



UNIVERSITÉ DE TEK-UP

Kdayes

Auteur :

Mehrez OUABI

Superviseur :

Mr. KarimHAMROUNI

11 novembre 2024

Table des matières

1	Étude préliminaire	1
1.1	Présentation générale de CodeCooperation	2
1.2	Secteur d'activité	2
1.3	Contexte et Définition du Projet	3
1.3.1	Contexte du Projet	3
1.3.2	Objectifs du Projet	4
1.4	Étude et Diagnostic de l'Existant	5
1.4.1	Problématique	5
1.4.2	Solutions Existantes	6
1.4.3	Analyse des Limites des Solutions Existantes	6
1.4.4	Solution Proposée	7
1.5	Méthodologie adoptée	8
1.5.1	Présentation du cadre du projet	8
1.5.2	Présentation du Scrum	9
1.5.3	Les rôles Scrum	10
2	Analyse des besoins	11
2.1	Spécification des besoins	12
2.1.1	Description fonctionnelle des besoins	12

2.1.2	Description non fonctionnelle des besoins	14
2.2	Backlog	17
2.3	Diagramme de cas d'utilisation	21
2.3.1	Identification des acteurs	21
2.3.2	Le diagramme de cas d'utilisation	22
2.4	Diagramme de classe	22
2.4.1	Classes principales	24
2.4.2	Explication du diagramme de classe	24
2.4.3	Flux de Travail	25
3	Sprint 0 Installation d'environnement logiciel et matériel	27
3.1	Environnement matériel	28
3.2	Environnement logiciel	28
3.2.1	Développement Backend	29
3.2.2	Développement Frontend	29
3.2.3	Base de données	29
3.2.4	Page de destination	30
3.2.5	Tests API	30
3.2.6	Documentation OpenAPI	31
3.2.7	Gestion de projet et gestion de versions	31
4	Développement du Sprint 1	33
4.1	Planification du Sprint	34
4.2	Backlog du Sprint 1	35
4.3	Capture des besoins	37

4.4	Analyse	40
4.4.1	Diagramme de cas d'utilisation pour Sprint 1	40
4.5	Les interfaces graphiques	40
4.6	Analyse	40
4.6.1	Diagramme de séquence pour Sprint 1	40
4.7	Les interfaces graphiques	41
5	Développement du Sprint 2	45
5.1	Planification du Sprint	46
5.2	Backlog du Sprint 2	47
5.3	Capture des besoins	50
5.4	Analyse	52
5.4.1	Diagramme de séquence pour Invoice Sending	52
5.5	Les interfaces graphiques	52
5.5.1	Création d'une Facture	52
5.5.2	Mise à Jour d'une Facture	57
5.5.3	Vue Détail d'une Facture	58
6	Développement du Sprint 3	60
6.1	Planification du Sprint	61
6.2	Backlog du Sprint 3	62
6.3	Capture des besoins	63
6.4	Les interfaces graphiques	65
6.4.1	Gestion du profil utilisateur	65
6.4.2	Tableau de bord	65

7 Développement du Sprint 4	67
7.1 Planification du Sprint	68
7.2 Backlog du Sprint 4	68
7.3 Capture des besoins	71
7.4 Analyse	73
7.4.1 Diagramme de séquence pour les paiements	73
7.5 Interfaces Graphiques	74
Conclusion générale	79

Table des figures

1.1	Exemple pour tableau de gestion de projet sur GitHub	9
2.1	Le diagramme de cas d'utilisation (zoom pour voir les details)	23
2.2	Le diagramme de classe (zoom pour voir les détails)	25
4.1	Diagramme de cas d'utilisation de sprint 1	40
4.2	Diagramme de séquence d'authentification	41
4.3	Entrée de l'email de l'utilisateur pour l'authentification	42
4.4	Écran de connexion de l'utilisateur à la plateforme	42
4.5	Redirection vers la page de complétion du profil lorsque le score est inférieur à 40%	43
4.6	Saisie des informations sur l'entreprise	43
4.7	Saisie des informations bancaires pour compléter le profil	44
5.1	Diagramme de séquence pour Invoice Sending	52
5.2	Entrée des informations d'en-tête de la facture	53
5.3	Saisie des informations personnelles de l'utilisateur	53
5.4	Ajout des articles ou services facturés	54
5.5	Définition des préférences de paiement	54
5.6	Activation des notifications pour rappel de paiement	55
5.7	Génération du lien de paiement sécurisé avec QR code	55

5.8	Sélection de la date limite de paiement dans le pop-up	56
5.9	Tableau de bord des factures permettant de vérifier l'ajout de la facture .	56
5.10	Bouton pour mettre à jour une facture	57
5.11	Mise à jour des informations de la facture	57
5.12	Vue détaillée d'une facture avec toutes les informations	58
5.13	Vue détaillée d'une facture	59
6.1	Vue d'ensemble du tableau de bord	66
7.1	Diagramme de séquence pour les paiements	73
7.2	Notification par e-mail avec bouton de paiement	74
7.3	Interface de visualisation de la facture avec option de paiement	75
7.4	Page de paiement par carte bancaire	75
7.5	Tableau de bord des paiements réussis	76
7.6	Vue des factures et des rapports financiers	77
7.7	Vue des factures et des rapports financiers	77

Liste des tableaux

1.1	L'équipe de travail selon le Framework SCRUM	10
2.2	Les acteurs principaux du projet	21
3.1	Spécifications des ordinateurs utilisés pour le projet	28

Introduction générale

Ce rapport de stage détaille le parcours de développement de Kdayes, un MVP destiné à devenir un produit fondamental pour une startup. Conçu comme une plateforme tout-en-un spécialement adaptée aux freelances tunisiens, Kdayes répond aux défis majeurs qu'ils rencontrent dans la gestion de leur facturation, de leurs paiements, de leurs obligations fiscales et de leur conformité réglementaire. Avec l'expansion rapide du marché des freelances en Tunisie, Kdayes se positionne comme un outil clé pour autonomiser les freelances en leur offrant des solutions administratives et financières essentielles, à la fois faciles à utiliser et fortement localisées.

L'objectif de ce stage n'était pas seulement le développement technique, mais également les étapes fondamentales pour créer une entreprise prospère et scalable. Dans cette optique, mon stage s'est concentré sur la traduction des besoins réels des freelances en une solution numérique robuste, en utilisant les méthodologies agiles, spécifiquement le cadre Scrum. Chaque sprint a ajouté de la valeur à Kdayes, assurant ainsi l'alignement avec les objectifs commerciaux et les attentes des utilisateurs. Ce rapport de stage reflète l'évolution de Kdayes depuis son concept jusqu'au MVP, détaillant son périmètre, son design, son développement et son potentiel futur en tant que produit de startup.

ÉTUDE PRÉLIMINAIRE

Introduction

Dans ce chapitre, nous commencerons par présenter l'organisme d'accueil ainsi que le service dans lequel nous avons effectué notre stage. Ce chapitre a pour objectif de situer le projet dans son contexte général. Nous commencerons par décrire l'organisme d'accueil, puis nous exposerons la problématique liée à notre projet. Enfin, nous détaillerons la méthodologie adoptée pour mener à bien ce travail.

1.1 Présentation générale de CodeCooperation

CodeCooperation est un laboratoire d'innovation numérique situé à Berlin et Tunis, spécialisé dans la création de produits numériques pour les startups, les scale-ups et les PME. L'entreprise adopte une approche agile, avec un état d'esprit de startup, pour concevoir et développer des solutions numériques sur mesure qui répondent aux besoins spécifiques des entreprises. Grâce à son équipe d'ingénieurs entrepreneurs issus de divers horizons, CodeCooperation est en mesure d'intégrer des processus d'innovation numérique, de développement de logiciels et d'applications mobiles, tout en offrant des services d'assurance qualité. Les bureaux de l'entreprise sont conçus pour encourager la créativité et la collaboration, permettant ainsi un environnement propice à l'innovation continue.

1.2 Secteur d'activité

CodeCooperation évolue dans le secteur de l'innovation numérique, avec une spécialisation dans le développement de produits logiciels, la transformation numérique des entreprises et l'intégration d'une mentalité agile dans les organisations. L'entreprise se distingue par son expertise dans l'accompagnement des startups et des PME dans leurs projets technologiques, en offrant des services de développement de logiciels, de création

d'applications mobiles, et de développement de sites web. CodeCooperation se positionne comme un acteur clé dans la transformation digitale des entreprises, en leur fournissant les outils nécessaires pour innover et se développer rapidement tout en respectant une méthodologie agile qui favorise la réactivité et l'adaptation.

1.3 Contexte et Définition du Projet

1.3.1 Contexte du Projet

Le freelancing en Tunisie est en forte croissance, alimenté par l'évolution des tendances mondiales du travail à distance et la flexibilité qu'il offre aux travailleurs. De plus en plus de Tunisiens choisissent de devenir indépendants pour offrir leurs services dans des domaines variés tels que la programmation, le design, le marketing digital, la rédaction, et bien d'autres. Cependant, malgré cette expansion rapide, de nombreux freelances rencontrent des difficultés majeures qui freinent leur développement et leur réussite à long terme.

Les problèmes rencontrés par les freelances en Tunisie incluent principalement la désinformation concernant les aspects fiscaux et réglementaires de leur activité, la difficulté à créer des factures conformes aux normes locales, ainsi que des problèmes de gestion des paiements en ligne. Ces obstacles créent une instabilité dans leur gestion quotidienne et exposent les freelances à des risques financiers et juridiques.

À cela s'ajoute l'absence de ressources locales qui pourraient guider ces travailleurs dans leur parcours. Les freelances tunisiens ne trouvent souvent pas d'outils ou de plateformes spécifiquement adaptés aux exigences fiscales et administratives locales, ce qui complique la gestion de leur activité et leur développement professionnel.

Face à cette réalité, **Kdayes** a été imaginée comme une solution pour simplifier et structurer la gestion des freelances en Tunisie. Kdayes est une plateforme numérique tout-en-un, conçue pour répondre aux besoins uniques des freelances tunisiens, en facilitant

la création de factures conformes, la gestion des paiements en ligne, et en fournissant des ressources éducatives adaptées au contexte local.

1.3.2 Objectifs du Projet

L'objectif principal de **Kdayes** est de révolutionner la gestion administrative des freelances en Tunisie, en leur offrant une plateforme numérique simple, accessible et spécifiquement conçue pour répondre à leurs besoins. L'objectif est de permettre aux freelances de se concentrer sur leur cœur de métier tout en ayant l'assurance que leur gestion financière et administrative est conforme aux régulations locales, sécurisée, et simplifiée.

Les objectifs spécifiques de **Kdayes** sont les suivants :

- **Faciliter la gestion des factures** : Permettre aux freelances de créer des factures professionnelles facilement, qui respectent les normes fiscales tunisiennes, afin de garantir leur conformité avec les autorités fiscales locales.
- **Simplifier la gestion des paiements en ligne** : Offrir une solution simple et sécurisée pour accepter les paiements en ligne, facilitant ainsi les transactions entre les freelances et leurs clients, tout en assurant un suivi clair et transparent des paiements.
- **Fournir des ressources éducatives locales** : Proposer des guides, des articles et des vidéos éducatives pour aider les freelances à mieux comprendre leurs obligations fiscales et juridiques en Tunisie, et optimiser la gestion de leur activité.
- **Créer un outil intuitif et facile à utiliser** : Développer une plateforme qui ne nécessite pas de formation ou de compétences techniques avancées, afin que tout freelance, quel que soit son niveau d'expérience, puisse l'utiliser sans difficulté.
- **Construire une communauté forte et solidaire** : Favoriser un environnement où les freelances peuvent échanger, apprendre les uns des autres, et trouver des

solutions à leurs problèmes communs. L'objectif est de créer un espace de confiance et de partage pour soutenir leur réussite professionnelle.

En somme, le projet **Kdayes** vise à être la solution incontournable pour les freelances en Tunisie, en fournissant un outil complet et accessible pour gérer efficacement leur activité, tout en respectant les normes fiscales et réglementaires locales. À plus long terme, l'ambition est de positionner **Kdayes** comme la plateforme de référence pour les freelances dans la région MENA.

1.4 Étude et Diagnostic de l'Existant

1.4.1 Problématique

Les freelances en Tunisie font face à plusieurs problèmes majeurs dans leur parcours professionnel :

- **Informations fiscales et réglementaires inexactes ou insuffisantes** : Beaucoup de freelances n'ont pas accès à des informations fiables sur leurs obligations fiscales, ce qui les expose à des erreurs et des sanctions potentielles. Ce manque de transparence conduit à une méconnaissance des droits et obligations légales liés à leur statut de travailleur indépendant.
- **Problèmes de création et de conformité des factures** : La création de factures professionnelles conformes aux exigences fiscales tunisiennes est un défi pour beaucoup de freelances. Une facture incorrecte ou mal rédigée peut entraîner des complications légales, des retards de paiement, et des tensions avec les clients.
- **Gestion des paiements et de la trésorerie** : De nombreux freelances ne savent pas comment accepter ou gérer les paiements en ligne de manière sécurisée, ce qui peut entraîner des problèmes de liquidités, des retards de paiement, voire la perte de clients.

Ces problèmes sont amplifiés par le manque de solutions adaptées, et par l’absence d’un cadre juridique spécifique pour le travail indépendant en Tunisie.

1.4.2 Solutions Existantes

Actuellement, plusieurs solutions sont proposées aux freelances pour résoudre ces problèmes, mais elles restent insuffisantes ou inadaptées à leurs besoins spécifiques :

- **Solutions de facturation en ligne** : Des plateformes comme Zoho Invoice, FreshBooks, ou Wave existent, mais elles ne sont pas localisées pour le marché tunisien. Elles ne prennent pas en compte les spécificités fiscales et réglementaires tunisiennes, ce qui complique leur utilisation pour les freelances en Tunisie.
- **Systèmes de paiement en ligne** : Des solutions comme PayPal ou Stripe permettent aux freelances d’accepter des paiements en ligne. Cependant, ces plateformes ne sont pas toujours adaptées aux besoins des freelances tunisiens en termes de compatibilité avec les banques locales, de frais élevés, ou d’absence de régulation fiscale locale.
- **Plateformes éducatives** : Bien qu’il existe des ressources éducatives générales, il n’existe pas de plateforme localisée spécifiquement pour les freelances tunisiens. Les informations disponibles sont souvent trop générales et ne couvrent pas les aspects uniques du freelancing en Tunisie, comme la gestion des obligations fiscales locales.

1.4.3 Analyse des Limites des Solutions Existantes

Bien que certaines solutions globales existent, elles ne répondent pas entièrement aux besoins spécifiques des freelances en Tunisie :

- **Inadaptation aux spécificités locales** : Les solutions existantes ne sont pas conçues pour répondre aux exigences fiscales et juridiques tunisiennes, créant ainsi un fossé pour les freelances qui peinent à se conformer à la législation locale.
- **Manque d’intégration** : Les outils actuels sont souvent trop segmentés. Par

exemple, un freelance peut utiliser un service de facturation et un autre pour les paiements, mais ces plateformes ne sont pas intégrées. Cette fragmentation rend la gestion plus complexe et sujette à des erreurs humaines.

- **Complexité et coût** : Les plateformes existantes sont souvent trop complexes à utiliser, nécessitant des formations ou un investissement en temps significatif pour comprendre comment les exploiter efficacement. De plus, les frais associés à l'utilisation de certaines de ces plateformes peuvent être prohibitifs pour de nombreux freelances, notamment pour ceux avec des ressources limitées.
- **Absence de support localisé** : Les freelances tunisiens se retrouvent souvent sans réponse à leurs questions ou préoccupations en matière de fiscalité locale, de création de factures conformes, ou de gestion des paiements. Le manque de support spécifique à la Tunisie rend l'adoption de solutions internationales encore plus complexe.

1.4.4 Solution Proposée

****Kdayes**** se positionne comme une solution tout-en-un, pensée spécifiquement pour répondre aux besoins des freelances tunisiens. Elle aborde les problèmes existants par les éléments suivants :

- **Création de factures conformes à la réglementation tunisienne** : ****Kdayes**** permet aux freelances de créer facilement des factures professionnelles qui respectent les exigences fiscales locales, réduisant ainsi le risque de non-conformité.
- **Gestion des paiements en ligne** : La plateforme permet l'intégration de solutions de paiement en ligne adaptées aux freelances tunisiens, garantissant des transactions sécurisées et une gestion simplifiée de la trésorerie.
- **Suivi des projets et des revenus** : ****Kdayes**** offre des fonctionnalités intuitives pour suivre les projets, gérer les revenus et maintenir une vue d'ensemble claire de la situation financière de chaque freelance.

- **Ressources éducatives locales** : La plateforme propose des ressources éducatives spécifiques au marché tunisien, permettant aux freelances d'acquérir une meilleure compréhension de leurs obligations fiscales et juridiques, et de gérer efficacement leur activité.
- **Simplicité et adaptabilité** : **Kdayes** est conçu pour être simple à utiliser, sans nécessiter de formation complexe ou de configuration avancée. Cela le rend accessible même aux freelances avec peu d'expérience en gestion d'entreprise.

Ainsi, **Kdayes** est une réponse aux besoins spécifiques des freelances tunisiens, offrant une solution simple, complète et localisée pour faciliter leur gestion administrative et financière.

1.5 Méthodologie adoptée

Pour différents types de projets, l'utilisation d'une méthodologie agile est l'option la plus populaire. La base de la méthodologie agile est un cycle de développement qui met l'accent sur la communication avec le client. L'implication du client dans le projet lors de sa réalisation permet toujours d'avoir une vision plus claire et de valoriser le travail au fur et à mesure de sa réalisation.

1.5.1 Présentation du cadre du projet

En raison de son adéquation avec nos besoins et de notre volonté de répondre aux attentes des clients tout en maximisant les chances de succès du projet dans un environnement de travail collaboratif, nous avons choisi d'adopter la méthode Scrum comme méthodologie de gestion. Par ailleurs, nous tirerons parti des tableaux de gestion de projet sur GitHub pour organiser, suivre et coordonner les différentes tâches et étapes du projet.

1.5.2 Présentation du Scrum

Ce modèle est le plus adapté aux sociétés ; il repose sur des réunions appelées « Daily Scrum Meeting (DSM) » qui durent 15 minutes chaque jour, dans le but d'identifier les problèmes rencontrés et de montrer ce qui s'est passé hier et ce qui reste à faire. Le cycle de vie Scrum est rythmé par des itérations de deux à quatre semaines. À chaque début du sprint, des réunions appelées « Sprint Planning » se réalisent. En effet, le sprint planning consiste à préciser une liste des tâches pour le prochain sprint, cette liste représente le Product Backlog. À la fin de chaque sprint, toute l'équipe se réunit et prend en général 15 à 30 minutes pour discuter et voir ce qui marche et ce qui ne marche pas, c'est la rétrospective.

Les acteurs participants dans le cycle SCRUM sont :

- **Le propriétaire du produit (Product Owner)** : Il représente le client et il est en contact avec le Scrum Master.
- **Le Scrum Master** : Il est responsable de l'application quotidienne des directives Scrum, d'où il est en charge d'accorder un environnement de travail agréable pour l'équipe.
- **L'équipe de travail** : Se compose de deux à neuf personnes. Chaque membre doit identifier pendant le « Sprint Planning » les tâches qu'il a achevées et les tâches restantes dans un « Product Backlog » spécifique à chaque sprint.



FIGURE 1.1 : Exemple pour tableau de gestion de projet sur GitHub

1.5.3 Les rôles Scrum

Dans le cas de notre projet, nous présentons l'équipe de travail selon les rôles Scrum dans le tableau suivant :

Rôle	Description
Product Owner	Mahrez Ourabi
Scrum Master	Karim Hamrouni
Business Developor	Bilel Naffeti
Fullstack Developors	Mahrez Ourabi et Eya Ben Hmida

TABLEAU 1.1 : L'équipe de travail selon le Framework SCRUM

Conclusion

En conclusion, ce premier chapitre a établi le contexte du projet, identifié la problématique, proposé des solutions novatrices, et défini la méthodologie Scrum. Ces éléments constituent une base solide pour la suite du rapport, orientant notre analyse et la mise en œuvre du projet.

ANALYSE DES BESOINS

Introduction

Nous consacrons ce deuxième chapitre à la phase de spécification et d'analyse des besoins. C'est la première étape de la conception qui consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent, afin d'assurer un ouvrage ou un processus répondant aux besoins.

2.1 Spécification des besoins

La spécification des besoins constitue la première étape essentielle de la conception du système. Elle consiste à analyser les exigences pour identifier les contraintes techniques, les risques potentiels et les éléments clés qui garantiront le bon fonctionnement du système. L'objectif est de s'assurer que le projet répond aux besoins des utilisateurs et des parties prenantes, tout en respectant les contraintes du projet.

2.1.1 Description fonctionnelle des besoins

Les besoins fonctionnels décrivent les fonctionnalités et les actions que la plateforme *Kdayes* doit offrir pour répondre aux attentes des freelances en Tunisie. Ces besoins sont directement liés à la manière dont les utilisateurs vont interagir avec la plateforme et les services qu'elle propose.

2.1.1.1 1.1. Gestion des Factures

- **Création de Factures Conformes** : La plateforme doit permettre aux freelances de générer des factures qui respectent les normes fiscales tunisiennes, incluant les informations légales (nom, adresse, numéro de TVA, etc.), le détail des prestations, les montants, les taxes et les conditions de paiement.
- **Modèles de Factures (de vente/d'achat)** : L'utilisateur doit pouvoir choisir

parmi plusieurs modèles de facture préconfigurés, personnalisables en fonction de son secteur d'activité.

- **Gestion des Factures Électroniques** : Les freelances doivent pouvoir envoyer des factures électroniques directement à leurs clients, avec la possibilité d'ajouter des fichiers annexes (par exemple, des contrats ou des rapports).
- **Historique des Factures** : La plateforme doit conserver un historique des factures créées, permettant aux utilisateurs de consulter, modifier ou réémettre des factures.

2.1.1.2 1.2. Gestion des Paiements en Ligne

- **Acceptation des Paiements** : La plateforme doit permettre aux freelances d'accepter des paiements en ligne via des systèmes sécurisés, comme PayPal, Stripe, ou des solutions locales adaptées aux spécificités tunisiennes.
- **Suivi des Paiements** : Chaque paiement effectué doit être enregistré dans un tableau de bord financier, avec une visibilité sur les montants reçus, les délais de paiement, et le statut des transactions.
- **Notifications de Paiement** : Les freelances doivent recevoir des notifications lorsque des paiements sont effectués ou lorsqu'une échéance approche.

2.1.1.3 1.3. Gestion des Projets

- **Rapports de Performance** : Les freelances doivent avoir la possibilité de générer des rapports sur l'état de leurs projets, ou les revenus.
- **Gestion de la Trésorerie** : La plateforme doit permettre de visualiser les revenus générés par chaque projet et d'assurer un suivi global de la trésorerie, afin de prévenir toute difficulté financière.

2.1.1.4 1.4. Ressources Éducatives

- **Guides et Tutoriels** : La plateforme doit fournir des ressources éducatives sous forme de guides, articles et vidéos, expliquant les obligations fiscales, les bonnes pratiques en matière de facturation, ainsi que des conseils pour réussir en freelance en Tunisie.
- **FAQ Interactive** : Une section FAQ doit être disponible pour répondre aux questions les plus courantes concernant la gestion administrative et financière des freelances en Tunisie.
- **Mise à Jour des Ressources** : Les ressources doivent être régulièrement mises à jour pour refléter les évolutions législatives et fiscales locales.

2.1.1.5 1.5. Création d'une Communauté

- **Forum ou Espace de Discussion** : La plateforme doit permettre aux freelances de poser des questions, échanger des conseils, et partager des bonnes pratiques à travers un forum ou une plateforme de discussion.
- **Événements et Webinaires** : Organiser des événements en ligne, des webinaires ou des formations pour aider les freelances à se perfectionner et à échanger des idées.

2.1.2 Description non fonctionnelle des besoins

Les besoins non fonctionnels se concentrent sur la qualité, la performance, la sécurité et l'expérience utilisateur de la plateforme. Ces besoins garantissent que *Kdayes* répondra aux exigences de performance et de fiabilité tout en offrant une expérience optimale pour les utilisateurs.

2.1.2.1 2.1. Performance et Scalabilité

- **Temps de Réponse :** La plateforme doit avoir un temps de réponse rapide (moins de 2 secondes pour chaque action) pour offrir une expérience fluide aux utilisateurs.
- **Scalabilité :** La plateforme doit pouvoir supporter une croissance rapide du nombre d'utilisateurs sans dégradation de la performance. Elle doit être capable d'ajouter de nouvelles fonctionnalités ou de nouveaux modules sans affecter la performance globale.
- **Capacité de Traitement des Transactions :** La plateforme doit pouvoir gérer simultanément plusieurs milliers de transactions de paiement, tout en maintenant une haute disponibilité.

2.1.2.2 2.2. Sécurité

- **Cryptage des Données Sensibles :** Toutes les données sensibles (notamment les informations bancaires et fiscales) doivent être cryptées à l'aide des standards de sécurité modernes (SSL, TLS, etc.).
- **Authentification et Autorisation :** La plateforme doit offrir des mécanismes de sécurité robustes pour l'authentification des utilisateurs (par exemple, l'authentification à deux facteurs) et gérer des niveaux d'autorisation pour restreindre l'accès à certaines fonctionnalités en fonction des rôles des utilisateurs.
- **Protection des Données Personnelles :** La plateforme doit se conformer aux lois locales et internationales en matière de protection des données personnelles, comme la loi sur la protection des données en Tunisie (Loi n° 2004-63) et les réglementations GDPR de l'Union européenne si applicable.
- **Audit et Logs :** Un système de journalisation (logs) doit être en place pour assurer un suivi des actions des utilisateurs, permettant de détecter toute activité suspecte ou anormale.

2.1.2.3 2.3. Accessibilité et Utilisabilité

- **Interface Intuitive** : L'interface de la plateforme doit être simple, claire, et facile à utiliser, même pour les utilisateurs sans compétences techniques ou administratives. Il est important que les utilisateurs puissent accomplir leurs tâches sans avoir besoin de formation préalable.
- **Multilinguisme** : La plateforme doit être disponible en plusieurs langues, dont le français, l'arabe et l'anglais, pour répondre aux besoins des freelances tunisiens et internationaux.
- **Compatibilité Mobile** : La plateforme doit être responsive et accessible depuis tous les types de dispositifs (smartphones, tablettes, ordinateurs de bureau). L'expérience utilisateur doit être optimisée pour chaque type d'appareil.

2.1.2.4 2.4. Fiabilité et Disponibilité

- **Haute Disponibilité** : La plateforme doit être disponible 24h/24, 7j/7, avec un taux de disponibilité de 99,9% ou plus. Des mécanismes de redondance et de sauvegarde doivent être mis en place pour garantir la continuité des services.
- **Surveillance et Maintenance** : Des outils de surveillance doivent être utilisés pour suivre la performance du système en temps réel, et des interventions de maintenance préventive doivent être réalisées régulièrement pour éviter les pannes.
- **Plan de Continuité de Service** : Un plan de reprise après sinistre (disaster recovery) doit être en place pour garantir la récupération des données et la remise en service rapide en cas de problème majeur.

2.1.2.5 2.5. Support et Assistance

- **Support Client** : Un support client accessible par email ou téléphone doit être disponible pour aider les freelances en cas de problème technique ou de question

liée à la plateforme.

- **Base de Connaissances** : La plateforme doit offrir une base de connaissances complète, permettant aux utilisateurs de trouver des réponses à leurs questions courantes sans avoir besoin de contacter le support.

2.2 Backlog

Avant même de démarrer le premier sprint, nous avons besoin d'un backlog du produit, c'est-à-dire une liste priorisée de caractéristiques orientées client.

Le backlog du produit existe tout au long de la vie du produit. C'est la feuille de route du produit. Tout au long du projet, le backlog du produit centralise la liste de « tout ce qui peut être fait par l'équipe, par ordre de priorité ». Il n'existe qu'un seul backlog pour un produit.

Pour estimer la complexité de nos user stories, nous avons utilisé les valeurs suivantes :

- 2 : facile
- 3 : simple
- 5 et 8 : normal
- 13 : difficile
- 21 : très difficile

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Complexité
1	User Registration and Login	Users can sign in using a valid email, receive an authentication email, and access the platform upon successful login.	Haute	8

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Complexité
2	Profile Completion Progress Indicator	Users see a progress indicator (e.g., "Your profile is 40% complete"), with prompts to add remaining details.	Moyenne	10
3	User Information Collection	Users can enter and save personal info, company details, and bank information; changes can be saved and updated easily.	Haute	12
4	Profile Picture Upload	Users can upload, view, and update their profile picture. A default image displays if no picture is uploaded.	Moyenne	8
5	Dashboard Overview	Users see a summary of recent payments, outstanding invoices, and general account insights on the home page.	Haute	10
6	Sidebar Navigation	Sidebar links direct users to respective sections, with active styling indicating the current page.	Moyenne	6
7	Invoice Creation	Users can create invoices, input client details, set due dates, and add service/item descriptions.	Haute	12
8	Invoice Editing	Users can modify invoice details (client, amount, due date) and save changes.	Haute	10

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Complexité
9	Invoice View and Side View	Users can click an invoice card to open a detailed side view, which allows viewing or updating information.	Moyenne	8
10	Invoice Sending	Users can send invoices to clients, with options to add a payment link and set automated reminders.	Haute	10
11	Bill Creation and Management	Users can generate new bills, view all bills, and edit them as necessary.	Haute	10
12	Online Payment via Konnect	Clients receive a link to securely pay invoices through Konnect, supporting online payment processing.	Haute	14
13	Payment Status Updates	Invoice status changes to "Paid" once payment is confirmed, with an optional email notification to the user.	Haute	8
14	Automatic Payment Reminders	Users can specify reminder intervals, and clients receive reminders if payment remains outstanding.	Moyenne	10
15	Invoice and Payment Reporting	Users can view and export reports by status, time range, or client, with visual summaries and detailed data.	Haute	14
16	Revenue and Expense Analytics	Users can view graphical insights into revenue, expenses, and trends, with filtering options.	Haute	16

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Complexité
17	Notification Preferences	Users can enable or disable email notifications for actions such as invoice payment, overdue reminders, and invoice views by clients.	Moyenne	6
18	Customizable Invoice Stamp	Users can upload a stamp image to appear on invoices, with optional size and positioning settings.	Basse	8
19	Help and Updates Section	Users can access the Help page, read blog posts, and find community news and freelance legal advice.	Basse	8
20	User Subscription Status	Users can view their subscription type, days remaining (if trial), and a renewal prompt if applicable.	Basse	6
21	Logout	Users can safely log out from any device, returning to the sign-in page upon logout.	Moyenne	4

Où :

- **ID** : représente l'identifiant de l'user story.
 - **Feature** : permet de mieux ordonner les user stories.
 - **User Story** : comporte la description des user stories suivant la forme « En tant que ... Je veux ... ».
 - **Priorité** : de l'user story selon la valeur métier et l'ordre de réalisation.
 - **Complexité** : de la réalisation de chaque user story selon la suite de Fibonacci.
- Cette dernière est une suite mathématique dans laquelle chaque terme est la somme des deux termes précédents, en considérant les deux termes initiaux 0 et 1. Le début

de cette suite est : 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 jusqu'à l'infini.

2.3 Diagramme de cas d'utilisation

Dans le cadre de ce projet, l'objectif est de mettre en place une plateforme de gestion de facturation dédiée aux freelances, incluant des fonctionnalités de gestion des paiements, de reporting financier, et de gestion des notifications. Le système s'adresse principalement aux Freelancers, aux Clients, et à un rôle d'Admin chargé de superviser et gérer les paramètres globaux de la plateforme. Chaque acteur joue un rôle essentiel pour assurer un service fluide et sécurisé pour l'ensemble des utilisateurs. Ce diagramme de cas d'utilisation vise à illustrer l'interaction entre les acteurs principaux et les fonctionnalités clés du système.

2.3.1 Identification des acteurs

TABEAU 2.2 : Les acteurs principaux du projet

Acteur	Rôle
Freelancer	Utilisateur principal du système, responsable de la création et de la gestion des factures, de l'analyse des revenus et des dépenses, et de l'envoi de notifications aux clients.

Client	Utilisateur qui reçoit et paie les factures envoyées par le freelancer, avec possibilité de consulter l'historique des factures et d'effectuer des paiements sécurisés.
Admin	Gère les utilisateurs et configure les paramètres du système, et a un accès à des rapports globaux de la plateforme.

2.3.2 Le diagramme de cas d'utilisation

Dans ce diagramme, nous présentons les acteurs ainsi que les cas d'utilisation qui sont liés à chaque acteur, tout en montrant les relations entre ces cas d'utilisation eux-mêmes.

2.4 Diagramme de classe

Le diagramme de classe représente les entités et leurs relations au sein du système. Ce modèle est structuré autour des fonctionnalités principales : gestion des utilisateurs (Freelancer et Client), création et envoi de factures, suivi des paiements, et génération de rapports financiers. Les classes principales permettent de structurer les informations et les relations, telles que l'association entre un Freelancer et ses Factures, ou entre un Client et ses paiements, pour assurer une expérience utilisateur fluide et cohérente.

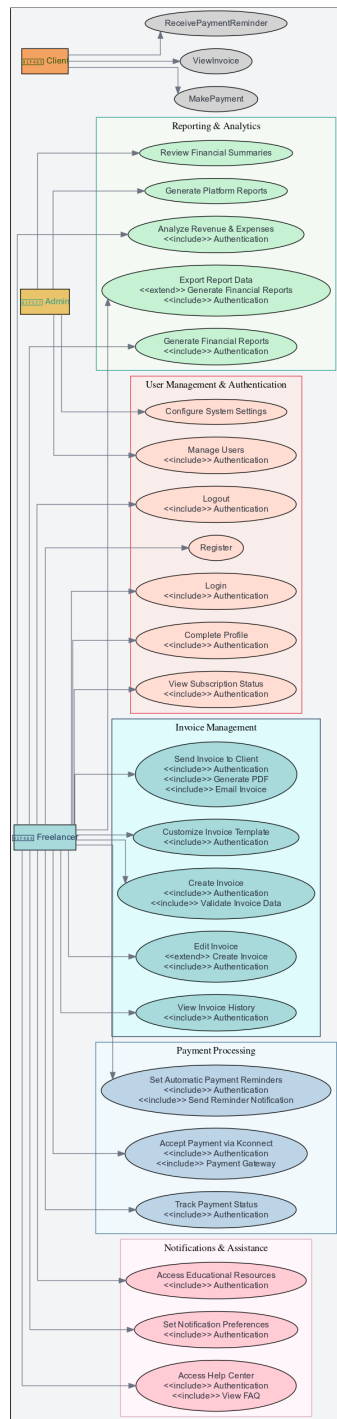


FIGURE 2.1 : Le diagramme de cas d'utilisation (zoom pour voir les details)

2.4.1 Classes principales

- **Freelancer** : Classe représentant les utilisateurs freelancers avec des attributs comme `fullName`, `email`, et des méthodes pour créer et gérer les factures et les notifications.
- **Client** : Classe dédiée aux utilisateurs clients avec des informations de contact et des méthodes d'interaction pour visualiser et payer les factures.
- **Facture (Invoice)** : Représente une facture avec des attributs tels que `issueDate`, `dueDate`, `total`, et des méthodes pour la création, la modification, et l'envoi.
- **Paiement (Payment)** : Classe gérant les paiements des factures avec `transactionDate`, `amount`, et des méthodes pour suivre l'état de paiement et effectuer des remboursements si nécessaire.
- **Rapport (Report)** : Classe utilisée pour produire des rapports financiers avec des méthodes pour l'exportation et la visualisation des données.
- **Notification** : Gère les préférences de notification, permettant aux freelancers de configurer des rappels automatiques.

2.4.2 Explication du diagramme de classe

- **Freelancer et Facture** : Un Freelancer peut créer plusieurs factures, chacune étant associée à un client. Les méthodes de cette classe permettent au freelancer de gérer efficacement ses activités de facturation.
- **Client et Paiement** : Les clients peuvent consulter et régler les factures via un portail de paiement intégré. Une relation entre **Client** et **Payment** permet de suivre les paiements des factures, avec des statuts mis à jour automatiquement.
- **Freelancer et Rapport** : Le Freelancer peut générer des rapports financiers basés sur ses données de facturation et de paiement. Ces rapports sont essentiels pour une analyse approfondie des performances financières.

- **Notification** : Cette classe permet de configurer des alertes automatiques, essentielles pour rappeler aux utilisateurs des actions importantes (paiements, échéances, etc.).

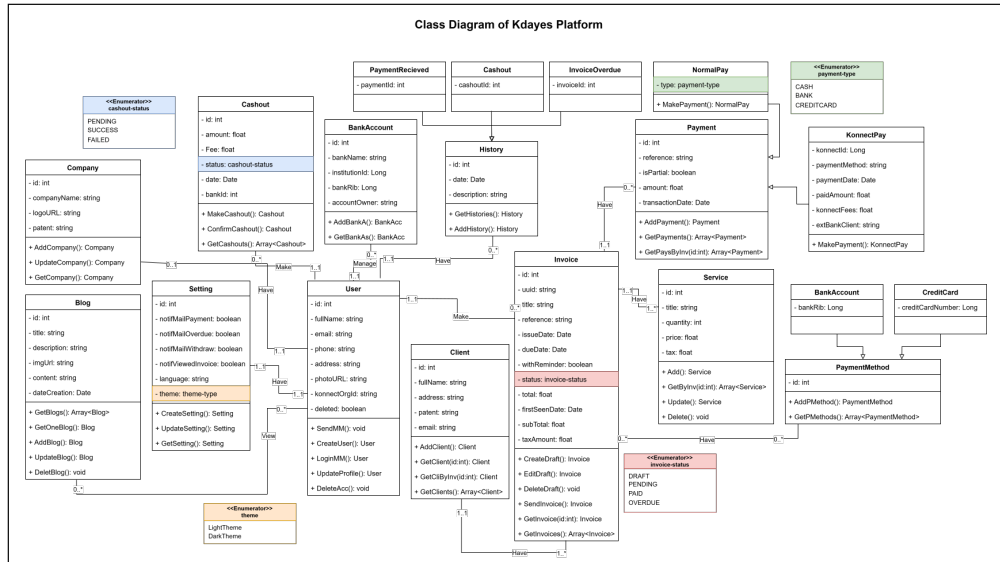


FIGURE 2.2 : Le diagramme de classe (zoom pour voir les détails)

2.4.3 Flux de Travail

- **Inscription et Connexion** : Un utilisateur Freelancer s'inscrit avec ses informations personnelles et complète son profil.
- **Gestion des Factures** : Le Freelancer peut créer une facture, la personnaliser, et l'envoyer au Client, avec des rappels si la date de paiement approche.
- **Traitement des Paiements** : Les Clients visualisent et paient les factures reçues. Le statut des factures est mis à jour automatiquement.
- **Rapports et Analyse** : Les Freelancers génèrent des rapports de revenus et dépenses pour une analyse financière approfondie.
- **Notification et Assistance** : Les utilisateurs configurent leurs préférences de notification pour être informés des événements clés via des alertes par e-mail ou SMS.

Conclusion

Ce chapitre a posé les bases de notre plateforme de facturation freelance en détaillant les besoins fonctionnels et non fonctionnels. L'analyse des cas d'utilisation, du diagramme de classes, et du backlog nous a permis de structurer efficacement les interactions et les priorités du projet. Ces éléments fournissent une vue claire des fonctionnalités et assurent une organisation cohérente pour le développement à venir. Nous sommes ainsi prêts à aborder la phase de conception technique pour répondre aux attentes des utilisateurs et aux exigences du système.

SPRINT 0 INSTALLATION
D'ENVIRONNEMENT LOGICIEL ET
MATÉRIEL

Introduction

Dans ce chapitre nous allons donner un aperçu sur les environnements matériels et logiciels utilisés pour la réalisation de ce projet

3.1 Environnement matériel

En cours de réalisation de notre projet, nous avons eu à notre disposition deux ordinateurs portables qui disposent des configurations suivantes :

TABLEAU 3.1 : Spécifications des ordinateurs utilisés pour le projet

Environnement matériel		
1 ^{er} ordinateur	2 ^{ème} ordinateur	3 ^{ème} ordinateur
		
Processeur : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz	Processeur : Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz	Processeur : Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
Mémoire installée (RAM) : 8.00 Go	Mémoire installée (RAM) : 8.00 Go	Mémoire installée (RAM) : 16.00 Go
Type de système : Windows 10 64 bits	Type de système : Windows 10 64 bits	Type de système : Windows 10 64 bits

3.2 Environnement logiciel

L'environnement logiciel pour la réalisation de notre projet comprend une série d'outils et de technologies essentiels à la conception, au développement, à la gestion et à la communication entre les membres de l'équipe. Voici une description détaillée des logiciels et plateformes utilisés :

3.2.1 Développement Backend



Pour le développement du serveur backend, nous avons choisi **NestJS**. NestJS est un framework Node.js moderne, conçu pour être extensible et performant. Il est basé sur TypeScript et suit une architecture modulaire qui permet de créer des applications backend robustes et maintenables. Il est particulièrement adapté pour des APIs RESTful et offre des outils puissants pour la gestion des services, des contrôleurs, et des routes.

3.2.2 Développement Frontend



Le développement du frontend est réalisé avec **Next.js**. Next.js est un framework React qui permet de créer des applications web performantes et évolutives, avec une prise en charge intégrée du rendu côté serveur (SSR) et du rendu statique. Il offre également des optimisations automatiques pour les performances et est idéal pour le SEO et les applications dynamiques.

3.2.3 Base de données

La gestion de nos données repose sur **MySQL**, une base de données relationnelle. MySQL est utilisée pour sa stabilité, sa performance, et sa large adoption dans les systèmes de gestion de bases de données. Avec sa capacité à gérer des requêtes complexes



et des relations entre les données, elle est idéale pour notre application et son architecture backend.

3.2.4 Page de destination



La création de la page de destination (landing page) est réalisée avec **Framer**. Framer est un outil de conception d'interfaces interactives qui permet de prototyper des pages web de manière rapide et visuelle. Il offre une intégration fluide avec React, permettant de transformer facilement des conceptions statiques en éléments interactifs et dynamiques.

3.2.5 Tests API



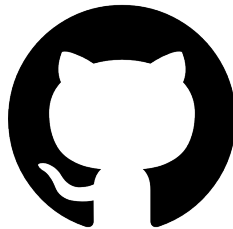
Pour tester les API développées, nous utilisons **Swagger**. Swagger est une plateforme qui permet de concevoir, documenter et tester des APIs de manière interactive. Grâce à son interface intuitive, les développeurs peuvent facilement créer des spécifications OpenAPI, générer des mocks, et tester les endpoints de manière rapide et fiable. Cela facilite aussi l'intégration des tests automatisés.

3.2.6 Documentation OpenAPI



La documentation et la définition des APIs sont réalisées avec **OpenAPI**. OpenAPI est une spécification standardisée qui permet de décrire les interfaces RESTful de manière structurée et lisible. Nous l'utilisons pour générer des documents API et pour assurer une intégration fluide entre le backend et le frontend. La documentation OpenAPI est essentielle pour l'automatisation de la génération de code et la validation des appels d'API.

3.2.7 Gestion de projet et gestion de versions



Nous utilisons **GitHub Project Board** pour gérer nos tâches et sprints avec des tableaux Kanban, facilitant la collaboration, le suivi de l'avancement et la gestion des tickets. **GitHub** sert également de plateforme centrale pour le versionnage du code, la révision, la gestion des branches et le suivi des modifications, assurant une collaboration fluide et un contrôle de version efficace.

Conclusion

Tout au long de ce chapitre, nous avons présenté l'environnement matériel et logiciel utilisé ainsi que les différents choix technologiques. Dans le chapitre suivant, nous

exposons la réalisation de notre travail ainsi que les résultats obtenus.

DÉVELOPPEMENT DU SPRINT 1

Introduction

Dans ce premier sprint, notre objectif était de mettre en place les bases de notre application en intégrant les principales fonctionnalités nécessaires à la création de comptes utilisateur et au lancement de l'enregistrement des consultations.

4.1 Planification du Sprint

Objectifs du Sprint 1

- Mettre en place les fonctionnalités de base du système pour permettre aux utilisateurs de s'inscrire, de compléter leur profil et d'interagir avec le tableau de bord.
- Permettre la création et la gestion des informations de l'utilisateur : inscription, collecte des informations personnelles et ajout de photo de profil.
- Construire la première version du tableau de bord pour afficher les informations essentielles de l'utilisateur de manière claire et accessible.
- Assurer la navigation et l'intégration des pages principales à travers une barre latérale intuitive.

Durée du Sprint 1

- Durée : 2 semaines (Semaine 1 à Semaine 2)

Équipe

- Développeurs backend : 1
- Développeurs frontend : 1

4.2 Backlog du Sprint 1

Éléments du backlog et tâches

ID	Fonctionnalité	Histoire utilisateur	Priorité	Durée estimée (jours)
1	Inscription et connexion utilisateur	— Créer le formulaire d'inscription avec validation de l'email. — Implémenter le mécanisme d'authentification par email et mot de passe. — Envoyer un email de validation après inscription.	Haute	8
2	Indicateur de progression du profil utilisateur	— Créer un indicateur de progression du profil utilisateur. — Mettre en place les différentes étapes du profil à compléter. — Ajouter un bouton pour inciter l'utilisateur à compléter son profil.	Moyenne	10

ID	Fonctionnalité	Histoire utilisateur	Priorité	Durée estimée (jours)
3	Collecte des informations utilisateur	— Créer des formulaires pour collecter les informations personnelles (nom, adresse, entreprise, etc.). — Implémenter une fonctionnalité de mise à jour des informations personnelles. — Sauvegarder les informations dans la base de données.	Haute	12
4	Téléchargement de photo de profil	— Implémenter une fonctionnalité de téléchargement d'image de profil. — Ajouter une image par défaut si l'utilisateur n'a pas encore téléchargé d'image. — Permettre la modification de l'image de profil.	Moyenne	8

ID	Fonctionnalité	Histoire utilisateur	Priorité	Durée estimée (jours)
5	Vue d'ensemble du tableau de bord	— Concevoir la structure du tableau de bord avec une vue d'ensemble des informations utilisateur (ex : paiements récents, factures impayées). — Ajouter les sections essentielles du tableau de bord. — Mettre en place une interface simple et claire.	Haute	10

4.3 Capture des besoins

Inscription et connexion utilisateur

Besoins fonctionnels :

- Le système doit permettre à l'utilisateur de s'inscrire avec un email valide et un mot de passe.
- Un email de validation doit être envoyé à l'utilisateur pour confirmer son inscription.
- Le processus de connexion doit être sécurisé et simple.

Besoins non fonctionnels :

- La sécurité des données doit être une priorité (stockage des mots de passe, validation email).
- L'expérience utilisateur doit être fluide et sans faille, notamment sur mobile et

desktop.

Indicateur de progression du profil utilisateur

Besoins fonctionnels :

- Un indicateur visuel doit être fourni pour que l'utilisateur puisse suivre la progression de son profil.
- Le système doit savoir quelles informations sont manquantes et inciter l'utilisateur à les ajouter.

Besoins non fonctionnels :

- L'indicateur doit être visuellement attractif et facile à comprendre.
- Il doit être mis à jour en temps réel lorsque l'utilisateur modifie son profil.

Collecte des informations utilisateur

Besoins fonctionnels :

- Les utilisateurs doivent pouvoir saisir, modifier et sauvegarder leurs informations personnelles et professionnelles.
- Les informations doivent être stockées de manière sécurisée.

Besoins non fonctionnels :

- Les formulaires doivent être clairs et simples à remplir.
- La sauvegarde des données doit être fiable et rapide.

Téléchargement de photo de profil

Besoins fonctionnels :

- L'utilisateur doit pouvoir télécharger une photo de profil via une interface simple.
- Une image par défaut doit être utilisée en l'absence de photo.

Besoins non fonctionnels :

- L'upload doit être rapide, et l'image téléchargée doit être redimensionnée pour un affichage correct sur le profil.

Vue d'ensemble du tableau de bord

Besoins fonctionnels :

- Le tableau de bord doit afficher des informations essentielles, comme les paiements récents, les factures impayées, et des résumés de l'état du compte.
- L'interface doit être claire et facile à comprendre.

Besoins non fonctionnels :

- L'interface doit être réactive et s'adapter à tous les types d'appareils (responsive design).
- Le tableau de bord doit se charger rapidement et offrir une expérience utilisateur fluide.

Description textuelle de « Inscription et connexion utilisateur »

Objectif : L'inscription et la connexion utilisateur sont essentielles pour permettre à l'utilisateur d'accéder à la plateforme. Le système doit permettre à l'utilisateur de s'inscrire en saisissant son email et un mot de passe, puis envoyer un email de validation pour confirmer l'adresse email. L'utilisateur pourra ensuite se connecter en utilisant ces informations.

4.4 Analyse

4.4.1 Diagramme de cas d'utilisation pour Sprint 1

Voici un diagramme de cas d'utilisation simplifié représentant le processus d'inscription et de connexion de l'utilisateur.

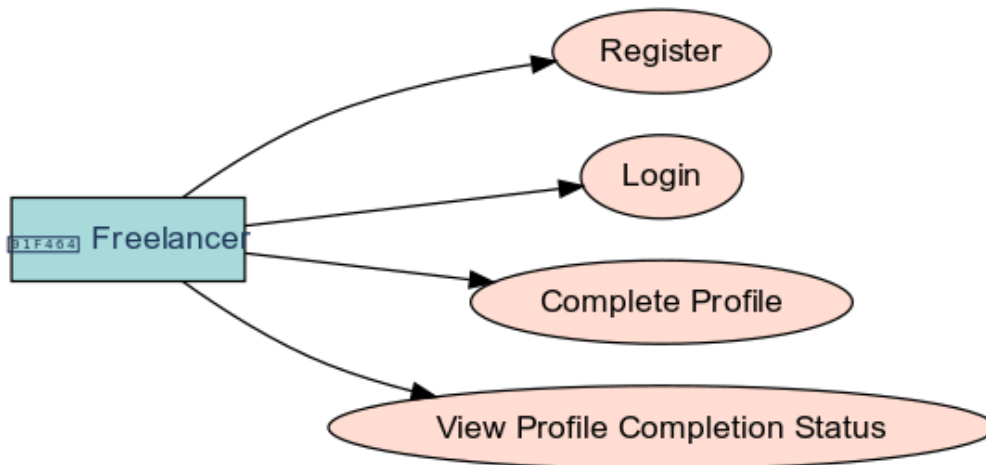


FIGURE 4.1 : Diagramme de cas d'utilisation de sprint 1

4.5 Les interfaces graphiques

4.6 Analyse

4.6.1 Diagramme de séquence pour Sprint 1

Voici un diagramme de séquence simplifié représentant le processus d'inscription et de connexion de l'utilisateur. Je vais décrire les différentes étapes ici.

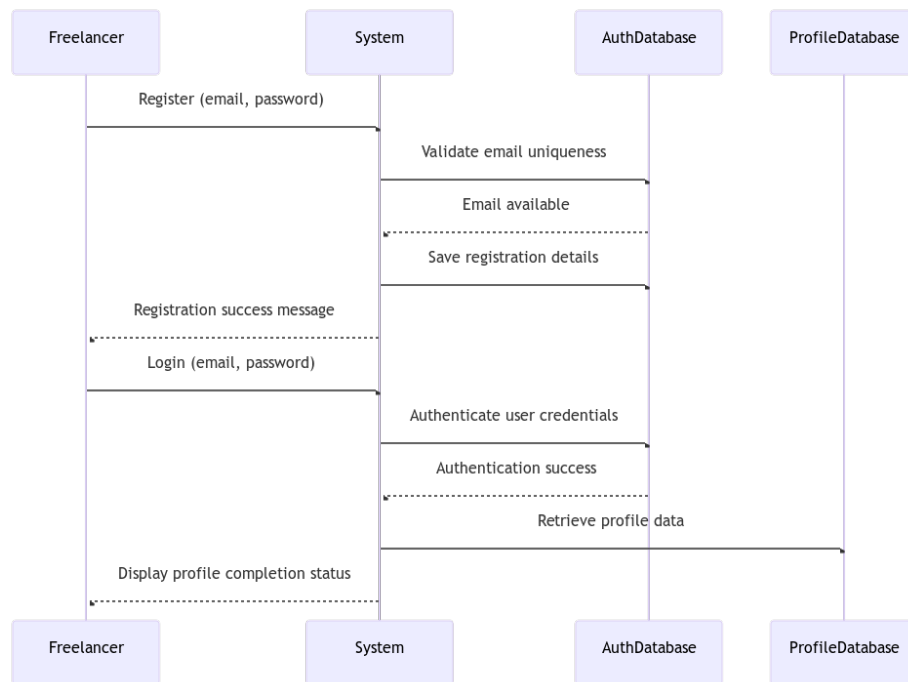


FIGURE 4.2 : Diagramme de séquence d'authentification

4.7 Les interfaces graphiques

Le processus d'authentification commence par l'entrée de l'email de l'utilisateur (voir première image). Ensuite, l'utilisateur se connecte à la plateforme (voir deuxième image). Si son profil est évalué à moins de 40%, il sera redirigé vers la page de complétion du profil. Le processus de complétion du profil se divise en trois étapes :

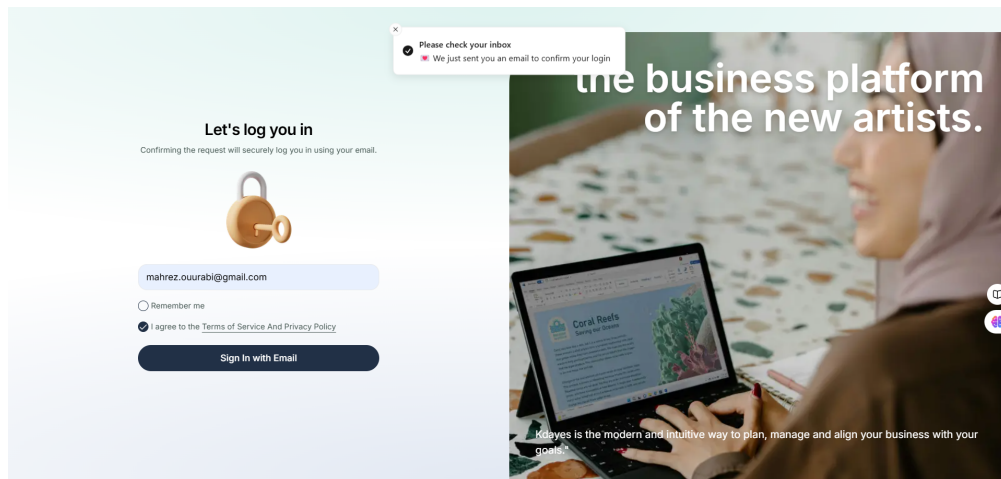


FIGURE 4.3 : Entrée de l'email de l'utilisateur pour l'authentification

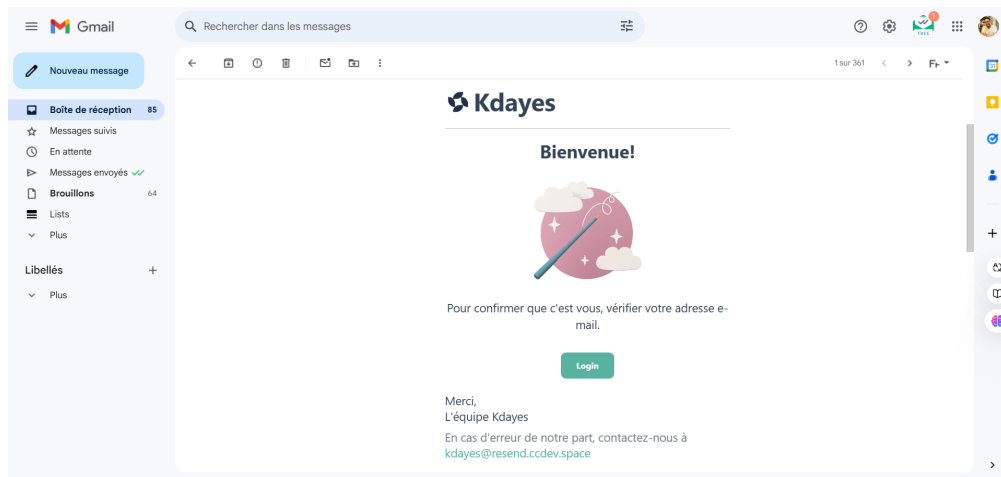


FIGURE 4.4 : Écran de connexion de l'utilisateur à la plateforme

La première étape de la complétion du profil consiste en la saisie des informations personnelles par l'utilisateur.

The screenshot shows a web application interface for profile completion. On the left is a sidebar with navigation links: Home, Invoices, Bills, Payments, Reports, Settings, Profile (highlighted), and Help. Below the sidebar is a 'Subscription' section with a progress bar indicating 'Your profile is 10% complete, you're almost there...'. The main content area is titled 'Tell us about yourself' with the subtitle 'Add your personal information'. It features three tabs: 'General infos' (selected), 'Company infos', and 'Bank infos'. The 'General infos' tab contains the following fields: 'Your full name' (text input with 'mahrez ourabi'), 'Your phone number' (text input with '+216 96 215 406' and a country code dropdown), 'Your address' (text input with 'Test'), and 'Your image' (a green button labeled 'Upload your profile picture'). A dark blue 'Save' button is at the bottom right. A small purple chat bubble is on the right edge.

FIGURE 4.5 : Redirection vers la page de complétion du profil lorsque le score est inférieur à 40%

La deuxième étape consiste à entrer les informations relatives à l'entreprise.

The screenshot shows the 'Tell us about your company' page. The sidebar is identical to the previous page. The main content area is titled 'Tell us about your company' with the subtitle 'Add your company name and your patente number'. It features three tabs: 'General infos', 'Company infos' (selected), and 'Bank infos'. The 'Company infos' tab contains the following fields: 'Your company name' (text input with 'Your company name'), 'Your patente number' (text input with 'Patente number'), and 'Your company logo image' (a green button labeled 'Upload your company logo'). Below these fields is a toggle switch for 'I don't have company yet'. A dark blue 'Save' button is at the bottom right. A small purple chat bubble is on the right edge.

FIGURE 4.6 : Saisie des informations sur l'entreprise

Enfin, l'utilisateur entre ses informations bancaires pour finaliser la complétion de son profil.

The screenshot shows a web application interface for entering bank account information. On the left is a sidebar menu with options: Home, Invoices, Bills, Payments, Reports, Settings, Profile (highlighted), and Help. The main content area is titled 'Bank account informations' with a sub-header 'Please provide us with informations regarding your bank account.' Below this are three tabs: 'General infos', 'Company infos', and 'Bank infos' (which is selected). The 'Bank infos' tab contains three input fields: 'Institution' (a dropdown menu with 'Select a bank'), 'IBAN' (a text field with a red border and a note 'must contain exactly 20 digits'), and 'Account owner' (a text field with a note 'Must be the same name associated with the account'). A 'Save' button is located at the bottom right of the form. At the bottom of the page, there is a 'Subscription' section with a progress bar indicating 'Your profile is 40% complete, you're almost there...' and the 'Kdayes' logo.

FIGURE 4.7 : Saisie des informations bancaires pour compléter le profil

Conclusion

Ce premier sprint a permis d'établir les bases de l'application, en mettant en place des fonctionnalités essentielles telles que l'inscription, la connexion et la gestion du profil utilisateur. Nous avons également introduit un tableau de bord fonctionnel et une interface utilisateur simplifiée pour une navigation fluide. La collecte des informations personnelles, d'entreprise et bancaires a été implémentée de manière sécurisée, et un indicateur de progression du profil a été mis en place pour inciter les utilisateurs à compléter leur profil. Ce sprint constitue une étape cruciale pour le développement ultérieur de l'application, avec une base solide pour intégrer des fonctionnalités supplémentaires dans les prochains sprints.

DÉVELOPPEMENT DU SPRINT 2

Introduction

Le Sprint 2 se concentre sur l'implémentation de fonctionnalités essentielles pour la gestion des factures et des utilisateurs. L'objectif est de permettre la création, l'édition, et l'envoi de factures, tout en assurant une navigation fluide au sein de l'application grâce à une barre latérale intuitive. Ce sprint vise à offrir une expérience utilisateur simple et efficace, en garantissant que les utilisateurs puissent facilement gérer les informations liées aux factures, de la création à l'envoi, avec une interface claire et fonctionnelle.

5.1 Planification du Sprint

Objectifs du Sprint 2

- Implémenter les fonctionnalités de gestion des factures et des utilisateurs, y compris la création, l'édition et l'affichage des factures.
- Mettre en place la navigation principale, en permettant aux utilisateurs de naviguer facilement entre les différentes sections de l'application à l'aide de la barre latérale.
- Assurer l'édition et la visualisation des factures, avec une interface permettant de modifier les informations liées aux factures (client, montant, échéance) et d'ajouter des détails.
- Garantir une expérience fluide et intuitive dans la gestion des factures et des informations utilisateurs, avec des tâches simples à effectuer (ajout/modification de factures).

Durée du Sprint 2

- 2 semaines (Semaine 3 à Semaine 4)

Équipe

- Développeurs backend : 1
- Développeurs frontend : 1

5.2 Backlog du Sprint 2

Éléments du backlog et Tâches

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
1	Sidebar Navigation	— Créer une barre latérale avec des liens vers les différentes sections de l'application (tableau de bord, factures, paramètres, etc.). — Implémenter un style actif pour indiquer la page en cours.	Moyenne	6

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
2	Invoice Creation	— Créer un formulaire pour générer des factures (client, description des services/produits, montant, date d'échéance). — Ajouter des options pour sélectionner un client existant ou créer un nouveau client. — Ajouter un bouton de sauvegarde pour créer la facture.	Haute	12
3	Invoice Editing	— Permettre l'édition d'une facture existante (modification du client, montant, description, échéance). — Ajouter un bouton pour enregistrer les modifications.	Moyenne	10

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
4	Invoice View and Side View	— Implémenter une vue détaillée d'une facture lorsqu'on clique sur une facture dans la liste. — Afficher un résumé des informations de la facture et permettre l'édition directe. — Ajouter des options pour envoyer la facture par email ou télécharger le PDF.	Moyenne	8
5	Invoice Sending	— Permettre l'envoi d'une facture par email à un client. — Ajouter une option pour inclure un lien de paiement sécurisé et des rappels automatiques pour les paiements en retard. — Sauvegarder l'historique des factures envoyées et leur statut.	Haute	10

5.3 Capture des besoins

Sidebar Navigation

Besoins fonctionnels :

- Navigation facile entre les pages.
- Indication de la page active dans la barre latérale pour éviter toute confusion.

Besoins non fonctionnels :

- La barre latérale doit être responsive, s'adaptant aussi bien à l'affichage sur mobile que sur desktop.
- La navigation doit être rapide et fluide, sans ralentissement dans le chargement des pages.

Invoice Creation

Besoins fonctionnels :

- L'utilisateur doit pouvoir créer une facture en sélectionnant un client, en ajoutant une description, un montant et une date d'échéance.
- La fonctionnalité doit permettre d'ajouter un nouveau client si nécessaire.

Besoins non fonctionnels :

- Le formulaire de création de facture doit être simple, clair, et réactif.
- Les informations saisies doivent être validées et sauvegardées de manière fiable.

Invoice Editing

Besoins fonctionnels :

- L'édition de facture doit permettre de modifier les informations de base (client, montant, services).

- Il doit être possible de sauvegarder les modifications de manière transparente.

Besoins non fonctionnels :

- L'édition des factures doit être rapide et ne pas nécessiter de rechargement complet de la page.
- L'utilisateur doit être alerté si des modifications n'ont pas été sauvegardées.

Invoice View and Side View

Besoins fonctionnels :

- Permettre à l'utilisateur de voir une vue détaillée d'une facture, y compris les informations de client, services fournis, et statut de la facture.
- Offrir la possibilité d'éditer directement les informations de la facture à partir de cette vue détaillée.

Besoins non fonctionnels :

- La vue détaillée doit se charger rapidement.
- L'interface doit être claire et sans ambiguïté, avec un accès facile aux options d'édition et d'envoi.

Invoice Sending

Besoins fonctionnels :

- L'utilisateur doit pouvoir envoyer une facture à un client par email avec un lien de paiement sécurisé.
- Il doit également être possible d'inclure des rappels automatiques pour les paiements en retard.

Besoins non fonctionnels :

- L'envoi des factures et des rappels doit être rapide et fiable.
- Le lien de paiement doit être sécurisé et fonctionnel à tout moment.

5.4 Analyse

5.4.1 Diagramme de séquence pour Invoice Sending

L'utilisateur peut envoyer une facture à un client par email. Ce processus peut inclure l'ajout d'un lien de paiement sécurisé, permettant au client de régler la facture en ligne. De plus, cette fonctionnalité doit prévoir l'envoi de rappels automatiques pour les paiements en retard et permettre de suivre l'historique des factures envoyées.

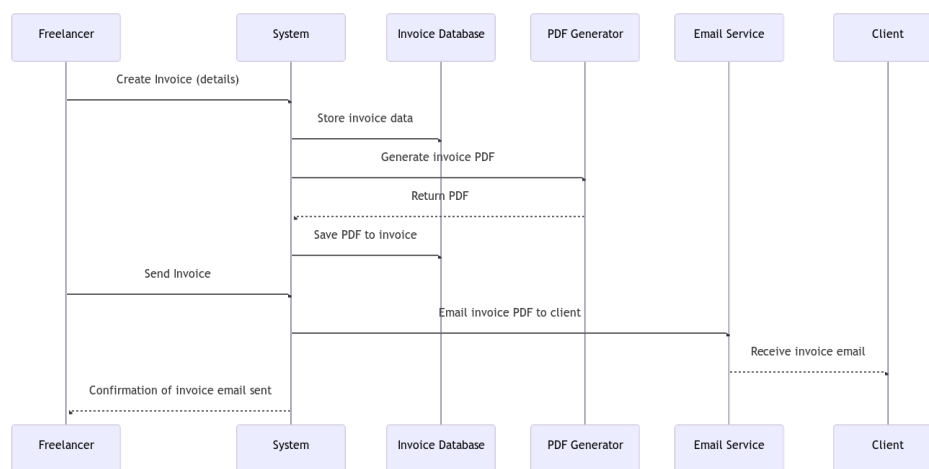


FIGURE 5.1 : Diagramme de séquence pour Invoice Sending

5.5 Les interfaces graphiques

5.5.1 Création d'une Facture

Le processus de création d'une nouvelle facture commence par la saisie des informations essentielles. L'utilisateur doit compléter les informations d'en-tête de la facture, les informations client, les articles facturés, les préférences de paiement, et enfin, générer un lien de paiement sécurisé avec un QR code.

D'abord, l'utilisateur saisit les informations d'en-tête, telles que le titre de la facture, ainsi que les informations du client (nom, adresse, matricule fiscal et email).

Create an invoice
Please make sure to fill in all the required fields to remove the watermark and successfully send the invoice to your client

Header
Client, description and invoice header

Invoice reference : Draft-Invoice

Invoice title
invoice-1

Client name*
Client-1

Client address
Client-address-1

Matricule fiscal* (client)
111111111

Client email*
mahrez.ourabi2018@gmail.com

From: mahrez ourabi
Test
vatNo:
Email: Mahreez.ourabi@gmail.com

To: Client-1
Client-address-1
vatNo: 111111111
Email: mahrez.ourabi2018@gmail.com

Invoice reference: Draft-Invoice
Invoice title: invoice-1
Issue date: 11/11/2024
Due date: 21/11/2024

Invoice items

PRODUCTS	QUANTITY	PRICE	VAT	TOTAL
test	1	3 DT	%	3 DT
			Subtotal	3 DT
			Tax	0 DT

FIGURE 5.2 : Entrée des informations d'en-tête de la facture

Ensuite, l'utilisateur entre ses informations personnelles, telles que son nom, son adresse et son email, pour une présentation claire et formelle de la facture.

Create an invoice
Please make sure to fill in all the required fields to remove the watermark and successfully send the invoice to your client

Header
Client, description and invoice header

Personal information
Please fill in your personal information

Your name*
mahrez ourabi

Your address
my-address

Matricule fiscal*
1111111112

Your email*
Mahreez.ourabi@gmail.com

From: mahrez ourabi
my-address
vatNo: 1111111112
Email: Mahreez.ourabi@gmail.com

To: Client-1
Client-address-1
vatNo: 111111111
Email: mahrez.ourabi2018@gmail.com

Invoice reference: Draft-Invoice
Invoice title: invoice-1
Issue date: 11/11/2024
Due date: 21/11/2024

Invoice items

PRODUCTS	QUANTITY	PRICE	VAT	TOTAL
test	1	3 DT	%	3 DT

FIGURE 5.3 : Saisie des informations personnelles de l'utilisateur

Une fois les informations d'en-tête et du client complétées, l'utilisateur ajoute les articles ou services facturés, en précisant pour chaque élément la description, la quantité, le prix unitaire, et le montant total. Ces détails permettent un calcul précis du total de la facture.

Personal information
Please fill in your personal information

my-address
vatNo: 1111111112
Email: Mahrez.ourabi@gmail.com

Client-address-1
vatNo: 1111111111
Email: mahrez.ourabi2018@gmail.com

Items
Products and services

Item N°1

Item name
test-1

Quantity: 4, Unit cost: 3, Tax: ex : 19%

Item N°2

Item name
test-2

Quantity: 1, Unit cost: 0, Tax: ex : 19%

Notice : the negative amount can be used as a discount.

Add Item

Invoice reference: Draft-Invoice
Invoice title: invoice-1
Issue date: 11/11/2024
Due date: 21/11/2024

Invoice items

PRODUCTS	QUANTITY	PRICE	VAT	TOTAL
test-1	4	3 DT	%	12 DT
test-2	1	0 DT	%	0 DT
Subtotal				12 DT
Tax				0 DT
Total				12 DT

Payment methods
Pay securely online in seconds via [this link](#), or simply by scanning the QR code below

FIGURE 5.4 : Ajout des articles ou services facturés

L'utilisateur peut ensuite définir les préférences de paiement, telles que le mode de règlement (carte, virement, etc.), afin de faciliter le paiement selon les préférences du client.

Payment preferences
Select the methods of payment

Credit Card
Clients can pay you directly by credit card

Bank transfer
Pay with bank transfer, you have to fill your bank informations.

FIGURE 5.5 : Définition des préférences de paiement

De plus, l'utilisateur peut activer une option de rappel de paiement pour notifier le client si la date limite approche. Cela facilite la gestion des paiements en minimisant les retards.

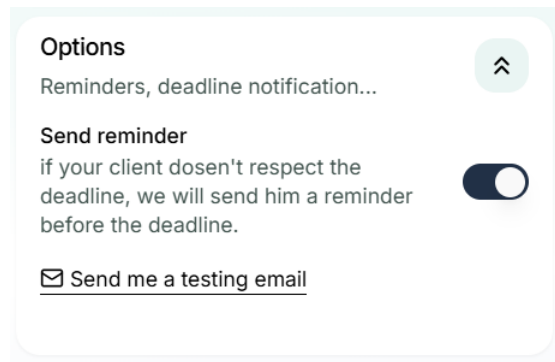


FIGURE 5.6 : Activation des notifications pour rappel de paiement

Enfin, un lien de paiement sécurisé avec QR code peut être généré, permettant au client de régler la facture rapidement et de manière sécurisée.

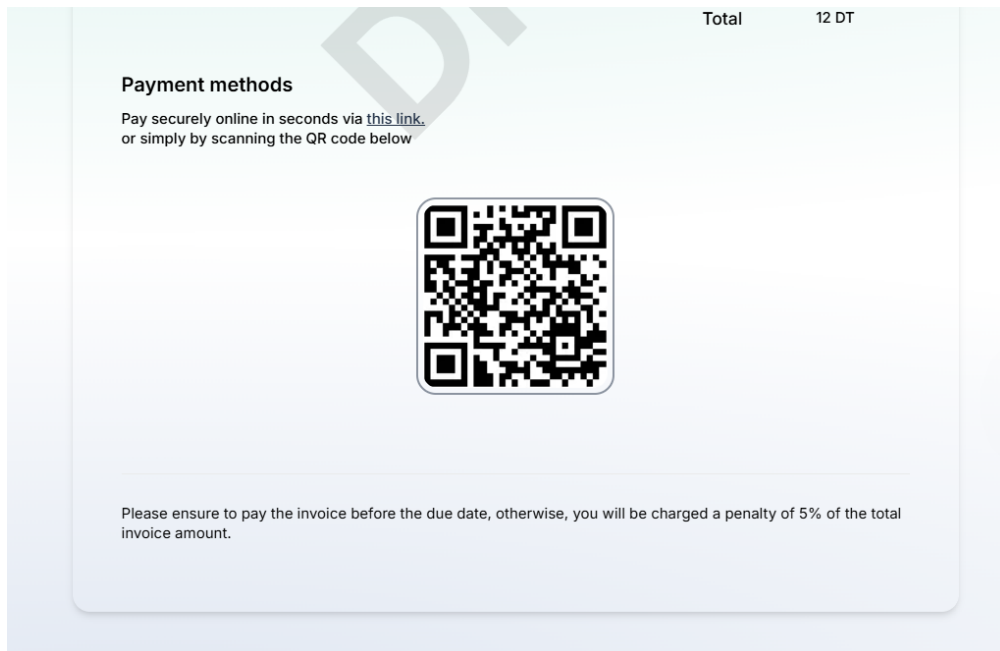


FIGURE 5.7 : Génération du lien de paiement sécurisé avec QR code

Une fois toutes les informations saisies, l'utilisateur peut cliquer sur **Sauvegarder et envoyer**. Un pop-up lui permet alors de sélectionner la date limite de paiement. Une fois la date choisie, la facture est enregistrée avec le statut **En attente**.

Si l'utilisateur choisit uniquement le bouton **Sauvegarder**, la facture est enregistrée

en tant que **Brouillon**, sans envoi ni statut de paiement.

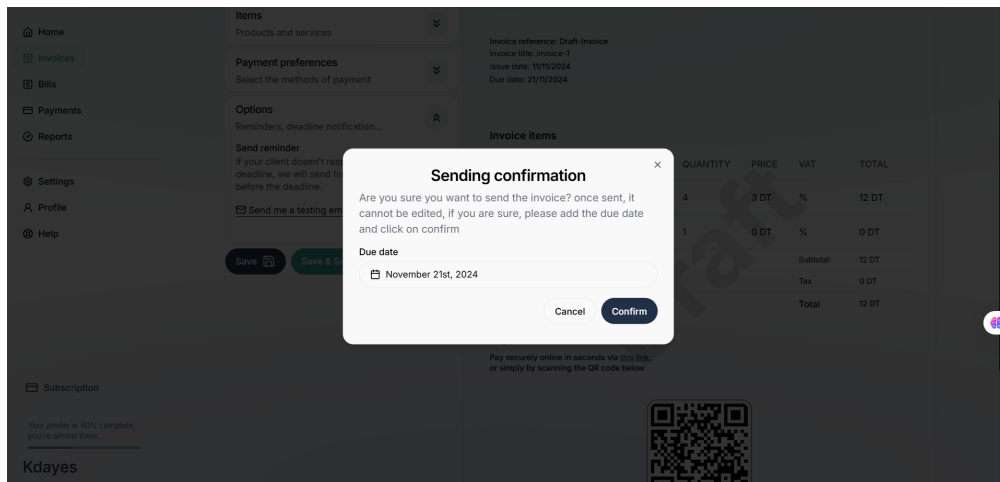


FIGURE 5.8 : Sélection de la date limite de paiement dans le pop-up

Une fois la facture sauvegardée ou envoyée, l'utilisateur est redirigé vers le tableau de bord des factures. Les factures en **Brouillon** apparaissent dans la section dédiée, tandis que celles envoyées figurent dans la liste des factures en attente de paiement.

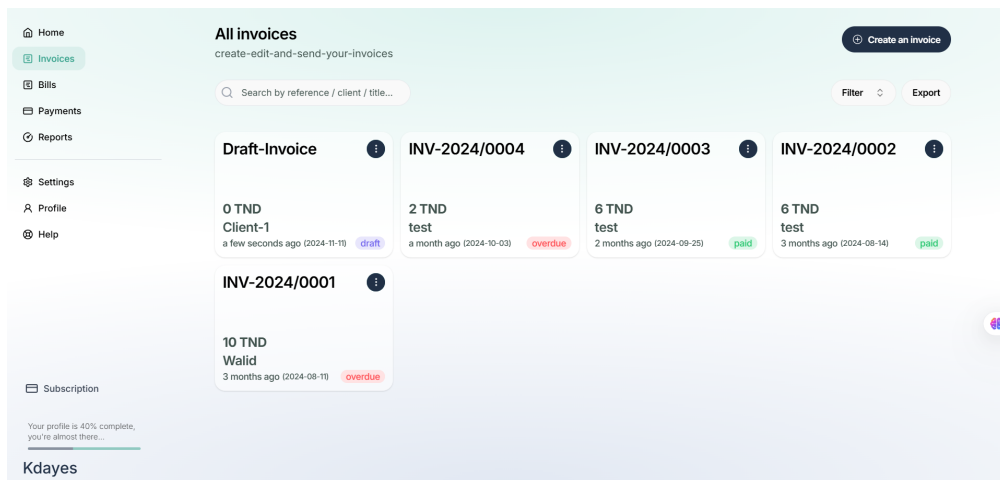


FIGURE 5.9 : Tableau de bord des factures permettant de vérifier l'ajout de la facture

5.5.2 Mise à Jour d'une Facture

L'utilisateur peut également modifier une facture existante pour actualiser des informations telles que les articles, les informations du client, ou les préférences de paiement. L'interface permet une édition simple et rapide.

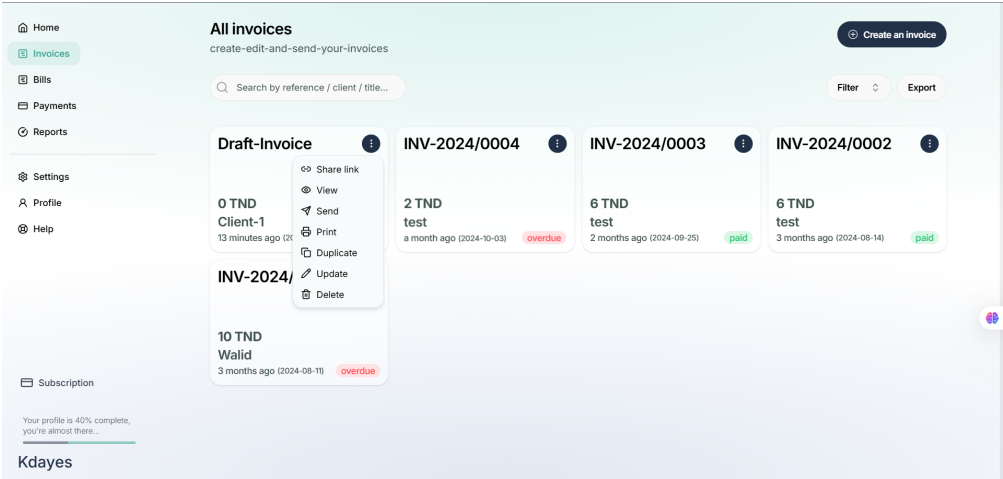


FIGURE 5.10 : Bouton pour mettre à jour une facture

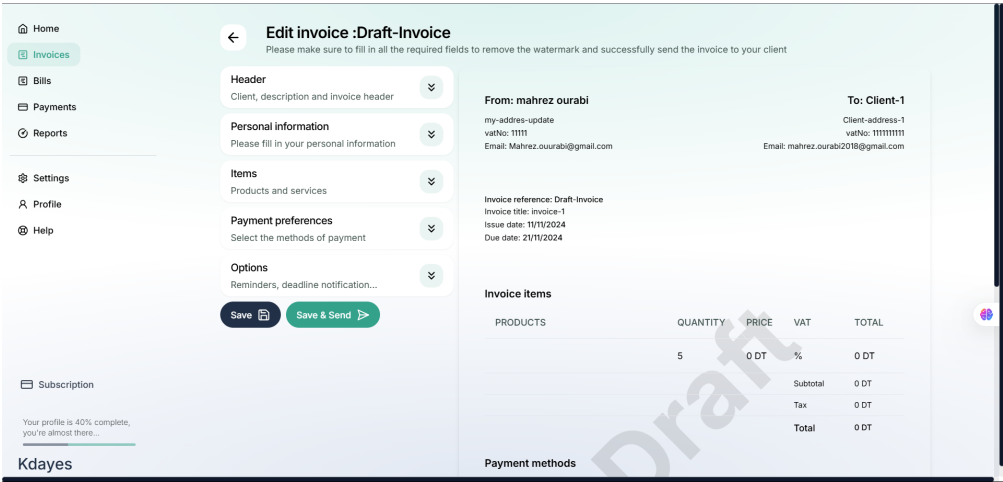


FIGURE 5.11 : Mise à jour des informations de la facture

5.5.3 Vue Détail d'une Facture

Après la création ou la mise à jour d'une facture, l'utilisateur peut consulter une vue détaillée, incluant toutes les informations saisies, telles que les articles, le montant total, et le statut de paiement. Cette vue permet aussi de visualiser l'historique des modifications de la facture.

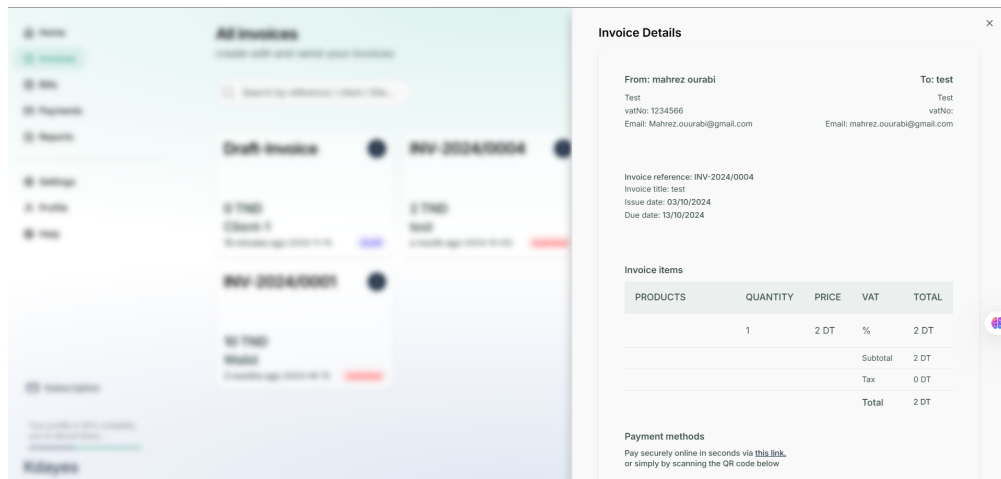


FIGURE 5.12 : Vue détaillée d'une facture avec toutes les informations

L'utilisateur peut également accéder à cette vue en cliquant sur le bouton "Voir" dans les options.

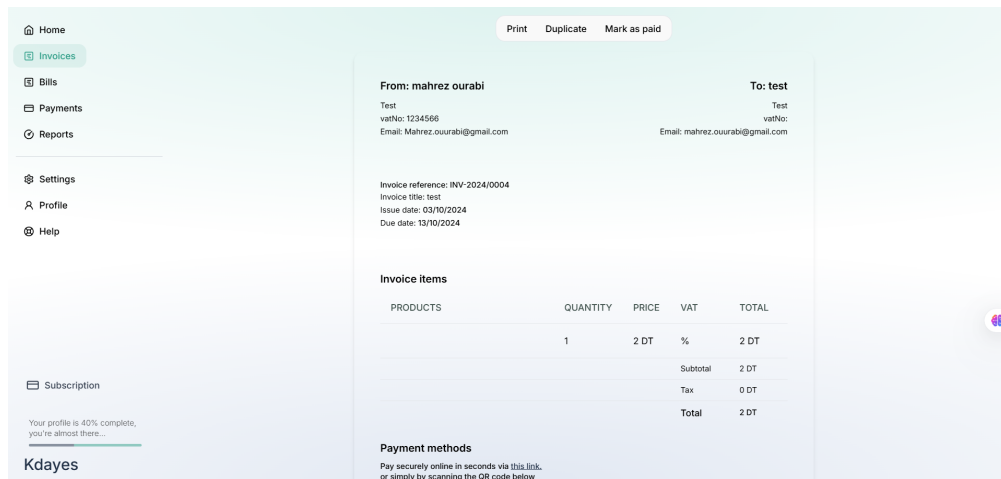


FIGURE 5.13 : Vue détaillée d'une facture

Conclusion

Le Sprint 2 se concentre sur la gestion des factures, un des aspects centraux de l'application. Ce sprint permet de couvrir la création, l'édition, et l'envoi des factures, tout en garantissant une navigation fluide via la barre latérale. Ces fonctionnalités sont essentielles pour offrir une expérience utilisateur complète, en permettant aux utilisateurs non seulement de gérer leurs informations personnelles, mais aussi d'effectuer des actions clés telles que la création et la gestion des factures. Le sprint se concentrera sur l'implémentation de ces fonctionnalités de manière intuitive, tout en assurant la fiabilité du système pour les utilisateurs finaux.

DÉVELOPPEMENT DU SPRINT 3

Introduction

Le Sprint 3 se concentre sur l'amélioration de l'interface utilisateur et la gestion du profil. Les objectifs incluent la finalisation de la gestion du profil utilisateur avec un indicateur de progression, l'optimisation de la création de factures, et l'amélioration de l'expérience utilisateur grâce à un tableau de bord interactif. Ce sprint vise à simplifier la navigation, renforcer la personnalisation du profil et améliorer l'efficacité dans la gestion des informations.

6.1 Planification du Sprint

Objectifs du Sprint 3

Finaliser la gestion du profil utilisateur et de l'interface du tableau de bord, en ajoutant les fonctionnalités de modification du profil et de visualisation des informations clés. Implémenter l'indicateur de progression de complétion du profil utilisateur, afin que les utilisateurs puissent suivre l'avancement de la complétion de leur profil (par exemple, "Votre profil est à 60 % complété"). Optimiser la création et la gestion des factures, en permettant l'ajout de produits/services détaillés et l'édition des informations de facturation pour une meilleure flexibilité. Améliorer l'expérience utilisateur avec un système de notification simple, pour signaler les événements clés, comme la création d'une facture ou l'édition d'un profil.

Durée du Sprint 3

— 2 semaines (Semaine 5 à Semaine 6)

Équipe

— Développeurs backend : 1

— Développeurs frontend : 1

6.2 Backlog du Sprint 3

Éléments du backlog et Tâches

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
1	Profile Completion Progress Indicator	1. Créer un indicateur de progression visuel pour le profil utilisateur (ex : barre de progression). 2. Afficher un pourcentage ou un texte (par exemple "Profil 60 complet") basé sur les champs remplis. 3. Ajouter des liens vers les sections manquantes pour faciliter la complétion du profil.	Haute	13
2	User Information Collection	1. Permettre à l'utilisateur de remplir ses informations personnelles (nom, email, etc.). 2. Ajouter des champs pour les informations bancaires et de l'entreprise. 3. Implémenter une fonctionnalité pour sauvegarder et mettre à jour les informations personnelles.	Haute	13

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
3	Profile Picture Upload	1. Permettre à l'utilisateur de télécharger une photo de profil. 2. Ajouter une image par défaut si l'utilisateur n'a pas téléchargé de photo. 3. Implémenter un mécanisme pour mettre à jour ou supprimer la photo de profil.	Haute	8
4	Dashboard Overview	1. Créer un tableau de bord qui affiche un résumé des informations clés : paiement récent, factures impayées, etc. 2. Ajouter des widgets d'informations pour les dernières actions de l'utilisateur (par exemple, dernier paiement effectué). 3. Optimiser l'affichage pour une meilleure lisibilité (adaptation mobile et desktop).	Haute	13

6.3 Capture des besoins

Profile Completion Progress Indicator

Besoins fonctionnels : L'utilisateur doit pouvoir voir un pourcentage de complétion de son profil. L'indicateur de progression doit être mis à jour automatiquement en fonction des informations saisies. Des liens ou des boutons doivent être fournis pour diriger l'utilisateur vers les sections incomplètes. **Besoins non fonctionnels :** L'indicateur de progression doit être bien visible, idéalement en haut de la page du profil. Il doit être

réactif et se mettre à jour instantanément lorsque des informations sont ajoutées ou modifiées.

User Information Collection

Besoins fonctionnels : Permettre à l'utilisateur de saisir et de sauvegarder ses informations personnelles et professionnelles. Prévoir des champs pour les informations bancaires et les informations liées à l'entreprise (nom, adresse, etc.).

Besoins non fonctionnels : L'interface de saisie doit être simple, claire, et sans ambiguïté. Les informations doivent être validées (par exemple, validation de l'email, du numéro de compte bancaire). Il doit être possible de mettre à jour ces informations facilement.

Profile Picture Upload

Besoins fonctionnels : Permettre l'upload d'une photo de profil et afficher une image par défaut si l'utilisateur n'en a pas. Ajouter une fonctionnalité pour modifier ou supprimer la photo de profil.

Besoins non fonctionnels : L'upload doit être rapide et sécurisé. La photo téléchargée doit être affichée correctement sur toutes les pages où elle est utilisée.

Dashboard Overview

Besoins fonctionnels : Le tableau de bord doit afficher un aperçu des paiements récents, des factures impayées, et d'autres informations pertinentes. Il doit être possible de personnaliser l'affichage pour que l'utilisateur puisse voir les informations les plus importantes pour lui.

Besoins non fonctionnels : Le tableau de bord doit être rapide à charger, même avec des données volumineuses. L'interface doit être intuitive et permettre à l'utilisateur

de naviguer facilement entre les différentes sections.

6.4 Les interfaces graphiques

Les interfaces graphiques développées lors du Sprint 3 visent à améliorer l'expérience utilisateur tout en facilitant l'accès aux fonctionnalités clés, telles que la gestion du profil et du tableau de bord. Ces interfaces se concentrent sur l'optimisation de la navigation et la présentation des informations.

6.4.1 Gestion du profil utilisateur

Pour la gestion du profil, l'interface reste similaire à celle du Sprint 1, avec quelques améliorations pour rendre la modification des informations plus intuitive. L'utilisateur peut facilement compléter son profil grâce à des champs clairs et des indicateurs de progression visibles. L'interface garantit une mise à jour en temps réel des informations et un système de validation pour assurer la qualité des données.

6.4.2 Tableau de bord

Le tableau de bord offre un aperçu immédiat des informations clés, telles que les paiements récents et les factures impayées. L'interface est optimisée pour une navigation rapide et intuitive, et le design réactif s'adapte à tous les appareils, mobiles ou de bureau.

Conclusion

Le Sprint 3 se concentre sur la gestion du profil utilisateur et l'amélioration de la fonctionnalité de création de factures. En ajoutant un indicateur de progression pour le profil, en permettant à l'utilisateur de compléter et mettre à jour ses informations personnelles et professionnelles, et en affinant la création de factures pour inclure des services/produits détaillés, le sprint améliore l'expérience utilisateur tout en garantissant

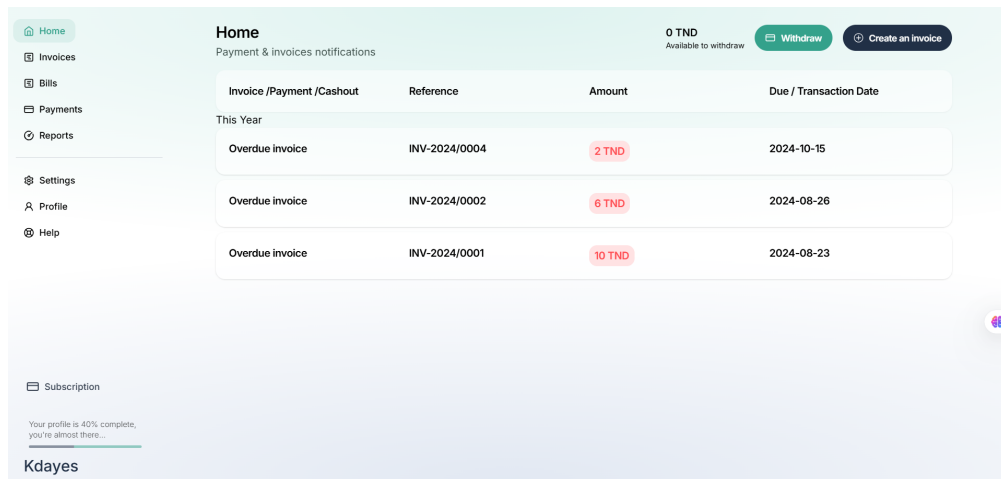


FIGURE 6.1 : Vue d'ensemble du tableau de bord

une plus grande flexibilité dans la gestion des informations et des factures. L'optimisation du tableau de bord offre également une vue d'ensemble claire des éléments clés de l'application, ce qui contribue à la productivité de l'utilisateur.

DÉVELOPPEMENT DU SPRINT 4

Introduction

7.1 Planification du Sprint

Objectifs du Sprint 4

Améliorer la gestion des paiements et des rappels, en mettant en place des notifications automatiques pour les paiements en retard et en intégrant le système de suivi des paiements avec des mises à jour automatiques de statut. Mettre en place des fonctionnalités de reporting et d'analyse, permettant aux utilisateurs de visualiser et exporter des rapports détaillés sur leurs paiements, factures et revenus. Introduire les préférences de notifications, permettant à l'utilisateur de personnaliser les alertes qu'il souhaite recevoir (par exemple, paiements, rappels de factures, etc.). Optimiser l'expérience utilisateur dans la gestion des paiements, en intégrant des options de paiement en ligne (notamment via Konnect) et en facilitant la gestion des différents états des factures.

Durée du Sprint 4

— 2 semaines (Semaine 7 à Semaine 8)

Équipe

- Développeurs backend : 1
- Développeurs frontend : 1

7.2 Backlog du Sprint 4

Éléments du backlog et tâches

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
1	Paielement en ligne via Konnect	— Intégrer le système de paiement en ligne via Konnect. — Ajouter un lien de paiement sécurisé dans les factures. — Tester les transactions pour s'assurer de la fiabilité.	Haute	13
2	Mise à jour du statut des paiements	— Mettre en place un système de mise à jour automatique du statut des paiements. — Ajouter une notification par email à l'utilisateur lorsque le statut de la facture change (payée, en retard, etc.). — Intégrer le suivi des paiements en temps réel.	Haute	8

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
3	Rappels de paiement automatiques	— Permettre à l'utilisateur de définir des intervalles pour les rappels de paiement. — Automatiser l'envoi des rappels par email aux clients. — Tester les différents scénarios d'envoi de rappels.	Haute	10
4	Reporting des paiements et factures	— Créer un rapport détaillé sur les paiements, les factures impayées et les revenus. — Ajouter des options de filtrage par statut, date et client.	Haute	12

ID	Fonctionnalité	Histoire Utilisateur	Priorité	Durée estimée (jours)
5	Préférences de notification	— Créer un écran permettant à l'utilisateur de définir ses préférences de notifications (emails). — Implémenter des options pour activer/désactiver des notifications spécifiques. — Tester le bon fonctionnement des préférences de notification.	Moyenne	8

7.3 Capture des besoins

Paieement en ligne via Konnect

Besoins fonctionnels : L'utilisateur doit pouvoir générer un lien de paiement sécurisé et l'inclure dans ses factures. Les paiements doivent être traités via un système sécurisé et fiable (Konnect). Les utilisateurs doivent recevoir une confirmation lorsque le paiement est effectué avec succès.

Besoins non fonctionnels : L'intégration de Konnect doit être fluide, sans interruption des processus de facturation. Le lien de paiement doit être valide et fonctionner correctement sur tous les appareils.

Mise à jour du statut des paiements

Besoins fonctionnels : Le statut de chaque facture doit être mis à jour automatiquement lorsqu'un paiement est effectué. Une notification doit être envoyée à l'utilisateur lorsque le statut d'une facture change.

Besoins non fonctionnels : Les mises à jour de statut doivent se produire en temps réel ou presque, pour garantir une expérience fluide. La notification doit être claire et contenir des informations pertinentes sur la facture (par exemple, montant payé, solde restant).

Rappels de paiement automatiques

Besoins fonctionnels : L'utilisateur doit pouvoir configurer des intervalles pour les rappels de paiement automatiques. Le système doit envoyer des rappels réguliers et automatisés en fonction des paramètres définis.

Besoins non fonctionnels : L'envoi des rappels doit être fiable, et les emails doivent arriver à temps. Le système de rappels doit être flexible et permettre des ajustements dans les paramètres selon les besoins de l'utilisateur.

Reporting des paiements et factures

Besoins fonctionnels : Les utilisateurs doivent pouvoir générer des rapports détaillés sur leurs paiements et factures. Les rapports doivent être filtrables par différents critères (client, statut, date). Les rapports doivent pouvoir être exportés sous différents formats, notamment CSV et PDF.

Besoins non fonctionnels : Les rapports doivent être générés rapidement, sans retarder l'expérience utilisateur. Le système doit permettre l'exportation facile et correcte des rapports.

Préférences de notification

Besoins fonctionnels : L'utilisateur doit pouvoir activer ou désactiver des notifications spécifiques (paiements reçus, rappels de paiement, etc.). Les préférences doivent pouvoir être modifiées à tout moment.

Besoins non fonctionnels : Les préférences doivent être enregistrées de manière fiable et appliquées immédiatement. Les options doivent être simples à utiliser et accessibles depuis l'interface utilisateur.

7.4 Analyse

7.4.1 Diagramme de séquence pour les paiements

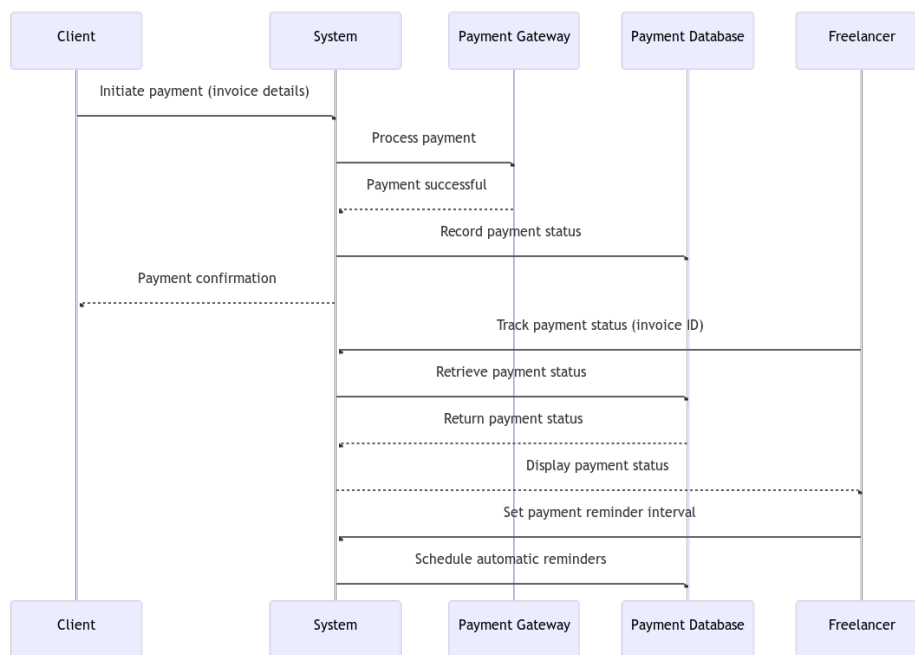


FIGURE 7.1 : Diagramme de séquence pour les paiements

7.5 Interfaces Graphiques

Lorsqu'une nouvelle facture est générée, le client reçoit une notification par e-mail contenant tous les détails de la facture ainsi qu'un bouton de paiement, offrant un accès direct et simplifié à l'interface de paiement pour régler la facture.

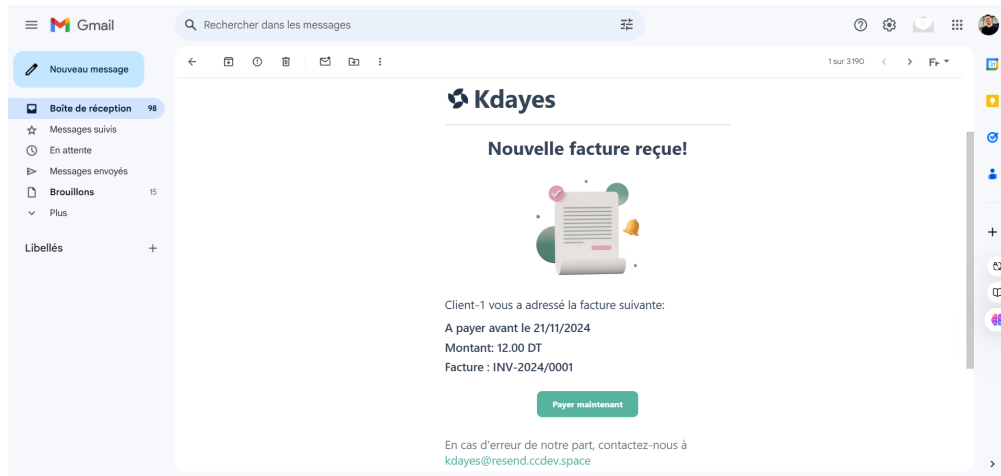


FIGURE 7.2 : Notification par e-mail avec bouton de paiement

Étapes Suivantes pour le Paiement

- En cliquant sur le bouton de paiement dans l'e-mail, le client est redirigé vers une interface de visualisation de la facture. Cette interface lui fournit un récapitulatif complet, incluant les produits ou services facturés, le montant total dû, et toute information complémentaire. Le client peut ensuite cliquer sur "Payer" pour accéder à la passerelle de paiement sécurisée.

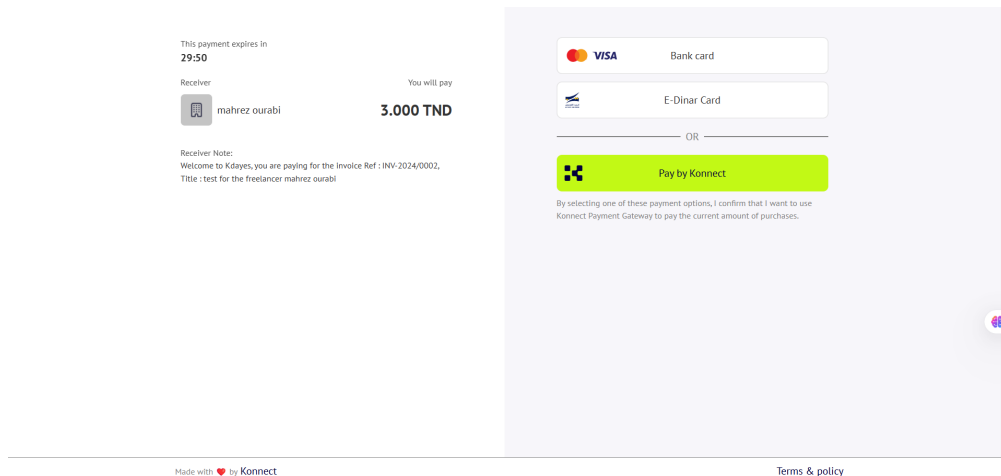


FIGURE 7.3 : Interface de visualisation de la facture avec option de paiement

Après avoir sélectionné un mode de paiement, par exemple "Carte bancaire", la page de paiement dédiée s'affiche.



FIGURE 7.4 : Page de paiement par carte bancaire

Tableau de Bord des Paiements Réussis

Le tableau de bord des paiements réussis présente les transactions complétées, permettant aux freelances de suivre facilement leurs paiements reçus. Les principales fonctionnalités de cette section incluent :

- Affichage des Paiements Reçus : Seuls les paiements marqués comme réussis sont affichés, facilitant la visualisation des transactions finalisées.
- Détails des Transactions : Chaque paiement affiche le montant, la date de réception, et le client, offrant une vue d'ensemble complète des transactions réussies.
- Filtrage des Paiements : L'utilisateur peut filtrer les paiements par date ou par client, facilitant la recherche de transactions spécifiques.
- Confirmation de Paiement : Chaque transaction est accompagnée d'une confirmation de statut "payé", assurant une traçabilité fiable.

Ref transaction	Ref invoice	Client	Amount	Date of transaction
Today				
PAY-2024/0004	INV-2024/0001	Walid	10 TND	2024-11-11
PAY-2024/0003	INV-2024/0004	test	2 TND	2024-11-11
This Year				
PAY-2024/0002	INV-2024/0003	test	6 TND	2024-09-25
PAY-2024/0001	INV-2024/0002	test	6 TND	2024-09-05

FIGURE 7.5 : Tableau de bord des paiements réussis

Cette vue simplifiée permet aux freelances de se concentrer sur les transactions réussies, facilitant ainsi le suivi de leurs revenus encaissés.

Processus de Reporting

Après chaque paiement, le freelance peut accéder à une section dédiée au reporting pour consulter et analyser les informations financières. Le processus de reporting inclut :

- Consultation des Factures : L'utilisateur peut consulter toutes les factures, organisées par statut (payée, en attente, en retard, etc.), pour suivre facilement l'état de

chaque transaction.

- Génération de Rapports Financiers : Le système permet de générer des rapports détaillés de l'historique des paiements et des factures, avec des options de filtrage par client, statut de facture, ou date.

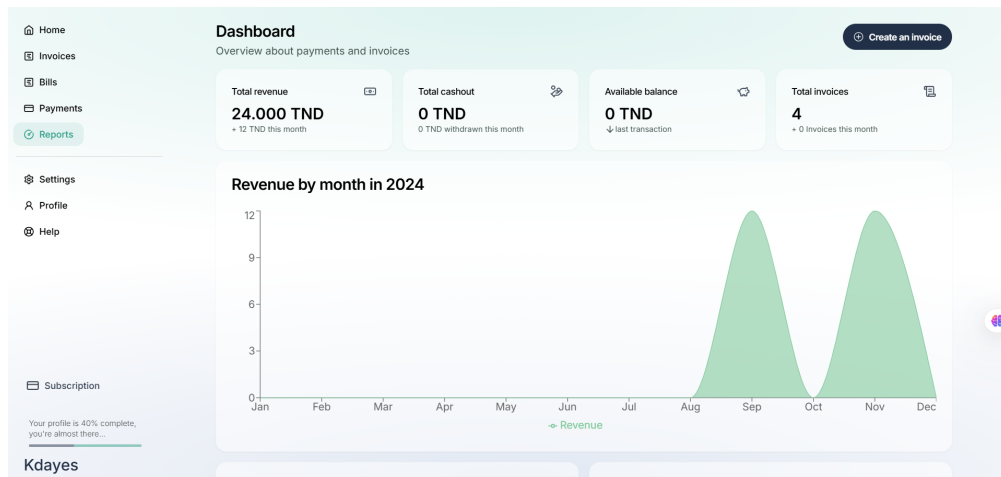


FIGURE 7.6 : Vue des factures et des rapports financiers

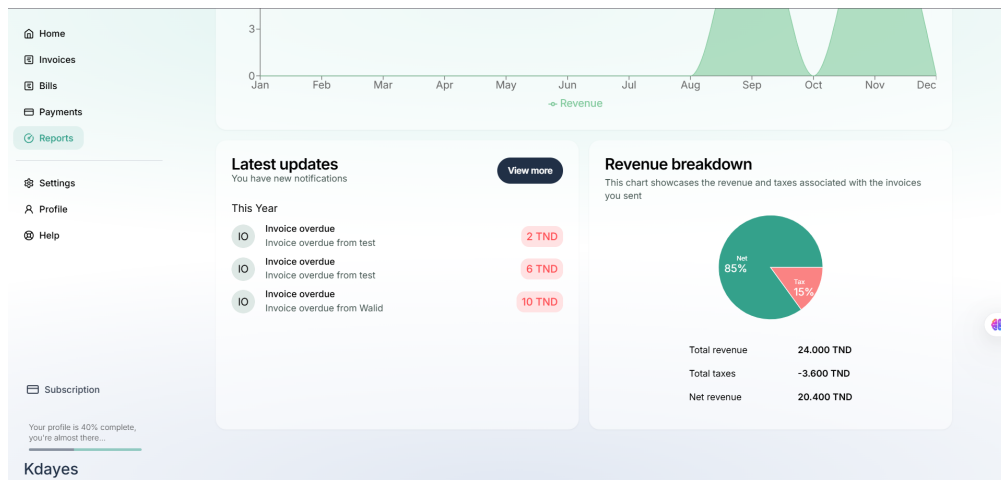


FIGURE 7.7 : Vue des factures et des rapports financiers

Ce processus de reporting structuré aide les freelances à gérer efficacement leurs finances et à obtenir une vue claire de leurs revenus et paiements.

Conclusion

Le Sprint 4 vise à renforcer la gestion des paiements et des rapports financiers. L'intégration des paiements en ligne via Konnect. En parallèle, les rapports détaillés permettront aux utilisateurs de mieux comprendre l'état de leurs finances et d'exporter des données pour une analyse plus approfondie. L'objectif global est d'améliorer la productivité des utilisateurs tout en garantissant une expérience fluide et personnalisable.

Conclusion générale

Le parcours de développement de Kdayes a posé les bases d'une startup prometteuse qui répond aux besoins urgents des freelances en Tunisie. Grâce à une analyse rigoureuse, un développement itératif et un retour continu des parties prenantes, Kdayes est devenu une solution qui simplifie les processus administratifs clés, renforce la conformité avec les réglementations locales et offre une expérience utilisateur intuitive adaptée aux non-techniciens. Au-delà des objectifs techniques atteints, ce projet a prouvé la viabilité et la demande pour Kdayes en tant que plateforme dédiée aux freelances, confirmant le potentiel d'expansion future et d'impact régional.

Ce stage n'a pas seulement été une démarche technique, mais aussi une expérience d'apprentissage entrepreneurial, renforçant l'intersection entre la technologie, la stratégie commerciale et le design centré sur l'utilisateur dans la création de solutions impactantes. À l'avenir, Kdayes détient un potentiel de croissance significatif, avec des projets d'ajout de nouvelles fonctionnalités et d'élargissement de sa portée sur le marché, incarnant ainsi l'esprit d'innovation et d'adaptabilité nécessaire dans l'économie freelance en constante évolution.