



**REPUBLIQUE TUNISIENNE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE TUNIS EL MANAR**

**FACULTE DES SCIENCES DE TUNIS
DEPARTEMENT DES SCIENCES DE L'INFORMATIQUE**

**RAPPORT DE
PROJET ACADEMIQUE**

**TITRE
Skill Marketplace**

Réalisé par :

Seif Eddine Jaafer

Housseem Eddine Hamzaoui

Abdelhamid Tanabene

Hind Gharsalli

Encadré par : Faouzi MOUSSA

Organisme d'accueil: FST – DSI

Année Universitaire 2024-2025

Table des matières

1. **Introduction Générale** : p 8
2. **Chapitre 1** : P11
 1. contexte du métier projet : p11
 - Domaine d'application du projet
 - Analyse des besoins et problématiques spécifiques
 2. problématique métier :p11
 - Défis identifiés
 - Objectifs du projet
 3. proposition de valeur:p12
 - Valeur ajoutée pour les parties prenantes
 - Avantages et bénéfices attendus
 4. étude de marché :p12
 - Analyse de l'environnement concurrentiel
 - Tendances et opportunités
 - Menaces potentielles
 5. backlog métier :p13
 - Définition des fonctionnalités
3. **Chapitre 2** : p14
 - 1.description des besoins : p15
 - Contexte
 - Acteurs principaux
 - Problématique
 - Objectifs
 2. Besoins fonctionnels : p16
 - 3.Besoins non fonctionnels : p17
 4. Besoins système : p18
 5. Conclusion : p18
4. **chapitre 3** : p19
 - 1.Introduction à la Conception : p 21
 - Présentation du chapitre
 - Objectifs de la phase de conception
 - 2.Analyse des Besoins : p21
 - Synthèse des besoins du projet identifiés dans les chapitres précédents
 - Priorisation des fonctionnalités en fonction de leur importance
 - 3.Architecture Globale du Système : p23
 - Proposition d'une architecture générale pour la solution

- Utilisation et description de diagrammes de composants et de déploiement
 - 4. Modélisation des Données : p26
 - Conception du schéma de la base de données
 - Utilisation de diagrammes de classes UML pour définir les entités et les relations
 - Spécification des contraintes avec UML
 - 5. Interfaces Utilisateur : p26
 - Esquisses ou maquettes des interfaces utilisateur
 - Explication des choix d'ergonomie et d'expérience utilisateur
 - 6. Algorithmes et Méthodes : p35
 - Description des algorithmes clés utilisés dans la solution
 - Utilisation de diagrammes d'activités UML pour modéliser les processus
 - 7. Sécurité et Confidentialité : p 37
 - Stratégies de sécurité mises en place
 - 8. Scénarios de Cas d'Utilisation :p 38
 - Présentation de scénarios d'utilisation typiques
 - Utilisation de diagrammes de cas d'utilisation UML pour illustrer les interactions fonctionnelles
 - 9. Diagrammes de Séquence : p 41
 - Utilisation de diagrammes de séquence UML pour modéliser les interactions entre les objets du système
 - 10. Plan de Développement : p44
 - Division du projet en modules ou itérations de développement
 - Calendrier de mise en œuvre de la conception
 - 11. Conclusion du Chapitre : p47
 - Récapitulation des principaux éléments de la conception
 - Transition vers les phases suivantes du développement
5. **Chapitre 4 : P49**
1. Introduction à la réalisation : p 51
 - Objectifs de la phase de réalisation :
 - Enjeux
 2. Environnement de Développement : P51
 - Outils utilisés
 - Débat sur les choix
 3. Développement des Modules : p52
 1. Les modules :
 - 1.1-Modules pour l'Administrateur
 - 1.2- Modules pour le Client
 - 1.3- Modules pour le Skill Provider
 - 1.4- Modules pour le Skill Expert
 - 2.Usage UML
 4. Tests et Débogage : p54
 5. Intégration des Composants: p54
 - Mise en œuvre de l'architecture globale du système
 - Coordination des différents modules pour assurer leur interopérabilité
 - 6.Validation et Vérification : p54

- Évaluation de la solution par rapport aux besoins initiaux
- Vérification de la conformité aux spécifications définies dans la conception

7.Optimisation des Performances: p55

- Identification des points d'optimisation éventuels :
- Actions entreprises pour améliorer les performances du système

8.Documentation Technique : p55

- Rédaction d'une documentation détaillée pour faciliter la maintenance
- Utilisation de UML pour compléter la documentation technique

9.Formation et Transfert de Compétences : p56

- Formation des utilisateurs finaux et de l'équipe de maintenance
- Transfert des compétences nécessaires à la gestion du projet

10.Conclusion du Chapitre : p 56

- Bilan de la Réalisation
- Validation du Système

6. Conclusion Générale et Perspectives : p 58

Liste des tableaux

Tableau 1 : Calendrier de la mise en œuvre de la conception

Liste des figures

Figure 1 : Diagramme de Composants
Figure 2 : Diagramme de Déploiement
Figure 3 : Diagramme de Classes de Skill Marketplace
Figure 4 : Interface de connexion
Figure 5 : Interface d'inscription.....
Figure 6 : Interface mot de passe oublié
Figure 7 : Interface de la liste des offres
Figure 8 : Interface des offres de l'utilisateur
Figure 9 : Interface pour la publication d'une offre
Figure 10 :Interface pour les informations pour les informations liées à une offre.....
Figure 11 : Exemple d'un review
Figure 12 : Gestion des comptes faites par l'admin
Figure 13 : Interface pour l'ajout d'un compte (uniquement pour l'admin)
Figure 14 : Gestion des compétences (skills)
Figure 15 : Ajout d'une nouvelle compétence
Figure 16 : Gestion des quiz
Figure 17 : Gestion des questions d'un quiz
Figure 18 : Diagramme d'Activité – Échange Client-Skill Provider
Figure 19 : Diagramme de Cas d'Utilisation
Figure 20 : Diagramme de Séquence – Connexion Utilisateur

Figure 21 : Diagramme de Séquence – Inscription Utilisateur
Figure 22 : Diagramme de Séquence – Réservation et Publication

Introduction Générale

Dans un environnement en constante évolution, les plateformes numériques jouent un rôle clé dans la mise en relation des individus. Qu'il s'agisse de trouver un emploi, de partager des compétences, ou de collaborer sur des projets spécifiques, ces outils sont devenus indispensables pour répondre aux attentes du marché moderne. Cependant, de nombreuses solutions existantes souffrent de limitations en termes d'efficacité, d'accessibilité, ou encore de fiabilité.

*C'est pour répondre à ces défis que le projet "**Skill Marketplace**" a été conçu. Il s'agit d'une plateforme dédiée à la valorisation et à l'échange de compétences, où freelances, entreprises et particuliers peuvent interagir de manière simple et sécurisée. Ce projet ambitionne de devenir une solution efficace pour faciliter la recherche de profils spécialisés et pour permettre à chacun de tirer parti de ses talents dans un environnement structuré.*

Ce rapport documente les différentes étapes de la réalisation de ce projet. Dans un premier temps, nous présenterons les objectifs et le contexte dans lequel le projet s'inscrit. Ensuite, nous détaillerons les spécifications des besoins, suivies par les étapes de conception et de modélisation, avant de conclure par une description approfondie de la réalisation technique. En espérant que ce travail reflète les efforts et l'engagement de notre équipe, nous souhaitons également qu'il puisse servir de base à de futures améliorations dans ce domaine passionnant qu'est la mise en réseau des compétences.

Présentation de l'organisme d'accueil et de l'Encadrant

*Le projet "Skill Marketplace" a été réalisé au sein du **Département des Sciences de l'Informatique** de la **Faculté des Sciences de Tunis (FST)**, relevant de l'**Université Tunis El Manar**. Située à El Manar, Tunis, cette faculté publique joue un rôle clé dans la formation des étudiants en sciences fondamentales et appliquées, notamment dans les domaines de l'informatique, des mathématiques et de la physique. Le département d'informatique est particulièrement actif dans la promotion de projets innovants, intégrant des technologies modernes et répondant aux besoins actuels du marché.*

*L'encadrement du projet a été assuré par **Pr. Faouzi Moussa**, un expert reconnu dans le domaine des systèmes d'information et des technologies web. En tant que membre du corps enseignant de la FST, il contribue activement à la formation des étudiants et à l'encadrement de projets académiques, en mettant l'accent sur l'innovation, la rigueur méthodologique, et la résolution de problématiques réelles.*

Grâce à cet encadrement et à l'environnement académique propice de la FST, ce projet a pu bénéficier des ressources nécessaires pour sa conception et sa réalisation

Chapitre 1 : État de l'Art et Fondements Métier

Le premier chapitre de ce rapport présente une revue complète de la littérature pertinente pour le projet, ainsi qu'une exploration approfondie du contexte métier dans lequel il s'inscrit. Il examine les progrès réalisés dans le domaine concerné tout en soulignant les lacunes et les opportunités non exploitées. Ce chapitre se concentre sur les problématiques clés à résoudre, tout en mettant en avant la proposition de valeur unique du projet. Une analyse de marché est également incluse afin de situer le projet par rapport aux acteurs existants et de mieux comprendre son positionnement concurrentiel

Plan du chapitre :

1- Contexte Métier du Projet

- Présentation du domaine d'application du projet
- Analyse des besoins et des problématiques spécifiques du domaine 2-

Problématique Métier

- Identification des principaux défis ou des lacunes à résoudre dans le domaine
- Définition claire des objectifs visés par le projet 3- Proposition de Valeur
- Mise en évidence de la valeur ajoutée du projet pour les parties prenantes
- Description des avantages et des bénéfices attendus 4- Étude de Marché
- Analyse de l'environnement concurrentiel et des tendances du marché liées au

projet

- Identification des opportunités et des menaces potentielles

5- Backlog Métier

- Définition des fonctionnalités et des exigences métier du projet

1.1 Contexte Métier du Projet

Domaine d'application du projet

Skill Marketplace s'inscrit dans le domaine de l'économie collaborative, où les plateformes numériques favorisent l'échange de compétences et les services en ligne. Ce projet fait le lien entre l'apprentissage en ligne et le travail freelance, offrant aux utilisateurs la possibilité de se former tout en valorisant leurs compétences. Il répond à des enjeux contemporains tels que l'accès simplifié aux services, l'apprentissage personnalisé, la monétisation des compétences et une flexibilité adaptée aux besoins variés d'une communauté en constante évolution.

Analyse des besoins et problématiques spécifiques

Le projet émerge d'un constat simple : le secteur actuel peine à combler certains besoins cruciaux. Parmi eux :

- Opportunités pour les freelances : Manque de visibilité et difficulté à se faire connaître et à valoriser leurs compétences dans un environnement compétitif.
- Sécurité des échanges : Manque de plateformes garantissant des interactions fiables entre ceux qui offrent et ceux qui recherchent des compétences.
- Apprentissage innovant : Nécessité de proposer des méthodes éducatives flexibles et collaboratives, adaptées aux exigences actuelles.

1.2 Problématique Métier

Défis identifiés

- Les freelances sont confrontés à un marché saturé, où il est difficile de se démarquer et de mettre en valeur leurs talents.
- Les apprenants rencontrent des obstacles pour trouver des opportunités d'apprentissage adaptées à leurs besoins.
- Les plateformes actuelles manquent d'outils permettant une interaction fluide, un suivi structuré des compétences, et une évaluation fiable des services.

Objectifs du projet

Pour répondre à ces défis, Skill Marketplace vise à :

- Offrir une plateforme intuitive et sécurisée qui connecte d'une part les formateurs « skill expert » et les apprenants et d'autre part ceux qui cherchent un service IT précis et ceux qui l'offrent les « skill provider ».
- Promouvoir une économie collaborative en mettant en avant les compétences individuelles.
- Proposer un système d'échange flexible et abordable, basé notamment sur un système de points innovant.

1.3 Proposition de Valeur

Valeur ajoutée pour les parties prenantes

Skill Marketplace se distingue par sa capacité à combiner économie collaborative et apprentissage personnalisé.

- Pour les « skill providers » et les « skill experts » : Plus de visibilité, des opportunités rémunératrices, et un système de points valorisant leurs compétences.
- Pour les apprenants : Une offre de formation flexible, accessible, et adaptée à leurs attentes, avec un accès direct à des experts.

Avantages et bénéfices attendus

- Création d'une communauté interactive et dynamique, favorisant l'échange et la croissance mutuelle.
- Mise en place d'un modèle hybride de monétisation et d'échange, stimulant un apprentissage accessible et motivant.
- Valorisation des talents et développement des compétences dans un environnement sécurisé.

1.4 Étude de Marché

Analyse de l'environnement concurrentiel

Il existe des plateformes qui dominent le marché actuel, on peut citer Upwork et Fiverr mais le problème de ces deux dernières réside dans la concentration sur des projets individuels ponctuels et non sur l'échange bidirectionnel entre les individus c'est dans ce cadre que vient Skill Marketplace offrant une combinaison entre apprentissage et échange de services à long terme.

Tendances et opportunités

- Une montée en exponentielle des modèles d'économie collaborative dans divers secteurs.
- Une demande croissante pour des plateformes localisées et personnalisées.

Menaces potentielles

- Une concurrence avec des plateformes déjà établies et reconnues mondialement.
- La nécessité de constituer une base d'utilisateurs pour garantir la viabilité du projet.

1.5 Backlog Métier

Définition des fonctionnalités

Le backlog métier comprend les éléments essentiels suivants :

- Inscription et gestion des profils : Création et personnalisation des profils utilisateurs (skill provider, skill expert et apprenants).
- Publication et recherche de compétences : Un moteur de recherche avancé pour filtrer les services ou les compétences proposés et ainsi donner la possibilité de réaliser un choix spécifique.
- Système de points et de monétisation : Échange de services contre des points ou de l'argent.
- Notation et avis : Promotion de la transparence et de la fiabilité grâce à un système de notation.
- Transactions sécurisées : Environnement sûr pour les paiements et les échanges.

Chapitre 2 : Spécification des besoins

Pour le projet *Skill Marketplace*, les besoins sont classifiés en trois catégories principales : fonctionnels, non fonctionnels, et systèmes, permettant de structurer et de prioriser les fonctionnalités et les spécifications techniques nécessaires au bon fonctionnement de la plateforme.

Plan du chapitre :

- 2.1- Description des besoins
- 2.2- Besoins Fonctionnels
- 2.3- Besoins Non Fonctionnels
- 2.4- Besoins Système
- 2.5- Conclusion

2.1 Description des besoins

Dans le cadre de notre projet Skill Marketplace, la spécification des besoins est une étape cruciale qui vise à déterminer les attentes d'une part et les exigences d'autre part des différents acteurs de notre système.

Ce chapitre souligne les besoins des différents acteurs du système en tenant compte des objectifs du projet.

1. Contexte :

Le projet Skill Marketplace vise à mettre en relation des individus ayant des compétences spécifiques avec des clients ou entreprises recherchant ces compétences. Le système doit donc permettre aux utilisateurs de publier des offres, rechercher des profils cibles et faciliter la communication entre les différents acteurs.

2. Acteurs principaux :

- Skill-Provider : propose des compétences et cherche des opportunités de collaboration notamment des échanges avec d'autres Skill-Providers.
- Skill-Expert : publie des cours en ligne et proposer des services (réunions one on one)
- Client : Cherche des talents spécifiques pour des projets ou des tâches.
- Administrateur : responsable de la gestion du système (utilisateurs , profils , transactions , quiz , évaluations ...)

3. Problématique

Les acteurs rencontrent des difficultés à trouver une plateforme simple, rapide, et efficace pour :

- Identifier les bonnes opportunités correspondant à leurs compétences ou besoins.
- Évaluer la fiabilité et la qualité des profils ou des offres.
- Simplifier les processus de communication et de négociation.

4. Objectifs

Le système doit donc répondre aux objectifs suivants :

- Fournir une interface intuitive et facile d'utilisation.
- Garantir la fiabilité des informations et la sécurité des échanges.
- Permettre un filtrage précis et des recherches avancées pour correspondre rapidement aux besoins spécifiques.
- Faciliter la communication entre les différents acteurs du système.

2.2. Besoins Fonctionnels

1. Gestion des utilisateurs

- Gestion des inscription et connexion des utilisateurs (Skill-Provider, Skill-expert, Client).
- Création et modification de profils incluant des informations personnelles, compétences, expériences.

2. Connexion

- L'inscription si le compte n'existe pas
- L'authentification si le compte existe déjà

3. Gestion des offres

- Valider ou Refuser une offre selon la structure

4. Gestion des compétences

- Proposition d'un test de niveau pour chaque compétence sélectionnée par un skill provider

5. Recherche et filtrage

- Moteur de recherche pour identifier des talents.
- Filtrage par compétence et par nombre de connexions effectuées.

6. Publication des services

- Les Skill-Providers peuvent publier des offres avec des descriptions incluant les informations nécessaires.

7. Système de messagerie

- Messagerie interne pour faciliter les communications entre utilisateurs.
- Notifications pour les nouveaux messages ou activités importantes.

8. Gestion des transactions financières

- Paiements sécurisés via la plateforme.

9. Évaluations et retours

- Les utilisateurs peuvent laisser des évaluations et des commentaires après chaque service.

10. Realiser des meet one to one

- Les skill experts peuvent offrir des réunions particulière : meet one to one à des clients ou des skill providers .

11. Demander des services

- Les skill expert ainsi que les clients peuvent profiter de ce service en effectuant un accord avec le skill provider

12. Gestion des cours

- Les skill expert peuvent proposer des cours
- Les clients peuvent participer à ces cours

2.3. Besoins Non Fonctionnels

1. Performance

- Temps de réponse acceptable pour les actions effectuées.
- Capacité à gérer plusieurs utilisateurs actifs simultanément.

2. Sécurité

- Protection des données personnelles et professionnelles des utilisateurs.
- Sécurisation des transactions financières.
- Sécurisation dans le processus de l'envoi des mails en utilisant le protocole ssl.

3. Convivialité

- Interface intuitive avec un design axé sur l'expérience utilisateur.

4. Évolutivité

- Conception modulaire pour permettre l'ajout de nouvelles fonctionnalités ou l'amélioration future.

2.4. Besoins Systèmes

1. Infrastructure

- Base de données non relationnelle (Mongo DB) pour gérer les utilisateurs, les offres, et les transactions.
- **Technologies**
 - Backend développé avec un framework robuste (par ex. : Node.js).
 - Frontend utilisant des technologies modernes comme React pour l'interactivité.
- **Sécurité système**
 - Authentification via JWT.
 - BCrypt pour le chiffrement des mots de passe.
- **Intégrations tierces**
 - Passerelles de paiement : Stripe pour les transactions financières.
 - Intégration d'API pour les notifications et les envois d'email.

2.5. Conclusion

Ces besoins fournissent une vision claire et complète pour guider le développement de la plateforme tout en assurant qu'elle répondra aux attentes des utilisateurs et aux exigences techniques.

Chapitre 3 : Conception de la Solution Informatique

Le chapitre 3 se concentre sur la conception du système, en détaillant l'architecture globale et les principaux modules fonctionnels. Il aborde la modélisation des données, la sécurité, ainsi que les stratégies de tests pour garantir la qualité du système. Ce chapitre définit également les étapes suivantes du développement, y compris l'intégration, le déploiement et les améliorations futures. L'objectif est d'assurer une structure solide et évolutive pour le bon fonctionnement de l'application.

Plan du chapitre :

1- Introduction à la Conception

- Brève présentation du chapitre
- Objectifs de la phase de conception

2- Analyse des Besoins

- Synthèse des besoins du projet identifiés dans les chapitres précédents
- Priorisation des fonctionnalités en fonction de leur importance

3- Architecture Globale du Système

- Proposition d'une architecture générale pour la solution
- Utilisation et description de diagrammes de composants et de déploiement

4- Modélisation des Données

- Conception du schéma de la base de données
- Utilisation de diagrammes de classes UML pour définir les entités et les relations
- Spécification des contraintes avec UML

5- Interfaces Utilisateur

- Esquisses ou maquettes des interfaces utilisateur
- Explication des choix d'ergonomie et d'expérience utilisateur

6- Algorithmes et Méthodes

- Description des algorithmes clés utilisés dans la solution
- Utilisation de diagrammes d'activités UML pour modéliser les processus

7- Sécurité et Confidentialité

Stratégies de sécurité mises en place

8- Scénarios de Cas d'Utilisation

- Présentation de scénarios d'utilisation typiques
- Utilisation de diagrammes de cas d'utilisation UML pour illustrer les interactions fonctionnelles

9- Diagrammes de Séquence

Utilisation de diagrammes de séquence UML pour modéliser les interactions entre les objets du système

10- Plan de Développement

- Division du projet en modules ou itérations de développement
- Calendrier de mise en œuvre de la conception

11- Conclusion du Chapitre

- Récapitulation des principaux éléments de la conception
- Transition vers les phases suivantes du développement

3.1 Introduction à la Conception

Présentation du chapitre

Ce chapitre présente la structure et les bases nécessaires pour la mise en œuvre du projet Skill Marketplace. Il couvre les étapes de la conception, incluant l'analyse des besoins, la modélisation de l'architecture du système, la gestion des données, la sécurité, et la conception des interfaces utilisateur. L'objectif est de garantir une solution technique robuste, évolutive et centrée sur l'utilisateur, tout en respectant les exigences fonctionnelles et non fonctionnelles identifiées.

Objectifs de la phase de conception

- Proposer une architecture technique robuste et extensible.
- Garantir une expérience utilisateur intuitive et efficace.
- Assurer la sécurité et la performance du système.

3.2 Analyse des besoins

Synthèse des besoins fonctionnels

La plateforme **Skill Marketplace** intègre diverses propriétés essentielles pour combler les besoins de ses utilisateurs. Elle propose une gestion complète des utilisateurs, incluant l'enregistrement, la connexion et la personnalisation de profils. On pourrait confirmer ou refuser les offres, tandis que l'évaluation des compétences s'effectue par le moyen de tests de niveau particuliers. Un moteur de recherche avancé et des filtres précis autorisent d'identifier rapidement les talents. Les utilisateurs ont la possibilité de diffuser des services détaillés, d'interagir via une messagerie intérieure équipée de notifications et de réaliser des opérations financières en pleine sûreté. La transparence est assurée par un système d'évaluations, tandis que la possibilité de rencontres en individu et la gestion des cours favorisent l'interaction et les chances de collaboration.

Synthèse des besoins non fonctionnels

La plateforme **Skill Marketplace** est conçue pour offrir des performances optimales, permettant la gestion simultanée de nombreux utilisateurs actifs tout en garantissant des temps de réponse rapides. Elle assure une sécurité renforcée pour protéger les données personnelles et professionnelles des utilisateurs, ainsi que la confidentialité des transactions financières. L'interface utilisateur intuitive, combinée à un design ergonomique, garantit une expérience conviviale et fluide. De plus, son architecture modulaire favorise une évolutivité permettant l'ajout de nouvelles fonctionnalités ou des améliorations futures en toute simplicité.

Synthèse des besoins systèmes

La solution **Skill Marketplace** repose sur une architecture technique moderne et performante. La base de données MongoDB offre une gestion flexible et évolutive des informations, adaptée aux besoins variés des utilisateurs. Le backend est développé en Node.js, garantissant robustesse et efficacité, tandis que le frontend utilise React pour proposer une interface interactive et dynamique. Côté sécurité, l'authentification est assurée par JWT et les mots de passe sont protégés par bcrypt. Enfin, des intégrations tierces, telles que Stripe pour les paiements sécurisés et des API dédiées pour les notifications et l'envoi d'e-mails, viennent compléter la plateforme pour offrir une expérience utilisateur fluide et fiable.

Priorisation des fonctionnalités en fonction de leur importance

Les fonctionnalités sont priorisées en fonction de leur importance :

Critiques : Gestion des utilisateurs, connexions, publication des services, messagerie interne, transactions financières.

Importantes : Recherche et filtrage, évaluations, gestion des compétences.

Optionnelles : Meet one-to-one, gestion des cours, demandes de services.

3.3 Architecture Globale du Système

Proposition d'une architecture générale pour la solution

L'architecture de la plateforme suit une approche modulaire et évolutive se basant sur l'architecture 3 tiers:

- Backend : Développé en **Node.js**, il gère les services principaux comme l'authentification, la gestion des données et les transactions.
- Frontend : Construit avec **React**, il offre une interface utilisateur interactive et dynamique.
- Base de données : **MongoDB** est utilisée pour une gestion flexible des données non structurées.
- Intégrations tierces : **Stripe** pour les paiements sécurisés et des API pour les notifications et l'envoi d'e-mails.

Utilisation de diagrammes de composants et de déploiement

- **Diagramme de composants**

Le diagramme de composants illustre la structure modulaire de notre application en mettant en évidence les principaux blocs fonctionnels et leurs interactions.

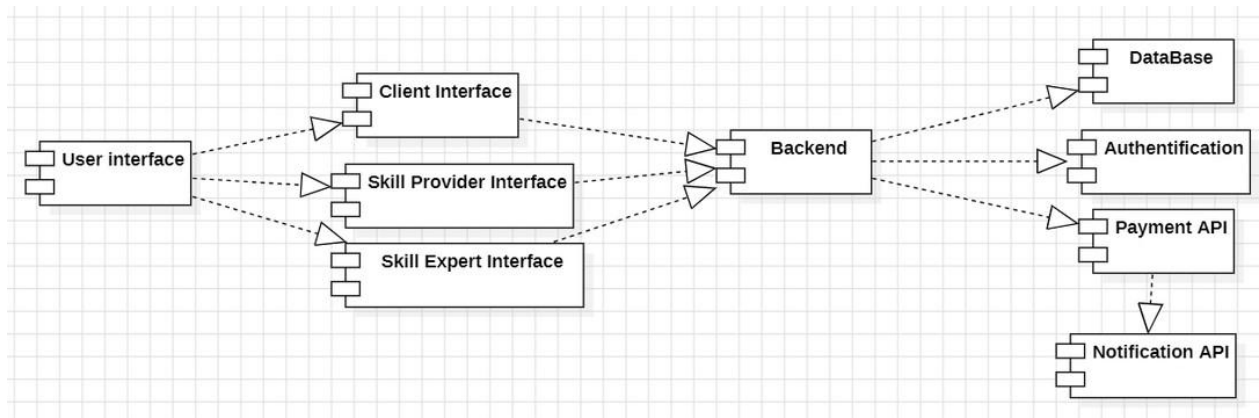


Figure 1 Diagramme de Composants

Le diagramme de composants détaille l'architecture logicielle de la solution, organisée en modules interconnectés pour assurer la maintenabilité et l'évolutivité.

- **User Interface (Interface Utilisateur)** : Permet aux utilisateurs (clients, Skill Providers, et Skill Experts) d'interagir avec la plateforme via une interface conviviale.

- **Client Interface** : Une interface pour gérer les actions spécifiques aux clients, telles que la recherche et le filtrage de talents ou la demande de services.
- **Skill Provider Interface** : Interface dédiée aux Skill Providers pour la publication des offres, la gestion des compétences et la communication avec les clients.
- **Skill Expert Interface** : Permet aux Skill Experts de proposer des cours, d'organiser des réunions privées.
- **Backend** : Géré par **Node.js**, il centralise la logique métier et coordonne les requêtes des interfaces utilisateur avec les autres composants.
- **Base de Données** : Utilisation de **MongoDB** pour stocker des informations sur les utilisateurs, les compétences, les offres et les transactions.
- **Composant d'authentification** : Sécurise l'accès à la plateforme avec **JWT** pour la gestion des sessions et **Bcrypt** pour le chiffrement des mots de passe.
- **Composant du paiement** : Intégré via l'API Stripe, ce composant gère les paiements sécurisés entre utilisateurs.
- **Composant de Notifications** : Envoie des notifications en temps réel pour des événements importants (messages, mises à jour de service).
- **Diagramme de Déploiement**

Le diagramme de déploiement montre comment les différents composants sont physiquement distribués sur les serveurs et les machines dans l'infrastructure

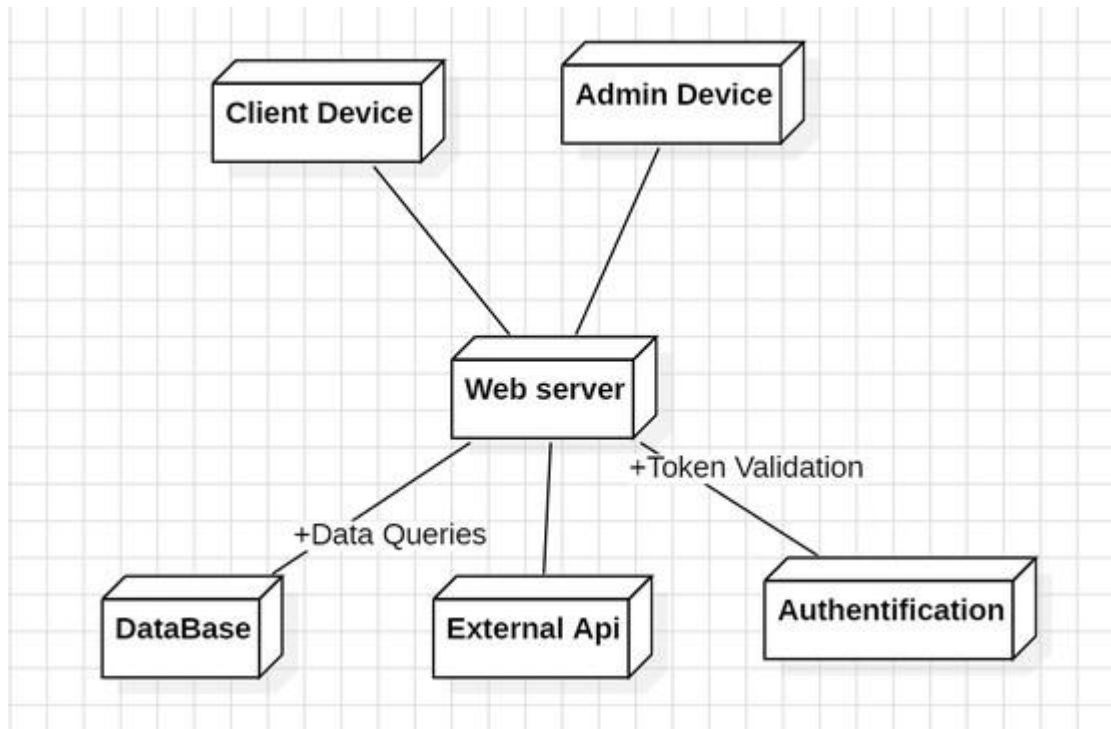


Figure 2 Diagramme de Déploiement

- **Frontend Server** : Héberge l'application React pour l'interface utilisateur.
- **Backend Server** : Héberge l'application Node.js, qui gère la logique métier et communique avec les autres services.
- **Database Server** : Contient la base de données MongoDB.
- **Services Externes** : Inclut les services externes tels que Stripe pour les paiements sécurisés et les API pour les notifications.

Interactions

- Les utilisateurs accèdent à la plateforme via leur navigateur ou application mobile (hébergée sur le Frontend Server).
- Les requêtes des utilisateurs sont envoyées au Backend Server, qui traite la logique métier.
- Le Backend Server interagit avec la Database Server pour récupérer ou stocker des données.
- Pour les paiements, le Backend Server communique avec Stripe via une API sécurisée.
- Les notifications importantes sont envoyées en temps réel par le Backend Server à l'interface utilisateur via les services de notification.

3.4. Modélisation des Données

La modélisation des données constitue une étape essentielle dans la conception de la solution, garantissant une structure solide et optimisée pour le stockage et la gestion des informations.

- **Conception du Schéma de la Base de Données**

La base de données de la plateforme est conçue pour répondre aux besoins spécifiques des différentes fonctionnalités. Une base de données **NoSQL**, **MongoDB**, est utilisée pour sa flexibilité et son évolutivité.

Les principales collections incluent :

- **Users** : Contient les informations des utilisateurs (clients, Skill Providers, Skill Experts), y compris leurs profils, compétences, expériences et rôles.
- **Skills** : Regroupe les compétences disponibles sur la plateforme avec des niveaux associés.
- **Services** : Stocke les détails des services proposés, leurs descriptions, et leurs conditions.
- **Transactions** : Suit les paiements effectués via la plateforme pour les services rendus.
- **Messages** : Archive les échanges dans la messagerie interne pour assurer la traçabilité des communications.

Utilisation de Diagrammes de Classes UML

Un diagramme de classes UML est utilisé pour modéliser les entités principales et leurs relations.

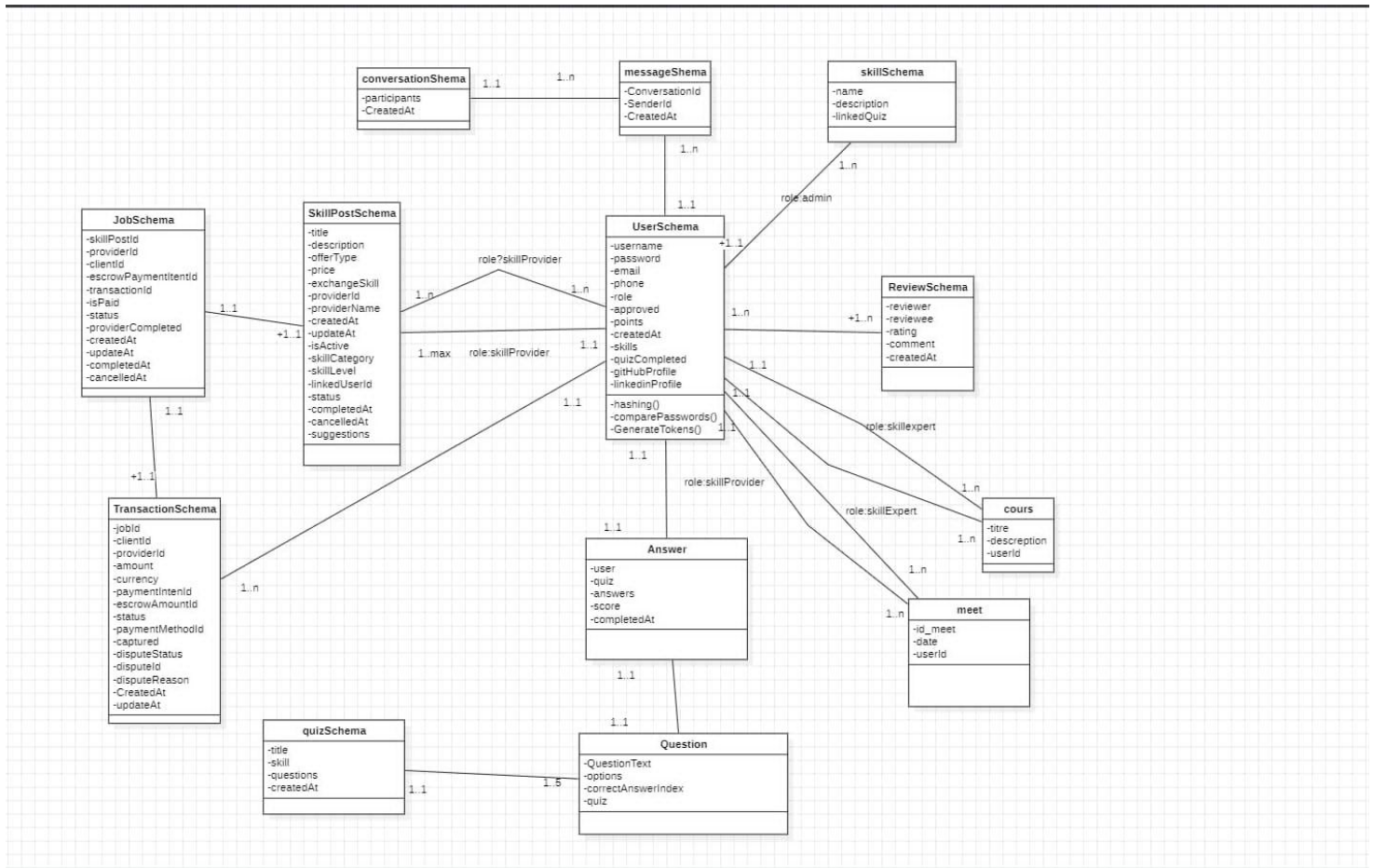


Figure 3 Diagramme de classes de Skill Marketplace

Entités principales

User :

- Gère les utilisateurs avec des attributs tels que username, email, role (ex. skillProvider, skillExpert) et leurs compétences (skills).
- Fournit des fonctions pour sécuriser les données, telles que le hachage des mots de passe et la génération de tokens.

Skill :

- Définit une compétence par son name, sa description, et son niveau (skillLevel).
- Peut être lié à des quizz pour valider les connaissances des utilisateurs.

SkillPost :

- Représente des offres ou demandes de compétences, associées à un utilisateur, avec des informations comme le price, status, et le provider.

Job et Transaction :**Job :**

- Gère les prestations entre un client et un fournisseur de compétence, incluant le suivi des paiements et des statuts.

Transaction :

- Suivi des paiements, avec des détails tels que amount, currency, et les identifiants du client et du fournisseur.

Message et Conversation :

- Permettent aux utilisateurs de communiquer au sein d'un canal sécurisé (ConversationSchema).

Review :

- Système d'évaluation où les utilisateurs peuvent donner une note et un commentaire aux prestataires.

Quiz et Answer :

- Valident les compétences grâce à des quizz liés à des compétences spécifiques.

Spécification des contraintes avec UML**Relations clés**

- Un utilisateur peut posséder plusieurs compétences et proposer plusieurs services.
- Un service (SkillPost) est offert par un utilisateur, tandis qu'un job relie un client à un fournisseur de compétences.
- Les évaluations et quizz permettent d'assurer la qualité des services et la validation des compétences.
- Les transactions garantissent la gestion sécurisée des paiements. Contraintes définies

Cardinalité :

- Un utilisateur peut proposer plusieurs services, mais un service appartient à un seul utilisateur.

- Une compétence est liée à un seul utilisateur, mais celui-ci peut en avoir plusieurs.

Unicité :

- Les emails et identifiants des utilisateurs doivent être uniques.
- Validation et sécurité : Les mots de passe sont hashés via des algorithmes sécurisés comme bcrypt.
- Les transactions nécessitent des montants et identifiants vérifiés.

3.5 Interfaces Utilisateur

Esquisses ou maquettes des interfaces utilisateur

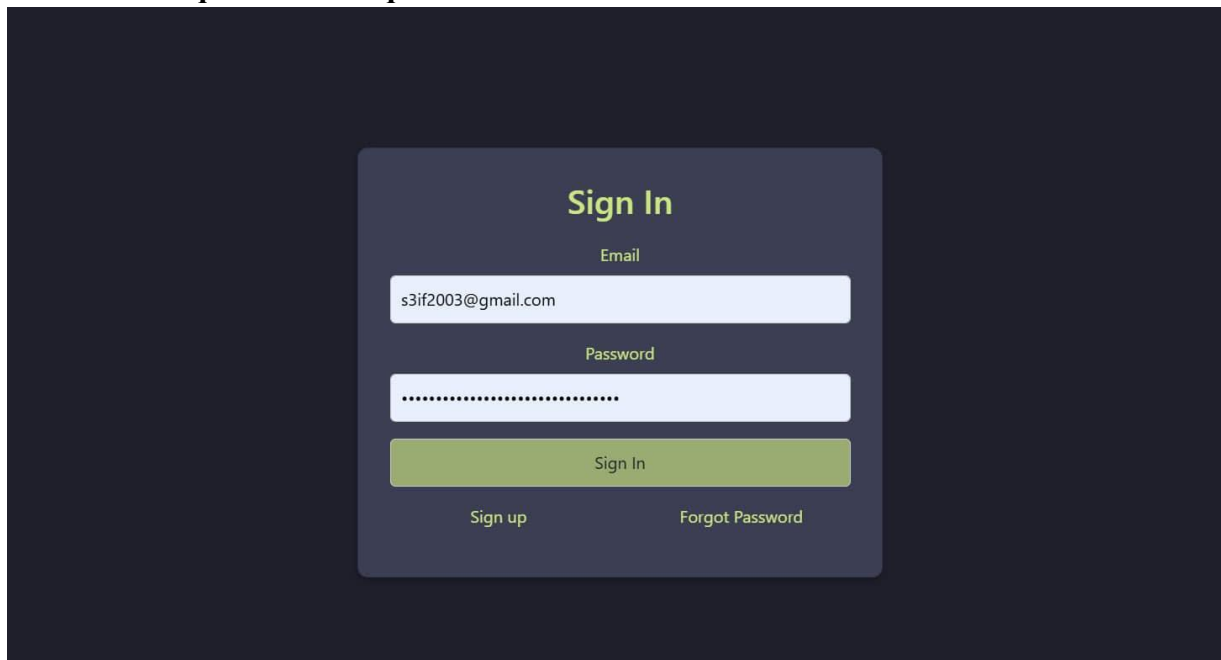
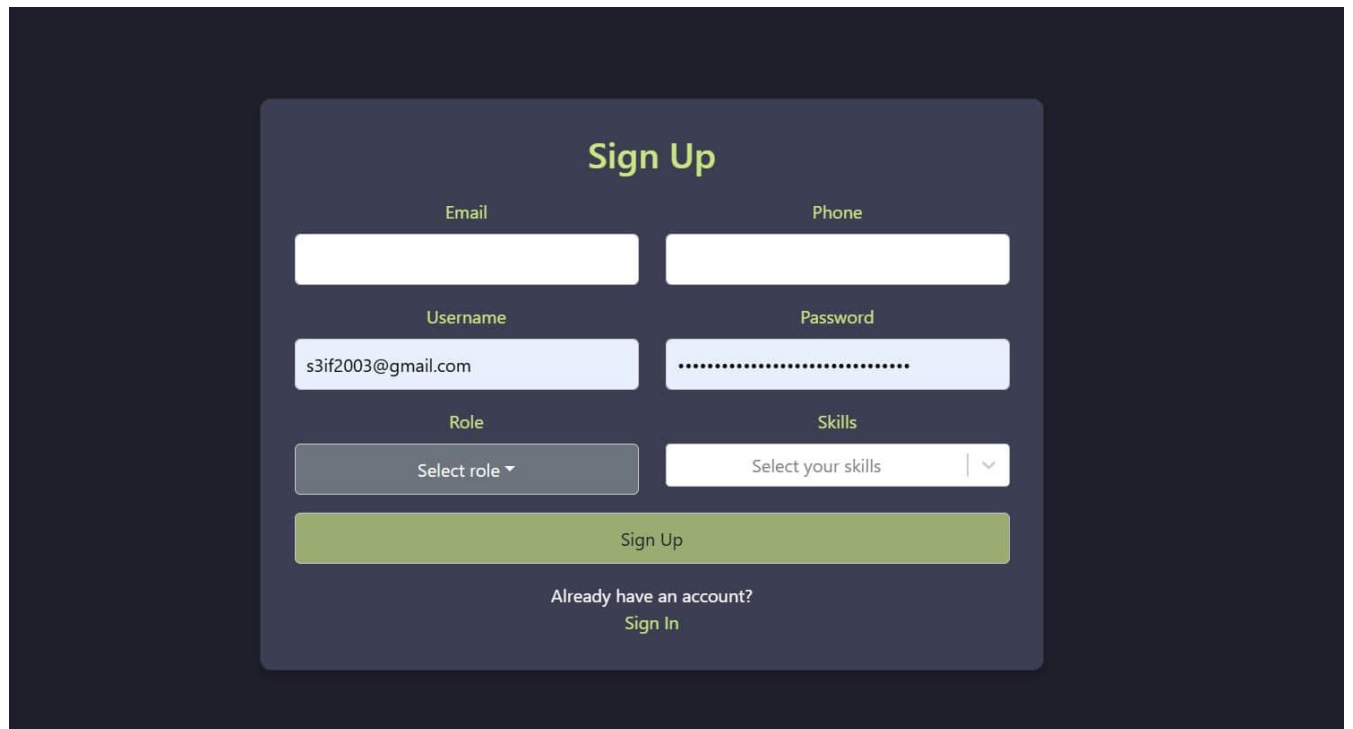


Figure 4 : Interface Sign In



The image shows a 'Sign Up' form on a dark blue background. The form is a lighter blue rectangle with the title 'Sign Up' in green at the top. It contains six input fields: 'Email' and 'Phone' (empty), 'Username' (containing 's3if2003@gmail.com'), and 'Password' (masked with dots). Below these are two dropdown menus: 'Role' (showing 'Select role') and 'Skills' (showing 'Select your skills'). A green 'Sign Up' button is at the bottom of the form. Below the button, the text 'Already have an account?' is followed by a green 'Sign In' link.

Sign Up

Email Phone

Username Password

s3if2003@gmail.com

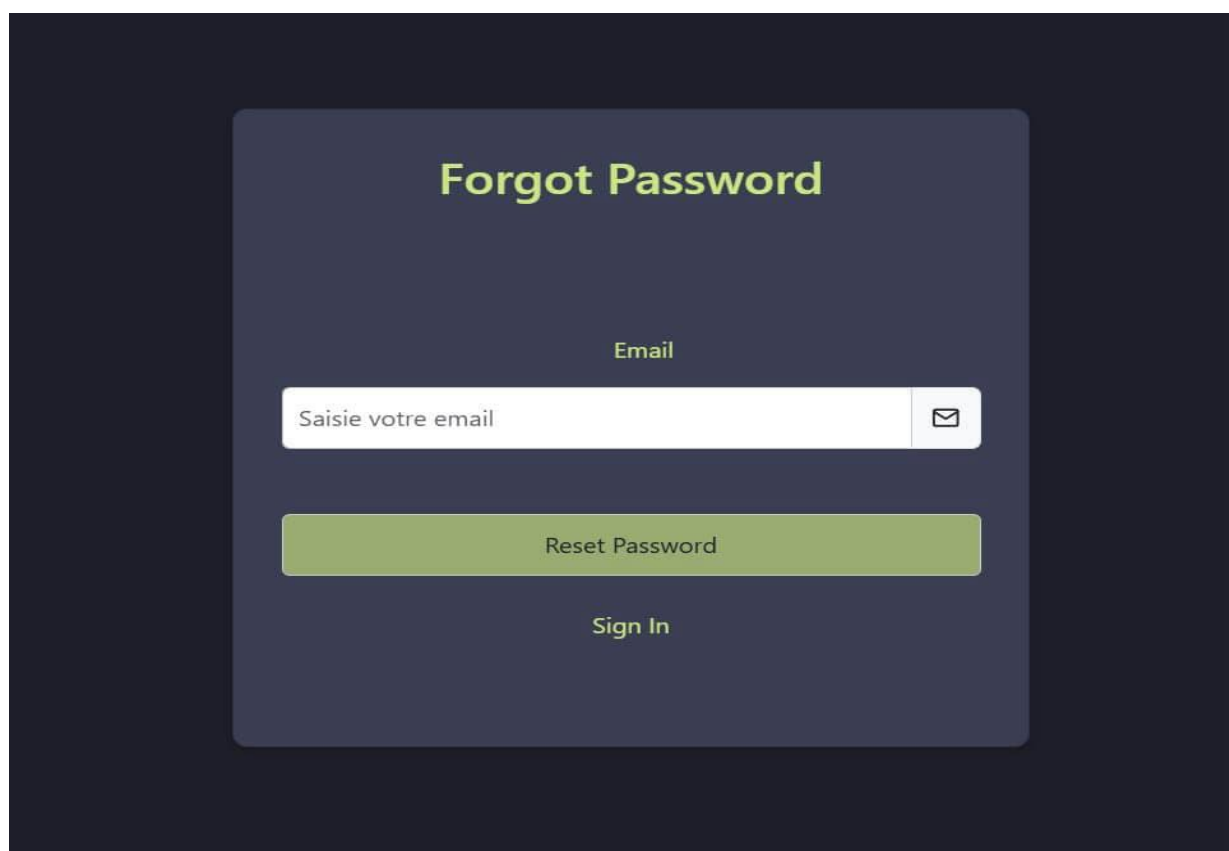
Role Skills

Select role Select your skills

Sign Up

Already have an account?
Sign In

Figure 5 : Interface Sign Up



The image shows a 'Forgot Password' form on a dark blue background. The form is a lighter blue rectangle with the title 'Forgot Password' in green at the top. It contains one input field for 'Email' with the placeholder text 'Saisie votre email' and an envelope icon. Below the input field is a green 'Reset Password' button. At the bottom of the form is a green 'Sign In' link.

Forgot Password

Email

Saisie votre email

Reset Password

Sign In

Figure 6 : Interface d'oubli de mot de passe

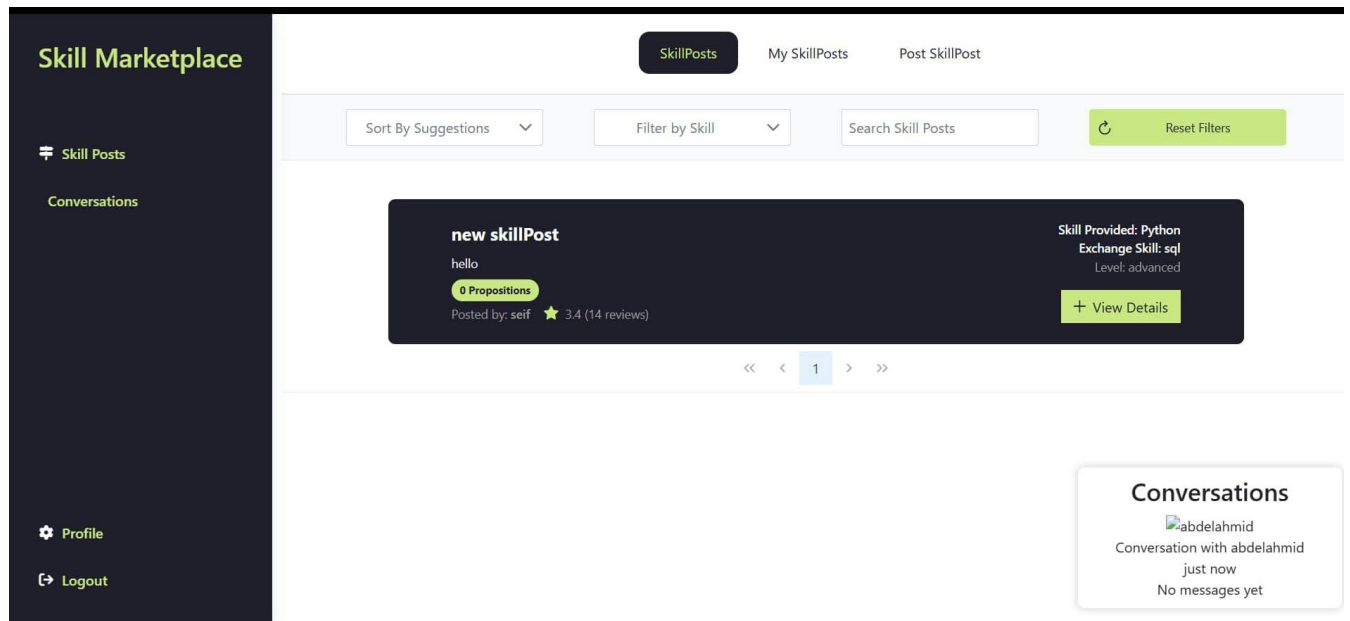


Figure 7 : Interface de liste des offres

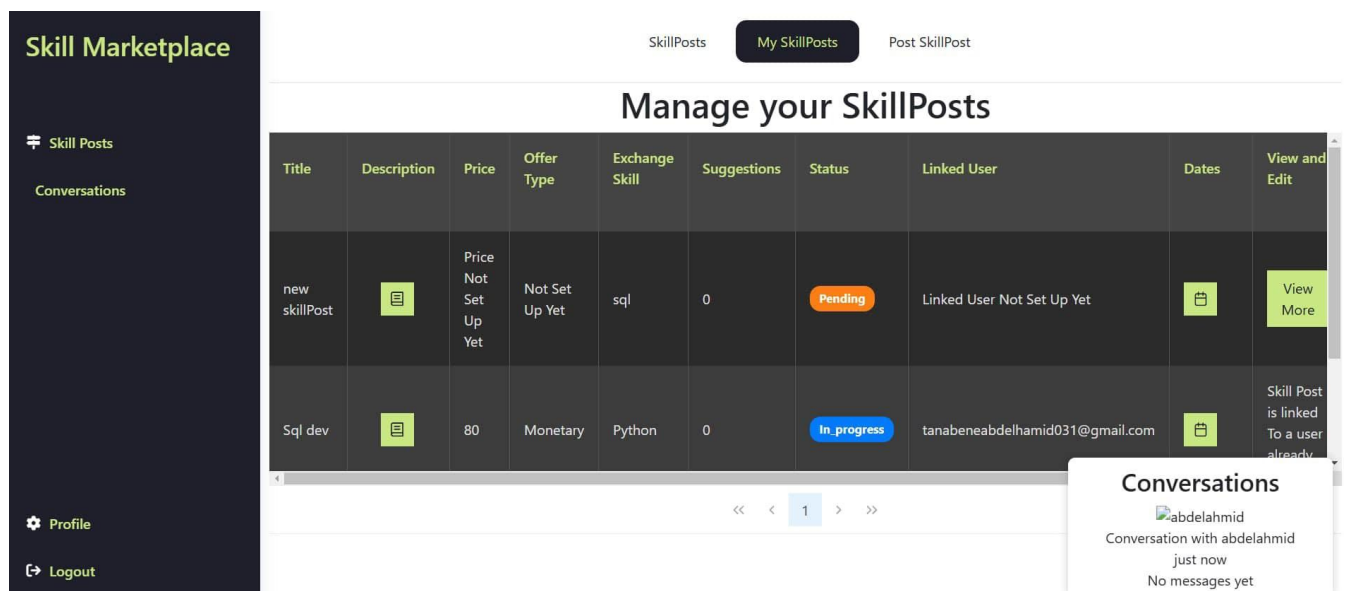


Figure 8 : Interface des offres de l'utilisateur

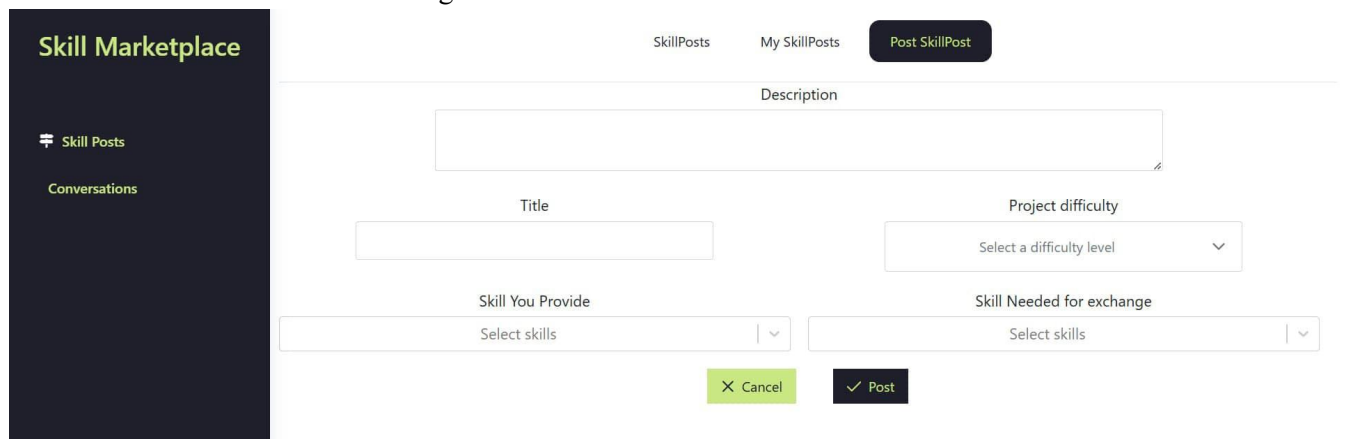


Figure 9 Interface pour la publication d'une offre

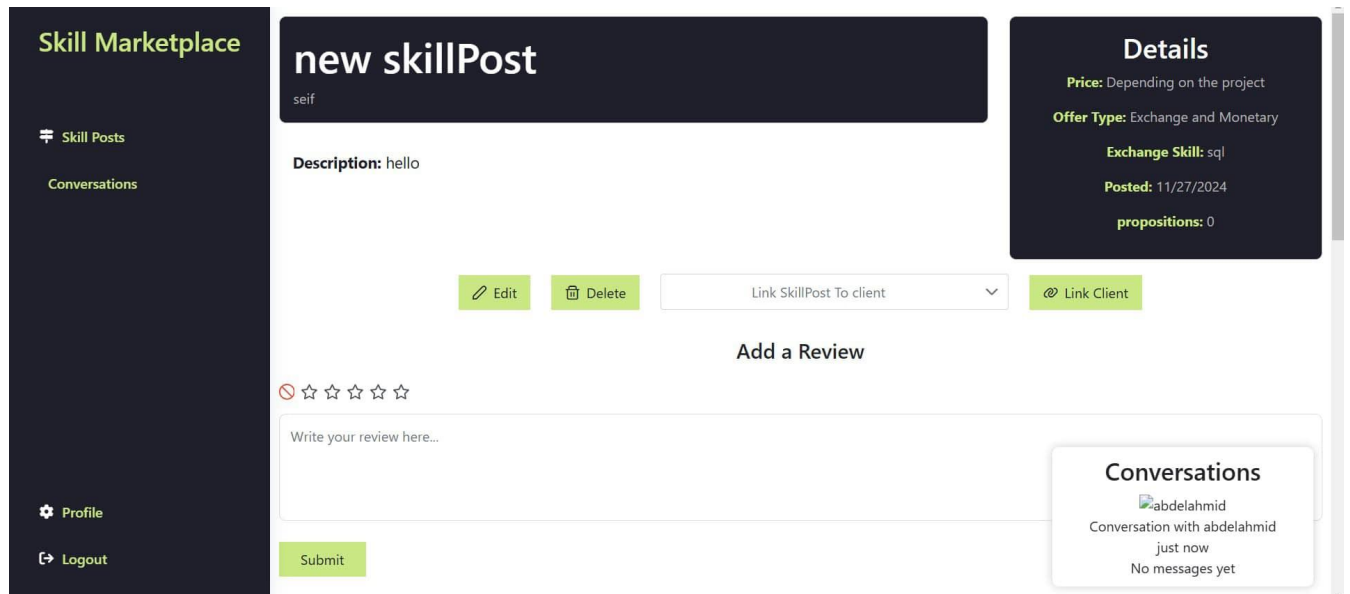


Figure 10 Interface pour les informations liées à une offre

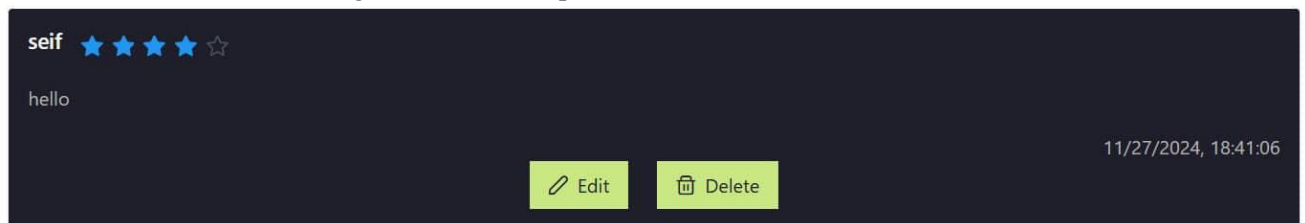


Figure 11 Exemple d'un niveau

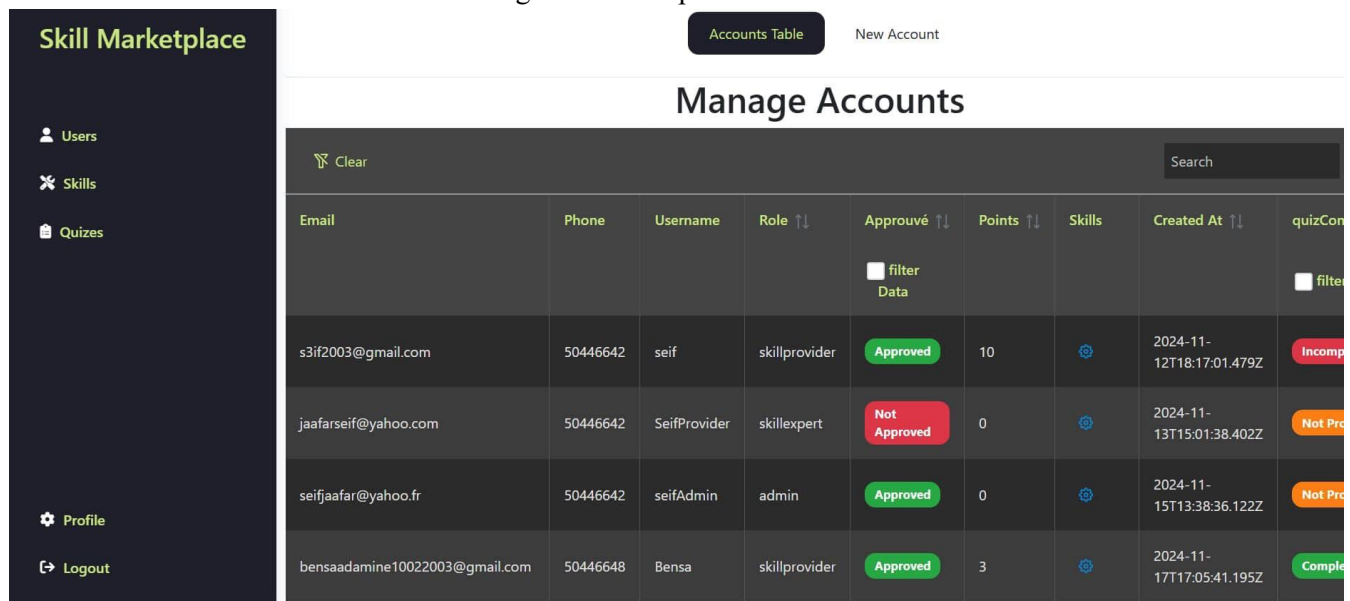


Figure 12 Gestion des comptes faites par l'admin

Accounts Table
New Account

Email

Username

Phone

Role

Approval Status

Skills

Points

Quiz Completed

0

false

Cancel

Add User

Figure 13 Interface pour l'ajout d'un compte (uniquement pour l'admin)

Skill Marketplace

Users
Skills
Quizes

Profile
Logout

Skills Table

New Skill

Manage Skills

Clear

Search

Name	Description	
Python		<div></div> <div></div> <div></div>
sql		<div></div> <div></div> <div></div>
Java		<div></div> <div></div> <div></div>

<<

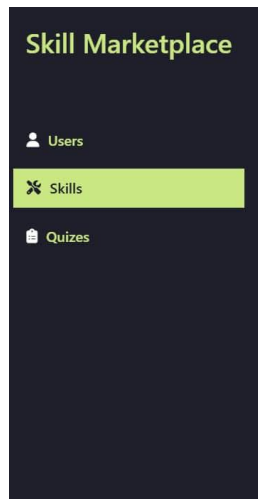
<

1

>

>>

Figure 14 Gestion des compétences (skills)



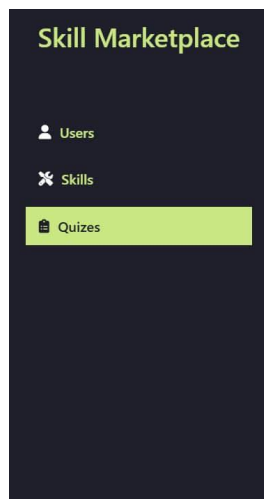
Skills Table [New Skill](#)

Name

Description

[X Cancel](#) [✓ Add Skill](#)

Figure 15 Ajout d'une nouvelle compétence



[Quizes Table](#) [New Quiz](#)

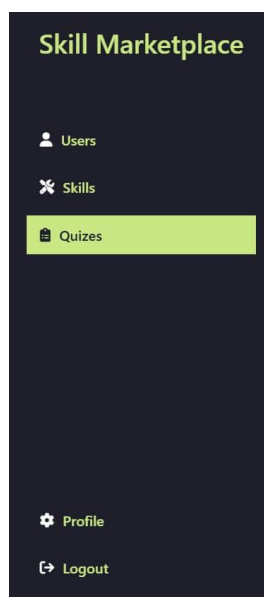
Manage Quizes

[Clear](#)

Title	Skill	Questions	
Java Basic Quiz	Java	Edit Questions	✎ ✖
Sql Basic Quiz	sql	Edit Questions	✎ ✖
Python Basics Quiz	Python	Edit Questions	✎ ✖

<< < 1 > >>

Figure 16 Gestion des quizzes



[Questions Table](#)

Manage Questions

[Clear](#)

QuestionText	Options	correct Answer Index	
Which SQL constraint ensures that all values in a column are unique?	?	2	✎ ✖
Which SQL function is used to count the number of rows in a table?	?	1	✎ ✖
What type of JOIN returns all records from the left table and the matched records from the right table?	?	1	✎ ✖
Which SQL clause is used to filter records based on a specified condition?	?	2	✎ ✖
Which SQL command is used to retrieve data from a database?	?	0	✎ ✖

<< < 1 > >>

Figure 17 Gestion des questions d'un quiz

Explication des choix d'ergonomie et d'expérience utilisateur

Les choix d'ergonomie et d'expérience utilisateur sont essentiels pour rendre la plateforme intuitive et agréable à utiliser. Nous avons opté pour une interface claire, avec des fonctionnalités bien structurées et faciles d'accès, afin que chaque utilisateur, qu'il soit novice ou expérimenté, puisse naviguer sans difficulté. L'objectif est d'offrir une expérience fluide, en réduisant le nombre d'étapes nécessaires pour accomplir une tâche et en intégrant des éléments visuels qui guident l'utilisateur tout au long de son parcours. Ainsi, chaque interaction est pensée pour être rapide et efficace.

3.6 Algorithmes et Méthodes

Description des algorithmes clés utilisés dans la solution

Dans le cadre de la plateforme, plusieurs algorithmes et méthodes ont été mis en place pour assurer une gestion fluide et efficace des données et des interactions entre les utilisateurs. Parmi les algorithmes clés, nous avons intégré des techniques de hachage sécurisé pour la gestion des mots de passe (Bcrypt), permettant de protéger les informations sensibles des utilisateurs.

Des algorithmes de recherche avancée sont utilisés pour permettre aux utilisateurs de trouver rapidement les compétences ou offres qui correspondent à leurs critères, avec des filtres personnalisables comme la compétence, le nombre de connexions, et le rôle des utilisateurs.

Les diagrammes UML ont été utilisés pour modéliser les processus métier de manière visuelle. Ces diagrammes permettent de décrire de manière détaillée les flux de données.

Enfin, des méthodes de gestion des sessions sécurisées (via JWT et cookies HttpOnly) sont utilisées pour protéger l'accès aux fonctionnalités sensibles. L'utilisation de ces techniques garantit la confidentialité et la sécurité des interactions sur la plateforme, tout en offrant une expérience utilisateur fluide et sécurisée.

Utilisation de diagrammes d'activités UML pour modéliser les processus

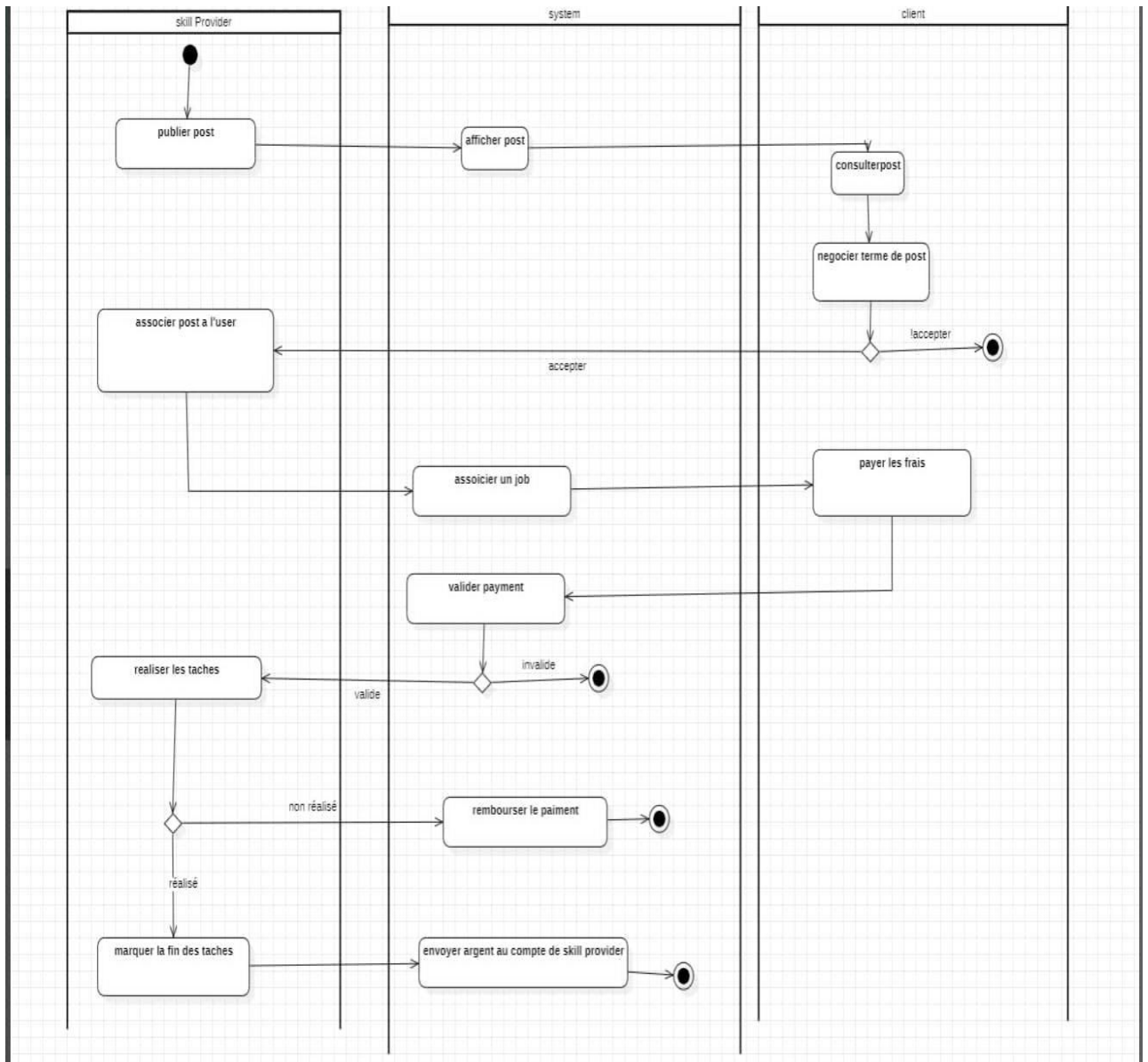


Figure 18 Diagramme d'activité de la fonction d'échange entre le client et le skill-provider

Ce diagramme illustre l'enchaînement des activités élaborées dans le système entre le skillprovider et le client qui englobe les fonctionnalités clés de notre projet Skill Marketplace mettant en œuvre l'échange établi entre ces deux acteurs, le client peut se bénéficier d'un service offert par le skill-provider, effectue une négociation et traite le paiement mettant ainsi fin à cette tâche.

3.7 Sécurité et Confidentialité

La sécurité et la confidentialité sont des aspects essentiels pour protéger les données sensibles des utilisateurs et garantir la fiabilité des opérations. Cette section détaille les mesures de sécurité mises en place pour répondre aux exigences de protection des données, en s'adaptant spécifiquement à l'architecture NoSQL utilisée.

Stratégies de sécurité mises en place

- Hachage des mots de passe
 - Un algorithme de hachage sécurisé (Bcrypt) est utilisé pour chiffrer les mots de passe avant leur stockage.
 - Cette méthode réduit les risques en cas de violation des données, évitant que les mots de passe soient stockés en clair.
- Gestion des Sessions et Cookies
 - Les sessions sont utilisées pour s'assurer que seuls les utilisateurs autorisés ont accès aux fonctionnalités sensibles.
 - Un contrôle strict des sessions et des cookies bloque toutes les tentatives d'accès non autorisées.
- Séparation des Rôles et des Permissions
 - Une architecture basée sur les rôles est implémentée pour restreindre les droits d'accès en fonction des responsabilités des utilisateurs.
 - Chaque utilisateur dispose de permissions spécifiques, limitant son accès aux fonctionnalités pertinentes.

Mesures de Sécurité Adaptées à la Base de Données NoSQL

- Hachage Sécurisé des Mots de Passe ◦ Tous les mots de passe sont hachés (avec Bcrypt) avant d'être stockés dans la base de données NoSQL.
- Protection contre les Accès Non Autorisés ◦ Les fonctionnalités sensibles sont protégées par des sessions valides.

- Les cookies sécurisés limitent l'exposition des identifiants de session et renforcent la sécurité globale.
- Vérification des Informations Utilisateur ○ Les informations saisies par les utilisateurs sont validées et nettoyées pour prévenir les attaques d'injection NoSQL, telles que l'injection d'opérateurs MongoDB.

Ces mesures assurent une protection robuste des données tout en offrant une expérience utilisateur fiable et sécurisée. Les couches multiples de sécurité garantissent que les utilisateurs peuvent utiliser la plateforme en toute confiance.

3.8 Scénarios de Cas d'Utilisation

Présentation de scénarios d'utilisation typiques

Les scénarios de cas d'utilisation mettent en lumière les principales fonctionnalités et montrent comment les rôles, comme le Skill-Provider, le SkillExpert et le Client, participent au système.

Inscription et Connexion

Un utilisateur souhaite accéder à la plateforme. S'il n'a pas encore de compte, il s'inscrit en renseignant ses informations personnelles. S'il est déjà inscrit, il se connecte en entrant ses identifiants. Une fois connecté, le système vérifie ses droits et lui donne accès aux fonctionnalités adaptées à son rôle.

Publication d'une Offre

Un Skill-Provider veut proposer un service. Il décrit son offre en précisant les détails importants : compétences nécessaires, tarif, et disponibilité. Une fois soumise, l'offre est examinée par un administrateur qui décide de l'approuver ou de la rejeter selon les critères de la plateforme.

Recherche et Filtrage

Un client cherche un talent spécifique. Grâce à un moteur de recherche intuitif, il peut trouver des Skill-Providers en filtrant par compétence, expérience, ou nombre de connexions effectuées. Il consulte les profils et offres disponibles, puis choisit celle qui correspond le mieux à ses besoins.

Participation à un Cours

Un Skill-Expert propose un cours pour partager ses connaissances. Le skillprovider intéressé s'inscrit, effectue un paiement sécurisé via la plateforme, et assiste au cours via des meets. Cette interaction favorise l'apprentissage tout en offrant une opportunité de collaboration directe.

Évaluations et Retours

Après avoir utilisé un service ou participé à un cours, le client laisse un avis et une note pour partager son expérience. Ces retours permettent à la communauté de mieux évaluer la qualité des services et aident les Skill-Providers et Skill-Experts à s'améliorer.

Utilisation de diagrammes de cas d'utilisation UML pour illustrer les interactions fonctionnelles

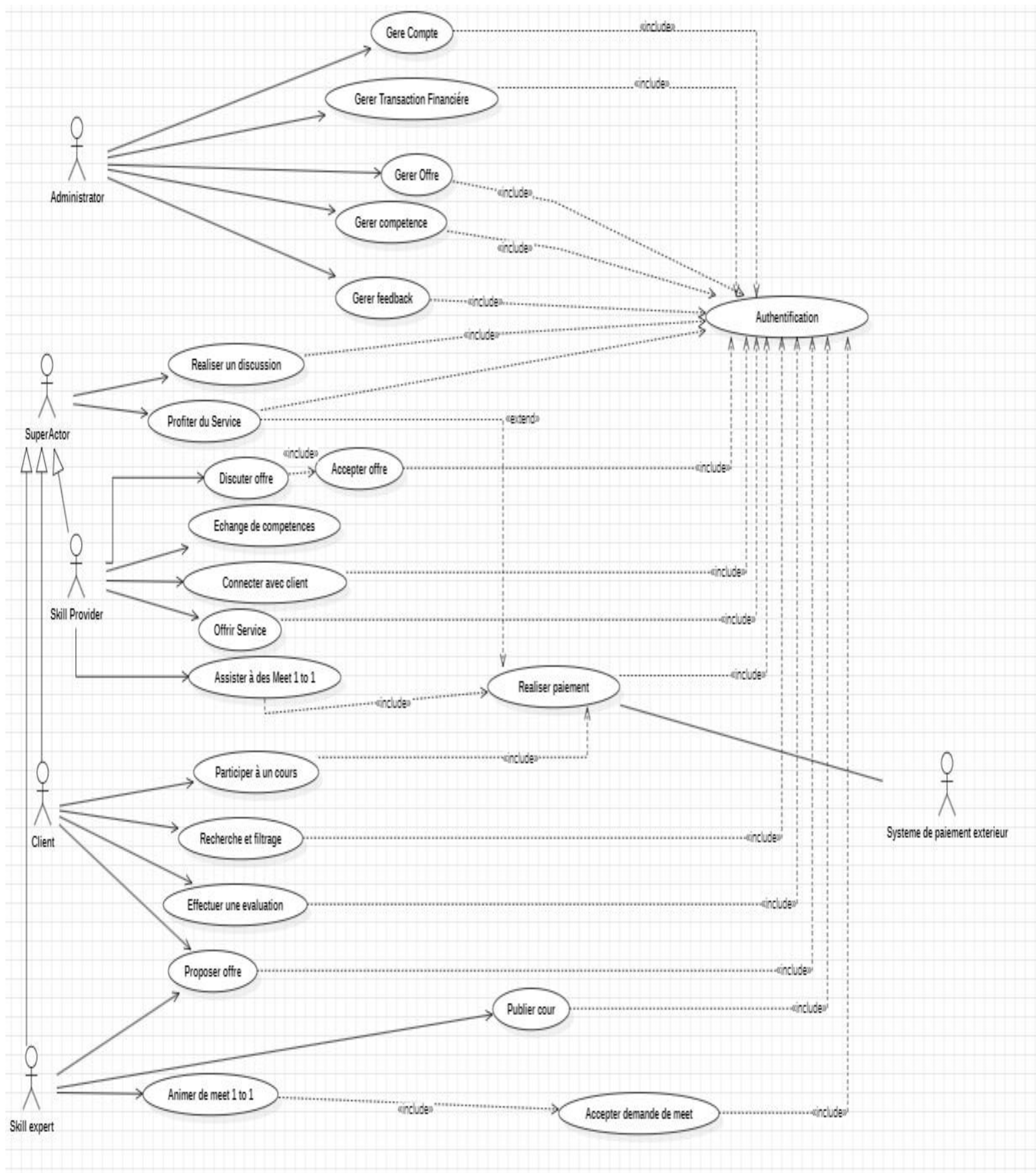


Figure 19 Diagramme de cas d'utilisation

3.9 Diagrammes de Séquence

Utilisation de diagrammes de séquence UML pour modéliser les interactions entre les objets du système

Dans le cadre de notre projet, plusieurs diagrammes de séquence peuvent être élaborés pour démontrer les processus critiques du système.

On a élaboré quelques exemples de scénarios qui sont représentés par ces diagrammes :

- **Scénario de Connexion (Utilisateur avec Compte Existant)**

Dans ce scénario, un utilisateur qui a déjà un compte se connecte à la plateforme en entrant ses identifiants (nom d'utilisateur et mot de passe).

Diagramme de Séquence :

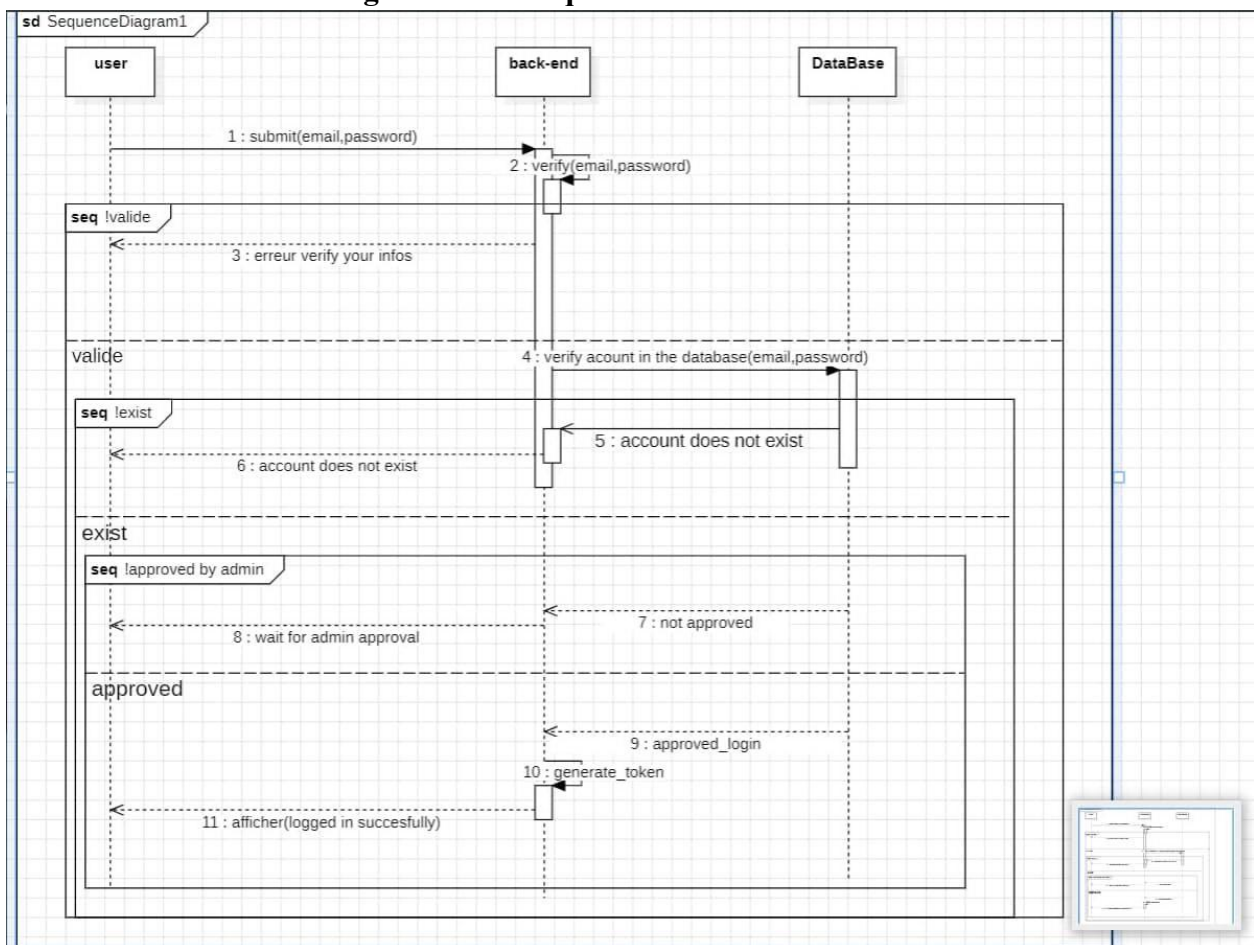


Figure 20 Diagramme de séquence illustrant le processus de la connexion pour un utilisateur existant

L'utilisateur entre ses identifiants (nom d'utilisateur et mot de passe) via l'interface utilisateur, qui envoie ensuite les données au backend pour vérification.

Le backend compare les informations avec celles stockées dans la base de données. Si les identifiants sont valides, un jeton de session (JWT) est généré et envoyé à l'interface, permettant l'accès aux fonctionnalités protégées. Si les informations sont incorrectes, un message d'erreur s'affiche à l'utilisateur.

- **Scénario d’Inscription (Utilisateur Sans Compte)**

Dans ce scénario, un utilisateur qui n’est pas encore connecté à la plateforme crée un compte.

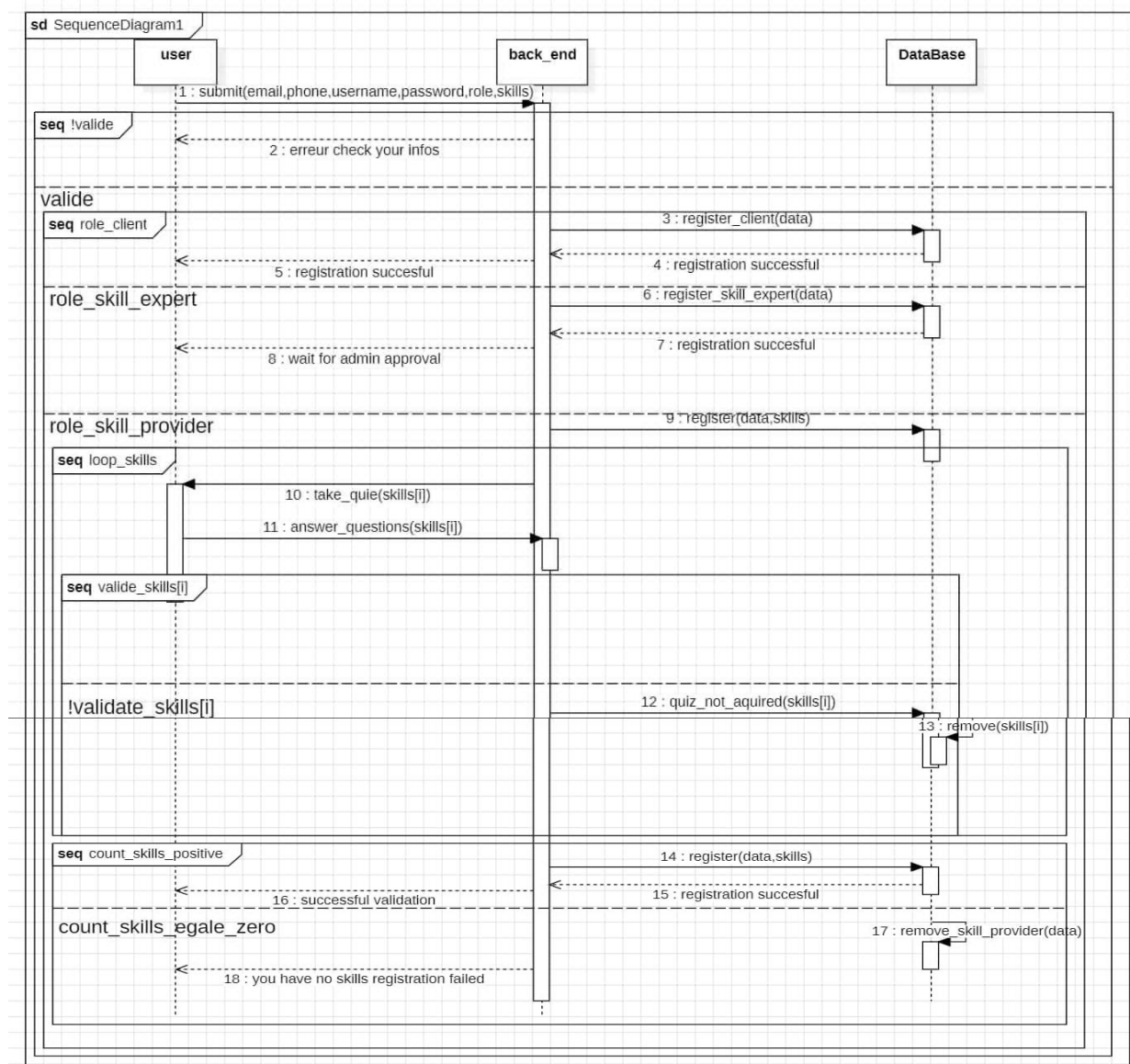


Figure 21 Diagramme de Séquence illustrant l’inscription à la plateforme

Dans ce scénario, un utilisateur qui n'a pas encore de compte s'inscrit sur la plateforme. L'utilisateur sélectionne l'option "S'inscrire" sur l'interface et entre

ses informations personnelles, telles que son nom, son email et son rôle. Ces informations sont envoyées au backend par l'interface utilisateur. Le backend vérifie alors si l'email est déjà associé à un compte existant. Si l'email est déjà utilisé, un message d'erreur est retourné à l'utilisateur. Si l'email est unique, le backend crée un nouveau compte en hachant le mot de passe avec un algorithme sécurisé (Bcrypt), puis enregistre les données dans la base de données. L'utilisateur reçoit une confirmation d'inscription réussie et peut se connecter avec ses nouveaux identifiants.

- **Achat des cours et réservation des meets one on one**

Dans ce scénario , un skill-provider ou un skill-expert a la possibilité d'acheter un cours fourni par un autre skill-expert ou bien de réserver une réunion privée (meet one on one) avec un expert.

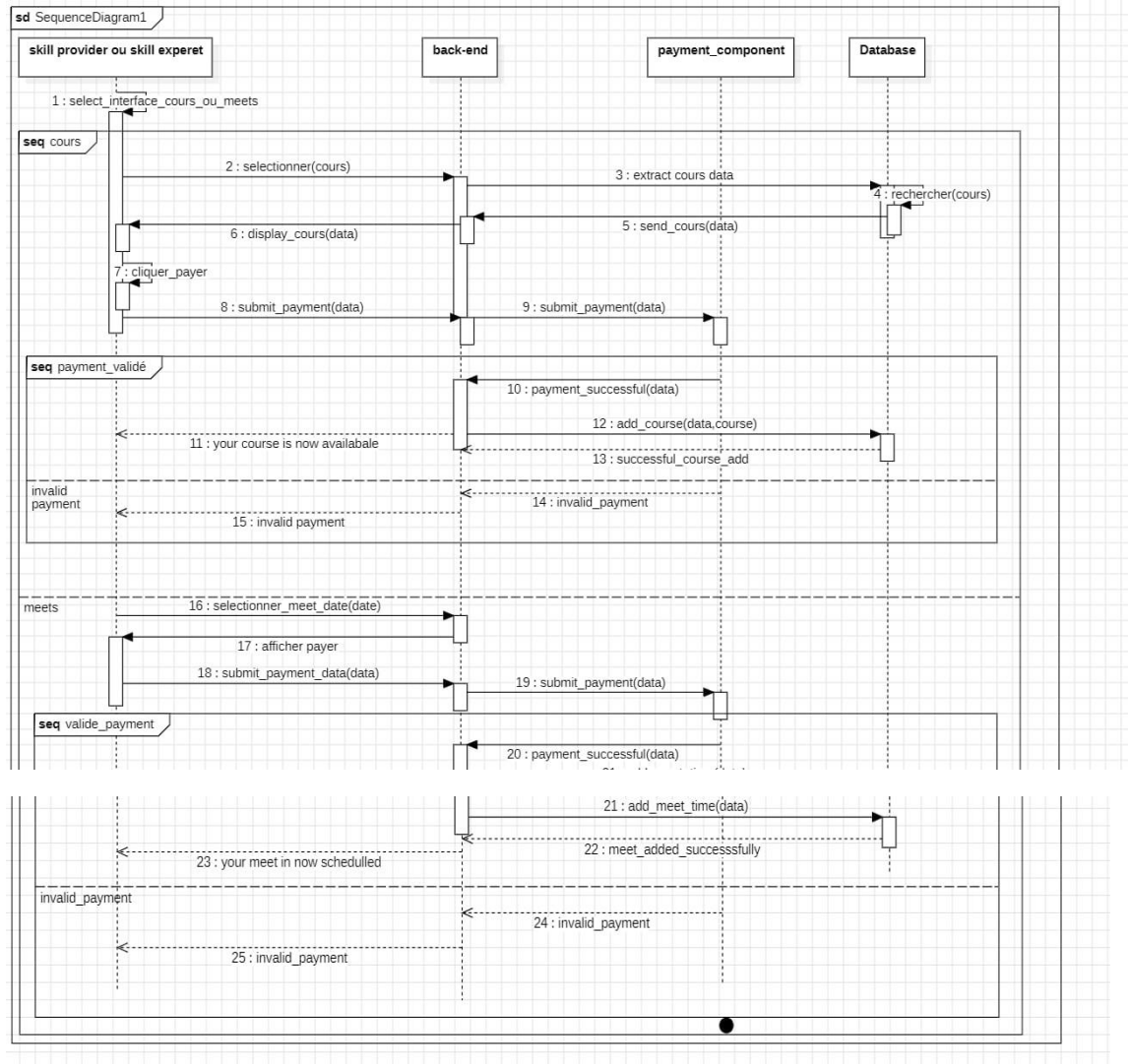


Figure 22 Diagramme de séquence illustrant la publication des cours et la réservation des meets

3.10 Plan de Développement

Division du projet en modules ou itérations de développement

- **Modules Backend**

- **Module Utilisateur :**

Ce module gère l'inscription, la connexion, la gestion des profils utilisateur et l'attribution des rôles (Skill-Provider, Skill-Expert, Client). Il assure également la gestion des mots de passe et l'authentification via JWT.

- **Module Offre :**

Ce module permet la création, modification et suppression des offres.

Il gère aussi la compétence requise pour chaque offre et l'attribution des

compétences appropriées pour chaque utilisateur (Skill-Provider). ○

Module Paiement :

L'intégration avec la passerelle de paiement (Stripe) pour la gestion des transactions financières entre Skill-Providers, Skill-Experts et Clients. Ce module assure la sécurité des paiements et la gestion des factures. ○

Module Messagerie :

Ce module gère les conversations entre utilisateurs. Il inclut également un système de notification pour alerter les utilisateurs de nouveaux messages ou événements.

○ **Module Évaluation :**

Ce module permet aux utilisateurs de laisser des évaluations et des commentaires après avoir utilisé un service. Les évaluations influencent les futures offres et recommandations sur la plateforme. ○

Module Cours :
Permet aux Skill-Experts de proposer des cours, et aux Clients d'y participer. Ce module gère également l'inscription aux cours et l'activation des contenus pédagogiques.

- **Modules FrontEnd**

- **Module Authentification :**

Gère l'interface de connexion et d'inscription des utilisateurs. Permet l'affichage des formulaires et la gestion des erreurs de connexion.

- **Module Recherche :**

Permet aux utilisateurs de rechercher des experts ou des offres en utilisant des filtres avancés (par compétences, évaluations, etc.). ○

- **Module Profil :**

Affiche les informations personnelles de l'utilisateur, avec la possibilité de modifier les données de profil comme le nom, l'email et les compétences.

- **Module Offres :**

Permet aux utilisateurs de visualiser, créer ou demander des offres de services, et de consulter les informations détaillées des Skill-Providers. ○

Module Messagerie :

Gère l'interface de conversation entre les utilisateurs, permettant une communication fluide, et le suivi des messages. ○ **Module**

Paielement :

Permet aux utilisateurs de payer pour les services via une interface sécurisée. Intégration avec le module backend de paiement.

Calendrier de mise en œuvre de la conception

Sprint	Backend	Frontend
Sprint 1 Fondations	API REST avec Node.js, connexion à MongoDB, création des modèles de données pour les utilisateurs et les offres.	Structure de l'application React, mise en place de l'authentification.
Sprint 2 Les fonctions métiers	Développement des modules Offre.	Implémentation de la fonctionnalité qui permet la création ainsi que l'affichage des offres
Sprint 3 Communication et collaboration	Développement du module de messagerie.	Intégration de l'interface de discussion
Sprint 4 Les profils et le système de point	Développement du module des profils et du système de point	Affichage des profils et de la fonctionnalité « laisser un feedback »
Sprint 5 Paielement en ligne	Développement du module Paiement.	Affichage d'un formulaire d'un paiement en ligne et associer l'API du paiement à l'interface.
Sprint 6 Les classes et le calendrier	Développement des classes et mise en œuvre du module de calendrier.	Affichage des cours, et la prise de rendez-vous

Tableau 1 Calendrier de la mise en œuvre de la conception

Pour le reste des sprints :

L'amélioration et l'ajout si le besoin se présente des nouvelles fonctionnalités comme notifications etc.

Maintenance : gestion des erreurs et "bug", mise à jour concernant la sécurité .

3.11 Conclusion du Chapitre

Récapitulation des principaux éléments de la conception

Dans cette section, nous avons exploré les principales composantes de la conception du système, en mettant l'accent sur la structure générale, l'architecture et les composants essentiels pour le bon fonctionnement de l'application. Voici les points clés que nous avons abordés :

1. Architecture du système :

- Le système repose sur une architecture modulaire avec un backend développé en Node.js, utilisant MongoDB pour la gestion des données.
- L'interface utilisateur est construite avec React, garantissant une expérience fluide et interactive.

2. Modules clés :

- Backend : Gestion des utilisateurs, des offres, des paiements, de la messagerie et des évaluations, ainsi que des fonctionnalités liées aux cours et aux calendriers.
- Frontend : Interfaces pour l'authentification, la recherche, la gestion des profils, la messagerie et le calendrier.
- Sécurité : Des mesures robustes telles que le hachage des mots de passe, l'utilisation de JWT pour les sessions, et la protection contre les injections assurent la confidentialité et l'intégrité des données.

Transition vers les phases suivantes du développement

Avec la conception du système en place, nous sommes désormais prêts à passer aux étapes suivantes du développement. Ces étapes incluent :

1. Développement des fonctionnalités clés :

- La mise en œuvre des modules comme l'inscription des utilisateurs, la gestion des offres et des paiements, ainsi que l'intégration de la messagerie et du calendrier.

2. Intégration et tests :

- Une fois les fonctionnalités développées, nous procéderons à l'intégration des différents modules et commencerons les tests fonctionnels et de performance pour valider la cohérence du système.

3. Déploiement progressif :

- Après avoir validé les tests, nous déploierons progressivement l'application dans un environnement de production, tout en assurant la maintenance continue et en surveillant les performances.

4. Améliorations continues :

- Des itérations futures permettront d'ajouter de nouvelles fonctionnalités et d'optimiser les performances, tout en prenant en compte les retours des utilisateurs pour améliorer l'expérience globale.

Chapitre 4 : Réalisation et validation

Ce plan couvre de manière systématique les différentes étapes de la réalisation et de la validation du projet Skill Marketplace, mettant en lumière les aspects clés du développement, des tests, de l'intégration, et de la documentation technique, tout en assurant une transition fluide vers la phase suivante du projet.

Plan du chapitre

1- Introduction à la Réalisation

- Présentation des objectifs de la phase de réalisation
- Enjeux liés à la mise en œuvre du projet

2- Environnement de Développement

- Description des outils, plateformes, et langages de programmation utilisés
- Justification des choix technologiques

3- Développement des Modules

- Détails sur la création des modules identifiés dans la conception
- Utilisation de UML pour guider l'implémentation

4- Tests et Débogage

- Stratégies de tests appliquées aux différents modules
- Processus de débogage pour résoudre les erreurs identifiées

5- Intégration des Composants

- Mise en œuvre de l'architecture globale du système
- Coordination des différents modules pour assurer leur interopérabilité

6- Validation et Vérification

- Évaluation de la solution par rapport aux besoins initiaux
- Vérification de la conformité aux spécifications définies dans la conception

7- Optimisation des Performances

- Identification des points d'optimisation éventuels

8- Documentation Technique

- Rédaction d'une documentation détaillée pour faciliter la maintenance
- Utilisation de UML pour compléter la documentation technique

9- Formation et Transfert de Compétences

- Formation des utilisateurs finaux et de l'équipe de maintenance
- Transfert des compétences nécessaires à la gestion du projet

10- Conclusion du Chapitre

- Bilan de la réalisation et de la validation
- Préparation à la phase de déploiement et d'exploitation du projet

1- Introduction à la Réalisation

Objectifs de la phase de réalisation

- Créer une application entièrement intégrée avec un environnement contemporain pour une interface utilisateur dynamique (React), une base de données NoSQL (MongoDB) et un backend compact (Node.js).
- Simplifier la gestion de données complexes grâce à MongoDB tout en rendant le développement plus simple grâce à une structure RESTful.

Enjeux

- Garantir un échange performant entre le frontend et le backend.
- Proposer une expérience d'utilisateur fluide et efficace, malgré des volumes de données considérables.
- Assurer la protection des données et la capacité d'extension du projet

Environnement de Développement :

Outils utilisés :

- **React** pour le frontend : flexibilité, réponse rapide et écosystème abondant.
- **Node.js** pour le backend : rapidité, simplicité et compatibilité avec MongoDB.
- **MongoDB** comme base de données : grâce à sa configuration ajustable et sa flexibilité d'adaptation.

Débat sur les choix :

- **React** : Son DOM virtuel facilite une performance rapide et une expérience d'utilisateur sans contrainte.
- **Node.js** : Parfait pour les API REST et les architectures fondées sur des services.
- **MongoDB** : Idéal pour les applications nécessitant des relations non statiques entre les données.

3- Développement des Modules

les modules :

Modules pour l'Administrateur :

Gestion des Comptes :

- Interface pour ajouter, supprimer ou modifier les informations des utilisateurs.
- Gestion des rôles (Client, Skill Provider, Skill Expert).
- Dashboard pour surveiller l'activité des utilisateurs.

Gestion des Transactions Financières :

- Module backend pour suivre les paiements, vérifier les transactions, et résoudre les litiges.
- Intégration avec un système de paiement externe via une API (Stripe)

Gestion des Offres :

- Administration des services proposés par les Skill Providers.
- Suivi des performances des offres (nombre de clients, évaluations).

Gestion des Compétences :

- Ajout de nouvelles compétences ou suppression de compétences obsolètes

Gestion des Feedbacks :

- Analyse des évaluations laissées par les utilisateurs
- Signalement des abus ou commentaires inappropriés

Modules pour le Client :

Recherche et Filtrage :

- Moteur de recherche avancé pour trouver des services
- Filtres par compétence, évaluation, disponibilité **Participation à un**

Cours :

- Gestion des inscriptions et affichage des cours disponibles

Proposition et Discussion des Offres :

- Interface permettant de soumettre une offre personnalisée

- Messagerie pour discuter directement avec le Skill Provider.

Évaluation des Services :

- Interface pour laisser des évaluations et des commentaires

Modules pour le Skill Provider :

Offrir des Services :

- Création d'offres détaillées avec des descriptions, prix et conditions.

Connexion avec les Clients :

- Messagerie en temps réel pour communiquer avec les clients
- Système de notifications pour les demandes ou les messages reçus

Paielements et Réalisations :

- Gestion des paiements pour chaque service fourni
- Suivi des paiements en attente ou effectués

Modules pour le Skill Expert :

Organiser des Meet 1 on 1 :

- Gestion des horaires et des invitation
- Intégration avec des outils de visioconférence comme Zoom ou Google Meet

Publication de Cours :

- Interface pour créer et publier des cours en ligne.

Acceptation des Demandes de Meet :

- Consultation des demandes en attente et acceptation ou rejet.
- Affichage des rendez-vous programmés.

3.2-Usage UML :

- Usage de diagramme de classe pour représenter les relations entre les entités
- Usage des diagrammes de séquence pour décrire les interactions frontend-backend
- Usage de diagramme de cas d'utilisation pour illustrer les fonctionnalités des acteurs avec le système.
- Usage de diagramme d'activité pour la description facile des fonctionnalités dans le système.
- Usage de digramme de composants pour identifier les composants principaux de système.

- Usage de diagramme de déploiement pour illustrer la structure de système

4- Tests et Débogage

- Tests des composants React avec **Jest** et **React Testing Library**.
- Validation des schémas MongoDB avec **Mocha** ou **Chai**
- Utilisation de **Postman** pour tester les routes de l'API
- Simulation des flux utilisateur sur l'interface

Processus de débogage pour résoudre les erreurs identifiées

- Déboguer React via **React Developer Tools**.
- Analyse des requêtes avec **MongoDB Compass**.

5- Intégration des Composants

Mise en œuvre de l'architecture globale du système :

- Structure RESTful pour une communication claire entre le frontend et le backend.
- Utilisation de l'API Fetch pour les appels réseau.

Coordination des différents modules pour assurer leur interopérabilité

- Validation des données entrantes via Mongoose dans le backend.
- Synchronisation du state global avec Context API.

6- Validation et Vérification

Évaluation de la solution par rapport aux besoins initiaux :

- On a comparé les fonctionnalités implémentées aux exigences initiales par la vérification module par module et tester les interactions entre les différents modules on a aussi étudié tous les scénarios possibles d'interaction et les tester pour évaluer notre projet .

Vérification de la conformité aux spécifications définies dans la conception

- Tout d'abord on a vérifier le système de routing pour assurer qu'il fonctionne pour les différentes requêtes possible puis on a vérifier les appels api pour assurer la conformité aux conception et au résultat prévue de système

7- Optimisation des Performances

Identification des points d'optimisation éventuels :

- Optimisation des mesures des sécurité en assurant que tous les transactions au niveau de payement sont sécurisé et bloquer les accès interdit par des personnes externes
- Optimisation concernant la taille par exemple réduction des tailles des bundles JavaScript avec **Webpack**.

Actions entreprises pour améliorer les performances du système :

- Implémenter une infrastructure bien structurée et bien sécurisée
- Implémenter des services Intelligence artificielle pour obtenir une page évolutive en temps

8- Documentation Technique :

Rédaction d'une documentation détaillée pour faciliter la maintenance :

- Définir chaque route REST avec ses méthodes, paramètres, et réponses attendues à l'aide de **Swagger** pour permet la possibilité de le reconfigurer et aussi l'ajout des nouvelles routes
- Définir les fichiers de chaque module avec une description au niveau de code en utilisant des commentaires pour une lecture claire.

Utilisation de UML pour compléter la documentation technique :

- Diagrammes UML déploiement et séquence pour illustrer le fonctionnement interne et comprendre la structure globale implémentée et pour avoir aussi la possibilité de changer des composants at l'adapter par l'infrastructure mise en place.

9- Formation et Transfert de Compétences

Formation des utilisateurs finaux et de l'équipe de maintenance

- Sessions pratiques pour expliquer les principales fonctionnalités.
- Création de vidéos ou de guides interactifs

Transfert des compétences nécessaires à la gestion du projet :

- Formation des développeurs sur l'environnement React/Node/MongoDB pour la maintenance.
- Documentation sur la gestion des bases de données et les meilleures pratiques.

10- Conclusion du Chapitre

Bilan de la Réalisation :

- L'ensemble des modules définis durant la phase de conception a été développé avec succès, en respectant les objectifs fixés.
- Les choix technologiques (React pour le frontend, Node.js pour le backend, et MongoDB pour la base de données) se sont avérés adaptés pour répondre aux besoins de performance, d'évolutivité, et de modularité.
- Une attention particulière a été portée à la sécurité (authentification JWT, gestion des permissions) et à l'expérience utilisateur.

Validation du Système :

- La documentation technique a été rédigée pour faciliter la maintenance et l'évolution future du projet.
- Une formation des utilisateurs finaux et des administrateurs a été réalisée, leur permettant de tirer pleinement parti du système.

- Les performances du système ont été optimisées pour répondre aux charges anticipées lors du déploiement.

Conclusion Générale et Perspectives

Le projet "Skill Marketplace" a été développé dans l'objectif de répondre à un besoin croissant : faciliter la mise en relation des compétences dans un cadre structuré et sécurisé. En s'appuyant sur une démarche rigoureuse et une mise en œuvre technique soignée, cette plateforme se positionne comme un outil pratique pour les freelances, entreprises et particuliers, leur permettant d'échanger et de collaborer efficacement.

Néanmoins, ce projet ne prétend pas être une finalité. À l'image de toute solution numérique, il offre un socle solide mais perfectible, propice à des évolutions futures.

Des pistes comme l'utilisation de l'intelligence artificielle pour améliorer la correspondance entre les utilisateurs ou le développement de nouvelles fonctionnalités interactives pourraient enrichir l'expérience et répondre à des attentes encore plus spécifiques.

En conclusion, "Skill Marketplace" se veut une première étape ambitieuse dans l'exploration des possibilités qu'offre la mise en réseau des compétences. Il est conçu pour évoluer, s'adapter et s'enrichir avec le temps, reflétant ainsi un esprit d'amélioration continue et d'innovation.

Ce projet est une invitation à aller plus loin, à concevoir de nouvelles solutions qui répondront encore mieux aux défis de demain.

