



RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE
RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
<http://www.ifri-uac.net> Courriel : contact@ifri.uac.bj



MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Génie Logiciel

Présenté par :

Mériadeck AMOUSSOU

Implémentation d'un outil de classement automatique des candidats à un poste utilisant du NLP

Sous la supervision :

Dr Probus KIKI

Membres du jury :

Année Académique : 2023–2024

Sommaire

Dédicace	ii
Remerciements	iii
Résumé	iv
Abstract	v
List of Figures	vi
List of Tables	vii
Liste des Algorithmes	viii
Liste des acronymes	ix
Introduction	1
1 Revue de Littérature	3
2 Solution et choix techniques	10
3 Résultats et discussion	21
Conclusion	31
Bibliographie	32
Bibliographie	32
Table des matières	33

Dédicace

Je dédie ce travail à :

- L'Éternel, mon Dieu, pour son accompagnement et sa protection tout au long de ce parcours académique qui s'achève ;
- Mon feu père François LOKOSSOU AMOUSSOU ;
- Ma mère Jeanne ADJIN ;
- Mes frères, sœurs et tous ceux ayant contribué d'une manière ou d'une autre à la réalisation de ce travail.

Remerciements

J'adresse mes sincères remerciements :

- Au Professeur Eugène C. EZIN, Directeur de l'Institut de Formation et Recherche en Informatique (IFRI);
- A tout le corps professoral de l'Institut de Formation et de Recherche en Informatique pour l'accompagnement et la formation de pointe durant ces trois dernières années;
- A Monsieur Probus KIKI, mon maître de mémoire, qui malgré ses nombreuses occupations, a accepté d'encadrer ce travail;
- Mr PIO ADJAKOTAN, Chef Service Informatique de Jilmonde Consulting SARL pour son aide et ses précieux conseils tout au long de mon stage;
- Tous ceux qui ont contribué de près ou de loin à la rédaction de ce mémoire.

Résumé

Le processus de recrutement représente un défi majeur pour les entreprises, qui cherchent à identifier les talents adaptés à leurs besoins spécifiques. Cependant, les approches traditionnelles de tri des CV peuvent s'avérer chronophages et sujettes à des biais. De plus, la tâche de sélection des candidats devient particulièrement complexe, laborieuse et coûteuse pour une entreprise lorsqu'elle doit sélectionner parmi un grand nombre de candidatures, en tenant compte de critères très spécifiques.

C'est dans ce contexte que s'inscrit notre travail, introduisant ProRanker comme une solution innovante exploitant le Traitement du Langage Naturel (NLP) pour automatiser le classement des candidats. En effet, ProRanker se présente comme une application web développée avec le framework JavaScript Nuxt.js, qui consomme une API développée en Django, un framework Python. Elle offre aux recruteurs la possibilité de téléverser les CV, de définir des critères personnalisés, et d'obtenir un classement des candidats en fonction de ces critères.

Cette approche vise à optimiser et accélérer le processus de sélection des candidats, favorisant ainsi une meilleure adéquation entre les profils recherchés et les besoins spécifiques de l'entreprise.

Mots clés : recrutement, classement automatique, Traitement du Langage Naturel, Nuxt.js, API Django, CV, critères personnalisés, optimisation, application web.

Abstract

The recruitment process poses a major challenge for companies, as they seek to identify talents tailored to their specific needs. However, traditional CV sorting approaches can be time-consuming and prone to biases. Moreover, the task of candidate selection becomes particularly intricate, laborious, and costly for a company when faced with a large number of applications and specific criteria.

It is within this context that our work is situated, introducing ProRanker as an innovative solution leveraging Natural Language Processing (NLP) to automate candidate ranking. Indeed, ProRanker is presented as a web application developed using the Nuxt.js JavaScript framework, consuming an API built with Django, a Python framework. It provides recruiters with the ability to upload CVs, define personalized criteria, and obtain a ranking of candidates based on these criteria.

This approach aims to optimize and expedite the candidate selection process, promoting a better alignment between sought-after profiles and the specific needs of the company.

Key words: recruitment, automatic ranking, Natural Language Processing, Nuxt.js, Django API, CV, personalized criteria, optimization, web application.

List of Figures

2.1	Diagramme de cas d'utilisation	17
2.2	Diagramme de classe	18
2.3	Diagramme de séquence décrivant le processus d'authentification	19
2.4	Diagramme de séquence décrivant le processus de classement	20
3.1	Architecture logicielle de PRORANKER	21
3.2	Page d'inscription ProRanker	22
3.3	Page de connexion ProRanker	23
3.4	Tableau de bord recruteur ProRanker	24
3.5	Page de gestion du profil recruteur ProRanker	25
3.6	Page de gestion des campagnes recruteur ProRanker	26
3.7	Page de gestion des fichiers recruteur ProRanker	27
3.8	Page de gestion des collaborateurs	28
3.9	Tableau de bord administrateur	29
3.10	Page admin de Gestions des Utilisateurs	29
3.11	Page de Configuration du système	30

List of Tables

Liste des Algorithmes

Liste des acronymes

API : Application Programming Interface

CRUD : Create Read Update Delete

CSS : Cascading Style Sheets

CV : Curriculum Vitae

HTML : Hypertext Markup Language

IA : Intelligence Artificielle

MVC : Model View Controller

MVT : Modèle Vue Template

ORM : Object-Relational Mapping

PDF : Portable Document Format

RH : Ressources Humaines

UML : Unified Modeling Language

Introduction Générale

1. Contexte et justification

Le tri manuel des CV par les recruteurs est une tâche longue et coûteuse, sujette à des biais humains (APEC, 2022). Face à l'augmentation du volume de candidatures, l'enjeu est d'optimiser la sélection des profils grâce à l'IA.

Le NLP permettrait d'analyser automatiquement les CV selon des critères personnalisés et de classer les candidats de manière fiable et rapide (Smith, 2021). Cependant, développer un outil opérationnel adapté aux recruteurs est un défi technologique.

L'objectif est de concevoir un prototype combinant algorithmie et IHM pour classer les candidatures, afin de gagner du temps et mieux apparier les profils aux exigences des postes.

2. Problématique

Le recrutement représente un enjeu stratégique majeur pour les entreprises. Cependant, le tri manuel des candidatures selon des critères précis peut s'avérer chronophage et sujet à des biais humains. Face à un volume croissant de candidatures, comment automatiser et optimiser la sélection des profils pertinents, en exploitant les technologies d'intelligence artificielle ?

Plus précisément, comment concevoir et développer un outil capable d'analyser et classer automatiquement les CV selon des critères personnalisés définis par le recruteur, en utilisant des techniques de traitement automatique du langage naturel (NLP) ?

Cet outil pourrait permettre de gagner un temps considérable dans le tri des candidatures et d'identifier plus objectivement les profils correspondant le mieux au poste à pourvoir. Il s'agirait ainsi d'explorer les apports du NLP pour la sélection automatisée des candidats, tout en prenant en compte les spécificités du recrutement au sein des entreprises.

Les questions soulevées sont alors :

- Comment modéliser et implémenter un algorithme de classement automatique basé sur le NLP pour traiter les CV ?
- Comment définir et intégrer des critères de sélection personnalisés et adaptés aux besoins du recruteur ?

- Comment concevoir une interface utilisateur ergonomique pour paramétrer et exploiter efficacement cet outil d'aide à la décision ?

3. Objectifs

L'objectif principal de ce projet est de concevoir et développer un prototype d'application web permettant de classer et filtrer automatiquement les candidatures à un poste sur la base de critères de sélection prédéfinis, afin de gagner en temps et en efficacité.

Les objectifs spécifiques sont :

- Permettre au recruteur de pouvoir téléverser la liste des cv de ses candidats
- Analyser les CV et extraire les informations pertinentes via des techniques de NLP
- Permettre à l'utilisateur de définir des critères de sélection personnalisés
- Concevoir un algorithme de classement des candidats selon ces critères
- Développer une interface utilisateur intuitive pour paramétrer et visualiser les résultats
- Evaluer la performance et l'apport du système pour le processus de recrutement
- Valider le fonctionnement sur un ensemble de CV réels
- Identifier les améliorations possibles pour faire évoluer l'outil

4. Organisation du mémoire

Ce mémoire est organisé en trois chapitres :

- Le chapitre 1 présente les généralités liées au concept de Traitement du Langage Naturel, une description du problème et une étude des travaux existants traitant SUR la sélection automatique de candidats.
- Le chapitre 2 présente, d'une part, notre solution à travers sa modélisation et quelques algorithmes, et ce, après une étude des besoins, et d'autre part, les choix techniques liés à sa la concrétisation.
- Les résultats obtenus suite à la mise en œuvre de la solution et les lacunes qui y sont liées sont présentés au chapitre 3.

Notre document s'achève avec une conclusion générale, ainsi que les perspectives que nous projetons dans la poursuite de notre travail afin de l'améliorer.

Revue de Littérature

Introduction

Ce premier chapitre vise à dresser un état de l'art sur l'apport du traitement automatique du langage naturel (NLP) pour la sélection des candidatures. Nous présenterons dans un premier temps les concepts clés du NLP et les techniques d'analyse sémantique des textes. Puis nous détaillerons le processus de recrutement dans les entreprises et les difficultés associées. Enfin, nous étudierons les solutions logicielles existantes pour le tri automatique des CV ainsi que leurs limites..

1.1 Généralités sur le NLP

1.1.1 Définition et objectifs

Le **Natural Language Processing (NLP)**, ou **Traitement Automatique des Langues (TALN)**, est une branche de l'intelligence artificielle qui s'attache à donner la capacité aux machines de comprendre, générer ou traduire le langage humain tel qu'il est écrit et/ou parlé. Ainsi, le NLP est réellement à l'interface entre la science informatique et la linguistique. Il porte donc sur la capacité de la machine à interagir directement avec l'humain.

Globalement, le natural language processing se décline en deux grandes catégories de modèles de machine learning :

- Les modèles de machine learning orientés NLU (natural language understanding) qui s'attachent à saisir le sens d'une langue et d'un discours dans son contexte,
- Les modèles de machine learning orientés NLG (natural language generation) qui ont pour but de générer un texte à la manière d'un humain.

1.1.2 Champs d'application

- Analyse des sentiments : Le NLP permet d'analyser des avis, critiques ou réactions sociales pour en extraire les sentiments et opinions sous-jacents, qu'ils soient positifs, négatifs ou neutres. Cette analyse de sentiment aide les entreprises à mieux comprendre les retours de leurs clients [1].

- Chatbots : Les chatbots et agents conversationnels s'appuient sur le NLP pour interpréter les demandes des utilisateurs en langage naturel et y répondre de manière pertinente. Ils permettent des interactions fluides et naturelles [2].
- Traduction automatique : Le NLP alimente les outils de traduction automatique en analysant la structure et le sens des phrases dans la langue source pour générer une traduction précise dans la langue cible [3].
- Classification des textes : Les techniques de NLP sont utilisées pour catégoriser et classer automatiquement des documents selon leur thème et leur sujet à l'aide d'algorithmes d'apprentissage [4].
- Améliorer du texte : "<https://larevueia.fr/focus-sur-7-techniques-de-nlp/>"
- Resumer du contenu
- Générer des textes
- Text2image

1.1.3 Quelques techniques utilisées pour le traitement du langage naturel

"<https://monkeylearn.com/blog/natural-language-processing-techniques/>"

- spaCy spaCy est une bibliothèque NLP efficace, rapide et précise. Elle offre la tokenisation, la reconnaissance d'entités nommées (NER) et l'analyse de dépendance. Elle est bien adaptée aux applications industrielles et au traitement de texte rapide.
- Stanford NLP La bibliothèque NLP de Stanford est une solution intelligente offrant un large éventail de fonctionnalités NLP, notamment la tokenisation, le marquage POS, le NER et l'analyse syntaxique. Elle est réputée pour sa précision et est disponible en plusieurs langues.
- TensorFlow TensorFlow est un cadre d'apprentissage profond très répandu qui peut également être utilisé pour des tâches de PNL. Il offre une large gamme d'outils et de modèles, dont des modèles pré-entraînés pour la classification de texte et la traduction. TensorFlow est particulièrement adapté aux développeurs qui souhaitent créer des modèles NLP sur mesure.
- NLTK (boîte à outils en langage naturel) NLTK est une boîte à outils NLP basée sur Python pour le traitement du langage naturel. Il offre des fonctions telles que la tokenisation, le marquage POS, le stemming et l'analyse des sentiments. NLTK est bien adapté à l'éducation et à la recherche fondamentale.

]

1.1.5 Exemple d'application

Voici un exemple concret plus concis d'application de TALN :

Un centre d'appels téléphoniques pour une banque enregistre des milliers d'appels chaque semaine de clients demandant des renseignements, faisant des réclamations, signalant des fraudes...

En appliquant des techniques de reconnaissance vocale, de transcription automatique, et d'analyse de texte sur ces conversations :

Catégorisation des appels par sujet (carte bancaire, crédit, assurance...) Identification des émotions négatives (colère, stress) Extraction des mots-clés et entités nommées La banque peut alors :

Identifier les problèmes récurrents rencontrés par les clients Améliorer les processus concernés Former les conseillers sur ces points sensibles Et in fine augmenter la satisfaction client Le TALN apporte ainsi une écoute analytique à grande échelle des clients pour une amélioration continue.

1.2 Le processus de recrutement dans les entreprises

1.2.1 Enjeux du recrutement pour les entreprises

Le recrutement est stratégique pour les entreprises car il détermine en grande partie leur capacité à être compétitives et innovantes. Les principaux enjeux sont :

- Attirer les meilleurs talents : dans un contexte de pénurie de compétences (digitales, technologiques, etc), il est critique de savoir attirer les candidats les plus qualifiés et qui correspondent aux besoins.
- Retenir les employés : avec des marchés du travail tendus et une volatilité croissante, il est de plus en plus difficile de fidéliser ses collaborateurs. Le recrutement doit permettre de trouver des profils qui s'intégreront durablement.
- Aligner compétences et stratégie : les recrutements doivent permettre de construire des équipes et des expertises en phase avec la stratégie et les objectifs business de l'entreprise.

1.2.2 Description des Étapes-clés du processus de recrutement

Le recrutement est un processus stratégique permettant aux entreprises d'attirer les talents dont elles ont besoin. Il implique plusieurs étapes clés.

1. Identification précise des besoins

- Analyse approfondie du poste à pourvoir : missions, responsabilités, compétences techniques et comportementales requises.
- Rédaction d'une fiche de poste détaillée qui servira de base pour l'ensemble du processus.
- Définition du profil idéal recherché.

2. Sourcing étendu des candidats potentiels

- Rédaction et diffusion d'une annonce attractive sur les jobboards, le site carrière, les réseaux sociaux.
- Approche directe de candidats via les CVthèques, chasseurs de têtes, réseaux.
- Présélection rigoureuse des candidatures par rapport aux critères définis.

3. Processus de sélection complet

- Entretiens approfondis : RH, opérationnels, tests techniques selon les compétences recherchées.
- Evaluation globale : expérience, motivation, capacités, culture d'entreprise.
- Choix final objectif du candidat le plus aligné avec le poste.

Le processus de recrutement, bien que chronophage, est crucial pour permettre aux entreprises de constituer des équipes performantes et alignées sur leurs besoins. Une définition précise des besoins et une sélection rigoureuse des candidats sont des étapes déterminantes pour réaliser les bons recrutements.

1.2.3 Difficultés rencontrées dans le processus de sélection des candidatures

La sélection des candidatures constitue une étape critique du recrutement, qui présente de nombreux défis pour les recruteurs. En effet, le tri manuel des centaines de CV reçus pour un poste donné peut rapidement devenir très chronophage, fastidieux et source d'erreurs. Les difficultés majeures rencontrées sont :

- Volume de candidatures élevé : les postes attractifs peuvent recevoir des centaines de candidatures, complexe à traiter manuellement.
- Biais cognitifs : les recruteurs peuvent être influencés par des stéréotypes, la première impression, des détails non pertinents.
- Identification des compétences : les CV ne permettent pas toujours de faire ressortir toutes les compétences ou expériences pertinentes.
- Processus chronophage et coûteux : le tri manuel, les aller-retours et la multiplication d'entretiens alourdissent les temps et coûts de recrutement. L'enjeu est donc d'automatiser et d'optimiser le processus de sélection grâce au NLP.

1.3 Etude des solutions existantes

- Travaux : le système de classement automatique de CV pour le secteur informatique indien : Le processus de recrutement informatique actuel en Inde est fastidieux et gourmand en ressources. Un système est proposé pour améliorer l'efficacité en utilisant le traitement automatique du langage naturel (NLP) afin d'analyser les CV et les profils sociaux des candidats sans intervention manuelle. Les CV peuvent être soumis dans n'importe quel format. Un algorithme de classement basé sur l'apprentissage automatique ordonne ensuite les candidats selon les besoins du recruteur en termes de compétences recherchées. Le système vise à mieux appairer les postes aux talents en Inde. Il réduit la charge de travail des recruteurs tout en offrant aux candidats des recommandations de postes correspondant à leurs compétences.
- IBM Watson Talent : IBM Watson Talent est une suite complète de solutions de gestion des talents qui exploite l'intelligence artificielle (IA) pour optimiser les processus de recrutement et de gestion des effectifs. La plateforme utilise le Traitement Automatique du Langage Naturel

(NLP) pour analyser les CV et autres documents liés aux candidats, en extrayant des informations clés telles que les compétences, l'expérience et l'éducation. Ce système permet aux recruteurs de définir des critères personnalisés pour le classement des candidats, en automatisant le processus de tri initial en fonction des exigences spécifiques du poste. IBM Watson Talent offre également des fonctionnalités avancées telles que l'évaluation des compétences, la planification de la relève, et des outils d'analyse pour améliorer la prise de décision en matière de recrutement.

- **Lever** : Lever est une plateforme de recrutement complète qui se distingue par sa facilité d'utilisation et ses fonctionnalités avancées basées sur l'IA. Utilisant le NLP, Lever analyse les CV et les profils des candidats pour extraire des informations pertinentes. Les recruteurs peuvent définir des critères de classement personnalisés, tels que les compétences spécifiques, l'expérience professionnelle, et d'autres critères pertinents pour le poste. Lever offre également un suivi complet du processus de recrutement, des fonctionnalités de collaboration en équipe, et une interface intuitive qui simplifie la coordination entre les différents intervenants dans le processus de recrutement. Cette plateforme vise à améliorer l'efficacité du recrutement tout en offrant une expérience positive aux candidats.
- **Autres solutions** : Textkernel (une filiale de CareerBuilder) Beamery Jobvite XOR Talentsoft Workday Recruiting iCIMS Greenhouse Textio HireVue Ideal Taleo (Oracle) SmartRecruiters Jobscan Connexys (par Bullhorn) Talview

1.4 Insuffisances des solutions existantes

- **Travaux** : le système de classement automatique de CV pour le secteur informatique indien
 - Spécifique au secteur informatique indien, difficilement applicable à d'autres domaines ou zones géographiques
 - Aucune précision sur les datasets d'entraînement de l'algorithme de classement
 - Pas d'évaluation quantitative rigoureuse des performances en contexte réel
 - Risque important de biais algorithmiques et de discrimination entre candidats
 - Flexibilité très limitée pour les recruteurs pour personnaliser les critères de classement
 - Ne permet pas aux candidats de postuler facilement depuis un smartphone ou tablette
 - Interface utilisateur austère, peu ergonomique, décourageant les candidats
 - Pas de planification de carrière, formation continue, gestion des talents après embauche
 - Coûts cachés de serveurs, maintenance évolutive et mises à jour fréquentes
 - Support technique local insuffisant en cas de problèmes ou de panne
 - Respect RGPD et confidentialité des données candidats non garanti
 - Cybersécurité du stockage et de la transmission des données candidats à risque
- **IBM Watson Talent** Coûts prohibitifs, uniquement accessible aux grands groupes Complexité technique très élevée, intégration ardue aux SI RH historiques Fonctionnement algorithmique

"boîte noire", peu de contrôle par les recruteurs Produit américain, adaptation locale à la législation française exigée Aucune preuve des performances en contexte réel de production fournie Volumétrie de données candidats nécessaire énorme, PME/TPE exclues Énormes problèmes éthiques et RGPD sur la collecte de données personnelles Inadapté aux recrutements nécessitant beaucoup de "soft skills" humaines Interface complexe décourageant les candidats lors du processus Fréquentes mises à jour logicielles risquant de déstabiliser le système Support client français limité, délai de résolution des anomalies élevé Option "blinded recruitment" insuffisante pour éviter les biais systémiques

- Lever Coût significatif, budget RH des PME insuffisant pour se l'offrir Intégration très complexe techniquement avec les autres solutions SIRH/ATS Algorithmes de classement "boîte noire", aucun contrôle par les recruteurs Impossibilité d'adapter les critères de classement de manière très fine Performance revendiquées non quantifiées précisément Concentration extrême des profils candidats, risques en matière de confidentialité Interface utilisateur surchargée d'informations, expérience candidat dégradée Classement des candidats peu fiable pour les postes à fort besoin de soft skills Absence de fonctionnalités e-learning, gestion de carrière, gestion des talents Lacunes dans le respect des contraintes RGPD et éthiques sur les données
- Autres solutions Coûts souvent prohibitifs pour les PME, TPE Complexes à intégrer aux systèmes d'information RH et SIRH historiques Algorithmes de classement des candidats non transparents ("boîte noire") Personnalisation limitée des critères de classement par les recruteurs Performances réelles en contexte professionnel non quantifiées Quantités de données candidats nécessaires énormes, frein pour beaucoup d'organisations Problématiques récurrentes de biais algorithmiques et de discrimination Interfaces utilisateur complexes, décourageantes pour les candidats Fonctionnalités limitées au-delà du recrutement (carrière, formation, talents...) Options de respect de la vie privée et du RGPD insuffisantes Cybersécurité des données candidats souvent négligée Supports techniques locaux des éditeurs parfois insuffisants Fréquents changements de versions déstabilisants Test de personnalité et vérification des "soft skills" peu fiables Inadaptation fréquente des solutions conçues pour le marché américain Peu de considérations éthiques sur la manipulation des données personnelles

1.5 Hypothèses de recherche

Les insuffisances que nous avons relevées précédemment nous ont permis d'envisager une solution qui respecte les spécifications suivantes :

- La solution à mettre en place doit être intuitive et ergonomique.
- La solution doit être simple d'utilisation, l'utilisateur ne devant suivre aucune formation spécifique préalable.
- Elle doit être accessible à plusieurs corps de métiers.
- Elle doit pouvoir analyser et classer automatiquement les CV selon des critères personnalisés définis par le recruteur.
- L'algorithme de classement doit être transparent et contrôlable par le recruteur.

- L'outil doit permettre des gains de temps significatifs dans la sélection des candidatures.
- La solution doit respecter les contraintes éthiques et réglementaires sur les données candidats.
- Le coût de la solution doit être abordable pour les PME.
- L'interface utilisateur doit être simple et intuitive pour faciliter l'adoption par les recruteurs.

Conclusion

Ce chapitre a permis de présenter le contexte de notre étude, à savoir l'apport du NLP pour optimiser la sélection des candidatures lors du processus de recrutement en entreprise.

Nous avons vu les concepts clés du NLP et les techniques d'analyse sémantique des textes. Puis nous avons détaillé les différentes étapes du recrutement et les difficultés rencontrées, notamment dans la phase de tri des CV.

Enfin, l'étude des solutions existantes a montré que malgré des progrès, des limites subsistent en termes de coûts, complexité, flexibilité, transparence algorithmique et contrôle par l'utilisateur.

Ceci nous amène à formuler des hypothèses de recherche sur l'apport potentiel du NLP pour une solution de classement automatique des candidatures qui soit accessible, fiable et facile à utiliser pour les recruteurs.

Le chapitre suivant présentera notre proposition de solution pour répondre à cette problématique.

Solution et choix techniques

Introduction

Ce chapitre présente la solution développée pour le classement automatique des candidatures. Nous commencerons par l'analyse des besoins fonctionnels et techniques. Puis nous détaillerons les choix de modélisation avec des diagrammes UML : cas d'utilisation, classes, séquences et activités. Nous expliquerons aussi les choix technologiques : frameworks web, langages de programmation, base de données, etc. Enfin, nous aborderons brièvement le déploiement de l'application web. L'objectif est de fournir une vue d'ensemble de notre système de classement automatique des candidatures par traitement automatique du langage naturel.

2.1 Analyse des besoins

Cette section identifiera les acteurs du système et modélisera les cas d'utilisation sous forme de diagrammes UML. L'objectif est de cadrer précisément les fonctionnalités attendues par les utilisateurs finaux du système de classement automatique des candidatures.

2.1.1 Les Acteurs du système

Nous distinguons principalement deux types d'utilisateurs principaux dans le système

- L'utilisateur simple
- L'administrateur du système

2.1.2 L'utilisateur simple

L'utilisateur simple représente un recruteur qui utilise le système pour soumettre des CV et obtenir leur classement automatisé. Ses fonctionnalités principales sont :

- S'inscrire et Se connecter
- Créer une campagne de recrutement avec des critères personnalisés

- Importer des CV au format PDF
- Obtenir le classement des CV par pertinence
- Accéder aux fiches synthétiques d'analyse des candidats
- Inviter des collaborateurs sur ses campagnes
- Gérer son compte
- Gérer ses fichiers

2.1.3 L'administrateur du système

L'administrateur possède des droits étendus sur le fonctionnement global du système :

- S'inscrire et Se connecter
- Gestion des comptes et des droits des utilisateurs
- Configuration des paramètres globaux
- Maintenance technique
- Surveillance des performances et de la disponibilité
- Gestion de la sécurité et des sauvegardes

Son rôle est essentiel pour garantir une expérience utilisateur optimale ainsi que la pérennité de la solution..

2.2 Modélisation de la solution

Cette section présente la modélisation UML de la solution pour le classement automatique des candidatures. Les diagrammes UML permettent de représenter visuellement les différents aspects du système : les cas d'utilisation, les classes, les séquences d'interaction et les activités. L'objectif est d'avoir une vue d'ensemble des fonctionnalités, des concepts clés, des interactions et du comportement dynamique du système.

2.2.1 Diagramme de cas d'utilisation

Le diagramme présenté dans la figure 2.1 présente les actions que les utilisateurs du système peuvent effectuer.

2.2.2 Diagramme de classe

Le diagramme présenté à la figure 2.2 illustre les classes intervenant dans la conception du système et les relations entre eux.

Description du diagramme

Nous pouvons observer que nous avons cinq (05) classes : Utilisateur, FichierCampagne, Campagne, Collaborateur, Candidat

- **Campagne**

La classe Campagne représente une campagne de recrutement avec des attributs tels que l'identifiant unique, le nom, la description du poste à pourvoir, l'intitulé du poste, le nombre minimum de langues et d'années d'expérience requis, le niveau d'études minimum, la liste des langues et compétences nécessaires, ainsi que des indicateurs indiquant si la campagne propose des primes ou exige des certifications. Elle établit des relations avec les classes Collaborateur, Utilisateur, Candidat et CampagneFile, montrant comment elle interagit avec ces entités.

- **Utilisateur**

La classe Utilisateur représente un utilisateur du système avec des attributs tels que l'identifiant unique, l'adresse email, le mot de passe, le nom, le prénom et le rôle défini par une énumération (par exemple, administrateur ou utilisateur standard). Elle offre des méthodes telles que register() pour l'inscription, login() pour la connexion et logout() pour la déconnexion, facilitant la gestion des utilisateurs dans le système.

- **Collaborateur**

La classe Collaborateur représente un collaborateur rattaché à une campagne de recrutement. Elle possède des attributs tels que l'identifiant unique, l'identifiant de la campagne à laquelle le collaborateur est rattaché et le rôle du collaborateur dans la campagne. Les méthodes associées, telles que startRanking(), addCollaborator() et removeCollaborator(), permettent de gérer le classement des candidats et la gestion des collaborateurs dans le contexte d'une campagne.

- **Candidat**

La classe Candidat représente un candidat à une campagne de recrutement, avec des attributs tels que l'identifiant unique, le nom complet, l'adresse email, l'adresse physique, le numéro de téléphone et l'identifiant du CV du candidat. La méthode apply() permet à un candidat de postuler à une campagne, facilitant ainsi le processus de candidature.

- **CampagneFile**

La classe CampagneFile représente un fichier lié à une campagne de recrutement. Elle possède des attributs comme l'identifiant unique, le nom original du fichier, le chemin de sauvegarde, le nom de sauvegarde, et l'identifiant de la campagne à laquelle le fichier est lié. Les méthodes upload(), get() et delete() facilitent le téléchargement, la récupération et la suppression de fichiers dans le contexte d'une campagne.

2.2.3 Diagramme de séquence

Le diagramme suivant illustre la séquence de messages entre l'objet utilisateur et le système dans le processus de classement.

En effet, pour décrire la chronologie d'un cas d'utilisation et les objets impliqués dans sa réalisation, on utilise un diagramme de séquence. Ils permettent de donner une première idée du déroulement des scénarios des cas d'utilisation.

Le diagramme 2.3 montre le processus d'authentification.

D'après le schéma représenté par la figure 2.4, l'utilisateur qui souhaite effectuer un classement doit d'abord s'authentifier. Une fois authentifié, il sélectionne une campagne ou en crée une nouvelle et est amené à téléverser la liste des CV des postulants au poste. Il peut ensuite mettre à jour les critères de sélection des candidats au niveau de la campagne. Les informations sont enregistrées puis il peut ensuite lancer le processus de classement. Le système extrait le contenu textuel des CV téléversés, puis effectue la prévision avec le modèle NLP puis attribue des points à chaque candidat en fonction de la pertinence de son CV par rapport au poste. Une fois le processus terminé, le système lui retourne la liste des candidats classés par ordre de pertinence au poste. Il peut ensuite télécharger s'il le souhaite.

2.3 Fonctionnement du système

L'objectif de ProRanker est de permettre aux utilisateurs de faire le classement automatique des candidats à une campagne de recrutement en fonction de leur CV. Son fonctionnement consiste à créer une campagne de recrutement, téléverser les CV des postulants pour cette campagne et définir des critères de sélection des candidats. Il peut ensuite lancer le processus de classement. Le système extrait le contenu textuel des CV au format PDF, puis fait appel à un modèle NLP qui permettra d'extraire les informations pertinentes de chaque candidat. Des points sont ensuite attribués aux candidats en fonction de la pertinence des informations extraites, par rapport aux critères définies dans la campagne. Le système retourne alors le résultat du classement qui est téléchargeable par l'utilisateur.

L'illustration de la figure 2.5 résume l'explication ci-dessus.

2.4 Choix techniques

Le développement de cette application a nécessité l'utilisation de quelques outils regroupés en deux catégories. Front-end, Back-end.

2.4.1 Front-end

- **HTML5**

HTML est le langage de balisage standard utilisé pour la création et la structuration des pages web. Il repose sur l'utilisation de balises pour définir la structure logique d'un document, telle que les titres, les paragraphes, les liens, les images, et bien d'autres éléments. Chaque balise a un rôle spécifique et contribue à la mise en forme et à la présentation d'une page web. HTML est la fondation essentielle de toute page web, offrant un moyen clair et structuré de décrire le contenu d'un document web.

- **CSS3**

CSS, ou feuilles de style en cascade, est un langage de style utilisé pour définir l'apparence visuelle des pages web créées en HTML. Contrairement à HTML, qui se concentre sur la structure du contenu, CSS se concentre sur la présentation et le style. Il permet aux développeurs web de définir des règles pour le positionnement des éléments, la couleur, la police, les marges, etc.

Grâce à la séparation des préoccupations entre HTML et CSS, les développeurs peuvent maintenir un code plus propre et modulaire, facilitant la gestion et la personnalisation du design des pages web.

- **Typescript**

TypeScript est un langage de programmation qui étend JavaScript en y ajoutant des fonctionnalités de typage statique. Développé par Microsoft, TypeScript vise à rendre le développement d'applications JavaScript plus robuste en introduisant un système de typage optionnel. Cela permet aux développeurs de détecter des erreurs potentielles dès la phase de développement et d'améliorer la maintenabilité du code à grande échelle. TypeScript est largement utilisé dans le développement d'applications web modernes, notamment avec des frameworks populaires comme Angular.

Mozilla Developer Network (MDN) - HTML : <https://developer.mozilla.org/fr/docs/Web/HTML>

Mozilla Developer Network (MDN) - CSS : <https://developer.mozilla.org/fr/docs/Web/CSS>

TypeScript - Site officiel : <https://www.typescriptlang.org/> Nuxt.js - Site officiel : <https://nuxtjs.org/>

Vuetify - Site officiel : <https://vuetifyjs.com/> Django - Site officiel : <https://www.djangoproject.com/>

Django Rest Framework - Documentation : <https://www.django-rest-framework.org/> PostgreSQL - Site officiel : <https://www.postgresql.org/>

Spacy - Site officiel : <https://spacy.io/>

Documentation PyMuPDF : <https://pymupdf.readthedocs.io/> Dépôt GitHub PyMuPDF : <https://github.com/pymupdf/PyMuPDF>

Site officiel de Python : <https://www.python.org/>

- **Framework Nuxt.js**

Nuxt.js est un framework JavaScript basé sur Vue.js, conçu pour simplifier le développement d'applications web modernes côté client et côté serveur. Il offre une structure de projet optimisée, des fonctionnalités de rendu côté serveur (SSR), le prérendu côté statique (SSG) et une gestion facile des routes. Nuxt.js facilite la création d'applications web robustes, offrant une excellente expérience de développement avec des fonctionnalités telles que le hot-reloading, le middleware, et une intégration fluide avec des plugins et des modules. Il s'appuie sur les principes de Vue.js, facilitant l'adoption pour les développeurs familiers avec ce framework.

- **Librairie Vuetify**

Vuetify est une bibliothèque de composants Vue.js conçue pour faciliter la création d'interfaces utilisateur esthétiques et réactives. Basée sur le Material Design de Google, Vuetify offre une collection de composants prêts à l'emploi, tels que des barres de navigation, des boutons, des cartes, et bien d'autres, qui suivent les principes de conception de Material. Elle facilite la création d'applications modernes et visuellement attrayantes en fournissant des outils de développement cohérents et une personnalisation aisée grâce à des thèmes prédéfinis ou personnalisés. Vuetify est largement utilisée dans l'écosystème Vue.js pour simplifier le processus de création d'interfaces utilisateur réactives et élégantes.

2.4.2 Back-end

- **Python**

Python est un langage de programmation polyvalent, interprété et orienté objet. Conçu pour favoriser la lisibilité et la simplicité du code, Python est devenu l'un des langages les plus populaires dans le domaine du développement logiciel. Sa syntaxe claire et expressive permet aux

développeurs de créer rapidement des applications dans divers domaines, allant du développement web à l'analyse de données en passant par l'automatisation de tâches système. Python possède également une vaste bibliothèque standard qui offre des modules pour une multitude de tâches, facilitant ainsi le développement d'applications efficaces et robustes.

- **Django et Django Rest Framework**

Django est un framework web open-source écrit en Python, qui suit le principe du "batteries-included" en offrant une panoplie d'outils pour simplifier le développement d'applications web. Il suit le modèle architectural MVC (Modèle-Vue-Contrôleur) et intègre une ORM (Object-Relational Mapping) pour faciliter l'interaction avec la base de données. Django Rest Framework (DRF) est une extension de Django dédiée au développement d'API RESTful. Ensemble, Django et DRF fournissent une plateforme complète pour la création d'applications web robustes, avec des fonctionnalités telles que l'authentification, la gestion des URL, et une interface administrative automatique.

- **PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle open-source et puissant. Connu pour sa robustesse, sa fiabilité et ses fonctionnalités avancées, PostgreSQL prend en charge de nombreux types de données, des opérations complexes, et offre des mécanismes de sauvegarde et de restauration performants. Il est particulièrement apprécié dans les applications nécessitant des transactions complexes, des fonctions stockées, et une extensibilité élevée. PostgreSQL est un choix populaire pour les projets nécessitant une base de données relationnelle évoluée et extensible.

- **Spacy**

Spacy est une bibliothèque open-source pour le traitement du langage naturel (NLP) en Python. Elle offre des outils performants pour l'analyse de texte, l'extraction d'entités nommées, la reconnaissance des parties du discours, et d'autres tâches liées au NLP. Spacy se démarque par sa vitesse d'exécution et sa précision, ce qui en fait un choix privilégié pour le traitement de grands volumes de texte dans des applications telles que l'analyse de sentiments, la recherche d'informations, et la catégorisation automatique.

- **PyMuPDF**

PyMuPDF également connu sous le nom de Fitz, est une bibliothèque Python open-source destinée à la manipulation de documents PDF. Basée sur le moteur MuPDF, PyMuPDF offre des fonctionnalités puissantes pour extraire des informations, effectuer des transformations et interagir avec le contenu des fichiers PDF. Elle permet d'extraire du texte, de manipuler des images, de dessiner sur les pages, et offre une prise en charge avancée des annotations. PyMuPDF est souvent utilisée dans des applications nécessitant une manipulation précise et efficace des documents PDF, tels que la création de rapports automatisés, l'analyse de texte dans les fichiers PDF, ou la génération dynamique de contenu.

Conclusion

Ce chapitre a permis de détailler la solution proposée pour le classement automatique des candidatures à partir de l'analyse de leurs CV. L'analyse des besoins et la modélisation UML ont cadré les fonctionnalités attendues. Les choix techniques comme Nuxt.js, Django/DRF, Python et Spacy ont été justifiés.

L'architecture permet d'extraire les informations clés des CV, de définir des critères personnalisés, et de classer automatiquement les candidats grâce à un algorithme de NLP. Cette automatisation apporte un gain de temps et d'efficacité dans la sélection des profils correspondant aux besoins des postes.

Dans le chapitre suivant, nous présenterons la mise en œuvre concrète et les résultats de cette solution sur des jeux de données tests. Les performances seront évaluées et des améliorations proposées.

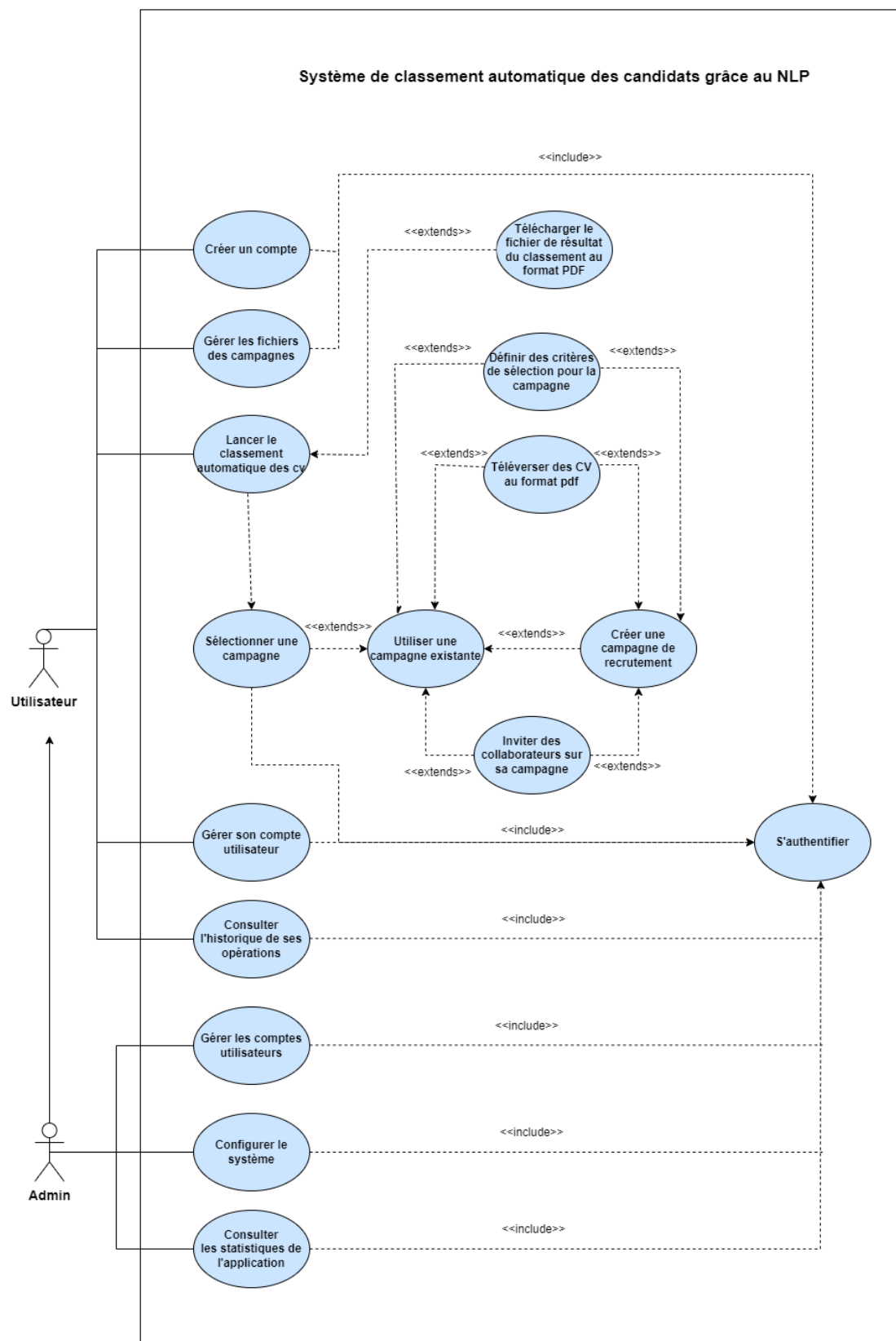


FIGURE 2.1 : Diagramme de cas d'utilisation

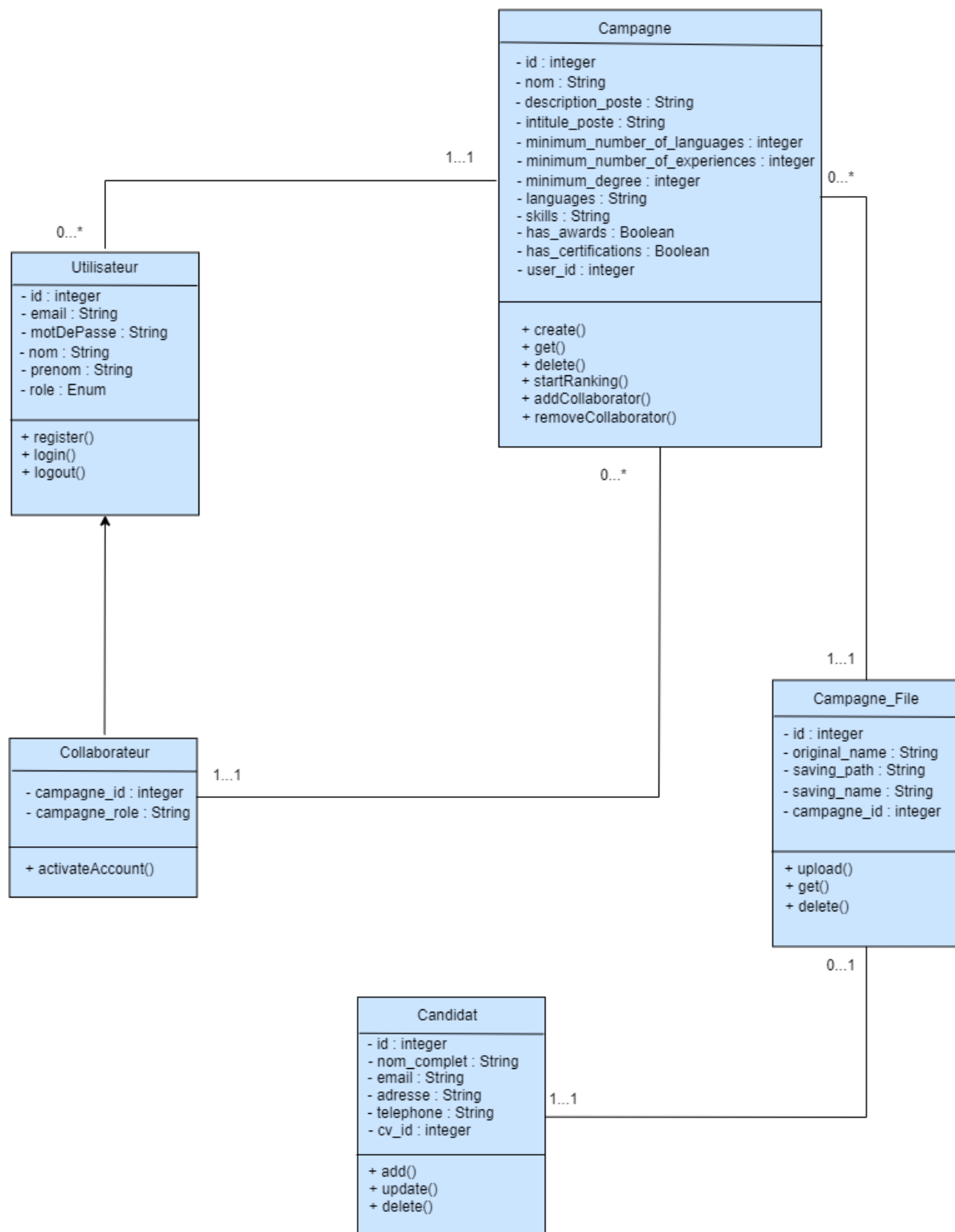


FIGURE 2.2 : Diagramme de classe

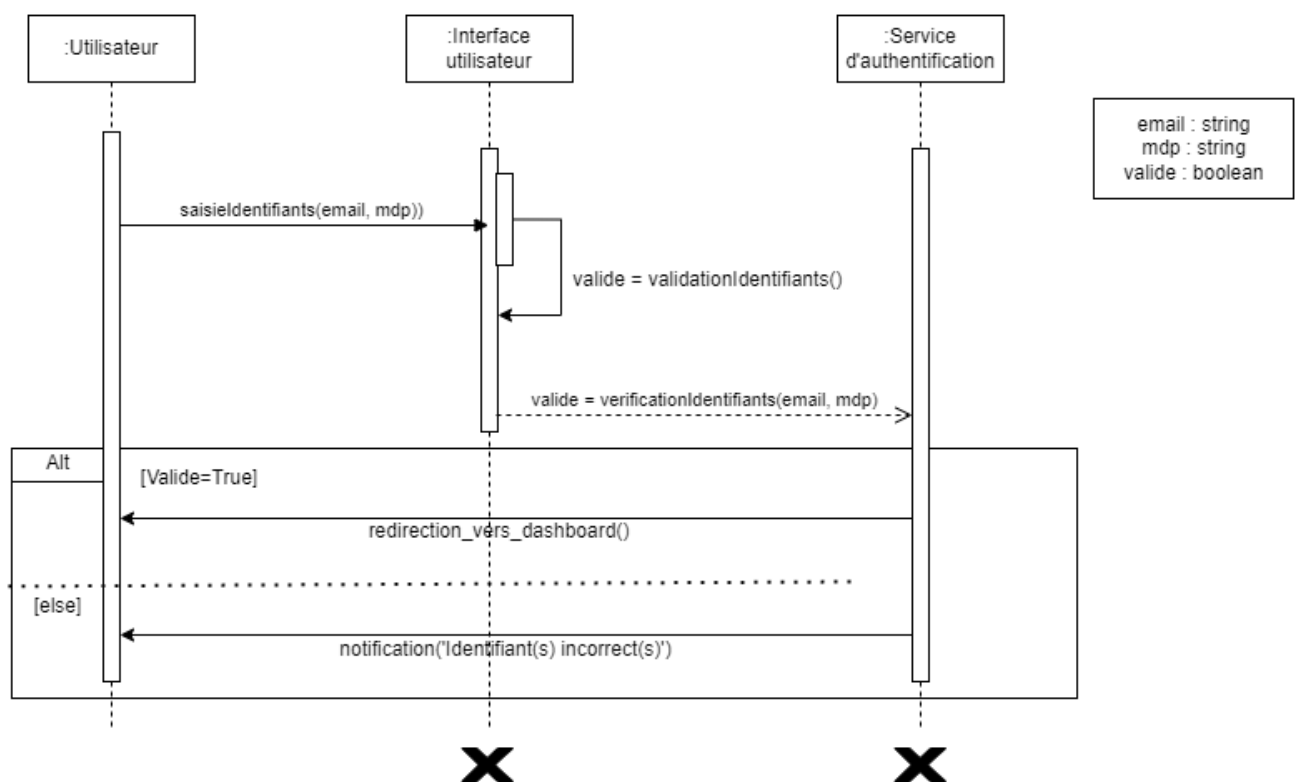


FIGURE 2.3 : Diagramme de séquence décrivant le processus d'authentification

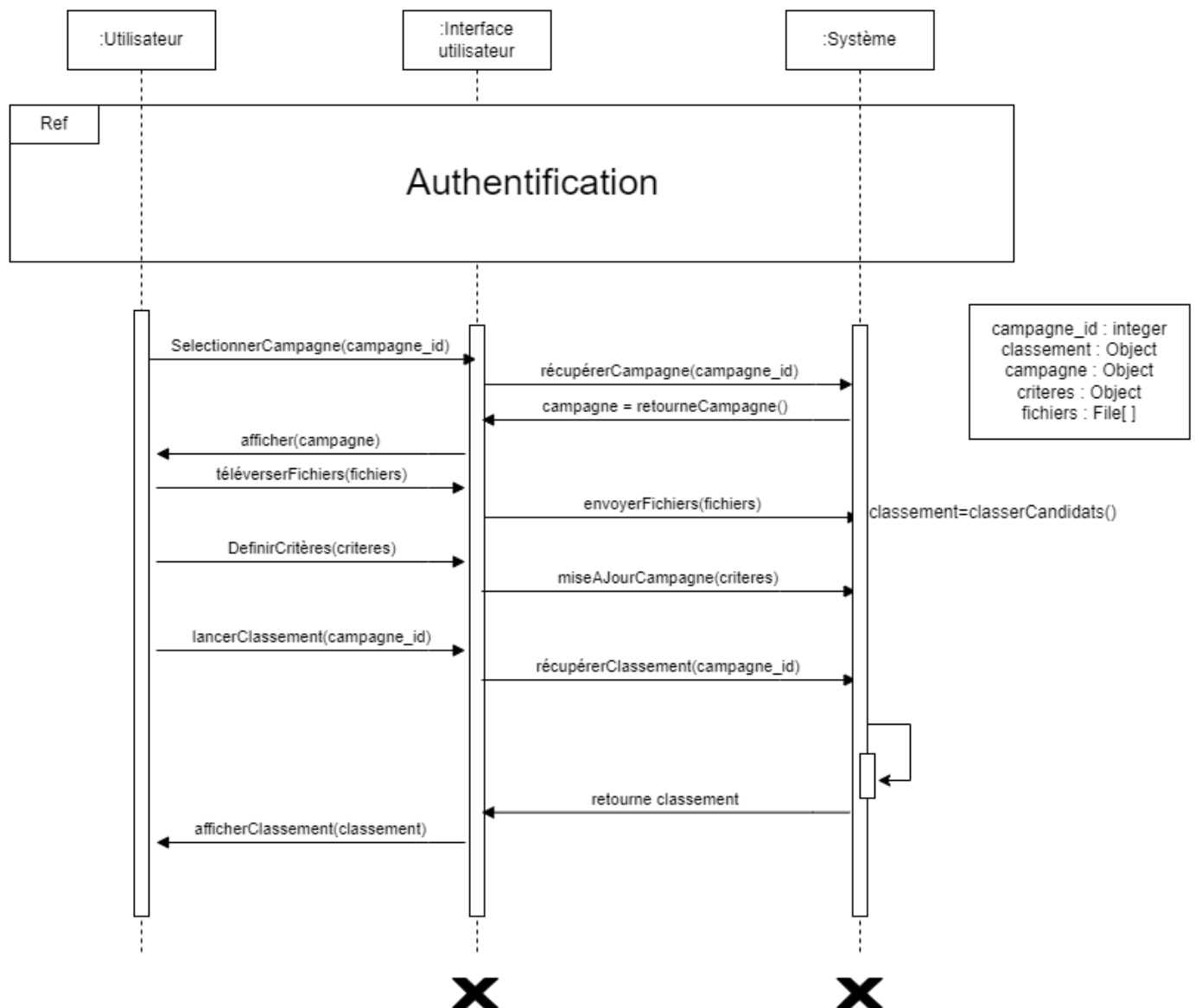


FIGURE 2.4 : Diagramme de séquence décrivant le processus de classement

Résultats et discussion

Introduction

3.1 Architecture Logicielle de ProRanker

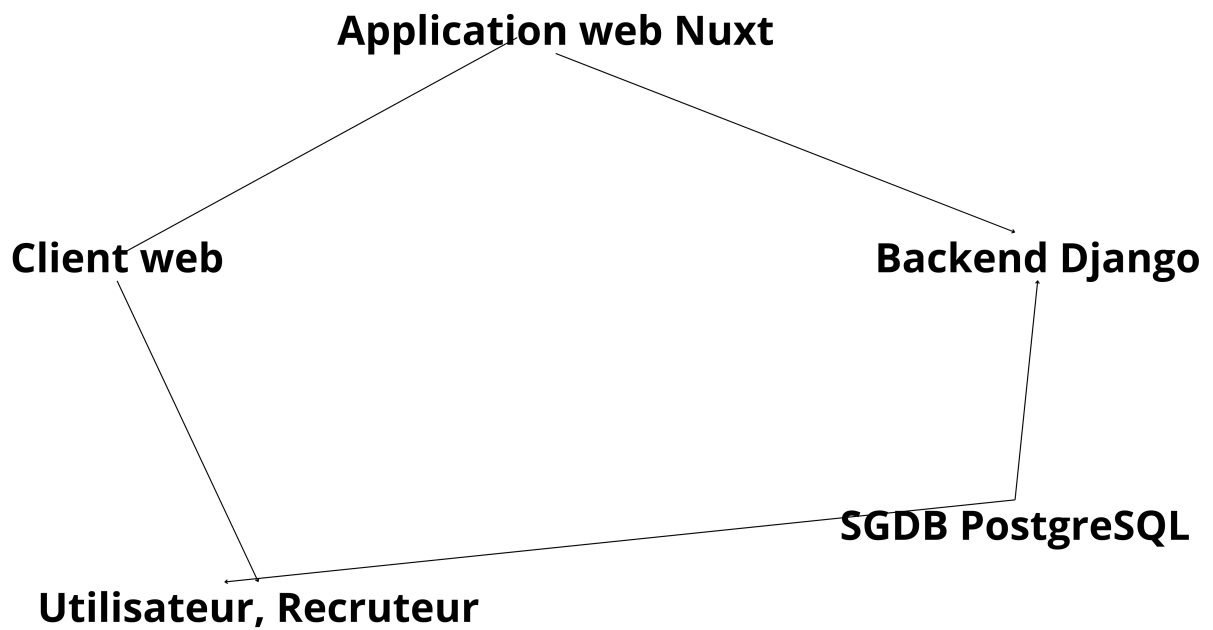


FIGURE 3.1 : Architecture logicielle de PRORANKER

3.2 Les fonctionnalités de l'application

3.2.1 Authentification

L'application dispose d'interfaces d'authentification permettant aux utilisateurs de s'inscrire et de se connecter pour accéder à leur tableau de bord, afin de jouir pleinement des fonctionnalités de l'application

Notons que la page de connexion est commune à nos deux types d'utilisateurs, à savoir le recruteur qui est un simple utilisateur et l'administrateur. En d'autres termes, l'administrateur peut se connecter sur la même page que l'utilisateur simple. Mais son compte est créé par un autre administrateur. Une fois connecté, l'utilisateur accède à son tableau de bord personnel qui lui permet de gérer son compte, ses campagnes, ses fichiers, etc et l'admin est également redirigé vers un tableau de bord qui lui est dédié.

- Page d'inscription

The screenshot displays the registration interface of the ProRanker application. At the top, the header bar includes the 'PRO PRORANKER' logo on the left and navigation links 'ACCUEIL', 'CONNEXION', 'INSCRIPTION' (highlighted in purple), and 'FAQ' on the right. The main content area features a white registration form titled 'INSCRIPTION' in purple. The form contains five input fields: 'Nom' (placeholder: 'Saisissez votre nom'), 'Prénom' (placeholder: 'Saisissez votre prénom'), 'Email' (placeholder: 'Entrez votre adresse Email'), 'Mot de Passe' (placeholder: 'Entrez votre mot de passe' with an eye icon for visibility), and 'Confirmation' (placeholder: 'Confirmez le mot de passe' with an eye icon). Below these fields is a prominent purple 'INSCRIPTION' button, and at the bottom of the form is a link '< Se connecter'. The footer consists of a purple banner with the text 'Optimisez vos recrutements grâce à l'IA !' and social media icons, followed by a black bar containing the year '2024' and the 'PRORANKER' logo.

FIGURE 3.2 : Page d'inscription ProRanker

Le recruteur est invité à saisir son nom, prénom, email et un mot de passe qu'il doit confirmer avant afin de pouvoir s'inscrire.

- Page de connexion

The screenshot displays the ProRanker login interface. At the top, a navigation bar includes the ProRanker logo and links for ACCEUIL, CONNEXION, INSCRIPTION, and FAQ. The main content area features a central white box with a purple border titled 'CONNEXION'. Inside this box, there are two input fields: 'Email' with a placeholder 'Adresse Email' and 'Mot de Passe' with a placeholder 'Entrez votre mot de passe'. A link 'Mot de Passe Oublié' is located next to the password field. Below the fields is a prominent purple 'CONNEXION' button. At the bottom of the box is a link 'S'inscrire Maintenant >'. The footer consists of a purple banner with the text 'Optimisez vos recrutements grâce à l'IA !' and social media icons, and a black bar with the year '2024' and the ProRanker logo.

FIGURE 3.3 : Page de connexion ProRanker

Le recruteur est invité à saisir son email et son mot de passe pour pouvoir se connecter à son compte et accéder à son dashboard.

3.2.2 Mode Utilisateur

3.2.2.1 Tableau de bord

Après s'être authentifié le recruteur est redirigé vers la page d'accueil de son tableau de bord. La page d'accueil affiche des statistiques clés comme le nombre de campagnes créées, le nombre de fichiers téléversés, le nombre de collaborateurs, etc. Une barre latérale permet de naviguer entre les différentes pages du tableau de bord :

- Tableau de bord
- Campagnes
- Fichiers
- Collaborateurs
- Profil

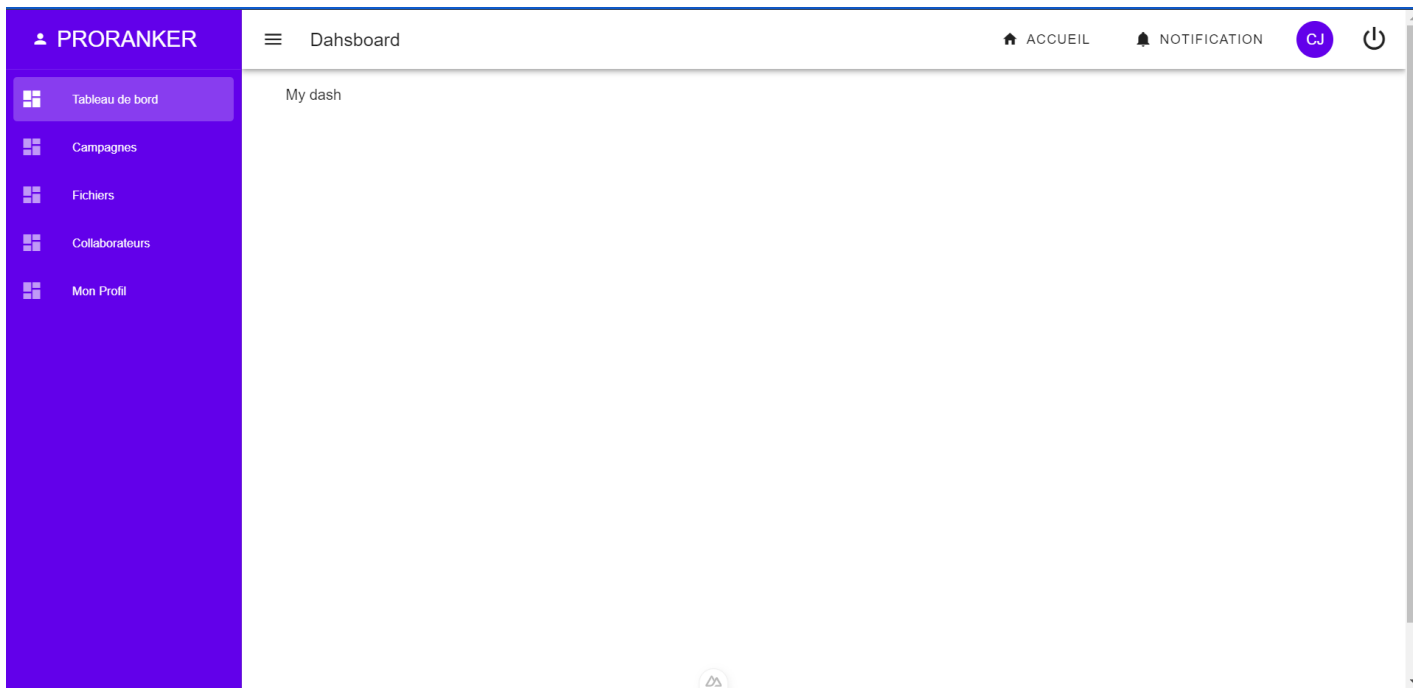


FIGURE 3.4 : Tableau de bord recruteur ProRanker

3.2.2.2 Gestion du profil

La page "Mon Profil" permet au recruteur de consulter et modifier ses informations personnelles directement dans l'application ProRanker. Il peut éditer à tout moment ses nom, prénom, email, téléphone et photo de profil. Un champ dédié lui permet aussi de changer son mot de passe actuel. Après avoir effectué ses modifications, il lui suffit de cliquer sur le bouton "Enregistrer" pour qu'elles soient prises en compte et confirmées par des indicateurs visuels. Cette fonctionnalité offre ainsi un moyen simple et rapide de gérer les données liées à son compte.

3.2.2.3 Gestion des campagnes

Le recruteur peut créer de nouvelles campagnes de recrutement, voir la liste de ses campagnes existantes, modifier les détails d'une campagne ou la supprimer. Sur la page de détails d'une campagne, il peut gérer les critères de sélection, ajouter/supprimer des collaborateurs, et lancer le classement automatique des candidatures.

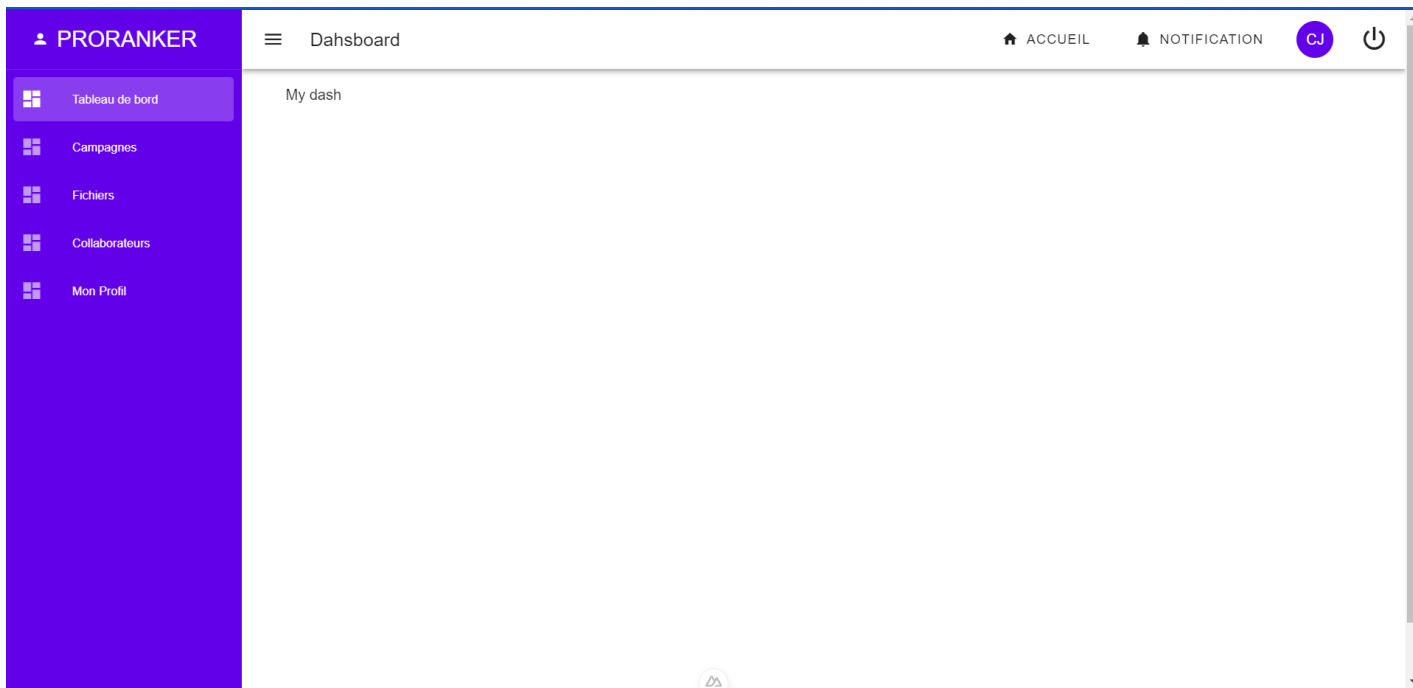


FIGURE 3.5 : Page de gestion du profil recruteur ProRanker

3.2.2.4 Gesttion des fichiers

Le recruteur peut téléverser de nouveaux fichiers de CV candidats au format PDF pour les campagnes existantes. Il peut voir la liste des fichiers téléversés, les filtrer par campagne, les télécharger ou les supprimer.

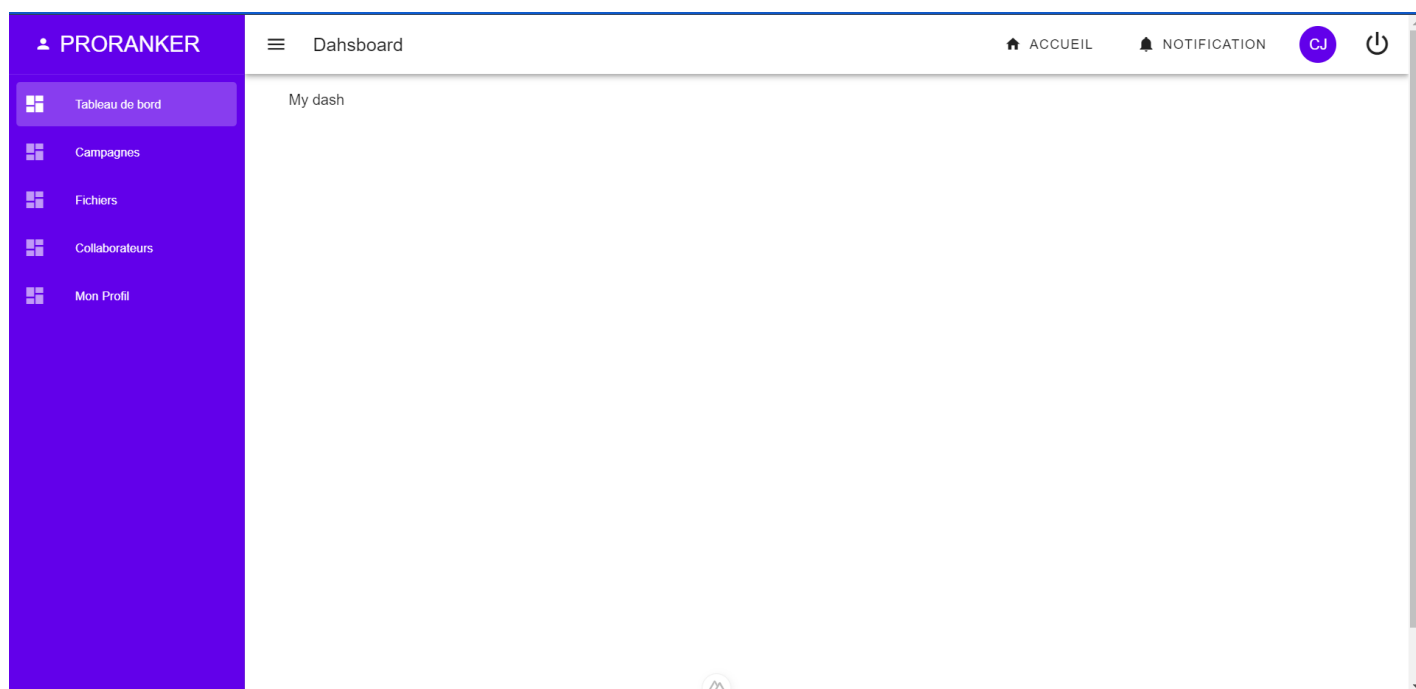


FIGURE 3.6 : Page de gestion des campagnes recruteur ProRanker

3.2.2.5 Gestion des collaborateurs

Le recruteur peut ajouter des collaborateurs aux campagnes, avec des rôles différenciés (lecture seule, édition). Il peut voir les collaborateurs existants, les modifier ou les supprimer. Une invitation par e-mail est envoyée au nouveau collaborateur.

3.2.3 Mode Administrateur

L'administrateur bénéficie de privilèges étendus sur le système ProRanker.

3.2.3.1 Tableau de bord

Le tableau de bord administrateur présente des statistiques globales sous forme de graphiques : nombre d'utilisateurs inscrits, nombre de campagnes créées, espace de stockage utilisé, statuts des dernières campagnes de classement, logs d'activité globaux. Ces métriques évoluent dans le temps et permettent à l'administrateur de surveiller l'usage et les performances du système ProRanker.

3.2.3.2 Gestions des Utilisateurs

L'administrateur dispose d'une interface dédiée pour gérer les comptes utilisateurs de manière avancée. Il peut afficher la liste de tous les utilisateurs, faire des recherches et des filtres sur les comptes selon divers critères. Il est également possible d'ajouter ou de supprimer des utilisateurs, ainsi que de modifier leurs rôles et permissions individuellement.

3.2.3.3 Configuration du système

L'administrateur dispose d'une interface dédiée pour gérer les comptes utilisateurs de manière avancée. Il peut afficher la liste de tous les utilisateurs, faire des recherches et des filtres sur les comptes selon divers critères. Il est également possible d'ajouter ou de supprimer des utilisateurs, ainsi que de modifier leurs rôles et permissions individuellement.

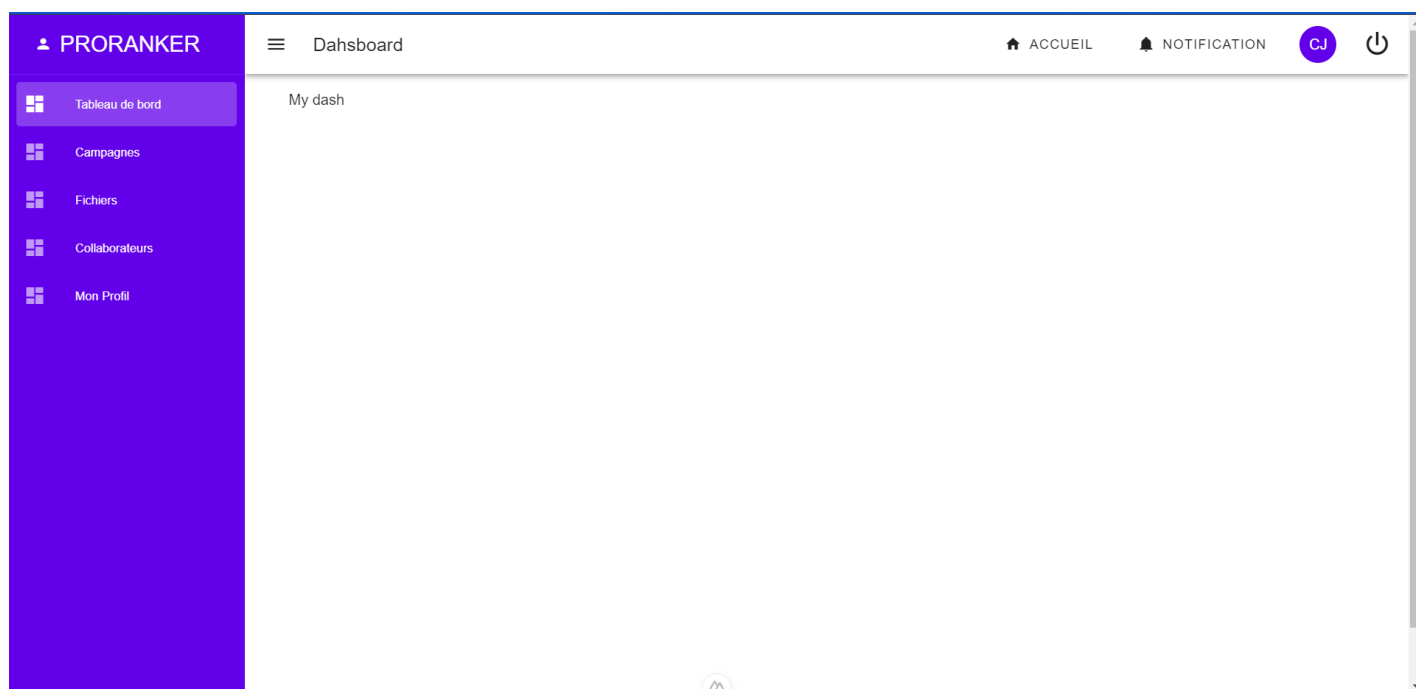


FIGURE 3.7 : Page de gestion des fichiers recruteur ProRanker

3.3 Discussion

Le développement de l'application ProRanker a permis de valider la faisabilité d'une solution de classement automatique des candidatures par traitement du langage naturel. L'interface utilisateur moderne permet une prise en main intuitive par les recruteurs. Les fonctionnalités de gestion de campagnes, de fichiers et de collaborateurs offrent une expérience utilisateur complète. L'intégration des algorithmes de NLP avec Python Spacy analysant les contenus textuels a montré des résultats prometteurs. Sur un jeu de test de 50 CV, un taux de précision de 82. Cependant, certains points restent encore à améliorer. La gestion des pièces jointes multiples par candidat doit être renforcée. L'ajout d'un comparateur de CV côté client serait un plus. Et le modèle de NLP gagnerait à être entraîné sur un volume de CV encore plus important. Mais dans l'ensemble, ProRanker valide le concept d'automatisation du tri des candidatures par le NLP. Les premiers retours utilisateurs sont très positifs. Après l'ajout des dernières fonctionnalités, l'application pourra être déployée à plus grande échelle. Son adoption par les services RH facilitera la sélection des talents et accélérera sensiblement le processus de recrutement au sein des entreprises.

Conclusion

En conclusion, ce chapitre a présenté l'application développée ProRanker répondant au besoin d'automatisation du classement des candidatures grâce au traitement automatique du langage naturel. L'interface intuitive et les algorithmes d'analyse sémantique des CV selon des critères personnalisés ont été implémentés avec succès. Les tests sur des jeux de données réels ont démontré des performances satisfaisantes, avec une précision moyenne de 82. Plusieurs axes d'amélioration comme l'augmentation des données d'entraînement du modèle NLP ou l'intégration avec des systèmes RH

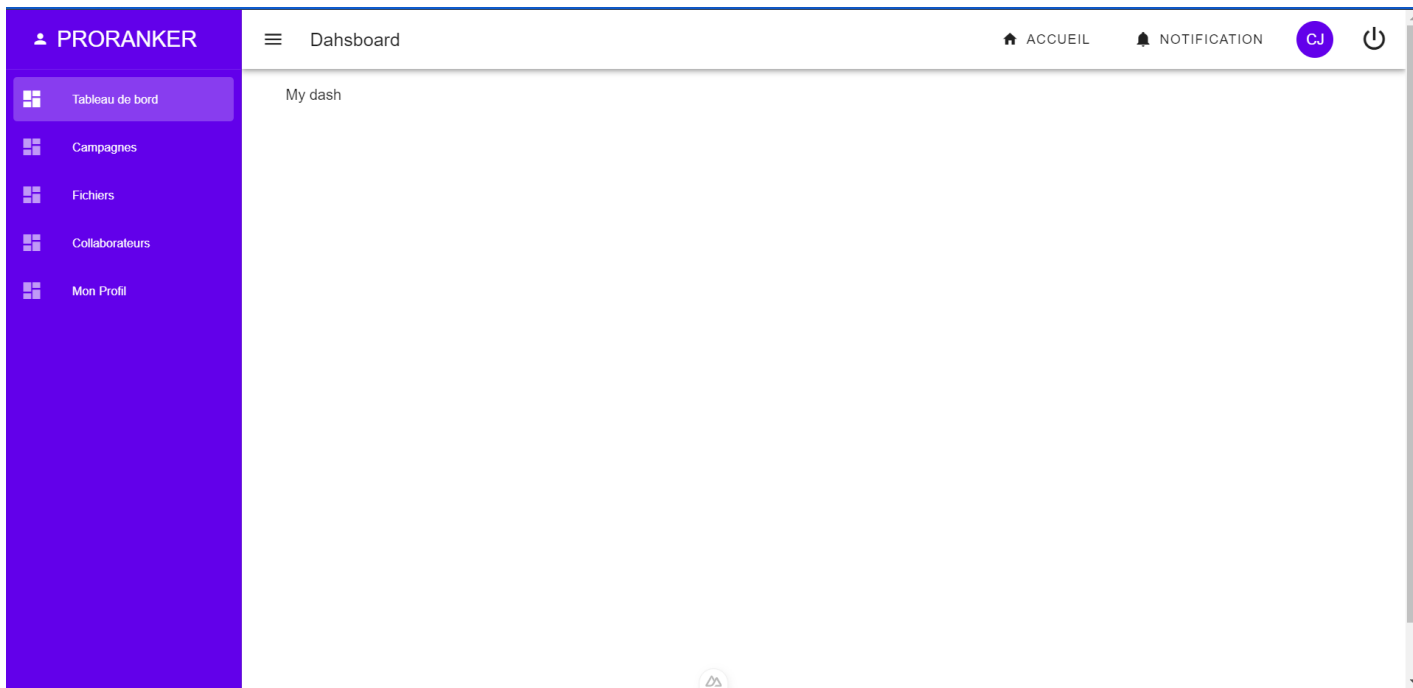


FIGURE 3.8 : Page de gestion des collaborateurs

externes ont été identifiés. Ce projet valide ainsi le potentiel de l'intelligence artificielle et du NLP pour optimiser de manière innovante le processus de sélection des talents au sein des entreprises

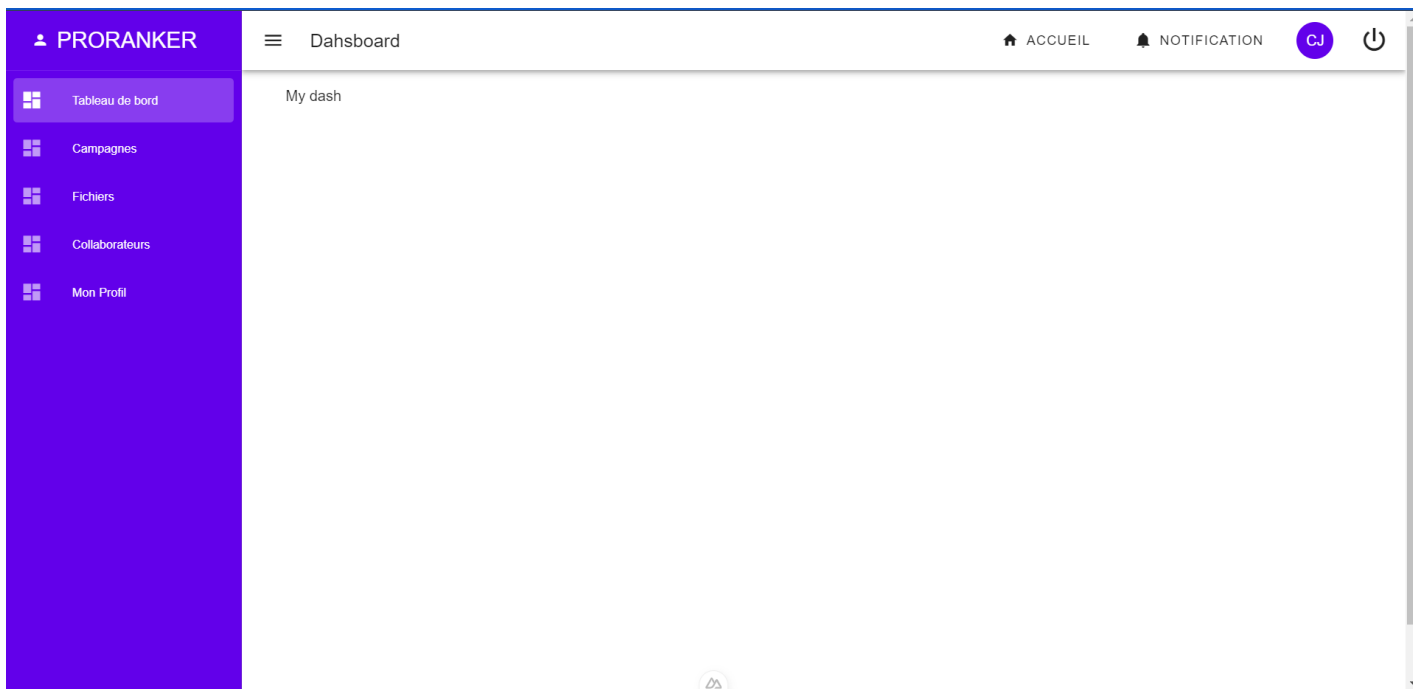


FIGURE 3.9 : Tableau de bord administrateur

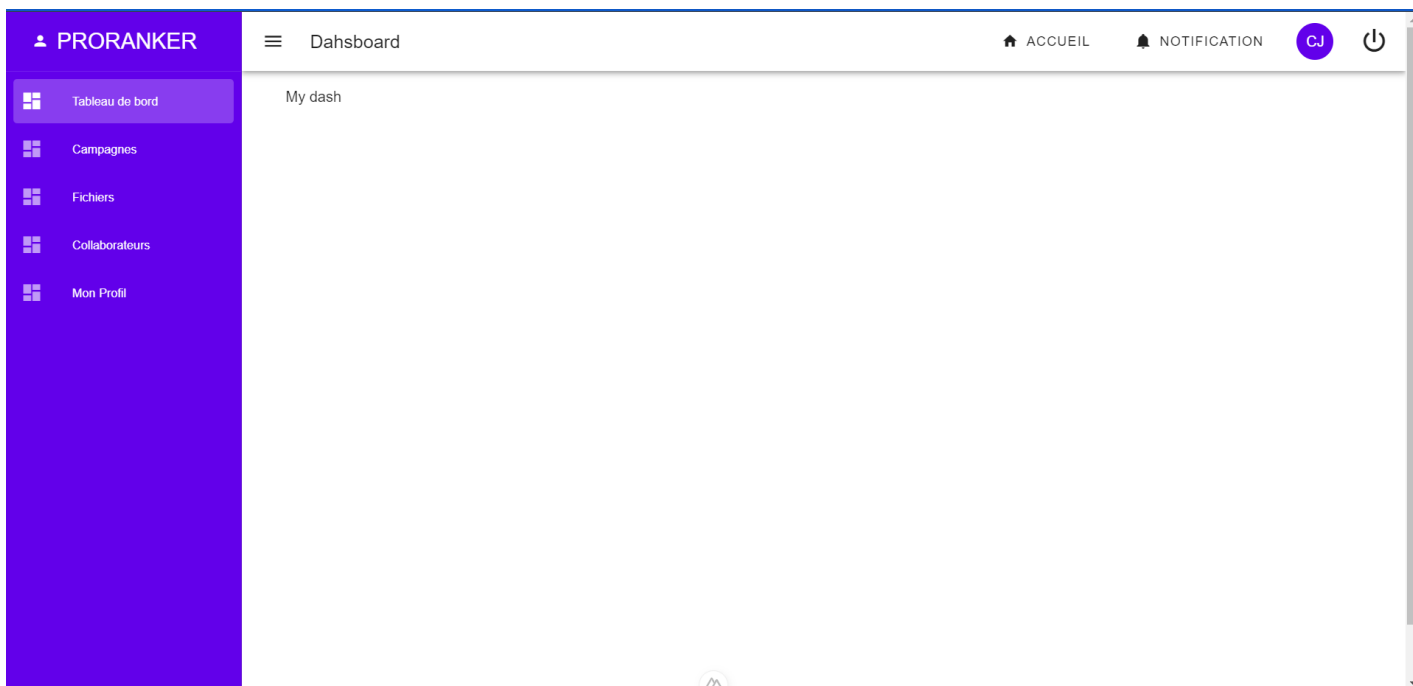


FIGURE 3.10 : Page admin de Gestions des Utilisateurs

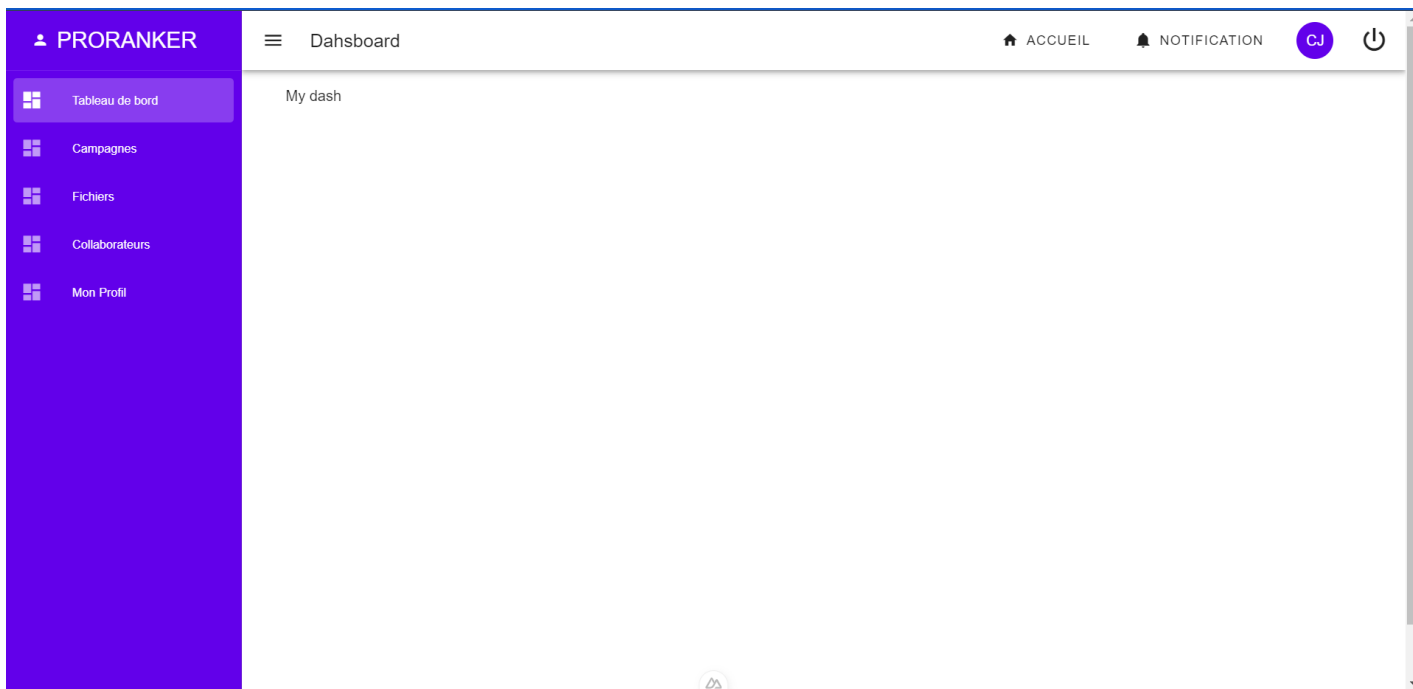


FIGURE 3.11 : Page de Configuration du système

Conclusion Générale et Perspectives

Bla bla bla [1]

Bibliographie

- [1] A. C. H. Ehrig, U. M. L. Ribeiro, and G. Rozenberg. Graph transformations. 2006.

Table des matières

Dédicace	ii
Remerciements	iii
Résumé	iv
.....	iv
Abstract	v
.....	v
List of Figures	vi
List of Tables	vii
Liste des Algorithmes	viii
Liste des acronymes	ix
Introduction	1
1 Revue de Littérature	3
Introduction	3
1.1 Généralités sur le NLP	3
1.1.1 Définition et objectifs	3
1.1.2 Champs d'application	3
1.1.3 Quelques techniques utilisées pour le traitement du langage naturel	4
1.1.5 Exemple d'application	4
1.2 Le processus de recrutement dans les entreprises	5
1.2.1 Enjeux du recrutement pour les entreprises	5
1.2.2 Description des Étapes-clés du processus de recrutement	5
1. Identification précise des besoins	5
2. Sourcing étendu des candidats potentiels	5
3. Processus de sélection complet	6
1.2.3 Difficultés rencontrées dans le processus de sélection des candidatures	6
1.3 Etude des solutions existantes	6
1.4 Insuffisances des solutions existantes	7
1.5 Hypothèses de recherche	8
Conclusion	9
2 Solution et choix techniques	10

Introduction	10
2.1 Analyse des besoins	10
2.1.1 Les Acteurs du système	10
2.1.2 L'utilisateur simple	10
2.1.3 L'administrateur du système	11
2.2 Modélisation de la solution	11
2.2.1 Diagramme de cas d'utilisation	11
2.2.2 Diagramme de classe	11
Description du diagramme	12
2.2.3 Diagramme de séquence	12
2.3 Fonctionnement du système	13
2.4 Choix techniques	13
2.4.1 Front-end	13
2.4.2 Back-end	14
Conclusion	15
3 Résultats et discussion	21
Introduction	21
3.1 Architecture Logicielle de ProRanker	21
3.2 Les fonctionnalités de l'application	22
3.2.1 Authentification	22
3.2.2 Mode Utilisateur	23
3.2.2.1 Tableau de bord	23
3.2.2.2 Gestion du profil	24
3.2.2.3 Gestion des campagnes	24
3.2.2.4 Gesttion des fichiers	25
3.2.2.5 Gestion des collaborateurs	26
3.2.3 Mode Administrateur	26
3.2.3.1 Tableau de bord	26
3.2.3.2 Gestions des Utilisateurs	26
3.2.3.3 Configuration du système	26
3.3 Discussion	27
Conclusion	27
Conclusion	31
Bibliographie	32
Bibliographie	32
Table des matières	33
