

Appréciations et signature de l'encadrant :

Remerciements

Tout d'abord, nous voulons exprimer notre profonde gratitude envers les personnes qui nous ont aidés à la réalisation de ce travail.

En premier lieu, Nous tenons à remercier Mme. **Lobna Jeribi** et Mme. **Chiraz Ben Abdelkader** , notre encadrante et co-encadrante de ce projet de conception et de développement, pour leur patience, pour le temps qu'elles nous ont consacré tout au long de cette période, pour nous avoir dirigé et suivi durant la réalisation de ce travail, leur correction pour notre démarche de stage et leurs précieux conseils.

Nous saisissons cette occasion pour adresser aussi notre gratitude aux membres du jury pour nous avoir honorés en acceptant d'évaluer ce travail et nous espérons qu'ils y trouvent les critères de la bonne rédaction et de motivation auxquels ils aspirent.

Ainsi que tous les enseignants de l'**ENSI** pour leur soutien et pour le savoir qu'ils ont partagé avec nous.

Table des matières

Introduction générale	5
1 Etude préalable	6
Introduction	6
1.1 Etude de l'art	6
1.2 Etude de l'existant	7
1.3 Critique de l'existant	8
1.4 Solution proposée	8
1.5 Modèles et méthodologies adoptés	8
1.5.1 Définition d'un modèle	8
1.5.2 Etude de quelques processus de développement	9
1.5.3 Choix du processus de développement du projet	9
Conclusion	10
2 Analyse et spécification des besoins	11
Introduction	11
2.1 Identification des acteurs	11
2.2 Besoins fonctionnels	11
2.3 Besoins non fonctionnels	12
2.4 Identification des besoins	12
2.4.1 Diagramme de cas d'utilisation	13
2.4.2 Description de quelques scénarii	15
Conclusion	18
3 Conception	19
Introduction	19
3.1 Conception globale	19
3.1.1 Vue logique	19
3.1.2 Vue physique	22
3.2 Conception détaillée	22
Netographie	23

Table des figures

2.1	Diagramme de cas d'utilisation global.	13
2.2	Diagramme de cas d'utilisation 'Gérer les offres'.	14
2.3	Diagramme de cas d'utilisation 'Gérer les CVs recommandés'	14
2.4	Diagramme de cas d'utilisation 'Gérer les offres'	15
2.5	Diagramme de séquence système pour le scénario 'S'authentifier'	16
2.6	Diagramme de séquence système pour le scénario 'Ajouter une nouvelle offre' . .	17
2.7	Diagramme de séquence système pour le scénario 'Postuler à une offre'.	18
3.1	Architecture générale de l'application	20
3.2	Diagramme des Paquets	21
3.3	Diagramme de déploiement	22

Introduction générale

De nos jours, le milieu professionnel impose une concurrence qui ne cesse d'augmenter, et les entreprises, soumises aux exigences du marché, ont besoin de suivre ce dynamisme, protéger leurs intérêts et renforcer la confiance de leurs clients. Ainsi, le choix des profils est désormais prioritaire pour assurer l'efficacité et la rentabilité. Pour atteindre ces objectifs, une grande importance doit être accordée aux responsables et à tout le personnel embauché. Ainsi, les entreprises et face à ces contraintes vont être plus sélectives et plus pointues pendant les recrutements afin de permettre aux entreprises et organisations d'avoir à leur disposition des ressources humaines de qualité, elles essayeront ainsi de garantir que les bons profils occuperont les postes adéquates.

Dans ce contexte marqué par des changements accélérés et par des incertitudes multiples et diverses sur les qualités de la main-d'œuvre, le recrutement des nouveaux employés devient de plus en plus complexe et demande beaucoup de ressources et du temps. Donc les entreprises se trouvent devant deux alternatives, renforcer le département des ressources humaines et avoir ainsi des dépenses supplémentaires, ou bien réduire l'effectif du département ressources humaines, ce qui va minimiser les coûts. Toutefois, les procédures de recrutement vont prendre énormément de temps.

C'est dans ce cadre que s'inscrit notre projet de conception et de développement, qui présente une solution efficace et économique à cette problématique. Ce projet a pour objectif de réaliser une application web qui aide à la prise de décision, en gérant automatiquement les offres d'embauche, de stage et en présentant une présélection des profils parmi les candidats.

Le but de ce rapport est de vous présenter la démarche suivie pour la mise en place de notre application. Celui-ci s'articule autour de quatre chapitres. Le premier chapitre est consacré à une étude préalable où nous définissons quelques notions concernant la procédure de recrutement, nous effectuons une étude de l'existant et nous montrons les carences des solutions existantes ainsi qu'une présentation de la solution proposée. Le second chapitre présente une analyse et une spécification des besoins fonctionnels et non fonctionnels auxquels il doit répondre. Le troisième chapitre se focalise sur l'architecture du système ainsi que sa conception détaillée. Le quatrième chapitre comporte une description des outils et des technologies employés dans le développement du projet et une illustration concise de la réalisation du système. Finalement nous clôturons ce rapport par une conclusion générale qui synthétise le travail élaboré et qui ouvre des perspectives des travaux futurs.

Chapitre 1

Etude préalable

L'étude préalable est une phase cruciale pour la bonne compréhension des exigences du marché, la définition concise des objectifs du système à mettre en place. Dans la première section, nous définissons quelques notions concernant le processus de recrutement. Ensuite nous analysons quelques solutions existantes sur le marché en discutant leurs avantages et leurs inconvénients. Enfin, nous étudions quelques processus de développement et nous fixons le modèle à suivre dans notre projet.

1.1 Etude de l'art

Dans cette section, Nous présentons les terminologies utilisées tout au long du processus de recrutement.

Recrutement : Le recrutement est une étape fondamentale et incontournable pour chaque organisme professionnel. C'est le processus qui consiste à choisir entre divers candidats en vue d'un poste de travail qu'on a jugé utile de créer, de maintenir ou de transformer. A priori, un recrutement comporte quatre étapes [1] :

- La perception d'un besoin susceptible d'ouvrir à candidature externe.
- La prospection (par annonces, réseaux ou appel à organismes spécialisés).
- La sélection (par CV, lettres de motivation, essais professionnels, tests et entretiens).
- La décision d'embauche, l'accueil et l'intégration.

Département Ressources Humaines : La majorité des entreprises possède un département qui se charge de la gestion des ressources humaines : C'est un ensemble de fonctions et de mesures ayant pour objectif de mobiliser et développer les ressources du personnel pour une plus grande efficacité, au profit de la stratégie d'une organisation[2].

Offre d'emploi : Une offre d'emploi est composée d'un descriptif présentant l'employeur, l'intitulé du poste, la définition du poste, la définition du profil recherché, une référence et une adresse de réponse.

Candidat : Un candidat est une personne qui sollicite, pour elle-même, une place, un poste, une mission, un mandat, un titre, une distinction ou un prix, éventuellement par la réussite à un examen, un concours ou une élection [4]. Un candidat est celui qui postule à :

- Un titre.
- Un emploi.

CV (Curriculum Vitæ) : Le CV récapitule le parcours professionnel d'une personne. C'est une clé essentielle pour décrocher un entretien d'embauche. Il doit de préférence être accompagné d'une lettre de motivation (même si cette pratique a tendance à se perdre de nos jours avec l'explosion des e-candidatures et du web). Il peut être présenté de façon chronologique ou mieux encore, antéchronologique (en partant de l'expérience la plus récente vers la plus ancienne)[5].

1.2 Etude de l'existant

Le recrutement, comme nous l'avons mentionné, est le processus qui consiste à choisir entre divers candidats en vue d'un poste de travail à travers la sélection des CVs les plus pertinents et adéquats pour le poste mis en question.

Cette procédure est à la fois lente et ennuyeuse, précisément pour le responsable des ressources humaines qui sera dans l'obligation de lire tous les CVs reçus et de les étudier cas par cas. C'est à partir de là qu'il a apparu le besoin de développer ce secteur et d'aider les responsables des ressources humaines dans la sélection des candidats afin d'assurer que les bons profils occuperont les postes qui leur sont convenables.

Plusieurs solutions ont été proposées pour développer ce secteur et faciliter les tâches des départements de Ressources Humaines. Les solutions les plus courantes à l'échelle nationale et les plus utilisées sont des applications web comme **TanitJobs.tn**, **Khedma.tn**, **Keejob.com**, **Manpower-time.com**, **ExcelJob.com**, **emploi-en-tunisie.com**, etc. Ce sont des plateformes de recrutement en ligne qui aident les candidats à trouver des offres et à y postuler. Elles proposent les fonctionnalités suivantes :

- Publication des offres : A travers ce site une entreprise peut publier leurs offres de stage et d'embauche.
- Postuler aux offres : Les diplômés et les étudiants à la recherche d'un stage ont la possibilité de postuler et de déposer leurs CVs à travers ce site.
- Téléchargement des modèles de CVs : Les candidats peuvent rédiger leur CVs en ligne ou télécharger des template de CV gratuitement.

1.3 Critique de l'existant

Toutes les solutions existantes déjà, présentent certaines limites liées à l'absence de plusieurs options à ces plateformes dont la fonctionnalité est assez importantes pour la bonne gestion des ressources humaines :

- Les recruteurs se trouvent encore dans l'obligation de consulter tous les CVs et de traiter toutes les candidatures qui est un travail gênant et très lent.
- Absence d'un module de sélection des offres par critères (par durée, par nature, par type, par localisation, etc).
- Ces sites ne présentent aucune garantie quant à la fiabilité des données saisies par le candidat (Sécurité et confidentialité).

Par ailleurs, on constate que la solution actuelle ne permet pas de résoudre la majorité des problèmes et la procédure de recrutement reste toujours incomplète et non efficace. De ce fait, notre application tente d'être parmi les premières plateformes qui intègrent ces différentes fonctionnalités citées ci-dessus et confronte les inconvénients des solutions existantes.

1.4 Solution proposée

L'étude de l'existant nous a permis de dégager plusieurs anomalies que nous avons détaillés dans la section précédente. Pour faire recours à ces anomalies nous proposons de développer une application Web, étant un apport de la solution proposée mais qui tente de résoudre ces problèmes en favorisant l'automatisation de plusieurs étapes de la procédure de recrutement, elle offrira en effet un outil de recommandation et d'aide à la prise de décision qui facilitera la sélection et l'étude des candidatures à travers l'analyse de leur CVs, tout en garantissant une efficacité et une optimisation par rapport aux solutions proposées et aux résultats données par les procédures habituelles.

1.5 Modèles et méthodologies adoptés

Dans cette section, nous présentons d'abord une définition générale d'un modèle puis nous étudions quelques processus de développement pour en choisir le modèle le plus adéquat à notre projet.

1.5.1 Définition d'un modèle

Un modèle de développement logiciel désigne toutes les étapes du développement, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement. L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé

qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

1.5.2 Etude de quelques processus de développement

Dans cette section nous expliquons le modèle en cascades, le modèle de prototypage et la méthodologie Scrum.

1.5.2.1 Le Modèle en cascade

Le modèle en cascade considère le développement logiciel comme une succession d'étapes réalisées de façon linéaire. C'est un modèle dirigé par les documents vu que chaque activité est validée par un document approprié. Dans le modèle en cascade, les retours en arrière se limitent à un retour sur les étapes immédiatement antérieures.

1.5.2.2 Le Modèle de prototypage

C'est une approche de développement logiciel qui privilégie la préparation d'une version du produit utilisée comme base d'évaluation et de préparation à la version complète.

Le modèle vise un raffinement des spécifications et des performances par des prototypes successifs.

1.5.2.3 La Méthodologie Scrum

Scrum est un cadre de développement logiciel agile, itératif et incrémental pour la gestion du développement des produits.

La méthodologie vise la satisfaction du client en lui livrant régulièrement des versions fonctionnelles du système.

1.5.3 Choix du processus de développement du projet

Nous avons estimé que le modèle en cascade est le plus approprié dans le contexte du projet de conception et développement pour les raisons suivantes :

- Le modèle en cascade est un modèle dirigé par les documents. Une documentation est produite à chaque étape de l'élaboration du modèle ce qui facilite la validation et la vérification à l'issue de chaque phase et assure un enchainement équilibré entre travail technique et documentation.
- Les spécifications des systèmes de gestion sont généralement bien définis et peu susceptibles d'évoluer au cours du projet.

Conclusion

Dans ce premier chapitre, nous avons défini le champ de notre étude suivi d'une étude de l'existant afin de préciser nos objectifs à atteindre. En effet, l'étude de l'existant nous a permis de préparer une bonne conception pour les améliorations que nous allons ajouter dans la solution proposée afin de répondre à nos besoins et aux insuffisances présentes. Dans la troisième section, nous avons présenté quelques processus de développement et nous avons pu identifier après une longue étude le modèle le plus adéquat à notre projet.

Dans le chapitre qui suit nous présenterons les démarches de développement et de conception de notre solution.

Chapitre 2

Analyse et spécification des besoins

Les phases d'expression et d'analyse du besoin permettent de décrire les fonctionnalités du logiciel et les contraintes sous lesquelles celui-ci doit être réalisé afin de respecter les critères fixés par le cahier des charges et de garantir l'efficacité et la réussite de notre système.

Dans la première section, nous aborderons la phase d'analyse, dans laquelle nous spécifierons les acteurs et leurs besoins fonctionnels et non fonctionnels de l'application à développer. Dans la deuxième section, nous procédons à une description détaillée des fonctionnalités assurées par la solution moyennant les représentations semi-formelles d'UML.

2.1 Identification des acteurs

Les acteurs sont des entités externes qui interagissent avec le système et ils sont décrits par leur rôle. Ce dernier définit les besoins et les capacités de l'acteur vis-à-vis au système. Les acteurs en interaction avec notre système sont :

- **Le recruteur** : Le recruteur est un acteur principal de notre système qui représente son entreprise, il gère ses offres, demande les résultats correspondants et consulte les profils recommandés ainsi qu'elle envoie des mails d'entretien en cas d'acceptation de candidature.
- **Le candidat** : Le candidat est un acteur principal de notre système, c'est lui qui va consulter les offres publiées sur l'application et postuler pour une offre bien déterminée.

2.2 Besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement du système. Ce dernier à concevoir doit permettre à l'utilisateur d'effectuer les opérations déjà mentionnées.

Notre application offrira au candidat les fonctionnalités suivantes :

- Gestion des offres : Le recruteur peut ajouter, modifier, supprimer et afficher ses offres.
- Gestion de profil : Le recruteur peut consulter et modifier le profil de son entreprise.
- Gestion de communication avec les candidats : Le recruteur peut rédiger et envoyer un mail à un candidat recommandé.
- Gestion des CVs recommandés : Le recruteur peut consulter les CVs classés par le système par ordre de pertinence ou les supprimer.

Elle offrira au candidat les fonctionnalités suivantes :

- Gestion des offres : Le candidat peut consulter et postuler pour une offre souhaitée.
- Gestion de profil : Le candidat peut ajouter et modifier son CV ainsi que son profil.
- Recherche automatique des offres par mots clés : Le candidat peut trouver des offres d'emploi qu'il désire en tapant les mots clés de ces dernières dans la barre de recherche.

2.3 Besoins non fonctionnels

Les besoins non fonctionnels sont les exigences qui caractérisent notre système. Ils se résument dans les points suivants :

- Performance : Le système doit être performant. Il doit répondre aux fonctionnalités de base avec un temps de chargement rapide des données.
- Ergonomie : Les interfaces doivent être simples et faciles à manipuler.
- Sécurité : L'application doit offrir aux utilisateurs un accès personnalisé et une connexion sécurisée.
- Disponibilité : Le système doit être accessible à tout moment.
- Fiabilité : Le système doit avoir le minimum d'erreur.

2.4 Identification des besoins

Pour mieux présenter l'architecture de notre application, on va choisir le langage de modélisation le plus adopté UML(Unified Modeling Language) vu son caractère polyvalent et sa souplesse en font un langage universel. Ainsi il facilite la compréhension de représentation abstraite et complexe. Cette sous-section comporte les diagrammes de cas d'utilisation et de séquence de notre application.

2.4.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation offre une vision globale du comportement du système de point de vue fonctionnel. Chaque cas d'utilisation représente une interaction possible entre l'utilisateur et le système.

2.4.1.1 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation global est le diagramme général du système qu'on modélise et qui décrit les interactions, les fonctionnalités et les relations globales entre les différents acteurs.

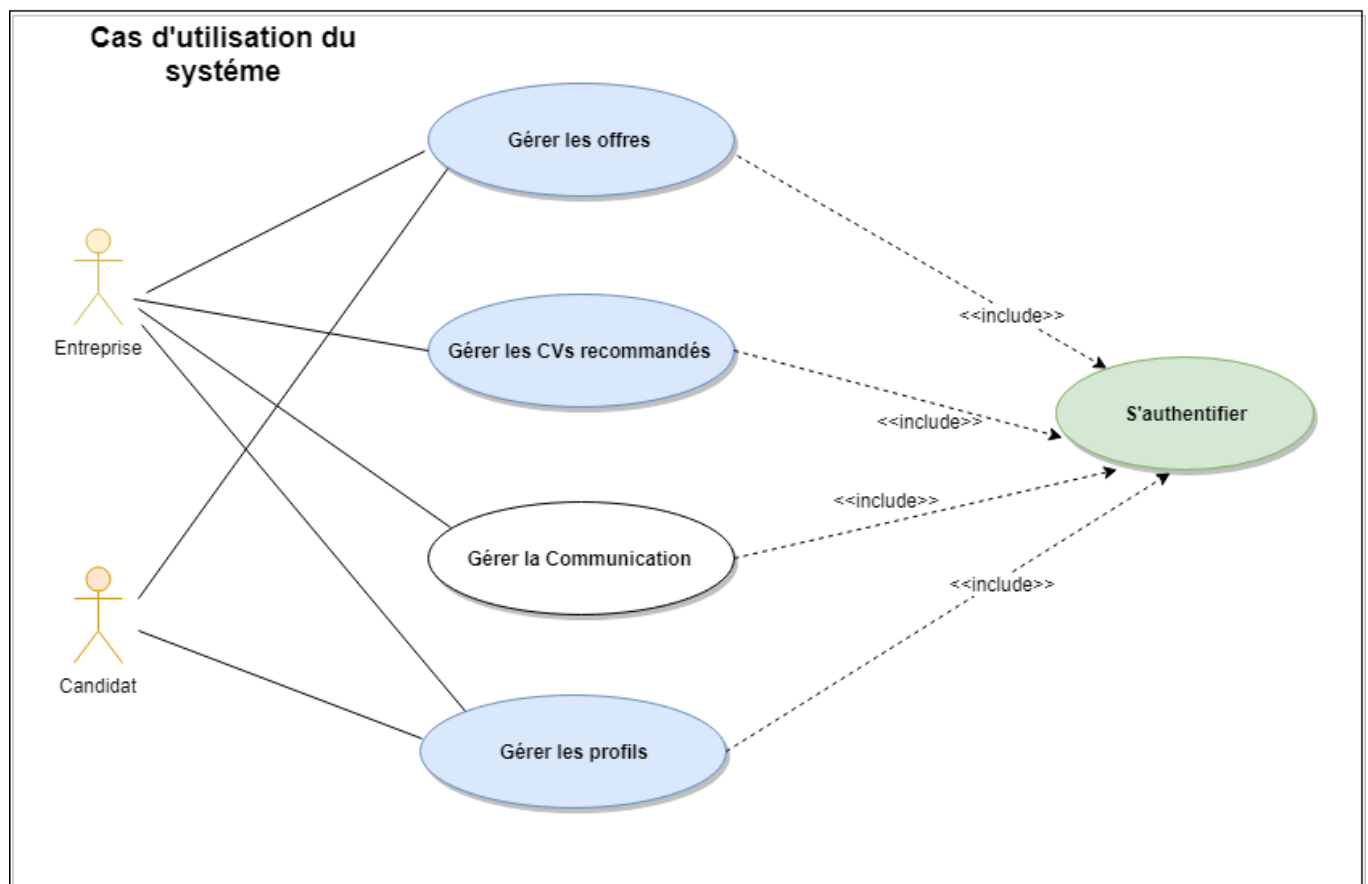


FIGURE 2.1 – Diagramme de cas d'utilisation global.

En ce qui concerne le candidat, après l'authentification, il peut modifier son profil, consulter les offres publiées et postuler ainsi que modifier son CV sur son profil.

Pour le côté Entreprise, le recruteur, après l'authentification, peut ajouter des offres ou modifier des offres déjà publiées. Ensuite, le système filtre les CVs des candidats selon des critères de sélection bien déterminés et affiche les CVs triés par score au recruteur. Ce dernier peut voir le classement des CVs recommandés, télécharger les CVs de candidats ainsi que les supprimer. De plus, il peut lui envoyer un mail d'entretien en cas d'acceptation de candidature.

2.4.1.2 Diagramme de cas d'utilisation côté Entreprise 'Gérer les offres'

Le cas d'utilisation 'Gérer les offres' peut être détaillé comme indique la figure 2.2. En effet la fonctionnalité 'Gestion des offres' permet à l'entreprise d'ajouter, de supprimer, de mettre à jour et afficher des offres d'emploi .

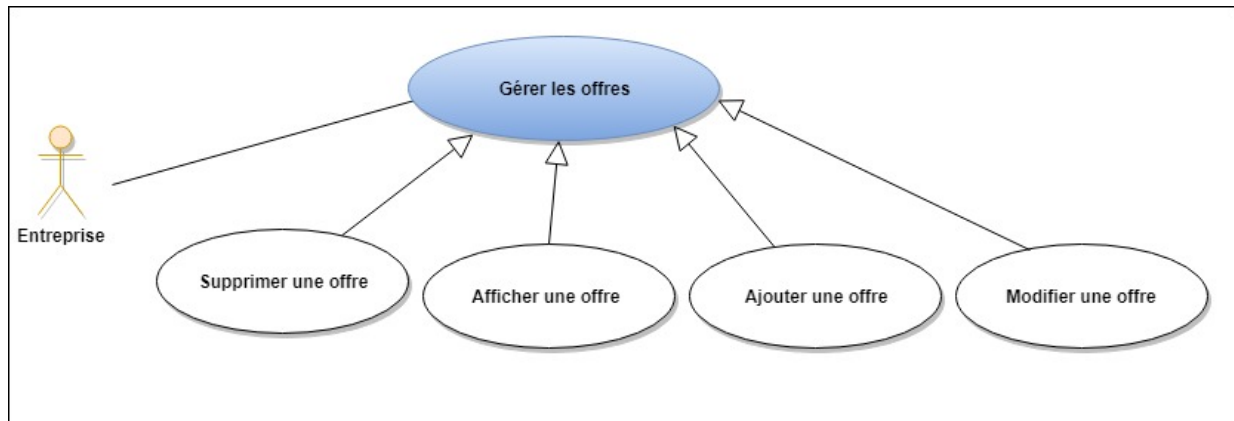


FIGURE 2.2 – Diagramme de cas d'utilisation 'Gérer les offres'

2.4.1.3 Diagramme de cas d'utilisation 'Gérer les CVs recommandés'

Le cas d'utilisation 'Gérer les CVs recommandés' (présenté par la figure 2.3) offre la possibilité à l'entreprise de télécharger les CVs des candidats ayant postulé pour ses offres et de supprimer ceux qu'elle trouve non adéquats.

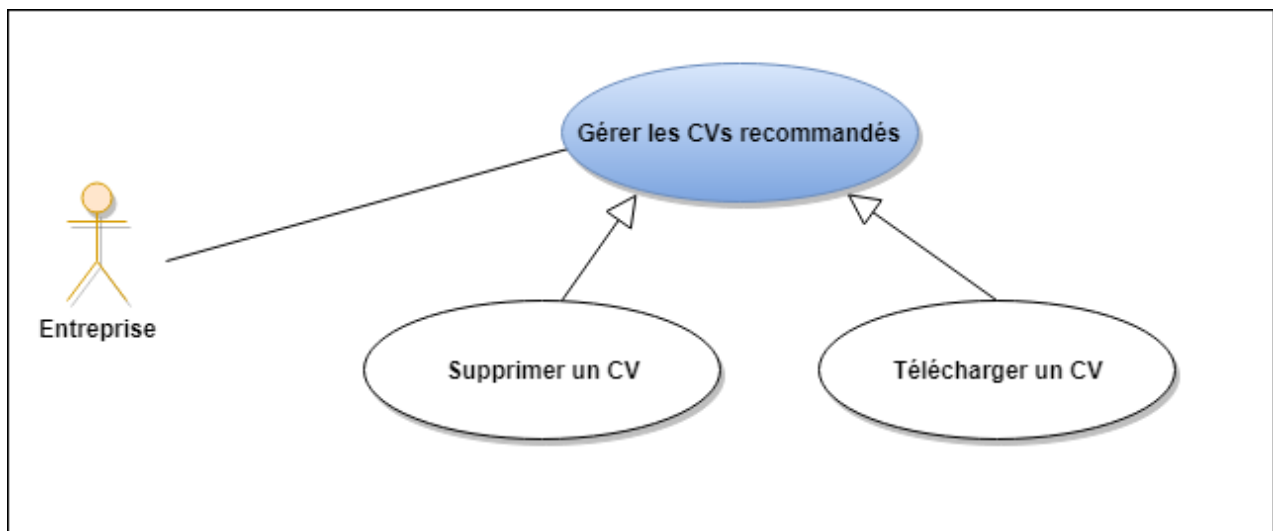


FIGURE 2.3 – Diagramme de cas d'utilisation 'Gérer les CVs recommandés'

2.4.1.4 Diagramme de cas d'utilisation côté Candidat 'Gérer les offres'

Le cas d'utilisation 'Gérer les offres' (présenté par la figure 2.4) permet au candidat de consulter les offres publiées et postuler à n'importe quelle offre souhaitée.

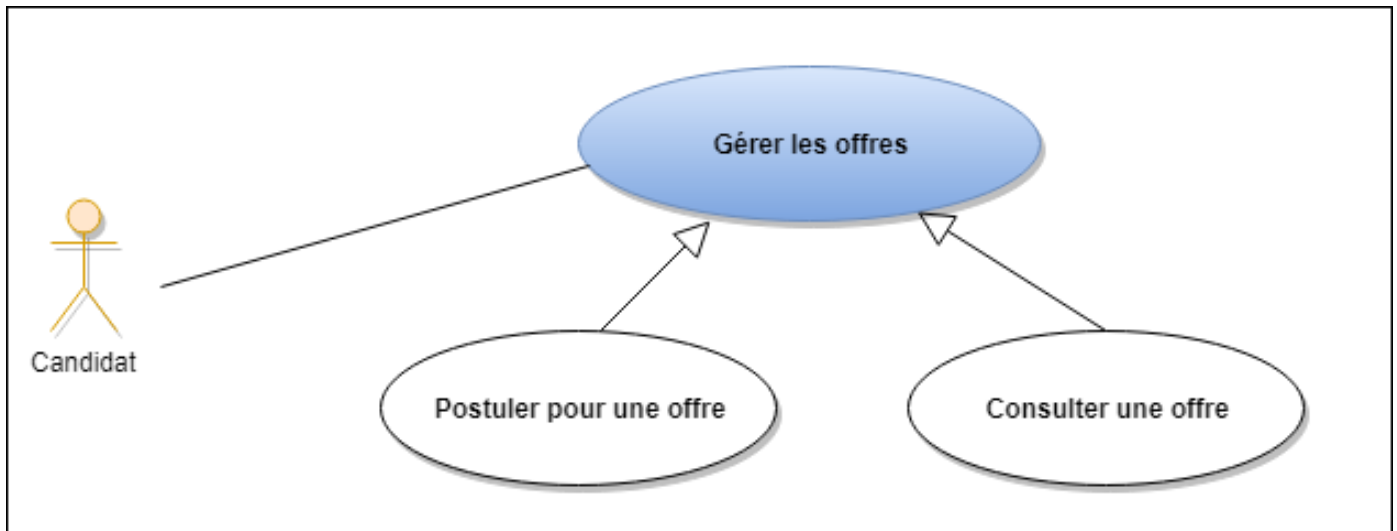


FIGURE 2.4 – Diagramme de cas d'utilisation 'Gérer les offres'

2.4.2 Description de quelques scénarii

Dans cette section, nous détaillons les scénarii de l'utilisation des principaux services offerts par l' application. Nous fournissons une représentation graphique à travers les diagrammes de séquences qui représentent les interactions entre les objets en indiquant la chronologie des séquences. Les diagrammes de séquences ajoutent une dimension temporelle aux diagrammes de collaborations.

2.4.2.1 Diagramme de séquence système pour le scénario 'S'authentifier'

Le système affiche le formulaire d'authentification. Le recruteur ou le candidat remplit le formulaire avec l'ensemble des informations nécessaires à son authentification (comme la montre la figure 2.5). Le système vérifie les informations saisies par l'utilisateur et le renvoie vers l'interface de bienvenue propre à lui.

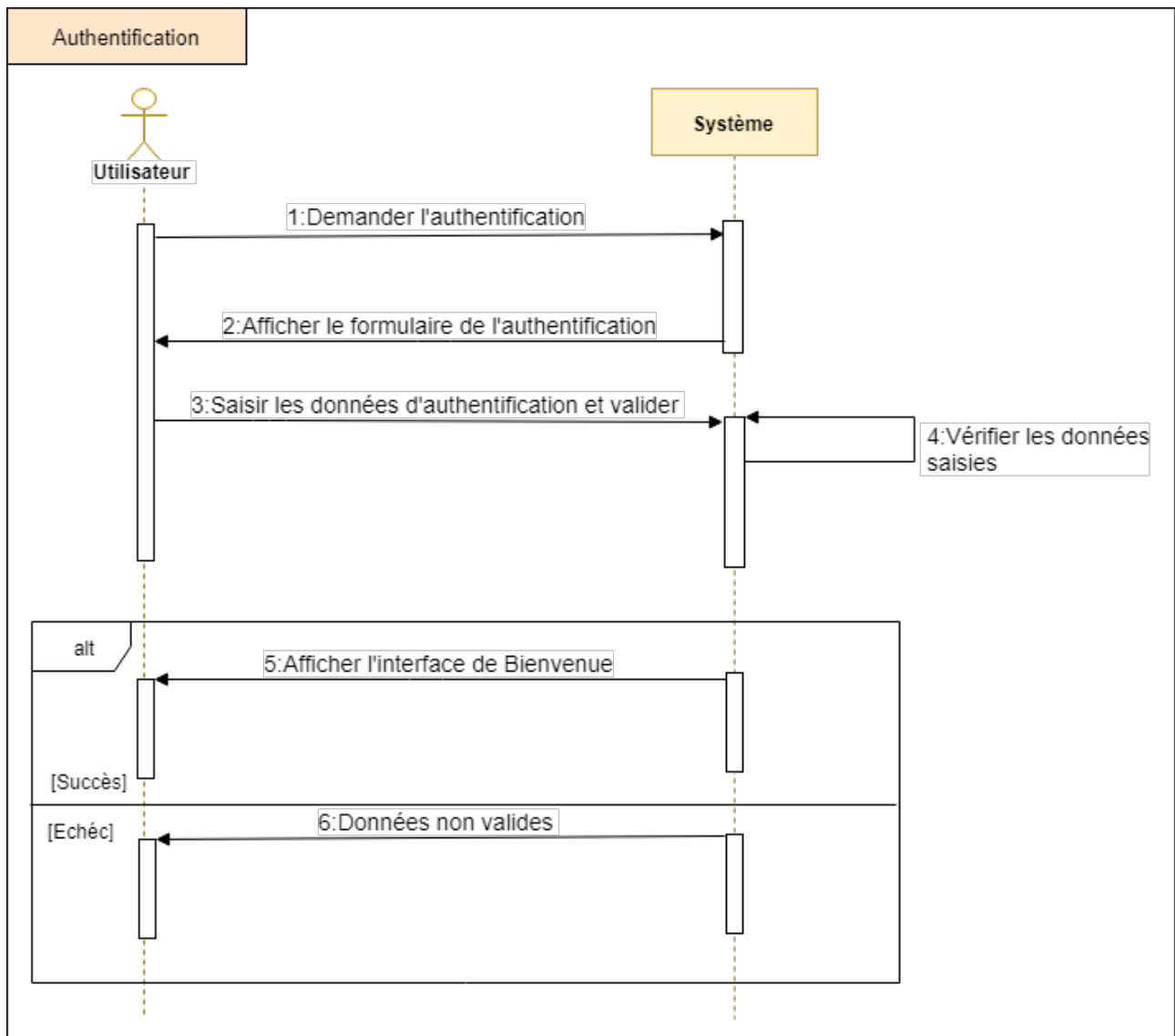


FIGURE 2.5 – Diagramme de séquence système pour le scénario 'S'authentifier'

2.4.2.2 Diagramme de séquence système pour le scénario 'Ajouter une nouvelle offre'

Dans un premier lieu, le recruteur saisit les informations d'authentification qui lui correspondent puis le système les valide (comme la montre la figure 2.6). L'application autorise l'accès au recruteur à son espace puis affiche le formulaire de saisie des données relatif à l'offre à ajouter. Le recruteur saisit alors les informations liés à cette offre. Par la suite, une nouvelle offre est publiée sur la liste des offres d'emploi et est ajoutée à la base de données. Elle sera visible pour tous les utilisateurs.

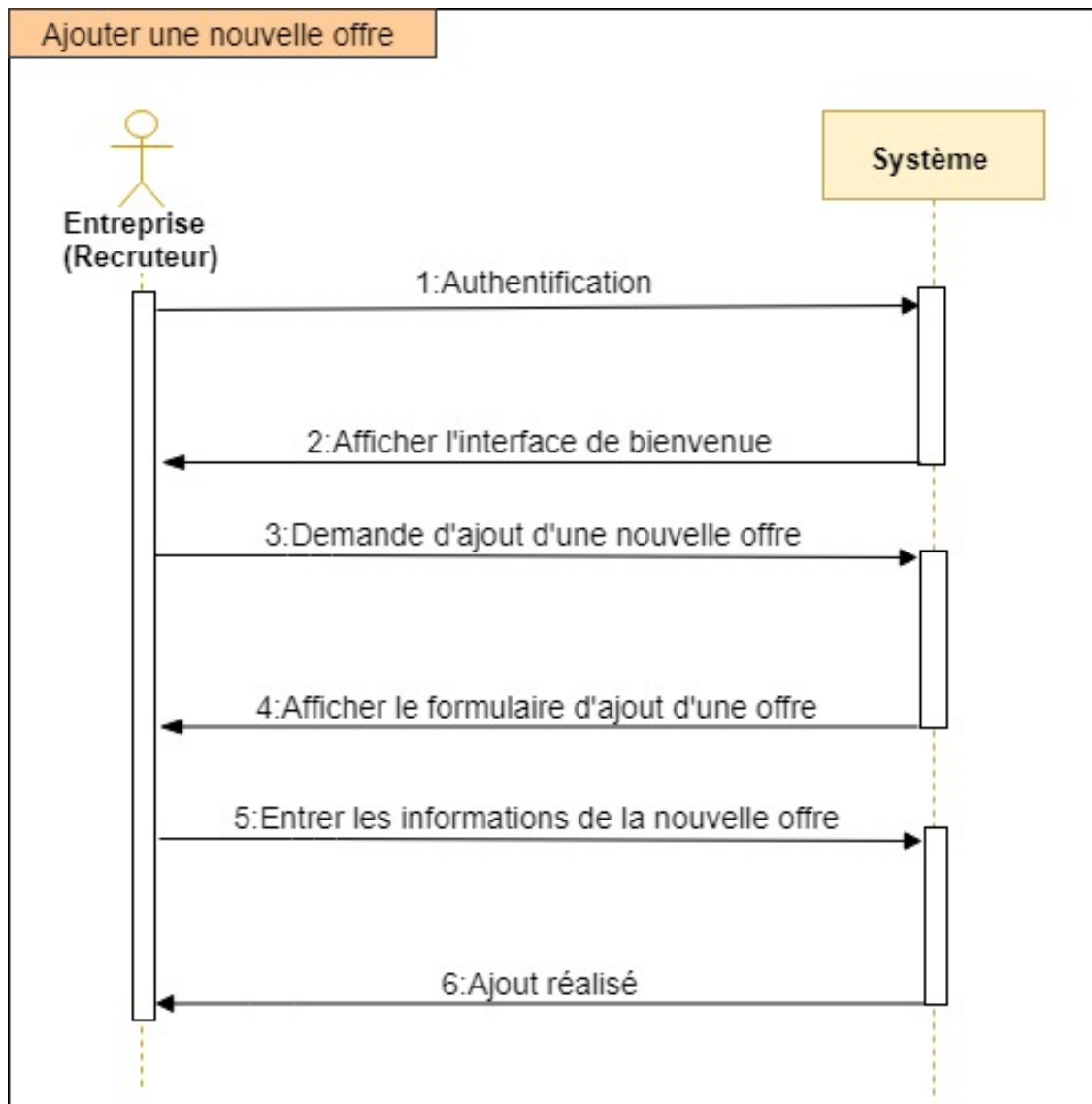


FIGURE 2.6 – Diagramme de séquence système pour le scénario 'Ajouter une nouvelle offre'

2.4.2.3 Diagramme de séquence système pour le scénario 'Postuler à une offre'

La figure 2.7 illustre le diagramme de séquence de 'postuler à une offre'. Une fois authentifié, le candidat peut postuler à une ou plusieurs offres, il aura besoin de choisir l'offre qui le convient et de confirmer par la suite sa candidature (cliquer sur 'postuler').

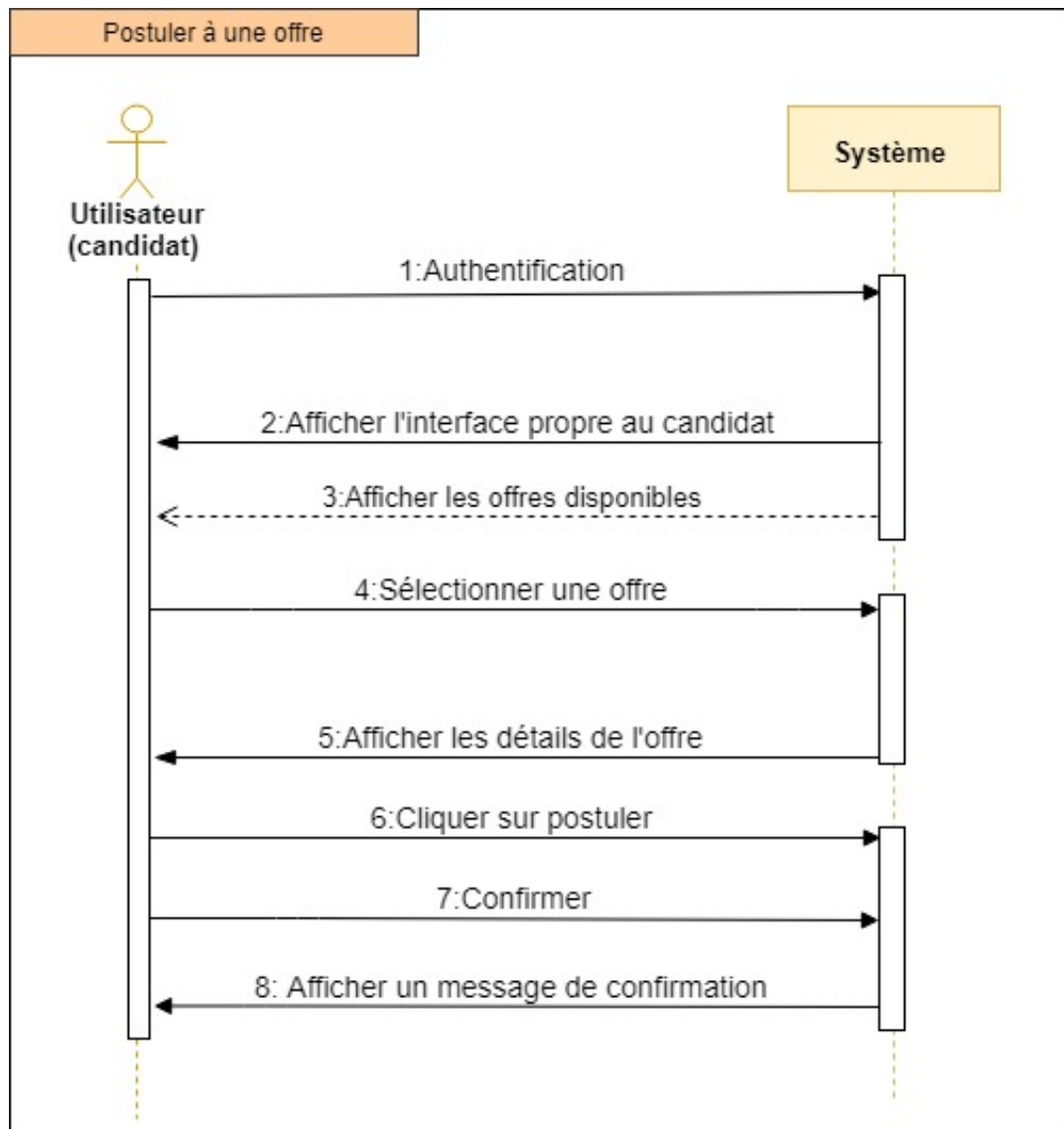


FIGURE 2.7 – Diagramme de séquence système pour le scénario 'Postuler à une offre'.

Conclusion

Ce chapitre nous a permis de couvrir dans la première section la spécification des besoins fonctionnels et non fonctionnels des acteurs de notre système. Dans la deuxième section nous avons fourni une analyse plus détaillée de ces besoins grâce à un diagramme de cas d'utilisation relatif aux acteurs réagissant avec le système (Entreprise et candidat) et trois diagrammes de séquence système décrivant des différents scénarii de notre application. Nous essayons dans le chapitre qui suit de concevoir clairement l'architecture utilisée pour notre système.

Chapitre 3

Conception

Après avoir cité les différentes spécifications et fonctionnalités de notre projet dans le chapitre précédent, nous allons commencer maintenant une phase importante et indispensable dans le cycle de vie d'une application.

Cette phase est la conception qui a pour but d'expliquer le déroulement de notre application ainsi qu'assurer une bonne compréhension des besoins des utilisateurs.

Nous allons exposer dans ce chapitre deux vues conceptuelles, l'une généraliste donnant une vue globale de l'application à partir de l'architecture du système en termes de composants et d'organisation de ses fonctions, et l'autre détaillée décrivant tous les choix conceptuels de la solution proposée sous ses aspects statiques et dynamiques.

3.1 Conception globale

En effet, l'architecture d'un logiciel décrit sa structure globale, sa squelette. Elle décrit les principaux éléments qui composent le logiciel, ainsi que les flux d'échanges entre ces éléments. Elle permet à l'équipe de développement d'avoir une vue d'ensemble de l'organisation du logiciel, et constitue donc en elle-même une forme de documentation.

On peut décrire l'architecture d'un logiciel selon différents points de vue. Une vue logique et une vue physique.

3.1.1 Vue logique

La vue logique décrit la structuration logique du système sans se soucier des détails physiques (équipements, machines...). Elle permet une décomposition modulaire du système hors de tout contexte d'exécution.

3.1.1.1 L'architecture globale de l'application

L'architecture Modèle/Vue/Contrôleur (MVC) est un pattern répandu et puissant qui permet d'organiser une interface graphique d'un programme. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts qui sont,

le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface :[6][7][8]

- **Modèle** : Élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application. Ils permettent d'envoyer des requêtes à la base de données , de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur).
- **Vue** : Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle. La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir tous les actions de l'utilisateur (clic de souris, sélection d'une entrées, boutons, etc).
- **Contrôleur** : C'est le cœur de l'application. C'est lui qui serait appelé en premier. Il serait l'intermédiaire entre le modèle, la vues et toute autre forme de ressources. Cette partie gère la logique du code qui prend desdécisions. Elle va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue.

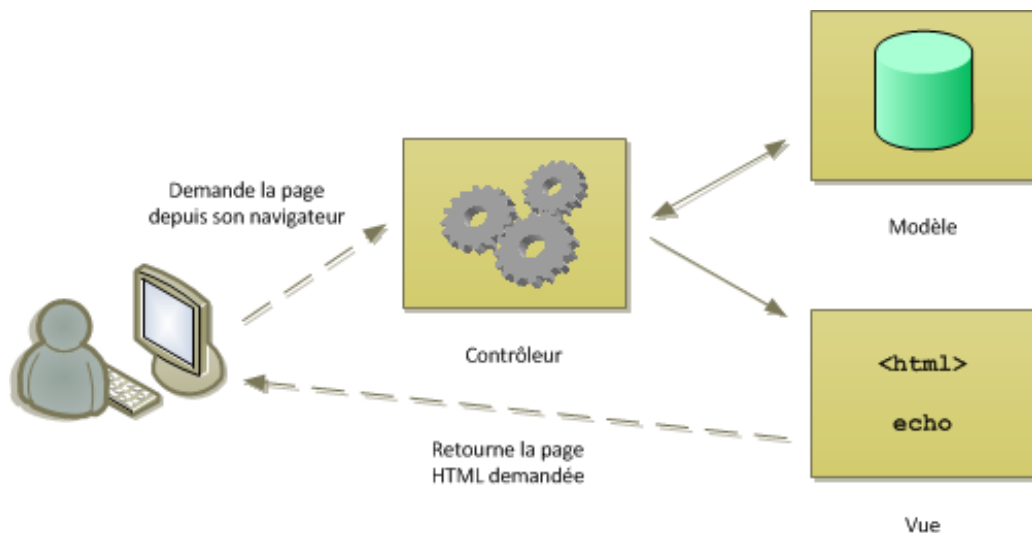


FIGURE 3.1 – Architecture générale de l'application

3.1.1.2 Justification du choix de l'architecture

Le choix correcte de l'architecture présente une étape primordiale pour la phase de conception. Pour notre application, nous avons opté pour le modèle MVC (modèle vue contrôleur).

L'architecture MVC apporte de nombreux avantages :

- Une conception claire et efficace grâce à la séparation des modèles, vue et contrôleur en assurant un couplage faible entre ces éléments. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple, on peut passer d'une base de données de typeSQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.
- Une plus grande souplesse pour organiser le développement du plateforme entre différents développeurs (indépendance des données).
- Un gain de temps de maintenance et d'évolution.
- Diminution de la complexité lors de la conception.
- Adaptée aux applications graphiques.

3.1.1.3 Diagramme de paquets

Un diagramme de paquetage est une collection d'éléments de modélisation statiques (classes, paquets, etc). C'est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de l'application, il aide à identifier les liens de généralisation et de dépendance entre les paquets ainsi que l'ensemble des classes et leurs associations.

Notre diagramme présente :

- Le paquet Vue : renferme les classes qui représentent les interfaces utilisées par l'utilisateur.
- Le paquet Contrôleur : renferme les méthodes responsables de l'implémentation de l'algorithme de recommandation et la gestion de candidature.
- Le paquet Modèle : renferme les classes qui représentent les tables de la blase de données tels que la classe Candidat, Entreprise, Offre et Candidature.

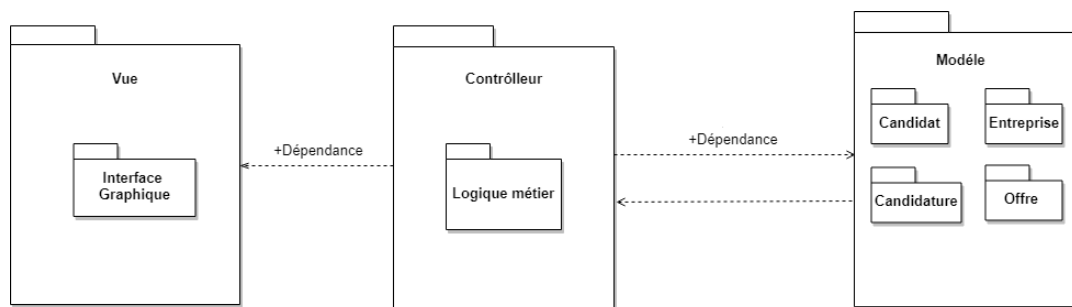


FIGURE 3.2 – Diagramme de Paquets

3.1.2 Vue physique

La vue physique décrit l'architecture physique du système. Elle sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants physiques du système sont répartis.

3.1.2.1 Architecture physique de l'application Web

L'architecture adéquate est L'architecture trois tiers ou architecture à trois couches (comme la montre la figure 3.3). Cette architecture est l'application du modèle plus général qu'est le n-tier, elle consiste à séparer la réalisation des trois parties : stockage des données, logique applicative et présentation. Cette séparation signifie qu'il est possible de déployer chaque partie sur un serveur indépendant.

- **Le tier Client** : C'est un client léger. Il accède à l'application web déployée sur le serveur via son navigateur web. Dans le cadre de notre projet, le client est le candidat et le recruteur.
- **Le tier applicatif** : C'est le serveur web sur lequel est déployée l'application. Dans le cadre de notre projet, nous avons employé le serveur Angular2 et Django.
- **Le tier données** : Il s'agit du serveur de base des données sur lequel est déployé le SGBD SQLite.

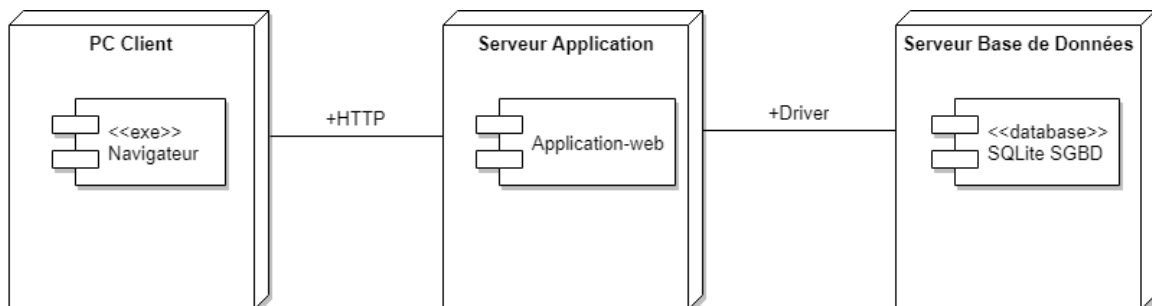


FIGURE 3.3 – Diagramme de déploiement

3.2 Conception détaillée

La conception détaillée d'un projet vise à expliquer en détail l'organisation et la répartition des tâches entre les modules et les objets qui le constituent afin de préparer la phase de réalisation.

Elle a pour but d'expliquer les solutions choisies afin de mettre en place les modules et les différents scénarii qui seront exécutés par notre application.

Netography

[URL1]<https://www.robinwieruch.de/react-testing-tutorial/react-enzyme-test-setup>, visited on 10 July 2018

[URL2]<https://www.adminitrack.com/articles/testingyourwebapps.aspx>, visited on 10 July 2018

[URL3]<https://hackernoon.com/testing-react-components-with-jest-and-enzyme-41d592c174f>, visited on 16 July 2018

[URL4]<https://www.codementor.io/vijayst/unit-testing-react-components-jest-or-enzyme-du1087lh8>, visited on 16 July 2018

[URL5]<https://www.infoworld.com/article/3212208/javascript/the-best-javascript-testing-tools-for-react.html>, visited on 26 July 2018

[URL6]<https://code.tutsplus.com/tutorials/8-things-that-make-jest-the-best-react-testing-framework-cms-30534>, visited on 26 July 2018

[URL7]<https://8thlight.com/blog/uncle-bob/2014/05/10/WhenToMock.html>, visited on 7 August 2018

