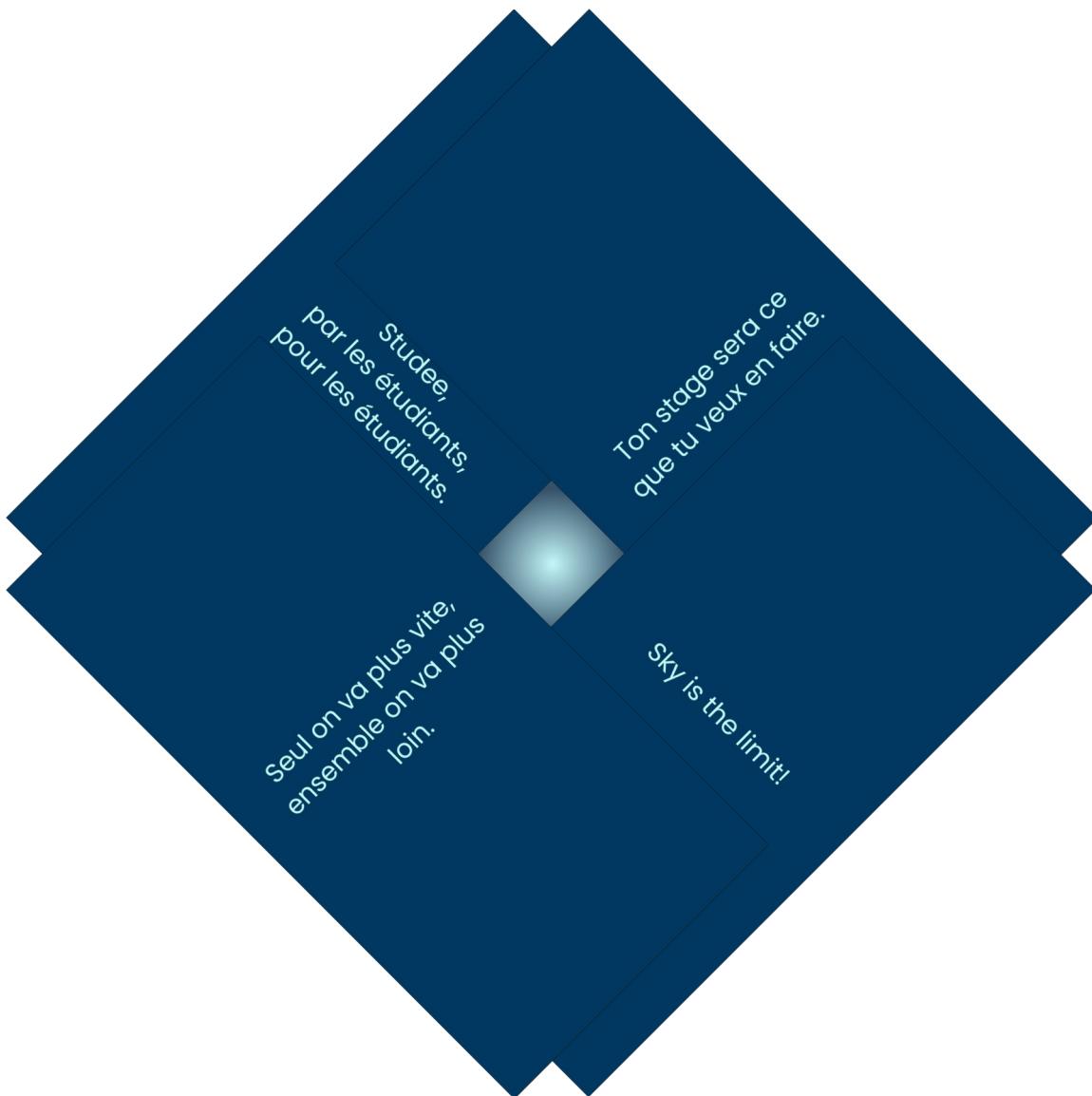




# ATTAR LATIFA

## DÉVELOPPEUR WEB ET WEB MOBILE

### DOSSIER DE PROJET STUDEE





## Table des matières

<b>I. Introduction</b> .....	3
<b>II. Compétences du référentiel couvertes par le projet</b> .....	4
1. Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité	
2. Développer la partie back-end d'une application web et web mobile en intégrant les recommandations de sécurité	
<b>III. Résumé du projet</b> .....	4
<b>IV. Remerciement</b> .....	5
<b>V. Environnement humain et technique</b> .....	6
<b>VI. Cahier des charges, expression des besoins et spécifications fonctionnelles</b> .....	7
1. Informations générales.....	7
2. Présentation du projet.....	8
3. Systèmes de gestion.....	9
4. Lien avec la stratégie.....	10
5. Budget.....	11
6. Délai et contraintes de planning.....	11
7. Risques.....	11
8. Commentaire interne.....	12
9. Spécifications fonctionnelles.....	12
<b>VII. Les spécifications technique du projet développés</b> .....	14
1. Projet Studee : Environnement de Développement et Sécurité.....	14
2. Partie Back-end.....	15
3. Partie Front-end.....	16
4. Installation et configuration.....	16
5. Déploiement et Test.....	20
<b>VIII. La réalisation du candidat comportant les extraits de code les plus significatifs et la justification de ces choix</b> .....	20
1. Modèle de données (Entity) : StudentProfile.....	20
2. DTO (Data Transfer Object) : SaveStudentProfileDto.....	22
3. Repository : StudentProfileRepository.....	23
4. Cas d'utilisation (Use Case) : SaveStudentProfileUseCase.....	24
5. Resolver : StudentProfileResolver.....	26
<b>IX. Présentation du jeu d'essai élaboré de la fonctionnalité la plus représentative</b> .....	27
<b>X. La description de la veille effectuée par le candidat sur les vulnérabilités de sécurité</b> .....	29
1. Outils et Méthodologie de Veille : Utilisation de Pearltrees.....	29
<b>XI. Situation de travail nécessitant une recherche via des sites anglophones</b> .....	30
<b>XII. Conclusion</b> .....	32
<b>XIII. Annexes</b> .....	33



## I. Introduction

Pour le projet personnel du titre professionnel de **Développeur Web et Web Mobile (DWWM)**, j'ai choisi de présenter le projet **Studee**, réalisé durant mon stage chez **HDM Network**. Ce projet, conçu pour faciliter le processus de recrutement en créant une plateforme qui met en relation étudiants et opportunités professionnelles, m'a permis de développer une application web mobile complète, en intégrant des technologies modernes telles que **React Native**, **Nest.js**, **GraphQL**, et **TypeScript**.

Ce projet a été particulièrement stimulant, car j'ai dû faire face à la difficulté de trouver un stage, lors de la formation.

Ce dossier détaillera comment le projet **Studee** répond aux compétences exigées par le référentiel, notamment :

- **Le développement de la partie front-end** d'une application web ou mobile, en suivant les recommandations de sécurité.
- **Le développement de la partie back-end**, en intégrant également les meilleures pratiques de sécurité.

Vous y trouverez une présentation des compétences couvertes, le cahier des charges, ainsi que les spécifications fonctionnelles et techniques du projet. Des extraits de code significatifs seront analysés pour justifier mes choix techniques. De plus, un jeu d'essai sera présenté pour démontrer la validité d'une des fonctionnalités développées.

Le dossier inclut aussi une réflexion sur la veille technologique, axée sur les vulnérabilités de sécurité, et un exemple de recherche effectuée via des sites anglophones, avec traduction.

Ce projet m'a offert une occasion unique d'approfondir mes compétences techniques et d'acquérir une expérience précieuse dans le développement d'applications web sécurisées et performantes.



## II. Compétences du référentiel couvertes par le projet

### II.1 Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité

Maquetter une application

Réaliser une interface utilisateur web statique et adaptable

Développer une interface utilisateur web dynamique

### II.2 Développer la partie back-end d'une application web et web mobile en intégrant les recommandations de sécurité

Créer une base de données

Développer les composants d'accès aux données

Développer la partie back-end d'une application web ou web mobile

## III. Résumé du projet

Le projet Studee offrant une expérience utilisateur intuitive et performante, a été réalisé dans le cadre de mon stage chez HDM NetWork au sein d'une équipe de développement et est toujours en cours de développement.

### Technologies Utilisées, Outils et Méthodes:

- Frontend : React Native et TypeScript
- Backend : Nest.js, GraphQL, et TypeScript
- Base de données : Prisma pour la création, modélisation et administration et MySQL Workbench pour des opérations spécifiques.
- Déploiement : Docker Desktop pour une plateforme stable et sécurisée
- Environnements de développement : JetBrains Toolbox pour WebStorm et Android Studio, Docker et MySQL en local
- Design UI : Miro et Figma pour une visualisation et une collaboration efficace
- Gestion de code source : GitHub pour faciliter la collaboration et le suivi des modifications
- Gestion de projet : Jira pour le suivi des tâches et la coordination des équipes

### Objectif Principal:

L'objectif principal de Studee est de simplifier le processus de recrutement :

- Aider les entreprises à trouver des candidats qualifiés



➤ Offrir aux étudiants un accès facile à des opportunités professionnelles correspondant à leurs compétences et intérêts

➤ Améliorer l'efficacité du marché de l'emploi étudiant

➤ Créer un pont solide entre le monde académique et le secteur professionnel.

#### **Fonctionnalités Clés:**

➤ Enregistrement et gestion des offres d'emploi par les entreprises

➤ Accès et filtration des offres par les étudiants

➤ Possibilité de postuler directement via l'application

## **IV. Remerciement**

Dans le cadre de la réalisation de ce dossier, je souhaite prendre un moment pour exprimer ma profonde gratitude à toutes les personnes qui ont joué un rôle clé dans cette aventure. Sans leur soutien, leur guidance et leur bienveillance, ce parcours n'aurait pas été le même.

Tout d'abord, je tiens à remercier du fond du cœur mes parents. Ils ont fait de moi la personne que je suis aujourd'hui. Leur soutien inconditionnel et leur amour m'ont donné la force et le courage de poursuivre mes rêves, même dans les moments les plus difficiles.

Je souhaite également m'adresser à moi-même, car il est essentiel de reconnaître la force intérieure qui m'a permis de surmonter les défis et de choisir un chemin aussi exigeant que celui du développement web. Ce parcours a été semé d'obstacles, mais ô combien enrichissant.

Un grand merci à mon école, ESECAD, pour son accompagnement et son soutien tout au long de ma formation. Je tiens particulièrement à exprimer ma reconnaissance envers mon coach, Stefann, dont l'encouragement et les conseils ont été des piliers essentiels dans mon évolution.

Je ne saurais exprimer assez ma gratitude envers Zakaria, développeur chez HDM, qui m'a offert une opportunité inestimable d'intégrer son équipe. Grâce à sa confiance et à son accompagnement bienveillant, j'ai pu franchir une étape décisive dans ma carrière.

Je remercie aussi chaleureusement toute l'équipe avec laquelle j'ai eu le plaisir de collaborer chez HDM, notamment Zakaria, team leader du pôle développeur web, ainsi qu'Alexandre et Yankhoba, membres de la team Studee. Leur esprit d'équipe, leur engagement et leur savoir-faire ont enrichi mes compétences et élargi mes horizons.

Enfin, je tiens à adresser une mention spéciale à Quentin, un mentor exceptionnel, dont les conseils avisés, la patience et le soutien constant ont été déterminants dans mon développement tant professionnel que personnel. Son accompagnement m'a aidé à surmonter des obstacles importants et à grandir tout au long de ce parcours.



En conclusion, ce projet n'aurait jamais pu être aussi profitable sans le soutien et l'influence positive de toutes ces personnes. À chacun de vous, un immense merci. Votre impact sur mon parcours est inestimable, et je me sens infiniment reconnaissant pour tout ce que vous m'avez apporté.

## **V. Environnement humain et technique**

Mon intégration chez HDM Network ASBL a été soigneusement planifiée pour garantir une transition fluide dans l'équipe. Dès le début, j'ai reçu un accueil chaleureux et des documents détaillés m'ont été fournis pour me guider tout au long de mon stage. La Welcome Day a été une occasion précieuse pour me familiariser avec l'entreprise, rencontrer les membres de l'équipe et comprendre la culture organisationnelle. Cette première journée m'a permis de me sentir immédiatement inclus et soutenu.

Comme l'entreprise travaille principalement avec ReactJs, React Native, Nest.js et GraphQL, j'ai bénéficié de deux semaines de formation intensive pour acquérir les compétences nécessaires avant de rejoindre officiellement le projet Studee. Cette période de formation m'a permis de me sentir soutenu et préparé pour intégrer pleinement l'équipe. J'ai donc rejoint une équipe dynamique déjà composée de trois stagiaires concepteurs développeurs web et web mobile qui achevaient leur stage et m'ont éclairé sur ce qui avait été fait jusqu'à présent et ce qui restait à faire. Ces derniers avaient réalisé des étapes préliminaires importantes telles que l'étude de marché, la rédaction du cahier des charges et la création des premières maquettes sur Paint, Miro et Figma.

J'ai été chargé de prendre en main le développement de l'interface étudiant, couvrant à la fois le frontend et le backend pour répondre aux besoins spécifiques des utilisateurs étudiants. Alexandre m'a rejoint pour prendre en charge la partie entreprise du projet et Yankhoba pour s'occuper du chat. J'ai rapidement réalisé l'importance de la collaboration et de l'échange au sein de l'équipe, sous la supervision de Quentin, notre développeur senior. Nous avons utilisé des outils tels que Jira pour la gestion de nos tâches, organisées en sprints de deux semaines selon la méthode agile, et Gather avec l'équipe Studee pour maintenir une communication efficace malgré le télétravail. Les réunions quotidiennes (Daily scrum) et les points de contrôle à chaque phase du sprint ont renforcé notre coordination et notre capacité à répondre rapidement aux défis techniques.

Cette interaction constante a favorisé un environnement de travail dynamique et stimulant, où les idées étaient échangées librement et où les problèmes étaient résolus collectivement. J'ai pu pleinement m'immerger dans le projet Studee et contribuer de manière significative à son développement.



## VI. Cahier des charges, expression des besoins et spécifications fonctionnelles

Pour rappel, Le cahier des charges de l'application mobile Studee était déjà complet à mon arrivée en stage.

### VI.1 Informations générales

Le nom du projet a été soigneusement choisi après une analyse approfondie des applications similaires sur le marché, dans le but de se démarquer de la concurrence. Après délibération, le nom retenu pour l'application est "Studee".

Notons que le client à l'origine de ce projet est également co-propriétaire de l'entreprise sous laquelle l'application sera développée, à savoir "HDM Network ASBL".

Monsieur Quentin Mousset, en plus de son rôle de co-propriétaire, assurera également le rôle de responsable côté client pour ce projet.

### VI.2 Présentation du projet

#### a. Contexte et objectif

L'entreprise souhaitait créer une application mobile de mise en relation entre étudiants et particuliers. Elle voulait se démarquer en ne proposant que des stages et des alternances, ainsi qu'en proposant des campagnes de push notifications aux entreprises. L'entreprise visait à résoudre un problème bien spécifique qui était la difficulté pour les entreprises de trouver des stagiaires et des alternants, et la difficulté pour les stagiaires et alternants de trouver une entreprise, en centralisant les recherches sur une application ne proposant que ce type de contrat sans être inondé par des offres d'autres types (CDI, CDD, intérim...), résolvant ainsi la « convergence des besoins » des acteurs cibles. L'objectif principal était de permettre aux entreprises d'exposer leurs offres de stage et d'alternance, ainsi que d'offrir une plateforme mobile aux étudiants recherchant des stages ou des alternances.

Un objectif secondaire était de monétiser les campagnes de push notifications, et de faire connaître l'application, permettant ainsi de valoriser le nom de l'entreprise HDM.



## b. Description

Lors de la première discussion avec le client, des besoins ont été exprimés verbalement :

- Concept similaire aux sites student.be/studl/Indeed.
- Inscription sur des canaux spéciaux pour être notifié précisément (campagne de push notification) et cibler des profils.
- Création d'une fiche entreprise avec des critères précis et détaillés, tout en maintenant la simplicité.
- Objectif d'une interface conviviale pour l'utilisateur.
- Ajout d'un chat in-app pour permettre la communication entre utilisateurs.

Pour clarifier ces besoins, plusieurs questions ont été posées lors des échanges avec le client :

- Sur le concept similaire aux sites student.be/studl/Indeed : clarification sur la charte graphique, le nombre et les noms des canaux de notification, et les critères de ciblage des profils.
- Concernant la fiche entreprise : spécification des critères, existence d'une fiche étudiant, et définition de la convivialité utilisateur.
- Pour l'ajout d'un chat in-app : précision sur les types de messages possibles et les interactions entre utilisateurs.

Ces échanges ont permis de définir un modèle de spécifications pour guider le développement du projet.

## c. Cibles

Ce projet vise principalement les entreprises à la recherche de stagiaires et/ou d'alternants, ainsi que les étudiants en quête d'opportunités de stage ou d'alternance. Les études internes ont révélé que ce type d'application attire un public dynamique et demandeur, en raison de l'augmentation des programmes de formation intégrant des périodes en entreprise et de la nécessité croissante des entreprises de recruter des stagiaires et des alternants. En tant qu'acteurs du développement de cette application lors de ce stage, l'équipe a été directement confrontés aux défis et aux besoins rencontrés par le public cible. Cette expérience a permis d'analyser les problématiques non seulement en tant que développeurs, mais aussi en tant que représentants du public visé.



#### **d. Critères de succès**

Les critères de succès incluent le développement d'une application sécurisée et robuste, prête à être déployée sur l'App Store et le Google Play Store.

De plus, l'objectif est d'atteindre une participation mensuelle d'environ une centaine de comptes actifs, accompagnée d'une note minimale de 3/5. (cf document Veille concurrentiel en Annexe).

#### **e. Volumes à traiter**

Les étudiants et les entreprises, équipés de mobiles et ayant une bonne maîtrise de leur utilisation, sont capables de gérer un volume important de données et de tâches via l'application. Grâce à l'accès à un réseau sans fil, ils peuvent utiliser l'application à tout moment et depuis divers endroits, ce qui facilite la gestion d'un grand nombre de transactions et d'interactions en temps réel.

#### **f. Rôles et degrés de liberté**

La mission consiste à concevoir l'application. Toutes les propositions seront soumises à Monsieur Quentin Mousset, co-propriétaire de HDM Network ASBL. Le client a expressément demandé un niveau de détail et de précision élevé dans la réalisation, exigeant un rendu parfait jusqu'au moindre détail. Bien que nous ayons une certaine liberté pour intégrer de nouvelles fonctionnalités, il est impératif que toutes celles préalablement définies avec le client soient implémentées intégralement.

### **VI.3 Systèmes de gestion**

#### **a. Système de gestion des entreprises (SGE)**

L'application doit être capable de collecter des informations sur les entreprises partenaires, telles que leur nom, leur secteur d'activité, leurs offres de stages et d'alternances disponibles, etc. Cette tâche est sous la responsabilité de l'équipe de développement chargée de la synchronisation des données des entreprises.

#### **b. Systèmes de gestion des stages et entreprise (SGSE)**

Afin de garantir la cohérence des informations, l'application doit intégrer les données concernant les stages et les alternances proposés par les entreprises, y compris les descriptions de poste, les critères de sélection, etc. Cette responsabilité incombe à l'équipe de développement en charge de la gestion des offres de stages et d'alternances.



### **c. Système de gestion des utilisateurs (SGU)**

Afin de permettre l'authentification des étudiants et des représentants d'entreprise, l'application doit interagir avec un système de gestion des utilisateurs pour vérifier les identités et les autorisations. Cette responsabilité incombe à l'équipe de développement chargée de la sécurité et de l'authentification.

### **d. Structure des données**

Les données pertinentes pour l'application incluent les profils des étudiants et des entreprises, les offres de stages et d'alternances, les messages échangés entre les utilisateurs, etc. Ces données sont structurées de manière à faciliter la recherche et la consultation pour les utilisateurs de l'application.

### **e. Intégration dans l'environnement**

L'application s'intègre harmonieusement dans l'environnement des utilisateurs en tant qu'outil de recherche et de candidature pour les stages et les alternances.

Elle propose une interface conviviale pour les étudiants et les entreprises, simplifiant ainsi la navigation et l'utilisation de ses fonctionnalités.

### **f. Authentification des utilisateurs**

Les étudiants et les représentants d'entreprise s'identifient à l'aide d'identifiants uniques tels que des adresses e-mail et des mots de passe, ou d'autres méthodes d'authentification sécurisées comme l'authentification via Google ou Facebook. Cette procédure d'authentification assure la sécurité et la confidentialité des données des utilisateurs.

### **g. Autres outils utilisés par l'utilisateur**

Outre l'application mobile, les étudiants et les entreprises ont la possibilité d'utiliser divers autres outils de recherche de stages et d'alternances, tels que les sites web spécialisés et les réseaux sociaux professionnels.

Il est essentiel que l'application se démarque en offrant une proposition de valeur unique par rapport à ces autres outils. Elle doit fournir une expérience utilisateur optimale et des fonctionnalités différenciatrices, en se concentrant exclusivement sur la recherche d'offres de stages et d'alternances, sans inclure d'autres types de contrat.

## **VI.4 Lien avec la stratégie**

Ce projet se distingue des autres applications similaires en se concentrant exclusivement sur les stages et les alternances, ce qui nous permet de cibler un public spécifique. Notre stratégie est



de devenir la plateforme leader dans ce secteur. Le succès de ce projet nous offrira l'opportunité de promouvoir HDM Network ASBL et d'attirer des partenaires et collaborateurs potentiels.

## VI.5 Budget

M. Quentin Mousset, co-owner et initiateur du projet, également notre maître de stage, n'a pas encore défini de budget.

## VI.6 Délai et contraintes de planning

Aucun délai spécifique n'a été fixé pour ce projet. Étant une association à but non lucratif, HDM Network ASBL, dirigée par les co-owners "Quentin Mousset" et "David VON GOISTOVEN, compte principalement sur des stagiaires pour la réalisation de ses projets. Les délais peuvent varier en fonction du recrutement et des compétences des stagiaires affectés au projet "Studee".

## VI.7 Risques

Risque	Description	Probabilité	Gravité	Solution/Plan d'action
Problèmes de sécurité des données	Fuite de données personnelles des utilisateurs, violation de la vie privée, attaque des pirates informatiques	4/5	5/5	Renforcer les mesures de sécurité, cryptage des données, audits de sécurité réguliers.
Baisse de la demande pour l'application	Manque d'intérêt des étudiants ou des entreprises, évolution du marché, compétition accrue.	3/5	4/5	Étude de marché continue, adaptation rapide aux changements, campagnes de marketing ciblées.
Problèmes de compatibilité avec les appareils	Dysfonctionnements sur certains types d'appareils mobiles, problèmes de performance, bugs	2/5	3/5	Test de compatibilité approfondis sur une large gamme d'appareils, mises à jour régulières.
Difficultés de recrutement pour les entreprises	Manque d'adhésion des entreprises au concept, résistance au changement, complexité des intégrations.	3/5	4/5	Campagne de sensibilisation auprès des entreprises, offres incitatives, soutien technique dédié.
Défaillance du système de gestion des utilisateurs	Problèmes d'authentification, accès non autorisé, perte d'informations utilisateurs.	2/5	4/5	Mise en place de protocoles de sécurité robustes, sauvegardes régulières des données utilisateurs.



## VI.8 Commentaire interne

Le client est un professionnel du développement informatique, également maître de stage des développeurs affectés à ce projet. Il attache une grande importance aux détails visuels et à la cohérence graphique, exigeant un rendu parfait après validation.

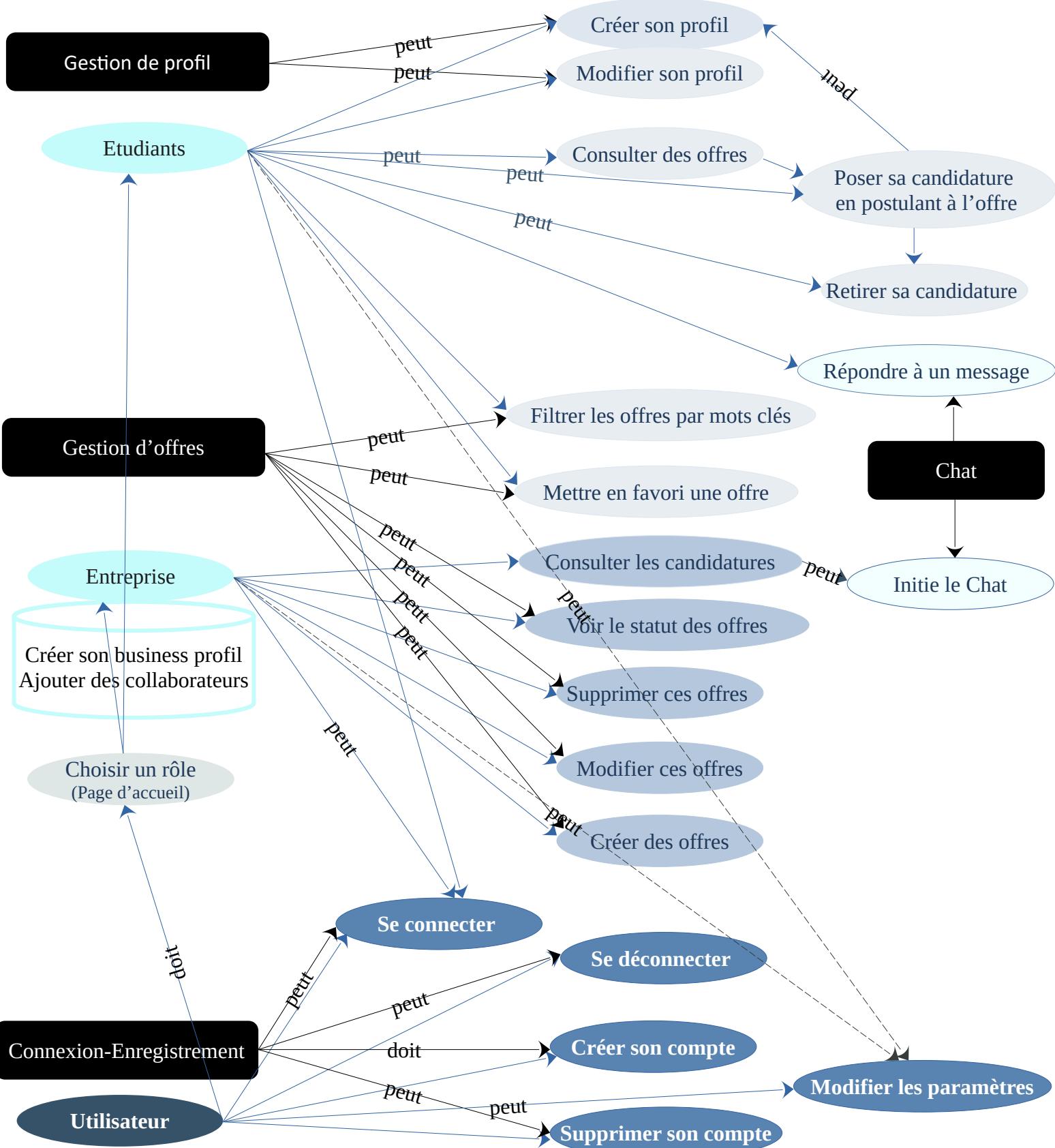
Le choix des technologies utilisées pour le développement de "Studee" a été effectué par le client lui-même. Aucun délai ni budget spécifique n'ont été définis par le client, préférant gérer ces aspects en interne étant donné que le projet est développé au sein de sa propre entreprise.

## VI.9 Spécifications fonctionnelles

Section	Cas d'utilisation	Acteur	Étapes principales
A. Pages d'accueil	Choix du rôle (Société ou Étudiant)	Utilisateur	<ol style="list-style-type: none"><li>Arrivée sur la page d'accueil.</li><li>Selection du rôle.</li><li>Redirection vers la page de login.</li></ol>
	Page de Login	Utilisateur	<ol style="list-style-type: none"><li>Saisie des identifiants.</li><li>Connexion ou création de compte.</li><li>Redirection vers l'interface spécifique.</li></ol>
	Création de compte	Étudiant ou Entreprise	<ol style="list-style-type: none"><li>Accès à la page de création de compte.</li><li>Saisie des informations personnelles.</li><li>Validation et création du compte.</li></ol>
B. Section Étudiant	Accueil étudiant	Étudiant	<ol style="list-style-type: none"><li>Redirection après connexion.</li><li>Affichage des offres, profils, paramètres.</li></ol>
	Consultation des offres	Étudiant	<ol style="list-style-type: none"><li>Recherche et filtre des offres.</li><li>Visualisation des détails.</li><li>Candidature ou sauvegarde.</li><li>Répondre aux messages des entreprises.</li></ol>
	CompléTION du Profil	Étudiant	<ol style="list-style-type: none"><li>Accès au profil.</li><li>Création du profil étudiant.</li><li>Validation et mise à jour.</li></ol>
C. Section Société	Création de Fiche d'Entreprise	Entreprise	<ol style="list-style-type: none"><li>Saisie des informations de base.</li><li>Ajout des réseaux sociaux.</li><li>Ajout des collaborateurs.</li></ol>
	Gestion des Offres d'Emploi	Entreprise	<ol style="list-style-type: none"><li>Création/modification/clôture des offres.</li><li>Consultation des candidatures.</li><li>Communication avec les candidats (Chat)</li></ol>
D. Paramètres	Gestion des Paramètres de Compte	Utilisateur	<ol style="list-style-type: none"><li>Modification du thème et des notifications.</li><li>Consultation des CGU et politique de confidentialité.</li><li>Gestion des informations de compte et déconnexion.</li></ol>



Le diagramme suivant présente les spécifications fonctionnelles entre les utilisateurs.





Cas d'utilisation	Consultation et Candidature à une Offre de stage	
Acteurs	Étudiant connecté	
Précondition	L'étudiant visualise la liste des offres de stage.	
Postcondition	Candidature soumise et profil étudiant mis à jour.	
Étapes principales	Description	
1. Sélection d'une Offre	L'étudiant sélectionne une offre dans la liste.	
2. Affichage des Détails	Les détails de l'offre (description, compétences, localisation) sont affichés.	
3. Candidature	L'étudiant clique sur "Poser sa candidature". Une modale s'ouvre avec deux options : sélectionner un profil étudiant existant ou en créer un afin de postuler.	
3. Candidature (suite)	<b>Option 1 : Profil existant :</b> L'étudiant sélectionne son profil et clique sur "Je postule". La candidature est enregistrée.	<b>Option 2 : Création de Profil :</b> L'étudiant est redirigé vers la page de création de profil. Après avoir complété les informations (description, expérience, formations, réseaux sociaux, téléchargement du CV, etc.), il valide le profil. L'étudiant est redirigé vers l'offre, sélectionne le profil créé, et postule.
4. Redirection	L'étudiant est redirigé vers la page d'accueil avec sa candidature en cours visible.	
Extensions	Description	
Retour à la Liste d'Offres	L'étudiant peut revenir à la liste des offres sans postuler.	
Échec de la Candidature	Message d'erreur en cas d'échec de la soumission (problème système/connectivité), possibilité de réessayer plus tard.	

## VII. Les spécifications technique du projet développés

### VII.1 Projet Studee : Environnement de Développement et Sécurité

Le projet **Studee** nécessite un environnement de développement bien configuré pour maximiser l'efficacité et intégrer des mesures de sécurité à chaque étape du processus de développement. Voici les principaux outils utilisés :

- **JetBrains Toolbox** pour **WebStorm** : Développement front-end.
- **Android Studio** : Lancement de l'application
- **Docker Desktop** : Conteneurisation
- **MySQL Workbench** : Gestion de la base de données.
- **Expo Go** : Lancement de l'application pour une visualisation en temps réel des modifications.
- **Jira** : Gestion des tâches, création de branches, et suivi des collaborations au sein de l'équipe.



## VII.2 Partie Back-end

La partie back-end de Studee est construite autour de technologies robustes pour assurer la sécurité et la maintenabilité du projet :

- **Nest.js** : Framework côté serveur basé sur Node.js, choisi pour sa modularité et son extensibilité. Il permet l'implémentation de pratiques de sécurité telles que le contrôle d'accès aux services.
- **TypeScript** : est un vérificateur statique de types du code. Compatible avec Nest.js, il détecte les erreurs avant la compilation en s'assurant que les types sont cohérents et corrects, réduisant ainsi les risques de vulnérabilités.
- **GraphQL** : Utilisé comme langage de requête pour interagir avec les services back-end. L'intégration avec TypeScript permet une validation stricte des schémas de requêtes et de mutations, contribuant à la sécurité des interactions avec les données.
- **GraphQL Playground** : Outil intégré pour tester les requêtes dans un environnement interactif. En production, cet accès est restreint pour éviter tout risque d'exposition involontaire des schémas et données sensibles.
- **Prisma** : Outil de gestion de base de données. En combinaison avec TypeScript, Prisma génère automatiquement des types basés sur le schéma de la base de données, garantissant une interaction cohérente et sécurisée avec les données. Lors de la **compilation**, Prisma utilise ce schéma pour vérifier les types et détecter les erreurs avant l'exécution. Cela améliore la sécurité des types, facilite l'autocomplétion dans les IDE, et aide à détecter les erreurs tôt dans le processus de développement.





### VII.3 Partie Front-end

Le front-end de Studee est développé pour assurer à la fois la sécurité et une expérience utilisateur fluide :

- **React Native** : Framework utilisé pour le développement d'applications mobiles multiplateformes.
- **TypeScript** : Employé pour accroître la robustesse du code, réduire les erreurs potentielles et assurer la cohérence des types à travers l'ensemble du projet.
- **IU** : conçue de manière intuitive afin d' offrir une expérience fluide aux utilisateurs finaux, avec des outils de conception tels que Miro et Figma utilisés pour la conception des maquettes et des designs.
- **CSS** : appliqué de manière dynamique dans le code React Native.



### VII.4 Installation et configuration

- Récupération des dépôts  
Téléchargez les dépôts **GitHub** pour l'API et l'application mobile.
- Environnement de Développement  
**Installation des Dépendances** : Utilisez yarn pour installer toutes les dépendances nécessaires.



**Configuration des Variables d'Environnement** : Modifiez les fichiers .env pour chaque composant du projet, en définissant les variables d'environnement requises

- **Configuration de la Base de Données**

**Utilisation de Docker** : Créez un conteneur MySQL avec Docker pour isoler l'environnement de la base de données et garantir sa portabilité.

**Installation de la Base de Données** : Configurez la base de données Studeedb dans MySQL Workbench. Cela permet aux développeurs d'accéder et de manipuler les données.

(Consultez le document « *Install\_studee* » fait par mes soins en Annexe pour plus de détails.)



```
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "mysql"
  url      = env("DATABASE_URL")
}

model User {
  id      Int      @id @default(autoincrement())
  email   String   @unique @db.VarChar(255)
  password String   @db.VarChar(255)
  isBusiness Boolean
  hasBusiness Boolean @default(false) // Added to check if user already have created a business
  createdAt DateTime @default(now())
  updatedAt DateTime @default(now()) @updatedAt
  profile   Profile? // utilisateur peut exister sans etre lié a un profil
}

model Profile {
  id      Int      @id @default(autoincrement())
  firstName String  @db.VarChar(50)
  lastName String  @db.VarChar(50)
  city    String  @db.VarChar(50)
  createdAt DateTime @default(now())
  updatedAt DateTime @default(now()) @updatedAt

  user      User      @relation(fields: [id], references: [id], onDelete: Cascade) // chaque
  instance de Profile est associée à une instance correspondante de User via l'ID.
  // Si user Supprimer Son Profile est supprimer aussi
  receivedMessages Message[] @relation("Receiver") // Relation inverse pour le champ receiver
  sentMessages   Message[] @relation("Sender") // Relation inverse pour le champ sender
  studentProfile StudentProfile[] // un Profile peut ou non être lié à plusieurs instance de StudentProfile.
  businessProfile BusinessProfile? // un Profile peut être lié ou non à une instance de BuisnessProfile.
  activityDomains ActivityDomain[] // Relation plusieurs à plusieurs avec ActivityDomain
```

Figure1 : Exemple d'une partie du schéma prisma/§

Commande effectuées afin de générer la base de donnée Studee :

➤ npx prisma migrate dev

```
-- CreateTable
CREATE TABLE `User` (
  `id` INTEGER NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(255) NOT NULL,
  `password` VARCHAR(255) NOT NULL,
  `isBusiness` BOOLEAN NOT NULL,
  `hasBusiness` BOOLEAN NOT NULL DEFAULT false,
  `createdAt` DATETIME(3) NOT NULL DEFAULT CURRENT_TIMESTAMP(3),
  `updatedAt` DATETIME(3) NOT NULL DEFAULT CURRENT_TIMESTAMP(3),

  UNIQUE INDEX `User_email_key`(`email`),
  PRIMARY KEY (`id`)
) DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

Figure 2: Exemple d'une partie du fichier de migration



➤ npx prisma generate dev

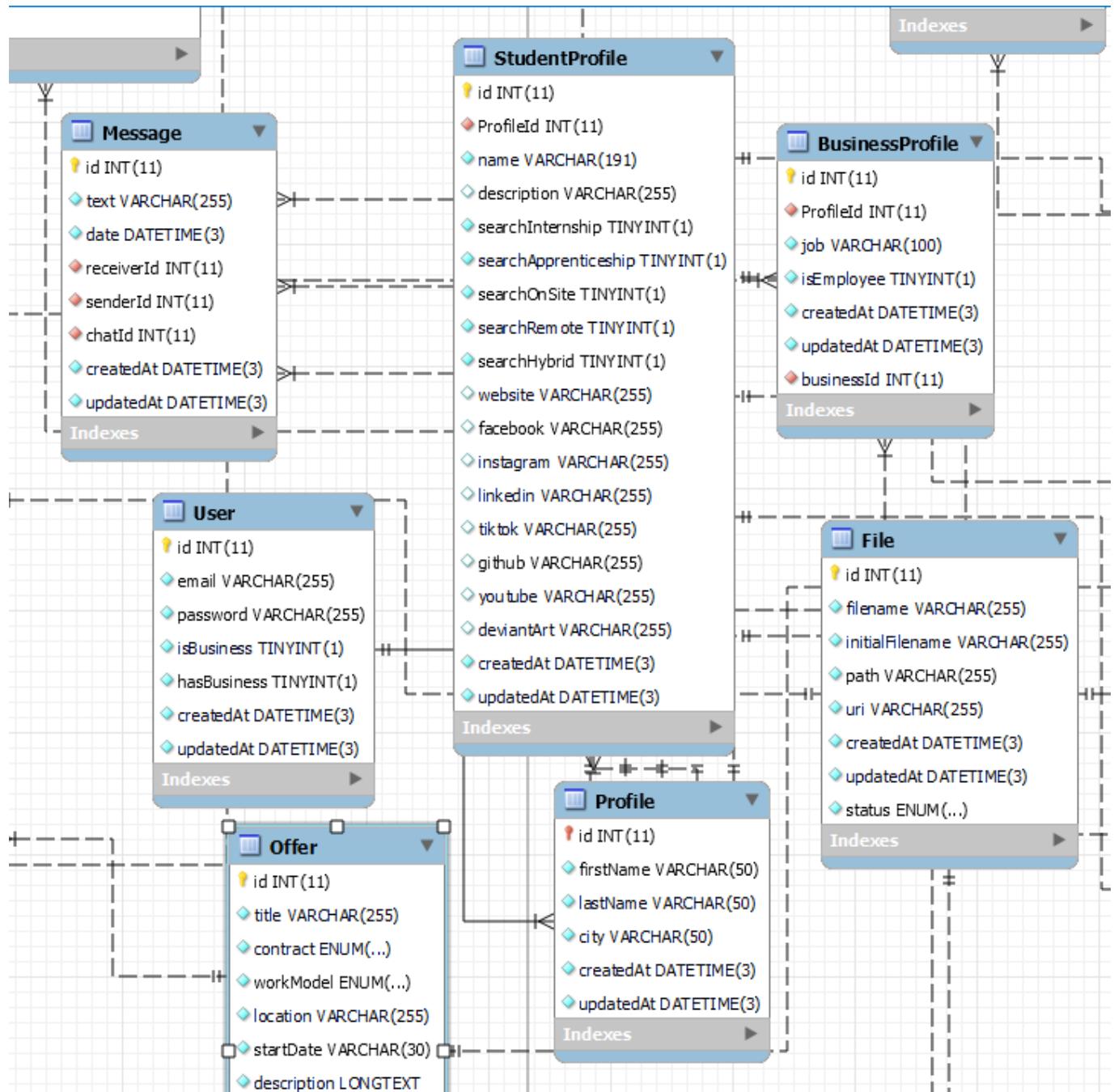


Figure 3 : Exemple d'une partie de Database Mysql Workbench Studee

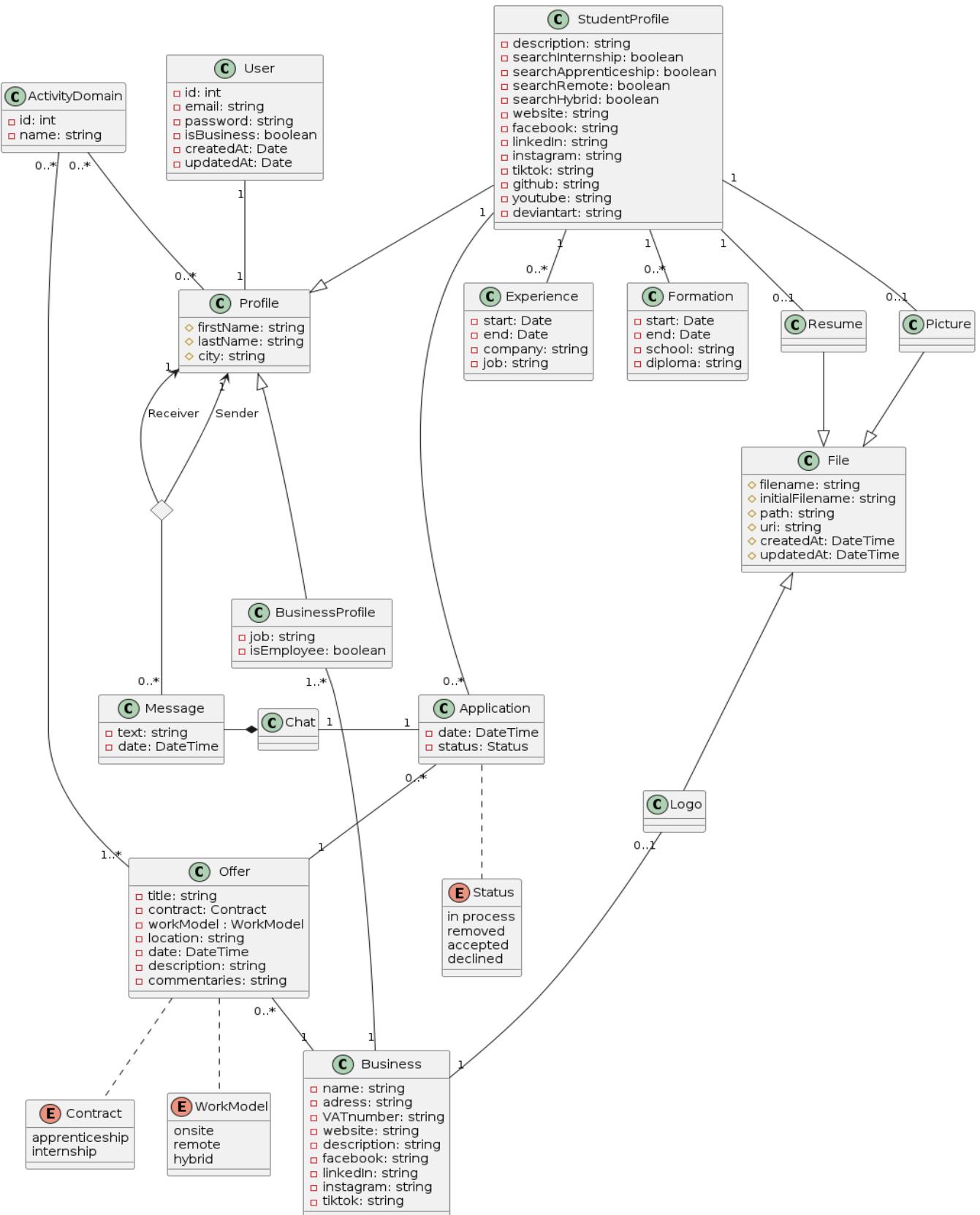


Figure 4 : Modèle conceptuel de données Studee



## VII.5 Déploiement et Test

### ➤ Lancement local

**Démarrage de l'Application** : Une fois l'installation et la configuration terminées, je lance l'application localement pour le développement et les tests.

### ➤ Utilisation des Outils de Développement

**Metro Bundler** : J'utilise Metro Bundler pour le front-end afin de voir les modifications en temps réel.

**Serveur Back-End** : Je démarre le serveur back-end pour vérifier l'intégration et le fonctionnement global de l'application.

**Expo Go ou Android Studio** : Je choisis entre Expo Go et Android Studio pour le développement et le test des applications React Native :

- **Expo Go** : Offre un prototypage rapide et une prévisualisation immédiate des changements sur divers appareils.
- **Android Studio** : Fournit des outils avancés pour tester l'application sur des emulateurs Android ou des appareils physiques, avec des fonctionnalités de débogage approfondies.

### ➤ Débogage et Vérification

**Débogueur** : J'utilise le débogueur intégré d'Expo Go ou les outils de débogage d'Android Studio pour identifier et résoudre rapidement les problèmes.

**Identification des Problèmes** : Je vérifie le bon fonctionnement de l'application et détecte tout problème potentiel à l'aide des outils de débogage disponibles.

### ➤ Mesures de sécurité prévues lors du déploiement

Au moment du déploiement sur l'App Store et Google Play, des mesures de sécurité spécifiques seront mises en place, telles que :

- **Chiffrement des données sensibles** .
- **Sécurisation des communications réseau (HTTPS)** .
- **Gestion des autorisations d'accès** pour protéger les données des utilisateurs.

## VIII. La réalisation comportant les extraits de code les plus significatifs et la justification de ces choix

La réalisation de la partie principale de mon projet Studee se concentre sur la création et la gestion des profils étudiants. Cette section présente les composants les plus significatifs associés à cette tâche, en mettant l'accent sur la robustesse, la flexibilité, la maintenabilité, ainsi que sur la sécurité du système.



## VIII.1 Modèle de données (Entity) : StudentProfile

```
— □ ×

@ObjectType()
export default class StudentProfile {
  @Field(() => Int)
  id: number;

  @Field()
  name: string;

  @Field({ nullable: true })
  description: string | null;

  @Field()
  searchInternship: boolean; ...

  @Field({ nullable: true })
  website: string | null; ...

  @Field(() => [Experiences])
  experiences?: Experiences[];

  @Field(() => [Formations])
  formations?: Formations[];

  @Field(() => Picture, { nullable: true })
  picture?: Picture | null;

  @Field(() => Resume, { nullable: true })
  resume?: Resume | null;

  @Field(() => [Application])
  application?: Application[];

  @Field(() => Int)
  ProfileId: number;

  context?: ContextualGraphqlRequest;
}
```

```
— □ ×

export type
ContextualGraphqlRequest = {
  userId?: number;
  email?: string;
  roles?: string[];
  password?: string;
  isBusiness?: boolean;
  args?: Record<string, any>;
  password?: string;
  profil:{

  }
};
```

Figure 6: ContextualGraphqlRequest

Figure 5: Entity StudentProfile

### a. Description

Le modèle StudentProfile définit les caractéristiques d'un profil étudiant, telles que :

- La description personnelle, les préférences de recherche d'emploi, les liens sociaux
- Il est implémenté en TypeScript avec NestJS, tirant parti d'un typage fort et d'une intégration fluide avec GraphQL.

### b. Justification

- Définition de la structure

**@ObjectType()** : Ce décorateur permet de déclarer StudentProfile comme un type objet dans GraphQL. Il facilite la définition claire et concise des champs et de leurs types.



**@Field** : Chaque champ du profil étudiant est défini à l'aide de ce décorateur, spécifiant le type (par exemple, Int, String, Boolean) et la possibilité de nullabilité (nullable: true). Cela assure un typage rigoureux des données et une gestion adéquate des valeurs nulles.

**Types Associés** : Les types Experiences et Formations, etc sont importés et associés à StudentProfile via les décorateurs @Field, permettant de définir les relations entre les entités et de récupérer les données associées lors des requêtes GraphQL.

#### ➤ Sécurité

**Validation des Données** : La définition stricte des types et la gestion de la nullabilité assurent que seules les données valides sont acceptées et stockées, réduisant les erreurs et renforçant la sécurité des opérations.

**Intégration du Contexte Utilisateur** : Le type **ContextualGraphqlRequest** encapsule des informations clés comme userId, roles, et isBusiness, permettant :

- **Identification et Contrôle** : De vérifier les priviléges et de limiter l'accès aux données sensibles en fonction de l'utilisateur.
- **Gestion de l'Accès** : D'assurer que seules les actions autorisées par le rôle et le statut de l'utilisateur sont exécutées.

**Relations avec d'autres Modèles** : L'attribut **ProfileId** permet des liens sécurisés avec le modèle **Profile**, facilitant une gestion contrôlée des accès et la récupération des informations utilisateur.

## VIII.2 DTO (Data Transfer Object) : SaveStudentProfileDto

### a. Fonctionnalités

```
– □ ×

@InputType()
export default class SaveStudentProfileDto {
  @Field(() => Int, { nullable: true })
  id?: number;

  @Field()
  name: string;

  @Field({ nullable: true })
  description: string | null;

  @Field({ nullable: true })
  searchInternship: boolean | null; ...

  @Field({ nullable: true })
  website: string | null; ...

  @Field(() => SaveResumePictureDto, { nullable: true })
  resume?: SaveResumePictureDto | null; ...

  @Field(() => Int)
  ProfileId: number;
}
```

Figure 7: Dto - SaveStudentProfileDto



Le **SaveStudentProfileDto** joue un rôle crucial dans la validation et la manipulation des données lors de la création ou de la mise à jour d'un profil étudiant. Il structure les données de manière cohérente et sécurisée lors de leur transfert entre le frontend et le backend.

### b. Justification

#### ➤ Structure des Données

Le **DTO** définit clairement les champs nécessaires pour créer ou mettre à jour un profil étudiant, comme name, description, et les préférences de recherche (searchInternship etc.).

Les décorateurs **@Field** spécifient les types de données pour chaque champ, ainsi que les contraintes telles que la nullabilité (nullable: true). Cette approche garantit une validation précoce des données, réduisant les erreurs lors du traitement.

#### ➤ Suivi et cohérence

Les champs **createdAt** et **updatedAt** permettent de suivre l'historique des modifications, assurant ainsi la traçabilité des données.

Le champ **ProfileId** établit un lien avec le profil utilisateur associé, garantissant la cohérence des informations entre les différents modèles de l'application.

#### ➤ Sécurité

**Validation des Données** : La validation stricte au niveau du DTO assure que seules les données conformes sont acceptées, réduisant les risques de faille de sécurité en empêchant des données non conformes ou malveillantes d'atteindre les couches internes de l'application.

## VIII.3 Repository : StudentProfileRepository

### a. Fonctionnalités

Le **StudentProfileRepository** gère l'accès aux données des profils étudiants dans la base de données en effectuant des opérations CRUD (Create, Read, Update, Delete).

```
@Injectable()
export default class StudentProfileRepository {
  constructor(private readonly prisma: PrismaService) {}
  async save(data: Prisma.XOR<Prisma.StudentProfileCreateInput, Prisma.StudentProfileUncheckedCreateInput> | Prisma.XOR<Prisma.StudentProfileUpdateInput, Prisma.StudentProfileUncheckedUpdateInput>) {
    if (!data.id) {
      return this.prisma.studentProfile.create({ data: data as Prisma.XOR<Prisma.StudentProfileCreateInput, Prisma.StudentProfileUncheckedCreateInput> });
    }

    return this.prisma.studentProfile.update({
      where: {
        id: data.id as number,
      },
      data: data as Prisma.XOR<Prisma.StudentProfileUpdateInput, Prisma.StudentProfileUncheckedUpdateInput>,
    });
  }
}
```

Figure 8 : StudentProfileRepository



## b. Justification

### ➤ Rôle du Repository

Le **StudentProfileRepository** agit comme un intermédiaire entre les services de l'application et la base de données. Il encapsule la logique d'accès aux données, simplifiant les interactions avec les profils étudiants, facilitant la gestion des opérations CRUD tout en garantissant la sécurité et l'intégrité des données.

### ➤ Méthodes Clés :

- **delete(id: number)**: Supprime un profil étudiant en utilisant son identifiant unique.
- **findById(id: number)**: Récupère un profil étudiant spécifique par son identifiant.
- **findAllByProfileId(ProfileId: number)**: Récupère tous les profils liés à un identifiant de profil utilisateur.
- **save(data: Prisma.XOR<...>)**: Crée ou met à jour un profil étudiant en fonction des données fournies.

### ➤ Sécurité

- **Gestion Sécurisée des Données** : En utilisant le service PrismaService, le repository assure une interaction sécurisée avec la base de données. Les requêtes SQL sont gérées efficacement, limitant ainsi les risques d'injection SQL ou d'autres vulnérabilités liées aux accès directs.
- **Contrôles d'Accès** : Les opérations CRUD sont encapsulées de manière à garantir que seuls les utilisateurs autorisés puissent effectuer des modifications, respectant ainsi les règles d'authentification et d'autorisation définies au niveau de l'application.

## VIII.4 Cas d'utilisation (Use Case) : SaveStudentProfileUseCase

### a. Fonctionnalités

Le SaveStudentProfileUseCase centralise la gestion des profils étudiants, que ce soit pour leur création ou mise à jour, en utilisant les données du SaveStudentProfileDto.

### b. Justification

### ➤ Modularité et Dépendances

- Structuré avec la **méthode UseCase**, il sépare les responsabilités pour un code clair et maintenable.



```
export type UseCase<Return, Parameters> = HandlerService<Return, [ContextualGraphqlRequest, ...Parameters]>
```

Figure 9: Méthode UseCase

- `@Injectable` permet d'injecter `StudentProfileRepository` et `RequestEventEmitter`, facilitant la gestion des dépendances et les tests unitaires.

```
@Injectable()
export default class SaveStudentProfileUseCase implements UseCase<Promise<StudentProfile>, [dto: SaveStudentProfileDto]> {
  constructor(
    private readonly studentProfileRepository: StudentProfileRepository,
    private readonly eventEmitter: RequestEventEmitter,
  ) {}
  async handle(context: ContextualGraphqlRequest, dto: SaveStudentProfileDto) : Promise<StudentProfile> {
    try {
      return { ...(await this.studentProfileRepository.save({
        ... (dto.id ? { id: dto.id } : {}),
        ...
        ProfileId: dto.ProfileId,
      })), context};
    } catch (error) {
      this.eventEmitter.emit('SaveStudentProfileUseCase::failed', { context: context, error: error.message });
      throw new BadRequestException(error.message);
    }
  }
}
```

Figure 10: UseCase: SaveStudentProfileUseCase

#### ➤ Gestion Dynamique des Données

- L'utilisation de `(dto.id ? {id: dto.id} : {})` simplifie la distinction entre création et mise à jour, rendant le code adaptable et flexible.

#### ➤ Sécurité et Gestion des Erreurs :

##### ➤ Contexte Utilisateur :

- Le `ContextualGraphqlRequest` encapsule les informations utilisateur (`userId`, `roles`, etc.) pour s'assurer que seules les opérations autorisées sont effectuées en fonction des droits de l'utilisateur.

##### ➤ Robustesse :

- Un **bloc try-catch** capture les erreurs, émet un événement d'échec via `RequestEventEmitter`, et lance une exception `BadRequestException` pour garantir la sécurité et une expérience utilisateur fluide.



## VIII.5 Resolver : StudentProfileResolver

### a. Fonctionnalité

Le **StudentProfileResolver** gère les requêtes GraphQL relatives aux profils étudiants, offrant des opérations CRUD modulaires pour les profils, formations, et expériences, etc.

```
@UseGuards(GraphqlAuthGuard)
  @Mutation(() => StudentProfile)
async saveStudentProfile(
  @ContextualRequest() context: ContextualGraphqlRequest,
  @Args('dto') dto: SaveStudentProfileDto,
): Promise<StudentProfile> {
  return (await
this.serviceFactory.create(SaveStudentProfileUseCase)).handle(context,dto);
}
```

Figure 11: Resolver: SaveStudentProfile

### b. Justification

#### ➤ Modularité

- Chaque méthode est clairement définie avec des décorateurs comme **@Mutation** ou **@Query**, assurant une séparation des responsabilités et facilitant la maintenance et l'évolution du code.

#### ➤ Gestion des Données :

- Les décorateurs **@Args** sont utilisés pour extraire les données des requêtes GraphQL, garantissant que chaque opération dispose des informations nécessaires.

#### ➤ Sécurité et Contexte Utilisateur :

##### ➤ Authentification et Autorisation :

- Le **@UseGuards(GraphqlAuthGuard)** protège les mutations et queries en s'assurant que seules les requêtes autorisées sont exécutées, renforçant ainsi la sécurité des opérations.

##### ➤ Contexte Utilisateur :

- Le contexte GraphQL, accessible via **@ContextualRequest**, encapsule des informations critiques comme userId, roles, et autres données utilisateur. Cela permet de valider les droits d'accès et de personnaliser les réponses en fonction de l'utilisateur authentifié.



## IX. Présentation du jeu d'essai élaboré de la fonctionnalité la plus représentative

Objectif	Vérifier que la méthode saveStudentProfile du StudentProfileResolver crée correctement un profil étudiant.
Jeu d'essai	Définition des données d'entrée sous la forme d'un objet SaveStudentProfileDto. Détermination des résultats attendus : création réussie d'un profil étudiant.
Configuration	Utilisation de Test.createTestingModule pour créer un module de test isolé. Importation de ApiModule. Inclusion des dépendances : StudentProfileResolver et SaveStudentProfileUseCase.
Exécution	Appel de la méthode saveStudentProfile du résolveur avec les données d'entrée définies.
Validation	Vérification des résultats obtenus par rapport aux résultats attendus en utilisant expect.

```
describe('StudentProfileResolver', () => {
  let resolver: StudentProfileResolver;
  // Mettre en place le module de test
  beforeEach(async () => {
    const module: TestingModule = await Test.createTestingModule({
      imports: [ApiModule]
    }).compile();
    resolver = module.get<StudentProfileResolver>(StudentProfileResolver);
  });
  // Test pour la fonction saveStudentProfile
  describe('saveStudentProfile', () => {
    it('should create a new student profile', async () => {
      // Données en entrée
      const testDto: SaveStudentProfileDto = {
        name: "dwwm", ...
        ProfileId: 3 // ID du profil (lié à un autre modèle)
      };
      // Contexte GraphQL simulé
      const mockContext: ContextualGraphqlRequest = {
        userId: 3, // ID de l'utilisateur authentifié
        profil:{}
      };
      const savedProfile = await resolver.saveStudentProfile(mockContext,
      testDto);
      // Vérification des résultats obtenus
      expect(savedProfile).toBeDefined();
    });
  });
});
```

Figure 12: Fichier test - StudentProfileResolver.spec.ts



```
Run: test
▶ C:\Users\Latifa\AppData\Roaming\npm\yarn.cmd test
🔧 yarn run v1.22.21
warning ..\..\..\package.json: No license field
$ jest
  PASS  src/Api/Profile/Test/studentProfileResolver.spec.ts (18.616 s)
    StudentProfileResolver
      saveStudentProfile
        ✓ should create a new student profile (678 ms)

    Test Suites: 1 passed, 1 total
    Tests:       1 passed, 1 total
    Snapshots:  0 total
    Time:        18.996 s, estimated 20 s
    Ran all test suites.
    Done in 22.00s.

Process finished with exit code 0
```

Figure 13: Résultat du test

Result Grid					Filter Rows:	Edit:	Export/Import:
	id	ProfileId	name	description			searchInternship
1	2	3	4	5	6	7	8

Figure 14: MySQL Workbench - Création du studentProfile via le test effectué



The screenshot shows the GraphQL playground interface with the following details:

- Query:** `query getOneStudentProfile($id: Float!){`
- Variables:** `getOneStudentProfile(id: $id){`
- HTTP Headers (1):** `Authorization: "Bearer evJhbGciOiJUzI1NiTnR5cCT6TkpoXVCJ9.evJhbWEpbCT6TmxhdGIAZ21haWwvY29tTiwidXN"`
- Result:** The result is a JSON object representing the student profile. It includes fields like `id`, `ProfileId`, `name`, `description`, `deviantArt`, `facebook`, `github`, `instagram`, `linkedin`, `searchApprenticeship`, `searchHybrid`, `searchInternship`, `searchOnSite`, `searchRemote`, `website`, `youtube`, `tiktok`, and `experiences`. The `experiences` field is a list containing one item with `id: 1`.

Figure 15: Playground-GetStudentProfile-dossier créé et modifié pour le jeu d'essai



## X. La description de la veille effectuée par le candidat sur les vulnérabilités de sécurité

### Citations Inspirantes

« Le combattant qui l'emporte est celui qui gagne la campagne de l'information. Nous en avons fait la démonstration au monde : l'information est la clef de la guerre moderne – stratégiquement, opérationnellement, tactiquement et techniquement. »

— Glen Otis

« La sécurité ne se mesure pas à l'absence de menaces, mais à la capacité à les détecter et à y répondre efficacement. »

— Glen Otis

### X.1 Outils et Méthodologie de Veille : Utilisation de Pearltrees

Pour effectuer une veille efficace sur les vulnérabilités de sécurité, j'utilise l'agrégateur de flux **Pearltrees**. Cet outil me permet de centraliser et de gérer mes sources d'information de manière structurée.

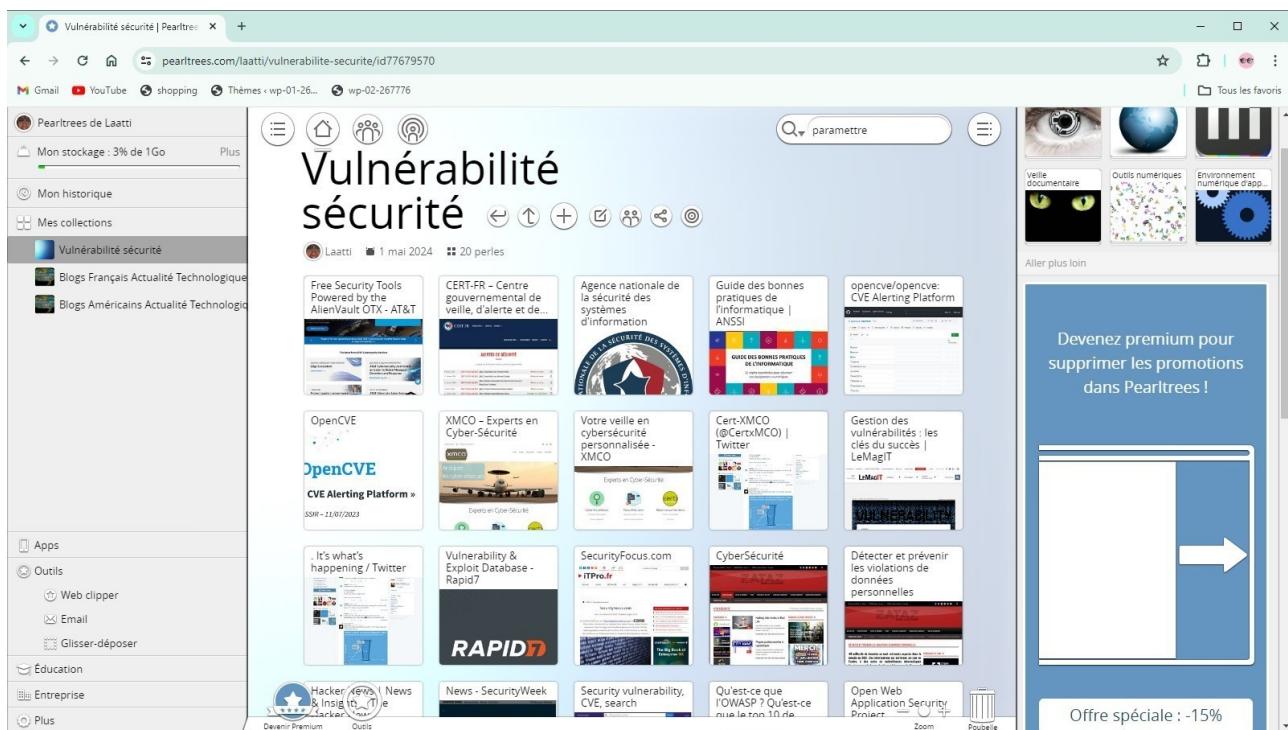


Figure 16 : Dashboard Pearltrees-Vulnérabilité-Sécurité

### Sources Intégrées dans mon Tableau de Bord :

- **OWASP** : Référence en matière de sécurité des applications web.
- **CVE Details** : Base de données des vulnérabilités connues.
- **SecurityWeek** et **The Hacker News** : Sites d'actualités sur la sécurité informatique.
- **Zataz Magazine** : Spécialisé dans la sécurité et le piratage.



- **SecurityFocus** : Informations sur les vulnérabilités et les exploits.
- **Exploit Database** : Archive d'exploits vérifiés.
- **Twitter World** : Pour suivre les experts et tendances en temps réel.
- **Le Mag IT** : Actualités IT avec une section dédiée à la sécurité.
- **CERT-XMCO** : Alertes et analyses de sécurité.
- **OpenCVE** : Suivi personnalisé des vulnérabilités.
- **ANSSI** : Agence Nationale de la Sécurité des Systèmes d'Information.
- **CER** : Centre d'Expertise en Réponse aux Incidents de sécurité.
- **AlienVault OTX** : Plateforme communautaire de renseignement sur les menaces.

## XI. Situation de travail nécessitant une recherche via des sites anglophones

### ➤ Contexte et apprentissage

Dans le cadre de mon projet **Studee**, j'ai appris et maîtrisé **React**, **React Native**, **NestJS**, et **GraphQL** grâce à des cours en ligne, des tutoriels, et la documentation officielle.

### ➤ Objectif

Mon objectif principal était de permettre aux utilisateurs de créer et de gérer leur profil étudiant afin de postuler à des stages. J'ai conçu et mis en œuvre le resolver `StudentProfile` à cet effet.

### ➤ Focus sur la sécurité

La sécurité était une préoccupation centrale. J'ai mené des recherches sur l'utilisation de `@UseGuards` dans **NestJS** et **GraphQL** pour sécuriser les mutations et query sensibles. J'ai appliqué ces connaissances dans le resolver `StudentProfile` (cf. figure VI. Resolver : `StudentProfileResolver`).

### ➤ Sources

Un article sur [Hashnode](#) m'a aidé à adopter les meilleures pratiques de sécurité pour les API **GraphQL**.

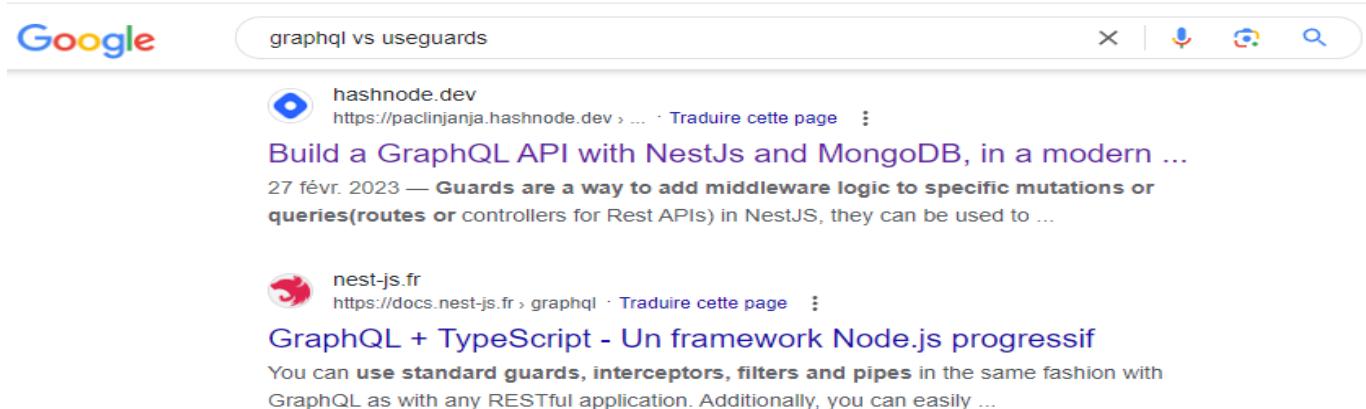


Figure 17: recherche via des sites anglophones faite sur Google



## ➤ Extrait du site anglophone :

Then, every strategy we will implement has to contain a validate() method, this is what the auth-guard will execute first, missing it in our class will make it fail, make sure we have it. Next, we are just calling the validateUser method from the authService, as we saw before, it will return a valid user if the credentials were correct, or a null if not, in the second case, we throw a native nest error UnauthorizedException(); telling the user that he's not authorized to proceed with the request, other wise, we just return the user(or not 😊) The fact is that, as you can see in our login Mutation, the data we are using for the auth.service is already part of the context, return this.authService.login (context.user); so we can or not return a user, since it won't be used in our use case. Well done, the login is completed. We have an additional guard implemented in the jwt-auth.guards.ts file. Now we can protect some resolvers from being queried by an unauthenticated user.

```
...  
// book.resolver.ts  
  
...  
// Only connected users with valid jwt tokens must create a book(Authentication)  
@Mutation(() => Book)  
@UseGuards(JwtAuthGuard)  
createBook(@Args('createBookInput') createBookInput: CreateBookInput) {  
  return this.bookService.createBook(createBookInput);  
}  
...  
...
```

Figure 18 : Code illustrant l'extrait du site anglophone

➤ **Extrait du site anglophone (traduction) :** Alors, chaque stratégie que nous implémenteront doit contenir une méthode « validate() », c'est ce que l'auth-guard exécutera en premier, le manque de celui-ci dans notre classe le fera échoué. Après, nous appelons juste la méthode « validateUser » depuis « authService », comme vu auparavant, cela retournera un utilisateur valide si les informations d'identification étaient correctes, ou si non un « null », dans le second cas, nous lançons une erreur native « UnauthorizedException(); » disant à l'utilisateur qu'il n'est pas autorisé à procéder à la demande, sinon, nous retournons juste l'utilisateur (ou pas 😊) Le fait est que, comme vous pouvez le voir dans notre mutation de connexion, les données que nous utilisons pour « auth.service » font déjà partie du contexte « return this.authService.login (context.user); » donc nous pouvons ou pas retourner un utilisateur, puisqu'il ne sera pas utilisé dans notre cas d'utilisation. Bien joué, la connexion est terminée. Nous avons un garde supplémentaire implémenté dans le fichier jwt-auth.guards.ts. Maintenant, nous pouvons protéger quelques résolveurs contre les requêtes d'utilisateur non authentifié.



## XII. Conclusion

Le projet Studee a été une expérience profondément enrichissante contribuant ainsi à mon développement tant sur le plan technique que personnel. La collaboration avec l'équipe de HDM Network, sous la direction et le mentorat de notre développeur senior, Quentin, a renforcé ma capacité à résoudre des problèmes de manière collective et à m'adapter rapidement aux exigences changeantes du projet.

Les réunions quotidiennes et les points de contrôle réguliers ont permis une coordination efficace, et l'environnement de travail agile a été un atout majeur pour surmonter les défis techniques rencontrés.

Ce projet m'a également donné l'occasion d'approfondir mes connaissances sur des stacks technologiques avancés et de me familiariser avec des outils essentiels du développement moderne.

Cette première expérience professionnelle dans le domaine du développement web m'a apporté une confiance accrue en mes compétences techniques et en ma capacité à assimiler rapidement de nouvelles technologies. Je suis profondément reconnaissant d'avoir pu contribuer à un projet aussi impactant, qui simplifie le processus de recrutement pour les étudiants et les entreprises, tout en me permettant de grandir en tant que développeur.

### XIII. Annexes



## Veille concurrentielle App stage et apprentissage

App Nom de l'app	Nombre d'étoiles	Nombre de téléchargements	Avantages	Inconvénients	Tarification	Services	Autres informations	Photos
<u>Vigijobs</u>	4,4 / 740 avis	+100 K téléchargement	-Application populaire +100k -Plus de 2000 sources d'offres d'emploi. - Application simple et performante - téléchargement rapide -tu ne verras plus 10 fois la même annonce	on peut postuler sans créer de compte ce qui dégrade la qualité de candidature.	gratuit	recherches un emploi, un stage, une alternance		
<u>Space Monk</u>	3,5/ 25 avis	+ 1000	- Facilité d'inscription - Design sympa - On peut voir ses statistiques	- Traitement des fichiers et chargement des pages trop longues. - Payer des forfaits pour candidater	- Forfaits basique ou premium Le premium on a les coordonnées du recruteur.	Recherche de contrat de stage, d'apprentissage de professionnalisation, de travail temporaire, de CDD et CDI		
<u>Workdating</u>	non précisé	9350	-Super design des pages - Chargement des pages rapides - Algorithme de Matching personnalisé par like -Tchat permettant d'échanger des messages et faire connaissance entre candidat et recruteur - Inscription gratuite pour les deux parties	-Abonnement obligatoire les recruteurs qui souhaitent échanger avec les candidats likés - Abonnement pour les candidats qui souhaitent voir les recruteurs qui les ont likés -N'incluent pas la recherche de stage.	Abonnement pour discuter avec un candidat qui correspond au profil recherché ou un recruteur qui a aimé votre profil	Recherche d'emploi et d'alternance, tout type de contrat		
<u>Visionjob</u>	3,5/5	100	-Super présentation des pages -Super design des pages - Chargement des pages rapides - Facilité de création de compte pour la demande et pour l'offre - Inscription	pas très riches en offres, et pas assez noté	aucune	recherche d'emploi, d'alternance et de stage		



App Nom de l'app	Nombre d'étoiles	Nombre de téléchargements	Avantages	Inconvénients	Tarification	Services	Autres informations	Photos
			gratuite pour les deux parties - Présentation Visio (cv) inclue Toutes les informations relatives au candidat peuvent être vues sur le profil de façon très détaillée					
<u>Joinrs</u>	Aucun avis	+ de 100 000	- Mise en avant des compétences dans un domaine. - Pourcentage pour savoir si le poste correspond au profil.	- Il faut impérativement une école physique pour s'y inscrire. - Aucun avis	Gratuite	annonces de stages et d'emplois pour étudiants et jeunes diplômés	+ de 250 entreprises ont des annonces sur l'applications	
<u>Jobteaser</u>	4,5 étoiles/ 756 avis	+ de 100k	- + de 4000 recruteurs partenaires - Triage des offres via un algorithme de recommandation - Chargement rapide Inscription rapide et simple.	L'offre ne correspond pas au profil	Gratuite	annonces de stage, d'alternances et de CDI pour étudiants et jeunes diplômés		

Figure 19: Veille Concurrentiel App stage et Apprentissage

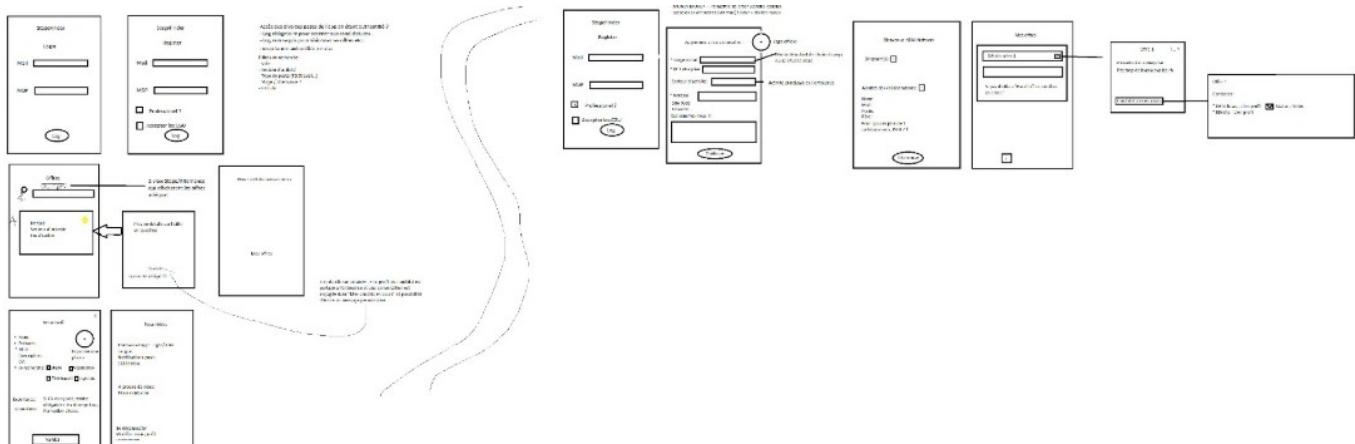


Figure 20: Première ébauche réalisée sur paint



Figure 21: Deuxième ébauche réalisé sur miro

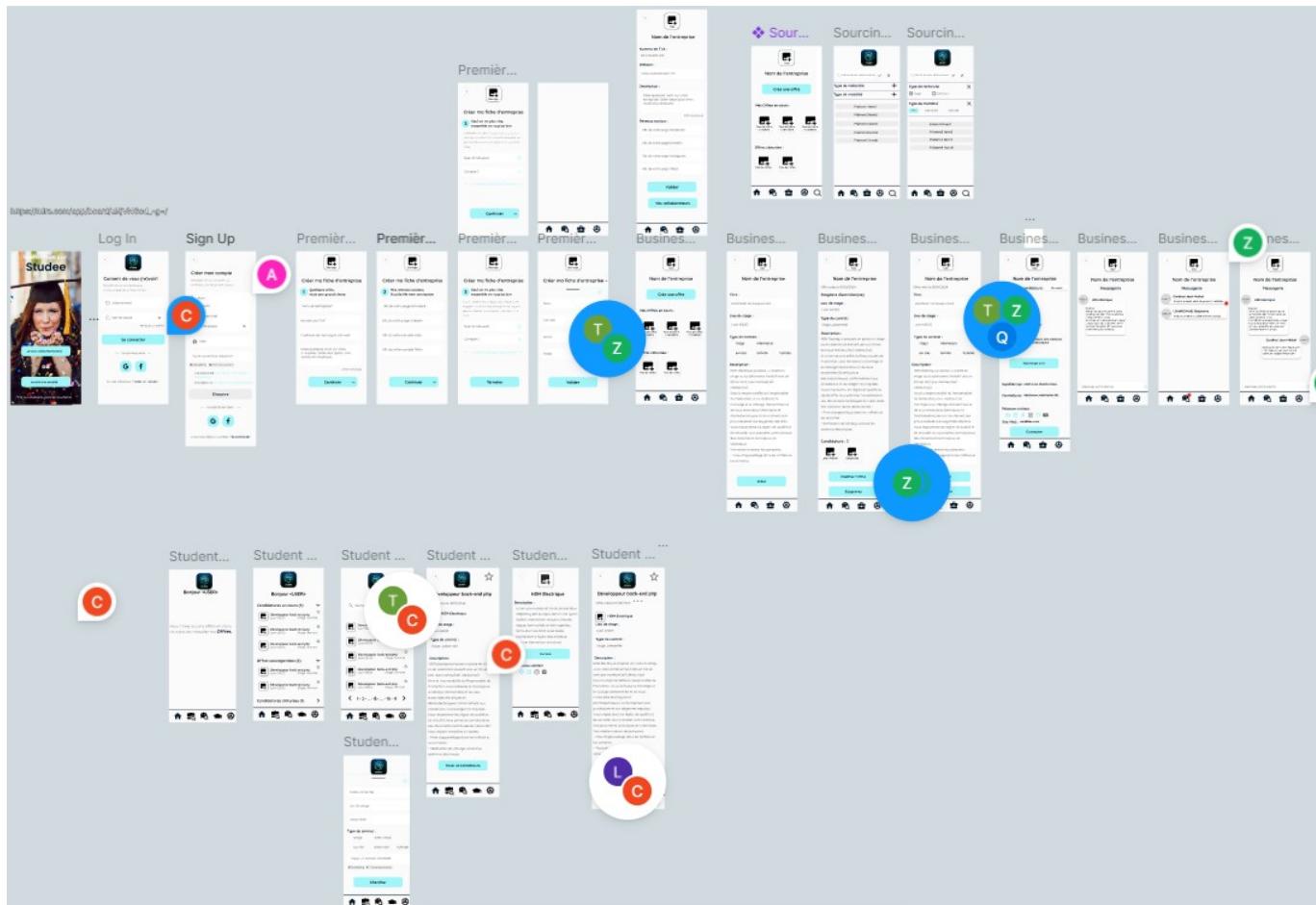


Figure 22: Maquette du projet Studee à mon arrivée en stage

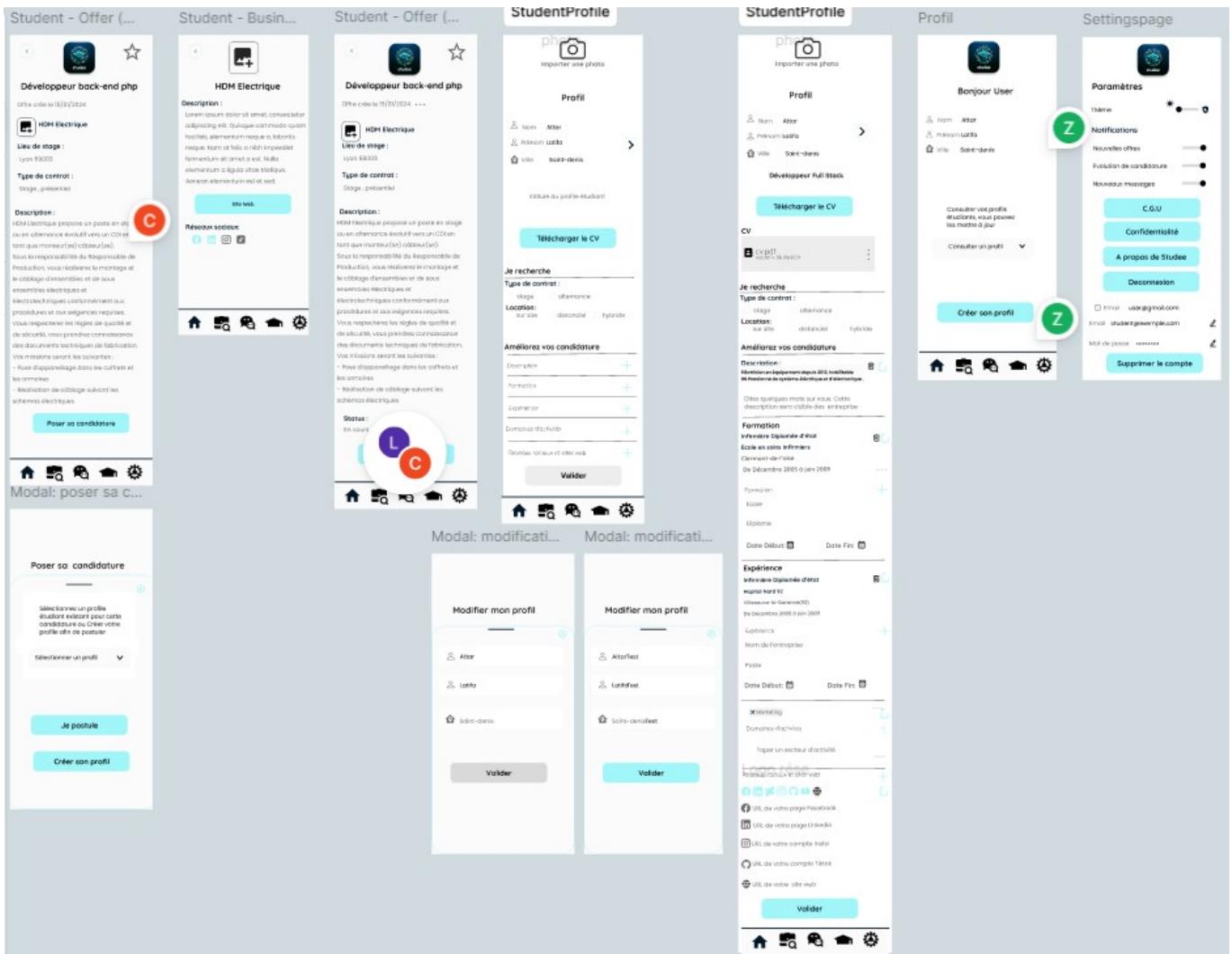


Figure 22: Maquette coté StudentProfile faite par mes soins

## INSTALLATION PROJET STUDEE

### Environnements

- ↗ Installer JetBrains Toolbox : Télécharger et installer webstorm version beta, Android Studio ( Play store : Expo Go, également sur son android)
- ↗ Installer Docker Desktop
- ↗ Installer MySQL Workbench



## Récupérer Github

- git clone <https://github.com/hdmnetwork/studee-mobile-app> repository
- git clone <https://github.com/hdmnetwork/studee-api> repository

## Mise en place du Projet

### Se rendre sur webstorm/Projet

Nommer le projet : Studee

git clone <https://github.com/hdmnetwork/studee-api.git>

Terminal : → git checkout dev  
→ yarn install

Aller sur ...webstorm/Studee/studee-api/src/.env

```
APP_ENV=local
JWT_SECRET=$2y$10$4zxKLip851WEYeVDDpsckeCrw6.f8wT7bXuOkCkWs6dAhTC9988
DATABASE_URL="mysql://${MYSQL_USER}:${MYSQL_PASSWORD}@localhost:3306/${MYSQL_DATABASE}"
API_PORT=3030
WORKERS_PORT=3032
GRAPHQL_PLAYGROUND_ENABLED=true
CDN_ACCESS_KEY_ID=ASK_TO_YOUR_TEAMLEADER_FOR_PROVIDING_IT
CDN_ACCESS_KEY=ASK_TO_YOUR_TEAMLEADER_FOR_PROVIDING_IT
CDN_PUBLIC_URL=https://hdmnetwork-websites-cdn.s3.fr-par.sgw.cloud/{myprojectname}
MYSQL_DATABASE=studee-db (choisir un nom de base de donnée)
MYSQL_USER=root
MYSQL_PASSWORD= choisir un mot de passe
MYSQL_ROOT_PASSWORD= choisir un mot de passe
```

git clone <https://github.com/hdmnetwork/studee-mobile-app.git>

Terminal : → git checkout dev  
→ yarn install

git clone <https://github.com/hdmnetwork/mysql5.7.git>

Terminal : → git checkout dev  
→ yarn install

Aller sur ...webstorm/Studee/mysql57/.env

```
COMPOSE_PROJECT_NAME=mysql57
## MySQL ##
MYSQL_DATABASE=studee-db
MYSQL_USER=mysql57
MYSQL_PASSWORD=choisir un mot de passe
```



```
MYSQL_ROOT_PASSWORD=choisir un mot de passe
```

Terminal : → ...webstorm/Studee/mysql57/  
(toutes les commandes sont dans install.md)

*How to install Mysql on your local machine*

1. Update .env file by adding user & root DB passwords
2. Install docker on your machine
3. Execute `docker network create mysql57`
4. Execute `docker volume create mysql57`
5. Execute `docker compose -f stack.local.yml up -d`
6. Execute `docker ps` and check if the container is running
7. Enjoy

## Allez sur Mysql workbench

MySQL Connection

Name : Studee-db

Paramètres : port : 3306

## Retour sur webstorm

Terminal

Aller ...\\WebstormProjects\\Studee\\studee-api

Commande à exécuter :

- ▶ npx prisma migrate dev
- ▶ npx prisma generate dev
- ▶ npx prisma db seed

(S'assurer que sur Mysql Workbench la base de donnée est bien générée)

Lancer studee-api → yarn run start dev

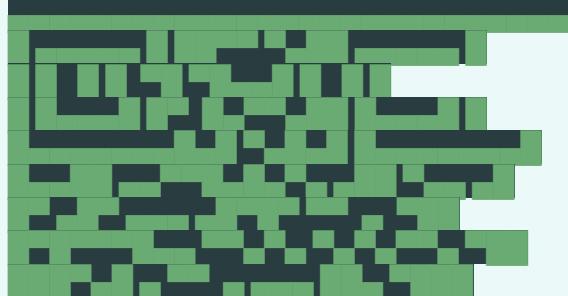
Ouvrir un autre terminal

Aller ...\\WebstormProjects\\Studee\\studee-mobile-app

Commande à exécuter :

- ▶ yarn install
- ▶ yarn web
- ▶ Enjoy :

Starting Metro Bundler





À récupérer

› Metro waiting on exp://127.0.0.1:8081  
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

ou ouvrir un autre terminal

ipconfig

recupérer ipv4

▶ Web is waiting on <http://localhost:8081>

Aller ...\\WebstormProjects\\Studee\\studee-mobile-app\\readme.md

```
{  
  /* */  
  "extra": {  
    "api": "http://xxx.xxx.x.xx:3030/graphql",  
    "api-websocket": "ws://xxx.xxx.x.xx:3030/graphql"  
    /* */  
  }  
}
```

Aller ...\\WebstormProjects\\Studee\\studee-mobile-app\\app.json

```
"extra": {  
  /* lati Connexion */  
  "api": "http://ip/graphql",  
  "api-websocket": "ws://ip/graphql",  
  // "api": "PROD_URL",  
  // "api-websocket": "PROD_URL",  
}
```

À intégrer

À intégrer

si pb avec ad ip npm start -clear  
ou npm start -- --reset-cache

lancer android  
virtual device manager  
play  
retourner sur terminal front, taper a , l'imulator se lance

Document *Install\_studee* fait par mes soins pour les stagiaires suivant