

PROJET DE GESTION DE STAGE

Par Cyril de Oliveira et Sébastien Verneyre – L3 Miage

Sommaire

Découpage des tâches.....	1
Mise en place du projet et création des diagrammes.....	1
BDD.....	5
Les classes.....	6
Interface graphique	7
Conclusion	12

Découpage des tâches

Dès le moment où notre groupe a été créé, nous avons décidé de répartir les tâches de façons à ce que chacun puisse faire un travail qui ne dépend pas d'un autre. Nous avons répartie les tâches de la façon suivante :

- Mise en place du projet, répartition des tâches (en commun)
- Production des divers Diagrammes (Sébastien)
- Création et mise en place du Git Hub (Sébastien)
- En parallèle, conception des classes à l'aide des premiers diagrammes (Cyril)
- Fin des diagrammes, mise en commun sur le Git Hub et tenu de l'avancement du projet (en commun et toutes les semaines)
- Élaboration de la Base de données (Cyril)
- Création de l'interface graphique (Cyril)
- Fin du code et test du projet (en commun)
- Écrire le compte rendu final (Sébastien)

Mise en place du projet et création des diagrammes

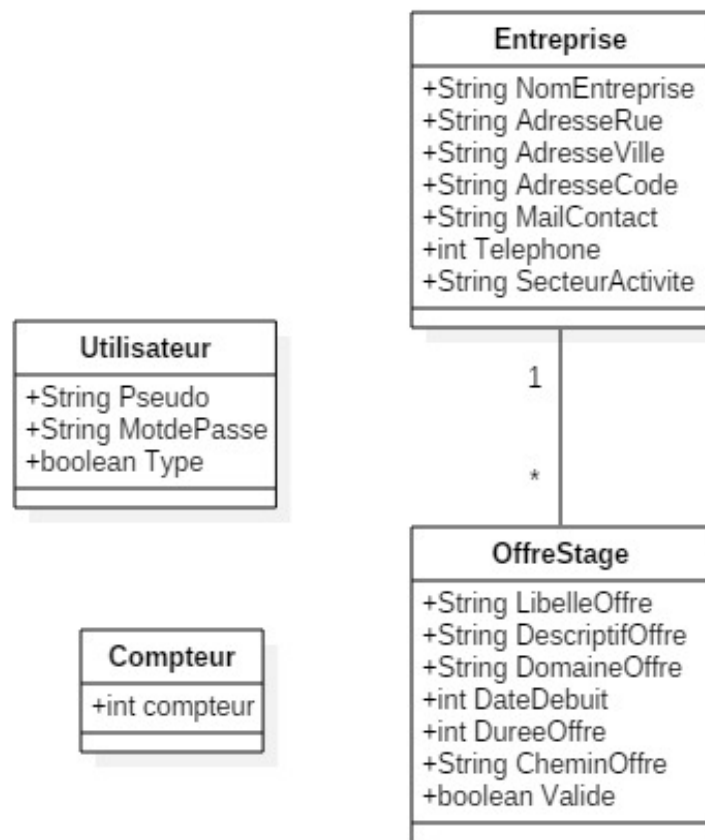
Pour commencer le projet, nous avons établi sur cacao un « brouillon » des 3 différents diagrammes que nous avons besoin. Puis nous avons recopié et amélioré les diagrammes suivants à l'aide du logiciel StarUML.

Après avoir attribué les différentes nous nous somme imposé un certain délai pour chaque tâche afin que le projet soit finir en temps et en heure. C'est pourquoi nous avons tenu chaque semaine un tableur résumant ce qui a été fait et ce que nous devons faire.

Pour la mise en commun, nous avons appris à nous servir de Git Hub qui et

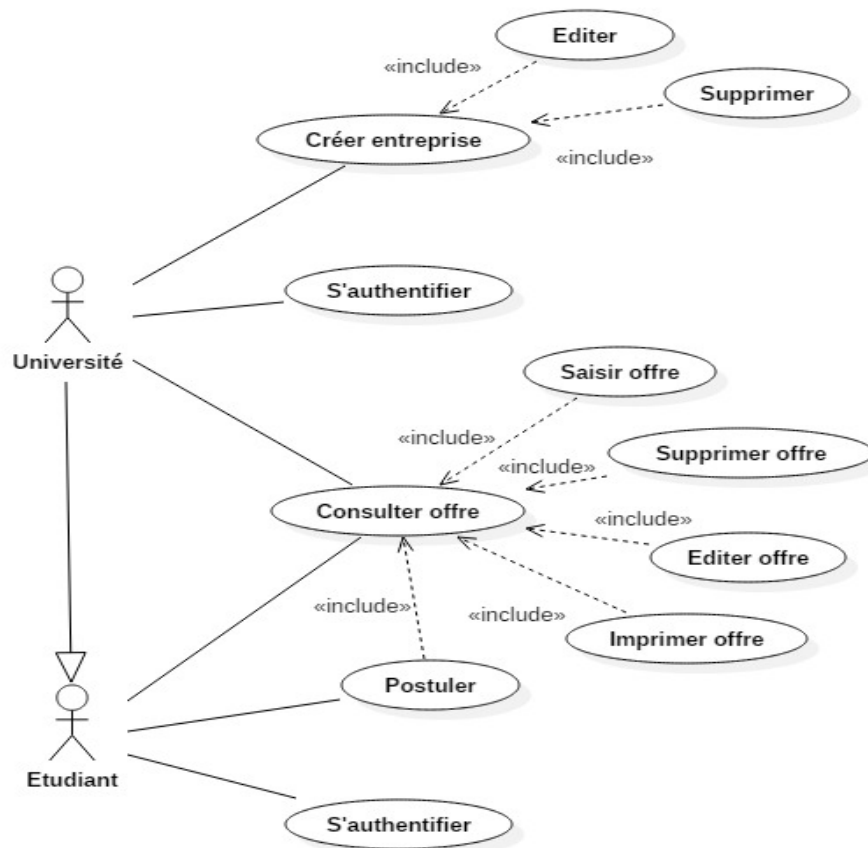
un système de gestion de la mise en commun d'un projet par plusieurs personne. Nous avons donc déployé notre Git Hub et le mettons à jour à chaque changement dans le code ou lors de l'ajout d'un élément.

Diagramme de classe :



Notre diagramme de classe est composé des classes **Entreprise** et **OffreStage** qui sont liées par une association (il peut y avoir 1 ou plusieurs offres de stage pour 1 entreprise) et des classes **Utilisateur** et **Compteur** qui sont des classes qui n'ont aucune interactions avec les autres classes.

Diagramme de Cas d'utilisation :



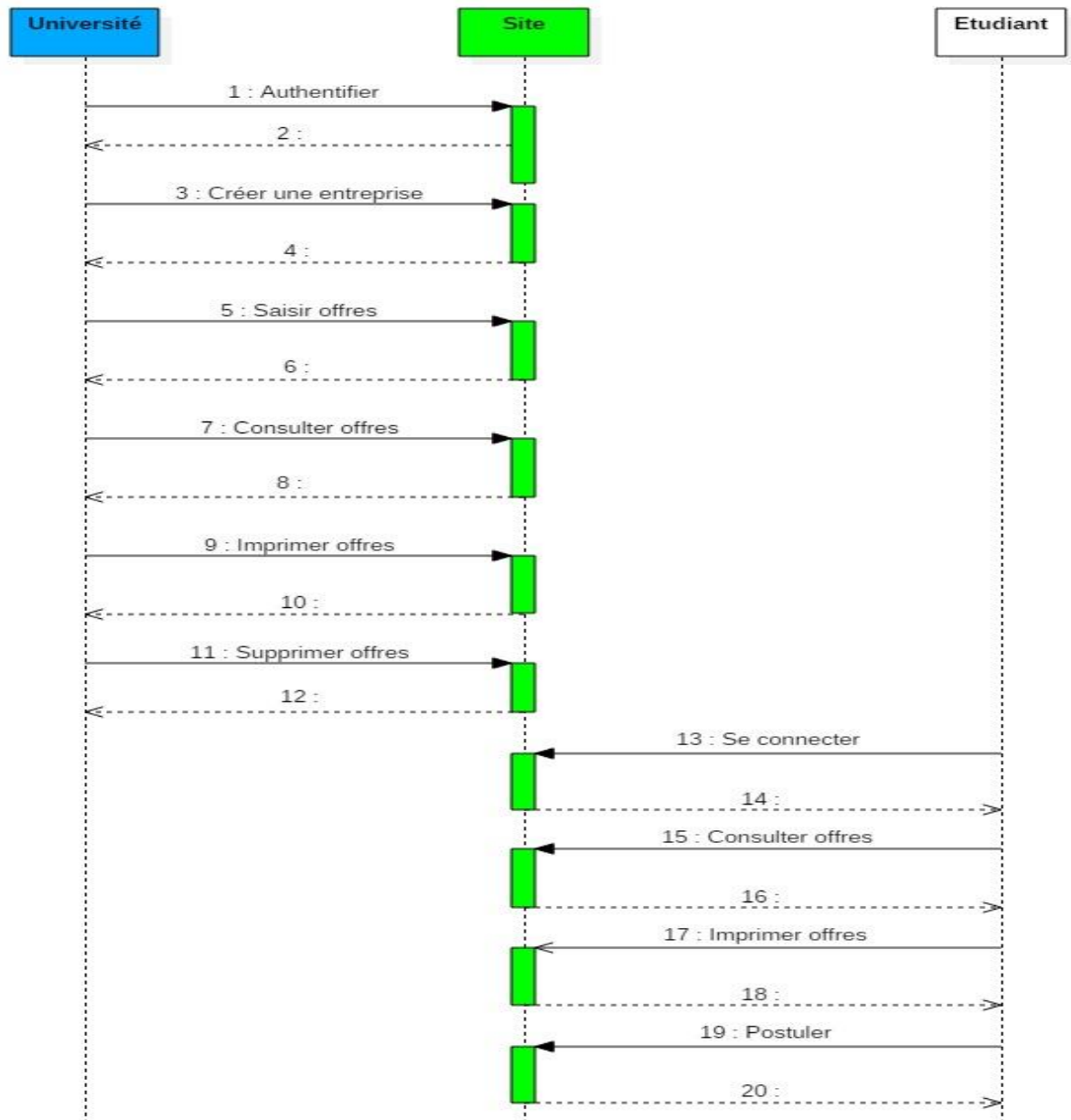
Dans notre diagramme de cas d'utilisation, nous avons pris en compte deux principaux acteurs qui sont l'Université (l'administrateur) et l'Étudiant (l'utilisateur), celui-ci hérite de certain droit de l'Université comme s'authentifier et consulter les offres de stage.

- L'université peut créer une entreprise, éditer une entreprise, supprimer une entreprise, s'authentifier et consulter,

saisir, supprimer, éditer et imprimer une offre.

- L'Étudiant peut s'authentifier, consulter, imprimer et s'inscrire à une offre de stage

Diagramme de séquence :



Le diagramme de séquence récapitule les différentes actions possibles sur une ligne de vie en montrant l'interaction entre les acteurs et le serveur. Nous avons d'abord schématisé les actions possible faites par l'université puis celle faite par l'étudiant (plusieurs de actions peuvent être faites sur la même ligne de vie, cependant nous n'avons pas pu le schématisé sur starUML).

Toutes les actions partent soit de l'Université soit de l'Étudiant et vont vers le serveur (le site) qui représente la Base de Données.

BDD

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> 1	<u>ID ENTREPRISE</u>	int(32)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/> 2	VILLE	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 3	RUE	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 4	CODE_POSTAL	int(32)			Oui	NULL	
<input type="checkbox"/> 5	SECTEUR	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 6	NUM_TEL	int(32)			Oui	NULL	
<input type="checkbox"/> 7	NOM	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 8	EMAIL	char(32)	latin1_swedish_ci		Oui	NULL	

La table Entreprise contient l'ensemble des informations pour définir une entreprise (Nom, Adresse, Numéro de téléphone), la clé primaire est le champ ID_Entreprise qui est un int en auto incrément.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> 1	<u>ID OFFRE</u>	int(32)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/> 2	ID_ENTREPRISE	int(32)			Non	Aucune	
<input type="checkbox"/> 3	LIBELLE	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 4	DESCRIPTIF	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 5	DOMAINE	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/> 6	DUREE	int(32)			Oui	NULL	
<input type="checkbox"/> 7	DATE	char(32)	latin1_swedish_ci		Oui	NULL	

La table Offre_Stage contient l'ensemble des informations pour définir une Offre de Stage (Libelle, descriptif, durée), la clé primaire est le champ ID_Entreprise qui est un int en auto incrément. Elle possède aussi une clé étrangère (ID_Entreprise en référence à ID_Entreprise de la table Entreprise).

	#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/>	1	<u>ID_POSTULER</u>	int(32)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/>	2	ID_OFFRE	int(32)			Oui	NULL	
<input type="checkbox"/>	3	ID_UTILISATEUR	int(32)			Oui	NULL	

La table Postuler contient l'ensemble des informations pour définir une postulation à une offre de stage (ID de l'offre, ID de l'utilisateur), la clé primaire est le champ ID_Postuler qui est un int en auto incrément. Elle possède aussi deux clés étrangères (ID_Utilisateur en référence à ID_Utilisateur de la table Utilisateur et ID_Offre et référence à ID_Offre de la table Offre_Stage).

	#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/>	1	<u>ID_UTILISATEUR</u>	int(32)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/>	2	PSEUDO	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/>	3	TYPE	tinyint(1)			Oui	NULL	
<input type="checkbox"/>	4	MDP	char(32)	latin1_swedish_ci		Oui	NULL	
<input type="checkbox"/>	5	NOM	char(50)	latin1_swedish_ci		Non	Aucune	
<input type="checkbox"/>	6	PRENOM	char(50)	latin1_swedish_ci		Non	Aucune	
<input type="checkbox"/>	7	ADRESSE	char(100)	latin1_swedish_ci		Non	Aucune	
<input type="checkbox"/>	8	NUM_TEL	int(11)			Non	Aucune	
<input type="checkbox"/>	9	CODE_POSTAL	int(11)			Non	Aucune	

La table Utilisateur contient l'ensemble des informations pour définir Utilisateur (Pseudo, Nom, mot de passe), la clé primaire est le champ ID_Utilisateur qui est un int en auto incrément.

Les classes

Classe BDD : Classe permettant la connexion et la déconnexion à la Base de donnée, et contenant la totalité des méthodes ayant une interaction direct avec la base de donnée (méthode permettant la modification, la suppression, la création ou la lecture d'un enregistrement).

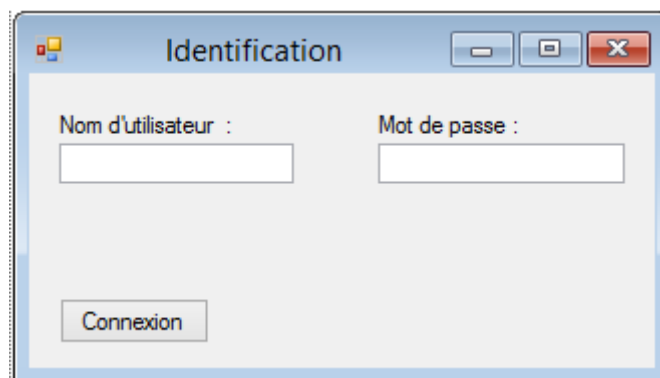
Classe Entreprise : Classe étant l'équivalent de la table Entreprise présente dans la base de données, elle contient les mêmes champs que celle-ci (IDEntreprise, Nom, Adresse...)

Classe OffreStage : Classe étant l'équivalent de la table Offre_Stage présente dans la base de données, elle contient les mêmes champs que celle-ci (IDOffreStage, IDEntreprise, Domaine...)

Class Utilisateur : Classe étant l'équivalent de la table Utilisateur présente dans la base de donnée, elle contient les mêmes champs que celle-ci (Pseudo, MDP...)

Class DonneeClass : Classe statique contenant un objet statique ainsi qu'une liste d'objet statique de chaque class présente dans le programme (par exemple la classe BDD continue un objet statique Entreprise et une liste d'objet statique Entreprise).

Interface graphique



Le formulaire d'identification permet à l'utilisateur de se connecter pour accéder à l'application, pour se connecter il doit indiquer son nom d'utilisateur (ou pseudo) ainsi que son mot de passe.

Le programme va chercher dans la base de données un enregistrement correspondant aux informations indiquées par l'utilisateur, si l'enregistrement existe l'utilisateur est redirigé vers le menu principal de l'application, sinon le programme affiche un message d'erreur



Le menu principal de l'application permet à l'utilisateur d'accéder à l'ensemble des sections, en fonction de ses droit d'accès il pourra accéder ou non à certaine. Si l'utilisateur est un élève, il pourra accéder seulement aux sections de consultation (Consulter les entreprises et consulter les offres de stage), l'utilisateur pourra y accéder seulement s'il est un membre de l'université.

The screenshot shows a window titled "FormSaisirOffre" with the following fields:

- ID Offre :
- Nom de l'entreprise :
- Domaine de l'offre :
- Ou :
- Libelle de l'offre :
- Date de debut de stage :
- Durée :
- Description :

At the bottom left, there is a button labeled "Envoyer".

Si l'utilisateur membre de l'université clique sur le bouton « Saisir une offre de stage », il accède à une page lui permettant de créer un nouvel enregistrement dans la table Offre_Stage de la base de données. Il y a une différence par rapport au formulaire de création d'une entreprise car dans ce formulaire l'utilisateur peut choisir d'utiliser un domaine déjà présent dans la base de données en choisissant dans la liste déroulante de droite, ou d'utiliser un nouveau domaine en l'écrivant dans la zone de texte de gauche.

FormCreerEntreprise

ID Entreprise :

Nom de l'entreprise :

Adresse :

Code Postal :

Ville :

Email :

Numéro de téléphone :

Secteur d'activité :

Envoyer Annuler

Si le membre de l'université sélectionne le bouton « Créer une entreprise », le programme affiche un formulaire permettant à l'utilisateur d'ajouter une nouvelle entreprise dans la table Entreprise de la base de données.

FormConsulterEntreprise

Nom de l'entreprise :

Adresse :

Code Postal :

Ville :

Email :

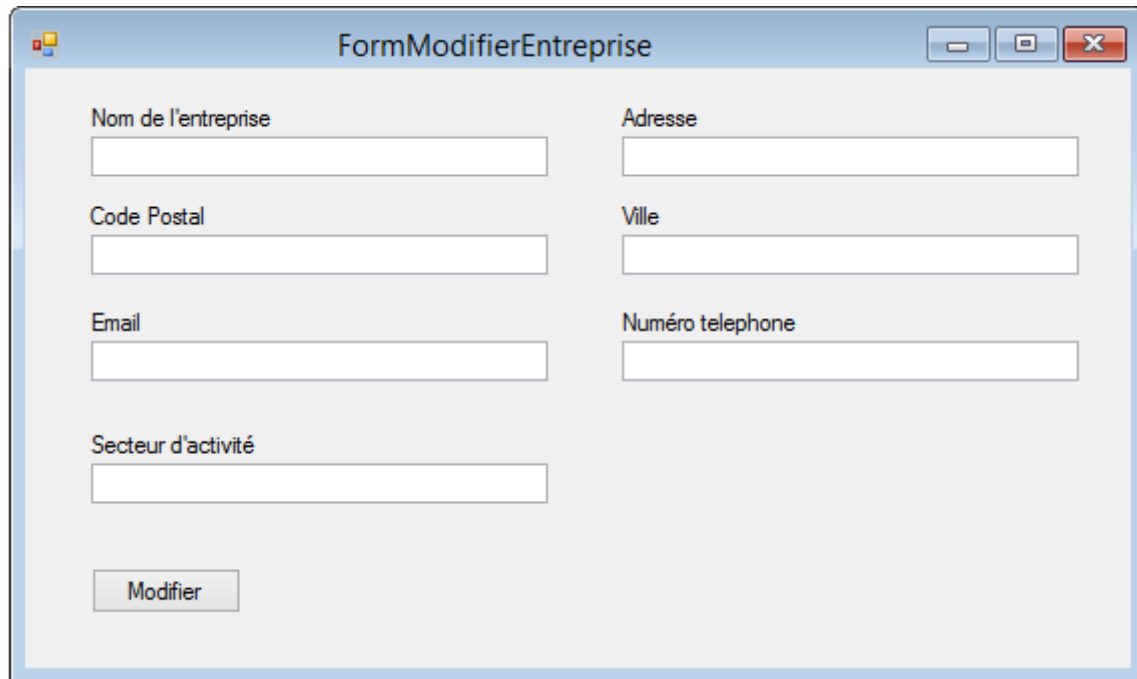
Numéro telephone :

Secteur d'activité :

Suivant Précédant Supprimer l'entreprise Modifier l'entreprise

Concernant le formulaire de consultation des entreprises, le programme va remplir chaque zone de texte avec les informations de l'entreprise concerné (En commençant par la première entreprise présente dans la base de

données). L'utilisateur peut passer à l'entreprise suivante en cliquant sur le bouton « Suivant » et accéder à celle d'avant en cliquant sur « précédant », les boutons permettant la suppression et la modification sont quant à eux accessibles uniquement à l'utilisateur étant membre de l'université



The image shows a software window titled "FormModifierEntreprise". It contains a form with the following fields and labels:

- Nom de l'entreprise
- Adresse
- Code Postal
- Ville
- Email
- Numéro telephone
- Secteur d'activité

At the bottom left of the form is a button labeled "Modifier".

Si l'utilisateur membre de l'université clique sur le bouton « modifier l'entreprise », le programme affiche un formulaire et remplit chaque zone de texte avec les informations de l'entreprise sélectionné dans le formulaire précédant. L'utilisateur pourra ensuite cliquer sur le bouton « modifier » pour que le programme enregistre les modifications dans la base de données.

The screenshot shows a Windows-style window titled "FormConsulterOffreStage". Inside the window is a form with the following fields and buttons:

- Nom de l'entreprise**: A text input field.
- Domaine de l'offre :**: A text input field.
- Ville :**: A text input field.
- Libelle de l'offre :**: A text input field.
- Date de debut de stage :**: A text input field.
- Durée :**: A text input field.
- E-Mail :**: A text input field.
- Description :**: A large text area.
- Suivant**: A button.
- Précédant**: A button.
- Postuler à l'offre**: A button.
- Supprimer l'offre**: A button.
- Modifier l'offre**: A button.

Même chose que pour le formulaire de consultation des entreprises, mais cette fois pour les offres de stages. Mais se formulaire contient un bouton de plus permettant à l'utilisateur de type élève de pouvoir postuler à une offre de stage, si il clique sur le bouton « postuler » une demande de confirmation va s'afficher à l'écran et une fois cette confirmation valider, le programme ajoute un enregistrement dans la table.

The screenshot shows a Windows-style window titled "FormModifierOffreStage". It contains the following fields and controls:

- ID Offre :** A single-line text input field.
- Nom de l'entreprise :** A dropdown menu.
- Domaine de l'offre :** A single-line text input field.
- Ou :** A dropdown menu.
- Libelle de l'offre :** A single-line text input field.
- Date de debut de stage :** A date picker showing "vendredi 15 avril 2016".
- Durée :** A single-line text input field.
- Description :** A large multi-line text area.
- Modifier :** A button located at the bottom left of the form.

En cliquant sur le bouton « modifier » le membre de l'université pourra accéder à une page permettant de modifier l'offre de stage précédemment sélectionné. Comme pour la création d'une offre de stage, l'utilisateur peut choisir soit un domaine existant, soit en insérer un nouveau.

Le programme affiche un formulaire et remplit chaque zone de texte avec les informations de l'offre de stage. L'utilisateur pourra ensuite cliquer sur le bouton « modifier » pour que le programme enregistre les modifications dans la base de données.

Conclusion

En conclusion cette application permet à chaque utilisateur de faire exactement ce dont il a besoin concernant la gestion des stages, tout en respectant une limite imposé par les droits d'accès.

Notre organisation nous a permis de finir le projet à temps, en respectant les contraintes de temps et les diverses restrictions. Lors de la réalisation de ce projet nous avons appris à manier différents outil comme Git Hub, StarUML ou encore C#