

Rapport De Stage

R. Zhu et J. Deletrez

18 mai 2008

Introduction

Le rapport que vous vous apprêtez à lire intervient dans le cadre de notre première année de Master Informatique. Cette formation se déroule au sein de la faculté des Sciences Jean Perrin, localisée à Lens.

Cette formation nous offre la chance de pouvoir intégrer le CRIL (Centre de Recherche Informatique de Lens) pour y faire un stage aux côtés d'enseignants chercheurs.

Dans ce rapport, vous allez découvrir la mission réalisée au CRIL. Nous apporterons des informations concernant le déroulement de cette mission et les problèmes survenus.

Table des matières

1	Présentation et explications	7
1.1	Présentation	7
1.1.1	Les services	7
1.1.2	Les utilisateurs de l'application	8
1.1.3	La gestion de l'application	8
1.2	Plus de détails	8
1.2.1	Les besoins des étudiants	8
1.2.2	Les besoins des professeurs	9
1.3	Comment réaliser ce projet	9
2	La base de donnée	11
2.1	Le Modèle Conceptuel de Données (MCD)	11
2.2	Le Modèle Logique de Données MLD	13
3	Les différents utilisateurs et leur rôle	15
3.1	Les étudiants	15
3.2	Les professeurs et professeurs responsables	16
3.2.1	Les professeurs	16
3.2.2	Les responsables	16
3.3	L'administrateur	17
4	La démarche adoptée	19
4.1	La répartition du travail	19
4.2	Les langages et outils utilisés	19
4.2.1	PHP	19
4.2.2	PostgreSQL	20
4.2.3	CSS / javascript	20
4.3	Les choix effectués	20
4.3.1	Le système d'identification	20
4.3.2	Les sessions, classes, interfaces et sérialisation PHP	21
4.3.3	JavaScript	21

TABLE DES MATIÈRES

4.3.4	CSS	21
5	Les détails de l'application	23
5.1	Les classes et l'interface	23
5.1.1	Les classes <i>personne</i> , <i>etudiant</i> et <i>professeur</i>	23
5.1.2	L'interface <i>database_interface</i> et la classe <i>postgresql</i>	24
5.2	Les pages importantes	24
5.2.1	La page <i>entete</i>	24
5.2.2	La page <i>type_accueil</i>	24
6	Les problèmes rencontrés	27
6.1	Base de données	27
6.2	Interface web	27
6.3	Pas si simple de répartir le travail	28
6.4	La barrière de la langue	28
6.5	Réussir le projet	28

Chapitre 1

Présentation et explications

Le sujet du stage porte sur le développement d'une application web permettant de gérer les stages au sein de la faculté. Avant de poursuivre la présentation de cette application, il est judicieux de s'intéresser aux raisons qui ont motivé son développement. Cette étude permettra de mieux la comprendre.

1.1 Présentation

Ce projet fait suite à une étude faite au premier semestre en Génie Logiciel sur un outil permettant la communication entre étudiants et professeurs. Il était notamment question de fournir des services au niveau des Travaux d'Etude et de Recherche (TER) et des emplois du temps avec une gestion des stages de TER du Master 1 actuellement, mais devant pouvoir s'adapter aux stages de Master 2 (professionnel et recherche) ainsi qu'à d'éventuels futurs stages de Licence 3. De plus, nous voulons mémoriser toutes les informations sur les stages année par année, d'où le besoin de gérer les étudiants par rapport à leur promotion et à l'année universitaire de leur stage.

L'application vous a été demandée pour pouvoir gérer les stages de ter du master 1 actuellement, mais doit pouvoir s'adapter aux stages de master 2 (pro ou recherche), ou des futurs stages de L3. De plus, on veut mémoriser toutes les informations sur les stages année par année.

1.1.1 Les services

Les services liés aux TER sont :

De permettre à un étudiant de déposer un sujet de stage qu'une entreprise lui a proposé en vue de le faire valider par l'enseignant responsable.

CHAPITRE 1. PRÉSENTATION ET EXPLICATIONS

De permettre aux étudiants d'accéder à l'ensemble des sujets proposés en laboratoire.

De permettre aux enseignants de gérer les affectations des sujets de TER aux étudiants.

Le service lié aux emplois du temps est essentiellement l'accessibilité de ceux-ci par les étudiants et le corps professoral.

En ce qui concerne les anciens étudiants, le service est de fournir un outil pour favoriser la constitution d'un réseau entre anciens et nouveaux étudiants du Master de Lens.

1.1.2 Les utilisateurs de l'application

Pour créer cet outil, il fallait donc mettre en place les différents acteurs de l'application. Il y a donc clairement des étudiants et des professeurs. Mais il faut ajouter aussi un responsable par promotion d'étudiant. Le responsable est un professeur en général. Il faut enfin ajouter un administrateur pour gérer en détail toutes ces promotions ainsi que les droits de chacun.

1.1.3 La gestion de l'application

Compte tenu de ces différents acteurs, l'application a été muni d'un système d'identification. Ainsi, ils ont tous des pages personnalisées en fonction de leurs besoins. Par exemple, les professeurs peuvent consulter tous les emplois du temps de toutes les promotions d'étudiants alors que les étudiants eux ne peuvent voir que leurs emplois du temps pour l'année en cours.

1.2 Plus de détails

Pour une meilleure compréhension de l'application, nous allons détailler maintenant les besoins de chaque utilisateur, hormis l'administrateur qui est un acteur "caché" de l'application.

1.2.1 Les besoins des étudiants

Pour gérer les TER des étudiants nous avons besoin qu'il puissent déposer un sujet de TER qu'il souhaite faire dans une entreprise (appelé sujet entreprise). Ce sujet a besoin d'être validé par le responsable de sa promotion pour que l'étudiant en fasse son stage. Si son responsable trouve que son sujet n'est pas dans ses critères d'acceptation, il a le droit de le refuser et

1.3. COMMENT RÉALISER CE PROJET

l'étudiant peut alors présenter un autre sujet ou modifier son précédent pour qu'il entre dans les critères de validation.

Pour consulter leurs emplois du temps (un par semestre), ceux-ci doivent être présent dans l'application. Il faut donc une page web où l'étudiant peut consulter son emploi du temps en fonction du semestre de l'année.

1.2.2 Les besoins des professeurs

Pour les TER, les professeurs peuvent présenter des sujets qui seront fait dans un laboratoire informatique, comme par exemple le CRIL (Centre de Recherche Informatique de Lens). Ainsi les professeurs doivent pouvoir proposer des sujets qui seront fait en laboratoire, ces sujets ont aussi besoin d'être valider par le responsable de la promotion à qui est destinée le sujet. Cependant, étant donné la communication entre les professeurs, les sujets laboratoires postés sont considérés directement comme des stages potentiels. Il se peut fort bien néanmoins qu'un sujet d'un professeur n'ait pas de stagiaire. Pour cela, les professeurs ont besoins, dans l'application, d'une page où ils peuvent poster leurs sujets. Un professeur peut poster plusieurs sujets ou ils peuvent être plusieurs pour un seul sujet.

En ce qui concerne les emplois du temps, les professeurs enseignent souvent à plusieurs promotions d'étudiants, ils ont donc besoin d'avoir accès aux divers emplois du temps. Ils ne sont pas limités aux emplois du temps des promotions auxquelles ils ont des enseignements mais ils peuvent avoir accès à tous les emplois du temps.

1.3 Comment réaliser ce projet

Pour rendre cet outil disponible à tous, la création d'une interface web s'imposait d'elle même. Cette interface couplée à une base de donnée nous permet de gérer tous les besoins des acteurs de l'application et toutes les fonctionnalités requises par le sujet.

De plus, grâce à cette implémentation interface web/base de donnée, l'application peut être maintenue à distance et par plusieurs personnes. Elle peut très bien être utilisée intranet au départ, c'est à dire seulement au sein de la faculté et étendue à l'internet ensuite si le besoins s'en fait sentir et que les contraintes techniques le permettent.

Enfin, cette implémentation permet un site web entièrement dynamique.

CHAPITRE 1. PRÉSENTATION ET EXPLICATIONS

Chapitre 2

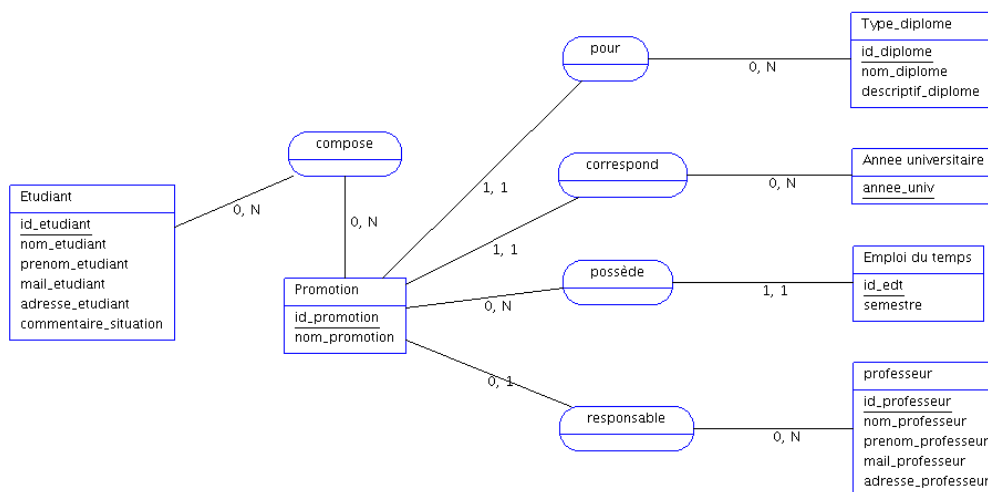
La base de donnée

Dans cette partie, nous allons vous présenter le modèle conceptuel de données auquel nous sommes arrivés pour cet outil ainsi que le modèle logique de données associé. C'est à partir de ces modèles que nous avons créé la base de données. Bien sûr chaque partie du modèle conceptuel de données a été morcellé pour une compréhension optimale.

2.1 Le Modèle Conceptuel de Données (MCD)

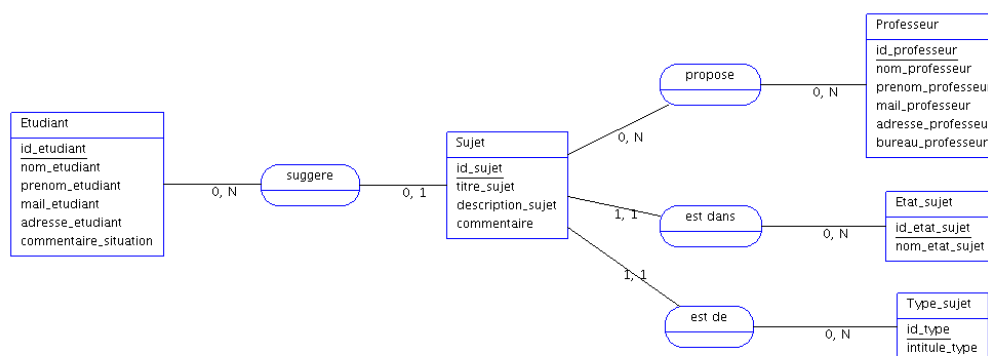
Une promotion est composée d'étudiants, d'un type de diplôme (général) : 'exemple : Licence 3 Mathématiques Informatiques, Master 1 Informatique, Master 2 Informatique', d'un emploi du temps par semestre, et ceci une certaine année universitaire (l'année universitaire a été séparée de la promotion par soucis d'écriture principalement).

CHAPITRE 2. LA BASE DE DONNÉE



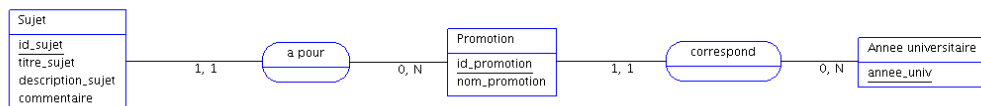
Un sujet est soit proposé par un ou plusieurs professeurs (le type du sujet sera 'Labo') soit l'étudiant suggère un sujet (le type de ce sujet sera 'entreprise'). Un sujet devra être validé par le responsable (l'état du sujet sera disponible : refusé, accepter, en cours de négociation pour acceptation,...).

Un sujet peut être modifié tant que le responsable ne l'aura pas consulté. Si le sujet a besoin d'être modifié et qu'il a déjà été consulté par le responsable, il faudra directement demander la modification au responsable.



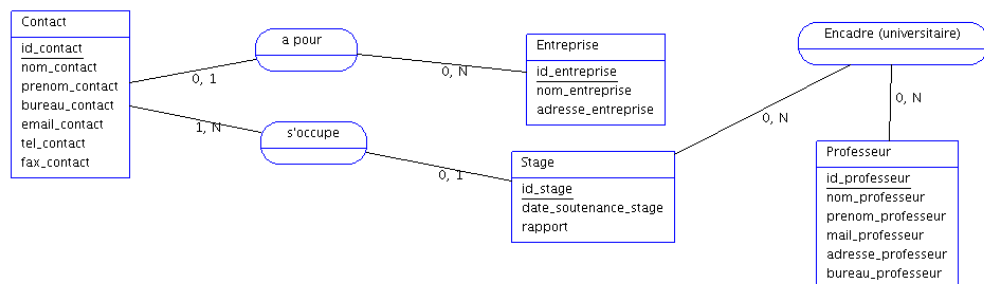
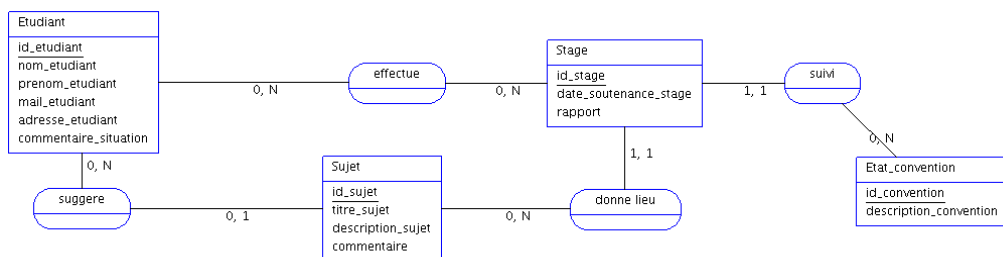
Un sujet ne pourra concerner qu'une seule promotion, une seule année universitaire.

2.2. LE MODÈLE LOGIQUE DE DONNÉES MLD



Lorsqu'un sujet a été refusé par le responsable, il est visible par tous les étudiants de la promotion. Pour une année et un semestre donnés, un étudiant ne peut avoir qu'un seul sujet accepté qui devient son stage.

Dès qu'il est accepté, il sera considéré comme un stage et une convention sera faite avec la possibilité pour l'étudiant de voir l'état dans lequel elle se trouve (ex : signé par le responsable, partie à Arras, revenue d'Arras, signé par l'étudiant,...). Quelque soit le type de stage, il sera encadré à la faculté par un ou plusieurs professeur(s). Dans le cas d'un stage en entreprise, le stagiaire sera suivi par une personne de l'entreprise qui sera son contact.



2.2 Le Modèle Logique de Données MLD

etudiant(**id_etudiant**, nom_etudiant, prenom_etudiant, numero_etudiant, adresse_etudiant, mail_etudiant)

professeur(**id_professeur**, nom_professeur, prenom_professeur, adresse_professeur, mail_professeur, bureau_professeur, *id_promotion*)

sujet(**id_sujet**, titre_sujet, description_sujet) commentaire, *id_promotion*, *id_type_sujet*,

CHAPITRE 2. LA BASE DE DONNÉE

id_etat_sujet, id_contact, id_etudiant)
entreprise(**id_entreprise**, nom_entreprise, adresse_entreprise)
contact(**id_contact**, nom_contact, prenom_contact, email_contact, bureau_contact, tel_contact, fax_contact, *id_entreprise*)
promotion(**id_promotion**, nom_promotion, *id_diplome*, *annee_univ*)
type_diplome(**id_type_diplome**, nom_diplome, descriptif_diplome)
emploi_du_temps(**id_edt**, semestre, *id_promotion*)
annee_universitaire(**annee_univ**)
stage(**id_stage**, date_soutenance_stage, rapport, nb_etudiant, *id_sujet*, *id_etat_convention*)

etat_convention(**id_convention**, description_convention)
type_sujet(**id_type_sujet**, intitule_type_sujet)
etat_sujet(**id_etat_sujet**, nom_etat_sujet)
encadre_stage(*id_stage*, *id_professeur*)
professeur_propose_sujet(*id_professeur*, *id_sujet*)
etudiant_compose_promotion(*id_etudiant*, *id_promotion*)
etudiant_effectue_stage(*id_etudiant*, *id_stage*)
inscrit(**login**, **password**, type_util, id)

légende :

- **Clé(s) primaire(s)**
- *Clé(s) étrangère(s)*

Remarque :

La table *inscrit* ne figure pas dans le MCD, ceci est tout à fait normal et pour plus d'information je vous renvoie au chapitre textitLa démarche adoptée et au point 4.3.1 sur le système d'identification des utilisateurs.

Chapitre 3

Les différents utilisateurs et leur rôle

Dans cette partie, nous allons vous expliquer les différentes possibilités qu'offre l'application en fonction de chaque catégorie d'utilisateur. Nous rappelons qu'il y a, comme acteurs, les étudiants, les professeurs, les professeurs responsable et l'administrateur.

3.1 Les étudiants

Les étudiants peuvent, dans cette application :

suggérer un sujet qu'ils souhaitent faire dans une entreprise pour qu'il soit validé par le responsable de sa promotion.

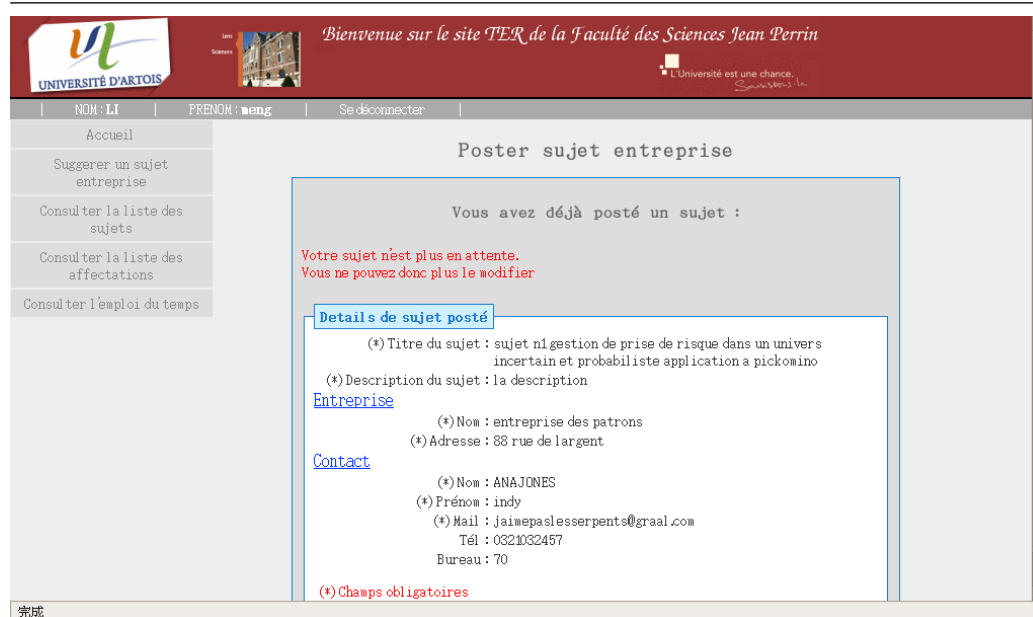
Lorsque la liste des sujets laboratoires de sa promotion sera disponible, Ils peuvent consulter cette liste et créer la liste de ses préférences et ainsi l'envoyer à leur responsable.

Ils peuvent aussi consulter, lorsque leur responsable l'aura décidé, la liste des affectations de chaque étudiant à tel ou tel stage.

Ils peuvent enfin consulter leur emploi du temps en ligne lorsqu'il est disponible.

Voici un exemple pour illustrer une des capacités de l'application pour les étudiants.

CHAPITRE 3. LES DIFFÉRENTS UTILISATEURS ET LEUR RÔLE



3.2 Les professeurs et professeurs responsables

Les professeurs et professeurs responsables ont des fonctionnalités en commun mais les responsables ont des possibilités, dans cet outil, que les professeurs standard n'ont pas. Nous vous les présentons.

3.2.1 Les professeurs

Les professeurs peuvent, dans cette application :

Modifier son mot de passe.

Consulter tous les emplois du temps lorsque il sera disponible.

Proposer des sujets qui se dérouleront dans un laboratoire du type du CRIL.

Consulter les détails des listes des affectations des étudiants aux stage à n'importe quel moment.

Consulter la liste des promotions qui ont une liste des sujets laboratoires disponible.

3.2.2 Les responsables

Les responsables, en plus des fonctions des professeurs, peuvent exécuter les opérations suivantes :

Rendre la liste des sujets laboratoires disponible aux étudiants.

3.3. L'ADMINISTRATEUR

Consulter les détails des sujets entreprises déposés.

Affecter des sujets laboratoires aux étudiants.

Valider des sujets entreprises déposés par les étudiants.

Voici un exemple pour illustrer une des capacités de l'application pour les professeurs non responsable.

Bienvenue sur le site TER de la Faculté des Sciences Jean Perrin

UNIVERSITÉ D'ARTOIS

NOU : NOMPROF1 | PRENOU : prenomprof1 | Change Mot de passe | Se déconnecter

Page d'accueil professeur

Les promotions qui ont la liste disponible

promotion 1

Les sujets que vous avez proposés : sujet n1 implementation d'un solveur SAT de type CDCI

Détails de ce sujet laboratoire

Titre du sujet : sujet n1 implementation d'un solveur SAT de type CDCI
Encadrant : NOMPROF1 prenomprof1
Promotion : promotion 1
Nombre d'étudiant(s) : 2
Année universitaire : 2007- 2008
Description du sujet :
la description

Consultation des étudiants de promotion 1 associés à ce sujet

les étudiants affectés

FFF ggg

les étudiants non affectés

MR SANS sujet
BBB abc
DDD abc
GGG abc

3.3 L'administrateur

L'administrateur peut, dans cette application, gérer toutes les insertions ou les mises à jour dans la base de données depuis le site. Il peut enregistrer un étudiant ou un professeur. Il peut créer des promotions et leurs associer des étudiants ainsi qu'un responsable. Il n'y a qu'un seul responsable par promotion. Il a la tâche aussi d'insérer les emplois du temps de chaque promotion. Il peut modifier son mot de passe aussi.

CHAPITRE 3. LES DIFFÉRENTS UTILISATEURS ET LEUR RÔLE

Voici un exemple pour illustrer une des capacités de l'application pour l'administrateur.

The screenshot shows a web application interface for 'Gestion des professeurs' (Professor Management) on the 'TER' site of the Faculty of Sciences Jean Perrin at the University of Artois. The header includes the university logo, a navigation menu with 'NOM : NOMADMIN' and 'PRENOM : prenomadmin', and a welcome message. A sidebar on the left lists administrative actions like 'Accueil', 'Enregistrer/Modifier un nouvel étudiant', and 'Enregistrer/Modifier un nouveau professeur'. The main content area features a 'Recherche rapide' search bar and a 'Renseignements professeur' form with fields for Name, First Name, Address, Email (pre-filled with '0cril.univ-artois.fr'), and Office. A red note indicates that fields with an asterisk are mandatory, and an 'Enregistrer' button is at the bottom.

Chapitre 4

La démarche adoptée

4.1 La répartition du travail

La partie base de donnée a été faite en commun car c'était délicat de séparer cette partie en deux.

La partie interface web quant à elle a été séparé en deux pour faciliter le travail et permettre de travailler plus rapidement et plus efficacement. Comme le site comporte trois parties plus ou moins distinctes : administrateur, professeur/responsable et enfin étudiant, nous nous sommes répartis les tâches. Un de nous deux a fait la partie administrateur, l'autre la partie professeur/responsable. Quant à la partie étudiant, elle a été faite en commun en réutilisant différentes fonctions disponibles dans la partie professeur et en ajoutant les parties qui nous manquées.

4.2 Les langages et outils utilisés

Pour programmer cet outil nous avons utilisé le PHP, de la CSS et du javascript. Un éditeur de texte classique du type de "gedit" suffit emplement pour programmer dans ces langages. Comme PHP est un langage interprété, il n'y a donc pas besoin de compilateur. Pour l'exécution du code, le serveur gérant PHP5 a été utilisé et Mozilla firefox a été choisi l'affichage HTML.

4.2.1 PHP

Les bases du langage PHP ont été vues au premier semestre de Master. Il a fallu néanmoins parfaire notre connaissance de ce langage puisque certains de nos besoins nous ont conduit à utiliser la gestion des sessions, la sérialisation (grâce aux fonctions php serialize et unserialize) et la façon de gérer les

CHAPITRE 4. LA DÉMARCHE ADOPTÉE

objets, classes et interfaces. Pour une gestion des classes abstraites et des interfaces, PHP5 est nécessaire. Nous avons aussi besoin de méthode d'upload de fichiers vers la base. Bref nous avons fait de nombreuses recherches sur la toile pour nous renseigner sur telle ou telle partie dont nous avons besoin.

4.2.2 PostgreSQL

Pour cette partie, l'ensemble des connaissances nécessaires dont nous avons besoin ont été vues pendant notre cursus. Ainsi nous nous sommes servis de nos acquis.

4.2.3 CSS / javascript

Quelques bases du langage CSS ont été vues au cours du premier semestre de master. Mais pour rendre quelque chose de correct, nous avons donc du nous perfectionner en la matière, ce qui nous a permis de rendre une feuille de style raisonnablement bien faite. Pour la feuille de style, nous nous sommes restreint à un rendu sous Mozilla Firefox, puisque celui-ci est gratuit et très bien fait. Quant au JavaScript, ce langage nous été inconnu. Nous avons du apprendre les bases, qui ont été suffisantes pour notre application. Néanmoins cet apprentissage ne fu pas trop difficile puisque ce langage est proche du Java qui est largement vu dans notre cursus.

4.3 Les choix effectués

Dans cet outil, nous devons gérer plusieurs type d'utilisateurs, nous avons donc besoin d'un système d'identification, nous allons donc vous expliquer comment fonctionne le système d'identification. Comme décrit précédemment nous avons utilisé le PHP pour notre application ainsi nous vous expliqueront les choix qui ont motivés l'utilisation des sessions, classes, interfaces et de la sérialisation de PHP. Nous veront aussi pourquoi nous avons créés certaines fonctions JavaScript et enfin nous allons expliquer l'utilidsation d'une feuille de style CSS.

4.3.1 Le système d'identification

Ce système gère les différents utilisateurs en fonction de leur type (étudiant, professeur, professeur responsable ou administrateur) et de l'identifiant de chacun dans la base de données. Pour faciliter les choses, une table *inscrit* a été ajouter à la base avec des champs contenant l'identifiant et le mot de

passer de l'utilisateur ainsi que le type de l'utilisateur et son identifiant dans sa table. Par exemple, nous pouvons avoir deux utilisateurs ayant le même *id* mais un sera un étudiant et l'autre sera un professeur. Le type de l'utilisateur est donc mémorisé dans la table *inscrit*. Ce champ est donc très important pour une bonne connexion à l'application.

4.3.2 Les sessions, classes, interfaces et sérialisation PHP

L'utilisation des sessions a découlé naturellement puisque l'application est munie, comme précisé ci-dessus, d'un système d'identification de chaque utilisateur.

Elles servent donc à :

- Gérer les différents utilisateurs en leur permettant de se connecter en même temps sans soucis.
- Rendre possible l'utilisation des variables de sessions, très utiles pour faire voyager les informations d'une page à l'autre

L'utilisation des classes et interfaces, c'est à dire de la partie objet de PHP, a été motivé par la volonté de créer un code simple, compréhensible et évolutif. Grâce aux objets, nous pouvons alléger un peu le code et surtout éviter de le dupliquer quand ceci s'avère nécessaire.

Une remarque est qu'en PHP tous les champs des classes sont considérés comme public, contrairement à d'autres langages comme Java par exemple, il n'y a donc pas besoin d'accesses.

Quant à l'utilisation de la sérialisation, elle découle des deux points précédents. En effet pour pouvoir passer des objets de page en page, il est nécessaire de les sérialiser. C'est à dire de les transformer en chaînes de caractères pour les transporter, il suffit ensuite de les désérialiser dans la page où nous voulons les utiliser.

4.3.3 JavaScript

L'utilisation de JavaScript n'était pas obligatoire puisqu'il nous sert, dans notre application, à vérifier la bonne saisie de certains formulaires du site. Aussi le PHP aurait pu faire l'affaire mais nous étions curieux de savoir comment fonctionnait un peu ce langage. C'est pour cela qu'il a été employé.

4.3.4 CSS

Les mises en pages à l'aide de feuille de style CSS sont incontournables pour qui veut faire un site internet digne de ce nom. Notre site ne déroge pas

CHAPITRE 4. LA DÉMARCHE ADOPTÉE

à la règle. Les CSS permettent une mise en page rapide de toutes les pages et entièrement paramétrable.

Chapitre 5

Les détails de l'application

Dans ce chapitre, les objets php utilisés vous sont décrit, ainsi que certaines pages essentielles pour l'application.

5.1 Les classes et l'interface

Nous avons besoin de transporter les informations de chacun dans différentes pages du site. Ainsi, nous avons créé les classes *personne*, *etudiant* et *professeur* pour réaliser cela. Ces classes ont volontairement le même nom que les types d'utilisateurs.

Nous voulions aussi avoir une application la plus générale possible, nous avons donc créé une interface *database_interface* et une classe *postgresql* qui implémente cette interface. Ceci permet donc de changer de gestionnaire de base de données pour peu que l'on crée une classe implémentant cette interface. Par exemple, une classe *mysql* peut être créée si la base de données est une base de données MySQL.

5.1.1 Les classes *personne*, *etudiant* et *professeur*

La classe *personne* est une classe abstraite, un `new Personne` n'est pas possible.

Elle contient les champs communs aux étudiants et aux professeurs, c'est à dire :

- Le nom
- Le prénom
- L'adresse
- L'adresse Email

Enfin une fonction permet l'affichage des champs.

CHAPITRE 5. LES DÉTAILS DE L'APPLICATION

Les classes *etudiant* et *professeur* héritent de la classe *personne* avec respectivement le numéro étudiant et un numéro de bureau en plus.

Remarque :

Une classe *responsable* a été créée mais elle ne sert qu'à la généralisation en permettant d'utiliser le type de l'utilisateur pour créer l'objet.

5.1.2 L'interface *database_interface* et la classe *postgresql*

L'interface *database_interface* permet de se servir de différents systèmes de gestion de base de données (SGBD). Ceci permet une future généralisation de l'application aux divers SGBD.

L'interface *database_interface* ne contient simplement que les fonctions de connexion, déconnexion, requête, retour d'une ligne ou sous forme de tableau associatif du résultat d'une requête.

Dans la classe *postgresql* qui implémente cette interface sont ajoutés :

- Le nom du serveur
- Le nom de la base
- L'identifiant de connexion au serveur
- Le mot de passe de connexion au serveur

5.2 Les pages importantes

Dans cette partie, les deux pages les plus importantes vous sont décrites.

5.2.1 La page *entete*

La page *entete* est en charge d'afficher un menu en fonction du type d'utilisateur, ceci est donc très important car chaque type d'utilisateur a son propre menu. Pour ajouter un lien dans le menu, il suffit donc de le mettre au bon endroit dans cette page. C'est elle qui permet aussi l'affichage du nom et du prénom de l'utilisateur.

5.2.2 La page *type_accueil*

Cette page se lance quand l'utilisateur s'est identifié. Elle est invisible par celui-ci car elle n'affiche rien et redirige l'utilisateur vers ses pages en fonction de son type.

Cette page est à notre sens la plus importante. Plusieurs raisons à cela :

5.2. LES PAGES IMPORTANTES

Elle crée l'objet qui sert à la connexion avec la base de données, cet objet n'est créé qu'une seule fois, il est ensuite sérialisé et mis dans la variable de session appelée `$_SESSION['type_base']`.

Elle vérifie la bonne identification de l'utilisateur, s'il est connu dans l'application.

C'est cette page qui crée les objets php en fonction de l'utilisateur, ces objets sont ensuite sérialisés et mis dans une variable de session en fonction de leur type (appelée `$_SESSION['etudiant']` si c'est un étudiant).

Elle crée le chemin relatif d'accès aux différentes pages en fonction de l'utilisateur.

CHAPITRE 5. LES DÉTAILS DE L'APPLICATION

Chapitre 6

Les problèmes rencontrés

6.1 Base de données

Comme expliqué précédemment, la partie base de donnée a été faite à deux. On entend souvent qu'à deux c'est mieux !. Mais, bien que ceci est bénéfique, il peut y avoir des points de vue différents que chacun expose. C'est ce qui nous ait arrivé pour certaines cardinalités du MCD. Cependant nous sommes toujours tombés d'accord mais cela nous a fait perdre du temps pour le développement.

6.2 Interface web

Comme l'interface web communique avec la base de donnée, nous avons du attendre la fin de cette partie pour commencer l'interface. Pour cette interface web, nous avons utilisé le PHP, qui est un langage non compilé, avec sa gestion des objets, des classes et des interfaces. Ce qui, quand un problème survenait, posais certains inconvénients puisque lors d'un appel à une fonction d'une classe c'est dans cette classe que PHP nous donnait ses lignes d'erreurs et non à l'endroit exact de l'erreur. Ainsi tout le défi était de retrouver les erreurs, le débogage était donc assez compliqué. Nous avons donc du programmer un peu pas à pas, c'est à dire executer très souvent notre code, pour palier à ce problème. Cet ennui s'est propagé notamment à la feuille de style CSS et aux fonctions JavaScript.

6.3 Pas si simple de répartir le travail

Ce n'est pas la première fois que l'on faisait un projet en binôme car au cours de notre cursus, nous en avons fait un certain nombre. Mais c'est la première fois que l'on faisait un si gros projet et la répartition du travail fut un peu dure au début, ce qui nous a fait perdre du temps dans le développement de l'application. C'est après plusieurs discussions que nous avons abouti à une bonne répartition pour les trois grosses parties de l'interface web.

6.4 La barrière de la langue

Pour ce gros projet, la barrière de la langue était un réel défi. Nous sommes un français et un chinois, chacun ayant du faire des efforts pour comprendre les explications de l'autre ou pour faire comprendre son point de vue sur chacune des parties de l'application. Heureusement le chinois connaît un minimum la langue française donc comprendre certaines explications n'étaient pas simple mais faisable. Pour se faire comprendre par les français, c'était plus dur mais pas insurmontable. Cette barrière de la langue nous a fait perdre pas mal de temps au début et nous avons dû accélérer par la suite pour rattraper ce temps perdu, mais nous pensons que nous y sommes arrivés.

6.5 Réussir le projet

Nous pensons que notre projet est assez abouti. Cette application ressemble vraiment à quelque chose de correcte. Nous sommes fier de notre travail et nous espérons que les personnes l'utilisant le seront aussi.

Ce projet comportait une partie de la gestion des anciens étudiants, malheureusement nous n'avons eu le temps que d'y penser un peu. Nous avons pas eu le temps de nous mettre à fond dans cette partie. C'est un regret de notre part car cette partie, nous pensons, devrait être bien intéressante. Si l'opportunité se présente, nous pensons poursuivre cette application après notre stage. Il faut bien entendu que les professeurs/tuteurs soient d'accord.

Conclusion

Le stage, présenté à travers ce rapport, nous a beaucoup apporté, tant sur le plan de la communication que sur l'approfondissement de nos connaissances.

Ce stage au CRIL a été bénéfique pour notre futur. Nous avons pu découvrir la liberté qu'offre le travail dans un laboratoire de recherche, notamment la gestion personnelle de notre travail,.

Ce stage nous a également permis d'appliquer nos connaissances à un projet de plus grande envergure que les projets auxquels nous avons pu participer jusqu'alors. Nous avons été aussi amené à développer nos compétences dans des domaines que nous rencontrerons probablement dans la suite de nos études et dans notre vie professionnelle dont l'échéance ne cesse de se rapprocher.

Nous retiendrons donc de ce stage une expérience très enrichissante, que nous ne regrettons nullement d'avoir vécus.