

Réaliser Par : Année : 2015/2016

Mohamed Salah BENBAKH LPU-SIR G1

Rapport de stage de fin de formation

Conception et réalisation d’Application web pour gestion des stages et stagiaires dans la Wilaya de Marrakech-Safi

# Introduction

Dans le cadre de la formation continue de la Licence Professionnel d’Université SIR au sein de la Faculté des Sciences et Techniques de Marrakech, Nous sommes amenés à passer un stage de deux mois dans une entreprise privé ou établissement public, en réalisant un projet de fin de formation pour approuver nos connaissances acquises au cours de l’année, et pour concrétiser le travail pédagogique par un travail Professionnelle.

Dans mon cas j’ai choisie de passer mon stage dans la Wilaya de Marrakech-Safi.

Au cours de mon stage j’ai proposé à mon Maitre de stage Mr Hassane (…) de réaliser une application web pour la gestion des stages, afin de pouvoir mieux faire le suivie ce cette activité.

Chapitre 1 :

# Contexte générale du projet

## Problématique :

Faire le suivi d’un stagiaire est une activité demandant beaucoup de temps et d’énergies, pas mal de documents :

* le candidat qui doit se déplacer plusieurs fois pour avoir une réponse à sa demande de stage
* Associer le stagiaire à un département
* Proposition de Sujets
* Suivie d’avancement de projet
* Etc.

Ceci est dit, mais faire le suivie de plusieurs stagiaires, chacun d’un établissement différente, spécialité différente et sujet différent peut très vite tournée assommant, pour la division de Ressources Humaines, les Maitres de stages et aussi pour le stagiaire et son établissement.

## Objectifs et Solutions :

Notre objectif pour le candidat :

* Faciliter son association
* Simplifier la réception de réponse
* Disponibilité de Sujets Riche

Pour Le Maitre de Stage :

* Coordination et communication meilleurs avec la RH
* Disponibilité de sujets (proposer par l’ensemble des départements de la Wilaya)
* Organisation des Stagiaires
* Simplifier le suivie d’avancement des projets
* Possibilité de communication distante avec les Stagiaires et leurs Etablissements
* Gestion d’absence

Pour la division de Ressources Humaines :

* Réponses rapide aux candidats
* Gestion de Documents
* Association de Stagiaires aux départements

On fixe l’utilisation de la solution au Maitre de Stage (Responsable du stagiaire dans la wilaya), Le Stagiaire et la Ressources Humaine.

**Description fonctionnelle :**

Ressources Humaine :

* + Ajouter les demandes de stages
  + Stocker les documents de stages
  + Envoyer réponses par email aux candidats
  + Informer le département concerné

Maitre de Stage :

* + Valider ou refuser une demande
  + Affecter un sujet à un stagiaire ou groupe de stagiaires
  + Suivre l’avancement des sujets
  + Contacter les stagiaires

Stagiaire :

* + Ajouter une description phase de son sujet
  + Contacter son Maitre de stage

**Règles de gestion :**

* L’accès à l’application est restreint et se fait par Authentification :
  + RH :
    - Gestion Demandes
    - Gestion stagiaire
    - Gestion sujets
    - Informer un département
    - Envoyer email de réponse au candidat
  + Maitre de stage :
    - Gestion de ses stagiaires
    - Communiquer avec ses stagiaires
    - Gestion de taches
    - Approuver ou non une tache ajouter par le stagiaire
    - Suivie de taches et délais
  + Stagiaire :
    - Ajouter une tache
    - Communiquer avec son Maitre de stage
* Chaque utilisateur ne peut faire que les actions dont il a droit (en dessus)
* La RH peut ajouter les documents sous forme d’image
* Aucune entité enregistrer ne peut être supprimé
* La communication entre les utilisateurs de l’Application doit se faire en temps réel
* Garder un journal d’authentification des Stagiaires pour valider la présence
* Le Maitre de stage peut valider ou annihiler une tache faite par le stagiaire

Chapitre 2 :

# Analyse et Conception

## Méthode :

Les traitements et les relaxions dans le projet rendent l’utilisation de la méthode MERISE handicapante, cette dernière ne répond pas aux besoins organisationnelles ni aux besoins du suivis, ce qui est le but de notre projet, c’est pour cela que mon choix s’est tendu au langage UML (pas comme si on a mille choix).

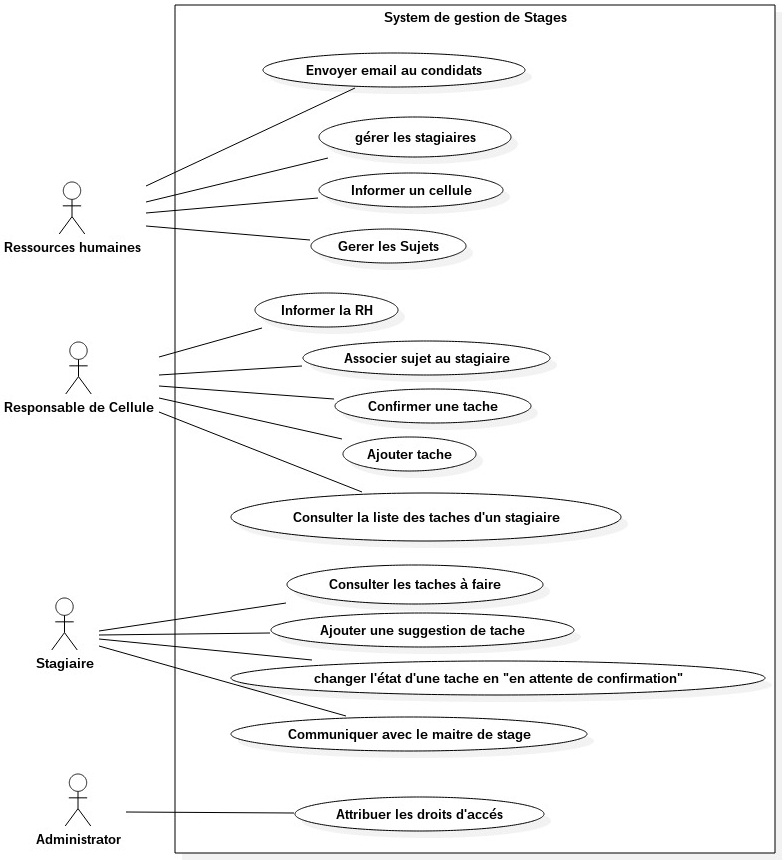
### Présentation UML :

Celon Wikipedia UML () est « *un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système* », Couramment usé en Programmation Logiciel et conception Orienté Objet, d’où son internet dans notre cas.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

* Activité d'un objet/logiciel
* Acteurs
* Processus
* Schéma de base de données
* Composants logiciels
* Réutilisation de composants

### Diagramme de Cas d’Utilisation :

  
Figure 1 : diagramme de case d’utilisation

### Diagramme de classes :

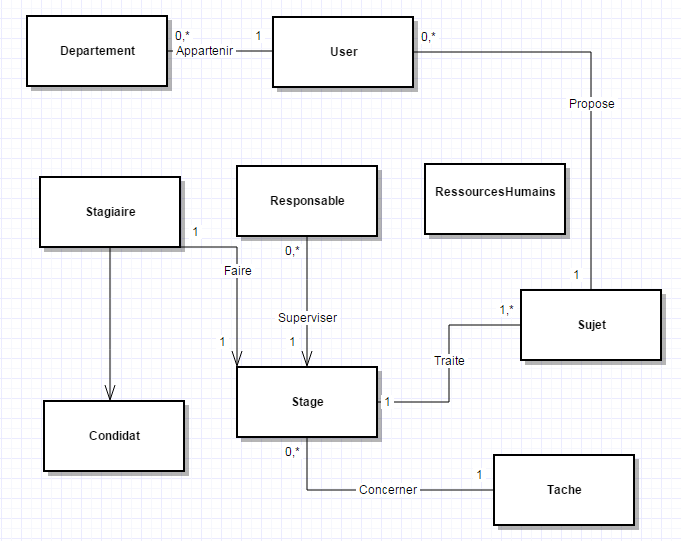


Figure 2 : Diagramme de classes sans attributs

### Diagrammes de séquences :

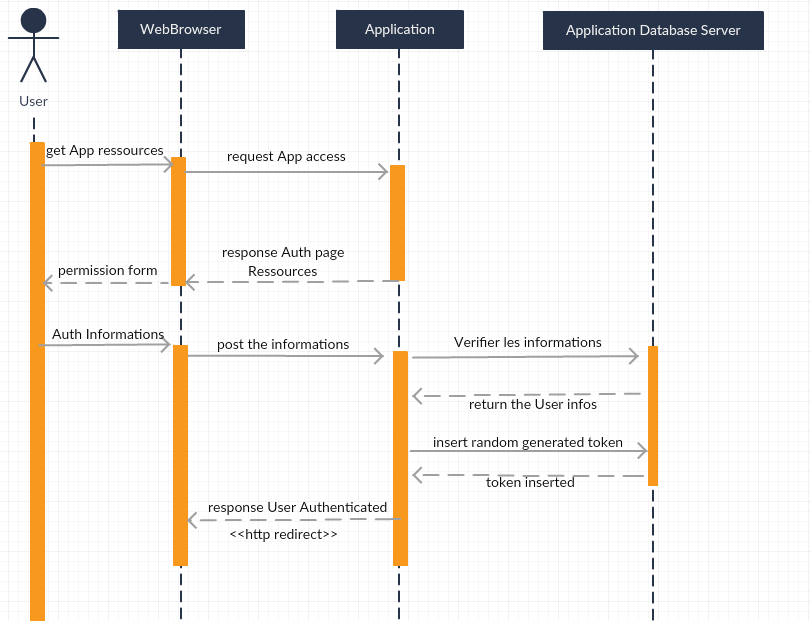


Figure 3 : Scenario de première authentification avec « remember me »

Lors de la première Authentification d’un utilisateur (non rappeler par le navigateur) le navigateur demande la page d’authentification, l’utilisateur remplie alors les champs « email » et « mot de passe » et soumit la requête, le système fait un cryptage du mot de passe et vérifie si ce couple du email/password existe sur la base de données. Dans le scenario parfait où l’utilisateur soumit les bon identifiants le system alors génère un nombre aléatoire et le crypte de la même méthode que le mot de passe et l’insert pour cet utilisateur autant que « remember\_token » qui servira pour le faire rentrer dans sa session prochainement sans avoir à saisir ses identifiants, la figure 4 expliquera mieux le mécanisme.

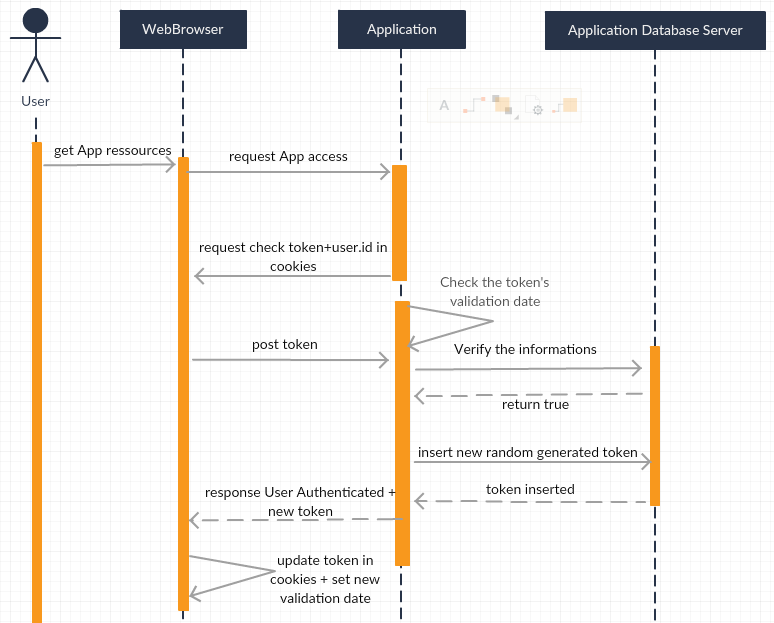


Figure 4 : Scenario parfait d’authentification sans login/psw

Dans notre system on veut que l’utilisateur peut avoir accès à sa session son reSaisir ses identifiant, cela uniquement si il a ouvert sa session dans les dernières 12 heures.

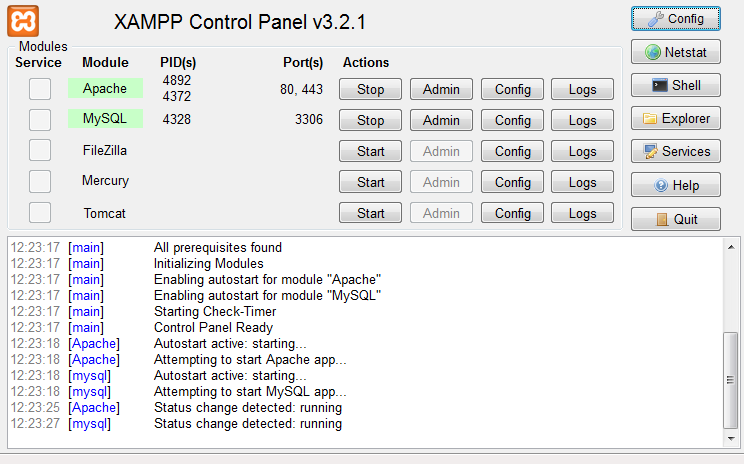
Après un authentification normal on stocke les informations de l’utilisateur (id, nom, prénom, email, token, dernier accès et autres) sans le mot de passe, alors si dans moins de 12 heures il veut rouvrir sa session le système n’aura qu’à vérifier le token (qui est aléatoirement générer et crypté) s’il correspond bien à l’id dans le cookie, au final, deux oiseaux avec une pierre, le mot de passe n’est pas stocké localement (même crypté est un risque) et le système est User Friendly.

Chapitre 3 :

# Outils de Travails

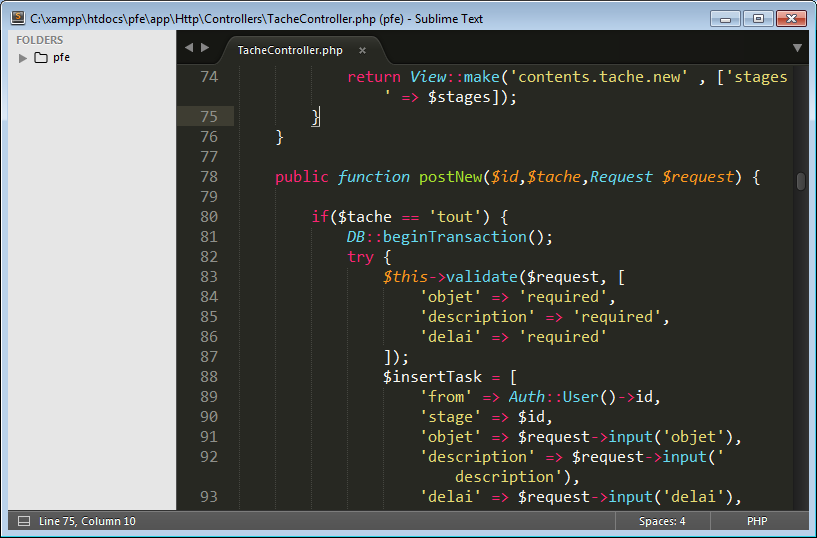
## Logiciels :

### XAMPP :



Environnement de travail, rassemble des logiciels permettant de mettre en place facilement un serveur Web confidentiel en local.

### Sublime Text 3 :



Editeur de texte hybride et extensible disponible du Windows, MacOS et Linux, puissant, simple et ergonomique avec une large communauté.

Sublime Text intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l’auto complétion, un système de plugins… L'éditeur propose cependant des fonctions plus avancées, dont :

* Minimap : prévisualisation de tout le fichier dans une barre latérale.
* Sélection et édition dans plusieurs sections de code en parallèle.
* Marque-page au sein même des fichiers.
* Sauvegarde automatique.
* Recherche et remplacement par expressions régulières.
* Support des macros et de plugins en Python.
* Personnalisation des raccourcis clavier.

### Git (github):

 Un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s’agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes (*en 2 mai 2016*).

Pour le développement de l’application j’ai choisie d’utiliser Github comme *Serveur origin.*

EXPLICATIONS :

L’intérêt du git est le suivi des étapes du développement, on remarque bien son impact lors de grand projets développer par une/des équipe(s).

Le principe est simple mais puissant, on a un serveur et client(s) partageant du code, on peut modifier le code en local en suite envoyer les modifications au serveur, pour cela on demande à git en local de verifier les différences (lignes ajouté/supprimer) modification d’une ligne = suppression du ligne + ajout ligne, et une fois détecter, on « commit » les changement avec un message significatif de ce changement (par ex : ajout model User) et ensuite on envoi cela au serveur Git, ça permet d’une part de faire un suivie du code et organiser les taches et d’autre part de bien distribuer les travaux on gardant le travail centralisé.

Pour une utilisation pour avancer je vous invite à consulter la documentation (simple et claire) :   
<https://git-scm.com/doc>

## Langages et Frameworks :

### PHP 5.6 + Laravel 5.2

 https://laravel.com/assets/img/laravel-logo.png

Laravel est un [Framework](https://fr.wikipedia.org/wiki/Framework) [web](https://fr.wikipedia.org/wiki/Web_application) [open-source](https://fr.wikipedia.org/wiki/Open-source) écrit en [PHP](https://fr.wikipedia.org/wiki/PHP) respectant le principe [*modèle-vue-contrôleur*](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur) et entièrement développé en [*programmation orientée objet*](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Laravel est distribué sous [licence MIT](https://fr.wikipedia.org/wiki/Licence_MIT), avec ses sources hébergées sur [GitHub](https://fr.wikipedia.org/wiki/GitHub).

Créer par Taylor Otwell en 2011, en se basant sur le Framework Symfony2, et collectant du monde du développement web les meilleurs composants et outils, par exemple :

* SwiftMailer pour l’envoie des emails (et non phpMailer)
* Eloquent pour buildeur de requêtes
* DotEnv pour les variables d’environnement

Et bien d’autres.

L’une des points forts de Laravel qui le distingue est son rapport simplicité/performance et son indépendance complète de la couche MODEL, qui rend son traitement complètement transparente grâce à Eloquent, et un routing moins pénible que ses concurrents.

Pour plus de détails consultez le chapitre 5 : bout de codes.

Vous pouvez aussi consulter la documentation :

<https://laravel.com/docs/5.2>

Et l’API :

<https://laravel.com/api/5.2/>

### HTML5 + twitter Bootstrap3 + JQuery :

L’***HyperText Markup Language***, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C’est un langage de balisage permettant d’écrire de l’hypertexte, d’où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents avec des équipements très variés de manière conforme aux exigences de l’accessibilité du web.

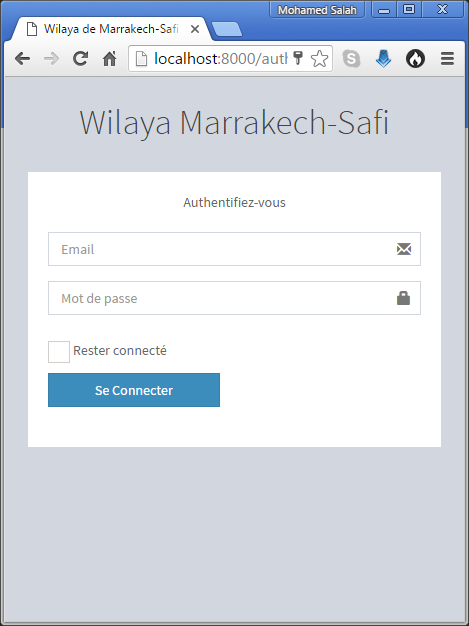
**HTML5** est la dernière révision majeure HTML. Cette version a été finalisée le 28 octobre 2014.

 **Bootstrap** une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

Chapitre 4 :

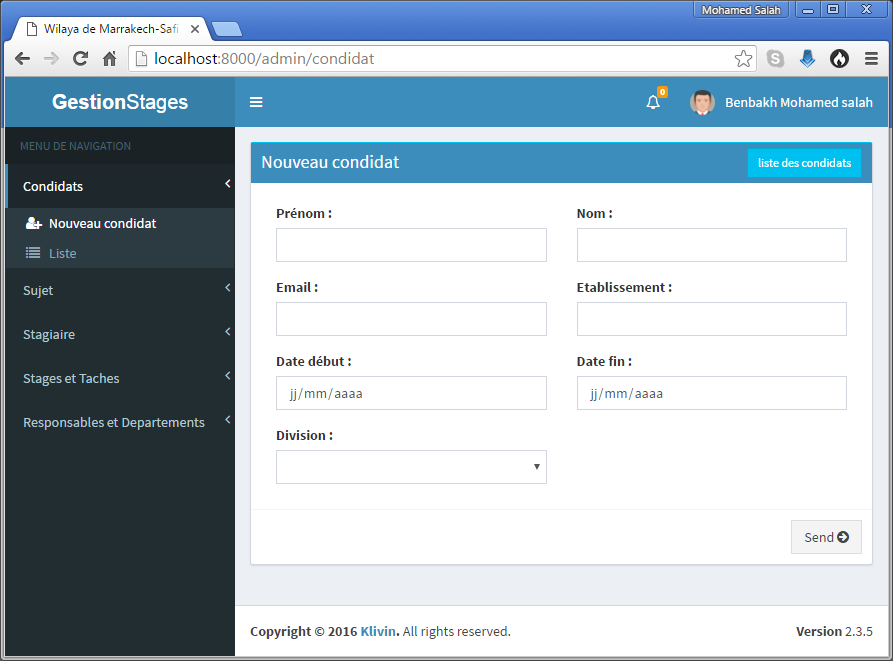
# Interfaces Principales

## Authentification :

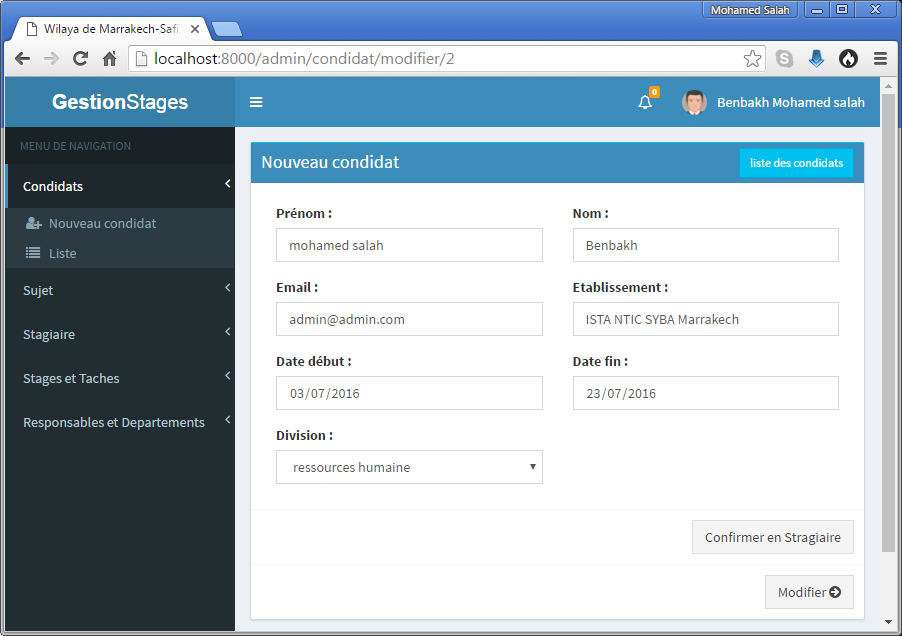


## Candidats :

### Nouveau :

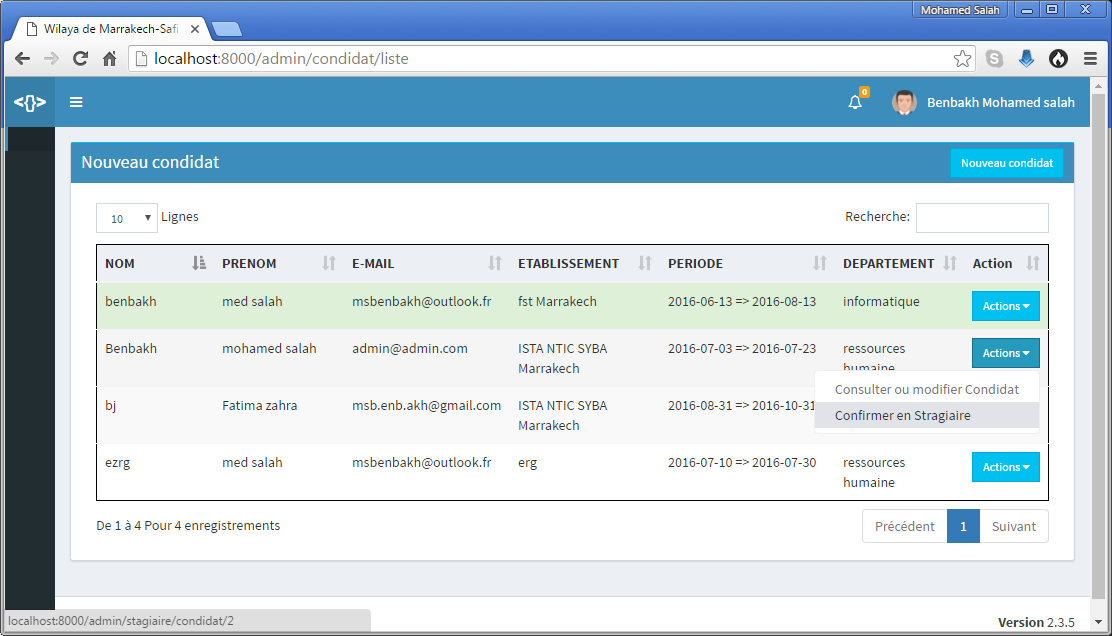


### Modifier :

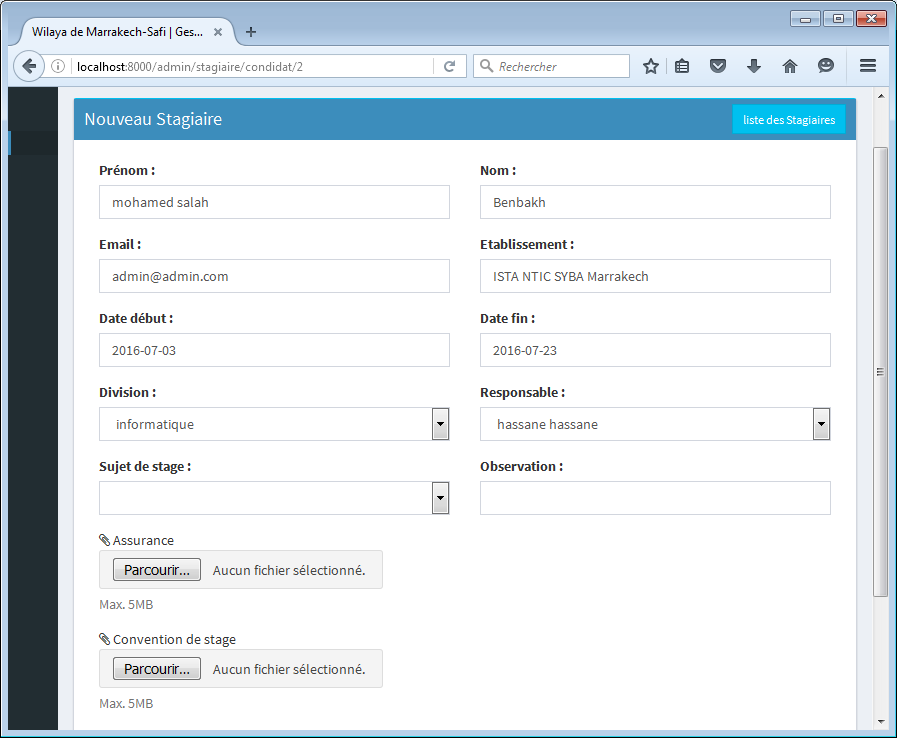


Dans le cas où le candidat n’est pas encore confirmé pour un stage on a la possibilité de le faire depuis cet interface (si l’utilisateur est un administrateur ou appartient au même département), ou bien dans la liste.

### Liste :

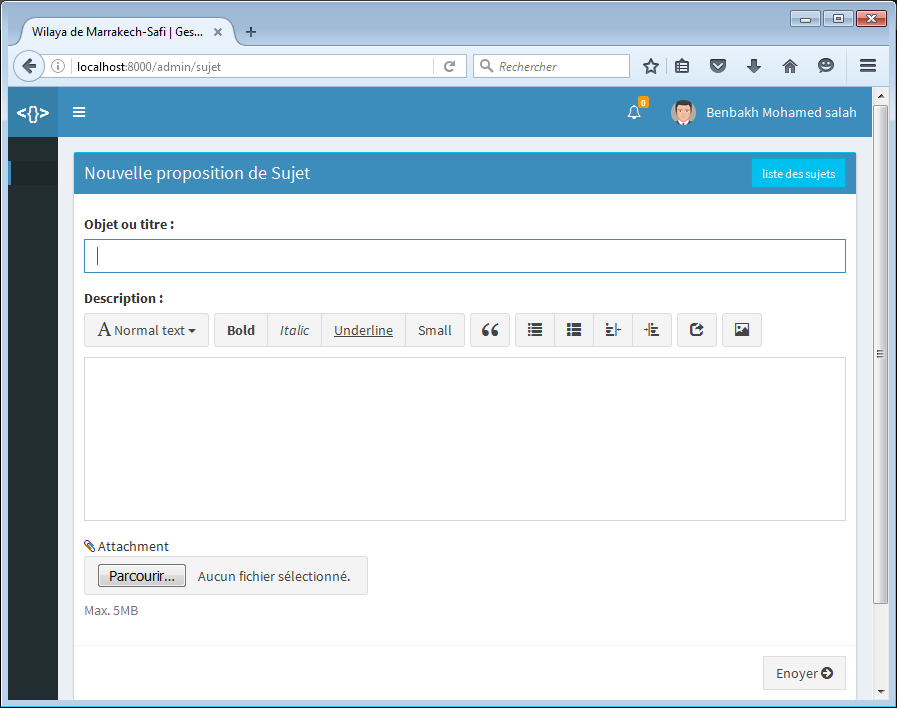


## Stagiaire :



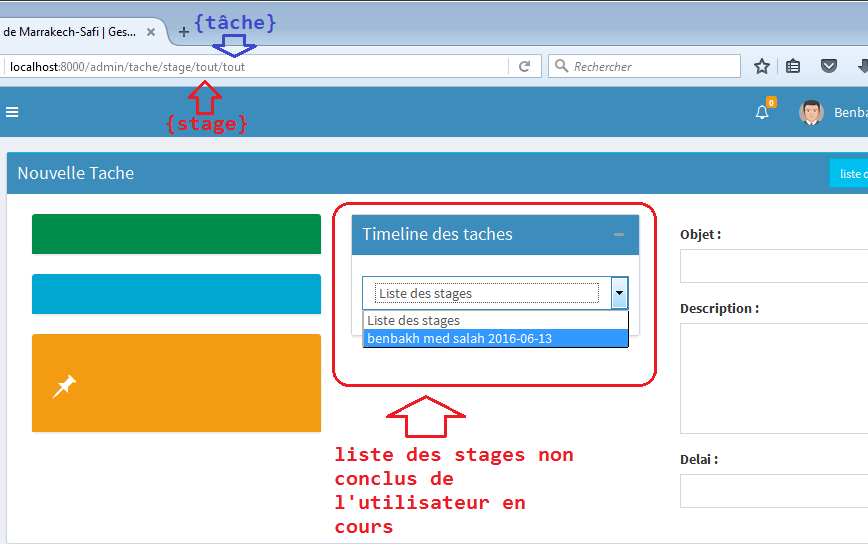
Après avoir choisie de confirmer le candidat en stage cette interface s’ouvre avec ses informations, demandant plus de détails pour le stage.

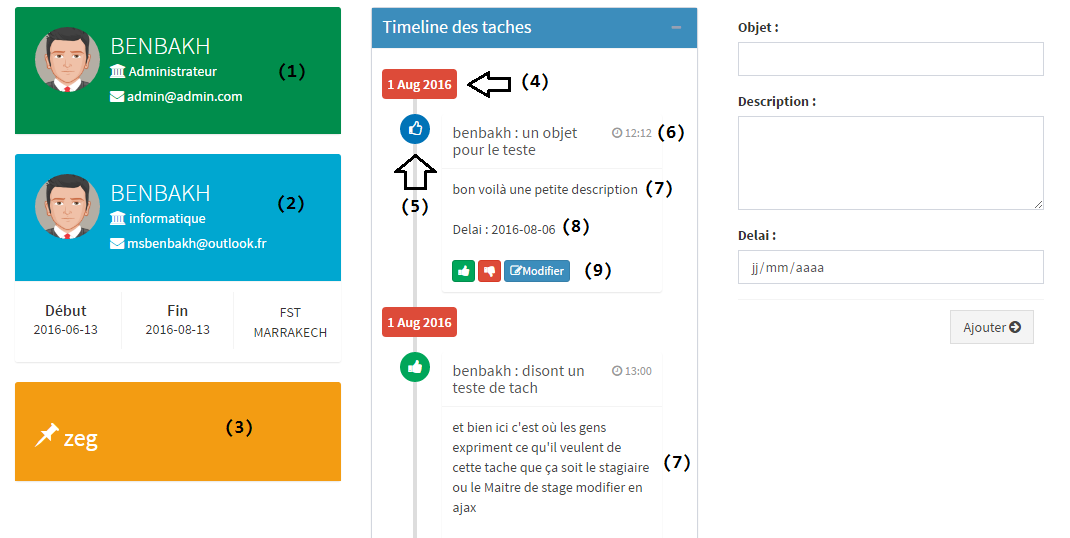
## Sujets :



Dans l’Application on veut (comme mentionnée dans le cahier des charges) que tout les personnel des départements puissant suggérer des sujets de stages qui leurs semble utile dans leurs propre département.

## Tâches :





1 : Maitre du stage

2 : Stagiaire

3 : Sujet

4 : Date de soumission de la tache

5 : Etat du tache (bleu : demande de validation | Rouge : à refaire | vert : Validé)

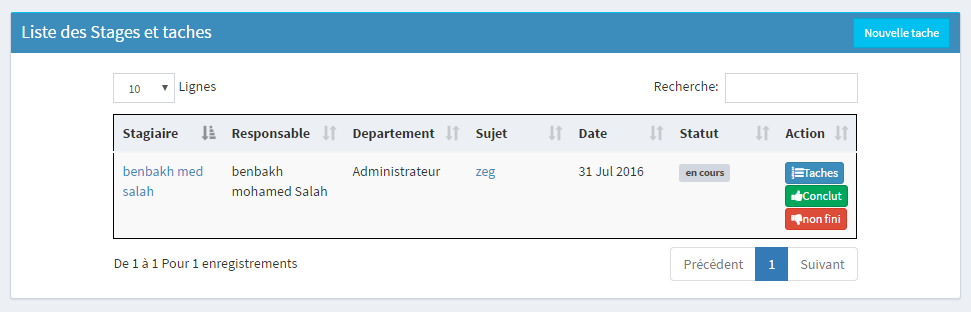
6 : Heure de soumission de la tache

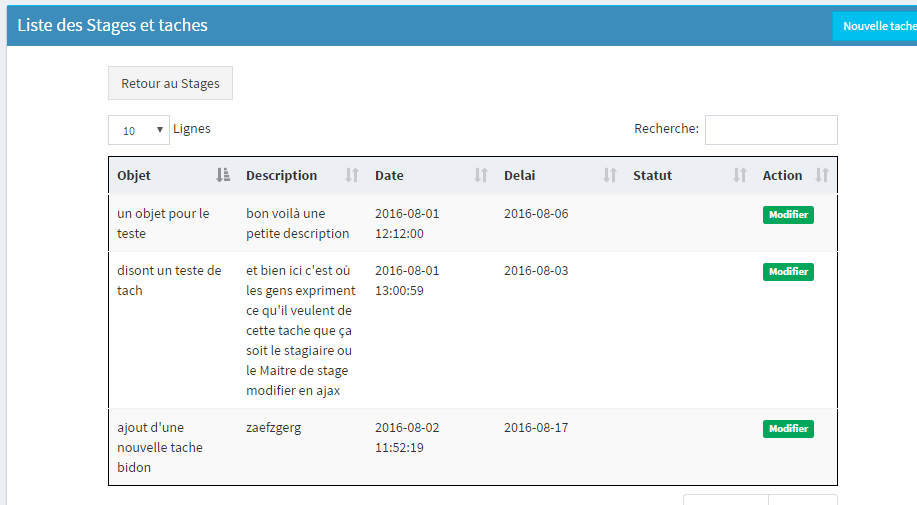
7 : Description de la tache

8 : Délai de réalisation

9 : Modificateur d’état

Il faut savoir que les modificateurs d’état ne sont pas accessible de la même façon pour tout les types d’users, un stagiaire ne peut que demander une validation d’une tache et non le mettre à refaire ni la valider.



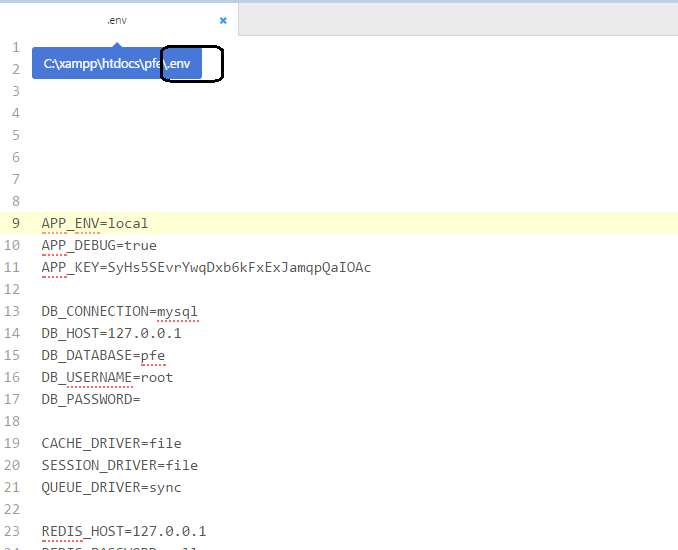


On propose une autre interface des listes des stages qui nous donne des informations globales, et sans quitter la page on peut voir aussi la liste des taches du stage en question.

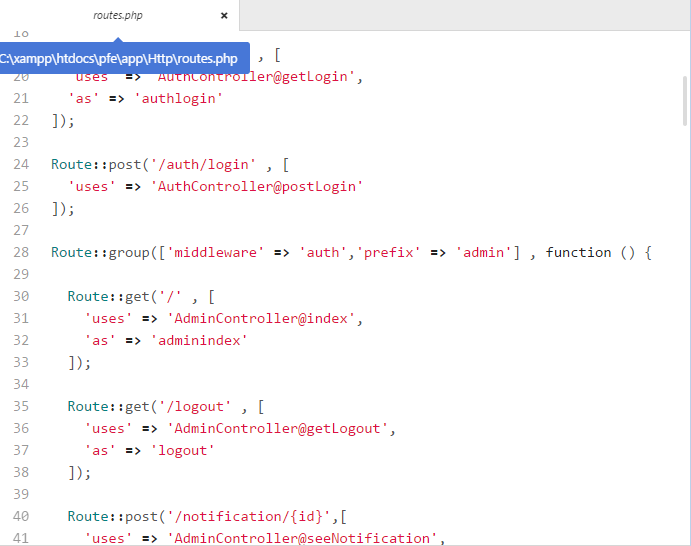
Chapitre 5 :

# Bout de Codes

## Configuration d’environnement :



## Routes :



Comme son nom l’indique le fichier route contient les différentes routes de l’application, un aspect puissant du design pattern MVC, en effet, avec les routes on n’a pas à se soucier de la sécurité (peu soucier) de notre Application, pas d’injection de code puisque les pages véritable restent inconnue pour l’internaute, par exemple il n’y a pas de page nommé « login » ni dossier nommé « auth », et ça permet d’envoyer des variable en GET d’une façon transparente.

<https://laravel.com/docs/5.2/routing>

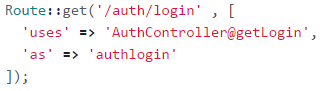
## Controlleurs :



Les Controllers sont le core de l’application, là où se font les traitements.

Après avoir saisir un chemin de route, l’application le ait correspondre à un controlleur (namespace : *App\Http\Controllers*) et une méthode dans le dit controlleur.

Dans cet exemple la route est :



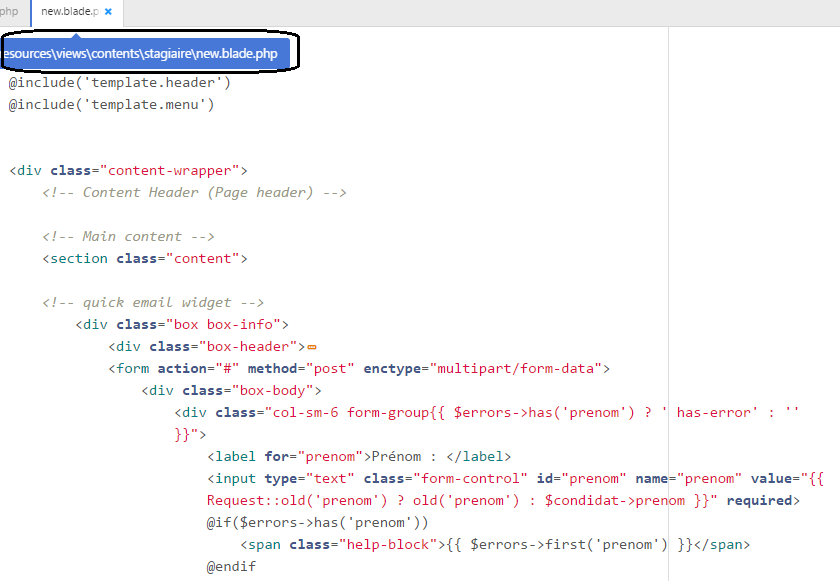
Donc la méthode est GET => Controller : AuthController::getLogin()

Dans cette fonction :

Si l’authentification est via une session déjà ouverte on fera une redirection vers une autre route, si non affiche la vue login.

<https://laravel.com/docs/5.2/controllers>

## Les Views :



Laravel utilise « Blade » comme système de templating pour les Vues, à l’égard de « twig » en Symfony et CakePHP.

On a choisie de décomposer une vue en quatre parties :

* Header : inclus l’entête des pages (profile et notifications) et les dépendances CSS.
* Menu : Partie menu des pages, le menu s’affiche en dépend du type d’utilisateur.
* Corps : c’est la partie dynamique de la page
* Footer : Pied du page comportant le ©Copyright et les scripts JavaScript nécessaire pour tout les pages.

<https://laravel.com/docs/5.2/blade>

## Les Migrations :

## 

Les migrations en Laravel c’est comme un contrôleur de versions de la base de données, permettant aux équipes d’avoir le même schéma de base de données.

Pour créer une migration on fait un coup de php artisan en terminal :

**$> php artisan make:migration nom\_de\_la\_migration**

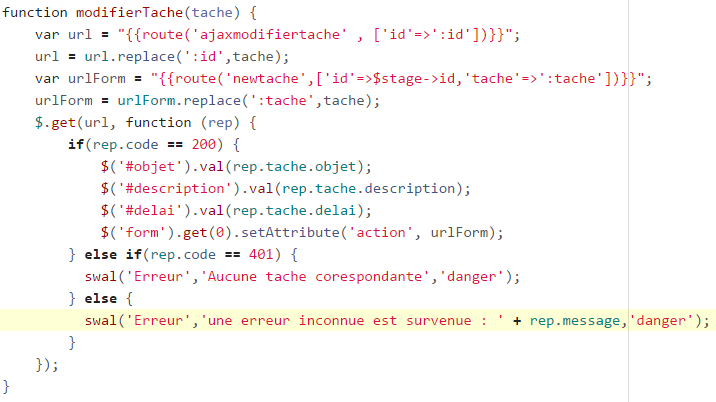
Par exemple si on veut ajouter une table life\_circle :

**$> php artisan make:migration create\_table\_lifeCirle –-create life\_circle**

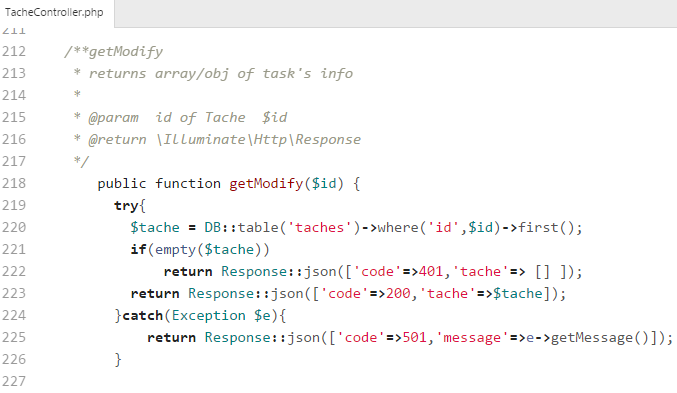
<https://laravel.com/docs/5.2/migrations>

## Ajax :

### Frontend :

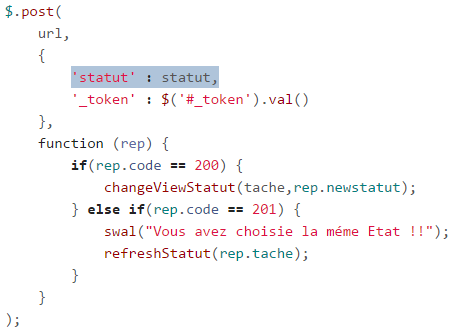


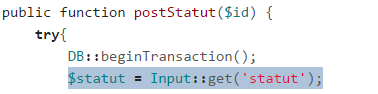
### Backend



La même structure et organisation pour tout les WebServices Ajax de l’Application.

Pour récupérer les Données soumit en méthode post en utilise la classe « Input »





# Conclusion :

Au terme de notre Fin de formation au sein de la FST Marrakech j’ai proposé comme projet de fin d’étude : une Solution pour Gestion de Stage et Stagiaires dans la Wilaya de Marrakech-Safi.

Bien qu’il ne soit pas mon premier projet Professionnel, ça m’a permis d’améliorer l’organisation de mon code, et d’éviter les erreurs faites dans les autres projets et découvrir d’autre pour y remédier.

Pour le Futur de la Solution, le Rapport subjectif, est pour permettre au éventuelle futur stagiaires d’avoir une base solide, bien organiser et susceptible aux améliorations.