

## Projet Java : Logiciel d'offre de stage

Identification client

×

### Guide pratique des stages étudiants

VOUS ÊTES...

ÉTUDIANT(E)



UN ORGANISME D'ACCUEIL  
(ENTREPRISE, ASSOCIATION,  
COLLECTIVITÉ...)



UN ÉTABLISSEMENT  
D'ENSEIGNEMENT  
OU DE FORMATION  
SUPÉRIEUR



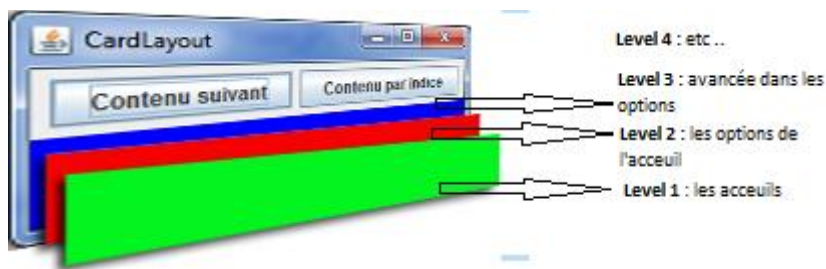
## Introduction

Pour ce projet l'objectif était de réaliser une plateforme gérée par une université (l'administrateur) et qui aurait pour fonction la mise en relation des étudiants et des entreprises afin qu'ils aient, l'un comme l'autre, une certaine visibilité. Les étudiants peuvent alors candidater et les entreprises trouver les candidats adaptés à leurs offres.

### L'aspect Graphique

Nous avons choisi la bibliothèque Swing pour la vue, au regard de sa portabilité et de sa facilité d'utilisation. Nous avons appris plus tard qu'il était possible d'utiliser Java FX mais notre projet étant déjà avancé de plusieurs heures, nous avons préféré continuer à enrichir notre interface plutôt que de repartir du début.

Pour gérer nos Panel, nous avons utilisé le CardLayout, c'est un moyen simple et efficace qui permet de superposer plusieurs Panel tout en ayant la possibilité de changer de vue à n'importe quel moment.



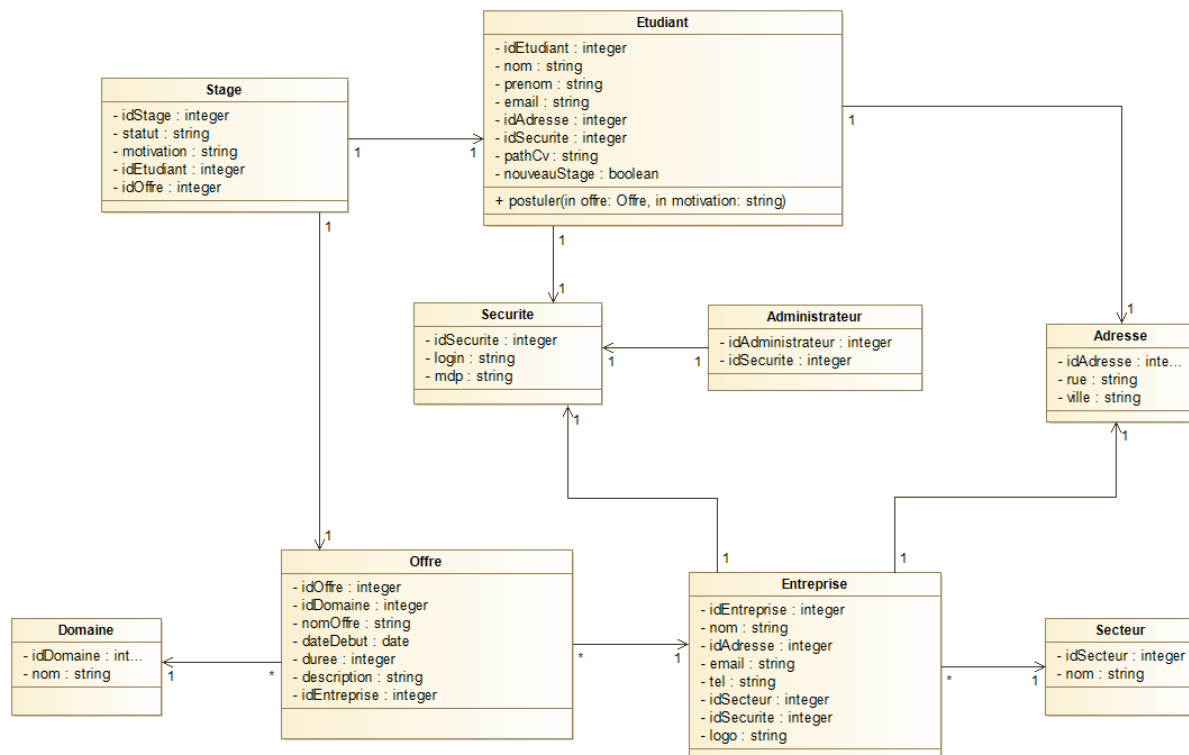
La fenêtre (*JFrame*) contient uniquement le panel principal (*mainPanel*) qui lui contient les différents étages du programme : level\_1, level\_2, level\_3, level\_4.

Ainsi plus l'utilisateur avance dans le programme plus le *mainPanel* contient de "level" instanciés. Les niveaux sont conservés en *static* par la classe *Fenetre* afin de permettre un retour au niveau précédent du programme à partir de n'importe quel page.

## La base de données

Pour pouvoir manipuler et enregistrer toutes les informations du programme, un simple *serialisable* ne suffisait pas. Nous avons mis en place une base de données (BDD) qui nous permettra de faire toutes les requêtes possibles. Nous l'avons tout d'abord modélisé sous MySQL Workbench qui permet, à partir d'un diagramme de classe, d'écrire le code SQL de la BDD. Après ça, nous avons travaillé sous PhpMyAdmin car il est bien plus pratique pour remplir les données.

### Le diagramme de classe



Nous pouvons donc voir dans ce diagramme les propriétés suivantes :

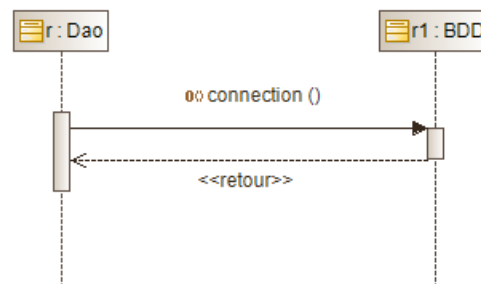
- Les 3 utilisateurs ont une classe sécurité qui leur fournit un login et un MDP
- Entreprise et étudiant ont une adresse
- Une offre possède une entreprise et un domaine
- L'étudiant et l'offre sont reliés grâce à l'objet Stage
- Une entreprise possède un secteur

## Les DAO

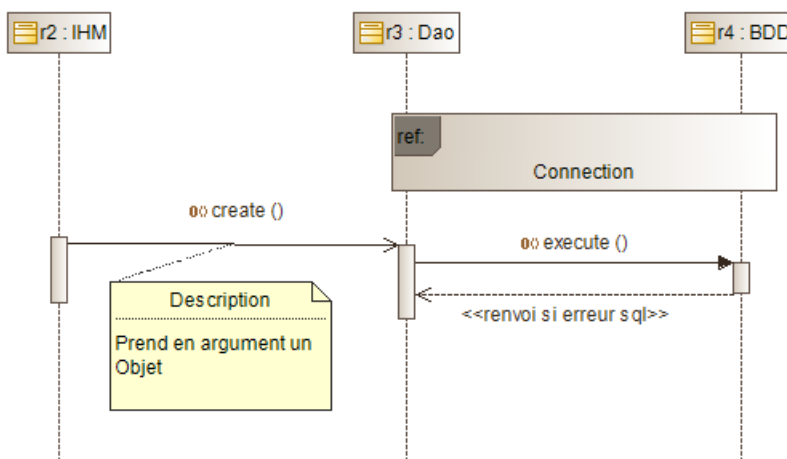
Les DAO sont les classes qui relient les objets de notre programme avec la base de donnée. Ils facilitent grandement les échanges en comportant une et une seule fois les requêtes SQL. il suffit ensuite d'instancier ces classes pour avoir accès au information de la BDD.

```
public class StageDAO implements DAO<Stage> {  
  
    Connection con;  
  
    final String create = "INSERT into stage (idEtudiant, idOffre, statut, motivation) VALUES (?";  
    final String update = "UPDATE stage SET idEtudiant=?, idOffre=?, statut=? WHERE id=?";  
    final String update_statut = "UPDATE stage SET statut=? WHERE id=?";  
    final String delete = "DELETE FROM stage WHERE id=?";  
    final String find = "SELECT * FROM stage WHERE id = ?";  
    final String find_double_id = "SELECT * FROM stage WHERE idEtudiant = ? AND idOffre = ?";  
    final String find_offre = "SELECT * FROM stage WHERE idOffre = ?";  
    final String find_etudiant = "SELECT * FROM stage WHERE idEtudiant = ?";  
    final String getAll = "SELECT * FROM stage;";  
  
    public StageDAO() {  
        // connexion avec BDD  
        con = Connection.connect();  
    }  
}
```

Tous les constructeurs des DAO lance la connexion à la base de données.

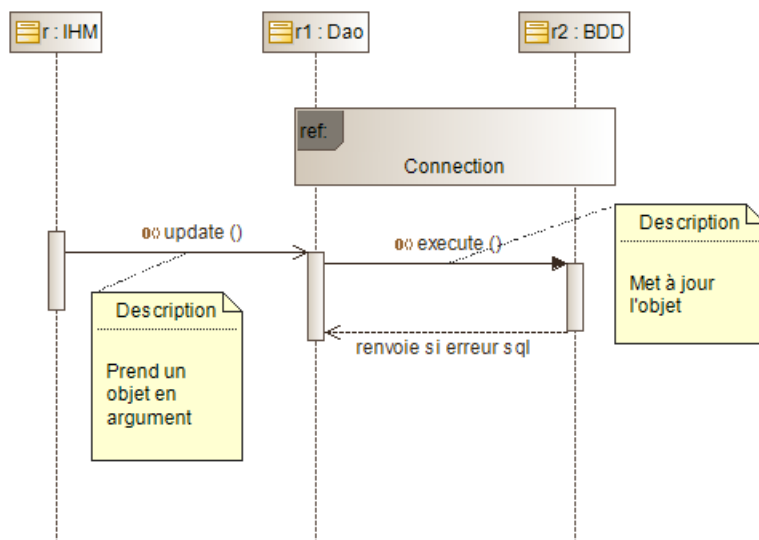
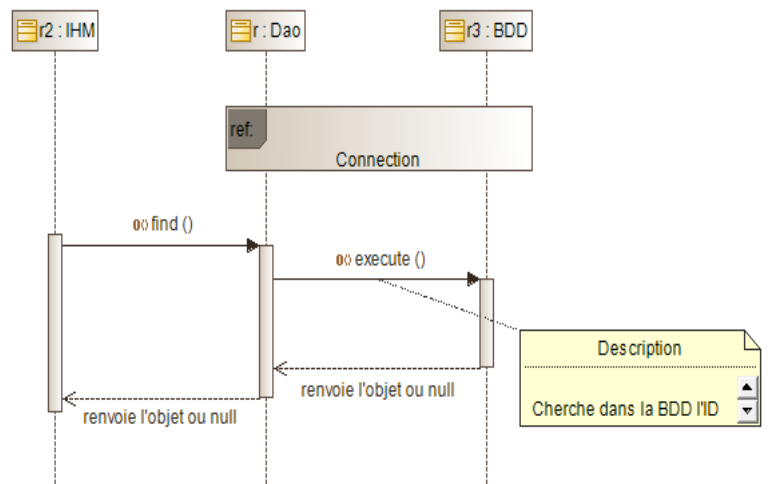


Voici quelques exemples de méthodes quelles comportent :



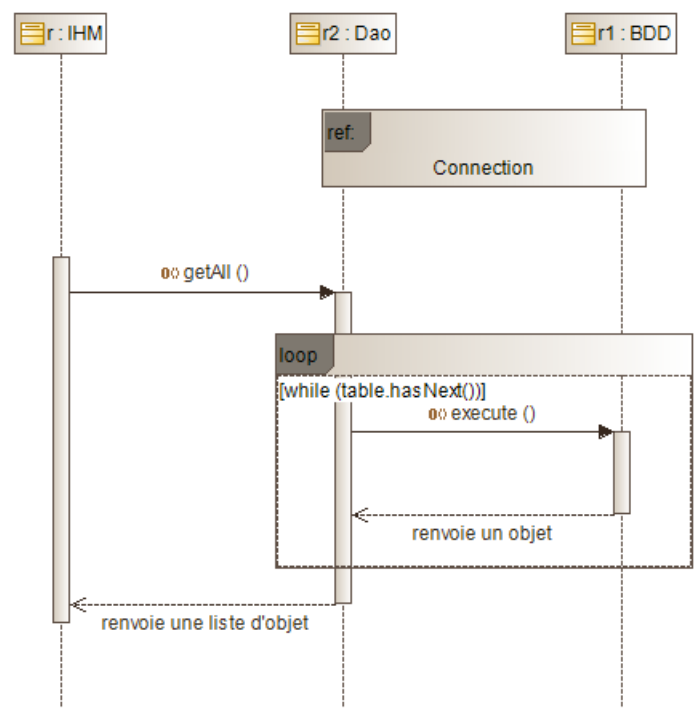
**create()** : Ajoute un nouvel élément a la base de données à partir d'un objet Java.

**find()** : Renvoie un objet Java en fonction du type de DAO choisit et de l'ID.

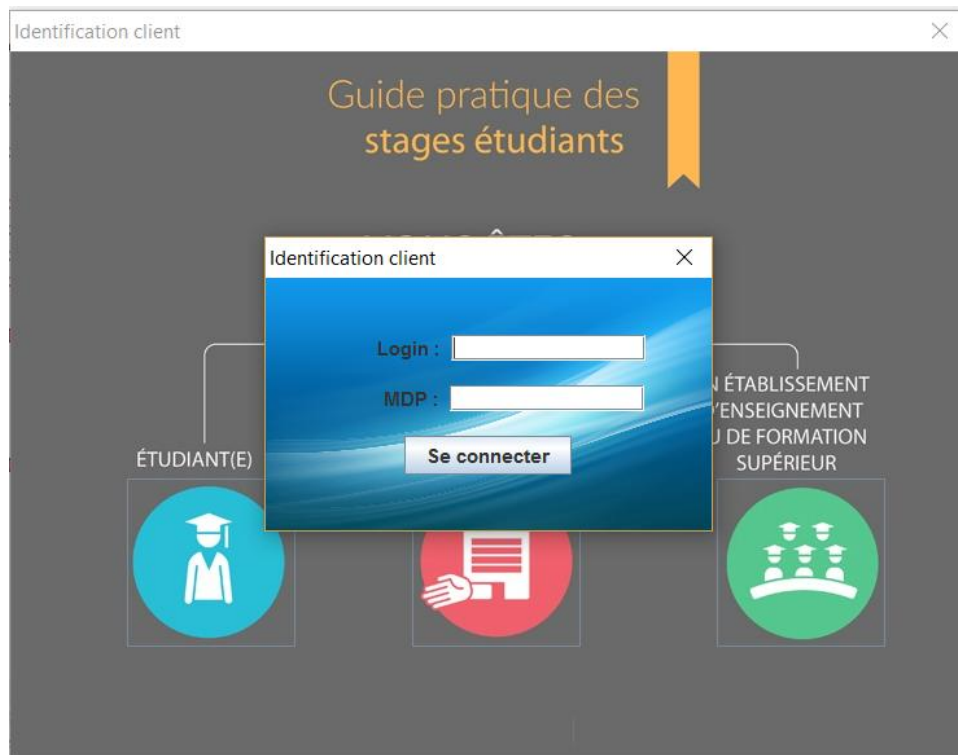


**update()** : Met à jour les informations d'un objet.

**getAll()** : Permet de récupérer tout une table dans une ArrayList.



## L'accueil et la connection



3 types de connection sont proposés :

- **Administrateur** : il permet de gérer les comptes de tout le monde, c'est lui qui crée et supprime les comptes des utilisateurs. Il peut également gérer les offres.
- **Entreprise** : une entreprise qui se connecte pour proposer des offres de stages, elle peut créer des offres, les modifier, accepter des étudiants.
- **Étudiant** : un étudiant se voit proposer des offres, il peut les consulter, postuler et gérer son profil.

*Le diagramme de cas d'utilisation est plus explicite :*

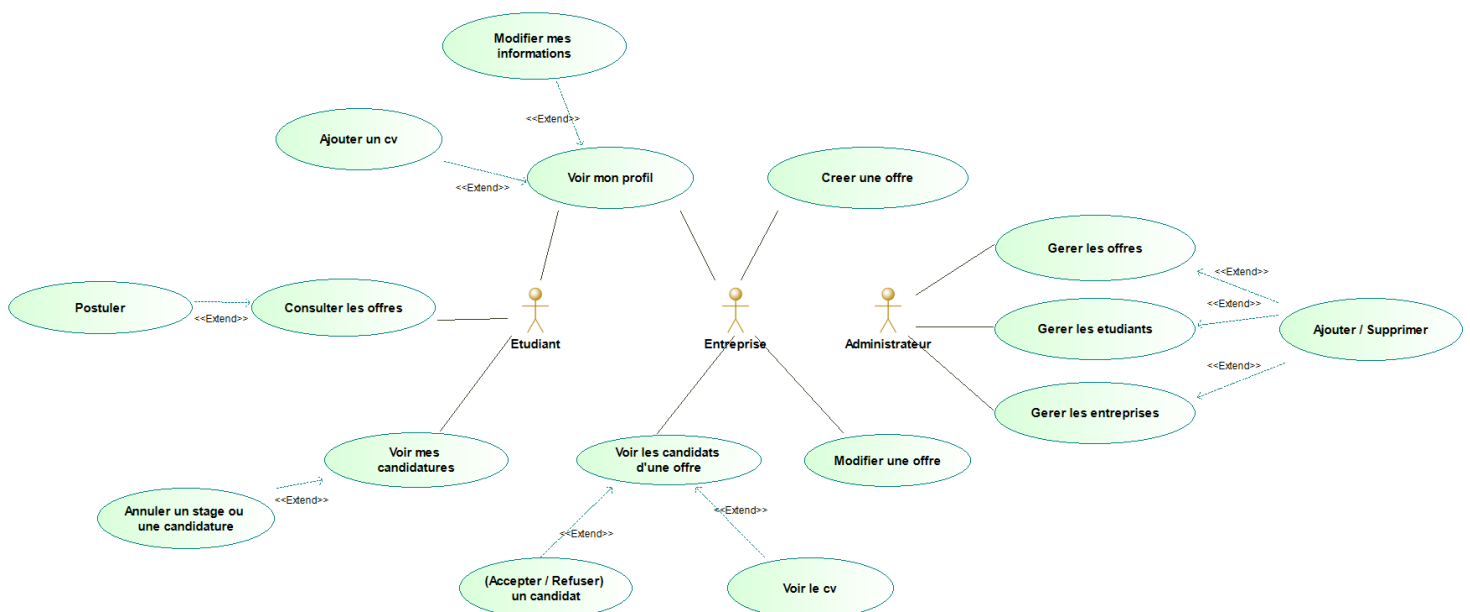
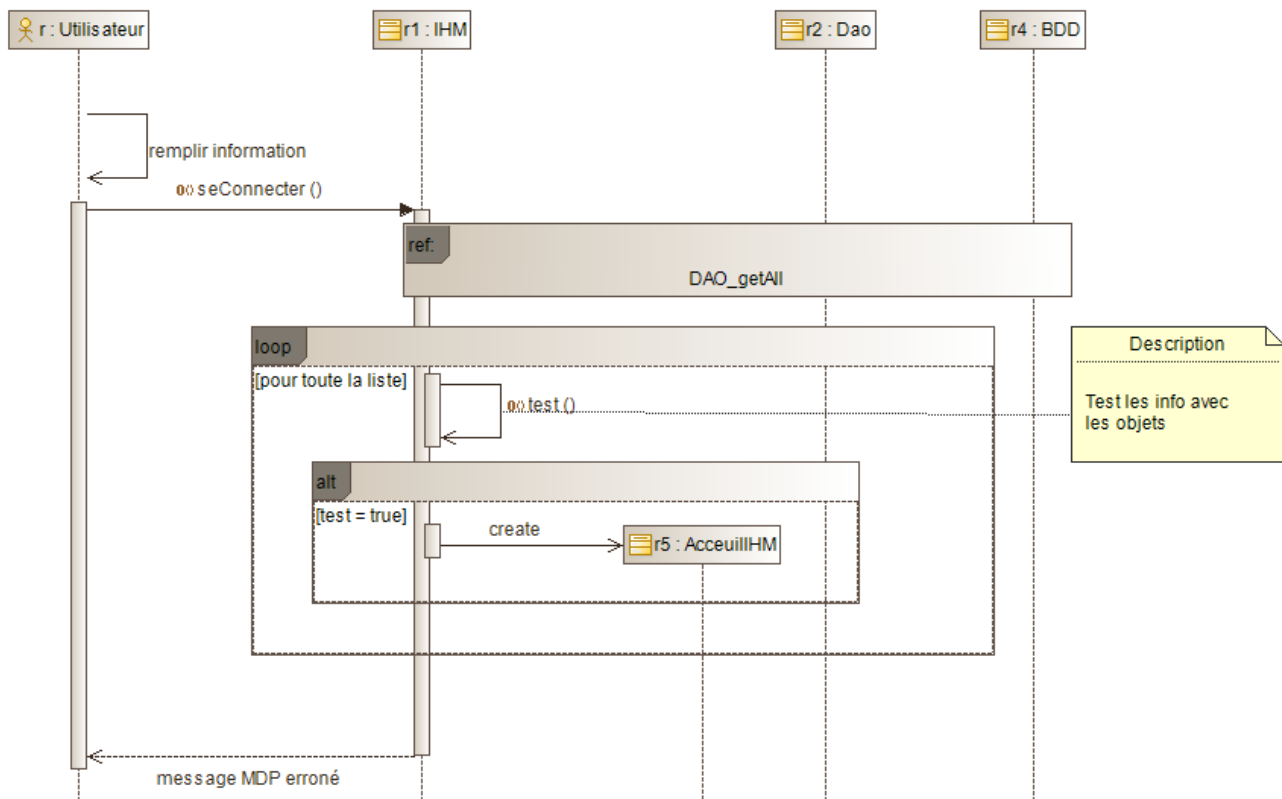


Diagramme de séquence d'une connection, peut importe l'utilisateur :



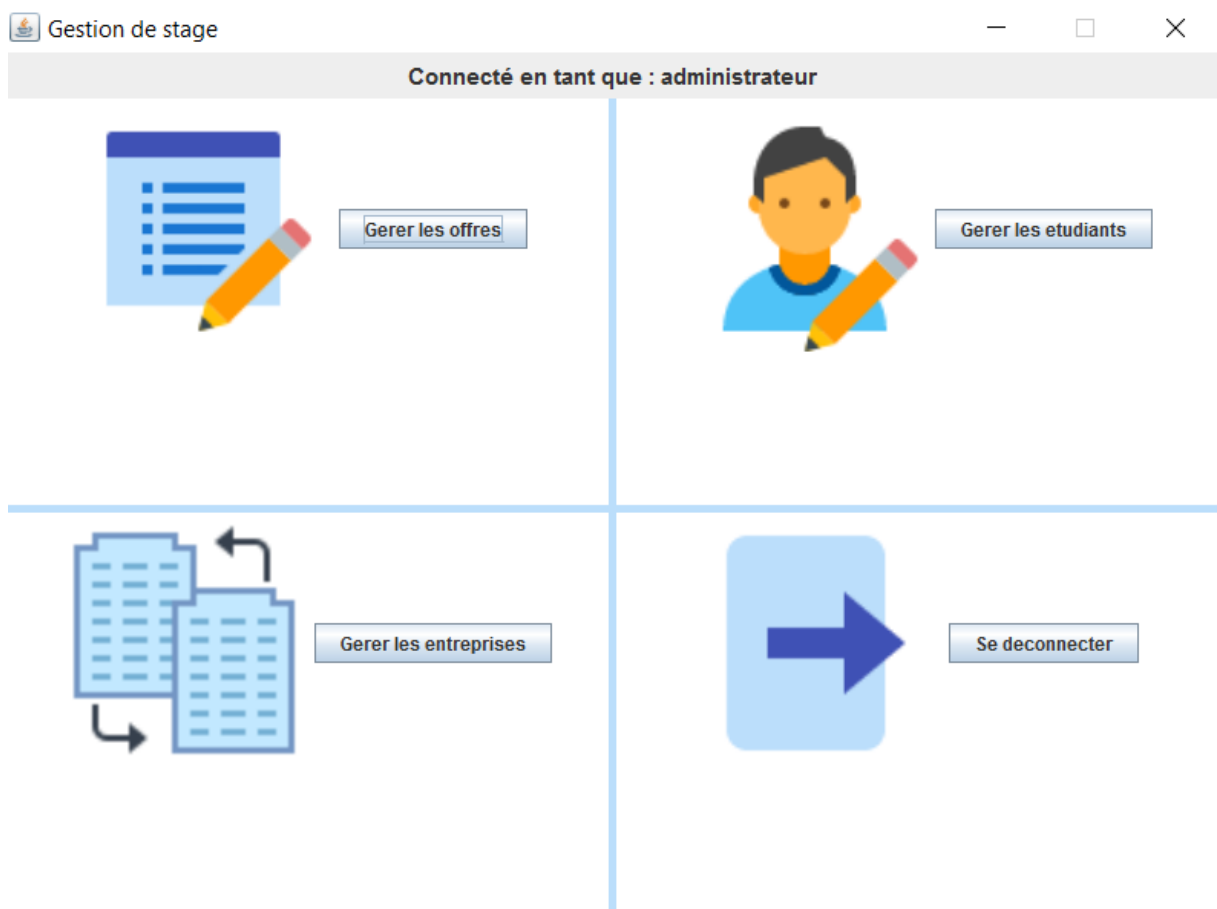
Lors de la connection, le mot de passe est crypté en MD5 dans la base de donnée. Il est impossible de décrypter du MD5. La solution est donc de crypter le MDP entrant (JPassword) et le comparer avec le MDP contenu dans la base de données.

Pour comparer le bon MDP on cherche a savoir si le login est également le bon, il y a donc une boucle qui va tester tous les objets de la table spécifiée.

## L'administrateur (l'université)



### *Le menu Administrateur*

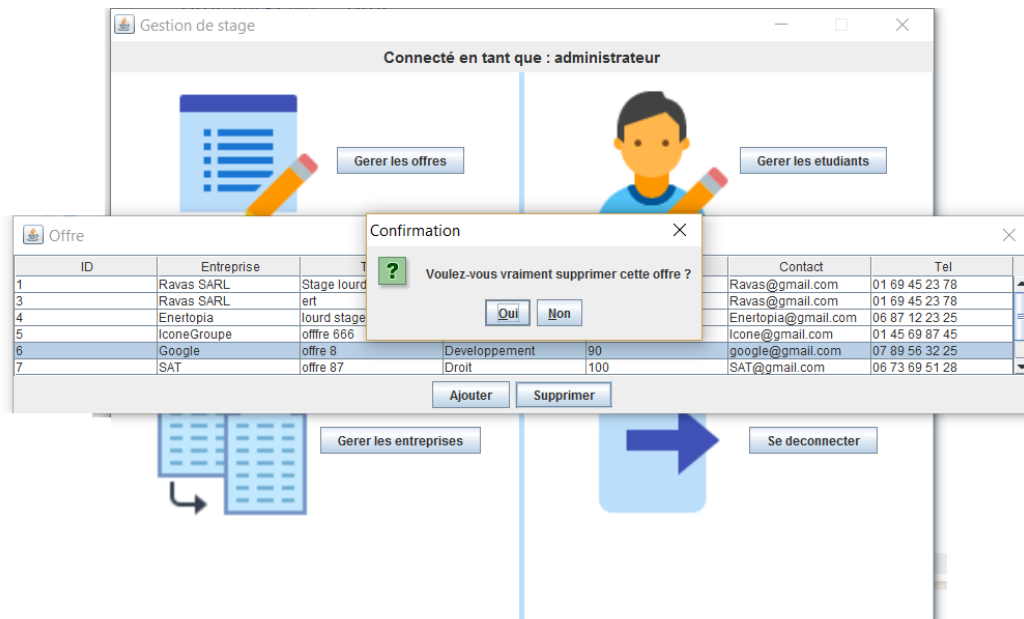


Lorsqu'on choisit une de ces options d'administration , une JTable s'ouvre et affiche le contenu de la base de données pour une table choisit. Il peut à tout moment se déconnecter ce qui le renvera vers l'accueil de connection.

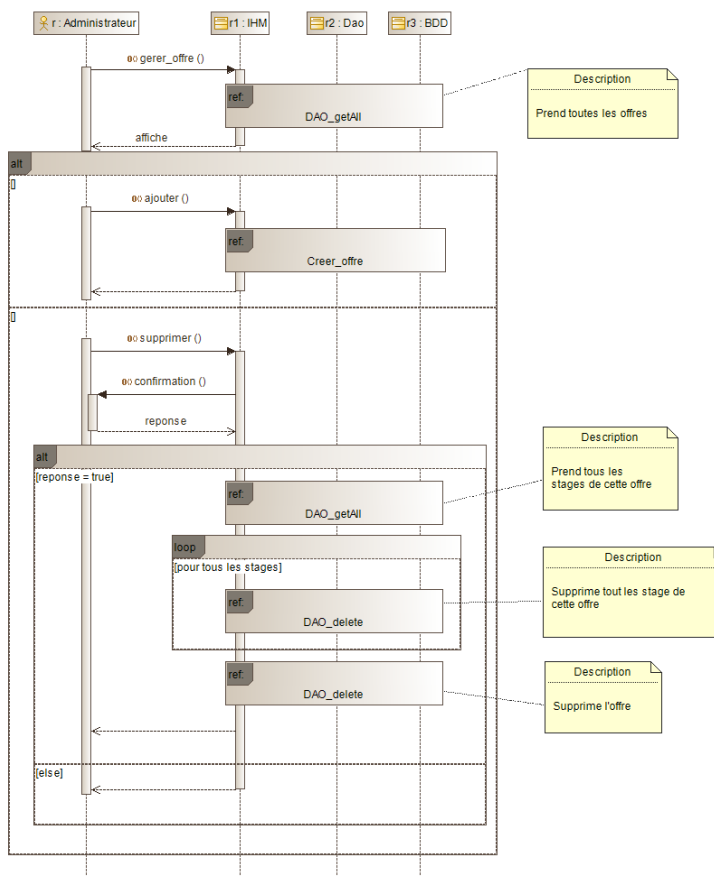


Ici par exemple lorsque nous choisissons de "gérer les offres" :

Il suffit de cliquer sur une ligne pour que la Jtable exécute l'action delete() des DAO avec l'ID de la ligne sélectionnée. Ainsi l'objet, d'efface. (ici l'offre avec l'ID 6)



### Diagramme de séquence d'une suppression d'entreprise



Les actions de suppressions prennent en compte toutes les tables avec lesquelles elles étaient reliées et supprime ainsi les attributs qui étaient propre à la ligne supprimée (Ex : son adresse, ses identifiants, ses offres , ses stages)

## Ajouter une offre

Offre

Nom de l'offre

altedia consultants

Entreprise : Altedia ESR

- Ravas SARL
- Altedia ESR
- Enertopia
- IconeGroupe
- Google
- SAT

Domaine :

Durée : 0 Jour(s)

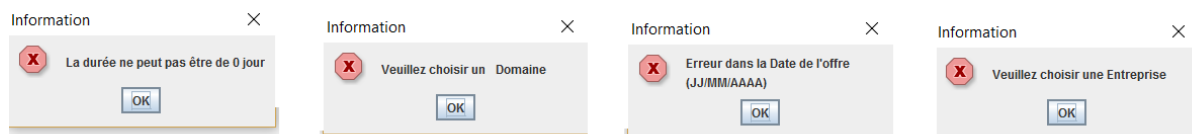
Date de Debut : jj/mm/aaaa

Enregistrer Quitter

Cette nouvelle page permet à l'administrateur de créer une nouvelle offre à la place de n'importe quelle entreprise. Ce panel existe aussi pour les entreprises mais il est restreint aux identifiants de l'entreprise connectée.

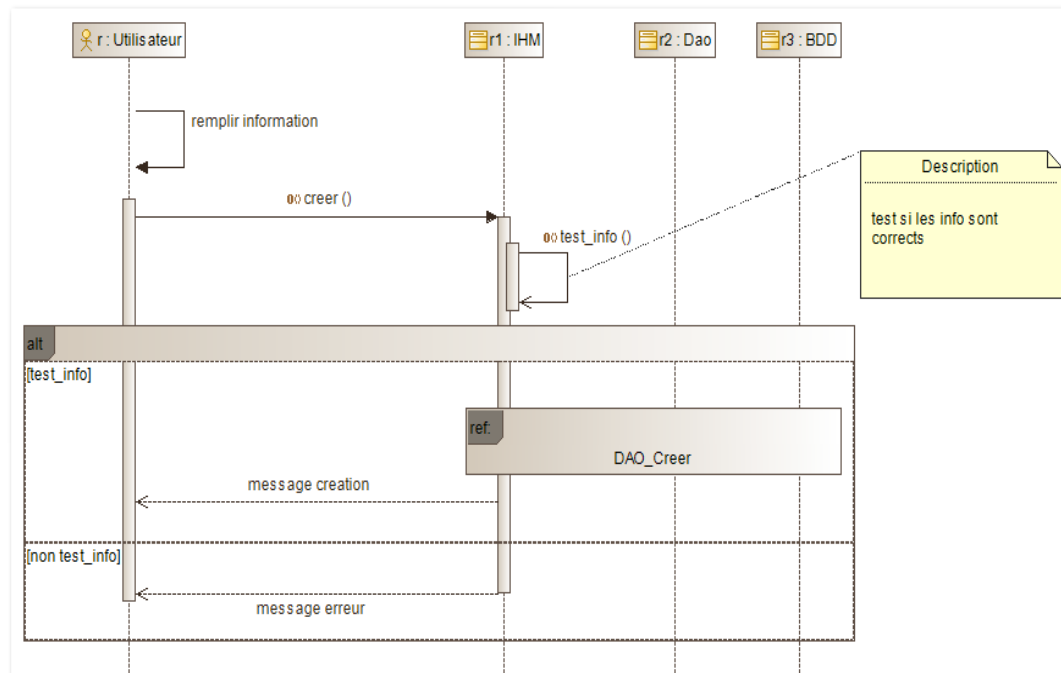
### Pour qu'une offre soit valide elle doit respecter différentes exceptions :

- Le domaine de l'offre doit être sélectionné.
- L'entreprise doit être sélectionnée (le logo se met à jour dynamiquement).
- La durée doit être composée uniquement de chiffres (exclu 0).
- La date doit être valide et au bon format.



Un message d'erreur s'affichera tant que l'on voudra créer une offre qui ne respecte pas ces trois conditions

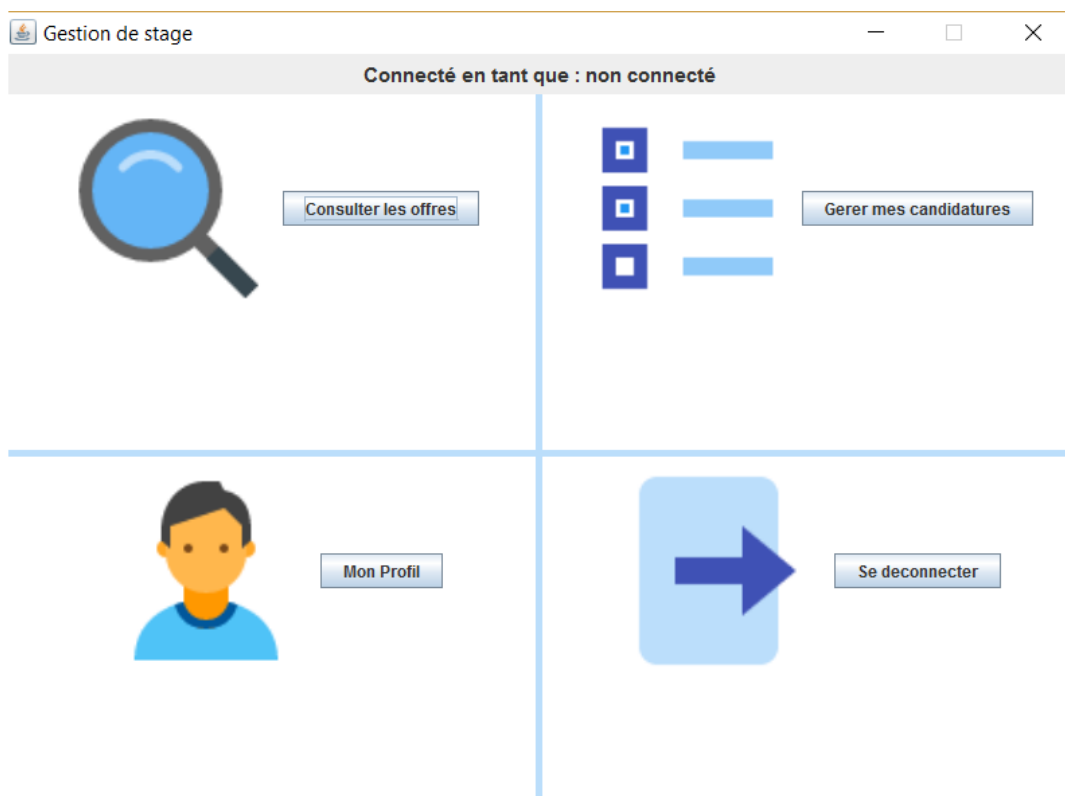
## La création d'une offre



## L'étudiant



## La page d'accueil



## ProfilPanel\_Etudiant

Gestion de stage

Connecté en tant que : non connecté

Nom :	ETUDIANT
Prenom :	etudiant
Pseudo :	etudiant
Adresse :	1 rue de la Tour des dames
Mot de passe :	~Crypté~
Email :	etudiant@gmail.com
Ville :	Paris

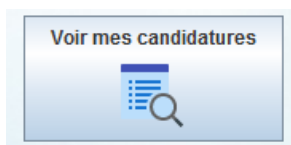
Quitter Modifier Voir mes candidatures Ajouter un CV

Voici donc à quoi ressemble le profil d'un étudiant, celui d'une entreprise est très similaire. la classe qui pour l'entreprise hérite d'ailleurs de celle-ci.

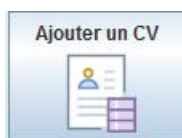
Les différentes fonctions sont :



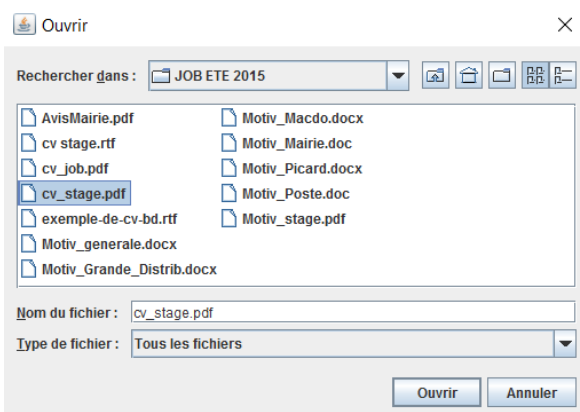
Transforme les informations en *JTextField* ce qui permet de les modifier puis de les enregistrer. Nous testons bien sur si les informations ont changées ou non.



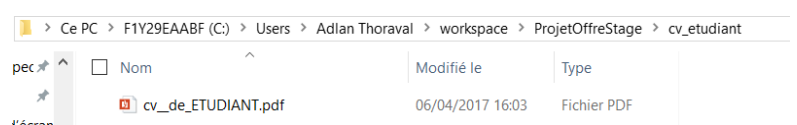
Nous présenterons cette page plus tard.



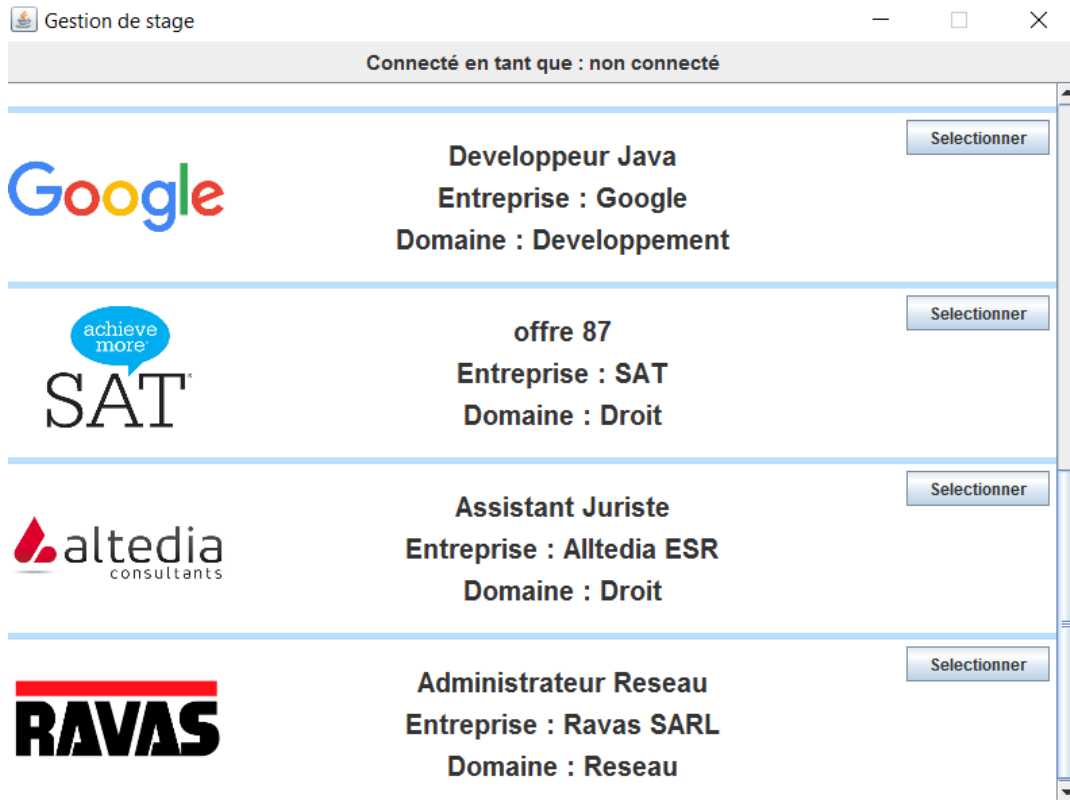
Ouvre un *JFileChooser* et permet ainsi de localiser un fichier pdf dans le système de fichier de l'utilisateur, si il choisit un pdf le fichier sera copié dans le dossier du programme et son chemin (relatif) sera mémorisé. Dans le cas contraire, une annonce sera faite.



### Résultat après la copie de cv\_stage.pdf

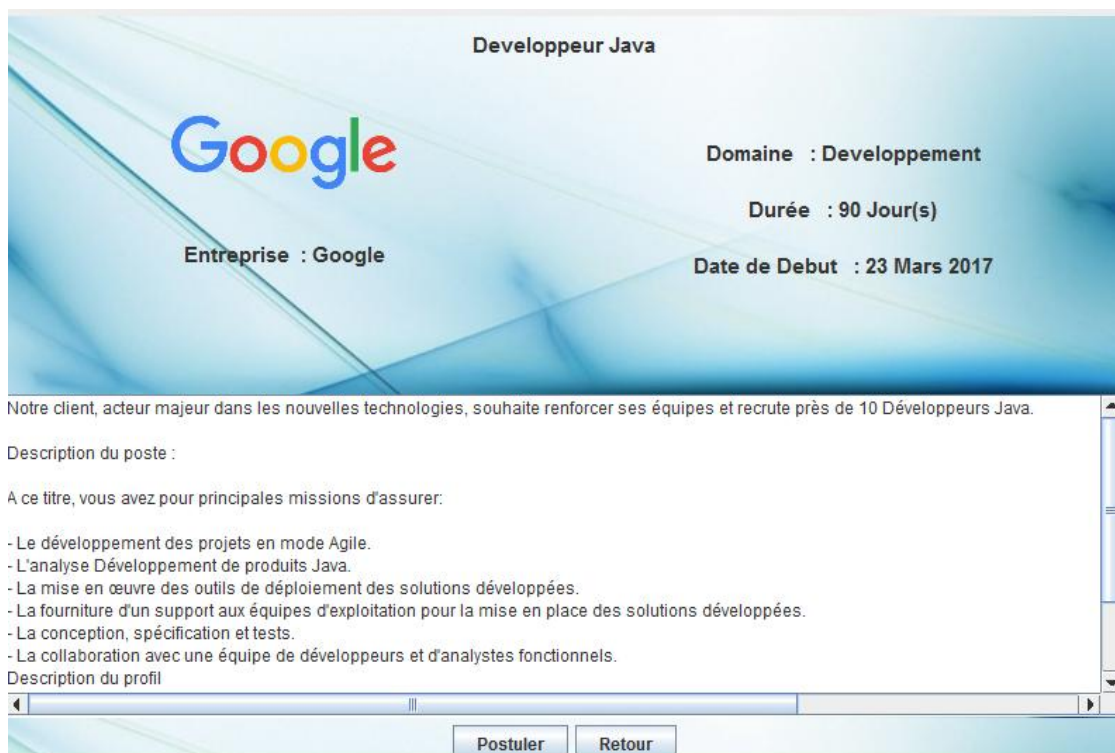


Voir les offres disponibles

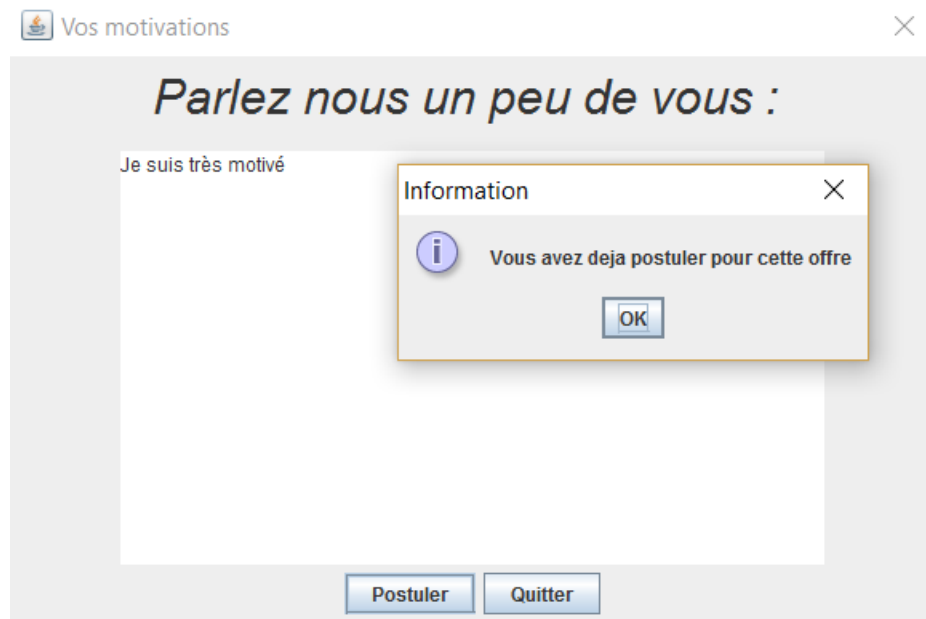


Les offres s'affiche dans un *JscrollPanel* grâce aux constructeurs présent dans : *Consulter\_Panel*.

Le bouton sélectionner affiche les détails de l'offre et la possibilité de postuler à cette dernière :

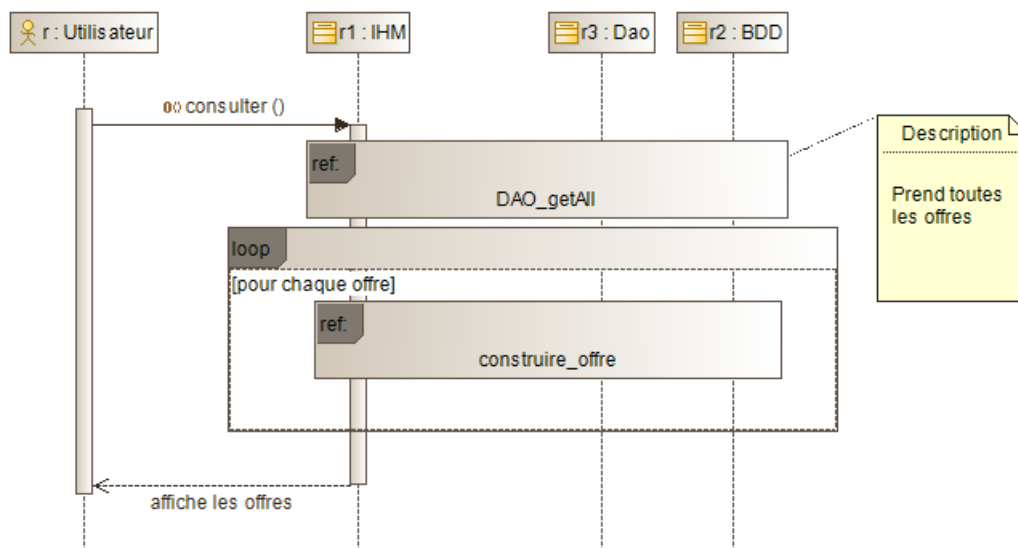


Enfin il pourra laisser un message que l'entreprise verra lorsqu'elle tombera sur sa candidature (il s'agit de sa lettre de motivation)



Une candidature étant unique , le système vérifiera si elle n'existe pas déjà dans la table des stage.

*Model selon lequel sont construite les offres*



[Retour](#)

## Gerer mes candidatures

	<b>En attente</b> Entreprise : Ravas SARL Stage bureautique	<a href="#">Voir les details</a>	<a href="#">Annuler la candidature</a>
	<b>Retenu</b> Entreprise : SAT offre 87	<a href="#">Voir les details</a>	<a href="#">Annuler ce stage</a>

Toutes les candidatures de l'étudiant sont enregistrées sur cette page. Elle permet de connaître le statut de notre demande et de pouvoir annuler un stage si nécessaire.

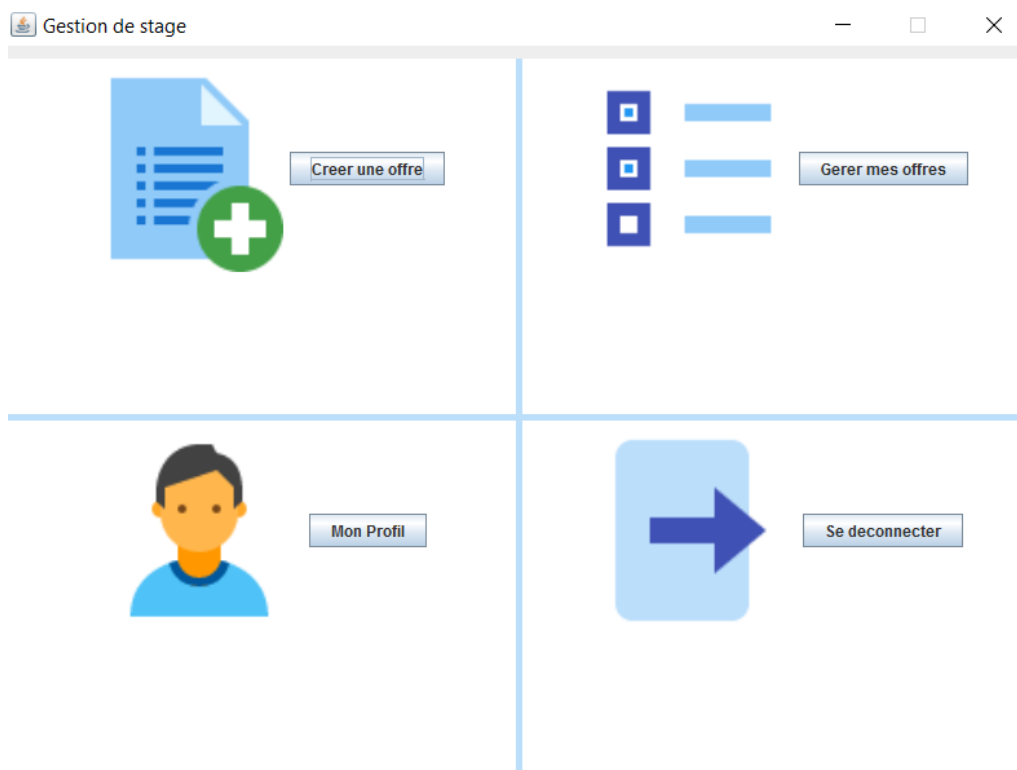
Elle permet également de retourner sur l'offre pour laquelle on a postulé.

Nous avons donc fait le tour des fonctionnalités proposer par le programme à l'étudiant.

## L'entreprise



### *La page d'accueil d'une entreprise*



Nous ne développerons pas pour ce cas-ci les parties "Créer une offre" (très similaire à celle de l'administrateur) ou "Mon Profil" qui est exactement celui de l'étudiant, sans la ligne d'information "prenom" et le bouton "ajouter un cv".



L'interet principal de ce menu est dans la partie "Gerer mes offres"

 Gestion de stage




[Retour](#)




## Gerer mes offres


	<b>Stage bureautique</b> Nombre de Postulant : 3 Domaine : Reseau	<a href="#">Modifier</a> <a href="#">Voir les candidats</a>
	<b>Architecte Reseau</b> Nombre de Postulant : 0 Domaine : Reseau	<a href="#">Modifier</a> <a href="#">Voir les candidats</a>
	<b>Administrateur Reseau</b> Nombre de Postulant : 0 Domaine : Reseau	<a href="#">Modifier</a> <a href="#">Voir les candidats</a>

Le bouton modifier permet à l'entreprise de retourner sur l'offre selectionnée pour en modifier certains critères : il s'agit du *Panel\_Offre* classique avec des champs pré-remplis.

Le bouton "Voir les candidats" nous redirige vers cette page :




 Gestion de stage



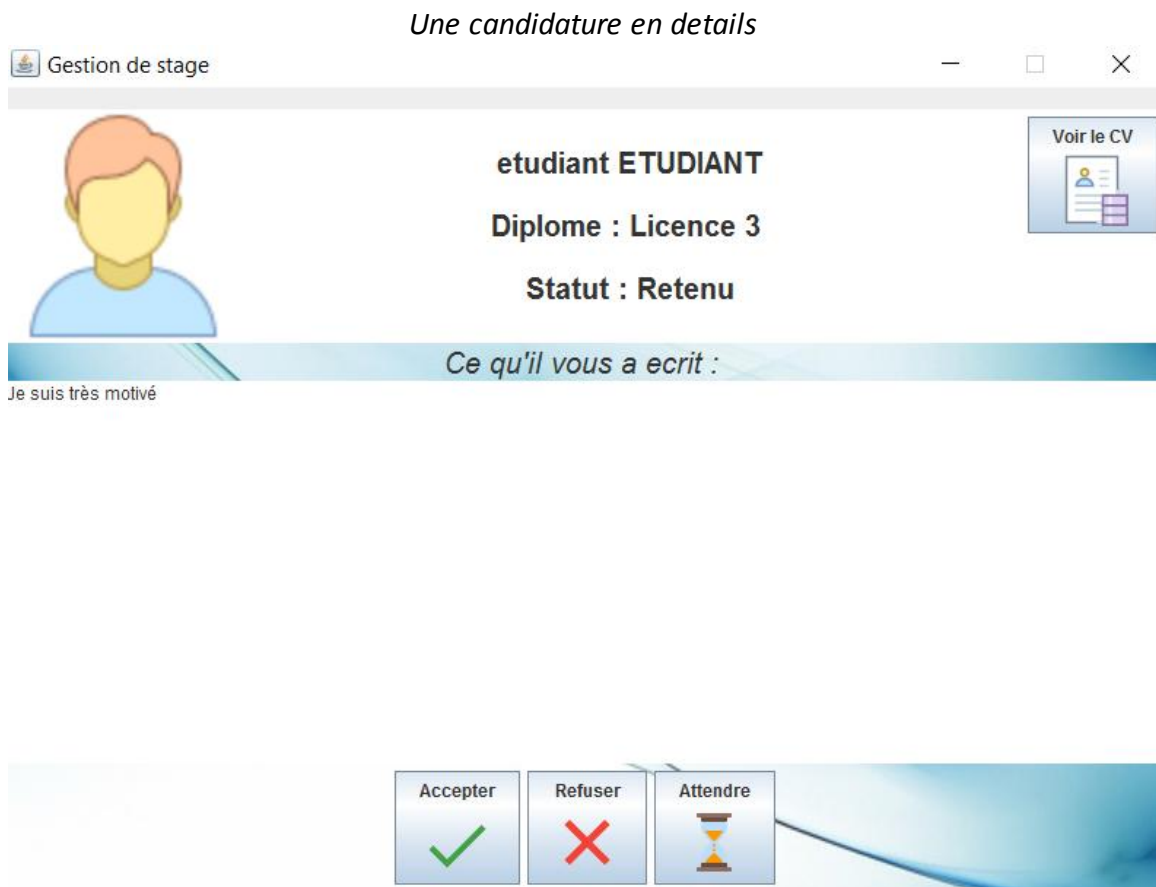


**Stage bureautique**  
Entreprise : Ravas SARL  
Domaine : Reseau

[Retourner aux offres](#)

	<b>Maxime Desbois</b> Diplome : Licence 3 Statut : Refuse	<a href="#">Voir sa candidature</a>
	<b>Adlan Thoraval</b> Diplome : Licence 3 Statut : En attente	<a href="#">Voir sa candidature</a>
	<b>etudiant ETUDIANT</b> Diplome : Licence 3 Statut : En attente	<a href="#">Voir sa candidature</a>

Ici sont chargés l'ensemble des étudiant ayant postulé sur l'offre présente au dessus du Panel. On peut cliquer sur "Voir sa candidature" pour obtenir plus de détails sur cet étudiant, afficher sa lettre de motivation ou encore consulter son cv s'il en a ajouté un.



Voir le cv aura pour action d'ouvrir votre lecteur de pdf par defaut avec le lien vers le pdf de l'étudiant (s'il existe)

Ensuite il pourra accepter , refuser ou attendre (= retour) la candidature de l'étudiant.

Si la candidature est acceptée l'étudiant recevra une notification lors de sa prochaine connexion :



*Aperçu de la notification lors de la prochaine connexion de l'étudiant*

**Vous avez un nouveau stage !**

## **Conclusion**

Ce projet nous à été très gratifiant, il nous a permis d'appréhender de nouveaux outils comme les bases de données ou les DAO. Il nous a permis de mieux gérer notre travail en groupe, même si l'utilisation de GitHub à été abandonné en cours de route car ne connaissant pas bien le logiciels, certaines données on été effacés. Craignant un autre incident, nous avons préféré ne plus travailler avec. En revanche il sera présent pour nos futurs projet.

Nous avons eu beaucoup de mal à se lancer dans le projet car nous n'avions aucunes connaissances en base de données et l'apprentissage de MySQL fut rude. Mais après cet aspect passé, nous avons pu être efficace dans notre travail.