

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Bejaïa

Faculté des Sciences Exactes

Département d'Informatique



*Mémoire de Master Professionnel*

En Informatique

*Option*

Génie Logiciel

*Thème*

---

# Conception et réalisation d'une application réseau pour la gestion de l'hôtel ROYALE en JAVA FX

---

Soutenu devant le jury composé de :

Président Mr LARBI Ali

Encadreur Mr OUZEGGANE Radouane

Examinatrice Mme GHERBI Meriem

Réalisé par :

Mr BOUYOUCHEF Tahar

Mr TENSAOUT Lyes

Promotion 2019/2020

# REMERCIEMENT

---

*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 honorés 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 tenons 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, leurs sacrifices et leurs patiences.

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

# DEDICACE

---

Ce modeste travail qui n'est que la conjugaison des énormes efforts accompli doit tout d'abord constituer un vibrant hommage à mon père et mon frère et ma grand-mère que dieu les accueille dans son vaste paradis.

En deuxième lieu je tiens à le dédier à ma famille qui m'a toujours soutenu et épaulé en particulier ma mère qui en permanence était ma fervente supportrice et mon pilier de principal.

A tous les amis avec qui j'ai fait ma scolarité en spécialement ceux que j'ai connu sur les bancs des amphis et ceux que avec qui j'ai partagé la joie, le malheur et les conditions de vie difficile au sein de la **R.U.T.O** sans exception qu'ils soient là ou ailleurs ou dans l'au-delà.

Une mention spéciale à mes camarades de lutte à tous les militants, activistes sincères que j'ai connus tout au long de mon passage à l'université et avec qui j'ai appris le sens de l'université « vive la lutte ».

A toutes les structures associatives de l'université de Vgayet en particulier à mon second foyer et ma deuxième famille « **Tikti. Tampusni. Tadukli** » et tous ses membres par excellence, à mon école de lutte « **Coordination Locale des Etudiants** » que je salue au passage l'engagement de ses membres pour une université meilleure.

A toi **B.B** qui malgré la distance, mes bourdes et bêtises a répétions tu as toujours était à mes cotes avec tes encouragements durant les périodes les plus difficile de ma vie que tu as su transformer par fois en bonheur, Merci.

Tensaout Lyes

# DEDICACE

---

*Je dédie ce modeste travail à ma famille et particulièrement à mes Parents pour leurs soutiens qu'ils m'ont accordés tout au long mon chemin.*

*A mes amis que j'ai côtoyé au sein de l'association « Tikti.Tamusni.Tadukli » et « Amzday.Adelsan.Inelmaden » ainsi que toutes ces belles personnes que j'ai eu l'immense bonheur de connaître au cours de ces années à l'université.*

*Une dédicace spéciale à Manel, Lynda, Lyes, Zahir, Ghani et Youva .*

*A mes amis intimes Sir Wahid, Walid, Salim's, Saddek, Aissa et Nadir .*

*Bouyoucef Tahar*

# TABLE DES MATIERES

<b>TABLE DES MATIERES</b> .....	<b>I</b>
<b>LISTE DES FIGURES</b> .....	<b>VI</b>
<b>LISTE DES TABLEAUX</b> .....	<b>IX</b>
<b>LISTE DES ABREVIATIONS</b> .....	<b>X</b>
<b>Introduction Générale</b> .....	<b>1</b>
<b>Chapitre I : Organisme d'Accueil et Capture des Besoins</b> .....	<b>5</b>
<b>Introduction</b> .....	<b>5</b>
<b>I.1.Présentation de l'hôtel Royale</b> .....	<b>5</b>
<b>I.2.Présentation des composants phares de l'établissement</b> .....	<b>7</b>
I.2.1.Hébergement.....	7
I.2.2.Restauration .....	7
I.2.3.Salle des fêtes .....	7
<b>I.3.Organigramme de l'organisme d'accueil</b> .....	<b>8</b>
I.3.1.Définition.....	8
I.3.2.Intérêt de l'organigramme.....	8
I.3.3.Organigramme de l'hôtel« ROYAL » .....	9
<b>I.4.Etude de l'existant</b> .....	<b>10</b>
I.4.1.Etude des postes de travail et les taches à exécuter .....	11
<b>I.5.Etude des documents</b> .....	<b>13</b>
<b>I.6.Présentation de l'étude</b> .....	<b>14</b>
I.6.1.Présentation du sujet.....	14
I.6.2.Problématique .....	15
I.6.3.Objectif de l'étude .....	15
<b>I.7.Recueil des besoins</b> .....	<b>16</b>

I.7.1.Besoins fonctionnels.....	16
I.7.2.Besoins non fonctionnels.....	18
<b><i>I.8.Démarche et Métrologie .....</i></b>	<b>18</b>
<b><i>Conclusion.....</i></b>	<b>19</b>
<b><i>Chapitre II : Analyse des Besoins .....</i></b>	<b>21</b>
<b><i>Introduction.....</i></b>	<b>21</b>
<b><i>II.1.Identification des acteurs .....</i></b>	<b>21</b>
<b><i>II.2.Diagramme de contexte.....</i></b>	<b>23</b>
<b><i>II.3.Cas d'utilisation.....</i></b>	<b>27</b>
<b><i>II.4.Diagramme de cas d'utilisation .....</i></b>	<b>28</b>
<b><i>II.5.Identification des cas d'utilisations.....</i></b>	<b>29</b>
<b><i>II.6.Description textuelle des cas d'utilisation.....</i></b>	<b>32</b>
<b><i>II.7.Cas d'utilisation concernant l'utilisateur.....</i></b>	<b>34</b>
II.7.1.Diagramme d'activité du cas d'utilisation «s'authentifier» .....	35
<b><i>II.8.Cas d'utilisation concernant le Réceptionniste.....</i></b>	<b>36</b>
II.8.1.Diagramme d'activité du cas d'utilisation «gérer les clients».....	39
<b><i>II.9.Cas d'utilisations concernant le chef de réception .....</i></b>	<b>39</b>
II.9.1.Diagramme d'activité du cas « gérer les factures ».....	42
<b><i>II.10.Cas d'utilisation concernant le maître d'hôtel .....</i></b>	<b>42</b>
II.10.1.Diagramme d'activité du cas « gérer les tables & salle ».....	45
<b><i>II.11.Cas d'utilisation concernant le directeur .....</i></b>	<b>45</b>
II.11.1.Diagramme d'activité du cas « gérer les utilisateurs ».....	48
<b><i>II.12.Modélisation dynamique .....</i></b>	<b>48</b>
<b><i>II.13.Diagramme de séquence.....</i></b>	<b>49</b>
II.13.1.Diagramme de séquence du cas d'utilisation «s'authentifier».....	49
II.13.2.Diagramme de séquence du cas d'utilisation « Gérer les Réservations (table, chambre, salle des fêtes)» .....	51
II.13.3.Diagramme de séquence du cas d'utilisation «Gérer les utilisateurs» .....	54
<b><i>Conclusion.....</i></b>	<b>56</b>

<i>Chapitre III : Conception et Elaboration du Schéma Relationne .....</i>	<i>56</i>
<i>Introduction.....</i>	<i>56</i>
<i>III.1.Diagramme de séquence d'interaction.....</i>	<i>56</i>
<i>III.3.Diagramme d'interaction pour le cas d'utilisation «s'authentifier» ...</i>	<i>58</i>
<i>III.4.Diagramme d'interaction pour le cas d'utilisation « gérer les utilisateurs » .....</i>	<i>60</i>
<i>III.5.Diagramme de classes de domaine .....</i>	<i>61</i>
III.5.1.Présentation du Diagramme de classes.....	61
III.5.2.Concepts du diagramme de classe .....	61
III.5.3.Diagramme de classe.....	63
III.5.5.Schéma Relationnel.....	67
<i>Conclusion.....</i>	<i>70</i>
<i>Chapitre IV : Réalisation .....</i>	<i>71</i>
<i>Introduction.....</i>	<i>71</i>
<i>IV.1.Environnement de Programmation et Bibliothèques.....</i>	<i>71</i>
IV.1.1.WampServer .....	71
IV.1.2.Eclipse .....	72
IV.1.3.SceneBuilder .....	72
IV.1.4.FXML.....	72
IV.1.5.JDBC (Java Database Connectivity) .....	72
IV.1.6.SQL (Structured Query Language) .....	73
IV.1.7.PHPMyAdmin .....	73
IV.1.8.JAVA.....	73
IV.1.9.JAVAFX.....	74
<i>IV.2.Schéma physique de la base de données.....</i>	<i>74</i>
<i>IV.3.Architecture de l'application.....</i>	<i>74</i>
<i>IV.4.Diagramme de déploiement.....</i>	<i>76</i>
<i>IV.5.Structure du code source.....</i>	<i>76</i>
<i>IV.6.Arborescence de l'application.....</i>	<i>77</i>
<i>IV.7.Aspect graphique de l'application .....</i>	<i>79</i>

IV.7.1. Logo de l'établissement .....	79
IV.7.2. Capture d'écran .....	80
IV.7.2.1. Interface d'authentification .....	80
IV.7.2.2. Tableau de Bord .....	81
IV.7.2.3. Ajouter une réservation de chambre .....	82
<b>Conclusion .....</b>	<b>83</b>
<b>Conclusion générale .....</b>	<b>84</b>
<b>Bibliographie .....</b>	<b>X</b>
<b>Annexe .....</b>	<b>X</b>
A Cas d'utilisation .....	X
A.0.1 Cas d'utilisation « gérer les réservations de chambre » .....	X
A.0.2 Description textuelle du cas d'utilisation « gérer les réservations de chambres » .....	XI
A.0.3 Diagramme d'activité « gérer les clients » : .....	XII
A.0.4 Cas d'utilisation « gérer les chambres » .....	XIII
A.0.5 Description textuelle du cas « Gérer les chambres » .....	XIV
A.0.6 Diagramme d'activité de « la gestion des chambres » .....	XV
A.0.7 Cas d'utilisation « Gérer les tarifs » .....	XVI
A.0.8 Description textuelle du cas « Gérer les tarifs » .....	XVI
A.0.9 Cas d'utilisation « Gérer les réservations de salle des fêtes » .....	XVII
A.0.10 Description textuelle du cas « Gérer réservation de salle des fêtes » .....	XVIII
A.0.11 Diagramme d'activité « Gérer les tables & salle » .....	XIX
A.0.12 Diagramme de Séquence « Gérer les clients » .....	XX
A.0.13 Diagramme de séquence « Gérer les factures » .....	XXI
A.0.14 Diagramme de séquence « Imprimer une facture » .....	XXII
B Diagramme de séquence d'interaction .....	XXIII
B.0.1 Diagramme de séquence d'interaction « Gérer les clients » .....	XXIII
C Installation & Interfaces .....	XXIV
C.0.1 Installation du module JavaFX .....	XXIV
C.0.2 Les composantes de l'interface graphique de JavaFX : .....	XXV
C.0.3 Interface ajouter une facture .....	XXVI
C.0.4 Interface tableau de bord Maître d'hôtel .....	XXVI

C.0.5	Interface Modifier une réservation de salle.....	XXVII
C.0.6	Interface gestion des tables .....	XXVII
C.0.7	Interface Imprimer une facture .....	XXVIII

# LISTE DES FIGURES

Figure I.1-Organigramme de l'Hôtel ROYALE.....	10
Figure II.1 Diagramme des utilisateurs.....	23
Figure II.2- Diagramme de Contexte.....	23
Figure II.3 Diagramme de cas d'utilisation global.....	28
Figure II.4 Cas d'utilisation « s'authentifier ».....	34
Figure II.5 Diagramme d'activité de l'authentification.....	36
Figure II.6 Cas d'utilisation « Gérer les clients ».....	37
Figure II.7 Diagramme d'activité « Gérer les clients ».....	39
Figure II.8 Cas d'utilisation « Gérer les factures ».....	40
Figure II.9 Diagramme d'activité « Gérer les factures ».....	42
Figure II.10 Cas d'utilisation « Gérer les tables & salle ».....	43
Figure II.11 Diagramme d'activité « Gérer les réservations de table & salle ».....	45
Figure II.12 Cas d'utilisation « Gérer les utilisateurs ».....	46
Figure II.13 Diagramme d'activité « Gérer les utilisateurs ».....	48
Figure II.14 Diagramme de Séquence « S'authentifier ».....	50
Figure II.15 Diagramme de Séquence « Gérer les Réservation de chambre ».....	53
Figure II.16 Diagramme de Séquence « Gérer les utilisateurs ».....	55
Figure III.1 Représentation d'IHM.....	57
Figure III.2 Représentation des contrôleurs.....	57
Figure III.3 Représentation des entités.....	58
Figure III.4 Diagramme de séquence détaillé « s'authentifier ».....	58
Figure III.5 Diagramme de séquence détaillé « Gérer les uses ».....	60

Figure III.6 Diagramme de classes des utilisateurs.....	63
Figure III.7 Diagramme de classes du domaine.....	63
Figure III.8 Représentation de la règle de passage N°1 .....	67
Figure III.9 Représentation de la règle de passage N°2 .....	68
Figure IV.1 Schéma physique de la base de données.....	75
Figure IV.2 Architecture du code.....	77
Figure VI.3 Arborescence des interfaces de l'application.....	78
Figure VI.4 Nouveau logo de l'hôtel ROYALE .....	79
Figure VI.5 Interface d'authentification .....	80
Figure VI.6 Tableau de bord du chef de réception.....	81
Figure VI.7 Interface d'ajout d'un client.....	82
Figure A.1 Cas d'utilisation « gérer les réservations de chambre ».....	XI
Figure A.2 Diagramme d'activité « gérer les clients » .....	XII
Figure A.3 Cas d'utilisation « gérer les chambres ».....	XIII
Figure A.4 Diagramme d'activité « Gérer les chambres ».....	XV
Figure A.5 Cas d'utilisation « gérer les tarifs » .....	XVI
Figure A.6 Cas d'utilisation « gérer les réservations de salle » .....	XVII
Figure A.7 Diagramme d'activité « gérer les réservations de salle » .....	XIX
Figure A.8 Diagramme de Séquence « Gérer les clients » .....	XX
Figure A.9 Diagramme de Séquence « Gérer les factures ».....	XXI
Figure A.10 Diagramme de Séquence « Imprimer une facture » .....	XXII
Figure B.1 Diagramme de séquence d'interaction « Gérer les clients » .....	XXIII

Figure C.1 Interface ajouter une facture .....	XXVI
Figure C.2 Interface tableau de bord Maître d'hôtel.....	XXVI
Figure C.3 Interface Modifier une réservation de salle .....	XXVII
Figure C.4 Interface gestion des tables.....	XXVII
Figure C.5 Interface Imprimer une facture.....	XXVIII

# LISTE DES TABLEAUX

Table I.1: Représentation des infrastructures de l'hôtel ROYALE .....	7
Table I.2 : Les taches exécutées par le poste 1.....	11
Table I.3 : Les taches exécutées par le poste 2.....	12
Table I.4 : Les taches exécutées par le poste 3.....	12
Table I.5 : Les taches exécutées par le poste 4.....	13
Table II.1-Identification des acteurs système .....	22
Table II.2 Identification des messages échangés entre les utilisateurs et le système	27
Table II.3 identification des cas d'utilisation.....	32
Table II.4 Formalisme tabulaire utilisé.....	33
Table II.5 Représentation de la description textuelle du cas « s'authentifier »....	35
Table II.6 Représentation de la description textuelle du cas « Gérer les client »	38
Table II.7 Représentation de la description textuelle du cas « Gérer les factures »	41
Table II.8 Représentation de la description textuelle du cas « Gérer les réservations de table & salle » .....	44
Table II.9 Représentation de la description textuelle du cas « Gérer les utilisateurs » .....	47
Table III.1 Représentation des éléments composant le diagramme de séquence « s'authentifier » .....	59
Table III.2 Représente les propriétés et méthodes de chaque classe.....	66

## LISTE DES ABBREVIATIONS

<b>API</b>	Application Programming Interface
<b>BDD</b>	BDD Base De Données
<b>F</b>	Fiche
<b>FXML</b>	Format eXtensible Markup Language
<b>FC</b>	Fiche Client
<b>FP</b>	Fiche Police
<b>IDE</b>	Integrated Development Environment
<b>IHM</b>	Interface Homme Machine
<b>JDBC</b>	Java Data Base Connectivity
<b>JDK</b>	Java Development Kit
<b>JRE</b>	Java Runtime Environment
<b>MLDR</b>	Modèle Logique de Données Relationnelles
<b>MVC</b>	Model View Controller
<b>MySQL</b>	My Structured Query Language
<b>PC</b>	Personnel Computer
<b>PHP</b>	Hypertext Preprocessor
<b>PHPMyAdmin</b>	Personal Home Page My Administration
<b>SD</b>	Sequence Diagram
<b>SDK</b>	Software Development Kit
<b>SGBD</b>	Système de Gestion de Base de Données
<b>SQL</b>	Structured Query Language
<b>UML</b>	Unified Modeling Language
<b>UP</b>	Unified Process
<b>WAMP</b>	Windows Apache MySQL PHP

**Introduction**

**Générale**

# INTRODUCTION GENERALE

---

La démocratisation de l'informatique et de ces métiers a permis de faciliter l'accès à d'innombrables services aux clients désirants les exploiter, ces derniers peuvent être atteints via les différents supports (Application web, Mobile...etc.) alloués à des fins commerciales pour nouer des liens étroits avec la clientèle et les fidéliser ou pour simplifier le quotidien de la population on leur offrant une alternative plus accessible et plus rapide et surtout plus efficaces comme viens de le démontrée la crise sanitaire de la pandémie mondiale du "Covid-19".

Au milieu du même registre et dans les secteurs de production qui se sont distingués pour être parmi les premiers à adopter cette nouveauté, l'informatisation dans ces secteurs s'apparente visiblement via les logiciels et les applications de gestion et d'archivage qui ont apporté un saut qualitatif dans l'organisation du travail, l'amélioration du système d'information(SI) des entreprises , une gestion plus efficace de la ressource humaine et des différents documents qui sont devenu numérique notamment pour facilite les recherches ainsi que les partages pour les collaborateurs de l'entreprise sans pour autant accroître la complexité de la procédure d'accès aux documents et réduire l'encombrement de

# INTRODUCTION GENERALE

---

ces derniers .ce bond s'affiche aussi dans l'automatisation de plusieurs tache tel que l'archivage qui fournit aux utilisateurs accrédités un environnement physique et informatique sécurisé garantissant une confidentialité, une intégrité et une pérennité pour l'ensemble des documents électroniques probants de l'entreprise .

Dans le cadre de notre étude dans laquelle nous avons constaté une multitude de problèmes dus à la gestion manuelle de l' « **HOTEL ROYAL-VGAYET** » ,dans une démarche innovatrice et pour pallier les nombreuses lacunes et faire face aux incommensurables contraintes de son système de gestion et d'archivage qui restait jusqu'à là, un souci major du fait des abyssales réclamations de la clientèle et la perte de temps énorme dans les différentes tâches. La direction de cette structure constatant les failles de ce système ont pris la décision de moderniser ce dernier en se concordant aux nouvelles technologies est passé vers l'informatisation des document et l'automatisation de certaines tâches(gestion des réservations ,gestion des clients...etc.) Pour améliorer son rendement et accélérer le processus lié à l'archivage, la sauvegarde, la recherche et le traitement des données et des requêtes émanant des clients dans le but de faciliter la tâche du personnel.

Pour une meilleure compréhension de notre étude, nous avons organisé notre travail comme suit :

Le premier chapitre intitulé « Organisme d'accueil et capture des besoins » sera dédié à la présentation de la structure d'accueil ses infrastructures et son organigramme ainsi qu'une vue globale du système existant. Par la suite nous allons soulever les problèmes liés à ce système de gestion pour lequel nous allons exposer notre solution qui consiste en la réalisation d'une application de gestion sous réseau en répondant aux besoins recueillis.

En deuxième lieu, l'analyse des besoins sera l'objet de notre chapitre

# INTRODUCTION GENERALE

---

considéré comme le noyau de notre étude du fait qu'il portera sur la modélisation dynamique dans laquelle nous allons identifier les acteurs qu'interagissent avec le système à développer puis recenser leurs cas d'utilisation et les décrire textuellement . Pour terminer par la présentation des diagrammes de séquence système.

La troisième partie de notre travail sera consacrée à la modélisation statique ainsi nous allons détailler la conception de notre système à travers les diagrammes d'interaction, puis nous passerons à la réalisation du diagramme de classes qui sera transformé en schéma en appliquant les règles de passage pour avoir une vue sur notre base de données.

Le quatrième chapitre et dernier sera porté sur la réalisation dans lequel nous présenterons les multiples techniques et langages de développement utilisés dans ce travail. Puis nous exposerons l'aspect graphique de notre application à travers le logo et les différentes interfaces de notre application.

Au final nous allons clôturer cette étude par une conclusion générale, des perspectives et un résumé des connaissances acquises suite à ce travail.

# CHAPITRE I

ORGANISME D'ACCUEIL

ET CAPTURE DES

BESOINS

# CHAPITRE I

## ORGANISME D'ACCUEIL ET CAPTURE DES BESOINS

---

### **Introduction**

Ce chapitre comporte la présentation globale de l'organisme d'accueil qui est « Hôtel ROYALE » et des différentes structures qui le composent ainsi que le traitement de son organigramme, suivi par un aperçu intégral de la situation informatique actuelle de l'hôtel à divers niveaux qui le compose, en clôture de ce chapitre la problématique ainsi que le but de cette étude seront également évoquer en détail dans cette partie.

### **I.1.Présentation de l'hôtel Royale**

L'hôtel ROYALE l'un des pionniers du domaine de l'hôtellerie au niveau de la wilaya de Bejaïa, son expérience fait de lui l'un des plus prestigieux au vu des services qu'il propose à sa clientèle, il se situe au quartier de « Ihhedadhen » non loin du campus universitaire de « Targa Ouzemour » et quelque minutes du port et de l'aéroport de Bejaia. Cet hôtel prestigieux mit au service de ses nombreux clients différentes infrastructures pour un séjour agréable notamment une grande piscine et un jardin botanique, un terrain et une salle de sport pour les clubs sportifs, une salle de réunion pour les séjours professionnels.

Le confort du client est la priorité de l'établissement pour cela une connexion « Wifi », une climatisation sont mis à la disposition des visiteurs,

l'hôtel dispose aussi de plusieurs types de chambre (chambre simple ou double, suite, appartement) ainsi que d'une pizzeria, restaurant, cafeteria et un Bar pour se restaurer et une discothèque et un mini Bar pour assurer un séjour plaisant.

Disposition des infrastructures de l'hôtel Royal :

<b>Niveau 0-RED</b>	<ul style="list-style-type: none"> <li>• Parking avec une capacité de 120 véhicules</li> <li>• Deux meubles de dix étages chacun</li> <li>• Salle des fêtes</li> <li>• Installation technique (chaudière, Bâche a eau, groupe électrogène...etc.)</li> <li>• Pizzeria</li> <li>• Piscine</li> <li>• Jardin</li> <li>• 2 grandes piscines</li> <li>• Magasin</li> </ul>
<b>Niveau0-RDC</b>	<ul style="list-style-type: none"> <li>• Réception et hall de réception</li> <li>• Services administratifs : <ul style="list-style-type: none"> <li>✓ Service finance et comptabilité</li> <li>✓ Service personnel</li> <li>✓ Service approvisionnement</li> <li>✓ Hébergement du personnel</li> <li>✓ Buanderie</li> </ul> </li> </ul>
<b>Niveau1-1<sup>er</sup> Etage</b>	<ul style="list-style-type: none"> <li>• Restaurant</li> <li>• Cuisine</li> <li>• Pizzeria</li> <li>• Salle de conférence</li> <li>• Salle de réunion</li> <li>• Salle d'exposition</li> </ul>

	<ul style="list-style-type: none"> <li>• Cafétéria</li> </ul>
<p style="text-align: center;"><b>Niveau3-2ème au 10ème Etage</b></p>	<ul style="list-style-type: none"> <li>• Trois (03) appartements</li> <li>• Dix chambres doubles</li> <li>• Onze chambres triples</li> <li>• Sept suites séniors</li> <li>• Trois suites juniors</li> <li>• Quarante chambres simples</li> </ul>

Table I.1: Représentation des infrastructures de l'hôtel ROYALE

## I.2.Présentation des composants phares de l'établissement

### I.2.1.Hébergement

Les pièces et les appartements sont équipés de toutes les commodités et accessoires nécessaires pour un séjour agréable au sein de l'établissement, les chambres offrent aussi une belle vue panoramique sur la ville de « Vgayet » ou sur le jardin botanique et la piscine de l'hôtel. La structure se compose de 73 chambres répartie comme l'indique le tableau ci-dessus.

### I.2.2.Restauration

Le restaurant se trouve au 1<sup>er</sup> étage de l'hôtel, doté d'une capacité d'accueil de 250 couvert et d'une vue appréciable qui surplombe la ville de Bejaia, au même niveau une cafeteria et une pizzéria sont à la disposition de la clientèle.

### I.2.3.Salle des fêtes

L'établissement renferme aussi une salle des fêtes aménagée spécialement pour tous les types de réceptions. Elle est dotée de tous les équipements nécessaires pour assurer des réceptions fastes, sa capacité d'accueil est de 300 places assises et d'une salle de restauration de 180 couverts.

### I.3.Organigramme de l'organisme d'accueil

#### I.3.1.Définition

L'organigramme est un schéma qui représente la structure de l'entreprise, c'est-à-dire son type d'organisation. Ce document est très utile car il permet de visualiser la répartition des tâches, les différents postes, le niveau hiérarchique et les responsabilités correspondantes. Ce dernier est élaboré par la DRH (Direction des Ressources Humaines) en collaboration avec le service Communication si celui-ci existe. Dans les établissements de plus petite taille, le Directeur peut s'en charger.

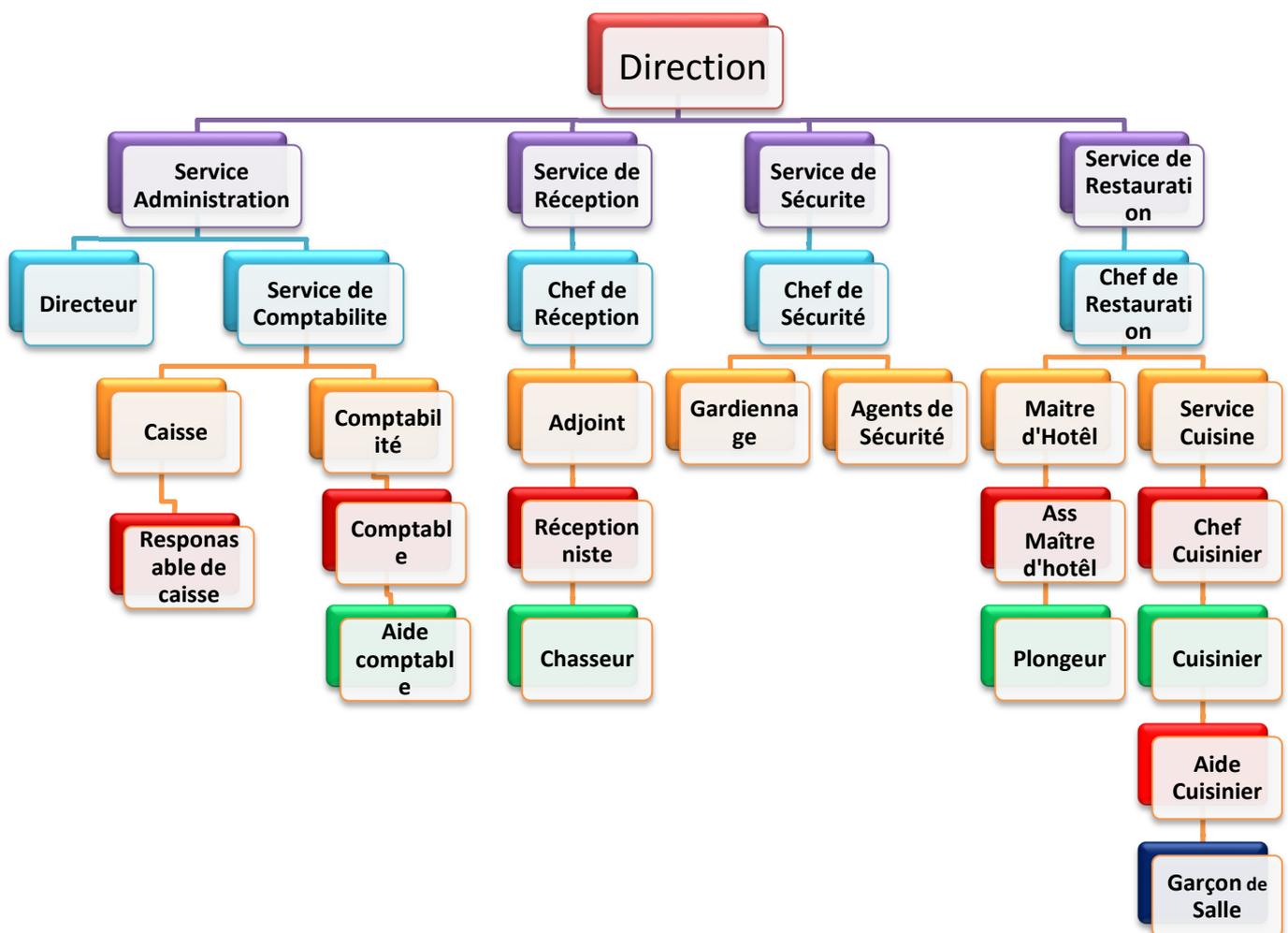
#### I.3.2.Intérêt de l'organigramme

- **Pour le personnel** : définir les postes et les responsabilités :  
Il permet de se situer, de savoir “qui fait quoi”, “qui dépend de qui”, et ainsi d'améliorer l'efficacité du travail.
- **Pour la Direction de l'établissement** : mettre en évidence des fonctions et des services : il peut être un moyen d'analyse car il peut faire apparaître des dysfonctionnements et inciter à une meilleure organisation.
- **Pour les usagers ou les partenaires** : déterminer les liaisons entre les responsables : Il permet de connaître les différents services et personnes de l'établissement, de les situer les uns par rapport aux autres et ainsi de s'adresser au service le plus approprié. C'est un moyen de communication externe.
  - Il permet aussi de visualiser les différentes relations qui composent la structure de l'établissement concerné, on distingue notamment deux (2) types de relation :

✚ Relations hiérarchiques (relations d'autorité) qui permettent la transmission des ordres, en traits pleins ; par exemple entre un éducateur-chef et un éducateur.

✚ Relations fonctionnelles (liées aux activités exercées) qui permettent la transmission des informations, le travail en groupe..., en pointillés ; par exemple : entre une secrétaire médicale et une secrétaire du service des Admissions

### I.3.3. Organigramme de l'hôtel « ROYAL »



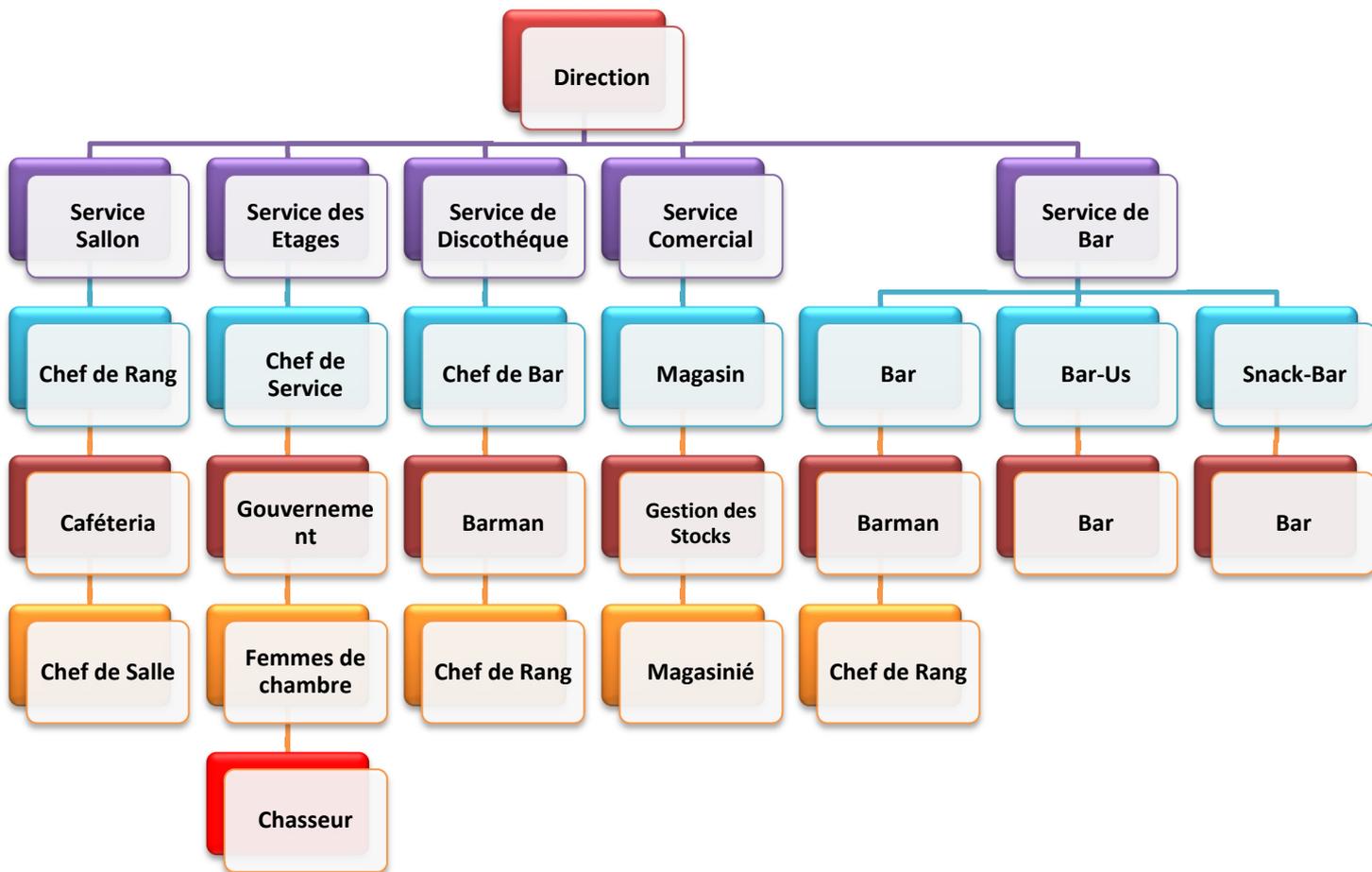


Figure I.1-Organigramme de l'Hôtel ROYALE

#### I.4.Etude de l'existant

Cette étude a pour objectif éclaircir de façon formelle le cheminement interne et externe des informations ainsi que le nombre de postes de travaux et déceler leurs surcharges pour aboutir aux avantages et inconvénients de l'organisation courante et pouvoir corriger ses lacunes qui freinent le bon fonctionnement de l'organisation de la structure en question .

### I.4.1. Etude des postes de travail et les tâches à exécuter

Le principe de cette étape est de capturer les principaux postes qui gèrent les différentes réservations que l'hôtel propose à ses clients par conséquent repérer les diverses tâches que le personnel rencontre régulièrement, ces dernières constituent des entraves à l'exécution des tâches et ralentissent le rendement de l'établissement.

#### a) Poste N°1

**Le responsable du poste :** Directeur.

**Service de rattachement :** La direction.

**Rôle :** Gestion de l'hôtel.

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

**Emplacement du poste :** 1<sup>er</sup> Etage

**Matériel utilisé :** Pc + imprimante + scanner.

#### LES TACHES EXECUTEES PAR LE POSTE N°1

N°	Désignation	Fréquence
01	Gère l'hôtel	Chaque jour

Table I.2 : Les tâches exécutées par le poste 1

#### b) Poste N°2

**Le responsable du poste :** Chef de Réception.

**Service de rattachement :** La Réception.

**Rôle :** Gérer la réception et recevoir les réservations via E-mail.

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

**Emplacement du poste :** La réception.

**Matériel utilisé :** Pc + imprimante + scanner.

**LES TACHES EXECUTEES PAR LE POSTE 2**

N°	Désignation	Fréquence
01	Etablissement de la fiche de police	Arrivé du client
02	Etablissement des factures	Arrivé du client
03	Envoyer le registre police	Chaque mois

Table I.3 : Les taches exécutées par le poste 2

**Remarque :** le réceptionniste s'occupe de travail de chef de réception lorsque ce dernier est absent.

**c) Poste N°3**

**Le responsable du poste :** Réceptionniste.

**Service de rattachement :** Réception.

**Rôle :** Gérer les clés magnétiques des chambres.

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

**Emplacement du poste :** La réception.

**Matériel utilisé :** Pc + imprimante + scanner.

**LES TACHES A EXECUTEES PAR LE POSTE 3**

N°	Désignation	Fréquence
01	Réception du client	Arrivé du client
02	Enregistrement des informations du client	Arrivé du client
03	Vérification la disponibilité des chambres.	A Chaque demande de réservation

Table I.4 : Les taches exécutées par le poste 3

**d) poste N°4 :**

**Le responsable du poste :** Maitre d'hôtel.

**Service de rattachement :** Restauration.

**Rôle :** Gérer le service de restauration.

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

**Emplacement du poste :** Restaurant.

**Matériel utilisé :** Pc + imprimante + scanner.

#### LES TACHES A EXECUTEES PAR LE POSTE 4

N°	Désignation	Fréquence
01	Récapitulation restaurant	Chaque commande
02	Vérification la disponibilité des tables et des salles.	Chaque commande

Table I.5 : Les taches exécutées par le poste 4

### I.5. Etude des documents

Est l'étude systématique des documents d'une organisation dans le but d'obtenir de l'information pour une évaluation

- N° 01 Code : FC

Désignation : Fiche Client

Nature de document : interne

Le document est-t-il rempli par : réceptionniste

Destiné pour : réceptionniste

Nombre d'exemplaires : 01.

- Etude du document N° 02 Code : FP

Désignation : Fiche Police

Nature de document : interne

Le document est-t-il rempli par: réceptionniste

Destiné pour: Police

Nombre d'exemplaires : 01

- Etude du document N° 03 Code : F

Désignation : Facture

Nature de document : interne

Le document est-t-il rempli par : chef de réception

Destiné pour: client

Nombre d'exemplaires : 02

## **I.6.Présentation de l'étude**

### **I.6.1.Présentation du sujet**

Des informations de nature diverse circulent au sein de l'hôtel, elles concernent les multiples postes rattachés aux nombreux services qui composent l'établissement, les données associées aux clients sont enregistrées depuis son arrivée jusqu'à son départ des lieux, pour assurer une bonne coordination et une meilleure transition des informations pour atteindre le rendement souhaité par la clientèle, nous nous avons penché sur cette question afin de mettre en œuvre un système informatique qui fera en sorte d'améliorer les tâches de gestion de l'hôtel et permettra une interaction rigoureuse entre les postes clés de l'établissement notamment le directeur ,chef-réception ,réceptionniste et le maître d'hôtel, ce qui donnera un meilleur pilotage à la structure.

### **I.6.2.Problématique**

La ville de Bejaïa est l'une des destinations préférée des touristes, elle attise la convoitise des algériens ou étrangers mais cette influence se réduit de saison estivales a une autre, l'une des causes de cette dégradation est les nombreuses imperfections que le secteur touristique, comme la majorité des infrastructures hôtelières de la ville Vgayet, l'hôtel ROYALE se trouve aussi confronter a un problème majeur qui consiste en les réservations qui deviennent de plus en plus dure qui engendre le cumul des informations fruit de la gestion et de la sauvegarde manuelle de ses dernières, qui sont la source de nombreuses anomalies tel que :

- Perte des documents.
- archivage catastrophique des données
- complexité dans la recherche des informations
- Difficultés des travailleurs à accomplir leurs taches correctement
- Enorme perte de temps

Ces problèmes sont souvent la racine des différents mécontentements des visiteurs ainsi que les travailleurs de l'établissement.

### **I.6.3.Objectif de l'étude**

L'objectif fondamental de ce travail est « La conception et la réalisation d'une application réseau pour la gestion de l'hôtel ROYALE », une application qui pourra palier aux nombreux problèmes de réservation que rencontre le personnel selon les besoins qui nous ont été cité comme :

- ✓ Assurer le suivi continu du client depuis son arrivé jusqu'à son départ
- ✓ Informatiser certaines tâches
- ✓ Améliorer la qualité du service en temps de réponse

- ✓ Eviter les erreurs fréquentes
- ✓ Rénover la manière de stocker et de gérer les données
- ✓ Parfaire l'infographie (logo ainsi que le slogan) .

En mettant en place une application suffisamment ergonomique et maniable par des personnes novices en informatique afin d'assurer une meilleure performance et un rendement à la hauteur de la renommée de l'hôtel ROYALE.

## **I.7.Recueil des besoins**

Nous avons focalisé notre travail sur la mise en œuvre d'une application réseau pour la gestion de l'hôtel ROYALE et l'automatisation de certaines tâches capital de l'établissement tel que :

- La gestion des clients de l'hôtel.
- La gestion des réservations et des infrastructures de l'établissement.
- La gestion des factures.

L'application offrira de multiples espaces spécifiques, ils contiendront les différentes opérations conformément au rôle qui caractérise les tâches définies, précédemment, chaque espace aura deux fonctionnalités majeures, la gestion et la consultation des données reliés à ses tâches après l'authentification.

### **I.7.1.Besoins fonctionnels**

Ils sont l'expression de ce que le produit ou le service délivré par le projet devrait être ou faire.

Les besoins ont été divisés en dix (10) grandes parties :

**1. L'authentification :** cette interface permet aux utilisateurs d'accéder à leurs espaces après le login.

**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 :** cet espace permet au directeur de créer ou de révoquer un compte utilisateur.

**4. Gestion des clients :** cette interface donne la possibilité au réceptionniste de gérer la clientèle de l'établissement ainsi que la consultation des informations de ces derniers

**5. Gestion des réservations de chambre :** cet espace permet au réceptionniste de gérer les réservations de chambres effectuées par les clients et les consultes.

**6. Gestion des chambres :** cette interface offre la possibilité au chef de réception de gérer les différentes chambres de l'hôtel.

**7. Gestion des tarifs :** cet espace permet au chef de réception de mettre à jour les prix des chambres.

**8. Gestion des factures :** cette interface donne la possibilité au chef de réception d'établir et de gérer les factures ainsi que les consultes et les imprimes.

**9. Gestion des tables et de la salle :** cet espace permet au maitre d'hôtel de gérer les tables du restaurant et la salle des fêtes.

**10. Gestion des réservations de table et de la salle :** cette interface donne la possibilité au maitre d'hôtel de gérer et de consulter les réservations concernant les tables et la salle.

**Remarque :** La gestion des tâches englobe trois(3) opérations l'ajout, la modification ainsi que la suppression.

### I.7.2. Besoins non fonctionnels

Ils sont des besoins en matière de performance liés à l'implémentation et à l'interopérabilité générale. Ils peuvent être fixés par le client, ou par le développeur.

Après avoir recensé les besoins fonctionnel qui représentent les grands axes de l'application, il faudrait aussi définir les besoins non fonctionnel qui sont tout aussi importants car ils agissent de façon direct sur la performance de logiciel et influencent le rendement de l'utilisateur, pour satisfaire ces besoins il faut assurer :

- **La fiabilité** : l'application doit être intolérante aux erreurs, et assurer la cohérence.
- **L'ergonomie** : les interfaces doivent être compréhensibles, les utilisateurs ne doivent fournir aucuns efforts particuliers pour la navigation entre les rubriques.
- **Sécurité** : elle doit garantir la confidentialité des données.
- **Portabilité** : l'application doit fonctionner sur n'importe quel terminal.
- **Le rendement et l'efficacité** : les temps de réponse doivent être conformes aux normes et raisonnables.

### I.8. Démarche et Métrologie

Pour la suite de notre travail, en ce qui concerne le formalisme à adopter, nous allons utiliser le langage de modélisation UML (Unified Modeling Language). Et en ce qui concerne les étapes d'analyse de conception, nous avons suivi une démarche alléger du processus UP bien détaillé en [1].

**Conclusion**

Dans ce premier chapitre, nous avons commencé par une présentation générale de l'organisme d'accueil qui est l'hôtel ROYALE, ainsi que toutes ses infrastructures, puis nous nous sommes intéressés aux différents postes de chaque service comme définit dans l'organigramme. Par la suite, nous nous sommes focalisés sur la problématique et fixé les objectifs à atteindre. Ce chapitre est conclu par le recueil des besoins qui seront analysés dans le chapitre suivant.

# CHAPITRE II

## ANALYSE DES BESOINS

# CHAPITRE II

## ANALYSE DES BESOINS

### Introduction

Dans ce chapitre nous allons aborder la phase d'analyse des besoins définie dans le chapitre précédent ainsi que l'étude de conception de notre application, pour bien engager notre étude nous allons au premier pas identifier et présenter les acteurs ainsi que leurs interactions avec le système afin de réaliser les différentes fonctionnalités prévues, elles seront modélisées sous forme de cas d'utilisation et chaque cas sera suivi de sa description textuelle ainsi que de son scénario nominal, et pour mieux cerner ces derniers des diagrammes de séquences seront réalisés afin de décrire les différentes interactions.

### II.1. Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant ou en recevant des messages susceptibles d'être porteurs de données, en ce qui concerne notre cas d'étude et selon les besoins recueillis, et pour avoir plus d'efficacité comme indiqué dans [19], un acteur abstrait « utilisateur » est associé au cas d'utilisation « s'authentifier »

Nous avons identifié les acteurs suivants :

<b>Acteur</b>	<b>Rôle</b>
L'utilisateur	est un rôle générique qui représente toute entité non authentifier par le système, la seule opération accessible à se rôle est l'authentification.
Réceptionniste	après avoir passé le contrôle d'accès le réceptionniste a les prérogatives suivantes : <ul style="list-style-type: none"> <li>• Gérer et consulter la liste des clients</li> <li>• Gérer et consulter les réservations de chambre</li> </ul>
Chef de réception	après l'authentification le chef de réception a accès aux opérations suivantes : <ul style="list-style-type: none"> <li>• Gérer les chambres</li> <li>• Gérer, consulter et imprimer les factures</li> <li>• Gérer les tarifs des chambres, tables et salle des fêtes</li> </ul>
Maître d'hôtel	cet acteur en s'authentifiant a comme tâches : <ul style="list-style-type: none"> <li>• La gestion et la consultation des réservations de table</li> <li>• La gestion des tables</li> <li>• La gestion de la salle des fêtes</li> </ul>
Directeur	en accédant à son espace après l'authentification le directeur a les privilèges suivants : <ul style="list-style-type: none"> <li>• Consulter la liste des clients</li> <li>• Consulter toutes les réservations dans tous les services</li> <li>• Gérer les utilisateurs de l'application</li> </ul>

Table II.1-Identification des acteurs système

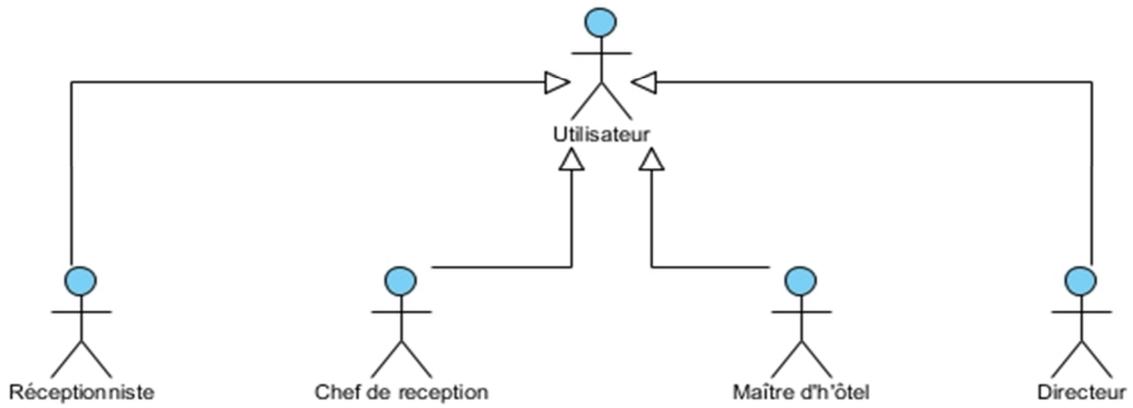


Figure II.1 Diagramme des utilisateurs

## II.2. Diagramme de contexte

Dans cette partie de notre travail, nous allons représenter le mouvement des données à l'intérieur du futur système et entre le système et ses éléments environnant, plus précisément ça portera sur la décomposition ainsi que l'identification des multiples messages échangés entre le système qui est considéré comme une boîte noire et les acteurs que nous avons identifié antérieurement .

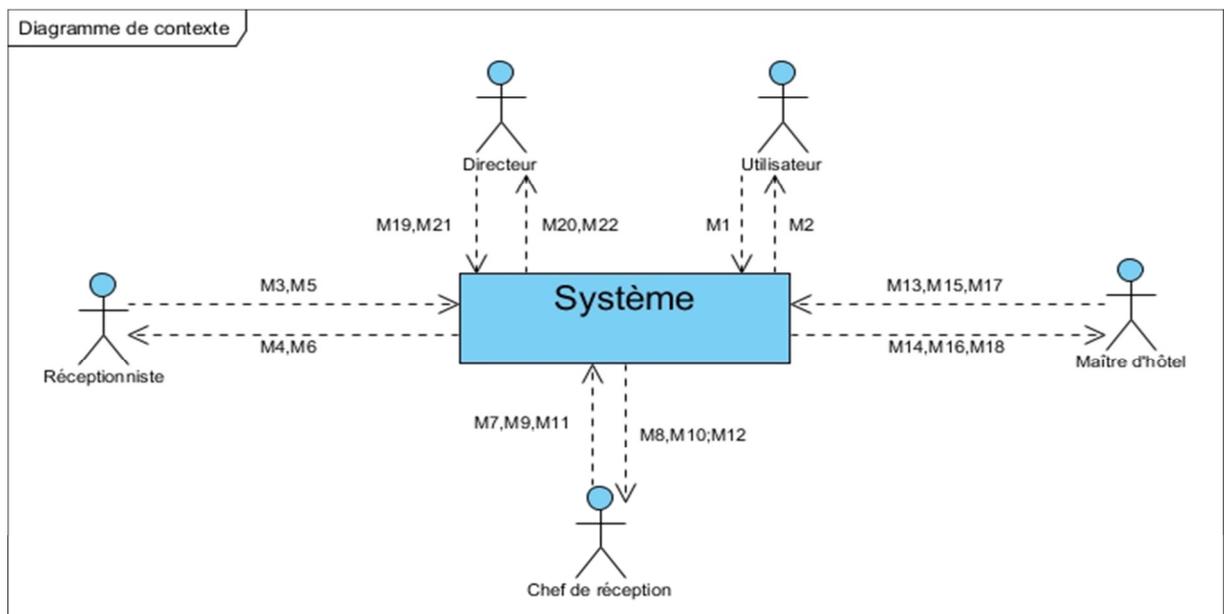


Figure II.2- Diagramme de Contexte

Les différents messages sont explicités dans le tableau suivant :

Acteur	Message Acteur -> Système		Message Système -> Acteur	
UTILISATEUR	M1	L'authentification (nom d'utilisateur et mot de passe).	M2	Réponse du système (soit erreur soit l'affichage du tableau de bord selon le rôle de chaque utilisateur.
	M3	Accès à la gestion des clients	M4	Affichage du formulaire d'ajout d'un client
			Affichage de la liste et des champs pour la modification d'un client	
RECEPTIONNISTE	M5	Accès à la gestion des réservations de chambre	M6	Affichage du formulaire d'ajout d'une réservation de chambre
				Affichage de la liste et des champs pour modifier une réservation de chambre
				Affichage de la liste des réservations pour la suppression

<b>CHEF DE RECEPTION</b>	<b>M7</b>	Accès à la gestion des factures	<b>M8</b>	Possibilité d'ajouter une facture via le formulaire d'ajout
				Affichage de la liste et des champs pour modifier une facture
				Affichage de la liste pour la suppression d'une facture
				Affichage de la liste des factures pour imprimer une.
	<b>M9</b>	Accès a la gestion des tarifs	<b>M10</b>	Affichage des tarifs des chambre, table et de la salle des fêtes avec possibilité de mettre à jour leur tarif via le champ indiqué
	<b>M11</b>	Accès a la gestion des chambres	<b>M12</b>	Possibilité d'ajouter une chambre à travers le formulaire d'ajout
			Affichage de la liste et des champs pour modifier une chambre	
			Affichage de la liste des chambres pour la suppression d'une de ces dernières	

Maitre d'hôtel	<b>M13</b>	Accès a la gestion des réservations de table	<b>M14</b>	Possibilité d'ajouter une réservation via le formulaire d'ajout
				Affichage de la liste des réservations de table ainsi que les champs pour une modification
				Affichage de la liste des réservations de table pour la suppression d'une de ces dernières
	<b>M15</b>	Accès a la gestion des réservations de salle	<b>M16</b>	Possibilité d'ajouter une réservation via le formulaire d'ajout
				Affichage de la liste des réservations de salle ainsi que les champs pour apporter une modification
				Affichage de la liste des réservations de salle pour la suppression d'une de ces dernières
<b>M17</b>	Accès a la gestion des tables	<b>M18</b>	Affichage de l'interface d'ajout	
			Affichage de l'interface de modification	
			Affichage de l'interface de suppression	

<b>Directeur</b>	<b>M19</b>	Demande de consultation des données	<b>M20</b>	Affichage du tableau de bord et des consultations disponibles
	<b>M21</b>	Accès a la gestion des utilisateurs	<b>M22</b>	Affichage de l'interface d'ajout d'un utilisateur Affichage de la liste des utilisateurs pour la suppression d'un utilisateur

**Table II.2 Identification des messages échangés entre les utilisateurs et le système**

### II.3.Cas d'utilisation

Selon [2] un cas d'utilisation (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Autrement les cas d'utilisation décrivent sous forme d'actions/réactions le comportement du système du point de vue de l'utilisateur, chaque cas représente une fonctionnalité du système visible depuis l'extérieur, les cas d'utilisations sont restreint a un cycle qui respecte trois principes, le déclenchement, le déroulement et la fin en réponse à une stimulation d'un acteur externe.

### II.4. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique [3].

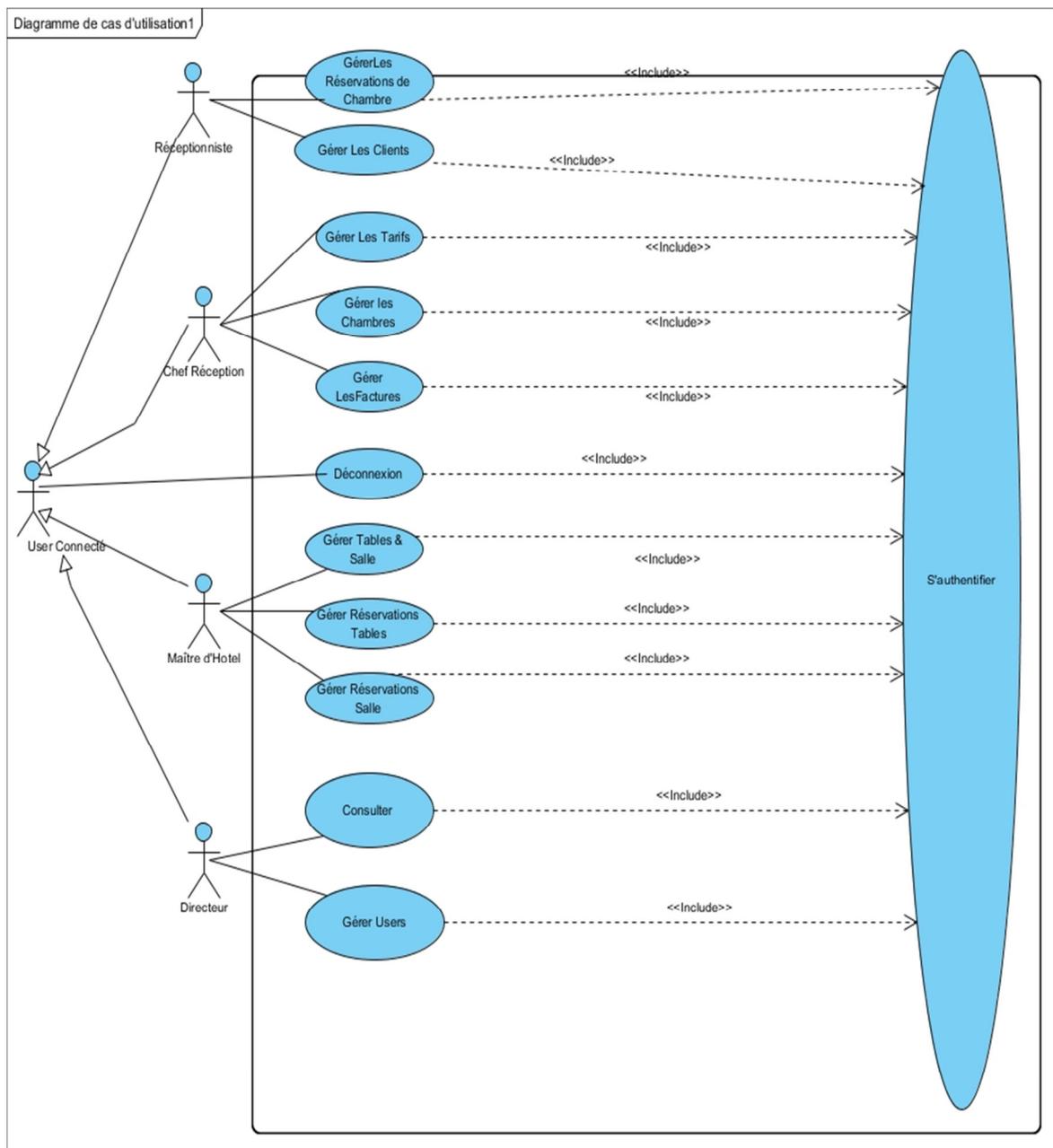


Figure II.3 Diagramme de cas d'utilisation global

**II.5. Identification des cas d'utilisations**

Dans cette section nous allons énumérer tous les cas d'utilisation qui concernent les acteurs du système identifié précédemment, Pour une meilleure présentation des cas d'utilisation, nous avons opté pour une structure tabulaire.

Le tableau suivant présente les différents cas d'utilisation identifié pour notre système :

Cas d'utilisation		Acteur Principal /Acteurs Secondaires	Messages Emis / Reçus par les acteurs
<i>S'authentifier</i>		<b>Utilisateur</b>	<b>Emet :</b> nom d'utilisateur et mot de passe  <b>Reçoit :</b> L'interface (tableau de bord) adéquate à l'utilisateur.
<i>Gestion des clients</i>	<b>Consulter</b>	<b>Réceptionniste</b>	<b>Emet :</b> Ajout, Modification, Suppression et consultation des informations liées aux clients.  <b>Reçoit :</b> Confirmation/échec d'ajout, de modification et de suppression.
	<b>Ajouter</b>		
	<b>Modifier</b>		
	<b>Supprimer</b>		
<i>Gestion des</i>	<b>Consulter</b>		<b>Emet :</b> Ajout,

<i>réservations de chambre</i>	Ajouter	<b>Chef de réception</b>		Modification, Suppression et consultation des réservations de chambre. <b>Reçoit :</b> Confirmation/échec d'ajout, de modification et de suppression.		
	Modifier					
	Supprimer					
<i>Gestion des factures</i>	Consulter					<b>Emet :</b> Ajout, Modification, Suppression, consultation et impression des factures. <b>Reçoit :</b> Confirmation/invalidité d'ajout, de modification et de suppression des factures.
	Ajouter					
	Modifier					
	Supprimer					
	Imprimer					
<i>Gestion des tarifs</i>	Table					<b>Emet :</b> Mise à jour du tarif d'une chambre, table, ou salle des fêtes. <b>Reçoit :</b> Confirmation/invalidité de la mise à jour.
	Salle					
	chambre					
<i>Gérer les chambres</i>	Ajouter					<b>Emet :</b> ajout, Modification, Suppression et consultation des
	Modifier					

	Supprimer	<b>Chef de réception</b>	chambres. <b>Reçoit :</b> Confirmation/invalidité d'ajout, de modification et de suppression.
	Consulter		
<i>Gérer les réservations de table</i>	Consulter		<b>Emet :</b> Ajout, Modification, Suppression et consultation des réservations de table <b>Reçoit :</b> Confirmation/invalidité d'ajout, de modification et de suppression d'une réservation de table.
	Ajouter		
	Modifier		
	Supprimer		
<i>Gérer les réservations de salle</i>	Ajouter	<b>Maitre d'hôtel</b>	<b>Emet :</b> Ajout, Modification, Suppression et consultation des réservations de la salle des fêtes <b>Reçoit :</b> Confirmation/invalidité d'ajout, modification et suppression d'une réservation de salle
	Modifier		
	Supprimer		
	Consulter		

<i>Gérer les tables et la salle</i>	Ajouter	<b>Directeur</b>	<b>Emet :</b> ajout, Modification, Suppression et consultation des tables. <b>Reçoit :</b> Confirmation/invalidité d'ajout, de modification et de suppression d'une table
	Modifier		
	Supprimer		
	Consulter		
<i>Gérer les utilisateurs</i>	Consulter	<b>Directeur</b>	<b>Emet :</b> ajout, Suppression et consultation des utilisateurs. <b>Reçoit :</b> Confirmation/invalidité d'ajout et de suppression d'un utilisateur.
	Ajouter		
	Supprimer		
<i>Consulter les différentes données</i>			<b>Emet :</b> Demande de consultation des données liées aux réservations. <b>Reçois:</b> les données

Table II.3 identification des cas d'utilisation

## II.6.Description textuelle des cas d'utilisation

D'après [4] les cas d'utilisation sont définis par une description textuelle, décrivant les objectifs et interactions entre le système et ses acteurs. Le format de présentation textuelle des cas d'utilisation est libre, mais il existe quelques propositions reconnues dans le domaine.

Chaque cas d'utilisation énoncé dans le tableau [Table II.3] sera décrit textuellement dans cette partie, on suivant le formalisme tabulaire [Table II.4] définit par Alistair Cockburn dans [5] est adapté au besoin de notre travail.

Pour consolider cette description textuelle nous allons les accompagner d'un diagramme d'activité qui va illustrer les scénarios décrit et le déroulement de chaque cas d'utilisation graphiquement.

<b>Numéro du cas d'utilisation</b>	<b>Nom du cas d'utilisation</b>
<b>Titre du cas</b>	L'intituler avec lequel on a défini le cas d'utilisation
<b>But</b>	L'objectif que le cas d'utilisation doit attendre
<b>Acteur</b>	Les personnes impliquées dans la réalisation de cas d'utilisation
<b>Pré conditions</b>	Les préalables à satisfaire avant le déclenchement du cas d'utilisation
<b>Scénario nominal</b>	La suite de séquence prescrite pour le cas d'utilisation
<b>Alternatives</b> <b>Exceptions</b>	Autres scénarios qui peuvent capable d'aboutir à un succès.  Suite d'actions qui mènent à l'échec du cas d'utilisation
<b>Besoins IHM</b>	Les éléments graphiques qui entrent dans l'interaction entre l'utilisateur et l'application durant le déroulement du cas d'utilisation

Table II.4 Formalisme tabulaire utilisé

## II.7.Cas d'utilisation concernant l'utilisateur

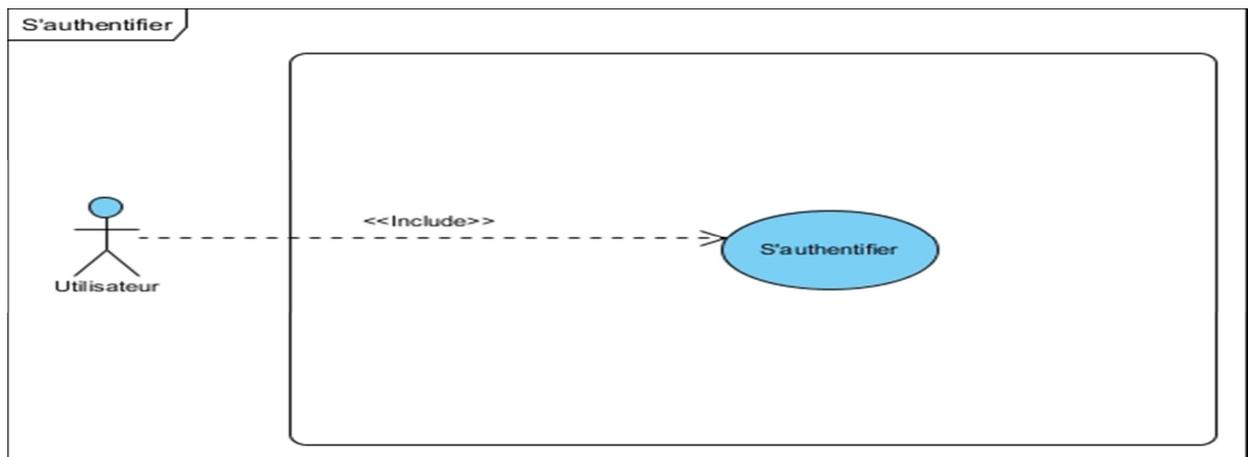


Figure II.4 Cas d'utilisation « s'authentifier »

Cas d'utilisation N°1	
<b>Titre du cas</b>	S'authentifie
<b>But</b>	Vérifier l'identité des utilisateurs et les diriger vers leur espace respectif selon leurs rôles
<b>Acteurs</b>	Utilisateurs
<b>Pré condition</b>	Aucune
<b>Scenarion nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. Le système affiche le formulaire d'authentification</li> <li>3. Si les champs sont vide aller vers <b>Exception(1)</b> sinon <b>exécuter (4)</b></li> <li>4. Le système vérifie le login insérer. <ul style="list-style-type: none"> <li>• Si les données sont correctes <b>exécuter (5)</b></li> <li>• Sinon exécuter <b>Exception (2)</b></li> </ul> </li> <li>5. Afficher le tableau de bord correspondant au rôle de l'utilisateur.</li> </ol>
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. Le système indique qu'un champs vide a été détecté et réaffiche la boite d'authentification</li> </ol>

	2. Le système affiche un message d'erreur concernant le login introduit et renvoie l'interface d'authentification.
<b>Besoin IHM</b>	Un formulaire d'authentification avec le couple username et mot de passe.

Table II.5 Représentation de la description textuelle du cas « s'authentifier »

### II.7.1. Diagramme d'activité du cas d'utilisation «s'authentifier»

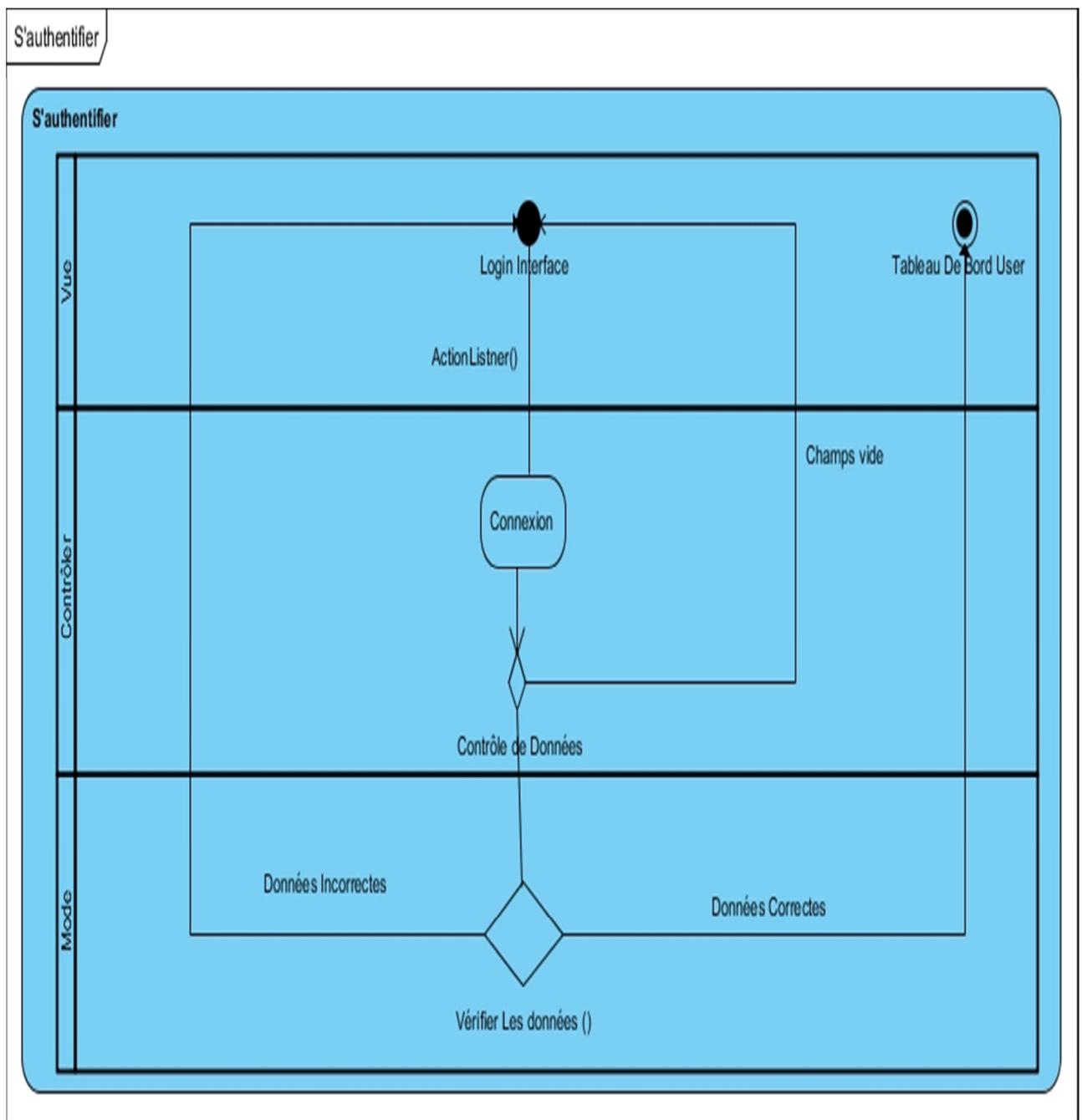


Figure II.5 Diagramme d'activité de l'authentification

## II.8.Cas d'utilisation concernant le Réceptionniste

A propos des cas d'utilisation relatif au réceptionniste nous allons décrire textuellement le cas « gérer les réservations de chambre » ce dernier est caractérisé par trois (3) opérations qui sont :

- Ajouter une réservation de chambre

- Modifier une réservation de chambre
- Supprimer une réservation de chambre

Et comme indiqué précédemment des consultations peuvent être faites sur les réservations enregistrer antérieurement.

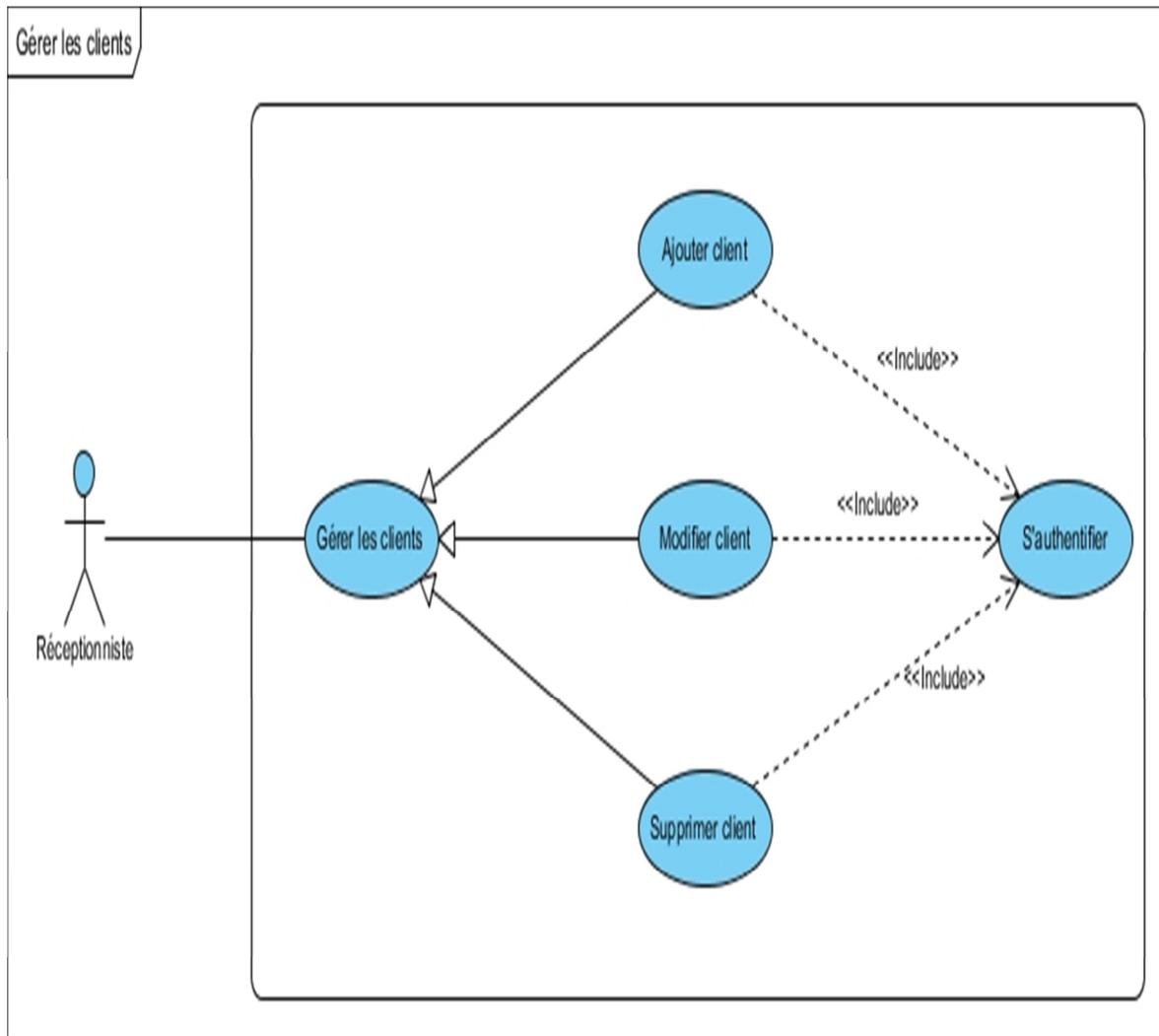


Figure II.6 Cas d'utilisation « Gérer les clients »

Cas d'utilisation N°2	
<b>Titre du cas</b>	Gérer les clients
<b>But</b>	Effectuer les opérations d'ajout, de modification et de suppression des clients
<b>Acteur</b>	Réceptionniste

<b>Pré condition</b>	Le réceptionniste doit s'authentifier
<b>Scenario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> </ol> <p>Pour l'ajout :</p> <ol style="list-style-type: none"> <li>4. Le réceptionniste accède à l'espace dédié</li> <li>5. Le système affiche le formulaire <ul style="list-style-type: none"> <li>• si les données sont incomplètes exécute <b>L'Exception (1)</b></li> <li>• Sinon aller (6)</li> </ul> </li> <li>6. Valider l'ajout <ul style="list-style-type: none"> <li>• Si les données sont erronées exécuter <b>l'Exception (2)</b></li> <li>• Sinon aller (7)</li> </ul> </li> <li>7. Réservation ajouté</li> </ol>
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. Un champ vide a été détecté réafficher le formulaire d'ajout</li> <li>2. Une donnée est erronée renvoyer le formulaire d'ajout</li> </ol>
<b>Besoin IHM</b>	Formulaire d'ajout

Table II.6 Représentation de la description textuelle du cas « Gérer les client »

### II.8.1. Diagramme d'activité du cas d'utilisation «gérer les clients»

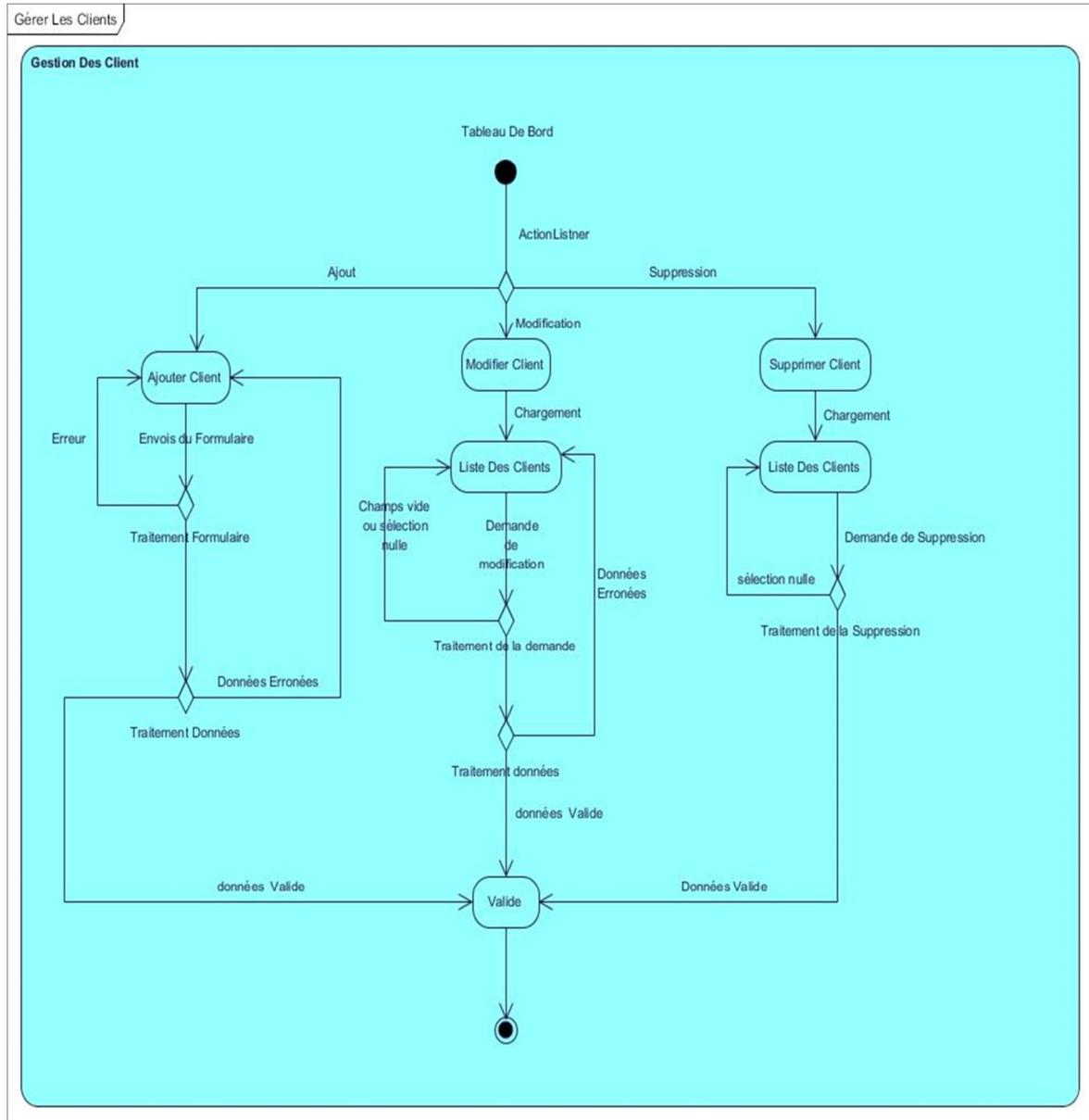


Figure II.7 Diagramme d'activité « Gérer les clients »

### II.9. Cas d'utilisations concernant le chef de réception

Concernant le chef de réception nous allons décrire le cas d'utilisation « gérer les factures » qui est composé de Quatre(4) scenarios relatif à la gestion des factures qui sont :

- L'ajout d'une facture
- La modification d'une facture
- La suppression d'une facture
- L'impression d'une facture.

Les consultations de factures déjà établis sont aussi disponibles pour ce cas d'utilisation.

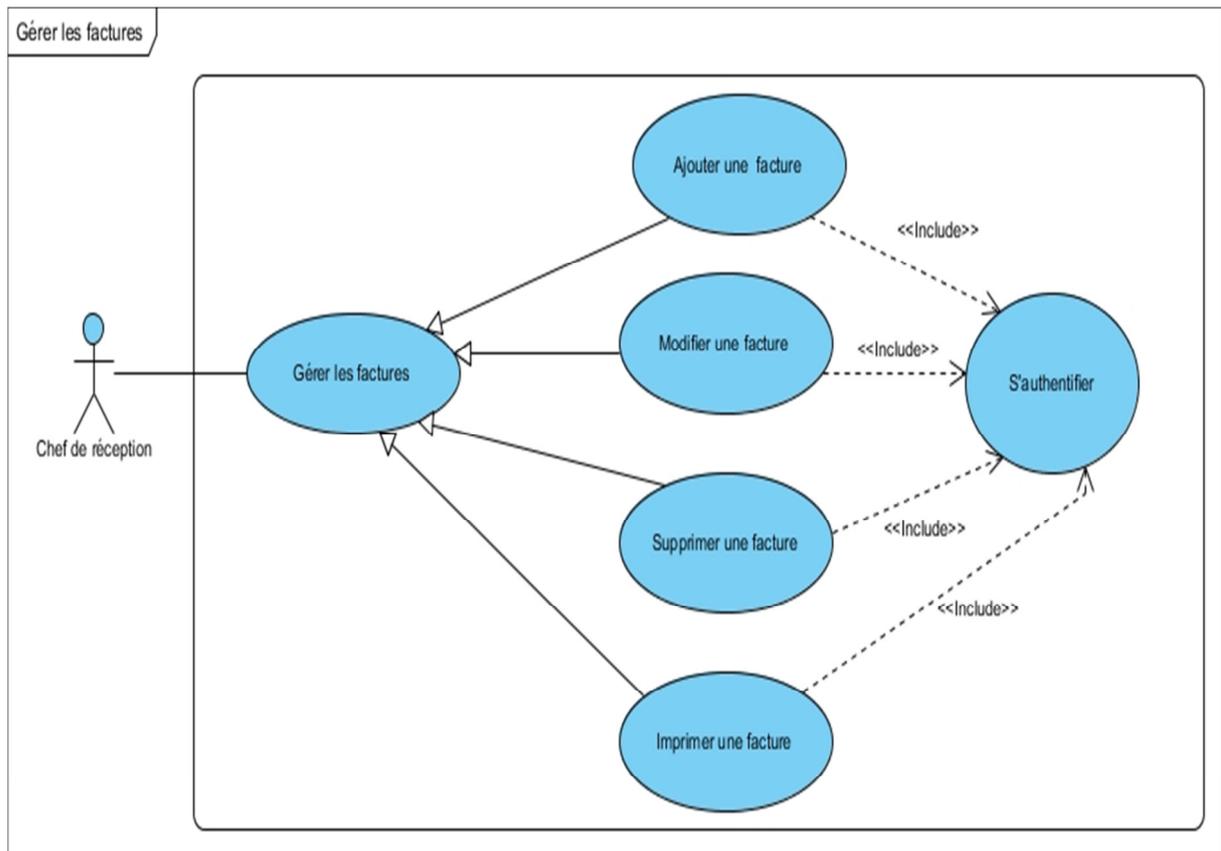


Figure II.8 Cas d'utilisation « Gérer les factures »

Cas d'utilisation N°4	
<b>Titre du cas</b>	Gérer les factures
<b>But</b>	Effectuer les opérations d'ajout, de modification, suppression et d'impression sur les factures
<b>Acteurs</b>	Chef de réception
<b>Pré condition</b>	S'authentifier

<b>Scenario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> </ol> <p>Pour l'impression :</p> <ol style="list-style-type: none"> <li>4. Le système renvoi le tableau des factures</li> <li>5. Le chef de réception effectue une recherche <ul style="list-style-type: none"> <li>• Si la facture est introuvable exécuter <b>l'exception (1)</b>.</li> <li>• Sinon <b>aller a (6)</b></li> </ul> </li> <li>6. Afficher la facture indiquée</li> <li>7. Valider l'opération d'impression</li> </ol>
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. La facture est introuvable et réafficher le tableau des factures</li> </ol>
<b>Besoin IHM</b>	Tableau des factures et une barre de recherche.

Table II.7 Représentation de la description textuelle du cas « Gérer les factures »

### II.9.1. Diagramme d'activité du cas « gérer les factures »

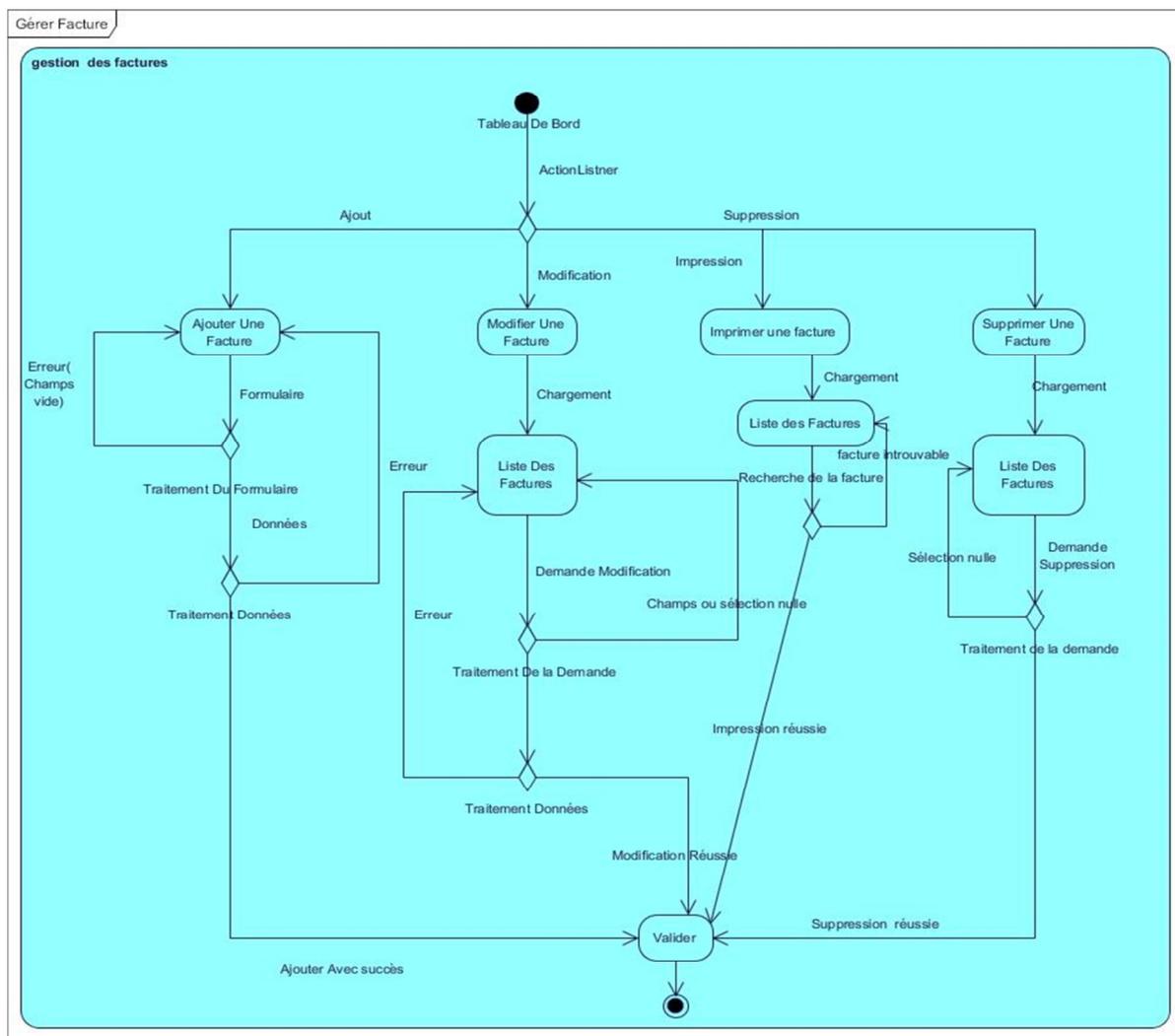


Figure II.9 Diagramme d'activité « Gérer les factures »

### II.10. Cas d'utilisation concernant le maître d'hôtel

Le cas « gérer les réservations de table » sera évoqué dans ce point, ce dernier est composé de trois (3) fonctionnalités répartie comme suit :

- L'ajout d'une réservation de table
- La modification d'une réservation de table
- La suppression d'une réservation de table

Ainsi que la possibilité d'effectuer des consultations sur les réservations ajouter antérieurement.

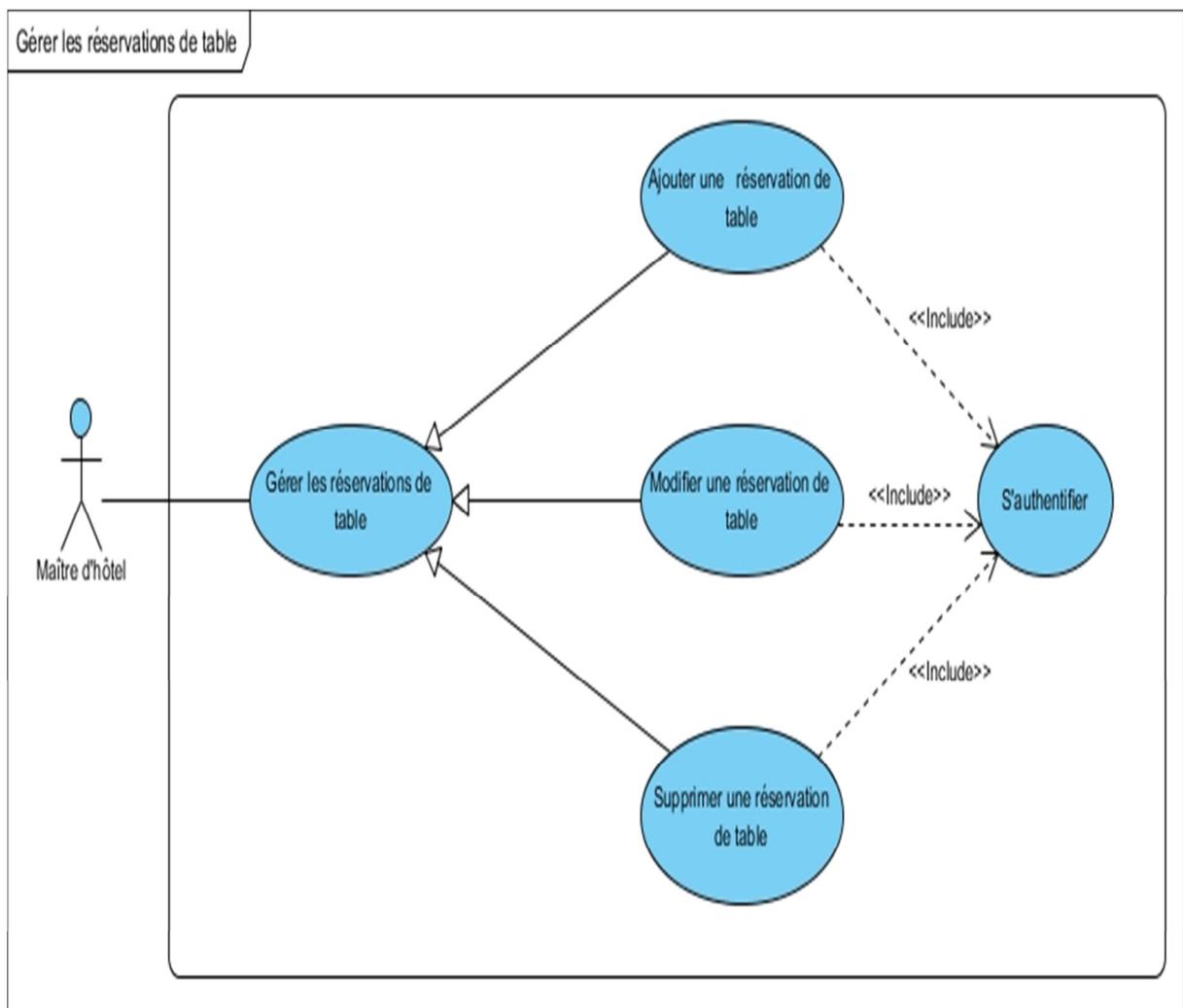


Figure II.10 Cas d'utilisation « Gérer les tables & salle »

<b>Numéro du cas d'utilisation</b>	N°7
<b>Titre du cas</b>	Gérer les réservations de table
<b>But</b>	Effectuer les opérations d'ajout, de modification et de suppression sur les réservations de table
<b>Acteurs</b>	Maitre d'hôtel
<b>Pré condition</b>	S'authentifier

<b>Scenario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> </ol> <p>Pour la Suppression :</p> <ol style="list-style-type: none"> <li>4. Le système renvoi le tableau des réservations de table existantes</li> <li>5. Le maitre d'hôtel effectue une recherche <ul style="list-style-type: none"> <li>• Si la réservation est introuvable exécuter <b>l'exception (1)</b></li> <li>• Sinon aller a <b>(6)</b></li> </ul> </li> <li>6. Afficher la table indiquée</li> <li>7. Valider l'action <ul style="list-style-type: none"> <li>• Si aucune table n'a été sélectionnée aller à <b>l'exception (2)</b></li> <li>• Sinon valider la suppression</li> </ul> </li> </ol>
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. La réservation est introuvable renvoyer la même interface</li> <li>2. Sélection nulle réafficher l'interface.</li> </ol>
<b>Besoin IHM</b>	Tableau des réservations, une barre de recherche.

Table II.8 Représentation de la description textuelle du cas « Gérer les réservations de table & salle »

### II.10.1. Diagramme d'activité du cas « gérer les tables & salle »

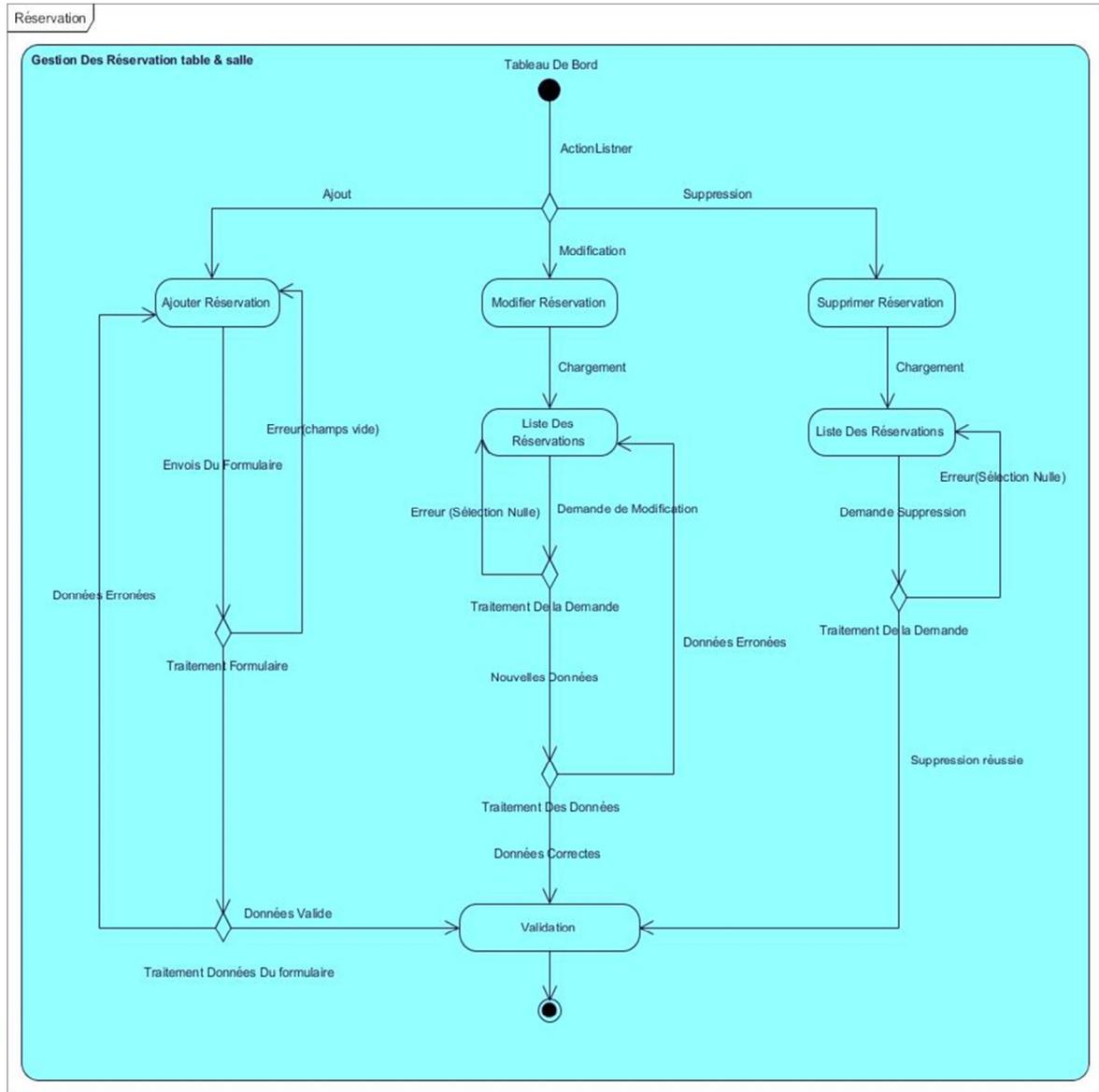


Figure II.11 Diagramme d'activité « Gérer les réservations de table & salle »

### II.11. Cas d'utilisation concernant le directeur

Pour le poste de directeur le cas d'utilisation « gérer les utilisateurs » sera décrit dans ce point, ce cas est caractérisé par deux (2) scénarios :

- Ajouter un utilisateur
- Supprimer un utilisateur

En plus de ces deux scenarios le directeur a le privilège de consulter les utilisateurs de l'application.

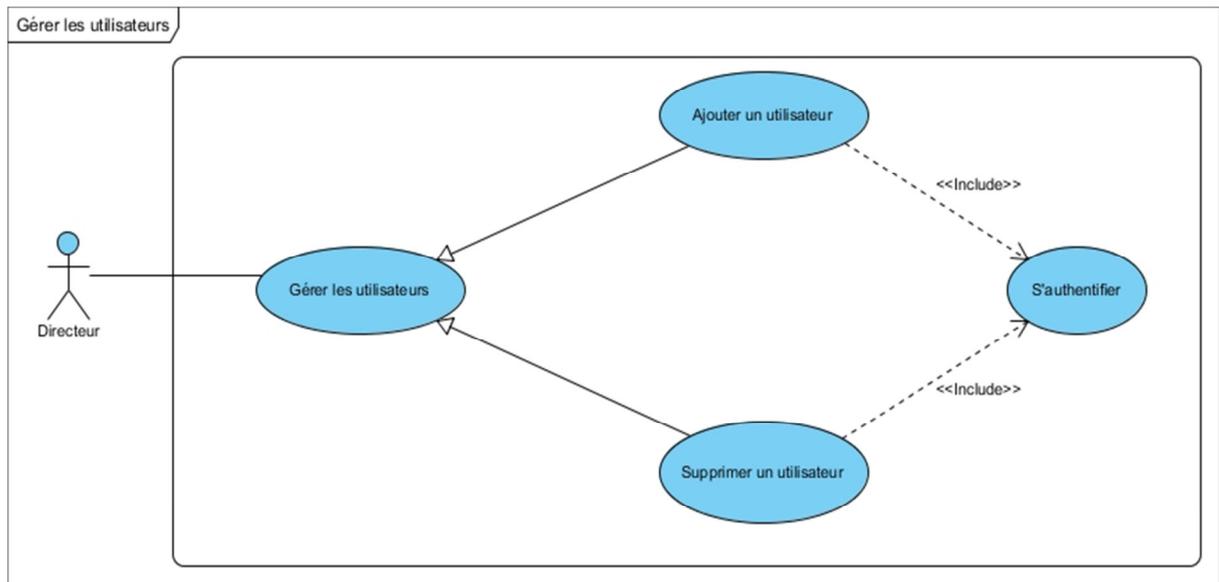


Figure II.12 Cas d'utilisation « Gérer les utilisateurs »

<b>Numéro du cas d'utilisation</b>	N°10
<b>Titre du cas</b>	Gérer les utilisateurs
<b>But</b>	Permet d'ajouter ou de supprimer un utilisateur de l'application
<b>Acteur</b>	Directeur
<b>Pré condition</b>	Le Directeur de l'hôtel doit s'authentifier
<b>Scenario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> </ol> <p>Pour l'ajout :</p> <ol style="list-style-type: none"> <li>4. Le directeur accède à l'espace d'ajout</li> <li>5. Le système affiche le formulaire                         <ul style="list-style-type: none"> <li>• si les données sont incomplètes exécute</li> </ul> </li> </ol>

	<p style="text-align: center;"><b>l'exception (1)</b></p> <ul style="list-style-type: none"> <li>• Sinon <b>aller (6)</b></li> </ul> <p>6. Valider l'ajout</p> <p>Pour la suppression :</p> <p>4. Le directeur accède à la section de suppression des utilisateurs</p> <p>5. Le système affiche le tableau des utilisateurs</p> <p>6. Il effectue une recherche de l'utilisateur à supprimer</p> <ul style="list-style-type: none"> <li>• Si l'utilisateur est introuvable aller à <b>l'exception (2)</b></li> <li>• Sinon exécuter <b>l'instruction (7)</b></li> </ul> <p>7. Valider la suppression</p> <ul style="list-style-type: none"> <li>• Si la sélection est vide aller à <b>l'exception (3)</b></li> <li>• Sinon exécuter la requête</li> </ul>
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. Un champ vide a été détecté, réafficher l'interface</li> <li>2. L'utilisateur recherché est introuvable, renvoyer l'interface de suppression</li> <li>3. Aucun utilisateur sélectionner, réafficher l'interface</li> </ol>
<b>Besoin IHM</b>	<p>Pour l'ajout :</p> <ul style="list-style-type: none"> <li>➤ Formulaire d'ajout</li> </ul> <p>Pour la suppression :</p> <ul style="list-style-type: none"> <li>➤ Tableau des utilisateurs et barre de recherche</li> </ul>

Table II.9 Représentation de la description textuelle du cas « Gérer les utilisateurs »

### II.11.1. Diagramme d'activité du cas « gérer les utilisateurs »

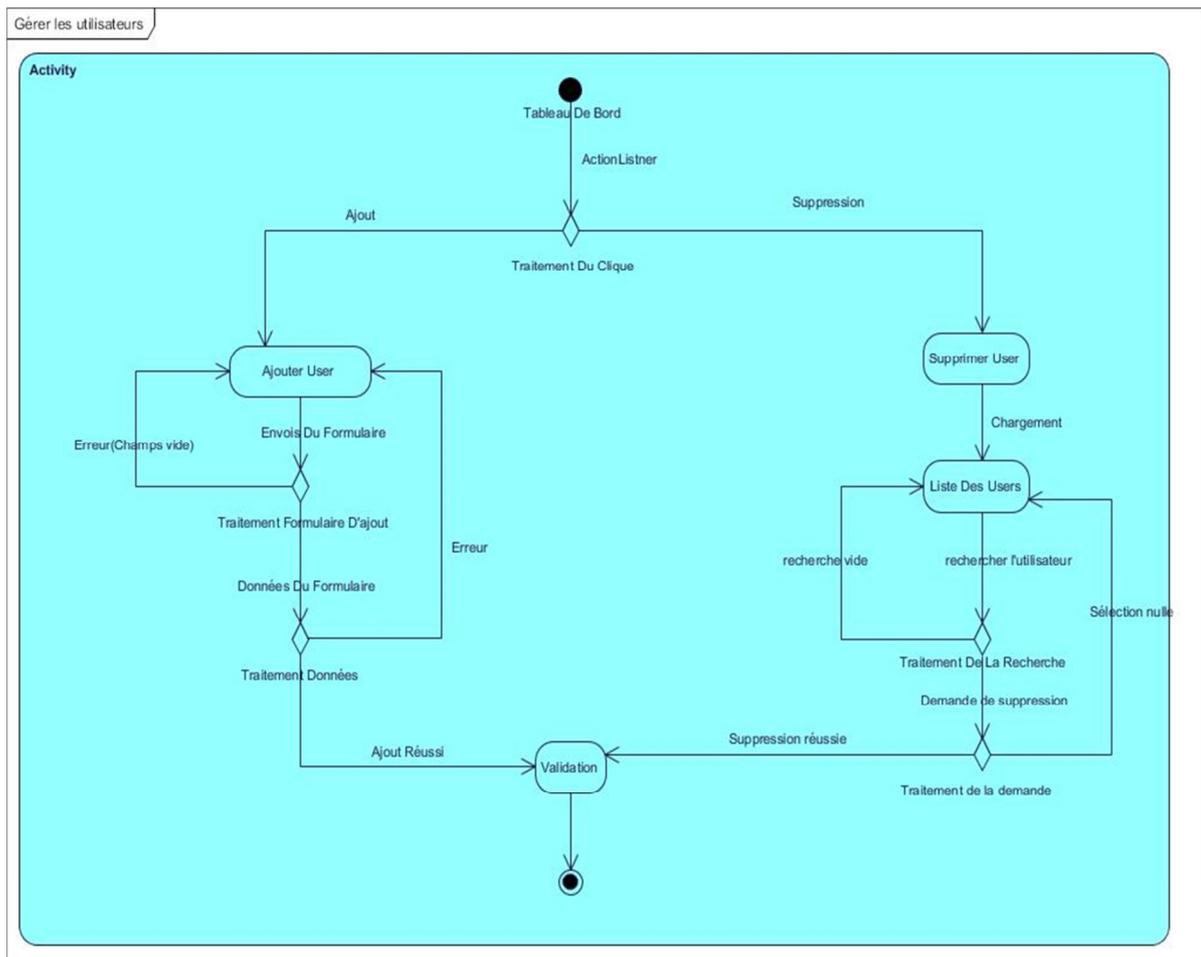


Figure II.13 Diagramme d'activité « Gérer les utilisateurs »

## II.12. Modélisation dynamique

Le modèle dynamique permet d'examiner le comportement des objets, et les modifications d'états des objets suite aux réceptions de messages. En phase d'analyse, les messages échangés entre objets sont vus comme des événements, UML modélise la dynamique sous la forme de quatre types de diagrammes :

- Diagramme de cas d'utilisation
- Diagrammes de séquences (pour chaque scénario)
- Diagrammes de collaboration
- Diagrammes d'états (pour chaque classe d'objet actif [6]).

## II.13. Diagramme de séquence

Pour mieux illustrer les interactions entre le système et les acteurs et pour ajouter des informations complémentaires sur les cas d'utilisations nous nous référons à un diagramme très important dans la phase d'analyse des besoins, il s'agit du **diagramme de séquence système** qui va clarifier les scénarios cités dans les descriptions textuelles et représenter la succession chronologique des opérations réalisées par les acteurs, les participants aux interactions (le système et les acteurs) seront symbolisés par une ligne de vie et une instance, les messages échangés entre ces deux entités seront représentés par des flèches du haut vers le bas le long de la ligne de vie et respecteront un ordre chronologique ainsi que les fragments d'interactions comme qui sont expliqués dans [7].

### II.13.1. Diagramme de séquence du cas d'utilisation «s'authentifier»

Le processus d'authentification permet au système de s'assurer de la légitimité de la demande d'accès à l'application émanant d'un utilisateur, cette opération est souvent composée de la vérification d'un couple d'information qui sont l'identifiant (nom d'utilisateur dans notre cas) et mot de passe avant d'accorder les privilèges liés à ce compte.

Dans cette étape deux situations peuvent se présenter des données complètes ou incomplètes, ce qui rend plus crédible l'utilisation du premier **Alt (alternative)**, dans la situation où les données sont complètes deux scénarios sont envisageables données correctes ou incorrectes c'est là l'utilité du deuxième **Alt** si ces dernières sont correctes alors le système accorde l'accès à l'interface adéquate au rôle de l'utilisateur sinon un message d'erreur est affiché et la même interface sera renvoyé.

Cette combine est répétée à chaque tentative d'accès au système ce qu'explique l'utilisation de l'opération « **Loop** ».

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

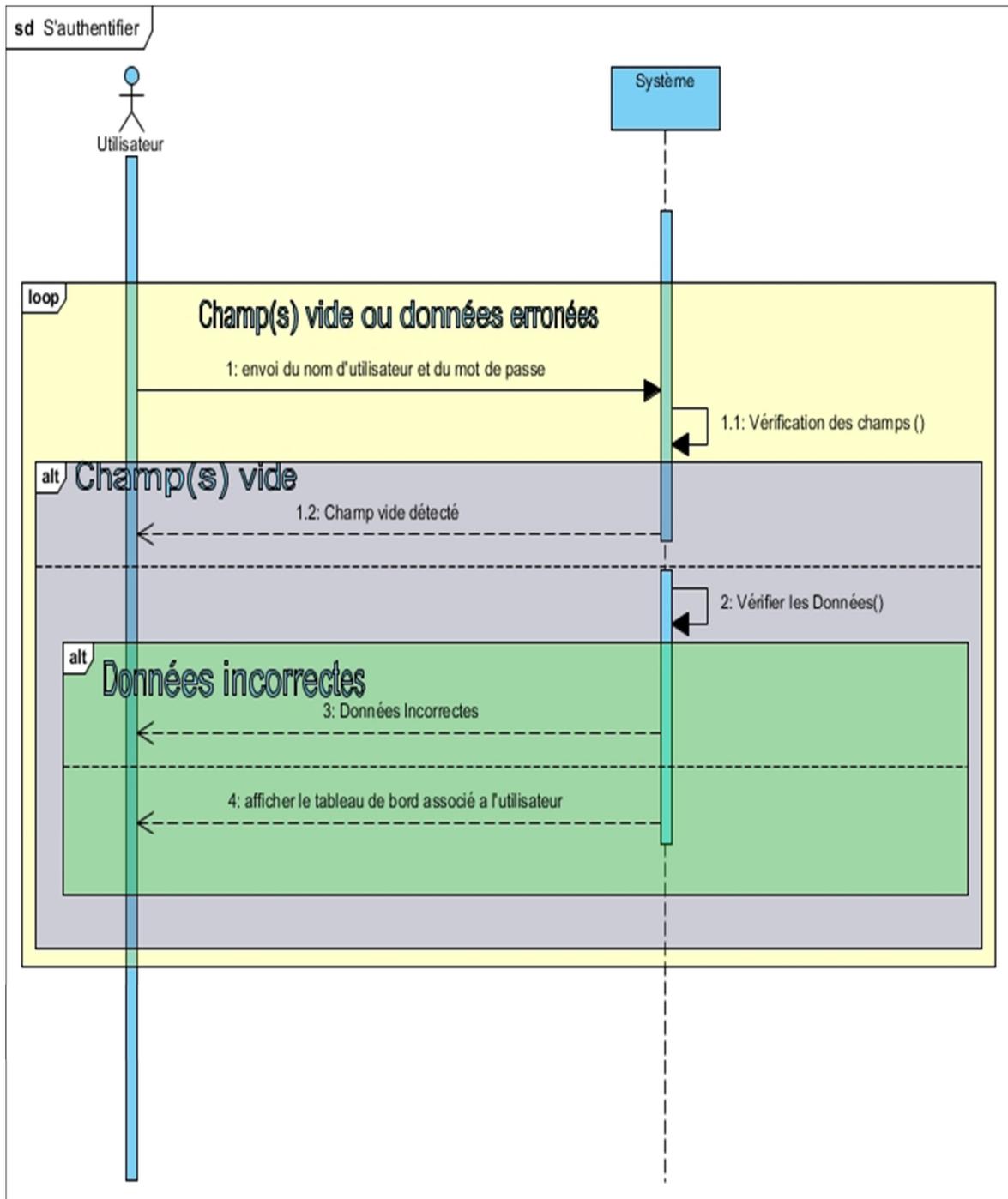


Figure II.14 Diagramme de Séquence « S'authentifier »

### II.13.2. Diagramme de séquence du cas d'utilisation « Gérer les Réservations (table, chambre, salle des fêtes) »

Après avoir passé le contrôle d'accès le système affiche l'espace approprié à l'utilisateur, cet espace comporte les fonctionnalités les plus importantes de notre application il s'agit de la gestion des multiples réservations, chaque réservation comporte trois scénarios qui reflète un choix de l'utilisateur c'est là l'utilité du fragment « **Opt** », indiquant que ces scénarios arrivent dans n'importe quel ordre.

#### a) Ajouter une réservation

Après l'envoi du formulaire d'ajout deux situations peuvent se présenter des informations incomplètes ou complètes ce qui explique la présence du premier « **Alt** » qui intercepte les erreurs dues aux champs vides, dans le cas où les informations sont complètes, ces données peuvent être correctes ou erronées ce qui éclairci le deuxième « **Alt** » qui pourra arrêter au passage les données incorrectes est affiché un message d'erreur.

Pour que ces vérifications se répètent un fragment « **Loop** » est combiné avec les fragments cite au-dessus.

#### b) Modifier une réservation

L'utilisateur effectue une recherche deux situations se présentent soit elle aboutit à un résultat concluant ou à un ensemble vide ce qui explique la présence du premier « **Alt** », qui capte les erreurs de recherche, dans le cas où le résultat de la recherche n'est pas vide le formulaire de modification est envoyé, une vérification des champs est effectuée c'est là que réside le rôle du deuxième « **Alt** » si un champ incomplet est trouvé une erreur est affichée sinon on passe au troisième « **Alt** » ou un contrôle de validité des données est effectué si les

information sont incorrectes une erreur est signalée sinon la modification est validé. Pour que ces vérifications se répètent un fragment « **Loop** » est combiné avec les fragments cite au-dessus.

### c) Supprimer une réservation

Dans le troisième scenario et pour pouvoir supprimer une réservation l'utilisateur est obligé d'effectuer une recherche ce mène à deux situation une réservation introuvable dans ce cas le système affiche un message d'erreur ou un ensemble de résultat c'est ce qui explique la présence du premier « **Alt** », si les résultats de la recherche sont concluants une suppression est effectuée dans une vérification de la sélection est lancée, si cette dernière est vide un message d'erreur est affiché par le système sinon la suppression est validée et c'est là l'importance du deuxième « **Alt** ». Pour que ces vérifications se répètent un fragment « **Loop** » est combiné avec les fragments cite au-dessus.

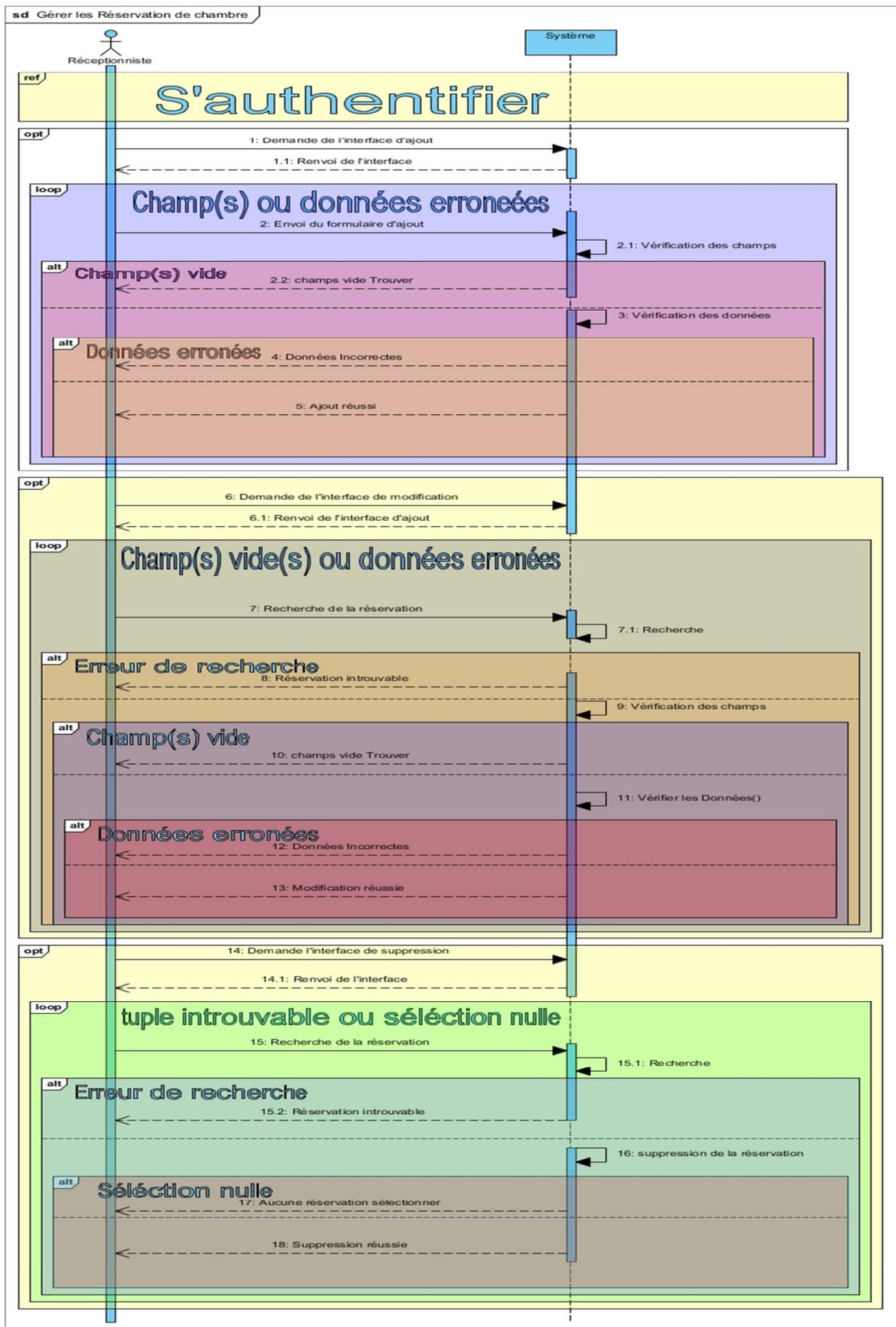


Figure II.15 Diagramme de Séquence « Gérer les Réservation de chambre »

### II.13.3. Diagramme de séquence du cas d'utilisation «Gérer les utilisateurs»

Après l'authentification, le directeur effectue une demande de gestion des utilisateurs. Deux 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.
- Supprimer un utilisateur : Le directeur effectue une recherche en saisissant un mot clé. Après l'affichage de résultat, l'administrateur sélectionne l'utilisateur concerné et valide la suppression.

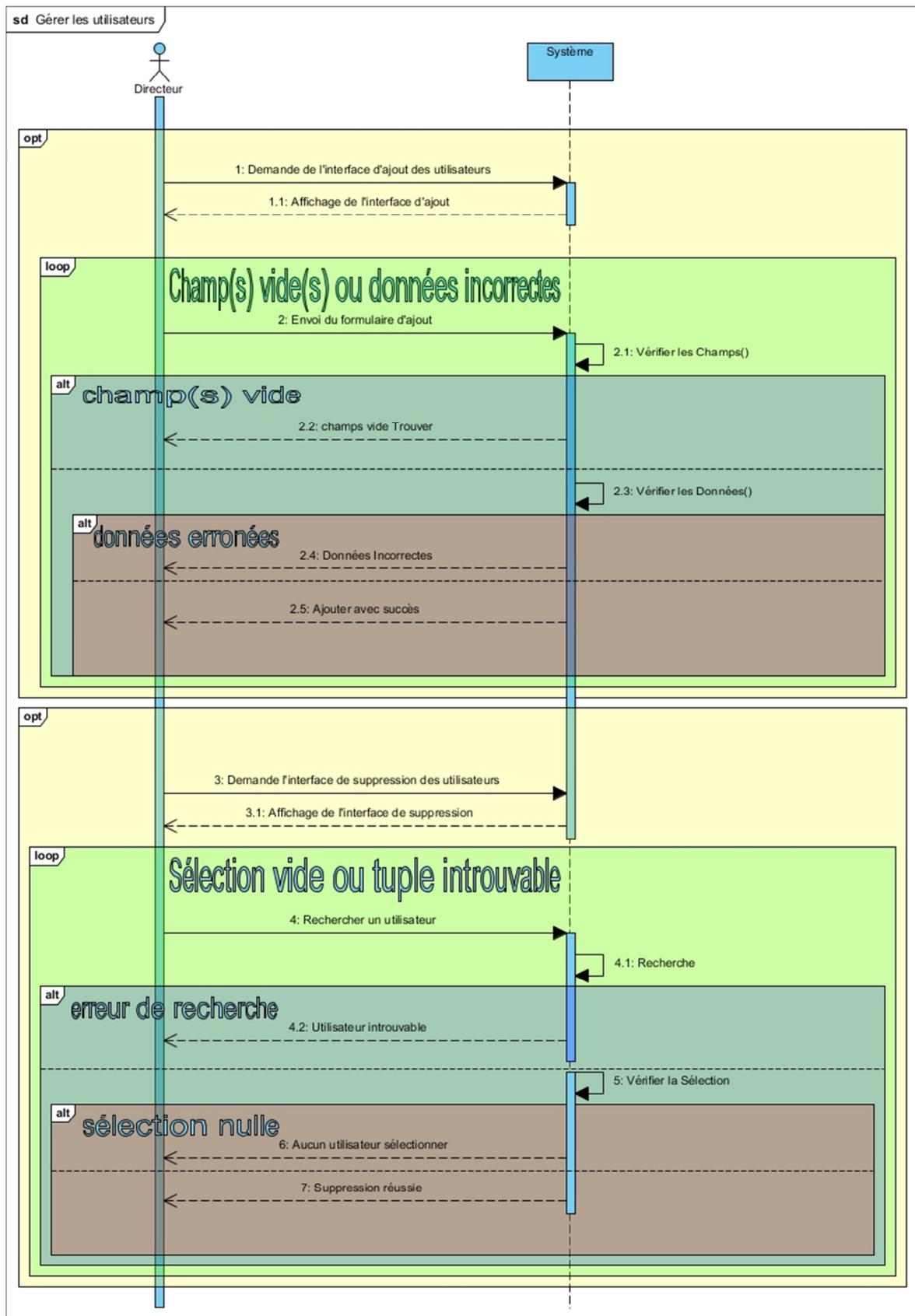


Figure II.16 Diagramme de Séquence « Gérer les utilisateurs »

## Conclusion

Dans cette partie de notre étude consacrée à l'analyse des besoins nous avons mis le point sur les contours du système en identifiant les acteurs ainsi que les messages échangés entre eux que nous avons représenté dans un diagramme de contexte , nous avons aussi dédié une partie de ce chapitre à la modélisation dynamique de notre application, où nous avons présenté les différents diagrammes de l'UML et énuméré les cas d'utilisation, leurs scénarios ainsi que leurs interaction avec le système. Dans le chapitre suivant intitulé **Conception et Elaboration du Schéma Relationnel**. Nous allons mettre l'accent sur l'architecteur globale du système au niveau physique et logique ainsi que l'élaboration du diagramme de classe du domaine et le passage au modèle relationnelle de donnée.

# CHAPITRE III

## CONCEPTION ET ELABORATION DU SCHEMA RELATIONNEL

# CHAPITRE III

## CONCEPTION ET ELABORATION DU SCHEMA RELATIONNEL

---

### Introduction

Ce chapitre sera consacré à la conception objet de notre système ainsi que son architecture globale, est ce dans le but de définir les objets métiers qui réalisent les cas d'utilisations recensés dans le chapitre précédent, pour atteindre ce résultat nous allons commencer par présenter les diagrammes d'interaction associés à notre application qui décriront les interactions entre acteur(s) identifier antérieurement et les objets de notre système, par la suite nous concevrons le diagramme de classe ,enfin nous illustrons le processus à suivre pour passer du modèle de classe au modelé relationnel .

### III.1.Diagramme de séquence d'interaction

Les diagrammes de séquence peuvent être aussi utilisés pour documenter un cas d'utilisation. Les interactions entre objets représentent, dans ce cas, des flux d'informations échangés et non pas de véritables messages entre les opérations des objets.[7].

Souvent à l'intérieur d'un système il existe des classes qui possèdent un rôle particulier il serait très intéressant de visualiser le comportement de ces instances,

à cet effet les stéréotypes de **Jacobson** ont fait leurs appréciation pour apporter les éclaircissements a la notion système représentée dans le diagramme de séquence comme une boite noire. Dans un diagramme d'interaction il existe trois types de classes qui sont :

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



Figure III.1 Représentation d'IHM

- **Classes de Contrôle (Control)** : Ce sont des classes qui contiennent les traitements et la cinématique de l'application. Elles font la transition entre les classes d'interface et les classes entité. Elles sont schématisées comme suit :

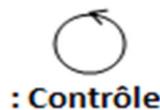


Figure III.2 Représentation des contrôleurs

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



Figure III.3 Représentation des entités

Dans cette partie de notre travail nous allons réaliser un diagramme de séquence d'interaction pour chaque diagramme de séquence du deuxième chapitre en remplaçant le système par les objets qu'interviennent dans les cas d'utilisations.

### III.3. Diagramme d'interaction pour le cas d'utilisation «s'authentifier»

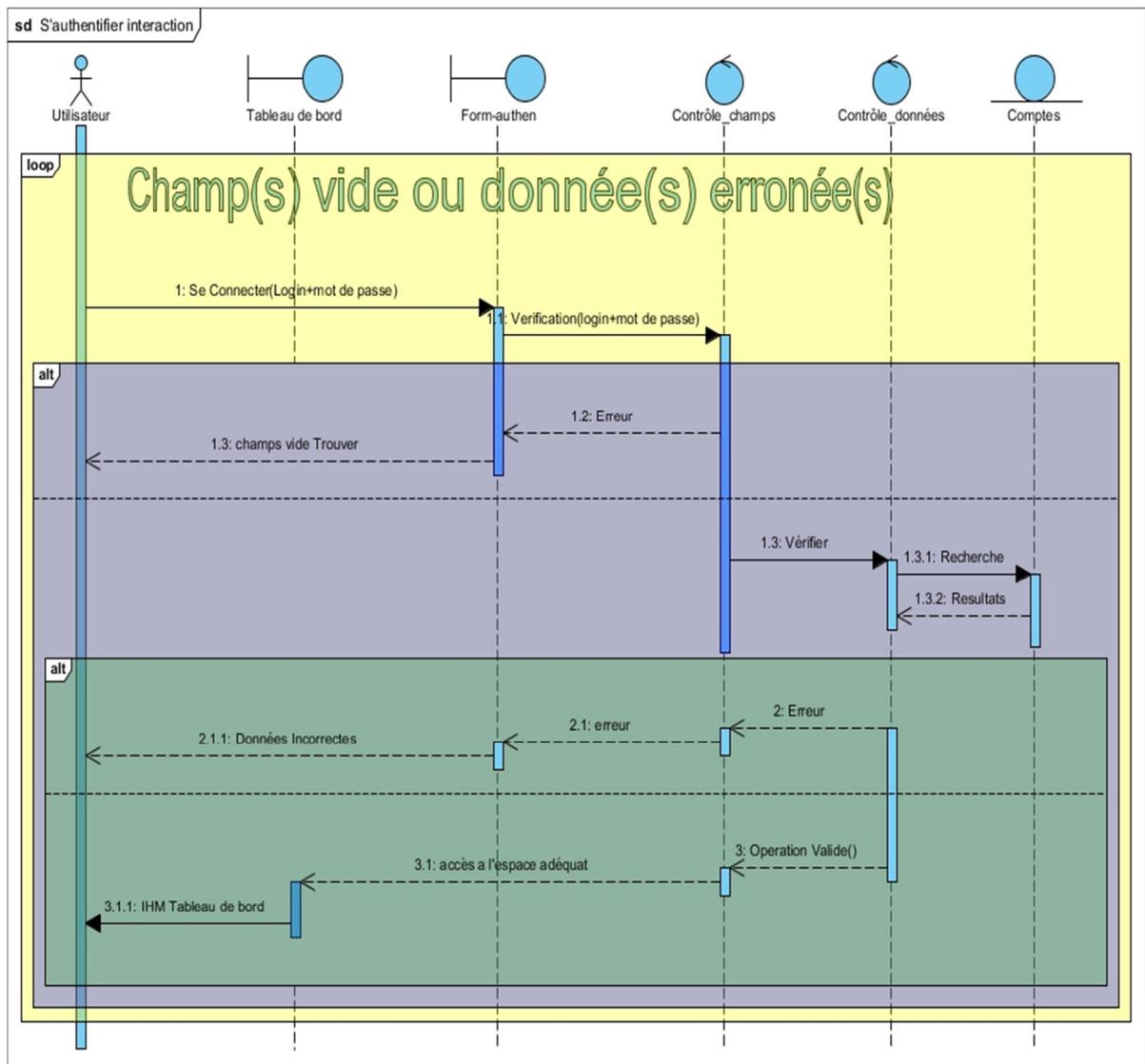
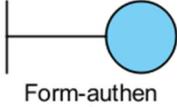
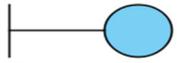


Figure III.4 Diagramme de séquence détaillé « s'authentifier »

Le tableau suivant résume les stéréotypes utilisés pour représenter les objets qui interviennent pour réaliser le cas d'utilisation « S'authentifier » :

<i>Stéréotype</i>	<i>Signification</i>
 <p>Form-authen</p>	<p>Objet interface pour le formulaire d'authentification. Ce dernier contient deux champs de saisie, à savoir, l'identifiant (login) et le mot de passe, qui permettent à l'utilisateur de renseigner son identifiant (nom d'utilisateur) et son mot de passe. En plus, il contient un bouton pour demander la connexion au système.</p>
 <p>Contrôle_champs</p>	<p>Objet contrôle qui permet de vérifier s'il y a un ou plusieurs champs vides.</p>
 <p>Contrôle_données</p>	<p>Deuxième objet contrôle qui permet de vérifier si les champs renseignés par l'utilisateur sont correctes. Cette vérification s'effectue en utilisant l'objet entité de la classe « Compte » (objet persistant qui enregistre les comptes des utilisateurs du système).</p>
 <p>Comptes</p>	<p>Objet entité (objet métier) qui contient tous les comptes des utilisateurs du système.</p>
 <p>Tableau de bord</p>	<p>Objet interface qui représente la vue principale de l'utilisateur connecté selon son rôle. Nous pouvons le considérer comme l'espace de travail qui contient tous les cas d'utilisation, ils seront accessibles à partir de cette interface.</p>

**Table III.1 Représentation des éléments composant le diagramme de séquence « s'authentifier »**

### III.4. Diagramme d'interaction pour le cas d'utilisation

#### « gérer les utilisateurs »

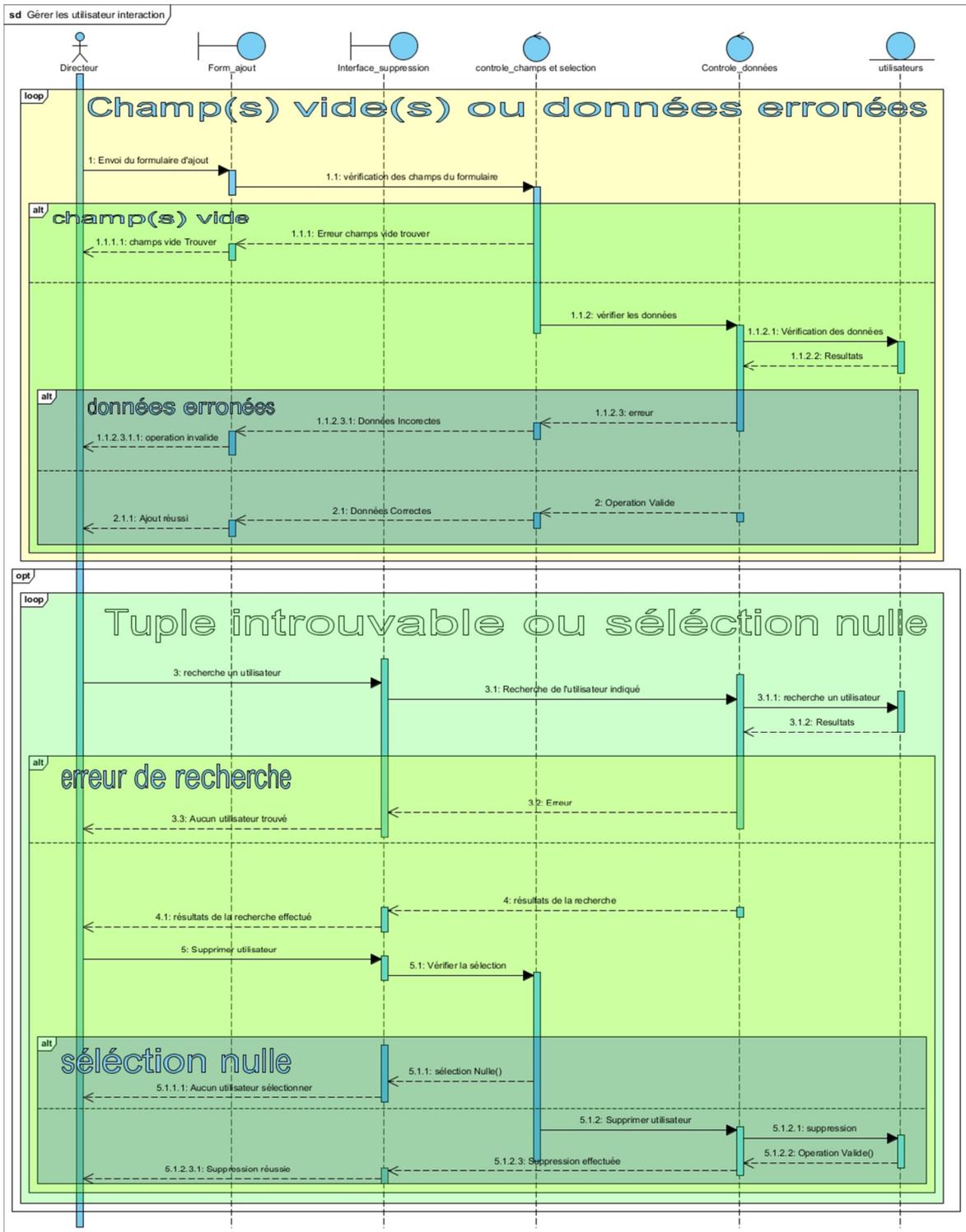


Figure III.5 Diagramme de séquence détaillé « Gérer les uses »

## III.5.Diagramme de classes de domaine

### III.5.1.Présentation du Diagramme de classes

Le diagramme de classes est l'un des plus importants de la modélisation UML, il constitue le pivot du développement orienté objet, selon [8] il est le seul diagramme obligatoire dans de tel développement, d'après [7] l'objectif de ce diagramme est de permettre d'avoir une représentation statique du système à développer ce qui veut dire que le facteur temporel n'est pris en compte dans le comportement de ce dernier, Cette représentation est centrée sur les concepts de classe du domaine d'application et le concepts 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 en se basant sur les liens qui les unis (héritage, agrégation, composition...etc.), l'avantage qu'offre ce diagramme est qu'il est facilement transformable en code [9].

### III.5.2.Concepts du diagramme de classe

- **La notion d'objet** Un objet est une entité aux frontières bien définies, possédant une identité et encapsulant un état et un comportement. Un objet est une instance (ou occurrence) d'une classe.
- **Classe :** Une classe est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.[8].
- **Attribut :** Un attribut est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur (sauf cas d'attributs multivalués). [7].
- **Opération :** Une opération représente un service, un traitement qui peut être demandé à n'importe quel objet de la classe. elle est

l'abstraction de ce que peut être réalisé sur un objet, et elle est partagée par tous les objets de la classe. [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. [10].

Il existe plusieurs types de relation UML dans le diagramme de classes nous distinguons :

- **Association** : Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées. [8].
- **Agrégation** : Une agrégation est une association qui représente une relation d'inclusion structurelle ou comportementale d'un élément dans un ensemble. [8].
- **Composition** : La composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre la classe « composant » et la ou les classes « composé ». Autrement dit la suppression de la classe « composé » implique la suppression de la ou des classes « composant ». [9].
- **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 [11].

### III.5.3. Diagramme de classe

Nous avons divisé notre diagramme en deux(2) parties, une partie pour illustrer la relation d'héritage qui lie les utilisateurs, et une pour la relation entre les classes qui compose le système

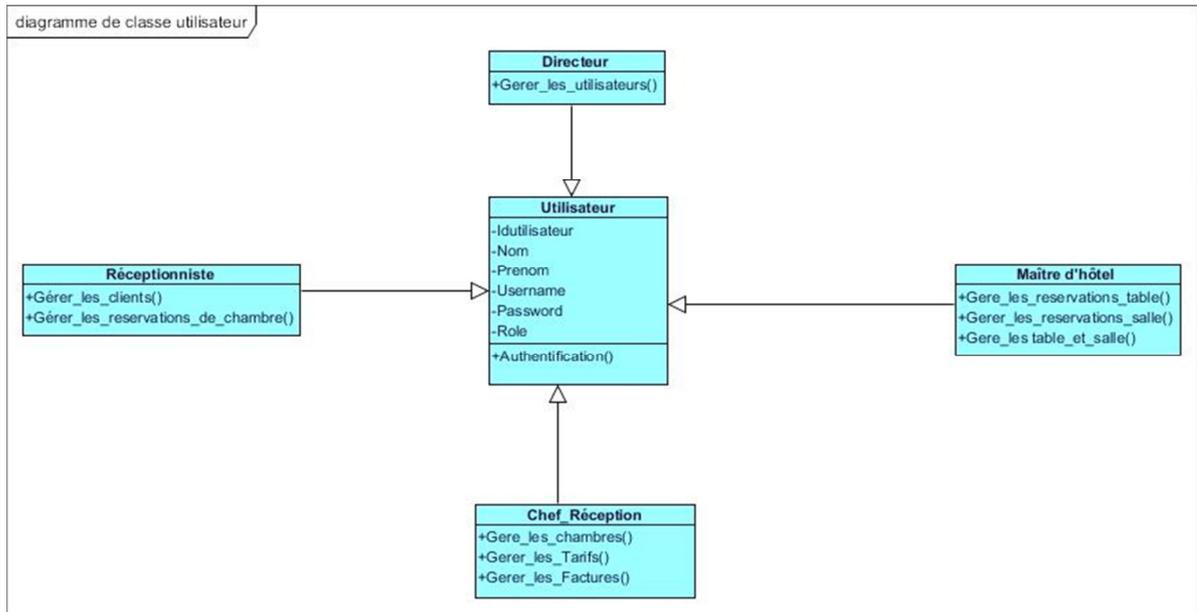


Figure III.6 Diagramme de classes des utilisateurs

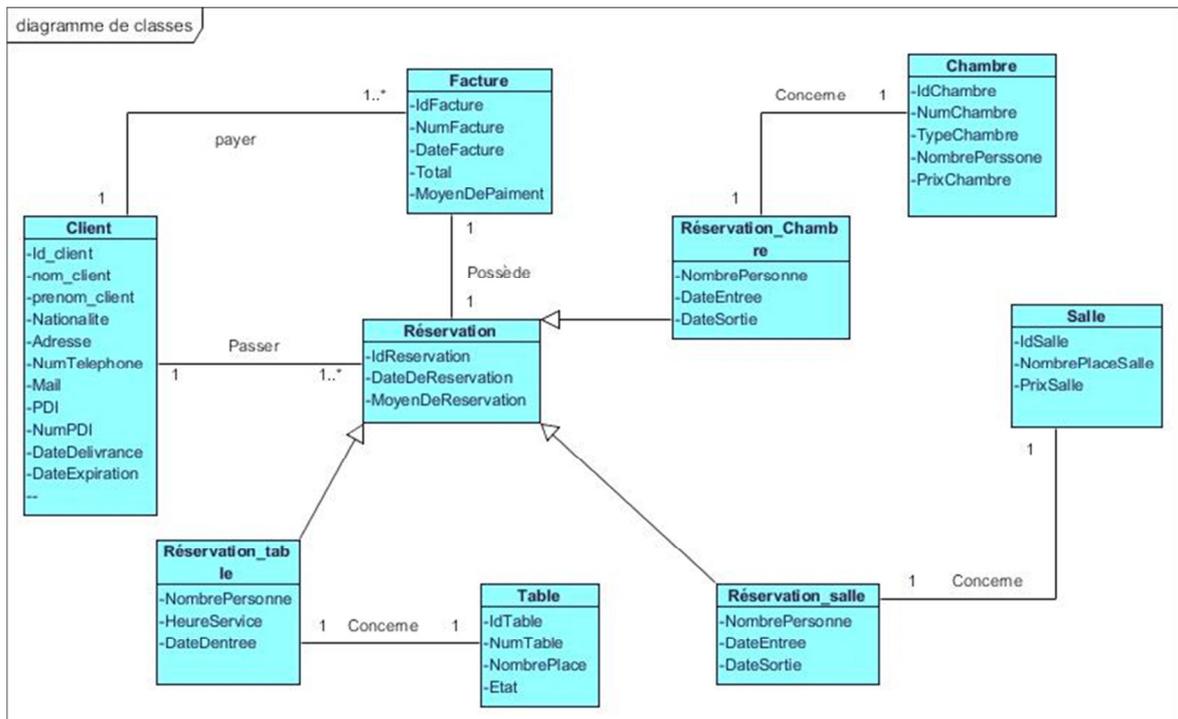


Figure III.7 Diagramme de classes du domaine

Classe	Responsabilité	Attributs		
		Désignation	Définition	Type (taille)
Client	Représente les clients de l'établissement aient émet une réservation	<u>Id</u>	Identifiant	Entier(8)
		Nom_client	Nom du client	Varchar(25)
		Prenom_client	Prénom du client	Varchar(25)
		Nationalite	Nationalité du client	Varchar(25)
		Adresse	Adresse du client	Varchar(35)
		NumTelephone	N° téléphone du client	Entier(10)
		Mail	Adresse email du client	Varchar(25)
		PDI	Type de pièce d'identité	Enuméré (CDI, PC, PSS)
		NumPDI	N° pièce d'identité	Entier (entier)
		DateDelivrance	Date de délivrance de la pièce d'identité	Date(jj/mm/aaaa)
		DateExpiration	Date d'expiration de la pièce d'identité	Date(jj/mm/aaaa)

Réservation	Représente les réservations émises par les clients	<b><u>IdReservation</u></b>	Identifiant	Entier (8)
		DateReservation	Date de la réservation	Date(jj/mm/aaaa)
		MoyenDeReservat ion	Le moyen utilisé pour la réservation	Énuméré( téléphone, email ,physique )
Chambre	Représente Les chambre de l'hôtel	<b><u>IdChambre</u></b>	Identifiant	Numérique.
		NumChambre	Numéro de la chambre	Entier(3)
		TypeChambre	Type de la chambre	Varchar(25)
		Nombre de personne	Nombre de lit dans la chambre	Entier (1)
		PrixChambre	Le coût de la chambre	Réel(10).
Table	Représente les tables du service restauration	<b><u>IdTable</u></b>	Identifiant.	Entier(8)
		NumTable	N° de la table	Entier(8)
		NombrePlace	Nombre de chaise par table	Entier(2)
		Etat	Etat actuel de la table	Enuméré (disponible, hors service)
Salle	Représente la salle des fêtes de l'hôtel	<b><u>IdSalle</u></b>	Identifiant.	Entier(8)
		NombrePlaceSalle	Nombre de place dont dispose la salle	Entier(4)
		PrixSalle	Coût de la salle	Réel(10)

Facture	Représente les factures des réservations destinées aux clients et à la comptabilité de l'hôtel	<b><u>IdFacture</u></b>	Identifiant	Entier(8)
		NumFacture	N° de la facture	Entier(10)
		DateFacture	Date de l'impression de la facture	Date (jj/mm/aaaa)
		TypePaiement	Le moyen de paiement	Enuméré (espèce, cheque, carte)
		Total	Total des consommations	Réel (10)
Utilisateur	Représente tous les acteurs de notre système	<b><u>IdUtilisateur</u></b>	Identifiant	Entier(8)
		Nom	Nom de l'utilisateur	Varchar(25)
		Prenom	Prénom de l'utilisateur	Varchar(25)
		Username	Nom d'utilisateur ou identifiant	Varchar(25)
		Password	Mot de passe de l'utilisateur	Varchar(25)
		Rôle	Le poste occupé par l'utilisateur	Enuméré (Réceptionniste, Chef de Réception, Maitre d'hôtel)

Table III.2 Représente les propriétés et méthodes de chaque classe.

### III.5.5.Schéma Relationnel

En partant du diagramme de classe nous pouvons déduire le schéma relationnel (model logique de données) en appliquant les règles de passage suivantes :

- **Règle 01** : Chaque classe du diagramme de classe devient une relation
  - Chaque attribut de classe se transforme en un champ de table
  - Choisir un attribut de la classe pouvant jouer le rôle de clé primaire

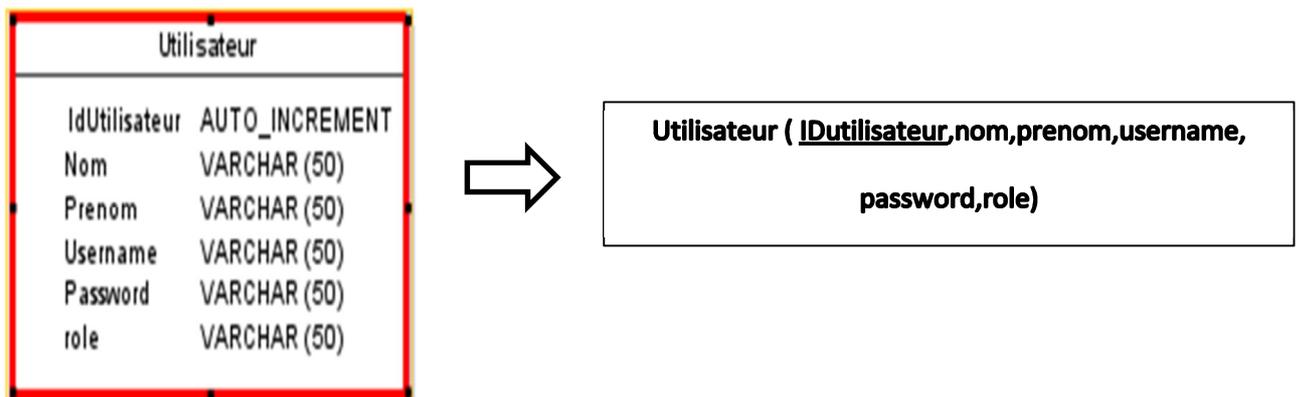
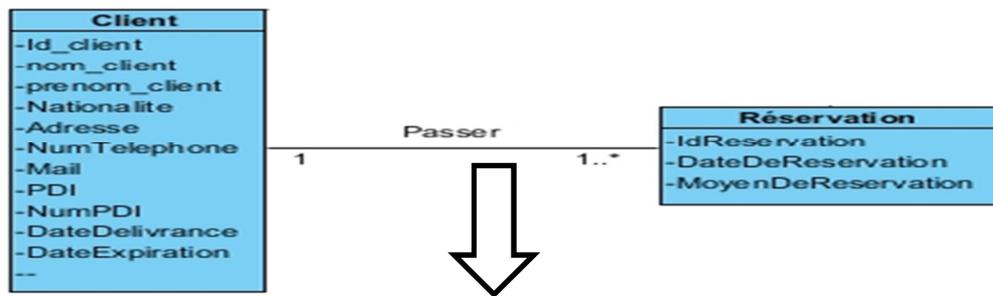


Figure III.8 Représentation de la règle de passage N°1

- **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.



**Réservation ( IdReservation, DateDeReservation, MoyenDeReservation, #Id\_client)**

Figure III.9 Représentation de la règle de passage N°2

- **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 surclasse, 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 surclasse. 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 surclasse.

Après l'application des règles ci-dessus citées, nous obtiendrons le schéma relationnel suivant :

**Utilisateur**(IdUtilisateur, nom, prenom, username ,password, role)

**Client**(IdClient,NomClient,PrenomClient,Nationalite,Adresse,NumTelephone,  
Mail,PDI,NumPDI,DateDelivarance,DateExpiration)

**Réservation\_\_Table**(IdReservationTable,DateDeReservation,MoyenDeReserv  
ation,NombrePersonne,DateDentree,HeureService,#IdTable,#IdClient,#IdFactur)

**Réservation\_\_Chambre**(IdReservationChambre,DateDeReservation,  
MoyenDeReservation,NombrePersonne,DateDentre,DateSortie,PrixChambre,#IdC  
hambre,#IdClient, #IdFacture)

**Réservation\_\_Salle**(IdReservationSalle,DateDeReservation,MoyenDeReservation,  
NombrePersonne,DateEntre,DateSortie,PrixSalle,#IdSalle,#IdClient,#IdFacture)

**Facture**(IdFacture,NumFacture,Date,Total,MoyenDePaiement#IdClient,  
#IdReseravtionChambre,#IdReservationTable,IdReservationSalle)

**Table**(IdTable, NumTable, NombrePlace, Etat)

**Salle**(IdSalle, NombrePlaceSalle, PrixSalle)

**Chambre**(IdChambre, NumChambre ,TypeChambre ,NombrePersonne,  
PrixChambre)

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

## Conclusion

A travers ce chapitre consacré à la conception de notre application selon l'approche orientée objet, nous avons pu avoir une réponse concrète à la question de comment le système accomplira les cas d'utilisations identifier précédemment ?

Pour cela, nous avons établis les diagrammes de séquence d'interaction qui nous ont permis de visualiser de près le comportement des classes interface, contrôle, Entité de notre système.

Puis une courte présentation du diagramme de classes et de ses concepts fondamentaux ont été exposé dans cette partie ce qui nous a profité à modéliser le diagramme de classes que nous avons au final exploité pour élaborer le schéma relationnel de notre application en appliquant les différentes règles de passage cité dans ce chapitre .

Cette partie de notre travail est un fondement pour la phase de réalisation qui sera l'objet du chapitre suivant.

# CHAPITRE IV

## REALISATION

---

# CHAPITRE IV

## REALISATION ET TESTS

---

### Introduction

Le quatrième chapitre sera consacré à la partie réalisation qui est une phase cruciale car elle représente le volet pratique de notre application.

Dans cette ultime partie nous allons commencer par exposer les outils et l'environnement de développement ainsi que les techniques mises en œuvre pour atteindre les objectifs fixés dans les chapitres précédents. Par la suite nous allons montrer un aperçu des interfaces les plus importantes de notre application.

### IV.1. Environnement de Programmation et Bibliothèques

Cette section va nous permettre d'énumérer les différentes technologies utilisées pour la mise en œuvre de notre système :

#### IV.1.1. WampServer

C'est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données. [12].

### IV.1.2.Eclipse

C'est un environnement de développement intégré **IDE** (Integrated Development Environment) libre dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques sous différents langages de programmation (Java, C++, PHP...).

### IV.1.3.SceneBuilder

Le JavaFX SceneBuilder est un outil initialement créé par Oracle et désormais faisant partie de l'OpenJFX qui permet de créer des fichiers au format FXML via un éditeur graphique. Cet outil est disponible en tant qu'application autosuffisante qui peut être lancée depuis votre bureau ou en tant qu'API intégrable dans des outils tiers tels que NetBeans.[13].

### IV.1.4.FXML

Le FXML est un format de fichier propre à JavaFX et servant, entre autres, à définir des interfaces graphiques. Ce format utilise tout simplement une syntaxe XML.

Bien que ce format de fichier soit avant tout destiné à définir des contrôles et écrans, vous pouvez en fait initialiser pratiquement n'importe quel type d'objet Java en FXML, même si cet objet n'a aucune vocation graphique. Il faudra cependant veiller à ce que votre classe soit publique et dispose d'un constructeur par défaut ou d'un constructeur sans paramètre qui est public et qu'elle soit accessible sur le CLASSPATH lors du chargement du fichier FXML [14].

### IV.1.5.JDBC (Java Database Connectivity)

Est un ensemble de classes Java qui permet de se connecter à une base de données distante sur le réseau, et d'interroger cette base afin d'en extraire des

données, Ce paquetage permet de formuler et gérer les requêtes aux bases de données relationnelles.

Pour travailler avec un SGBD il faudrait disposer des classes drivers qui implémentent les interfaces JDBC.

#### **IV.1.6.SQL (Structured Query Language)**

SQL est le langage des bases de données relationnelles répondant à la fois à la problématique de création des objets de bases de données (modèle), de manipulation des données (algèbre relationnelle), de gestion de la sécurité (« droits d'accès »), de traitements locaux de données (procédures). De plus, il est désormais doté d'extensions objet [15].

#### **IV.1.7.PHPMyAdmin**

PhpMyAdmin est un outil logiciel libre écrit en PHP , destiné à gérer l'administration de MySQL sur le Web. PhpMyAdmin prend en charge un large éventail d'opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) peuvent être effectuées via l'interface utilisateur, tandis que vous avez toujours la possibilité d'exécuter directement toute instruction SQL.(  
PhpMyAdmin Documentation Release 5.1.0-dev)

#### **IV.1.8.JAVA**

Java est un langage de programmation dit compilé avec un aspect orientée objet, Une de ses plus grandes forces est son excellente portabilité : une fois votre programme créé, il fonctionnera automatiquement sous Windows, Mac, Linux, Java permet de réaliser une très grande quantité d'applications différentes.[16].

### IV.1.9.JAVAFX

Est une famille de technologies moderne permettant de développer des applications visuellement riches sur divers appareils, JavaFX contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc[17].

Le SDK (Software Development Kit) de JavaFX n'est plus intégré au JDK (Java Development Kit) standard depuis la version 12 cette bibliothèque est disponible sur le projet OpenJFX.

## IV.2.Schéma physique de la base de données

Pour l'implémentation de la base de données nous avons utilisé l'application web PhpMyAdmin qu'est un outil intégré dans WampServer, ce qui nous a permis de schématiser les tables et leurs relations distinctes, à travers cette opération nous avons indiqué le type de chaque attribut de chaque table, les clés primaires, les clés étrangères ainsi que les champs référence comme indiqué dans la **Figure IV.1.**

## IV.3.Architecture de l'application

Après avoir exposé l'ossature de la partie données de notre application, nous allons épiloguer l'architecture de notre application en terme de déploiement des modules qui la composent sur le différent terminaux (Ordinateur, serveur) ainsi nous terminerons par une présentation de la structure du code source de notre application.



## IV.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...) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) sur ces nœuds. C'est un véritable réseau constitué de nœuds et de connexions entre ces nœuds qui modélise cette architecture [7].

## IV.5. Structure du code source

Pour apporter une plus d'éclaircissement au diagramme de déploiement nous allons détailler l'armature de notre code source, que nous avons organisé selon le concept du design pattern « MVC » qui sépare le code en trois niveaux ainsi il définit le rôle de chaque fichier

- **Modèle** : cette partie gère les données de notre application. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.
- **Vue** : cette partie entièrement programmé en FXML à l'aide du Framework SceneBuilder, assure l'interaction entre les utilisateurs et la machine autrement dit c'est la partie visible de l'application ou front-end.
- **Contrôleur** : cette partie gère la logique du code qui prend des *décisions*. C'est en quelque sorte l'intermédiaire entre le modèle et la vue, son codage est du pure Java.

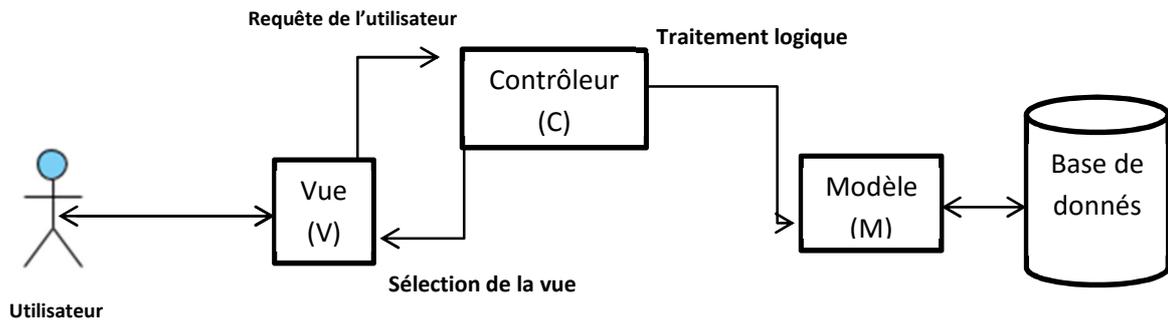


Figure IV.2 Architecture du code

## IV.6. Arborescence de l'application

Ce point va nous permettre de visualiser les imbrications et les connexions entre les multiples vues de notre application de sorte à respecter les besoins indiqués ultérieurement.

La figure suivante va faire office d'illustrer cette arborescence plus clairement :

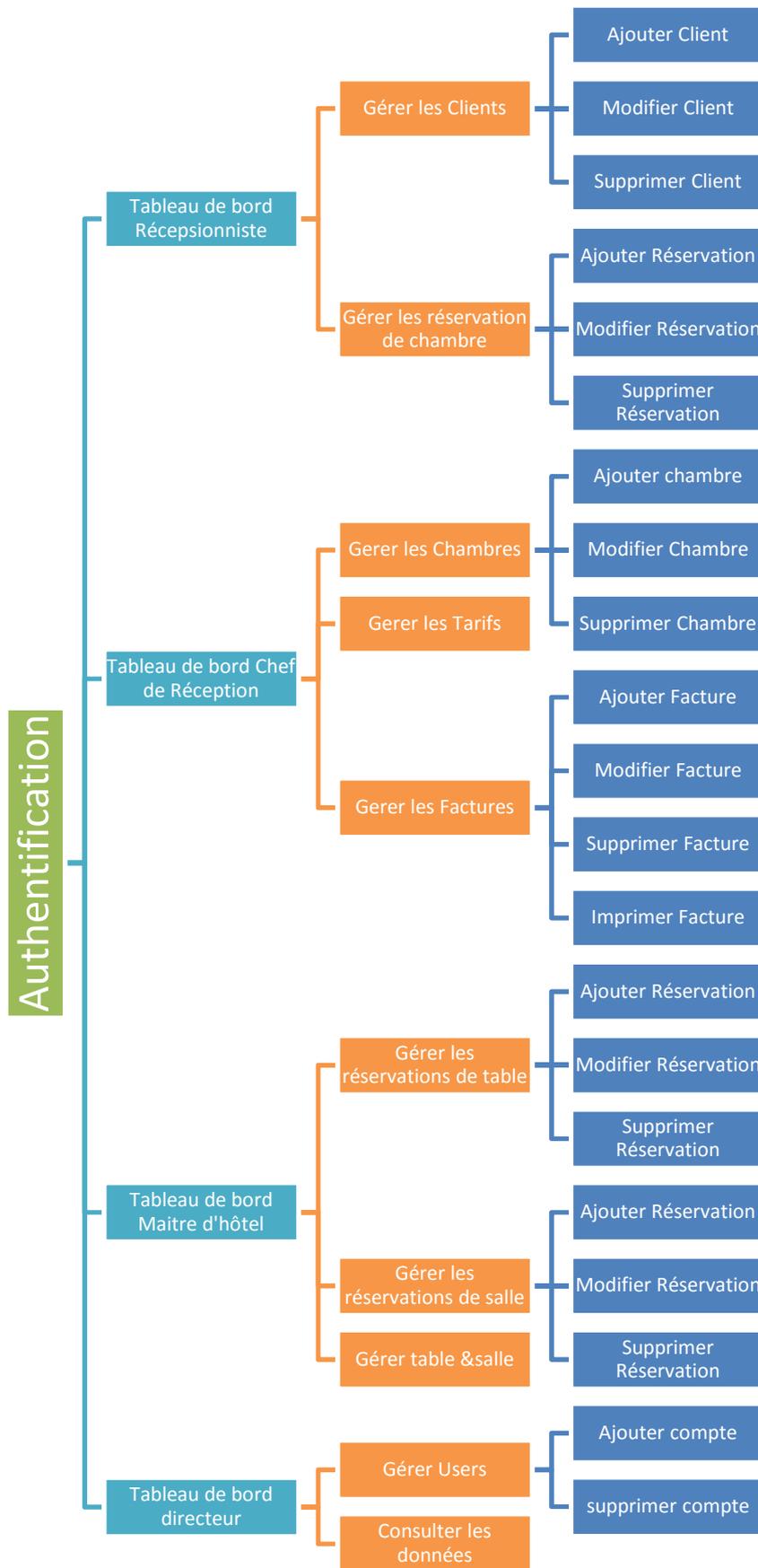


Figure VI.3 Arborescence des interfaces de l'application

## IV.7.Aspect graphique de l'application

### IV.7.1.Logo de l'établissement

Pour être en total conformité avec la nouvelle stratégie marketing et pour rester dans l'air du temps et redynamiser son image, la direction de l'établissement avait formulé son souhait de moderniser l'aspect graphique et le logo de sorte à ce qu'il soit représentatif.

En se penchant sur les principes de base de l'infographie et de la création des logos [18], au terme de notre travail nous avons débouché sur le logo suivant :



Figure VI.4 Nouveau logo de l'hôtel ROYALE

### IV.7.2.Capture d'écran

Dans ce point nous allons exposer les différentes interfaces de notre application associées aux multiples acteurs que nous avons identifiés ultérieurement, ces interfaces demeurent le seul point de contact entre les utilisateurs et le système.

#### IV.7.2.1.Interface d'authentification

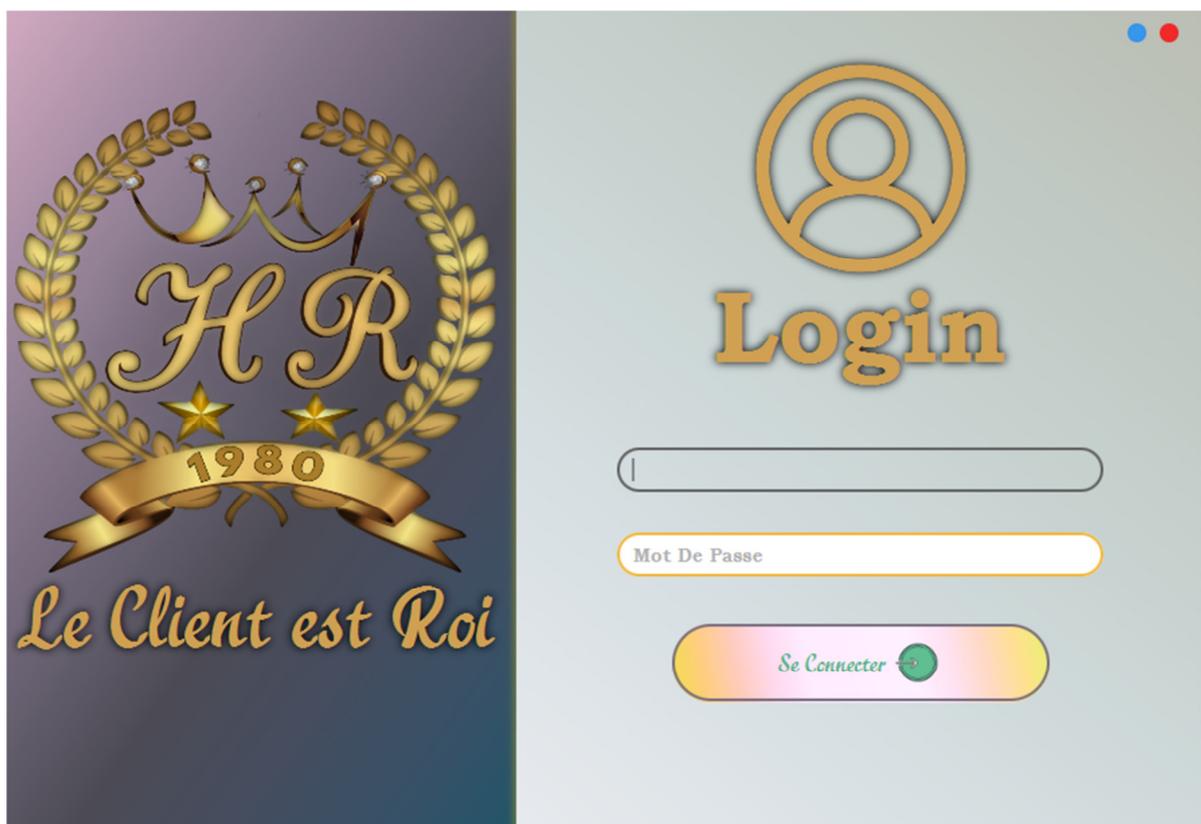


Figure VI.5 Interface d'authentification

Cette interface est le premier point de contact entre les utilisateurs et l'application, une fois l'authentification réussie l'utilisateur sera redirigé vers l'espace adéquat à son rôle.

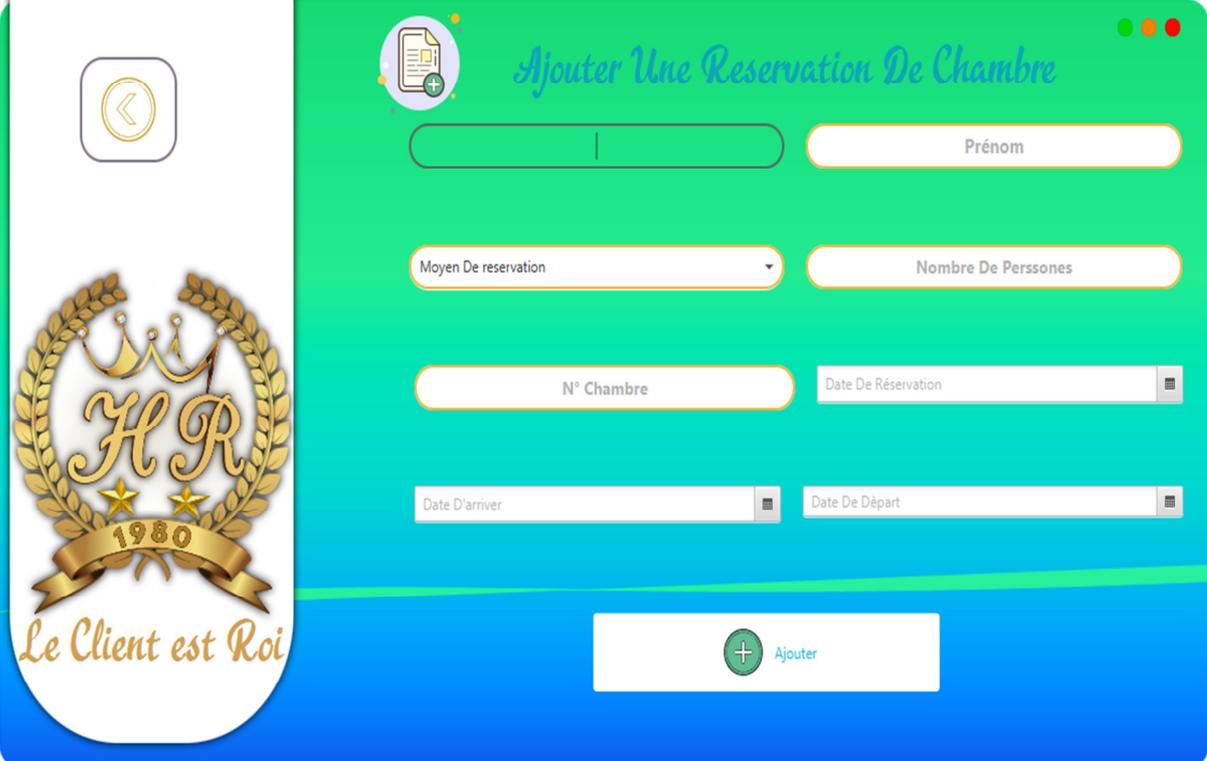
## IV.7.2.2 Tableau de Bord



Figure VI.6 Tableau de bord du chef de réception

Les tableaux de bords occupent une place centrale dans nos interfaces, ils permettent d'organiser d'une façon très lucide les fonctionnalités propres aux utilisateurs en fonction de leurs rôles, cette interface respecte les règles de l'ergonomie des IHM un simple clic suffit à ouvrir l'une des options disponible, en plus comme toutes les autres interfaces elle est adéquate a l'utilisation via les écrans tactiles.

## IV.7.2.3. Ajouter une réservation de chambre



The image shows a web application interface for adding a room reservation. On the left side, there is a logo for 'H R 1980' with the slogan 'Le Client est Roi'. The main interface is a form titled 'Ajouter Une Reservation De Chambre' with a green header and a blue gradient footer. The form contains several input fields: a text field for 'Prénom', a dropdown menu for 'Moyen De reservation', a text field for 'Nombre De Persones', a text field for 'N° Chambre', and three date pickers for 'Date De Réservation', 'Date D'arriver', and 'Date De Départ'. A green '+ Ajouter' button is located at the bottom center of the form.

Figure VI.7 Interface d'ajout d'un client

Cette interface simple d'utilisation permet au réceptionniste d'ajouter une réservation en renseignant les champs et en sélectionnant une date depuis l'objet DatePicker.

## Conclusion

Dans ce dernier chapitre, nous avons représenté une phase très importante de notre travail qui consiste à la partie pratique considérée comme la partie finale de l'aboutissement du processus de développement de notre application ou nous avons exposé les différentes techniques, outils et langages de programmation que nous avons adoptée pour la concrétisation de notre démarche. Par la suite nous avons montré la structure de notre base de données à travers les tables qui la constitue et les interrelations. Ensuite une présentation de l'architecture de notre application et l'organisation du code source.

Nous avons clôturé notre travail par une exposition des aspects graphiques de notre application notamment la réalisation du nouveau logo de l'hôtel ROYALE et une présentation de quelques interfaces utilisateur.

**CONCLUSION**

**GENERALE**

# CONCLUSION GENERALE

---

Le facteur temps et l'organisation sont considérés comme des paramètres fondamentaux dans le monde des entreprises, en se rabattant sur l'informatisation ces dernières ont réussi à passer un cap dans leurs gestions.

Ceci nous a motivé à réaliser ce travail qui s'inscrit dans le cadre de la conception et réalisation d'une application sous réseau pour la gestion et l'automatisation des certaines tâches au sein de l'hôtel ROYAL.

Pour mener à bien cette mission et mettre en œuvre notre projet nous avons effectué un stage dans l'établissement hôtelier où nous avons au premier lieu découvert l'organisme d'accueil ainsi que son organigramme d'organisation ce qui nous a permis d'analyser la problématique, cette partie constitue le premier pas de notre travail pour concevoir une solution efficace et bénéfique et qui répond aux exigences et aux besoins recueilli .

Par la suite, nous avons entamé la seconde phase qui consiste à analyser les besoins déjà recueillis pour cela nous avons identifié les différents acteurs qui interagissent avec le système réalisé. Par la suite nous nous sommes penché sur la modélisation fonctionnelle à travers les diagrammes dynamiques dédiés à cet

effet et la description textuelle des cas d'utilisation ceci dans l'objectif de bien spécifier ce que le système doit faire.

Une fois les fonctionnalités du système ont été analysée nous avons entamé la troisième partie consacrée à la conception Orienté objet du système on se basant principalement sur le diagramme de séquence d'interaction. Ce diagramme a permet de mieux l'illustrer les échanges entre les objets du système puis nous avons élaboré le diagramme de classes et un schéma logique des données.

La dernière partie de notre étude a été dédiée à la réalisation de notre travail en spécifiant les outils de développement et les langages de programmation utilisés pour la concrétisation de notre application l'application, ensuite nous avons présenté l'architecture de cette dernière, suivi d'un aperçu des aspects graphique et des interfaces du logiciel.

Au cours de cette expérience professionnelle très bénéfique nous avons pu enrichir nos connaissances déjà acquises au long de notre cursus et de mettre en pratique le savoir cumulé durant cette période pas seulement en matière de conception et de théorie mais aussi dans la programmation et le développement de bases de données, sur le plan pratique .Ce tremplin nous a permis aussi de nous rapprocher un peu plus du monde professionnel.

De ce projet, nous espérons qu'il puisse servir comme un outil d'aide aux promotions avenir et leur procurer les informations dont ils auront besoin, une base de travail pour l'établissement concerne.

# BIBLIOGRAPHIE

---

- [1] **LAFDAL, R. (2018).** « *Conception et réalisation d'une application pour le suivi des opérations de contrôles des appareils disconnecteurs* »  
<https://fr.scribd.com/document/469775059/Memoire-Final>
- [2] **Pascal, R. (2009).** « *UML 2 PAR LA PRATIQUE. ETUDES DE CAS ET EXERCICES CORRIGES : ETUDES DE CAS ET EXERCICES CORRIGES (EYROLLES). Dans UML 2 PAR LA PRATIQUE. ETUDES DE CAS ET EXERCICES CORRIGES : ETUDES DE CAS ET EXERCICES CORRIGES* » (EYROLLES éd 1<sup>er</sup>. p. 23). EYROLLES.
- [3] **Audibert, L. (2014).** « *UML 2 - De l'apprentissage à la pratique* » 1e édition (Info +). ELLIPSES.
- [4] **Capuozzo, O. (2004).** « *Cas d'utilisation, une introduction. CERTA* »  
<http://www.academia.edu/download/54635619/uc-intro.pdf>
- [5] **Cockburn, A. (2001).** « *Rédiger des cas d'utilisation efficaces* » (Technologies objet) (French Edition) (1<sup>re</sup> éd.). EYROLLES.
- Villemin, F.-Y. (2015).** « *Serveur Cortes* ».  
<http://deptinfo.cnam.fr/Enseignement/CycleSpecialisation/MAI/Doc/Dynamicue13.pdf>  
<http://deptinfo.cnam.fr/Enseignement/CycleSpecialisation/MAI/index.html>

- [6] Gabay, J., & Gabay, D. (2008). « *UML 2 Analyse et conception - Mise en œuvre guidée avec études de cas : Mise en œuvre guidée avec études de cas (UML (1))* » (French Edition) (DUNOD éd.). DUNOD.
- [7] Audibert, L. (2014). « *UML 2 - De l'apprentissage à la pratique* » 2e édition (Info +). ELLIPSES.
- [8] Roques, P., & Vallée, F. (2007). « *UML 2 en action : De l'analyse des besoins à la conception* » (Architecte logiciel) (French Edition) (EYROLLES éd.). Eyrolles Group.
- [9] Roques, P. (2006). « *UML 2 par la pratique* » (5<sup>e</sup> éd.). Eyrolles.
- [10] Soutou, C. (2012). « *UML2 POUR LES BASES DE DONNEES. MODELISATION, NORMALISATION, GENERATION SQL, OUTI : MODELISATION, NORMALISATION, GENERATION, SQL, OUTILS.* » (EYROLLES éd.). EYROLLES.
- [11] *WampServer*. (2020, 14 août). <https://www.wampserver.com/> consulté : **Septembre 2020**
- [12] Bouyé, F. (2014, 6 octobre). *FAQ JavaFX, le club des développeurs et IT Pro*. Developpez.com.  
<https://java.developpez.com/faq/javafx?page=SceneBuilder#:%7E:text=Le%20JavaFX%20Scene%20Builder%20est,FXML%20via%20un%20%C3%A9diteur%20graphique.&text=Oui%2C%20%C3%A0%20partir%20de%20sa,le%20Mercurial%20de%20l'OpenJFX> Consulté : **avril 2020**
- [13] Bouyé, F. (2016, 2 novembre). *FAQ JavaFX, le club des développeurs et IT Pro*. Developpez.com.  
<https://java.developpez.com/faq/javafx?page=FXML-et-SceneBuilder>  
 Consulté : **avril 2020**

- [14] Brouard, F., Soutou, C., & Bruchez, R. (2010). « *SQL 3E ED. SYNTHES* » (SYNTHES INFORMATIQUE) (French Edition) (3<sup>e</sup> éd.). PEARSON.
- [15] Herby, C. (2012). « *APPRENEZ A PROGRAMMER EN JAVA 2E EDITION : LA PROGRAMMATION PROFESSIONNELLE A LA PORTEE DE TOUS* » (2<sup>e</sup> éd.). EYROLLES.
- [16] Simon, M. (2009). “*JavaFX in Action by Simon Morris*” (1<sup>re</sup> éd.). Manning Publications.
- [17] Chaigne, F. (2010). « *GUIDE PRATIQUE À L’USAGE DES ENTREPRISES POUR CRÉER OU CHANGER DE LOGO* » (1<sup>re</sup> éd.) [E-book].  
<https://www.inaativ.com/e-books-graphisme/guide-pratique-a-lusage-des-entreprises-pour-creer-ou-changer-de-logo.html> Consulté :  
septembre 2020
- [18] Roques, P., & Vallée, F. (2000). « *UML en action* » (1<sup>re</sup> éd.). Eyrolles.

# ANNEXE

# ANNEXE A

## A Cas d'utilisation

### A.0.1 Cas d'utilisation « gérer les réservations de chambre »

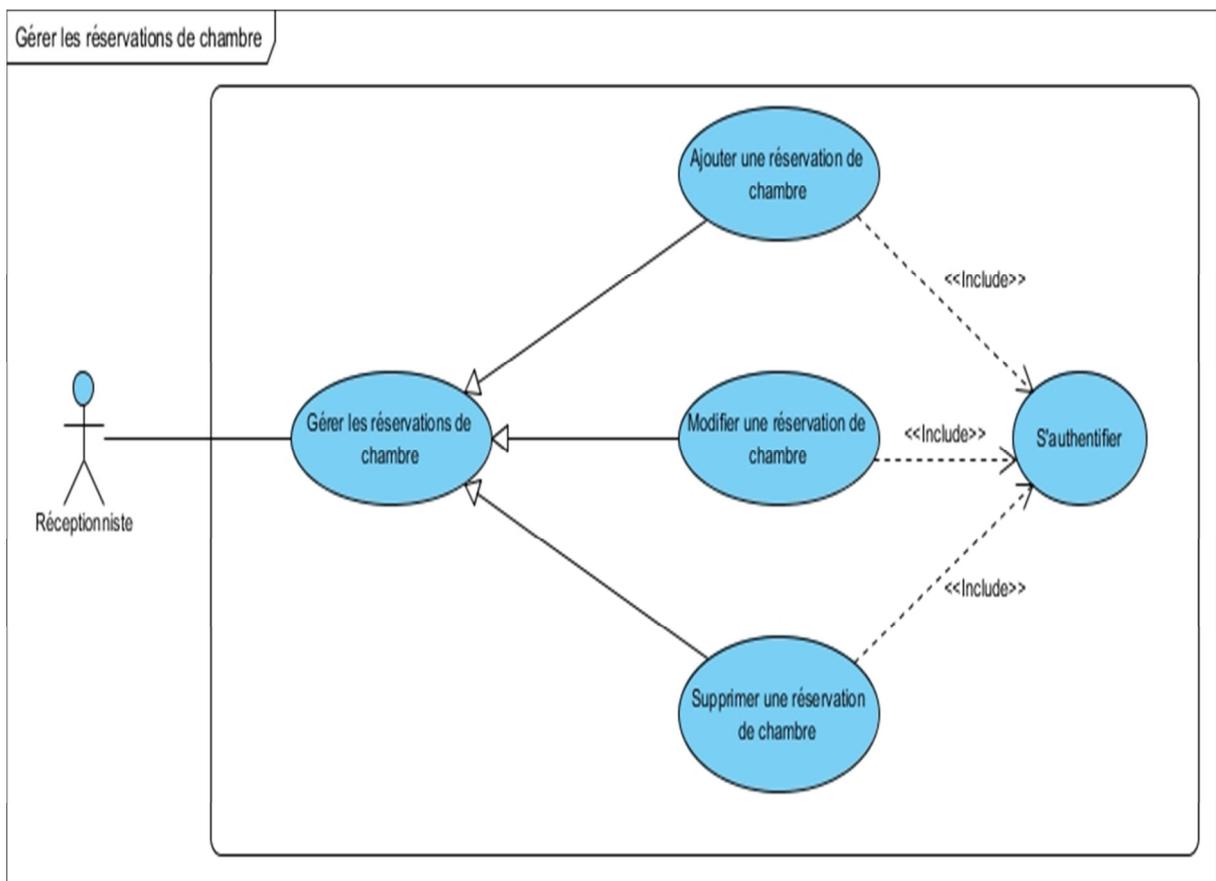


Figure A.1 Cas d'utilisation « gérer les réservations de chambre »

**A.0.2 Description textuelle du cas d'utilisation « gérer les réservations de chambres »**

<b>Numéro du cas d'utilisation</b>	N°3
<b>Titre du cas</b>	Gérer les réservations
<b>But</b>	Effectuer les opérations d'ajout, de modification et de suppression sur les réservations de chambre
<b>Acteurs</b>	Réceptionniste
<b>Pré condition</b>	S'authentifier
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> <li style="padding-left: 20px;">Pour la modification :</li> <li>4. Le système renvoi le tableau des réservations ainsi que le formulaire de modification</li> <li>5. Le réceptionniste effectue une recherche <ul style="list-style-type: none"> <li>• Si la réservation est introuvable exécuter l'exception (1)</li> <li>• Sinon aller a (6)</li> </ul> </li> <li>6. Le réceptionniste envoie le formulaire <ul style="list-style-type: none"> <li>• Si le formulaire les données ne sont pas complètes aller à l'exception (2)</li> <li>• Sinon exécuter (7)</li> </ul> </li> <li>7. Valider l'action</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. La réservation est introuvable renvoyer la même</li> </ol>

	<p>interface</p> <p>2. Un champ vide a été détecté réafficher le tableau et le formulaire</p>
<b>Besoin IHM</b>	Tableau des réservations, le formulaire de modification et une barre de recherche.

**A.0.3 Diagramme d'activité « gérer les clients » :**

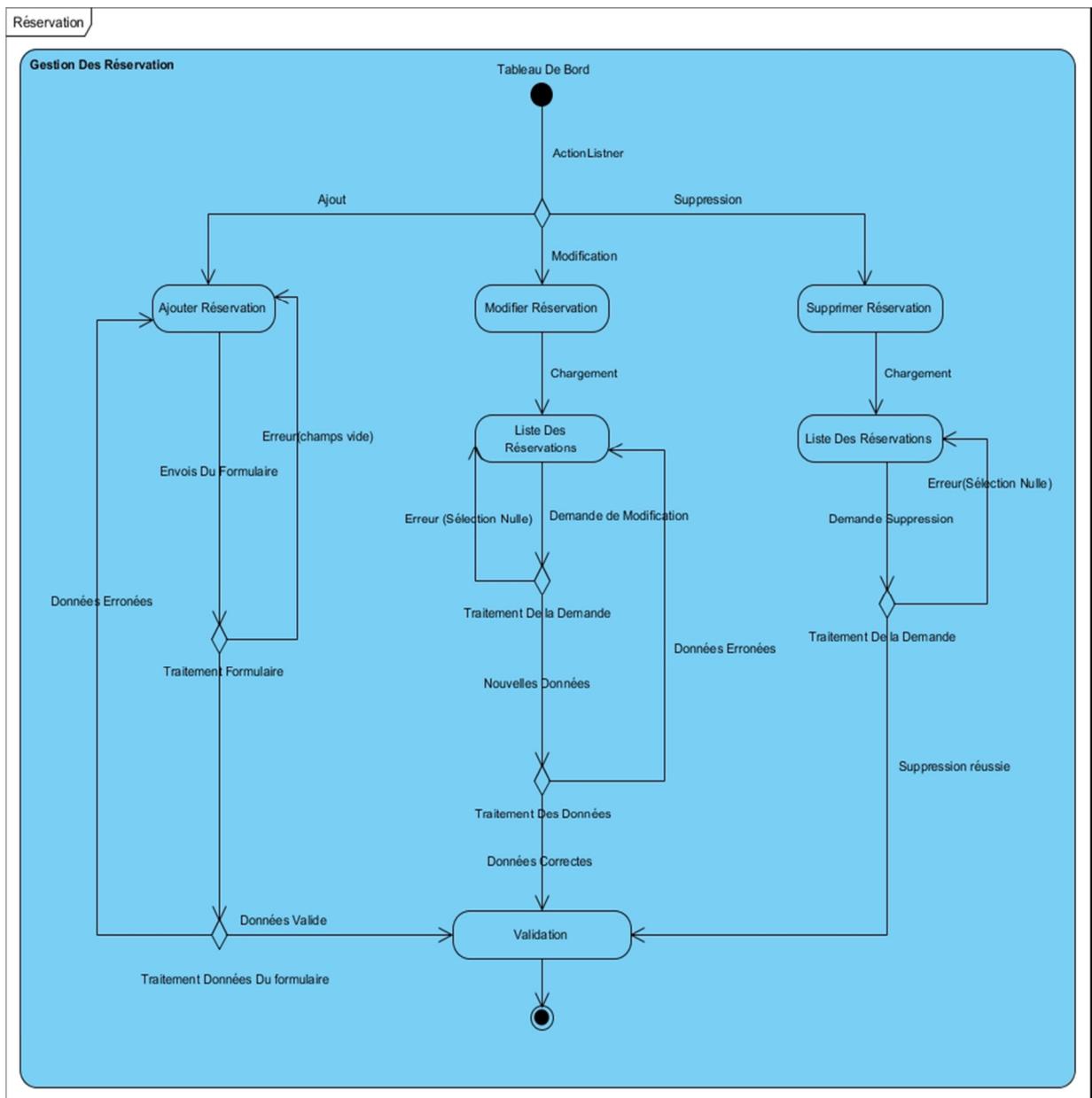
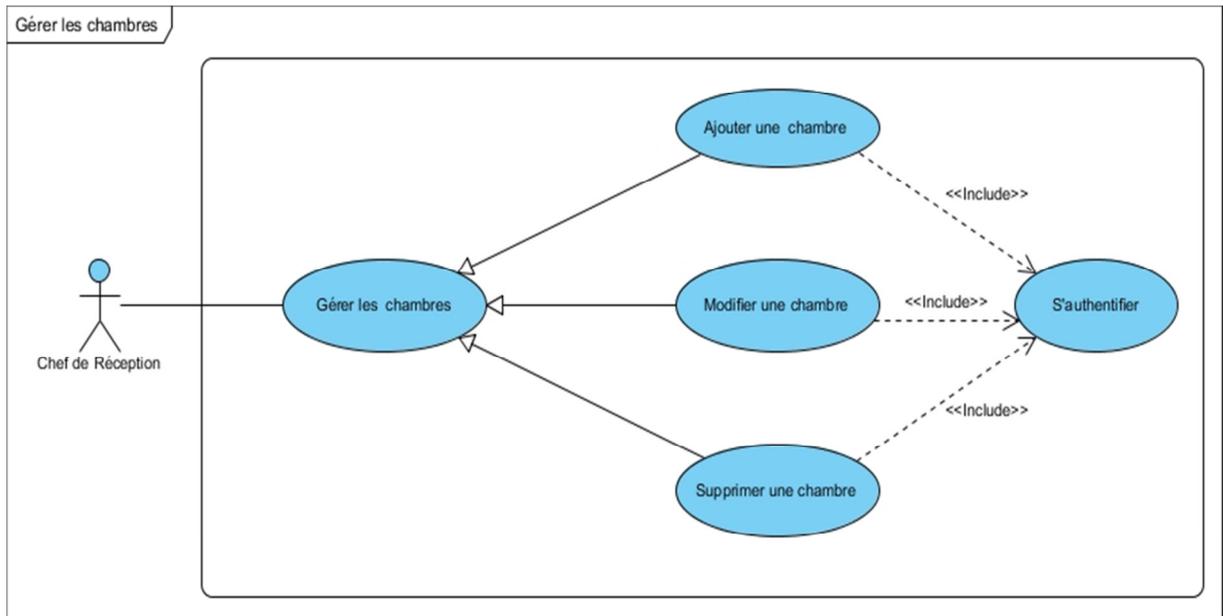


Figure A.2 Diagramme d'activité « gérer les clients »

**A.0.4 Cas d'utilisation « gérer les chambres »**



**Figure A.3 Cas d'utilisation « gérer les chambres »**

**A.0.5 Description textuelle du cas « Gérer les chambres »**

<b>Numéro du cas d'utilisation</b>	Nom du cas d'utilisation
<b>Titre du cas</b>	Gérer les chambres
<b>But</b>	Gérer les chambres à travers les opérations d'ajout, modification et de suppression
<b>Acteurs</b>	Chef de réception
<b>Pré condition</b>	S'authentifier
<b>Scenario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> <li style="padding-left: 40px;">Pour la suppression :</li> <li>4. Le système renvoi le tableau des chambres</li> <li>5. Le chef de réception effectue une recherche <ul style="list-style-type: none"> <li>• Si la chambre est introuvable exécuter l'exception (1).</li> <li>• Sinon aller a (6)</li> </ul> </li> <li>8. Afficher la chambre recherchée</li> <li>9. Valider l'opération de suppression</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. La chambre indiqué est introuvable et réafficher le tableau des factures.</li> </ol>
<b>Besoin IHM</b>	Tableau des chambres et une barre de recherche.



A.0.7 Cas d'utilisation « Gérer les tarifs »

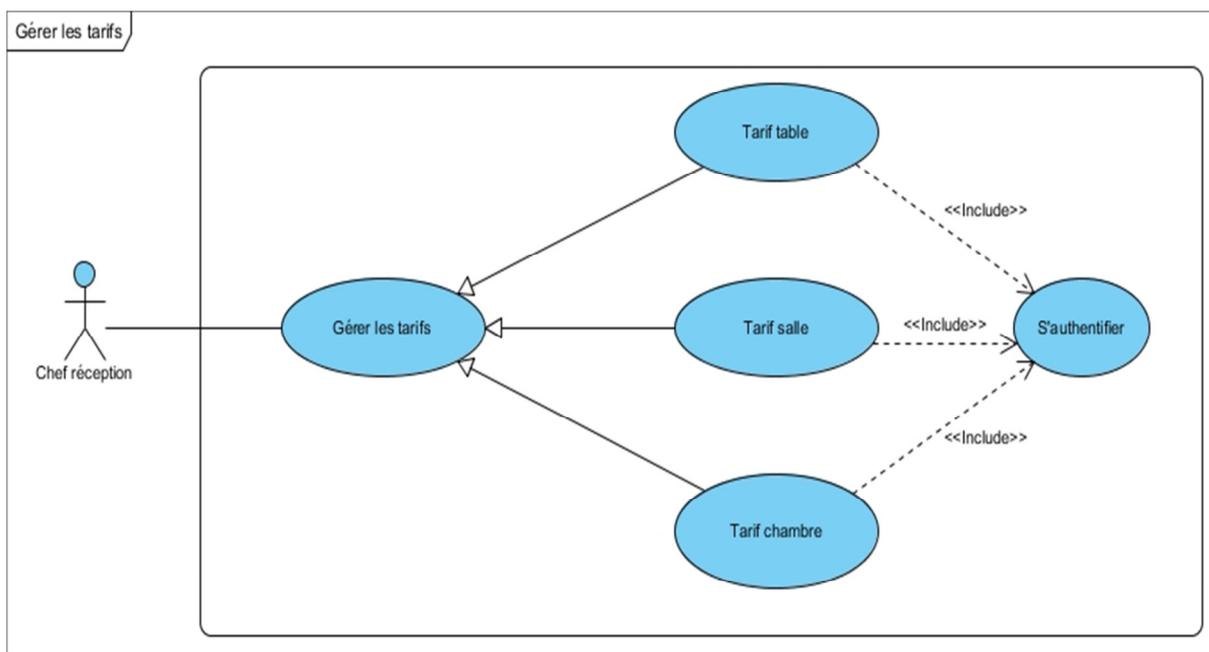


Figure A.5 Cas d'utilisation « gérer les tarifs »

A.0.8 Description textuelle du cas « Gérer les tarifs »

<b>Numéro du cas d'utilisation</b>	N°8
<b>Titre du cas</b>	Gérer les tarifs
<b>But</b>	Permettre la mise à jour des tarifs des chambres, des tables et de la salle des fêtes
<b>Acteurs</b>	Chef de réception
<b>Pré condition</b>	S'authentifier
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> <li>4. Le système renvoi l'interface</li> </ol>

	<p>5. Le chef de réception effectue une recherche</p> <p>Pour les chambres :</p> <ul style="list-style-type: none"> <li>• Si la chambre est introuvable exécuter l'exception (1).</li> <li>• Sinon aller a (6)</li> </ul> <p>6. Afficher la chambre recherchée</p> <p>7. Valider l'opération d'impression</p>
<b>Alternative</b>	<p>2. La chambre est introuvable et réafficher le tableau des chambres</p>
<b>Besoin IHM</b>	<p>Tableau, une barre de recherche et le champ de modification.</p>

### A.0.9 Cas d'utilisation « Gérer les réservations de salle des fêtes »

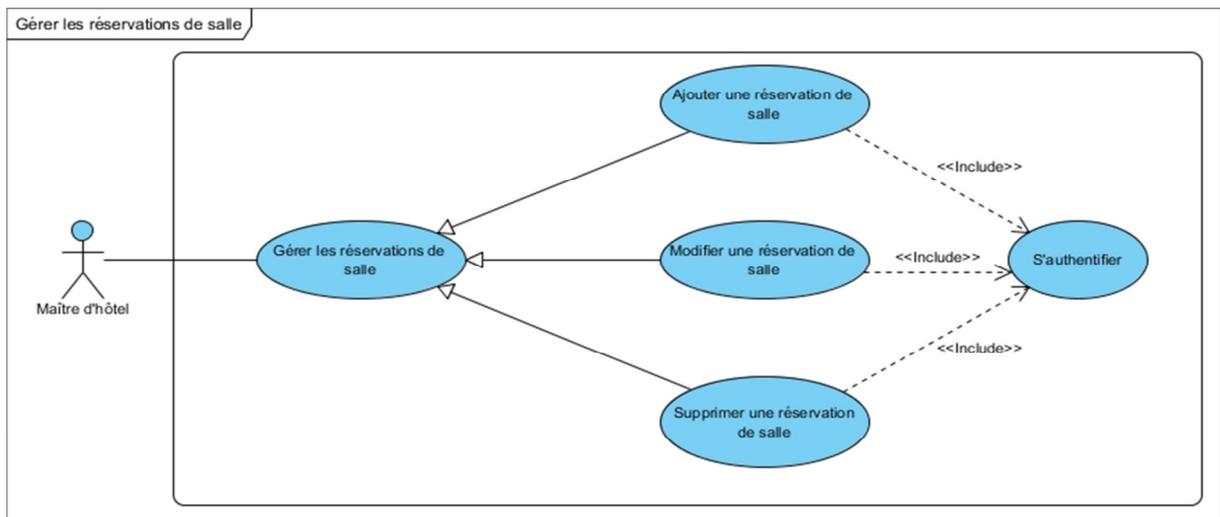


Figure A.6 Cas d'utilisation « gérer les réservations de salle »

**A.0.10 Description textuelle du cas « Gérer réservation de salle des fêtes »**

<b>Numéro du cas d'utilisation</b>	N°10
<b>Titre du cas</b>	Gérer les réservations de salle
<b>But</b>	Permet d'effectuer les opérations de gestion sur les réservations de salle des fêtes
<b>Acteur</b>	Maitre d'hôtel
<b>Pré condition</b>	Le maitre d'hôtel doit s'authentifier
<b>Scenario nominal</b>	<ol style="list-style-type: none"> <li>1. Accéder à l'application</li> <li>2. S'authentifier</li> <li>3. Le système affiche le tableau de bord</li> </ol> <p>Pour l'ajout :</p> <ol style="list-style-type: none"> <li>4. Le maitre d'hôtel accède à l'espace dédié</li> <li>5. Le système affiche le formulaire <ul style="list-style-type: none"> <li>• si les données sont incomplètes exécute l'alternative (1)</li> <li>• Sinon aller (6)</li> </ul> </li> <li>6. Valider l'ajout</li> </ol>
<b>Alternative</b>	1. Un champ vide a été détecté
<b>Besoin IHM</b>	Formulaire d'ajout

A.0.11 Diagramme d'activité « Gérer les tables & salle »

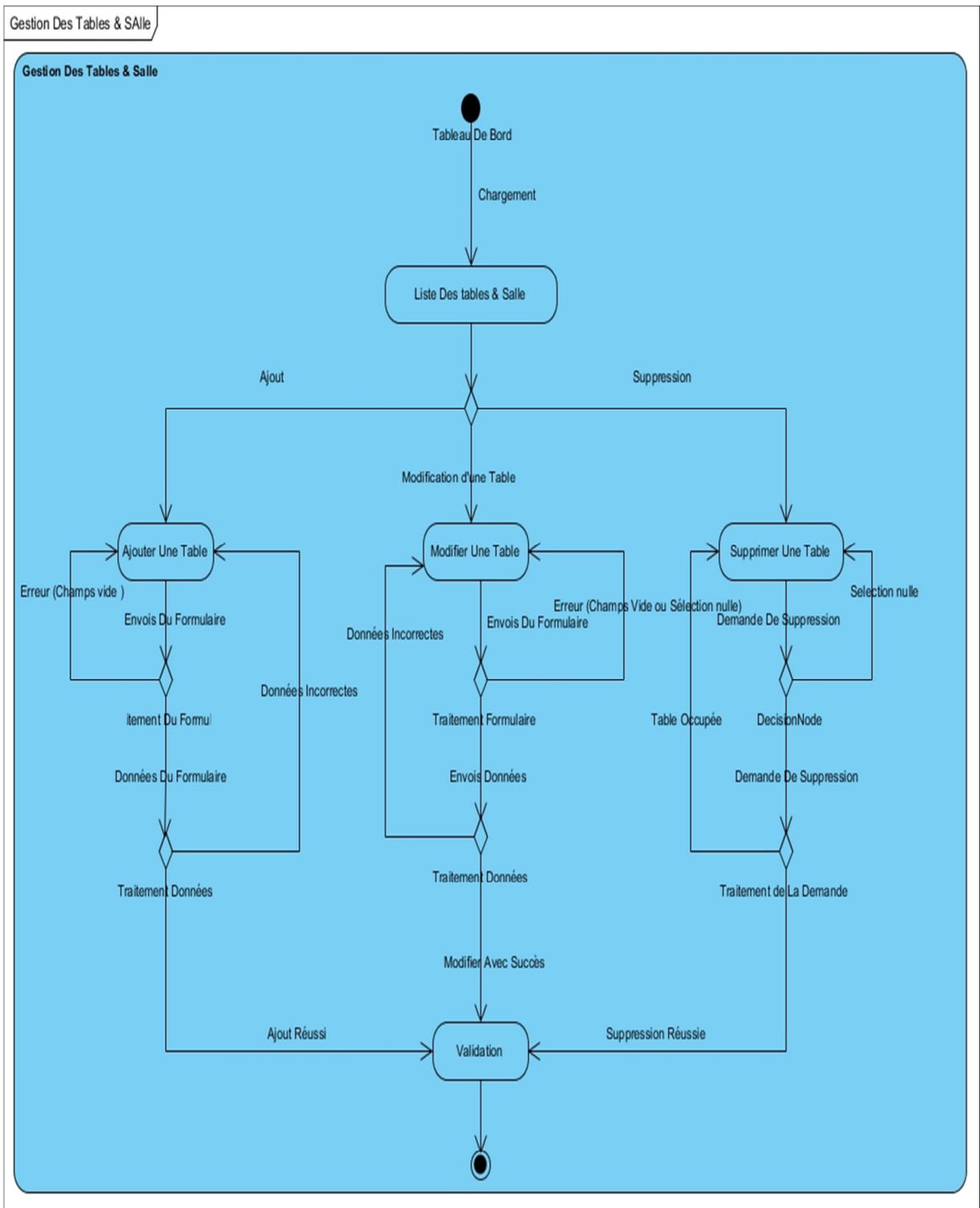


Figure A.7 Diagramme d'activité « gérer les réservations de salle »

A.0.12 Diagramme de Séquence « Gérer les clients »

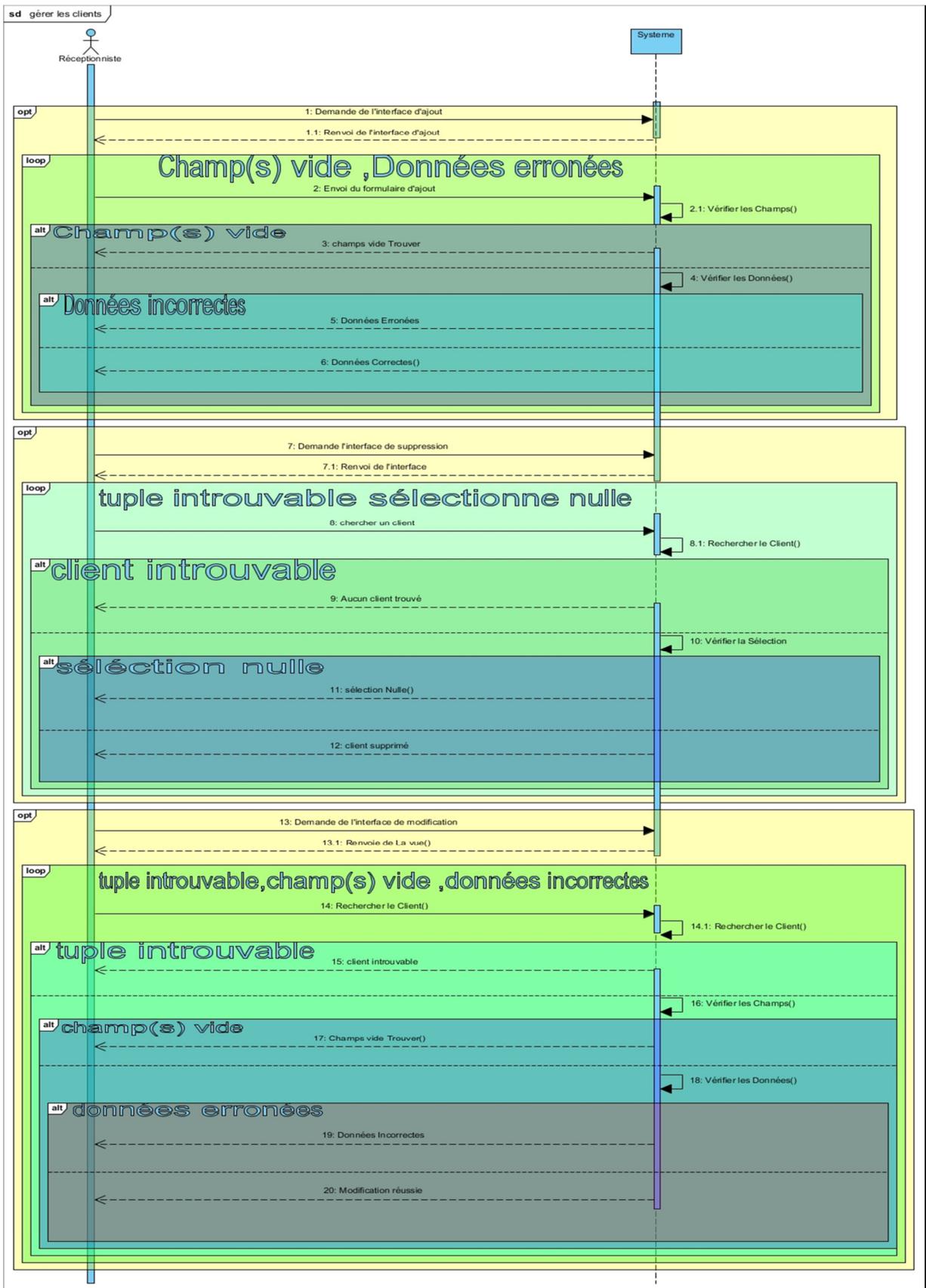


Figure A.8 Diagramme de Séquence « Gérer les clients »

A.0.13 Diagramme de séquence « Gérer les factures »

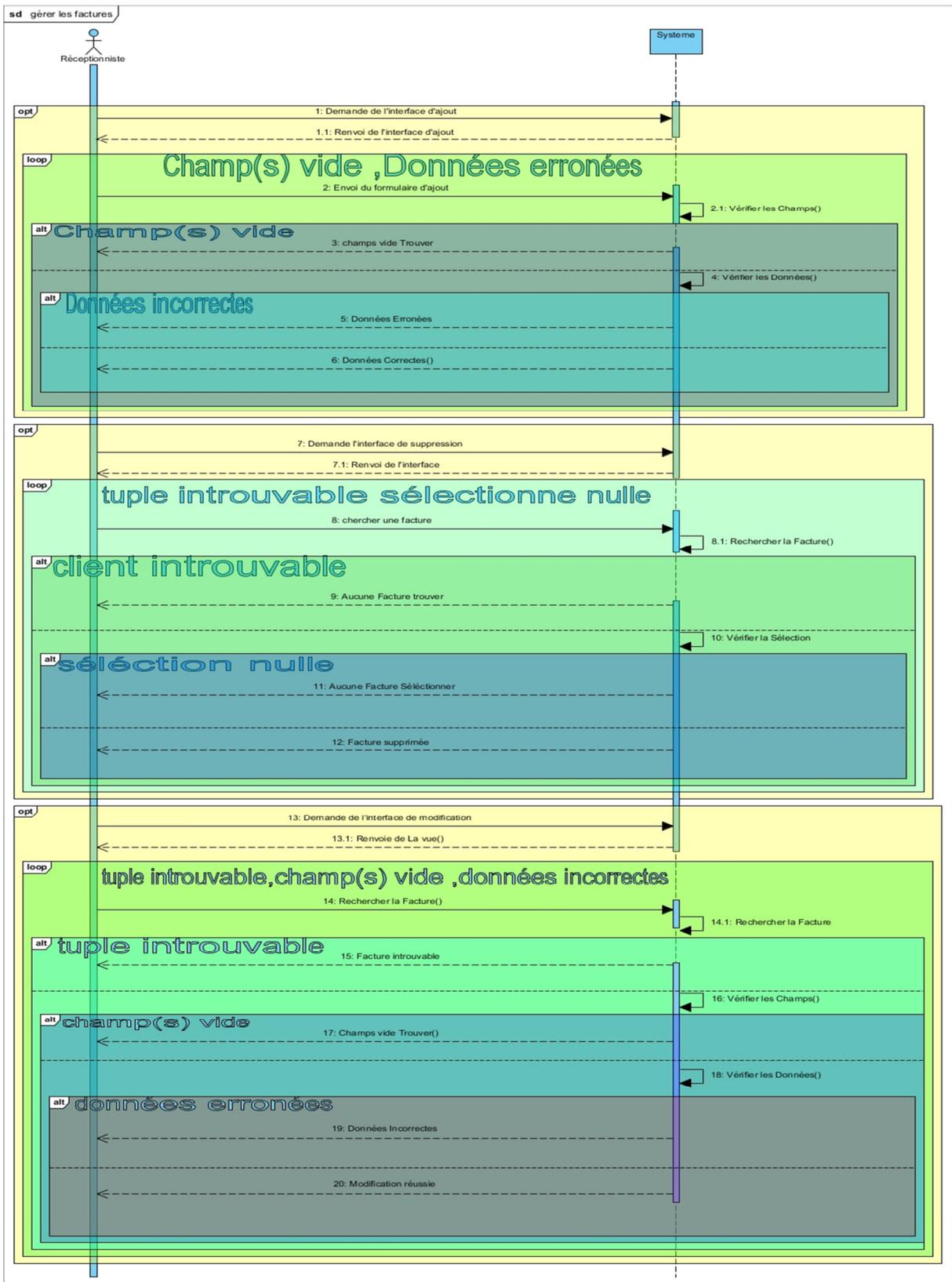


Figure A.9 Diagramme de Séquence « Gérer les factures »

A.0.14 Diagramme de séquence « Imprimer une facture »

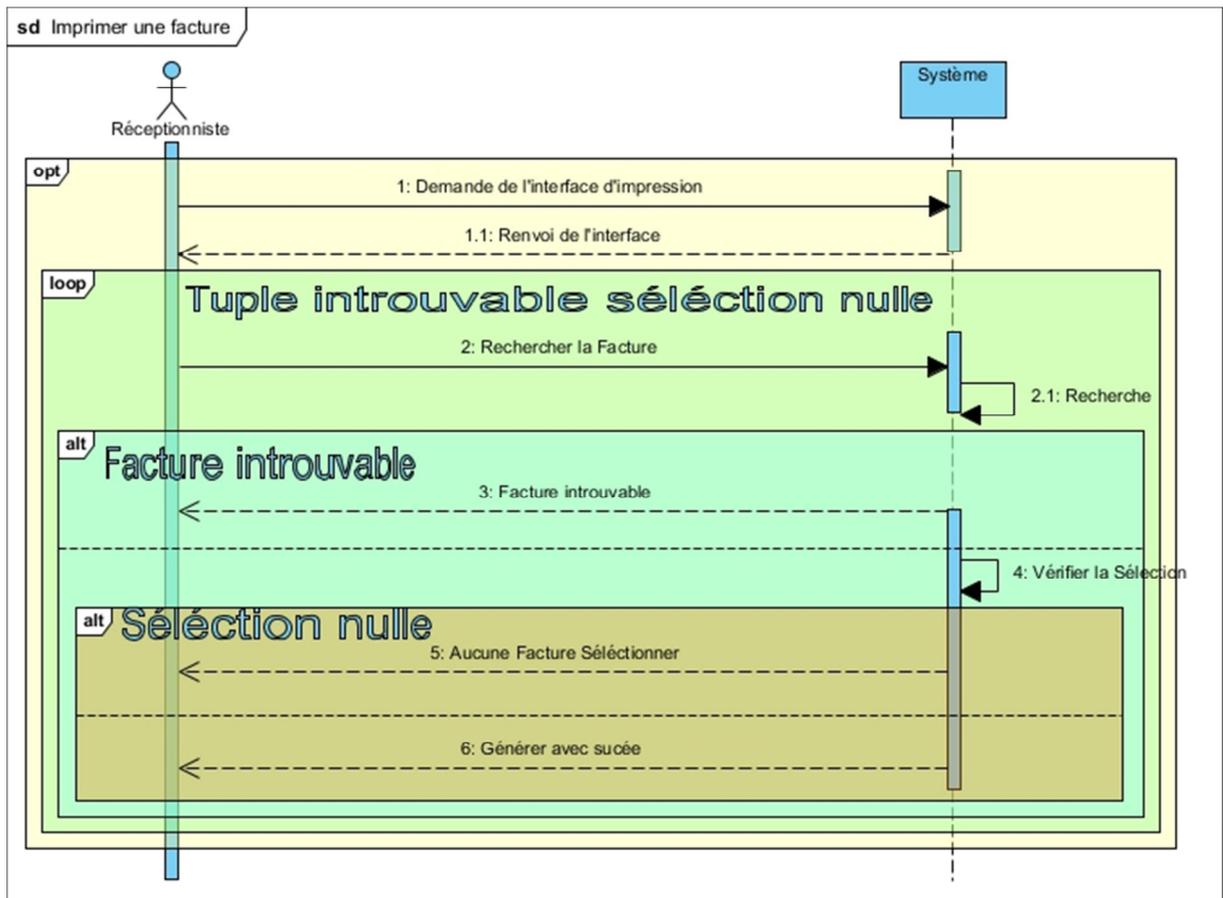


Figure A.10 Diagramme de Séquence « Imprimer une facture »

# ANNEXE B

## B Diagramme de séquence d'interaction

### B.0.1 Diagramme de séquence d'interaction « Gérer les clients »

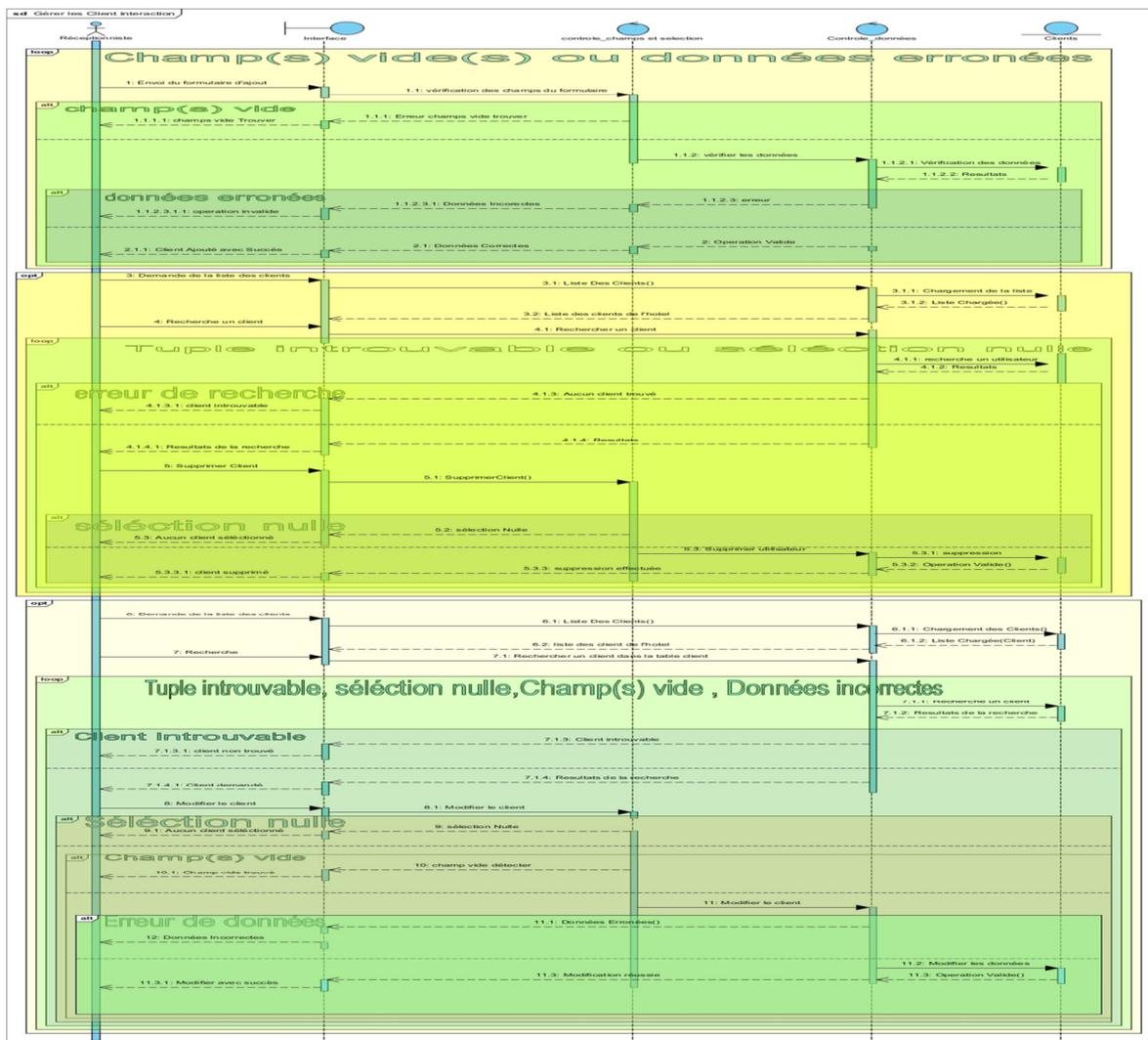


Figure B.1 Diagramme de séquence d'interaction « Gérer les clients »

# ANNEXE C

---

## C Installation & Interfaces

### C.0.1 Installation du module JavaFX

Depuis la sortie de Java 11, JavaFX est disponible en tant que module distinct du JDK.

1. récupérer le SDK de JavaFX en fonction du système sous lequel vous êtes sur le site :

<https://www.gluonhq.com/products/javafx/>.

2. télécharger l'outil Scene Builder qui permet de créer des fichiers FXML avec une interface graphique sur le site :

<https://gluonhq.com/products/scene-builder/#download>

3. Extraire l'archive du SDK de JavaFX dans un dossier (C:\Program Files\Java\jdk-14\openjFX)
4. Paramétrage de l'IDE
  - Allez dans l'onglet "Librairies"
  - Cliquez sur le "+"
  - Sélectionnez New Project Library — Java
  - Renseignez la localisation des fichiers .jar qui sont présents dans le dossier lib du SDK.

5. Pour pouvoir compiler, nous devons modifier les paramètres de la VM. Pour cela, cliquez sur Run — Edit Configurations... et modifiez le champ “**VM options**”, ajoutez ceci : `--module-path "\path\to\javafx-sdk-11\lib" --add-modules=javafx.controls,javafx.fxml` (en changeant le chemin du SDK de JavaFX).

## C.0.2 Les composantes de l’interface graphique de JavaFX :

### 1. Stage :

Le Stage d’une application JavaFX est similaire au cadre d’une application Swing. Il agit comme un conteneur pour tous les objets JavaFX. nous pouvons ajouter divers objets à ce stage primaire. Les objets ne peuvent être ajoutés que de manière hiérarchique.

### 2. Scène :

La scène contient en fait tout le contenu physique (nœuds) d’une application JavaFX. La classe **Javafx.scene.Scene** fournit toutes les méthodes pour traiter un objet de scène. La création d’une scène est nécessaire pour visualiser son contenu. L’objet scène ne peut être ajouté qu’à un seul Stage .

### 3. Les nœuds Graphique de la scène

Le graphe de scène existe au niveau le plus bas de la hiérarchie. Il peut être vu comme la collection de divers nœuds. Un nœud est l’élément qui est visualisé sur la scène. Il peut s’agir de n’importe quel bouton, zone de texte, mise en page, image, bouton radio, case à cocher, etc. Les nœuds sont implémentés dans une sorte de structure arborescente. Il y a toujours une racine dans le graphe de scène. Cela agira comme un nœud père pour

tous les autres nœuds présents dans le graphique de scène. Les nœuds feuilles existent au niveau le plus bas de la hiérarchie de l'arborescence.

### C.0.3 Interface ajouter une facture



Figure C.1 Interface ajouter une facture

### C.0.4 Interface tableau de bord Maître d'hôtel



Figure C.2 Interface tableau de bord Maître d'hôtel

### C.0.5 Interface Modifier une réservation de salle



Figure C.3 Interface Modifier une réservation de salle

### C.0.6 Interface gestion des tables



Figure C.4 Interface gestion des tables

### C.0.7 Interface Imprimer une facture



Figure C.5 Interface Imprimer une facture



## Résumé :

Devenue nécessaire l'informatique a prouvé son utilité et son efficacité dans les domaines des recherches mais aussi dans la vie quotidienne jusqu'à advenir indétournable car elle apporte des solutions efficaces aux nombreux problèmes rencontrés régulièrement telles que l'automatisation des tâches au sein des secteurs de travail et le gain de temps qui est un facteur fondamental dans ces derniers. Ce qui nous a poussés à concevoir et réaliser une application sous réseau pour la gestion de l'hôtel **ROYALE Vgayet**.

Notre application de gestion permet à ses utilisateurs selon leurs rôles d'accéder à une panoplie de fonctionnalités de gestion qu'étaient auparavant énormément pénibles à accomplir. L'étude de l'existant a été menée au sein de l'hôtel « ROYALE ». 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 tels que, WampServer, PHPMyAdmin, le JDK, eclipse etc. Le langage de programmation utilisé est le JavaFX pour un rendu graphique de qualité et de niveau supérieur.

**Mots-clés:** Gestion d'un hôtel, Application réseau, Base de données, UML2, MySQL, WampServer, Java, JavaFX, FXML.

### Abstract:

Computer science has become necessary and has proved its usefulness and effectiveness in the fields of research but also in everyday life until it becomes undetournable because it brings efficient solutions to the many problems encountered regularly such as the automation of tasks within work areas and the time saving which is a fundamental factor in them. This prompted us to design and implement a networked application for the management of the **ROYALE hotel**.

Our management application allows its users according to their roles to access a range of management features that were previously extremely difficult to accomplish. The study of the existing one was carried out within the hotel «ROYALE». To guarantee the completion of the application we chose to model with the UML2 language, a choice dictated by its simplicity and performance in terms of design. MySQL is the database server of the application and the latter was developed using different computer tools such as, Wampserver, PHPMyAdmin, JDK, eclipse etc. The programming language used is JavaFX for high-quality and higher-level graphics.

**Keywords:** Hotel management, Network application, Database, UML2, MySQL, WampServer, Java, JavaFX, FXML.

