



République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Tunis El Manar
Faculté des Sciences de Tunis
Département des Sciences de l'informatique



RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention du
Diplôme National d'Ingénieur en Informatique :
Génie Logiciel
Par
Ziedi Safa

Conception et implémentation d'un module de gestion de notes de frais et de génération de factures électroniques

Organisme d'accueil : Tech-Expert

Président de jury : M. Khalil Gargouri

Rapporteur : Mme Manel Zekri

Encadrée par : Mme Naama Amdouni

Supervisée par : M. Mohamed Amine Othmani

Section - N° S03002

Dédicaces

Je dédie ce travail à la prunelle de mes yeux, mes très chers parents, pour tout leur amour, tendresse et sacrifices. Que Dieu, le Tout-Puissant puisse vous accorder santé, longue vie et bonheur.

À mon cher frère, en gage de ma profonde estime pour l'aide que tu m'as apportée. Tu m'as soutenue, réconfortée et encouragée. Puissent nos liens fraternels se consolider et se pérenniser encore plus.

À mes chères ami(e)s, Votre amitié est un honneur et une fierté pour moi. Je vous remercie d'avoir embellie ma vie par de précieux moments de bonheur. Merci de croire en moi.

Safa Ziedi

Remerciements

Après avoir rendu grâce à Dieu le Tout-Puissant et le Miséricordieux, je tiens à exprimer mes remerciements les plus vifs à mon encadrante académique, Madame Naama Amdouni, dont son sens de l'écoute, sa confiance et sa patience ont grandement contribué à l'accomplissement de mon projet de fin d'études.

Mes sincères remerciements s'adressent également à mes encadrants professionnels, Monsieur Tarek Kamoun et Monsieur Mohamed Amine Othmani, pour leurs soutiens sincères.

J'ai aussi le plaisir d'adresser mes vifs remerciements à toute l'équipe de Tech-expert pour l'accueil chaleureux.

Je profite de cette occasion pour exprimer ma gratitude envers le corps enseignant et le personnel administratif de la faculté des sciences de Tunis.

Finalement, je tiens à exprimer mes profondes gratitude et mes immenses respects à tous les membres du jury qui ont accepté d'évaluer ce modeste travail.

Table des matières

Introduction générale	1
1 Cadre général du projet	3
Introduction	4
1.1 Présentation de l'organisme d'accueil	4
1.2 Présentation de l'entreprise cliente Mindquest	4
1.3 Présentation du projet	5
1.3.1 Contexte du projet	5
1.3.2 Étude de l'existant	5
1.3.3 Critique de l'existant	6
1.3.4 Solution proposée	7
1.4 Méthodologie et processus de travail	7
1.4.1 Approches agiles	9
1.4.2 Choix méthodologique	10
1.4.3 SCRUM	10
1.5 Langage de Modélisation	12
Conclusion	12
2 Préparation du projet	13
Introduction	14
2.1 Identification des acteurs	14
2.2 Spécification des besoins	14
2.2.1 Spécifications des besoins fonctionnels	14
2.2.2 Spécification des besoins non fonctionnels	15
2.3 Diagramme de cas d'utilisation global	15
2.4 Pilotage du projet avec Scrum	16
2.4.1 Equipe et rôle	16
2.4.2 Le Backlog du produit	17
2.4.3 Planification des sprints	20

2.5	Environnement de travail	20
2.5.1	Environnement matériel	21
2.5.2	Environnement logiciel	21
2.6	Technologies utilisées	22
2.6.1	Technologie backend	22
2.6.2	Technologies front-End	23
2.6.3	Base de données	23
2.7	Architecture de la solution	23
2.7.1	Architecture physique	24
2.7.2	Architecture logique	25
	Conclusion	30
3	Sprint 1 : Gestion des dépenses sur la plateforme Mindquest	32
	Introduction	33
3.1	Sprint Planning	33
3.2	Backlog du Sprint 1	33
3.3	Analyse et conception	34
3.3.1	Diagramme de cas d'utilisation raffiné du Sprint 1	34
3.3.2	Diagramme de séquence	36
3.3.3	Diagramme de classes	37
3.4	Réalisation	38
3.4.1	La page initiale des notes de frais	38
3.4.2	Les pages de la liste des notes de frais	39
3.4.3	La page d'ajout d'une note de frais	41
3.4.4	La page d'ajout/modification d'une dépense dans une note de frais	42
3.4.5	Page d'acceptation/refus	43
	Conclusion	43
4	Sprint 2 : Gestion des notifications	44
	Introduction	45
4.1	Sprint Planning	45
4.2	Backlog du Sprint 2	46

4.3	Analyse et conception	46
4.3.1	Diagramme de cas d'utilisation raffiné du Sprint 2	46
4.3.2	Diagramme de séquence	48
4.4	Réalisation	48
4.4.1	Les notifications sur la plateforme	49
4.4.2	Les notifications par email	49
	Conclusion	52
5	Sprint 3 : Intégration et déploiement du module de notes de frais sur la plateforme Mindquest	
	Introduction	55
5.1	Sprint Planning	55
5.2	Backlog du Sprint 3	55
5.3	Analyse des besoins	56
5.3.1	Diagramme de cas d'utilisation raffiné du Sprint 3	56
5.4	Conception	58
5.5	Réalisation	58
	Conclusion	61
6	Sprint 4 : Développement de la partie Cockpit	62
	Introduction	63
6.1	Sprint Planning	63
6.2	Backlog du Sprint 4	63
6.3	Analyse détaillée du sprint 4	65
6.3.1	Diagramme de cas d'utilisation raffiné du Sprint 4	65
6.4	Conception	66
6.4.1	Diagramme de séquence	66
6.4.2	Diagramme de classes	66
6.5	Réalisation	67
6.5.1	Gestion des notes de frais sur la plateforme Cockpit	67
6.5.2	Page de modification	68
	Conclusion	69

7 Sprint 5 : Intégration et déploiement de la partie Cockpit	70
Introduction	71
7.1 Sprint Planning	71
7.2 Backlog du Sprint 5	71
7.3 Conception	72
7.3.1 Diagramme de déploiement	72
7.4 Réalisation	73
Conclusion	75
Conclusion générale	76
Netographie	77

Table des figures

1.1	Logo Tech-Expert [1]	4
1.2	Logo Mindquest [3]	4
1.3	Capture de la solution existante [4]	6
1.4	Les principes Agiles [5]	9
1.5	Feuille de route du framework scrum [6]	11
2.1	Diagramme de cas d'utilisation global	16
2.2	Architecture physique	24
2.3	Architecture logique	25
2.4	L'architecture REST [16]	26
2.5	L'architecture GraphQL [18]	27
2.6	Spécification Serveur GraphQL [19]	28
2.7	Architecture Apollo client [21]	29
2.8	Redux [22]	30
3.1	Diagramme cas d'utilisation détaillé de "Gestion des dépenses"	35
3.2	Diagramme de séquence détaillé de "Ajouter une note de frais"	37
3.3	Diagramme de classes du sprint 1	38
3.4	La page des notes de frais - freelancer	39
3.5	La page des notes de frais - client	39
3.6	Liste des notes de frais - freelancer	40
3.7	Liste des notes de frais - client	40
3.8	Filtrer les notes de frais en fonction du statut	41
3.9	Rechercher note de frais	41
3.10	La page d'ajout d'une note de frais	42
3.11	La page d'ajout d'une dépense dans une note de frais	42
3.12	La page de modification d'une dépense	43
3.13	Page d'acceptation/refus	43
4.1	Gestion des notifications par email	45

4.2	Diagramme de cas d'utilisation raffiné du Sprint 2	47
4.3	Diagramme de séquence des notifications	48
4.4	Notification pour un client	49
4.5	Email Rappel : Note de frais en brouillon depuis 2 semaines	50
4.6	Email de confirmation d'envoi de note de frais	50
4.7	Email : Nouvelle note de frais	51
4.8	Email vers client : note de frais validée	51
4.9	Email vers freelancer : Acceptation Note de Frais	52
4.10	Email vers freelancer : Refus de Note de Frais	52
5.1	Diagramme de cas d'utilisation raffiné du Sprint 3	57
5.2	Diagramme de paquets pour la gestion des notes de frais	58
5.3	Page de connexion sur Mindquest	59
5.4	Espace des notes de frais sur Mindquest	59
5.5	Note de frais en détail	60
5.6	Téléchargement du reçu	60
5.7	Email de confirmation en anglais	61
6.1	Diagramme de cas d'utilisation raffiné du Sprint 4	65
6.2	Diagramme de séquence détaillé de "Accepter une note de frais"	66
6.3	Diagramme de classes du Sprint 4	66
6.4	Page de gestion des notes de frais sur la plateforme Cockpit	67
6.5	Liste des notes de frais filtrées sur la page	67
6.6	Page de détail d'une note de frais	68
6.7	Formulaire de modification d'une note de frais	68
6.8	Formulaire de modification d'une dépense	69
7.1	Diagramme de déploiement Sprint 5	73
7.2	Page de connexion sur Cockpit	73
7.3	Espace des notes de frais sur Cockpit	74
7.4	Les permissions	74
7.5	Accorder une permission	75

Liste des tableaux

1.1	Agile vs Cycle en V	8
1.2	Tableau comparatif des processus agiles	10
2.1	Backlog du produit	20
2.2	Plan des sprints	20
2.3	Environnement matériel	21
3.1	Backlog du Sprint 1	34
3.2	Description textuelle du cas d'utilisation «Ajouter une note de frais»	36
4.1	Backlog du Sprint 2	46
4.2	Description textuelle du cas d'utilisation «Recevoir notification de refus»	47
5.1	Backlog du Sprint 3	56
5.2	Description textuelle du cas d'utilisation «Voir reçu d'une dépense»	58
6.1	Backlog du Sprint 4	64
6.2	Description textuelle du cas d'utilisation «Accepter une note de frais»	65
7.1	Backlog du Sprint 5	72

Liste des abréviations

- **API** = Application Programming Interface
- **GUI** = Graphical User Interface
- **GraphQL** = Graph Query Language
- **IHM** = Interface Homme-Machine
- **MERN** = MongoDB, Express, React, and Node. js
- **MS** = Microservice
- **NoSql** = Not only SQL
- **PBI** = Product Backlog Item
- **REST** = Representational State Transfer
- **SQL** = Structured Query Language
- **TVA** = Taxe sur la Valeur Ajoutée
- **UI** = User Interface
- **XP** = eXtreme Programming

Introduction générale

Autrefois, le freelance était moins populaire. Les freelancers trouvaient du travail régulier, mais leur portée était limitée à leurs villes proches. Le plus grand défi était de créer un réseau pour obtenir des missions continues, rendant le concept du freelancing très restreint.

Aujourd’hui, grâce aux sociétés d’investissement et aux fédérations de consultants indépendants en technologies de l’information, le freelancing gagne en popularité. Des sites web, comme Mindquest et Upwork, facilitent la coordination entre freelancers et clients, aidant les indépendants à développer leurs activités spécialisées.

Pour offrir des services efficaces et à jour, ces entreprises doivent bien connaître le marché des freelances et maintenir une base de données dynamique et précise des freelancers.

Notre projet de fin d’études vise à concevoir et développer un module de gestion des notes de frais pour la plateforme web de Mindquest. Mindquest, un service de recrutement en freelance, facilite la mise en relation entre entreprises et freelances. Ce module permettra aux freelances en mission de soumettre leurs frais pour remboursement par le client.

Ce rapport détaille les étapes de notre travail, structuré en sept chapitres.

- Chapitre 1 : Cadre Général du Projet. Il présente l’organisme d’accueil, décrit le contexte de notre projet ainsi que la méthodologie adoptée.
- Chapitre 2 : Préparation du projet, présente les acteurs du système, les besoins fonctionnels et non fonctionnels, le backlog du produit, le diagramme de cas d’utilisation global, ainsi que la description de l’architecture et des technologies choisies pour réaliser notre solution.
- Le chapitre 3 intitulé « Sprint 1 : Gestion des dépenses sur la plateforme Mindquest » Ce chapitre s’intéresse à l’étude et à la réalisation du premier sprint de notre projet. Il présente le backlog, la description, la conception et la réalisation de ce dernier.
- Le chapitre 4 intitulé « Sprint 2 : Gestion des notifications » Ce chapitre se concentre sur le développement des fonctionnalités liées aux notifications. Il expose le backlog du sprint, le cas d’utilisation du sprint ainsi que le diagramme de séquence.
- Le chapitre 5 intitulé « Sprint 3 : Intégration et déploiement du module de notes de frais sur la plateforme Mindquest » Ce chapitre traite l’intégration du module dans l’ensemble des autres modules puis déploiement du module de gestion des notes de frais sur la plateforme Mindquest.

Introduction générale

Il couvre les aspects techniques et opérationnels du déploiement, les tests effectués, ainsi que les défis rencontrés et les solutions apportées.

- Le chapitre 6 intitulé « Sprint 4 : Développement de la partie Cockpit » Ce chapitre porte sur le développement de la partie Cockpit, un tableau de bord pour la gestion et le suivi des dépenses. Il inclut le backlog du sprint, la conception de l'interface utilisateur, les fonctionnalités implémentées et les tests réalisés.
- Le chapitre 7 intitulé « Sprint 5 : Intégration et déploiement de la partie Cockpit » Le dernier chapitre aborde le déploiement de la partie Cockpit. Il décrit le processus de mise en production et les ajustements finaux.

Nous clôturons notre rapport par une conclusion générale sur le travail effectué et sur les perspectives envisagées.

CHAPITRE 1

CADRE GÉNÉRAL DU PROJET

Plan

Introduction	4
1 Présentation de l'organisme d'accueil	4
2 Présentation de l'entreprise cliente Mindquest	4
3 Présentation du projet	5
4 Méthodologie et processus de travail	7
5 Langage de Modélisation	12
Conclusion	12

Introduction

Au niveau de ce chapitre, nous présenterons le contexte général du projet. Initialement, nous présentons l'organisme d'accueil Tech-Expert, ainsi que son client principal l'entreprise Mindquest. Par la suite, nous allons dévoiler l'étude de la problématique et l'étude de l'existant. En dernier lieu, nous illustrons la méthodologie du travail adoptée.

1.1 Présentation de l'organisme d'accueil

Tech-Expert est une société jeune et dynamique. Elle se démarque en tant que société de conseil et de développement informatique qui assiste principalement ses clients dans le domaine du recrutement, en faisant du numérique un véritable moteur d'accélération de leur entreprise. Tech-Expert propose un soutien agile complet, depuis l'élaboration de l'architecture jusqu'à l'utilisation des applications en production, en passant par le développement complet. Sa compétence technique englobe également des domaines sophistiqués tels que l'intelligence artificielle, l'ingénierie des données et l'intégration d'applications pour les entreprises.



FIGURE 1.1 : Logo Tech-Expert [1]

1.2 Présentation de l'entreprise cliente Mindquest

Mindquest est une société de placement de consultants indépendants dans le domaine des technologies de l'information. Créeée en 2014, elle a son siège à Londres, au Royaume-Uni, et possède des bureaux à Paris, France et Barcelone, Espagne [2].

L'entreprise se concentre sur le recrutement dans l'ensemble des marchés informatiques, notamment dans les domaines du génie logiciel, de l'analyse des données, des systèmes d'information, de la Business Intelligence et de la gestion de projet.



FIGURE 1.2 : Logo Mindquest [3]

1.3 Présentation du projet

Au cours de cette partie, nous présenterons le contexte du projet, puis nous étudierons la situation existante et la critiquerons afin de pouvoir présenter notre solution.

1.3.1 Contexte du projet

Mindquest est un service de recrutement en CDI et en freelance, visant à faire un matching parfait entre entreprises et talents, en trouvant les meilleurs profils pour les clients et les meilleures opportunités pour les candidats. Mindquest opère via deux plateformes distinctes.

La première est la plateforme principale de Mindquest, accessible aux candidats freelance en informatique et aux clients cherchant des talents. La seconde, Cockpit, est destinée à l'équipe interne de Mindquest (marketing, ventes, managers, sourceurs, administrateurs) pour la gestion interne.

Lorsqu'une offre est publiée, un candidat peut postuler pour une mission. Si un client s'intéresse à un profil, l'application envoie les informations à la plateforme Cockpit, où les sourceurs sauvegardent les données du candidat, du client et de l'offre. Le sourceur peut ensuite discuter avec le candidat pour entamer le processus de recrutement.

Une fois accepté, le consultant a la possibilité de soumettre chaque mois les frais au client directement ou à la société de portage pour remboursement.

Nous avons pour mission de développer un module de gestion des notes de frais qui sera intégré à la plateforme Web digitale de Mindquest et à la plateforme administrative Cockpit.

1.3.2 Étude de l'existant

Dans le cadre de ses activités professionnelles, un freelancer est souvent amené à engager des dépenses pour lesquelles il doit ensuite demander un remboursement en fournissant les justificatifs nécessaires.

Actuellement, ce processus est géré à l'aide d'un fichier Excel chez Mindquest. Le freelancer soumet ses notes de frais à l'entreprise cliente avec laquelle il travaille pour obtenir un remboursement.

mindquest

NOTES DE FRAIS

NOM DU CONSULTANT	HAUE
PRENOM DU CONSULTANT	TARIK
FONCTION	SAP Consultant
NUMERO DE CONTRAT	

Date du déplacement	Lieu du déplacement	Type de dépenses	Divers	Montant	TVA
11/14/2023	MONTOIR	Train Aller		95	
11/17/2023	MONTOIR	Train Retour		95	
11/15/2023	MONTOIR	Repas		18.2	
Du 14/11/2023 -> 17/11/2023	MONTOIR	Hébergement		399.33	
11/14/2023	MONTOIR	Taxi		400	
	MONTOIR				
	MONTANT TOTAL			1007.55	0.00

REMARQUES

SIGNATURE

Tarik HAUE



Contact :

facture@mindquest.io

FIGURE 1.3 : Capture de la solution existante [4]

Le freelance renseigne manuellement ce template de note de frais au format Excel. Ensuite, il envoie par mail au sourceur une demande de remboursement de note de frais. Le sourceur examine la demande et notifie par mail le freelance de l'acceptation ou du rejet de sa demande de note de frais.

1.3.3 Critique de l'existant

Bien que l'utilisation d'un fichier Excel pour gérer le remboursement des dépenses puisse sembler pratique, elle présente des inconvénients majeurs. Cette méthode expose le processus à des risques de perte de données, de corruption de fichiers et de difficultés de partage. De plus, le processus manuel basé sur Excel, suivi des échanges de mails, est vulnérable aux erreurs humaines, entraînant des retards et des frustrations.

L'absence de fonctionnalités avancées, comme la gestion automatisée des workflows et la génération de rapports personnalisés, limite l'efficacité du processus. La collaboration entre les

comptables, les ressources humaines et les collaborateurs est entravée par les limitations de partage en temps réel.

Le manque d'automatisation nécessite de nombreuses tâches manuelles, telles que la vérification des justificatifs et la notification des parties prenantes, entraînant une utilisation inefficace des ressources et des retards.

En conclusion, bien qu'un fichier Excel ait pu servir temporairement, ses limitations en termes d'efficacité, de précision et de collaboration soulignent la nécessité d'alternatives plus avancées et automatisées pour améliorer la gestion des dépenses et des remboursements.

1.3.4 Solution proposée

Pour résoudre les défis liés à la gestion manuelle des dépenses et des remboursements chez Mindquest, nous proposons la création et l'intégration d'un outil dédié de gestion des notes de frais dans la plateforme existante. Cet outil automatisera et simplifiera le suivi, la gestion et le traitement des dépenses des freelancers.

L'outil offrira une interface conviviale pour saisir les dépenses et joindre les justificatifs. Il intégrera des fonctionnalités avancées comme la gestion automatisée des workflows, la génération des factures et la notification en temps réel des parties prenantes.

Pour une intégration fluide avec le système actuel de Mindquest, l'outil synchronisera les données et facilitera la collaboration entre les différents départements impliqués dans le processus de remboursement.

Cette solution vise à améliorer l'efficacité opérationnelle, la précision des données et la collaboration, tout en réduisant les erreurs humaines, optimisant les workflows et assurant une meilleure traçabilité des dépenses.

1.4 Méthodologie et processus de travail

Dans le domaine de la gestion de projet, deux principaux paradigmes méthodologiques émergent : le classique et l'agile. La méthode traditionnelle, souvent désignée sous les noms de cascade, cycle en V, repose sur une planification exhaustive avant le début du projet. Une fois le cahier des charges établi et le contrat signé, les développeurs sont chargés de suivre scrupuleusement le plan établi. En revanche, les méthodologies agiles adoptent une approche plus flexible, permettant des ajustements et des réorientations au fur et à mesure de l'avancement du projet.

Dans notre démarche de sélection de la méthodologie la plus adaptée, nous avons comparé de manière approfondie la méthode Agile et la méthode cycle en V, comme illustré dans le tableau 1.1. Cette analyse nous a permis de déterminer la méthode la plus efficace en fonction des exigences et des contraintes spécifiques du projet.

	Cycle en V	Agile
Cycle de vie	Phases séquentielles sprint	Processus itératif
Livraison	A la fin de la réalisation de toutes les fonctionnalités => livraison tardive	A chaque livraison partielle au client
Contrôle qualité	A la livraison finale(fin du cycle de développement) => effet tunnel	A chaque livraison partielle au client
Spécification	Pas de changement possible sans revenir à la phase de spécifications et repasser par toutes les autres phases => délais et coûts supplémentaires	Spécifications plus souples en ajoutant ou en modifiant les fonctionnalités
Planification	Plans détaillés stables définies dès le début du projet	Planification adaptative qui tolère les modifications en fonction des nouvelles demandes
Equipe	Intervention uniquement dans la phase de développement	échange et prises de décisions collectives par l'équipe

TABLEAU 1.1 : Agile vs Cycle en V

Après cette comparaison, nous avons opté pour la méthode Agile pour la gestion du notre processus de développement.

1.4.1 Approches agiles

Les méthodologies agiles représentent des ensembles de pratiques utilisées dans le développement de projets informatiques, avec lesquelles une équipe peut finaliser un projet en le découplant en étapes tout en impliquant au maximum le client et en assurant une grande réactivité à ses demandes, par la suite prioriser sa satisfaction réelle aux termes d'un contrat de développement.

Les méthodes agiles sont basées sur un cycle de développement commun : itératif, incrémental et adaptatif. À chaque module, l'équipe suit un processus de planification, d'exécution et d'évaluation sur de courtes sessions de deux à quatre semaines pour que le produit final serait correspondant aux besoins du client.

Les principes fondamentaux des méthodologies agiles sont les suivants :



FIGURE 1.4 : Les principes Agiles [5]

1.4.2 Choix méthodologique

Critères	Scrum	Kanban	XP
Planification	Au début de chaque sprint	Kanban board Flux continu	Planning game
Estimation de l'effort	Au début de chaque sprint	Optionnel prédictibilité	Pratiques XP
Rôles	Scrum master (SM) Product owner (PO) Développeur (Dev)	Team	Team + client
Quand choisir ?	Équipe travaillant sur un seul projet	Équipe travaillant sur plusieurs projets simultanément	Amélioration de la qualité du logiciel critique
Top 3 bénéfices	Productivité Scalabilité Engagement des équipes	Mise en place rapide sans changement des processus existants Pilotage Visuel	Qualité du code plus importante Réactivité Niveau d'expertise des équipes
Boards Artifacts	Product backlog Scrum Board Burndown/burnup	Kanban board Diagramme des flux cumulés	Priorisation par le client Pratiques XP

TABLEAU 1.2 : Tableau comparatif des processus agiles

Pour réussir notre projet dans les conditions les plus favorables et pour clarifier toute ambiguïté en cours de la réalisation, la collaboration et l'échange quotidien avec les autres membres de l'équipe est primordial.

Après avoir étudier le tableau comparatif des processus agiles et fait la liaison avec notre projet, nous avons choisi le framework Scrum comme méthodologie de travail.

Nous présentons ci-après les différents intervenants dans notre projet et la feuille de route du framework Scrum.

1.4.3 SCRUM

Scrum est un framework agile qui implique une équipe unifiée guidant le projet tout au long de son développement pour atteindre un objectif défini. Cette méthode se révèle à la fois dynamique et

productive, favorisant la réalisation itérative des fonctionnalités tout en impliquant la participation active du client. Chaque itération peut durer de deux à quatre semaines, à la fin de chaque sprint un produit fonctionnel doit être livré. En effet, scrum établit trois rôles principaux, à savoir :

- **Responsable Produit (Product Owner)** : Il représente officiellement le client au sein du projet SCRUM. Il définit et priorise les besoins du produit.
- **SCRUM Master** : Il facilite l'application du framework SCRUM et veille à ce que ses objectifs soient respectés. Responsable de la communication, il ne joue pas le rôle de chef de projet mais s'occupe de lever les obstacles qui pourraient entraver le progrès de l'équipe et du projet pendant les différents Sprints.
- **Les développeurs** : Chargés de la réalisation des Sprints et des livrables, ils doivent collaborer et être autonomes pour répondre aux exigences du client et aux engagements pris.

La figure suivante présente le déroulement de la gestion de projet par Scrum :

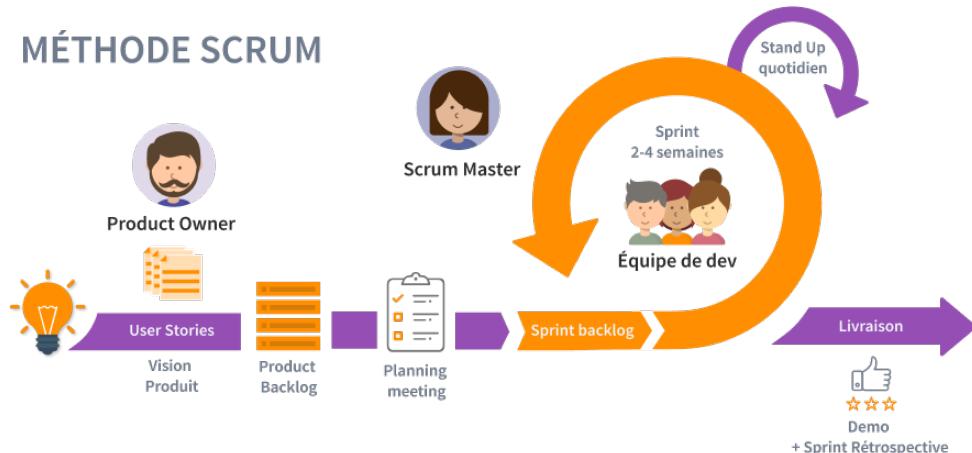


FIGURE 1.5 : Feuille de route du framework scrum [6]

Plusieurs réunions sont organisées pour permettre la communication entre les acteurs mentionnés.

Parmi elles :

- **Réunion pour la planification du Sprint (Sprint Planning Meeting)** : En début de Sprint, cette réunion cruciale permet à l'équipe projet et au responsable produit de définir les objectifs du Sprint en se basant sur le "Backlog du produit" priorisé et d'estimer la capacité de production nécessaire.
- **Mélée quotidienne (Daily SCRUM)** : Chaque jour, les membres de l'équipe projet se réunissent pendant environ 15 minutes pour discuter de leurs activités de la veille, des obstacles rencontrés et des tâches à accomplir le jour-même.

- **Revue de Sprint (Sprint Review Meeting)** : À la fin du Sprint, l'équipe projet présente les fonctionnalités réalisées et recueille les retours du Product Owner et des utilisateurs finaux. C'est également le moment propice pour anticiper le périmètre des prochains Sprints et pour ajuster la planification de la sortie (release) si nécessaire.
- **Rétrospective de Sprint (Sprint Retrospective Meeting)** : À la fin de chaque Sprint, le SCRUM Master anime cette réunion où l'équipe projet évalue son comportement et sa performance dans le processus de développement, et prend des mesures pour améliorer les prochains Sprints.

Les artefacts de communication font également partie du cycle de vie d'un projet développé selon la méthodologie SCRUM, comme illustré dans la figure 3. Parmi ces artefacts, nous trouvons :

- **Backlog du produit (Product Backlog)** : Liste des fonctionnalités souhaitées par le client sous forme d'User Stories. Visible par toutes les parties prenantes, elles peuvent y ajouter des éléments. Les priorités sont constamment mises à jour par le Product Owner.
- **Backlog du Sprint (Sprint Backlog)** : Il contient les exigences à réaliser, négociées lors de la planification du Sprint entre le responsable produit et l'équipe projet. Visible par toute l'équipe, il sert de référence lors des mêlées quotidiennes.

1.5 Langage de Modélisation

Tout au long de ce projet, nous utiliserons UML (Unified Modeling Language) comme langage de modélisation. Il s'agit d'un langage graphique basé sur des pictogrammes conçu pour visualiser la conception d'un système sous divers aspects. Il est largement utilisé dans le développement logiciel et peut être appliqué à tout type de système.

Conclusion

Ce chapitre nous a permis d'introduire les notions de base de notre projet. Nous avons d'abord présenté la société Tech-Expert, qui accueille le projet, ainsi que la société cliente, Mindquest. Ensuite, nous avons exposé notre sujet, suivi d'une analyse et d'une critique de l'existant. Enfin, nous avons détaillé la méthodologie de travail à adopter.

CHAPITRE 2

PRÉPARATION DU PROJET

Plan

Introduction	14
1 Identification des acteurs	14
2 Spécification des besoins	14
3 Diagramme de cas d'utilisation global	15
4 Pilotage du projet avec Scrum	16
5 Environnement de travail	20
6 Technologies utilisées	22
7 Architecture de la solution	23
Conclusion	30

Introduction

Dans ce chapitre, nous aurons une vision plus claire du sujet. Pour cela, nous procèderons à l'identification des acteurs de notre système. Puis, nous nous focaliserons sur la capture de ses besoins fonctionnels et non fonctionnels, au pilotage de projet avec Scrum, par élaboration du Backlog du produit ainsi qu'à la description de l'environnement de travail.

2.1 Identification des acteurs

Un acteur est une entité externe qui intervient dans le système et lance les opérations que ce dernier lui fournit. Dans cette section, nous présentons les acteurs de notre système :

- **freelancer** : Son principal rôle est d'envoyer sa note de frais pour une mission en cours.
- **client** : Son rôle consiste à accepter ou refuser les notes de frais des freelances avec lesquels il collabore, en vue de les rembourser.
- **sourceurs** : sont l'équipe d'employés assurant le matching entre les freelances et les entreprises. Ils agissent en tant qu'administrateurs de notre plateforme.

2.2 Spécification des besoins

La spécification des besoins est une série d'informations qui décrivent, sans ambiguïté, le fonctionnement d'un système. Ils se répartissent en deux volets essentiels : besoins fonctionnels et besoins non fonctionnels.

2.2.1 Spécifications des besoins fonctionnels

Les besoins fonctionnels expriment les actions que doit effectuer notre module :

- Le système donne au freelancer la possibilité de créer des notes de frais depuis le Dashboard Mindquest, sachant qu'une note de frais est un ensemble de dépenses.
- Le système offre au freelancer la possibilité d'ajouter une ou plusieurs dépenses à une note de frais, en y joignant leurs reçus.
- Le système donne au freelancer la possibilité d'envoyer ses notes de frais afin que le client la valide depuis le Dashboard Mindquest.
- Le système donne au client la possibilité d'accepter ou refuser une note de frais depuis le Dashboard Mindquest.

- Le système donne au freelancer la possibilité de modifier une note de frais suite à un refus afin que le client les valide depuis le Dashboard Mindquest.
- Le système donne aux freelancers, clients et admins la possibilité de retrouver l'ensemble de leurs notes de frais selon leurs statuts (brouillon, en attente, validée, rejetée).
- Le système envoie des e-mails aux freelancers et clients à chaque changement de statut.
- Le système affiche une notification aux clients concernant les nouvelles notes de frais dans le Dashboard de Mindquest.
- Le système permet aux sourceurs de suivre les notes de frais de chaque freelancer et leurs statuts, via la plateforme interne, appelée cockpit.

2.2.2 Spécification des besoins non fonctionnels

Ce sont les exigences permettant de juger un fonctionnement du système et de définir le taux de satisfaction du client. Les exigences non fonctionnelles liées à notre solution sont :

- **L'ergonomie** : Les diverses fonctionnalités du système doivent être intelligibles et faciles à traiter par l'utilisateur. Le système doit fournir à l'utilisateur une meilleure expérience.
- **La disponibilité** : Nos modules doivent être toujours disponibles pour tout usager.
- **La fiabilité** : Tous les modules doivent garantir des fonctionnalités correctes et sans aucune erreur.
- **La sécurité** : Notre système met en place des mesures de sécurité robustes pour éviter toute faille d'accès aux données.
- **La performance** : Le temps de réponse du système est l'un des priorités dans notre projet .
- **La maintenabilité** : Le code devra être lisible, commenté et réparti selon les tâches à réaliser.

2.3 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation global définit les fonctionnalités dont disposent les utilisateurs par notre micro service de gestion des notes de frais.

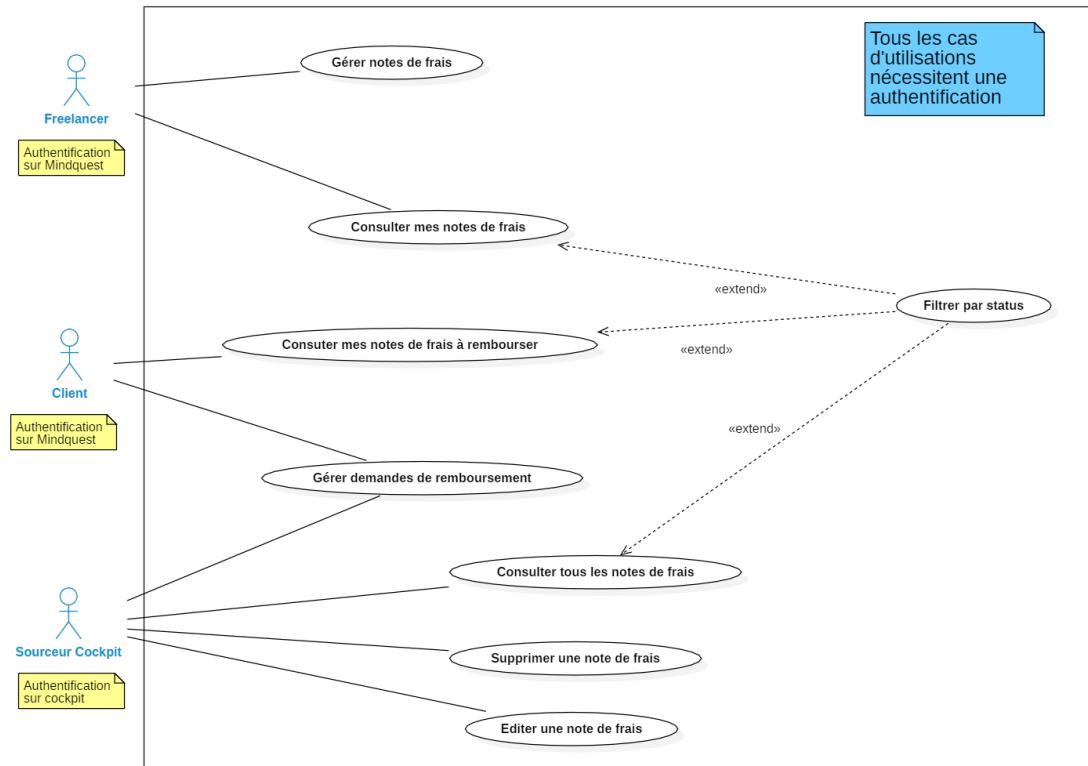


FIGURE 2.1 : Diagramme de cas d'utilisation global

2.4 Pilotage du projet avec Scrum

Dans ce qui suit, nous allons identifier les différents membres de l'équipe Scrum qui va mener à terme le présent projet. Puis, nous allons établir le backlog du produit. Enfin, nous allons clôturer la présente section par la planification des différents sprints.

2.4.1 Équipe et rôle

Dans notre équipe Scrum nous avons trois rôles :

- **Le product owner ou Propriétaire du produit** : le représentant officiel du client M.Felix. Il spécifie les besoins du client et détermine les spécifications du produit.
- **Le scrum master (Maître de la mêlée)** : M.Mohamed Amine Othmani chargé de la coordination du flux de travail.
- **L'équipe de développement** : Safa Ziedi.

2.4.2 Le Backlog du produit

Le Backlog du produit est une liste hiérarchisée de fonctionnalités à développer en tant que partie d'un produit informatique. Le tableau 2.1 illustre le Product Backlog de notre solution. Il contient un identifiant unique pour chaque événement (ID), le titre de l'élément (PBI), l'identifiant de chaque user story aboutissant à la fonctionnalité visée (User Story ID), la description de ce que l'acteur souhaitait réaliser (User Story) et un attribut permettant de définir la priorité de chaque user story (Priorité) qui varie entre 1 et 3 « la priorité la plus importante est 3 et la moins importante est 1 ». De plus, une estimation de l'effort nécessaire en Homme jour (HJ) est également fournie.

ID	PBI	US id	User Story	Priorité	Estimation(HJ)
1	Gestion des dépenses sur la plateforme Mindquest	1.1	En tant freelancer/client, je peux accéder à l'espace des notes de frais.	3	2.5
		1.2	En tant que freelancer, je peux créer une note de frais liée à une mission en cours.	3	4.5
		1.3	En tant que freelancer, je peux ajouter une dépense dans une note de frais et la modifier.	3	2.5
		1.4	En tant que freelancer/client, j'ai la possibilité de filtrer les notes de frais en fonction de leur statut.	1	3
		1.5	En tant que freelancer, je peux envoyer une note de frais au client.	3	2.5
		1.6	En tant que client, je peux accepter/refuser une note de frais.	3	2.5
		1.7	En tant que freelancer, je peux modifier une note de frais rejetée.	2	2.5

2	Gestion des notifications	2.1	En tant que client, je souhaite recevoir un email pour me notifier qu'une nouvelle note de frais est en attente de validation.	3	5.5
		2.2	En tant que freelancer, je souhaite recevoir un email de rappel d'envoi si la note de frais est en brouillon depuis 2 semaines.	1	3.5
		2.3	En tant que freelancer/client, je souhaite recevoir un compte rendu (montant total des notes de frais et les reçus de chaque dépense) une fois qu'une note de frais est validée.	3	4
		2.4	En tant que freelancer, je souhaite recevoir un email de notification si ma note de frais est refusée.	2	4
		2.5	En tant que client, je souhaite recevoir une notification concernant les nouvelles notes de frais dans le Dashboard de Mindquest.	3	3
3	Intégration et déploiement du module de notes de frais sur la plateforme Mindquest	3.1	En tant que freelancer/client, je peux accéder à l'espace des notes de frais dans Mindquest.	3	6.5

		3.2	En tant que freelancer/client, je peux changer la langue de la page des notes de frais.	3	3
		3.3	En tant que freelancer/client, je souhaite recevoir les emails dans la langue que j'ai sélectionnée sur le site.	2	5
		3.4	En tant que client, je peux voir un reçu d'une dépense.	1	5.5
4	Développement de la partie Cockpit	4.1	En tant qu'Admin (sourceur), je peux consulter toutes les notes de frais.	3	2.5
		4.2	En tant que sourceur, je peux filtrer en utilisant tous les champs disponibles.	1	3
		4.3	En tant que sourceur, je peux modifier une note de frais.	2	3
		4.4	En tant que sourceur, je peux modifier une dépense.	2	4
		4.5	En tant que sourceur, je peux accepter/refuser une note de frais.	3	1
		4.6	En tant que sourceur, je peux supprimer une note de frais avec ses dépenses.	1	3
		4.7	En tant que sourceur, je peux supprimer une dépense.	1	3
5	Intégration et déploiement de la partie Cockpit	5.1	En tant que sourceur, je peux accéder à l'espace des notes de frais dans la plateforme Cockpit.	3	5.5

	5.2	En tant que sourceur, je peux ajouter des permissions sur les notes de frais.	3	2
	5.3	En tant que sourceur, je peux accorder une permission sur les notes de frais à un autre sourceur.	2	1

TABLEAU 2.1 : Backlog du produit

2.4.3 Planification des sprints

Nous organisons désormais les différentes tâches de notre projet en sprints. À chaque sprint, nous sélectionnons une partie des éléments du backlog produit à développer. Le tableau 2.2 présente la répartition des fonctionnalités sur six sprints distincts, avec une estimation effectuée par semaine.

Le sprint 0 est dédié à l'étude théorique du projet. Il comprend la lecture du cahier des charges, la compréhension des différentes fonctionnalités de la plateforme, l'analyse du code existant et des différents microservices avec lesquels je vais travailler, l'étude technologique, ainsi que la conception de la solution.

ID Sprint	Nom Sprint	Durée (semaines)
1	Gestion des dépenses	4
2	Gestion des notifications	4
3	Intégration et déploiement du module de notes de frais sur la plateforme Mindquest	4
4	Développement de la partie Cockpit	4
5	Intégration et déploiement de la partie Cockpit	2

TABLEAU 2.2 : Plan des sprints

2.5 Environnement de travail

Tout au long de la réalisation de notre plateforme, nous avons utilisé des technologies et des outils bien particuliers que nous présentons dans ce qui suit.

2.5.1 Environnement matériel

Composant	Description
PC portable	Hp Laptop 15s-fq2xxx
RAM	16 Go
Micropuceur	Intel® Core™ i7-1165G7 @ 2.80 GHz
Disque dur	1 To
Système d'exploitation	Windows 11

TABLEAU 2.3 : Environnement matériel

2.5.2 Environnement logiciel

2.5.2.1 Outils de développement et modélisation

Pour cette partie, nous nous intéressons aux IDE et outils employés tout au long de la réalisation de notre Solution :

- **Visual Studio Code (VSC)** : Éditeur de code open-source, gratuit et multiplateforme développé par Microsoft. Principalement conçu pour le développement d'applications avec JavaScript, TypeScript et Node.js. Il peut s'adapter à d'autres langages grâce à un système d'extensions bien fourni. Il inclut la coloration syntaxique et l'auto-complétion du code[7].
- **Interface utilisateur graphique MongoDB (Compass)** : Compass est un outil interactif gratuit qui permet d'interroger, d'optimiser et d'analyser vos données MongoDB. Obtenez des informations clés, effectuez un glisser-déposer pour créer des pipelines, et plus encore[8].
- **Git** : Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Il s'agit d'un projet open source avancé, bénéficiant d'un développement actif. Initialement conçu en 2005 par Linus Torvalds, le créateur renommé du noyau du système d'exploitation Linux, Git est devenu un pilier essentiel pour la gestion des versions dans de nombreux projets logiciels, qu'ils soient commerciaux ou open source[9].

- **Bitbucket** : Un service d'hébergement de code et de collaboration basé sur Git, spécialement conçu pour les équipes de développement[10].
- **Overleaf** : Un éditeur LaTeX en ligne utilisé pour la création et la modification de documents LaTeX, notamment pour la rédaction de ce rapport[11].
- **Star UML** : Un logiciel de modélisation qui a été précieux pour créer les différents diagrammes présentés dans ce rapport.
- **Jira** : Un outil de gestion de projet Agile, prenant en charge diverses méthodologies Agile, y compris Scrum. Utilisé pour la planification et le suivi du projet.

L'équipe de Tech-Expert utilise Jira pour la gestion de projet et le suivi des tickets, afin d'assurer une planification efficace et un suivi rigoureux du travail. Les tâches sont exécutées comme suit : chaque tâche assignée à un membre de l'équipe doit être réalisée dans les délais convenus. Ensuite, un pull request est soumis à l'expert Hybris pour validation du code. Une fois validé, le code est fusionné dans la branche principale. Enfin, la tâche est attribuée à un testeur pour effectuer les tests fonctionnels nécessaires.

2.6 Technologies utilisées

Nous avons utilisé un certain nombre de frameworks et de bibliothèques dans le développement de la solution. Nous donnons dans ce qui suit leurs définitions et leurs caractéristiques principales.

2.6.1 Technologie backend

Pour le développement de la partie serveur, nous avons choisi Node.js avec le framework Express.js. Node.js est un environnement d'exécution JavaScript qui permet de construire des applications back-end rapides et évolutives. Express.js facilite le développement d'API robustes et flexibles.

- **TypeScript** : TypeScript est un langage de programmation open source développé par Microsoft. Il améliore et sécurise la création de code JavaScript. En tant que sur-ensemble de JavaScript, TypeScript permet l'utilisation de tout code JavaScript valide. Une fois écrit en TypeScript, le code est transcompilé en JavaScript, ce qui lui permet d'être interprété par n'importe quel navigateur web ou moteur JavaScript [12].
- **Express.js** : Express.js, souvent appelé simplement « Express », est un framework backend pour Node.js. Il se distingue par son minimalisme et sa rapidité, offrant des fonctionnalités solides pour créer des applications backend évolutives. Express fournit un système de routage

et des outils simplifiés permettant d'étendre facilement le framework pour répondre aux besoins spécifiques de l'application [13].

2.6.2 Technologies front-End

- **React Js :** React est une bibliothèque JavaScript pour la construction d'interfaces utilisateur (UI).

2.6.3 Base de données

MongoDB est une base de données NoSQL open source. Puisqu'il s'agit d'une base de données non relationnelle, elle peut traiter des données structurées, semi-structurées et non structurées. Elle utilise un modèle de données non relationnel, orienté document, et un langage de requête non structuré. [14]

- NoSQL : fait référence aux bases de données non fondées sur l'architecture classique des bases de données relationnelles.
- Orienté Document : Au contraire des bases de données relationnelles dans lesquelles les données sont sauvegardées sous forme de relations ou de table, MongoDB représente une base de données orientée documents. L'unité de sauvegarde (assimilable à une ligne) est un document ou un objet, et plusieurs documents sont conservés dans des collections (assimilable à un tableau). Chaque document d'une collection dispose d'un identifiant unique qui permet d'y accéder. Cette identification est automatiquement indexée.
- Schéma flexible : les champs peuvent être rapidement ajoutés dans notre code d'application sans devoir se soucier des scripts de migration de base de données. Ce choix de technologie est fondé sur la capacité de MongoDB à soutenir des données de grande taille. Il s'agit d'une base de données plus puissante que d'autres bases de données relationnelles telles que SQL.

2.7 Architecture de la solution

Dans cette section, nous présentons l'architecture physique et l'architecture logique de notre solution en identifiant ses différents éléments et comment ils communiquent et interagissent les uns avec les autres.

2.7.1 Architecture physique

Nous présentons dans la figure 2.1 l'architecture physique de notre application qui n'est autre qu'une architecture N-tiers. L'architecture N-tiers, également appelée architecture à multi niveaux, est une architecture dans laquelle coexistent et sont maintenus des modules indépendants permettant le rendu d'une interface utilisateur (GUI), les processus logiques, fonctionnels et métiers ainsi que l'accès aux données. Les caractéristiques typiques de l'architecture N-tiers peuvent inclure la sécurité, la disponibilité, l'extensibilité, la facilité de gestion, la facilité d'utilisation, la facilité d'accès et la facilité de mise en œuvre.

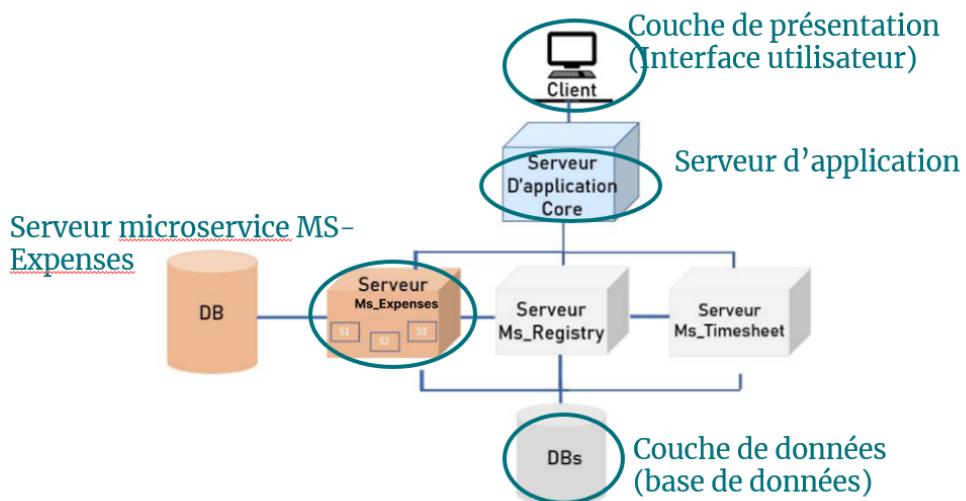


FIGURE 2.2 : Architecture physique

L'architecture de notre système est répartie en quatre couches :

- **Couche de présentation (React JS)** : Cette couche est la première couche de notre application web. Elle assure les fonctionnalités suivantes : Représenter l'interface utilisateur, recevoir les demandes du client, transférer les demandes au deuxième tiers, recevoir les réponses du deuxième tiers et les transmettre au client.
- **Serveur d'application** : Il est responsable de la gestion de la logique métier générale de l'application. C'est dans cette couche que la logique applicative principale est implémentée et gérée.
- **Serveur micro-service MS-Expenses** : Il est responsable de la gestion spécifique de la logique métier relative au module de gestion des notes de frais. Cette couche implémente la logique applicative dédiée à ce micro-service.

- Couche de données (base de données) : Ce tiers représente le stockage des données du système. Il est constitué d'une base de données MongoDB.

2.7.2 Architecture logique

2.7.2.1 Architecture micro-services

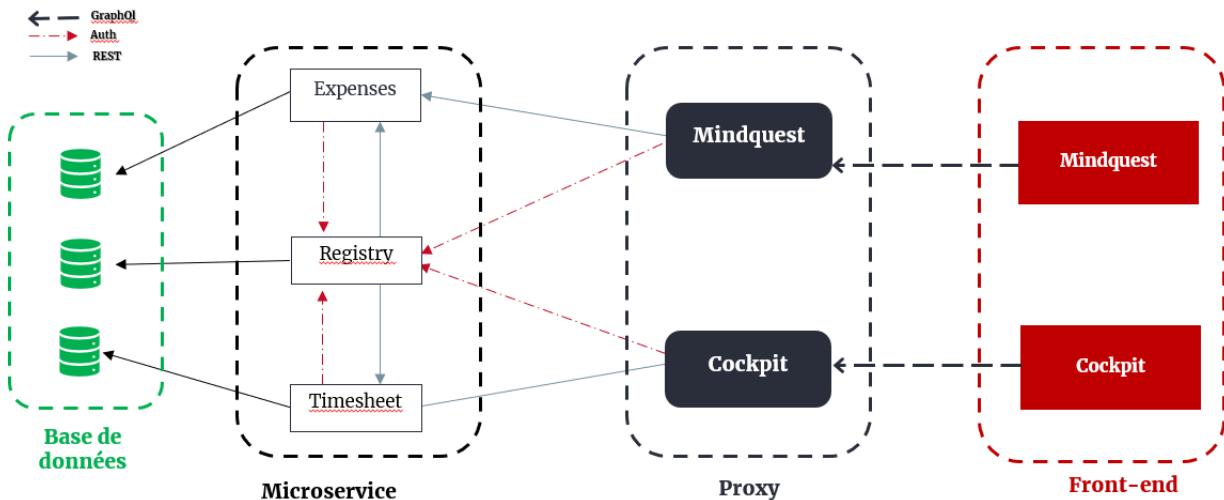


FIGURE 2.3 : Architecture logique

L'architecture de notre application est basée sur une approche micro-services, ce qui permet de découpler les différentes fonctionnalités en services indépendants qui interagissent entre eux. Cette section décrit comment les différents micro-services interagissent entre eux et avec la base de données MongoDB.

Les front-ends (Mindquest et Cockpit) communiquent avec les proxies via GraphQL, tandis que les backends (Mindquest et Cockpit) interagissent avec les autres micro-services (MS-Expenses, MS-Registry, MS-Timesheet) via des API REST.

2.7.2.2 Interaction des micro-services

- **MS-Registry** : Ce service est responsable de l'authentification des micro-services. Un micro-service ne peut communiquer avec les autres que s'il est correctement enregistré auprès du MS-Registry.
- **Backend Mindquest et backend Cockpit** : Ces backends jouent le rôle de proxies. Le front-end ne peut pas communiquer directement avec les autres micro-services sans passer par ces backends.

- **MS-Expenses** : Ce service gère les notes de frais. Il communique avec MS-Timesheet pour obtenir des informations sur les freelancers.
- **MS-Timesheet** : Ce service fournit des informations sur les horaires et les tâches des freelancers.

2.7.2.3 Communication et API : Utilisation de REST et GraphQL

- **GraphQL** : Utilisé pour la communication entre les front-ends (Mindquest et Cockpit) et les backends.
- **REST** : Utilisé pour la communication entre les backends (Mindquest et Cockpit) et les autres micro-services (MS-Expenses, MS-Registry, MS-Timesheet).

Dans le cadre de notre projet, nous avons utilisé deux technologies essentielles pour développer l'API : REST et GraphQL. Ces deux approches sont cruciales pour définir le fonctionnement de l'API et la manière dont les applications interagissent avec les données fournies. Nous avons opté pour GraphQL pour le développement de notre micro-service, tandis que REST a été utilisé pour son intégration dans la plateforme Mindquest.

- **REST** : REST (Representational State Transfer) est un style d'architecture logicielle qui définit un ensemble de contraintes pour créer des services web. Les services web conformes à REST, également appelés services web RESTful, établissent une interopérabilité entre les ordinateurs sur Internet. Les services web REST permettent aux systèmes effectuant des requêtes de manipuler des ressources web via leurs représentations textuelles à travers un ensemble d'opérations uniformes et prédéfinies sans état [15].

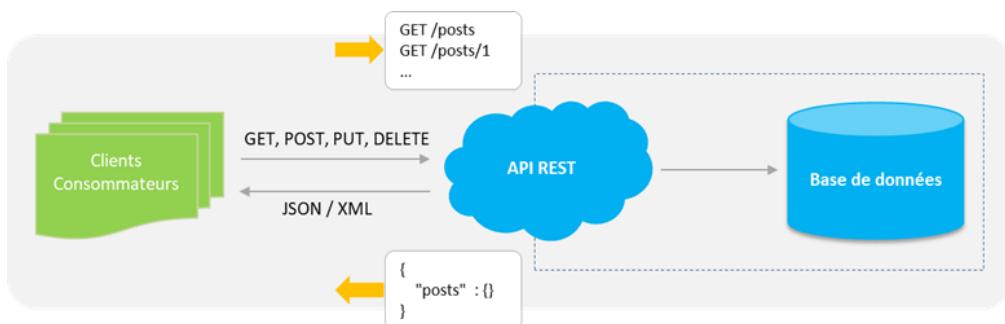


FIGURE 2.4 : L'architecture REST [16]

- **GraphQL** : GraphQL est un langage de requête et un environnement d'exécution côté serveur pour les interfaces de programmation d'application (API) qui permet aux clients de recevoir uniquement les données qu'ils demandent [17].

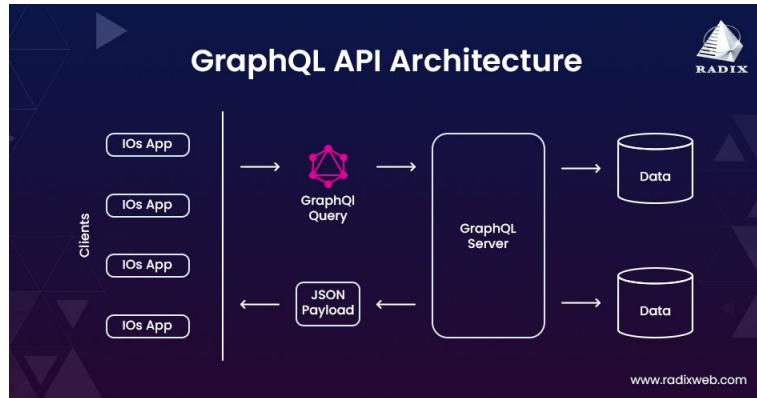


FIGURE 2.5 : L'architecture GraphQL [18]

Pour créer une API GraphQL, nous avons utilisé **Apollo Server**. Apollo Server est un serveur GraphQL open-source, idéal pour créer une API GraphQL prête à la production, auto-documentée et capable d'utiliser des données provenant de n'importe quelle source, y compris le client Apollo. Cette architecture offre plusieurs avantages, notamment l'autonomie des services et leur découplage, ce qui permet une meilleure scalabilité et maintenabilité de l'application. Chaque micro-service peut être développé, déployé et mis à jour indépendamment des autres.

Ensuite, nous détaillerons plus en profondeur les architectures backend et frontend.

2.7.2.4 Architecture backend

Ici, nous allons détailler l'architecture de notre micro-service MS-Expenses. GraphQL API : Le serveur GraphQL Apollo assure l'analyse des requêtes provenant des applications clientes GraphQL. Le schéma suivant résume l'implémentation du serveur Apollo.

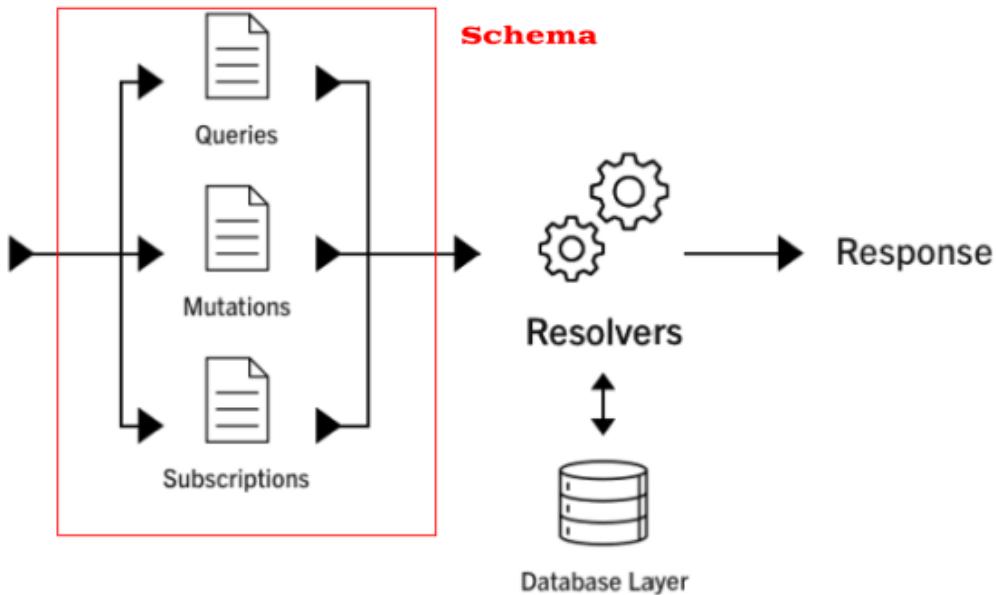


FIGURE 2.6 : Spécification Serveur GraphQL [19]

Le serveur GraphQL procure à un schéma prédéfini, c'est un modèle des données qui pourraient être demandées au serveur. En effet, le schéma sert de terrain d'entente entre le client et le serveur en précisant la manière d'accéder aux données.

Le schéma définit les requêtes autorisées, les types de données récupérables et les relations entre ces types.

Les trois requêtes autorisées principales de GraphQL :

- Query : des requêtes qui ont pour objet de récupérer de la donnée. Aucune modification n'est réalisée via cette dernière[20].
- Mutation : des requêtes visant à modifier l'état des données. En d'autres termes, la mutation vise à modifier les données[20].
- Abonnement : abonnement pour l'obtention automatique de données en temps réel.

Chaque fois qu'une opération GraphQL atteint l'application back-end, elle est analysée par rapport à l'ensemble du schéma et traitée avec les données de l'application front-end.

Le résolveur comportera le code pour retirer les données à partir d'une base de données, ce qui assure l'implémentation de la logique métier.

2.7.2.5 Architecture front-End

Cette section se focalise sur deux composants essentiels de l'architecture front-End : Apollo client et React Redux. Nous examinerons en détail leurs fonctionnalités et leurs contributions à la conception

globale de l'application.

- Apollo client : Apollo client constitue un élément central de notre architecture front-End. Il offre une approche simplifiée pour interagir avec une API tout en offrant des fonctionnalités avancées de mise en cache et de gestion des requêtes. Dans la suite de ce rapport, nous présenterons en profondeur l'architecture sous-jacente du client Apollo et ses principaux avantages dans le développement d'applications modernes.

La figure ci-après illustre l'architecture du client Apollo, mettant en évidence son fonctionnement fondamental. Le client Apollo simplifie le processus de mise en cache des requêtes et leur gestion, offrant ainsi une expérience utilisateur fluide et réactive.

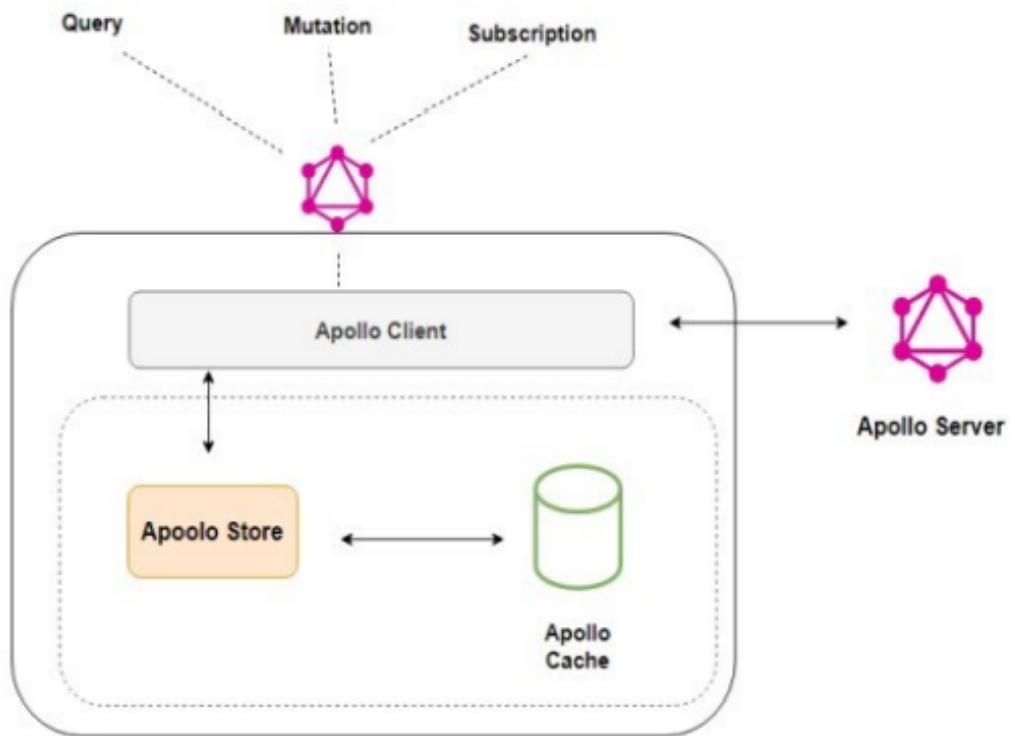


FIGURE 2.7 : Architecture Apollo client [21]

L'utilisation fondamentale du client est très simple et similaire à ce que l'on peut trouver dans n'importe quel client API. Le client Apollo offre la possibilité de mettre en cache les requêtes et de gérer leurs cycles. Une fois qu'une requête est stockée en mémoire cache, elle peut être exécutée autant de fois que désiré, aucun appel réseau ne sera passé.

De plus, le client Apollo prend en charge les mutations, permettant ainsi des modifications de données fluides. Son cache, qui agit comme un gestionnaire d'état similaire à Redux, offre la possibilité

d'accéder aux données obtenues par une requête, ainsi que d'insérer ou de modifier ces données sans relancer la requête.

- React Redux :

En parallèle avec Apollo client, React Redux joue un rôle crucial dans la gestion de l'état de notre application front-End. Il fournit un mécanisme robuste pour gérer les interactions utilisateur et mettre à jour l'interface utilisateur en conséquence.

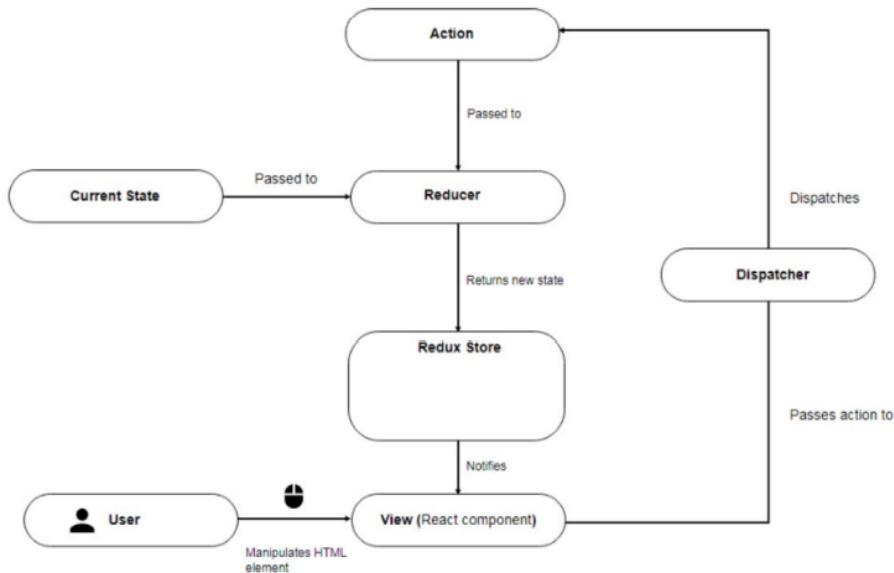


FIGURE 2.8 : Redux [22]

Dès que l'utilisateur interagit avec un élément de la vue, celui-ci déclenche une action qui est interceptée par un dispatcher global. Le dispatcher, en tant que point central, orchestre le processus de mise à jour de l'état de l'application.

En résumé, lorsque l'utilisateur interagit avec la vue, une action est envoyée, accompagnée de l'état actuel, vers le réducteur pour générer un nouvel état dans le store. Ce nouvel état est ensuite propagé à tous les composants de la vue, assurant ainsi une mise à jour cohérente et réactive de l'interface utilisateur.

Conclusion

Dans ce chapitre, nous avons identifié les acteurs de notre système. Nous avons également identifié les besoins fonctionnels de notre client et souligné l'importance de comprendre ses exigences pour proposer une solution adaptée. Aussi, nous avons établi le backlog du produit et précisé

l'environnement de travail. Enfin, nous avons choisi l'architecture de notre projet. Dans le prochain chapitre, nous aborderons la réalisation concrète de notre solution à travers les Sprints. Il comportera une analyse détaillée du premier sprint.

SPRINT 1 : GESTION DES DÉPENSES SUR LA PLATEFORME MINDQUEST

Plan

Introduction	33
1 Sprint Planning	33
2 Backlog du Sprint 1	33
3 Analyse et conception	34
4 Réalisation	38
Conclusion	43

Introduction

Après avoir analysé et spécifié nos besoins globaux, nous détaillerons dans ce chapitre les différentes étapes effectuées durant le premier sprint qui a pour objectif la gestion des dépenses. Nous présenterons le sprint planning, le backlog du premier sprint, l'analyse des besoins, la conception proposée et par la suite la réalisation.

3.1 Sprint Planning

L'objectif de ce sprint de 4 semaines est de développer les fonctionnalités liées aux notes de frais, notamment la création et la modification de notes de frais et de dépenses, la liste et le filtrage des notes de frais, ainsi que l'acceptation ou le refus de celles-ci.

3.2 Backlog du Sprint 1

Le tableau suivant présente le Backlog du premier sprint que nous devons implémenter tout au long de cette itération. L'estimation des user stories est mesurée en Homme jours (HJ).

ID	User Story	Tâches	Estimation (HJ)
1.1	En tant qu'utilisateur, je peux accéder à l'espace des notes de frais.	Préparation de la page de note de frais dans le frontend Mindquest.	1
		Ajouter la page pour le freelancer.	1
		Ajouter la page pour le client.	0.5
1.2	En tant que freelancer, je peux créer une note de frais liée à une mission en cours.	Préparation de micro-service ms-expenses.	1
		Préparation de l'api de recherche des missions actives.	1
		Préparation de l'api d'ajout de note de frais.	1
		Préparation de la partie front.	1
		Consommation des APIs.	0.5

ID	User Story	Tâches	Estimation (HJ)
1.3	En tant que freelancer, je peux ajouter une dépense dans une note de frais et la modifier.	Préparation de la partie back.	1
		Préparation de la partie front.	1
		Consommation des APIs.	0.5
1.4	En tant qu'utilisateur, j'ai la possibilité de filtrer les notes de frais en fonction de leurs statuts.	Préparation de la partie back.	1
		Préparation de la partie front.	1
		Consommation des APIs.	1
1.5	En tant que freelancer, je peux envoyer une note de frais au client.	Préparation de la partie back.	1
		Préparation de la partie front.	1
		Consommation des APIs.	0.5
1.6	En tant que client, je peux accepter/refuser une note de frais.	Préparation de la partie back.	1
		Préparation de la partie front.	1
		Consommation des APIs.	0.5
1.7	En tant que freelancer, je peux modifier une note de frais rejetée.	Préparation de la partie back.	1
		Préparation de la partie front.	1
		Consommation des APIs.	0.5

TABLEAU 3.1 : Backlog du Sprint 1

3.3 Analyse et conception

3.3.1 Diagramme de cas d'utilisation raffiné du Sprint 1

Nous détaillons dans la figure 3.1 les différents cas d'utilisation du premier sprint.

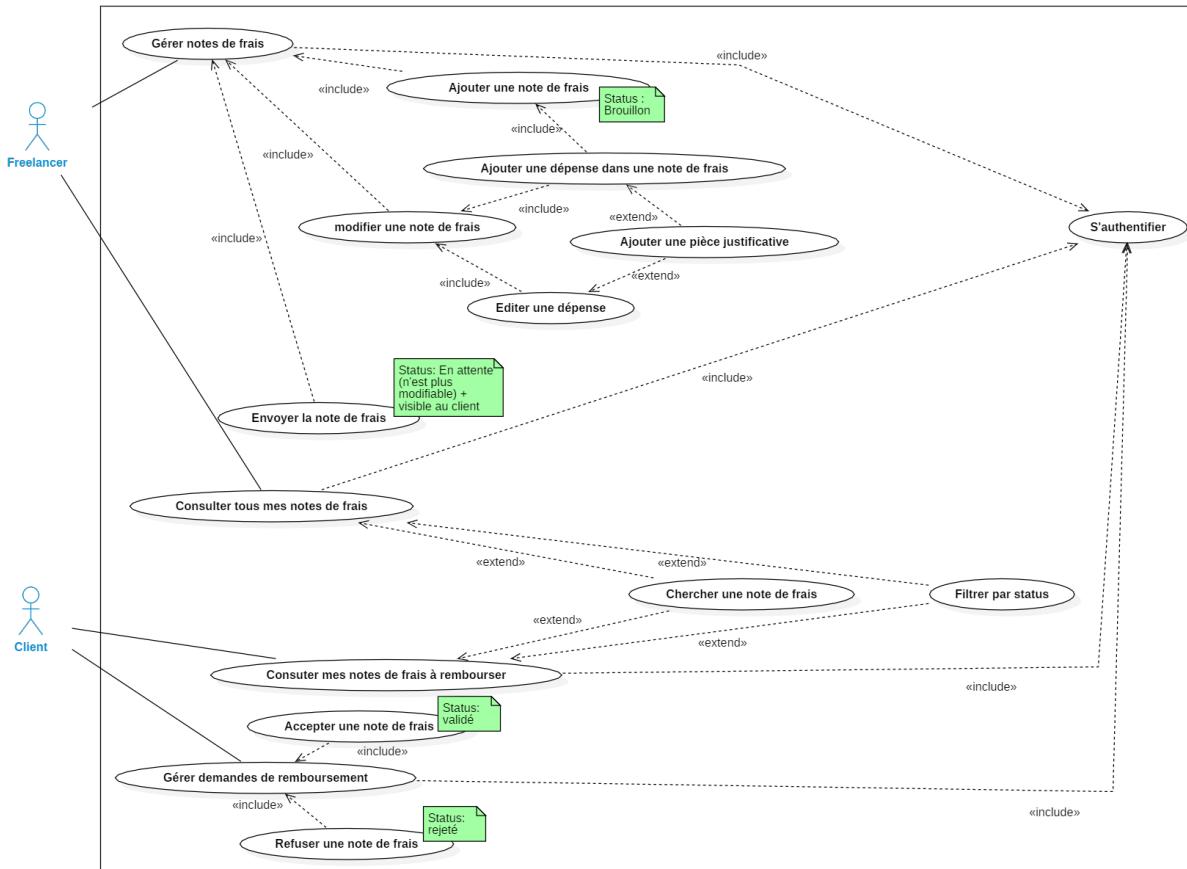


FIGURE 3.1 : Diagramme cas d'utilisation détaillé de "Gestion des dépenses"

À l'issue de ce diagramme raffiné, et pour mieux comprendre le fonctionnement de ce CU, nous décrirons textuellement les cas d'utilisation qui en résultent. Nous choisissons d'expliquer les cas d'utilisation qui nous semblent les plus pertinents sur le plan technique.

Titre	Ajouter une note de frais
Acteurs	freelancer
Pré-condition	Le freelancer doit s'authentifier.
Post-condition	La création d'une nouvelle note de frais.

Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur consulte son espace des notes de frais. 2. Le système affiche un formulaire à remplir. 3. L'utilisateur remplit tous les champs obligatoires. 4. Le système assure la création de la note de frais. 5. Le système assure la redirection vers cette note de frais. 6. Le freelancer peut ajouter des dépenses dans cette note de frais. 7. Le système assure le calcul automatique du montant total qui correspond à la somme des montants des dépenses et de la TVA totale.
Scénario alternatif	<ol style="list-style-type: none"> 4.1 Le système retourne un message d'erreur s'il y a une exception. 4.2 Le système affiche une erreur si l'email est invalide.

TABLEAU 3.2 : Description textuelle du cas d'utilisation «Ajouter une note de frais»

3.3.2 Diagramme de séquence

Le diagramme de séquence détaillé illustre le processus d'ajout d'une note de frais, fournissant une vue complète des interactions entre les acteurs et le système lors de cette opération spécifique.

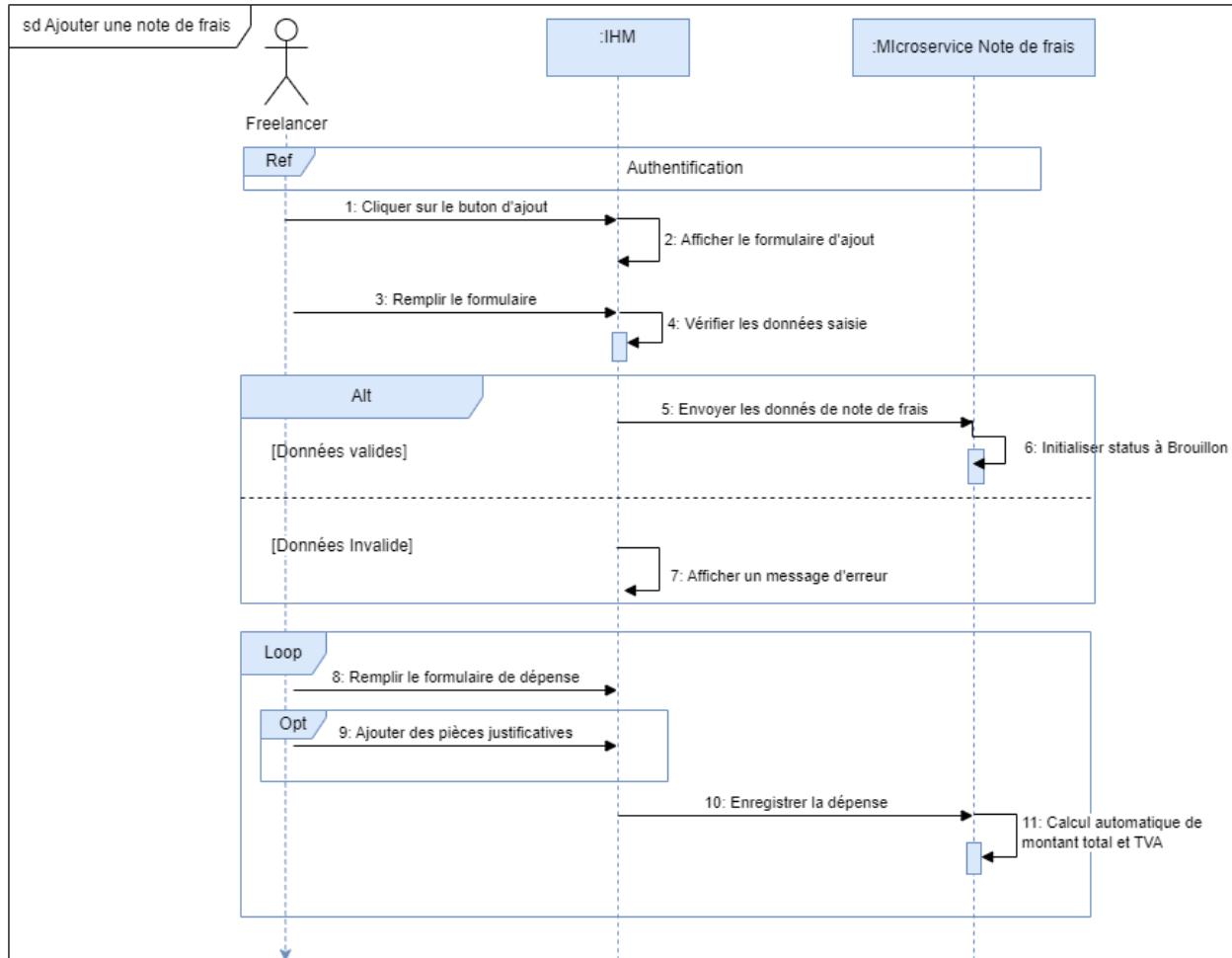


FIGURE 3.2 : Diagramme de séquence détaillé de "Ajouter une note de frais"

3.3.3 Diagramme de classes

La figure 3.3 illustre la structure statique du Sprint 1 schématisée dans un diagramme de classes.

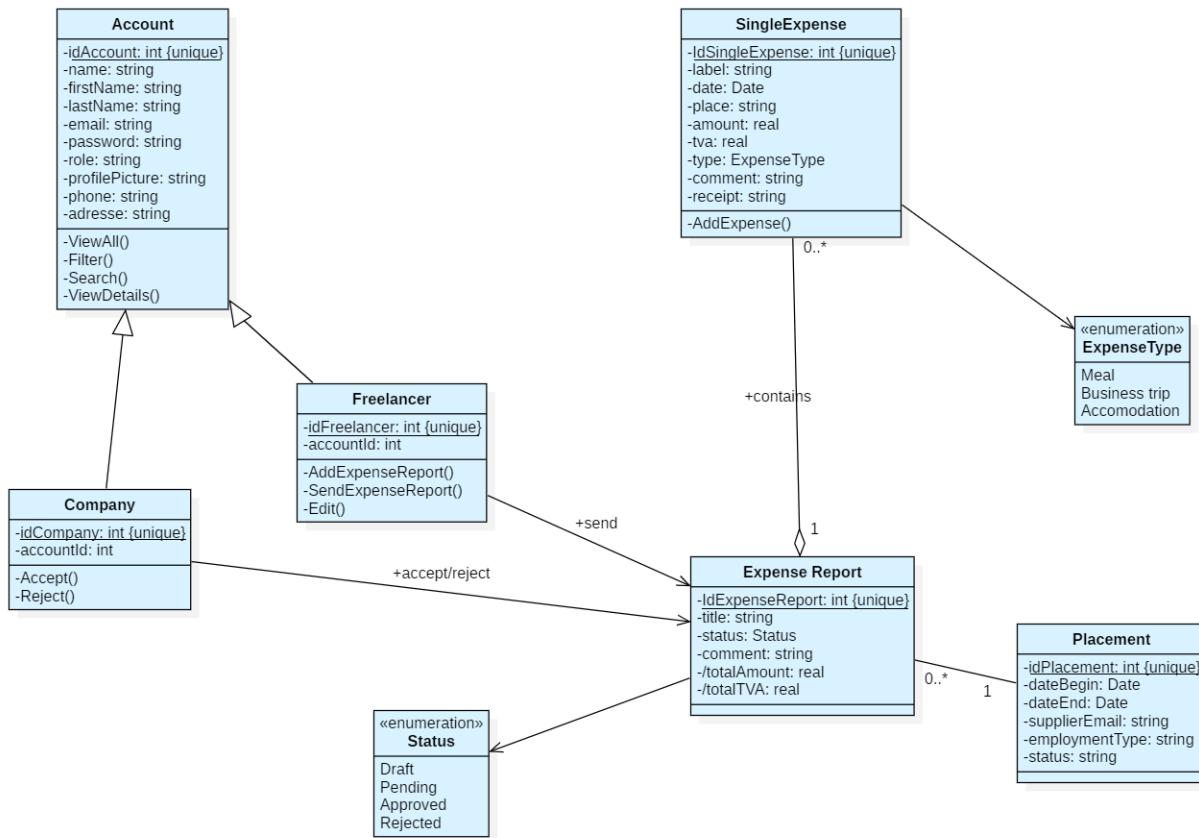


FIGURE 3.3 : Diagramme de classes du sprint 1

3.4 Réalisation

Dans cette section, nous allons présenter la simulation avec des captures d'écran du processus de modélisation de la maquette.

3.4.1 La page initiale des notes de frais

Les figures 3.4 montre l'interface "Espace de note de frais" pour un freelancer. Après l'authentification, l'utilisateur est redirigé vers cette page. Pour tester, nous utilisons des comptes de test pour accéder aux notes de frais de l'utilisateur connecté, car le micro-service d'authentification est déjà développé.

Chapitre 3. Sprint 1 : Gestion des dépenses sur la plateforme Mindquest

The screenshot shows the 'mindquest & CLUB FREELANCE' website. At the top, there's a call-to-action button 'Générez un CV structuré' (Generate a structured CV) with the sub-instruction 'Completez votre profil pour générer un CV téléchargeable et partageable en ligne' (Complete your profile to generate a downloadable and shareable CV). A red button 'Compléter mon profil' (Complete my profile) is also present. The main content area is titled 'Notes de Frais' (Expense Notes). It includes a search bar 'Chercher une note de frais' and a dropdown menu 'Tous'. Below the search bar, a message says 'Vous n'avez pas encore de note de frais' (You haven't yet created an expense note) and 'Ajoutez une note de frais puis créez des dépenses. Le montant total sera calculé automatiquement.' (Add an expense note and then create expenses. The total amount will be calculated automatically.). A red button '+ Note de frais' is located at the bottom right of this section. On the left sidebar, under the 'Notes de frais' heading, there's a link 'fake client test note de frais'.

FIGURE 3.4 : La page des notes de frais - freelancer

La figure 3.5 présente la page initiale d'une note de frais, c'est-à-dire lorsque le client n'a aucune note de frais à rembourser.

The screenshot shows the 'mindquest & CLUB FREELANCE' website. At the top, there's a message 'Vous avez un besoin CDI ou Freelance' (You have a CDI or Freelance need) with the sub-instruction 'Nos équipes vous accompagnent sur tous vos besoins IT, Digital et Finance' (Our teams support you in all your IT, Digital and Finance needs). A red button 'Poster un besoin' (Post a need) is also present. The main content area is titled 'Notes de Frais' (Expense Notes). It includes a search bar 'Chercher une note de frais' and a dropdown menu 'Tous'. Below the search bar, a message says 'Vous n'avez pas encore de note de frais à rembourser.' (You haven't yet created an expense note to be reimbursed). A red button '+ Note de frais' is located at the bottom right of this section. On the left sidebar, under the 'Notes de frais' heading, there's a link 'fake client test note de frais'.

FIGURE 3.5 : La page des notes de frais - client

3.4.2 Les pages de la liste des notes de frais

La figure 3.6 montre la liste des notes de frais pour un freelancer, avec des boutons affichés selon le statut de chaque note. Chaque note présente le titre, la date, le montant total et la TVA totale. Les notes sont organisées par ordre de création, la plus récente en haut.

Chapitre 3. Sprint 1 : Gestion des dépenses sur la plateforme Mindquest

The screenshot shows the 'mindquest & CLUB FREELANCE' interface. At the top, there's a call-to-action 'Générez un CV structuré' (Generate a structured CV) with a 'Compléter mon profil' button. The main area is titled 'Notes de Frais' and displays three expense notes:

Date	ID Note	Freelancer	TVA	Montant	Status	Action
16/05/24	Note de frais 2		0€	0€	Brouillon	Editor Envoyer
08/05/24	Note de frais 1		0€	0€	En attente	Consulter
17/05/24			- -	- -		

FIGURE 3.6 : Liste des notes de frais - freelancer

De même, pour le client, la liste de ses notes de frais à rembourser est présentée par la figure 3.7. Chaque note inclut l'ID du contrat associé et le nom du freelancer ayant envoyé la note.

The screenshot shows the 'mindquest & CLUB FREELANCE' interface from the client's perspective. At the top, it says 'Vous avez un besoin CDI ou Freelance' and 'Poster un besoin'. The main area is titled 'Notes de Frais' and displays one note:

Date	ID Note	Freelancer	TVA	Montant	Status	Action
03/05/24	#101214	Fake for Safa	0€	350€	En attente	Voir le détail

FIGURE 3.7 : Liste des notes de frais - client

Sachant qu'un freelancer ou un client peut filtrer les notes de frais en fonction de leurs statuts comme le montre la figure 3.8.

Chapitre 3. Sprint 1 : Gestion des dépenses sur la plateforme Mindquest

Date	Titre	TVA	Montant	Statut
10/05/24	NF4	0€	0€	En attente
16/05/24	NF3	0€	0€	En attente

FIGURE 3.8 : Filtrer les notes de frais en fonction du statut

Ils peuvent également effectuer une recherche en fonction du titre, du montant, de l'ID ou de la date comme le montre la figure 3.9.

Date	Titre	TVA	Montant	Action
08/05/24	Note de frais 1	0€	0€	Consulter
16/05/24	Note de frais 2	0€	0€	Editor Envoyer

FIGURE 3.9 : Rechercher note de frais

3.4.3 La page d'ajout d'une note de frais

Lorsque l'utilisateur clique sur le bouton "Note de frais", un formulaire apparaît, comprenant les champs suivants à remplir : titre, date et l'ID du contrat. Ce dernier est une liste des missions actives disponibles pour le freelancer, parmi lesquelles il peut choisir.

Chapitre 3. Sprint 1 : Gestion des dépenses sur la plateforme Mindquest

The screenshot shows the Mindquest platform's interface. On the left, there's a sidebar with links for Dashboard, Offres, Timesheets, Notes de frais (which is currently selected), Documents, Guide, and Contactez-nous. The main area is titled 'Notes de Frais' and displays a message: 'Vous n'avez pas encore de note de frais. Ajoutez une note de frais puis créez des dépenses. Le montant total sera calculé automatiquement.' A red button '+ Note de frais' is visible. A modal window titled 'Ajouter une note de frais' is open, asking for 'Note de frais 1', 'Date' (set to 03/05/2024), and 'Code client' (set to 101214). A red 'Ajouter' button is at the bottom right of the modal.

FIGURE 3.10 : La page d'ajout d'une note de frais

3.4.4 La page d'ajout/modification d'une dépense dans une note de frais

La figure 3.11 présente la page d'ajout d'une dépense dans une note de frais. Elle comporte un formulaire à remplir comprenant les champs obligatoires tels que le titre, le montant, etc., ainsi que des champs facultatifs comme le commentaire et les pièces justificatives.

This screenshot shows the 'Ajouter une dépense' (Add expense) form within the 'Notes de frais' (Expense Notes) section. On the left, a note titled 'Note de frais 1' is displayed with a single expense entry: 'Dépense 1' for '150€'. This entry has 'Déplacement' as the category, 'TVA 0€', and 'Reçu' checked. A red 'Editer' button is next to it. Below this is a 'Commentaire' field and an 'Envoyer' button. To the right, a larger form for a new expense is shown. It has fields for 'Dépense 2', 'Date' (03/05/2024), 'Lieu' (Paris), 'Montant' (150€), and 'Commentaire'. A red 'Ajouter un reçu' button is at the bottom. A note at the bottom right says 'Valider dépense'.

FIGURE 3.11 : La page d'ajout d'une dépense dans une note de frais

De plus, il est possible de modifier une dépense. Les champs existants sont préremplis et peuvent être modifiés selon les besoins de l'utilisateur.

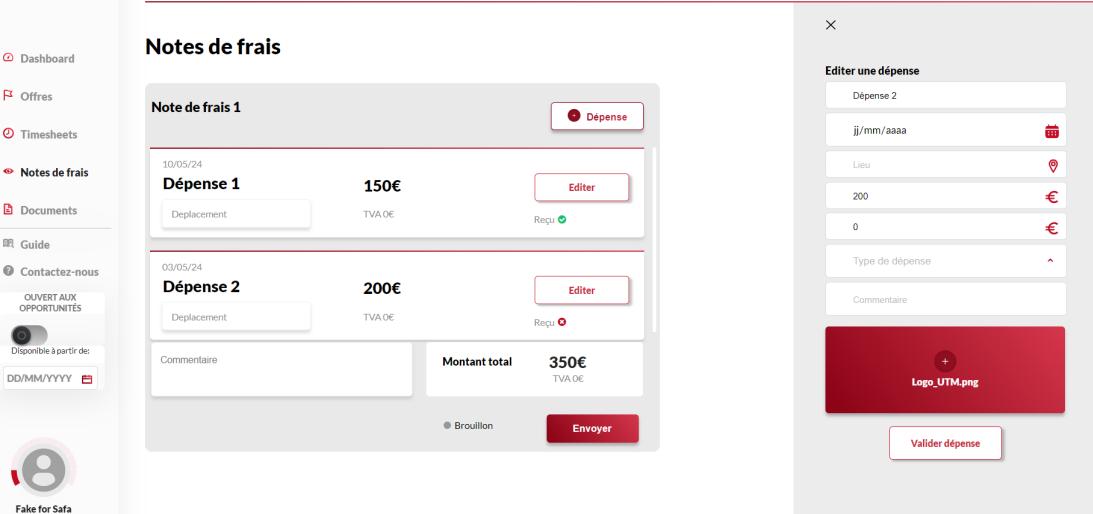


FIGURE 3.12 : La page de modification d'une dépense

3.4.5 Page d'acceptation/refus

Une fois envoyée par le freelancer, la note de frais apparaît dans la liste du client. Le client peut sélectionner une note spécifique et décider de l'accepter ou de la refuser, comme illustré dans la figure 3.13.

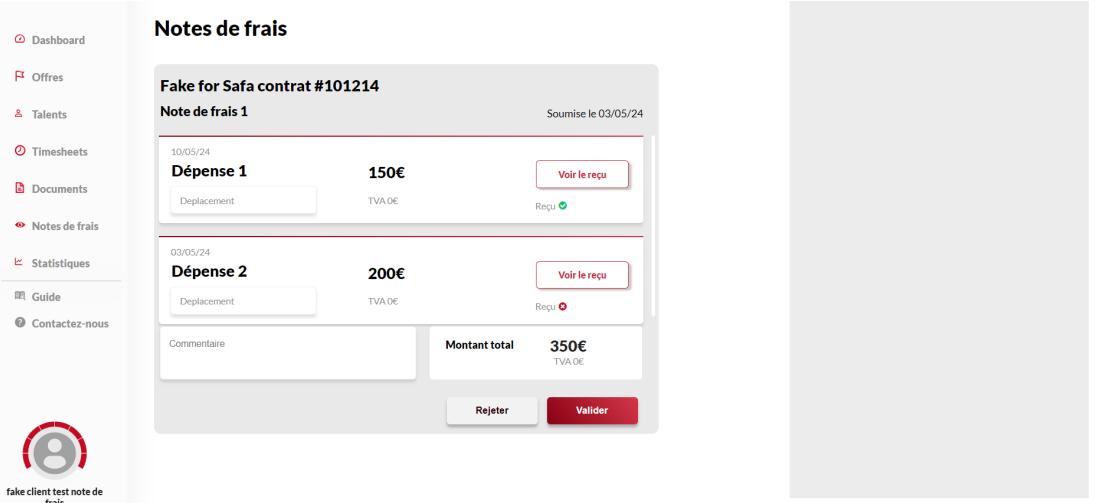


FIGURE 3.13 : Page d'acceptation/refus

Conclusion

Au cours du premier sprint, nous avons établi son Sprint Backlog, suivi de la création d'un diagramme de cas d'utilisation détaillé et de diagrammes de séquence pour certains cas spécifiques. Enfin, nous avons illustré les résultats obtenus à l'aide de captures d'écran de notre solution. Le prochain chapitre résumera le déroulement du travail tout au long du "Sprint 2 : Gestion des notifications".

SPRINT 2 : GESTION DES NOTIFICATIONS

Plan

Introduction	45
1 Sprint Planning	45
2 Backlog du Sprint 2	46
3 Analyse et conception	46
4 Réalisation	48
Conclusion	52

Introduction

Dans ce chapitre dédié au deuxième sprint "Gestion des notifications", nous commencerons par élaborer le planning du sprint, puis le backlog du sprint, qui comprendra les récits à réaliser. Ensuite, nous passerons à l'analyse et à la conception, pour enfin terminer par la réalisation.

4.1 Sprint Planning

L'objectif de ce sprint de 4 semaines est de développer les fonctionnalités liées aux notifications sur la plateforme pour le client, afin de lui indiquer qu'il a de nouvelles notes de frais à rembourser, ainsi que par email pour les deux, freelancer et client, lors de chaque changement de statut des notes de frais, comme le montre la Figure 4.1.

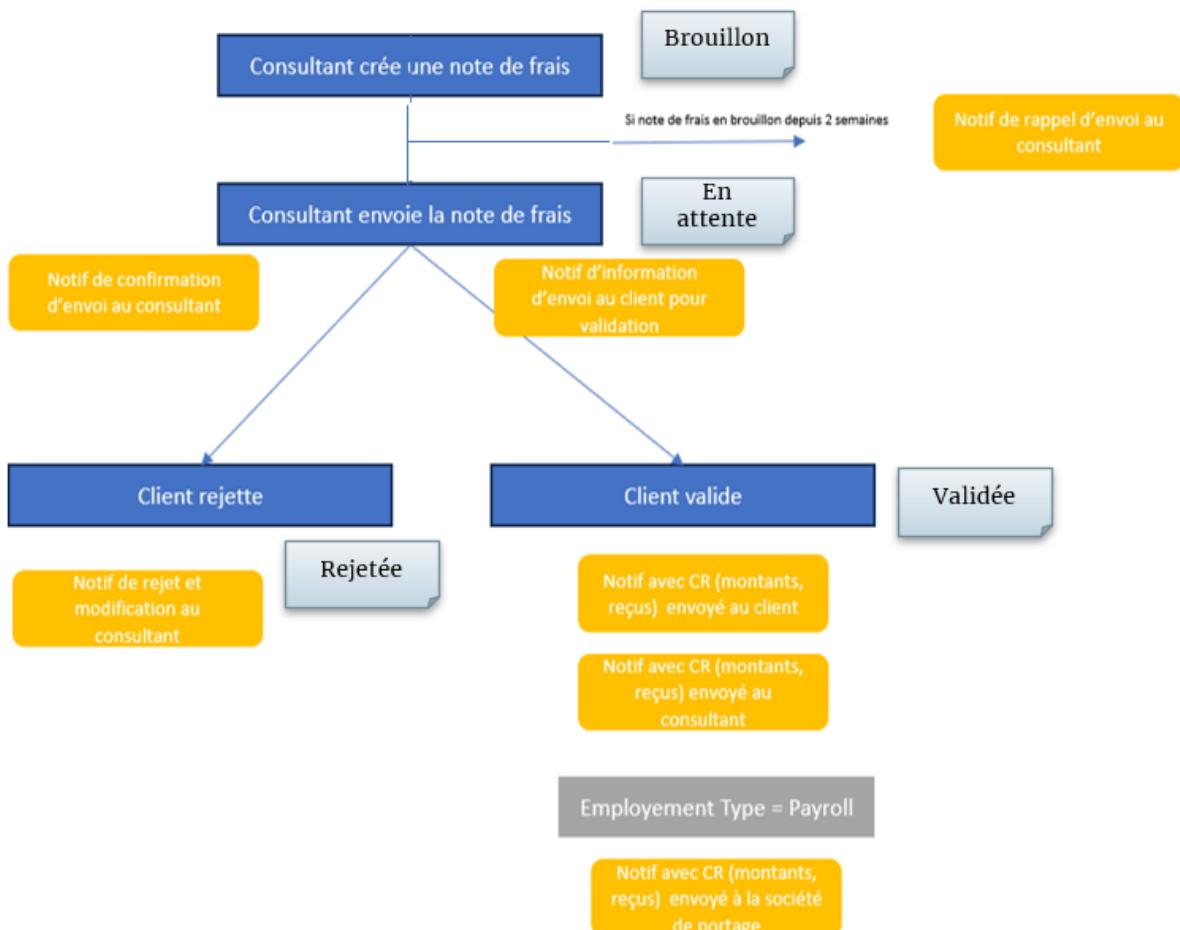


FIGURE 4.1 : Gestion des notifications par email

4.2 Backlog du Sprint 2

Le tableau ci-dessous présente le backlog du deuxième sprint que nous devons implémenter au cours de cette itération. L'estimation des user stories est exprimée en Homme jours (HJ).

ID	User Story	Tâches	Estimation (HJ)
2.1	En tant que client, je souhaite recevoir un email pour être notifié qu'une nouvelle note de frais est en attente de validation.	Configuration serveur d'email Intégration de l'api de serveur d'email Préparation de template Préparation de la partie back Préparation de la partie front Consommation des APIs	0.5 1 1 1 1 1
2.2	En tant que freelancer, je souhaite recevoir un email de rappel d'envoi si la note de frais est en brouillon depuis 2 semaines.	Préparation de template Préparation de la partie back Préparation de la partie front Consommation des APIs	1 1 1 0.5
2.3	En tant que freelancer/client, je souhaite recevoir un compte rendu (montant total de note de frais et les reçus de chaque dépense).	Préparation de template Préparation de la partie back Préparation de la partie front Consommation des APIs	1 1 1 1
2.4	En tant que freelancer, je souhaite recevoir un email de notification si ma note de frais est refusée.	Préparation de template Préparation de la partie back Préparation de la partie front Consommation des APIs	1 1 1 1
2.5	En tant que client, je souhaite recevoir une notification concernant les nouvelles notes de frais dans le Dashboard de Mindquest.	Préparation de la partie back Préparation de la partie front Consommation des APIs	1 1 1

TABLEAU 4.1 : Backlog du Sprint 2

4.3 Analyse et conception

4.3.1 Diagramme de cas d'utilisation raffiné du Sprint 2

Dans la figure 4.2, nous exposons en détail les divers cas d'utilisation du deuxième sprint.

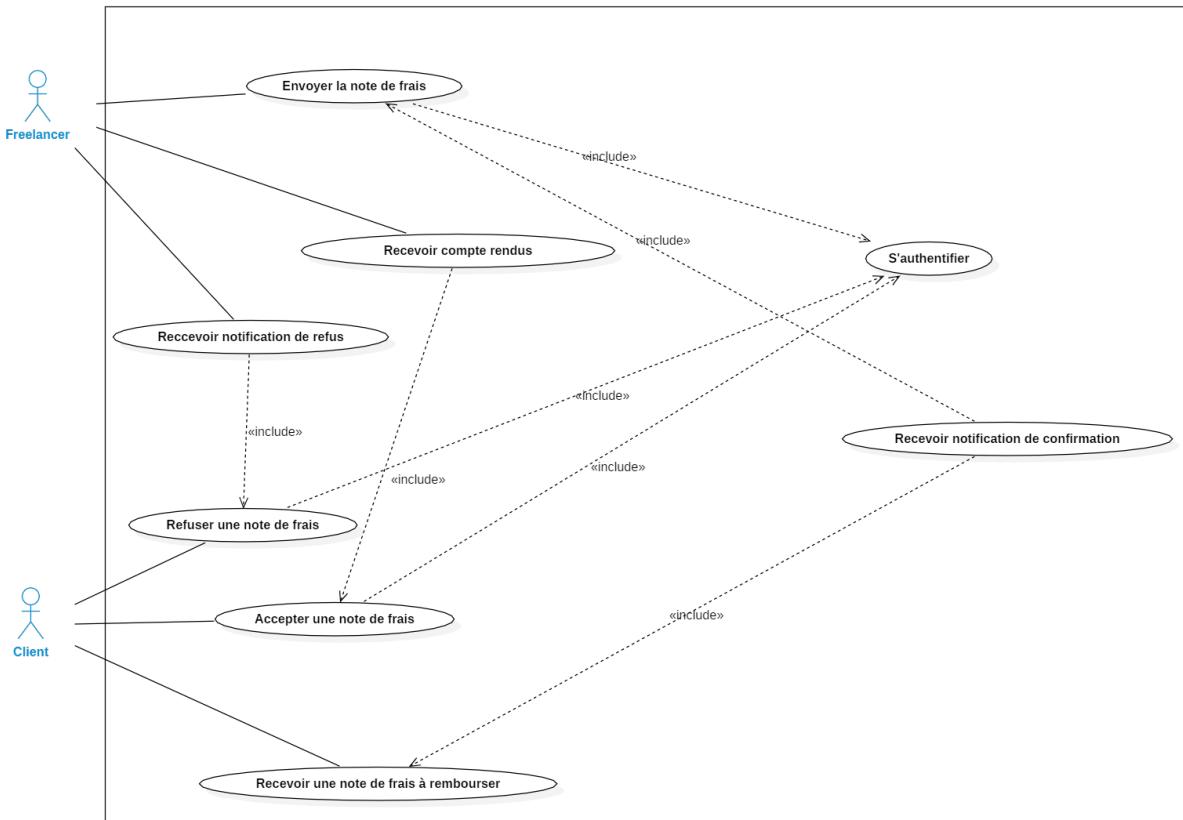


FIGURE 4.2 : Diagramme de cas d'utilisation raffiné du Sprint 2

Nous décrirons les cas d'utilisation les plus pertinents techniquement à partir du diagramme raffiné.

Titre	Recevoir notification de refus
Acteurs	Freelancer
Pré-condition	Le freelancer doit avoir déjà envoyé une note de frais.
Post-condition	Le freelancer reçoit une notification de refus pour une note de frais.
Scénario nominal	<ol style="list-style-type: none"> Le freelancer envoie une note de frais pour validation. Le client reçoit une notification d'une nouvelle note de frais à rembourser. Le client refuse la note de frais à partir du site web de Mindquest. Le système envoie un email de notification de refus au freelancer. Le freelancer reçoit l'email de notification et prend connaissance du refus.
Scénario alternatif	<ol style="list-style-type: none"> Le système affiche une erreur dans les logs si l'email est invalide.

TABLEAU 4.2 : Description textuelle du cas d'utilisation «Recevoir notification de refus»

4.3.2 Diagramme de séquence

Le diagramme de séquence des notifications offre une représentation visuelle des interactions séquentielles entre les différentes entités du système lors de la gestion et de l'envoi des notifications.

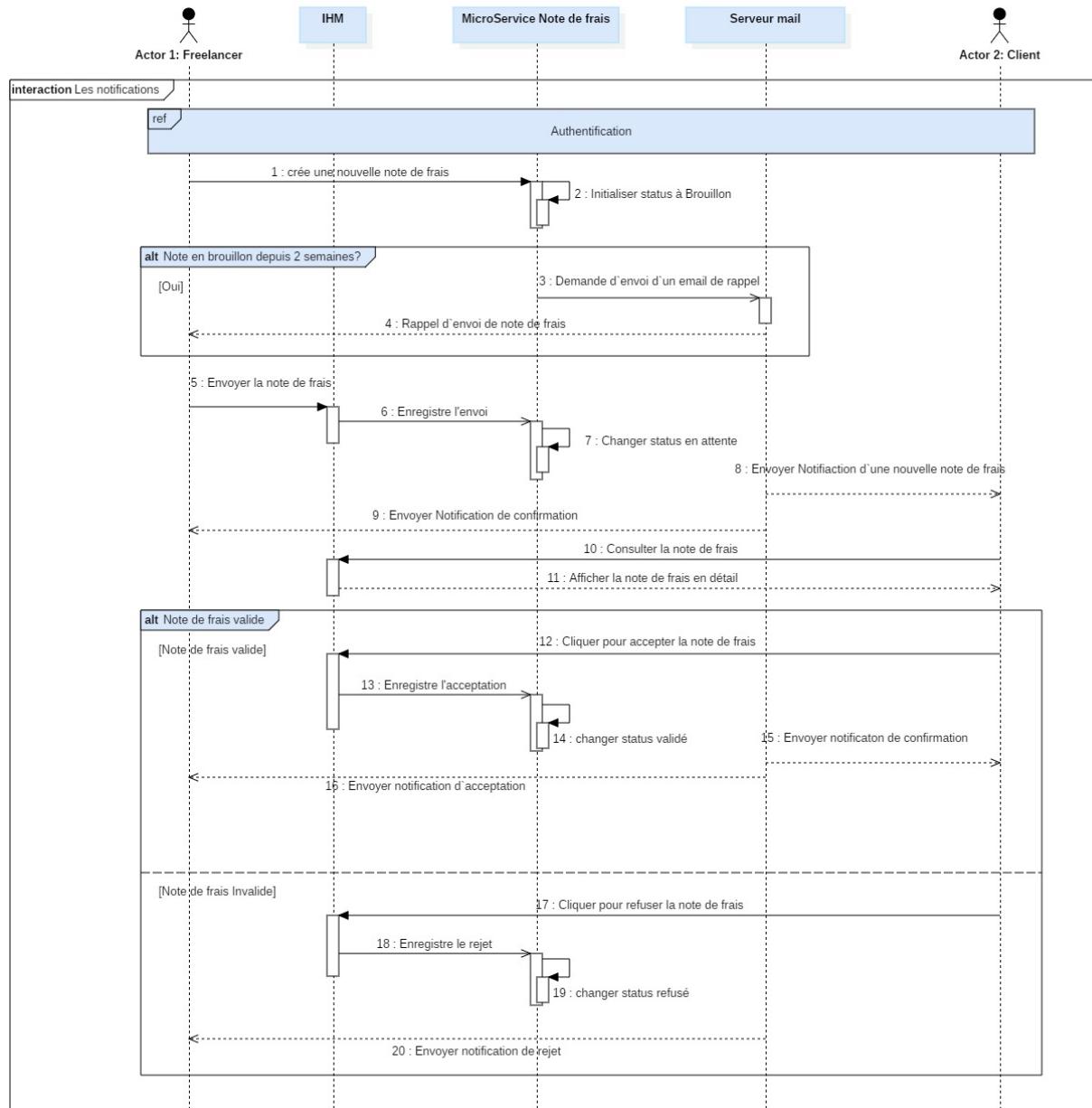


FIGURE 4.3 : Diagramme de séquence des notifications

4.4 Réalisation

Dans cette section, nous allons présenter les réalisations de ce sprint avec des captures d'écran.

4.4.1 Les notifications sur la plateforme

La capture d'écran, présentée par la figure 4.4, montre une notification informant un client d'une nouvelle note de frais en attente, avec le numéro de contrat et le nom du freelancer. Fermer l'icône fait disparaître la notification, tandis que cliquer sur "Consulter" redirige vers la note de frais pour plus de détails.

mindquest &
CLUB FREELANCE

Vous avez un besoin CDI ou Freelance
Nos équipes vous accompagnent sur tous vos besoins IT, Digital et Finance

Notes de Frais

03/05/24 #101214
Fake for Safa TVA 0€
350€
Note de frais 1 ● En attente

Chercher une note de frais Tous

Voir le détail

Nouvelle note de frais
Vous avez une nouvelle note de frais pour Fake for Safa
contrat 101214
Consulter >

Dashboard
Offres
Talents
Timesheets
Documents
Notes de frais
Statistiques
Guide
Contactez-nous

FIGURE 4.4 : Notification pour un client

4.4.2 Les notifications par email

La première notification par email rappelle au consultant d'envoyer une note de frais laissée en brouillon depuis 2 semaines. La capture d'écran de la figure 4.5 montre un exemple d'email qui sera envoyé au freelancer.

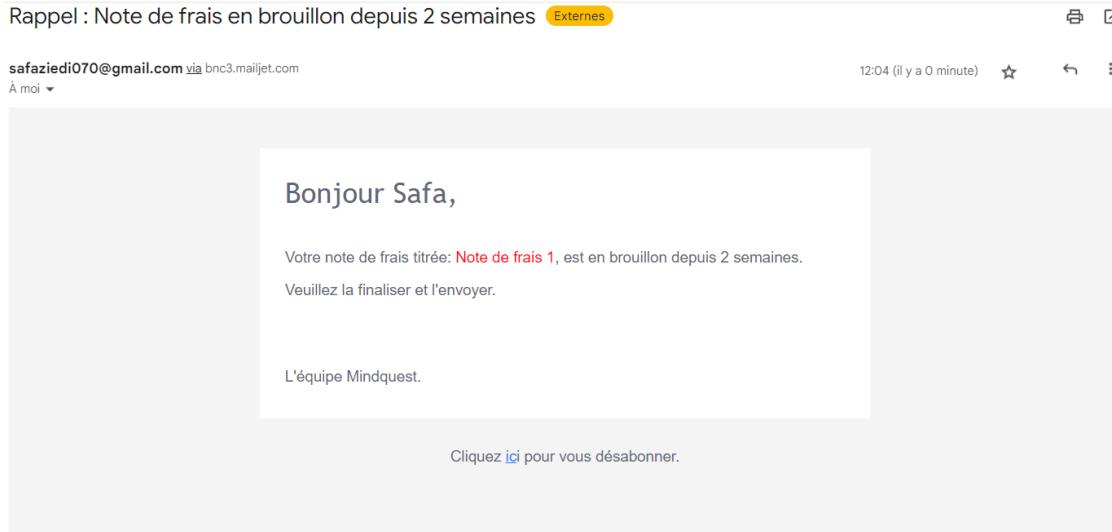


FIGURE 4.5 : Email Rappel : Note de frais en brouillon depuis 2 semaines

Les deux captures d'écran, présentée par la figure 4.6 et 4.7, montrent un email de confirmation d'envoi de note de frais pour le freelancer et un email pour le client, l'informant d'une nouvelle note de frais en attente de validation. Les deux emails contiennent le nom du freelancer et l'ID du contrat associé.

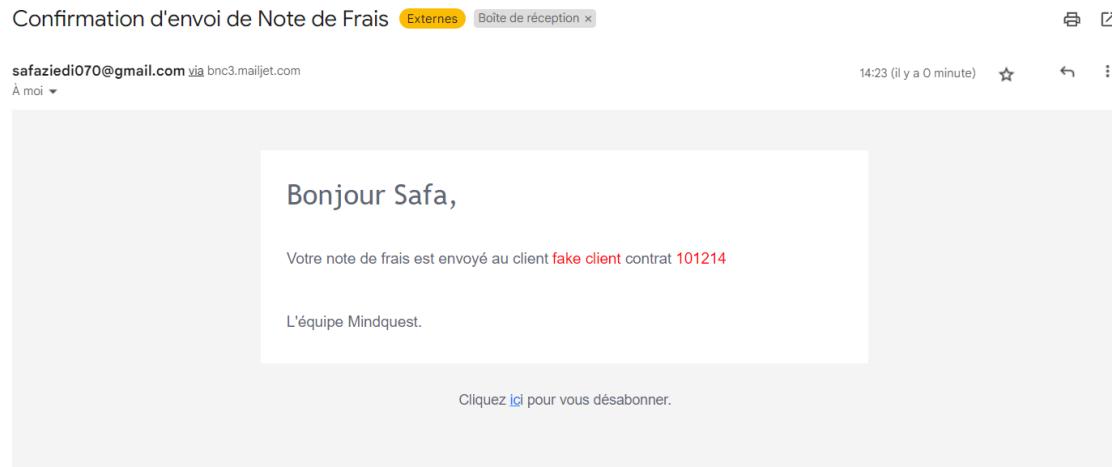


FIGURE 4.6 : Email de confirmation d'envoi de note de frais

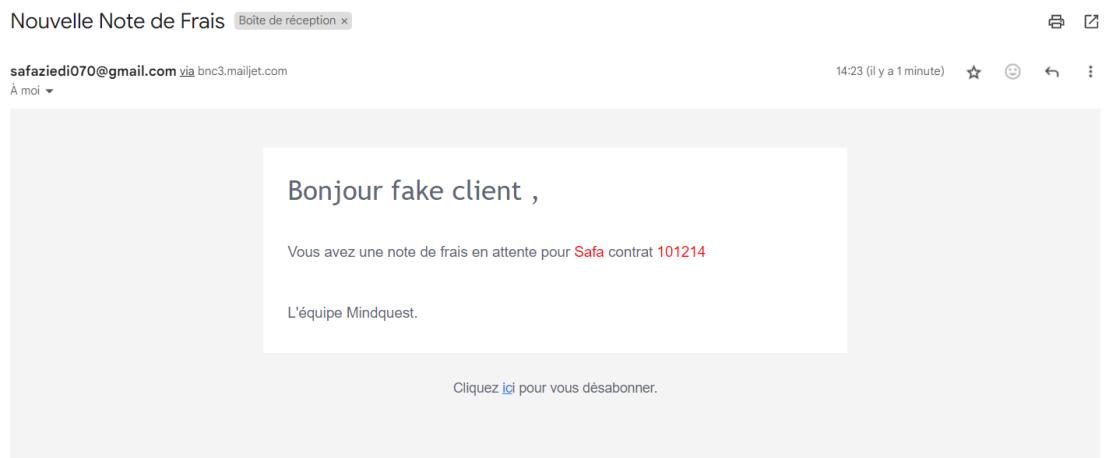


FIGURE 4.7 : Email : Nouvelle note de frais

Les deux avant-dernières captures d'écran, présentées par les figures 4.8 et 4.9, montrent les emails envoyés après l'acceptation d'une note de frais. Ces emails contiennent les reçus en pièces jointes, le montant total, et d'autres détails pertinents. L'un est destiné au consultant et l'autre au client.

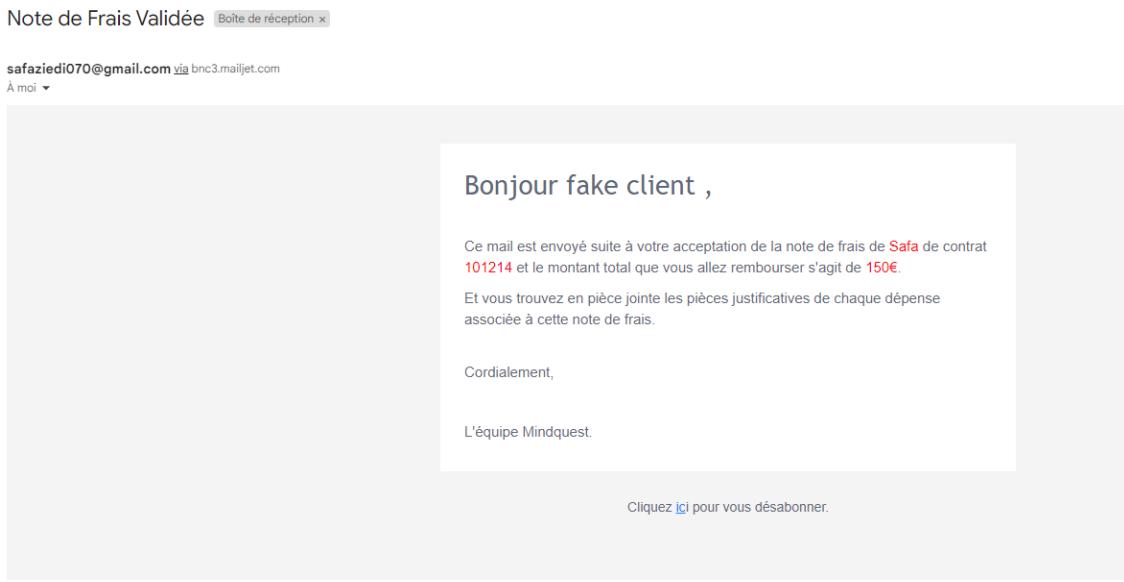


FIGURE 4.8 : Email vers client : note de frais validée

Chapitre 4. Sprint 2 : Gestion des notifications

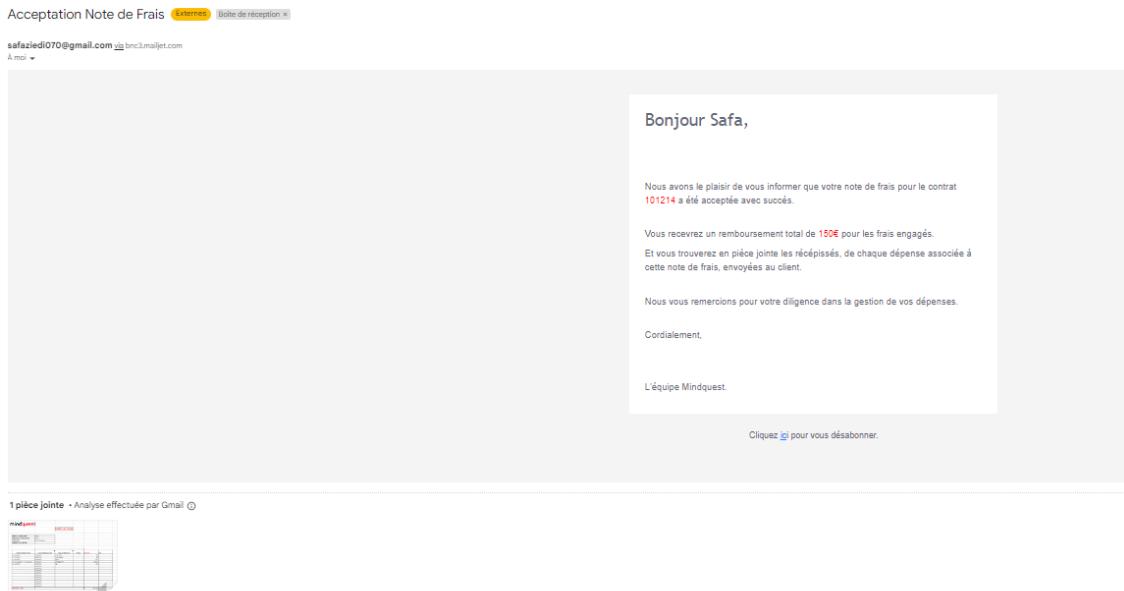


FIGURE 4.9 : Email vers freelancer : Acceptation Note de Frais

La dernière capture d'écran, présentée par la figure 4.10, montre un email de notification de rejet envoyé au freelance. Cet email indique que sa note de frais a été refusée et qu'il doit la modifier pour qu'elle soit acceptée.

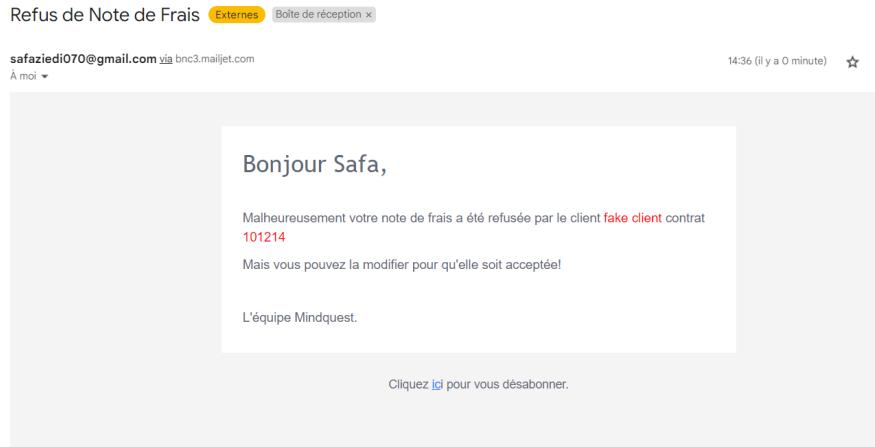


FIGURE 4.10 : Email vers freelancer : Refus de Note de Frais

Conclusion

Lors de ce sprint, nous avons présenté le Sprint Backlog, raffiné le diagramme de cas d'utilisation et le diagramme de séquence système. Nous avons clôturé le chapitre en démontrant le résultat du développement avec des captures d'écran de notre solution. Le prochain chapitre récapitulera le déroulement du "Sprint 3 : Intégration et déploiement du module de notes de frais sur la plateforme

Mindquest".

SPRINT 3 : INTÉGRATION ET DÉPLOIEMENT DU MODULE DE NOTES DE FRAIS SUR LA PLATEFORME MINDQUEST

Plan

Introduction	55
1 Sprint Planning	55
2 Backlog du Sprint 3	55
3 Analyse des besoins	56
4 Conception	58
5 Réalisation	58
Conclusion	61

Introduction

Dans ce chapitre consacré au troisième sprint, qui s'étend sur quatre semaines et est intitulé "Intégration et déploiement sur la plateforme Mindquest", nous débuterons par l'élaboration du planning du sprint, suivie de la création du backlog du sprint contenant les récits à réaliser. Ensuite, nous procéderons à l'analyse et à la conception avant de terminer par la phase de réalisation.

5.1 Sprint Planning

Pour ce troisième sprint, l'objectif est d'intégrer le module de notes de frais sur la plateforme Mindquest, puis de le déployer en collaboration avec l'ingénieur DevOps. Cela permettra aux ingénieurs de test d'évaluer les différentes fonctionnalités et d'ajouter ou de corriger les éléments nécessaires.

5.2 Backlog du Sprint 3

Dans le tableau de planification du sprint, chaque récit utilisateur sera décomposé en tâches simples et claires, accompagnées d'une estimation du temps nécessaire en homme jours pour leurs réalisations.

ID	User Story	Tâches	Estimation (HJ)
3.1	En tant que freelancer/client, je peux accéder à l'espace des notes de frais dans Mindquest.	Installation du projet	0.5
		Découverte de l'architecture du projet	2
		Intégrer le micro-service dans l'ensemble des micro-services.	1
		Assurer la communication entre le ms-expenses et les autres micro-services.	1
		Implémenter la navigation vers l'espace des notes de frais.	1
		Tester l'accès et la fonctionnalité de l'espace des notes de frais.	1

ID	User Story	Tâches	Estimation (HJ)
3.2	En tant que freelancer/client, je peux changer la langue de la page des notes de frais.	Intégrer les langues disponibles pour la page des notes de frais	2
		Tester le changement de langue sur la page des notes de frais	1
3.3	En tant que freelancer/client, je souhaite recevoir les emails dans la langue que j'ai sélectionnée sur le site.	Préparation des templates pour la langue anglaise.	1
		Configurer l'API d'envoi d'email pour qu'elle puisse choisir le modèle en fonction de la langue.	2
		Détecter la langue choisie.	1
		Tester l'envoi d'emails dans différentes langues	1
3.4	En tant que client, je peux voir un réçu d'une dépense.	Concevoir l'interface de réception et téléchargement de reçus.	1
		Préparation de la partie back	1
		Consommation des APIs	0.5
		Tester différents types de pièces justificatives (PDF, Word, PNG, etc.) avec différentes tailles.	3

TABLEAU 5.1 : Backlog du Sprint 3

5.3 Analyse des besoins

5.3.1 Diagramme de cas d'utilisation raffiné du Sprint 3

Dans cette section, nous détaillons les divers cas d'utilisation du troisième sprint.

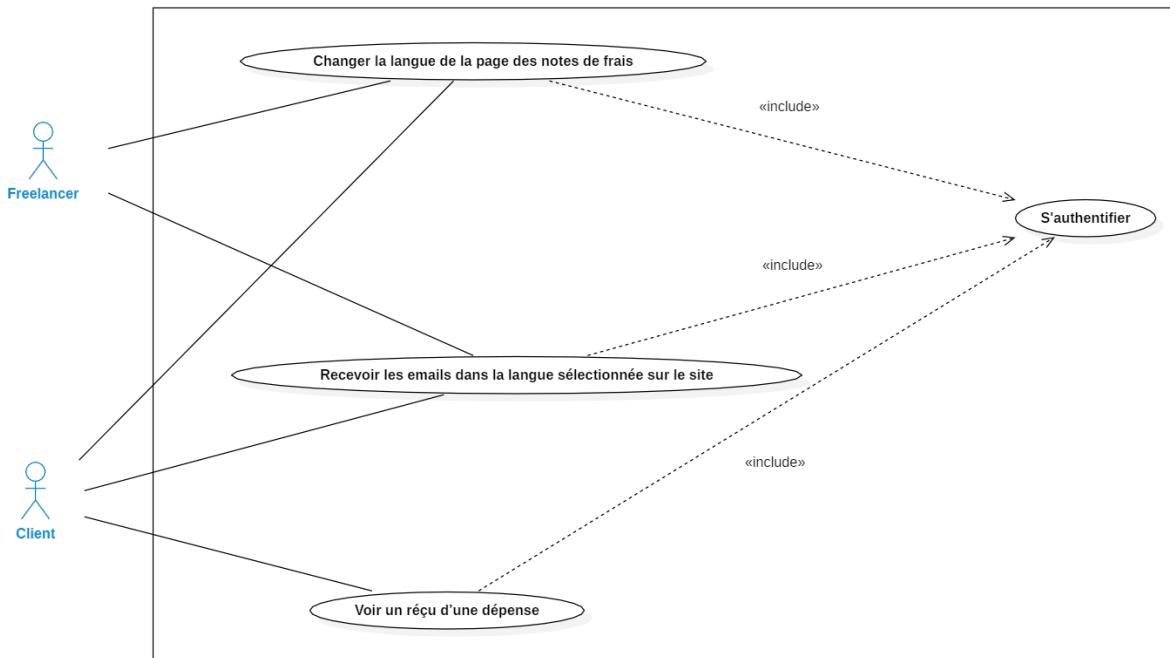


FIGURE 5.1 : Diagramme de cas d'utilisation raffiné du Sprint 3

À la suite de cette représentation détaillée du diagramme, afin de mieux appréhender le fonctionnement de ces cas d'utilisation, nous fournirons des descriptions textuelles des cas les plus pertinents sur le plan technique.

Titre	Voir reçu d'une dépense
Acteurs	Client
Pré-condition	Le freelancer a déjà soumis la note de frais.
Post-condition	Le client peut voir les reçus des différentes dépenses.
Scénario nominal	<ol style="list-style-type: none"> 1. Le client accède à la liste des notes de frais soumises par le freelancer. 2. Le client clique sur une note de frais pour en voir les détails. 3. Le système affiche une vue détaillée de la note de frais, incluant les dépenses qu'elle contient. 4. Le client clique sur le reçu de la dépense qu'il souhaite consulter. 5. Le reçu va être téléchargé localement.
Scénario alternatif	<ol style="list-style-type: none"> 4.1 Le système affiche une erreur s'il ne parvient pas à afficher le reçu.

TABLEAU 5.2 : Description textuelle du cas d'utilisation «Voir reçu d'une dépense»

5.4 Conception

Nous détaillerons dans cette section la phase de conception du sprint 4.

Nous présenterons le diagramme du paquetage pour la gestion des notes de frais comme exemple.

Pour la partie Front-end :

- **Src.components** : Contient tous les composants réutilisables (input, modal, etc.).
- **Src.containers.dashboard-expenses** : Contient des composants qui renvoient des éléments React décrivant ce qui devrait apparaître à l'écran.
- **Query** : Présente une collection de hooks pour la recherche, la mise en cache et la mise à jour de l'état asynchrone dans React. Il s'agit d'une petite API simple qui peut être utilisée immédiatement sans aucune configuration.

Pour la partie Back-end :

- **Src.config** : Contient les configurations nécessaires.
- **Src.schema** : Contient des schémas pour modéliser, définir les types de variables et structurer les données.
- **Src.schema.typeDefs** : Contient les différents schémas.
- **Src.schema.resolvers** : Contient les mutations et les requêtes.

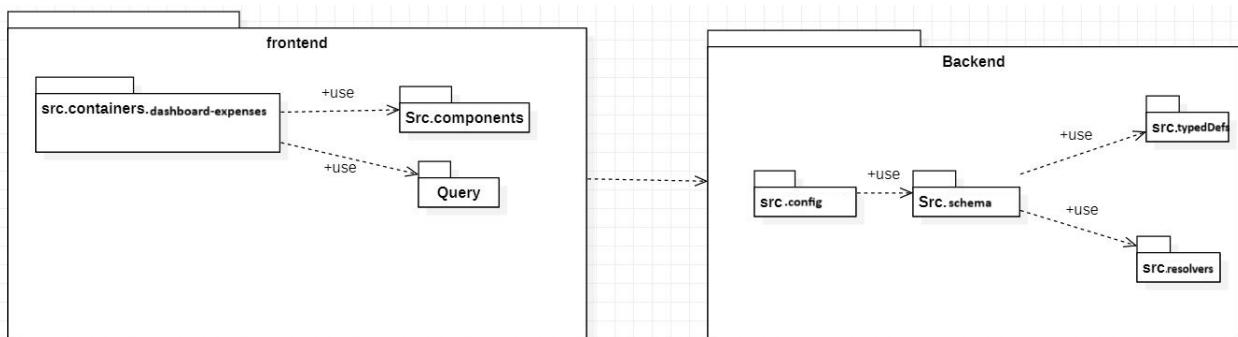


FIGURE 5.2 : Diagramme de paquets pour la gestion des notes de frais

5.5 Réalisation

Cette section est consacrée à la présentation des résultats du sprint, accompagnés de captures d'écran.

Chapitre 5. Sprint 3 : Intégration et déploiement du module de notes de frais sur la plateforme Mindquest

La figure 5.3 illustre la connexion à la plateforme.

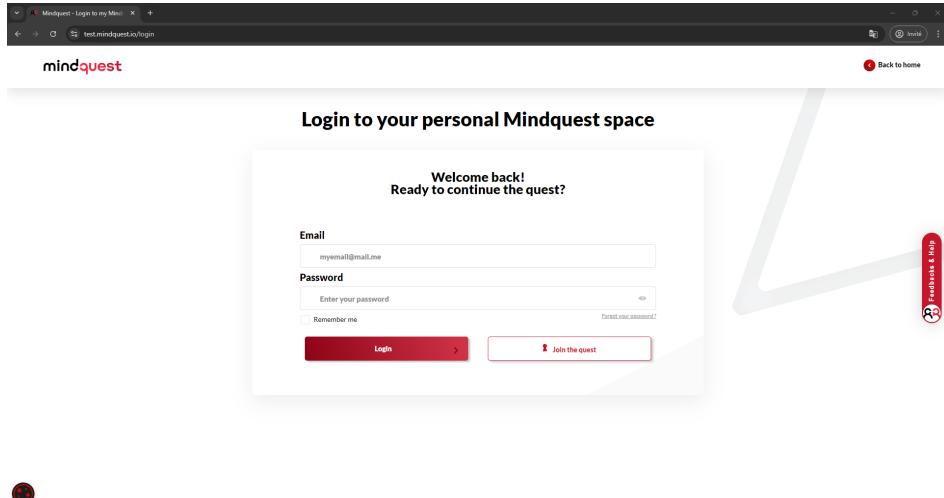


FIGURE 5.3 : Page de connexion sur Mindquest

Après la connexion, que ce soit en tant que client ou freelancer, l'utilisateur peut consulter l'ensemble de ses notes de frais.

La figure 5.4 montre l'espace des notes de frais sur Mindquest.

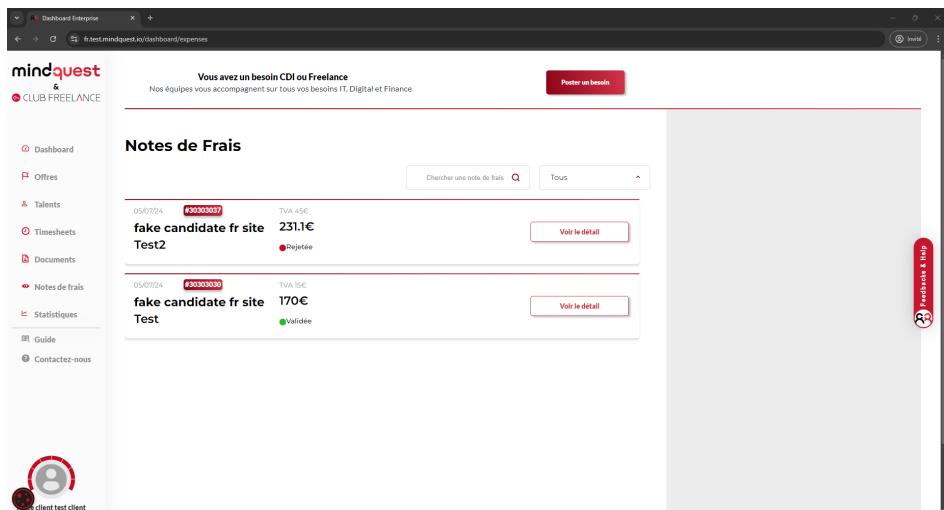


FIGURE 5.4 : Espace des notes de frais sur Mindquest

Le client peut ouvrir une note de frais en détail et voir les reçus disponibles uniquement pour les dépenses qui contiennent des notes de frais.

La figure 5.5 présente une note de frais en détail.

Chapitre 5. Sprint 3 : Intégration et déploiement du module de notes de frais sur la plateforme Mindquest

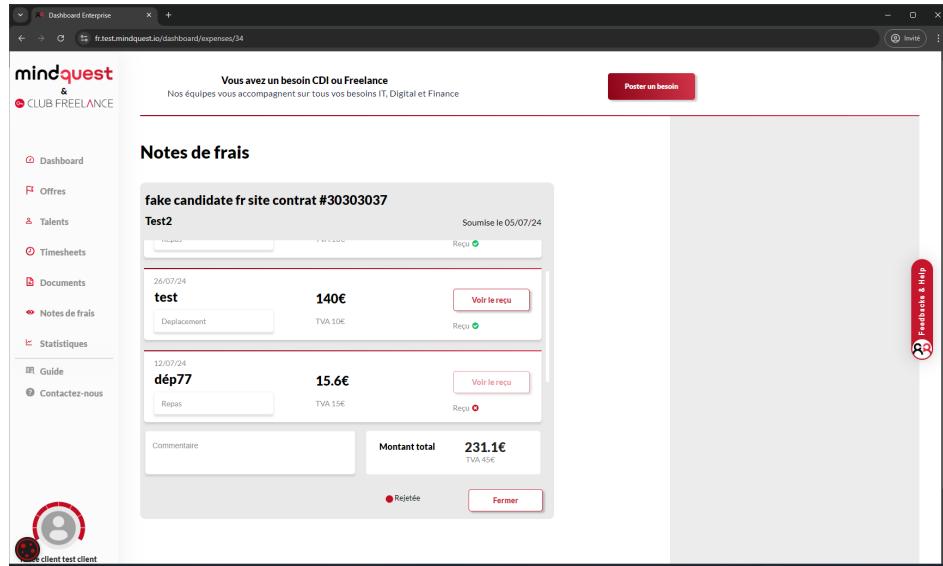


FIGURE 5.5 : Note de frais en détail

Pour voir le reçu, une fenêtre de téléchargement s'ouvre permettant de le récupérer, comme illustré dans la figure 5.6.

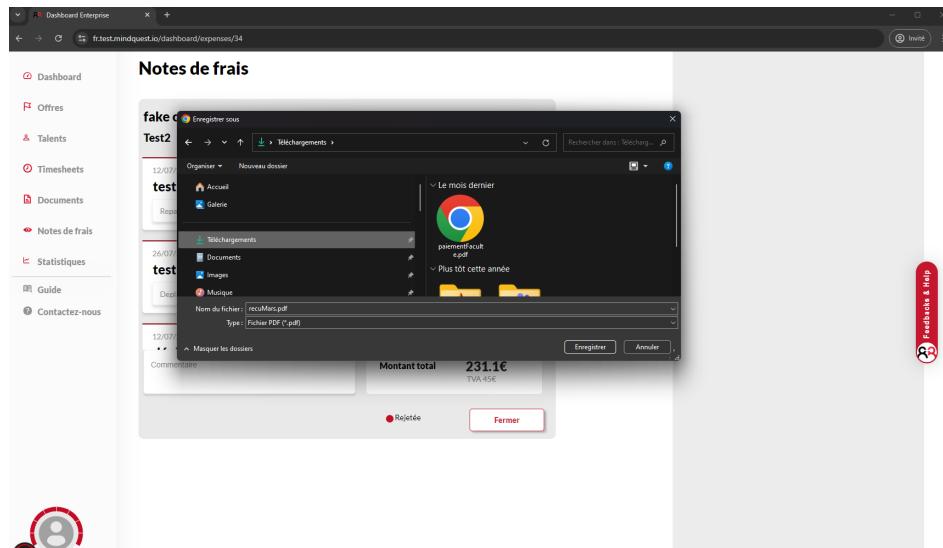


FIGURE 5.6 : Téléchargement du reçu

Par ailleurs, l'utilisateur reçoit des emails dans la langue qu'il a choisie. Par exemple, la figure 5.7 montre un email de confirmation d'envoi de note de frais en anglais.

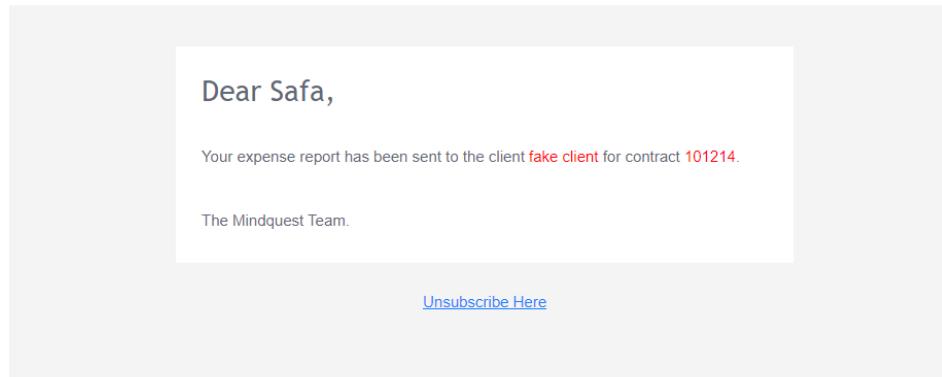


FIGURE 5.7 : Email de confirmation en anglais

Conclusion

Au cours de ce sprint, nous avons partagé le Sprint Backlog, le diagramme de cas d'utilisation raffiné ainsi que le diagramme de déploiement. Nous avons, ensuite, conclu en présentant les résultats du déploiement à travers des captures d'écran illustrant notre solution. Le chapitre suivant fournira un récapitulatif du travail effectué tout au long du "Sprint 4 : Développement de la partie Cockpit".

SPRINT 4 : DÉVELOPPEMENT DE LA PARTIE COCKPIT

Plan

Introduction	63
1 Sprint Planning	63
2 Backlog du Sprint 4	63
3 Analyse détaillée du sprint 4	65
4 Conception	66
5 Réalisation	67
Conclusion	69

Introduction

Après avoir achevé le développement et le déploiement de la partie Mindquest, nous nous concentrerons désormais sur la création de Cockpit. Cette plateforme offre des avantages significatifs à l'équipe marketing, aux ventes, aux managers, aux sourceurs et aux administrateurs de Mindquest en simplifiant la gestion interne.

Ce chapitre détaillera les étapes clés du quatrième sprint, axé sur le développement de Cockpit. Nous aborderons le planning du sprint, le backlog, l'Analyse et la conception, la proposition de conception, ainsi que la mise en œuvre de ces éléments.

6.1 Sprint Planning

L'objectif de ce sprint de 4 semaines est de développer les fonctionnalités liées aux administrateurs concernant les notes de frais. Cela inclut la création et la modification de notes de frais et de dépenses, la liste et le filtrage des notes de frais, ainsi que leurs suppressions, acceptation ou refus.

6.2 Backlog du Sprint 4

Le tableau suivant présente le Backlog du sprint que nous devons implémenter tout au long de cette itération.

ID	User Story	Tâches	Estimation (HJ)
4.1	En tant que sourceur, je peux consulter toutes les notes de frais.	Création de la page de note de frais	1
		Préparation de l'API.	1
		Consommation de l'API.	0.5
4.2	En tant que sourceur, je peux filtrer avec tous les champs.	Préparation de l'API de filtrage coté backend.	2
		Consommation de l'API.	1
4.3	En tant que sourceur, je peux modifier une note de frais.	Préparation de la partie back.	1
		Préparation de la partie front en identifiant les champs qui ne doivent pas être modifiés.	1
		Consommation des APIs.	1
4.4	En tant que sourceur, je peux modifier une dépense.	Intégrer l'API de gestion des modifications de dépenses impactant les notes de frais associées.	1
		Mise en place du formulaire de modification.	2
		Consommation des APIs.	1
4.5	En tant que sourceur, je peux envoyer/accepter/refuser une note de frais.	Intégrer les boutons d'envoi, d'acceptation et de refus en fonction du statut.	1
4.6	En tant que sourceur, je peux supprimer une note de frais avec ses dépenses.	Préparation de l'API.	1
		Inclure le bouton permettant de supprimer une note de frais.	1
		Consommation de l'API.	1
4.7	En tant que sourceur, je peux supprimer une dépense.	Préparer l'API backend pour gérer la suppression, ce qui impacte le montant total et la TVA des notes de frais.	1
		Ajouter le bouton permettant de retirer une dépense.	1
		Consommation de l'API.	1

TABLEAU 6.1 : Backlog du Sprint 4

6.3 Analyse détaillée du sprint 4

6.3.1 Diagramme de cas d'utilisation raffiné du Sprint 4

Dans cette section, nous détaillons les divers cas d'utilisation du quatrième sprint.

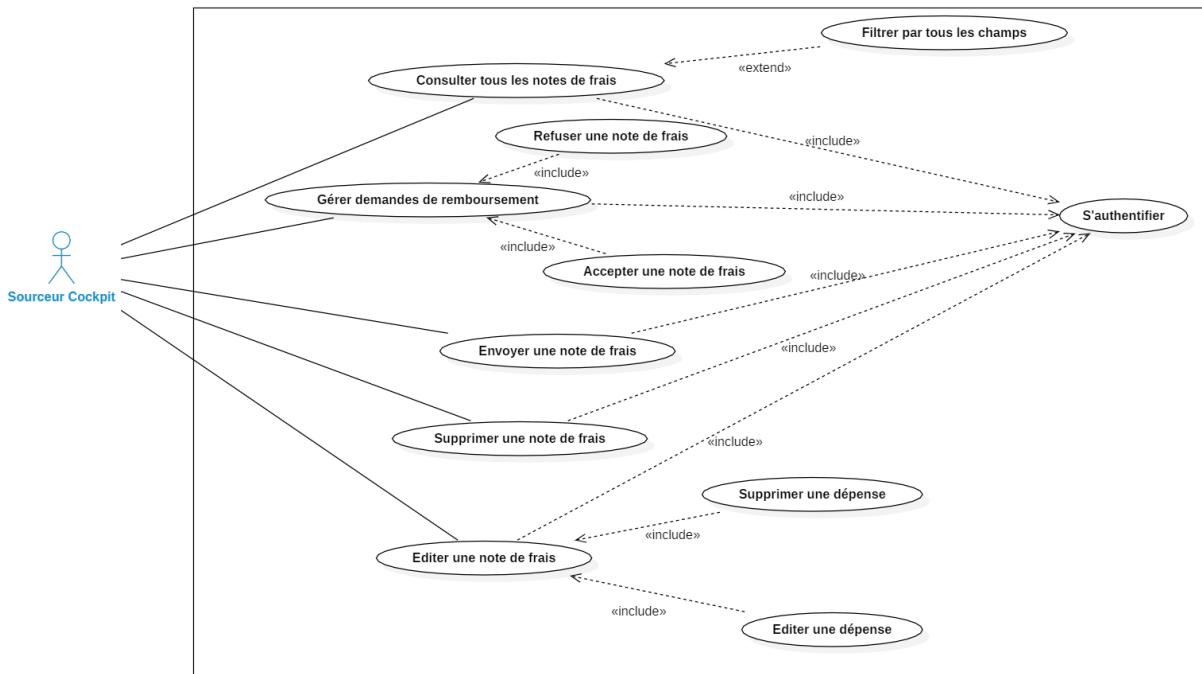


FIGURE 6.1 : Diagramme de cas d'utilisation raffiné du Sprint 4

À la suite de ce diagramme détaillé, nous rédigerons les descriptions textuelles des cas d'utilisation qui en découlent, en mettant l'accent sur ceux que nous jugeons les plus pertinents à expliquer.

Titre	Accepter une note de frais
Acteurs	Sourceur Cockpit
Pré-condition	Le Sourceur doit s'authentifier.
Post-condition	L'acceptation d'une nouvelle note de frais.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur consulte son espace des notes de frais. 2. Le système affiche la liste de toutes les notes de frais. 3. L'utilisateur choisit une note de frais à voir en détail. 4. L'utilisateur accepte une note de frais.
Scénario alternatif	4.1 Le système retourne un message d'erreur s'il y a une exception.

TABLEAU 6.2 : Description textuelle du cas d'utilisation «Accepter une note de frais»

6.4 Conception

Cette section est consacrée pour la conception de la partie Cockpit.

6.4.1 Diagramme de séquence

Le diagramme de séquence décrit comment le sourceur accepte une note de frais, montrant les interactions entre l'acteur et le système.

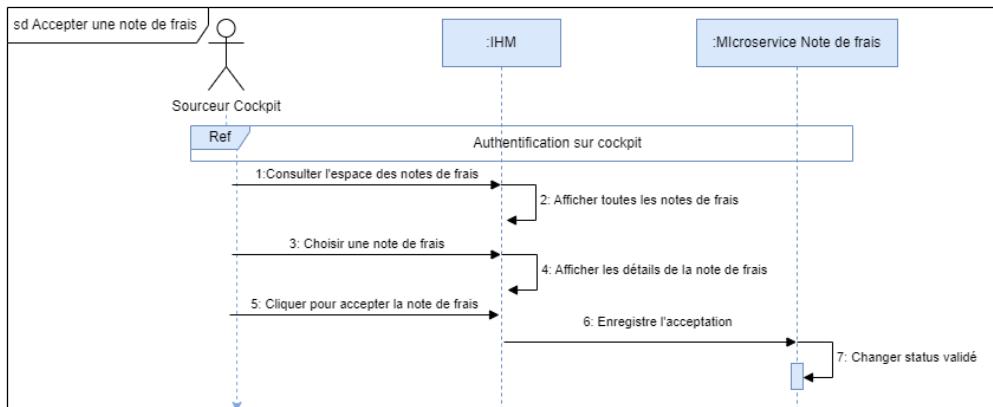


FIGURE 6.2 : Diagramme de séquence détaillé de "Accepter une note de frais"

6.4.2 Diagramme de classes

La figure illustre la structure statique du Sprint 4 schématisée dans un diagramme de classes.

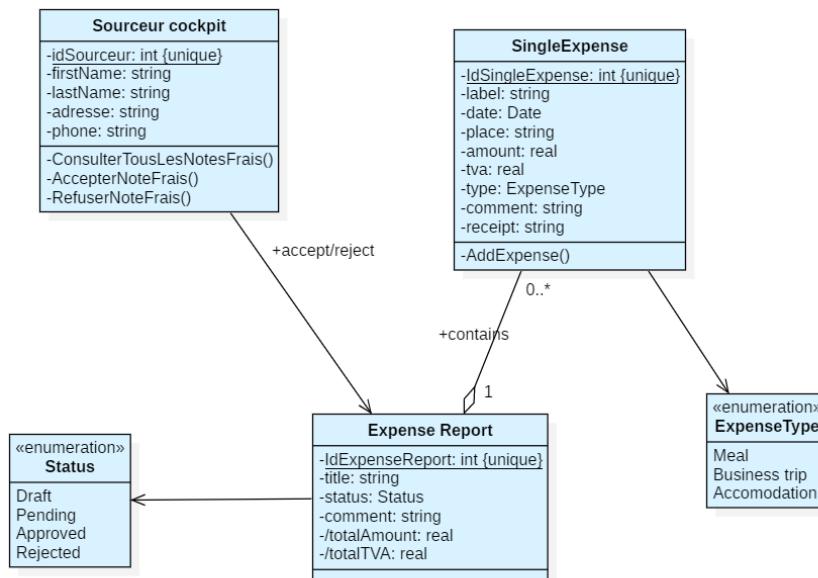


FIGURE 6.3 : Diagramme de classes du Sprint 4

6.5 Réalisation

Dans cette section, nous allons présenter les réalisations de ce sprint avec des captures d'écran.

6.5.1 Gestion des notes de frais sur la plateforme Cockpit

Sur la plateforme Cockpit, nous pouvons consulter la page de gestion des notes de frais comme le montre la figure 6.3.

ID	Title	Placement ID	Date	Freelancer	Client	First Approval	Status
14	Avril	101214	06-11-2024	Safa Ziedi	John Gaynor	fake client test note de frais	Rejected
13	MAI	101214	06-13-2024	Safa Ziedi	John Gaynor	fake client test note de frais	Pending
4	JUIN	101214	06-01-2024	Safa Ziedi	John Gaynor	fake client test note de frais	Draft
1	Févr	101215		Safa Ziedi	John Gaynor	fake client test note de frais	Draft

FIGURE 6.4 : Page de gestion des notes de frais sur la plateforme Cockpit

De plus, il est possible de filtrer par tous les champs disponibles, tels que l'ensemble des statuts, comme le montre la figure 6.4.

Placement ID	Date	Freelancer	Client	First Approval	Status	Total Amount
Ziedi	John Gaynor	fake client test note de frais	Draft X	Pending X	Approved X	0
Ziedi	John Gaynor	fake client test note de frais	Reset	Reset All		0
Ziedi	John Gaynor	fake client test note de frais	<input checked="" type="button"/> Draft		<input checked="" type="button"/> Pending	7
Ziedi	John Gaynor	fake client test note de frais	<input checked="" type="button"/> Draft		<input checked="" type="button"/> Approved	91
Ziedi	John Gaynor	fake client test note de frais	<input checked="" type="button"/> Approved			10

FIGURE 6.5 : Liste des notes de frais filtrées sur la page

Chapitre 6. Sprint 4 : Développement de la partie Cockpit

La figure 6.5 présente une vue détaillée d'une note de frais, où nous pouvons voir l'ensemble des dépenses qu'elle contient, afficher un reçu s'il existe, accepter/refuser/supprimer une note de frais, supprimer une dépense.

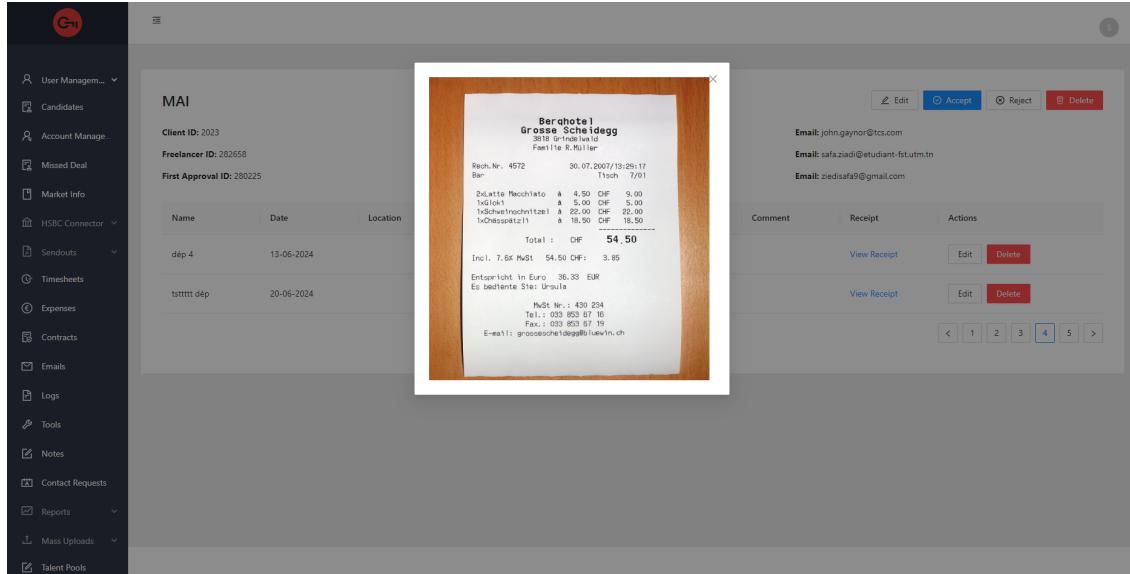


FIGURE 6.6 : Page de détail d'une note de frais

6.5.2 Page de modification

Dans la figure 6.6, nous illustrons le formulaire de modification d'une note de frais, avec les champs à ne pas modifier.

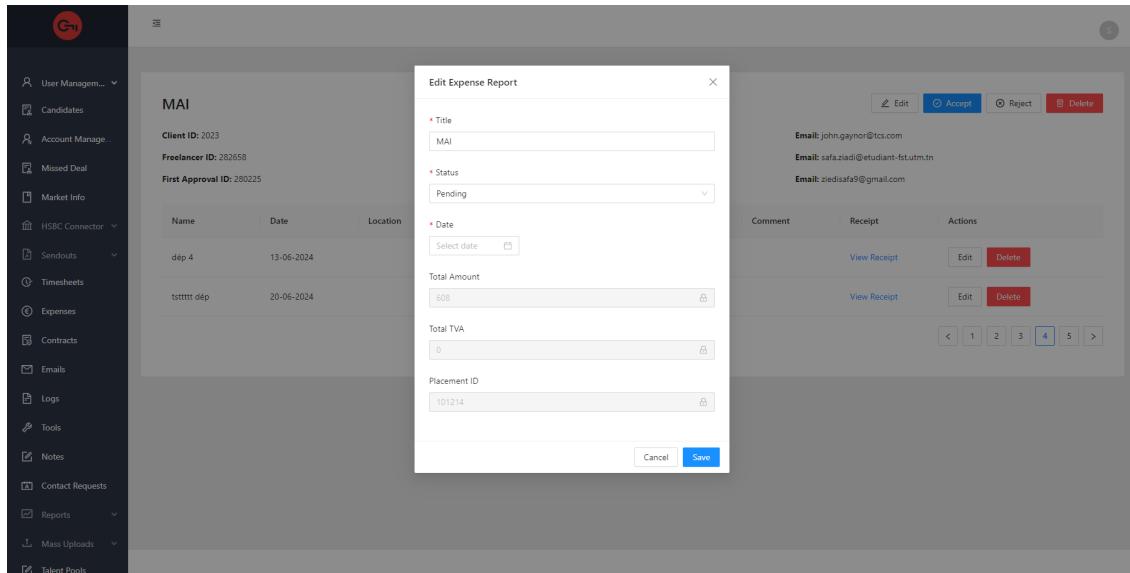


FIGURE 6.7 : Formulaire de modification d'une note de frais

De même, vous trouverez dans la figure 6.7 le formulaire de modification d'une dépense.

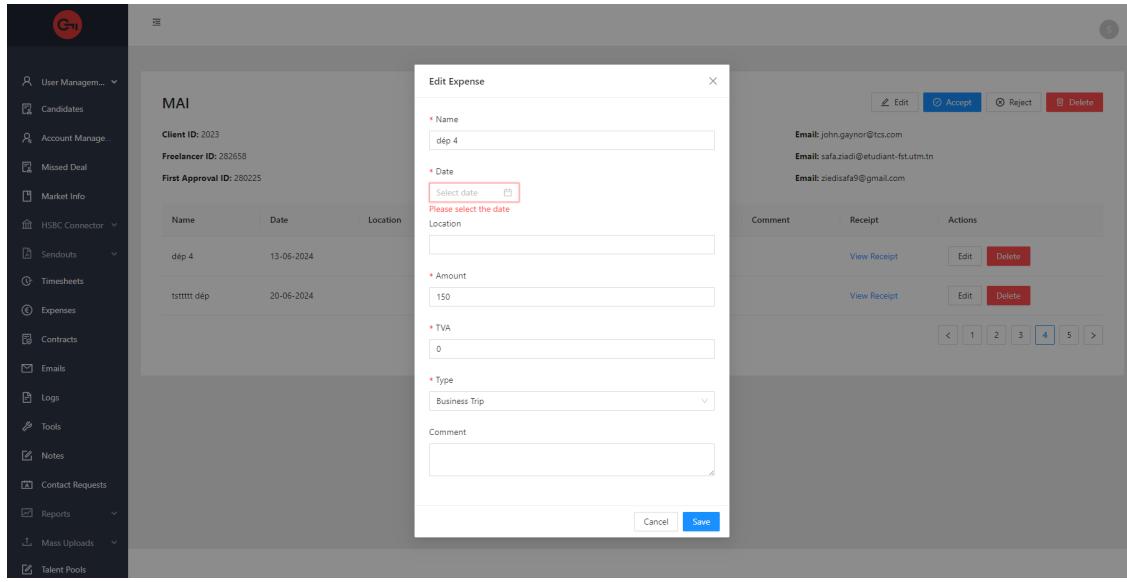


FIGURE 6.8 : Formulaire de modification d'une dépense

Conclusion

Au cours de ce chapitre, nous avons intégré toutes les parties réalisées dans notre travail sur la plateforme Cockpit. Nous avons expliqué comment nous avons développé notre solution en détaillant l'analyse et la conception, et nous avons conclu en présentant le travail à travers des interfaces de réalisation.

SPRINT 5 : INTÉGRATION ET DÉPLOIEMENT DE LA PARTIE COCKPIT

Plan

Introduction	71
1 Sprint Planning	71
2 Backlog du Sprint 5	71
3 Conception	72
4 Réalisation	73
Conclusion	75

Introduction

Ce chapitre est consacré au cinquième et dernier sprint intitulé "Intégration et déploiement de la partie Cockpit". Nous commencerons par examiner le sprint planning, puis le backlog du sprint, qui comprend les user stories à réaliser. Ensuite, nous passerons à l'analyse et la conception, et nous terminerons avec la phase de réalisation.

7.1 Sprint Planning

Ce sprint de deux semaines a pour objectif d'intégrer le module de notes de frais, puis de le déployer, en particulier la partie Cockpit. Cela implique de le mettre en pré-production pour les testeurs afin d'améliorer l'expérience des utilisateurs finaux, qui sont les administrateurs de notre plateforme, avant le déploiement en production.

7.2 Backlog du Sprint 5

ID	User Story	Tâches	Estimation (HJ)
5.1	En tant que sourceur, je peux accéder à l'espace des notes de frais dans la plateforme Cockpit.	Installation du projet	0.5
		Découverte de l'architecture du projet	1
		Intégrer le micro-service dans l'ensemble des micro-services.	1
		Assurer la communication entre le ms-expenses et les autres micro-services.	1.5
		Implémenter la navigation vers l'espace des notes de frais.	1
		Tester l'accès et la fonctionnalité de l'espace des notes de frais.	1

ID	User Story	Tâches	Estimation (HJ)
5.2	En tant que sourceur, je peux ajouter des permissions sur les notes de frais.	Ajouter des permissions de lecture et de modification sur les notes de frais.	2
		Tester l'ajout d'une permission.	1
5.3	En tant que sourceur, je peux accorder une permission sur les notes de frais à un autre sourceur.	Tester l'accord d'une autorisation auprès d'un sourceur.	1

TABLEAU 7.1 : Backlog du Sprint 5

7.3 Conception

Pour la conception, nous allons présenter le diagramme de déploiement.

7.3.1 Diagramme de déploiement

Dans ce diagramme de déploiement, la communication entre le navigateur et le serveur web se fait via HTTP, permettant au navigateur d'envoyer des requêtes comme GET et POST pour récupérer des pages web, des images et d'autres ressources. Cette approche basée sur un modèle requête-réponse facilite une interaction standardisée et compatible avec les navigateurs modernes. En parallèle, la communication entre le serveur web et la base de données utilise TCP/IP, assurant une transmission fiable et sécurisée des données entre les deux serveurs. Cette combinaison de protocoles optimise l'efficacité et la sécurité des échanges dans l'environnement réseau.

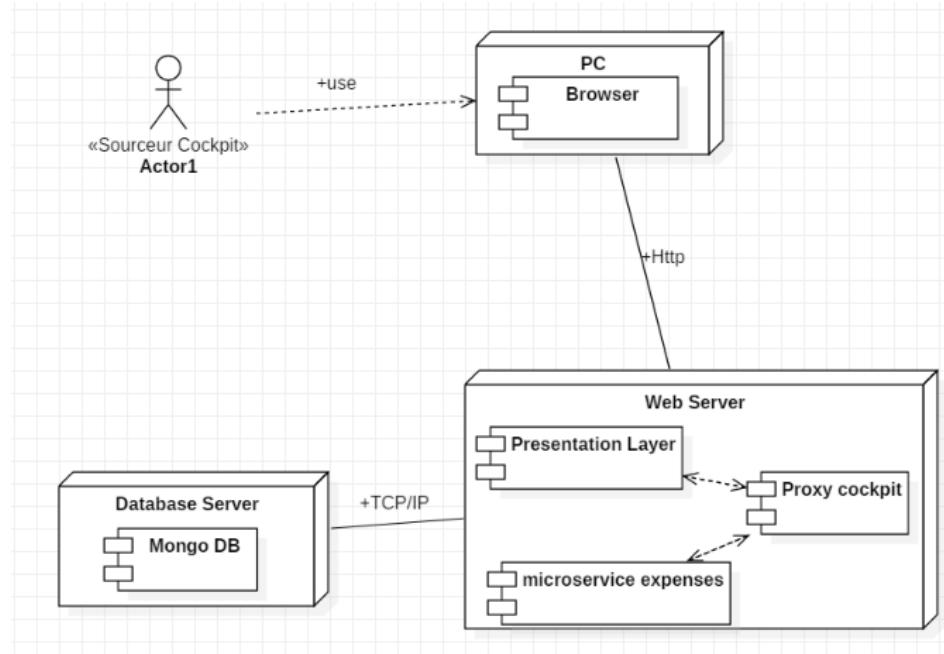


FIGURE 7.1 : Diagramme de déploiement Sprint 5

7.4 Réalisation

Nous détaillerons les réalisations de ce sprint dans cette section, en fournissant des captures d'écran pour illustrer chaque point.

Dans la figure 7.2, vous pouvez voir la page de connexion du Cockpit. Cette interface permet aux utilisateurs de s'identifier afin d'accéder à leurs fonctionnalités personnalisées en fonction de leur rôle.

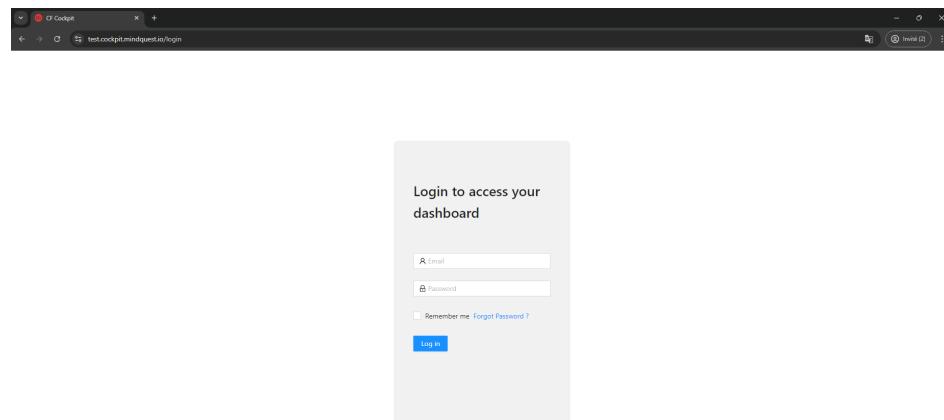


FIGURE 7.2 : Page de connexion sur Cockpit

Dans la figure 7.3, l'espace des notes de frais sur Cockpit est illustré. Cette section permet aux

Chapitre 7. Sprint 5 : Intégration et déploiement de la partie Cockpit

utilisateurs de consulter et de gérer les notes de frais.

The screenshot shows the Cockpit application interface. On the left, there is a vertical sidebar with various menu items: User Management, Candidates, Account Management, Missed Deal, Market Info, HSBC Connection, Sendouts, Timesheets, Expenses (which is currently selected), Contracts, Emails, Logs, Tools, Notes, Contact Requests, Reports, Mass Uploads, and Talent Pools. The main content area is titled "Expenses Reports". It features a table with columns: ID, Title, Placement ID, Date, Freelancer, Client, First Approval, and Status. There are four rows of data:

ID	Title	Placement ID	Date	Freelancer	Client	First Approval	Status
35	Test3	3030300	07-11-2024	fake candidate fr site	Dayo Adeyemi	fake client test client	Draft
34	Test2	3030307	07-05-2024	fake candidate fr site	Dayo Adeyemi	fake client test client	Rejected
33	Test	3030309	07-05-2024	fake candidate fr site	Dayo Adeyemi	fake client test client	Approved
32	July	101214	07-06-2024	Safa Ziedi	John Gaynor	fake client test note de frais	Draft

FIGURE 7.3 : Espace des notes de frais sur Cockpit

La figure 7.4 montre les paramètres de permissions au sein du Cockpit. Cette interface permet de configurer les accès, avec trois types de permissions disponibles : lecture, écriture, ou administrateur (tous les droits).

The screenshot shows the Cockpit application interface. On the left, there is a vertical sidebar with various menu items: User Management, Users, Permissions (which is currently selected), Candidates, Account Management, Missed Deal, Market Info, HSBC Connection, Sendouts, Timesheets, Expenses, Contracts, Emails, Logs, Tools, Notes, Contact Requests, Reports, and Mass Uploads. The main content area is titled "Permissions". It features a table with columns: Name and several icons representing different permission levels (e.g., edit, read, administrator). There are four rows of data:

Name	Permissions
Expense Edit	Edit Read Administrator
Expense Only Read	Read Edit Administrator
Expense Manager	Administrator Edit Read

Below this table, there is another section with rows for "Timesheet manager", "CV GENERATOR Edit only", "CV GENERATOR read only", and "CV GENERATOR". Each row has a set of three icons (blue, green, red) corresponding to the permission levels.

FIGURE 7.4 : Les permissions

Enfin, la figure 7.5 illustre la procédure pour accorder une permission à un autre sourceur. Cette fonctionnalité est cruciale pour la gestion des accès et la collaboration au sein de l'application.

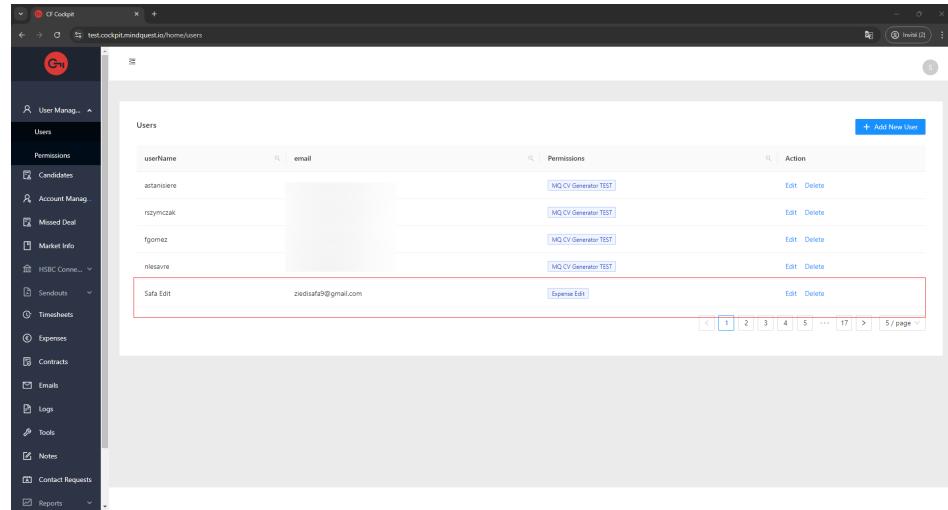


FIGURE 7.5 : Accorder une permission

Conclusion

Lors de ce dernier sprint, nous avons présenté le Sprint Backlog, le diagramme de déploiement et nous avons fini par démontrer le résultat grâce à des captures d'écran de notre solution.

Conclusion générale

Ce travail s'inscrit dans le cadre de notre projet de fin d'études réalisé au sein de l'entreprise Tech-Expert. Son objectif principal a été de développer un module de gestion des notes de frais et de l'intégrer dans la plateforme Mindquest afin de moderniser et optimiser le processus de gestion des dépenses, offrant ainsi une solution efficace et intuitive pour les utilisateurs de la plateforme.

Pour évaluer ce travail, nous avons eu l'opportunité de travailler sur l'application Web existante « Mindquest », tout en développant une nouvelle solution répondant aux besoins fonctionnels de l'entreprise cliente. Cette expérience m'a permis d'appréhender les défis techniques et organisationnels inhérents à un tel projet.

La première étape de ce travail a consisté à réaliser une étude approfondie du contexte général du sujet, ainsi qu'une analyse critique de la solution existante afin d'identifier les lacunes et les besoins à adresser. Ensuite, nous avons identifié et défini les besoins fonctionnels et non fonctionnels auxquels notre application devait répondre, tout en présentant sa conception globale.

Chaque fonctionnalité majeure a été abordée dans un chapitre dédié, où nous avons présenté une analyse et conception détaillées et exposé le travail réalisé à l'aide de captures d'écran et de descriptions pertinentes.

La contribution de ce projet a été extrêmement enrichissante sur le plan technologique. Nous avons acquis des compétences précieuses et maîtrisé les nouvelles technologies de pointe, notamment l'architecture microservices, Express JS, React JS et MongoDB. Nous avons, aussi, appris les bonnes pratiques nécessaires pour livrer un produit logiciel de qualité.

En conclusion, ce travail a atteint les objectifs fixés au début de notre stage. Cependant, des perspectives innovantes s'ouvrent à notre application. Nous envisageons d'ajouter la fonctionnalité OCR (Reconnaissance Optique de Caractères) à la solution, afin de permettre la détection automatique des informations pertinentes à partir des reçus téléchargés dans la plateforme, ainsi que d'améliorer la responsivité pour une meilleure expérience utilisateur sur tous les appareils.

Enfin, nous espérons que ce modeste travail contribuera au progrès, à l'évolution et à la satisfaction de l'équipe de Tech-Expert, de nos encadrants et des membres du jury.

Netographie

- [1] <https://tech-expert.io/accueil/>, Consulté le 3 Avril 2024.
- [2] <https://mindquest.io/>, Consulté le 1 Mars 2024.
- [3] <https://fr.mindquest.io/blog/news/22179/club-freelance-devient-mindquest-le-16-mai-2022/>, Consulté le 5 Avril 2024.
- [4] <https://visite.mindquest.io/kit-demarrage-freelance/>, Consulté le 12 Avril 2024.
- [5] <https://blog-gestion-de-projet.com/manifester-agile-valeurs-et-principes/>, Consulté le 1 Mai 2024.
- [6] <https://www.bocasay.com/fr/methode-scrum-benefices-developpements-web/>, Consulté le 29 Mai 2024.
- [7] https://edutechwiki.unige.ch/fr/Visual_studio_code, Consulté le 1 Mars 2024.
- [8] <https://www.mongodb.com/fr-fr/products/tools/compass>, Consulté le 2 Mars 2024.
- [9] <https://www.atlassian.com/fr/git/tutorials/what-is-git>, Consulté le 12 Mars 2024.
- [10] <https://bitbucket.org/product/fr/guides/getting-started/overview#a-brief-overview-of-bitbucket>, Consulté le 3 Mars 2024.
- [11] <https://fr.overleaf.com/project>, Consulté le 11 Mars 2024.
- [12] <https://thetribe.io/metier-typescript/>, Consulté le 20 février 2024.
- [13] <https://kinsta.com/fr/base-de-connaissances/qu-est-express-js/>, Consulté le 20 février 2024.
- [14] <https://www.purestorage.com/fr/knowledge/what-is-mongodb.html>, Consulté le 12 Mars 2024.
- [15] https://fr.wikipedia.org/wiki/Representational_state_transfer, Consulté le 19 février 2024.
- [16] <https://www.nicolashachet.com/blog/developpement-php/larchitecture-rest-expliquee-en-5-regles/>, Consulté le 15 Juin 2024.
- [17] <https://www.redhat.com/fr/topics/api/what-is-graphql>, Consulté le 3 Mars 2024.
- [18] <https://radixweb.com/blog/graphql-vs-rest/>, Consulté le 23 Juin 2024.

- [19] <https://rohitgupta4.medium.com/what-is-graphql-bab7021c6dfe/>, Consulté le 4 Juillet 2024.
- [20] <https://blog.cellenza.com/cloud-2/graphql-api-presentation-et-cas-dusage/>, Consulté le 12 Mars 2024.
- [21] <https://khalilstemmller.com/articles/graphql/graphql-architectural-advantages/>, Consulté le 19 Juillet 2024.
- [22] <https://medium.com/@ianovenden/redux-websocket-integration-c1a0d22d3189/>, Consulté le 19 Juillet 2024.

ملخص

المشروع الحالي يتضمن تصميم وتنفيذ وحدة لإدارة تقارير النفقات وتوليد الفواتير الإلكترونية. تهدف هذه الوحدة إلى تمكين المستشارين من الحصول على موافقة العملاء أو شركات التوظيف على تقارير النفقات التي تكتبواها، وذلك لاسترداد أموالهم. باستخدام تقنيات متطرفة مثل ReactJs للواجهة الأمامية و ExpressJS للواجهة الخلفية، واتباع نهج رشيق باستخدام إطار عمل ضروري، يهدف المشروع إلى تقديم حل فعال وسهل الاستخدام وبدائي لمستخدمي المنصة.

كلمات مفاتيح : Microservices, MongoDB, Express, Nodejs, React, MERN

Résumé

Le présent projet consiste en la conception et l'implémentation d'un module de gestion des notes de frais et de génération de factures électroniques. Ce module doit permettre aux consultants de faire valider par leurs clients ou leurs sociétés de portage les notes de frais qu'ils ont engagées, afin d'obtenir leurs remboursements. En utilisant des technologies de pointe telles que ReactJs pour le front-end et ExpressJS pour le back-end, et en suivant une approche agile avec le framework SCRUM, le projet a visé à fournir une solution efficace, conviviale et intuitive pour les utilisateurs de la plateforme.

Mots clés : MERN, React, Nodejs, Express, MongoDB, Microservices

Abstract

The current project involves the design and implementation of an expense report management module and electronic invoice generation. This module aims to enable consultants to have their expense reports approved by their clients or umbrella companies in order to receive reimbursements. By utilizing cutting-edge technologies such as ReactJs for the front end, and ExpressJS for the back end, and by following an agile approach with the SCRUM framework, the project aims to deliver an efficient, user-friendly, and intuitive solution for platform users.

Keywords : MERN, React, Nodejs, Express, MongoDB, Microservices