



République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Monastir

Institut Supérieur d'Informatique et de Mathématiques de Monastir



N° d'ordre : L8INFO

Mémoire de Projet de Fin d'Études

Présenté en vue de l'obtention du :

**Diplôme National de Licence en
Sciences Informatiques**

Spécialité :

Génie Logiciel et Système d'Information

Par :

Mohamed Amira

**CONCEPTION ET RÉALISATION D'UNE
APPLICATION WEB DE FACTURATION DES
PRESTATIONS DE SERVICE.**

Soutenu le jj/mm/aaaa devant le jury composé de :

M./Mme :

M./Mme :

M./Mme : MAHMOUDI Ramzi

M./Mme : MOHAMED Bilel

Président

Rapporteur

Encadrant Pédagogique

Encadrant Professionnel



Dédicace

À mes chers parents, pour leur soutien inconditionnel et leurs sacrifices tout au long de mon parcours. Leur amour, leur encouragement et leur dévouement ont été une source constante de motivation et de force. Je suis reconnaissant(e) d'avoir des parents aussi merveilleux.

À mes sœurs et mon frère pour leur écoute attentive, leur soutien indéfectible et leur précieuse guidance. Leur présence dans ma vie a été une source de réconfort et d'inspiration, et je suis reconnaissant(e) de les avoir à mes côtés.

À mes deux proches, Hana Jdey et Jihen Zemzem, pour leur support émotionnel et leurs encouragements constants. Leur présence bienveillante et leur soutien ont été d'une valeur inestimable et ont contribué à mon épanouissement personnel et professionnel.

À vous tous,

Je dédie ce travail.

Amira MOHAMED

Remerciement

Tout d'abord, je souhaite exprimer ma profonde reconnaissance envers M. Ramzi Mahmoudi, qui a été mon encadrant pédagogique tout au long de ce projet. Sa présence bienveillante et ses conseils avisés ont été essentiels pour la réalisation de ce rapport. J'ai eu la chance de bénéficier de son encadrement attentif et de sa grande expertise. Je tiens à le remercier chaleureusement pour sa disponibilité et sa contribution précieuse à la réussite de ce projet.

Je tiens à exprimer ma reconnaissance envers M. Bilel Mohamed, qui a été d'une grande aide en tant qu'encadrant professionnel. Ses conseils précieux et son expertise ont joué un rôle essentiel dans notre développement dans le domaine de l'informatique. Grâce à lui, nous avons pu acquérir des connaissances approfondies et explorer de nouvelles perspectives dans ce domaine. Sa guidance attentive et sa passion pour l'informatique ont été une source d'inspiration tout au long du projet.

Je souhaite également exprimer ma reconnaissance envers Mme Wafa Mhamdi, notre encadrante de conception, qui a déployé tous ses efforts pour faciliter notre compréhension de l'application et des exigences fonctionnelles.

Enfin, je tiens à exprimer ma profonde gratitude envers tous les enseignants qui, tout au long de mon parcours universitaire, ont joué un rôle essentiel en m'enseignant les fondamentaux essentiels du domaine informatique. Leurs conseils avisés, leur expertise précieuse et leur soutien indéfectible ont été d'une valeur inestimable pour mon développement professionnel. Je suis reconnaissant envers chacun d'entre eux pour leur contribution significative à ma formation et à mon parcours dans le domaine de l'informatique.

■ TABLE DES MATIÈRES

LISTE DES FIGURES	vii
LISTE DES ABRÉVIATIONS	viii
INTRODUCTION GÉNÉRALE	ix
1 Contexte du projet	2
1.1 Introduction	3
1.2 Contexte générale	3
1.2.1 Cadre du projet	3
1.2.2 Présentation de l'organisme d'accueil	3
1.2.3 Idée maîtresse	4
1.2.4 Description des documents de travail	4
1.2.5 Règles générales appliquées	4
1.3 Problématique	5
1.4 Étude de l'existant	5
1.4.1 Les méthodes de facturation classiques	5
1.4.2 Les méthodes de facturation avancées	7
1.5 Solutions existantes	8
1.5.1 Solutions existantes en Tunisie	8
1.5.1.1 Swivre	8
1.5.2 Solutions existantes à l'étranger	9
1.5.2.1 Odoo	9
1.5.2.2 EVOLIZ	10
1.5.2.3 Sage Business Cloud Facturation	11
1.5.2.4 Critique de l'existante	12
1.5.3 Solution proposée	12
1.5.3.1 Description textuelle détaillée	13
1.5.3.2 Description Visuelle	15

TABLE DES MATIÈRES

1.6 Méthodologie de développement	15
1.6.1 Processus unifié	15
1.6.1.1 Représentation du Processus unifié	16
1.6.2 Two Track Unified Process(2TUP)	17
1.7 Diagramme de GANTT	19
1.8 CONCLUSION	20
2 Identification analyse des besoins	21
2.1 Introduction	22
2.2 Analyse et identification des besoins :	22
2.2.1 Identification des acteurs :	22
2.2.2 Besoins fonctionnels	22
2.2.2.1 Besoins fonctionnels pour « l'administrateur »	22
2.2.2.2 Besoins fonctionnels pour « le prestataire »	23
2.2.2.3 Diagramme de cas d'utilisation globale	24
2.2.3 Besoins non fonctionnels	25
2.2.4 Besoins technique	25
2.2.4.1 L'environnement logiciel	25
2.3 Modèle Conceptuel	29
2.3.1 Modèle BFF	30
2.3.2 Fonctionnement du BFF	30
2.4 CONCLUSION	31
3 Conception	32
3.1 Introduction	33
3.2 Conception Générique	33
3.3 Conception détaillée	33
3.3.1 Langage UML	34
3.3.2 Les vues de Kruchten	35
3.3.3 Diagramme de cas d'utilisation	37
3.3.4 Diagrammes de séquence	38
3.3.4.1 Diagramme de séquence « S'Authentifier »	38
3.3.4.2 Diagramme de séquence « Gérer client » et « Gérer produit »	39
3.3.4.3 Diagramme de séquence « Gérer Facture »	40
3.3.4.4 Diagramme de séquence « Gérer entreprise »	41
3.3.5 Diagramme de classe	41

TABLE DES MATIÈRES

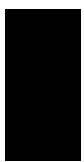
3.3.6	Diagramme de composants	42
3.4	Conculusion	43
4	Réalisation	44
4.1	Introduction	45
4.2	Présentation de la solution WEB	45
4.2.1	Interface d'authentification	45
4.2.2	Interface utilisateur du tableau de bord	46
4.2.3	Interfaces de gestion des entreprises	46
4.2.3.1	Liste des entreprises ajoutées	47
4.2.3.2	Le formulaire d'ajout d'une entreprise	48
4.2.3.3	Validation des données	49
4.2.3.4	50
4.2.3.5	La gestion de services	50
4.2.4	Interface de gestion de clients	51
4.2.5	Interface de gestion de produits	52
4.2.6	Interfaces de gestion des factures et de génération de PDF	53
4.2.6.1	Liste des factures créées	53
4.2.6.2	Validation des factures	53
4.2.6.3	Processus d'ajout pour une facture	54
4.2.6.4	Le PDF généré	55
4.2.7	Interface de visualisation des statistiques	56
4.3	Conclusion	56
CONCLUSION GÉNÉRALE		57
BIBLIOGRAPHIE		57

LISTE DES FIGURES

1.1	Logiciel de facturation tunisien : Swivre.	8
1.2	Logiciel de facturation : Odoo.	9
1.3	Logiciel de facturation : EVOLIZ.	10
1.4	Logiciel de facturation : Sage.	11
1.5	Pictogramme : Description visuelle	15
1.6	Phases du développement UP	17
1.7	2TUP	18
1.8	Diagramme de GANTT	19
2.1	Diagramme de cas d'utilisation globale	24
2.2	Angular 15	26
2.3	Spring boot	26
2.4	Keycloak	27
2.5	PostgreSQL	27
2.6	Amazon Web Services	28
2.7	GitLab	28
2.8	IntelliJ IDEA	29
2.9	BFF Design Pattern	30
3.1	Maquette de l'application	33
3.2	Les vues de kruchten	35
3.3	Diagramme de cas d'utilisation détaillé	37
3.4	Diagramme de séquence : « S'Authentifier »	38
3.5	Diagramme de séquence « Gérer client »,« Gérer produit »	39
3.6	Diagramme de séquence « Gérer facture »	40
3.7	Diagramme de séquence : « Gérer entreprise »	41
3.8	Diagramme de classe	42
3.9	Diagramme de classe	43
4.1	Interface d'authentification fournit par keycloak	45

LISTE DES FIGURES

4.2	Interface utilisateur du tableau de bord	46
4.3	Création des roles géré par keycloak	47
4.4	Affectation des roles géré par keycloak	47
4.5	Interface de consultation des entreprises	48
4.6	Processus de la création d'une entreprise	49
4.7	Processus de la création : Validation des données	50
4.8	Processus du suppression des entreprises	50
4.9	Gestion des services	51
4.10	Gestion de clients	52
4.11	Gestion de produits	52
4.12	Interface de consultation des factures	53
4.13	Validation des facture	54
4.14	Formulaire de création pour les factures	55
4.15	Le PDF généré d'une facture	55
4.16	Interface des statistiques	56



LISTE DES ABRÉVIATIONS

BFF : Backend For Frontend

CI/CD : Continuous Integration/Continuous Delivery

ERP : Enterprise Resource Planning

JWT : JSON Web Token

PME : Petite et Moyenne Entreprise

PDF : Portable Document Format

SIREN : Système d'Identification du Répertoire des Entreprises

SIRET : Système d'Identification du Répertoire des Etablissements

2TUP : Two Tracks Unified Process

TPE : Très Petite Entreprise

UML : Unified Modeling Language

UP : Unified Process



INTRODUCTION GÉNÉRALE

De nos jours, de nombreuses entreprises se tournent vers les applications web pour simplifier leur gestion. Ces applications facilitent les tâches administratives, les ressources humaines, la gestion de la production, etc. Elles centralisent les données et favorisent la collaboration entre les équipes, améliorant ainsi la productivité. Les applications web sont simples, rapides et économiques, ce qui en fait une tendance incontournable pour les entreprises soucieuses d'économiser du temps et de l'argent.

En particulier, les applications web de facturation sont essentielles pour la gestion financière des entreprises. Elles simplifient les processus de facturation, de suivi des paiements et de gestion des factures en un seul endroit, réduisant les erreurs et les retards. Ces applications automatisent des tâches fastidieuses, facilitent le partage des données avec les clients et les partenaires, génèrent des rapports financiers détaillés et favorisent la conformité fiscale. En utilisant une application de facturation en ligne, les entreprises améliorent leur processus de facturation, accèdent à distance à leurs données, suivent leurs finances en temps réel et assurent la sécurité de leurs données.

Dans ce cadre, la société « INFOGO »[1], en tant qu'entreprise de services, ressent un besoin crucial d'informatiser son activité commerciale centrée sur le suivi de ses clients et la facturation des prestations fournies. Pour répondre à ce besoin, un stage de fin d'études nous a été confié qui a pour but de développer une application opérationnelle pour faciliter la gestion interne de l'entreprise. Ce produit logiciel standard pourra également être utilisé par des sociétés similaires, apportant ainsi des avantages supplémentaires à « INFOGO ».

Ce rapport synthétise notre stage de fin d'études en quatre sections distinctes. La première section expose le contexte et la méthodologie adoptée, incluant une analyse approfondie de

INTRODUCTION GÉNÉRALE

l'existant. La deuxième section se concentre sur l'analyse fonctionnelle, technique et l'utilisation de design patterns dans le projet. La troisième section aborde en détail la conception de la solution envisagée. Enfin, la dernière section présente les aspects concrets de la réalisation du projet.

Contexte du projet

Sommaire

1.1	Introduction	3
1.2	Contexte générale	3
1.2.1	Cadre du projet	3
1.2.2	Présentation de l'organisme d'accueil	3
1.2.3	Idée maitresse	4
1.2.4	Description des documents de travail	4
1.2.5	Règles générales appliquées	4
1.3	Problématique	5
1.4	Étude de l'existant	5
1.4.1	Les méthodes de facturation classiques	5
1.4.2	Les méthodes de facturation avancées	7
1.5	Solutions existantes	8
1.5.1	Solutions existantes en Tunisie	8
1.5.2	Solutions existantes à l'étranger	9
1.5.3	Solution proposée	12
1.6	Méthodologie de développement	15
1.6.1	Processus unifié	15
1.6.2	Two Track Unified Process(2TUP)	17
1.7	Diagramme de GANTT	19
1.8	CONCLUSION	20

1.1 Introduction

Dans ce premier chapitre introductif, nous commençons par présenter le cadre de notre projet de fin d'études, en mettant en évidence le contexte et la problématique qui ont motivé notre travail. Ensuite, nous abordons la solution proposée, en nous appuyant sur une étude préalable des travaux existants. Enfin, nous détaillons la méthodologie de travail que nous avons adoptée.

1.2 Contexte générale

1.2.1 Cadre du projet

Ce travail s'inscrit dans le cadre de notre projet de fin d'études en vue de l'obtention du diplôme national de licence en sciences informatiques au sein de l'institut supérieur d'informatique et de mathématique de Monastir. Il consiste à concevoir, réaliser et déployer une application web de facturation des services. Cette application est destinée aux PME, aux start-ups et aux entrepreneurs libres. Il offre un ensemble unique de fonctionnalités qui sont conçus pour simplifier la gestion de leurs activités commerciales.

1.2.2 Présentation de l'organisme d'accueil

INFOGO est une société spécialisée dans le secteur d'activité du conseil en systèmes et logiciels informatiques, l'entreprise s'adresse principalement aux grandes entreprises spécialisées dans le domaine de la finance, comme les banques et les assurances, afin de les accompagner dans la transformation Devops de leurs systèmes d'informations, ce qui permet d'avoir un parc applicatif complètement automatisé, qui implémente les bonnes pratiques de sécurité. Elle se caractérise par son approche collaborative avec ses clients, sa réactivité et sa flexibilité. Les différents services proposés :

- Automatisation des chaînes CI/CD.

- Installation et gestions des clusters Kubernetes.
- Migration des systèmes d'informations vers les conteneurs.
- Audit de sécurité.

1.2.3 Idée maîtresse

Suite à l'adoption d'une nouvelle loi en France en mars 2023, qui oblige la généralisation de la facture électronique dans tous les secteurs d'activité, un stage de fin d'études nous a été confié. L'entreprise Info Go a saisi cette opportunité pour développer une application de facturation qui lui permettrait de rationaliser ses processus internes et de proposer un produit innovant pouvant être utilisé par d'autres entreprises. Ce contexte nous a ainsi permis de travailler sur un projet concret et d'apporter notre contribution à l'effort de modernisation des entreprises françaises.

1.2.4 Description des documents de travail

- Une facture est un document émis par une entreprise à l'attention d'un client, afin de récapituler les biens ou services vendus et le montant à payer.
- Un avoir est un document comptable annulant une facture validée, partiellement ou en totalité. Il est émis en cas de retour de produit ou de réduction accordée au client après l'émission de la facture initiale.
- Un devis est un document commercial fournissant une estimation de prix pour des biens ou services proposés par l'entreprise aux clients. Il est généralement établi avant la facturation pour informer le client du coût prévisionnel de la prestation demandée.

1.2.5 Règles générales appliquées

Les différentes règles proposées sont les suivantes :

- Un client ou une entreprise ne peut être ajouté(e) qu'après la validation de ses coordonnées juridiques (SIREN, SIRET, N°TVA, etc.).

- La numérotation des factures et des retours doit être successive.
- Les numéros des factures et des avoirs sont générés lors de leur validation, seules les factures et les avoirs validés admettent un numéro.
- Une facture non validée ne peut pas être transformée en avoir.
- Une facture ou un avoir validé(e) ne peut être supprimé(e) ou modifié(e).
- La génération des PDF n'est possible qu'après la validation.
- Une facture peut être créée à partir d'un produit ou à partir d'un devis établi.

1.3 Problématique

Un logiciel de facturation peut présenter des inconvénients s'il n'est pas adapté aux besoins spécifiques de l'entreprise ou de l'utilisateur. Par exemple, un logiciel de facturation avec des fonctionnalités de comptabilité approfondie peut être inutile pour un auto-entrepreneur avec une comptabilité allégée. En outre, certains logiciels peuvent être considérés coûteux pour les indépendants ou les micro-entrepreneurs, et leur utilisation peut nécessiter une formation préalable. La société « INFO GO » étant une institution qui vend du service, éprouve un besoin urgent d'informatiser son activité commerciale basée essentiellement sur le suivi général de ses clients et la facturation des prestations rendues et par la même occasion la faire bénéficier d'un produit logiciel standard utilisable par les sociétés similaires.

1.4 Étude de l'existant

Dans cette section, nous allons nous concentrer sur les méthodes de facturation employées par les entreprises et les particuliers, en les classant en deux catégories distinctes.

1.4.1 Les méthodes de facturation classiques

Facturation par papier : La facturation en papier est une méthode traditionnelle de gestion des factures qui implique l'impression et l'envoi postal de documents papier aux clients. Bien

CONTEXTE DU PROJET

que cette méthode ait été largement utilisée dans le passé, elle présente aujourd’hui des avantages et des inconvénients qui doivent être pris en compte lors de la sélection de la méthode de facturation la plus appropriée pour une entreprise.

Avantages :

- Accessibilité : La facturation par papier ne nécessite pas de connexion Internet ou d’ordinateur, ce qui la rend facilement accessible à tous.
- Pas de coût supplémentaire : La facturation par papier ne nécessite pas d’acheter un logiciel ou de payer des frais d’abonnement, ce qui peut être une option rentable pour les petites entreprises.
- Preuve tangible : Les factures imprimées fournissent une preuve tangible de la transaction pour les deux parties.

Inconvénients :

- Coûts élevés : Les coûts d’impression, de papier et d’envoi postal peuvent être élevés, surtout pour les entreprises avec un grand nombre de clients.
- Manque d’efficacité : La facturation par papier peut être lente et peut entraîner des erreurs de saisie de données, ce qui peut ralentir le processus de paiement et causer des retards.
- Impact environnemental de l’impression excessive de factures : consommation de ressources naturelles et génération de déchets.

Facturation Excel : La facturation en utilisant des feuilles de calcul Excel est une méthode courante pour les petites entreprises et les indépendants qui cherchent une alternative à la facturation papier. Bien qu’elle offre une certaine flexibilité, cette méthode présente des avantages et des inconvénients qui doivent être considérés.

Avantages :

- Accessibilité : Excel est largement utilisé et accessible, ce qui rend la facturation Excel facile à mettre en place pour de nombreuses entreprises.
- Flexibilité : La facturation Excel permet aux entreprises de personnaliser facilement leur modèle de facture en fonction de leurs besoins spécifiques.

CONTEXTE DU PROJET

- Mises à jour en temps réel : Excel permet de suivre en temps réel les paiements, les soldes impayés et les comptes clients, ce qui peut être utile pour une gestion plus efficace des flux de trésorerie.

Inconvénients :

- Les erreurs de saisie de données : Ces erreurs peuvent conduire à des erreurs de facturation, affectant ainsi la gestion adéquate des factures.
- Manque de fonctionnalités : Excel n'est pas aussi complet qu'un logiciel spécialisé en facturation en termes de fonctionnalités de gestion, ce qui peut poser des problèmes aux entreprises ayant des besoins plus complexes.
- Difficulté à l'échelle : La facturation Excel peut être difficile à gérer pour les entreprises qui ont un grand nombre de clients.

1.4.2 Les méthodes de facturation avancées

Facturation En-ligne : La facturation en ligne est devenue une méthode populaire de gestion des factures, offrant une alternative pratique et efficace aux méthodes traditionnelles.

Avantages :

- Efficacité : La facturation en ligne est rapide et efficace. Les factures sont envoyées rapidement et les paiements peuvent être reçus instantanément.
- Flexibilité : Les outils de facturation en ligne offrent une plus grande flexibilité et productivité aux entreprises grâce à leurs fonctionnalités supplémentaires.
- Les outils de facturation en ligne sont accessibles depuis n'importe quel appareil connecté à Internet, offrant une solution pratique pour la gestion des factures en déplacement.

Inconvénients :

- Coûts élevés : Les logiciels de facturation peuvent représenter un investissement important pour les petites entreprises ou les travailleurs indépendants ayant un budget limité.
- Sécurité : La facturation en ligne peut présenter des risques de sécurité, notamment en ce qui concerne la confidentialité des données clients et la sécurité des paiements.

- Formation : L'utilisation d'un nouvel outil de facturation en ligne peut nécessiter une formation préalable pour les employés, ce qui peut être chronophage et coûteux pour les petites entreprises.

1.5 Solutions existantes

Nous avons élargi notre perspective sur les plateformes cibles en étudiant l'existant sur les marchés internationaux. Cependant, notre solution sera spécifiquement conçue pour le marché français. Nous avons examiné les marchés internationaux existants dans le but de créer des solutions à l'échelle mondiale et de favoriser l'expansion future de la plateforme cible.

1.5.1 Solutions existantes en Tunisie

1.5.1.1 Swivre

Swivre[2] est un produit de facturation et de devis tunisien conçu pour aider les entreprises à gérer leur processus de facturation et à générer des fiches de paie. Étant un produit local, il est spécialement conçu pour répondre aux besoins des entreprises tunisiennes et offre un support local pour aider les utilisateurs en cas de besoin.

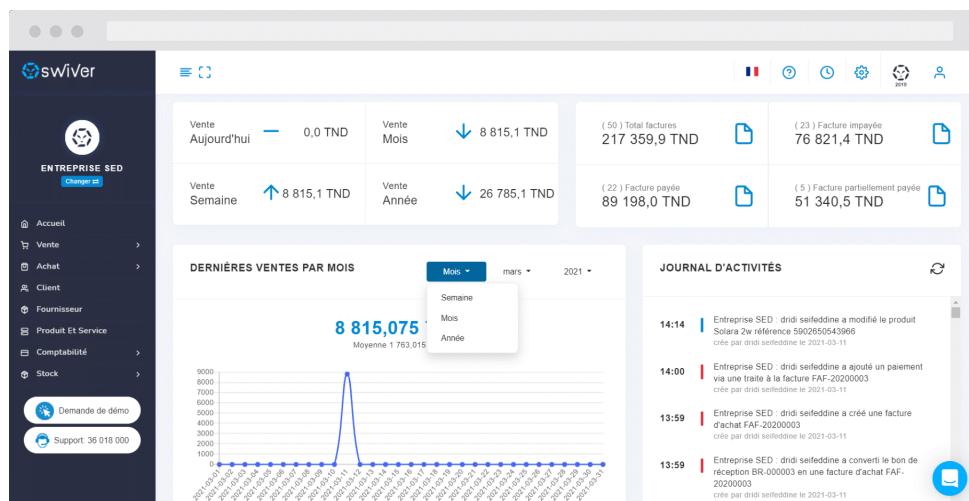


FIGURE 1.1 – Logiciel de facturation tunisien : Swivre.

Avantages :

- Interface utilisateur intuitive : Swivre offre une interface simple et facile à utiliser, adaptée aux utilisateurs novices en informatique.
- Adaptabilité aux besoins locaux : Swivre est développé en Tunisie pour répondre aux besoins spécifiques des entreprises tunisiennes.

Inconvénients :

- Les mises à jour de Swivre peuvent causer des bugs qui affectent l'expérience utilisateur malgré leur but d'améliorer la performance de l'application.
- Swivre propose de nombreuses fonctionnalités, mais certaines peuvent ne pas être adaptées aux besoins spécifiques de certaines entreprises.

1.5.2 Solutions existantes à l'étranger

1.5.2.1 Odoo

Odoo[5] est un des logiciels de gestion les plus utilisés au monde, il est donc tout naturel que nous le mentionnions ici. Ce logiciel open source de 2004 est très complet et sera largement suffisant pour une entreprise de taille moyenne.

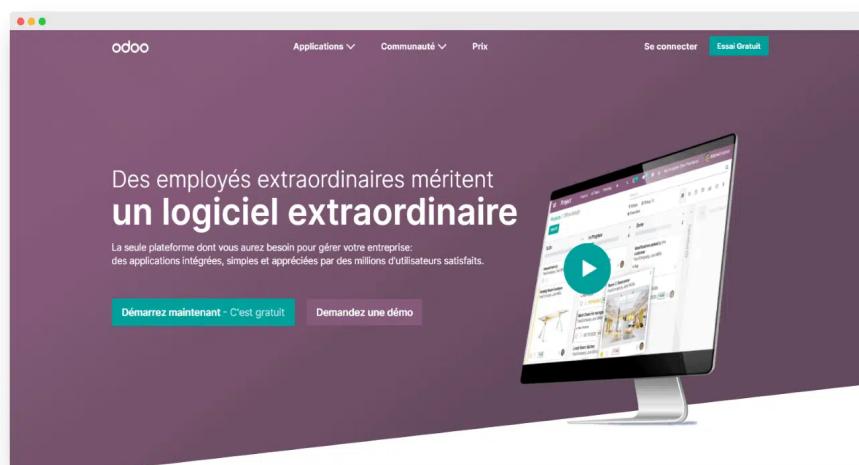


FIGURE 1.2 – Logiciel de facturation : Odoo.

Avantages :

- Profondeur fonctionnelle : Odoo est plus qu'un simple logiciel de facturation : c'est un ERP complet qui vous permet d'ajouter les options dont vous avez réellement besoin.
- Tarification : Le rapport qualité/prix d'Odoo est particulièrement bon, d'autant que vous adaptez les applications nécessaires en fonction de vos besoins.

Inconvénients :

- Mise en place laborieuse : Pour une utilisation assez standard, la mise en place est rapide, en revanche pour personnaliser le logiciel en fonction de vos besoins, mieux vaut être accompagné afin de ne pas vous perdre dans les paramétrages.
- Service client : Un peu difficile à joindre, notamment lorsqu'il s'agit d'une question d'ordre technique.

1.5.2.2 EVOLIZ

Evoliz[3] est un logiciel de facturation qui s'adresse à tous les entrepreneurs, quel que soit leur statut social. Evoliz propose un outil de gestion complet en mettant l'accent sur l'automatisation de certains processus de gestion fastidieux. L'outil permet également la gestion des stocks, ce qui en fait une solution intéressante pour les e-commerçants.

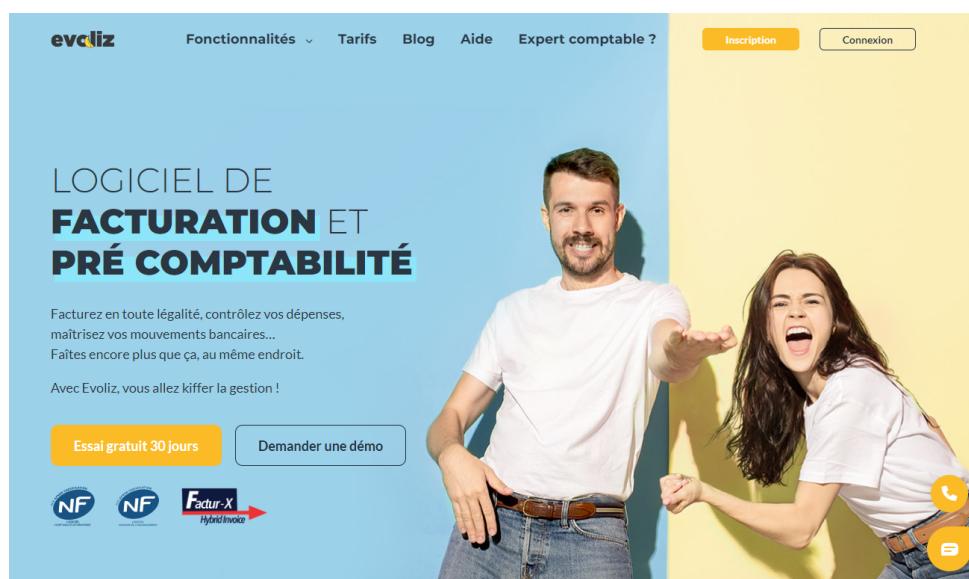


FIGURE 1.3 – Logiciel de facturation : EVOLIZ.

CONTEXTE DU PROJET

Avantages :

- 100% français, personnalisable, évolutif et collaboratif.
- Outil complet de gestion : Facturation mais aussi gestion des stocks, des frais, de la relation client et des fournisseurs.

Inconvénients :

- Configuration : Long à configurer la première fois.
- TVA : Pas de pré-remplissage de déclaration de TVA.

1.5.2.3 Sage Business Cloud Facturation

Sage, anciennement Sage One[4] est l'outil le plus léger de la suite Sage. Ce logiciel de facturation permet aux TPE(s) et PME(s) de simplifier leur gestion de la comptabilité et de la facturation, afin de se concentrer sur leur activité principale.

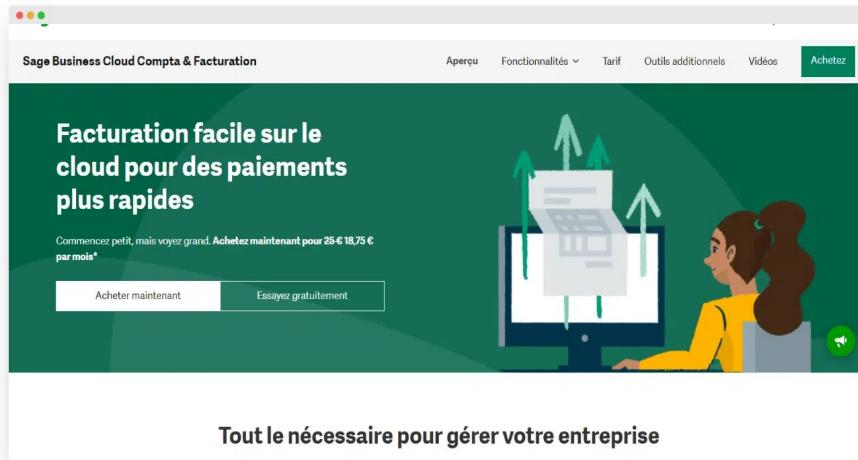


FIGURE 1.4 – Logiciel de facturation : Sage.

Avantages :

- Facile d'utilisation : Une plateforme facile à prendre en main.
- Profondeur fonctionnelle : Sage possède de nombreuses fonctionnalités, qui conviendront à beaucoup de scénarios différents.

Inconvénients :

- Reporting un peu léger.
- Manque de réactivité du service client.
- Moins puissant que d'autres logiciels au même prix.

1.5.2.4 Critique de l'existantes

Il est possible de catégoriser les résultats de l'analyse des applications web mentionnées précédemment en fonction de six critères qui ont été pris en compte dans le processus d'évaluation de ces applications.

1. Facilité d'utilisation : La facilité d'utilisation de l'interface utilisateur, la simplicité de la navigation et la convivialité globale de l'application.
2. Fonctionnalités de facturation : La variété et la qualité des fonctionnalités liées à la facturation.
3. Sécurité des données : La sécurité et la confidentialité des données de l'entreprise.
4. Support client : La qualité et la disponibilité du support client.
5. Coût : Le coût total de l'application, y compris les frais d'abonnement.
6. Rapportage : Processus de création et de présentation de rapports d'analyse et de synthèse de données, généralement à partir de données brutes collectées dans diverses sources.

	SWIVRE	ODOO	EVOLIZ	SAGE
Facilité d'utilisation	Facile	Compliqué	Compliqué	Facile
Coût	Moyenne	Moyenne	Élevé	Élevé
Fonctionnalités	Limité	Complet	Complet	Complet
Support client	Insatisfaisant	Insatisfaisant	Satisfaisant	Insatisfaisant
Rapportage	Pas de rapportage	Pas précis	Pas précis	Pas précis
Sécurité des données	Basique	Élevé	Élevé	Élevé

TABLE 1.1 – Tableau comparatif des différents logiciels de facturation

1.5.3 Solution proposée

Dans cette section du premier chapitre, nous présenterons notre solution proposée basée sur les recommandations et les conclusions de notre étude comparative. Nous avons réalisé

une analyse détaillée de différents logiciels en se basant sur des critères clés. Cette approche nous a permis d'identifier les points forts et les points faibles de chaque logiciel, ainsi que les besoins et les préférences des utilisateurs.

1.5.3.1 Description textuelle détaillée

Notre solution est conçue pour offrir une expérience de facturation simplifiée, tout en garantissant des fonctionnalités avancées pour répondre aux besoins des utilisateurs. Notre application devrait être en mesure de satisfaire les besoins suivants :

a) Au niveau de l'interface utilisateur

1. Conception minimaliste : L'interface de l'application doit être conçue de manière minimale pour une expérience utilisateur agréable et intuitive.
2. Clarté : Les éléments de l'interface doivent être clairement présentés pour aider les utilisateurs à comprendre facilement les fonctionnalités de l'application.
3. Assistance utilisateur : L'interface doit offrir des aides visuelles et textuelles pour guider les utilisateurs dans leur utilisation de l'application et faciliter leur apprentissage.
4. Facilité d'utilisation : L'interface doit être facile à utiliser pour permettre aux utilisateurs de naviguer facilement dans l'application et d'accomplir leurs tâches de manière efficace.

b) Au niveau de fonctionnalités

Notre application doit respecter les réglementations gouvernementales tout en offrant une expérience utilisateur simplifiée pour faciliter l'utilisation des fonctionnalités essentielles liées à la gestion du processus de facturation.

1. Gestion des clients : Gestion sécurisée de la base de données clients pour l'utilisateur.
2. Gestion des services : Offrir à l'utilisateur la possibilité de présenter facilement ses services, avec une indication claire des prix et du taux de TVA correspondant pour chaque service.

3. Gestion des factures : La fonction de facturation doit inclure des fonctionnalités telles que le calcul automatique des montants de TVA, la génération automatique de factures au format PDF.
4. Gestion des avoirs : L'application doit permettre d'annuler ou de corriger une facture tout en respectant la numérotation des documents.
5. Espace statistique : Permettre à l'utilisateur de visualiser son chiffre d'affaires via un espace statistique dédié.

c) Au niveau de sécurité

1. Authentification sécurisée : L'application doit offrir un processus d'authentification sécurisé pour garantir la confidentialité et l'intégrité des données de l'utilisateur.
2. Gestion des autorisations : L'application doit permettre la gestion fine des autorisations d'accès aux différentes fonctionnalités pour éviter tout accès non autorisé.
3. Accès à tout moment : L'application doit être disponible à tout moment pour permettre à l'utilisateur de se connecter en toute sécurité et d'accéder à ses données.

d) Au niveau de service client

1. Support client : L'application doit offrir un support client efficace pour répondre aux demandes de renseignements et de réclamation des utilisateurs.
2. Maintenance : L'application doit être maintenue régulièrement pour garantir son bon fonctionnement et sa disponibilité à tout moment.
3. Amélioration continue : L'application doit être continuellement améliorée en fonction des retours des utilisateurs pour répondre au mieux à leurs besoins et attentes.
4. Transparence : L'application doit être transparente envers les utilisateurs en ce qui concerne les mises à jour, les changements de fonctionnalités et les problèmes techniques éventuels.

1.5.3.2 Description Visuelle



FIGURE 1.5 – Pictogramme : Description visuelle

1.6 Méthodologie de développement

Il est très important d'avoir une méthode de travail bien définie pour organiser les tâches efficacement et faciliter la gestion et le suivi du projet ou de la tâche. Par la présente, il est important de mentionner que l'entreprise a imposé l'utilisation de la méthodologie 2TUP (Two-Track Unified Process) pour ce projet et par conséquence, notre équipe doit suivre les processus et les pratiques recommandés par la méthodologie 2TUP, conformément aux directives de l'entreprise.

1.6.1 Processus unifié

Avant de commencer à présenter l'approche 2TUP, il est important de passer en revue le Processus Unifié (UP) dont il est dérivé. En effet, l'UP est une méthodologie de développement logiciel itérative et incrémentale largement reconnue dans le domaine du génie logiciel. Comprendre les principes et les bonnes pratiques de l'UP permettra de mieux appréhender

les fondements de l'approche 2TUP et de renforcer ainsi le choix de cette dernière. Dans cette section, nous allons donc décrire les différentes phases du cycle de vie de l'UP et expliquer comment elles sont appliquées dans le contexte de l'approche 2TUP.

1.6.1.1 Représentation du Processus unifié

Le Processus Unifié (UP)[6] est une méthodologie itérative et incrémentale qui améliore la qualité et la productivité du développement logiciel. Elle favorise la collaboration et la communication entre les acteurs du projet, tout en utilisant des bonnes pratiques et des normes reconnues pour assurer la qualité et la fiabilité du logiciel développé.

Le processus unifié (UP) suit un modèle de cycle de vie qui est divisé en quatre phases principales :

- La phase d’Inception : elle consiste à définir la portée du projet, à identifier les objectifs et les risques potentiels, à élaborer une stratégie de développement préliminaire, à établir un calendrier approximatif et à déterminer la faisabilité du projet. À la fin de cette phase, une décision est prise pour continuer ou non le projet.
- La phase d’Elaboration : elle vise à élaborer une architecture logicielle de base, à définir les exigences fonctionnelles et non fonctionnelles, à établir une planification détaillée, à évaluer les risques et à établir un plan de test. Cette phase s’achève par la validation de l’architecture du système.
- La phase de Construction : elle consiste à développer les fonctionnalités du système, à tester les composants logiciels et à intégrer les différentes parties du système. Les tests de validation sont effectués à chaque itération de développement, ce qui permet de corriger les erreurs et d’améliorer le système au fil du temps.
- La phase de Transition : elle vise à installer, à déployer et à livrer le système aux utilisateurs finaux, à effectuer des tests finaux et à former les utilisateurs et le personnel d’exploitation. Cette phase comprend également le suivi des performances du système et la correction des défauts éventuels.

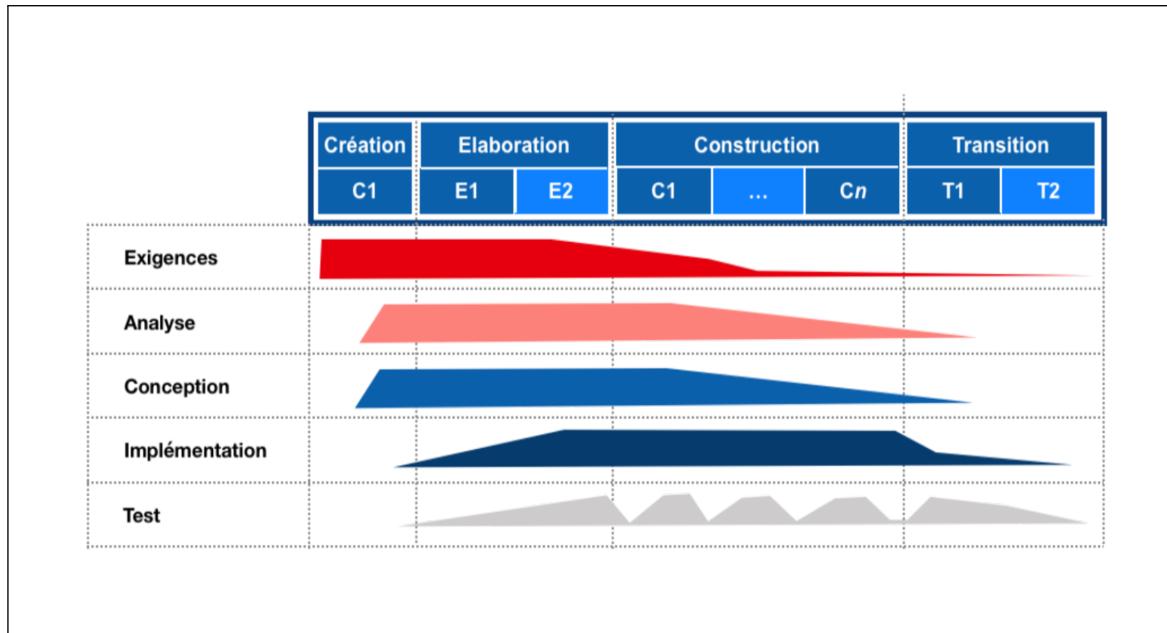


FIGURE 1.6 – Phases du développement UP

1.6.2 Two Track Unified Process(2TUP)

Le 2TUP[7] , également connu sous le nom de Two-Track Unified Process, est une déclinaison du Processus Unifié, qui est une méthodologie de développement logiciel itérative et incrémentale. Cette approche permet de diviser le projet en plusieurs cycles de développement, chacun délivrant une version fonctionnelle du produit final.

Le 2TUP ajoute une dimension supplémentaire à cette approche en permettant de séparer le développement des fonctionnalités principales et des fonctionnalités de qualité en deux pistes distinctes. Cette séparation permet d'optimiser le processus de développement et de garantir une qualité supérieure pour l'application.

- La première piste, appelée "Track 1", est consacrée aux fonctionnalités principales, c'est-à-dire celles qui sont essentielles au bon fonctionnement de l'application. Cette piste se concentre sur le développement des fonctionnalités les plus importantes et les plus critiques en priorité, tout en intégrant les feedbacks et les tests au fur et à mesure.
- La deuxième piste, appelée "Track 2", est quant à elle dédiée aux fonctionnalités de qualité, telles que la sécurité, la performance et la convivialité de l'interface

utilisateur. Ces fonctionnalités sont développées en parallèle de la piste principale afin d'assurer une qualité optimale de l'application. Cette approche permet également de réduire les risques liés à la sécurité, à la performance et à la fiabilité de l'application.

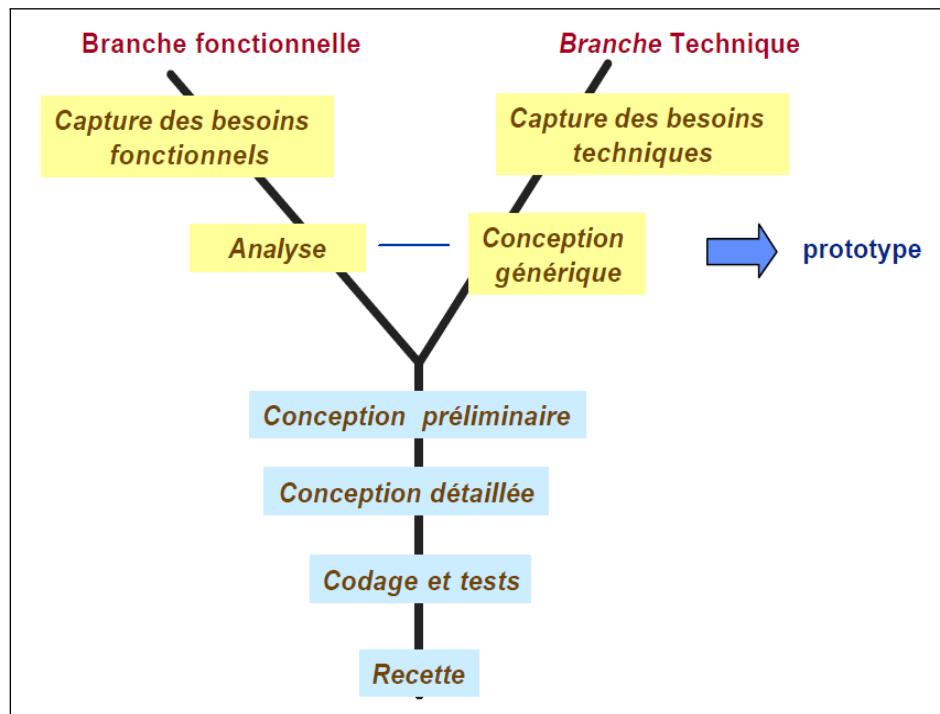


FIGURE 1.7 – 2TUP

- 1 **L'étude préliminaire :** cette étape consiste à décrire le service et les cas d'utilisation principaux. Elle permet d'avoir une première version de la spécification générale qui servira de référence tout au long du développement.
- 2 **La capture des besoins fonctionnels :** cette étape permet de définir les fonctionnalités du service à travers une spécification générale et détaillée qui précise les traitements associés à chaque scénario de cas d'utilisation. Cette spécification prend en compte les contraintes fonctionnelles et non fonctionnelles.
- 3 **L'analyse :** cette étape permet d'analyser simultanément les données et les traitements à effectuer pour répondre aux besoins fonctionnels identifiés dans l'étape précédente.
- 4 **La capture des besoins techniques :** cette étape permet de satisfaire les contraintes techniques du cahier des charges en identifiant les solutions techniques les plus adaptées pour répondre aux attentes du client.

- 5 **La conception générique** : cette étape consiste à définir le "comment faire" en identifiant les solutions techniques et les modèles d'architecture logicielle les plus adaptés pour réaliser les fonctionnalités définies dans la spécification.
- 6 **La conception détaillée** : cette étape permet de préciser l'implémentation technique de l'application en fusionnant la spécification détaillée et la conception générique. On y trouve notamment le schéma de base de données, les diagrammes de classes et les diagrammes de séquence qui détaillent les interactions entre les différents composants du logiciel.
- 7 **Le Codage et les Tests** : cette étape consiste à réaliser l'application et à la tester pour s'assurer que les fonctionnalités ont été correctement implémentées. Elle permet également de définir la stratégie de validation et de tester l'application dans un environnement d'exécution adapté.

1.7 Diagramme de GANTT

La planification occupe une place essentielle parmi les étapes préliminaires d'un projet. Elle implique l'identification et la séquence des tâches du projet, ainsi que l'estimation de leurs charges respectives. Le diagramme de GANTT s'avère un outil précieux pour planifier le projet et suivre son avancement.

La Figure 1.8 présente notre projet à travers un diagramme de GANTT, offrant une vue claire et organisée de la planification.

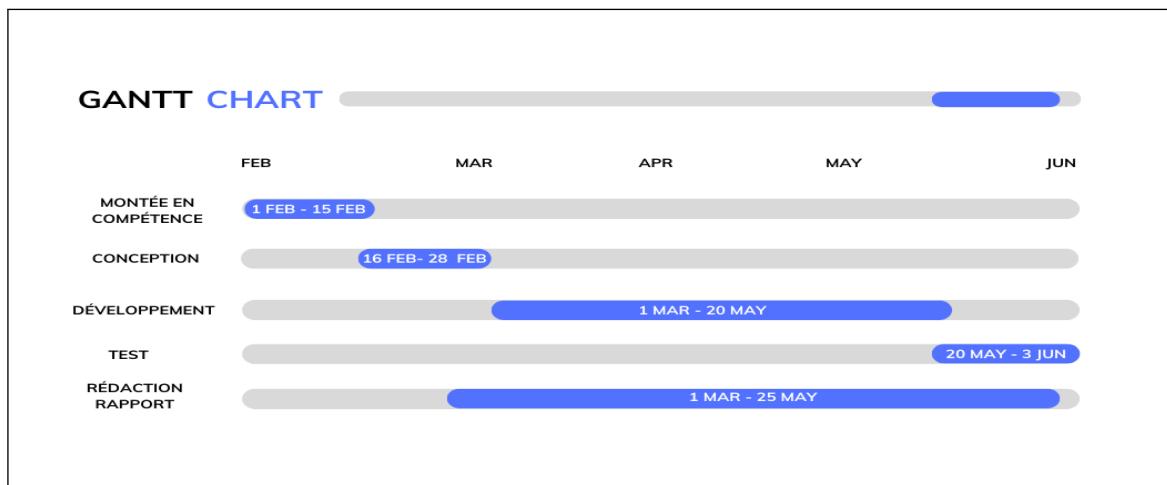


FIGURE 1.8 – Diagramme de GANTT

1.8 CONCLUSION

Dans ce chapitre, nous avons exposé le cadre général de notre projet ainsi que son objectif. Nous avons procédé à une étude approfondie de l'état actuel de la technologie afin de concevoir une application de qualité répondant aux exigences du marché. Enfin, nous avons présenté la méthodologie de travail adaptée pour le développement et la réalisation du projet.

Identification analyse des besoins

Sommaire

2.1	Introduction	22
2.2	Analyse et identification des besoins :	22
2.2.1	Identification des acteurs :	22
2.2.2	Besoins fonctionnels	22
2.2.3	Besoins non fonctionnels	25
2.2.4	Besoins technique	25
2.3	Modèle Conceptuel	29
2.3.1	Modèle BFF	30
2.3.2	Fonctionnement du BFF	30
2.4	CONCLUSION	31

2.1 Introduction

Après avoir présenté une vue d'ensemble de notre projet dans le chapitre précédent, ce nouveau chapitre se concentre sur la spécification et l'analyse des besoins. L'objectif est de définir et d'analyser les caractéristiques fonctionnelles et non fonctionnelles requises pour notre application.

2.2 Analyse et identification des besoins :

2.2.1 Identification des acteurs :

Un acteur est une entité externe qui définit le rôle joué par un utilisateur, humain ou non humain, qui interagit avec un système interactif. Notre système comporte les acteurs suivants :

- Administrateur : Il désigne le rôle du propriétaire de l'application, celui qui est responsable pour la conception et la maintenance de l'application.
- Prestataire : Il désigne le rôle de l'utilisateur de l'application, celui qui va utiliser l'application pour gérer son propre processus de financement.

2.2.2 Besoins fonctionnels

Les besoins fonctionnels sont les fonctionnalités principales que l'application doit fournir pour répondre aux besoins de ses utilisateurs.

2.2.2.1 Besoins fonctionnels pour « l'administrateur »

- « Assurer l'assistance technique auprès des utilisateurs » : L'administrateur doit être toujours à l'écoute aux questions de ses clients, les orienter sur les utilisations possibles et résoudre leurs problèmes.
- « Sécuriser l'application » : vise à garantir une connexion sécurisée et conviviale pour les utilisateurs en mettant en place des mesures de sécurité telles qu'un système

d'authentification solide, la surveillance des activités et l'application de politiques de sécurité strictes.

- « Gérer entreprise » : Cette fonctionnalité permet à l'administrateur de se connecter en tant qu'utilisateur pour d'autres entreprises, d'accéder à divers services de l'application. L'administrateur peut créer des rôles et les attribuer aux clients pour qu'ils puissent gérer leurs propres entreprises.

À noter que : l'administrateur hérite du prestataire : Cette relation d'héritage, entre deux acteurs, exprime le fait que l'acteur du côté opposé à la pointe de la flèche est une sorte de sous-type de l'acteur général. Dans ce cas, l'administrateur est spécialisé au sens où il peut réaliser tout ce que l'acteur plus général (le prestataire) peut réaliser, plus d'autres fonctionnalités supplémentaires.

2.2.2.2 Besoins fonctionnels pour « le prestataire »

- « S'authentifier » : L'authentification désigne le mécanisme de contrôle d'accès permettant de vérifier l'identité d'un utilisateur souhaitant accéder à une application.
- « Gérer facture » : La gestion de factures consiste à garantir la traçabilité des transactions effectuées entre commerçants, en suivant un ensemble de règles à respecter lors de la facturation.
- « Gérer avoirs » : Permet à l'utilisateur de convertir une facture validée en un avoir, dans le but de corriger une erreur déjà mentionnée ou d'annuler la facture.
- « Gérer client » : permet à l'utilisateur de gérer sa base de données clients en ajoutant, modifiant ou supprimant les informations relatives à ces derniers. Cela permet une gestion efficace et précise des informations clients.
- « Gérer devis » : Un devis signé engage les parties : le prestataire doit fournir les services et le client doit payer selon les termes du devis. Cela assure une transaction claire et transparente. Un devis signé équivaut à une commande.

- « Gérer produit » : permet de mettre à jour les informations sur les produits, pour garantir la précision des devis et offrir une vision claire aux clients lors de la commande. On peut ajouter, supprimer ou modifier les données correspondantes.
- « Accéder à l'espace statistique » : Cette fonctionnalité donne une vue d'ensemble du chiffre d'affaires et permet de suivre l'évolution de l'activité.
- « Visualiser son historique d'activité » : Cette fonctionnalité permet de voir les actions réalisées dans l'application sur une période donnée.

2.2.2.3 Diagramme de cas d'utilisation globale

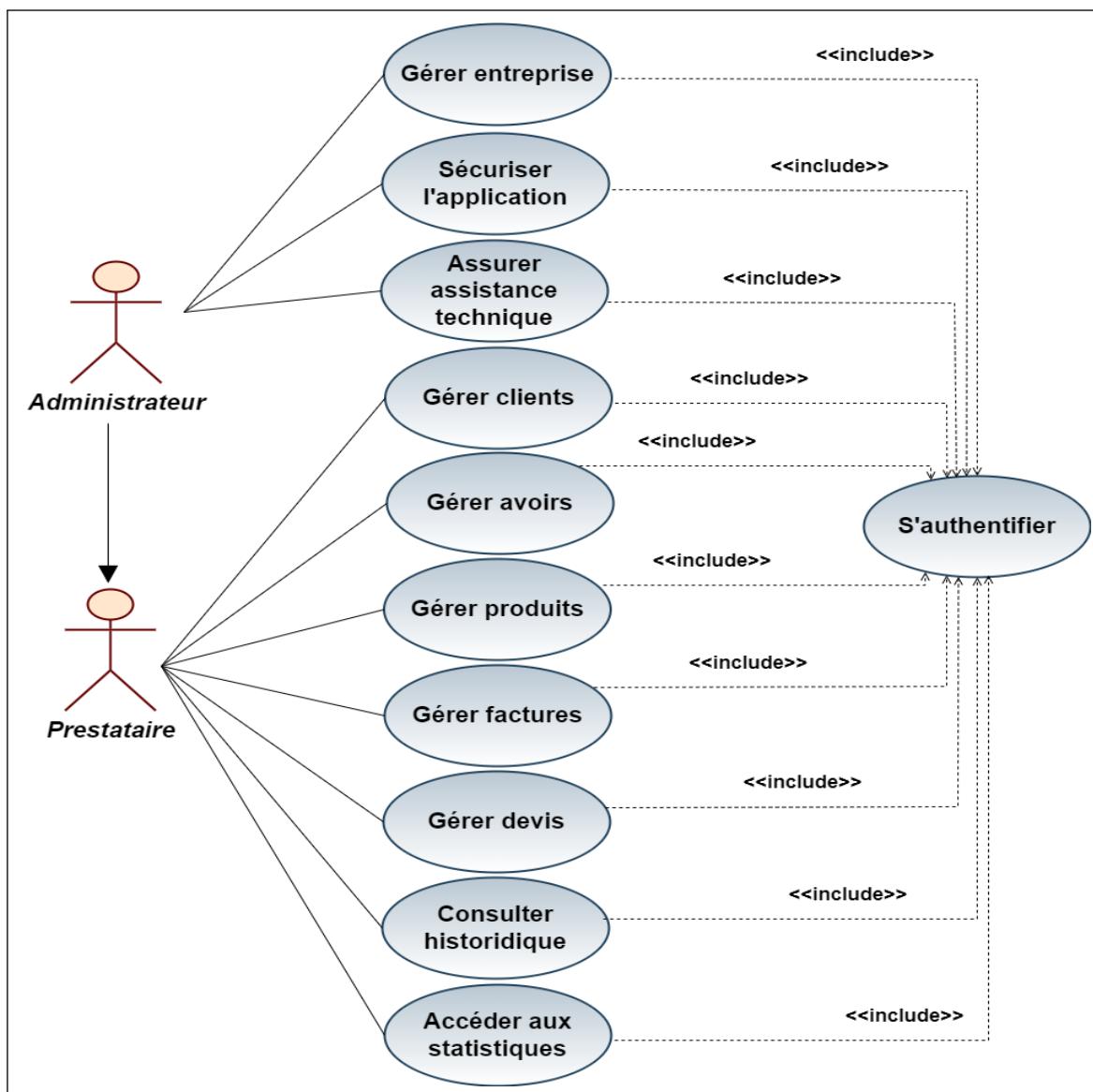


FIGURE 2.1 – Diagramme de cas d'utilisation globale

2.2.3 Besoins non fonctionnels

Quand les besoins fonctionnels expriment les fonctionnalités concrètes du système, les besoins non fonctionnels sont des indicateurs de qualité de l'exécution des besoins fonctionnels.

- La maintenabilité : conçue de manière à faciliter sa maintenance dans le temps, en utilisant des bonnes pratiques de programmation pour faciliter la compréhension et la modification du code par les développeurs.
- Performance : favorisant l'économie de l'utilisation des ressources.
- L'utilisabilité : se réfère à la facilité avec laquelle un utilisateur peut accéder et utiliser l'application sans rencontrer de problèmes. Cela inclut une interface utilisateur claire et intuitive.
- Sécurité : Pour garantir la sécurité, une application doit mettre en œuvre des mesures telles que l'authentification de l'utilisateur et la gestion des accès.

2.2.4 Besoins technique

Tandis que la spécification fonctionnelle décrit ce que le produit ou le système doit faire du point de vue de l'utilisateur final, et la spécification non fonctionnelle décrit les indicateurs de qualité, la spécification technique, quant à elle, est axée sur les détails de la mise en œuvre de ces fonctionnalités.

2.2.4.1 L'environnement logiciel

L'environnement logiciel comprend tous les outils nécessaires à la conception, au déploiement de l'application. Cela inclut les outils de développement, les outils de gestion de projet, les outils de test et les outils de déploiement.



FIGURE 2.2 – Angular 15

Angular[12] est un framework JavaScript open-source développé par Google pour la création d'applications web dynamiques et interactives. Il a une structure organisée, des outils de développement rapide, une modularité, une gestion des données efficace, des performances optimisées et une communauté active.



FIGURE 2.3 – Spring boot

Spring Boot[11] est un framework open-source pour le développement d'applications Java. Il est construit sur le framework Spring de Pivotal, avec l'objectif de simplifier et d'accélérer le processus de développement d'applications. Il utilise une approche dite "convention over configuration", qui signifie qu'il fournit des configurations par défaut pour les fonctionnalités courantes, réduisant ainsi la quantité de code boilerplate à écrire. Il est également modulaire, ce qui permet aux développeurs de choisir les composants spécifiques de Spring qu'ils souhaitent utiliser dans leur application.

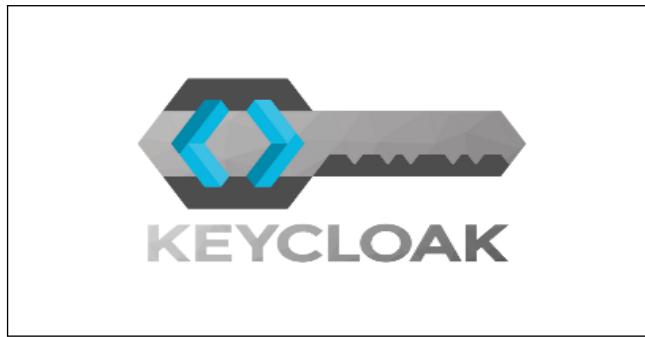


FIGURE 2.4 – Keycloak

Keycloak[8] est un logiciel open-source d’authentification et d’autorisation pour les applications web et mobiles. Il fournit des fonctionnalités telles que la gestion des identités, la sécurité et l’autorisation d’accès aux ressources.

Du côté frontend, Keycloak peut être utilisé pour authentifier les utilisateurs et autoriser l'accès aux ressources à partir d'applications web ou mobiles. Il peut également être utilisé pour implémenter des fonctionnalités de Single Sign-On (SSO) pour que les utilisateurs n'aient pas à se connecter à chaque application séparément.

Du côté backend, Keycloak peut être utilisé pour protéger les API et autres ressources en ligne, en gérant les autorisations d'accès pour les utilisateurs authentifiés. Il peut également être utilisé pour la gestion des identités et des autorisations pour les utilisateurs internes de l'application.



FIGURE 2.5 – PostgreSQL

PostgreSQL est un SGBDR open-source très puissant, fiable et extensible, offrant de nombreuses fonctionnalités avancées pour le stockage et la gestion de gros volumes de données. C'est un choix populaire pour de nombreuses applications, y compris les applications Web, les applications mobiles et les applications d'entreprise.



FIGURE 2.6 – Amazon Web Services

Amazon Web Services (AWS) est une plateforme cloud qui fournit une grande variété de services pour le stockage de données, le traitement de données, la mise en réseau, la sécurité et le déploiement d'applications.

AWS est considéré comme un outil de déploiement efficace en raison de sa large gamme de services, de sa flexibilité, de sa fiabilité, de sa sécurité et de ses outils d'automatisation. Ces avantages permettent aux développeurs de déployer rapidement des applications, de les rendre disponibles et de les maintenir en toute sécurité, tout en réduisant les coûts et les erreurs humaines.



FIGURE 2.7 – GitLab

GitLab est une plateforme de gestion de code source, de collaboration et d'outils de développement logiciel basée sur Git.

Le choix de cet outil est basé sur les critères suivantes : Documentation, Gestion de version, Intégration Continue, La possibilité d'appliquer la méthodologie Agile.

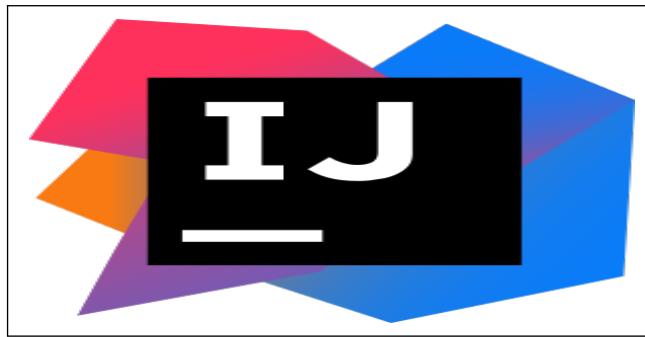


FIGURE 2.8 – IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré (IDE) pour les langages de programmation Java, Kotlin et Groovy. Il est développé par JetBrains et offre une suite complète d'outils pour le développement de logiciels Java, y compris la prise en charge de Spring Boot, de Maven et de Gradle.

Il offre des fonctionnalités avancées pour le développement d'applications Spring Boot et Angular, y compris des fonctionnalités de refactoring, une intégration avec les outils de build, la compatibilité avec les tests, des outils de débogage avancés et une intégration avec Git. Ces fonctionnalités permettent de développer plus rapidement et plus efficacement des applications de haute qualité.

2.3 Modèle Conceptuel

La mise en place d'un modèle conceptuel est une étape cruciale dans le processus de conception d'un système ou d'un produit. En d'autres termes, le modèle conceptuel est un ensemble de concepts et d'abstractions qui décrivent les différents composants, leurs interactions et les flux de données du système à développer.

Par la présente, il est important de mentionner que l'entreprise a imposé l'utilisation du modèle conceptuel Backend For Frontend (BFF), indiquant qu'elle était considérée comme l'approche la plus adaptée pour répondre aux exigences du projet

2.3.1 Modèle BFF

Le modèle BFF[?], ou Backend For Frontend, est une approche de conception logicielle qui vise à améliorer la performance et la flexibilité des applications en offrant un backend spécifique pour chaque interface utilisateur. Concrètement, cela signifie que pour chaque client (application mobile, application web, etc.), une API distincte est développée pour répondre à ses besoins spécifiques. Ainsi, chaque client dispose d'une API optimisée qui ne contient que les fonctionnalités dont il a besoin, ce qui réduit la charge du backend et améliore la réactivité de l'application. Le BFF permet également d'améliorer la sécurité en limitant l'exposition des données sensibles aux clients qui n'en ont pas besoin.

2.3.2 Fonctionnement du BFF

Le fonctionnement de ce modèle est assez simple : chaque backend est conçu pour répondre aux besoins de l'interface utilisateur à laquelle il est destiné. Cela permet de créer des fonctionnalités spécifiques pour chaque interface utilisateur, sans avoir à modifier le backend principal ou à créer des workarounds pour des besoins spécifiques.

La figure ci-dessous explique le fonctionnement du modèle en question :

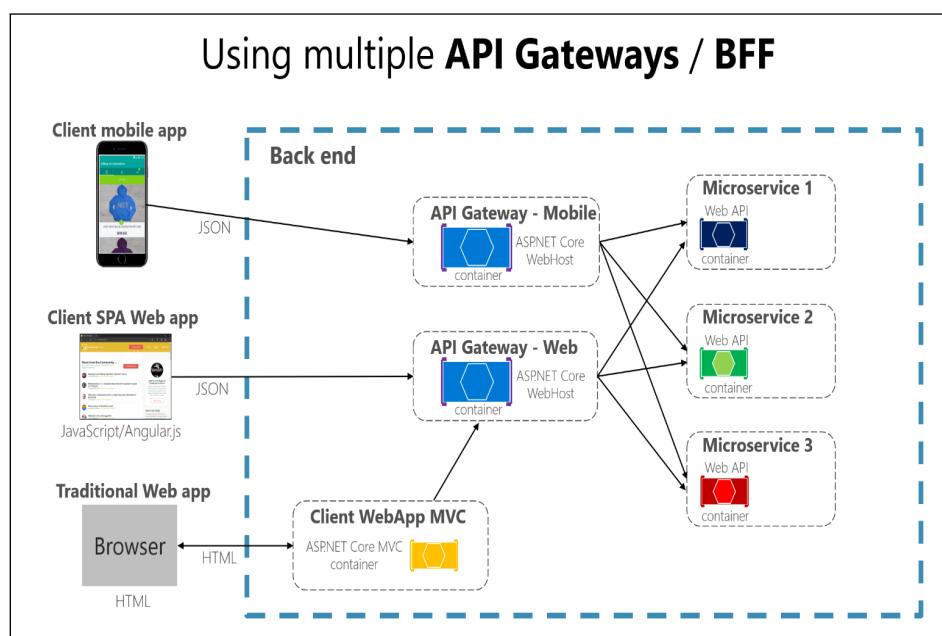


FIGURE 2.9 – BFF Design Pattern

2.4 CONCLUSION

Au cours de ce chapitre, nous avons effectué l'identification des exigences fonctionnelles et non fonctionnelles de notre application, ainsi que les rôles assignés à chaque utilisateur. Nous avons également présenté un diagramme de cas d'utilisation général dans le but de clarifier la vision du projet. Enfin, nous avons exposé les différents outils et technologies de développement et de mise en œuvre envisagés pour la réalisation du projet.

Sommaire

3.1	Introduction	33
3.2	Conception Générique	33
3.3	Conception détaillée	33
3.3.1	Langage UML	34
3.3.2	Les vues de Kruchten	35
3.3.3	Diagramme de cas d'utilisation	37
3.3.4	Diagrammes de séquence	38
3.3.5	Diagramme de classe	41
3.3.6	Diagramme de composants	42
3.4	Conclusion	43

3.1 Introduction

Après avoir examiné les exigences spécifiques de notre projet, nous avons commencé la conception en introduisant un nouveau chapitre. Cette étape comprend la création de diagrammes fonctionnels pour représenter visuellement les fonctionnalités et les interactions du projet.

3.2 Conception Générique

La conception générique est une approche de conception qui consiste à créer des modèles et des systèmes réutilisables et modulaires pour réduire les coûts et les délais de développement, améliorer la qualité du logiciel et faciliter la maintenance et l'évolution.

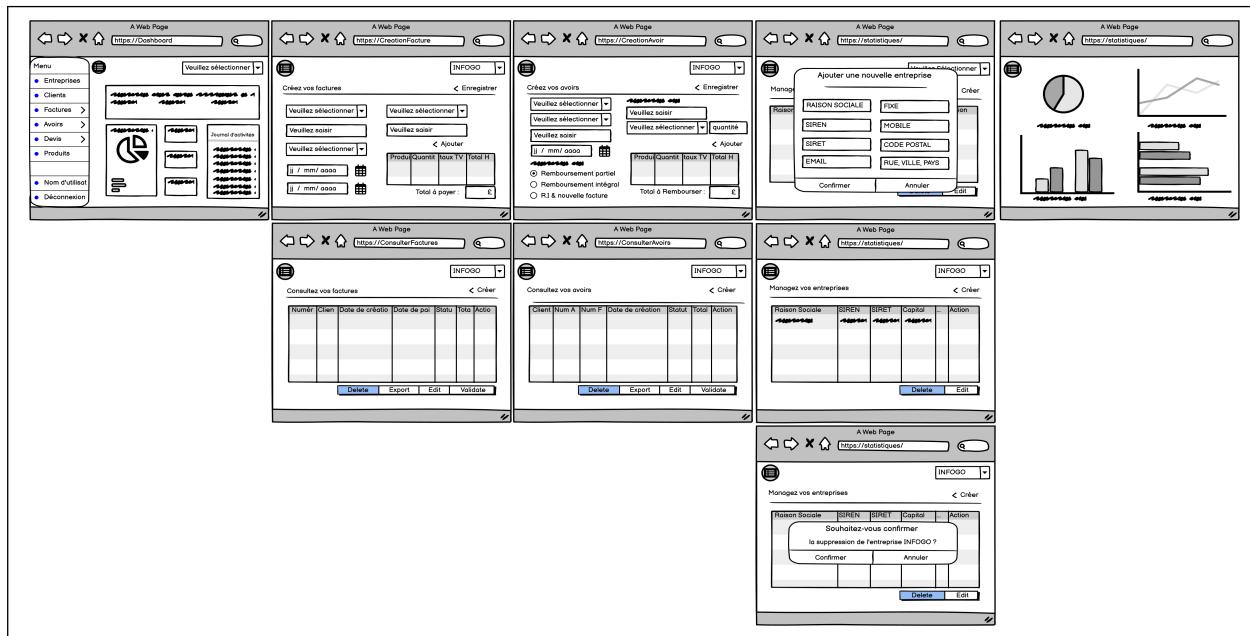


FIGURE 3.1 – Maquette de l'application

3.3 Conception détaillée

La conception détaillée est une étape de la conception de logiciels qui consiste à élaborer une représentation détaillée du système logiciel, qui prend en compte toutes les exigences et les spécifications fonctionnelles identifiées lors de la phase de conception générale. Elle

permet de traduire les concepts abstraits en une structure de code concrète. Cette phase se concentre sur la conception des différentes parties du système, comme les modules, les interfaces, les fonctions, les classes, etc. Elle implique souvent l'utilisation de diagrammes et de modèles pour décrire de manière précise le comportement et la structure du système.

3.3.1 Langage UML

L'Unified Modeling Language (UML) est un langage de modélisation graphique largement utilisé dans le domaine du génie logiciel pour représenter visuellement les systèmes logiciels. Il fournit un ensemble de notations standardisées et de conventions de représentation qui permettent aux développeurs de communiquer et de documenter efficacement les différentes phases de développement d'un logiciel.

UML offre une approche visuelle pour représenter les aspects statiques et dynamiques d'un système logiciel. Il se compose de plusieurs types de diagrammes qui servent à représenter différentes perspectives du système. Les diagrammes les plus couramment utilisés sont :

- Les diagrammes de cas d'utilisation décrivent les interactions entre les acteurs (utilisateurs) et le système, en mettant l'accent sur les fonctionnalités offertes par le système.
- Les diagrammes de classes représentent les classes du système, leurs relations et leurs attributs et opérations associés.
- Les diagrammes d'objets montrent des instances spécifiques de classes et les relations entre ces instances.
- Les diagrammes de séquence décrivent l'ordre des messages échangés entre les objets du système au fil du temps, mettant l'accent sur la séquence des interactions.
- Les diagrammes d'activités modélisent le flux de contrôle entre différentes activités ou étapes du système, montrant les décisions, les boucles et les actions parallèles.
- Les diagrammes d'état représentent les différents états d'un objet ou d'un système, ainsi que les transitions entre ces états.

- Les diagrammes de déploiement décrivent la configuration matérielle et logicielle d'un système, montrant comment les composants logiciels sont déployés sur différentes machines physiques.

En utilisant ces diagrammes, les concepteurs et les développeurs peuvent décrire les exigences du système, la structure de celui-ci, le comportement des différents éléments et leur déploiement sur des machines physiques.

3.3.2 Les vues de Kruchten

Les vues de Kruchten sont une approche de la conception logicielle qui propose de diviser le système en différentes perspectives ou vues, chacune présentant une vue particulière du système en fonction des besoins de différents acteurs ou parties prenantes. Cette approche a été proposée par Philippe Kruchten en 1995 dans le contexte du développement de grands systèmes d'information.

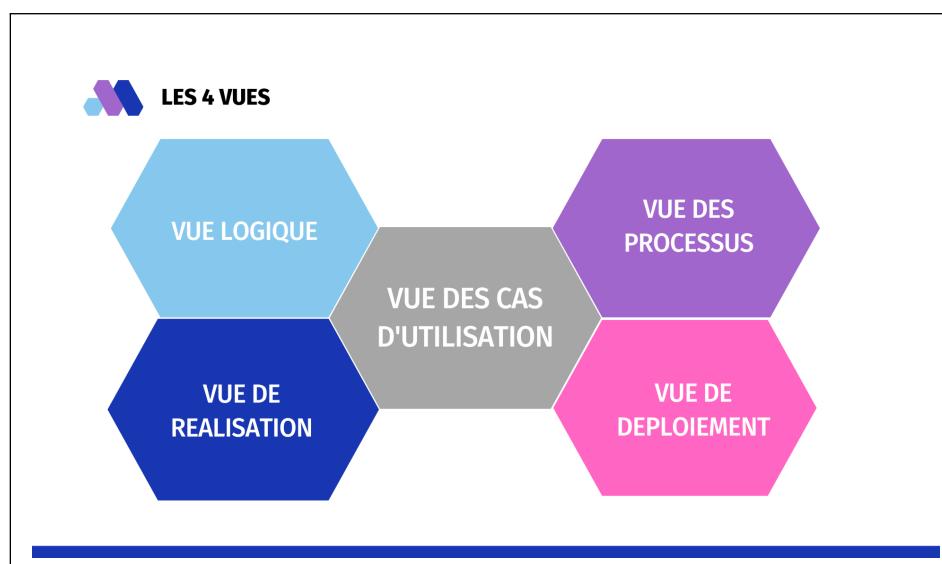


FIGURE 3.2 – Les vues de kruchten

Les 4 vues de Kruchten sont un modèle de conception qui fournit une approche systématique pour décomposer les systèmes logiciels en sous-systèmes plus petits et plus faciles à gérer :

1. La vue d'implémentation : également appelée vue de conception, présente la structure

interne du système telle qu'elle est perçue par les développeurs. Elle se concentre sur les aspects techniques du système, tels que les classes, les packages et les relations entre eux. Cette vue est utile pour assurer la cohérence de l'architecture du système et faciliter la maintenance à long terme.

2. La vue logique : également appelée vue fonctionnelle, décrit le comportement du système et montre comment les différents composants interagissent pour produire des résultats. Cette vue est utile pour comprendre les fonctionnalités du système, identifier les scénarios d'utilisation et valider les exigences fonctionnelles.

3. La vue de déploiement : également appelée vue physique, décrit la mise en œuvre physique du système et montre comment les différents composants sont déployés sur des machines et des réseaux. Cette vue est utile pour assurer la compatibilité entre les différents matériels et logiciels, et pour optimiser les performances du système.

2. La vue de composant : également appelée vue d'interface utilisateur, décrit l'interface utilisateur du système et montre comment les utilisateurs interagissent avec le système. Cette vue est utile pour valider les exigences d'utilisabilité, optimiser l'expérience utilisateur et guider la conception de l'interface utilisateur.

Dans le cadre de ce projet, les quatre diagrammes sélectionnés ont été choisis en fonction de ces quatre vues de Kruchten.

Diagramme de cas d'utilisation : Le diagramme de cas d'utilisation : représente la vue d'implémentation des fonctionnalités utilisateur.

Diagramme de séquence : représente la vue de processus et d'interaction entre les composants du système.

Diagramme de classe : représente la vue logique du système et les relations entre les classes.

Diagramme de composants : représente la vue de déploiement et la façon dont les composants du système sont distribués sur les différents matériels et réseaux.

3.3.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation détaillé montre les interactions entre les acteurs et le système, en détaillant les fonctionnalités de l'application. Il décompose chaque cas d'utilisation en sous-cas pour montrer les différentes étapes de chaque scénario, ce qui permet de comprendre en profondeur les fonctionnalités et les rôles des utilisateurs.

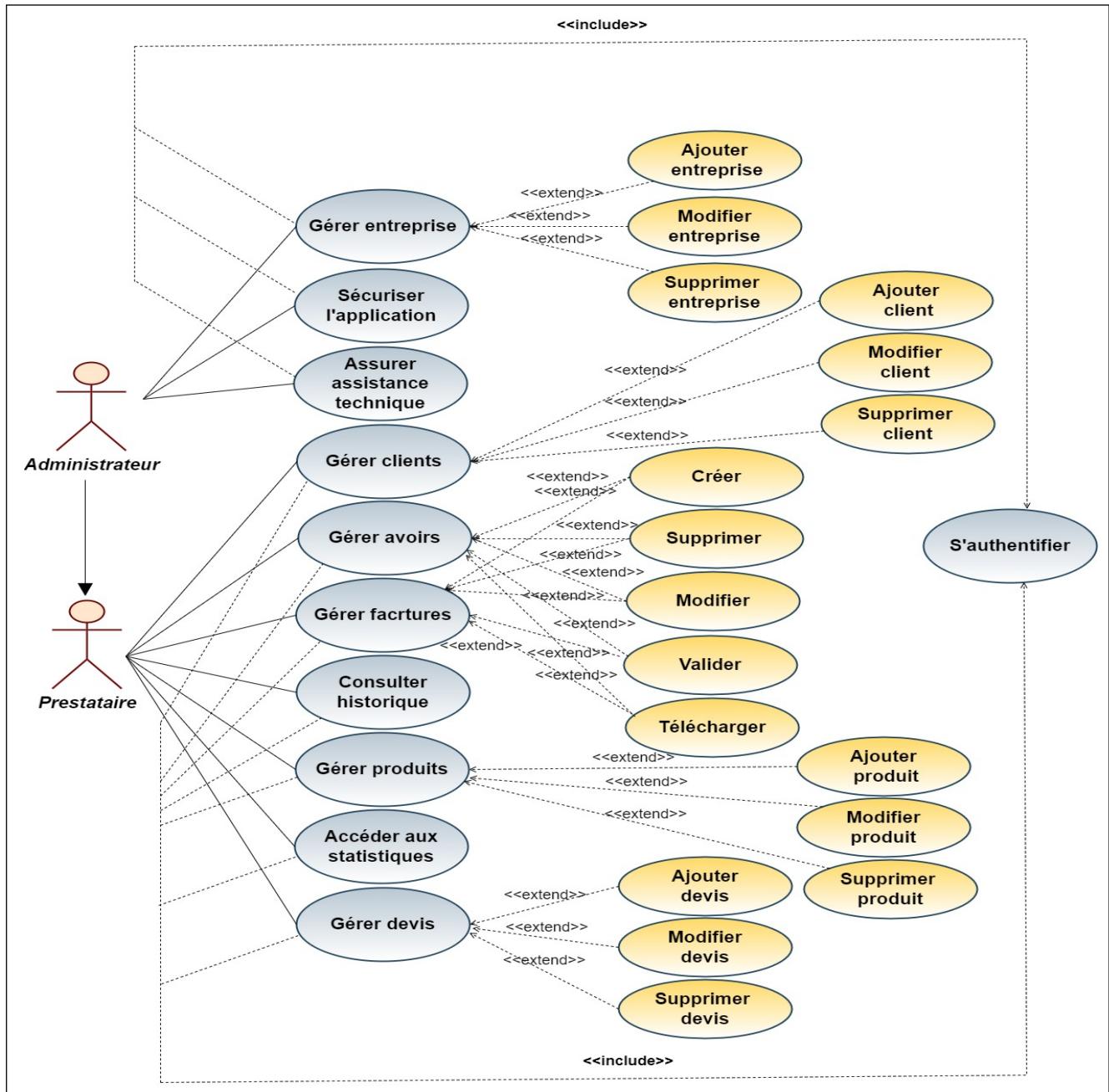


FIGURE 3.3 – Diagramme de cas d'utilisation détaillé

3.3.4 Diagrammes de séquence

Le diagramme de séquence permet de visualiser les messages échangés entre les différents objets du système, ainsi que l'ordre dans lequel ces messages sont envoyés et reçus. Il est souvent utilisé pour modéliser les scénarios d'utilisation d'un système et faciliter la compréhension des interactions complexes entre les composants d'un système.

3.3.4.1 Diagramme de séquence « S'Authentifier »

Le diagramme de séquence suivant représente le processus d'authentification d'un utilisateur sur le système à l'aide de Keycloak. Il illustre les interactions entre l'acteur utilisateur, l'interface utilisateur (Keycloak adapter), le serveur d'application et Keycloak pour permettre l'identification et l'authentification de l'utilisateur.

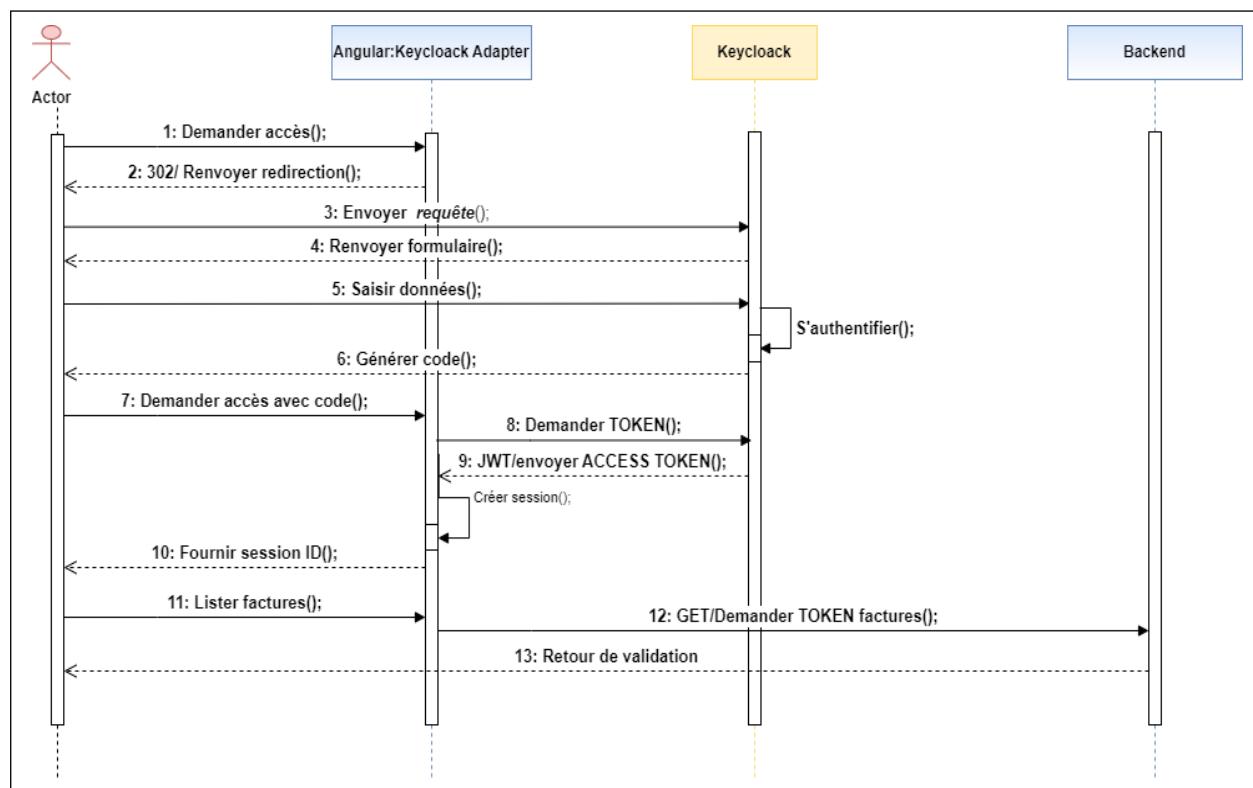


FIGURE 3.4 – Diagramme de séquence : « S'Authentifier »

Veuillez noter que les étapes 11, 12 et 13 ont été conçues pour mettre en évidence l'aspect de sécurité que Keycloak assure du côté backend.

3.3.4.2 Diagramme de séquence « Gérer client » et « Gérer produit »

Le diagramme de séquence correspondant illustrerait les échanges entre l'utilisateur, le backend et la base de données lors de l'exécution des opérations de gestion des clients. Ce même processus est également appliqué lors de la gestion des produits.

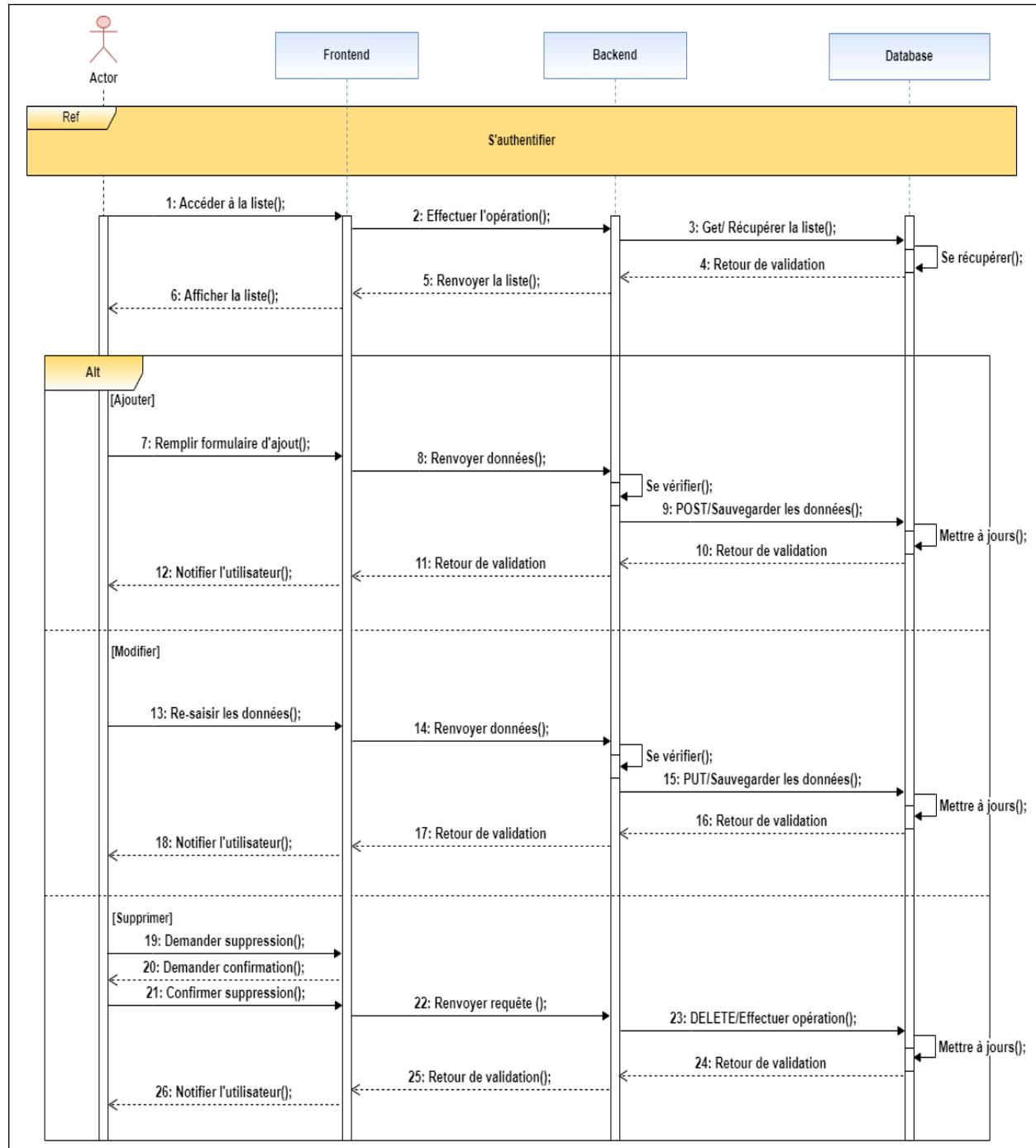


FIGURE 3.5 – Diagramme de séquence « Gérer client »,« Gérer produit »

3.3.4.3 Diagramme de séquence « Gérer Facture »

Le diagramme de séquence suivant illustre le processus de gestion des factures dans le système :

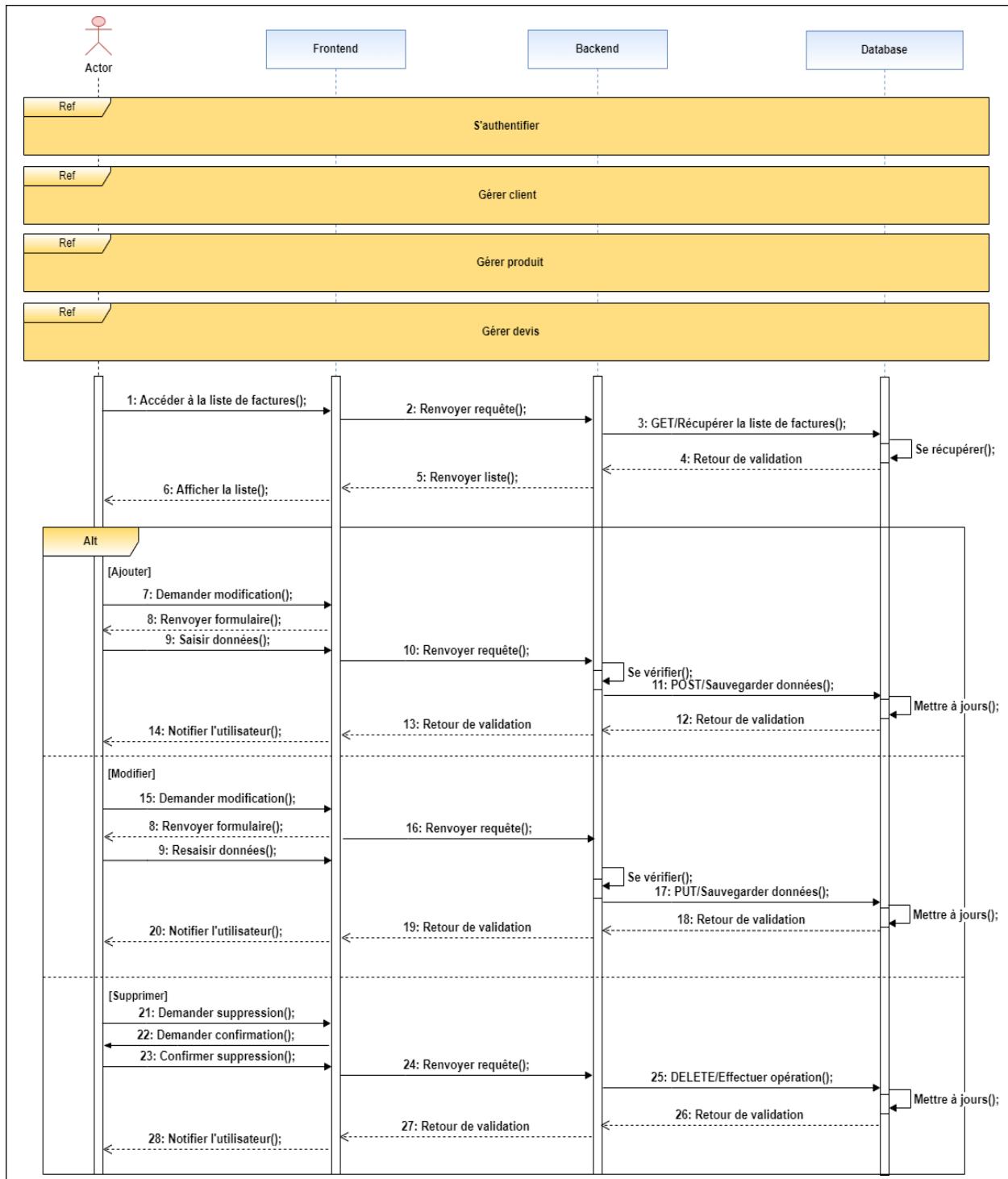


FIGURE 3.6 – Diagramme de séquence « Gérer facture »

3.3.4.4 Diagramme de séquence « Gérer entreprise »

Le diagramme de séquence suivant décrit le fonctionnement de la tâche principale d'un administrateur pour gérer une entreprise, depuis l'authentification jusqu'à la gestion des données dans la base de données, en passant par les interactions entre le frontend, le backend et l'acteur du système.

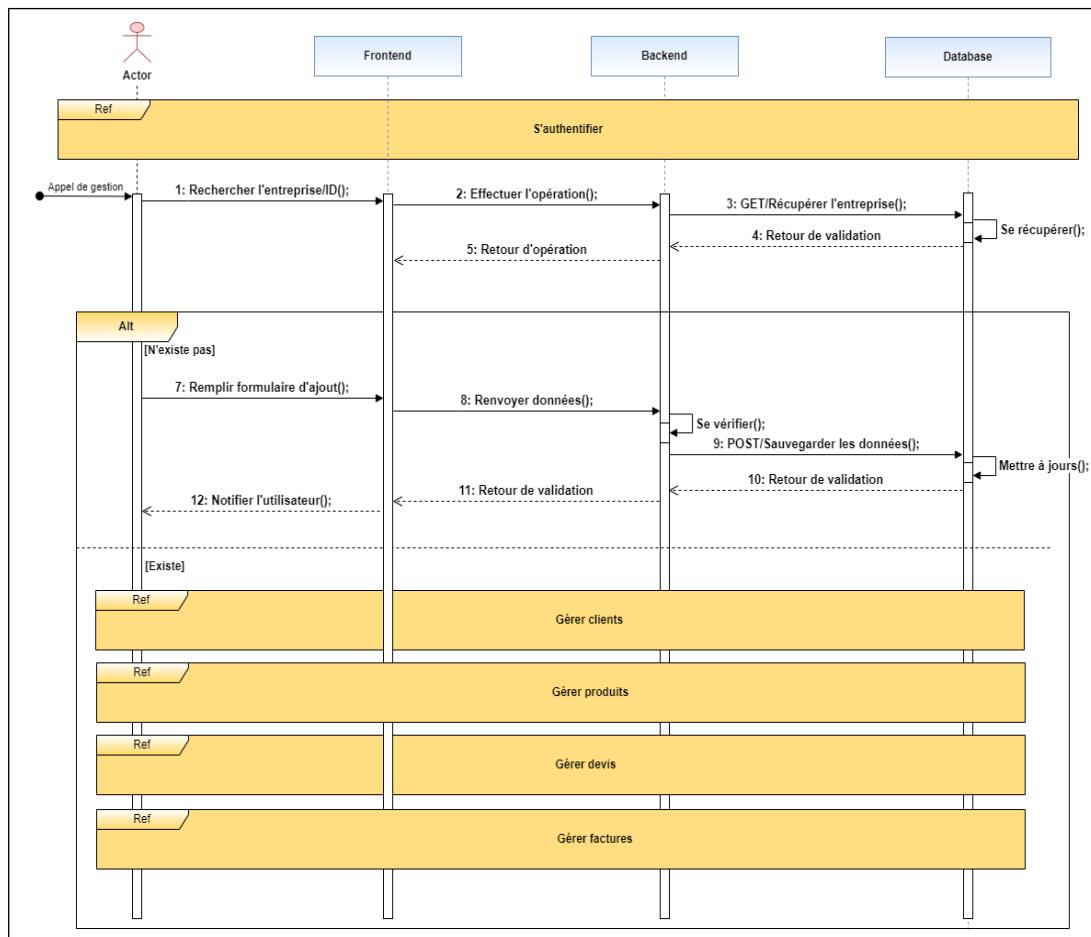


FIGURE 3.7 – Diagramme de séquence : « Gérer entreprise »

3.3.5 Diagramme de classe

Le diagramme ci-dessous est divisé en deux parties : la partie supérieure est dédiée à la présentation de la conception du backend tandis que la partie inférieure représente les classes du système avec leurs attributs, relations et méthodes, offrant ainsi une vue d'ensemble de la structure logique du projet.

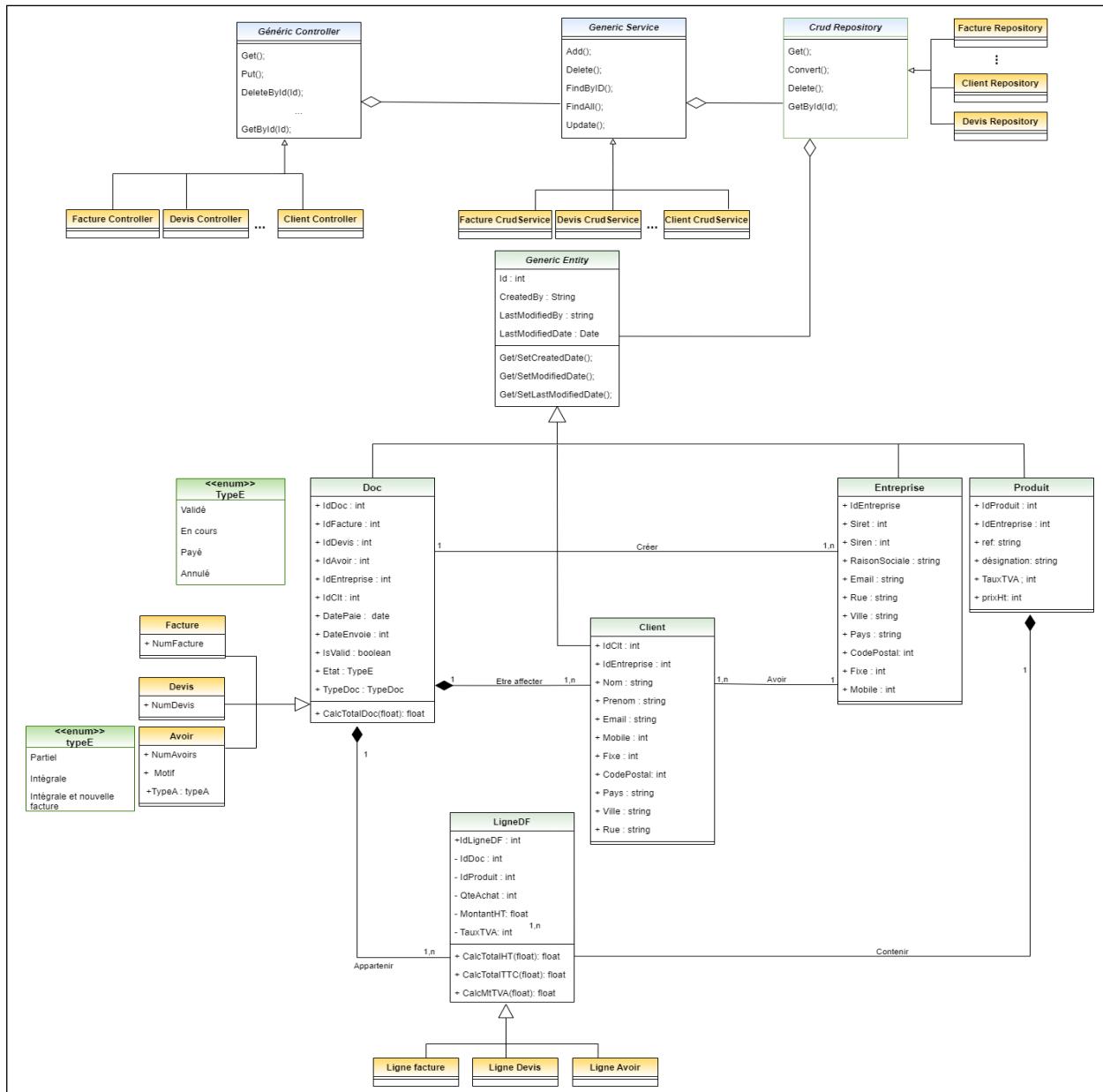


FIGURE 3.8 – Diagramme de classe

3.3.6 Diagramme de composants

La figure 1.1 présente le diagramme de composants de notre application, qui représente les différents composants principaux et leurs interactions.

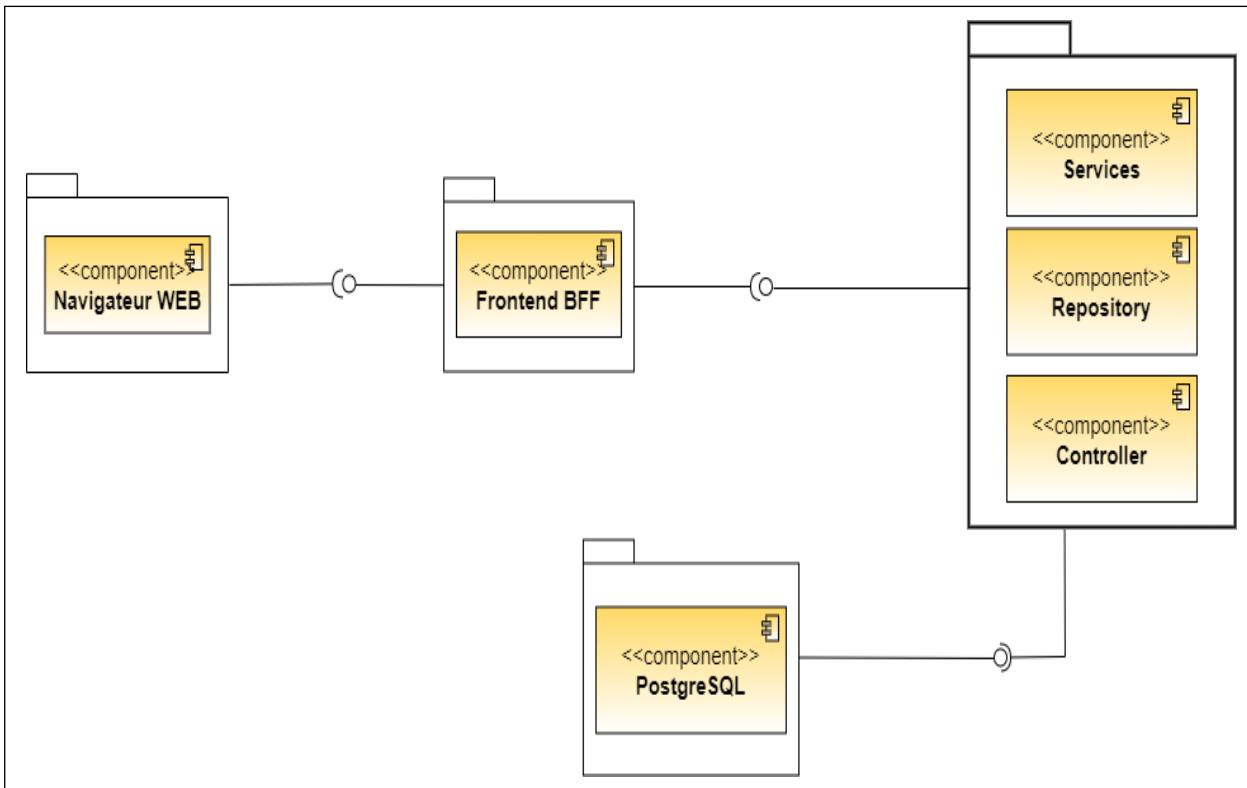


FIGURE 3.9 – Diagramme de classe

3.4 Conclusion

En conclusion de ce chapitre de conception, nous avons élaboré les diagrammes fonctionnels qui détaillent les différentes fonctionnalités et interactions de notre projet. Cette étape nous a permis d'avoir une vision claire et précise des différentes parties du projet, afin de faciliter la réalisation concrète. Le chapitre suivant portera sur la mise en œuvre de ces éléments de conception dans le cadre de la réalisation du projet.

Sommaire

4.1	Introduction	45
4.2	Présentation de la solution WEB	45
4.2.1	Interface d'authentification	45
4.2.2	Interface utilisateur du tableau de bord	46
4.2.3	Interfaces de gestion des entreprises	46
4.2.4	Interface de gestion de clients	51
4.2.5	Interface de gestion de produits	52
4.2.6	Interfaces de gestion des factures et de génération de PDF	53
4.2.7	Interface de visualisation des statistiques	56
4.3	Conclusion	56

4.1 Introduction

Dans le chapitre final de ce rapport, nous nous concentrerons sur la présentation de notre solution après avoir abordé sa conception dans le chapitre précédent. L'objectif principal est de décrire de manière détaillée les différentes fonctionnalités offertes par notre solution. Nous mettrons en évidence comment elle répond spécifiquement aux besoins identifiés, en soulignant ses caractéristiques principales et les avantages qu'elle apporte.

4.2 Présentation de la solution WEB

4.2.1 Interface d'authentification

Dans le deuxième chapitre de la spécification des besoins non fonctionnels, il est souligné que la sécurité occupe une place primordiale dans notre projet. Afin de répondre à ces exigences, nous avons opté pour l'utilisation de Keycloak. Ce choix permet d'établir un système d'authentification sécurisé et fiable. Dans la Figure 4.1, vous pouvez voir la page d'authentification fournie par Keycloak. Cette page constitue l'interface utilisateur par laquelle les utilisateurs accèdent au système et fournissent leurs informations d'identification pour s'authentifier.

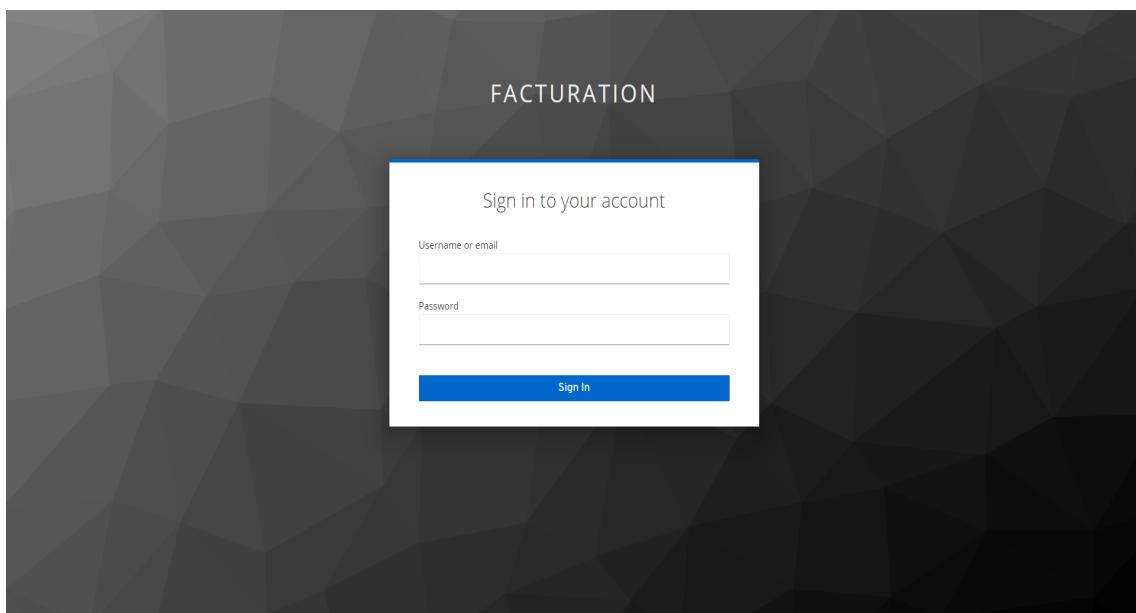


FIGURE 4.1 – Interface d'authentification fournit par keycloak

4.2.2 Interface utilisateur du tableau de bord

La figure 4.2 illustre le tableau de bord affiché immédiatement après l’authentification, qui comprend des informations générales sur les factures ainsi qu’un journal d’activité qui enregistre les actions de l’utilisateur lors de sa navigation dans l’application.

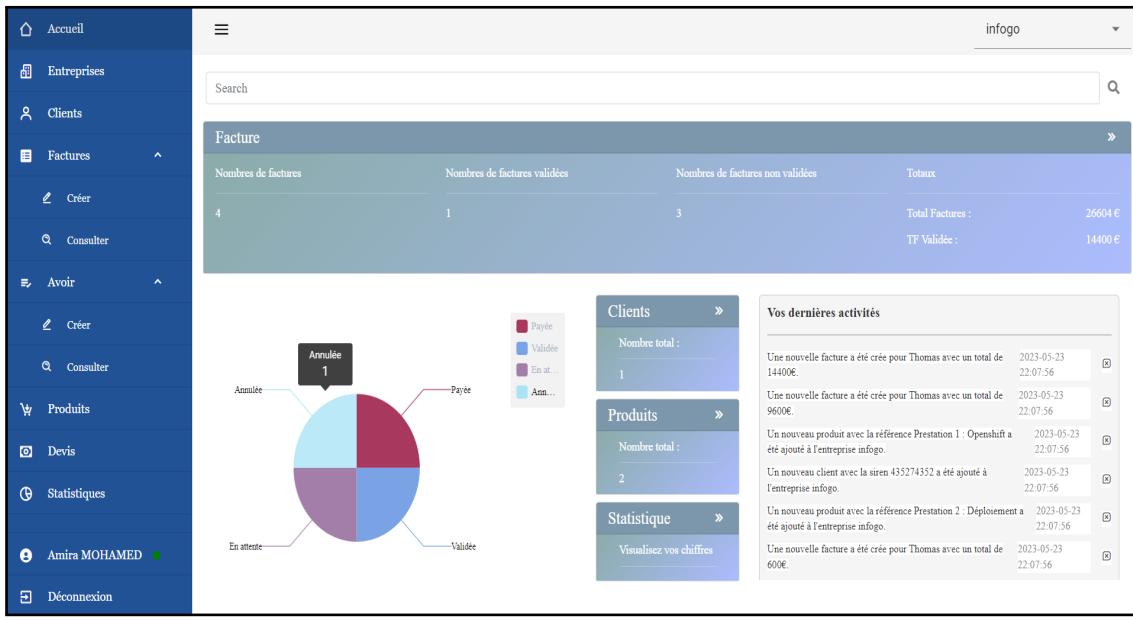


FIGURE 4.2 – Interface utilisateur du tableau de bord

4.2.3 Interfaces de gestion des entreprises

Keycloak n’est pas seulement utilisé pour l’authentification, mais également pour la gestion des rôles dans notre projet. Comme indiqué dans le diagramme de cas d’utilisation, l’administrateur est autorisé à « Gérer entreprise », ce qui lui permet ensuite d’affecter des rôles. Grâce à Keycloak, l’administrateur a pu créer des rôles pour chaque entreprise et les attribuer à leurs utilisateurs respectifs.

En d’autres termes, le prestataire n’a pas la possibilité de gérer les entreprises, cette tâche étant réservée aux autres utilisateurs disposant des rôles appropriés comme l’administrateur.

La Figure 4.3 illustre le processus de création des rôles dans Keycloak.

RÉALISATION

The screenshot shows the Keycloak interface for creating a new role. On the left, the navigation sidebar is visible with sections like 'Facturation', 'Configure', 'Realm Settings', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', and 'Authentication'. The 'Roles' section is currently selected. The main content area is titled 'Add Role' and contains fields for 'Role Name' (set to 'INFOGO') and 'Description'. A 'Save' button is at the bottom right. Below this, a table lists existing roles under the 'Realm Roles' tab. The table has columns for 'Role Name', 'Composite', and 'Description'. The 'INFOGO' role is listed as composite=False. Other roles shown include 'administrator', 'default-roles-facturation', 'offline_access', 'sacmi', and 'udev'. The 'Default Roles' tab is also present but empty.

FIGURE 4.3 – Crédation des rôles gérés par keycloak

La Figure 4.4 décrit le processus d'affectation des rôles dans Keycloak, une étape essentielle pour définir les autorisations et les accès des utilisateurs.

The screenshot shows the Keycloak interface for managing user roles. The left sidebar shows 'Facturation', 'Configure', 'Realm Settings', 'Clients', 'Client Scopes', 'Roles' (selected), 'Identity Providers', 'User Federation', and 'Authentication'. The main area shows a user named 'Amira'. The 'Role Mappings' tab is active. It displays four panels: 'Realm Roles' (listing 'administrator', 'INFOGO', and 'sacmi'), 'Available Roles' (listing 'administrator', 'INFOGO', and 'sacmi'), 'Assigned Roles' (listing 'default-roles-facturation' and 'udev'), and 'Effective Roles' (listing 'default-roles-facturation', 'offline_access', 'udev', and 'uma_authorization'). Buttons for 'Add selected' and 'Remove selected' are visible between the 'Available Roles' and 'Assigned Roles' panels. A 'Client Roles' section at the bottom allows selecting a client for the mappings.

FIGURE 4.4 – Affectation des rôles gérés par keycloak

4.2.3.1 Liste des entreprises ajoutées

L'interface ci-jointe représente la page permettant de consulter la liste des entreprises ajoutées.

RÉALISATION

Raison sociale	Siren	Siret	Capital	Code postal	Email	Fixe	Mobile	Rue	Ville	Pays	Action
Udev	879193704	87919370400019	1,000	78500	udev.info@gmail.com	0879193704	0879193704	17 RUE DE TOCQUEVIL	SARTROUVILLE	FRANCE	
INFO GO	851792747	85179274700012	1,000	78500	info.go.contact@gmail.com	0851792747	0851792747	VICTOR HUGO 512	PARIS	FRANCE	
NAKO	532612546	53261254600020	1,000	75011	contact@nakoo.fr	0532612546	0532612546	5 B IMP MONT LOUIS	Ile de France	FRANCE	
KERNIX	435274352	43527435200051	8,900	75014	contact@kernix.fr	0435274352	0435274352	6 RUE LALANDE	PARIS	FRANCE	

FIGURE 4.5 – Interface de consultation des entreprises

4.2.3.2 Le formulaire d'ajout d'une entreprise

L'interface ci-jointe présente le formulaire permettant d'ajouter une entreprise. L'utilisateur est tenu de fournir le SIRET, le SIREN, la raison sociale, le numéro de téléphone fixe, le numéro de téléphone mobile, l'adresse et l'adresse e-mail. Tous les champs doivent être validés. La validation du champ SIRET est effectuée à l'aide de l'algorithme de Luhn, tandis que les numéros de téléphone doivent respecter le format des numéros de téléphone français.

RÉALISATION

Veuillez remplir le formulaire suivant :

Raison sociale * : INFO GO
La raison sociale est valide.

Fixe * : 0851792747
Le numéro de téléphone fixe est valide.

Siren * : 851792747
Le numéro Siren est valide.

Mobile * : 0851792747
Le numéro de téléphone mobile est valide.

Siret * : 85179274700012
Le numéro Siret est valide.

Code postal * : 78500
Le code postal est valide.

Capital * : 1,000
Le capital est valide.

Rue : VICTOR HUGO 512

Adresse e-mail * : infogo.contact@gmail.com
L'adresse e-mail est valide.

Pays : FRANCE

Rue	Ville	Pays	Action
7 RUE DE TOCQUEVILLE	SARTROUVILLE	FRANCE	

FIGURE 4.6 – Processus de la création d'une entreprise

L'**algorithme de Luhn**,[9] également connu sous le nom de formule de Luhn ou de modulo 10, est une méthode de vérification utilisée pour valider les numéros de carte de crédit, les numéros de sécurité sociale, les numéros de SIRET et d'autres types de numéros d'identification. Cet algorithme a été inventé par Hans Peter Luhn, un ingénieur des systèmes de traitement de données, et il est largement utilisé dans de nombreux domaines pour détecter les erreurs de saisie ou les numéros invalides.

L'algorithme de Luhn fonctionne en suivant ces étapes :

1. À partir de droite à gauche, doublez la valeur de chaque deuxième chiffre.
2. Si le résultat de la multiplication est supérieur à 9, soustrayez 9 du résultat.
3. Additionnez tous les chiffres obtenus.
4. Si la somme totale est divisible par 10, le numéro est considéré comme valide selon l'algorithme de Luhn.

Cela permet de détecter rapidement les erreurs de saisie telles que des chiffres inversés, des chiffres manquants ou des chiffres invalides dans un numéro.

4.2.3.3 Validation des données

Veuillez noter que le formulaire est désactivé jusqu'à sa validation.

RÉALISATION

Veuillez remplir le formulaire suivant :

Raison sociale *	Fixe *	
INFO GO	1851792747	Ajouter
La raison sociale est valide.		
Siren *	Mobile *	
879193704	0851792747	7 RUE DE TOCQUEVILLE SARTROUVILLE FRANCE
Le numéro Siren est valide.		
Siret *	Code postal *	
85189274700012	7850	VILLE DE TOCQUEVILLE SARTROUVILLE FRANCE
Le numéro Siret est invalide.		
Capital *	Rue	
1,000	VICTOR HUGO 512	PARIS
Le capital est valide.		
Adresse e-mail *	Ville	
info.og@domaine.com	PARIS	FRANCE
L'adresse e-mail est valide.		
Pays		
FRANCE		

FIGURE 4.7 – Processus de la création : Validation des données

4.2.3.4

La suppression d'une entreprise La suppression de l'entreprise requiert une confirmation.

Raison sociale	Siren	Siret	Capital	Code postal	Email	Fixe	Mobile	Rue	Ville	Pays	Action
Udev	879193704	87919370400019	1,000	78500	udev.info@gmail.com	0879193704	0879193704	17 RUE DE TOCQUEVIL	SARTROUVILLE	FRANCE	
Info go	851792747				Confirmez la suppression	X		VICTOR HUGO 512	PARIS	FRANCE	
NAKO	532612546				Êtes-vous sûr(e) de vouloir supprimer ?			5 B IMP MONT LOUIS	Île de France	FRANCE	
KERNIX	435274352							5 RUE LALANDE	PARIS	FRANCE	

FIGURE 4.8 – Processus du suppression des entreprises

4.2.3.5 La gestion de services

La liste déroulante située en haut de la page est étroitement synchronisée avec la gestion des entreprises. Lorsqu'une entreprise est ajoutée, elle est automatiquement chargée dans la liste déroulante. De même, en cas de modification ou de suppression, la liste déroulante est mise à jour en conséquence.

L'objectif de cette liste déroulante est de permettre à l'utilisateur de sélectionner l'entreprise

RÉALISATION

qu'il souhaite gérer. Cela signifie que l'entreprise sélectionnée doit disposer de tous les services et fonctionnalités disponibles dans l'application.

Raison sociale	Siren	Siret	Capital	Code postal	Email	Fixe	Mobile	Rue	Ville
Udev	879193704	87919370400019	1,000	78500	udev.info@gmail.com	0879193704	0879193704	17 RUE DE TOCQUEVIL	SARTROUVILLE
INFO GO	851792747	85179274700012	1,000	78500	info.go.contact@gmail.com	0851792747	0851792747	VICTOR HUGO 512	PARIS
NAKO	532612546	53261254600020	1,000	75011	contact@nakco.fr	0532612546	0532612546	5 B IMP MONT LOUIS	Île de France
KERNIX	435274352	43527435200051	8,900	75014	contact@kernix.fr	0435274352	0435274352	6 RUE LALANDE	PARIS
									FRANCE

FIGURE 4.9 – Gestion des services

4.2.4 Interface de gestion de clients

La figure ci-dessus illustre la page de consultation de la liste des clients ajoutés. Le bouton "Ajouter" est désactivé tant qu'aucune entreprise n'est sélectionnée dans la liste déroulante. Il est nécessaire qu'un client soit associé à une entreprise.

Concernant la base de données, la table "client" doit inclure l'attribut "identreprise" en tant que clé étrangère, établissant ainsi une relation de "One-to-Many" (un à plusieurs) avec la table "entreprise". Cela permet de garantir qu'un client puisse appartenir à une seule entreprise.

RÉALISATION

The screenshot shows a web-based application interface for managing clients. On the left, a sidebar menu lists various modules: Accueil, Entreprises, Clients, Factures (with sub-options Créeer and Consulter), Avoir (with sub-options Créeer and Consulter), Produits, Devis, Statistiques, and a user profile for Amira MOHAMED. The main content area is titled "Managez vos clients" and displays a table of client data. The table has columns for Description, Siren, Nom, Prénom, Numéro TVA, Email, Fixe, Mobile, Rue, Ville, Pays, and Action. Two rows of data are visible:

Description	Siren	Nom	Prénom	Numéro TVA	Email	Fixe	Mobile	Rue	Ville	Pays	Action
INFO CONSULT	852126789	Thomas	Robert	FR85212678905	contactexclu@devpro.fr	0852126789	0120633024	LILLE	PARIS	FRANCE	
Service réseaux	852126789	Louis	Bernard	FR85212678985	codingexpert@gmail.com	0852126789	0852126789	VICTOR HUGO 512	PARIS	FRANCE	

FIGURE 4.10 – Gestion de clients

4.2.5 Interface de gestion de produits

La gestion des produits est similaire à celle des clients, comme décrit dans la section précédente. La page de consultation de la liste des produits ajoutés présente une fonctionnalité similaire à celle de la liste des clients. Le bouton "Ajouter" est désactivé tant qu'aucune entreprise n'est sélectionnée dans la liste déroulante. Chaque produit doit obligatoirement appartenir à une entreprise.

The screenshot shows a web-based application interface for managing products. The sidebar menu is identical to Figure 4.10, listing Accueil, Entreprises, Clients, Factures, Avoir, Produits, Devis, Statistiques, and the user profile for Amira MOHAMED. The main content area is titled "Managez vos produits" and displays a table of product data. The table has columns for Référence, Désignation, Prix HT, Taux TVA, and Action. Two rows of data are visible:

Référence	Désignation	Prix HT	Taux TVA	Action
Prestation 1 : Openshift	Openshift 4.1	4000	20	
Prestation 2 : Déploiement	Cloud	500	20	

FIGURE 4.11 – Gestion de produits

4.2.6 Interfaces de gestion des factures et de génération de PDF

4.2.6.1 Liste des factures créées

La figure ci-jointe représente l'interface de consultation des listes des factures créées. Chaque facture est associée à quatre types de boutons : "Modifier", "Supprimer", "Voir les détails" et un bouton de validation. Avant la validation, le bouton "Voir les détails" est désactivé. Une fois la validation effectuée, le bouton "Voir les détails" devient actif et permet de générer le fichier PDF contenant les détails de la facture. La suppression et la modification sont désactivées après la validation.

Numéro	Client	Date de création	Date de paiement	Status	Total à payer	Action
	Louis Bernard	04/04/2023	07/09/2023	En attente	€9,600.00	
	Thomas Robert	04/05/2023	07/09/2023	En attente	€1,800.00	

FIGURE 4.12 – Interface de consultation des factures

4.2.6.2 Validation des factures

Il est important de noter que le numéro de facture n'est généré qu'après la validation. Cette approche a été adoptée afin d'éviter toute lacune dans la numérotation des factures et de garantir une séquence numérique continue. En attendant la validation, le système réserve un emplacement pour le numéro de facture, mais il est attribué uniquement lorsque la facture est validée.

Cette méthode assure une gestion précise des numéros de facture, permettant ainsi de suivre facilement les transactions et d'éviter toute confusion ou duplication.

RÉALISATION

The screenshot shows a web-based application interface. On the left, there is a dark blue sidebar menu with various options: Accueil, Entreprises, Clients, Factures (with sub-options Créeer and Consulter), Avoir (with sub-options Créeer and Consulter), Produits, Devis, Statistiques, and a user profile section for Amira MOHAMED. The main content area has a light gray header with the text "Consultez vos factures" and a "Créer" button. Below this is a table with columns: Numéro, Client, Date de création, Date de paiement, Status, Total à payer, and Action. Two rows of data are visible: one for Louis Bernard (status En attente) and another for Thomas Robert (status Validée). Each row has a set of icons in the Action column.

Numéro	Client	Date de création	Date de paiement	Status	Total à payer	Action
	Louis Bernard	04/04/2023	07/09/2023	En attente	€9,600.00	
FV-001	Thomas Robert	04/05/2023	07/09/2023	Validée	€1,800.00	

FIGURE 4.13 – Validation des facture

4.2.6.3 Processus d'ajout pour une facture

Cette interface représente le formulaire d'ajout d'une facture. Il comporte trois listes déroulantes (dropdown lists). La première liste est dédiée au choix du client et ne doit afficher que les clients appartenant à l'entreprise sélectionnée en haut du formulaire. De même, la deuxième liste est dédiée au choix du produit. La troisième liste déroulante permet de sélectionner l'état de la facture parmi les options suivantes : annulée, en attente, payée ou validée.

Le tableau présent dans l'interface est utilisé pour ajouter plusieurs produits à une seule facture. Il représente la table "ligne facture" du côté de la base de données, où chaque ligne de facture est associée à un identifiant de facture et un identifiant de produit.

L'interface assure le calcul automatique des totaux. Ainsi, lorsque des produits sont ajoutés dans le tableau, les calculs nécessaires sont effectués automatiquement pour afficher le montant total de la facture.

RÉALISATION

FIGURE 4.14 – Formulaire de création pour les factures

Cette conception permet une saisie facile des informations de la facture, en liant chaque facture aux clients et produits appropriés. De plus, la possibilité d'ajouter plusieurs produits à une facture simplifie le processus de facturation et garantit un calcul précis des montants totaux.

4.2.6.4 Le PDF généré

Description	Quantité	Prix total HT	Taux TVA	Montant TVA	Prix total TTC
Prestation 1 : Openshift	4	€16,000.00	%20.00	€0.20	€19,200.00
Prestation 2 : Déploiement	5	€10,000.00	%20.00	€0.20	€12,000.00
					Total à payer : €31,200.00

FIGURE 4.15 – Le PDF généré d'une facture

4.2.7 Interface de visualisation des statistiques

L'interface située en dessous présente quatre types de diagrammes recommandés par l'entreprise d'accueil. Le premier diagramme illustre la répartition du montant total par facture selon l'état. Chaque état est accompagné de son total de factures. Le deuxième diagramme représente le chiffre d'affaires mensuel, correspondant au total des factures validées chaque mois. Le troisième diagramme présente la classification des produits en fonction de la quantité vendue. Enfin, le dernier diagramme décrit la croissance mensuelle des clients.



FIGURE 4.16 – Interface des statistiques

4.3 Conclusion

En conclusion de ce chapitre dédié à la réalisation, nous avons présenté en détail les différentes interfaces de notre solution. Chacune de ces interfaces a été conçue pour répondre à des besoins spécifiques et offrir des fonctionnalités distinctes.



CONCLUSION GÉNÉRALE

En conclusion, ce rapport a exploré en détail notre solution, en mettant l'accent sur sa conception, sa réalisation et ses fonctionnalités. Nous avons présenté un aperçu approfondi de l'environnement matériel et logiciel utilisé pour le développement, ainsi que de la phase d'implémentation avec des interfaces spécifiques. Notre solution a été conçue pour répondre aux besoins identifiés, en offrant des fonctionnalités clés qui apportent des avantages significatifs.

Grâce à ce rapport, nous espérons avoir fourni une vision claire et compréhensible de notre solution, démontrant son potentiel à résoudre les problèmes et à répondre aux exigences spécifiques. Nous sommes convaincus que notre solution apportera des améliorations notables dans le domaine concerné et ouvrira de nouvelles perspectives. Nous tenons à souligner l'importance de la collaboration et des efforts d'équipe qui ont permis de concrétiser cette solution.

Nous exprimons notre gratitude envers toutes les personnes impliquées dans ce projet, et nous sommes confiants dans le fait que notre solution contribuera de manière significative à l'avancement et à la satisfaction des besoins identifiés. Nous sommes impatients de voir notre solution en action et de recueillir les retours des utilisateurs pour continuer à l'améliorer et à l'adapter aux futurs défis.



BIBLIOGRAPHIE

- [1] **INFOGO.** Récupéré le 20 Janvier 2023, de <https://dirigeant.societe.com/dirigeant/Bilel.MOHAMED.64229092.html>
- [2] **SWIVRE.** Récupéré le 15 mars 2023, de <https://app.swiver.io/sign-in>
- [3] **Oddo.** Récupéré le 15 mars 2023, de https://www.odoo.com/fr_FR
- [4] **Evoliz.** Récupéré le 15 mars 2023, de <https://www.evoliz.com/>
- [5] **Sage One.** Récupéré le 15 mars 2023, de <https://www.sage.com/fr-fr/produits/sage-one/>
- [6] **Unified Process** Récupéré le 28 mars 2023, de <https://homepages.laas.fr/guiochet/telecharge/Guiochet-SupportCoursProcessusUnifie.pdf>
- [7] **Two Tracks Unified Process** Récupéré le 28 mars 2023, de https://fr.wikipedia.org/wiki/Two_Tracks_Unified_Process
- [8] **Keycloak.** Récupéré le 20 mars 2023, de <https://www.keycloak.org/documentation.html>, Récupéré le 20 mars 2023, de <https://www.youtube.com/watch?v=mdZauKsMDiI>
- [9] **Angular.** Récupéré le 20 mars mars 2023, de <https://app.swiver.io/sign-in>
- [10] **Spring.** Récupéré le 20 mars 2023, de <https://app.swiver.io/sign-in>
- [11] **BFF** Récupéré le 26 mars 2023, de <https://christianlydemann.com/the-complete-guide-to-backend-for-frontend-bff/>
- [12] **Formule de Luhn.** Récupéré le 01 avril 2023, de https://fr.wikipedia.org/wiki/Formule_de_Luhn

Résumé

Ce rapport présente le projet de fin d'études réalisé au sein de la société INFO GO, dans le cadre de l'obtention du Diplôme de Licence Nationale en Informatique de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM). Le projet consiste en la conception et le développement d'une application Web de facturation des prestations de service. L'application simplifie le processus de facturation en proposant des fonctionnalités complètes pour gérer les devis, les produits et les avoirs, tout en offrant des statistiques détaillées. Son objectif est de faciliter la gestion financière et de fournir une solution pratique pour les utilisateurs.

Mots clés : Angular, Spring boot, PostgreSQL.

Abstract

This report presents the final year project carried out within INFO GO company, as part of obtaining the National License Degree in Computer Science from the Higher Institute of Computer Science and Mathematics in Monastir (ISIMM). The project involves the design and development of a web-based service billing application. The application simplifies the billing process by offering comprehensive features for managing quotes, products, and credits, while providing detailed statistics. Its objective is to facilitate financial management and provide a practical solution for users.

Key-words : Angular, Spring boot, PostgreSQL.