

Rapport de projet

Application Java de gestion de stages

Groupe G6 : ANTOINET Dorian, DELPY Quentin et SCHINELLA Johan

Lien GitHub : <https://github.com/Horoneru/ProjetOffresStage>

Table des matières

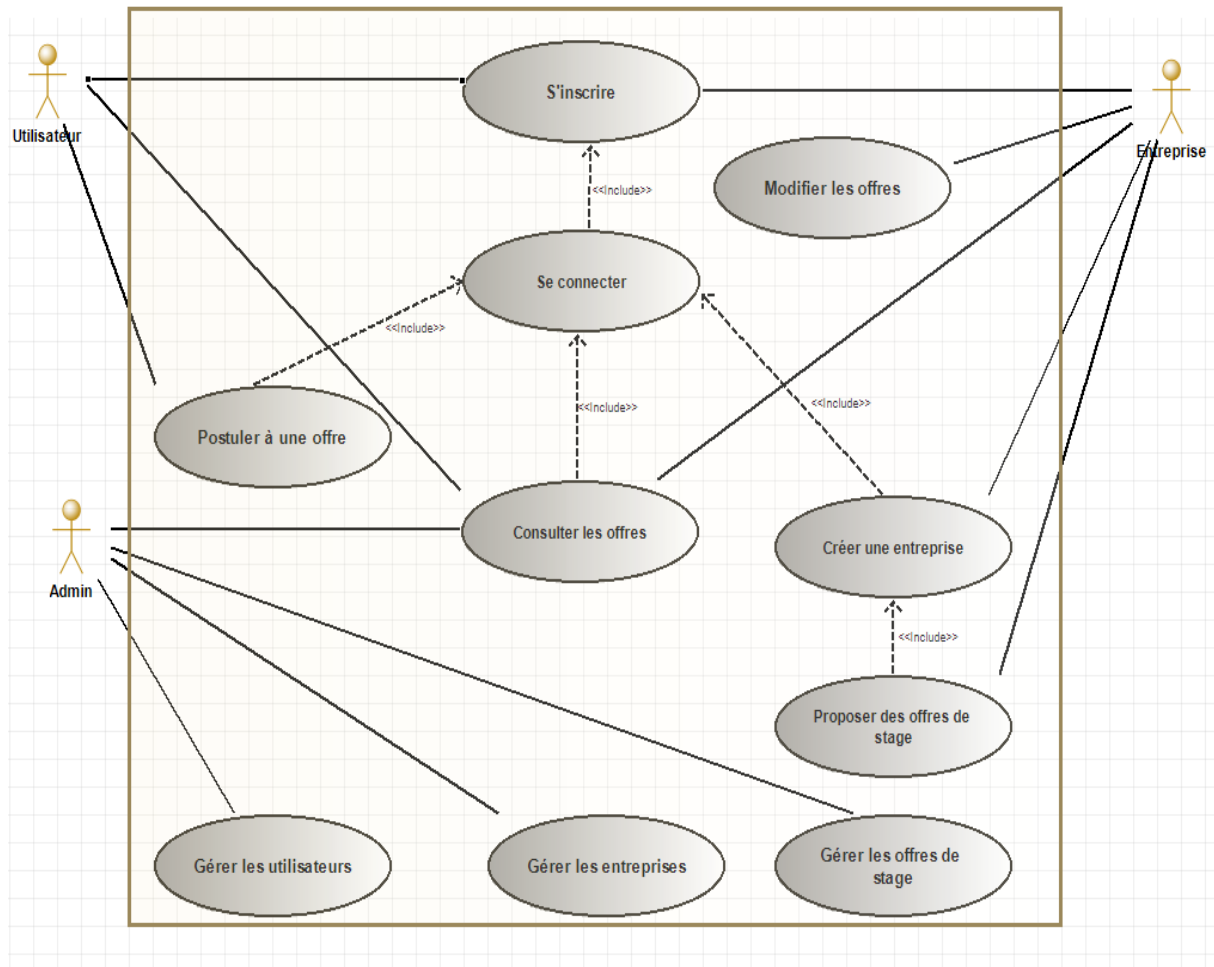
| | |
|--|-----------|
| <u>Rapport de projet.....</u> | <u>1</u> |
| <u>Introduction.....</u> | <u>2</u> |
| <u>I – Diagrammes UML.....</u> | <u>3</u> |
| <u>1)Diagramme de cas d’utilisation.....</u> | <u>3</u> |
| <u>2)Diagramme de séquence.....</u> | <u>4</u> |
| <u>3)Diagramme de classes.....</u> | <u>4</u> |
| <u>4)Schéma de la base de données.....</u> | <u>5</u> |
| <u>II – Interface.....</u> | <u>6</u> |
| <u>1)Connexion.....</u> | <u>6</u> |
| <u>2)Inscription.....</u> | <u>6</u> |
| <u>3)Page principale.....</u> | <u>7</u> |
| <u>4)À propos.....</u> | <u>9</u> |
| <u>III – Base de données.....</u> | <u>10</u> |
| <u>1)Table « Utilisateur ».....</u> | <u>10</u> |
| <u>2)Table « Entreprise ».....</u> | <u>10</u> |
| <u>3)Table « OffreStage ».....</u> | <u>10</u> |
| <u>IV – DAO.....</u> | <u>11</u> |
| <u>1)Classe « Dao ».....</u> | <u>11</u> |
| <u>2)Relation entre les classes « Utilisateur » et « UtilisateurDao ».....</u> | <u>11</u> |
| <u>3)Relation entre les classes « Entreprise » et « EntrepriseDao ».....</u> | <u>12</u> |
| <u>4)Relation entre les classes « OffreStage » et « OffreStageDao ».....</u> | <u>12</u> |
| <u>Conclusion.....</u> | <u>12</u> |

Introduction

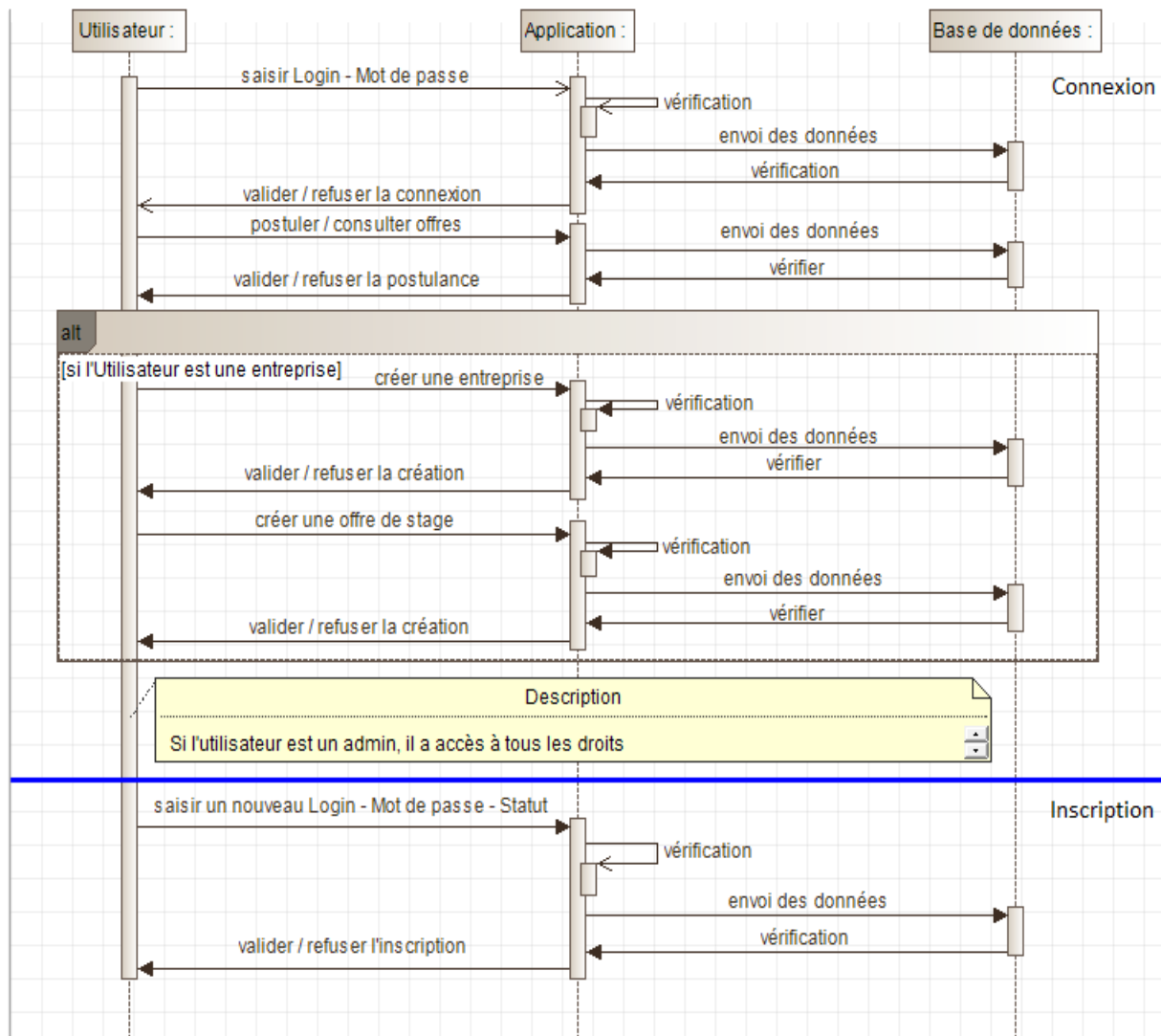
Le sujet consistait à créer une application permettant la gestion d'offres de stage entre différents utilisateurs (administrateur, étudiant ou entreprise). Selon le rôle, un utilisateur peut avoir plus ou moins de privilèges : un étudiant peut consulter les offres de stage et y postuler, une entreprise peut se déclarer et modifier ses offres de stage (en ajouter, supprimer, mettre à jour) et les consulter et l'administrateur peut consulter toutes les offres et gérer chacune des entités (entreprise, utilisateur, offre de stage).

I – Diagrammes UML

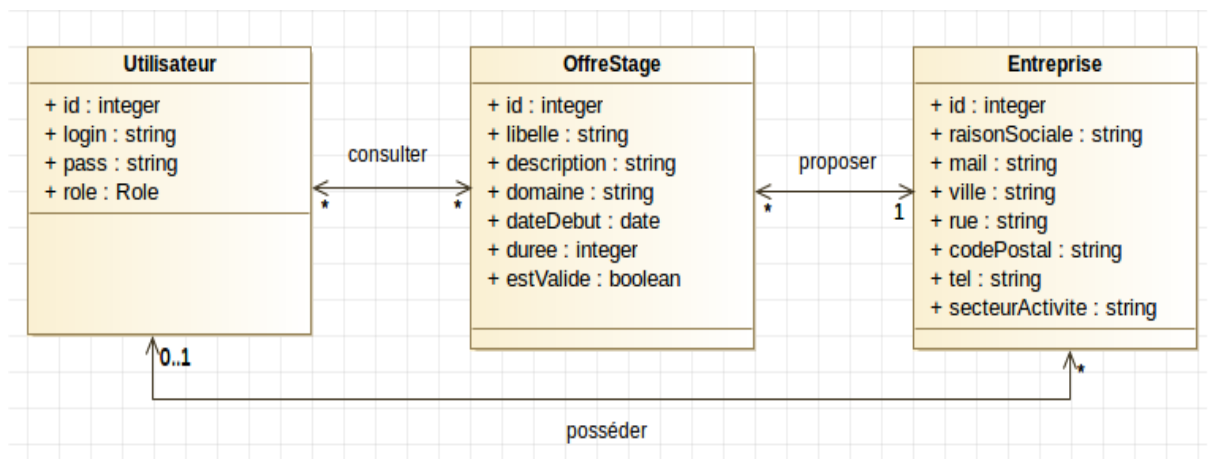
1) Diagramme de cas d'utilisation



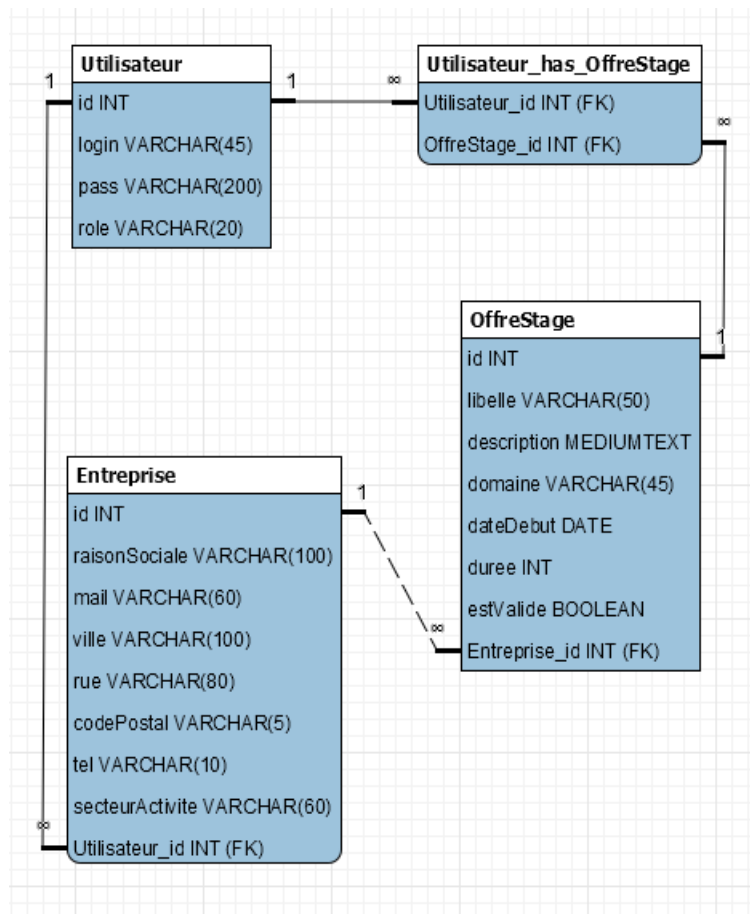
2) Diagramme de séquence



3) Diagramme de classes



4) Schéma de la base de données



II – Interface

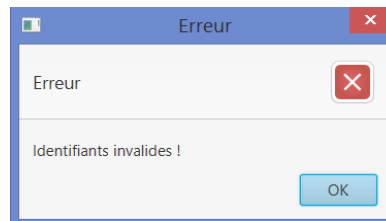
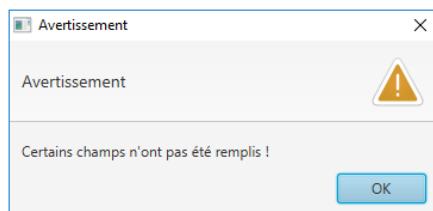
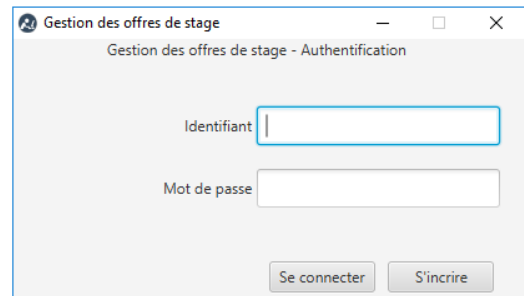
L'interface est ce qui relie l'utilisateur à l'application. Chaque évènement appelle une méthode spécifique en fonction de l'action effectuée, et ce depuis le contrôleur correspondant.

On compte 6 contrôleurs différents, dont un contrôleur « principal » lié au menu principal qui charge les différentes interfaces et gère le chiffrement des données sensibles (mots de passe) en SHA-1.

1) Connexion

Lors du lancement de l'application, l'utilisateur a accès à une interface de connexion à partir de laquelle il peut se connecter ou s'inscrire.

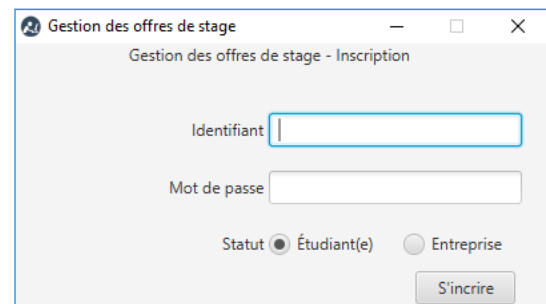
Pour se connecter, l'utilisateur doit saisir son identifiant et son mot de passe dans les champs correspondants. Si l'identifiant ainsi que le mot de passe saisis sont corrects, l'utilisateur sera redirigé vers la page principale du projet. Sinon, un message d'erreur apparaîtra et la connexion sera annulée.



2) Inscription

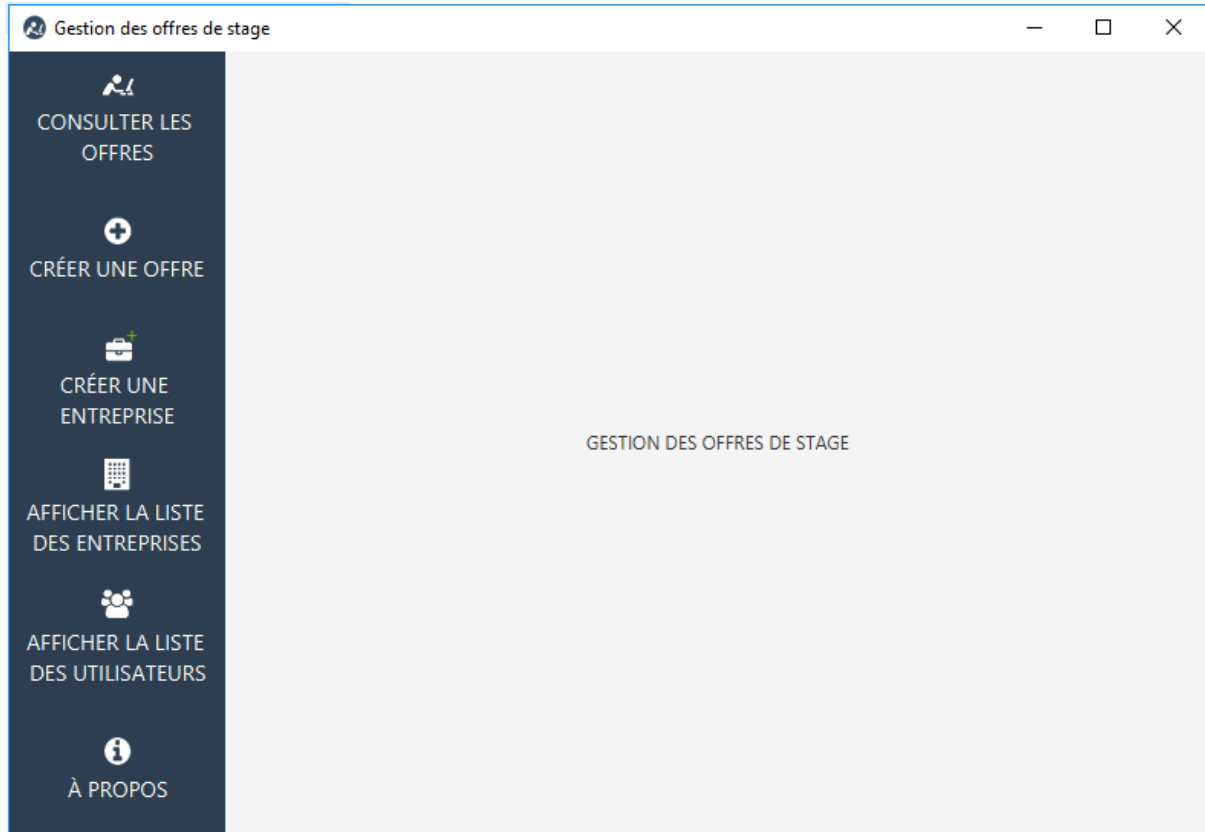
Lors de l'inscription, l'utilisateur saisit son identifiant et son mot de passe. Il choisit également son statut : « étudiant(e) » ou « entreprise ». Ensuite, le nouveau compte créé est inséré dans la base de données et l'utilisateur est redirigé vers l'interface de connexion.

En cas d'erreur (par exemple, un identifiant déjà pris ou un champ vide), un message d'erreur apparaît et indique la raison de l'échec de l'inscription.



C'est le contrôleur « *ControllerAuth* » qui gère à la fois l'inscription et la connexion. Il se base sur l'élément source de l'évènement pour déterminer le traitement à effectuer.

3) Page principale



Menu principal de l'application du point de vue d'un administrateur

Le menu principal change en fonction du type d'utilisateur (étudiant(e) ou entreprise). Un seul point commun : tout le monde peut consulter les offres de stage et accéder au menu « À propos ».

Les étudiants ont accès à un menu leur permettant de consulter les offres de stage proposées par les entreprises. Ils peuvent y voir le nom des entreprises, le lieu du stage, l'adresse e-mail du recruteur, l'intitulé et le domaine de l'offre, la durée (en jours) ainsi qu'un descriptif.

Il est possible de changer l'offre affichée sur la page à l'aide de boutons « suivant » et « précédent » en bas de la fenêtre. On peut également postuler à l'offre avec le bouton « postuler ».

Gestion des offres de stage

CONSULTER LES OFFRES

CRÉER UNE OFFRE

CRÉER UNE ENTREPRISE

AFFICHER LA LISTE DES ENTREPRISES

AFFICHER LA LISTE DES UTILISATEURS

À PROPOS

Entreprise entreprise

Ville là-bas

Mail mail

Domaine de l'offre info

Intitulé travail

Date de début du stage 02/05/2017

Durée (jours) 41

Descriptif bosse

Précédent Suivant Fermer Modifier Supprimer

Les entreprises ont, elles, accès à deux menus supplémentaires : ils leur servent à créer de nouvelles offres de stages et à insérer des entreprises dans la base de données.

Pour créer une offre de stage, il faut saisir le nom de l'entreprise, le domaine de l'offre, l'intitulé, la date de début du stage, la durée (en jours) et un descriptif.

Pour compléter la création de l'offre de stage, il suffit d'appuyer sur le bouton « Envoyer ».

C'est le contrôleur « *ControllerOffre* » qui gère les offres de stage en tant qu'entités. Pour ce qui est de leur consultation, il en est du contrôleur « *ControllerConsulter* ».

Pour créer une entreprise dans la base de données, il faut saisir le nom, l'adresse (rue et numéro), la ville et son code postal, l'adresse e-mail du recruteur de l'entreprise ainsi que son numéro de téléphone, et enfin le secteur d'activité de l'entreprise.

Une fois la saisie effectuée, l'utilisateur en mode entreprise pourra s'enregistrer dans la base de données et ainsi être accessible par les autres utilisateurs.

La gestion des entités « entreprise » est faite par le contrôleur « *ControllerEntreprise* ».

Gestion des offres de stage

CONSULTER LES OFFRES

CRÉER UNE OFFRE

CRÉER UNE ENTREPRISE

AFFICHER LA LISTE DES ENTREPRISES

AFFICHER LA LISTE DES UTILISATEURS

À PROPOS

Entreprise

Domaine de l'offre

Intitulé

Date de début du stage

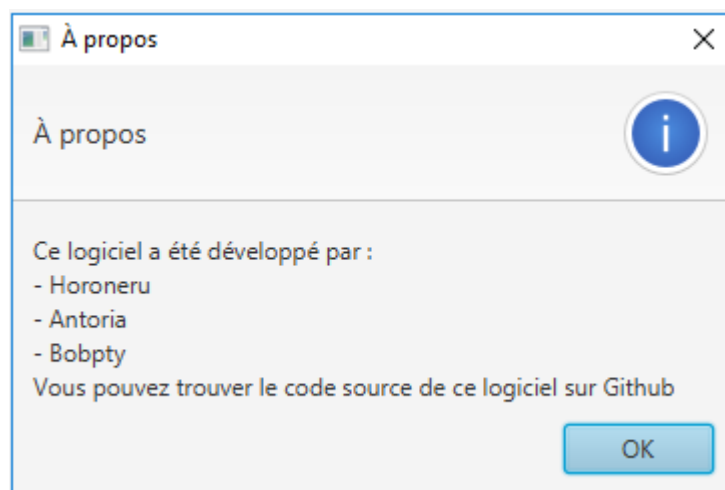
Durée (jours)

Descriptif

Envoyer Annuler

4) À propos

Ce menu permet d'ouvrir une fenêtre modale qui donne des informations générales sur le projet.



III – Base de données

Dans l'application, la classe « *Db* » permet la connexion à la base de données. Celle-ci est composée de 4 tables : « *Utilisateur* », « *Entreprise* », « *OffreStage* » et une table de jointure « *Utilisateur_has_OffreStage* », qui représente la liaison entre une entité Utilisateur et les offres de stage auxquelles il a postulé.

1) Table « Utilisateur »

Elle définit les utilisateurs qui se connectent pour accéder aux offres de stage. Un utilisateur est caractérisé par un numéro d'identifiant unique, son identifiant de connexion et son mot de passe crypté. Est également stocké le rôle de l'utilisateur. Il s'agit d'une énumération qui détermine les privilèges dont il dispose.

| Utilisateur |
|--|
| + id : integer + login : string + pass : string + role : Role |
| |

2) Table « Entreprise »

Il s'agit des entreprises ajoutées via l'application par des utilisateurs « entreprise ». Une entreprise est caractérisée par un numéro d'identifiant unique, une raison sociale, une adresse e-mail, une rue, une ville, un code postal, un numéro de téléphone et un secteur d'activité.

Il s'agit d'une entité qui possède des droits similaires à un utilisateur. Elle peut en plus proposer de nouvelles offres de stage, modifier les siennes et créer une nouvelle entreprise.

| Entreprise |
|--|
| + id : integer + raisonSociale : string + mail : string + ville : string + rue : string + codePostal : string + tel : string + secteurActivite : string |
| |

3) Table « OffreStage »

La table « *OffreStage* » contient les offres de stage émises par les entreprises. Une offre de stage est caractérisée par un numéro d'identifiant unique, un libellé, une description du stage, le domaine du stage, la date de début, la durée et enfin un booléen qui indique si l'offre de stage est encore valide ou non.

| OffreStage |
|--|
| + id : integer + libelle : string + description : string + domaine : string + dateDebut : date + duree : integer + estValide : boolean |
| |

IV – DAO

Le patron de conception DAO (« Data Access Object », « objet d'accès aux données ») a été utilisé pour les requêtes SQL, ce qui permet de faciliter leur appel ainsi que leur utilisation, tout en séparant des responsabilités qui ajoutent une couche d'abstraction au niveau de la persistance des entités. Le DAO est notamment utilisé dans les architectures logicielles objet.

1) Classe « Dao »

La classe abstraite « *Dao* » implémente les principales fonctions en lien avec la base de données (« CRUD » : Create, Read, Update, Delete).

On y retrouve les principales fonctions de recherche, d'ajout, de mise à jour et de suppression d'objet depuis un paramètre tel que l'objet ou son identifiant :

Méthode de recherche : *public abstract T find (int id)*

Méthode de création : *public abstract T create (T objet)*

Méthode de mise à jour : *public abstract boolean update (T objet)*

Méthode de suppression : *public abstract boolean delete (T objet)*

2) Relation entre les classes « Utilisateur » et « UtilisateurDao »

La classe « *Utilisateur* » contient les attributs qui permettent de gérer une entité d'utilisateur. Il s'agit du numéro d'identifiant unique, de l'identifiant de connexion, du mot de passe, du rôle (administrateur, étudiant(e) ou entreprise), et, en fonction du rôle, la liste des offres de stages auxquelles l'utilisateur a postulé ou alors la liste des entreprises qu'il a créées.

On emploiera différents constructeurs pour l'entité en fonction de la situation (inscription, connexion...).

L'entité « *Utilisateur* » dispose de deux méthodes : une pour postuler à une offre de stage, et une autre pour créer une entreprise.

Postuler : *public void postuler (OffreStage o)*

Créer une entreprise : *public void creerEntreprise (Entreprise e)*

La classe « *UtilisateurDao* » permet d'exécuter les principales requêtes liées aux entités « *Utilisateur* ». Certaines de ces requêtes sont initialisées dans la classe mère « *Dao* ». Parmi ces requêtes, en plus de celles déjà implémentées, on retrouve :

Recherche : *public Utilisateur find (String login)*

Récupération de liste d'utilisateurs : *public ArrayList<Utilisateur> findAll ()*

Postuler à une offre : *public boolean postuler (Utilisateur u, OffreStage o)*

Récupération de liste d'offres : *public ArrayList<OffreStage> findAlloffresStage (Utilisateur u)*

3) Relation entre les classes « Entreprise » et « EntrepriseDao »

La classe « *Entreprise* » regroupe les attributs et constructeurs d'une entreprise : son numéro d'identifiant unique, sa raison sociale, son adresse e-mail, son adresse (rue, ville et code postal), son secteur d'activité, son créateur (référence à un utilisateur) et la liste des offres de stages proposées.

La classe « *EntrepriseDao* » permet la persistance des entreprises. On y retrouve les requêtes d'ajout, de suppression, de modification et de recherche liées aux entreprises. Deux autres méthodes sont aussi présentes : elles sont liées au fonctionnement général de l'application.

Liste des offres de stage d'une entreprise : `public ArrayList<OffreStage> findAlloffres (Entreprise e)`

Liste des entreprises d'un utilisateur : `public ArrayList<Entreprise> findAllByUtilisateur (Utilisateur u)`

4) Relation entre les classes « OffreStage » et « OffreStageDao »

Les attributs de la classe « *OffreStage* » sont les mêmes que ceux qui apparaissent dans le diagramme de classes et dans le schéma de la base de données. On y rajoute simplement l'entreprise associée à l'offre et ses différents constructeurs.

La classe « *OffreStageDao* » exécute les requêtes implémentées dans la classe mère et ne dispose pas de requêtes spécifiques.

Conclusion

Ce projet a été réalisé par Bobpty (Johan SCHINELLA), Horoneru (Quentin DELPY) et Antoria (Dorian ANTOINET). Le code source de l'application est disponible sur GitHub à l'adresse :

<https://github.com/Horoneru/ProjetOffresStage>