



UNIVERSITÉ DE MONTPELLIER  
FACULTÉ DES SCIENCES

RAPPORT DE TER DE MASTER 1 INFORMATIQUE

---

# Réingénierie logicielle sur la plate-forme de gestion des stages

---



Khaled Ismail BELHAINE

Rabie HOUTI

Yann ESSIANE MEDJA

Jérémy PASTOR

*Tuteur :* Eric BOURREAU

*Rapporteur :* Nancy RODRIGUEZ

21 Janvier 2015 — 2 Juin 2015

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Généralités . . . . .	3
1.2	Présentation de la plate-forme . . . . .	3
1.3	Sujet et objectifs de ce TER . . . . .	5
<b>2</b>	<b>Correction des bugs existants</b>	<b>6</b>
2.1	Stages qui n'apparaissent pas dans les listings . . . . .	6
2.2	Mauvais tuteur en entreprise pour un stage . . . . .	6
2.3	Suppression des liens vers les conventions . . . . .	7
2.4	Tuteurs ne peuvent pas encadrer plus de 5 stages . . . . .	7
<b>3</b>	<b>Réingénierie logicielle (refactoring)</b>	<b>8</b>
3.1	Qu'est ce que la réingénierie logicielle ? . . . . .	8
3.1.1	Définition . . . . .	8
3.1.2	Intérêts . . . . .	8
3.1.3	Processus de mise en oeuvre . . . . .	9
3.2	Réingénierie sur la base de données . . . . .	11
3.2.1	Rétro-ingénierie sur la base de données . . . . .	11
3.2.2	Recomposition de la base de données . . . . .	15
3.2.3	Restructuration du code de la plate-forme . . . . .	20
3.3	Réingénierie sur les permissions . . . . .	23
3.4	Harmonisation de l'ajout des utilisateurs . . . . .	25
3.4.1	Import CSV des enseignants . . . . .	25
3.4.2	Import CSV des étudiants . . . . .	26
3.4.3	Ajouter un utilisateur à la fois . . . . .	26
3.5	Suppression du code mort . . . . .	28
3.6	Application d'un patron de conception : MVC . . . . .	29
3.6.1	Définition . . . . .	29
3.6.2	Intérêts . . . . .	29
3.6.3	Analyse préliminaire . . . . .	30
3.6.4	Mise en oeuvre . . . . .	31
<b>4</b>	<b>Ajouts de nouvelles fonctionnalités</b>	<b>32</b>
4.1	Mise en place du SCRUD . . . . .	32
4.1.1	Gestion des entreprises . . . . .	32
4.1.2	Gestion des parcours . . . . .	34
4.1.3	Ajouts de liens vers les fiches . . . . .	35
4.1.4	Recherche par étudiant dans le listing des stages . . . . .	36
4.2	Listing stages/utilisateurs de l' année courante . . . . .	36
4.3	Connexion via le CAS . . . . .	37
<b>5</b>	<b>Vers un déploiement à tous les départements ?</b>	<b>39</b>

<b>6</b>	<b>Bilan du projet</b>	<b>40</b>
6.1	Bilan chiffré . . . . .	40
6.1.1	Sur la base de données . . . . .	40
6.1.2	Sur le code source . . . . .	40
6.2	Perspectives concernant la plate-forme . . . . .	42
6.3	Conclusion . . . . .	43
<b>7</b>	<b>Bibliographie</b>	<b>44</b>
<b>8</b>	<b>Annexes</b>	<b>45</b>
8.1	Compte rendus des réunions . . . . .	45
8.1.1	Compte rendu de la réunion du 21/01 . . . . .	45
8.1.2	Compte rendu de la réunion du 04/02 . . . . .	48
8.1.3	Compte rendu de la réunion du 10/02 . . . . .	50
8.1.4	Compte rendu de la réunion du 26/02 . . . . .	51
8.1.5	Compte rendu des réunions du 11/03 . . . . .	52
8.1.6	Compte rendu de la réunion du 25/03 . . . . .	54
8.1.7	Compte rendu de la réunion du 14/04 . . . . .	55
8.1.8	Compte rendu de la réunion du 20/05 . . . . .	55

# 1 Introduction

## 1.1 Généralités

Ce projet a été réalisé dans le cadre de l'unité d'enseignement GMIN208 "Travail d'Etude et de Recherche (TER)" qui est commune à tous les étudiants de première année de Master Informatique à la faculté des sciences de l'Université Montpellier.

- Cette unité d'enseignement poursuit les objectifs suivants :
- travail en groupe (communication, organisation et synchronisation)
  - programmation d'une vraie application informatique
  - tests unitaires et tests d'intégration
  - soutenance orale
  - réalisation d'un rapport écrit

Le responsable de cette unité d'enseignement est Mathieu Lafourcade, maître de conférences à la faculté de sciences de l'Université Montpellier et associé au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM).

Quant à notre TER, il a été proposé et encadré par Eric Bourreau. Nous tenons à lui adresser nos remerciements pour le suivi qu'il a apporté à ce projet, ses conseils et ses explications. Nous remercions également les enseignants qui ont accepté de nous rencontrer afin de répondre à nos questions (Marianne Huchard, Michelle Joab) ainsi que Michelle Auge, secrétaire au département informatique

## 1.2 Présentation de la plate-forme

**GStage** est une plate-forme de gestion des stages professionnels du Master 2 au sein du département d'informatique de l'Université de Montpellier. Elle a été développée par des étudiants du Master 1 IPS (Informatique pour les sciences) en 2009. Depuis, chaque année d'autres étudiants de Master 1 assurent son développement et y ajoutent des fonctionnalités en fonction des différents besoins des utilisateurs de la plate-forme (étudiants, enseignants, entreprises, secrétaires, responsables de parcours et administrateurs).

Accessible sur le web à l'adresse <http://www.stagefds.univ-montp2.fr/>, la plate-forme a été conçue grâce aux langages HTML, CSS, PHP et JavaScript et utilise MySQL comme système de gestion de base de données.

GStage permet d'assurer toutes les opérations nécessaires à la gestion des stages, principalement :

- saisie des offres de stages
- listings automatiques pour le secrétariat
- système de mails pré-remplis pour faciliter la communication
- affectation et suivi des étudiants et des tuteurs
- gestion des soutenances
- stockage des différents documents (rapports, fiches...)

Elle est destinée à plusieurs types d'utilisateurs :

- les étudiants
- les entreprises
- les tuteurs (enseignants)
- les secrétaires du département informatique
- les responsables de parcours/formation
- les administrateurs

Le fonctionnement du site suit les étapes suivantes :

1. Publication d'une offre de stage par une entreprise, un étudiant ou un enseignant
2. Validation de l'offre par le responsable de parcours
3. Les étudiants consultent et postulent aux offres de stages
4. Acceptation de l'étudiant par une entreprise
5. Les tuteurs choisissent les stages qu'ils souhaitent encadrer
6. Affectations des tuteurs par le responsable de formation
7. Signature de la convention du stage par l'étudiant, l'entreprise et le responsable de formation.
8. Suivi du stage
9. Soutenance de stage



FIGURE 1 – Page d’accueil de GStage

### 1.3 Sujet et objectifs de ce TER

Au début du projet, ces 3 objectifs nous ont été présentés :

- réingénierie logicielle sur la plate-forme
- ajouts de nouvelles fonctionnalités
- correction des bugs existants

En effet, comme pour tout logiciel, nous étions arrivés au moment où chaque nouvel ajout de fonctionnalité sur la plate-forme engendrait des dysfonctionnements sur les autres modules de l’application...

L’objectif principal de ce TER a donc été d’apporter une vision génie logiciel sur la plate-forme afin de réorganiser ses différents modules :

- étude de la documentation des précédents développements
- fusion ou scission de tables de la base de données
- consolidation du modèle UML
- intégration de patrons de conception
- suppression du code mort et des fonctionnalités non utilisées

Tout ceci dans l’optique de robustifier les futurs développements de l’application et faciliter sa maintenance.

Nous avons également pour objectif d’ajouter quelques fonctionnalités demandées par les utilisateurs. La mise en place du CRUD et l’authentification via le CAS ont notamment été évoqués au début de ce projet.

Enfin, le dernier objectif a été la correction des bugs existants, nous expliquons cette étape du projet dans le prochain chapitre.

## 2 Correction des bugs existants

Dès le début du projet, nous avons hébergé le code source de l'application sur le GitLab proposé par le service informatique de la Faculté des Sciences (SIF). En effet, ce dernier offre un logiciel de suivi de problèmes qui a permis à notre encadrant de nous soumettre les bugs existants à l'aide de tickets, afin que nous les corrigions.

Pour résoudre ces bugs, nous nous sommes inspirés du modèle du logiciel de gestion des bugs, Bugzilla, à savoir : descriptif de bug par l'utilisateur, analyse fonctionnelle, proposition de correction, puis cas de test et enfin mise en production.

### 2.1 Stages qui n'apparaissent pas dans les listings

#### Descriptif du bug : (par E.Bourreau et M.Joab)

Certains stages saisis, par les étudiants notamment, n'apparaissent pas dans le listing des stages des responsables

#### Cause :

Après analyse du code et quelques tests, nous nous sommes aperçus que, lors de la saisie de l'offre d'un stage, l'utilisateur n'est pas obligé de saisir la date de début du stage.

Or la requête qui récupère la liste des stages ne récupère que ceux de l'année universitaire courante grâce à une condition sur la date de début du stage. Par conséquent, si le stage n'a pas de date de début alors il n'est pas affiché...

#### Proposition de correction :

Si la date de début de stage n'a pas été saisie par l'utilisateur, on la met par défaut à la date à laquelle a eu lieu la saisie grâce à la fonction prédéfinie NOW() de PHP.

```
if ($this->dateDebut == null || $this->dateDebut == "----/--/--")
    $sql .= "NOW() ";
else
    $sql .= "' ' . $BD->escapeString($this->dateDebut) . "' ' ;
```

### 2.2 Mauvais tuteur en entreprise pour un stage

#### Descriptif du bug : (par E.Bourreau et M.Auge)

Le tuteur en entreprise qui s'affiche sur la fiche d'un stage ne correspond pas à celui qui a été saisi initialement, c'est l'utilisateur Yannick L. qui est affiché sur la fiche.

**Cause :**

Les paramètres spécifiés pour la méthode Utilisateur : :init() ne correspondent pas aux arguments qui sont passés pour créer le tuteur en entreprise

**Proposition de correction :**

On a utilisé la méthode qui a les bons paramètres (à savoir Utilisateur : :initOld()) en attendant le refactoring...

```
$contact = new Utilisateur();  
$contact->initOld($_POST["prenomContactEnt"], $_POST["nomContactEnt"],  
$_POST["emailContactEnt"], SECURITY_ENT, $_POST["telContactEnt"],  
$_POST["faxContactEnt"], $_SESSION['createStage']['ent']->id);
```

## 2.3 Suppression des liens vers les conventions

**Descriptif du bug (par M.Auge) :**

Supprimer le lien de téléchargement de la convention de stage qui n'a plus lieu d'être.

**Proposition de correction :**

On a tout simplement supprimer les lignes de code suivantes, qui affichaient le lien de téléchargement de la convention :

```
$content .= "<fieldset><legend_class=\"legendh2\">  
Téléchargement de la convention de stage : </legend>";  
$content .="<tr><td_width=\"15%\"><b>Convention de stage en format pdf :  
</b></td><td><input_type=\"submit\"_class=\"bouton\"_name=\"submit\"  
value=\"Télécharger\"_onclick=\"window.open('convention.php');\"_/>  
</td></fieldset>";
```

## 2.4 Tuteurs ne peuvent pas encadrer plus de 5 stages

**Descriptif du bug : (par E.Bourreau)**

Un enseignant qui a encadré 5 stages dans le passé ne peut plus encadrer de stage. Or la limite de 5 stages devrait être remise à jour au début de chaque année scolaire.

**Cause :**

La requête qui retourne le nombre de stages affectés à un tuteur ne se fait pas sur l'année scolaire courante.

**Proposition de correction :**

On ajoute une condition à la requête de manière à ce qu'elle retourne le nombre de stages affecté à un tuteur de l'année scolaire courante seulement.



## **3 Réingénierie logicielle (refactoring)**

### **3.1 Qu'est ce que la réingénierie logicielle ?**

#### **3.1.1 Définition**

La réingénierie logicielle (ou refactoring en anglais) est l'étude et la modification d'un système logiciel en vue de le recréer sous une nouvelle forme plus performante. Elle n'altère pas les fonctionnalités du logiciel mais peut éventuellement en créer de nouvelles.

#### **3.1.2 Intérêts**

Dans un secteur en évolution permanente, les logiciels deviennent de plus en plus rapidement obsolètes. En effet, certaines applications ne sont pas adaptées au changement de technologies et perdent de ce fait de leur performance. La réingénierie logicielle permet donc d'améliorer les performances générales de ces logiciels, d'augmenter leur fiabilité et également de réduire les coûts d'exploitation.

De surcroît, le coût de maintenance de vieux logiciels est élevé : certaines fonctionnalités sont obsolètes, le support matériel est dépassé... La réingénierie logicielle est moins coûteuse et plus sécurisante qu'un nouveau développement complet de l'application.

De plus, faire de la réingénierie logicielle est parfois inévitable sur certaines applications indispensables. L'objectif du refactoring est d'améliorer la maintenabilité du logiciel et d'assurer une migration des logiciels anciens vers de nouveaux matériels plus performants et de nouvelles technologies de programmation plus modulables et plus aisées à maintenir.

Dans notre cas, la réingénierie logicielle nous a permis entre autre, de consolider le modèle de la plate-forme et de supprimer les fonctionnalités obsolètes ou plus utilisés. En faisant cela, les développements futurs et la maintenance du site est grandement facilité.

### 3.1.3 Processus de mise en oeuvre

Comme nous l'a expliqué Marianne Huchard, enseignante à l'université, chercheuse au LIRMM et spécialiste en Génie Logiciel et ingénierie des modèles, le champ de domaine de la réingénierie logicielle est très vaste et il n'existe pas une mise en œuvre unique mais plusieurs. Il convient donc de cibler et de classer chaque démarches à effectuer.

En amont de la phase de réingénierie logicielle proprement dite, il est nécessaire de situer et d'analyser chaque contexte dans lequel le processus s'inscrit. Il faut évaluer la faisabilité et les bénéfices de l'opération. En premier lieu il faut décrire l'environnement, c'est à dire comprendre l'organisation logicielle et l'architecture. Puis il convient d'effectuer un inventaire afin d'identifier tous les composants et donner leurs caractéristiques. A partir de là, la phase d'analyse peut commencer afin d'évaluer le potentiel de chaque composant et leur pertinence.

Le processus de réingénierie logicielle s'organise autour de deux grands groupes de processus. D'un côté, l'ingénierie inverse et de l'autre l'ingénierie de reconstruction.

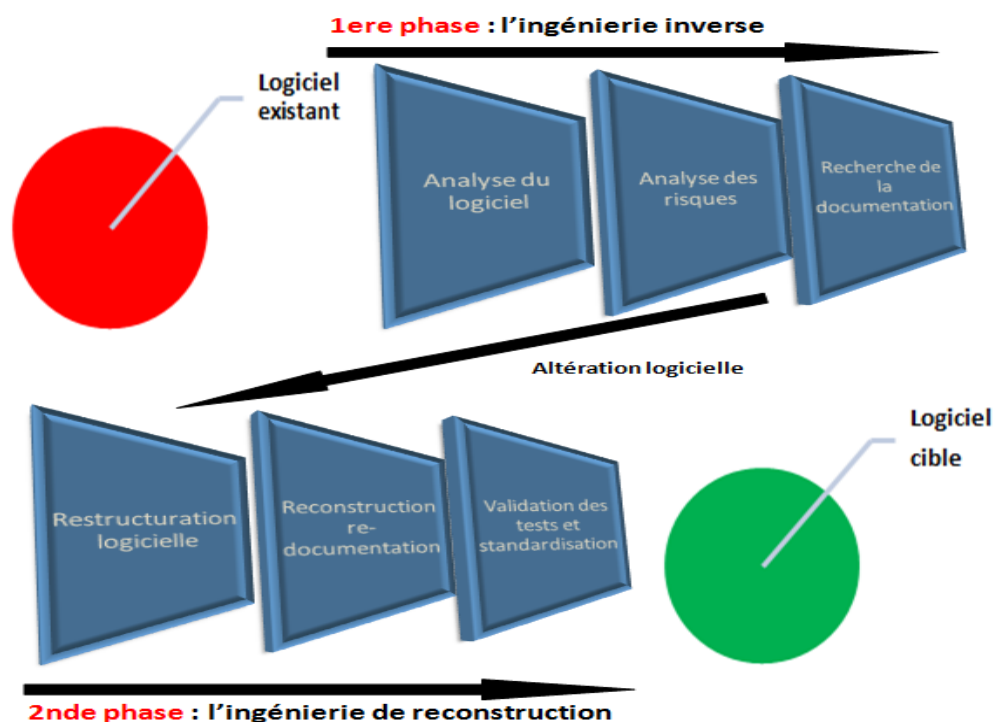


FIGURE 2 – Etapes de la réingénierie logicielle (© Wikipedia)

La première étape de la réingénierie logicielle est la **rétroingénierie** ou ingénierie inverse (reverse engineering en anglais). Ce processus a pour but d'analyser un programme afin d'identifier ses composants et leurs relations. Cette étape inclut également la recherche de la documentation pour améliorer la compréhension du logiciel. Cette phase est utile à la création d'une représentation du logiciel sous une autre forme ou à un niveau d'abstraction plus élevé. Elle permet de reconnaître les éléments à conserver de ceux qui doivent être écartés ou remodelés.

La seconde étape est le processus d'ingénierie de reconstruction (forward engineering). Ce processus englobe des sous domaines tels que la redocumentation, la restructuration du code et la recomposition du logiciel. Elle se termine par un ensemble de tests pour valider la nouvelle solution et permettre sa standardisation.

Le refactoring sur notre plate-forme a été mis en oeuvre en suivant ces 2 étapes.

Dans un premier temps, nous avons étudié et recensé les composants (tables, classes, fichiers etc.) de la plate-forme afin d'en comprendre son fonctionnement. C'est également lors de cette étape que nous avons étudié les travaux entrepris par les précédents développeurs, grâce à la lecture des rapports qui nous ont été transmis.

Une fois les rouages du site compris, nous avons proposé des solutions pour l'améliorer. C'est alors que nous avons entrepris des travaux de recomposition et de restructuration du code de la plate-forme.

## 3.2 Réingénierie sur la base de données

### 3.2.1 Rétro-ingénierie sur la base de données

#### 3.2.1.1 Suppression des tables phpBB

Lors des précédentes phases de développements, les étudiants avaient voulu relier la plate-forme à un forum phpBB afin que les utilisateurs puissent communiquer entre eux plus facilement.

Or depuis, le projet a été abandonné mais une soixantaine de tables vides pour la plupart et non-utilisés sont restés dans la base de données.




 <b>phpbb_acl_roles</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_acl_roles_data</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_acl_users</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_attachments</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_banlist</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_bbcodes</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_bookmarks</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_bots</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer
 <b>phpbb_config</b>	 Afficher	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer

FIGURE 3 – Quelques tables phpBB vides et non utilisés...

Avant de démarrer la phase de rétro-ingénierie en elle-même, nous avons donc fait le choix de les supprimer.

#### 3.2.1.2 Génération du modèle physique des données

Afin de nous faire une idée des différentes relations entre les tables, nous avons recherché un outil capable de générer le modèle physique des données (MPD) de la base de données.

Nous avons trouvé **MySQL Workbench**, logiciel d'administration de bases de données MySQL développé par Oracle.

Ce dernier contient une fonctionnalité "Reverse Engineer" capable de générer le MPD d'une base de données à partir d'un script de création SQL ou bien en se connectant directement au serveur MySQL.

Nous avons donc appliqué cet outil sur la base de données, et nous avons obtenu le modèle physique des données représenté par la FIGURE 4.



FIGURE 4 – Modèle physique des données d'origine



En visualisant le modèle générée, nous avons remarqué à notre grande surprise, que techniquement il n'y aucune relation entre les tables ! En effet, les développeurs des années passées n'ont défini aucune clés étrangères lors de la création des tables !

Par conséquent, afin d'avoir une représentation visuelle des associations supposés entre les tables, nous avons simulé les clés étrangères, c'est à dire, défini les clés étrangères de chaque table manuellement avec MySQL Work-Bench et nous avons obtenu la FIGURE 5 ci-dessus.

### 3.2.1.3 Description du modèle physique des données

On se retrouve avec 24 tables :

- **pgs\_utilisateur** qui contient diverses informations spécifiques aux utilisateurs (nom, prénom, mail, rôle etc.) et qui contient une clé étrangère **idEntreprise** pour lier les utilisateurs ayant le rôle "ENT" à l'entreprise qu'ils représentent.
- **pgs\_stage** qui détient des données sur les stages (titre, description, durée etc.) et dispose de 6 clés étrangères : une pour chaque utilisateur intervenant dans le stage (étudiant, ajoutateur, tuteur entreprise, tuteur universitaire, rapporteur) + **idEntreprise** pour relier le stage à une entreprise.
- **pgs\_entreprise** qui dispose d'informations sur les entreprises (nom, adresse, siret) et possède un responsable administratif dans la table Utilisateur (clé étrangère **idrespadm**)
- **pgs\_parcours** qui a pour clé primaire l'acronyme du parcours sous forme de chaîne de caractères (AIGLE, IMAGINA...) et qui possède l'intitulé et la description du parcours. Elle est reliée à la table **pgs\_utilisateur** par la clé étrangère **idResponsable** qui référence le responsable du parcours.
- **pgs\_stage\_parcours** qui représente le fait qu'un stage peut concorder avec plusieurs parcours.
- **pgs\_stage\_commentaire** qui permet à un utilisateur de poster un commentaire sur la fiche d'un stage à une date donnée.
- **pgs\_pieces\_jointes** qui permet à un utilisateur de lier une pièce jointe d'un certain type (CV, rapport etc.) à un stage.

- **pgs\_soutenance** qui contient les noms des acteurs de la soutenance (étudiant, tuteurs, entreprises, rapporteurs) sous forme de chaîne de caractères. Ses 2 clés étrangères référencent un stage et un créneau.
- **pgs\_creneau** qui est relié à la table **pgs\_salles** et qui stocke des données telles que la date et l’heure du créneau et si celui-ci est disponible ou pas.
- **pgs\_salles** qui détient des données sur les salles accueillant les soutenances comme son nom, le bâtiment dans lequel elle se trouve et le nombre de personnes qu’elle peut accueillir.
- **pgs\_info** et **temp\_info** stockent quant à elles des informations complémentaires sur les utilisateurs de type étudiant, comme le numéro d’INE, le parcours et l’année de promotion.
- Les tables **pgs\_adresse**, **pgs\_telephone**, **pgs\_langueparlee**, **pgs\_plateformemaitrisee**, **pgs\_langueprogmaitrisee**, **pgs\_mail**, **pgs\_logicielmaîtrise**, **pgs\_competenceinfo**, **pgs\_scolarite**, **pgs\_experience** sont là pour stocker des informations sur les utilisateurs afin de générer des CV automatiquement.
- Les tables **pgs\_config** et **table 85** contiennent les noms et les valeurs des variables globales qui configurent la plate-forme, comme par exemple les dates de saisies et des soutenances, des codes couleurs, le mail de l’admin etc.

### 3.2.2 Recomposition de la base de données

#### 3.2.2.1 Analyse du modèle physique des données

Nous avons relevé de nombreuses incohérences sur ce modèle physique des données (FIGURE 5) :

- Dans la table **pgs\_utilisateur** :
  - L’attribut **idEntreprise** est spécifique aux utilisateurs qui ont le rôle de tuteur en entreprise (ENT), or même les étudiants et les secrétaires possèdent cet attribut qui ne leur sera pas utile.
  - Présence de 3 attributs pour gérer le fait qu’un utilisateur possède une adresse (**adresse**, **ville**, **codePostal**) alors qu’il existe une table **pgs\_adresse**.
  - Présence d’un attribut **telephone** alors qu’il existe une table **pgs\_telephone**.
  - Présence d’un attribut **email** alors qu’il existe une table **pgs\_mail**.



- Concernant la table `pgs_info` :
  - Présence d'un attribut `parcours` de type `VARCHAR` qui détermine le parcours de l'étudiant. Or il existe une table `pgs_parcours`, une référence vers celle-ci aurait été plus judicieux.
  - Elle semble avoir le même objectif que la table `temp_info` : il y a de la redondance.
- La clé primaire de la table `pgs_parcours` correspond à l'acronyme du parcours (AIGLE ou IMAGINA par exemple), c'est à dire une chaîne de caractères. Or une bonne clé primaire doit être définie comme un entier, non nulle, auto-incrémentée et surtout non visible par l'utilisateur.
- La table `pgs_stage` contient une référence sur la table `pgs_entreprise`, or cela est inutile car on peut récupérer l'entreprise en passant par `idTuteurEntreprise` qui référence la table `pgs_utilisateur` et cette dernière possède un attribut `idEntreprise`.
- La table `pgs_creneau` contient 2 attributs pour stocker la date et l'heure de début du créneau : `datedeb` de type `DATE` et `heuredeb` de type `TIME`. On peut fusionner ces deux attributs en un seul grâce au type `DATETIME`.

### 3.2.2.2 Souhaits des utilisateurs

Lors de la recomposition de la base de données, en plus de nos propres remarques sur les incohérences du modèle (voir ci-dessus), il a fallu prendre en compte celle des utilisateurs, et notamment du responsable technique de la plate-forme (Eric Bourreau) et de la responsable de stages (Michelle Joab).

Mme Joab nous a été demandé de trouver une solution pour pouvoir gérer les stages de M1 ou de Licence en plus de ceux du M2, et pour ne plus avoir s'afficher les anciens parcours (GL, CASAR, DIWEB...). Elle nous a également demandé un moyen de stocker plus d'informations sur les entreprises (chiffre d'affaires, nombre de salariés etc.) et sur les tuteurs en entreprise (fonction occupée).

Quant à M.Bourreau, il a souhaité que nous supprimons tout ce que touche à la gestion des CVs. En effet, la plate-forme offrait la possibilité aux étudiants de rentrer des informations susceptibles d'intéresser les entreprises (formations, langages et logiciels maîtrisés, langues parlées etc.) mais cette fonctionnalité a été très peu utilisée et a donc été jugée inutile.

Le MPD présenté précédemment (FIGURE 5) ne permet pas à un utilisateur d'avoir plusieurs rôles, puisque le rôle est un attribut dans la table `pgs_utilisateur`. Or un utilisateur peut avoir plusieurs rôles :

- Un responsable de formation (RESP) est également enseignant (TUT)
- Un administrateur (ADM) est là aussi, souvent enseignant (TUT)
- Un enseignant (TUT) peut être amené à être aussi tuteur en entreprise (ENT) si il prend un étudiant en stage dans son laboratoire.

C'est pourquoi nous avons proposer une solution (FIGURE 6) impliquant la création de 2 nouvelles tables : une table `pgs_role` contenant les différents rôles (ETUD, ENT, TUT etc.) et une table `pgs_utilisateur_role` permettant à un étudiant d'avoir plusieurs rôles.

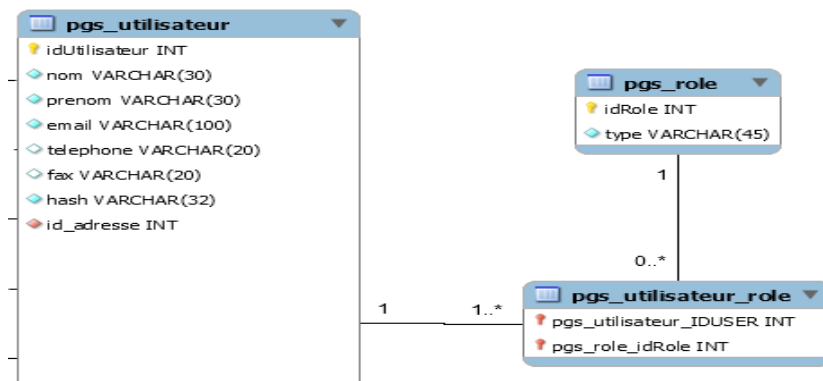


FIGURE 6 – Solution pour qu'un utilisateur ait plusieurs rôles

Cependant, cette solution n'a pas été implémentée. Notre tuteur nous conseillant plutôt de gérer la "double casquette" avec un compte différent pour chaque rôle de l'utilisateur.

Enfin, nous avons remarqué que la table `pgs_soutenance` stocke les différents acteurs de la soutenance (étudiants, tuteurs, entreprise) sous forme de chaînes de caractères. Or ces informations sont récupérables grâce à la clé étrangère qui référence la table `pgs_stage` et cette dernière contient des références sur tous les acteurs.

Mais après concertation avec notre tuteur, il s'avère que ceci était voulu, en effet, les acteurs d'une soutenance ne sont pas toujours les acteurs d'un stage, notamment à cause de l'indisponibilité de certains. Stocker ces informations avec des attributs de type VARCHAR permet donc de les modifier.

### 3.2.2.3 Proposition du nouveau modèle physique des données

La prise en compte des souhaits des utilisateurs, ainsi que nos remarques sur les incohérences du modèle d'origine, nous ont permis de générer un nouveau modèle physique des données de la plate-forme (FIGURE 7).

Premièrement, comme souhaité, les tables `pgs_scolarite`, `pgs_experience`, `pgs_plateformemaitrisee`, `pgs_langueparlee`, `pgs_langageprogmaîtrise`, `pgs_logicielmaîtrise`, `pgs_compétenceinfo`, `pgs_mail` et `pgs_telephone`, qui permettent de stocker des informations apparaissant sur un CV ont disparu du modèle.

La table `pgs_utilisateur` a été nettoyé, seuls les attributs ayant une utilité et communs à TOUS les utilisateurs ont été conservés, à savoir : nom, prénom, mail, mot de passe, téléphone, fax et le rôle. Les attributs relatifs à l'adresse ont notamment été supprimé car jugés inutiles pour les utilisateurs.

Ensuite, pour gérer le fait que les utilisateurs peuvent avoir des attributs spécifiques en fonction de leur rôle, nous avons mis en place de l'héritage.

Concrètement, nous avons créé une table `pgs_tuteurEntreprise` dont le but est de stocker toutes les données spécifiques aux utilisateurs qui ont le rôle de tuteur en entreprise (ENT). Sa clé primaire `idTuteurEntreprise` est également une clé étrangère qui référence la clé primaire de la table `pgs_utilisateur`.

La table `pgs_info` a été remplacé par la table `pgs_etudiant` qui comme son nom l'indique contient des informations spécifiques aux étudiants : le numéro INE, la promotion ainsi que le parcours grâce à une clé étrangère qui référence la table `pgs_parcours`. Là aussi, l'héritage a été mis en place grâce à la clé primaire `idEtudiant`, qui est aussi une clé étrangère référençant la table `pgs_utilisateur`. La table `temp_info` a quant à elle, était supprimée car redondante.

La table `pgs_parcours` a aussi été modifié, à commencer par sa clé primaire qui est dorénavant un entier non nul, auto-incrémenté et non plus une chaîne de caractères visible et manipulable par les utilisateurs. De plus, un attribut permettant de stocker le niveau du parcours (L3, M1, M2 etc.) a été ajouté afin de pouvoir déployer la plate-forme à tous les étudiants du département informatique comme le souhaite la responsable des stages.

Enfin, dans la table `pgs_entreprise`, pour respecter la première forme normale (1NF), les attributs `RUE1`, `RUE2`, `CODEPOSTAL` et `VILLE` ont été remplacé par une clé étrangère qui référence la nouvelle table `pgs_adresse`.

A noter également l'ajout des attributs demandé par Mme Joab dans les tables `pgs_entreprise` et `pgs_tuteurEntreprise`.

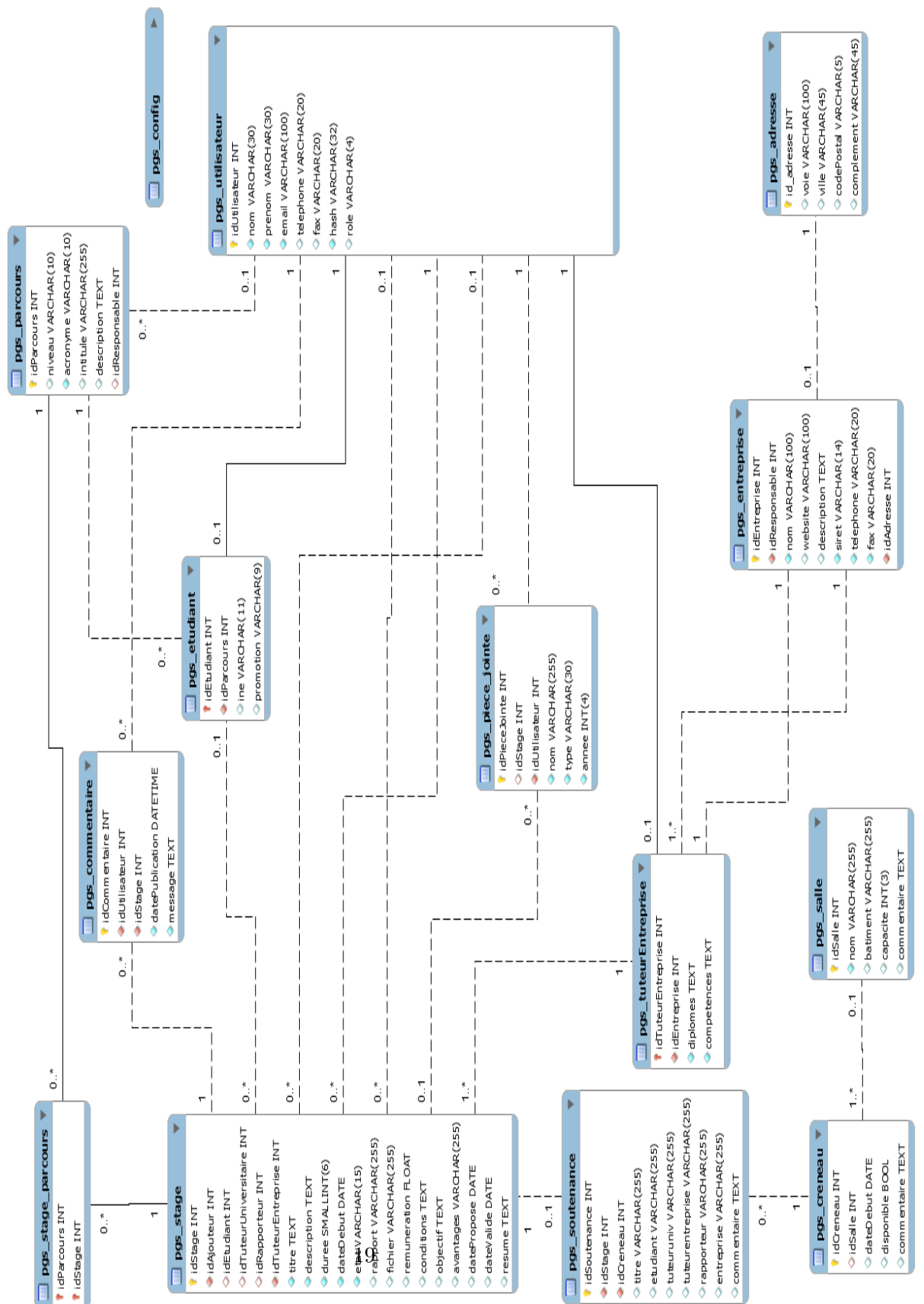


FIGURE 7 – Nouveau modèle physique des données

### 3.2.3 Restructuration du code de la plate-forme

Évidemment, toutes ces modifications apportées sur la base de données ont entraîné un remaniement du code de la plate-forme. Pour chaque changement dans la base de données, nous avons fait une analyse d'impact afin de bien cibler les conséquences de ces changements, et pour détecter les lignes de codes à transformer.

#### 3.2.3.1 Suppression des classes liés au CVs

Suite à la suppression des tables liés à la gestion des CVs, nous avons supprimé du modèle les classes associés : CompetenceInfo, Experience, LangageProgMaitrise, LangueParlee, LogicielMaitrise, PlateFormeMaitrisee, Scolarite, Mail, Telephone.

Puis, nous avons recherché, en faisant une analyse statique, les lignes de code où étaient instanciés des objets de ce type afin de les supprimer. Cela nous a permis de nous débarrasser de quelques fichiers qui n'utilisaient que des objets de ce type.

Nous avons également supprimé une dizaine de fichiers permettant la génération de CV en PDF.

#### 3.2.3.2 Modification de la classe Utilisateur

Suite à la suppression de nombreux attributs dans la table `pgs_utilisateur`, le code de la classe `Utilisateur` a dû être remanié, notamment le code des méthodes et des requêtes SQL.

Là aussi, il a fallu faire une analyse d'impact pour voir où les attributs supprimés et les méthodes modifiés étaient appelées dans le code source de l'application afin de faire les modifications adéquates.

#### 3.2.3.3 Mise en place de l'héritage entre la classe Utilisateur et les classes Etudiant/TuteurEntreprise

Après la création des tables `pgs_etudiant` et `pgs_tuteurEntreprise` qui ont des clés primaires référençant la clé primaire de la table `pgs_utilisateur`, il a fallu créer 2 nouvelles classes `Etudiant` et `TuteurEntreprise` et mettre en place l'héritage avec la classe `Utilisateur` (FIGURE 8).

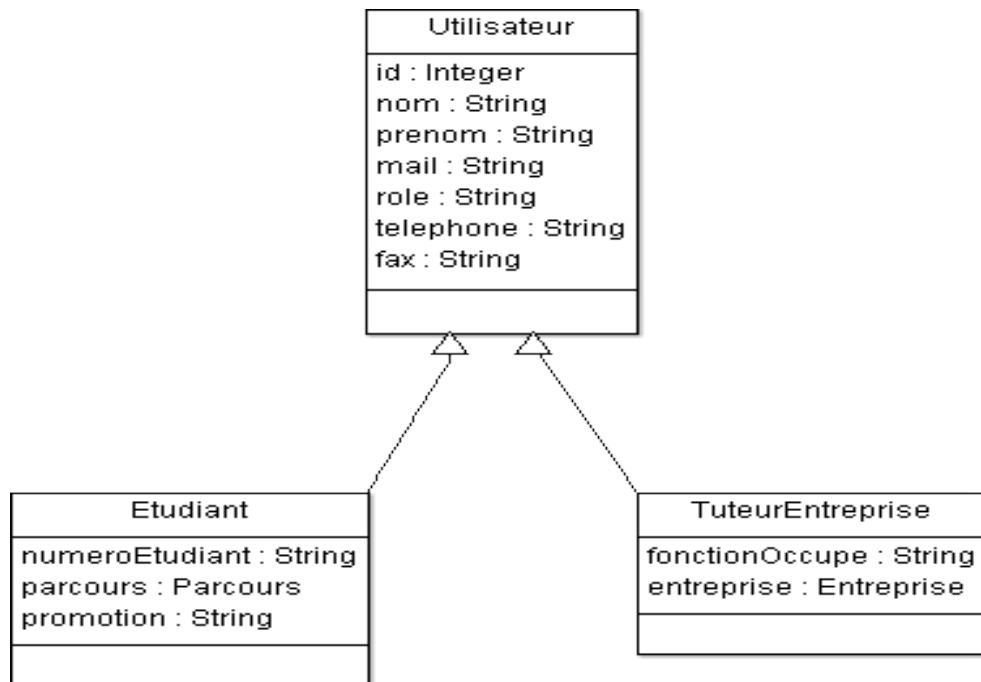


FIGURE 8 – Fiche complète d’une entreprise

Une fois l’héritage mit en place grâce au mot clé `extends` de PHP, il a fallu écrire des méthodes pour chacune de nos 2 nouvelles classes :

- un constructeur
- une méthode permettant de charger les attributs de l’objet en allant récupérer les données dans la base de données
- une méthode permettant d’insérer ou de modifier les valeurs contenus dans les attributs de l’objet, dans la base de données

Dans un souci de réutilisabilité, nous avons fait en sorte que ces 3 méthodes appellent leur équivalent dans la classe mère grâce au mot-clé `parent` de PHP.

Enfin, il a fallu chercher dans le code de l’application, les lignes où étaient instanciés des objets de type `Utilisateur`, et les remplacer par des instances de `Etudiant` ou de `TuteurEntreprise` quand cela s’avérait nécessaire.

#### 3.2.3.4 Modification de la classe **Parcours**

Suite a la modification de la table `pgs_parcours` , nous avons du modifier la classe **Parcours**. Nous avons ajouté 3 nouveaux attributs à cette classe (`niveau`, `acronyme`, `existe`) et il a fallu modifier toutes les méthodes et les requêtes en conséquence.

Ensuite, nous avons fait une analyse d'impact, afin de trouver les fichiers dans lesquels sont créés des instances de la classe **Parcours** afin d'effectuer les changements.

La code de classe **Stage** a également été modifié, car elle a pour attribut un tableau de **Parcours**. Nous avons d'ailleurs fait en sorte qu'ils contiennent des objets de type **Parcours** et non plus seulement des identifiants de **Parcours** (tableau d'entiers) comme c'était le cas auparavant.

#### 3.2.3.5 Modification de la classe **Entreprise**

Comme la classe **Parcours**, la classe **Entreprise** a elle aussi connue une greffe d'attributs, conséquence directe de l'ajout d'attributs dans la table `pgs_entreprise`. Cela s'en est suivi par une mise à jour des méthodes, des requêtes et des fichiers utilisant des instances d'**Entreprise**.

Les 4 attributs permettant de stocker une adresse (`rue1`, `rue2`, `ville`, `codePostal`) ont été remplacé par un unique attribut de type **Adresse**. Il a donc fallu parcourir le code source de l'application, afin d'instancier des objets de type **Adresse** pour remplacer l'utilisation de ces 4 attributs.

### 3.3 Réingénierie sur les permissions

La réingénierie sur les permissions est l'étape au cours de laquelle nous avons vérifié les droits et les restrictions de chaque utilisateur en fonction de leurs rôles (administrateur, responsable, secrétaire, étudiant, entreprise, tuteur).

Pour effectuer ces vérifications, nous avons créé une matrice (FIGURE 9) dans laquelle, une ligne représente une action et une colonne représente un type d'utilisateur. De cette manière, toutes les lignes représentent toutes les actions proposées par le site et les colonnes tous les types d'utilisateurs. On a ensuite coché les cases où un type d'utilisateur pouvait effectuer une action.

Nous n'avons repéré aucune anomalie donc on peut affirmer que les droits et les restrictions ont bien été définis lors des précédents développements.

Nous n'avons eu aucune modification à apporter, mais il était primordial de faire cette étude sur les permissions pour ne pas avoir de surprise dans la suite du projet.



	ADM	RESP	SECR	ETUD	ENT	TUT
<b>ADMINISTRATION</b>						
Configurer la plateforme	x					
Gérer les parcours	x					
Envoyer des mails	x					
<b>STAGES</b>						
Proposer un stage	x		x	x	x	
Consulter la fiche d'un stage	x	x	x	x (ses stages)	x (ses stages)	x
Modifier un stage (dont affecter un etud, un tut etc.)	x	x				
Supprimer un stage	x	x				
Affecter des parcours à un stage	x	x				
Listing étudiants disponibles	x	x			x	
Chosir d'accompagner un stage						x
Ajouter une pièce jointe à un stage	x	x	x			
Ajouter des commentaire a un stage	x	x	x	x	x	x
Télécharger la fiche synoptique d'un stage	x	x	x			
Consulter la liste des étudiants	x	x	x			
Consulter la liste des tuteurs	x	x				
Export CSV des stages et des étudiants	x	x	x			
<b>UTILISATEURS</b>						
Gérer les utilisateurs (consulter, modifier, supprimer)	x		x			
Importer des utilisateurs	x		x			
Importer des étudiants et leur scolarité	x		x			
Ajouter un utilisateur	x					
<b>SOUTENANCES</b>						
Consulter le planning des soutenances	x	x	x		x	
Devenir le rapporteur d'une soutenance						x
Ajouter un rapporteur à une soutenance (si TUT du stage)						x
Consulter les statistiques enseignants	x	x	x			x
Télécharger la fiche d'appréciation de stage par entreprise	x	x	x	x	x	x
Télécharger la fiche de suivi du stage d'un étudiant	x	x	x	x	x	x
Télécharger la fiche de notation d'une soutenance	x	x	x			x
Activier/Desactiver un créneau	x	x	x			
Modifier une soutenance	x	x	x			
Listing des soutenances	x	x	x			
Export CSV des soutenances	x	x	x			
Configurer le planning	x					
<b>AUTRES</b>						
Modifier son mot de passe	x	x	x	x	x	x
Modifier son profil				x		
Consulter les statistiques de la plateforme	x	x	x			
Consulter l'historique des stages	x	x	x	x		
Télécharger d'anciens rapports de stages	x	x		x		

FIGURE 9 – Permissions des utilisateurs en fonction de leur rôle

### 3.4 Harmonisation de l'ajout des utilisateurs

Il existe sur la plate-forme plusieurs moyens d'importer ou d'ajouter des utilisateurs, notamment au moyen de fichiers CSV.

Cependant, ces fonctionnalités présentaient des inconvénients, notamment pour l'ajout et l'importation d'étudiants. En effet, certaines de ces fonctionnalités ne demandent pas de saisir le parcours et la promotion des étudiants, ce qui pose ensuite certaines complications notamment au niveau de l'affichage des listings par année scolaire ou par parcours. Nous avons donc dû faire un travail d'harmonisation de toutes ces options.

#### 3.4.1 Import CSV des enseignants

Anciennement appelé "Importer des utilisateurs", cette fonctionnalité permettait d'importer des étudiants et des enseignants à l'aide d'un fichier CSV contenant leurs noms, prénoms et adresses mails seulement.

Cela posait problème car avec cette fonctionnalité, on pouvait ajouter des étudiants sans spécifier leur parcours et leur promotion, informations pourtant essentielles au bon fonctionnement de l'application.

Nous avons donc renommé cette fonctionnalité "Importer des enseignants" et nous l'avons remanié de manière à ce que l'on ne puisse plus importer des étudiants, mais seulement des enseignants.

On oblige l'administrateur à remplir les trois champs demandés sinon l'import ne se fait pas et on lui demande de ressaisir tous les champs.

**Import des enseignants**

-----

Importer un fichier .csv (séparateur ; - format : **prenom;nom;adresse\_mail**) -----

Fichier :  Aucun fichier sélectionné.

FIGURE 10 – Importer des enseignants

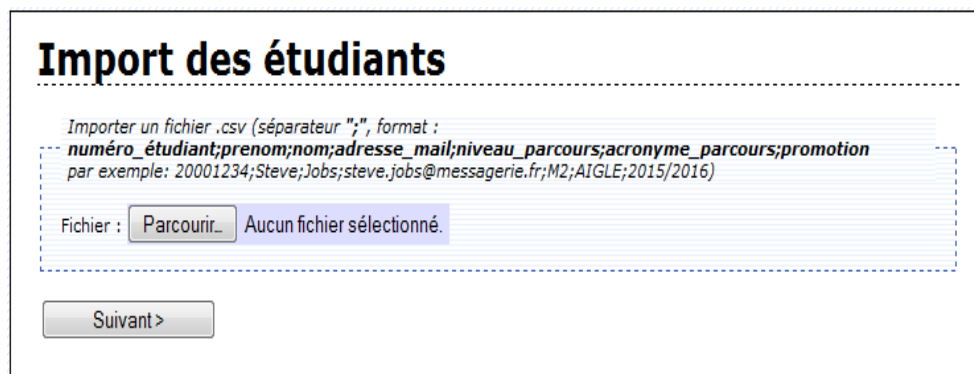
### 3.4.2 Import CSV des étudiants

Cette fonctionnalité permet d'importer des étudiants avec des informations sur leur scolarité (parcours et promotion), à l'aide de fichiers CSV.

Nous l'avons optimisé de manière à gérer les redoublants : on vérifie si l'étudiant existe déjà dans notre base de données. Si oui on écrase ses informations concernant sa scolarité, sinon on l'ajoute.

Nous avons également ajouté un champ permettant d'insérer le niveau de l'étudiant (M1, M2...) en adéquation avec les changements du modèle physique des données que nous avons réalisé.

La aussi, on oblige l'utilisateur à remplir tous les champs demandés sinon l'import ne se fait pas.



**Import des étudiants**

Importer un fichier .csv (séparateur ";", format :  
**numéro\_étudiant;prenom;nom;adresse\_mail;niveau\_parcours;acronyme\_parcours;promotion**  
par exemple: 20001234;Steve;Jobs;steve.jobs@messagerie.fr;M2;AIGLE;2015/2016)

Fichier :  Aucun fichier sélectionné.

FIGURE 11 – Importer des étudiants

### 3.4.3 Ajouter un utilisateur à la fois

C'est une fonctionnalité qui permet d'ajouter un utilisateur en saisissant ses informations dans un formulaire et en spécifiant son rôle.

Là aussi, cette fonctionnalité présentait l'inconvénient ne pas obliger l'administrateur à saisir des informations liés à la scolarité (numero étudiant, promotion, parcours) si il ajoute un étudiant.

Nous avons donc écrit une fonction JavaScript qui fait apparaître les champs relatifs à la scolarité d'un étudiant si on souhaite ajouter un utilisateur de type étudiant, et qui inversement, les masque si on souhaite ajouter un autre type d'utilisateurs (tuteurs, secrétaires etc.). De cette manière, on affiche les champs appropriés en fonction du rôle de l'utilisateur que l'on souhaite ajouter (voir FIGURE 12)

Ajout d'un utilisateur	Ajout d'un utilisateur
<p><i>Donnees personnelles</i></p> <p>Nom : <input type="text"/></p> <p>Prénom : <input type="text"/></p> <p>Adresse email : <input type="text"/></p> <p>Mot de passe : <input type="text"/> <input type="checkbox"/> Générer</p> <p>Role : Enseignant Tuteur ▼</p> <p>Telephone : <input type="text"/></p> <p>Fax : <input type="text"/></p>	<p><i>Donnees personnelles</i></p> <p>Nom : <input type="text"/></p> <p>Prénom : <input type="text"/></p> <p>Adresse email : <input type="text"/></p> <p>Mot de passe : <input type="text"/> <input type="checkbox"/> Générer</p> <p>Role : Etudiant ▼</p> <p>Telephone : <input type="text"/></p> <p>Fax : <input type="text"/></p> <p>Numero etudiant : <input type="text"/></p> <p>Promotion : <input type="text" value="2014/2015"/></p> <p>Parcours : M2 AIGLE ▼</p> <p><input type="button" value="Créer cet utilisateur"/></p>

FIGURE 12 – Ajout d'un enseignant (gauche) et d'un étudiant (droite)

### 3.5 Suppression du code mort

Le code mort désigne les lignes de code qui ne sont jamais exécutées et qui par conséquent ne servent à rien.

Cela peut être des méthodes jamais appelées, des fichiers entiers non utilisés, ou bien des blocs de commentaires contenant d'anciennes lignes de codes.

Le code mort rend la lecture du code source plus complexe et complique le travail du développeur responsable de la maintenance de l'application.

La crainte d'avoir un jour besoin de certains fichiers ou certaines méthodes pousse à les garder en place, et le code mort s'empile...

Pour palier à ce problème, nous avons utilisé un outil de gestion de version (GitLab de la FdS) pour archiver notre code et retrouver les anciennes versions d'un fichier si nécessaire.

Nous avons procédé à l'élimination du code mort en utilisant l'outil "Find Usages" de NetBeans qui permet de faire une recherche statique sur le code source afin de voir le cas échéant, où une classe et ses méthodes sont appelés.

Nous avons également utilisé l'outil "Rechercher dans les fichiers d'un dossier" de l'éditeur de texte NotePad++ afin de détecter les fichiers jamais appelés. Cela nous a permis de supprimer de nombreux fichiers : des libraires non utilisés (PHPBB, Facebook), des fichiers de tests oubliés et des fichiers obsolètes.

Quant aux blocs de commentaires inutiles, nous les avons supprimé manuellement dès lors que nous en avons rencontré.

## 3.6 Application d'un patron de conception : MVC

### 3.6.1 Définition

Le patron de conception MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application. Il a pour but de mieux organiser le code source de d'une application en le divisant en trois parties :

- le modèle est la partie de l'application qui contient le code source associé aux données.
- la vue contient le code source lié à l'affichage de l'application.
- le contrôleur contient le code source qui traite les actions et les différentes requêtes des utilisateurs et retourne une réponse avec l'aide du modèle et de la vue.

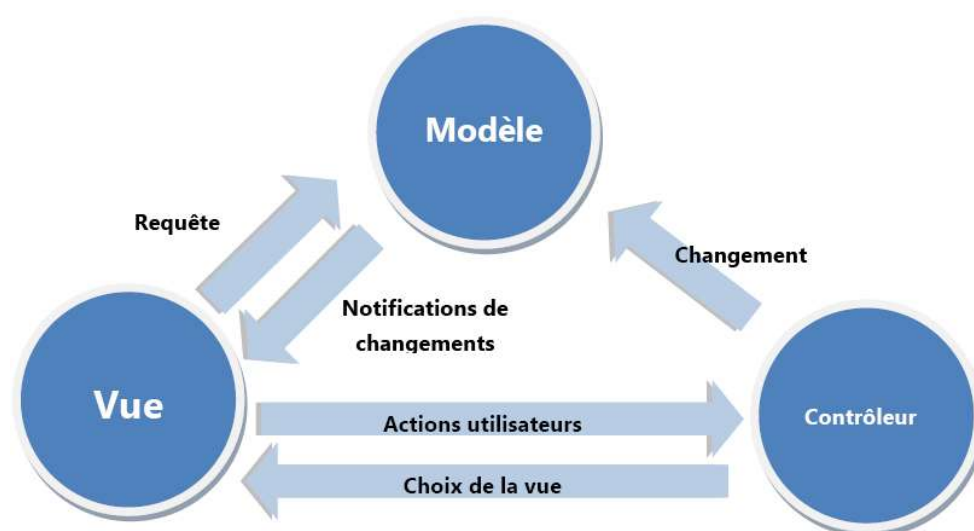


FIGURE 13 – Fonctionnement du patron de conception MVC (© C.Maneu)

Le patron de conception MVC permet donc d'organiser le code source en trois parties qui se distinguent par leurs rôles dans l'application.

### 3.6.2 Intérêts

On peut se demander quels sont les intérêts de l'application du patron de conception MVC à notre site de gestion de stage ?

Tout d'abord, il faut savoir que les intérêts du patron MVC sont surtout visibles sur les grands projets. Notre site de gestion de stage est bien un grand projet puisqu'il se compose de 1071 fichiers et 100 dossiers.

L'application du patron MVC à notre site apporte une conception plus claire et plus efficace. En effet, la séparation du modèle, de la vue et du contrôleur permet aux développeurs de se situer plus facilement dans le code source puisque chaque partie correspond à un rôle précis dans le site.

Le modèle MVC va permettre aux développeurs de gagner du temps. Plusieurs groupe de TER ont travaillé sur ce projet avant nous et d'autres groupe l'amélioreront après nous. La clarté apportée par le patron MVC va permettre au prochain groupe une adaptation et une prise en main plus rapide.

Enfin, le modèle MVC permet de rendre chaque partie indépendante et apporte une grande souplesse au niveau du développement. On peut par exemple travailler sur les fichiers correspondant à l'affichage (la vue) sans modifier les différents traitements (le contrôleur).

### **3.6.3 Analyse préliminaire**

On a fait une analyse préliminaire sur le site de gestion de stage afin de savoir comment y appliquer le mieux possible le patron MVC.

Grâce à cette analyse on constate que le patron MVC a partiellement été appliqué sur le site. En effet on constate que le modèle (les données) a déjà été séparé de la vue et du contrôleur. Le modèle dans le site de gestion correspond au dossier nommé "includes". Ce dossier contient toutes les classes qui nous permettent d'exploiter toutes les données de la base de données du site. Ces classes sont directement calquées sur les tables de la base de données. En effet chaque classe et ses attributs correspondent directement à une table et ses attributs dans la base de données. Les classes définissent aussi toutes les méthodes et les requêtes qui seront utilisées afin de les manipuler et ainsi manipuler les données.

La convention utilisée pour nommer les fichiers de ces classes est la suivante : "class.nomdelaclass.php"

Pour appliquer complètement le patron MVC il ne nous reste plus qu'à séparer la vue (l'affichage) et le contrôleur (les traitements).

### 3.6.4 Mise en oeuvre

Il n'est pas très pertinent d'appliquer le patron MVC sur de très petits fichiers (inférieur à 5 Ko). On ne l'applique donc que sur des fichiers d'assez grande taille.

Dans certains cas, la séparation du code correspondant à l'affichage (la vue) et celui correspondant aux traitements (le contrôleur) s'est faite assez naturellement. Mais dans d'autres cas, le code correspondant à l'affichage ou aux traitements ou même les deux, était situé dans des "if" ou des boucles "while" ou "for" imbriquées. Il a donc fallu dans ces cas ajouter les bonnes conditions (if,else) afin de pouvoir séparer les lignes de l'affichage et celles du traitements sans créer de bugs sur le site. Dans la plupart des cas le code correspondant au contrôleur a bien été extrait des fichiers correspondant à l'affichage. Dans certains cas il était très difficile de séparer les lignes de l'affichage à celles du traitement sans générer des bugs. Dans ces cas nous n'avons pas fait la séparation. Donc il y a encore certains fichiers de l'affichage (vue) qui contiennent un peu de traitement (contrôleur).

On a réuni tous les fichiers contenant du traitement dans le dossier nommé "contrôleur". La convention de nommage utilisée pour nommer ces fichiers est la suivante : "controleur-nomFichierVue.php". Prenons par exemple le fichier vue de nom "edit-parcours.php". Le fichier contrôleur qui lui est associé est le fichier nommé "controleur-edit-parcours.php" situé dans le répertoire contrôleur.

Une fois que les trois parties (la vue, le modèle et le contrôleur) sont bien séparées, il faut pouvoir les mettre en relation pour qu'elles travaillent ensemble. Pour faire cela, on utilise la fonction "include\_once" de PHP en en-tête des fichiers contenant la vue afin d'inclure et évaluer les fichiers du contrôleur et du modèle durant l'exécution.



## 4 Ajouts de nouvelles fonctionnalités

### 4.1 Mise en place du SCRUD

SCRUD est un acronyme informatique pour Search, Create, Read, Update, Delete, ce qui donne en français : rechercher, créer, lire, mettre à jour, supprimer. Ce qui correspond aux 5 opérations élémentaires en base de données.

SCRUD désigne un principe en développement logiciel permettant de faciliter la navigation d'un internaute en lui proposant une interface graphique permettant de gérer et de manipuler toutes les données avec facilité. Un logiciel ou un site qui ne contient pas ses 5 opérations est considéré comme incomplet.

Dans le cadre de notre TER, l'administrateur technique de la plate-forme (E.Bourreau) nous a fait part de son souhait de vouloir manipuler toutes les données sans à avoir à traiter avec phpMyAdmin. C'est pourquoi nous avons ajouté à l'application quelques fonctionnalités manquantes, permettant de manipuler les informations de certaines entités.

#### 4.1.1 Gestion des entreprises

### Listing des entreprises

Rechercher une entreprise :

Par le nom :

Par ville :

Résultats de 0 à 40. Suivant >>

	Nom	Ville	Responsable
	Agglomération de Montpellier	Montpellier	
	AGT Groupe	Mende	TEIXIDO Céline
	AL6	LATTES	Simon Nathalie
	ALES GROUP	DUBLIN	GAUTHIER TRISTAN
	Alliance Services Plus	VANVES	ESCLAINE Julien
	Alliance Telecom / Analytel	Montpellier	Denis Alain
	Amitel	VILLEURBANNE	
	Antek-Studio	St Mathieu de Tréviers	FERNEZ Jean-Luc
	Anuman Interactive	Montreuil Paris	LONGEARD Stéphane
	Aragon-eRH	Saint-Malo	CHATEAU Annie
	Arcagée	Bègles	Mauboussin Thierry
	ARDIAN FRANCE	Paris	
	AREVA	Montpellier	Artigues olivier

FIGURE 14 – Interface de la gestion des entreprises

Nous avons mis en place une interface (FIGURE 14) permettant aux administrateurs de gérer les entreprises, ce qu'ils ne pouvaient pas faire jusqu'à présent avec l'application.

Elle offre un listing de toutes les entreprises présentes dans la base de données en affichant le nom de l'entreprise, la ville où elle se situe ainsi que le nom et prénom du responsable administratif.

L'interface propose également un formulaire qui permet de rechercher des entreprises par nom ou par ville et permet de trier les entreprises par ordre alphabétique sur le nom, la ville ou le nom du responsable administratif grâce à des clics sur l'en-tête du tableau.

Enfin, l'interface propose pour chaque entreprise, 3 opérations possibles représentées par des icônes :

- la loupe pour consulter la fiche complète de l'entreprise (FIGURE 15) et retrouver plus d'informations
- le crayon pour modifier les données de l'entreprise grâce à un formulaire pré-rempli avec ses informations actuelles
- la gomme qui permet de supprimer toutes les informations de l'entreprise de la base de données (FIGURE 16)

### Fiche de NouvelleEntreprise

Nom :	NouvelleEntreprise
Responsable adm. :	Responsable Responsable
N° siret	00000000000000
Adresse :	Adresse
	34000 Montpellier
Telephone :	0000000000
Fax :	0000000000
Site internet :	<a href="http://www.nouvelleentreprise.fr">http://www.nouvelleentreprise.fr</a>
Description de l'entreprise :	Ma Nouvelle Entreprise
Secteur d'activité :	Développement web
Nombre de salariés :	10
Possède un local?	Oui
Chiffre d'affaires	100000

FIGURE 15 – Fiche complète d'une entreprise

### Suppression d'une entreprise

Etes-vous sûr de vouloir supprimer cet entreprise ?

Entreprise : NouvelleEntreprise

Oui

Non

FIGURE 16 – Suppression d'une entreprise

#### 4.1.2 Gestion des parcours

Jusqu'à présent l'application offrait seulement la possibilité d'ajouter des parcours. Il était ensuite impossible de consulter la fiche d'un parcours, de modifier ses informations ou même de le supprimer. Nous avons là aussi, comme pour les entreprises, mis en place une interface permettant la gestion des parcours par les administrateurs.

	Parcours	Intitulé	Existe ?
	M2 AIGLE	Architecture et ingénierie du logiciel et du web	Oui
	M2 BIO INFO	Bioinformatique	Oui
	M2 CASAR	Combinatoire, Algorithmes, Sécurité, Administration des Réseaux	Non
	M2 DECOL	Données, connaissances et langage naturel	Oui
	M2 DIWEB	Données, Information et Web	Non
	M2 GEOMATIQUE	Géomatique	Oui
	M2 GL	Génie Logiciel	Non
	M2 IC CCI	Intégration de Compétences - Compétence Complémentaire en Informatique	Non
	M2 IMAGINA	Image, Games and intelligent Agents	Oui
	M2 IPS	Informatique pour les Sciences	Oui
	M2 MOCA	Modélisation, Optimisation, Combinatoire et Algorithme	Oui
	M2 PHY INFO	Physique Informatique	Oui

[Ajouter un parcours](#)

FIGURE 17 – Interface de la gestion des parcours

#### Modification du parcours M2 AIGLE

Données du parcours

Niveau :

Acronyme :

Intitulé :

Description :

Le parcours existe-il encore? ☒ Oui ☐ Non

Responsable :

FIGURE 18 – Modification d'un parcours

Il arrive que des parcours soient amenés à disparaître car remplacés par de nouveaux parcours.

Pour gérer ces cas, nous avons implémenté une option permettant de spécifier si un parcours existe toujours ou non.

De cette manière, un parcours qui n'existe plus ne pourra plus être sélectionné sur l'application mais son historique est conservé car il n'est pas supprimé de la base de données.

### 4.1.3 Ajouts de liens vers les fiches




	Parcours	Sujet	Entreprise	Etat	Validé	Proposé	Etudiant	Tuteur
	AIGLE	Développement d'interface utilisateur mobile pour une application de géolocalisation et suivi en température	NAOCOM		10/10/2014	10/10/2014	Test Test	Eric Bourreau
	MOCA	Amélioration des Bornes Inférieures produite par le solveur de programmation mathématique LocaSolver	Innovation24		07/10/2014	07/10/2014	Jeremy Pastor	Eric Bourreau

FIGURE 19 – Exemple de listing avant ajout des liens





	Parcours	Sujet	Entreprise	Etat	Validé	Proposé	Etudiant	Tuteur
	<a href="#">M2 AIGLE</a>	Développement d'interface utilisateur mobile pour une application de géolocalisation et suivi en température	<a href="#">NAOCOM</a>		<a href="#">10/10/2014</a>	<a href="#">10/10/2014</a>	<a href="#">Test Test</a>	<a href="#">Eric Bourreau</a>
	<a href="#">M2 MOCA</a>	Amélioration des Bornes Inférieures produite par le solveur de programmation mathématique LocaSolver	<a href="#">Innovation24</a>		<a href="#">07/10/2014</a>	<a href="#">07/10/2014</a>	<a href="#">Jérémy Pastor</a>	<a href="#">Eric Bourreau</a>

FIGURE 20 – Exemple de listing après ajout des liens

Certains utilisateurs, notamment les enseignants, se sont plaints de ne pas pouvoir naviguer entre les différents listings et à travers les fiches d'informations.

Nous avons donc rajouté ses hyperliens qu'avaient omis les précédents développeurs de l'application.

De cette manière, on peut facilement accéder à toutes les informations concernant un stage, un utilisateur, une entreprise, ou un parcours à partir des listings.

Et on peut également naviguer à travers les fiches. Par exemple, accéder à la fiche du parcours d'un étudiant à partir de la fiche de ce dernier, ou encore accéder aux informations d'une entreprise dans laquelle travaille un tuteur à partir de sa fiche etc.

La navigation et la recherche d'informations devient aisé pour tout utilisateur de la plate-forme.

#### 4.1.4 Recherche par étudiant dans le listing des stages

Très attendu par le secrétariat du département informatique, nous avons implémenté une option qui permet aux administrateurs, responsables et secrétaires de trouver le stage associé à un étudiant en sélectionnant son nom dans une liste déroulante.

A noter que la liste déroulante ne contient que les étudiants de l'année scolaire courante.

Etudiant : Test Test Tous les étudiants EtudiantTest EtudiantTest Test Test Rechercher

Résultats de 0 à 1.

	Parcours	Sujet	Entreprise	Etat	Validé	Proposé	Etudiant	Tuteur
	M2 AIGLE	Développement d'interface utilisateur mobile pour une application de géolocalisation et suivi en température	NAOCOM		10/10/2014	10/10/2014	Test Test	Eric Bourreau

FIGURE 21 – Recherche d'un stage par nom de l'étudiant

De cette manière, plus besoin de parcourir le listing des stages page par page pour trouver le stage associé à un étudiant : on accède à l'information rapidement.

#### 4.2 Listing stages/utilisateurs de l' année courante

Certains listings présentaient l'inconvénient d'afficher des informations obsolètes. C'est le cas du listing des étudiants sans stages, qui listait absolument tous les étudiants qui n'ont pas de stages affectés, sans prendre en compte l'année de promotion des étudiants. Par conséquent on retrouvait des étudiants qui n'étaient plus à la faculté depuis des années... Même chose pour les listings des étudiants et des tuteurs avec leurs stages associés qui recensaient des stages ayant eu lieu il y a des années.

Pour palier à ce problème, nous avons introduit une variable globale `anneeScolaireCourante` qui est calculé en fonction de la date actuelle et change de valeur au mois d'octobre, mois durant lequel démarre une nouvelle année scolaire.

De cette manière, on a pu ajouter à nos requêtes SQL une condition sur l'année scolaire, afin qu'elles ne récupèrent seulement les étudiants et les stages de l'année courante pour établir nos listings.

Nous avons quand même laissé aux utilisateurs la possibilité de sélectionner des années scolaires passés grâce à une liste déroulante.

### 4.3 Connexion via le CAS

Le Central Authentication Service (CAS) est un système d'authentification unique (Single Sign-On) pour le web, il s'agit d'une méthode permettant à un utilisateur d'accéder à plusieurs applications informatiques ou sites web sécurisés en ne procédant qu'à une seule authentification.

L'intérêt du CAS est qu'il permet de s'authentifier sur un site web et on est alors authentifié sur tous les sites Web qui utilisent le même serveur CAS.

Il existe 2 versions du CAS : une version de base sans proxy et une version qui offre un service d'authentification proxy. Dans notre cas, nous avons utilisé un CAS avec proxy, ce qui est intéressant dans une solution multi-tier (serveur CAS, les différents clients CAS, et les librairies).

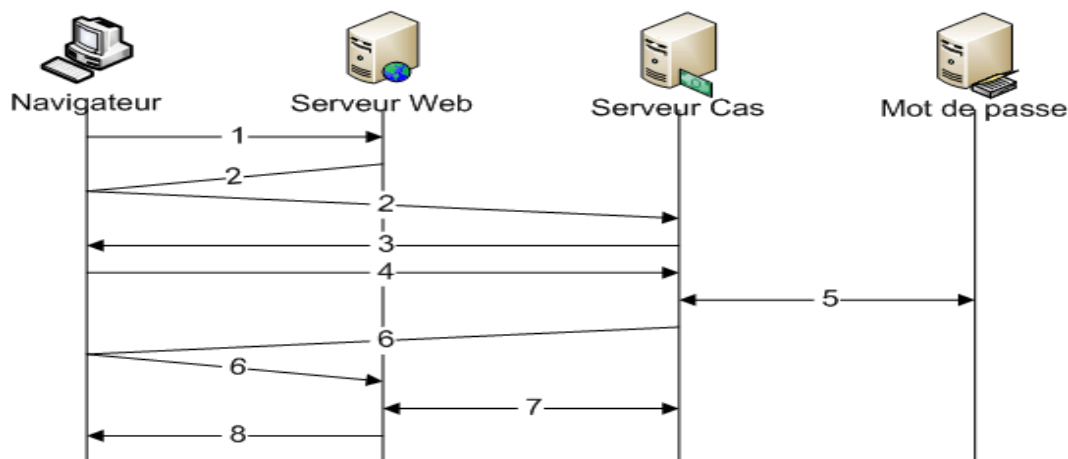


FIGURE 22 – Fonctionnement du CAS

Le fonctionnement du CAS (FIGURE 22) est le suivant :

1. Un internaute accède à une ressource web.
2. Le navigateur est redirigé vers le serveur cas
3. Le serveur envoie le formulaire d'authentification
4. Réponse de l'internaute.
5. Le serveur vérifie le couple compte / mot de passe.
6. Le serveur CAS crée un cookie de session et redirige le navigateur vers la ressource web avec un Service Ticket (ST) à utilisation unique
7. L'application valide le ST en contactant directement le serveur cas qui retourne l'identifiant de la personne.
8. L'internaute accède à la ressource demandée

Afin d'éviter la multitude des mots de passe et répondre aux souhaits des utilisateurs, nous avons proposé une connexion via le CAS en mettant en place sur la plate-forme, un système qui utilise les pré-requis à PHPCas mis en place sur l'environnement de la DSI (Direction du Système d'Information).

Le principe est relativement simple, l'utilisateur clique sur le lien "Connexion via CAS", et à l'aide de phpCas on le redirige sur la page d'authentification du CAS. L'utilisateur s'identifie alors sur le CAS, et il est redirigé sur la page d'accueil de notre site en étant connecté sur son compte.

Dans un premier temps, on charge le fichier de configuration et la librairie phpCAS, puis on initialise phpCAS. L'étape suivante consiste à forcer l'authentification de l'utilisateur sur le CAS, s'il n'est pas déjà authentifié. Ensuite, on utilise la méthode `getUser()` pour récupérer l'adresse mail de l'utilisateur authentifié. Enfin, pour connecter l'utilisateur à notre plate-forme, nous avons créé une méthode `loadFromEmail()` qui récupère toutes les données de l'utilisateur de notre base de données à partir son adresse mail que nous avons récupéré.

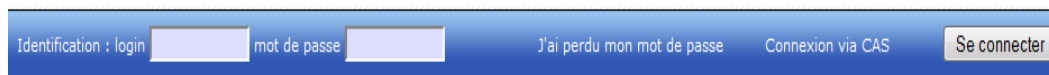


FIGURE 23 – Lien de connexion au CAS




FIGURE 24 – Page d'authentification du CAS

## 5 Vers un déploiement à tous les départements ?

GStage est actuellement propre au département informatique de la faculté des sciences. Mais il serait intéressant dans un avenir proche d'étendre ce type d'outil à tous les départements constituant la faculté des sciences. Ceci permettrait de gérer tous les aspects opérationnels concernant les stages de manière automatique et rendrait tous les traitements moins fastidieux pour tous les acteurs.

Des souhaits allant dans ce sens ont été émis par une poignée d'enseignants de la faculté des sciences. Certains ont développé avec leurs propres moyens une application du même type que GStage, tandis que d'autres ont fait appel à des informaticiens professionnels.

Les démarches pour déployer ce type de plate-forme à tous les départements ne manquent pas : nous avons notamment assisté à une réunion en présence de 3 enseignants favorables à cette extension.

Deux solutions peuvent être envisagées :

- soit on clone GStage ou une application du même type afin que chaque département ait son propre outil et le gère comme il le souhaite
- soit on fusionne les quelques sites existants et on met en place une et une seule plate-forme commune à tous départements et qui gère les stages de toute la faculté

La seconde solution est clairement la plus intéressante et la plus ambitieuse, mais c'est aussi la plus difficile à mettre en place car elle pose un certain nombre de problèmes. Notamment le fait que d'un département à l'autre, les responsables pédagogiques ont des besoins spécifiques et les stages ne sont pas gérés de la même façon.

D'après les spécialistes, pour obtenir un outil adaptable à chaque département, il faudrait un travail de 2 à 3 mois pour élaborer un cahier des charges complet avec les différents départements, suivi du travail d'un développeur pour une durée de 5 à 6 mois.

Enfin, la réalisation d'un tel projet implique l'aval de la direction de la faculté et l'investissement de la Direction du Système d'Information (DSI), sans quoi ce projet ne verra pas le jour. Or plusieurs questions restent à ce jour toujours sans réponses :

- Il y a-t-il une réelle volonté politique de la part de la faculté pour que ce projet voit le jour ?
- La DSI a-t-elle les moyens d'affecter un informaticien sur ce projet pour une durée de 6 mois ?



## 6 Bilan du projet

### 6.1 Bilan chiffré

#### 6.1.1 Sur la base de données

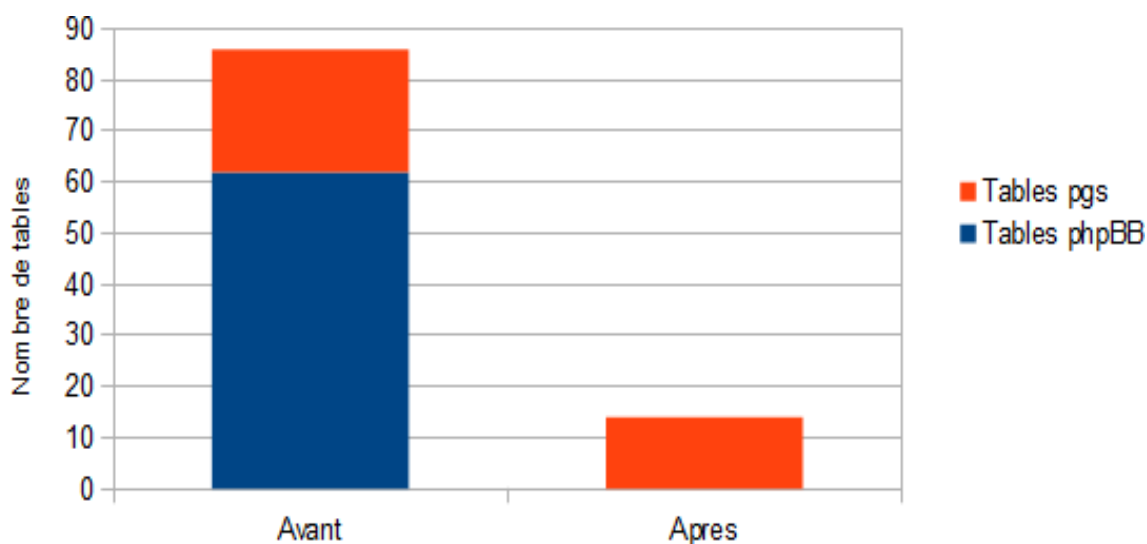


FIGURE 25 – Nombre de tables dans la base de données (avant et après)

La base de données contenait 86 tables au début du projet dont 62 rien que pour le fonctionnement du forum phpBB inutilisé.

Après notre travail de réingénierie, elle ne contient plus que 14 tables :

- les 62 tables phpBB ont été supprimées
- 10 tables propre à la plateforme (pgs) ont aussi été supprimées

#### 6.1.2 Sur le code source

Le code source de la plate-forme contenait 508 745 lignes au début du projet, il n'en contient plus que 270 636 aujourd'hui, soit une différence de 238 109 lignes de code malgré l'ajout de la librairie phpCAS (FIGURE 26).

Par conséquent la taille du projet a elle aussi chuté, passant de 29.4 Mo à 20.1 Mo, soit une diminution d'un tiers de la taille initiale (FIGURE 27).

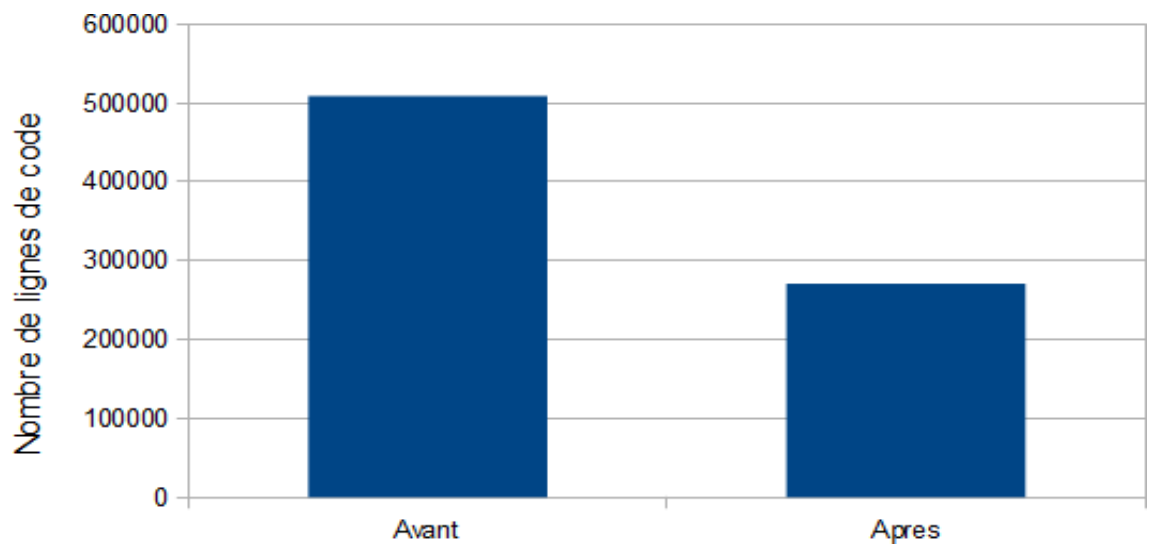


FIGURE 26 – Nombre de lignes de code (avant et après refactoring)

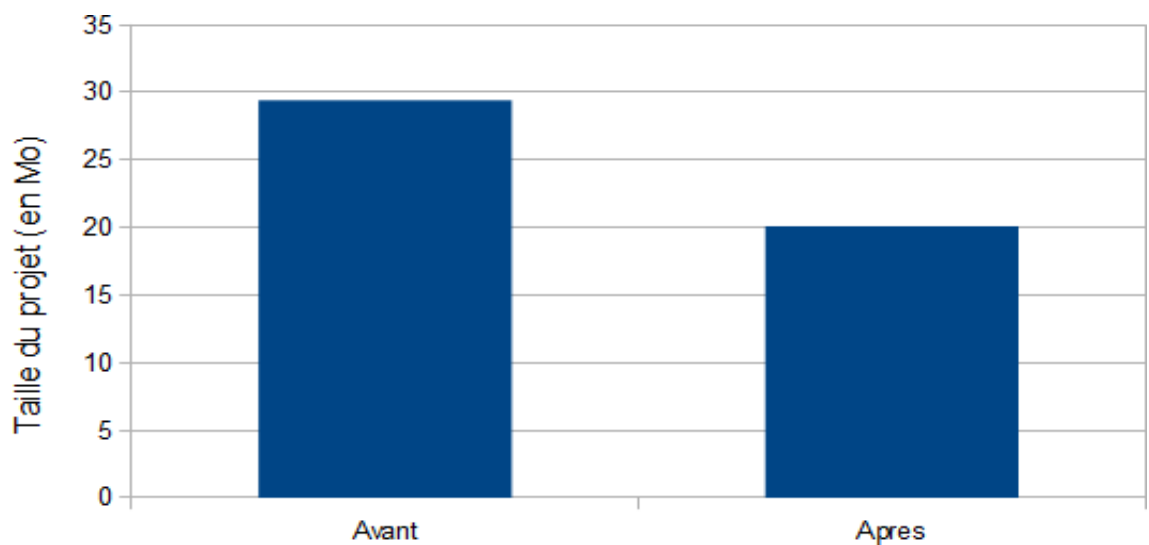


FIGURE 27 – Taille du projet (avant et après refactoring)

Ces chiffres nous permettent d'affirmer que le processus de réingénierie logicielle était réellement nécessaire sur l'application. Il a permis de nettoyer l'application, sans pour autant altérer les fonctionnalités existantes et cela a même facilité l'ajout de nouvelles.

## 6.2 Perspectives concernant la plate-forme

La possibilité de travailler sur une vraie application existante dans le cadre de notre projet a été très stimulante pour nous.

Cependant, nombre d'idées n'ont pas pu voir le jour, principalement par faute de temps, notamment en ce qui concerne l'ajout de quelques fonctionnalités :

- La plateforme GStage pourrait être liée à la plate-forme PStage, qui permet de gérer les conventions de stages. On aurait ainsi une application informatique complète de la gestion des stages comprenant l'aspect gestion (gérée par GStage) et la partie administrative (gérée par PStage). De cette manière, on pourrait par exemple générer automatiquement des conventions de stage pré-remplies avec les informations saisies sur GStage, cela faciliterait grandement la saisie administrative pour les étudiants
- De nombreux documents circulent sur GStage : rapports de stages, fiches d'évaluation, fiches de suivi, ressenti de l'entreprise sur l'étudiant, ressenti de l'étudiant sur l'entreprise etc. Tous ces documents sont actuellement stockés dans un unique dossier qui grossit en taille au fil des années. A l'avenir, il faudra donc trouver un meilleur moyen de gérer ça. Une solution consisterait à importer les notes attribués aux stages depuis Apogée, afin de créer des requêtes permettant de gérer les documents en fonction des notes obtenus. On pourrait par exemple ne conserver que les documents concernant des stages où l'étudiant a obtenu une note supérieure à 15/20.
- GStage est actuellement utilisé au sein du département informatique seulement et nous aurions aimé participer au déploiement de cette plate-forme (ou d'une application du même genre) à tous les départements de la faculté, afin que eux aussi puissent bénéficier d'une gestion informatisée des stages.
- Enfin, notre travail a consisté à remanier la plate-forme d'un point de vue informatique, mais il reste à la nettoyer d'un point de vue métier. En effet, il reste dans la base de données, des informations obsolètes, par exemple des enseignants qui ne sont plus à la faculté ou des entreprises qui ont changé d'adresse.

### 6.3 Conclusion

Ce TER a été très enrichissant, il nous a permis d'acquérir des connaissances sur la réingénierie logicielle, ses étapes de mise en oeuvre et sur l'importance de celle ci pour assurer la maintenance d'une application.

De plus, ce projet nous a permis de travailler sur une application existante, accessible sur le web et utilisé par des centaines d'internautes, comme si nous étions dans le cadre d'un vrai projet professionnel.

De ce fait, nous avons pu rencontrer et interviewer plusieurs utilisateurs de la plateforme (responsables, secrétaires, enseignants) pour qu'ils expriment leurs souhaits en terme d'améliorations à apporter sur l'application, comme si il s'agissait de vrais clients.

Quant aux objectifs initiaux, ils ont globalement été atteints :

- les bugs existants et connus ont été corrigés
- la plate-forme et sa base de données ont été nettoyés et remaniés sans altérer les fonctionnalités existantes
- de nouvelles fonctionnalités réclamés par les utilisateurs ont fait leur apparition

Maintenant que la plate-forme a été restructuré et son modèle solidifié, il serait intéressant de la déployer à d'autres départements de la faculté, ou qu'elle serve de base à un outil commun de gestion des stages de tous les étudiants de la faculté. Cependant, cela ne sera pas possible sans une véritable volonté politique et sans le feu vert de la direction et de la DSI.

## 7 Bibliographie

- Grabkina J., Sokhna K, et Adone V. (2010). « Extension de la plateforme de gestion des stages », Rapport TER M1 CCI.
- Belkebir S., Bouchemella H., et Bongo F. (2013). « Généralisation de l'utilisation d'une plateforme de gestion des stages », Rapport TER M1 IPS.
- Renaud M., Ji H., Duc Nco T.M., Mellaoui N., Yakoubi H. (2014). « Extension de la plateforme de gestion des stages du département informatique », Rapport TER M1 IPS.
- Opdyke W.F. (1992). « Refactoring Object-Oriented Frameworks », (Ph.D. thesis) University of Illinois

## 8 Annexes

### 8.1 Compte rendus des réunions

#### 8.1.1 Compte rendu de la réunion du 21/01

**Date :** 21 janvier 2015 de 11h30 à 12h30 (1 heure)

**Lieu :** Montpellier, Faculté des sciences, Bâtiment 16, salle 16.46

**Présents :** Eric Bourreau (encadrant), les 4 membres

##### 8.1.1.1 Présentation de la plate-forme et des fonctionnalités

Eric Bourreau nous a présenté la plate-forme sur laquelle nous allons travailler pour ce projet.

Il s'agit d'une plate-forme développée depuis 5 ans par plusieurs générations successives de Master 1 IPS (Informatique pour les Sciences) et permettant la gestion des stages professionnels des étudiants du Master 2 Informatique.

Elle est destinée à plusieurs types d'utilisateurs :

- les étudiants
- les entreprises
- les tuteurs (enseignants)
- les secrétaires du département informatique
- les responsables de la plate-forme ou des stages

En ce qui concerne les principales fonctionnalités du site, les entreprises peuvent saisir des offres de stage. Une fois ces dernières validées par les responsables, les étudiants peuvent les consulter au moment de leur recherche de stage (entre novembre et mars) et même y ajouter des commentaires (après un échec lors d'un entretien par exemple).

De février à août, elle permet aux tuteurs s'assurer le suivi de leurs étudiants (maximum 5 étudiants par tuteur).

Enfin, la plate-forme permet de faciliter l'organisation des soutenances qui ont lieu durant le mois de septembre et d'assurer la présence de chacun (l'étudiant, le tuteur, le représentant de l'entreprise et l'enseignant extérieur).

##### 8.1.1.2 Correction des bugs

La plate-forme renferme quelques bugs connus, par exemple :

- un étudiant peut consulter des stages que le responsable ne voit pas
- l'affichage du tuteur pour un stage ne correspond pas toujours à celui qui a été spécifié initialement (cela pourrait venir d'un décalage dans les index)

Il serait donc judicieux de les résoudre au cours de ce TER en s'inspirant du modèle du logiciel de gestion des bugs, Bugzilla, à savoir : descriptif de bug par l'utilisateur, analyse fonctionnelle, proposition de correction, puis cas de test et mise en production, et enfin validation par l'utilisateur.

#### **8.1.1.3 Ajouts de nouvelles fonctionnalités**

Ce TER peut être aussi l'occasion d'ajouter de nouvelles fonctionnalités à la plate-forme.

En effet, certaines fonctionnalités ont été omises lors des précédents développements et il serait bien d'en rendre certaines possibles, par exemple :

- l'ajout, la suppression et la modification de toutes les entités via une interface graphique afin d'éviter au responsable de la plate-forme d'avoir à traiter avec phpMyAdmin
- lors de l'ajout d'une entité, vérifier si elle n'existe pas déjà dans la base de données afin d'éviter les redondances
- permettre aux étudiants de se connecter au site avec leurs identifiants de l'Environnement Numérique de Travail (ENT) de l'UM2, et non plus avec des identifiants propres à la plate-forme
- envoyer un mail au responsable dès lors qu'une offre de stage a été ajouté par une entreprise afin qu'il la valide
- lier la plate-forme avec PStage afin de générer automatiquement les conventions de stage avec des formulaires pré-remplis

De plus, il serait pertinent de contacter Michelle Joab, la responsable des stages, ainsi que Michelle Auge, la secrétaire du Master Informatique pour leur demander d'établir une liste de souhaits en terme de nouvelles fonctionnalités.

#### **8.1.1.4 "Refactoring" et intégration de patrons de conception**

La plate-forme est développée depuis 5 ans par plusieurs générations d'étudiants, cela implique une dégradation de la lisibilité du code d'année en année. Il est donc de plus en plus difficile d'assurer sa maintenance et de lui ajouter des fonctionnalités.

Notre principal objectif au cours de ce TER sera d'améliorer la lisibilité du code afin de simplifier la maintenance de la plate-forme. De plus, une uniformisation propre du code source permettrait de déployer la plate-forme à d'autres départements de la faculté, notamment à celui de Physique, afin que le département Informatique n'ait plus à gérer les stages de Master 2 Physique Informatique.

Il faudra également analyser et revoir la modélisation UML qui a l'heure actuelle contient quelques incohérences. Par exemple, la secrétaire n'a pas les droits suffisants pour faire son travail, du coup elle a le rôle de responsable sur le site.

Ce TER sera aussi l'occasion de mettre en oeuvre ce que nous avons acquis en suivant le module d'Ingénierie Logicielle du Semestre 1. Et notamment l'intégration de patrons de conception (design patterns) dans le but d'optimiser le code.

Enfin, dans les années passées, des tentatives ont été faites pour lier la plate-forme à un forum phpBB et à une application Facebook. Ces tentatives ont depuis été avortées, et il faudra donc supprimer le code mort, qui n'a plus de raison d'être, car il n'est plus appelé et rend la lecture du code source plus complexe.

#### **8.1.1.5 Planification des réunions**

Eric Bourreau propose de se rencontrer une fois par semaine, le mercredi matin, au bâtiment 16 de la faculté.

Un compte rendu devra être rédigé pour chaque réunion et rendu la semaine suivante.



### 8.1.2 Compte rendu de la réunion du 04/02

**Date :** 4 février 2015 de 11h30 à 12h30 (1 heure)

**Lieu :** Montpellier, Faculté des sciences, Bâtiment 16

**Présents :** Eric Bourreau (encadrant), les 4 membres

- évocation de CRUD (Create, Read, Update, Delete), méthode permettant à l'utilisateur la création, la lecture, la mise à jour et la suppression de toutes les entités et qu'il faudrait appliquer à la plate-forme
- envoyer un mail au responsable lorsqu'un stage a été validé par l'étudiant et relancer ceux qui n'ont rien (préparer le mailing). Plus généralement, notifier l'acteur du prochain état.
- laisser la possibilité aux utilisateurs si ils veulent recevoir des notifications ou pas.
- interviewer David Cassagne pour voir comment se passe le processus de gestion des stages pour les M2 Physique et Physique-Informatique, et ainsi faciliter la gestion des stages de chaque spécialité. Faut-il une plate-forme indépendante ? Dissocier les bases de données ?
- lorsque le stage est validé pour l'étudiant, celui-ci reçoit un mauvais mail, il faut corriger ce bug en annulant ce mail
- créer un Git afin de gérer le signalement des bugs (issues)
- remise de 3 rapports des années précédentes
  - Grabkina J., Sokhna K., et Adone V. (2010). « Extension de la plateforme de gestion des stages », Rapport TER M1 CCI.
  - Belkebir S., Bouchemella H., et Bongo F. (2013). « Généralisation de l'utilisation d'une plateforme de gestion des stages », Rapport TER M1 IPS.
  - Renaud M., Ji H., Duc Nco T.M., Mellaoui N., Yakoubi H. (2014). « Extension de la plateforme de gestion des stages du département informatique », Rapport TER M1 IPS.
- remise du code de la plateforme et de la base
- interviewer Clémentine Nébut et Marianne Huchard, afin d'évoquer des méthodes de "refactoring" et l'existence d'un outil qui permettrait l'abstraction des classes à partir du code existant

- dans un souci d’archivage, le responsable peut uploader des documents sur une fiche de stage : rapport de stage, fiche d’évaluation, fiche de suivi, ressenti de l’entreprise sur l’étudiant, ressenti de l’étudiant sur l’entreprise, et des annexes. Cependant, les documents sont stockés dans un dossier à côté du PHP, et la taille de ce dernier devient de plus en plus conséquente, il faudrait donc trouver un meilleur moyen de gérer ça
- faire une analyse formelle des concepts (AFC) afin de dégager les fonctionnalités nécessaires
- importer les fichiers notés depuis Apogée sur la plate-forme de stage
- créer des requêtes qui vont permettre de gérer les rapports en fonction des notes et donner la possibilité de filtrer les rapports de stage en fonction des notes (on ne veut voir que les rapports de stage dont les notes sont supérieures à 16)
- notification lorsqu’un stage a été accepté par l’entreprise (mail par exemple)
- si cela est nécessaire, faire une scission intelligente des tables de la base de données qui ont trop d’attributs
- faire une décomposition des parties importantes et avoir une idée claire des tâches à effectuer
- commencer à corriger les bugs rapidement
- commencer à faire des propositions d’amélioration

### 8.1.3 Compte rendu de la réunion du 10/02

**Date :** 10 février 2015 de 13h45 à 14h15 (30 min)

**Lieu :** Montpellier, Faculté des sciences, Bât 16, secrétariat master info.

**Présents :** Michelle Auge (secrétaire), Ismail Belhaine, Jérémy Pastor

- problème lors de la saisie d'une offre de stage
  - "Saisie complète" : ne fonctionne pas, l'offre de stage n'apparaît pas dans les listings et n'est donc visible par aucun utilisateur
  - "Saisie rapide" : l'offre de stage est visible, mais peu importe le nom du tuteur qui a été saisi, c'est celui de "Yannick Auffrey" qui est affiché à chaque fois et il est impossible de le modifier.
- cacher ou supprimer le lien permettant de télécharger les conventions de stage pour tous les utilisateurs, afin de bien faire la distinction entre GStage (notre plateforme) et PStage (gestion des conventions)
- permettre une recherche par nom de l'étudiant dans le listing de stages
- retravailler les fiches téléchargeables (notamment changer les logos de l'UM2 par ceux de l'Université de Montpellier)

#### 8.1.4 Compte rendu de la réunion du 26/02

**Date :** 26 février 2015 de 16H00 à 17h00 (1 heure)

**Lieu :** Montpellier, Faculté des sciences, Bât 16, secrétariat master info.

**Présents :** Michelle Joab (responsable des stages), les 4 membres

- ne pas permettre d'affecter un étudiant a un stage qui n'est pas encore validé (ajouter un feedback)
- afficher dans les listings seulement les étudiants de l'année courante
- supprimer les parcours qui n'existent plus
- ajouter un attribut parcours (AIGLE, IMAGINA...) et le niveau (L3, M1, M2) et dans la table Utilisateur
- ajouter le niveau requis lors de la saisie d'un stage
- ajouter des informations sur la personne qui a saisi le stage (role, parcours)
- ajouter des informations sur une entreprise : secteur d'activité, idée du nombre de salariés, local ou pas, chiffres d'affaires, numero de SIRET
- ajouter des informations sur le tuteur de l'entreprise (compétences, diplômes...)
- Bug : parfois, le responsable ne voit pas certains stages saisies par des étudiants
- listing des étudiants (de l'année courante!) qui n'ont toujours pas de stages ou de tuteurs
- lorsqu'un étudiant saisit un stage, envoyer un mail au responsable, de même, envoyer un mail a l'étudiant lorsqu'il est affecté à un stage
- recherche pas nom d'étudiant dans le listing des stages
- permettre a un professeur de voir et de montrer son intérêt pour être le tuteur d'un stage qui a déjà été choisi par un autre tuteur
- supprimer le lien vers la mauvaise convention de stage
- pour les offres de stages génériques, donner la possibilité de dupliquer l'offre afin de le mettre à disposition à d'autres étudiants
- pouvoir configurer et mettre comme valeur par défaut le salaire minimum d'un stagiaire

### 8.1.5 Compte rendu des réunions du 11/03

**Date :** 11 mars 2015 de 11H30 à 13h00 (1 heure 30)

**Lieu :** Montpellier, Faculté des sciences, Bâtiment 16.

**Présents :** Eric Bourreau (tuteur), les 4 membres

- affectation du tuteur sur le GitLab en tant que "Guest" afin qu'il puisse poster des "issues"
- Présentation des différentes plateformes :
  - plateforme de pré-production hébergé à l'UM2 et géré par le SIF (stagefds.proj.info-ufr.univ-montp2.fr)
  - vraie plateforme (en production) héberge également au sein de l'UM2 mais géré par la DSI
- supprimer tout ce qui touche à phpBB ou Facebook
- faire le modèle logique des données et lui apporter des modifications
- la modification des tables de la base de données, va avoir une incidence sur les attributs et les méthodes des classes dans le PHP, par conséquent faire une analyse d'impact à chaque changement
- l'idée que ce sont les entreprises qui choisissent directement leur stagiaire a été abandonnée, donc les tables competencesinfo, languesparlees n'ont plus lieu d'être
- optimiser la table utilisateur, et éventuellement la diviser en fonction des rôles (etud, ent, tut, resp etc.)
- permettre de n'afficher que les parcours actuels, mais ne pas supprimer les anciens dans un souci d'archivage
- faire un diagramme de cas d'utilisation afin de revoir les actions que peuvent effectuer chaque utilisateur
- empêcher l'import d'étudiants sans l'année scolaire
- lors de l'import d'étudiants, si il y'a des étudiants redoublants (c'est à dire existants dans la base de données), faire un UPDATE sur l'année scolaire

**Date :** 11 mars 2015 de 16H00 à 17h00 (1 heure)

**Lieu :** Montpellier, Faculté des sciences, Bât 16.

**Présents :** Marianne Huchard, les 4 membres

- Le refactoring est un domaine très large qui comprend de nombreuses approches : analyse d'impact, normalisation de la base de données, application de design patterns, factorisation du modèle...
- consulter la thèse de Opdyke qui est un précurseur du refactoring ainsi que le livre de Fowler et Beck "Refactoring : Improving the Design of Existing Code"
- Eventuellement, demandez conseil à Pascal Poncelet pour la normalisation de la base de données
- Pistes et termes à explorer pour l'analyse d'impact : graphe d'appel, graphe d'accès, graphe d'exécution (qui permet aussi de détecter le code mort), modèle analyse statique...
- Consultez le livre "Design Pattern" de Gamma pour avoir des idées de design pattern à appliquer.
- Rechercher un plugin PHP pour Eclipse afin d'utiliser ces outils
- Formal Concept Analysis (FCA) est une méthode pour analyser et manipuler les données (des objets avec des attributs) avec une approche formelle (dans un tableau binaire). Le but est de regrouper des entités qui ont des caractéristiques communes (factorisation)
- Relationnal Concept Analysis (RCA) étend le principe de FCA en prenant en compte les catégories d'objets et les associations entre objets.

### 8.1.6 Compte rendu de la réunion du 25/03

**Date :** 25 mars 2015 de 11h15 à 12h15 (1 heure)

**Lieu :** Montpellier, Faculté des sciences, Bât 16.

**Présents :** Eric Bourreau, Yann Essiane Medja, Jérémy Pastor

- Proposition du nouveau MPD :
  - Pas besoin de l'adresse pour un utilisateur (mais la conserver pour une entreprise)
  - Mettre le niveau (M1, M2 etc...) dans la table Parcours au lieu de Etudiant
  - conserver le rôle comme un attribut : dans le cas ou un utilisateur a plusieurs rôles (ce qui est très rare) celui ci doit créer un compte pour chaque rôle
- Bugs :
  - Résoudre les problèmes d'encodage
  - Dans display-entreprise : double affichage du prénom et non affichage du nom (ex : Michel Michel)
  - Dans display-entreprise : mauvais affichage du responsable administratif quelquefois
- Pour chaque rôle, faire la liste des permissions sous forme de matrice. Et faire proposer des modifications (ex : supprimer "Etudiants disponibles" pour le rôle ENT)

### 8.1.7 Compte rendu de la réunion du 14/04

**Date :** 14 avril 2015 de 11H à 12h (1 heure)

**Lieu :** Montpellier, Faculté des sciences, Bât 16.

**Présents :** Eric Bourreau, Arnaud Virazel, Agnes Fichard, Khaled Ismail Belhaine

- dispatcher l'application de GStage par département (cloner)
- l'application doit être gérée par la fac et non par un département
- mettre en évidence une application intégrée qui implique les autres applications qui existent et ce n'est pas évident avec certains problèmes politiques et surtout avec les besoins spécifiques de chaque département
- il faut définir un cahier de charge complet avec les différents départements pour tout retenir
- comprendre les besoins de chaque responsable pédagogique
- trouver un équilibre entre les coûts de temps et d'argent, environ 6 mois développement par un informaticien pour obtenir un outil adaptable avec différentes applications intégrées.

### 8.1.8 Compte rendu de la réunion du 20/05

**Date :** 20 mai 2015 de 16H à 18h (2 heures)

**Lieu :** Montpellier, Faculté des sciences, Bât 16.

**Présents :** Eric Bourreau, Yann Essiane, Jérémie Pastor, Rabie Houti

- ajouter les hyperliens où l'on estime que c'est nécessaire de façon à naviguer à travers les listings et les fiches des utilisateurs, stages ou entreprises
- avec la fusion de l'UM2 et l'UM3, le CAS de l'université de Montpellier a changé, il faut donc le retester.
- Bug : un enseignant qui a encadré 5 stages dans le passé ne peut plus encadrer de stage
- ajouter la possibilité de faire une sélection par année sur les listings