



# SERVICE WEB DE GESTION DE CANDIDATURES DE STAGE

**Rédigé par** : Bill MBIAWA & Momo NANA

**Supervisé par** : M. Mario VALENCIA

**Année académique** : 2022-2023,





## Sommaire

I - Introduction générale.....	3
II - Description du cas d'utilisation .....	3
II.a. - Interaction Etudiant-Entreprise .....	3
II.b. - Contenu des différentes couches .....	5
III - Description de l'architecture à mettre en place.....	6
IV - Technologies utilisées.....	6
IV.a. - Springboot .....	6
IV.b. - Angular .....	7
IV.c. - Hibernate .....	<b>Erreur ! Signet non défini.</b>



## I - Introduction générale

Dans le cadre de notre projet du semestre 7 de la spécialité IA2R (Informatique Automatique Réseaux & Robotique), il nous a été demandé de créer un service web associé à un cas d'utilisation précis. En effet, nous avons fait un choix relatif à la situation que nous subissons tous en tant qu'étudiant en quatrième et cinquième année ici à Polytech Nancy pour ne pas citer les étudiants de par la France et le monde entier. Nous avons choisi de mettre sur pied un service web portant le nom de **MoBill\_Stage** permettant les interactions entre étudiants et les entreprises pour la recherche de stage. Ici il est donc question de donner la possibilité aux entreprises de publier leurs offres de stages sur la plateforme ; réciproquement, les étudiants auront aussi la possibilité de les consulter et de postuler aux différentes offres publiées. Aussi il n'est plus à préciser que l'ergonomie et l'esthétique des différentes pages est un facteur à ne pas négliger dans la conception de notre service web car, après les potentielles fonctionnalités, le garant de l'attractivité de notre plateforme.

## II - Description du cas d'utilisation

### II.a. - Interaction Etudiant-Entreprise

Comme nous l'avons dit dans l'introduction, notre service web a pour but de rendre plus aisées les recherches de stage (pour les étudiants) et les recherches de stagiaires (pour les entreprises). Le principe est tout simple, l'étudiant ou l'entreprise peut se créer un compte en s'inscrivant (en donnant les informations relatives à ce dernier ou cette dernière) sur la plateforme. Immédiatement après, l'entreprise a la possibilité de générer des offres de stage en fonction de leur besoin, mais peut aussi consulter le profil des différents étudiants inscrits sur la plateforme. L'étudiant quant à lui peut voir les différentes offres postées par les entreprises et postuler s'il est intéressé ; il peut aussi consulter directement les comptes des multiples entreprises présentes sur la plateforme.

Néanmoins, il existe plusieurs autres fonctionnalités auxiliaires :

- Toute personne ou entreprise inscrite a la possibilité de modifier ses informations personnelles,
- Toute entreprise peut modifier ou supprimer une offre après avoir recruté son stagiaire (ou pas),
- Tout étudiant peut retirer sa candidature d'une offre.
- Toute entreprise et tout étudiant peut supprimer son compte et ainsi, ses données des serveurs de l'application



GestionCandidatures x Connexion - CAS - Centr... x Spring Boot Upload/Do... x Angular 14 File upload... x Boîte de réception (2.80) x

localhost:4200

Série Limitée... Dashboard + S... Accueil | Mons... Direct TV | To... Accueil | S.C.T... Phrasal Verbs... Courrier - MBI... Regarde gratu... YouTube News Calculatrice Sc... Comparer prix... Consignes po...

# MoBill\_Stage.

Entreprises Etudiants Offres Mes Candidatures Mon profil

Vous êtes: ☐ Etudiant ☐ Entreprise

Id de connexion: (email ou pseudo de l'entreprise)

  

Mot de passe

  

Votre mot de passe ne sera jamais divulgué.

GestionCandidatures x Connexion - CAS - Centr... x Spring Boot Upload/Do... x Angular 14 File upload... x Boîte de réception (2.80) x

localhost:4200/creerEmployeur

Série Limitée... Dashboard + S... Accueil | Mons... Direct TV | To... Accueil | S.C.T... Phrasal Verbs... Courrier - MBI... Regarde gratu... YouTube News Calculatrice Sc... Comparer prix... Consignes po...

# MoBill\_Stage.

Entreprises Etudiants Offres Mes Candidatures Mon profil

Nom Entreprise :

Description :

Domaine :

Email :

Importez votre logo :

GestionCandidatures x Connexion - CAS - Centr... x Spring Boot Upload/Do... x Angular 14 File upload... x Boîte de réception (2.80) x

localhost:4200/employeurs

Série Limitée... Dashboard + S... Accueil | Mons... Direct TV | To... Accueil | S.C.T... Phrasal Verbs... Courrier - MBI... Regarde gratu... YouTube News Calculatrice Sc... Comparer prix... Consignes po...

# MoBill\_Stage.

Entreprises Etudiants Offres Mes Candidatures Mon profil

Logo : snapface.jpg

Nom : IMO

Description : Société de prestation de services Informatiques

Domaine : ingenierieInformatique

Logo : Q2.png

Nom : ea

Description : Fabrication de matériel d'imagerie Biomedicale

Domaine : ingenierieBiomedicale



## II.b. - Contenu des différentes couches

Nous commençons par créer les principales classes correspondant aux axes (avec leurs caractéristiques comme attributs) de notre application :

- Etudiant
  - Prénom
  - Nom
  - Matricule
  - Profil
  - Email
  - Etablissement scolaire fréquenté
  - Domaine d'étude
  - Année d'étude
  - Liste de candidatures
- Entreprise
  - Nom
  - Domaine
  - Identifiant
  - Description
  - Logo
  - Liste d'offres de stage publiées
  - Liste de candidatures reçues
- Candidature
  - Offre
  - Etudiant
  - CV
  - Lettre de motivation
- Offre
  - Entreprise
  - Titre
  - Description
  - Domaine

Evidemment, nos données sont gérées dans une base de données composée des tables correspondant respectivement à chacune de ces classes (et d'autres...). On voit avec la liste plus haut que des principales classes, que certaines dépendent d'autres, c'est-à-dire que certaines classes sont les attributs des autres. Pour implémenter cette spécificité, nous avons lié les tables dans la base de données en respectant les clés primaires et secondaires de chacune d'entre elles.



### III - Description de l'architecture à mettre en place

L'architecture de notre application est basée sur le modèle MVC (Modèle-Vue-Contrôleur) très répandu pour les technologies basées sur le JAVA. En effet, ce modèle permet non seulement de respecter un canevas précis tout le long de la réalisation de notre projet, mais il nous aide aussi à ne pas se perdre dans nos propres lignes de code.

Le principe est simple, le visiteur envoie sa requête http, la transmet au serveur d'application, qui lui la transmet directement au contrôleur, lui il se contente de faire le routage de l'information. Le contrôleur peut ensuite générer une page web appelée Vue, qui représente en réalité ce que le visiteur verra au final. Si nous superposons ces informations avec notre projet, nous obtenons ceci :

- Le modèle qui contient les informations structurées de nos données, qui peut faire des calculs sur celles-ci. Ceci représente donc les classes énoncées plus haut représentant les entités plus ou moins concrètes.
- Le contrôleur est pour le service qui nous permet de faire le routage de nos données dans les différentes classes.
- La vue représente le front qui quant à elle diffère du modèle MVC classique mais garde le même principe. En effet, nous avons créé un autre projet pour gérer le front en utilisant une toute autre technologie.

### IV - Technologies utilisées

Pour parvenir au résultat que nous avons ici, nous nous servons de différents langages et technologies. Nous allons maintenant les énoncer en précisant leur rôle dans le projet.

#### IV.a. - Spring Boot

C'est une framework dérivant de J2E qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en *jar*, totalement autonome.

Nous commençons par créer un projet Spring Boot dans lequel nous avons créé nos différentes classes et nos multiples services. On peut donc dire que nous nous sommes reposés sur ce langage pour définir l'ossature de notre projet. Pour faire simple l'API en elle-même est codé à partir de Spring à partir de modèle MVC (mais sans les vues).



## IV.b. - Angular

C'est une plateforme de développement construite sur TypeScript. Il comprend un cadre basé sur des composants pour la création d'applications web évolutives, une collection de bibliothèques bien intégrées qui couvrent une grande variété de fonctionnalités, une suite d'outils de développement pour vous aider à développer, construire, tester et mettre à jour votre code. Un composant Angular est composé de :

- Une page TypeScript qui décrit le fonctionnement de ce composant de par ses méthodes.
- Une page HTML (dans laquelle nous avons incorporé du Bootstrap pour optimiser la mise en forme de l'affichage) qui s'occupe d'afficher les informations du composant
- Une page SCSS qui met en forme l'affiche produite par HTML en lui donnant du style
- Une page spec.ts, il s'agit d'un fichier décrivant des tests unitaires à appliquer sur le composant pour vérifier son comportement.

Angular a donc servi principalement pour le front, c'est-à-dire les différents affichages et les animations nous retrouvons dans le tout le projet.

## IV.c. - MySQL

MySQL est un SGBD (système de gestion de base de données) relationnel utilisant le langage de programmation SQL. Nous avons donc créé la base de données, les tables et les liens entre les tables.



## V - Conclusion générale

Nous avons donc réalisé ce projet qui consistait à créer un service web permettant de promouvoir la relation entre les étudiants et les entreprises dans le cadre de la recherche de stage. Nous parlons de réalisation mais pour nous, cette période a plus été une phase d'apprentissage. En effet, les différentes notions et les différents langages et les différentes notions (Springboot que nous avons étudié préalablement en cours, Angular...) qui nous avons utilisé n'étaient pas forcément très connues. Cependant, cela ne nous a pas dissuader de faire des efforts pour les assimiler et les appliquer à notre cas d'utilisation. Il serait présomptueux de dire que le travail est terminé bien qu'étant satisfaisant. Nous nous sommes fixés des objectifs qui n'ont pas tous été atteints. Nous devons encore implémenter un service de messagerie. Pour finir, ce projet a été une réelle période d'agacement dans les moments d'échec, de satisfaction, de mécontentement et principalement d'apprentissage.