

INF211 Fil Rouge

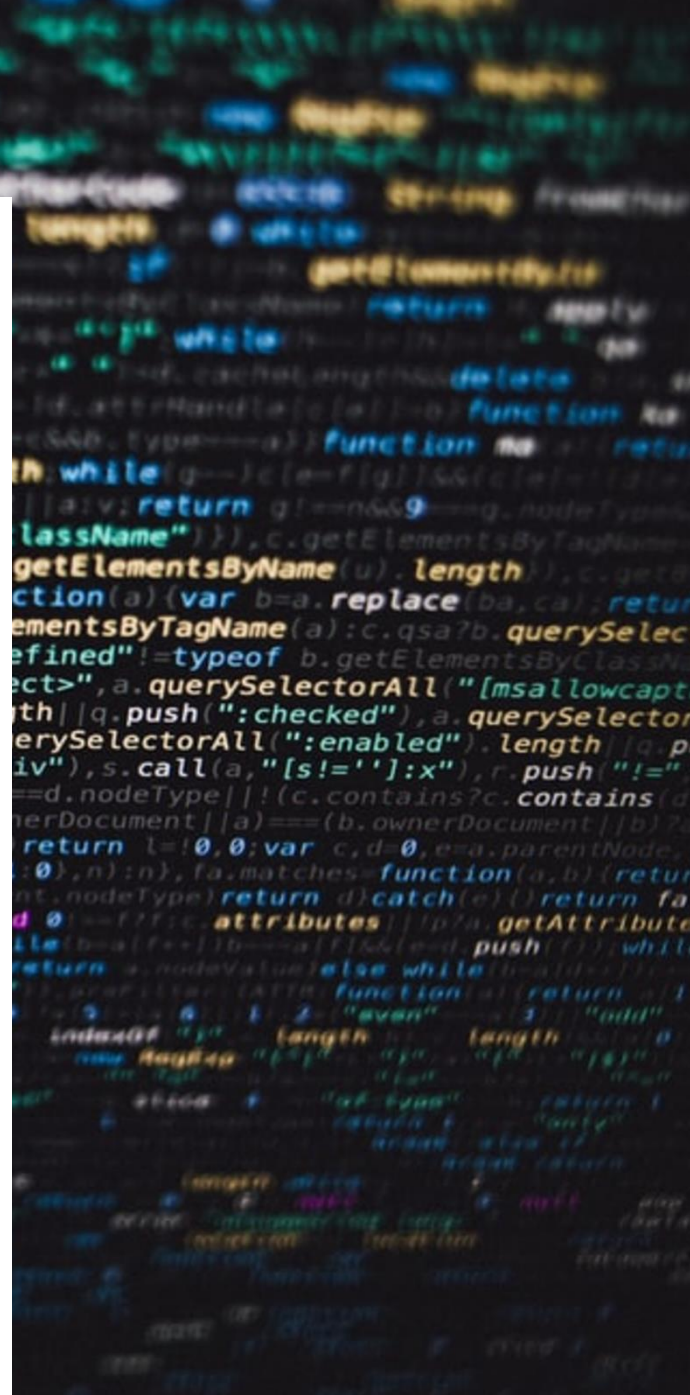
IMT Atlantique



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

14/03/2021

Créé par :
LE DUC Elouan
LE GRUIEC Clément



Architecture technique mis en place

Le projet fil rouge est basé sur une architecture dite « 3-tiers ». Il s'agit d'une application basée sur 3 couches différentes. La couche « accès aux données », la couche « métier » et la couche « présentation ». Chacune des couches représente une partie de notre logiciel. Ces 3 tiers sont représentés sur le schéma ci-dessous :

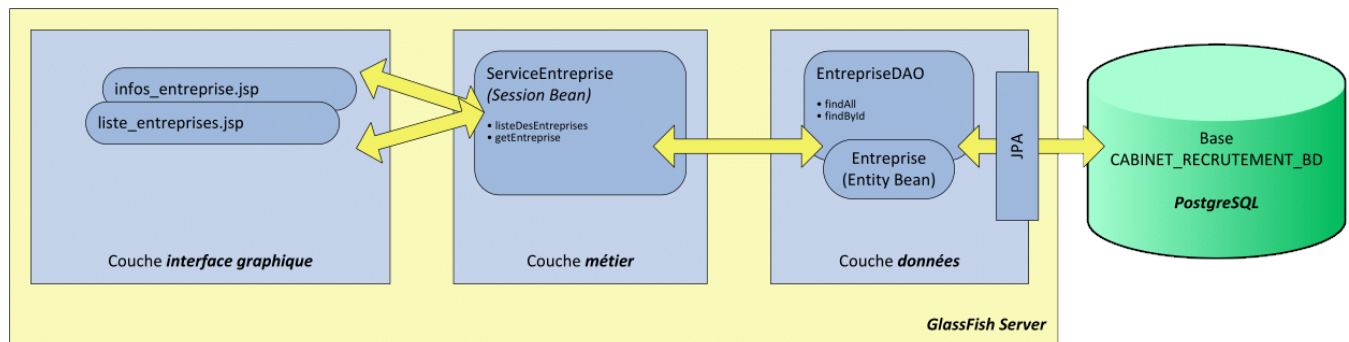


Figure 1 : Architecture simplifiée de l'application cabinet de recrutement

A. Le serveur SQL

Dans le cadre de notre projet nous utilisons un serveur de SGBDR nommé PostgreSQL sur lequel nous avons créé un ensemble de tables correspondant à notre cahier des charges. Pour automatiser cette opération nous avons écrit un script SQL, qui créer la structure et ajoute des données pour les tests. Le script peut également être utilisé pour remettre à zéro la base de données.

B. Couche données

La couche « données » gère le lien entre notre BDD (PostgreSQL ici) et notre application logicielle. Pour ce faire, nous avons eu besoin d'utiliser un outil d'ORM (**O**bject **R**elational **M**apping). Il est chargé de faire le lien entre nos classes Java et nos bases de données. Dans le cadre de ce projet, nous utilisons le framework Hibernate qui implémente JPA (**J**ava **P**ersistence **A**PI) pour réaliser ce travail d'interfacage. Ensuite, pour créer nos classes Java associées à notre BDD via JPA, nous avons utilisé les entités JPA. Cela représente les objets « Entreprise, Candidature, OffreEmploi ... ».java manipulés par l'ensemble de notre application. Puis pour manipuler ces objets, nous avons créé des fichiers DAO. Ce sont des patrons de conception (**D**ata **A**ccess **O**bject) dans lesquels nous avons défini l'ensemble des fonctions permettant d'utiliser nos entités JPA. Ces fonctions implémentées sont les méthodes dites CRUD (**C**reate, **R**ead, **U**ppdate, **D**eleate). Elles permettent donc de créer un objet, le persister dans notre BDD, de le récupérer depuis la table pour le modifier ou même de le supprimer. Ces fonctionnalités sont indispensables pour le bon fonctionnement de notre application.

C. Couche métier

La couche « métier » permet d'implémenter des fonctionnalités spécifiques dans des services. On y retrouve des méthodes réalisant des ensembles d'opérations à partir des DAOs. Cette couche n'est pas essentielle dans le bon déroulement de l'application mais c'est une bonne pratique. L'utilisation des services permet de factoriser le code et de renforcer la possibilité d'évolution du logiciel (pas obligé de réécrire les fonctions fondamentales). Il est donc possible d'avoir plusieurs type d'IHM rattachées à la même couche de données sans devoir tout réécrire et adapter, la couche métier se charge de cette unicité. Ces services sont également en coupure de la couche « données », cela permet de vérifier et de traiter la donnée avant d'interroger un DAO. Les services sont segmentés sous formes de thématique, ici cela correspond aux tables de la BDD.

D. Couche IHM

« IHM » pour Interface **H**omme **M**achine. Les interactions avec le logiciel se font au travers de pages WEB. Ces pages sont créées par GlassFish à partir de fichiers *.jsp*. Dans ces fichiers on retrouve à la fois du langage HTML et Java. Cela nous permet de pouvoir afficher et traiter des données de manière dynamique (similaire à du PHP). La page de base qui est constamment appelée s'appelle *template.jsp*, elle permet de poser les bases du site avec l'intégration des entêtes, menus, pied de pages etc... La page va également rajouter un « fragment » spécifique à l'action demandée par l'utilisateur, par exemple le formulaire d'ajout d'une entreprise. Les données saisies par l'utilisateur sont récupérées grâce à la méthode GET. Chaque attribut à une valeur associée. Le traitement de la donnée est ensuite délégué aux services de la couches « métier ». Des traitements spécifiques à l'action y sont effectués.

Organisation du travail

Au début du projet nous avançons en parallèle afin de pouvoir prendre en main le projet. Rapidement nous avons réparti le travail afin d'être plus efficace. Globalement Elouan s'est concentré sur la partie « tests » avec *controleDao* ainsi que le développement des « DAO » de la couche données et l'ajout de données de tests dans le script SQL. Clément quant à lui développait la partie « services » de la couche métier et la partie « JSP » de la couche IHM. Bien sûr nous nous sommes entraides sur les 2 parties, nous avons pu toucher à tout.



Nous avons choisi de travailler avec GitHub pour héberger notre code et ainsi pouvoir utiliser les outils agiles. Il était très facile de pouvoir fusionner nos avancements dans le logiciel. L'outil permettait également de détecter les incohérences entre nos codes et ainsi prévenir des problèmes dans l'application après la mise en commun des fonctionnalités développées par chacun.

Fonctionnalités du logiciel (rappel)

Fonctionnalité	Importance	Etat de fonctionnement
Référencer une nouvelle entreprise	Vitale	Réalisée
Lister les entreprises référencées	Vitale	Réalisée
Afficher les informations d'une entreprise	Vitale	Réalisée
Mettre à jour les informations d'une entreprise	Mineure	Réalisée
Supprimer une entreprise	Mineure	Non réalisée
Envoyer un message à un candidat pour une offre d'emploi donnée	Mineure	Non réalisée
Lister les messages reçus des candidats	Mineure	Réalisée
Lister les messages envoyés aux candidats	Mineure	Réalisée
Référencer une offre d'emploi	Vitale	Réalisée
Lister les offres d'emploi	Vitale	Réalisée
Afficher les informations d'une offre d'emploi	Vitale	Réalisée
Mettre à jour les informations d'une offre d'emploi	Mineure	Non réalisée
Supprimer une offre d'emploi	Mineure	Non réalisée
Référencer une nouvelle candidature	Vitale	Réalisée
Lister les candidatures référencées	Vitale	Réalisée
Lister les offres d'emploi qui correspondent à une candidature	Vitale	Réalisée
Mettre à jour les informations d'une candidature	Mineure	Réalisée
Supprimer une candidature	Mineure	Partiellement réalisée
Envoyer un message à une entreprise pour une offre d'emploi donnée	Mineure	Non réalisée
Lister les messages reçus des entreprises	Mineure	Réalisée
Lister les messages envoyés aux entreprises	Mineure	Réalisée
Se connecter au système d'information	Vitale	Réalisée
Se déconnecter du système d'information	Vitale	Réalisée
Exporter la liste des offres d'emploi au sein d'un flux RSS	Mineur	Non réalisée
Exporter la liste des candidatures au sein d'un flux RSS	Mineur	Non réalisée

Commentaire(s) :

Les fonctions vitales sont écrites et en service.

Les suppressions ne sont pas opérationnelles. Il y a un problème au moment de supprimer l'objet de la table, dans notre cas une candidature.

Message d'erreur :

Caused by: java.lang.IllegalArgumentException: Cannot merge an Entity that has been removed: eu.telecom_bretagne.cabinet_recrutement.data.model.MessageCandidature@6dfdc6c