

République Tunisienne



**Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique**

Université de Tunis El Manar

Faculté des Sciences de Tunis



Département des Sciences de l'informatique

RAPPORT DE PROJET DE FIN D ÉTUDES

**Présenté en vue de l'obtention du
Diplôme Licence en Sciences de l'Informatique**

Par **Chaima Chaari**

Intitulé

**Application Web de gestion du contrôle du processus de
simulation de facture**

Organisme d'accueil : **Orange Tunisie**

Encadré par : **Mme.Ahlem Ben Cherifa.**

Mr.Mohamed Ksentini.

Année Universitaire :2021-2022

DÉDICACE

Ma mère, qui a œuvré pour ma réussite, par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Ma grande mère qui m'a accompagné par ses prières, sa douceur. Puisse Dieu lui prêter longue vie et beaucoup de santé et de bonheur dans sa vie

Mes chères sœurs Nourhéne et Lina pour leurs encouragements permanents, et leur soutien moral.

Toute ma famille pour leur soutien tout au long de mon parcours universitaire.

À mes très chers amis, en souvenir de nos éclats de rire et des bons moments, en souvenir de tout ce qu'on a vécu ensemble. J'espère de tout mon cœur que notre amitié durera éternellement.

À tous ceux que j'aime et à tous ceux qui m'aiment, je dédie ce modeste travail avec tous mes souhaits de bonheur.

REMERCIEMENTS

Ces quelques lignes sont réservées en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à la concrétisation de ce travail.

Je tiens à remercier mon encadrant professionnel **Mr. Mohamed Ksentini** pour son assistance, ses précieuses directives, ses fructueuses explications, ses éclaircissements, sa disponibilité et son encouragement perpétuel pour mener à bien ce travail.

Je tiens à présenter mes reconnaissances et mes remerciements à mon encadrante **Mme. Ahlem Ben Cherifa** pour sa supervision, sa contribution à la rédaction de ce rapport et ses encouragements dans le cadre de ce projet.

Mes vifs remerciements vont également aux membres du jury qui nous ont honoré en acceptant de juger ce modeste travail.

Que tout le corps professoral et administratif de la Faculté des Sciences de Tunis trouve ici le témoignage de ma profonde reconnaissance pour leur contribution à cette superbe formation.

TABLE DES MATIÈRES

Introduction générale	1
1 CONTEXTE GÉNÉRAL	3
1.1 Cadre général du projet	4
1.2 Organisme d'accueil : Orange Tunisie	4
1.2.1 Présentation d'Orange Tunisie	4
1.2.2 Activités d'Orange Tunisie	5
1.2.3 Organigramme de la société	6
1.3 État de l'art	6
1.4 Étude et critique de l'existant	7
1.4.1 Étude de l'existant	7
1.4.2 Critique de l'existant	7
1.5 Solution proposée	8
1.6 Méthodologie	8
1.6.1 Etude comparative	8
1.6.2 Choix de la méthode	10
Conclusion	12
2 Analyse et spécification des besoins	13
2.1 Identification des acteurs	14
2.2 Identification des besoins	14

2.2.1	Besoins fonctionnels	14
2.2.2	Besoins non fonctionnels	15
2.3	Modélisation des besoins	16
2.3.1	Langage de modélisation	16
2.3.2	Identification des cas d'utilisation globales	16
	Conclusion	17
3	Conception	18
3.1	Conception architecturale	19
3.1.1	Architecture opérationnelle	19
3.1.2	Architecture technique	20
3.1.3	Architecture logicielle de l'application : MVC	21
3.2	Conception détaillée	21
3.2.1	Description de la vue statique	21
3.2.1.1	Raffinement des cas d'utilisation de l'application web . . .	21
3.2.1.1.1	Cas d'utilisation s'authentifier	22
3.2.1.1.2	Cas d'utilisation : Gérer utilisateurs :	23
3.2.1.1.3	Cas d'utilisation : Affecter facture :	24
3.2.1.1.4	Cas d'utilisation : Vérifier factures :	25
3.2.1.1.5	Cas d'utilisation : Gérer suivi factures :	27
3.2.1.1.6	Cas d'utilisation : Editer tableau de board :	28
3.2.1.2	Diagramme de classe	29
3.2.2	Description de la vue dynamique	30
3.2.2.1	Diagrammes des séquences	30
3.2.2.1.1	Diagramme de séquence « S'authentifier »	31
3.2.2.1.2	Diagramme de séquence « Modifier utilisateur »	32
3.2.2.1.3	Diagramme de séquence « Affecter facture »	33
3.2.2.1.4	Diagramme de séquence « Vérifier facture »	33
3.2.2.2	Diagramme d'activité	35
4	Réalisation Et Test	37
4.1	Environnement matériel	38
4.2	Environnement logiciel	38
4.3	Technologies utilisées Application web	39

4.4	Présentation de l'application réalisée	41
4.4.1	Inscription	41
4.4.2	Authentification	41
4.4.3	Dashboard	42
4.4.4	Gestion utilisateurs	43
4.4.5	Affecter facture	44
4.4.6	Vérifier facture	44
Conclusion générale		47

TABLE DES FIGURES

1.1	Logo Orange Tunisie	5
1.2	Organigramme Orange Tunisie	6
1.3	Méthode de développement RUP	11
2.1	Diagramme de cas d'utilisation global	17
3.1	Architecture opérationnelle de l'application	19
3.2	Diagramme de déploiement	20
3.3	Architecture technique de l'application	20
3.4	Le modèle de conception MVC	21
3.5	Raffinement du cas d'utilisation « s'authentifier »	22
3.6	Raffinement du cas d'utilisation « Gérer utilisateurs »	23
3.7	Raffinement du cas d'utilisation « Affecter facture »	24
3.8	Raffinement du cas d'utilisation « Vérifier factures »	25
3.9	Raffinement du cas d'utilisation « Gérer suivi factures »	27
3.10	Raffinement du cas d'utilisation « Editer tableau de board »	28
3.11	Diagramme de classe	29
3.12	Diagramme de séquence « S'authentifier »	31
3.13	Diagramme de séquence « Modifier utilisateur »	32
3.14	Diagramme de séquence « Affecter facture »	33
3.15	Diagramme de séquence « Vérifier facture »	34
3.16	Diagramme d'activité »	35
4.1	Interface d'inscription	41

4.2	Interface d'authentification	42
4.3	Dashboard	42
4.4	Interface de la liste des utilisateurs	43
4.5	Dialogue de modification d'utilisateur	43
4.6	Interface d'affectation	44
4.7	Interface de contrôle	45
4.8	Interface de contrôle	45
4.9	Interface de contrôle	46

LISTE DES TABLEAUX

1.1	Étude comparative des méthodologies	9
3.1	Description du Cas d'utilisation : s'authentifier	22
3.2	Description du Cas d'utilisation : Gérer utilisateur	24
3.3	Description du Cas d'utilisation : Affecter facture	25
3.4	Description du Cas d'utilisation : Vérifier factures	26
3.5	Description du Cas d'utilisation : Gérer suivi factures	27
3.6	Description du Cas d'utilisation : Consulter statistiques	28
3.7	Description textuelle du diagramme de classes	30
4.1	Environnement matériel	38

INTRODUCTION GÉNÉRALE

LA génération des factures est une étape essentielle dans le cycle de gestion des services d'Orange Tunisie mais qui peut générer des erreurs, par conséquent des Avoirs et réclamations seront engendrés, ce qui conduit à l'insatisfaction des clients ou à des déficits financiers pour l'entreprise. Afin d'éviter ces problèmes, Orange a mis en place un processus de contrôle de factures simulées afin de corriger ces anomalies avant la facturation réelle.

Et comme Orange Tunisie aspire à optimiser et fiabiliser la gestion de ses structures internes et pour éviter les complications de communication entre les différents intervenants, cette entreprise souhaite mettre en place une application pour gérer son processus de contrôle. En effet, notre travail consiste à réaliser une application web permet de gérer et de contrôler les factures simulées dont les administrateurs peuvent superviser le travail de ses contrôleurs.

Ce rapport présentera les différentes étapes de la réalisation de ce projet et s'étalera sur quatre chapitres comme suit :

- Le premier chapitre "CONTEXTE GÉNÉRAL" est un chapitre introductif dans lequel nous faisons une brève description de l'entreprise. Ensuite, nous exposons le cadre général du projet et de la solution proposée et une brève description de la méthodologie adoptée.

- Le deuxième chapitre "Analyse et spécification des besoins", sera consacré à l'analyse et à la spécification des besoins fonctionnels et non fonctionnels. Nous commencerons par

l'identification des acteurs suivie de la clarification des besoins fonctionnels et non fonctionnels. Ensuite, nous illustrerons le diagramme global des cas d'utilisation.

- Le troisième chapitre "CONCEPTION" présentera la conception architecturale et la conception détaillée contenant la description textuelle des cas d'utilisation de notre application, le diagramme de classe global et sa description, les diagrammes de séquence et le diagramme d'activité.

- Le quatrième chapitre "Réalisation Et Test" se concentrera sur l'architecture matérielle et logicielle de l'application ainsi que sur l'environnement de développement et la réalisation de l'application présentée par des captures d'écran de l'application.

CHAPITRE 1

CONTEXTE GÉNÉRAL

1.1	Cadre général du projet	4
1.2	Organisme d'accueil : Orange Tunisie	4
1.2.1	Présentation d'Orange Tunisie	4
1.2.2	Activités d'Orange Tunisie	5
1.2.3	Organigramme de la société	6
1.3	État de l'art	6
1.4	Étude et critique de l'existant	7
1.4.1	Étude de l'existant	7
1.4.2	Critique de l'existant	7
1.5	Solution proposée	8
1.6	Méthodologie	8
1.6.1	Etude comparative	8
1.6.2	Choix de la méthode	10
	Conclusion	12

Introduction

Dans ce chapitre nous présenterons le contexte et les circonstances dans lesquelles notre projet s'est réalisé. Nous placerons, tout le long de ce chapitre, le projet dans son cadre général en se concentrant sur la présentation de l'organisme d'accueil, de la problématique et enfin la description du projet ainsi que ses objectifs.

1.1 Cadre général du projet

Ce projet de fin d'études vise à compléter la formation universitaire et à obtenir le diplôme de licence en sciences de l'informatique, à la suite d'un parcours d'études étalé sur trois ans à la Faculté des Sciences de Tunis (FST). Ce projet consiste également à renforcer les connaissances théoriques et à les rendre plus utiles dans un environnement professionnel. Il est également destiné à tester notre esprit de développement.

Afin d'atteindre ces objectifs, nous avons choisi d'effectuer un stage chez Orange Tunisie. Plus précisément, au sein de la DSI : direction du système d'information au département RUN.

1.2 Organisme d'accueil : Orange Tunisie

1.2.1 Présentation d'Orange Tunisie

Orange Tunisie est une filiale du Groupe Orange, qui est une société française des télécommunications qui compte à la fin de 2019 plus de 200 millions de clients dans le monde. Le Groupe Orange se présente comme le leader ou le second opérateur dans plus de 70% des pays européens et dans plus de 80% des pays en Afrique et au Moyen-Orient. Orange Tunisie a été lancée le 5 mai 2010 résultant de la fusion entre Orange SA (49%), présidée par Didier Lombard et la société Investec (51%), présidée par Marwan Mabrouk.

Orange Tunisie est le deuxième opérateur privé des télécommunications en Tunisie. Notons également qu'Orange Tunisie se présente comme étant le second opérateur privé qui a bénéficié d'une licence de téléphonie mobile en Tunisie. Aussi, cet opérateur est le premier en termes de licence pour exploiter un réseau 3G et de téléphonie fixe.



FIGURE 1.1 – Logo Orange Tunisie

1.2.2 Activités d'Orange Tunisie

Orange Tunisie est présente dans de nombreux secteurs d'activités dont nous citons les suivants :

- Les services des communications résidentiels, à savoir la téléphonie fixe, l'internet (bas, haut et très haut débit), la téléphonie IP, la visiophonie, la télévision numérique avec la TV Orange et les contenus multimédia, comme la vidéo à la demande.
- Les services des communications personnels, à savoir la téléphonie mobile.
- Les services des communications d'entreprise, à travers sa marque Orange Business Services.
- Les services bancaires via Orange Money.

1.2.3 Organigramme de la société

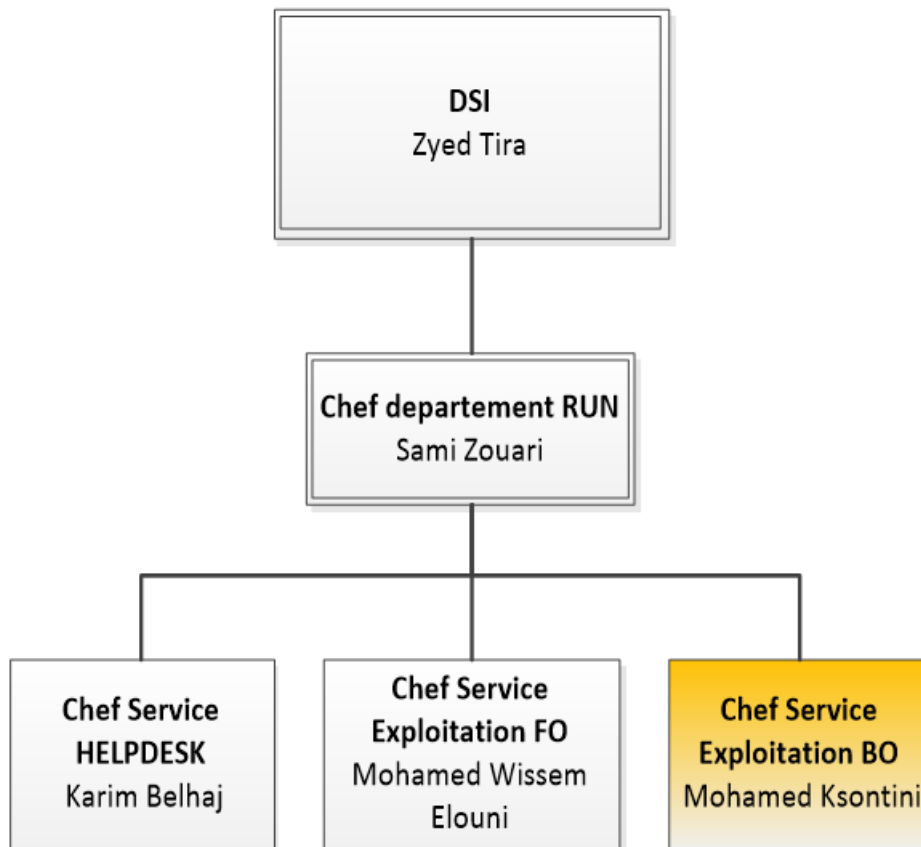


FIGURE 1.2 – Organigramme Orange Tunisie

1.3 État de l'art

Les méthodes adoptées pour la génération des factures présentent un certain nombre de problèmes et de lacunes, tels que des erreurs dans le montant de la TVA, du droit de timbre... qui peuvent être en excès ou en défaut, entraînant soit la déception des clients, soit des déficits financiers pour l'entreprise.

Pour résoudre ces anomalies, Orange Tunisie utilise un processus de simulation de factures qui permet à l'entreprise de réduire le taux d'erreur et de la rendre plus efficace et compétitive sur le marché. Comme Orange est consciente de l'impact de l'automatisation des ressources internes sur l'amélioration de ses performances ils nous ont demandé de développer une application web pour suivre ce processus car sur le marché il n'y a pas d'application qui répond à leurs besoins et notre application sera la première à le réaliser. C'est pour cette raison que nous nous sommes tournés vers l'étude de l'existant.

1.4 Étude et critique de l'existant

1.4.1 Étude de l'existant

Pour connaître et déterminer nos objectifs, l'étude de l'existant est une phase très importante. C'est une étude suivie de la critique du système actuel pour présenter l'amélioration de la solution proposée. Pour gérer le processus du contrôle de simulation de factures, Orange conserve le mode de fonctionnement euclidien qui est basé sur le fait que chaque responsable doit envoyer par e-mail des factures simulées à chaque conseiller client "contrôleur" pour les vérifier et envoyer leur diagnostic.

1.4.2 Critique de l'existant

Après une étude bien détaillée, Les procédures actuellement adoptées pour la gestion des factures simulées présentent les déficiences suivantes :

- Manque d'organisation :
 - L'utilisation des feuilles de calcul Excel présente un certain nombre de problèmes.
 - Difficulté d'échanger des données entre les différents acteurs.
 - Risque de ne pas recevoir les factures ou les résultats du contrôle.
 - Des factures peuvent ne pas être affectées.
- Problème de contrôle et de suivi :
 - les conseillers client peuvent ne pas vérifier Les factures mais les valider.
 - Les responsables ne peuvent pas bien suivre le processus.
 - La difficulté d'analyser et traiter les documents pour faire les statistiques ou pour d'éventuelles prises de décisions.
- Manque de traçabilité :
 - Pas de visibilité de l'action tout au long du processus.
- Risque d'erreur élevé :
 - La saisie manuelle des informations augmente le risque d'erreur.

1.5 Solution proposée

En tenant compte des critiques et des besoins d'informatiser la facturation des services rendus par l'entreprise, la solution est de concevoir et développer une application permettant de satisfaire au maximum possible le gestionnaire.

Cette application devrait obéir les besoins suivants :

Informatiser les tâches de travail :

- Faciliter l'affectation des factures aux conseillers client.
- Faciliter la répartition des tâches.
- Faciliter la vérification des factures.
- Faciliter l'accès à l'information en améliorant les recherches.
- Assurer la visibilité des opérations effectuées en conservant leur historique et en produisant les statistiques nécessaires.
- Créer des interfaces claires et ergonomiques.
- Conserver les données en toute sécurité.
- Réduire le temps de travail.
- Limiter les pertes dues aux retards ou aux contraintes.

1.6 Méthodologie

1.6.1 Etude comparative

Analysant les différentes méthodologies rencontrées, on distingue deux principaux types des approches :

- Approche traditionnelle
- Approche Agile

Et pour mieux faire, nous avons effectué une étude comparative :[1]

Theme	Approach traditionnelle	Approach agile
Cycle de vie	En cascade ou en V, sans Retroaction possible, phases séquentielles	Iterative et incremental.
Équipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée ou l'initiative et la communication sont privilégiées, soutenue par le chef de project
Suivi de l'avancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Gestion des risques	Processus distinct, rigoureux de gestion de risques	Gestion des risques intégrée dans le processus global, avec responsabilisation de chacun dans l'identification et la résolution des risques.
Mesure de succès	Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité.	Satisfaction client par la livraison de valeur ajoutée.

TABLE 1.1 – Étude comparative des méthodologies

Selon cette étude, nous avons choisi de travailler avec Agile, qui propose de réduire considérablement cet effet tunnel en donnant plus de visibilité, en impliquant le client du début à la fin du projet et en adoptant un processus itératif et incrémental. Il considère que le besoin ne peut être imposé et propose au contraire de s'adapter à ses évolutions. Mais pas sans un minimum de règles.

1.6.2 Choix de la méthode

L'approche AGILE comporte plusieurs méthodes dont on peut citer RUP et SCRUM. Comme dès le départ, nous avons discuté avec l'entreprise le projet et tous ses détails, et pendant la période de travail nous ne nous sommes pas beaucoup rencontrés. Et vu que la méthode RUP aborde l'ensemble du projet dès le début contrairement à Scrum qui se concentre principalement sur les tâches quotidiennes, nous avons vu que la méthode la plus appropriée pour notre projet est la méthode RUP.

RUP : (Rational Unified Process) [2] est une approche itérative de gestion de projet qui s'intéresse au cycle de vie d'un logiciel et le gère. En ce qui concerne les logiciels orientés objet, elle se concentrera sur son développement. La méthode RUP (Rational Unified Process), quant à elle, est une des émanations de la méthode PU, qui s'attache à donner un cadre précis au développement du logiciel.

C'est une méthode générique, itérative et incrémentale assez lourde mais qui s'adapte très facilement aux processus et aux besoins du développement et qui permet de s'assurer de toujours répondre aux besoins des clients.

Les plus de la méthode RUP, ce sont ses revues fréquentes avec toutes les parties prenantes : le produit est ainsi construit en sécurité. Chaque itération est considérée comme un livrable et l'on peut revenir sur les phases précédentes et réadapter le logiciel. Le retour utilisateur prend une grande place dans la construction.

En effet, cette méthode propose de se concentrer sur quatre phases d'évolution. Ils sont découpés en cycles :

- La création :
 - Évaluation des risques
 - Architecture
 - Planification
- L'élaboration :
 - Spécification du besoin
 - Validation
 - Environnement projet
- La construction :
 - Production du logiciel ,
 - La documentation support
 - Tests
- La transition :
 - Test système et utilisateur
 - Correction
 - Déploiement

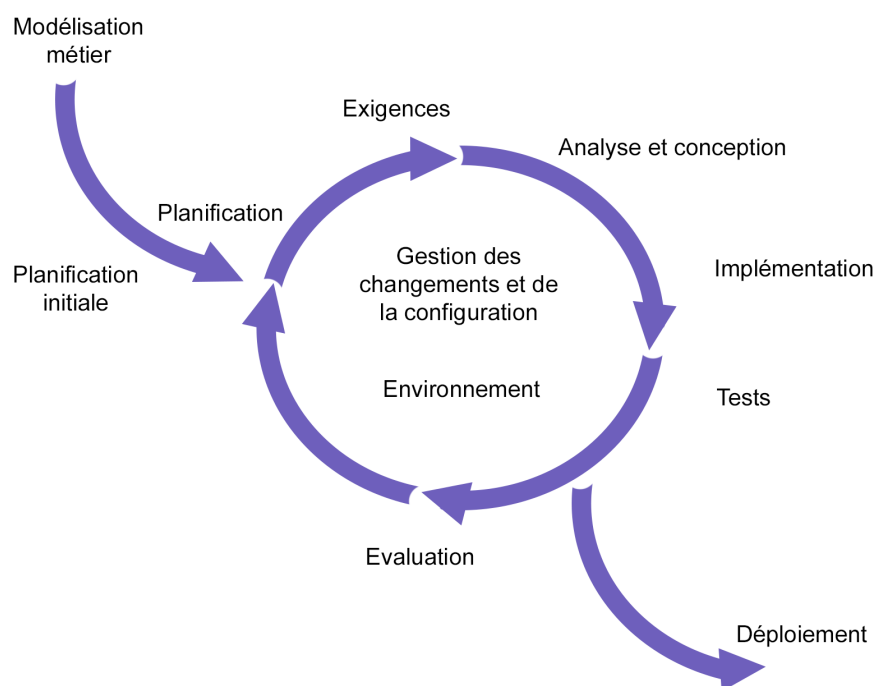


FIGURE 1.3 – Méthode de développement RUP

Conclusion

Ce premier chapitre a été consacré à la présentation de l'organisation hôte et à la mise en place du projet dans son cadre général en introduisant la problématique et les objectifs du projet. Nous avons également annoncé la démarche qui sera suivie tout au long de ce projet. Dans le chapitre suivant, nous verrons les différents besoins fonctionnels et non fonctionnels et les différents cas d'utilisation.

CHAPITRE 2

ANALYSE ET SPÉCIFICATION DES BESOINS

2.1	Identification des acteurs	14
2.2	Identification des besoins	14
2.2.1	Besoins fonctionnels	14
2.2.2	Besoins non fonctionnels	15
2.3	Modélisation des besoins	16
2.3.1	Langage de modélisation	16
2.3.2	Identification des cas d'utilisation globales	16
Conclusion		17

Introduction

Dans ce chapitre, nous allons tout d'abord identifier les acteurs, analyser les besoins fonctionnels et non fonctionnels, puis nous terminerons ce chapitre en modélisant les besoins sous la forme d'un diagramme général de cas d'utilisation.

2.1 Identification des acteurs

Un acteur représente un rôle joué par une personne qui interagit avec le système. Par définition, les acteurs sont externes du système. Les acteurs que nous avons identifiés sont :

Administrateur : il s'agit de la personne principale de cette application qui gère les utilisateurs et supervise les opérations. Il peut aussi exécuter toutes les fonctionnalités assurées par le responsable .

Responsable : Cet acteur attribue les factures simulées aux contrôleurs et assure le suivi de la vérification. Il peut également exécuter toutes les missions assurées par le conseiller client

User : Cet acteur présente le conseiller client (contrôleur) chez Orange Tunisie qui peut effectuer la vérification des factures affectées.

2.2 Identification des besoins

2.2.1 Besoins fonctionnels

- Gérer les utilisateurs : L'application doit vous permettre de :
 - Consulter la liste des utilisateurs.
 - Modifier des utilisateurs, leurs rôles et leurs statuts.
 - Désactiver des utilisateurs.

- Superviser les opérations :

L'application doit permettre de :

- Consulter l'historique des opérations : afficher la liste des affectations (chaque facture à qui elle a été affectée et son état valide ou non).
- Consulter les statistiques des opérations : il s'agit de générer un tableau de bord composé de graphiques représentant les statistiques sur les factures simulées qui sont vérifiées par rapport aux factures affectées.

- Gérer les factures simulées :

L'application doit permettre aux utilisateurs de :

- Consulter la liste des factures et effectuer une recherche parmi elles.
- Affecter les factures aux users.

- Vérifier les factures :

L'application doit permettre aux utilisateurs de :

- Consulter la liste des factures.
- Consulter une facture.
- Vérifier les critères de contrôle.
- Valider la facture ou en cas d'anomalie la réclamer.

- Automatiser les notifications :

L'application doit permettre de :

- Générer des notifications pour informer l'utilisateur par l'ajout de nouvelles factures à vérifier.
- Générer des notifications pour informer le gestionnaire par l'ajout d'une réclamation.

- Sécurisation de l'application : l'accès à l'application doit être permis uniquement aux personnes autorisées. L'accès aux différentes pages doit être contrôlé. Tous les utilisateurs doivent s'authentifier pour utiliser seulement les services autorisés de cette application.

2.2.2 Besoins non fonctionnels

- Disponibilité : l'application doit être accessible à l'utilisateur à tout moment et être performante. Il suffit de vous identifier vous êtes autorisé à travailler sur cette dernière.
- La convivialité : L'application doit être simple à utiliser, c'est-à-dire que les interfaces

utilisateur doivent être ergonomiques, adaptées à l'utilisateur. C'est pour ces raisons que nous avons développé une plate-forme claire, lisible et facile à comprendre et à naviguer.

- Maintenance : Le code de l'application doit être bien lisible, commenté et facile à comprendre afin de pouvoir le maintenir de manière simple et rapide.
- Rapidité : le temps de réponse doit être acceptable.

2.3 Modélisation des besoins

2.3.1 Langage de modélisation

Comme langage de modélisation nous avons choisi UML (Unified Modeling Language) qui est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Le langage de modélisation peut servir de modèle pour un projet et assurer une architecture d'information structurée ; il peut également aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour des spécialistes externes. [3]

2.3.2 Identification des cas d'utilisation globales

Les cas d'utilisation de l'application sont déduits des besoins fonctionnels mentionnés ci-dessus. Chaque besoin fonctionnel donne lieu à un cas d'utilisation. La figure suivante montre le diagramme de cas d'utilisation globale qui reflète les principaux besoins fonctionnels.

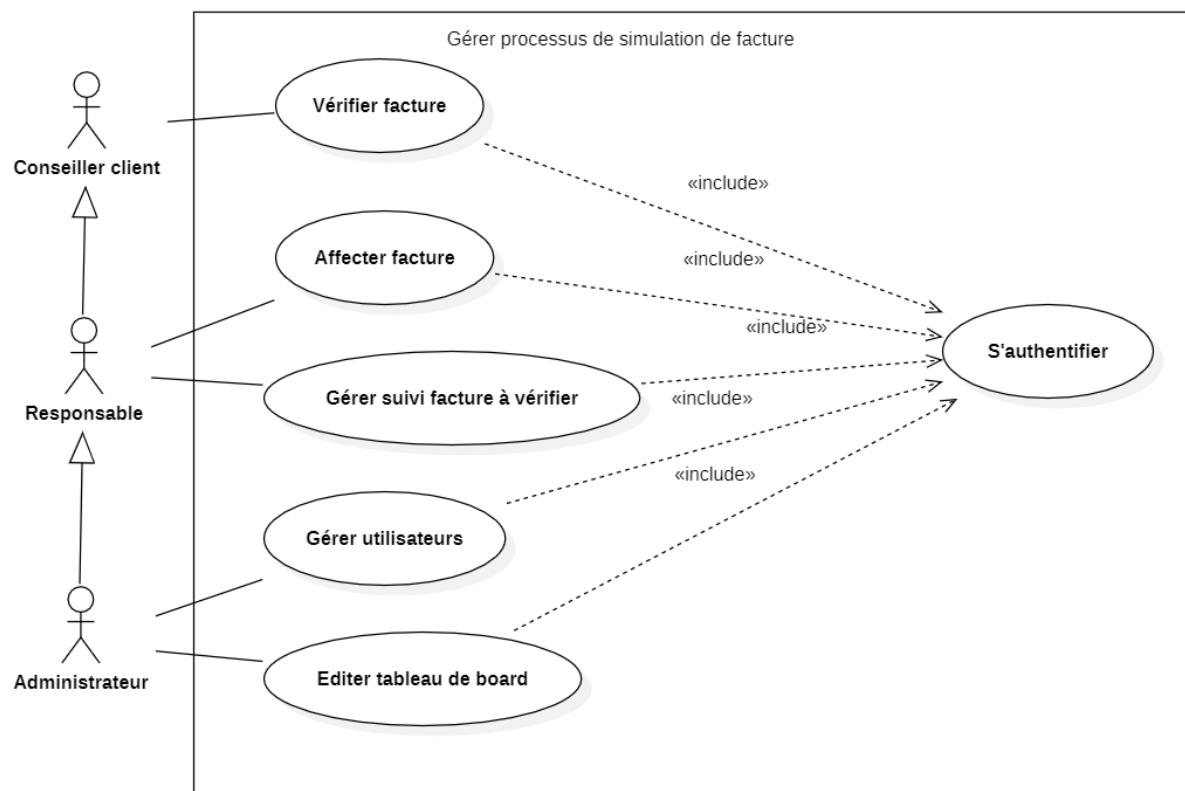


FIGURE 2.1 – Diagramme de cas d'utilisation global

Conclusion

Ce deuxième chapitre a été consacré à l'identification des acteurs et à l'analyse des besoins fonctionnels et non fonctionnels, puis nous avons terminé ce chapitre par la modélisation des besoins sous la forme d'un diagramme général de cas d'utilisation. Dans le chapitre suivant, nous abordons la partie conceptuelle du projet.

CHAPITRE 3

CONCEPTION

3.1	Conception architecturale	19
3.1.1	Architecture opérationnelle	19
3.1.2	Architecture technique	20
3.1.3	Architecture logicielle de l'application : MVC	21
3.2	Conception détaillée	21
3.2.1	Description de la vue statique	21
3.2.2	Description de la vue dynamique	30

Introduction

Dans ce chapitre, nous allons introduire la partie conception du projet. Celle qui contient la conception de l'architecture et la conception détaillée : les différents diagrammes de cas d'utilisation, de classe et de séquence dans le but de comprendre l'interaction entre les divers acteurs de l'application.

3.1 Conception architecturale

3.1.1 Architecture opérationnelle

L'architecture opérationnelle de notre application est constituée d'un client, d'un serveur d'applications et d'un serveur de base de données. Il s'agit donc d'une architecture 3-tiers.

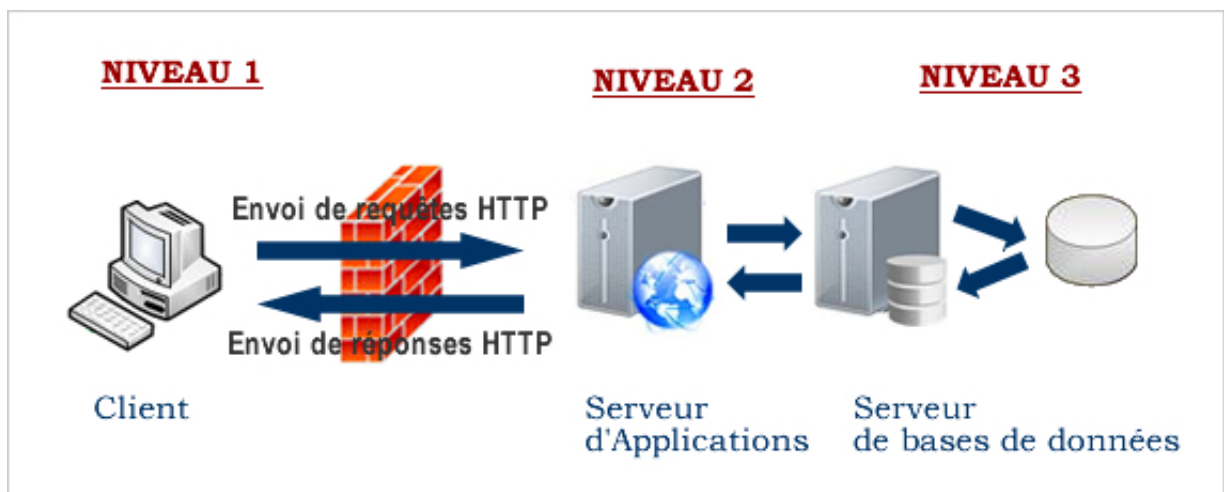


FIGURE 3.1 – Architecture opérationnelle de l'application

Comme l'indique la figure 3.1, les éléments constitutifs de cette architecture sont les suivants :

- Client : qui représente le poste de travail avec lequel les utilisateurs accèdent aux différentes interfaces de l'application à l'aide d'un navigateur web.
- Serveur d'application : C'est le composant matériel qui héberge les composants logiciels de base et l'application développée.
- Serveur de données : représente le serveur de base de données utilisé pour la partie persistance de notre application.

Dans le contexte du langage de modélisation unifié (UML), un diagramme de déploiement fait partie de la catégorie des diagrammes structurels, car il décrit un aspect du système même.[4]

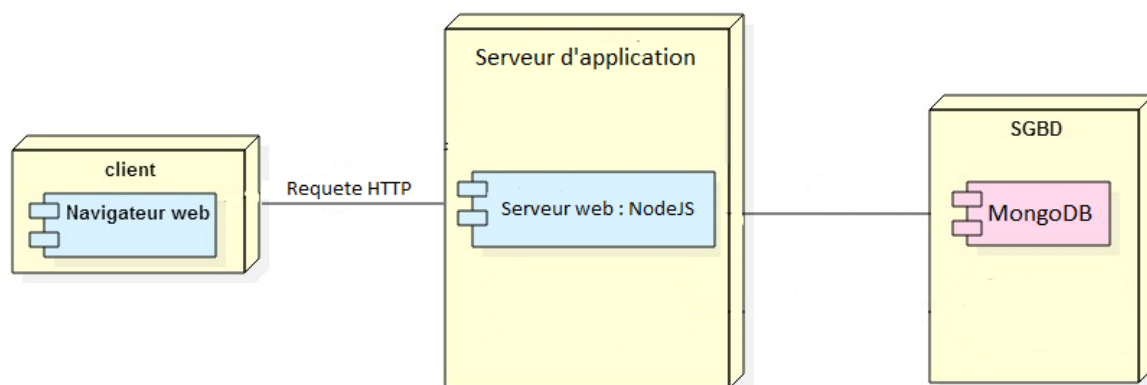


FIGURE 3.2 – Diagramme de déploiement

3.1.2 Architecture technique

L'architecture technique est une vue du fonctionnement de notre application, notamment la séparation entre le BackEnd et le FrontEnd, et les liens entre les différentes couches. Cette figure correspond à l'architecture technique de notre application.

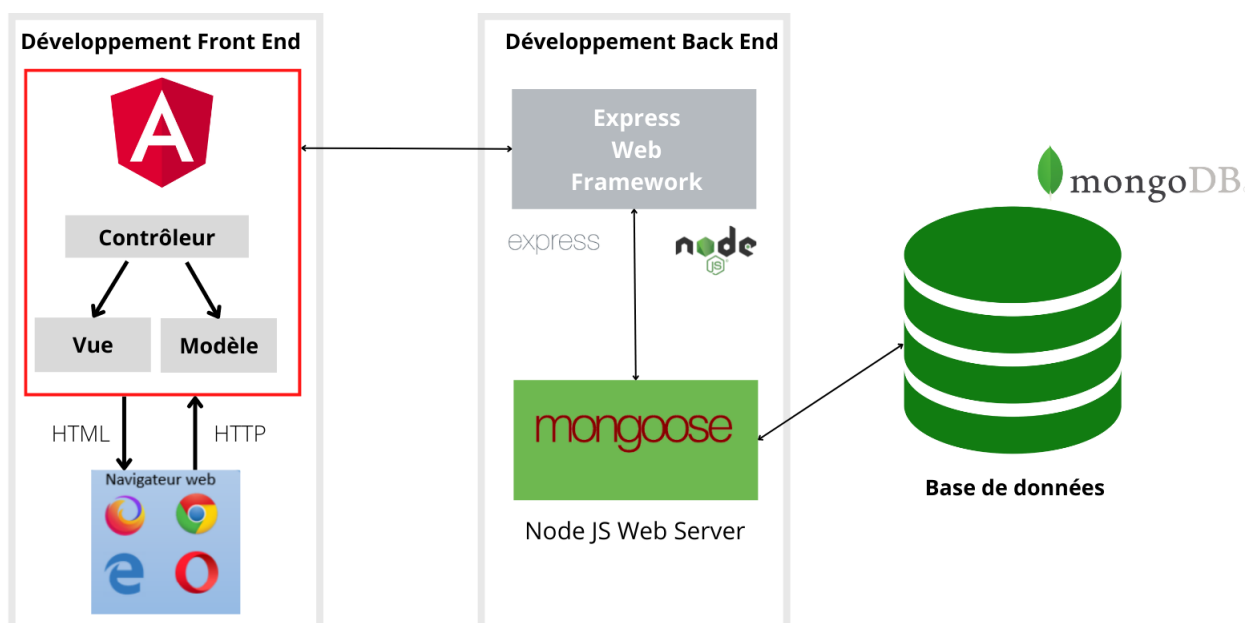


FIGURE 3.3 – Architecture technique de l'application

3.1.3 Architecture logicielle de l'application : MVC

MVC (Modèle-Vue-Contrôleur) est un modèle de conception logicielle utilisé pour implémenter les interfaces utilisateur, les données et la logique de contrôle. Il met l'accent sur une séparation entre la logique métier du logiciel et l'affichage. Cette "séparation des préoccupations" permet une meilleure répartition des tâches et une meilleure maintenance[5].

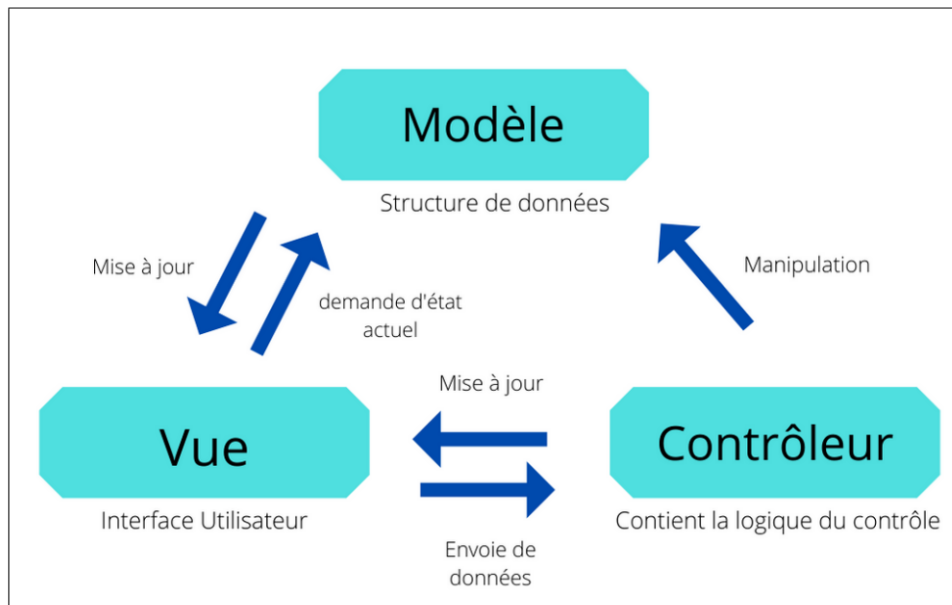


FIGURE 3.4 – Le modèle de conception MVC

- Le modèle : définit les données que l'application doit contenir. Si l'état de ces données change, le modèle notifie la vue et parfois le contrôleur (si une logique différente est nécessaire pour contrôler la vue mise à jour).
- La vue : définit la façon dont les données de l'application doivent être affichées.
- Le contrôleur : contient une logique qui met à jour le modèle et / ou la vue en réponse aux entrées des utilisateurs de l'application.

3.2 Conception détaillée

3.2.1 Description de la vue statique

3.2.1.1 Raffinement des cas d'utilisation de l'application web

Dans cette partie, nous allons voir les détails de chaque cas d'utilisation identifié dans le diagramme de cas d'utilisation global.

3.2.1.1.1 Cas d'utilisation s'authentifier

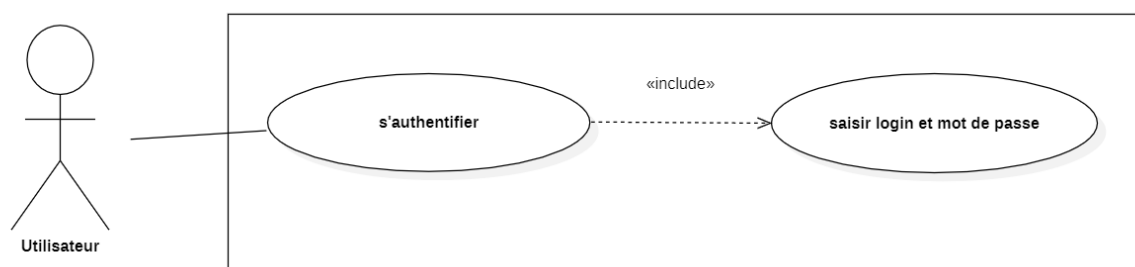


FIGURE 3.5 – Raffinement du cas d'utilisation « s'authentifier»

Titre	S'authentifier
Acteurs	Administrateur ou Responsable ou le conseiller client
Pré condition	Acteur doit être inscrit (existe dans la base de données)
Description	Ce cas d'utilisation permet à un utilisateur d'accéder aux fonctionnalités de l'application par la saisie de son e-mail et de mot de passe
Scénario nominal	1. l'utilisateur saisit son e-mail et son mot de passe 2. Le système vérifie les données saisies : existence de l'utilisateur et validité du mot de passe 3. Le système récupère le privilège associé à l'utilisateur authentifié avec succès. 4. Le système redirige l'utilisateur vers la page d'accueil
Scénario alternatif	Si les données ne sont pas valides, le système renvoie un message d'erreur
Post conditions	Authentification réussie, le système redirige l'utilisateur vers la page d'accueil. Authentification non réussie.
Compléments	Le cryptage du mot de passe

TABLE 3.1 – Description du Cas d'utilisation : s'authentifier

3.2.1.1.2 Cas d'utilisation : Gérer utilisateurs :

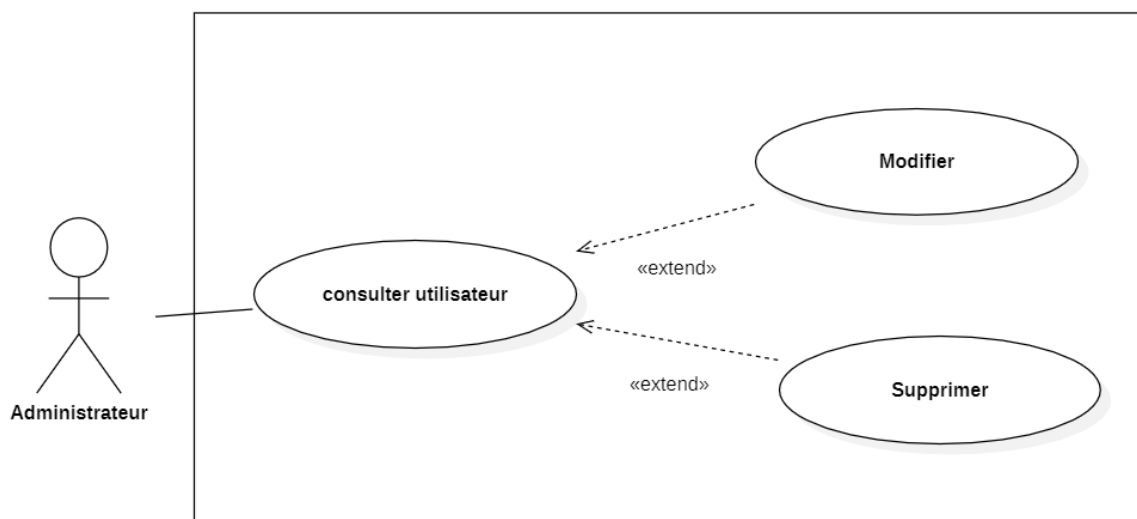


FIGURE 3.6 – Raffinement du cas d'utilisation « Gérer utilisateurs »

Titre	Consulter utilisateur
Acteurs	Administrateur
Pré condition	Acteur authentifié
Description	Ce cas d'utilisation permet à l'administrateur de gérer les utilisateurs de l'application
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur demande la liste d'utilisateurs 2. Le système affiche la liste des utilisateurs
Modifier utilisateur	<ol style="list-style-type: none"> 1. Le système affiche les informations relatives à l'utilisateur sélectionné 2. L'administrateur édite les données relatives à l'utilisateur (le statut et le rôle) 3. Le système vérifie les données saisies 4. Le système enregistre les données relatives à l'utilisateur dans la base de données

Supprimer utilisateur	<p>1.L'administrateur sélectionne un utilisateur à supprimer.</p> <p>2.Le système demande la confirmation de suppression.</p> <p>3.L'administrateur confirme la suppression.</p> <p>4.Le system supprime les données relatives à l'utilisateur sélectionné de la base de données.</p>
Scénario alternatif	Si les données saisies ne sont pas valides, le système renvoie un message d'erreur.
Post conditions	<p>Liste des utilisateurs affichée.</p> <p>Utilisateur modifié ou supprimé</p>

TABLE 3.2 – Description du Cas d'utilisation : Gérer utilisateur

3.2.1.1.3 Cas d'utilisation : Affecter facture :

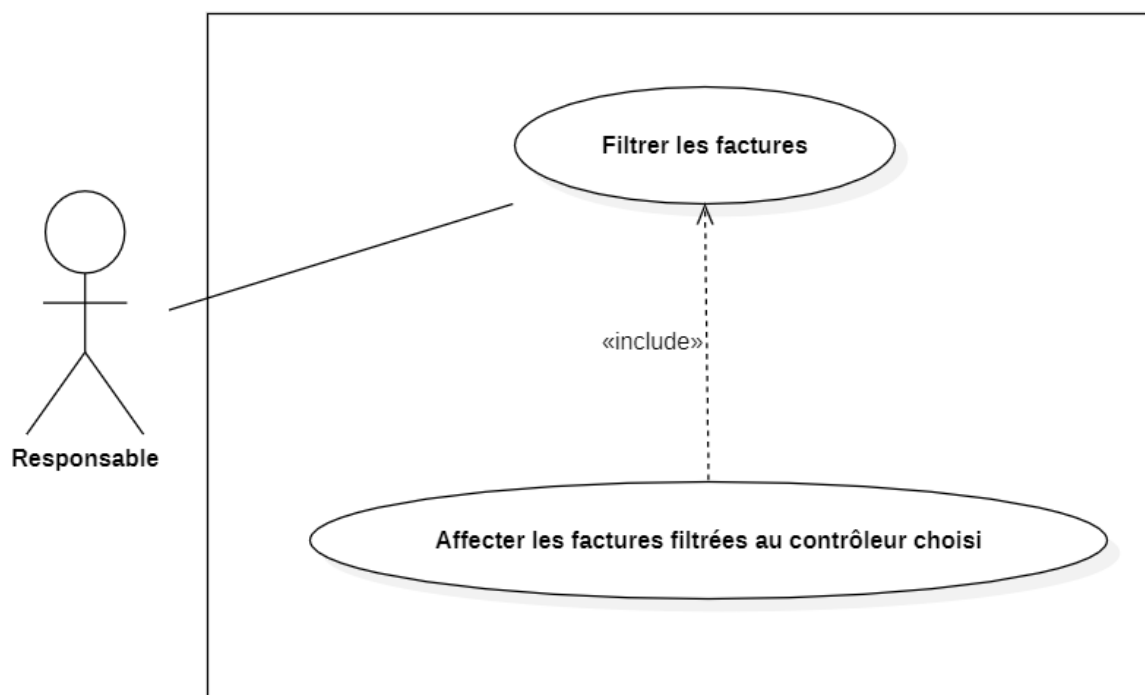


FIGURE 3.7 – Raffinement du cas d'utilisation « Affecter facture »

Titre	Affecter facture
Acteurs	Responsable
Pré condition	Acteur authentifié. L'existence des factures pour les affecter.
Description	Ce cas d'utilisation permet la filtration des factures et de les affecter à un contrôleur
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur applique un filtre sur les factures. 2. Le système affiche les factures filtrées et la liste des contrôleurs. 3. Le responsable sélectionne un contrôleur à partir de la liste déroulante des contrôleurs. 4. Le système enregistre les informations dans la base de données. 5. Le système génère une nouvelle notification.
Post conditions	Affectation d'une facture à un contrôleur Notification envoyée

TABLE 3.3 – Description du Cas d'utilisation : Affecter facture

3.2.1.1.4 Cas d'utilisation : Vérifier factures :

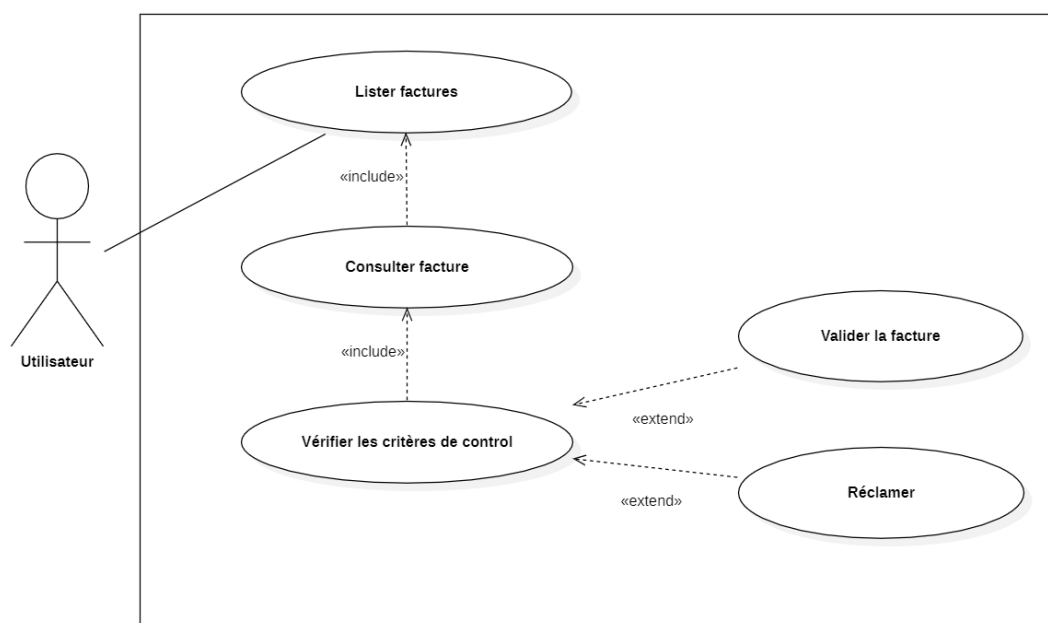


FIGURE 3.8 – Raffinement du cas d'utilisation « Vérifier factures »

Titre	Vérifier factures
Acteurs	Conseiller client
Pré condition	Acteur authentifié
Description	Ce cas d'utilisation permet à l'utilisateur de vérifier les factures affectées
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur demande la liste des factures à vérifier. 2. Le système affiche la liste des factures à vérifier. 3. L'utilisateur sélectionne une facture à vérifier. 4. Le système affiche le PDF, les informations relatives à la facture choisie et les critères de control. 5. L'utilisateur confirme les critères. 6. Le système insère les données dans la base de données.
Scénario alternatif	Si les critères ne sont pas valides, l'utilisateur saisie son réclamation et le système insère les données dans la base de données.
Post conditions	<p>Liste des factures affichée.</p> <p>Facture valide ou invalide.</p>

TABLE 3.4 – Description du Cas d'utilisation : Vérifier factures

3.2.1.1.5 Cas d'utilisation : Gérer suivi factures :

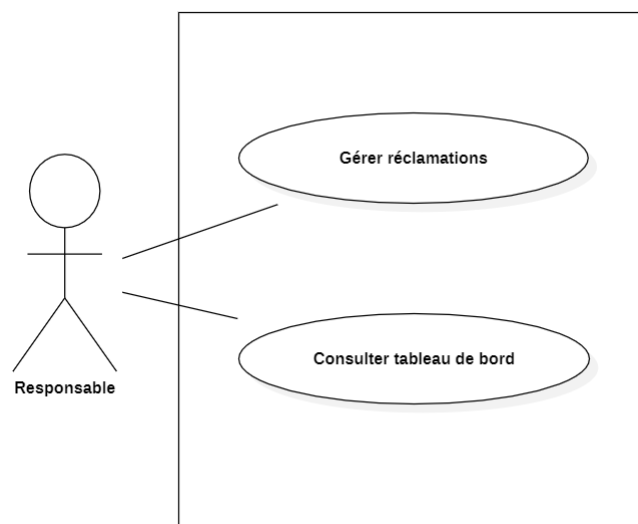


FIGURE 3.9 – Raffinement du cas d'utilisation « Gérer suivi factures »

Titre	Gérer suivi factures
Acteurs	Responsable
Pré condition	Acteur authentifié
Description	Ce cas d'utilisation permet au responsable de suivre le processus
Gérer réclamations	<ol style="list-style-type: none"> 1. L'utilisateur remplit les champs de saisie de réclamation. 2. Le système enregistre les données dans la base de données. 3. L'administrateur demande la liste des réclamations. 4. Le système affiche la liste des réclamations Le responsable valide les réclamations.
Consulter tableau de board	<ol style="list-style-type: none"> 1. Le responsable choisit l'interface « Tableau de bord» 2. Le système affiche les statistiques des factures affectées.

TABLE 3.5 – Description du Cas d'utilisation : Gérer suivi factures

3.2.1.1.6 Cas d'utilisation : Editer tableau de board :

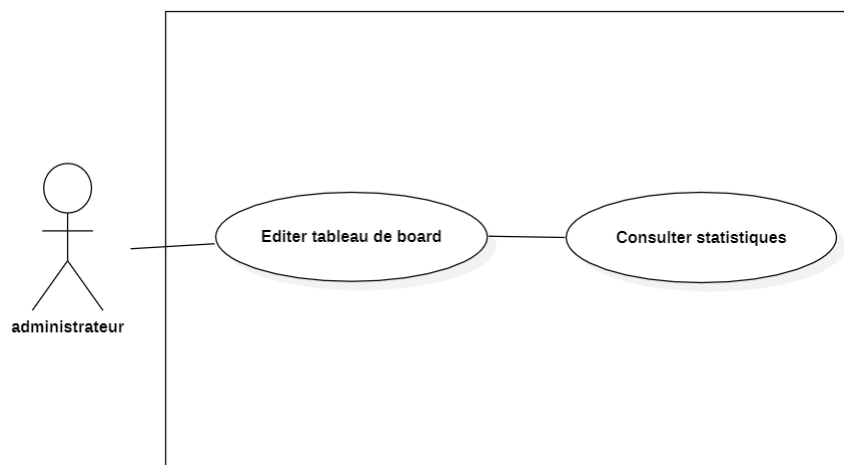


FIGURE 3.10 – Raffinement du cas d'utilisation « Editer tableau de board »

Titre	Consulter statistiques
Acteurs	L'administrateur
Pré condition	Acteur authentifié
Description	Ce cas d'utilisation permet de Consulter les statistiques sur les opérations
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur choisit l'interface « Tableau de bord » 2. le système affiche le tableau de bord qui contient un graphique qui représente les statistiques du nombre des factures valides par rapport au nombre des factures affectées

TABLE 3.6 – Description du Cas d'utilisation : Consulter statistiques

3.2.1.2 Diagramme de classe

Il représente les classes intervenant dans le système. Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations. Il est considéré comme le plus important de la modélisation orienté objet.

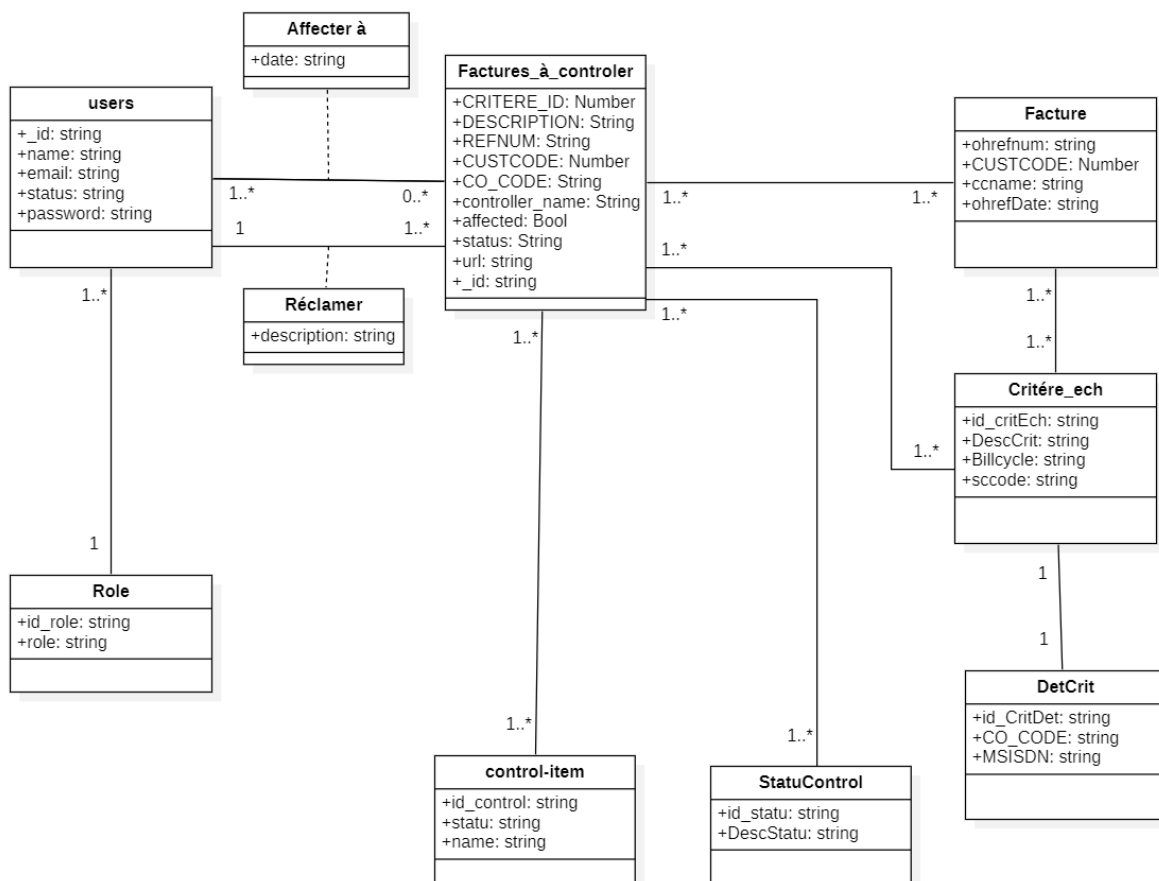


FIGURE 3.11 – Diagramme de classe

Nom de la classe	Description de la classe
Users	Elle désigne l'utilisateur du système qui profite du service fourni par l'application. Cette classe contient des informations sur les utilisateurs, qui ont été enregistrés après leur inscription.
Role	Cette classe décrit la liste des rôles à affecter pour chaque utilisateur.
Facture-à-controler	Cette classe représente les factures simulées à vérifier pour le mois actuel.
Control-item	Cette classe contient les critères de contrôle nécessaires pour la vérification de facture et leur statut valide ou invalide.
StatuControl	Cette classe représente le statut d'une facture (affectée, non affectée, valide, invalide).
DetCrit	Cette classe représente la description du critère d'échantillonnage
Critère-ech	Cette classe représente les critères d'échantillon
Facture	Cette classe représente la liste des toutes les factures simulées.

TABLE 3.7 – Description textuelle du diagramme de classes

3.2.2 Description de la vue dynamique

3.2.2.1 Diagrammes des séquences

Dans ce qui suit, nous présentons les diagrammes de séquence qui correspondent aux diagrammes de cas d'utilisation de notre système.

3.2.2.1.1 Diagramme de séquence « S'authentifier »

Dans notre application, chaque utilisateur (que ce soit un conseiller client ou un responsable ou un administrateur) doit s'authentifier. L'authentification se fait via la page d'authentification où l'utilisateur doit saisir son LOGIN (e-mail) et son MOT DE PASSE. Après la saisie des données, un contrôle est effectué au niveau du contrôleur pour vérifier la validité des données transmises. Si toutes les données sont valides, elles seront transférées au même contrôleur qui vérifie l'existence de l'utilisateur dans l'entité "Users"; s'il existe, et qu'il a un statut actif, il sera redirigé vers la page d'accueil; sinon, il devra s'authentifier une autre fois ou attendre la modification de l'administrateur.

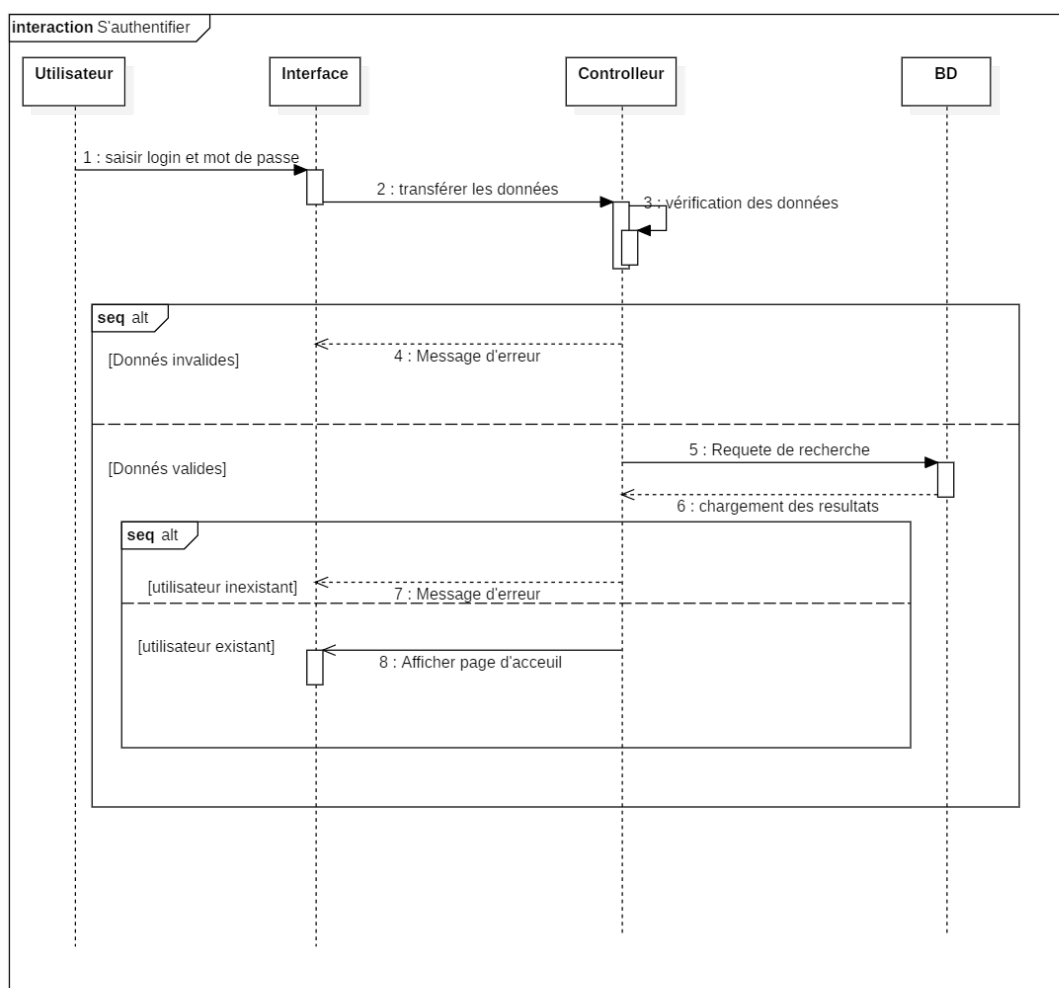


FIGURE 3.12 – Diagramme de séquence « S'authentifier »

3.2.2.1.2 Diagramme de séquence « Modifier utilisateur »

Pour mettre à jour un utilisateur l'administrateur sélectionne un utilisateur à modifier, une recherche des données de l'utilisateur est effectuée et affichée, l'administrateur modifie le rôle ou le statut de l'utilisateur. Si toutes les données sont valides, elles sont transférées vers la base de données, sinon un message d'erreur est affiché.

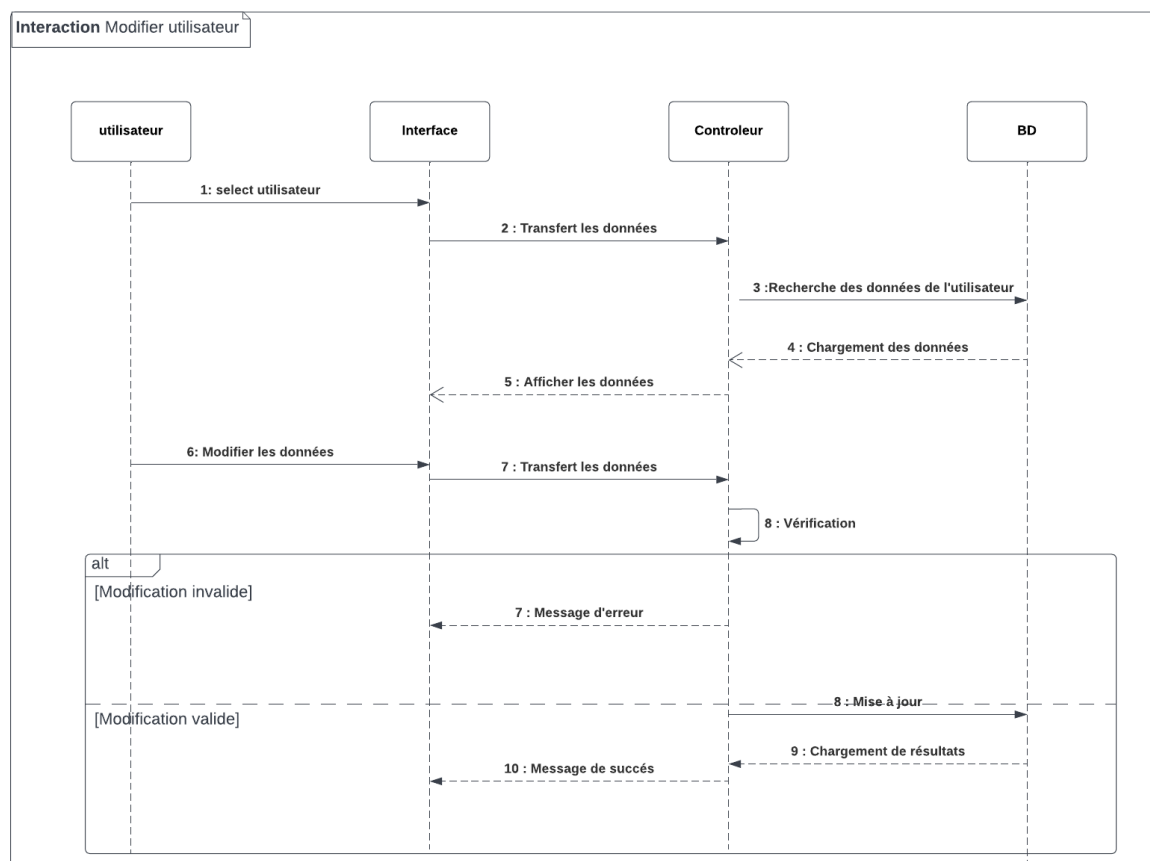


FIGURE 3.13 – Diagramme de séquence « Modifier utilisateur »

3.2.2.1.3 Diagramme de séquence « Affecter facture »

Pour affecter une facture à un contrôleur, le responsable doit s'authentifier. Ensuite, il sélectionne les factures à affecter, puis il demande la liste des contrôleurs et sélectionne un contrôleur. Si tout se passe bien, une nouvelle notification sera générée en sauvegardant les données dans la base de données.

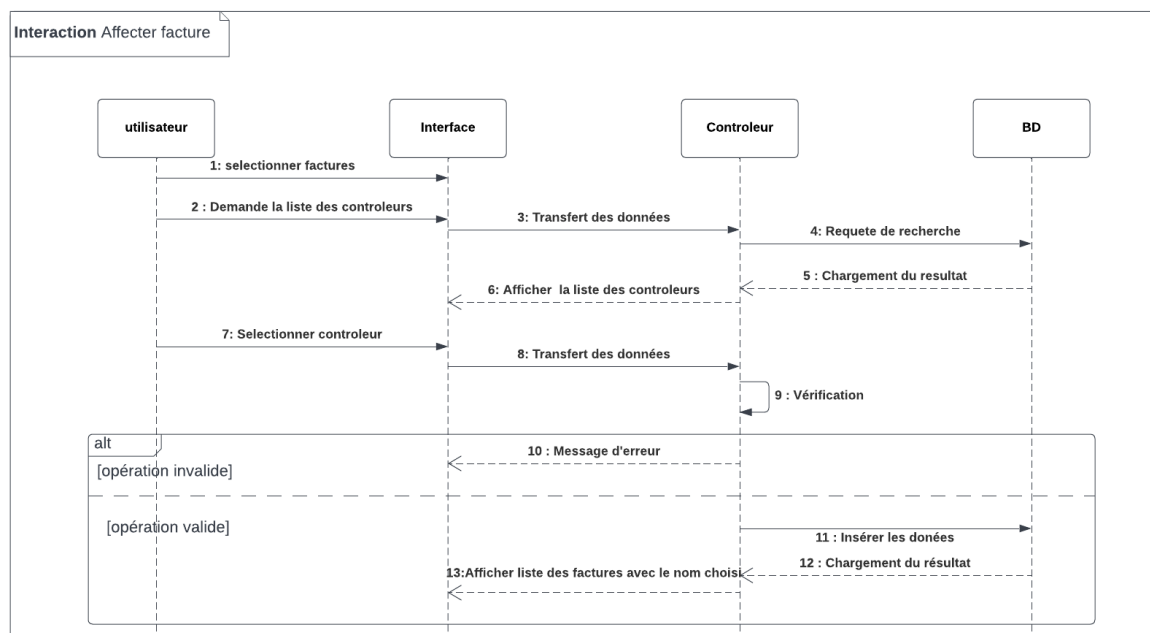


FIGURE 3.14 – Diagramme de séquence « Affecter facture »

3.2.2.1.4 Diagramme de séquence « Vérifier facture »

Pour contrôler une facture par un contrôleur, il doit d'abord s'authentifier. Ensuite, une requête de recherche est effectuée dans la base de données pour afficher le pdf qui correspond à cette facture, puis le contrôleur vérifie les critères de contrôle. Si les critères sont valides, les données seront enregistrées dans la base de données. Dans le cas contraire, ou les critères ne sont pas valides, une zone de texte s'affiche pour rédiger une réclamation et enfin l'insérer dans la base de données.

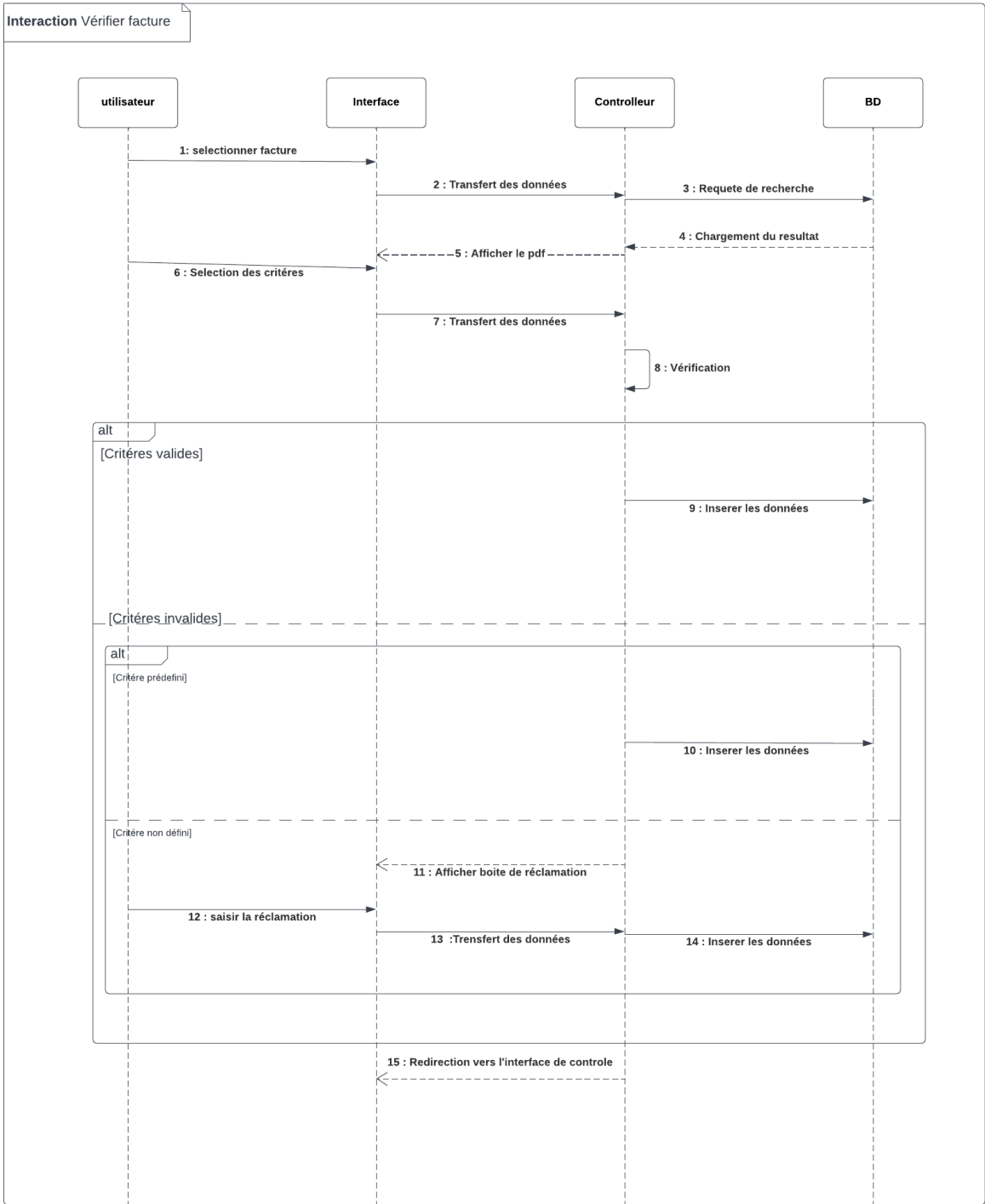


FIGURE 3.15 – Diagramme de séquence « Vérifier facture »

3.2.2.2 Diagramme d'activité

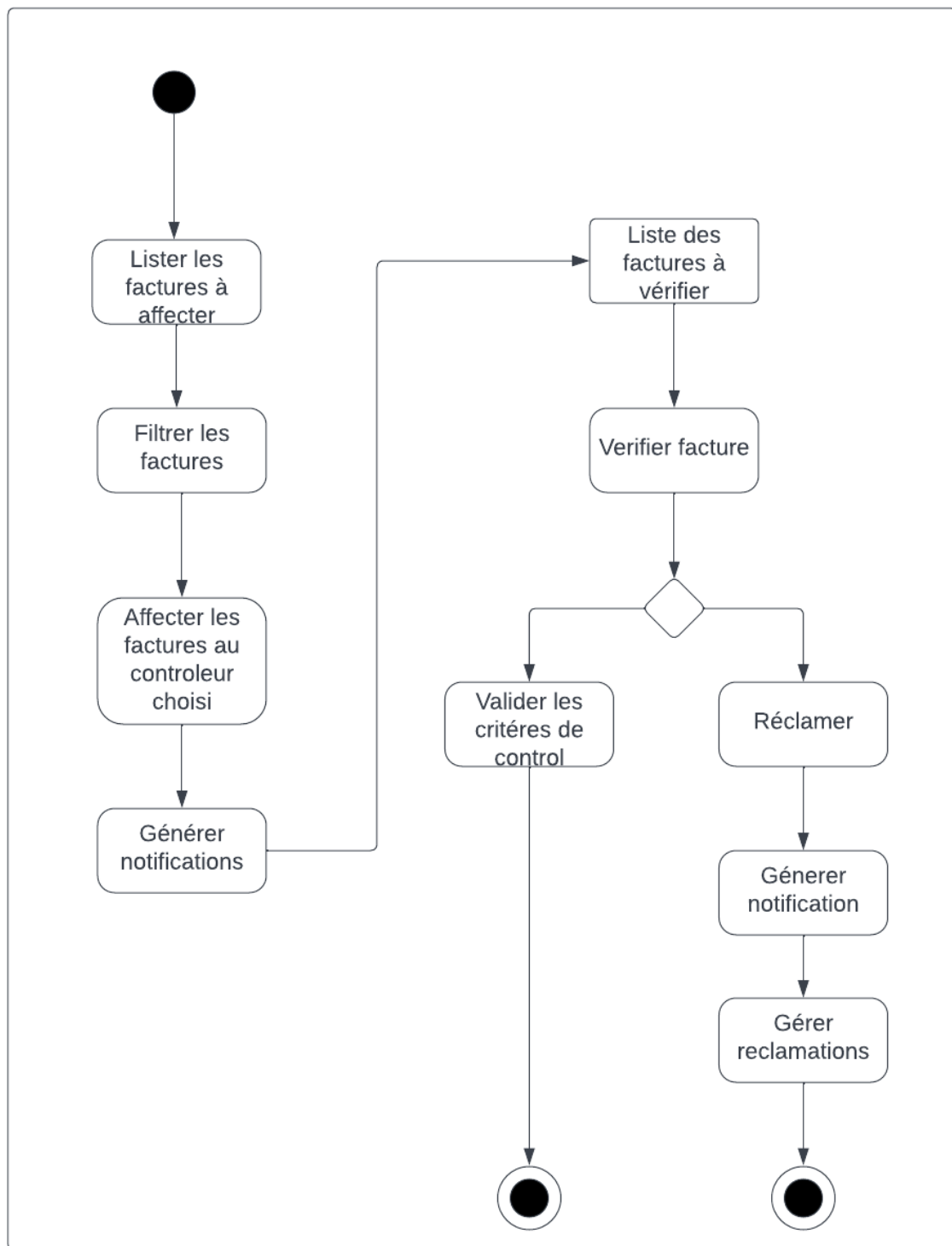


FIGURE 3.16 – Diagramme d'activité »

Conclusion

Dans ce chapitre, nous avons présenté les différents aspects conceptuels : conception architecturale et conception détaillée. Nous avons créé quelques diagrammes UML statiques, dont le raffinement de cas d'utilisation, le diagramme de classe et les diagrammes dynamiques tels que le diagramme de séquence et le diagramme d'activité. Nous pouvons maintenant aborder la dernière partie de notre rapport " la phase de réalisation " cette dernière partie se concentre sur le développement de l'application et également sur l'observation des résultats obtenus.

CHAPITRE 4

RÉALISATION ET TEST

4.1	Environnement matériel	38
4.2	Environnement logiciel	38
4.3	Technologies utilisées Application web	39
4.4	Présentation de l'application réalisée	41
4.4.1	Inscription	41
4.4.2	Authentification	41
4.4.3	Dashboard	42
4.4.4	Gestion utilisateurs	43
4.4.5	Affecter facture	44
4.4.6	Vérifier facture	44

Introduction

Dans ce chapitre, nous allons présenter la réalisation (Implémentation) de la solution proposée dans le chapitre précédent. La première partie sera donc consacré de la description de l'environnement matériel et logiciel de notre application. Dans la deuxième partie, nous décrivons les principaux modules constituant l'application développée à travers les différentes interfaces graphiques correspondantes.

4.1 Environnement matériel

Concernant la partie matérielle, il est important d'utiliser un équipement puissant au niveau du processeur et de la mémoire pour pouvoir gagner en temps de réponse lors de l'exécution de l'application. Ainsi notre environnement matériel est composé d'un ordinateur portable ayant les caractéristiques présentées dans le tableau ci-dessous.

Marque	HP
Processeur	Intel(R)Core(TM)i5-10300H CPU @2.50GHz 2.50GHz
Memoire	16 GO
Disque dur	512 GO
Système d'exploitation	Windows 10

TABLE 4.1 – Environnement matériel

4.2 Environnement logiciel

La mise en place de ce projet a nécessité un certain nombre de logiciels.

- Outil de modélisation UML « StarUML » [6] :

StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose ou Together. Étant simple d'utilisation, nécessitant

peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation.

- Visual Studio Code :[7]

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js.

- Postman :[8]

Parmi les nombreuses solutions pour interroger ou tester webservices et API, Postman propose de nombreuses fonctionnalités, une prise en main rapide et une interface graphique agréable. Postman existe sous la forme d'une App (Windows/MacOS/Linux) et d'une Chrome App.

- MongoDB Compass :[9]

est une interface graphique permettant d'explorer, d'analyser et d'interagir avec le contenu stocké dans une base de données MongoDB sans connaître ni utiliser de requêtes. MongoDB Compass est une bien meilleure alternative au shell Mongo. Compass peut effectuer toutes les opérations effectuées par Mongo Shell et plus encore[10]

- LaTeX [10] :

est Un système de préparation de documents LaTeX est un système de composition de haute qualité; il comprend des fonctionnalités destinées à la production de documentation technique et scientifique. LaTeX est le standard de facto pour la communication et la publication de documents scientifiques.

4.3 Technologies utilisées Application web

Angular [11] est une plate-forme et un framework permettant de créer des applications clientes d'une seule page à l'aide de HTML et de TypeScript. Angular est écrit en TypeScript. Il implémente les fonctionnalités de base et facultatives sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans vos applications.

MongoDB [12] est un programme de gestion de base de données NoSQL open source . NoSQL est utilisé comme alternative aux bases de données relationnelles traditionnelles. Les bases de données NoSQL sont très utiles pour travailler avec de grands ensembles de données distribuées.

MongoDB est un outil qui peut gérer des informations orientées document, stocker ou récupérer des informations.

Node.js[13] sert à faire du Javascript server side, et peut être utilisé dans des applications de bases de données, la plus populaire étant MySQL. Ce n'est ni un framework ni un serveur. Node.js est souvent confondu avec ce dernier, car c'est sa base : créer des applications en temps réel, où le serveur a la possibilité de transmettre de l'information au client.

Express.js [14] est un cadre d'exploitation libre et gratuit pour Node.js. Il comporte des fonctionnalités, des outils, des plugins et des paquets qui aident à simplifier les processus de développement. Également connu sous le nom d'Express, TJ Holowaychuk l'a créé et publié pour la première fois en 2010. Il s'appuie sur les principes et les approches de Node.js et les utilise pour créer des applications web. Express.js est essentiel pour le développement d'applications, notamment pour l'automatisation et l'intégration des technologies.

HTML :[15] L'HTML est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage tout comme XML. Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme le texte et de placer des éléments interactif, tel des liens, des images.

Css : [16] est l'acronyme anglais de « Cascading Style Sheets » qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

Bootstrap : [17] est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

4.4 Présentation de l'application réalisée

4.4.1 Inscription

En premier lieu, l'utilisateur doit s'inscrire en écrivant son e-mail, son nom et son mot de passe qui doit dépasser les 4 caractères.

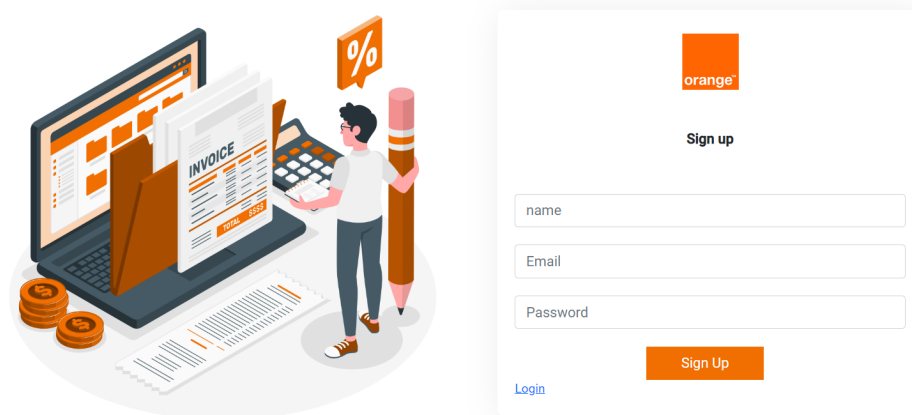


FIGURE 4.1 – Interface d'inscription

4.4.2 Authentification

Afin de permettre à l'utilisateur de bénéficier des fonctionnalités de l'application, il doit tout d'abord s'authentifier. Il doit saisir son e-mail et son mot de passe comme indiqué sur la figure ci-dessous.

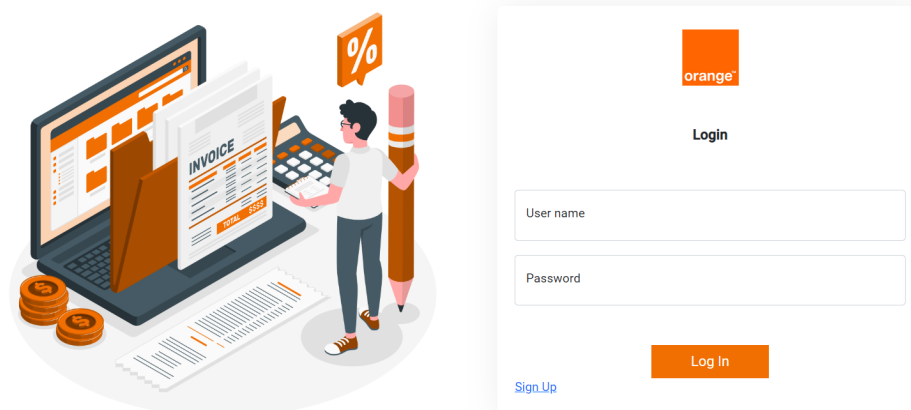


FIGURE 4.2 – Interface d'authentification

4.4.3 Dashboard

Cette interface permet aux administrateurs de suivre les statistiques des factures affectées par rapport aux factures simulées et le pourcentage de factures valides par rapport aux factures affectées.

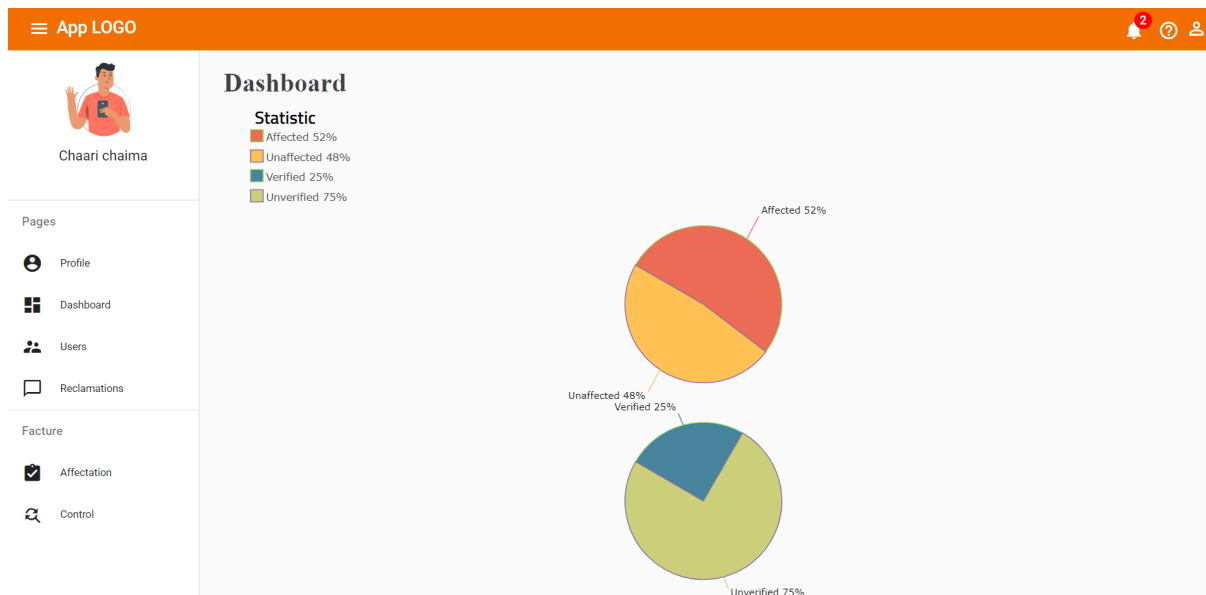


FIGURE 4.3 – Dashboard

4.4.4 Gestion utilisateurs

Donner à un nouvel utilisateur un rôle et un statut pour accéder à l'application. L'administrateur choisit l'utilisateur, clique sur l'icône de modification et édite les données souhaitées. Maintenant L'utilisateur peut alors s'authentifier.

Name	email	Role	Status	Action
chaima	chaarichaima11@gmail.com	admin	Active	
sana chaari	sana@gmail.com			
ala ncibi	ala@gmail.com	responsable	Active	
lina	lina@gmail.com	responsable	Active	
ali	ali@gmail.com			
mabrouk cherif	mabrouk@gmail.com			
meriam baroudi	meriam@gmail.com			

FIGURE 4.4 – Interface de la liste des utilisateurs

Edit user !

Name:

Email:

Role:

Status: ☒ Active ☐ Inactive

FIGURE 4.5 – Dialogue de modification d'utilisateur

4.4.5 Affecter facture

Pour affecter une facture, le responsable doit filtrer, sélectionner les factures, choisir le nom du contrôleur à partir de la liste déroulante et valider l'action en cliquant sur le bouton "submit".

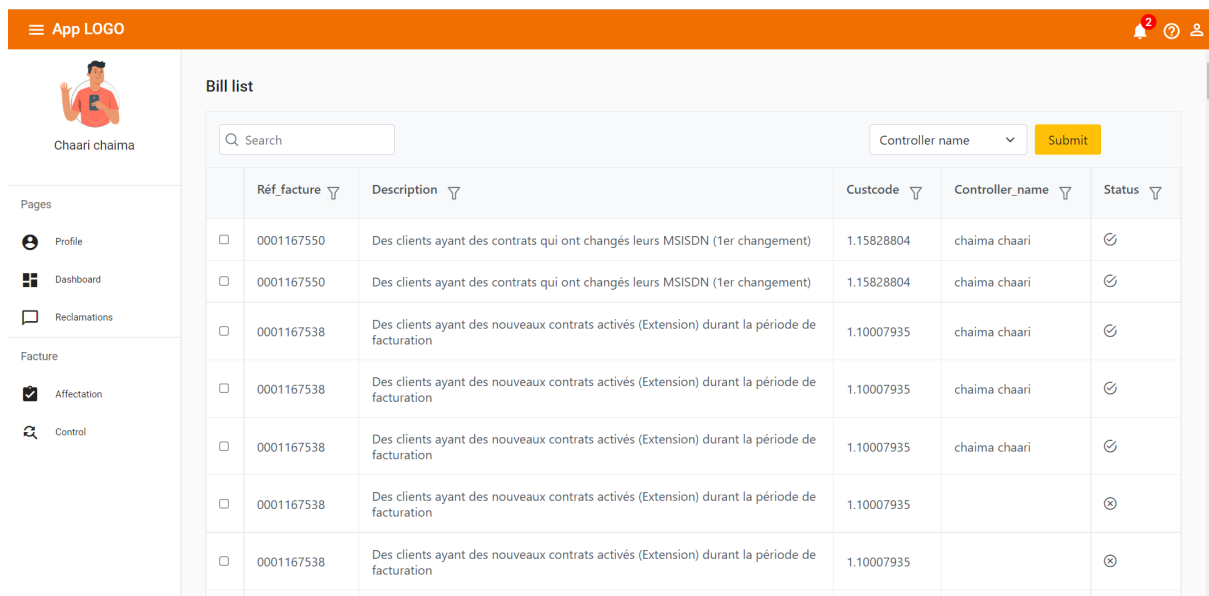


FIGURE 4.6 – Interface d'affectation

4.4.6 Vérifier facture

Après l'authentification, l'utilisateur trouve la liste des factures qui lui ont été attribuées. Dans un premier temps, il choisit une facture à vérifier en cliquant sur "check" il sera automatiquement redirigé vers une page qui contient le pdf de la facture et les critères à vérifier. En second lieu, l'utilisateur lit le pdf si les critères sont valides la facture sera valide sinon si les critères sont invalides le champ de saisie de la réclamation sera obligatoire et la facture sera invalide.

App LOGO

Chaari chaïma

Pages

Profile

Facture

Control

list of invoices to check

Réf_facture	Description	Custcode	Verification	Status
0001167550	Des clients ayant des contrats qui ont changés leurs MSISDN (1er changement)	1.15828804	Check	✓
0001167550	Des clients ayant des contrats qui ont changés leurs MSISDN (1er changement)	1.15828804	Check	✓
0001167538	Des clients ayant des nouveaux contrats activés (Extension) durant la période de facturation	1.10007935	Check	✓
0001167538	Des clients ayant des nouveaux contrats activés (Extension) durant la période de facturation	1.10007935	Check	✓
0001167538	Des clients ayant des nouveaux contrats activés (Extension) durant la période de facturation	1.10007935	Check	✓
0001167538	Des clients ayant des nouveaux contrats activés (Extension) durant la période de facturation	1.10007935	Check	⊗
0001167538	Des clients ayant des nouveaux contrats activés (Extension) durant la période de facturation	1.10007935	Check	⊗
0001167539	Des clients ayant des contrats désactivés durant la période de facturation	1.11054256	Check	⊗
0001167539	Des clients ayant des contrats désactivés durant la période de facturation	1.11054256	Check	⊗

FIGURE 4.7 – Interface de contrôle

App LOGO

Chaari chaïma

Pages

Profile

Facture

Control

votre facture
avril 2022

votre compte
N° client : ACNT05436602
nombre de contrats : 2
facture N° : 0000417476
émise le : 01/04/2022

M. ADSL RECETTE
1001 Republique

bonjour

total hors taxes
119.439
TVA 07%
8.361
(montant soumis à la TVA 07%)
montant TTC
128.400

montant à payer
128.400
date limite de paiement
01/04/2022

عسلامة

المبلغ الجملي الحالي من الأداء
الاداء على القيمة المضافة 07%
(المبلغ الخاضع للاداء 07%)
المبلغ الجملي
المبلغ المطلوب
آخر أجل للدفع

FIGURE 4.8 – Interface de contrôle

Page 45

Item	Statu
Layout	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
TVA	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
OCC	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
Remise	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
usage	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
Droit timbre	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
Montant TTC	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>
Montant HT	<input type="button" value="Valide"/> <input type="button" value="Invalide"/>

Reclamations

FIGURE 4.9 – Interface de contrôle

Conclusion

Dans ce chapitre, nous avons d'abord présenté la description de l'environnement matériel et logiciel de notre application. Ensuite, nous avons exposé notre projet à travers les différentes interfaces graphiques. C'est avec ce chapitre que nous clôturons notre rapport.

CONCLUSION GÉNÉRALE

LA société Orange Tunisie étant soucieuse de l'importance de la bonne gestion des factures, nous a chargé de proposer une solution permettant de gérer son processus de simulation de factures. En effet, le projet réalisé vise à faciliter les tâches liées au suivi des factures simulées.

La réalisation de cette application nous a permis de nous rapprocher de la vie professionnelle dans le domaine du développement d'applications. De plus, elle nous a donné l'opportunité d'appliquer nos connaissances théoriques dans un cadre réel. Pour atteindre cet objectif, et après une étape de familiarisation avec le contexte technique et fonctionnel du projet, nous avons implémenté une application web en utilisant l'approche RUP.

Nous pensons que les objectifs du projet sont globalement atteints, même si notre travail pourrait être enrichi par les perspectives suivantes :

- Interfaçage du module d'authentification avec l'AD d'Orange (LDAP).
- Liaison de l'application avec la solution de ticketing d'Orange.
- Automatiser la récupération des factures simulées et de leurs données.
- Intégrer le module d'échantillonnage et de simulation à cette interface.

Nous avons commencé notre projet par une présentation du cadre général, afin d'assurer le bon déroulement du projet nous avons préparé un planning de travail précis. Ensuite, nous nous sommes concentrés sur l'analyse minutieuse des besoins fonctionnels et non fonctionnels auxquels l'application à développer doit répondre. Par la suite, nous avons

réalisé une conception architecturale et une conception détaillée de l'application en utilisant différents diagrammes du langage UML, à savoir : le diagramme de cas d'utilisation global et son raffinement, le diagramme de classe et les diagrammes de séquence. Afin de mieux comprendre la structure de l'application, un diagramme de déploiement a été mis en place. Nous sommes ensuite passés à la partie la plus importante de notre projet, la phase de réalisation, que nous avons présentée en insérant plusieurs captures d'écran de notre plate-forme, afin de donner une idée globale de ce que nous avons réalisé au cours de ce PFE.

Pour conclure, nous espérons que ce modeste travail est à la hauteur des attentes de la SOCETE ORANGE TUNISIE et des membres du jury.

BIBLIOGRAPHIE

- [1] <https://www.supinfo.com/articles/single/3093comparatifmethodesagiles>. [avril 2022]
- [2] <https://www.nutcache.com/fr/blog/methode-rup/> [avril 2022]
- [3] <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-programmation-orientee-objet> [avril 2022]
- [4] <https://www.lucidchart.com/pages/fr/diagramme-de-deploiement-uml> [avril 2022]
- [5] <https://developer.mozilla.org/fr/docs/Glossary/MVC> [avril 2022]
- [6] <https://inf1410.teluq.ca/teluqDownload.php?file=2014/01/INF1410-PresentationStarUML.pdf> [avril 2022]
- [7] <https://code.visualstudio.com/docs> [avril 2022]
- [8] <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web/> [avril 2022]
- [9] <https://www.bmc.com/blogs/mongodb-compass> [avril 2022]
- [10] <https://www.latex-project.org/> [avril 2022]
- [11] <https://angular.io/guide/architecture> [avril 2022]
- [12] <https://www.techtarget.com/searchdatamanagement/definition/MongoDB> [avril 2022]
- [13] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445306-node-js-definition-simple-et-utilisation-pratique/> [avril 2022]

- [14] <https://www.startechup.com/fr/blog/express-js-what-it-is-used-for-and-when-where-to-use-it-for-your-enterprise-app-development/>[avril 2022]
- [15] <https://glossaire.infowebmaster.fr/html>[avril 2022]
- [16] <https://glossaire.infowebmaster.fr/css/>[avril 2022]
- [17] <https://fr.wikipedia.org/wiki/Bootstrap>[avril 2022]

ملخص

التطبيق المطور هو جزء من مشروع نهاية العمل الحالي الذي يتم تنفيذه داخل شركة اورونج تونس هو جزء من مشروع نهاية الدراسة للحصول على درجة الإجازة في علوم الاعلامية في كلية العلوم بتونس. وهو يتألف من تصميم وتطوير حل يسمح بأتمتة عملية محاكاة الفواتير. وذلك لتقليل معدل الخطأ في الفواتير المنشأة.

الكلمات المفتاحية : فاتورة ، متابعة ، محاكاة.

Résumé

Le présent travail, réalisé au sein de la société Orange Tunisie s'inscrit dans le cadre du projet de fin d'études pour l'obtention du diplôme de licence en informatique à la Faculté des Sciences de Tunis. Il consiste à concevoir et développer une solution permettant l'automatisation du processus de simulation des factures. Ceci dans le but de réduire le taux d'erreur dans les factures générées.

Mots clés : Facture, suivi, simulation .

Abstract

The present work, carried out within the company Orange Tunisia is part of the end-of-studies project for the obtaining of the degree of licence in computer science at the Faculty of Sciences of Tunis. It consists in designing and developing a solution allowing the automation of the process of simulation of invoices. This in order to reduce the error rate in the generated invoices.

Key words : Invoice, follow-up, simulation.