

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de Master Professionnel
en Informatique
Option
Administration et Sécurité des Réseaux
Thème

Conception et réalisation d'une application réseau pour la gestion d'un hôtel

Réalisé par :

M^r *MEDDAH* Amine

M^r *MEHDAOUI* Mohamed Larbi

Soutenu devant le jury composé de :

Président M^r *MEHAOUED* Kamal

Encadreur M^r *OUZEGGANE* Radouane

Examinatrice M^{me} *KHALED* Hayette

Examineur M^r *CHELIK* Mourad

Promotion 2014/2015

Remerciements

**Louange A Dieu, le miséricordieux, sans lui rien de tout cela
n'aurait pu être.**

Nous tenons à remercier vivement Mr OUZZAGANE Radouane, pour nous avoir honoré par son encadrement, pour sa disponibilité, ses orientations, ses précieux conseils et ses encouragements qui nous ont permis de mener à bien ce travail.

Nous tennons à exprimer notre gratitude aux membres de jury pour avoir accepté de juger ce travail.

Nous remercions chaleureusement tous nos enseignants pour leurs conseils, leurs gentillesse, et leurs générosités.

Un merci particulier à nos parents, pour leur amour, leur sacrifices et leurs patiences.

Un énorme merci à nos familles et amis pour leurs éternel soutient et la confiance qu'ils ont en nos capacité.

Dédicaces

Je dédie ce modeste travail à :

A mes parents,

A mes frères et soeurs,

A toute la famille,

A mes amies et collègues, et tous ceux qui m'ont aidé ;

A mon binôme Larbi et sa famille.

MEDDAH Amine

Dédicaces

Je dédie ce modeste travail à :

A mes parents,

A mes frères et soeurs,

A toute la famille,

A mes amies et collègues, et tous ceux qui m'ont aidé ;

A mon binôme Amine et sa famille.

MEHDAOUI Mohamed Larbi

Liste de abréviations

B

BDD Base De Données

F

F Fiche

FC Fiche Client

FP Fiche Police

I

IBM International Business Machines

IDE Integrated Development Environment

J

JDBC Java Data Base Connectivity

JDK Java Developement Kit

JRE Java Runtime Environment

JSP Java Server Page

M

MLDR Modèl Logique de Ddonnées Relationnelles

MySQL My Structured Query Language

P

PC Personnel Computer

PHP Personal Home Page

S

SD Sequence de Diagram

SGBD System Gestion Base Données

SGBDR System de Gestion de Base Données Relationnelles

SQL Structured Query Language

U

UML Unified Modeling Language

UP Unified de Process

W

WSW Websphere Studio Workbench

WWW World Wide Web

TABLE DES MATIÈRES

Table des Matières	i
Liste des tableaux	v
Table des figures	vi
Introduction Générale	1
1 ETUDE PRELIMINAIRE ET CAPTURE DES BESOINS	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.2.1 Présentation de l'hôtel CRISTAL	3
1.2.2 L'organigramme général de l'hôtel	4
1.2.2.1 Définition d'organigramme	4
1.2.2.2 Intérêt de l'organigramme	5
1.2.3 Installation informatique existante	5
1.3 Etude de l'existant	6
1.3.1 Etude des postes de travail	6
1.3.2 Etude des documents	12
1.3.3 La codification existante	12
1.4 Présentation de l'étude	15
1.4.1 Présentation du sujet	15
1.4.2 Problématique	15
1.4.3 Objectif de l'étude	15
1.5 Cahier de charges	16
1.5.1 Besoins fonctionnels	16

1.5.2	Besoins non fonctionnels	17
1.6	Conclusion	18
2	ANALYSE DES BESIONS	19
2.1	Introduction	19
2.2	Acteurs du système	19
2.2.1	Diagramme de contexte du système à réaliser	20
2.2.2	Identification des messages échangés	21
2.2.3	Identification des cas d'utilisation	22
2.2.3.1	Description textuel des cas d'utilisation	23
2.2.3.2	Diagramme de cas d'utilisation	32
2.2.4	Modélisations dynamique	33
2.2.4.1	Principes et définitions de base	33
2.2.5	Les diagrammes de séquence de l'application à réaliser	34
2.2.5.1	Digramme de séquence du cas d'utilisation " Authentification "	34
2.2.5.2	Digramme de séquence du cas d'utilisation " Effectuer une re- cherche "	35
2.2.5.3	Digramme de séquence du cas d'utilisation " Gérer les clients "	36
2.2.5.4	Digramme de séquence du cas d'utilisation " Gérer les réserva- tions "	37
2.2.5.5	Digramme de séquence du cas d'utilisation " Gérer les utiliza- teurs "	38
2.2.5.6	Digramme de séquence du cas d'utilisation " Gérer les chambres "	39
2.3	Conclusion	40
3	ANALYSE DE DOMAINE ET CONCEPTION	41
3.1	Introduction	41
3.2	Diagrammes de séquence d'interactions	41
3.2.1	Diagramme de séquence d'interaction du cas d'utilisation S'authentification	42
3.2.2	Diagramme de séquence d'interaction du cas d'utilisation "Effectuer une recherche"	42
3.2.3	Diagramme de séquence d'interaction du cas d'utilisation Gérer les clients	43
3.2.4	Diagramme de séquence d'interaction du cas d'utilisation Gérer les réser- vations	44
3.2.5	Diagramme de séquence d'interaction du cas d'utilisation Gérer les utili- sateurs	45

3.2.6	Diagramme de séquence d'interaction du cas d'utilisation Gérer les chambres	46
3.2.7	Présentation du diagramme de classes	47
3.2.8	Les concepts d'un diagramme de classe	47
3.2.9	Diagramme de classes	48
3.3	Présentation des propriétés et méthodes de chaque classe	49
3.4	Passage au modèle relationnel	52
3.5	Conclusion	55
4	REALISATION	56
4.1	Introduction	56
4.2	Les outils de développements	56
4.2.1	Wamp Server	56
4.2.2	Définition de MySQL	56
4.2.3	PHPMyAdmin	57
4.2.4	Java Development Kit (JDK)	57
4.2.5	Définition de l'éclipse	57
4.2.6	JDBC	58
4.3	Les langages de programmation utilisés	58
4.3.1	Le langage SQL	58
4.3.2	Définition de Java	59
4.4	Diagramme de déploiement	59
4.4.1	Diagramme de déploiement d'application réalisé	60
4.5	Arborescence de l'application	60
4.6	Présentation des interfaces de l'application	61
4.6.1	Page d'accueil de l'application	61
4.6.2	formulaire d'authentification	62
4.6.3	Inteface gestion des utilisateurs	63
4.6.4	Inteface gestion des chambres	63
4.6.5	Inteface gestion des clients	64
4.6.6	Inteface gestion des factures	65
4.6.7	Inteface gestion des services	66
4.7	Conclusion	67
	Bibliographie	viii
	ANNEXE	x

A	Diagramme de séquence	xi
A.0.0.1	Diagramme de séquence du cas d'utilisation " Gérer les consommations "	xi
A.0.0.2	Diagramme de séquence du cas d'utilisation " Gérer les factures "	xii
A.0.0.3	Diagramme de séquence du cas d'utilisation " Gestion les catégories "	xiii
A.0.0.4	Diagramme de séquence du cas d'utilisation " Gérer les services "	xiv
B	Diagramme de séquence d'interaction	xv
B.0.1	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les consommations "	xv
B.0.2	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les factures "	xvi
B.0.3	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les catégories "	xvii
B.0.4	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les services "	xviii
C	Interface de l'application	xix
C.0.5	Inteface gestion des catégories	xix
C.0.6	Inteface gestion des consommations	xx
C.0.7	Inteface gestion des réservations	xxi

LISTE DES TABLEAUX

1.1	Représente les caractéristiques d'un ordinateur.	5
1.2	Représente les caractéristiques d'une Imprimante.	6
1.3	Représente les tâches exécutées dans le poste 01.	7
1.4	Représente les documents entrants dans le poste 01.	7
1.5	Représente les tâches exécutées dans le poste 02.	8
1.6	Représente les documents entrants dans le poste 02.	8
1.7	Représente les documents à remplir dans le poste 02.	8
1.8	Représente les documents sortants du le poste 02.	9
1.9	Représente les fichiers utilisés dans le poste 02.	9
1.10	Représente les tâches exécutées dans le poste 03.	9
1.11	Représente les Documents sortants du le poste 03.	10
1.12	Représente les tâches exécutées dans le poste 04.	10
1.13	Représente les Documents sortants du le poste 04.	11
1.14	Représente les tâches exécutées dans le poste 05.	11
1.15	Représente les Les Documents à remplir dans le poste 05.	11
1.16	Représente les Les Documents sortants du le poste 05.	12
2.1	Représente les différentes tâches associées aux acteurs du système.	20
2.2	Représente les messages entrants et message sortants échangés avec le système.	21
2.3	Représente les cas d'utilisations associés au système à développer.	23
2.4	Représente le formalisme de description des cas d'utilisations.	23
2.5	Représente la description du cas d'utilisation " S'authentifier ".	24
2.6	Représente la description du cas d'utilisation " Gérer les clients ".	25
2.7	Représente la description du cas d'utilisation " Gérer les réservations ".	26
2.8	Représente la description du cas d'utilisation " Gérer les consommations ".	27

2.9	Représente la description du cas d'utilisation " Gérer les factures ".	28
2.10	Représente la description du cas d'utilisation " Effectuer une recherche ".	28
2.11	Représente la description du cas d'utilisation " Gérer les utilisateurs ".	30
2.12	Représente la description du cas d'utilisation " Gérer les catégories ".	30
2.13	Représente la description du cas d'utilisation " Gérer les chambres ".	31
2.14	Représente la description du cas d'utilisation " Gérer les services ".	32
3.1	Représente les propriétés et méthodes de chaque classe.	51
4.1	Représente les principales requêtes SQL.	59

TABLE DES FIGURES

1.1	hôtel CRISTAL.	4
1.2	Organigramme général de l'hôtel CRISTAL.	5
1.3	Codifications du numéro d'une pièce d'identité.	13
1.4	Codifications du numéro d'une catégorie chambre.	13
1.5	Codifications du code d'un client envoyé par une entreprise.	13
1.6	Codifications du numéro d'une facture.	14
1.7	Codifications du numéro d'une récapitulation restaurant (LUNGH-DINER).	14
1.8	Codifications du numéro d'un registre journalier des communications.	14
1.9	Codifications du numéro d'une table.	15
2.1	La relation entre les acteurs.	20
2.2	Diagramme de contexte du système à réaliser.	21
2.3	Diagramme de contexte utilisateur.	23
2.4	Diagramme de contexte réceptionniste.	24
2.5	Diagramme de contexte administrateur.	29
2.6	Diagramme global de cas d'utilisations du processus de gestion d'hôtel.	32
2.7	Formalisme général d'un diagramme de séquence.	33
2.8	Diagramme de séquence du cas d'utilisation " S'authentifier ".	35
2.9	Diagramme de séquence du cas d'utilisation " Effectuer une recherche ".	35
2.10	Diagramme de séquence du cas d'utilisation " Gérer les clients ".	36
2.11	Diagramme de séquence du cas d'utilisation " Gérer les réservations ".	37
2.12	Diagramme de séquence du cas d'utilisation " Gérer les utilisateurs ".	38
2.13	Diagramme de séquence du cas d'utilisation " Gérer les chambres ".	39
3.1	Diagramme de séquence d'interaction du cas d'utilisation "S'authentification".	42
3.2	Diagramme de séquence d'interaction du cas d'utilisation "effectuer une recherche".	42

3.3	Diagramme de séquence d'interaction du cas d'utilisation "Gérer les clients". . .	43
3.4	Diagramme de séquence d'interaction du cas d'utilisation "Gérer les réservations".	44
3.5	Diagramme de séquence d'interaction du cas d'utilisation "Gérer les utilisateurs".	45
3.6	Diagramme de séquence d'interaction du cas d'utilisation "Gérer les chambres".	46
3.7	Diagramme de classe de l'application à réalisé.	49
3.8	Transformation des classes.	52
3.9	Association un-à-plusieurs.	52
3.10	Transformation de l'héritage.	53
4.1	Notation du diagramme de déploiement.	60
4.2	Diagramme de déploiement d'application réalisé.	60
4.3	Arborescence de l'application.	61
4.4	Page d'accueil de l'application.	62
4.5	Formulaire d'authentification.	62
4.6	Interface de gestion des utilisateurs.	63
4.7	Interface de gestion des chambres.	64
4.8	Interface de gestion des clients.	65
4.9	Interface de gestion des factures.	66
4.10	Interface de gestion des service.	67
A.1	Diagramme de séquence du cas d'utilisation " Gérer les consommations ".	xi
A.2	Diagramme de séquence du cas d'utilisation " Gérer les factures ".	xii
A.3	Diagramme de séquence du cas d'utilisation " Gérer les catégories ".	xiii
A.4	Diagramme de séquence du cas d'utilisation " Gérer les services ".	xiv
B.1	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les consommations ".	xv
B.2	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les factures ".	xvi
B.3	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les catégories ".	xvii
B.4	Diagramme de séquence d'interaction du cas d'utilisation " Gérer les services ".	xviii
C.1	Interface de gestion des catégories.	xix
C.2	Interface de gestion des consommations.	xx
C.3	Interface de gestion des réservations.	xxi

INTRODUCTION GÉNÉRALE

Dès l'apparition de l'informatique, l'homme a cherché à concevoir des langages lui permettant de communiquer avec l'ordinateur et de concevoir des traitements attendus de celui-ci avec autant d'aisance qu'on le fait avec le langage naturel. Beaucoup de langage ont eu du succès dans différents domaine d'application et ont permis de produire des logiciels spécifiques ou généraux assez puissants. Or, le déploiement rapide de l'informatique vers des secteurs s'accroissant sans cesse et les besoins en rapidité de développement ont fait atteindre leurs limites aux langages de programmation impératifs ou procéduraux. En effet les problèmes tels que la redondance, l'imbrication et la complexité ont rendu la tâche difficile aux développeurs. Pour palier à ces problèmes, sont apparus les langages orientés objets.

Dans le domaine hôtellerie, la vitesse de traitement des réservations et la gestion des besoins des clients sont fondamentales dans la stratégie commerciale, c'est pour cela que l'automatisation et l'informatisation de la gestion d'un hôtel est devenue indispensable.

Dans le cadre de notre étude, nous avons été affecté au sein de l'hôtel CRISTAL de Béjaïa, dans le but d'étudier, analyser, concevoir et à réaliser une application réseau de domaine " GESTION DE L'HOTEL ".

Notre application doit permettre aux utilisateurs de l'hôtel une gestion complète de ce dernier. Afin d'atteindre notre objectif par rapport à ce projet, nous avons fragmenté notre processus de développement en quatre étapes, un chapitre est réglé chaque une de ces étapes dans ce mémoire.

Le premier chapitre "*Etude Préliminaire et Capture des Besoins*" sera consacré à la présentation de l'organisme d'accueil de l'hôtel, ainsi d'avoir une vue globale du système

existant, de détecté tous les problèmes que l'hôtel confronte sans une application et proposer une solution.

Le deuxième chapitre sera dédié à la partie "*Analyse des Besoins*" de notre application qui est le noyau de notre travail. Où nous recenserons les acteurs qui interagissent avec le système à développer puis nous décrivons les besoins de chaque acteur sous forme de cas d'utilisation. Et aussi, pour chaque cas d'utilisation, nous établirons le diagramme de séquence dont l'objectif est de représenter les interactions entre les objets du système en indiquant la chronologie des échanges.

Le troisième chapitre sera focalisé sur l' "*Analyse du Domaine et Conception*" de notre application. En effet, nous réaliserons les diagrammes de séquence d'interactions associés à chaque cas d'utilisations. Ensuite, nous établirons un modèle statique représenté par le diagramme de classes associé à notre système. Enfin, nous déduirons le modèle relationnel par l'application des règles de passage, ce qui nous permettra d'avoir un aperçu sur la base de données.

Le quatrième chapitre concerne la "*Réalisation*" du projet. Dans lequel nous définirons les outils de développement que nous utiliserons. Puis, nous illustrerons quelques interfaces de l'application à développer.

Enfin, nous conclurons ce travail par un résumé de toutes les connaissances acquises pendant le développement de l'application.

CHAPITRE 1

ETUDE PRELIMINAIRE ET CAPTURE DES BESOINS

1.1 Introduction

Dans ce chapitre, nous commençons en premier lieu, par la présentation de l'organisme d'accueil, qui est l'hôtel CRISTAL, dont nous présentons son organigramme et sa situation informatique. Ensuite, nous étudions les postes de travail et les documents entrants et sortants. En dernier lieu, nous présentons notre sujet.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation de l'hôtel CRISTAL

Hôtel CRISTAL est un nouvel hôtel, construit en plein cœur de la ville de BEJAIA, sa construction revient à 1999, il a été exploité en 2003, sa structure est de sept étages dont le dernier étage est réservé au service de restauration, de 60 couverts. La réception et salle d'attente au premier étage et on trouve la cafétéria et service comptabilité au rez-de-chaussée et les autres pour les chambres.

L'hôtel CRISTAL est un hôtel qui abrite de 12 appartements dont 6 appartements F3 et 6 autres F4 et 5 suites équipées de 3 lits séparés, 3 chambres dotées d'un lit à 2 place et d'un à une place. Salles de bain, télévision, un port RJ45, Wifi, etc. Plusieurs salles et salons sont à la disposition du client et un ascenseur s'ajoute, facilite l'accès aux différents niveaux.

L'hôtel se dispose de plusieurs services :

- Réception
- Restauration
- Cafétéria

- Parking
- Blanchisserie
- Téléphone
- Room-services
- Pizzeria



FIGURE 1.1 – hôtel CRISTAL.

1.2.2 L'organigramme général de l'hôtel

1.2.2.1 Définition d'organigramme

C'est une représentation graphique de la structure interne de l'entreprise en termes de départements, services et emplois. Dans l'hôtellerie, l'organisation du travail est découpée en "département" et "services" [1].

L'organigramme fait apparaître pour chaque emploi :

- La dénomination du poste.
- La position hiérarchique.
- Il permet de visualiser les :

Liaisons hiérarchiques.

Liaisons fonctionnelles.

- Définition : Liaison hiérarchique Il s'agit de la liaison entre un supérieur (chef de service) et un subalterne (employé). Elle est unique car un employé ne peut recevoir d'ordre que d'une seule personne, son supérieur hiérarchique.
- Définition : Liaison fonctionnelle Liaison établie entre les services pour des raisons de compétences techniques ou opérationnelles. Exemple : La femme de chambre (service des étages) a besoin de connaître les chambres en départ pour les nettoyer (réception).

1.2.2.2 Intérêt de l’organigramme

L’organigramme :

- Fixe les responsabilités.
- Renseigne le personnel sur :
 - Sa place et sa fonction dans l’entreprise.
 - Ses relations avec son service et les autres services de l’établissement.
- Détermine les services (clés).
- Permet d’identifier les principales activités de l’entreprise.
- C’est un outil vis-à-vis de l’extérieur : par exemple pour les fournisseurs.

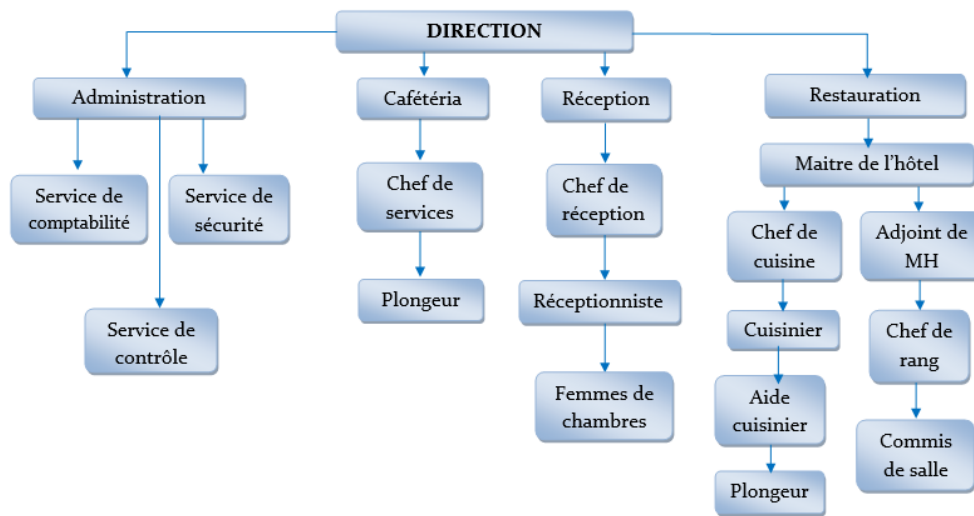


FIGURE 1.2 – Organigramme général de l’hôtel CRISTAL.

1.2.3 Installation informatique existante

Chacun des postes de travail dispose de la même gamme de matériel informatique. Voici la situation informatique résumée dans les tableaux ci-dessous :

● 05 Micro-ordinateurs

Désignation	Capacité Mémoire	Vitesse D’horloge	Capacité Disque dur	Système D’exploitation	Poste de travail
MAXIPOWER	2 GO	2.80 HZ	250 GO	Windows XP	Chef de service

TABLE 1.1 – Représente les caractéristiques d’un ordinateur.

- **05 Imprimantes**

Type	Poste de travail
Laser jet 4300 Canon	Ensemble des employés du service.

TABLE 1.2 – Représente les caractéristiques d’une Imprimante.

- **Logiciels existants**

Microsoft office 2010 : logiciels bureautiques.

Système d’exploitation : Windows 7.

- **05 Onduleurs 220 VA.**

- **02 Scanners.**

- **03 Photocopieuses.**

1.3 Etude de l’existant

Le but de l’étude de l’existant est de connaître le cheminement des informations interne et externe et de prendre connaissance dans le domaine dans lequel un établissement souhaite améliorer son fonctionnement. Lors de cette étape toutes les procédures existantes doivent être étudiées.

1.3.1 Etude des postes de travail

Le but de l’étude des postes de travail est de déceler les postes surchargés ainsi que les principaux défauts de l’organisation existante.

- **Fiche d’étude du poste N°1**

a - Description

Le responsable du poste : Directeur.

Service de rattachement : La direction.

Rôle : Gestion de l’hôtel.

Le nombre de personne qui occupe le poste : 01.

Matériel utilisé : Pc + imprimante + scanner.

b - Les tâches exécutées

N°	Désignation	fréquence
01	Gérer l'hôtel	Chaque jours

TABLE 1.3 – Représente les tâches exécutées dans le poste 01.

c - Les documents entrants

Provenance	Désignation	fréquence	Nombre d'exemplaires
Chef de Réception	Récapitulation du caissier de la réception, Récapitulation restaurant	Chaque jours	01

TABLE 1.4 – Représente les documents entrants dans le poste 01.

● Fiche d'étude du poste N°2**a - Description**

Le responsable du poste : Chef de réception.

Service de rattachement : La réception.

Rôle : Gérer la brigade de la réception, la formation et la réception des clients.

Dépendance hiérarchique : La direction.

Le nombre de personne qui occupe le poste : 02.

Matériel utilisé : Pc + imprimante + photocopieuse.

b - Les tâches exécutées

N°	Désignation	fréquence	Délai
01	Réception du client	Arrivé du client	Aléatoire
02	Enregistrement des informations sur le client	Arrivé du client	Aléatoire
03	Etablissement de la fiche et le registre de police	Arrivé du client	Aléatoire
04	Vérifier la disponibilité des chambres	A chaque demande de réservation	Aléatoire
05	Etablissement de la facture et son règlement	Départ du client	Aléatoire

06	Envoyer la FP	Chaque jour	Aléatoire
07	Envoyer le registre police	Chaque fin du mois	Aléatoire

TABLE 1.5 – Représente les tâches exécutées dans le poste 02.

c - Les documents entrants

Provenance	Désignation	fréquence	Nombre d'exemplaires
Client	Fiche client	Arrivé du client	01
Maître d'hôtel	Récapitulation restaurant	Chaque fin du jour	01
Chaque service	Récapitulation du chaque service	Chaque fin de service	01

TABLE 1.6 – Représente les documents entrants dans le poste 02.

c - Les documents à remplir

Désignation	fréquence	Nombre d'exemplaires
Fiche police	Arrivé du client	01
Registre police	Arrivé du client	01
Facture	Départ du client	02
Fiche statistique	Chaque fin du mois	01
Récapitulation du caissier	Chaque jour	01
Récapitulation restaurant	Chaque fin du jour	01

TABLE 1.7 – Représente les documents à remplir dans le poste 02.

d - Les documents sortants

Désignation	fréquence	Destinataire	Nombre d'exemplaires
Fiche police	Chaque jour	Police	02
Registre police	Chaque fin du mois	Police	01
Facture	Fin de séjour	Client	01
Fiche statistique	Fin du mois	Direction de tourisme	01
Récapitulation du caissier	Chaque jour	Directeur	01
Récapitulation restaurant	Chaque jour	Directeur	01

TABLE 1.8 – Représente les documents sortants du le poste 02.

e - Fichiers utilisés pour ce poste

Désignation	Support	Localisation	Opération réalisées
Fichier client	Registre	Réception	Mise à jour Consultation
Facture	Papier	Réception	Facturation

TABLE 1.9 – Représente les fichiers utilisés dans le poste 02.

● Fiche d'étude du poste N°3**a - Description**

Le responsable du poste : Maître de l'hôtel.

Service de rattachement : Restauration.

Rôle : La gestion de restauration.

Dépendance hiérarchique : La direction.

Le nombre de personne qui occupe le poste : 04.

Matériel utilisé : Pc + imprimante.

b - Les tâches exécutées

N°	Désignation	fréquence
01	Récapitulation restaurant (Lunch-Diner)	Chaque commande

TABLE 1.10 – Représente les tâches exécutées dans le poste 03.

c - Les Documents sortants

Désignation	Destination	fréquence	Nombre d'exemplaires
Récapitulation restaurant	Réception	Chaque commande	01
Récapitulation restaurant	Client	Chaque fin de service	01

TABLE 1.11 – Représente les Documents sortants du le poste 03.

● Fiche d'étude du poste N°4**a - Description**

Le responsable du poste : Chef de rang.

Service de rattachement : Restauration.

Rôle : S'occupe des clients dans un rang.

Dépendance hiérarchique : Maître de l'hôtel.

Le nombre de personne qui occupe le poste : 03.

Matériel utilisé : Pc + imprimante.

b - Les tâches exécutées

N°	Désignation	fréquence
01	Etablir les bons de commande concernant les clients de son rang	Chaque commande
02	Servir les plats commandés aux clients	Chaque commande

TABLE 1.12 – Représente les tâches exécutées dans le poste 04.

c - Les Documents sortants

Désignation	Destination	fréquence	Nombre d'exemplaires
Bons de suite d'entrées	Maître d'hôtel	Chaque commande	01
Bons de suite de boissons	Chef de cuisine	Chaque commande	01
Bons de suite d'entrées	Maître d'hôtel	Chaque commande	01

Bons de suite de boissons	Chef de cuisine	Chaque commande	01
---------------------------	-----------------	-----------------	----

TABLE 1.13 – Représente les Documents sortants du le poste 04.

- **Fiche d'étude du poste N°5**

- a - **Description**

- Le responsable du poste : Chef de service.

- Service de rattachement : Cafétéria.

- Rôle : La gestion de la cafétéria.

- Dépendance hiérarchique : La direction.

- Le nombre de personne qui occupe le poste : 01.

- b - **Les tâches exécutées**

N°	Désignation	fréquence
01	Gérer la cafétéria	Chaque jour

TABLE 1.14 – Représente les tâches exécutées dans le poste 05.

- c - **Les Documents à remplir**

Désignation	fréquence	Nombre d'exemplaires
Facture de service	Chaque jour	01

TABLE 1.15 – Représente les Les Documents à remplir dans le poste 05.

- d - **Les Documents sortants**

Désignation	Destination	fréquence	Nombre d'exemplaires
Facture de service	Réception	Chaque fin de service	01

TABLE 1.16 – Représente les Les Documents sortants du le poste 05.

1.3.2 Etude des documents

- **Etude du document N° : 01**

Code : FC.

Désignation : Fiche Client.

Nature de document : interne.

Par qui ce document est-t-il rempli : Réceptionniste.

Pour qui est-t-il destiné : Réceptionniste.

Nombre d'exemplaire : 01.

- **Etude du document N° : 02**

Code : FP.

Désignation : Fiche Police.

Nature de document : interne.

Par qui ce document est-t-il rempli : Réceptionniste.

Pour qui est-t-il destiné : Police.

Nombre d'exemplaire : 01.

- **Etude du document N° : 03**

Code : F.

Désignation : Fiche Client.

Nature de document : Facture.

Par qui ce document est-t-il rempli : Réceptionniste.

Pour qui est-t-il destiné : Client.

Nombre d'exemplaire : 02.

1.3.3 La codification existante

Chaque étude informatique dans le centre de gestion conduit aux choix du système technique de codification qui permet l'interprétation de l'information littéraire à une représentation abrégée qui utilise certain nombre de symboles.

Le but de cette technique est la simplicité, la facilité, la conception et la souplesse de l'élément codifié.

Voici quelques codifications existantes :

– **N° de pièce d'identité :**

Exemple : 346063.

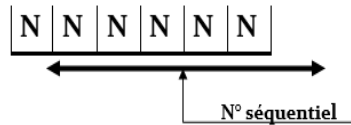


FIGURE 1.3 – Codifications du numéro d'une pièce d'identité.

– **N° de catégorie chambre :**

Exemple 1 : 201 veut dire Etage N° 2 et Suite N° 01.

Exemple 2 : 302 veut dire Etage N° 3 et Appartement F3 N° 02.

Exemple 3 : 603 veut dire Etage N° 6 et Appartement F4 N° 03.

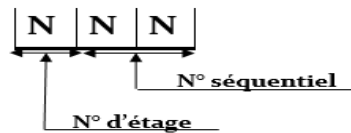


FIGURE 1.4 – Codifications du numéro d'une catégorie chambre.

– **Code de client envoyé par une entreprise :**

Exemple : SONATRACH.

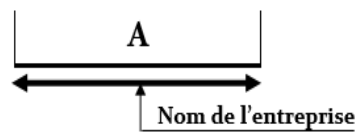


FIGURE 1.5 – Codifications du code d'un client envoyé par une entreprise.

– **N° de la facture :**

Exemple : N° 000940.

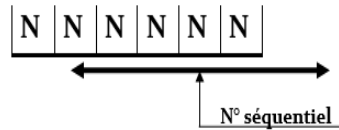


FIGURE 1.6 – Codifications du numéro d’une facture.

– **N° de la récapitulation restaurant (LUNGH-DINER) :**

Exemple : N° 000131.

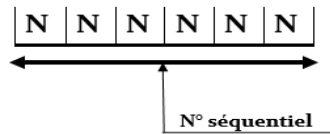


FIGURE 1.7 – Codifications du numéro d’une récapitulation restaurant (LUNGH-DINER).

– **N° de registre journalier des communications :**

Exemple : N° 000720.

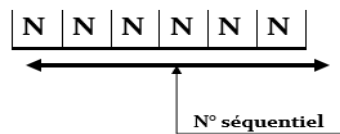


FIGURE 1.8 – Codifications du numéro d’un registre journalier des communications.

– **N° de la table :**

Exemple : N° 12.

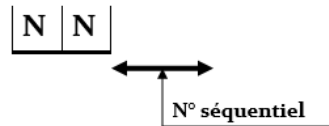


FIGURE 1.9 – Codifications du numéro d’une table.

1.4 Présentation de l’étude

1.4.1 Présentation du sujet

Les informations circulantes dans l’hôtel sont de nature diverse, elles concernent le client depuis son arrivée jusqu’à son départ de l’hôtel, et pour le bon fonctionnement de travail, nous avons pensé à mettre en place un système informatique qui facilitera les tâches de gestion de l’hôtel. L’application à mettre en œuvre permettra une interaction rigoureuse entre le réceptionniste, le gérant et les clients, en assurant ainsi une bonne gestion de l’hôtel.

1.4.2 Problématique

Les réservations à l’hôtel " CRISTAL " sont ouvertes pendant toute l’année aux clients Algériens ou étrangers voulant séjourner à Bejaia et le nombre de ces derniers augmente d’année en année ce qui engendre cumul d’information, les responsables trouvent des difficultés dans leurs travaux, le taux d’erreur ne cesse d’accroître et certaines tâches font qu’elles sont pénibles à traiter, dont on souligne les problèmes suivants :

- Perte des documents.
- Toutes les procédures sont faites manuellement.
- Mauvais archivage des documents.
- Difficulté et retard lors de la recherche de l’information.
- Perte de temps.

1.4.3 Objectif de l’étude

L’objectif principal de ce travail se résume à la "conception et réalisation d’une application réseaux pour la gestion d’un hôtel".

Après avoir recensé les problèmes existants, les responsables nous ont cités les besoins suivants :

- Assurer le suivi des clients dès leurs arrivés jusqu’à leurs départs.
- Informatiser certaines tâches.
- Améliorer la qualité du service rendu en matière de temps de réponse.

– Minimiser les erreurs et stocker les informations et les protéger pendant une longue durée.

Et tout ça c'est mettre en place une application suffisamment simple d'utilisation afin qu'elle soit manipulable par des personnes novices en informatique.

1.5 Cahier de charges

Notre travail vise à réaliser une application réseau pour la gestion d'un hôtel qui consiste à gérer les clients, les chambres, les services, les factures, ainsi que les comptes, qui seront soumis à un seul acteur. Afin d'atteindre les objectifs, l'application à concevoir offrira les espaces suivant :

- Un espace réservé à l'administrateur de l'application qui lui permet la gestion des clients, la gestion des chambres, la gestion des services, la gestion des factures et la mise à jour de ces derniers (l'ajout, modification, suppression).
- Un espace réservé aux autres utilisateurs de l'application (employés de l'hôtel par exemple le réceptionniste, ..etc) après avoir un compte (login et mot de passe) configuré par l'administrateur et des privilèges pour avoir accès à leurs fonctionnalités.

1.5.1 Besoins fonctionnels

Les besoins fonctionnels se présentent en dix (10) grandes parties :

1. **Authentification** : Cette interface permet à l'utilisateur d'accéder à son interface après une authentification par un login et un mot de passe.
2. **Effectuer une recherche** : Cette interface permet à un utilisateur d'effectuer une recherche avec la saisie d'un mot clé concernant le sujet de la recherche.
3. **Gestion des utilisateurs** : Gestion des utilisateurs
4. **Gestion des clients** : Cette interface offre à l'utilisateur la possibilité de gérer les différents clients de notre application c'est-à-dire de faire des mises à jour sur ces derniers (l'ajout, la modification ou la suppression).
5. **Gestion des factures** : Cette interface offre à l'utilisateur la possibilité de gérer les différentes factures de l'hôtel c'est-à-dire de faire des mises à jour sur ces derniers (l'ajout, la modification ou la suppression).
6. **Gestion des réservations** : Cette interface offre à l'utilisateur la possibilité de gérer les différentes réservations de l'hôtel effectuées par les clients c'est-à-dire de faire des mises à jour (l'ajout, la modification ou la suppression) sur ces réservations.
7. **Gestion des chambres** : Cette interface offre à l'administrateur la possibilité de gérer les différentes chambres d'hôtel c'est-à-dire de faire des mises à jour sur ces derniers (l'ajout, la modification ou la suppression).

8. **Gestion des services** : Cette interface offre à l'administrateur la possibilité de gérer les différents services offerts par l'hôtel c'est-à-dire de faire des mises à jour sur ces derniers (l'ajout, la modification ou la suppression).
9. **Gestion des catégories** : Cette interface offre à l'administrateur la possibilité de gérer les différentes catégories des chambres de l'hôtel c'est-à-dire de faire des mises à jour sur ces derniers (l'ajout, la modification ou la suppression).
10. **Gestion des consommations** : Cette interface offre à l'utilisateur la possibilité de gérer les différentes consommations des clients aux services de l'hôtel c'est-à-dire de faire des mises à jour (l'ajout, la modification ou la suppression) sur ces consommations.

1.5.2 Besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes [2] :

1. **Fiabilité** : L'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.
2. **Les erreurs** : Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre application.
3. **Ergonomie et bonne Interface** : L'application doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.
4. **Sécurité** : Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans les applications.
5. **Aptitude à la maintenance et la réutilisation** : Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.
6. **Compatibilité et portabilité** : Une application quel que soit son domaine, son éditeur et son langage de programmation ne peut être fiable qu'avec une compatibilité avec toutes les plateformes et tous les systèmes.

1.6 Conclusion

Dans ce chapitre nous avons commencé par présenter l'hôtel " CRISTAL " et sa structure, puis nous avons étudié des postes de travail de chaque service de l'organisme d'accueil, ensuite nous avons mis l'accent sur la problématique, générant ainsi les objectifs à atteindre.

Dans le chapitre suivant, nous allons entamer la partie qui consiste à analyser nos besoins et de les appliquer sur notre conception.

CHAPITRE 2

ANALYSE DES BESIIONS

2.1 Introduction

Dans ce chapitre nous nous intéressons à l'étude de conception de notre application, nous adoptons l'UML comme langage de modélisation, et le processus unifié UP comme démarche. Cela consiste à présenter les diagrammes de cas d'utilisation décrivant les scénarios nominaux de chaque acteur ainsi que les diagrammes de séquence qui représentent les interactions entre ces acteurs et le système.

2.2 Acteurs du système

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 ou en recevant des messages susceptibles d'être porteurs de données [3].

Dans notre cas, Nous avons trois acteurs qui sont : Utilisateur, Administrateur et Réceptionniste.

La relation entre les acteurs : relation d'héritage.

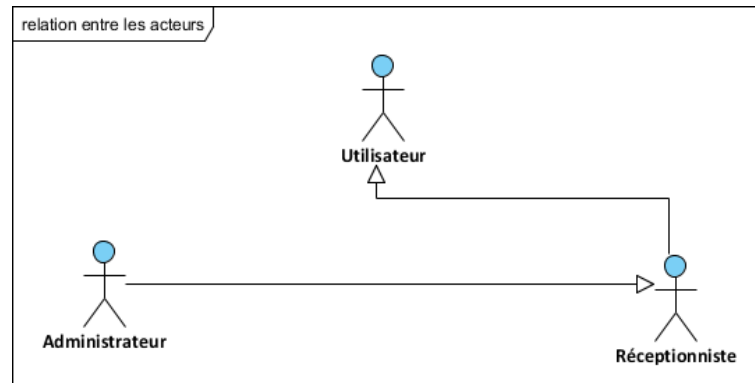


FIGURE 2.1 – La relation entre les acteurs.

Description des acteurs :

Protocoles	Connaissance
Utilisateurs	L'utilisateur représente tout acteur non authentifié par le système. Il peut effectuer la tâche d'authentification, celle qui lui permet d'être un administrateur ou un réceptionniste.
Réceptionniste	Réceptionniste a un accès au système via un contrôle d'accès (login et mot de passe). Les opérations qu'il peut effectuer sont : gérer des consommations, gérer des clients, gérer des factures, gérer des réservations, effectuer une recherche.
Administrateur	L'administrateur a un accès au système via un contrôle d'accès (login et mot de passe). Les opérations qu'il peut effectuer sont : gérer des utilisateurs, gérer des catégories, gérer des chambres, gérer des services, gérer des consommations, gérer des clients, gérer des factures, gérer des réservations, effectuer une recherche.

TABLE 2.1 – Représente les différentes tâches associées aux acteurs du système.

2.2.1 Diagramme de contexte du système à réaliser

La figure ci-dessous montre l'interaction entre les différents acteurs et le système que nous allons mettre en place :

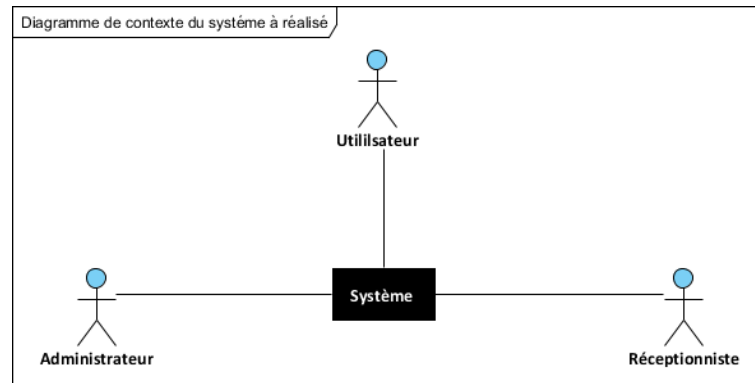


FIGURE 2.2 – Diagramme de contexte du système à réaliser.

2.2.2 Identification des messages échangés

Un message est un moyen de communication entre acteurs. Il définit un événement, c'est-à-dire une information envoyée à un acteur et provoquant en réponse le commencement d'actions associées à ce dernier.

Les différents acteurs de notre système, peuvent échanger des messages qu'ils soient entrants ou sortants. En effet, les messages entrants représentent les demandes qu'un acteur effectue tandis que les messages sortants représentent la réponse du système à une demande donnée. Ces messages seront utilisés par la suite dans les diagrammes de séquence [4].

Acteurs	Messages entrants	Message sortants
Utilisateurs	Demande d'authentification	formulaire d'authentification
Réceptionniste	Demande de gestion les clients Demande de gestion les réservations Demande de gestion les consommations Demande de gestion les factures Demande d'effectuer une recherche	Interface de gestion des clients Interface de gestion des réservations Interface de gestion des consommations Interface de gestion des factures Interface de recherche
Administrateur	Demande de gestion les utilisateurs Demande de gestion les catégories Demande de gestion les chambres Demande de gestion les services	Interface de gestion des utilisateurs Interface de gestion des catégories Interface de gestion des chambres Interface de gestion des services

TABLE 2.2 – Représente les messages entrants et message sortants échangés avec le système.

2.2.3 Identification des cas d'utilisation

Les cas d'utilisations sont les outils formels qui permettent de consigner et d'examiner les interactions et les dialogues des utilisateurs (acteurs) avec le système. Un cas d'utilisation est une narration qui décrit un scénario appliqué à une utilisation particulière, dans lequel un acteur fournit des entrées et pour lequel le système produit une sortie observable. Un cas d'utilisation doit exprimer ce que doit faire un acteur, sans préjuger de la façon dont cela sera fait, il décrit le comportement du système vu de l'extérieur. Le tableau ci-après décrit l'ensemble des cas d'utilisation du système à réaliser [4].

N°	Cas d'utilisation		Acteur
01	S'authentifier		utilisateur
02	Effectuer une recherche		utilisateur
03	Gérer les utilisateurs	Ajouter un utilisateur Modifier un utilisateur Supprimer un utilisateur	Administrateur
04	Gérer les catégories	Ajouter une catégorie Modifier une catégorie Supprimer une catégories	Administrateur
05	Gérer les chambres	Ajouter une chambre Modifier une chambre Supprimer une chambre	Administrateur
06	Gérer les services	Ajouter un service Modifier un service Supprimer un service	Administrateur
07	Gérer les clients	Ajouter un client Modifier un client Supprimer un client	Administrateur et Réceptionniste
08	Gérer les réservations	Ajouter une réservation Modifier une réservation Supprimer une réservation	Administrateur et Réceptionniste
09	Gérer les consommations	Ajouter une consommation Modifier une consommation Supprimer une consommation	Administrateur et Réceptionniste

10	Gérer les factures	Ajouter une facture Modifier une facture Supprimer une facture	Administrateur et Réceptionniste
----	--------------------	--	--

TABLE 2.3 – Représente les cas d'utilisations associés au système à développer.

2.2.3.1 Description textuel des cas d'utilisation

A chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs. Pour exprimer les cas d'utilisations de notre système, nous avons choisi le formalisme suivant [5] :

Numéro du cas d'utilisation	Nom du cas d'utilisation
Résumé	But du cas d'utilisation.
Acteurs	Acteurs participants au cas d'utilisation.
Pré-condition	Condition qui doit être remplie avant le début du cas d'utilisation.
Scénario nominal	Séquence d'actions normales associées au cas d'utilisation.
Alternative	Séquence d'actions alternatives pouvant conduire également à un succès.
Exception	Séquences d'actions conduisant à un échec.

TABLE 2.4 – Représente le formalisme de description des cas d'utilisations.

a - Les cas d'utilisations concernant l'utilisateur :

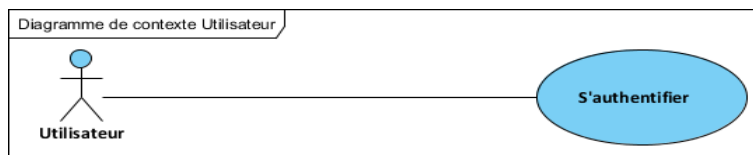


FIGURE 2.3 – Diagramme de contexte utilisateur.

- Le cas d'utilisation " S'authentifier " :

Cas d'utilisation N° 1	S'authentifier
Résumé	Vérification de l'identité des utilisateurs (Login et mot de passe).
Acteurs	Utilisateur.
Pré-condition	Aucune.
Scénario nominal	[début] 1. Accès à l'application ; 2. Le système affiche le formulaire d'authentification ; 3. L'utilisateur saisit son login et son mot de passe ; -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (4) ; 4. Le système vérifie la validité des informations fournies ; -Si les champs sont incorrects Alors Exécuter l'exception ; -Sinon Aller à (5) ; 5. Le système donne l'accès à l'interface correspondante. [fin]
Alternative Exception	Le système affiche un message d'erreur et réaffiche la boîte d'authentification et attend que l'utilisateur ressaisisse ses informations.

TABLE 2.5 – Représente la description du cas d'utilisation " S'authentifier ".

b - Les cas d'utilisations concernant le réceptionniste

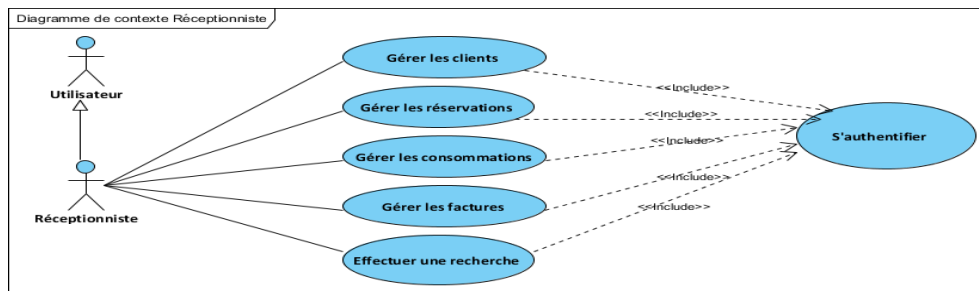


FIGURE 2.4 – Diagramme de contexte réceptionniste.

- Le cas d'utilisation " Gérer les clients " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter un client ;
 - Modifier un client ;
 - Supprimer un client.

Cas d'utilisation N° 7	Gérer les clients
Résumé	Gérer les clients.
Acteurs	Réceptionniste.
Pré-condition	Le réceptionniste s'authentifier.
Scénario nominal	<p>[début]</p> <ol style="list-style-type: none"> 1. Accès à l'application ; 2. S'authentifier ; 3. Le réceptionniste demande le formulaire de gestion des clients ; 4. Le système affiche le formulaire ; 5. Le réceptionniste effectue l'action souhaitée (ajout, modification, suppression) ; <p>Pour l'ajout :</p> <ul style="list-style-type: none"> -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (6) ; <ol style="list-style-type: none"> 6. Confirmer l'action ; <p>[fin]</p>
Alternative Exception	Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.

TABLE 2.6 – Représente la description du cas d'utilisation " Gérer les clients ".

- Le cas d'utilisation " Gérer les réservations " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter une réservation ;
 - Modifier une réservation ;
 - Supprimer une réservation.

Cas d'utilisation N° 8	Gérer les réservations
Résumé	Gérer les réservations.
Acteurs	Réceptionniste.
Pré-condition	Le réceptionniste s'authentifier.

Scénario nominal	[début] 1. Accès à l'application ; 2. S'authentifier ; 3. Le réceptionniste demande le formulaire de gestion des réservations ; 4. Le système affiche le formulaire ; 5. Le réceptionniste effectue l'action souhaitée (ajout, modification, suppression) ; Pour l'ajout : -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (6) ; 6. Confirmer l'action ; [fin]
Alternative Exception	Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.

TABLE 2.7 – Représente la description du cas d'utilisation " Gérer les réservations ".

- Le cas d'utilisation " Gérer les consommations " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter une consommation ;
 - Modifier une consommation ;
 - Supprimer une consommation.

Cas d'utilisation N° 9	Gérer les consommations
Résumé	Gérer les consommations.
Acteurs	Réceptionniste.
Pré-condition	Le réceptionniste s'authentifier.
Scénario nominal	[début] 1. Accès à l'application ; 2. S'authentification ; 3. Le réceptionniste demande le formulaire de gestion des consommations ; 4. Le système affiche le formulaire ;

	<p>5. Le réceptionniste effectue l'action souhaitée (ajout, modification, suppression);</p> <p>Pour l'ajout :</p> <p>-Si les champs sont incomplets Alors Exécuter l'exception;</p> <p>-Sinon Aller à (6);</p> <p>6. Confirmer l'action;</p> <p>[fin]</p>
Alternative Exception	<p>Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.</p>

TABLE 2.8 – Représente la description du cas d'utilisation " Gérer les consommations " .

- Le cas d'utilisation " Gérer les factures " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter une facture;
 - Modifier une facture;
 - Supprimer une facture.

Cas d'utilisation N° 10	Gestion des factures
Résumé	Gérer les factures.
Acteurs	Réceptionniste.
Pré-condition	Le réceptionniste s'authentifier.
Scénario nominal	<p>[début]</p> <p>1. Accès à l'application;</p> <p>2. S'authentifier;</p> <p>3. Le réceptionniste demande le formulaire de gestion des factures;</p> <p>4. Le système affiche le formulaire;</p> <p>5. Le réceptionniste effectue l'action souhaitée (ajout, modification, suppression);</p> <p>Pour l'ajout :</p> <p>-Si les champs sont incomplets Alors Exécuter l'exception;</p> <p>-Sinon Aller à (6);</p> <p>6. Confirmer l'action;</p>

	[fin]
Alternative Exception	Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.

TABLE 2.9 – Représente la description du cas d'utilisation " Gérer les factures ".

- Le cas d'utilisation " Effectuer une recherche " est caractérisé par les trois (03) scénarios suivants :
 - Recherche d'un client ;
 - Recherche d'une réservation ;
 - Recherche d'une consommation ;
 - Recherche d'une facture ;

Cas d'utilisation N° 2	Effectuer une recherche
Résumé	Le réceptionniste peut effectuer une recherche sur les clients, les réservations, les consommations, les factures.
Acteurs	Réceptionniste.
Pré-condition	Le réceptionniste s'authentifier.
Scénario nominal	[début] 1. Accès à l'application ; 2. S'authentifier ; 3. Le réceptionniste saisi le mot clé concerne le sujet de la recherche ; -Si le mot clé n'existe pas Alors Exécuter l'exception ; -Sinon Aller à (4) ; 4. Confirmer l'action ; [fin]
Alternative Exception	Le système affiche une liste vide.

TABLE 2.10 – Représente la description du cas d'utilisation " Effectuer une recherche ".

c - Les cas d'utilisations concernant l'administrateur :

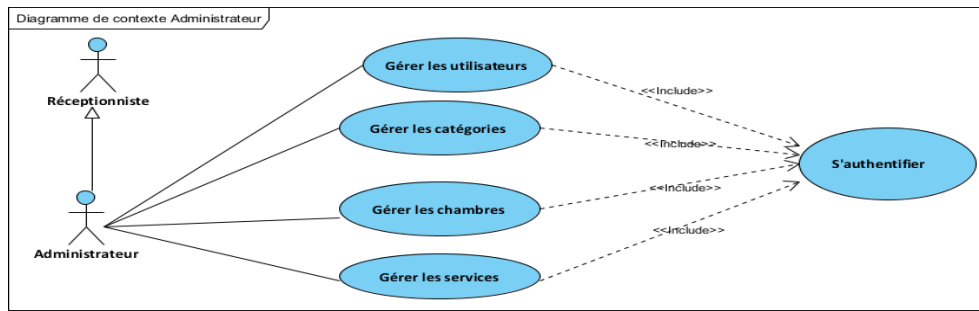


FIGURE 2.5 – Diagramme de contexte administrateur.

- Le cas d'utilisation " Gérer les utilisaturs " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter un utilisateur ;
 - Modifier un utilisateur ;
 - Supprimer un utilisateur.

Cas d'utilisation N° 3	Gérer les utilisaturs
Résumé	Gérer les utilisaturs.
Acteurs	Administrateur.
Pré-condition	L'administrateur s'authentifier.
Scénario nominal	<p>[début]</p> <ol style="list-style-type: none"> 1. Accès à l'application ; 2. S'authentifier ; 3. L'administrateur demande le formulaire de gestion des utilisaturs ; 4. Le système affiche le formulaire ; 5. L'administrateur effectue l'action souhaitée (ajout, modification, suppression) ; <p>Pour l'ajout :</p> <ul style="list-style-type: none"> -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (6) ; <ol style="list-style-type: none"> 6. Confirmer l'action ; <p>[fin]</p>

Alternative Exception	Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.
--------------------------	---

TABLE 2.11 – Représente la description du cas d'utilisation " Gérer les utilisateurs " .

- Le cas d'utilisation " Gérer les catégories " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter une catégorie ;
 - Modifier une catégorie ;
 - Supprimer une catégorie.

Cas d'utilisation N° 4	Gérer les catégories
Résumé	Gérer les catégories.
Acteurs	Administrateur.
Pré-condition	L'administrateur s'authentifier.
Scénario nominal	<p>[début]</p> <ol style="list-style-type: none"> 1. Accès à l'application ; 2. S'authentifier ; 3. L'administrateur demande le formulaire de gestion des catégories ; 4. Le système affiche le formulaire ; 5. L'administrateur effectue l'action souhaitée (ajout, modification, suppression) ; <p>Pour l'ajout :</p> <ul style="list-style-type: none"> -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (6) ; <ol style="list-style-type: none"> 6. Confirmer l'action ; <p>[fin]</p>
Alternative Exception	Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.

TABLE 2.12 – Représente la description du cas d'utilisation " Gérer les catégories " .

- Le cas d'utilisation " Gérer les chambres " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter une chambre ;
 - Modifier une chambre ;
 - Supprimer une chambre.

Cas d'utilisation N° 5	Gestion des chambres
Résumé	Gérer les chambres.
Acteurs	Administrateur.
Pré-condition	L'administrateur s'authentifier.
Scénario nominal	<p>[début]</p> <ol style="list-style-type: none"> 1. Accès à l'application ; 2. S'authentifier ; 3. L'administrateur demande le formulaire de gestion des chambres ; 4. Le système affiche le formulaire ; 5. L'administrateur effectue l'action souhaitée (ajout, modification, suppression) ; <p>Pour l'ajout :</p> <ul style="list-style-type: none"> -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (6) ; <ol style="list-style-type: none"> 6. Confirmer l'action ; <p>[fin]</p>
Alternative Exception	Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.

TABLE 2.13 – Représente la description du cas d'utilisation " Gérer les chambres ".

- Le cas d'utilisation " Gérer des services " est caractérisé par les trois (03) scénarios suivants :
 - Ajouter un service ;
 - Modifier un service ;
 - Supprimer un service.

Cas d'utilisation N° 6	Gérer des services
Résumé	Gérer les services.
Acteurs	Administrateur.
Pré-condition	L'administrateur s'authentifier.

Scénario nominal	<p>[début]</p> <ol style="list-style-type: none"> 1. Accès à l'application ; 2. S'authentifier ; 3. L'administrateur demande le formulaire de gestion des services ; 4. Le système affiche le formulaire ; 5. L'administrateur effectue l'action souhaitée (ajout, modification, suppression) ; <p>Pour l'ajout :</p> <ul style="list-style-type: none"> -Si les champs sont incomplets Alors Exécuter l'exception ; -Sinon Aller à (6) ; <ol style="list-style-type: none"> 6. Confirmer l'action ; <p>[fin]</p>
Alternative Exception	<p>Le système affiche un message d'erreur et réaffiche le formulaire d'ajout et attend que l'utilisateur ressaisisse ses informations.</p>

TABLE 2.14 – Représente la description du cas d'utilisation " Gérer les services ".

2.2.3.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un formalisme permettant de modéliser le fonctionnement d'un système par un découpage en fonctionnalités. Il illustre de plus la nature des interactions avec ces fonctionnalités offertes à titre de services à des acteurs externes au système. Chaque Fonctionnalité est appelée un cas d'utilisation [6].

Le diagramme ci-après représente le diagramme général de cas d'utilisation associés au processus de gestion d'un hôtel.

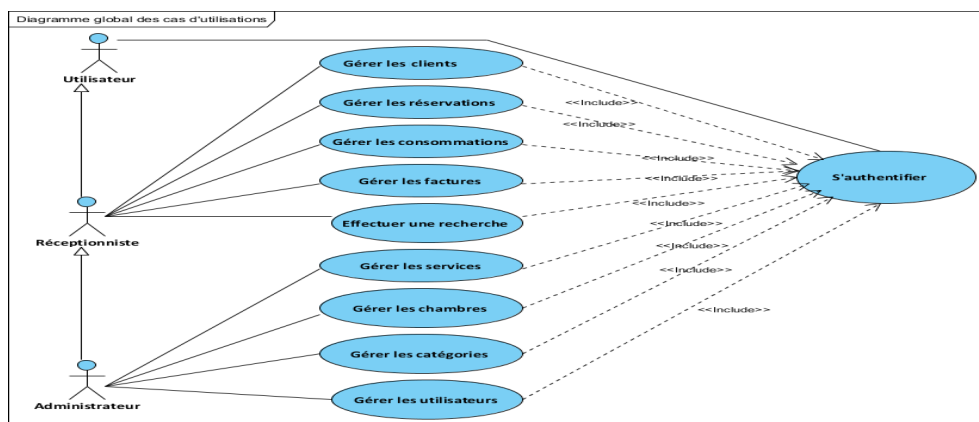


FIGURE 2.6 – Diagramme global de cas d'utilisations du processus de gestion d'hôtel.

2.2.4 Modélisations dynamique

L'objectif de la phase d'analyse, basée sur les diagrammes de séquence, est de décrire d'une manière compréhensible est correcte tous les besoin du client. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation sous forme de scénarios pour mieux faciliter la compréhension, la modification et la maintenance du système a réaliser [7].

2.2.4.1 Principes et définitions de base

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

1- formalisme d'un diagramme de séquence : Un diagramme de séquence se représente globalement dans un grand rectangle avec indication du nom du diagramme en haut à gauche et une abréviation " sd " qui signifie séquence diagramme [8].



FIGURE 2.7 – Formalisme général d'un diagramme de séquence.

2- Objet : Un objet est une instance de la classe ou de l'interface responsable de la réalisation des opérations définies. Il est représenté graphiquement dans le diagramme de séquence par un carré comportant son identifiant [9].

3- Ligne de vie Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour [8].

4- Message synchrone et asynchrone Dans un diagramme de séquence, deux types de messages peuvent être distingués [8] :

- Message synchrone : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.
- Message asynchrone : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

5- Fragment d'interaction Un fragment d'interaction, dit " combiné ", correspond à un ensemble d'interactions auquel on applique un opérateur. Un fragment combiné se représente globalement comme un diagramme de séquence avec indication dans le coin à gauche du nom de l'opérateur. Treize opérateurs ont été définis dans UML : alt, opt, loop, par, strict/weak, break, ignore/ consider, critical, negative, assertion et ref [8].

Nous décrivons dans ce qui suit les opérateurs que nous utilisons dans les diagrammes de séquence de notre projet : alt, opt, loop et ref.

- Opérateur " alt " : il correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon.
- Opérateur " opt " :(optional) il correspond à une instruction de test sans alternative (sinon).
- Opérateur " loop " : il correspond à une instruction de boucle permettant d'exécuter une séquence d'interactions tant qu'une condition est satisfaite.
- Opérateur " ref " : il permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

2.2.5 Les diagrammes de séquence de l'application à réaliser

En se basant sur les opérations décrites précédemment, nous présentons les diagrammes de séquence des cas d'utilisations de l'application que nous allons mettre en œuvre.

2.2.5.1 Diagramme de séquence du cas d'utilisation " Authentification "

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à une entité (généralement un login et un mot de passe). Ces informations sont préétablies dans le système.

Lors de l'authentification de l'utilisateur, deux cas peuvent se présenter : données complètes ou données incomplètes, ce qui explique l'utilisation du premier opérateur " alt ".

Si les données sont incomplètes le système affiche un message d'erreur et réaffiche la page d'authentification, sinon deux cas peuvent se présenter : informations correctes ou informations incorrectes ce qui explique l'utilisation du deuxième opérateur " alt ". Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface appropriée. En revanche, si l'utilisateur saisit des informations incorrectes, le système génère un message d'erreur et réaffiche la page d'authentification.

Ce procédé est exécuté à chaque fois que l'utilisateur tente de s'authentifier, c'est pourquoi nous avons utilisé l'opérateur " Loop ".

L'identification du fragment " Authentification " permet d'alléger les diagrammes de séquence présentés ci-dessous en utilisant le cadre " ref ".

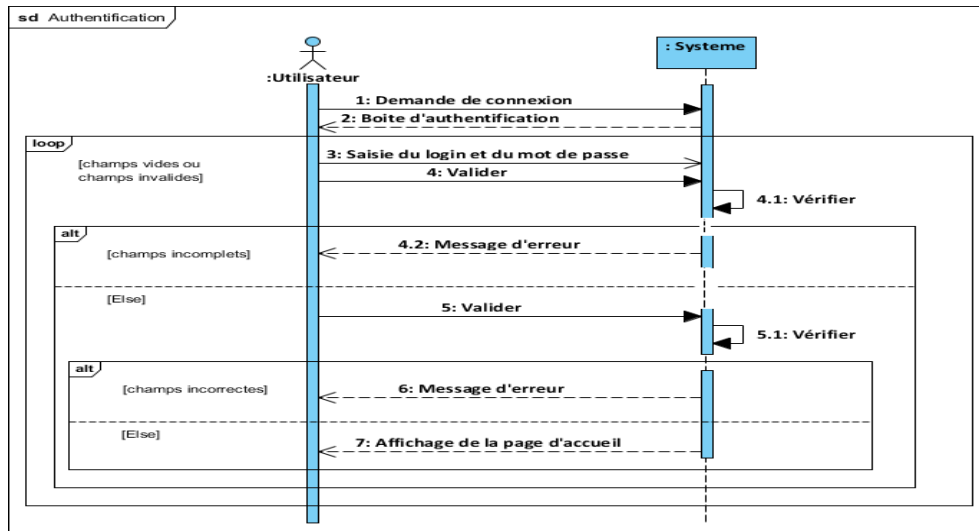


FIGURE 2.8 – Diagramme de séquence du cas d’utilisation " S’authentifier " .

2.2.5.2 Diagramme de séquence du cas d’utilisation " Effectuer une recherche "

Après l’authentification, l’utilisateur peut effectuer une recherche en introduisant un mot clés concernant la recherche. On distingue deux types de réponse : " résultat non trouvé ", dans ce cas le système affiche le message " pas de résultats " ou bien " résultat trouvé " et le système affiche les résultats.

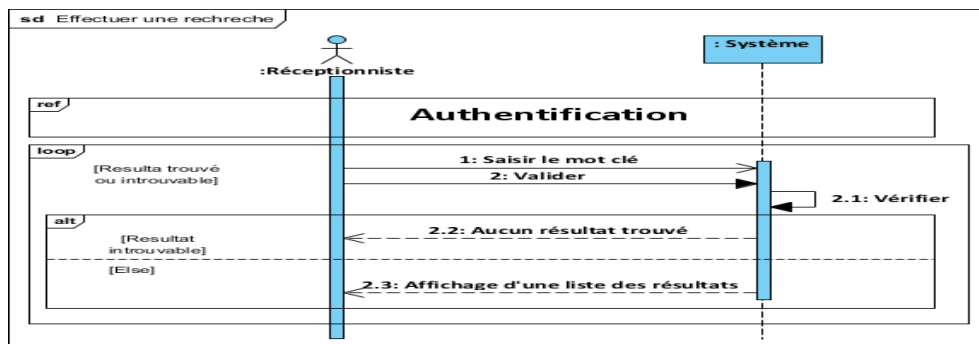


FIGURE 2.9 – Diagramme de séquence du cas d’utilisation " Effectuer une recherche " .

2.2.5.3 Digramme de séquence du cas d'utilisation " Gérer les clients "

Après l'authentification, l'utilisateur effectue une demande de gestion des clients. Trois scénarios sont représentés chacun d'entre eux correspond à un choix, d'où l'utilisation du fragment de type " Opt " indiquant que ces scénarios arrivent dans n'importe quel ordre.

- Ajouter un client : Après l'affichage du formulaire, l'utilisateur saisi les informations d'un client et valide l'action.
- Modifier un client : L'utilisateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'utilisateur sélectionne le client concerné, saisit les modifications dans un formulaire et valide l'opération.
- Supprimer un client : L'utilisateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'utilisateur sélectionne le client concerné et valide la suppression.

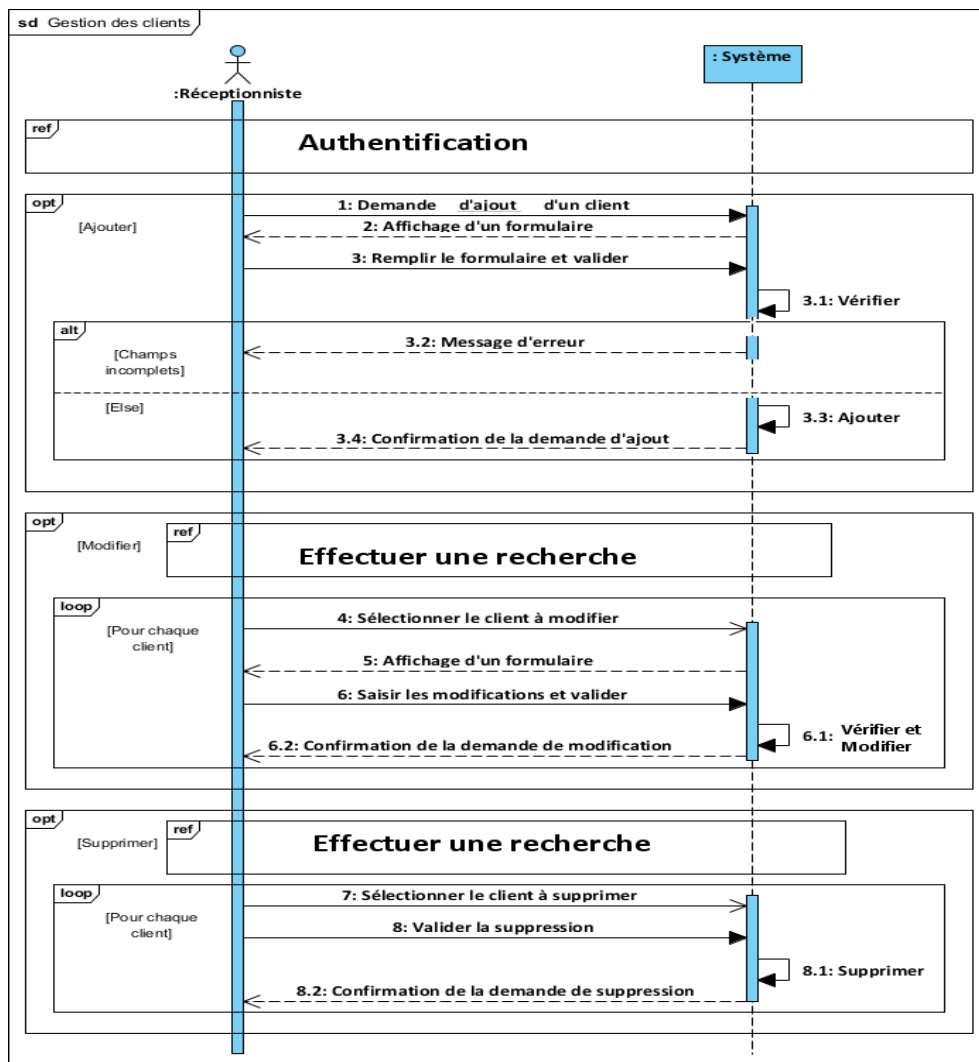


FIGURE 2.10 – Diagramme de séquence du cas d'utilisation " Gérer les clients ".

2.2.5.4 Digramme de séquence du cas d'utilisation " Gérer les réservations "

Après l'authentification, l'utilisateur effectue une demande de gestion des réservations. Trois scénarios sont représentés chacun d'entre eux correspond à un choix, d'où l'utilisation du fragment de type " Opt " indiquant que ces scénarios arrivent dans n'importe quel ordre.

- Ajouter une réservation : Après l'affichage du formulaire, l'utilisateur saisi les informations d'une réservation et valide l'action.
- Modifier une réservation : L'utilisateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'utilisateur sélectionne la réservation concernée, saisit les modifications dans un formulaire et valide l'opération.
- Supprimer une réservation : L'utilisateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'utilisateur sélectionne la réservation concernée et valide la suppression.

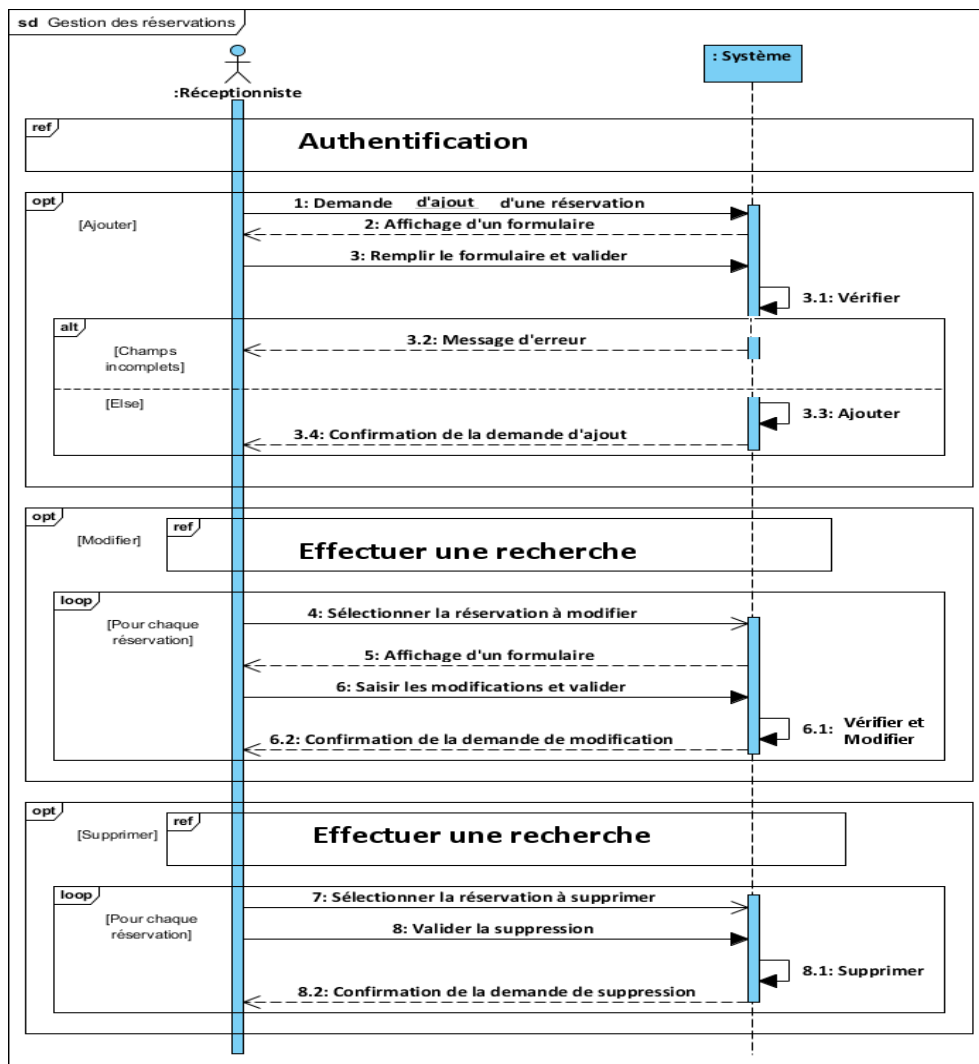


FIGURE 2.11 – Diagramme de séquence du cas d'utilisation " Gérer les réservations ".

2.2.5.5 Digramme de séquence du cas d'utilisation " Gérer les utilisateurs "

Après l'authentification, l'administrateur effectue une demande de gestion des utilisateurs. Trois scénarios sont représentés chacun d'entre eux correspond à un choix, d'où l'utilisation du fragment de type " Opt " indiquant que ces scénarios arrivent dans n'importe quel ordre.

- Ajouter un utilisateur : Après l'affichage du formulaire, l'administrateur saisi les informations d'un utilisateur et valide l'action.
- Modifier un utilisateur : L'administrateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'administrateur sélectionne l'utilisateur concerné, saisit les modifications dans un formulaire et valide l'opération.
- Supprimer un utilisateur : L'utilisateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'administrateur sélectionne l'utilisateur concerné et valide la suppression.

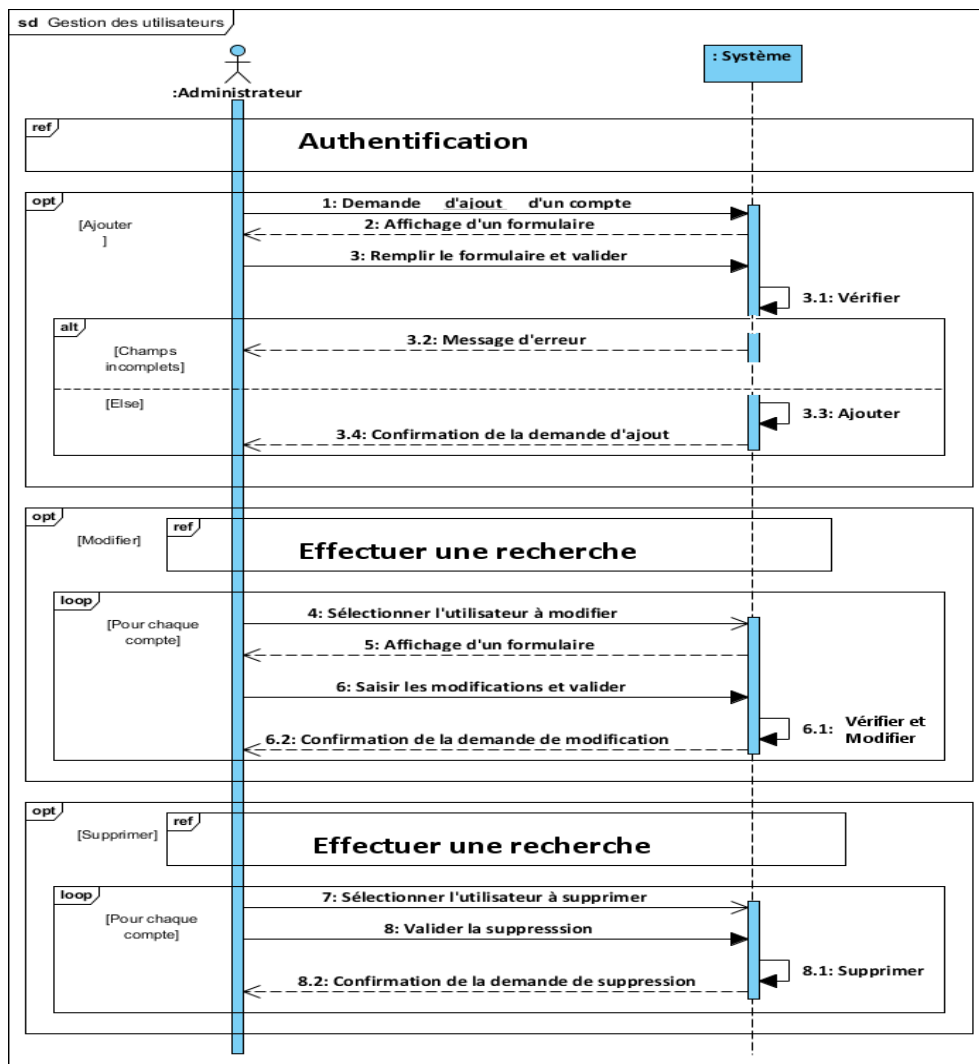


FIGURE 2.12 – Diagramme de séquence du cas d'utilisation " Gérer les utilisateurs ".

2.2.5.6 Digramme de séquence du cas d'utilisation " Gérer les chambres "

Après l'authentification, l'administrateur effectue une demande de gestion des chambres. Trois scénarios sont représentés chacun d'entre eux correspond à un choix, d'où l'utilisation du fragment de type " Opt " indiquant que ces scénarios arrivent dans n'importe quel ordre.

- Ajouter une catégorie : Après l'affichage du formulaire, l'administrateur saisi les informations d'une chambre et valide l'action.
- Modifier une chambre : L'administrateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'administrateur sélectionne la chambre concernée, saisit les modifications dans un formulaire et valide l'opération.
- Supprimer une chambre : L'utilisateur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'administrateur sélectionne la chambre concernée et valide la suppression.

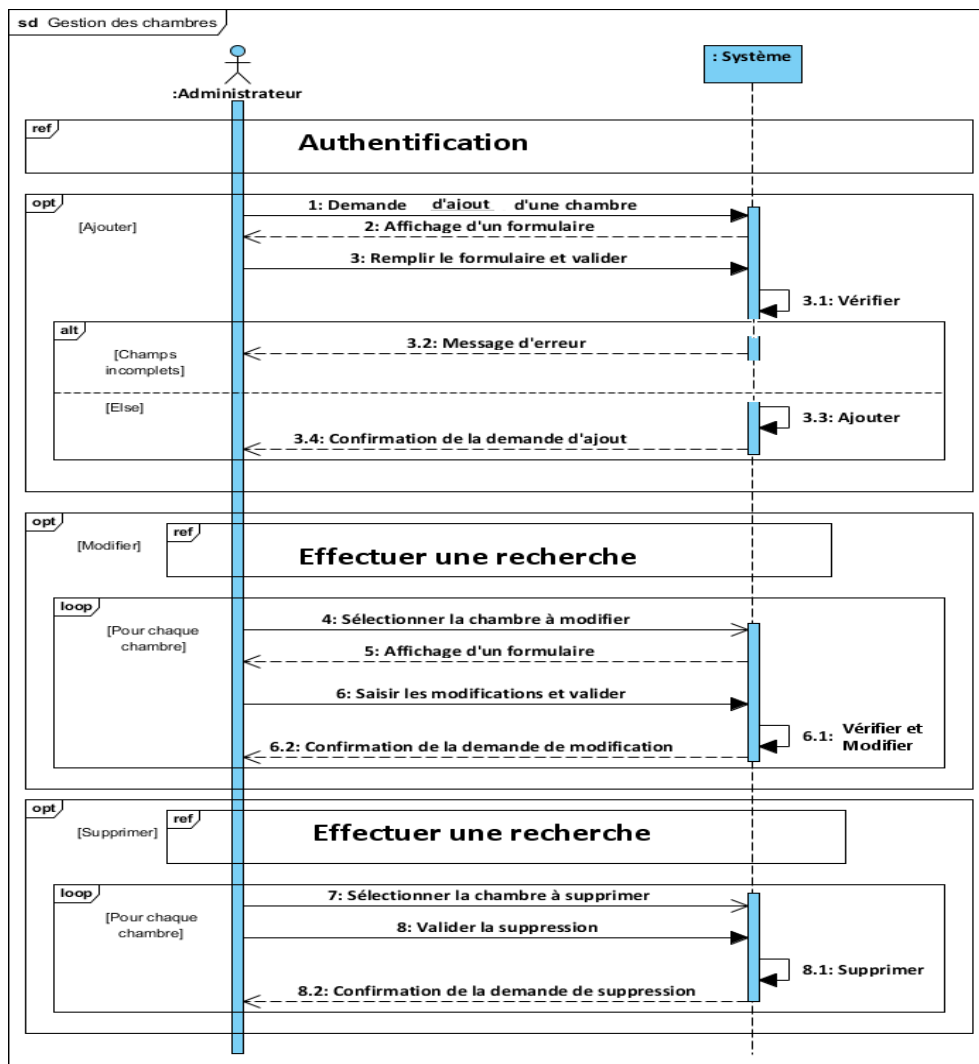


FIGURE 2.13 – Diagramme de séquence du cas d'utilisation " Gérer les chambres ".

2.3 Conclusion

Ce chapitre a été consacré au contexte de l'étude et de la modélisation fonctionnelle de notre application, où nous avons présenté les différents diagrammes de l'UML. En premier lieu, nous avons présenté le diagramme de contexte de notre système. Ensuite nous avons recensé les cas d'utilisation associés aux différents acteurs ainsi que les diagrammes de séquence qui correspondent à ces cas d'utilisation.

CHAPITRE 3

ANALYSE DE DOMAINE ET CONCEPTION

3.1 Introduction

La conception du système est une étape très importante, elle définit l'architecture globale du système aux niveaux logique et physique.

Dans ce chapitre, nous commençons par la présentation des diagrammes de séquence d'interaction associés à notre application. Nous passerons ensuite, à la présentation du diagramme de classes de l'application à réaliser. Enfin, nous terminons par une illustration du passage du modèle de classe au modèle relationnel décrivant l'implémentation de la base de données d'application à mettre en œuvre.

3.2 Diagrammes de séquence d'interactions

Un diagramme de séquence d'interaction se représente par un rectangle contenant, dans le coin supérieur gauche, un pentagone accompagné du mot-clef `sd` lorsqu'il s'agit d'un diagramme de séquence et `com` lorsqu'il s'agit d'un diagramme de communication. Le mot-clef est suivi du nom de l'interaction. Dans le pentagone, on peut aussi faire suivre le nom par la liste des lignes de vie impliquées, précédée par le mot-clef `lifelines` :. Enfin, des attributs peuvent être indiqués dans la partie supérieure du rectangle contenant le diagramme. La syntaxe de ces attributs est la même que celle des attributs d'une classe [10].

Nous présentons les diagrammes de séquence d'interactions des cas d'utilisations de l'application que nous mettons en œuvre.

3.2.1 Diagramme de séquence d'interaction du cas d'utilisation S'authentification

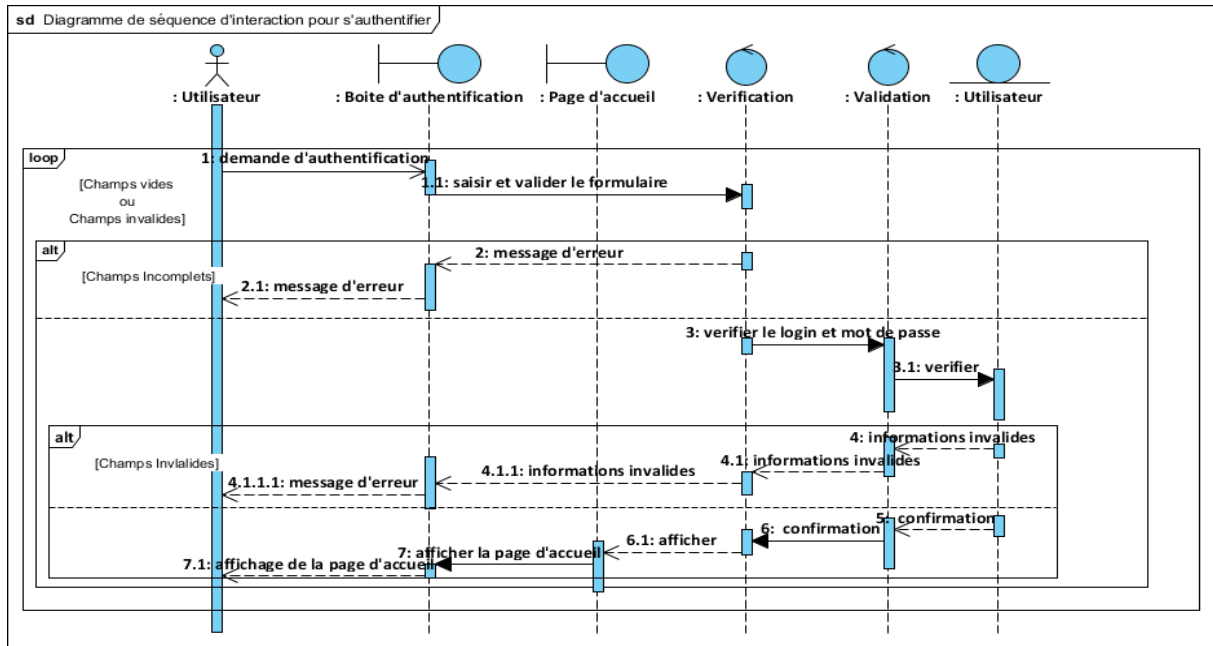


FIGURE 3.1 – Diagramme de séquence d'interaction du cas d'utilisation "S'authentification".

3.2.2 Diagramme de séquence d'interaction du cas d'utilisation "Effectuer une recherche"

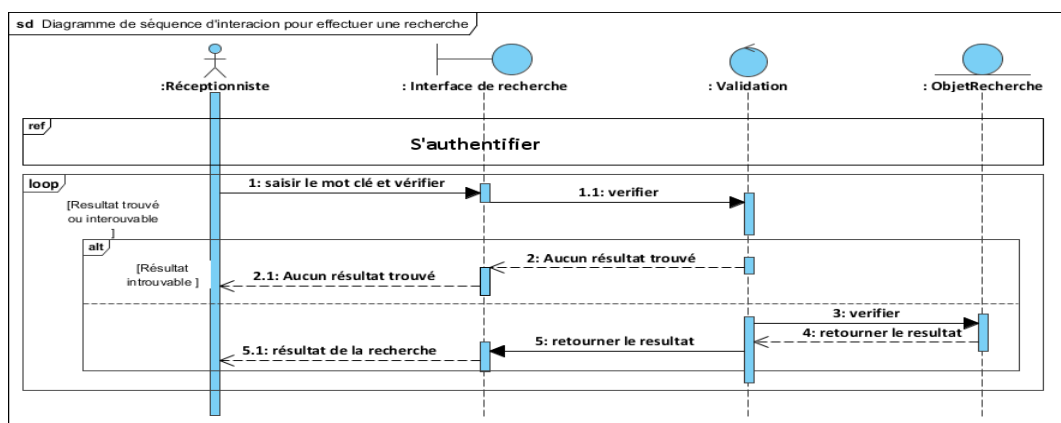


FIGURE 3.2 – Diagramme de séquence d'interaction du cas d'utilisation "effectuer une recherche".

3.2.3 Diagramme de séquence d'interaction du cas d'utilisation Gérer les clients

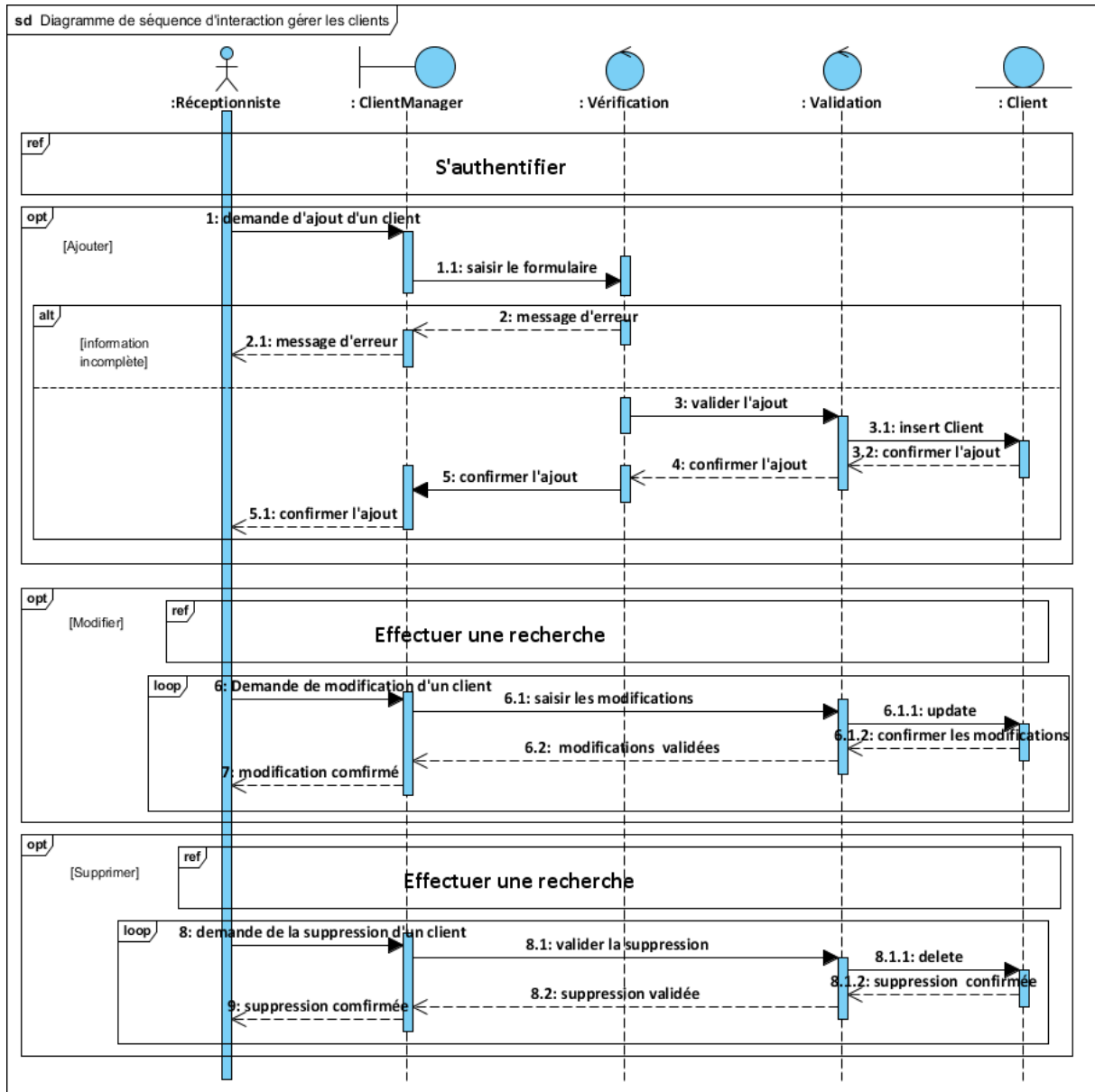


FIGURE 3.3 – Diagramme de séquence d'interaction du cas d'utilisation "Gérer les clients".

3.2.4 Diagramme de séquence d'interaction du cas d'utilisation Gérer les réservations

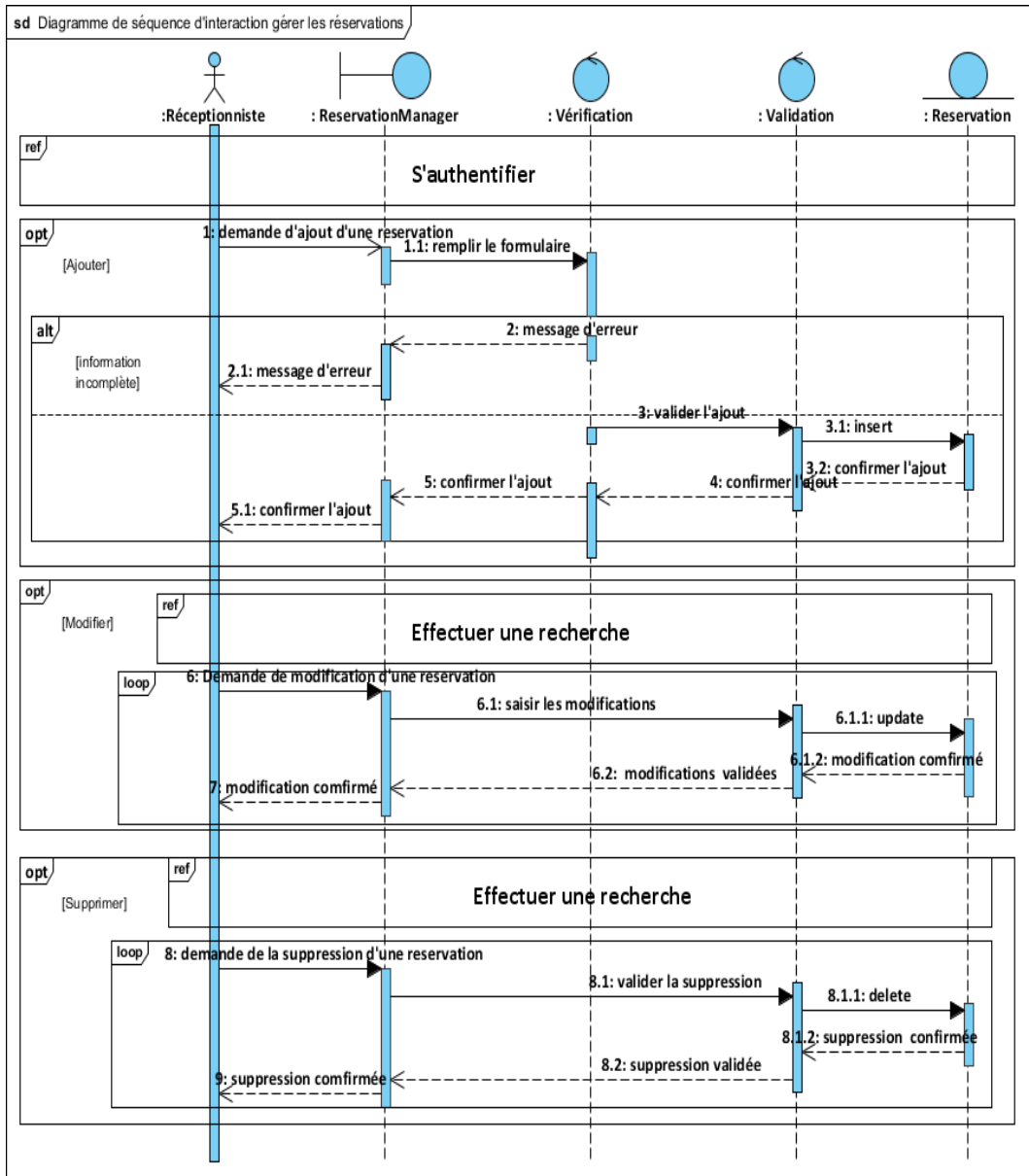


FIGURE 3.4 – Diagramme de séquence d'interaction du cas d'utilisation "Gérer les réservations".

3.2.5 Diagramme de séquence d'interaction du cas d'utilisation Gérer les utilisateurs

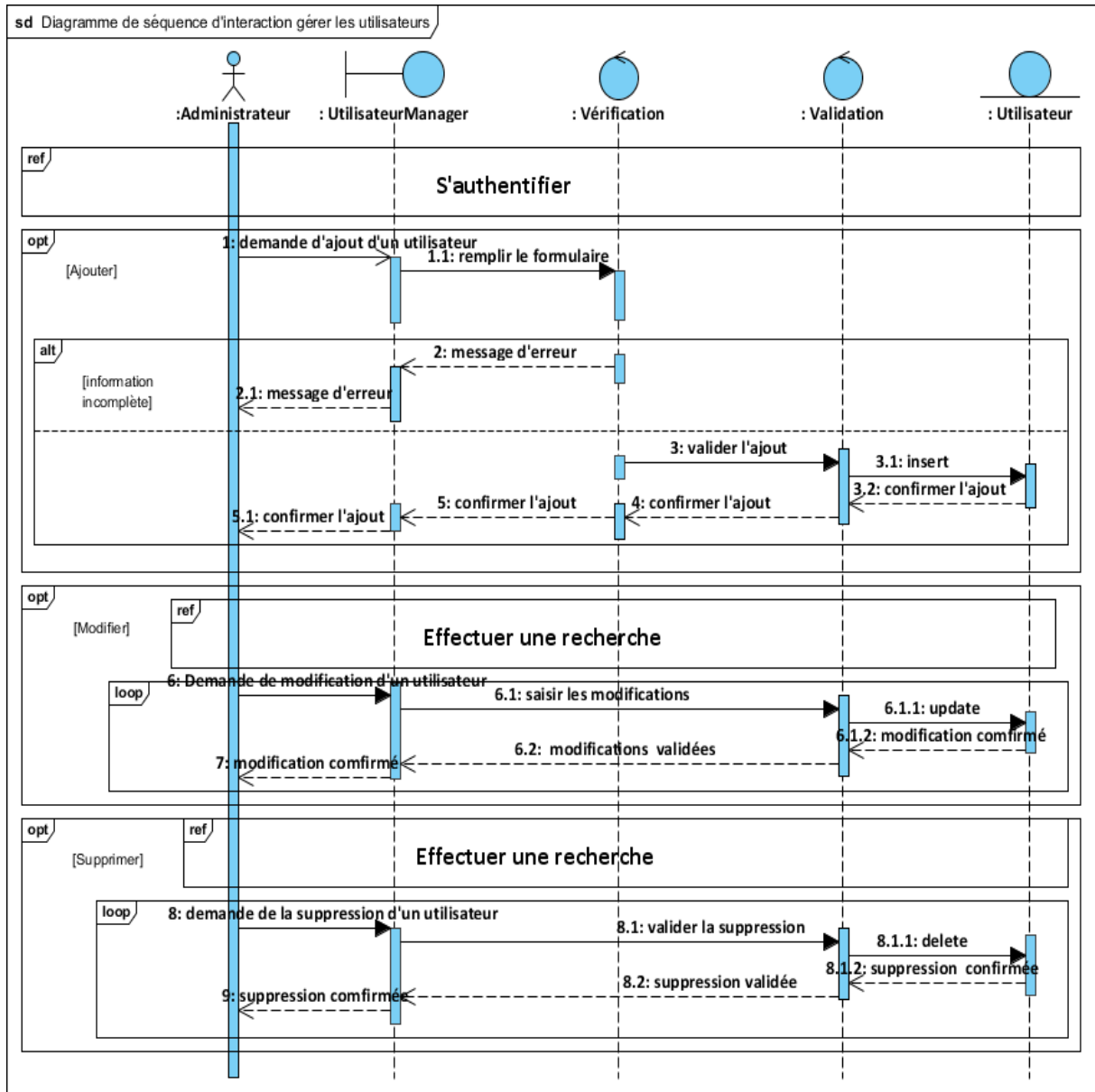


FIGURE 3.5 – Diagramme de séquence d'interaction du cas d'utilisation "Gérer les utilisateurs".

3.2.6 Diagramme de séquence d'interaction du cas d'utilisation Gérer les chambres

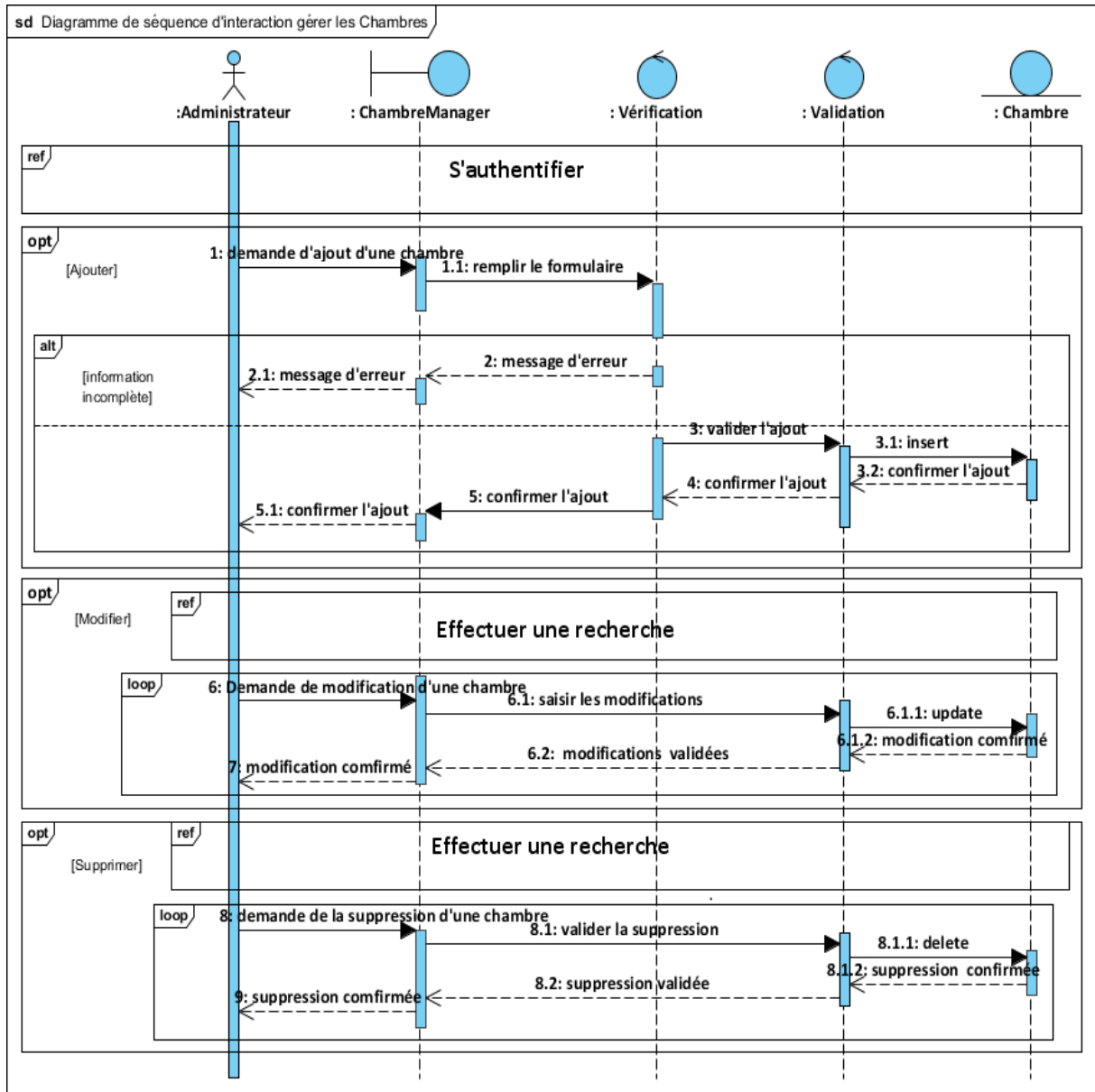


FIGURE 3.6 – Diagramme de séquence d'interaction du cas d'utilisation "Gérer les chambres".

3.2.7 Présentation du diagramme de classes

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe ; seul l'algorithme général et le pseudo-code correspondant peuvent être associés à la modélisation [9].

La description du diagramme de classe est fondée sur :

- Le concept d'objet.
- Le concept de classe comprenant les attributs et les opérations.
- Les différents types d'association entre classes.

3.2.8 Les concepts d'un diagramme de classe

Les éléments de base d'un diagramme de classes sont les suivants :

- **Objet** : Un Objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés [8].
- **Classe** : Une classe représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type [11].
Un objet est une instance d'une classe. La classe représente l'abstraction de ses objets. Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond à une adresse mémoire [8].
- **Opération** : Une opération représente un élément de comportement (un service) contenu dans une classe. Nous ajouterons plutôt les opérations en conception objet, car cela fait partie des choix d'attribution des responsabilités aux objets [11].
- **Attribut** : Un attribut est une donnée élémentaire servant à caractériser les classes et les relations [9].
- **Classe-association** : Il s'agit d'une association promue au rang de classe. Elle possède de tout à la fois les caractéristiques d'une association et celle d'une classe et peut donc porter des attributs qui se valorisent pour chaque lien [12].
- **Relation entre les classes** : En UML, il existe plusieurs types de relation entre les classes :

Agrégation : Une agrégation est un cas particulier d'association non symétrique exprimant une relation de contenance. Les agrégations n'ont pas besoin d'être nommées : implicitement elles signifient " contient " ou " est composé de " [12].

La composition : La composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre le "composant" et la ou les classes "composé". Autrement dit la suppression de la classe "composé" implique la suppression de la ou des classes "composant" [8].

Une composition est une association contraignante : la suppression d'un objet agrégat entraîne la suppression des objets agrégés.

Association : Une association se fait entre deux classes. Elle a un nom et deux extrémités qui permettent de la connecter à chacune des classes associées. Lorsqu'une association est définie entre deux classes, cela signifie que les instances de ces deux classes peuvent être reliées entre eux [13].

Héritage : En UML, une classe peut hériter d'autres classes. L'héritage entre classes UML doit être considéré comme une inclusion entre les ensembles d'instances de classes. Les objets instances des sous-classes sont des objets instances des superclasses [13].

3.2.9 Diagramme de classes

Le diagramme suivant, représente le diagramme de classe associé à l'application que nous réalisons :

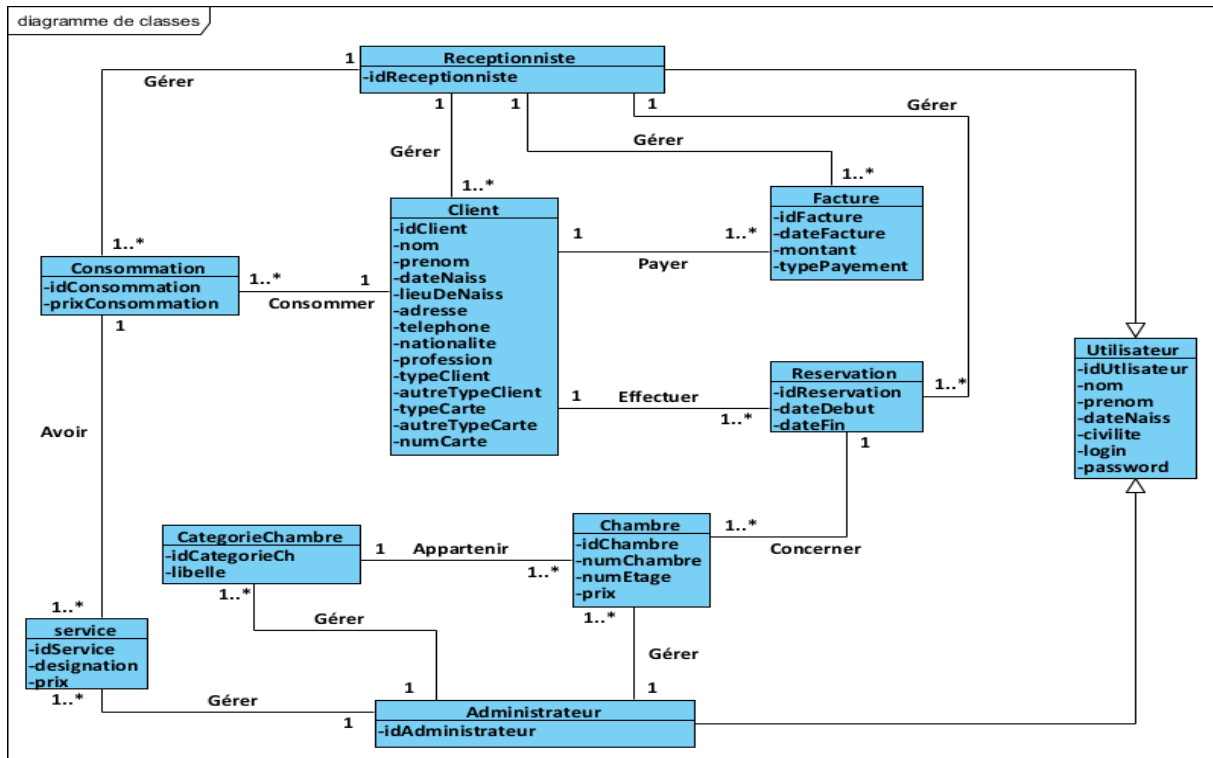


FIGURE 3.7 – Diagramme de classe de l’application à réaliser.

3.3 Présentation des propriétés et méthodes de chaque classe

Le tableau suivant décrit les attributs et les méthodes de chacune des classes et classe d’association du diagramme de notre système.

Classe/Classe-association	Attribut	Définition de l’attribut	Type /taille de l’attribut	Méthodes de classe
utilisateurs	Id	Identificateur de l’utilisateurs	Entier (04)	Authentification()
	Login	Login de l’utilisateurs	Chaine de caractères (25)	
	passWord	Mot de passe de l’utilisateurs	Chaine de caractères (25)	

client	Id	Identificateur de client	Entier (04)	Ajout() Modification() Suppression()
	typeClient	Type de client qui se soit envoyé par une entreprise ou par un simple client	ENUM/héberger, nonHéberger	
	autreType Client	S'il y a lieu à d'autre type de client	Chaine de caractères (25)	
	Nom	Nom du client	Chaine de caractères (25)	
	Prenom	Prénom du client	Chaine de caractères (25)	
	datNaiss	date de naissance du client	DATE (07)	
	Adresse	Adresse du client	Chaine de caractères (40)	
	Nationalite	Nationalité du client	Chaine de caractères (15)	
	profession	profession du client	Chaine de caractères (15)	
	telephone	Numéro de téléphone du client	Entier(10)	
	typeCarte	Pièce d'identité, passeport ou permis de conduire du client	ENUM/PC,CIN, passeport	
	autreType Carte	S'il y a lieu à d'autre type de Carte	Chaine de caractères (25)	
	numCarte	Numéro de la Pièce d'identité, passeport ou permis de conduire du client	Entier(20)	
Chambre	Id	Identificateur de la chambre	Entier (11)	Ajout()
	numChambre	Numéro de la chambre	Chaine de caractères (20)	

	numEtage	Numéro d'étage de la chambre	Chaine de caractères (20)	Modification() Suppression()
	prix	prix de la chambre	Réel	
catégorie Chambre	Id	Identificateur de la catégorie	Entier (04)	Ajout() Modification() Suppression()
	Libelle	Libellé de la catégorie d'une Chambre (suite, appartement)	Chaine de caractères (25)	
service	Id	Identificateur du service	Entier (04)	Ajout() Modification() Suppression()
	Designation	Nom du service	Chaine de caractères (25)	
facture	Id	Identificateur de la facture	Entier (11)	Ajout() Modification() Suppression()
	dateFacture	Date de la facture	DATE (07)	
	montant	Montant de la facture	Réel	
	typePaiement	Type de paiement de la facture	Réel	
reserva- tion	Id	Identificateur de la réservation	Entier (11)	Ajout () Modification() Suppression()
	dateDebut	Date où le client a effectué une réservation	DATE (10)	
	dateFin	Date de fin de la réservation	DATE (10)	
Consomma- tion	Id	Identificateur de la consommation	Entier (11)	Ajout() Modification() Suppression()
	prixService	prix du service	Réel	
	dateConsom	Date de consommation d'un service par un client	DATE (07)	

TABLE 3.1 – Représente les propriétés et méthodes de chaque classe.

3.4 Passage au modèle relationnel

Les règles utilisées pour le passage du diagramme de classes de notre application au modèle relationnel, sont tirées de [14] :

- **Règle 1 : Transformation des classes**

Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle de l'identifiant. Dans le cas où aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

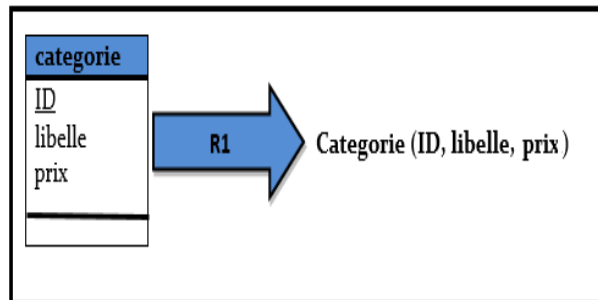


FIGURE 3.8 – Transformation des classes.

- **Règle 2 : Association un-à-plusieurs**

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

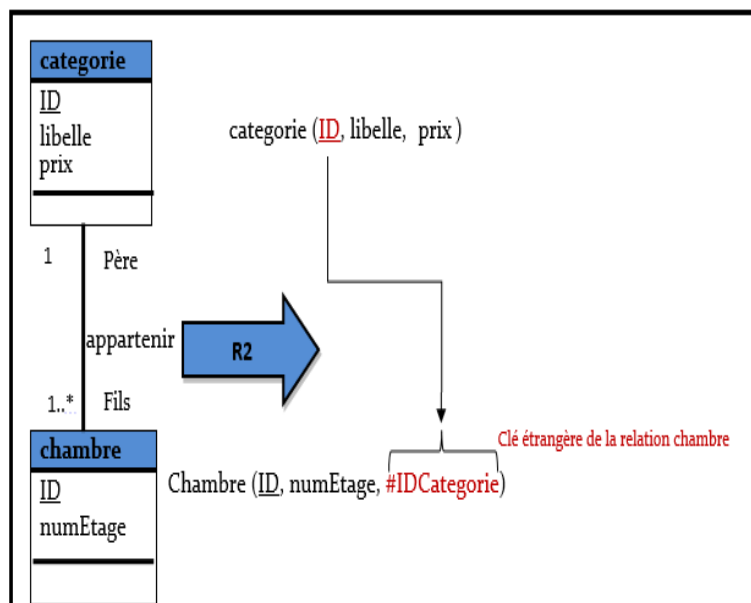


FIGURE 3.9 – Association un-à-plusieurs.

● **Règle 3 : Transformation de l'héritage**

Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes :

1. **Décomposition par distinction** : il faut transformer chaque sous-classe en une relation. La clé primaire de la sur-classe, migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.
2. **Décomposition descendante (push-down)** : s'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la sur-classe. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).
3. **Décomposition ascendante (push-up)** : il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de la sur-classe.

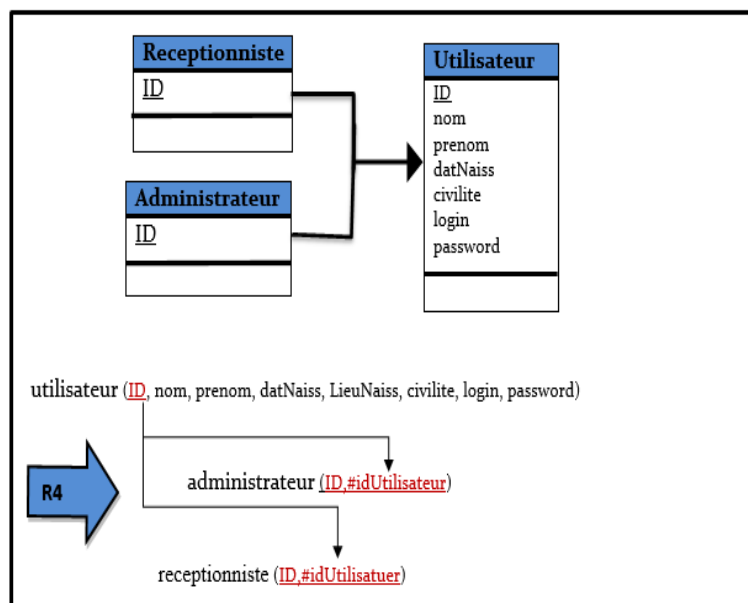


FIGURE 3.10 – Transformation de l'héritage.

Remarque : Les règles 4, 5 et 6 ne sont pas prises en considération dans le passage faute d'absence de cas les concernant, mais on les explique Comme même :

- **Règle 4 : Associations plusieurs-à-plusieurs**

L'association (classe-association) devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association (classe-association). Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

- **Règle 5 : Association un-à-un**

Il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la règle R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux entités (classes) en une seule.

- **Règle 6 : Composition**

La clé primaire des relations déduites des classes composantes doit contenir l'identifiant de la classe composite (quelles que soient les multiplicités).

L'application des règles de passage énumérées précédemment, nous permet d'avoir le modèle relationnel de la base de données de l'application à mettre en œuvre qu'est le suivant :

- **client** (id, nom, prenom, datNaiss, lieuNaiss, adresse, nationalite, telephone, profession, typeClient, AutreTypeClient, typeCarte, autreTypeCarte, numCarte, # idReceptionniste);
- **chambre** (id, numEtage, prix, #idCategorie, #idreservation, #idAdministrateur);
- **categoriechambre** (id, libelle, #idAdministrateur);
- **service** (id, designation, prix, #idconsommation, #idAdministrateur);
- **facture** (id, dateFact, montant, typePaiement, #idClient, #idReceptionniste);
- **consommation** (id, prixConsommation, dateConsommation, #idClient, #idReceptionniste);
- **reservation** (id, dateDebut, dateFin, #idClient, #idReceptionniste);
- **utilisateur** (idUtilisateur, nom, prenom, datNaiss, LieuNaiss, civilite, login, password);
- **administrateur** (idAdministrateur, #IdUtilisateur);
- **receptionniste** (idReceptionniste, #idUtilisateur);

Remarque : Pour la notation, nous avons choisi de souligner les clés primaires et de préfixer les clés étrangères par #.

3.5 Conclusion

Ce chapitre a été consacré à la conception de l'application à réaliser selon l'approche orientée objet. En premier, nous avons présenté les différents diagrammes de séquence d'interaction constitues notre application. Ensuite, nous avons recensé les concepts de base du diagramme de classes ainsi que les règles de modélisation, les règles de passage d'un diagramme de classes vers le modèle relationnelle qui nous permet d'avoir le schéma de la base de données de l'application à réaliser. Cela fait la base pour la phase réalisation tel qu'en vas garantir la fiabilité et l'efficacité de l'application à réaliser.

CHAPITRE 4

REALISATION

4.1 Introduction

L'étape de réalisation est la dernière de notre projet, elle se présente comme étant l'étape la plus cruciale vu qu'elle traite l'onglet pratique du projet.

Nous commençons d'abord par une brève illustration de l'environnement de travail ainsi que l'ensemble des logiciels qu'on a utilisé dans la réalisation de notre application de gestion d'un hôtel et l'implémentation de base de données, puis nous passons à un aperçu des interfaces les plus importantes de notre application.

4.2 Les outils de développements

4.2.1 Wamp Server

Wamp Serveur est une plateforme de développement web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Wamp-Server n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (apache et MySQL), un interpréteur de script(PHP), ainsi que PhpMyAdmin pour l'administration web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray-icon (icône près de l'horloge de Windows)[15].

4.2.2 Définition de MySQL

MySQL (My Structured Query Language) est un système de gestion de base de données Relationnelles (SGBDR) et basé sur un modèle client - serveur. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde. Son rôle consiste à stocker et à gérer une

grande quantité de données en les organisant sous forme de tables. Le système MySQL doit aussi permettre la manipulation de ces données à travers le langage standard du traitement des bases de données SQL. Les bases de données MySQL sont accessibles en utilisant les langages de programmation , Java, Perl, etc [16].

4.2.3 PHPMyAdmin

PHPMyAdmin est un logiciel libre, écrit en PHP destiné à gérer l'administration de MySQL sur le World Wide Web . PhpMyAdmin supporte une large gamme d'opérations avec MySQL. Les opérations les plus fréquemment utilisés sont pris en charge par l'interface utilisateur (bases de données de gestion, tables, champs, relations, les index, les utilisateurs, les permissions), alors que vous avez toujours la possibilité d'exécuter directement une instruction SQL. PhpMyAdmin permet de faire toutes sortes d'opérations comme : Créer et détruire des bases de données (à condition d'avoir les droits) ; Créer, détruire et modifier la description des tables ; Consulter le contenu des tables, modifier certaines lignes ou le détruire [17].

4.2.4 Java Development Kit (JDK)

kit de développement Java est un environnement de développement de logiciel utilisé pour développer des applications et des applets Java. Il inclut le Java Runtime Environment (JRE), un interprète / chargeur (java), un compilateur (javac), un programme d'archivage (pot), un générateur de documentation (javadoc) et d'autres outils nécessaires au développement Java [18].

4.2.5 Définition de l'éclipse

Eclipse est un environnement de développement intégré (IDE) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques. I.B.M. est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le cœur de son outil Websphere Studio Workbench (WSW), lui-même à la base de la famille des derniers outils de développement en Java d'I.B.M. Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse. Les principaux modules fournis en standard avec Eclipse concernent Java mais des modules sont en cours de développement pour d'autres langages notamment C++, Cobol, mais aussi pour d'autres aspects du développement (base de données, conception avec UML, ...). Ils sont tous développés en Java soit par le projet

Eclipse soit par des tiers commerciaux ou en open source. Les modules agissent sur des fichiers qui sont inclus dans l'espace de travail (Workspace). L'espace de travail regroupe les projets qui contiennent une arborescence de fichiers. Bien que développé en Java [19].

4.2.6 JDBC

Java DataBase Connectivity est très importante, appelée aussi " passerelle ", est composée d'un ensemble de classes permettant le dialogue entre une application Java et une source de données compatibles SQL (tables relationnelles en général). La passerelle JDBC a été développée de telle façon à permettre à un programme de se connecter à n'importe quelle base de données en utilisant la même syntaxe, c'est-à-dire que la passerelle JDBC est indépendante du SGBD. De plus, JDBC bénéficie des avantages de Java, dont la portabilité du code, ce qui lui vaut en plus d'être indépendant de la base de données d'être indépendant de la plate-forme sur laquelle elle s'exécute [20].

4.3 Les langages de programmation utilisés

4.3.1 Le langage SQL

SQL signifie " Structured Query Language " c'est-à-dire " Langage d'interrogation Structuré ". En fait SQL est un langage complet de gestion de bases de données relationnelles. L'accès aux BDD (bases de données) se fait de façon standard à l'aide de requêtes du langage SQL. Il existe un outil d'administration, PhpMyAdmin, qui nous offre une interface pour manipuler les tables. La connaissance de quelques requêtes permet de répondre à la majorité des besoins de programmation [21] :

	Requête	Description
SELECT	Sélectionner toute d'une table	SELECT * FROM nom table
	Extraire des lignes d'une table	SELECT * FROM nom table where champ='valeur'
	Extraire des colonnes d'une table	SELECT champ1, champ2, champ3 FROM nom table
DELETE	Supprimer des lignes d'une table	DELETE FROM nom table WHERE champ='valeur'

INSERT	Insérer des lignes dans une table d'une table	INSERT INTO nom table (champ1, champ2, champ3) VALUES ('xx', 'YY', 'zz')
UPDATE	Modifier des lignes dans une table d'une table	UPDATE nom table SET champ1="valeur1', champ2='valeur2' WHERE champid='valeur'

TABLE 4.1 – Représente les principales requêtes SQL.

4.3.2 Définition de Java

Java C'est un langage de programmation orienté objet, développé par Sun Microsystems. Dont le squelette principale et constitué du langage C++. En revanche, ces deux langages sont très différents dans leurs structures (organisation du code et gestion des variables). Ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp) et la réalisation de jeux fortement basé sur le graphisme. On effet, il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris) et la création de graphismes élaborés, animés, ainsi que la présentation de texte évolué (hypertexte). L'avantage principal de Java par rapport aux autres langages c'est sa PORTABILITE, le fait qu'un programme Java puisse théoriquement être exécuté sur n'importe qu'elle plate- forme (type de processeur et système d'exploitation) [22].

4.4 Diagramme de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs, etc.) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables, etc.) sur ces nœuds. C'est un véritable réseau constitué de nœuds et de connexions entre ces nœuds qui modélise cette architecture [9].

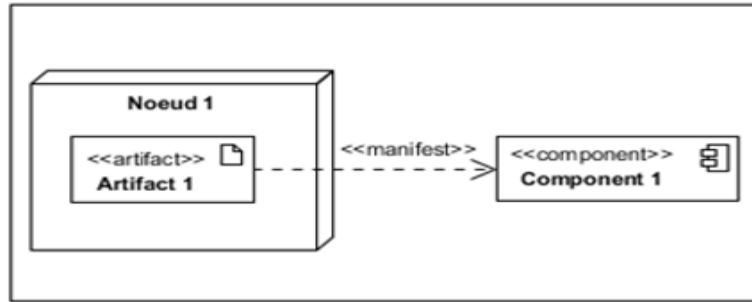


FIGURE 4.1 – Notation du diagramme de déploiement.

4.4.1 Diagramme de déploiement d’application réalisé

Chaque acteur a son propre poste client qui est un " PC " connecté au serveur web Apache, qui est lui-même un serveur d’applications sur le lequel est déployée en particulier l’application d’authentification.

Chacun de ces acteurs partagent la même interface homme-machine qui utilise le service d’authentification contenu par le serveur Apache. Toutes les tables sont stockées dans une base de données spécifique qui est hébergée par le serveur de base de données.

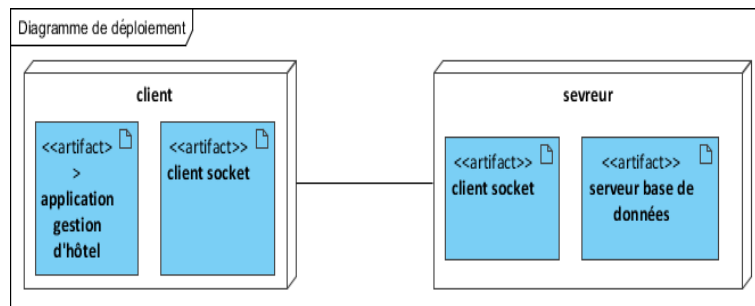


FIGURE 4.2 – Diagramme de déploiement d’application réalisé.

4.5 Arborescence de l’application

La figure ci-dessous illustre le fonctionnement de l’application :

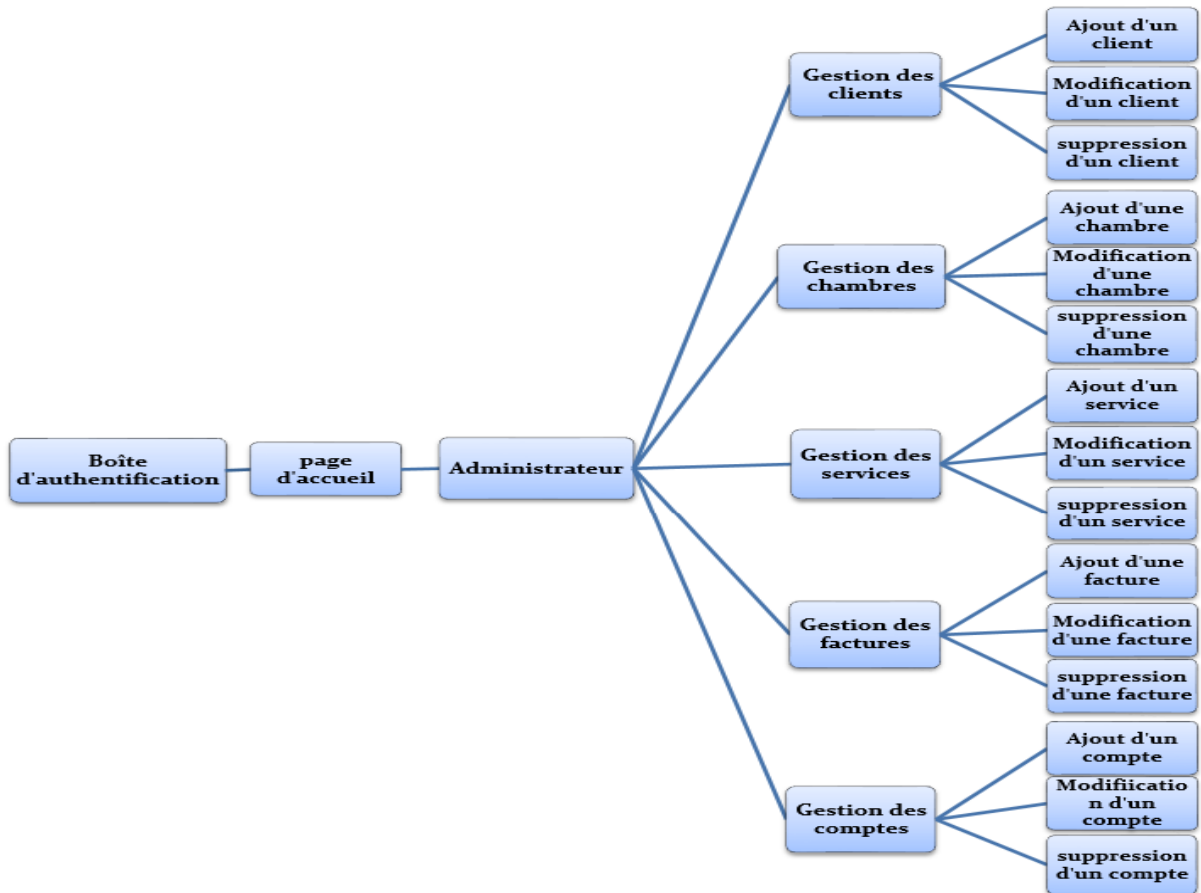


FIGURE 4.3 – Arborescence de l'application.

4.6 Présentation des interfaces de l'application

Notre application contient deux catégories d'utilisateurs : l'administrateur et le Réceptionniste. Dans ce qui suit, nous présentons les interfaces les plus importantes de notre application.

4.6.1 Page d'accueil de l'application

Cette page représente la page d'accueil dont l'utilisateur accède vers la toute première fois de son accès à l'application, elle présente l'organisme d'accueil, et elle fournit par ailleurs des informations liant l'application.

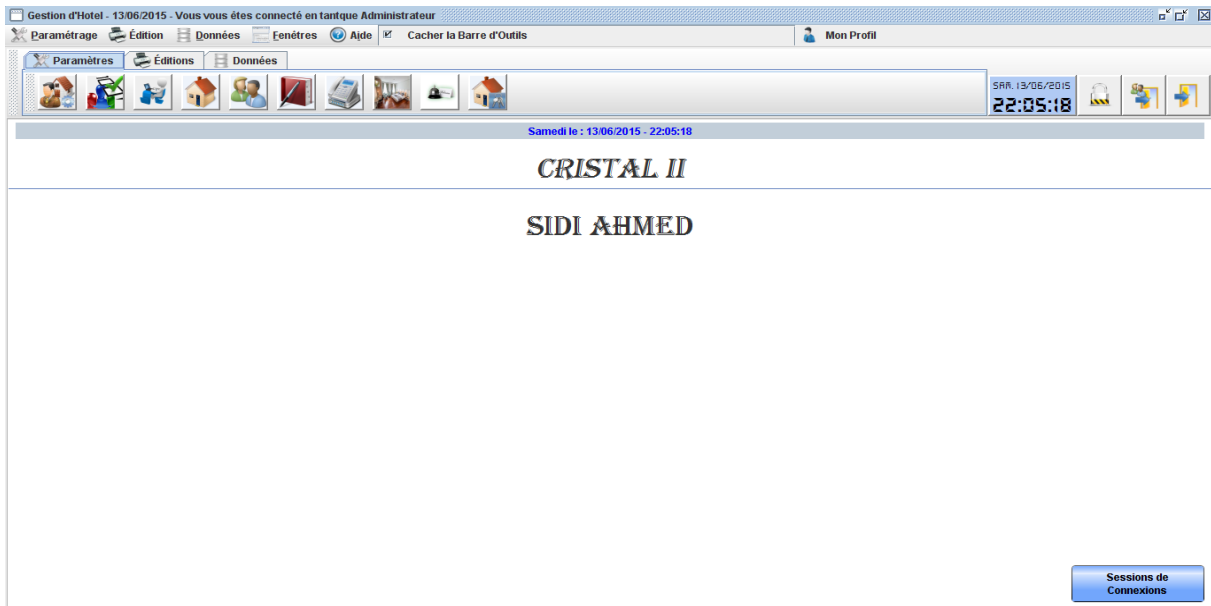


FIGURE 4.4 – Page d'accueil de l'application.

4.6.2 formulaire d'authentification

Un formulaire d'authentification illustré par la figure, l'utilisateur doit s'identifier par un login (nom d'utilisateur) et un mot de passe. Si ses informations existent dans la base de données et qu'elles correspondent l'une à l'autre, le système affiche la page d'accueil appropriés à l'utilisateur, si non la page sera réaffichée.

The image shows a dialog box titled "Gestion d'Hotel - 13/06/2015" with a close button (X) in the top right corner. The main heading is "Authentification". Below the heading, there are two input fields: "Nom d'utilisateur (Login) : root" and "Mot de passe : [masked]". At the bottom, there are three buttons: "OK" with a green checkmark, "Annuler" with a red X, and "Changer le mot de passe" with a red padlock icon.

FIGURE 4.5 – Formulaire d'authentification.

4.6.3 Interface gestion des utilisateurs

Cette interface permet à l'administrateur de gérer les comptes des utilisateurs c'est-à-dire. Il peut ajouter un nouveau compte en remplissant un formulaire contenant ses informations. Elle permet également la modification ou la suppression d'un compte après avoir le recherché en fonction d'un mot clé, ici deux cas se distinguent, si le mot clé existe alors le formulaire d'information relative à ce compte sera afficher, à fin de la modifier ou la supprimer, dans le cas contraire, un message d'erreur sera afficher.

Elle permet aussi l'impression de cette dernière en utilisant le bouton "imprimer ".

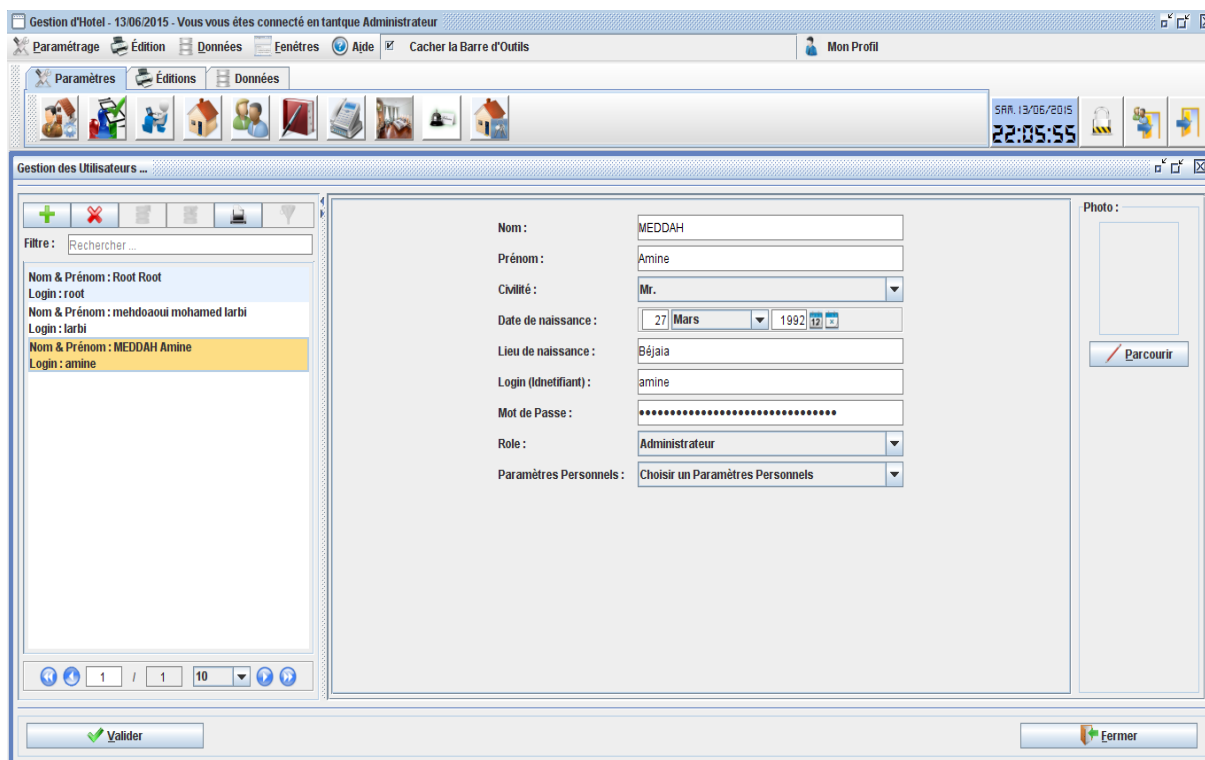


FIGURE 4.6 – Interface de gestion des utilisateurs.

4.6.4 Interface gestion des chambres

Cette interface permet à l'administrateur de gérer les chambres. Il peut ajouter une chambre en remplissant un formulaire contenant ses informations. Elle permet également la modification ou la suppression d'une chambre après avoir la recherchée en fonction de son numéro, ici deux cas se distinguent, si le numéro de la chambre existe alors le formulaire d'information relative à cette chambre sera afficher, à fin de la modifier ou la supprimer, dans le cas contraire, un message d'erreur sera afficher.

Elle permet aussi l'impression de cette dernière en utilisant le bouton "imprimer ".

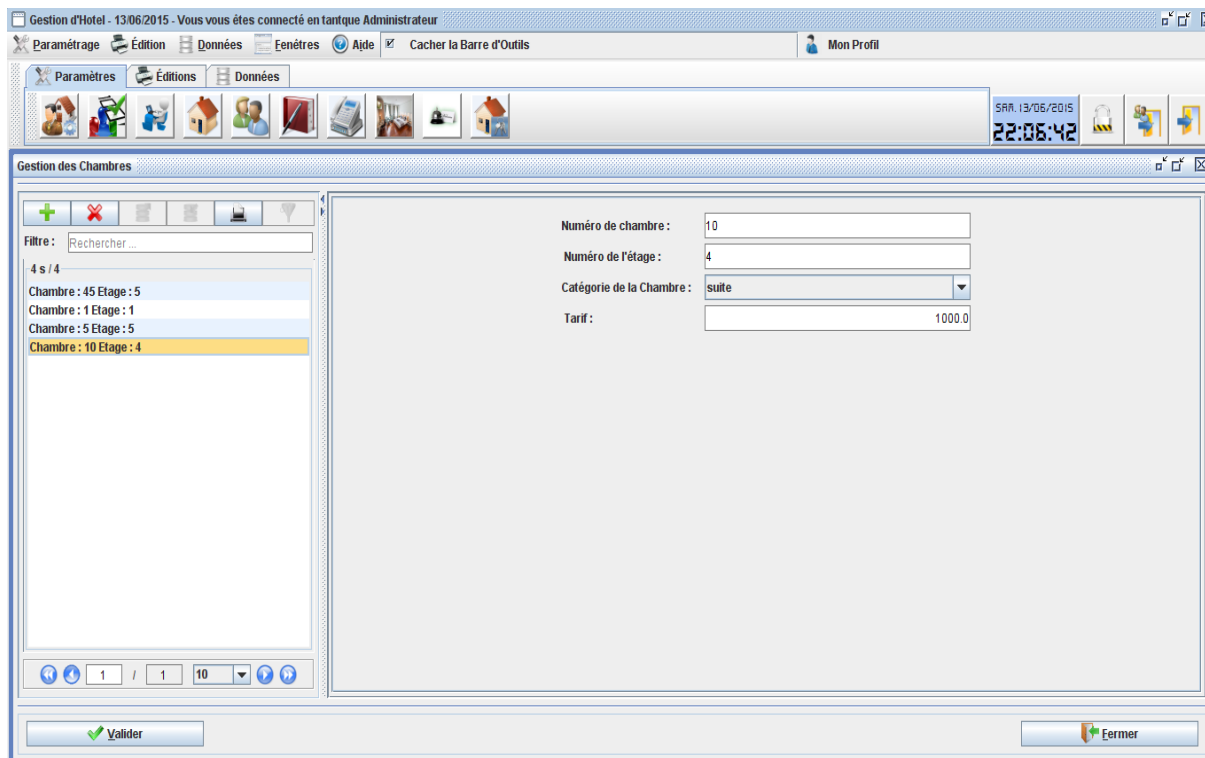


FIGURE 4.7 – Interface de gestion des chambres.

4.6.5 Inteface gestion des clients

Cette interface permet au réceptionniste de gérer les clients. Il peut ajouter un nouveau client en remplissant un formulaire contenant ses informations. Elle permet également la modification ou la suppression d'un client après avoir recherché en fonction de son identifiant, ici deux cas se distinguent, si le nom et prénom du client existe alors le formulaire d'information relative à ce client sera afficher, à fin de le modifier ou le supprimer, dans le cas contraire, un message d'erreur sera afficher.

Elle permet aussi l'impression de cette dernière en utilisant le bouton "imprimer ".

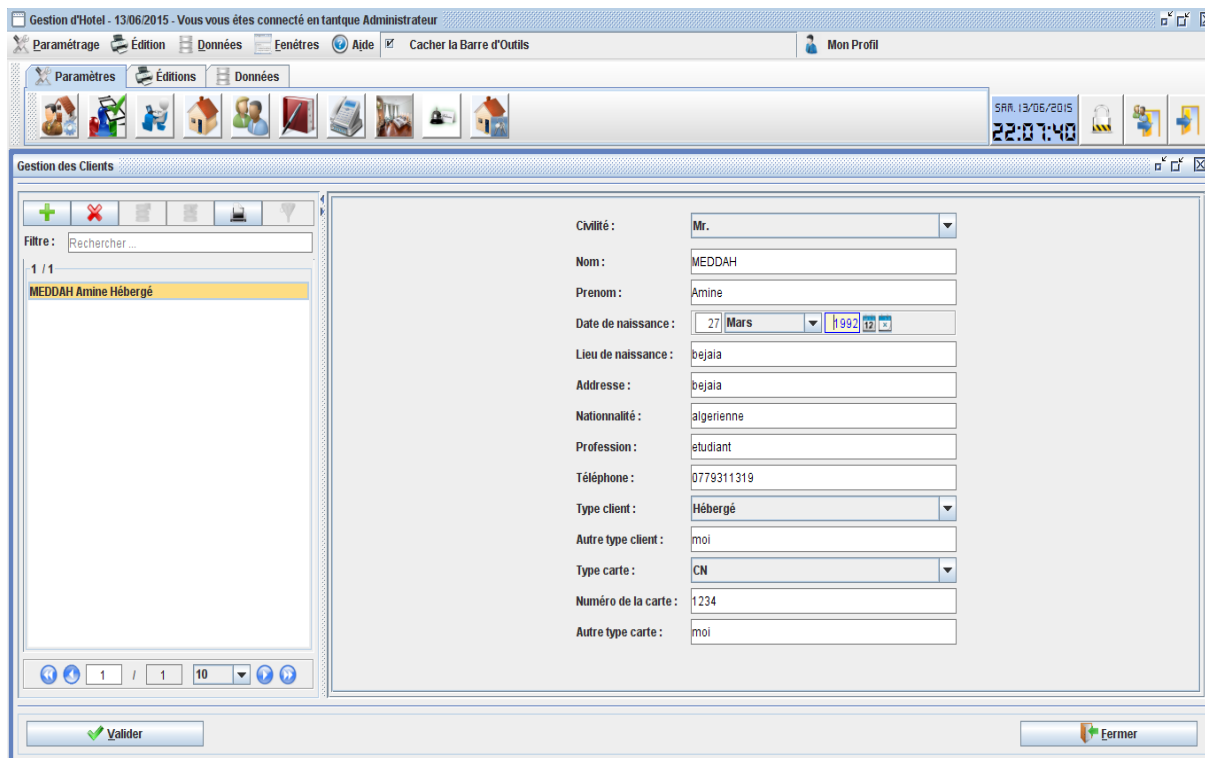


FIGURE 4.8 – Interface de gestion des clients.

4.6.6 Inteface gestion des factures

Cette interface permet à le réceptionniste de gérer les factures. Il peut ajouter un nouveau client en remplissant un formulaire contenant ses informations. Elle permet également la modification ou la suppression d’une facture après avoir la recherchée en fonction de son numéro, ici deux cas se distinguent, si le numéro de la facture existe alors le formulaire d’information relative à cette facture sera afficher, à fin de le modifier ou la supprimer, dans le cas contraire, un message d’erreur sera afficher.

Elle permet aussi l’impression de cette dernière en utilisant le bouton "imprimer ".

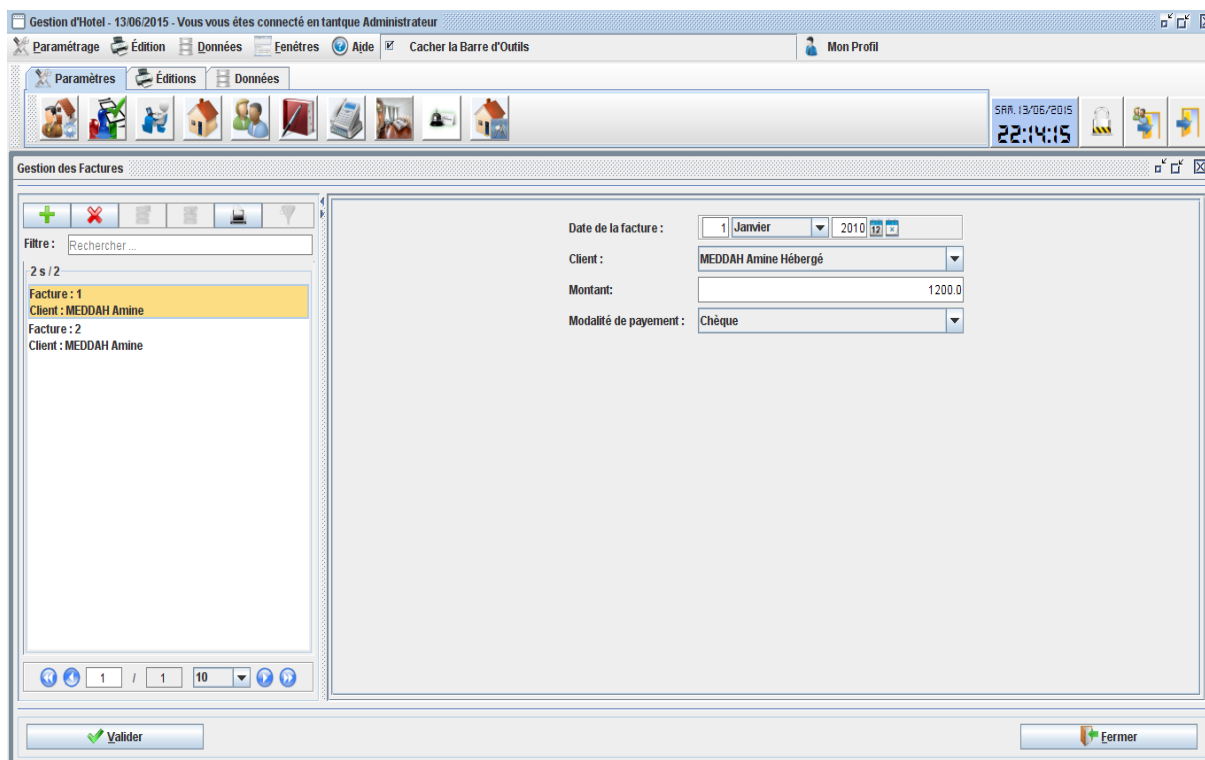


FIGURE 4.9 – Interface de gestion des factures.

4.6.7 Inteface gestion des services

Cette interface permet à l'administrateur de gérer les services. Il peut ajouter un service en remplissant un formulaire contenant ses informations. Elle permet également la modification ou la suppression d'un service après avoir le recherché en fonction de son nom, ici deux cas se distinguent, si le nom de service existe alors le formulaire d'information relative à ce service sera afficher, à fin de le modifier, dans le cas contraire, un message d'erreur sera afficher. Elle permet aussi l'impression de cette dernière en utilisant le bouton "imprimer ".

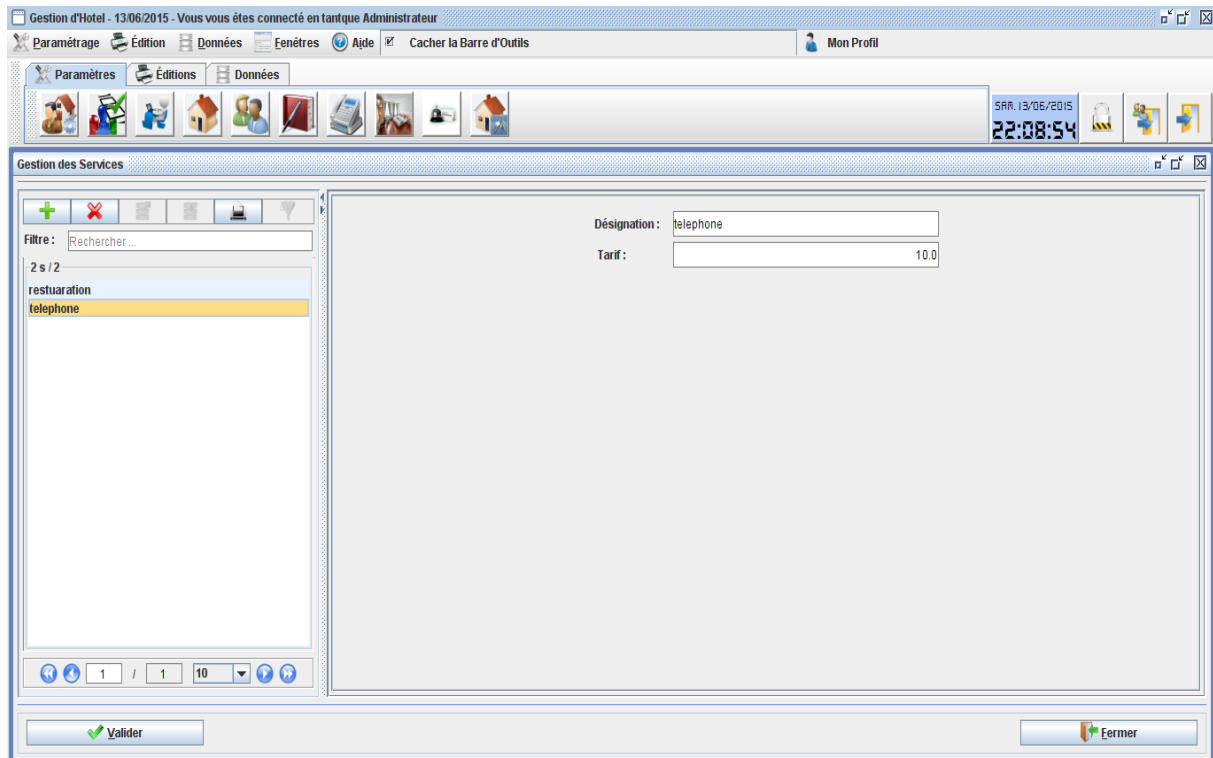


FIGURE 4.10 – Interface de gestion des service.

4.7 Conclusion

A travers ce chapitre, nous avons présenté la réalisation de notre application en justifiant nos choix technologique et en représentant les interfaces que nous avons jugé les plus importantes. La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre nous avons présenté les outils mis en œuvre pour réaliser notre application ainsi que le plan de l'application dans ses différentes cas en fin en conclure par la présentation des interfaces pour permettre d'expliquer notre application.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Le travail effectué dans ce mémoire s'inscrit dans le cadre de la conception et la réalisation d'une application réseau pour la gestion d'un hôtel.

Nous avons analysé la problématique et arrivé à concevoir une application qui est une solution efficace et bénéfique.

Pour mettre en œuvre ce projet, ce dernier a été amené dans un premier lieu à une étude axée sur la présentation de l'organisme d'accueil, qui est l'hôtel CRISTAL.

Ensuite, nous avons entamé la seconde partie dans laquelle nous avons établi une étude préliminaire pour identifier les différents acteurs interagissant avec le système réalisé. Après cela, nous avons réalisé la modélisation fonctionnelle. En effet, nous avons décrit les besoins des acteurs, suivi de la spécification des besoins fonctionnels à travers les diagrammes de cas d'utilisation et de l'analyse des besoins en utilisant les diagrammes de séquence.

Dans la troisième partie, une fois les fonctionnalités du système ont été analysées, nous sommes passés à la conception de notre application en se basant principalement sur le diagramme de séquence détaillé qui permet d'illustrer les interactions entre les objets du système. Par la suite, nous avons élaboré le diagramme de classes qui représente la description statique de notre système.

Enfin, dans la dernière partie, nous avons réalisé notre travail en spécifiant les outils de développement de l'application et les langages de programmation utilisés, suivi d'un aperçu sur les interfaces que comprend celui-ci.

Ce projet nous a été bénéfique, car nous avons eu la chance d'enrichir nos connaissances dans le domaine de la conception pas seulement sur le plan théorique ; mais aussi de découvrir et d'acquérir de nouvelles connaissances en matière de programmation et de développement de bases de données, sur le plan pratique.

Nous souhaitons que ce travail puisse servir comme un outil d'aide et de documentations pour les étudiants à l'avenir, et une base de travail pour les utilisateurs concernés.

BIBLIOGRAPHIE

- [1] http://phguilpain.free.fr/lycee/seconde/chap5/co/chap5organisation_grain.html.
- [2] R. Adel. *Conception et réalisation d'un site web de e-commerce pour le compte de LSAT_Nokia*. Thèse de Mastre en informatique, Université virtuelle de tunis (UVT), 2012-2013.
- [3] Pascal Roques. *Uml 2 modéliser une application web*. EYROLLES, 2008.
- [4] F. Vallée P. Roques. *processus de développement UML par action de l'analyse des besoins à la conception*. EYROLLES, 4ème édition, 2004.
- [5] D. Gaba J. Gabay. *UML2 Analyse et conception*. DUNOD, 1ère édition, 2008.
- [6] G. Roy. *Conception de bases de données avec UML*. Université de Strasbourg, 1ère édition, 2010.
- [7] R. fanader et H. Leroux. *UML principes de modélisation*. Presses de l'Université de Québec, paris, 2009.
- [8] J. Gabay et D. Gabay. *UML2 Analyse et conception*. DUNOD, 1ère édition, 2008.
- [9] J. Gabay et D. Gabay. *UML2 Analyse et conception*. DUNOD, 2008.
- [10] <http://www.google.fr/developpez.com/uml2-de-l'apprentissage-a-la-pratique.html>, février 2015.
- [11] P. Roques et F. Vallée. *UML2 en action de l'analyse des besoins à la conception*. EYROLLES, 4ème édition, 2007.
- [12] P. Roques. *UML2 par la pratique*. EYROLLES, paris, , 5ème édition, 2006.
- [13] C. Soutou. *UML2 pour les bases de données*. EYROLLES, 2002.
- [14] C. Soutou. *UML2 pour les bases de données*. EYROLLES, 1ère édition, 2006.
- [15] <http://fr.developer.org/wiki/wampserver.>, février 2015.

- [16] M. Contensin. *Bases de données et Internet avec PHP et MySQL*. DUNOD, 1ère édition, 2004.
- [17] <http://fr.developer.org/wiki/phpmyadmin>, février 2015.
- [18] <http://www.techopedia.com/definition/5594/javadevelopmentkitjdk> ., février 2015.
- [19] J. Michel. *Développons en Java avec Eclipse*. DOUDOUX, 2006.
- [20] J. HANIN. *Réalisation et mise en place d'une application de GESTION DE FLOTTE*
Rapport de Travail de Fin d'Etudes.
- [21] R. Grin. *Le langage SQL, version 2.3*. Univesité de Nice Sophia- Antipolis, 2000.
- [22] http://www.futurasciences.com/fr/definition/t/internet2/d/java_485/., février 2015.

ANNEXE A

DIAGRAMME DE SÉQUENCE

A.0.0.1 Diagramme de séquence du cas d'utilisation " Gérer les consommations "

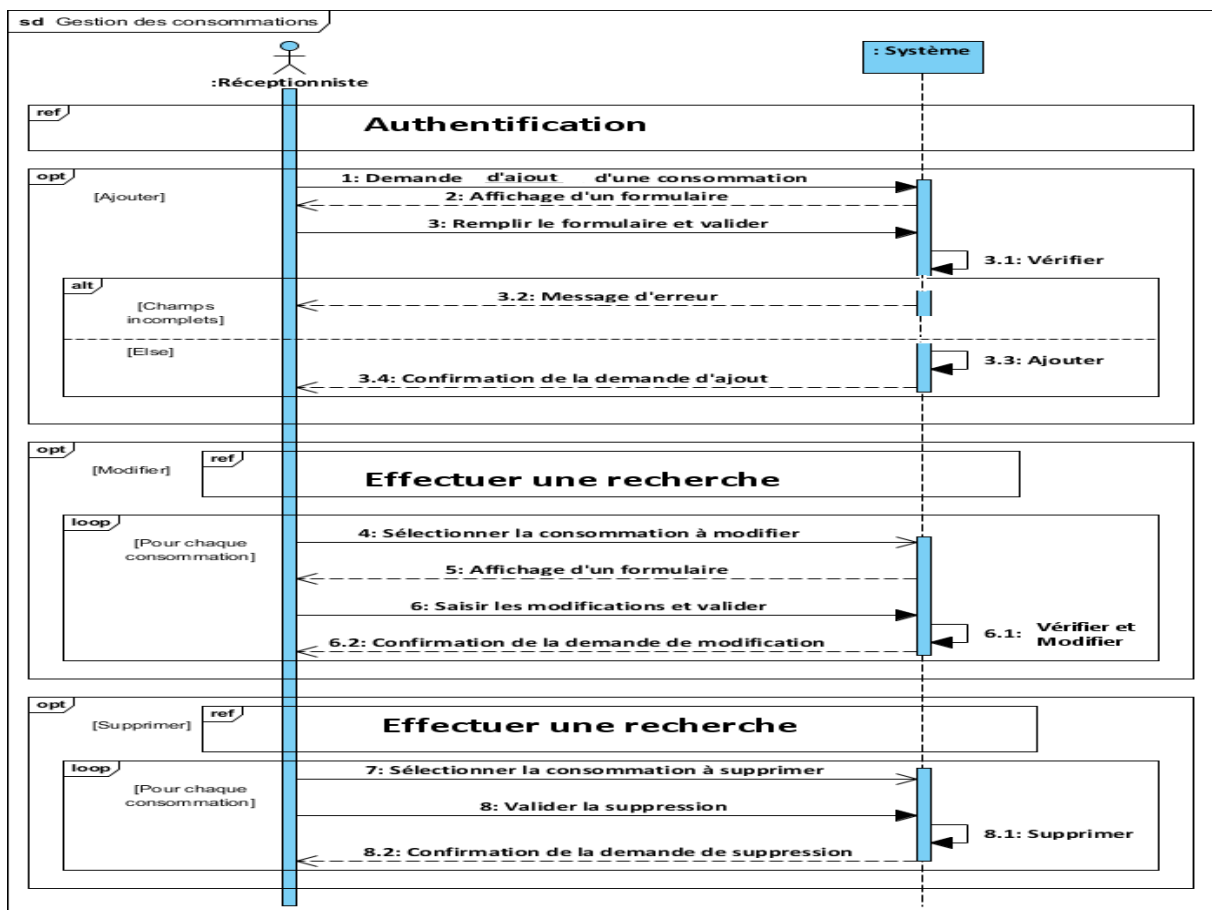


FIGURE A.1 – Diagramme de séquence du cas d'utilisation " Gérer les consommations ".

A.0.0.2 Diagramme de séquence du cas d'utilisation " Gérer les factures "

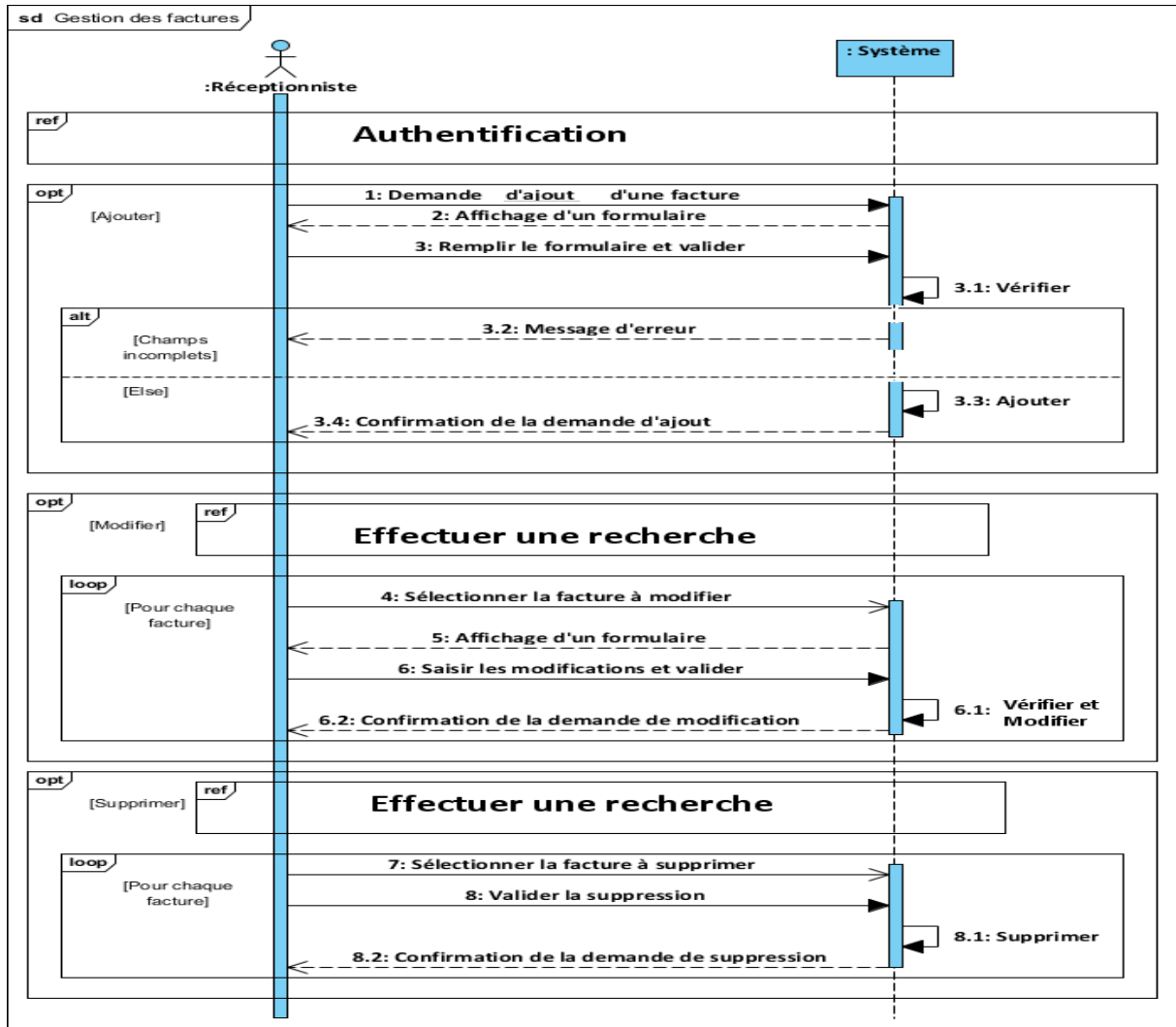


FIGURE A.2 – Diagramme de séquence du cas d'utilisation " Gérer les factures ".

A.0.0.3 Digramme de séquence du cas d'utilisation " Gestion les catégories "

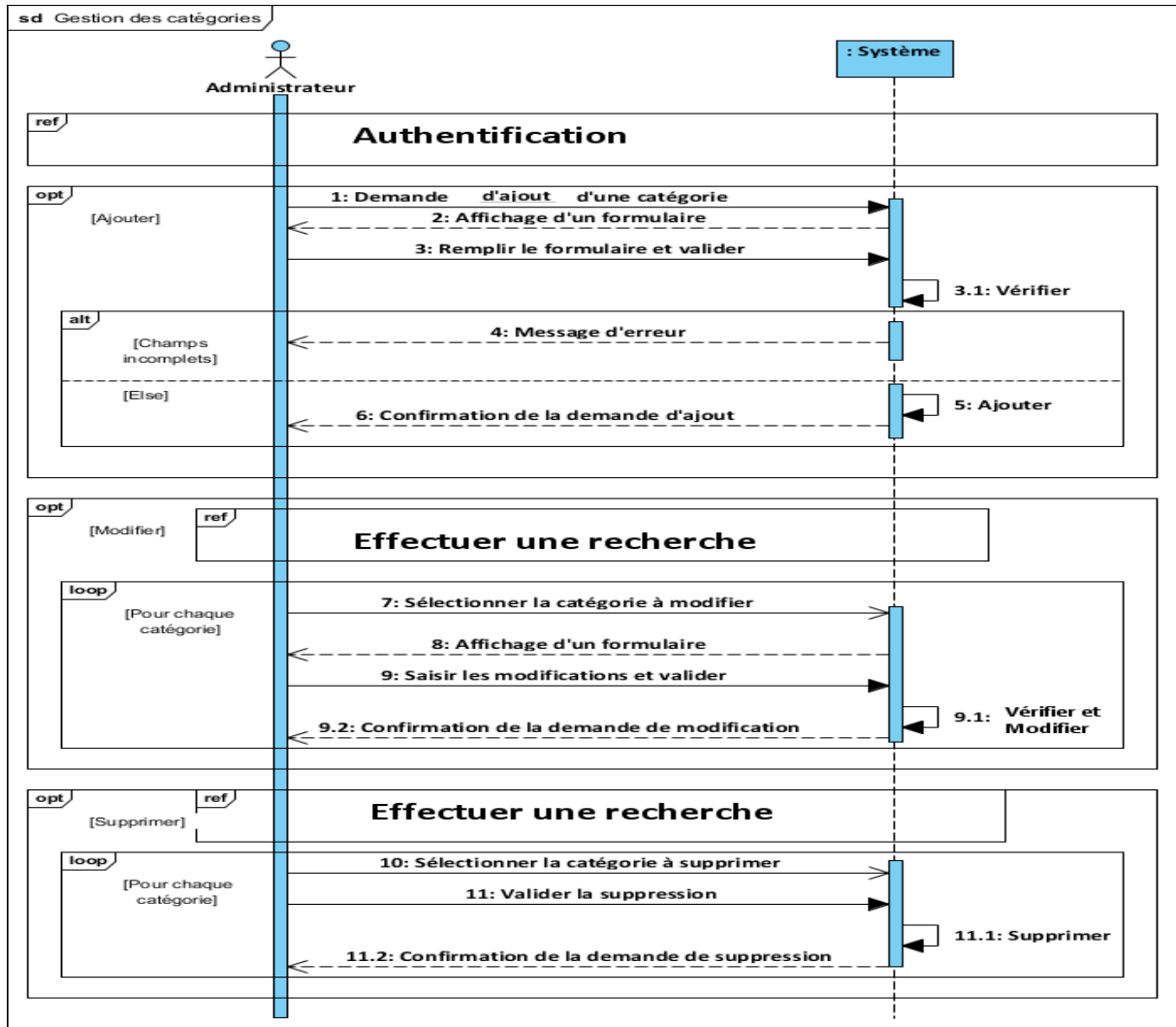


FIGURE A.3 – Diagramme de séquence du cas d'utilisation " Gérer les catégories ".

A.0.0.4 Diagramme de séquence du cas d'utilisation " Gérer les services "

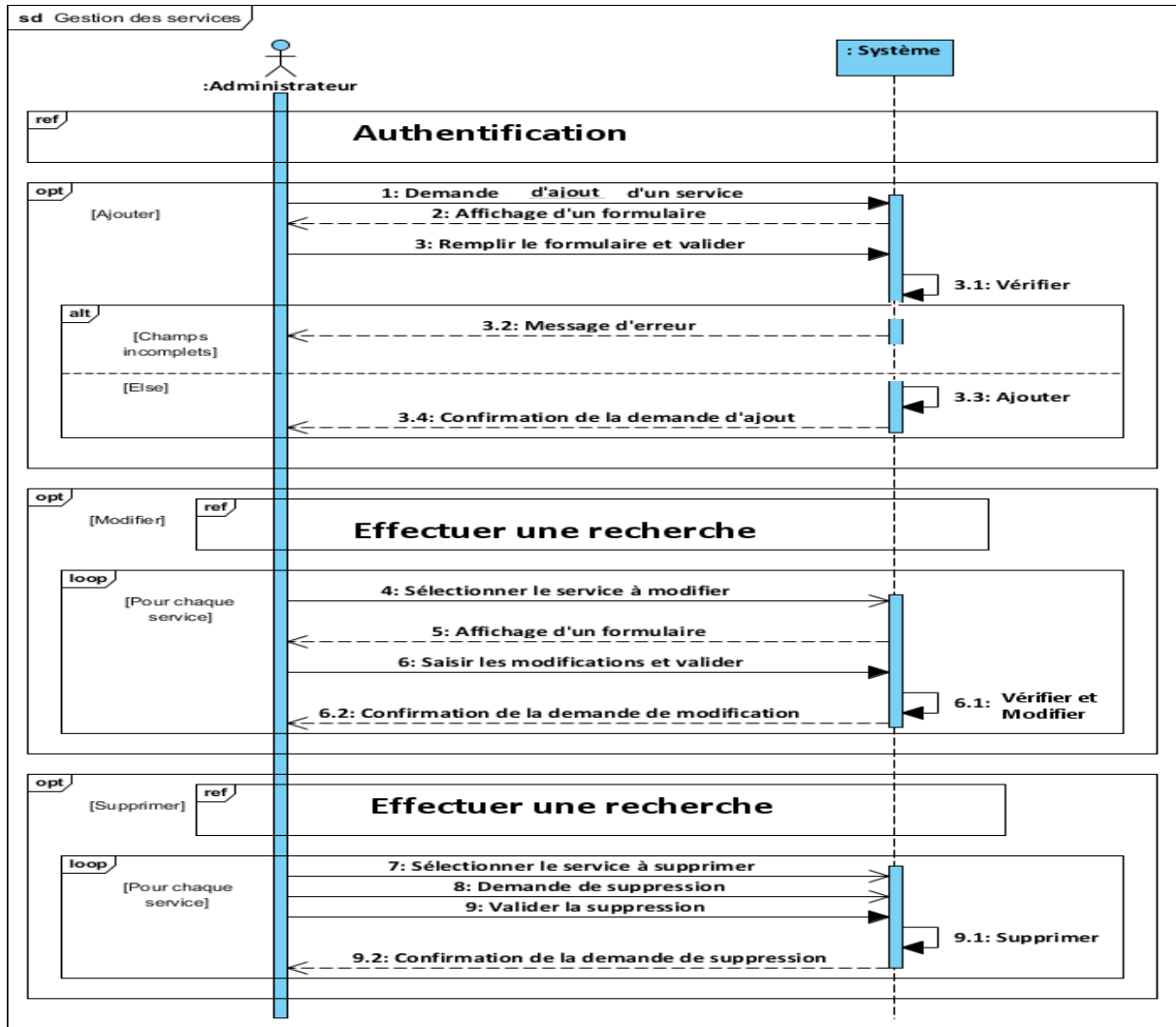


FIGURE A.4 – Diagramme de séquence du cas d'utilisation " Gérer les services ".

ANNEXE B

DIAGRAMME DE SÉQUENCE D'INTERACTION

B.0.1 Diagramme de séquence d'interaction du cas d'utilisation " Gérer les consommations "

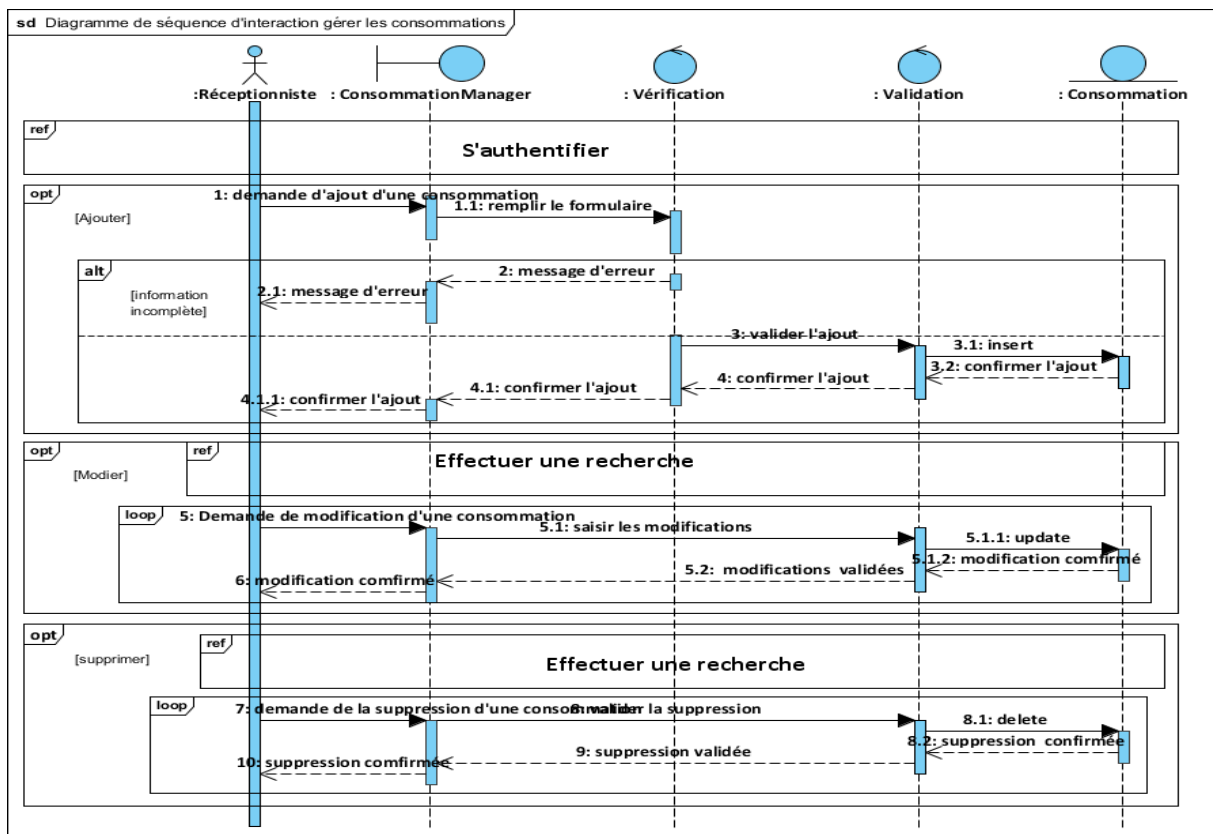


FIGURE B.1 – Diagramme de séquence d'interaction du cas d'utilisation " Gérer les consommations ".

B.0.2 Diagramme de séquence d'interaction du cas d'utilisation " Gérer les factures "

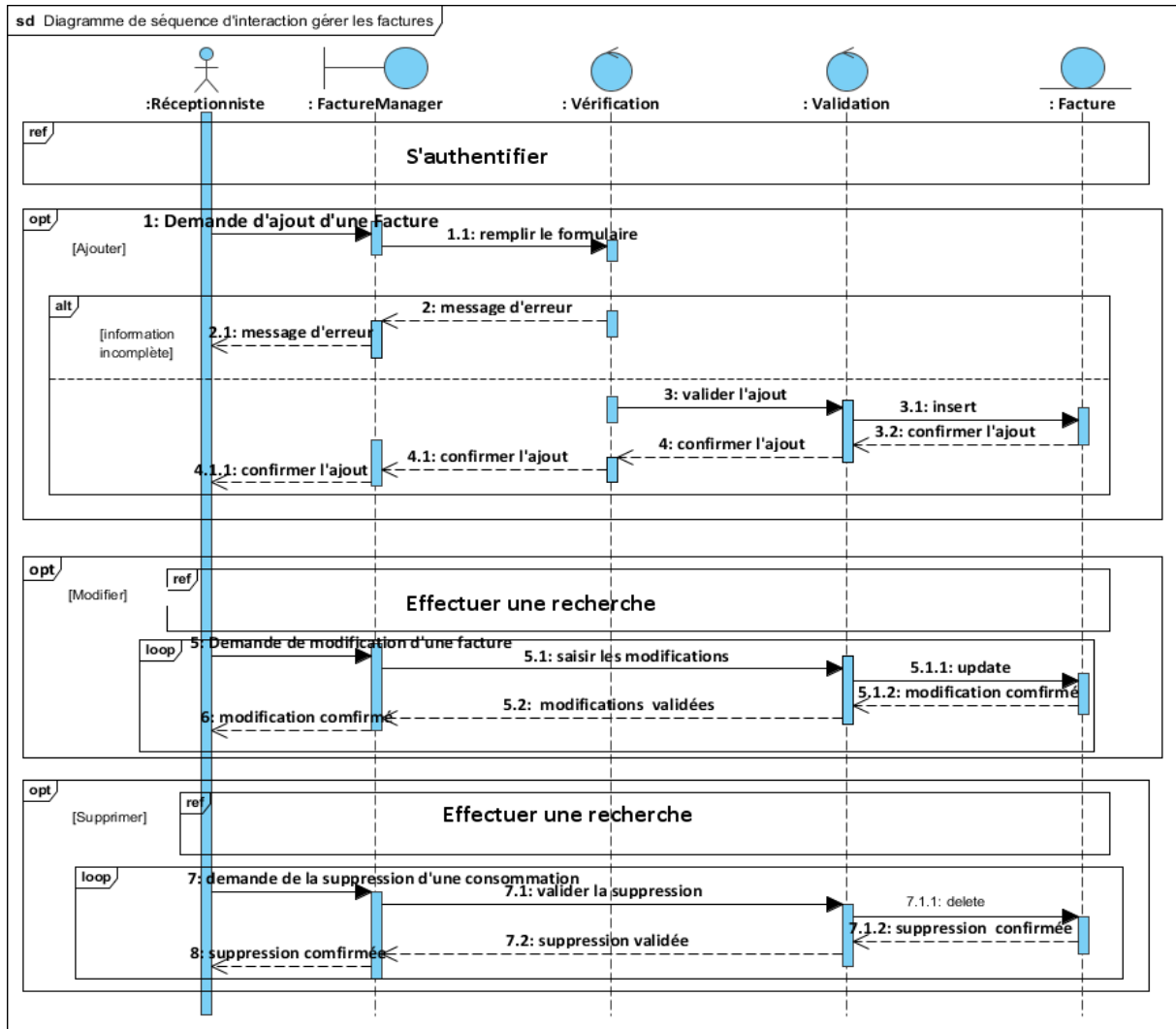


FIGURE B.2 – Diagramme de séquence d'interaction du cas d'utilisation " Gérer les factures ".

B.0.3 Diagramme de séquence d'interaction du cas d'utilisation " Gérer les catégories "

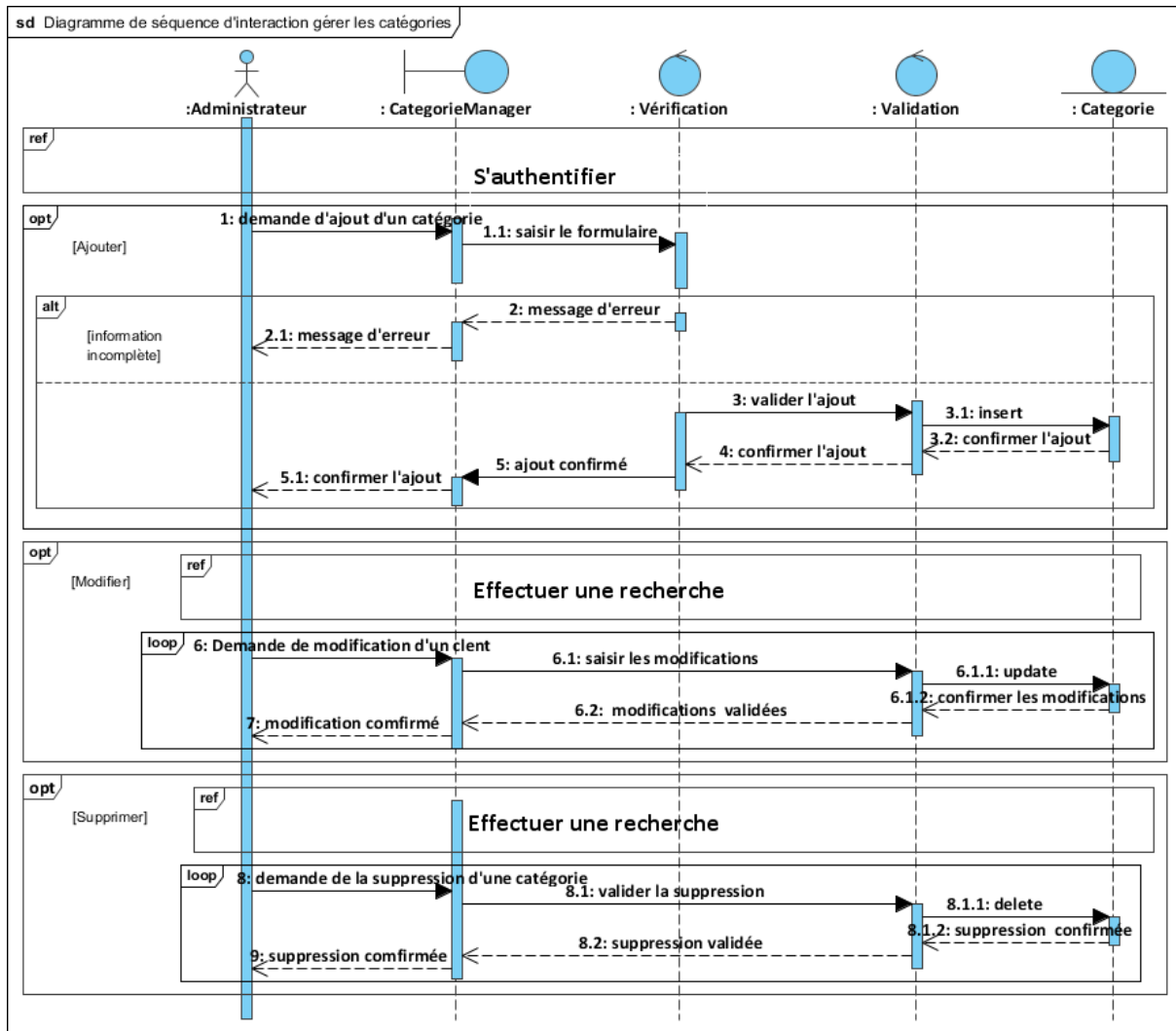


FIGURE B.3 – Diagramme de séquence d'interaction du cas d'utilisation " Gérer les catégories "

" .

B.0.4 Diagramme de séquence d'interaction du cas d'utilisation " Gérer les services "

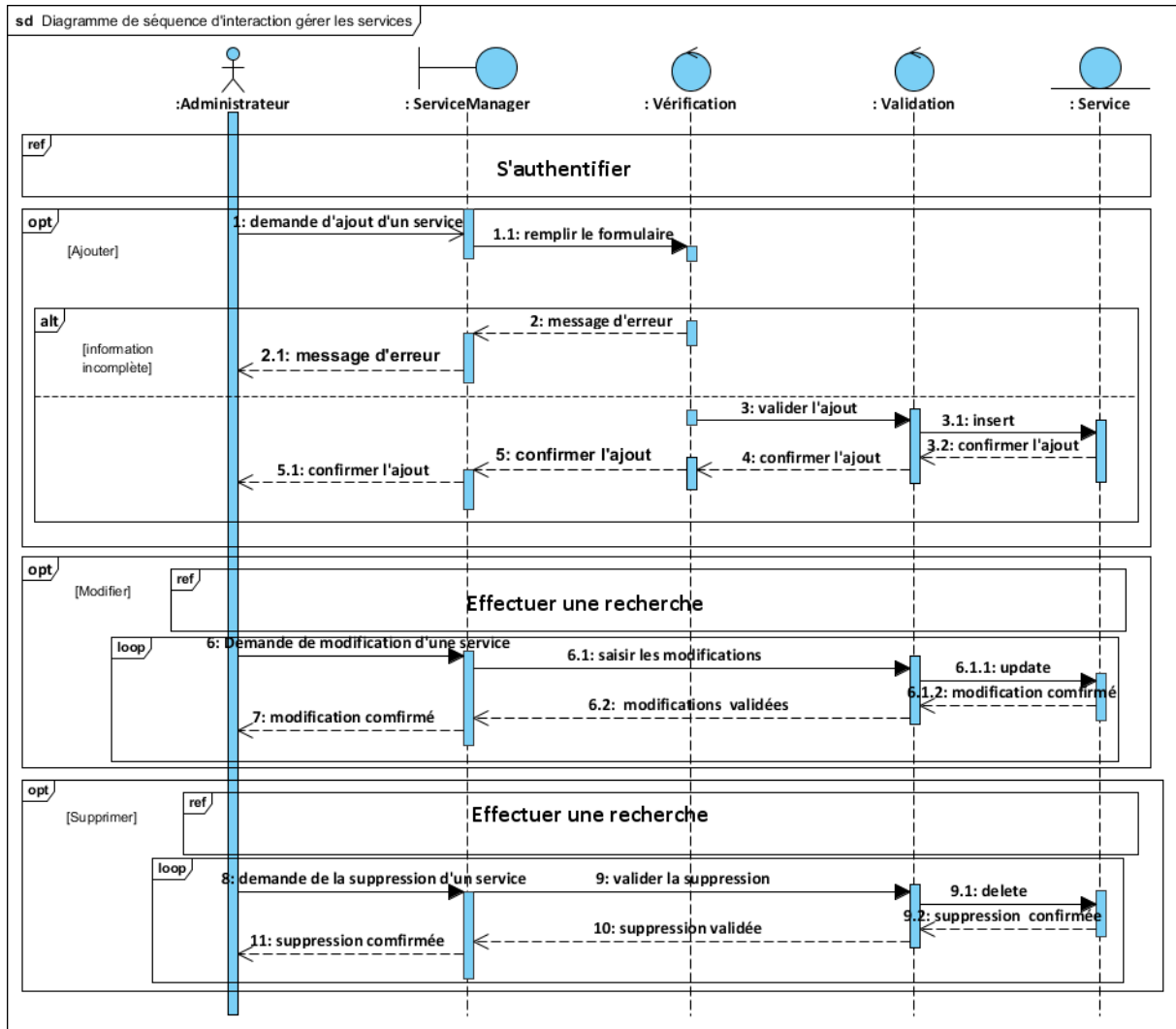


FIGURE B.4 – Diagramme de séquence d'interaction du cas d'utilisation " Gérer les services ".

ANNEXE C

INTERFACE DE L'APPLICATION

C.0.5 Interface gestion des catégories

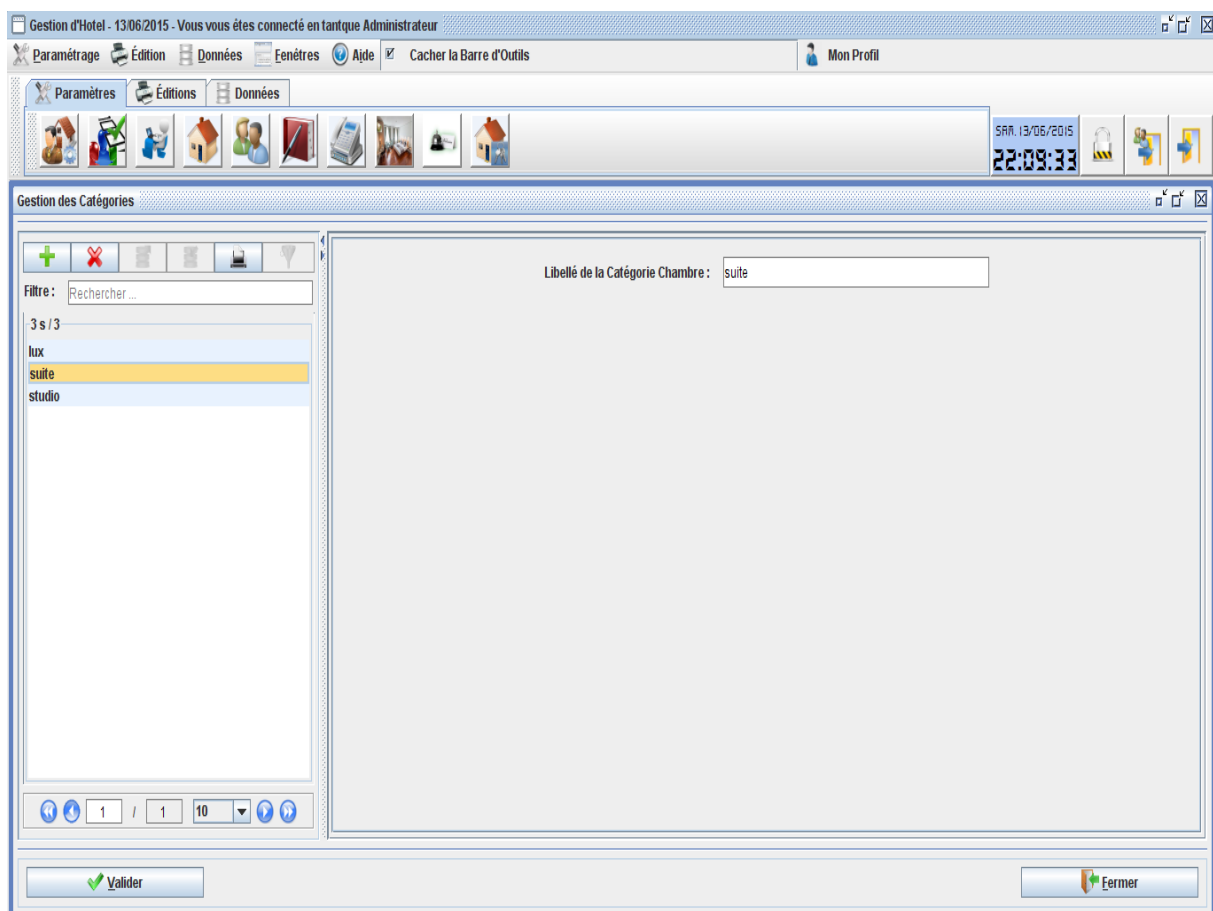


FIGURE C.1 – Interface de gestion des catégories.

C.0.6 Interface gestion des consommations

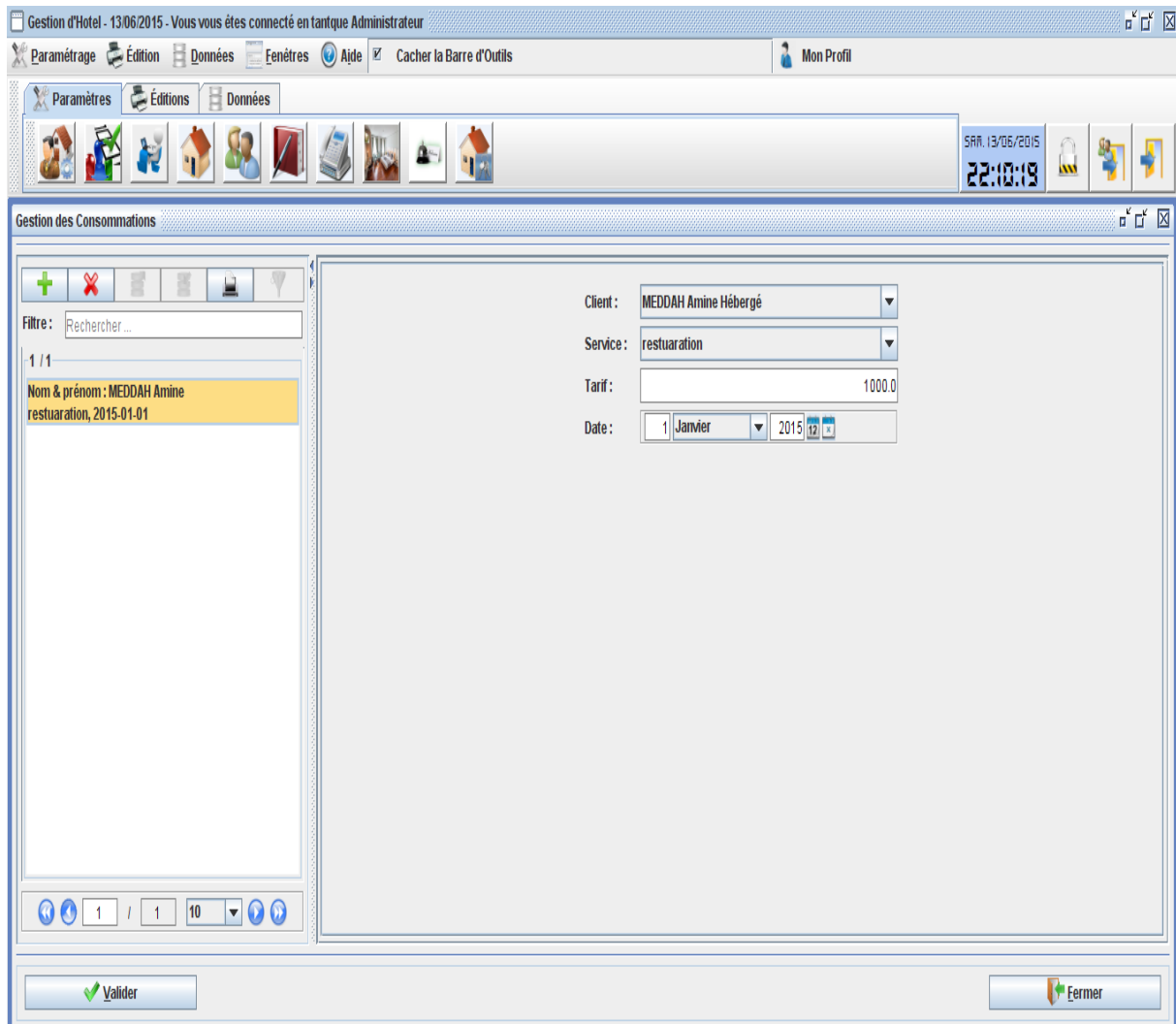


FIGURE C.2 – Interface de gestion des consommations.

C.0.7 Interface gestion des réservations

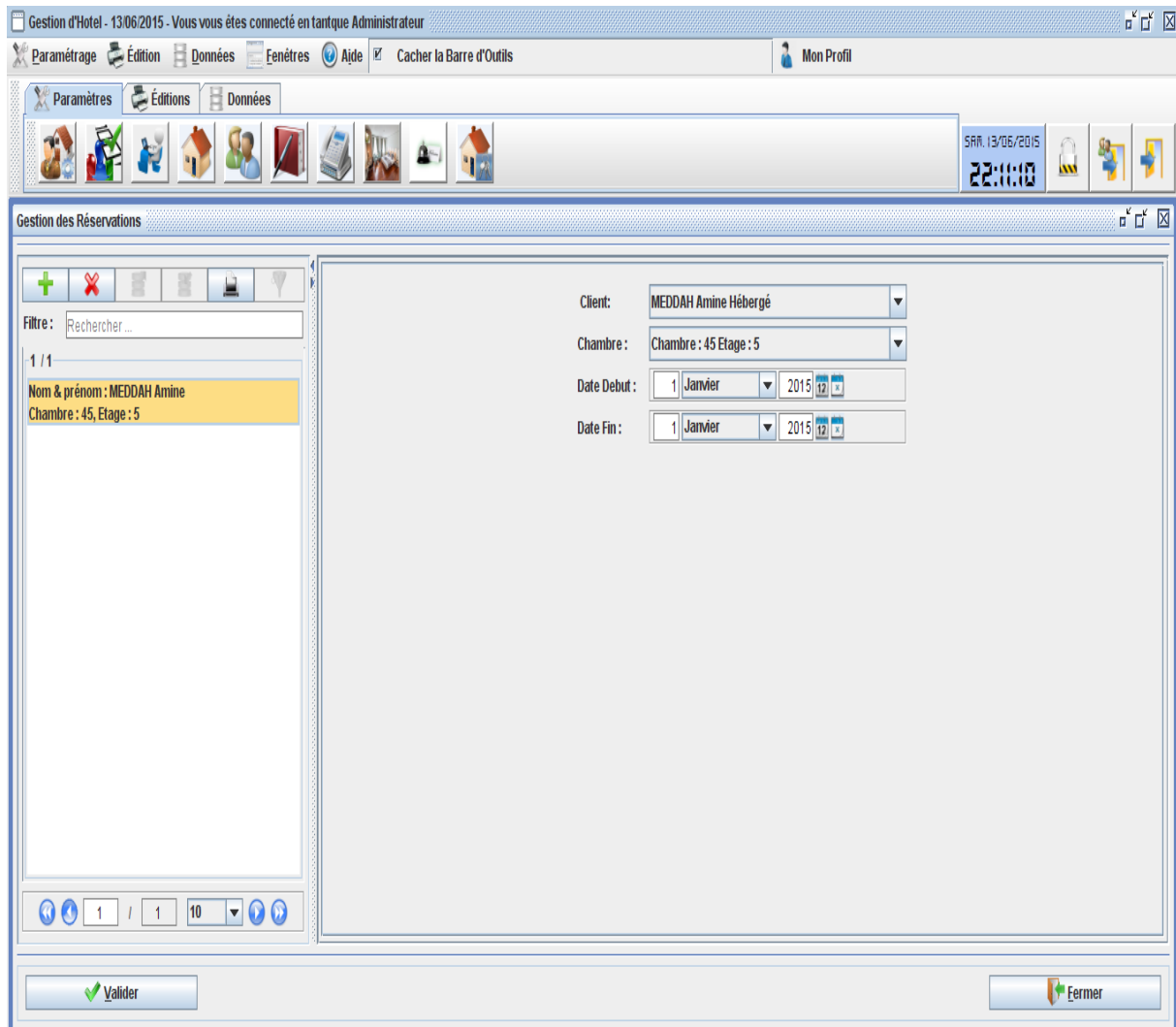


FIGURE C.3 – Interface de gestion des réservations.

Résumé

Aujourd'hui, l'informatique a atteint une prodigieuse évolution technologique dans différents domaines (réseaux informatiques, bases de données...). Cette évolution est nécessaire pour remédier aux problèmes rencontrés dans la vie quotidienne. Automatiser des informations est l'un des rôles essentiels de l'informatique. C'est ceci qui nous a poussés à créer une application réseau pour la gestion d'un hôtel accessible par des utilisateurs (l'administrateur et autres utilisateurs privilégiés).

Notre application « Gest-Hôtel » permet à l'administrateur de l'hôtel de gérer les clients, les chambres, les services, les factures ainsi que les comptes. L'étude de l'existant a été faite au sein de l'hôtel « CRISTAL ». Pour garantir l'achèvement de l'application nous avons choisi de modéliser avec le langage UML2, choix dicté par la simplicité et la performance de celui-ci en matière de conception. MySQL est le serveur de base de données de l'application et cette dernière a été développée en utilisant différents outils informatiques tel que, WampServer, PHP-MyAdmin, le JDK, eclipse etc. Le langage de programmation utilisé est le Java.

Mots-clés : Gestion d'un hôtel, Application réseau, Base de données, UML2, MySQL, Wamp-Server, Java.

Abstract

Today, computer science has reached a tremendous technological development in different fields (computer networks, databases ...). This is necessary to address the problems in now a days life. Automate information processing is one of the essential roles of the computer. It is this that led us To create a network application for the management of a hotel accessed by users(administrator and other privileged users).

Our application « Gest-Hôtel » allows the administrator to manage the hotel customers, rooms, services, invoices and accounts. The study of existing was made in the hotel "CRISTAL" To ensure the completion of the application we have chosen to model with UML2 language, that has been choice dictated by the simplicity and performance of it in the design process. MySQL server database application and it has been developed using different tools such as computer, WampServer, PHPMyAdmin, JDK, Eclipse etc. The programming language used is java.

Keywords : management of a hotel, Network Application, Database, UML2, MySQL, Wamp-Server, Java.