



Conception d'une application web de freelance

Encadrée par : Mr O. Zemzami

Année universitaire : 2019 / 2020

Remerciement :

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique de l'école supérieure de technologie et les intervenants professionnels responsables de la formation génie informatique.

Avant d'entamer ce rapport, nous profitons de l'occasion pour remercier tout d'abord notre professeur Monsieur Omar Zemzami d'avoir accepté nous encadrer durant la réalisation de notre PFE. Nous le remercions également pour l'aide et les conseils concernant les missions évoquées dans ce rapport.

En fin, nous ne pouvons achever ce projet sans exprimer nos gratitude à tous nos professeurs, pour leur dévouement et leur assistance tout au long de cette année.

Dédicace :

On dédie ce travail :

À nos parents, pour leurs amour et sacrifices.

À nos adorables frères, sœurs pour leur patience,

À nos proches amis et toute notre grande famille, pour leurs soutient et encouragements.

À toutes les personnes qui nous connaissent de près ou de loin.

Table of Contents

Remerciement :	2
Dédicace :	3
Table of Contents :	4
Notation et Symboles :	7
I. Introduction générale :	8
II. Présentation du projet :	9
1. Introduction :	9
2. Description du projet :	9
a. C'est quoi un site web dynamique ?	9
b. Comment fonctionne un site web dynamique ?	9
3. Présentation du projet :	10
a. C'est quoi un site de freelance ?	10
c. L'objectif :	10
d. Comment ça marche ?	10
4. A qui s'adresse de projet ?	10
5. Les fonctionnalités de l'application :	10
6. Les classes d'usages du système :	11
a. Administrateur :	11
e. Freelancer / Entreprise :	11
7. L'étude de l'existence :	11
8. Critique de l'existence :	11
9. Solution :	11
10. Conclusion :	12
III. Conception :	13
1. Introduction :	13
2. Modélisation UML :	13
a. A quoi sert cette méthode ?	13
f. Diagramme de classe :	13
g. Diagramme de cas d'utilisation :	14
h. Diagramme d'activité :	15
3. Conclusion :	18

IV.	Outils de travail :	19
1.	Introduction :	19
2.	Logiciels :	19
a.	Node.js :	19
b.	Visual Studio code :	19
c.	MongoDB compass :	20
d.	Navigateur WEB :	20
e.	Sparx Systems Entreprise Architect :	20
3.	Langage :	20
a.	React JS :	20
b.	JavaScript :	21
c.	Graph QL :	21
d.	Mongoose JS :	21
e.	Apollo Server :	22
f.	Bootstrap :	22
g.	jQuery :	23
h.	Tippy JS :	23
4.	Conclusion :	23
V.	Réalisation :	24
1.	Introduction :	24
2.	Les interfaces graphiques :	24
a.	Accueil :	24
a.	Inscription & connexion :	25
b.	Profile du client :	27
c.	Tableau de bord :	29
d.	Poster un travail :	30
e.	Page de travail :	31
f.	Poster un projet :	31
g.	Page de projet :	32
h.	Gérer les postes :	32
i.	La messagerie :	34
a.	Les paramètres :	35
b.	Chercher un employant / une entreprise:	37
3.	Conclusion :	38

VI.	Conclusion :	39
VII.	Bibliographie :	40

Notation et Symboles :

Figure 1: Diagramme de classe	14
Figure 2: Diagramme de cas d'utilisation	15
Figure 3: Diagramme d'activité "Inscription"	16
Figure 4: Diagramme d'activité « Poster un travail / projet »	17
Figure 5: Diagramme d'activité "enchérir sur projet / travail "	18
Figure 6: page d'accueil	25
Figure 7: inscription étape 1	26
Figure 8: inscription étape 2	26
Figure 9: inscription étape final	27
Figure 10: formulaire de connexion	27
Figure 11: Profile d'utilisateur (freelancer)	28
Figure 12: profile d'entreprise	29
Figure 13: tableau de bord	30
Figure 14: poster un nouveau travail	30
Figure 15: Page de travail (job)	31
Figure 16: poster un nouveau projet	32
Figure 17: manager les travaux postés	33
Figure 18: gérer les candidatures reçu	33
Figure 19: Manager les projets postés	34
Figure 20: manager les enchère du projet	34
Figure 21: interface de messagerie	35
Figure 22: interface des paramètres	36
Figure 23: chercher un employant (freelancer)	37
Figure 24: chercher une entreprise	38

I. Introduction générale :

Être un Freelancer, signifie que plutôt que de signer un contrat pour travailler pour une entreprise, vous travaillez pour vous-même et gérez vos propres affaires fiscales. Cela vous donne la liberté de travailler pour qui vous voulez, autant ou aussi peu que vous le souhaitez. Vous êtes flexible et le maître de votre propre destin.

Lorsque vous lancez votre carrière après avoir obtenu votre diplôme - ou même pendant que vous étudiez encore - le travail en freelance offre un excellent moyen d'essayer différentes industries ou entreprises et d'obtenir cette expérience si importante que les employeurs exigent.

L'une des caractéristiques les plus encourageantes en freelance est que vous n'avez pas besoin de commencer avec un capital. Tout ce dont vous aurez besoin est vos compétences techniques et un bon niveau de communication. Par conséquent vous allez être capable de produire un capital pour commencer un business ou lancer sa propre entreprise.

En effet, le Freelance est considéré comme une solution pour plusieurs personnes – surtout les jeunes – qui ont un grand talent dans un domaine spécifique mais ils n'ont pas le budget pour en investir.

L'objectif de notre projet est de modéliser puis mettre en place une base de données permettant d'alimenter un site web qui va permettre aux entreprises et aux Freelances de communiquer entre eux. Afin de mettre en place ce projet, il est tout d'abord nécessaire de comprendre le fonctionnement d'un système de freelance. Nous allons donc présenter ce modèle de fonctionnement sous forme des diagrammes. Une fois cette modélisation terminée, il faut traduire le diagramme de classe obtenu dans le modèle relationnel afin de l'introduire dans Mongo DB.

II. Présentation du projet :

1. Introduction :

Dans ce chapitre, nous commencerons par une présentation de notre projet. Ensuite, nous déterminerons les objectifs à atteindre. Ainsi, nous nous intéresserons à l'étude de l'existant et de ses critiques pour proposer des solutions appropriées et possibles.

2. Description du projet :

a. C'est quoi un site web dynamique ?

Un site web dynamique est un ensemble de pages web dynamiques qui sont des pages web générées à la demande, par opposition à des pages web statiques. Le contenu d'une page web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page web statique est a priori identique à chaque consultation.

b. Comment fonctionne un site web dynamique ?

Lorsque vous voulez visiter un site web, vous tapez son adresse dans votre navigateur web, que ce soit Google Chrome, Internet Explorer, Opera, Safari ou un autre. Mais ne vous êtes-vous jamais demandé comment faisait la page web pour arriver jusqu'à vous ?

Il faut savoir qu'Internet est un réseau composé d'ordinateurs. Ceux-ci peuvent être classés en deux catégories.

Les clients : Ce sont les ordinateurs des internautes comme vous. Votre ordinateur fait donc partie de la catégorie des clients. Chaque client représente un visiteur d'un site web. Dans les schémas qui vont suivre, l'ordinateur d'un client sera représenté par l'image suivante.

Les serveurs : Ce sont des ordinateurs puissants qui stockent et délivrent des sites web aux internautes, c'est-à-dire aux clients. La plupart des internautes n'ont jamais vu un serveur de leur vie. Pourtant, les serveurs sont indispensables au bon fonctionnement du Web.

3. Présentation du projet :

a. C'est quoi un site de freelance ?

Un site web de freelance est un outil qui permet aux freelancers de bénéficier de ses compétences en faisant un revenu et aux employés de trouver la bonne personne pour exécuter une ou plusieurs tâches avec un timeout et une somme de money spécifiée.

c. L'objectif :

L'objectif de notre projet c'est créer un site web dynamique de freelance puis mettre en place une base de données afin d'aider les personnes qui ont besoin des opportunités pour bénéficier de leurs compétences dans le but de faire un revenu. Pour les employés (entreprise, personne...) notre plateforme doit faciliter la recherche des profils convenables pour un poste ou une tâche ou bien un projet.

d. Comment ça marche ?

Chaque client doit commencer par s'inscrire afin de créer un compte en spécifiant le type d'utilisateur. Une fois le compte est complètement créé, le client peut consulter les divers services de l'application. Chaque client peut poster une tâche ou un projet à faire avec les détails qui seront consultables par les autres clients. Chaque client peut poser sa candidature sur une tâche/projet, cette candidature sera acceptée ou refusée par le posteur selon le profil du candidat.

4. A qui s'adresse de projet ?

Notre site web est adressé à toutes personnes qui veulent bénéficier de ses compétences aussi les personnes qui ont besoin de ces compétences.

5. Les fonctionnalités de l'application :

Après une analyse du sujet, nous avons décidé de faire apparaître les fonctionnalités principales suivantes :

❖ Partie client :

- S'inscrire
- Se connecter/déconnecter
- Poster tâche/projet
- Poser une candidature
- Chercher tâche/projet/freelancer/entreprise
- Consulter sa boîte de messagerie et notification
- Consulter tableau de bord
- Ajouter note/review

- Noter opération/compte

❖ **Partie administrateur :**

- Se connecter/déconnecter
- Voir les statistiques
- Vérifier les comptes
- Bloquer les comptes

6. Les classes d'usages du système :

a. Administrateur :

L'administrateur est le seul à avoir accès à l'information complète du système. Il peut faire la gestion globale du système, par exemple créer des comptes virtuels. Il lui est aussi possible de suspendre n'importe quel compte.

e. Freelancer / Entreprise :

Chacun d'eux est un utilisateur (client) qui possède un compte qui contient leurs informations, soit des informations privées comme mot de passe ou public comme les compétences. Chaque utilisateur peut poster une tâche ou un projet que les autres utilisateurs peuvent consulter

7. L'étude de l'existence :

La plupart des personnes qui veulent commencer un business rencontrent le problème de manque du capital. Alors, ils auront deux choix. Soit prendre un crédit chez une banque, la chose qui n'est plus recommandée. Soit chercher un travail pour gagner de l'argent la chose qui prendra du temps et de l'énergie.

8. Critique de l'existence :

Les deux solutions actuelles posent des problèmes différents :

- ❖ L'incertitude du succès du business à commencer.
- ❖ Le risque de perdre beaucoup de temps et d'énergie en cherchant un travail
- ❖ Le risque d'échouer à payer les devoirs du crédit au temps exact

9. Solution :

Afin de répondre à ces problématiques, le freelance est une solution véritable qui peut diminuer les risques et les pertes financières. Vous n'êtes plus besoin de plusieurs diplômes pour trouver un poste qui vous aidera à équilibrer votre situation financière. Tous que vous en aurez besoin c'est vos expériences et vos compétences, et le plus important c'est que vous pouvez commencer le temps qu'il vous convient.

10. Conclusion :

Dans ce chapitre nous avons présenté notre projet en général, ainsi que les fonctionnalités et les tâches que nous allons réaliser, dans le prochain chapitre nous allons discuter la phase de conception de notre projet.

III. Conception :

1. Introduction :

Ce chapitre expose la solution conceptuelle qui a pour but de rendre flexible la tâche de la gestion et une projection de la réalité physique. Donc nous avons opté pour l'utilisation de la notation UML qui est un standard très largement utilisé dans l'analyse et la conception, cette phase de conception est effectuée par plusieurs niveaux, afin d'aboutir à un système fonctionnel reflétant une réalité physique.

2. Modélisation UML :

a. A quoi sert cette méthode ?

Modéliser et construire des systèmes logiciels de manière fiable.

Une méthode définit :

- ❖ Des éléments de modélisation.
- ❖ Une représentation graphique.
- ❖ Du savoir-faire et des règles.

La notation UML (Unified Modeling Language) est une méthode de modélisation objet. Elle permet d'élaborer des modèles valables pour n'importe quel langage de programmation afin de faire une modélisation de notre base de données, nous allons utiliser UML. Nous détaillerons donc la réalisation de trois diagrammes : tout d'abord le diagramme de classe puis le diagramme des cas d'utilisations enfin le diagramme d'activité.

f. Diagramme de classe :

Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

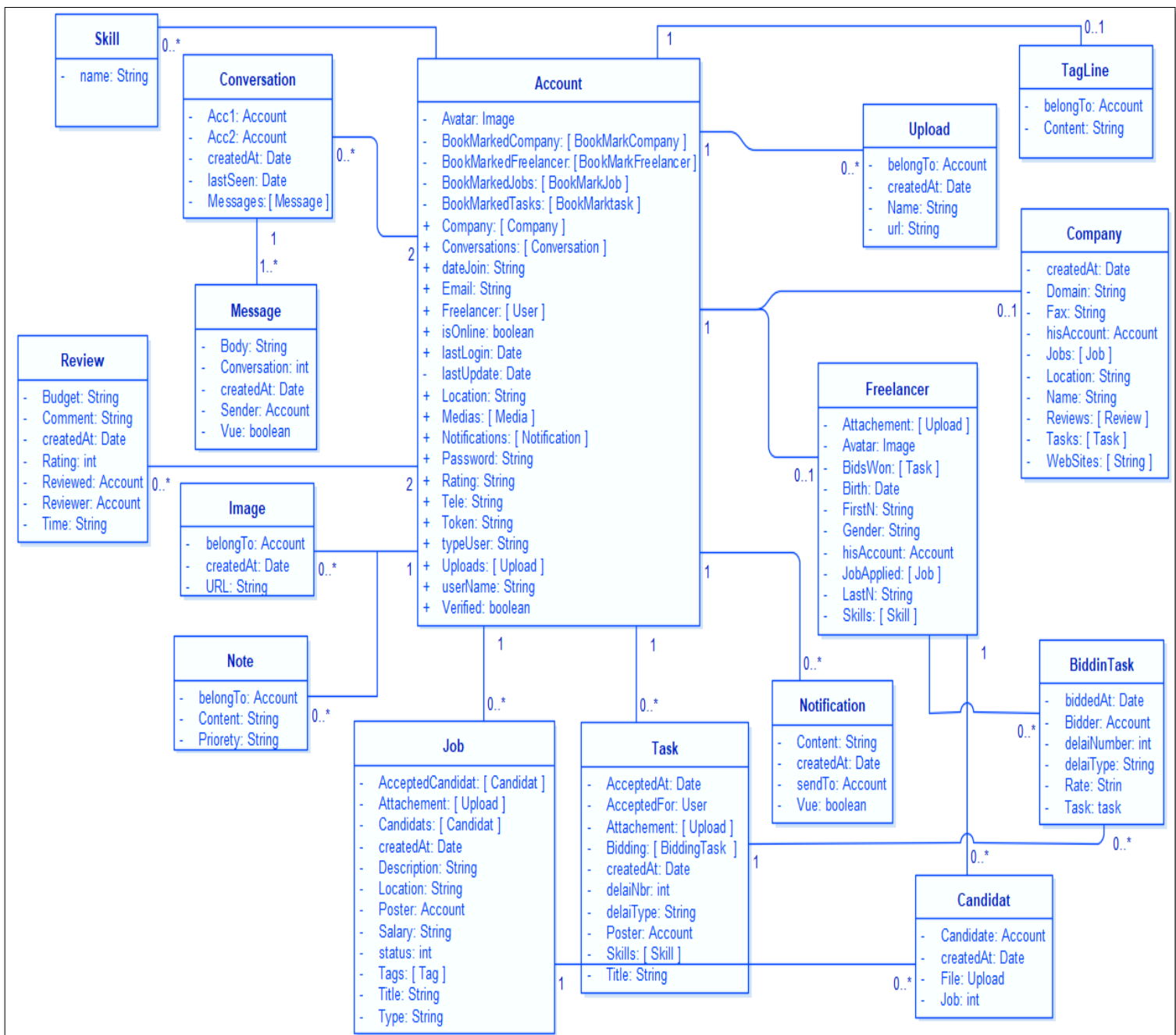


Figure 1: Diagramme de classe

g. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur et un système. Il est une entité significative de travail.

Dans un diagramme de cas d'utilisation il existe des acteurs qui interagissent avec des cas d'utilisation. Les cas d'utilisation structurent les besoins de client et les objectifs du système.

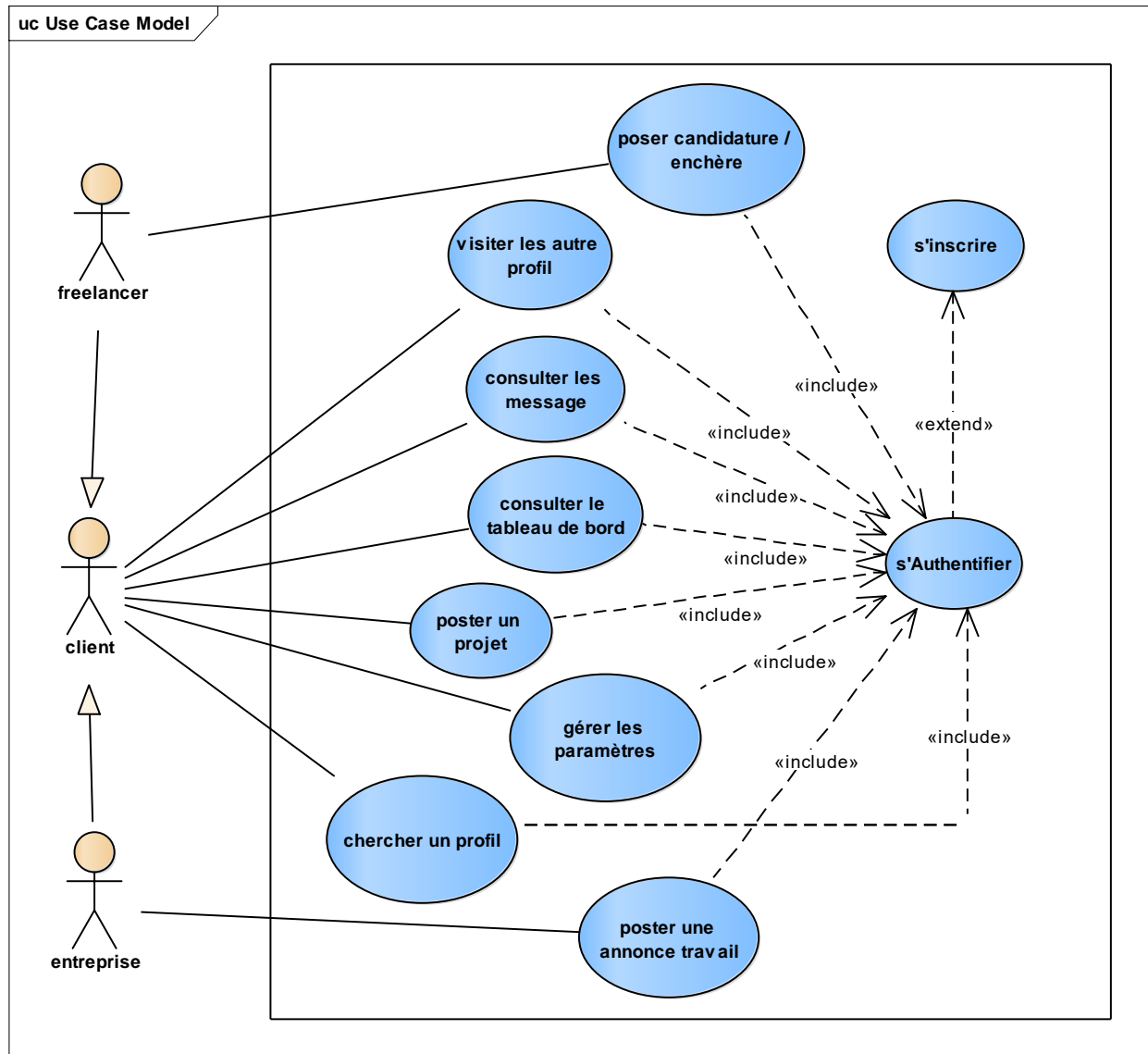


Figure 2: Diagramme de cas d'utilisation

h. Diagramme d'activité :

Un diagramme d'activité permet de modéliser un processus interactif global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de diagrammes de classes ou de cas d'utilisation

Par exemple :

- ❖ Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.
- ❖ Le diagramme d'activité est sémantiquement proche des diagrammes de communication (appelés diagramme de collaboration en UML), ou d'état-transitions, ces derniers offrant une vision microscopique des objets du système.

i. Inscription :

Cette activité est disponible pour tout internaute, il suffit d'avoir une adresse email valide pour réussir cette étape.

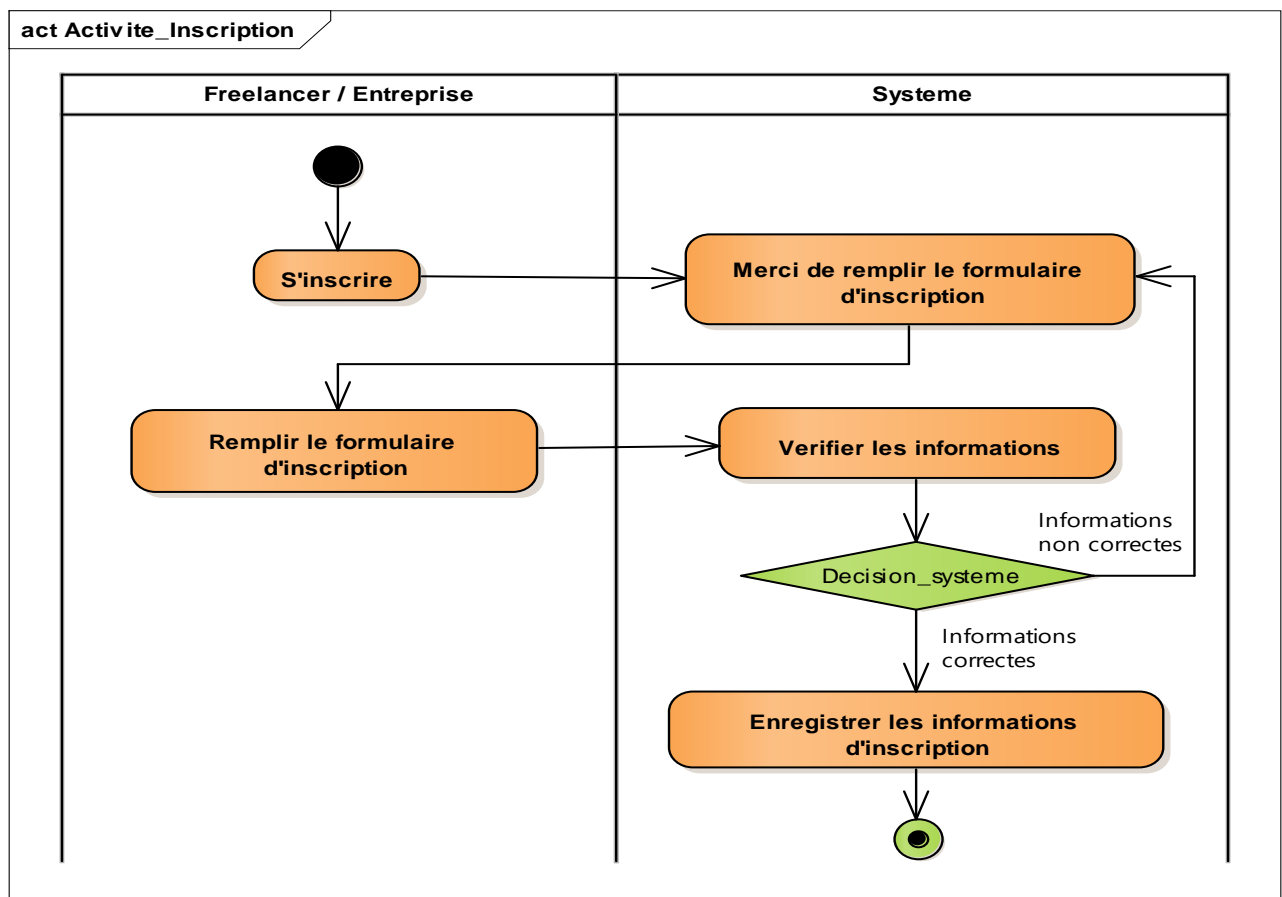


Figure 3: Diagramme d'activité "Inscription"

ii. Poster un Travail / Projet :

Le client peut créer un post soit pour annoncer un travail (job) ou un projet. Ce poste sera consulté par les autres clients. Ils peuvent aussi poser leurs candidatures pour prendre le travail / projet.

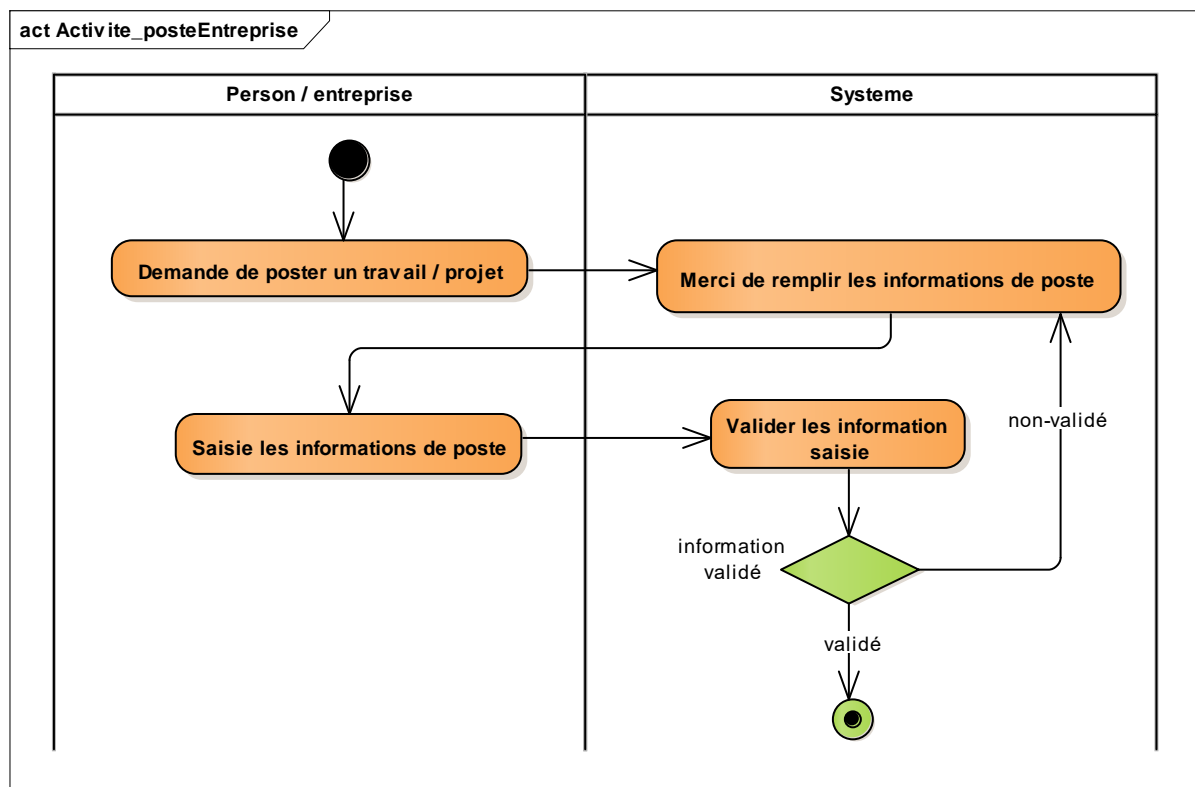


Figure 4:Diagramme d'activité « Poster un travail / projet »

iii. Enchérir sur un emploi / projet :

Une fois un poste de travail / projet est créé. Les freelances peuvent poser leurs candidatures pour gagner l'opportunité. Les candidatures seront transmises au propriétaire du poste. Ce dernier va sélectionner les candidats qui lui convient selon leurs profils ou leurs cv (ou les deux).

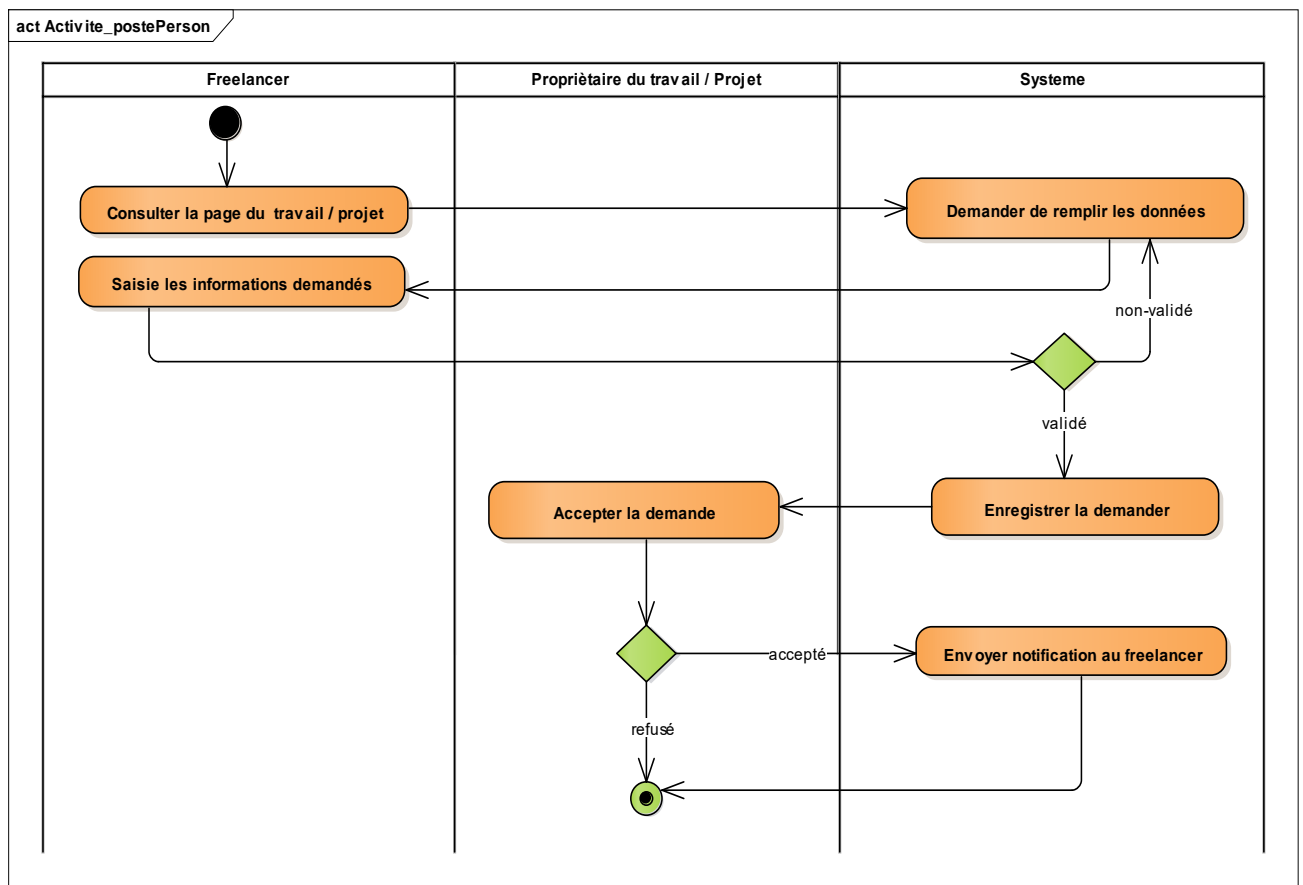


Figure 5: Diagramme d'activité "enchérir sur projet / travail "

3. Conclusion :

La phase conceptuelle est une étape fondamentale pour la réalisation de n'importe quel projet. Elle permet de faciliter le système d'information et réaliser l'implémentation de la base de données et le traitement. Par la suite, nous devons chercher les moyens et les outils possibles pour développer l'application, ce que nous allons présenter dans le chapitre suivant.

IV. Outils de travail :

1. Introduction :

Ce chapitre a pour objet de présenter les outils et technologies mis en œuvre pour la réalisation de la plateforme. Un accent particulier est mis sur les solutions open-source, qui n'ont aujourd'hui rien à envier aux solutions propriétaires, qui offrent un large éventail de possibilités et opportunités toutes aussi intéressantes les unes que les autres. Cette étude technique exhibera la richesse et la puissance des technologies déployées pour réussir la réalisation du projet.

2. Logiciels :

a. Node.js :

Node.js est un environnement d'exécution JavaScript open source et multiplateforme. C'est un outil populaire pour presque n'importe quel type de projet! Node.js exécute le moteur JavaScript V8, le cœur de Google Chrome, en dehors du navigateur. Cela permet à Node.js d'être très performant.

Une application Node.js est exécutée en un seul processus, sans créer de nouveau thread pour chaque demande. Node.js fournit un ensemble de primitives d'E / S (I / O) asynchrones dans sa bibliothèque standard qui empêchent le code JavaScript de bloquer.

Cela permet à Node.js de gérer des milliers de connexions simultanées avec un seul serveur sans introduire le fardeau de la gestion de la simultanéité des threads, ce qui pourrait être une source importante de bogues.

Node.js a un avantage unique car des millions de développeurs frontend qui écrivent JavaScript pour le navigateur sont désormais en mesure d'écrire le code côté serveur en plus du code côté client sans avoir besoin d'apprendre une langue complètement différente.

b. Visual Studio code :

Visual Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et MacOS. Il inclut la prise en charge du débogage, le contrôle Git et GitHub, la mise en évidence de la syntaxe, l'achèvement intelligent du code, des extraits de code et de la refactorisation du code. Il est hautement personnalisable, permettant aux utilisateurs de modifier le thème, les raccourcis clavier, les préférences et d'installer des extensions qui ajoutent des fonctionnalités supplémentaires.

c. MongoDB compass :

MongoDB Compass est l'outil graphique de facto pour MongoDB tout comme MySQL Workbench est l'outil associé de MySQL. Il vous permet d'explorer visuellement vos données, d'exécuter des requêtes ad hoc, d'interagir avec vos données avec la fonctionnalité CRUD complète, ainsi que de visualiser et d'optimiser les performances de vos requêtes. Il est disponible pour Linux, MacOS et Windows.

d. Navigateur WEB :

Le navigateur Web est une application logicielle utilisée pour afficher des pages Web ou des sites Web, par exemple Internet Explorer, Google Chrome, Safari, Firefox etc. Chaque navigateur a une barre d'adresse où l'utilisateur met son URL requise. Le navigateur Web récupère les informations de l'ordinateur serveur et les affiche sur l'ordinateur client via le protocole http ou https.

e. Sparx Systems Enterprise Architect :

Sparx Systems Enterprise Architect est un outil de modélisation et de conception visuelle basé sur l'OMG UML. La plateforme prend en charge: la conception et la construction de systèmes logiciels; modélisation des processus d'affaires; et la modélisation de domaines industriels. Il est utilisé par les entreprises et les organisations non seulement pour modéliser l'architecture de leurs systèmes, mais pour traiter la mise en œuvre de ces modèles tout au long du cycle de vie du développement d'applications.

3. Langage :

a. React JS :

React.JS est une bibliothèque JavaScript open source qui est utilisée pour créer des interfaces utilisateur en particulier pour les applications d'une seule page. React JS utilisé pour gérer la couche de vue pour les applications Web et mobiles. React JS permet à l'utilisateur de créer des composants d'interface utilisateur réutilisables.

React JS n'est pas un framework. Ce n'est qu'une bibliothèque développée par Facebook pour résoudre certains problèmes auxquels nous sommes confrontés lors de la création de composants d'interface utilisateur.

React JS permet aux développeurs de créer de grandes applications Web qui peuvent modifier les données, sans recharger la page. La fonction principale de React JS est de rendre la page Web rapide et simple. Il ne fonctionne que sur les interfaces utilisateur dans l'application. React JS peut être utilisé avec une combinaison d'autres bibliothèques ou framework JavaScript, tels que Angular JS dans MVC.

Une application React est composée de plusieurs composants, chacun étant responsable de l'interprétation d'un petit morceau HTML réutilisable. Les composants peuvent être imbriqués dans d'autres composants pour permettre la création d'applications complexes à partir de blocs de construction simples. Un composant peut également conserver un état interne par exemple; un composant Tab List peut stocker une variable correspondant à l'onglet actuellement ouvert.

b. JavaScript :

JavaScript est un langage de script conçu principalement pour ajouter de l'interactivité aux pages Web et créer des applications Web. Les programmes ou scripts JavaScript côté client peuvent être intégrés directement dans la source HTML des pages Web. Selon l'intention du développeur Web, le code de script peut s'exécuter lorsque l'utilisateur ouvre la page Web, clique ou fait glisser un élément de page avec la souris, tape quelque chose sur le clavier, soumet un formulaire ou quitte la page.

JavaScript est un langage orienté objet avec héritage prototypique. Le langage prend en charge plusieurs objets intégrés et les programmeurs peuvent créer ou supprimer leurs propres objets. L'héritage prototypique rend JavaScript très différent des autres langages de programmation populaires tels que C ++, C # ou Java avec des classes et l'héritage classique.

c. GraphQL :

GraphQL est une nouvelle et passionnante technologie qui permet aux développeurs de créer des API évolutives sans se sauter dans les limites des API REST traditionnelles. Essentiellement, il fournit une description facile à comprendre des données de votre API et permet aux clients de demander exactement ce dont ils ont besoin - rien de plus, rien de moins.

La principale motivation derrière GraphQL est de rendre la requête plus intelligente. Ainsi, au lieu d'une API où vous tapez une URL et acceptez toutes les données qui reviennent, GraphQL vous permet de spécifier exactement les données dont vous avez besoin. De cette façon, vous pouvez:

- ❖ Demander des types de données spécifiques.
- ❖ Agréger les données de plusieurs sources.
- ❖ Creusez plus profondément dans le point de terminaison de votre API et récupérez des informations sélectives.

d. Mongoose JS :

Mongoose JS est un mappeur de document objet (ODM) qui facilite l'utilisation de MongoDB en traduisant des documents dans une base de données MongoDB en objets dans le programme.

Outre Mongoose JS, plusieurs autres ODM ont été développés pour MongoDB, notamment Doctrine, MongoLink et Mandango.

e. Apollo Server :

Apollo Server est une bibliothèque qui vous aide à connecter un schéma GraphQL à un serveur HTTP dans Node. Apollo Server fonctionne avec n'importe quel schéma GraphQL construit avec GraphQL.js, vous pouvez donc construire votre schéma avec celui-ci directement ou avec une bibliothèque pratique telle que graphql-tools. Vous pouvez utiliser Apollo Server avec tous les serveurs HTTP JavaScript courants, notamment Express.

Ce serveur peut être interrogé à partir de n'importe quel client GraphQL, car il prend en charge toutes les sémantiques courantes pour l'envoi de GraphQL sur HTTP. Apollo Server prend également en charge certaines petites extensions du protocole, telles que l'envoi de plusieurs opérations GraphQL en une seule demande. En savoir plus sur la page d'envoi des demandes.

Apollo Server est un serveur GraphQL open source conforme aux spécifications, compatible avec tout client GraphQL. C'est le meilleur moyen de créer une API GraphQL prête à la production et auto-documentée qui peut utiliser les données de n'importe quelle source.

f. Bootstrap :

Bootstrap est un framework front-end très puissant pour un développement Web plus rapide et plus facile. Il comprend des modèles de conception basés sur HTML et CSS pour les composants d'interface utilisateur courants tels que la typographie, les formulaires, les boutons, les tableaux, les navigations, les listes déroulantes, les alertes, les modaux, les onglets, l'accordéon, le carrousel et bien d'autres, ainsi que des extensions JavaScript optionnelles.

Bootstrap vous donne également la possibilité de créer une mise en page réactive avec beaucoup moins d'efforts. Voici quelques avantages supplémentaires, pourquoi on devrait opter pour Bootstrap:

- ❖ Gagnez beaucoup de temps - Vous pouvez gagner beaucoup de temps et d'efforts en utilisant les modèles et classes de conception prédéfinis de Bootstrap et vous concentrer sur d'autres travaux de développement.
- ❖ Fonctionnalités réactives - En utilisant Bootstrap, vous pouvez facilement créer des conceptions réactives. Les fonctionnalités réactives au Bootstrap
- ❖ Permettent à vos pages Web d'apparaître de manière plus appropriée sur différents appareils et résolutions d'écran sans aucune modification du balisage.
- ❖ Conception cohérente - Tous les composants Bootstrap partagent les mêmes modèles et styles de conception via une bibliothèque centrale, afin que les conceptions et les mises en page de vos pages Web soient cohérentes tout au long de votre développement.
- ❖ Facile à utiliser - Bootstrap est très facile à utiliser. Toute personne possédant les connaissances de base de HTML et CSS peut commencer le développement avec Bootstrap.

- ❖ Compatible avec les navigateurs - Bootstrap est créé avec les navigateurs modernes à l'esprit et il est compatible avec tous les navigateurs modernes tels que Mozilla Firefox, Google Chrome, Safari, Internet Explorer et Opera.
- ❖ Open Source - Et la meilleure partie est, il est entièrement gratuit à télécharger et à utiliser.

g. jQuery :

jQuery est une bibliothèque JavaScript multi-navigateur conçue pour simplifier les scripts côté client de HTML. Il a été publié en janvier 2006 par John Resig. Utilisée par plus de 31% des 10 000 sites Web les plus visités, jQuery est la bibliothèque JavaScript la plus populaire actuellement utilisée.

jQuery est un outil gratuit et open source, sous double licence sous la licence MIT et la licence publique générale GNU. La syntaxe de jQuery est conçue pour faciliter la navigation dans un document, la sélection d'éléments DOM, la création d'animations, la gestion d'événements et le développement et les applications Ajax.

jQuery offre également aux développeurs la possibilité de créer des plugins au-dessus de la bibliothèque JavaScript. À l'aide de ces fonctionnalités, les développeurs peuvent créer des abstractions pour une interaction et une animation de bas niveau, des effets avancés et des widgets thématiques de haut niveau. Cela contribue à la création de pages Web puissantes et dynamiques. jQuery utilisé pour rendre la page plus rapide sans recharger les pages, compatible avec plusieurs navigateurs, peut être utilisé pour la conception de type d'animation, etc.

h. Tippy JS :

Tippy Js est la solution complète d'info-bulles, de pop-over, de liste déroulante et de menu pour le Web, optimisée par Popper. C'est une abstraction générique pour la logique et le style des éléments qui sortent du flux du document et flottent à côté d'un élément de référence, superposé au-dessus de l'interface utilisateur

4. Conclusion :

Les logiciels et les langages cités précédemment sont indispensables à la réalisation de notre projet. Par la suite, nous allons montrer quelques captures d'écrans des pages de notre site web qui vont vous donner une idée plus claire sur les composants de notre site web.

V. Réalisation :

1. Introduction :

Dans ce chapitre, notre objectif sera présenter le produit final et le résultat des chapitres précédents. Nous allons présenter quelques interfaces dans le but de vous situer dans cadre du projet et vous expliquer comment utiliser notre application.

2. Les interfaces graphiques :

a. Accueil :

La page d'accueil est consultable par tous internaute, il contient des informations à propos de notre application et quelques statistiques :

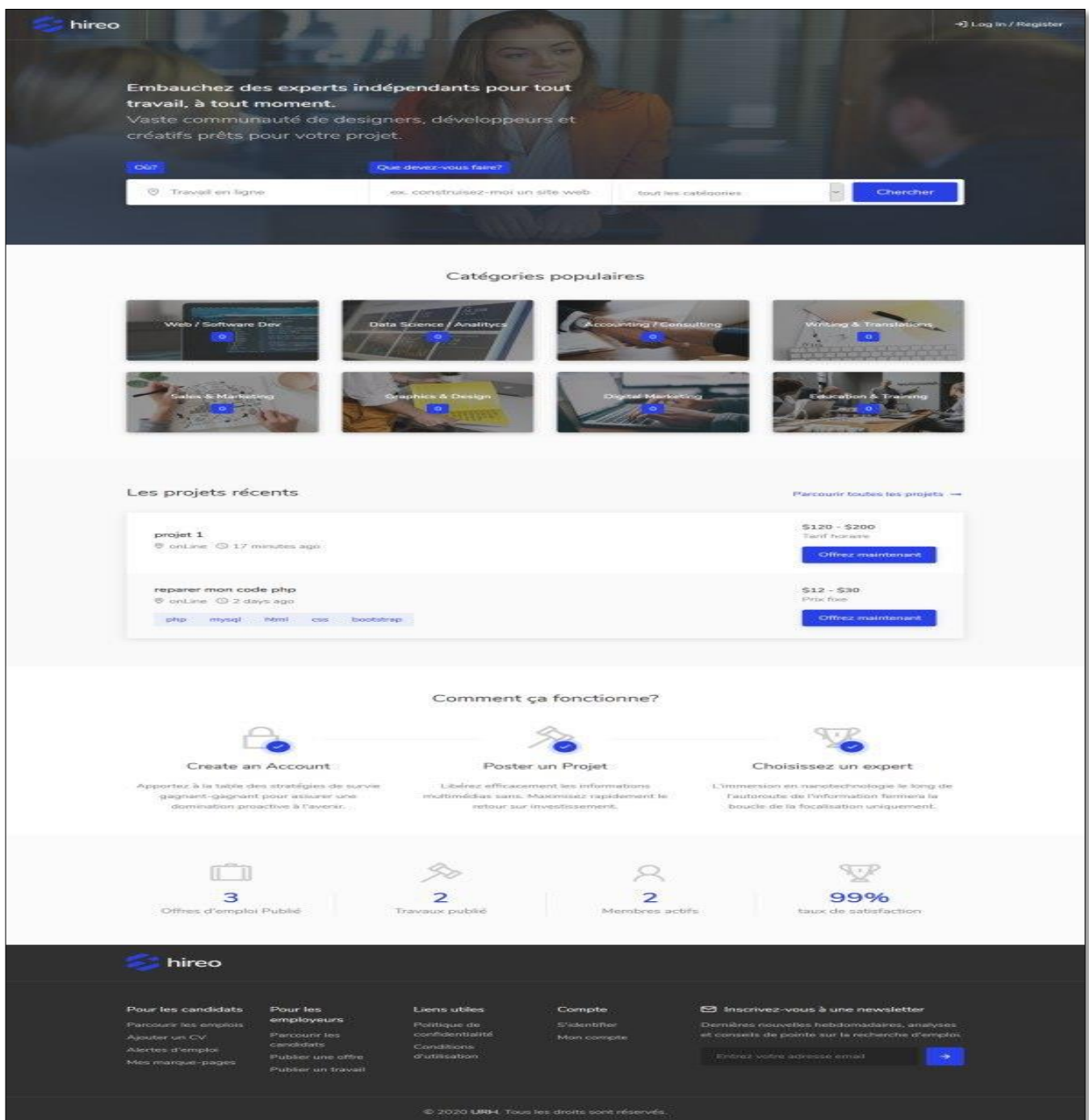


Figure 6:page d'accueil

a. Inscription & connexion :

Pour bénéficier de nos services, le client doit d'abord s'inscrire puis il est obligé à se connecter. Ces deux étapes ont lieu sur les figures suivantes :

Créons votre compte!
Vous avez déjà un compte? [Log In!](#)

étape 1 **étape 2** étape 3 <

☒ Homme ☐ Femme

Suivant

Figure 7: inscription étape 1

Créons votre compte!
Vous avez déjà un compte? [Log In!](#)

étape 1 **étape 2** étape 3 <

Suivant

Figure 8: inscription étape 2



The image shows a registration form titled "Créons votre compte!" with a link "Vous avez déjà un compte? Log In!". The form is divided into three steps: "étape 1", "étape 2", and "étape 3", with "étape 3" being the active step. The form contains three input fields: "Nom d'utilisateur ..." (with a shield icon), "Mot de passe .." (with a lock icon), and "Repeter le mot de passe .." (with a lock icon). A blue "Suivant" button is at the bottom.

Figure 9: inscription étape final



The image shows a login form titled "Nous sommes heureux de vous revoir!". It includes a link "Vous n'avez pas de compte? S'inscrire!". The form has two input fields: "Nom d'utilisateur ..." (with an envelope icon) and "Mot de passe ..." (with a lock icon). Below the fields is a link "Forgot Password?". A blue "Log In" button is at the bottom.

Figure 10: formulaire de connexion

b. Profile du client :

Cette partie de l'application contient les informations du client (image, nom, prénom, email, ...). Ces informations seront visitées par les autres clients.

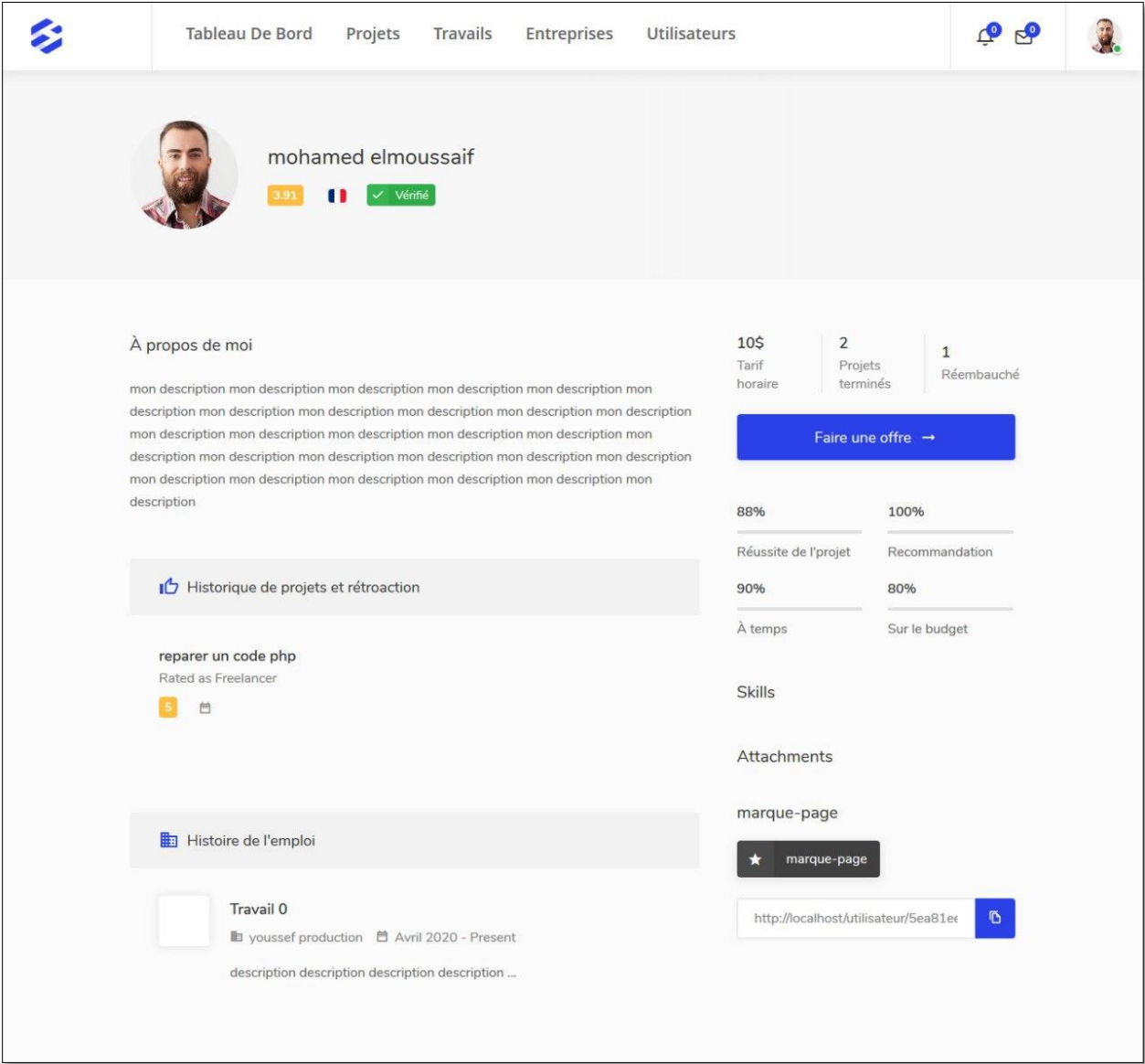


Figure 11: Profile d'utilisateur (freelancer)

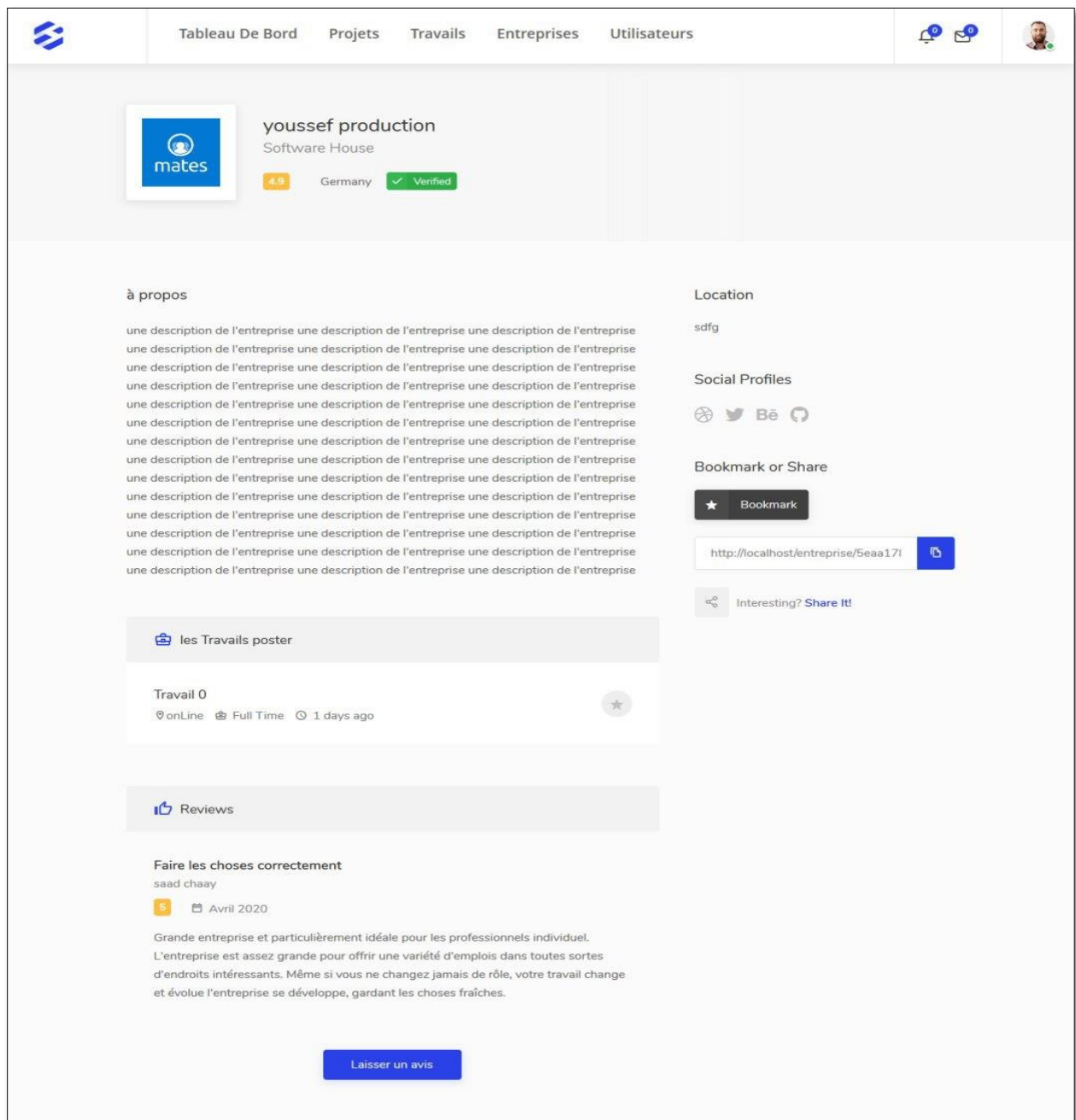


Figure 12: profile d'entreprise

c. Tableau de bord :

Dans cette interfaces le client trouvera les statistiques consternant son compte (les travaux gagnés, les projets en attend, ...). Aussi les notifications et les messages reçu.

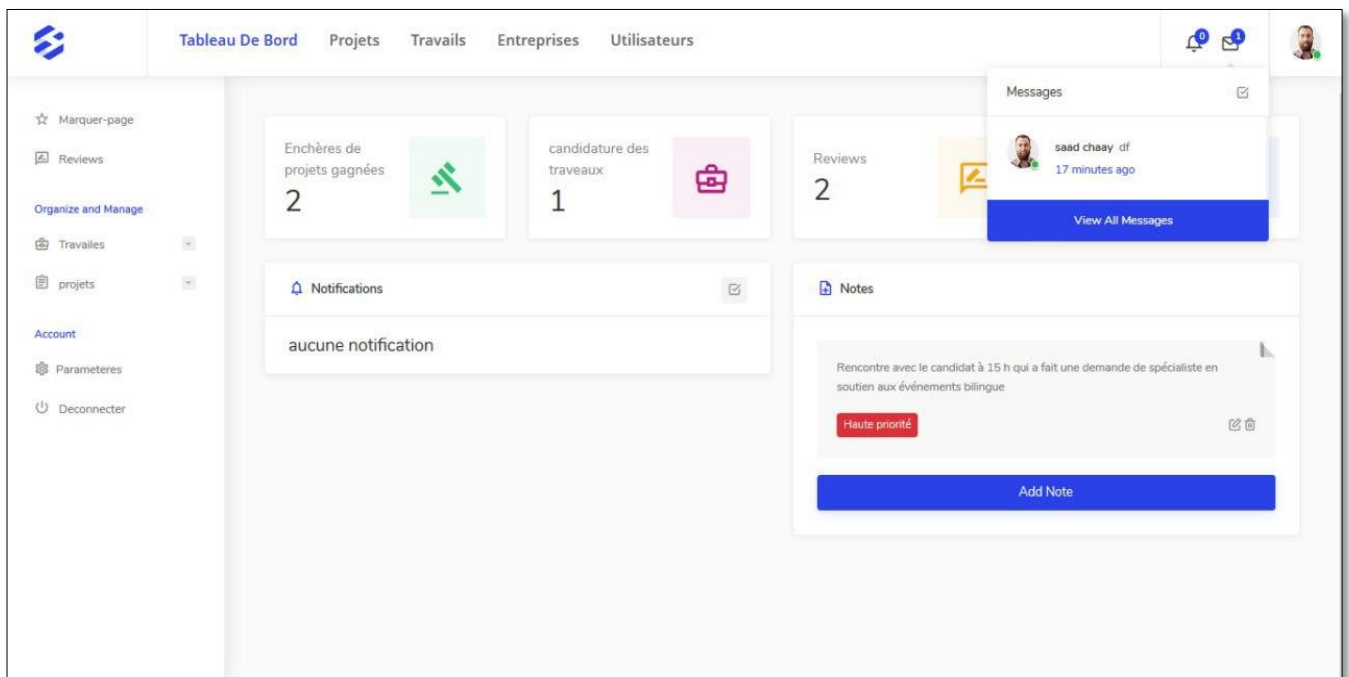


Figure 13: tableau de bord

d. Poster un travail :

Cette interface permet de créer une nouvelle annonce de travail.

Figure 14: poster un nouveau travail

e. Page de travail :

Cette page contient les détails d'une annonce de travail créer avec notre application. Il sera visité par tous les clients.

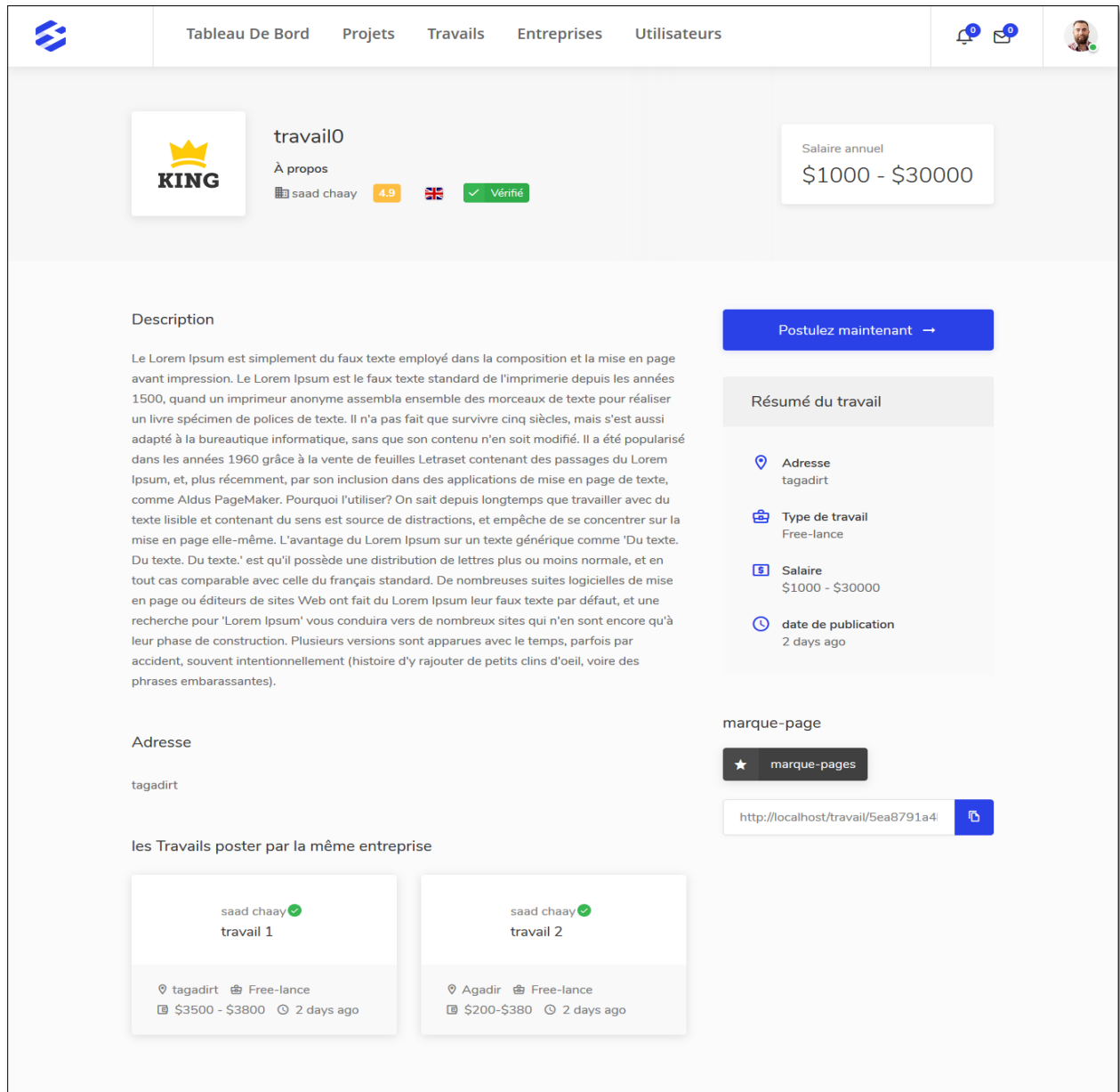


Figure 15: Page de travail (job)

f. Poster un projet :

Cette interface permet de créer une nouvelle annonce de projet.

Figure 16: poster un nouveau projet

g. Page de projet :

Cette page contient les détails d'un projet déjà créer sur notre application. Il sera visité par tous les clients.

h. Gérer les postes :

Notre application contient des interfaces pour manager les annonces poster soit pour les travaux ou les projets. Ils permettent d'accepter/refuser les candidatures reçues ou annuler l'annonce.

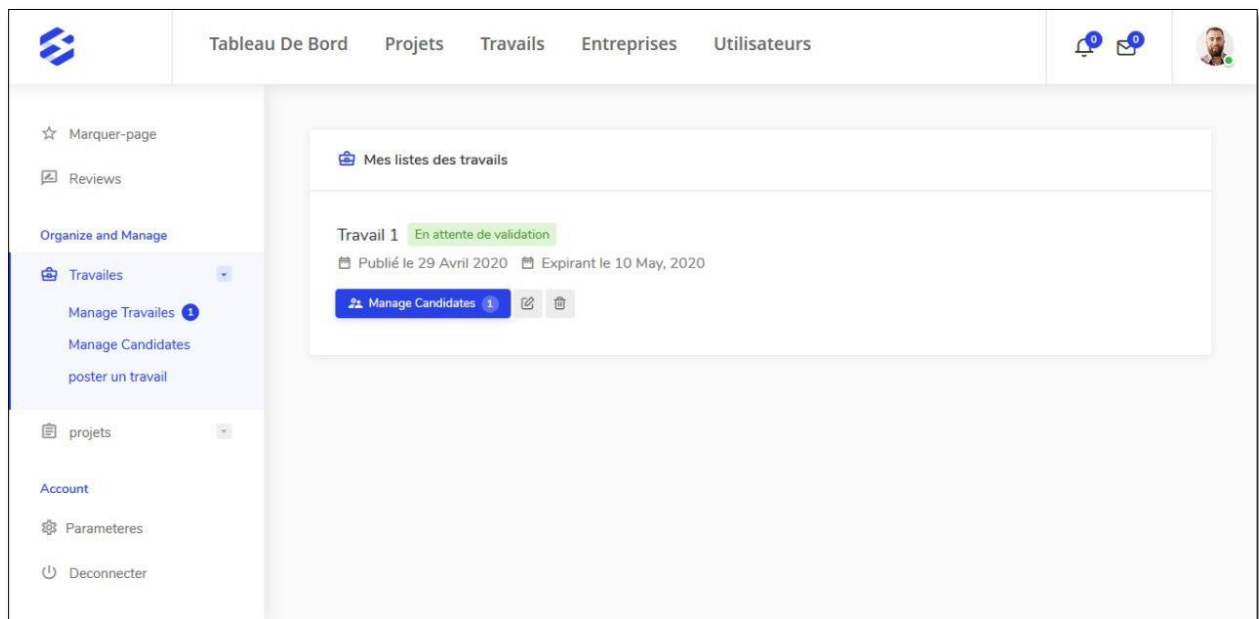


Figure 17: manager les travaux postés

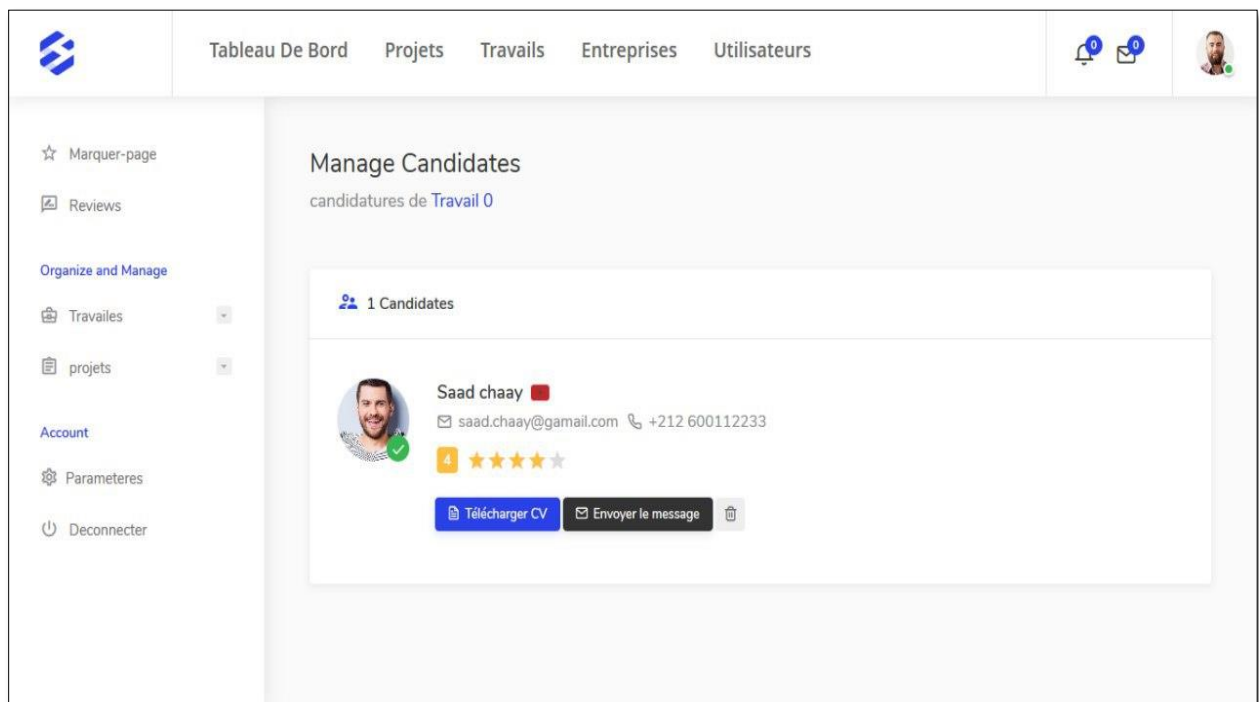


Figure 18: gérer les candidatures reçu

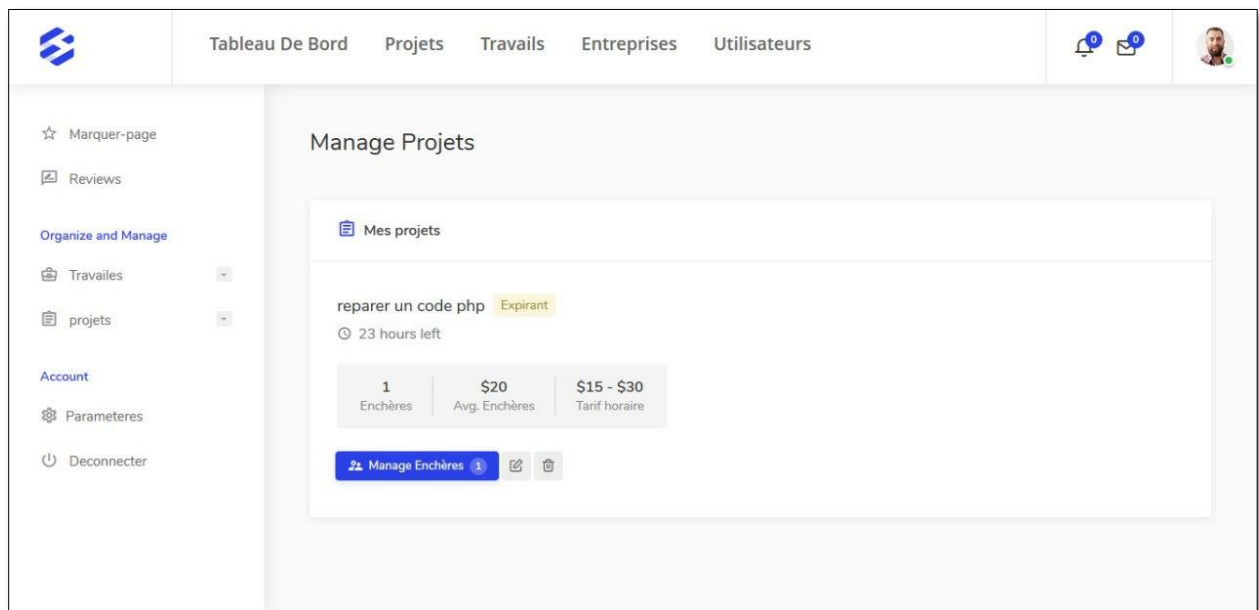


Figure 19: Manager les projets postés

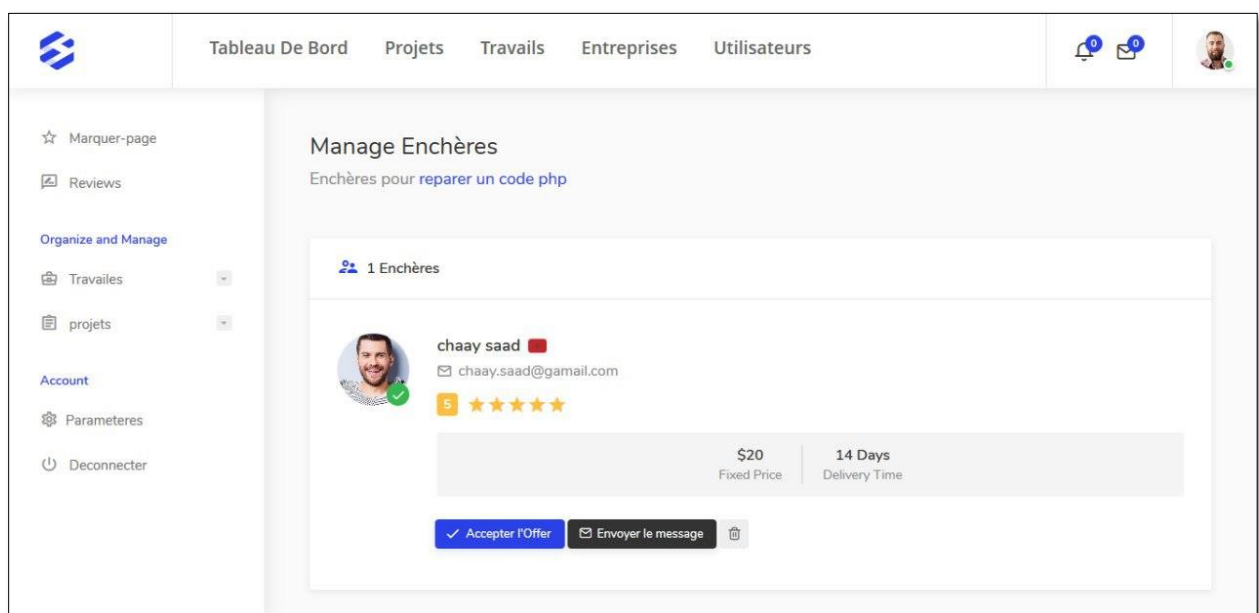


Figure 20: manager les enchère du projet

i. La messagerie :

Notre application contient un espace de contact. Chaque client peut créer des conversations et envoyer des messages aux autre clients

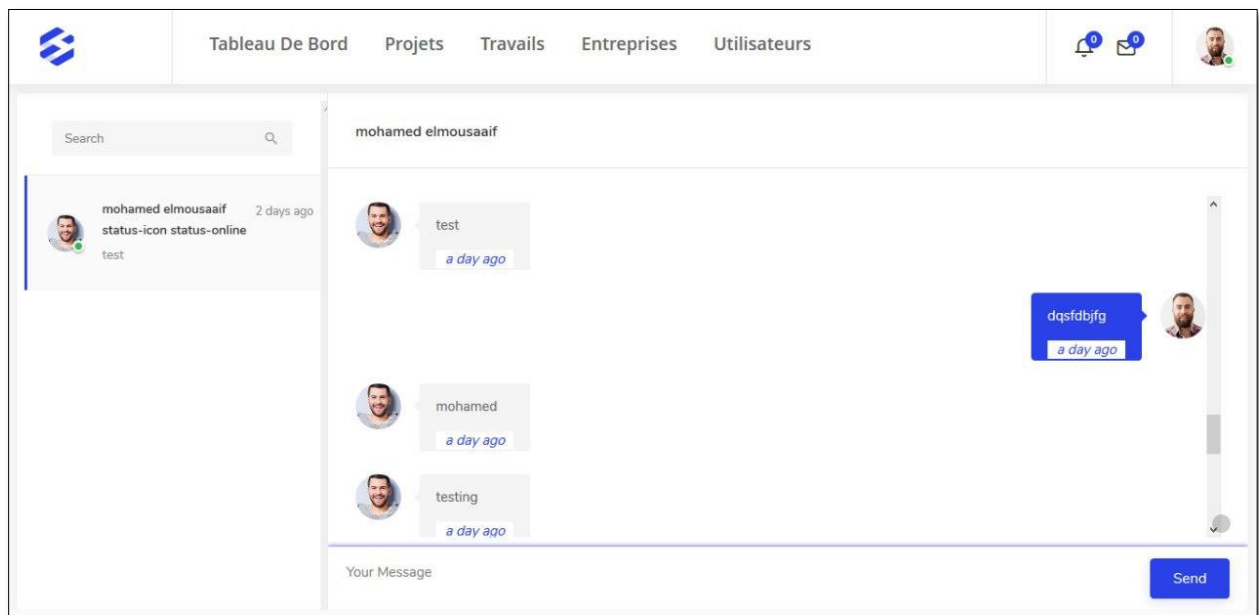


Figure 21: interface de messagerie

a. Les paramètres :

Notre application contient aussi une interface pour gérer le compte et les informations du client en quelques clics.

[illegible]

Figure 22: interface des paramètres

b. Chercher un employant / une entreprise:

Dans cette partie chaque client peut chercher un employant ou une entreprise en filtrant les résultats par un outil de filtrage comme apparaitre dans la figure suivante :

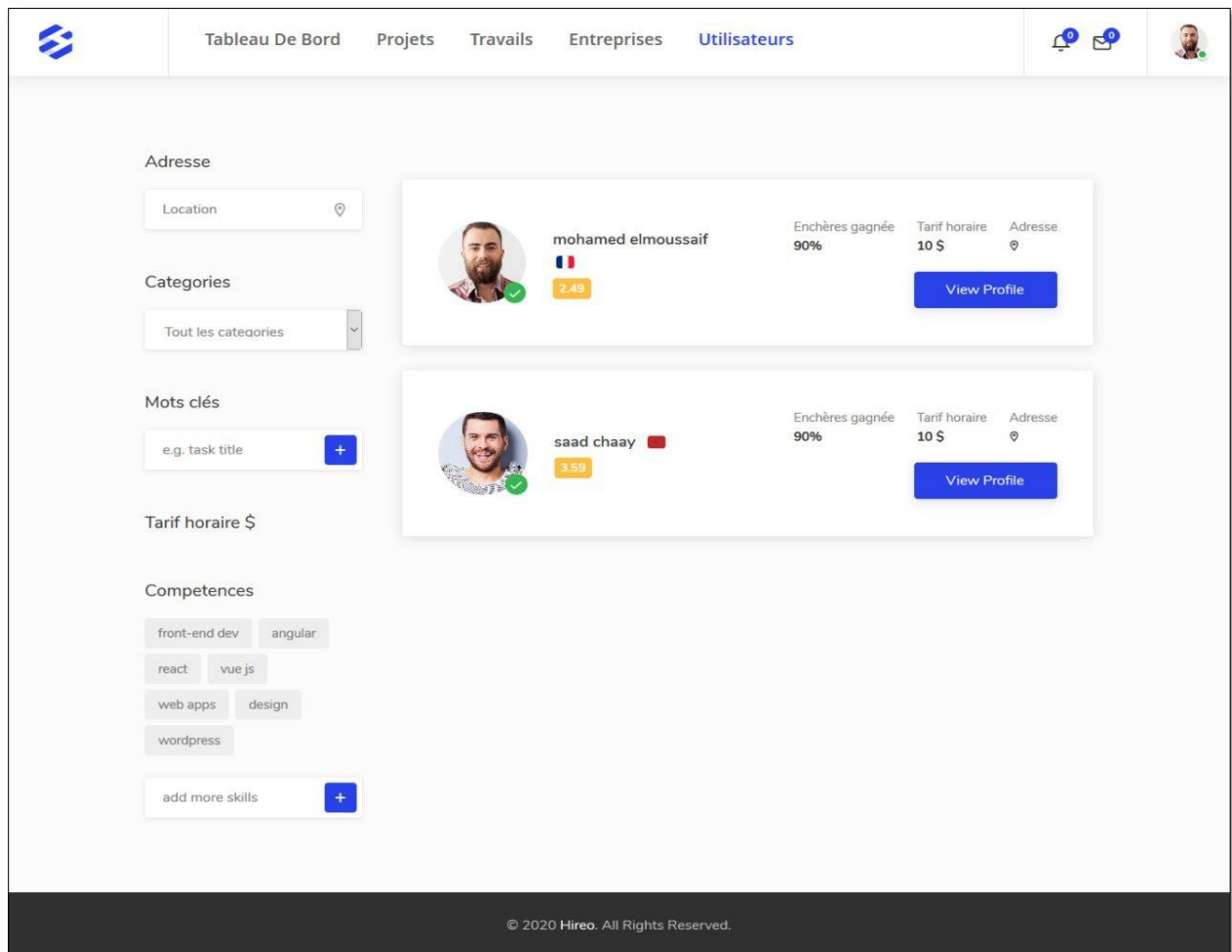


Figure 23: chercher un employant (freelancer)

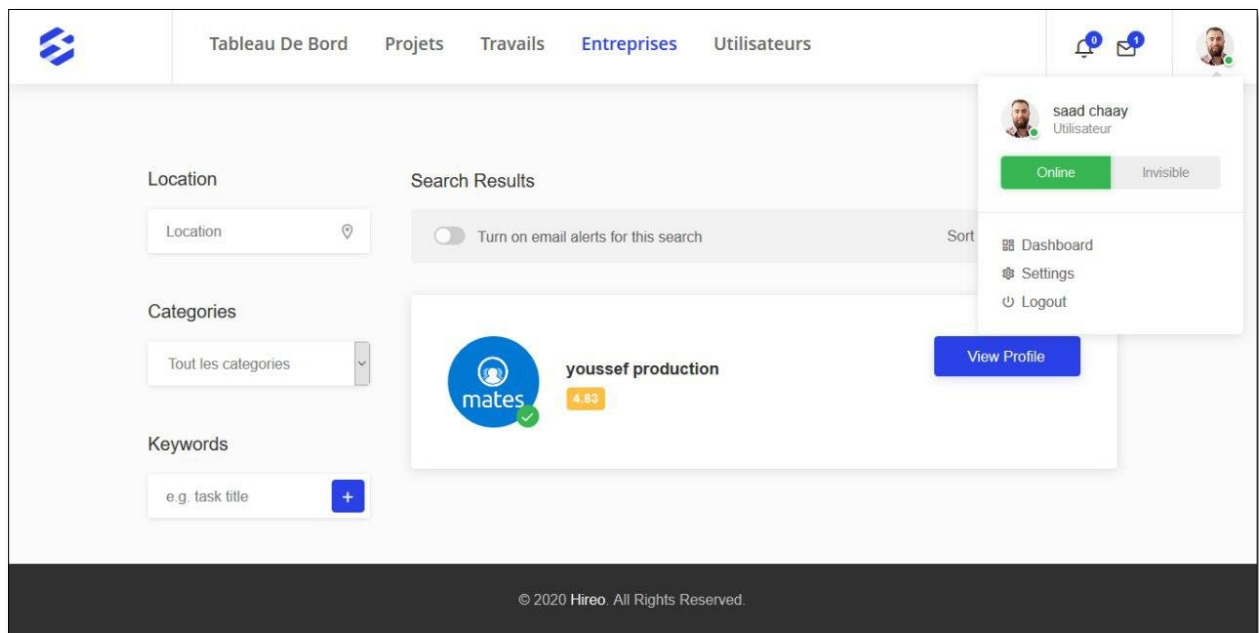


Figure 24: chercher une entreprise

3. Conclusion :

Ce chapitre a donné une vision plus claire sur les composants de notre application et la stratégie que chaque client doit suivre pour l'utiliser correctement, et cela par la présentation de ses interfaces graphiques qui illustrent bien ses diverses fonctionnalités.

VI. Conclusion :

Ce projet de fin d'étude consiste à réaliser un site web dynamique de freelance. Au cours de ce rapport, nous avons présenté les différentes étapes de la conception et la réalisation de notre application.

Afin de satisfaire les besoins des utilisateurs nous avons commencé la conception en utilisant les diagrammes UML et la mise en œuvre des bases de données avec mongoose.js puis en utilisant le gestionnaire de bases de données MongoDB compass. Ensuite la création des types avec l'outil GraphQL et ensuite l'implémentation des requêtes, des abonnements et des mutations en utilisant le javascript

Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation. Nous avons appris à mieux manipuler le langage JavaScript et plusieurs bibliothèques qui sont créées même en javascript. Le plus important que ça, c'est que nous sommes adaptés à travailler en équipe.

En effet, ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêtes à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude. Etant donné que tout travail informatique a été toujours l'œuvre d'une équipe.

VII. Bibliographie :

- <https://stackoverflow.com/>
- <https://mongoosejs.com/>
- <http://www.w3schools.com/>
- <https://graphql.org/learn/>
- <https://www.apollographql.com/docs/>