



UN-IT

UNIVERSITE DE FIANARANTSOA

ECOLE NATIONALE D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME DE MASTER PROFESSIONNEL

Mention : Informatique

Parcours : Génie logiciel et Base de Données.

Intitulé

Conception et réalisation d'une plateforme de mise en
relation professionnelle pour Madagascar

Présenté le 2025

Par : Monsieur ANDRIANJANDRAMAMINIAINA Tolojanahary Safidy

Membres du Jury :

- **Président :**
- **Examineur :**
- **Rapporteurs :**

CURRICULUM VITAE

ANDRIANJANDRAMAMINIAINA

Tolojanahary Safidy

Age: 24 Ans

Adresse : Pres lot 309/3706 Isaha Anjoma
Fianarantsoa 301

Tel : +261 34 06 492 56

E-mail : safidytolojanahary026@gmail.com

4 x 4

FORMATIONS ET DIPLOMES OBTENUS

2023-2024 : Troisième année en licence professionnel à l'Ecole Nationale d'Informatique (ENI), à Fianarantsoa.

Parcours : Génie Logiciel et Base de données

2022-2023 : Deuxième année en licence professionnel à l'Ecole Nationale d'Informatique (ENI), à Fianarantsoa.

.Parcours : Génie Logiciel et Base de données

2021-2022 : Première année en licence professionnelle en Informatique à l'Ecole Nationale d'Informatique (ENI), à Fianarantsoa.

.Parcours : Génie Logiciel et Base de données

2019 : Obtention du diplôme de BAC Série C au Collège Saint François Xavier Ambatomena Fianarantsoa.

STAGES ET EXPERIENCES PROFESSIONNELLES

Octobre 2024 – Decembre 2024 : Stage au sein de l'Un-iT

Thème : Conception et réalisation d'une plateforme web reliant les entreprises aux candidats

Langage : Javascript

SGBD : MySQL

Bibliothèque : React, node

Aout 2023– Novembre 2023 : Stage au sein de DSI du Ministère intérieur et de la décentralisation (MID)

Thème : Conception et Réalisation d’une application pour la gestion des courriers

Langage : Java

Framework : Springboot

COMPETENCES EN INFORMATIQUE

Langages de programmation et Frameworks : PHP, JavaScript, Laravel, React.js, Node.js

Technologies web : HTML5/CSS3, Javascript, JQuery, Ajax, JSP

Systèmes de Gestion de Base de Données : MySQL, PostgreSQL, Oracle

Méthodes d’analyse et de conception et notation : MERISE, UML

Systèmes de gestion de versions : Git

IDE: PhpStorm, NetBeans, Eclipse, Visual Studio

CONNAISSANCES LINGUISTIQUES

| Langue | Parler | Lire | Ecrire |
|----------|--------|------|--------|
| Français | TB | TB | TB |
| Anglais | AB | AB | AB |

Grille d’évaluation : **TB** : Très Bien ; **B** : Bien ; **AB** : Assez Bien

ACTIVITES EXTRA-PROFESSIONNELLES

- **Jeux video**
- **Basket**

SOMMAIRE

| | |
|--|------|
| CURRICULUM VITAE | I |
| SOMMAIRE | III |
| REMERCIEMENTS | VI |
| LISTES DES FIGURES..... | VII |
| LISTES DES TABLEAUX..... | VIII |
| NOMENCLATURE..... | IX |
| INTRODUCTION GENERALE..... | 1 |
| PARTIE I. PRESENTATIONS | 2 |
| CHAPITRE 1. Présentation de l'Ecole Nationale d'Informatique..... | 3 |
| 1.1. Information d'ordre général | 3 |
| 1.2. Missions et historiques..... | 3 |
| 1.3. Organigramme institutionnel de l'ENI | 5 |
| 1.4. Domaine de spécialisation | 6 |
| 1.5. Architecture des formations pédagogiques | 7 |
| 1.6. Relations de l'ENI avec les entreprises et les organismes | 8 |
| 1.7. Débouchés professionnels avec les diplômés | 9 |
| 1.8. Ressources Humaines | 10 |
| CHAPITRE 2. Présentation de l'UN-IT | 12 |
| 2.1. Localisation et contacts | 12 |
| 2.2. Description et Statuts | 12 |
| 2.3. Activités | 12 |
| 2.4. Organigramme | 14 |
| CHAPITRE 3. Description du projet..... | 15 |
| 3.1. Formulation..... | 15 |
| 3.2. Objectif et besoins de l'utilisateurs..... | 15 |
| 3.3. Moyens nécessaires à la réalisation du projet | 16 |
| 3.4. Résultats attendus..... | 17 |
| 3.5. Chronogramme de travail | 17 |
| PARTIE II. ANALYSE ET CONCEPTION | 18 |
| CHAPITRE 4. Analyse préalable | 19 |
| 4.1. Analyse de l'existant..... | 19 |

| | |
|---|-----|
| 4.2. Critique de l'existant..... | 20 |
| 4.3. Conception avant-projet | 20 |
| CHAPITRE 5. Analyse conceptuelle | 28 |
| 5.1. Présentation de la méthode utilisée..... | 28 |
| 5.2. Dictionnaire des données | 30 |
| 5.3. Règle de gestion..... | 32 |
| 5.4. Représentation et spécification des besoins | 32 |
| 5.5. Spécification des besoins techniques | 48 |
| 5.6. Modélisation du domaine..... | 49 |
| CHAPITRE 6. Conception détaillée..... | 50 |
| 6.1. Architecture système..... | 50 |
| 6.2. Diagramme de séquence de conception pour chaque cas d'utilisation..... | 51 |
| 6.3. Diagramme de classe de conception global | 54 |
| 6.4. Diagramme de paquetage..... | 56 |
| 6.5. Diagramme de déploiement | 57 |
| PARTIE III. REALISATION..... | 59 |
| CHAPITRE 7. Mise en place de l'environnement du développement..... | 60 |
| 7.1. Installation et configuration des outils | 60 |
| 7.1.2 Installation NodeJs | 61 |
| 7.1.3 Installation de Visual Studio Code..... | 62 |
| 7.1.4 Installation de Google Chrome | 62 |
| 7.1.5 Installation de Visual Paradigm | 63 |
| 7.2. Architecture de l'application..... | 64 |
| CHAPITRE 8. Développement de l'application | 65 |
| 8.1. Création de la base de données | 65 |
| 8.2. Codage de l'application (capture du code et description)..... | 66 |
| 8.3. Présentation de l'application..... | 67 |
| Conclusion..... | 69 |
| Bibliographie | IX |
| Webographie | IX |
| Glossaire..... | X |
| Annexes | XI |
| Table des matières | XII |

| | |
|----------------|----|
| Résumé..... | XV |
| Abstract | XV |

REMERCIEMENTS

Ce présent projet n'a pas pu être fait et réalisé sans l'intervention de plusieurs personnes qui m'ont aidé tout au long de cette période de stage ; nous tenons à remercier, en premier lieu, Dieu tout-puissant, sans qui nous n'aurons pas eu l'inspiration et la faculté intellectuelle nécessaire pour l'élaboration de ce mémoire ; en second lieu, nous exprimons nos sincères remerciements à :

- **Monsieur HAJALALAINA Aimé Richard, Docteur HDR, Président de l'Université de Fianarantsoa, pour le bon déroulement de l'année Universitaire.**
- **Monsieur MAHATODY Thomas, Directeur de l'Ecole Nationale d'Informatique, qui m'a offert la chance de réaliser un stage pour approfondir mes connaissances et de s'initier dans le monde professionnel.**
- **Monsieur..., pour avoir accepté mon inscription dans son établissement et d'avoir bien voulu accepter de présider le Jury de ce mémoire.**
- **Monsieur..., Ingénieur en Informatique, pour son appui logistique au bénéfice des stagiaires et d'avoir accepté de m'encadrer durant cette période de stage.**
- **Monsieur RALAIVAO Christian, Docteur en Informatique qui a bien voulu assurer l'encadrement de ce mémoire et faire partie du membre du Jury.**
- **Monsieur..., qui a accepté d'examiner ce travail de mémoire.**

J'adresse également mes vifs remerciements à :

- ✓ **Tous les enseignants et tout le personnel de l'ENI qui ont assuré notre formation pendant cette année ;**
- ✓ **Tout le personnel de Digital pour leur accueil chaleureux dans l'environnement de travail agréable;**
- ✓ **Tous mes amis ainsi que ma famille de m'avoir supportés et soutenus durant mes études.**

LISTES DES FIGURES

| | |
|--|----|
| Figure 1 . Organigramme de l'ENI..... | 6 |
| Figure 2 . Architecture des études correspondant au système LMD..... | 7 |
| Figure 3. Organigramme de l'UN-IT | 14 |
| Figure 4: Les diagrammes d'UML..... | 23 |
| Figure 5: Cycle de développement « Y »..... | 29 |
| Figure 6: le formalisme d'interaction entre un acteur et un cas d'utilisation..... | 33 |
| Figure 7cas d'utilisation du système | 36 |
| Figure 8 Diagramme de séquence du cas d'utilisation « créer profil » | 45 |
| Figure 9 :Diagramme de sequence du cas d'utilisation « s'authentifier » | 46 |
| Figure 10 Diagramme de séquence du cas d'utilisation “gérer profil” | 46 |
| Figure 11. Diagramme de séquence du cas d'utilisation “publier une annonce” | 47 |
| Figure 12 Diagramme de séquence du cas d'utilisation “consulter annonce” | 47 |
| Figure 13. Diagramme de séquence du cas d'utilisation “supprimer annonce” | 47 |
| Figure 14 Diagramme de séquence du cas d'utilisation “valider ou supprimer des contenus” | 48 |
| Figure 15: Modèle du domaine de l'application | 49 |
| Figure 16: Architecture MVC | 50 |
| Figure 17 Diagramme de séquence de conception « créer profile » | 51 |
| Figure 18 Diagramme de séquence de conception « s'authentifier » | 51 |
| Figure 19: Diagramme de séquence de conception « s'authentifier » | 52 |
| Figure 20: Diagramme de séquence de conception « publier annonce » | 52 |
| Figure 21 Diagramme de séquence de conception « consulter annonce »..... | 53 |
| Figure 22: Diagramme de séquence de conception « supprimer annonce » | 53 |
| Figure 23: Diagramme de séquence de conception « gérer contenus » | 54 |
| Figure 24: Diagramme de classe de conception du cas d'utilisation | 55 |
| Figure 25: Diagramme de classe de conception globale du système | 56 |
| Figure 26 Diagramme de paquetage..... | 57 |
| Figure 27: Diagramme de déploiement..... | 58 |
| Figure 28: Installation de WampServer | 61 |
| Figure 29: installation de Node.js | 61 |
| Figure 30: Interface de Visual Studio Code | 62 |
| Figure 31: Interface de Google Chrome..... | 63 |
| Figure 32: Interface de Visual Paradigme..... | 63 |
| Figure 33: Architecture à trois tiers..... | 64 |
| Figure 34 création des tables | 65 |
| Figure 35 création des tables | 65 |
| Figure 36 Codage de l'application pour la partie Modèle | 66 |
| Figure 37 Codage de l'application pour la partie Controleur | 67 |
| Figure 38 Page d'accueil | 68 |
| Figure 39 page d'authentification | 68 |

LISTES DES TABLEAUX

| | |
|--|----|
| <u>Tableau 1 . Organisation du système de formation pédagogique de l'Ecole</u> | 6 |
| <u>Tableau 2 . Mention et parcours au sein de l'ENI</u> | 7 |
| <u>Tableau 3 . Liste des formations existantes à l'ENI</u> | 8 |
| <u>Tableau 4. Débouchés éventuels des jeunes diplômés</u> | 9 |
| <u>Tableau 5 : Matériels informatiques</u> | 16 |
| <u>Tableau 6 : Diagramme de GANTT</u> | 17 |
| <u>Tableau 7 : : Inventaire des moyens matériels</u> | 19 |
| <u>Tableau 8: Avantages et Inconvénients des solutions</u> | 21 |
| <u>Tableau 9: Comparaison entre 2TUP et MERISE</u> | 22 |
| <u>Tableau 10 Comparaison entre Oracle et MySQL</u> | 23 |
| <u>Tableau 11: Comparaison entre JavaScript et Java</u> | 24 |
| <u>Tableau 12: Comparaison entre angular, et React</u> | 26 |
| <u>Tableau 13: Comparaison des outils de modélisation</u> | 26 |
| <u>Tableau 14: Comparaison des environnement de développements</u> | 27 |
| <u>Tableau 15: Dictionnaire des données du domaine</u> | 30 |
| <u>Tableau 16: Identification des acteurs</u> | 34 |
| <u>Tableau 17: Description textuelle du cas d'utilisation " Créer un profil "</u> | 37 |
| <u>Tableau 18 : Description textuelle du cas d'utilisation "S'authentifier"</u> | 38 |
| <u>Tableau 19: Description textuelle du cas d'utilisation "Gérer profil"</u> | 39 |
| <u>Tableau 20: Description textuelle du cas d'utilisation "Publier une annonce"</u> | 40 |
| <u>Tableau 21: Description textuelle du cas d'utilisation " Gérer les candidatures "</u> | 40 |
| <u>Tableau 22: Description textuelle du cas d'utilisation " Consulter les annonces"</u> | 41 |
| <u>Tableau 23 Description textuelle du cas d'utilisation " Supprimer une annonce "</u> | 42 |
| <u>Tableau 24 Description textuelle du cas d'utilisation " Valider ou supprimer les contenus »</u> . | 42 |
| <u>Tableau 25: Priorisation des cas d'utilisation</u> | 43 |

NOMENCLATURE

| | |
|------|--|
| 2TUP | : Two Tracks Unified Process |
| BTS | : Brevet de Technicien Supérieur |
| DTS | : Diplôme de Technicien Supérieur |
| ENI | : Ecole Nationale d'Informatique |
| GB | : GigaBit |
| GHz | : GigaHertz |
| GO | : GigaOctet |
| HTML | : HyperText Markup Language |
| HTTP | : Hypertext Transfer Protocol |
| JSON | : Javascript Object Notation |
| LMD | : Licence-Master-Doctorat |
| MVC | : Modèle-Vue-Contrôleur |
| | |
| RAM | : Random Access Memory |
| RG | : Règle de Gestion |
| CUR | : Centre Universitaire Régional |
| SE | : Système d'Exploitation |
| SQL | : Structured Query Language |
| CNH | : Commission Nationale d'Habilitation |
| L | : Licence |
| M | : Master |
| L1 | : 1 ^{ère} Année de Licence |
| L2 | : 2 ^{ème} Année de Licence |
| L3 | : 3 ^{ème} Année de Licence |
| M1 | : 1 ^{ère} Année en Master |
| M2 | : 2 ^{ème} Année en Master |
| D | : Doctorat |
| | |
| A | : Alphabétique |
| AN | : Alphanumérique |
| N | : Numérique |
| SI | : Systeme d'Information |
| CPU | : Central Processing Unit |
| SGBD | : Systeme de Geston des Bases de Données |
| ID | : Identifiant |
| 2TUP | :2 Tracks Unified Process |
| UML | : Unified Modeling Language |
| CSS | : Cascading Style Sheets |

INTRODUCTION GENERALE

Ces dernières années, l'évolution des réseaux sociaux fut tellement progressive que cela est devenu aujourd'hui une tendance en matière de communications de masse et d'échanges sur internet. UN-IT, une boîte informatique spécialisé dans les conception et réalisation des applications, souhaite améliorer l'expérience des recruteurs ainsi que les intéressés en offrant une plateforme web de mise en relation des deux cites précédemment. Cela répond au besoin croissant des utilisateurs d'avoir une vue des différentes offres d'emploi ou stage sans se déplacer.

C'est pour cela qu'on a opté pour thème de mémoire la réalisation et la conception d'une plateforme web interactive qui permet de mettre en relation entre les entreprises et les chercheurs d'offres à l'aide de node et react. Celui-ci détaille la mise en place d'un nouvel environnement de gestion des informations nécessaires aux utilisateurs et de faciliter la recherche d'offre dans différente secteur d'activité.

La problématique principale abordée dans ce mémoire est la suivante : comment concevoir une application intuitive et fiable pour la présentation des offres de recrutement en ligne ? En répondant a cette question, nous cherchons à surmonter les défis techniques lies aux présentation des différentes offres présenter par les différentes sociétés, ainsi qu'à la présentation d'une interface utilisateur conviviale.

Le présent document de mémoire comporte trois grandes parties, la première partie sera consacrée à la présentation de l'Ecole Nationale d'Informatique, à celle de UN-IT et à la description du projet. La deuxième partie traitera de l'analyse et de la conception relative au projet. Et la troisième partie portera sur la réalisation et la mise en œuvre du projet.

PARTIE I. PRESENTATIONS

CHAPITRE 1. Présentation de l'Ecole Nationale d'Informatique

1.1. Information d'ordre général

L'Ecole Nationale d'Informatique, en abrégé ENI, est un établissement d'enseignement supérieur rattaché académiquement et administrativement à l'Université de Fianarantsoa. Le siège de l'Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa. L'adresse pour la prise de contact avec l'Ecole est la suivante : Ecole Nationale d'Informatique (ENI) Tanambao, Fianarantsoa.

Le numéro de sa boîte postale est 1487 avec le code postal 301.

Téléphone : 034 05 733 36 ou 032 15 204 28.

Son adresse électronique est la suivante : eni@eni.mg.

Il dispose également d'un site web : www.eni.mg

1.2. Missions et historiques

L'ENI se positionne sur l'échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd'hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L'Ecole s'est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l'ENI était constituée et créée au sein du (CUR) par le décret N° 83185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d'Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L'ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

- En fournissant à des étudiants des connaissances de base en informatique ;

- En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.
- En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l'information et de la communication (TIC).

La filière de formation d'Analystes Programmeurs a été mise en place à l'Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d'ingénieurs a été ouverte à l'Ecole en 1986.

Une formation de troisième cycle a été ouverte à l'Ecole a été ouverte à l'Ecole depuis l'année 2003 – 2004 grâce à la coopération académique et scientifique entre l'Université de Fianarantsoa pour le compte de l'ENI et l'Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l'Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l'année 2007-2008, la formation en vue de l'obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l'ENI avec les deux options suivantes de formation :

- Génie Logiciel et base de Données.
- Administration des Système et réseaux.

La mise en place à l'Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD). Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l'effectif des étudiants accueillis à l'Ecole, notamment à cause du manque d'infrastructures, un système de « Formation Hybride » a été mise en place à partir de l'année 2010. Il s'agit en effet d'un système de formation semi présentielle et à distance avec l'utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu'Université de Toliara. Cette formation est à l'origine du parcours Informatique Générale.

En 2023, une nouvelle mention Intelligence Artificielle (IA) a été ouverte au sein de l'Ecole pour répondre aux besoins des entreprises. La formation est destinée aux étudiants titulaires du diplôme de licence (Bac +3) en Mathématiques ou en Statistiques ou en Informatique, etc. La mention IA comporte trois parcours :

- Gouvernance et Ingénierie de Données (GID),
- Objets Connectés et Cybersécurité (OCC).

Et en 2024, la mention Intelligence Artificielle (IA) a ajouté le parcours MDi(Métiers du Digital) qui est destiné aux étudiants titulaires du diplôme BACALAUREAT.

Le principe de l'enseignement pour le parcours GID offre aux étudiants des compétences scientifiques et techniques spécialisées en Science de données. Pour le parcours OCC, les étudiants obtiennent la double spécialité premièrement en internet des objets et deuxièmement en cybersécurité. La formation de master est axée sur l'ensemble d'applications de l'Intelligence Artificielle.

1.3. Organigramme institutionnel de l'ENI

L'organigramme de l'Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983. L'ENI est administrée par un Conseil d'Ecole, et dirigée par un directeur nommé par un décret adopté en Conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs permanents de l'Ecole est chargé de résoudre les problèmes liés à l'organisation pédagogique des enseignements. Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l'établissement, en tenant compte notamment de l'évolution du marché de travail et de l'adéquation des formations dispensées par rapport aux besoins des entreprises.

La figure 1 représente l'organigramme actuel de l'ENI

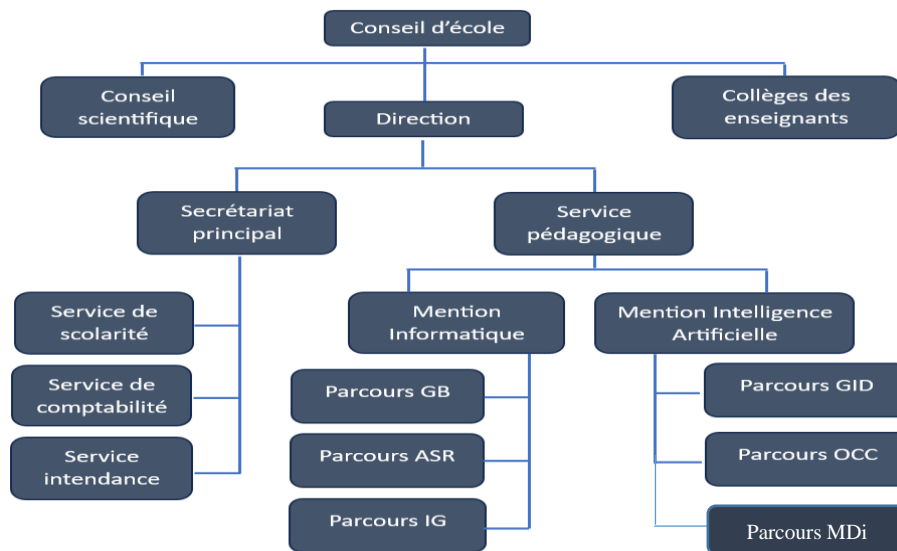


Figure 1 . Organigramme de l'ENI

1.4. Domaine de spécialisation

Les activités de formation et de recherche organisées à l'ENI portent sur les domaines suivants :

- ✓ Génie logiciel et Base de Données ;
- ✓ Administration des Systèmes et Réseaux ;
- ✓ Informatique Générale ;
- ✓ Modélisation informatique et mathématique des Systèmes complexes ;
- ✓ Intelligence artificielle ;
- ✓ Métiers du Digital.

Le tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole

Tableau 1 . Organisation du système de formation pédagogique de l'Ecole

| Formation Théorique | Formation Pratique |
|---|--|
| <ul style="list-style-type: none"> - Enseignement théorique - Travaux dirigés - Travaux pratiques - Conférences | <ul style="list-style-type: none"> - Etude de cas - Travaux de réalisation - Projets / Projets tutorés - Voyage d'études - Stages en entreprise |

1.5. Architecture des formations pédagogiques

Le recrutement des étudiants à l'ENI se fait uniquement par voie de concours d'envergure nationale en première année. Les offres de formation organisées à l'Ecole ont été validées par la Commission Nationale d'Habilitation (CNH). Au sein de l'ENI, il existe deux mentions et cinq parcours.

Le tableau 2 récapitule les mentions et les parcours au sein de l'Ecole :

Tableau 2 . Mention et parcours au sein de l'ENI

| Mention | Parcours |
|----------------------------------|--|
| Informatique | Génie logiciel et Base de Données (GB) |
| | Administration des Systèmes et Réseaux (ASR) |
| | Informatique Générale (IG) |
| Intelligence Artificielle | Gouvernance et Ingénierie de Données (GID) |
| | Objets Connectés et Cyber sécurités (OCC) |
| | Métiers du Digital (MDi) |

La figure 2 représente l'architecture des études correspondant au système LMD.

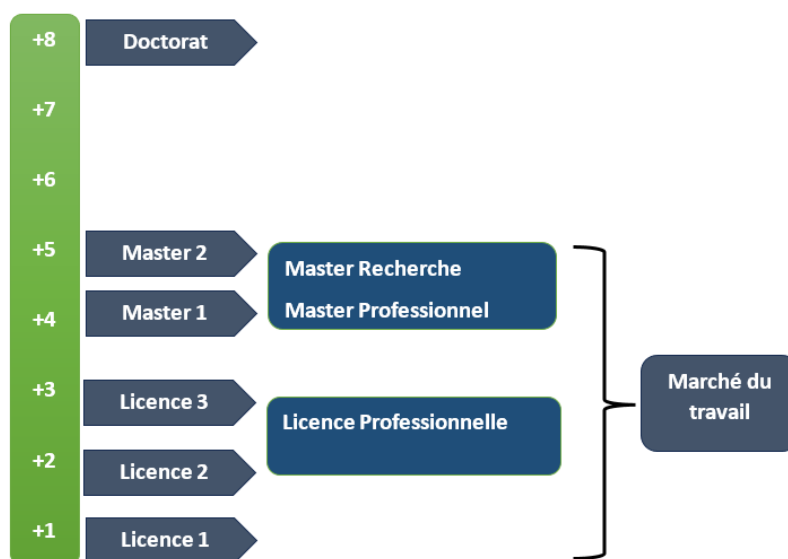


Figure 2 . Architecture des études correspondant au système LMD

La licence peut avoir une vocation générale ou professionnelle. Le master peut avoir une vocation professionnelle ou de recherche. L'accès en première année de MASTER se fait

automatiquement pour les étudiants de l'Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le tableau 3 illustre la liste des formations existantes à l'ENI.

Tableau 3 . Liste des formations existantes à l'ENI

| | FORMATION | |
|---------------------|------------------------------------|--|
| | LICENCE PROFESSIONNELLE | MASTER |
| Condition admission | Par voie de concours | Par voie de concours pour la mention IA |
| Condition d'accès | Bac de série C, D ou Technique | Être titulaire de licence professionnelle |
| Durée de Formation | 3 ans | 2 ans |
| Diplôme délivré | Diplôme de Licence Professionnelle | Diplôme de Master Professionnel Diplôme de Master Recherche |

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale.

Les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (Canada, Suisse, France, ...)

1.6. Relations de l'ENI avec les entreprises et les organismes

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 400 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux. L'Ecole dispose ainsi d'un réseau d'entreprises, de sociétés et d'organismes publics et privés qui sont des partenaires par l'accueil en stage de ses étudiants, et éventuellement pour le recrutement après l'obtention des diplômes par ces derniers. Les compétences que l'Ecole cherche à développer chez ses étudiants sont l'adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l'expérimentation et l'innovation.

En effet, la vocation de l'ENI est de former des licenciés et des ingénieurs de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d'évoluer professionnellement dans des secteurs d'activité variés intégrant l'informatique. Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l'Ecole et les besoins évolutifs du marché de l'emploi.

Parmi les sociétés, les entreprises et les organismes partenaires de l'Ecole, on peut citer : ACCENTURE Mauritius, AKATA Goavana, Air Madagascar, Ambre Associates, Airtel,

Agence Universitaire de la Francophonie (AUF), AXIAN, B2B, Banque Centrale, , BIANCO, BlueLine, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, COLAS, Direction Générale des Douanes, DLC, E-Tech Consulting, , FID, FIHARY Soft, FTM, GNOSYS, GENIUS AT WORK, Hello Tana, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MANAO, MEF, MEN, MESupRES, MFB, , MININTER, Min des Postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SG Madagasikara SMMC, SMMEC, SNEDADRS Antsirabe, Sénat, Société d'Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF, UGD, ARATO, MANAO, MNDPT, NG ACADEMY.NG, Relia...

1.7. Débouchés professionnels avec les diplômés

Les formations proposées par l'Ecole permettent aux diplômés d'être immédiatement opérationnels sur le marché du travail avec la connaissance d'un métier complet lié à l'informatique aux TIC.

L'Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante. Les diplômés en LICENCE et en MASTER issus de l'ENI peuvent faire carrière dans différents secteurs.

L'Ecole bénéficie aujourd'hui de 40 années d'expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C'est une Ecole Supérieure de référence en matière informatique.

D'une manière générale, les diplômés de l'ENI n'éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l'ENI recommande à ses diplômés de promouvoir l'entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d'études. Le tableau 4 représente les débouchés éventuels des jeunes diplômés.

Tableau 4. Débouchés éventuels des jeunes diplômés

| LICENCE | MASTER |
|---|---|
| ➤ Analyste - Programmeur | ➤ Administrateur de réseau et système |
| ➤ Administrateur de site web/de portail web | ➤ Architecture de système d'information |

| | |
|--|--|
| <ul style="list-style-type: none"> ➤ Assistant Informatique et internet ➤ Chef de projet web ou multimédia ➤ Développeur Informatique ou multimédia ➤ Intégrateur web ou web designer ➤ Hot liner/Hébergeur Internet ➤ Agent de référencement ➤ Technicien/Supérieur de help desk sur Informatique ➤ Responsable de sécurité web ➤ Administrateur de réseau | <ul style="list-style-type: none"> ➤ Développeur d'applications ➤ Ingénieur réseau ➤ Webmaster / Web Designer ➤ Concepteur et réalisateur d'application ➤ Directeur du système d'informations ➤ Chef de projet informatique ➤ Responsable de sécurité informatique ➤ Consultant fonctionnel ou freelance |
|--|--|

1.8. Ressources Humaines

Les ressources humaines sont citées ci-dessous selon leurs responsabilités :

- Directeur de l'Ecole : Monsieur MAHATODY Thomas, Professeur
- Responsable de la Mention « Informatique » : Monsieur RABETAFIKA Louis Haja, Maître de Conférences
- Responsable de la Mention « Intelligence Artificielle » : Monsieur DIMBISOA William Germain, Maître de Conférences
- Responsable du Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian, Maître de Conférences
- Responsable du Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA, Assistant d'Enseignement Supérieur et de Recherche
- Responsable du Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy, Assistant d'Enseignement Supérieur et de Recherche
- Responsable du Parcours « Gouvernance et Ingénierie de Données » : Madame RATIANANTITRA Volatiana Marielle, Maître de Conférences
- Responsable du Parcours « Objets Connectés et Cybersécurité » : Monsieur RAZAFIMAHATRATRA Hajarisena, Maître de Conférences

L'ENI compte quinze (15) enseignants permanents dont un (01) Professeur Titulaire, un (01) Professeur, un (01) Docteur HDR, huit (08) Maîtres de Conférences, quatre (04) Assistants d'Enseignement Supérieur et de Recherche, dix (10) enseignants vacataires, quarante un (41) personnel administratif.

CHAPITRE 2. Présentation de l'UN-IT

2.1. Localisation et contacts

Actuellement le siège de UN-IT se trouve à Antamponjina Fianarantsoa Lot 0303 X014 et à Antananarivo.

- Contact : +26134930349
- Email : contact@unityfianar.site
- Site web: www.unityfianar.site
- Adresse: Antamponjina Fianarantsoa

2.2. Description et Statuts

UN-IT, une agence web dynamique, a été fondée en 2022 par un groupe d'ingénieurs passionnés issus de l'**ENI** et de l'**EMIT** de l'Université de Fianarantsoa. Leur ambition ? Transformer les idées en solutions numériques sur mesure tout en stimulant le développement industriel et socio-économique de Madagascar.

Grâce à une expertise technologique reconnue et une approche centrée sur l'humain, **UN-IT** conçoit des solutions innovantes pour permettre aux entreprises et aux particuliers d'accroître leur **visibilité**, leur compétitivité et leur rentabilité.

Notre Vision

Devenir un partenaire de confiance pour les entreprises malgaches et internationales en proposant des solutions numériques de pointe, entièrement adaptées à leurs besoins spécifiques.

Convaincue que la transformation digitale est un levier essentiel de croissance, **UN-IT** s'engage à :

- Fournir des solutions technologiques performantes et durables.
- Stimuler l'emploi local en accompagnant les jeunes talents dans leur développement.
- Soutenir les entreprises dans leur transition numérique pour créer un impact positif et durable.

UN-IT, c'est l'alliance entre innovation, expertise et responsabilité sociale, au service de la croissance des entreprises et du développement de Madagascar.

2.3. Activités

UN-IT est une entreprise qui propose une gamme variée de services pour aider les entreprises et les particuliers à atteindre leurs objectifs mais son activité principale est la création de logiciels personnalisés, d'applications web ou mobiles et de sites web pour répondre aux besoins de chaque client.

Les principales activités et service de UN-IT sont les suivant :

- **Conception UI/UX**

Nous plaçons l'utilisateur au centre de la conception, avec des interfaces intuitives et engageantes basées sur des recherches approfondies et des profils utilisateurs détaillés.

- **Développement Web**

De sites e-commerce dynamiques à des blogs personnalisés, nous développons des solutions web sur-mesure qui répondent à vos objectifs commerciaux.

- **Développement Mobile**

Des applications multiplateformes optimisées pour iOS et Android, offrant une expérience fluide et cohérente sur tous les appareils.

- **Solutions d'Intelligence Artificielle**

Automatisation, personnalisation et optimisation : nous intégrons l'IA pour transformer vos processus métiers et offrir des solutions innovantes.

- **Transformation Digitale**

Accompagnement stratégique pour optimiser vos processus et adopter les dernières technologies, afin de rester compétitif dans un monde numérique en évolution rapide.

- **Maintenance Logicielle**

Une surveillance proactive pour garantir la performance et la stabilité de vos logiciels à long terme.

2.4. Organigramme

Figure 3. Organigramme de l'UN-IT

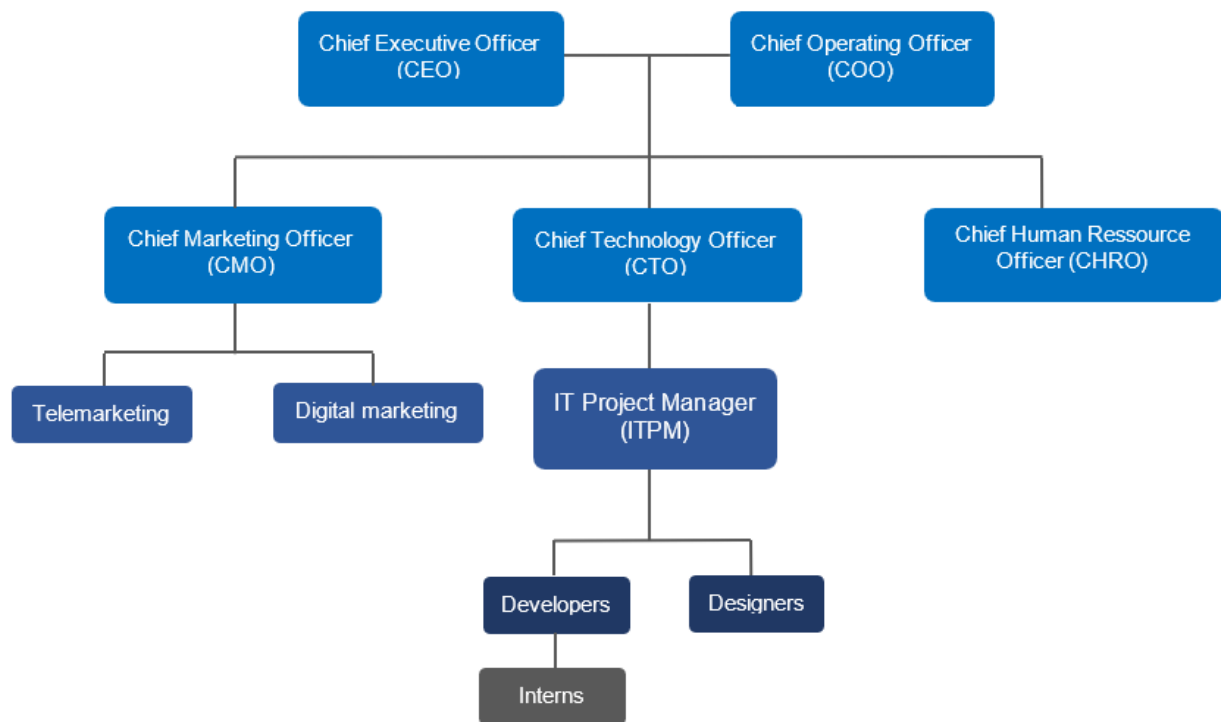


Figure 3. Organigramme de l'UN-IT

CHAPITRE 3. Description du projet

Dans ce chapitre, nous allons faire la description du projet, voir ses objectifs ainsi que les besoins de l'utilisateur. Nous énumérerons les moyens nécessaires à la réalisation du projet ainsi que les résultats attendus.

3.1. Formulation

Dans ce projet, L'objectif principal de "MadaPro" est de créer une plateforme web interactive qui permet de mettre en relation les entreprises, agences, et professionnels à Madagascar. Cette plateforme facilitera la recherche d'entreprises et de profils de personnel par région et activité. Elle offrira également la possibilité aux entreprises de publier des offres d'emploi ou de stage et aux candidats de postuler directement.

3.2. Objectif et besoins de l'utilisateurs

Il s'agit de concevoir une application dédiée à la mise en relation des entreprises aux candidats.

➤ Besoins fonctionnels

L'expression du besoin fonctionnel identifie explicitement le fonctionnement technique, spécifiant ce que l'application devra effectuer et ce qu'elle sera.

A la fin de ce projet les utilisateurs doivent pouvoir :

- S'authentifier et se connecter
- Créer un profil
- Gérer un profil
- Créer une offre (entreprise)
- Postuler à une offre (candidat)

➤ Besoins non fonctionnels

Les besoins non fonctionnels sont des caractéristiques supplémentaires essentielles pour la fonctionnalité technique d'un système. Dans ce cas, l'application doit intégrer les éléments suivants :

- ✓ Sécuration : Dotée d'un système d'authentification, l'application assure que les utilisateurs sont correctement identifiés avant toute interaction. Cette mesure est cruciale pour la protection des données et informations.

- ✓ Facilité d'utilisation : L'application est conçue pour être facilement utilisable, avec une interface intuitive permettant aux utilisateurs de minimiser le temps consacré à l'apprentissage de son fonctionnement.

3.3. Moyens nécessaires à la réalisation du projet

➤ Moyens humains

Afin de bien mener ce projet, il est essentiel de disposer de ressources humaines adéquates. Les moyens humains sont :

- Un encadreur professionnel pour mener et gérer de façon efficace le projet
- Un concepteur/développeur pour analyser et développer le projet (Un stagiaire)

➤ Moyens matériels

Chaque projet informatique nécessite une ressource appropriée pour que son exécution réussisse.

Les matériels utilisés pour ce projet sont détaillés dans le tableau 5

Tableau 5 : Matériels informatiques

| Matériel | Système | Détaille |
|----------|--------------------------|---|
| ASUS | Windows 10 professionnel | ❖ Disque dur : 500 Go ❖ Processeur : Core i5 4eme génération ❖ Ram : 8 Go |

➤ Moyens logiciels

Pour garantir une progression efficace dans la réalisation de ce projet, il est impératif que tous les participants utilisent la même version des logiciels, évitant ainsi des perturbations inutiles liées à des problèmes techniques mineurs.

- IDE : Visual Studio Code
- SGBD : Wampserver
- Outil de modélisation : Visual Paradigm
- Navigateur : Google Chrome

3.4. Résultats attendus








En considérant les objectifs, les besoins de l'utilisateur et les moyens nécessaires pour la réalisation du projet, à la fin, on doit obtenir les résultats suivants :

- Une application web de recrutement fiable et exploitable
- Une base de données sécurisée

3.5. Chronogramme de travail

Durant le stage, nous avons dessiné le diagramme de Gantt comme le tableau 6 représente

Tableau 6 : Diagramme de GANTT

| ID | Nom de tâche | Début | Fin | Durée | Octobre 2024 | Novembre 2024 | Décembre 2024 |
|----|--|----------|----------|---------|---|---|---|
| | | | | | 20 Octobre-30 Octobre | 01 Novembre-30 Novembre | 01 Décembre-20 Janvier |
| 1 | Visite de l'entreprise | 20/10/24 | 22/10/24 | 2jours |  | | |
| 3 | Enquête sur le thème | 24/10/24 | 27/10/24 | 3jours |  | | |
| 4 | Elaboration du planning | 28/10/24 | 29/10/24 | 1jours |  | | |
| 5 | Analyse et Conception du projet | 31/10/24 | 11/11/24 | 13Jours |  | | |
| 6 | Formation et installation des outils nécessaires | 11/11/24 | 13/11/24 | 2jours |  | | |
| 7 | Réalisation du projet | 13/11/24 | 03/01/25 | 30Jours | |  | |
| 8 | Elaboration du mémoire | 13/11/24 | 03/01/25 | 22jours | | |  |

PARTIE II. ANALYSE ET CONCEPTION

CHAPITRE 4. Analyse préalable

En raison de l'importance de cette étape, la solution à appliquer, le modèle de modélisation à adopter, l'outil de modélisation à utiliser, etc., sont tous déterminés par le résultat de cette analyse.

En premier lieu, nous procéderons à l'analyse de l'existant ; ensuite, nous formulerons des critiques à l'égard de l'existant et, pour finir, nous réaliserons une conception avant-projet

4.1. Analyse de l'existant

L'analyse de l'existant permet de décortiquer, d'analyser en profondeur l'état actuel du système.

4.1.1. Organisation actuelle

A présent, UN-IT se lance dans cette application pour mettre en relation les candidats et les entreprises dans leurs domaines respectifs, afin de faciliter le recrutement de nouveaux employés compétents et expérimentés pour les entreprises, ainsi que de faciliter la recherche d'emploi pour les candidats.

4.1.2. Inventaire des moyens matériels et logiciels

L'UN-IT possède des moyens matériels et logiciels. Le tableau 6 regroupe les listes des matériels et logiciel appartenant à l'entreprise.

Tableau 7 : : Inventaire des moyens matériels

| Désignation | CPU | RAM | HDD | Fréquence | Système | Nombre |
|-------------|---------|------|-------|-----------|---------------|--------|
| Bureau | Core i7 | 8 Go | 1To | 3.20GHz | Windows 10 | 2 |
| Portable | Core i5 | 8 Go | 500Go | 3.70GHz | Windows 10 | 4 |

4.2. Critique de l'existant

Après avoir fait l'analyse de l'existant, on peut en déduire les atouts et les inconvénients de l'existant :

Tableau 8: Les atouts et inconvénients de l'existant

| Atouts | Inconvénients |
|---|--|
| <ul style="list-style-type: none">- Les utilisateurs peuvent familiariser avec le système actuel, ce qui réduit le temps de formation nécessaire pour l'utilisation du système.- La mise en place du système actuel a nécessité un investissement initial relativement faible en termes de matériel et de logiciels. | <ul style="list-style-type: none">- Perte de temps sur le développement de l'application.-Dépendance de la technologie et des systèmes informatiques.- Possibilité de privilégier les clients qui se présentent bien en ligne plutôt que ceux qui ne sont pas. |

4.3. Conception avant-projet

4.3.1. Propositions des solutions

Pour résoudre les problèmes, on propose les solutions suivantes :

- Solution 1 : Utilisation d'un logiciel libre comme LinkedIn ou PortailJob.
- Solution 2 : Conception et réalisation d'une application web pour la mise en relation entre les entreprises et les candidats.

➤ Avantages et inconvénients de chaque solution

Le tableau 8 représente les avantages et les inconvénients de ces deux environnements.

Tableau 8: Avantages et Inconvénients des solutions

| Solutions | Avantages | Inconvénients |
|------------|---|--|
| Solution 1 | <ul style="list-style-type: none"> - Logiciel prêt à l'emploi -Sécurisation des données | <ul style="list-style-type: none"> - N'est pas conforme aux besoins de la société |
| Solution 2 | <ul style="list-style-type: none"> - Résultat plus adapté aux besoins de la société -Bénéfique. | <ul style="list-style-type: none"> - Perte de temps sur le développement de l'application. - Changement d'habitude journalier. |

➤ **Solution retenue**

Après analyse, en tenant compte les critiques et les besoins de l'utilisateurs, nous avons opté pour la Solution 2.

4.3.2. Méthodes de conception et outils utilisés

Pour réaliser la solution numéro 2, nous avons besoin d'une méthode de conception, d'un langage de programmation, d'un outil de modélisation et un système de base de données.

➤ **Choix de méthode de conception**

On entend par la méthode de conception un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.

La comparaison entre les méthodes 2TUP et MERISE est dressée dans le tableau 9.

Tableau 9: Comparaison entre 2TUP et MERISE

| Méthode | MERISE | 2TUP |
|----------------|--|--|
| Avantages | <ul style="list-style-type: none"> - Modéliser correctement une application, au niveau des données et du traitement - Distinguez les données des traitements | <ul style="list-style-type: none"> - Itératif et incrémentale - Cycle de développement en Y - Utilise UML comme langage de modélisation |
| Inconvénients | <ul style="list-style-type: none"> - Merise n'est orienté objet - Cette méthode n'est pas reconnue mondialement | <ul style="list-style-type: none"> - Ne propose pas de document types - Prend beaucoup de temps à mettre en œuvre |

D'après la comparaison, la meilleure solution est d'utiliser la méthode 2TUP pilotée par UML, un langage universel pour la modélisation. Puisque l'application doit être ouverte pour des améliorations futures. Cette méthode est aussi itérative, orienté objet et fait une large place à la technologie. Cela permet en effet de clarifier la conception et de faciliter les échanges.

➤ **Présentation de l'UML**

UML : « *Unified Modeling Language* » traduit « *langage de modélisation objet unifié* » standardisé par l'OMG (Object Management Group) est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE et se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

UML est un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions ;
- L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions ;

- Son indépendance par rapport aux langages de programmation aux domaines d'application et aux processus, en fait un langage universel.

Le formalisme UML s'articule autour de 13 types de diagrammes. Ces types de diagrammes sont répartis en deux grands groupes : des diagrammes structurels et comportementaux. UML n'étant pas une méthode, son utilisation est laissée à l'appréciation de chacun, même si le diagramme des cas d'utilisation est généralement considéré comme l'élément central d'UML. De même, on peut se contenter de modéliser partiellement un système, par exemple certaines parties critiques.

La figure 4 dans la page suivante montre les diagrammes d'UML.

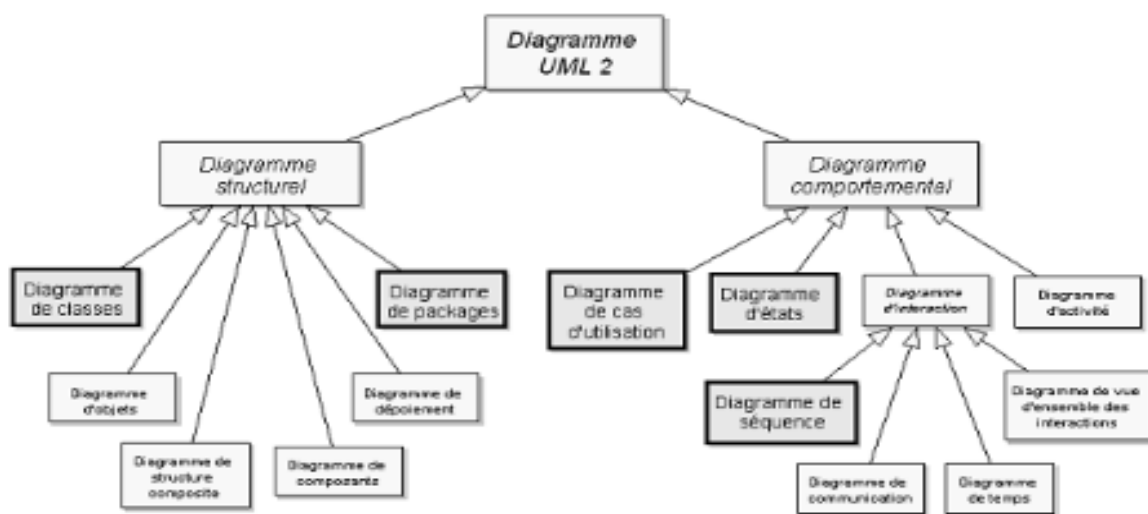


Figure 4: Les diagrammes d'UML.

➤ Choix de système de gestion de base de données

Pour mener à bien le projet, on a besoin d'un SGBD.

Tableau 10 Comparaison entre Oracle et MySQL

| SGBD | Avantages | Inconvénients |
|-------|---|---|
| Mysql | <ul style="list-style-type: none"> - Open source et logiciel libre - Simplicité d'installation et de prise en main de l'administration - Bonne vitesse de gestion des requêtes | <ul style="list-style-type: none"> - Sécurité faible - Faible performance pour les données très volumineuses. |

| | | |
|------------|--|--|
| PostgreSQL | <ul style="list-style-type: none"> - Simplicité de migration vers un autre BD. - Sécurité de données avancées. | <ul style="list-style-type: none"> - Installation et mis en place plus complexe que MySQL. - Vitesse de lecture relativement faible - Documentation |
|------------|--|--|

On a choisi le SGBD MySQL grâce à sa facilité d'utilisation et son omniprésence dans le milieu SGBD.

➤ Présentation de MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) basé sur le langage SQL. Il est composé d'une collection de programmes qui permettent de gérer des bases de données relationnelles. C'est l'un des systèmes de gestion de bases de données les plus populaires et il est largement utilisé sur le Web. Il est open source et disponible pour une variété de plates-formes, notamment Linux, Windows, macOS et Solaris. MySQL est utilisé pour stocker des données enregistrées dans des tables, qui sont regroupées dans des bases de données.

➤ Choix de langage de programmation

Dans cette partie on va comparer différents langages de programmation web qui pourraient être utilisés dans le cadre du développement d'application web qui permettraient de répondre aux besoins. Le tableau 11 représente la comparaison des langages de programmation.

Tableau 11: Comparaison entre JavaScript et Java

| Langage | JavaScript | Java |
|-----------|--|---|
| Avantages | <ul style="list-style-type: none"> - Il est intégré au navigateur web. - Un langage de script côté client. - Compatible avec de nombreux Framework. | <ul style="list-style-type: none"> - Multiplateforme (peut s'exécuter sur des nombreux types de systèmes d'exploitation) - Compatible avec des nombreuses bibliothèques et Framework - Sécurisé. |

| | | |
|---------------|--|--|
| | - Un langage multiplateforme. | |
| Inconvénients | <ul style="list-style-type: none"> - Faible sécurisation dans le code - Faible performance | <ul style="list-style-type: none"> - Nécessite un environnement d'exécution spécifique - Sensible aux vulnérabilités - Nécessite un grand nombre de bibliothèques supplémentaires |

D'après le tableau comparatif numéro 10, nous avons choisi JavaScript comme langage de programmation pour développer le projet.

➤ **Présentation de JavaScript**

JavaScript est un langage de programmation multiplateforme, très populaire et léger, qui permet de créer des applications web interactives, des expériences utilisateur riches et des interfaces utilisateur. Il est intégré aux navigateurs web et s'exécute directement dans le navigateur de l'utilisateur sans nécessiter d'installer des logiciels supplémentaires.

JavaScript est un langage de script qui s'exécute directement sur le navigateur de l'utilisateur, ce qui évite de surcharger les serveurs et rend les applications plus rapides et plus réactives. Il peut également être utilisé pour créer des applications côté serveur grâce à des Framework tels que Node.js, ce qui permet d'exécuter du JavaScript sur le serveur.

JavaScript est un langage qui ne nécessite pas de déclaration de type de données pour les variables et adopte des principes de programmation modernes tels que la programmation orientée objet, les fonctions anonymes et les closures. Divers Framework et bibliothèques, comme jQuery, AngularJS, ReactJS et VueJS, sont disponibles pour faciliter la création d'applications web plus avancées.

➤ **Choix du framework**

Le tableau 12 résume les avantages et les inconvénients de l'utilisation entre Angular et React.

Tableau 12: Comparaison entre angular, et React

| Framework | Avantages | Inconvénients |
|-----------|---|--|
| Angular | <ul style="list-style-type: none"> -Mise en route simple - Liaison de données bidirectionnelle -Utilisation de directives personnalisées -Injection de dépendante | <ul style="list-style-type: none"> -Exécution et chargement lent -Taille des fichiers volumineux à charger |
| React.js | <ul style="list-style-type: none"> -Flexibilité et responsive (intégration très rapide) -Une véritable boîte à outils offrant un très large choix aux développeurs -100% open source -réduction du nombre d'opération sur DOM | <ul style="list-style-type: none"> -Pas de moteur pour injection de dépendances - Requiert une connaissance approfondie pour intégrer l'interface utilisateur au framework MVC |

Puisque, dans ce projet, la performance et la légèreté de l'application est à cibler, React.js est le Framework le mieux adapté.

➤ **Choix d'un outil de modélisation**

Pour faciliter les modélisations, on a besoin d'un outil. Il en existe plusieurs. Nous avons comparé trois outils dans le tableau 13.

Tableau 13: Comparaison des outils de modélisation

| Nom | Créateur | Plateforme | Open source | Licence logiciel |
|--------------|----------|--------------------------------|-------------|------------------|
| UML Designer | Obeo | Multiplate-forme(Java/Eclipse) | Oui | EPL |

| | | | | |
|-------------------------|----------------------------|------------------------|-----|--|
| Visual Paradigm for UML | Visual Paradigm Int'l Ltd. | Multiplate-forme(java) | Non | Commercial avec une édition communautaire gratuite |
| WhiteStarUML | Janusz Szpilewski | Windows | Oui | GPL |

Selon les données présentées dans le tableau 14, le choix de Visual Paradigm comme outil de modélisation s'impose en raison de ses nombreux atouts et de sa convivialité.

➤ Environnement de développement

Pour le développement de l'application, nous avons eu des choix multiples sur une vaste et forte compétition entre eux.

Nous allons faire une comparaison dans le tableau 14 pour nous décider lequel on va utiliser pour notre projet.

Tableau 14: Comparaison des environnement de développements

| | Visual Code Studio | Atom | Sublime text |
|-------------|--------------------|----------------|---------------------------------|
| Coût | Gratuit | Gratuit | 70 \$ |
| Développeur | Microsoft | GitHub | Jon Skinner, Ingénieur à Google |
| Platform | Windows, Linux | Windows, Linux | Windows, Linux |

Comme Visual Code Studio est multiplateforme, gratuit et présente d'innombrable plugin et surtout il supporte git, nous l'avons choisi pour mener à bien notre projet.

CHAPITRE 5. Analyse conceptuelle

Dans ce chapitre, nous détaillerons les dictionnaires de données, suivis des règles de gestion, puis aborderons la représentation et la spécification des besoins.

5.1. Présentation de la méthode utilisée

Un processus définit une séquence d'étapes ordonnées qui permet de produire un système logiciel ou de faire évoluer un système existant. 2TUP ou Two Tracks Unified Process est un processus unifié c'est-à-dire construit sur UML, itératif, centré sur l'architecture et conduit par le cas d'utilisation. 2TUP a pour but d'apporter une réponse aux contraintes de changements fonctionnels et techniques qui s'imposent aux systèmes d'information.

La diversité des systèmes et des techniques de modélisation ne permet pas de définir un seul processus universel. Les Créateurs d'UML ont travaillé à unifier meilleures pratiques de développement orienté objet pour donner le processus unifié.

2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels. Il part du constat que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique. Il distingue ainsi deux branches (fonctionnelle et technique) dont les résultats sont fusionnés pour réaliser le système. On obtient un processus de développement en Y et le processus s'articule autour de trois phases essentielles :

➤ Branche fonctionnelle ou « gauche »

Elle vise la capture des besoins fonctionnels et analyse des spécifications fonctionnelles de manière à déterminer ce que va réaliser le système en termes de métier. C'est ici, qu'on identifie et dégage toutes les fonctionnalités du système à réaliser.

➤ Branche technique ou « droite »

Elle permet la capture des besoins non fonctionnels. Il s'agit essentiellement des contraintes que l'application doit prendre en compte

comme par exemple les contraintes d'intégrations, les contraintes de développements et les contraintes de performances.

➤ Phase de réalisation

Cette phase est la fusion des deux précédentes et mène à la conception applicative et à la solution adaptée aux besoins des utilisateurs. Elle concerne les étapes de la conception préliminaire, la conception détaillée, le codage et les tests puis l'étape de recette.

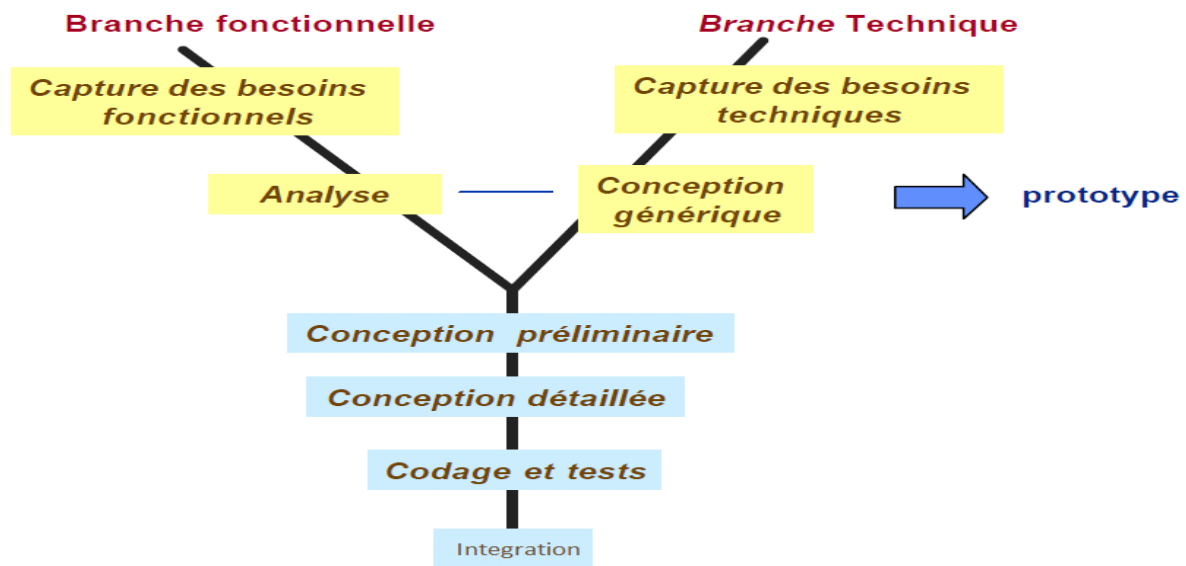


Figure 5: Cycle de développement « Y »

Comme la figure 5 le présente, le processus s'articule ensuite autour de trois phases essentielles :

- Une branche technique
- Une branche fonctionnelle
- Une phase de réalisation

La branche droite (architecture technique) : capitalise un savoir-faire technique et/ou des contraintes techniques. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

- La capture des besoins techniques ;

- La conception générique.

La branche gauche (fonctionnelle) : capitalise la connaissance du *métier* de l'entreprise.

Elle constitue généralement un investissement pour le moyen et le long terme. Les fonctions du système d'information sont en effet indépendantes des technologies utilisées. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux. Cette branche comporte les étapes suivantes :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs ;
- L'analyse.

La branche du milieu (réalisation) : à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins. Cette fusion conduit à l'obtention d'un processus en forme de **Y**.

Cette branche comporte les étapes suivantes :

- La conception préliminaire
- La conception détaillée
- Le codage et tests
- L'intégration

5.2. Dictionnaire des données

Un dictionnaire des données est une collection de métadonnées ou de données de référence nécessaire à la conception d'une base de données relationnelle. Ainsi après collecte des informations, on a pu élaborer un dictionnaire de données qui sera présenté par le tableau 15 :

Tableau 15: Dictionnaire des données du domaine

| Nom | Description | Type | Taille | Observation |
|------------|--|------|--------|-------------|
| Adresse | Adresse complète de l'entreprise | AN | 20 | |
| Compétence | Liste des compétences principales | AN | 50 | |
| Conctact | Informations de contact (email, téléphone) | AN | 50 | |

| | | | | |
|------------------|---|----|----|--|
| Date_candidature | Date de soumission de la candidature | D | 10 | |
| Date_publication | Date de publication de l'annonce | D | 10 | |
| Description | Détails de l'annonce ou brève description de l'entreprise | AN | 50 | |
| id_annonce | Identifiant unique de l'annonce | N | 5 | |
| id_entreprise | Identifiant unique de l'entreprise | N | 5 | |
| id_candidature | Identifiant unique de la candidature | N | 5 | |
| id_region | Identifiant unique de la région | N | 5 | |
| Id_profil | Identifiant unique du profil | N | 5 | |
| Id_secteur | Identifiant unique du secteur d'activité | N | 5 | |
| localisation | Localisation géographique | AN | 50 | |
| nom | Nom de l'entreprise ou nom complet du personnel | AN | 20 | |
| niveau | Niveau de qualification | AN | 20 | |
| Nom_region | Nom de la région | AN | 50 | |
| nom_secteur | Nom du secteur d'activité | AN | 50 | |
| région | Région de Madagascar où se trouve l'entreprise | AN | 20 | |
| Secteur_activite | Secteur d'activité de l'entreprise | AN | 20 | |
| statut | Statut de la candidature | AN | 20 | |
| Taille | Taille de l'entreprise | AN | 20 | |
| experience | Nombre d'années d'expérience | AN | 50 | |

Types : AN : Alphanumérique

 D : Date

 N : Numérique

5.3. Règle de gestion

Une règle de gestion permet de définir le cahier des charges qui constitue la base de l'engagement que prend le concepteur vis à vis des utilisateurs. Le système est soumis à quelques règles de gestions telles que :

- RG01 : Chaque entreprise doit avoir un nom, une adresse, **un** contact, **une** région, **et** un secteur d'activité.
- RG02 : Une entreprise est classée en trois tailles : petite, moyenne **ou** grande.
- RG03 : La suppression d'une entreprise supprime automatiquement toutes ses annonces.
- RG04 : Un profil doit contenir un nom, des compétences, un niveau (junior, intermédiaire, senior), et une localisation.
- **RG05** : Un profil peut être supprimé à tout moment, ce qui supprime aussi toutes ses candidatures
- RG06 : Une annonce doit avoir un titre, une description, et un type (service, emploi, stage).
- RG07 : Seules les entreprises peuvent publier des annonces.
- **RG08** : La suppression d'une annonce supprime automatiquement toutes les candidatures associées
- RG09 : Un candidat peut postuler à une annonce s'il a un profil valide.
- RG10 : Une candidature ne peut être envoyée qu'une seule fois par annonce.
- RG11 : Une candidature a trois statuts : en attente, acceptée, ou rejetée.
- RG12 : Les utilisateurs doivent se connecter pour publier une annonce (entreprises).
- RG13 : Les utilisateurs doivent se connecter pour postuler à une annonce (candidats).

5.4. Représentation et spécification des besoins

5.4.1. Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du langage UML, il assure la relation entre l'utilisateur et les objets que le système met en œuvre.

Un cas d'utilisation peut être divisé en diagrammes de séquence qui détaillent les différentes fonctions du cas d'utilisation. Il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

- **Acteur** : représente un rôle joué par une entité externe « utilisateur humain, dispositif matériel ou autre système » qui interagit directement avec le système étudié. [1]
- **L'interaction** : Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation. L'interaction entre un acteur et un cas d'utilisation se représente comme une association. Elle peut comporter des multiplicités comme toute association entre classes.
- **Cas d'utilisation** : représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation est une description du système étudié privilégiant le point de vue de l'utilisateur. Il permet une meilleure structuration des besoins des utilisateurs qui définissent clairement la manière dont ils interagissent avec le système. Les cas d'utilisation sont liés par des relations de plusieurs types.

La figure 6 représente la formalisation d'interaction entre un acteur et un cas d'utilisation.



Figure 6: le formalisme d'interaction entre un acteur et un cas d'utilisation

5.4.1.1 Relation entre cas d'utilisation

Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas d'utilisation issu. Une relation simple entre un acteur et une utilisation est un trait simple. UML définit deux types de relations standardisées entre cas d'utilisation détaillées ci-après :

- Une relation d'inclusion, formalisée par le mot-clé « include »
- Une relation d'extension, formalisée par le mot-clé « extend »

➤ La relation « include » entre cas d'utilisation

Le cas de base incorpore explicitement un autre, à un endroit spécifié dans ses enchaînements. Le cas inclus n'est jamais exécuté seul, mais seulement en tant que partie d'un cas de base plus vaste.

Dans une relation « include », le cas d'utilisation de base utilise systématiquement les enchaînements provenant du cas inclus. On utilise fréquemment cette relation pour éviter de décrire plusieurs fois le même enchaînement, en factorisant le comportement commun dans un cas d'utilisation à part.

➤ La relation « extend » entre cas d'utilisation

Le cas de base incorpore implicitement un autre à un endroit spécifié indirectement dans celui qui étend. Le cas de base peut fonctionner tout seul, mais il peut également être complété par un autre, sous certaines conditions, et uniquement à certains points particuliers de son flot d'évènements appelés points d'extension.

Dans la relation « extend », le cas de base recourt optionnellement aux enchaînements provenant du cas d'extension. On utilise principalement cette relation pour séparer le comportement optionnel du comportement obligatoire.

5.4.1.2 Identification des acteurs

Les acteurs de notre système sont :

- Entreprise
- Administrateur
- Candidat

Nous allons maintenant décrire ces acteurs dans le tableau 16

Tableau 16: Identification des acteurs

| Acteurs | Description |
|------------|--|
| - Candidat | <ul style="list-style-type: none">- Crée et gère son profil (nom, compétences, niveau, localisation, contact, etc.)- Consulte les annonces publiées par les entreprises |

| | |
|----------------|--|
| | <ul style="list-style-type: none"> - Postule directement aux annonces d'emploi ou de stage - Peut mettre à jour ou supprimer son profil à tout moment |
| - Entreprise | <ul style="list-style-type: none"> - Crée et gère son profil (nom, adresse, description, contact, etc.) - Publie des annonces de services, d'emploi ou de stage - Reçoit et gère les candidatures des candidats - Change le statut des candidatures (en attente, acceptée, rejetée) - Consulte les profils des candidats disponibles - Gère l'offre - Postule à une offre |
| Administrateur | <ul style="list-style-type: none"> - Gère l'ensemble des données de la plateforme - Valide ou modifie les fiches des entreprises. - Modère les annonces et les profils des candidats - Gère les listes de régions et de secteurs d'activité - Supervise les interactions entre les entreprises et les candidats - Supprime les contenus inappropriés ou obsolètes |

- Candidat : Les personnes talentueuses qui cherchent des offres pour lesquelles postuler. Elles créent un profil.
- Entreprise : Les recruteurs sont ceux qui mettent en ligne des offres d'emploi pour trouver des candidats à embaucher. Ils sont à l'origine des offres et les publient pour que les candidats puissent les voir.

- #### 5.4.1.3 Identification des CU

[illegible]

36

5.4.2. Description textuelle pour chaque cas d'utilisation

Pour donner une autre définition du cas d'utilisation, on peut dire que c'est une collection de scénarios de succès ou d'échec qui décrit la façon dont un acteur particulier utilise un système pour atteindre un objectif.

Pour détailler la dynamique du cas d'utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système.

a) Cas d'utilisation « Créer un profil »

À travers le tableau 17, nous présentons la description textuelle du cas d'utilisation « Créer un profil ».

Tableau 17: Description textuelle du cas d'utilisation " Créer un profil "

| Cas d'utilisation | Créer un profil |
|-------------------|---|
| Objectif | Permet à l'utilisateur de publier des annonces ou postuler. |
| Résumé | Pour voir l'ensemble du site. |
| Acteur | Entreprise et Candidat |

- Description des enchainements :
 - Préconditions : L'utilisateur doit s'authentifier sur la plateforme.
 - Post Conditions : Après authentification, l'acteur peut publier ou postuler.
- Scénario nominal :
 - L'acteur clique sur "Créer un profil".
 - Il remplit les champs requis dans le formulaire.
 - Il valide la création.
 - Le système enregistre le profil et affiche un message de confirmation.
- Scénario alternatif :
 - L'acteur n'a pas rempli tous les champs.
 - Le système affiche un message d'erreur.
 - Erreur d'authentification : nom utilisateur ou mot de passe non valide.

b) Cas d'utilisation « S'authentifier »

À travers le tableau 18, nous présentons la description textuelle du cas d'utilisation « S'authentifier ».

Tableau 18 : Description textuelle du cas d'utilisation "S'authentifier"

| | |
|-------------------|--|
| Cas d'utilisation | <u>S'authentifier</u> |
| Objectif | Permet à l'utilisateur de s'identifier en saisissant son adresse email et son mot de passe. |
| Résumé | Pour pouvoir accéder aux fonctionnalités de l'application l'utilisateur doit s'authentifier. |
| Acteur | Entreprise ou Candidat |

- Description des enchainements :
 - Préconditions : L'acteur doit avoir un compte : être présent dans la base de données, Connexion internet.
 - Post Conditions : Acteur authentifié
- Scénario nominal :
 - . L'acteur ouvre le site
 - . La page d'accueil s'affiche.
 - . L'acteur click sur le bouton se connecter
 - . Le système affiche la page d'authentification.
 - . L'acteur saisit son adresse email et son mot de passe.
 - . Le système vérifie si les données existent.
- Scénario alternatif :
 - L'acteur n'a pas rempli tous les champs,
 - Le système affiche un message d'erreur,
 - Erreur d'authentification,
 - Le système affiche un message d'erreur.

c) Cas d'utilisation « Gérer profil »

À travers le tableau 19, nous présentons la description textuelle du cas d'utilisation « Gérer profil »

Tableau 19: Description textuelle du cas d'utilisation "Gérer profil"

| Cas d'utilisation | Gérer profil |
|-------------------|--|
| Objectif | Permet à l'acteur de modifier, ajouter et supprimer des nouvelles informations |
| Acteur | Entreprise ou candidat |

- Description des enchainements :

- Préconditions après être connecté, l'acteur doit être connecté,
- Post Conditions : avoir un profil modifiable.

- Scénario nominal :

- L'acteur se connecte et accède à son profil.
- Il clique sur "Modifier".
- Il modifie les champs souhaités.
- Il valide les modifications.
- Le système met à jour le profil et affiche un message de confirmation.

d) **Cas d'utilisation « Publier une annonce »**

À travers le tableau 20, nous présentons la description textuelle du cas d'utilisation « Publier une annonce ».

Tableau 20: Description textuelle du cas d'utilisation "Publier une annonce"

| | |
|-------------------|--|
| Cas d'utilisation | Publier une annonce |
| Objectif | Permet aux entreprises de publier une annonce d'emploi, de stage ou de service. Elle doit remplir un formulaire précisant le titre, la description, et le type de l'annonce. |
| Acteur | Entreprise |

- Description des enchainements :

- Préconditions : L'entreprise doit avoir un profil valide et être connectée.
- Post Conditions : Publication des annonces

- Scénario nominal :

- L'entreprise se connecte à la plateforme.
- Elle remplit le formulaire d'annonce.
- Elle valide l'annonce.
- Le système enregistre l'annonce et la rend visible aux utilisateurs.

- Scénario alternatif :

- L'entreprise n'a pas rempli tous les champs.

e) **Cas d'utilisation « Gérer les candidatures »**

À travers le tableau 21, nous présentons la description textuelle du cas d'utilisation « Gérer les candidatures ».

Tableau 21: Description textuelle du cas d'utilisation " Gérer les candidatures "

| | |
|-------------------|---|
| Cas d'utilisation | Gerer les candidatures |
| Objectif | Permet au entreprise consulter les candidatures reçues pour leurs annonces. Elles peuvent changer le statut d'une candidature (en attente, acceptée, rejetée). Les candidats peuvent suivre l'état de leurs candidatures. |

| | |
|--------|------------|
| Acteur | Entreprise |
|--------|------------|

- Description des enchainements :

Préconditions : Une annonce doit exister.

Post Conditions : voir la liste des candidatures

- Scénario nominal :

- L'entreprise se connecte à la plateforme.
- Elle consulte une candidature.
- Elle change le statut de la candidature.
- Le système met à jour le statut et notifie le candidat.

- Scénario alternatif :

- L'acteur n'as pas rempli les formulaires.

f) Cas d'utilisation « Consulter les annonces »

À travers le tableau 22, nous présentons la description textuelle du cas d'utilisation « Consulter les annonces ».

Tableau 22: Description textuelle du cas d'utilisation " Consulter les annonces"

| | |
|-------------------|---|
| Cas d'utilisation | Consulter les annonces |
| Objectif | Le candidat peut consulter les annonces publiées sur la plateforme. Les annonces sont filtrables par type (emploi, stage, service) et par région. |
| Acteur | Entreprise |

- Description des enchainements :

- Préconditions : Des annonces doivent être publiées sur la plateforme.
- Post Conditions : consultation des annonces

- Scénario nominal :

- L'utilisateur accède à la liste des annonces,
- Il sélectionne les critères de filtrage,
- Le système affiche les annonces correspondant aux critères,
- L'utilisateur consulte les détails d'une annonce.

- Scénario alternatif :
 - L'acteur n'a pas rempli tous les champs.

g) Cas d'utilisation « Supprimer une annonce »

À travers le tableau 23, nous présentons la description textuelle du cas d'utilisation

« Supprimer une annonce »

Tableau 23 Description textuelle du cas d'utilisation " Supprimer une annonce "

| Cas d'utilisation | Supprimer une annonce |
|-------------------|--|
| Objectif | Une entreprise peut supprimer une annonce qu'elle a publiée. Cette suppression entraîne également la suppression des candidatures associées. |
| Acteur | Entreprise |

- Description des enchainements :
 - Préconditions : L'entreprise doit avoir une publication sur une annonce.
 - Post Conditions : suppression des annonces
- Scénario nominal :
 - L'entreprise se connecte et accède à la liste de ses annonces.
 - Elle sélectionne l'annonce à supprimer.
 - Le système supprime l'annonce et toutes les candidatures associées.

h) Cas d'utilisation « Valider ou supprimer les contenus »

À travers le tableau 24, nous présentons la description textuelle du cas d'utilisation

« Valider ou supprimer les contenus »

Tableau 24 Description textuelle du cas d'utilisation " Valider ou supprimer les contenus »

| Cas d'utilisation | Valider ou supprimer les contenus |
|-------------------|--|
| Objectif | L'administrateur peut valider ou supprimer des contenus publiés sur la plateforme. Il supervise le bon fonctionnement de la plateforme et assure la conformité des contenus. |
| Acteur | Admin |

- Description des enchainements :
 - Préconditions : L'administrateur doit être connecté au système.
 - Post Conditions : validation ou suppression des contenus
- Scénario nominal :
 - L'administrateur accède à la liste des contenus à valider.
 - Il consulte un contenu (profil, annonce ou candidature).
 - Il valide ou supprime le contenu selon les règles de gestion.
 - Le système met à jour l'état du contenu et notifie l'utilisateur concerné.

5.4.3. Priorisation des cas d'utilisation

La priorité des cas d'utilisation est déterminée par rapport aux besoins de l'utilisateur : les cas les plus prioritaires sont ceux qui garantissent un fonctionnement minimal du logiciel

Tableau 25: Priorisation des cas d'utilisation

| Cas d'utilisation | Priorité |
|------------------------------|----------|
| S'authentifier | Forte |
| Créer profile | Forte |
| Gerer candidatures | Forte |
| Gérer profiles | Moyen |
| Publier annonce | Moyen |
| Consulter annonce | Moyen |
| Supprimer annonce | Faible |
| Valider ou supprimer contenu | Faible |

5.4.4. Diagramme de séquence système pour chaque cas d'utilisation

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Ils peuvent être utilisés pour montrer les interactions entre les objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

Il fait ressortir :

- Les acteurs : représentés par un bonhomme et une ligne verticale (ligne de vie).
- Les objets : représentés par un rectangle et une ligne verticale (ligne de vie de l'objet). Les messages : traduisent les interactions (échange d'informations) entre objets. Ils sont représentés par des flèches orientées de l'émetteur au récepteur. Deux types de messages peuvent être distingués :
 - Message synchrone : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.
 - Message asynchrone : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

Le fonctionnement d'un cas d'utilisation est notamment décrit sous la forme d'une séquence de message échangé entre les acteurs et le système. Le système est vu de l'extérieur (par les acteurs) sans préjuger de comment il est réalisé. On obtient donc le diagramme de séquence système.

La figure 8 représente le diagramme de séquence système du cas d'utilisation « créer profil ».

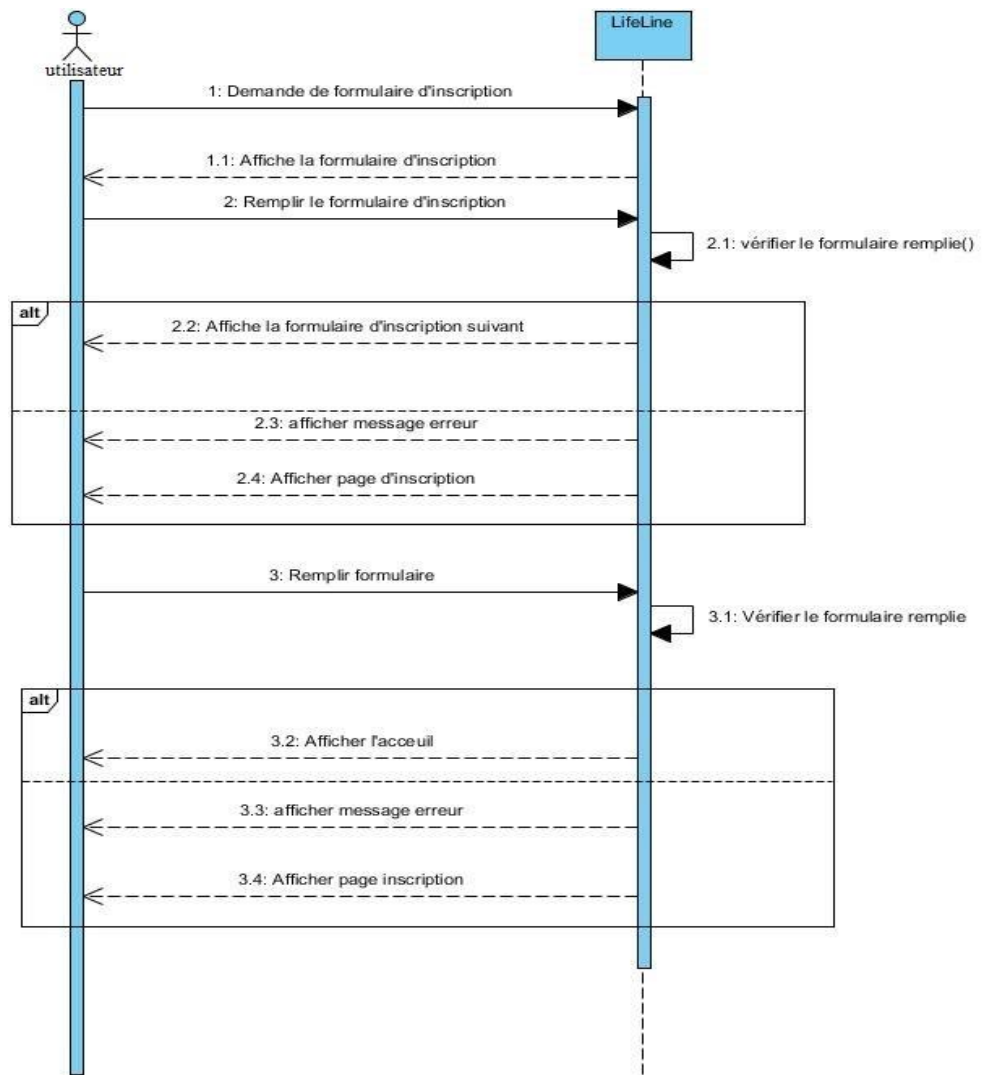


Figure 8 Diagramme de séquence du cas d'utilisation « créer profil »

La figure 9 représente le diagramme de séquence système du cas d'utilisation « s'authentifier ».

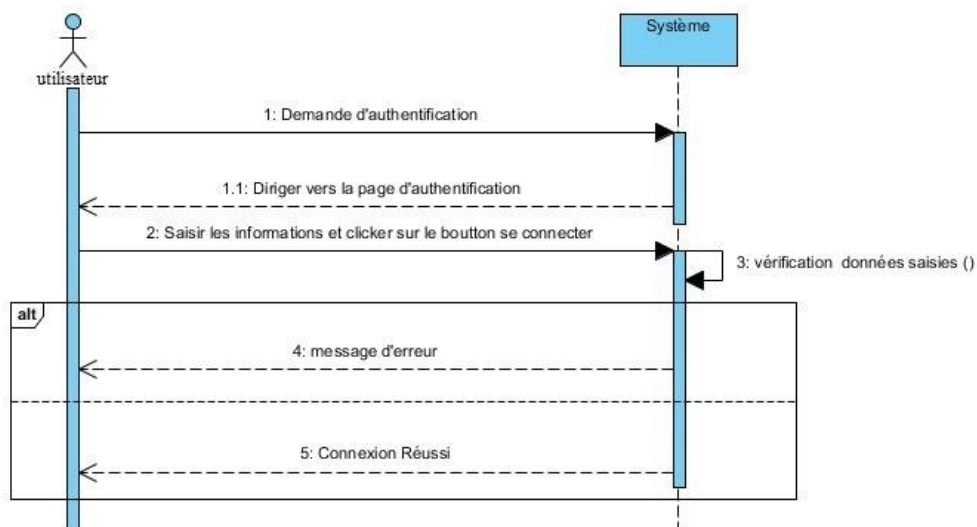


Figure 9 :Diagramme de sequence du cas d'utilisation « s'authentifier »

La figure 10 représente le diagramme de séquence système du cas d'utilisation « gérer profil ».

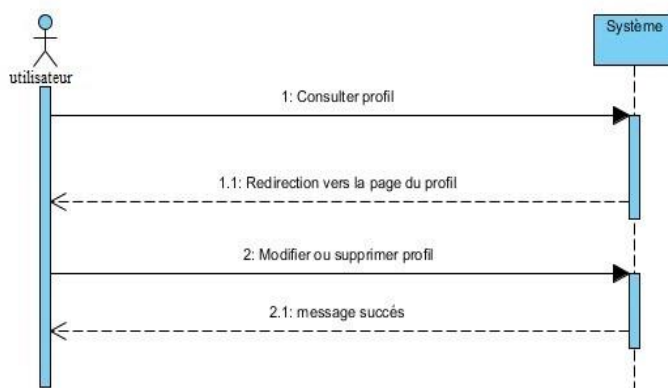


Figure 10 Diagramme de séquence du cas d'utilisation "gérer profil"

La figure 11 représente le diagramme de séquence système du cas d'utilisation « Publier une annonce »

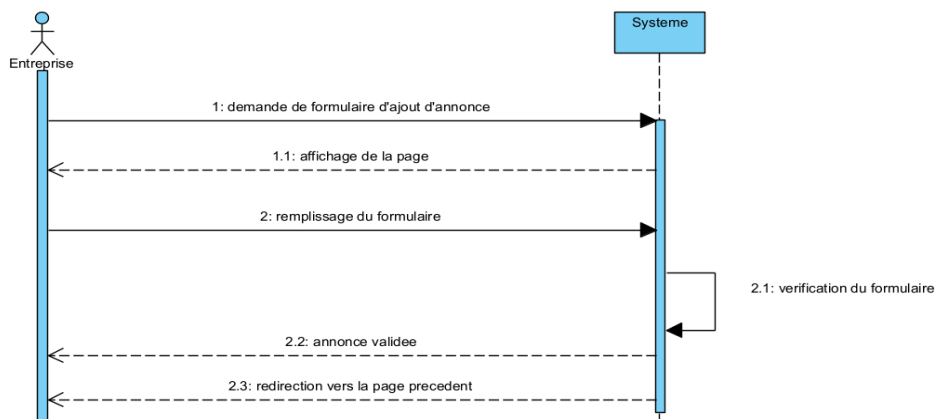


Figure 11. Diagramme de séquence du cas d'utilisation “publier une annonce”

La figure 12 représente le diagramme de séquence système du cas d'utilisation « consulter une annonce ».

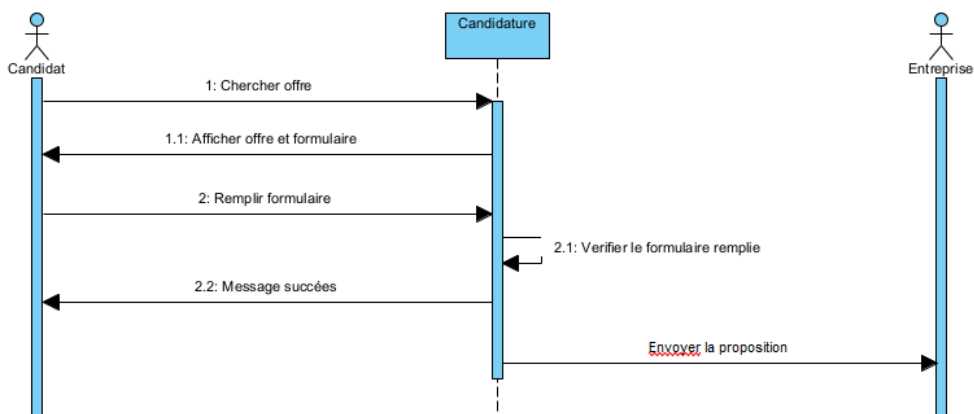


Figure 12 Diagramme de séquence du cas d'utilisation “consulter annonce”

La figure 13 représente le diagramme de séquence système du cas d'utilisation « supprimer annonce ».

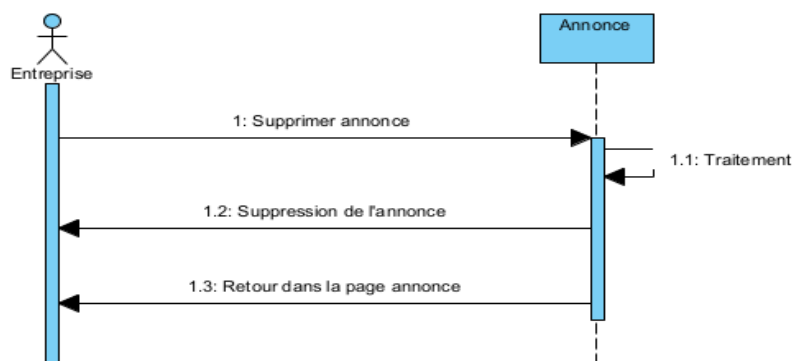


Figure 13. Diagramme de séquence du cas d'utilisation “supprimer annonce”

La figure 14 représente le diagramme de séquence système du cas d'utilisation « valider ou supprimer des contenus ».

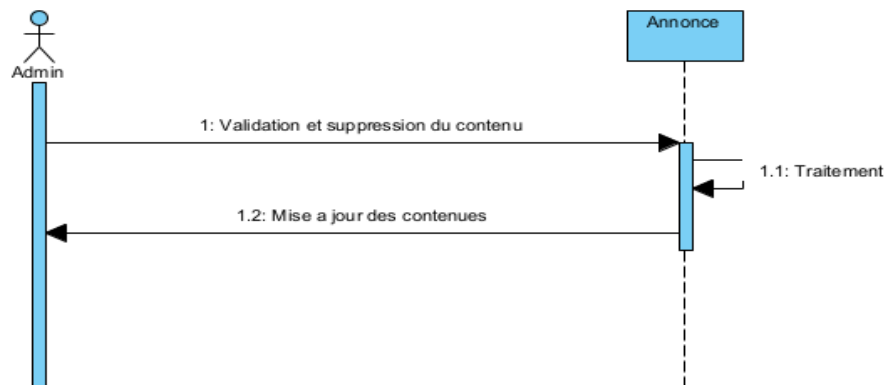


Figure 14 Diagramme de séquence du cas d'utilisation “valider ou supprimer des contenus”

5.5. Spécification des besoins techniques

Les besoins techniques décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et en ce qui concerne notre application, nous avons identifié les besoins suivants :

- La disponibilité : l’application doit être accessible à tous les utilisateurs.
- La fiabilité : les données fournies par l’application doivent être authentiques et crédibles
- La convivialité de l’interface graphique : l’application doit fournir une interface intuitive et facile à prendre en main afin de donner une première impression positive à l’utilisateur et lui permettre de découvrir ses fonctionnalités.
- Une solution ouverte et évoluée : l’application peut être perfectionnée en intégrant d’autres composants afin de garantir une flexibilité et une évolutivité.
- La possibilité de retourner au menu principal depuis n’importe quelle page de l’application.

5.6. Modélisation du domaine

L'élaboration du modèle des classes du domaine permet de passer à une modélisation objet complète.

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classe appelé modèle du domaine. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine de l'application. Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Ces entités ou concepts sont identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine.

Pour établir le diagramme, on procède aux étapes suivantes :

- Identifier les entités ou concepts du domaine ;
- Identifier et ajouter les associations ;
- Organiser et simplifier le modèle en éliminant les classes redondantes et en utilisant l'héritage.

La figure 15 représente le modèle du domaine de l'application.

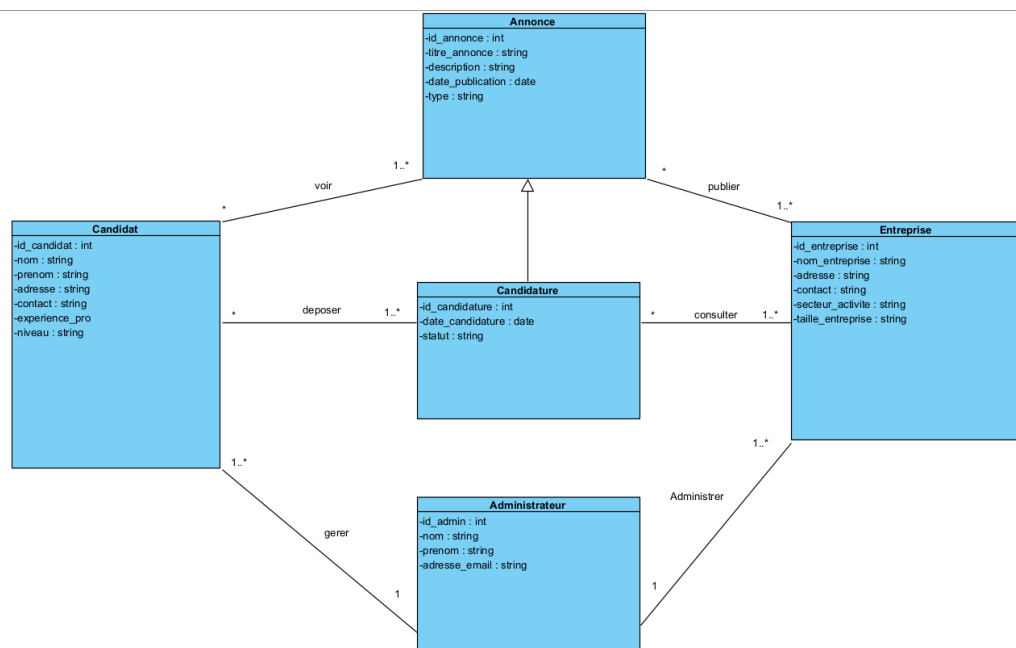


Figure 15: Modèle du domaine de l'application

CHAPITRE 6. Conception détaillée

6.1. Architecture système

L'architecture Modèle-Vue-Contrôleur (MVC) est un pattern architectural qui sépare les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

Ce modèle de conception impose donc une séparation en 3 couches :

- Le modèle : Il représente les données de l'application. Il définit aussi l'interaction avec la base de données et le traitement de ces données. Dans notre cas, le modèle est l'ensemble des classes qui interagissent avec les données.
- La vue : Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle présente tout simplement les résultats renvoyés par le modèle et envoie les événements aux contrôleurs. Les vues sont les fichiers XML dans les ressources layout et menu du projet.
- Le contrôleur : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

La figure 16 illustre le schéma de l'architecture MVC.



Figure 16: Architecture MVC

6.2. Diagramme de séquence de conception pour chaque cas d'utilisation.

Le diagramme de séquence de conception est un diagramme de séquence qui est plus détaillé. En effet, le diagramme de séquence système ne montre que les scénarios d'échange de messages entre les acteurs et le système tandis que le diagramme de séquence de conception permet de donner une vue « en largeur » du déroulement d'une opération en montrant tous les objets impliqués et en donnant plus de détails sur la façon dont cette méthode procède.

La figure 27 illustre le diagramme de séquence de conception du cas d'utilisation « créer un profil »

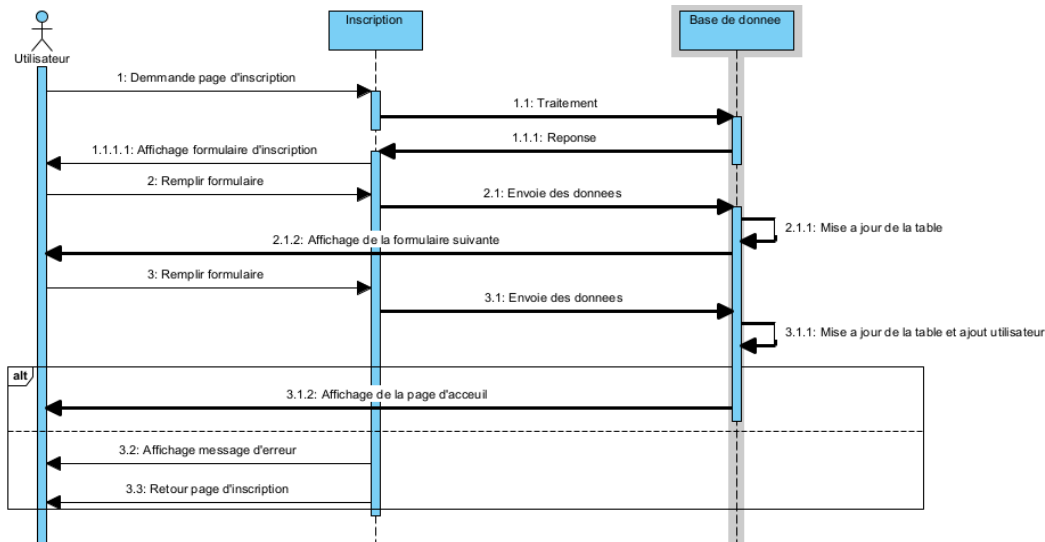


Figure 17 Diagramme de séquence de conception « créer profile »

La figure 18 illustre le diagramme de séquence de conception du cas d'utilisation « s'authentifier »

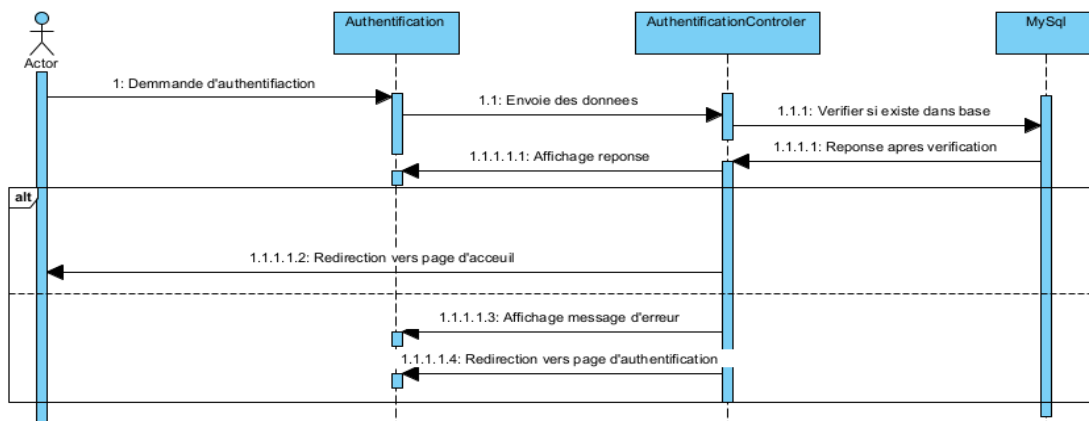


Figure 18 Diagramme de séquence de conception « s'authentifier »

La figure 19 illustre le diagramme de séquence de conception du cas d'utilisation « gérer profile »

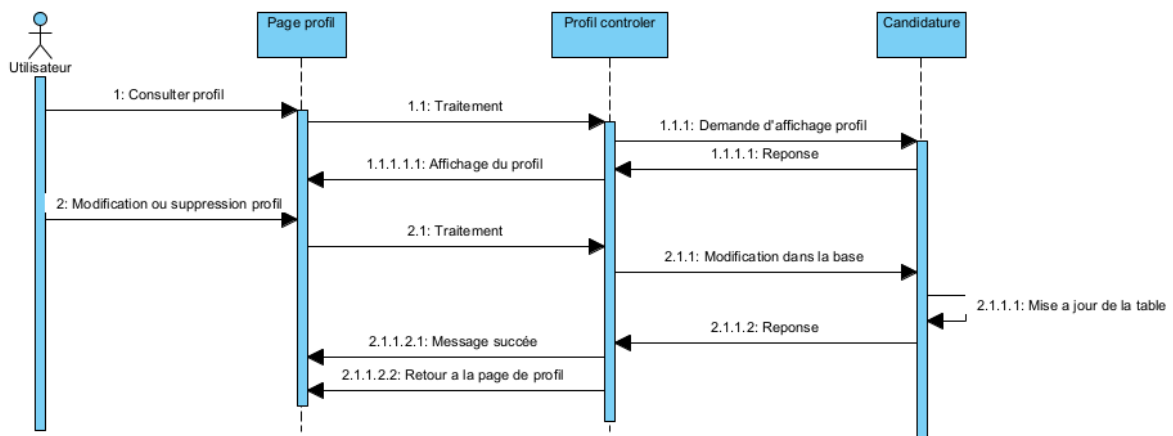


Figure 19: Diagramme de séquence de conception « s'authentifier »

La figure 20 illustre le diagramme de séquence de conception du cas d'utilisation « publier une annonce »

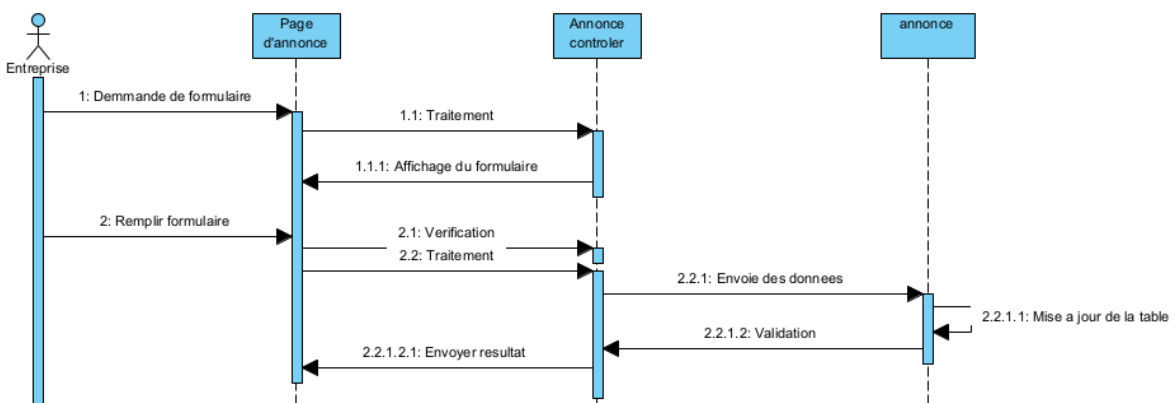


Figure 20: Diagramme de séquence de conception « publier annonce »

La figure 21 illustre le diagramme de séquence de conception du cas d'utilisation « consulter annonce »

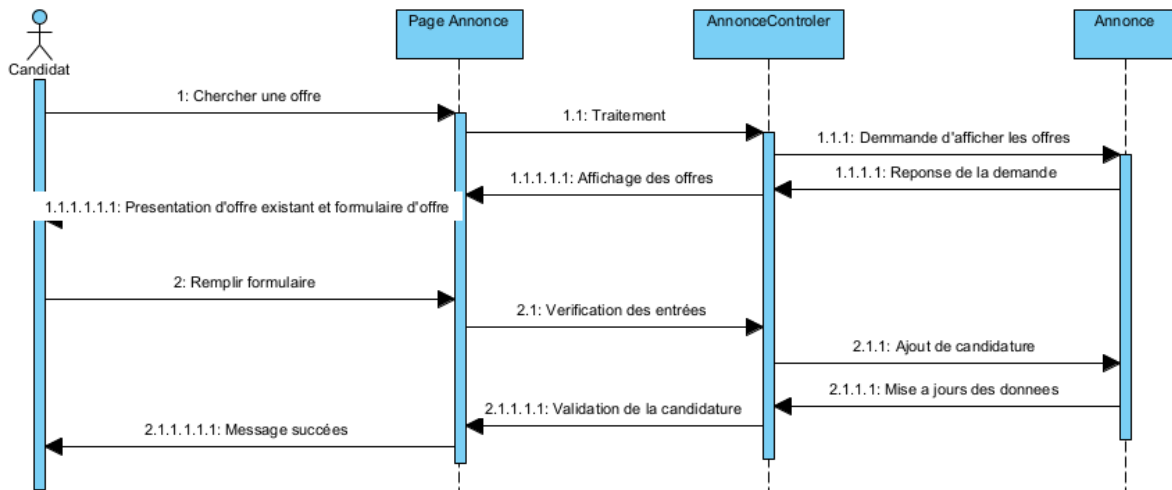


Figure 21 Diagramme de séquence de conception « consulter annonce »

La figure 22 illustre le diagramme de séquence de conception du cas d'utilisation « supprimer annonce »

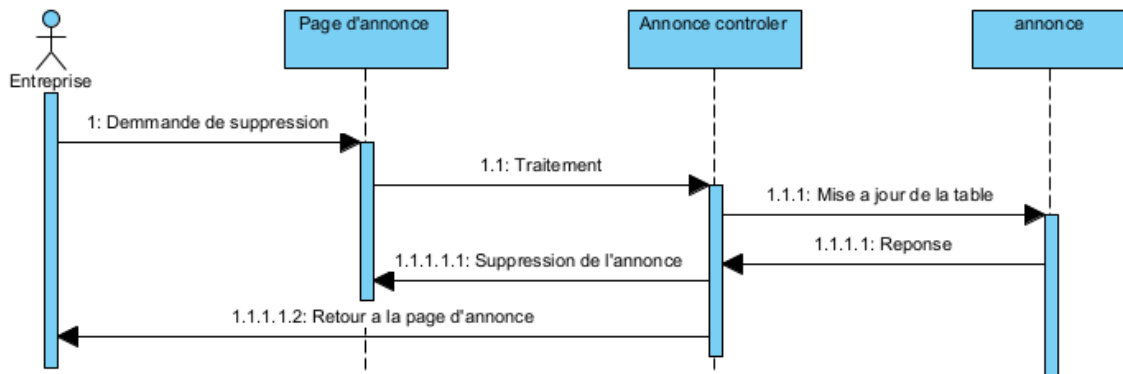


Figure 22: Diagramme de séquence de conception « supprimer annonce »

La figure 23 illustre le diagramme de séquence de conception du cas d'utilisation « valider ou supprimer des contenus »

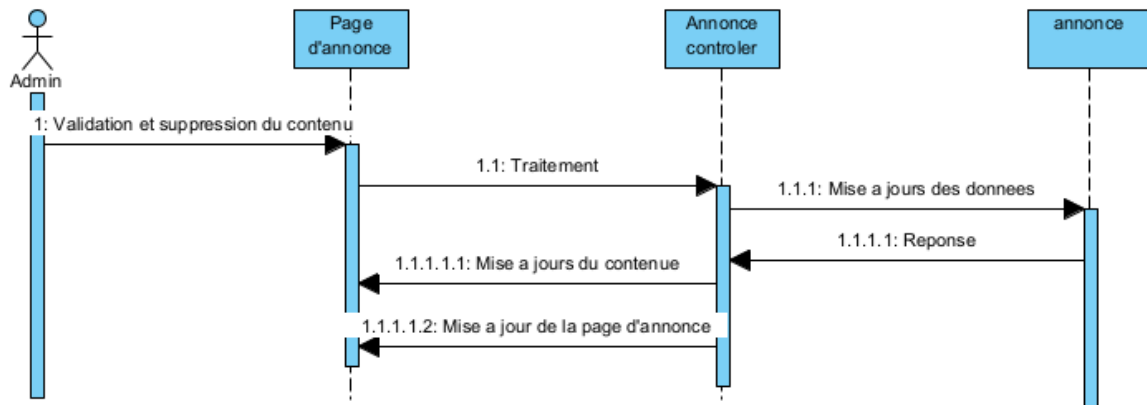


Figure 23: Diagramme de séquence de conception « gérer contenus »

6.3. Diagramme de classe de conception global

Dans cette section, nous allons étudier le diagramme de classe qui est le pivot de l'ensemble de la modélisation du système et permet de donner la représentation statique du système à développer.

6.3.1 Généralité

Le diagramme des classes est la représentation de la structure interne du système. Il identifie la structure des classes de ce système, y compris les propriétés et les méthodes de chaque classe.

Un diagramme des classes permet de :

- Fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation,
- Modéliser les classes du système et leurs relations indépendamment du langage de programmation. Eléments d'un diagramme de classe.

Les éléments d'un diagramme des classes sont les classes et les relations qui les lient.

Classes : Les classes sont les modules de base de la programmation orientée objet. Une classe est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe. La section centrale définit les propriétés de la classe.

Association : Une association est une relation générique entre deux classes. Elle est modélisée par une ligne reliant les deux classes. Cette ligne peut être qualifiée avec le type de relation, et peut également comporter des règles de multiplicité (par exemple un à un, un à plusieurs, plusieurs à

plusieurs) pour la relation.

Composition : Si une classe ne peut pas exister par elle-même, mais doit être un membre d'une autre classe, alors elle possède une relation de composition avec la classe contenant. Une relation de composition est indiquée par une ligne avec un "diamant" rempli.

Dépendance : Quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe. Une relation de dépendance est représentée par une flèche pointillée.

Agrégation : Les agrégations indiquent une relation de contenant contenu. Elle est décrite par une relation « possède ». Une relation d'agrégation est représentée par une ligne avec un « diamant » creux.

Généralisation : Une relation de généralisation est l'équivalent d'une relation d'héritage en terme orienté objet (relation « est-un »). Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe "parent".

La figure 24 montre le diagramme de classe de conception du cas d'utilisation du système.

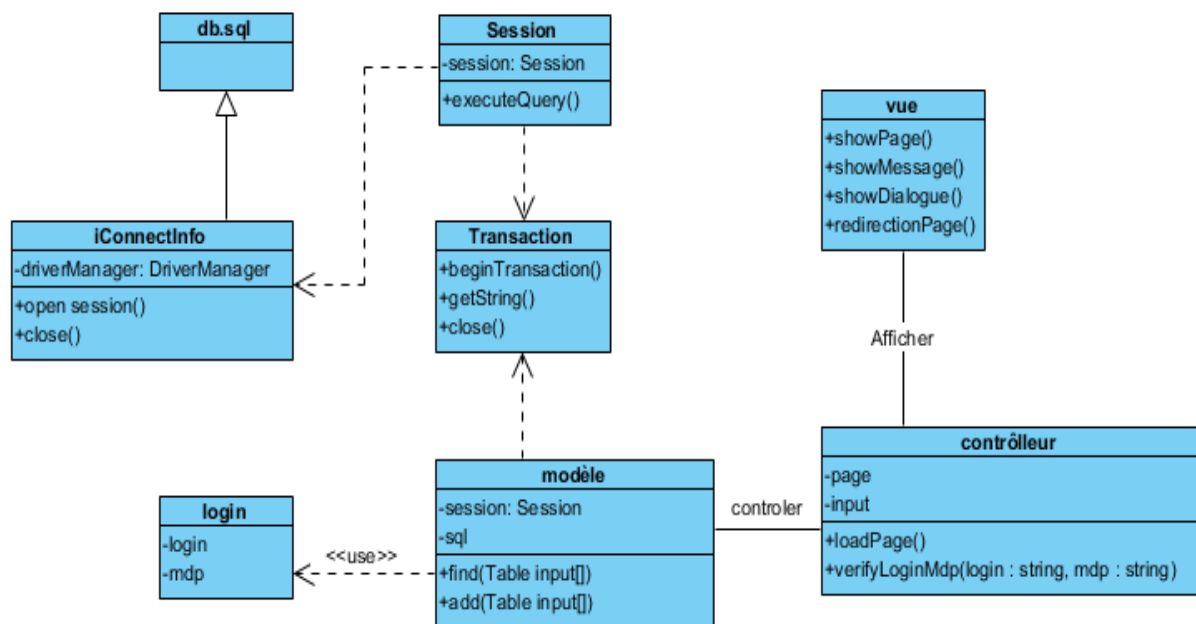


Figure 24: Diagramme de classe de conception du cas d'utilisation

6.3.2 Diagramme de classe de conception globale

Le diagramme de classe de conception représente la structure statique du code, par le biais des attributs et des relations entre classes, le diagramme de classes est le point central dans un développement orienté objet

La figure 25 illustre le diagramme de classe de conception globale du système.

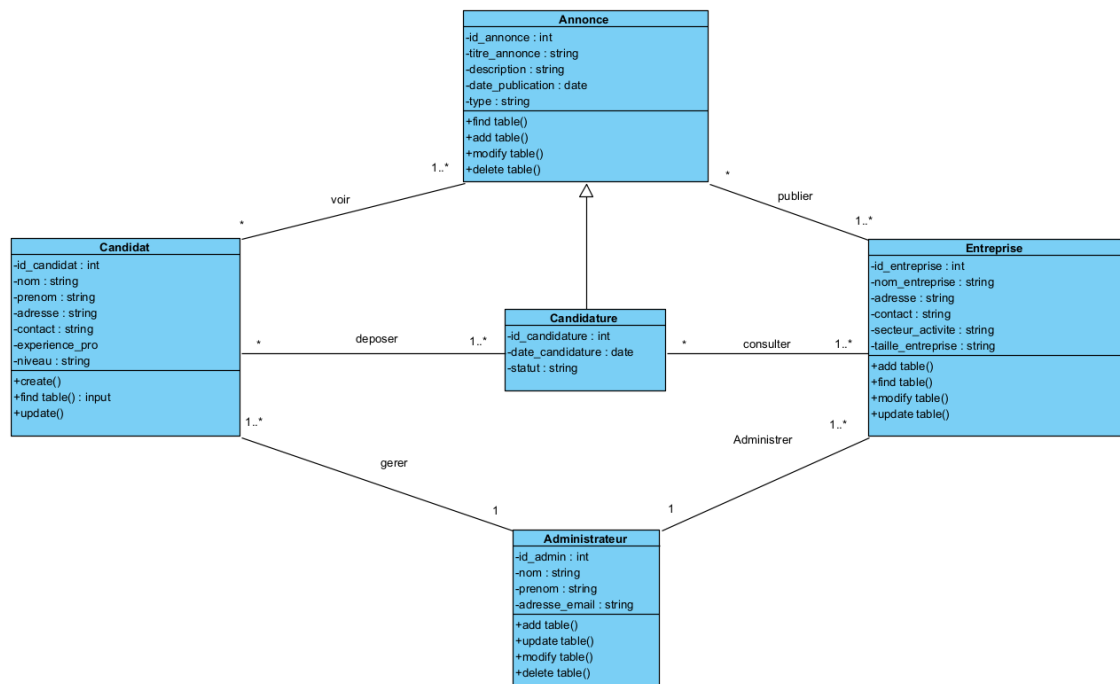


Figure 25: Diagramme de classe de conception globale du système

6.4. Diagramme de paquetage

Un package en UML (ou *paquetage* en français) est un groupe d'éléments, dans le but de les grouper dans des ensembles cohérents. Il est appelé aussi membres, portant sur un sous-ensemble du système.

Le découpage en paquetage doit traduire un découpage logique du système à construire qui corresponde à des espaces de nommage homogènes.

Un package peut contenir la plupart des éléments UML : classes, objets, cas d'utilisations, composantes, etc. Il peut également contenir des packages, créant une hiérarchie complète.

L'avantage des packages est qu'ils permettent de structurer les diagrammes et de donner une vision globale plus claire. La figure 26 représente le diagramme de paquetage.

paramètres d'exécution d'un artefact déployé sur un nœud.

Le diagramme de déploiement du système est alors représenté sur la figure 27.

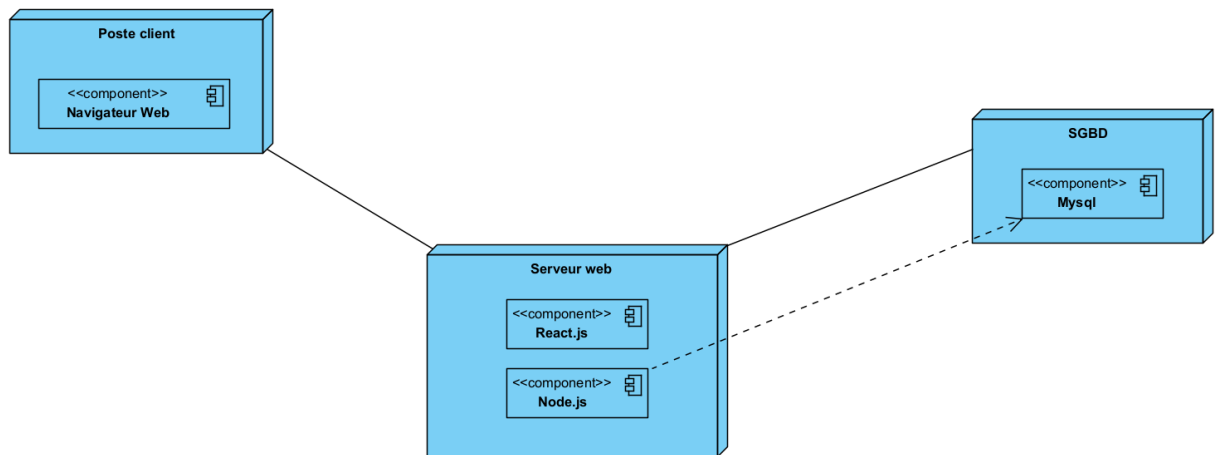


Figure 27: Diagramme de déploiement

PARTIE III. REALISATION

CHAPITRE 7. Mise en place de l'environnement du développement

Dans ce chapitre, nous allons détailler la mise en place des moyens logiciels utilisés lors de la réalisation du projet.

7.1. Installation et configuration des outils

Comme nous avons déjà proposé la solution adéquate pour le développement de notre application pour mener à bien notre projet, nous devons installer les outils nécessaires suivants :

- WampServer pour un serveur.
- Node js et React js
- Visual Studio Code comme éditeur de texte.
- Google Chrome comme navigateur
- VISUAL PARADIGM 10.0 utilisé pour la conception et modélisation UML.

7.1.1 Installation de WampServer

WAMP est un acronyme informatique signifiant : Windows Apache MYSQL PHP. Son avantage réside dans le fait que le SGBD MySQL, le serveur Apache et PHP y sont déjà intégrés. Les rôles de ces quatre composants sont les suivants :

- Apache est le serveur web « frontal » : il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur).
- MySQL stocke toutes les données de l'application.
- Windows assure l'attribution des ressources à ces trois composants.

La figure 28 représente l'installation du WampServer.

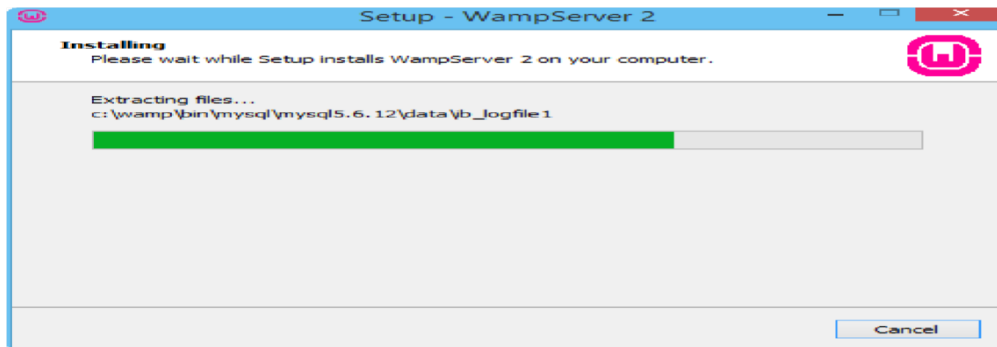


Figure 28: Installation de WampServer

Après avoir exécuté quelques instructions et saisi quelques paramètres, l'installation se poursuit toute seule et s'achève.

Une fois installé, on va pouvoir créer notre base de données ainsi que les tables qui servent de stockage de différentes données nécessaires pour l'application dans PhpMyAdmin à l'adresse <http://localhost/phpmyadmin>

7.1.2 Installation NodeJs

Figure 29 représente l'installation de node.js.

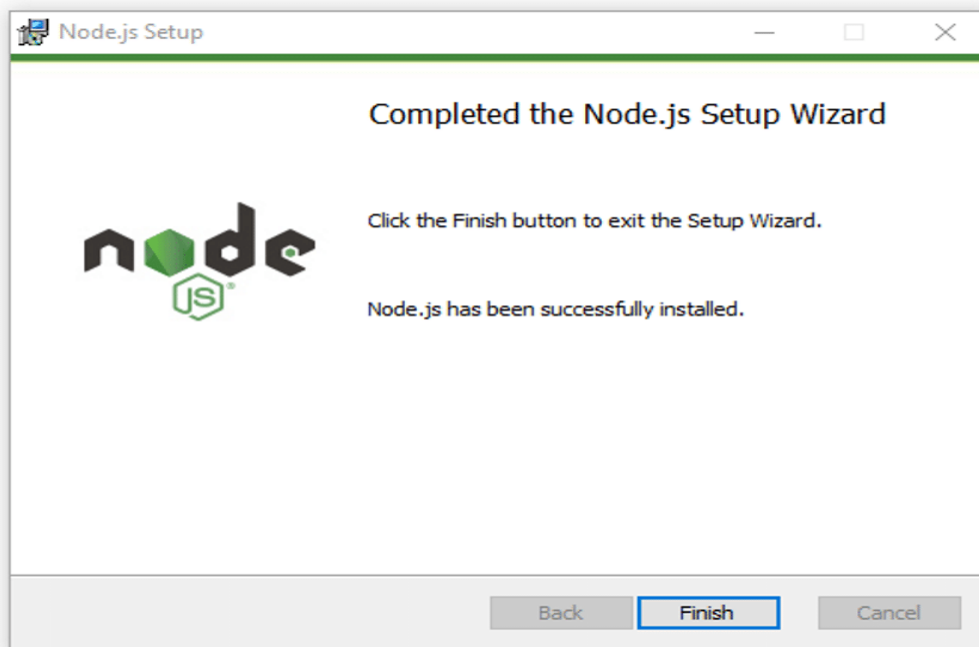


Figure 29: installation de Node.js

Une fois installé, on va pouvoir démarrer notre serveur d'application en choisissant le port d'écoute de notre application.

7.1.3 Installation de Visual Studio Code

Pour installer l'éditeur Visual Studio Code, il faut aller dans le site officiel, dans l'option download et choisir selon le système d'exploitation utilisé. Après l'installation, on n'a pas besoin plus de configuration, mais on peut ajouter des « plugins » pour faciliter son utilisation. La figure 30 présente l'interface de l'IDE Visual Studio Code.

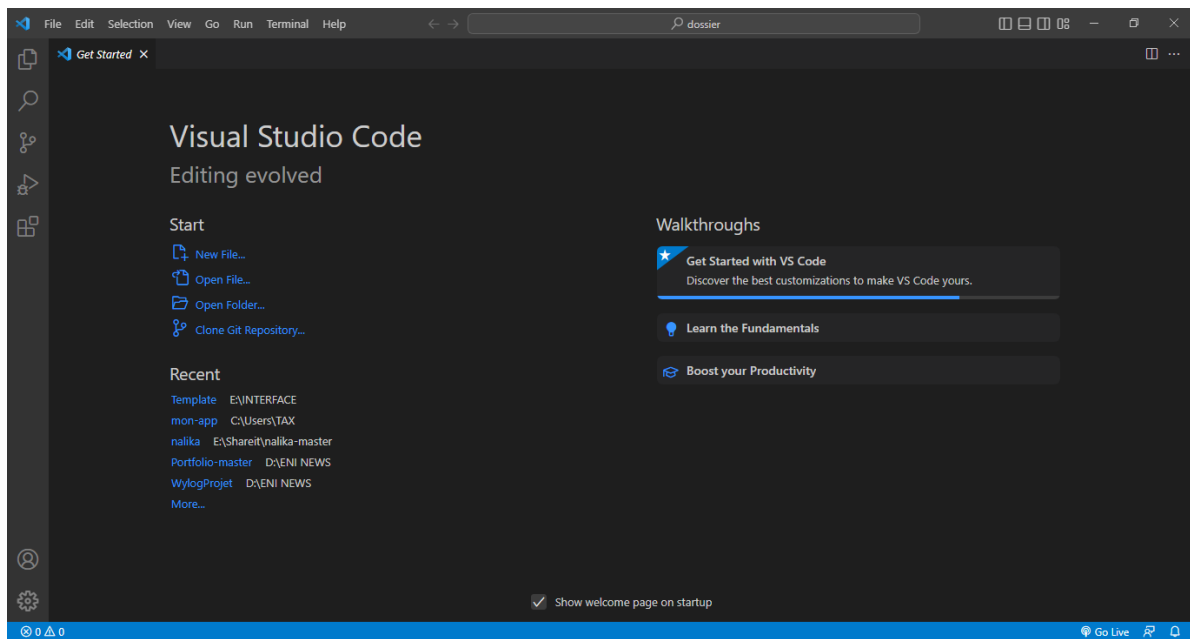


Figure 30: Interface de Visual Studio Code

7.1.4 Installation de Google Chrome

Il faut télécharger le programme d'installation Google Chrome sur son site officiel. Ensuite, exécuter le fichier téléchargé et l'installation se poursuit automatiquement en téléchargeant les fichiers nécessaires pour le bon fonctionnement de Google Chrome.

La figure 31 représente l'interface de Google Chrome.

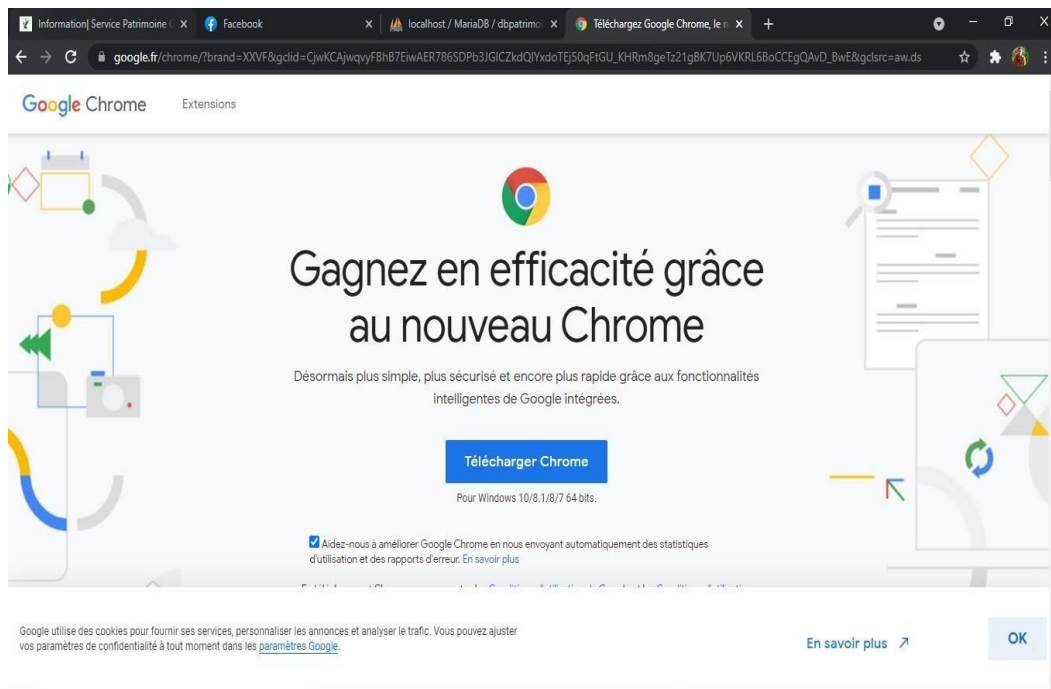


Figure 31: Interface de Google Chrome

7.1.5 Installation de Visual Paradigm

Pour l'installation de Visual Paradigm, il faut télécharger l'exécutable et une fois téléchargé, on exécute le fichier téléchargé. Une fois lancé, l'installation va s'effectuer toute seule et l'outil de conception Visual Paradigm est prêt pour notre conception.

La figure 32 montre un aperçu de l'interface de Visual Paradigm.

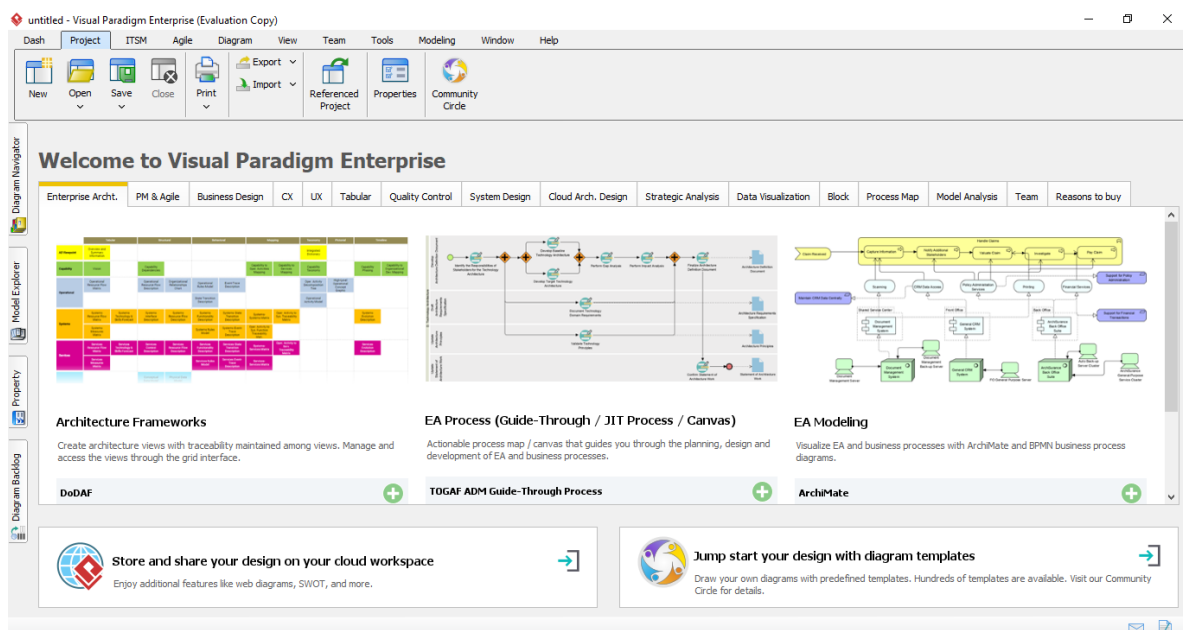


Figure 32: Interface de Visual Paradigme

7.2. Architecture de l'application

L'application se base sur l'architecture à trois niveaux ou 3-tiers composée de serveur de base de données, de serveur d'application, et de poste client, illustrée par la figure 33.

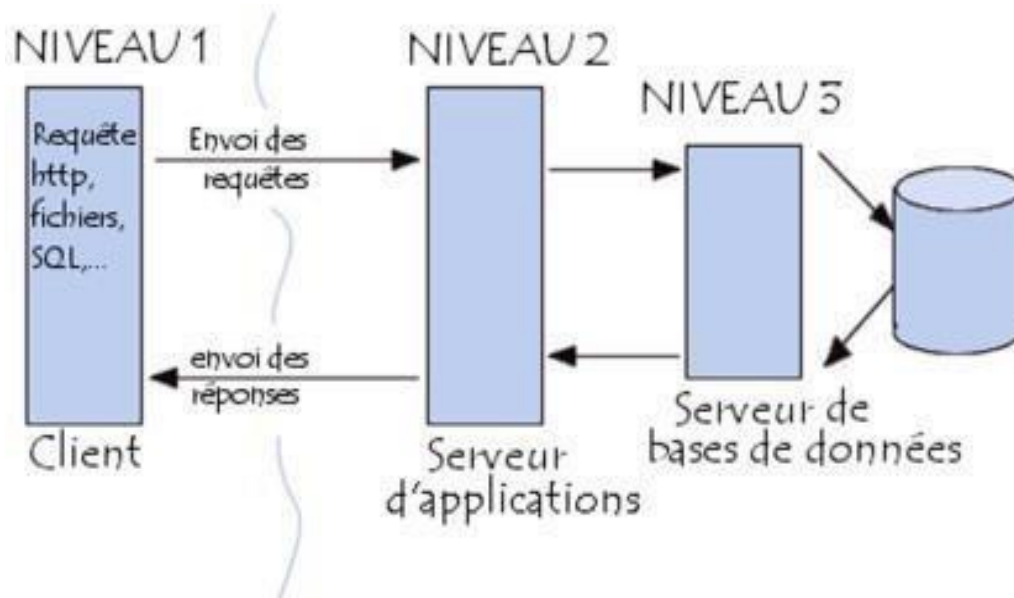


Figure 33: Architecture à trois tiers

- Le client : c'est le demandeur de ressources, équipé d'une interface utilisateur chargée de la présentation ;
 - Le serveur d'application : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur.
 - Le serveur de bases de données : fournissant au serveur d'application les données dont il a besoin.
- Les avantages de l'architecture 3-tiers sont principalement au nombre de quatre :
- Les requêtes clients vers le serveur sont d'une plus grande flexibilité que dans celles de l'architecture 2-tiers basées sur le langage SQL ;
 - Cette flexibilité permet à une entreprise d'envisager dans le cadre d'une architecture 3-tiers une grande souplesse pour l'introduction de toutes nouvelles technologies ;
 - D'un point de vue développement, la séparation qui existe entre le client, le serveur et le SGBD permet une spécialisation des développeurs sur chaque tiers de l'architecture ;
 - Plus de flexibilité dans l'allocation des ressources ; la portabilité du tiers serveur permet d'envisager une allocation et ou modification dynamique au gré des besoins évolutifs au sein d'une entreprise.

CHAPITRE 8. Développement de l'application

Ce chapitre consiste à la création de la base de données et au codage de l'application.

8.1. Création de la base de données

La figure 34 nous montre une portion de code de la création des tables de notre application.

```
1 CREATE DATABASE madapro;  
2 USE madapro;  
3  
4 CREATE TABLE users (  
5     id INT AUTO_INCREMENT PRIMARY KEY,  
6     name VARCHAR(255) NOT NULL,  
7     email VARCHAR(255) UNIQUE NOT NULL,  
8     password VARCHAR(255) NOT NULL,  
9     role ENUM('candidate', 'company') NOT NULL,  
10    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
11 );  
12 |
```

Figure 34 création des tables

La figure 35 nous montre la structure de notre base de données dans phpMyAdmin

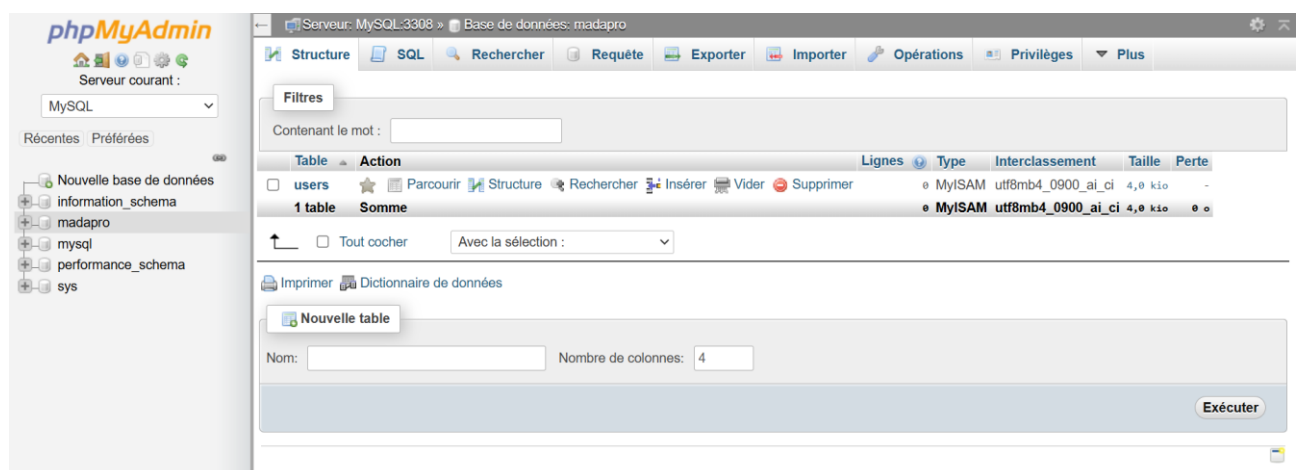


Figure 35 création des tables

8.2. Codage de l'application (capture du code et description)

Dans cette partie, on va présenter successivement des extraits des codes de l'application selon le Framework React (MVC).

Modèle :

```
src > components > Login.tsx > Login > handleLogin
1  import { useState } from "react";
2  import { useNavigate } from "react-router-dom";
3
4  export function Login() {
5    const [email, setEmail] = useState("");
6    const [password, setPassword] = useState("");
7    const [role] = useState("candidate"); // "candidate" ou "company"
8    const navigate = useNavigate();
9
10   const handleLogin = async (e: { preventDefault: () => void }) => {
11     e.preventDefault();
12
13     try {
14       const response = await fetch("http://localhost:8286/api/auth/login", {
15         method: "POST",
16         headers: {
17           "Content-Type": "application/json",
18         },
19         body: JSON.stringify({ email, password, role }),
20       });
21
22       const data = await response.json();
23       if (response.ok) {
24         alert(`Bienvenue, ${data.user.name}`);
25
26         // Stocker le token ou rediriger
27         localStorage.setItem("token", data.token);
28         localStorage.setItem("role", data.user.role);
29
30         // Rediriger selon le rôle
31         if (data.user.role === "candidate") {
32           navigate("/candidate/dashboard");
33         } else if (data.user.role === "company") {
34           navigate("/company/dashboard");
35         }
36       } else {
37         alert(data.message || "Erreur de connexion");
38       }
39     } catch (error) {
40       console.error("Erreur lors de la connexion :", error);
41     }
42   };
43
44   return (
45
```

Figure 36 Codage de l'application pour la partie Modèle

Controller :

```
JS login.js > ...
1  const express = require('express');
2  const bcrypt = require('bcryptjs');
3  const jwt = require('jsonwebtoken');
4  const db = require('./db');
5  const router = express.Router();
6
7  // Route pour l'enregistrement d'utilisateur
8  router.post('/register', (req, res) => {
9      const { name, email, password, role } = req.body;
10
11      if (!name || !email || !password || !role) {
12          return res.status(400).json({ message: 'Tous les champs sont requis' });
13      }
14
15      db.query('SELECT * FROM users WHERE email = ?', [email], (err, results) => {
16          if (err) return res.status(500).json({ message: 'Erreur de serveur' });
17          if (results.length > 0) return res.status(400).json({ message: 'Email déjà utilisé' });
18
19          const hashedPassword = bcrypt.hashSync(password, 10);
20
21          db.query(
22              'INSERT INTO users (name, email, password, role) VALUES (?, ?, ?, ?)',
23              [name, email, hashedPassword, role],
24              (err) => {
25                  if (err) return res.status(500).json({ message: 'Erreur lors de l\'enregistrement' });
26                  res.status(201).json({ message: 'Utilisateur enregistré avec succès' });
27              }
28          );
29      });
30  });
31
```

Figure 37 Codage de l'application pour la partie Controleur

8.3. Présentation de l'application

Dans ce qui suit, nous allons présenter quelques interfaces de notre application

8.3.1 Page d'Accueil

La figure 38 décrit la page d'accueil.

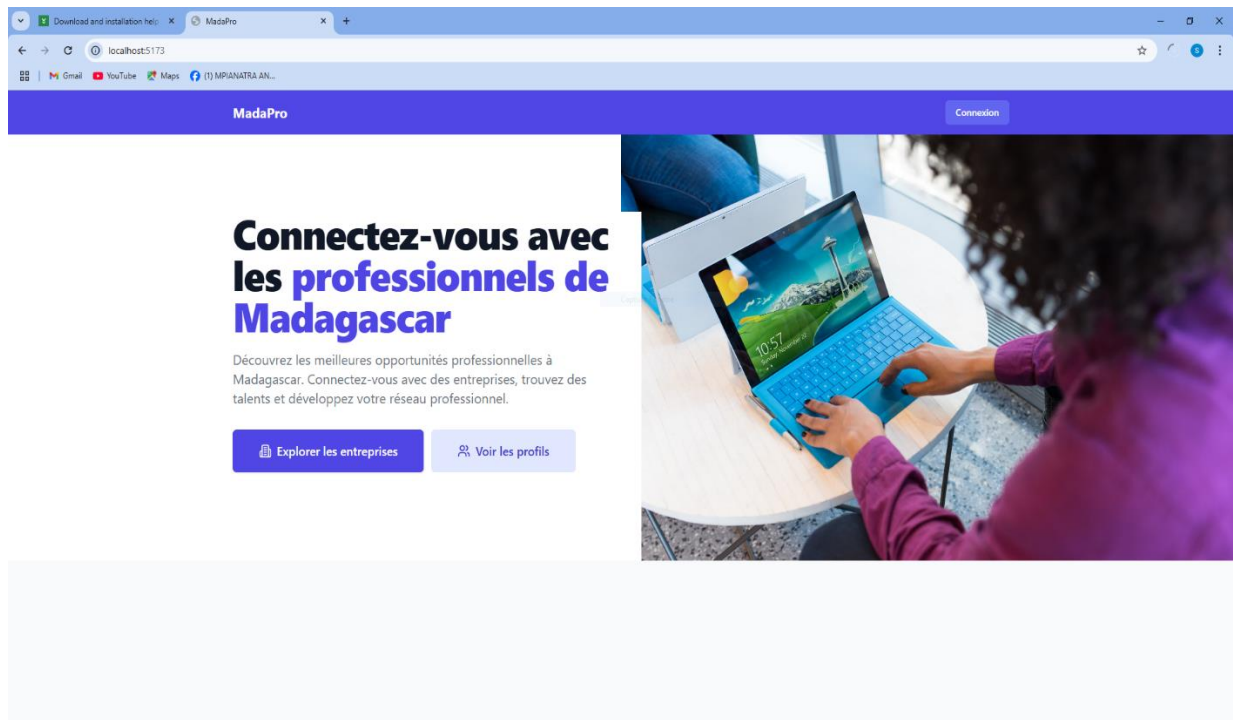


Figure 38 Page d'accueil

8.3.2 Page d'Authentification

La figure 39 nous montre la page d'authentification.

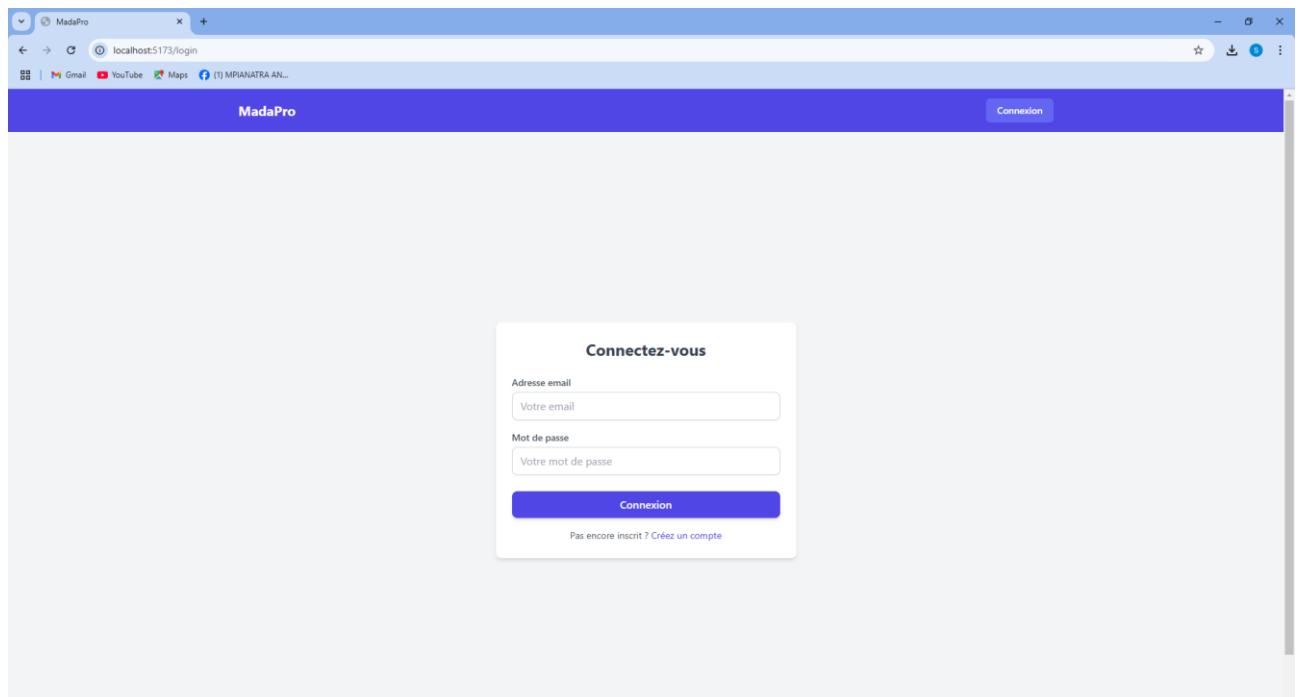


Figure 39 page d'authentification

Conclusion

Ce projet m'a permis d'avancer dans le monde de programmation grâce à la découverte des nouvelles instructions qu'on voit dans plusieurs documents car la programmation ne peut se limiter uniquement à l'enseignement à l'école mais aussi par les recherches personnelles.

Pour concevoir cette application, on a utilisé le langage de modélisation UML en adoptant la méthode 2TUP pour le planning et le recueil des besoins. L'utilisation du langage de programmation Javascript avec l'IDE Visual Studio Code et Framework Node.js et React.js MVC gérés par Windows 10 nous a permis de réaliser ce logiciel.

L'application permet de gérer les problèmes entre la recherche de travail, pour les jeunes talents et aussi la recherche de travailleurs pour les recruteurs. L'application est exploitable par les services dédiés à son usage même s'il est encore en développement. En perspective, l'amélioration dans le futur est toujours envisageable en ajoutant certaines fonctionnalités et c'est le cas, comme l'ajout d'un autre acteur, visiteurs.

Ce stage nous a permis de comprendre et vivre dans le domaine professionnel de l'entreprise avec un objectif précis qui est d'approfondir mes connaissances pratiques en matière de développement d'application web.

Bibliographie

- [1] Modéliser une application web -Pascal Roques, 2008, 4ème Edition, [EYROLLES] UML2
- [2] La route vers React –Robin Wieruch, Edition 2022, 281 pages.
- [3] Développez le Front End de vos applications web et mobiles avec Javascript – Sébastien CASTIEL, 2020, Edition Eni, 322 pages.
- [4] Nodejs et Javascript-Alexandre Brilliant, Julien Fontanet, Olivier Lambert, 2016, Edition Eni, 597 pages.

Webographie

- [5]<https://www.futurama-sciences.com/tech/definitions/informatique-uml-3979/>, consulté en Septembre 2022
- [6]<https://www.visual-paradigm.com/download/>, consulté en Octobre 2022
- [7]<https://fr.react.org/>, consulté en Octobre 2022
- [8] <https://nodejs.org/en/download>, consulté en Octobre 2022
- [9] <https://www.stackoverflow.com/>, [site](#) officielle de stackoverflow , consulté tout au long du projet
- [10]<https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml>, Cours UML de OpenClassrooms, consulté en Novembre 2022.

Glossaire

API : est un ensemble de définitions de sous-programmes, de protocoles de communication et d'outils permettant de créer un logiciel.

Candidat : un employé qui se postule à une offre de stage ou de travail.

Compétence : spécialité de l'employé.

Annonce : publication d'offre de stage ou de travail.

Candidature : Action de postuler à une offre.

Offre : offre de travail dans une société.

Annexes

Table des matières

| | |
|--|------|
| CURRICULUM VITAE | I |
| SOMMAIRE | III |
| REMERCIEMENTS | VI |
| LISTES DES FIGURES..... | VII |
| LISTES DES TABLEAUX..... | VIII |
| NOMENCLATURE..... | IX |
| INTRODUCTION GENERALE..... | 1 |
| PARTIE I. PRESENTATIONS | 2 |
| CHAPITRE 1. Présentation de l'Ecole Nationale d'Informatique..... | 3 |
| 1.1. Information d'ordre général | 3 |
| 1.2. Missions et historiques..... | 3 |
| 1.3. Organigramme institutionnel de l'ENI | 5 |
| 1.4. Domaine de spécialisation | 6 |
| 1.5. Architecture des formations pédagogiques | 7 |
| 1.6. Relations de l'ENI avec les entreprises et les organismes | 8 |
| 1.7. Débouchés professionnels avec les diplômés | 9 |
| 1.8. Ressources Humaines | 10 |
| CHAPITRE 2. Présentation de l'UN-IT | 12 |
| 2.1. Localisation et contacts | 12 |
| 2.2. Description et Statuts | 12 |
| 2.3. Activités | 12 |
| 2.4. Organigramme | 14 |
| CHAPITRE 3. Description du projet..... | 15 |
| 3.1. Formulation..... | 15 |
| 3.2. Objectif et besoins de l'utilisateurs..... | 15 |
| 3.3. Moyens nécessaires à la réalisation du projet | 16 |
| 3.4. Résultats attendus..... | 17 |
| 3.5. Chronogramme de travail | 17 |
| PARTIE II. ANALYSE ET CONCEPTION | 18 |
| CHAPITRE 4. Analyse préalable | 19 |
| 4.1. Analyse de l'existant..... | 19 |
| 4.1.1. Organisation actuelle | 19 |
| 4.1.2. Inventaire des moyens matériels et logiciels | 19 |

| | | |
|-----------------|--|----|
| 4.2. | Critique de l'existant..... | 20 |
| 4.3. | Conception avant-projet..... | 20 |
| 4.3.1. | Propositions des solutions..... | 20 |
| 4.3.2. | Méthodes de conception et outils utilisés | 21 |
| CHAPITRE 5. | Analyse conceptuelle | 28 |
| 5.1. | Présentation de la méthode utilisée..... | 28 |
| 5.2. | Dictionnaire des données | 30 |
| 5.3. | Règle de gestion..... | 32 |
| 5.4. | Représentation et spécification des besoins | 32 |
| 5.4.1. | Diagramme des cas d'utilisation | 32 |
| 5.4.2. | Description textuelle pour chaque cas d'utilisation..... | 37 |
| 5.4.3. | Priorisation des cas d'utilisation..... | 43 |
| 5.4.4. | Diagramme de séquence système pour chaque cas d'utilisation | 43 |
| 5.5. | Spécification des besoins techniques | 48 |
| 5.6. | Modélisation du domaine..... | 49 |
| CHAPITRE 6. | Conception détaillée..... | 50 |
| 6.1. | Architecture système..... | 50 |
| 6.2. | Diagramme de séquence de conception pour chaque cas d'utilisation..... | 51 |
| 6.3. | Diagramme de classe de conception global | 54 |
| 6.3.1 | Généralité..... | 54 |
| 6.4. | Diagramme de paquetage..... | 56 |
| 6.5. | Diagramme de déploiement | 57 |
| PARTIE III. | REALISATION..... | 59 |
| CHAPITRE 7. | Mise en place de l'environnement du développement..... | 60 |
| 7.1. | Installation et configuration des outils | 60 |
| 7.1.2 | Installation NodeJs | 61 |
| | | 61 |
| 7.1.3 | Installation de Visual Studio Code..... | 62 |
| 7.1.4 | Installation de Google Chrome | 62 |
| 7.1.5 | Installation de Visual Paradigm | 63 |
| 7.2. | Architecture de l'application..... | 64 |
| CHAPITRE 8. | Développement de l'application | 65 |
| 8.1. | Création de la base de données | 65 |
| 8.2. | Codage de l'application (capture du code et description)..... | 66 |
| 8.3. | Présentation de l'application..... | 67 |
| Conclusion..... | | 69 |

| | |
|--------------------------|------|
| Bibliographie | VIII |
| Webographie | IX |
| Glossaire | X |
| Annexes | XI |
| Table des matières | XII |
| Résumé | XV |
| Abstract | XV |

Résumé

Dans ce projet, nous avons conçu et développé une plateforme web mise en relation professionnelle pour Madagascar. Ce projet a visé de créer une plateforme web interactive permettant de lister toutes les entreprises et agences par région de Madagascar en fonction de leur activité, du petit au grand, ainsi que de fournir une base de données des profils de personnel par activité et catégorie.

La méthode de gestion de projet 2TUP a été adoptée, avec une modélisation UML pour la structure de l'application. Les technologies utilisées sont JavaScript incluant son Framework NodeJS, MySQL pour la gestion de base de données, et ReactJs pour dynamiser la page. Visual Studio Code comme outil de développement.

L'application est désormais opérationnelle et améliore l'efficacité de la recherche d'emploi et stage, tout en restant flexible pour des optimisations futures.

Mots-Clés : Annonce, Candidat, Entreprise, Région.

Abstract

In this project, we designed and developed a professional networking web platform for Madagascar. This project aimed to create an interactive web platform to list all companies and agencies by region of Madagascar according to their activity, from small to large, as well as to provide a database of staff profiles by activity and category.

The 2TUP project management method was adopted, with UML modeling for the structure of the application. The technologies used are JavaScript including its NodeJs framework, MySQL for database management, and ReactJs to boost the page. Visual Studio code as a development tool.

The application is now operational and improves the efficiency of job and internship searches, while remaining flexible for future optimizations.

Keywords: Announcement, Candidate, Company, Region.