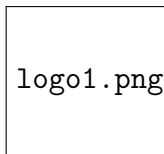
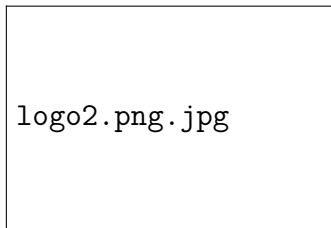


République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Sousse



Polytechnique Sousse



DÉPARTEMENT INFORMATIQUE
RAPPORT DE STAGE DE FIN D'ÉTUDES

En vue de l'obtention du diplôme National d'Ingénieur en :
Génie Informatique

Conception et development d'une solution de
gestion des candidats "TalenCloud" pour une
ESN Belge

Elaboré par :

Ben Akkari Bacem

Encadré par :

Mr Haythem Saoudi

Encadrante Académique

Mme Sondos Aouachri

Encadrante professionnelle

Année Universitaire : 2024 - 2025

Dédicaces

À mes parents :

Je vous remercie profondément pour votre amour, vos sacrifices et votre soutien constant. Vous avez toujours cru en moi et m'avez donné la force d'avancer malgré les difficultés. Votre présence a été essentielle tout au long de mon parcours. Vous êtes les véritables piliers de ma réussite.

À mes sœurs :

Merci pour votre affection, votre patience et votre écoute dans les moments difficiles. Votre soutien moral m'a beaucoup aidé durant ces années d'études. Vous avez su m'apporter du réconfort et de la sérénité. Je vous suis sincèrement reconnaissant pour cela.

À chers amis :

Votre présence a été précieuse tout au long de cette aventure. Merci pour les encouragements, les échanges et les souvenirs partagés. Vous avez su rendre cette période plus agréable et motivante. Votre amitié a été un vrai moteur dans mon quotidien.

À tous mes professeurs et encadrants :

Je vous remercie pour votre disponibilité, vos conseils et votre patience. Votre accompagnement a grandement contribué à ma formation. Chaque enseignement reçu a enrichi mon parcours personnel et académique. Votre implication a été pour moi une réelle source d'inspiration.

Remerciement

Afin de m'assurer du succès de mon stage et de mon projet, je suis redevable à de nombreuses personnes.

Tout d'abord, je tiens à remercier, en premier lieu, l'ensemble de **l'équipe pédagogique de l'École Polytechnique de Sousse**, ainsi que les intervenants professionnels responsables de la formation du Cycle d'Ingénieur en Génie Informatique, pour leur engagement dans la transmission des connaissances théoriques.

Je souhaite également exprimer ma profonde gratitude à mon encadrant académique, **Monsieur Haythem saoudi**, pour son accompagnement et ses conseils avisés durant ce travail. Leur expertise et leur bienveillance ont été indispensables à l'aboutissement de ce projet.

Je voudrais aussi exprimer mes profonds remerciements à mon encadrante professionnelle, **Madame Sondos Aouachri**, pour son aide précieuse et son soutien tout au long de la réalisation de ce projet.

Enfin, je tiens à remercier chaleureusement les membres du jury pour leur temps et leur expertise dans l'évaluation de mon travail. Je suis honoré qu'ils aient accepté de juger ce projet et j'espère qu'il a été à la hauteur de leurs attentes.

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Introduction générale	1
1 Contexte général du projet	2
I Introduction	2
II Entreprise d'accueil	2
II.1 Présentation de l'entreprise	2
II.2 Spécialisations	3
II.3 Les services de l'entreprise	3
II.4 Fiche signalétique	4
III Fondements du projet	4
III.1 Microservices et React	4
III.2 API Gateway	5
IV Cadre du projet	6
IV.1 Contexte Général et problématique	6
1 Présentation du projet	6
2 Problématique	6
IV.2 Étude de l'existant	6
1 EmploiTunisie.com	7
2 TanitJobs.com	7
3 KeeJob.com	8
IV.3 Critique de l'existant	9
IV.4 Solution Proposée	9
IV.5 Etude de la methodologie de développement	10
1 Choix de la methodologie de développement	10
2 Avantages de Scrum	10
3 Mise en pratique de la méthodologie SCRUM	11
IV.6 Technologies utilisées	12
1 Langages et Frameworks de programmation	12

2	Plateformes de développement utilisées	13
3	Autres Technologies	14
V	Conclusion	15
2	Analyse et spécification des besoins	16
I	Introduction	16
II	Identification des acteurs	16
III	Identification des besoins	16
III.1	Besoins Fonctionnels	16
III.2	Administrateur	17
III.3	Candidat	17
III.4	Recruteur	17
III.5	Système IA	17
IV	Besoins non fonctionnels	18
IV.1	Convivialité	18
IV.2	Performance	18
IV.3	Sécurité	18
IV.4	Compatibilité de l'application	19
IV.5	Stockage et gestion des données	19
V	Elaboration du backlog produit	20
VI	Conclusion	20
3	Etude conceptuelle	21
I	Introduction	21
II	Architecture générale de l'application	21
II.1	Architecture logique	21
1	Pattern de conception "MVC"	21
II.2	Architecture physique	22
1	Diagramme de déploiement	22
III	Conception détaillée	24
III.1	Diagramme des Tables de Base de Données	24
III.2	Spécification et raffinement des cas d'utilisation pour un "User"	28
1	Créer un compte pour s'inscrire	28
2	Mise à jour et Modifier Profil	31
3	Consulter des notifications	34
4	Échanger les messages	36
III.3	Spécification et raffinement des cas d'utilisation pour un "Influencer"	38
1	Consulter les offres de collaborations	38
III.4	Spécification et raffinement des cas d'utilisation pour un "Company"	40
1	Rechercher les influenceurs	40
2	Envoyer les offres de collaborations	41

III.5	Diagrammes de séquences	43
1	Créer un compte pour s'inscrire	43
2	Authentification	45
3	Mise à jour et Modifier Profil	46
4	Recevoir des Notifications	48
5	Échanger les messages	49
6	Consulter les offres de collaborations	51
7	Rechercher les Influenceurs	53
8	Envoyer les Offres de Collaborations	54
IV	Conclusion	54
4	Réalisation	55
I	Introduction	55
II	Environnement de développement	55
II.1	Technologies utilisées :	55
1	NodeJs :	55
2	ExpressJs :	56
3	MongoDB	56
4	Firebase Cloud Messaging	56
5	Socket.io	57
6	Flutter	57
7	GitLab	58
8	Draw.io	58
9	dbdiagram.io	58
10	Figma	59
III	Interfaces d'applications	59
IV	Conclusion	81
	Conclusion générale	82
	Bibliographie	84
	Résumé	85

Table des figures

1.1	Logo de Tekboot Solutions	3
1.2	Page d'accueil du site web <i>EmploiTunisie.com</i>	7
1.3	Page d'accueil du site web <i>TanitJobs</i>	8
1.4	Page d'accueil du site web <i>KeeJob</i>	8
1.5	La méthodologie agile « SCRUM »	11
1.6	Logo Java	12
1.7	Logo Spring Boot	12
1.8	Logo Kafka	13
1.9	Logo React	13
1.10	Logo GitHub	13
1.11	Logo Jira	13
1.12	Logo IntelliJ IDEA	14
1.13	Logo VS Code	14
1.14	Logo Postman	14
1.15	Logo Keycloak	15
1.16	Logo MongoDB	15
1.17	Logo PostgreSQL	15
3.1	Schéma de l'architecture logicielle MVC	22
3.2	Diagramme de déploiement	23
3.3	Diagramme complet	25
3.4	Diagramme de cas d'utilisation "Créer un compte pour s'inscrire"	29
3.5	Diagramme de cas d'utilisation "Mise à jour et modifier profil"	32
3.6	Diagramme de cas d'utilisation "Consulter des notifications"	34
3.7	Diagramme de cas d'utilisation "Consulter les offres de collaborations"	38
3.8	Diagramme de cas d'utilisation "Rechercher les influenceurs"	40
3.9	Diagramme de cas d'utilisation "Envoyer les offres de collaborations"	42
3.10	Diagramme de séquence "S'inscrire"	44
3.11	Diagramme de séquence "Authentification"	45
3.12	Diagramme de séquence "Mise à jour et Modifier Profil"	47
3.13	Diagramme de séquence "Consulter les notifications"	48
3.14	Diagramme de séquence "Échanger les Messages"	50
3.15	Diagramme de séquence "Consulter les offres de collaborations"	52

3.16	Diagramme de séquence "Rechercher les Influenceurs"	53
3.17	Diagramme de séquence "Envoyer les Offres de Collaborations"	54
4.1	Login	60
4.2	Register	61
4.3	Register as project owner	62
4.4	Register as an influencer	63
4.5	Forgot password (Part 1)	64
4.6	Forgot password (Part 2)	65
4.7	Menu	66
4.8	Influencer Profile	67
4.9	Collaboration request	68
4.10	Company Profile	69
4.11	Edit Company Profile	70
4.12	Board	71
4.13	Collaboration details	72
4.14	Accept request	73
4.15	Reject request	74
4.16	Terminate request	75
4.17	Notification page	76
4.18	Convo-box page	77
4.19	Chat screen	78
4.20	Influencer Profile	79
4.21	Edit Profile	80
4.22	Notifications	81

Liste des tableaux

1.1	Fiche signalétique de l'entreprise Tekboot Solutions	4
1.2	Tableau comparatif des différents sites	9
2.1	Rôles et fonctions des utilisateurs dans TalentCloud	16
2.2	Backlog des User Stories par Sprint	20
3.1	Description textuelle du Diagramme des Tables de Base de Données	28
3.2	Description textuel du cas d'utilisation : Créer un compte pour s'inscrire	31
3.3	Description textuel du cas d'utilisation : Mise à jour et Modifier Profil	34
3.4	Description textuel du cas d'utilisation : Consulter les notifications	36
3.5	Description textuel du cas d'utilisation : Échanger les messages	36
3.5	Description textuel du cas d'utilisation : Échanger les messages	37
3.6	Description textuel du cas d'utilisation : Consulter les offres de collaborations	39
3.7	Description textuel du cas d'utilisation : Rechercher les influenceurs	41
3.8	Description textuel du cas d'utilisation : Envoyer les offres de collaborations	43

Introduction générale

Dans un contexte où la transformation numérique redéfinit les modes de gestion au sein des entreprises, la recherche de solutions technologiques efficaces devient primordiale, notamment dans le domaine des ressources humaines. Le recrutement, en particulier, nécessite des outils modernes capables de s'adapter aux nouveaux défis liés à la compétitivité, à l'analyse de données et à l'automatisation des processus.

C'est dans cette optique que nous avons développé **TalentCloud**, une plateforme innovante dédiée à l'optimisation du processus de recrutement. Elle repose sur une architecture avancée basée sur des microservices et des micro-frontends, offrant ainsi flexibilité, évolutivité et facilité de maintenance. La solution permet une gestion intelligente des candidatures, une analyse approfondie des performances et une prise de décision éclairée grâce à l'exploitation des données.

Le développement de TalentCloud s'appuie sur des méthodologies modernes garantissant un cycle de vie logiciel agile, des déploiements continus, ainsi qu'une haute disponibilité du système. Cette approche assure une expérience utilisateur fluide tout en répondant aux exigences techniques et fonctionnelles du secteur.

Ce rapport présente notre travail structuré autour de quatre chapitres :

- **Contexte général** : introduction au domaine du recrutement numérique, présentation du projet TalentCloud, problématique et méthodologie de développement adoptée.
- **Spécification des besoins** : identification des besoins fonctionnels (gestion des candidatures, tableau de bord analytique, alertes) et non fonctionnels (sécurité, performance, scalabilité), accompagnée de diagrammes d'analyse.
- **Conception** : description de l'architecture modulaire, des choix technologiques, et des modèles de données assurant la cohérence du système.
- **Réalisation** : présentation de l'environnement technique (frameworks, bases de données, services), illustrée par des interfaces clés de la plateforme.

Enfin, une conclusion générale mettra en lumière les résultats obtenus et proposera des perspectives d'évolution pour faire de TalentCloud un acteur majeur du recrutement intelligent.

Chapitre 1

Contexte général du projet

I Introduction

Ce chapitre constitue une étude préliminaire du projet. Il débute par une présentation de l'entreprise hôte, suivie d'une analyse de l'existant afin de mieux comprendre le contexte et les contraintes du projet. Ensuite, les différentes méthodologies seront explorées avant de justifier le choix de celle adoptée. Enfin, la phase de conception sera détaillée, mettant en avant l'architecture et les principales composantes de la solution proposée.

II Entreprise d'accueil

Dans cette partie, nous présentons la société d'accueil ainsi que ses domaines d'activité.

II.1 Présentation de l'entreprise

Tekboot Solutions est une entreprise technologique basée à Bruxelles, spécialisée dans le développement logiciel, les architectures microservices, le DevOps, et l'intégration de solutions cloud modernes. Créée en 2020, elle accompagne les entreprises dans leur transformation digitale en leur fournissant des solutions innovantes, scalables et sécurisées.

Tekboot agit à la fois comme startup studio (en développant ses propres produits) et ESN (Entreprise de Services du Numérique), en mettant à disposition des talents pour des missions techniques en Belgique et à l'international.

Sa mission est d'autonomiser les entreprises et les particuliers à réaliser leurs aspirations numériques et à prospérer dans un monde technologique en constante évolution.



FIGURE 1.1 – Logo de Tekboot Solutions

II.2 Spécialisations

- Front-end development (HTML/CSS/JavaScript).
- Back-end development (Node.js/Express.js).
- Full-stack development.
- Développement mobile (Android/iOS, Flutter).
- Développement de jeux (Unity).
- Développement d'applications web et e-commerce.
- Cloud computing (AWS, Google Cloud).
- DevOps et CI/CD.
- Gestion agile de projet et design UI/UX.

II.3 Les services de l'entreprise

- **Développement** : Conception et réalisation d'applications web et mobiles en adoptant des technologies modernes et évolutives.
- **Architecture logicielle** : Élaboration de systèmes robustes, modulaires et maintenables grâce à des architectures logicielles bien pensées.
- **Exploitation des données** : Création de tableaux de bord décisionnels et extraction d'indicateurs clés pour guider les décisions stratégiques.
- **Sécurité** : Mise en place de mécanismes de contrôle d'accès stricts et de pratiques assurant la protection des données.
- **Qualité logicielle** : Intégration de tests automatisés et d'analyses continues du code afin de garantir la fiabilité et la performance des solutions livrées.

Dénomination sociale	Tekboot Solutions
Forme juridique	Société privée
Date de création	2020
Secteur d'activité	Technologies de l'information et du numérique
Activité principale	Développement web, mobile et exploitation des données
Directeur général	Non spécifié
Adresse	Boulevard Bischoffsheim 39, 1000 Bruxelles, Belgique
Téléphone	Non spécifié
Site Web	https://tekboot.io
Effectif total	11-50 employés

Table 1.1 – Fiche signalétique de l'entreprise Tekboot Solutions

II.4 Fiche signalétique

III Fondements du projet

Le projet TalentCloud repose sur des principes fondamentaux visant à garantir une solution flexible, évolutive et sécurisée pour l'automatisation du processus de recrutement. L'architecture du système est basée sur une approche modulaire utilisant des microservices et des micro-frontends, permettant une séparation claire des responsabilités et une gestion simplifiée des différentes fonctionnalités. Chaque microservice est indépendant et peut être mis à l'échelle en fonction des besoins, assurant ainsi une grande résilience et une haute disponibilité du système. L'utilisation de technologies modernes telles que Spring Boot pour le backend, Angular pour le frontend et AWS pour l'infrastructure cloud permet de répondre aux exigences de performance et de sécurité. Ce choix d'architecture garantit non seulement la scalabilité du système, mais aussi une maintenance facilitée et une évolution continue adaptée aux besoins du marché du travail.

III.1 Microservices et React

L'architecture de l'application *TalentCloud* repose sur une approche modulaire utilisant les **microservices** et le framework **React** pour le développement de l'interface utilisateur.

Les microservices permettent de décomposer l'application *backend* en composants indépendants, chacun responsable d'un domaine fonctionnel spécifique. Cette architecture offre plusieurs avantages pour le développement logiciel :

- **Scalabilité indépendante** : chaque microservice peut être mis à l'échelle indépendamment selon les besoins.
- **Résilience accrue** : en cas de panne d'un service, cela n'affecte pas l'ensemble de l'application.
- **Facilité de maintenance et de mise à jour** : les microservices peuvent être mis à jour de manière indépendante, facilitant ainsi le déploiement de nouvelles fonctionnalités.

Pour la partie frontend, **React** est utilisé pour construire une interface utilisateur moderne et

modulaire. Grâce à sa structure basée sur les composants, React offre plusieurs avantages :

- **Flexibilité des équipes de développement** : les composants React permettent aux équipes de travailler indépendamment sur différentes parties de l'interface.
- **Réutilisation des composants** : les composants React peuvent être facilement réutilisés dans d'autres projets ou sections de l'application.
- **Expérience utilisateur cohérente** : React permet de maintenir une interface uniforme tout en conservant une base de code bien structurée et maintenable.

III.2 API Gateway

Une **API Gateway** est un composant essentiel dans une architecture de microservices qui sert de point d'entrée unique pour toutes les requêtes client. Elle joue le rôle d'intermédiaire entre les clients et les différents microservices, offrant une couche d'abstraction qui simplifie la complexité d'un système distribué.

a) Architecture fondamentale

L'architecture d'une API Gateway repose sur trois composants principaux :

- **Routes** : Une route est définie par un identifiant (ID), une destination (URI) et un ensemble de *prédicats* et *filtres*. Les routes déterminent comment les requêtes entrantes sont dirigées vers les services appropriés.
- **Prédicats** : Ce sont des fonctions de prédicat Java 8 utilisées pour faire correspondre les requêtes HTTP en utilisant les en-têtes, méthodes ou paramètres. Les prédicats permettent d'appliquer des conditions spécifiques pour déterminer si une route doit être utilisée.
- **Filtres** : Les filtres permettent de modifier les requêtes HTTP entrantes ou les réponses sortantes. Ils peuvent être appliqués globalement ou spécifiquement à certaines routes.

b) Implémentations courantes

Dans l'écosystème Spring, **Spring Cloud Gateway** est une solution moderne pour implémenter une API Gateway :

- **Spring Cloud Gateway** : Construit sur *Spring 6*, *Spring Boot 3* et *Project Reactor*, il offre un moyen simple mais efficace de router les API et de fournir des préoccupations transversales telles que la sécurité, la surveillance/métriques et la résilience. Il s'agit d'une passerelle non-bloquante et réactive, adaptée aux systèmes à haute performance.
- **Configuration** : La configuration peut se faire par programmation (en utilisant Java) ou via des fichiers de propriétés **YAML/properties**, ce qui offre une grande flexibilité dans la définition des routes, prédicats et filtres.
- **Intégration** : L'API Gateway s'intègre facilement avec d'autres composants de l'écosystème des microservices, comme les services de découverte (Eureka, Consul), les équilibrateurs de charge et les outils de surveillance.

c) Avantages stratégiques

L'API Gateway apporte plusieurs avantages stratégiques aux architectures de microservices :

- **Abstraction de l'implémentation** : Elle permet de modifier l'implémentation du service *backend* sans affecter l'API publique, offrant ainsi une grande flexibilité dans l'évolution de l'architecture.
- **Simplification pour les développeurs** : Les développeurs d'applications clients n'ont besoin de connaître qu'un seul point d'entrée, au lieu de multiples services avec des interfaces différentes.
- **Gestion des versions** : L'API Gateway facilite la gestion des versions des services et permet des stratégies de déploiement progressif comme les versions *Canary*.
- **Réduction de la complexité** : Sans API Gateway, on se retrouverait rapidement avec un maillage de points d'interaction, augmentant la complexité et le couplage dans le système.

IV Cadre du projet

IV.1 Contexte Général et problématique

1 Présentation du projet

Le projet **TalentCloud** consiste à développer et déployer une plateforme moderne dédiée à la gestion de carrière et au recrutement. Cette application repose sur une architecture innovante combinant microservices et microfrontends, assurant ainsi scalabilité, modularité et sécurité. L'objectif est de répondre aux besoins évolutifs des entreprises en matière de gestion des talents, en facilitant le suivi des candidatures et en optimisant le processus de recrutement grâce à une interface utilisateur intuitive et des technologies avancées.

2 Problématique

Les solutions traditionnelles de recrutement manquent d'automatisation et présentent de nombreuses lacunes, notamment en matière de validation des données, de sécurité et de performance. Elles peinent à s'adapter à la croissance rapide des besoins des entreprises modernes. Cette absence de fiabilité et de flexibilité impacte négativement l'efficacité des processus RH. Il est donc impératif d'introduire une solution technologique innovante capable de pallier ces limites, tout en assurant une expérience utilisateur optimale.

IV.2 Étude de l'existant

L'étude de l'existant est une phase primordiale pour la réalisation d'un projet. Il s'agit en effet d'observer et d'analyser les solutions existantes puis d'identifier leurs points faibles et leurs points forts afin d'en tenir compte lors de la mise en place de l'application concernée. Il se compose de trois parties : description de l'existant, critique de l'existant et proposition d'une solution.

1 EmploiTunisie.com

EmploiTunisie.com est l'un des principaux sites de recrutement en Tunisie. Il propose des offres d'emploi dans divers secteurs et niveaux d'expérience, ainsi que des offres de stages pour les étudiants. Les entreprises peuvent également publier des annonces d'emploi et rechercher des CV en ligne. Le site est convivial et facile à naviguer. La Figure 1.2 présente la page d'accueil du site web *EmploiTunisie* [emploi tunisie].

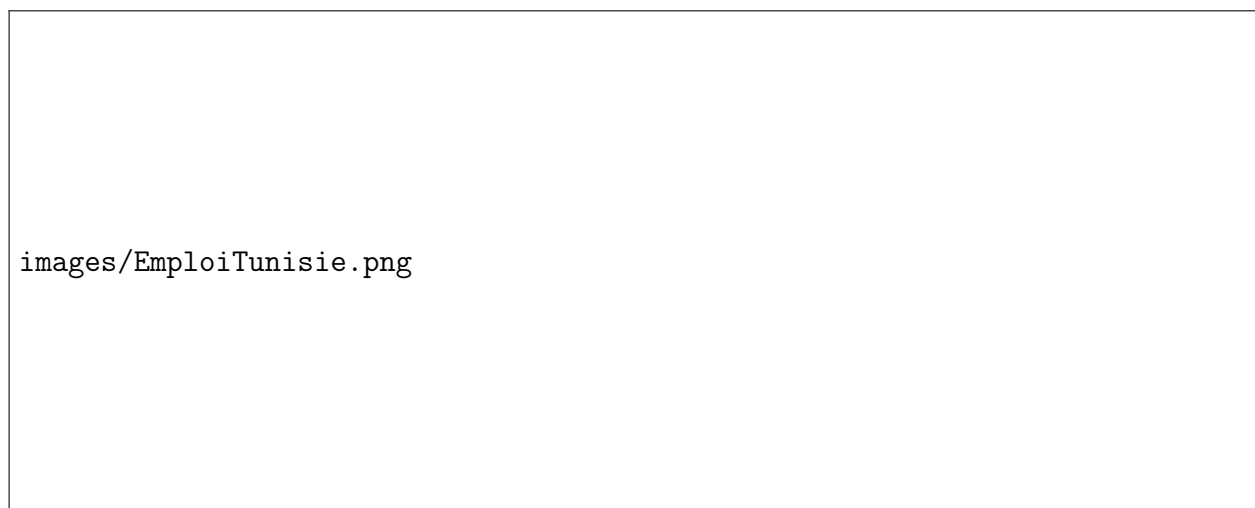


FIGURE 1.2 – Page d'accueil du site web *EmploiTunisie.com*

Avantages

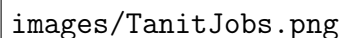
- Un large éventail d'annonces d'offres de stages est disponible.
- Un système de filtrage avancé pour affiner les résultats de recherche.
- Une option de dépôt de CV en ligne pour les candidats.

Inconvénients

- Les fonctionnalités sont limitées pour les recruteurs.
- Les annonces de stages ne sont pas toujours vérifiées.
- Un manque de transparence concernant les informations sur les étudiants et les recruteurs.

2 TanitJobs.com

TanitJobs.com C'est également un site de recrutement en Tunisie bien connu, proposant des offres de stages dans divers secteurs et régions. Le site offre également des services tels que la publication de CV et des alertes e-mail pour les offres de stages pertinentes. Tanit-jobs.com est réputé pour son interface utilisateur intuitive et sa grande base de données de CV. La Figure 1.3 présente la page d'accueil du site web *TanitJobs* [TanitJobs].



images/TanitJobs.png

FIGURE 1.3 – Page d’accueil du site web *TanitJobs*

Avantages

- Une interface utilisateur simple et facile à utiliser.
- Un large éventail d’annonces d’offres de stages est disponible.
- Une bonne réputation et notoriété sur le marché tunisien.

Inconvénients

- Le manque de transparence concernant les informations sur les recruteurs.
- Le coût relativement élevé pour les recruteurs pour poster des annonces d’emploi.
- Le manque d’interactivité et d’engagement avec les utilisateurs.

3 KeeJob.com

KeeJob.com C’est une autre plateforme de recrutement en ligne populaire en Tunisie. Le site propose des offres de stages dans divers secteurs et niveaux d’expérience, ainsi que des fonctionnalités pour la publication de CV et la recherche d’emploi. Keejob.com se distingue également par son interface utilisateur conviviale et sa compatibilité mobile. La Figure 1.4 présente la page d’accueil du site web *KeeJob* [**KeeJob.com**].



images/KeeJob.png

FIGURE 1.4 – Page d’accueil du site web *KeeJob*

Avantages

- Un système de filtrage avancé pour affiner les résultats de recherche.
- Des annonces d'emploi vérifiées.
- Une option de CV en ligne pour les candidats.

Inconvénients

- Une interface utilisateur un peu désuète.
- Un manque d'interactivité et d'engagement avec les utilisateurs.
- sUn coût élevé pour les recruteurs pour poster des annonces d'emploi.

IV.3 Critique de l'existant

Dans cette partie, nous allons présenter sous forme d'un tableau comparatif une synthèse résultante des solutions existantes. Cette synthèse est faite selon des critères d'évaluation de ces solutions.

Critères	EmploiTunisie	KeeJob	TanitJobs
C1 - Passage de test technique en ligne sur la plateforme	X	X	X
C2 - Possibilité de recevoir instantanément les résultats des tests techniques et savoir si vous avez réussi ou non	X	X	X
C3 - Possibilité de réserver un rendez-vous pour un entretien avec les ressources humaines	X	X	X

Table 1.2 – Tableau comparatif des différents sites

Après cette comparaison, nous avons relevé plusieurs insuffisances dans les solutions actuelles. Ces lacunes sont particulièrement visibles en ce qui concerne la possibilité de passer le test technique en ligne sur la plateforme et, en cas de réussite, la possibilité de choisir une date pour l'entretien RH.

La plupart des applications mentionnées se concentrent uniquement sur l'objectif principal, négligeant les détails qui pourraient faire la différence, notamment en ignorant la composante cruciale de l'entretien technique en ligne. Remédier à ces aspects pourrait contribuer à fidéliser les utilisateurs et à accroître la fréquentation de la plateforme.

IV.4 Solution Proposée

La solution proposée repose sur une architecture *cloud-native*, utilisant des microservices et des microfrontends pour garantir l'évolutivité et la flexibilité de l'application. Cette architecture inclut les éléments suivants :

- **Microservices** : chaque service est indépendant et responsable d'une fonction spécifique, permettant de garantir la scalabilité et la résilience de l'application.

- **Microfrontends** : l'interface utilisateur est divisée en composants autonomes, chacun pouvant être développé et mis à jour indépendamment pour assurer une agilité accrue.
- **Technologies modernes** : utilisation de technologies comme *React*, *Node.js* et *Kubernetes* pour déployer et gérer les microservices et les microfrontends.
- **Base de données** : intégration d'une base de données *PostgreSQL/MongoDB* pour assurer une gestion robuste des données de recrutement et des profils candidats.
- **Sécurité** : sécurisation des échanges de données par des mécanismes de chiffrement et de gestion des accès.

Cette architecture garantit que l'application **TalentCloud** soit prête pour une mise en production avec une haute disponibilité, des performances optimisées et une gestion simplifiée des fonctionnalités.

IV.5 Etude de la methodologie de développement

La méthodologie de travail est un élément essentiel à la réussite de tout projet. Il existe plusieurs approches, chacune étant adaptée à des contextes et besoins spécifiques. Parmi les plus connues, on peut citer le **modèle en cascade**, qui est particulièrement efficace lorsque les besoins du projet sont clairement définis dès le départ. On trouve également la **méthodologie Scrum**, qui propose une approche itérative et flexible, favorisant l'adaptation continue aux évolutions du projet. Enfin, la **méthodologie 2TUP** répond aux contraintes de changement continu imposées aux systèmes d'information des entreprises, en proposant une démarche centrée sur l'adaptabilité.

1 Choix de la methodologie de développement

Après avoir étudié les différentes méthodologies de développement logiciel, nous avons opté pour Scrum en tant que méthodologie de développement agile pour la conception et le développement de notre plateforme.

2 Avantages de Scrum

La méthodologie **Scrum** s'adapte parfaitement à la décomposition de notre projet. Elle repose sur plusieurs avantages clés qui en font une approche particulièrement efficace pour le développement logiciel :

- **Flexibilité** : Scrum peut être utilisée pour différents types de projets, quelle que soit leur nature ou leur complexité.
- **Collaboration et communication** : Elle favorise un fort niveau d'interaction entre les membres de l'équipe, ce qui contribue à une meilleure qualité du travail fourni.
- **Cycles courts et itératifs** : Grâce à des sprints de courte durée, Scrum permet de livrer rapidement des fonctionnalités, tout en intégrant régulièrement les retours des utilisateurs pour améliorer le produit.
- **Satisfaction du client** : La transparence du processus et sa capacité à s'adapter aux besoins permettent de mieux répondre aux attentes du client.

- **Adaptabilité aux changements** : Scrum offre une grande réactivité face aux imprévus ou aux évolutions des exigences, ce qui est essentiel dans les projets dynamiques.

3 Mise en pratique de la méthodologie SCRUM

Dans une approche Scrum, le projet est structuré autour de cycles de développement appelés **sprints**, dont la durée varie généralement entre deux et quatre semaines. Chaque sprint vise à livrer un incrément fonctionnel du produit.

Le projet commence par un **sprint 0**, qui constitue une phase préparatoire essentielle. Ce sprint est consacré à la mise en place de l'architecture du projet, à la configuration des environnements de développement, ainsi qu'à l'installation des outils nécessaires pour le suivi des tâches et l'intégration continue.

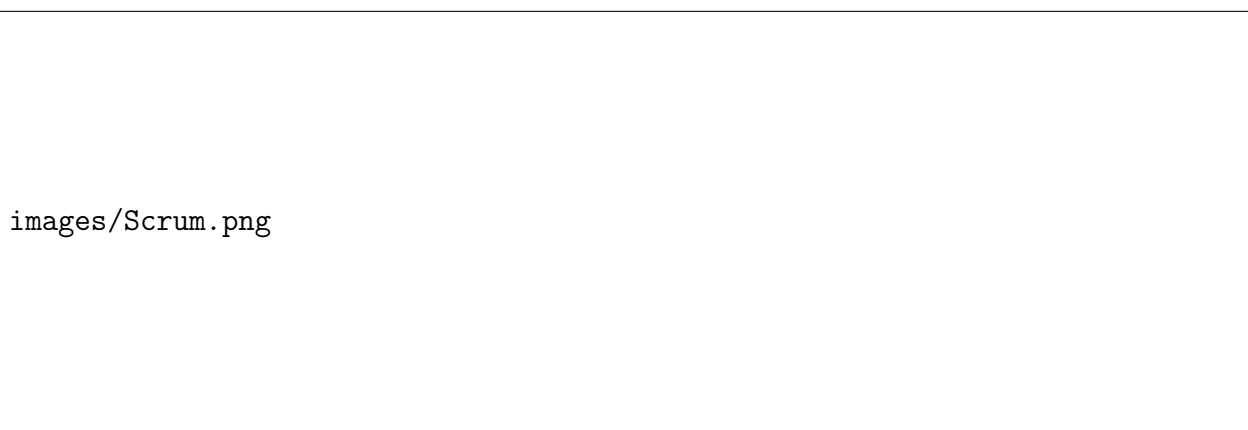


FIGURE 1.5 – La méthodologie agile « SCRUM »

Comme illustré dans la figure 1.5, la méthodologie **Scrum** repose sur trois éléments fondamentaux : les **sprints et leurs événements**, les **trois rôles principaux** (Product Owner, Équipe de Développement, Scrum Master), et le **backlog produit**.

- **Le Product Owner** : il représente les clients et utilisateurs du produit. Il est chargé de gérer le backlog produit, de prioriser les fonctionnalités et de s'assurer que les besoins métier sont bien pris en compte.
- **L'équipe de développement** : elle transforme les besoins exprimés par le Product Owner en fonctionnalités concrètes. Composée de développeurs, testeurs, designers et autres contributeurs, elle est responsable de livrer, à la fin de chaque sprint, les éléments définis comme prioritaires.
- **Le Scrum Master** : garant du bon déroulement de la méthodologie Scrum, il joue un rôle de coach auprès du Product Owner et de l'équipe de développement. Il veille à la bonne application des principes Scrum et à la résolution des obstacles.
- **Le Sprint** : chaque itération (appelée sprint) a une durée fixe, généralement entre deux et quatre semaines, et reste constante tout au long du projet. Cette régularité permet de maintenir un rythme soutenu et de produire des indicateurs fiables. Le projet débute souvent par un **sprint 0**, consacré aux préparatifs : mise en place de l'architecture, des environnements et des outils.
- **Le Product Backlog** : il s'agit d'une liste priorisée des fonctionnalités et exigences du produit. Ce backlog est évolutif et géré par le Product Owner, qui y classe les éléments selon leur valeur

métier, le niveau de risque, l'effort requis, et la maturité technique ou fonctionnelle.

- **La Sprint Review** : réunion organisée à la fin de chaque sprint, elle permet à l'équipe Scrum et aux parties prenantes de passer en revue l'incrément de produit livré. Sa durée maximale est généralement de quatre heures.
- **La Sprint Retrospective** : cette réunion interne à l'équipe Scrum vise à analyser le sprint écoulé pour en tirer des enseignements. Elle permet d'identifier ce qui a bien fonctionné, ce qui peut être amélioré, et d'adapter les processus de développement en conséquence.

IV.6 Technologies utilisées

Dans cette partie, nous présentons les différentes technologies utilisées dans le cadre du projet TalentCloud, notamment les langages de programmation, frameworks, plateformes de déploiement et outils de travail.

1 Langages et Frameworks de programmation

Java : Langage de programmation orienté objet et plateforme informatique, conçu pour être portable, sécurisé et performant. Java est utilisé pour le développement d'applications multiplateformes via la machine virtuelle Java (JVM).



FIGURE 1.6 – Logo Java

Spring Boot : Framework open-source pour Java, basé sur le projet Spring, qui simplifie le développement d'applications autonomes, prêtes à produire, en fournissant des configurations par défaut et des outils intégrés.

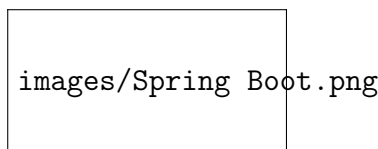


FIGURE 1.7 – Logo Spring Boot

Kafka : Apache Kafka est une plateforme open-source de streaming d'événements distribuée, utilisée par des milliers d'entreprises pour des pipelines de données haute performance, des analyses en continu, de l'intégration de données et des applications critiques.

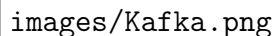
images/Kafka.png

FIGURE 1.8 – Logo Kafka

React : Apache Kafka est une plateforme open-source de streaming d'événements distribuée, utilisée par des milliers d'entreprises pour des pipelines de données haute performance, des analyses en continu, de l'intégration de données et des applications critiques.

images/React.png

FIGURE 1.9 – Logo React

2 Plateformes de développement utilisées

GitHub : Plateforme de développement collaboratif basée sur Git, permettant de gérer et d'héberger des projets de code source, avec des fonctionnalités comme les pull requests et les issues pour collaborer facilement.

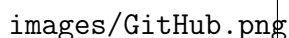
images/GitHub.png

FIGURE 1.10 – Logo GitHub

Jira : Outil de gestion de projet Agile et de suivi des tâches, utilisé pour la planification, le suivi des sprints et des backlog dans des projets de développement logiciel.

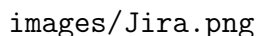
images/Jira.png

FIGURE 1.11 – Logo Jira

IntelliJ IDEA : Environnement de développement intégré (IDE) principalement destiné au développement Java, offrant une gamme d'outils pour l'analyse de code, la gestion de projet et le refactoring.

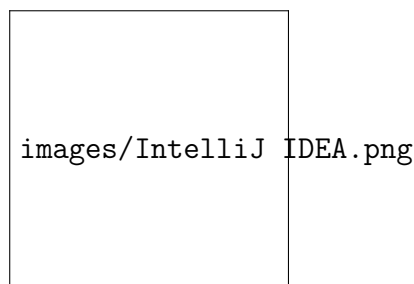


FIGURE 1.12 – Logo IntelliJ IDEA

Visual Studio Code (VS Code) : Éditeur de code open-source léger, développé par Microsoft, offrant des fonctionnalités avancées d'édition, de débogage et d'intégration avec Git.

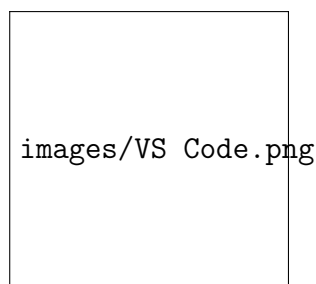


FIGURE 1.13 – Logo VS Code

Postman : Plateforme collaborative pour le développement et le test d'APIs, qui permet de créer des requêtes HTTP, de les tester, et de documenter des APIs RESTful.

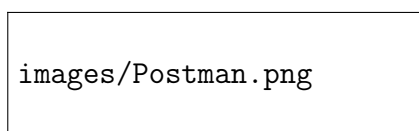


FIGURE 1.14 – Logo Postman

3 Autres Technologies

Keycloak : Système de gestion d'identités et d'accès, utilisé pour l'authentification et l'autorisation des utilisateurs. Keycloak est intégré dans l'architecture pour gérer l'authentification des utilisateurs via des protocoles comme OAuth2 et JWT, et pour contrôler l'accès basé sur les rôles (RBAC).



FIGURE 1.15 – Logo Keycloak

MongoDB : Base de données NoSQL orientée documents, offrant une grande flexibilité pour le stockage et la manipulation de données structurées et semi-structurées.

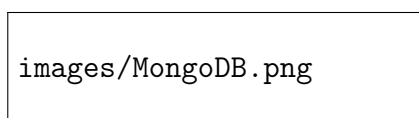


FIGURE 1.16 – Logo MongoDB

PostgreSQL : Système de gestion de base de données relationnelle objet (SGBDRO) open-source, connu pour sa fiabilité, sa robustesse et sa conformité aux normes SQL.



FIGURE 1.17 – Logo PostgreSQL

V Conclusion

Ce chapitre a posé les bases du projet en précisant son contexte, ses objectifs, et la méthodologie appliquée pour garantir un développement fluide et structuré de l'application TalentCloud. Le chapitre suivant abordera les spécifications techniques et les choix technologiques retenus pour la mise en œuvre de la solution.

Chapitre 2

Analyse et spécification des besoins

I Introduction

Ce chapitre se concentre d'abord sur la spécification des besoins fonctionnels et non-fonctionnels de l'application, puis le pilotage du projet avec Scrum en définissant le backlog du produit et en planifiant les sprints

II Identification des acteurs

Rôle	Fonctions
Administrateur (Admin)	Après la connexion, il peut gérer les utilisateurs, valider les projets, gérer les abonnements et superviser les activités des recruteurs.
Candidat (Candidate)	Après la connexion, il peut créer et gérer son profil, consulter les offres d'emploi et postuler, passer des tests techniques et choisir une date pour un entretien.
Recruteur (Recruiter)	Après la connexion, il peut gérer les offres d'emploi, consulter les candidatures, planifier et organiser les entretiens, et communiquer avec les candidats.
Système IA (AI System)	Le système IA fournit des recommandations de matching, analyse les CV, attribue des scores de compatibilité, et améliore le processus de recrutement en apprenant des interactions passées.

Table 2.1 – Rôles et fonctions des utilisateurs dans TalentCloud

III Identification des besoins

III.1 Besoins Fonctionnels

Les besoins fonctionnels décrivent les fonctionnalités attendues de la plateforme.

Les besoins fonctionnels d'un système décrivent les fonctionnalités clés de celui-ci. Ils permettent de spécifier les actions attendues et les résultats nécessaires pour satisfaire les attentes des utilisateurs. Ces besoins seront présentés en fonction de chaque acteur interagissant avec l'application.

III.2 Administrateur

- Ajouter, modifier et supprimer des utilisateurs (candidats, recruteurs, etc.).
- Ajouter et gérer les entreprises partenaires.
- Superviser la création, la publication et la gestion des offres d'emploi.
- Valider ou rejeter les profils des candidats et gérer l'historique des candidatures et des documents associés.
- Assurer la conformité RGPD en gérant les demandes de suppression de données.
- Accéder à des rapports de performance et à des statistiques sur les KPIs et tendances du recrutement.
- Configurer les paramètres de sécurité, y compris l'authentification à deux facteurs (2FA).

III.3 Candidat

- Créer et gérer leur profil en ajoutant des informations personnelles, compétences, expériences et documents comme le CV.
- Rechercher des offres d'emploi et postuler directement via la plateforme.
- Suivre le statut de leurs candidatures en temps réel.
- Recevoir des invitations pour des entretiens et les reprogrammer si nécessaire.
- Participer aux entretiens en ligne, recevoir des notifications et échanger des messages avec les recruteurs.

III.4 Recruteur

- Créer, publier et gérer des offres d'emploi.
- Examiner les candidatures, analyser les profils des candidats et effectuer un classement.
- Organiser des entretiens, envoyer des invitations et rappels automatiques.
- Suivre les performances des entretiens et évaluer les candidats via des scorecards automatisées.
- Contacter les candidats, négocier les offres et gérer les retours d'expérience.

III.5 Système IA

- Utiliser des algorithmes de matching pour associer les compétences des candidats aux exigences des offres d'emploi.
- Extraire automatiquement des informations clés des CV (compétences, expériences, formations) et suggérer des améliorations.
- Attribuer un score de compatibilité entre le candidat et l'offre en fonction des analyses.
- Proposer des candidats aux recruteurs en fonction de la pertinence des profils.

- Analyser les entretiens vidéo et générer des scorecards automatiques pour évaluer les candidats.
- Optimiser le processus de recrutement en apprenant des interactions passées pour améliorer l'efficacité.

IV Besoins non fonctionnels

Les besoins non fonctionnels sont essentiels pour garantir la qualité et la fiabilité de la plateforme. Bien qu'ils ne concernent pas directement les fonctionnalités, ces critères sont cruciaux pour offrir une expérience optimale aux utilisateurs et assurer le succès du projet. Ils incluent des aspects comme la performance, la sécurité, l'esthétique et la compatibilité.

IV.1 Convivialité

- L'interface doit être simple, agréable et intuitive pour que les utilisateurs puissent naviguer facilement et trouver rapidement ce dont ils ont besoin.
- Une interface moderne et facile à utiliser, développée avec **React**.
- La plateforme prend en charge plusieurs langues pour que chacun puisse l'utiliser dans sa langue préférée.
- Notifications personnalisables pour informer les utilisateurs de manière adaptée à leurs besoins.
- Fonctionnalité de recherche et filtrage avancés, pour faciliter l'accès aux informations importantes.

IV.2 Performance

- La plateforme doit être rapide, réactive et capable de supporter de nombreux utilisateurs simultanément, sans compromettre la fluidité de l'expérience.
- Adaptée aux ESN de toutes tailles (de 10 à 200 collaborateurs).
- Architecture modulaire, permettant de déployer les fonctionnalités nécessaires selon les besoins.
- Traitement en temps réel des communications, pour des échanges instantanés.
- Analyse en temps réel des vidéos d'entretien pour un retour rapide.
- Notifications push instantanées, pour que l'utilisateur soit toujours informé.

IV.3 Sécurité

- Authentification à deux facteurs (2FA) pour renforcer la sécurité des comptes.
- Vérification en deux étapes pour les actions administratives sensibles.
- Chiffrement des données pour protéger les informations échangées.
- Conformité RGPD : gestion transparente des données et possibilité de suppression sur demande des utilisateurs.

IV.4 Compatibilité de l'application

- L'application doit être accessible sur différents appareils, navigateurs et systèmes d'exploitation, garantissant ainsi une expérience fluide quel que soit l'environnement de l'utilisateur.
- Intégration avec des calendriers externes, pour une meilleure synchronisation des événements.
- APIs RESTful pour une gestion fluide des profils et des documents.
- Communication via **Kafka**, assurant une mise à jour en temps réel des informations.

IV.5 Stockage et gestion des données

- Utilisation de **MongoDB** pour stocker les profils des utilisateurs et les historiques de communication.
- **AWS S3** pour un stockage sécurisé des documents des candidats.
- **MySQL** pour gérer les plannings et obtenir des statistiques agrégées sur les activités de la plateforme.

V Elaboration du backlog produit

Sprint	Fonctionnalité	User Story	Priorité
Sprint 1	Authentification (1.1)	En tant qu'administrateur, je veux m'authentifier pour accéder au système.	Élevée
Sprint 1	Gestion des utilisateurs (1.3)	En tant qu'administrateur, je veux ajouter des utilisateurs avec des rôles spécifiques.	Moyenne
Sprint 1	Blocage des utilisateurs (1.4)	En tant qu'administrateur, je veux bloquer ou débloquer un utilisateur en fonction de son comportement.	Moyenne
Sprint 1	Visibilité des profils (1.5)	En tant qu'administrateur, je veux gérer la visibilité des profils utilisateurs.	Moyenne
Sprint 1	Téléchargement des documents (1.6)	En tant que candidat, je veux télécharger et vérifier mes documents.	Moyenne
Sprint 1	Validation des documents (1.7)	En tant qu'administrateur, je veux valider les documents téléchargés par les candidats.	Moyenne
Sprint 2	Gestion du profil (2.1)	En tant que candidat, je veux créer et modifier mon profil.	Élevée
Sprint 2	Documents expirés (2.2)	En tant que candidat, je veux voir mes documents expirés.	Moyenne
Sprint 2	Visibilité du profil (2.3)	En tant que candidat, je veux gérer la visibilité de mon profil.	Moyenne
Sprint 2	Auto-complétion du CV (2.4)	En tant que candidat, je veux que mon profil soit automatiquement complété avec les informations de mon CV.	Moyenne
Sprint 2	Préférences de notification (2.5)	En tant que candidat, je veux gérer mes préférences de notification.	Moyenne
Sprint 2	Rappel d'entretien (3.2)	En tant que candidat, je veux recevoir un rappel d'entretien.	Moyenne

Table 2.2 – Backlog des User Stories par Sprint

VI Conclusion

Ce chapitre a défini les besoins fonctionnels et non fonctionnels de la plateforme, assurant une compréhension claire des attentes des utilisateurs et des exigences techniques. Le diagramme de cas d'utilisation offre une vue d'ensemble des interactions, facilitant la conception et le développement de l'application.

Chapitre 3

Etude conceptuelle

I Introduction

Dans ce chapitre, nous présentons les différents diagrammes UML qui illustrent le comportement dynamique et l'architecture du système proposé. Ces représentations visuelles permettent de mieux comprendre les interactions entre les acteurs et le système (diagrammes de séquence), ainsi que l'architecture physique de l'application (diagramme de déploiement) et les tables de base de données.

II Architecture générale de l'application

L'architecture logicielle est la structure et l'interaction des différents éléments d'une application. En fait, cette étape est nécessaire pour assurer un développement logiciel efficace et évolutif.

II.1 Architecture logique

1 Pattern de conception "MVC"

Dans notre plateforme, nous avons décidé d'utiliser le modèle MVC (Modèle-Vue-Contrôleur) pour organiser notre application. Le modèle MVC divise l'application en trois composants principaux : le modèle, la vue et le contrôleur. Le modèle gère les données et la logique métier, la vue s'occupe de l'interface utilisateur et le contrôleur agit en tant qu'intermédiaire entre les deux. Cette architecture nous permet de séparer clairement les responsabilités et de favoriser la modularité et la réutilisabilité du code [1].

La figure 3.1 présente un schéma descriptif qui permet de mieux comprendre le modèle MVC :

images/MVC_3d1d885667.jpg

FIGURE 3.1 – Schéma de l'architecture logicielle MVC

- **Modèle** : Le modèle représente la couche de données et la logique métier de l'application. Il encapsule les opérations de manipulation des données, les règles de validation et les algorithmes métier. Le modèle peut être constitué d'une base de données, d'un système de fichiers, d'API ou de toute autre source de données. Il est conçu pour être indépendant de la présentation ou de l'interaction avec l'utilisateur.

- **Vue** : La vue est responsable de l'interface utilisateur et de l'affichage des données. Elle présente les informations au format approprié, que ce soit une page web, une interface graphique ou un affichage mobile. La vue récupère les données nécessaires du modèle et les présente de manière structurée et conviviale. Elle permet également à l'utilisateur d'interagir avec l'application, en transmettant les actions de l'utilisateur au contrôleur.

- **Contrôleur** : Le contrôleur agit comme un médiateur entre le modèle et la vue. Il reçoit les demandes et les actions de l'utilisateur à travers la vue, puis interagit avec le modèle en conséquence. Il coordonne les interactions entre les différentes parties du système, en appliquant la logique métier appropriée. Le contrôleur met à jour le modèle en fonction des actions de l'utilisateur, puis informe la vue des changements nécessaires pour refléter les mises à jour effectuées.

II.2 Architecture physique

1 Diagramme de déploiement


Un diagramme de déploiement est un type de diagramme UML (Unified Modeling Language) qui représente l'architecture physique d'un système, en montrant comment les logiciels sont déployés sur le

matériel et les connexions entre les nœuds (serveurs, clients, bases de données, etc.). Il met l'accent sur :

- Les nœuds physiques (ordinateurs, serveurs, appareils mobiles).
- Les artefacts logiciels (applications, bases de données, services) déployés sur ces nœuds.
- Les relations ou communications entre les nœuds (par exemple, via des réseaux ou des API).

Il est utile pour visualiser la topologie du système, identifier les points de défaillance potentiels et planifier la mise en production.

La figure 3.2 présente le diagramme de déploiement de l'application de gestion des collaborations entre Companies et Influenceurs. Le système s'articule autour d'un client mobile hébergeant l'application utilisateur, un serveur central exécutant l'API Backend avec Node.js et Socket.IO, une base de données MongoDB pour stocker les données des requêtes, notifications et chats, un serveur Firebase Realtime Database pour la synchronisation des chats en temps réel, et un service FCM pour les notifications push. Les communications entre ces nœuds utilisent HTTP pour les requêtes API, WebSocket pour les interactions en temps réel, Mongoose pour MongoDB, et le SDK Firebase pour la Realtime Database, assurant une architecture sécurisée et performante pour les collaborations.



images/diagramme de d  ploiement1.drawio.png

FIGURE 3.2 – Diagramme de déploiement

III Conception détaillée

Dans le but d'assurer une implémentation efficace et cohérente du système, la phase de conception est cruciale pour une compréhension approfondie de son fonctionnement. Cette étape sera réalisée en utilisant des outils tels que les diagrammes des Tables de Base de Données et les diagrammes de séquences. En somme, la phase de conception est un élément clé pour la réussite du projet.

III.1 Diagramme des Tables de Base de Données

La figure 3.3 présente le diagramme des tables de base de données de l'application de gestion des collaborations. Il illustre les entités principales telles que User, Influencer, Company, Request, Notification, et Chat, ainsi que leurs relations.

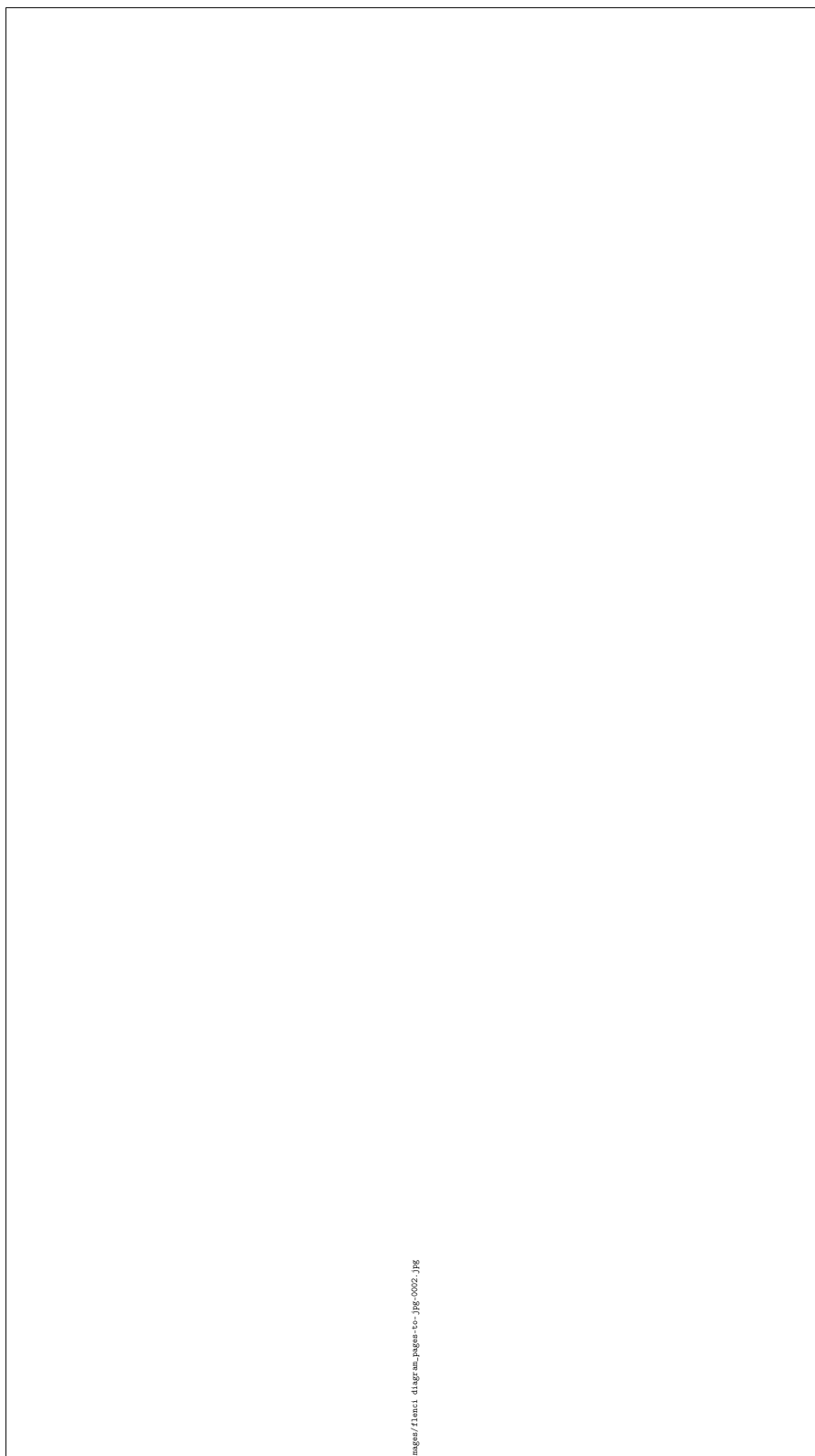


FIGURE 3.3 – Diagramme complet

Section	Contenu
Nom du Diagramme	Diagramme des Tables de Base de Données
Objectif	Représenter les entités principales et leurs relations dans la base de données de l'application de gestion des collaborations.
Entités Principales	<ul style="list-style-type: none">— User : id, username, email, password, profilePicture, role, expiration, followers.— Influencer : id, user_id, followers, publications, views, reach, impressions.— Company : id, user_id, fieldOfBusiness.— Request : id, company_id, influencer_id, offreTitle, estimatedBudget, duration, status, startDate, endDate, chatId.— Notification : id, recipient_id, sender_id, type, content, relatedRequest_id.— Chat : id, relatedRequest_id, participants, lastMessage, updatedAt.— Messages : id, chat_id, sender_id, text, imageData, timestamp.— MessageReactions : id, message_id, user_id, reactionType.— AcceptancesDetails : id, request_id, acceptedBy, dateTime.— TerminationDetails : id, request_id, terminatedBy, reason, date, wasEarly.— SocialMediaData : id, influencer_id, platform, profileLink, numberOfPublications, followers, views, reach, impressions, images.

Section	Contenu
Relations	<ul style="list-style-type: none"> — User-Influencer : 1 :1 (via user_id) — User-Company : 1 :1 (via user_id) — Request-Company : 1 :N (via company_id) — Request-Influencer : 1 :N (via influencer_id) — Notification-User (recipient) : 1 :N (via recipient_id) — Notification-User (sender) : 1 :N (via sender_id) — Notification-Request : 1 :N (via relatedRequest_id) — Chat-Request : 1 :1 (via chatId) — Chat-User (participants) : N :N (via participants list) — Messages-Chat : N :1 (via chat_id) — MessageReactions-Messages : 1 :N (via message_id) — AcceptancesDetails-Request : 1 :N (via request_id) — TerminationDetails-Request : 1 :N (via request_id) — SocialMediaData-Influencer : 1 :N (via influencer_id)
Clés Primaires	<ul style="list-style-type: none"> — User : id — Influencer : id — Company : id — Request : id — Notification : id — Chat : id — Messages : id — MessageReactions : id — AcceptancesDetails : id — TerminationDetails : id — SocialMediaData : id

Section	Contenu
Clés Étrangères	<ul style="list-style-type: none"> — Influencer : user_id (référence User.id) — Company : user_id (référence User.id) — Request : company_id (référence Company.id), influencer_id (référence Influencer.id), chatId (référence Chat.id) — Notification : recipient_id (référence User.id), sender_id (référence User.id), relatedRequest_id (référence Request.id) — Chat : relatedRequest_id (référence Request.id), participants (liste de User.id) — Messages : chat_id (référence Chat.id), sender_id (référence User.id) — MessageReactions : message_id (référence Messages.id), user_id (référence User.id) — AcceptancesDetails : request_id (référence Request.id) — TerminationDetails : request_id (référence Request.id) — SocialMediaData : influencer_id (référence Influencer.id)
Contraintes	<ul style="list-style-type: none"> — Les champs company_id et influencer_id dans Request doivent être non nuls. — Les statuts de Request sont limités à "pending", "going", "rejected", "done", "terminated". — Les champs recipient_id et sender_id dans Notification doivent exister dans User. — Le champ chat_id dans Messages doit exister dans Chat.
Notes	<ul style="list-style-type: none"> — Les données des chats sont synchronisées dans Firebase Realtime Database pour une mise à jour en temps réel, structure principale dans MongoDB. — Les relations N :N entre Chat et User (participants) sont gérées via une liste d'identifiants dans Chat.

Table 3.1 – Description textuelle du Diagramme des Tables de Base de Données

III.2 Spécification et raffinement des cas d'utilisation pour un "User"

1 Créer un compte pour s'inscrire

La figure 3.4 présente le raffinement du cas d'utilisation "Créer un compte pour s'inscrire".

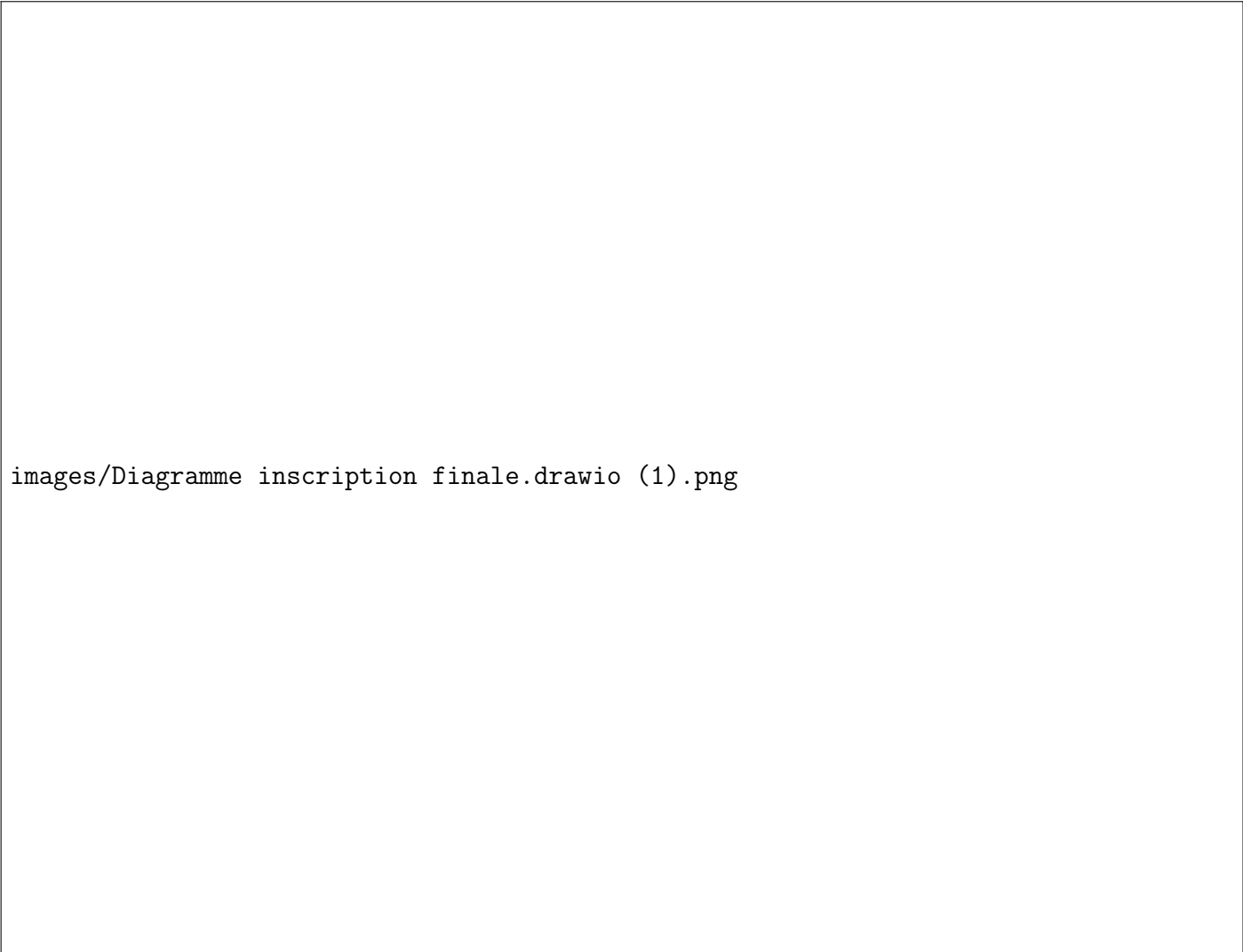


FIGURE 3.4 – Diagramme de cas d'utilisation "Créer un compte pour s'inscrire"

Section	Contenu
Nom du cas d'utilisation	Créer un compte pour s'inscrire
Acteur principal	Utilisateur (Company ou Influencer)
Acteurs secondaires	Système
Préconditions	L'utilisateur doit avoir accès à l'application Flenci et ne pas déjà posséder un compte. Sur la première page, l'utilisateur clique sur le bouton "Register" pour démarrer le processus d'inscription, puis voit les options "S'inscrire en tant que Company" et "S'inscrire en tant qu'Influencer".

Section	Contenu
Postconditions	Un nouveau compte est créé pour l'utilisateur avec un rôle défini (Company ou Influencer) et les informations requises enregistrées (username, email, password, profile picture, et données spécifiques selon le rôle).
Brève Description	Permet à un utilisateur de créer un compte sur Flenci en saisissant ses informations de base (username, email, password, profile picture) et en définissant obligatoirement son rôle (Company ou Influencer). Une Company doit saisir son domaine d'activité, tandis qu'un Influencer doit fournir ses intérêts, choisir une plateforme, et entrer des données sociales (lien, capture d'écran, suivis, publications, vues, reach), avec la possibilité d'ajouter d'autres plateformes.
Flux Principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la première page de l'application Flenci et clique sur le bouton "Register". 2. Le système affiche les options "S'inscrire en tant que Company" et "S'inscrire en tant qu'Influencer", et l'utilisateur sélectionne une option. 3. Le système inclut "Saisir username" : l'utilisateur entre un nom d'utilisateur unique. 4. Le système inclut "Saisir email" : l'utilisateur entre une adresse email valide. 5. Le système inclut "Saisir password" : l'utilisateur entre un mot de passe. 6. Le système inclut "Saisir profile picture" : l'utilisateur télécharge une photo de profil. 7. Selon le rôle choisi à l'étape 2 : <ul style="list-style-type: none"> - Le système inclut "S'inscrire en tant que Company" et demande "Saisir field of business" si l'utilisateur a choisi "Company".
	<ul style="list-style-type: none"> - Le système inclut "S'inscrire en tant que Influencer" et demande "Saisir ses intérêts", "Choisir Plateforme", "Saisir lien de plateforme", "Saisir une capture d'écran du profil", "Saisir nombre de suivis liés", "Saisir nombre de publications liées", "Saisir nombre de vues liées", et "Saisir reach liés" si l'utilisateur a choisi "Influencer". <ol style="list-style-type: none"> 8. Si l'utilisateur est un Influencer, il peut choisir d'exécuter "Ajouter Plateforme" (extension) pour ajouter une autre plateforme, répétant les étapes de saisie (lien, capture d'écran, suivis, publications, vues, reach). 9. Le système enregistre le compte avec les informations fournies.

Section	Contenu
	10. Le cas se termine avec succès.
Flux Alternatifs	<ul style="list-style-type: none"> - Rôle non sélectionné : À l'étape 2, si l'utilisateur ne choisit aucun rôle, le système affiche un message d'erreur "Rôle requis" et reste à l'étape 2. - Nom d'utilisateur déjà pris : À l'étape 3, si le nom d'utilisateur est déjà utilisé, le système affiche un message d'erreur "Nom d'utilisateur déjà pris" et retourne à l'étape 3. - Email invalide : À l'étape 4, si l'email est invalide ou déjà utilisé, le système affiche un message d'erreur "Email invalide ou déjà utilisé" et retourne à l'étape 4. - Mot de passe trop faible : À l'étape 5, si le mot de passe ne respecte pas les critères de sécurité, le système affiche un message d'erreur "Mot de passe trop faible" et retourne à l'étape 5. - Lien de plateforme invalide (Influencer) : À l'étape 7, si l'utilisateur est un Influencer et que le lien de la plateforme saisi est invalide, le système affiche un message d'erreur "Lien invalide" et demande une correction. - Capture d'écran non conforme (Influencer) : À l'étape 7, si l'utilisateur est un Influencer et que la capture d'écran ne contient pas son nom, le système affiche un message d'erreur "Capture d'écran non conforme" et demande une correction.
	- Ajout de plateforme annulé : À l'étape 8, si l'utilisateur choisit d'ajouter une plateforme mais annule, le système poursuit sans ajouter de plateforme.
Exigences Spéciales	Le système doit valider l'unicité du nom d'utilisateur et de l'email, ainsi que la force du mot de passe. Pour les Influenceurs, les liens des plateformes doivent être valides (ex. : URL fonctionnelle) et les captures d'écran doivent contenir le nom de l'Influencer pour confirmer l'identité.

Table 3.2 – Description textuel du cas d'utilisation : Créer un compte pour s'inscrire

2 Mise à jour et Modifier Profil

La figure 3.5 présente le raffinement du cas d'utilisation "Mise à jour et Modifier Profil".



FIGURE 3.5 – Diagramme de cas d'utilisation "Mise à jour et modifier profil"

Section	Contenu
Nom du cas d'utilisation	Mise à jour et Modifier Profil
Acteur principal	Utilisateur (Company ou Influencer)
Acteurs secondaires	Système
Préconditions	L'utilisateur doit être authentifié (relation «include» avec "Authentification").
Postconditions	Le profil de l'utilisateur est mis à jour avec les nouvelles informations fournies, selon son rôle (Company ou Influencer).

Section	Contenu
Brève Description	Ce cas d'utilisation permet à un utilisateur authentifié de mettre à jour et modifier son profil. Selon le rôle de l'utilisateur (Company ou Influencer), des fonctionnalités spécifiques sont disponibles pour modifier les informations pertinentes, comme changer la photo de profil, l'email ou le mot de passe. Si l'utilisateur est un Influencer, il peut également ajouter ou supprimer une plateforme, ainsi que modifier les données associées (publications, suivis, impressions, reach, views). Le système vérifie le mot de passe actuel avant d'effectuer les modifications.
Flux Principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à son profil et clique sur l'icône d'édition. 2. Le système présente les options de modification disponibles en fonction du rôle de l'utilisateur : <ul style="list-style-type: none"> - Si le rôle est "Company", le cas d'utilisation inclut "Mise à jour et Modifier Profil Company", permettant de modifier le "FieldOfBusiness". - Si le rôle est "Influencer", le cas d'utilisation inclut "Mise à jour et Modifier Profil Influencer", permettant de modifier les données sociales comme le nombre de publications, suivis, impressions, reach, et views, ainsi que d'ajouter ou de supprimer une plateforme. 3. L'utilisateur fournit les nouvelles données à modifier (par exemple, email, mot de passe, photo de profil). 4. Le système inclut la vérification du mot de passe actuel ("vérifie le mot de passe actuel").
	<ol style="list-style-type: none"> 5. Le système met à jour le profil avec les nouvelles données. 6. Le système enregistre les modifications et informe l'utilisateur de la réussite de la mise à jour. 7. Le cas d'utilisation se termine avec succès.
Flux Alternatifs	<ul style="list-style-type: none"> - Mot de passe incorrect : À l'étape 4, si le mot de passe actuel fourni est incorrect, le système informe l'utilisateur de l'erreur et le flux retourne à l'étape 3 pour une nouvelle tentative. - Rôle non reconnu : Si le rôle de l'utilisateur ne correspond ni à "Company" ni à "Influencer", le système affiche un message d'erreur et le cas d'utilisation se termine.
	<ul style="list-style-type: none"> - Ajouter une plateforme (Influencer) : À l'étape 2, si le rôle est "Influencer", l'utilisateur peut choisir d'ajouter une nouvelle plateforme ("Ajouter une nouvelle plateforme"). Il entre les données liées (nom de la plateforme, publications, suivis, impressions, reach, views). Le système enregistre la nouvelle plateforme et retourne à l'étape 3.

Section	Contenu
	<p>- Supprimer une plateforme (Influencer) : À l'étape 2, si le rôle est "Influencer", l'utilisateur peut choisir de supprimer une plateforme existante. Le système supprime la plateforme sélectionnée et retourne à l'étape 3.</p> <p>- Modifier les données sociales (Influencer) : À l'étape 2, si le rôle est "Influencer", l'utilisateur peut modifier les données sociales ("Modifier le nombre de publication / nombre de suivis / Impressions / Reach / Views"). Le système met à jour les données et retourne à l'étape 3.</p>
Exigences Spéciales	Aucune

Table 3.3 – Description textuel du cas d'utilisation : Mise à jour et Modifier Profil

3 Consulter des notifications

La figure 3.6 présente le raffinement du cas d'utilisation "Consulter les notifications".



FIGURE 3.6 – Diagramme de cas d'utilisation "Consulter des notifications"

Section	Contenu
Nom du cas d'utilisation	Consulter les notifications
Acteur principal	Utilisateur (Influencer ou Company)
Acteurs secondaires	Système
Préconditions	L'utilisateur doit être authentifié (relation «include» avec "Authentification") et avoir un token FCM enregistré pour recevoir des notifications.
Postconditions	L'utilisateur a consulté les notifications appropriées selon son rôle et l'état de la collaboration.
Brève Description	Permet à un utilisateur (Influencer ou Company) de consulter des notifications en temps réel concernant les collaborations. Selon le rôle : les Influenceurs reçoivent des notifications lorsqu'une collaboration leur est envoyée ou terminée, et les Companies reçoivent des notifications lorsqu'un Influencer accepte, refuse ou termine une collaboration, ou quand une collaboration est terminée.
Flux Principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page des notifications. 2. Le système détecte un événement lié à une collaboration (envoi, acceptation, refus, terminaison). 3. Selon le rôle de l'utilisateur : <ul style="list-style-type: none"> - Pour un Influencer : Le système affiche une notification pour une collaboration envoyée ("Collaborations envoyées") ou terminée ("Collaboration terminée"). - Pour une Company : Le système affiche une notification pour une acceptation ("Acceptation d'une collaboration envoyée"), un refus ("Refus d'une collaboration envoyée"), ou une terminaison ("Collaboration terminée") d'une collaboration. 4. Le système envoie la notification via FCM et l'affiche sur la page des notifications. 5. L'utilisateur consulte la notification. 6. Le cas se termine.
Flux Alternatifs	<ul style="list-style-type: none"> - Utilisateur hors ligne : À l'étape 5, si l'utilisateur est hors ligne, la notification est stockée par le système et envoyée dès que l'utilisateur se reconnecte. Une fois reconnecté, le flux reprend à l'étape 6. - Aucune notification disponible : À l'étape 4, si aucune notification n'est disponible, le système affiche un message "Aucune notification" et le cas se termine.

Section	Contenu
	- Rôle non reconnu : À l'étape 4, si le rôle de l'utilisateur ne correspond ni à "Influencer" ni à "Company", le système affiche un message d'erreur et le cas se termine.
Exigences Spéciales	Les notifications doivent être envoyées en temps réel via FCM et différenciées selon le rôle de l'utilisateur.

Table 3.4 – Description textuel du cas d'utilisation : Consulter les notifications

4 Échanger les messages

La figure 3.7 présente le raffinement du cas d'utilisation "Échanger les messages".

Table 3.5 – Description textuel du cas d'utilisation : Échanger les messages

Section	Contenu
Nom du cas d'utilisation	Échanger les messages
Acteur principal	Utilisateur (Influencer ou Company)
Acteurs secondaires	Système
Préconditions	L'utilisateur doit être authentifié (relation «include» avec "Authentification"). Une collaboration doit exister (par exemple, une offre acceptée via "Accepter les offres de collaborations") entre les parties pour activer le chat.
Postconditions	Les messages sont échangés entre l'Influencer et la Company dans le chat actif, et les actions supplémentaires (modification, suppression, fichiers, photos, appels) sont enregistrées si effectuées.
Brève Description	Permet à un utilisateur (Influencer ou Company) d'échanger des messages dans un chat lié à une collaboration existante. Selon le rôle, des fonctionnalités spécifiques sont disponibles. L'utilisateur peut optionnellement modifier ou supprimer des messages, échanger des fichiers ou photos, répondre à des messages, ou passer un appel.
Flux Principal	<ol style="list-style-type: none"> 1. L'utilisateur est authentifié (via "Authentification"). 2. L'utilisateur accède au chat lié à une collaboration existante. 3. Selon le rôle de l'utilisateur : <ul style="list-style-type: none"> - Si "Influencer", le système inclut "Échanger les messages Influencer". - Si "Company", le système inclut "Échanger les messages Company". 4. L'utilisateur envoie un message texte dans le chat.

Section	Contenu
	<p>5. Le système enregistre le message et le transmet au destinataire en temps réel.</p> <p>6. Le destinataire reçoit et peut visualiser le message.</p> <p>7. Le cas se termine.</p>
Flux Alternatifs	<ul style="list-style-type: none"> - Rôle non reconnu : À l'étape 3, si le rôle de l'utilisateur ne correspond ni à "Influencer" ni à "Company", le système affiche un message d'erreur et le cas se termine. - Modifier un message : À l'étape 4, l'utilisateur peut choisir de modifier un message envoyé ("Modifier des messages"). Le système met à jour le message et informe le destinataire. - Supprimer un message : À l'étape 4, l'utilisateur peut choisir de supprimer un message envoyé ("Supprimer des messages"). Le système supprime le message et informe le destinataire. - Échanger des fichiers : À l'étape 4, l'utilisateur peut choisir d'envoyer un fichier ("Échanger des fichiers"). Le système enregistre le fichier et le transmet au destinataire. - Répondre à un message : À l'étape 4, l'utilisateur peut choisir de répondre à un message spécifique ("Relire aux messages"). Le système lie la réponse au message initial. - Échanger des photos : À l'étape 4, l'utilisateur peut choisir d'envoyer une photo ("Échanger des photos"). Le système enregistre la photo et la transmet au destinataire. - Passer un appel : À l'étape 4, l'utilisateur peut choisir de passer un appel ("Passer un appel"). Le système initie l'appel entre les deux parties si elles sont disponibles. - Utilisateur hors ligne : À l'étape 5, si le destinataire est hors ligne, le système stocke le message (ou fichier/photo) et le transmet dès que le destinataire se reconnecte.
Exigences Spéciales	<p>Les messages doivent être transmis en temps réel. Les appels nécessitent une connexion stable et la disponibilité des deux parties. Les fichiers et photos doivent respecter une limite de taille (ex. : 10 Mo).</p>

Table 3.5 – Description textuel du cas d'utilisation : Échanger les messages

III.3 Spécification et raffinement des cas d'utilisation pour un "Influencer"

1 Consulter les offres de collaborations

La figure 3.8 présente le raffinement du cas d'utilisation "Consulter les offres de collaborations" pour un Influencer.

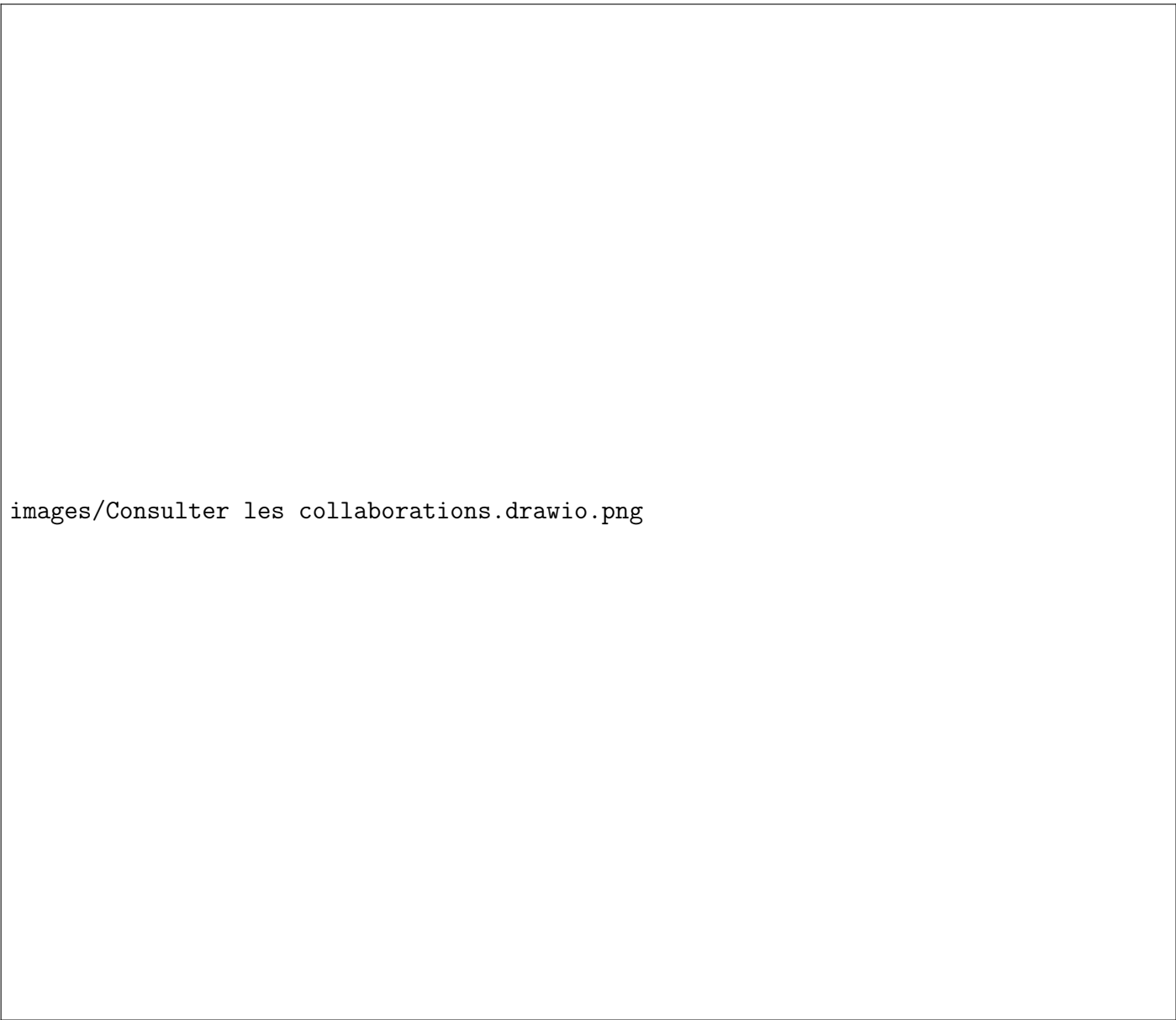


FIGURE 3.7 – Diagramme de cas d'utilisation "Consulter les offres de collaborations"

Section	Contenu
Nom du cas d'utilisation	Consulter les collaborations
Acteur principal	Utilisateur (rôle : Influenceur)

Section	Contenu
Acteurs secondaires	Système (Flenci)
Préconditions	L'Influenceur doit être authentifié. Des collaborations doivent exister pour l'Influenceur.
Postconditions	L'Influenceur a consulté ses collaborations dans les statuts pending, going ou done, avec les détails associés.
Brève Description	Permet à un Influenceur de consulter ses collaborations selon leur statut (pending, going, done) et de voir les détails associés après authentification par le système.
Flux Principal	<ol style="list-style-type: none"> 1. L'Influenceur accède à la section des collaborations. 2. Le système affiche les collaborations selon leur statut : <ul style="list-style-type: none"> - "Consulter les collaborations en attente (Pending)" pour les statuts pending. - "Consulter les collaborations en cours (Going)" pour les statuts going. - "Consulter les collaborations terminées (Done)" pour les statuts done. 3. Le cas se termine.
Flux Alternatifs	<p>A1. Si aucune collaboration n'est disponible pour un statut donné, un message "Aucune collaboration" est affiché.</p> <p>A2. Si l'Influenceur n'a aucune collaboration, un message "Aucune collaboration trouvée" est affiché.</p> <p>A3. Depuis l'étape 2, l'Influenceur peut choisir d'accepter une offre en attente :</p> <ul style="list-style-type: none"> - L'Influenceur sélectionne une collaboration en attente. - Le système exécute "Accepter une offre de collaboration", incluant "Prévisualiser et accepter les termes et conditions". - La collaboration passe au statut "Going". <p>A4. Depuis l'étape 2, l'Influenceur peut choisir de terminer une collaboration en cours :</p> <ul style="list-style-type: none"> - L'Influenceur sélectionne une collaboration en cours. - Le système exécute "Terminer une offre de collaboration", incluant "Fournir une raison de terminaison". - La collaboration passe au statut "Done".
Exigences Spéciales	L'interface doit permettre un filtrage clair par statut (pending, going, done).

Table 3.6 – Description textuel du cas d'utilisation : Consulter les offres de collaborations

III.4 Spécification et raffinement des cas d'utilisation pour un "Company"

1 Rechercher les influenceurs

La figure 3.9 présente le raffinement du cas d'utilisation "Rechercher les influenceurs" pour une Company.

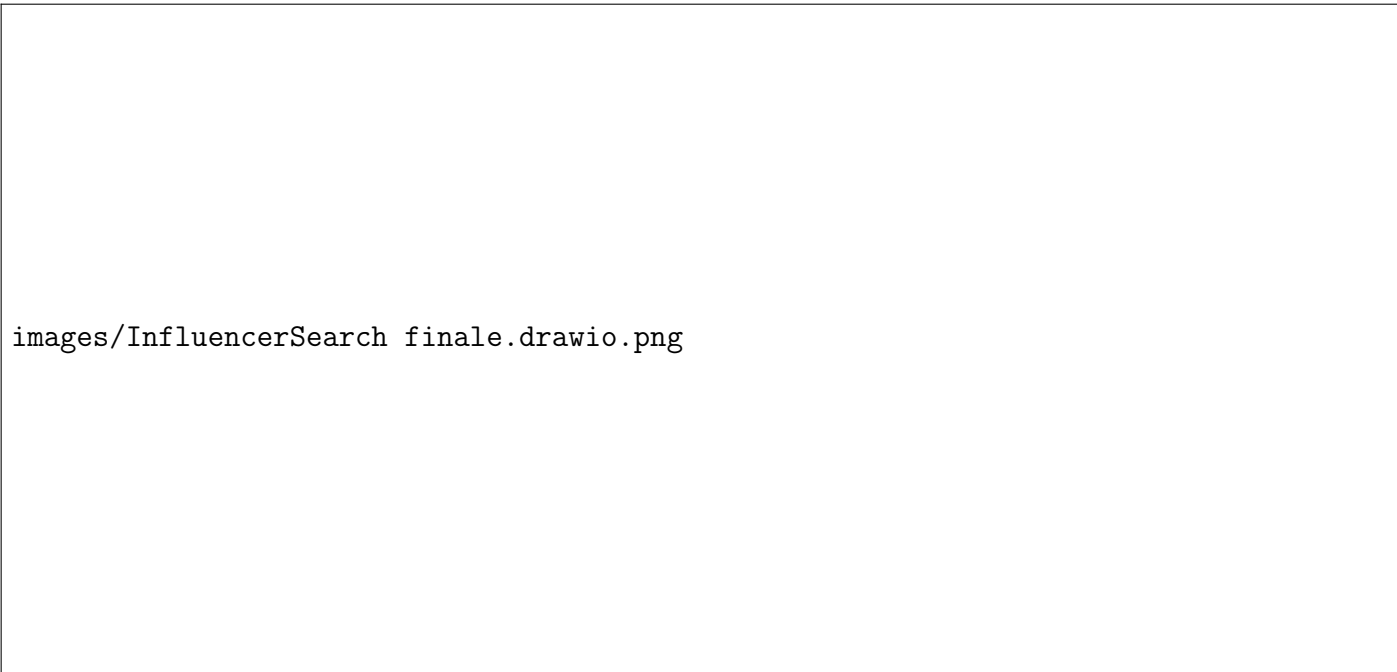


FIGURE 3.8 – Diagramme de cas d'utilisation "Rechercher les influenceurs"

Section	Contenu
Nom du cas d'utilisation	Rechercher les influenceurs
Acteur principal	Utilisateur (rôle : Influencer)
Acteurs secondaires	Système
Préconditions	L'utilisateur doit être authentifié (relation «include» avec "Authentification"). Des influenceurs doivent être disponibles dans la base de données.
Postconditions	L'Influencer a effectué une recherche d'autres influenceurs, avec la possibilité d'accéder à leurs statistiques ou de naviguer vers leurs profils de réseaux sociaux si désiré.
Brève Description	Permet à un Influencer de rechercher d'autres influenceurs via un menu dédié. L'Influencer peut optionnellement consulter leurs statistiques ou naviguer vers leurs profils de réseaux sociaux pour explorer leur contenu.
Flux Principal	1. L'Influencer accède au menu de recherche des influenceurs.

Section	Contenu
	<p>2. L'Influencer effectue une recherche (par exemple, par nom ou critères).</p> <p>3. Le système affiche une liste d'influenceurs correspondants.</p> <p>4. Le cas se termine.</p>
Flux Alternatifs	<p>- Aucun influenceur trouvé : À l'étape 3, si aucun influenceur ne correspond à la recherche, un message "Aucun influenceur trouvé" est affiché, et le cas se termine.</p> <p>- Consulter les statistiques des influenceurs : À l'étape 3, l'Influencer peut choisir de consulter les statistiques des influenceurs listés ("Consulter les statistiques des influenceurs"). Le système affiche des détails comme le nombre de followers, publications, vues, reach, et impressions pour chaque influenceur, puis le cas se termine.</p> <p>- Naviguer vers le profil de réseau social : À l'étape 3, l'Influencer peut choisir de naviguer vers le profil de réseau social d'un influenceur ("Naviguer vers le profil de réseau social"). Le système redirige vers le lien du profil social, puis le cas se termine.</p> <p>- Action annulée : À l'étape 3, si l'Influencer choisit une extension mais annule, le cas se termine sans action supplémentaire.</p>
Exigences Spéciales	Le système doit valider l'authenticité de l'utilisateur et assurer que les données des influenceurs (statistiques et liens de réseaux sociaux) sont à jour dans la base de données.

Table 3.7 – Description textuel du cas d'utilisation : Rechercher les influenceurs

2 Envoyer les offres de collaborations

La figure 3.10 présente le raffinement du cas d'utilisation "Envoyer les offres de collaborations" pour une Company.

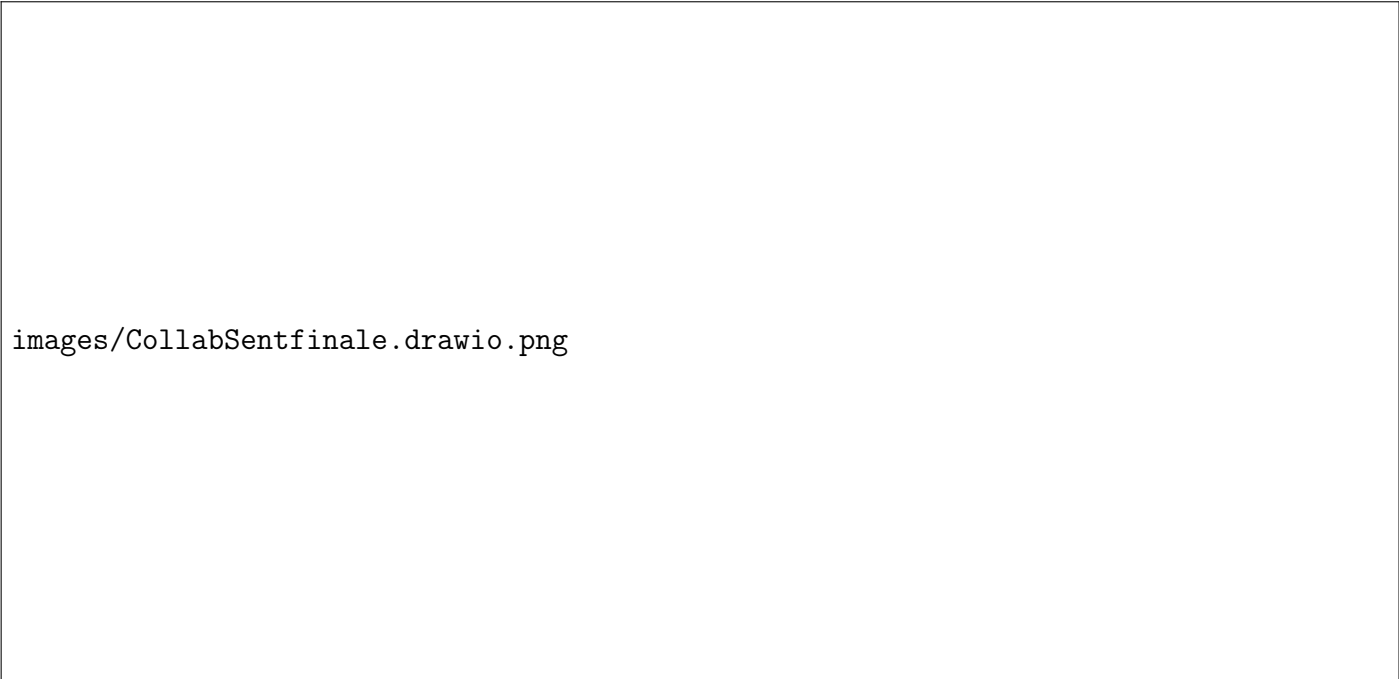


FIGURE 3.9 – Diagramme de cas d'utilisation "Envoyer les offres de collaborations"

Section	Contenu
Nom du cas d'utilisation	Envoyer les offres de collaborations
Acteur principal	Utilisateur (rôle : Company)
Acteurs secondaires	Système
Préconditions	L'utilisateur doit être authentifié (relation «include» avec "Authentification"). Des influenceurs doivent être disponibles dans la base de données pour la sélection.
Postconditions	Une offre de collaboration est créée et envoyée à l'Influencer sélectionné, avec les détails (titre, description, budget, durée) enregistrés. Une notification est envoyée à l'Influencer.
Brève Description	Permet à une Company de créer et d'envoyer une offre de collaboration à un Influencer, en sélectionnant un Influencer cible, définissant le titre, la durée, le budget et la description de l'offre, puis en envoyant une notification à l'Influencer.
Flux Principal	1. La Company accède à la section d'envoi d'offres de collaboration. 2. Le système inclut "Sélectionner un influenceur" : la Company choisit un Influencer cible. 3. Le système inclut "Définir le titre de l'offre" : la Company entre le titre de l'offre.

Section	Contenu
	<p>4. Le système inclut "Définir la durée de l'offre" : la Company spécifie la durée de la collaboration.</p> <p>5. Le système inclut "Définir le budget" : la Company entre le budget estimé.</p> <p>6. Le système inclut "Écrire une description de l'offre" : la Company rédige la description de l'offre.</p> <p>7. Le système enregistre l'offre avec le statut "pending".</p> <p>8. Une notification est envoyée à l'Influencer (via un mécanisme de notification).</p> <p>9. Le cas se termine.</p>
Flux Alternatifs	<p>- Influenceur non valide : À l'étape 2, si l'Influencer sélectionné n'existe pas ou n'est pas valide, un message d'erreur "Influenceur non trouvé" est affiché, et le cas se termine.</p> <p>- Champs manquants : À l'étape 3, 4, 5 ou 6, si l'un des détails obligatoires (titre, durée, budget, description) n'est pas fourni, un message d'erreur "Champs manquants" est affiché, et le flux retourne à l'étape correspondante.</p> <p>- Autorisations insuffisantes : À l'étape 1, si la Company n'a pas les autorisations nécessaires (par exemple, compte limité), un message d'erreur est affiché, et le cas se termine.</p>
Exigences Spéciales	<p>Le système doit valider que les champs obligatoires (titre, durée, budget, description) sont remplis avant l'enregistrement de l'offre.</p> <p>Les notifications doivent être envoyées en temps réel à l'Influencer.</p>

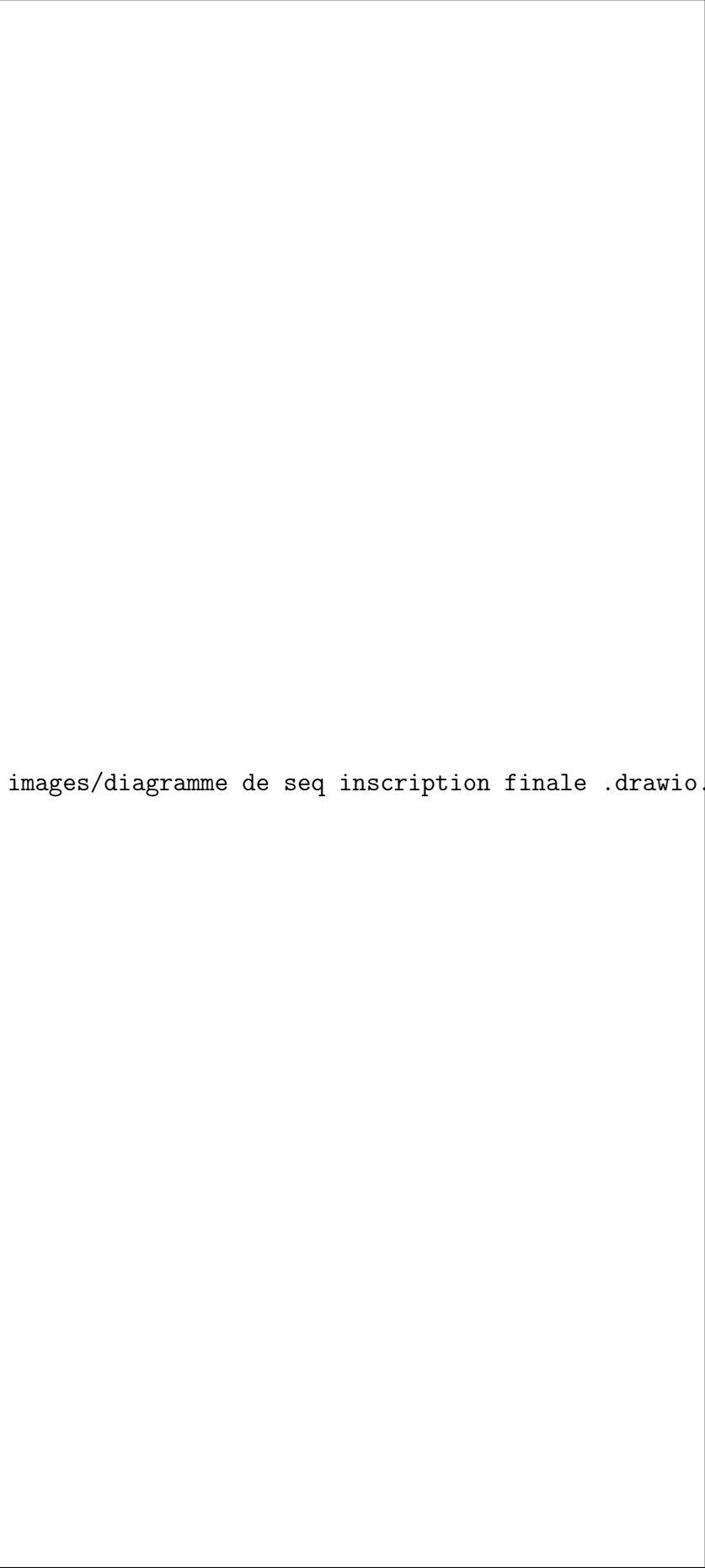
Table 3.8 – Description textuel du cas d'utilisation : Envoyer les offres de collaborations

III.5 Diagrammes de séquences

Pendant cette étape, une fois les cas d'utilisation identifiés, nous illustrons quelques scénarios à l'aide de diagrammes de séquences.

1 Créer un compte pour s'inscrire

La figure 3.11 présente le diagramme de séquence détaillé du processus de création du compte pour s'inscrire à l'application.



images/diagramme de seq inscription finale .drawio.png

FIGURE 3.10 – Diagramme de séquence "S'inscrire"

2 Authentication

La figure 3.12 présente le diagramme de séquence détaillé du processus d'authentification de l'application.

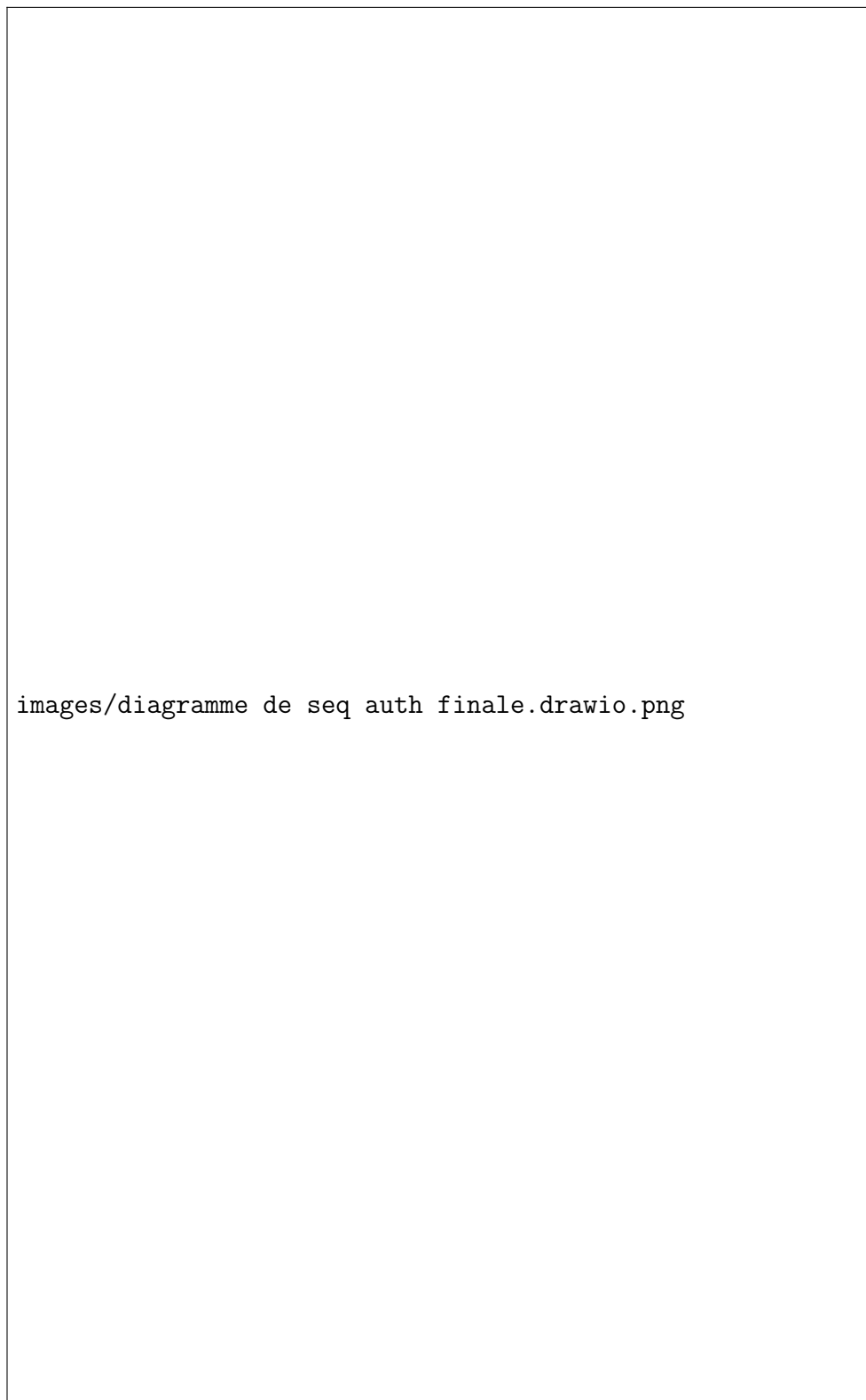


FIGURE 3.11 – Diagramme de séquence "Authentication"

3 Mise à jour et Modifier Profil

La figure 3.13 présente le diagramme de séquence pour le cas d'utilisation "Mise à jour et Modifier Profil" de l'application.

images/diagramm sÃ¼q modifier finale.drawio.png

FIGURE 3.12 – Diagramme de s  quence "Mise    jour et Modifier Profil"

4 Recevoir des Notifications

La figure 3.14 présente le diagramme de séquence pour le cas d'utilisation "Recevoir des Notifications" de l'application.

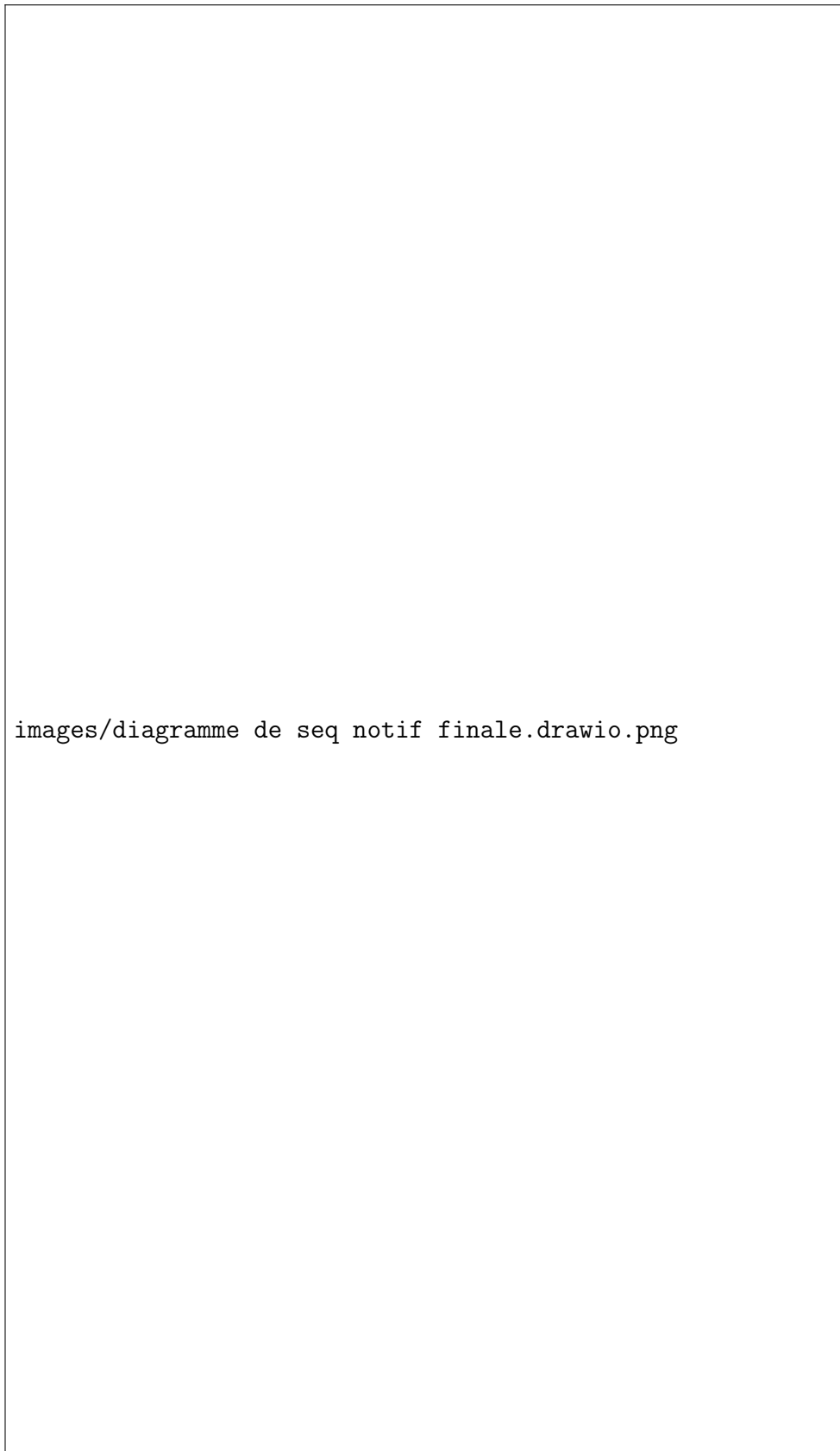


FIGURE 3.13 – Diagramme de séquence "Consulter les notifications"

5 Échanger les messages

La figure 3.15 présente le diagramme de séquence pour le cas d'utilisation "Échanger les Messages" de l'application.

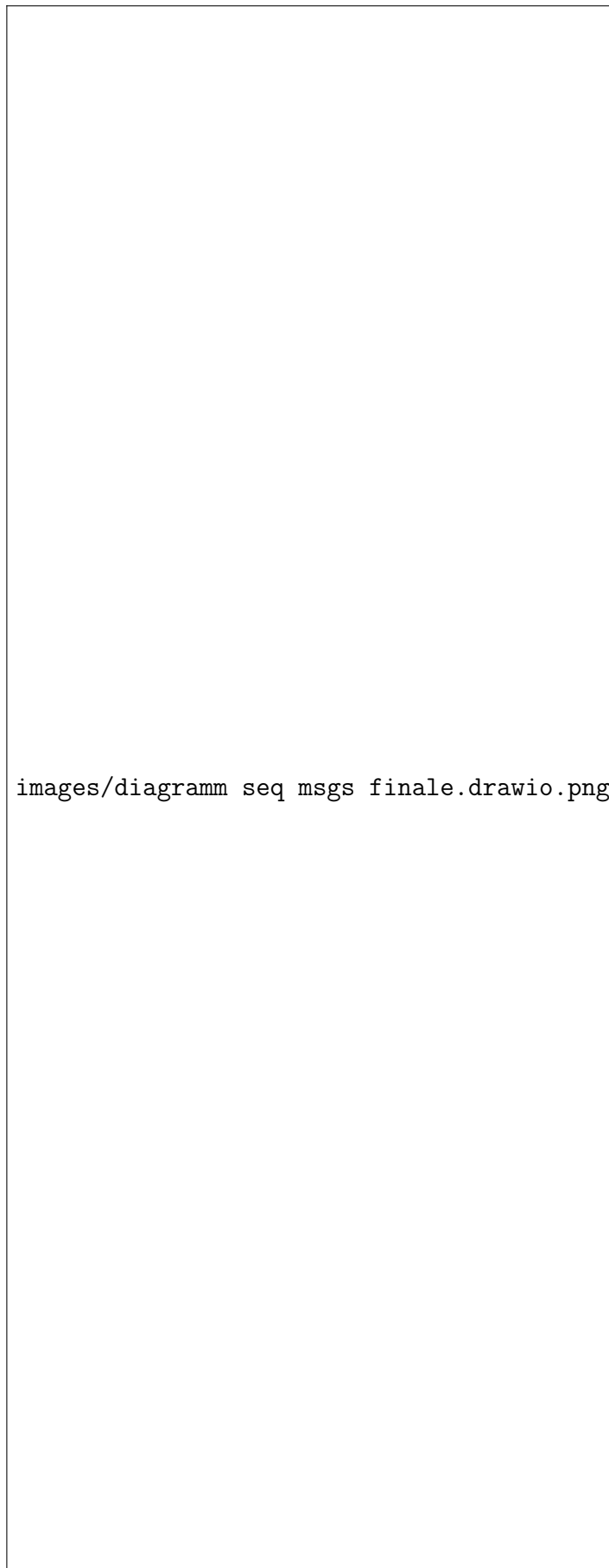


FIGURE 3.14 – Diagramme de séquence "Échanger les Messages"

6 Consulter les offres de collaborations

La figure 3.8 présente le diagramme de séquence pour le cas d'utilisation "Consulter les offres de collaborations" de l'application.

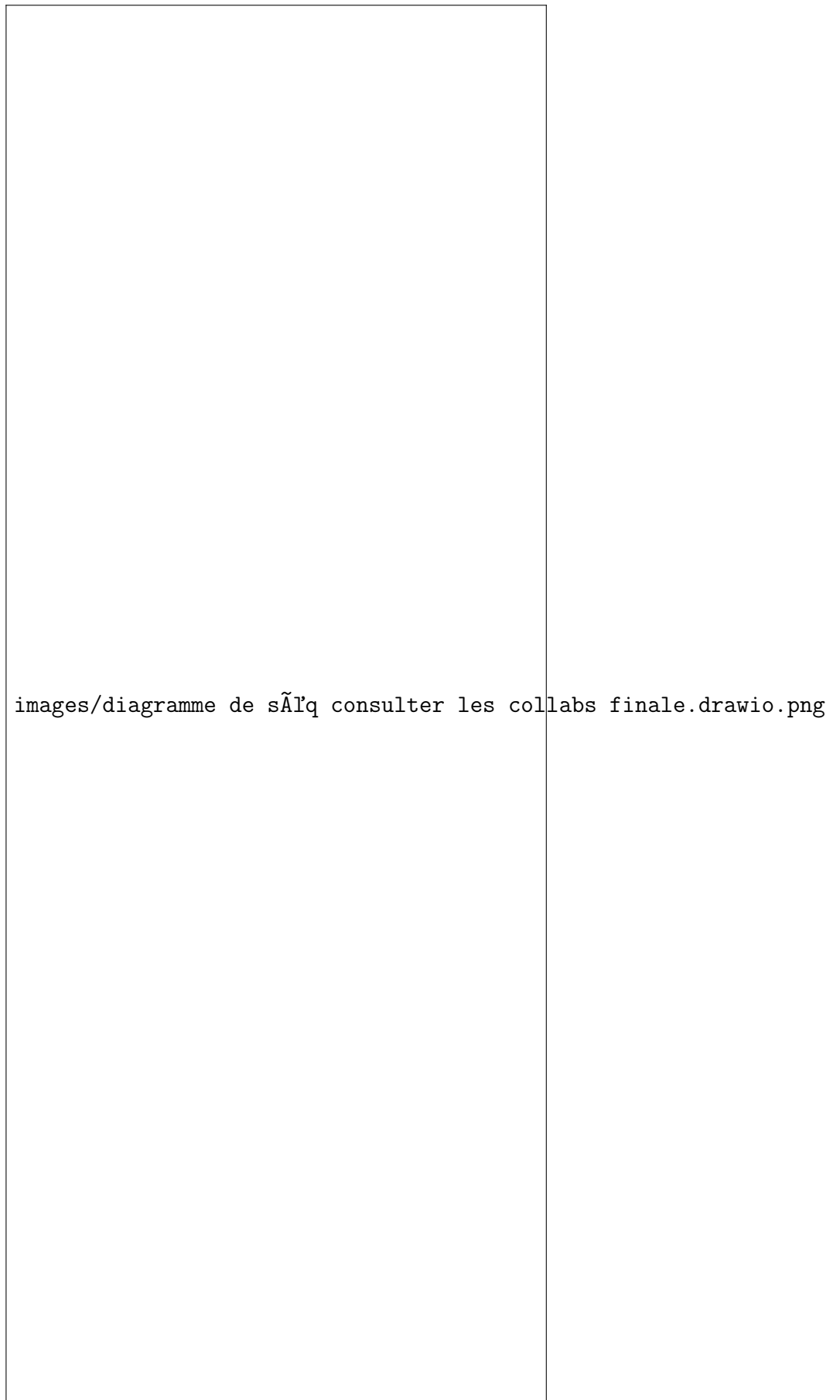


FIGURE 3.15 – Diagramme de s  quence "Consulter les offres de collaborations"

7 Rechercher les Influenceurs

La figure 3.17 présente le diagramme de séquence pour le cas d'utilisation "Rechercher les Influenceurs" de l'application.

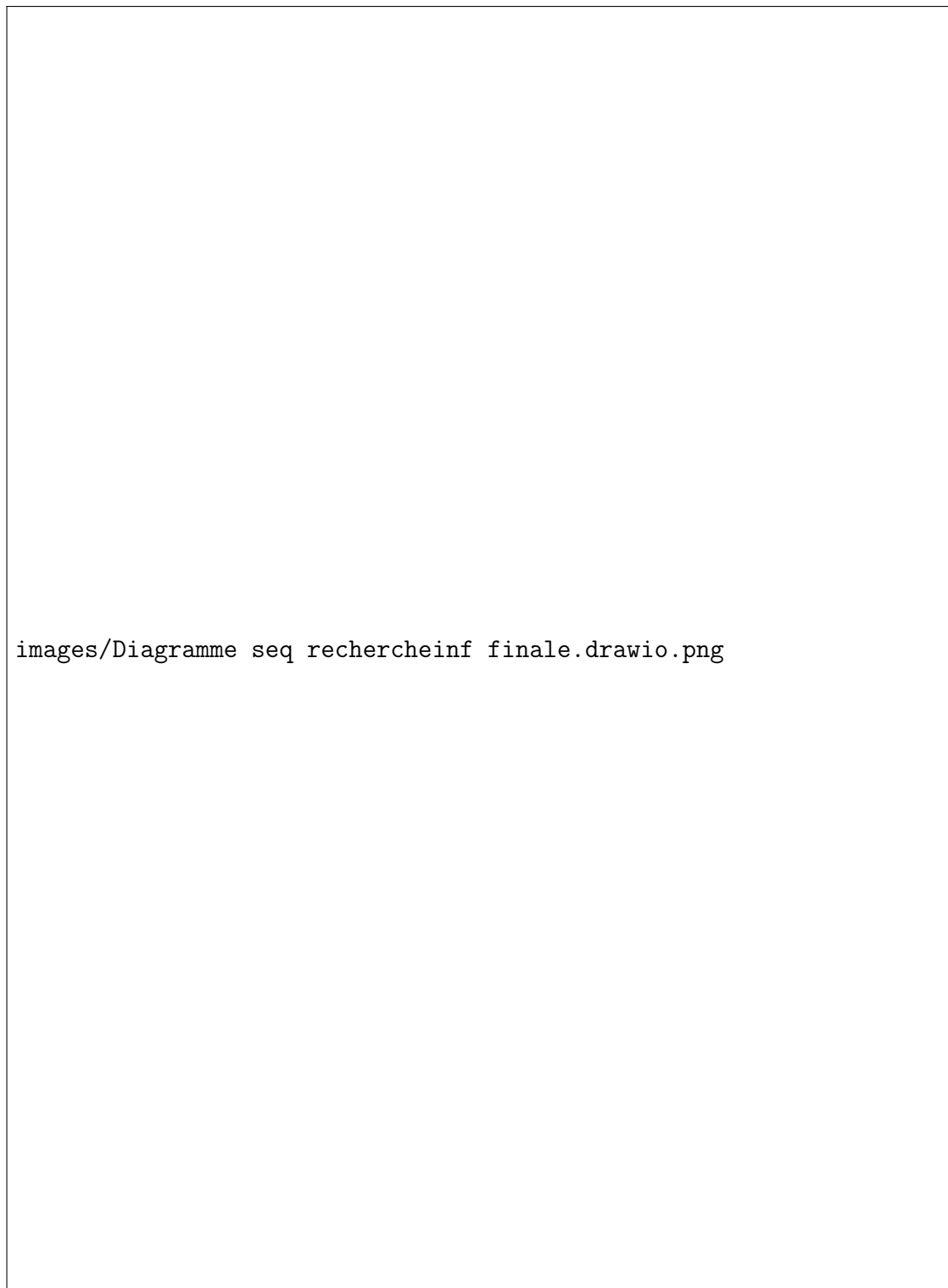


FIGURE 3.16 – Diagramme de séquence "Rechercher les Influenceurs"

8 Envoyer les Offres de Collaborations

La figure 3.11 présente le diagramme de séquence pour le cas d'utilisation "Envoyer les Offres de Collaborations" de l'application.

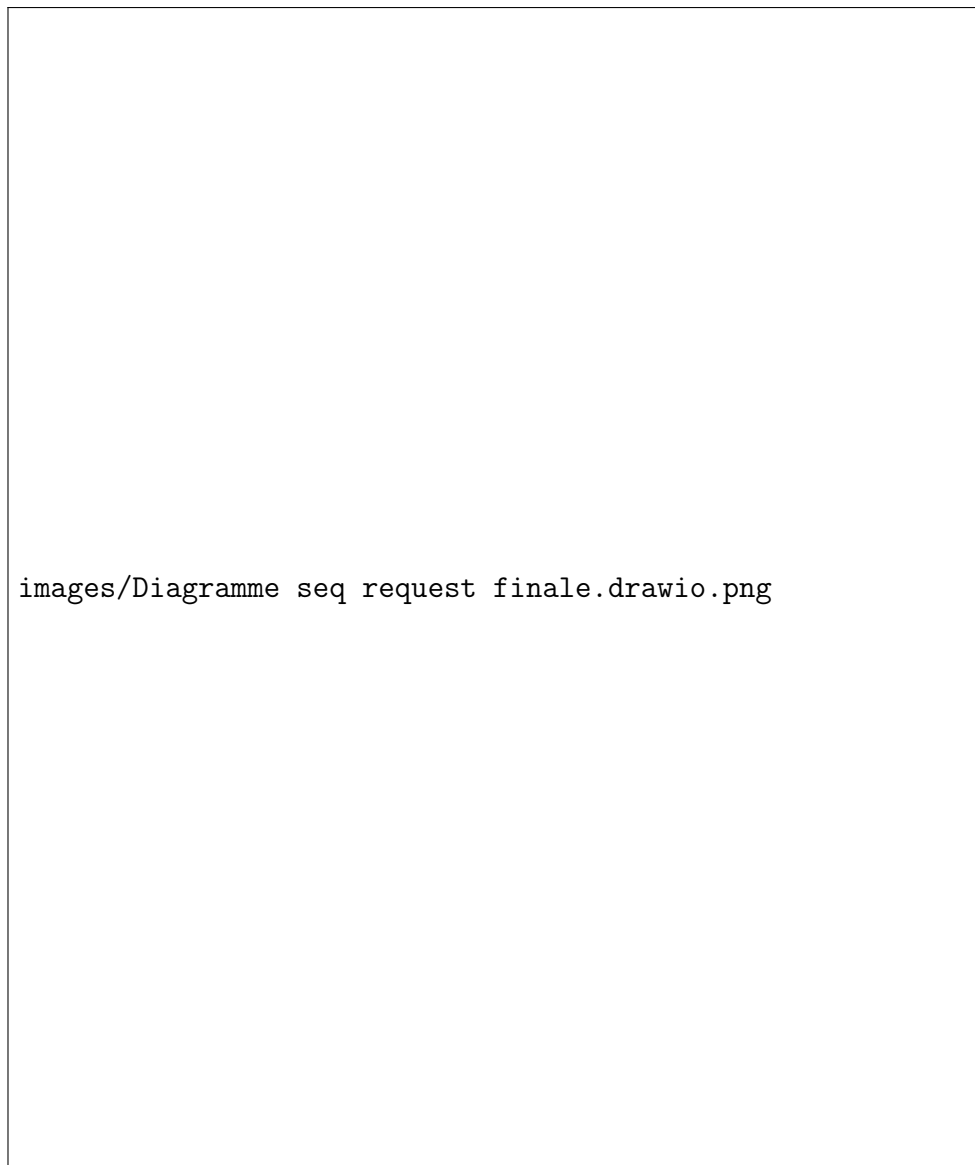


FIGURE 3.17 – Diagramme de séquence "Envoyer les Offres de Collaborations"

IV Conclusion

Les diagrammes exposés dans ce chapitre constituent une base solide pour la conception technique du projet. Ils permettent d'avoir une vision claire sur le déroulement des traitements, les échanges entre composants et les conditions d'exécution dans un environnement réel. Ces éléments faciliteront par la suite le développement et l'implémentation du système.

Chapitre 4

Réalisation

I Introduction

Une fois la phase de conception achevée, nous sommes passés à l'étape finale de la réalisation du projet. Dans ce chapitre, nous débutons en présentant l'environnement logiciel utilisé pour la création de notre système, ainsi que les diverses interfaces de notre application.

II Environnement de développement

II.1 Technologies utilisées :

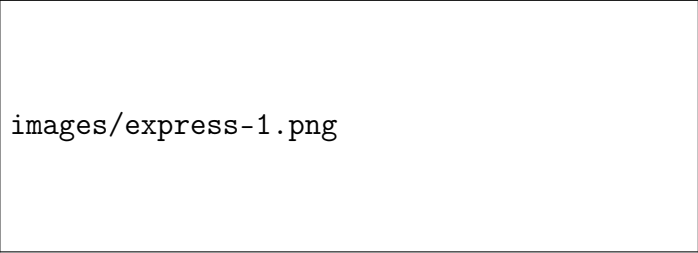
1 NodeJs :

Node.js® est un environnement d'exécution JavaScript gratuit, open-source et multiplateforme qui permet aux développeurs de créer des serveurs, des applications web, des outils en ligne de commande et des scripts [2].



2 ExpressJs :

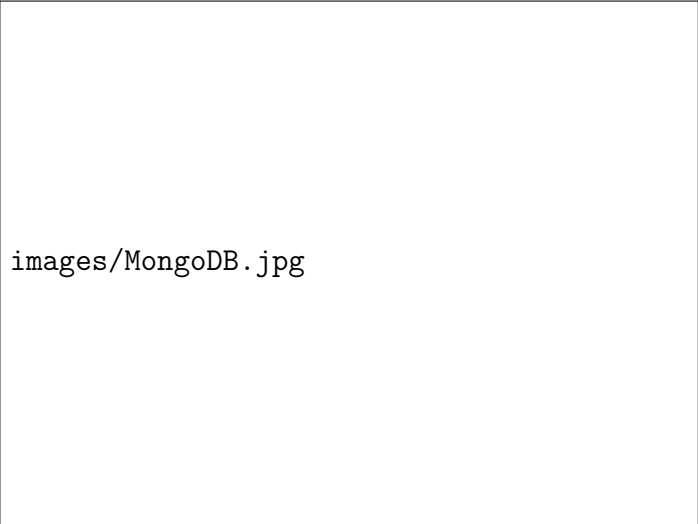
Express.js est un framework pour construire des applications web basées sur Node.js. C'est de fait le framework standard pour le développement de serveur en Node.js [3].



images/express-1.png

3 MongoDB

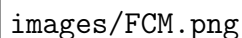
MongoDB est une base de données orientée documents. En clair, vous bénéficiez de la scalabilité et de la flexibilité que vous voulez, avec les fonctions d'interrogation et d'indexation qu'il vous faut. MongoDB stocke les données dans des documents flexibles de type JSON, ce qui signifie que les champs peuvent varier d'un document à l'autre et que la structure des données peut être modifiée au fil du temps [4].



images/MongoDB.jpg

4 Firebase Cloud Messaging

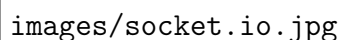
Firebase Cloud Messaging (FCM) est une solution de messagerie multiplateforme qui vous permet d'envoyer des messages de manière fiable. Avec FCM, vous pouvez avertir une application cliente qu'un nouvel e-mail ou d'autres données sont disponibles à synchroniser. Vous pouvez envoyer des messages de notification pour favoriser le réengagement et la fidélisation des utilisateurs [5].



images/FCM.png

5 Socket.io

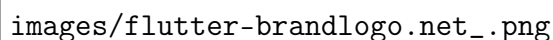
Socket.IO est une bibliothèque qui permet une communication à faible latence, bidirectionnelle et basée sur les événements entre un client et un serveur [6].



images/socket.io.jpg

6 Flutter

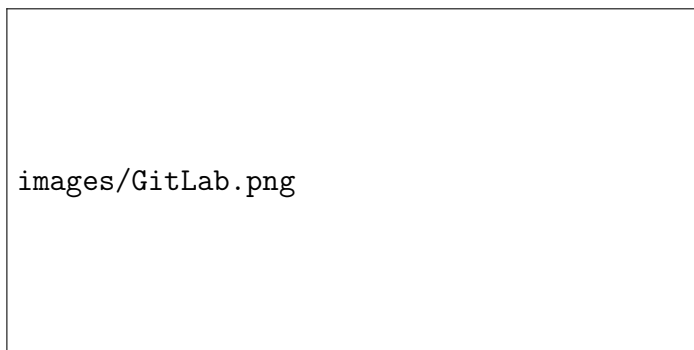
Flutter est un cadre open source développé et pris en charge par Google. Les développeurs front-end et full-stack utilisent Flutter pour créer l'interface utilisateur (UI) d'une application pour plusieurs plateformes avec une seule base de code. Flutter utilise le langage de programmation open source Dart, qui a été mis au point par Google. Dart est optimisé pour la création d'interfaces utilisateur, et nombre de ses atouts sont inclus dans Flutter [7].



images/flutter-brandlogo.net_.png

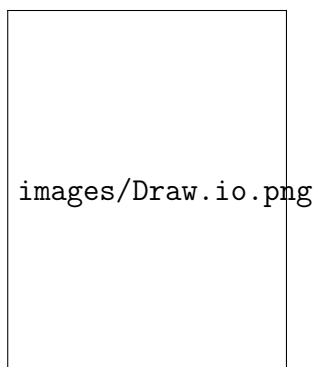
7 GitLab

GitLab est une plateforme DevOps open source très utile pour les développeurs d'application qui désirent améliorer leur productivité. À l'heure où la performance rime avec l'automatisation des process, l'on découvre perpétuellement de nouveaux systèmes d'optimisation de la productivité sous toutes ses formes. GitLab est un outil de gestion de version de code source basé sur Git, une technologie de gestion de code source très populaire [8].



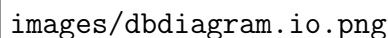
8 Draw.io

Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Draw.io est un véritable couteau suisse de la frise chronologique, de la carte mentale et des diagrammes de tout genre [9].



9 dbdiagram.io

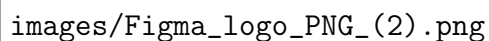
dbdiagram.io est un outil gratuit permettant d'élaborer des diagrammes de base de données en ligne très rapidement et simplement. L'objectif principal de l'outil est que les utilisateurs (pas forcément des développeurs) puissent comprendre simplement la structure d'une base de données [10].



images/dbdiagram.io.png

10 Figma

Figma est une plateforme collaborative pour éditer des graphiques vectoriels et faire du prototypage. Elle permet de concevoir des design systems pour faciliter la création de sites web et d'applications mobiles. C'est une solution à destination des UI et UX designers et des développeurs [11].



images/Figma_logo_PNG_(2).png

III Interfaces d'applications

Cette section est spécialement conçue pour montrer et expliquer en détail les différentes interfaces de notre plateforme. Elle a pour objectif d'aider les utilisateurs à mieux comprendre et à profiter pleinement de leur expérience sur notre plateforme.

La figure 4.1 présente l'interface "Login", permettant à l'utilisateur de s'authentifier via son nom d'utilisateur et son mot de passe.



FIGURE 4.1 – Login

La figure 4.2 présente l’interface “Register”, permettant à l’utilisateur de créer un compte. Elle offre deux options distinctes d’inscription : “Register as Company” et “Register as Influencer”.



FIGURE 4.2 – Register

La figure 4.3 présente l’interface “Register as Project Owner”, permettant à une entreprise de s’enregistrer en renseignant les informations nécessaires à la création de son compte.



FIGURE 4.3 – Register as project owner

La figure 4.4 présente les interfaces “Register as an Influencer”, permettant à un utilisateur de s’inscrire en tant qu’influenceur en fournissant les informations personnelles, les centres d’intérêt ainsi que les statistiques liées à son activité.

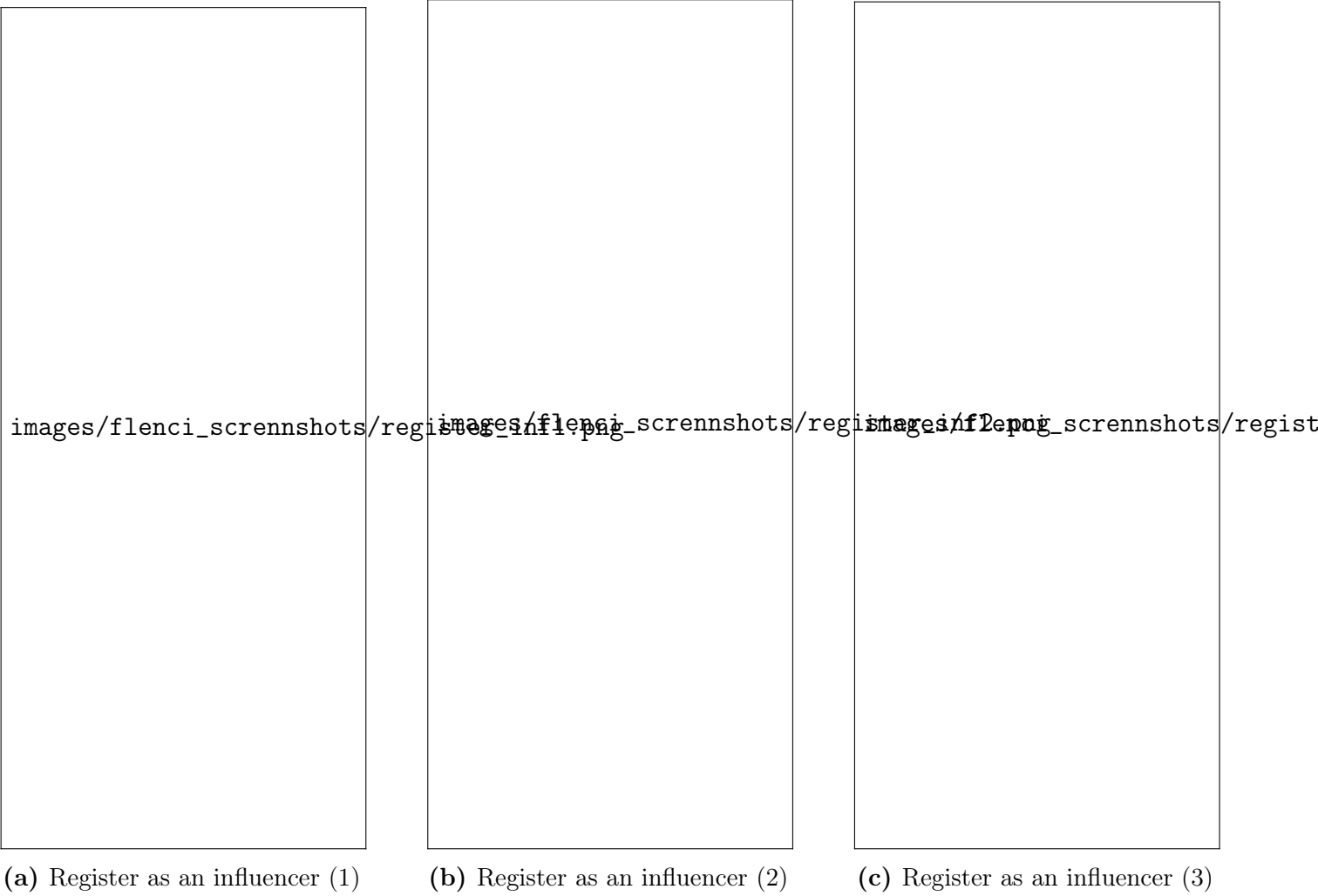


FIGURE 4.4 – Register as an influencer

Les figures 4.6 et 4.7 présentent les interfaces “Forgot Password”, permettant à l'utilisateur de réinitialiser son mot de passe. Ce processus se déroule en plusieurs étapes, incluant la saisie de l'adresse e-mail, la vérification par code OTP, ainsi que la définition d'un nouveau mot de passe.

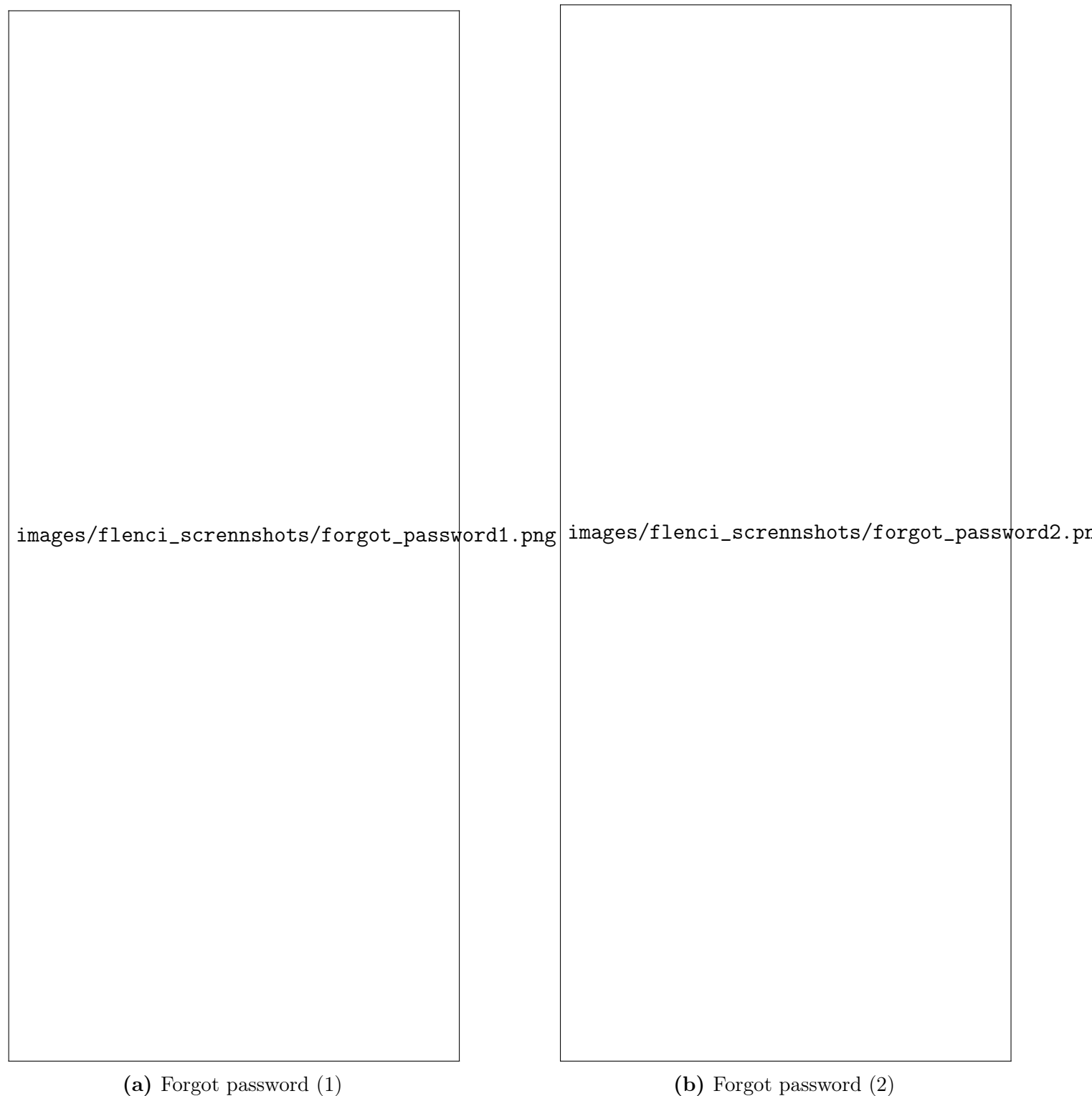


FIGURE 4.5 – Forgot password (Part 1)

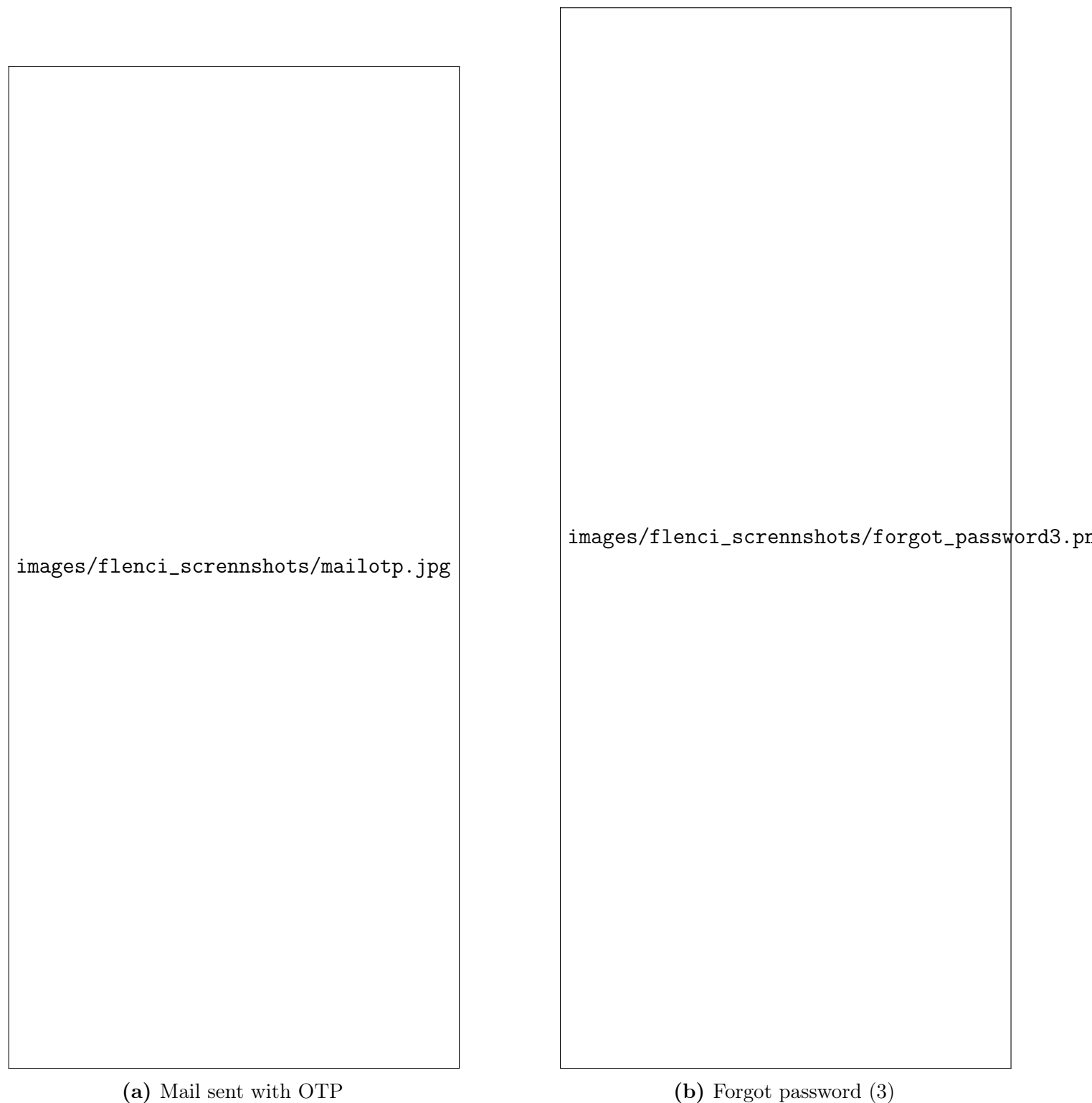


FIGURE 4.6 – Forgot password (Part 2)

La figure 4.7 présente l’interface “Menu”, qui regroupe l’ensemble des influenceurs disponibles sur la plateforme.



FIGURE 4.7 – Menu

La figure 4.8 présente l’interface “Influencer Profile”, permettant de consulter les informations détaillées d’un influenceur, telles que ses centres d’intérêt, ses statistiques de performance et ses réseaux sociaux.



FIGURE 4.8 – Influencer Profile

La figure 4.9 présente l’interface “Collaboration Request”, permettant à une entreprise d’envoyer une demande de collaboration à un influenceur en précisant les détails du projet.



FIGURE 4.9 – Collaboration request

La figure 4.10 présente l’interface “Company Profile”, permettant de consulter les informations détaillées d’une entreprise, incluant ses projets, ses coordonnées et ses offres de collaboration.

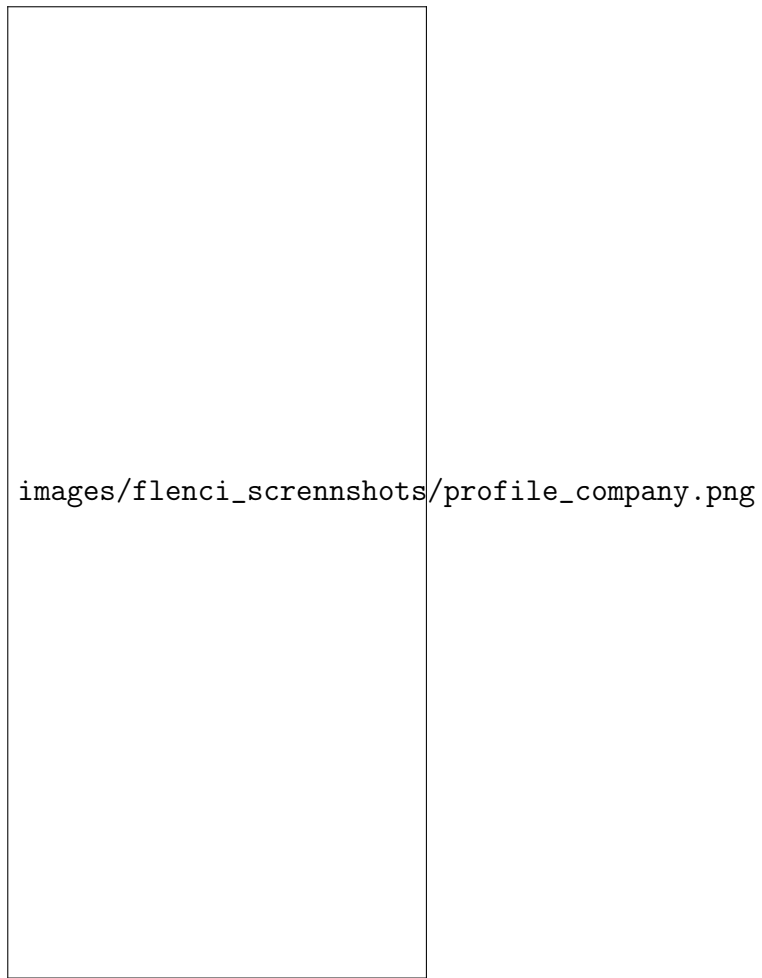


FIGURE 4.10 – Company Profile

La figure 4.11 présente l’interface “Edit Company Profile”, permettant à une entreprise de modifier ses informations sur la plateforme.

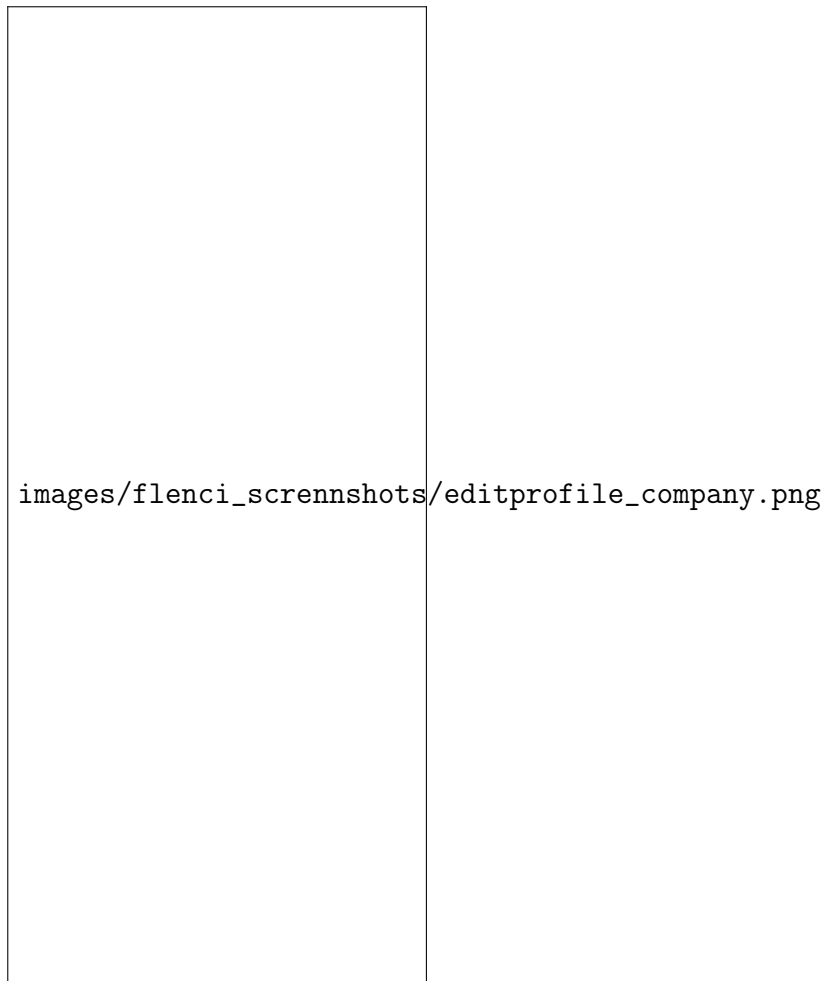


FIGURE 4.11 – Edit Company Profile

La figure 4.12 présente l’interface “Notification Page Company”, affichant les notifications reçues par l’entreprise.

La figure 4.13 présente l’interface “Messages Page Company”, permettant à une entreprise de gérer ses échanges et communications avec les influenceurs.

Les figures 4.14, 4.15 et 4.16 présentent les interfaces "Board" où l’influenceur peut voir les offres de collaborations "going", "pending", "done"



FIGURE 4.12 – Board

La figure 4.17 présente l’interface “Collaboration Details”, affichant les informations complètes d’une demande de collaboration entre une entreprise et un influenceur.



FIGURE 4.13 – Collaboration details

La figure 4.18 présente l’interface “Accept Request”, permettant à un influenceur d’accepter une proposition de collaboration.



FIGURE 4.14 – Accept request

La figure 4.19 présente l’interface “Reject Request”, offrant à un influenceur la possibilité de refuser une demande de collaboration.



FIGURE 4.15 – Reject request

La figure 4.20 présente l’interface “Terminate Request”, permettant à un utilisateur de mettre fin à une collaboration en cours.

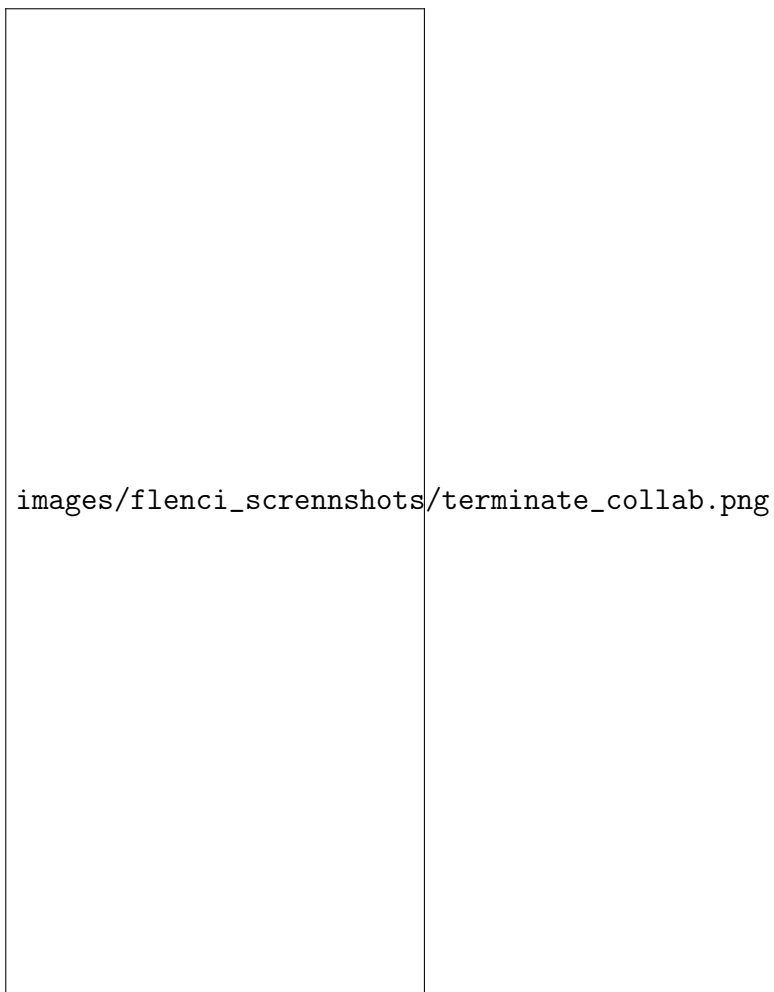


FIGURE 4.16 – Terminate request

La figure 4.21 présente l’interface “Notification Page” pour l’influenceur, affichant les alertes et mises à jour importantes concernant ses collaborations.

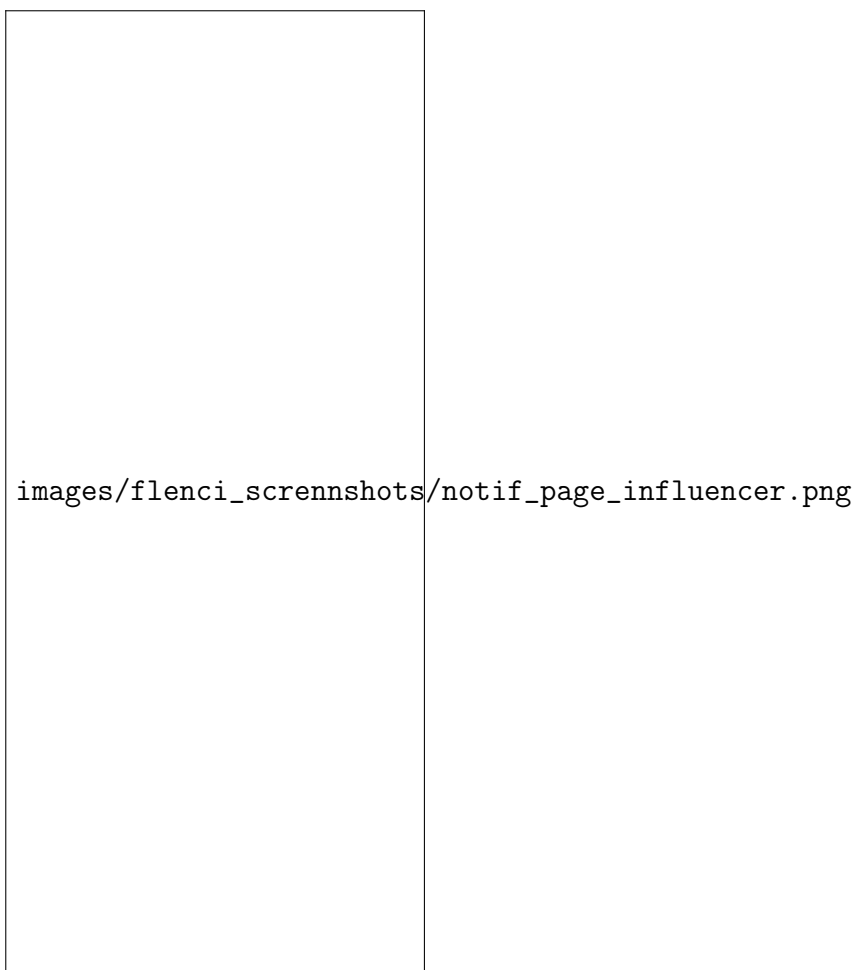


FIGURE 4.17 – Notification page

La figure 4.22 présente l’interface “Convo-box Page” pour l’influenceur, offrant un aperçu des conversations en cours avec les entreprises.



FIGURE 4.18 – Convo-box page

La figure 4.23 présente l’interface “Chat Screen” pour l’influenceur, permettant d’échanger des messages en temps réel avec les entreprises.

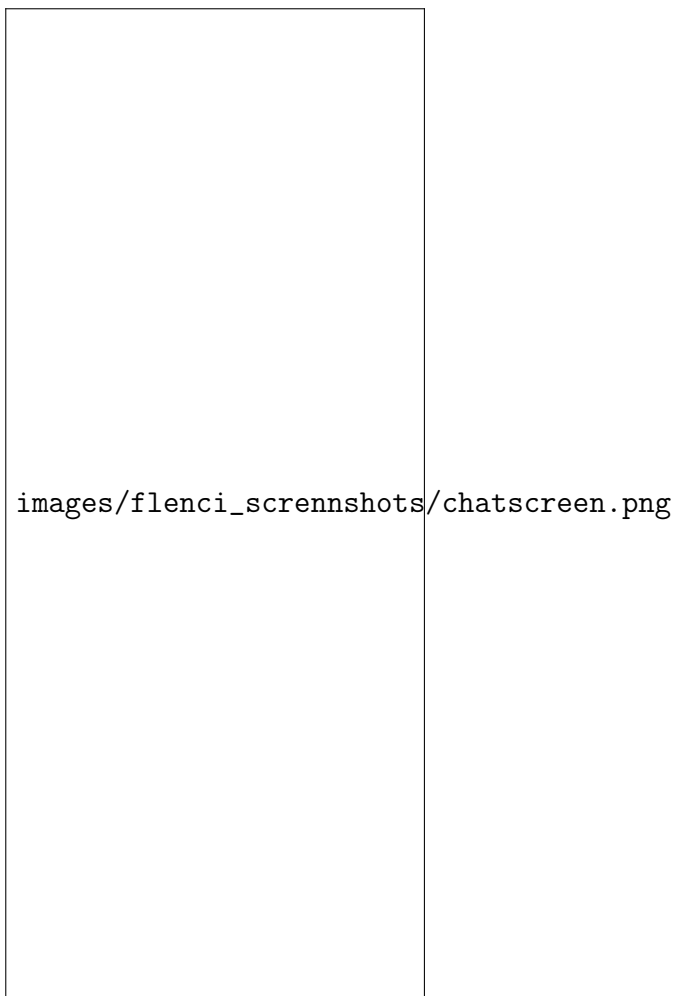


FIGURE 4.19 – Chat screen

La figure 4.24 présente l’interface “Influencer Profile”, permettant de consulter les informations détaillées d’un influenceur, telles que ses centres d’intérêt, ses statistiques et ses réseaux sociaux.

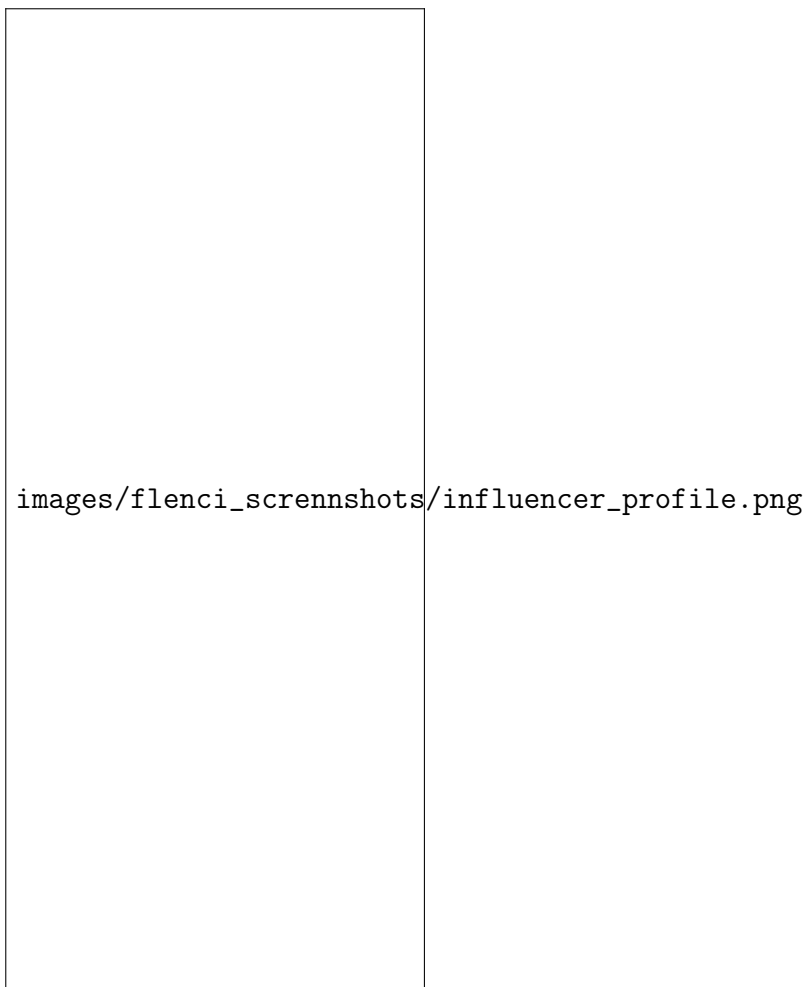


FIGURE 4.20 – Influencer Profile

Les figures 4.25 et 4.26 présentent l’interface “Edit Profile”, offrant à l’influenceur la possibilité de modifier ses informations personnelles et professionnelles.



images/flenci_scrennshots/editprofile_inf1.png

(a) Edit Profile (1)



images/flenci_scrennshots/editpr

(b) Edit Profile (2)

FIGURE 4.21 – Edit Profile

Les figures 4.27 et 4.28 présentent les interfaces de notifications, affichant les alertes et mises à jour importantes destinées à l'influenceur.

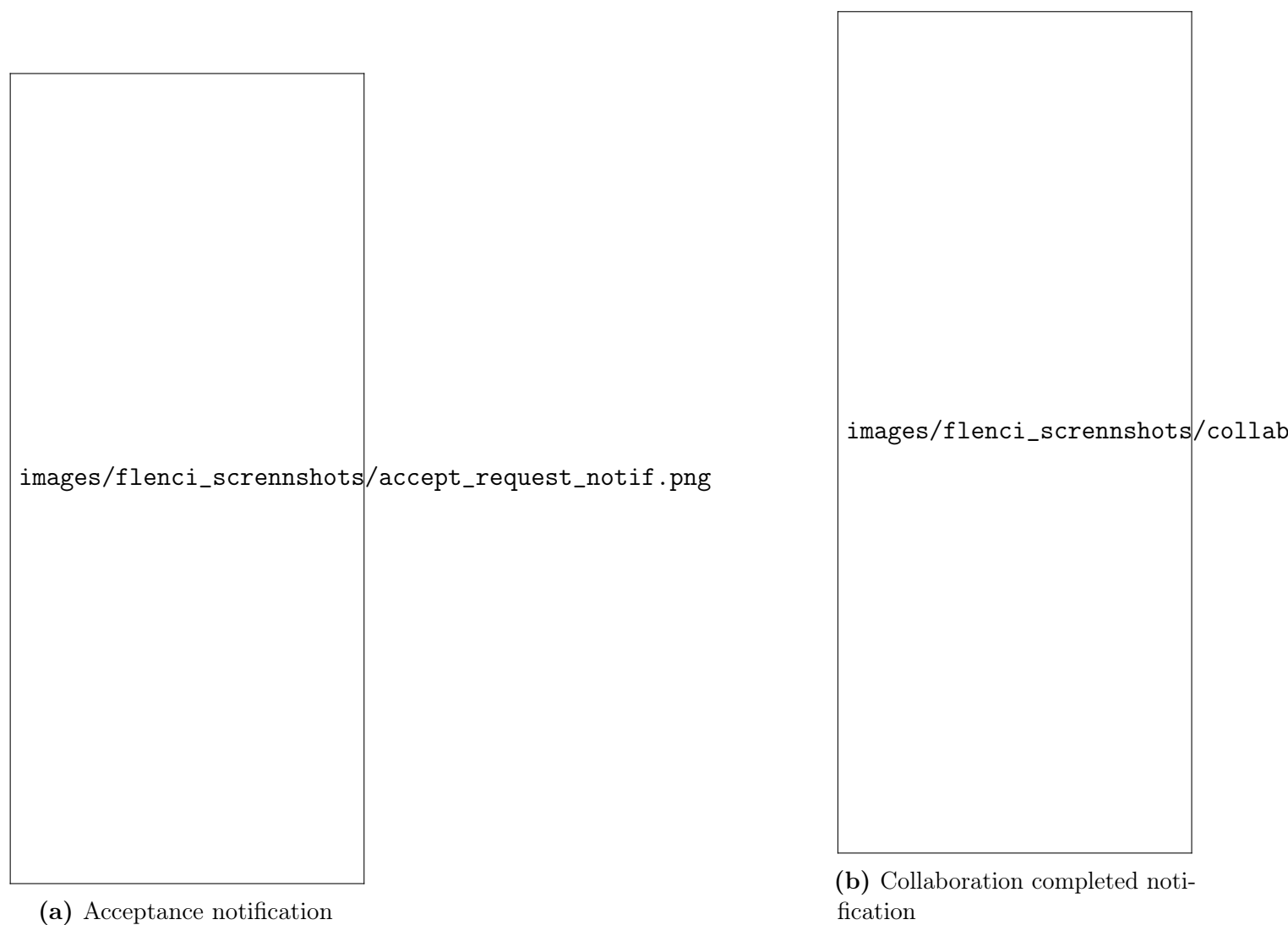


FIGURE 4.22 – Notifications

IV Conclusion

Au cours de ce chapitre, nous avons examiné en détail les outils logiciels que nous utilisons. Ensuite, afin de vous offrir une meilleure compréhension du travail accompli, nous avons inclus des captures d'écran des différentes interfaces utilisateur réelles. Ces captures vous seront utiles en tant que guide d'utilisation.

Conclusion générale

Dans ce rapport, nous avons détaillé le travail accompli en partenariat avec OSTRIX, une entreprise spécialisée dans le développement informatique, dans le cadre de notre projet de fin d'études. Notre objectif était de concevoir **Flenci**, une application mobile visant à structurer et optimiser les collaborations entre entreprises et influenceurs en Tunisie, en répondant aux besoins spécifiques du marché local tout en favorisant des campagnes promotionnelles efficaces.

Ce rapport a retracé les différentes étapes de la réalisation du projet. Tout d'abord, nous avons analysé le contexte général, en présentant l'entreprise hôte, l'étude des solutions existantes, la problématique, ainsi que la méthodologie agile Kanban adoptée pour un développement flexible. Ensuite, nous avons spécifié les besoins fonctionnels (authentification, gestion des profils, mise en relation, messagerie, notifications) et non fonctionnels (performance, sécurité, scalabilité), accompagnés de diagrammes de cas d'utilisation pour illustrer les interactions des acteurs. La partie conception a permis de définir l'architecture MVC, le diagramme de déploiement, les diagrammes de séquence, ainsi que le modèle de base de données, offrant une vision claire de la structure technique de l'application. Enfin, nous avons décrit l'environnement de développement, incluant des technologies modernes telles que Flutter, Node.js, Express.js, MongoDB, Socket.IO et Firebase Cloud Messaging, et présenté les interfaces graphiques à travers des captures d'écran pour une meilleure compréhension de l'expérience utilisateur.

Au cours de ce projet, j'ai acquis de nouvelles compétences, tant sur le plan technique que personnel. J'ai exploré et maîtrisé des technologies innovantes, renforçant ainsi mes compétences en développement mobile et backend. Cette expérience m'a également permis de développer ma capacité à travailler de manière autonome tout en adoptant une approche agile, essentielle pour répondre aux exigences changeantes d'un projet.

Grâce à la solution développée, nous avons atteint une grande partie de nos objectifs initiaux, en proposant une plateforme intuitive et accessible qui facilite les interactions entre entreprises et influenceurs, tout en ouvrant la voie à de nouvelles perspectives d'amélioration. Parmi les évolutions envisagées, nous pouvons citer :

- **Amélioration de la fonctionnalité de chat** : Poursuivre le développement du système de messagerie en temps réel pour offrir une expérience plus fluide, avec des fonctionnalités comme l'envoi de pièces jointes ou l'archivage automatique des conversations terminées.
- **Intégration d'un système de paiement** : Ajouter une fonctionnalité permettant aux entreprises de finaliser les transactions directement sur la plateforme, simplifiant ainsi la gestion des

collaborations.

- **Analytique avancée** : Développer un tableau de bord analytique plus détaillé pour les entreprises, incluant des métriques sur les performances des campagnes (reach, engagement, ROI).
- **Options premium pour les influenceurs** : Proposer des abonnements premium permettant une mise à jour automatique des données de profil (vues, nombre de publications, abonnés, portée) via un système de scraping périodique, offrant ainsi une meilleure visibilité, des recommandations prioritaires et des statistiques détaillées sur leur audience.

En conclusion, le développement de Flenci a été une expérience enrichissante, tant sur le plan technique que sur le plan de la compréhension des besoins du marché tunisien. Cette plateforme constitue une base solide pour de futures améliorations, avec un potentiel significatif pour transformer la manière dont les collaborations entre entreprises et influenceurs sont gérées localement.

Bibliographie

- [1] CARTON, IRIF. *Modèle-Vue-Contrôleur (MVC) avec Swing*. Accédé le 07 mai 2025 à 09 :45 CET. 2020. URL : <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html> (visité le 07/05/2025).
- [2] NODE.JS FOUNDATION. *Node.js*. Accédé le 07 mai 2025. 2025. URL : <https://nodejs.org/fr> (visité le 07/05/2025).
- [3] *Express.js*. Accédé le 07 mai 2025. Wikipédia. 2025. URL : <https://fr.wikipedia.org/wiki/Express.js> (visité le 07/05/2025).
- [4] MONGODB, INC. *Qu'est-ce que MongoDB ?* Accédé le 07 mai 2025. 2025. URL : <https://www.mongodb.com/fr-fr/company/what-is-mongodb> (visité le 07/05/2025).
- [5] GOOGLE FIREBASE. *Firebase Cloud Messaging (FCM)*. Accédé le 07 mai 2025. 2025. URL : <https://firebase.google.com/docs/cloud-messaging?hl=fr> (visité le 07/05/2025).
- [6] SOCKET.IO TEAM. *Qu'est-ce que Socket.IO ?* Accédé le 07 mai 2025. 2025. URL : <https://socket.io/fr/docs/v4/> (visité le 07/05/2025).
- [7] AMAZON WEB SERVICES. *Qu'est-ce que Flutter ?* Accédé le 07 mai 2025. 2025. URL : <https://aws.amazon.com/fr/what-is/flutter/> (visité le 07/05/2025).
- [8] IA SCHOOL. *GitLab : tout connaître sur le logiciel Open Source*. Accédé le 07 mai 2025 à 09 :00 CET. 2024. URL : <https://www.intelligence-artificielle-school.com/ecole/technologies/gitlab-tout-connaître-sur-le-logiciel-open-source/> (visité le 07/05/2025).
- [9] TICE ÉDUCATION. *draw.io : un outil pour dessiner des diagrammes en ligne*. Accédé le 07 mai 2025. 2014. URL : <https://www.tice-education.fr/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne> (visité le 07/05/2025).
- [10] SHEVAREZO. *Créer un schéma de base de données avec dbdiagram.io*. Accédé le 07 mai 2025. 2018. URL : <https://blog.shevarezo.fr/post/2018/10/04/creer-schema-base-de-donnees-dbdiagramio> (visité le 07/05/2025).
- [11] LE BLOG DU MODÉRATEUR. *Figma : l'outil collaboratif pour le design d'interfaces et le prototypage*. Accédé le 07 mai 2025. 2023. URL : <https://www.blogdumoderateur.com/tools/figma/> (visité le 07/05/2025).

Résumé

En partenariat avec la société "OSTRIX", j'ai réalisé le projet Flenci dans le cadre de mon projet de fin d'études en Licence en Ingénierie des Systèmes Informatique à l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse (ISSAT SOUSSE) pour l'année universitaire 2024/2025. Dans le but de faciliter la mise en relation entre influenceurs et entreprises en Tunisie, notamment pour la gestion des collaborations, la communication en temps réel et la promotion de partenariats, nous avons créé la plateforme mobile "Flenci". Cette plateforme conviviale propose des fonctionnalités telles que l'inscription, la connexion, la gestion des profils, les demandes de collaboration, le chat en temps réel et les notifications push via FCM. Grâce à Flenci, nous avons atteint nos objectifs initiaux tout en adoptant une méthodologie agile (Kanban), et nous envisageons de continuer à l'améliorer avec des optimisations du chat et des fonctionnalités premium.

Mots clés : Plateforme mobile, Influenceurs, Flutter, Node.js, Express.js, MongoDB, Socket.IO, FCM