

REPUBLIQUE DU SENEGAL

Un Peuple - Un But – Une Foi



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université Amadou Hampathé BA (UAHB)



MEMOIRE DE FIN DE CYCLE POUR L'OBTENTION DU
DIPLOME DE LICENCE PROFESSIONNELLE

SUJET

CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION HOTELIERE CAS DE L'HOTEL NEPTUNE

OPTION : INFORMATIQUE

PRESENTE PAR :

IBRAHIMA LO

ENCADRE PAR :

M. LAMINE DIOP

(Ingénieur en Système d'Information)

ANNEE : 2015/2016

DEDICACES

*Par la grâce de DIEU le miséricordieux, je dédie ce travail à
tous ceux qui me sont chers,
A MES CHÈRES PARENTS*

Moussa LÔ et Adame THIAM

*Que ce travail soit l'expression de ma reconnaissance pour
vos sacrifices
consentis, votre soutien moral et matériel que vous n'avez
cesse de me prodiguer.*

*Vous avez tout fait pour mon bonheur et ma réussite.
Que Dieu vous préserve en bonne santé et vous accorde une
longue vie*

A MES FRÈRES, SŒURS

Mariama Dianké LÔ, Amadou Cupidon LÔ et Mademba LÔ

*Vous êtes toujours présents pour m'aider et m'encourager.
Sachez que vous serez toujours dans mon cœur.*

*A mon oncle Alioune LÔ, merci pour le soutien moral et
matériel*

*A tous mes amis qui n'ont cessé de m'encourager et de me
soutenir*

REMERCIEMENTS

*Nous exprimons nos profondes gratitude et
respectueuse reconnaissance à
notre encadreur:*

M. Lamíne DIOP

*Pour sa bonne volonté d'accepter de nous encadrer,
pour tout le temps qu'il
nous a octroyé et pour tous les conseils qu'il nous a
prodigué.*

*Mon professeur : **M. Baye NIASS***

*Votre rigueur scientifique, votre talent pédagogique
témoignant d'un savoir étendu et d'une grande
compétence.*

*Votre quête du meilleur est un exemple à suivre et
un motif de respect et d'admiration.*

*Faire partie de vos étudiants est une immense fierté
et nous espérons être dignes de votre enseignement.*

*Mes remerciements vont aussi à tous mes
enseignants pour tout ce qu'ils m'ont
appris et pour les beaux moments vécus au cours de
leurs séances.*

*Mes remerciements vont aussi à tout le personnel de
l'hôtel NEPTUNE plus particulièrement à **Nadine
NAEKANE** une réceptionniste stagiaire qui m'a
beaucoup aidé pour l'étude du projet.*

IN MEMORIAM

A la mémoire de tous ceux qui ont contribué à ma réussite, mon éducation et ma formation mais qui ne sont plus là malheureusement :

- *El Hadj Mademba LÔ mon grand-père ;*
- *Mariétou SOKHNA ma grand-mère ;*

Paix à leur âme amine !

Qu'Allah le Tout Puissant les accueille dans son paradis.

Amine !

TABLE DES ILLUSTRATIONS

Figure 1 Organigramme de l'hôtel.....	9
Figure 2 Use Case Générale	22
Figure 3 Use Case Réservation.....	23
Figure 5 Use Case Facturation	24
Figure 6 Diagramme de Séquence Réservation	26
Figure 7 Diagramme de Séquence Facturation	27
Figure 8 Diagramme de Séquence Authentification	28
Figure 10 Architecture 1-Tier	38
Figure 11 Architecture 2-Tier	39
Figure 12 Architecture 3-Tier	39
Figure 13 Architecture n-Tiers	40
Figure 14 MVC.....	41
Figure 15 : Fenêtre de connexion.....	43
Figure 16 Gestion Des Clients	44
Figure 17 Gestion de la catégorie des chambres	45
Figure 18Gestion Des Chambres	46
Figure 19 Gestion Des Réservations.....	47
Figure 20 Diagramme de déploiement.....	51

GLOSSAIRE

API : Application Programming Interface

UML : Unified Modeling Language

POO : Programmation Orientée Objet

IDE : Integrated Development Environment

SGBD : Système de Gestion de Base de Données

ASP : Active Server Pages

PDF : Portable Document Format

SQL : Structured Query Language

SI : Système D'Information

ASCII : American Standard Code for Information Interchange

INTRODUCTION

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement du travail.

Avant l'invention de l'ordinateur, on enregistrerait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières...etc.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui.

Les hôtels font partie intégrante des établissements que l'informatique pourra beaucoup aider. Jusqu'à ce jour, la manière de gérer manuellement est encore dominante, d'où la nécessité d'introduire l'informatique dans la gestion des hôtels.

L'objectif de mon projet de fin de cycle présenté dans ce rapport est la conception et la réalisation d'une application simple pour informatiser la gestion des réservations et la gestion des clients dans un hôtel.

Notre travail est présenté par trois parties :

La première partie sera consacrée à la présentation des besoins fonctionnels et non fonctionnels.

Dans la seconde partie, nous nous intéressons à l'analyse et l'étude conceptuelle et nous détaillons les diagrammes adoptés.

Finalement la dernière partie sera consacrée l'implémentation, au déploiement ainsi qu'à la présentation et la description des différentes interfaces de l'application.

PREMIERE PARTIE CADRE THEORIQUE ET METHODOLOGIQUE

CHAPITRE 1 : CADRE THEORIQUE

I. Présentation Générale de la structure

A.Présentation

Présentant une décoration à la fois sobre et chaleureuse, les 60 suites lumineuses possèdent du carrelage, une télévision ainsi qu'un espace de vie avec canapé-lit et minibar. Certaines suites disposent d'une salle de séjour séparée, tandis que les villas de 2 étages sont équipées d'une kitchenette et d'une terrasse meublée.

L'établissement possède une plage privée, une piscine extérieure, un centre de fitness, un bar et 2 restaurants, dont un qui organise une soirée sénégalaise une fois par semaine sur sa terrasse.

B. Situation Géographique

Situé à 3 km du parcours de 18 trous du golf de Saly, cet hôtel de style décontracté est installé face à la plage, à 80 km de Dakar.

C.Organisation

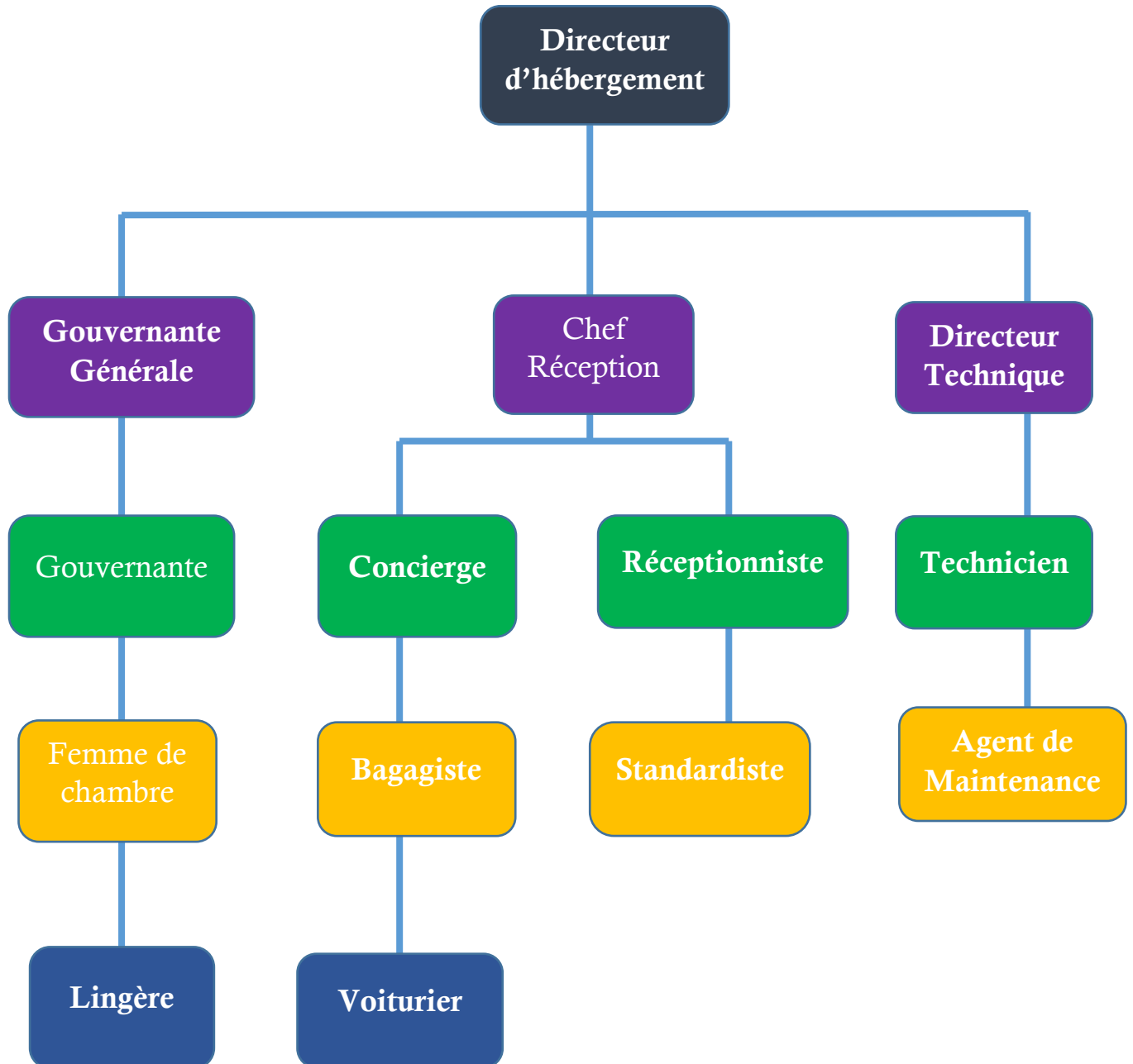


Figure 1 Organigramme de l'hôtel

1. Directeur D'hébergement

Le directeur d'un hôtel confie à son directeur de l'hébergement la gestion des chambres de l'établissement, et à son directeur de la restauration la responsabilité du ou des restaurant(s) de l'hôtel. La tâche principale du directeur de l'hébergement est la gestion de l'enveloppe financière de la partie " chambres " : gestion des coûts et des recettes, élaboration des budgets prévisionnels... Mais la mission essentielle du directeur de l'hébergement est de veiller au taux de remplissage de l'établissement. Pour que le maximum des chambres soit plein, il contrôle, en collaboration avec les responsables de service, la qualité des prestations et la bonne organisation des services. Il peut ainsi décider de nouvelles méthodes de travail ou de mettre en place de nouvelles prestations pour séduire une clientèle plus large (service des petits déjeuners en chambre très tôt le matin, abonnement aux chaînes câblées...). Sur les conseils du directeur, il dirige le personnel des étages et de la réception, participe à leur recrutement et à leur formation. Toujours pour remplir l'hôtel, le directeur de l'hébergement participe à la politique de vente des chambres : élaboration des tarifs, opérations de promotion, relations publiques... Enfin, avec les autres membres de l'équipe de la direction, il veille à appliquer la politique générale de l'hôtel.

2. Gouvernante

La gouvernante est un peu l'alter ego féminin du concierge, mais dans les étages. Comme le concierge, ses tâches sont diverses et variées. Tout d'abord, elle contrôle le travail des femmes de chambre. Après leur passage, elle vérifie la propreté des chambres et des couloirs. En un coup d'œil, elle est capable de juger de la bonne tenue des lieux. C'est elle qui apporte une touche personnelle aux chambres, en ajoutant un bouquet ou en changeant la disposition des meubles. Et quand la direction entreprend des travaux de réfection, on fait appel à son sens de l'organisation et de la décoration. En tant que chef d'une équipe, la gouvernante fixe les emplois du temps de ses femmes de chambre en fonction de l'activité de l'hôtel et leur distribue le travail. Elle gère l'approvisionnement des minibars et du stock des fournitures servant à l'entretien des chambres et des couloirs. Elle rend compte de l'activité du personnel des étages au directeur de l'hébergement ou au directeur. La direction compte aussi sur son expérience pour former efficacement le personnel et peut même lui demander son avis lors d'un recrutement.

La gouvernante a le sens des responsabilités, le goût du détail et sait faire preuve de fermeté comme de diplomatie pour gérer son équipe ou répondre aux réclamations des clients.

3. Chef Réception

Le chef réceptionniste laisse généralement le soin de l'accueil à son ou à ses réceptionniste(s) pour se consacrer aux tâches administratives et commerciales : le planning des réservations, les relations avec les agences de voyages et les centrales de réservation, le secrétariat de l'accueil, la vérification de la caisse... Il organise également des opérations spéciales, comme l'arrivée d'un groupe pour un congrès. On compte aussi beaucoup sur lui pour remplir l'hôtel : à lui de savoir vendre aux " gros clients " (agences, tour-opérateur...) les qualités de son hôtel, d'anticiper une période de creux, de partir à la recherche de la clientèle ou encore d'être en bonne relation avec les concurrents pour se rendre des services, en cas de surréservation par exemple et ainsi de pouvoir s'adresser mutuellement des clients en surnombre. Quand le chef réceptionniste remplit de telles

responsabilités (et en l'absence de directeur de l'hébergement), il peut être considéré comme le bras droit du directeur, qu'il est amené à représenter en son absence. Le chef réceptionniste est aussi un chef de service : il dirige son équipe et gère le planning des prises de service. Enfin, le chef réceptionniste peut être amené à recruter le personnel de l'accueil.

4. Directeur Technique

Son rôle est :

- La tenue de la documentation technique,
- La mise à jour des plans,
- La collaboration à la standardisation des pièces et machines,
- La réalisation des études adaptatives et d'organisation,
- L'établissement du bilan économique et des solutions d'amélioration,

5. Réceptionniste

Derrière son comptoir ou au téléphone, le réceptionniste accueille chaque client. Il lui précise les capacités d'accueil de l'hôtel, en se référant au planning des réservations dont il a la charge. Il communique les tarifs pratiqués et les différents services proposés (petit déjeuner, salle de restaurant, coffre...). A son arrivée, il lui remet sa clef, s'occupe de ses bagages et le guide jusqu'à sa chambre. Le réceptionniste doit aussi être en mesure de renseigner son interlocuteur sur l'environnement proche de l'hôtel : " où se trouve le cinéma le plus proche, quel bus passe à proximité de l'hôtel... " Pour qu'aucune question ne reste sans réponse, un bon réceptionniste connaît sa ville aussi bien que son hôtel. En plus de sa fonction d'accueil et d'information, le réceptionniste effectue des tâches administratives : la gestion, de plus en plus à l'aide de l'outil informatique, des réservations et l'enregistrement des arrivées et des départs. En tenant compte de ces renseignements, c'est lui qui indique aux femmes de ménage les chambres à préparer entièrement pour une arrivée ou simplement à remettre en ordre, on emploie alors le terme de " recouche ", pour une seconde nuit. Quand il n'y a pas de caissier, le réceptionniste prépare la note de chaque client et encaisse les factures. Enfin, quand il n'y a pas de chef réceptionniste, il doit veiller au remplissage de l'hôtel. Ainsi, si l'hôtel doit, par exemple, faire face à un important désistement, il lui faut réagir très vite et informer les agences de réservation que son établissement a plusieurs chambres libres et négocier leur tarif au meilleur prix.

Le réceptionniste est la première et la dernière personne que rencontre un client en franchissant la porte de l'hôtel. Par la qualité de son travail, il défend l'image de l'établissement qui l'emploie. Il doit avoir un sens certain de l'accueil, une présentation impeccable et une bonne dose de sang-froid. Homme de contact, il est nécessairement très organisé : il doit assurer une présence physique permanente à l'accueil, tout en prêtant attention au roulement interne de l'hôtel. En ayant ainsi l'œil sur tout, il sait à tout moment de la journée combien il reste de chambres libres et s'il faut prévoir du personnel en extra pour accueillir un groupe important.

6. Concierge

" Tout voir, tout entendre et ne rien dire ", telle est la devise d'un concierge, que l'on reconnaît grâce aux deux clés d'or entrecroisées, brodées au revers de son veston. En effet, cet homme de confiance doit avoir un œil averti pour reconnaître les clients fidèles et leur réserver un accueil de choix ou repérer ce qui est inhabituel pour intervenir avec diplomatie, une oreille complaisante pour prêter attention aux demandes, même les plus farfelues, de la clientèle et bien sûr, lui apporter satisfaction (leur faire livrer la presse étrangère, réserver des places de concert, prendre un rendez-vous chez le coiffeur...) tout en ne dévoilant rien de ses petits secrets. Un confident dévoué qui est prêt à se plier en quatre pour que le plus exigeant des clients se sente toujours compris et choyé. C'est en grande partie grâce à son concierge qu'un palace arrive à offrir un service personnalisé. Le concierge est aussi le chef du hall : c'est lui qui forme et qui dirige son équipe de voiturier, portier, liftier, bagagiste, portier, groom... auxquels peut se joindre un ou plusieurs assistant(s)-concierge... Il peut même être amené à recruter ce personnel.

Tout en côtoyant de très près une clientèle parfois prestigieuse, le concierge doit toujours se rappeler qu'il est à son service. Discret, distingué, avenant, le concierge doit en outre posséder une bonne culture générale et maîtriser au moins deux langues étrangères afin de pouvoir donner le change à une clientèle raffinée et cosmopolite. Il se doit ainsi d'être au fait des habitudes culturelles de chaque pays.

7. Femme Chambre

Les femmes de chambre et leurs homologues masculins les valets, qui font figure d'exception dans la profession, ont en charge la propreté, l'ordre et le confort des chambres de l'hôtel. Chaque matin, la femme de chambre reçoit son plan de travail qui lui indique quelles sont les chambres à remettre en état, soit, au minimum une dizaine de chambres. La femme de chambre nettoie chambre et salle de bain, change les draps, et si c'est nécessaire, remplit le minibar. Avec ses collègues, elle entretient les parties communes des étages. A noter : les femmes de chambres travaillent généralement le matin : de 7 à 15 heures, et presque toujours en l'absence de la clientèle.

Une bonne femme de chambre est une spécialiste de la propreté. Elle connaît la différence entre " donner un petit coup de neuf " et nettoyer efficacement. Elle ne ménage pas ses efforts, elle connaît les bons gestes et la meilleure façon d'utiliser ses outils et ses produits. Elle sait aussi comment s'organiser pour travailler rapidement, car il n'est pas non plus question de s'éterniser. Une femme de chambre ne passe pas plus de 15 à 20 minutes par chambre. Une bonne femme de chambre est également observatrice : à elle de signaler au responsable des étages toute irrégularité, du peignoir manquant à la literie hors d'usage. Enfin, une bonne femme de chambre est discrète et honnête. Nettoyer une chambre donne accès à l'intimité d'inconnus. Elle peut ainsi remarquer des détails qui devront rester strictement confidentiels ou avoir accès à des objets de valeur.

II. Présentation du projet

A. Contexte

Le système comportera différentes fonctionnalités nécessaires pour une meilleure gestion. L'application doit accomplir les traitements suivants : L'ajout, la modification et la suppression d'un client d'où il est nécessaire d'avoir tous les renseignements qui lui

concerne tel que son nom, son prénom, son numéro de CIN, son numéro de téléphone... Le suivi des chambres des chambres selon leurs types et leurs catégories. La manipulation des réservations : ajout, annulation, modification de la réservation. La facturation de la réservation selon la saison. Ces informations seront stockées dans une base de données qui peut être mise à jour au fur et à mesure des besoins.

B. Problématique

La problématique se définit comme étant un ensemble de questions que se pose un chercheur sur l'objet de sa recherche, lesquelles stimulent l'observation et l'analyse en vue de l'explication de ces dernières.

La gestion des réservations, des clients, des chambres et des paiements rend la tâche ardue au service des réceptionnistes de l'hôtel et la mise en commun des informations enregistrées n'est pas si sûre, pas de suivi en cette matière en ce qui concerne l'enregistrement des réservations, car il arrive d'attribuer une chambre à deux clients.

En effet nous avons détecté après interviews du personnel de l'hôtel les problèmes suivants :

- ❖ La gestion des réservations traitée manuellement
- ❖ Le montant de la facture donné sur calculatrice
- ❖ La disponibilité des chambres informée par le valet de chambre

La problématique de notre travail tournera autour des réflexions ci-dessous, lesquelles comportent des questions fondamentales que voici :

- ✓ Quels apports notre application fournira-t-elle à l'hôtel Neptune?
- ✓ L'implémentation d'un système d'information informatisé, pourrait-elle réduire le risque d'attribution de chambre à deux clients?

Voilà les préoccupations principales et globales, auxquelles nous tenterons de réserver des réponses provisoires.

C. Objectif

1. Objectif Principal

Notre objectif principal est la mise en place d'une application de gestion de réservations, des clients des chambres catégories de chambres, factures et paiements de l'hôtel Neptune.

2. Objectif Spécifique

Pour atteindre l'objectif principal nous nous sommes assigné les objectifs spécifiques suivants :

- Automatisation des tâches qui se font manuellement
- Gérer l'état de la disponibilité des chambres
- Gérer les paiements
- Effectuer une bonne gestion des réservations

D. Délimitation du Champ d'Etude

Nous ne prétendons pas étudier l'univers entier étant un travail en informatique, il nous est difficile de nous délimiter sur le plan temporaire. Cependant dans le cadre de cette étude nous avons délimité notre travail en deux points :

1. Délimitation spatiale

Notre champ d'investigation est la gestion des réservations, des clients, des chambres catégories de chambres et de la facturation.

2. Délimitation du point de vue matérielle (Contrainte)

Concernant ce point, une première conception en phase test nous est proposée par le responsable technique de l'hôtel. Ainsi notre application sera développée et tournera sur une plateforme Windows en monoposte et d'éventualité mise à jour induira à une conception client-serveur.

CHAPITRE 2 : CADRE METHODOLOGIQUE

I. Méthode d'Investigation

A.Méthode

La méthode est une démarche intellectuelle qui permet à l'auteur d'atteindre ses objectifs. Dans le cadre de notre étude, nous nous appuyons sur la conduite de projets informatiques en procédant à la modélisation basée sur le Processus Unifié (UP) qui est un processus du Langage de Modélisation Unifié (Unified Modeling Language) UML en sigle.

B. Technique

La technique est l'outil pratique de la méthode ou encore elle est l'instrument qui permet à la méthode de collecter les informations nécessaires.

En ce qui concerne les techniques utilisées, nous avons utilisé les techniques suivantes pour la collection des informations :

1. Technique d'observation directe

Partant de cette technique, nous avons récolté quelques informations en discrétion, lors de notre passage à l'hôtel Neptune et avons rencontré certaines difficultés que rencontrent les réceptionnistes.

2. Technique d'interview

Nous avons été voir le chef de département de l'état-civil de la commune de Lubumbashi qui de par sa courtoisie nous a accordé une interview qui nous a permis de recueillir les données dont nous avons besoin afin d'avoir un bon moyen pour assurer des taux de réponses à nos questions et inquiétudes.

3. Technique des questionnaires

Ces personnes ont pu répondre au questionnaire élaboré et leurs réponses nous ont permis de recueillir des informations riches et spontanées, plus brutes et plus fraîches nécessaires à notre travail. Avec cette technique, nous avons utilisé les questionnaires ouverts et les questionnaires fermés.

II. Etude comparative de quelque méthode d'analyse

A.Présentation

La conception est une étape fondamentale dans le cycle de vie d'une application informatique. En effet, c'est d'elle que dépendent la qualité et la cohérence du produit

réalisé au développement. Des méthodes de génie logiciel ont alors été développées afin de guider le concepteur dans sa tâche. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

1. MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assurent une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

La méthode MERISE date des années 1978-1979, et fait suite à une consultation nationale lancée en 1977 par le ministère de l'industrie français dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'informations. Les deux principales sociétés ayant mis au point cette méthode est le CTI (Centre Technique d'informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Equipe) chargé de la mise en œuvre. La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et traitements afin de s'assurer que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues. Cette succession d'étape est appelée cycle d'abstraction pour la conception des systèmes d'informations

- Expressions des besoins
- Modèle Conceptuel
- Modèle Logique
- Modèle Physique
- Système d'Information Automatisé

L'expression des besoins est une étape qui consiste à définir ce que l'on attend du système d'information automatisé, il faut pour cela:

- Faire l'inventaire des éléments nécessaires au système d'information.
- Délimiter le système en s'informant auprès des futurs utilisateurs.

Cela va permettre de créer le «MCC» (Modèle Conceptuel de Communication) qui définit les flux d'informations à prendre en compte. L'étape suivante consiste à mettre au point le «MCD» (Modèle Conceptuel de Données) et le «MCT» (Modèle Conceptuel des Traitements) décrivant les règles et les contraintes à prendre en compte. Le modèle organisationnel consiste à définir le «MOT» (Modèle Organisationnel des Traitements) décrivant les contraintes dues à l'environnement (organisationnel, spatial et temporel).

2. UML

UML (Unified Modeling Language), que l'on peut traduire par «langage de modélisation unifié» est une notation permettant de modéliser un problème de façon «standard». Ce

langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

Les méthodes de modélisation orientée objets

La modélisation objet consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse, sans se préoccuper de l'implémentation. Il s'agit donc de déterminer les objets présents et d'isoler leurs données et les fonctions qui les utilisent. Pour cela, des méthodes ont été mises au point. Entre 1970 et 1990, de nombreux analystes ont mis au point des approches orientées objets, si bien qu'en 1994 il existait plus de 50 méthodes objet. Toutefois, seules 3 méthodes ont véritablement émergé :

- La méthode OMT de Rumbaugh
- La méthode BOOCH de Booch
- La méthode OOSE de Jacobson (Object Oriented Software Engineering).

A partir de 1994, Rumbaugh et Booch (rejoints en 1995 par Jacobson) ont unis leurs efforts pour mettre au point la méthode unifiée (unified method), incorporant les avantages de chacune des méthodes précédentes.

La méthode unifiée à partir de la version 1.0 devient UML (Unified Modeling Language), une notation universelle pour la modélisation objet.

UML 1.0 est soumise à l'OMG (Object Management Group) en janvier 1997, mais elle ne sera acceptée qu'en novembre 1997 dans sa version 1.1, date à partir de laquelle UML devient un standard international.

Les outils de modélisation UML

UML offre une panoplie d'outils de modélisation utilisés librement par l'utilisateur pour répondre à ses besoins sans être autant obligé à respecter toutes les subtilités de l'outil. Au contraire, une méthode telle que MERISE exige une application complète des formalismes fournis.

Voilà quelques outils de modélisation qu'offre UML

Diagramme de classes: Plus on ajoute de classe au modèle, moins leur représentation textuelle est suffisante. C'est la raison pour laquelle on fait appel à des diagrammes de classes pour fournir une vue de tout ou une partie des classes du modèle. Le diagramme de classes principal de la vue logique du modèle est le plus souvent une image des paquetages du système. Chaque paquetage possède aussi son diagramme principal de classes qui, normalement, présente les classes publiques du paquetage.

Diagramme de cas d'utilisation : C'est une vue graphique de tout ou une partie des acteurs d'un système, de ses cas d'utilisation et de leurs interactions. Chaque système possède normalement un diagramme de cas d'utilisation principal (Main), qui précise les frontières

du système (Acteurs) et celles des fonctionnalités principales fournies par le système (Cas d'utilisation).

Diagramme de séquences : il représente les interactions entre des objets dans un enchaînement temporel. Il montre les objets et les classes impliquées dans un scénario, ainsi que la succession des messages échangés entre les objets pour réaliser la fonctionnalité du scénario. Dans la vue logique du système en cours de développement, les diagrammes de séquences sont couramment associés à la réalisation des cas d'utilisation.

B. Comparaison

Merise et UML sont deux grands principes de « traduction » ou modélisation d'un système d'information. Néanmoins, ils ne sont pas aussi proches qu'on pourrait le penser.

Le choix de l'un ou de l'autre se fait selon trois axes à savoir l'accessibilité, la précision et l'exploitabilité.

Pour le premier axe (**accessibilité**) MERISE présente l'intérêt d'avoir des modèles logiques moins détaillés facilement compréhensibles.

Tandis qu'UML conçu pour s'adapter à n'importe quel langage de programmation orientée objet (POO), présente plusieurs modèles (diagrammes) dont leurs compréhensions nécessitent une grande attention.

En ce qui concerne le deuxième critère (**précision**), MERISE est moins préférable. Malgré sa clarté, il manque une précision du fait qu'elle est éloignée du langage donc difficile à implémenter alors qu'UML intègre les éléments communs des différents langages, sa volonté est d'être fidèle à la réalisation finale. Elle est beaucoup plus complète avec ses différents diagrammes.

Pour en finir avec l'**exploitabilité**, MERISE est une méthode plus généraliste. Elle donne une vue globale de la solution sans autant entrer dans les petits détails. Contrairement à UML qui est conçu pour l'implémentation objet avec ses différents détails et sa portabilité (s'adapte à n'importe quelle plateforme) elle est donc plus exploitable.

L'une ou l'autre présente des avantages et des inconvénients. Il est réservé au concepteur de choisir la méthode la mieux adaptée pour son cas. Si on cherche la précision et l'exploitabilité UML devance MERISE. Tandis que, si c'est la clarté et l'accessibilité qui sont en question MERISE est préférable.

C. Choix de la méthode

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer.

Notre application gère des données plus complexes d'où nous avons choisi UML qui est la mieux recommandée, vu qu'elle rend la modélisation plus simple à implémenter.

DEUXIEME PARTIE

ANALYSE ET CONCEPTION

CHAPITRE 1 : ANALYSE

I. Prérequis théorique

A.Approche Fonctionnelle

Ce travail concerne dans l'ordre l'analyse et la définition des besoins de tests pour la vérification fonctionnelle durant les étapes de maintenance, mise au point en validation de prototype et en intégration série, conception du système. Pour chacun des besoins recensés, on propose une méthodologie de conception et de réalisation. L'analyse fait apparaître la nécessité d'un environnement propre à l'application de ces tests. On l'étudiera particulièrement pour le cas des tests des sous-systèmes périphériques et des tests de simultanéité au niveau système.

Exemple de méthode fonctionnelle : MERISE

B. Approche Orientée Objet

L'approche orientée objet considère le logiciel comme une collection d'objets, leurs traitements dissociés et associés définies par des propriétés. Une propriété est attribut dont la valeur peut intervenir sur l'aspect comportemental de l'objet. La fonctionnalité du logiciel émerge alors que l'interaction entre les différents objets qui le constituent. L'une des particularités de cette approche est qu'elle rapproche les données et les traitements associés au sein d'un unique objet. La modélisation orientée objet est basée essentiellement sur trois concepts dont :

- L'encapsulation
- L'héritage
- Le polymorphisme

Exemple de modélisation orientée objet : UML

C.Concepts et Formalisme de la méthode choisie

Connaître les concepts des approches orientées objets; concevoir le développement de systèmes informatiques dans une optique orientée objets; comparer des méthodes et outils orientés objets afin de pouvoir choisir ceux qui conviennent le mieux dans le développement de systèmes particuliers.

Concepts de base : objet, acteur, agent, classe, message, héritage, délégation, instanciation, spécialisation, différenciation, classe abstraite, polymorphisme, persistance. Méthodes d'analyse et de conception orientées objets. Langages orientés objets, tels que Smalltalk, C++, Eiffel, CLOS, ABCL. Outils pour le développement orienté objets.

II. Analyse de l'existant

A.Recensement et Analyse Théorique

Après avoir détecté les anomalies de la procédure, une approche de solution consiste à concevoir et développer une application qui facilitera les insuffisances et les défaillances.

B. Descriptions des procédures

Lorsqu' un client se présente à la réception de l'hôtel pour faire sa réservation il exprime sa requête selon ses besoins. Pour pouvoir répondre au client la réceptionniste consulte son planning pour voir la disponibilité des chambres selon la description du client. Si la demande est favorable la réceptionniste lui communique le prix et les différents arrangements possibles. Suite à la décision favorable du client, la réceptionniste contacte la gouvernante pour s'enquérir de l'état de la chambre, inscrit le nom du client sur la liste d'arrivés du jour ainsi que la date et l'heure d'arrivé et la date de départ .Après l'avoir invité à remplir la fiche de police, la clef lui est remise tout en le mettant en rapport avec un bagagiste qui le conduira à sa chambre. C'est à partir de la fiche de police que la réceptionniste se base pour pouvoir remplir le registre de police, qu'on remettra pour d'éventuel control.

La réceptionniste établit deux facture préforma pour l'hébergement dont l'originale sera remis au client et servira au moment du paiement et l'autre sera gardé dans les archives de l'hotel.

Un acompte est obligatoire pour tout séjour.

A la fin de la journée la réceptionniste établit la situation global de l'hébergement (nombre de client reçu, les départs du jour, les clients no show (Pa venu)).

C.Le Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

Ils permettent de décrire l'interaction entre l'acteur et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système ! Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Les *use case* (cas d'utilisation) sont représentés par une ellipse sous-titrée par le nom du cas d'utilisation (éventuellement le nom est placé dans l'ellipse). Un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne.

NOTE : le plus souvent, le diagramme des cas d'utilisation est établi par la maîtrise d'ouvrage (MOA) d'un projet lors de la rédaction du cahier des charges afin de transmettre les besoins des utilisateurs et les fonctionnalités attendues associées à la maîtrise d'œuvre (MOE).

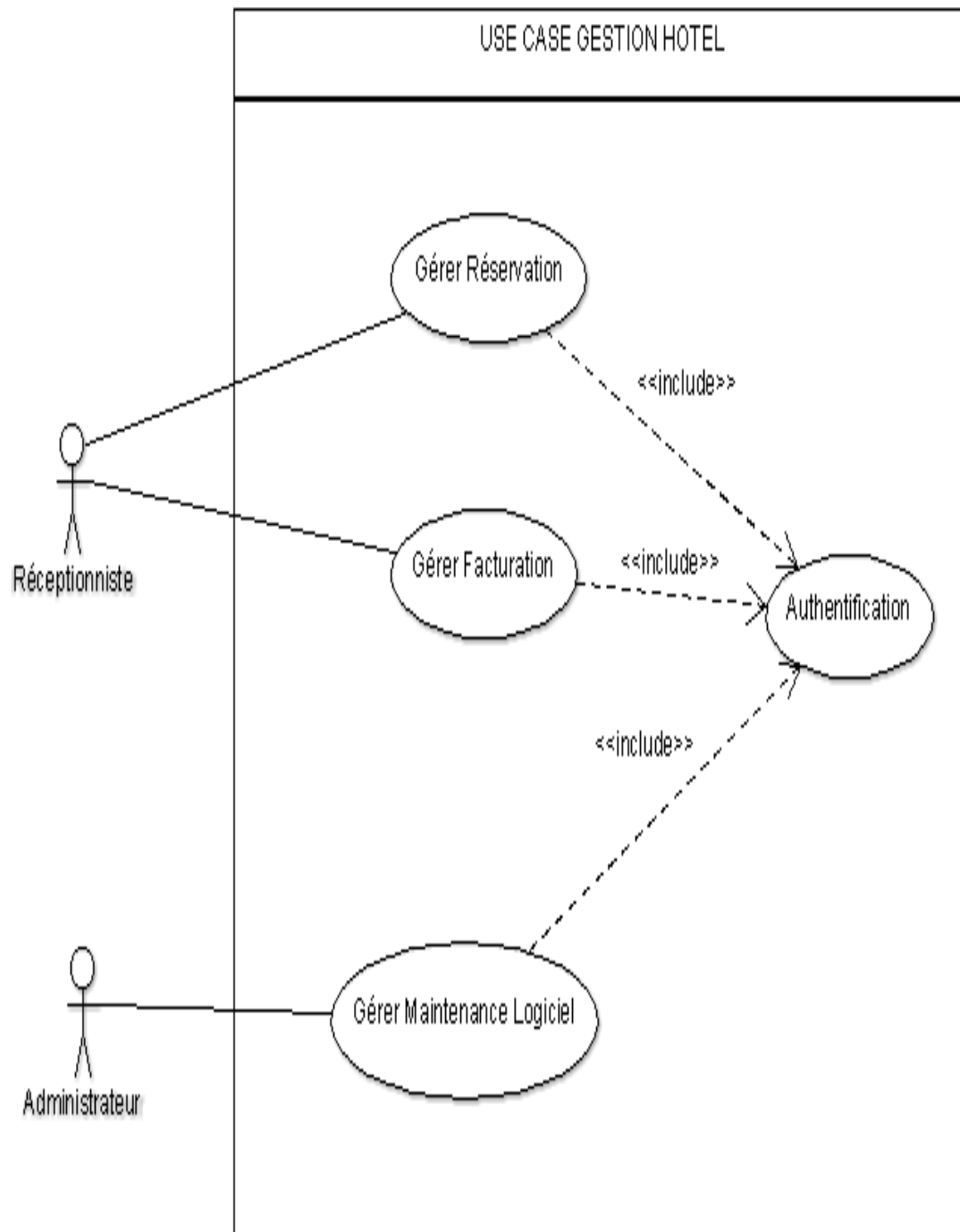


Figure 2 Use Case Générale

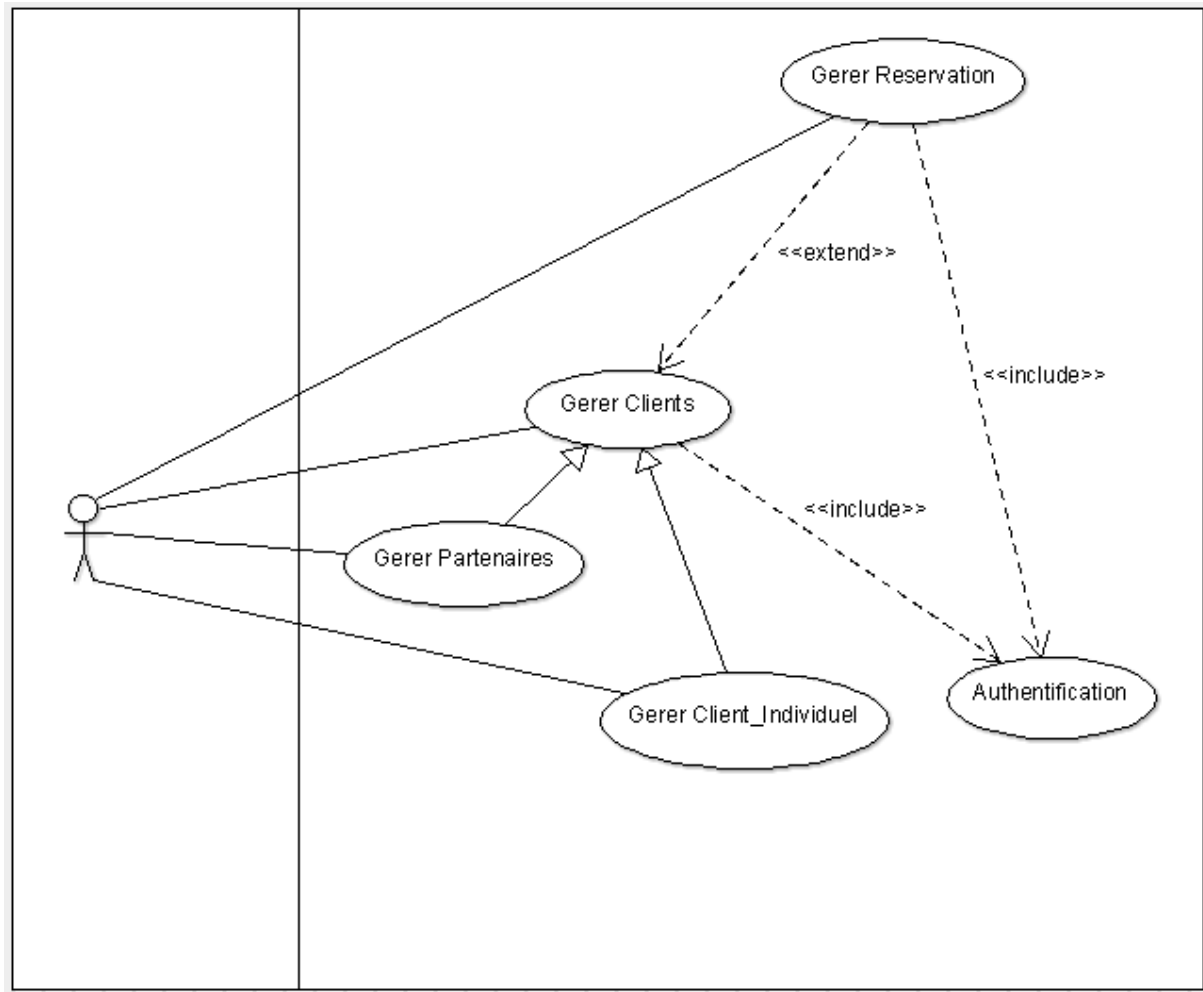


Figure 3 Use Case Réserveation

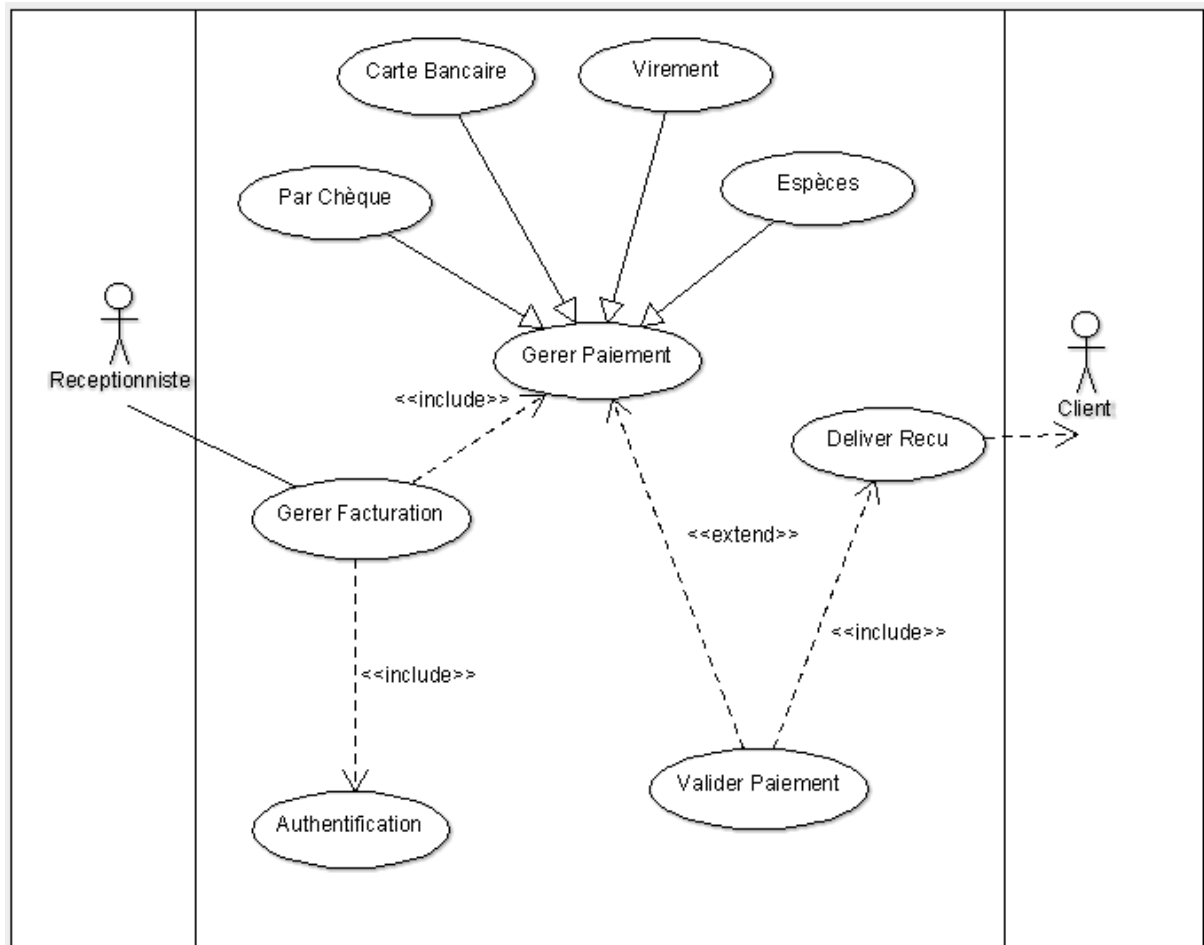


Figure 4 Use Case Facturation

CHAPITRE 2 : CONCEPTION

I. Le Diagramme de Séquence Système

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets.

Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages.

II. Le Diagramme de Classe au Niveau Analyse

Un diagramme de classes UML décrit les structures des objets et des informations utilisés par votre application, à la fois en interne et en communication avec ses utilisateurs. Il décrit les informations sans référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de différentes manières, par exemple avec des tables de bases de données, des nœuds XML ou des compositions d'objets logiciels.

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

III. Conception Détaillée

A. Diagramme de Séquence Détaillée

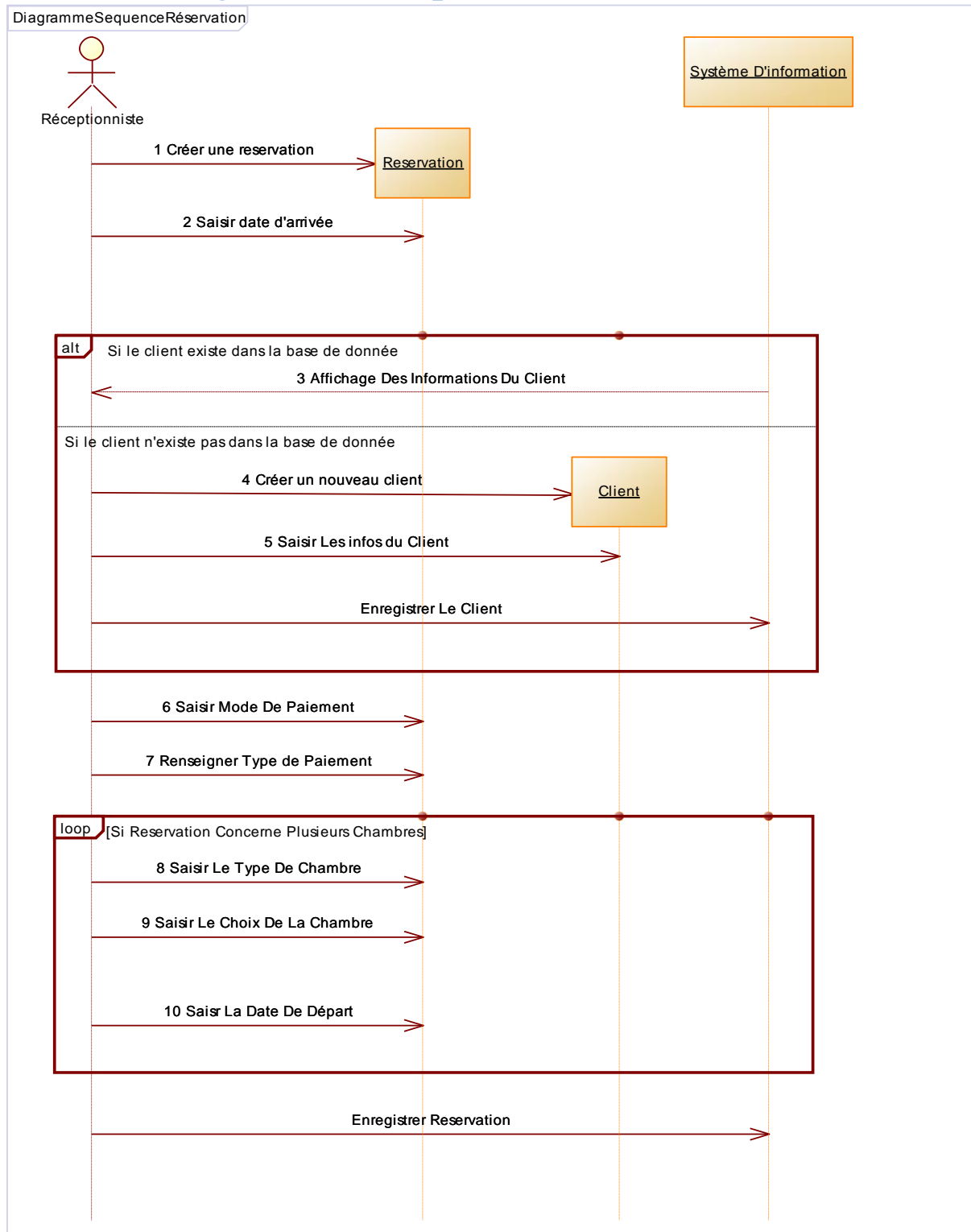


Figure 5 Diagramme de Séquence Réservation

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

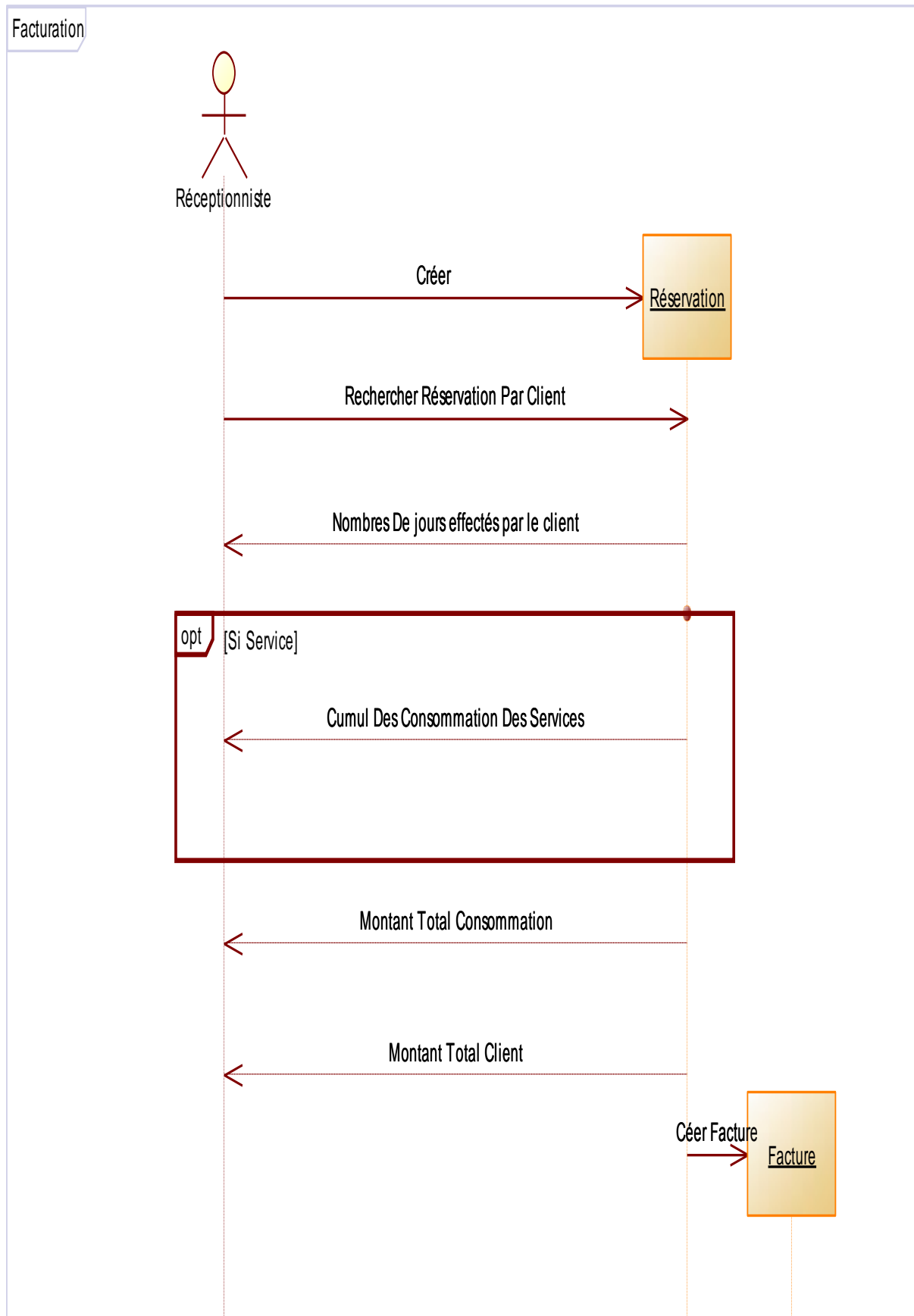


Figure 6 Diagramme de Séquence Facturation

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

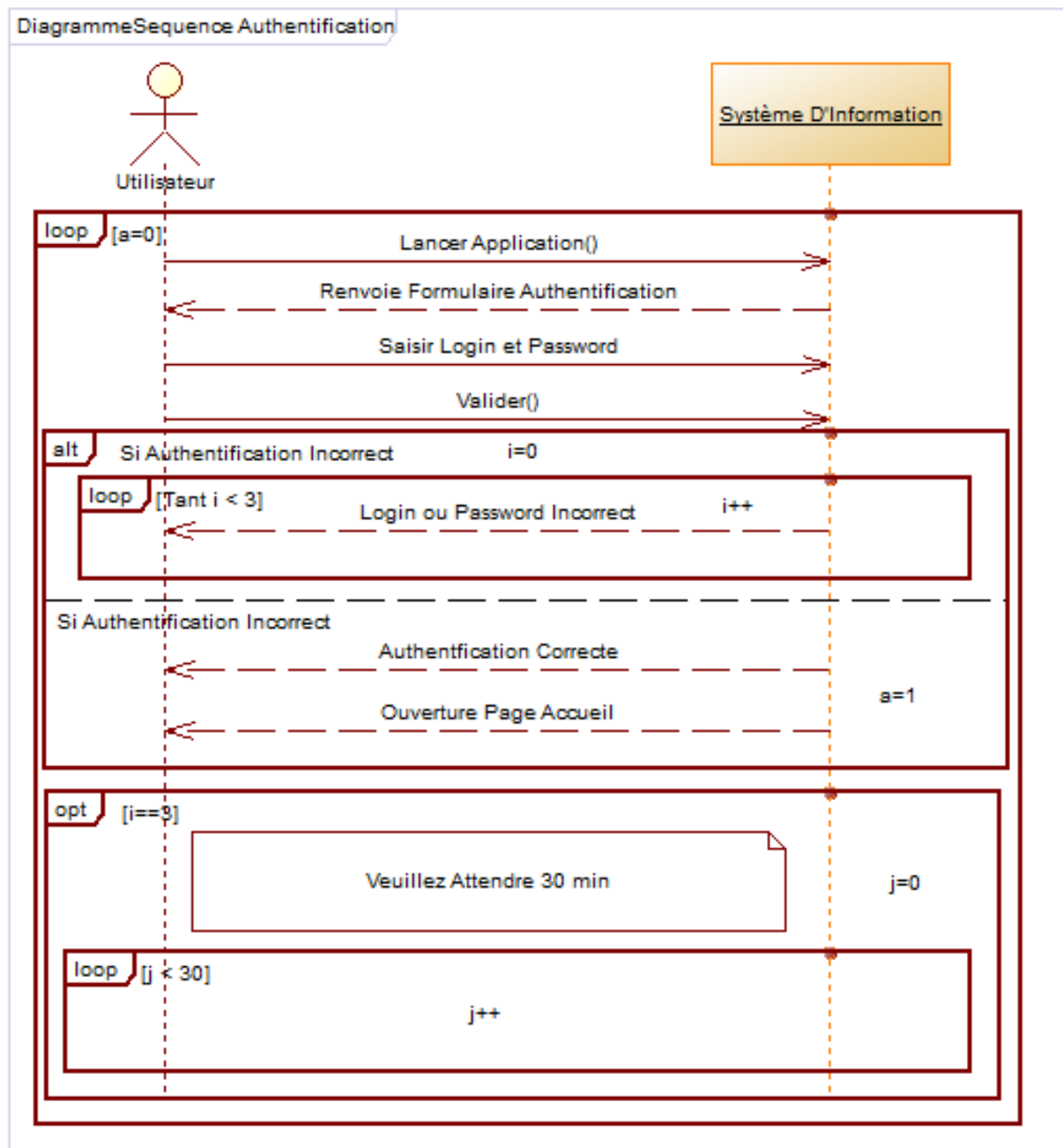
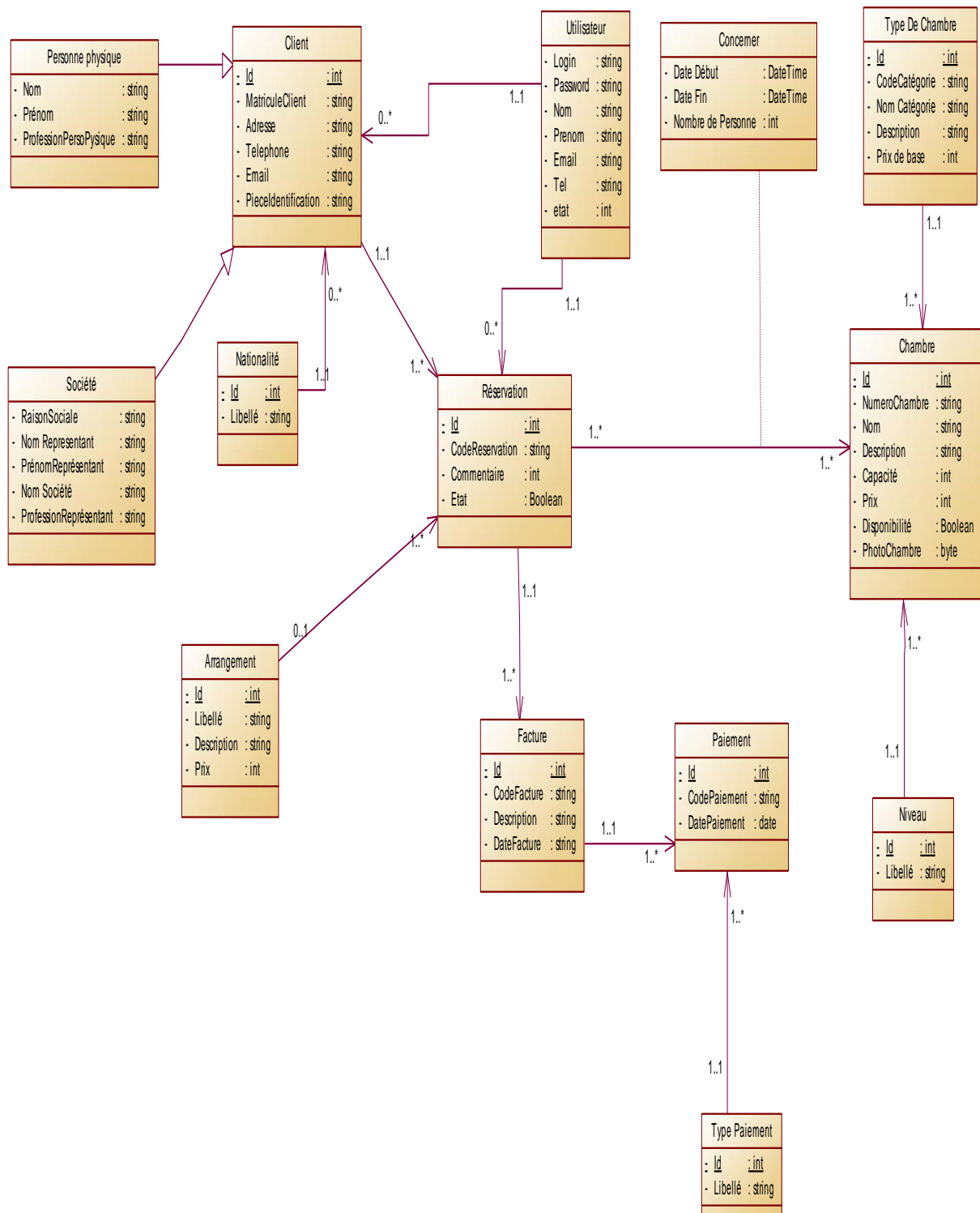


Figure 7 Diagramme de Séquence Authentification

B. Diagramme de Classe niveau Conception



TROISEME PARTIE IMPLEMENTATION ET DEPLOIEMENT

CHAPITRE : 1

IMPLEMENTATION

I. Environnement de développement

En programmation informatique, un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier.

Dans un environnement de développement « intégré » (abrégié EDI en français ou IDE en anglais, pour Integrated Development Environment) les outils sont prévus pour être utilisés ensemble (le produit d'un outil peut servir de matière première pour un autre outil). Les outils peuvent être intégrés dès le départ, c'est-à-dire qu'ils sont construits dans le but d'être utilisés ensemble. Il peut aussi s'agir d'un ensemble d'outils développés sans lien entre eux et intégrés à posteriori.

L'objectif d'un environnement de développement est d'augmenter la productivité des programmeurs en automatisant une partie des activités et en simplifiant les opérations. Les environnements de développement visent également à améliorer la qualité de la documentation en rapport avec le logiciel en construction. Certains environnements de développement offrent également la possibilité de créer des prototypes, de planifier les travaux et de gérer des projets.

Exemple D'environnement de développement

- Eclipse
- NetBeans
- Visual Studio
- Android Studio
- PhpStorm
- Bloc Note
- Sublime Text
- Nod Pad++

II. Etude de quelque langage de programmation

A. Définition

Un langage de programmation est un langage permettant de formuler des algorithmes et de produire des programmes informatiques qui s'appliquent à ces algorithmes.

B. Exemple de langages de programmation

1. Le langage C

Le langage C a été inventé au cours de l'année 1972 dans les Laboratoires Bell. Il était développé en même temps qu'UNIX par Dennis Ritchie et Ken Thompson. Ken Thompson avait développé un prédécesseur de C, le langage B, qui est lui-même inspiré de BCPL. Dennis Ritchie a fait évoluer le langage B dans une nouvelle version suffisamment différente, en ajoutant notamment les types, pour qu'elle soit appelée C.

Le langage C reste un des langages les plus utilisés actuellement. Cela est dû au fait que le langage C est un langage qui comporte des instructions et des structures de haut niveau (contrairement à l'assembleur par exemple) tout en générant un code très rapide grâce à un compilateur très performant.

Un des principaux intérêts du C est que c'est un langage très portable. Un programme écrit en C en respectant la norme ANSI est portable sans modifications sur n'importe quel système d'exploitation disposant d'un compilateur C : Windows, UNIX, VMS (système des VAX) ou encore OS/390 ou z/Os (l'OS des mainframes IBM).

La rapidité des programmes écrits en C est en grande partie due au fait que le compilateur présuppose que le programmeur sait ce qu'il fait : il génère un code ne contenant pas de vérifications sur la validité des pointeurs, l'espace d'adressage, etc.

Ainsi, les programmes en C sont très compacts.

De plus, une des caractéristiques du C est qu'il est un langage « faiblement typé » : les types de données qu'il manipule sont très restreints, et proches de la représentation interne par le processeur : par exemple, le type 'Chaîne de caractères' n'existe pas en C. A l'inverse, comparer un entier et un caractère a un sens en C car un caractère est bien représenté en interne par le processeur par une valeur de type entier (le code ASCII ou le code EBCDIC).

Enfin et pour conclure, il est inexact que le C est un langage difficile à apprendre ! Au contraire : le C dispose de peu d'instructions, les structures de données sont limitées, etc.

2. Le langage C++

C++ est un langage de programmation compilé, permettant la programmation sous de multiples paradigmes comme la programmation, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. C++ est l'un des langages de programmation les plus populaires, avec une grande variété de plates-formes matérielles et de systèmes d'exploitation.

En langage C, ++ est l'opérateur d'incrément, c'est-à-dire l'augmentation de la valeur d'une variable de 1. C'est pourquoi C++ porte ce nom : cela signifie que C++ est un niveau

au-dessus de C. Il existe de nombreuses bibliothèques C++ en plus de la bibliothèque (C++ *Standard Library*) qui est incluse dans la norme. Par ailleurs, C++ permet l'utilisation de l'ensemble des bibliothèques C existantes.

3. Le langage Java

Java est à la fois un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Java permet de développer des applications autonomes mais aussi, et surtout, des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Java Runtime Environment - Environnement d'exécution Java). Cette dernière est constituée d'une JVM (Java Virtual Machine - Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

4. Le langage C#

C# est un langage de programmation orientée objet, fortement typé, dérivé de C et C++, ressemblant au langage Java. Il est utilisé pour développer des applications web, ainsi que des applications de bureau, des services web, des commandes, des widgets ou des bibliothèques de classes. En C# une application est un lot de classes où une des classes comporte une méthode *Main*, comme cela se fait en Java.

Bien que le C# soit similaire à Java, il existe des différences notables, par exemple :

Java n'autorise pas la surcharge des opérateurs ;

Java a des exceptions vérifiées, alors que les exceptions du C# ne sont pas vérifiées, comme en C++ ;

Java permet la génération automatique de la documentation HTML à partir des fichiers sources à l'aide des descriptions Javadoc-syntax, tandis que le C# utilise des descriptions basées sur le XML ;

Java n'a pas de langage préprocesseur ;

C# supporte les indexeurs, les méthodes déléguées, et les événements (là où Java se contente du patron de conception Observateur) ;

C# ne supporte pas les implémentations anonymes d'interfaces et de classes abstraites ;

C# ne supporte que les classes internes statiques ;

C# supporte les structures en plus des classes (les structures sont des types *valeur* : on stocke le contenu et non l'adresse) ;

C# utilise une syntaxe intégrée au langage (DllImport) et portable pour appeler une bibliothèque native, tandis que Java utilise *Java Native Interface* ;

C# supporte la généricité, et la machine .NET a été modifiée pour permettre cela (Java la supporte également, mais son implémentation a été réalisée dans le compilateur javac sans changer le byte code Java).

5. Le langage PHP

Le langage PHP a été inventé par Rasmus LERDORF en 1995 pour son usage personnel. Autrefois abréviation de Personal HomePage devenue aujourd'hui Hypertext Preprocessor, PHP s'impose comme un standard dans le monde de la programmation web par ses performances, sa fiabilité, sa souplesse et sa rapidité.

PHP a été inventé à l'origine pour le développement d'applications web dynamiques qui constituent encore le cas d'utilisation le plus courant et son point fort. Cependant, les évolutions qui lui ont été apportées jusqu'à aujourd'hui assurent à PHP une polyvalence non négligeable. PHP est par exemple capable d'interagir avec Java, de générer des fichiers PDF, d'exécuter des commandes Shell, de gérer des objets (au sens programmation orientée objet), de créer des images ou bien de fournir des interfaces graphiques au moyen de PHP GTK.

C.Choix du langage

La description des langages de programmation faites ci-dessus nous permettent de choisir C# comme langage d'implémentation de notre application. Avec ses multiples avantages en comparaison autres langages qui peuvent servir un début pour l'apprentissage des langages de Programmation Orientés Objets (POO), C# s'impose.

III. Etudes Comparatives de quelques SGBD

A. Présentation

En informatique, un système de gestion de base de données (abr. *SGBD*) est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un SGBD (en anglais *DBMS* pour *database management system*) permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il permet d'effectuer des comptes rendus des informations enregistrées et comporte des mécanismes pour assurer la cohérence des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation par d'autres logiciels. Selon le modèle, le SGBD peut comporter une simple interface graphique jusqu'à des langages de programmation sophistiqués.

Les systèmes de gestion de base de données sont des logiciels universels, indépendants de l'usage qui est fait des bases de données. Ils sont utilisés pour de nombreuses applications informatiques, notamment les guichets automatiques bancaires, les logiciels de réservation, les bibliothèques numériques, les logiciels d'inventaire, les progiciels de gestion intégrés ou la plupart des blogs et sites web. Il existe de nombreux systèmes de gestion de base de données. Les SGBD sont souvent utilisés par d'autres logiciels ainsi que les administrateurs ou les développeurs. Ils peuvent être sous forme de composant logiciel, de serveur, de logiciel applicatif ou d'environnement de programmation.

B. Exemple de SGBD

1. MySQL

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur.

C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Un logiciel qui intègre du code MySQL ou intègre MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Cependant, si la base de données est séparée du logiciel propriétaire qui ne fait qu'utiliser des API tierces (par exemple en C# ou PHP), alors il n'y a pas besoin d'acquérir une licence payante MySQL.

2. SQL Server

Le SQL server désigne couramment un serveur de base de données. La définition du SQL server est étroitement liée à celle du langage SQL (Structured Query Language), un langage informatique permettant d'exploiter des bases de données.

Concrètement, un SQL server est un outil qui possède toutes les caractéristiques pour pouvoir accompagner l'utilisateur dans la manipulation, le contrôle, le tri, la mise à jour, et bien d'autres actions encore, de bases de données grâce au langage SQL.

Le terme désigne également le nom donné au système de gestion de base de données (SGBD) commercialisé par Microsoft, ou plus précisément le nom du moteur de bases de données de ce SGDB produit par le fabricant de produits informatiques américain.

3. Oracle

Oracle est un SGBD (système de gestion de bases de données) édité par la société du même nom Oracle Corporation (<http://www.oracle.com>), leader mondial des bases de données.

La société *Oracle Corporation* a été créée en 1977 par Lawrence Ellison, Bob Miner, et Ed Oates. Elle s'appelle alors **Relational Software Incorporated (RSI)** et commercialise un Système de Gestion de Bases de données relationnelles (SGBDR ou RDBMS pour **Relational Database Management System**) nommé *Oracle*.

Oracle se décline en plusieurs versions

- Oracle Server Standard, une version comprenant les outils les plus courants de la solution Oracle. Il ne s'agit pas pour autant d'une version bridée...
- Oracle Server Enterprise Edition

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents

4. Microsoft Access

MS Access est un logiciel utilisant des fichiers au format Access (extension de fichier *.mdb*) pour Microsoft DataBase (extension **.accdb* depuis la version 2007). Il est compatible avec les requêtes SQL (sous certaines restrictions) et dispose d'une interface graphique pour saisir les requêtes (QBE - Query by Example - « Requête par l'exemple »). Il permet aussi de configurer, avec des assistants ou librement, des formulaires et sous-formulaires de saisie, des états imprimables (avec regroupements de données selon divers critères et des totalisations, sous-totalisations, conditionnelles ou non), des pages html liées aux données d'une base, des macros et des modules VBA.

Comme beaucoup de systèmes de gestion de bases de données relationnelles, ses données peuvent être utilisées dans des programmes écrits dans divers langages.

La dernière version en date est la version 2016 ; elle fait partie de la suite Microsoft Office 2016 et est incluse dans l'abonnement à Office 365. Access 2016 intègre de nouvelles fonctionnalités dont de nouveaux thèmes, la modernisation des cinq modèles les plus populaires et l'exportation d'informations de sources de données liées vers Excel.

C.Choix du SGBD

Vu la multidisciplinarité et sa domination croissante, plusieurs Systèmes de Gestion Base de données ne cessent de voir le jour. On peut rencontrer pas mal de Systèmes de Gestion Base de données, certains sont en open source et d'autres commerciaux. Pour notre projet nous avons choisi SQL Server en particulier SQL SERVER 2014.

IV. Architecture Applicative

A. Définition

En informatique, une architecture applicative désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et/ou matériels et/ou humains et/ou informations) et des relations entre les éléments. Cette structure fait suite à un ensemble de décisions stratégiques prises durant la conception de tout ou partie du système informatique, par l'exercice d'une discipline technique et industrielle du secteur de l'informatique dénommée elle aussi architecture, et dont le responsable est l'architecte informatique. Les différentes architectures existantes suivantes :

- Architecture 1-Tier
- Architecture 2-Tier
- Architecture 3-Tier
- Architecture N-Tiers

B. Exemple D'Architecture

1. Présentation de l'architecture 1-Tier

Dans une architecture 1-tiers, l'API peut être utilisée soit en monoposte ou sur un serveur de fichiers ou sur un système central. Autrement dit, la couche présentation la couche traitement et la couche données s'exécute sur le même serveur. L'avantage est qu'il y'a une simplicité d'administration et une centralisation des données.

Inconvénients

- Interface simple
- Un risque élevé de monter en charge

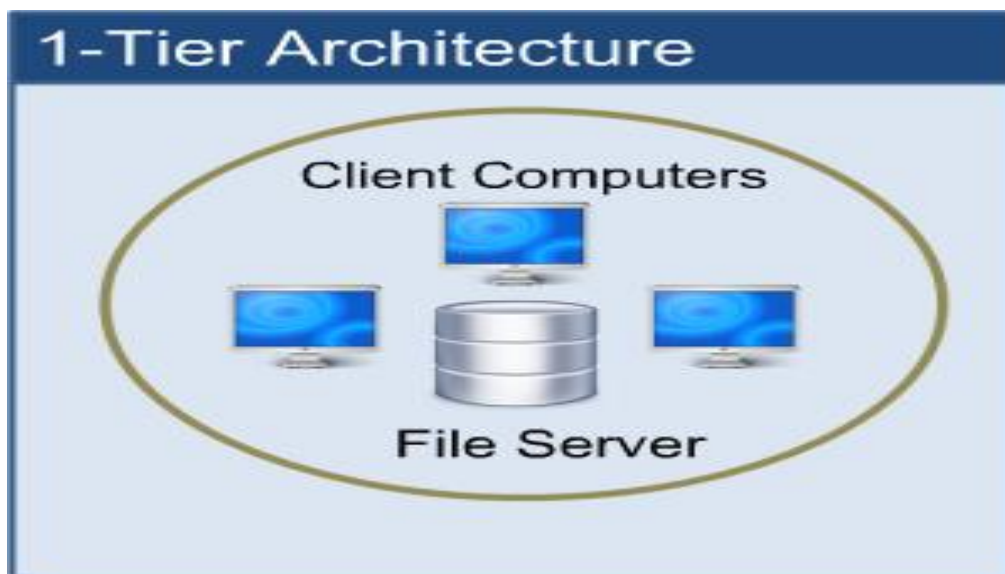


Figure 8 Architecture 1-Tier

2. Présentation de l'architecture 2-Tier

Elle est encore appelée client server de première génération et repose sur l'utilisation de moteur de base de données relationnels. Ce moteur permet de distribuer la gestion de la persistance sur un serveur. Dans les 2 éléments principaux on distingue :

Le client qui reçoit la présentation et l'interface utilisateur.

Le serveur qui reçoit la partie persistance et la gestion physique des données.

Les services métiers peuvent être soit entièrement du côté client intégré à la présentation et le serveur ne gère que les données, soit entièrement serveur et le client ne gère que l'interface utilisateur.

Le plus de cette architecture est :

L'interface est plus riche

La possibilité d'installer l'API sur le poste-client qui devient une poste intelligent.

Le moins de cette architecture est :

Le client est très sollicité et peut être lourd.

Un important dialogue se fait entre le serveur et le client.

Une relation étroite entre le programme client et l'organisation côté serveur qui entraîne une difficulté d'évolution.

Pour corriger ce défaut il faut donner la possibilité d'alléger le client et rendre le système plus souple.

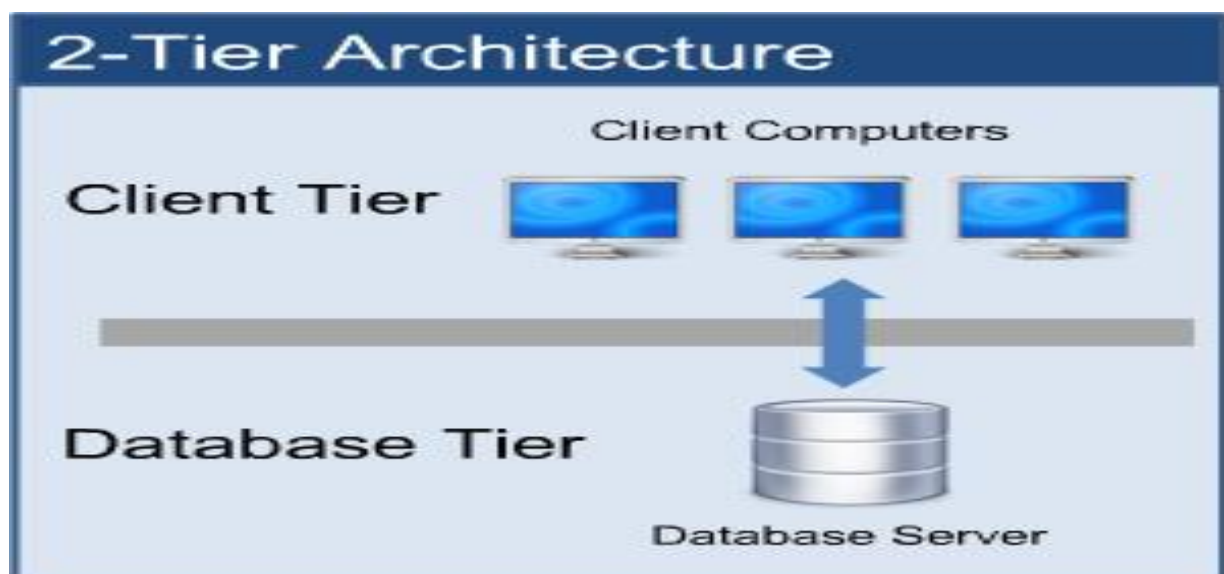


Figure 9 Architecture 2-Tier

3. Présentation de l'architecture 3-Tier

Dans cette architecture les trois (3) principaux tiers exécutent chacun sur machine différente

La présentation sur la machine cliente

L'application métier sur les serveurs d'API

La persistance sur le serveur de base de données

Les principes de cette architecture sont que :

Le client gère la présentation et les traitements locaux

Le serveur d'API gère les traitements

Le serveur de données accueille le SGBD

Ces caractéristiques sont :

Les trois niveaux sont indépendantes ce qui facilite les évolutions.

La répartition des taches devient de plus en plus homogène

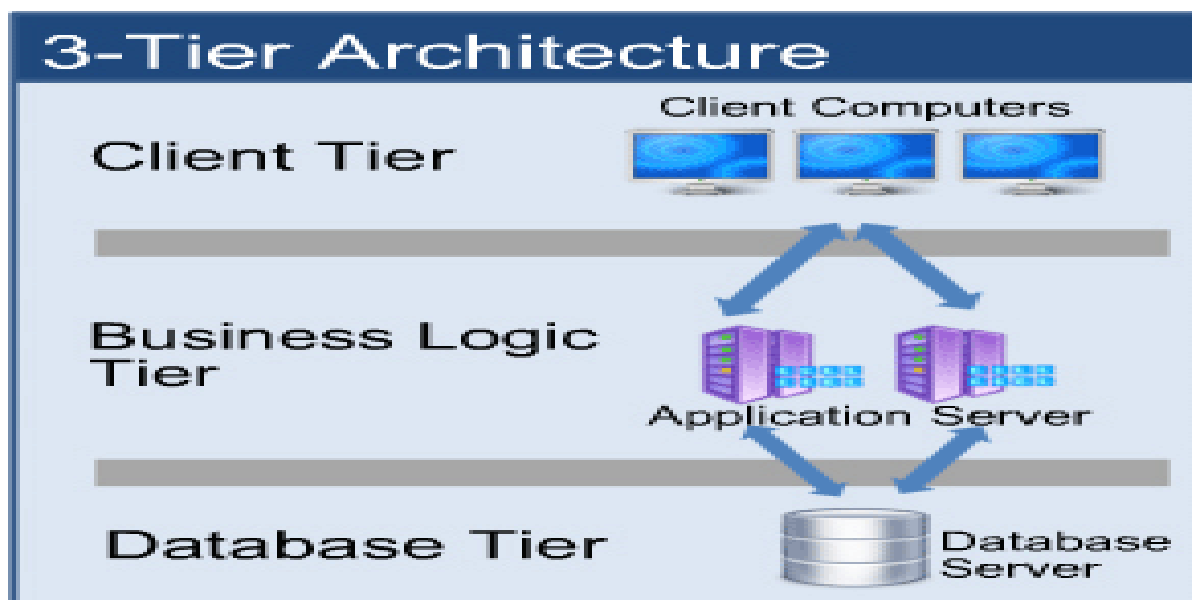


Figure 10 Architecture 3-Tier

4. Présentation de l'architecture N-Tiers

C'est une architecture qui permet de rajouter des couches en plus (3-tiers évolué). Plusieurs serveurs d'API peuvent être mise en œuvre, ce qui facilite l'évolution et la montée en charge. Seulement elle est très complexe et son coût de mise en œuvre très onéreux. L'autre inconvénient cette architecture, ces services s'appliquent sur des technologies très variées nécessitant de gérer l'hétérogénéité et l'interopérabilité.

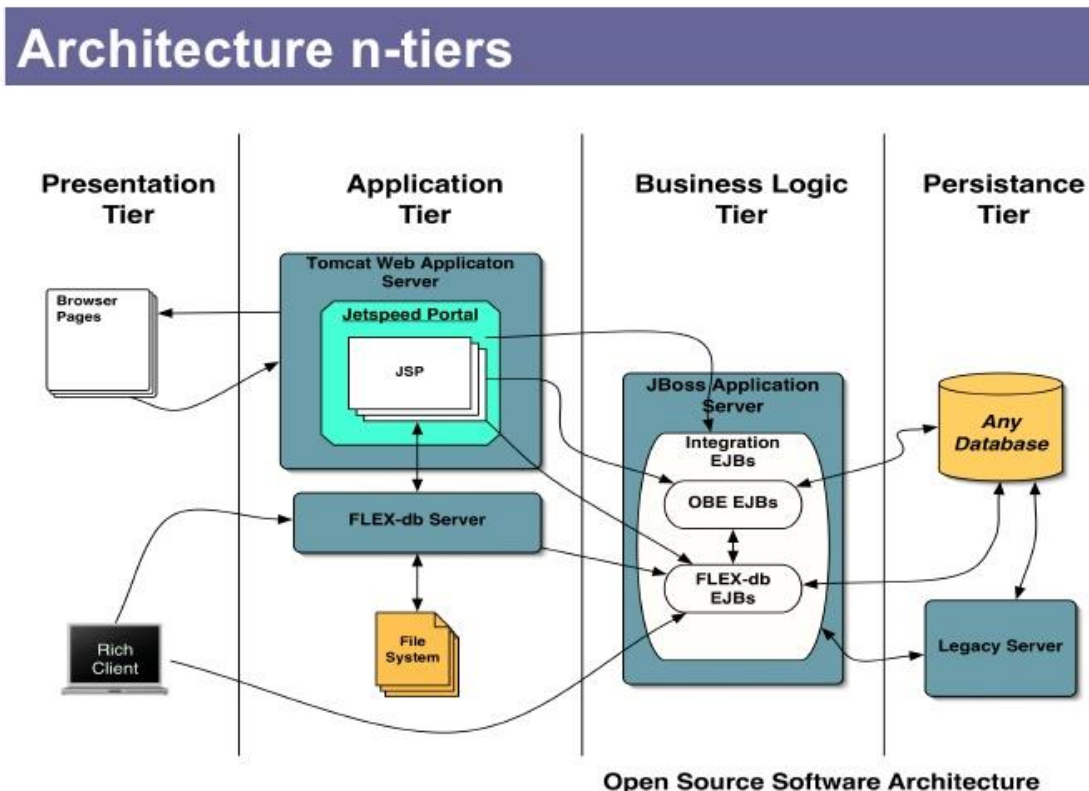


Figure 11 Architecture n-Tiers

C.Choix de l'architecture

Après l'étude réalisée des différents types d'architectures, notre choix s'est porté sur l'architecture 1-Tiers du fait de sa flexibilité, sa robustesse, sa facilité à gérer correctement les problèmes de sécurité. L'objectif premier d'un système d'information quel qu'il soit est de permettre à l'utilisateur d'accéder aux informations.

V. MVC

Modèle d'architecture qui cherche à séparer nettement les couches de présentation (UI : User Interface), métier (BLL : Business Logic Layer) et d'accès aux données (DAL : Data Access Layer). Le but étant d'avoir une dépendance minimale entre les différentes couches de l'application ; ainsi les modifications effectuées sur n'importe quelle couche de l'application n'affectent pas les autres couches.

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

- Modèle – Encapsule le cœur fonctionnel de l'application, le domaine logique.
 - Vue – les données sont envoyées, par le modèle, à la vue qui les présente à l'utilisateur.
 - Contrôleur – reçoit les données et les transmet au modèle ou à la vue.
- Une telle architecture est communément appelée architecture 3-tier ou à 3 niveaux.

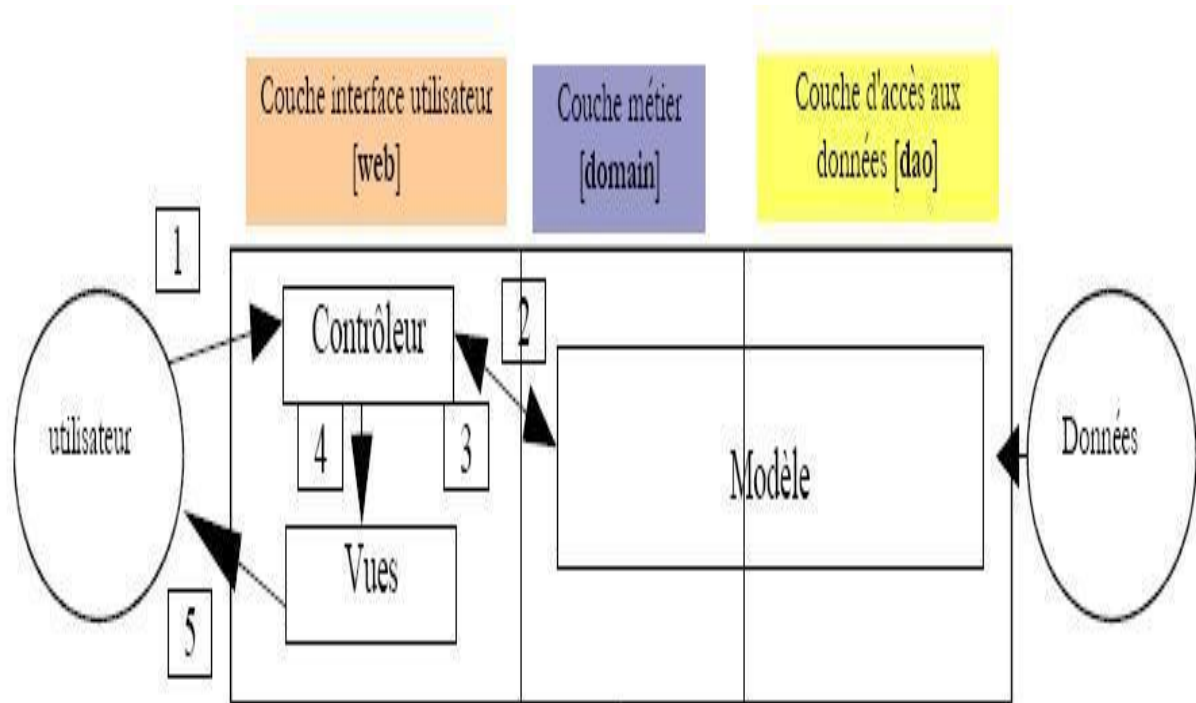


Figure 12 MVC

1. Le client fait une demande au contrôleur. Ce contrôleur voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le C de MVC.
2. Le contrôleur traite cette demande. Pour ce faire, il peut avoir besoin de l'aide de la couche métier, ce qu'on appelle le modèle M dans la structure MVC.
3. Le contrôleur reçoit une réponse de la couche métier. La demande du client a été traitée. Celle-ci peut appeler plusieurs réponses possibles. Un exemple classique est :
 - une page d'erreurs si la demande n'a pu être traitée correctement
 - une page de confirmation sinon
4. Le contrôleur choisit la réponse (= vue) à envoyer au client. Celle-ci est le plus souvent une page contenant des éléments dynamiques. Le contrôleur fournit ceux-ci à la vue.
5. la vue est envoyée au client. C'est le V de MVC.

Définition inspirée des tutoriels de Serge TAHE.

CHAPITRE : 2

DEPLOIEMENT

I. Présentation de l'application

L'application est un logiciel de gestion hôtelière pour votre hôtel, camping, gîte, chambre d'hôte...

Cet outil simple et intuitif, vous permet de gérer vos réservations, clients, chambres, catégories de chambres, factures, arrivées, départs...

L'interface ergonomique de celui-ci reste unique pour sa simplicité.

En effet, un menu permanent avec les principales fonctions reste toujours à votre disposition. Des icônes représentent les actions ou les fonctions de chaque élément. Un planning contenant toutes les réservations permettra d'avoir un aperçu en temps réel de votre hôtel. Vous pourrez modifier très simplement vos réservations par le simple clic, le redimensionnement et les accès rapides.

De même, l'application s'adapte à votre façon de travailler car il gère les réservations par chambres et par types de chambres (catégories). Le paramétrage du logiciel vous permet de travailler exactement comme vous en avez.

De nombreux utilitaires intégrés au logiciel vous permettront un contrôle total de votre hôtel : historique de facturation, sauvegarde et restauration, tableau de bord avec statistiques...

L'application reste la solution professionnelle à adopter. Le caractère intuitif et simple d'utilisation, conviendra au plus grand nombre. Les paramètres sont personnalisables par vos soins sans grande compétence informatique. Utilisable avec une souris ou un écran tactile, la performance et la puissance de l'application vous donneront satisfaction et vous confortera dans votre choix professionnel. Afin de répondre parfaitement à des besoins variés, l'application est un logiciel hôtelier modulable et évolutif.



The image shows a web browser window with a green title bar containing the text 'Connexion' and standard window controls. The main content area has a teal background. At the top left, the word 'Authentification' is written in a large, white, stylized font. Below it, there are three input fields: 'Login' (a text box), 'Password' (a text box), and 'Profil' (a dropdown menu with a small downward arrow). To the left of these fields is a white rectangular area containing a 3D rendering of a white humanoid figure holding a large golden key. At the bottom right of the teal area is a white button with the text 'Connexion' in a dark blue font.

Figure 13 : Fenêtre de connexion

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

GestionClient

☐
☐
☐

Civilité

Nom

Prénom

Nationalité

Ville

21:57:23

Adresse

Tel

Email

Profession

Pièce Identification

Ajouter

Modifier

Supprimer

Annuler

Recherche

Tel

	Code Client	Civilité	Nom	Prénom	Nationalité	Ville	Adresse	Tel	Email	Pièce Identification	Profession
07	C0008	Madame	SENE	Mage	Bulgarie	Dakar	Zone A	772307850	anouche...	3256149870	Telecom
08	C0009	Monsieur	DIENG	Ibra	Grèce	Dakar	Sangalkam	773210147	ibra@gma...	14721435795	Web master
09	C0010	Madame	SECK	Penda Gu...	Algérie	Alger	Grand Da...	77632147	pendsi@g...	7539516482	DBA
010	C0012	Monsieur	DIOP	Djibril	Malawi	Dakar	Pikine	774034314	djibson@...	6284571302	Développ...
011	C0013	Madamoi...	LO	DIANKE	Liban	Dakar	Liberté 6	777915947	kédian@g...	54357859510	Commercial
012	C0014	Monsieur	LO	Moussa	Sénégal	Dakar	Hlm gran...	777104456	mussa@g...	2255447780	Retraite
013	C0015	Monsieur	DIOUF	Prince	Koweït	Dakar	Fadia	771442536	prince@g...	3355779910	Développ...

Figure 14 Gestion Des Clients

Mémoire de Fin de Cycle De Licence Professionnelle

**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

Gestion Catégorie de Chambre

Nom Catégorie

Description

Prix de Base

	Code Catégorie	Nom Catégorie	Description	Prix de Base
01	Cat0001	Suite	Chambre de...	150000
02	Cat0002	Villas	Mini maison...	230000
03	Cat0003	Suite Préside...	Chambre	260000

Ajouter Modifier Supprimer Annuler

Figure 15 Gestion de la catégorie des chambres

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

GestionChambres

-
□
×

Nom

Capacité

Catégorie Chambre

Prix de Base

Niveau

Description



[Parcourir...](#)

Enregistrer
Modifier
Supprimer
Annuler

	Numéro Chambre	Nom	Description	Prix de Base	Capacité	Nom Catégorie	Niveau
01	CH0005	Single	Chambre d...	150000	1	Suite	Niveau2
02	CH0006	Twin	Chambre d...	260000	2	Suite Présid...	Niveau3
03	CH0007	Twin	Chambre à ...	150000	2	Suite	Niveau2
04	CH0008	Single Suite	Chambre u...	150000	1	Suite	Niveau2

Figure 16Gestion Des Chambres

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

GestionReservation
- □ ×

23:45:55

Date d'Arrivée

Nom Client

Montant Chambre

Option Réservation

N° Chambre

Nbre de Personnes

Arrangement

Montant Arrange...

Montant Total

Nombre de Jours

Date de Départ

31

31

Informations du Client

Civilité

Prénom

Nationalité

Ville

Listes des Chambres

Adresse

Tel

Email

Profession

Pièce Identification

Ajouter
Modifier
Supprimer
Annuler

	Code Réservation	Date Arrivée	Date Départ	Nombre Nuit	Nombre Personne	Nom Client	Prénom Client	Numéro Chambre	Montant TTC	Type Arrangement	Etat Réservation
01	R19/11/20...	19/11/2016	24/11/2016	5	1	BA	Fatima	CH0008	981000	Pension C...	Fermé
02	R20/11/20...	20/11/2016	24/11/2016	4	2	LO	Ibrahima	CH0005	768800	Demi-Pen...	En Attente
03	R20/11/20...	07/12/2016	09/12/2016	2	2	THIAM	Absa	CH0007	375200	Breton Bre...	Dépassée
04	R20/11/20...	20/11/2016	23/11/2016	3	2	SENE	Mage	CH0006	906600	Demi-Pen...	Occupée
*05											

Figure 17 Gestion Des Réservations

Les points forts de l'application de gestion hôtelière

Le planning : (Version 2.0)

- Simplicité et ergonomie
- Prise de réservation en 1 clic
- Modification, déplacement de réservation directement sur le planning
- Visualisation instantanée des informations de la réservation au survol
- Taux d'occupation en direct

Les réservations :

- Visualisation rapide
- Recherches sur de nombreux critères
- Départs et arrivées
- Prise de réservation par chambre ou par catégorie
- Tarifs définis ou personnalisables avec calculs automatiques selon le prix des chambres
- Classement des tarifs par famille
- Envoi d'email de confirmation

Outils :

- Statistiques avancées : graphiques par jour, semaine, mois, année, saison... (**Version 2.0**)
- Rapports complets (rapport des ventes, gouvernante, arrhes...) (**Version 2.0**)
- Tableau de bord avec les principaux chiffres de votre hôtel (**Version 2.0**)
- Export comptable inclus (**Version 2.0**)
- Sauvegarde automatique

Paramètres :

- Personnalisation totale du logiciel : couleurs, affichage, saisons, modes de paiements...
- Agencement des factures : informations à afficher, positions...

Mais aussi :

- Accès par mot de passe, gestion des utilisateurs avancés : réceptionniste, superviseur, stagiaire...
- Logiciel compatible avec Microsoft Windows 2000, XP, Vista 32 et 64 bits et Seven 32 et 64 bits, Windows 8/8.1 32 et 64 bits, Windows 10 32 et 64 bits!
- Devis et factures pro-forma

II. Gestion de la sécurité

La sécurité des systèmes d'information (SSI) est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité du système d'information. Assurer la sécurité du système d'information est une activité du management du système d'information.

Aujourd'hui, la sécurité est un enjeu majeur pour les entreprises ainsi que pour l'ensemble des acteurs qui l'entourent. Elle n'est plus confinée uniquement au rôle de l'informaticien. Sa finalité sur le long terme est de maintenir la confiance des utilisateurs et des clients. La

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

finalité sur le moyen terme est la cohérence de l'ensemble du système d'information. Sur le court terme, l'objectif est que chacun ait accès aux informations dont il a besoin.

Le système d'information représente un patrimoine essentiel de l'organisation, qu'il convient de protéger. La sécurité informatique consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prév¹. »

La sécurité des systèmes d'information vise les objectifs suivants :

- La disponibilité : Le système doit fonctionner sans faille durant les plages d'utilisation prévues et garantir l'accès aux services et ressources installées avec le temps de réponse attendu.
- L'intégrité : Les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets.
- La confidentialité : Seule les personnes autorisées ont accès aux informations qui leur sont destinées. Tout accès indésirable doit être empêché.

D'autres aspects peuvent aussi être considérés comme des objectifs de la sécurité des systèmes d'information, tels que :

- La traçabilité (ou « Preuve ») : garantie que les accès et tentatives d'accès aux éléments considérés sont tracés et que ces traces sont conservées et exploitables.
- L'authentification : L'identification des utilisateurs est fondamentale pour gérer les accès aux espaces de travail pertinents et maintenir la confiance dans les relations d'échange.
- La non-répudiation et l'imputation : Aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées, et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

Une fois les objectifs de la sécurisation déterminés, les risques pesant sur chacun de ces éléments peuvent être estimés en fonction des menaces. Le niveau global de sécurité des systèmes d'information est défini par le niveau de sécurité du maillon le plus faible. Les précautions et contre-mesures doivent être envisagées en fonction des vulnérabilités propres au contexte auquel le système d'information est censé apporter service et appui.

III. Estimation du cout de la solution

Après une étude réalisée au niveau de l'Hotel NEPTUNE, nous nous sommes rapprochés d'un expert qui nous a permis d'établir l'estimation suivante pour implémenter notre logiciel.

Outils	Quantité	Prix Unitaire	Prix Total
PC Corei5	1	325000	325000
Licence Windows 8	1	40000	40000
Imprimante	1	95000	95000
Serveur	1	500000	500000
Total			960000

IV. Diagramme de Déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.

Mémoire de Fin de Cycle De Licence Professionnelle
**CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION
HOTELIERE CAS DE L'HOTEL NEPTUNE**

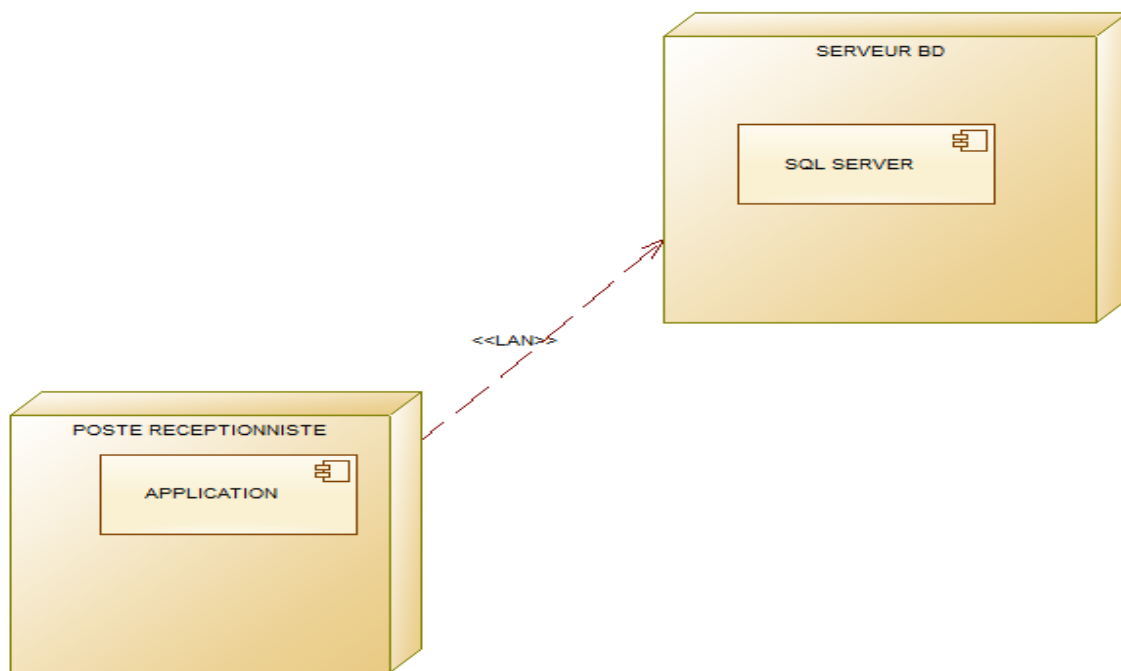


Figure 18 Diagramme de déploiement

CONCLUSION

La perfection n'est pas de ce monde. Le travail de fin de cycle que nous venons de présenter aujourd'hui est le fruit de divers sacrifices caractérisant ainsi notre détermination à en finir une fois pour toutes avec l'étape de graduat que nous considérons comme un premier pas et que d'autres étapes, notamment le master et doctorat pourront s'accomplir dans l'avenir.

Nous avons, à travers les différentes pages, donné une certaine lumière sur la conception et réalisation d'une application de gestion hôtelière tout en respectant les règles de la méthode UML.

Dans l'introduction, nous avons évoqué les causes qui ont militées que nos enquêtes soient réussites. Loin de nous, l'idée de dire que notre solution apportée est la meilleure, nous avons essayé d'y apporter notre contribution et nous laissons aux autres la latitude d'apprécier ce fameux travail.

Dans l'analyse conceptuelle, nous avons dégrossi de façon logique la solution choisie dans l'étude d'opportunité. Nous avons, de façon claire, saisi la portée même du problème car nous avons tenu compte des matériels et de ses contraintes. Expliquant l'enchaînement du menu et la programmation avec ses notions, son langage, et ses méthodes et choix. Ce travail étant réalisé par un être humain peut parfois porter d'illusion. Et c'est à ce titre que nous restons très ouvert aux conseils, suggestions et encouragement qui seront adressés à ce dernier et qui constituerons.

Nous ne pouvons prétendre avoir réalisé un travail sans imperfection, car c'est une œuvre humaine. Mais nous osons croire que le système proposé apportera satisfaction aux responsables de l'hôtel NEPTUNE. Cela étant, les remarques et suggestions des lecteurs sont souhaitées pour l'élaboration d'un édifice scientifique aussi constant qu'on peut le croire.

BIBLIOGRAPHIE

UML 2 par la pratique: études de cas et exercices corrigés

Livre de Pascal Roques

Modélisation avec UML, Eyrolles

Pierre-Alain MULLER, Nathalie GAERTNER

WEBOGRAPHIE

<http://uml.developpez.com>

<https://fr.wikipedia.org>

<http://www.commentcamarche.net>

<http://csharp.net-informations.com>

<http://sql.sh>

TABLE DES MATIERES

Dedicaces	1
Remerciements	2
In memoriam	3
Table des illustrations	4
Glossaire.....	5
INTRODUCTION	6
PREMIERE PARTIE CADRE THEORIQUE ET METHODOLOGIQUE.....	7
CHAPITRE 1 : CADRE THEORIQUE.....	8
I. Présentation Générale de la structure	8
A. Présentation	8
B. Situation Géographique	8
C. Organisation	9
1. Directeur D'hébergement	10
2. Gouvernante	10
3. Chef Réception	10
4. Directeur Technique	11
5. Réceptionniste.....	11
6. Concierge.....	12
7. Femme Chambre	12
II. Présentation du projet	12
A. Contexte	12
B. Problématique	13
C. Objectif.....	13
1. Objectif Principal	13
2. Objectif Spécifique.....	13

D.	Délimitation du Champ d'Etude.....	14
1.	Délimitation spatiale.....	14
2.	Délimitation du point de vue matérielle (Contrainte)	14
CHAPITRE 2 : CADRE METHODOLOGIQUE		15
I.	Méthode d'Investigation	15
A.	Méthode.....	15
B.	Technique.....	15
1.	Technique d'observation directe.....	15
2.	Technique d'interview	15
3.	Technique des questionnaires	15
II.	Etude comparative de quelque méthode d'analyse	15
A.	Présentation	15
1.	MERISE	16
2.	UML.....	16
B.	Comparaison.....	18
C.	Choix de la méthode.....	18
DEUXIEME PARTIE ANALYSE ET CONCEPTION		19
CHAPITRE 1 : ANALYSE.....		20
I.	Prérequis théorique	20
A.	Approche Fonctionnelle	20
B.	Approche Orientée Objet.....	20
C.	Concepts et Formalisme de la méthode choisie	20
II.	Analyse de l'existant	20
A.	Recensement et Analyse Théorique	20
B.	Descriptions des procédures	21
C.	Le Diagramme de cas d'utilisation	21
CHAPITRE 2 : CONCEPTION		24
I.	Le Diagramme de Séquence Système	24

II. Le Diagramme de Classe au Niveau Analyse.....	25
III. Conception Détaillée	26
A. Diagramme de Séquence Détaillée	26
B. Diagramme de Classe niveau Conception	29
TROISIÈME PARTIE IMPLEMENTATION ET DEPLOIEMENT	30
CHAPITRE : 1 IMPLEMENTATION	31
I. Environnement de développement	31
II. Etude de quelque langage de programmation.....	32
A. Définition.....	32
B. Exemple de langages de programmation	32
1. Le langage C	32
2. Le langage C++	32
3. Le langage Java	33
4. Le langage C#	33
5. Le langage PHP	34
C. Choix du langage.....	34
III. Etudes Comparatives de quelques SGBD	35
A. Présentation	35
B. Exemple de SGBD	35
1. MySQL.....	35
2. SQL Server	35
3. Oracle	36
4. Microsoft Access	36
C. Choix du SGBD	36
IV. Architecture Applicative	37
A. Définition.....	37
B. Exemple D'Architecture.....	37
1. Présentation de l'architecture 1-Tier	37
2. Présentation de l'architecture 2-Tier	38

3. Présentation de l'architecture 3-Tier	39
4. Présentation de l'architecture N-Tiers	40
C. Choix de l'architecture.....	40
V. MVC	40
CHAPITRE : 2 DEPLOIEMENT	42
I. Présentation de l'application	42
II. Gestion de la sécurité	48
III. Estimation du cout de la solution	49
IV. Diagramme de Déploiement.....	50
CONCLUSION.....	52
Bibliographie	53
Webographie	54
Table des matières	55