

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira – Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



EN VUE DE L'OBTENTION DU DIPLOME DE MASTER PROFESSIONNEL EN
INFORMATIQUE ET D'UN DIPLOME DANS PROJET D'ENTREPRISE
ECONOMIQUE DANS LE CADRE DE L'ARRETÉ MINISTÉRIEL N°1275

Option :Génie Logiciel

Conception et développement d'une plateforme de Micro-jobbing dédiée aux étudiants

Réalisé par :

- Djoudad LOUIZA
- Djermoune CELINA
- Djebara LIZA

Encadrante :

- Dr EL BOUHISSI HOUDA

Mémoire soutenu le 29/09/2025 devant le jury composé de :

- **Président :** Dr CHIBANI Samia MCA
- **Examinatrice :** Dr AIT HACENE Souhila MCB
- **Représentant du l'incubateur :** Dr CHAFI Zahia Epse SOUCI MCA
- **Représentant du secteur socio-économique :** KHIREDDINE Rahim

Année Universitaire : **2024-2025**

Dédicace

À mes parents, Merci pour votre amour infini, votre patience et votre sacrifices silencieux, votre soutien sans faille, Merci pour la confiance que vous m'avez accordée et pour votre présence à chaque étape. Maman et Papa, vous êtes mes racines, ma force et mon refuge, et ce travail vous appartient autant qu'à moi.

À ma grande sœur Fahima et ses deux petites étoiles, Nourhane et Nylia, merci pour votre douceur, vos encouragements et les moments précieux partagés.

À mes frères Ikhlef et sa femme Tiziri, Fares et sa femme Roza, et Menad, le petit chouchou de la maison, Je vous suis infiniment reconnaissante pour tout ce que vous avez fait pour moi. Merci pour votre complicité, vos conseils et vos sourires, vos gestes, vos mots et votre soutien.

À toute ma famille, mon oncle et sa femme, mes cousins et cousines, Merci Pour les encouragements et votre bonté que vous m'avez accordée, merci pour votre présence et vos mots réconfortants tout au long de ce parcours.

À la mémoire de mes chères grand-mères et grands-pères Bien que vous ne soyez plus parmi nous, votre amour et les prières que vous m'avez transmises continuent de m'accompagner chaque jour.

À mes deux binômes Louiza et Liza , Partenaires de ce projet, Merci pour votre engagement, votre travail acharné et votre esprit d'équipe qui ont rendu cette expérience inoubliable.

À mes amis Pour nos sourires et vos mots réconfortants dans les moments de doute, **enfin,** à une personne très chère, présente dans chaque réussite et chaque pas que j'ai franchi, merci pour la lumière que tu apportes à ma vie.

À vous tous, je dédie ce travail avec toute ma gratitude et mon affection. Ce travail est le fruit d'un long chemin, mais il porte en lui un peu de vous à chaque page.

Celina

Dédicace

Ce mémoire, bien qu'il porte ma signature, est le fruit d'un chemin parcouru à plusieurs.

À travers ces lignes, je souhaite exprimer ma profonde gratitude à celles et ceux qui, de près ou de loin, ont contribué à cette étape importante de ma vie.

Avant toute chose, je rends d'abord hommage à mon papa et ma maman, pour leur amour inconditionnel, leur patience infinie et leurs sacrifices silencieux.

Vous êtes mon ancre et ma lumière, ceux vers qui mon cœur revient toujours.

À mes deux sœurs, Liza et Elina, complices de tous les instants, gardiennes de mes fous rires et témoins de mes doutes.
Votre présence a été mon refuge et ma force.

À mon frère Samir, pour sa présence constante, ses mots simples mais vrais, et cette manière unique de veiller sur moi, sans bruit mais avec force.

À Celina et Liza, plus que des coéquipières, de véritables amies avec qui j'ai partagé rires, défis, souvenirs inoubliables et chaque étape de ce parcours.
Merci pour votre complicité, votre bienveillance bâtie au fil des mois, et pour avoir rendu ce PFE aussi riche humainement que professionnellement.

Ce travail est le fruit d'un long chemin, mais il porte en lui un peu de vous à chaque page.

Louiza

Dédicace

Ce mémoire représente bien plus qu'un travail académique, il est le résultat d'un chemin marqué par l'appui, l'affection et la confiance de plusieurs personnes qui m'ont accompagnée.

À travers ces lignes, je souhaite exprimer toute ma gratitude envers celles et ceux qui ont contribué, de près ou de loin, à cette étape importante de ma vie.

Je rends avant tout hommage à mon papa et à ma maman, pour la tendresse qu'ils m'ont toujours donnée, leurs sacrifices silencieux et leur patience infinie.

Vous êtes ma force et mon repère, ceux vers qui je me tourne toujours avec confiance.

À mon frère Aymen, pour ton soutien indéfectible, ta présence réconfortante et cette façon unique de me motiver avec simplicité et sincérité.

À mes deux binômes Celina et Louiza, dont la complicité et l'engagement ont rendu ce parcours plus léger et plus riche.

Ce mémoire est aussi le vôtre, car il est marqué par votre appui et votre inspiration.

Liza

Remerciements

Ce jour symbolise bien plus que la fin d'un cycle universitaire : il marque la clôture d'un chapitre précieux de notre vie, écrit au fil des années à l'Université Abderrahmane Mira de Béjaïa.

Un chapitre rempli de découvertes, d'efforts, de rêves partagés et de liens tissés avec passion. C'est avec une profonde émotion et une immense gratitude que nous tournons cette page, le cœur chargé de souvenirs et d'espoirs pour l'avenir.

Tout d'abord, nos remerciements s'adressent au bon Dieu le tout puissant et miséricordieux, qui nous a donné la force, la volonté et la patience pour accomplir ce modeste travail.

À nos chers parents et nos familles, qui nous ont toujours encouragé et soutenu durant toute la période de nos études jusqu'à atteindre ce stade de notre formation en général et notre projet de fin d'étude en particulier.

Nous souhaitons également exprimer notre profonde gratitude envers notre encadrante, Mme EL BOUHISSI Houda, pour son suivi attentif, ses précieux conseils et sa disponibilité constante, tout au long de notre travail. Sa contribution a été d'une grande valeur, et nous lui témoignons toute notre reconnaissance.

Nous remercions tout particulièrement les membres de jury pour avoir accepté l'évaluation de notre travail.

Nos sincères remerciements s'adressent à tous les enseignants du département d'informatique, pour leur précieuse contribution à notre apprentissage.

Enfin, nous tenons à exprimer notre reconnaissance à toutes celles et ceux qui, directement ou indirectement, ont apporté leur soutien à la réalisation de ce travail.

Celina, Louiza, Liza

Résumé

Ce mémoire s'inscrit dans le cadre d'un projet de fin de cycle de Master en Informatique. Il porte sur la conception et la réalisation d'une plateforme numérique baptisée JobLynk, destinée à faciliter la mise en relation entre étudiants à la recherche de micro-jobs et particuliers, petites entreprises ou recruteurs ayant des besoins ponctuels.

La plateforme offre la possibilité aux étudiants de créer et de valoriser leur profil, de rechercher des missions en fonction de leurs compétences, de leur localisation et de leur disponibilité, tandis que les recruteurs peuvent publier des offres et consulter des profils adaptés.

Le développement du projet s'est appuyé sur la méthodologie UP (Unified Process), permettant un cycle de vie itératif et incrémental, intégrant à chaque phase l'analyse, la conception, l'implémentation et les tests. La solution a été développée en utilisant des technologies modernes du développement web, à savoir : Node.js et Express.js pour le backend, React.js pour le frontend, et PostgreSQL pour la gestion des données associé à TypeORM pour simplifier les interactions avec la base.

La plateforme répond efficacement aux besoins fonctionnels définis, en facilitant la recherche et la consultation de missions, tout en respectant les exigences non fonctionnelles telles que la sécurité, la fiabilité et la facilité d'utilisation.

Mots clés : Micro-jobs, Étudiants, Recruteurs, Emploi ponctuel, JobLynk, UP, UML.

Abstract

This thesis is part of a Master's final project in Computer Science. It focuses on the design and development of a digital platform called JobLynk, aimed at facilitating connections between students seeking micro-jobs and individuals, small businesses, or recruiters with occasional staffing needs.

The platform allows students to create and showcase their profiles, search for tasks based on their skills, location, and availability, while recruiters can post job offers and consult suitable profiles.

The project was developed using the Unified Process (UP) methodology, enabling an iterative and incremental life cycle that integrates analysis, design, implementation, and testing at each phase. The solution was implemented using modern web development technologies : Node.js and Express.js for the backend, React.js for the frontend, and PostgreSQL with TypeORM to manage database interactions efficiently.

The platform effectively meets the defined functional requirements by facilitating task search and profile consultation, while also satisfying non-functional requirements such as security, reliability, and ease of use.

Keywords : Micro-jobs, Students, Recruiters, Occasional employment, JobLynk, UP, UML.

Table des matières

1	Contexte du Projet	8
1.1	Introduction	8
1.2	Domaine du projet	8
1.3	Problématique et solution	9
1.3.1	Attentes des acteurs	9
1.3.2	Obstacles identifiés	9
1.4	Analyse du marché et positionnement	10
1.4.1	Marché cible	10
1.4.2	Étude concurrentielle	12
1.5	Présentation du projet	13
1.5.1	Objectifs du projet	13
1.5.1.1	Objectif principal :	13
1.5.1.2	Objectifs spécifiques :	13
1.5.2	Proposition de valeur	14
1.6	Planification du développement et de la mise en œuvre de JobLynk	14
1.7	Méthodologie de travail	15
1.7.1	Processus de développement	15
1.7.2	Processus Unifié (UP)	15
1.7.3	Langage de modélisation UML	16
1.7.4	Technologies utilisées	16
1.7.4.1	Frameworks web :	16
1.7.4.2	Technologies mobiles :	16
1.8	Conclusion	16
2	Conception du projet	18
2.1	Introduction	18
2.2	Analyse des besoins	18
2.2.1	Phase d'inception	18
2.2.2	Identification des besoins	18
2.2.2.1	Besoins fonctionnels :	19
2.2.2.2	Besoins non fonctionnels :	19
2.2.3	Identification des cas d'utilisation	20
2.2.4	Description textuelle des cas d'utilisation :	22
2.2.4.1	Cas d'utilisation 1 : Se connecter	23
2.2.4.2	Cas d'utilisation «Bloquer ou suspendre les comptes utilisateurs»	23
2.2.4.3	Cas d'utilisation « Publication d'une mission »	24
2.2.4.4	Cas d'utilisation « Recherche et consultation de profils d'étudiants »	24
2.2.4.5	Cas d'utilisation « Postuler à une mission »	25
2.2.4.6	Cas d'utilisation « Laisser une note et un avis »	25
2.3	Diagramme de séquence	25
2.3.1	Diagramme de séquence Système « S'inscrire »	26
2.3.2	Diagramme de séquence Système « S'authentifier »	27
2.3.3	Diagramme de séquence Système « Publier une mission »	28
2.3.4	Diagramme de séquence Système « Postuler à une offre »	29
2.4	Diagramme de classe	30
2.5	Diagramme de déploiement	30
2.6	Modèle relationnel	31
2.7	Conclusion	31

3	Implémentation	33
3.1	Introduction	33
3.2	Outils de développement :	33
3.3	Langages et frameworks	33
3.3.1	JavaScript & TypeScript	33
3.3.2	Node.js	34
3.3.3	NestJS	34
3.3.4	React.js (Framework web)	34
3.3.5	React Native (Framework Mobile)	34
3.3.6	TypeORM	35
3.4	Outils de développement et de gestion	35
3.4.1	PostgreSQL	35
3.4.2	Visual Studio Code	35
3.4.3	Git & GitHub	36
3.4.4	Docker	36
3.5	Schéma de navigation de l'application	36
3.5.1	Logo JobLynk	37
3.5.2	Inscription / Connexion	37
3.5.3	Interface Offres d'emplois	38
3.5.4	Interface Profils publics et messagerie	39
3.5.5	Page de publications	40
3.5.6	Page Profile connecté	40
3.5.7	Espace admin (pour l'équipe du site)	41
3.6	Conclusion	41

Liste des tableaux

1.1	Tableau comparatif des plateformes de micro-jobbing et freelancing	12
2.1	Acteurs et rôles du système	19
2.2	Cas d'utilisation « Se connecter »	23
2.3	Cas d'utilisation «Bloquer ou suspendre les comptes en cas de comportement inapproprié»	24
2.4	Cas d'utilisation « Publication d'une mission »	24
2.5	Cas d'utilisation « Recherche et consultation de profils d'étudiants »	24
2.6	Cas d'utilisation « Postuler à une mission »	25
2.7	Cas d'utilisation « Laisser une note et un avis »	25

Table des figures

2.1	Diagramme de cas d'utilisation ADMIN	20
2.2	Diagramme de cas d'utilisation Étudiant	21
2.3	Diagramme de cas d'utilisation Particulier	22
2.4	Diagramme de séquence S'inscrire	26
2.5	Diagramme de séquence S'authentifier	27
2.6	Diagramme de séquence publier une mission	28
2.7	Diagramme de séquence postuler à une offre	29
2.8	Diagramme de classe	30
2.9	Diagramme de déploiement	31
3.1	Logo JavaScript	34
3.2	Logo Node.js	34
3.3	Logo NestJS	34
3.4	Logo React.js	34
3.5	Logo React Native	35
3.6	Logo TypeORM	35
3.7	Logo PostgreSQL	35
3.8	Logo VS Code	36
3.9	Logo GitHub	36
3.10	Logo Docker	36
3.11	Logo JobLynk	37
3.12	Interfaces d'authentification (Inscription/Connexion)	37
3.13	Interface des offres d'emploi	38
3.14	Interface des profils publics et messagerie	39
3.15	Page de publications	40
3.16	Page de mon profile	40
3.17	Espace administrateur	41

Liste des abréviations

API :	Application Programming Interface
REST :	Representational State Transfer
UML :	Unified Modeling Language
UP :	Unified Process
CRUD :	Create, Read, Update, Delete
DBMS :	Database Management System
SQL :	Structured Query Language
ORM :	Object-Relational Mapping
JWT :	JSON Web Token
UI :	User Interface
UX :	User Experience
WEB :	World Wide Web
JSON :	JavaScript Object Notation
HTTP :	Hypertext Transfer Protocol
HTTPS :	Hypertext Transfer Protocol Secure
FE :	Frontend
BE :	Backend
DB :	Database (PostgreSQL)
TS :	TypeScript (si utilisé)

Introduction générale

L'évolution rapide des technologies numériques a profondément transformé les dynamiques sociales, économiques et professionnelles à l'échelle mondiale [1], [2]. Le numérique s'impose aujourd'hui comme un levier stratégique dans des domaines aussi variés que l'éducation, la santé, le commerce et, de manière croissante, dans le monde du travail [2]. Cette transformation ouvre de nouvelles opportunités, mais soulève également des défis inédits, notamment pour les jeunes et les étudiants qui aspirent à une autonomie financière sans compromettre leur parcours académique [3].

En Algérie, la situation des étudiants face au marché de l'emploi reste préoccupante [4], [5]. Bien compétents motivés, beaucoup peinent à trouver des opportunités compatibles avec leurs contraintes académiques [5]. Parallèlement, particuliers et petites structures recherchent des services ponctuels (garde d'enfants, soutien scolaire, petits travaux numériques, etc.), mais manquent de moyens efficaces pour contacter des prestataires fiables et proches [6], [5].

Dans ce contexte socio-économique que s'inscrit JobLynk, une plateforme numérique innovante de micro-jobbing localisé, pensée pour répondre à un double besoin : faciliter l'accès à de petits emplois pour les étudiants, tout en offrant aux particuliers un moyen simple, sécurisé et géolocalisé de déléguer des tâches quotidiennes.

JobLynk repose sur une idée fondatrice : celle de créer un pont intelligent entre les étudiants à la recherche de revenus complémentaires et les citoyens en quête de services ponctuels. La plateforme permet aux étudiants de proposer ou d'accepter des missions dans leur périmètre géographique. Conçue pour être intuitive et accessible, elle s'appuie sur des fonctionnalités modernes : messagerie intégrée, système d'évaluation, filtrage par zone, et à terme, intégration de solutions de paiement sécurisé.

Le présent mémoire retrace l'ensemble du processus de développement de cette solution, depuis sa conception théorique jusqu'à sa mise en œuvre technique. Il se structure autour de trois chapitres complémentaires :

Le premier chapitre aborde les généralités du projet, en présentant son origine, son contexte d'émergence, ses objectifs et les solutions existantes dans l'état de l'art.

Le deuxième chapitre est consacré à la conception fonctionnelle et technique, incluant la modélisation de l'architecture du système, les choix technologiques, les interfaces utilisateurs.

Le troisième chapitre détaille la phase de réalisation, en décrivant le développement de l'application, les outils utilisés assurer la stabilité et la performance du système..

À travers cette démarche, nous cherchons à démontrer la pertinence de JobLynk non seule-

ment comme réponse technologique à un problème concret, mais aussi comme outil d'inclusion sociale et économique, capable de dynamiser l'engagement des jeunes, de valoriser les compétences locales, et d'amorcer une transformation progressive du marché des micro-services en Algérie.

Chapitre 1

Contexte du Projet

1.1 Introduction

Ce chapitre est consacré à la présentation du domaine d'étude dans lequel s'inscrit notre projet JobLynk, en mettant en lumière les enjeux socio-économiques et les spécificités du secteur ciblé. Il constitue une étape essentielle pour comprendre le contexte dans lequel la solution sera déployée.

Nous commencerons par définir le périmètre du projet, en identifiant les besoins concrets auxquels il répond dans le domaine des micro-services localisés à destination des étudiants. Cette analyse permettra de souligner les problématiques existantes sur le marché algérien.

Ensuite, Nous analyserons également le marché cible et la concurrence existante, afin de situer notre projet dans un environnement économique et technologique plus large.

Enfin, ce chapitre jettera les bases nécessaires pour aborder, dans les sections suivantes, la phase de conception technique et fonctionnelle, en offrant une compréhension globale du secteur visé, des utilisateurs cibles et des défis à relever.

1.2 Domaine du projet

Le domaine d'étude dans lequel s'inscrit notre projet est celui des services à la demande et plus particulièrement des micro-services localisés destinés aux étudiants. Il s'agit d'un secteur émergent qui vise à faciliter la mise en relation rapide entre des personnes recherchant de l'aide ponctuelle pour de petites tâches, et des prestataires capables de les réaliser à proximité géographique.

Ce domaine, bien qu'en plein essor à l'échelle mondiale, reste encore peu structuré en Algérie, où l'emploi informel est très répandu et où les jeunes rencontrent des difficultés importantes pour accéder à des opportunités de travail flexibles, adaptées à leurs contraintes d'études et de mobilité [4], [5] .

Par ailleurs, les services proposés couvrent une large variété d'activités, allant des tâches manuelles (ménage, bricolage, garde d'enfants) aux missions numériques (soutien scolaire, assistance informatique, traduction), ce qui complexifie la gestion et la coordination entre prestataires et demandeurs [2], [7] .

Dans un contexte où le numérique et la mobilité jouent un rôle central dans la vie quotidienne, l'intégration d'une plateforme digitale adaptée apparaît comme une solution moderne et

nécessaire pour structurer ce marché fragmenté [6]. Cette transition numérique permettra d'organiser, sécuriser et simplifier les échanges, tout en rendant ces services accessibles de manière rapide et fiable.

1.3 Problématique et solution

Dans le contexte socio-économique actuel en Algérie, les étudiants rencontrent d'importantes difficultés pour accéder à un emploi rémunéré compatible avec leur statut et leur emploi du temps universitaire [3], [4], [5]. Si le travail étudiant est perçu comme une nécessité par une majorité d'étudiants, que ce soit pour subvenir à leurs besoins quotidiens, financer leurs études ou acquérir une première expérience professionnelle, les opportunités concrètes restent très limitées [5].

Les offres d'emploi traditionnelles, souvent rigides en matière d'horaires, de localisation ou d'exigences de qualification, ne répondent que partiellement, voire pas du tout, aux besoins spécifiques des jeunes en formation.[5] De nombreux étudiants se retrouvent ainsi face à un dilemme : soit renoncer à une activité rémunératrice, soit compromettre la qualité de leur parcours académique en acceptant un emploi inadapté.

Malgré l'existence d'initiatives publiques telles que l'ANSEJ, qui soutiennent l'emploi des jeunes à travers des dispositifs de financement et d'accompagnement entrepreneurial, peu d'outils ciblent spécifiquement les besoins ponctuels et flexibles des étudiants [8]. Parallèlement, une demande réelle et croissante de services ponctuels (tâches ménagères, aide à domicile, soutien scolaire, assistance informatique, etc.) émerge au sein de la population, notamment chez les particuliers et les très petites entreprises [2], [6]. Cependant, cette demande reste dispersée, informelle et difficile à satisfaire dans un cadre structuré, sécurisé et fiable [6], [5].

1.3.1 Attentes des acteurs

- **Étudiants :**
 - o Accéder rapidement à des missions courtes et géographiquement proches.
 - o Bénéficier d'un cadre sécurisé (contrat, assurance) et d'une rémunération garantie.
 - o Préserver l'équilibre études/travail grâce à des horaires souples et à une gestion simplifiée.
- **Demandeurs de services (particuliers, TPE)** o Trouver en quelques clics un prestataire évalué et disponible.
- o Disposer de retours d'expérience fiables pour guider leur choix.
- o Simplifier le suivi des prestations.

1.3.2 Obstacles identifiés

- **Fragmentation de l'offre et de la demande :** absence d'un point d'entrée unique pour recenser les missions disponibles et les profils d'étudiants compétents.
- **Manque de transparence et de confiance :** aucun mécanisme fiable d'évaluation, de recommandations ou de retours d'expérience n'existe pour garantir la qualité des prestations.
- **Complexité administrative :** facturation, suivi des heures et paiements s'effectuent souvent de façon artisanale (espèces, promesses verbales), ce qui alourdit la gestion pour les deux parties.
- **Couverture géographique limitée :** les canaux classiques (bouche-à-oreille, petites annonces) ne tiennent pas compte de la localisation précise, rendant les mises en relation longues et inefficaces.

La question principale que soulève notre projet est donc la suivante :

Comment concevoir et développer une plateforme numérique locale de micro-prestation de services, capable de mettre en relation de manière rapide, fiable et géolocalisée les étudiants en quête de missions et les demandeurs de petits services en Algérie ?

Cette problématique soulève plusieurs sous-questions, parmi lesquelles :

- - Quelles fonctionnalités clés répondre aux besoins respectifs des étudiants et des demandeurs ?
- - Comment garantir la sécurité, la fiabilité et la confidentialité des échanges et données ?
- - Comment intégrer un mécanisme d'évaluation et de confiance entre utilisateurs ?

C'est à ces interrogations que le projet **JobLynk** entend répondre, à travers la conception d'un système complet, centré sur l'utilisateur, évolutif et respectueux des normes en vigueur.

Cette approche vise à structurer l'économie de proximité, à professionnaliser les échanges et à renforcer l'autonomie financière des jeunes, tout en offrant aux demandeurs un service souple, fiable et traçable.

1.4 Analyse du marché et positionnement

Les plateformes de mise en relation pour missions ponctuelles connaissent une croissance significative dans les pays occidentaux, avec des acteurs majeurs tels que TaskRabbit (spécialisé dans le bricolage et les services à domicile), Fiverr (axé sur les services numériques) ou encore Upwork (dédié au freelancing globalisé).

Ces solutions ont su s'imposer grâce à un écosystème numérique mature, une forte pénétration des moyens de paiement en ligne et une culture du service bien ancrée [9], [10] [11].

Toutefois, ces modèles peinent à être transposés tels quels dans le contexte algérien, en raison de plusieurs barrières structurelles : une forte diversité linguistique (prédominance de l'arabe dialectal et du français), l'inadéquation des systèmes de paiement électronique, ainsi qu'une faible digitalisation des usages liés aux services de proximité. De plus, les plateformes internationales ne tiennent généralement pas compte des spécificités sociales, économiques et culturelles du marché local, ce qui limite leur adoption.

Par ailleurs, malgré l'émergence des startups en Algérie, aucune plateforme nationale ne propose aujourd'hui une solution intégrée de micro-jobbing combinant services physiques (livraison, aide à domicile. . .) et numériques (assistance bureautique, design, saisie de données. . .), avec une interface accessible, multilingue (arabe, français, anglais) et adaptée aux étudiants.

Ce manque sur le marché constitue une opportunité stratégique majeure : développer une solution locale, inclusive et adaptée, capable de structurer un secteur encore informel, tout en répondant aux attentes d'une jeunesse connectée cherchant flexibilité, autonomie et reconnaissance.

1.4.1 Marché cible

Le marché cible de **JobLynk** s'articule autour de deux segments principaux :

- **Les étudiants** : principalement en licence ou master, ils cherchent des revenus complémentaires adaptés à leurs horaires rigides et leur mobilité limitée. **JobLynk** leur propose

une plateforme flexible pour créer un profil valorisant leurs compétences, postuler facilement à des missions courtes et locales, et gérer leurs tâches simplement.

- **Les particuliers et petites entreprises (TPE)** : ce segment regroupe des profils variés, allant de la famille recherchant une aide ponctuelle (ménage, courses), aux commerçants et indépendants ayant besoin d'un renfort temporaire (livraison, inventaire, aide bureautique). Ce public recherche avant tout une solution rapide, fiable et locale, permettant de poster une annonce en quelques minutes, filtrer les profils disponibles par zone et compétences.

1.4.2 Étude concurrentielle

TABLE 1.1 – Tableau comparatif des plateformes de micro-jobbing et freelancing

Plateforme	Domaine principal	Type de services	Modèle économique	Langues supportées	Paiements	Adaptabilité au contexte algérien
TaskRabbit	Services physiques (bricolage, déménagement, aide à domicile)	Missions ponctuelles sur site	Commission sur chaque prestation	Anglais principalement	Paiement en ligne (carte bancaire, PayPal)	Non : services de proximité mais absence de support multilingue et dépendance à des moyens de paiement non disponibles en Algérie
Fiverr	Services numériques (graphisme, rédaction, développement, etc.)	Missions dématérialisées à distance	Commission sur les ventes (à partir de 20%)	Anglais (interface), autres langues limitées	Paiement en ligne, frais importants pour les prestataires	Limitée : bonnes opportunités pour freelancing numérique mais peu accessible aux étudiants non anglophones et sans moyens de paiement
Upwork	Freelancing global (IT, rédaction, marketing, etc.)	Projets souvent de moyenne à longue durée	Commission + abonnement premium	Anglais uniquement	Paiement en ligne, retrait via banques partenaires	Faible : profils exigeants, en anglais, peu adapté à des micro-missions locales
JobLynk (projet proposé)	Services physiques et numériques (ménage, baby-sitting, cours particuliers, aide bureautique, etc.)	Micro-tâches locales ou à distance	Gratuit pour les étudiants, commissions minimales pour les demandeurs	Multilingue : arabe, français, anglais	Paiement en main propre ou solutions locales (Baridi Mob, CIB...)	Élevée : plateforme pensée pour le marché algérien, adaptée aux réalités linguistiques, culturelles et techniques locales

1.5 Présentation du projet

- - **Application mobile et web (étudiants & particuliers)** : Une interface unique, quel que soit le support, permettant de publier et de consulter rapidement des offres de micromissions, de candidater ou de recevoir des propositions, et de suivre simplement son activité (historique, évaluations, notifications).
- - **Application web Admin** : Un backoffice centralisé pour l'équipe JobLynk, assurant :
 - la gestion des comptes utilisateurs (étudiants, particuliers, TPE),
 - la modération des annonces et le traitement des signalements,
 - le suivi des indicateurs clés (nombre de missions, taux de réponse, satisfaction),
 - la configuration globale (langues, moyens de paiement).

Cette architecture à deux niveaux garantit à la fois une expérience fluide pour les utilisateurs finaux et un pilotage opérationnel efficace de la plateforme.

1.5.1 Objectifs du projet

JobLynk vise à dépasser le simple lancement d'une plateforme, en construisant une solution durable ancrée dans les besoins réels des étudiants et particuliers algériens.

1.5.1.1 Objectif principal :

- - Développer une application mobile et web de micro-jobbing localisée, pensée spécifiquement pour les étudiants algériens. L'objectif est de faciliter l'accès à des missions ponctuelles, tout en structurant une offre locale de micro-services.
- - Double enjeu : soutenir l'autonomie financière des étudiants et offrir une alternative fiable aux besoins quotidiens des particuliers.

1.5.1.2 Objectifs spécifiques :

Afin de concrétiser cette vision, plusieurs objectifs opérationnels sont définis :

- - Offrir une interface ergonomique, intuitive et mobile-first, pensée pour une utilisation fluide et accessible depuis un smartphone, outil principal d'accès à Internet pour la majorité des étudiants en Algérie.
- - Mettre en œuvre un système de géolocalisation intelligent, permettant de rapprocher en temps réel les étudiants disponibles et les demandeurs de services dans une même zone géographique, optimisant ainsi la réactivité et la pertinence des mises en relation.
- - Intégrer des mécanismes de confiance, tels qu'un système de notation et de retour d'expérience, afin de garantir la fiabilité des échanges et de favoriser une dynamique de responsabilité mutuelle entre les parties.
- - Favoriser l'autonomie économique des étudiants, en leur donnant un accès facilité à des revenus complémentaires, sans compromettre la poursuite de leurs études ni leur bien-être.
- - Jeter les bases d'une plateforme évolutive, capable d'intégrer à terme de nouveaux modules et services (tels que des missions numériques à distance, des initiatives d'entrepreneuriat étudiant, ou encore des formations certifiantes), avec pour finalité la construction

d'un véritable écosystème de valorisation des compétences locales.

1.5.2 Proposition de valeur

JobLynk se démarque par :

- **Une solution pensée pour le contexte algérien et le public étudiant** : JobLynk prend en compte les réalités économiques, linguistiques et technologiques locales. L'interface multilingue (arabe, français, anglais), la prise en charge des paiements alternatifs (main propre, BaridiMob, CIB), ainsi qu'un design simple et intuitif permettent de démocratiser l'accès à des missions rémunérées pour les jeunes, tout en répondant aux besoins concrets des particuliers et des structures de proximité.
- **Un système de mise en relation transparent et fiable** : Chaque étudiant ou demandeur est invité à compléter un profil détaillé incluant vérifications basiques (email, numéro de téléphone, carte d'étudiant), évaluations post-mission et historique d'activité. Ce système de notation bidirectionnel renforce la confiance entre les utilisateurs et valorise les profils les plus engagés et professionnels.
- **Une digitalisation complète de la recherche et du suivi des micro-jobs** : L'ensemble du parcours utilisateur (inscription, publication d'une offre ou d'une mission, candidature, acceptation, suivi, évaluation) est centralisé sur la plateforme, que ce soit sur mobile ou sur web. Cela garantit une expérience fluide, rapide et efficace, sans recours à des démarches externes ou intermédiaires, en phase avec les attentes des jeunes actifs et des particuliers.

1.6 Planification du développement et de la mise en œuvre de JobLynk

Le projet JobLynk sera déployé de manière progressive, selon une planification structurée en plusieurs étapes. L'objectif est de garantir une montée en charge maîtrisée, tout en livrant une solution alignée avec les besoins des étudiants et des recruteurs locaux.

Phase 1 : Développement du MVP (Minimum Viable Product)

La première phase consistera à développer une version initiale fonctionnelle de la plateforme, disponible à la fois en version mobile et web. Ce MVP contiendra les fonctionnalités de base indispensables :

- - Création de comptes (étudiants et recruteurs),
- - Publication d'offres de missions ponctuelles,
- - Candidature rapide pour les étudiants,
- - Messagerie intégrée,
- - Gestion de profil (compétences, expériences, CV),
- - Notifications en temps réel,
- - Interface d'administration basique.

L'accent sera mis sur l'ergonomie, la rapidité de mise en relation, et la sécurité des données. Ce MVP permettra d'expérimenter le fonctionnement global de la plateforme tout en limitant les risques techniques et financiers.

Phase 2 : Lancement en bêta auprès d'un groupe restreint d'utilisateurs

Une fois le MVP achevé, une phase de test sera lancée avec un panel ciblé :

- - Étudiants (universités partenaires),
- - Petites entreprises locales (cafés, coiffeurs, commerçants...),
- - Freelancers ou auto-entrepreneurs.

Objectifs de cette phase :

- - Collecter des retours d'usage (ergonomie, facilité de publication/candidature),
- - Identifier les bugs ou points faibles techniques,
- - Tester les premières offres payantes (formule pro, visibilité renforcée).

Les ajustements issus de cette phase permettront d'améliorer le produit avant une mise sur le marché élargie.

Phase 3 : Lancement officiel et stratégie de croissance

Une fois le produit stabilisé, JobLynk sera lancé officiellement à l'échelle nationale. Ce lancement sera accompagné d'un plan d'acquisition progressif, basé sur :

- - Des campagnes digitales ciblées (Instagram, TikTok, Facebook),
- - Des partenariats avec universités, incubateurs et espaces de coworking,
- - La promotion de témoignages utilisateurs pour instaurer la confiance.

L'objectif est d'installer JobLynk comme le réflexe n°1 pour les missions ponctuelles étudiantes, avec une stratégie de croissance agile et centrée sur les retours terrain.

1.7 Méthodologie de travail

Pour garantir la réussite du projet Joblynk, nous adopterons une méthodologie rigoureuse basée sur les meilleures pratiques du développement logiciel, avec une attention particulière portée à la qualité, aux coûts et aux délais.

1.7.1 Processus de développement

Un processus définit un cadre séquentiel et ordonné destiné à l'obtention d'un système ou à l'évaluation d'un système existant. L'objet est de produire des logiciels de qualité qui répondent aux besoins utilisateurs dans des conditions prévisibles [12], [13].

1.7.2 Processus Unifié (UP)

Le processus unifié (UP) est une famille de méthodes de développement de logiciels orientés objets. Elle se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML [13], [14], [15]. Elle définit un processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement composés d'une phase de création, d'une phase d'élaboration, d'une phase de construction et d'une phase de transition, comprenant chacune plusieurs itérations [12] :

- **Itératif et incrémental** : Le projet est découpé en itérations courtes (typiquement 2 à 4 semaines). Chaque itération aboutit à une version partielle mais exécutable du système, enrichie à chaque cycle. Cela permet de valider régulièrement les avancées, d'ajuster rapidement en fonction des retours et de maîtriser les risques techniques et fonctionnels [13].

- **Centré sur l'architecture** : L'architecture du système est conçue dès les premières étapes et affinée progressivement. Une attention particulière est portée à la modularité, à la séparation des responsabilités et à la capacité d'évolution. L'ensemble est modélisé en UML (Unified Modeling Language) [13], [16].
- **Piloté par les risques** : L'identification et la réduction des risques sont au cœur de la planification. Les itérations sont priorisées en fonction des risques les plus critiques (techniques, organisationnels, fonctionnels), afin de sécuriser le projet dès les premières phases [12].
- **Conduit par les cas d'utilisation** : Le développement est guidé par les besoins fonctionnels exprimés sous forme de cas d'utilisation détaillés, ce qui garantit que le produit final correspond bien aux attentes réelles [13].

1.7.3 Langage de modélisation UML

Le processus unifié est un processus entièrement basé sur UML. Ce dernier se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [16], [17], [18].

Dans notre projet nous utiliserons uniquement le diagramme de cas d'utilisation, le diagramme de séquence, le diagramme de classe et le diagramme de déploiement.

1.7.4 Technologies utilisées

1.7.4.1 Frameworks web :

- **React.js** : pour le développement des interfaces web. Ce framework nous offre une grande flexibilité, des performances optimisées et une excellente expérience utilisateur.
- **NestJS** : pour la création du backend. Ce framework Node.js basé sur TypeScript fournit une architecture solide, modulaire et adaptée aux applications complexes, tout en facilitant la scalabilité et la maintenance.

1.7.4.2 Technologies mobiles :

- **React Native** : sera utilisé pour développer une application mobile multiplateforme (iOS et Android) à partir d'une seule base de code, garantissant ainsi un déploiement plus rapide et des coûts de maintenance réduits.

Le choix de cette stack technologique assure à Joblynk une base technique fiable, moderne et capable de soutenir l'évolution du projet à long terme.

1.8 Conclusion

Le domaine des micro-services localisés pour étudiants constitue un secteur à la fois prometteur et encore peu exploité, notamment dans le contexte algérien. Ce projet s'inscrit dans une dynamique d'innovation technologique et sociale, visant à répondre aux besoins spécifiques d'une population jeune en quête de flexibilité et d'opportunités économiques. L'étude préalable de ce domaine révèle les défis liés à la fragmentation du marché, à l'absence d'outils numériques adaptés, ainsi qu'à la nécessité de garantir la confiance et la sécurité dans les échanges. Ces éléments justifient pleinement le développement d'une plateforme digitale telle que JobLynk, qui ambitionne d'apporter une réponse intégrée, accessible et moderne à ces problématiques. Le chapitre suivant présentera la conception détaillée de cette solution, en explicitant les choix

techniques, fonctionnels et ergonomiques qui permettront de concrétiser cette vision.

Chapitre 2

Conception du projet

2.1 Introduction

Ce chapitre est consacré à la phase de conception de notre projet JobLynk, qui repose sur une méthodologie rigoureuse afin d'assurer une transition fluide vers la phase de réalisation. Nous y exposerons l'ensemble des étapes suivies pour définir les fondations du système.

Nous commencerons par analyser les besoins des utilisateurs et identifier les fonctionnalités essentielles du système. Cette analyse permettra de mettre en évidence les objectifs de notre projet ainsi que les différents acteurs impliqués. Ensuite, nous représenterons ces besoins sous forme de diagrammes UML, offrant ainsi une vision structurée des interactions et du fonctionnement du système.

Enfin, nous procéderons à la conception du modèle relationnel de la base de données, garantissant une structure cohérente et optimisée pour la gestion des informations.

2.2 Analyse des besoins

L'analyse des besoins permet d'identifier les attentes des utilisateurs et de prendre en compte les contraintes liées au micro-jobbing localisé et à la digitalisation des prestations de proximité.

2.2.1 Phase d'inception

Expression des besoins : Cette section recense les attentes et contraintes des principaux acteurs de *JobLynk*, à savoir :

- **Étudiants** : cherchent des missions courtes, flexibles et proches de leur lieu d'étude ou de résidence.
- **Particuliers et TPE** : en quête de prestataires fiables pour des services ponctuels.
- **Administrateurs** : responsables de la supervision, de la modération et de l'évolution de la plateforme.

Public visé :

- **Les étudiants** : désireux de valoriser leurs compétences et de générer des revenus complémentaires sans nuire à leur parcours académique.
- **Les particuliers et petites entreprises** : souhaitant déléguer rapidement des tâches ponctuelles à des prestataires évalués et géolocalisés.
- **Les administrateurs de JobLynk** : chargés d'assurer le bon fonctionnement, la qualité de service et la sécurité de l'écosystème.

2.2.2 Identification des besoins

L'identification des besoins a pour objectif de recenser et structurer les exigences du système en utilisant le concept de cas d'utilisation proposé par UML. Cette phase permet également

d'identifier les classes candidates du modèle statique d'analyse.

2.2.2.1 Besoins fonctionnels :

Un système ne devient opérationnel que s'il satisfait les besoins fonctionnels (besoins métiers).

Acteurs	Rôles / Responsabilités
Administrateur	<ul style="list-style-type: none"> — Gérer les comptes des utilisateurs (étudiants, particuliers/TPE). — Superviser l'ensemble des missions et offres publiées. — Valider et modérer les profils. — Bloquer les comptes en cas de comportement inapproprié ou frauduleux.
Étudiant	<ul style="list-style-type: none"> — Créer et mettre à jour son profil (compétences, expériences, disponibilités). — Chercher, consulter et postuler aux missions publiées. — Suivre l'historique de ses missions et les évaluations reçues. — Chat en temps réel.
Particulier / TPE	<ul style="list-style-type: none"> — S'inscrire et publier des offres de missions détaillées (type de service, durée, localisation). — Rechercher, consulter et comparer les profils d'étudiants. — Accepter une candidature et gérer les réservations. — Suivre l'historique de ses missions proposées. — Laisser un avis sur la prestation réalisée. — Chat en temps réel.

TABLE 2.1 – Acteurs et rôles du système

2.2.2.2 Besoins non fonctionnels :

Ces besoins ont un aspect visible pour l'utilisateur, mais ne sont pas en rapport direct avec le comportement du système. Il s'agit des contraintes techniques et des caractéristiques du système :

1. Performance & Scalabilité

- Temps de réponse rapide pour les recherches, publications et mises à jour de profil.
- Capacité à gérer un fort nombre d'utilisateurs et de transactions simultanément.

2. Accessibilité & Multiplateforme

- Disponibilité 24h/24, 7j/7.
- Application mobile Android/iOS et web, adaptée aux smartphones de moyenne gamme.
- Interface multilingue (arabe, français, anglais).

3. Simplicité & Ergonomie

- Expérience utilisateur fluide et intuitive pour un public jeune et varié.
- Interfaces conviviales, attractives et responsives, adaptées aux rôles des différents acteurs.

4. Sécurité & Confidentialité

- Chiffrement des données personnelles et des échanges (mots de passe, profils, messages).
- Authentification sécurisée et traçabilité des opérations sensibles.
- Protection des informations de paiement et des transactions.

5. Modularité & Évolutivité

- Architecture modulaire pour faciliter l’ajout de nouvelles fonctionnalités.
- Structure de code et base de données permettant des évolutions sans perturber le service.

2.2.3 Identification des cas d’utilisation

Un cas d’utilisation désigne un ensemble de séquences d’actions effectuées par le système qui aboutissent à un résultat perceptible pour un utilisateur spécifique [19].

— Diagramme de cas d’utilisation (Admin) :

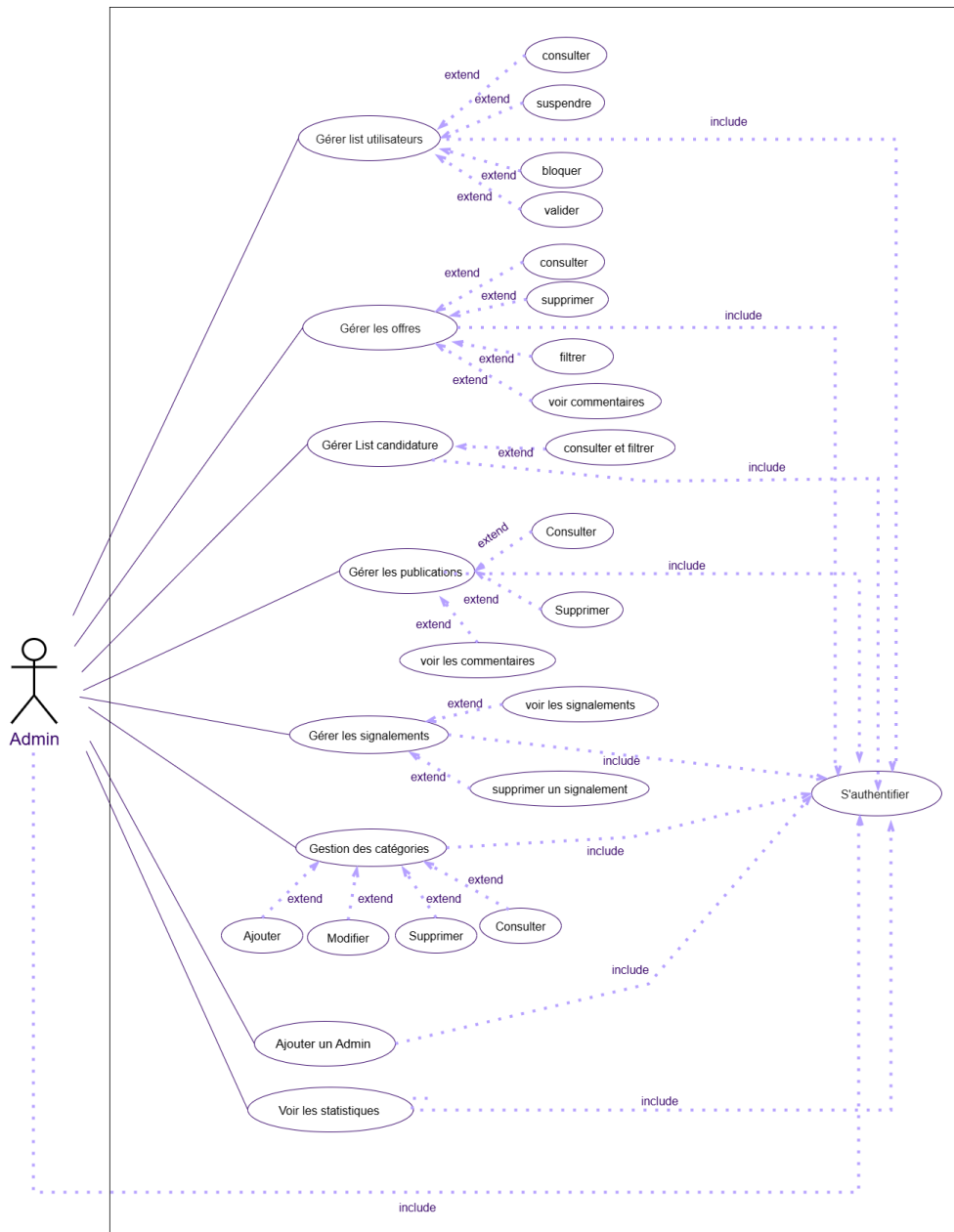


FIGURE 2.1 – Diagramme de cas d’utilisation ADMIN

— Diagramme de cas d'utilisation (Étudiant) :

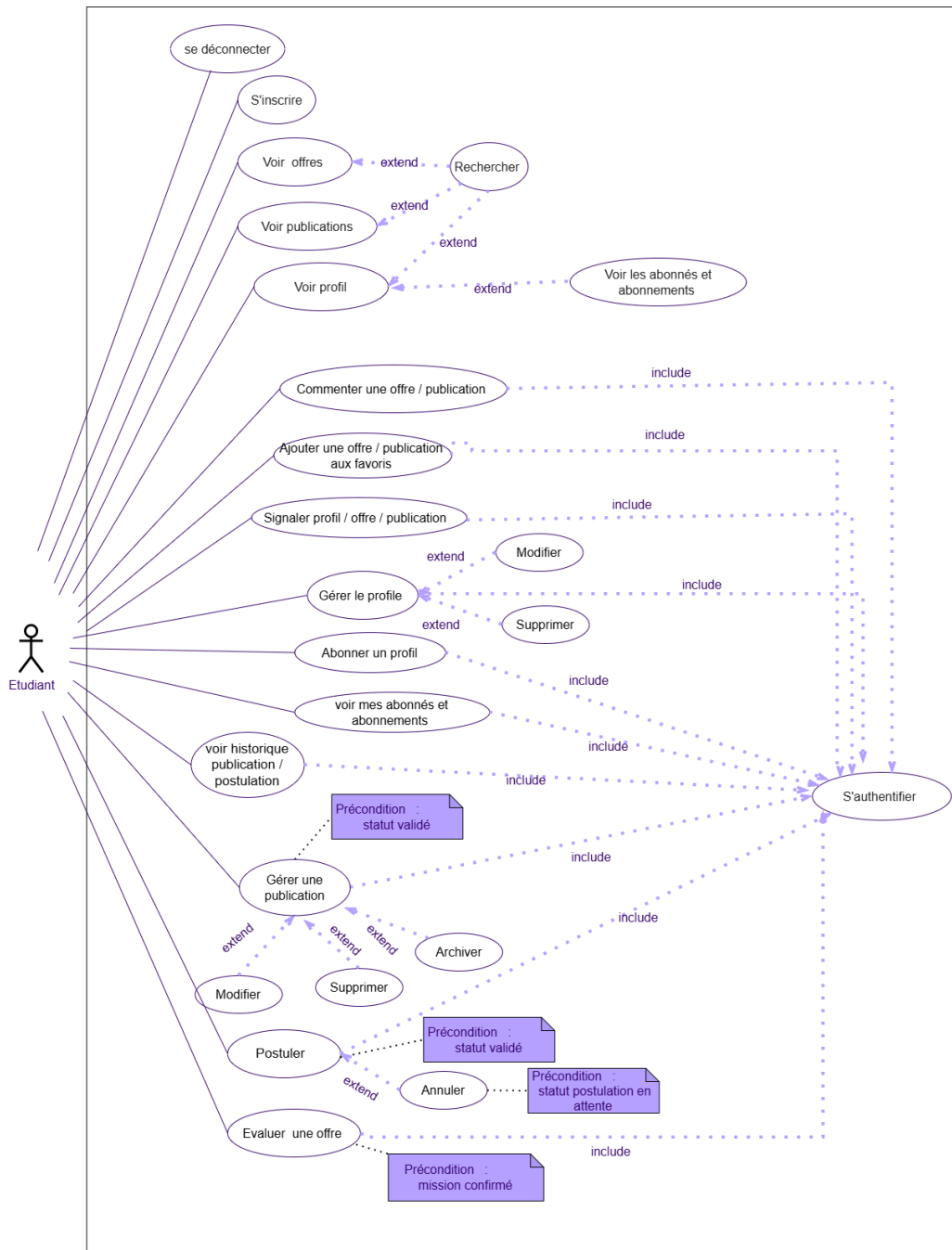


FIGURE 2.2 – Diagramme de cas d'utilisation Étudiant

— Diagramme de cas d'utilisation (Particulier) :

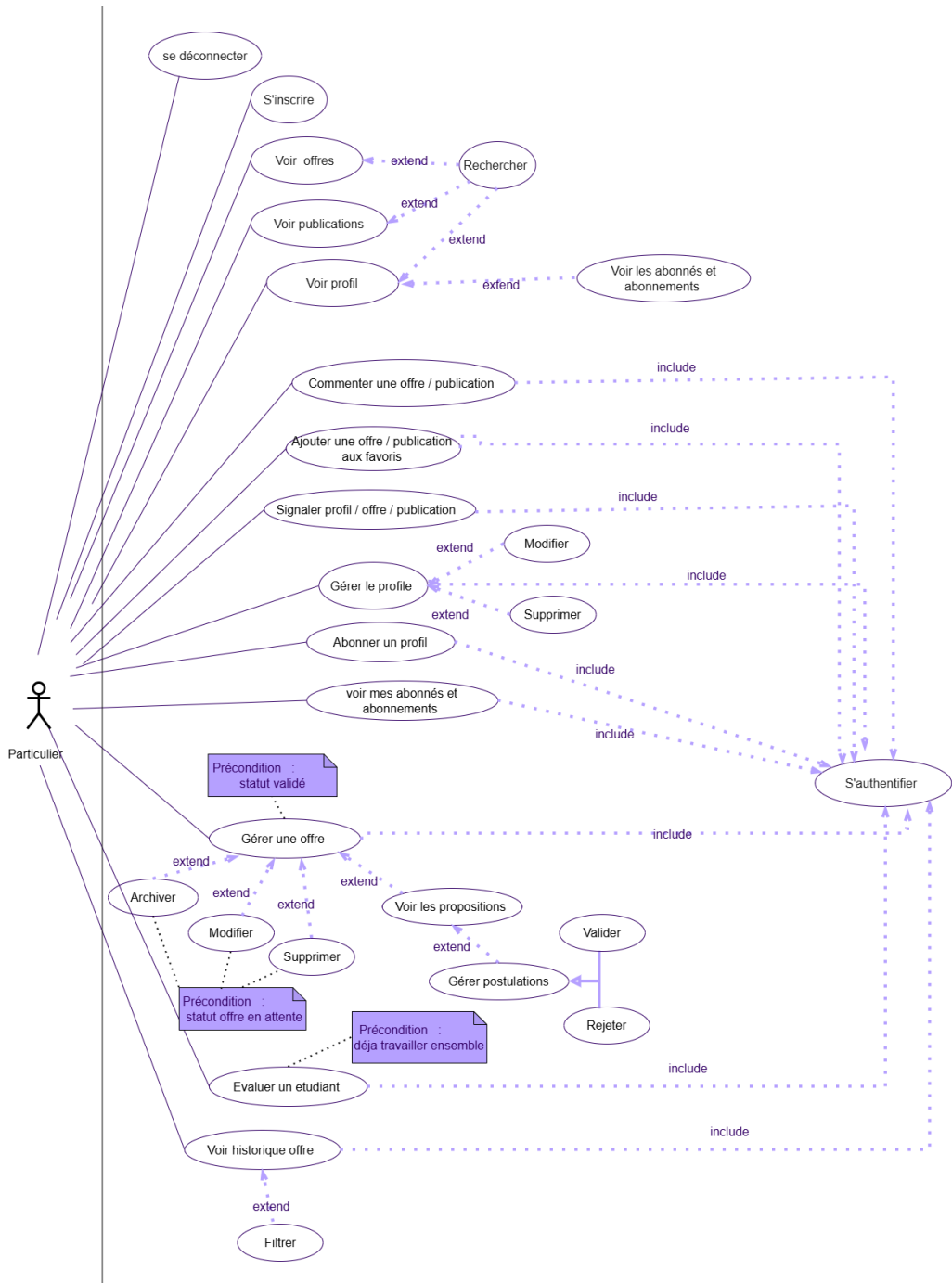


FIGURE 2.3 – Diagramme de cas d'utilisation Particulier

2.2.4 Description textuelle des cas d'utilisation :

Dans cette section, nous décrivons de manière textuelle l'ensemble des cas d'utilisation identifiés lors de l'analyse des besoins. Chaque cas d'utilisation représente un scénario d'interaction complet entre un acteur (ou plusieurs) et le système.

2.2.4.1 Cas d'utilisation 1 : Se connecter

Élément	Description
Acteur	Administrateur, Étudiant, Particulier / Petite entreprise
Précondition	Le compte utilisateur a été créé.
Déclencheur	L'utilisateur tente de se connecter à la plateforme JobLynk.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de connexion. 2. Il entre son email et mot de passe. 3. Le système vérifie que l'email est confirmé. 4. Le système vérifie que le compte est validé par l'administrateur. 5. L'utilisateur est redirigé vers l'espace interne correspondant à son rôle.
Scénario alternatif	<ul style="list-style-type: none"> - Si l'email n'est pas confirmé : message avec lien de validation. - Si le compte est en attente de validation admin : message d'attente. - Identifiants invalides : message d'erreur.
Postcondition	L'utilisateur accède à l'application ou est informé de l'état de son compte.

TABLE 2.2 – Cas d'utilisation « Se connecter »

2.2.4.2 Cas d'utilisation «Bloquer ou suspendre les comptes utilisateurs»

Élément	Description
Acteur	Administrateur
Précondition	<ul style="list-style-type: none"> — Un signalement de comportement inapproprié a été enregistré dans le système. — L'administrateur est authentifié.
Déclencheur	Réception d'un signalement ou d'une alerte indiquant un comportement suspect sur le compte d'un utilisateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur se connecte au back-office. 2. Il consulte la liste des signalements. 3. Il examine les preuves ou commentaires. 4. Il décide de bloquer ou suspendre le compte. 5. Le système passe le compte au statut «bloqué» ou «suspendu».
Scénario alternatif	<ul style="list-style-type: none"> — Si le signalement est injustifié : aucune action, le compte reste actif. — L'administrateur peut aussi envoyer un avertissement.
Postcondition	Le compte de l'utilisateur signalé est bloqué ou suspendu. L'action est enregistrée dans la base de données pour garantir la traçabilité.

Élément	Description
---------	-------------

TABLE 2.3 – Cas d'utilisation «Bloquer ou suspendre les comptes en cas de comportement inapproprié»

2.2.4.3 Cas d'utilisation « Publication d'une mission »

Élément	Description
Acteur	Particulier ou Petite entreprise
Précondition	L'utilisateur est connecté et son compte a été validé.
Déclencheur	Le particulier souhaite poster une tâche ponctuelle.
Scénario nominal	<ol style="list-style-type: none"> 1. Il accède à l'interface « Publier une mission ». 2. Il renseigne les détails (titre, description, date, rémunération, localisation). 3. Il valide la publication. 4. Le système enregistre et rend la mission visible aux étudiants.
Scénario alternatif	<ul style="list-style-type: none"> — Si le profil de l'utilisateur n'est pas validé, le système refuse la publication et l'invite à compléter son dossier. — Si des informations obligatoires manquent, un message d'erreur est renvoyé.
Postcondition	La mission est accessible aux étudiants pour postuler.

TABLE 2.4 – Cas d'utilisation « Publication d'une mission »

2.2.4.4 Cas d'utilisation « Recherche et consultation de profils d'étudiants »

Élément	Description
Acteur	Particulier ou Petite entreprise
Précondition	L'utilisateur accède à la plateforme sans besoin de connexion.
Déclencheur	L'utilisateur accède à la section « recherche des profils »
Scénario nominal	<ol style="list-style-type: none"> 1. Il utilise les filtres (compétences, localisation, disponibilité). 2. Il consulte les profils. 3. Il compare les évaluations, avis et expériences. 4. Il sélectionne un profil pour prise de contact ou invitation à une mission.
Scénario alternatif	<ul style="list-style-type: none"> — Aucun profil trouvé : affichage d'un message approprié.
Postcondition	L'utilisateur identifie un étudiant pertinent pour une mission.

TABLE 2.5 – Cas d'utilisation « Recherche et consultation de profils d'étudiants »

2.2.4.5 Cas d'utilisation « Postuler à une mission »

Élément	Description
Acteur	Étudiant
Précondition	L'étudiant est connecté et son profil est complet.
Déclencheur	Il consulte une mission disponible sur la plateforme.
Scénario nominal	<ol style="list-style-type: none"> 1. Il accède à la description complète d'une mission. 2. Il clique sur « Postuler ». 3. Une confirmation est affichée. 4. Le système enregistre la candidature et notifie l'auteur de la mission.
Scénario alternatif	<ul style="list-style-type: none"> — Mission supprimée ou désactivée : message d'indisponibilité. — Profil incomplet : message bloquant la postulation.
Postcondition	La candidature est prise en compte dans le suivi de la mission. L'historique des candidatures est mis à jour pour référence ultérieure.

TABLE 2.6 – Cas d'utilisation « Postuler à une mission »

2.2.4.6 Cas d'utilisation « Laisser une note et un avis »

Élément	Description
Acteur	Étudiant ou Particulier/Petite entreprise
Précondition	La mission a été réalisée et clôturée.
Déclencheur	L'un des deux acteurs souhaite donner une évaluation.
Scénario nominal	<ol style="list-style-type: none"> 1. Il accède à l'historique des missions. 2. Il sélectionne la mission concernée. 3. Il laisse une note (1 à 5 étoiles) et un avis. 4. Le système enregistre et publie l'évaluation sur le profil de l'autre partie.
Scénario alternatif	<ul style="list-style-type: none"> — Contenu inapproprié : modération par l'admin.
Postcondition	L'avis influence la réputation de l'utilisateur concerné.

TABLE 2.7 – Cas d'utilisation « Laisser une note et un avis »

2.3 Diagramme de séquence

Un diagramme de séquence est une représentation graphique qui détaille l'ordre chronologique des interactions entre les objets ou acteurs d'un système lors de l'exécution d'un scénario particulier [20]. Dans notre projet, plusieurs diagrammes de séquence ont été élaborés pour illustrer les scénarios clés.

2.3.1 Diagramme de séquence Système « S’inscrire »

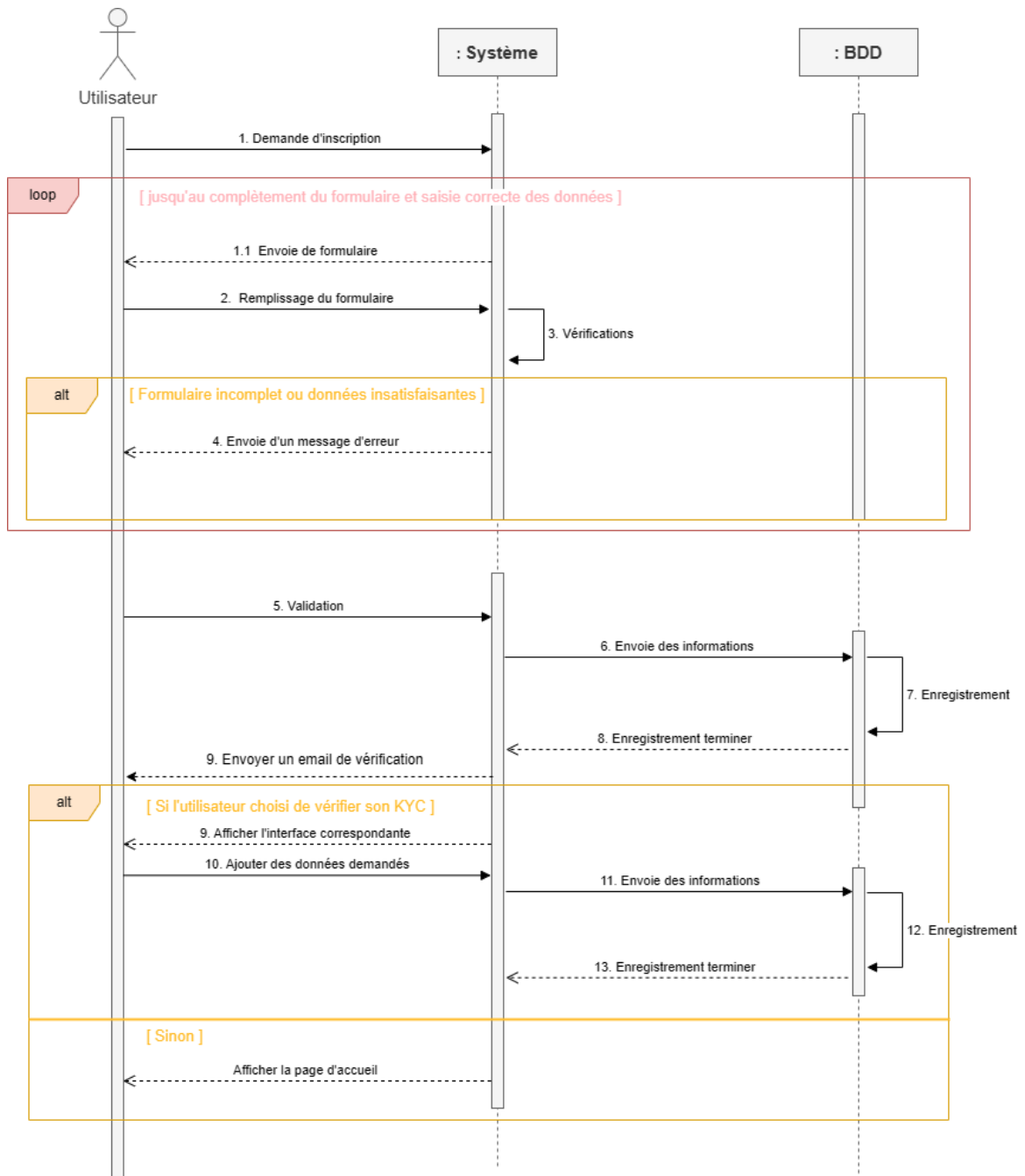


FIGURE 2.4 – Diagramme de séquence S’inscrire

2.3.2 Diagramme de séquence Système « S’authentifier »

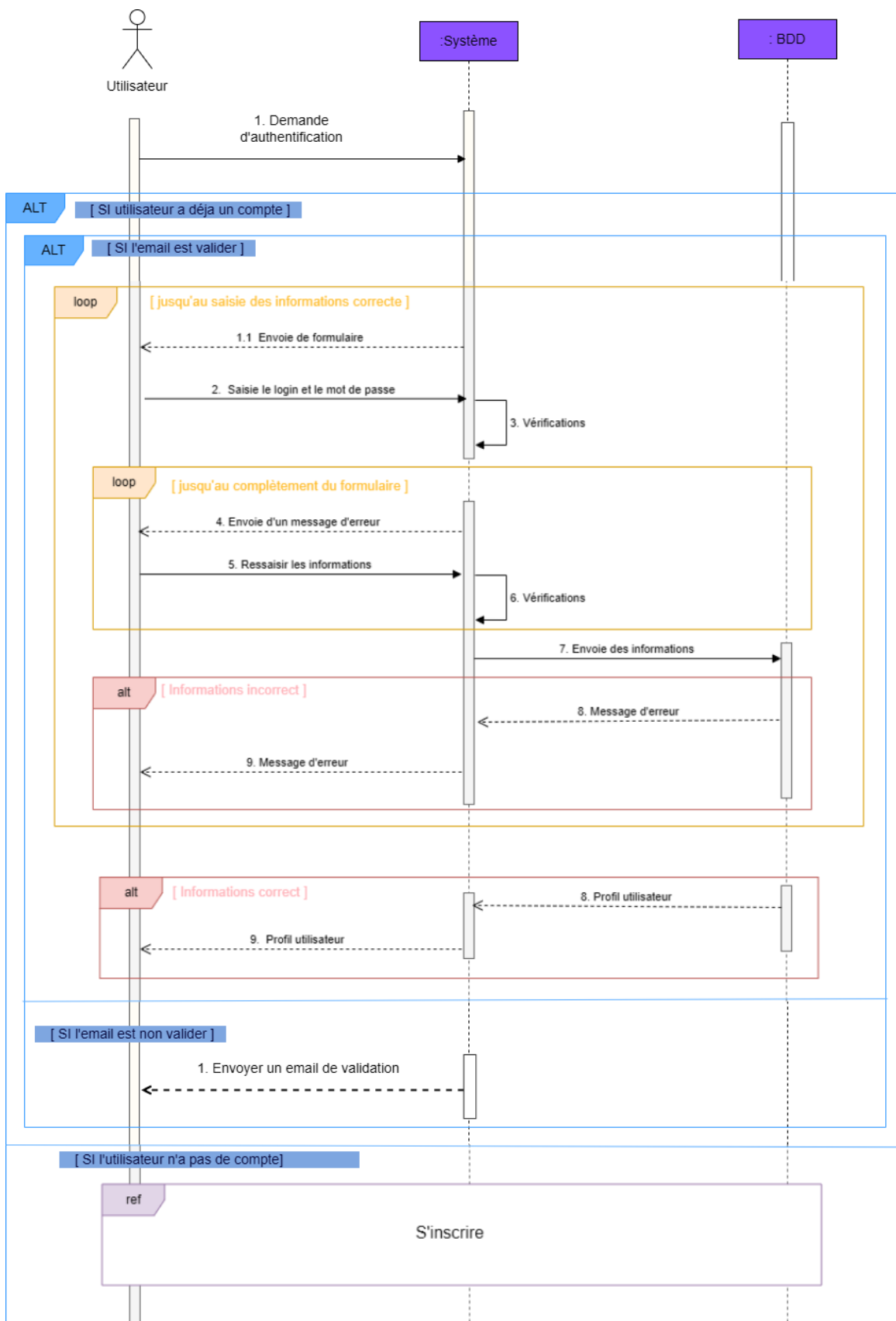


FIGURE 2.5 – Diagramme de séquence S’authentifier

2.3.3 Diagramme de séquence Système « Publier une mission »

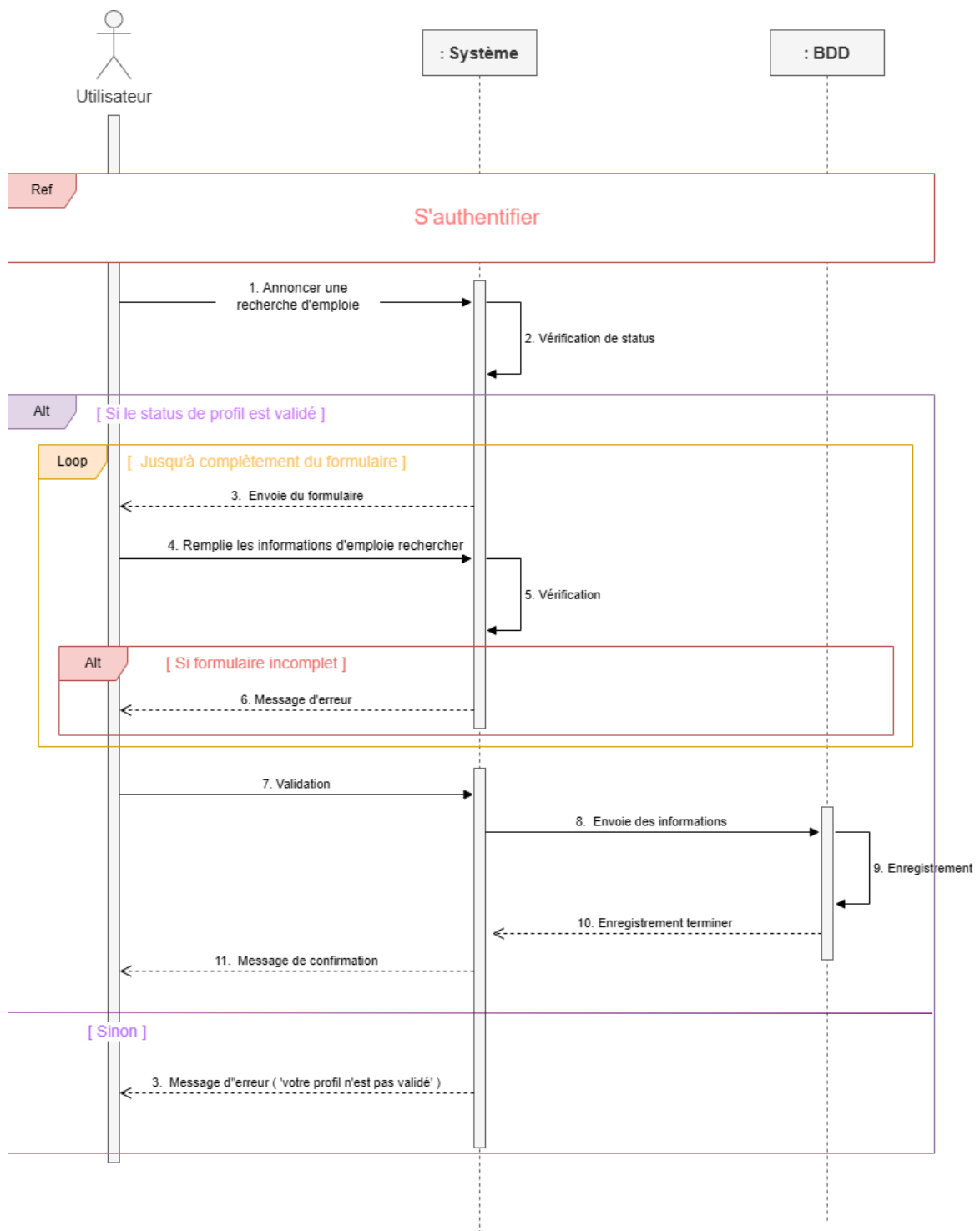


FIGURE 2.6 – Diagramme de séquence publier une mission

2.3.4 Diagramme de séquence Système « Postuler à une offre »

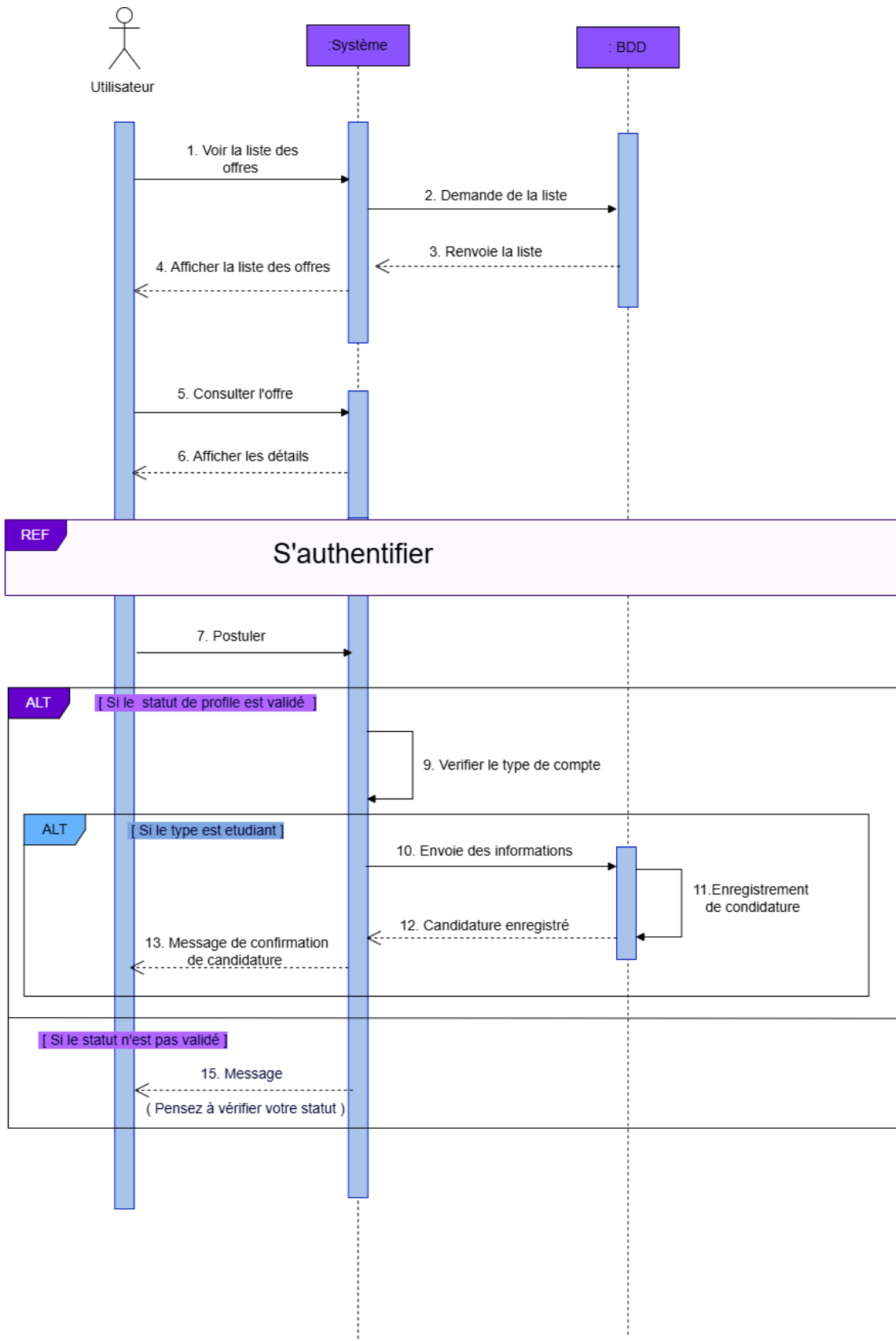


FIGURE 2.7 – Diagramme de séquence postuler à une offre

2.4 Diagramme de classe

Le diagramme de classes présenté dans la figure de notre écosystème reflète directement chaque entité du monde réel et traduit leur structure dans le système. permettant ainsi de visualiser comment les différentes entités interagissent entre elles au sein de l'application [21].

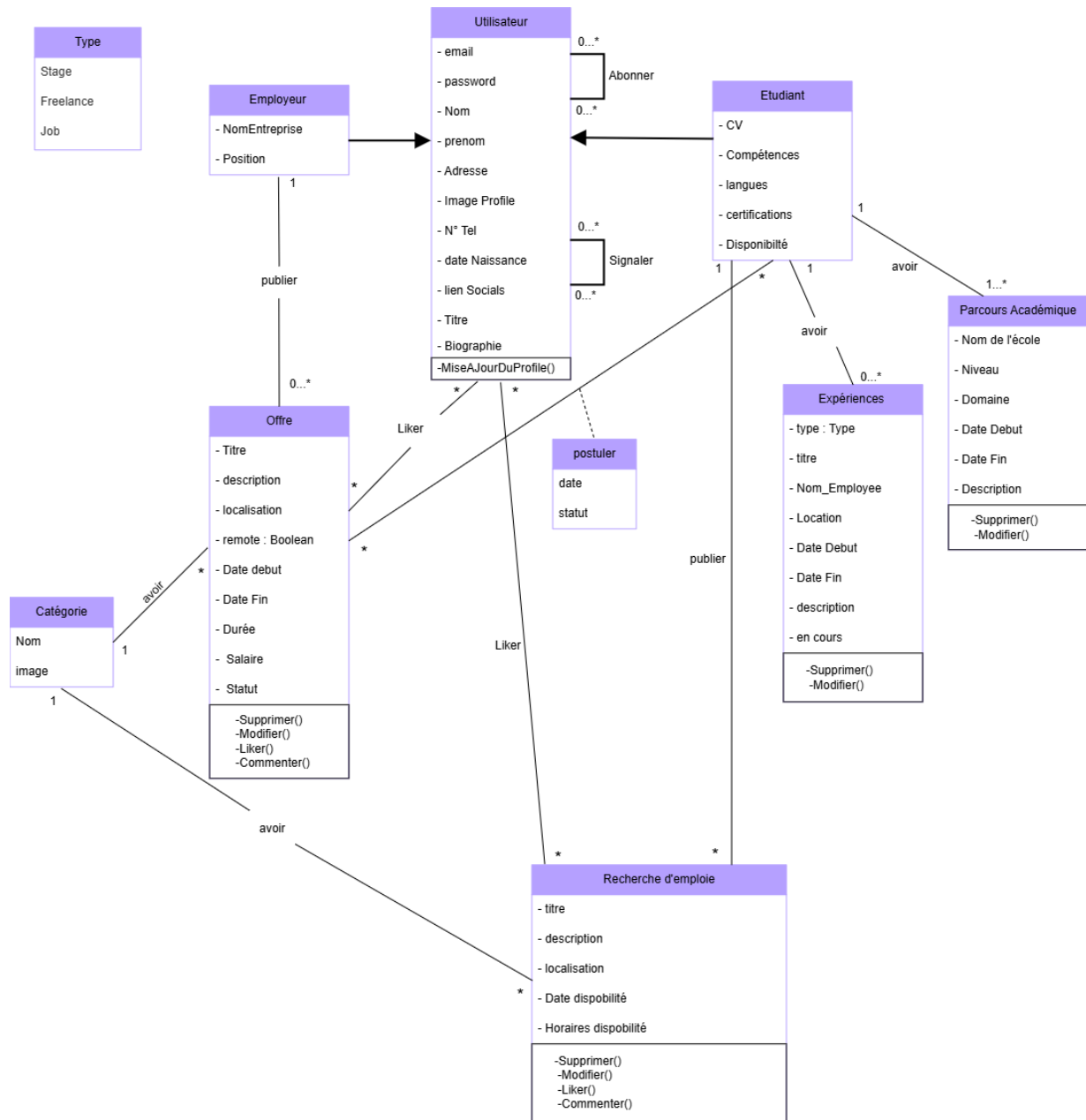


FIGURE 2.8 – Diagramme de classe

2.5 Diagramme de déploiement

Le diagramme de déploiement est un outil de modélisation qui illustre l'architecture physique du système. Il montre comment les composants logiciels sont déployés sur des nœuds matériels [22].

Ce diagramme est crucial pour la phase de déploiement d'une application distribuée comme JobLynk..

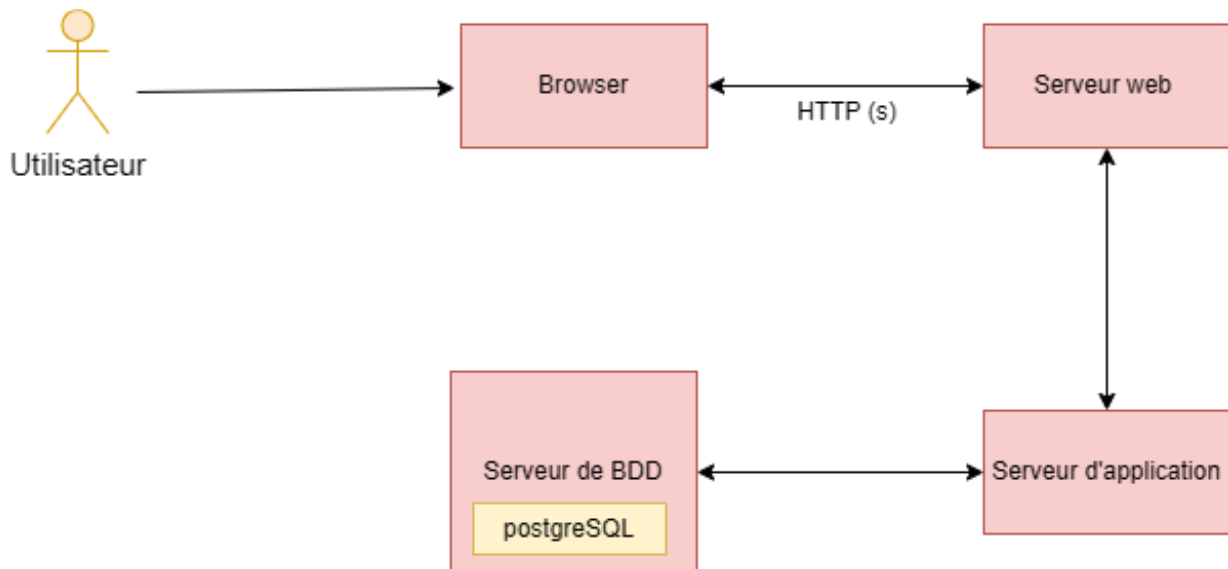


FIGURE 2.9 – Diagramme de déploiement

2.6 Modèle relationnel

Nous appelons modèle relationnel un ensemble de concepts permettant de formaliser logiquement la description d'articles de fichiers plats qui sont des fichiers de données contenant un enregistrement par ligne, indépendamment de la façon dont ils sont physiquement stockés dans une mémoire numérique. Le modèle relationnel inclut des concepts pour la description de données, ainsi que des concepts pour la manipulation de données [23].

En appliquant les règles de transformation d'un diagramme de classe vers un modèle relationnel, nous avons abouti au schéma relationnel suivant :

- **Employeur** (#idEmployeur, NomEntreprise, Position, ...)
- **Utilisateur** (idUtilisateur, Email, Password, Nom, Adresse, ImageProfile, ...)
- **Etudiant** (#idEtudiant, Cv, Compétences, Langues, ...)
- **Offre** (idOffre, #idEmployeur, #idCategorie, Titre, Description, ...)
- **Catégorie** (idCategorie, Nom)
- **Expérience** (idExperience, #idEtudiant, Type, Titre, ...)
- **ParcoursAcadémique** (idParcoursAcadémique, #idEtudiant, NomEcole, Niveau, ...)
- **RechercheD'Emploi** (idRechercheD'Emploi, #idCategorie, #idEtudiant, Titre, Description, ...)
- **Postuler** (idEtudiant, idOffre, Date, Statut)
- **Signaler** (idUtilisateur)
- **Abonner** (idUtilisateur)
- **LikerRechercheD'Emploi** (idRechercheD'Emploi, idUtilisateur)
- **LikerOffer** (idOffer, idUtilisateur)

2.7 Conclusion

Dans ce chapitre, nous avons entamé la phase de conception de l'écosystème JobLynk en commençant par l'analyse des besoins et l'identification des fonctionnalités essentielles. Ensuite, nous avons détaillé les cas d'utilisation ainsi que les diagrammes de séquence et de classes permettant de représenter graphiquement les interactions et l'architecture du système. Enfin, nous avons élaboré le modèle relationnel de données qui définit la structure de la base de

données de notre application.

Dans le chapitre suivant, nous aborderons l'implémentation des différentes solutions JobLynk, en présentant les technologies, les outils et l'environnement de développement utilisés. Des copies d'écran seront également fournies afin d'illustrer les principales fonctionnalités développées.

Chapitre 3

Implémentation

3.1 Introduction

Tout projet informatique passe une étape clé au moment du choix de son environnement et de ses outils de développement. Ce troisième chapitre expose les solutions techniques retenues pour concrétiser JobLynk, nos plateformes de micro-jobbing dédiées aux étudiants. Nous y détaillerons d'abord l'infrastructure logicielle globale : langages, frameworks, bases de données et services tiers, qui servent de fondation de l'application JobLynk mobile, de l'application web et de la console d'administration.

Ensuite, nous présenterons l'ensemble des outils de développement sur lesquels nous nous sommes appuyés. L'objectif est de mettre en lumière les raisons qui ont guidé nos choix (productivité, maintenabilité, performance, communauté) et de montrer comment chaque brique technique s'intègre dans l'architecture client-serveur envisagée.

Enfin, nous décrirons la mise en place de cet environnement, depuis l'installation locale jusqu'à la configuration des outils nécessaires, afin de garantir une intégration fluide et fiable des nouvelles fonctionnalités.

3.2 Outils de développement :

Cette section présente d'abord les langages et frameworks choisis pour JobLynk, en soulignant leurs atouts (productivité, maintenabilité, performance et soutien communautaire). Puis, nous détaillons les outils de collaboration et de maintenance qui ont permis d'assurer la qualité, la traçabilité et la cohérence de notre code.

3.3 Langages et frameworks

3.3.1 JavaScript & TypeScript

JavaScript domine le développement web modernes par sa flexibilité, sa rapidité et son usage côté client comme serveur [24].

TypeScript, améliore la qualité du code grâce au typage statique et à la détection précoce des erreurs [25].

Dans JobLynk, JavaScript et TypeScript sont utilisés à la fois pour le frontend (avec React et React Native) et le backend (avec Node.js et NestJS). L'utilisation de ces deux langages est essentielle pour maintenir une application moderne, réactive et bien structurée, tout en réduisant les bugs et en améliorant la productivité d'équipe.



FIGURE 3.1 – Logo JavaScript

3.3.2 Node.js

Node.js est un environnement d'exécution JavaScript côté serveur, optimisé pour les applications réseau rapides, évolutives et en temps réel nécessitant des connexions simultanées grâce à son modèle d'E/S non-bloquant [26].

Dans JobLynk, Node.js est utilisé pour développer le backend, gérant les requêtes API, l'interaction avec la base de données et la logique métier. En permettant l'utilisation de JavaScript des deux côtés (client et serveur), Node.js simplifie le développement et la maintenance, tout en offrant des performances élevées pour des applications comme la gestion des notifications en temps réel.



FIGURE 3.2 – Logo Node.js

3.3.3 NestJS

NestJS est un framework Node.js bas sur TypeScript, offrant une architecture modulaire qui structure le code et facilite la maintenance ainsi que l'extensibilité des applications [27].

Dans JobLynk, NestJS est utilisé pour structurer le backend, assurant une gestion modulaire et une maintenance facilitée. Grâce à son intégration native avec TypeScript, il optimise le développement tout en offrant un environnement solide pour les tests unitaires et d'intégration, garantissant ainsi une haute qualité du code.



FIGURE 3.3 – Logo NestJS

3.3.4 React.js (Framework web)

React est une bibliothèque JavaScript pour créer des interfaces modernes et interactives, optimisée grâce à la mise à jour ciblée des éléments et à la réutilisation de composants, ce qui rend le développement plus rapide, simple et efficace [28].

Dans JobLynk, React est utilisé pour développer l'interface web destinée aux micro-jobbing, et de l'administration. Cette technologie permet de construire une interface fluide, réactive et offrant une expérience utilisateur optimale, tout en simplifiant la gestion des composants.



FIGURE 3.4 – Logo React.js

3.3.5 React Native (Framework Mobile)

React Native est un framework mobile basé sur JavaScript et React qui permet de partager un même code pour iOS et Android, tout en offrant accès aux fonctionnalités natives, réduisant

ainsi le temps et le coût de développement [29].

Dans JobLynk, React Native est utilisé pour développer l'application mobile. Cette technologie garantit une expérience utilisateur fluide et cohérente, tout en simplifiant le processus de développement grâce à la réutilisation du code.



FIGURE 3.5 – Logo React Native

3.3.6 TypeORM

TypeORM est un ORM (Object-Relational Mapping) pour JavaScript et TypeScript qui facilite l'interaction avec les bases de données via des objets plutôt que des requêtes SQL, rendant le code plus lisible, structuré et moins sujet aux [30].

Dans JobLynk, TypeORM est utilisé pour gérer les opérations liées aux bases de données, comme la création, la mise à jour et la suppression des données, sans avoir à écrire de longues requêtes SQL. Cela permet une intégration fluide entre le code métier et la base de données.



FIGURE 3.6 – Logo TypeORM

3.4 Outils de développement et de gestion

3.4.1 PostgreSQL

PostgreSQL est un SGBDR open-source reconnu pour sa robustesse et sa conformité SQL, offrant des fonctionnalités avancées et une grande extensibilité, idéal pour les applications exigeant haute disponibilité et intégrité des données [31].

Dans JobLynk, PostgreSQL est utilisé pour gérer l'ensemble des données de l'application. Sa capacité à gérer des transactions complexes et à assurer la cohérence des données est cruciale pour maintenir l'intégrité du système. De plus, PostgreSQL est conçu pour supporter un grand nombre d'utilisateurs et de transactions simultanées, ce qui garantit des performances optimales même à grande échelle. Cela permet à l'application de fonctionner de manière fluide et fiable, même avec une forte charge d'utilisateurs.



FIGURE 3.7 – Logo PostgreSQL

3.4.2 Visual Studio Code

Visual Studio Code est un éditeur de code gratuit et open source, léger et performant, offrant un large support de langages, des extensions variées, ainsi que des outils comme IntelliSense, le débogage intégré et Live Share pour la collaboration, il devient un outil incontournable pour le développement de l'application JobLynk [32].



FIGURE 3.8 – Logo VS Code

3.4.3 Git & GitHub

Git est un système de gestion de version qui enregistre les modifications du code via des commits, permettant de revenir en arrière si nécessaire [33]. GitHub, basé sur Git, héberge les projets et facilite le travail collaboratif grâce aux branches [34].

Pour le projet JobLynk, Git et GitHub ont été des outils essentiels pour organiser et suivre le développement sur plusieurs plateformes (web et mobile). Ces outils ont facilité la collaboration entre les différents intervenants, permettant à chacun de travailler sur des fonctionnalités spécifiques sans craindre de perturber le travail des autres.



FIGURE 3.9 – Logo GitHub

3.4.4 Docker

Docker est une plateforme de virtualisation qui exécute les applications dans des conteneurs, assurant leur cohérence entre développement, test et production, tout en simplifiant le déploiement [35].

Dans JobLynk, Docker a été utilisé pour containeriser la base de données PostgreSQL et Node.js. Cela permet de garantir que les versions de ces technologies sont cohérentes dans tous les environnements, de la phase de développement à la mise en production. En isolant PostgreSQL et Node.js dans des conteneurs, Docker a garanti une gestion uniforme des versions et une configuration constante, éliminant ainsi les conflits de version et les problèmes de compatibilité. Cette approche nous a permis de travailler dans un environnement identique, quel que soit le stade du projet, tout en facilitant le déploiement rapide et fiable des applications.



FIGURE 3.10 – Logo Docker

Ces technologies et outils ont été essentiels pour le développement et la gestion efficace de JobLynk, en garantissant des performances optimales, une scalabilité et une expérience utilisateur fluide. Grâce à leur intégration cohérente, nous avons pu construire un système robuste et adapté aux besoins spécifiques de chaque utilisateur, tout en facilitant la collaboration et le déploiement rapide des nouvelles fonctionnalités.

3.5 Schéma de navigation de l'application

Dans cette section, nous détaillons le schéma de navigation des trois applications composant le système JobLynk : l'application web, l'application mobile, et l'application admin. Chacune de ces applications a une interface adaptée à son utilisateur et une organisation spécifique de ses fonctionnalités.

Les captures d'écran suivantes illustrent le parcours typique de chaque utilisateur au sein de son application respective. Chaque flux de navigation montre comment l'utilisateur interagit avec les différentes sections.

3.5.1 Logo JobLynk



FIGURE 3.11 – Logo JobLynk

3.5.2 Inscription / Connexion

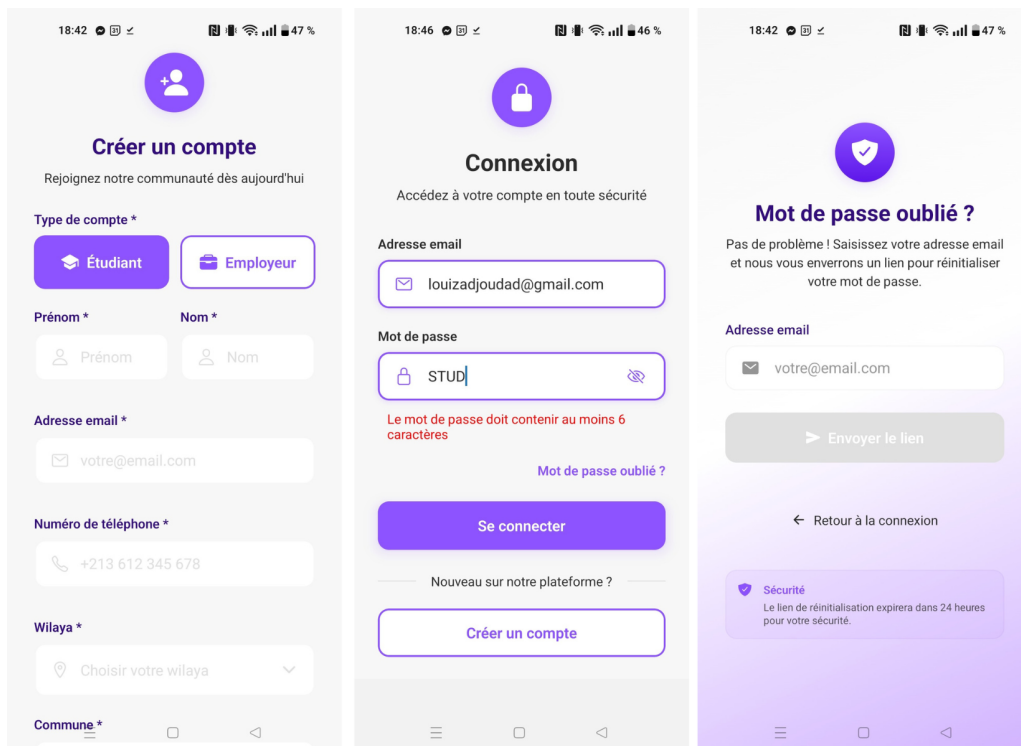


FIGURE 3.12 – Interfaces d'authentification (Inscription/Connexion)

3.5.3 Interface Offres d'emplois

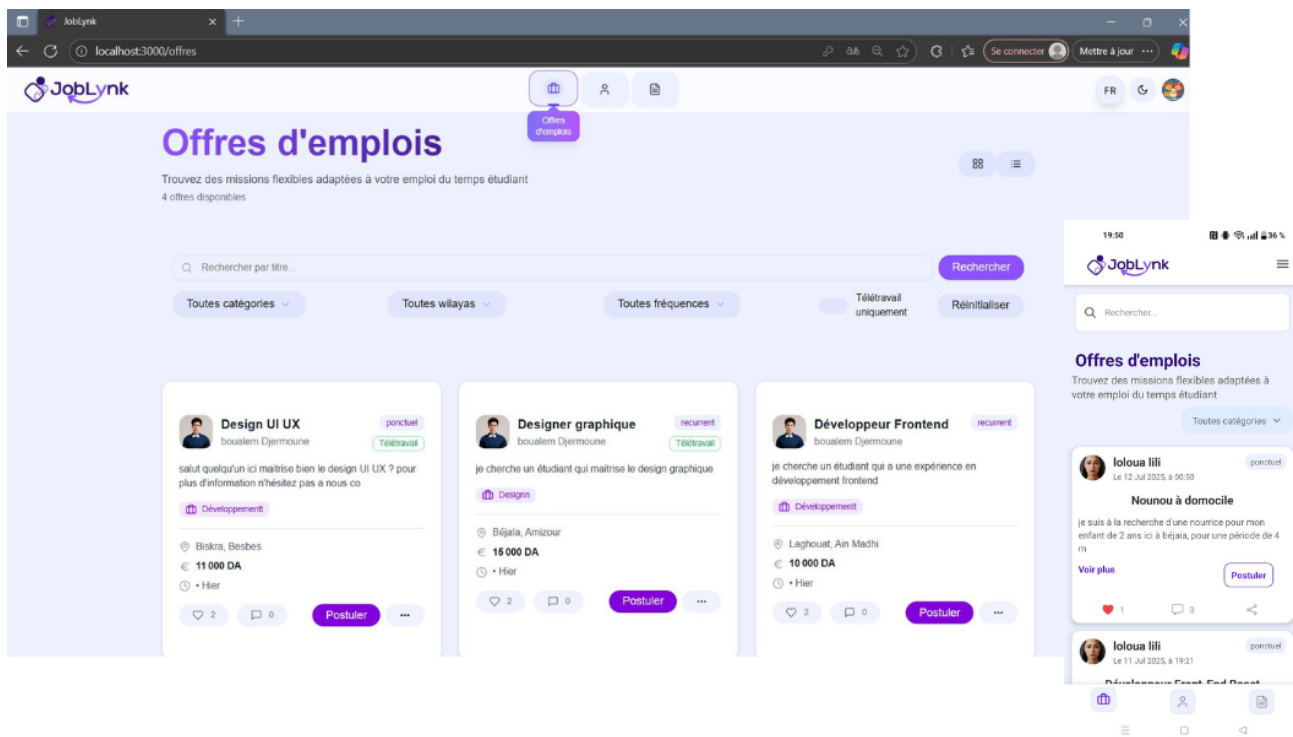


FIGURE 3.13 – Interface des offres d'emploi

3.5.4 Interface Profils publics et messagerie

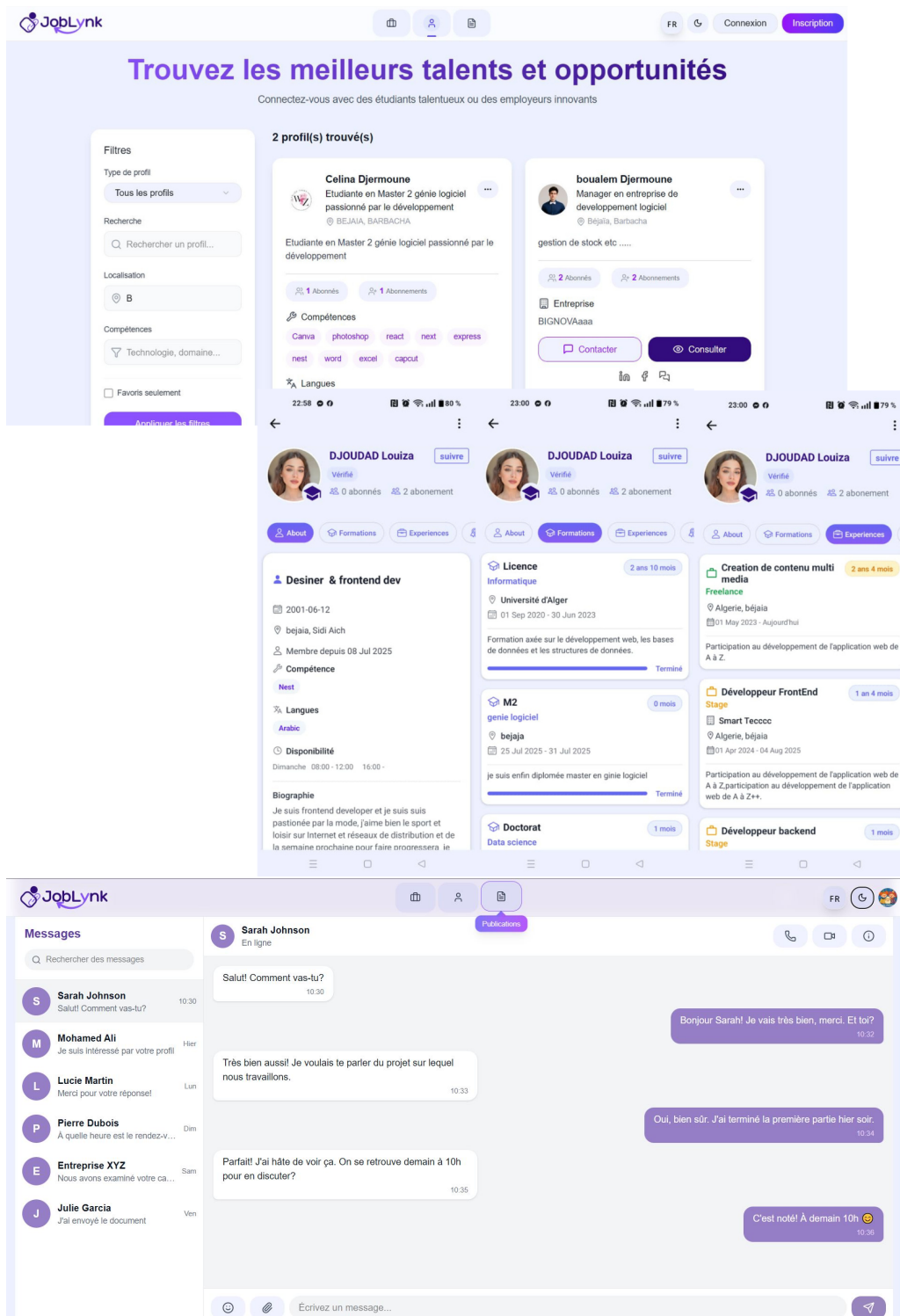


FIGURE 3.14 – Interface des profils publics et messagerie

3.5.5 Page de publications

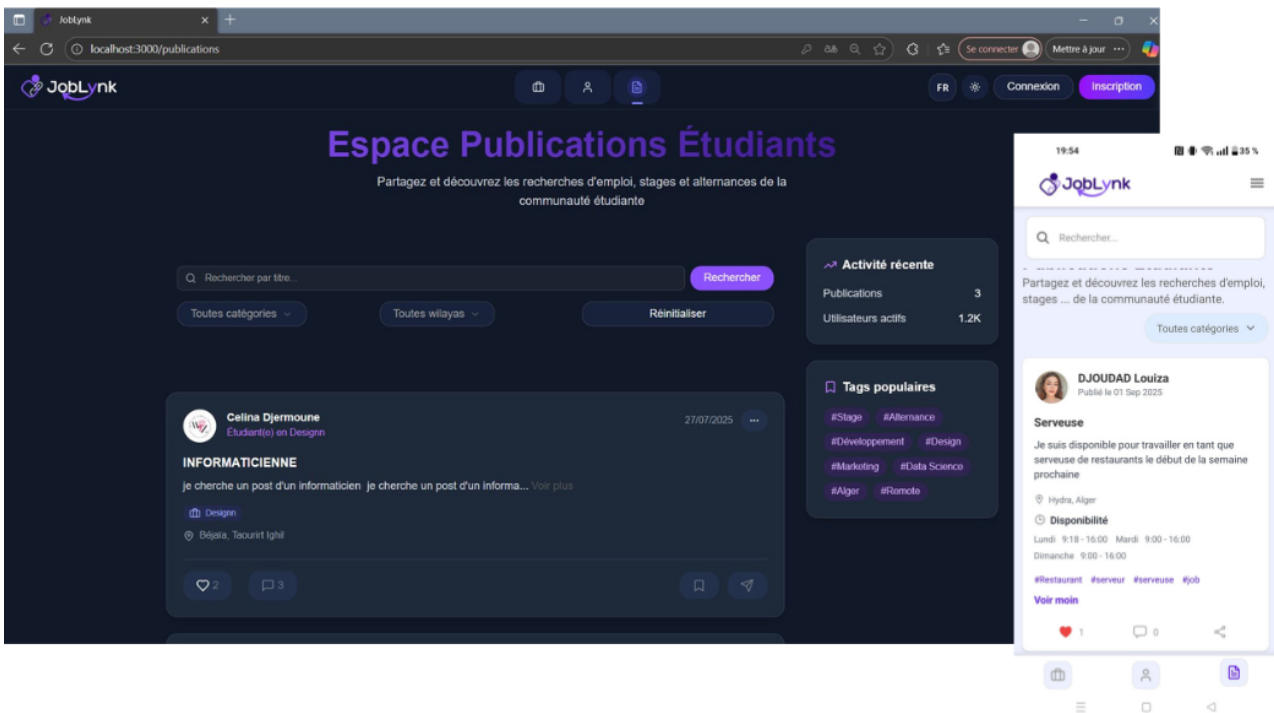


FIGURE 3.15 – Page de publications

3.5.6 Page Profile connecté

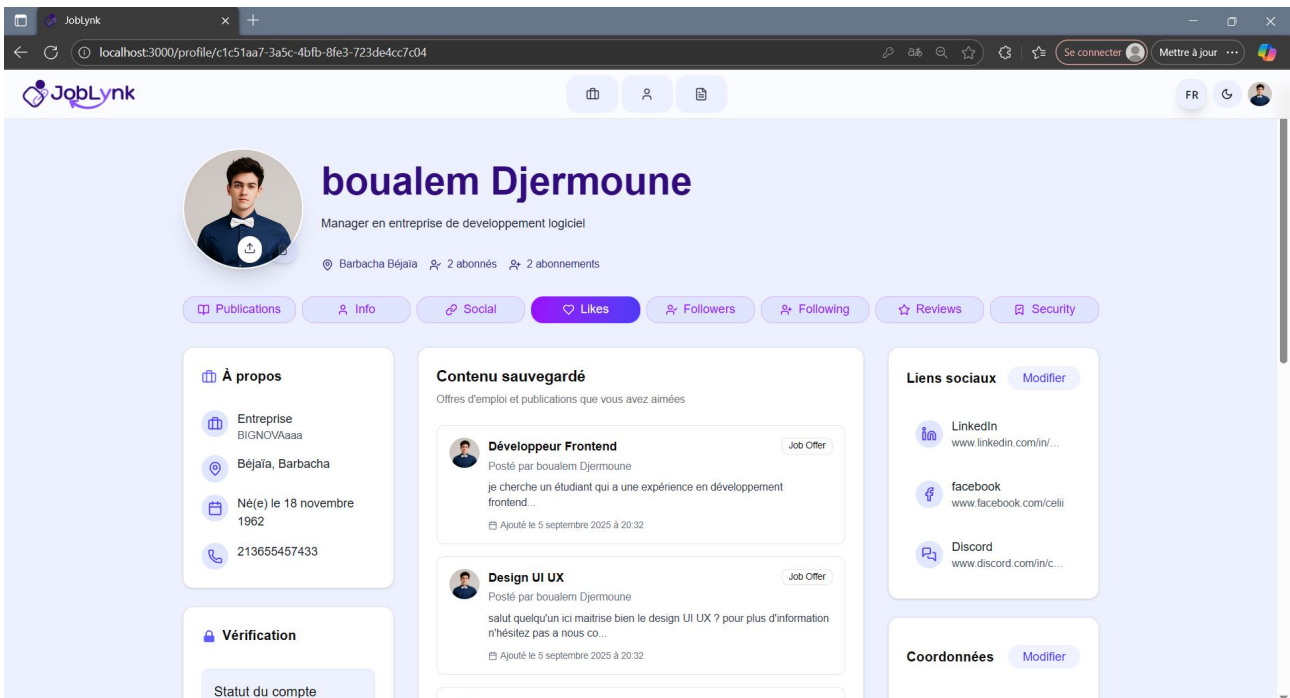


FIGURE 3.16 – Page de mon profile

3.5.7 Espace admin (pour l'équipe du site)

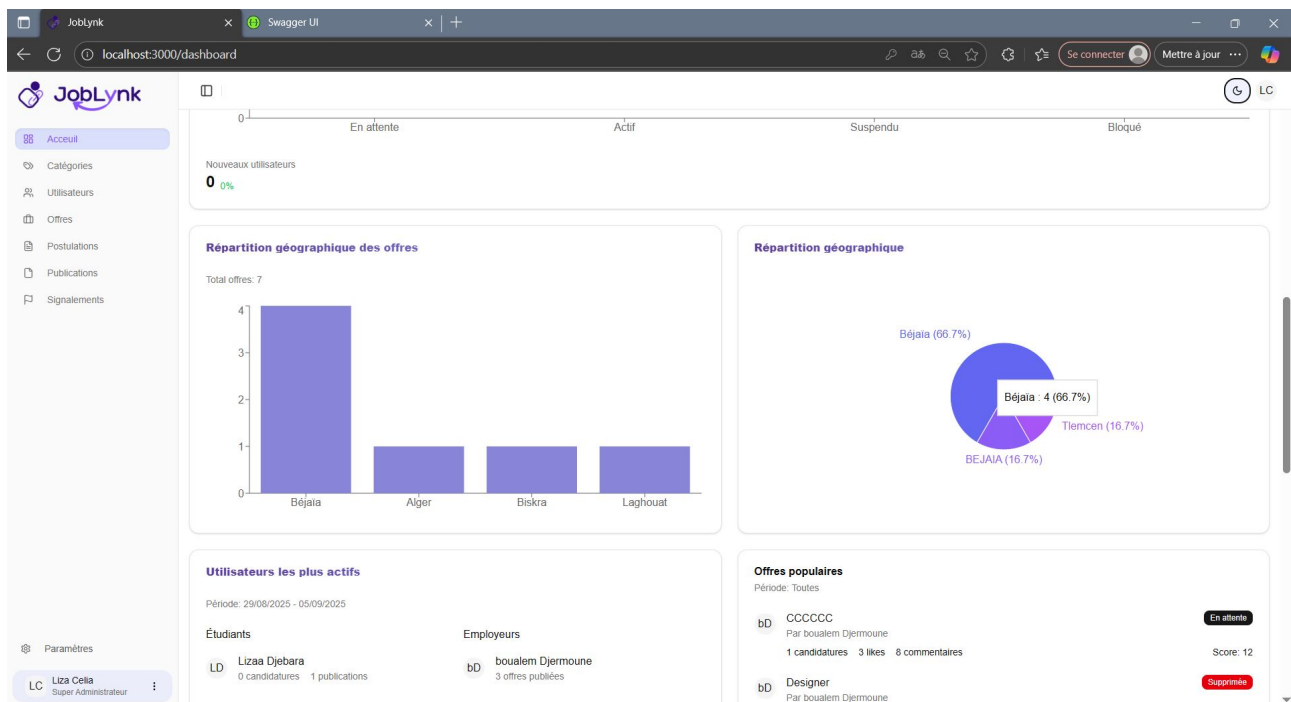


FIGURE 3.17 – Espace administrateur

3.6 Conclusion

Ce chapitre a mis en évidence les choix technologiques et les outils de développement qui ont permis de concrétiser la plateforme JobLynk. Les langages, frameworks et services utilisés, tels que JavaScript/TypeScript, React, React Native, Node.js, NestJS, TypeORM et PostgreSQL, ont été sélectionnés pour garantir performance, maintenabilité et évolutivité.

De plus, l'intégration d'outils modernes de collaboration et de gestion comme GitHub, Docker et Visual Studio Code a favorisé une organisation efficace, une meilleure traçabilité et une homogénéité entre les environnements de développement et de production.

En somme, l'implémentation de ces solutions a permis de poser des bases solides pour le développement des différentes applications (web, mobile et admin), assurant ainsi un système robuste, fluide et adapté aux besoins des utilisateurs.

Conclusion et perspectives

Dans le cadre de ce mémoire de fin d'études, nous avons conçu et développé JobLynk, une plateforme numérique innovante de micro-jobbing dédiée aux étudiants. L'objectif de ce projet est de faciliter la mise en relation entre les étudiants en quête de missions ponctuelles, de stages ou de petits emplois, et les recruteurs (particuliers, PME, entreprises) à la recherche de profils compétents, disponibles et adaptés à leurs besoins.

Pour mener à bien ce projet, nous avons adopté le processus unifié (UP), qui nous a permis d'organiser le travail en plusieurs phases progressives (inception, élaboration, construction et transition). Ce choix méthodologique a facilité la planification et la gestion des risques, tout en garantissant une évolution structurée de la solution depuis l'analyse des besoins jusqu'à la mise en œuvre des différentes fonctionnalités.

Sur le plan technique, JobLynk s'appuie sur un ensemble d'outils et de technologies modernes :

- React.js pour l'interface web destinée aux étudiants et recruteurs.
- React Native pour l'application mobile multiplateforme, assurant une expérience fluide sur Android et iOS.
- Node.js et NestJS pour le backend, avec une architecture modulaire et extensible.
- PostgreSQL pour la gestion des données, associé à TypeORM pour simplifier les interactions avec la base.
- Docker pour la containerisation et l'homogénéité des environnements.
- Git/GitHub et Visual Studio Code pour la gestion collaborative du code et la productivité.

Le système JobLynk est composé de trois applications principales :

- Une application web multi-profil (étudiant/recruteur), permettant la création de comptes, la publication et la consultation d'offres, ainsi que la recherche et la candidature.
- Une application mobile multi-profil (étudiant/recruteur), offrant les mêmes fonctionnalités pour garantir une accessibilité optimale aux utilisateurs.
- Une application administrateur (back-office), destinée à la supervision et à la gestion globale de la plateforme (utilisateurs, offres, statistiques, retours d'expérience).

Parmi les fonctionnalités majeures réalisées, nous pouvons citer :

- La gestion multi-profils (étudiant/recruteur).
- Un système de recherche et de filtrage avancé.
- La possibilité de sauvegarder et de liker des offres.
- La possibilité de publier une offre ou une recherche d'emploi, de visiter les profils et de

- postuler aux offres.
- Une interface ergonomique, moderne et responsive.
- Une architecture modulaire et évolutive garantissant la scalabilité du projet.

JobLynk constitue aujourd’hui une solution fonctionnelle, pensée pour répondre à une problématique réelle : l’accès des étudiants à des opportunités professionnelles et la facilitation du recrutement local. En numérisant cette démarche, le projet contribue à l’amélioration de l’employabilité des étudiants et à la dynamisation du marché de l’emploi à petite échelle.

Néanmoins, plusieurs axes d’amélioration et d’extension sont envisageables afin de renforcer l’impact de la plateforme :

- Intégration d’un système de notifications en temps réel (nouvelles offres, candidatures).
- Développement d’une messagerie interne étudiant-recruteur pour fluidifier la communication.
- Mise en place d’un système d’évaluation et de recommandation intelligente afin d’optimiser les correspondances.
- Développement de fonctionnalités premium (statistiques avancées, services de visibilité renforcée).

En somme, ce projet nous a permis de mettre en pratique nos acquis en génie logiciel et d’expérimenter un cycle complet de développement, de la conception à la réalisation. Il nous a également permis de renforcer notre sens de l’organisation, notre capacité à collaborer efficacement et notre compétence à développer une solution numérique innovante répondant à des besoins concrets du terrain.

Résumé

Ce mémoire s'inscrit dans le cadre d'un projet de fin de cycle de Master en Informatique. Il porte sur la conception et la réalisation d'une plateforme numérique baptisée JobLynk, destinée à faciliter la mise en relation entre étudiants à la recherche de micro-jobs et particuliers, petites entreprises ou recruteurs ayant des besoins ponctuels.

La plateforme offre la possibilité aux étudiants de créer et de valoriser leur profil, de rechercher des missions en fonction de leurs compétences, de leur localisation et de leur disponibilité, tandis que les recruteurs peuvent publier des offres et consulter des profils adaptés.

Le développement du projet s'est appuyé sur la méthodologie UP (Unified Process), permettant un cycle de vie itératif et incrémental, intégrant à chaque phase l'analyse, la conception, l'implémentation et les tests. La solution a été développée en utilisant des technologies modernes du développement web, à savoir : Node.js et Express.js pour le backend, React.js pour le frontend, et PostgreSQL pour la gestion des données associé à TypeORM pour simplifier les interactions avec la base.

La plateforme répond efficacement aux besoins fonctionnels définis, en facilitant la recherche et la consultation de missions, tout en respectant les exigences non fonctionnelles telles que la sécurité, la fiabilité et la facilité d'utilisation.

Mots clés : Micro-jobs, Étudiants, Recruteurs, Emploi ponctuel, JobLynk, UP, UML.

Abstract

This thesis is part of a Master's final project in Computer Science. It focuses on the design and development of a digital platform called JobLynk, aimed at facilitating connections between students seeking micro-jobs and individuals, small businesses, or recruiters with occasional staffing needs.

The platform allows students to create and showcase their profiles, search for tasks based on their skills, location, and availability, while recruiters can post job offers and consult suitable profiles.

The project was developed using the Unified Process (UP) methodology, enabling an iterative and incremental life cycle that integrates analysis, design, implementation, and testing at each phase. The solution was implemented using modern web development technologies : Node.js and Express.js for the backend, React.js for the frontend, and PostgreSQL with TypeORM to manage database interactions efficiently.

The platform effectively meets the defined functional requirements by facilitating task search and profile consultation, while also satisfying non-functional requirements such as security, reliability, and ease of use.

Keywords : Micro-jobs, Students, Recruiters, Occasional employment, JobLynk, UP, UML.

Références

- [1] OCDE. Youth and the labour market in the mediterranean region. Organisation de coopération et de développements économiques, 2020. [En ligne]. Disponible sur : <https://www.oecd.org/>.
- [2] Banque Mondiale. Économie numérique pour l’algérie – rapport de diagnostic. Banque mondiale, 2023. [En ligne]. Disponible sur : <https://www.worldbank.org>.
- [3] Organisation Internationale du Travail (OIT). Tendances mondiales de l’emploi des jeunes 2022. Genève : OIT, 2022. [En ligne]. Disponible sur : <https://www.ilo.org>.
- [4] Office National des Statistiques (ONS). Statistiques de l’emploi et du chômage en algérie – rapport annuel 2023. Alger : ONS, 2023. [En ligne]. Disponible sur : <http://www.ons.dz>.
- [5] Social et Environnemental (CNESE) Conseil National Économique. Insertion professionnelle des jeunes diplômés en algérie. Alger : CNESE, 2022.
- [6] Ministère de la Poste et des Télécommunications. Algérie numérique 2025 – stratégie nationale. Alger : MPT, 2023. [En ligne]. Disponible sur : <https://www.mpt.gov.dz>.
- [7] Startups.dz. Annuaire des startups algériennes. Startups.dz, 2024. [En ligne]. Disponible sur : <https://startups.dz>.
- [8] Agence Nationale de Soutien à l’Emploi des Jeunes (ANSEJ). Rapport d’activités 2022. Alger : ANSEJ, 2022.
- [9] TaskRabbit. Company information & services. TaskRabbit, 2023. [En ligne]. Disponible sur : <https://www.taskrabbit.com/about>.
- [10] Upwork Global Inc. Research & reports. Upwork, 2023. [En ligne]. Disponible sur : <https://www.upwork.com/research/>.
- [11] Statista. Gig economy platforms - global statistics report 2023. Statista, 2023. [En ligne]. Disponible sur : <https://www.statista.com>.
- [12] I. Jacobson, G. Booch, and J. Rumbaugh. *The Unified Software Development Process*. Addison-Wesley, Boston, MA, 1999.
- [13] P. Kruchten. *The Rational Unified Process : An Introduction*. Addison-Wesley, Boston, MA, 3rd edition, 2003.
- [14] An introduction to the uml and the unified process. SpringerLink, 2003. [En ligne]. Disponible sur : https://link.springer.com/chapter/10.1007/1-85233-856-3_3.
- [15] Uml and the unified process. Academia.edu. [En ligne]. Disponible sur : https://www.academia.edu/8751505/UML_and_the_Unified_Process.
- [16] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, Boston, MA, 2nd edition, 2005.
- [17] M. Blaha and J. Rumbaugh. *Object-Oriented Modeling and Design with UML*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2005.
- [18] UML Diagrams. Unified modeling language (uml) description. [En ligne]. Disponible sur : <https://www.uml-diagrams.org/>.

- [19] IBM. Defining use cases. IBM Documentation, 2025. [En ligne]. Disponible sur : <https://www.ibm.com/docs/en/engineering-lifecycle-management-suite/doors-next/7.0.3?topic=requirements-defining-use-cases> (consulté le 25 mai 2025).
- [20] Lucidchart. Qu'est-ce qu'un diagramme de séquence uml. Lucidchart, 2025. [En ligne]. Disponible sur : <https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml> (consulté le 26 mai 2025).
- [21] Lucidchart. Qu'est-ce qu'un diagramme de classes uml. Lucidchart, 2025. [En ligne]. Disponible sur : <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml> (consulté le 26 mai 2025).
- [22] Visual Paradigm. Deployment diagram. Visual Paradigm Documentation, 2025. [En ligne]. Disponible sur : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/> (consulté le 26 mai 2025).
- [23] LibreCours.net. Introduction au modèle relationnel. LibreCours, 2025. [En ligne]. Disponible sur : <https://librecours.net/modules/bdd/relationnel/solweb/co/definition.html> (consulté le 26 mai 2025).
- [24] Mozilla Contributors. Javascript. Mozilla Developer Network, 2025. [En ligne]. Disponible sur : <https://developer.mozilla.org/fr/docs/Web/JavaScript> (consulté le 20 avril 2025).
- [25] TypeScript Contributors. Typescript. TypeScript, 2025. [En ligne]. Disponible sur : <https://www.typescriptlang.org/> (consulté le 20 avril 2025).
- [26] Node.js Foundation. Node.js. Node.js Official Website, 2025. [En ligne]. Disponible sur : <https://nodejs.org/> (consulté le 20 avril 2025).
- [27] NestJS Contributors. Nestjs - a progressive node.js framework. NestJS, 2025. [En ligne]. Disponible sur : <https://nestjs.com/> (consulté le 21 avril 2025).
- [28] React Contributors. React - a javascript library for building user interfaces. React, 2025. [En ligne]. Disponible sur : <https://reactjs.org/> (consulté le 21 avril 2025).
- [29] React Native Contributors. React native - a framework for building native apps using react. React Native, 2025. [En ligne]. Disponible sur : <https://reactnative.dev/> (consulté le 21 avril 2025).
- [30] TypeORM Contributors. Typeorm - an orm for typescript and javascript. TypeORM, 2025. [En ligne]. Disponible sur : <https://typeorm.io/> (consulté le 21 avril 2025).
- [31] PostgreSQL Global Development Group. Postgresql : The world's most advanced open-source database. PostgreSQL, 2023. [En ligne]. Disponible sur : <https://www.postgresql.org/> (consulté le 22 avril 2025).
- [32] Microsoft. Visual studio code. Microsoft, 2025. [En ligne]. Disponible sur : <https://code.visualstudio.com/> (consulté le 22 avril 2025).
- [33] Git Contributors. Git - version control system. Git, 2025. [En ligne]. Disponible sur : <https://git-scm.com/> (consulté le 22 avril 2025).
- [34] Inc. GitHub. Github - a platform for software development and version control. GitHub, 2025. [En ligne]. Disponible sur : <https://github.com/> (consulté le 22 avril 2025).
- [35] Inc. Docker. Docker - a platform for building, shipping, and running applications. Docker, 2025. [En ligne]. Disponible sur : <https://www.docker.com/> (consulté le 22 avril 2025).
- [36] Fiverr International Ltd. Fiverr investor relations. Fiverr, 2023. [En ligne]. Disponible sur : <https://investors.fiverr.com>.
- [37] B. Bruegge and A. H. Dutoit. *Object-Oriented Software Engineering Using UML, Patterns, and Java*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2009.

[38] OpenClassrooms. Transformez les associations de votre diagramme de classes uml. OpenClassrooms, 2025. [En ligne]. Disponible sur : <https://openclassrooms.com/fr/courses/6938711-modelisez-vos-bases-de-donnees/7507122-transformez-les-associations-de-votre-diagramme-de-classes-uml> (consulté le 26 mai 2025).