

Liste des Figures

1.1	Entreprise logo [1]	5
1.2	Organizational Chart [1]	6
1.3	4D Map [1]	7
1.4	Logo de Skype [2]	10
2.1	Schéma explicatif de la méthode Agile.	14
2.2	Logo du 4D [1]	21
2.3	Logo du 4D Client [10]	23
2.4	Logo du 4D Server [1]	24
2.5	Logo du Visual Studio Code [9]	25
2.6	Logo du Axios [4]	26
2.7	Logo du ChartJs [5]	27
2.8	Logo du TailwindCss [18]	28
2.9	Logo du Git [6]	29
2.10	Logo du Gitlab [6]	30
2.11	Stack Overflow Annual Developer Survey 2023 [11]	33
2.12	State Of Dev In Morocco 2023 [12]	33
2.13	Classement des frameworks python 2023 selon JetBrains [14]	37
2.14	Classement des SGBD les plus utilisés avec Django [15]	38
3.1	Diagramme de Gantt	43
3.2	Diagramme des cas d'utilisation globale.	47
3.3	Diagramme des cas d'utilisation en détails	48
3.4	Diagramme de séquence du cas d'utilisation "S'authentifier"	53
3.5	Diagramme de de séquence du cas d'utilisation "ajouter une note"	54
3.6	Diagramme de séquence du cas d'utilisation "ajouter une ligne"	55

3.7	Diagramme de séquence du cas d'utilisation "modifier une note"	56
3.8	Diagramme de séquence du cas d'utilisation "modifier une ligne"	56
3.9	Diagramme de classe	57
4.1	Page de connexion.	60
4.2	Page de connexion.	61
4.3	Page d'accueil principale.	62
4.4	Page d'accueil de gestion de note de frais.	62
4.5	liste des notes de frais.	63
4.6	Filtrage et Recherche.	63
4.7	Ajouter une note.	64
4.8	Détail d'une note.	64
4.9	Modifier une note.	65
4.10	voir ligne spécifique d'une note.	65
4.11	Liste des lignes de frais.	66
4.12	détail de la ligne de frais.	66
4.13	Liste des lignes de frais.	67
4.14	Ajout d'une ligne.	67
4.15	Liste des catégories de frais.	68
4.16	Statistiques des frais.	69

Liste des Tables

2.1	Comparaison entre React [3] et Angular.	35
2.2	Classement des langages de programmation selon l'indice TIOBE. [13]	36
2.3	Comparaison entre 4D, Python et C++.	37
3.1	Description du cas d'utilisation "S'authentifier".	48
3.2	Description du cas d'utilisation "Réinitialiser le mot de passe".	49
3.3	Description du cas d'utilisation "Ajouter une note".	49
3.4	Description du cas d'utilisation "Ajouter une ligne".	50
3.5	Description du cas d'utilisation "Visualiser la liste des notes".	50
3.6	Description du cas d'utilisation "Modifier une note".	51
3.7	Description du cas d'utilisation "Modifier une ligne".	51

Table des Matières

Liste des Figures	2
Liste des Tables	3
Table des Matières	6
Introduction générale	1
1 Contexte général du projet	3
1.1 Présentation de l'organisation hôte	5
1.1.1 Aperçu de l'entreprise	5
1.1.2 Organigramme de l'entreprise	6
1.1.3 Le groupe 4D	6
1.1.4 4D_Framework	7
1.1.5 Cadre du projet	8
1.1.5.1 Définition et problématique	8
1.1.6 Objectifs du projet	9
1.1.7 Tâches du projet	9
1.1.8 Outils de collaboration	10
2 Concepts et technologies	12
2.1 Concepts fondamentaux	13
2.1.1 Développement Logiciel (Software Engineering)	13
2.1.1.1 Définition et Importance	13
2.1.1.2 Les Étapes du Développement Logiciel	13
2.1.2 Méthodologie de Développement : SCRUM	14
2.1.3 La Méthodologie Agile en Pratique	15

2.1.4	Validation de la Solution	16
2.1.4.1	Validation de la Solution par Tests Manuels	16
2.1.4.2	Simulation de Scénarios d'Utilisation Réelle	16
2.1.4.3	Tests des API REST	17
2.1.4.4	Intégration Backend-Frontend	17
2.1.5	Résultats	17
2.1.6	Cycles de développement	17
2.1.7	Les Outils Utilisés en Ingénierie Logicielle	18
2.1.7.1	Conclusion	18
2.1.8	Conception de Bases de Données	19
2.1.8.1	Définition et Importance	19
2.1.8.2	Modélisation des Données	19
2.1.8.3	Normalisation et Optimisation des Bases de Données	19
2.2	Les technologies	20
2.2.1	ReactJS	20
2.2.2	4D	21
2.2.3	4D Client	23
2.2.4	4D serveur	24
2.2.5	Qu'est ce qu'un composant 4D ?	25
2.2.6	Visual Studio Code	25
2.2.7	Axios	26
2.2.8	Chart JS	27
2.2.9	TailwindCss	28
2.2.10	Systèmes de contrôle de version	29
2.2.10.1	Git	29
2.2.10.2	GitLab	30
2.3	Benchmarking des outils	32
2.3.1	Frontend	32
2.3.2	Backend	35
2.3.3	Conclusion	39
3	Étude, analyse, spécification des besoins et conception	41
3.1	Analyse et spécification des besoins	42

3.1.1	Description de l'existant	42
3.1.2	Diagramme de Gantt	42
3.2	Les exigences	43
3.2.1	Les exigences fonctionnels	43
3.2.2	Les exigences non fonctionnels	44
3.3	Modélisation des exigences	45
3.3.1	Présentation de l'UML	45
3.3.2	Diagrammes des cas d'utilisations	47
3.3.3	Description de quelques cas d'utilisation	48
3.3.4	Conception de la solution	52
3.3.4.1	Diagrammes des séquences	52
3.3.4.2	Diagramme de classes	57
4	Mise en oeuvre de la solution	59
4.1	Réalisation	60
4.1.1	Authentification	60
4.1.1.1	Connexion	60
4.1.1.2	Mot de passe oublié	61
4.1.2	Page d'accueil	61
4.1.3	Le module de gestion de note de frais	62
4.1.3.1	Le module de note de frais	62
4.1.3.2	Le module de ligne de frais	65
4.1.3.3	Le module des catégories de frais	68
4.1.3.4	Statistiques des dépenses	68
4.2	Validation de la solution	69
4.2.1	Validation de la solution par Tests Manuels	69
4.2.1.1	Simulation de Scénarios d'Utilisation Réelle	69
4.2.1.2	Tests des API REST	70
4.2.1.3	Intégration Backend-Frontend	70
4.2.2	Résultats	70
	Conclusion et Perspectives	72
	Bibliographie	75

Introduction Générale

La gestion des notes de frais représente une tâche essentielle mais souvent complexe pour les entreprises. Les collaborateurs, lors de leurs missions, sont amenés à engager divers frais tels que des déplacements, des repas, ou des hébergements. Un traitement inefficace de ces dépenses peut entraîner non seulement des frustrations parmi les employés, mais également des inefficacités administratives et des risques de non-conformité financière.

Traditionnellement, les entreprises utilisaient des processus manuels pour gérer les notes de frais, impliquant des formulaires papier, des reçus physiques et des approbations hiérarchiques multiples. Cette méthode présente plusieurs inconvénients majeurs. La collecte, la vérification et le traitement des notes de frais sont chronophages et requièrent une implication importante des ressources humaines. Les processus manuels sont sujets à des erreurs de saisie, des omissions et des risques de fraude. De plus, la visibilité limitée sur les dépenses en cours et la difficulté de suivi en temps réel des frais peuvent conduire à une mauvaise gestion des budgets.

Face à ces défis, l'implémentation d'une solution numérique de gestion des notes de frais apparaît comme une nécessité. Le projet que nous présentons vise à développer un système intégré et automatisé qui facilitera la gestion des dépenses professionnelles. L'automatisation permettra de réduire les interventions manuelles en automatisant la saisie, la validation et le remboursement des notes de frais. Le processus de gestion des frais sera ainsi accéléré, permettant un remboursement plus rapide aux employés et une réduction du temps administratif. Une visibilité en temps réel sur les dépenses offrira un meilleur contrôle budgétaire et une prise de décision éclairée. Enfin, le système assurera le respect des politiques internes de l'entreprise et des réglementations externes en vigueur.

Pour atteindre ces objectifs, notre projet s'appuie sur l'utilisation de technologies modernes, telles que les applications mobiles pour la saisie des frais, l'intégration de systèmes de gestion des ressources humaines (ERP) et l'utilisation d'algorithmes de vérification automatique.

Le présent rapport est structuré en plusieurs parties, chacune détaillant un aspect spécifique du projet. L'analyse des besoins examine les exigences fonctionnelles et non fonctionnelles du système

de gestion des notes de frais, en identifiant les besoins des utilisateurs et les contraintes du projet. La conception du système détaille l'architecture du système, les choix technologiques et les différents composants de la solution proposée. L'implémentation technique décrit le processus de développement, les outils utilisés et les techniques de programmation mises en œuvre pour réaliser le projet. Les tests et la validation présentent les différentes phases de test, les méthodologies employées pour assurer la qualité du système et les résultats obtenus. Le déploiement et la formation couvrent les stratégies de déploiement du système, les étapes de mise en production et les plans de formation des utilisateurs. L'évaluation des bénéfices analyse les avantages attendus du nouveau système en termes d'efficacité, de transparence et de conformité, et présente des indicateurs de performance. Enfin, la conclusion et les perspectives récapitulent les principaux résultats du projet, discutent des défis rencontrés et des solutions apportées, et explorent les perspectives d'évolution futures du système de gestion des notes de frais.

Chapter 1

Contexte général du projet

Introduction

La gestion des notes de frais représente une tâche essentielle mais souvent complexe pour les entreprises. Les collaborateurs, lors de leurs missions, sont amenés à engager divers frais tels que des déplacements, des repas, ou des hébergements. Un traitement inefficace de ces dépenses peut entraîner non seulement des frustrations parmi les employés, mais également des inefficacités administratives et des risques de non-conformité financière.

L'entreprise 4D Logiciels Maroc, l'organisation hôte de notre stage, nous a confié la mission de développer un système intégré de gestion des notes de frais. Traditionnellement, les entreprises utilisaient des processus manuels pour gérer les notes de frais, impliquant des formulaires papier, des reçus physiques et des approbations hiérarchiques multiples. Cette méthode présente plusieurs inconvénients majeurs, notamment la consommation de temps et de ressources, les risques d'erreurs et de fraudes, et le manque de transparence dans la gestion des budgets.

La problématique centrale de ce projet réside dans les inefficacités et les risques associés aux processus manuels de gestion des notes de frais. Les entreprises, y compris 4D Logiciels Maroc, cherchent à moderniser et à automatiser ces processus pour gagner en efficacité, en transparence et en conformité. Les principaux objectifs du projet sont les suivants : réduire les interventions manuelles en automatisant la saisie, la validation et le remboursement des notes de frais ; accélérer le processus de gestion des frais, permettant un remboursement plus rapide aux employés et une réduction du temps administratif ; offrir une visibilité en temps réel sur les dépenses, permettant un meilleur contrôle budgétaire et une prise de décision éclairée ; et assurer le respect des politiques internes de l'entreprise et des réglementations externes en vigueur.

1.1 Présentation de l'organisation hôte

1.1.1 Aperçu de l'entreprise



Figure 1.1: Entreprise logo [1]

Notre stage de fin d'études s'est déroulé au sein de l'entreprise 4D Logiciels Maroc, une société faisant partie du marché de l'édition de logiciels, dont le siège social est situé en France. Fondée en 1984, 4D Logiciels Maroc est reconnue pour ses outils de développement innovants qui permettent la création de solutions professionnelles efficaces et fiables pour les entreprises

Notre stage de fin d'études s'est déroulé au sein de l'entreprise 4D Logiciels Maroc, une société faisant partie du marché de l'édition de logiciels, dont le siège social est situé en France. Fondée en 1984, 4D Logiciels Maroc est reconnue pour ses outils de développement innovants qui permettent la création de solutions professionnelles efficaces et fiables pour les entreprises. 4D Logiciels Maroc fait partie d'un grand groupe appelé 4D Software. En effet, le fondateur de 4D et actuel Directeur Général (CEO) du groupe, M. Laurent Ribardi'ere, a créé 4D en 1984 avec l'ambition de simplifier la création d'applications professionnelles pour les entreprises grâce à une base de données relationnelle entièrement graphique. Cette innovation, saluée par l'industrie du logiciel, a façonné l'industrie informatique moderne et inspiré des générations de développeurs. Sous sa direction, 4D est restée une entreprise pionnière, reconnue pour son approche créative des nouvelles technologies.

4D est ainsi devenue l'un des principaux éditeurs de logiciels français avec une portée internationale grâce à sa présence sur 5 continents et à ses filiales et distributeurs dans plus de 10 pays. Le succès de 4D découle de sa capacité à relever les défis de son époque, grâce à une plateforme évolutive, simplifiant la création d'expériences client réussies sur mobile, web et desktop. En effet, en 1987, 4D Software a offert le premier système de gestion de bases de données relationnelles 32 bits, puis a maintenu son leadership en offrant le premier :

- Serveur client intégré.
- Serveur web intégré.

La version 4D 2004 a été lancée en tant que produit permettant aux développeurs de créer des applications autonomes, client-serveur, web et à architecture orientée services (SOA) sans apporter de modifications au code. Plus récemment, 4D a introduit une plateforme de développement JavaScript qui facilite la création d'applications professionnelles en utilisant la gamme de produits Wakanda. Dans le but d'étendre ses activités à tous les continents, notamment en Afrique, 4D a décidé d'établir sa première filiale sur le continent en 2012 au Maroc, une filiale qui reste la seule à ce jour sur le continent.

1.1.2 Organigramme de l'entreprise

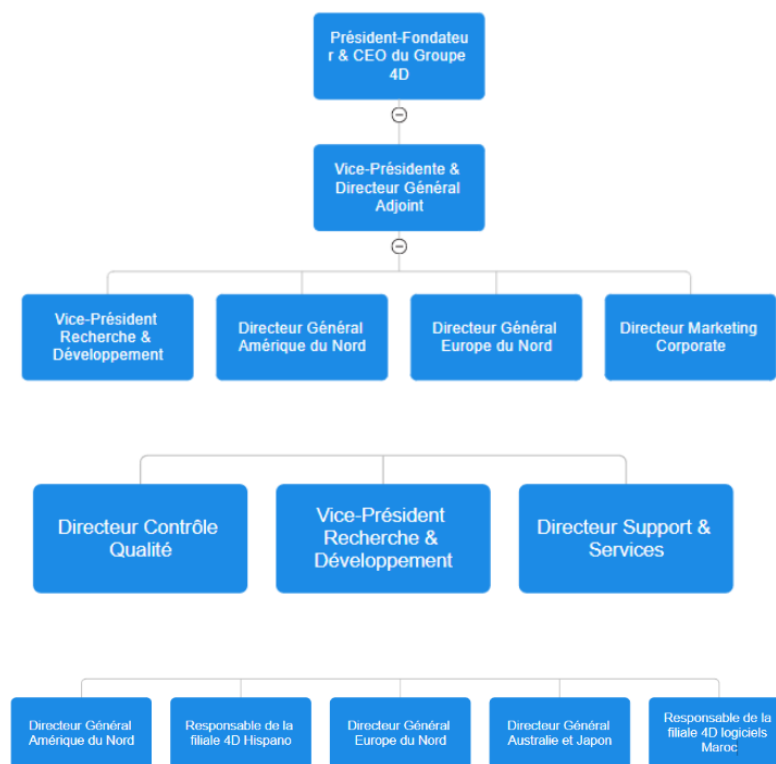


Figure 1.2: Organizational Chart [1]

1.1.3 Le groupe 4D

Le groupe 4D, dont le siège social est en France, possède cinq filiales internationales, notamment aux États-Unis, en Allemagne, au Japon, en Australie et au Maroc. Le groupe dispose également de plusieurs distributeurs et représentants à travers le monde, y compris en Belgique, en Italie, en République tchèque, à Singapour, en Suisse, en Pologne, etc. Les rôles présents chez 4D Logiciels Maroc sont similaires, voire identiques, à ceux de la filiale en France, allant du support technique et du

développement au contrôle de qualité pour les produits "4D" et "Wakanda". La figure suivante montre les différentes filiales, distributeurs affiliés et partenaires de 4D Logiciels dans le monde entier.



Figure 1.3: 4D Map [1]

1.1.4 4D_Framework

4D est une plateforme de développement productive qui permet aux clients de se concentrer sur leur modèle de données et les règles et spécificités de leur entreprise [?]. Le framework 4D prend en charge l'exécution native de leur code d'application sur macOS et Windows. 4D Server exécute leurs applications simultanément sur des postes de travail / clients mobiles et sur le Web. Ils peuvent déployer des applications entièrement personnalisées sous leur propre marque. 4D est un système de gestion de base de données relationnelle avec un langage de programmation de quatrième génération. Environnement de développement intégré, 4D [1] comprend :

- un compilateur
- un débogueur
- un système de sauvegarde
- un serveur web
- un serveur et client de services web

1.1.5 Cadre du projet

1.1.5.1 Définition et problématique

Dans le contexte du développement d'une application de gestion des dépenses, le projet est confronté à plusieurs défis. L'un des principaux problèmes est d'assurer l'intégration transparente de la fonctionnalité de gestion des notes de frais en utilisant React pour le front-end et 4D pour le back-end. Cette intégration nécessite la conception d'un système robuste permettant aux utilisateurs de créer, modifier et supprimer efficacement des notes de frais tout en garantissant la cohérence, la sécurité et la conformité des données.

De plus, l'application doit gérer divers types de dépenses, notamment les frais de déplacement, d'hébergement et de repas, chacun avec ses propres exigences et validations. La gestion de ces différents types de dépenses et de leurs documents associés, tels que les reçus et les factures, pose un défi important en termes d'organisation des données et de conception de l'interface utilisateur. En outre, l'application doit prendre en charge plusieurs utilisateurs avec différents rôles, tels que les employés, les gestionnaires et les administrateurs, chacun nécessitant différents niveaux d'accès et de fonctionnalités. La mise en place d'un système d'authentification et d'autorisation des utilisateurs sécurisé et évolutif est crucial pour garantir la confidentialité et la sécurité des données.

Un autre défi est de concevoir une interface utilisateur conviviale et intuitive pour saisir et gérer les informations des notes de frais. Cela implique de créer des formulaires pour capturer les détails des dépenses, d'intégrer des fonctionnalités pour joindre des reçus et des factures, et de fournir des fonctionnalités de recherche, de filtrage et de tri des données de dépenses.

De plus, l'application doit gérer une logique métier complexe, telle que le calcul des dépenses totales, l'application des politiques et des limites de dépenses, et la génération de rapports. La mise en œuvre de ces fonctionnalités nécessite une compréhension approfondie des besoins métier et une planification minutieuse pour garantir l'exactitude et l'efficacité.

En résumé, le développement d'une fonction de gestion des dépenses pour l'application nécessite de relever divers défis liés à l'intégration, à la gestion des données, à la conception de l'interface utilisateur, à la gestion des utilisateurs et à la logique métier. Surmonter ces défis nécessite une combinaison d'expertise technique, de créativité et d'attention aux détails pour fournir une solution réussie et conviviale.

1.1.6 Objectifs du projet

Les objectifs de ce projet sont d'améliorer les fonctionnalités du système existant en développant un utilitaire spécifiquement conçu pour la gestion des notes de frais. Cet utilitaire permettra de rationaliser le processus de création, de soumission et d'approbation des notes de frais, améliorant ainsi l'efficacité et la précision de la gestion des dépenses. Les principaux objectifs incluent l'intégration transparente de l'utilitaire dans le système ERP existant, la fourniture d'une interface conviviale pour un accès et une navigation faciles, la garantie de l'intégrité et de la sécurité des données, et l'amélioration des capacités de reporting pour une meilleure prise de décision.

1.1.7 Tâches du projet

- Analyse des besoins
 - Comprendre le processus actuel de gestion des dépenses dans le système.
 - Identifier les besoins spécifiques et les fonctionnalités nécessaires pour l'utilitaire de gestion des notes de frais.
- Conception
 - Concevoir l'interface utilisateur pour l'utilitaire de gestion des notes de frais en utilisant React.
 - Planifier les modifications du schéma de base de données nécessaires pour stocker et gérer efficacement les données des notes de frais en utilisant 4D.
- Développement
 - Implémenter les composants de l'interface utilisateur en utilisant React.
 - Développer la fonctionnalité backend en utilisant 4D pour gérer les opérations pour les notes de frais.
 - Intégrer le frontend et le backend pour assurer une communication transparente entre React et 4D.
- Tests Manuels
 - Effectuer des tests manuels pour vérifier la fonctionnalité de l'application.
 - Simuler divers scénarios d'utilisation réelle pour évaluer la robustesse de la solution.

- Tester les API REST pour vérifier leur conformité aux spécifications et assurer une communication fluide entre le frontend et le backend.
- Vérifier que les modifications apportées aux données côté backend sont correctement reflétées dans les tableaux et interfaces du frontend.
- Assurer que les données apparaissent correctement dans les tableaux et que les modifications de données se reflètent en temps réel.

1.1.8 Outils de collaboration



Figure 1.4: Logo de Skype [2]

Skype est un logiciel qui permet aux utilisateurs de passer des appels téléphoniques ou vidéo via Internet, ainsi que le partage d'écran. Les appels d'utilisateur à utilisateur sont gratuits, tandis que les appels vers les lignes fixes et les téléphones mobiles sont payants. Il existe des fonctionnalités supplémentaires telles que la messagerie instantanée, le transfert de fichiers et la visioconférence. Il a été très utile pour collaborer avec les autres stagiaires et le chef de projet. [2]

Conclusion

Dans ce chapitre, nous avons présenté le contexte global de notre projet de stage en décrivant l'entreprise qui nous a accueillis, ses objectifs, missions, ainsi que les méthodologies de développement qu'elle utilise. Nous avons également détaillé la portée et la planification du projet, ainsi que les collaborations et les outils de communication nécessaires à sa réussite.

Chapter 2

Concepts et technologies

Introduction

Notre projet se compose principalement de deux parties principales : le module de gestion des notes de frais et le module de collaboration et de communication. Ce sont les deux concepts fondamentaux que nous allons explorer. Nous détaillerons l'environnement de travail en donnant un aperçu des outils et technologies qui ont permis la réalisation de ces deux parties.

2.1 Concepts fondamentaux

2.1.1 Développement Logiciel (Software Engineering)

2.1.1.1 Définition et Importance

Le développement logiciel, également connu sous le nom d'ingénierie logicielle, est un domaine de l'informatique qui concerne la conception, la création, le déploiement et la maintenance de logiciels. Cette discipline utilise des principes d'ingénierie et des méthodologies éprouvées pour garantir que les logiciels produits sont de haute qualité, fiables et répondent aux besoins des utilisateurs.

L'importance de l'ingénierie logicielle réside dans sa capacité à structurer et à organiser le processus de développement de logiciels, ce qui permet de gérer efficacement la complexité des projets logiciels et de minimiser les risques d'erreurs et de dysfonctionnements.

2.1.1.2 Les Étapes du Développement Logiciel

Le développement logiciel se déroule en plusieurs étapes essentielles. Tout d'abord, l'**analyse des besoins** consiste à identifier les besoins des utilisateurs et des parties prenantes, ainsi qu'à documenter les exigences fonctionnelles et non fonctionnelles. Ensuite, la **conception** implique l'élaboration de l'architecture logicielle et des modèles de conception, ainsi que la définition des interfaces utilisateur et des interactions système. La troisième étape, l'**implémentation**, se concentre sur l'écriture du code source en utilisant des langages de programmation appropriés et l'intégration des différentes composantes du logiciel. Une fois le logiciel développé, il est soumis à des **tests** rigoureux pour vérifier et valider son bon fonctionnement, en utilisant des tests unitaires, des tests d'intégration, des tests système et des tests de régression. Le **déploiement** suit les tests, avec la distribution du logiciel aux utilisateurs finaux et sa configuration sur les systèmes cibles. Enfin, la **maintenance** continue après le déploiement, avec la correction des bugs et des erreurs, ainsi que la mise à jour et l'amélioration continue du logiciel en réponse aux retours des utilisateurs et aux évolutions technologiques.

2.1.2 Méthodologie de Développement : SCRUM

- La méthodologie agile est une façon de gérer un projet en le décomposant en plusieurs phases. En d'autres termes, il s'agit d'une méthode ou d'une approche de gestion de projet qui fonctionne par itérations et incréments, capable de prendre en compte les besoins initiaux du projet et ceux liés à l'évolution. La méthode Agile est basée sur un cycle de développement qui place le client au centre. Le client est impliqué dans la réalisation du début à la fin du projet. Grâce à la méthode agile, le client obtient une meilleure visibilité de la gestion du travail qu'avec une méthode classique. L'implication du client dans le processus permet à l'équipe d'obtenir un feedback régulier afin d'appliquer directement les changements nécessaires.

Expansion de la Méthodologie Agile

Cette méthode a d'abord été conçue pour le développement de logiciels puis elle s'est généralisée pour couvrir d'autres secteurs tels que le marketing. Elle vise à accélérer la réalisation d'un projet. De plus, elle assure la réalisation d'un produit fonctionnel pendant toute la durée de sa création. Le principe de base est de proposer une version minimale du produit puis d'intégrer des fonctionnalités supplémentaires à cette base, par processus itératif. Le processus itératif regroupe une séquence d'instructions à répéter autant de fois que nécessaire, en fonction du besoin.

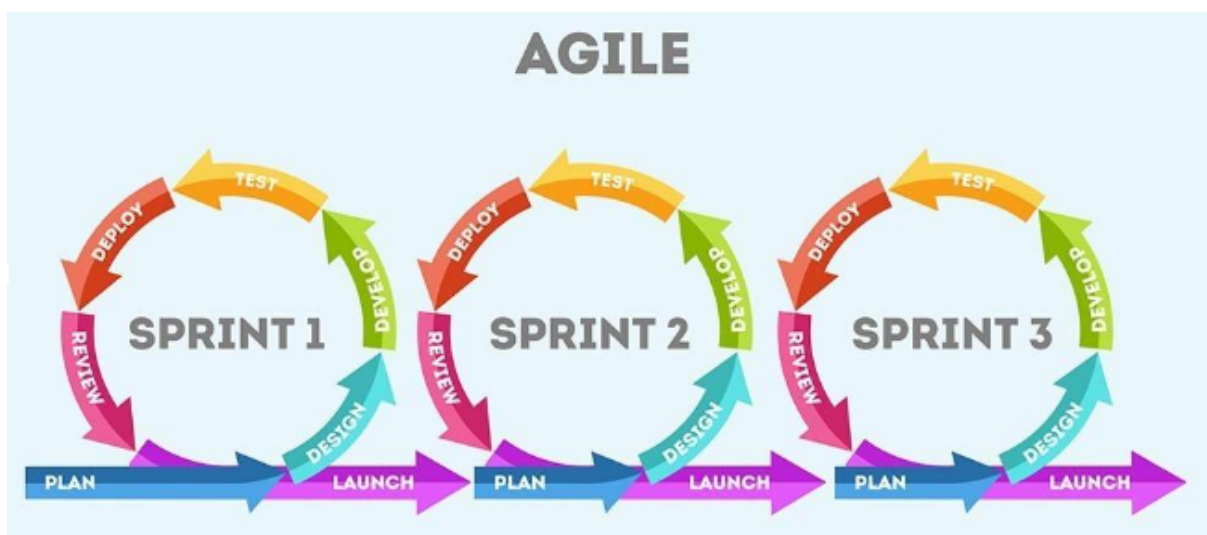


Figure 2.1: Schéma explicatif de la méthode Agile.

Les Quatre Grands Principes du Manifeste Agile

La méthode Agile nommée Manifeste Agile est basée sur quatre grands principes :

- **COLLABORATION** : La communication et la cohésion de l'équipe passent avant les outils et les processus.
- **ÉQUIPE** : Le privilège de la relation équipe/client est mis en avant plutôt que la négociation contractuelle.
- **PRODUIT** : Préférer un produit bien construit à une documentation détaillée.
- **ACCEPTATION** : Le choix d'accepter le changement et la flexibilité plutôt qu'un plan rigide.

En effet, le changement de contexte et les modifications interviennent dans le processus suite aux demandes du client, ce qui fera évoluer le projet plus rapidement.

Conclusion

Cette méthodologie permet d'améliorer la productivité globale et d'accélérer la réalisation du projet. En conclusion, cette méthodologie améliorera de manière significative l'efficacité de conception et de réalisation de notre projet ainsi que l'augmentation de la productivité, de la qualité de service et de la production. Cela permet également de réduire le temps de livraison du service, de gagner ainsi la confiance et la réputation sur le marché, et enfin d'augmenter le chiffre d'affaires et les bénéfices.

2.1.3 La Méthodologie Agile en Pratique

Dans le cadre du travail sur le projet, j'ai adopté une approche itérative pour organiser mon travail de manière efficace. Chaque itération, d'une durée d'une à deux semaines, est soigneusement planifiée pour inclure la définition des objectifs, le développement des fonctionnalités, les tests et les ajustements nécessaires. Je divise mes tâches en éléments plus petits et plus gérables, en me concentrant sur des fonctionnalités spécifiques à livrer à la fin de chaque itération.

- Pendant le développement, j’ai appliqué des pratiques de développement incrémental en ajoutant des fonctionnalités de base en premier lieu, puis en enrichissant le projet avec des fonctionnalités plus avancées au fil du temps.
- J’ai également veillé à tester chaque fonctionnalité développée pour garantir sa qualité et sa conformité aux exigences du projet.
- La méthode Agile qui correspond le mieux à ce travail avec une forte adaptabilité aux changements et une capacité à modifier les plans en cours de route est probablement l’Extreme Programming (XP). XP encourage une approche flexible du développement, avec un focus sur les retours fréquents, l’adaptation aux changements et la collaboration étroite avec le client. Elle met également l’accent sur la qualité du code, la simplicité et la communication efficace, ce qui en fait un choix pertinent pour les projets en solo où les besoins et les conditions peuvent évoluer rapidement.
- Revues de code permanentes.
- Développement piloté par les tests.
- Conception au cours du projet (Refactoring).
- Privilégier systématiquement la méthode la plus simple.
- L’intégration des modifications se fait quotidiennement, voire plusieurs fois par jour.

2.1.4 Validation de la Solution

2.1.4.1 Validation de la Solution par Tests Manuels

La validation de notre solution a été effectuée principalement à travers des tests manuels, une approche qui s’est révélée efficace pour assurer la qualité et la fonctionnalité de notre application de gestion des notes de frais. Bien que nous n’ayons pas utilisé de tests automatisés comme les tests unitaires ou d’intégration, nos tests manuels ont été minutieux et rigoureux.

2.1.4.2 Simulation de Scénarios d’Utilisation Réelle

Pour évaluer la robustesse de notre solution, nous avons simulé divers scénarios d’utilisation réelle. Cette approche proactive nous a permis de détecter et de corriger les éventuels défauts avant le déploiement final.

2.1.4.3 Tests des API REST

Parallèlement, nous avons testé les API REST pour vérifier leur conformité aux spécifications. Les résultats de ces tests ont été positifs, confirmant que les interactions entre le frontend et le backend se déroulent sans accroc. Chaque requête HTTP a été validée pour s'assurer que les données sont transmises correctement et que les réponses respectent les attentes en termes de structure et de contenu.

2.1.4.4 Intégration Backend-Frontend

Nous avons confirmé que les modifications apportées aux données côté backend étaient correctement reflétées dans les tableaux et interfaces du frontend. Ce processus a été crucial pour assurer une expérience utilisateur fluide et en temps réel.

2.1.5 Résultats

Même sans recourir à des tests automatisés tels que les tests unitaires ou d'intégration, notre méthode de tests manuels s'est révélée très efficace. Elle a permis de livrer une application de gestion de notes de frais fonctionnelle, fiable et conforme aux attentes. Les résultats obtenus démontrent notre engagement à fournir une solution de haute qualité et garantissant une expérience utilisateur optimale.

2.1.6 Cycles de développement

Le cycle de développement de mon projet se déroule en plusieurs étapes itératives, suivant les principes de l'Extreme Programming (XP) pour un travail en solo :

1. **Planification de l'itération** : Je définis les objectifs et les fonctionnalités à développer lors de chaque itération, en tenant compte des retours précédents et des besoins actuels du projet.
2. **Développement des fonctionnalités** : Je commence par développer les fonctionnalités de base, en suivant les bonnes pratiques de développement et en m'assurant de leur qualité grâce à des tests unitaires.

3. **Tests** : Une fois les fonctionnalités développées, je les teste pour vérifier leur bon fonctionnement .
4. **Revue de code** : Je réalise des revues de code régulières pour garantir la qualité du code et détecter les éventuels bugs ou problèmes de conception.
5. **Refactoring** : Si nécessaire, je procède à des refactorings pour améliorer la qualité du code et faciliter l'ajout de nouvelles fonctionnalités à l'avenir.
6. **Livraison et rétroaction** : À la fin de chaque itération, je livre les fonctionnalités développées au client ou à l'utilisateur final, et je recueille leur rétroaction pour orienter les prochaines itérations.

Ce cycle de développement itératif me permet d'avancer efficacement dans mon projet, en m'adaptant aux changements et en assurant la qualité du code et des fonctionnalités livrées.

2.1.7 Les Outils Utilisés en Ingénierie Logicielle

- **Systèmes de Contrôle de Version** : Git, GitLab, pour gérer les versions du code source et faciliter la collaboration entre développeurs.
- **Environnements de Développement Intégré (IDE)** : Visual Studio Code, pour écrire, tester et déboguer le code.
- **Frameworks et Bibliothèques** : ReactJS pour le développement d'interfaces utilisateur, 4D pour la gestion de bases de données.

2.1.7.1 Conclusion

L'ingénierie logicielle est essentielle pour développer des applications et des systèmes logiciels efficaces et fiables. En appliquant des méthodologies structurées et des outils appropriés, les développeurs peuvent s'assurer que le logiciel répond aux besoins des utilisateurs, est de haute qualité et est maintenable à long terme. Dans le contexte de ton projet de gestion des notes de frais, l'ingénierie logicielle permet de créer une application robuste qui simplifie le processus de gestion des dépenses et améliore l'efficacité opérationnelle de l'entreprise.

2.1.8 Conception de Bases de Données

2.1.8.1 Définition et Importance

La conception de bases de données est le processus de création d'un modèle de données détaillé pour une base de données. Ce modèle inclut toutes les structures nécessaires à la gestion et à l'accès aux données. Une conception de base de données bien pensée est essentielle pour garantir l'intégrité, la performance, et la facilité de maintenance des données.

2.1.8.2 Modélisation des Données

La modélisation des données est une étape clé de la conception de bases de données. Elle permet de représenter visuellement la structure des données et les relations entre les différentes entités.

Parmi les principaux modèles de modélisation, on trouve :

- **Modèle Relationnel** : Ce modèle utilise des tables pour représenter les données et les relations entre elles. Chaque table est composée de lignes (enregistrements) et de colonnes (attributs).
- **Schémas ERD (Entity-Relationship Diagram)** : Les diagrammes ERD sont des outils graphiques utilisés pour représenter les entités, leurs relations, et les attributs de chaque entité dans un système d'information.

2.1.8.3 Normalisation et Optimisation des Bases de Données

La normalisation est un processus qui organise les données dans une base de données de manière à minimiser la redondance et à éviter les anomalies d'insertion, de mise à jour et de suppression.

Les principales formes normales sont :

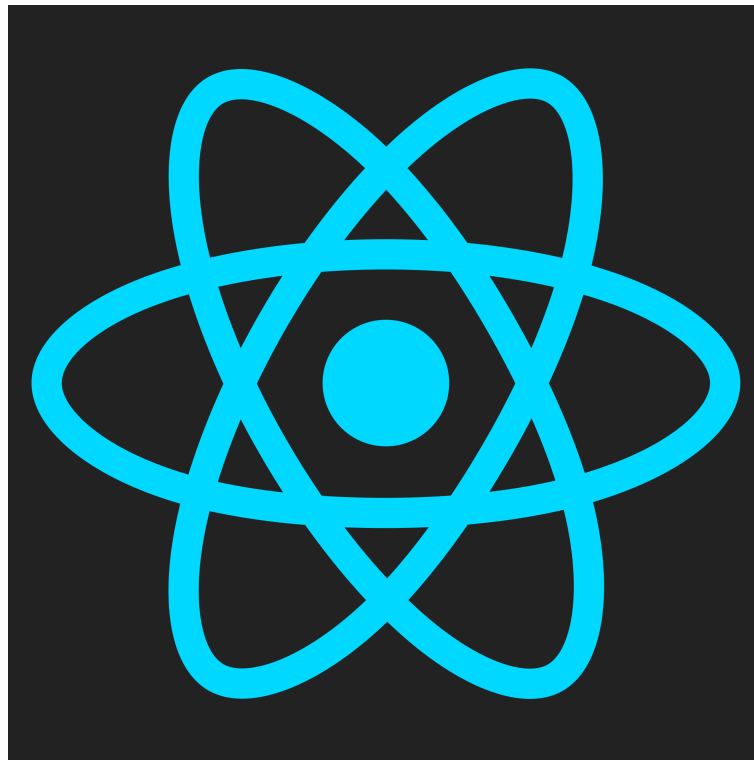
- **Première Forme Normale (1NF)** : Toutes les valeurs des colonnes doivent être atomiques (indivisibles).
- **Deuxième Forme Normale (2NF)** : La table doit être en 1NF et tous les attributs non-clés doivent dépendre entièrement de la clé primaire.
- **Troisième Forme Normale (3NF)** : La table doit être en 2NF et tous les attributs non-clés doivent être indépendants les uns des autres.

L'optimisation des bases de données inclut également l'indexation des colonnes fréquemment interrogées, la partition des tables volumineuses, et l'utilisation de vues matérialisées pour améliorer les performances des requêtes complexes.

La performance de la base de données peut être influencée par la conception des requêtes, la configuration matérielle, et les paramètres du système de gestion de bases de données (SGBD). Une bonne conception de la base de données, associée à une indexation appropriée et à des pratiques d'optimisation, assure une gestion efficace et performante des données.

2.2 Les technologies

2.2.1 ReactJS



ReactJS, ou simplement React, est une bibliothèque JavaScript open-source maintenue par Facebook pour la construction d'interfaces utilisateur ou de composants UI. Elle est couramment utilisée pour la création d'applications web monopages et d'applications mobiles. React permet aux développeurs de créer de grandes applications web qui peuvent modifier les données sans recharger la page. Elle est connue pour son efficacité et sa flexibilité, ainsi que pour sa capacité à gérer facilement des interfaces complexes et dynamiques.

Les principales fonctionnalités de React incluent:

- **Basé sur les composants** : React utilise une architecture basée sur les composants où les interfaces utilisateur sont composées de composants indépendants et réutilisables, ce qui facilite la gestion et la maintenance des applications à grande échelle.
- **Virtual DOM** : React utilise un DOM virtuel (Document Object Model) pour améliorer les performances en minimisant le nombre d'opérations DOM nécessaires pour les mises à jour. Cela permet de créer des applications web rapides et réactives.
- **JSX** : JSX est une extension de syntaxe pour JavaScript qui permet aux développeurs d'écrire du code similaire à HTML dans leurs fichiers JavaScript. Cela rend le code plus lisible et permet une intégration plus facile de HTML et JavaScript.
- **Flux de données unidirectionnel** : React suit un flux de données unidirectionnel, où les données circulent dans une seule direction, des composants parent aux composants enfants. Cela aide à maintenir la cohérence des données dans toute l'application.
- **React Native** : React peut également être utilisé pour créer des applications mobiles natives en utilisant React Native, ce qui permet aux développeurs d'utiliser le même code source pour les applications web et mobiles.

Dans l'ensemble, React est largement utilisé dans la communauté de développement web pour ses performances, sa flexibilité et sa facilité d'utilisation dans la construction d'applications web modernes. [3]

2.2.2 4D



Figure 2.2: Logo du 4D [1]

4D est un logiciel de développement d'applications qui permet de créer des applications pour les ordinateurs Mac et Windows, ainsi que pour le web et les appareils mobiles. Il a été initialement développé par la société française 4D SAS et est maintenant distribué à l'international.

Le langage de programmation 4D est basé sur un environnement de développement intégré (EDI) qui fournit un ensemble d'outils pour faciliter la création d'applications. L'EDI de 4D est convivial et offre une interface graphique intuitive pour créer des interfaces utilisateur, concevoir des bases de données et développer des scripts.

Les applications créées avec 4D peuvent être utilisées pour une variété de tâches, telles que la gestion de bases de données, la gestion de projets, la facturation, le suivi des ventes, et bien plus encore. Le logiciel est également adapté pour créer des applications métier personnalisées pour les entreprises.

4D offre une multitude de fonctionnalités pour faciliter la création d'applications, notamment:

- Fonctionnalités de base de données : 4D intègre un puissant système de gestion de base de données relationnelle (SGBDR) qui permet de stocker et de gérer les données de manière efficace. Les développeurs peuvent utiliser le langage de requête SQL (Structured Query Language) pour interroger les données, ainsi que des outils de modélisation pour concevoir les schémas de base de données.
- Fonctionnalités de conception d'interface utilisateur : 4D propose une variété d'outils pour créer des interfaces utilisateur attrayantes et conviviales, comprenant des boutons, des menus déroulants, des champs de saisie, et des listes déroulantes. Les développeurs peuvent également utiliser des thèmes personnalisés pour personnaliser l'apparence de leurs applications.
- Fonctionnalités de script de développement : 4D utilise un langage de programmation propriétaire basé sur les langages BASIC et C++, appelé langage 4D. Ce langage offre une variété de fonctionnalités pour les développeurs, notamment des variables, des fonctions, des boucles et des instructions de contrôle de flux.
- Fonctionnalités de sécurité : 4D offre des fonctionnalités de sécurité avancées pour protéger les données des utilisateurs, comprenant des options pour chiffrer les données et les communications réseau. [1]

2.2.3 4D Client



Figure 2.3: Logo du 4D Client [10]

4D Client est un composant du logiciel de développement d'applications 4D qui permet aux utilisateurs d'accéder aux applications 4D à distance via un réseau local ou Internet. Il est disponible pour les ordinateurs Mac et Windows, ainsi que pour les appareils mobiles.

Lorsque les utilisateurs se connectent à une application 4D via 4D Client, ils peuvent accéder à toutes les fonctionnalités de l'application, y compris les bases de données, les interfaces utilisateur et les scripts. Les développeurs peuvent personnaliser l'expérience utilisateur en fonction des besoins de l'application, en créant des formulaires, des rapports et des tableaux de bord sur mesure.

4D Client offre également des fonctionnalités de sécurité avancées pour protéger les données des utilisateurs. Les développeurs peuvent utiliser des certificats SSL pour chiffrer les communications entre le client et le serveur, ainsi que des identifiants de connexion pour contrôler l'accès des utilisateurs à des fonctionnalités spécifiques de l'application.

En résumé, 4D Client est un composant clé du logiciel de développement d'applications 4D, permettant aux utilisateurs d'accéder aux applications 4D à distance via un réseau local ou Internet. Il fournit toutes les fonctionnalités de l'application, ainsi que des fonctionnalités de sécurité avancées pour protéger les données des utilisateurs. [10]

2.2.4 4D serveur



Figure 2.4: Logo du 4D Server [1]

4D Server est un composant clé du logiciel de développement d'applications 4D qui permet aux développeurs de déployer leurs applications sur un serveur et de les rendre accessibles aux utilisateurs à distance via un réseau local ou Internet. Il est disponible pour les ordinateurs Mac et Windows, ainsi que pour les serveurs Linux.

4D Server offre des fonctionnalités avancées de gestion de bases de données, notamment la capacité de gérer des milliers de connexions simultanées, d'optimiser les performances des requêtes et de gérer les transactions de base de données. Il prend également en charge les sauvegardes automatiques et les restaurations de bases de données, ainsi que les mises à jour en temps réel du schéma de la base de données.

En plus de la gestion des bases de données, 4D Server offre des fonctionnalités de sécurité avancées pour protéger les données des utilisateurs. Les développeurs peuvent utiliser des certificats SSL pour chiffrer les communications entre le serveur et les clients, ainsi que des identifiants de connexion pour contrôler l'accès des utilisateurs à des fonctionnalités spécifiques de l'application.

En résumé, 4D Server est un composant clé du logiciel de développement d'applications 4D, permettant aux développeurs de déployer leurs applications sur un serveur et de les rendre accessibles aux utilisateurs à distance via un réseau local ou Internet. Il offre des fonctionnalités avancées de gestion de bases de données et de sécurité pour protéger les données des utilisateurs. [1]

2.2.5 Qu'est ce qu'un composant 4D ?

Dans le cadre du logiciel de développement d'applications 4D, un composant 4D est une unité logicielle réutilisable qui peut être intégrée dans diverses applications. Ces composants sont conçus pour encapsuler des fonctionnalités spécifiques, telles que des interfaces utilisateur, des algorithmes de traitement des données ou des routines de communication avec des services externes.

Les composants 4D sont développés en utilisant l'environnement de développement 4D et sont ensuite stockés dans une bibliothèque de composants. Les développeurs peuvent facilement intégrer ces composants dans leurs projets, ce qui permet de gagner du temps et d'améliorer la productivité en évitant la duplication de code et en éliminant la nécessité de développer des fonctionnalités à partir de zéro.

En outre, les composants 4D sont souvent distribués sous forme de modules complémentaires, qui peuvent être achetés et installés dans des applications existantes pour ajouter des fonctionnalités supplémentaires. Ces modules, développés par 4D ou par des tiers, couvrent une large gamme de fonctionnalités, y compris la gestion de projets, la comptabilité, le traitement des images et la communication avec des services externes.

2.2.6 Visual Studio Code

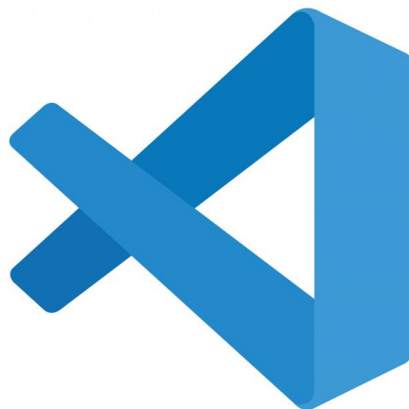


Figure 2.5: Logo du Visual Studio Code [9]

Visual Studio Code, souvent abrégé en VS Code, est un éditeur de code source gratuit et open source développé par Microsoft. Conçu pour répondre aux besoins des développeurs, qu'ils

travaillent sur des projets web, cloud, mobiles ou autres, VS Code offre un environnement de développement polyvalent et puissant. Grâce à sa conception légère et modulaire, VS Code combine une interface utilisateur épurée avec des fonctionnalités avancées, ce qui en fait un choix populaire parmi les développeurs du monde entier. L'éditeur prend en charge une large gamme de langages de programmation et propose des extensions riches qui permettent d'ajouter des fonctionnalités supplémentaires en fonction des besoins du projet. Parmi ses caractéristiques notables, on trouve la coloration syntaxique, l'achèvement automatique du code, le débogage intégré, la gestion de versions via des extensions Git, la prise en charge de l'intégration continue et bien plus encore. VS Code se distingue également par sa communauté active qui développe et maintient des extensions pour divers frameworks, langages et outils de développement. Cette flexibilité et cette extensibilité font de Visual Studio Code un choix précieux pour les développeurs souhaitant personnaliser leur environnement de développement en fonction de leurs besoins spécifiques. [9]

2.2.7 Axios



Figure 2.6: Logo du Axios [4]

Axios est une bibliothèque JavaScript utilisée pour effectuer des requêtes HTTP depuis le navigateur et Node.js. Elle simplifie l'envoi de requêtes asynchrones vers des API RESTful, facilitant ainsi la récupération et l'envoi de données.

Fonctionnalités principales :

- **Requêtes HTTP simples** : Axios simplifie la syntaxe pour effectuer des requêtes HTTP telles que GET, POST, PUT, DELETE, etc.
- **Gestion des promesses** : Axios utilise les promesses pour gérer les requêtes asynchrones, permettant ainsi une meilleure gestion du flux de contrôle avec `.then()` et `.catch()`.

- **Intercepteurs** : Axios offre des intercepteurs de requêtes et de réponses, permettant de transformer les requêtes ou réponses avant qu'elles ne soient gérées par le `then` ou le `catch`.
- **Annulation des requêtes** : Il est possible d'annuler des requêtes en cours, ce qui est utile pour gérer les scénarios où les requêtes ne sont plus nécessaires.
- **Support de JSON** : Axios gère automatiquement la transformation des données en JSON pour les requêtes et les réponses.

Utilisation courante avec React :

Dans un projet React, Axios est couramment utilisé pour effectuer des appels API afin de récupérer ou envoyer des données. Par exemple, pour récupérer des données depuis un serveur. [4]

2.2.8 Chart JS

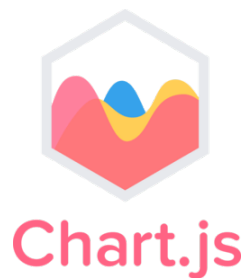


Figure 2.7: Logo du ChartJs [5]

Chart.js est une bibliothèque JavaScript open-source qui permet aux développeurs de créer des graphiques et des visualisations de données de manière simple et flexible. Elle est utilisée pour créer des graphiques interactifs et réactifs qui peuvent être intégrés dans des applications web. [5]

Les principales fonctionnalités de Chart.js incluent :

- **Simplicité d'utilisation** : Chart.js est facile à utiliser et permet de créer des graphiques en quelques lignes de code.
- **Types de graphiques variés** : La bibliothèque prend en charge différents types de graphiques, y compris les graphiques en ligne, les graphiques à barres, les graphiques à secteurs, les graphiques radar, les graphiques en bulles, etc.
- **Personnalisation** : Les graphiques créés avec Chart.js sont hautement personnalisables, permettant aux développeurs de modifier les couleurs, les tailles, les polices, et d'ajouter des animations.

- **Réactivité** : Les graphiques sont réactifs et s'adaptent automatiquement aux différentes tailles d'écran et aux appareils.
- **Support des plugins** : Chart.js permet d'ajouter des plugins pour étendre ses fonctionnalités et répondre à des besoins spécifiques.

2.2.9 TailwindCss



Tailwind CSS

Figure 2.8: Logo du TailwindCss [18]

Tailwind CSS offre un paradigme de conception centré sur les classes utilitaires, ce qui permet une approche modulaire et réutilisable dans la création d'interfaces utilisateur. Grâce à sa vaste bibliothèque de classes prédéfinies, Tailwind simplifie le processus de stylisation en offrant une flexibilité maximale et en éliminant le besoin de CSS personnalisé. [18]

2.2.10 Systèmes de contrôle de version

2.2.10.1 Git



Figure 2.9: Logo du Git [6]

Git est un système de contrôle de versions distribué qui permet aux développeurs de travailler efficacement en équipe sur un projet de développement de logiciel. Il stocke le code source du projet dans un référentiel Git, qui peut être local ou distant. Les développeurs travaillent sur des branches de code séparées et effectuent des modifications de code en parallèle. Les branches peuvent être fusionnées pour intégrer les modifications de code et les tests de validation. Dans le contexte DevOps, Git est utilisé pour automatiser le flux de travail de développement de logiciels. Les développeurs créent des branches de code pour travailler sur des fonctionnalités ou des correctifs spécifiques, et utilisent des outils d'intégration continue pour effectuer des tests de validation et de qualité automatiquement. Les branches de code sont revues par d'autres membres de l'équipe avant d'être fusionnées dans la branche principale, qui contient le code de production. L'utilisation de Git dans le contexte DevOps permet de faciliter la collaboration entre les développeurs, de garantir la qualité du code et de réduire les erreurs et les conflits lors de la fusion de modifications de code. Elle permet également de suivre l'historique des modifications de code et de restaurer des versions antérieures en cas de besoin. Techniquement, Git utilise un système de fichiers distribué pour stocker les versions du code source. Il permet aux développeurs de travailler en mode déconnecté et de synchroniser leur travail avec le référentiel Git lorsqu'ils ont une connexion Internet. Les modifications de code sont enregistrées sous forme de "commits" qui sont accompagnés de messages décrivant les modifications

apportées. Les branches de code peuvent être créées et fusionnées en utilisant des commandes Git, qui sont disponibles en ligne de commande ou via des interfaces utilisateur graphiques. Les développeurs peuvent également utiliser des outils de gestion de versions Git tels que GitHub, GitLab ou Bitbucket pour gérer les référentiels Git et collaborer avec d'autres membres de l'équipe. L'un des modèles de flux de travail les plus populaires basés sur Git est le Git Flow. Le Git Flow est un ensemble de directives pour la gestion des branches dans un projet Git. Dans notre projet, l'organisation des branches semble suivre le Git Flow, qui est largement utilisé dans les projets DevOps. Le Git Flow fournit une structure et des directives claires pour la gestion des branches de code source dans un projet de développement de logiciel. [6]

2.2.10.2 GitLab



Figure 2.10: Logo du Gitlab [6]

GitLab est un outil de cycle de vie DevOps basé sur le web qui fournit un gestionnaire de référentiels Git, de suivi des problèmes, et de pipelines CI/CD (intégration continue et déploiement continu), sous une licence open-source. Il s'agit d'une plateforme DevOps complète, livrée en tant qu'application unique, couvrant toutes les étapes du cycle de vie DevOps, de la planification du projet à la gestion du code source, en passant par la surveillance et la sécurité. Voici quelques fonctionnalités clés de GitLab, détaillées en français :

– **Gestion de version (Version Control) :**

- * GitLab utilise Git pour la gestion des versions de code source, permettant aux équipes de collaborer sur le code et de suivre les modifications au fil du temps. Chaque modification est enregistrée, et il est possible de revenir à des versions antérieures si nécessaire.

– **Intégration Continue et Déploiement Continu (CI/CD) :**

- * GitLab inclut des capacités d'intégration continue (CI) et de déploiement continu (CD), permettant de tester automatiquement et de déployer le code. Les pipelines CI/CD automatisent les tests et les déploiements, assurant que les nouvelles modifications sont validées avant d'être intégrées.

– **Gestion de projet (Project Management) :**

- * GitLab offre des outils pour le suivi des problèmes, la planification des projets et la collaboration. Il inclut des fonctionnalités telles que les tableaux de bord, les jalons, et les listes de tâches, facilitant la gestion des projets de développement logiciel.

– **Revue de code (Code Review) :**

- * GitLab supporte les processus de revue de code avec les *merge requests* (demandes de fusion), permettant aux membres de l'équipe de réviser et de discuter des modifications avant leur intégration dans la base de code principale. Cela aide à maintenir la qualité du code.

– **Sécurité (Security) :**

- * GitLab propose des fonctionnalités de sécurité intégrées telles que l'analyse des dépendances, l'analyse des conteneurs, et la détection des secrets. Ces outils aident à identifier et corriger les vulnérabilités dans le code, assurant ainsi une meilleure qualité et sécurité du code.

– **Surveillance (Monitoring) :**

- * GitLab inclut des outils de surveillance et de gestion des performances pour suivre la santé et les performances des applications. Cela permet de détecter rapidement les problèmes et d'assurer un fonctionnement optimal des applications.

– **Wiki et Documentation :**

- * GitLab dispose d'une fonctionnalité de wiki intégrée, permettant aux équipes de créer et de maintenir une documentation à jour, accessible à tous les membres du projet.

– **Déploiement Multicloud :**

- * GitLab permet le déploiement des applications sur plusieurs plateformes de cloud computing, offrant une flexibilité et une scalabilité accrues pour les environnements de production.

– **Automatisation des Processus :**

- * Avec GitLab, vous pouvez automatiser de nombreux processus de développement, de la création des environnements de développement aux déploiements de production, en passant par les tests de sécurité et de performance.

GitLab est disponible en versions auto-hébergées et SaaS, offrant aux équipes la flexibilité de l'héberger sur leur propre infrastructure ou d'utiliser les services hébergés par GitLab. [6]

2.3 Benchmarking des outils

Pour satisfaire les exigences changeantes du développement web, il est essentiel de rester constamment à jour sur les avancées technologiques et de sélectionner les technologies les plus adaptées. Cette sélection est fondée sur une analyse approfondie qui prend en compte les diverses options disponibles

2.3.1 Frontend

Afin de cerner notre étude, nous allons commencer par une consultation des plateformes de recommandation d'outils. Nous avons effectué une recherche sur divers sites de comparaison des frameworks et nous avons retenu les deux suivants :

Stack Overflow Annual Developer Survey 2023

C'est une enquête annuelle qui fournit des informations sur la communauté des développeurs. Depuis treize ans, cette enquête est une source essentielle pour les analystes, les leaders informatiques et les développeurs eux-mêmes, leur permettant de se tenir au courant des expériences des développeurs, des technologies émergentes et des tendances du secteur.



Figure 2.11: Stack Overflow Annual Developer Survey 2023 [11]

En ce qui concerne le frontend React, et Angular sont les technologies les plus adoptées par la communauté des développeurs dans le monde. [11]

State Of Dev In Morocco 2023

C'est un rapport annuel qui présente des données et des analyses sur la communauté des développeurs au Maroc. En décembre 2023, 1764 développeurs marocains ont participé à cette enquête, fournissant des informations précieuses sur divers aspects de leur profession et de leur expérience.

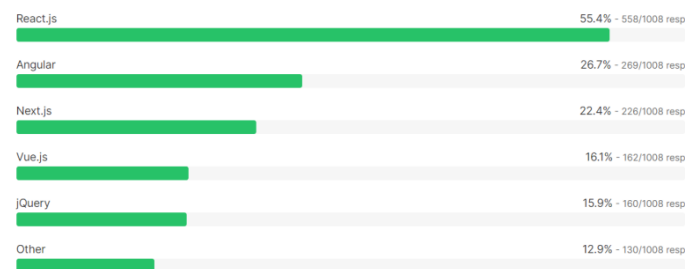


Figure 2.12: State Of Dev In Morocco 2023 [12]

React a également dominé en popularité, avec un pourcentage significatif de développeurs locaux l'adoptant pour leurs projets. Angular suit, avec des communautés actives et un intérêt croissant. [12]

React vs Angular:

React, développé par Facebook, est extrêmement populaire grâce à sa simplicité et sa flexibilité. Il dispose d'une grande communauté, d'une documentation étendue et de nombreuses bibliothèques tierces. Angular, créé par Google, possède également une communauté importante et un soutien robuste, mais sa complexité peut être un obstacle pour les débutants. Les deux frameworks bénéficient d'un solide soutien corporatif, garantissant un support à long terme et un développement continu. La courbe d'apprentissage de React est relativement faible, surtout pour les développeurs familiers avec JavaScript. Il se concentre sur la couche de vue, permettant aux développeurs de choisir leurs outils préférés pour le routage et la gestion d'état. Angular, en revanche, est un framework MVC complet, offrant plus de fonctionnalités prêtes à l'emploi mais avec une courbe d'apprentissage plus raide en raison de sa nature exhaustive. La mise en œuvre du DOM virtuel de React conduit à des mises à jour efficaces et à de hautes performances, le rendant adapté aux applications nécessitant des mises à jour fréquentes. Son architecture basée sur les composants favorise la réutilisabilité et la flexibilité. Le data binding bidirectionnel d'Angular simplifie le développement en synchronisant automatiquement les données entre le modèle et la vue, mais peut entraîner des problèmes de performance dans les grandes applications. Cependant, les outils intégrés et le CLI d'Angular rationalisent les processus de développement et augmentent la productivité. L'écosystème de React est vaste, avec une multitude de bibliothèques et d'outils disponibles pour étendre ses fonctionnalités. Son intégration avec divers outils de développement et le support pour le développement mobile via React Native sont des avantages significatifs. Angular offre une solution plus intégrée avec son ensemble riche de fonctionnalités, y compris l'injection de dépendances, le client HTTP et la validation des formulaires. Le CLI d'Angular simplifie la configuration des projets et les tâches de développement, offrant une expérience de développement plus cohérente.

Comparaison entre React et Angular		
Critères	React	Angular
Apprentissage	Simple	Complexe et longue
Péformance	Performant	Performant
Taille	Petite	Grande
Strucuture	Fléxible	Structure strice
Écosystème	Vaste	Vaste
Norme d'entreprise	Adopté	Non adopté

Table 2.1: Comparaison entre React [3] et Angular.

Dans l'évaluation comparative entre React et Angular, il est indéniable que les deux frameworks offrent des fonctionnalités solides pour notre projet. Cependant, React se démarque avec sa simplicité d'apprentissage, sa performance optimisée grâce à son DOM virtuel, son écosystème étendu comprenant de nombreuses bibliothèques et outils, ainsi que son adoption répandue par les entreprises. Alors que Angular propose également des avantages tels qu'une structure rigoureuse et des fonctionnalités prêtes à l'emploi, sa courbe d'apprentissage plus raide et son adoption moins répandue par les entreprises pourraient être des obstacles pour certains développeurs. En conséquence, bien que les deux frameworks soient des choix valables, React semble avoir un léger avantage, notamment pour les projets où la simplicité, les performances et le soutien de l'entreprise sont des facteurs critiques.

2.3.2 Backend

Etant donné que "4D Maroc" utilise le langage propriétaire 4D pour le développement de ses applications, voici une étude menée en utilisant l'indice TIOBE [13] qui est un indicateur de la popularité des langages de programmation. Mis à jour mensuellement, cet indice se base sur divers critères, incluant le nombre de développeurs qualifiés dans le monde, les cours disponibles et les fournisseurs tiers. Les évaluations sont calculées en utilisant des données provenant de sites web populaires tels que Google, Amazon, Wikipedia, Bing, ainsi que plus de 20 autres sources.


Jun 2024	Jun 2023	Change	Programming Language		Ratings	Change
1	1			Python	15.39%	+2.93%
2	3	▲		C++	10.03%	-1.33%
3	2	▼		C	9.23%	-3.14%
4	4			Java	8.40%	-2.88%
5	5			C#	6.65%	-0.06%
6	7	▲		JavaScript	3.32%	+0.51%
7	14	▲		Go	1.93%	+0.93%
8	9	▲		SQL	1.75%	+0.28%

Table 2.2: Classement des langages de programmation selon l'indice TIOBE. [13]

] D'après la figure ci-dessus, Python, et C++ sont les mieux classés selon l'étude effectuée, c'est pour cela que nous allons réaliser une comparaison générale entre les deux premiers langages et le langage 4D. Selon le tableau 3.3, le langage 4D se révèle être une option particulièrement avantageuse pour les projets nécessitant une forte interaction avec les bases de données. Sa syntaxe simplifiée, sa gestion automatique de la mémoire, et son intégration native des bases de données en font un choix privilégié pour les développeurs cherchant à allier performance et simplicité. En comparaison, bien que Python soit très flexible et possède une large gamme de bibliothèques, et que C offre des performances optimales, 4D propose un équilibre unique qui favorise la productivité et la robustesse dans le développement d'applications de gestion de données.

Comparaison entre 4D, Python et C++			
Critères	4D	Python	C++
Syntaxe	Simple et orientée vers les BDs.	Simple et lisible	Plus complexe
Gestion du mémoire	Automatique	Automatique	Manuel
Héritage	Prend en charge	Prend en charge	Non
Gestion des exceptions	Prise en charge	Prise en charge	Non
Performance	Compilé	Interprété	Compilé
Connectivité avec la BD	Intégrée nativement et optimisée	Large gamme de bibliothèques.	bibliothèques tierces spécifiques..

Table 2.3: Comparaison entre 4D, Python et C++.

State Of Dev In Morocco 2023

Allons plus loin maintenant, nous allons retenir 4D et Python pour une étude comparative exhaustive en terme d'environnement de developpement complet. Commençons par Python, en ce qui suit on va choisir un framework python populaire, en combinaison avec un système de gestion de base de donnes adéquat. Python dispose d'une grande variété de frameworks pour différents types de développement. Selon le classement JetBrains [14] Python Developers Survey 2023 en classe les frameworks Python par utilisation, nous trouvons que pour les frameworks Python, il y a Django en premier, puis Flask en deuxième.

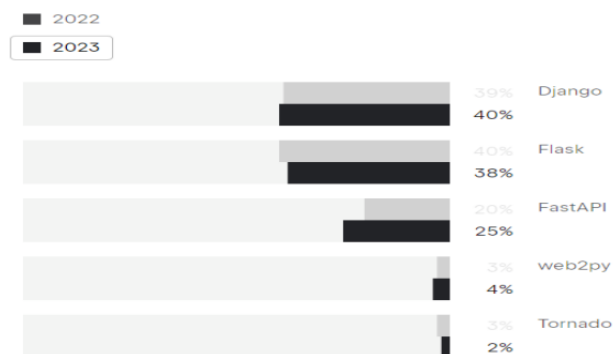


Figure 2.13: Classement des frameworks python 2023 selon JetBrains [14]

Selon ce classement Django est le framework plus utilisés par les développeurs python, c'est est un framework open-source complet et gratuit. Il tente d'inclure toutes les fonctionnalités nécessaires par défaut, au lieu de les proposer sous forme de bibliothèques séparées. Maintenant on'a le framework , on va opter pour le SGBD [15]. Pour rester en cohérence avec le SGBD relationnel intégré dans l'environnement de développement de 4D et dans le but de les compares ultérieurement, on va utiliser l'étude de la figure ci-dessous pour savoir ce qui le SGBD le plus utilisé avec Django. Selon Django Developers Survey 2023, Le SGBD le plus utilisé est PostgreSQL avec un grand pourcentage d'utilisation lors de l'exploitation de Django.

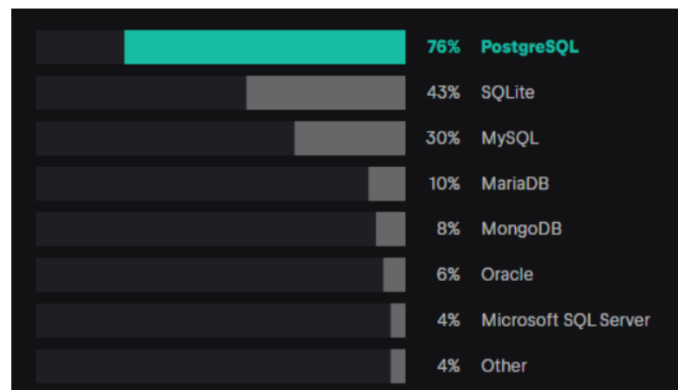


Figure 2.14: Classement des SGBD les plus utilisés avec Django [15]

PostgreSQL est un système de gestion de base de données relationnelle open source puissant et très fiable. Développé à l'origine par Michael Stonebraker à l'Université de Californie à Berkeley, PostgreSQL est constamment amélioré par une communauté active d'experts. Maintenant que nous avons fixé les points de départ en matière de environnements de développement backend pour l'étude comparative, qui sont :

1. **Django + PostgreSQL [16]:** L'environnement de développement Django+PostgreSQL combine le framework web Django avec la base de données relationnelle PostgreSQL, offrant une solution puissante et flexible pour le développement d'applications web. Django, écrit en Python, est un framework de haut niveau qui encourage le développement rapide et une conception propre et pragmatique. Il fournit une structure robuste et des outils intégrés pour les fonctionnalités courantes, comme l'authentification, l'administration, et la gestion des formulaires. [16]
2. **L'environnement développement 4D :** 4D est un environnement de développement complet qui propose une approche intégrée pour la conception, le déploiement et la gestion d'applications professionnelles. Il est largement utilisé dans le développement de logiciels d'entreprise en

raison de ses fonctionnalités avancées et de sa robustesse. Voici quelques points clés sur l'outil 4D. 4D offre un environnement de développement intégré qui comprend un éditeur de code, un gestionnaire de base de données, un serveur d'applications et des outils de déploiement. Il intègre une base de données relationnelle robuste prenant en charge la gestion des données, les requêtes SQL, les transactions et la gestion des utilisateurs. Des outils de conception permettent de créer des interfaces utilisateur conviviales avec des fonctionnalités avancées comme les formulaires dynamiques et les listes déroulantes. ORDA, une technologie introduite par 4D, facilite l'accès aux données relationnelles de manière orientée objet, permettant aux développeurs de manipuler les données comme des objets. En matière de sécurité, 4D propose des fonctionnalités avancées incluant la gestion des utilisateurs, les autorisations d'accès et la cryptographie des données. Comparé à Python+Django, 4D est performant et optimisé pour des applications d'entreprise, avec une bonne scalabilité et une intégration fluide des composants, bien que Python+Django offre une flexibilité et une extensibilité supérieures via de nombreux modules tiers, nécessitant toutefois des configurations plus complexes pour certaines intégrations.

L'étude comparative entre l'environnement de développement Django+PostgreSQL et l'environnement 4D met en lumière les forces et les faiblesses de chaque solution.

Django+PostgreSQL, en exploitant la combinaison du framework web Django avec la base de données relationnelle PostgreSQL, offre une approche robuste et flexible pour le développement d'applications web. Cependant, pour notre application spécifique, l'environnement 4D semble mieux adapté. 4D propose une approche intégrée qui simplifie la conception, le déploiement et la gestion d'applications professionnelles, ce qui répond parfaitement à nos besoins. Avec sa base de données relationnelle intégrée, et la technologie ORDA pour la manipulation orientée objet des données, 4D offre une solution robuste et sécurisée qui peut être facilement adoptée. En synthèse, bien que Django+PostgreSQL offre une flexibilité et une performance appréciables, l'environnement 4D se distingue par son intégration étroite des composants et son adaptation particulièrement adéquate aux exigences de notre application. [17]

2.3.3 Conclusion

React s'est démarqué comme le choix optimal en raison de sa popularité, de sa flexibilité et de ses performances. Grâce à sa communauté active et à son écosystème riche en outils et en

bibliothèques, React offre une base solide pour le développement d'interfaces utilisateur modernes et dynamiques. De plus, son approche de rendu virtuel garantit des performances élevées, ce qui est essentiel pour offrir une expérience utilisateur réactive et fluide, et 4D a émergé comme le choix le plus prometteur en raison de sa nature tout-en-un et de son orientation vers le développement rapide d'applications (RAD). Avec 4D, nous bénéficions d'une intégration complète entre la base de données et la logique métier, ce qui simplifie le processus de développement et réduit les efforts de maintenance à long terme. De plus, ses fonctionnalités avancées de sécurité, de performances optimisées et son support technique complet offrent une solide fondation pour la construction d'applications robustes et évolutives.

Chapter 3

Étude, analyse, spécification des besoins et conception

Introduction

Ce chapitre est consacré à l'analyse et à la spécification. Nous commencerons par examiner le système existant, puis nous spécifierons les exigences fonctionnelles et non fonctionnelles.

L'analyse de ces besoins nous a permis d'identifier les différentes fonctionnalités du système.

Toutes les différentes technologies utilisées tout au long de notre projet sont mentionnées dans la dernière section du chapitre.

3.1 Analyse et spécification des besoins

3.1.1 Description de l'existant

- L'étude de l'existant est une étape clé pour un bon positionnement dans le projet. Elle permet de bien comprendre les besoins de la maîtrise d'ouvrage et les finalités attendues dans le cadre de ce projet.

Actuellement, il existe une plateforme qui a pour but de faciliter les tâches des ressources humaines. Cette plateforme couvre divers aspects de la gestion administrative, notamment l'administration des déplacements, la gestion des stagiaires, la gestion des visas, et d'autres fonctions connexes. Elle permet de centraliser et de simplifier ces processus, offrant ainsi une meilleure efficacité et une réduction des tâches administratives manuelles.

Cependant, malgré ces fonctionnalités, il manque actuellement à cette plateforme une gestion efficace des notes de frais. L'ajout de cette fonctionnalité permettrait de compléter l'ensemble des services proposés et d'améliorer davantage la gestion administrative, en fournissant une solution intégrée pour le suivi et le traitement des dépenses des employés.

3.1.2 Diagramme de Gantt

La plateforme de gestion des notes de frais du projet est utilisée par deux types d'utilisateurs distincts :

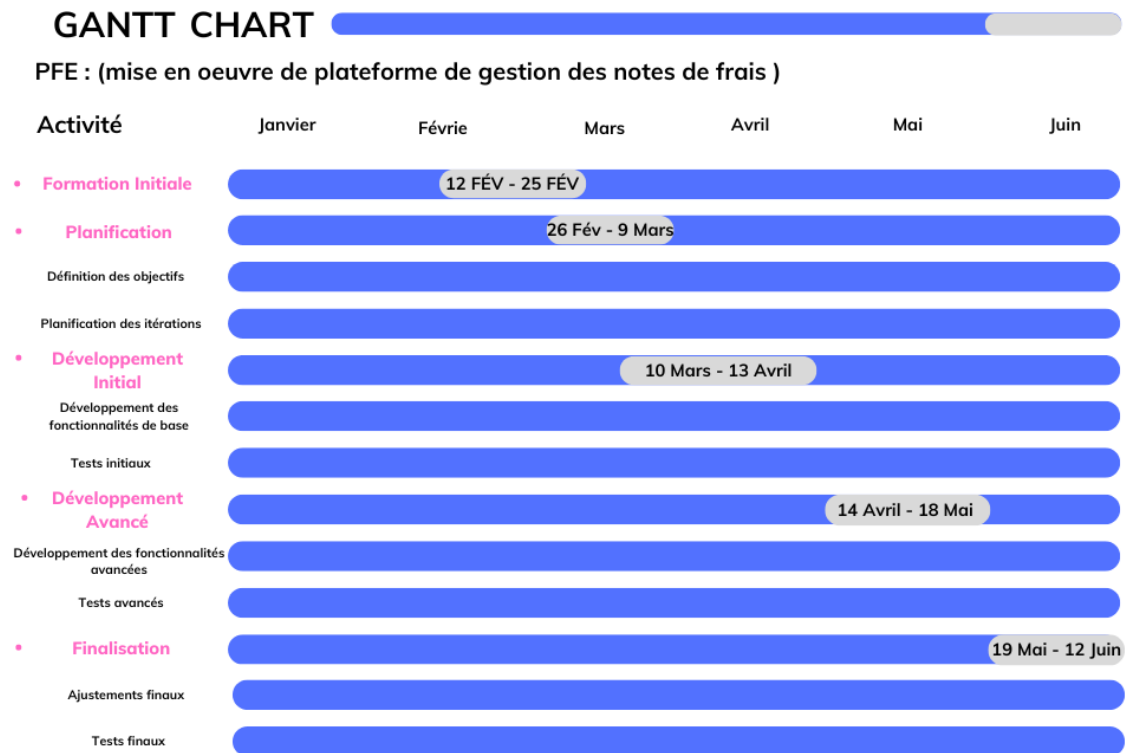


Figure 3.1: Diagramme de Gantt

3.2 Les exigences

3.2.1 Les exigences fonctionnels

Les exigences fonctionnelles décrivent les fonctionnalités spécifiques que l'application de gestion des notes de frais doit offrir pour répondre aux besoins des utilisateurs et des parties prenantes. Voici les exigences fonctionnelles pour ce projet :

Module de gestion des notes de frais

Ce module permet aux utilisateurs de créer, modifier et supprimer des notes de frais. Les utilisateurs peuvent associer les lignes de frais correspondantes à chaque note de frais, ce qui permet de regrouper les dépenses de manière logique et claire.

Module de gestion des lignes de frais

Le module de gestion des lignes de frais offre la possibilité de créer, modifier et supprimer des lignes de frais. Les utilisateurs peuvent spécifier la date, le montant, la description et éventuellement la catégorie de chaque dépense, facilitant ainsi la gestion détaillée des dépenses.

Module de gestion des catégories de frais

Ce module permet de définir les différentes catégories de frais. Les utilisateurs peuvent associer ces catégories aux lignes de frais, ce qui facilite la classification et l'organisation des dépenses en fonction de leur nature.

Module d'aperçu des dépenses

Le module d'aperçu des dépenses offre une vue d'ensemble des dépenses. Il affiche les notes de frais soumises, les dépenses par catégorie et les dépenses totales sur une période donnée, permettant aux utilisateurs de suivre et d'analyser leurs dépenses de manière efficace.

Module de filtrage et de recherche

Ce module permet aux utilisateurs de filtrer et de rechercher facilement des notes de frais en fonction de différents critères. Il offre une fonctionnalité de recherche avancée pour trouver rapidement des informations spécifiques parmi les notes de frais enregistrées.

Module de téléchargement d'une attestation de note de frais

Ce module permet aux utilisateurs de télécharger une attestation de note de frais pour chaque note de frais enregistrée, offrant ainsi une preuve documentée des dépenses engagées.

3.2.2 Les exigences non fonctionnels

Les exigences non fonctionnelles définissent les critères de qualité, de performance et de sécurité que l'application doit respecter. Premièrement, en termes de performance, l'application doit être capable de gérer efficacement un grand nombre de notes de frais et de lignes de frais sans compromettre les performances. En ce qui concerne la sécurité, les données des utilisateurs

doivent être sécurisées et protégées contre tout accès non autorisé, garantissant ainsi la confidentialité des informations sensibles. De plus, l'application doit être fiable et stable, minimisant les risques de perte de données ou d'indisponibilité. Sur le plan de l'interface utilisateur, celle-ci doit être conviviale et intuitive, facilitant ainsi la navigation et l'utilisation de l'application pour les utilisateurs finaux. En termes de maintenabilité, le code source de l'application doit être facilement modifiable et évolutif pour permettre d'ajouter de nouvelles fonctionnalités ou de corriger des bugs facilement.

3.3 Modélisation des exigences

Dans cette partie, nous présentons d'abord le diagramme de cas d'utilisation général. Cette phase représente la vue fonctionnelle de l'architecture du système. Ensuite, nous identifions les principaux cas d'utilisation de notre solution. Cette identification sera ensuite suivie d'une description détaillée de chacun d'entre eux.

Dans cette section, nous présentons une série de diagrammes UML qui illustrent visuellement l'architecture et les processus du système de gestion des notes de frais.

Afin de répondre aux exigences du projet de gestion des notes de frais, il est crucial de concevoir et de mettre en œuvre une solution complète qui réponde aux objectifs du projet. Un aspect clé du projet est l'introduction d'un moteur de gestion des dépenses robuste. Ce moteur facilitera une gestion efficace et précise des dépenses, optimisera l'analyse financière et soutiendra les processus de prise de décision. Pour garantir une compréhension claire de la solution proposée, divers diagrammes UML ont été utilisés pour illustrer l'architecture du système, les flux de travail et les interactions.

3.3.1 Présentation de l'UML

Introduction à UML

UML (Unified Modeling Language) est une méthode de modélisation orientée objet développée en réponse à l'appel aux propositions lancées par l'OMG dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets. Elle est héritée de plusieurs autres méthodes telles que Booch, OOSE et OMT. UML est une méthode utilisant une

représentation graphique qui est un excellent complément aux représentations textuelles. En effet, l'une comme l'autre sont ambiguës mais leur utilisation simultanée permet de diminuer les ambiguïtés de chacune d'elles. Un dessin permet bien souvent d'exprimer clairement ce qu'un texte exprime difficilement et un bon commentaire permet d'enrichir une figure. [7]

Pourquoi choisir UML ?

Nous avons opté pour UML comme langage de modélisation vu ses multiples points forts. UML est un langage formel et normalisé : il représente un gain de précision, un gage de stabilité et encourage l'utilisation d'outils. UML est un support de communication performant puisqu'il cadre l'analyse, facilite la compréhension de représentations abstraites complexes, en outre son caractère polyvalent et sa souplesse en font un langage universel.

Contrairement aux diagrammes ERD qui sont principalement utilisés pour la conception de bases de données, et à SysML qui est plus adapté aux systèmes complexes d'ingénierie, UML nous offre une approche plus globale et versatile pour la modélisation des applications web. UML fournit une gamme complète de diagrammes nous permettant de capturer les différents aspects de notre application web, tels que les fonctionnalités, l'architecture logicielle, les interactions entre les composants et les flux de données. Cette richesse de représentation fait d'UML un standard de facto pour la conception et le développement d'applications orientées objet, ce qui correspond parfaitement aux besoins d'une application web moderne.

D'un point de vue plus pratique, UML va me permettre de décrire les différents aspects de mon application via les diagrammes suivants [8]:

- Diagramme de cas d'utilisation pour recenser les grandes fonctionnalités du système.
- Diagramme de séquence pour représenter les interactions entre les entités.
- Diagramme de classe pour représenter les éléments qui composent le système et leurs relations.

3.3.2 Diagrammes des cas d'utilisations

Diagramme des cas d'utilisation globale

- Ce diagramme permettra d'identifier les différents acteurs et les interactions entre eux, fournissant ainsi une vision globale des principales fonctionnalités de l'application. En examinant ce diagramme, les parties prenantes pourront avoir une compréhension des scénarios d'utilisation de l'application, des actions possibles des acteurs principales.

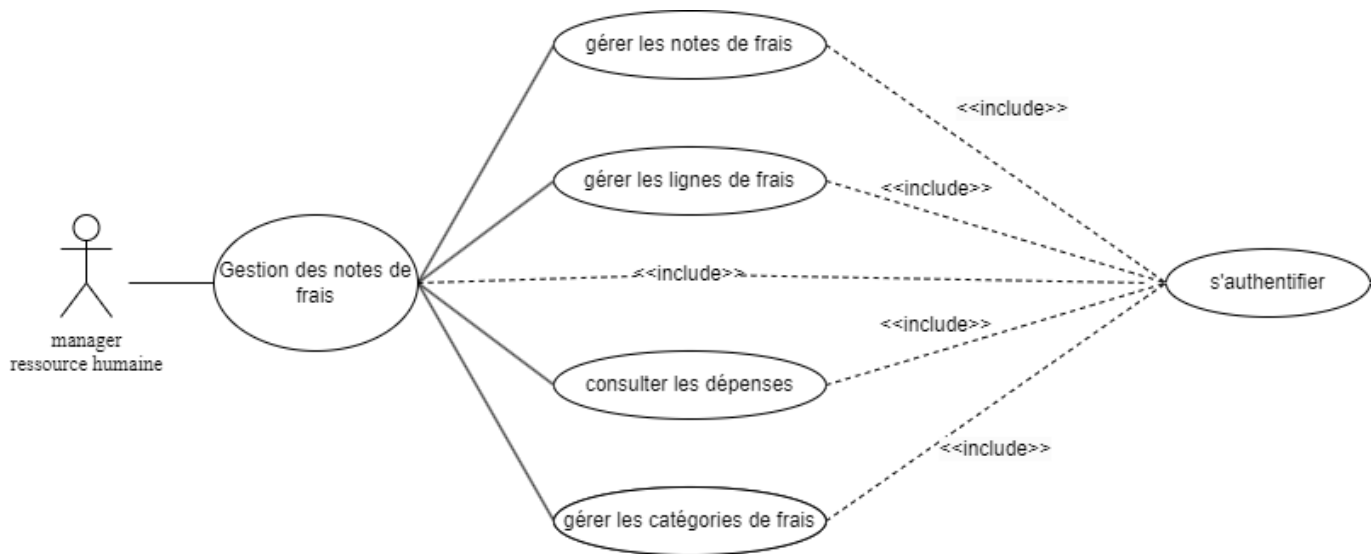


Figure 3.2: Diagramme des cas d'utilisation globale.

Diagramme des cas d'utilisation en détails

- Pour un système de gestion de notes de frais avec un seul participant (le manager ou le RH) jouant le rôle d'administrateur, voici comment vous pourriez décrire le diagramme UML des cas d'utilisation. Ce diagramme va capturer les principales fonctionnalités et interactions que l'administrateur (manager/RH) a pour assurer la gestion et le bon fonctionnement de la plateforme de gestion des notes de frais.

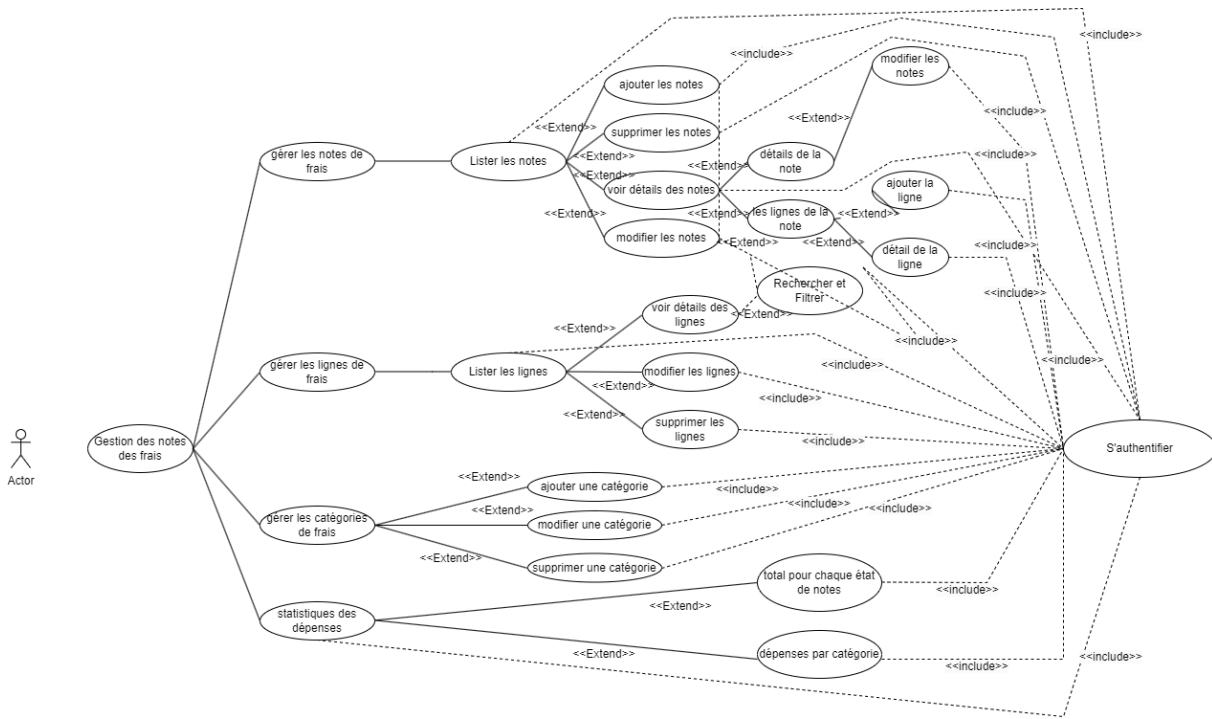


Figure 3.3: Diagramme des cas d'utilisation en détails

3.3.3 Description de quelques cas d'utilisation

Description du cas d'utilisation « S'authentifier »

Cas d'utilisation	S'authentifier
Acteurs	Un administrateur, un organisateur ou un participant
Description	utilisateur enregistré peut se connecter au système en saisissant son adresse e-mail et son mot de passe. L'application authentifie l'utilisateur s'il est valide.
Pré-conditions	L'utilisateur doit posséder un compte valide.
Post-conditions	L'utilisateur est connecté au plateforme et peut accéder aux fonctionnalités autorisées.
Scénario nominal	<ol style="list-style-type: none"> 1. Sélection de "Login" sur la page d'accueil. 2. L'application affiche un formulaire de connexion. 3. L'utilisateur saisit ses informations d'identification et valide. 4. L'application vérifie la validité des informations saisies. 5. Si les informations sont valides, le système authentifie l'utilisateur et le redirige vers la page d'accueil.
Scénario alternative	<ol style="list-style-type: none"> 1) Champs vide : le système affiche un message d'erreur 2) Email invalide : le système affiche un message d'erreur. 3) Mot de passe invalide : le système affiche un message d'erreur 4) Mot de passe oublié : un email de réinitialisation de mot de passe est envoyé à l'utilisateur.

Table 3.1: Description du cas d'utilisation "S'authentifier".

Description du cas d'utilisation « Réinitialiser le mot de passe »

Cas d'utilisation	Réinitialiser le mot de passe
Acteurs	Un manager, un employé, ou responsable ressource humaine
Description	Si un utilisateur oublie son propre mot de passe , il peut le réinitialiser en utilisant son email
Pré-conditions	L'utilisateur doit avoir un compte valide
Post-conditions	Un nouveau mot de passe est créé, connexion possible.
Scénario nominal	<ol style="list-style-type: none"> 1. Sélection de "Forgot password ?" sur la page d'authentification. 2. Insertion d'adresse email d'utilisateur. 3. Réception d'un email contenant un mot de passe temporaire. 4. Accès au plateforme avec le mot de passe généré. 5. Création d'un nouveau mot de passe sur la page profil. 6. Valider le nouveau mot de passe.
Scénario alternative	<ol style="list-style-type: none"> 1) Service web ou serveur de base de données indisponible : le système affiche un message d'erreur.

Table 3.2: Description du cas d'utilisation "Réinitialiser le mot de passe".

Description du cas d'utilisation « Ajouter une note»

Cas d'utilisation	Créer une note
Acteurs	Un administrateur
Description	Un utilisateur peut créer une note et désigné l'employé en question.
Pré-conditions	l'administrateur s'est authentifié, et peut créer des notes.
Post-conditions	Une note est créée avec un statut initial.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède au détail de la note. 2. L'utilisateur peut ajouter les dépenses correspondantes.

Table 3.3: Description du cas d'utilisation "Ajouter une note".

Description du cas d'utilisation « Ajouter une ligne »

Cas d'utilisation	Ajouter une ligne
Acteurs	Un administrateur
Description	L'administrateur peut créer une nouvelle ligne en remplissant une formulaire des informations nécessaires.
Pré-conditions	Administrateur authentifié.
Post-conditions	Nouvelle ligne créée .
Scénario nominal	<ol style="list-style-type: none"> 1. Sélection de bouton Ajouter. 2. Affichage d'un formulaire d'informations de la ligne. 3. Saisie et validation des informations. 4. Ajout de justificatif.

Table 3.4: Description du cas d'utilisation "Ajouter une ligne".

Description du cas d'utilisation « Visualiser la liste des notes »

Cas d'utilisation	Visualiser la liste des notes
Acteurs	Un administrateur
Description	Un utilisateur consulte la page des notes et accède au services associées.
Pré-conditions	l'administrateur s'est authentifié, et peut crée des notes.
Post-conditions	Nouvelle ligne créée .
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page des notes. 2. L'utilisateur consulte les notes. 3. L'utilisateur peut effectuer plusieurs actions pour bénéficier des services de cet espace.
Scénario alternative	Service web ou serveur de base de données indisponible : le système affiche un message d'erreur.

Table 3.5: Description du cas d'utilisation "Visualiser la liste des notes".

Description du cas d'utilisation « Modifier une note »

Acteurs	Un administrateur
Description	Un utilisateur peut modifier les informations d'une note de frais existante.
Pré-conditions	La note de frais doit exister
Post-conditions	Les modifications sont enregistrées et mises à jour dans le système.
Scénario nominal	<ul style="list-style-type: none"> • Le utilisateur sélectionne une note de frais à modifier. • Le système affiche les détails actuels de la note de frais. <ul style="list-style-type: none"> • L'utilisateur modifie les informations nécessaires. <ul style="list-style-type: none"> • Le utilisateur soumet les modifications. • Le système valide les nouvelles informations
Scénario alternatif	<ul style="list-style-type: none"> • Si les nouvelles informations ne sont pas valides, le système affiche un message d'erreur.

Table 3.6: Description du cas d'utilisation "Modifier une note".

Description du cas d'utilisation « Modifier une ligne »

Acteurs	Un administrateur
Description	Un utilisateur peut modifier les informations d'une ligne de frais existante.
Pré-conditions	La note de frais doit exister
Post-conditions	Les modifications sont enregistrées et mises à jour dans le système.
Scénario nominal	<ul style="list-style-type: none"> • Le utilisateur sélectionne une ligne de frais à modifier. • Le système affiche les détails actuels de la ligne de frais. <ul style="list-style-type: none"> • L'utilisateur modifie les informations nécessaires. <ul style="list-style-type: none"> • Le utilisateur soumet les modifications. • Le système valide les nouvelles informations
Scénario alternatif	<ul style="list-style-type: none"> • Si les nouvelles informations ne sont pas valides, le système affiche un message d'erreur.

Table 3.7: Description du cas d'utilisation "Modifier une ligne".

La phase de spécification des besoins nous a permis de bien définir les différents rôles et responsabilités au sein de notre application. Nous avons ainsi identifié le rôle principal de l'utilisateur, chargé de la saisie et de la gestion des notes de frais. Nous avons également clarifié la distinction entre les différents éléments. Cette compréhension approfondie des besoins et des profils d'utilisateurs va nous servir de base pour la suite du projet. Forts de cette compréhension des exigences, nous pouvons désormais passer à l'étape de conception de l'application. Cette phase va nous permettre de traduire ces besoins en une architecture et des fonctionnalités concrètes. Nous allons notamment concevoir les diagrammes de séquence pour les cas d'utilisation clés, illustrant les interactions entre les acteurs et les composants du système. De plus, nous établirons le diagramme de classes, qui définira la structure des objets, leurs attributs et leurs relations, servant de base pour le développement ultérieur du code source.

3.3.4 Conception de la solution

3.3.4.1 Diagrammes des séquences

Les cas d'utilisation dans les diagrammes précédent sont raffinés par la suite dans des diagrammes de séquence afin d'avoir une présentation claire sur les différentes interactions entre les acteurs et le système selon un ordre chronologique. Dans cette partie, nous allons décrire les scénarios les plus importants ainsi que leurs représentations par les diagrammes de séquences.

Diagramme de séquence du cas d'utilisation "S'authentifier"

Le diagramme de séquence pour l'authentification (figure 2.5) décrit les interactions nécessaires à la validation des identifiants de l'utilisateur. Initialement, l'utilisateur insère ses identifiants d'authentification, à savoir l'adresse mail et le mot de passe, dans l'interface de connexion. Ces identifiants sont ensuite envoyés par le front-end au back-end pour vérification. Le back-end vérifie alors les informations fournies par l'utilisateur en les comparant à celles stockées dans la base de données. Si les informations sont correctes, le back-end ouvre une session pour l'utilisateur et envoie une réponse positive au front-end pour maintenir la session ouverte. Le front-end stocke cette information localement pour les futures requêtes. Suite à une authentification réussie, l'utilisateur est redirigé vers la page d'accueil. En cas d'authentification invalide, le back-end renvoie une erreur d'authentification au front-end, qui affiche alors un message d'erreur à l'utilisateur.

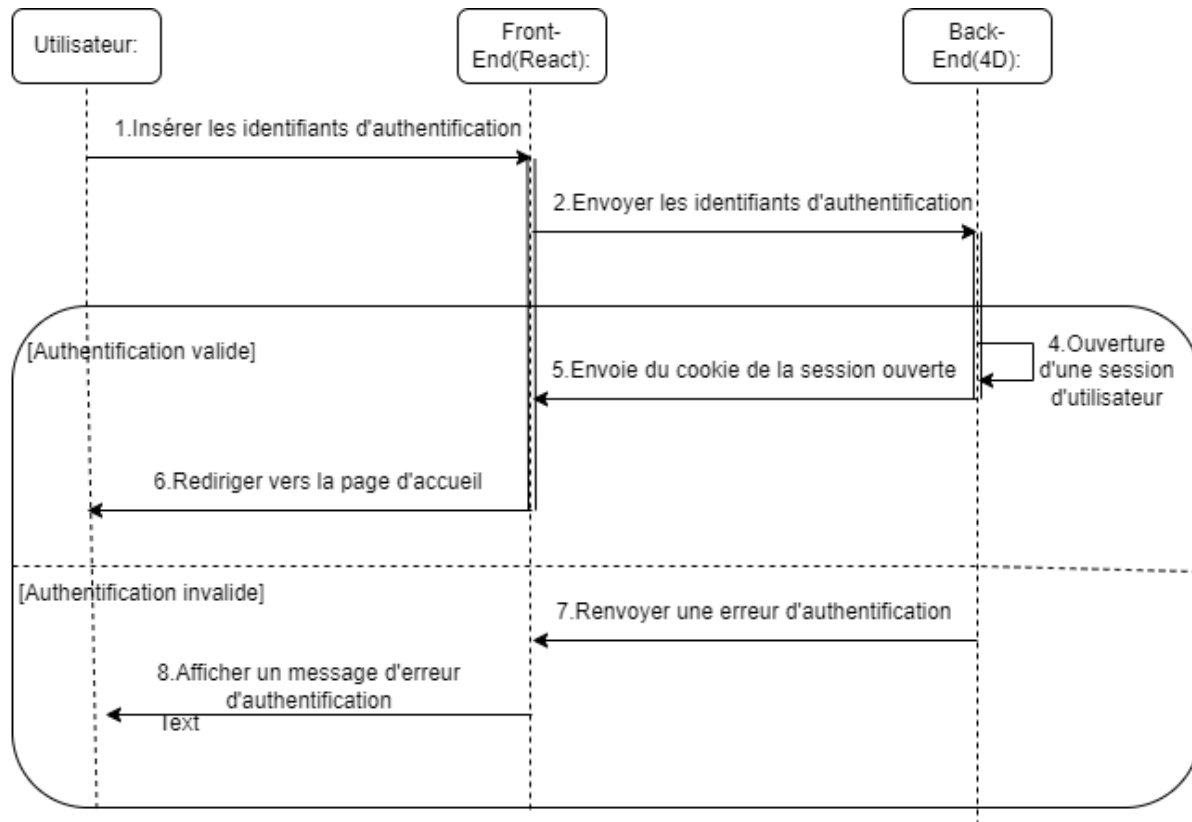


Figure 3.4: Diagramme de séquence du cas d'utilisation "S'authentifier"

Diagramme de séquence du cas d'utilisation "Ajouter une note"

Le processus d'ajout d'une note sur notre plateforme démarre par la sélection de l'option "Ajouter une note" sur la page dédiée aux notes, déclenchant ainsi l'affichage d'un formulaire spécifique pour renseigner les informations relatives à la note. Les informations valides sont envoyées au back-end qui enregistre la note dans la base de données.

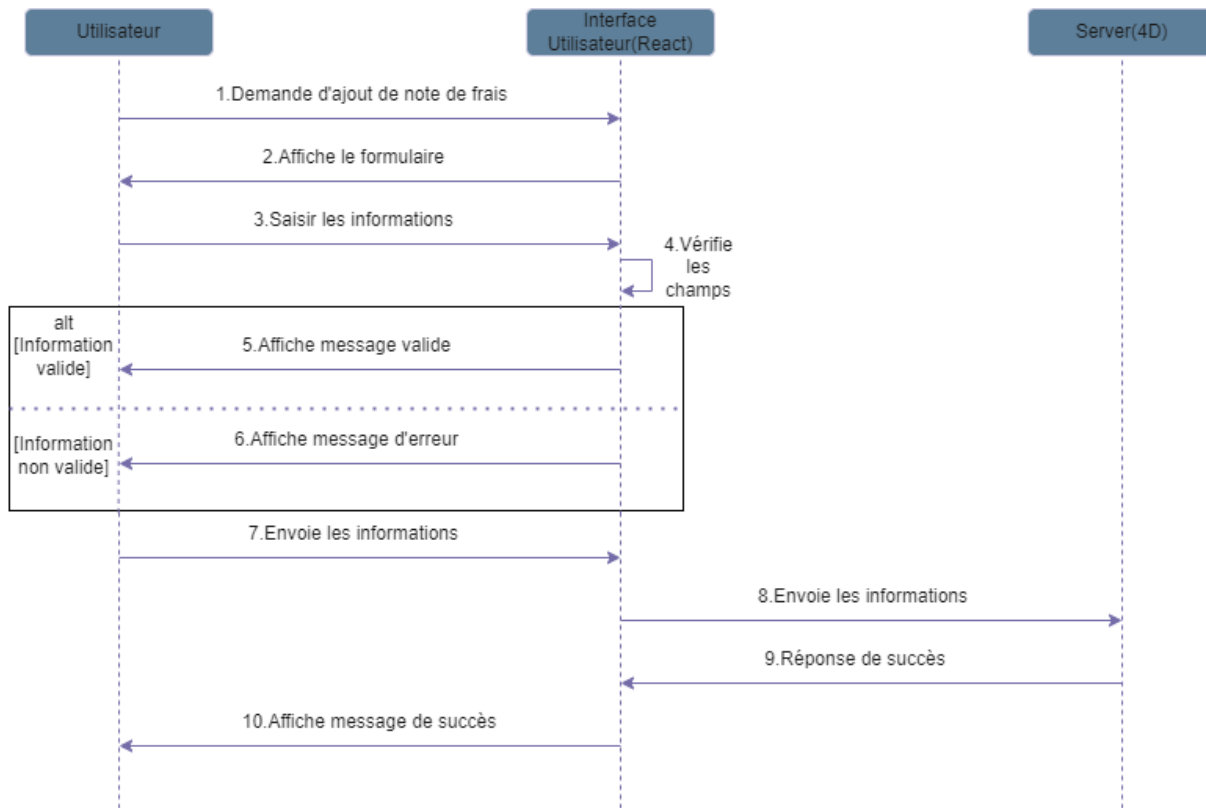


Figure 3.5: Diagramme de de séquence du cas d'utilisation " ajouter une note"

Diagramme de séquence du cas d'utilisation "Ajouter une ligne"

Le processus d'ajout d'une ligne de frais sur notre plateforme démarre par la sélection de l'option "Ajouter" sur la page dédiée aux frais, déclenchant ainsi l'affichage d'un formulaire spécifique pour renseigner les informations relatives à la ligne de frais. Les informations valides sont envoyées au back-end qui enregistre la ligne de frais dans la base de données.

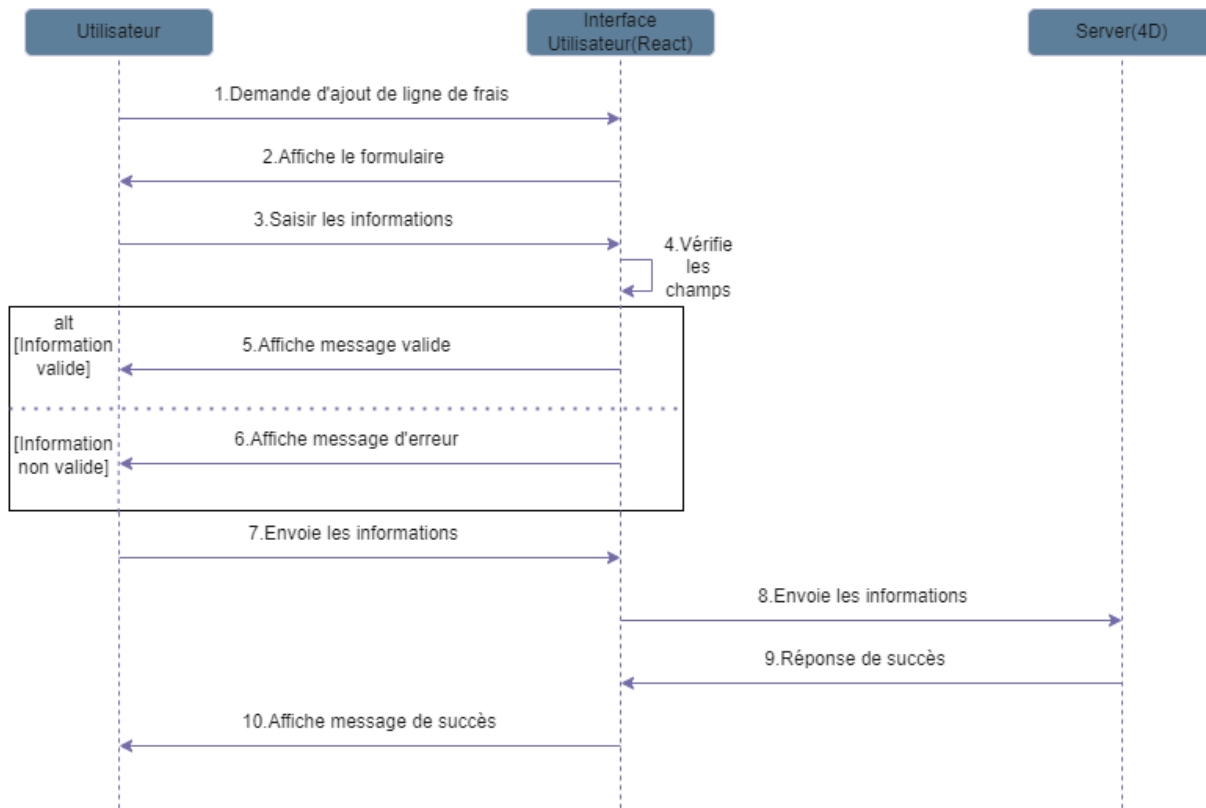


Figure 3.6: Diagramme de séquence du cas d'utilisation "ajouter une ligne"

Diagramme de séquence du cas d'utilisation "modifier une note"

Le processus de modification d'une note sur notre plateforme commence par la sélection de l'option "Modifier" sur la page dédiée aux notes, ce qui déclenche l'affichage d'un formulaire spécifique permettant de modifier les informations de la note. Une fois les modifications effectuées et validées, les données sont envoyées au back-end pour mettre à jour la note dans la base de données.

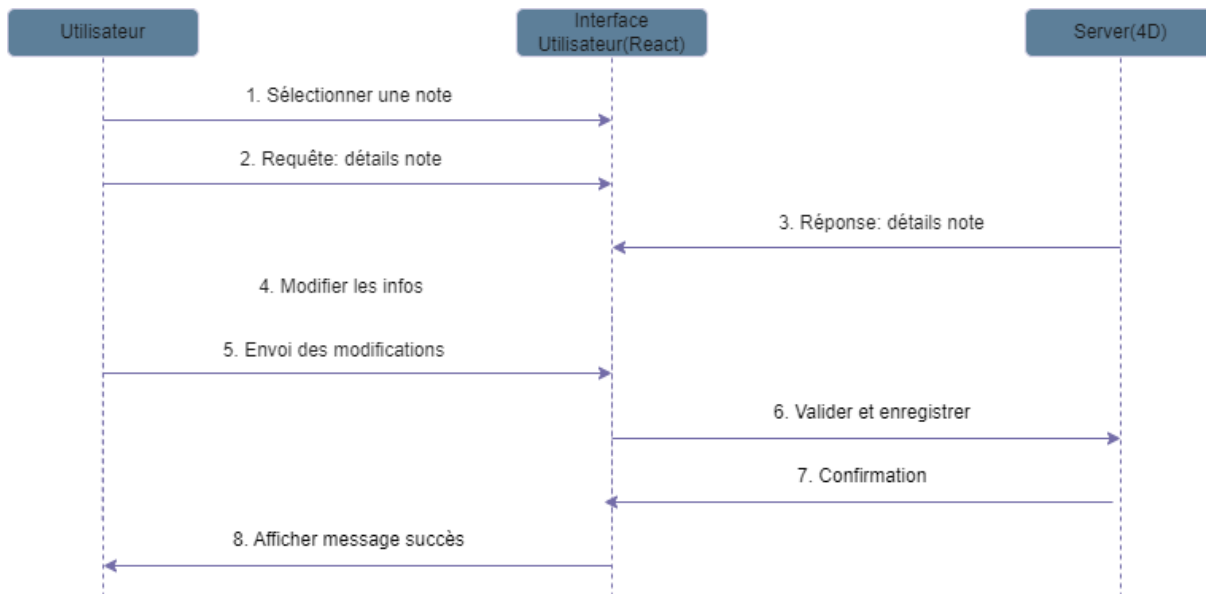


Figure 3.7: Diagramme de séquence du cas d'utilisation "modifier une note"

Diagramme de séquence du cas d'utilisation "modifier une ligne"

Le processus de modification d'une ligne sur notre plateforme commence par la sélection de l'option "Modifier" sur la page dédiée aux lignes, ce qui déclenche l'affichage d'un formulaire spécifique permettant de modifier les informations de la ligne. Une fois les modifications effectuées et validées, les données sont envoyées au back-end pour mettre à jour la note dans la base de données.

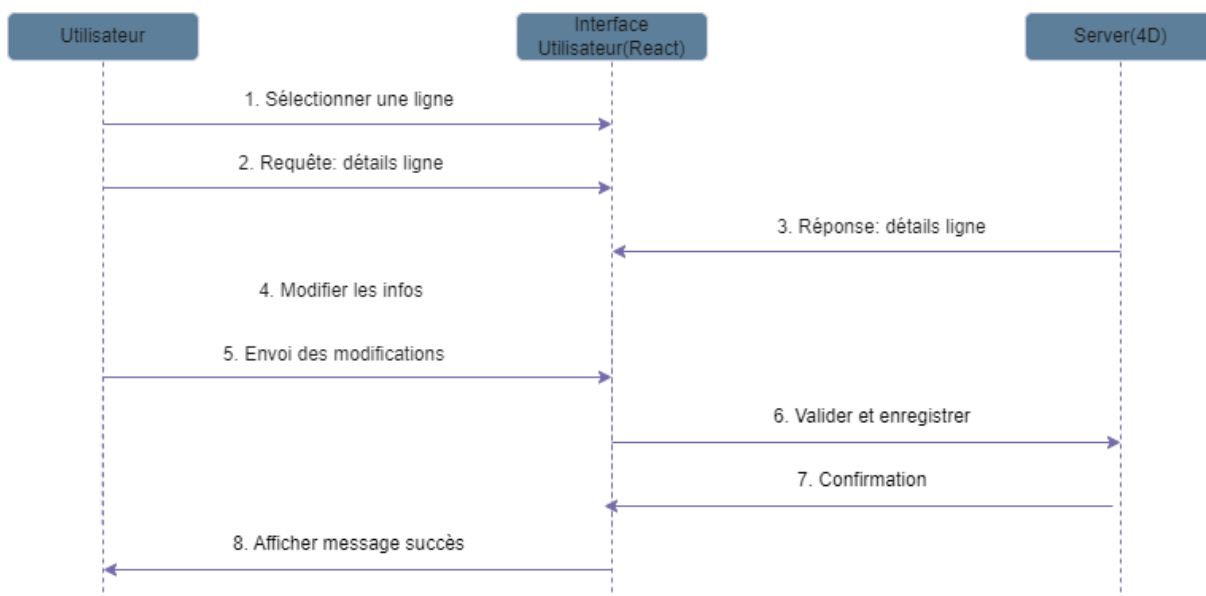


Figure 3.8: Diagramme de séquence du cas d'utilisation "modifier une ligne"

- Les opérations doivent être effectuées rapidement, avec des temps de réponse minimaux.

3.3.4.2 Diagramme de classes

- Le diagramme de classes est considéré comme l'un des éléments les plus importants de la modélisation orientée objet. Il en montre la structure interne et permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Nous avons réalisé le diagramme de classes suivant pour expliquer la structure intérieure de la partie "Notes" de la base de données. Le diagramme représente les classes ainsi que les associations qui les relient. Puisqu'on travaille avec la méthode agile, cette structure peut être mise à jour lors du développement de l'application, en ajoutant d'autres classes ou en modifiant les attributs et méthodes des classes affichées dans le diagramme ci-dessous.

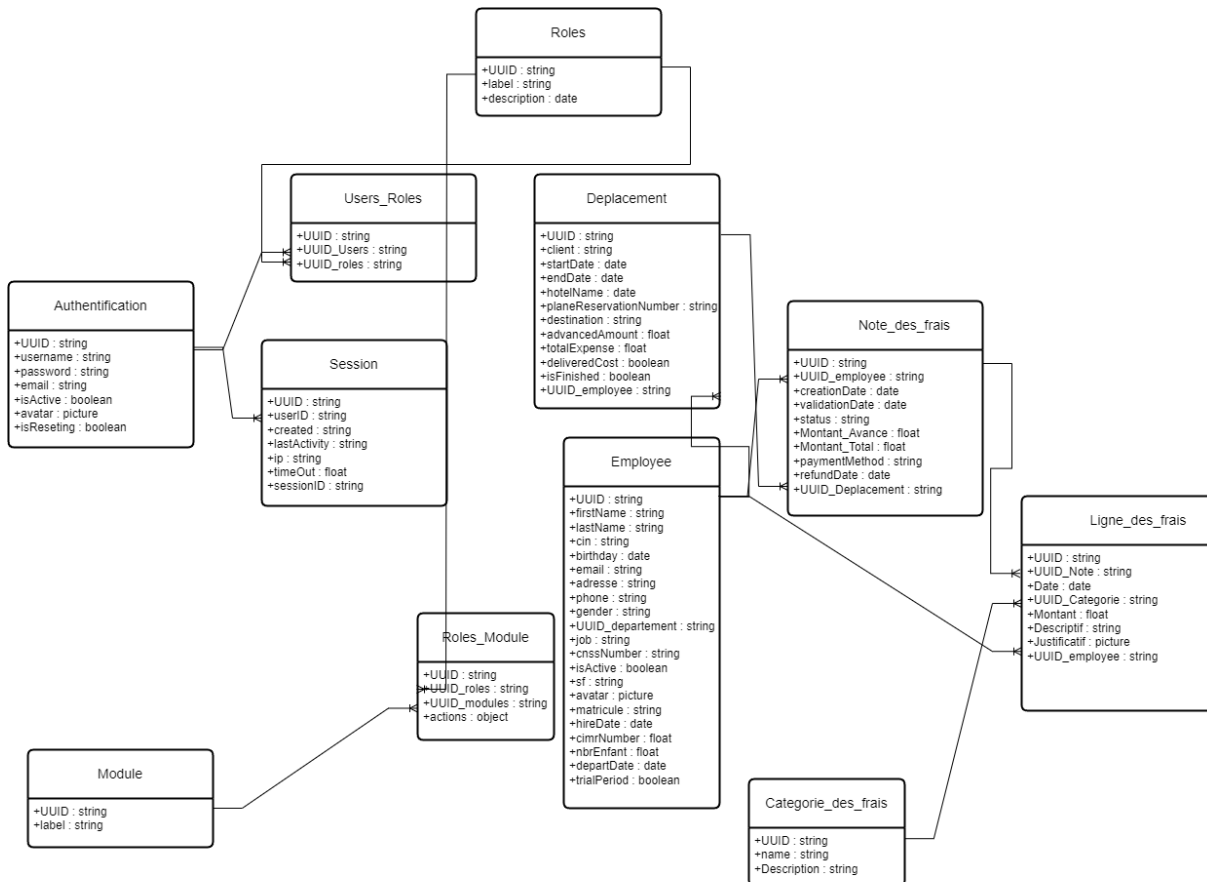


Figure 3.9: Diagramme de classe

Conclusion

Ce chapitre a exploré la gestion des notes de frais à travers des plateformes numériques, en détaillant les différents acteurs impliqués et leurs interactions. Nous avons approfondi l'étude

fonctionnelle de notre projet en identifiant ses besoins, qu'ils soient fonctionnels ou non fonctionnels, tout en établissant les fondements d'une plateforme robuste et conviviale. La conception de l'application a été enrichie par des diagrammes de séquences et de classes, offrant une vue claire des interactions et de l'architecture système. Ces démarches cruciales posent les bases d'un succès futur pour notre application.

Chapter 4

Mise en oeuvre de la solution

Introduction

Cette partie, de la mise en œuvre du projet, constitue le dernier volet du rapport. Après avoir terminé la phase de la spécification et la conception, la phase du choix des technologies et environnements techniques à utiliser, il nous reste qu'à illustrer les fonctionnalités implémentées pour notre application EventHub en montrant quelques interfaces réalisées tout en expliquant les différentes fonctionnalités.

4.1 Réalisation

Cette partie illustre les fonctionnalités que nous avons implémenté dans ce projet en montrant quelques interfaces des différents modules composant la solution développée, tout en expliquant comment ces fonctionnalités répondent aux différents besoins des acteurs préalablement définis.

4.1.1 Authentification

4.1.1.1 Connexion



The image shows a login form for a system branded with a logo consisting of a white '4' inside a dark blue square. The form is titled 'CONNECTION' in bold, uppercase letters. It contains two input fields: 'Email' with the value 'hamaoui.yaflman@4d.com' and 'Mot de passe' with masked characters '.....'. A toggle icon for password visibility is present on the right of the password field. Below the password field is a link labeled 'mot de passe oublié' in blue text. At the bottom of the form is a large blue button with the text 'Se connecter' in white.

Figure 4.1: Page de connexion.

La page de connexion ci-dessus permet à l'utilisateur de se connecter à son espace client en saisissant le bon mot de passe et email. Dans le cas contraire, L'utilisateur récupère son mot de

passer client en utilisant la fonctionnalité "Mot de passe oublié". Cette fonctionnalité est accessible via le lien "Mot de passe oublié" .

4.1.1.2 Mot de passe oublié





MOT DE PASSE OUBLIÉ

Indiquez votre adresse e-mail. Vous recevrez un email avec un lien pour chnager le mot de passe

Email

Annuler **Envoyer**

Figure 4.2: Page de connexion.

La page "Nouveau mot de passe" est essentiel pour permettre la redéfinition du mot de passe à travers l'adresse e-mail.

4.1.2 Page d'accueil

La page d'accueil est la première page affichée aux utilisateurs. Lors de sa conception, il a été nécessaire de trouver une disposition respectant simultanément la simplicité, l'ergonomie et l'efficacité. Cette page est composée de plusieurs modules qui apparaissent selon le rôle donné aux utilisateurs tels Administration , Gestion de document , et autres. Notre module principale est celui de gestion de note des frais.



Figure 4.3: Page d'accueil principale.



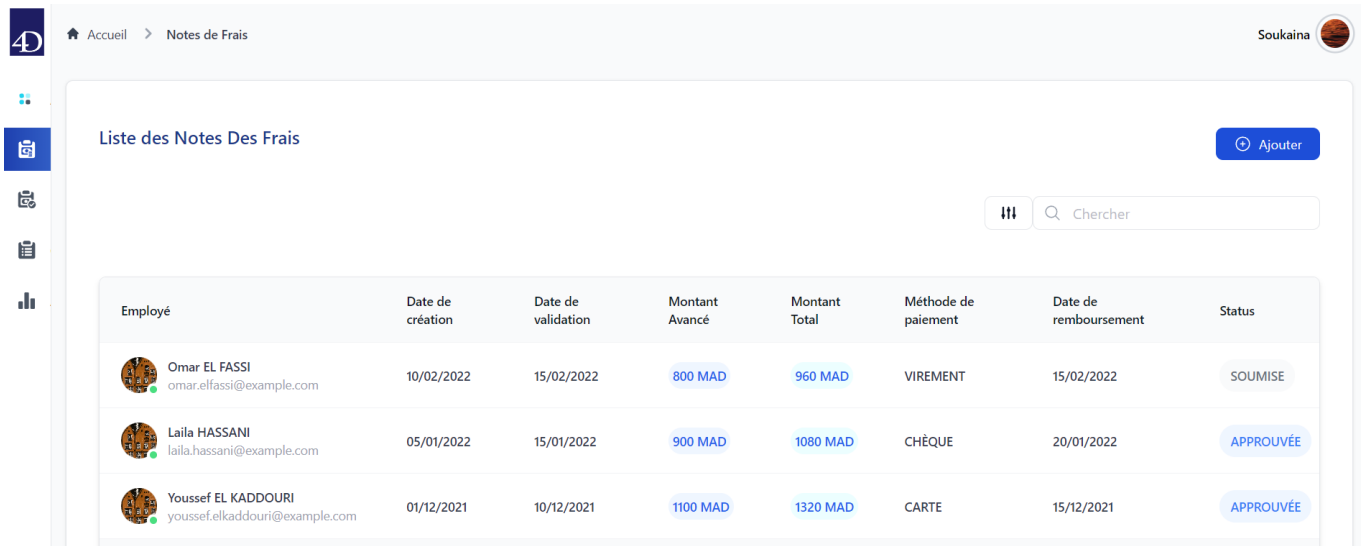
Figure 4.4: Page d'accueil de gestion de note de frais.

4.1.3 Le module de gestion de note de frais

4.1.3.1 Le module de note de frais

Liste des notes:

Cette section permet à l'utilisateur de voir les notes créées ainsi que leurs détails. Elle peut également filtrer ou rechercher selon les attributs comme méthode de paiement ou employé concerné tout en fournissant la possibilité d'ajouter une autre note ou de modifier une qui existe déjà en remplissant les attributs nécessaires.






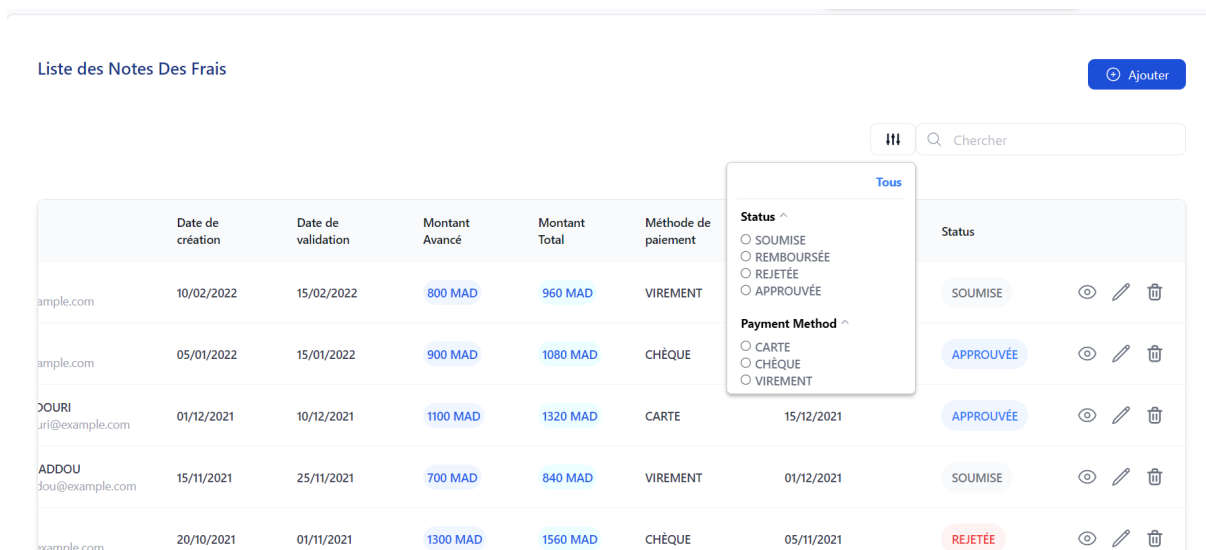
Employé	Date de création	Date de validation	Montant Avancé	Montant Total	Méthode de paiement	Date de remboursement	Status
 Omar EL FASSI omar.elfassi@example.com	10/02/2022	15/02/2022	800 MAD	960 MAD	VIREMENT	15/02/2022	SOUMISE
 Laïla HASSANI laïla.hassani@example.com	05/01/2022	15/01/2022	900 MAD	1080 MAD	CHÈQUE	20/01/2022	APPROUVÉE
 Youssef EL KADDOURI youssef.elkaddouri@example.com	01/12/2021	10/12/2021	1100 MAD	1320 MAD	CARTE	15/12/2021	APPROUVÉE

Figure 4.5: liste des notes de frais.

Filtrage et recherche:

Cette section permet à l'utilisateur de filtrer ou chercher des notes selon les options de chaque champ.



	Date de création	Date de validation	Montant Avancé	Montant Total	Méthode de paiement		Status
ample.com	10/02/2022	15/02/2022	800 MAD	960 MAD	VIREMENT		SOUMISE
ample.com	05/01/2022	15/01/2022	900 MAD	1080 MAD	CHÈQUE		APPROUVÉE
DOURI uri@example.com	01/12/2021	10/12/2021	1100 MAD	1320 MAD	CARTE	15/12/2021	APPROUVÉE
ADDOU dou@example.com	15/11/2021	25/11/2021	700 MAD	840 MAD	VIREMENT	01/12/2021	SOUMISE
example.com	20/10/2021	01/11/2021	1300 MAD	1560 MAD	CHÈQUE	05/11/2021	REJETÉE

Figure 4.6: Filtrage et Recherche.

Ajouter une note:

Nouvelle note de frais


Employé	Date de Création	Date de Validation
<input type="text" value="employé *"/>	<input type="text" value="jj/mm/aaaa"/>	<input type="text" value="jj/mm/aaaa"/>
Montant Avance	Montant Total	methode de paiement
<input type="text" value="Montant *"/>	<input type="text" value="Montant *"/>	<input type="text" value="methode de paiement *"/>
Date de remboursement	status	
<input type="text" value="jj/mm/aaaa"/>	<input type="text" value="status *"/>	

AnnulerAjouter

Figure 4.7: Ajouter une note.

Détail d'une note:

Dossier note de frais



Omar EL FASSI
omar.elfassi@example.com
Compte Active

Date de creation : 10/02/2022
Date de validation : 15/02/2022

Informations de notes de frais

Montant Avance	800 MAD
Montant Total	960 MAD
Methode de paiement	VIREMENT
Date de remboursement	15/02/2022
Status	SOUMISE

Télécharger attestation de note de frais

ModifierRetour

Figure 4.8: Détail d'une note.

Modifier une note:

The screenshot shows a web application interface for modifying a note. The breadcrumb trail is 'Accueil > Notes de Frais > Modifier Notes de Frais'. The user 'Soukaina' is logged in. The form 'Modifier la note de frais' contains the following fields:

- Employé:** A dropdown menu with 'Select...' as the placeholder.
- Date de Création:** A date input field showing '10/02/2022'.
- Date de Validation:** A date input field showing '15/02/2022'.
- Montant Avance:** A text input field with the value '800'.
- Montant Total:** A text input field with the value '960'.
- methode de paiement:** A dropdown menu with 'methode de paiement *' as the placeholder.
- Date de remboursement:** A date input field with the placeholder 'jj/mm/aaaa'.
- status:** A dropdown menu with 'status *' as the placeholder.

At the bottom of the form are two buttons: 'Annuler' (grey) and 'Ajouter' (blue).

Figure 4.9: Modifier une note.

voir ligne spécifique d'une note:

The screenshot shows a table titled 'Lignes de frais'. Above the table is a 'Montrer Moins' button. To the right of the table is an 'Ajouter' button. The table has three columns: 'Date', 'Catégorie', and 'Montant'. Each row also has a 'Détails' link on the right.

Date	Catégorie	Montant	
15/06/2023	Transport	7500 MAD	Détails
10/06/2023	Hébergement	5000 MAD	Détails
05/06/2023	Repas	30000 MAD	Détails
01/06/2023	Transport	15000 MAD	Détails

Figure 4.10: voir ligne spécifique d'une note.

4.1.3.2 Le module de ligne de frais

Ce module permet aux utilisateurs de gérer les lignes de frais. Les fonctionnalités incluent la visualisation, la modification et l'ajout de nouvelles lignes de frais.

Liste des lignes:

Cette fonctionnalité affiche toutes les lignes de frais enregistrées dans le système. Les utilisateurs peuvent visualiser les détails de chaque ligne, y compris la date, le montant et la catégorie associée.

Liste des dépenses

Chaque note de frais peut contenir une ou plusieurs dépenses liées. Utilisez le tableau ci-dessous pour visualiser et gérer les dépenses associées à chaque note de frais.

🔍 Chercher


















Employé	Date	Catégorie	Montant	Descriptif	Justificatif
 Laila HASSANI laila.hassani@example.com	15/06/2023	Transport	7500 MAD	Frais de bureau	Aucun justificatif ajouté   
 Laila HASSANI laila.hassani@example.com	10/06/2023	Hébergement	5000 MAD	Transport en taxi	Aucun justificatif ajouté   
 Laila HASSANI laila.hassani@example.com	05/06/2023	Repas	30000 MAD	Hébergement lors d'un déplacement professionnel	Aucun justificatif ajouté   
 Laila HASSANI laila.hassani@example.com	01/06/2023	Transport	15000 MAD	Déjeuner d'affaires avec client	Aucun justificatif ajouté   

Figure 4.11: Liste des lignes de frais.

Détail de la ligne:

En cliquant sur une ligne de frais spécifique, l'utilisateur peut voir les détails complets de cette dépense.

Dossier d'une dépense

 **Laila HASSANI**
laila.hassani@example.com
Compte Active

Date
15/06/2023

Informations sur la dépense

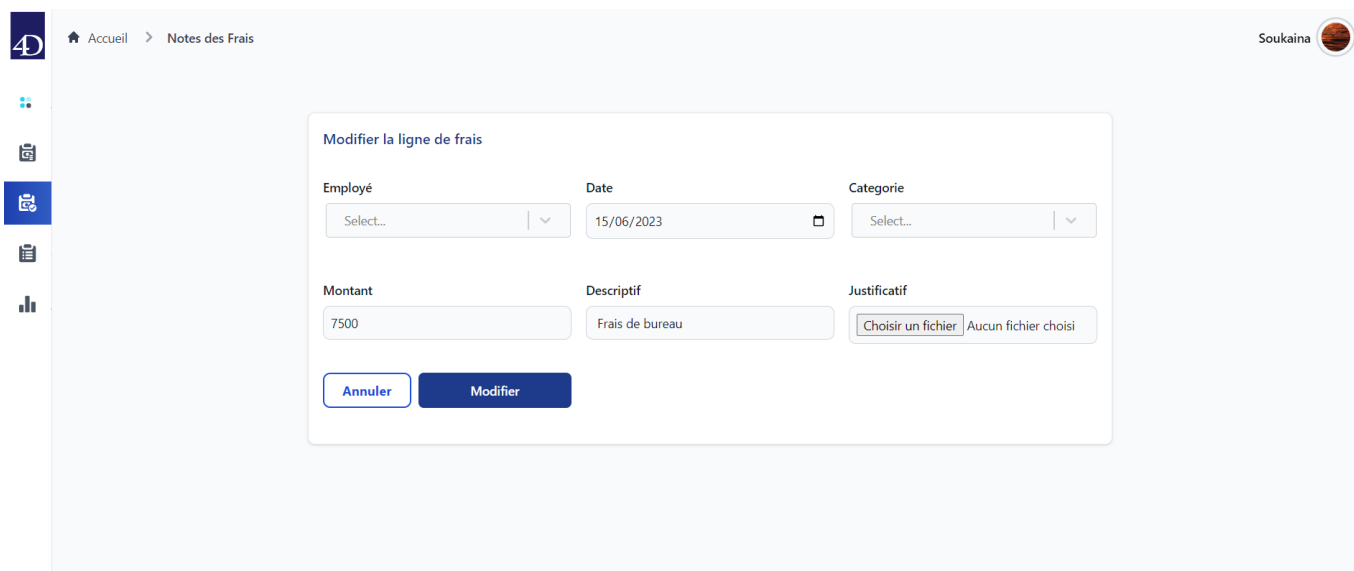
Catégorie	Transport
Montant	7500
Description	Frais de bureau
Justificatif	Veillez fournir un justificatif

ModifierRetour

Figure 4.12: détail de la ligne de frais.

Modifier une ligne:

Les utilisateurs peuvent modifier les détails d'une ligne de frais existante.



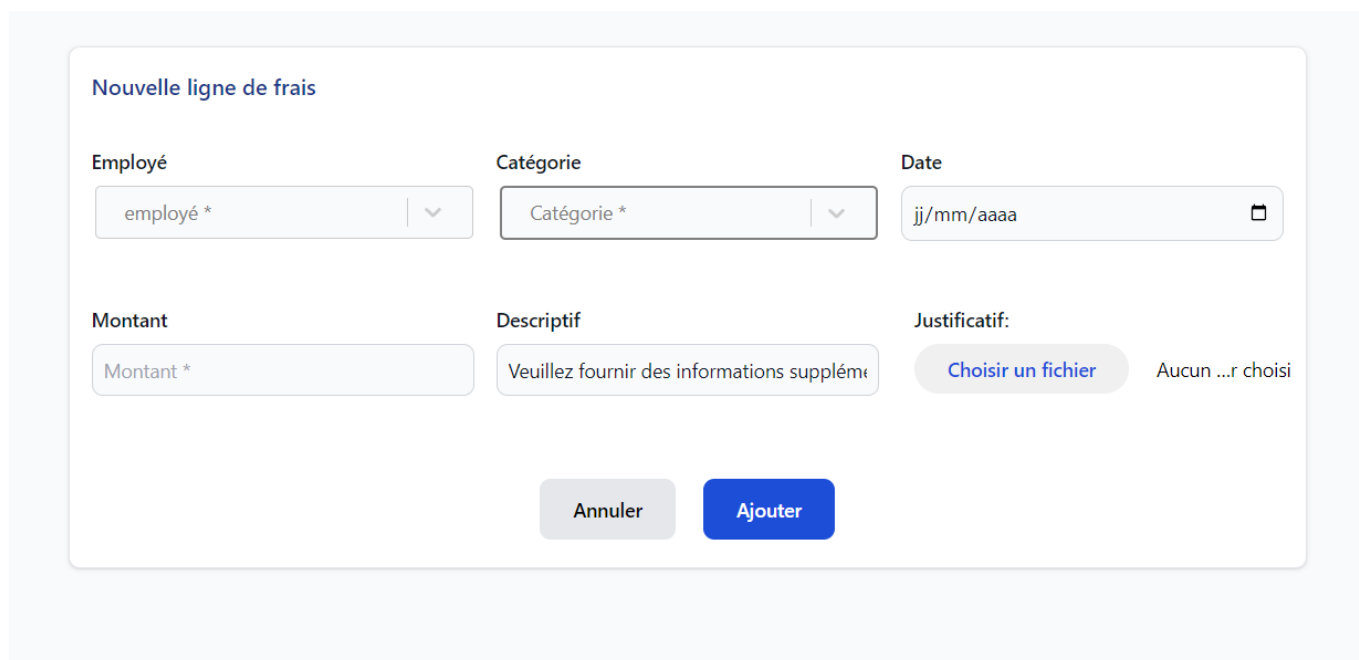
The screenshot shows a web application interface for modifying a line item. The top navigation bar includes a home icon, 'Accueil', and 'Notes des Frais'. The user 'Soukaina' is logged in. The main form is titled 'Modifier la ligne de frais' and contains the following fields:

- Employé:** A dropdown menu with 'Select...' and a downward arrow.
- Date:** A text input with '15/06/2023' and a calendar icon.
- Categorie:** A dropdown menu with 'Select...' and a downward arrow.
- Montant:** A text input with '7500'.
- Descriptif:** A text input with 'Frais de bureau'.
- Justificatif:** A button 'Choisir un fichier' and the text 'Aucun fichier choisi'.

At the bottom of the form are two buttons: 'Annuler' (outlined) and 'Modifier' (solid blue).

Figure 4.13: Liste des lignes de frais.

Ajouter une ligne:



The screenshot shows a web application interface for adding a new line item. The form is titled 'Nouvelle ligne de frais' and contains the following fields:

- Employé:** A dropdown menu with 'employé *' and a downward arrow.
- Catégorie:** A dropdown menu with 'Catégorie *' and a downward arrow.
- Date:** A text input with 'jj/mm/aaaa' and a calendar icon.
- Montant:** A text input with 'Montant *'.
- Descriptif:** A text input with 'Veuillez fournir des informations supplém'.
- Justificatif:** A button 'Choisir un fichier' and the text 'Aucun ...r choisi'.

At the bottom of the form are two buttons: 'Annuler' (outlined) and 'Ajouter' (solid blue).

Figure 4.14: Ajout d'une ligne.


4.1.3.3 Le module des catégories de frais


Ce module permet aux utilisateurs de gérer les catégories de frais. Les catégories aident à organiser et à classer les différentes dépenses.

Liste des catégories de frais:

Les utilisateurs peuvent visualiser toutes les catégories de frais existantes. Chaque catégorie peut être modifiée ou supprimée si nécessaire.

Les types des documents

 Ajouter

 Chercher

Type de frais	Description	
Transport	Frais liés aux déplacements	 
Hébergement	Frais liés à l'hébergement	 
Repas	Frais liés aux repas d'affaires et aux repas quotidiens	 
Communications	Frais de téléphone portable internet et fax	 
Autres	toutes autres frais	 

Figure 4.15: Liste des catégories de frais.

4.1.3.4 Statistiques des dépenses

Cette fonctionnalité fournit des statistiques et des analyses sur les dépenses. Les utilisateurs peuvent voir des graphiques et des rapports qui résument les dépenses par catégorie, par période ou par montant total.

Graphes des notes de frais

Vous trouverez ici toutes les informations sur vos dépenses pour les analyser et les partager avec votre comptable

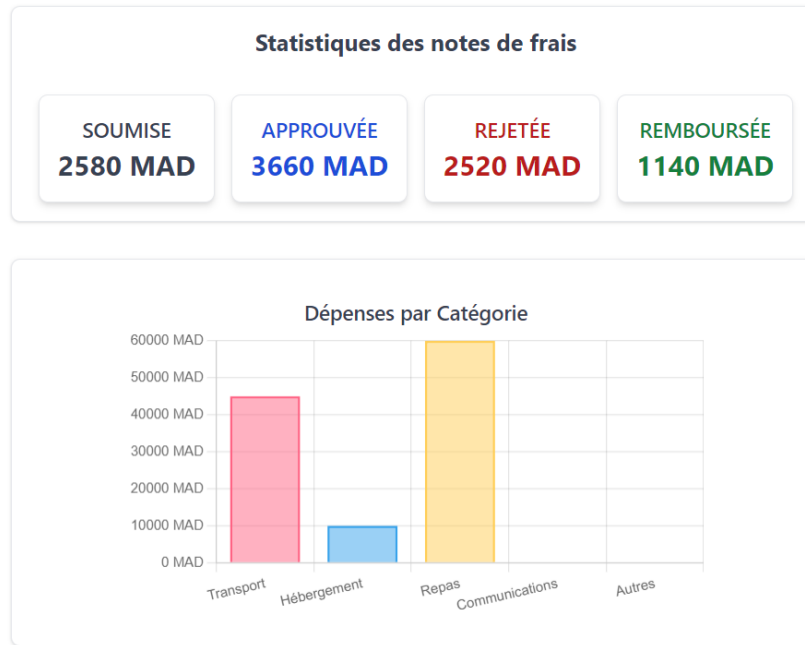


Figure 4.16: Statistiques des frais.

4.2 Validation de la solution

4.2.1 Validation de la solution par Tests Manuels

La validation de notre solution a été effectuée principalement à travers des tests manuels, une approche qui s'est révélée efficace pour assurer la qualité et la fonctionnalité de notre application de gestion des notes de frais. Bien que nous n'ayons pas utilisé de tests automatisés comme les tests unitaires ou d'intégration, nos tests manuels ont été minutieux et rigoureux.

4.2.1.1 Simulation de Scénarios d'Utilisation Réelle

Pour évaluer la robustesse de notre solution, nous avons simulé divers scénarios d'utilisation réelle. Cette approche proactive nous a permis de détecter et de corriger les éventuels défauts avant le déploiement final.

4.2.1.2 Tests des API REST

Parallèlement, nous avons testé les API REST pour vérifier leur conformité aux spécifications. Les résultats de ces tests ont été positifs, confirmant que les interactions entre le frontend et le backend se déroulent sans accroc. Chaque requête HTTP a été validée pour s'assurer que les données sont transmises correctement et que les réponses respectent les attentes en termes de structure et de contenu.

4.2.1.3 Intégration Backend-Frontend

Nous avons confirmé que les modifications apportées aux données côté backend étaient correctement reflétées dans les tableaux et interfaces du frontend. Ce processus a été crucial pour assurer une expérience utilisateur fluide et en temps réel.

4.2.2 Résultats

Même sans recourir à des tests automatisés tels que les tests unitaires ou d'intégration, notre méthode de tests manuels s'est révélée très efficace. Elle a permis de livrer une application de gestion de notes de frais fonctionnelle, fiable et conforme aux attentes. Les résultats obtenus démontrent notre engagement à fournir une solution de haute qualité, garantissant une expérience utilisateur optimale.

Conclusion

Nous avons démontré la réussite du projet en présentant les interfaces utilisateur et en assurant que l'ensemble du travail accompli garantit que l'application est conforme aux attentes.

Conclusion et Perspectives

Le présent projet de fin d'études porte sur l'étude, la conception et l'implémentation d'une application de gestion de notes de frais. Pour atteindre les objectifs initialement fixés, nous avons commencé par une étude préalable qui nous a permis de comprendre ce qui existe déjà et d'apprendre les principaux concepts autour desquels tourne notre projet. Ensuite, nous avons procédé à une analyse détaillée pour spécifier nos exigences fonctionnelles et non fonctionnelles. Cette phase d'analyse a été suivie par la conception des différents diagrammes UML, ce qui nous a aidé à structurer notre application de manière claire et cohérente. Enfin, nous avons défini une architecture opérationnelle et sélectionné les technologies les plus appropriées avant de passer à l'étape de développement de la solution.

Les objectifs primaires ont été atteints. Cependant, ce succès n'a pas été sans défis. Nous avons été confrontés à la complexité inhérente à la conception d'un système de gestion de notes de frais en l'intégrant dans ce qui existe déjà, nécessitant une analyse et une conception approfondies. Nous avons ajouté des fonctionnalités telles que la création, la modification et la visualisation des notes de frais et des lignes de frais en catégorisant les dépenses. Nous avons également intégré des méthodes de visualisation et des graphiques pour une meilleure analyse des dépenses. De plus, nous avons dû nous familiariser avec diverses technologies telles que React, 4D et Git, ce qui a représenté un travail substantiel accompli avec succès.

Sur le plan professionnel, ce projet nous a permis de compléter les notions théoriques acquises durant notre formation à l'INPT et de développer une expertise technique pointue. En résumé, cette expérience s'est révélée très enrichissante et nous prépare efficacement pour une insertion facile et réussie dans le monde professionnel.

Sur le plan personnel, cette période de développement a été une opportunité précieuse pour nous épanouir et développer nos compétences en communication et en relations interpersonnelles. Nous avons pu affiner notre capacité à travailler en équipe et à gérer des projets complexes, ce

qui constituera un atout majeur dans notre future carrière.

Notre application, bien qu'encore en phase de développement, offre de nombreuses perspectives d'amélioration. Nous envisageons l'ajout de fonctionnalités complémentaires à l'interface web, comme des modules de gestion avancée des dépenses et des options de personnalisation pour les utilisateurs. L'implémentation de systèmes de paiement intégrés pourrait simplifier davantage les processus de remboursement.

L'intégration des technologies de machine learning pour offrir des recommandations personnalisées et des analyses prédictives est aussi une possibilité, améliorant ainsi l'expérience utilisateur et la prise de décision ainsi que l'amélioration des tableaux de bord pour une visualisation plus intuitive.

Acronymes

API	Application Programming Interface
CSS	Cascading Style Sheets
DBMS	Database Management System
EDI	Integrated Development Environment
GUI	Software-Defined Network
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
ORDA	Object Relational Data Access
PGSQL	PostgreSQL
REST	Representational State Transfer
SQL	Structured Query Language
XML	Extensible Markup Language

Bibliography

- [1] *4D site officiel*, <https://fr.4d.com/>.
- [2] *What is Skype*, <https://fr.wikipedia.org/wiki/Skype>
- [3] *React*, <https://reactjs.org/>
- [4] *Axios*, <https://axios-http.com/>
- [5] *Chart.js*, <https://www.chartjs.org/>
- [6] *Why Gitlab*, <https://about.gitlab.com/why-gitlab/>
- [7] *UML : Définitions et objectifs*, <https://commentouvrir.com/info/>
- [8] *Les diagrammes UML*, <https://cours.khalilmamouny.com>
- [9] *Vs code*, <https://code.visualstudio.com/>
- [10] *4D Client*, <https://file.org/free-download/4d-client>
- [11] *Stack Overflow survey*, <https://survey.stackoverflow.co/2023/>
- [12] *State Of Dev In Morocco*, <https://stateofdev.ma/technology>
- [13] *Étude D'indice TIOBE*, <https://www.tiobe.com/tiobe-index/>
- [14] *Classement des frameworks python*, <https://www.jetbrains.com/lp>
- [15] *Classement des SGBD avec Django*, <https://lp.jetbrains.com>
- [16] *Django avec PostreSQL*, <https://www.enterprisedb.com/postgres-tutorials/>
- [17] *Environnement 4D*, <https://fr.4d.com/developpement>
- [18] *TailwindCss*, <https://tailwindcss.com/>