisi d'adapter un framework conçu lors d'un travail précédent. Les détails de son fonctionnement et son architecture seront traités dans la partie sur le développement PHP.

#### **8.4. Choix de l'environnement de programmation intégré (IDE)**

En théorie, un simple éditeur de texte peut suffire à la programmation PHP et javascript. Cependant, des environnements de programmation intégrés (IDE) existent pour ce type de tâche. Ils offrent des fonctionnalités qui facilitent la programmation.

La première fonctionnalité importante est le correcteur syntaxique. Il permet de déboguer le code avant même de l'exécuter. Cela ne prévient pas les erreurs de programmation, mais évite de lancer un code que le serveur ne pourra de toute manière pas interpréter. Le correcteur s'accompagne souvent d'une autocomplétion syntaxique, ce qui signifie que le logiciel devine la fin des instructions pour le développeur.

Ils disposent en général d'une autre fonctionnalité utile : l'explorateur de projet. Il permet de naviguer dans le projet sans utiliser l'explorateur de fichier fourni avec le système d'exploitation. Ceci évite les aller-retour incessants entre les deux programmes, surtout lorsque le code est très segmenté (réparti dans plusieurs fichiers) comme ici.

Plusieurs logiciels sont disponibles gratuitement sur internet pour ce type de développement. Pour citer les plus importants :

- éclipse (développé par oracle) + PDT plug-in
- Zend studio
- Netbeans



Le choix s'est porté sur Netbeans de par sa très bonne intégration du code JavaScript. Il est de plus disponible sur toutes les plateformes. L'installation est simple en comparaison de celle d'éclipse qui dispose d'un très grand nombre de versions auquel il faudra rajouter des plug-ins. Il dispose de toutes les fonctionnalités que ses concurrents possèdent. L'autocomplétion de ce logiciel est de plus relativement puissante. Celui-ci regarde dans tout le projet afin de proposer des classes, fonctions, variables créées par le développeur.

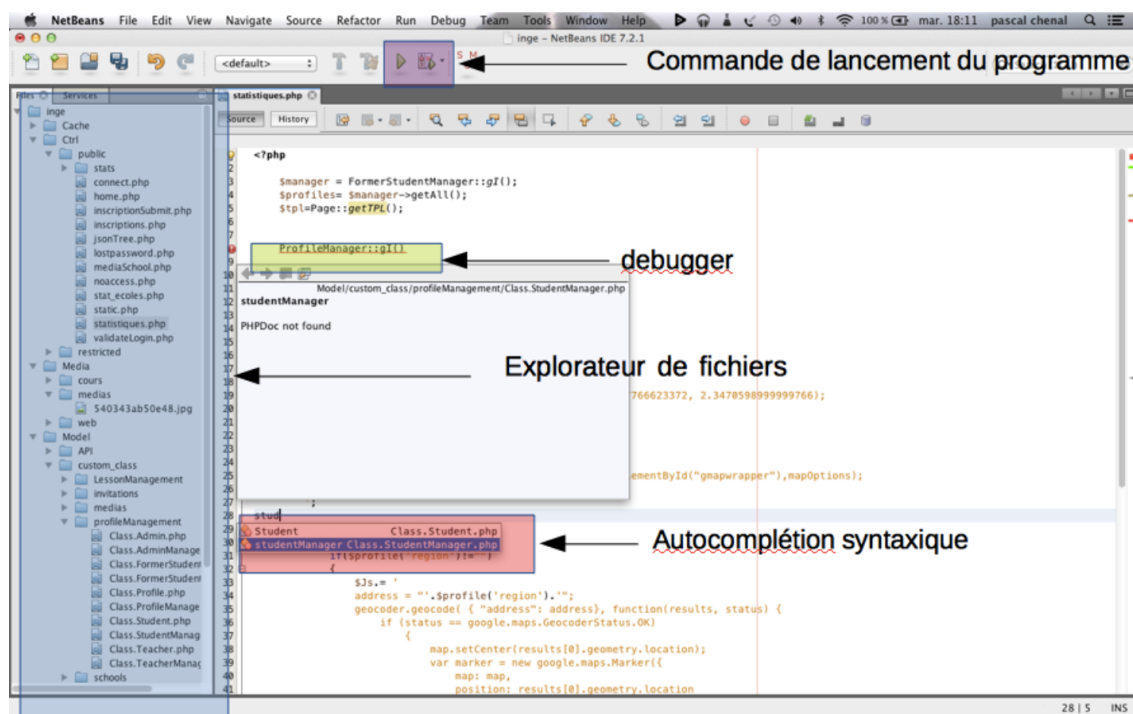


Figure 5 : L'environnement NetBeans

### 8.5. L'environnement d'exécution

L'utilisation de PHP nécessite un environnement d'exécution afin de tester le programme en local (le site n'est pas mis en ligne). Il est donc nécessaire d'installer un serveur comportant PHP et le SGBD (service de gestion de base de donnée).

Il est possible d'installer et de configurer tous les éléments de manière indépendante. Cependant, des solutions globales gratuites existent sur le web. Celle qui a été utilisée est MAMP (la seule disponible pour MacOSX).

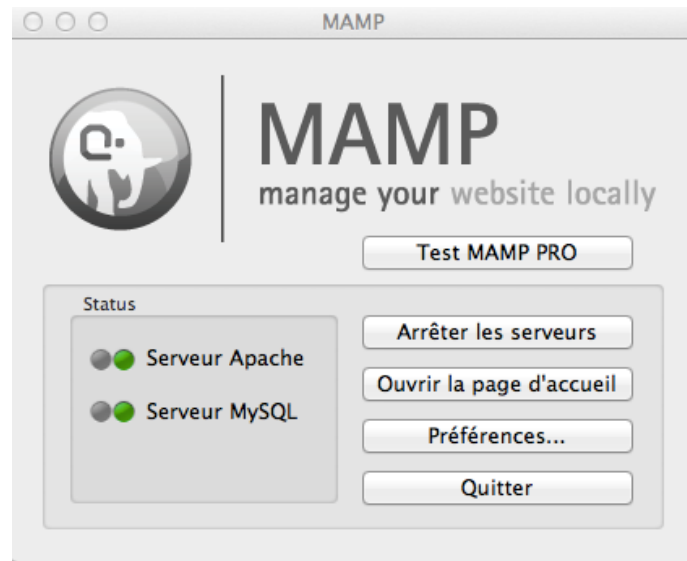
Ce logiciel propose d'installer les éléments suivants :

- le serveur apache : serveur web principal, il gère les requêtes
- PHP 5 : moteur d'exécution PHP, interprète le code PHP
- Mysql : SGBD, gestion des requêtes SQL et stockage de la base
- PHPMyAdmin : interface web pour mySQL (édition de base de données)
- Plug-ins utiles : essentiellement des plug-ins PHP



Le serveur apache est un programme dont la tâche est de répondre aux requêtes de l'utilisateur sur le port 80 (port des services web). Il implémente le protocole HTTP. Ainsi, il faut l'interfacer avec le moteur d'interprétation PHP qui lui renverra le résultat de l'exécution du code. MySQL se charge de gérer la base de données. Il est nécessaire de configurer PHP pour qu'il puisse communiquer avec MySQL. PHPMyAdmin est une interface programmée en PHP qui permet d'administrer la base de données. Il s'installe de la même manière que le site web qui sera programmé.

Le principal intérêt de MAMP est d'éviter la configuration de ces différents éléments afin qu'ils fonctionnent correctement ensemble. Le logiciel fournit une interface qui permet de gérer tous ces services au même endroit. Le démarrage et l'extinction de ceux-ci se fait de manière intuitive grâce à une fenêtre. Des versions pour Windows et Linux sont aussi disponibles; respectivement WAMP et LAMP.



**Figure 6 : interface de gestion de MAMP**

L'accès au site web se fait grâce à un navigateur traditionnel sous l'adresse « <http://localhost> ». Il est alors possible de configurer l'IDE pour qu'il ouvre cette page pour le test du site.

## **9. Concepts de programmation**

### **9.1. Les classes**

Avant d'entrer dans le détail de la structure du logiciel, il est important de parler de la méthode de programmation qui a été utilisée.

Les concepts classiquement vus en programmation sont ceux de variable (stockage de valeurs), de fonctions (réalisation d'une action spécifique), de conditions (effectuer des tests) et de boucles (répétition d'une tâche). Ces concepts correspondent aux « briques » essentielles de tout langage de programmation. Pour la tâche qui a été réalisée, ils sont cependant trop restrictifs. Ils ne permettent pas une réutilisation optimale du code et nécessiteraient un nombre plus important de lignes pour effectuer une même action. Il sera discuté ici de programmation orientée objet (POO)(3).

Le premier concept à cerner est celui de classe. Cette dernière est un regroupement de variables (appelées attributs) et de fonctions (appelés méthodes). Une classe définit les attributs et les actions qui définissent un objet.

Ex : Définir une voiture nécessite de connaître :

- ses attributs : couleur, puissance, taille...
- ses méthodes : avancer, reculer, tomber en panne...

Les méthodes utilisent les attributs ou bien les modifient (avancer nécessite de connaître la puissance, tomber en panne met une pièce hors d'usage). La syntaxe PHP pour la déclaration d'une nouvelle classe est la suivante :

**PHP**

```
Class voiture                                //déclaration d'une classe « voiture »
{
    public $couleur= 'rouge' ;                //déclaration d'attribut
    public $puissance =100;                   // déclaration d'attribut

    public function rouler()                  //méthode
    {exécuter l'action de rouler...}
}
```

Afin de créer un objet à partir d'une classe, il suffit d'appeler cette dernière

**PHP**

```
$voiture_rouge = new voiture() ;
```

Une voiture générique sera construite en une ligne de code. Un nombre indéfini d'objets peuvent être créés à partir d'une même classe; tous avec des attributs différents. Les méthodes restent identiques, en revanche leurs actions peuvent différer du fait que leurs attributs soient différents. Ceci permet de faire une économie substantielle de code.

Une valeur de défaut est donnée à tous les attributs, pour pouvoir créer un objet « générique ». Ils pourront être récupérés après la création de l'objet grâce à un accesseur, qui est en réalité une méthode. La modification d'un attribut se fait grâce à un mutateur, qui est aussi une méthode :

## PHP

```
$voiture_rouge = new voiture() ; //construction d'une voiture générique
$voiture_rouge->couleur ; //accesseur, renvoi la couleur
$voiture_rouge->couleur = 'rouge' ; //mutateur, modifie la couleur en rouge
$voiture_rouge->rouler() ; //appel d'une méthode
```

### 9.2. écriture d'une méthode

L'écriture d'une méthode se fait de la même manière que l'écriture d'une fonction. Elle prend des arguments et renvoie une valeur. La seule différence notable est la possibilité d'interagir avec des attributs de l'objet. Ceci se fait grâce au mot clef « this ».

## PHP

```
Public fonction rouler($vitesse,$temps) //entête de la méthode avec les arguments
{
    $this->position = $vitesse * $temps ; //modification de l'attribut par un mutateur
    return $this->position ; //retourne la valeur de la nouvelle position
}
```

### 9.3. Constructeur de classe

Le constructeur de classe est une méthode particulière appelée lors de l'instanciation (ou création) de l'objet. Celle-ci est totalement optionnelle lors de la création d'une classe. En effet, si elle n'est pas spécifiée un constructeur générique (n'effectuant aucune action) sera appelé.

Cette méthode permet de réaliser certaines tâches, le plus souvent vitales au bon fonctionnement de l'objet. Elle est souvent utilisée pour fixer les attributs initiaux de l'objet. Voici un exemple :

## PHP

```
Class voiture
{
    public $puissance
    public function __construct($puissance) //constructeur avec argument puissance
    {
        $this->puissance = $puissance ; //mutateur
        $this->Démarrer_voiture() ; //appel de méthode
    }
}
$voiture1= new voiture('90') ; //création d'une voiture de 90 cv
```

#### 9.4. Portée des variables et des méthodes

Dans le code précédent, les attributs et méthodes sont précédés du mot clé « public ». En réalité, il s'agit de la portée de la variable. Cela signifie qu'elle peut être modifiée à l'extérieur de l'objet par un mutateur comme vu précédemment. Dans le cas d'une méthode, c'est le même principe. La méthode peut directement être appelée de l'extérieur. En remplaçant le mot « public » par « private », la méthode ou l'attribut ne pourra être accédé que par une méthode interne.

Il convient, pour respecter le principe d'encapsulation, de mettre le mot « private » devant toutes les entités auxquelles le développeur ne doit pas accéder. Ceci permet d'éviter d'éventuelles erreurs. Le constructeur devra toujours être public. Dans le cas contraire, les objets ne pourront être créés.

Les mots « private » et « public » peuvent être succédés par l'attribut « static ». Ceci sert à définir une méthode ou un attribut statique. Ces dernières ne sont alors pas attachées à l'objet, mais à la classe. Toute modification d'une variable statique se répercute dans tous les objets qui ont été créés à partir de celle-ci.

#### PHP

```
Class voiture
{
    public static $couleur ='rouge';    //attribut statique
    public $puissance =30;              //attribut
}

$voiture1 = new voiture() ;    //voiture rouge,30 de puissance
$voiture2= new voiture() ;    //voiture rouge,30 de puissance
$voiture1->puissance = 20 ;    //30 pour voiture 2 et 20 pour voiture 1
$voiture1::couleur = 'bleu' ;  //les voiture 1 et 2 sont bleues
```

Pour modifier une variable statique dans une méthode, le mot « this » est remplacé par « self ».

#### 9.5. Le pattern singleton

Le pattern singleton est une manière spécifique de concevoir une classe. Dans certains cas, il peut être utile qu'une classe ne puisse créer qu'un seul et unique objet. Cet objet



sera appelé « instance » de classe. Lorsque la classe est appelée, elle crée un objet si cela n'a pas été fait, sinon il renvoie l'objet précédemment créé. C'est la méthode `getInstance()` qui fera cette vérification. Elle appellera le constructeur au besoin. L'accès à l'objet se fera toujours par cette méthode statique. Tous les attributs et méthodes sont alors statiques.

## PHP

```
Class voiture
{
    private static $instance;

    protected function __construct()
    {
        Instructions d'instanciation... //instructions dans le constructeur
    }

    public static function getInstance ()
    {
        if (!isset (static::$instance)) //test sur la présence de l'instance
            self::$instance = new self(); //créé un nouvel objet
        return static::$instance; //retourne l'objet créé
    }
}

$instance_voiture = voiture ::getInstance(); //retourne l'instance dans le code
```

### 9.6. L'héritage

L'héritage est vraisemblablement le concept le plus puissant de la programmation orientée objet. Il permet de définir une classe fille à partir d'une autre classe et ainsi de partager tous les attributs et méthodes avec cette dernière.

Ex :

Une classe «Mammifère» a déjà été programmée. Elle contient tous les attributs d'un animal de cette classe. Nous décidons de créer une classe «félin» qui comporte ses attributs propres. Plutôt que de réécrire l'ensemble des caractéristiques de la classe «mammifère» dans la classe «félin», il suffit de la déclarer comme héritant de la classe «mammifère» et ensuite de rajouter ses attributs propres. Ce concept est extensible à plusieurs générations (création d'une classe «chat» par exemple).

```

class Mammifere
{
    public $vertebre=true ;
    public $amniote=true ;

    public function manger() {...}
}

class felin extends Mammifere
{
    public dents =30 ;
    public griffes =true ;
}

```

Il est important de préciser que les méthodes et attributs parents peuvent être réécrits différemment dans la classe fille. Ceci vaut pour le constructeur qui est aussi hérité. Les méthodes parentes peuvent être accédées dans la classe fille par le mot-clef « parent :: ».

### 9.7. Architecture et format UML

Le concept d'héritage fait apparaître une 'arborescence' de classes (comme un arbre généalogique). Il convient avant de commencer à programmer de décider de l'allure de cette arborescence : les méthodes qui seront partagées, les attributs communs...pour une réutilisation optimale du code. Ceci s'appelle l'architecture logicielle. Une convention graphique a été inventée pour modéliser cette architecture : le format UML (Unified modeling Language). Ce format sera utilisé dans ce document afin de décrire l'arborescence des classes du site web.

Voici un schéma de l'exemple précédent:

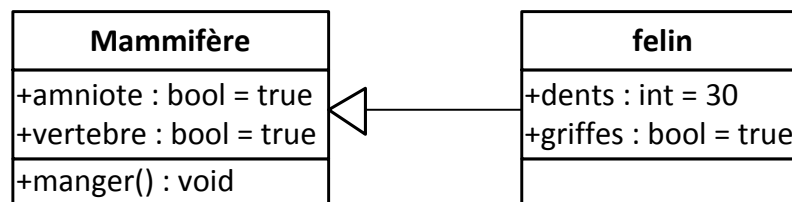


Figure 7 : exemple de schéma UML

Chaque boîte représente une classe. Le nom de la classe figure dans le premier cadre à partir du haut. Le second cadre liste les propriétés de la classe, le type de variable ainsi que sa valeur par défaut. Le troisième cadre liste les méthodes avec leurs paramètres entre parenthèses ainsi que la valeur de retour. « void » est utilisé lorsqu'il n'y a pas de valeur de retour. Une flèche pleine signifie l'héritage entre les classes. Les propriétés et méthodes de la classe mère ne sont pas répétées dans la classe fille. Une flèche en pointillé signifie l'utilisation d'une classe à l'intérieur d'une autre classe.

### **9.8. Les variables superglobales**

L'écriture d'un programme web est particulière, car elle s'inscrit dans un contexte de transactions entre un client et un serveur. Cette transaction n'est pas gérée à l'intérieur du programme, mais par un programme tiers (le serveur apache). Ce dernier assure la gestion des sessions utilisateurs et des paramètres de requête. Voici comment il s'interface avec le programme PHP.

#### **9.8.1. Les paramètres de requête**

Une requête nécessite de connaître l'adresse de la page demandée. Cette dernière est composée de plusieurs parties :

- Le protocole : langage de transaction entre serveur et client
- le nom de domaine : il correspond à l'adresse du serveur
- le nom de la page : adresse du fichier dans l'arborescence du serveur
- les paramètres de requête : paramètres à l'intention du programme serveur

ex : <http://www.ancephal.fr/index.php?p=home&v=e>

Cette requête demande la page « index.php » sur le serveur « www.ancephal.fr » selon le protocole « http » en passant les variables p= « home » et v= « e ». Lors de l'exécution du programme PHP, les variables de requête sont regroupées dans un tableau nommé « \$\_GET['nom de la variable'] ». Cette variable est une variable superglobale, car elle est utilisable dans tout le code PHP.

Les données issues de formulaires sont postées non pas dans l'adresse URL, mais dans l'en-tête de requête. Ces dernières sont accessibles dans la variable superglobale « \$\_POST['nom de la variable'] ».

### **9.8.2. Les sessions utilisateur**

La création d'un compte utilisateur nécessite de pouvoir identifier un usager. Lors de sa première visite, un cookie (données stockées dans la mémoire du navigateur) contenant un identifiant est envoyé vers le navigateur. À chaque connexion, cet identifiant est renvoyé en même temps que la requête. Le serveur compare le numéro qui lui est envoyé avec une base de données interne. Il renvoie ensuite au programme PHP toutes les données stockées sous la forme d'une variable superglobale appelée variable de session. Ces données sont accessibles grâce à la commande « \$\_SESSION['nom de la variable'] ». De nouvelles variables peuvent être créées et seront disponibles à la prochaine connexion.

Les cookies et la base de données utilisateur ont une certaine durée de vie. C'est pourquoi elles ne sont pas utilisées pour stocker des données sur le long terme. Une base de données externe se chargera de cette tâche. Les variables de session servent juste à identifier le client pour un certain temps. C'est pourquoi la plupart des services demandent une reconnexion au bout de quelques semaines.

## **10. Développement PHP**

### **10.1. Ressources PHP**

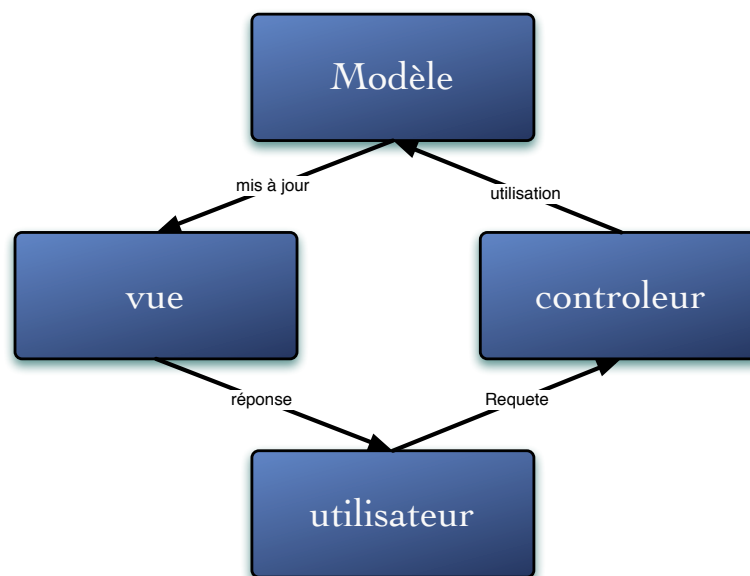
Les seules ressources qui ont été utilisées pour le programme PHP sont les suivantes :

- Talus' TPL (moteur de rendu TPL)(4)
- JpGraph (réalisation de graphiques en PHP)(5)

## **10.2. Architecture logicielle**

### **10.2.1. Généralités**

Tout le développement du framework s'est axé autour de la programmation orientée objet. Le programme utilise le pattern MVC (modèle, vue, contrôleur)(6). Un pattern représente une manière d'organiser le code, afin d'optimiser la réutilisation de celui-ci. Il permet d'imposer la même logique d'organisation aux développeurs. Ceci facilite donc la prise en main du logiciel par une personne externe au projet. Le programme est organisé de la manière suivante. La page d'index (la première appelée) sert à charger le contrôleur. Celui-ci se charge de réaliser les opérations de calcul complexes (ajouter un nouveau membre par exemple). Il utilise pour ce faire le modèle, une couche des classes qui réalisent les tâches bas niveau (ajouter les entrées du nouveau membre dans la base de données). Les modèles ne sont donc programmés qu'une seule fois et peuvent être utilisés par des contrôleurs différents. Une fois l'information traitée, elle sera mise en forme par la vue. Celle-ci se chargera ensuite de renvoyer la réponse à l'utilisateur.



**Figure 8 : Pattern MVC**

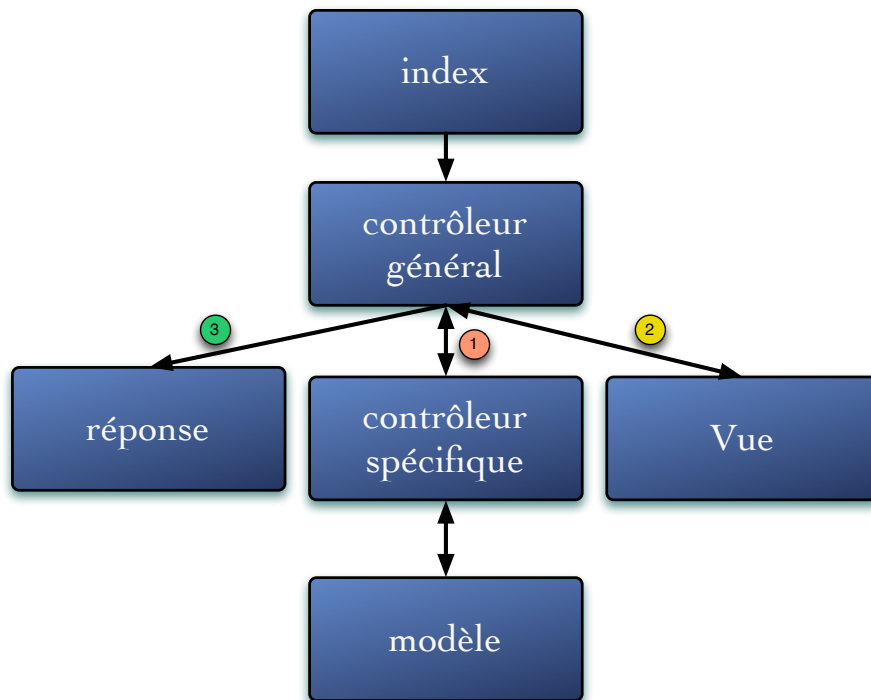
### **10.2.2. Index et contrôleurs**

Le premier élément appelé lors d'une requête est l'index ('index.php' à la racine du projet). Celui-ci ouvre la session utilisateur sur le serveur et appelle le chargeur automatique de classes. Ce dernier cherche dans l'arborescence (plus exactement le dossier « /MODEL ») la classe demandée afin de la charger. Ceci se fait de manière transparente pour le développeur lorsqu'il appelle un modèle dans le contrôleur.

La seconde fonction de l'index est d'appeler le contrôleur générique (la classe « Page ») avec les arguments nécessaires au chargement d'un contrôleur spécifique. Le contrôleur générique se charge de prendre en compte la partie invariable de toute requête :

- Détermination du type de retour attendu (AJAX, JSON, HTML)
- Appel du contrôleur spécifique
- Appel de la vue
- Retour de l'information vers l'utilisateur

Le contrôleur spécifique réalise les actions spécifiques à la page demandée. Lorsqu'il se termine, la classe « Page » appelle directement la vue. Lorsque la vue renvoie le code HTML et JS, le contrôleur général retourne la réponse à l'utilisateur et termine le programme.



**Figure 9 : Architecture MVC du code PHP**

### **10.2.3. Le contrôleur générique**

Le contrôleur générique est la classe « Page ». Celle-ci prend en argument le nom de la page ainsi que le type de page. Le type de page est donné par la variable `$_GET['ajax']`. Si celle-ci est vraie, alors le contrôleur ne chargera pas le doctype et le header, sinon ils seront chargés. Le nom de la page est donné par la variable d'environnement `$_GET['link']`. Cette variable permet de charger le contrôleur spécifique ainsi que la vue qui lui est attribuée.

Une fois la classe instanciée, le contrôleur spécifique est appelé. La classe se charge automatiquement de rechercher le contrôleur ayant le nom de la variable `$_GET['link']`. Une fois son action terminée, la vue est appelée. Les fichiers contenant la vue et le contrôleur doivent posséder le même nom. Ils seront respectivement placés dans les dossiers « /Ctrl » et « /View ».

#### 10.2.4. Architecture des modèles

Pour ce projet, les modèles se chargent essentiellement de l'interaction avec les objets de la base de données. Chaque élément de la base de données est représenté par un objet PHP. Une classe de type manager permet de gérer ces objets PHP. Les managers utilisent le pattern singleton. Une copie de cette instance est implémentée en propriété dans la classe de référence pour permettre l'utilisation des raccourcis ; l'objet peut appeler son propre manager pour effectuer des opérations sur lui-même. Tous les objets ayant une entrée dans la base de données fonctionnent selon ce même principe.

Exemple :

Voici un ordre que l'on pourrait retrouver dans un contrôleur spécifique.

Première syntaxe:

**PHP**

```
$manager = ProfileManager ::gl() ; //retourne l'instance du manager  
$profile = $manager->load(3) ; //retourne le profil avec l'id 3
```

Autre syntaxe :

**PHP**

```
$profile = new profile(array('id' => 3)) ; //créé un nouveau profile avec l'id 3  
$profile->load() ; //recherche de l'information dans la base de données
```

Le schéma UML de l'architecture est la suivante. Elle est applicable pour toutes les classes SQL du modèle.



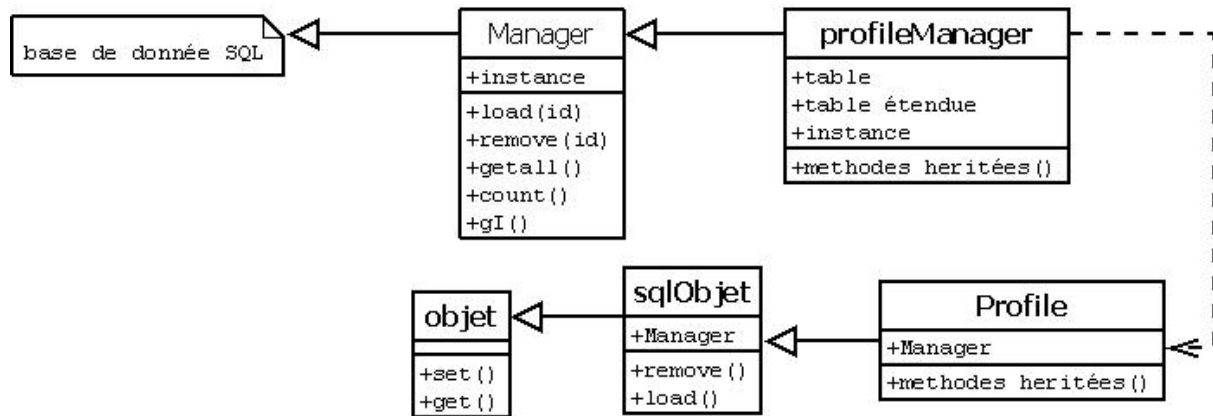


Figure 10 : Schéma UML de la gestion d'objets SQL

### 10.2.5. Le contrôleur spécifique

Le contrôleur spécifique coordonne les actions spécifiques à la page demandée; soit il renvoie une page HTML en transmettant les données à la vue, soit il réalise une action et renvoie une erreur en cas de problème. Les actions les plus courantes concernent la modification d'une entrée de la base de données (suppression, modification, ajout d'une entrée). La vue est dans ce cas désactivé grâce à la méthode « Page::noView() ». Le contrôleur spécifique vérifie aussi que l'utilisateur a le droit d'effectuer l'action concernée. Si ce n'est pas le cas, une page HTML contenant l'erreur sera envoyée grâce à la méthode « Page::redirect('noaccess'); ». Le profile de l'utilisateur est récupéré par la méthode « ProfileManager::getSession(); ».

Voici un exemple de contrôleur spécifique. Il permet de supprimer un média de la base de données :

## PHP

```

<?php
$user = ProfileManager::getSession();           //récupération de la session

if($user!=false && $user('type')== ProfileManager::ADMIN) //vérification des droits
{
    $xManager = mediaManager::gl();           //récupération de l'instance du Manager
    $xManager->remove($_GET['id']);           //suppression du média
    Page::noView();                           //désactivation de la VUE
}
else
{
    Page::redirect('noaccess');               redirection si l'accès n'est pas autorisé
}
?>

```

## 10.2.6. La vue

La vue est un moteur TPL (moteur de templates). Ce moteur utilise une trame HTML et y insère du contenu variable. Ceci permet d'externaliser le code HTML du code PHP. Ce type de moteur est suivant utilisé sur des projets impliquant plusieurs collaborateurs. Le designer du site peut se charger de créer les templates HTML sans se soucier de la programmation. Les fichiers modèles comportent une extension en « .tpl » et sont regroupés dans le dossier « /View/HTML ». Il s'agit d'une structure HTML comprenant des balises XML indiquant au moteur l'endroit où le contenu variable doit être inséré. Ces balises permettent l'insertion de variables, de boucles et de conditions.

La première étape pour lancer le moteur est d'appeler la classe correspondante. Celle-ci est directement lancée par le contrôleur général sous le nom de variable «Page::\$TPL». L'instance de cette classe peut être récupérée dans le contrôleur spécifique pour ajouter des variables. Voici un exemple d'insertion de variable.

## PHP

```

$tpl = Page::getTPL() ;           //récupère l'instance de la vue
$tpl->set('VAR',$var) ;           //assigne une nouvelle variable

```

Le moteur sera lancé automatiquement à la fin de l'exécution du contrôleur spécifique.

Les objets SQL possèdent tous une méthode pour être envoyés directement au moteur, Les managers correspondants aussi. On préférera donc envoyer les objets directement au moteur plutôt que d'entrer les variables une à une.

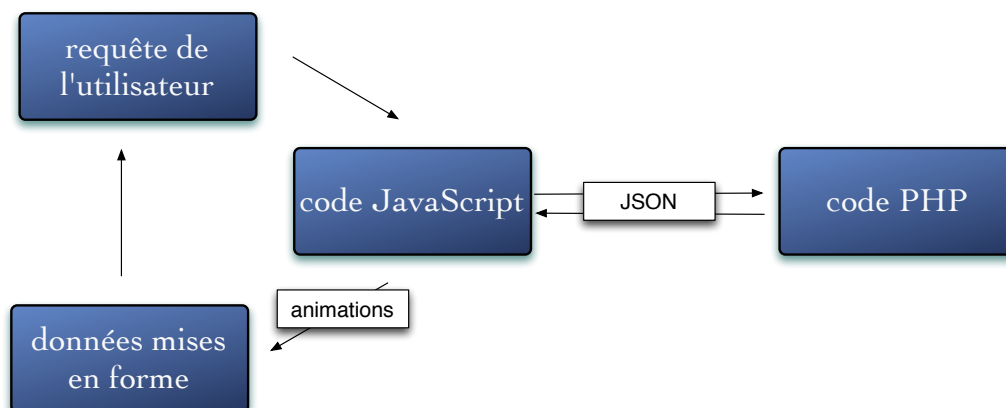
**PHP**

```
$profile1 = new Profile();           //création d'un objet profile
$profile1 = $profile->load(3) ;      //récupération des données pour l'id spécifiée
$profile1->to_TPL("user");           //envoi les informations à la vue sous le nom user
```

Le contrôleur spécifique peut ainsi rechercher un profil et l'envoyer à la vue en seulement trois lignes de code.

### 10.2.7. Utilisation du JSON

Lorsque l'on veut passer un nombre plus important d'objets, la méthode précédente est peu efficace. En terme de volume de données, il est préférable d'envoyer seulement l'objet et non la mise en forme (celle-ci représentant la majeure partie du volume de données). L'interface graphique sera reconstituée côté client grâce au code JavaScript. Cette méthode présente aussi l'avantage d'être plus souple vis-à-vis des effets graphiques. Le code JavaScript peut être développé pour accueillir plusieurs types d'objets. On pourra alors réutiliser le code (voir section sur le développement JavaScript, notamment la réutilisation de la classe de tableaux). Voici le diagramme du fonctionnement d'un tel système :



**Figure 11 : Transfert de données au format JSON**

L'implémentation du JSON se fait comme suit côté PHP :

## PHP

```
Page::noView();           //désactive la vue
$manager=ProfileManager::gl(); //récupère l'instance du manager
$rep = $manager->getAll();  //récupère tous les profils utilisateur
$array=$manager->jsonArray($rep,array("id","nom","prenom","type",));
                           //créé un tableau json
echo json_encode($array);  //renvoi l'encodage du tableau
```

Pour renvoyer du JSON, il faut désactiver la vue. Dans le cas contraire, le contrôleur général va essayer de trouver une vue qui n'existe pas. Les arguments pris par la méthode jsonArray sont les champs (propriétés) de l'objet qui vont constituer le tableau et le tableau d'objets à transformer.

### 10.3. Procédures de modification du programme PHP

#### 10.3.1. Modification de l'arborescence du site

Comme dit précédemment, l'arborescence du site est générée de manière automatique par le code PHP. C'est le contrôleur spécifique « jsonTree » (/Ctrl/public/jsonTree.php) qui se charge de calculer l'arbre. Il utilise pour ce faire le modèle « menu ».

Pour ajouter un nouvel onglet dans le menu, il faut créer un objet « menu » et le rajouter dans le tableau représentant l'arborescence. Les propriétés qui doivent être renseignées sont : un nom arbitraire (nom), le nom qui apparaît dans le menu (definition) et le nom du contrôleur spécifique (link). Le lien de parenté peut être réalisé avec la méthode « AppendTo ».

## PHP

```
$sortie[] = new menu(array("nom"=>"?", "definition"=>"?", "link"=>"?"));
//ajout d'un objet menu au tableau
$sortie[sizeof($sortie)-1]->appendTo($sortie[0]);
//relation de parenté avec la première entrée du tableau
```

Si le menu doit apparaître uniquement pour un type d'utilisateur, il doit être mis dans la condition (le « if ») appropriée.

La mise en place d'un nouvel onglet devra s'accompagner de l'écriture d'un contrôleur spécifique et d'une vue.

### **10.3.2. Ajout d'une nouvelle école**

La procédure afin de rajouter une nouvelle école dans l'interface du site a été facilitée grâce au concept de classes. Ainsi, il suffit d'ajouter une nouvelle entrée dans la base SQL grâce à l'interface graphique du SGBD pour que le site se mette automatiquement à jour. L'arborescence est générée grâce à une boucle qui explore toutes les entrées de la base et est donc mise à jour automatiquement.

### **10.3.3. Modification du choix des listes déroulantes**

Les options disponibles dans les listes déroulantes présentes dans les formulaires du site sont stockées à l'intérieur des classes de type manager. Par exemple, les différents secteurs d'activités proposés lors de la création d'un profil d'ancien sont stockés dans la classe « FormerStudentManager. ». Seul l'index de table SQL est enregistré dans la base de données. La modification du tableau doit être réalisée avant la mise en service du site afin de ne pas modifier l'ordre (l'index) de ces options. La seule opération qui peut être réalisée après la mise en service est l'ajout en fin de table d'une nouvelle entrée.

### **10.3.4. Ajout d'un nouvel objet PHP rattaché à la base de données**

La création d'un nouvel objet PHP se fait selon plusieurs étapes :

- Création d'une table dans la base de données
- Écriture du Manager PHP
- Programmation de la classe objet PHP associée
- Programmation des contrôleurs spécifiques permettant la suppression, l'ajout, et la modification d'entrées

La visualisation et l'interface d'administration doivent être programmées côté client (en JavaScript). L'écriture de contrôleurs additionnels en PHP peut être requise, notamment pour l'exploitation de ces nouvelles données. L'arborescence du site peut avoir besoin d'être modifiée (voir §10.3.1) afin d'insérer une interface d'administration.

## **11. Programmation JavaScript**

### **11.1. Avant Propos**

#### **11.1.1. Capacités de JavaScript**

JavaScript permet de faire beaucoup de choses. Les principales fonctionnalités utilisées sont les suivantes :

- éditer des balises HTML
  - o supprimer/créer des balises
  - o modifier le style CSS : position, couleurs, bordures...
  - o ajouter du texte
- effecteur du transfert de données avec PHP par le biais d'AJAX
- effectuer un travail de calcul : vérification de formulaires...

L'une des utilisations de JavaScript est la modification du style CSS. En réalité, pour JavaScript, chaque balise est un objet dont les propriétés CSS sont des attributs (regroupés dans la variable objet style). Une fois la page chargée et le style CSS appliqué, il est possible de modifier dynamiquement le style des balises via JavaScript (de même qu'il est possible de manipuler tous les autres attributs d'une balise). Ceci permet entre autres de réaliser des animations en modifiant le style CSS de manière répétée. Par exemple, un déplacement latéral de quelques pixels toutes les 100 msec donnera l'illusion que l'objet se déplace progressivement.

#### **11.1.2. Choix du framework principal**

Le framework principal utilisé sur ce projet est Mootools. L'une des raisons qui ont aiguillé le choix vers ce framework est la présence d'une interface qui facilite la création est la gestion des classes de programmation. Il offre des solutions qui ne sont pas implémentées initialement dans le langage, comme l'héritage de classe, les événements...etc. Toutes ces notions ont été largement utilisées pour la réalisation du site. Il est ainsi plus aisé de reconstruire les objets présents en PHP sur le serveur dans l'environnement JavaScript. Les informations nécessaires à cette reconstruction sont

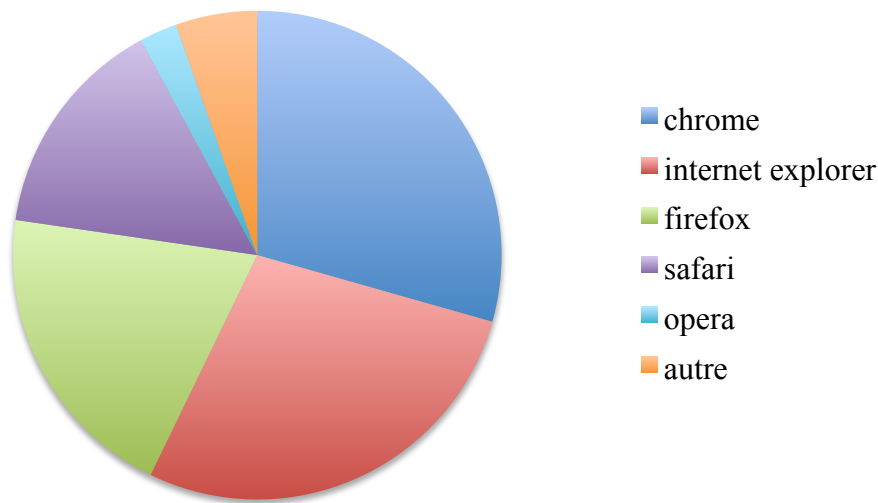
transportées par JSON. Des fonctions d'animation sont qui plus est présentes dans le framework.

Mootools dispose d'une collection relativement abondante de plug-ins libres. Ce point est important, car tout plug-in est développé pour un framework spécifique (que ce soit JQuery, mootools ou Dojo). Des problèmes de compatibilité peuvent de plus apparaître avec l'utilisation de deux frameworks différents sur un même page (en plus de surcharger la page inutilement). Il convient donc de se renseigner sur la présence de ces plug-ins avant de faire un choix sous peine de devoir les développer soi-même. La seule librairie qui n'est pas spécifique à Mootools est l'API google qui s'accorde avec tous les frameworks.

### **11.1.3. Différences d'interprétation entre navigateurs**

Comme énoncé précédemment, JavaScript est un langage de script, en comparaison avec les langages compilés. La différence entre ces deux types de langages est la suivante : le « texte » d'un langage compilé est envoyé à un programme nommé « compilateur » qui convertit ce texte en langage machine (spécifique à l'architecture de l'ordinateur et à son système d'exploitation). Celui-ci pourra alors être exécuté sous forme binaire (le .exe sous Windows). Au contraire, le script est dépendant d'un programme (ou moteur) pour être interprété. L'avantage de ce système est qu'il suffit de créer un moteur spécifique à chaque plateforme. Le script devient universel. Pour un langage compilé, il faut créer autant de versions du programme qu'il y a de plateformes. La particularité de JavaScript est qu'il existe une grande variété de moteurs d'interprétation (tous les navigateurs en implémentent un différent). Ainsi, d'un navigateur à l'autre, l'interprétation n'est pas forcément totalement identique (tout comme le code HTML et CSS). Des spécifications sont éditées par un comité international d'industriels afin de définir les standards pour ce langage. Cependant, les industriels ne parviennent pas toujours à un accord, et l'implémentation continue à diverger malgré les longues années d'existence de cette institution. En ignorant ces différences, le site risque d'être inaccessible à une grande partie des utilisateurs. Les parts de marché des différents navigateurs sont les suivants:

**parts de marché des différents navigateurs en  
novembre 2012 (sondage W3Counter)**



**Figure 12 : parts de marché des différents navigateurs en novembre 2012**

Les principaux navigateurs se partagent le marché de manière presque égale. Ne pas prendre en compte une exception pour un navigateur peut alors exclure jusqu'à 30% des internautes.

Tenir compte de ces différences peut devenir relativement lourd, car cela nécessite d'introduire des exceptions dans le code. Dans ce cas, l'utilisation d'un framework JavaScript est relativement utile. La plupart des frameworks sont dits cross-browsers. Ils s'exécutent de manière égale sur tous les navigateurs. En passant par les classes de ce programme, la compatibilité est assurée de manière transparente.

#### **11.1.4. Liste des ressources utilisées**

Voici la liste des bibliothèques utilisées dans le développement de la plateforme :

- Mootools : framework principal(7)
- formcheck : fonction qui facilite la validation de formulaires(8)
- datepicker : un sélectionneur de date(9)
- fancyupload : fonction d'upload avec barre de progression (utilise flash)(10)



- mediabox Advance : permet de visualiser des images, des sons ou des vidéos (utilise flash)(11)
- Google API : Pour la gestion de la carte des anciens(12)

#### **11.1.5. Connexion asymétrique et programmation événementielle**

La différence majeure entre la programmation coté serveur (PHP) et celle côté client (JavaScript) est la gestion du temps. Précédemment, le code était relativement linéaire. Le code s'exécute à la vitesse maximale permise par le serveur. Côté client, l'utilisateur peut demander d'exécuter plusieurs actions en même temps. Certaines actions peuvent nécessiter qu'une action précédente soit terminée avant de débiter. C'est alors que la notion d'événements est utile.

Les événements sont au cœur de la programmation JavaScript. Ils sont utilisés habituellement pour interagir avec les événements du DOM<sup>1</sup>. Le programme écoute les actions de l'utilisateur et déclenche des fonctions en conséquence. L'utilisation du framework Mootools permet de faciliter la mise en place de nouveaux événements à l'intérieur même des classes.

La programmation événementielle permet de plus la gestion de connexions asymétriques. Pour le chargement classique d'une page web en HTML, le type de connexion utilisée est symétrique. Cela veut dire que le programme attend que les différentes phases de la transaction soient terminées pour continuer (requête, calcul, réponse). Dans le cas d'AJAX<sup>2</sup>, le programme n'attend pas que la fin de la requête pour continuer. La connexion est dite asymétrique. On utilise alors les événements pour assurer le traitement de la réponse lorsqu'elle arrive.

---

<sup>1</sup> DOM : Document object model. Représentation permettant à un langage tier (ici Javascript) d'interagir avec les balises HTML du document.

<sup>2</sup> AJAX : asynchronous javascript and XML : Architecture Javascript permettant l'échange de données et sa mise en forme sur le poste client.

Exemple de transaction :

**Table 2 : Exemple de transaction JavaScript**

Synchrone	Asynchrone
1 - Action 1 2 - Requête 3 - Attente 4 - Réponse 5 - Action 2 6 - Action 3	1 - Action 1 2 - Exécuter Action 2 en fin de transaction (déclaration d'événement) 3 - Requête 4 - Action 3 5 - Réponse 6 - Lancement de l'action 2

#### 11.1.6. Inclusion du JavaScript

Pour inclure du JavaScript dans un contenu HTML, il existe deux méthodes. La première consiste à insérer directement le contenu dans le code HTML grâce aux balises adéquates :

**HTML**

```
<script> code JavaScript...</script>
```

Une deuxième méthode consiste à externaliser ce code dans un fichier annexe et d'insérer un lien vers ces fichiers grâce aux balises suivantes :

**HTML**

```
<head>
  <link language= 'javascript' type='text/javascript' src='lien vers le fichier
  javascript'/>
</head>
```

La seconde méthode sera la plus utilisée. En effet, elle apporte un gain de clarté dans le code. De plus, un même fichier peut être utilisé sur plusieurs pages HTML. C'est pourquoi les librairies JavaScript sont chargées de cette manière. Les mises à jour sur le fichier se répercutent directement sur toutes les pages dans lesquelles il est inclus. En revanche, les composants spécifiques à une page seront écrits avec la première méthode. Ceci se traduit en général par le chargement et l'initialisation de classes contenues dans un fichier annexe.

## 11.2. Architecture des composantes graphiques du site web

### 11.2.1. Organisation générale

JavaScript est au cœur du développement de cette plateforme. En réalité, tout l'affichage du site se fait par son biais. Lorsque l'utilisateur se connecte au site, seul le code JavaScript est initialisé. Par la suite, toutes les requêtes de l'utilisateur seront prises en charge par JavaScript. La construction graphique de la page s'axe sur un menu principal en haut de l'écran, et d'un espace qui sera rechargé dynamiquement par le menu. La première étape a donc été de créer le menu principal et la zone d'affichage afin de pouvoir travailler sur le contenu.



Figure 13 : Organisation générale du site web

La zone de rechargement contient le plus souvent du code HTML renvoyé par PHP. L'interface d'administration est cependant réalisée à l'aide de tableaux placés dans la zone de rechargement. Ces tableaux sont générés en JavaScript. Ce code est envoyé en même temps que le HTML par le serveur.

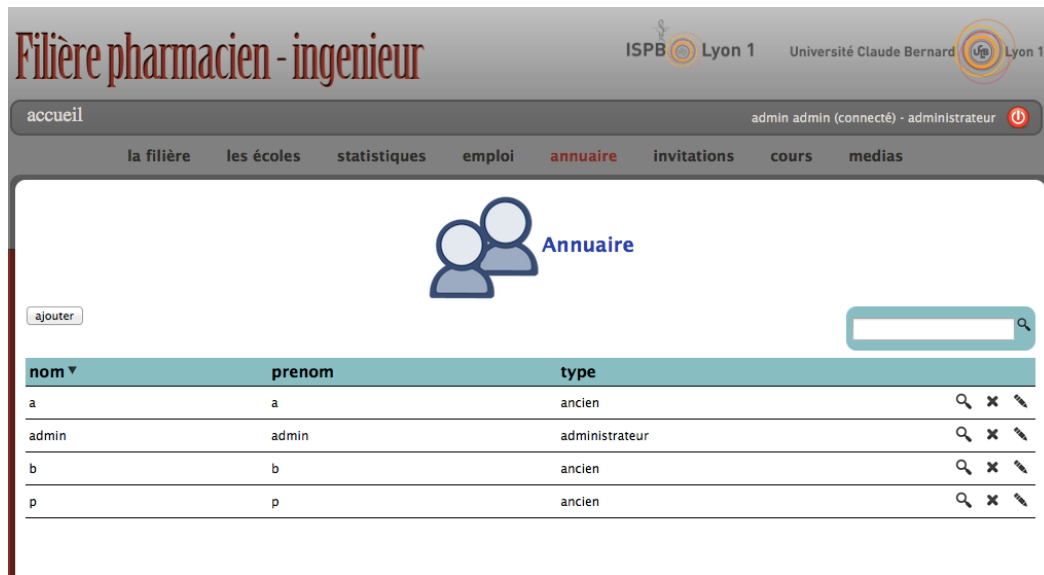


Figure 14 : Insertion de l'interface d'administration

### 11.2.2. Le menu principal

Le menu principal a deux fonctions :

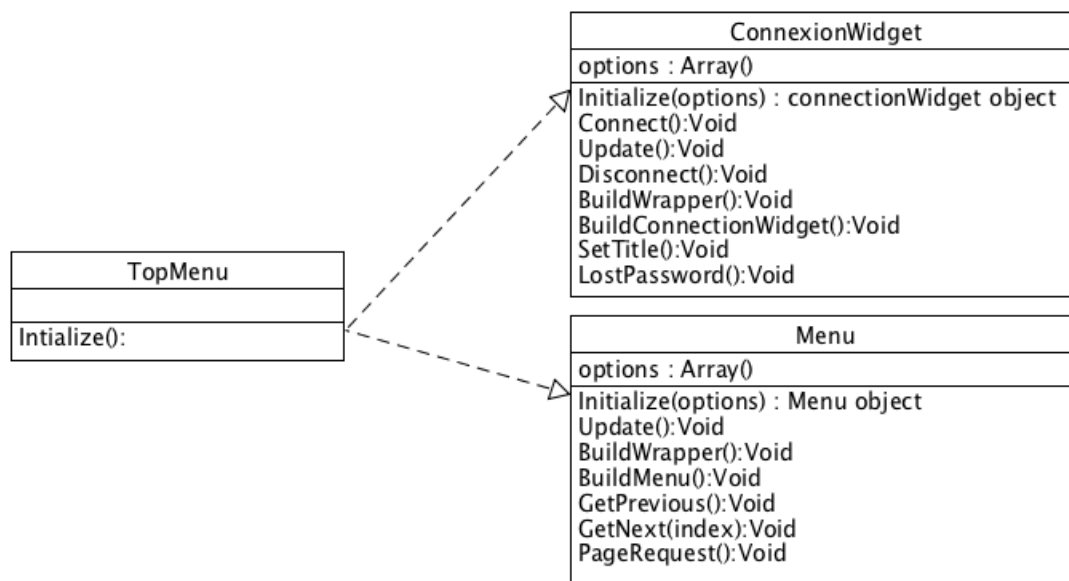
- permettre la connexion des utilisateurs
- assurer la navigation sur le site web

Les tâches sont réalisées selon un ordre bien défini :

- envoyer une requête au serveur pour savoir si une session est ouverte et si oui, quel utilisateur est connecté (enseignant, étudiant...etc.)
- Construire le widget de connexion
- Envoyer une requête au serveur pour connaître l'arbre de navigation pour l'utilisateur concerné.
- construire l'arbre de navigation

Ces tâches ne peuvent être interverties, car chacune d'elles nécessite une information que donne la tâche précédente.

Voici le schéma UML des classes formant le menu principal :



**Figure 15: Schéma UML du widget de connexion**

Les classes ci-dessus contiennent uniquement des méthodes sans valeur de retour. Ceci est volontaire. Toutes les informations sont stockées dans des propriétés de l'objet afin d'être accessibles par toutes les méthodes. Ces propriétés ne sont pas listées dans la seconde partie du tableau, car elles sont créées à l'intérieur des méthodes sans être déclarées au préalable. C'est une des libertés du langage JavaScript. L'instancieur de classe est la méthode « Initialize ». Elle prend en argument un objet JavaScript contenant les options de la classe.

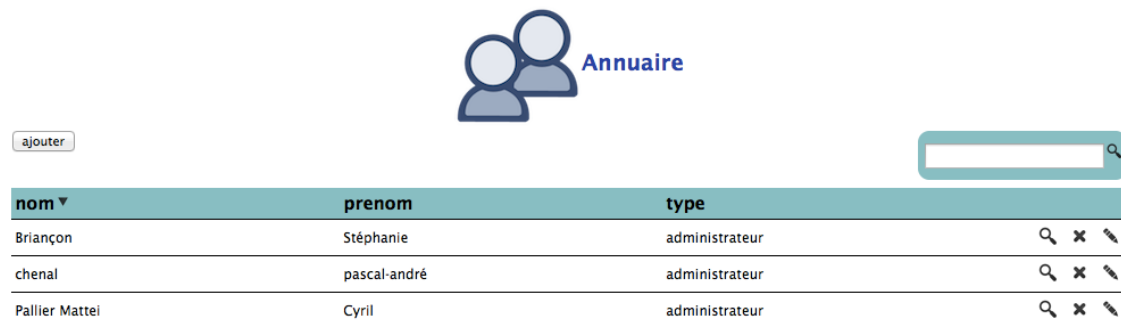
La classe principale « TopMenu » appelle successivement les classes « ConnexionWidget » et « Menu » dans sa méthode d'instanciation.

La méthode « initialize » de « connexionWidget » construit le cadre de l'élément de connexion grâce à la méthode « BuildWrapper ». Elle vérifie ensuite le statut de connexion grâce à la méthode « Update ». L'intérieur de l'élément est alors rempli grâce à « BuildConnexionWidget ». Des événements sont ajoutés sur les boutons de l'interface et appellent les méthodes « connect » ou « disconnect » en fonction du statut de connexion de l'utilisateur.

La classe « menu », qui dessine le menu permettant l'accès aux différentes pages du site, fonctionne selon le même principe. La méthode « update » récupère l'arborescence sur le serveur au format JSON. Les méthodes « BuildWrapper » et « BuildMenu » dessinent le menu à partir de l'arborescence récupérée. Des événements sont attachés sur les méthodes « GetNext » et « GetPrevious » pour permettre la navigation. Lorsqu'une nouvelle page doit être chargée, la méthode « PageRequest » sera appelée. Une requête AJAX est alors lancée pour obtenir le code HTML du contenu à afficher.

### 11.2.3. Les tableaux dans l'interface de gestion

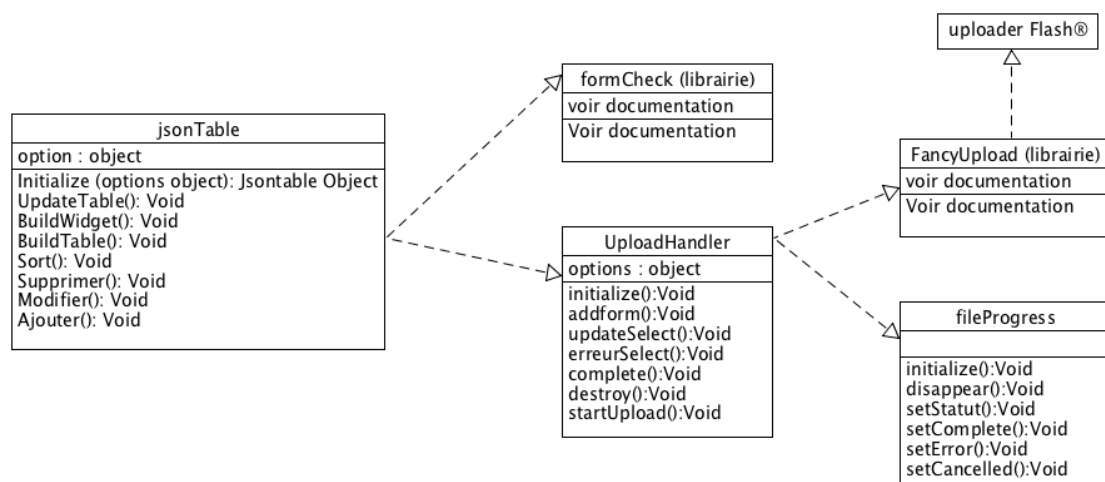
Les objets présentant une entrée dans la base de données sont gérés grâce à des tableaux. Voici une capture d'écran de l'un d'entre eux :



nom ▼	prenom	type	
Briançon	Stéphanie	administrateur	🔍 ✕ 🖋️
chenal	pascal-andré	administrateur	🔍 ✕ 🖋️
Pallier Mattei	Cyril	administrateur	🔍 ✕ 🖋️

Figure 16 : tableau JavaScript

Ces tableaux permettent l'ajout, la modification, la suppression ou la visualisation complète d'entrées de la base de données. Des fonctions de recherche et de tri ont aussi été ajoutées pour faciliter la manipulation d'un grand nombre d'entrées. Voici le Schéma UML de la classe :



**Figure 17 : schéma UML de la classe JsonTable**

L'initialisation de la classe `jsonTable` se fait comme suit. La méthode d'instanciation appelle la méthode « `BuildWidget` » qui crée le cadre du tableau. Cette dernière fonction appelle « `UpdateTable` », qui envoie une requête au serveur pour connaître le contenu de la table. La méthode « `BuildTable` » est ensuite appelée pour construire les lignes à partir du résultat de cette requête. Elle ajoute aussi les événements sur les balises, qui sont reliées aux méthodes d'ajout, de suppression et de modification d'entrées ainsi qu'aux méthodes de tri. Lorsqu'un formulaire est requis pour effectuer une action, une requête est envoyée au serveur pour le charger dynamiquement. La classe de validation (« `formCheck` ») est appelée lors de la validation de ce formulaire. La classe « `uploadHandler` » est appelée lorsqu'un fichier doit être envoyé avec le formulaire (pour les cours ou les médias). Cette classe orchestre la mise à jour de la barre de progression (classe « `fileProgress` ») en récupérant les événements issus de la classe « `fancyUpload` ». Cette dernière assure l'interface avec l'uploader Flash®. Cette technologie a été utilisée, car à l'instar de JavaScript, elle est capable de compter le nombre de bits envoyés. L'utilisateur peut ainsi être renseigné de l'avancement de l'upload.

La programmation de cette classe a été voulue assez large pour être utilisée dans toutes les situations. Voici la liste des options de cette classe :

- `width` : largeur du conteneur principal

- JsonRequest : url pour obtenir les données du tableau
- openURL : url permettant la visualisation d'un élément
- deleteURL : url de suppression d'un élément
- hideid : désactive la visualisation du numéro d'entrée dans la base de donnée
- downURL : URL prenant en charge le téléchargement d'un fichier
- storeURL : URL pour stocker une nouvelle entrée ou la modifier
- Pagination : nombre d'entrées par pages
- sortedby : champ sur lequel est exécuté le tri en première intention
- upload : true/false. Si une nouvelle entrée nécessite l'upload d'un fichier
- filterType : type de fichiers accepté par l'uploader
- maxSize : taille maximale des fichiers acceptée par l'uploader.

## 12. L'arborescence du projet

Afin de faciliter la reprise ou la modification du projet, un descriptif de l'emplacement des différents fichiers a été réalisé. Voici l'arborescence du programme :

/Index.php	Contrôleur général
/info.php	informations du serveur
/serveur_parameters.php	Informations spécifiques au serveur
/DB_ini.sql	Requêtes SQL d'initialisation de la base de donnée
/Cache/	Dossier de cache utilisé par le moteur TPL
/Ctrl/	Dossier contenant les contrôleurs PHP spécifiques
Public/	Contrôleurs accessible par tous les utilisateurs
restricted/	Contrôleurs accessibles aux utilisateurs connectés
ecole_annuaire/	Contrôleurs dédiés à l'annuaire spécifiques aux écoles
invitation/	Contrôleurs pour les invitations
lesson/	Contrôleurs pour l'échange de cours
medias/	Contrôleurs pour l'upload de médias
membres/	Contrôleurs pour l'administration des membres
monprofile/	Contrôleurs pour l'administration du compte connecté
work/	Contrôleurs pour l'administration des offres d'emploi
/Media/	Dossier contenant les ressources multimédias
cours/	Dossier où les cours mis en ligne sont enregistrés
medias/	Dossier où les médias uploadés sont enregistrés
web/	Dossier contenant les images utilisés par le site et les librairies
site/	Images du site
(...)	Dossiers utilisés par les librairies Javascripts
/Model/	Modèles PHP
API/	Classes générales du framework



custom_class/.....	Classes dédiées au projet
LesssonManagement/.....	Classes dédiées aux cours
inivitations/.....	Classes dédiés aux invitations
medias/.....	Classes dédiées aux medias
profileManagement/.....	Classes dédiés à la gestion des membres
schools/.....	Classes dédiées à la gestion des écoles
workManagement/.....	Classes dédiées aux offres d'emploi
Class.autoinput.php.....	Raccourcis pour les injections directes dans le TPL
Jpgraph/.....	Librairie dédiée à la réalisation de graphiques
/View/.....	Ensemble des ressources utilisées par la vue
CSS/.....	Code CSS utilisé par le site et les librairies Javascript
Js/.....	Code JavaScript du site (librairies et code principal)
flash/.....	Ressources flash du site (uploader et lecteur de vidéos)
html/.....	Templates du site
mail/.....	Templates dédiés aux mails
public/.....	Templates des pages accessibles pour tous les utilisateurs
restricted/.....	Templates dédiés aux utilisateurs connectés
administration/.....	Templates des formulaires
(...)	même organisation que pour les contrôleurs

## 13. Mise en service de la plateforme

### 13.1. Contraintes technologiques

L'hébergement du site web doit répondre à plusieurs contraintes. La première étant la présence de PHP dans la liste de ses programmes. Le système d'exploitation importe peu sur le choix du type d'hébergement, PHP étant un logiciel multiplateforme. La deuxième condition est la présence d'une base de données compatible avec PHP. Ces conditions sont en général remplies par tous les hébergeurs.

La question peut se poser d'utiliser un hébergement mutualisé ou un hébergement privé ; la seconde option étant la plus chère. Ces deux types d'hébergement consistent à partager ou non le serveur et la connexion internet. D'un point de vue technique, l'hébergement mutualisé ne pose pas de problèmes particuliers vu le nombre restreint d'utilisateurs attendus par mois. Une promotion ne dépassant que rarement les 20 étudiants, en supposant 5 années d'études, on ne dépasse pas les 100 utilisateurs. Ce type d'hébergement peut assurer nettement plus de connexions. En revanche, une des restrictions les plus importantes est l'attribution d'une et même adresse IP pour plusieurs dizaines de sites différents. Si un des sites se fait blacklister pour une raison

quelconque (spamming, activités illégales...), l'ensemble des sites hébergés à la même adresse est atteint. Cependant, les hébergeurs adoptent habituellement une politique assez stricte envers ce genre de pratiques, ce qui peut d'ailleurs complexifier la programmation. Afin de limiter le risque de spamming, l'hébergeur limite le nombre de mails que le serveur peut envoyer en un temps donné. Lorsque le site essaye d'envoyer des informations (nouveaux cours disponibles, emplois...etc.) cette limite est rapidement atteinte, différant ainsi l'envoi des mails.

Dans ce cas, la performance de l'hébergement n'est pas le point dominant et les contraintes peuvent être contournées ou ignorées. Ainsi il sera choisi d'utiliser un hébergement mutualisé pour des raisons économiques.

### **13.2.      *Installation de la plateforme sur le serveur***

#### **13.2.1.      Copie des fichiers du programme**



La mise en ligne du site commence par le chargement du programme sur le serveur. Pour ce faire, on utilise couramment un accès FTP (File Transfer Protocole) ou SFTP (variante cryptée). C'est un protocole standard d'échange de fichiers. Il est nécessaire de récupérer les codes d'accès FTP auprès de l'hébergeur.

Pour utiliser ce protocole, il est nécessaire d'installer un client FTP sur le poste qui fera le transfert. C'est le logiciel FileZilla, de la fondation Mozilla, qui a été utilisé pour ce projet.

The image shows the 'Général' (General) tab of the Filezilla configuration window. It contains the following fields and options:

- Hôte :** A text input field with a blue border.
- Port :** A text input field.
- Protocole :** A dropdown menu showing 'FTP – Protocole de Transfert de Fichiers'.
- Chiffrement :** A dropdown menu showing 'Connexion FTP simple (non sécurisée)'.
- Type d'authentification :** A dropdown menu showing 'Anonyme'.
- Identifiant :** A text input field containing 'anonymous'.
- Mot de passe :** A text input field filled with dots.
- Compte :** A text input field.
- Commentaires :** A large text area at the bottom.

**Figure 18 : écran de configuration de Filezilla**

Dans la partie gestion des sites, il faut remplir les champs suivants :

- hôte : l'adresse du domaine
- Protocole : choisir entre FTP et SFTP en fonction de l'accès mis à disposition
- Type d'authentification : normale
- L'identifiant
- Le mot de passe

Après s'être connecté, il suffira de glisser-déposer l'ensemble des fichiers sur le serveur.

### **13.2.2. Enregistrement des paramètres du serveur**

Les paramètres de connexion au serveur doivent être enregistrés dans le code source du site pour qu'il puisse fonctionner correctement. Pour le développement, ce sont ceux du serveur local (localhost) qui ont été utilisés. Lors de la mise en service du site, il est nécessaire de demander ces paramètres à l'hébergeur. Ceux-ci sont en général disponibles sur l'interface d'administration fournie avec l'offre d'hébergement.

Il y a trois types de paramètres :

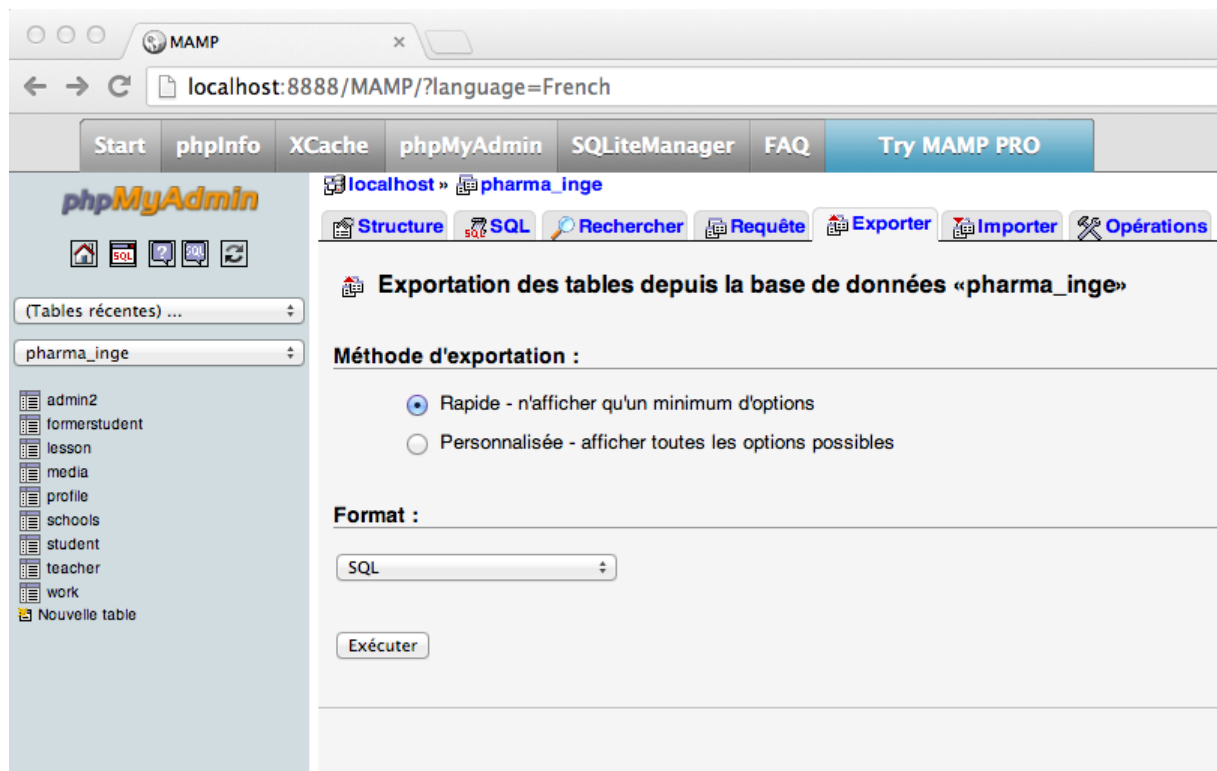
- les paramètres de la base de données
  - l'adresse du serveur hébergeant la base de données
  - le nom d'utilisateur
  - le mot de passe
  - le nom de la base
- les paramètres du serveur
  - l'adresse du site web
- les paramètres de mail
  - l'adresse du serveur SMTP (serveur d'envoi de mail)
  - le nom d'utilisateur
  - le mot de passe
  - le port de connexion
  - le type de sécurisation (ssl, tsl)
  - l'adresse mail de réponse
  - le nom de l'organisation

Ces derniers ont été regroupés sous forme de constantes dans un même fichier à la racine du site (/server\_parameters.php). Ils sont inclus automatiquement par l'index du site lors du calcul d'une page. La procédure d'installation consiste à modifier ce fichier et à l'uploader sur le serveur grâce au FTP, comme expliqué dans la partie précédente.

### **13.2.3. Installation de la base de données**

Il est nécessaire de mettre en place la structure de la base de données sur le serveur. Ceci peut se faire grâce à l'interface du SGBD fournie avec l'hébergement. La plupart des hébergeurs utilisent PHPmyadmin, celui qui est aussi installé par MAMP sur l'ordinateur de développement.

Tout d'abord, il convient d'exporter la structure de la base de l'ordinateur de développement. Ceci peut se faire dans l'onglet « exporter » de PHPmyadmin.



**Figure 19 : Exportation de la base de données**

La base devra être exportée dans le format SQL (sous forme de requêtes), en ne sélectionnant que la structure (export personnalisé). Le résultat pourra être exporté dans un fichier « .sql » ou par copier-coller.

On ouvrira ensuite l'écran PHPmyAdmin de l'hébergeur dans l'onglet « importer », dans le cas d'un fichier ou « requête » dans le cas d'un copier-coller. La base devra être créée au préalable. Pour initialiser le site, un compte administrateur doit être créé et la table contenant les informations sur les écoles doit être remplie. Pour faciliter ce processus, une copie d'une base de données initialisée se trouve à la racine du site (/DB\_ini.sql). Lors de l'installation sur le serveur, un import de cette base créera le compte administrateur (login : admin, mot de passe :admin). Pour des raisons de sécurité, ce compte devra par la suite être supprimé ou modifié.

Figure 20 : écran de requête de PHPmyadmin

## 14. Bilan du développement

Le développement de cette plateforme s'est étalé sur trois ans. Une phase de négociation avec les partenaires potentiels s'est déroulée en début de projet et a duré environ six mois. Lorsque la décision de développer un programme de manière autonome a été prise, le développement a pu commencer. Il a duré jusqu'à la mise en service définitive du site en septembre 2014. Une étape de débogage s'est déroulée pendant deux semaines après l'upload de la plateforme sur le serveur. Voici un tableau récapitulatif du code qui a été écrit au cours de cette thèse. Ces données ont été générées grâce à commande console `wc -l` dans le terminal MacOSX. Il est important de noter que le code est sous forme lisible par un développeur, c'est à dire présentant une indentation conventionnelle.

**Table 3 : détail des statistiques du projet**

		Nombre de fichiers	Nombre de lignes de code
JavaScript	Code du projet	3	1224
	Librairies	8	≈ 25 000
PHP	Code du projet	88	4 363
	Librairies	64	43 535
CSS	Code du projet	2	636
	Librairies	4	739
Templates	Code du projet	38	1577
	somme	207	77077

**Table 4 : résumé de la taille du projet**

	Nombre de fichiers	Nombre de lignes de code
Code développé pour le projet	131	7 800
librairies	76	69 274

Le site représente environ 70 000 lignes de codes, dont 7 800 ont été écrites pour ce projet. Ce chiffre semble relativement élevé pour un projet mené de manière individuelle, mais reste très raisonnable en comparaison de programmes industriels. En comparaison, un système d'exploitation comme MacOSX représente plusieurs dizaines de millions de lignes de code.

## 16. Bibliographie

1. Rabut M. Mise en place du site internet de la filière pharmacien-ingénieur de l'ISPB-faculté de pharmacie (université Claude Bernard Lyon 1). [Lyon]: Université Claude Bernard; 2003.
2. Votre premier programme avec Flex SDK.  
<http://fr.openclassrooms.com/informatique/cours/apprenez-a-programmer-en-actionscript-3/votre-premier-programme-avec-flex-sdk>. (consulté le 26/10/2014).
3. programmez en orienté objet en PHP.  
<http://fr.openclassrooms.com/informatique/cours/programmez-en-oriente-objet-en-php>. OpenClassrooms. (consulté le 26/10/2014).
4. Documentation - Talus' Works. <http://www.talus-works.net/forum-7-p1-documentation.html>. (consulté le 06/10/2014).
5. JpGraph - Most powerful PHP-driven charts. <http://jpgraph.net/>. (consulté le 06/10/2014).
6. Modèle-vue-contrôleur. <http://fr.wikipedia.org/w/index.php?title=Mod%C3%A8le-vue-contr%C3%B4leur&oldid=107958977>. Wikipédia. 2014 (consulté le 06/10/2014).
7. MooTools - a compact javascript framework. <http://mootools.net/>. (consulté le 29/09/2014).
8. MooTools Forge | FormCheck. <http://mootools.net/forge/p/formcheck>. (consulté le 29/09/2014).
9. MooTools Forge | MooTools-DatePicker.  
[http://mootools.net/forge/p/mootools\\_datepicker](http://mootools.net/forge/p/mootools_datepicker). (consulté le 29/09/2014).
10. FancyUpload. <http://digitalald.de/project/fancyupload/>. (consulté le 29/09/2014).
11. mediaboxadvanced. <http://iaian7.com/webcode/mediaboxAdvanced>. (consulté le 29/09/2014).
12. google-api-javascript-client - Google APIs Client Library for JavaScript - Google Project Hosting. <https://code.google.com/p/google-api-javascript-client/>. (consulté le 29/09/2014).

L'ISPB - Faculté de Pharmacie de Lyon et l'Université Claude Bernard Lyon 1 n'entendent donner aucune approbation ni improbation aux opinions émises dans les thèses ; ces opinions sont considérées comme propres à leurs auteurs.



## **C O N C L U S I O N S**

THESE SOUTENUE PAR : M PASCAL-ANDRE CHENAL

L'objectif de toute formation est de faciliter l'insertion de ses étudiants dans le monde professionnel et d'entretenir une cohésion au sein de sa communauté d'anciens élèves. À l'heure actuelle, internet est vraisemblablement l'outil le plus adapté pour remplir ces objectifs. Tout d'abord, un site web permet d'assurer une communication rapide entre utilisateurs quelque soit la distance qui les séparent. D'autre part, la création d'un programme permet d'automatiser une grande partie des tâches qui seraient trop lourdes à gérer pour les responsables de filière. C'est pourquoi la création d'une plateforme pour la filière pharmacien-ingénieur est devenue indispensable.

L'analyse des besoins de la filière a abouti à la création d'un cahier des charges. Le premier point est de fournir de l'information sur les débouchés de la filière. Ceci a été réalisé grâce à la création d'un annuaire d'anciens, dont les données sont récupérées à des fins statistiques. Le second point est de fournir un service additionnel aux étudiants actuels. Ceci s'est traduit par la création d'une base de données d'emplois et d'une plateforme d'échanges de supports pédagogiques. Le dernier point est d'assurer la pérennité du site. Un système de téléversement de médias permet sur le long terme d'enrichir le contenu initialement prévu.

Le site web créé au cours de cette thèse utilise les technologies PHP et JavaScript. PHP est le langage de programmation côté serveur. Son rôle est de gérer la base de données et de fournir du contenu au format XHTML. JavaScript récupère ces informations pour les mettre en forme côté utilisateur. Ces deux technologies sont couramment utilisées sur le web et permettent une compatibilité optimale avec les hébergeurs traditionnels. Le site a pu être hébergé par ANCEPHAL, une association partageant les mêmes objectifs que ce projet. Cette collaboration a permis d'aboutir à la création d'un sous domaine et d'une base de donnée distincte pour la filière pharmacien-ingénieur.

La partie écrite de cette thèse consiste à fournir une documentation précise du fonctionnement de la plateforme, afin de permettre sa modification et son administration. Une première partie s'attache à la description des besoins de la filière ainsi qu'au cahier des charges qui en découle. Une analyse des technologies et concepts de programmation

utilisés a été effectuée afin de faciliter la compréhension de l'organisation générale du programme. L'architecture du site a été ensuite abordée afin de permettre sa modification. La mise en service de la plateforme sur un serveur externe est décrite afin de permettre la relocalisation du service sur un autre serveur ou pour effectuer une sauvegarde des données. Une dernière partie dresse un bilan chiffré de ce développement.

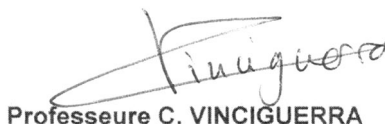
**Le Président de la thèse,**  
Nom : Mr CYRIL PAILLER-MATTEI

Signature :



Vu et permis d'imprimer, Lyon, le **29 JUIN 2015**  
Vu, la Directrice de l'Institut des Sciences Pharmaceutiques et  
Biologiques, Faculté de Pharmacie

Pour le Président de l'Université Claude Bernard Lyon 1,

  
**Professeure C. VINCIGUERRA**

**CHENAL Pascal-André**

**Création d'une plateforme web pour la filière pharmacien-ingénieur**

Th. D. Pharm., Lyon 1, 2015, 69 p.

## **RESUME**

Le but de ce travail est de fournir à la filière pharmacien-ingénieur une plateforme web afin de pouvoir soutenir son activité. Le premier objectif est de centraliser toute l'information sur les débouchés de la filière. Ceci a été réalisé grâce à la création d'un annuaire d'anciens, dont les données sont récupérées à des fins statistiques. Le second objectif est de fournir un service additionnel aux étudiants actuels. Ceci s'est traduit par la création d'une base de données d'emplois et d'une plateforme d'échanges de supports pédagogiques. Afin d'assurer la pérennité du site, un système de téléversement de médias a été mis en place afin d'enrichir le contenu initialement prévu.

Le site a pu être hébergé par ANCEPHAL, une association partageant les mêmes objectifs que ce projet.

La partie écrite de cette thèse consiste à fournir une documentation précise du fonctionnement de la plateforme, afin de permettre sa modification et son administration. Une première partie s'attache à la description des besoins de la filière ainsi qu'au cahier des charges qui en découle. Une analyse des technologies et des concepts de programmation utilisés a été effectuée afin de faciliter la compréhension de l'organisation générale du programme. L'architecture du site a été ensuite abordée afin de permettre l'ajout ou la modification de fonctionnalités. La mise en service de la plateforme sur un serveur externe y est aussi décrite afin de permettre la relocalisation du service sur un autre serveur ou pour effectuer une sauvegarde des données. Une dernière partie dresse un bilan chiffré de ce développement.

## **MOTS CLES**

Internet  
Pharmacien ingénieur  
Réseau d'ancien  
Information

## **JURY**

Président:	Mr Cyril PAILLER MATTEI,	Maitre de conférences
Membres :	Mme Stéphanie BRIANÇON	Professeure
	Mr Stéphane MICHEL	Développeur web
	Mr Rami MROUEH	Pharmacien industriel
	Mme Christine VINCIGUERRA	Professeure

## **DATE DE SOUTENANCE**

Vendredi 17 juillet 2015

## **ADRESSE DE L'AUTEUR**

40, rue du 24 mars 1852 – 69009 Lyon