

AP-5		Projet
J2EE		6 décembre 2011

Projet

Algo-Prog 5 – J2EE

- Le projet sera rendu sous forme d'archive compressée (jar), au plus tard lors des dernières séances de TD AP5 (la date sera précisée ultérieurement)

A Objectifs

- Il s'agit de concevoir une application Web permettant la gestion des stages, dans un contexte organisationnel comparable à celui du DUT informatique de Caen.

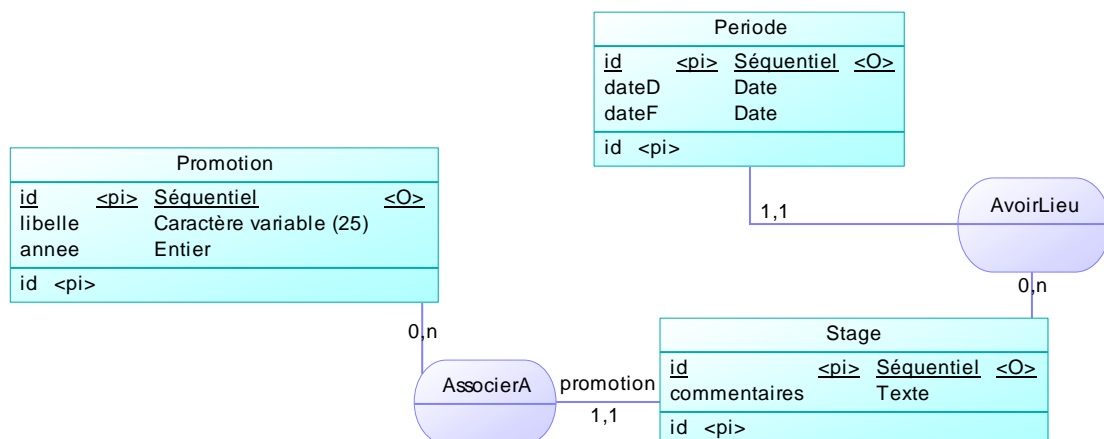
B Contraintes

- L'application sera développée en Java (j2ee),
- Elle respectera au mieux le design pattern MVC2, en séparant modèle (classes Métier), vues (interfaces web de saisie et d'affichage) et contrôleur (vérifiant l'intégrité des données).
- Elle utilisera la base de données Mysql fournie en annexe.
- L'utilisation de bibliothèques ou de Frameworks supplémentaires, vivement conseillée, est cependant laissée à la libre appréciation de l'étudiant.
- L'utilisation de scripts côté client (javascript et ajax) complétera les validations côté serveur.
- L'application sera constituée de six modules indépendants (cf plus loin)

C Règles de gestion :

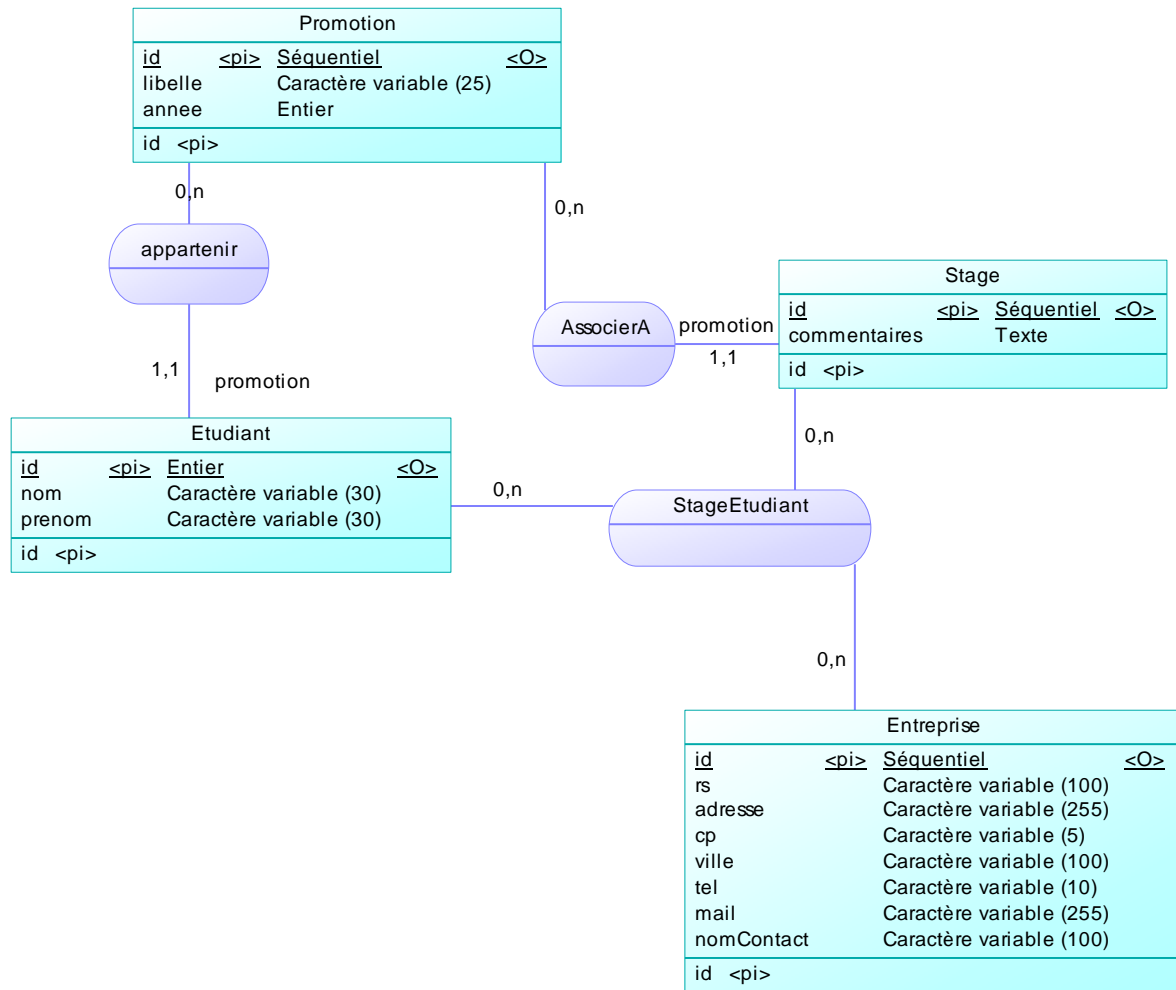
1 Stages

- un **stage** peut avoir lieu sur plusieurs périodes, définies par une date de début et une date de fin. Il est associé à une promotion.



2 Réalisation de stage

- Les étudiants effectuent leur stage dans une entreprise. On veillera à ce qu'un étudiant effectue un stage correspondant à la promotion à laquelle il appartient.
- Exemple de stage :**
 - {id :1, commentaires : "Fin de S4", promotion : {id :1, libelle : info2, promo : 2010-2011}}
 - Périodes [{id :1, dateD : 26/03/2012, DateF :01/06/2012}]

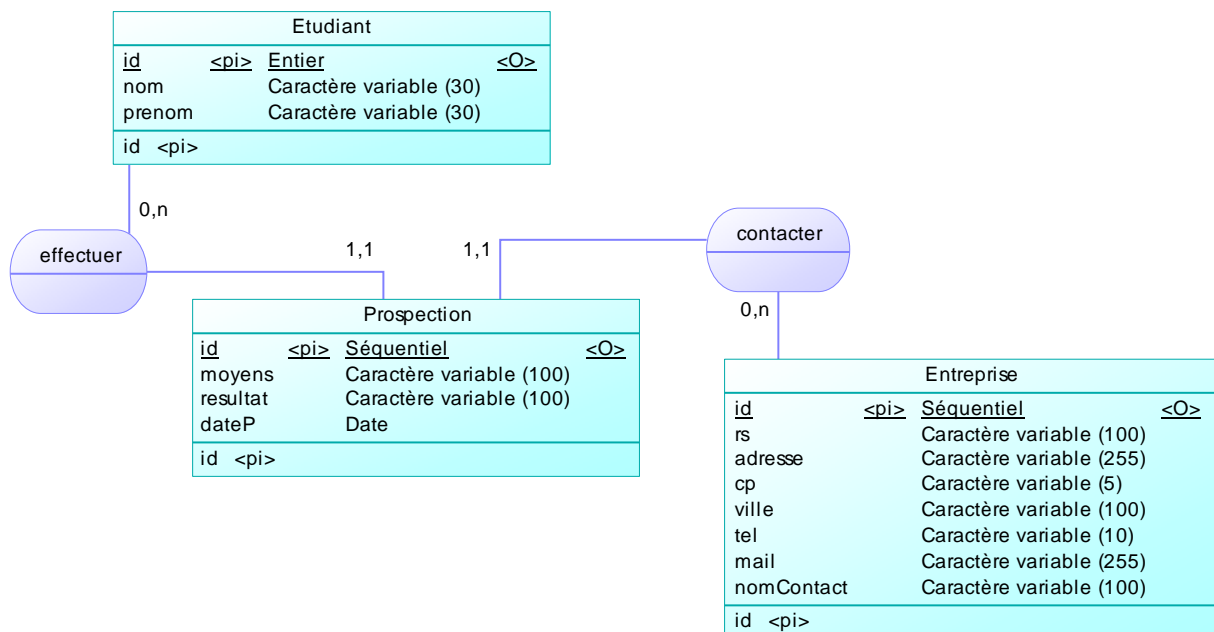


3 Prospection avant stage

De façon à mieux connaître les entreprises (et les entreprises susceptibles d'accueillir des stagiaires, on mémorise également les contacts pris par les étudiants pour trouver un stage.

Un étudiant peut donc prendre contact et prospector plusieurs entreprises pour y effectuer un stage. On notera la date du premier contact, les moyens de prospection (mail, tél, courrier, visite...), ainsi que les résultats de ces démarches (positif, négatif, en attente...).

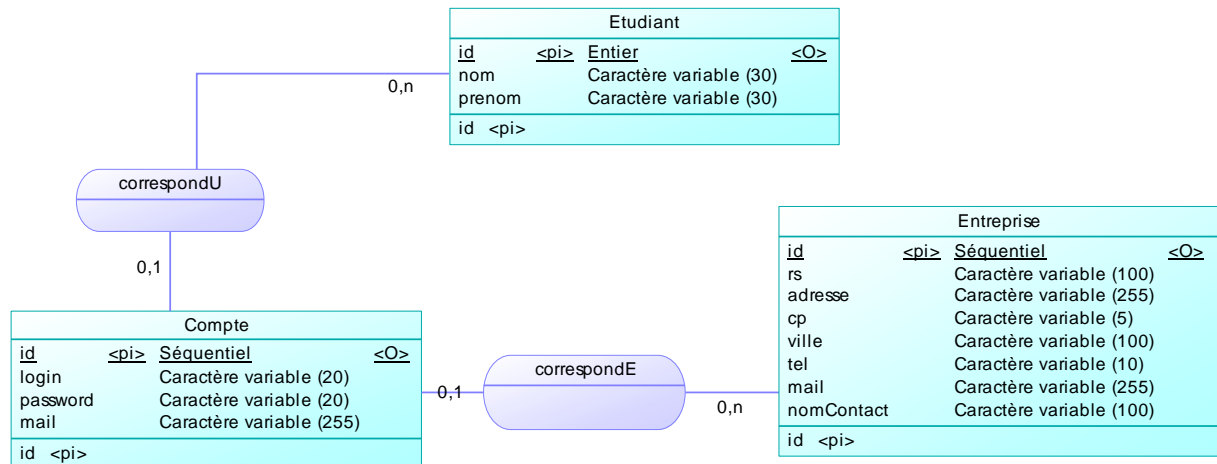
Pour simplifier la base de données, et son interrogation, on produit un modèle non normalisé où seront regroupées dans un même champ (moyens) plusieurs valeurs.



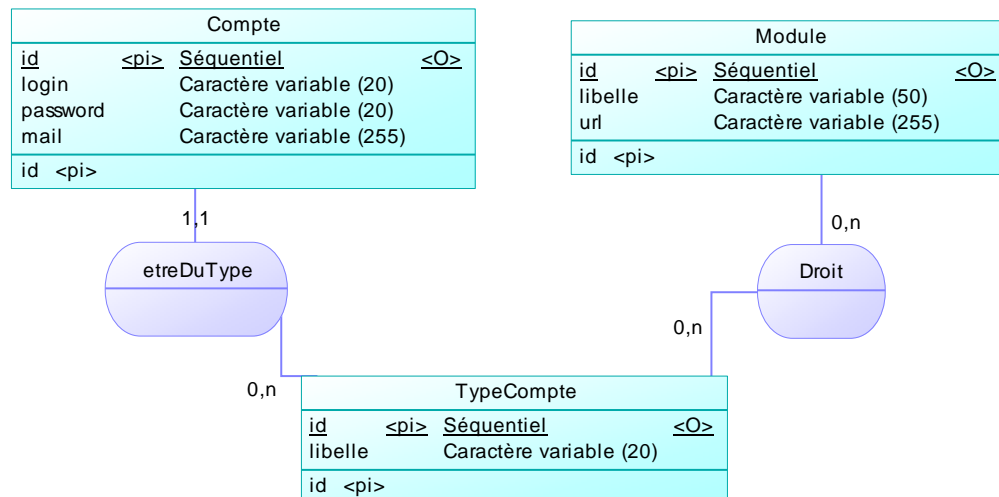
4 Utilisateurs, étudiants, entreprises et comptes

Toutes les personnes accédant à l'application doivent s'y connecter avec un compte utilisateur. Ce compte est muni d'un login (unique), et d'un password. L'adresse mail doit permettre de vérifier la validité du compte utilisateur.

Chaque compte est associé, soit à un étudiant, soit à une entreprise, soit à aucun des deux, lorsqu'il s'agit d'un compte enseignant ou administrateur.

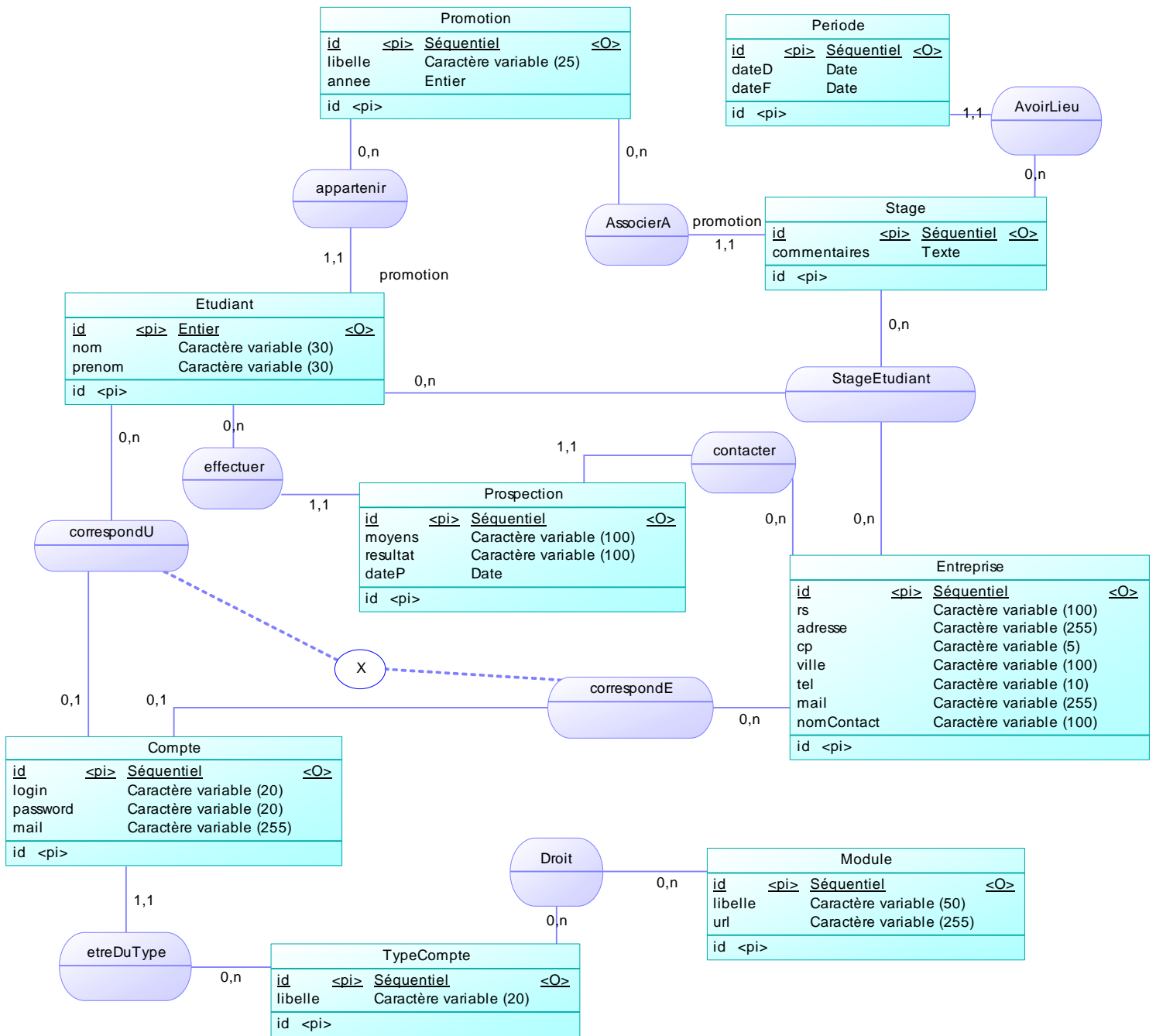


Chaque compte est d'un certain type, ce type définissant les droits d'accès aux différents modules du site.

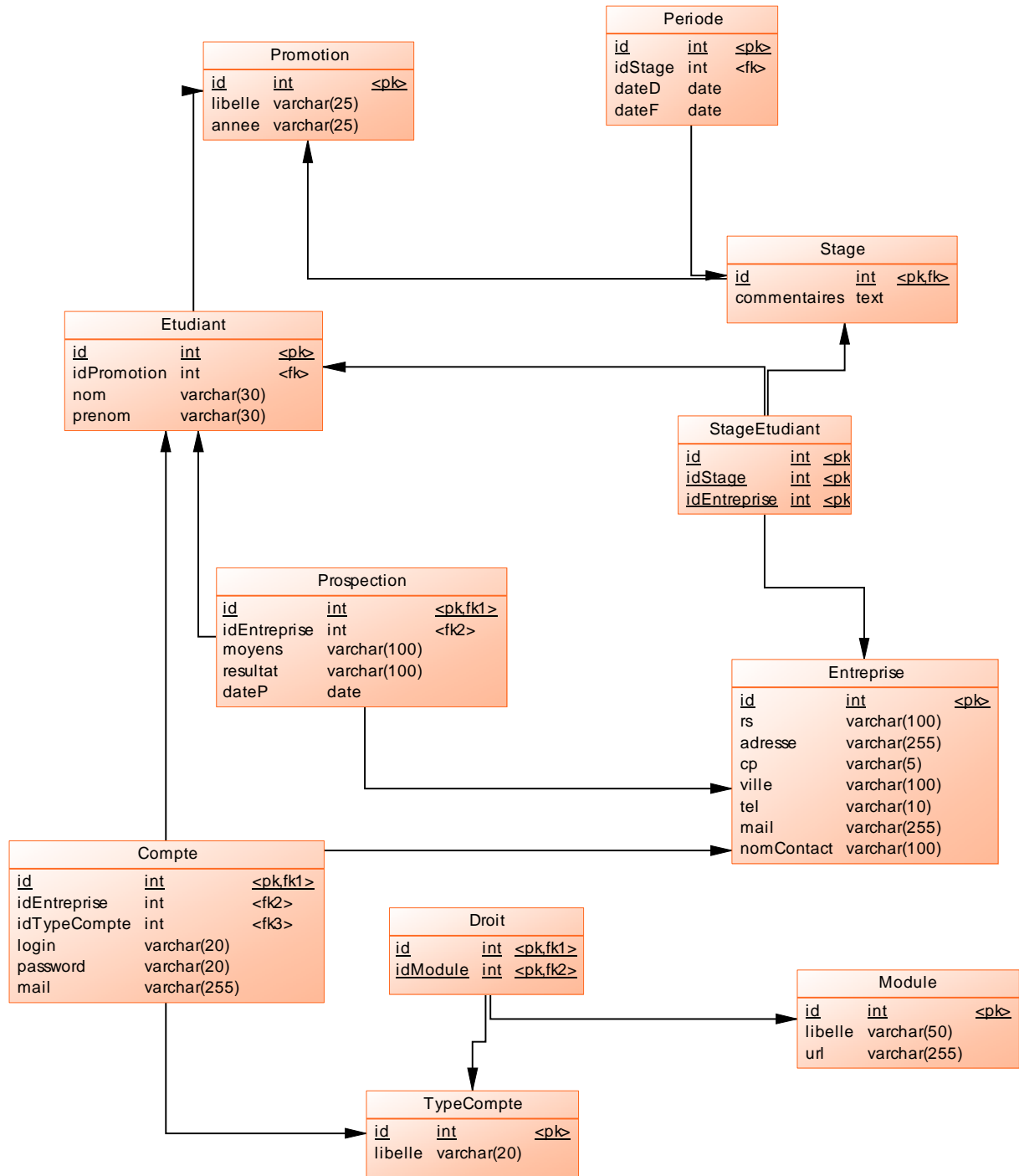


On distinguera les types de comptes suivants : Étudiant, Enseignant, Administrateur, Entreprise et invité.

Le modèle conceptuel de données obtenu est le suivant :



On obtient le modèle physique suivant :



AP-5		Projet
J2EE		6 décembre 2011

D Descriptif de la création d'un stage :

La création d'un stage devra suivre un scénario comparable à celui-ci :

L'application ne laissera le choix que de solutions valides, plutôt que d'afficher des messages d'erreur face à un choix impossible.

Cas d'utilisation : Création d'un stage

Acteur principal : Administrateur ou Responsable stages

Précondition : Il existe des promotions. L'utilisateur est logué et dispose de droits sur le module.

Scénario nominal :

1. L'utilisateur sélectionne une promotion
2. Il renseigne au besoin les commentaires sur le stage
3. Il saisit une première période de stage (date de début e date de fin)
4. [Le système affiche une zone permettant de saisir la période suivante](#)
5. L'utilisateur valide
6. [Le système sauvegarde le stage et le\(s\) période\(s\) associée\(s\)](#)

Scénarii alternatifs :

En 3 :

- L'utilisateur peut valider

En 5 :

- L'utilisateur peut saisir une nouvelle période, puis retour en 4

L'utilisateur peut abandonner à tout moment la saisie.

E Saisie du stage d'un étudiant

Cas d'utilisation : Saisie d'un stage

Acteur principal : Responsable stages

Précondition : Il existe des stages et des étudiants. L'utilisateur est logué et dispose de droits sur le module.

Scénario nominal :

1. L'utilisateur choisit le stage dans la liste. La liste devra être organisée par promotions, puis par dates de stages (en mettant en premier les stages récents).
2. [Le système affiche la liste des étudiants concernés](#)
3. L'utilisateur sélectionne un étudiant
4. [Le système affiche une zone de recherche unique de l'entreprise et un bouton Créer Entreprise](#)
5. L'utilisateur renseigne la zone de recherche (la recherche doit pouvoir se faire sur la raison sociale, l'adresse, le téléphone...)
6. [Le système met à jour la liste des entreprises trouvées, il affiche un bouton choisir](#)
7. L'utilisateur choisit une entreprise
8. [Il valide la saisie du stage pour l'étudiant](#)

Scénarii alternatifs :

En 4 et 5 :

- Si l'utilisateur crée une nouvelle entreprise, l'entreprise est sélectionnée puis passage en 8

En 6 :

- On veillera particulièrement à l'efficacité de la recherche d'entreprises sur plusieurs champs, pour éviter de saisir une entreprise déjà présente dans la base.

AP-5		Projet
J2EE		6 décembre 2011

L'utilisateur peut abandonner à tout moment la saisie.

F Gestion des modules et des droits

Un module est un point d'entrée dans l'application, il n'est accessible que par les types d'utilisateurs autorisés. Chaque utilisateur n'appartient qu'à un seul type.

Un module dispose d'une url, et correspond à une ressource web (jsp ou servlet).

Le type de l'utilisateur sera mémorisé dans sa session, et l'affichage d'une même page sera différent, en fonction des droits dont dispose l'utilisateur (grâce à son type) sur les modules existants.

On veillera également à protéger l'accès direct aux modules via leur url.

G Acteurs et fonctionnalités :

Rubrique		Fonctionnalités
DS :Données de structure		
<ul style="list-style-type: none"> Promotions Etudiants Types de compte Entreprises 	Administrateur	<ul style="list-style-type: none"> Ajout/modification/suppression Ajout/modification/suppression Ajout/modification/suppression Ajout/modification/suppression
Gestion des droits		
<ul style="list-style-type: none"> Comptes Modules Droits des types de compte sur les modules 	Administrateur	<ul style="list-style-type: none"> Ajout/modification/suppression/blocage Ajout/modification/suppression Ajout /suppression
Saisie des stages/périodes		
<ul style="list-style-type: none"> Stage période 	Responsable	<ul style="list-style-type: none"> Création/modification/suppression Ajout/modification/suppression de périodes dans le stage
Saisie des stages étudiant		
<ul style="list-style-type: none"> stage étudiant Entreprise 	Responsable Etudiant	<ul style="list-style-type: none"> Saisie stage/recherche d'entreprise Ajout d'entreprise
Saisie prospection		
<ul style="list-style-type: none"> prospection Entreprise 	Etudiant	<ul style="list-style-type: none"> Saisie stage/recherche d'entreprise Ajout d'entreprise
Affichages/recherches		
	Etudiant Entreprise Invité	<ul style="list-style-type: none"> Recherche d'entreprise Liste des entreprises Liste des stages avec période Liste des étudiants en stage (par promotion)

AP-5		Projet
J2EE		6 décembre 2011

H Travail à rendre

Le projet est à réaliser en binômes (**obligatoire**), il sera rendu sous la forme d'une archive WAR nommée **Prenom1Nom1-Prenom2Nom2.zip**, où prenom1Nom1 et prenom2Nom2 représentent les prénoms et noms des 2 étudiants participant.

L'archive devra comprendre :

1. L'application web
2. une documentation technique composée :
 - a. de la javadoc complète des classes, objets, variables, méthodes et fichiers créés
 - b. d'un plan détaillé du site précisant les interfaces, les fonctionnalités par interface, et la possible navigation entre elles
 - c. d'un bref manuel permettant l'installation de l'application
3. une documentation utilisateur au format html ou pdf permettant la prise en main de l'outil pour les non-initiés.
4. Un aperçu de la répartition des tâches entre les membres du binôme, basé sur le découpage en modules et fonctionnalités (Tableau avec couleurs).

I Critères de notation

▪ Le code :

Il devra être maintenable et évolutif, facilement repreneable par une autre équipe de développement.

- Nommage correct (classes, variables, fichiers, membres...) respectant des normes
- respect du principe de délégation
- respect des principes de la POO
- Séparation des couches (MVC2)
- Lisibilité et documentation du code

▪ Les interfaces :

On veillera particulièrement à leur ergonomie :

- à la simplicité de prise en main (sans documentation),
- au respect des normes de présentation des interfaces web,
- à l'accessibilité des fonctionnalités,
- à la normalisation W3C (+ intégration de HTML 5).

▪ L'application :

- Elle devra implémenter de façon complète et accessible les fonctionnalités des 4 modules décrits, en respectant les contraintes données.