

TPI – Canap-Gest

« CANAP-GEST » APPLICATION WEB DE GESTION DE
CANDIDATURES D'APPRENTISSAGE EPFL

Sommaire

| | |
|---|----------|
| SOMMAIRE | 1 |
| 1 ANALYSE PRÉLIMINAIRE | 2 |
| 1.1 INTRODUCTION..... | 2 |
| 1.2 SITUATION ACTUELLE | 2 |
| 1.3 OBJECTIF | 4 |
| 1.4 TRAVAIL PRÉALABLE..... | 5 |
| 2 ANALYSE DU PROJET | 7 |
| 2.1 POINTS IMPORTANTS DU CAHIER DES CHARGES | 7 |
| 2.2 MATÉRIEL ET RESSOURCES À DISPOSITION | 7 |
| 2.2.1 GLOBAL | 7 |
| 2.2.2 DÉPENDANCES | 7 |
| 2.3 FAISABILITÉ DU PROJET | 7 |
| 2.4 PLANIFICATION..... | 7 |
| 2.5 VERSIONNING ET SAUVEGARDES | 7 |
| 2.6 TECHNOLOGIES & DÉPENDANCES..... | 7 |
| 2.6.1 DESCRIPTION | 7 |
| 2.6.2 INSTALLATION | 8 |
| 3 RÉALISATION | 8 |
| 3.1 MODÈLE DB | 8 |
| 3.2 STRUCTURE DE L'APPLICATION..... | 8 |
| 3.3 AUTHENTIFICATION | 8 |
| 3.4 REQUÊTES API | 8 |
| 3.5..... | 8 |
| 4 CONCLUSION | 8 |
| 4.1 ATTEINTE DES OBJECTIFS | 8 |
| 4.2 PLANIFICATION | 8 |
| 4.3 PROBLÈMES RENCONTRÉS | 8 |
| 4.4 AMÉLIORATIONS POSSIBLES | 8 |
| 4.5 ÉVOLUTION..... | 9 |
| 5 ANNEXES | 9 |
| 5.1 SOURCES | 9 |
| 5.2 COMPTE-RENDU DES DISCUSSIONS..... | 9 |
| 5.3 JOURNAL DE TRAVAIL | 9 |
| 6 GLOSSAIRE | 9 |

1 Analyse préliminaire

1.1 Introduction

Il y a deux ans, j'ai développé un formulaire permettant la postulation en ligne des potentiels apprentis, pour la formation apprentis EPFL. Ceci visait à remplacer le formulaire papier envoyé par poste qui était jusqu'à présent utilisé.

Ce formulaire permet de saisir diverses informations (personnelles, scolaire, professionnelles ...) et de joindre des fichiers annexes (cv, lettre de motivation etc.).

Le système est stable et fonctionne bien, cependant son utilisation comporte encore quelques contraintes. L'idée du projet « Canap-Gest » naît donc du principe de l'amélioration de ce système.

1.2 Situation actuelle

La personne désirant postuler pour un apprentissage à l'EPFL se rend sur le formulaire, crée un compte au travers du service « compte guest » de l'EPFL, puis remplit ses informations et joint ses fichiers annexes.

Les résultats des postulations sont stockés sur un partage réseau réservé à cet effet, dans un dossier « nouvelles postulations ».

Le partage est structuré en 4 dossiers :

- Nouvelles : Résultats des postulations
- Valides : Dossiers valides
- Exclues : Dossiers non valides, incomplets, etc.
- Engagées : Dossiers retenus et prochainement engagés

Les responsables de la formation apprentis EPFL s'occupent de procéder à la sélection des dossiers et de transmettre les dossiers aux formateurs qui choisiront leur(s) futur(s) apprenti(s).

Les résultats des postulations suivent une structure de dossier spécifique, triée premièrement par métier, le dossier du postulant contient ensuite les fichiers annexes et les informations, celles-ci sont stockées dans un fichier JSON.

Le diagramme suivant récapitule la stratégie actuelle :

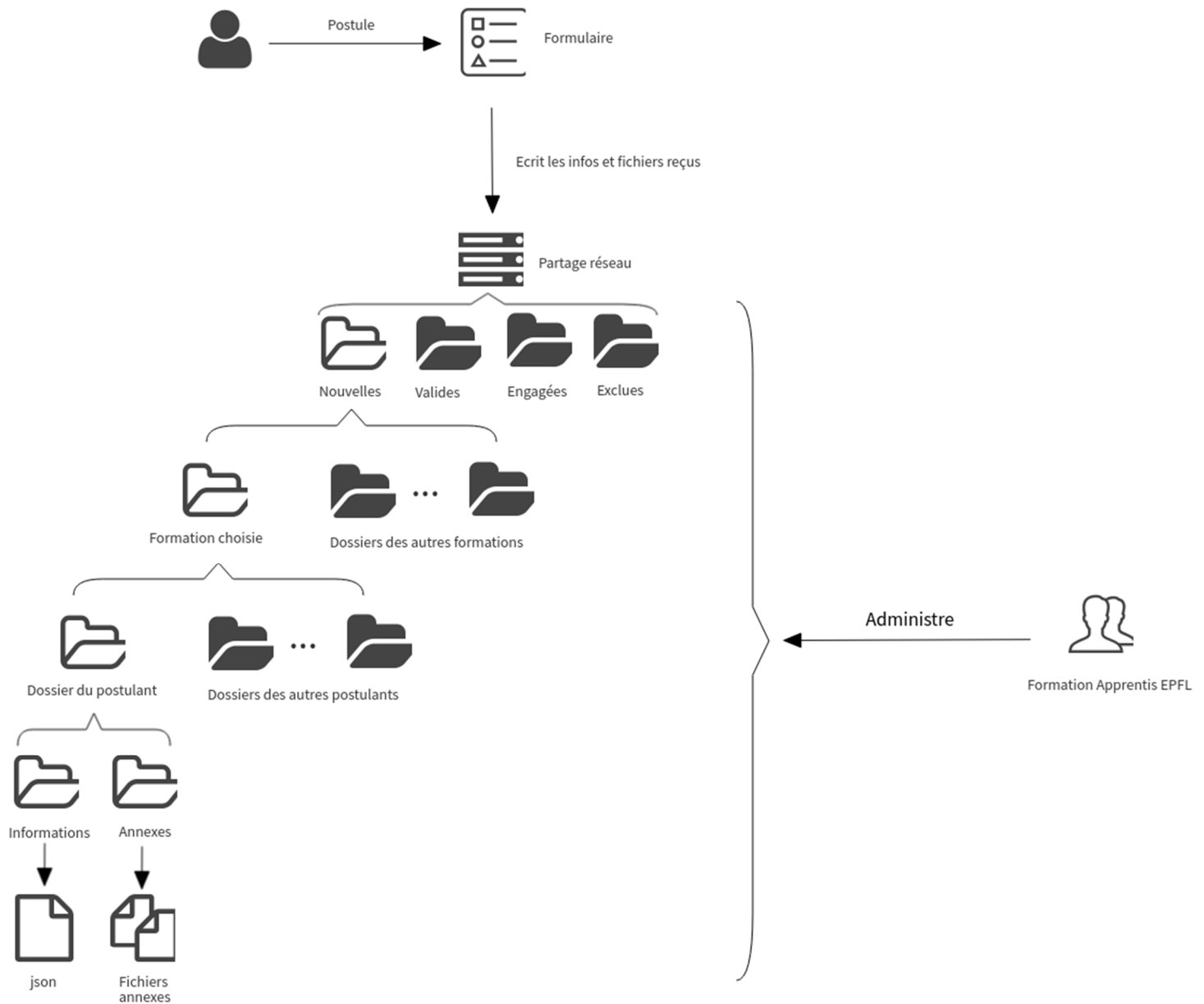


Figure 1: Fonctionnement actuel du système

Quelles sont donc les contraintes actuelles ?

- Les informations personnelles du postulant sont stockées dans un fichier JSON, il n'est donc pas facilement lisible sans utiliser quelconque outil pour faciliter son accès.
- Les responsables modifient directement les dossiers sur le partage, en cas de mauvaise manipulation (par exemple glisser-déposer au lieu de copier-coller) cela peut entraîner des problèmes (fichiers corrompus, permissions modifiées etc.).
- Nous ne disposons pas d'une vue globale sur les postulations.
- Nous ne disposons pas d'une manière de trier et comparer efficacement les postulations entre elles selon certains critères.

Des améliorations au niveau de la protection des données doivent également être appliquées, par exemple, dans l'état actuel, un candidat ne peut pas voir, modifier ou supprimer sa ou ses postulations effectuées, ce qui entraîne des conflits avec la loi RGPD.

1.3 Objectif

Le but principal serait donc de trouver des compromis pour éliminer les contraintes actuelles, il faut donc appliquer une solution permettant principalement de :

- Eviter la manipulation des fichiers de postulation directement sur le partage réseau.
- Améliorer l'accès et la facilité de traitement des données des candidats.
- Améliorer et faciliter le travail de sélection des candidatures valide par les responsables Formation Apprentis EPFL.
- Faciliter le travail de sélection et recrutement des postulants par les formateurs EPFL.

L'idée qui a été retenue pour obtenir ses résultats est de développer une interface de gestion centralisée, permettant aux responsables de la Formation Apprentis EPFL et des formateurs d'effectuer tout leur travail de tri/recrutement.

Pour ce faire, il a fallu avant tout réaliser les points suivants :

- Enregistrer les informations des postulations (sans les fichiers) dans une base de données pour en faciliter l'accès ultérieur.
- Développer une API permettant l'accès à la base de données, faisant le lien entre l'interface et la base de données.

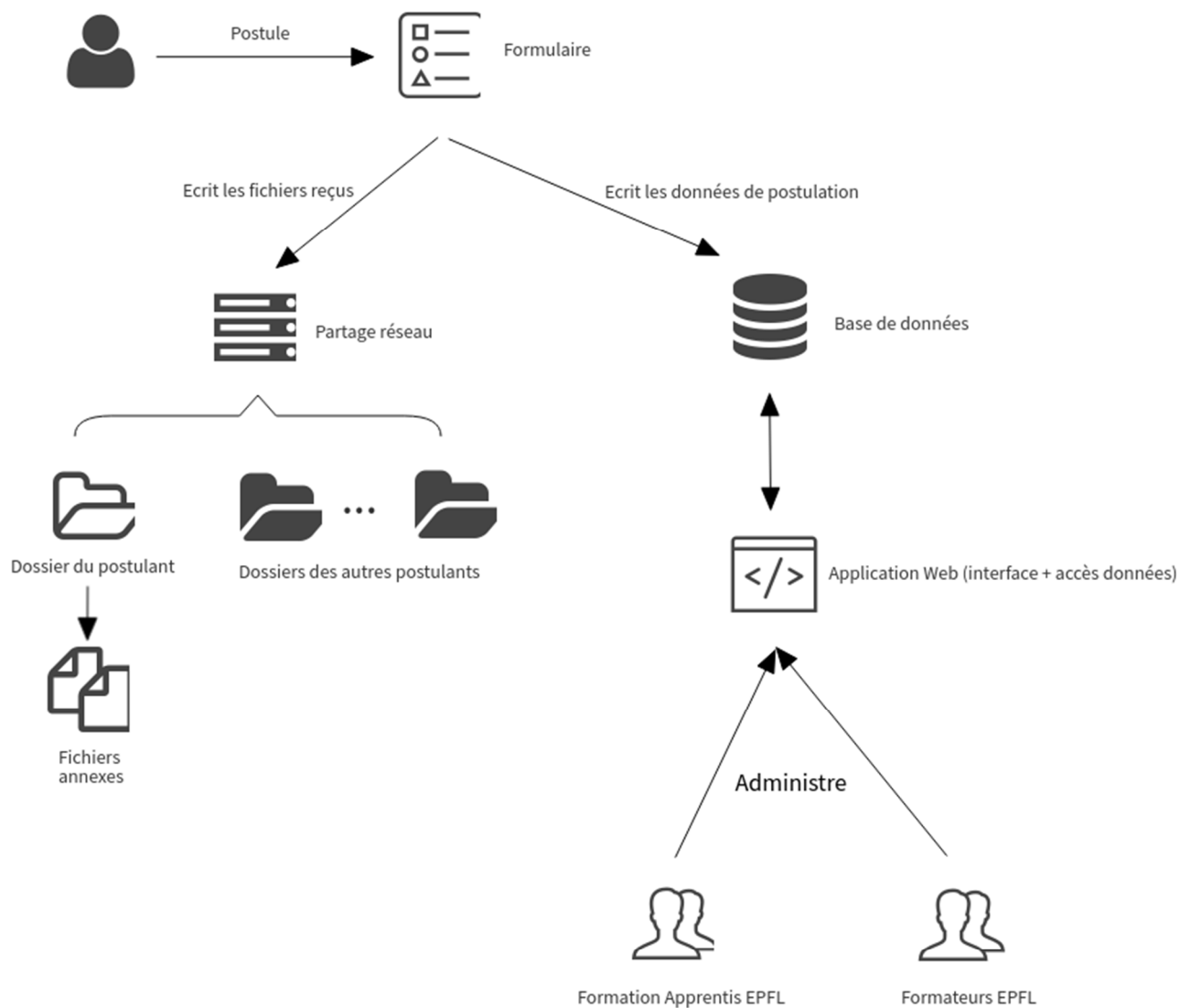
1.4 Travail préalable

Une grande partie du travail a été réalisée au préalable. Premièrement j'ai créé et mis en production une base de données (DB), qui est hébergée sur un serveur MySQL de l'EPFL. Ensuite, le formulaire à évidemment dû être adapté pour pouvoir enregistrer les données en DB, et que le postulant puisse voir, éditer et supprimer ses postulations.

L'API pour l'accès aux données a également été développée en partie pour subvenir aux « besoins » de la future interface. J'ai choisi le *Framework* « *Lumen* » pour son développement.

L'ajout de la base de données à déjà permis de combler la problématique du traitement des fichiers d'information directement sur le partage réseau et d'en simplifier sa structure, puisque que celui-ci ne sera utiliser que pour les fichiers annexes des candidatures et que l'accès direct au serveur ne sera plus utile.

Le diagramme suivant résume la situation :



2 Analyse du projet

2.1 Points importants du cahier des charges

Expliquer les différents points demandés du cahier des charges, fonctionnalités demandées et les étapes de réalisation de chacune.

2.2 Matériel et ressources à disposition

2.2.1 Global

- Un PC sous Windows 10 Education
- Un IDE, Visual Studio Code
- Un client SQL, MySQLWorkbench
- Un repository Git

2.2.2 Dépendances

- PHP version 7.2
- Composer (php packet manager)
 - Ajouter php au « path » windows
- Lumen Framework
- Node.js (v.11)
 - Npm (node.js packet manager)
- @vue/cli
- Vue-router
- Vuex
-

2.3 Faisabilité du projet

Connaissances techniques, charge de travail, temps a dispo etc.

2.4 Planification

2.5 Versionning et sauvegardes

C4science, git

2.6 Technologies & dépendances

2.6.1 Description

Liste
Vue
Lumen

2.6.2 Installation

3 Réalisation

3.1 Modèle DB

3.2 Structure de l'application

3.3 Authentification

3.4 Requêtes API

3.5

4 Conclusion

4.1 Atteinte des objectifs

- Objectifs atteints / non-atteints
- Éléments restants

4.2 Planification

- Comparaison du réel avec la planification initiale

4.3 Problèmes rencontrés

- Points positifs / négatifs
- Difficultés particulières

4.4 Améliorations possibles

Amélioration état actuel (fin TPI)

4.5 Évolution

- Suites possibles pour le projet (évolutions long terme)

5 Annexes

5.1 Sources

<https://nodejs.org/en/>
<https://lumen.laravel.com/>
<https://cli.vuejs.org/guide/>
<https://vuejs.org/>
<https://github.com/axios/axios>
<http://tequila.epfl.ch/>
<https://vuetifyjs.com/en/>

Services :

<https://mockflow.com/>

5.2 Compte-rendu des discussions

Discussions et échanges avec le formateurs et experts

5.3 Journal de travail

6 Glossaire

A

API

application programming interface, interface permettant de fournir des services à d'autres applications.....1, 5, 8

C

Canap-Gest..... 2
 Nom donné au projet, abréviation de "Candidature Apprentissage - Gestion" 2
 compte guest..... 2

J

| | |
|--|---|
| JSON | 4 |
| Format de données dérivé de la syntaxe des object JavaScript | 2 |

R

| | |
|---|---|
| RGPD | |
| Règlement général sur la protection des données | 4 |

Git

Rest

Framework

<https://www.edoeb.admin.ch/edoeb/fr/home/documentation/bases-legales/Datenschutz%20-%20International/DSGVO.html>