

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Bejaia
Faculté des Sciences Exactes
Département Informatique



Mémoire de fin de cycle

En vue de l'obtention du Diplôme de Master Professionnel en
Informatique

Option:

GENIE LOGICIEL

Thème

**Conception et réalisation d'une application web pour la
facturation en ligne, cas d'étude : EPB**

Réalisé par :

- BENCHIKH Djamila
- HAMMICHE Lamia

Soutenu le 13 Octobre 2021 devant le jury composé de :

Encadrant : M.OUZEGGANE Redouane

Présidente : Mme. ALOUI Souraya

Examinatrice : Mme. MAAMERI Karima

Année universitaire 2020/2021

Remerciements

Nous tenons tout d'abord à remercier **DIEU** le tout puissant de nous avoir donné la santé et la volonté afin d'élaborer ce projet.

Ensuite nous remercions :

Nos encadreurs : Monsieur **OUZEGGANE** Redouane au niveau de l'université Abderrahmane Mira, Monsieur **BEDAUCHE** Belal au niveau de l'entreprise d'accueil EPB pour leurs encadrements qui nous a été d'une grande utilité et leurs présences à tout moment de la réalisation de ce modeste projet, sans lesquels il n'aurait jamais été accompli , ainsi que les membres du jury : Madame **ALOU** Souraya et Madame **MAAMERI** Karima pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nous remercions également tous les professeurs qui ont contribué de près ou de loin à notre formation universitaire, qui nous a aidé à mener à terme notre projet.

Dédicace

C'est avec un immense plaisir que je dédie ce travail

*À mes chers parents en témoignage et en gratitude de leurs dévouements,
de leur soutien permanent durant*

Toutes mes années d'études (que dieu les gardes).

À mes grands-parents pour leurs présences, soutiens et leurs tendresses.

À mes frères Djamel, Azzedine, Djahid et Toufik

À mes belles-sœurs Nawel et amina.

À mes chers neveux Céline et Anes.

À mes cousines Ouahiba, Lynda, Melissa et Yasmine.

À mes amies Asma, Tina, Djouhra, Souhila, Nina, Sabrina et Saliha

A ma binôme : Djamila

H.Lamia

Dédicace

*Je dédie ce modeste travail aux personnes chères à mon cœur.
À mes parents, qui ont su m'inculquer une éducation de sagesse et de
claire voyance.*

À ma chère sœur Kahina.

À mes frères Alyazid, Ahmed, Ferhat, Salem.

A mes chers Neveux Adem, éline, Elyas,

*À tous mes amis qui m'ont soutenu tout où cours de
réalisation de ce travail.*

À ma binôme Lamia.

B. Djamila

TABLE DES MATIERES

REMERCIEMENTS.....	I
DEDICACES.....	II
TABLE DES MATIERES.....	IV
LISTE DES FIGURES	VII
LISTE DES TABLES.....	VIII
LISTE DES ABREVIATIONS	IV
INTRODUCTION GENERALE.....	1
CHAPITRE I : PRESENTATION DE L'ORGANISME D'ACCUEIL ET METHODOLOGIE DE DEVELOPPEMENT	
.1 Introduction.....	3
I.2 Présentation de l'organisme d'accueil.....	3
I.2.1 Historique de création de l'EPB.....	4
I.2.2 Position géographique de l'EPB.....	5
I.2.3 Missions de l'EPB.....	6
I.2.4 Activités de l'EPB.....	6
I.2.5 Structures de l'EPB.....	7
I.3 Etude de l'existant.....	9
I.4 Problématique	9
I.5 Objectifs et solutions.....	10
I.6 Recueil des besoins	10
I.6.1 Les Besoins fonctionnels	10
I.6.2 Besoins non fonctionnels.....	11
I.7 Démarche de développement.....	11
I.7.1 Processus unifié (Unified Process)	12
I.7.2 XP (eXtreme Programming)	13
I.7.3 UML (Unified Modeling Language).....	14
I.8 Conclusion.....	14

CHAPITRE II : ANALYSE DES BESIONS

II.1	Introduction.....	15
II.2	Identification des acteurs.....	15
II.3	Diagramme de contexte.....	17
II.3.1	Identification des messages échangés.....	18
II.4	Cas d'utilisation.....	20
II.4.1	Identification des cas d'utilisation.....	20
II.4.2	Description textuelle des cas d'utilisation.....	23
II.4.3	Diagramme de cas d'utilisation	28
II.4.3.1	Diagramme de cas d'utilisation pour l'acteur «Client»	28
II.4.3.2	Diagramme de cas d'utilisation pour l'acteur «responsable»	30
II.4.3.3	Diagramme de cas d'utilisation pour l'acteur «administrateur»	31
II.4.3.4	Diagramme de cas d'utilisation global.....	32
II.5	Diagramme de séquence système.....	33
II.5.1	Diagramme de séquence système cas d'utilisation «s'authentifier».....	33
II.5.2	Diagramme de séquence système cas d'utilisation «Rechercher»	35
II.5.3	Diagramme de séquence système cas d'utilisatio«télécharger une facture non payée».....	36
II.5.4	Diagramme de séquence système «imprimer»	37
II.5.5	Diagramme de séquence système «consulter les factures non payées»	38
II.5.6	Diagramme de séquence système cas d'utilisation«consulter les avoirs non payés»	39
II.5.7	Diagramme de séquence système pour cas d'utilisation«lister les comptes clients »	40
II.6	Conclusion.....	43

CHAPITRE III : CONCEPTION ET ELABORATION DU SCHEMA RELATIONNEL

III.2.1	Les Diagrammes d'interaction des cas d'utilisation.....	46
III.2.1.1	Diagramme d'interaction de cas d'utilisation «S'authentifier »	46
III.2.1.2	Diagramme d'interaction de cas d'utilisation «Rechercher»	47
III.2.1.3	Diagramme d'interaction de cas d'utilisation «imprimer»	48
III.2.1.4	Diagramme d'interaction de cas d'utilisation «télécharger».....	49
III.2.1.5	Diagramme d'interaction de cas d'utilisation «consulter les factures non payées»	50
III.2.1.6	Diagramme d'interaction de cas d'utilisation «consulter les avoirs non payés».....	51

III.3 Diagramme de classe.....	52
III.3.1 Description détaillée des attributs de classes	52
III.3.2 Description détaillée des attributs des classes d'associations.....	55
III.3.3 Passage du modèle objet au modèle relationnel.....	58
III.4 Conclusion.....	59

CHAPITRE IV: REALISATION

IV.1 Introduction.....	61
IV.2 Langage et outils de développement.....	61
IV.2.1 Les technologies utilisés.....	62
IV.2.2 Les langages utilisés	64
IV.3 Diagramme déploiement	66
IV.4 Présentation des interfaces.....	67
IV.4.1 Interface « authentification »	67
IV.4.2 Interface « Lister les clients ».....	68
IV.4.2.1 Interface « ajouter un client ».....	68
IV.4.2.2 Interface « Modifier un client ».....	69
IV.4.2.3 Interface « Supprimer un client ».....	70
IV.4.3 Interface « Profil »	71
IV.4.3.1 Interface «Modifier un profil»	72
IV.4.3.2 Interface «Modifier mot de passe »	72
IV.4.4 Interface «consulter les factures non payées »	73
IV.4.5 Interface «afficher la facture non payée».....	74
IV.4.6 Interface «consulter les notifications»	75
IV.5 Conclusion.....	75
CONCLUSION GENERALE	73
BIBLIOGRAPHIE	75

LISTE DES FIGURES

Figure I.1 : La carte géographique de l'EPB.....	5
Figure I.2 : Organigramme général de l'Entreprise Portuaire de Bejaia	8
Figure I.3 : Déroulement du processus Up	13
Figure II.1: Diagramme relation entre les acteurs du systeme	17
Figure II.2: Diagramme de contexte dynamique	18
Figure II.3 : Diagramme de cas d'utilisation pour l'acteur «Client»	29
Figure II.4: Diagramme de cas d'utilisation pour l'acteur «responsable»	30
Figure II.5 : Diagramme de cas d'utilisation pour l'acteur «administrateur»	31
Figure II.6 : Diagramme de cas d'utilisation global.....	32
Figure II.7 : Diagramme de séquence système «s'authentifier»	34
Figure II.8 : Diagramme de séquence système «rechercherr».....	34
Figure II.9 : Diagramme de séquence système «télécharger»	35
Figure II.10 : Diagramme de séquence système «imprimer»	36
Figure II.11 : Diagramme de séquence système «consulter factures non payées»	37
Figure II.12: Diagramme de séquence système «consulter avoirs non payés »	39
Figure II.13: Diagramme de séquence système «Lister les comptes clients»	42
Figure III.1: Diagramme d'interaction de cas d'utilisation «s'authentifier»	44
Figure III.2: Diagramme d'interaction de cas d'utilisation «rechercher».....	45
Figure III.3: Diagramme d'interaction de cas d'utilisation «imprimer».....	48
Figure III.4: Diagramme d'interaction de cas d'utilisation «télécharger».....	49
Figure III.5 : Diagramme d'interaction de cas d'utilisation «consulter les factures non payées»	50
Figure III.6: Diagramme d'interaction de cas d'utilisation «consulter les avoirs non payés».....	51
Figure III.7: Diagramme classe.....	56
Figure IV.1 : Diagramme déploiement.....	66
Figure IV.2 : Interface « authentification »	67
Figure IV.3 : Interface « ajouter un client »	68
Figure IV.4 : Interface «modifier un client »	69
Figure IV.5 : Interface « supprimer un client »	70
Figure IV.6 : Interface « Profil ».....	71
Figure IV.7 : Interface «Modifier un profil».....	72
Figure IV.8 : Interface «Modifier mot de passe».....	72
Figure IV.9 : Interface «consulter les factures non payées »	73
Figure IV.10 : Interface «afficher la facture non payée »	74
Figure IV.11 : Interface «consulter les notifications»	75

LISTE DES TABLES

Table II.1: Messages échangés entre le système et ses acteurs.....	20
Table II.2 : Identification des cas d'utilisation.....	23
Table II.3 : Description du cas d'utilisation « Authentification »	24
Table II.4: Description du cas d'utilisation « Consulter les factures payées »	25
Table II.5 : Description du cas d'utilisation « Consulter les avoirs non payés ».....	26
Table II.6 : Description du cas d'utilisation « Lister les clients ».....	26
Table III.1: Description des classes d'objets et leurs attributs.	54
Table III.2: Description des classes d'associations et leurs attributs.....	56
Table III.3: Équivalences entre les concepts objets et relationnels.	57

LISTE DES ABBREVIATIONS

CSS	Cascade Style Sheet
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol secure
JSON	JavaScript Objet Notation
MIT	Massachusetts Institute of Technology
MYSQL	My Structured Query Language
PDF	Portable Document Format
PDO	PHP DATA OBJECTS
PHP	HyperText Preprocessor
SGBD	Système de Gestion de Base de Données
UML	HyperText Markup Language
UP	Processus Unifié
XP	Extrême Programming
XAMPP	Apache MariaDB Perl Php

INTRODUCTION

GENERALE

L'informatique a pris une importance considérable dans l'entreprise d'aujourd'hui. Partout dans le monde du travail, petites et moyennes entreprises utilisent l'outil informatique pour gérer leur structure et leur business au quotidien. Les écrans d'ordinateur ont remplacé avantageusement les registres d'écriture manuels. Les données sont stockées sur disques durs ou sur serveurs et doivent absolument être protégées contre les attaques extérieures.

L'entreprise portuaire de Bejaia (EPB) fait partie intégrante des entreprises que l'informatique pourra beaucoup aider. En effet, à l'issue de nos observations lors de notre stage qui s'y est bien déroulé, nous avons constaté que certaines directions de l'EPB connaît assez de difficultés lorsque les factures doivent être transférées du port aux clients par mail ou par courrier, cette manière de transfert n'est pas la meilleure pour gérer un nombre aussi important de clients et de factures.

Pour prendre en compte ces difficultés, nous proposons de mettre en place une application web pour la télé-facturation et afin de bien présenter la solution proposée, ce mémoire est organisé en 4 chapitres.

Le premier chapitre, s'intitulé «Présentation de l'organisme d'accueil et méthodologie de Development», permet de présenter l'organisme d'accueil et les différentes directions concernées par la facturation. Par la suite nous avons proposé notre solution qui consiste en la conception et développement d'une application web pour la télé-facturation. Cette application doit répondre a certain nombre de besoins énumérés dans le recueil des besoins.

La phase d'analyse des besoins sera l'objet de le deuxième chapitre qui s'agira d'une phase de modélisation en UML de notre application nous allons présenter les différents acteurs de notre système, leurs rôles, les différentes interactions avec le système ainsi que les besoins qui seront modélisés par un diagramme de cas d'utilisation, et des diagrammes de séquence système.

Le chapitre suivant s'intitule « Conception et élaboration du schéma relationnel. Dans ce chapitre nous allons présenter en détails la conception du projet à travers les diagrammes d'interaction, le diagramme de classes et aussi le modèle relationnel.

Comme dernière étape de notre démarche, la phase de réalisation sera détaillée dans le quatrième chapitre dans lequel nous mettrons en œuvre une description détaillée des outils utilisés pour le développement de l'application web, ainsi que l'architecture du système.

Nous achevons ce mémoire par une conclusion générale qui résumera notre travail avec des perspectives pour les futures améliorations de ce projet.

CHAPITRE I

PRÉSENTATION DE L'ORGANISME D'ACCUEIL & MÉTHODOLOGIE DE DÉVELOPPEMENT

I.1 Introduction

Dans ce chapitre, nous allons présenter l'organisme d'accueil qui est EPB (l'Entreprise Portuaire de Bejaia) son historique, ses missions et activités, puis nous s'intéresserons aux départements qui offrent leurs services aux clients afin d'établir des factures. Par la suite nous allons cerner la problématique et proposer une solution ainsi que des objectifs assignés dans le but d'assurer une meilleure gestion.

I.2 Présentation de l'organisme d'accueil

Afin d'avoir une idée générale de l'organisme d'accueil, nous décrirons brièvement son évolution à travers le temps, sa localisation géographique, ses missions ainsi que ses activités. [1]

I.2.1 Historique de création de l'EPB

Le décret n°82-285 du 14 Août 1982 publié dans le journal officiel n° 33 porta création de l'Entreprise Portuaire de Bejaia ; entreprise socialiste à caractère économique ; conformément aux principes de la charte de l'organisation des entreprises, aux dispositions de l'ordonnance n° 71-74 du 16 Novembre 1971 relative à la gestion socialiste des entreprises et les textes pris pour son application à l'endroit des ports maritimes.

L'entreprise, réputée commerçante dans ses relations avec les tiers, fut régie par la législation en vigueur et soumise aux règles édictées par le susmentionné décret.

Pour accomplir ses missions, l'entreprise est substituée à l'Office National des Ports (ONP), à la Société Nationale de Manutention (SO.NA.MA) et pour partie à la Compagnie Nationale Algérienne de Navigation (CNAN).

Elle fut dotée par l'Etat, du patrimoine, des activités, des structures et des moyens détenus par l'ONP, la SO.NA.MA et de l'activité Remorquage, précédemment dévolue à la CNAN, ainsi que des personnels liés à la gestion et au fonctionnement de celles-ci.

En exécution des lois n° 88.01, 88.03 et 88.04 du 02 Janvier 1988 s'inscrivant dans le cadre des réformes économiques et portant sur l'autonomie des entreprises, et suivant les prescriptions des décrets n°88.101 du 16 Mai 1988, n°88.199 du 21 Juin 1988 et n°88.177 du 28 Septembre 1988.

L'Entreprise Portuaire de Bejaia ; entreprise socialiste ; est transformée en Entreprise Publique Economique, Société par Actions (EPE-SPA) depuis le 15 Février 1989, son capital social fut fixé à Dix millions (10.000.000) de dinars algériens par décision du conseil de la planification n°191/SP/DP du 09 Novembre 1988. Actuellement, le capital social de l'entreprise a été ramené à 1.700.000.000 Da, détenues à 100% par la Société de Gestion des Participations de l'Etat «Ports », par abréviation « SOGEPORTS ».

I.2.2 Position géographique de l'EPB

Le port de Bejaia joue un rôle très important dans les transactions internationales vu sa place et sa position géographique. Il est délimité par :

- Au nord par la route nationale N°9.
- Au sud par les jetées de fermeture et du large sur une largeur de 2 750m.
- A l'est par la jetée Est.
- A l'ouest par la zone industrielle de Bejaia.



Figure I.1 : La carte géographique de l'EPB

I.2.3 Missions de l'EPB

L'entreprise Portuaire de Bejaia a pour missions :

- La gestion, l'exploitation et le développement du domaine portuaire sont les charges essentielles de la gestion de l'EPB, c'est dans le but de promouvoir les échanges extérieurs du pays. Elle se doit d'assumer la police et la sécurité au sein du pays.
- Elle est chargée des travaux d'entretien, d'aménagement, de renouvellement et de création d'infrastructures
- L'EPB assure également des prestations à caractère commercial,

à savoir ; le Remorquage, la manutention et l'acconage.

I.2.4 Activités de l'EPB

Les principales activités de l'entreprise sont :

- L'exploitation de l'outillage et des installations portuaires.
- L'exécution des travaux d'entretien, d'aménagement et de renouvellement de la super structure portuaire.
- L'exercice du monopole des opérations d'acconage et de manutention portuaire.
- L'exercice du monopole des opérations de remorquage, de pilotage et d'amarrage.
- La police et la sécurité portuaire dans la limite géographique du domaine public portuaire.

I.2.5 Structures de l'EPB

L'EPB est organisée selon des directions opérationnelles et fonctionnelles :

- **Directions opérationnelles**

Il s'agit des structures qui prennent en charge les activités sur le terrain et qui ont une relation directe avec les clients.

- **Directions Fonctionnelles**

Il s'agit des structures de soutien aux structures opérationnelles.

CHAPITRE I : PRÉSENTATION DE L'ORGANISME D'ACCUEIL & MÉTHODOLOGIE DE DÉVELOPPEMENT

La Figure représente l'organigramme récapitulant l'ensemble des directions de l'EPB

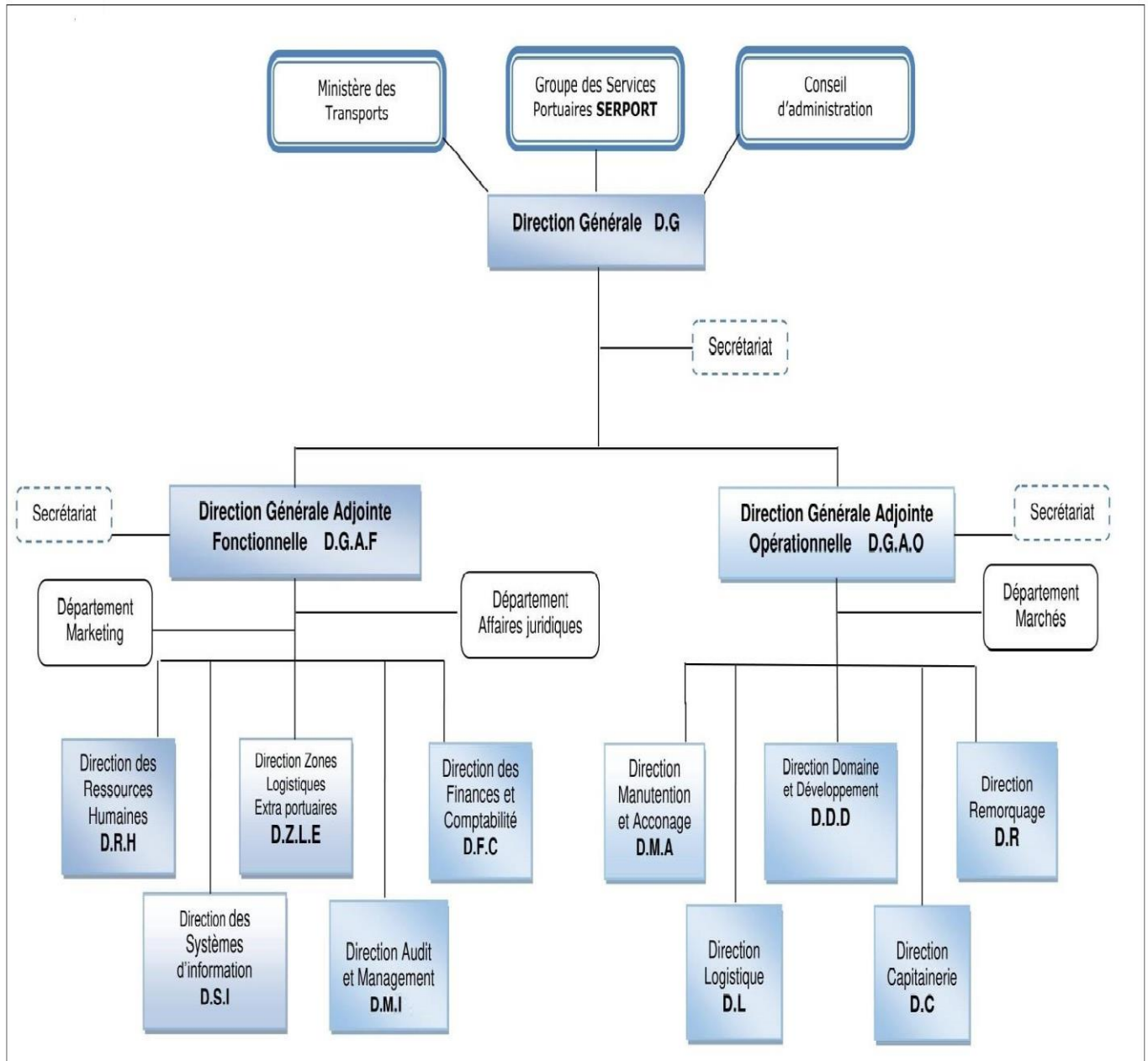


Figure I.2 : Organigramme général de l'Entreprise Portuaire de Bejaia [1]

I.3 Etude de l'existant

78 La facturation est une tâche importante au sein de plusieurs directions de l'EPB comme DDD (la Direction Domaine et Développement), DMA (Direction Manutention et Acconage), DZLE (Direction Zones Logistiques Extra portuaires), DR (la Direction Remorquage) et DC (la Direction Capitainerie). Ces derniers besoin d'émettre des factures pour faciliter leur comptabilité et rester en norme vis-à-vis des lois et professionnel vis-à-vis de leurs clients. Pour transmette ces factures aux clients, ils ont opté pour un outil traditionnel de facturation, comme : la facturation électronique par mail ou bien le client doit se présenter au sien de l'entreprise pour qu'il puisse récupérer ses factures.

Ces différentes solutions malgré leurs avantages évidents vous exposent à plusieurs inconvénients qui risquent de nuire à la gestion de business de l'entreprise et son évolution sur le court terme et le long terme.

I.4 Problématique

Le problème se pose lorsque les factures doivent être transférées du port aux clients par mail ou par courrier, cette manière de transfert n'est pas la meilleure pour gérer un nombre aussi important de clients et de factures

En outre, d'autre éléments viennent se greffer à cette situation telle que:

- absence système informatique pour transférer les factures.
- nombre des factures augmente dans les archives.
- perte de temps lors de transformation des factures.

I.5 Objectifs et solutions

Le but de notre projet est la réalisation d'une application web pour la téléfacturation. Cette solution va permettre de remédier les insuffisances citées ci-dessus et d'accomplir certains objectifs qui seront utiles au EPB et ses clients. Parmi ceux-ci, on trouve :

- Réduire le temps de transmission des factures et permettre de les bien suivre.
- diminuer le cout d'émissions d'une facture.
- garantie la sécurité des données transmises.
- Gagner en rapidité et améliorer la productivité.

I.6 Recueil des besoins

Dans cette section, nous présenterons les différents besoins que l'application doit répondre. Nous avons deux types de besoins, besoins fonctionnels et besoins non-fonctionnels.

I.6.1 Les Besoins fonctionnels

Les besoins fonctionnels se rapportent aux fonctionnalités que l'application doit offrir pour satisfaire les utilisateurs.

Les fonctionnalités que doit intégrer l'application à développer sont :

- Un espace qui permettra aux clients de consulter leurs factures/avoirs.
- Un espace qui permettra aux clients de rechercher une facture/avoir par N°escale, par N°navire ou par date.
- Espace Responsable qui permettra de créer et gérer les comptes clients.
- Espace Responsable qui permettra de lister les factures/ avoirs des clients.
- Espace administrateur qui permettra de lister les comptes des responsables.

I.6.2 Besoins non fonctionnels

Après avoir déterminé les besoins fonctionnels, nous présentons ci-dessous l'ensemble des contraintes à respecter pour garantir la performance du système tout en respectent les exigences des utilisateurs.

- **L'extensibilité:** Dans le cadre de ce travail, l'application devra être extensible, c'est-à dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- **Performance:** Notre application doit assurer un temps de réponse minimum tout en répondant aux besoins du manipulateur
- **L'ergonomie:** l'application offrira une interface conviviale c'est-à-dire simples et facile à utiliser.
- **La modularité de l'application:** Avoir un code simple facile à maintenir et à comprendre en cas de besoin.
- **La sécurité** L'application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde, c'est-à-dire que l'application est accessible par un identifiant et un mot de passe attribué à une personne physique.

I.7 Démarche de développement

Afin de réaliser notre application, nous avons choisi le Processus Unifié (UP) combiné avec la méthode agile Extrême Programming (XP) qui définit un certain nombre de bonnes pratiques permettant de développer un logiciel dans des conditions optimales en plaçant le client au cœur du processus de développement. En effet, la méthode du processus unifié s'appuie principalement sur la modélisation et les diagrammes UML d'où notre choix pour ces derniers.

I.7.1 Processus unifié (Unified Process)

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

- Processus guidé par les cas d'utilisation

Le système à construire se définit d'abord avec les utilisateurs. Les cas d'utilisation permettent d'exprimer les interactions du système avec ces derniers, donc de capturer les besoins.

- Processus centré sur l'architecture

Dès le début des travaux d'analyse et de conception, les auteurs d'UP mettent en avant la préoccupation de l'architecture du système. Il est important de définir le plus tôt possible, l'architecture type qui sera retenue pour le développement, l'implémentation et ensuite le déploiement du système. Le vecteur des cas d'utilisation peut aussi être utilisé pour la description de l'architecture.

- Processus orienté par la réduction des risques

Les risques sont présents à tous les stades de développement d'un système. Il est important de bien évaluer les risques des développements afin d'aider à la bonne prise de décision. Le fait que le processus UP est itératif, cela contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives.

- Processus itératif et incrémental

Ce type de démarche étant relativement connu dans l'approche objet, il paraît naturel qu'UP préconise l'utilisation du principe de développement par itérations successives. Le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement globale. A la fin de chaque itération une partie exécutable du système finale est produite, de façon incrémentale.[1]

La Figure qui suit illustre le déroulement du processus Up.

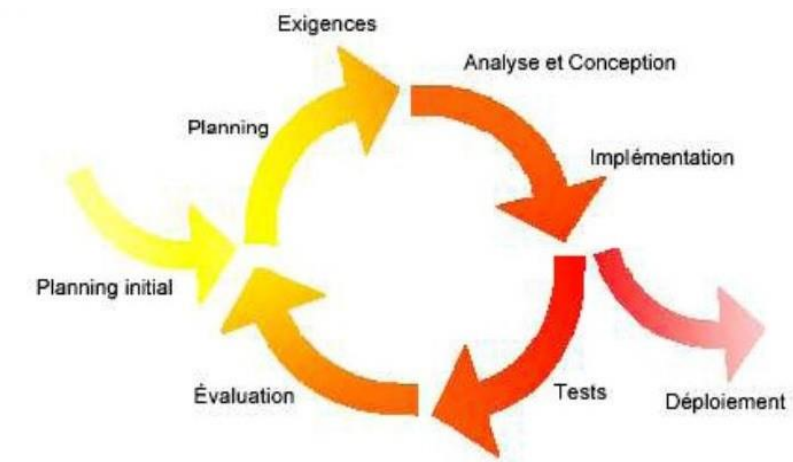


Figure I.3 : Déroulement du processus Up [2].

I.7.2 XP (eXtreme Programming)

La méthode XP (pour *eXtreme Programming*) définit un certain nombre de bonnes pratiques permettant de développer un logiciel dans des conditions optimales en plaçant le client au cœur du processus de développement.

L'eXtreme Programming est notamment basé sur les concepts suivants :

- Les équipes de développement travaille directement avec le client sur des cycles très courts d'une à deux semaines maximum.

- Les livraisons de versions du logiciel interviennent très tôt et à une fréquence élevée pour maximiser l'impact des retours utilisateurs.
- L'équipe de développement travaille en collaboration totale sur la base de binômes.
- Le code est testé et nettoyé tout au long du processus de développement.
- Des indicateurs permettent de mesurer l'avancement du projet afin de permettre de mettre à jour le plan de développement. [3]

I.7.3 UML (Unified Modeling Language)

Le Langage de Modélisation Unifié, se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. [4]

I.8 Conclusion

Dans ce chapitre nous avons présenté l'entreprise portuaire de Bejaia (EPB), puis on s'est focalisé sur les directions qui ont besoin d'émettre leurs factures aux clients. Suite à l'étude de l'existant nous avons pu cerner la problématique et proposer une solution ainsi que des objectifs à atteindre dans le but d'assurer une meilleure gestion.

CHAPITRE II

ANALYSE DES BESOINS

II.1 Introduction

Dans ce chapitre nous allons présenter les différents acteurs de notre système, leurs rôles, les différentes interactions avec le système ainsi que les besoins qui seront modélisés par un diagramme de cas d'utilisation, la description textuelle de ces cas d'utilisation et des diagrammes de séquence système.

II.2 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. D'autre part, le même acteur peut jouer alternativement plusieurs rôles et le même rôle peut être joué par plusieurs acteurs. Dans notre cas, nous avons identifié trois acteurs :

Client: cet acteur pourra s'authentifier et effectuer les tâches suivantes :

- Consulter les notifications.
- Consulter les factures payées.
- Consulter les factures non payées.
- Consulter les avoirs payés
- Consulter les avoirs non payés
- Gérer ses paramètres.

Responsable : cet acteur pourra s'authentifier et effectuer les tâches suivantes :

- Lister les comptes clients.
- Lister les factures payées et non payées.
- Lister les avoirs payées et non payées.
- Gérer ses paramètres.

Administrateur : cet acteur pourra s'authentifier et effectuer la tâche suivante :

- Lister les comptes des responsables.

En plus de ces trois acteurs, nous ajoutons l'acteur générique « Utilisateur »] qui représente toute entité non encore identifiée par le système. Ainsi pour cet acteur, la seule fonctionnalité accessible est l'authentification. Le diagramme suivant illustre la relation entre les trois acteurs :

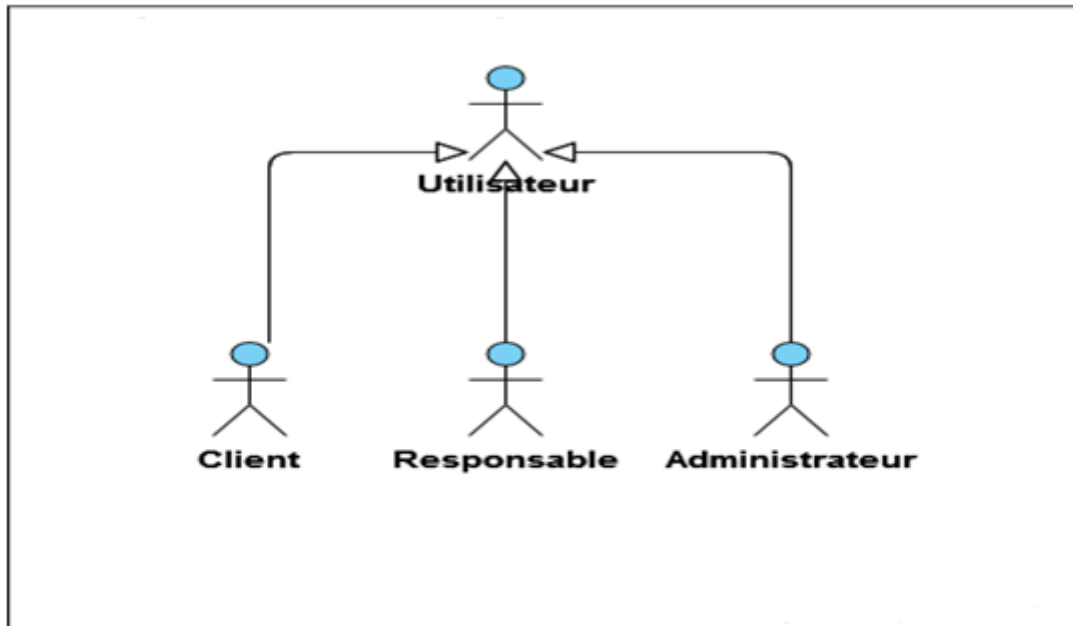


Figure II.1 : Relation entre quelques acteurs du système.

II.3 Diagramme de contexte

Le diagramme de contexte dynamique permet de positionner le système dans son environnement selon le point de vue des communications. Il précise les échanges d'informations qui sont réalisées entre le système et les éléments matériels extérieurs au système. Le système est donc décrit physiquement et logiquement. [5]

La Figure ci-dessous représente le diagramme de contexte qui permet de présenter et de visualiser tous les messages échangés entre les acteurs du système identifiés auparavant.

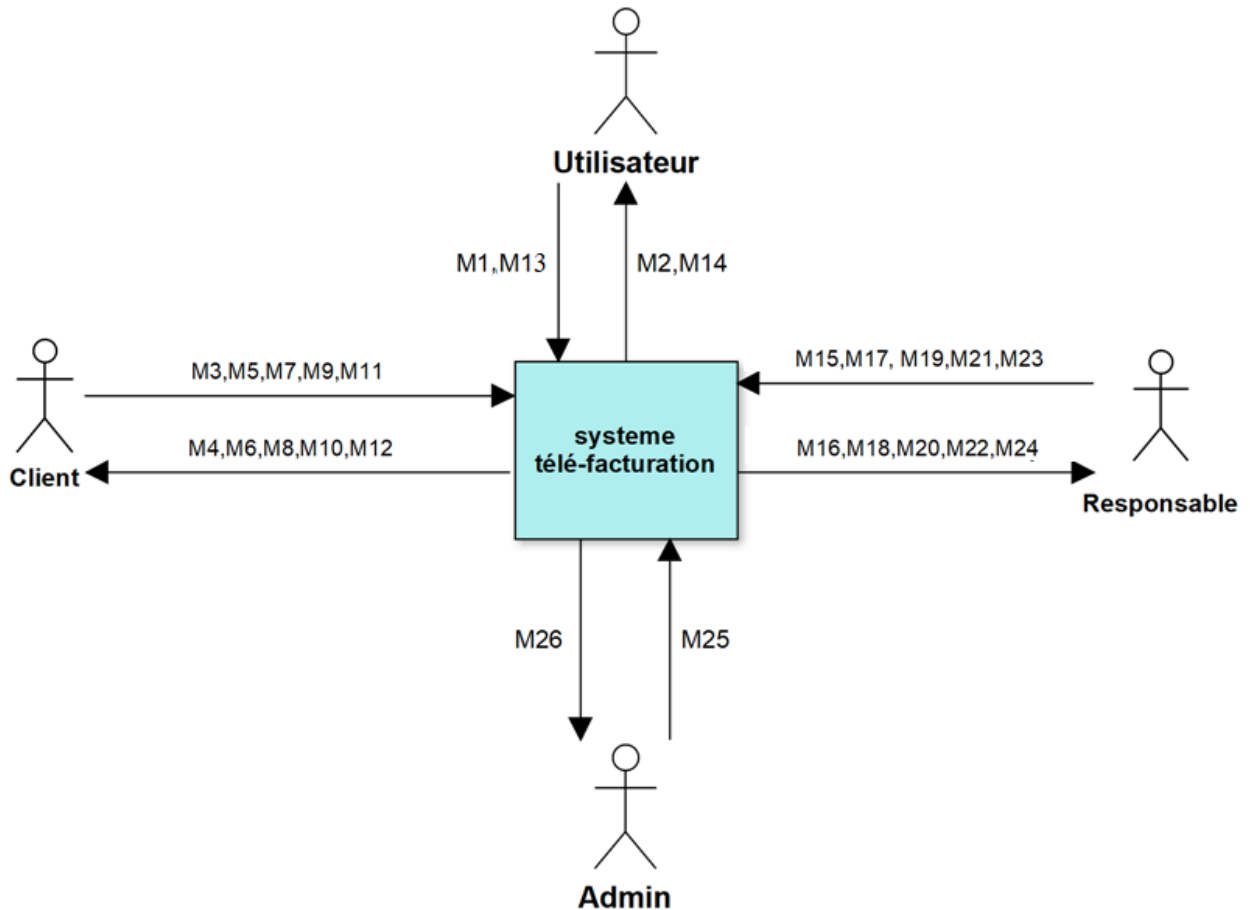


Figure II.2: diagramme de contexte dynamique

II.3.1 Identification des messages échangés

Un message est une information envoyée par un acteur et provoquant le déclenchement de certaines actions associées à ce dernier, en terme général c'est un moyen de communication entre les différents acteurs du système.

Le Tableau ci-dessous représente les différents messages échangés entre les acteurs et le système

Acteur	N°	Message entrant	N°	Message sortant
Utilisateur	M1	Demande l'authentification	M2	Affichage de l'interface d'accueil correspondant à chaque acteur
	M13	Gestion des paramètres	M14	Affichage de l'interface gestion des paramètres
Client	M3	Consultation des notifications	M4	Affichage de l'interface consultation des notifications
	M5	Consultation des factures payées	M6	Affichage de l'interface consultation des factures payées
	M7	Consultation des factures non payées	M8	Affichage de l'interface consultation des factures non payées
	M9	Consultation des avoirs payés	M10	Affichage de l'interface consultation des avoirs payés
	M11	Consultation des avoirs non payés	M12	Affichage de l'interface consultation des avoirs non payés
Responsable	M15	lister des comptes	M16	Affichage de l'interface lister des comptes utilisateurs
	M17	Lister des factures payées	M18	Affichage de l'interface lister les factures payées
	M19	Lister des factures non payées	M20	Affichage de l'interface lister les factures non payées
	M21	Lister des avoirs non payés	M22	Affichage de l'interface lister les avoirs payés
	M23	Lister des avoirs non payés	M24	Affichage de l'interface lister les avoirs non payés

Admin	M25	Lister les comptes responsables	M26	Affichage de l'interface lister les comptes responsables
-------	-----	---------------------------------	-----	--

Table II.1: Messages échangés entre le système et ses acteurs.

II.4 Cas d'utilisation

Un cas d'utilisateur représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. [6]

II.4.1 Identification des cas d'utilisation

Dans ce qui suit, nous allons énumérer les différents cas d'utilisation pour chaque acteur du système. Pour mieux présenter ces cas d'utilisation, nous avons opté pour une structure tabulaire [06]. Le tableau suivant présente les différents cas d'utilisation identifiés pour notre système :

N°	Cas d'utilisation		Acteur	Messages Emis / Reçus par les acteurs
1	<i>S'authentifier</i>		utilisateur	Emet : Information d'identification et le mot de passe. Reçoit : L'interface adéquate à l'utilisateur.
2	<i>Gérer les paramètres</i>	Consulter profil	utilisateur	Emet : Consultation profil, mise à jour informations de profil. Reçoit : affichage du profil, confirmation de la modification du profil.
		MAJ informations		
3	<i>Consulter les notifications</i>		client	Emet : Consultation des notifications des factures/avoirs par leur escale. Reçoit : afficher la facture/avoir de la notification.
4	<i>Consulter les factures payées</i>	Consulter	client	Emet : Consultation, impression, téléchargement, recherche d'une facture payée Reçoit : affichage de la facture payée , confirmation d'impression, de téléchargement et de la recherche
		Imprimer		
		télécharger		
		Rechercher		
5	<i>Consulter les factures non payées</i>	Consulter	Client	Emet : Consultation, impression, téléchargement, recherche d'une facture non payée Reçoit : affichage de la facture non payée , confirmation d'impression, de téléchargement et de la recherche
		Imprimer		
		Télécharger		
		Rechercher		

6	<i>Consulter les avoirs payés</i>	Consulter	Client	<p>Emet : Consultation, impression, téléchargement, recherche d'un avoir payé</p> <p>Reçoit : affichage d'un avoir payé , confirmation d'impression, de téléchargement et de la recherche.</p>
		Imprimer		
		Télécharger		
		Rechercher		
7	<i>Consulter les avoirs non payés</i>	Consulter	Client	<p>Emet : Consultation, impression, téléchargement, recherche d'un avoir non payé.</p> <p>Reçoit : affichage d'un avoir non payé, confirmation d'impression, de téléchargement et de la recherche.</p>
		Imprimer		
		Télécharger		
		Rechercher		
8	<i>Lister comptes clients</i>	Ajouter	Responsable	<p>Emet : ajout, modification et la suppression des comptes clients.</p> <p>Reçoit : confirmation d'ajout, de la modification et de la suppression.</p>
		Modifier		
		Supprimer		
9	<i>Lister les factures payées</i>	Consulter	Responsable	<p>Emet : Consultation, impression, téléchargement, recherche d'une facture payée.</p> <p>Reçoit : affichage de la facture payée , confirmation d'impression, de téléchargement et de la recherche.</p>
		Imprimer		
		Télécharger		
		Rechercher		
10	<i>Lister les factures non payées</i>	Consulter	Responsable	<p>Emet : Consultation, impression, téléchargement, recherche et de valider d'une facture non payée.</p> <p>Reçoit : affichage de la facture payée , confirmation d'impression, de téléchargement, de la recherche et de validation.</p>
		Imprimer		
		Télécharger		
		Valider		
		Rechercher		
11		Consulter		<p>Emet : Consultation, impression, téléchargement, recherche d'un</p>
		Imprimer		

	<i>Lister les avoirs payés</i>	Télécharger Rechercher	Responsable	avoir payé Reçoit : affichage d'un avoir payé , confirmation d'impression, de téléchargement et de la recherche.
12	<i>Lister les avoirs non payés</i>	Télécharger Imprimer Rechercher	Responsable	Emet : Consultation, impression, téléchargement, recherche et de valider d'un avoir non payé. Reçoit : affichage d'un avoir non payé, confirmation d'impression, de téléchargement, de la recherche et de validation.
13	<i>Lister comptes Responsables</i>	Ajouter Modifier Supprimer	Admin	Emet : ajout, modification et la suppression des comptes responsables. Reçoit : confirmation d'ajout, de la modification et de la suppression.

Table II.2 : Identification des cas d'utilisation

II.4.2 Description textuelle des cas d'utilisation

La description d'un cas d'utilisation permet de clarifier le déroulement de la fonctionnalité. Nous allons décrire quelques cas d'utilisation à travers les tableaux qui suivent.

Cas d'utilisation « s'authentifier »	
Sommaire d'Identification	
Titre	S'authentifier
But	Permet d'identifier l'utilisateur et de savoir son rôle dans l'application (privilèges accordés à cet utilisateur).
Résumé	L'utilisateur doit fournir un identifiant et un mot de passe puis valide le formulaire d'authentification. Le système utilise l'identifiant et le mot de passe pour vérifier si ces informations sont correctes. Dans le cas d'erreur, le système affiche un message d'erreur et invite l'utilisateur et refaire l'authentification, sinon (dans le cas de succès), le système affiche l'interface adéquate pour l'utilisateur.
Acteurs	Utilisateur
Description des enchaînements	
Pré-conditions	/
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsqu'un utilisateur veut accéder au système.</p> <p>Enchaînement (a) – S'identifier par login et mot de passe</p> <p>L'utilisateur fournit un identifiant (login) et un mot de passe secret, et envoie les informations au système afin d'accéder à son espace de travail. Le système effectue des vérifications.</p> <ul style="list-style-type: none"> - Si l'utilisateur a omis l'identification ou mot de passe alors il faut exécuter [Exception 01 : champs Vides]. - Si l'identifiant et/ou mot de passe ne sont pas correcte alors il faut exécuter [Exception 02 : champs Incorrect].
Exceptions	
<p>[Exception 01 : champs Vides] :</p> <p>Le système notifie une erreur à l'utilisateur lui indiquant qu'il a oublié un ou plusieurs champs à saisir (identifiant ou mot de passe), et l'invite à compléter les champs manquants.</p>	
<p>[Exception 02 : champs Incorrect] :</p> <p>Le système indique à l'utilisateur qu'une erreur est détectée liée à son identifiant et/ou à son mot de passe, il l'invite à ressaisir son identifiant et/ou son mot de passe.</p>	

Table II.3 : Description du cas d'utilisation « Authentification »

Cas d'utilisation « Consulter les factures payées »	
Sommaire d'Identification	
Titre	Consulter les factures payées
But	Permet au client de consulter ses factures déjà payées
Résumé	<p>Le client accède à l'espace « Consulter les factures payées ». Le système affiche la liste des factures payées correspondantes à ce client. à travers cette interface le client peut effectuer plusieurs actions :</p> <ul style="list-style-type: none"> - Afficher une facture détaillée sous forme PDF. - Télécharger une facture payée. - Rechercher une facture payée.
Acteurs	Client
Description des enchaînements	
Pré-conditions	S'authentifier
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsqu'un client veut accéder à sa liste des factures payées</p> <p>Enchaînement (a) – afficher une facture payée Le client demande d'afficher une facture payée. Le système affiche la facture sous forme d'un PDF d'où le client peut l'imprimer ou la télécharger.</p> <p>Enchaînement (b) –télécharger une facture payée Le client demande de télécharger une facture payée. Le système affiche la fenêtre de téléchargement. Le client confirme le téléchargement et finalement le système télécharge la facture sélectionnée.</p> <p>Enchaînement (c) –recherche une facture payée Le client doit définir le type de recherché (N°escale, N°navire ou par date), Et remplir le champ de recherche et envoie les informations au système afin d'effectuer la recherche. Le système effectue des vérifications.</p> <ul style="list-style-type: none"> - Si le client a omis le champ de recherche alors il faut exécuter [Exception 01 : champs Vides]. - Si le champ de recherche n'existe pas alors il faut exécuter [Exception 02 : aucun résultat].
Exceptions	
[Exception 01 : champs Vides] :	
Le système notifie une erreur au client lui indiquant qu'il a oublié de remplir le champ de recherche et l'invite à remplir le champ.	
[Exception 02 : aucun résultat] :	
Le système indique au client qu'aucun résultat n'est disponible à sa recherche.	

Table II.4: Description du cas d'utilisation «Consulter les factures payées »

Cas d'utilisation « Consulter les avoirs non payés »	
Sommaire d'Identification	
Titre	Consulter les avoirs non payés
But	Permet au client de consulter ses avoirs non payés
Résumé	<p>Le client accède à l'espace « Consulter les avoirs non payés ». Le système affiche la liste des avoirs non payés correspondantes à ce client. à travers cette interface le client peut effectuer plusieurs actions :</p> <ul style="list-style-type: none"> - Afficher un avoir détaillé sous forme PDF. - Télécharger un avoir payé. - Rechercher un avoir payé.
Acteurs	Client
Description des enchaînements	
Pré-conditions	S'authentifier
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsqu'un client veut accéder à sa liste des factures payées</p> <p>Enchaînement (a) – afficher un avoir non payée Le client demande d'afficher un avoir non payé. Le système affiche l'avoir sous forme d'un PDF d'où le client peut l'imprimer ou le télécharger.</p> <p>Enchaînement (b) –télécharger un avoir non payée Le client demande de télécharger un avoir non payée. Le système affiche la fenêtre de téléchargement. Le client confirme le téléchargement et finalement le système télécharge l'avoir sélectionné.</p> <p>Enchaînement (c) –rechercher un avoir non payée Le client doit définir le type de recherché (N°escale, N°navire ou par date), Et remplir le champ de recherche et envoie les informations au système afin d'effectuer la recherche. Le système effectue des vérifications.</p> <ul style="list-style-type: none"> - Si le client a omis le champ de recherche alors il faut exécuter [Exception 01 : champs Vides]. - Si le champ de recherche n'existe pas alors il faut exécuter [Exception 02 : aucun résultat].
Exceptions	
[Exception 01 : champs Vides] : Le système notifie une erreur au client lui indiquant qu'il a oublié de remplir le champ de recherche et l'invite à remplir le champ.	
[Exception 02 : aucun résultat] : Le système indique au client qu'aucun résultat n'est disponible à sa recherche.	

Table II.5 : Description du cas d'utilisation « Consulter les avoirs non payés »

Cas d'utilisation « lister les clients »	
Sommaire d'Identification	
Titre	Lister les clients.
But	Permet au responsable de lister les clients.
Résumé	<p>Le responsable accède à l'espace « lister les clients ». Le système affiche la liste des clients. à travers cette interface le responsable peut effectuer plusieurs actions :</p> <ul style="list-style-type: none"> - Ajouter un client. - Modifier un client. - Supprimer un client. - Rechercher un client.
Acteurs	Responsable
Description des enchaînements	
Pré-conditions	S'authentifier
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsqu'un responsable veut accéder à la liste des clients.</p> <p>Enchaînement (a) – Ajouter un client Le responsable demande d'accéder à l'interface d'ajout. Le système affiche l'interface correspondante. Le responsable remplit le formulaire. Le système ajoute le client.</p> <p>Enchaînement (b) – Modifier un client. Le responsable demande d'accéder à l'interface de modification. Le système affiche l'interface correspondante. Le responsable remplit le formulaire. Le système modifie le client.</p> <p>Enchaînement (b) – supprimer un client. Le responsable demande d'accéder à l'interface de suppression. Le système affiche l'interface correspondante. Le responsable confirme la suppression. Le système supprime le client.</p> <p>Enchaînement (c) –rechercher un client. Le responsable doit définir le type de recherché (code client ou par date), Et remplir le champ de recherche et envoie les informations au système afin d'effectuer la recherche. Le système effectue des vérifications.</p> <ul style="list-style-type: none"> - Si le client a omis le champ de recherche alors il faut exécuter [Exception 01 : champs Vides]. - Si le champ de recherche n'existe pas alors il faut exécuter [Exception 02 : aucun résultat].
Exceptions	
[Exception 01 : champs Vides] :	

Le système notifie une erreur au responsable lui indiquant qu'il a oublié de remplir le champ de recherche et l'invite à remplir le champ.
--

[Exception 02 : aucun résultat] :
--

Le système indique au responsable qu'aucun résultat n'est disponible à sa recherche.
--

Table II. 6 : Description du cas d'utilisation « Consulter les avoirs non payés »

II.4.3 Diagramme de cas d'utilisation

Dans ce qui suit, et après avoir énuméré les différents cas d'utilisation ainsi que leur description textuelle, nous schématisons tous les cas d'utilisation dans un diagramme, connu sous le nom de Diagramme des cas d'utilisation.

II.4.3.1 Diagramme de cas d'utilisation pour l'acteur « Client »

Le diagramme suivant représente les cas d'utilisation associé à un client, ce dernier peut réaliser toutes les tâches représentées dans la Figure ci-dessous

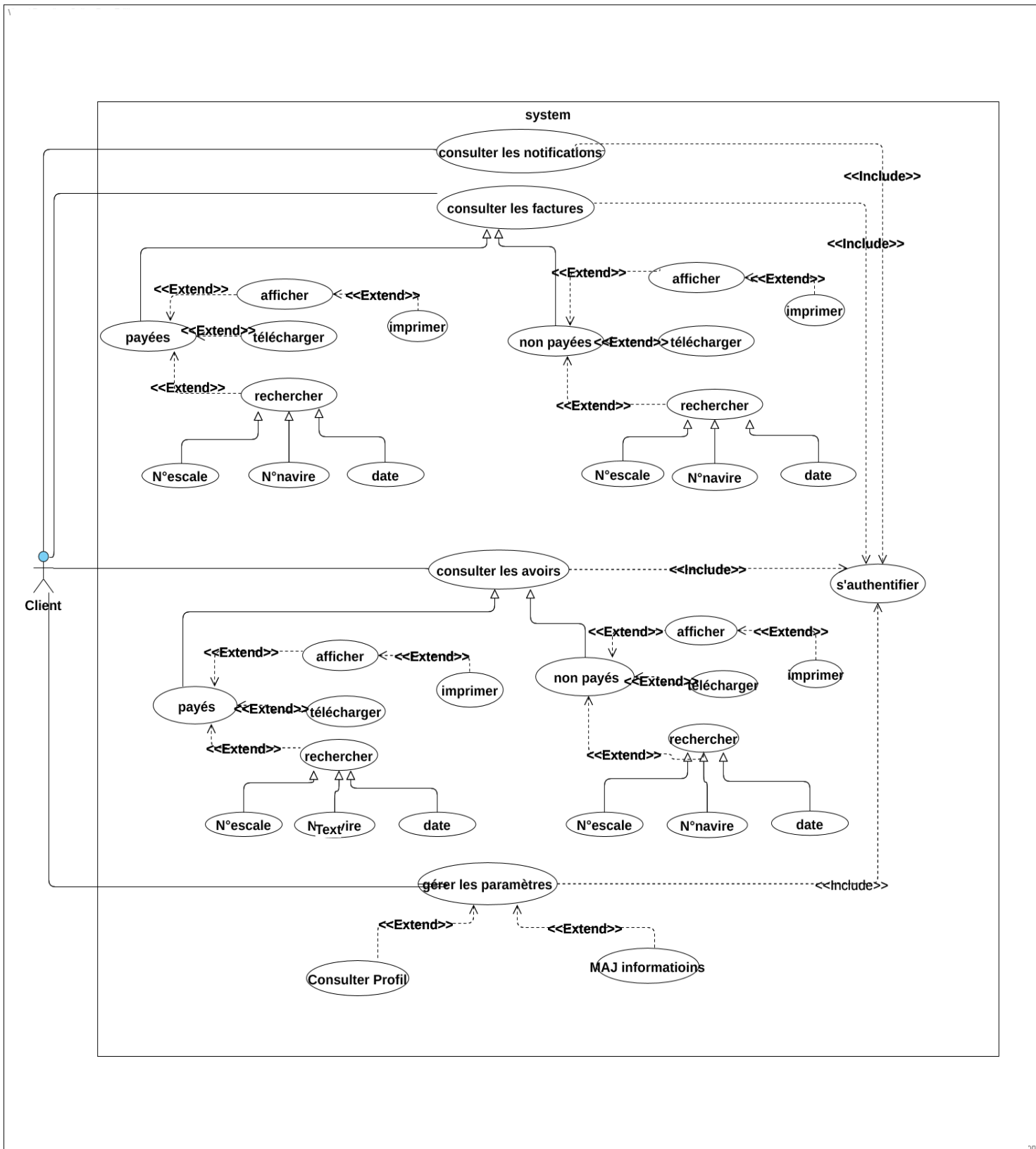


Figure II.3 : Diagramme de cas d'utilisation pour l'acteur «Client»

II.4.3.2 Diagramme de cas d'utilisation pour l'acteur «responsable»

Le diagramme suivant représente les différentes tâches réalisées par un responsable

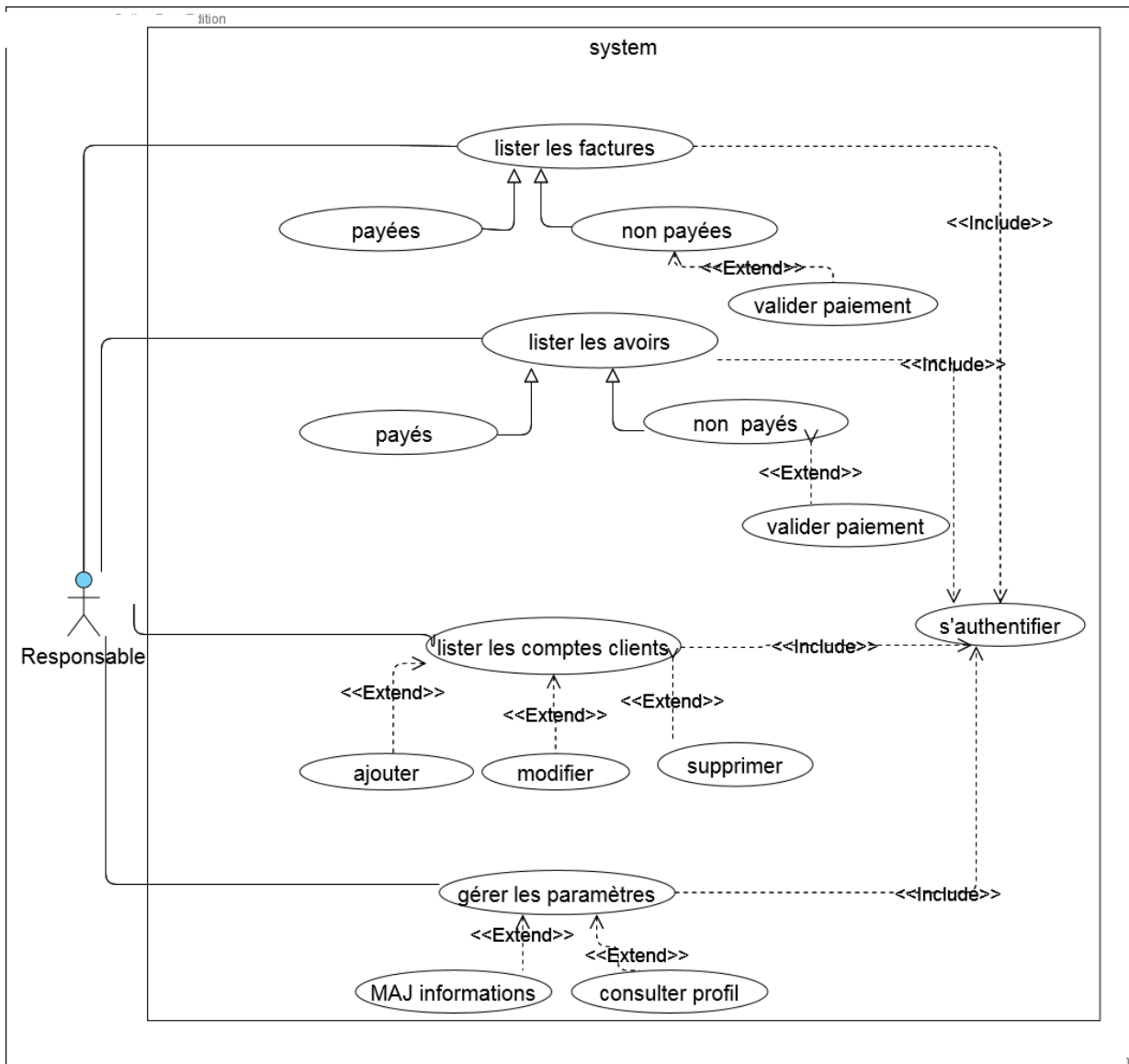


Figure II.4: Diagramme de cas d'utilisation pour l'acteur «responsable»

II.4.3.3 Diagramme de cas d'utilisation pour l'acteur «administrateur»

Le diagramme ci-dessus représente les tâches spécifiées à un administrateur

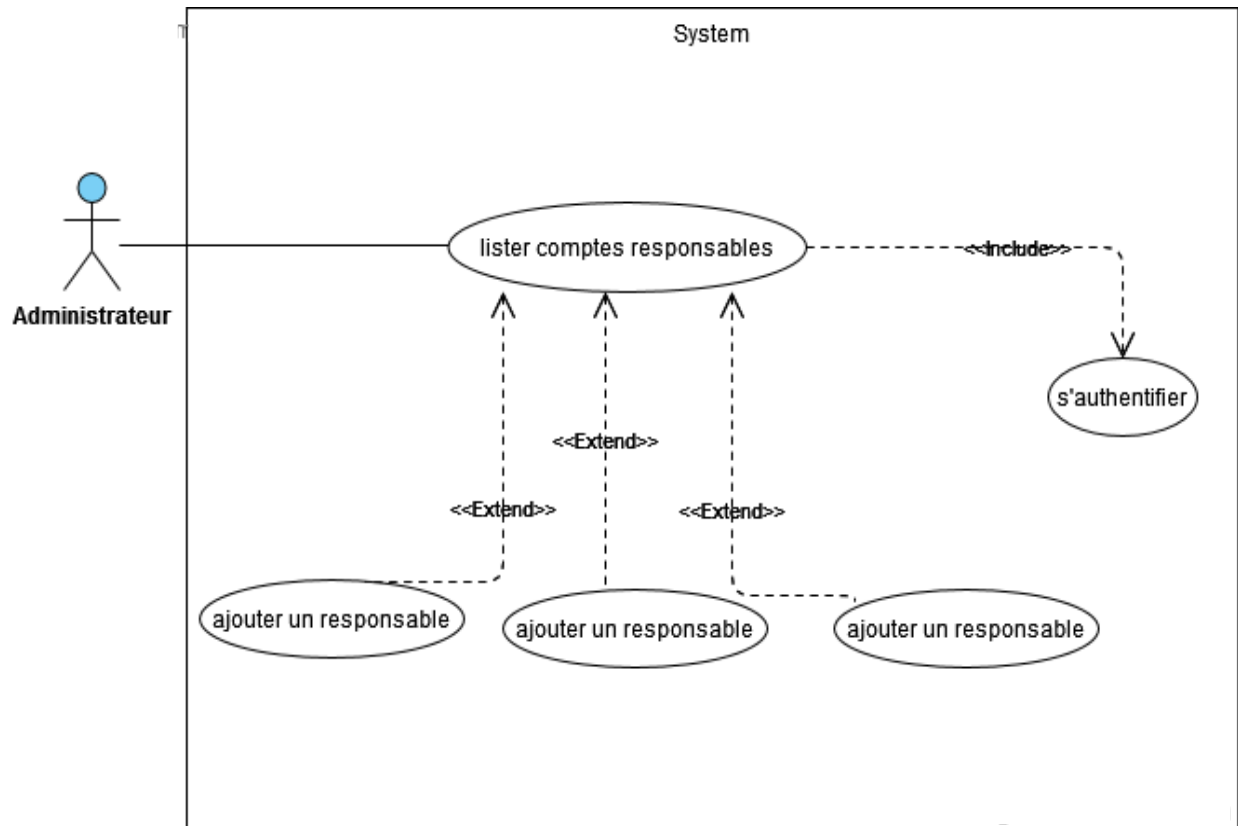


Figure II.5 : Diagramme de cas d'utilisation pour l'acteur «administrateur»

II.4.3.4 Diagramme de cas d'utilisation global

Pour notre système, le diagramme de cas d'utilisation global est représenté dans la figure suivante

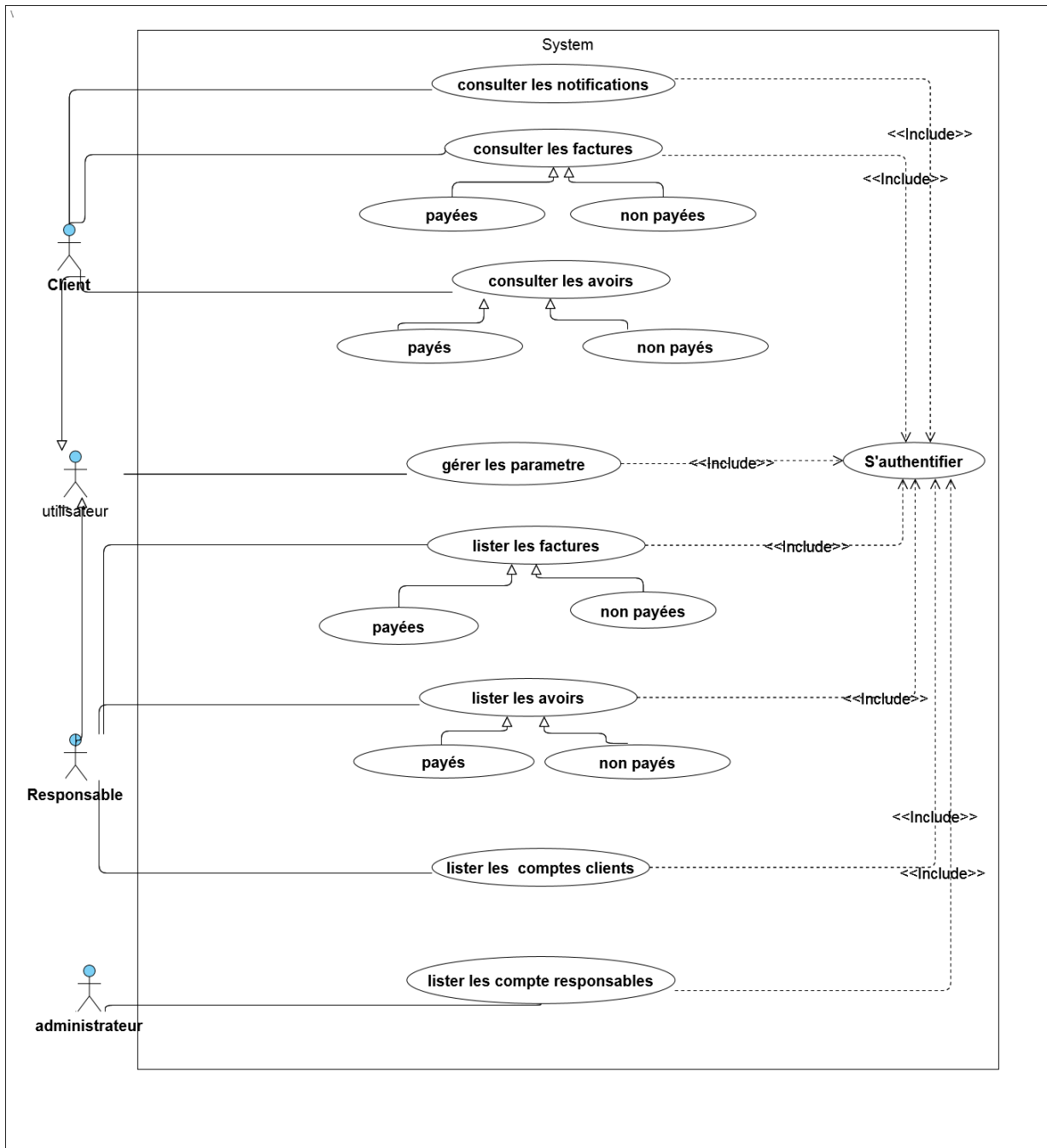


Figure II.6 : Diagramme de cas d'utilisation global

II.5 Diagramme de séquence système

Le diagramme de séquence est un diagramme d'interaction UML. Il représente des échanges de messages entre éléments au cours d'une interaction, Il comprend un groupe d'objets, représentés par des lignes de vie. L'intérêt d'un diagramme de séquence système est de décrire textuellement de façon détaillée chaque cas d'utilisation [5]. Afin de mieux comprendre le fonctionnement de notre système nous allons dans ce qui suit détailler les cas d'utilisation donnés dans le chapitre précédent par des diagrammes de séquences système.

II.5.1 Diagramme de séquence système cas d'utilisation « s'authentifier »

À chaque authentification, l'utilisateur doit saisir son nom utilisateur et mot de passe, d'où l'usage de l'opérateur « Loop ». L'utilisateur valide sa demande de connexion, deux cas peuvent se présenter : si les champs sont vides, le système affiche un message d'erreur entrer login ou mot de passe, si les champs sont bien remplis, le système vérifie l'existence de ce compte, deux autres cas peuvent se présenter : si le compte n'existe pas affichage d'un message d'erreur 'Ce compte n'existe pas' sinon affichage de l'interface correspondante au statut de l'utilisateur.

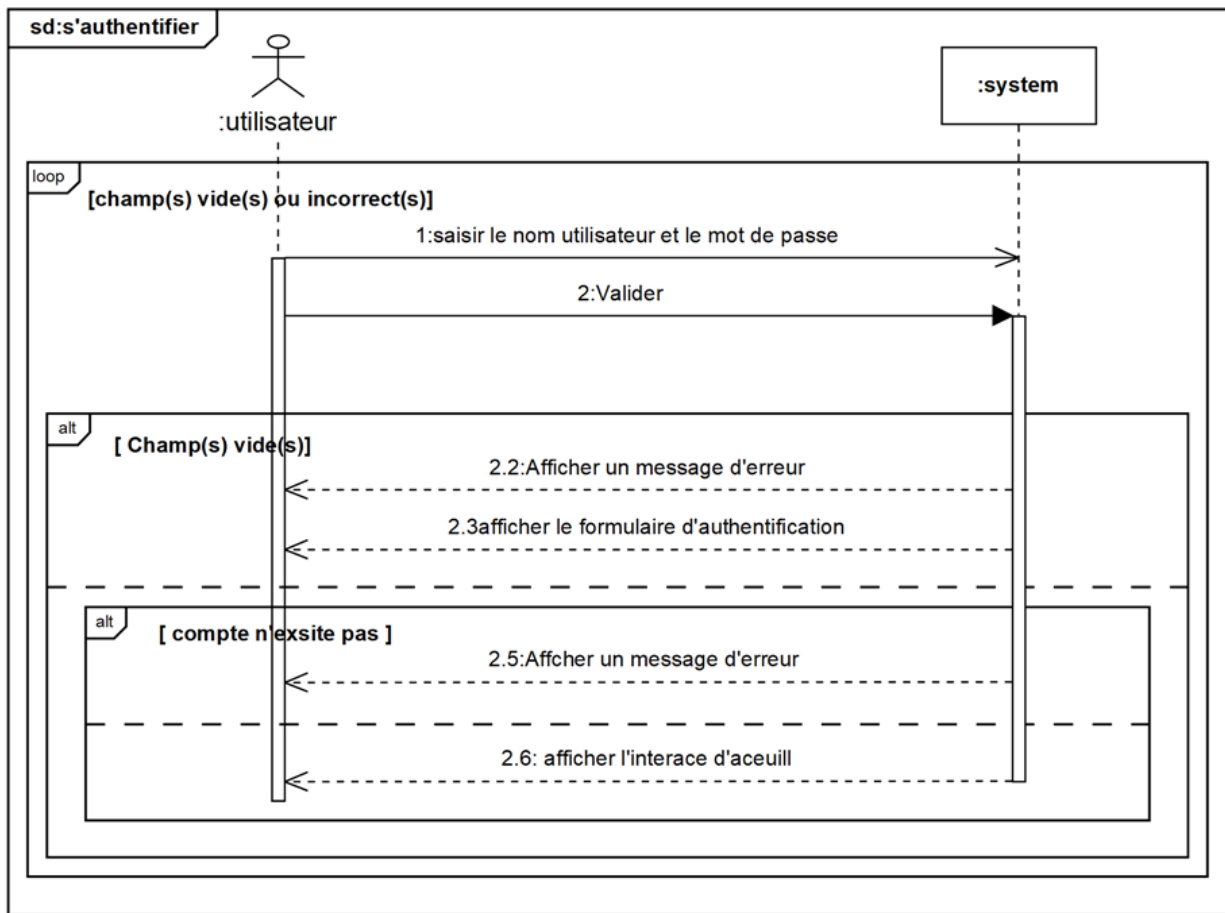


Figure II.7 : Diagramme de séquence système «s’authentifier»

II.5.2 Diagramme de séquence système cas d'utilisation « Rechercher »

Pour faire une recherche d'une facture, l'utilisateur remplir les champs et valide l'opération, le système effectue une vérification si l'un des champs est vide ou incorrecte un message d'erreur est envoyé, si non le système affiche le résultat.

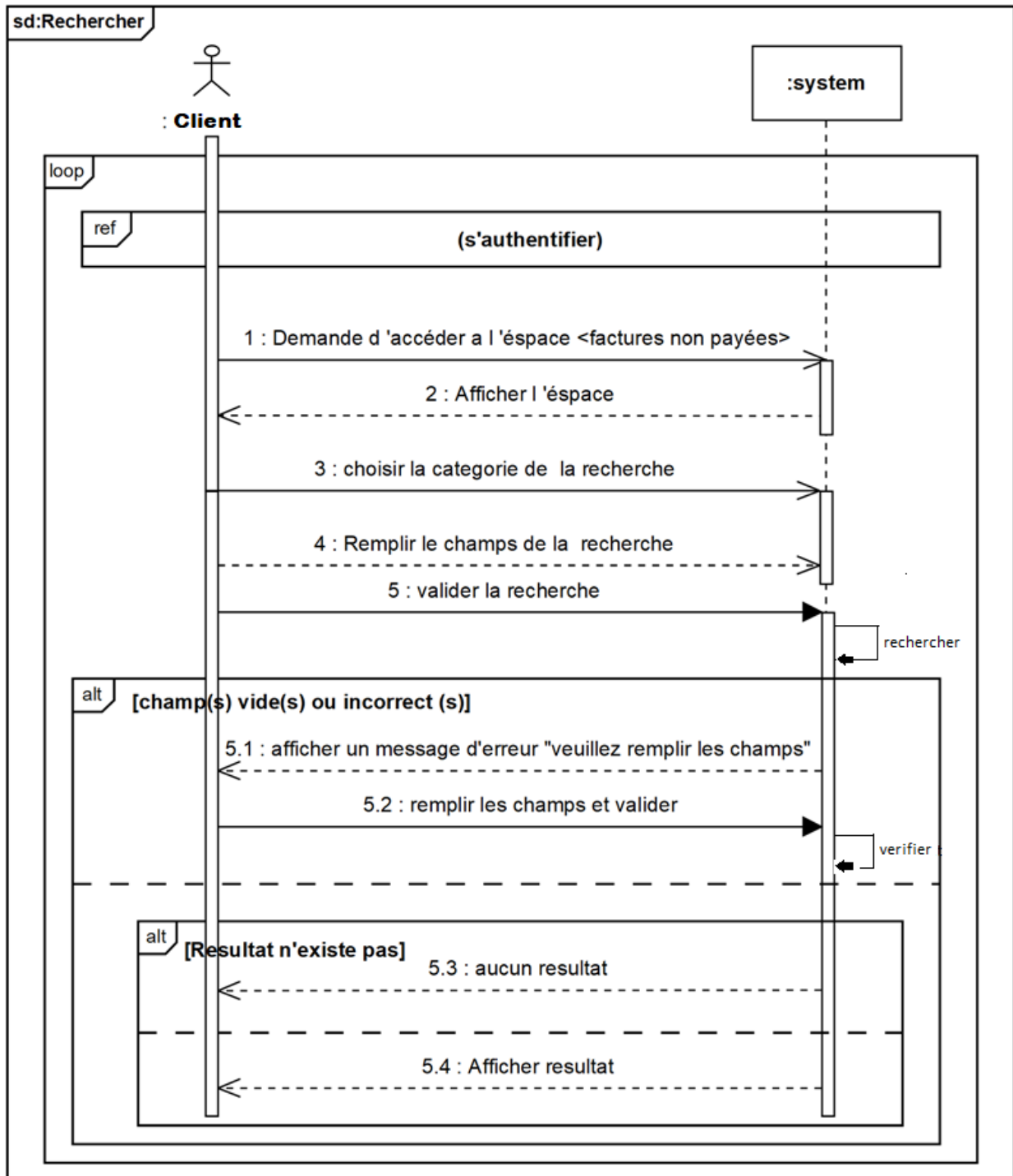


Figure II.8 : Diagramme de séquence système « rechercher »

II.5.3 Diagramme de séquence système cas d'utilisation « télécharger une facture non payée »

Pour faire un téléchargement d'une facture non payée, le client accède d'abord à l'espace « consulter les factures non payées » puis espace téléchargement, il valide l'opération, le système récupère les données et confirme le téléchargement.

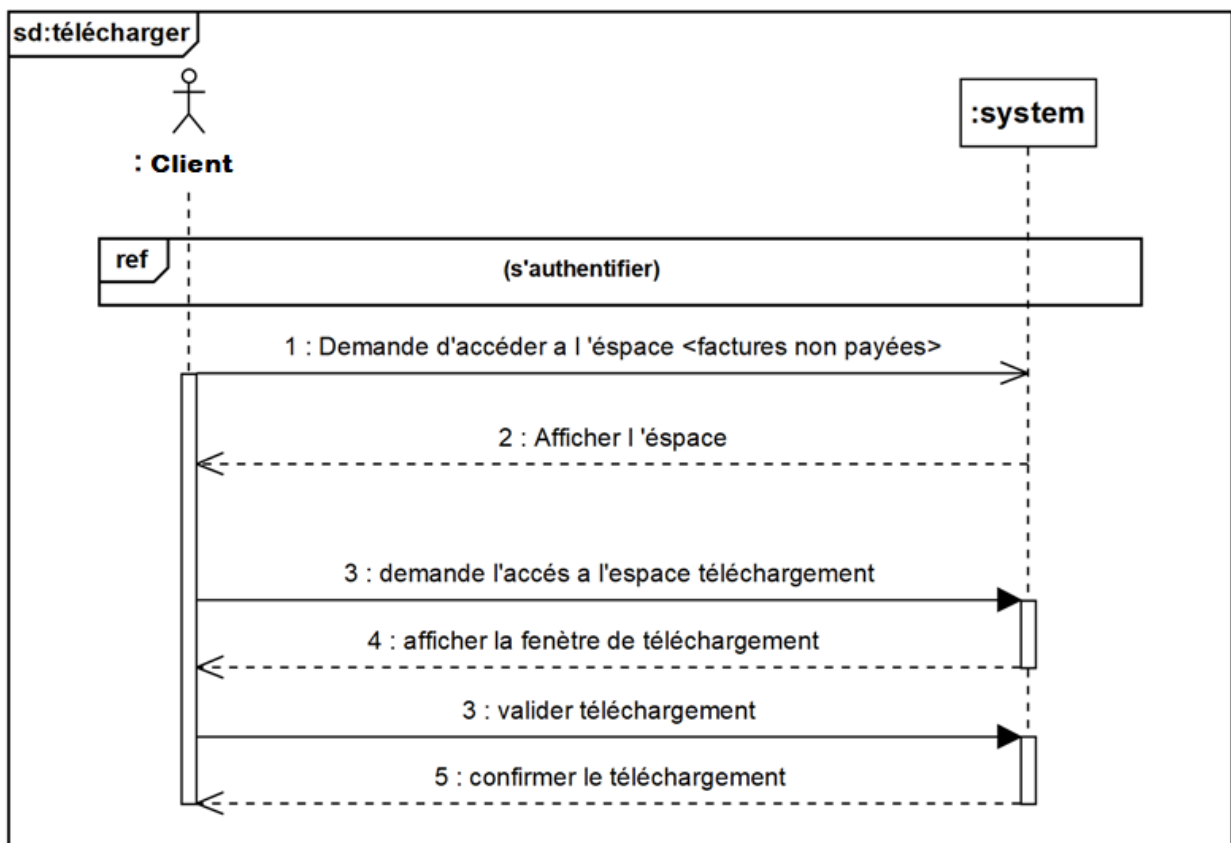


Figure II.9 : Diagramme de séquence système « télécharger »

:

II.5.4 Diagramme de séquence système « imprimer »

Pour faire une impression d'une facture ou d'un avoir, l'utilisateur accède à l'espace d'impression et valide l'opération, le système récupère les données et confirme l'impression.

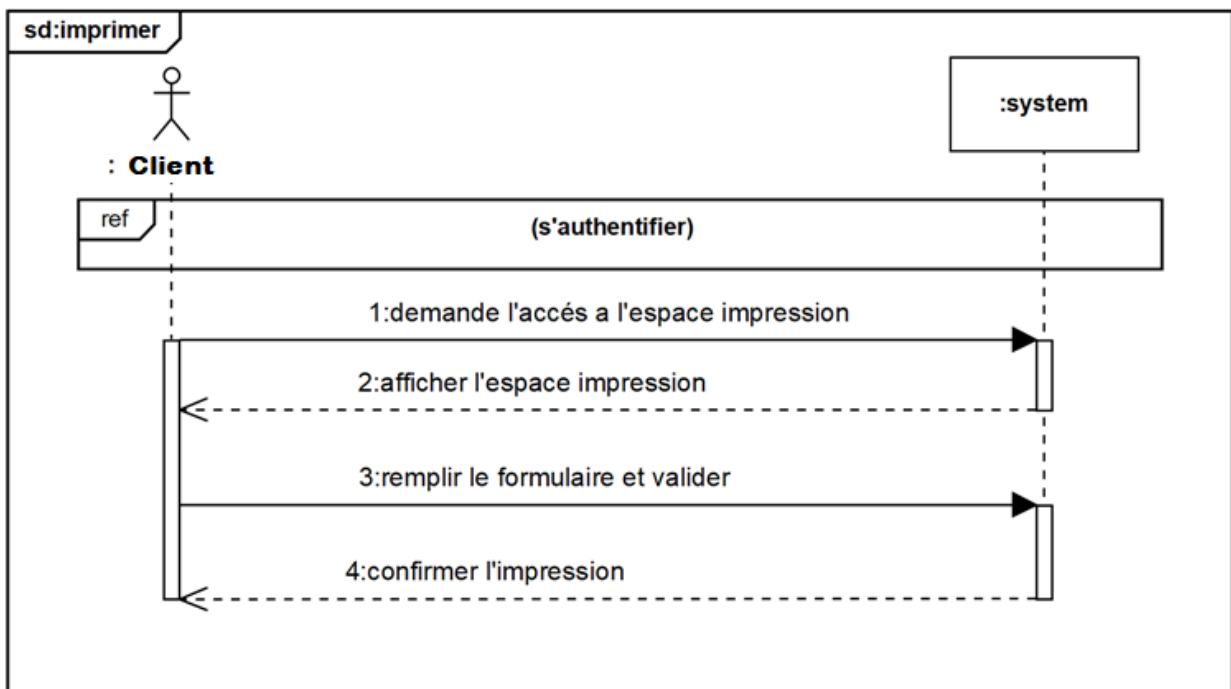


Figure II.10: Diagramme de séquence système « imprimer »

II.5.5 Diagramme de séquence système «consulter les factures non payées»

Pour faire une consultation des factures non payées, le client accède à cette interface pour afficher la liste des factures. Comme il peut via cette interface de rechercher, d'imprimer ou de télécharger une facture non payée.

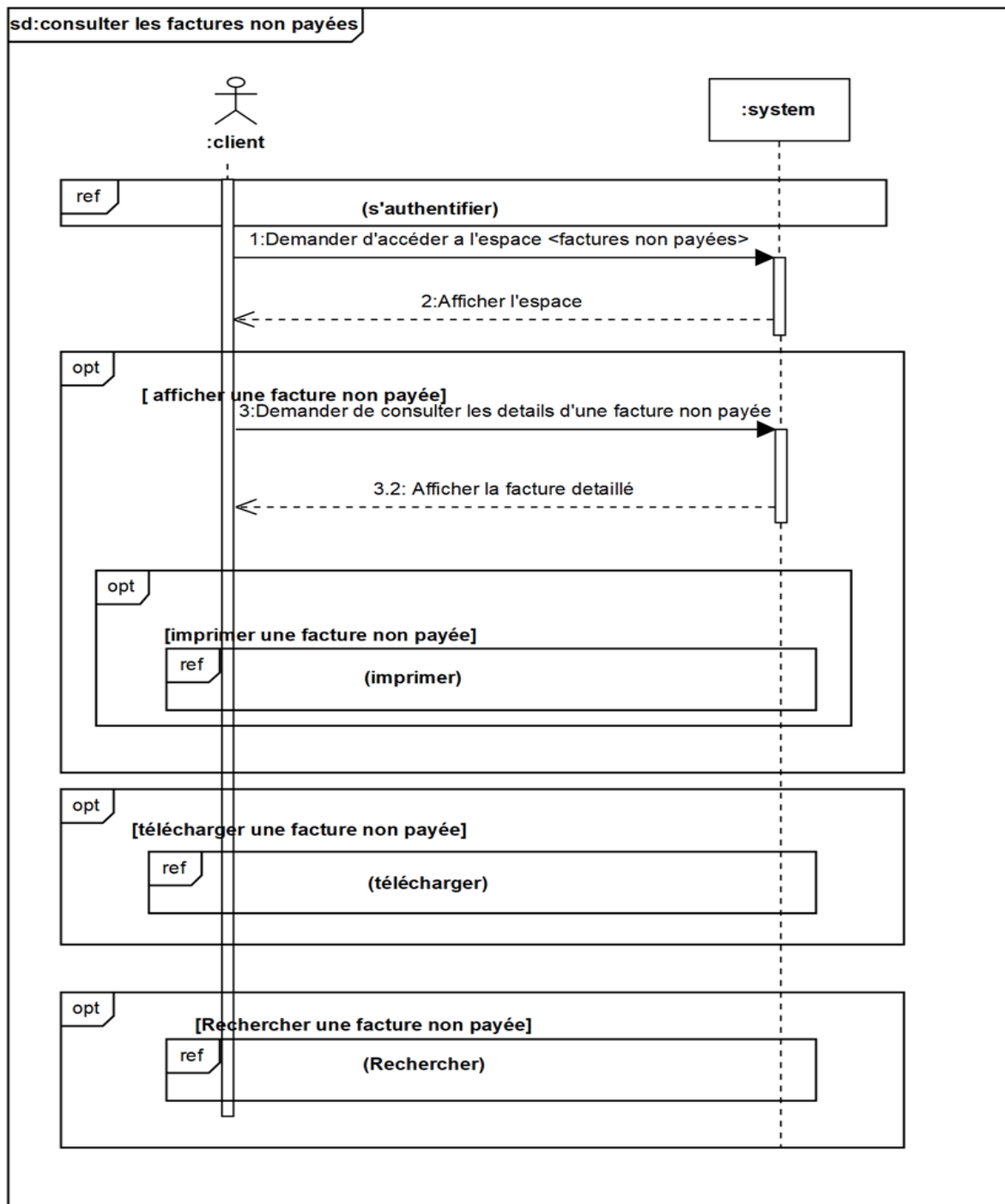


Figure II.11: Diagramme de séquence système «consulter les factures non payées»

II.5.6 Diagramme de séquence système cas d'utilisation «consulter les avoirs non payés»

Pour faire une consultation des avoirs non payés, le client accède à cette interface pour afficher la liste des avoirs non payés. Comme il peut via cette interface de rechercher, d'imprimer ou de télécharger un avoir non payé

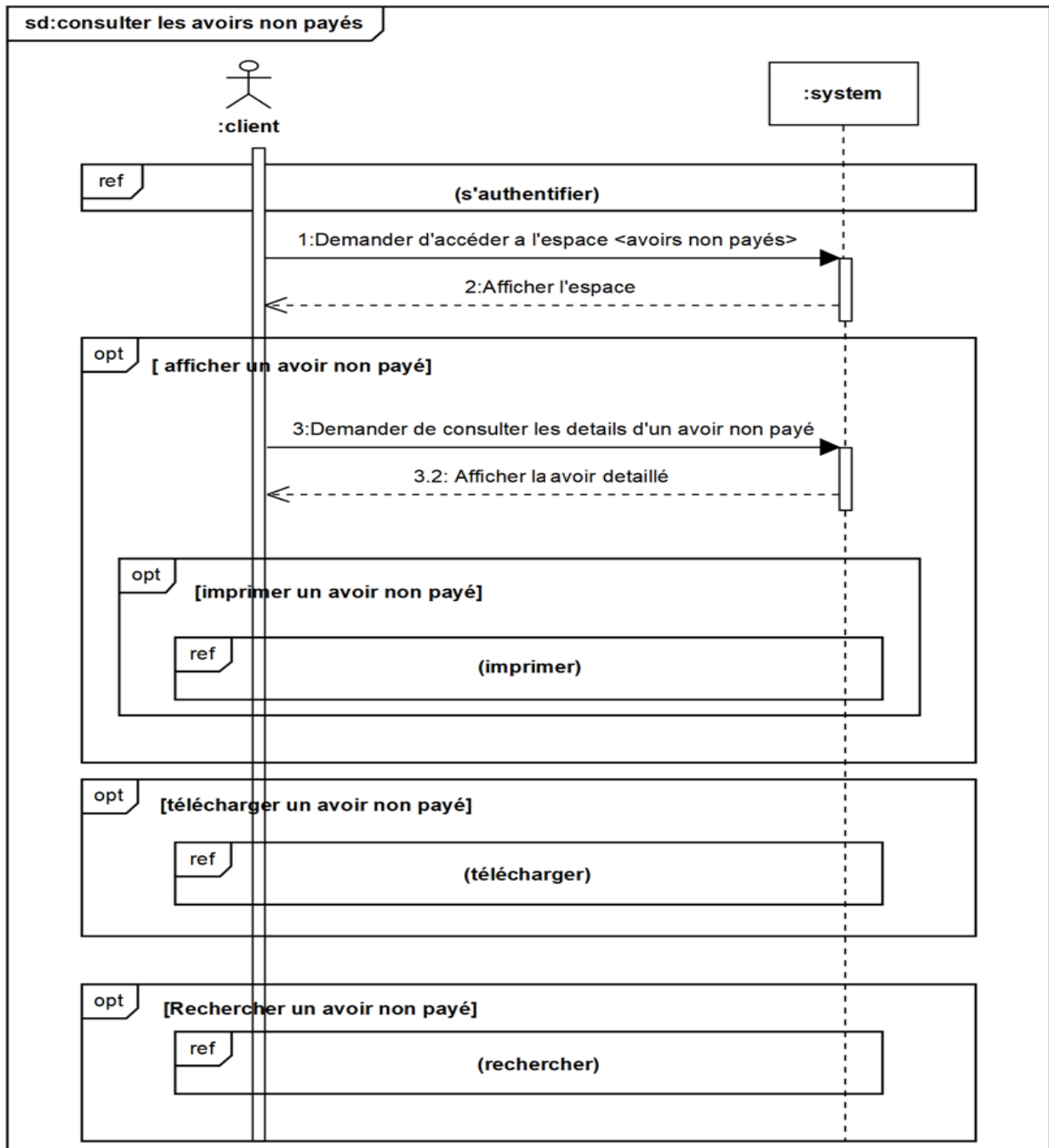
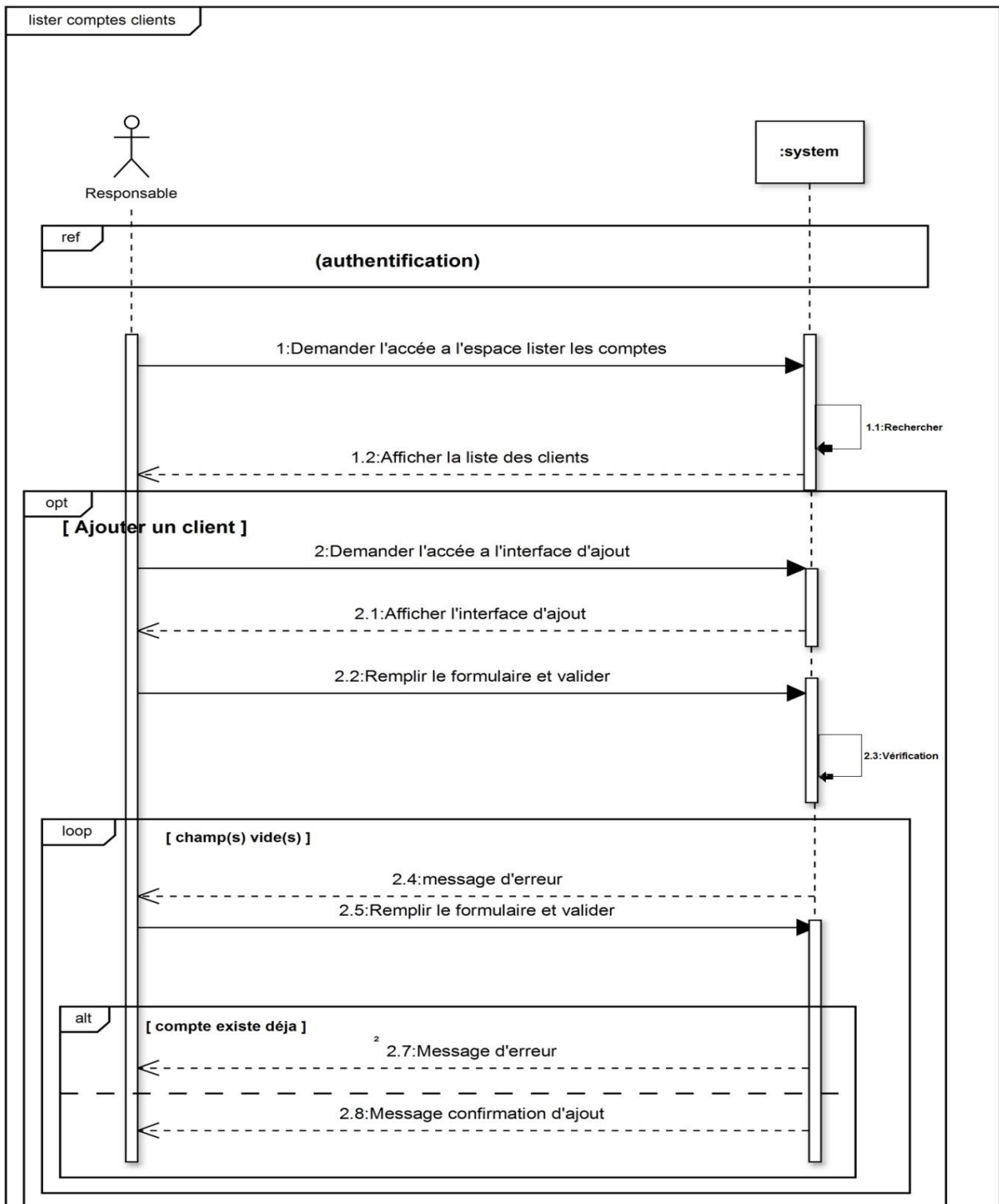


Figure II.12 Diagramme de séquence système «consulter avoirs non payés »

II.5.7 Diagramme de séquence système pour cas d'utilisation «lister les comptes clients »

Un responsable peut lister les comptes des clients à travers cette interface, il peut ajouter, modifier et supprimer un client



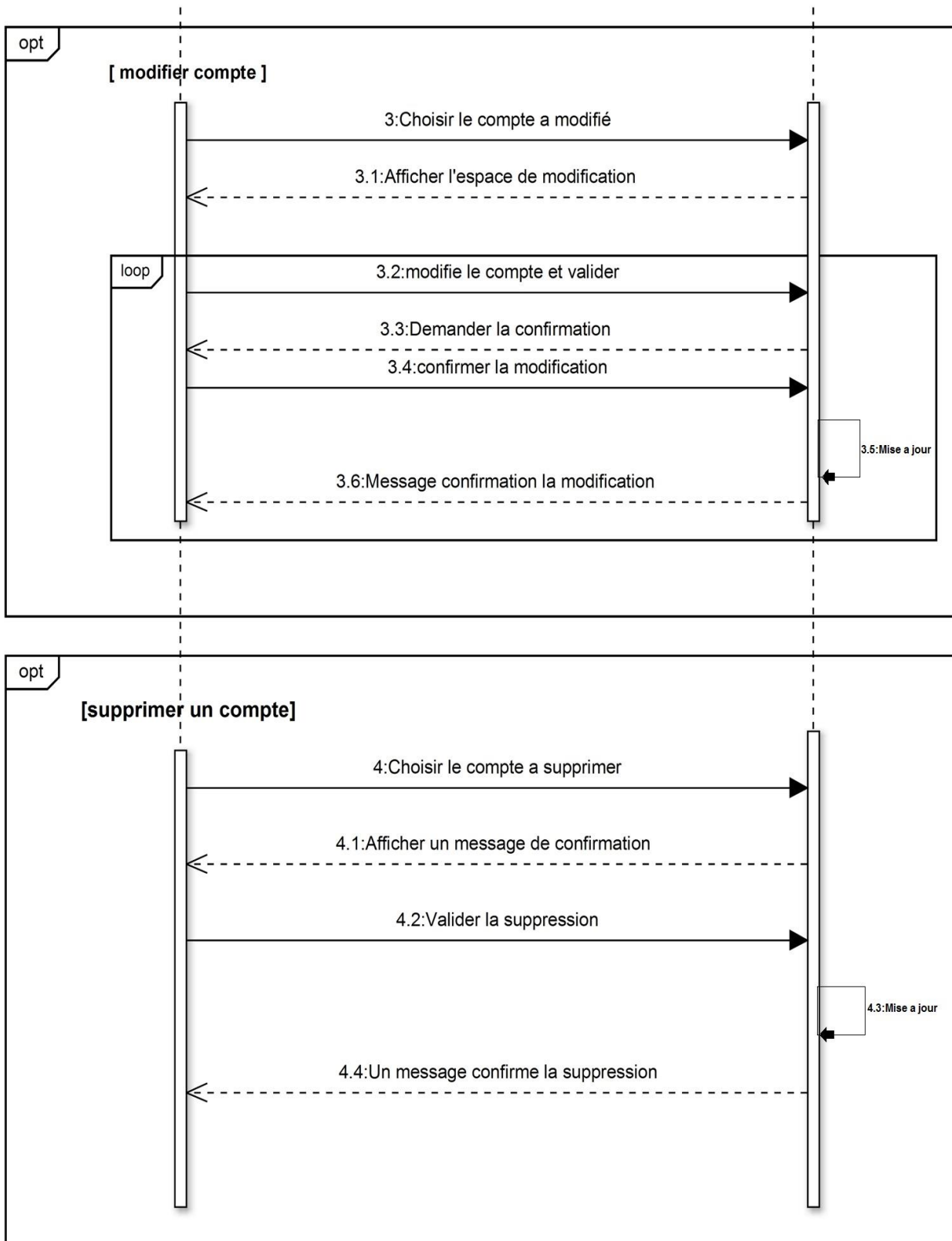


Figure II.13: Diagramme de séquence système «Lister les comptes clients»

II.6 Conclusion

Nous avons, dans ce chapitre, analysé les besoins de notre application via des diagrammes de cas d'utilisation, suivi par des diagrammes de séquence système. Cette phase d'analyse nous a permis de décrire de manière globale les besoins de nos utilisateurs, le fonctionnement désiré du système afin d'en faciliter la réalisation et la maintenance. Dans le chapitre suivant, nous entamons une phase très importante dans laquelle nous décrirons de manière détaillée comment ces besoins seront réalisés dans notre application.

CHAPITRE III

CONCEPTION ET ELABORATION DU SCHEMA RELATIONNEL

III.1 Introduction

La phase de la conception est une étape importante de réflexion dans le cycle de développement logiciel après la phase de l'analyse et de spécification. Elle permet de structurer, organiser, planifier le projet. Dans ce chapitre, nous allons présenter en détails la conception du projet à travers les diagrammes d'interaction, le diagramme de classes et aussi le modèle relationnel.

III.2 Diagrammes d'interactions

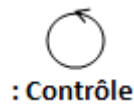
Pour chaque diagramme de séquence système, nous établirons le diagramme de séquence d'interaction, dans lequel le système est remplacé par les objets qui interviennent pour réaliser le cas d'utilisation concerné. Pour ce type de diagramme, nous avons trois types de classes [6] :

- Classes d'interface (boundary) : des classes qui permettent l'interaction entre l'application et ses utilisateurs. Pour chaque cas d'utilisation, il y a au moins une classe d'interface. Ce type de classe est schématisé comme suit :



Classes de Contrôle (Control) : Ce sont des classes qui contiennent les traitements et la cinématique de l'application. Elles font la transition

- entre les classes d'interface et les classes entité. Elles sont schématisées comme suit :



- Classes entité (entity) : Elles représentent les objets métiers, et ce sont très souvent des entités persistantes, c'est-à-dire qui vont garder leurs informations (données) après l'exécution d'un cas d'utilisation particulier. En général, elles sont enregistrées dans une base de données. Leur schématisation se fait grâce à ce stéréotype :



III.2.1 Les Diagrammes d'interaction des cas d'utilisation

III.2.1.1 Diagramme d'interaction de cas d'utilisation «S'authentifier »

L'utilisateur de l'application doit s'authentifier pour accéder à l'application, et de profiter de ces fonctionnalités, et ceci en saisissant le login et le mot de passe.

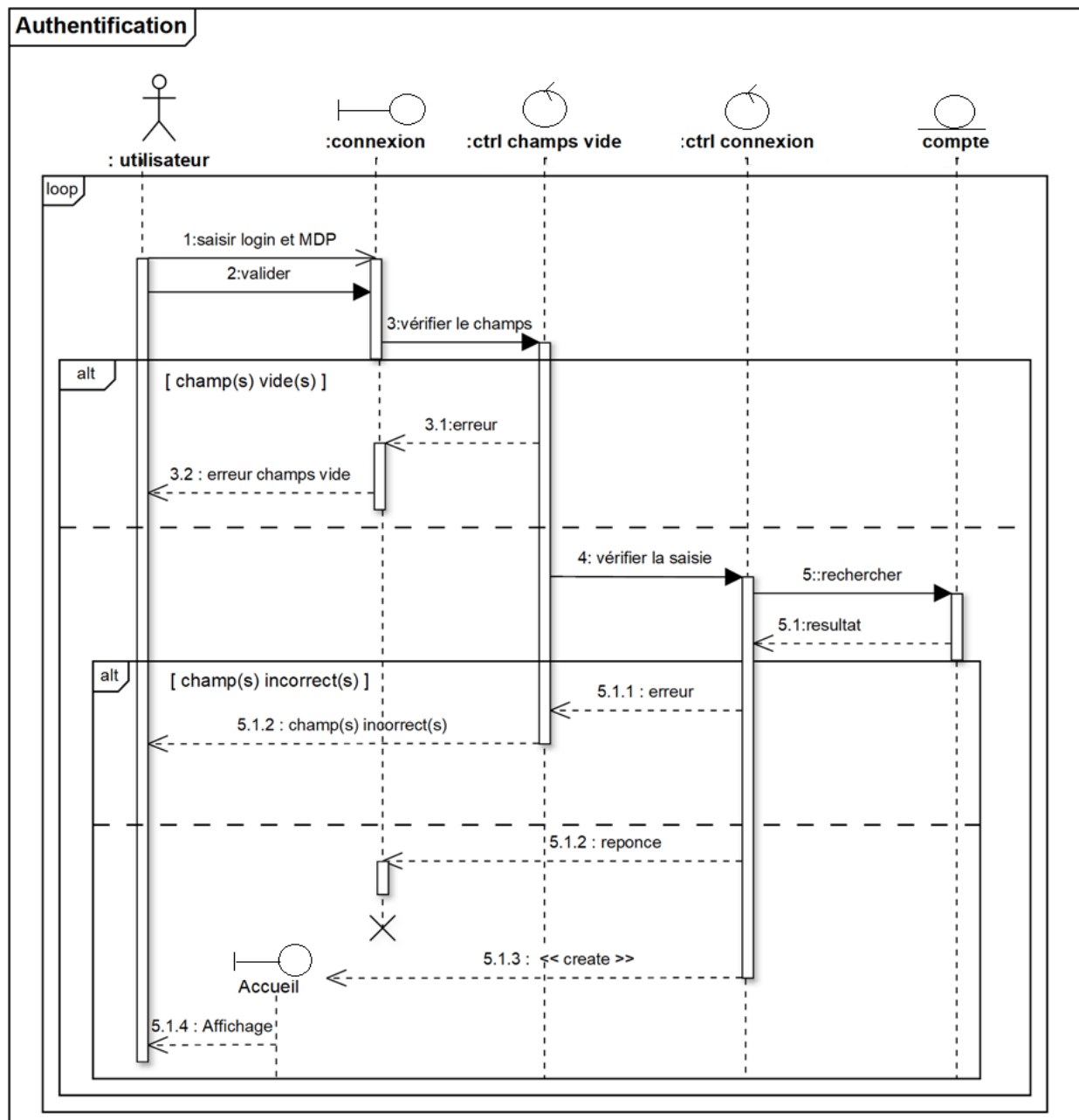


Figure III.1 : Diagramme d'interaction de cas d'utilisation «s'authentifier»

III.2.1.2 Diagramme d'interaction de cas d'utilisation «Rechercher»

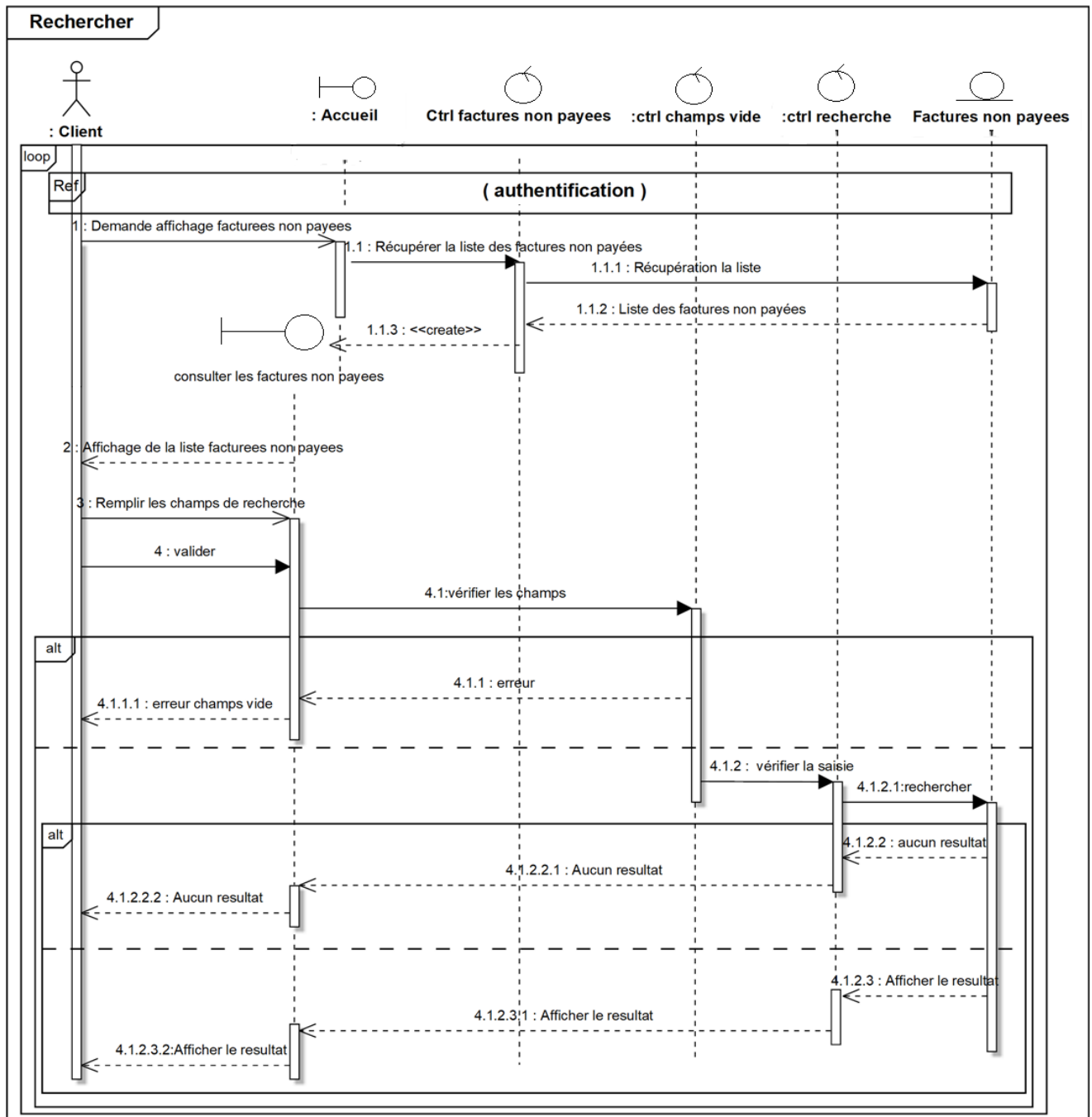


Figure III.2 : Diagramme d'interaction de cas d'utilisation «Rechercher»

III.2.1.3 Diagramme d'interaction de cas d'utilisation « imprimer »

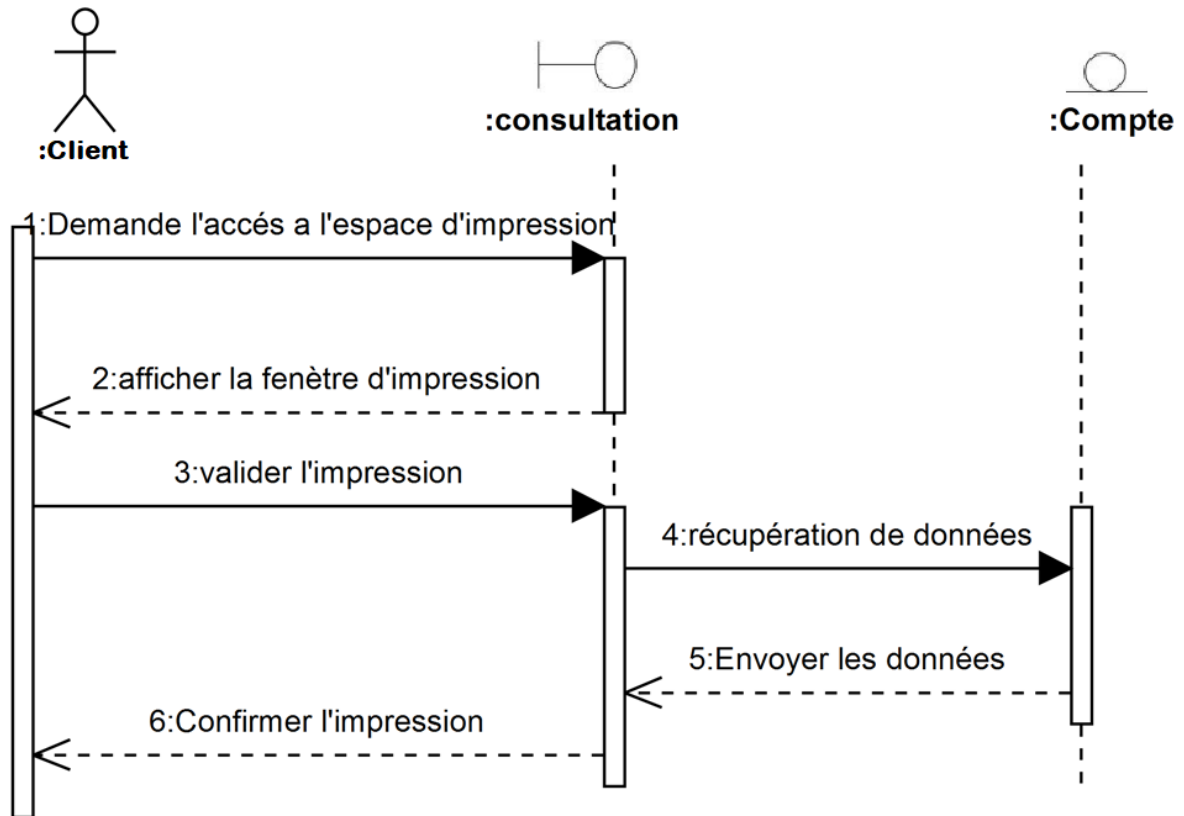


Figure III.2: Diagramme d'interaction de cas d'utilisation « imprimer »

III.2.1.4 Diagramme d'interaction de cas d'utilisation «télécharger»

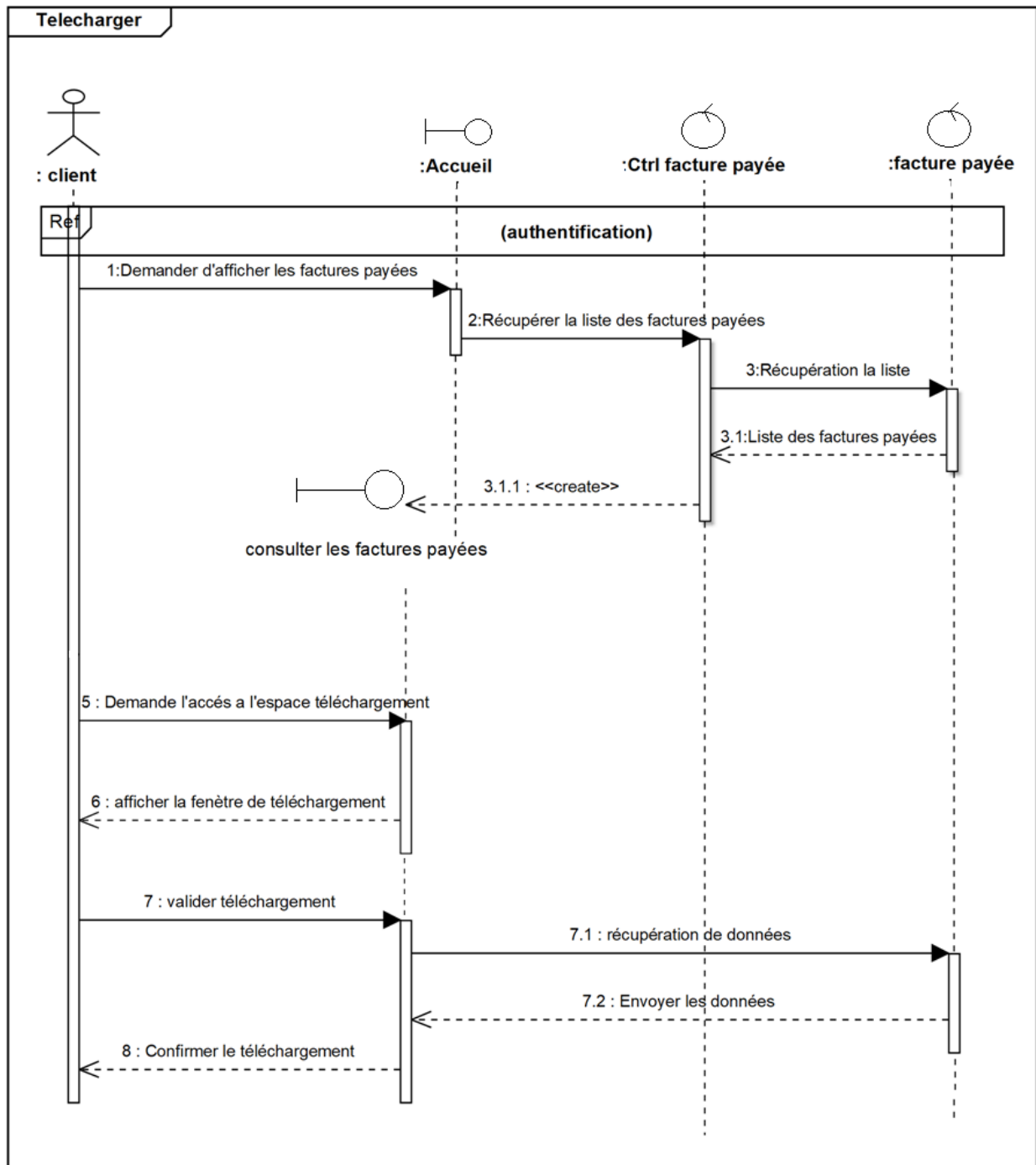


Figure III.3: Diagramme d'interaction de cas d'utilisation «télécharger»

III.2.1.5 Diagramme d'interaction de cas d'utilisation «consulter les factures non payées»

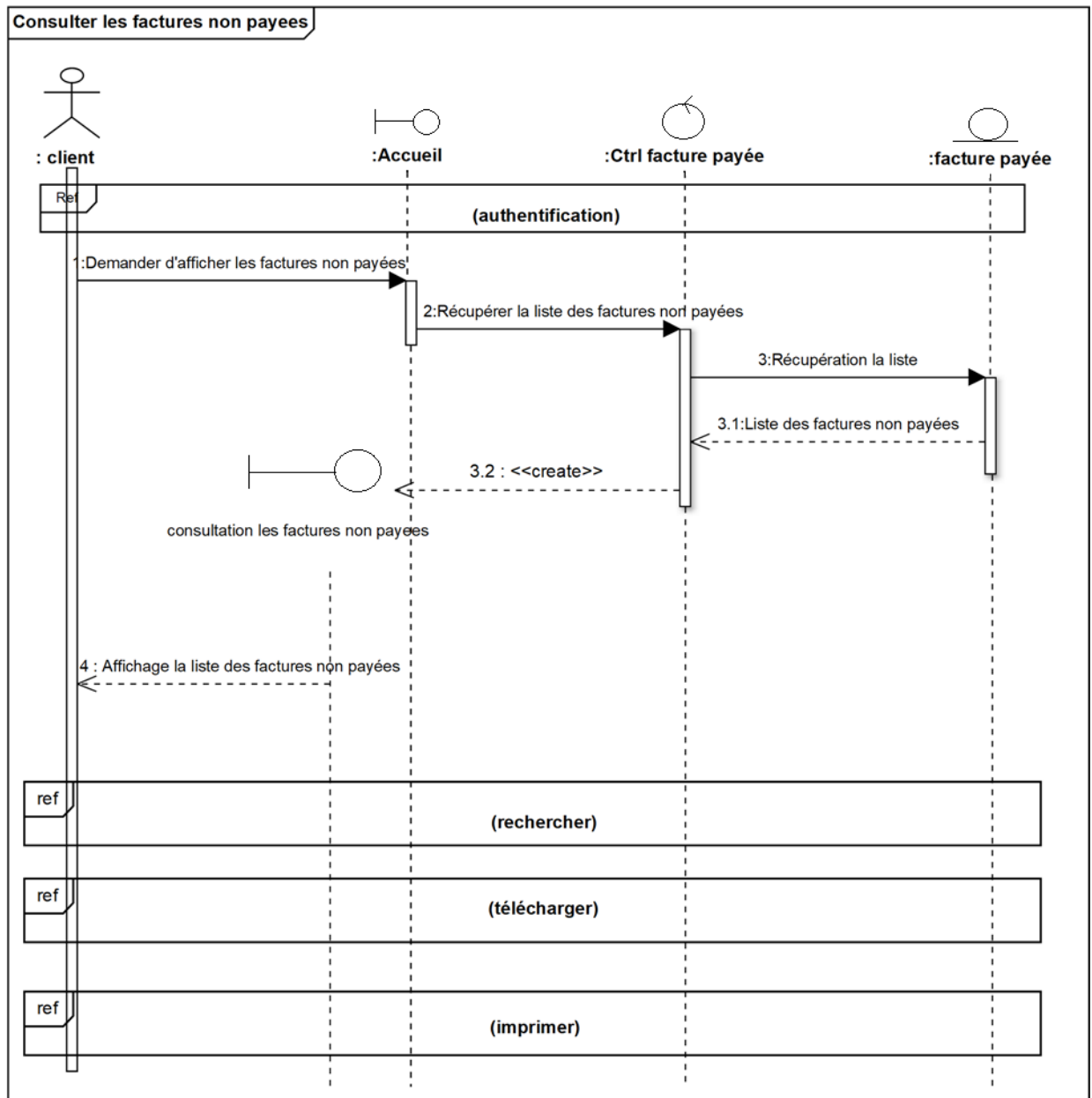


Figure III.4 : Diagramme d'interaction de cas d'utilisation «consulter les factures non payées»

III.2.1.6 Diagramme d'interaction de cas d'utilisation «consulter les avoirs non payés»

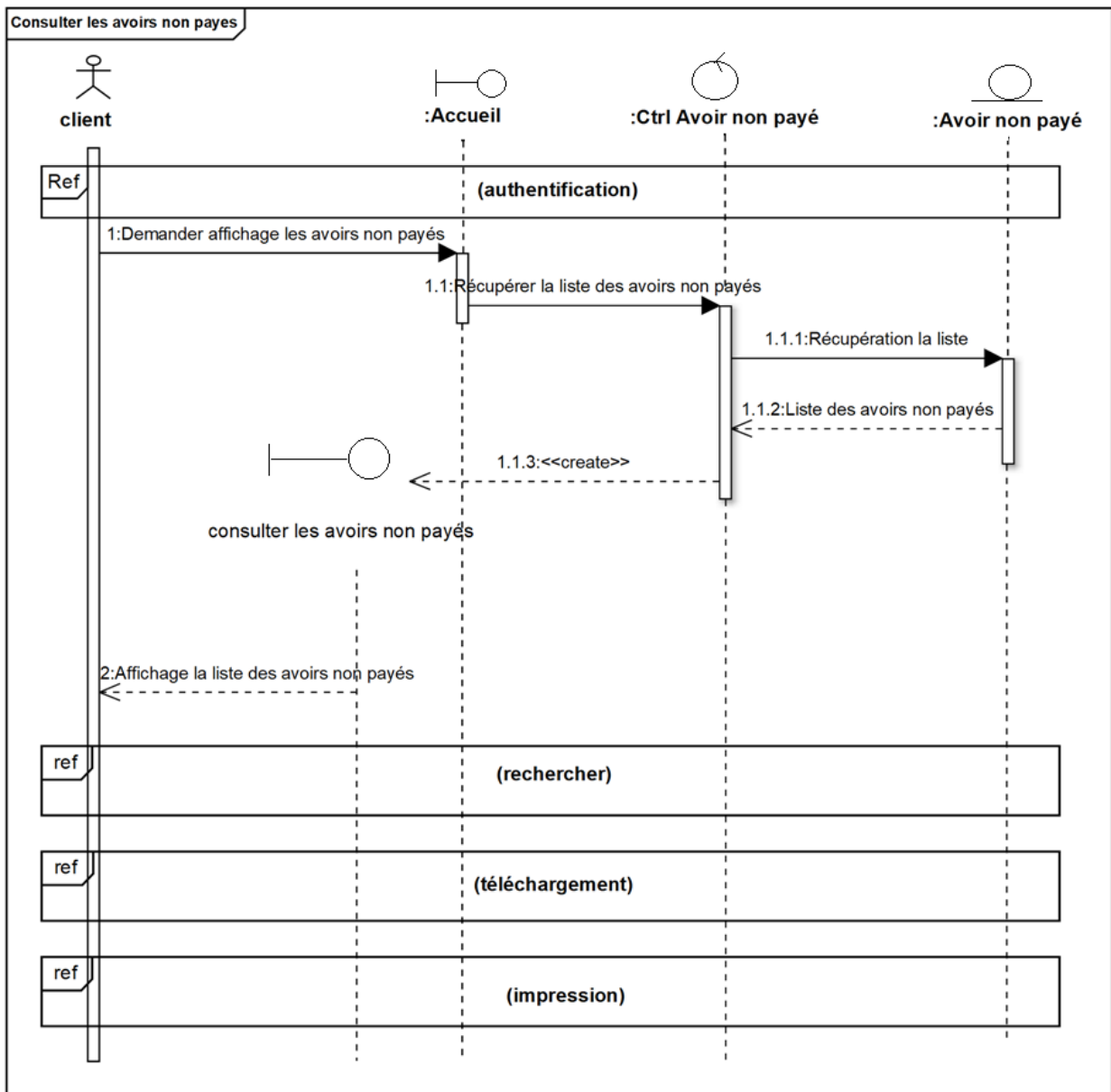


Figure III.5: Diagramme d'interaction de cas d'utilisation «consulter les avoirs non payés»

III.3 Diagramme de classe

Un diagramme de classes est un type de graphique et fait partie du langage de modélisation unifié. Il est principalement utilisé pour visualiser une structure système. En dehors de cela, il montre également les principaux composants, sa relation les uns avec les autres et leurs attributs respectifs. En tant que langage dans le domaine de la programmation, les diagrammes de classes peuvent être utilisés pour traduire des modèles en codes. [6]

III.3.1 Description détaillée des attributs de classes

Classes	Responsabilité	Attributs		
		Signification	Définition	Type
Utilisateur	Classe qui enregistre les informations des utilisateurs.	Id_utilisateur	Identifiant	Numérique
		Téléphone	Numéro de téléphone	Chaîne de caractères
		Fax	Fax	Chaîne de caractères
		Nom_utilisateur	Nom d'un utilisateur associe à un compte	Chaîne de caractères
		Mot_de_passe	Mot de passe associe à un compte	Chaîne de caractères
Client	Classe qui enregistre les informations des clients.	id_client	Identifiant d'un client	Numérique
		code_client	Code d'un client	Chaîne de caractères
		raison_social	Raison social d'un client	Chaîne de caractères
		Adresse	Adresse d'un client	Chaîne de caractères
		siege_social	Siege social d'un	Chaîne de

			client	caractères
		n_registre_commerce	Numéro de registre de commerce d'un client	Chaine de caractères
		n_nis	Numéro d'identification statique	Chaine de caractères
		n_fiscal	Numéro fiscal	Chaine de caractères
Responsable	Classe qui enregistre les informations d'un responsable.	Id_res	Identifiant d'un responsable	Numérique
		Nom	Nom d'un responsable	Chaine de caractères
		Prénom	Prénom d'un responsable	Chaine de caractères
		Service	Service d'un responsable	Chaine de caractères
Facture	Classe qui enregistre les informations des factures	-id_facture	Identifiant d'une facture	Numérique
		num_facture	Numéro d'une facture	Chaine de caractères
		num_article	Numéro d'un article	Chaine de caractères
		num_dossier	Numéro d'un dossier	Chaine de caractères
		num_escale	Numéro d'une escale	Chaine de caractères
		HT	Hors taxe	Chaine de caractères
		TVA	Taxe sur la valeur ajoutée	Chaine de caractères
Avoir	Classe qui	-id_avoir	Identifiant d'un avoir	Numérique

	enregistre les informations des factures	num_avoir	Numéro d'un avoir	Chaîne de caractères
		num_article	Numéro d'un article	Chaîne de caractères
		num_dossier	Numéro d'un dossier	Chaîne de caractères
		num_escale	Numéro d'une escale	Chaîne de caractères
		HT	Hors taxe	Chaîne de caractères
		TVA	Taxe sur la valeur ajoutée	Chaîne de caractères
Prestation	Classe qui enregistre les informations des prestations	id_prestation	Identifiant d'une prestation	Numérique
		code_prestation	Code d'une prestation	Chaîne de caractères
		Designation_prestation	Désignation d'une prestation	Chaîne de caractères
		unité:	Unité d'une prestation	Chaîne de caractères
Notification_facture	Classe qui enregistre les informations des notifications des factures	Id_notification_f	Identifiant d'une notification d'une facture	Numérique
		Description	Description d'une notification d'une facture	Chaîne de caractères
Notification_avoir	Classe qui enregistre les informations des notifications des avoirs	Id_notification_a	Identifiant d'une notification d'un avoir	Numérique
		Description	Description d'une notification d'un avoir	Chaîne de caractères

Table III.0.1: Description des classes d'objets et leurs attributs.

III.3.2 Description détaillée des attributs des classes d'associations

Le tableau ci-dessous représente les classes d'associations et leurs attributs

Classes	Responsabilité	Attributs		
		Signification	Définition	Type
Ligne_facture	Classe qui enregistre les informations des lignes facture	Id_ligne_f	Identifiant d'une ligne facture	Numérique
		designation	Designation d'une ligne facture	Chaine de caractères
		Quantité	Quantité	
		Unité	Unité d'une ligne facture	Chaine de caractères
		prix_unitaire	Prix unitaire d'une prestation	Réel
		montant	Montant	Réel
Ligne_avoir	Classe qui enregistre les informations des lignes avoirs	Id_ligne_f	Identifiant d'une ligne avoir	Numérique
		designation	Designation d'une ligne avoir	Chaine de caractères
		Quantité	Quantité	
		Unité	Unité d'une ligne facture	Chaine de caractères
		prix_unitaire	Prix unitaire d'une prestation	Réel
		Montant	Montant	Réel

Période	Classe qui enregistre les informations des périodes	Date_tarif _prestation	Date tarif et prestation	Date
---------	---	---------------------------	--------------------------	------

Table III.2 : Description détaillée des attributs des classes d'associations

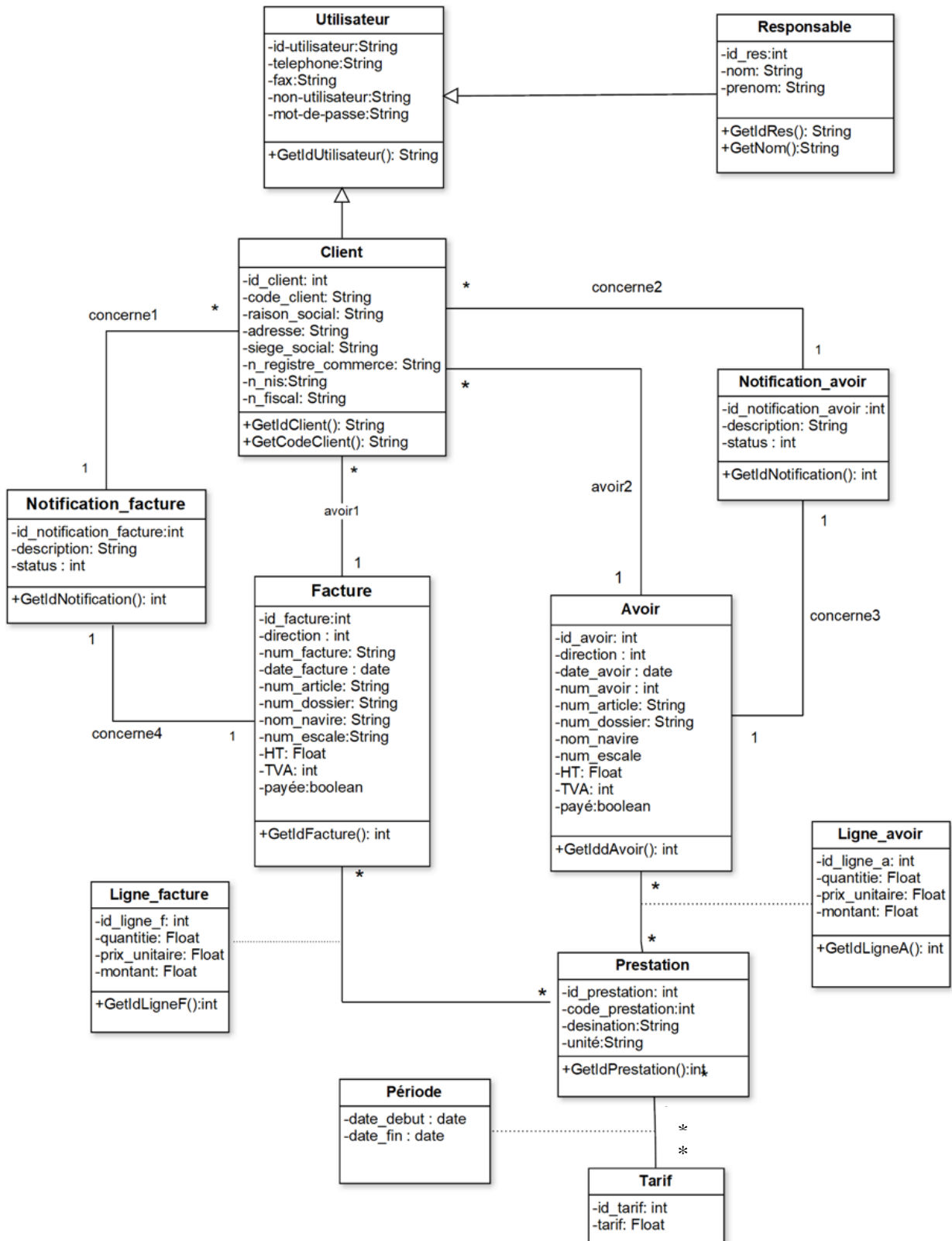


Figure III.7 – Diagramme de classes.

III.3.3 Passage du modèle objet au modèle relationnel

L'utilisation d'un SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles. Les deux structures ayant des analogies, les équivalences sont utilisées pour en réaliser le rapprochement.

Une classe définit une structure de données à laquelle souscrivent des instances, elle correspond donc à une table du modèle relationnel : chaque attribut donne lieu à une colonne, chaque instance stocke ses données dans une ligne (*T-uplet*) et son OID sert de clé primaire.

Certains attributs de type complexe ne correspondent à aucun des types de SQL ; on rencontre fréquemment ce cas pour les attributs représentant une structure de données. Un type complexe peut être conçu ;

- Soit avec plusieurs colonnes, chacune correspondant à un champ de la structure ;
- Soit avec une table spécifique dotée d'une clé étrangère pour relier les instances aux valeurs de leur attribut complexe. [6]

Modèle objet	Modèle relationnel
Classe	Table
Attribut de type simple	Colonne
Attribut de type composé	Colonne ou clé étrangère
Instance	T-uplet
OID	Clé primaire
Association	Clé étrangère ou Table de liens
Heritage	Clé primaire identique sur plusieurs tables

Table III.3: Équivalences entre les concepts objets et relationnels

Après avoir appliqué tous les règles de passage au modèle relationnel en utilisant la Décomposition descendante (push-down) : Il faut faire migrer tous les attributs de la classe mère dans la (les) relation(s) issue(s) de la (des) sous classe(s), nous avons obtenu le schéma suivant :

Client	(<u>id_client</u> , téléphone, fax, Nom_utilisateur, mot-de-passe, code_client, raison_social, n_registre_commerce, n_nis, n_fiscal, adresse).
Responsable	(<u>id_responsable</u> , téléphone, fax, nom utilisateur, mot_de_passe, nom, prénom, service).
Facture	(<u>id_facture</u> , num_facture, num_article, num_dossier, nom_navire, num_escale, HT, TVA, payée, #id_client).
Avoir	(<u>id_avoir</u> , num_avoir, num_article, num_dossier, nom_navire, num_escale, HT, TVA, payé, #id_client).
Prestation	(<u>id_prestation</u> , code_prestation, designation, unité).
Tarif	(<u>id_tarif</u> , tarif).
Notification_facture	(<u>id_notification_f</u> , #id_client, #id_facture, description, status).
Notification_avoir	(<u>Id_notification_a</u> , description, #id_client #, id_facture, description, status).
Ligne_facture	(<u>id_ligne_f, # id_facture, # id_prestation</u> , designation, quantité, unité, prix_unitaire, montant).

Ligne_avoir (id_ligne_a, #id_avoir, #id_prestation, designation, quantité, unité, prix_unitaire, montant).

Période (#id_tarif, #id_prestation, date_tarif_prestation).

III.4 Conclusion

Nous avons présenté dans ce chapitre la phase de conception de notre projet via les diagrammes d'interactions, qui nous ont permis de décrire de manière globale et détaillée, le fonctionnement désiré du système. Nous avons recensé par la suite les règles de passage du diagramme de classe vers le modèle relationnel qui nous permet d'avoir le schéma de la base de données de l'application à réaliser.

CHAPITRE III

REALISATION

IV.1 Introduction

Dans les chapitres précédents nous avons achevé la spécification et la conception de notre application web. Ainsi nous arrivons au terme de ce projet avec cet ultime chapitre intitulé « Réalisation ».

Dans ce chapitre, en premier lieu nous traiterons les aspects liés à la description de l'environnement matériel et logiciel adopté. Par la suite nous donnerons un aperçu sur les fonctionnalités de notre application.

IV.2 Langage et outils de développement

Dans cette section, nous allons énumérer les différentes technologies et langages qui sont utilisés pour développer notre système.

IV.2.1 Les technologies utilisés

- **XAMP**

Xampp est un ensemble de logiciels servant à mettre en place aisément un serveur Web, un serveur FTP et un serveur de messagerie électronique. C'est une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, reconnue pour son installation simple et rapide. Ainsi, il est à la portée de la plupart de personnes dans la mesure où il ne requiert pas de connaissances spécifiques et fonctionne, qui plus est, sur les dispositifs d'exploitation les plus communs. [7]

- **PhpMyAdmin**

PhpMyAdmin est un outil logiciel gratuit écrit en PHP, destiné à gérer l'administration de MySQL sur le Web. PhpMyAdmin prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) peuvent être effectuées via l'interface utilisateur, tout en vous permettant d'exécuter directement n'importe quelle instruction SQL. [8]

- **Visual Paradigm**

Visual Paradigm est comme son nom le laisse supposer, un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. Disposant d'un outil créant des rapports personnalisables aux formats PDF, Word ou HTML afin de les partager et les publier sur Internet, cette application est compatible avec de nombreuses applications, standards et environnements. Ainsi, vous pourrez générer notamment des diagrammes de séquences ou de cas d'utilisation et ainsi produire du

code source dans de nombreux langages comme le Java ou encore le C++, ou bien faire l'inverse, générer des diagrammes à partir de code déjà existant. [9]

- **Visual studio code**

Visual studio code **ou** VS Code est un éditeur de code développé par Microsoft en 2015. Contrairement à ce à quoi Microsoft a eu l'habitude de nous habituer durant des années, il est l'un de ces premiers produits open source et gratuit, et surtout disponible sur les systèmes d'exploitation Windows, Linux et Mac. **Vs code** est développé avec le framework Electron et conçu principalement pour développer des projets avec JavaScript, Node.js ou encore TypeScript. [10]

- **Bootstrap**

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les Smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de framework un "Front-End Framework". [11]

- **PyCharm**

PyCharm est un environnement de développement intégré (IDE) utilisé pour la programmation en Python. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec les systèmes de contrôle de version (VCS)

et prend en charge le développement Web avec Django. PyCharm est développé par la société tchèque JetBrains.

Il fonctionne sur plusieurs plates-formes sous Windows, Mac OS X et Linux. PyCharm a une édition professionnelle, publiée sous une licence propriétaire et une édition communautaire publiée sous la licence Apache. L'édition communautaire de PyCharm est moins étendue que l'édition professionnelle. [12]

▪ **Json**

JSON (JavaScript Object Notation) est un format d'échange de données léger. Il est facile pour les humains de lire et d'écrire. Il est facile pour les machines d'analyser et de générer. Il est basé sur un sous-ensemble de la norme de langage de programmation JavaScript ECMA-262 3eme édition - décembre 1999. JSON est un format de texte totalement indépendant du langage mais qui utilise des conventions familières aux programmeurs de la famille des langages C, y compris C, C++, C#, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal. [13]

IV.2.2 Les langages utilisés

▪ **Python**

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des

contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages. [14]

- **PHP**

PHP (Hypertext Preprocessor), désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.

Créé au début des années 1990 par le Canadien et Groenlandais Rasmus Lerdorf, le langage PHP est souvent associé au serveur de base de données MySQL et au serveur Apache. Avec le système d'exploitation Linux, il fait partie intégrante de la suite de logiciels libres LAMP. [15]

- **HTML**

HTML est le bloc de construction le plus basique du Web. Il définit le sens et la structure du contenu Web. D'autres technologies que HTML sont généralement utilisées pour décrire l'apparence/la présentation (CSS) ou la fonctionnalité/le comportement (JavaScript) d'une page Web. [16]

- **CSS**

CSS est un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML. Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc. [17]

▪ JavaScript

JavaScript ("JS" en abrégé) est un langage de programmation dynamique à part entière qui peut ajouter de l'interactivité à un site Web. Il a été inventé par Brendan Eich (co-fondateur du projet Mozilla, la Fondation Mozilla et la Mozilla Corporation. [18])

IV.3 Diagramme déploiement

La figure suivante illustre les modules logiciels de notre application, et leur répartition sur différentes machines physiques :

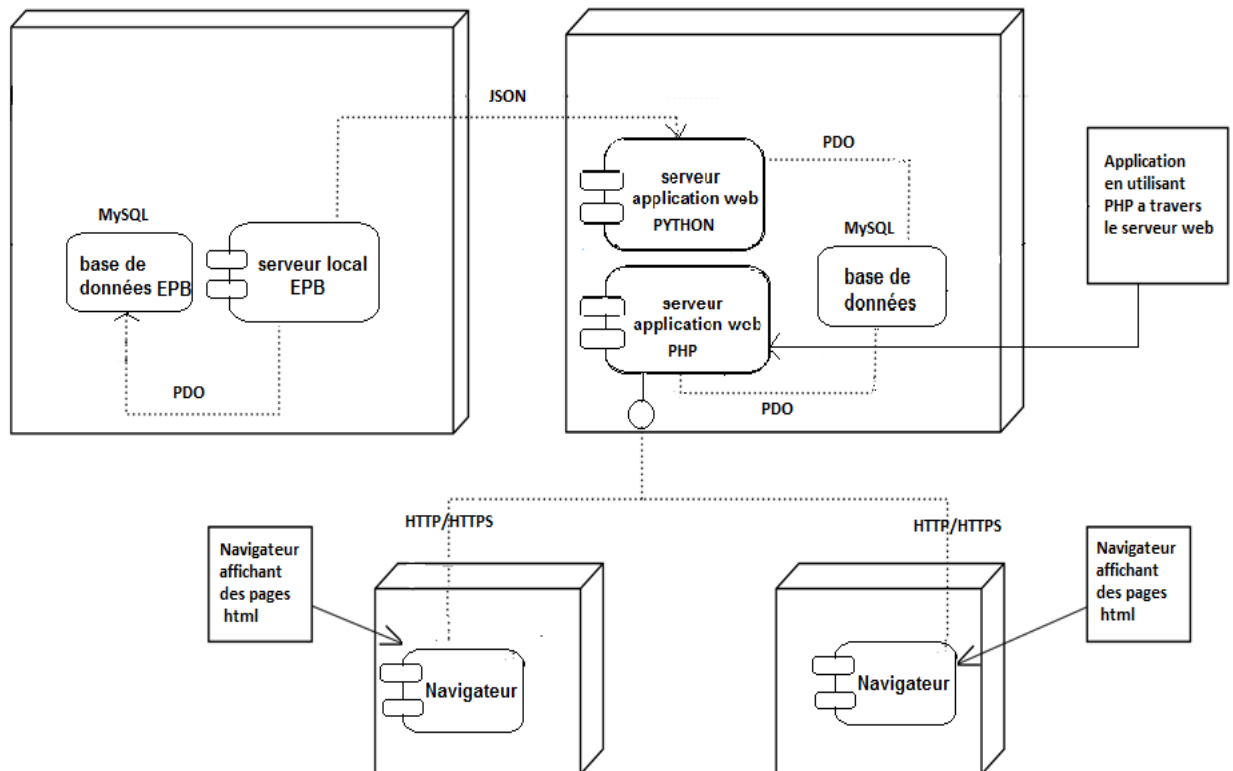


Figure IV.1 : Diagramme déploiement

IV.4 Présentation des interfaces

Dans cette section nous allons présenter quelques interfaces Homme-Machine de notre application.

IV.4.1 Interface « authentication »

Cette interface s'affiche au lancement de notre application, le formulaire d'authentification nous demandera d'introduire le nom d'utilisateur et le mot de passe. Cette étape met en valeur l'aspect sécurité de l'application. Cette interface est illustrée la Figure suivante :

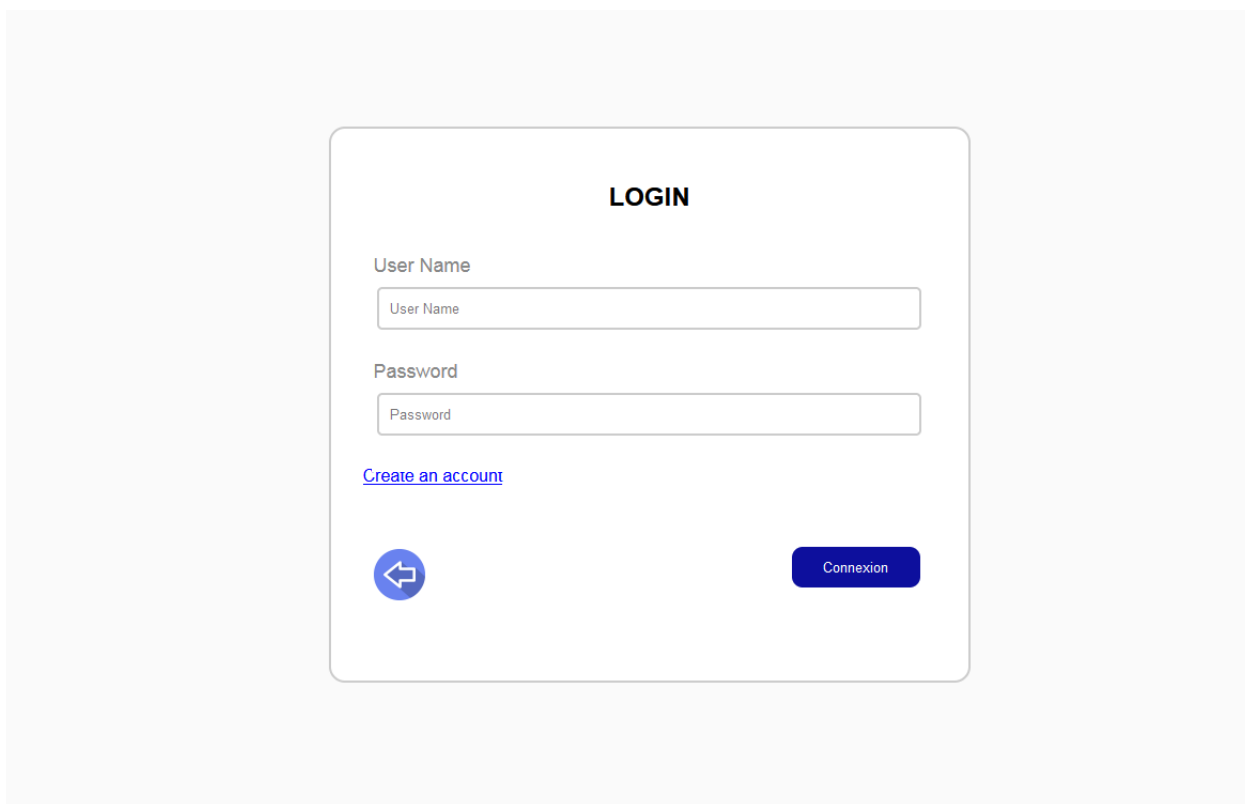


Figure IV.2 : Interface « authentication »

IV.4.2 Interface « Lister les clients »

Cette interface affiche la liste des clients qui existe dans la base de données avec comme option la possibilité d'ajouter, modifier ou de supprimer un client (le clic sur ce bouton nous affiche une boite de confirmation avec la possibilité de confirmer ou d'annuler la suppression), ceci est illustré dans la Figure suivante :

IV.4.2.1 Interface « ajouter un client »

The figure shows two screenshots of a web application interface. The top screenshot displays the 'liste des clients' page with a table of client records. The bottom screenshot shows a modal form titled 'ajouter un client' with various input fields for client details.

ID	code_client	adresse	N°RC	num_tel	num_fax	modifier	supprimer
1	475004	Bejaia	98-A-0419810-34/03	054378901	03578		
2	472585	Bejaia	0934087 A03	098765443	098765		
3	475003	Bejaia	98-A-0419810-34/02	064539289	04321		
4	472580	Bejaia	0934087 A02	098765443	098765		

The modal form 'ajouter un client' contains the following fields:

- code_client:
- raison_social:
- adresse:
- siege_social:
- N°RC:
- n_nis:
- n_fiscal:
- num_tel:
- num_fax:
- login:
- Password:

Buttons:

Figure IV.3 : Interface « ajouter un client »

IV.4.2.2 Interface « Modifier un client »

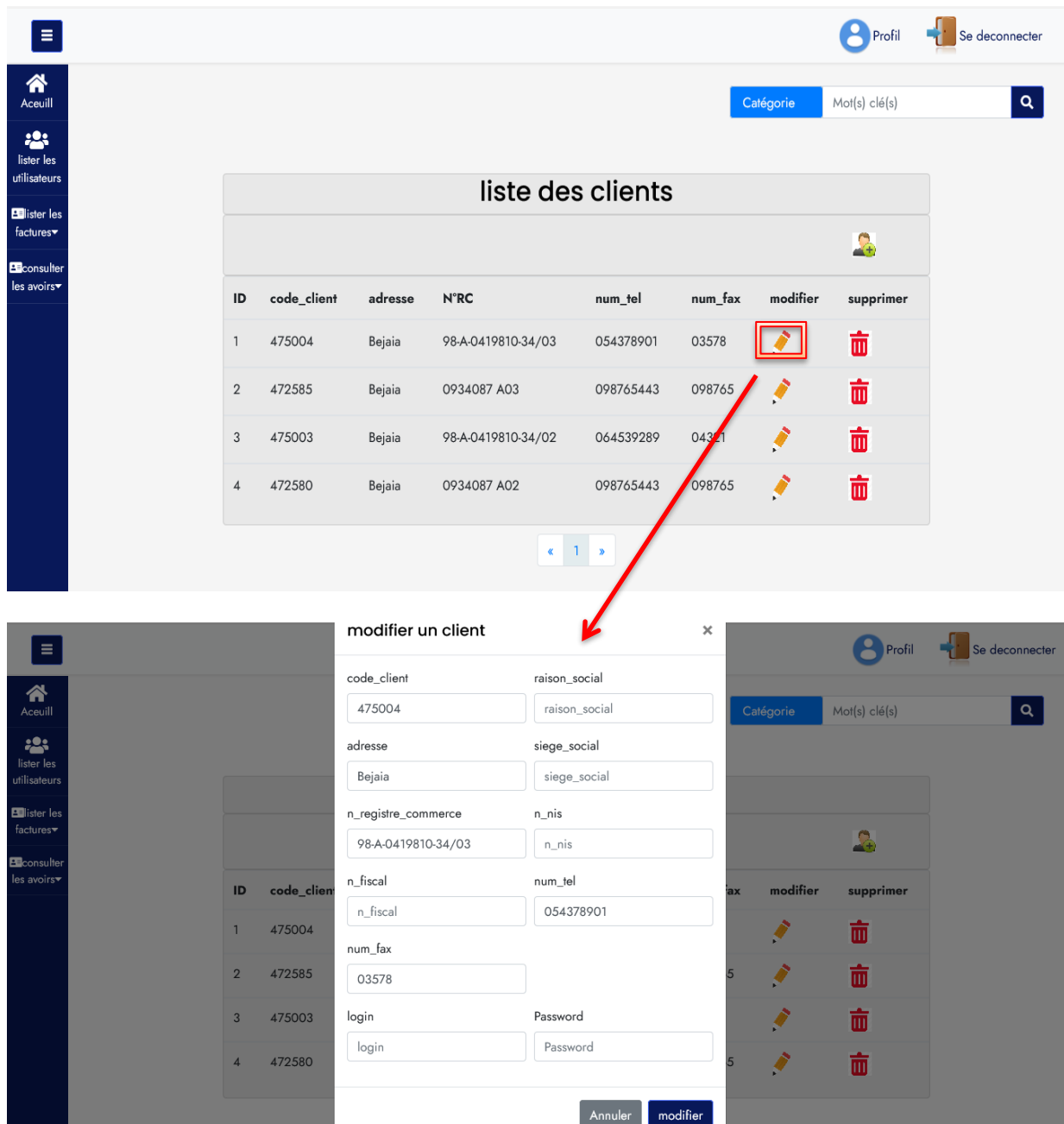


Figure IV.4 : Interface «modifier un client »

IV.4.2.3 Interface « Supprimer un client »

The figure consists of two screenshots of a web application interface. The top screenshot shows a page titled 'liste des clients' with a table of client data. A red box highlights the 'supprimer' icon (a trash can) for the first client. The bottom screenshot shows the same page with a confirmation dialog box overlaid on top. The dialog box contains the text 'supprimer un client' and 'vous etre sur ?' with two buttons: 'non' and 'oui'. A red arrow points from the 'supprimer' icon in the top screenshot to the dialog box in the bottom screenshot.

ID	code_client	adresse	N°RC	num_tel	num_fax	modifier	supprimer
1	475004	Bejaia	98-A-0419810-34/03	054378901	03578		
2	472585	Bejaia	0934087 A03	098765443	098765		
3	475003	Bejaia	98-A-0419810-34/02	064539289	04321		
4	472580	Bejaia	0934087 A02	098765443	098765		

Figure IV.5 : Interface « supprimer un client »

IV.4.3 Interface « Profil »

Cette interface affiche les informations concernant l'utilisateur ainsi elle donne comme option la possibilité de faire mise à jour des ainsi de donner la possibilité de modifier le mot de passe.

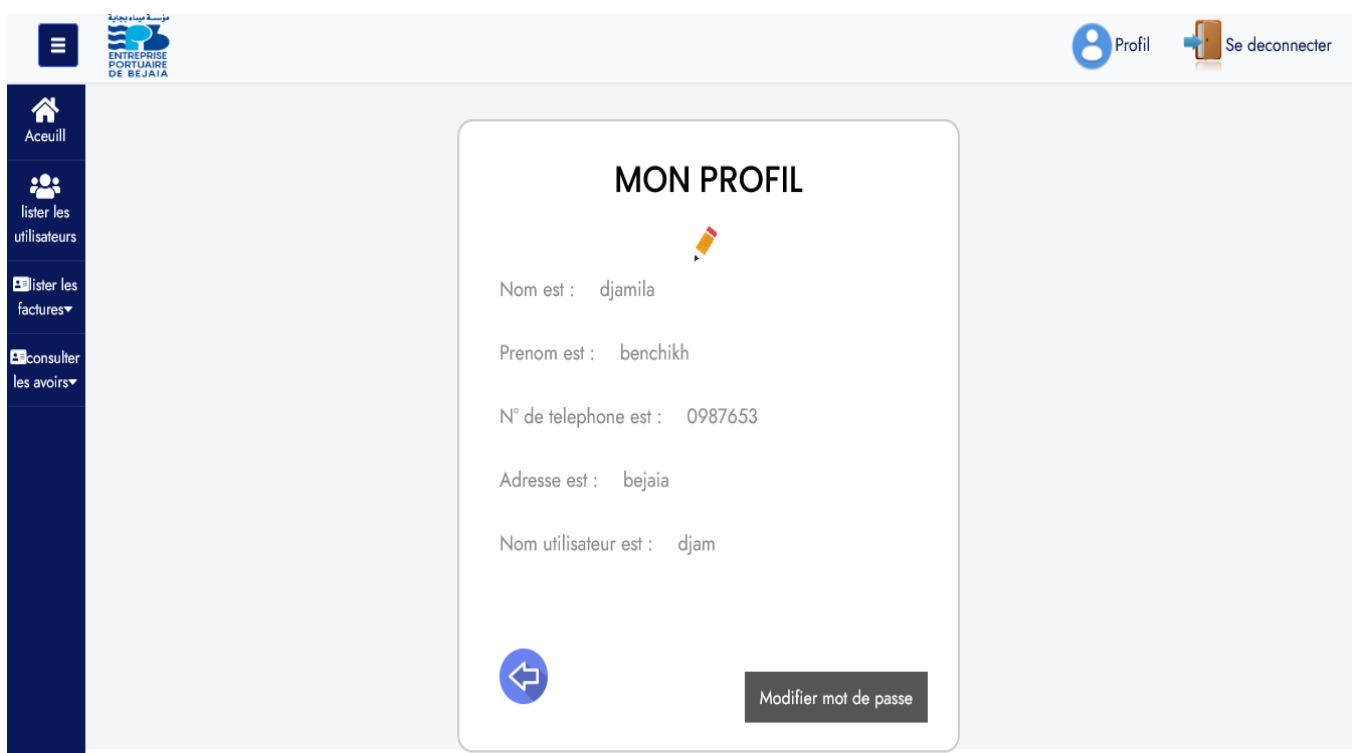



Figure IV.6 : Interface « Profil »

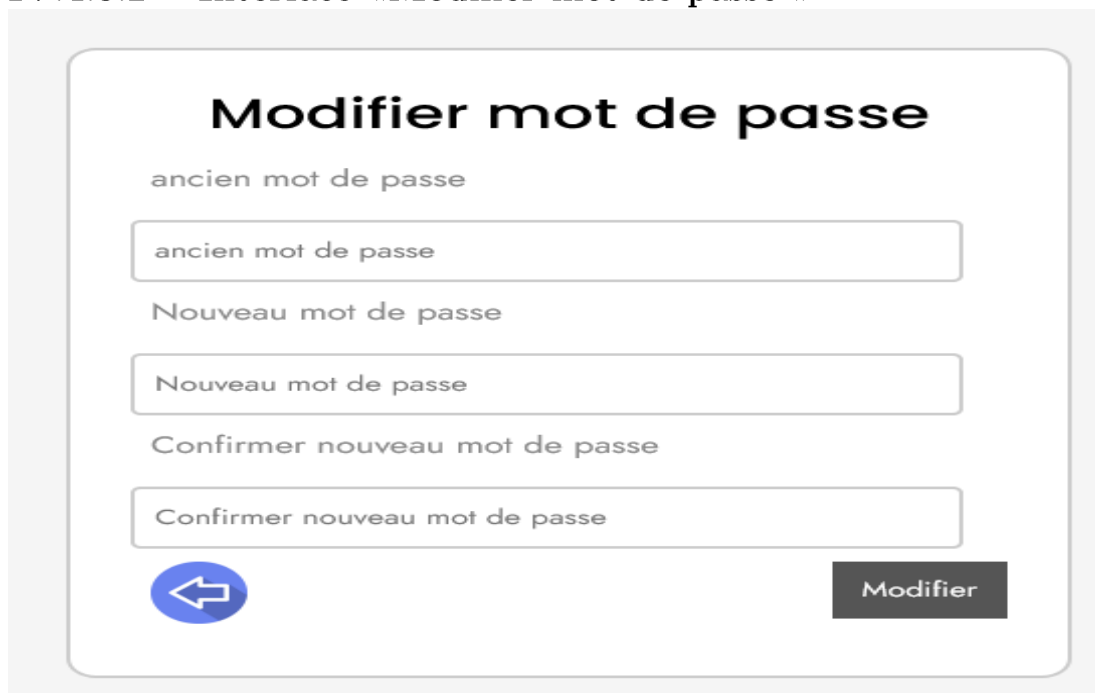
IV.4.3.1 Interface «Modifier un profil»



The screenshot shows a web interface titled "MODIFIER MON PROFIL". It contains five text input fields with the following values: "djamila", "benchikh", "0987653", "bejaia", and "djam". At the bottom left, there is a blue circular button with a white left-pointing arrow. At the bottom right, there is a dark grey rectangular button labeled "Modifier".

Figure IV.7 : Interface «Modifier un profil»

IV.4.3.2 Interface «Modifier mot de passe »



The screenshot shows a web interface titled "Modifier mot de passe". It contains three text input fields with the following labels and values: "ancien mot de passe" (value: "ancien mot de passe"), "Nouveau mot de passe" (value: "Nouveau mot de passe"), and "Confirmer nouveau mot de passe" (value: "Confirmer nouveau mot de passe"). At the bottom left, there is a blue circular button with a white left-pointing arrow. At the bottom right, there is a dark grey rectangular button labeled "Modifier".

Figure IV.8 : Interface «Modifier mot de passe»

IV.4.4 Interface «consulter les factures non payées »

Cette interface affiche la liste des factures non payées d'un utilisateur ainsi elle donne comme option la possibilité d'afficher la facture PDF d'où on peut l'imprimer comme elle donne également la possibilité de télécharger et rechercher une facture non payée.

The screenshot displays the 'Espace client' interface for the 'ENTREPRISE PORTUAIRE DE BEJAIA'. The main content area is titled 'Liste des factures non payées' and contains a table with the following data:

N° Facture	Navire	Escale	Date	Format	Telecharger
205	VICTORIA VI	20210226	2021-08-12		
1234	VICTORIA VI	20201546	2021-09-20		
208	CONTSHIP ZEN	20893456	2021-09-22		

The interface also features a navigation menu on the left with options like 'Acceuil', 'consulter les factures', 'consulter les avoirs', 'paramètre', and 'A propos'. At the top right, there are icons for notifications (with a '3' badge), user profile, and a search bar with a 'Catégorie' dropdown and 'Mot(s) clé(s)' input field.

Figure IV.9 : Interface «consulter les factures non payées »

IV.4.5 Interface «afficher la facture non payée»

Cette interface affiche la facture non payée d'un client sous forme d'un fichier PDF

Entreprise Portuaire de Bejaia Siege Social : 13,Avenue des freres Amran B.P.94-Bejaia Tel:034 167 631/35/36/37/38/39 - 034 167 573 Fax: 034 16 75 71		Bejaia le 2000-12-12 N RC:00 b 0183582-06/00 NIS:0982 060 100 098 40 NIF : 00000 60183582 99 N d'Article : 06010106010	
Facture N: 200			
Code Client : 475004 Raison Social : TRANSIT BOUDJELLOUD P/C SARL TDI PLAST Siege Social : LOCAL N 02 CITE 05 JUILLETN 07 BBA Tel : 054378901 Fax : 03578 Navire : VICTORIA VI		N RC : 98-A-0419810-34/03 N NIS : 1961 06010 1721 49 NIF : 167895432123344	
N Gros : 20210225		N article : 20	
		Dossier : 300	
Code	Designation	Qte	Unite Prix.U Montant
1	Levage conteneurs(/unite 20 ou 40)pour transfert (pliens import)	3.000	unite 4500.000 13500.0000
10	Frais divers a appliquer aux conteneur	3.000	unite 3000.000 9000.0000
12	Scanning	1.000	unite 150.000 150.0000
33	Scanning	1.000	unite 150.000 150.0000
36	Levage conteneurs(/unite 20 ou 40)pour transfert (pliens import)	3.000	unite 4500.000 13500.0000
37	Levage conteneur (/unite 20 ou 40)pour Pesee	5.000	unite 4500.000 22500.0000
38	Scanning	1.000	unite 2000.000 2000.0000
39	Levage conteneur (/unite 20 ou 40)pour Visite	4.000	unite 4500.000 18000.0000
		Total HT: 490000.00	
		TVA : 19% 93100.00	
		Total TTC : 45619000000	

Figure IV.10 : Interface «afficher la facture non payée »

IV.4.6 Interface «consulter les notifications»

Cette interface affiche les notifications des factures et des avoirs d'un client non consultées

The screenshot shows the 'Espace client' interface for 'ENTREPRISE PORTUAIRE DE BEJAIA'. A notification pop-up is displayed over a table of invoices. The pop-up contains the following text:

- vous avez 1 nouvelles notification**
- Vous aver une nouvelle avoir pour l'escale **27654320** avec le navire **FESCO ASKOLD**
- Vous aver une nouvelle avoir pour l'escale **22654300** avec le navire **COMSTSHIP ZEN**
- Vous aver une nouvelle facture pour l'escale **20210225** avec le navire **VICTORIA VI**
- Vous aver nouvelle facture pour l'escale **20201546** avec le navire **VICTORIA VI**
- VOIR TOUT**

The table below shows the invoice data:

N° Facture	Navire				
200	VICTORIA VI				
1234	VICTORIA VI				
202	CONTSHP ZEN	20893456	2000-12-13		

Figure IV.11 : Interface «consulter les notifications»

IV.5 Conclusion

Dans ce chapitre réservé à la partie développement d'une application téléfacturation, nous avons commencé par la présentation des langages et des outils de développement web que nous avons utilisé. Nous avons présenté quelques interfaces de notre application pour montrer l'implémentation de quelques cas d'utilisation cités dans le chapitre II.

CONCLUSION GENERALE

Dans ce mémoire, nous avons présenté l'EPB sous ses différents aspects organisationnels et fonctionnels afin de dégager les besoins fonctionnels pour une meilleure conception.

L'étude menée pendant notre stage au sein de cette entreprise nous a permis de recenser les lacunes, et de déduire les besoins concrets pour définir les objectifs à atteindre afin de répondre à ces besoins. Le directeur de service d'informations nous a exprimé son souhait d'automatiser la facturation ceci en développant une application web.

La solution proposée répond aux critères de gestion et facilite toutes sortes d'interaction avec l'utilisateur en diminuant le cout et réduisant le temps de transmission des factures.

Pour la conception de notre application, nous avons utilisé le langage UML pour la réalisation des différents diagrammes couvrant toutes les phases du processus de développement, se basant sur le processus UP et offrant ainsi une base solide sur laquelle nous nous sommes appuyés afin de passer à la partie réalisation du projet. Pour la mise en œuvre de l'application, nous avons eu recours à la programmation avec le langage Python coté serveur local de l'EPB et PHP coté serveur à distance (télé-facturation) et le

SGBD MySQL.

Après la réalisation de l'application, nous avons effectué plusieurs tests qui nous ont permis de souligner quelques lacunes. Ces tests nous ont également permis d'envisager des améliorations à notre application en lui ajoutant d'autres modules comme un système qui permet de superviser la situation des impayés en envoyant un avertissement lorsque le délai de paiement arrive à son terme.

Enfin, ce stage, ainsi que le travail fourni, nous ont permis de nous épanouir et d'avoir un aperçu sur le monde du travail. Par conséquent, ceci nous a poussés à développer nos compétences en termes de communication, d'analyse des besoins, de programmation et de gestion de projets.

BIBLIOGRAPHIE

- [1] Source Entreprise Portuaire de Bejaia (EPB). Consulté le 12/05/2021

- [2] Processus de développement UP, <http://files.gl3.webnode.fr/200000067-4b6db4c670/Chapitre1-ACOO2.pdf>. Consulté le 03/06/2021.

- [3] <https://www.commentcamarche.net/contents/477-methodes-agiles-rad-xp>. Consulté le 03/06/2021.

- [4] GRADY Booch et al. Le guide de l'utilisateur UML, Eyrolles, Paris, 2003. Consulté le 03/06/2021.

- [5] Pascal Roques, UML 2 : Modéliser une application web, Les Cahiers Programmeur, Eyrolles ,4e édition, Paris, 2008. Consulté le 21/06/2021.

- [6] Roy GILLES. '*UML2 en action, de l'analyse des besoins en action*'. Presses de l'Université de Québec, 2009, première édition. Consulté le 13/07/2021.

- [7] <http://www.standard-du-web.com/xampp.php> . Consulté le 05/09/2021.

- [8] <https://www.phpmyadmin.net/>. Consulté le 10/09/2021.

- [9] <https://www.clubic.com/telecharger-fiche384046-visual-paradigm-for-uml.html>. Consulté le 07/08/2021.

- [10] <https://blog.webnet.fr/visual-studio-code>. Consulté le 06/09/2021.

- [11] <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques>, consulté le 08/09/2021.
- [12] <https://findwords.info/term/pycharm>, consulté le 10/09/2021.
- [13] <https://www.json.org/json-en.html>, consulté le 08/09/2021.
- [14] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>, consulté le 06/09/2021.
- [15] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/>. Consulté le 06/09/2021
- [16] <https://runebook.dev/fr/docs/html/>, consulté le 06/09/2021
- [17] <https://www.futura-sciences.com/tech/definitions/internet-css-4050/> , consulté le 06/09/2021.
- [18] <https://jide.fr/apprendre-javascript>, consulté le 10/09/2021.

Résumé

Ce travail est réalisé en vue de l'obtention du diplôme master professionnel en génie logiciel, il consiste à la conception et la réalisation d'une application web pour la téléfacturation. Notre stage a été effectué au sein de l'entreprise portuaire de Bejaia (EPB). Notre application permet de pallier les insuffisances et d'améliorer le fonctionnement des directions concerné par la facturation. Ce travail a été réalisé en utilisant le langage de modélisation Unified Modeling Language (UML) qui permet de schématiser la solution ainsi que le langage de programmation PHP et MySQL comme Système de Gestion de base de données (SGBD).

Mots clés : Télé-facturation, UML, PHP, Python, MySQL, SGBD.

Abstract

This work is carried out with a view to obtaining a professional master's degree in software engineering, it consists of the design and implementation of a web application for tele-invoicing. Our internship was carried out within the Port Company of Bejaia (EPB). Our application makes it possible to remedy the shortcomings and improve the functioning of the departments concerned with invoicing. This work was carried out using the Unified Modeling Language (UML) modeling language which makes it possible to schematize the solution as well as the programming language PHP and MySQL as a Database Management System (DBMS).

Keywords: Tele-invoicing, UML, PHP, Python, MySQL, SGBD.