

Dossier d'architecture

Client	Université de Lille1	Projet	Contrats étudiants
Créé le	06/12/2016	MAJ	16/01/2016
Auteur	DigiTeam	Version	2.0
Destinataire	Mireille Clerbout		

Membres :

- Sofiane Ouarab
- Cédric Duhautois
- Willy Pezant
- Alexandre Lefebvre
- Clément Dilly

Plan :

I) Architecture du projet

a. Contexte du projet	3
b. Plateforme technique	
Architecture logicielle	4
Architecture physique	4
c. Persistence	4
d. Sécurité	5
e. Fiabilité / Disponibilité	5
f. Performance	

II) Rappel des cas d'utilisation **6**

III) Vue logique

Architecture logique	9
Architecture des packages	10
Modèle de données	11

I. Architecture du projet

a. Contexte du projet

L'université propose aux étudiants des jobs, ponctuels ou réguliers, sous forme de vacations ou de contrats étudiants. Ces offres peuvent émaner de différents services de l'université, chacun de ces services gérant ses offres et ses contrats de manière indépendante. Ainsi, les offres ne sont pas regroupées à un même endroit (différents sites web, portes de certains bâtiments, ...), rendant difficile leur diffusion auprès des étudiants.

La réglementation impose également par ailleurs aux étudiants de ne pas dépasser un certain nombre d'heures de vacations. Les différents services ne disposant pas d'une vision globale des contrats signés, il se peut qu'un étudiant soit engagé pour un travail pour lequel il ne pourra finalement pas être payé. L'université souhaite promouvoir les contrats étudiants et désire donc mettre en place un outil facilitant la diffusion des offres, la constitution d'un stock de CV des étudiants intéressés par un job et la gestion des contrats.

L'objet du projet est donc mettre en place une plate-forme web pour répondre à ces problématiques. Elle devra permettre de centraliser les offres et les profils des étudiants/candidats et, également, faciliter l'attribution des contrats afin qu'ils puissent s'inscrire dans un cadre légal strict.

L'application, destinée à l'ensemble du personnel de l'université, devra pouvoir être accessible de partout et être consultable à partir d'un smartphone ou d'une tablette.

b. Plateforme technique

1. Architecture logicielle

L'application sera une application Java de type Spring développée avec le framework **Spring Boot** (v1.4.2) .

Elle sera packagée et distribuée via le moteur de production Maven.

L'application suivra le pattern Modèle Vue Contrôleur (**MVC**) avec une architecture en **5 couches** : Couche accès données (DAO), Couche services, Couche model (objets métiers), la Couche contrôleur et la enfin couche Présentation (vue).

Concernant le développement, nous utiliserons le moteur de template Thymeleaf afin de générer les vues au format HTML5 ainsi que le framework Bootstrap, pour la partie CSS, JavaScript et HTML5.

Pour la partie gestion des données nous utiliserons une base de donnée relationnelle MySql (voir schéma d'architecture logique pour plus de détails).



2. Architecture physique

L'application sera hébergée sur une machine virtuelle Debian grâce à la plateforme "OpenStack" proposée par l'université.

Nous avons également installé:

- Apache2 pour le serveur HTTP
- Jenkins pour l'intégration continue
- Le framework Spring pour les développements J2EE
- MySql pour la Base de données

c. Persistence

La persistance des données se fera en JPA via Spring-Data qui est une technologie développée par Spring permettant d'implémenter facilement un repository JPA. Spring Data repose sur le design pattern Repository du DDD (Domain Driven Design). Les données seront stockées dans la base de données MySQL.

Pour les C.V collectés des étudiants ils seront stockés à l'emplacement :
" /home/debian/digiteam/CVs/upload-dir " de notre VM.

Les types de mission seront elles stockées en base.

d. Sécurité

La sécurité de la plateforme se fera grâce au système d'authentification LDAP et aux rôles que nous associerons aux utilisateurs conformément aux différents acteurs que nous avons identifiés dans le système, les rôles seront les suivants :

- Super Admin
- Admin
- Référent
- Personnel de lille 1
- Étudiant

Ces rôles permettent d'accéder aux parties du site qui correspondent aux actions que les utilisateurs pourront effectuer.

Les autorisations et les rôles seront gérés grâce au framework Spring Security via une connexion LDAP.

e. Fiabilité / Disponibilité

Le site doit être accessible et fonctionnel à tout moment depuis un ordinateur, un smartphone ou une tablette.

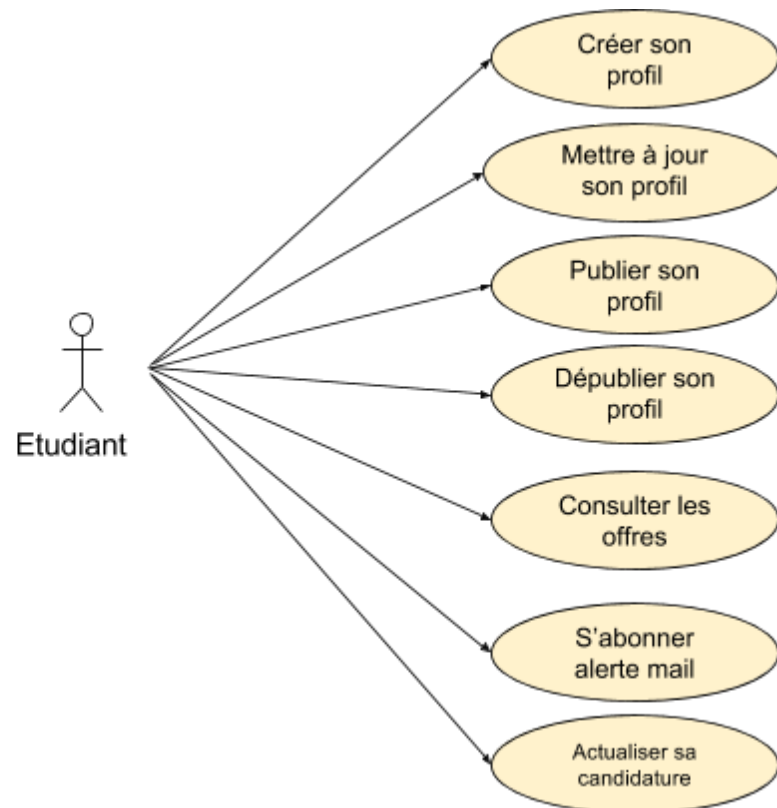
La disponibilité du site sera de 22/7 : 22 heures par jour / 7 jours par semaine
Avec 2 heures réservées pour la maintenance.

f. Performance

Le site devra être rapide d'accès depuis n'importe quel navigateur.

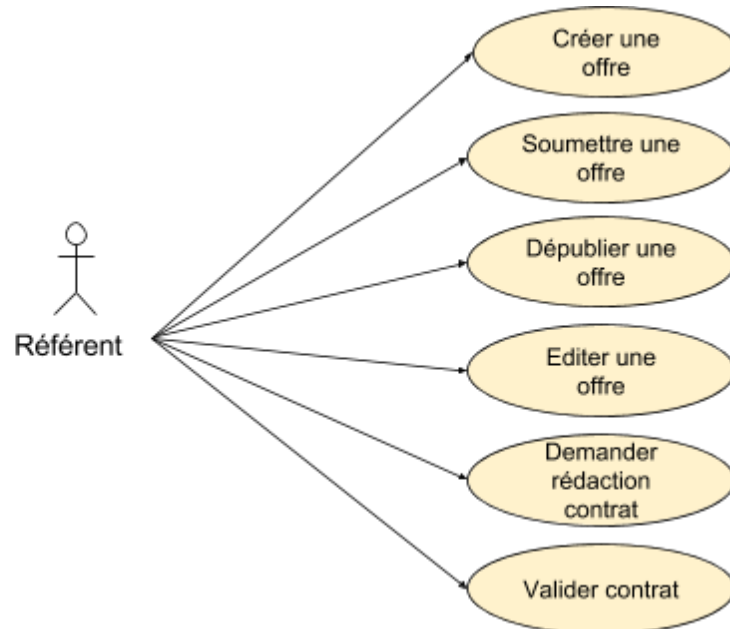
II. Rappel des cas d'utilisation

Cas d'utilisation de l'étudiant



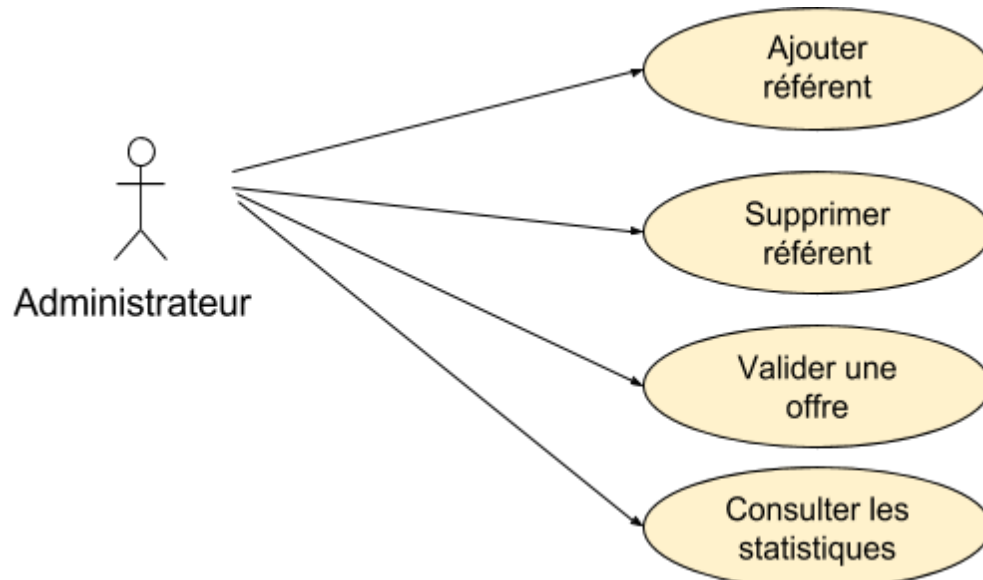
L'étudiant peut créer son profil et publier sa candidature sur l'application. Il peut aussi consulter les différentes offres qui ont été publiées par les administrateurs et ainsi obtenir les coordonnées pour postuler.

Cas d'utilisation du référent



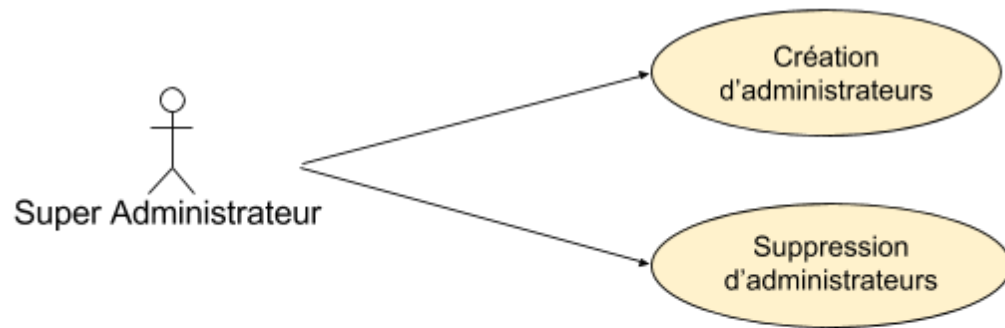
L'acteur référent est celui qui soumet les offres à publier aux administrateurs, il peut éditer et publier une offre en cours. Il est aussi celui qui demande la rédaction des contrats étudiants.

Cas d'utilisation de l'administrateur



L'administrateur effectue la gestion des référents (ajout, suppression) et valider les offres soumises par les référents.

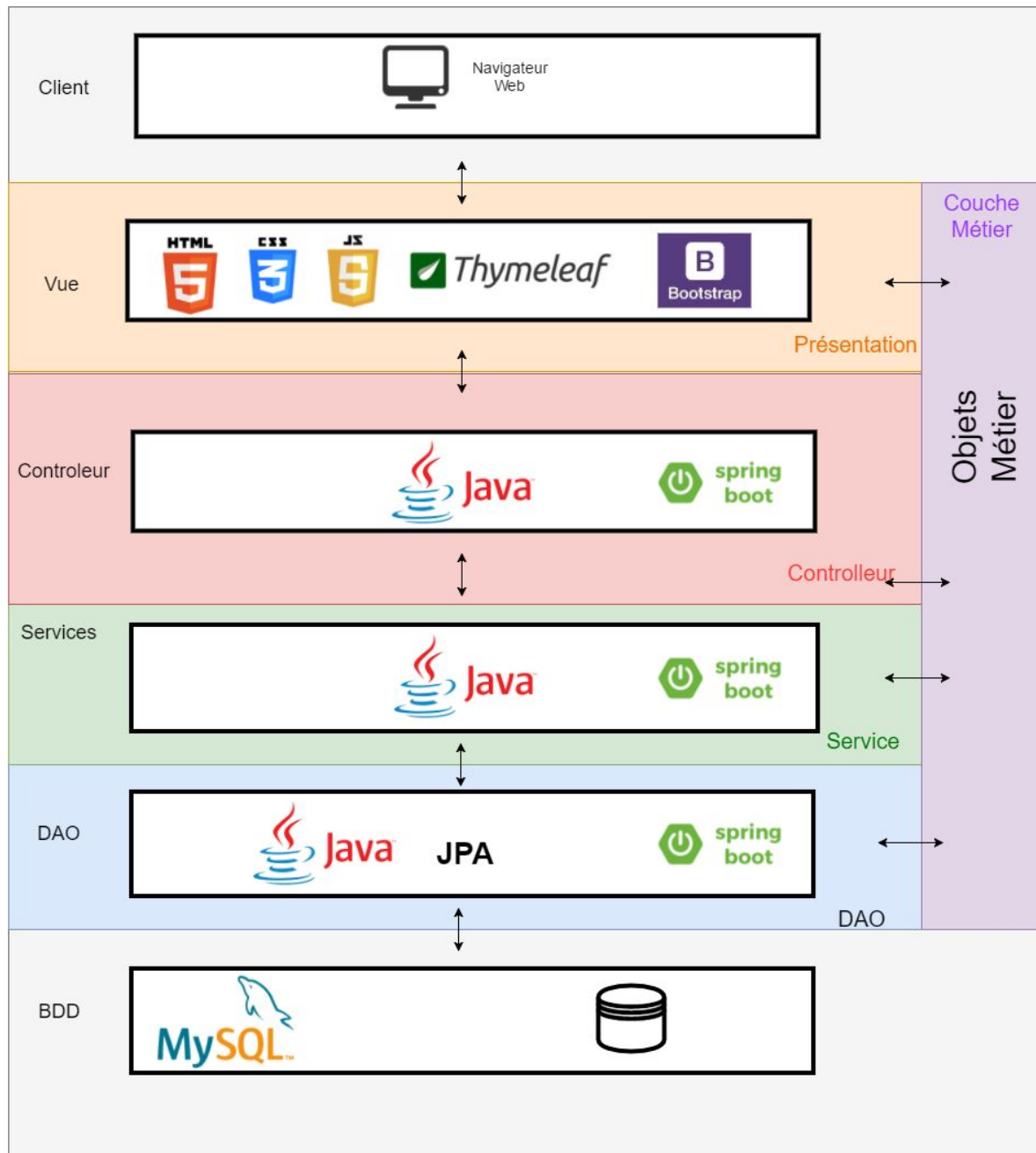
Cas d'utilisation du super administrateur



Le super administrateur gère les administrateurs (ajout, suppression)

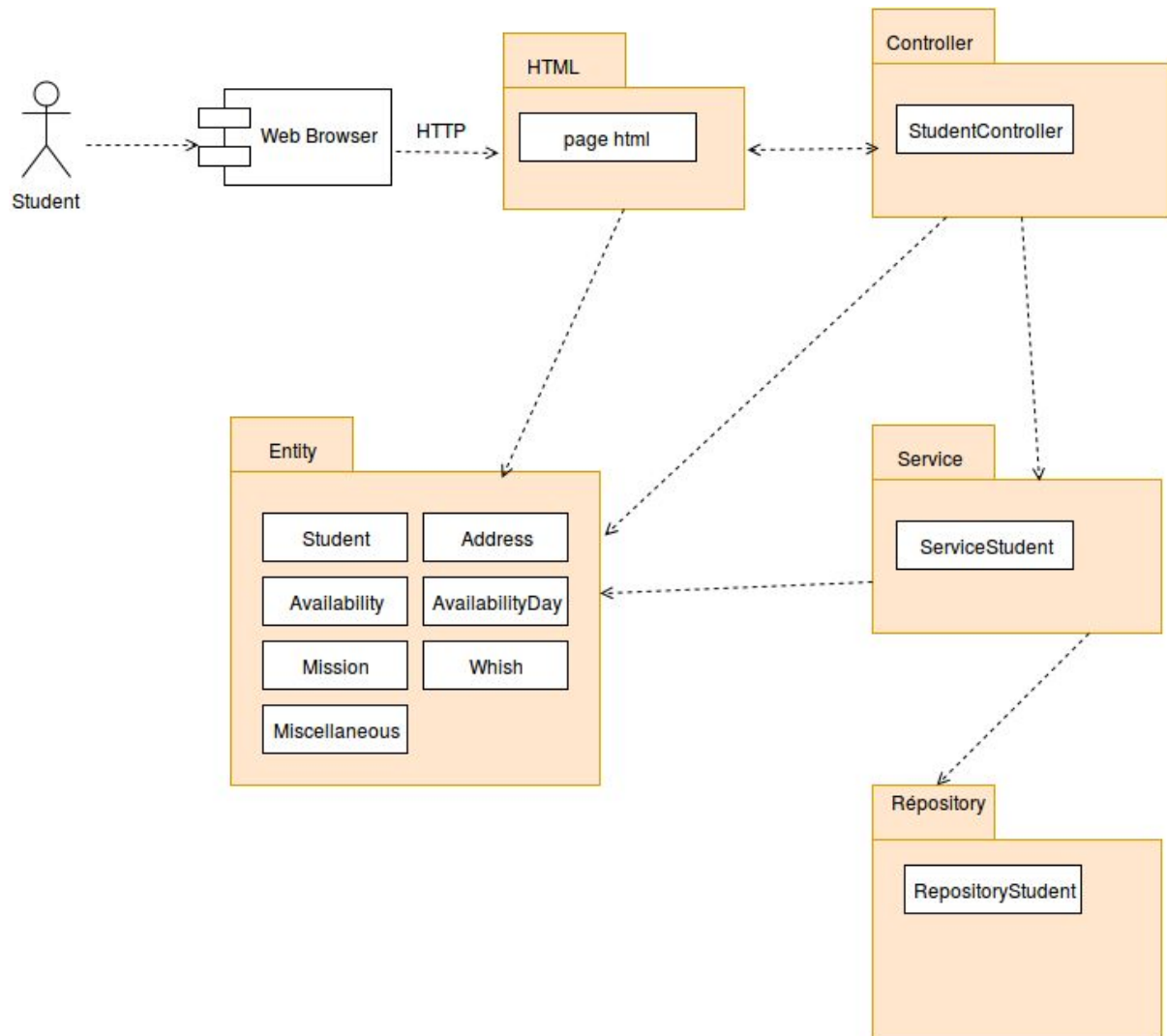
III. Vue Logique

a. Architecture logique



b. Architecture des packages

Diagramme de paquetage des classes



c. Modèle de données

Modèle de données concernant l'utilisateur : Etudiant

