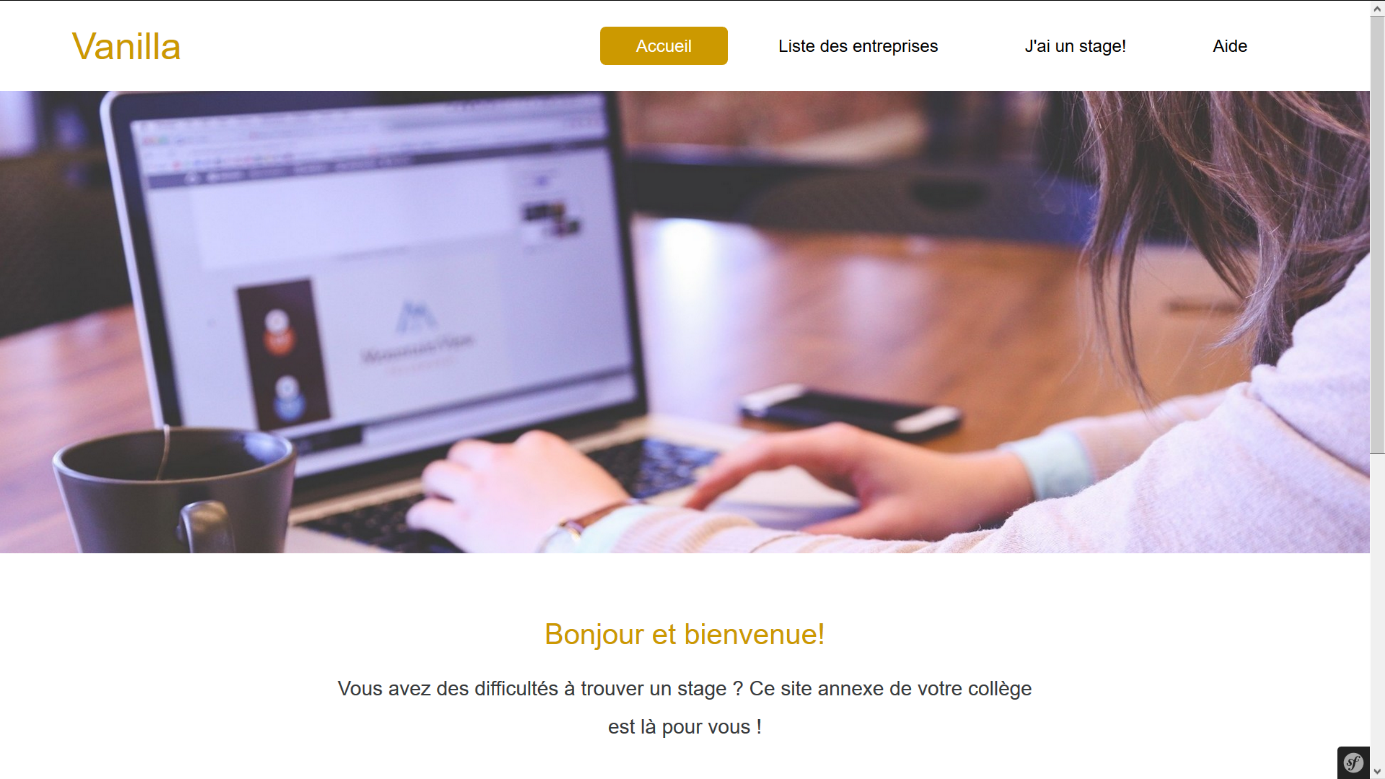
|  |
| --- |
| Documentation de l’application web : ‘Vanilla’ |



Projet professionnel -‘Vanilla’

Développer à partir d’un projet existant, refactoring avec la solution Symfony 4 et mySQL.

Mars 2018Crété​ ​Jonathan/Adelaide Donovan/Ball Philippe​ ​–​ **Lycée Léonard de Vinci**

**Cahier des charges**

Besoins essentiels :

* Reprendre un projet existant et le déporter avec le framework web Symfony 4.
* La gestion de l’authentification n’est pas nécessaire car le composant FOSUserBundle n’est pas disponible pour symfony 4.
* Réaliser une analyse complète avec un diagramme UML des entités du domaine, un schèma Relationnel et une diagramme USECASE complet
* L’application devra faciliter l’accès à la liste des entreprises locales qui accueillent, ou qui ont accueilli dans le passé, des élèves de 3eme.

Besoins additionnels :

* Concevoir un système de session à l’aide du composant ‘security’
* Accorder des droits administrateur a l’utilisateur « professeur »
* Intégration de Google Maps

**Description de l’application**

« Vanilla » est une application web destinée au collège les capucins, pour M.Fortin, à Melun.

Elle a pour principal objectif de faciliter l’accès à la liste des entreprises locales qui accueillent, ou qui ont accueilli dans le passé, des élèves de 3eme.

L’application comprend est adressé à deux types d’utilisateurs : Le professeur et l’éléve.

Cette application est un « refactoring » d’un application web déjà existante conçue en février 2018.

Afin d’assurer une maintenance évolutive, Vanilla a été développée avec le framework php

Symfony 4, réduisant le nombre de dépendances par rapport à Symfony 3 (au nombre de 3)

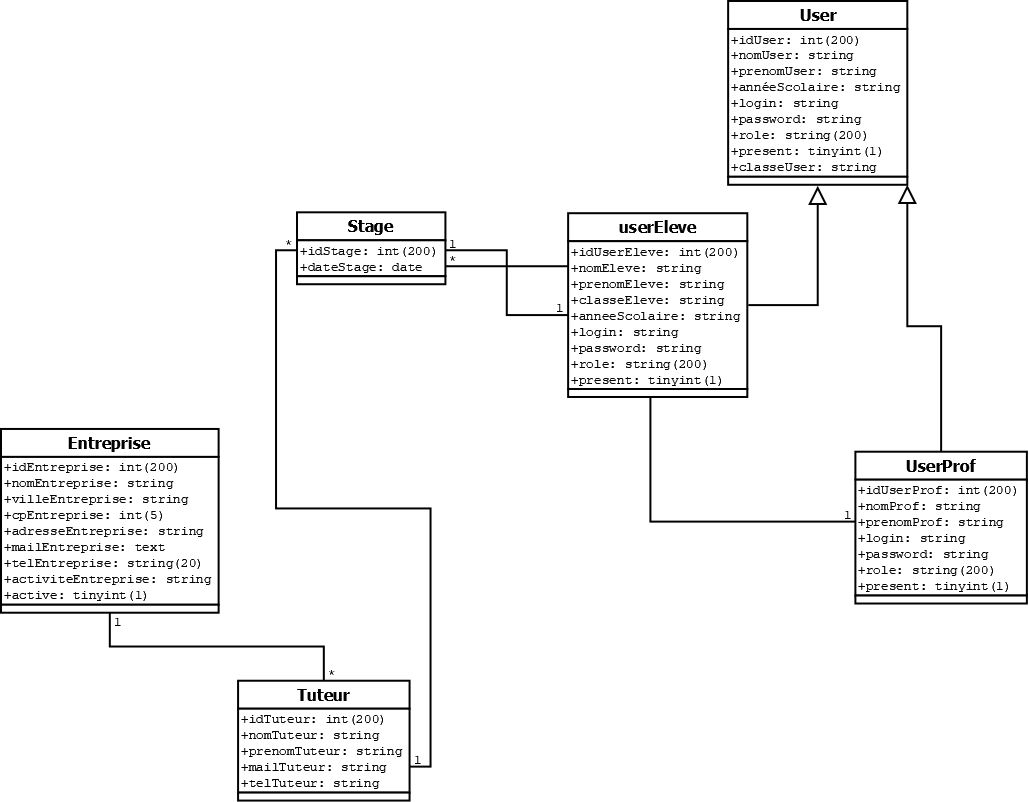
Le modèle MVC (Modèle\Vue\Contrôleur) à été utilisé afin de mieux structurer le code.

Celui-ci permet également une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur, mais aussi d’un gain de temps pour une potentielle team de développeurs qui reprendraient le projet dans le futur.

**Choix\contraintes techniques :**

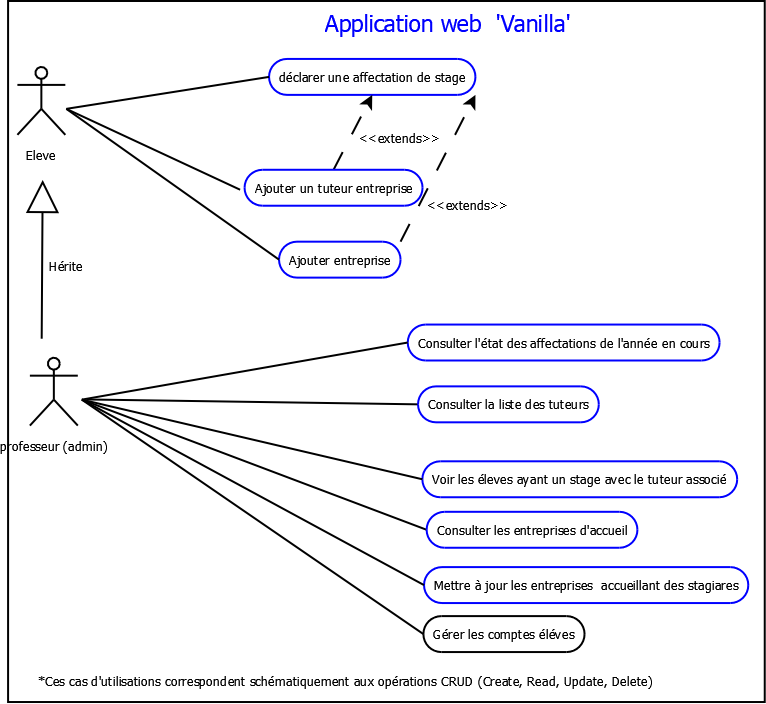
* Framework coté serveur : Symfony 4 (Php)
* Framework CSS (design) coté client : Bootstrap 4
* Utilisation du logiciel Laragon pour administrer une base de données mySql.
* Template responsive issu du site freecss.com
* Respect de la norme W3C
* Apprentissage du langage de template twig

**Diagramme UML d’analyse des entités du domaine**



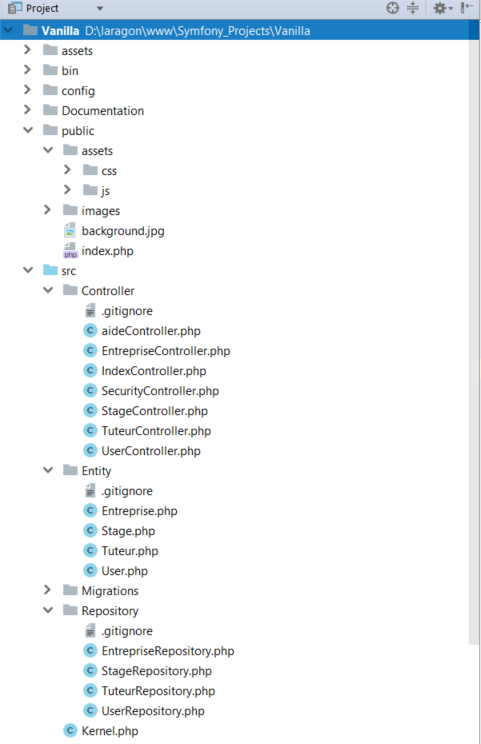
**Schéma Relationnel**

**Diagramme USECASE**



Les cas d’utilisations matérialisés par la couleur bleu ont été traités.

**Arborescence des dossiers**



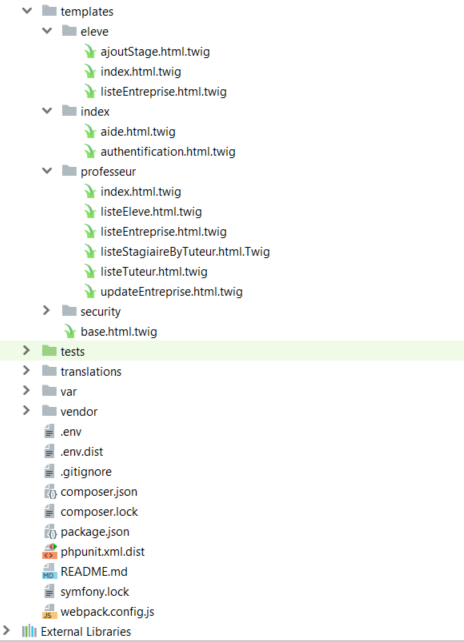
Répertoire contenant les différents fichiers de migrations BDD

Répertoire contenant les différents contrôleurs

Répertoire contenant les différentes entités

Composants web en accès public (css/js/images)

Fichiers de configuration



.gitignore : Fichier énumérant les fichiers qui doivent être ignorés par Git lors d’un commit

/vendor : Répertoire contenant les composants tiers

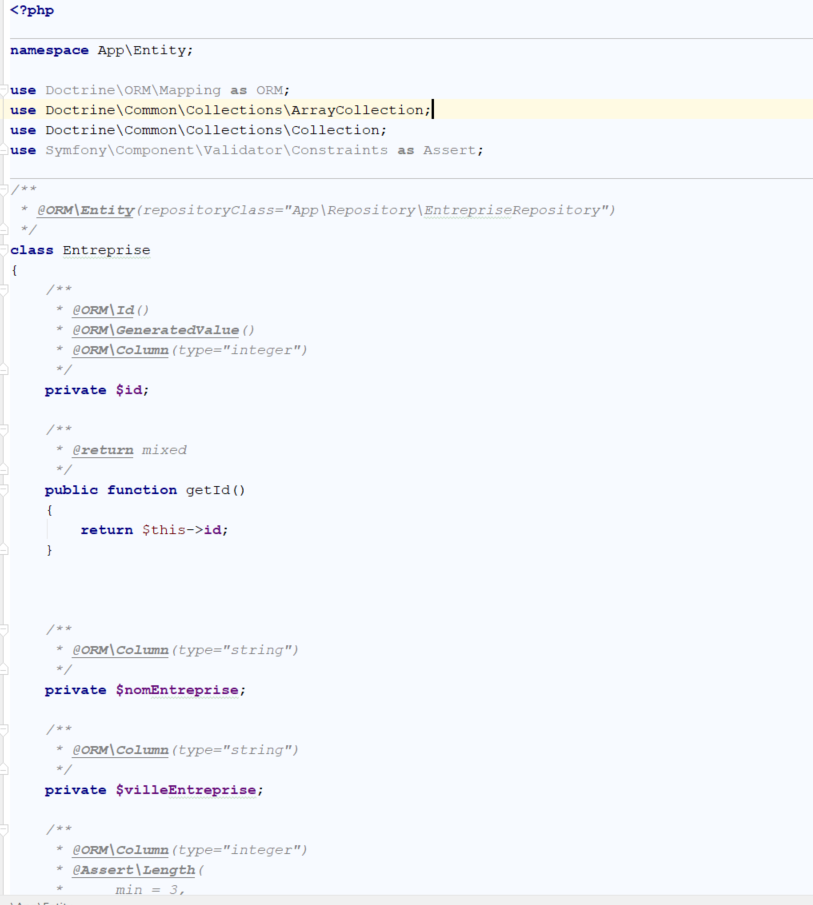
Répertoire contenant les différentes vues :

* templates/eleve : Vues liées à « l’espace élève »
* templates/index : Vues dites communes tant aux élèves que aux professeurs
* templates/professeur : Vues associés à l’espace professeur

**Extraits d’implémentation**

Cas d’utilisation : Lecture des entreprises d’accueil pour l’élève :

Extrait de l’entité Entreprise :



Contrôleur EntrepriseController.php  :



Appel de la méthode héritée (render) en lui passant le nom de la vue ‘listeEntreprise.html.twig’

Formulaire de déclaration d’une entreprise de stage élève:

Ici le screen....

Extrait de la vue ‘listeEntreprise.html.twig’ :



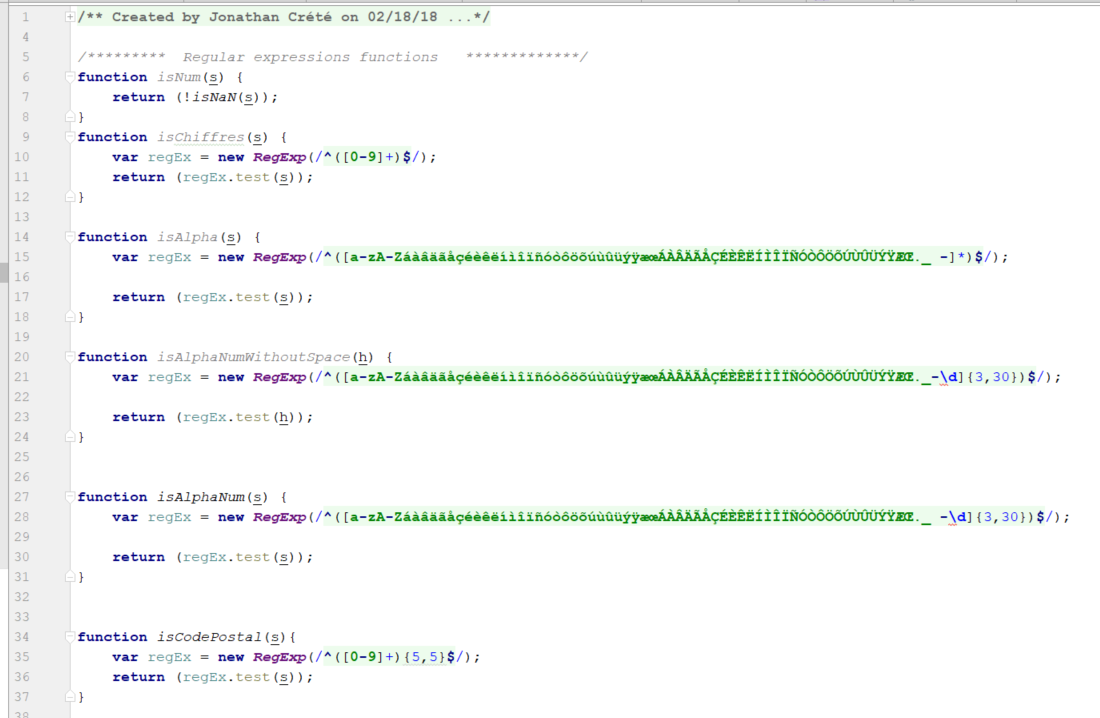
On remarque que les classes propres à Bootstrap 4 sont utilisées :

« Container » : Le container est un élément de disposition le plus fondamental et est requis afin d’utiliser le système de grille de Bootstrap v4.0.0

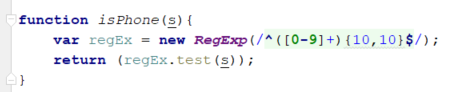
« for i in listeEntreprises » : Boucle permettant l’affichage des lignes du tableau.

**Fonctions javaScript**

Extraits du fichier RegFunctions.js :



Exemple :



1

2

Manipulation de la chaine de caractère du champ « téléphone »

1 -> Instanciation de l’objet RegExp, seul les chiffres de 0 à 9 sont autorisés et la chaine de caractère doit avoir une taille minimum de 10.

2 –> La méthode test() renvoie true si le test est réussi ou false si le test échoue

Extrait du fichier Vanilla.js (Utilisation librairie jQuery)



3

3 🡪 (jQuery) Si la valeur du champ form\_TelEntreprise est différente du format attendue alors on ajoute une classe ‘has-danger’ à l’élément parent de form\_TelEntreprise ( div\_formTelPy) pour signifier une erreur de saisie et on affiche du texte en dessous du champ.

Sinon, on ajoute une classe de succès signifiant la bonne saisie de l’utilisateur.

**Suivi & Problèmes rencontrés**

**Suivi :**

02/04/18 :

- modification des attributs des entités

- correction des liaison entre entité

- read des eleve pour les prof

- read des tuteur par entreprise pour les prof

- read des eleve ayant un stage selon le tuteur

- redirection vers google map selon l'adresse de l'entreprise

- intégration des fonctionnalité bootstrap pour tout les formulaire

- validateur sur les attribut par entité (pour les form)

- changement du background du template

- création de l'entité et controller user

**Problèmes rencontrés et éventuelle correction :**

* Problème d’implémentation du composant security (01/04/18)

**Hébergement**

Volonté d’héberger l’application web sur le service d’hébergement Hostinger.