

Remerciement

Il m'est agréable d'exprimer ma reconnaissance auprès de toutes les personnes, dont l'intervention au cours de mon stage, a favorisé son aboutissement.

Je tiens à exprimer ma gratitude à mon encadrant **Mr. DE LA MORLAIS Bertrand** chef du **Département des Systèmes d'Information (DSI)** à l'**Université de Technologie de Troyes** pour ses directives précieuses et ses conseils pertinents qui m'ont appuyé considérablement dans ma démarche.

Mes remerciements les plus sincères au **Mme. MERGHEM-BOULAHIA Leila**, mon encadrant à l'**UTT**.

Je tiens aussi à remercier Mme. **THOORIS Valérie**, représentante du service REE de l'UTT pour sa patience, ses directives et ses qualités humaines, ainsi que toutes les personnes du service **REE** qui m'ont aidé à élaborer ce travail grâce à leurs conseils constructifs.

Je profite de l'occasion pour remercier ma famille, mes amis et surtout mes parents pour leurs conseils et soutiens permanents.

Que tous ceux qui m'ont aidé, de près ou de loin, trouvent ici l'expression de mes sentiments les meilleurs.

Sommaire

<i>Remerciement</i>	1
<i>Introduction</i>	3
I. Chapitre1 : Contexte général du projet.....	4
1. <i>Présentation de l'organisme d'accueil</i>	5
2. <i>Présentation détaillée du projet</i>	6
i. <i>Présentation du projet</i>	6
ii. <i>La documentation</i>	8
iii. <i>Présentation du travail des stagiaires précédents et la mission du stage actuel</i>	9
II. Chapitre 2 : Analyse fonctionnelle	11
1-Technologies Utilisées	12
a. <i>Zend Framework</i>	12
b. <i>Architecture MVC</i>	12
c. <i>JavaScript et AJAX</i>	16
III. Chapitre3 : Développement	17
<i>Conclusion</i>	24
<i>Liste des figures</i>	25
<i>BIBLIOGRAPHIE</i>	26

Introduction

L'université de technologie de Troyes comporte 2500 étudiant annuellement dont 800 en moyenne partent en stage, le service REE prend en charge la procédure de recherche et proposition des stages aux étudiants via la base de données GERRY.

A cet effet une solution informatique répondant à la procédure de recherche et proposition des stages doit être élaborée et mise en œuvre, en permettant de stocker les coordonnées des entreprises sur la base de données d'une part, de proposer les sujets des stages aux enseignants pour qu'ils soient validés et finalement lancer le processus de recherche des stages aux étudiants ayant les droits d'accès à l'application pour se positionner sur les sujets qui les intéressent.

Ainsi le présent rapport résume le travail accompli dans le cadre du stage TNo9 au sein du Centre d'Etude, Développement, Réalisation et Exploitation (CEDRE) à la DSI de l'UTT. Ce service s'occupe du développement de toutes les applications utilisées au sein de l'UTT.

Pour le développement de cette application alias REE gestion des stages (REEGESSTG), j'ai utilisé le ZEND FRAMEWORK comme support pour le langage PHP et ORACLE comme SGBD.

Ce rapport est divisé en trois chapitres. Le premier consiste à présenter l'organisme d'accueil et expliquer le contexte général du projet, le deuxième chapitre consiste à citer les technologies utilisées durant la réalisation du projet, et le dernier explique les démarches suivies durant le développement du projet.



I. Chapitre1 : Contexte général du projet

1- *Présentation de l'organisme d'accueil*

2- *Présentation détaillée du projet*

a. *Présentation du projet*

b. *La documentation*

c. *Présentation du travail des stagiaires précédents et la mission du stage actuel*

1. Présentation de l'organisme d'accueil



L'université de technologie de Troyes est considérée parmi les dix meilleures écoles d'ingénieurs en France, cet établissement public créé en 1994 comporte 2500 étudiants annuellement ainsi d'un personnel de 370, partagé entre les fonctionnaires et les enseignants chercheurs, avec tout cet effectif la présence d'un système d'information fiable est un élément majeur pour la réussite de l'école.



Le département du système d'information de l'UTT se coordonne régulièrement avec tous les services de l'école pour optimiser leur travail en créant des applications fiables afin de réduire la complexité de leurs tâches quotidiennes.

C'est le cas pour le service relation entreprise étudiant (REE), où l'application actuelle n'optimise pas le processus de recherche et d'affectation des stages aux étudiants, vu que le responsable du service est toujours amené à autoriser ou interdire à l'étudiant d'accéder à l'application des stages actuelle, or ceci doit être fait automatiquement par l'application en se basant sur les remarques attribuer par le jury à chaque étudiant dont chaque remarque est représentée par un code sur la base de données, ce qui implique un énorme gaspillage de temps et d'énergie pour le service.

A cet effet, la réalisation de l'application REEGESSTG doit permettre au service REE d'optimiser les processus de recherche, proposition et affectation des stages aux étudiants.

2. Présentation détaillée du projet

i. *Présentation du projet*

Face à la présence des multiples problèmes du module de gestion des stages, le service REE souhaite se doter d'un outil à la fois simple et performant afin de répondre aux attentes des missions à traiter. Cet outil aura comme principale fonction de permettre des gains de temps assez considérables dans la saisie et l'enregistrement des données, mais aussi de pouvoir diminuer la saisie des doublons et de ce fait assurer la fiabilité des flux de données entrantes.

L'application est divisée en plusieurs modules dont la liste est présente en permanence à gauche de l'écran (figure1). Ils ont été rangés par ordre de priorité en collaboration avec la maîtrise d'ouvrage.



Figure 1 : Menu principal pour Administrateur

Le module principal sur l'application est celui du « Sujet » dont il a divisé en 3 sous menus : « Création sujet », « Sujet » et « Validation sujet ».

L'architecture fonctionnelle que ce module de gestion supporte est clarifiée sur la figure 2.

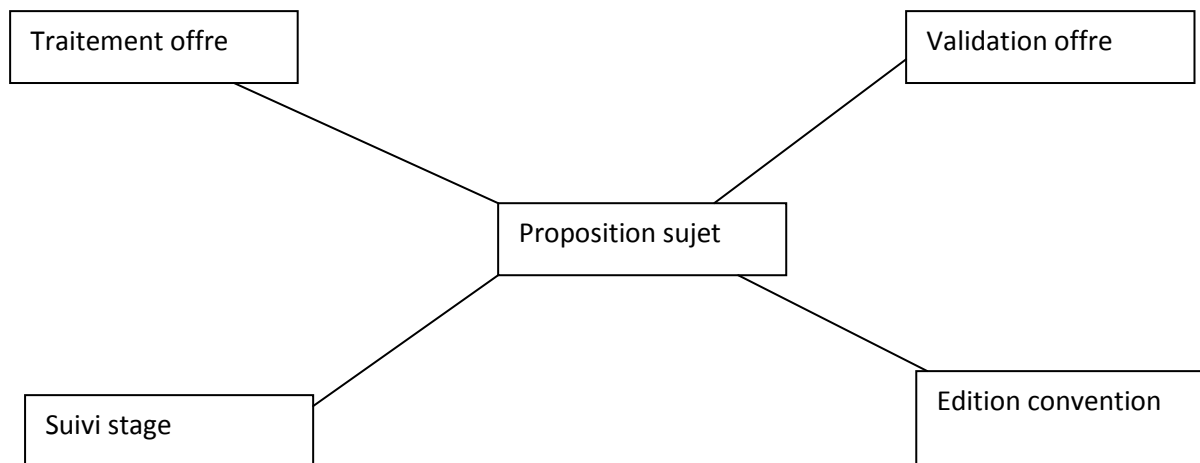


Figure 2: processus de proposition du stage

Le sous menu « Création sujet » concerne la création du sujet, ce dernier est soit provisoire ou définitif.

Pour la partie « Sujet », on peut consulter ou même modifier les sujet créé sur la base de données.

Tous les sujets proposés sur la base de données sont en état 'PR', c'est-à-dire 'proposé', et ils doivent être validés par les responsables des filières afin qu'ils deviennent lisibles aux étudiants pour postuler.

Le module « Sujet » est lié au module « Entreprise » sachant qu'on ne peut pas créer un sujet sans le relier avec une entreprise, alors le choix d'une entreprise est obligatoire avant la création du sujet.

A cet effet, tout un processus accompagne la création d'une entreprise, commençant par la possibilité de modifier ou bien consulter une entreprise, la création d'une entreprise avec des filières ou sous traitants, ajouter une entreprise à la 'Black List', la possibilité du déménagement en gardant la trace de tous les déménagements et leurs dates.

A ces deux modules principaux, il y a d'autres modules qui s'ajoutent, par exemple : « Suivi Recherche », « Attribution Suiveur », « Suivi de Stage » et « Soutenance ».

Tous ces modules permettent le suivi du stagiaire d'une façon dynamique afin d'optimiser les tâches de tous les interlocuteurs en parallèle lors d'un stage.

Le module « Gestion des droits d'Etudiants » permet de gérer la liste des étudiants ayant le droit de partir en stage le semestre prochain.

« Taches à faire » est un tableau où on trouve toutes les informations sur l'application inachevée, par exemple une entreprise sans numéro de SIRET/SIREN ou un sujet une date de début et de fin...

ii. La documentation

Le projet REEGESSTG est développé en sa globalité par des stagiaires de l'UTT, ce changement de développeur régulier impose l'existence d'une documentation qui permet un passage intégral des informations lors du changement d'intégrateur.

Le premier document, la *Spécification des Besoins des Utilisateurs (SBU)* (figure 3) est un document servant à analyser les besoins de la maîtrise d'ouvrage et à les organiser de façon logique. On y trouve par exemple des diagrammes d'activité s'occupant de détailler les processus métier afin de mieux les comprendre et pouvoir trouver un moyen de les traiter.

Tous les besoins de la maîtrise d'ouvrage sont indiqués dedans.

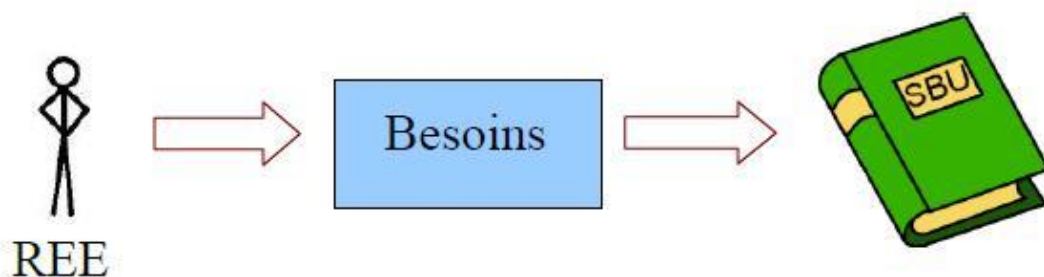


Figure 3 : Spécification des Besoins des Utilisateurs(SBU)

Pour les Spécifications Fonctionnelles Détaillées (SFD), ce document permet principalement de proposer des interfaces graphiques pour le projet et de décrire l'utilité de chacun des éléments les constituant. Chaque chapitre de la SFD se compose d'une maquette d'interface graphique puis d'un tableau décrivant un par un les champs du formulaire de cette dernière (figure 4).



Figure 4: Validation de la SFD

La SFD met par écrit les demandes fonctionnelles que l'application devra respecter, elle est donc validée par la maîtrise d'ouvrage. La maîtrise d'œuvre quant à elle doit assurer la faisabilité de ces fonctions d'un point de vue purement technique, et c'est ensuite à l'intégrateur de parvenir à les programmer (figure 5).

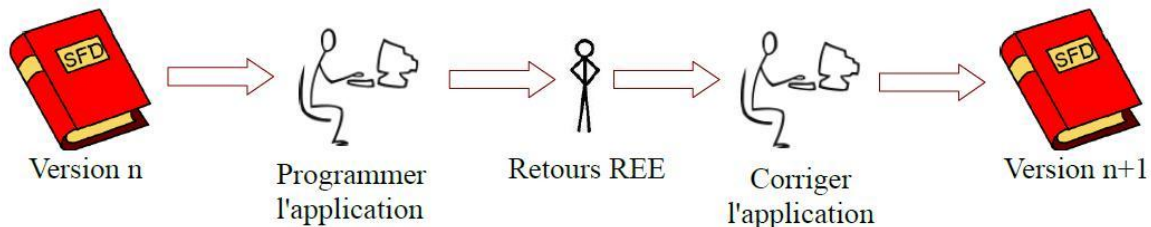


Figure 5: processus de la validation de la SFD

La troisième et dernière partie de la documentation : la Spécifications Techniques Détaillées (STD) est directement liée à la SFD, son rôle principal est de décrire le fonctionnement purement technique des aspects évoqués dans la SFD afin que l'intégrateur suivant puisse en comprendre la logique en un minimum de temps pour ensuite les améliorer. La STD est donc un document purement destiné à la maintenance du code et aux améliorations et il n'est accessible qu'aux développeurs du projet.

iii. Présentation du travail des stagiaires précédents et la mission du stage actuel

L'application REEGESSTG a été développée par trois stagiaires : **LIZO Duongbeu arsene**, **SAMB Modou mamoune** et **COUASNON Maxime**.

Les deux premiers stagiaires ont passé leurs stages en parallèle, dont le premier a travaillé sur la Spécification des Besoins Utilisateurs (SBU), son rôle principal était de faire un point sur les processus métiers qui interviennent lors de la gestion des stages.

Par contre le deuxième stagiaire a travaillé sur les Spécifications Fonctionnelles détaillées (SFD), dont son rôle se réduisait à présenter de manière détaillée les différentes modifications qui seront apportées à l'application « Gestions des Stages », ces modifications ont été décrites initialement dans la SBU, donc il a passé la majorité de son temps à développer l'interface graphique de la nouvelle application.

Pour Maxime, mon prédécesseur, il a commencé à programmer les modules principaux de l'application. Il a démarré son travail par le module « Entreprise », ensuite il a enchainé par le module « Sujet ». Il a fait de grands pas sur les deux modules, mais ils n'ont pas été clôturés définitivement.

Ma mission, était au début de développer les modules de « Validation Sujet », « gestion des droits d'étudiant » et la partie du déménagement des entreprises, mais après un certain nombre de tests effectués par la maîtrise d'ouvrage, chose faite car la maîtrise d'ouvrage a été changée lors du changement de stagiaire, on s'est rendu compte que la plupart des fonctionnalités développées par le stagiaire précédent ne sont pas complètes, Alors j'ai repris la tâche de compléter les tâches inachevées, avant de reprendre la bonne route .

La chose la plus difficile pour un stagiaire débutant est de reprendre le code d'un autre développeur, c'était le cas pour moi, mais j'ai dépassé les difficultés rencontrées, même si ceci m'a pris un énorme délai, chose qui m'a obligé d'annuler une partie prévue lors du début du stage, c'était la partie concernant « Gestion des Droits d'Etudiants ».

II. Chapitre 2 : Analyse fonctionnelle

Chapitre 2

1-Technologies Utilisées

- a- Zend Framework*
- b- Architecture MVC*
- c- JavaScript et AJAX*

2- Conception

1-Technologies Utilisées

a. Zend Framework

Le Zend Framework est un Framework pour PHP5 créé en mars 2006 par Zend Technologies. Le Zend Framework, aussi nommé ZF, a été développé dans le but de simplifier le développement Web tout en recommandant les bonnes pratiques et la conception orientée objets en offrant des outils aux développeurs (figure 6). ZF permet aussi d'utiliser nativement le principe de MVC (Modèle-Vue-Contrôleur) mais ne l'oblige pas.

L'utilisation du zend Framework dans l'application, a apporté une véritable valeur ajoutée en terme normalisation du code, d'une maîtrise des développements et d'un gain de temps.

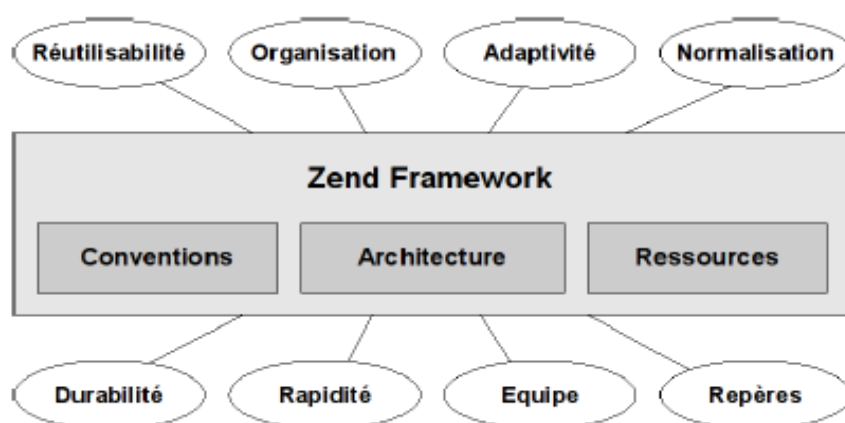


Figure 6: Zend Framework

b. Architecture MVC

Sur ce projet, Zend est utilisé ici suivant l'architecture Modèle-Vue-Contrôleur, alias MVC, ce qui explique l'arborescence du projet : un dossier 'model', un dossier 'views' et un dossier 'Controller' (figure 7).

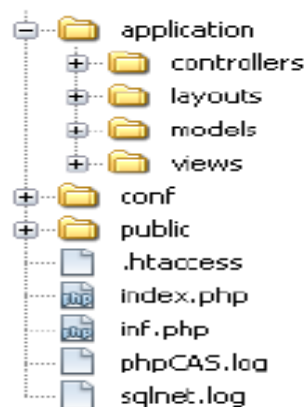


Figure 7: Architecture MVC

Sur cette figure on remarque que cette architecture permet de structurer l'application selon certaines règles et d'uniformiser les méthodes de développement entre les différents acteurs du projet.

Le zend Framework permet également de récupérer une URL unique des pages, cette dernière est décomposée de plusieurs parties suivant la logique de Zend :

```

http://gestion-dev.utt.fr/REEGESSTG/modsujet/zommd/id/70145
      Contrôleur- |
                  |
                  Action-
                  Paramètres-
  
```

Le « contrôleur » est en fait l'élément du menu de gauche dans lequel on se trouve. Il possède lui-même un certain nombre de pages qui sont appelées des « actions » par exemple la consultation des sujets créés. Enfin, l'URL permet aussi de passer des paramètres. Dans ce cas précis, on interpréterait cette adresse par « le numéro id du sujet que l'on veut consulter est 70145 ».

D'une manière plus générale, on peut dire que chaque contrôleur présente dans l'arborescence un 'Module'. (Figure 8)

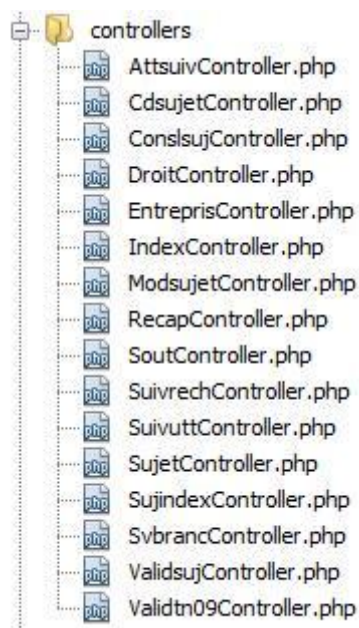


Figure 8: Le dossier 'Controller'

On mentionne aussi que chaque fichier 'Controller' représente une classe unique, dont toutes les classes 'Controller' héritent de la classe 'Zend_Controller_Action'.

On y trouve plusieurs fichiers PHP : un fichier par contrôleur du menu de gauche de l'interface graphique.

Dans un projet Zend classique, les classes `XxxController` héritent de la classe `Zend_Controller_Action`. Mais pour que les droits d'accès aux différentes pages suivant les utilisateurs soient respectés, la DSI a programmé la classe `BlMController`, qui hérite elle-même de `Zend_Controller_Action` ce qui est bien clarifié sur la figure 9.

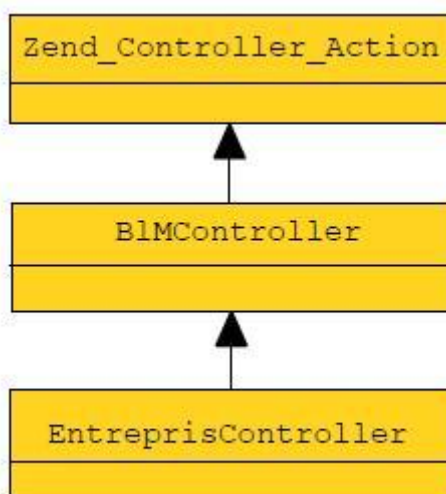


Figure 9: gestion des droits d'accès

Pour le dossier `views/script` Chaque sous-dossier contient plusieurs fichiers PHTML, où chacun portant le nom d'une action appartenant au contrôleur en question. (Figure 10).

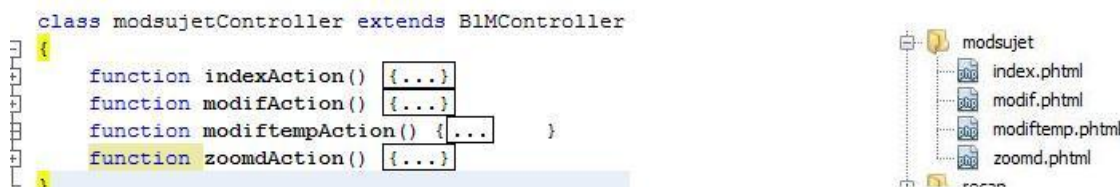


Figure 10: Relation entre Controller/Views

Le dossier 'layouts' de l'arborescence fait aussi partie de la couche Vue de MVC. Il s'agit en fait de la structure première des pages Web de l'application. On y trouve plusieurs fichiers PHTML, chacun représentant une partie de la page. (Figure 11)

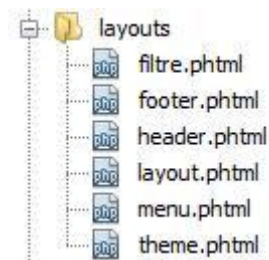


Figure 11: le dossier 'Layout'

On commence par écrire `layout.phtml`, il est capable d'appeler le contenu des autres fichiers aux endroits où on en a besoin avec la fonction `render()`. Ainsi on possède un squelette de page modulaire où chaque partie est située dans un fichier différent puis toutes rassemblées en une seule page.

Pour le modèle, la partie du MVC qui s'occupe de l'objectif central de l'application, a deux rôles principaux, le premier est la création de tous les champs des formulaires de l'application en héritent du `Zend_Form`.

```
class formSujet extends Zend_Form
```

Alors chaque élément du formulaire est déclaré dans un 'Model' pour être utilisé à la 'Views'. (Figure 12)

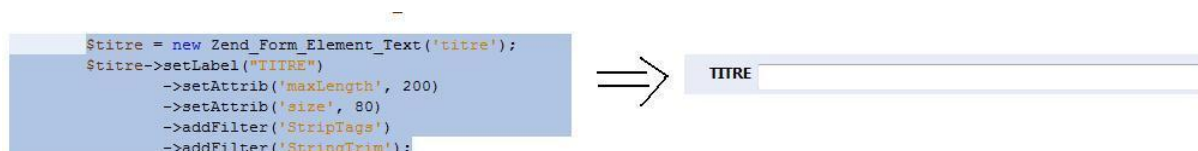


Figure 12: Model/Views

L'autre rôle du 'Model' est de s'occuper du dialogue avec la base de données, où la classe `Zend_Db_Table` est utilisée pour la recherche, l'ajout, la modification ou bien la suppression des enregistrements de la base de données. (Figure13)

```
$s->_db->insert("STAGE_SUJET",array("SUJ_ID"=>new Zend_Db_Expr("STG_SUJ_ID_SEQ.NEXTVAL"),
    "SUJ_TITRE"=>$data['titre'], "STG_OBLIGATOIRE_\\$F"=>$oblig, "STG_OPTIONNEL_\\$F"=>$optionnel,
    "SUJ_DESCR"=>$data['Sujet'], "SUJ_DATE"=>new Zend_Db_Expr("TO_DATE(CURRENT_DATE)"), "NB_STAG"=>1,
    "STG_COND_MAT"=>$data['avantages'], "AN_UNIV"=>$annee, "SIGNATAIRE_ID"=>$sign_id,
    "ADR_ID"=>$adr_id, "RESP_ADM_ID"=>$adm_id,
    "RESP_DRT_ID"=>$tut_id, "COMP_ID"=>$comp_id, "SUJ_ETAT_\\$C"=>"PR",
    "CRE_DATE"=>new Zend_Db_Expr("TO_DATE(CURRENT_DATE)"), "STG_DEBUT_DATE"=>$data['date_debut'],
    "STG_FIN_DATE"=>$data['date_fin'], "GRATIFICATION"=>$gratif, "RESP_DRT_SIGN_\\$F"=>$tut_sign,
    "RESP_ADM_SIGN_\\$F"=>$adm_sign, "COMP_CONV_ID"=>$comp_conv_id,
    "ADR_CONV_ID"=>$adr_conv_id, "TUT_SUPP_ID"=>$tut_s_id,
    "STG_SUS_DEBUT1"=>$data['date_debut1'], "STG_SUS_FIN1"=>$data['date_fin1'],
    "STG_SUS_DEBUT2"=>$data['date_debut2'], "STG_SUS_FIN2"=>$data['date_fin2'],
    "STG_SUS_DEBUT3"=>$data['date_debut3'], "STG_SUS_FIN3"=>$data['date_fin3']));
```

Figure 13:Exemple d'insertion d'un sujet définitif a la base de données

c. JavaScript et AJAX

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacune équipées de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

Le JavaScript a été utilisé pour vérifier si les champs de quelques formulaires sont bien remplis.



Pour l'AJAX (acronyme d'Asynchronous Javascript and XML), c'est une manière de construire des applications Web et des sites web dynamiques basés sur diverses technologies Web ajoutées aux navigateurs.

De même, le langage AJAX était utilisé pour générer la liste des départements et villes directement en utilisant le code postal.

III. Chapitre3 : Développement

Chapitre 3

Quand le stage a débuté, il était prévu que je développe trois modules, la partie du déménagement des entreprises, « validation des sujets » et « gestion des droits d'étudiants ».

Mais avant il était indispensable que je clôture les modules développés par mon prédécesseur, après un certain nombre de tests effectués par moi-même et la maîtrise d'ouvrage, on s'est rendu compte que la plupart des fonctionnalités développées par le stagiaire précédent ne sont pas complètes, Alors j'ai repris la tâche de compléter les tâches inachevées, avant de reprendre la bonne route.

Alors j'ai commencé par le module de « création de sujets » dont la plupart des saisies se faisait sans tests sur les champs, alors il y avait une possibilité de saisir un sujet vide sans titre ni description ni formations et même pas de tuteur, et le pire pas d'entreprise, à cet effet l'utilisation de javascript pour les tests des champs vides ou l'obligation de choisir une entreprise avant la saisie du sujet étaient indispensables. (Figure 14 et 15)

CP: Département: Ville:

Nom: Prenom: Fonction: Tel: Email: Rôle:

Etat: ☒ Actif ☐ Inactif ☒ Provisoire

Le choix d'une entreprise avant la saisi du sujet est obligatoire

OK

Résultats de la recherche

	S	Nom entreprise	Adresse 1	Adresse 2	Pays	Date
<input checked="" type="radio"/>	gfdsg				FR	
<input checked="" type="radio"/>	entr test				FR	
<input checked="" type="radio"/>	pop				FR	
<input checked="" type="radio"/>	entreprise10				FR	
<input checked="" type="radio"/>	entr4	adr1	adr2		FR	
<input checked="" type="radio"/>	entr3				FR	
<input checked="" type="radio"/>	entr2				FR	
<input checked="" type="radio"/>	entr1				FR	
<input checked="" type="radio"/>	test	adre1	adre2		FR	
<input checked="" type="radio"/>	a				FR	
<input checked="" type="radio"/>	test	ad1	ad2		FR	TROYES
<input checked="" type="radio"/>	15524454	adr	adr		FR	
<input checked="" type="radio"/>	code APE	adr	adr2		FR	
<input checked="" type="radio"/>	soc?7t71				FR	
<input checked="" type="radio"/>	ghj	a	d		FR	
<input checked="" type="radio"/>	a	a			FR	ville
<input checked="" type="radio"/>	a	a			FR	
<input checked="" type="radio"/>	12345				FR	
<input checked="" type="radio"/>	12345				FR	
<input checked="" type="radio"/>	test	adr1			FR	

Prop TEMP Prop DEF

< précédent | 1 | 2 | 3 | 4 | 5 | suivant >

connecté en tant que :Mohammed OUAADGHIRI - application optimisée pour Firefox

Figure 14 : Obligation du choix d'une entreprise avant la saisie du sujet

The form contains the following fields and sections:

- Administrative Data:** Nom, Prénom, Fonction, Service, Complément d'adresse, Email, Tel, Portable, Fax.
- Signatory Information:** Signataire (si différent responsable administratif et/ou tuteur) checkbox, Civilité dropdown, Nom, Prénom, Fonction, Service, Complément d'adresse, Email, Tel, Portable, Fax.
- Stage Details:** TITRE, SUJET, Date début stage, Date fin stage, NB semaines (with a 'calculer' button), Gratification (with a '€' symbol), Avantages, Suspension 1 (Début, Fin), Stage Obligatoire ? (checkbox), Suspension 2 (Début, Fin), Stage Optionnel ? (checkbox), Suspension 3 (Début, Fin).

A modal dialog box is shown with the message: "Il faut compléter toutes les informations du sujet concernant : le Tuteur, l'Etudiant, le Responsable administratif, le Titre et la Description." and an "OK" button.

Figure 15: test sur les champs vides pour la création des sujets

Toujours sur le module du sujet, il n'y avait pas la possibilité de consulter les sujets, l'utilisateur était obligé de passer par la page de modification pour consulter les sujets, alors j'ai créé une nouvelle page seulement pour la consultation, mais à ce moment la question de qui a le droit de consulter et modifier parmi les utilisateurs, et qui n'est sensé qu'à consulter s'est posée.

Alors j'ai créé trois profils pour l'application : le premier est considéré comme l'administrateur de l'application, cet utilisateur est représenté par les fonctionnaires du service REE avec le menu suivant : (Figure 16)



Figure 16: menu du service REE

Le deuxième utilisateur est présenté par les enseignants, qui auront le droit d'accéder à l'application pour consulter les sujets ou bien les valider, mais ils ne pourront en aucun cas les modifier, le menu suivant concerne les enseignants : (Figure17)

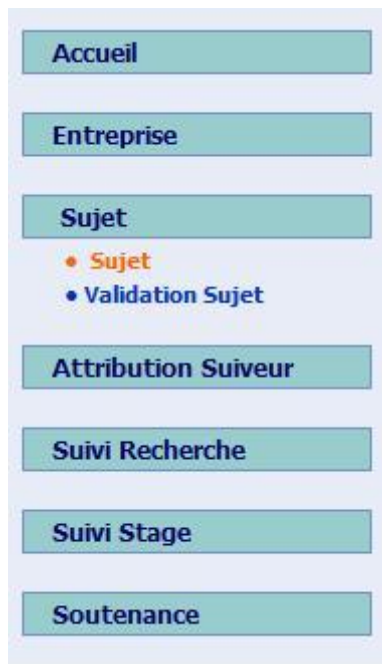


Figure 17: menu enseignants

Le troisième utilisateur est l'étudiant, ce dernier aura encore moins de possibilité de naviguer sur l'application dont il n'aura que la possibilité de consulter les sujets et faire suivre son stage. (Figure18)

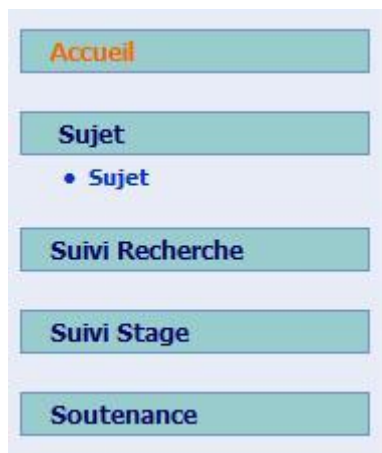


Figure 18: Profil étudiant

Après ces problèmes sur le modules, il fallait passer au module de « L'entreprise », où il y avait pas mal de 'bugs', parmi ces bugs on peut citer par exemple l'absence des tests lors de la création d'une entreprise, ou l'existence des lettres et caractères champs qui sont censés être numériques, comme les numéros de téléphone ou bien

le numéro de SIRET|SIREN, d'où l'obligation de rendre les champs numériques. (Figure 19) :

The screenshot shows a web form titled "Recherche par personne morale". It contains several input fields for company information: "Nom/Sigle/Groupe", "N°id", "N°SIRET/SIREN" (with a value 'a'), "TVA intra", "Dumbs numb", "Adresse 1", "Adresse 2", "CP", "Ville", "Département", "Tel entreprise", "Fax", "Type d'entreprise" (with a dropdown), "Nombre de salariés", "Code APE", "Domaine d'activité", and "Etat" (with checkboxes for "Actif", "Inactif", and "Provisoire"). A modal dialog box is displayed over the form, containing the text "Ceci est un champs numerique" and an "OK" button. Below the form, there is a section for "résultats de la recherche" which includes a table with columns for "ENTREPRISE" (containing checkboxes, S, Nom entreprise, Adresse 1, Adresse 2, P, Ville) and personal information (Nom, Prénom). A link "Ajouter Entreprise" is also visible.

Figure 19: champs SIRET numérique

Autre problème considéré ergonomique, pourquoi ne pas optimiser la saisie des adresses en 'Ajax' en utilisant un Code Postal seulement, l'utilisateur peut récupérer la ville et le département automatiquement, chose qui a été faite avec succès. (Figure 20)

The screenshot shows a close-up of the address section of the form. It includes fields for "Adresse 2", "CP" (with the value "90"), "Département" (with a dropdown menu showing "90 - Territoire-de-Belfort"), and "Pays" (with a dropdown menu showing "FRANCE").

Figure 20: l'affichage du nom du département automatiquement après la saisie du CP

Après la résolution de tous ces bugs et d'autres, j'ai commencé la deuxième étape du stage où j'étais amené à développer mes propres modules. J'ai commencé alors par développer la partie du « Déménagement des Entreprises », la difficulté de cette partie était qu'après chaque déménagement d'une entreprise, il faut garder l'ancienne adresse, avec la date du début et date de fin, et lors qu'il y aura un stagiaire qui demandera l'attestation de stage, il doit y être marqué l'ancienne adresse de l'entreprise avec toutes les coordonnées associées tel que le numéro de SIRET & SIREN, date de déménagement (Figure 21)

DEMENAGEMENT

Activer ☒

Date de Déménagement : 12/05/07 Date MAJ : 27/01/12 Auteur Mise à Jour : ouadghim

SIRET : 11455214458789 Adresse 1: adr1 Adresse 2: adr2 Boite Postale : 5478

Code Postal: Nouvelle Ville : Département : Pays : FRANCE

[Lancer la procédure](#)

Historique des Déménagements de l'Entreprise					
Auteur	Date de Modification	Date Déménagement	Date Fin	N° SIRET	Adresse
MAJ_AUTO			01/01/04		Aeroparc ,90150 . FONTAINE

Figure 21:déménagement d'une entreprise

J'ai commencé à développer aussi la partie de validation des sujets par les enseignants, mais malheureusement je n'avais pas assez de temps pour clôturer ce module.

Conclusion

Ce stage est pour moi une vraie expérience professionnelle dans le domaine informatique.

J'ai pu découvrir le fonctionnement d'un système informatique aussi complexe au sein du Département des Systèmes d'Information de l'UTT, ce qui m'a permis également de mettre en œuvre mes compétences et mes connaissances que j'ai pu acquérir tout au long de ma formation universitaire.

D'autre part, je suis fier d'avoir participé d'une manière efficace et constructive à la réalisation d'un projet d'envergure au sein de l'UTT, et ceci en se basant sur les connaissances acquise principalement dans les UV IFo2 et LOo7.

Liste des figures

Figure 1 : Menu principal pour Administrateur.....	6
Figure 2: processus de proposition du stage.....	7
Figure 3 : Spécification des Besoins des Utilisateurs(SBU).....	8
Figure 4: Validation de la SFD.....	8
Figure 5: processus de la validation de la SFD.....	9
Figure 6: Zend Framework.....	12
Figure 7: Architecture MVC.....	13
Figure 8: Le dossier 'Controller'.....	14
Figure 9:gestion des droits d'accès	14
Figure 10:Relation entre Controller/Views	15
Figure 11: le dossier 'Layout'	15
Figure 12: Model/Views	15
Figure 13:Exemple d'insertion d'un sujet définitif a la base de données	16
Figure 14 : Obligation du choix d'une entreprise avant la saisie du sujet.....	18
Figure 15: test sur les champs vides pour la création des sujets	19
Figure 16: menu du service REE	20
Figure 17: menu enseignants	21
Figure 18: Profil étudiant.....	21
Figure 19: champs SIRET numérique	22
Figure 20: l'affichage du nom du département automatiquement après la saisie du CP.....	22
Figure 21:déménagement d'une entreprise	23

BIBLIOGRAPHIE

ZEND FRAMEWORK: API DOCUMENTATION

<http://framework.zend.com/apidoc/1.11/>

<http://www.toutjavascript.com>

<http://www.xul.fr/xml-ajax.html>

http://fr.wikipedia.org/wiki/Ajax_%28informatique%29

<http://fr.wikipedia.org/wiki/JavaScript>

COUASSON Maxime. *Spécification Fonctionnelles Détaillées(SFD) : Evolution ergonomique de l'outil de gestion des stages*. Version 1.1.0

De la Morlais Bertrand. *Spécification des Besoins Utilisateur(SBU) : Evolution ergonomique de l'outil de gestion des stages*. Version 1.1.0

SARR, Seydi. *Spécifications Techniques Détaillées(STD) : Remise à Niveau de l'interface de Saisie des données du module Gestion des Entreprises de Gerry*. Version 1.0.0