

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DÉPARTEMENT D'INFORMATIQUE**



**PROJET DE FIN D'ETUDES**

**Licence Sciences et Techniques Génie Informatique**

---

**IMPLEMENTATION D'UN RESEAU SOCIAL POUR LA FST DE FES**

---

**Lieu de stage : Faculté des Sciences et Techniques de Fès**

**Réalisé par :**

**EDOUIKI Hamid  
ELKOURCHI Mohammed**

**Encadré par :**

**Pr. LAMRINI Loubna**

**Soutenu le 09/06/2016 devant le jury composé de :**

**Pr. ZARGHILI Aرسالane  
Pr. BENABBOU Abderrahim  
Pr. LAMRINI Loubna**

**Année Universitaire 2015-2016**

# TABLE DES MATIERES

Liste des abréviations .....	4
Liste des figures.....	5
Remerciement.....	6
Introduction .....	7
<b>Chapitre 1: Contexte du projet et Étude fonctionnelle .....</b>	<b>8</b>
1. Contexte du projet.....	9
1. Organisme d'accueil .....	9
2. Description du projet.....	10
3. Objectifs du projet.....	11
2. Conduite du projet.....	11
1. Choix du processus de développement .....	11
2. Planification .....	13
3. Étude fonctionnelle.....	14
<b>Chapitre 2 : Choix et Étude de la plate-forme .....</b>	<b>17</b>
1. Étude comparative des plates-formes SMS.....	18
1. BuddyPess .....	18
2. Drupal.....	18
3. Liferay.....	19
4. Elgg .....	19
2. Étude de la plate-forme Elgg .....	20
1. Les fonctionnalités .....	20
2. Le moteur D'Elgg.....	20
3. Le Modèle - vue - contrôleur d'Elgg .....	21
4. Les plug-ins .....	23

5. Modèle de données .....	23
<b>Chapitre 3 : Analyse et Conception .....</b>	<b>26</b>
1. Langage de modélisation .....	27
2. Analyse des besoins.....	27
1. identification des acteurs.....	27
2. Identification des cas d'utilisation.....	27
3. Diagramme de package .....	29
4. Diagrammes de séquences .....	29
5. Diagramme de classes .....	32
6. Diagramme de déploiement .....	33
<b>Chapitre 4 : Adaptation de la plate-forme Elgg.....</b>	<b>34</b>
1. Les outils de développement .....	35
2. Les plugins développés .....	36
1. Plugin gestion des groupes.....	36
2. Plugin gestion des étudiants .....	37
3. Plugin du thème.....	38
3. Les plugins installés .....	39
1. Plugin d'agenda .....	39
Conclusion et perspectives .....	41
References bibliographiques.....	42

## LISTE DES ABREVIATIONS

<b>FST</b>	Faculté des Sciences et Techniques
<b>RS</b>	Réseau Social
<b>RSE</b>	Réseau Social d'entreprise
<b>SNS</b>	Social Networking Service
<b>SMS</b>	Social Network Management System
<b>CMS</b>	Content Management System
<b>MVC</b>	Model-View-Controller
<b>LAMP</b>	Linux, Apache, MySQL, PHP
<b>HTML</b>	HyperText Markup Language
<b>PHP</b>	Hypertext PreProcessor
<b>UML</b>	Unified Modeling Language

# LISTE DES FIGURES

- Figure 1: Architecture pédagogique de la FST
- Figure 2: Cycle des méthodes agiles
- Figure 3: Une itération selon la méthode Scrum
- Figure 4: Planning de travail détaillant l'aspect gestion du temps du projet
- Figure 5: Architecture Modèle - vue - contrôleur
- Figure 6: Les blocs de vue d'Elgg
- Figure 7: Modèle de données d'Elgg
- Figure 8: Extrait des tables de base de données d'Elgg
- Figure 9: Diagramme de cas d'utilisation de l'enseignant
- Figure 10: Diagramme de cas d'utilisation de l'utilisateur
- Figure 11: Diagramme de cas d'utilisation de l'administrateur
- Figure 12: Diagramme de package
- Figure 13: Diagramme de séquence de création d'un groupe
- Figure 14: Diagramme de séquence d'ajout des étudiants
- Figure 15: Extrait de diagramme de classes
- Figure 16 : Diagramme de déploiement d'Elgg
- Figure 17: Hiérarchie de création des groupes
- Figure 18: Formulaire de création d'un groupe
- Figure 19: Table Elgg\_entity\_subtype
- Figure 20: Formulaire d'ajout des étudiants
- Figure 21: La page principal du réseau social
- Figure 22: L'interface d'authentification des utilisateurs
- Figure 23: L'interface de la page d'accueil
- Figure 24: une partie du fichier fr.php (plugin Agenda)
- Figure 25: L'interface du plugin agenda
- Figure 26: Formulaire de création d'un événement

# REMERCIEMENT

Louange à Dieu le clément, le miséricordieux, qui nous a donné le courage et la patience de mener à bien ce travail.

Il nous est un devoir d'exprimer notre profonde gratitude et notre estime A notre encadrante Mme Loubna Lamrini.

Nous avons eu le privilège de travailler sous votre supervision et d'apprécier vos qualités et vos valeurs. Votre sérieux, votre compétence et votre sens du devoir nous ont énormément marqués. Veuillez trouver ici l'expression de notre respectueuse considération et notre profonde admiration pour toutes vos qualités scientifiques et humaines.

Vous avez bien voulu nous confier ce travail riche d'intérêt et nous guider à chaque étape de sa réalisation. Vous nous avez toujours réservé le meilleur accueil, malgré vos obligations professionnelles. Vos encouragements inlassables, votre amabilité, votre gentillesse méritent toute admiration. Nous saisissons cette occasion pour vous exprimer notre profonde gratitude tout en vous témoignant notre respect.

À tous les membres de jury, vous nous faites l'honneur d'accepter avec une très grande amabilité d'évaluer ce rapport. Veuillez accepter ce travail modeste, en gage de notre grand respect et notre profonde reconnaissance.

# INTRODUCTION

De nos jours, les réseaux sociaux sont devenus une révolution qui change la façon dont les gens interagissent avec les autres et donnent la possibilité d'élargir et de renforcer les relations avec les personnes qui l'entourent.

Durant notre stage au sein de la Faculté des Sciences et Techniques, nous a été proposé d'implémenter un réseau social permettant de donner la possibilité à tous les acteurs de l'établissement de construire leur identité numérique universitaire. En offrant un espace numérique pour communiquer, s'informer, rechercher et collaborer.

Ce stage s'inscrit dans le cadre de nos études à la FST. Tout étudiant doit effectuer un stage, d'une période de deux mois à la fin de son année de licence, qui permet de découvrir le monde de l'entreprise et du travail, de mettre en pratique ses études et d'être prêt à affronter la vie active.

Le présent rapport décrit l'essentiel du travail réalisé lors de ce projet. Il est organisé en quatre chapitres :

Dans le premier chapitre nous présenterons l'organisme d'accueil, une description de projets et ses objectifs, ainsi une étude fonctionnelle et le processus de développement choisi.

Le deuxième chapitre contient une étude comparative des principales plates-formes et une étude technique de la plate-forme choisie Elgg.

Dans le troisième nous exposerons une conception de la plate-forme Elgg ainsi les modifications apportées à cette conception.

Le dernier chapitre est consacré à la présentation des plugins réalisés ainsi que les outils de développement utilisés.

**CHAPITRE 1:**  
CONTEXTE DU PROJET ET ÉTUDE FONCTIONNELLE



# 1. CONTEXTE DU PROJET

## 1. ORGANISME D'ACCUEIL

La Faculté des Sciences et Techniques de Fès a été créée en 1995. Elle fait partie d'un réseau national formé de 6 autres établissements du même genre situés à Béni Mellal, Errachidia, Marrakech, Mohammédia, Settat et Tanger auquel est ajouté dernièrement la Fst d'Al-Hoceima. Les facultés des Sciences et Techniques sont des établissements universitaires à caractère scientifique et technique. Elles ont été créées dans le but de développer et de diversifier les formations offertes aux bacheliers scientifiques en vue d'une meilleure intégration de l'Université dans son environnement socio-économique.

La FST de Fès se compose de huit départements qui s'activent à proposer un large éventail de formations. La figure 1 montre l'architecture Pédagogique Globale dans la FST.

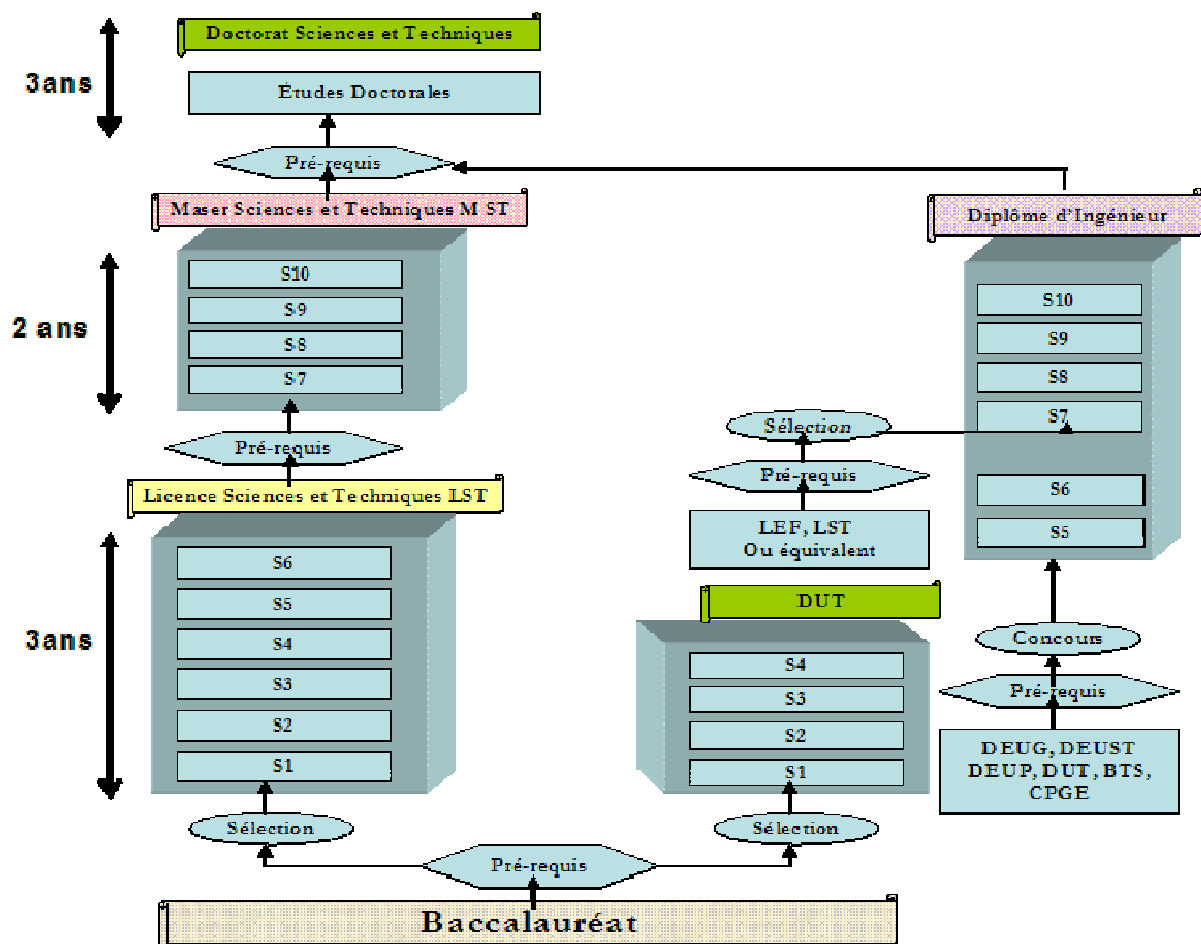


Figure 1: Architecture pédagogique de la FST

## 2. DESCRIPTION DU PROJET

Notre objectif est de mettre en œuvre une plate-forme communautaire permettant de donner la possibilité à tous les acteurs de la faculté (étudiants et professeurs) de construire leur identité numérique universitaire et de participer à un réseau social en ligne dédié à leurs activités.

Les étapes principales du projet :

- Étude comparative des principales plates-formes « open source » qui peuvent répondre aux besoins (Elgg, Drupal, ...).
- Choix d'une plate-forme en adéquation avec le contexte du projet.
- Paramétrage et développement.
- Mise en place de la plate-forme.

Avant d'étaler ces objectifs, nous définissons quelques concepts fondamentaux :

### 1. Réseau social

Un réseau social est un ensemble de relations entre un ensemble d'acteurs. Cet ensemble peut être organisé (une entreprise, par exemple) ou non (comme un réseau d'amis) et ces relations peuvent être de nature fort diverse (pouvoir, conseil, etc.), spécialisées ou non, symétriques ou non.

### 2. Réseau social d'entreprise (RSE)

Un réseau social d'entreprise est un système de communication interne accessible aux employés d'une entreprise et éventuellement aux partenaires de cette entreprise (fournisseurs, clients, distributeurs...). Son objectif est de favoriser la communication entre les individus et/ou des groupes d'individus au travers de fonctionnalités dites sociales, assez proches des réseaux sociaux traditionnels. Il se base sur plusieurs dimensions :

- **Gestion des informations personnelles**, gestion et représentation de ses données personnelles et vue personnelle sur le réseau.
- **Les réseaux sociaux et la recherche d'experts**, rapprochement des membres et recherche d'expertise.
- **Communication d'entreprise**, les différentes formes de communication entre les membres.
- **Collaboration et échange de connaissances**, travail collaboratif et capitalisation du savoir-faire, incluant les notions de gestion de projet.

### 3. Réseau socionumérique

Réseau socionumérique ou un site de réseau social est une plate-forme de communication en réseau dans laquelle les participants :

1) disposent de profils associés à une identification unique qui sont créés par une combinaison de contenus fournis par l'utilisateur, de contenus fournis par des « amis », et de données système.

2) peuvent exposer publiquement des relations susceptibles d'être visualisées et consultées par d'autres.

3) peuvent accéder à des flux de contenus incluant des contenus générés par l'utilisateur notamment des combinaisons de textes, photos, vidéos, mises à jour de lieux et/ou liens – fournis par leurs contacts sur le site.

#### **4. Réseau socionumérique de faculté**

Nous pouvons considérer la faculté comme une entreprise, ce qui donne un réseau socionumérique similaire à un RSE, et permet de déterminer un contour des besoins fonctionnels précis.

Dans chaque dimension des RSE, nous choisissons les principales fonctionnalités permettant de mettre en œuvre un RS adapté à tous les acteurs de la faculté :

- **Gestion des informations personnelles** : Profil, Tableau de bord, Flux d'activités.
- **Les réseaux sociaux et la recherche d'experts** : Contacts, Recherche des membres.
- **Communication d'entreprise** : Commentaires, Messagerie, Forums.
- **Collaboration et échange de connaissances** : Groupes, Blogs, Gestion documentaire, Calendrier.

### **3. OBJECTIFS DU PROJET**

Les principaux objectifs de ce projet :

- Choisir la plate-forme la plus adéquate aux besoins généraux : comparer entre les plates-formes open-source permettant la mise en œuvre d'un réseau socio-numérique.
- Adapter cette plate-forme aux besoins spécifiques : développer la plate-forme afin de la rendre plus adaptée avec les besoins de l'établissement.

## **2. CONDUITE DU PROJET**

### **1. CHOIX DU PROCESSUS DE DEVELOPPEMENT**

Le choix du processus de développement est déterminant dans la réussite d'un projet. Et notamment pour la mise en œuvre d'un réseau social d'entreprise.

En analysant les besoins, il s'avère que le choix d'une méthode de développement agile est le plus pertinent. Notre choix s'est porté sur la méthode Scrum.

## 1. Les méthodes agiles

Les méthodes agiles sont des formes de gestion de projets qui se veulent beaucoup plus pragmatiques que les méthodes traditionnelles. La notion de méthode agile a été formalisée aux États-Unis en 2001 dans un document intitulé *Le manifeste agile* signé par 17 spécialistes du génie logiciel sur la base de la confrontation de leurs propres méthodes utilisées depuis les années 1990.

Le but des méthodes agiles est d'augmenter le niveau de satisfaction des clients tout en rendant le travail de développement plus facile.

Ces méthodes sont plus adaptatives que prédictives et sont orientées vers les personnes plutôt que vers les processus. Elles sont principalement utilisées dans les projets informatiques et attachent une grande importance à l'implication du client (l'utilisateur final) tout au long du processus de réalisation du produit.

L'aboutissement : un cycle adopté par l'ensemble des méthodes Agiles actuelles

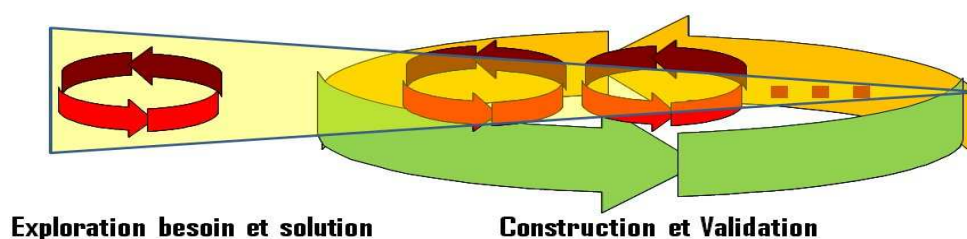


Figure 2: Cycle des méthodes agiles

Elles se veulent beaucoup plus pragmatiques que les méthodes traditionnelles et sont fondées sur quatre valeurs fondamentales :

- être réactif face au changement plutôt que s'enfermer dans le plan initial ;
- privilégier l'interaction avec les personnes plutôt que les processus et les outils ;
- privilégier un produit opérationnel plutôt qu'une documentation pléthorique ;
- privilégier la collaboration avec le client plutôt que la négociation contractuelle.

## 2. Scrum

Scrum (« la mêlée ») est une méthode agile qui s'applique à la gestion de projet. Son principe est abordé pour la première fois dans un article de Hirotaka Takeuchi et Ikujiro Nonaka intitulé « The New New Product Development Game » publié en 1986 dans la *Havard Business Review*. Elle a ensuite été développée dans les années 1990 et formalisée par Ken Schwaber et Jeff Sutherland en 1995.

Le principe de cette méthode est de focaliser l'équipe de manière itérative sur les fonctionnalités à réaliser. Le projet est ainsi découpé en modules fonctionnels qui seront réalisés, testés et livrés par séquences itératives appelés « sprint ». Chaque sprint vise à atteindre un but à partir duquel sont choisies les fonctionnalités à implémenter dans cette phase.

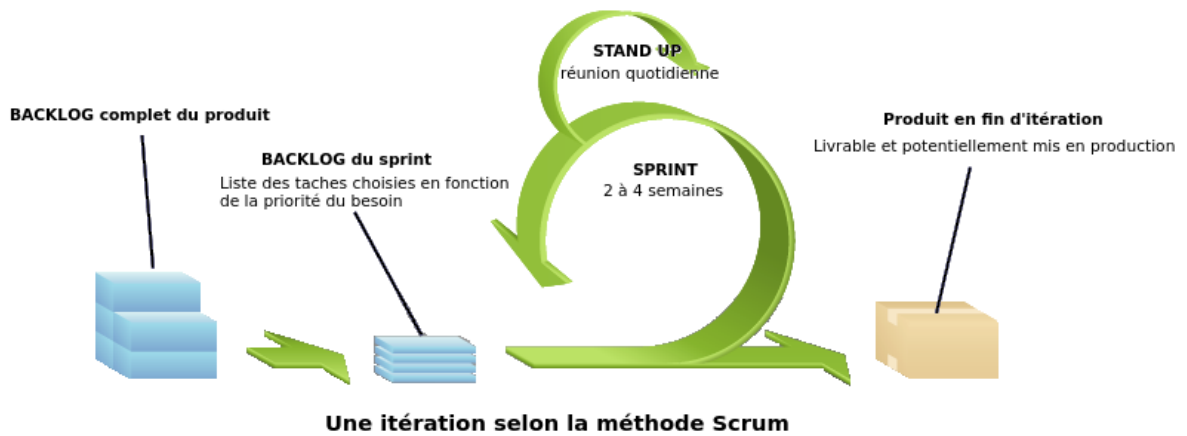


Figure 3: Une itération selon la méthode Scrum

## 2. PLANIFICATION

Ci-dessous est présenté le planning, qui prend en compte les priorités de développement et permet de suivre un chemin bien tracé lors du développement.

Il permet également de ne pas se perdre en cours de route et de gérer efficacement le temps. Ce planning a été revu et amélioré tout au long du processus de réalisation, pour prendre en compte les changements de priorité ou les difficultés rencontrées lors du développement.

Nom de la tâche	Date de début	Date de fin	Durée
<b>Etude préalable</b>	<b>08/04/16</b>	<b>21/04/16</b>	<b>10j</b>
Familiarisation avec le sujet	08/04/16	15/04/16	6j
Étude fonctionnelle	14/04/16	18/04/16	3j
Choix du processus de développement	19/04/16	21/04/16	3j
<b>Choix et Etude de la plate-forme</b>	<b>21/04/16</b>	<b>06/05/16</b>	<b>12j</b>
Choix de la plate-forme	21/04/16	28/04/16	6j
Etude technique de la plate-forme Elgg	29/04/16	06/05/16	6j
<b>Adaptation de la plate-forme Elgg</b>	<b>06/05/16</b>	<b>06/06/16</b>	<b>22j</b>
Plugin Gestion des groupes	06/05/16	16/05/16	7j
Plugin du thème	17/05/16	24/05/16	6j
Plugin d'Agenda	25/05/16	27/05/16	3j
Plugin de gestion des étudiants	28/05/16	06/06/16	7j

Figure 4: Planning de travail détaillant l'aspect gestion du temps du projet

### 3. ÉTUDE FONCTIONNELLE

Pour répondre aux besoins spécifiques de le FST, le tableau suivant présente les principales questions à se poser avant de se lancer dans ce type de projet :

Typologie des utilisateurs	Qui seront les utilisateurs ?	Étudiants, professeurs, fonctionnaires, ...
Type de réseau	Un ou plusieurs espaces d'échanges pour tous les utilisateurs ?	Chaque utilisateur sera ajouté aux groupes qui lui concernent
Type de contenus	Que va-t-on gérer comme contenus, que va-t-on échanger ?	Des documents, des informations, ...
Accès au réseau	Ouvert ou fermé ? Qui peut s'inscrire (et comment) ? Quel contrôle mettre en œuvre (ou non) à l'entrée ? Si le réseau comporte différentes « zones », doivent-elles être visibles ou masquées ?	Fermé. Seulement les membres de la FSTF. Vérifier si le demandeur d'inscription au RS est membre de FST. L'utilisateur ne sera ajouté qu'aux groupes qui lui concernent, et ne peut pas voir les autres groupes (masquées).
Confidentialité des données personnelles	Quelles informations concernant les utilisateurs doit-on gérer (déterminer les contenus donnés personnelles des « profils ») ? Qu'est-ce qui est visible par les autres utilisateurs ?  Quel contrôle l'utilisateur doit-il avoir sur ses données personnelles et sa visibilité dans le réseau ?	Nom, prénom, email, profession (Étudiant, professeur, ...). Pour :  Etudiant : CNE, niveau d'étude (LMD, DUT, ING), année.  Professeur : ID, département.  Autres : ID, nom de profession.  Nom, Prénom sont visibles à tous les utilisateurs.  L'utilisateur a le droit d'afficher et de masquer ses autres informations.
Visibilité des participants	Qui est visible, par qui et comment ?	Tous les membres sont visibles par recherche.

Fonctions relationnelles	<p>Qui peut entrer en relation avec qui ?</p> <p>Qui peut voir la liste des personnes inscrites (toute la liste, une partie, avec un moteur de recherche, etc.) ?</p> <p>Doit-on voir les listes des contacts d'un utilisateur ?</p>	<p>N'importe quel utilisateur peut ajouter d'autres comme des contacts.</p> <p>Tous les utilisateurs sont visibles pour le moteur de recherche.</p> <p>Non, On ne peut pas voir la liste des contacts d'un utilisateur.</p>
Fonctions conversationnelles	<p>Qui peut écrire des commentaires, pour quels contenus –idées, sondages, petites annonces, questions, etc. – et de quelle manière ?</p>	<p>Les membres d'un groupe peuvent écrire des commentaires sur toutes les publications de ce groupe, et l'utilisateur peut écrire des commentaires sur les publications des membres qui sont en relations avec lui.</p>
Messagerie	<p>A-t-on besoin d'une messagerie interne ? Qui peut écrire à qui ? Pour quels usages ?</p>	<p>Oui, Un membre aux utilisateurs qui lui sont visibles.</p> <p>Pour questionner, s'informer ...</p>
Notifications et alertes	<p>Comment les utilisateurs doivent-ils être alertés de l'activité du réseau social (tableau de bord, envoi de notification par courrier ou SMS, etc.) ? Quel contrôle auront-ils sur ces alertes ?</p>	<p>Tableau de bord et notification.</p> <p>Voir les alertes comme les notifications.</p>
Partage documentaire	<p>Quels types de contenus devons-nous</p> <p>gérer, comment, avec quels accès et pour qui ?</p>	<p>On gèrera les documents (PDF, doc,), des vidéos, des images. L'utilisateur peut télécharger ces documents.</p> <p>Accès privé pour les utilisateurs concernés.</p>
Édition de contenus	<p>Comment devra être faite la gestion des contenus ? Par qui ? Avec quel contrôle, quels outils et quels niveaux d'accès ?</p>	<p>La gestion de contenu est donnée aux utilisateurs. Ils peuvent créer, modifier, supprimer du contenu, même contrôler la visibilité de ses</p>

		publications.
Classement des ressources	Comment doit-on organiser les contenus à gérer ? Quel mode d'indexation ? Quel moteur de recherche ?	Une indexation de contenu selon les Tags.
Coproduction et collaboration	A-t-on besoin d'éditer et/ou annoter des contenus, comment, avec qui ? Doit-on gérer des circuits de validation/diffusion ? A-t-on besoin de gérer des groupes d'utilisateurs ?	Non, le contenu peut être discuté avec des commentaires mais ne peut pas être édité.  Non.  Oui, on a besoin de gérés les groupes selon les sections, TD,...
Gestion d'événements	Doit-on gérer un ou plusieurs agendas, des événements, des inscriptions à ceux-ci ?	Oui, il y aura un agenda général du site géré par l'administrateur.
Administration	De quels outils d'administration a-t-on besoin ? Qui gère quoi et comment ? Y a-t-il un seul ou plusieurs administrateurs ?	Il y aura un seul administrateur, qui a rôle de valider les inscriptions des nouveaux membres et de gérer les rapports.
Intégration et maintenance	Comment doit-on intégrer cet outil au sein du système d'information de l'entreprise ? Doit-il être interopérable avec les outils déjà existants dans l'entreprise ? A-t-on les compétences techniques pour installer et maintenir ce nouvel outil en interne ? Doit-on externaliser ?	La FST doit fournir chaque année les listes des nouveaux et anciens étudiants afin de les ajouter au RS.  Oui.  Ce RS demande un serveur LAMP.

### Conclusion

Dans cette partie, nous avons identifié les objectifs et les besoins fonctionnelles spécifiques qui déterminent la plate-forme à choisir. Ainsi, les modifications et les adaptations à implémenter.



**CHAPITRE 2 :**  
CHOIX ET ÉTUDE DE LA PLATE-FORME

## 1. ÉTUDE COMPARATIVE DES PLATES-FORMES SMS

Pour répondre à la question quelle plate-forme choisir pour la communauté virtuelle. Il faudrait voir quelles sont les SMS sociaux les plus utilisés, les outils de communication dont ils disposent et juger leur pertinence à la lumière des outils les plus indispensables pour faire vivre une communauté virtuelle d'étudiants et d'enseignants.

Il existe un nombre très impressionnant de solutions de réseaux sociaux open source, nous avons sélectionné les meilleures solutions open source (BuddyPress, Drupal, Elgg et Liferay) pour couvrir notre besoin de réseau social, car ce sont les plates-formes les plus utiles et offrent plusieurs fonctionnalités.

### 1. BUDDYPRESS

BuddyPress a été conçu en 2008 par Andy Peatling alors qu'il travaillait à l'ajout de fonctionnalités de networking sur le célèbre moteur de blog WordPress. BuddyPress est livré par défaut avec de nombreuses fonctionnalités qu'il est très simple d'activer. Citons par exemple : flux d'activité, profil, groupes, gestion des amis, messagerie, blogs, forums, etc. On note quelques points très agréables sous BuddyPress :



- la flexibilité de la politique de sécurité des attributs (nom, prénom, etc.) d'un membre (le membre peut définir les données – champ par champ – qu'il souhaite afficher et à qui).
- la possibilité de géolocaliser les membres du réseau ou de calculer les distances qui les séparent (à l'aide de modules complémentaires).
- des outils de gestion de projets (centre des tâches, suivi de projet, etc.) à l'aide de l'extension BP GTP System.

### 2. DRUPAL

Drupal est un CMS aux multiples facettes. Conçu à l'origine pour être un blog collectif, il trouve aujourd'hui des applications très variées : du site corporatif au portail communautaire, il sait tout faire ! Il a été conçu dans les années 2000 par Dries Buytaert et connaît depuis un succès mondial. Une communauté énorme s'est créée autour du produit. Drupal est l'outil idéal pour coupler RSE et CMS, On note quelques points très agréables sous Drupal :



- la flexibilité globale du système de mise en page et de templating qui offre une véritable souplesse aux webmasters.
- un interfaçage réussi avec le moteur de recherche open source Solr (plugin) qui permet d'envisager une très bonne exploitation des expertises.

- le module « userpoints » qui offre aux « community manager » de véritables outils d'animation du réseau, une couverture globale excellente avec très peu de fonctionnalités manquantes.

### 3. LIFERAY

Liferay Social Office est un produit de la société Liferay, déjà éditrice du portail Liferay. Social Office est une solution de collaboration sociale pour l'entreprise.



Le portail Liferay a vu le jour au début des années 2000, lancé par Brian Chan. 2008, Liferay a lancé une déclinaison de son portail avec un packaging spécifique au travail collaboratif : Liferay Social Office.

Liferay est très orienté Entreprise avec de nombreuses fonctionnalités Projet et peu d'éléments conversationnels. On peut tout de même souligner la qualité du Chat embarqué sous Liferay. On note quelques points particulièrement soignés sous Liferay, listons notamment :

- une page profil et un tableau de bord entièrement personnalisables, plus efficaces encore que ceux offerts par la solution Elgg.
- un Chat complètement embarqué en lien avec la liste de ses amis pour plus d'interactivité.
- une vraie dimension portail complètement adaptée à un existant avec de nombreuses interfaces possibles avec le SI de l'entreprise.
- une vraie dimension projet avec des outils de wiki, groupe, centre de tâches, client email, notification par SMS, etc.

### 4. ELGG

Elgg a été créée en 2004 par Dave Tosh et Ben Werdmuller. Ces derniers ont, par la suite, créé la société Curverider Limited pour maintenir et faire évoluer le produit. Elgg propose un périmètre fonctionnel très riche et très complet.



On note quelques points particulièrement adaptés aux RSE avec Elgg, signalons notamment:

- une page profil et un tableau de bord entièrement personnalisables à l'aide de dizaines de widgets, avec une vraie prise en compte de la sécurité.
- une communauté mobilisée proposant des centaines de plugins permettant de dépasser le périmètre initial de la solution.
- de nombreux outils favorisant les échanges de type conversationnel.

## Conclusion

Nous avons opté pour la plate-forme Elgg qui est la mieux adaptée au besoin du projet. De plus, Elgg est une des solutions leaders en matière de RSE open source. Elle propose un périmètre fonctionnel très riche et très complet. Elle est notamment la seule des solutions étudiées avec Liferay, à proposer une interface personnalisée aux membres du réseau.

## 2. ÉTUDE DE LA PLATE-FORME ELGG

Elgg est écrit en PHP et fonctionne sous technologies LAMP. Son architecture autorise la notion de plugins et permet d'étendre facilement et rapidement ses fonctionnalités.

### 1. LES FONCTIONNALITES

Elgg propose en standard toutes les fonctionnalités de base d'un réseau social : gestion de profils et de contacts, messagerie, blog, microblogging, etc. Chacune de ces fonctionnalités pouvant être activée ou non suivant les usages auxquels est destiné le site motorisé par Elgg.

### 2. LE MOTEUR D'ELGG

Sur le plan technique, le moteur Elgg est bâti sur une architecture rigoureusement « objet » et « événements ». Chaque événement peut être utilisé par un plugin indépendant du noyau du moteur lui-même, pour générer ses propres actions.

Dans le même esprit, le noyau offre une multitude d' « ancrés » auxquelles les plugins indépendants peuvent s'accrocher pour intégrer leur propre code, qui devient donc virtuellement une partie du noyau lui-même.

Du côté de la base de données, Elgg propose une architecture unifiée autour d'une logique « entités » dans lesquelles les plugins externes peuvent intégrer leurs propres objets et/ou agir sur des objets existants.

D'une manière maintenant classique, Elgg sépare le code de l'affichage et propose une structure basée sur des thèmes.

Cependant, là aussi, Elgg innove et propose une architecture très souple, fondée sur des « vues » qu'il suffit de sélectionner au moment de la connexion pour offrir un mode d'affichage adapté au matériel avec lequel on se connecte.

De plus, chaque « vue » d'un module peut être remplacée par la vue d'un autre module, et ce sans avoir à modifier la vue originale. Cela permet d'adapter finement l'affichage des objets et l'organisation des pages d'un site développé sur Elgg.

Grâce à ces différentes caractéristiques, l'architecture de Elgg offre aux développeurs des accès très souples pour adapter le comportement du moteur à chaque projet, sans jamais avoir besoin de modifier le noyau.

### 3. LE MODELE - VUE - CONTROLEUR D'ELGG

Le patron d'architecture logicielle modèle-vue-contrôleur (en abrégé MVC, en anglais model-view-controller), tout comme les patrons modèle-vue-présentation ou présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

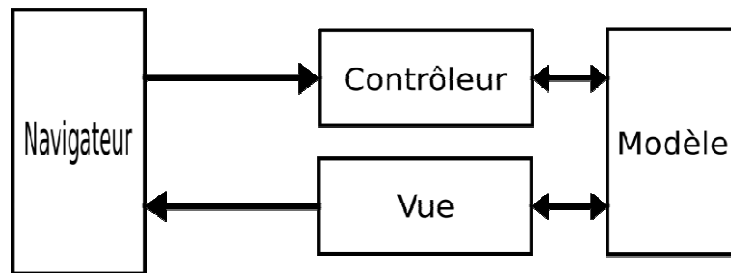


Figure 5: Architecture Modèle - vue - contrôleur

#### 1. Contrôleur

Elgg utilise le patron *Front Controller* avec une torsion. Dans le patron *Front Controller*, chaque demande est acheminée via un seul contrôleur. C'est différent des applications Web qui utilisent des scripts individuels à gérer différents types de demandes. La torsion avec la mise en œuvre de Elgg de ce patron est qu'il a deux contrôleurs avant primaire. Un contrôleur frontal gère les actions, qui impliquent généralement la soumission d'un formulaire. L'autre contrôleur frontal gère les demandes de pages, qui sont le plus souvent des pages HTML. Dans Elgg, ces contrôleurs sont appelés gestionnaires (*handlers*).

Le travail du contrôleur frontal charge le plate-forme et envoie la demande au contrôleur secondaire approprié. Un contrôleur de page traite les paramètres de la demande, met à jour le modèle de données, et transforme la demande traitée vers le système de vues. Un contrôleur d'action est similaire, sauf qu'il ne tourne pas une demande sur le système de vue, mais transmet plutôt le demandeur vers une nouvelle page.

#### 2. Modèle

Le modèle d'Elgg se compose de trois parties:

1. les classes de modèle de données qui gèrent les entités, les méta-données et les relations.
2. Les Fonctions d'assistances (*Helper*) qui construisent des requêtes basées sur des paramètres.
3. La Couche d'accès de base de données qui fournit une interface à la base de données MySQL.

Le modèle de données prend en charge un large éventail de cas d'utilisation en raison de sa conception simple et flexible. Les développeurs doivent utiliser le modèle de données de Elgg plutôt que d'écrire leurs propres requêtes ou de créer leurs propres tables.

### 3. Vue

Le système de vue rend les données en HTML. Il comporte deux étapes. Dans la première étape, le contrôleur demande au système d'affichage de rendre les données à partir du modèle de présentation. Dans la deuxième étape, cette sortie est insérée dans une template et la réponse complète est rendue. Ceci est différent des plates-formes qui utilisent une template par page qui définit une page entière. Un avantage de l'approche en deux étapes est la facilité de maintenir un aspect cohérent sur toutes les pages.

La sortie de la première étape est représentée par le bloc de contenu dans le schéma suivant. La deuxième étape traite les sections restantes d'une page.

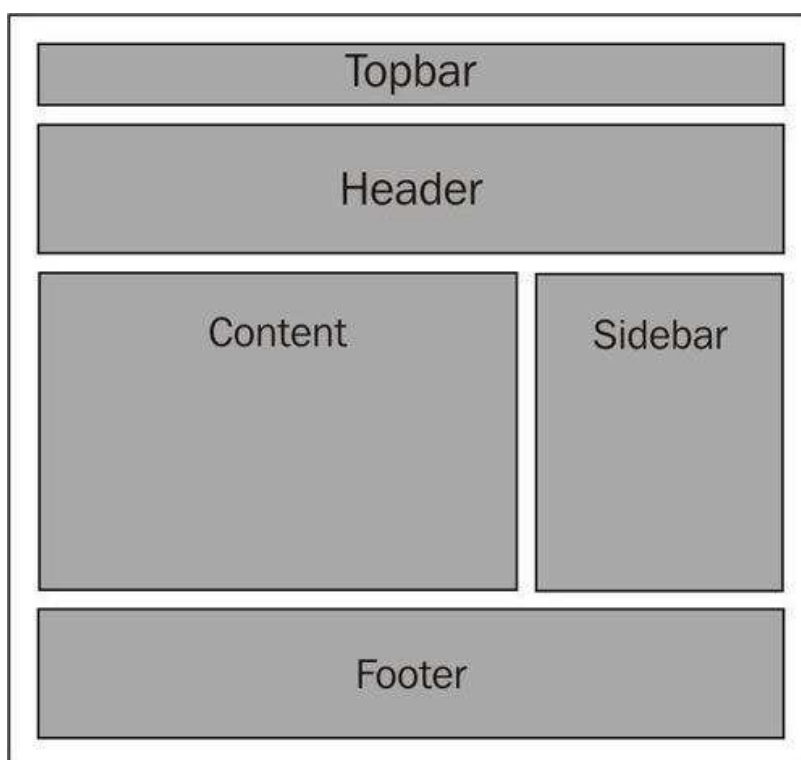


Figure 6: Les blocs de vue d'Elgg

Les deux étapes utilisent des modèles qui sont appelés views à Elgg. Les vues sont des scripts courts qui fournissent des blocs de construction pour la création de pages Web et d'autres types de sortie. Dans le schéma précédent, la barre du haut, en-tête, la barre latérale, et pied de page sont créés par des vues.

Chaque demande ayant un type de vue qui détermine le format de réponse. Si le type d'affichage est par défaut, une page HTML est créée. Le type de vue est définie par le régulateur en fonction des paramètres de la demande.

## 4. LES PLUGINS

Le moteur d'Elgg fournit les fonctionnalités de bases et les plugins déterminent ce que l'application Web fait vraiment. Elgg est distribué avec plus de 30 plugins. des plugins additionnels sont disponibles sur le site de la communauté.

Les plugins fournissent plusieurs moyens pour ajouter ou modifier les fonctionnalités proposées par Elgg. Ils peuvent aussi correspondre à une fonctionnalité minimale, ou à une fonctionnalité complète comme un moteur de blog ou une galerie de photos.

Les Plugins sont installés dans le répertoire `/mod/`. Chaque répertoire dans `/mod/` est un plugin. Les plugins sont structurés comme des mini versions du noyau Elgg et peut contenir les dossiers et les fichiers suivants :

*actions, classes, graphiques, langues, pages, vendors, views, start.php et manifest.xml*

Le fichier *manifest.xml* décrit la fonction du plugin et cette information est utilisée sur la page d'administration du plugin. Le fichier *start.php* est le script de démarrage du plugin. il enregistre les fonctions de rappel, étend des vues, et effectue d'autres initialisation.

## 5. MODELE DE DONNEES

En Elgg, tout fonctionne sur un modèle de données unifié basé sur des unités atomiques de données appelées entités.

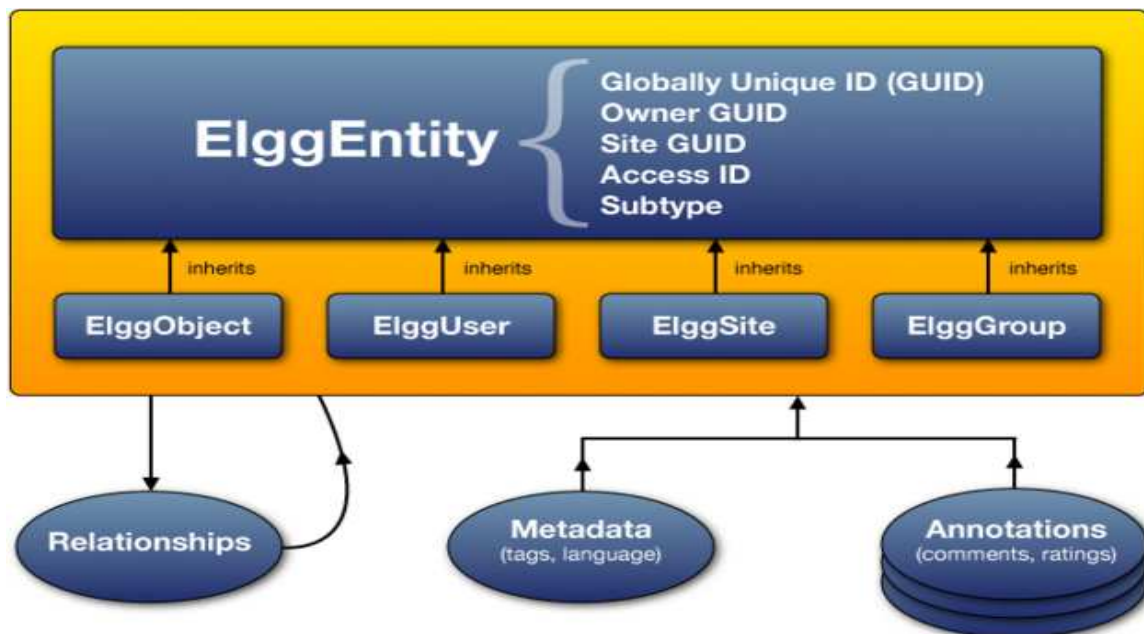


Figure 7: Modèle de données d'Elgg

Chaque entité du système hérite de la classe *ElggEntity*. Cette classe contrôle les droits d'accès, et les propriétaires d'une entité.

Nous pouvons étendre des entités avec des informations supplémentaires de deux manières:

*Metadata* : Cette information décrivant l'entité, généralement ajoutée par l'auteur de l'entité lorsque l'entité est créée. Par exemple, les étiquettes, un numéro ISBN, un emplacement de fichier, ou de la langue source est un metadata.

*Annotations* : Ce sont des informations sur l'entité, généralement ajoutée par un tiers après que l'entité est créée. Par exemple, notes, goûts, et les votes sont annotations.

*ElggEntity* est la classe parente de toutes les entités. Il est une classe abstraite héritée par quatre classes: *ElggGroup*, *ElggObject*, *ElggSite* et *ElggUser*.

Chacun de ces *ElggEntity* peut être sous-typé. Par exemple, nous pouvons affirmer que l'objet est un sous-type de blog, définit comme un billet de blog.

Chaque *ElggEntity* a un *GUID* - un numéro d'identification global unique - plutôt que d'un ID standard ou identifiant.

Les relations (*Relationships*) vous permettent de lier des entités ensemble. Exemples: un membre a des contacts, un utilisateur est un membre d'une organisation, etc.

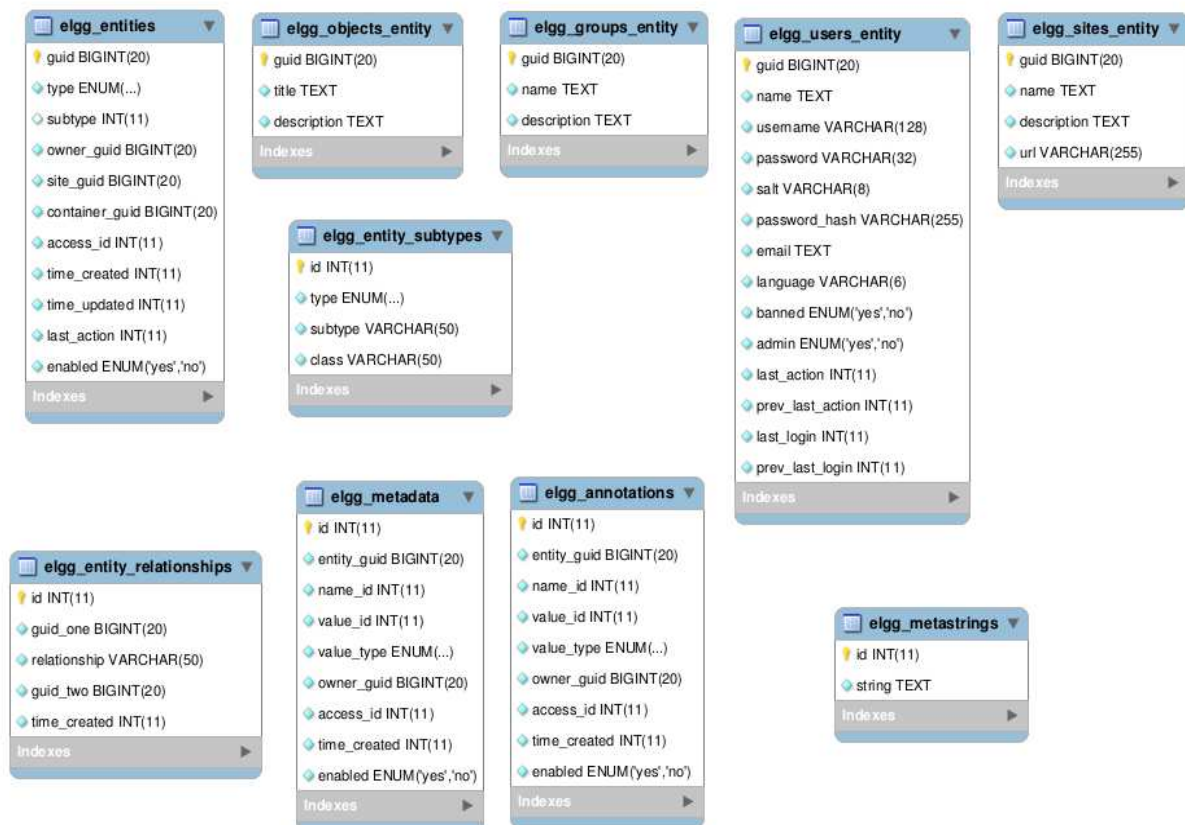


Figure 8: Extrait des tables de base de données d'Elgg



## Conclusion

Dans ce chapitre, nous avons choisi Elgg parmi plusieurs plates-formes open source pour l'implémentation du réseau social. De plus nous avons fait une étude technique détaillée de son architecture MVC et son modèle de données. Et comment étendre ses fonctionnalités de base avec les plugins.

# CHAPITRE 3 : ANALYSE ET CONCEPTION

Ce chapitre comporte une description de la phase de spécification et d'analyse de la plate-forme Elgg. A la lumière de l'étude fonctionnelle, nous entamerons une étude conceptuelle de la solution proposée.

## 1. LANGAGE DE MODELISATION

Nous avons choisi de travailler avec UML comme langage de modélisation. UML ou Langage de Modélisation Unifié, est un langage de modélisation graphique à base de diagrammes. Il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML est couramment utilisé dans les projets logiciels.

## 2. ANALYSE DES BESOINS

### 1. IDENTIFICATION DES ACTEURS

D'après une étude de la plate-forme, nous avons identifié deux acteurs :

- L'utilisateur, c'est le rôle de la personne qui utilise le système. Nous avons spécifié ce dernier en deux acteurs: étudiant et enseignant.
- L'administrateur, c'est le rôle de la personne responsable de la maintenance de l'application.

### 2. IDENTIFICATION DES CAS D'UTILISATION

Un cas d'utilisation (en anglais use case) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre mais UML propose un formalisme et des concepts issus de bonnes pratiques. Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant un résultat sur un acteur, et ceci indépendamment de son fonctionnement interne.

Après l'identification des acteurs, nous avons élaboré les digrammes des cas d'utilisations disponibles dans la plate-forme. Ces diagrammes non exhaustifs présentent les cas essentiels pour notre application et ceux que nous allons modifier.

Les figures suivantes montrent les diagrammes respectifs de l'administrateur, l'utilisateur et enseignant :

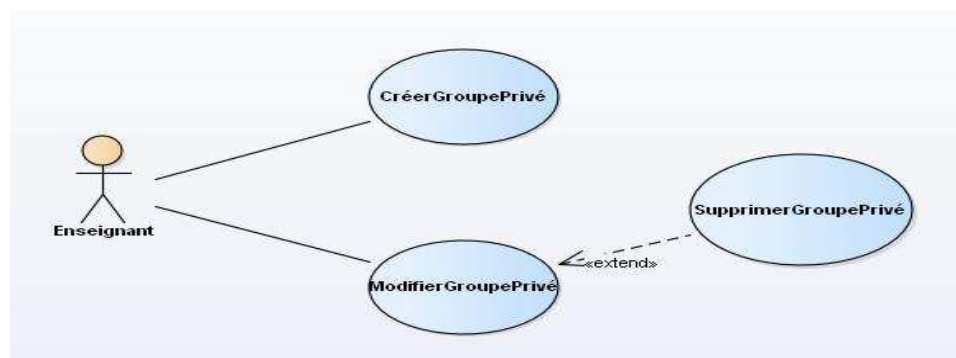


Figure 9: Diagramme de cas d'utilisation de l'enseignant



### 3. DIAGRAMME DE PACKAGE

C'est un moyen pour regrouper les différents éléments de la modélisation. Il permet de représenter les relations entre les différents profils de l'application. Il rassemble les cas d'utilisations propres à chaque acteur de façon cohérente.

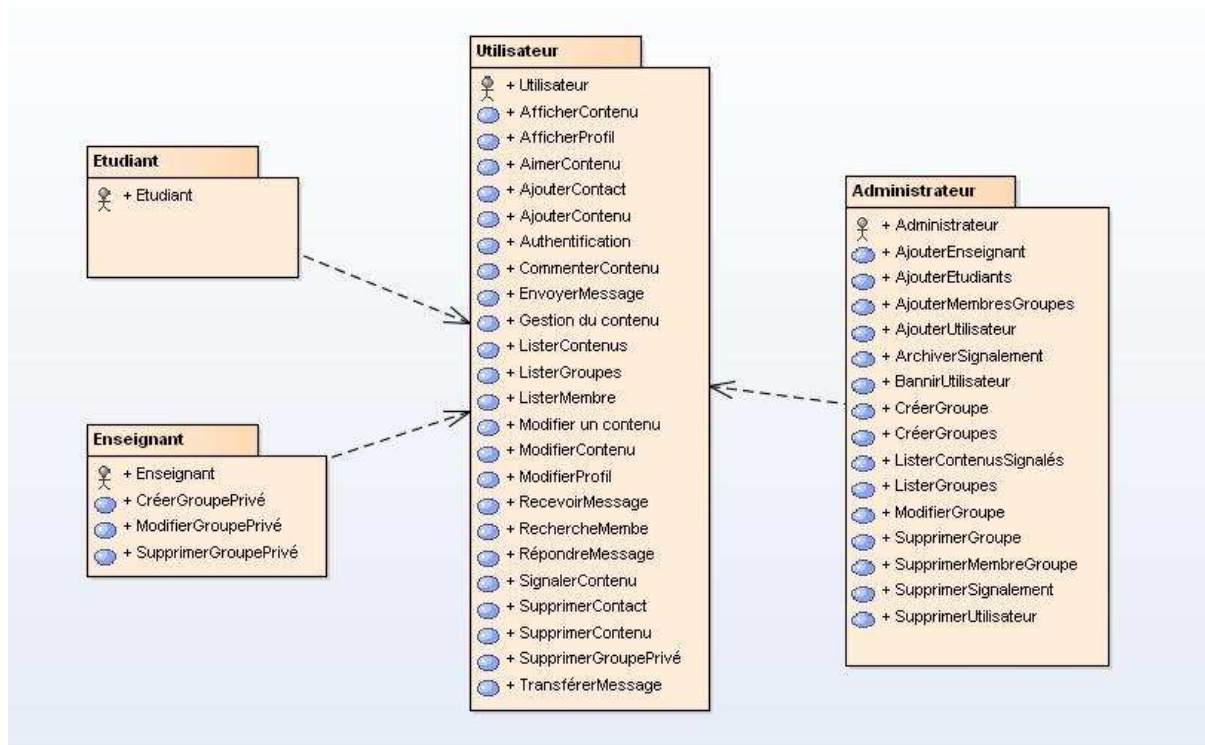


Figure 12: Diagramme de package

### 4. DIAGRAMMES DE SEQUENCES

Le diagramme de séquence permet d'illustrer les cas d'utilisation et de représenter le séquençage des interactions entre les objets du système dans le temps.

Nous présentons les diagrammes des cas d'utilisation que nous avons ajouté ou modifié à la plate-forme << créer un groupe >> et << ajouter des étudiants >>.

## 1. Création d'un groupe

Pour créer un groupe, il suffit de choisir le type de groupe (cycle, section ...), et après lui donner un nom (DEUST, MIP, ...). Il est obligatoire de respecter la hiérarchie de création des groupes.

Par exemple la création du groupe MIP doit être précédée par la création du groupe DEUST.

Le diagramme suivant explique le scénario de la création d'un groupe :

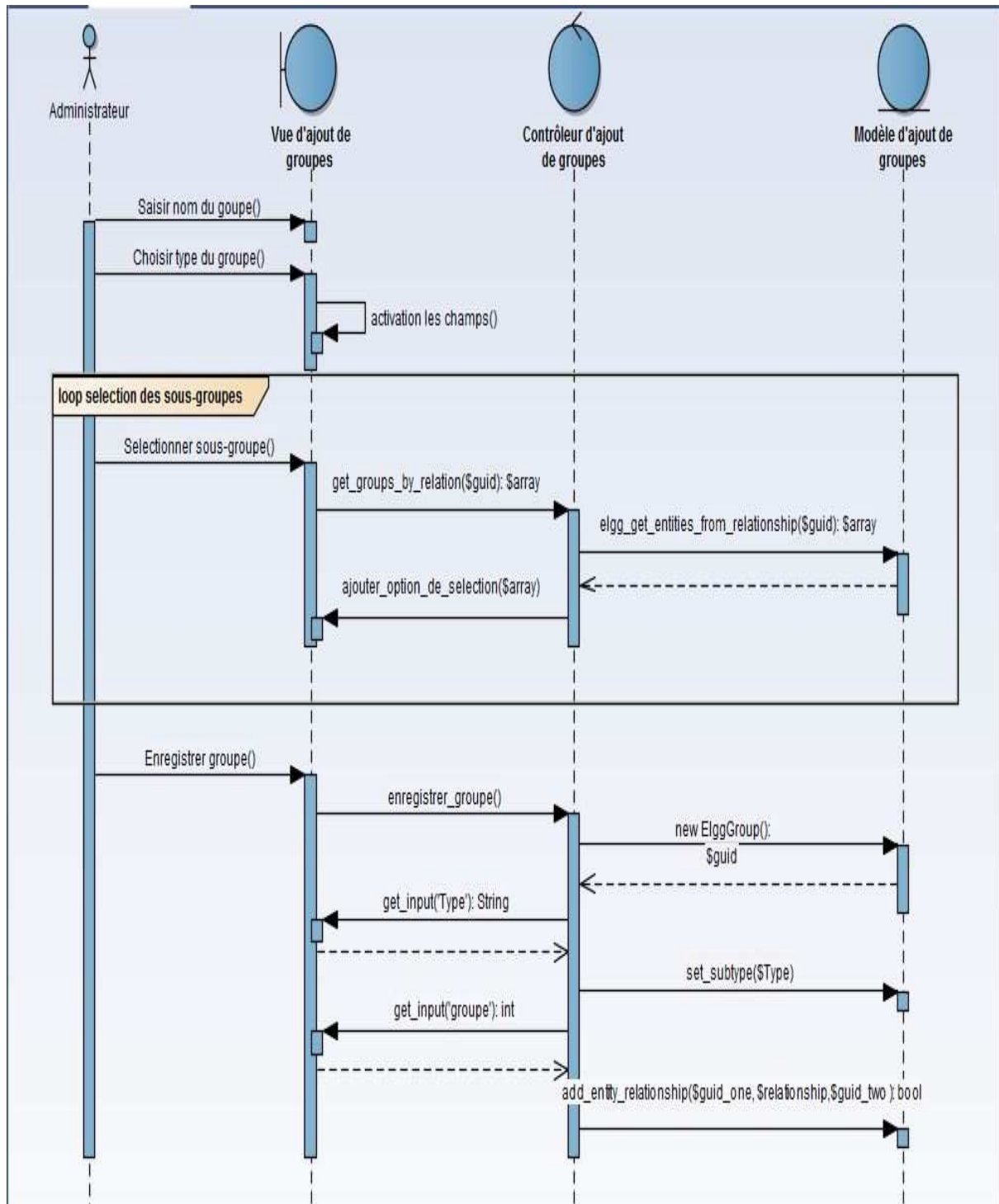


Figure 13: Diagramme de séquence de création d'un groupe

## 2. L'ajout des étudiants

Les étudiants sont ajoutés par l'administrateur dans les groupes qui leurs concernent selon le diagramme de séquence suivant :

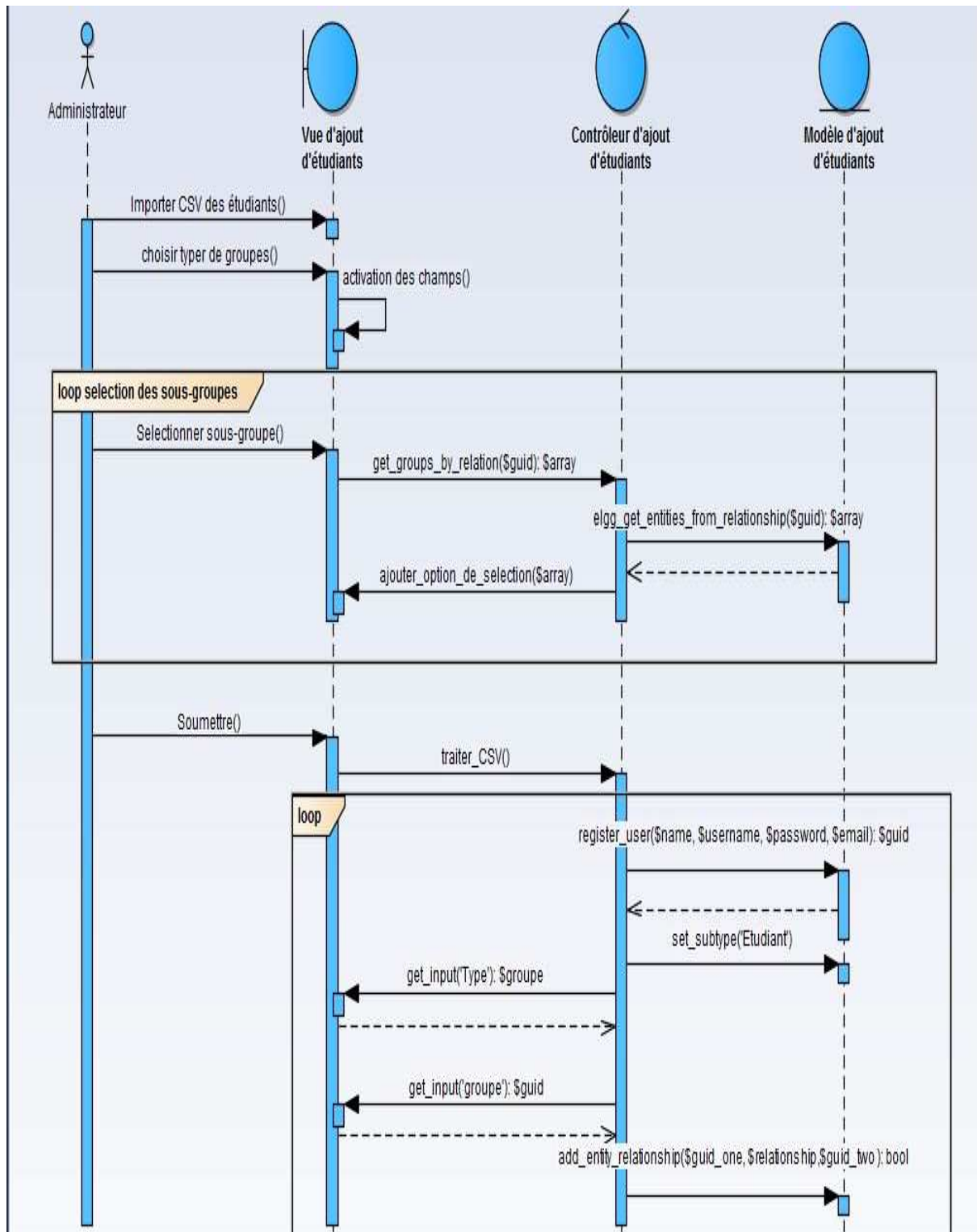


Figure 14: Diagramme de séquence d'ajout des étudiants

## 5. DIGRAMME DE CLASSES

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations.

Après une ingénierie inverse du code de la plate-forme, nous avons eu un digramme de classes avec 141 classes.

La figure suivante représente un extrait des classes intervenant dans le système.

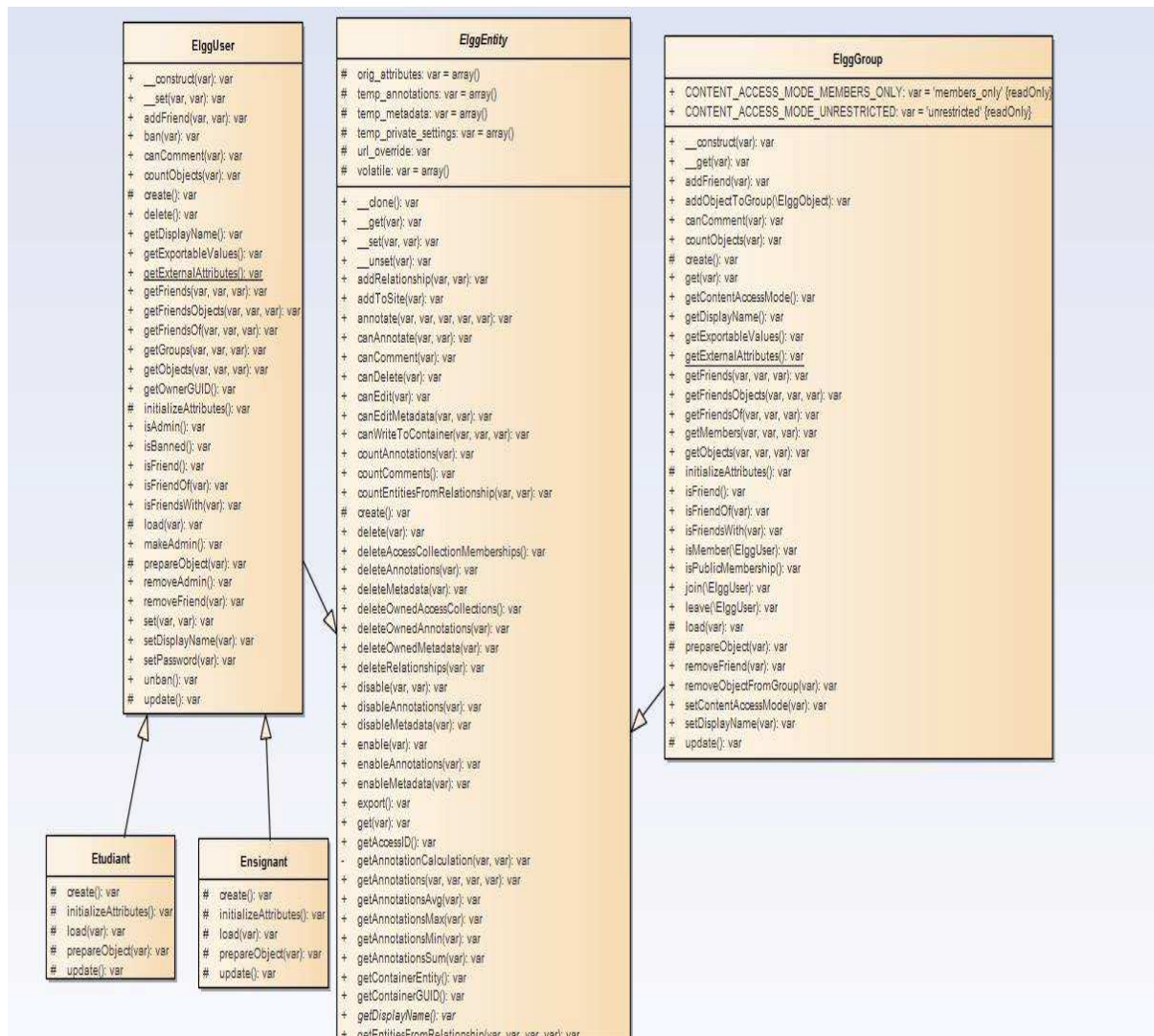


Figure 15: Extrait de diagramme de classes



## 6. DIAGRAMME DE DEPLOIEMENT

Le diagramme de déploiement est description de la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.

Le diagramme de déploiement suivant, expose les exigences de la plate-forme Elgg :

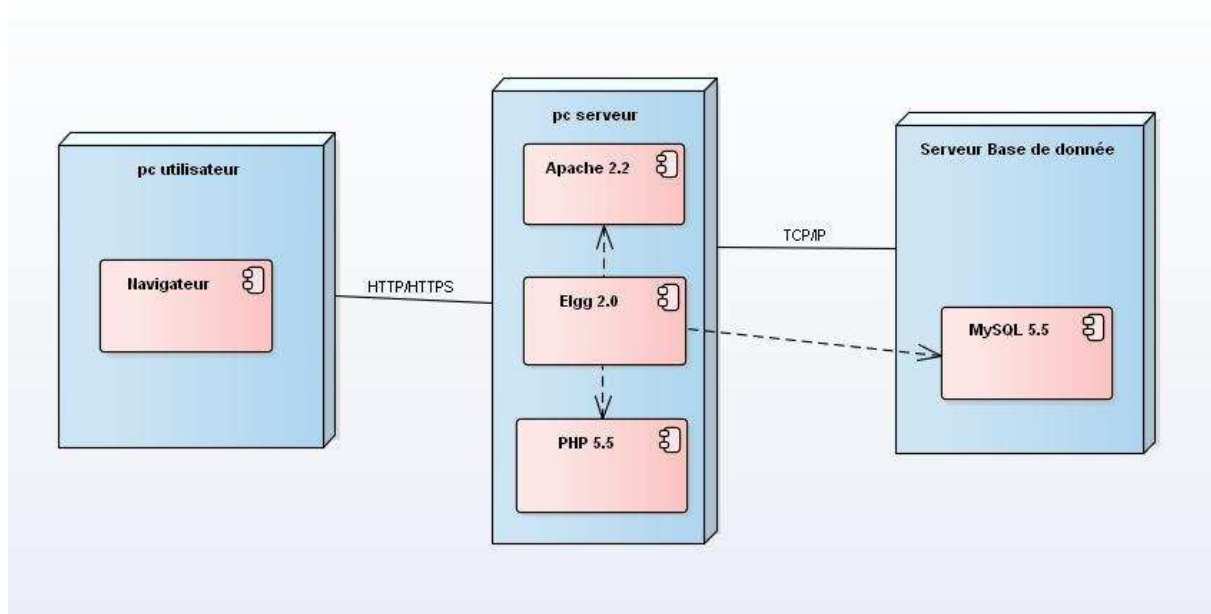


Figure 16 : Diagramme de déploiement d'Elgg

## CHAPITRE 4 : ADAPTATION DE LA PLATE-FORME ELGG

Dans ce chapitre nous exposons le travail réalisé afin d'adapter la plate-forme Elgg aux besoins spécifiés dans l'étude fonctionnelle.

Pour ajouter des fonctionnalités, nous étions amenés à développer des plugins :

- Plugin gestion de groupes
- Plugin gestion des étudiants
- Plugin du thème

Et installer d'autres :

- Plugin d'agenda

## 1. LES OUTILS DE DEVELOPPEMENT



**PHP** : Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.



**Bootstrap** est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



**jQuery** est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.



**Ajax** (Asynchronous JavaScript and XML) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.



**JavaScript** est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs.



**Cloud9 IDE** est un environnement de développement intégré en ligne. Il permet aux développeurs de commencer le codage immédiatement avec des espaces de travail de préconfiguré, de collaborer avec leurs pairs avec des fonctionnalités de collaboration. De plus, les caractéristiques de développement web comme la prévisualisation en direct et les tests de compatibilité du navigateur.

## 2. LES PLUGINS DEVELOPPES

### 1. PLUGIN GESTION DES GROUPES

Ce plugin modifie la façon de création des groupes dans Elgg. Il crée une hiérarchie des groupes en ajoutant une relation entre les entités *ElggGroup* "sous-groupes".

Par exemple : le groupe de Licence Génie Informatique est un sous-groupe de Licence.

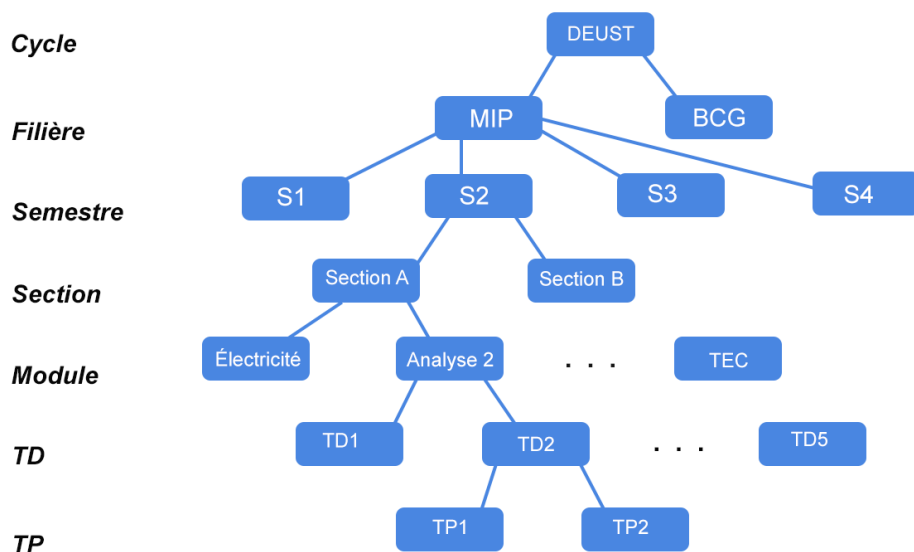


Figure 17: Hiérarchie de création des groupes

Nous avons ajouté des sous-types à l'entité *ElggGroup* lors de l'activation du plugin, pour différencier entre les groupes.

La figure suivante présente l'interface de création d'un groupe :

Le formulaire est intitulé "Gestion des groupes : Ajouter un groupe". Il contient les champs suivants :

- Nom du groupe :
- Type de groupe :
- Cycle :
- Filière :
- Semestre :
- Section :
- Module :
- TD :

Un bouton "Enregistrer" est situé en bas à gauche.

Figure 18: Formulaire de création d'un groupe

id	type	subtype	class
9	group	Cycle	
10	group	Filière	
11	group	Semestre	
12	group	Module	
13	group	TD	
14	group	TP	
15	group	Générale	
16	group	Section	

Figure 19: Table Elgg\_entity\_subtype

## 2. PLUGIN GESTION DES ETUDIANTS

Dans ce plugin, nous gérons l'inscription des étudiants. Après avoir un fichier CSV d'étudiants qui contient leurs informations (CNE, nom, prénom, email, date de naissance), l'administrateur doit l'importer et choisir le groupe ou vont être ajoutés.

**Utilisateurs : Ajouter des étudiants**

**Importer CSV des étudiants**

No file chosen

**Type de groupe**

Cycle ▾

**Cycle**

Master ▾

**Filière**

▾

**Semestre**

▾

**Section**

▾

**Module**

▾

**TD**

▾

Figure 20: Formulaire d'ajout des étudiants

### 3. PLUGIN DU THEME

Dans ce plugin, nous effectuons des changements dans le thème par défaut d'Elgg. Ces changements concernent :

- La page d'accueil.
- Le logo.
- Le style du bar de navigation.

Afin de personnaliser le thème, nous remplaçons les vues du noyau d'Elgg avec de nouvelles vues dans le plugin.



Figure 21: La page principale du réseau social



Figure 22: L'interface d'authentification des utilisateurs

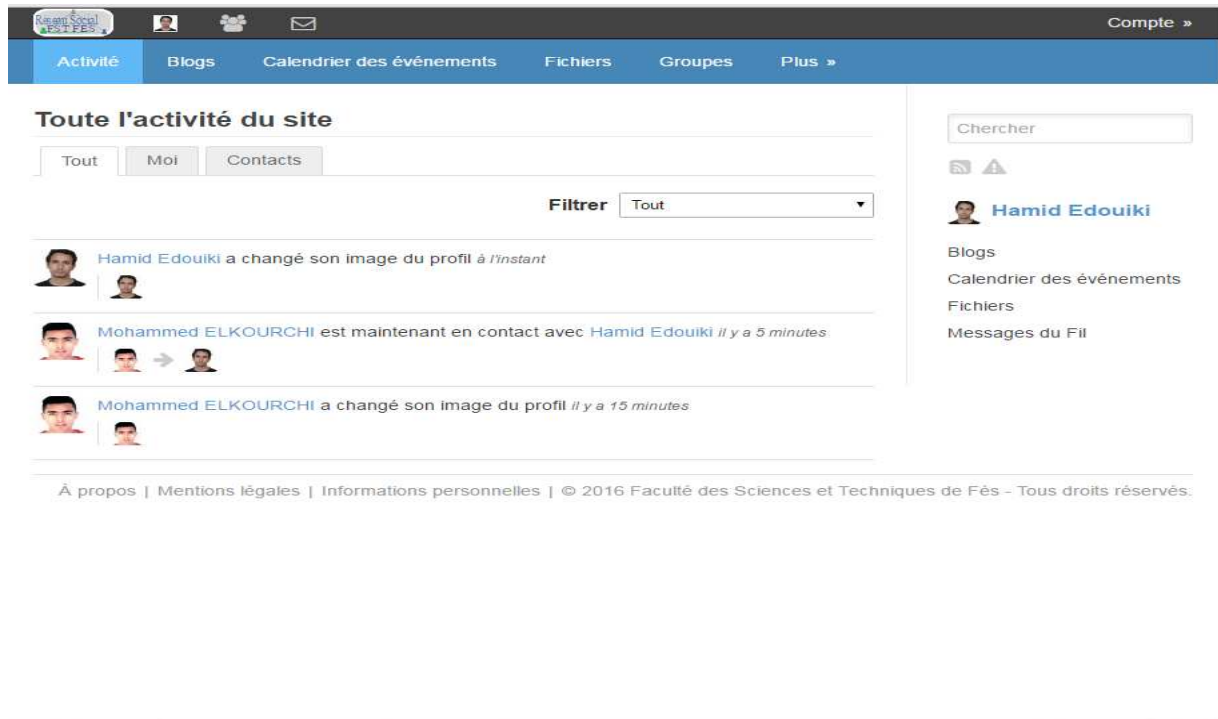


Figure 23: L'interface de la page d'accueil

### 3. LES PLUGINS INSTALLES

#### 1. PLUGIN D'AGENDA

Pour gérer les événements, nous avons installé un plugin d'agenda qui est développé par la communauté d'Elgg.

Notre travail a consisté à traduire le plugin en Français. Dans Elgg, les fichiers de traductions se trouvent dans le dossier `/languages` du plugin. Chaque fichier correspond à un langage (par exemple `fr.php` pour la langue française), et contient un tableau de correspondance entre des clés avec des chaînes de caractères. Ces clés sont utilisées par la fonction `elgg_show()` afin d'afficher les chaînes appropriées avec le langage d'utilisateurs.

```
return array(
    'admin:administer_utilities:event_calendar' => "Calendrier des événements",
    'item:object:event_calendar' => "Calendrier des événements",
    'tag_names:event_tags' => "Tags",
    'event_calendar:optional' => "Optionnel.",
    'event_calendar:required' => "Requis.",
    'event_calendar:new_event' => "Nouveau événement",
    'event_calendar:add_event_title' => "Ajouter événement",
    'event_calendar:manage_event_title' => "Modifier événement",
    'event_calendar:title_label' => "Titre",
    'event_calendar:title_description' => "Un à quatre mots",
    'event_calendar:brief_description_label' => "Description",
    'event_calendar:brief_description_description' => "Une courte phrase.",
    'event_calendar:venue_label' => "Lieu",
    'event_calendar:venue_description' => "Où va cet événement aura lieu?",
    'event_calendar:start_date_label' => "Date de début",
    'event_calendar:start_date_description' => "Quand cet événement démarre?",
    'event_calendar:fees_label' => "Honoraires",
    'event_calendar:fees_description' => "Le coût de cet événement, le cas échéant.",
    'event_calendar:contact_label' => "Contact",
```

Figure 24: une partie du fichier fr.php (plugin Agenda)

Tous les événements

Ajouter un événement

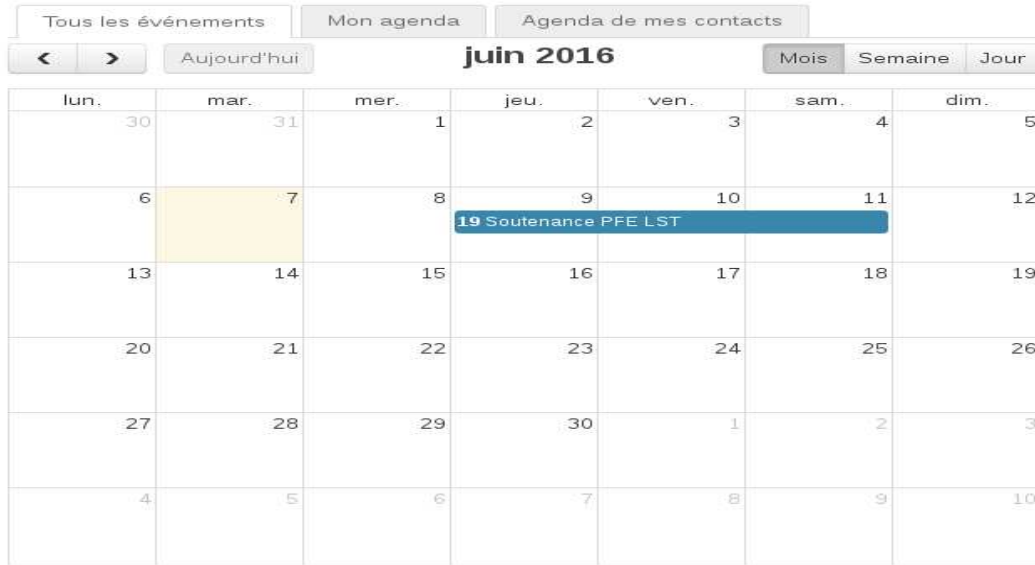


Figure 25: L'interface du plugin agenda

Un utilisateur peut créer un événement en cliquant sur Ajouter événement, et remplir les champs qui s'affichent dans l'interface suivante :

Calendrier des événements

**Ajouter événement**

**Titre**  
Soutenance PFE LST

**Lieu**  
ESTC

**Description**

**Tags**

**Calendrier** Site à l'échelle calendrier

**Annexe**

- Tous les jours de l'événement  
2016-06-17
- Régler le jour / heure

**Autorisations**

**L'accès en lecture** Membres connectés

**Honoraire**

**Contact**

**Organisateur**

**Description longue** [Éditer le HTML](#)

B I U I<sub>x</sub> S ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶

Enregistrer

Figure 26: Formulaire de création d'un événement



## CONCLUSION ET PERSPECTIVES

Au terme de la réalisation de ce rapport de projet de fin d'études qui s'intitule « Implémentation d'un réseau social pour la FST de Fès », nous avons adapté une plateforme SMS open source aux besoins de la faculté.

La réalisation de ce projet de fin d'études, nous a donné une véritable occasion pour rencontrer les difficultés et l'environnement du travail qu'un informaticien devra affronter lors de la conception et de la réalisation d'un projet. Ce projet nous a demandé beaucoup d'efforts afin de respecter les attentes.

En fait, il nous a aussi permis d'enrichir nos connaissances et notre formation théorique et pratique. Notamment, dans le domaine des plates-formes SMS.

Malgré que nous ayons pris le soin de mettre en place un réseau social fonctionnel, ce projet n'est pas encore terminé. Plusieurs améliorations peuvent être ajoutées, nous citons à titre d'exemple le traitement des réservistes et la gestion des notes.

## REFERENCES BIBLIOGRAPHIQUES

- Michel Forsé, « Définir et analyser les réseaux sociaux. Les enjeux de l'analyse structurale », Informations sociales 2008/3 (n°147), p. 10-19.
- Nicole Ellison, Annike Thierry, « Réseaux sociaux, numérique et capital social. Entretien réalisé par Thomas Stenger et Alexandre Coutant », Hermès, La Revue 2011/1 (n°59), p. 21-23.
- Grégory BECUE, « Réseaux Sociaux d'Entreprise. Panorama des outils et des grands concepts fonctionnels d'un RSE », 2010.
- Bruno Louis Séguin et al., « Des outils pour toutes les plateformes », Documentaliste-Sciences de l'Information 2010/3 (Vol. 47), p. 54-67.
- Cash Costello, « Elgg 1.8 Social Networking », PACKT Publishing, coll. « PHP, Open source », 2012.
- Nikolay Nikolov, « l'enseignement a distance - choix d'une plate-forme pour la mise en ligne d'une communauté d'étudiants et d'enseignants », Université Libre de Bourgas, 2013.
- « Mise en place d'une plate-forme de capitalisation et de partage de savoir-faire dans une communauté d'enseignants par l'outil Elgg », Master SIR, FSTF, 2012.
- « Elgg Documentation », Disponible sur : <<http://learn.elgg.org/>>, (Consulté le 03/06/2016).